1989

National Semiconductor

400017

DRAM Management

Handbook

- *Dynamic Memory Control*
- *Error Detection And Correction*
- *Application Support*

DRAM Management Handbook

National Semiconductor

A Corporate Dedication to Quality and Reliability

National Semiconductor is an industry leader in the manufacture of high quality, high reliability integrated circuits. We have been the leading proponent of driving down IC defects and extending product lifetimes. From raw material through product design, manufacturing and shipping, our quality and reliability is second to none.

We are proud of our success . . . it sets a standard for others to achieve. Yet, our quest for perfection is ongoing so that you, our customer, can continue to rely on National Semiconductor Corporation to produce high quality products for your design systems.

Charles E. Sporck
President, Chief Executive Officer
National Semiconductor Corporation

Wir fühlen uns zu Qualität und Zuverlässigkeit verpflichtet

National Semiconductor Corporation ist führend bei der Herstellung von integrierten Schaltungen hoher Qualität und hoher Zuverlässigkeit. National Semiconductor war schon immer Vorreiter, wenn es galt, die Zahl von IC Ausfällen zu verringern und die Lebensdauern von Produkten zu verbessern. Vom Rohmaterial über Entwurf und Herstellung bis zur Auslieferung, die Qualität und die Zuverlässigkeit der Produkte von National Semiconductor sind unübertroffen.

Wir sind stolz auf unseren Erfolg, der Standards setzt, die für andere erstrebenswert sind. Auch ihre Ansprüche steigen ständig. Sie als unser Kunde können sich auch weiterhin auf National Semiconductor verlassen.

La Qualité et La Fiabilité:

Une Vocation Commune Chez National Semiconductor Corporation

National Semiconductor Corporation est un des leaders industriels qui fabrique des circuits intégrés d'une très grande qualité et d'une fiabilité exceptionelle. National a été le premier à vouloir faire chuter le nombre de circuits intégrés défectueux et a augmenter la durée de vie des produits. Depuis les matières premières, en passant par la conception du produit sa fabrication et son expédition, partout la qualité et la fiabilité chez National sont sans équivalents.

Nous sommes fiers de notre succès et le standard ainsi défini devrait devenir l'objectif à atteindre par les autres sociétés. Et nous continuons à vouloir faire progresser notre recherche de la perfection; il en résulte que vous, qui êtes notre client, pouvez toujours faire confiance à National Semiconductor Corporation, en produisant des systèmes d'une très grande qualité standard.

Un Impegno Societario di Qualità e Affidabilità

National Semiconductor Corporation è un'industria al vertice nella costruzione di circuiti integrati di altà qualità ed affidabilità. National è stata il principale promotore per l'abbattimento della difettosità dei circuiti integrati e per l'allungamento della vita dei prodotti. Dal materiale grezzo attraverso tutte le fasi di progettazione, costruzione e spedizione, la qualità e affidabilità National non è seconda a nessuno.

Noi siamo orgogliosi del nostro successo che fissa per gli altri un traguardo da raggiungere. Il nostro desiderio di perfezione è d'altra parte illimitato e pertanto tu, nostro cliente, puoi continuare ad affidarti a National Semiconductor Corporation per la produzione dei tuoi sistemi con elevati livelli di qualità.

Charles E. Sporck
President, Chief Executive Officer
National Semiconductor Corporation

DRAM MANAGEMENT

1989 Edition

TRADEMARKS

Following is the most current list of National Semiconductor Corporation's trademarks and registered trademarks.

Abuseable™
Anadig™
ANS-R-TRAN™
APPS™
ASPECT™
Auto-Chem Deflasher™
BCP™
BI-FET™
BI-FET II™
BI-LINE™
BIPLAN™
BLC™
BLX™
Brite-Lite™
BTL™
CheckTrack™
CIM™
CIMBUS™
CLASIC™
Clock✓Chek™
COMBO™
COMBO I™
COMBO II™
COPS™ microcontrollers
Datachecker®
DENSPAK™
DIB™
Digitalker®
DISCERN™
DISTILL™
DNR®
DPVM™
ELSTAR™
Embedded System Processor™
E-Z-LINK™
FACT™
FAIRCAD™
Fairtech™
FAST®
5-Star Service™
GENIX™
GNX™
HAMR™
HandiScan™
HEX 3000™
HPC™
I³L®
ICM™
INFOCHEX™
Integral ISE™
Intelisplay™
ISE™
ISE/06™
ISE/08™
ISE/16™
ISE32™
ISOPLANAR™
ISOPLANAR-Z™
KeyScan™
LMCMOS™
M²CMOS™
Macrobus™
Macrocomponent™
MAXI-ROM®
Meat✓Chek™
MenuMaster™
Microbus™ data bus
MICRO-DAC™
μtalker™
Microtalker™
MICROWIRE™
MICROWIRE/PLUS™
MOLE™
MST™
Naked-8™
National®
National Semiconductor®
National Semiconductor Corp.®
NAX 800™
Nitride Plus™
Nitride Plus Oxide™
NML™
NOBUS™
NSC800™
NSCISE™
NSX-16™
NS-XC-16™
NTERCOM™
NURAM™
OXISS™
P²CMOS™
PC Master™
Perfect Watch™
Pharma✓Chek™
PLAN™
PLANAR™
Plus-2™
Polycraft™
POSilink™
POSitalker™
Power + Control™
POWERplanar™
QUAD3000™
QUIKLOOK™
RAT™
RTX16™
SABR™
Script✓Chek™
SCX™
SERIES/800™
Series 900™
Series 3000™
Series 32000®
Shelf✓Chek™
SofChek™
SONIC™
SPIRE™
Staggered Refresh™
STAR™
Starlink™
STARPLEX™
Super-Block™
SuperChip™
SuperScript™
SYS32™
TapePak®
TDS™
TeleGate™
The National Anthem®
Time✓Chek™
TINA™
TLC™
Trapezoidal™
TRI-CODE™
TRI-POLY™
TRI-SAFE™
TRI-STATE®
TURBOTRANSCEIVER™
VIP™
VR32™
WATCHDOG™
XMOS™
XPU™
Z STAR™
883B/RETS™
883S/RETS™

PAL® and PALASM™ are trademarks of and are used under license from Monolithic Memories, Inc.

Z80® and Z80A® are registered trademarks of Zilog Corporation.

Z8000™ is a trademark of Zilog Corporation.

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation 2900 Semiconductor Drive, P.O. Box 58090, Santa Clara, California 95052-8090 (408) 721-5000
TWX (910) 339-9240

National Semiconductor

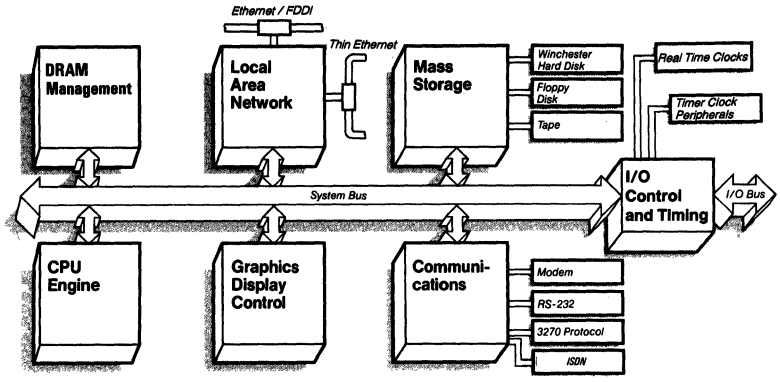
DRAM Management Introduction

Today's large Dynamic Random Access Memory (DRAM) arrays require sophisticated high performance devices to provide timing access arbitration on board drive and control. National Semiconductor offers the broadest range of DRAM controllers with the highest "No-waitstate" performance available on the market. Controllers are available in Junction Isolated LS, Oxide Isolated ALS, and double metal CMOS for DRAMs from 64k bit through 4M bit devices, supporting memory arrays up to 64 Mbyte in size with only one LSI/VLSI device. For critical applications, National Semiconductor has developed several 16- and 32-bit Error Checking and Correction (ECC) devices to provide maximum data integrity.

The DRAM Management Handbook contains complete product information. This includes the largest number and most complete set of DRAM control and ECC products, peripheral support devices and application notes detailing complete DRAM memory system design.

National Semiconductor

Introduction to VLSI Products



TL/XX/0058–1

National Semiconductor VLSI products include complex peripheral circuits designed to serve a variety of applications. The VLSI products are especially well suited for microcomputer and microprocessor systems such as graphics workstations, personal computers, and many others. National Semiconductor VLSI devices are fully described in a series of databooks and handbooks.

Among the books are the following titles:

MASS STORAGE

The National Semiconductor family of mass storage interface products offers the industry's highest performance and broadest range of products for Winchester hard disks, high performance ESDI and SCSI hard disks and floppy disks. Combined with CLASIC™, analog and high performance microcontroller devices, these products offer unparalleled solutions for integration.

DRAM MANAGEMENT

National Semiconductor offers the broadest range of DRAM controllers with the highest "No-waitstate" performance available on the market. For critical applications, National Semiconductor has developed several 16- and 32-bit Error Checking and Correction (ECC) devices to provide maximum data integrity.

MICROCONTROLLER

As one of the broadest cost/performance product offerings in the industry today, National's microcontrollers provide the intelligence required for high performance applications such as laser printers, ISDN terminal adapters, floppy disks and SCSI hard disks. Complete support tools are available, including applications specific software, Designer's Kits, emulators, simulators, and development systems. Whether the application demands 4-, 8- or 16-bit performance, National has the right embedded control solution.

LOCAL AREA NETWORKS, DATA COMMUNICATIONS, UARTS

National Semiconductor provides a complete three-chip solution for an entire IEEE 802.3 standard for Ethernet/Thin Ethernet LANs. National Semiconductor offers a completely integrated solution for the IBM 370 class mainframes, System 3X and AS/400 systems for physical layer front end and processing of the IBM 3270/3299 "coaxial" and 5250 "twinaxial" protocols. National's family of UARTs provides high performance, low power serial data input/output interface.

INTERFACE

To drive the communications lines, National Semiconductor has drivers and receivers designed to meet all the major standards such as RS-232, RS-422, and RS-485.

GRAPHICS

The graphics chip set is designed to provide the highest level of performance with minimum demands and loading on the system CPU. The graphics system may be expanded to any number of color planes with virtually unlimited resolution.

REAL TIME CLOCKS

The RTC family provides a simple μP bus compatible interface to any system requiring accurate, reliable, on-going real time and calender functions.

EMBEDDED SYSTEMS PROCESSORS

National's Embedded System Processor™ family offers the most complete solution to 32-bit embedded processor needs via CPUs, slave processors, system peripherals, evaluation/development tools and software.

Our total product system solution approach includes the hardware, software, and development support products necessary for your design. Evaluation board, in-system emulator, software development tools, and third party software are available now.

National Semiconductor

Product Status Definitions

Definition of Terms

Data Sheet Identification	Product Status	Definition
Advance Information	Formative or In Design	This data sheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	This data sheet contains preliminary data, and supplementary data will be published at a later date. National Semiconductor Corporation reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
No Identification Noted	Full Production	This data sheet contains final specifications. National Semiconductor Corporation reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

National Semiconductor Corporation reserves the right to make changes without further notice to any products herein to improve reliability, function or design. National does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights, nor the rights of others.

Table of Contents

Table of Contents (Continued)

Alpha-Numeric Index

Alpha-Numeric Index (Continued)

Section 1

Dynamic Memory Control

Section 1 Contents

DRAM Controller Master Selection Guide

The data below is intended to highlight the key differentiable features of each DRAM Controller/Driver offered by National Semiconductor. All NSC DRAM controllers integrate onboard delay line timing, high capacitive drive, row/column muxing logic, refresh counter, row and column input latches, memory bank select logic. As a result of the family feature commonality, most devices offer pin for pin up/downward compatiblity. Beyond this however, the process and design differences between the devices result in a broad selection of feature and performance options for the best system fit.

Device # & Speed Options	DRAMS Supported	Process	Typ I_{CC}	A.C. Specified Word Width	Max $\overline{RAS}$ to $\overline{CAS}$ Out *Fast Mode | Slow Mode	Guaranteed Row Address Hold *Fast Mode | Slow Mode	V_{CC}	Operating Temp Range	Package	Page No.
DP8408A A-2 A-3	16, 64k	Junction Isolated (S)	210 mA	4 Banks of 16 Bit Data w/ 6 Bit ECC ea.	105 ns/125 ns 85 ns/100 ns 120 ns/145 ns	20 ns/30 ns 12 ns/20 ns 20 ns/30 ns	+5V ±5%	0°–70°C 0°–85°C	48N 48D	1-4
DP8409A A-2 A-3	16, 64, 256k	Junction Isolated (S)	210 mA	4 Banks of 16 Bit Data w/ 6 Bit ECC ea.	105 ns/125 ns 85 ns/100 ns 120 ns/145 ns	20 ns/30 ns 12 ns/20 ns 20 ns/30 ns	+5V ±5%	0°–70°C 0°–85°C	48N 48D 68V	1-22
DP8417-80 -70	16, 64, 256k	Oxide Isolated (ALS)	150 mA	4 Banks of 16 Bit Data w/ 6 Bit ECC ea.	63 ns/80 ns 50 ns/72 ns	15 ns/25 ns 15 ns/25 ns	+5V ±10%	0°–70°C −40°–+85°C −55°–+125°C	48 N 48D 68V	1-44
DP8418-80 -70	16, 64, 256k	Oxide Isolated (ALS)	150 mA	2 Banks of 32 Bit Data w/ 7 Bit ECC ea.	63 ns/80 ns 50 ns/72 ns	15 ns/25 ns 15 ns/25 ns	+5V ±10%	0°–70°C −40°–+85°C −55°–+125°C	48 N 48D 68V	1-44
DP8419-80 -70	16, 64, 256k	Oxide Isolated (ALS)	150 mA	4 Banks of 16 Bit Data w/ 6 Bit ECC ea.	63 ns/80 ns 50 ns/72 ns	15 ns/25 ns 15 ns/25 ns	+5V ±10%	0°–70°C −40°–+85°C −55°–+125°C	48 N 48D 68V	1-44
DP8420A, DP8421A & DP8422A	16, 64, 256k, 1 Mega Bit, 4 Mega Bit	2μ CMOS	5 mA	2 Banks of 32 Bit Data w/ 7 Bit ECC ea.	53 ns/63 ns	15 ns/25 ns	+5V ±10%	0°–70°C −40°–+85°C −55°–+125°C	68V	1-92
DP8428-80 -70	16, 64, 256k & 1 Mega Bit	Oxide Isolated (ALS)	150 mA	2 Banks of 32 Bit Data w/ 7 Bit ECC ea.	63 ns/80 ns 50 ns/72 ns	15 ns/25 ns 15 ns/25 ns	+5V ±10%	0°–70°C −40°–+85°C −55°–+125°C	52D 68V	1-69
DP8429-80 -70	16, 64, 256k & 1 Mega Bit	Oxide Isolated (ALS)	150 mA	4 Banks of 16 Bit Data w/ 6 Bit ECC ea.	63 ns/80 ns 50 ns/72 ns	15 ns/25 ns 15 ns/25 ns	+5V ±10%	0°–70°C −40°–+85°C −55°–+125°C	52D 68V	1-69

*All AC valves shown factor in worst case loading (including all ouputs switching simultaneously), operating temperature, and V_{CC} supply variables. All delays assume the use of National's on-board automatic timing and delay line logic although external delay line control timing is allowed and supported.

National Semiconductor

DP8408A Dynamic RAM Controller/Driver

General Description

Dynamic memory system designs, which formerly required several support chips to drive the memory array, can now be implemented with a single IC . . . the DP8408A Dynamic RAM Controller/Driver. The DP8408A is capable of driving all 16k and 64k Dynamic RAMs (DRAMs). Since the DP8408A is a one-chip solution (including capacitive-load drivers), it minimizes propagation delay skews, the major performance disadvantage of multiple-chip memory drive and control.

The DP8408A's 6 modes of operation offer a wide selection of DRAM control capabilities. Memory access may be controlled externally or on-chip automatically; an on-chip refresh counter makes refreshing less complicated.

The DP8408A is a 48-pin DRAM Controller/Driver with 8 multiplexed address outputs and control signals. It consists of two 8-bit address latches, an 8-bit refresh counter, and control logic. All output drivers are capable of driving 500 pF loads with propagation delays of 25 ns. The DP8408A timing parameters are specified driving the typical load capacitance of 88 DRAMs, including trace capacitance.

The DP8408A has 3 mode-control pins: M2, M1, and M0, where M2 is in general $\overline{\text{REFRESH}}$. These 3 pins select 6 modes of operation. Inputs B1 and B0 in the memory access modes (M2 = 1), are select inputs which select one of four $\overline{\text{RAS}}$ outputs. During normal access, the 8 address outputs can be selected from the Row Address Latch or the Column Address Latch. During refresh, the 8-bit on-chip refresh counter is enabled onto the address bus and in this mode all $\overline{\text{RAS}}$ outputs are selected, while $\overline{\text{CAS}}$ is inhibited.

The DP8408A can drive up to 4 banks of DRAMs, with each bank comprised of 16k's, or 64k's. Control signal outputs $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and $\overline{\text{WE}}$ are provided with the same drive capability. Each $\overline{\text{RAS}}$ output drives one bank of DRAMs so that the four $\overline{\text{RAS}}$ outputs are used to select the banks, while $\overline{\text{CAS}}$, $\overline{\text{WE}}$, and the multiplexed addresses can be connected to all of the banks of DRAMs. This leaves the non-selected banks in the standby mode (less than one tenth of the operating power) with the data outputs in TRI-STATE®. Only the bank with its associated $\overline{\text{RAS}}$ low will be written to or read from.

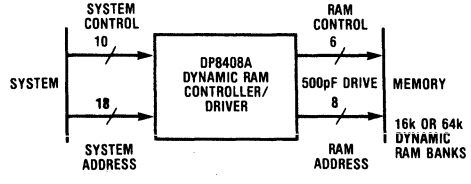
Operational Features

- All DRAM drive functions on one chip—minimizes skew on outputs, maximizes AC performance
- On-chip capacitive-load drives (specified to drive up to 88 DRAMs)
- Drive directly all 16k and 64k DRAMs
- Capable of addressing 64k and 256k words
- Propagation delays of 25 ns typical at 500 pF load
- $\overline{\text{CAS}}$ goes low automatically after column addresses are valid if desired
- Auto Access mode provides $\overline{\text{RAS}}$, Row to Column, select, then $\overline{\text{CAS}}$ automatically and fast
- $\overline{\text{WE}}$ follows $\overline{\text{WIN}}$ unconditionally—offering READ, WRITE or READ-MODIFY-WRITE cycles
- On-chip 8-bit refresh counter with selectable End-of-Count (127 or 255)
- End-of-Count indicated by RF I/O pin going low at 127 or 255
- Low input on RF I/O resets 8-bit refresh counter
- $\overline{\text{CAS}}$ inhibited during refresh cycle
- Fall-through latches on address inputs controlled by ADS
- TRI-STATE outputs allow multi-controller addressing of memory
- Control output signals go high-impedance logic "1" when disabled for memory sharing
- Power-up: counter reset, control signals high, address outputs TRI-STATE, and End-of-Count set to 127

Mode Features

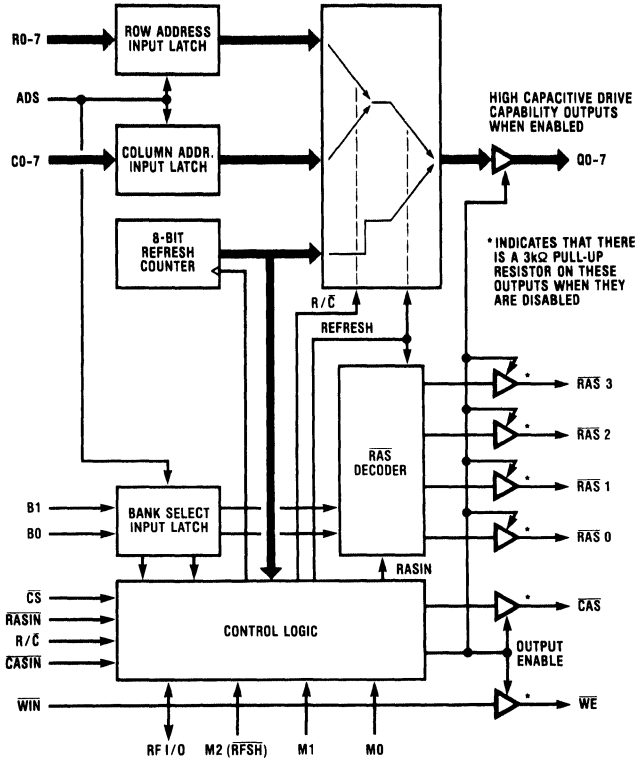
- 6 modes of operation: 3 access, 1 refresh, and 2 set-up
- 2 externally controlled modes: 1 access (Mode 4) and 1 refresh (Modes 0, 1 , 2)
- 2 auto-access modes $\overline{\text{RAS}} \rightarrow \text{R}/\overline{\text{C}} \rightarrow \overline{\text{CAS}}$ automatic, with t_{RAH} = 20 or 30 ns minimum (Modes 5, 6)
- Externally controlled All-$\overline{\text{RAS}}$ Access modes for memory initialization (Mode 3)
- End-of-Count value of Refresh Counter set by B1 and B0 (Mode 7)

DP8408A Interface Between System & DRAM Banks



TL/F/8408–1

Block Diagram



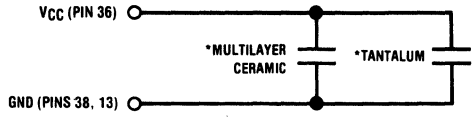
TL/F/8408–2

TABLE I. DP8408A Mode Select Options

Mode	($\overline{RFSH}$) M2	M1	M0	Mode of Operation	Conditions
0	0	0	0	Externally Controlled Refresh	RF I/O = $\overline{EOC}$
1	0	0	1		
2	0	1	0		
3	0	1	1	Externally Controlled All-$\overline{RAS}$ Write	All-$\overline{RAS}$ Active
4	1	0	0	Externally Controlled Access	Active $\overline{RAS}$ defined by Table II
5	1	0	1	Auto Access, Slow t_{RAH}	Active $\overline{RAS}$ defined by Table II
6	1	1	0	Auto Access, Fast t_{RAH}	Active $\overline{RAS}$ defined by Table II
7	1	1	1	Set End of Count	See Table III for Mode 7

Pin Definitions

V_{CC}, GND, GND—V_{CC} = 5V ±5%. The three supply pins have been assigned to the center of the package to reduce voltage drops, both DC and AC. There are also two ground pins to reduce the low level noise. The second ground pin is located two pins from V_{CC}, so that decoupling capacitors can be inserted directly next to these pins. It is important to adequately decouple this device, due to the high switching currents that will occur when all 8 address bits change in the same direction simultaneously. A recommended solution would be a 1 μF multilayer ceramic capacitor in parallel with a low-voltage tantalum capacitor, both connected as close as possible to pins 36 and 38 to reduce lead inductance. See *Figure* below.



TL/F/8408–3

*Capacitor values should be chosen depending on the particular application.

R0–R7: Row Address Inputs.

C0–C7: Column Address Inputs.

Q0–Q7: Multiplexed Address Outputs—Selected from the Row Address Input Latch, the Column Address Input Latch, or the Refresh Counter.*

$\overline{RASIN}$: Row Address Strobe Input—Enables selected $\overline{RAS}_n$ output when M2 ($\overline{RFSH}$) is high, or all $\overline{RAS}_n$ outputs when $\overline{RFSH}$ is low.

R/$\overline{C}$: Row/Column Select Input—Selects either the row or column address input latch onto the output bus.

$\overline{CASIN}$: Column Address Strobe Input—Inhibits $\overline{CAS}$ output when high in Modes 4 and 3. In Mode 6 it can be used to prolong $\overline{CAS}$ output.

ADS: Address (Latch) Strobe Input—Row Address, Column Address, and Bank Select Latches are fall-through with ADS high; Latches on high-to-low transition.

$\overline{CS}$: Chip Select Input—TRI-STATE the Address Outputs and puts the control signal into a high-impedance logic "1" state when high (except in Mode 0); enables all outputs when low.

M0, M1, M2: Mode Control Inputs—These 3 control pins determine the 6 major modes of operation of the DP8408A as depicted in Table I.

RF I/O—The I/O pin functions as a Reset Counter Input when set low from an external open-collector gate, or as a flag output. The flag goes active-low when M2 = 0 and the End-of-Count output is at 127 or 255 (see Table III).

$\overline{WIN}$: Write Enable Input.

$\overline{WE}$: Write Enable Output—Buffered output from $\overline{WIN}$.*

$\overline{CAS}$: Column Address Strobe Output—In Modes 5 and 6, $\overline{CAS}$ goes low following valid column address. In Modes 3 and 4, it transitions low after R/$\overline{C}$ goes low, or follows $\overline{CASIN}$ going low if R/$\overline{C}$ is already low. $\overline{CAS}$ is high during refresh.*

$\overline{RAS}$ 0–3: Row Address Strobe Outputs—Selects a memory bank decoded from B1 and B0 (see Table II), if $\overline{RFSH}$ is high. If $\overline{RFSH}$ is low, all banks are selected.*

B0, B1: Bank Select Inputs—Strobed by ADS. Decoded to enable one of the $\overline{RAS}$ outputs when $\overline{RASIN}$ goes low. Also used to define End-of-Count in Mode 7 (Table III).

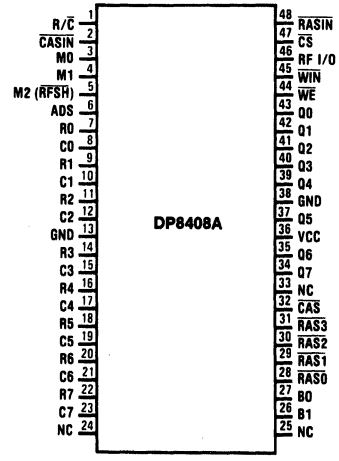
*These outputs may need damping resistors to prevent overshoot, undershoot. See AN-305 "Precautions to Take When Driving Memories."

TABLE II. Memory Bank Decode

Bank Select (Strobed by ADS)		Enabled $\overline{RAS}_n$
B1	B0	
0	0	$\overline{RAS}_0$
0	1	$\overline{RAS}_1$
1	0	$\overline{RAS}_2$
1	1	$\overline{RAS}_3$

Connection Diagram

Dual In-Line Package



NC = No Connection

TL/F/8408–4

Top View

Order Number DP8408AD, DP8408AN or DP8408AN-3
See NS Package Number D48A or N48A

Conditions for all Modes

INPUT ADDRESSING

The address block consists of a row-address latch, a column-address latch, and a resettable refresh counter. The address latches are fall-through when ADS is high and latch when ADS goes low. If the address bus contains valid addresses until after the valid address time, ADS can be permanently high. Otherwise ADS must go low while the addresses are still valid.

In normal memory access operation, $\overline{RASIN}$ and R/$\overline{C}$ are initially high. When the address inputs are enabled into the address latches, the row addresses appear on the Q outputs. The address strobe also inputs the bank-select address, (B0 and B1). If $\overline{CS}$ is low, all outputs are enabled. When $\overline{CS}$ is transitioned high, the address outputs go TRI-STATE and the control outputs first go high through a low impedance, and then are held by an on-chip high impedance. This allows output paralleling with other DP8408As for multi-addressing. All outputs go active about 50 ns after the chip is selected again. If $\overline{CS}$ is high, and a refresh cycle begins, all the outputs become active until the end of the refresh cycle.

DRIVE CAPABILITY

The DP8408A has timing parameters that are specified with up to 600 pF loads. In a typical memory system this is equivalent to about 88, 5V-only DRAMs, with trace lengths kept to a minimum. Therefore, the chip can drive four banks each of 16 or 22 bits, or two banks of 32 or 39 bits, or one bank of 64 or 72 bits.

Less loading will slightly reduce the timing parameters, and more loading will increase the timing parameters, according to the graph of *Figure 6*. The AC performance parameters are specified with the typical load capacitance of 88 DRAMs. This graph can be used to extrapolate the variations expected with other loading.

Because of distributed trace capacitance and inductance and DRAM input capacitance, current spikes can be created, causing overshoots and undershoots at the DRAM inputs that can change the contents of the DRAMs or even destroy them. To remove these spikes, a damping resistor (low inductance, carbon) can be inserted between the DP8408A driver outputs and the DRAMs, as close as possible to the DP8408A. The values of the damping resistors may differ between the different control outputs; $\overline{RAS}$'s $\overline{CAS}$, Q's and $\overline{WE}$. The damping resistors should be determined by the first prototypes (not wire-wrapped due to larger distributed capacitance and inductance). The best values for the damping resistors are the critical values giving a critically damped transition on the control outputs. Typical values for the damping resistors will be between 15Ω and 100Ω, the lower the loading the higher the value. (For more information, see AN-305 "Precautions to Take When Driving Memories.")

DP8408A DRIVING ANY 16K OR 64K DRAMS

The DP8408A can drive any 16k or 64k DRAMs. All 16k DRAMs are basically the same configuration, including the newer 5V-only version. Hence, in most applications, different manufacturers' DRAMs are interchangeable (for the same supply-rail chips), and the DP8408A can drive all 16k DRAMS (see *Figure 1a*).

There are three basic configurations for the 5V-only 64k DRAMs: a 128-row by 512-column array with an on-RAM refresh counter, a 128-row by 512-column array with no on-RAM refresh counter, and a 256-row by 256-column array with no on-RAM refresh counter. The DP8408A can drive all three configurations, and at the same time allows them all to be interchangeable (as shown in *Figure 1b* and *1c*), providing maximum flexibility in the choice of DRAMs. Since the 8-bit on-chip refresh counter can be used as a 7-bit refresh counter for the 128-row configuration, or as an 8-bit refresh counter for the 256-row configuration, the on-RAM refresh counter (if present) is never used. As long as 128 rows are refreshed every 2 ms (i.e. 256 rows in 4 ms) all DRAM types are correctly refreshed.

When the DP8408A is in a refresh mode, the RF I/O pin indicates that the on-chip refresh counter has reached its end-of-count. This end-of-count is selectable as 127 or 255 to accommodate 16k or 64k DRAMs, respectively. Although the end-of-count may be chosen to be either of these values, the counter is not reset and always counts to 255 before rolling over to zero.

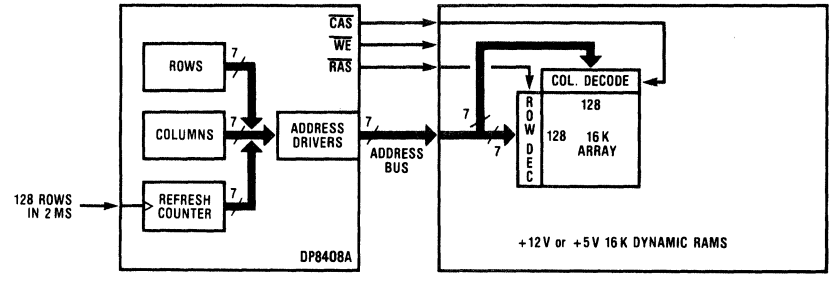
READ, WRITE AND READ-MODIFY-WRITE CYCLES

The output signal, $\overline{WE}$, determines what type of memory access cycle the memory will perform. If $\overline{WE}$ is kept high while $\overline{CAS}$ goes low, a read cycle occurs. If $\overline{WE}$ goes low before $\overline{CAS}$ goes low, a write cycle occurs and data at DI (DRAM input data) is written into the DRAM as $\overline{CAS}$ goes low. If $\overline{WE}$ goes low later than t_{CWD} after $\overline{CAS}$ goes low, first a read occurs and DO (DRAM output data) becomes valid; then data DI is written into the same address in the DRAM when $\overline{WE}$ goes low. In this read-modify-write case, DI and DO cannot be linked together. The type of cycle is therefore controlled by $\overline{WE}$, which follows $\overline{WIN}$.

POWER-UP INITIALIZE

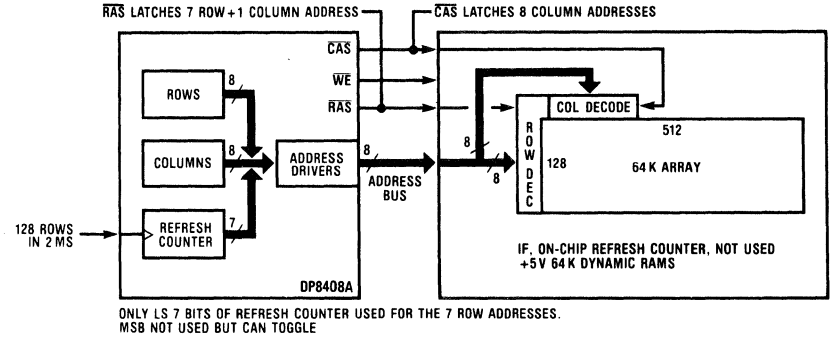
When V_{CC} is first applied to the DP8408A, an initialize pulse clears the refresh counter, the internal control flip-flops, and sets the End-of-Count of the refresh counter to 127 (which may be changed via Mode 7). As V_{CC} increases to about 2.3V, it holds the output control signals at a level of one Schottky diode-drop below V_{CC}, and the output address to TRI-STATE. As V_{CC} increases above 2.3V, control of these outputs is granted to the system.

DP8408A Driving any 16k or 64k Dynamic RAMs



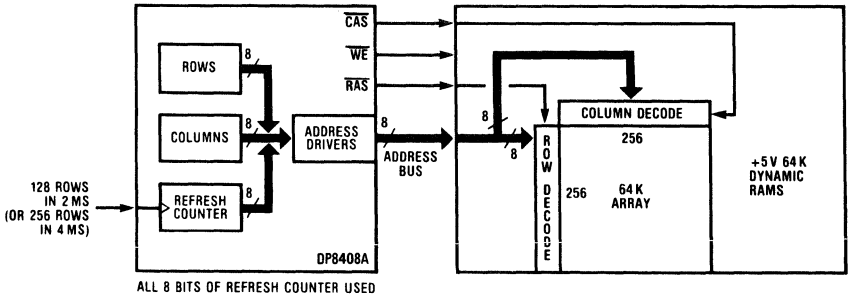
TL/F/8408–5

FIGURE 1a. DP8408A with any 16k DRAMS



TL/F/8408–6

FIGURE 1b. DP8408A with 128 Row × 512 Column 64k DRAM



TL/F/8408–7

FIGURE 1c. DP8408A with 256 × 256 Column 64k DRAM

Functional Mode Descriptions

Note: All delay parameters stated in text refer to the DP8408A. Substitute the respective delay numbers for the DP8408-2 or DP8408-3 when using these devices.

MODES 0, 1, 2 — EXTERNALLY CONTROLLED REFRESH

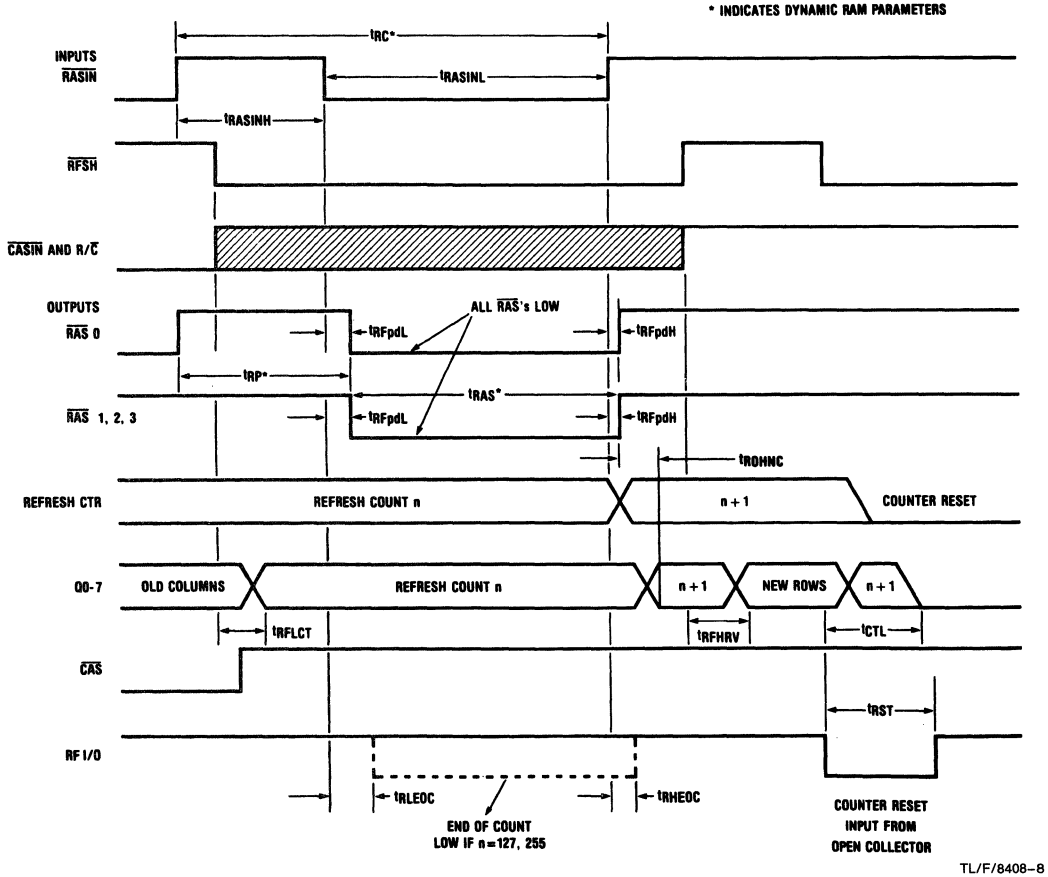
In this mode, the input address latches are disabled from the address outputs and the refresh counter is enabled. When $\overline{RAS}$ occurs, the enabled row in the DRAM is refreshed. In the Externally Controlled Refresh mode, all $\overline{RAS}$ outputs are enabled following $\overline{RASIN}$, and $\overline{CAS}$ is inhibited. This refreshes the same row in all four banks. The refresh counter increments when either $\overline{RASIN}$ or $\overline{RFSH}$ goes low-to-high after a refresh. RF I/O goes low when the count is 127 or 255, as set by End-of-Count (see Table III), with $\overline{RASIN}$ and $\overline{RFSH}$ low. To reset the counter to all zeros, RF I/O is set low through an external open-collector driver.

During refresh, $\overline{RASIN}$ and RFSH must be skewed transitioning low such that the refresh address is valid on the address outputs of the controller before the $\overline{RAS}$ outputs go low. The amount of time that RFSH should go low before $\overline{RASIN}$ does depends on the capacitive loading of the address and $\overline{RAS}$ lines. For the load specified in the switching characteristics of this data sheet, 10 ns is sufficient. Refer to *Figure 2*.

To perform externally controlled burst refresh, $\overline{RASIN}$ is toggled while RFSH is held low. The refresh counter increments with $\overline{RASIN}$ going low to high, so that the DRAM rows are refreshed in succession by $\overline{RASIN}$ going high to low.

MODE 3 — EXTERNALLY CONTROLLED ALL-$\overline{RAS}$ WRITE

This mode is useful at system initialization. The memory address is provided by the processor, which also performs the incrementing. All four $\overline{RAS}$ outputs follow $\overline{RASIN}$ (supplied by the processor), strobing the row address into the DRAMs. R/$\overline{C}$ can now go low, while $\overline{CASIN}$ may be used to control $\overline{CAS}$ (as in the Externally Controlled Access mode), so that $\overline{CAS}$ strobes the column address contents into the DRAMs. At this time $\overline{WE}$ should be low, causing the data to be written into all four banks of DRAMs. At the end of the write cycle, the input address is incremented and latched by the DP8408A for the next write cycle.



TL/F/8408–8

FIGURE 2. External Control Refresh Cycle (MODES 0, 1, 2)

Functional Mode Descriptions (Continued)

MODE 4 — EXTERNALLY CONTROLLED ACCESS

This mode facilitates externally controlling all access-timing parameters associated with the DRAMs. The application of modes 0 and 4 are shown in *Figure 3*.

Output Address Selection

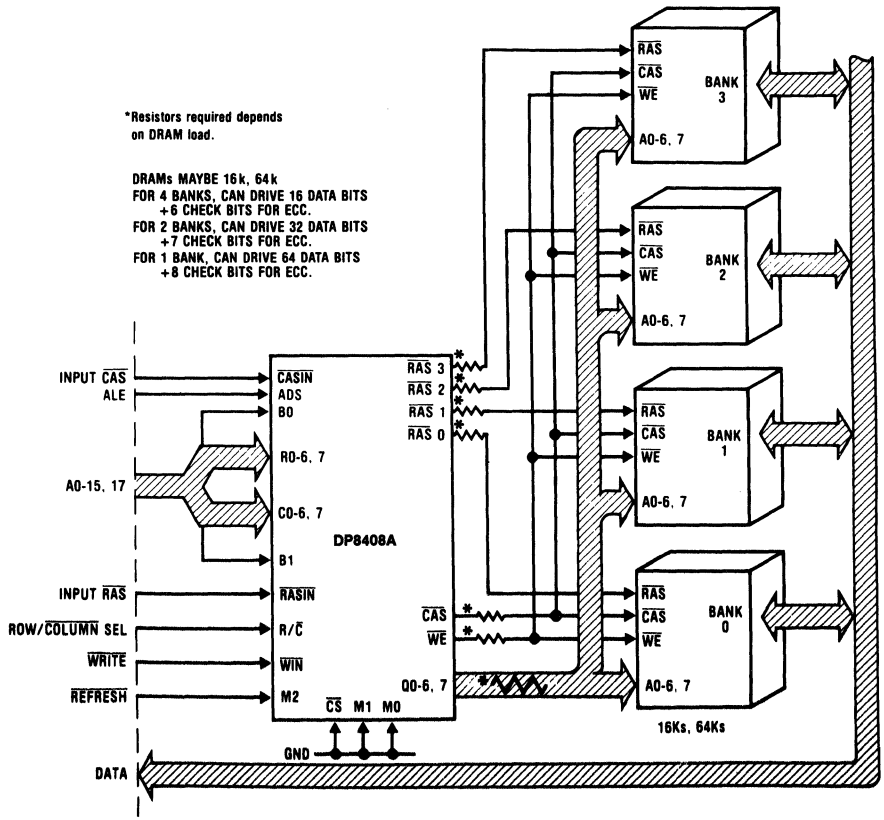
Refer to *Figure 4a*. With M2 ($\overline{RFSH}$) and R/$\overline{C}$ high, the row address latch contents are transferred to the multiplexed address bus output Q0–Q7, provided $\overline{CS}$ is set low. The column address latch contents are output after R/$\overline{C}$ goes low. $\overline{RASIN}$ can go low after the row addresses have been set up on Q0–Q7. This selects one of the $\overline{RAS}$ outputs, strobing the row address on the Q outputs into the desired bank of memory. After the row-address hold-time of the DRAMs, R/$\overline{C}$ can go low so that about 40 ns later column addresses appear on the Q outputs.

Automatic $\overline{CAS}$ Generation

In a normal memory access cycle $\overline{CAS}$ can be derived from inputs $\overline{CASIN}$ or R/$\overline{C}$. If $\overline{CASIN}$ is high, then R/$\overline{C}$ going low switches the address output drivers from rows to columns. $\overline{CASIN}$ then going low causes $\overline{CAS}$ to go low approximately 40 ns later, allowing $\overline{CAS}$ to occur at a predictable time (see *Figure 4b*). If $\overline{CASIN}$ is low when R/$\overline{C}$ goes low, $\overline{CAS}$ will be automatically generated, following the row to column transition by about 20 ns (see *Figure 4a*). Most DRAMs have a column address set-up time before $\overline{CAS}$ (t_{ASC}) of 0 ns or −10 ns. In other words, a t_{ASC} greater than 0 ns is safe. This feature reduces timing-skew problems, thereby improving access time of the system.

Fast Memory Access

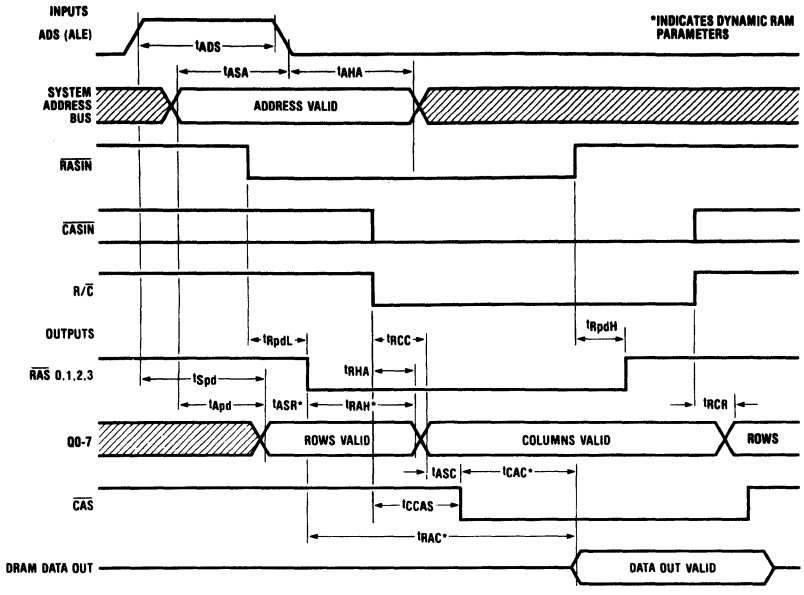
AC parameters t_{DIF1}, t_{DIF2} may be used to determine the minimum delays required between $\overline{RASIN}$, R/$\overline{C}$, and $\overline{CASIN}$ (see Application Brief 9; "Fastest DRAM Access Mode").



TL/F/8408–9

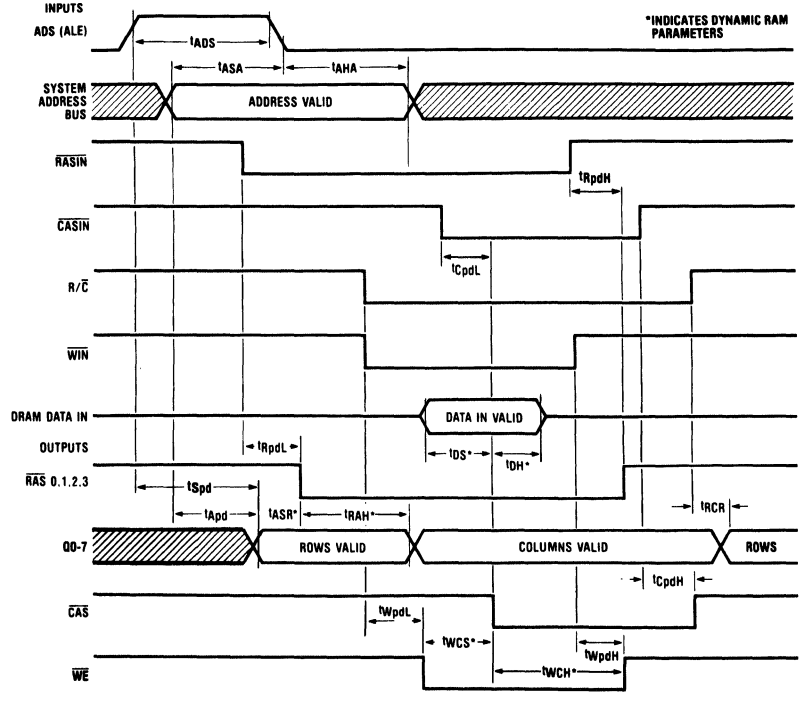
FIGURE 3. Typical Application of DP8408A Using Externally Controlled Access and Refresh in Modes 0 and 4

Timing Diagrams



TL/F/8408–10

FIGURE 4a. Read Cycle Timing (Mode 4)



TL/F/8408–11

FIGURE 4b. Write Cycle Timing (Mode 4)

Functional Mode Descriptions (Continued)

MODE 5—AUTOMATIC ACCESS

The Auto Access mode has two advantages over the Externally Controlled Access mode, due to the fact that all outputs except $\overline{WE}$ are initiated from $\overline{RASIN}$. First, inputs R/$\overline{C}$ and $\overline{CASIN}$ are unnecessary. Secondly, because the output control signals are derived internally from one input signal ($\overline{RASIN}$), timing-skew problems are reduced, thereby reducing memory access time substantially or allowing use of slower DRAMs. The automatic access features of Mode 5 (and Mode 6) of the DP8408A make DRAM accessing appear essentially "static".

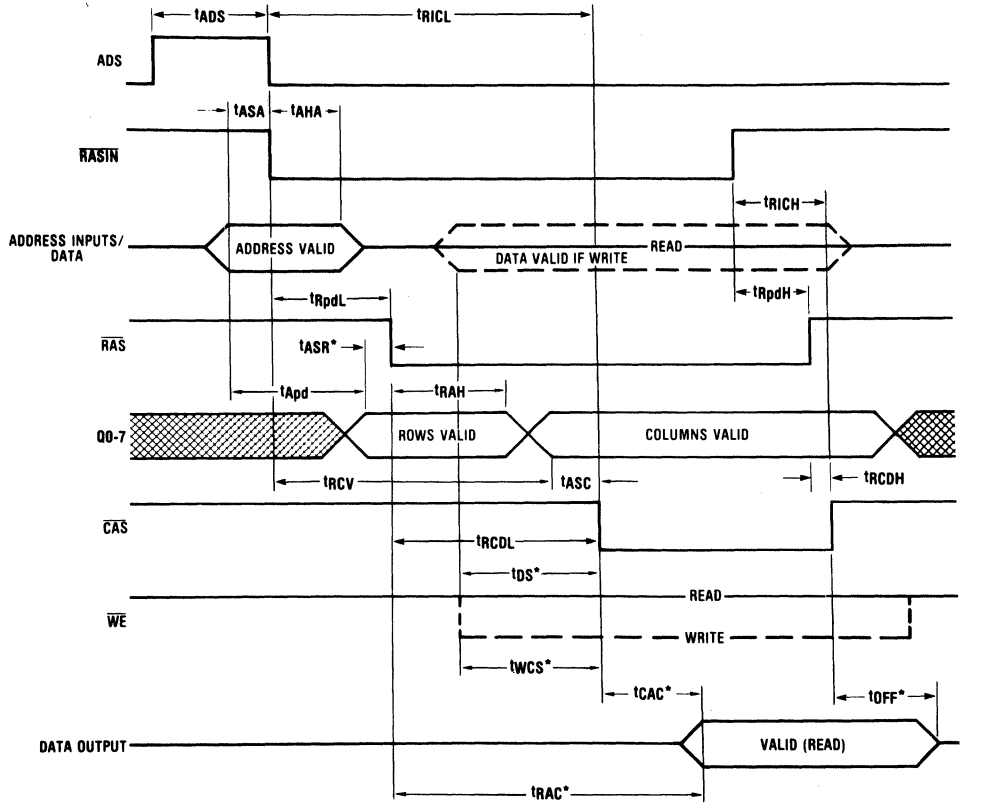
AUTOMATIC ACCESS CONTROL

The major disadvantage of DRAMs compared to static RAMs is the complex timing involved. First, a $\overline{RAS}$ must occur with the row address previously set up on the multiplexed address bus. After the row address has been held for t_{RAH}, (the Row-Address hold-time of the DRAM), the column address is set up and then $\overline{CAS}$ occurs. This is all performed automatically by the DP8408A in this mode.

Provided the input address is valid as ADS goes low, $\overline{RASIN}$ can go low any time after ADS. This is because the selected $\overline{RAS}$ occurs typically 27 ns later, by which time the row address is already valid on the address output of the DP8408A. The Address Setup-Up time (t_{ASR}), is 0 ns on most DRAMs. The DP8408A in this mode (with ADS and $\overline{RASIN}$ edges simultaneously applied) produces a minimum t_{ASR} of 0 ns. This is true provided the input address was valid t_{ASA} before ADS went low (see *Figure 5a*).

Next, the row address is disabled after t_{RAH} (30 ns minimum); in most DRAMs, t_{RAH} minimum is less than 30 ns. The column address is then set up and t_{ASC} later, $\overline{CAS}$

Timing Diagram



TL/F/8408–12

*Indicates Dynamic RAM Parameters

FIGURE 5a. Modes 5, 6 Timing ($\overline{CASIN}$) High in Mode 6

Functional Mode Descriptions (Continued)

occurs. The only other control input required is $\overline{WIN}$. When a write cycle is required, $\overline{WIN}$ must go low at least 30 ns before $\overline{CAS}$ is output low.

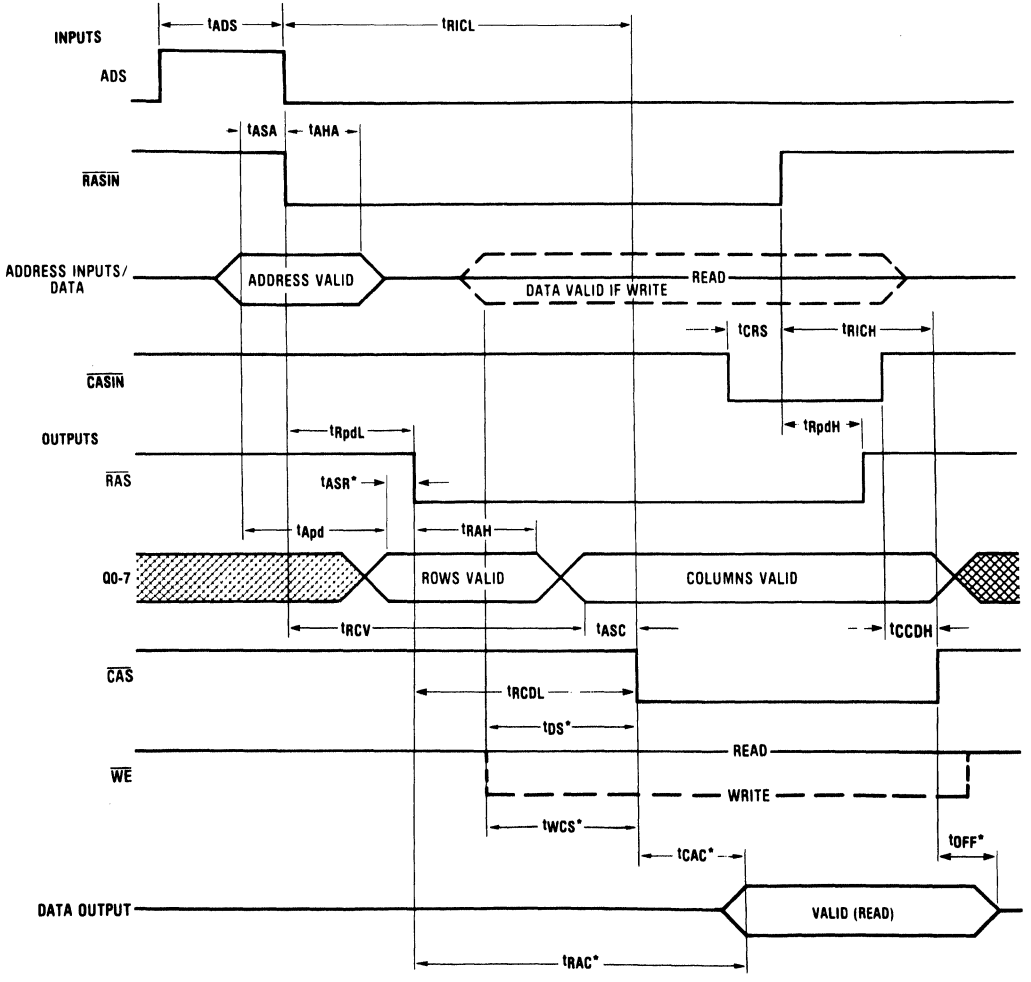
This gives a total typical delay from: input address valid to $\overline{RASIN}$ (15 ns); to $\overline{RAS}$ (27 ns); to rows held (50 ns); to columns valid (25 ns); to $\overline{CAS}$ (23 ns) = 140 ns (that is, 125 ns from $\overline{RASIN}$. All of these typical figures are for heavy capacitive loading, of approximately 88 DRAMs. This mode is therefore extremely fast. The external timing is greatly simplified for the memory system designer: the only system signal required is $\overline{RASIN}$.

MODE 6—FAST AUTOMATIC ACCESS

The Fast Access mode is similar to Mode 5, but has a faster t_{RAH} of 20 ns, minimum. It therefore can only be used with fast 16k or 64k DRAMs (which have a t_{RAH} of 10 ns to 15 ns) in applications requiring fast access times; $\overline{RASIN}$ to $\overline{CAS}$ is typically 105 ns.

In this mode, the R/$\overline{C}$ pin is not used, but $\overline{CASIN}$ is used to allow an extended $\overline{CAS}$ after $\overline{RAS}$ has already terminated. Refer to *Figure 5b*. This is desirable with fast cycle-times where $\overline{RAS}$ has to be terminated as soon as possible before the next $\overline{RAS}$ begins (to meet the precharge time, or t_{RP}, requirements of the DRAM). $\overline{CAS}$ may then be held low by $\overline{CASIN}$ to extend the data output valid time from the DRAM to allow the system to read the data. $\overline{CASIN}$ subsequently going high ends $\overline{CAS}$. If this extended $\overline{CAS}$ is not required, $\overline{CASIN}$ should be set high in Mode 6.

Timing Diagram



*Indicates Dynamic RAM Parameters

TL/F/8408–13

FIGURE 5b. Mode 6 Timing, Extended $\overline{CAS}$

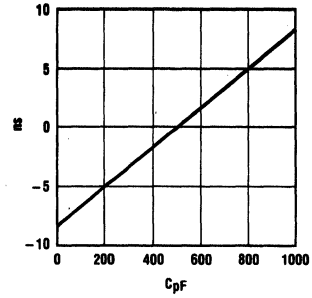
Functional Mode Descriptions (Continued)

MODE 7—SET END-OF-COUNT

The End-of-Count can be externally selected in Mode 7, using ADS to strobe in the respective value of B1 and B0 (see Table III). With B1 and B0 the same EOC is 127; with B1 = 0 and B0 = 1, EOC is 255; and with B1 = 1 and B0 = 0, EOC is 127. This selected value of EOC will be used until the next Mode 7 selection. At power-up the EOC is automatically set to 127 (B1 and B0 set to 11).

TABLE III. Mode 7

Bank Select (Strobed by ADS)		End of Count Selected
B1	B0	
0	0	127
0	1	255
1	0	127
1	1	127



TL/F/8408–14

FIGURE 6. Change in Propagation Delay vs. Loading Capacitance Relative to a 500 pF Load

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage, V_{CC}	7.0V
Storage Temperature Range	−65°C to +150°C
Input Voltage	5.5V
Output Current	150 mA
Lead Temperature (Soldering, 10 sec)	300°C

*Derate cavity package 23.6 mW/°C above 25°C; derate molded package 22.7 mW/°C above 25°C.

Maximum Power Dissipation* at 25°C	
Cavity Package	3542 mW
Molded Package	2833 mW

Operating Conditions

		Min	Max	Units
V_{CC}	Supply Voltage	4.75	5.25	V
T_A	Ambient Temperature	0	+70	°C

Electrical Characteristics V_{CC} = 5.0V ±5%, 0°C ≤ T_A ≤ 70°C (unless otherwise noted) (Notes 2, 6)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_C	Input Clamp Voltage	V_{CC} = Min., I_C = −12 mA		−0.8	−1.2	V
I_{IH1}	Input High Current for ADS, R/$\overline{C}$ only	V_{IN} = 2.5V		2.0	100	μA
I_{IH2}	Input High Current for All Other Inputs*	V_{IN} = 2.5V		1.0	50	μA
I_I RSI	Output Load Current for RF I/O	V_{IN} = 0.5V, Output High		−1.5	−2.5	mA
I_I CTL	Output Load Current for $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$	V_{IN} = 0.5V, Chip Deselect		−1.5	−2.5	mA
I_{IL1}	Input Low Current for ADS, R/$\overline{C}$ only	V_{IN} = 0.5V		−0.1	−1.0	mA
I_{IL2}	Input Low Current for All Other Inputs*	V_{IN} = 0.5V		−0.05	−0.5	mA
V_{IL}	Input Low Threshold				0.8	V
V_{IH}	Input High Threshold		2.0			V
V_{OL1}	Output Low Voltage*	I_{OL} = 20 mA		0.3	0.5	V
V_{OL2}	Output Low Voltage for RF I/O	I_{OL} = 10 mA		0.3	0.5	V
V_{OH1}	Output High Voltage*	I_{OH} = −1 mA	2.4	3.5		V
V_{OH2}	Output High Voltage for RF I/O	I_{OH} = −100 μA	2.4	3.5		V
I_{1D}	Output High Drive Current*	V_{OUT} = 0.8V (Note 3)		−200		mA
I_{0D}	Output Low Drive Current*	V_{OUT} = 2.7V (Note 3)		200		mA
I_{OZ}	TRI-STATE Output Current (Address Outputs)	0.4V ≤ V_{OUT} ≤ 2.7V, $\overline{CS}$ = 2.0V, Mode 4	−50	1.0	50	μA
I_{CC}	Supply Current	V_{CC} = Max.		210	285	mA

*Except RF I/O Output.

Switching Characteristic DP8408A/DP8408-3

V_{CC} = 5.0V ±5%, 0°C ≤ T_A ≤ 70°C unless otherwise noted (Notes 2, 4, 5). The output load capacitance is typical for 4 banks of 22 DRAMs each of 88 DRAMs including trace capacitance. These values are: Q0–Q7, C_L = 500 pF; $\overline{RAS}$0–$\overline{RAS}$3, C_L = 150 pF; $\overline{WE}$, C_L = 500 pF; $\overline{CAS}$, C_L = 600 pF, unless otherwise noted. See *Figure 7* for test load. Switches S1 and S2 are closed unless otherwise noted, and R1 and R2 are 4.7 kΩ unless otherwise noted. Maximum propagation delays are specified with all outputs switching.

Symbol	Access Parameter	Conditions	8408A			8408-3			Units
			Min	Typ	Max	Min	Typ	Max	
t_{RICL}	$\overline{RASIN}$ to $\overline{CAS}$ Output Delay (Mode 5)	*Figure 5a*	95	125	160	95	125	185	ns
t_{RICL}	$\overline{RASIN}$ to $\overline{CAS}$ Output Delay (Mode 6)	*Figures 5a, 5b*	80	105	140	80	105	160	ns
t_{RICH}	$\overline{RASIN}$ to $\overline{CAS}$ Output Delay (Mode 5)	*Figure 5a*	40	48	60	40	48	70	ns
t_{RICH}	$\overline{RASIN}$ to $\overline{CAS}$ Output Delay (Mode 6)	*Figures 5a, 5b*	50	63	80	50	63	95	ns
t_{RCDL}	$\overline{RAS}$ to $\overline{CAS}$ Output Delay (Mode 5)	*Figure 5a*		98	125		98	145	ns
t_{RCDL}	$\overline{RAS}$ to $\overline{CAS}$ Output Delay (Mode 6)	*Figures 5a, 5b*		78	105		78	120	ns
t_{RCDH}	$\overline{RAS}$ to $\overline{CAS}$ Output Delay (Mode 5)	*Figure 5a*		27	40		27	40	ns
t_{RCDH}	$\overline{RAS}$ to $\overline{CAS}$ Output Delay (Mode 6)	*Figure 5a*		40	65		40	65	ns
t_{CCDH}	$\overline{CASIN}$ to $\overline{CAS}$ Output Delay (Mode 6)	*Figure 5b*	40	54	70	40	54	80	ns
t_{RAH}	Row Address Hold Time (Mode 5)	*Figure 5a*	30			30			ns
t_{RAH}	Row Address Hold Time (Mode 6)	*Figures 5a, 5b*	20			20			ns
t_{ASC}	Column Address Setup Time (Mode 5)	*Figure 5a*	8			8			ns
t_{ASC}	Column Address Setup Time (Mode 6)	*Figures 5a, 5b*	6			6			ns
t_{RCV}	$\overline{RASIN}$ to Column Address Valid (Mode 5)	*Figure 5a*		90	120		90	140	ns
t_{RCV}	$\overline{RASIN}$ to Column Address Valid (Mode 6)	*Figures 5a, 5b*		75	105		75	120	ns
t_{RPDL}	$\overline{RASIN}$ to $\overline{RAS}$ Delay	*Figures 4a, 4b, 5a, 5b*	20	27	35	20	27	40	ns
t_{RPDH}	$\overline{RASIN}$ to $\overline{RAS}$ Delay	*Figures 4a, 4b, 5a, 5b*	15	23	32	15	23	37	ns
t_{APDL}	Address Input to Output Low Delay	*Figures 4a, 4b, 5a, 5b*		25	40		25	46	ns
t_{APDH}	Address Input to Output High Delay	*Figures 4a, 4b, 5a, 5b*		*25*	*40*		*25*	*46*	*ns*
t_{SPDL}	*Address Strobe to Address Output Low*	*Figures 4a, 4b,*		40	60		40	70	ns
t_{SPDH}	Address Strobe to Address Output High	*Figures 4a, 4b,*		40	60		40	70	ns
t_{ASA}	Address Setup Time to ADS	*Figures 4a, 4b, 5a, 5b*	15			15			ns
t_{AHA}	Address Hold Time from ADS	*Figures 4a, 4b, 5a, 5b*	15			15			ns
t_{ADS}	Address Strobe Pulse Width	*Figures 4a, 4b, 5a, 5b*	30			30			ns
t_{WPDL}	$\overline{WIN}$ to $\overline{WE}$ Output Delay	*Figure 4b*	15	25	30	15	25	35	ns
t_{WPDH}	$\overline{WIN}$ to $\overline{WE}$ Output Delay	*Figure 4b*	15	30	60	15	30	70	ns
t_{CRS}	$\overline{CASIN}$ Setup Time to $\overline{RASIN}$ High (Mode 6)	*Figure 5b*	35			35			ns
t_{CPDL}	$\overline{CASIN}$ to $\overline{CAS}$ Delay (R/$\overline{C}$ low in Mode 4)	*Figure 4b*	32	41	68	32	41	77	ns
t_{CPDH}	$\overline{CASIN}$ to $\overline{CAS}$ Delay	*Figure 4b*	25	39	50	25	39	60	ns
t_{RCC}	Column Select to Column Address Valid	Figure 4a		40	58		40	67	ns
t_{RCR}	Row Select to Row Address Valid	*Figures 4a, 4b*		40	58		40	67	ns
t_{RHA}	Row Address Held from Column Select	*Figure 4a*	10			10			ns
t_{CCAS}	R/$\overline{C}$ Low to $\overline{CAS}$ Low (Mode 4 Auto $\overline{CAS}$)	*Figure 7a*		65	90				ns

Switching Characteristics DP8408A/DP8408-3 (Continued)

V_{CC} = 5.0V ±5%, 0°C ≤ T_A ≤ 70°C unless otherwise noted (Notes 2, 4, 5). The output load capacitance is typical for 4 banks of 22 DRAMs each of 88 DRAMs including trace capacitance. These values are: Q0–Q7, C_L = 500 pF; $\overline{RAS}$0–$\overline{RAS}$3, C_L = 150 pF; $\overline{WE}$, C_L = 500 pF; $\overline{CAS}$, C_L = 600 pF, unless otherwise noted. See *Figure 7* for test load. Switches S1 and S2 are closed unless otherwise noted, and R1 and R2 are 4.7 kΩ unless otherwise noted. Maximum propagation delays are specified with all outputs switching.

Symbol	Access Parameter	Conditions	8408A			8408-3			Units
			Min	Typ	Max	Min	Typ	Max	
t_{DIF1}	Maximum ($t_{RPDL} - t_{RHA}$)	See Mode 4 description			13			18	ns
t_{DIF2}	Maximum ($t_{RCC} - t_{CPDL}$)	See Mode 4 description			13			18	ns
Refresh Parameter									
t_{RC}	Refresh Cycle Period	*Figure 2*	100			100			ns
$t_{RASINL, H}$	Pulse Width of $\overline{RASIN}$ during Refresh	*Figure 2*	50			50			ns
t_{RFPDL}	$\overline{RASIN}$ to $\overline{RAS}$ Delay during Refresh	*Figure 2*	35	50	70	35	50	80	ns
t_{RFPDH}	$\overline{RASIN}$ to $\overline{RAS}$ Delay during Refresh	*Figure 2*	30	40	55	30	40	65	ns
t_{RFLCT}	$\overline{RFSH}$ Low to Counter Address Valid	$\overline{CS}$ = X, *Figure 2*		47	60		47	70	ns
t_{RFHRV}	RFSH High to Row Address Valid	*Figure 2*		45	60		45	70	ns
t_{ROHNC}	$\overline{RAS}$ High to New Count Valid	*Figure 2*		30	55		30	55	ns
t_{RLEOC}	$\overline{RASIN}$ Low to End-of-Count Low	C_L = 50 pF, *Figure 2*			80			80	ns
t_{RHEOC}	$\overline{RASIN}$ High to End-of-Count High	C_L = 50 pF, *Figure 2*			80			80	ns
t_{RST}	Counter Reset Pulse Width	*Figure 2*	70			70			ns
t_{CTL}	RF I/O Low to Counter Outputs All Low	*Figure 2*			100			100	ns
TRI-STATE Parameter									
t_{ZH}	$\overline{CS}$ Low to Address Output High from Hi-Z	*Figure 8* R1 = 3.5k, R2 = 1.5k		35	60		35	60	ns
t_{HZ}	$\overline{CS}$ High to Address Output Hi-Z from High	C_L = 15 pF, *Figure 8* R2 = 1k, S1 open		20	40		20	40	ns
t_{ZL}	$\overline{CS}$ Low to Address Output Low from Hi-Z	*Figure 8* R1 = 3.5k, R2 = 1.5k		35	60		35	60	ns
t_{LZ}	$\overline{CS}$ High to Address Output Hi-Z from Low	C_L = 15 pF, *Figure 8* R1 = 1k, S2 open		25	50		25	50	ns
t_{HZH}	$\overline{CS}$ Low to Control Output High from Hi-Z High	*Figure 8* R2 = 750Ω, S1 open		50	80		50	80	ns
t_{HHZ}	$\overline{CS}$ High to Control Output Hi-Z High from High	C_L = 15 pF, *Figure 8* R2 = 750Ω, S1 open		40	75		40	75	ns
t_{HZL}	$\overline{CS}$ Low to Control Output Low from Hi-Z High	*Figure 8* S1, S2 open		45	75		45	75	ns
t_{LHZ}	$\overline{CS}$ High to Control Output Hi-Z High from Low	C_L = 15 pF, *Figure 8*, R2 = 750Ω, S1 open		50	80		50	80	ns

Switching Characteristics DP8408-2

V_{CC} = 5.0V ±5%, 0°C ≤ T_A ≤ 70°C unless otherwise noted (Notes 2, 4, 5, 7). The output load capacitance is typical for 4 banks of 22 DRAMs each or 88 DRAMS including trace capacitance. These values are: Q0–Q7, C_L = 500 pF; $\overline{RAS}$0–$\overline{RAS}$3, C_L = 150 pF, $\overline{WE}$, C_L = 500 pF; $\overline{CAS}$, C_L = 600 pF, unless otherwise noted. See *Figure 7* for test load. Switches S1 and S2 are closed unless otherwise noted, and R1 and R2 are 4.7 kΩ unless otherwise noted. Maximum propagation delays are specified with all outputs switching.

Symbol	Access Parameter	Conditions	8408-2			Units
			Min	Typ	Max	
t_{RICL}	$\overline{RASIN}$ to $\overline{CAS}$ Output Delay (Mode 5)	*Figure 5a*	75	100	130	ns
t_{RICL}	$\overline{RASIN}$ to $\overline{CAS}$ Output Delay (Mode 6)	*Figures 5a, 5b*	65	90	115	ns
t_{RICH}	$\overline{RASIN}$ to $\overline{CAS}$ Output Delay (Mode 5)	*Figure 5a*	40	48	60	ns
t_{RICH}	$\overline{RASIN}$ to $\overline{CAS}$ Output Delay (Mode 6)	*Figures 5a, 8b*	50	63	80	ns
t_{RCDL}	$\overline{RAS}$ to $\overline{CAS}$ Output Delay (Mode 5)	*Figure 5a*		75	100	ns
t_{RCDL}	$\overline{RAS}$ to $\overline{CAS}$ Output Delay (Mode 6)	*Figures 5a, 5b*		65	85	ns
t_{RCDH}	$\overline{RAS}$ to $\overline{CAS}$ Output Delay (Mode 5)	*Figure 5a*		27	40	ns
t_{RCDH}	$\overline{RAS}$ to $\overline{CAS}$ Output Delay (Mode 6)	*Figure 5a*		40	65	ns
t_{CCDH}	$\overline{CASIN}$ to $\overline{CAS}$ Output Delay (Mode 6)	*Figure 5b*	40	54	70	ns
t_{RAH}	Row Address Hold Time (Mode 5) (Note 7)	*Figure 5a*	20			ns
t_{RAH}	Row Address Hold Time (Mode 6) (Note 7)	*Figures 5a, 5b*	12			ns
t_{ASC}	Column Address Setup Time (Mode 5)	*Figure 5a*	3			ns
t_{ASC}	Column Address Setup Time (Mode 6)	*Figures 5a, 8b*	3			ns
t_{RCV}	$\overline{RASIN}$ to Column Address Valid (Mode 5)	*Figure 5a*		80	105	ns
t_{RCV}	$\overline{RASIN}$ to Column Address Valid (Mode 6)	*Figures 5a, 5b*		70	90	ns
t_{RPDL}	$\overline{RASIN}$ to $\overline{RAS}$ Delay	*Figures 4a, 4b, 5a, 5b*	20	27	35	ns
t_{RPDH}	$\overline{RASIN}$ to $\overline{RAS}$ Delay	*Figures 4a, 4b, 5a, 5b*	15	23	32	ns
t_{APDL}	Address Input to Output Low Delay	*Figures 4a, 4b, 5a, 5b*		25	40	ns
t_{APDH}	Address Input to Output High Delay	*Figures 4a, 4b, 5a, 5b*		25	40	ns
t_{SPDL}	Address Strobe to Address Output Low	*Figures 4a, 4b*		40	60	ns
t_{SPDH}	Address Strobe to Address Output High	*Figures 4a, 4b*		40	60	ns
t_{ASA}	Address Set-up Time to ADS	*Figures 4a, 4b, 5a, 5b*	15			ns
t_{AHA}	Address Hold Time from ADS	*Figures 4a, 4b, 5a, 5b*	15			ns
t_{ADS}	Address Strobe Pulse Width	*Figures 4a, 4b, 5a, 5b*	30			ns
t_{WPDL}	$\overline{WIN}$ to $\overline{WE}$ Output Delay	*Figure 4b*	15	25	30	ns
t_{WPDH}	$\overline{WIN}$ to $\overline{WE}$ Output Delay	*Figure 4b*	15	30	60	ns
t_{CRS}	$\overline{CASIN}$ Set-up Time to $\overline{RASIN}$ High (Mode 6)	*Figure 5b*	35			ns
t_{CPDL}	$\overline{CASIN}$ to $\overline{CAS}$ Delay (R/$\overline{C}$ low in Mode 4)	*Figure 4b*	32	41	58	ns
t_{CPDH}	$\overline{CASIN}$ to $\overline{CAS}$ Delay (R/$\overline{C}$ low in Mode 4)	*Figure 4b*	25	39	50	ns
t_{RCC}	Column Select to Column Address Valid	*Figure 4a*		40	58	ns
t_{RCR}	Row Select to Row Address Valid	*Figures 4a, 4b*		40	58	ns
t_{RHA}	Row Address Held from Column Select	*Figure 4a*	10			ns
t_{CCAS}	R/$\overline{C}$ Low to $\overline{CAS}$ Low (Mode 4 Auto $\overline{CAS}$)	*Figure 7a*		55	75	ns

Switching Characteristics DP8408-2 (Continued)

V_{CC} = 5.0V ±5%, 0°C ≤ T_A ≤ 70°C unless otherwise noted (Notes 2, 4, 5, 7). The output load capacitance is typical for 4 banks of 22 DRAMs each or 88 DRAMS including trace capacitance. These values are: Q0–Q7, C_L = 500 pF; $\overline{RAS}$0–$\overline{RAS}$3, C_L = 150 pF, $\overline{WE}$, C_L = 500 pF; $\overline{CAS}$, C_L = 600 pF, unless otherwise noted. See *Figure 7* for test load. Switches S1 and S2 are closed unless otherwise noted, and R1 and R2 are 4.7 kΩ unless otherwise noted. Maximum propagation delays are specified with all outputs switching.

Symbol	Access Parameter	Conditions	8408-2			Units
			Min	Typ	Max	
t_{DIF1}	Maximum ($t_{RPDL} - t_{RHA}$)	See Mode 4 description			13	ns
t_{DIF2}	Maximum ($t_{RCC} - t_{CPDL}$)	See Mode 4 description			13	ns
Refresh Parameter						
t_{RC}	Refresh Cycle Period	*Figure 2*	100			ns
$t_{RASINL, H}$	Pulse Width of $\overline{RASIN}$ during Refresh	*Figure 2*	50			ns
t_{RFPDL}	$\overline{RASIN}$ to $\overline{RAS}$ Delay during Refresh	*Figure 2*	35	50	70	ns
t_{RFPDH}	$\overline{RASIN}$ to $\overline{RAS}$ Delay during Refresh	*Figure 2*	30	40	55	ns
t_{RFLCT}	$\overline{RFSH}$ Low to Counter Address Valid	$\overline{CS}$ = X, *Figure 2*		47	60	ns
t_{RFHRV}	$\overline{RFSH}$ High to Row Address Valid	*Figure 2*		45	60	ns
t_{ROHNC}	$\overline{RAS}$ High to New Count Valid	*Figure 2*		30	55	ns
t_{RLEOC}	$\overline{RASIN}$ Low to End-of-Count Low	C_L = 50 pF, *Figure 2*			80	ns
t_{RHEOC}	$\overline{RASIN}$ High to End-of-Count High	C_L = 50 pF, *Figure 2*			80	ns
t_{RST}	Counter Reset Pulse Width	*Figure 2*	70			ns
t_{CTL}	RF I/O Low to Counter Outputs All Low	*Figure 2*			100	ns
TRI-STATE Parameter						
t_{ZH}	$\overline{CS}$ Low to Address Output High from Hi-Z	*Figures 9, 12* R1 = 3.5k, R2 = 1.5k		35	60	ns
t_{HZ}	$\overline{CS}$ High to Address Output Hi-Z from High	C_L = 15 pF, *Figures 9, 12* R2 = 1k, S1 open		20	40	ns
t_{ZL}	$\overline{CS}$ Low to Address Output Low from Hi-Z	*Figures 9, 12* R1 = 3.5k, R2 = 1.5k		35	60	ns
t_{LZ}	$\overline{CS}$ High to Address Output Hi-Z from Low	C_L = 15 pF, *Figures 9, 12* R1 = 1k, S2 open		25	50	ns
t_{HZH}	$\overline{CS}$ Low to Control Output High from Hi-Z High	*Figures 9, 12* R2 = 750Ω, S1 open		50	80	ns
t_{HHZ}	$\overline{CS}$ High to Control Output Hi-Z High from High	C_L = 15 pF, *Figures 9, 12* R2 = 750Ω, S1 open		40	75	ns
t_{HZL}	$\overline{CS}$ Low to Control Output Low from Hi-Z High	*Figure 12,* S1, S2 open		45	75	ns
t_{LHZ}	$\overline{CS}$ High to Control Output Hi-Z High from Low	C_L = 15 pF, *Figure 12,* R2 = 750Ω, S1 open		50	80	ns

Input Capacitance T_A = 25°C (Notes 2, 6)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
C_{IN}	Input Capacitance ADS, R/$\overline{C}$			8		pF
C_{IN}	Input Capacitance All Other Inputs			5		pF

Note 1: "Absolute Maximum Ratings" are the values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: All typical values are for T_A = 25°C and V_{CC} = 5.0V.

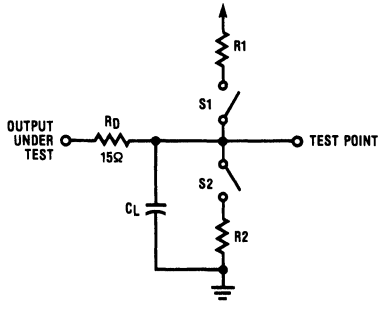
Note 3: This test is provided as a monitor of Driver output source and sink current capability. Caution should be exercised in testing this parameter. In testing these parameters, a 15Ω resistor should be placed in series with each output under test. One output should be tested at a time and test time should not exceed 1 second.

Note 4: Input pulse 0V to 3.0V, $t_R = t_F$ = 2.5 ns, f = 2.5 MHz, t_{PW} = 200 ns. Input reference point on AC measurements is 1.5V. Output reference points are 2.7V for High and 0.8V for Low.

Note 5: The load capacitance on RF I/O should not exceed 50 pF.

Note 6: Applies to all DP8408A versions unless otherwise specified.

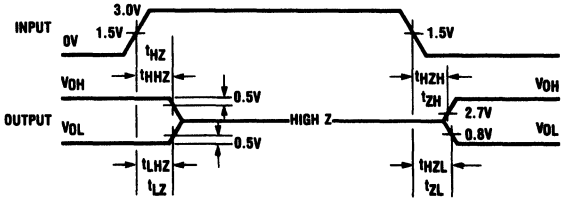
Note 7: The DP8408-2 device can only be used with memory devices that meet the t_{RAH} specification indicated.



TL/F/8408–15

FIGURE 7. Output Load Circuit

Timing Waveform



TL/F/8408–16

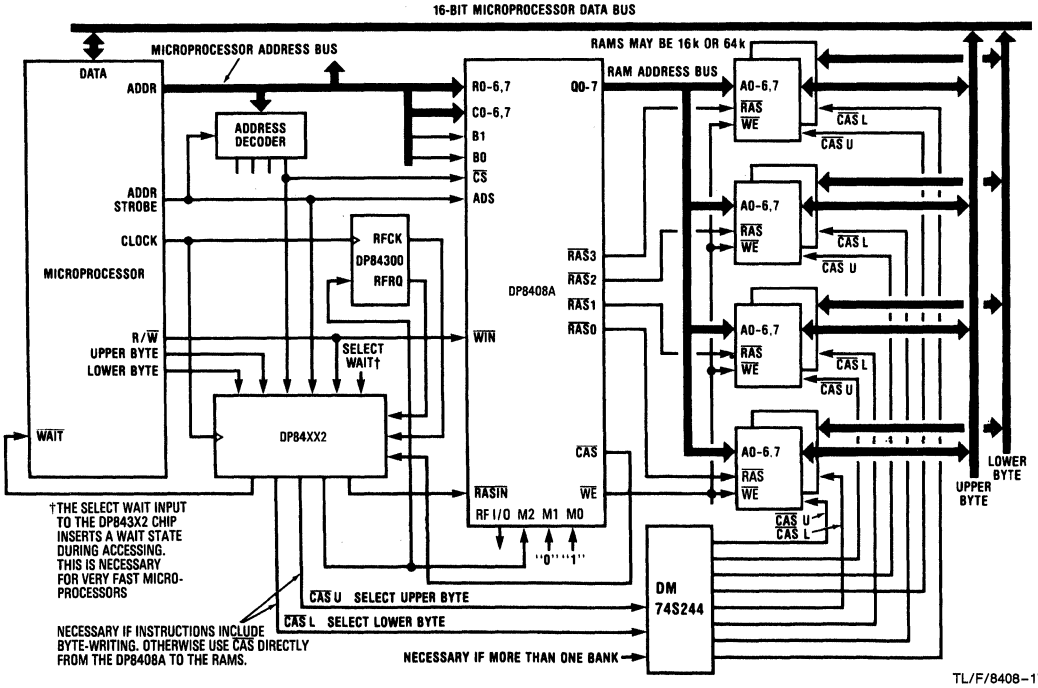
FIGURE 8

Applications

If external control is preferred, the DP8408A may be used in Modes 0 or 4, as in *Figure 3*.

If basic auto access and refresh are required, then in cases where the user requires the minimum of external complexity, Modes 0 and 5 are ideal, as shown in *Figure 9a*. The DP843X2 is used to provide proper arbitration between memory access and refresh. This chip supplies all the necessary control signals to the processor as well as the DP8408A. Furthermore, two separate $\overline{CAS}$ outputs are also included for systems using byte-writing. The refresh clock RFCK may be divided down from either RGCK using an IC counter such as the DM74LS393 or better still, the DP84300 Programmable Refresh Timer. The DP84300 can provide RFCK periods ranging from 15.4 μs to 15.6 μs based on the input clock of 2 to 10 MHz. *Figure 9b* shows the general timing diagram for interfacing the DP8408A to different microprocessors using the interface controller DP843X2.

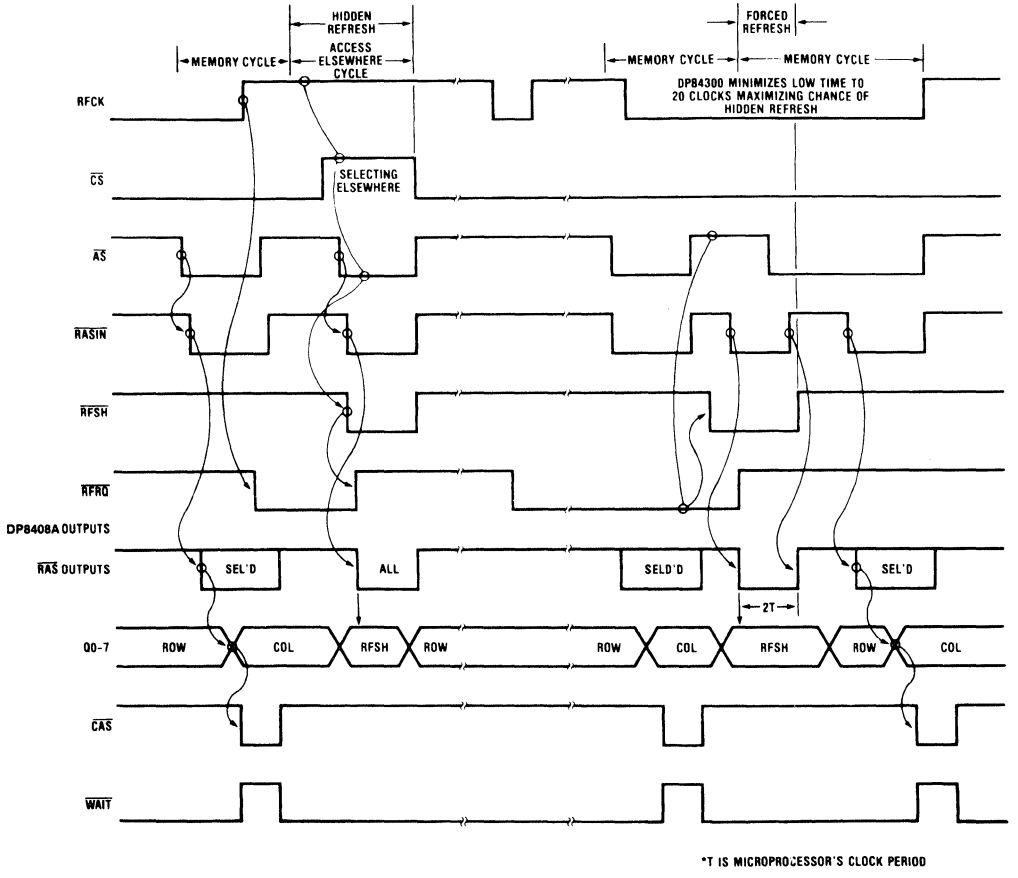
Applications (Continued)



TL/F/8408–17

FIGURE 9a. Connecting the DP8408A between the 16-Bit Microprocessor and Memory

Applications (Continued)



TL/F/8408–18

FIGURE 9b. DP8408A Auto Refresh

National Semiconductor

DP8409A Multi-Mode Dynamic RAM Controller/Driver

General Description

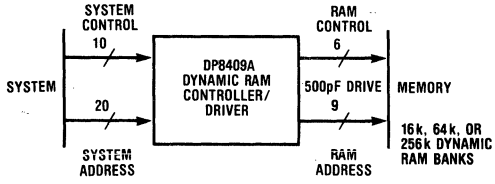
Dynamic memory system designs, which formerly required several support chips to drive the memory array, can now be implemented with a single IC ... the DP8409A Multi-Mode Dynamic RAM Controller/Driver. The DP8409A is capable of driving all 16k and 64k Dynamic RAMs (DRAMs) as well as 256k DRAMs. Since the DP8409A is a one-chip solution (including capacitive-load drivers), it minimizes propagation delay skews, the major performance disadvantage of multiple-chip memory drive and control.

The DP8409A's 8 modes of operation offer a wide selection of DRAM control capabilities. Memory access may be controlled externally or on-chip automatically; an on-chip refresh counter makes refreshing (either externally or automatically controlled) less complicated; and automatic memory initialization is both simple and fast.

The DP8409A is a 48-pin DRAM Controller/Driver with 9 multiplexed address outputs and 6 control signals. It consists of two 9-bit address latches, a 9-bit refresh counter, and control logic. All output drivers are capable of driving 500 pF loads with propagation delays of 25 ns. The DP8409A timing parameters are specified driving the typical load capacitance of 88 DRAMs, including trace capacitance.

The DP8409A has 3 mode-control pins: M2, M1, and M0, where M2 is in general $\overline{\text{REFRESH}}$. These 3 pins select 8 modes of operation. Inputs B1 and B0 in the memory access modes (M2 = 1), are select inputs which select one of four $\overline{\text{RAS}}$ outputs. During normal access, the 9 address outputs can be selected from the Row Address Latch or the Column Address Latch. During refresh, the 9-bit on-chip refresh counter is enabled onto the address bus and in this mode all $\overline{\text{RAS}}$ outputs are selected, while $\overline{\text{CAS}}$ is inhibited.

The DP8409A can drive up to 4 banks of DRAMs, with each bank comprised of 16k's, 64k's, or 256k's. Control signal outputs $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, and $\overline{\text{WE}}$ are provided with the same drive capability. Each $\overline{\text{RAS}}$ output drives one bank of DRAMs so that the four $\overline{\text{RAS}}$ outputs are used to select the banks, while $\overline{\text{CAS}}$, $\overline{\text{WE}}$, and the multiplexed addresses can be connected to all of the banks of DRAMs. This leaves the non-selected banks in the standby mode (less than one tenth of the operating power) with the data outputs in TRI-STATE®. Only the bank with its associated $\overline{\text{RAS}}$ low will be written to or read from.



TL/F/8409–1

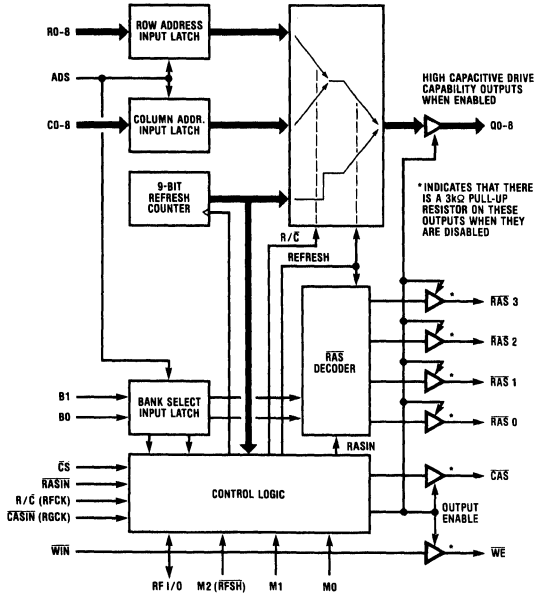
Operational Features

- All DRAM drive functions on one chip—minimizes skew on outputs, maximizes AC peformance
- On-chip capacitive-load drives (specified to drive up to 88 DRAMs)
- Drives directly all 16k, 64k, and 256k DRAMs
- Capable of addressing 64k, 256k, or 1M words
- Propagation delays of 25 ns typical at 500 pF load
- $\overline{\text{CAS}}$ goes low automatically after column addresses are valid if desired
- Auto Access mode provides $\overline{\text{RAS}}$, row to column select, then $\overline{\text{CAS}}$ automatically and fast
- $\overline{\text{WE}}$ follows $\overline{\text{WIN}}$ unconditionally—offering READ, WRITE or READ-MODIFY-WRITE cycles
- On-chip 9-bit refresh counter with selectable End-of-Count (127, 255 or 511)
- End-of-Count indicated by RF I/O pin going low at 127, 255 or 511
- Low input on RF I/O resets 9-bit refresh counter
- $\overline{\text{CAS}}$ inhibited during refresh cycle
- Fall-through latches on address inputs controlled by ADS
- TRI-STATE outputs allow multi-controller addressing of memory
- Control output signals go high-impedance logic "1" when disabled for memory sharing
- Power-up: counter reset, control signals high, address outputs TRI-STATE, and End-of-Count set to 127

Mode Features

- 8 modes of operation: 3 access, 3 refresh, and 2 set-up
- 2 externally controlled modes: 1 access and 1 refresh (Modes 0, 4)
- 2 auto-access modes $\overline{\text{RAS}} \rightarrow \text{R}/\overline{\text{C}} \rightarrow \overline{\text{CAS}}$ automatic, with t_{RAH} = 20 or 30 ns minimum (Modes 5, 6)
- Auto-access mode allows Hidden Refreshing (Mode 5)
- Forced Refresh requested on RF I/O if no Hidden Refresh (Mode 5)
- Forced Refresh performed after system acknowledge of request (Mode 1)
- Automatic Burst Refresh mode stops at End-of-Count of 127, 255, or 511 (Mode 2)
- 2 All-$\overline{\text{RAS}}$ Acces modes externally or automatically controlled for memory initialization (Modes 3a, 3b)
- Automatic All-$\overline{\text{RAS}}$ mode with external 8-bit counter frees system for other set-up routines (Mode 3a)
- End-of-Count value of Refresh Counter set by B1 and B0 (Mode 7)

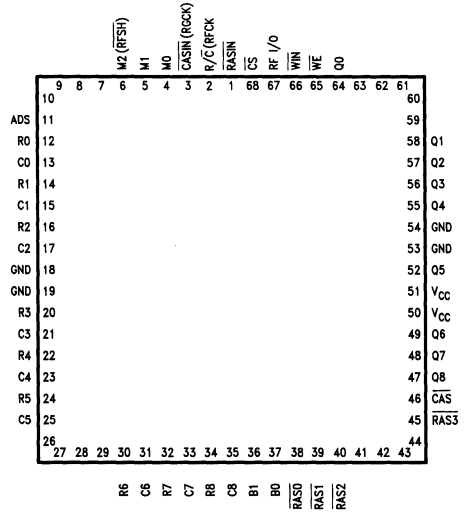
Block and Connection Diagrams



TL/F/8409-2

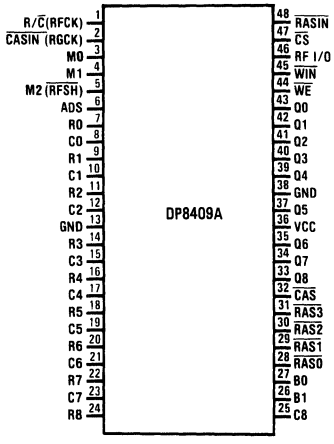
Order Number DP8409AD, DP8409AN, DP8409AN-3 or DP8409AV-2
See NS Package Number D48A, N48A or V68A

68 Pin PCC



TL/F/8409-3

Dual-In-Line Package



TL/F/8409-5

Top View

Pin Definitions

V_{CC}, GND, GND—V_{CC} = 5V ±5%. The three supply pins have been assigned to the center of the package to reduce voltage drops, both DC and AC. There are also two ground pins to reduce the low level noise. The second ground pin is located two pins from V_{CC}, so that decoupling capacitors can be inserted directly next to these pins. It is important to adequately decouple this device, due to the high switching currents that will occur when all 9 address bits change in the same direction simultaneously. A recommended solution would be a 1 μF multilayer ceramic capacitor in parallel with a low-voltage tantalum capacitor, both connected as close as possible to pins 36 and 38 to reduce lead inductance. See figure below.

TL/F/8409-4

*Capacitor values should be chosen depending on the particular application.

R0-R8: Row Address Inputs.

C0-C8: Column Address Inputs.

Q0-Q8: Multiplexed Address Outputs—Selected from the Row Address Input Latch, the Column Address Input Latch, or the Refresh Counter.*

$\overline{RASIN}$: Row Address Strobe Input—Enables selected $\overline{RAS}_n$ output when M2 ($\overline{RFSH}$) is high, or all $\overline{RAS}_n$ outputs when $\overline{RFSH}$ is low.

R/$\overline{C}$ (RFCK)—In Auto-Refresh Mode this pin is the external Refresh Clock Input: one refresh cycle has to be performed each clock period. In all other modes it is Row/Column Select Input: selects either the row or column address input latch onto the output bus.

Pin Definitions (Continued)

TABLE I. DP8409A Mode Select Options

Mode	(RFSH) M2	M1	M0	Mode of Operation	Conditions
0	0	0	0	Externally Controlled Refresh	RF I/O = EOC
1	0	0	1	Auto Refresh—Forced	RF I/O = Refresh Request (RFRQ)
2	0	1	0	Internal Auto Burst Refresh	RF I/O = EOC
3a	0	1	1	All RAS Auto Write	RF I/O = EOC; All RAS Active
3b	0	1	1	Externally Controlled All RAS Access	All RAS Active
4	1	0	0	Externally Controlled Access	Active RAS Defined by Table II
5	1	0	1	Auto Access, Slow t_{RAH}, Hidden Refresh	Active RAS Defined by Table II
6	1	1	0	Auto Access, Fast t_{RAH}	Active RAS Defined by Table II
7	1	1	1	Set End of Count	See Table III for Mode 7

CASIN (RGCK)—In Auto-Refresh Mode, Auto Burst Mode, and All-RAS Auto-Write Mode, this pin is the RAS Generator Clock input. In all other modes it is CASIN (Column Address Strobe Input), which inhibits CAS output when high in Modes 4 and 3b. In Mode 6 it can be used to prolong CAS output.

ADS: Address (Latch) Strobe Input—Row Address, Column Address, and Bank Select Latches are fall-through with ADS high; Latches on high-to-low transition.

CS: Chip Select Input—The TRI-STATE mode will Address Outputs and puts the control signal into a high-impedance logic "1" state when high (unless refreshing in one of the Refresh Modes). Enables all outputs when low.

M0, M1, M2: Mode Control Inputs—These 3 control pins determine the 8 major modes of operation of the DP8409A as depicted in Table I.

RF I/O—The I/O pin functions as a Reset Counter Input when set low from an external open-collector gate, or as a flag output. The flag goes active-low in Modes 0 and 2 when the End-of-Count output is at 127, 255, or 511 (see Table III). In Auto-Refresh Mode it is the Refresh Request output.

WIN: Write Enable Input.

WE: Write Enable Output—Buffered output from WIN.*

CAS: Column Address Strobe Output—In Modes 3a, 5, and 6, CAS transitions low following valid column address. In Modes 3b and 4, it goes low after R/C goes low, or follows CASIN going low if R/C is already low. CAS is high duing refresh.*

RAS 0–3: Row Address Strobe Outputs—Selects a memory bank decoded from B1 and B0 (see Table II), if RFSH is high. If RFSH is low, all banks are selected.*

B0, B1: Bank Select Inputs—Strobed by ADS. Decoded to enable one of the RAS outputs when RASIN goes low. Also used to define End-of-Count in Mode 7 (Table III).

Conditions for All Modes

INPUT ADDRESSING

The address block consists of a row-address latch, a column-address latch, and a resettable refresh counter. The address latches are fall-through when ADS is high and latch when ADS goes low. If the address bus contains valid addresses until after the valid address time, ADS can be permanently high. Otherwise ADS must go low while the addresses are still valid.

In normal memory access operation, RASIN and R/C are initially high. When the address inputs are enabled into the address latches, the row addresses appear on the Q outputs. The address strobe also inputs the bank-select address, (B0 and B1). If CS is low, all outputs are enabled. When CS is transitioned high, the address outputs go TRI-STATE and the control outputs first go high through a low impedance, and then are held by an on-chip high impedance. This allows output paralleling with other DP8409As for multi-addressing. All outputs go active about 50 ns after the chip is selected again. If CS is high, and a refresh cycle begins, all the outputs become active until the end of the refresh cycle.

DRIVE CAPABILITY

The DP8409A has timing parameters that are specified with up to 600 pF loads. In a typical memory system this is equivalent to about 88, 5V-only DRAMs, with trace lengths kept to a minimum. Therefore, the chip can drive four banks each of 16 or 22 bits, or two banks of 32 or 39 bits, or one bank of 64 or 72 bits.

Less loading will slightly reduce the timing parameters, and more loading will increase the timing parameters, according to the graph of *Figure 10*. The AC performance parameters are specified with the typical load capacitance of 88 DRAMs. This graph can be used to extrapolate the variations expected with other loading.

Because of distributed trace capacitance and inductance and DRAM input capacitance, current spikes can be created, causing overshoots and undershoots at the DRAM inputs that can change the contents of the DRAMs or even destroy them. To remove these spikes, a damping resistor (low inductance, carbon) can be inserted between the DP8409A driver outputs and the DRAMs, as close as possible to the DP8409A. The values of the damping resistors may differ between the different control outputs; RASs, CAS, Q's, and WE. The damping resistors should be determined by the first prototypes (not wire-wrapped due to the larger distributed capacitance and inductance). The best values for the damping resistors are the critical values giving a critically damped transition on the control outputs. Typical values for the damping resistors will be between 15Ω and 100Ω, the lower the loading the higher the value. (For more information, see AN-305 "Precautions to Take When Driving Memories.")

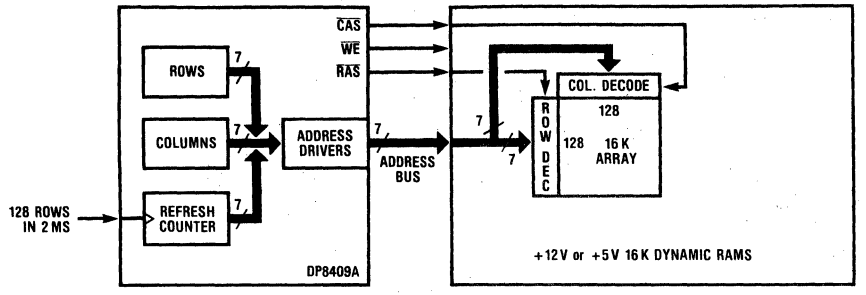
Conditions for All Modes (Continued)

DP8409A DRIVING ANY 16k OR 64k DRAMs

The DP8409A can drive any 16k or 64k DRAMs. All 16k DRAMs are basically the same configuration, including the newer 5V-only version. Hence, in most applications, different manufacturers' DRAMs are interchangeable (for the same supply-rail chips), and the DP8409A can drive all 16k DRAMs (see *Figure 1a*).

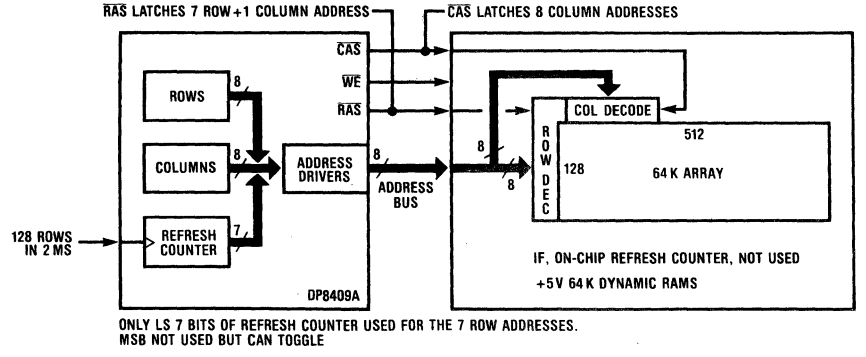
There are three basic configurations for the 5V-only 64k DRAMs: a 128-row by 512-column array with an on-RAM refresh counter, a 128-row by 512-column array with no on-RAM refresh counter, and a 256-row by 256-column array with no on-RAM refresh counter. The DP8409A can drive all three configurations, and at the same time allows them all to be interchangeable (as shown in *Figures 1b* and *1c*), providing maximum flexibility in the choice of DRAMs. Since the 9-bit on-chip refresh counter can be used as a 7-bit refresh counter for the 128-row configuration, or as an 8-bit refresh counter for the 256-row configuration, the on-RAM refresh counter (if present) is never used. As long as 128 rows are refreshed every 2 ms (i.e. 256 rows in 4 ms) all DRAM types are correctly refreshed.

DP8409A Interface between System and DRAM Banks



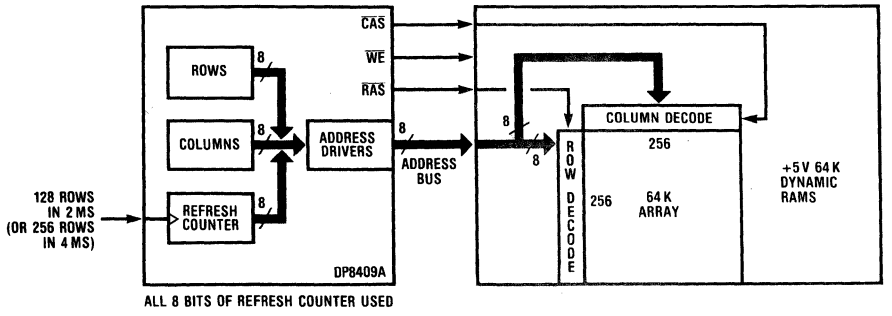
TL/F/8409–6

FIGURE 1a. DP8409A with any 16k DRAMs



TL/F/8409–7

FIGURE 1b. DP8409A with 128 Row x 512 Column 64k DRAM



TL/F/8409–8

FIGURE 1c. DP8409A with 256 x 256 Column 64k DRAM

Conditions for All Modes (Continued)

When the DP8409A is in a refresh mode, the RF I/O pin indicates that the on-chip refresh counter has reached its end-of-count. This end-of-count is selectable as 127, 255 or 512 to accommodate 16k, 64k or 256k DRAMs. Although the end-of-count may be chosen to be any of these, the counter always counts to 511 before rolling over to zero.

READ, WRITE, AND READ-MODIFY-WRITE CYCLES

The output signal, $\overline{WE}$, determines what type of memory access cycle the memory will perform. If $\overline{WE}$ is kept high while $\overline{CAS}$ goes low, a read cycle occurs. If $\overline{WE}$ goes low before $\overline{CAS}$ goes low, a write cycle occurs and data at DI (DRAM input data) is written into the DRAM as $\overline{CAS}$ goes low. If $\overline{WE}$ goes low later than t_{CWD} after $\overline{CAS}$ goes low, first a read occurs and DO (DRAM output data) becomes valid; then data DI is written into the same address in the DRAM when $\overline{WE}$ goes low. In this read-modify-write case, DI and DO cannot be linked together. The type of cycle is therefore controlled by $\overline{WE}$, which follows $\overline{WIN}$.

POWER-UP INITIALIZE

When V_{CC} is first applied to the DP8409A, an initialize pulse clears the refresh counter, the internal control flip-flops, and set the End-of-Count of the refresh counter to 127 (which may be changed via Mode 7). As V_{CC} increases to about 2.3V, it holds the output control signals at a level of one Schottky diode-drop below V_{CC}, and the output address to TRI-STATE. As V_{CC} increases above 2.3V, control of these outputs is granted to the system.

DP8409A Functional Mode Descriptions

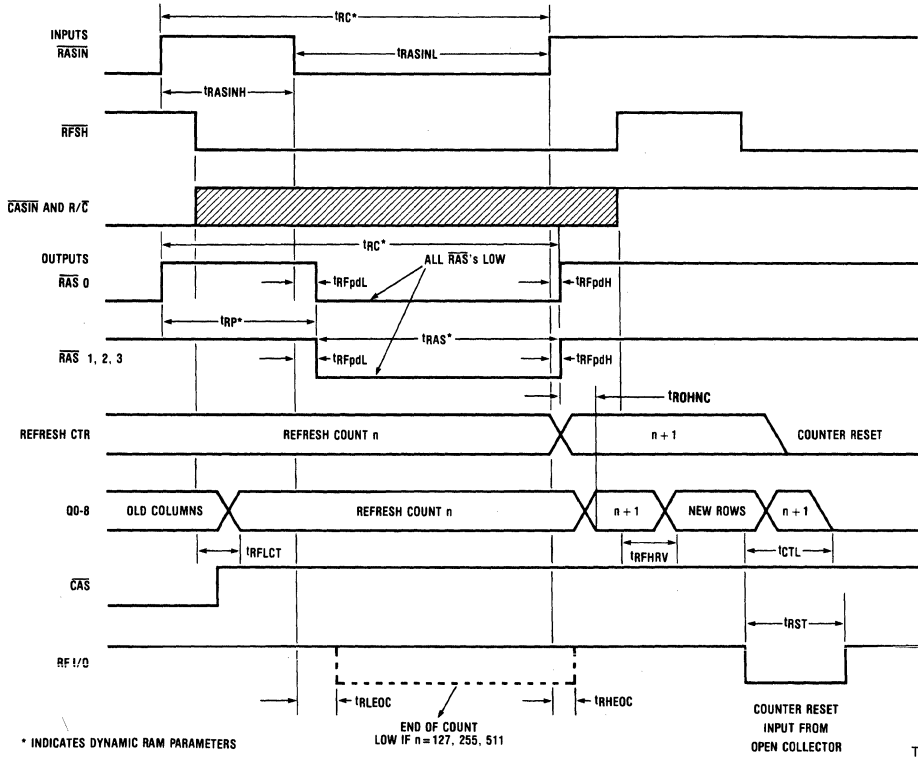
Note: All delay parameters stated in text refer to the DP8409A. Substitute the respective delay numbers for the DP8409-2 or DP8409-3 when using these devices.

MODE 0—EXTERNALLY CONTROLLED REFRESH

Figure 2 is the Externally Controlled Refresh Timing. In this mode, the input address latches are disabled from the address outputs and the refresh counter is enabled. When $\overline{RAS}$ occurs, the enabled row in the DRAM is refreshed. In the Externally Controlled Refresh mode, all $\overline{RAS}$ outputs are enabled following $\overline{RASIN}$, and $\overline{CAS}$ is inhibited. This refreshes the same row in all four banks. The refresh counter increments when either $\overline{RASIN}$ or $\overline{RFSH}$ goes low-to-high after a refresh. RF I/O goes low when the count is 127, 255, or 511, as set by End-of-Count (see Table III), with $\overline{RASIN}$ and $\overline{RFSH}$ low. To reset the counter to all zeros, RF I/O is set low through an external open-collector driver.

During refresh, $\overline{RASIN}$ and RFSH must be skewed transitioning low such that the refresh address is valid on the address outputs of the controller before the $\overline{RAS}$ outputs go low. The amount of time that RFSH should go low before $\overline{RASIN}$ does depends on the capacitive loading of the address and $\overline{RAS}$ lines. For the load specified in the switching characteristics of this data sheet, 10 ns is sufficient. Refer to *Figure 2*.

To perform externally controlled burst refresh, $\overline{RASIN}$ is toggled while RFSH is held low. The refresh counter increments with $\overline{RASIN}$ going low to high, so that the DRAM rows are refreshed in succession by $\overline{RASIN}$ going high to low.



TL/F/8409–9

FIGURE 2. External Control Refresh Cycle (Mode 0)

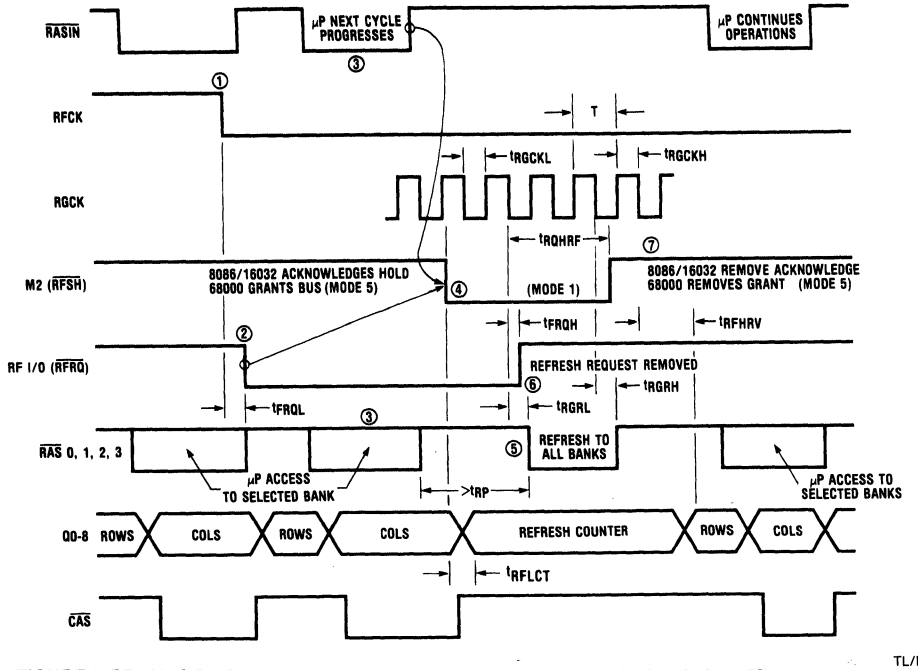
DP8409A Functional Mode Descriptions (Continued)

MODE 1—AUTOMATIC FORCED REFRESH

In Mode 1, the R/$\overline{C}$ (RFCK) pin becomes RFCK (refresh cycle clock), instead of R/$\overline{C}$, and $\overline{CAS}$ remains high. If RFCK is kept permanently high, then whenever M2 ($\overline{RFSH}$) goes low, an externally controlled refresh will occur and all $\overline{RAS}$ outputs will follow $\overline{RASIN}$, strobing the refresh counter contents to the DRAMs. The RF I/O pin will always output high, but when set low externally through an open-collector driver, the refresh counter resets as normal. This externally controlled method may be preferred when operating in the Automatic Access mode (Mode 5), where hidden or forced refreshing is undesirable, but refreshing is still necessary.

If RFCK is an input clock signal, one (and only one) refresh cycle must take place every RFCK cycle. Refer to *Figure 9*. If a hidden refresh does not occur while RFCK is high, in Mode 5, then RF I/O ($\overline{Refresh\ Request}$) goes low immediately after RFCK goes low, indicating to the system that a forced refresh is requested. The system must allow a forced refresh to take place while RFCK is low (refer to *Figure 3*). The $\overline{Refresh\ Request}$ signal on RF I/O may be connected to a $\overline{Hold}$ or $\overline{Bus\ Request}$ input to the system. The system acknowledges the $\overline{Hold}$ or $\overline{Bus\ Request}$ when ready, and outputs Hold Acknowledge or $\overline{Bus\ Request\ Acknowledge}$. If this is connected to the M2 ($\overline{RFSH}$) pin, a forced-refresh cycle will be initiated by the DP8409A, and $\overline{RAS}$ will be internally generated on all four $\overline{RAS}$ outputs, to strobe the refresh counter contents on the address outputs into all the DRAMs. An external $\overline{RAS}$ Generator Clock (RGCK) is required for this function. It is fed to the $\overline{CASIN}$ (RGCK) pin, and may be up to 10 MHz. Whenever M2 goes low (inducing a forced refresh), $\overline{RAS}$ remains high for one to two periods of RGCK, depending on when M2 goes low relative to the high-to-low triggering edge of RGCK; $\overline{RAS}$ then goes low for two periods, performing a refresh on all banks. In order to obtain the minimum delay from M2 going low to $\overline{RAS}$ going low, M2 should go low $t_{\overline{RFSRG}}$ before the next falling edge of RGCK. The $\overline{Refresh\ Request}$ on RF I/O is terminated as $\overline{RAS}$ begins, so that by the time the system has acknowledged the removal of the request and disabled its Acknowledge, (i.e., M2 goes high), Refresh $\overline{RAS}$ will have ended, and normal operations can begin again in the Automatic Access mode (Mode 5). If it is desired that Refresh $\overline{RAS}$ end in less than 2 periods of RGCK from the time $\overline{RAS}$ went low, then M2 may be high earlier than t_{RQHRF} after RGCK goes low and $\overline{RAS}$ will go high t_{RFRH} after M2, if $\overline{CS}$ is low. If $\overline{CS}$ is high, the $\overline{RAS}$ will go high after 25 ns after M2 goes high.

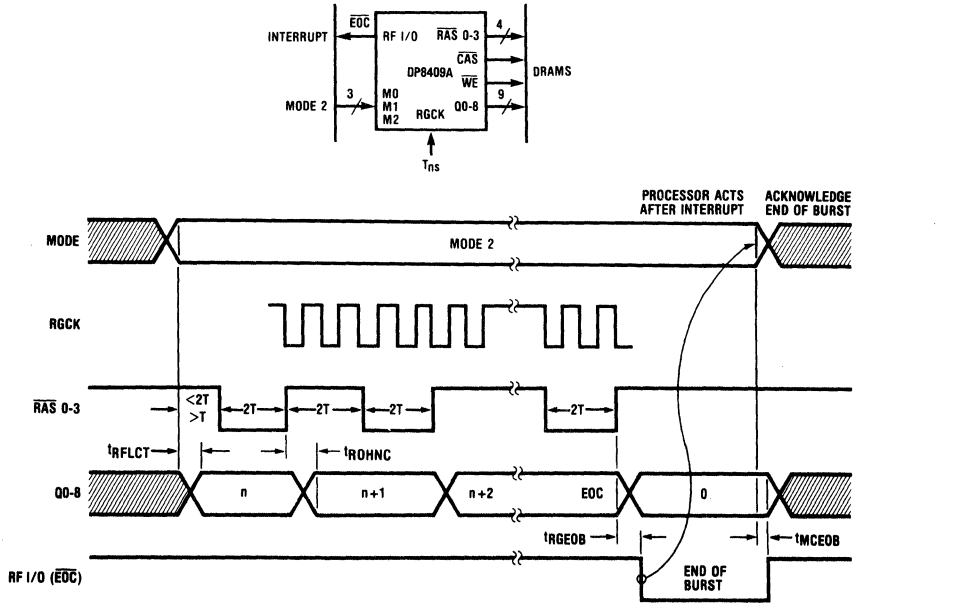
To allow the forced refresh, the system will have been inactive for about 4 periods of RGCK, which can be as fast as 400 ns every RFCK cycle. To guarantee a refresh of 128 rows every 2 ms, a period of up to 16 μs is required for RFCK. In other words, the system may be down for as little as 400 ns every 16 μs, or 2.5% of the time. Although this is not excessive, it may be preferable to perform a Hidden Refresh each RFCK cycle, which is allowed while still in the Auto-Access mode, (Mode 5).



TL/F/8409–10

FIGURE 3. DP8409A Performing a Forced Refresh (Mode 5 → 1 → 5) with Various Microprocessors

DP8409A Functional Mode Descriptions (Continued)



TL/F/8409–11

FIGURE 4. Auto-Burst Mode, Mode 2

MODE 2—AUTOMATIC BURST REFRESH

This mode is normally used before and/or after a DMA operation to ensure that all rows remain refreshed, provided the DMA transfer takes less than 2 ms (see *Figure 4*). When the DP8409A enters this mode, $\overline{\text{CASIN}}$ (RGCK) becomes the $\overline{\text{RAS}}$ Generator Clock (RGCK), and $\overline{\text{RASIN}}$ is disabled. $\overline{\text{CAS}}$ remains high, and RF I/O goes low when the refresh counter has reached the selected End-of-Count and the last $\overline{\text{RAS}}$ has ended. RF I/O then remains low until the Auto-Burst Refresh mode is terminated. RF I/O can therefore be used as an interrupt to indicate the End-of-Burst conditions.

The signal on all four $\overline{\text{RAS}}$ outputs is just a divide-by-four of RGCK; in other words, if RGCK has a 100 ns period, $\overline{\text{RAS}}$ is high and low for 200 ns each cycle. The refresh counter increments at the end of each $\overline{\text{RAS}}$, starting from the count it contained when the mode was entered. If this was zero, then for a RGCK with a 100 ns period with End-of-Count set to 127, RF I/O will go low after 128 $\times$ 0.4 μs, or 51.2 μs. During this time, the system may be performing operations that do not involve DRAM. If all rows need to be burst refreshed, the refresh counter may be cleared by setting RF I/O low externally before the burst begins.

Burst-mode refreshing is also useful when powering down systems for long periods of time, but with data retention still required while the DRAMs are in standby. To maintain valid refreshing, power can be applied to the DP8409A (set to Mode 2), causing it to perform a complete burst refresh. When end-of-burst occurs (after 26 μs), power can then be removed from the DP8409A for 2 ms, consuming an average power of 1.3% of normal operating power. No control signal glitches occur when switching power to the DP8409A.

MODE 3a—ALL-$\overline{\text{RAS}}$ AUTOMATIC WRITE

Mode 3a is useful at system initialization, when the memory is being cleared (i.e., with all-zeros in the data field and the corresponding check bits for error detection and correction). This requires writing the same data to each location of memory (every row of each column of each bank). All $\overline{\text{RAS}}$ outputs are activated, as in refresh, and so are $\overline{\text{CAS}}$ and $\overline{\text{WE}}$. To write to all four banks simultaneously, every row is strobed in each column, in sequence, until data has been written to all locations.

To select this mode, B1 and B0 must have previously been set to 00, 01, or 10 in Mode 7, depending on the DRAM size. For example, for 16k DRAMs, B1 and B0 are 00. For 64k DRAMs, B1 and B0 are 01, so that for the configuration of *Figure 1b*, the 8 refresh counter bits are strobed by $\overline{\text{RAS}}$ into the 7 row addresses and the ninth column address. After this Automatic-Write process, B1 and B0 must be set again in Mode 7 to 00 to set End-of-Count to 127. For the configuration of *Figure 1c*, B1 and B0 set to 01 will work for Automatic-Write and End-of-Count equals 255.

In this mode, R/$\overline{\text{C}}$ is disabled, $\overline{\text{WE}}$ is permanently enabled low, and $\overline{\text{CASIN}}$ (RGCK) becomes RGCK. RF I/O goes low whenever the refresh counter is 127, 255, or 511 (as set by End-of-Count in Mode 7), and the $\overline{\text{RAS}}$ outputs are active.

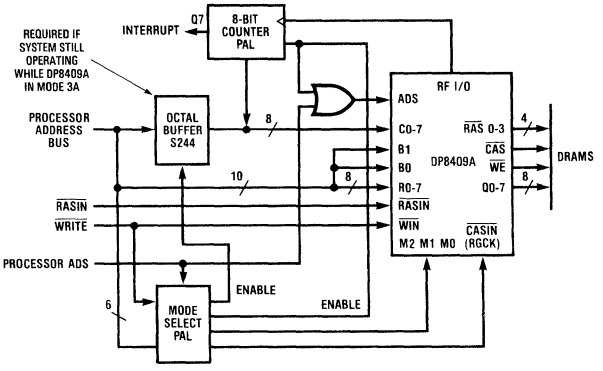
Referring to *Figure 5a*, an external 8-bit counter (for 64k DRAMs) with TRI-STATE outputs is required and must be connected to the column address inputs. It is enabled only during this mode, and is clocked from RF I/O. The DP8409A refresh counter is used to address the rows, and the column address is supplied by the external counter. Every row for each column address is written to in all four banks. At the End-of-Count RF I/O goes low, which clocks the external counter.

Therefore, for each column address, the refresh counter first outputs row-0 to the address bus and all four $\overline{\text{RAS}}$ outputs strobe this row address into the DRAMs (see *Figure 5b*). A minimum of 30 ns after $\overline{\text{RAS}}$ goes low (t_{RAH} = 30 ns), the refresh counter is disabled and the column ad-

DP8409A Functional Mode Descriptions (Continued)

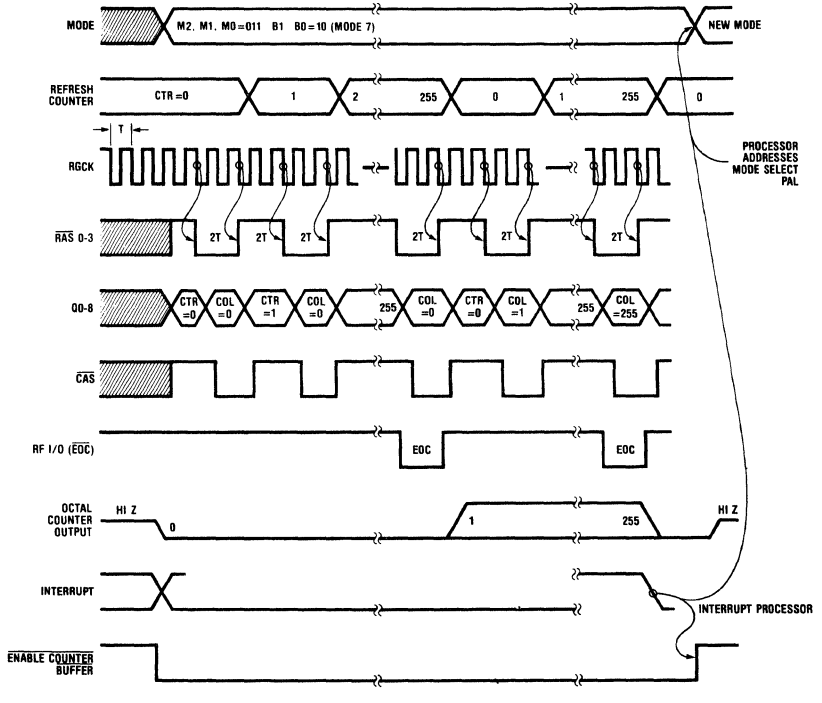
dress input latch is enabled onto the address bus. About 14 ns after the column address is valid, $\overline{CAS}$ goes low, (t_{ASC} = +14 ns), strobing the column address into the DRAMs. When $\overline{RAS}$ and $\overline{CAS}$ go high the refresh counter increments to the next row and the cycle repeats. Since $\overline{WE}$ is kept low in this mode, the data at DI (input data) of the DRAMs is written into each row of the latched column. During each cycle $\overline{RAS}$ is high for two periods of RGCK and low for two periods, giving a total write-cycle time of 400 ns minimum, which is adequate for most 16k and 64k DRAMs. On the last row of a column, RF I/O increments the external counter to the next column address.

At the end of the last column address, an interrupt is generated from the external counter to let the system know that initialization has been completed. During the entire initialization time, the system can be performing other initialization functions. This approach to memory initialization is both automatic and fast. For instance, if four banks of 64k DRAMs are used, and RGCK is 100 ns, a write cycle to the same location in all four banks takes 400 ns, so the total time taken in initializing the 64k DRAMs is 65k × 400 ns or 26 ms. When the system receives the interrupt, the external counter must be permanently disabled. ADS and $\overline{CS}$ are interfaced by the system, and the DP8409A mode is changed. The interrupt must then be disabled.



TL/F/8409–12

FIGURE 5a. DP8409A Extra Circuitry Required for All-$\overline{RAS}$ Auto Write Mode, Mode 3a



TL/F/8409–13

FIGURE 5b. DP8409A All-$\overline{RAS}$ Auto Write Mode, Mode 3a, Timing Waveform

DP8409A Functional Mode Descriptions (Continued)

MODE 3b—EXTERNALLY CONTROLLED ALL-$\overline{RAS}$ WRITE

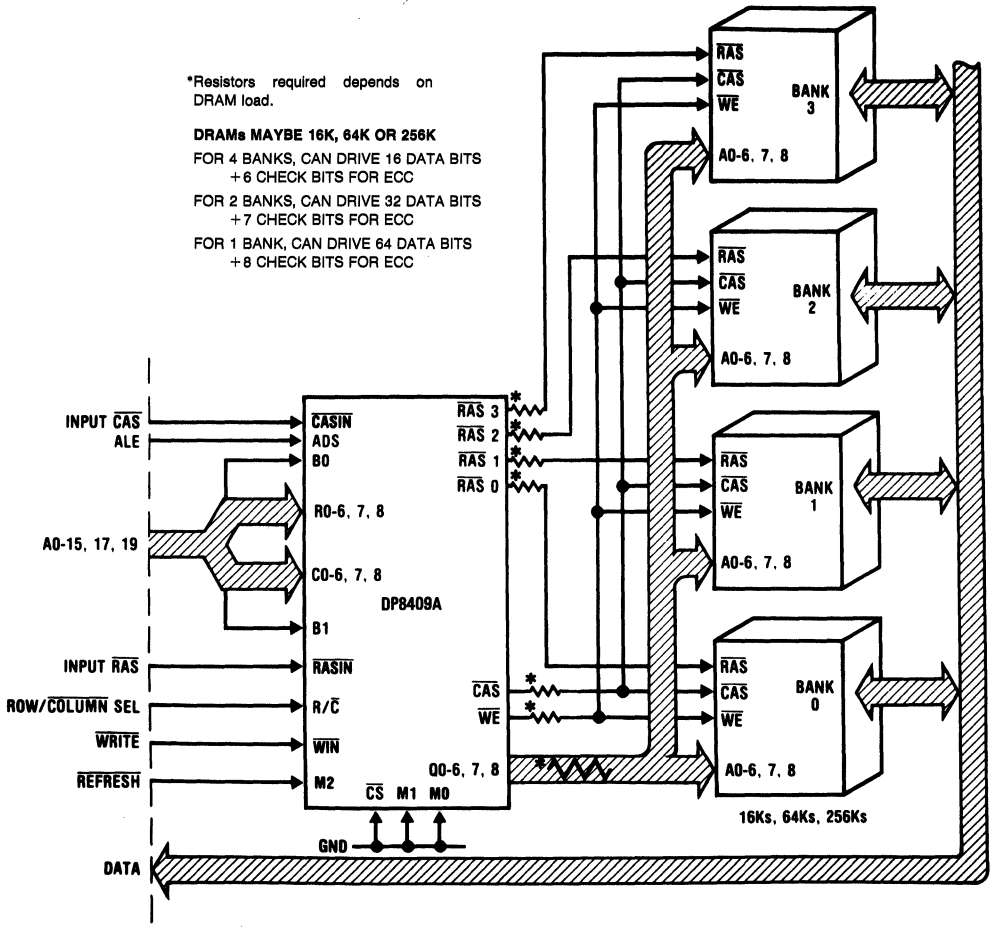
To select this mode, B1 and B0 must first have been set to 11 in Mode 7. This mode is useful at system initialization, but under processor control. The memory address is provided by the processor, which also performs the incrementing. All four $\overline{RAS}$ outputs follow $\overline{RASIN}$ (supplied by the processor), strobing the row address into the DRAMs. R/$\overline{C}$ can now go low, while $\overline{CASIN}$ may be used to control $\overline{CAS}$ (as in the Externally Controlled Access mode), so that $\overline{CAS}$ strobes the column address contents into the DRAMs. At this time $\overline{WE}$ should be low, causing the data to be written into all four banks of DRAMs. At the end of the write cycle, the input address is incremented and latched by the DP8409A for the next write cycle. This method is slower than Mode 3a since the processor must perform the incrementing and accessing. Thus the processor is occupied during RAM initialization, and is not free for other initialization operations. However, initialization sequence timing is under system control, which may provide some system advantage.

MODE 4—EXTERNALLY CONTROLLED ACCESS

This mode facilitates externally controlling all access-timing parameters associated with the DRAMs. The application of modes 0 and 4 are shown in *Figure 6*.

Output Address Selection

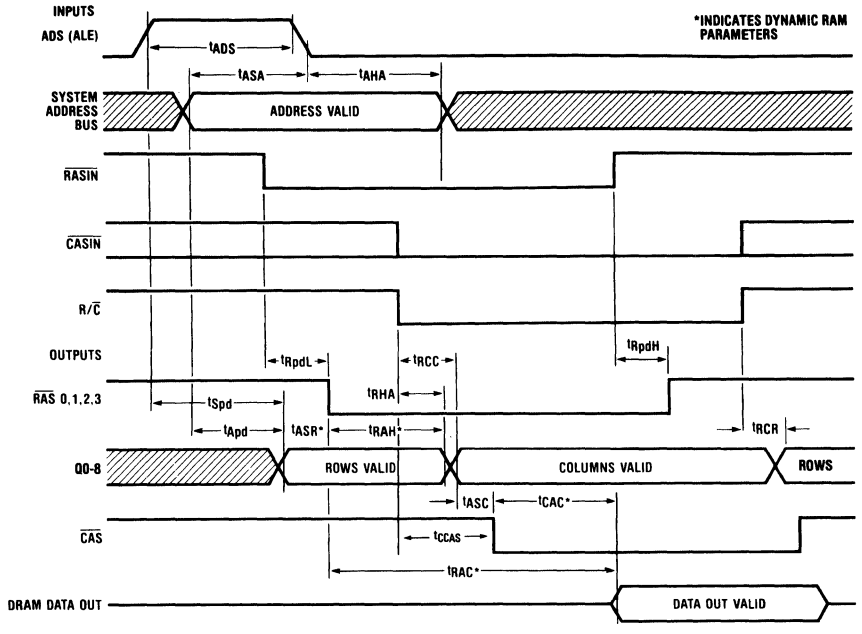
Refer to *Figure 7a*. With M2 ($\overline{RFSH}$) and R/$\overline{C}$ high, the row address latch contents are transferred to the multiplexed address bus output Q0–Q8, provided $\overline{CS}$ is set low. The column address latch contents are output after R/$\overline{C}$ goes low. $\overline{RASIN}$ can go low after the row addresses have been set up on Q0–Q8. This selects one of the $\overline{RAS}$ outputs, strobing the row address on the Q outputs into the desired bank of memory. After the row-address hold-time of the DRAMs, R/$\overline{C}$ can go low so that about 40 ns later column addresses appear on the Q outputs.



TL/F/8409–14

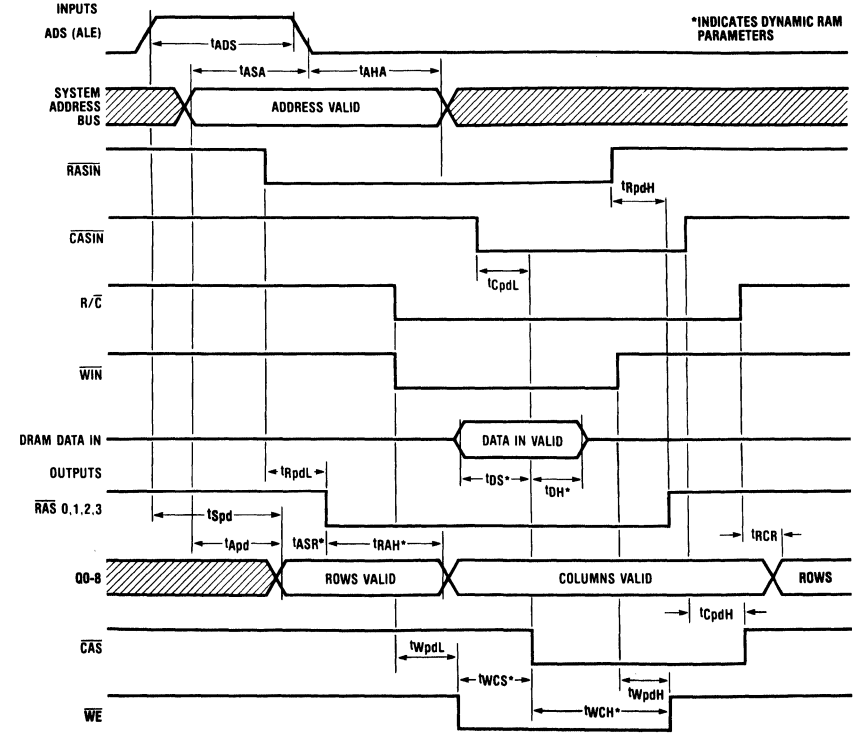
FIGURE 6. Typical Application of DP8409A Using External Control Access and Refresh in Modes 0 and 4

DP8409A Functional Mode Descriptions (Continued)



TL/F/8409–15

FIGURE 7a. Read Cycle Timing (Mode 4)



TL/F/8409–16

FIGURE 7b. Write Cycle Timing (Mode 4)

DP8409A Functional Mode Descriptions (Continued)

Automatic CAS Generation

In a normal memory access cycle $\overline{CAS}$ can be derived from inputs $\overline{CASIN}$ or R/$\overline{C}$. If $\overline{CASIN}$ is high, then R/C going low switches the address output drivers from rows to columns. $\overline{CASIN}$ then going low causes $\overline{CAS}$ to go low approximately 40 ns later, allowing $\overline{CAS}$ to occur at a predictable time (see *Figure 7b*). If $\overline{CASIN}$ is low when R/$\overline{C}$ goes low, $\overline{CAS}$ will be automatically generated, following the row to column transition by about 20 ns (see *Figure 7a*). Most DRAMs have a column address set-up time before $\overline{CAS}$ (t_{ASC}) of 0 ns or −10 ns. In other words, a t_{ASC} greater than 0 ns is safe.

Fast Memory Access

AC parameters t_{DIF1}, t_{DIF2} may be used to determine the minimum delays required between $\overline{RASIN}$, R/$\overline{C}$, and $\overline{CASIN}$ (see Application Brief 9; "Fastest DRAM Access Mode").

MODE 5—AUTOMATIC ACCESS WITH HIDDEN REFRESH

The Auto Access with Hidden Refresh mode has two advantages over the Externally Controlled Access mode, due to the fact that all outputs except $\overline{WE}$ are initiated from $\overline{RASIN}$. First, inputs R/$\overline{C}$ and $\overline{CASIN}$ are unnecessary and can be used for other functions (see Refreshing, below). Secondly, because the output control signals are derived internally from one input signal ($\overline{RASIN}$), timing-skew problems are reduced, thereby reducing memory access time substantially or allowing use of slower DRAMs. The automatic access features of Mode 5 (and Mode 6) of the DP8409A make DRAM accessing appear essentially "static".

Automatic Access Control

The major disadvantage of DRAMs compared to static RAMs is the complex timing involved. First, a $\overline{RAS}$ must occur with the row address previously set up on the multiplexed address bus. After the row address has been held for t_{RAH}, (the Row-Address hold-time of the DRAM), the column address is set up and then $\overline{CAS}$ occurs. This is all performed automatically by the DP8409A in this mode.

Provided the input address is valid as ADS goes low, $\overline{RASIN}$ can go low any time after ADS. This is because the selected $\overline{RAS}$ occurs typically 27 ns later, by which time the row address is already valid on the address output of the DP8409A. The Address Set-Up time (t_{ASR}), is 0 ns on most DRAMs. The DP8409A in this mode (with ADS and $\overline{RASIN}$ edges simultaneously applied) produces a minimum t_{ASR} of 0 ns. This is true provided the input address was valid t_{ASA} before ADS went low (see *Figure 8a*).

Next, the row address is disabled after t_{RAH} (30 ns minimum); in most DRAMs, t_{RAH} minimum is less than 30 ns. The column address is then set up and t_{ASC} later, $\overline{CAS}$ occurs. The only other control input required is $\overline{WIN}$. When a write cycle is required, $\overline{WIN}$ must go low at least 30 ns before $\overline{CAS}$ is output low.

This gives a total typical delay from: input address valid to $\overline{RASIN}$ (15 ns); to $\overline{RAS}$ (27 ns); to rows held (50 ns); to columns valid (25 ns); to $\overline{CAS}$ (23 ns) = 140 ns (that is, 125 ns from $\overline{RASIN}$). All of these typcial figures are for heavy capacitive loading, of approximately 88 DRAMs.

This mode is therefore extremely fast. The external timing is greatly simplified for the memory system designer: the only system signal required is $\overline{RASIN}$.

Refreshing

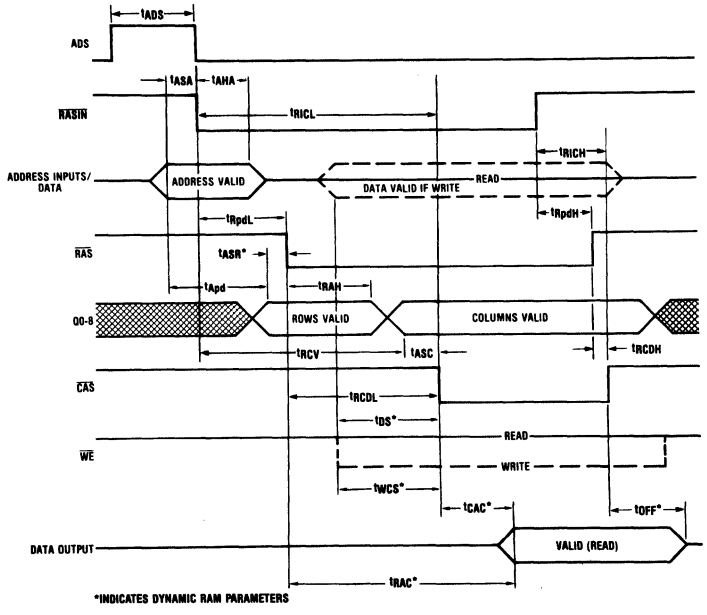
Because R/$\overline{C}$ and $\overline{CASIN}$ are not used in this mode, R/$\overline{C}$ becomes RFCK (refresh clock) and $\overline{CASIN}$ becomes RGCK ($\overline{RAS}$ generator clock). With these two signals it is possible to peform refreshing without extra ICs, and without holding up the processor.

One refresh cycle must occur during each refresh clock period and then the refresh address must be incremented to the next refresh cycle. As long as 128 rows are refreshed every 2 ms (one row every 16 μs), all 16k and 64k DRAMs will be correctly refreshed. The cycle time of RFCK must, therefore, be less than 16 μs. RFCK going high sets an internal refresh-request flip-flop. First the DP8409A will attempt to perform a hidden refresh so that the system throughput will not be affected. If, during the time RFCK is high, $\overline{CS}$ on the DP8409A goes high and $\overline{RASIN}$ occurs, a hidden refresh will occur. In this case, $\overline{RASIN}$ should be considered a common read/write strobe. In other words, if the processor is accessing elsewhere (other than the DRAMs) while RFCK is high, the DP8409A will perform a refresh. The refresh counter is enabled to the address outputs whenever $\overline{CS}$ goes high with RFCK high, and all $\overline{RAS}$ outputs follow $\overline{RASIN}$. If a hidden refresh is taking place as RFCK goes low, the refresh continues. At the start of the hidden refresh, the refresh-request flip-flop is reset so no further refresh can occur until the next RFCK period starts with the positive-going edge of RFCK. Refer to *Figure 9*.

To determine the probability of a Hidden Refresh occurring, assume each system cycle takes 400 ns and RFCK is high for 8 μs, then the system has 20 chances to not select the DP8409A. If during this time a hidden refresh did not occur, then the DP8409A forces a refresh while RFCK is low, but the system chooses when the refresh takes place. After RFCK goes low, (and the internal-request flip-flop has not been reset), RF I/O goes low indicating that a refresh is requested to the system. Only when the system acknowledges this request by setting M2 ($\overline{RFSH}$) low does the DP8409A initiate a forced refresh (which is performed automatically). Refer to Mode 1, and *Figure 3*. The internal refresh request flip-flop is then reset.

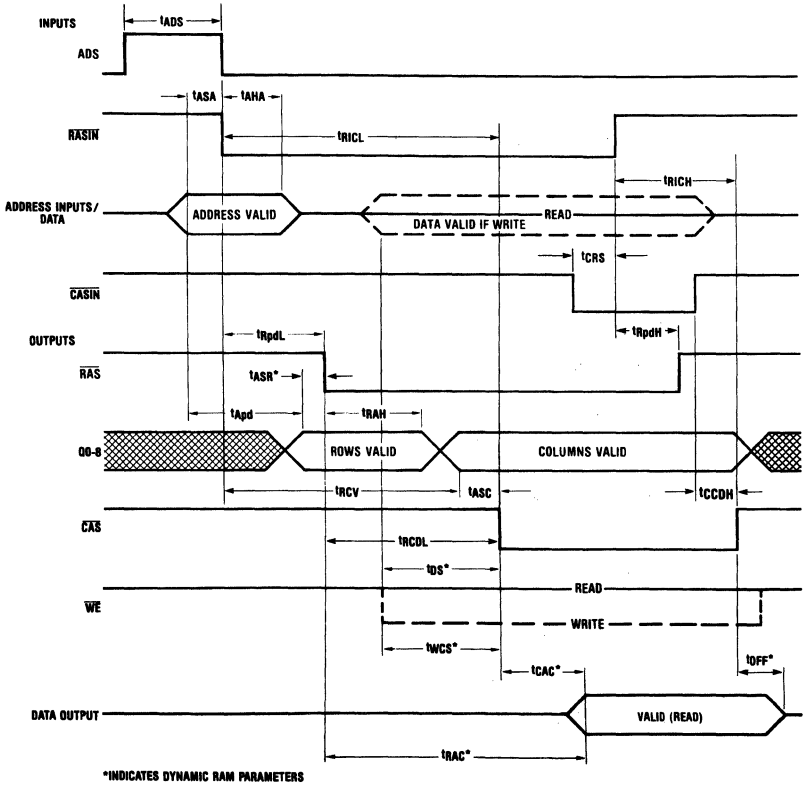
Figure 9 illustrates the refresh alternatives in Mode 5. If a hidden refresh has occurred and $\overline{CS}$ again goes high before RFCK goes low, the chip is deselected. All the control signals go high-impedance high (logic "1") and the address outputs go TRI-STATE until $\overline{CS}$ again goes low. This mode (combined with Mode 1) allows very fast access, and automatic refreshing (possibly not even slowing down the system), with no extra ICs. Careful system design can, and should, provide a higher probability of hidden refresh occurring. The duty cycle of RFCK need not be 50-percent; in fact, the low-time should be designed to be a minimum. This is determined by the worst-case time (required by the system) to respond to the DP8409A's forced-refresh request.

DP8409A Functional Mode Descriptions (Continued)



TL/F/8409–17

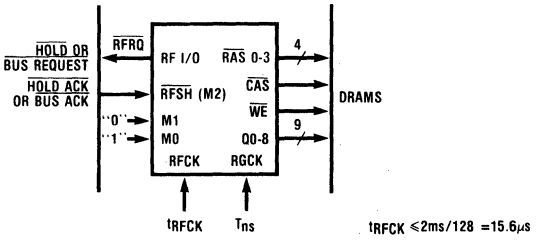
FIGURE 8a. Modes 5, 6 Timing ($\overline{\text{CASIN}}$ High in Mode 6)

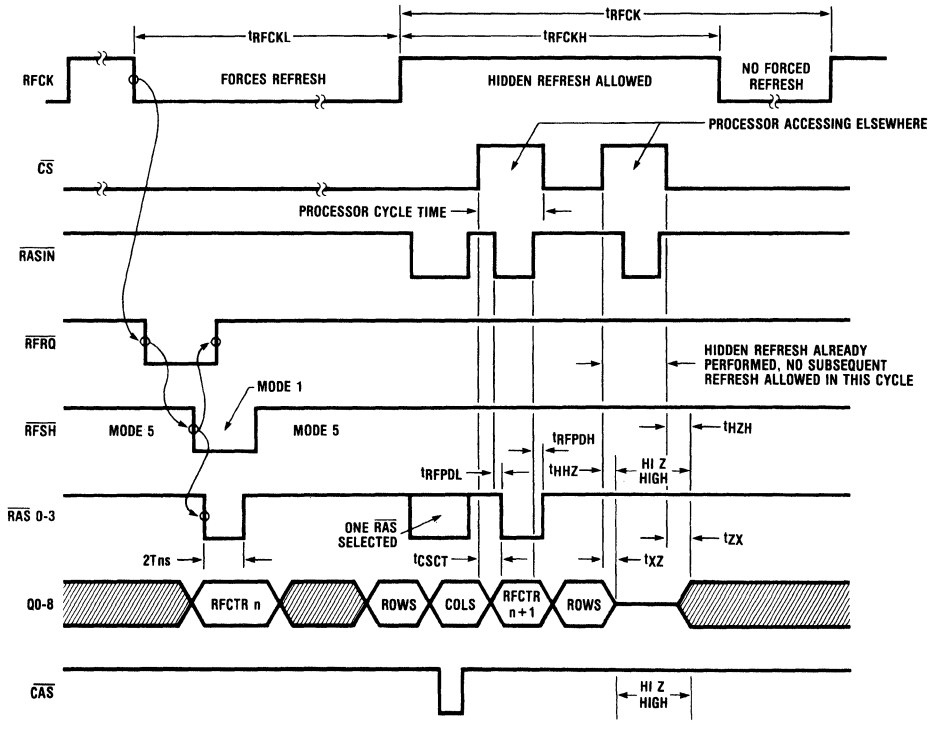


TL/F/8409–18

FIGURE 8b. Mode 6 Timing, Extended $\overline{\text{CAS}}$

DP8409A Functional Mode Descriptions (Continued)





TL/F/8409-19

FIGURE 9. Hidden Refreshing (Mode 5) and Forced Refreshing (Mode 1) Timing

DP8409A Functional Mode Descriptions (Continued)

TABLE II. Memory Bank Decode

Bank Select (Strobed by ADS)		Enabled $\overline{RAS}_n$
B1	B0	
0	0	$\overline{RAS}_0$
0	1	$\overline{RAS}_1$
1	0	$\overline{RAS}_2$
1	1	$\overline{RAS}_3$

Note that $\overline{RASIN}$ going low earlier than t_{CSRL} after $\overline{CS}$ goes low may result in the DP8409A interpreting the $\overline{RASIN}$ as a hidden refresh $\overline{RASIN}$ if no hidden refresh has occurred in the current RFCK cycle. In this case, all $\overline{RAS}$ outputs would go low for a short time. Thus, it is suggested that when using Mode 5, $\overline{RASIN}$ should be held high until t_{CSRL} after $\overline{CS}$ goes low if a refresh is not intended. Similarly, $\overline{CS}$ should be held low for a minimum of t_{CSRL} after $\overline{RASIN}$ returns high when ending the access in Mode 5.

MODE 6—FAST AUTOMATIC ACCESS

The Fast Access mode is similar to Mode 5, but has a faster t_{RAH} of 20 ns, minimum. It therefore can only be used with fast 16k or 64k DRAMs (which have a t_{RAH} of 10 ns to 15 ns) in applications requiring fast access times; $\overline{RASIN}$ to $\overline{CAS}$ is typically 105 ns.

In this mode, the R/$\overline{C}$ (RFCK) pin is not used, but $\overline{CASIN}$ (RGCK) is used as $\overline{CASIN}$ to allow an extended $\overline{CAS}$ after $\overline{RAS}$ has already terminated. Refer to *Figure 8b*. This is desirable with fast cycle-times where $\overline{RAS}$ has to be terminated as soon as possible before the next $\overline{RAS}$ begins (to meet the precharge time, or t_{RP}, requirements of the DRAM). $\overline{CAS}$ may then be held low by $\overline{CASIN}$ to extend the data output valid time from the DRAM to allow the system to read the data. $\overline{CASIN}$ subsequently going high ends $\overline{CAS}$. If this extended $\overline{CAS}$ is not required, $\overline{CASIN}$ should be set high in Mode 6.

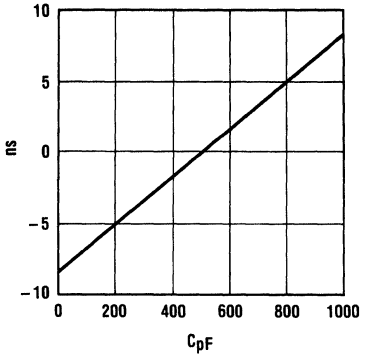
There is no internal refresh-request flip-flop in this mode, so any refreshing required must be done by entering Mode 0 or Mode 2.

MODE 7—SET END-OF-COUNT

The End-of-Count can be externally selected in Mode 7, using ADS to strobe in the respective value of B1 and B0 (see Table III). With B1 and B0 the same $\overline{EOC}$ is 127; with B1 = 0 and B0 = 1, $\overline{EOC}$ is 255; and with B1 = 1 and B0 = 0, $\overline{EOC}$ is 511. This selected value of $\overline{EOC}$ will be used until the next Mode 7 selection. At power-up the $\overline{EOC}$ is automatically set to 127 (B1 and B0 set to 11).

TABLE III. Mode 7

Bank Select (Strobed by ADS)		End of Count Selected
B1	B0	
0	0	127
0	1	255
1	0	511
1	1	127



TL/F/8409–20

FIGURE 10. Change in Propagation Delay vs. Loading Capacitance Relative to a 500 pF Load

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage, V_{CC}	7.0V
Storage Temperature Range	−65°C to +150°C
Input Voltage	5.5V
Output Current	150 mA
Lead Temperature (Soldering, 10 seconds)	300°C

Maximum Power Dissipation* at 25°C

Cavity Package	3542 mW
Molded Package	2833 mW

*Derate cavity package 23.6 mW/°C above 25°C; derate molded package 22.7 mW/°C above 25°C.

Operating Conditions

	Min	Max	Units
V_{CC} Supply Voltage	4.75	5.25	V
T_A Ambient Temperature	0	+70	°C

Electrical Characteristics V_{CC} = 5.0V ±5%, 0°C ≤ T_A ≤ 70°C (unless otherwise noted) (Notes 2, 6)

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_C	Input Clamp Voltage	V_{CC} = Min, I_C = −12 mA		−0.8	−1.2	V
I_{IH1}	Input High Current for ADS, R/$\overline{C}$ Only	V_{IN} = 2.5V		2.0	100	μA
I_{IH2}	Input High Current for All Other Inputs*	V_{IN} = 2.5V		1.0	50	μA
I_I RSI	Output Load Current for RF I/O	V_{IN} = 0.5V, Output High		−1.5	−2.5	mA
I_I CTL	Output Load Current for $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$	V_{IN} = 0.5V, Chip Deselect		−1.5	−2.5	mA
I_{IL1}	Input Low Current for ADS, R/$\overline{C}$ Only	V_{IN} = 0.5V		−0.1	−1.0	mA
I_{IL2}	Input Low Current for All Other Inputs*	V_{IN} = 0.5V		−0.05	−0.5	mA
V_{IL}	Input Low Threshold				0.8	V
V_{IH}	Input High Threshold		2.0			V
V_{OL1}	Output Low Voltage*	I_{OL} = 20 mA		0.3	0.5	V
V_{OL2}	Output Low Voltage for RF I/O	I_{OL} = 10 mA		0.3	0.5	V
V_{OH1}	Output High Voltage*	I_{OH} = −1 mA	2.4	3.5		V
V_{OH2}	Output High Voltage for RF I/O	I_{OH} = − 100 μA	2.4	3.5		V
I_{1D}	Output High Drive Current*	V_{OUT} = 0.8V (Note 3)		−200		mA
I_{0D}	Output Low Drive Current*	V_{OUT} = 2.7V (Note 3)		200		mA
I_{OZ}	TRI-STATE Output Current (Address Outputs)	0.4V ≤ V_{OUT} ≤ 2.7V, $\overline{CS}$ = 2.0V, Mode 4	−50	1.0	50	μA
I_{CC}	Supply Current	V_{CC} = Max		250	325	mA

*Except RF I/O Output.

Switching Characteristics: DP8409A/DP8409A-3

V_{CC} = 5.0V ±5%, 0°C ≤ T_A ≤ 70°C (unless otherwise noted) (Notes 2, 4, 5). The output load capacitance is typical for 4 banks of 22 DRAMs each or 88 DRAMs including trace capacitance. These values are: Q0–Q8, C_L = 500 pF; $\overline{RAS0}$–$\overline{RAS3}$, C_L = 150 pF; $\overline{WE}$, C_L = 500 pF; $\overline{CAS}$, C_L = 600 pF, (unless otherwise noted). See *Figure 11* for test load. Switches S1 and S2 are closed unless otherwise noted, and R1 and R2 are 4.7 kΩ unless otherwise noted. Maximum propagation delays are specified with all outputs switching.

Symbol	Parameter	Conditions	8409			8409A-3			Units
			Min	Typ	Max	Min	Typ	Max	
ACCESS									
t_{RICL}	$\overline{RASIN}$ to $\overline{CAS}$ Output Delay (Mode 5)	*Figure 8a*	95	125	160	95	125	185	ns
t_{RICL}	$\overline{RASIN}$ to $\overline{CAS}$ Output Delay (Mode 6)	*Figures 8a, 8b*	80	105	140	80	105	160	ns
t_{RICH}	$\overline{RASIN}$ to $\overline{CAS}$ Output Delay (Mode 5)	*Figure 8a*	40	48	60	40	48	70	ns
t_{RICH}	$\overline{RASIN}$ to $\overline{CAS}$ Output Delay (Mode 6)	*Figures 8a, 8b*	50	63	80	50	63	95	ns
t_{RCDL}	$\overline{RAS}$ to $\overline{CAS}$ Output Delay (Mode 5)	*Figure 8a*		98	125		98	145	ns
t_{RCDL}	$\overline{RAS}$ to $\overline{CAS}$ Output Delay (Mode 6)	*Figures 8a, 8b*		78	105		78	120	ns
t_{RCDH}	$\overline{RAS}$ to $\overline{CAS}$ Output Delay (Mode 5)	*Figure 8a*		27	40		27	40	ns
t_{RCDH}	$\overline{RAS}$ to $\overline{CAS}$ Output Delay (Mode 6)	*Figure 8a*		40	65		40	65	ns

Switching Characteristics: DP8409A/DP8409A-3 (Continued)

V_{CC} = 5.0V ±5%, 0°C ≤ T_A ≤ 70°C (unless otherwise noted) (Notes 2, 4, 5). The output load capacitance is typical for 4 banks of 22 DRAMs each or 88 DRAMs including trace capacitance. These values are: Q0–Q8, C_L = 500 pF; $\overline{RAS0}$–$\overline{RAS3}$, C_L = 150 pF; $\overline{WE}$, C_L = 500 pF; $\overline{CAS}$, C_L = 600 pF, (unless otherwise noted). See *Figure 11* for test load. Switches S1 and S2 are closed unless otherwise noted, and R1 and R2 are 4.7 kΩ unless otherwise noted. Maximum propagation delays are specified with all outputs switching.

Symbol	Parameter	Conditions	DP8409A			DP8409A-3			Units
			Min	Typ	Max	Min	Typ	Max	
ACCESS (Continued)									
t_{CCDH}	$\overline{CASIN}$ to $\overline{CAS}$ Output Delay (Mode 6)	*Figure 8b*	40	54	70	40	54	80	ns
t_{RAH}	Row Address Hold Time (Mode 5)	*Figure 8a*	30			30			ns
t_{RAH}	Row Address Hold Time (Mode 6)	*Figures 8a, 8b*	20			20			ns
t_{ASC}	Column Address Setup Time (Mode 5)	*Figure 8a*	8			8			ns
t_{ASC}	Column Address Setup Time (Mode 6)	*Figures 8a, 8b*	6			6			ns
t_{RCV}	$\overline{RASIN}$ to Column Address Valid (Mode 5)	*Figure 8a*		90	120		90	140	ns
t_{RCV}	$\overline{RASIN}$ to Column Address Valid (Mode 6)	*Figures 8a, 8b*		75	105		75	120	ns
t_{RPDL}	$\overline{RASIN}$ to $\overline{RAS}$ Delay	*Figures 7a, 7b, 8a, 8b*	20	27	35	20	27	40	ns
t_{RPDH}	$\overline{RASIN}$ to $\overline{RAS}$ Delay	*Figures 7a, 7b, 8a, 8b*	15	23	32	15	23	37	ns
t_{APDL}	Address Input to Output Low Delay	*Figures 7a, 7b, 8a, 8b*		25	40		25	46	ns
t_{APDH}	Address Input to Output High Delay	*Figures 7a, 7b, 8a, 8b*		25	40		25	46	ns
t_{SPDL}	Address Strobe to Address Output Low	*Figures 7a, 7b*		40	60		40	70	ns
t_{SPDH}	Address Strobe to Address Output High	*Figures 7a, 7b*		40	60		40	70	ns
t_{ASA}	Address Set-Up Time to ADS	*Figures 7a, 7b, 8a, 8b*	15			15			ns
t_{AHA}	Address Hold Time from ADS	*Figures 7a, 7b, 8a, 8b*	15			15			ns
t_{ADS}	Address Strobe Pulse Width	*Figures 7a, 7b, 8a, 8b*	30			30			ns
t_{WPDL}	$\overline{WIN}$ to $\overline{WE}$ Output Delay	*Figure 7b*	15	25	30	15	25	35	ns
t_{WPDH}	$\overline{WIN}$ to $\overline{WE}$ Output Delay	*Figure 7b*	15	30	60	15	30	70	ns
t_{CRS}	$\overline{CASIN}$ Set-Up Time to $\overline{RASIN}$ High (Mode 6)	*Figure 8b*	35			35			ns
t_{CPDL}	$\overline{CASIN}$ to $\overline{CAS}$ Delay (R/$\overline{C}$ Low in Mode 4)	*Figure 7b*	32	41	68	32	41	77	ns
t_{CPDH}	$\overline{CASIN}$ to $\overline{CAS}$ Delay (R/$\overline{C}$ Low in Mode 4)	*Figure 7b*	25	39	50	25	39	60	ns
t_{RCC}	Column Select to Column Address Valid	*Figure 7a*		40	58		40	67	ns
t_{RCR}	Row Select to Row Address Valid	*Figures 7a, 7b*		40	58		40	67	ns
t_{RHA}	Row Address Held from Column Select	*Figure 7a*	10			10			ns
t_{CCAS}	R/$\overline{C}$ Low to $\overline{CAS}$ Low (Mode 4 Auto $\overline{CAS}$)	*Figure 7a*		65	90				ns
t_{DIF1}	Maximum (t_{RPDL} − t_{RHA})	See Mode 4 Descrip.			13			18	ns
t_{DIF2}	Maximum (t_{RCC} − t_{CPDL})	See Mode 4 Descrip.			13			18	ns
REFRESH									
t_{RC}	Refresh Cycle Period	*Figure 2*	100			100			ns
$t_{RASINL, H}$	Pulse Width of $\overline{RASIN}$ during Refresh	*Figure 2*	50			50			ns
t_{RFPDL}	$\overline{RASIN}$ to $\overline{RAS}$ Delay during Refresh	*Figures 2, 9*	35	50	70	35	50	80	ns
t_{RFPDH}	$\overline{RASIN}$ to $\overline{RAS}$ Delay during Refresh	*Figures 2, 9*	30	40	55	30	40	65	ns
t_{RFLCT}	$\overline{RFSH}$ Low to Counter Address Valid	$\overline{CS}$ = X, *Figures 2, 3, 4*		47	60		47	70	ns

Switching Characteristics: DP8409A/DP8409A-3 (Continued)

V_{CC} = 5.0V ±5%, 0°C ≤ T_A ≤ 70°C (unless otherwise noted) (Notes 2, 4, 5). The output load capacitance is typical for 4 banks of 22 DRAMs each or 88 DRAMs including trace capacitance. These values are: Q0–Q8, C_L = 500 pF; $\overline{RAS0}$–$\overline{RAS3}$, C_L = 150 pF; $\overline{WE}$, C_L = 500 pF; $\overline{CAS}$, C_L = 600 pF, (unless otherwise noted). See *Figure 11* for test load. Switches S1 and S2 are closed unless otherwise noted, and R1 and R2 are 4.7 kΩ unless otherwise noted. Maximum propagation delays are specified with all outputs switching.

Symbol	Parameter	Conditions	DP8409A			DP8409A-3			Units
			Min	Typ	Max	Min	Typ	Max	
REFRESH (Continued)									
t_{RFHRV}	$\overline{RFSH}$ High to Row Address Valid	*Figures 2, 3*		45	60		45	70	ns
t_{ROHNC}	$\overline{RAS}$ High to New Count Valid	*Figures 2, 4*		30	55		30	55	ns
t_{RLEOC}	$\overline{RASIN}$ Low to End-of-Count Low	C_L = 50 pF, *Figure 2*			80			80	ns
t_{RHEOC}	$\overline{RASIN}$ High to End-of-Count High	C_L = 50 pF, *Figure 2*			80			80	ns
t_{RGEOB}	RGCK Low to End-of-Burst Low	C_L = 50 pF, *Figure 4*			95			95	ns
t_{MCEOB}	Mode Change to End-of-Burst High	C_L = 50 pF, *Figure 4*			75			75	ns
t_{RST}	Counter Reset Pulse Width	*Figure 2*	70			70			ns
t_{CTL}	RF I/O Low to Counter Outputs All Low	*Figure 2*			100			100	ns
$t_{RFCKL, H}$	Minimum Pulse Width of RFCK	*Figure 9*	100			100			ns
T	Period of $\overline{RAS}$ Generator Clock	*Figure 3*	100			100			ns
t_{RGCKL}	Minimum Pulse Width Low of RGCK	*Figure 3*	35			40			ns
t_{RGCKH}	Minimum Pulse Width High of RGCK	*Figure 3*	35			40			ns
t_{FRQL}	RFCK Low to Forced $\overline{RFRQ}$ Low	C_L = 50 pF, *Figure 3*		20	30		20	30	ns
t_{FRQH}	RGCK Low to Forced $\overline{RFRQ}$ High	C_L = 50 pF, *Figure 3*		50	75		50	75	ns
t_{RGRL}	RGCK Low to $\overline{RAS}$ Low	*Figure 3*	50	65	95	50	65	95	ns
t_{RGRH}	RGCK Low to $\overline{RAS}$ High	*Figure 3*	40	60	85	40	60	85	ns
t_{RQHRF}	$\overline{RFSH}$ Hold Time from $\overline{RFSH}$ $\overline{RQST}$ (RF I/O)	*Figure 3*	2T			2T			ns
t_{RFRH}	$\overline{RFSH}$ High to $\overline{RAS}$ High (ending forced RFSH)	See Mode 1 Descrip.	55	80	110	55	80	125	ns
t_{RFSRG}	$\overline{RFSH}$ Low Set-Up to RGCK Low (Mode 1)	See Mode 1 Descrip.	35			40			ns
t_{CSCT}	$\overline{CS}$ High to RFSH Counter Valid	*Figure 9*		55	70		55	75	ns
t_{CSRL}	$\overline{CS}$ Low to Access $\overline{RASIN}$ Low	See Mode 5 Descrip.	30			30			ns
TRI-STATE									
t_{ZH}	$\overline{CS}$ Low to Address Output High from Hi-Z	*Figures 9, 12,* R1 = 3.5k, R2 = 1.5k		35	60		35	60	ns
t_{HZ}	$\overline{CS}$ High to Address Output Hi-Z from High	C_L = 15 pF, *Figures 9, 12,* R2 = 1k, S1 Open		20	40		20	40	ns
t_{ZL}	$\overline{CS}$ Low to Address Output Low from Hi-Z	*Figures 9, 12,* R1 = 3.5k, R2 = 1.5k		35	60		35	60	ns
t_{LZ}	$\overline{CS}$ High to Address Output Hi-Z from Low	C_L = 15 pF, *Figures 9, 12,* R1 = 1k, S2 Open		25	50		25	50	ns
t_{HZH}	$\overline{CS}$ Low to Control Output High from Hi-Z High	*Figures 9, 12,* R2 = 750Ω, S1 Open		50	80		50	80	ns
t_{HHZ}	$\overline{CS}$ High to Control Output Hi-Z High from High	C_L = 15 pF, *Figures 9, 12,* R2 = 750Ω, S1 Open		40	75		40	75	ns

Switching Characteristics: DP8409A/DP8409A-3 (Continued)

V_{CC} = 5.0V ±5%, 0°C ≤ T_A ≤ 70°C (unless otherwise noted) (Notes 2, 4, 5). The output load capacitance is typical for 4 banks of 22 DRAMs each or 88 DRAMs including trace capacitance. These values are: Q0–Q8, C_L = 500 pF; $\overline{RAS0}$–$\overline{RAS3}$, C_L = 150 pF; $\overline{WE}$, C_L = 500 pF; $\overline{CAS}$, C_L = 600 pF, (unless otherwise noted). See *Figure 11* for test load. Switches S1 and S2 are closed unless otherwise noted, and R1 and R2 are 4.7 kΩ unless otherwise noted. Maximum propagation delays are specified with all outputs switching.

Symbol	Parameter	Conditions	DP8409A			DP8409A-3			Units
			Min	Typ	Max	Min	Typ	Max	
TRI-STATE (Continued)									
t_{HZL}	$\overline{CS}$ Low to Control Output Low from Hi-Z High	*Figure 12*, S1, S2 Open		45	75		45	75	ns
t_{LHZ}	$\overline{CS}$ High to Control Output Hi-Z High from Low	C_L = 15 pF, *Figure 12*, R2 = 750Ω, S1 Open		50	80		50	80	ns

Switching Characteristics: DP8409A-2

V_{CC} = 5.0V ±5%, 0°C ≤ T_A ≤ 70°C (unless otherwise noted) (Notes 2, 4, 5). The output load capacitance is typical for 4 banks of 22 DRAMs each or 88 DRAMs including trace capacitance. These values are: Q0–Q8, C_L = 500 pF; $\overline{RAS0}$–$\overline{RAS3}$, C_L = 150 pF; $\overline{WE}$, C_L = 500 pF; $\overline{CAS}$, C_L = 600 pF, (unless otherwise noted). See *Figure 11* for test load. Switches S1 and S2 are closed unless otherwise noted, and R1 and R2 are 4.7 kΩ unless otherwise noted. Maximum propagation delays are specified with all outputs switching.

Symbol	Parameter	Conditions	8409A-2			Units
			Min	Typ	Max	
ACCESS						
t_{RICL}	$\overline{RASIN}$ to $\overline{CAS}$ Output Delay (Mode 5)	*Figure 8a*	75	100	130	ns
t_{RICL}	$\overline{RASIN}$ to $\overline{CAS}$ Output Delay (Mode 6)	*Figures 8a, 8b*	65	90	115	ns
t_{RICH}	$\overline{RASIN}$ to $\overline{CAS}$ Output Delay (Mode 5)	*Figure 8a*	40	48	60	ns
t_{RICH}	$\overline{RASIN}$ to $\overline{CAS}$ Output Delay (Mode 6)	*Figures 8a, 8b*	50	63	80	ns
t_{RCDL}	$\overline{RAS}$ to $\overline{CAS}$ Output Delay (Mode 5)	*Figure 8a*		75	100	ns
t_{RCDL}	$\overline{RAS}$ to $\overline{CAS}$ Output Delay (Mode 6)	*Figures 8a, 8b*		65	85	ns
t_{RCDH}	$\overline{RAS}$ to $\overline{CAS}$ Output Delay (Mode 5)	*Figure 8a*		27	40	ns
t_{RCDH}	$\overline{RAS}$ to $\overline{CAS}$ Output Delay (Mode 6)	*Figure 8a*		40	65	ns
t_{CCDH}	$\overline{CASIN}$ to $\overline{CAS}$ Output Delay (Mode 6)	*Figure 8b*	40	54	70	ns
t_{RAH}	Row Address Hold Time (Mode 5) (Note 7)	*Figure 8a*	20			ns
t_{RAH}	Row Address Hold Time (Mode 6) (Note 7)	*Figures 8a, 8b*	12			ns
t_{ASC}	Column Address Set-Up Time (Mode 5)	*Figure 8a*	3			ns
t_{ASC}	Column Address Set-Up Time (Mode 6)	*Figures 8a, 8b*	3			ns
t_{RCV}	$\overline{RASIN}$ to Column Address Valid (Mode 5)	*Figure 8a*		80	105	ns
t_{RCV}	$\overline{RASIN}$ to Column Address Valid (Mode 6)	*Figures 8a, 8b*		70	90	ns
t_{RPDL}	$\overline{RASIN}$ to $\overline{RAS}$ Delay	*Figures 7a, 7b, 8a, 8b*	20	27	35	ns
t_{RPDH}	$\overline{RASIN}$ to $\overline{RAS}$ Delay	*Figures 7a, 7b, 8a, 8b*	15	23	32	ns
t_{APDL}	Address Input to Output Low Delay	*Figures 7a, 7b, 8a, 8b*		25	40	ns
t_{APDH}	Address Input to Output High Delay	*Figures 7a, 7b, 8a, 8b*		25	40	ns
t_{SPDL}	Address Strobe to Address Output Low	*Figures 7a, 7b*		40	60	ns
t_{SPDH}	Address Strobe to Address Output High	*Figures 7a, 7b*		40	60	ns
t_{ASA}	Address Set-Up Time to ADS	*Figures 7a, 7b, 8a, 8b*	15			ns
t_{AHA}	Address Hold Time from ADS	*Figures 7a, 7b, 8a, 8b*	15			ns
t_{ADS}	Address Strobe Pulse Width	*Figures 7a, 7b, 8a, 8b*	30			ns

Switching Characteristics: DP8409A-2 (Continued)

V_{CC} = 5.0V ±5%, 0°C ≤ T_A ≤ 70°C (unless otherwise noted) (Notes 2, 4, 5). The output load capacitance is typical for 4 banks of 22 DRAMs each or 88 DRAMs including trace capacitance. These values are: Q0–Q8, C_L = 500 pF; $\overline{RAS0}$–$\overline{RAS3}$, C_L = 150 pF; $\overline{WE}$, C_L = 500 pF; $\overline{CAS}$, C_L = 600 pF, (unless otherwise noted). See *Figure 11* for test load. Switches S1 and S2 are closed unless otherwise noted, and R1 and R2 are 4.7 kΩ unless otherwise noted. Maximum propagation delays are specified with all outputs switching.

Symbol	Parameter	Conditions	8409A-2			Units
			Min	Typ	Max	
ACCESS (Continued)						
t_{WPDL}	$\overline{WIN}$ to $\overline{WE}$ Output Delay	*Figure 7b*	15	25	30	ns
t_{WPDH}	$\overline{WIN}$ to $\overline{WE}$ Output Delay	*Figure 7b*	15	30	60	ns
t_{CRS}	$\overline{CASIN}$ Set-Up Time to $\overline{RASIN}$ High (Mode 6)	*Figure 8b*	35			ns
t_{CPDL}	$\overline{CASIN}$ to $\overline{CAS}$ Delay (R/$\overline{C}$ Low in Mode 4)	*Figure 7b*	32	41	58	ns
t_{CPDH}	$\overline{CASIN}$ to $\overline{CAS}$ Delay (R/$\overline{C}$ Low in Mode 4)	*Figure 7b*	25	39	50	ns
t_{RCC}	Column Select to Column Address Valid	*Figure 7a*		40	58	ns
t_{RCR}	Row Select to Row Address Valid	*Figures 7a, 7b*		40	58	ns
t_{RHA}	Row Address Held from Column Select	*Figure 7a*	10			ns
t_{CCAS}	R/$\overline{C}$ Low to $\overline{CAS}$ Low (Mode 4 Auto $\overline{CAS}$)	*Figure 7a*		55	75	ns
t_{DIF1}	Maximum ($t_{RPDL} - t_{RHA}$)	See Mode 4 Descript.			13	ns
t_{DIF2}	Maximum ($t_{RCC} - t_{CPDL}$)	See Mode 4 Descript.			13	ns
REFRESH						
t_{RC}	Refresh Cycle Period	*Figure 2*	100			ns
$t_{RASINL, H}$	Pulse Width of $\overline{RASIN}$ during Refresh	*Figure 2*	50			ns
t_{RFPDL}	$\overline{RASIN}$ to $\overline{RAS}$ Delay during Refresh	*Figures 2, 9*	35	50	70	ns
t_{RFPDH}	$\overline{RASIN}$ to $\overline{RAS}$ Delay during Refresh	*Figures 2, 9*	30	40	55	ns
t_{RFLCT}	$\overline{RFSH}$ Low to Counter Address Valid	$\overline{CS}$ = X, *Figures 2, 3, 4*		47	60	ns
t_{RFHRV}	$\overline{RFSH}$ High to Row Address Valid	*Figures 2, 3*		45	60	ns
t_{ROHNC}	$\overline{RAS}$ High to New Count Valid	*Figures 2, 4*		30	55	ns
t_{RLEOC}	$\overline{RASIN}$ Low to End-of-Count Low	C_L = 50 pF, *Figure 2*			80	ns
t_{RHEOC}	$\overline{RASIN}$ High to End-of-Count High	C_L = 50 pF, *Figure 2*			80	ns
t_{RGEOB}	RGCK Low to End-of-Burst Low	C_L = 50 pF, *Figure 4*			95	ns
t_{MCEOB}	Mode Change to End-of-Burst High	C_L = 50 pF, *Figure 4*			75	ns
t_{RST}	Counter Reset Pulse Width	*Figure 2*	70			ns
t_{CTL}	RF I/O Low to Counter Outputs All Low	*Figure 2*			100	ns
$t_{RFCKL, H}$	Minimum Pulse Width of RFCK	*Figure 9*	100			ns
T	Period of $\overline{RAS}$ Generator Clock	*Figure 3*	100			ns
t_{RGCKL}	Minimum Pulse Width Low of RGCK	*Figure 3*	35			ns
t_{RGCKH}	Minimum Pulse Width High of RGCK	*Figure 3*	35			ns
t_{FRQL}	RFCK Low to Forced $\overline{RFRQ}$ Low	C_L = 50 pF, *Figure 3*		20	30	ns

Switching Characteristics: DP8409A-2 (Continued)

V_{CC} = 5.0V ±5%, 0°C ≤ T_A ≤ 70°C (unless otherwise noted) (Notes 2, 4, 5). The output load capacitance is typical for 4 banks of 22 DRAMs each or 88 DRAMs including trace capacitance. These values are: Q0–Q8, C_L = 500 pF; $\overline{RAS0}$–$\overline{RAS3}$, C_L = 150 pF; $\overline{WE}$, C_L = 500 pF; $\overline{CAS}$, C_L = 600 pF, (unless otherwise noted). See *Figure 11* for test load. Switches S1 and S2 are closed unless otherwise noted, and R1 and R2 are 4.7 kΩ unless otherwise noted. Maximum propagation delays are specified with all outputs switching.

Symbol	Parameter	Conditions	8409A-2			Units
			Min	Typ	Max	
REFRESH (Continued)						
t_{FRQH}	RGCK Low to Forced $\overline{RFRQ}$ High	C_L = 50 pF, *Figure 3*		50	75	ns
t_{RGRL}	RGCK Low to $\overline{RAS}$ Low	*Figure 3*	50	65	95	ns
t_{RGRH}	RGCK Low to $\overline{RAS}$ High	*Figure 3*	40	60	85	ns
t_{RQHRF}	$\overline{RFSH}$ Hold Time from $\overline{RFSH\ RQST}$ (RF I/O)	*Figure 3*	2T			ns
t_{RFRH}	$\overline{RFSH}$ High to $\overline{RAS}$ High (Ending Forced RFSH)	See Mode 1 Descrip.	55	80	110	ns
t_{RFSRG}	$\overline{RFSH}$ Low Set-Up to RGCK Low (Mode 1)	See Mode 1 Descrip.	35			ns
t_{CSCT}	$\overline{CS}$ High to RFSH Counter Valid	*Figure 9*		55	70	ns
t_{CSRL}	$\overline{CS}$ Low to Access $\overline{RASIN}$ Low	See Mode 5 Descrip.	30			ns
t_{ZH}	$\overline{CS}$ Low to Address Output High from Hi-Z	*Figures 9, 12,* R1 = 3.5k, R2 = 1.5k		35	60	ns
t_{HZ}	$\overline{CS}$ High to Address Output Hi-Z from High	C_L = 15 pF, *Figures 9, 12,* R2 = 1k, S1 Open		20	40	ns
t_{ZL}	$\overline{CS}$ Low to Address Output Low from Hi-Z	*Figures 9, 12,* R1 = 3.5k, R2 = 1.5k		35	60	ns
t_{LZ}	$\overline{CS}$ High to Address Output Hi-Z from Low	C_L = 15 pF, *Figures 9, 12,* R1 = 1k, S2 Open		25	50	ns
t_{HZH}	$\overline{CS}$ Low to Control Output High from Hi-Z High	*Figures 9, 12,* R2 = 750Ω, S1 Open		50	80	ns
t_{HHZ}	$\overline{CS}$ High to Control Output Hi-Z High from High	C_L = 15 pF, *Figures 9, 12,* R2 = 750Ω, S1 Open		40	75	ns
t_{HZL}	$\overline{CS}$ Low to Control Output Low from Hi-Z High	*Figure 12,* S1, S2 Open		45	75	ns
t_{LHZ}	$\overline{CS}$ High to Control Output Hi-Z High from Low	C_L = 15 pF, *Figure 12,* R2 = 750Ω, S1 Open		50	80	ns

Input Capacitance T_A = 25°C (Notes 2, 6)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
C_{IN}	Input Capacitance ADS, R/$\overline{C}$			8		pF
C_{IN}	Input Capacitance All Other Inputs			5		pF

Note 1: "Absolute Maximum Ratings" are the values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: All typical values are for T_A = 25°C and V_{CC} = 5.0V.

Note 3: This test is provided as a monitor of Driver output source and sink current capability. Caution should be exercised in testing these parameters. In testing these parameters, a 15Ω resistor should be placed in series with each output under test. One output should be tested at a time and test time should not exceed 1 second.

Note 4: Input pulse 0V to 3.0V, t_R = t_F = 2.5 ns, f = 2.5 MHz, t_{PW} = 200 ns. Input reference point on AC measurements is 1.5V. Output reference points are 2.7V for High and 0.8V for Low.

Note 5: The load capacitance on RF I/O should not exceed 50 pF.

Note 6: Applies to all DP8409A versions unless otherwise specified.

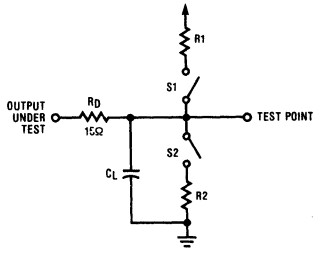
Note 7: The DP8409A-2 device can only be used with memory devices that meet the t_{RAH} specification indicated.

Applications

If external control is preferred, the DP8409A may be used in Mode 0 or 4, as in *Figure 6*.

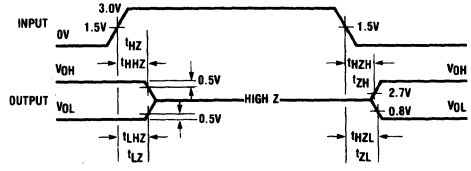
If basic auto access and refresh are required, then in cases where the user requires the minimum of external complexity, Modes 1 and 5 are ideal, as shown in *Figure 13a*. The DP843X2 is used to provide proper arbitration between memory access and refresh. This chip supplies all the necessary control signals to the processor as well as the DP8409A. Furthermore, two separate $\overline{CAS}$ outputs are also included for systems using byte-writing. The refresh clock RFCK may be divided down from either RGCK using an IC counter such as the DM74LS393 or better still, the DP84300 Programmable Refresh Timer. The DP84300 can provide RFCK periods ranging from 15.4 μs to 15.6 μs based on the input clock of 2 to 10 MHz. *Figure 13b* shows the general timing diagram for interfacing the DP8409A to different microprocessors using the interface controller DP843X2.

If the system is complex, requiring automatic access and refresh, burst refresh, and all-banks auto-write, then more circuitry is required to select the mode. This may be accomplished by utilizing a PAL®. The PAL has two functions. One as an address comparator, so that when the desired port address occurs (programmed in the PAL), the comparator gates the data into a latch, where it is connected to the mode pins of the DP8409A. Hence the mode of the DP8409A can be changed as desired with one PAL chip merely by addressing the PAL location, and then outputting data to the mode-control pins. In this manner, all the automatic modes may be selected, assigning R/C as RFCK always, and $\overline{CASIN}$ as RGCK always. The output from RF I/O may be used as End-of-Count to an interrupt, or $\overline{Refresh\ Request}$ to $\overline{HOLD}$ or $\overline{BUS\ REQUEST}$. A complex system may use Modes 5 and 1 for automatic access and refresh, Modes 3a and 7 for system initialization, and Mode 2 (auto-burst refresh) before and after DMA.



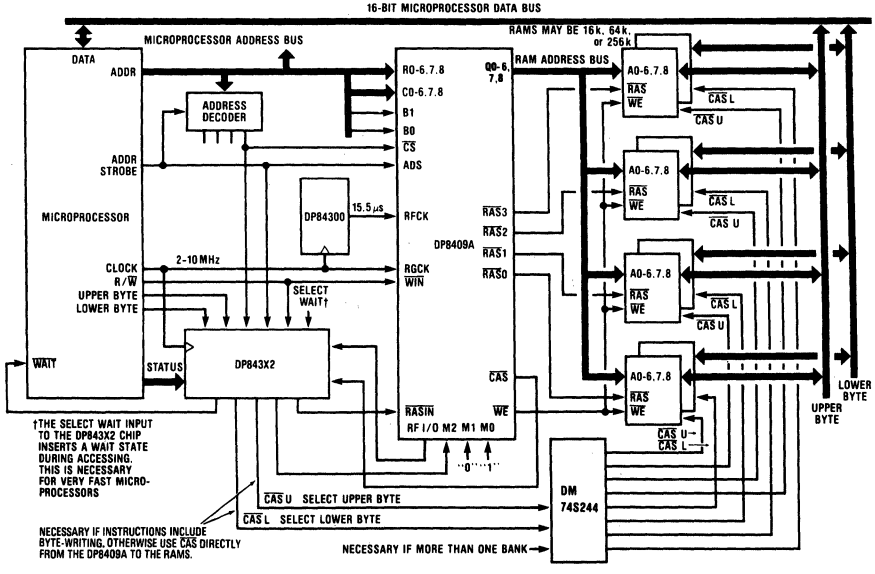
TL/F/8409–21

FIGURE 11. Output Load Circuit



TL/F/8409–22

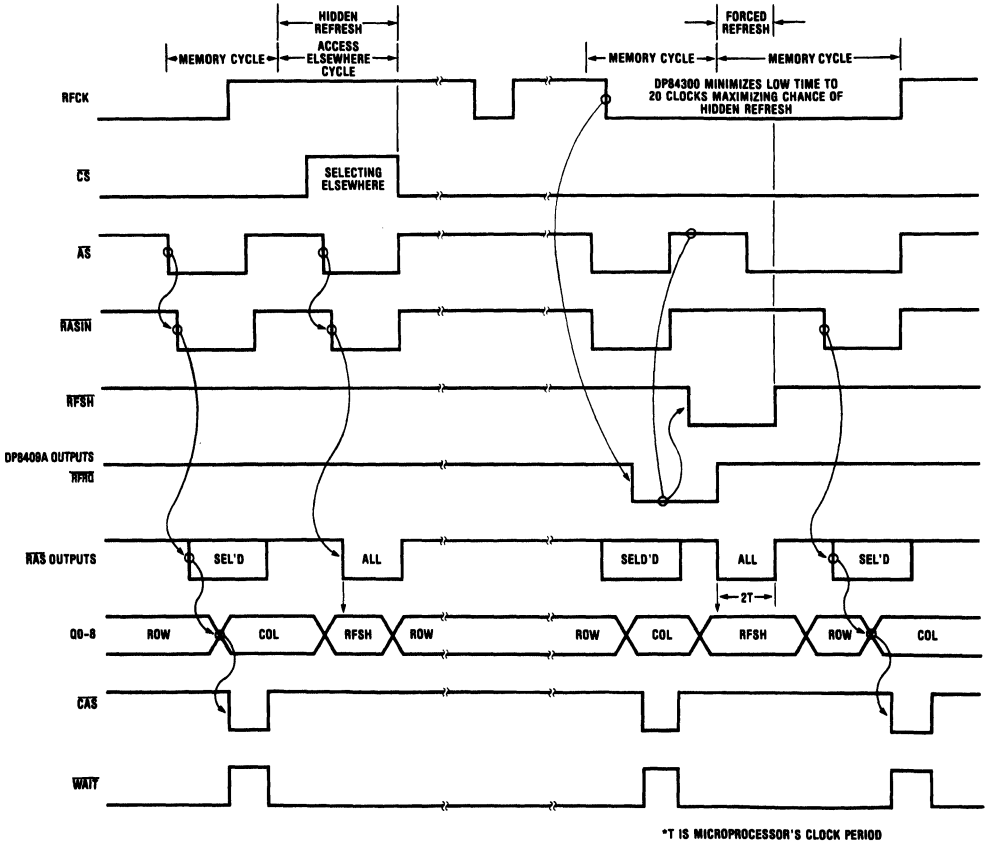
FIGURE 12. Waveform



TL/F/8409–23

FIGURE 13a. Connecting the DP8409A Between the 16-Bit Microprocessor and Memory

Applications (Continued)



TL/F/8409–24

FIGURE 13b. DP8409A Auto Refresh

National Semiconductor

PRELIMINARY

DP8417/NS32817, 8418/32818, 8419/32819, 8419X/ 32819X 64k, 256k Dynamic RAM Controller/Drivers

General Description

The DP8417/8418/8419/8419X represent a family of 256k DRAM Controller/Drivers which are designed to provide "No-Waitstate" CPU interface to Dynamic RAM arrays of up to 2 Mbytes and larger. Each device offers slight functional variations of the DP8419 design which are tailored for different system requirements. All family members are fabricated using National's new oxide isolated Advanced Low power Schottky (ALS) process and use design techniques which enable them to significantly out-perform all other LSI or discrete alternatives in speed, level of integration, and power consumption.

Each device integrates the following critical 256k DRAM controller functions on a single monolithic device: ultra precise delay line; 9-bit refresh counter; fall-through row, column, and bank select input latches; Row/Column address muxing logic; on-board high capacitive-load $\overline{RAS}$, $\overline{CAS}$, and Write Enable & Address output drivers; and, precise control signal timing for all the above.

There are four device options of the basic DP8419 Controller. The DP8417 is pin and function compatible with the DP8419 except that its outputs are TRI-STATE®. The DP8418 changes one pin and is specifically designed to offer an optimum interface to 32 bit microprocessors. The DP8419X is functionally identical to the DP8419, but is available in a 52-pin DIP package which is upward pin compatible with National's new DP8429D 1 Mbit DRAM Controller/ Driver.

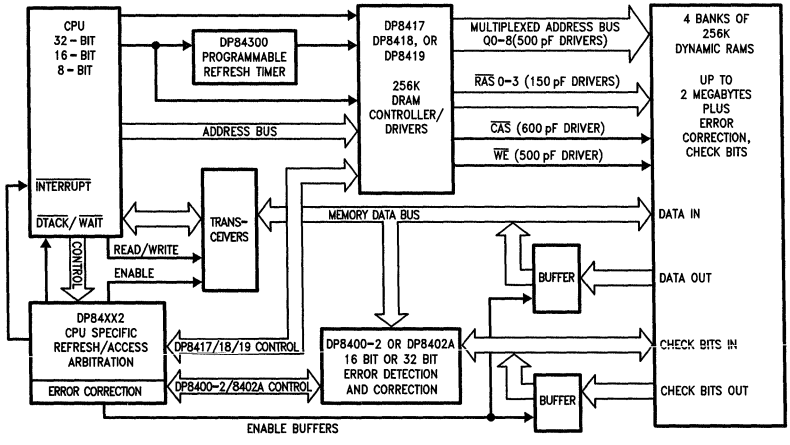
Each device is available in plastic DIP, Ceramic DIP, and Plastic Chip Carrier (PCC) packaging. (Continued)

Operational Features

- Makes DRAM Interface and refresh tasks appear virtually transparent to the CPU, making DRAMs as easy to use as static RAMs
- Specifically designed to eliminate CPU wait states up to 10 MHz or beyond
- Eliminates 15 to 20 SSI/MSI components for significant board real estate reduction, system power savings and the elimination of chip-to-chip AC skewing
- On-board ultra precise delay line
- On-board high capacitive $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$, and address drivers (specified driving 88 DRAMs directly)
- AC specified for directly addressing up to 8 Megabytes
- Low power/high speed bipolar oxide isolated process
- Upward pin and function compatible with new DP8428/ DP8429 1 Mbit DRAM controller drivers
- Downward pin and function compatible with DP8408A/ DP8409A 64k/256k DRAM controller/drivers
- 4 user selectable modes of operation for Access and Refresh (2 automatic, 2 external)

Contents

System Diagram



TL/F/8396-25

General Description (Continued)

In order to specify each device for "true" worst case operating conditions, all timing parameters are guaranteed while the chip is driving the capacitive load of 88 DRAMs including trace capacitance. The chip's delay timing logic makes use of a patented new delay line technique which keeps A.C. skew to ±3 ns over the full V_{CC} range of ±10% and temperature range of −55°C to +125°C. The DP8417, DP8418, DP8419, and DP8419X guarantee a maximum $\overline{RASIN}$ to CASOUT delay of 80 ns or 70 ns even while driving a 2 Mbyte memory array with error correction check bits included. Speed selected options of these devices are shown in the switching characteristics section of this document.

With its four independent $\overline{RAS}$ outputs and nine multiplexed address outputs, the DP8419 can support up to four banks of 16k, 64k or 256k DRAMs. Two bank select pins, B1 and B0, are decoded to activate one of the $\overline{RAS}$ signals during an access, leaving the three non-selected banks in the standby mode (less than one tenth of the operating power) with data outputs in TRI-STATE.

The DP8419 has two mode-select pins, allowing for two refresh modes and two access modes. Refresh and access timing may be controlled either externally or automatically. The automatic modes require a minimum of input control signals.

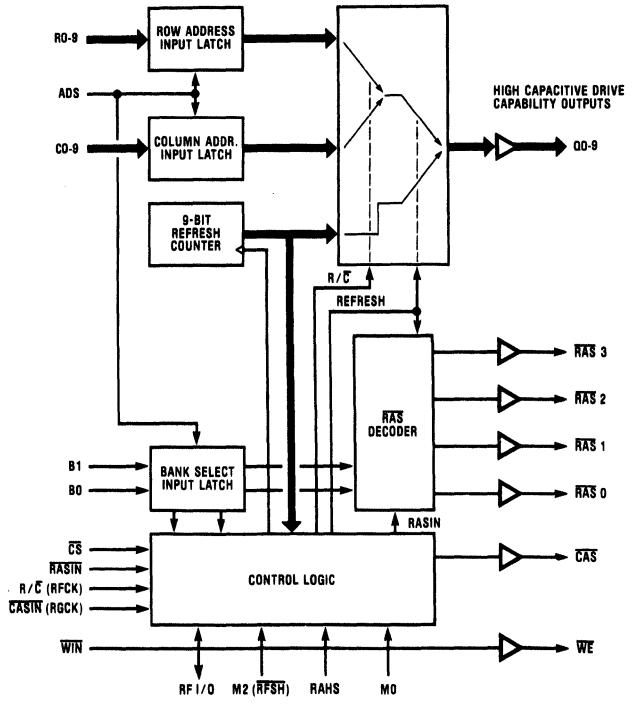
A refresh counter is on-chip and is multiplexed with the row and column inputs. Its contents appear at the address outputs of the DP8419 during any refresh, and are incremented at the completion of the refresh. Row/Column and bank address latches are also on-chip. However, if the address inputs to the DP8419 are valid throughout the duration of the access, these latches may be operated in the fall-through mode.

System Companion Components

Device #	Function
DP84300	Programmable Refresh Timer for DP84xx DRAM Controller
DP84412	NS32008/16/32 to DP8409A/17/18/19/28/29 Interface
DP84512	NS32332 to DP8417/18/19/28/29 Interface
DP84322	68000/08/10 to DP8409A/17/18/19/28/29 Interface (up to 8 MHz)
DP84422	68000/08/10 to DP8409A/17/18/19/28/29 Interface (up to 12.5 MHz)
DP84522	68020 to DP8417/18/19/28/29 Interface
DP84432	8086/88/186/188 to DP8409A/17/18/19/28/29 Interface
DP84532	80286 to DP8409A/17/18/19/28/29 Interface
DP8400-2	16-bit Expandable Error Checker/Corrector
DP8400-4	16-bit Expandable Error Checker/Corrector
DP8402A	32-bit Error Detector and Corrector (EDAC)

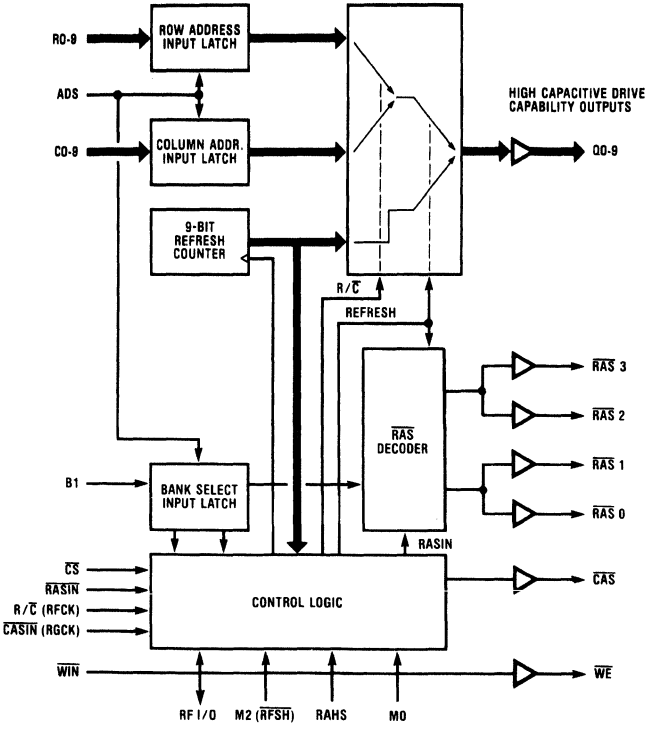
Block Diagrams

DP8417, 8419 and 8419X



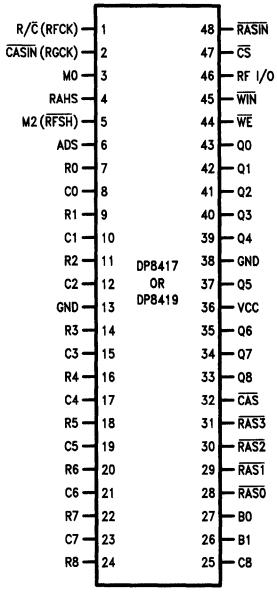
TL/F/8396-26

DP8418

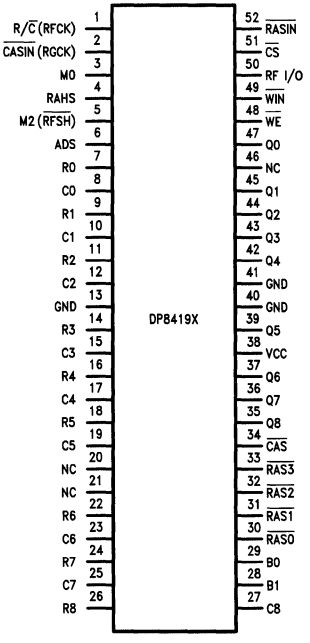


TL/F/8396-27

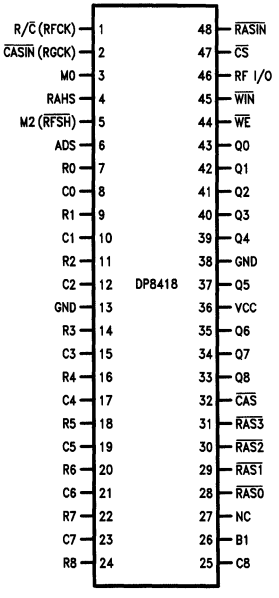
Connection Diagrams (Dual-In-Line Package)



TL/F/8396–28



TL/F/8396–29

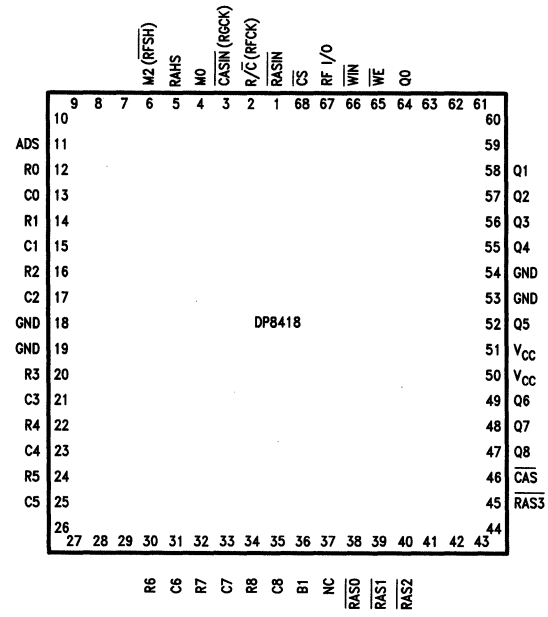


TL/F/8396–30

Order Number DP8417D-70, DP8417D-80, DP8417N-70, DP8417N-80, DP8418D-70, DP8418D-80, DP8418N-70, DP8418N-80, DP8419D-70, DP8419D-80, DP8419N-70, DP8419N-80, DP8419XD-70 or DP8419XD-80.
See NS Package Number D48A, D52A, or N48A

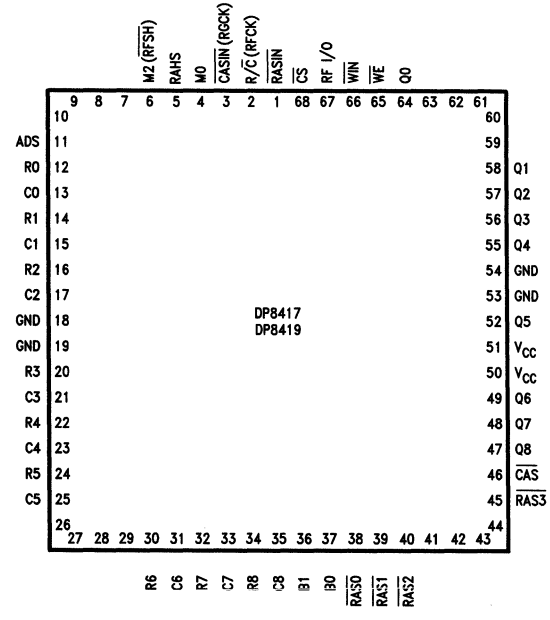
Connection Diagrams (Continued)

Plastic Chip Carrier Package



TL/F/8396–31

Plastic Chip Carrier Package



TL/F/8396–32

Order Number DP8417V-70, DP8417V-80, DP8418V-70, DP8418V-80, DP8419V-70 or DP8419V-80
See NS Package Number V68A

Family Device Differences

DP8417 vs DP8419

The DP8417 is identical to the DP8419 with the exception that its $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$ and Q (Multiplexed Address) outputs are TRI-STATE when $\overline{CS}$ (Chip Select) is high and the chip is not in a refresh mode. This feature allows access to the same DRAM array through multiple DRAM Controller/Driver DP8417s. All AC specifications are the same as the DP8419 except t_{CSRLO} which is 34 ns for the DP8417 versus 5 ns for the DP8419. Separate delay specifications for the TRI-STATE timing paths are provided in the AC tables of this data sheet.

DP8418 vs DP8419

The DP8418 DYNAMIC RAM CONTROLLER/DRIVER is identical to the DP8419 with the exception of two functional differences incorporated to improve performance with 32-bit microprocessors.

1) Pin 26 (B1) is used to enable/disable a pair of $\overline{RAS}$ outputs, and pin 27 (B0 on the DP8419) is a no connect. When B1 is low, $\overline{RAS}$0 and $\overline{RAS}$1 are enabled such that they both go low during an access. When B1 is high, $\overline{RAS}$2 and $\overline{RAS}$3 are enabled. This feature is useful when driving words to 32 bits or more since each $\overline{RAS}$ would be driving only one half of the word. By distributing the load on each $\overline{RAS}$ line in this way, the DP8418 will meet the same AC specifications driving 2 banks of 32 DRAMs each as the DP8419 does driving 4 banks of 16 bits each.
2) The hidden refresh function available on the DP8419 has been disabled in order to reduce the amount of setup time necessary from $\overline{CS}$ going low to $\overline{RASIN}$ going low during an access of DRAM. This parameter, called t_{CSRL1}, is 5 ns for the DP8418 whereas it is 34 ns for the DP8419. The hidden refresh function only allows a very small increase in system performance, at best, at microprocessor frequencies of 10 MHz and above.

DP8419 vs DP8409A

The DP8419 High Speed DRAM Controller/Driver combines the most popular memory control features of the DP8408A/9A DRAM Controller/Driver with the high speed of bipolar oxide isolation processing.

The DP8419 retains the high capacitive-load drive capability of the DP8408A/9A as well as its most frequently used access and refresh modes, allowing it to directly replace the DP8408A/9A in applications using only modes 0, 1, 4 and 5. Thus, the DP8419 will allow most DP8408A/9A users to directly upgrade their system by replacing their old controller chip with the DP8419.

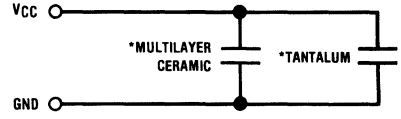
The highest priority of the DP8419 is speed. By peforming the DRAM address multiplexing, control signal timing and high-capacitive drive capability on a single chip, propagation delay skews are minimized. Emphasis has been placed on reducing delay variation over the specified supply and temperature ranges.

Except for the following, a DP8419 will operate essentially the same as a DP8409A.

1) The DP8419 has significantly faster AC performance.
2) The DP8419 can replace the DP8409A in applications which use modes 0, 1, 4, and 5. Modes 2, 3, 6, and 7 of the DP8409A are not available on the DP8419.
3) Pin 4 on the DP8419 is RAHS instead of M1, as on the DP8409A, and allows for two choices of t_{RAH} in mode 5.
4) RFI/O does not function as an end-of-count signal in Mode 0 on the DP8419 as it does on the DP8409A.
5) DP8419 address and control outputs do not TRI-STATE when $\overline{CS}$ is high as on the DP8409A. DP8419 control outputs are active high when $\overline{CS}$ is high (unless refreshing).

Pin Definitions

V_{CC}, GND, GND – V_{CC} = 5V ±10%. The three supply pins have been assigned to the center of the package to reduce voltage drops, both DC and AC. There are two ground pins to reduce the low level noise. The second ground pin is located two pins from V_{CC}, so that decoupling capacitors can be inserted directly next to these pins. It is important to adequately decouple this device, due to the high switching currents that will occur when all 9 address bits change in the same direction simultaneously. A recommended solution would be a 1 μF multilayer ceramic capacitor in parallel with a low-voltage tantalum capacitor, both connected as close as possible to V_{CC} and GND to reduce lead inductance. See Figure below.



TL/F/8396–4

*Capacitor values should be chosen depending on the particular application.

R0–R8: Row Address Inputs.

C0–C8: Column Address Inputs.

Q0–Q8: Multiplexed Address Outputs - This address is selected from the Row Address Input Latch, the Column Address Input Latch or the Refresh Counter.

$\overline{RASIN}$: Row Address Strobe Input - $\overline{RASIN}$ directly controls the selected $\overline{RAS}$ output when in an access mode and all $\overline{RAS}$ outputs during hidden or external refresh.

R/$\overline{C}$ (RFCK) - In the auto-modes this pin is the external refresh clock input; one refresh cycle should be performed each clock period. In the external access mode it is Row/Column Select Input which enables either the row or column address input latch onto the output bus.

$\overline{CASIN}$ (RGCK) - In the auto-modes this pin is the $\overline{RAS}$ Generator Clock input. In external access mode it is the Column Address Strobe input which controls $\overline{CAS}$ directly once columns are enabled on the address outputs.

ADS: Address (Latch) Strobe Input - Row Address, Column Address, and Bank Select Latches are fall-through with ADS high; latching occurs on high-to-low transition of ADS.

$\overline{CS}$: Chip Select Input - When high, $\overline{CS}$ disables all accesses. Refreshing, however, in both modes 0 and 1 is not affected by this pin.

M0, M2 ($\overline{RFSH}$): Mode Control Inputs - These pins select one of the four available operational modes of the DP8419 (see Table III).

RFI/O: Refresh Input/Output - In the auto-modes this pin is the Refresh Request Output. It goes low following RFCK

Pin Definitions (Continued)

indicating that no hidden refresh was performed while RFCK was high. When this pin is set low by an external gate the on-chip refresh counter is reset to all zeroes.

$\overline{WIN}$: Write Enable Input.

$\overline{WE}$: Write Enable Output - $\overline{WE}$ follows $\overline{WIN}$ unconditionally.

RAHS: Row Address Hold Time Select - Selects the t_{RAH} to be generated by the DP8419 delay line to allow use with fast or slow DRAMs.

$\overline{CAS}$: Column Address Strobe Output - In mode 5 and in mode 4 with $\overline{CASIN}$ low before R/$\overline{C}$ goes low, $\overline{CAS}$ goes low automatically after the column address is valid on the address outputs. In mode 4 $\overline{CAS}$ follows $\overline{CASIN}$ directly after R/$\overline{C}$ goes low, allowing for nibble accessing. $\overline{CAS}$ is always high during refresh.

$\overline{RAS}$ 0–3: Row Address Strobe Outputs - The enabled $\overline{RAS}$ output (see Table II) follows $\overline{RASIN}$ directly during an access. During refresh, all $\overline{RAS}$ outputs are enabled.

B0, B1: Bank Select Inputs - These pins are decoded to enable one of the four $\overline{RAS}$ outputs during an access (see Table I and Table II).

TABLE I. DP8417, DP8419, DP8419X Memory Bank Decode

Bank Select (Strobed by ADS)		Enabled $\overline{RAS}_n$
B1	B0	
0	0	$\overline{RAS}_0$
0	1	$\overline{RAS}_1$
1	0	$\overline{RAS}_2$
1	1	$\overline{RAS}_3$

TABLE II. DP8418 Memory Bank Decode

Bank Select (Strobed by ADS)		Enabled $\overline{RAS}_n$
B1	NC	
0	X	$\overline{RAS}_0$ and $\overline{RAS}_1$
1	X	$\overline{RAS}_2$ and $\overline{RAS}_3$

Conditions for All Modes

INPUT ADDRESSING

The address block consists of a row-address latch, a column-address latch, and a resettable refresh counter. The address latches are fall-through when ADS is high and latch when ADS goes low. If the address bus contains valid addresses until after $\overline{CAS}$ goes low at the end of the memory cycle, ADS can be permanently high. Otherwise ADS must go low while the addresses are still valid.

DRIVE CAPABILITY

The DP8419 has timing parameters that are specified driving the typical capacitance (including traces) of 88, 5V-only DRAMs. Since there are 4 $\overline{RAS}$ outputs, each is specified driving one-fourth of the total memory. $\overline{CAS}$, $\overline{WE}$ and the address outputs are specified driving all 88 DRAMs.

The graph in *Figure 10* may be used to determine the slight variations in timing parameters, due to loading conditions other than 88 DRAMs.

Because of distributed trace capacitance and inductance and DRAM input capacitance, current spikes can be created, causing overshoots and undershoots at the DRAM inputs that can change the contents of the DRAMs or even destroy them. To reduce these spikes, a damping resistor (low inductance, carbon) should be inserted between the DP8419 outputs and the DRAMs, as close as possible to the DP8419. The damping resistor values may differ depending on how heavily an output is loaded. These resistors should be determined by the first prototypes (not wire-wrapped due to the larger distributed capacitance and inductance). Resistors should be chosen such that the transition on the control outputs is critically damped. Typical values will be from 15Ω to 100Ω, with the lower values being used with the larger memory arrays. Note that AC parameters are specified with 15Ω damping resistors. For more information see AN-305 "Precautions to Take When Driving Memories".

DP8419 DRIVING ANY 16k, 64k or 256k DRAMs

The DP8419 can drive any 16k, 64k or 256k DRAMs. All 16k DRAMs use basically the same configuration, including the 5V-only version. Hence, in most applications, different manufacturers' DRAMs are interchangeable (for the same supply-rail chips), and the DP8419 can drive them all (see *Figure 1a*).

There are three basic configurations for the 5V-only 64k DRAMs: a 128-row by 512-column array with an on-RAM refresh counter, a 128-row by 512-column array with no on-RAM refresh counter, and a 256-row by 256-column array with no on-RAM refresh counter. The DP8419 can drive all three configurations, and allows them all to be interchangeable (as shown in *Figures 1b* and *1c*), providing maximum flexibility in the choice of DRAMs. Since the 9-bit on-chip refresh counter can be used as a 7-bit refresh counter for the 128-row configuration, or as an 8-bit refresh counter for the 256-row configuration, the on-RAM refresh counter, if present, is never used.

256k DRAMs require all 18 of the DP8419's address inputs to select one memory location within the DRAM. $\overline{RAS}$-only refreshing with the nine-bit refresh-counter on the DP8419 makes $\overline{CAS}$ before $\overline{RAS}$ refreshing, available on 256k DRAMs, unnecessary.

READ, WRITE AND READ-MODIFY-WRITE CYCLES

The output signal, $\overline{WE}$, determines what type of memory access cycle the memory will perform. If $\overline{WE}$ is kept high while $\overline{CAS}$ goes low, a read cycle occurs. If $\overline{WE}$ goes low before $\overline{CAS}$ goes low, a write cycle occurs and data at DI (DRAM input data) is written into the DRAM as $\overline{CAS}$ goes low. If $\overline{WE}$ goes low later than t_{CWD} after $\overline{CAS}$ goes low, first a read occurs and DO (DRAM output data) becomes valid, then data DI is written into the same address in the DRAM as $\overline{WE}$ goes low. In this read-modify-write case, DI and DO cannot be linked together. $\overline{WE}$ always follows $\overline{WIN}$ directly to determine the type of access to be performed.

POWER-UP INITIALIZE

When V_{CC} is first applied to the DP8419, an initialize pulse clears the refresh counter and the internal control flip-flops.

Mode Features Summary

- 4 modes of operation: 2 access and 2 refresh
- Automatic or external control selected by the user
- Auto access mode provides $\overline{RAS}$, row to column change, and then $\overline{CAS}$ automatically
- Choice between two different values of t_{RAH} in auto-access mode
- $\overline{CAS}$ controlled independently in external control mode, allowing for nibble mode accessing
- Automatic refreshing can make refreshes transparent to the system
- $\overline{CAS}$ is inhibited during refresh cycles

DP8419 Mode Descriptions

MODE 0–EXTERNALLY CONTROLLED REFRESH

Figure 2 shows the Externally Controlled Refresh timing. In this mode the refresh counter contents are multiplexed to the address outputs. All $\overline{RAS}$ outputs are enabled to follow $\overline{RASIN}$ so that the row address indicated by the refresh counter is refreshed in all DRAM banks when $\overline{RASIN}$ goes low. The refresh counter increments when $\overline{RASIN}$ goes high. $\overline{RFSH}$ should be held low at least until $\overline{RASIN}$ goes high (they may go high simultaneously) so that the refresh address remains valid and all $\overline{RAS}$ outputs remain enabled throughout the refresh.

A burst refresh may be performed by holding $\overline{RFSH}$ low and toggling $\overline{RASIN}$ until all rows are refreshed. It may be useful in this case to reset the refresh counter just prior to beginning the refresh. The refresh counter resets to all zeroes when RFI/O is pulled low by an external gate. The refresh counter always counts to 511 before rolling over to zero. If there are 128 or 256 rows being refreshed then Q7 or Q8, respectively, going high may be used as an end-of-burst indicator.

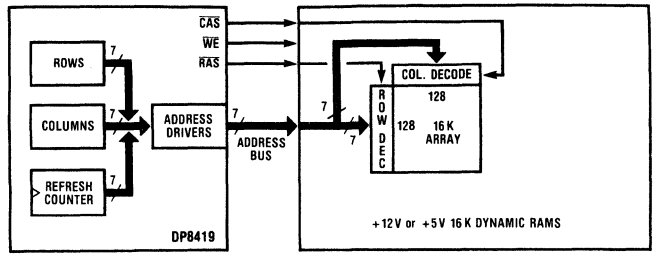
In order that the refresh address is valid on the address outputs prior to the $\overline{RAS}$ lines going low, $\overline{RFSH}$ must go low before $\overline{RASIN}$. The setup time required is given by t_{RFLRL} in the Switching Characteristics. This parameter may be adjusted using *Figure 10* for loading conditions other than those specified.

TABLE III. DP8419 Mode Select Options

Mode	($\overline{RFSH}$) M2	M0	Mode of Operation
0	0	0	Externally Controlled Refresh
1	0	1	Auto Refresh–Forced
4	1	0	Externally Controlled Access
5	1	1	Auto Access (Hidden Refresh)

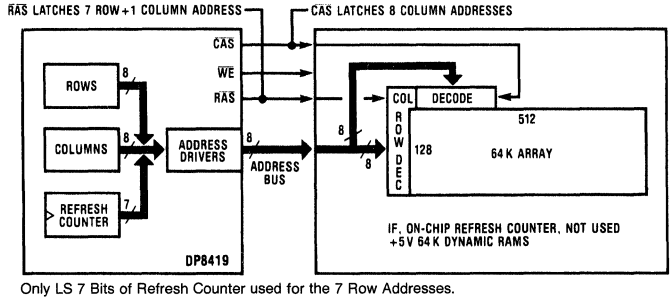
DP8419 Mode Descriptions (Continued)

DP8419 Interface Between System & DRAM Banks



TL/F/8396–5

FIGURE 1a. DP8419 with any 16k DRAMS

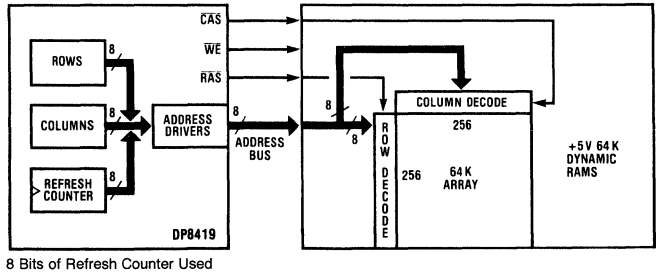


TL/F/8396–6

Only LS 7 Bits of Refresh Counter used for the 7 Row Addresses.

MSB not used but can toggle.

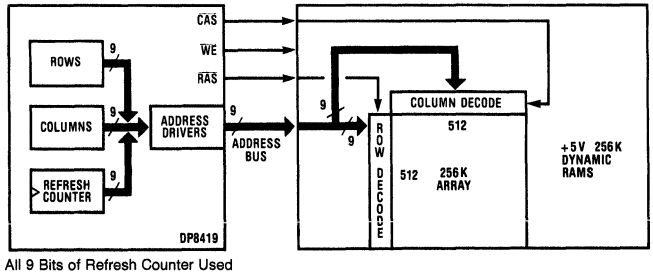
FIGURE 1b. DP8419 with 128 Row x 512 Column 64k DRAM



TL/F/8396–7

8 Bits of Refresh Counter Used

FIGURE 1c. DP8419 with 256 Row x 256 Column 64k DRAM

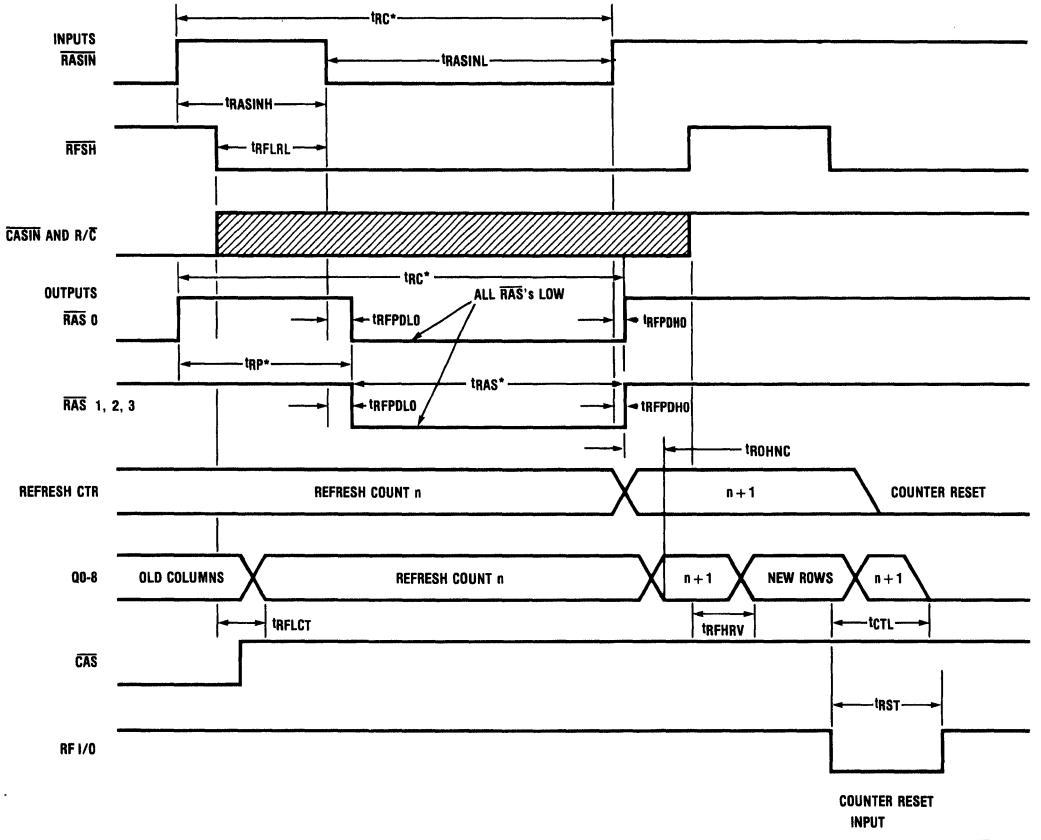


TL/F/8396–8

All 9 Bits of Refresh Counter Used

FIGURE 1d. DP8419 with 256k DRAMs

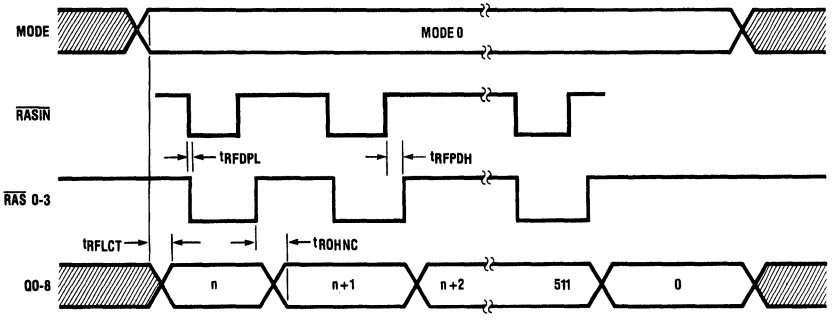
DP8419 Mode Descriptions (Continued)



TL/F/8396-9

*Indicates Dynamic RAM Parameters

FIGURE 2a. External Control Refresh Cycle (Mode 0)



TL/F/8396-10

FIGURE 2b. Burst Refresh Mode 0

DP8419 Mode Descriptions (Continued)

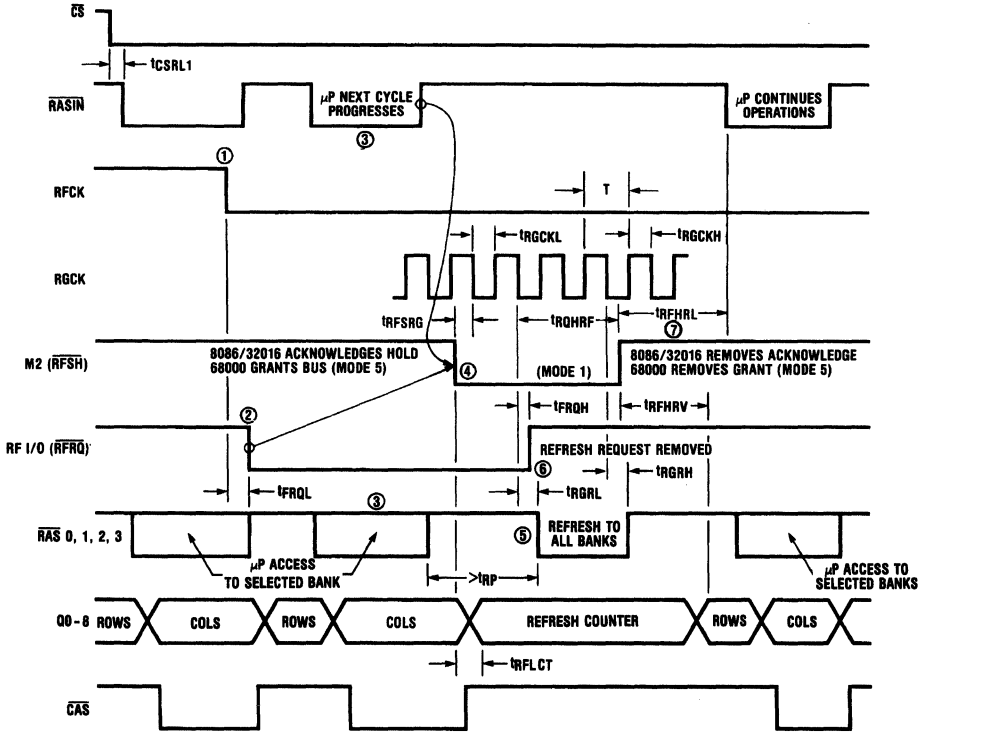
MODE 1–AUTOMATIC FORCED REFRESH

In Mode 1 the R/$\overline{C}$ (RFCK) pin becomes RFCK (refresh cycle clock) and the $\overline{CASIN}$ (RGCK) pin becomes RGCK ($\overline{RAS}$ generator clock). If RFCK is high and Mode 1 is entered then the chip operates as if in MODE 0 (externally controlled refresh), with all $\overline{RAS}$ outputs following $\overline{RASIN}$. This feature of Mode 1 may be useful for those who want to use Mode 5 (automatic access) with externally controlled refresh. By holding RFCK permanently high one need only toggle M2 ($\overline{RFSH}$) to switch from Mode 5 to external refresh. As with Mode 0, RFI/O may be pulled low by an external gate to reset the refresh counter.

When using Mode 1 as automatic refresh, RFCK must be an input clock signal. One refresh should occur each period of RFCK. If no refresh is performed while RFCK is high, then when RFCK goes low RFI/O immediately goes low to indicate that a refresh is requested. (RFI/O may still be used to reset the refresh counter even though it is also used as a refresh request pin, however, an open-collector gate should be used to reset the counter in this case since RFI/O is forced low internally for a request).

After receiving the refresh request the system must allow a forced refresh to take place while RFCK is low. External logic can monitor $\overline{RFRQ}$ (RFI/O) so that when $\overline{RFRQ}$ goes low this logic will wait for the access currently in progress to be completed before pulling M2 ($\overline{RFSH}$) low to put the DP8419 in mode 1. If no access is taking place when $\overline{RFRQ}$ occurs, then M2 may immediately go low. Once M2 is low, the refresh counter contents appear at the address outputs and $\overline{RAS}$ is generated to perform the refresh.

An external clock on RGCK is required to derive the refresh $\overline{RAS}$ signals. On the second falling edge of RGCK after M2 is low, all $\overline{RAS}$ lines go low. They remain low until two more falling edges of RGCK. Thus $\overline{RAS}$ remains high for one to two periods of RGCK after M2 goes low, and stays low for two periods. In order to obtain the minimum delay from M2 going low to $\overline{RAS}$ going low, M2 should go low t_{RFSRG} before the falling edge of RGCK.



TL/F/8396–11

① RFCK goes low
② $\overline{RFRQ}$ goes low if no hidden refresh occurred while RFCK was high
③ Next $\overline{RASIN}$ starts next access
④ μP acknowledges refresh request
⑤ Forced refresh $\overline{RAS}$ starts after > T (> t_{RP})
⑥ Forced refresh $\overline{RAS}$ ends $\overline{RFRQ}$
⑦ μP removes refresh acknowledge

FIGURE 3. DP8419 Performing a Forced Refresh (Mode 5 → 1 → 5) with Various Microprocessors

DP8419 Mode Descriptions (Continued)

The Refresh Request on RFI/O is terminated as $\overline{RAS}$ goes low. This signal may be used to end the refresh earlier than it normally would as described above. If M2 is pulled high while the $\overline{RAS}$ lines are low, then the $\overline{RAS}$s go high t_{RFRH} later. The designer must be careful, however, not to violate the minimum $\overline{RAS}$ low time of the DRAMs. He must also guarantee that the minimum $\overline{RAS}$ precharge time is not violated during a transition from mode 1 to mode 5 when an access is desired immediately following a refresh.

If the processor tries to access memory while the DP8419 is in mode 1, WAIT states should be inserted into the processor cycles until the DP8419 is back in mode 5 and the desired access has been accomplished (see *Figure 9*).

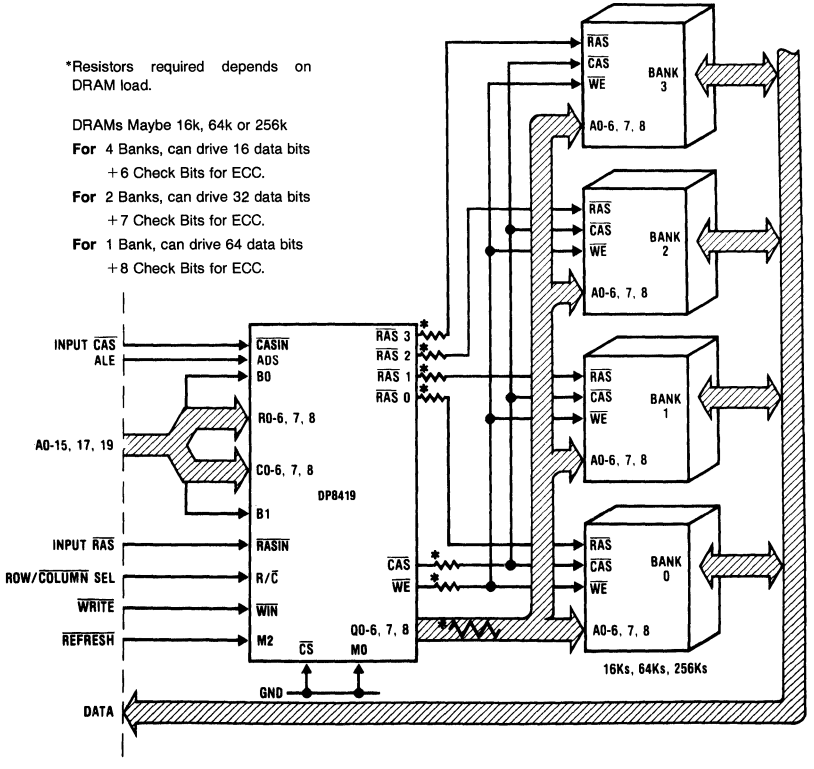
Instead of using WAIT states to delay accesses when refreshing, HOLD states could be used as follows. $\overline{RFRQ}$ could be connected to a HOLD or Bus Request input to the system. When convenient, the system acknowledges the $\overline{HOLD}$ or $\overline{Bus\ Request}$ by pulling M2 low. Using this scheme, $\overline{HOLD}$ will end as the $\overline{RAS}$ lines go low (RFI/O goes high). Thus, there must be sufficient delay from the time $\overline{HOLD}$ goes high to the DP8419 returning to mode 5, so that the $\overline{RAS}$ low time of the DRAMs isn't violated as described earlier (see *Figure 3* for mode 1 refresh with Hold states).

To perform a forced refresh the system will be inactive for about four periods of RGCK. For a frequency of 10 MHz, this is 400 ns. To refresh 128 rows every 2 ms an average of about one refresh per 16 μs is required. With a RFCK period of 16 μs and RGCK period of 100 ns, DRAM accesses are delayed due to refresh only 2.5% of the time. If using the Hidden Refresh available in mode 5 (refreshing with RFCK high) this percentage will be even lower.

MODE 4 - EXTERNALLY CONTROLLED ACCESS

In this mode all control signal outputs can be controlled directly by the corresponding control input. The enabled $\overline{RAS}$ output follows $\overline{RASIN}$, $\overline{CAS}$ follows $\overline{CASIN}$ (with R/$\overline{C}$ low), $\overline{WE}$ follows $\overline{WIN}$ and R/$\overline{C}$ determines whether the row or the column inputs are enabled to the address outputs (see *Figure 4*).

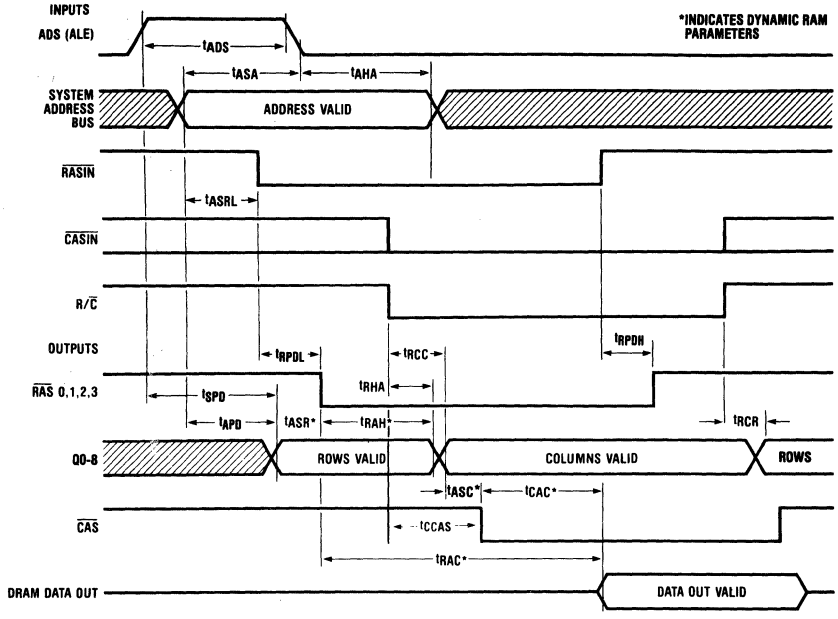
With R/$\overline{C}$ high, the row address latch contents are enabled onto the address bus. $\overline{RAS}$ going low strobes the row address into the DRAMs. After waiting to allow for sufficient row-address hold time (t_{RAH}) after $\overline{RAS}$ goes low, R/$\overline{C}$ can go low to enable the column address latch contents onto the address bus. When the column address is valid, $\overline{CAS}$ going low will strobe it into the DRAMs. $\overline{WIN}$ determines whether the cycle is a read, write or read-modify-write access. Refer to *Figures 5a* and *5b* for typical Read and Write timing using mode 4.



TL/F/8396–12

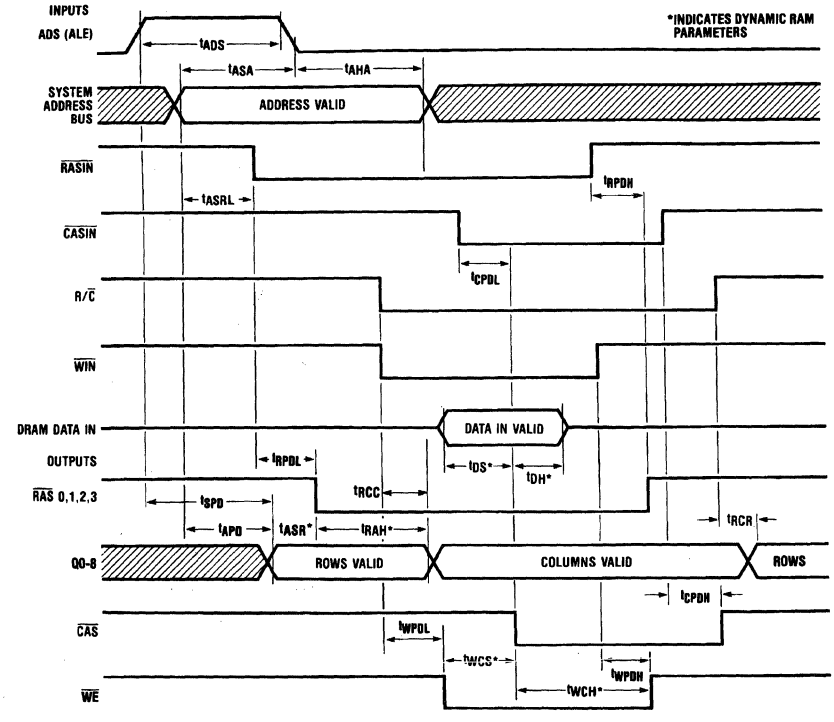
FIGURE 4. Typical Application of DP8419 Using External Control Access and Refresh in Modes 0 and 4

DP8419 Mode Descriptions (Continued)



TL/F/8396–13

FIGURE 5a. Read Cycle Timing (Mode 4)



TL/F/8396–14

FIGURE 5b. Write Cycle Timing (Mode 4)

DP8419 Mode Descriptions (Continued)

Page or Nibble mode may be performed by toggling $\overline{CASIN}$ once the initial access has been completed. In the case of page mode the column address must be changed before $\overline{CASIN}$ goes low to access a new memory location (see *Figure 5c*). Parameter t_{CPdif} has been specified in order that users may easily determine minimum $\overline{CAS}$ pulse widths when $\overline{CASIN}$ is toggling.

AUTOMATIC $\overline{CAS}$ GENERATION

$\overline{CAS}$ is held high when R/$\overline{C}$ is high even if $\overline{CASIN}$ is low. If $\overline{CASIN}$ is low when R/$\overline{C}$ goes low, $\overline{CAS}$ goes low automatically, t_{ASC} after the column address is valid. This feature eliminates the need for an externally derived $\overline{CASIN}$ signal to control $\overline{CAS}$ when performing a simple access (*Figure 5a* demonstrates Auto-$\overline{CAS}$ generation in mode 4). Page or nibble accessing may be performed as shown in *Figure 5c* even if $\overline{CAS}$ is generated automatically for the initial access.

FASTEST MEMORY ACCESS

The fastest mode 4 access is achieved by using the automatic $\overline{CAS}$ feature and external delay line to generate the required delay between $\overline{RASIN}$ and R/$\overline{C}$. The amount of delay required depends on the minimum t_{RAH} of the DRAMs being used. The DP8419 parameter t_{DIF1} has been specified in order that the delay between $\overline{RASIN}$ and R/$\overline{C}$ may be minimized.

$$t_{DIF1} = \text{MAXIMUM } (t_{RPDL} - t_{RHA})$$

where t_{RPDL} = $\overline{RASIN}$ to $\overline{RAS}$ delay

and t_{RHA} = row address held from R/$\overline{C}$ going low.

The delay between $\overline{RASIN}$ and R/$\overline{C}$ that guarantees the specified DRAM t_{RAH} is given by

$$\text{MINIMUM } \overline{RASIN} \text{ to R/}\overline{C} = t_{DIF1} + t_{RAH}.$$

Example

In an application using DRAMs that require a minimum t_{RAH} of 15 ns, the following demonstrates how the maximum $\overline{RASIN}$ to $\overline{CAS}$ time is determined.

With t_{DIF1} (from Switching Characteristics) = 7 ns,

$\overline{RASIN}$ to R/$\overline{C}$ delay = 7 ns + 15 ns = 22 ns.

A delay line of 25 ns will be sufficient.

With Auto-$\overline{CAS}$ generation, the maximum delay from R/$\overline{C}$ to $\overline{CAS}$ (loaded with 600 pF) is 46 ns. Thus the maximum $\overline{RASIN}$ to $\overline{CAS}$ time is 71 ns, under the given conditions.

With a maximum $\overline{RASIN}$ to $\overline{RAS}$ time (t_{RPDL}) of 20 ns, the maximum $\overline{RAS}$ to $\overline{CAS}$ time is about 51 ns. Most DRAMs with a 15 ns minimum t_{RAH} have a maximum t_{RCD} of about 60 ns. Thus, memory accesses are likely to be $\overline{RAS}$ limited instead of $\overline{CAS}$ limited. In other words, memory access time is limited by DRAM performance, not controller performance.

REFRESHING IN CONJUNCTION WITH MODE 4

If using mode 4 to access memory, mode 0 (externally controlled refresh) must be used for all refreshing.

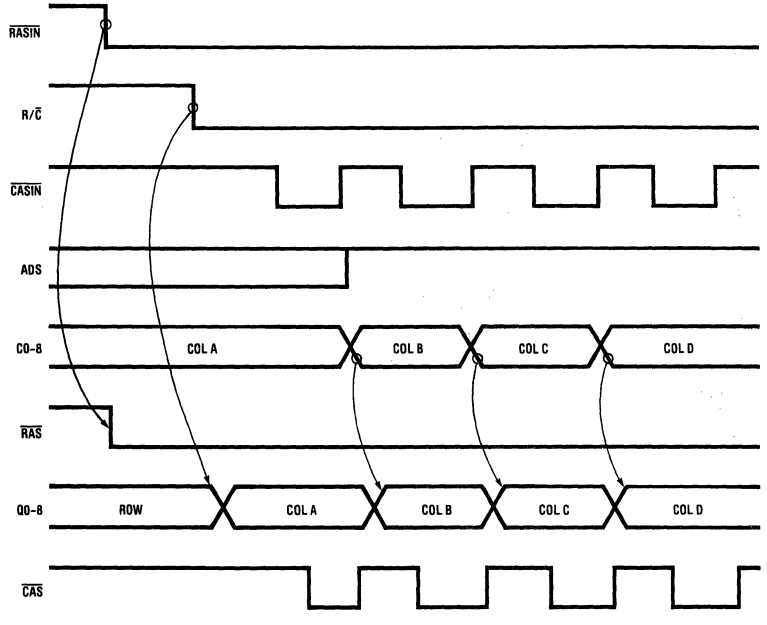
MODE 5 – AUTOMATIC ACCESS WITH HIDDEN REFRESHING CAPABILITY

Automatic-Access has two advantages over the externally controlled access (mode 4). First, $\overline{RAS}$, $\overline{CAS}$ and the row to column change are all derived internally from one input signal, $\overline{RASIN}$. Thus the need for an external delay line (see mode 4) is eliminated.

Secondly, since R/$\overline{C}$ and $\overline{CASIN}$ are not needed to generate the row to column change and $\overline{CAS}$, these pins can be used for the automatic refreshing function.

AUTOMATIC ACCESS CONTROL

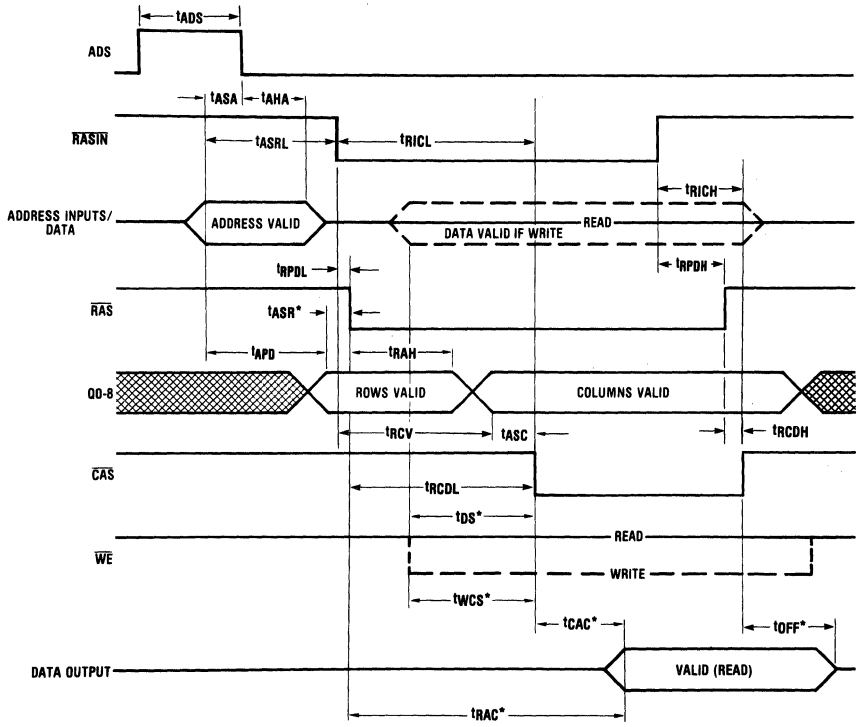
Mode 5 of the DP8419 makes accessing Dynamic RAM nearly as easy as accessing static RAM. Once row and column addresses are valid (latched on the DP8419 if necessary), $\overline{RASIN}$ going low is all that is required to perform the memory access.



TL/F/8396–15

FIGURE 5c. Page or Nibble Access in Mode 4

DP8419 Mode Descriptions (Continued)



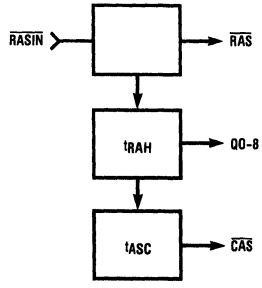
*Indicates Dynamic RAM Parameters

TL/F/8396–17

FIGURE 6. Mode 5 Timing

(Refer to *Figure 6*) In mode 5 the selected $\overline{RAS}$ follows $\overline{RASIN}$ immediately, as in mode 4, to strobe the row address into the DRAMs. The row address remains valid on the DP8419 address outputs long enough to meet the t_{RAH} requirement of the DRAMs (pin 4, RAHS, of the DP8419 allows the user two choices of t_{RAH}). Next, the column address replaces the row address on the address outputs and $\overline{CAS}$ goes low to strobe the columns into the DRAMs. $\overline{WIN}$ determines whether a read, write or read-modify-write is done.

The diagram below illustrates mode 5 automatic control signal generation.



TL/F/8396–16

REFRESHING IN CONJUNCTION WITH MODE 5

When using mode 5 to perform memory accesses, refreshing may be accomplished:

(a) externally (in mode 0 or mode 1)

(b) by a combination of mode 5 (hidden refresh) and mode 1 (auto-refresh)

or (c) by a combination of mode 5 and mode 0

(a) Externally Controlled Refreshing in Mode 0 or Mode 1

All refreshing may be accomplished using external refreshes in either mode 0 or mode 1 with R/$\overline{C}$ (RFCK) tied high (see mode 0 and mode 1 descriptions). If this is desired, the system determines when a refresh will be performed, puts the DP8419 in the appropriate mode, and controls the $\overline{RAS}$ signals directly with $\overline{RASIN}$. The on-chip refresh counter is enabled to the address outputs of the DP8419 when the refresh mode is entered, and increments when $\overline{RASIN}$ goes high at the completion of the refresh.

(b) Mode 5 Refreshing (hidden) with Mode 1 refreshing (auto)

(Refer to *Figure 7a*) If RFCK is tied to a clock (see mode 1 description), RFI/O becomes a refresh request output and goes low following RFCK going low if no refresh occurred while RFCK was high. Refreshes may be performed in mode 5 when the DP8419 is not selected for access ($\overline{CS}$ is high) and RFCK is high. If these conditions exist the refresh counter contents appear on the DP8419 address outputs and all $\overline{RAS}$ lines follow $\overline{RASIN}$ so that if $\overline{RASIN}$ goes low (an access other than through the DP8419 occurs), all $\overline{RAS}$ lines go low to perform the refresh. The DP8419 allows only one refresh of this type for each period of RFCK, since RFCK should be fast enough such that one refresh per period is sufficient to meet the DRAM refresh requirement.

DP8419 Mode Descriptions (Continued)

Once it is started, a hidden refresh will continue even if RFCK goes low. However, $\overline{CS}$ must be high throughout the refresh (until $\overline{RASIN}$ goes high).

These hidden refreshes are valuable in that they do not delay accesses. When determining the duty cycle of RFCK, the high time should be maximized in order to maximize the probability of hidden refreshes. If a hidden refresh doesn't happen, then a refresh request will occur on RFI/O when RFCK goes low. After receiving the request, the system must perform a refresh while RFCK is low. This may be done by going to mode 1 and allowing an automatic refresh (see mode 1 description). This refresh must be completed while RFCK is low, thus the RFCK low time is determined by the worst-case time required by the system to respond to a refresh request.

(c) Mode 5 Refresh (Hidden Refresh) with mode 0 Refresh (External Refresh)

This refresh scheme is identical to that in (b) except that after receiving a refresh request, mode 0 is entered to do the refresh (see mode 0 description). The refresh request is terminated (RFI/O goes high) as soon as mode 0 is entered. This method requires more control than using mode 1 (auto-refresh), however, it may be desirable if the mode 1 refresh time is considered to be excessive.

Example

Figure 7b demonstrates how a system designer would use the DP8419 in mode 5 based on certain characteristics of his system.

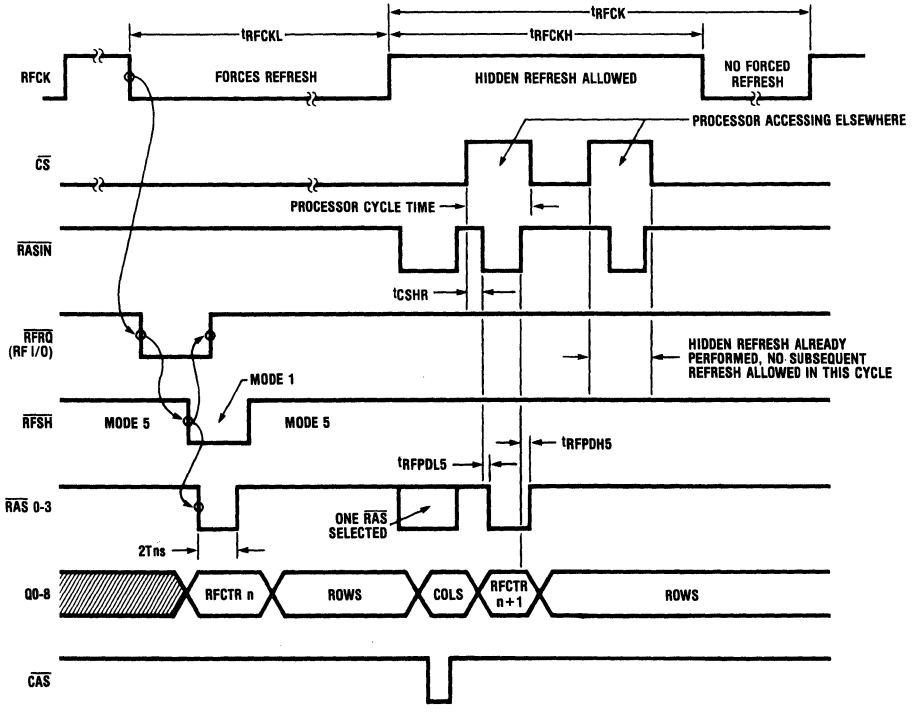
System Characteristics:

1) DRAM used has min t_{RAH} requirement of 15 ns and min t_{ASR} of 0 ns
2) DRAM address is valid from time T_V to the end of the memory cycle
3) four banks of twenty-two 256K memory chips each are being driven

Using the DP8419 (see *Figure 7b*):

1) Tie pin 4 (RAHS) high to guarantee a 15 ns minimum t_{RAH} which is sufficient for the DRAMs being used
2) Generate $\overline{RASIN}$ no earlier than time $T_V + t_{ASRL}$ (see switching characteristics), so that the row address is valid on the DRAM address inputs before $\overline{RAS}$ occurs
3) Tie ADS high since latching the DRAM address on the DP8419 is not necessary
4) Connect the first 18 system address bits to R0-R8 and C0-C8, and bits 19 and 20 to B0 and B1
5) Connect each $\overline{RAS}$ output of the DP8419 to the $\overline{RAS}$ inputs of the DRAMs of one bank of the memory array; connect Q0-Q8 of the DP8419 to A0-A8 of all DRAMs; connect $\overline{CAS}$ of the DP8419 to $\overline{CAS}$ of all the DRAMs

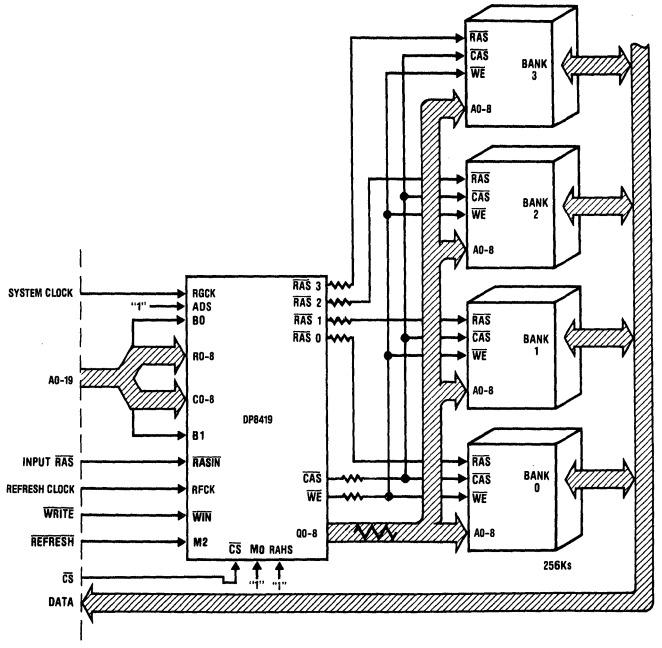
Figure 7c illustrates a similar example using the DP8418 to drive two 32-bit banks.



TL/F/8396–18

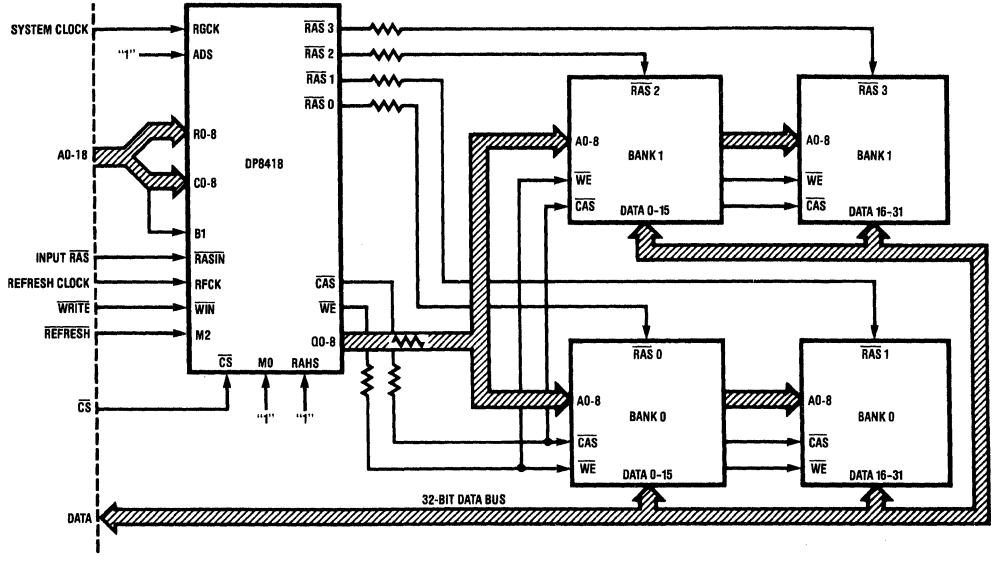
FIGURE 7a. Hidden Refreshing (Mode 5) and Forced Refreshing (Mode 1) Timing

DP8419 Mode Descriptions (Continued)



TL/F/8396–19

FIGURE 7b. Typical Application of DP8419 Using Modes 5 and 1



TL/F/8396–33

FIGURE 7c. Typical Application of DP8418 Using Modes 5 and 1

Applications

If one desires a memory interface containing the DP8419 that minimizes the number of external components required, modes 5 and 1 should be used. These two modes provide:

1) Automatic access to memory (in mode 5 only one signal, $\overline{RASIN}$, is required in order to access memory)
2) Hidden refresh capability (refreshes are performed automatically while in mode 5 when non-local accesses are taking place, as determined by $\overline{CS}$)
3) Refresh request capability (if no hidden refresh took place while RFCK was high, a refresh request is generated at the RFI/O pin when RFCK goes high)
4) Automatic forced refresh (If a refresh request is generated while in mode 5, as described above, external logic should switch the DP8419 into mode 1 to do an automatic forced refresh. No other external control signals need be issued. WAIT states can be inserted into the processor machine cycles if the system tries to access memory while the DP8419 is in mode 1 doing a forced refresh).

Some items to be considered when integrating the DP8419 into a system design are:

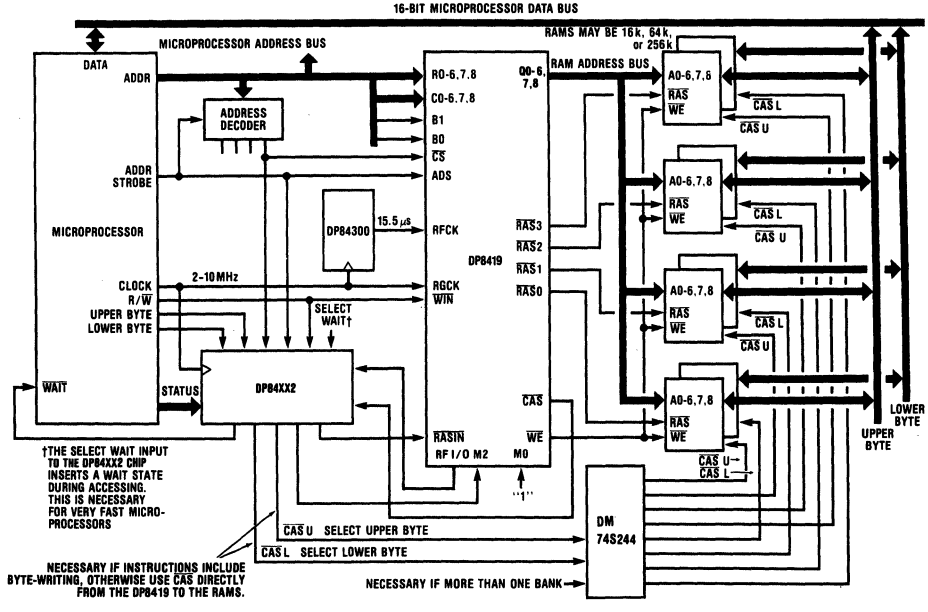
1) The system designer should ensure that a DRAM access not be in progress when a refresh mode is entered. Similarly, one should not attempt to start an access while a refresh is in progress. The parameter t_{RFHRL} specifies the minimum time from $\overline{RFSH}$ high to $\overline{RASIN}$ going low to initiate an access.
2) One should always guarantee that the DP8419 is enabled for access prior to initiating the access (see t_{CSRL1}).
3) One should bring $\overline{RASIN}$ low even during non-local access cycles when in mode 5 in order to maximize the chance of a hidden refresh occurring.
4) At lower frequencies (under 10 Mhz), it becomes increasingly important to differentiate between READ and WRITE cycles. $\overline{RASIN}$ generation during READ cycles can take place as soon as one knows that a processor READ access cycle has started. WRITE cycles, on the other hand, cannot start until one knows that the data to be written at the DRAM inputs will be valid a setup time before $\overline{CAS}$ (column address strobe) goes true at the DRAM inputs. Therefore, in general, READ cycles can be initiated earlier than WRITE cycles.
5) Many times it is possible to only add WAIT states during READ cycles and have no WAIT states during WRITE cycles. This is because it generally takes less time to write data into memory than to read data from memory.

The DP84XX2 family of inexpensive preprogrammed medium Programmable Array Logic devices (PALs) have been developed to provide an easy interface between various microprocessors and the DP84XX family of DRAM controller/drivers. These PALs interface to all the necessary control signals of the particular processor and the DP8419. The PAL controls the operation of the DP8419 in modes 5 and 1, while meeting all the critical timing considerations discussed above. The refresh clock, RFCK, may be divided down from the processor clock using an IC counter such as the DM74LS393 or the DP84300 programmable refresh timer. The DP84300 can provide RFCK periods ranging from 15.4 μs to 15.6 μs based on an input clock of 2 to 10 MHz. *Figure 8* shows a general block diagram for a system using the DP8419 in modes 1 and 5. *Figure 9* shows possible timing diagrams for such a system (using $\overline{WAIT}$ to prohibit access when refreshing). Although the DP84XX2 PALs are offered as standard peripheral devices for the DP84XX DRAM controller/drivers, the programming equations for these devices are provided so the user may make minor modification, for unique system requirements.

ADVANTAGES OF DP8419 OVER A DISCRETE DYNAMIC RAM CONTROLLER

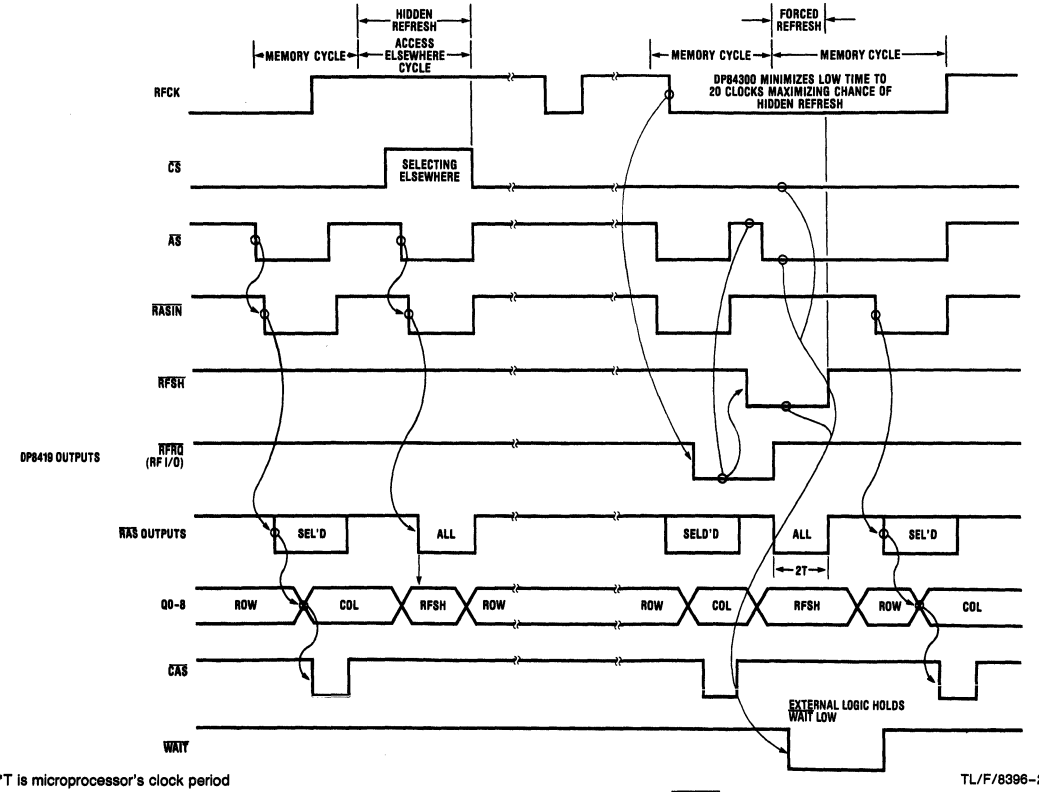
1) The DP8419 system solution takes up much less board space because everything is on one chip (latches, refresh counter, control logic, multiplexers, drivers, and internal delay lines).
2) Less effort is needed to design a memory system. The DP8419 has automatic modes (1 and 5) which require a minimum of external control logic. Also programmable array logic devices (PALs) have been designed which allow an easy interface to most popular microprocessors (Motorola 68000 family, National Semiconductor 32032 family, Intel 8086 family, and the Zilog Z8000 family).
3) Less skew in memory timing parameters because all critical components are on one chip (many discrete drivers specify a minimum on-chip skew under worst-case conditions, but this cannot be used if more then one driver is needed, such as would be the case in driving a large dynamic RAM array).
4) Our switching characteristics give the designer the critical timing specifications based on TTL output levels (low = 0.8V, high = 2.4V) at a specified load capacitance. All timing parameters are specified on the DP8419:
   A) driving 88 DRAM's over a temperature range of 0–70 degrees centigrade (no extra drivers are needed).
   B) under worst-case driving conditions with all outputs switching simultaneously (most discrete drivers on the market specify worst-case conditions with only one output switching at a time; this is not a true worst-case condition!).

Applications (Continued)



TL/F/8396–20

FIGURE 8. Connecting the DP8419 Between the 16-bit Microprocessor and Memory



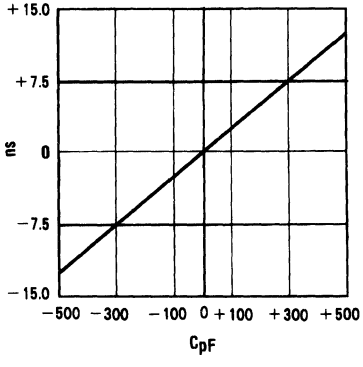
*T is microprocessor's clock period

TL/F/8396–21

FIGURE 9. DP8419 Auto Refresh, Access with $\overline{\text{WAIT}}$ States

Switching Characteristics

All AC parameters are specified with the equivalent load capacitances, including traces, of 88 DRAMs organized as 4 banks of 22 DRAMs each. Maximums are based on worst-case conditions including all outputs switching simultaneously. This, in many cases, results in the AC values shown in the DP84XX DRAM controller data sheet being much looser than true worst case (maximum) AC delays. The system designer should estimate the DP8419 load in his/her application, and modify the appropriate AC parameters using the graph in *Figure 10*. Two example calculations are provided below.



TL/F/8396–22

FIGURE 10. Change in Propagation Delay Relative to "True" (Application) Load Minus AC Specified Data Sheet Load

2 Examples

#1) A mode 4 user driving 2 16-bit banks of DRAM has the following approximate "true" loading conditions:

$\overline{CAS}$ - 300 pF

Q0-Q8 - 250 pF

$\overline{RAS}$ - 150 pF

max t_{RPDL} = 20 ns − 0 ns = 20 ns (since $\overline{RAS}$ loading is the same as that which is spec'ed)

max t_{CPDL} = 32 ns − 7 ns = 25 ns

max t_{CCAS} = 46 ns − 7 ns = 39 ns

max t_{RCC} = 41 ns − 6 ns = 35 ns

min t_{RHA} is not significantly effected since it does not involve an output transition

Other parameters are adjusted in a similar manner.

#2) A mode 5 user driving one 16-bit bank of DRAM has the following approximate "true" loading conditions:

$\overline{CAS}$ - 120 pF

Q0-Q8 - 100 pF

$\overline{RAS}$ - 120 pF

A. C. parameters should be adjusted as follows:

with RAHS = "1",

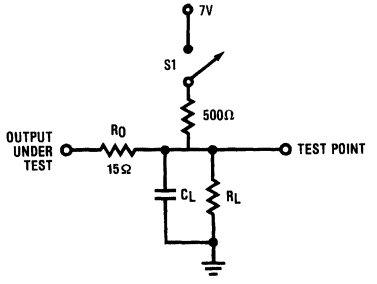
max t_{RICL} = 70 ns − 11 ns = 59 ns

max t_{RCDL} = 55 ns + 1 ns − 11 ns = 45 ns

(the + 1 ns is due to lighter $\overline{RAS}$ loading; the − 11 ns is due to lighter $\overline{CAS}$ loading)

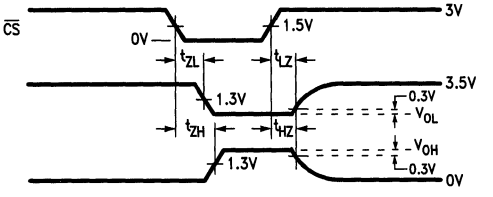
min t_{RAH} = 15 ns + 1 ns = 16 ns

The additional 1 ns is due to the fact that the $\overline{RAS}$ line is driving less (switching faster) than the load to which the 15 ns spec applies. The row address will remain valid for about the same time irregardless of address loading since it is considered to be not valid at the beginning of its transition.



TL/F/8396–23

FIGURE 11a. Output Load Circuit



TL/F/8396–34

FIGURE 11b. DP8417 TRI-STATE Waveforms

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply voltage, V_{CC}	7.0V
Storage Temperature Range	−65°C to +150°C
Input Voltage	5.5V
Output Current	150 mA
Lead Temp. (Soldering, 10 seconds)	300°C

Operating Conditions

		Min	Max	Units
V_{CC}	Supply Voltage	4.50	5.50	V
T_A	Ambient Temperature	0	+70	°C

Electrical Characteristics V_{CC} = 5.0V ±10%, 0°C ≤ T_A ≤ 70°C unless otherwise noted (Note 2)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_C	Input Clamp Voltage	V_{CC} = Min, I_C = − 12 mA		− 0.8	− 1.2	V
I_{IH}	Input High Current for all Inputs	V_{IN} = 2.5V		2.0	100	μA
I_I RSI	Output Load Current for RFI/O	V_{IN} = 0.5V, Output high		−0.7	−1.5	mA
I_{IL1}	Input Low Current for all Inputs**	V_{IN} = 0.5V		−0.02	−0.25	mA
I_{IL2}	ADS, R/$\overline{C}$, $\overline{CS}$, M2, $\overline{RASIN}$	V_{IN} = 0.5V		−0.05	−0.5	mA
V_{IL}	Input Low Threshold				0.8	V
V_{IH}	Input High Threshold		2.0			V
V_{OL1}	Output Low Voltage*	I_{OL} = 20 mA		0.3	0.5	V
V_{OL2}	Output Low Voltage for RFI/O	I_{OL} = 8 mA		0.3	0.5	V
V_{OH1}	Output High Voltage*	I_{OH} = − 1 mA	2.4	3.5		V
V_{OH2}	Output High Voltage for RFI/O	I_{OH} = − 100 μA	2.4	3.5		V
I_{1D}	Output High Drive Current*	V_{OUT} = 0.8V (Note 3)	−50	− 200		mA
I_{0D}	Output Low Drive Current*	V_{OUT} = 2.4V (Note 3)	50	200		mA
I_{CC}	Supply Current	V_{CC} = Max		150	240	mA

*Except RFI/O

**Except RFI/O, ADS, R/$\overline{C}$, $\overline{CS}$, M2, $\overline{RASIN}$

Switching Characteristics: DP8417, DP8418, DP8419, DP8419X

V_{CC} = 5.0V ±10%, 0°C ≤ T_A ≤ 70°C unless otherwise noted (Notes 2, 4, 5), the output load capacitance is typical for 4 banks of 22 DRAMs each or 88 DRAMs, including trace capacitance.

*** These values are Q0-Q8, C_L = 500 pF; $\overline{RAS0}$–$\overline{RAS3}$, C_L = 150 pF; $\overline{WE}$, C_L = 500 pF; $\overline{CAS}$, C_L = 600 pF; RL = 500Ω unless otherwise noted. See *Figure 11a* for test load. S1 is open unless otherwise noted. Maximum propagation delays are specified with all outputs switching.**

**** Preliminary**

Symbol	Parameter	Condition	*CL		**All C_L = 50 pF		Units
			Min	Max	Min	Max	
ACCESS							
t_{RICL0}	$\overline{RASIN}$ to $\overline{CAS}$ Low Delay (RAHS = 0)	*Figure 6* DP8417, 18, 19-80	57	97	42	85	ns
t_{RICL0}	$\overline{RASIN}$ to $\overline{CAS}$ Low Delay (RAHS = 0)	*Figure 6* DP8417, 18, 19-70	57	87	42	75	ns
t_{RICL1}	$\overline{RASIN}$ to $\overline{CAS}$ Low Delay (RAHS = 1)	*Figure 6* DP8417, 18, 19-80	48	80	35	68	ns
t_{RICL1}	$\overline{RASIN}$ to $\overline{CAS}$ Low Delay (RAHS = 1)	*Figure 6* DP8417, 18, 19-70	48	70	35	58	ns
t_{RICH}	$\overline{RASIN}$ to $\overline{CAS}$ High Delay	*Figure 6*		37			ns
t_{RCDL0}	$\overline{RAS}$ to $\overline{CAS}$ Low Delay (RAHS = 0)	*Figure 6* DP8417, 18, 19-80	43	80			ns
t_{RCDL0}	$\overline{RAS}$ to $\overline{CAS}$ Low Delay (RAHS = 0)	*Figure 6* DP8417, 18, 19-70	43	72			ns
t_{RCDL1}	$\overline{RAS}$ to $\overline{CAS}$ Low Delay (RAHS = 1)	*Figure 6* DP8417, 18, 19-80	34	63			ns
t_{RCDL1}	$\overline{RAS}$ to $\overline{CAS}$ Low Delay (RAHS = 1)	*Figure 6* DP8417, 18, 19-70	34	55			ns
t_{RCDH}	$\overline{RAS}$ to $\overline{CAS}$ High Delay	*Figure 6*		22			ns
t_{RAH0}	Row Address Hold Time (RAHS = 0, Mode 5)	*Figure 6*	25		25		ns
t_{RAH1}	Row Address Hold Time (RAHS = 1, Mode 5)	*Figure 6*	15		15		ns
t_{ASC}	Column Address Set-up Time (Mode 5)	*Figure 6*	0		0		ns

Switching Characteristics: DP8417, DP8418, DP8419, DP8419X (Continued)

V_{CC} = 5.0V ±10%, 0°C ≤ T_A ≤ 70°C unless otherwise noted (Notes 2, 4, 5). The output load capacitance is typical for 4 banks of 22 DRAMs each or 88 DRAMs, including trace capacitance.

*** These values are Q0-Q8, C_L = 500 pF; $\overline{RAS}$0–$\overline{RAS}$3, C_L = 150 pF; $\overline{WE}$, C_L = 500 pF; $\overline{CAS}$, C_L = 600 pF; RL = 500Ω unless otherwise noted. See *Figure 11a* for test load. S1 is open unless otherwise specified. Maximum propagation delays are specified with all outputs switching.**

**** Preliminary**

Symbol	Parameter	Condition	*CL		**All C_L = 50 pF		Units
			Min	Max	Min	Max	
ACCESS (Continued)							
t_{RCV0}	$\overline{RASIN}$ to Column Address Valid (RAHS = 0, Mode 5)	*Figure 6* DP8417, 18, 19-80		94			ns
t_{RCV0}	$\overline{RASIN}$ to Column Address Valid (RAHS = 0, Mode 5)	*Figure 6* DP8417, 18, 19-70		85			ns
t_{RCV1}	$\overline{RASIN}$ to Column Address Valid (RAHS = 1, Mode 5)	*Figure 6* DP8417, 18, 19-80		76			ns
t_{RCV1}	$\overline{RASIN}$ to Column Address Valid (RAHS = 1, Mode 5)	*Figure 6* DP8417, 18, 19-70		68			ns
t_{RPDL}	$\overline{RASIN}$ to $\overline{RAS}$ Low Delay	*Figures 5a, 5b, 6*		21		18	ns
t_{RPDH}	$\overline{RASIN}$ to $\overline{RAS}$ High Delay	*Figures 5a, 5b, 6*		20		17	ns
t_{ASRL}	Address Set-up to $\overline{RASIN}$ low	*Figures 5a, 5b, 6*	13				ns
t_{APD}	Address Input to Output Delay	*Figures 5a, 5b, 6*		36		25	ns
t_{SPD}	Address Strobe High to Address Output Valid	*Figures 5a, 5b*		48			ns
t_{ASA}	Address Set-up Time to ADS	*Figures 5a, 5b, 6*	5				ns
t_{AHA}	Address Hold Time from ADS	*Figures 5a, 5b, 6*	10				ns
t_{ADS}	Address Strobe Pulse Width	*Figures 5a, 5b, 6*	26				ns
t_{WPD}	$\overline{WIN}$ to $\overline{WE}$ Output Delay	*Figure 5b*		28			ns
t_{CPDL}	$\overline{CASIN}$ to $\overline{CAS}$ Low Delay (R/$\overline{C}$ low, Mode 4)	*Figure 5b*	17	33			ns
t_{CPDH}	$\overline{CASIN}$ to $\overline{CAS}$ High Delay (R/$\overline{C}$ low, Mode 4)	*Figure 5b*	13	33			ns
t_{CPdif}	t_{CPDL} - t_{CPDH}	See Mode 4 Description		13			ns
t_{RCC}	Column Select to Column Address Valid	*Figure 5a*		41			ns
t_{RCR}	Row Select to Row Address Valid	*Figures 5a, 5b*		45			ns
t_{RHA}	Row Address Held from Column Select	*Figure 5a*	7				ns
t_{CCAS}	R/$\overline{C}$ Low to $\overline{CAS}$ Low Delay ($\overline{CASIN}$ Low, Mode 4)	*Figure 5a* DP8417, 18, 19-80		50			ns
t	R/$\overline{C}$ Low to $\overline{CAS}$ Low Delay ($\overline{CASIN}$ Low, Mode 4)	*Figure 5a* DP8417, 18, 19-70		46			ns
t_{DIF1}	Maximum (t_{RPDL} - t_{RHA})	See Mode 4 Description		7			ns
t_{DIF2}	Maximum (t_{RCC} - t_{CPDL})			13			ns
REFRESH							
t_{RC}	Refresh Cycle Period	*Figure 2a*	100				ns
$t_{RASINL,H}$	Pulse Width of $\overline{RASIN}$ during Refresh	*Figure 2a*	50				ns
t_{RFPDL0}	$\overline{RASIN}$ to $\overline{RAS}$ Low Delay during Refresh (Mode 0)	*Figure 2a*		28			ns

Switching Characteristics: DP8417, DP8418, DP8419, DP8419X (Continued)

V_{CC} = 5.0V ±10%, 0°C ≤ T_A ≤ 70°C unless otherwise noted (Notes 2, 4, 5). The output load capacitance is typical for 4 banks of 22 DRAMs each or 88 DRAMs, including trace capacitance.

*** These values are Q0-Q8, C_L = 500 pF; $\overline{RAS}$0-$\overline{RAS}$3, C_L = 150 pF; $\overline{WE}$, C_L = 500 pF; $\overline{CAS}$, C_L = 600 pF; RL = 500Ω unless otherwise noted. See *Figure 11a* for test load. S1 is open unless otherwise specified. Maximum propagation delays are specified with all outputs switching.**

Symbol	Parameter	Condition	*CL		All C_L = 50 pF		Units
			Min	Max	Min	Max	
REFRESH (Continued)							
t_{RFPDL5}	$\overline{RASIN}$ to $\overline{RAS}$ Low Delay during Hidden Refresh	*Figure 7*		38			ns
t_{RFPDH0}	$\overline{RASIN}$ to $\overline{RAS}$ High Delay during Refresh (Mode 0)	*Figure 2a*		35			ns
t_{RFPDH5}	$\overline{RASIN}$ to $\overline{RAS}$ High Delay during Hidden Refresh	*Figure 7*		44			ns
t_{RFLCT}	$\overline{RFSH}$ Low to Counter Address Valid	*Figures 2a, 3* $\overline{CS}$ = X		38			ns
t_{RFLRL}	$\overline{RFSH}$ Low Set-up to $\overline{RASIN}$ Low (Mode 0), to get Minimum t_{ASR} = 0	*Figure 2a*	12				ns
t_{RFHRL}	$\overline{RFSH}$ High Setup to Access $\overline{RASIN}$ Low	*Figure 3*	25				ns
t_{RFHRV}	$\overline{RFSH}$ High to Row Address Valid	*Figure 3*		43			ns
t_{ROHNC}	$\overline{RAS}$ High to New Count Valid	*Figure 2a*		42			ns
t_{RST}	Counter Reset Pulse Width	*Figure 2a*	60				ns
t_{CTL}	RFI/O Low to Counter Outputs All Low	*Figure 2a*		100			ns
$t_{RFCKL,H}$	Minimum Pulse Width of RFCK	*Figure 7*	100				ns
T	Period of $\overline{RAS}$ Generator Clock	*Figure 3*	30				ns
t_{RGCKL}	Minimum Pulse Width Low of RGCK	*Figure 3*	15				ns
t_{RGCKH}	Minimum Pulse Width High of RGCK	*Figure 3*	15				ns
t_{FRQL}	RFCK Low to Forced $\overline{RFRQ}$ (RFI/O) Low	*Figure 3* C_L = 50 pF RL = 35k		66			ns
t_{FRQH}	RGCK Low to Forced $\overline{RFRQ}$ High	*Figure 3* C_L = 50 pF RL = 35k		55			ns
t_{RGRL}	RGCK Low to $\overline{RAS}$ Low	*Figure 3*	20	41			ns
t_{RGRH}	RGCK Low to $\overline{RAS}$ High	*Figure 3*	20	48			ns

Switching Characteristics: DP8417, DP8418, DP8419, DP8419X (Continued)

V_{CC} = 5.0V ±10%, 0°C ≤ T_A ≤ 70°C unless otherwise noted (Notes 2, 4, 5). The output load capacitance is typical for 4 banks of 22 DRAMs each or 88 DRAMs, including trace capacitance.

*** These values are Q0-Q8, C_L = 500 pF; $\overline{RAS0}$-$\overline{RAS3}$, C_L = 150 pF; $\overline{WE}$, C_L = 500 pF; $\overline{CAS}$, C_L = 600 pF; RL = 500Ω unless otherwise noted. See *Figure 11a* for test load. S1 is open unless otherwise specified. Maximum propagation delays are specified with all outputs switching.**

Symbol	Parameter	Condition	*CL		All C_L = 50 pF		Units
			Min	Max	Min	Max	
REFRESH (Continued)							
t_{RQHRF}	$\overline{RFSH}$ Hold Time from RGCK	*Figure 3*	2T				ns
t_{RFRH}	$\overline{RFSH}$ High to $\overline{RAS}$ High (Ending Forced Refresh early)	(See Mode 1 Description)		42			ns
t_{RFSRG}	$\overline{RFSH}$ Low Set-up to RGCK Low (Mode 1)	(See Mode 1 Description) *Figure 3*	12				ns
t_{CSHR}	$\overline{CS}$ High to $\overline{RASIN}$ Low for Hidden Refresh	*Figure 7*	10				ns
t_{RKRL}	RFCK High to $\overline{RASIN}$ low for hidden Refresh		50				ns
DP8419, DP8419X ONLY							
t_{CSRL1}	$\overline{CS}$ Low to Access $\overline{RASIN}$ Low (Using Mode 5 with Auto Refresh Mode)	*Figure 3*	34				ns
t_{CSRL0}	$\overline{CS}$ Low to Access $\overline{RASIN}$ Low (Using Modes 4 or 5 with externally controlled Refresh)	(See Mode 5 Description)	5				ns
DP8418 ONLY							
t_{CSRL1}	$\overline{CS}$ Low to Access $\overline{RASIN}$ Low (Using Mode 5 with Auto Refresh Mode)	*Figure 3*	5				ns
t_{CSRL0}	$\overline{CS}$ Low to Access $\overline{RASIN}$ Low (Using Modes 4 or 5 with externally controlled Refresh)	(See Mode 5 Description)	5				ns
DP8417 ONLY — PRELIMINARY							
t_{CSRL1}	$\overline{CS}$ Low to Access $\overline{RASIN}$ Low (Using Mode 5 with Auto Refresh Mode)	*Figure 3*	34				ns
t_{CSRL0}	$\overline{CS}$ Low to Access $\overline{RASIN}$ Low (Using Modes 4 or 5 with externally controlled Refresh)	(See Mode 5 Description)	34				ns
TRI-STATE (DP8417 ONLY)							
t_{ZH}	$\overline{CS}$ Low to Output High from Hi-Z	S1 Open *Figure 11b*		50			ns
t_{HZ}	$\overline{CS}$ High to Output Hi-Z from High	S1 Open, Q, $\overline{WE}$ *Figure 11b*				50	ns
t_{HZ}	$\overline{CS}$ High to Output Hi-Z from High	S1 Open, $\overline{RAS}$0-3 $\overline{CAS}$0-3 *Figure 11b*				95	ns
t_{ZL}	$\overline{CS}$ Low to Output Low from Hi-Z	S1 Closed *Figure 11b*		50			ns
t_{LZ}	$\overline{CS}$ High to Output Hi-Z from Low	S1 Closed *Figure 11b*				50	ns

Input Capacitance T_A = 25°C (Note 2)

Symbol	Parameter	Condition	Min	Typ	Max	Units
C_{IN}	Input Capacitance ADS, R/$\overline{C}$, $\overline{CS}$, M2, $\overline{RASIN}$			8		pF
C_{IN}	Input Capacitance All Other Inputs			5		pF

Note 1: "Absolute Maximum Ratings" are the values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: All typical values are for $T_A = 25°C$ and $V_{CC} = 5.0V$.

Note 3: This test is provided as a monitor of Driver output source and sink current capability. Caution should be exercised in testing this parameter. In testing these parameters, a 15Ω resistor should be placed in series with each output under test. One output should be tested at a time and test time should not exceed 1 second.

Note 4: Input pulse 0V to 3.0V, $t_R = t_F = 2.5$ ns, f = 2.5 MHz, $t_{PW} = 200$ ns. Input reference point on AC measurements is 1.5V Output reference points are 2.4V for High and 0.8V for Low.

Note 5: The load capacitance on RF I/O should not exceed 50 pF.

National Semiconductor

DP8428/NS32828, DP8429/NS32829
1 Megabit High Speed Dynamic RAM Controller/Drivers

General Description

The DP8428 and DP8429 1M DRAM Controller/Drivers are designed to provide "No-Waitstate" CPU interface to Dynamic RAM arrays of up to 8 Mbytes and larger. The DP8428 and DP8429 are tailored for 32-bit and 16-bit system requirements, respectively. Both devices are fabricated using National's new oxide isolated Advanced Low power Schottky (ALS) process and use design techniques which enable them to significantly out-perform all other LSI or discrete alternatives in speed, level of integration, and power consumption.

Each device integrates the following critical 1M DRAM controller functions on a single monolithic device: ultra precise delay line; 9 bit refresh counter; fall-through row, column, and bank select input latches; Row/Column address muxing logic; on-board high capacitive-load $\overline{RAS}$, $\overline{CAS}$, Write Enable and Address output drivers; and, precise control signal timing for all the above.

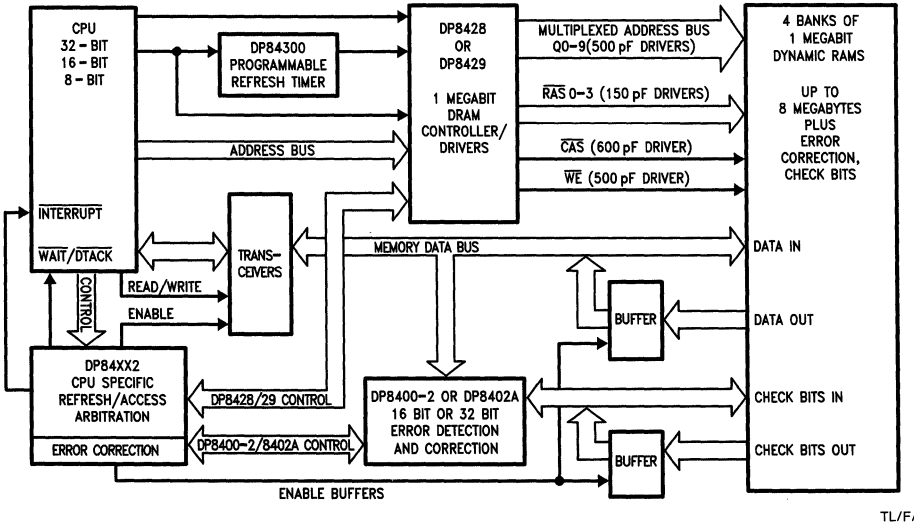
In order to specify each device for "true" worst case operating conditions, all timing parameters are guaranteed while the chip is driving the capacitive load of 88 DRAMs including trace capacitance. The chip's delay timing logic makes use of a patented new delay line technique which keeps AC skew to ±3 ns over the full V_{CC} range of ±10% and temperature range of −55°C to +125°C. The DP8428 and DP8429 guarantee a maximum RASIN to CASOUT delay of 80 ns or 70 ns even while driving an 8 Mbyte memory array with error correction check bits included. Two speed selected options of these devices are shown in the switching characteristics section of this document. (Continued)

Features

- Makes DRAM interface and refresh tasks appear virtually transparent to the CPU making DRAMs as easy to use as static RAMs
- Specifically designed to eliminate CPU wait states up to 10 MHz or beyond
- Eliminates 20 discrete components for significant board real estate reduction, system power savings and the elimination of chip-to-chip AC skewing
- On-board ultra precise delay line
- On-board high capacitive $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$ and Address drivers (specified driving 88 DRAMs directly)
- AC specified for directly addressing up to 8 Mbytes
- Low power/high speed bipolar oxide isolated process
- Downward pin and function compatible with 256k DRAM Controller/Drivers DP8409A, DP8417, DP8418, and DP8419

Contents

System Diagram



TL/F/8649-1

General Description (Continued)

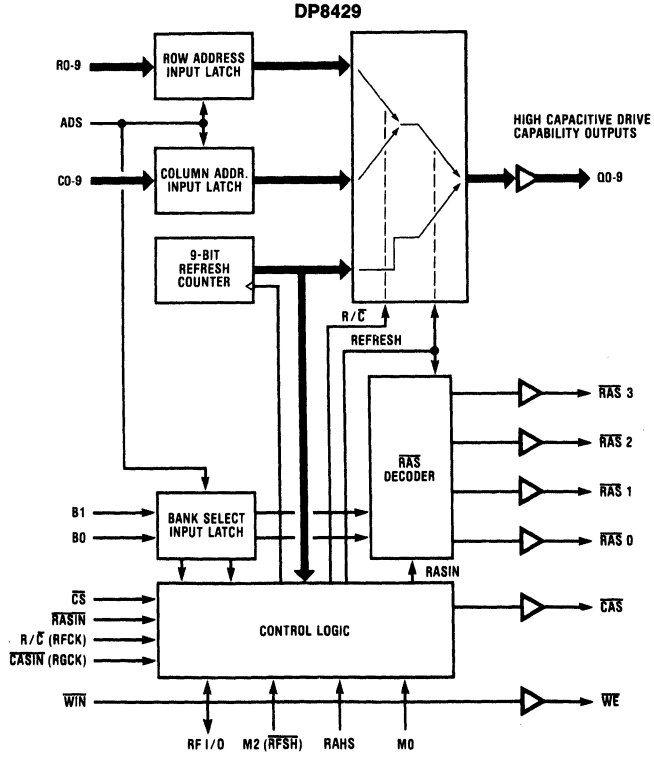
With its four independent $\overline{RAS}$ outputs and ten multiplexed address outputs, the DP8429 can support up to four banks of 64k, 256k or 1M DRAMs. Two bank select pins, B1 and B0, are decoded to activate one of the $\overline{RAS}$ signals during an access, leaving the three non-selected banks in the standby mode (less than one tenth of the operating power) with data outputs in TRI-STATE®. The DP8428's one Bank Select pin, B1, enables 2 banks automatically during an access in order to provide an optimum interface for 32-bit microprocessors.

The DP8428 and DP8429 each have two mode-select pins, allowing for two refresh modes and two access modes. Refresh and access timing may be controlled either externally or automatically. The automatic modes require a minimum of input control signals.

A refresh counter is on-chip and is multiplexed with the row and column inputs. Its contents appear at the address outputs of the DP8428 or DP8429 during any refresh, and are incremented at the completion of the refresh. Row, Column and bank address latches are also on-chip. However, if the address inputs to the DP8428 or DP8429 are valid throughout the duration of the access, these latches may be operated in the fall-through mode.

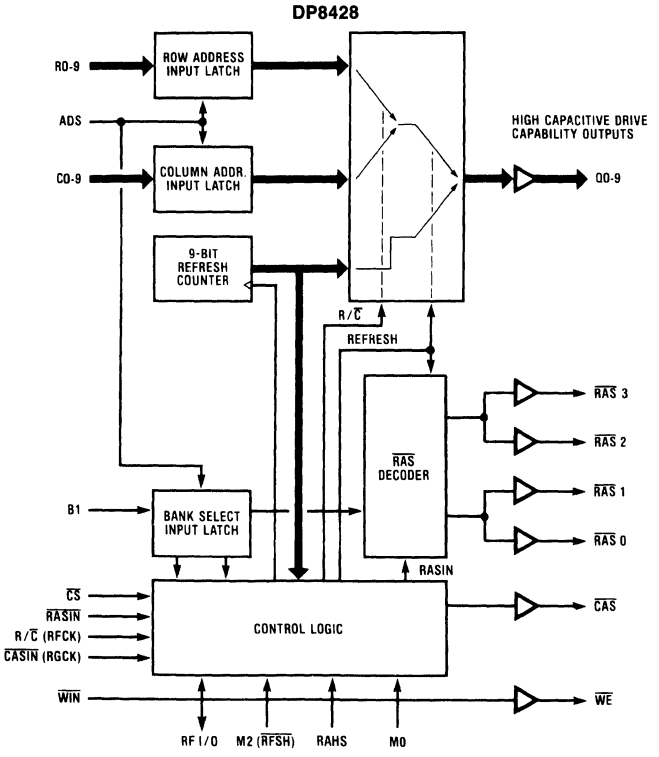
Each device is available in either the 52 pin Ceramic DIP, or the low cost JEDEC standard 68 pin Plastic Chip Carrier (PCC) package.

Functional Block Diagrams



TL/F/8649–2

Functional Block Diagrams (Continued)



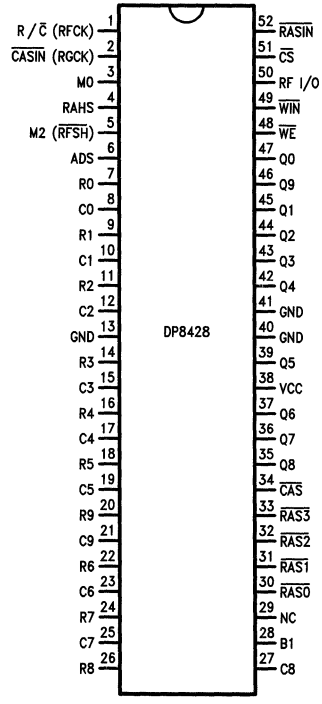
TL/F/8649–3

System Companion Components

Device #	Function
DP84300	Programmable Refresh Timer for DP84xx DRAM Controller
DP84412	NS32008/16/32 to DP8409A/17/18/19/28/29 Interface
DP84512	NS32332 to DP8417/18/19/28/29 Interface
DP84322	68000/08/10 to DP8409A/17/18/19/28/29 Interface (up to 8 MHz)
DP84422	68000/08/10 to DP8409A/17/18/19/28/29 Interface (up to 12.5 MHz)
DP84522	68020 to DP8417/18/19/28/29 Interface
DP84432	8086/88/186/188 to DP8409A/17/18/19/28/29 Interface
DP84532	80286 to DP8409A/17/18/19/28/29 Interface
DP8400-2	16-Bit Expandable Error Checker/Corrector (E2C2)
DP8402A	32-Bit Error Detector And Corrector (EDAC)

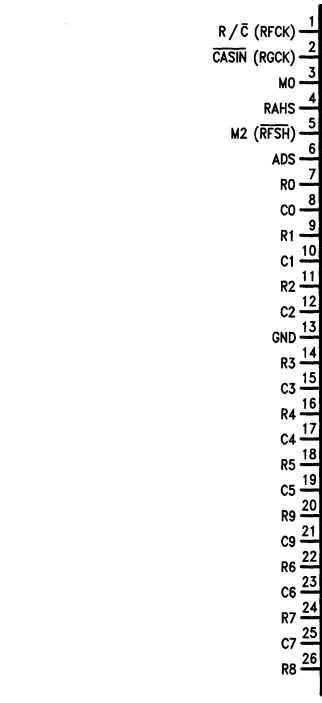
Connection Diagrams

Dual-In-Line Package



TL/F/8649–4

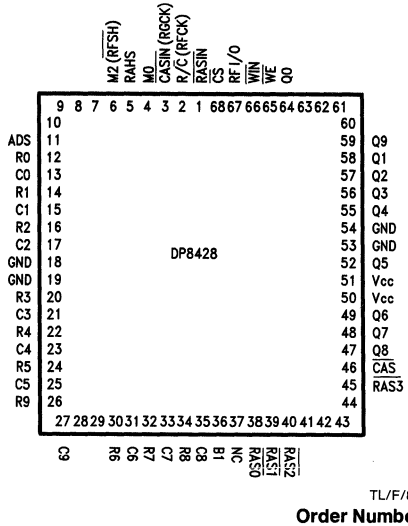
Dual-In-Line Package



TL/F/8649–5

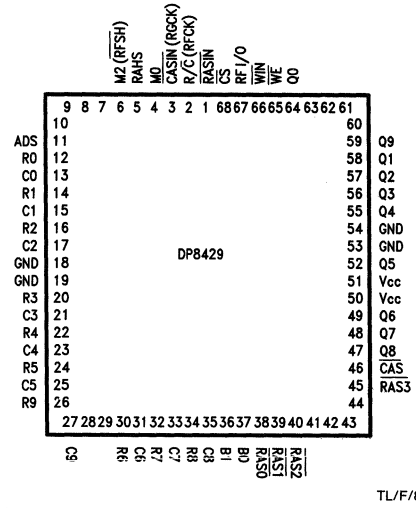
Order Number DP8428D-70, DP8428D-80 or
DP8429D-70, DP8429D-80
See NS Package Number D52A

Plastic Chip Carrier Package



TL/F/8649–6

Plastic Chip Carrier Package



TL/F/8649–7

Order Number DP8428V-70, DP8428V-80 or
DP8429V-70, DP8429V-80
See NS Package Number V68A

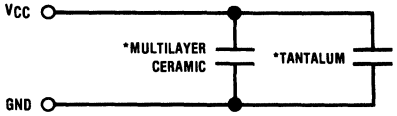
DP8428 vs DP8429

The DP8428 DYNAMIC RAM CONTROLLER/DRIVER is identical to the DP8429 with the exception of two functional differences incorporated to improve performance with 32-bit microprocessors.

1) Pin 28 (B1) is used to enable/disable a pair of $\overline{RAS}$ outputs, and pin 29 (B0 on the DP8429) is a no connect. When B1 is low, $\overline{RAS0}$ and $\overline{RAS1}$ are enabled such that they both go low during an access. When B1 is high, $\overline{RAS2}$ and $\overline{RAS3}$ are enabled. This feature is useful when driving words of 32 bits or more since each $\overline{RAS}$ would be driving only one half of the word. By distributing the load on each $\overline{RAS}$ line in this way, the DP8428 will meet the same AC specifications driving 2 banks of 32 DRAMs each as the DP8429 does driving 4 banks of 16 bits each.
2) The hidden refresh function available on the DP8429 has been disabled on the DP8428 in order to reduce the amount of setup time necessary from $\overline{CS}$ going low to $\overline{RASIN}$ going low during an access of DRAM. This parameter, called t_{CSRL1}, is 5 ns for the DP8428 whereas it is 34 ns for the DP8429. The hidden refresh function allowed only a very small increase in system performance, at microprocessor frequencies of 10 MHz and above.

Pin Definitions

V_{CC}, GND, GND – V_{CC} = 5V ±10%. The three supply pins have been assigned to the center of the package to reduce voltage drops, both DC and AC. There are two ground pins to reduce the low level noise. The second ground pin is located two pins from V_{CC}, so that decoupling capacitors can be inserted directly next to these pins. It is important to adequately decouple this device, due to the high switching currents that will occur when all 10 address bits change in the same direction simultaneously. A recommended solution would be a 1 μF multilayer ceramic capacitor in parallel with a low-voltage tantalum capacitor, both connected as close as possible to GND and V_{CC} to reduce lead inductance. See Figure below.



TL/F/8649–8

*Capacitor values should be chosen depending on the particular application.

R0–R9: Row Address Inputs.

C0–C9: Column Address Inputs.

Q0–Q9: Multiplexed Address Outputs – This address is selected from the Row Address Input Latch, the Column Address Input Latch or the Refresh Counter.

$\overline{RASIN}$: Row Address Strobe Input – $\overline{RASIN}$ directly controls the selected $\overline{RAS}$ output when in an access mode and all $\overline{RAS}$ outputs during hidden or external refresh.

R/$\overline{C}$ (RFCK) – In the auto-modes this pin is the external refresh clock input; one refresh cycle should be performed each clock period. In the external access mode it is Row/Column Select Input which enables either the row or column address input latch onto the output bus.

$\overline{CASIN}$ (RGCK) – In the auto-modes this pin is the $\overline{RAS}$ Generator Clock input. In external access mode it is the Column Address Strobe input which controls $\overline{CAS}$ directly once columns are enabled on the address outputs.

ADS: Address (Latch) Strobe Input – Row Address, Column Address, and Bank Select Latches are fall-through with ADS high; latching occurs on high-to-low transition of ADS.

$\overline{CS}$: Chip Select Input – When high, $\overline{CS}$ disables all accesses. Refreshing, however, in both modes 0 and 1 is not affected by this pin.

M0, M2 ($\overline{RFSH}$): Mode Control Inputs – These pins select one of the four available operational modes of the DP8429 (see Table III).

RFI/0: Refresh Input/Output – In the auto-modes this pin is the Refresh Request Output. It goes low following RFCK indicating that no hidden refresh was performed while RFCK was high. When this pin is set low by an external gate the on-chip refresh counter is reset to all zeroes.

$\overline{WIN}$: Write Enable Input.

$\overline{WE}$: Write Enable Output – $\overline{WE}$ follows $\overline{WIN}$ unconditionally.

RAHS: Row Address Hold Time Select – Selects the t_{RAH} to be guaranteed by the DP8428 or DP8429 delay line to allow for the use of fast or slow DRAMs.

$\overline{CAS}$: Column Address Strobe Output – In mode 5 and in mode 4 with $\overline{CASIN}$ low before R/$\overline{C}$ goes low, $\overline{CAS}$ goes low automatically after the column address is valid on the address outputs. In mode 4 $\overline{CAS}$ follows $\overline{CASIN}$ directly after R/$\overline{C}$ goes low, allowing for nibble accessing. $\overline{CAS}$ is always high during refresh.

$\overline{RAS}$ 0–3: Row Address Strobe Outputs – The enabled $\overline{RAS}$ output (see Table II) follows $\overline{RASIN}$ directly during an access. During refresh, all $\overline{RAS}$ outputs are enabled.

Pin Definitions (Continued)

B0, B1: Bank Select Inputs – These pins are decoded to enable one or two of the four $\overline{RAS}$ outputs during an access (see Table I and Table II).

TABLE I. DP8429 Memory Bank Decode

Bank Select (Strobed by ADS)		Enabled $\overline{RAS}_n$
B1	B0	
0	0	$\overline{RAS}_0$
0	1	$\overline{RAS}_1$
1	0	$\overline{RAS}_2$
1	1	$\overline{RAS}_3$

TABLE II. DP8428 Memory Bank Decode

Bank Select (Strobed by ADS)		Enabled $\overline{RAS}_n$
B1	NC	
0	X	$\overline{RAS}_0$ & $\overline{RAS}_1$
1	X	$\overline{RAS}_2$ & $\overline{RAS}_3$

Conditions for All Modes

INPUT ADDRESSING

The address block consists of a row-address latch, a column-address latch, and a resettable refresh counter. The address latches are fall-through when ADS is high and latch when ADS goes low. If the address bus contains valid addresses until after $\overline{CAS}$ goes low at the end of the memory cycle, ADS can be permanently high. Otherwise ADS must go low while the addresses are still valid.

DRIVE CAPABILITY

The DP8429 has timing parameters that are specified driving the typical capacitance (including traces) of 88, 5V-only DRAMs. Since there are 4 $\overline{RAS}$ outputs, each is specified driving one-fourth of the total memory. $\overline{CAS}$, $\overline{WE}$ and the address outputs are specified driving all 88 DRAMs.

The graph in *Figure 10* may be used to determine the slight variations in timing parameters, due to loading conditions other than 88 DRAMs.

Because of distributed trace capacitance and inductance and DRAM input capacitance, current spikes can be created, causing overshoots and undershoots at the DRAM inputs that can change the contents of the DRAMs or even destroy them. To reduce these spikes, a damping resistor (low inductance, carbon) should be inserted between the DP8429 outputs and the DRAMs, as close as possible to the DP8429. The damping resistor values may differ depending on how heavily an output is loaded. These resistors should be determined by the first prototypes (not wire-wrapped due to the larger distributed capacitance and inductance). Resistors should be chosen such that the transition on the control outputs is critically damped. Typical values will be from 15Ω to 100Ω, with the lower values being used with the larger memory arrays. Note that AC parameters are specified with 15Ω damping resistors. For more information see AN-305 "Precautions to Take When Driving Memories".

DP8429 DRIVING ANY 256k or 1M DRAMS

The DP8429 can drive any 256k or 1M DRAMs. 256k DRAMs require 18 of the DP8429's address inputs to select one memory location within the DRAM. $\overline{RAS}$-only refreshing with the nine-bit refresh-counter on the DP8429 makes $\overline{CAS}$ before $\overline{RAS}$ refreshing, available on 256k DRAMs, unnecessary (see *Figure 1a*).

1 Mbit DRAMs require the use of all 10 of the DP8429 Address Outputs (see *Figure 1b*).

READ, WRITE AND READ-MODIFY-WRITE CYCLES

The output signal, $\overline{WE}$, determines what type of memory access cycle the memory will perform. If $\overline{WE}$ is kept high while $\overline{CAS}$ goes low, a read cycle occurs. If $\overline{WE}$ goes low before $\overline{CAS}$ goes low, a write cycle occurs and data at DI (DRAM input data) is written into the DRAM as $\overline{CAS}$ goes low. If $\overline{WE}$ goes low later than t_{CWD} after $\overline{CAS}$ goes low, first a read occurs and DO (DRAM output data) becomes valid, then data DI is written into the same address in the DRAM as $\overline{WE}$ goes low. In this read-modify-write case, DI and DO cannot be linked together. $\overline{WE}$ always follows $\overline{WIN}$ directly to determine the type of access to be performed.

POWER-UP INITIALIZE

When V_{CC} is first applied to the DP8429, an initialize pulse clears the refresh counter and the internal control flip-flops.

Mode Features Summary

- 4 modes of operation: 2 access and 2 refresh
- Automatic or external selected by the user
- Auto access mode provides $\overline{RAS}$, row to column change, and then $\overline{CAS}$ automatically.
- Choice between two different values of t_{RAH} in auto-access mode
- $\overline{CAS}$ controlled independently in external control mode, allowing for nibble mode accessing
- Automatic refreshing can make refreshes transparent to the system
- $\overline{CAS}$ is inhibited during refresh cycles

DP8428/DP8429 Mode Descriptions

MODE 0–EXTERNALLY CONTROLLED REFRESH

Figure 2 shows the Externally Controlled Refresh timing. In this mode the refresh counter contents are multiplexed to the address outputs. All $\overline{RAS}$ outputs are enabled to follow $\overline{RASIN}$ so that the row address indicated by the refresh counter is refreshed in all DRAM banks when $\overline{RASIN}$ goes low. The refresh counter increments when $\overline{RASIN}$ goes high. $\overline{RFSH}$ should be held low at least until $\overline{RASIN}$ goes high (they may go high simultaneously) so that the refresh address remains valid and all $\overline{RAS}$ outputs remain enabled throughout the refresh.

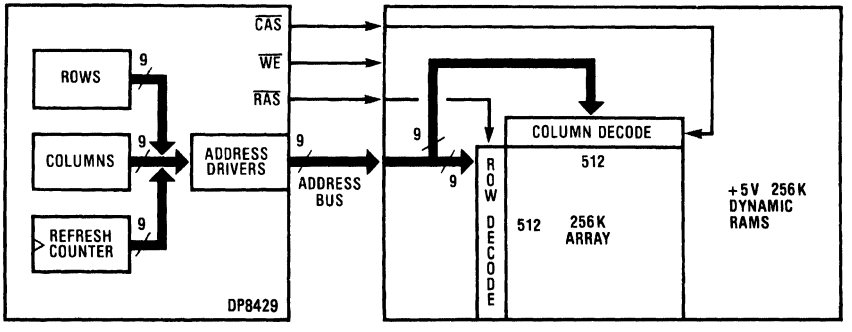
A burst refresh may be performed by holding $\overline{RFSH}$ low and toggling $\overline{RASIN}$ until all rows are refreshed. It may be useful in this case to reset the refresh counter just prior to beginning the refresh. The refresh counter resets to all zeroes when RFI/O is pulled low by an external gate. The refresh counter always counts to 511 before rolling over to zero. If there are 128 or 256 rows being refreshed then Q7 or Q8, respectively, going high may be used as an end-of-burst indicator.

In order that the refresh address is valid on the address outputs prior to the $\overline{RAS}$ lines going low, $\overline{RFSH}$ must go low before $\overline{RASIN}$. The setup time required is given by t_{RFLRL} in the Switching Characteristics. This parameter may be adjusted using *Figure 10* for loading conditions other than those specified.

TABLE III. DP8428/DP8429 Mode Select Options

Mode	($\overline{RFSH}$) M2	M0	Mode of Operation
0	0	0	Externally Controlled Refresh
1	0	1	Auto Refresh–Forced
4	1	0	Externally Controlled Access
5	1	1	Auto Access (Hidden Refresh)

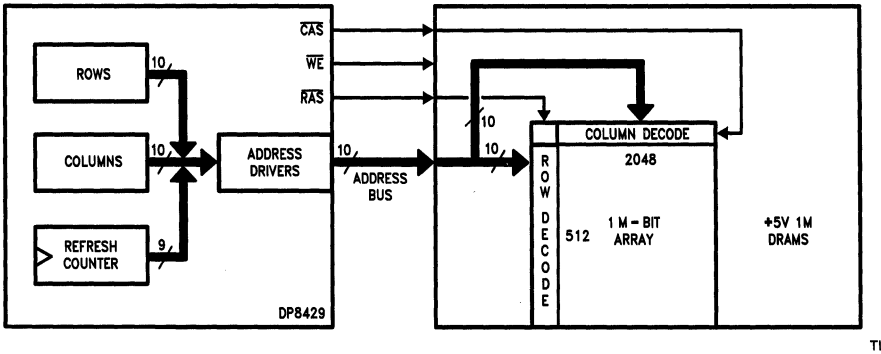
DP8428/DP8429 Interface Between System and DRAM Banks



TL/F/8649–12

All 9 Bits of Refresh Counter Used

FIGURE 1a. DP8428/DP8429 with 256k DRAMs

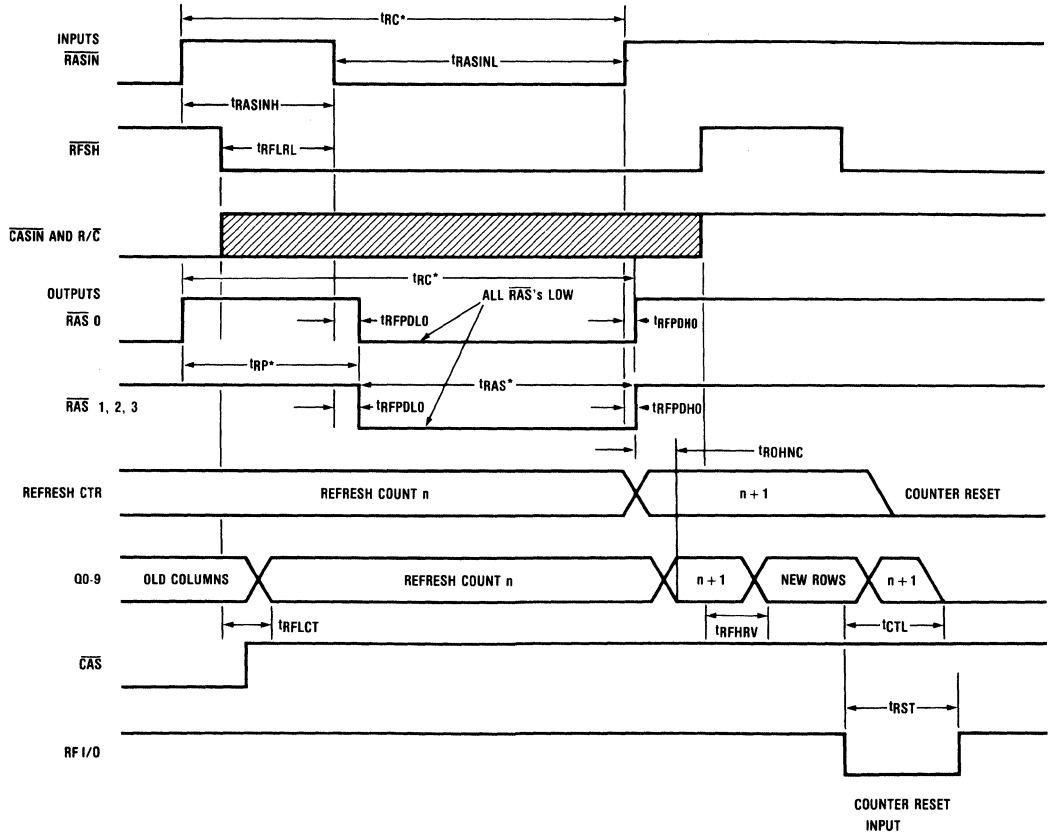


TL/F/8649–25

All 9 Bits of Refresh Counter Used

FIGURE 1b. DP8428/DP8429 with 1M DRAMs

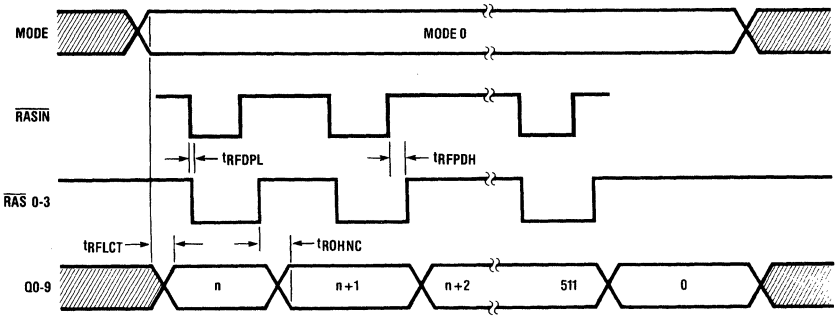
DP8428/DP8429 Mode Descriptions (Continued)



*Indicates Dynamic RAM Parameters

TL/F/8649–13

FIGURE 2a. External Control Refresh Cycle (Mode 0)



TL/F/8649–14

FIGURE 2b. Burst Refresh Mode 0

DP8428/DP8429 Mode Descriptions (Continued)

MODE 1–AUTOMATIC FORCED REFRESH

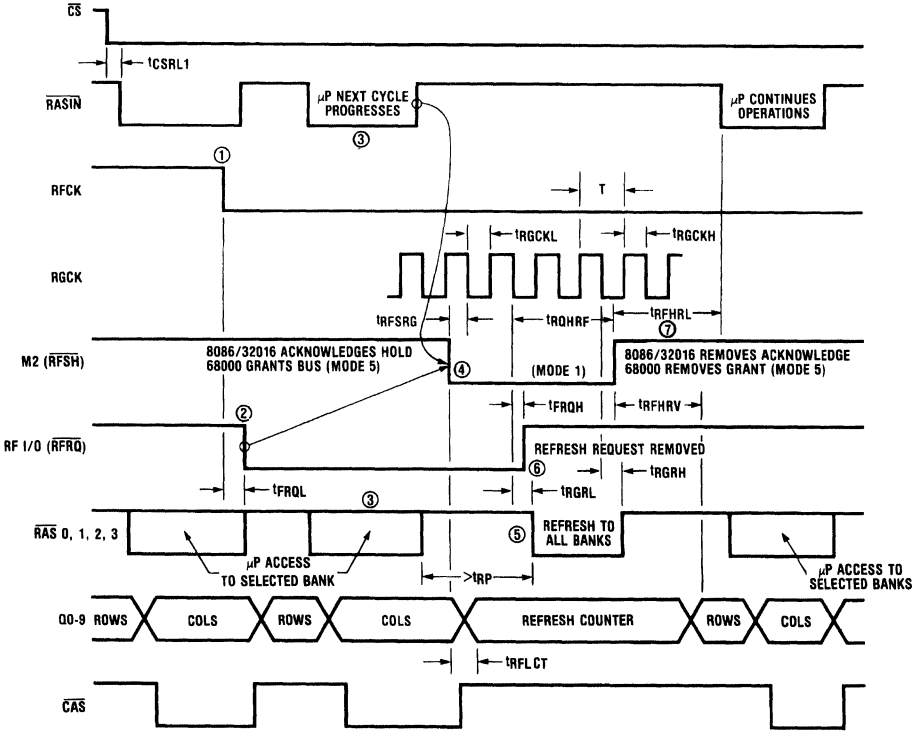
In Mode 1 the R/$\overline{C}$ (RFCK) pin becomes RFCK (refresh cycle clock) and the $\overline{CASIN}$ (RGCK) pin becomes RGCK ($\overline{RAS}$ generator clock). If RFCK is high and Mode 1 is entered then the chip operates as if in MODE 0 (externally controlled refresh), with all $\overline{RAS}$ outputs following $\overline{RASIN}$. This feature of Mode 1 may be useful for those who want to use Mode 5 (automatic access) with externally controlled refresh. By holding RFCK permanently high one need only toggle M2 ($\overline{RFSH}$) to switch from Mode 5 to external refresh. As with Mode 0, RFI/O may be pulled low by an external gate to reset the refresh counter.

When using Mode 1 as automatic refresh, RFCK must be an input clock signal. One refresh should occur each period of RFCK. If no refresh is performed while RFCK is high, then when RFCK goes low RFI/O immediately goes low to indicate that a refresh is requested. (RFI/O may still be used to reset the refresh counter even though it is also used as a refresh request pin, however, an open-collector gate should be used to reset the counter in this case since RFI/O is forced low internally for a request).

After receiving the refresh request the system must allow a forced refresh to take place while RFCK is low. External logic can monitor $\overline{RFRQ}$ (RFI/O) so that when $\overline{RFRQ}$ goes low this logic will wait for the access currently in progress to be completed before pulling M2 ($\overline{RFSH}$) low to put the DP8429 in mode 1. If no access is taking place when $\overline{RFRQ}$ occurs, then M2 may immediately go low. Once M2 is low, the refresh counter contents appear at the address outputs and $\overline{RAS}$ is generated to perform the refresh.

An external clock on RGCK is required to derive the refresh $\overline{RAS}$ signals. On the second falling edge of RGCK after M2 is low, all $\overline{RAS}$ lines go low. They remain low until two more falling edges of RGCK. Thus $\overline{RAS}$ remains high for one to two periods of RGCK after M2 goes low, and stays low for two periods. In order to obtain the minimum delay from M2 going low to $\overline{RAS}$ going low, M2 should go low t_{RFSRG} before the falling edge of RGCK.

The Refresh Request on RFI/O is terminated as $\overline{RAS}$ goes low. This signal may be used to end the refresh earlier than it normally would as described above. If M2 is pulled high



TL/F/8649–15

① RFCK goes low
② $\overline{RFRQ}$ goes low if no hidden refresh occurred while RFCK was high
③ Next $\overline{RASIN}$ starts next access
④ μP acknowledges refresh request
⑤ Forced refresh $\overline{RAS}$ starts after > T (> t_{RP})
⑥ Forced refresh $\overline{RAS}$ ends $\overline{RFRQ}$
⑦ μP removes refresh acknowledge

FIGURE 3. DP8428/DP8429 Performing a Forced Refresh (Mode 5 → 1 → 5) with Various Microprocessors

DP8428/DP8429 Mode Descriptions (Continued)

while the $\overline{RAS}$ lines are low, then the $\overline{RAS}$s go high t_{RFRH} later. The designer must be careful, however, not to violate the minimum $\overline{RAS}$ low time of the DRAMs. He must also guarantee that the minimum $\overline{RAS}$ precharge time is not violated during a transition from mode 1 to mode 5 when an access is desired immediately following a refresh.

If the processor tries to access memory while the DP8429 is in mode 1, WAIT states should be inserted into the processor cycles until the DP8429 is back in mode 5 and the desired access has been accomplished (see *Figure 9*).

Instead of using WAIT states to delay accesses when refreshing, HOLD states could be used as follows. $\overline{RFRQ}$ could be connected to a HOLD or Bus Request input to the system. When convenient, the system acknowledges the $\overline{HOLD}$ or Bus Request by pulling M2 low. Using this scheme, $\overline{HOLD}$ will end as the $\overline{RAS}$ lines go low (RFI/O goes high). Thus, there must be sufficient delay from the time $\overline{HOLD}$ goes high to the DP8429 returning to mode 5, so that the $\overline{RAS}$ low time of the DRAMs isn't violated as described earlier (see *Figure 3* for mode 1 refresh with Hold states).

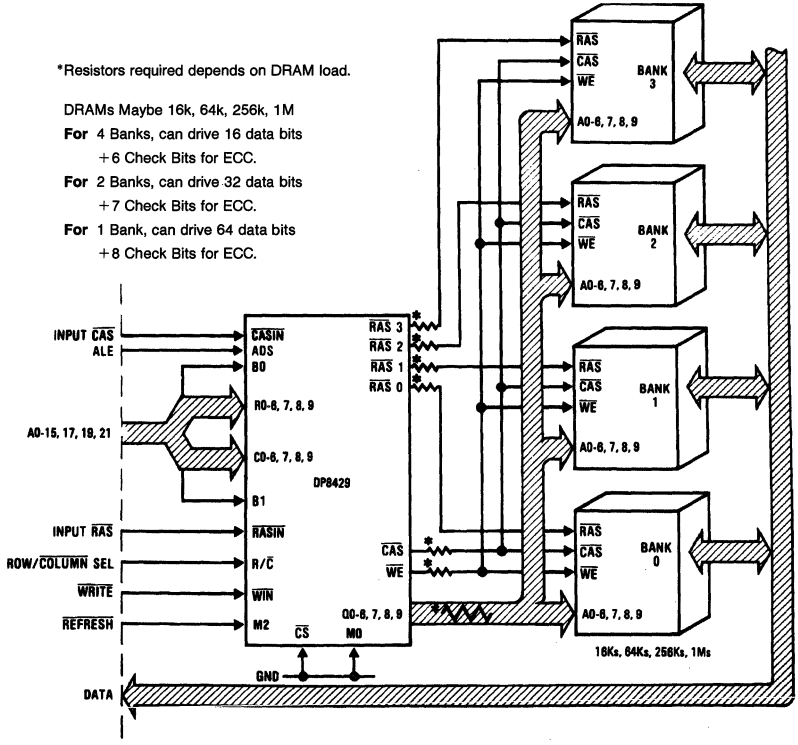
To perform a forced refresh the system will be inactive for about four periods of RGCK. For a frequency of 10 MHz, this is 400 ns. To refresh 128 rows every 2 ms an average of about one refresh per 16 μs is required. With a RFCK period of 16 μs and RGCK period of 100 ns, DRAM accesses are delayed due to refresh only 2.5% of the time. If using the Hidden Refresh available in mode 5 (refreshing with RFCK high) this percentage will be even lower.

MODE 4 - EXTERNALLY CONTROLLED ACCESS

In this mode all control signal outputs can be controlled directly by the corresponding control input. The enabled $\overline{RAS}$ output follows $\overline{RASIN}$, $\overline{CAS}$ follows $\overline{CASIN}$ (with R/$\overline{C}$ low), $\overline{WE}$ follows $\overline{WIN}$ and R/$\overline{C}$ determines whether the row or the column inputs are enabled to the address outputs (see *Figure 4*).

With R/$\overline{C}$ high, the row address latch contents are enabled onto the address bus. $\overline{RAS}$ going low strobes the row address into the DRAMs. After waiting to allow for sufficient row-address hold time (t_{RAH}) after $\overline{RAS}$ goes low, R/$\overline{C}$ can go low to enable the column address latch contents onto the address bus. When the column address is valid, $\overline{CAS}$ going low will strobe it into the DRAMs. $\overline{WIN}$ determines whether the cycle is a read, write or read-modify-write access. Refer to *Figures 5a* and *5b* for typical Read and Write timing using mode 4.

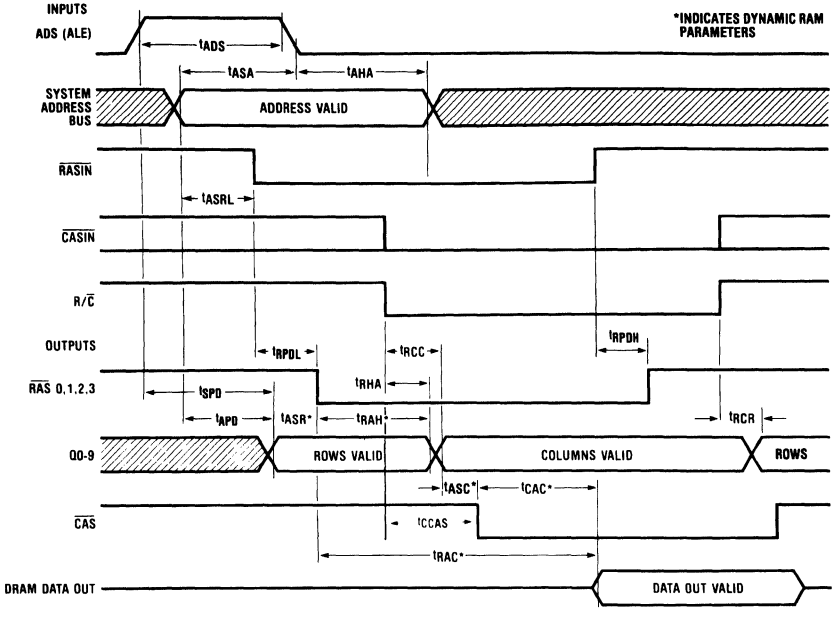
Page or Nibble mode may be performed by toggling $\overline{CASIN}$ once the initial access has been completed. In the case of page mode the column address must be changed before



TL/F/8649–16

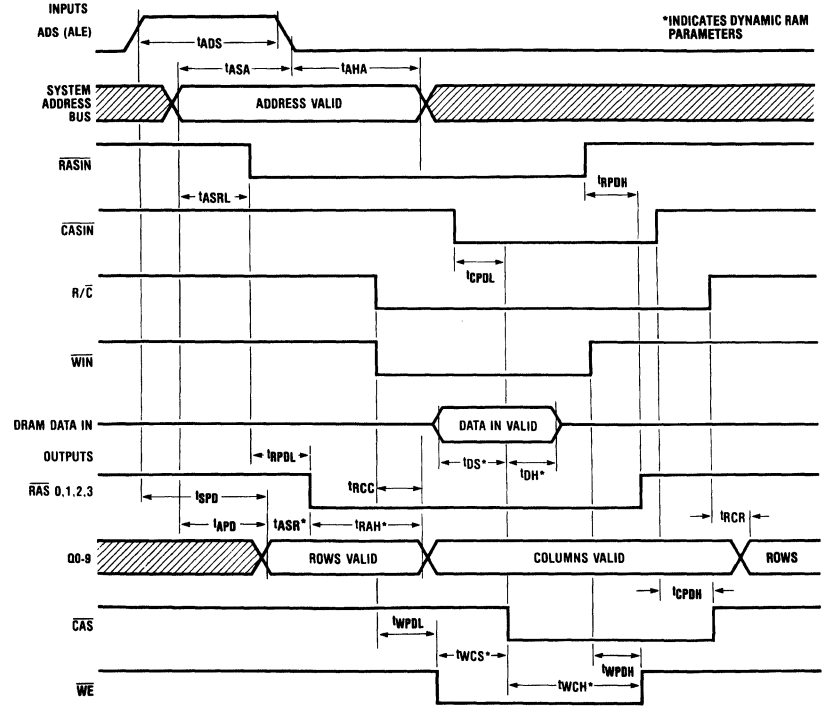
FIGURE 4. Typical Application of DP8429 Using External Control Access and Refresh in Modes 0 and 4

DP8428/DP8429 Mode Descriptions (Continued)



TL/F/8649–17

FIGURE 5a. Read Cycle Timing (Mode 4)



TL/F/8649–18

FIGURE 5b. Write Cycle Timing (Mode 4)

DP8428/DP8429 Mode Descriptions (Continued)

$\overline{CASIN}$ goes low to access a new memory location (see *Figure 5c*). Parameter t_{CPdif} has been specified in order that users may easily determine minimum $\overline{CAS}$ pulse widths when $\overline{CASIN}$ is toggling.

AUTOMATIC $\overline{CAS}$ GENERATION

$\overline{CAS}$ is held high when R/$\overline{C}$ is high even if $\overline{CASIN}$ is low. If $\overline{CASIN}$ is low when R/$\overline{C}$ goes low, $\overline{CAS}$ goes low automatically, t_{ASC} after the column address is valid. This feature eliminates the need for an externally derived $\overline{CASIN}$ signal to control $\overline{CAS}$ when performing a simple access (*Figure 5a* demonstrates Auto-$\overline{CAS}$ generation in mode 4). Page or nibble accessing may be performed as shown in *Figure 5c* even if $\overline{CAS}$ is generated automatically for the initial access.

FASTEST MEMORY ACCESS

The fastest Mode 4 access is achieved by using the automatic $\overline{CAS}$ feature and external delay line to generate the required delay between $\overline{RASIN}$ and R/$\overline{C}$. The amount of delay required depends on the minimum t_{RAH} of the DRAMs being used. The DP8429 parameter t_{DIF1} has been specified in order that the delay between $\overline{RASIN}$ and R/$\overline{C}$ may be minimized.

$$t_{DIF1} = \text{MAXIMUM } (t_{RPDL} - t_{RHA})$$

where t_{RPDL} = $\overline{RASIN}$ to $\overline{RAS}$ delay

and t_{RHA} = row address held from R/$\overline{C}$ going low.

The delay between $\overline{RASIN}$ and R/$\overline{C}$ that guarantees the specified DRAM t_{RAH} is given by

$$\text{MINIMUM } \overline{RASIN} \text{ to R/}\overline{C} = t_{DIF1} + t_{RAH}.$$

Example

In an application using DRAMs that require a minimum t_{RAH} of 15 ns, the following demonstrates how the maximum $\overline{RASIN}$ to $\overline{CAS}$ time is determined.

With t_{DIF1} (from Switching Characteristics) = 7 ns,

$\overline{RASIN}$ to R/$\overline{C}$ delay = 7 ns + 15 ns = 22 ns.

A delay line of 25 ns will be sufficient.

With Auto-$\overline{CAS}$ generation, the maximum delay from R/$\overline{C}$ to $\overline{CAS}$ (loaded with 600 pF) is 46 ns. Thus the maximum $\overline{RASIN}$ to $\overline{CAS}$ time is 71 ns, under the given conditions.

With a maximum $\overline{RASIN}$ to $\overline{RAS}$ time (t_{RPDL}) of 20 ns, the maximum $\overline{RAS}$ to $\overline{CAS}$ time is about 51 ns. Most DRAMs with a 15 ns minimum t_{RAH} have a maximum t_{RCD} of about 60 ns. Thus memory accesses are likely to be $\overline{RAS}$ limited instead of $\overline{CAS}$ limited. In other words, memory access time is limited by DRAM performance, not controller performance.

REFRESHING IN CONJUNCTION WITH MODE 4

If using mode 4 to access memory, mode 0 (externally controlled refresh) must be used for all refreshing.

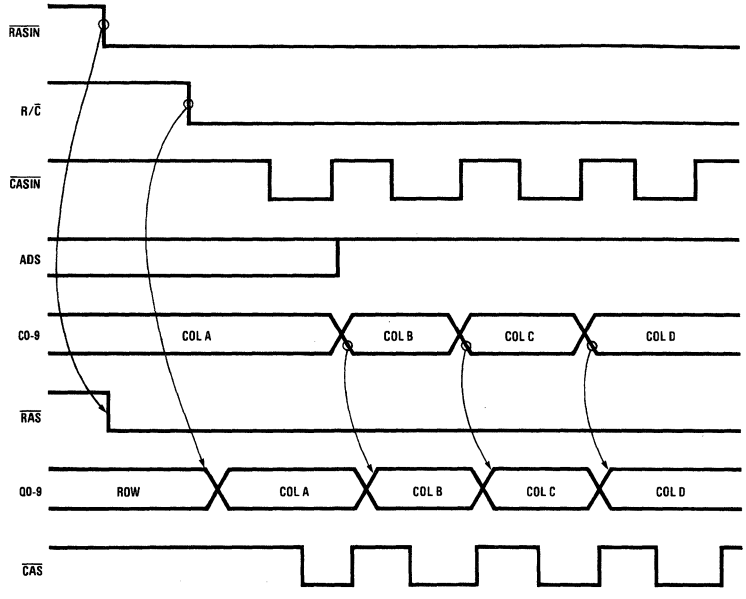
MODE 5 – AUTOMATIC ACCESS WITH HIDDEN REFRESHING CAPABILITY

Automatic-Access has two advantages over the externally controlled access (mode 4). First, $\overline{RAS}$, $\overline{CAS}$ and the row to column change are all derived internally from one input signal, $\overline{RASIN}$. Thus the need for an external delay line (see mode 4) is eliminated.

Secondly, since R/$\overline{C}$ and $\overline{CASIN}$ are not needed to generate the row to column change and $\overline{CAS}$, these pins can be used for the automatic refreshing function.

AUTOMATIC ACCESS CONTROL

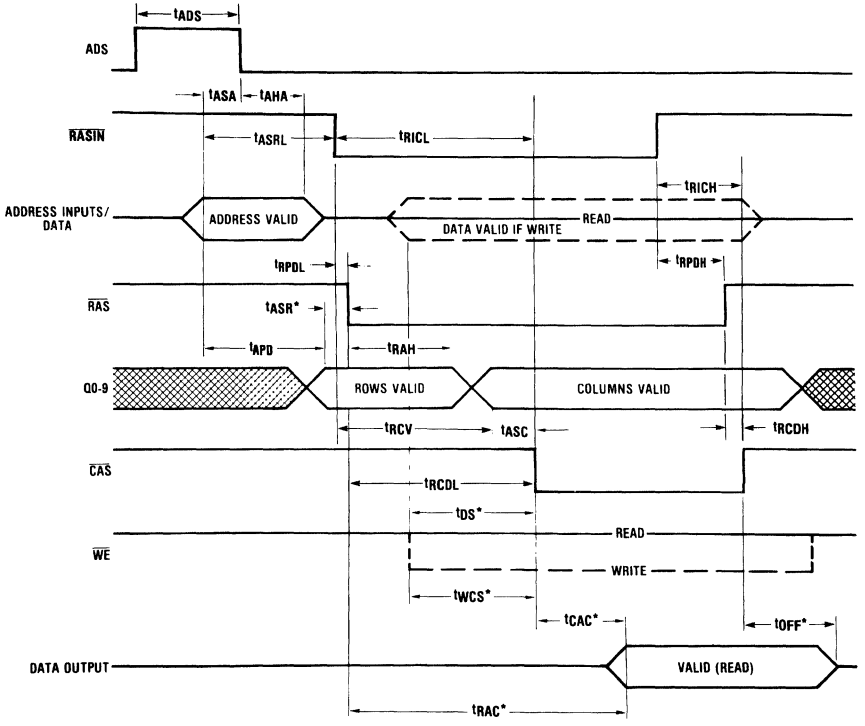
Mode 5 of the DP8429 makes accessing Dynamic RAM nearly as easy as accessing static RAM. Once row and column addresses are valid (latched on the DP8429 if necessary), $\overline{RASIN}$ going low is all that is required to perform the memory access.



TL/F/8649–19

FIGURE 5c. Page or Nibble Access in Mode 4

DP8428/DP8429 Mode Descriptions (Continued)



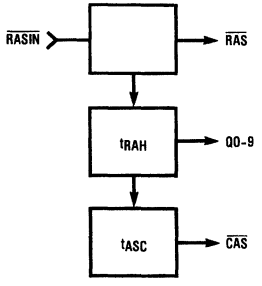
*Indicates Dynamic RAM Parameters

TL/F/8649–20

FIGURE 6. Mode 5 Timing

(Refer to *Figure 6*) In mode 5 the selected $\overline{RAS}$ follows $\overline{RASIN}$ immediately, as in mode 4, to strobe the row address into the DRAMs. The row address remains valid on the DP8429 address outputs long enough to meet the t_{RAH} requirement of the DRAMs (pin 4, RAHS, of the DP8429 allows the user two choices of t_{RAH}). Next, the column address replaces the row address on the address outputs and $\overline{CAS}$ goes low to strobe the columns into the DRAMs. $\overline{WIN}$ determines whether a read, write or read-modify-write is done.

The diagram below illustrates mode 5 automatic control signal generation.



TL/F/8649–21

REFRESHING IN CONJUNCTION WITH MODE 5

When using mode 5 to perform memory accesses, refreshing may be accomplished:

(a) externally (in mode 0 or mode 1)

(b) by a combination of mode 5 (hidden refresh) and mode 1 (auto-refresh)

or (c) by a combination of mode 5 and mode 0

(a) Externally Controlled Refreshing in Mode 0 or Mode 1

All refreshing may be accomplished using external refreshes in either mode 0 or mode 1 with R/$\overline{C}$ (RFCK) tied high (see mode 0 and mode 1 descriptions). If this is desired, the system determines when a refresh will be performed, puts the DP8429 in the appropriate mode, and controls the $\overline{RAS}$ signals directly with $\overline{RASIN}$. The on-chip refresh counter is enabled to the address outputs of the DP8429 when the refresh mode is entered, and increments when $\overline{RASIN}$ goes high at the completion of the refresh.

(b) Mode 5 Refreshing (hidden) with Mode 1 refreshing (auto)

(Refer to *Figure 7a*) If RFCK is tied to a clock (see mode 1 description), RFI/O becomes a refresh request output and goes low following RFCK going low if no refresh occurred while RFCK was high. Refreshes may be performed in mode 5 when the DP8429 is not selected for access ($\overline{CS}$ is high) and RFCK is high. If these conditions exist the refresh counter contents appear on the DP8429 address outputs and all $\overline{RAS}$ lines follow $\overline{RASIN}$ so that if $\overline{RASIN}$ goes low (an access other than through the DP8429 occurs), all $\overline{RAS}$ lines go low to perform the refresh. The DP8429 allows only one refresh of this type for each period of RFCK, since RFCK should be fast enough such that one refresh per period is sufficient to meet the DRAM refresh requirement.

DP8428/DP8429 Mode Descriptions (Continued)

Once it is started, a hidden refresh will continue even if RFCK goes low. However, $\overline{CS}$ must be high throughout the refresh (until $\overline{RASIN}$ goes high).

These hidden refreshes are valuable in that they do not delay accesses. When determining the duty cycle of RFCK, the high time should be maximized in order to maximize the probability of hidden refreshes. If a hidden refresh doesn't happen, then a refresh request will occur on RFI/O when RFCK goes low. After receiving the request, the system must perform a refresh while RFCK is low. This may be done by going to mode 1 and allowing an automatic refresh (see mode 1 description). This refresh must be completed while RFCK is low, thus the RFCK low time is determined by the worst-case time required by the system to respond to a refresh request.

(c) Mode 5 Refresh (Hidden Refresh) with mode 0 Refresh (External Refresh)

This refresh scheme is identical to that in (b) except that after receiving a refresh request, mode 0 is entered to do the refresh (see mode 0 description). The refresh request is terminated (RFI/O goes high) as soon as mode 0 is entered. This method requires more control than using mode 1 (auto-refresh), however, it may be desirable if the mode 1 refresh time is considered to be excessive.

Example

Figure 7b demonstrates how a system designer would use the DP8429 in mode 5 based on certain characteristics of his system.

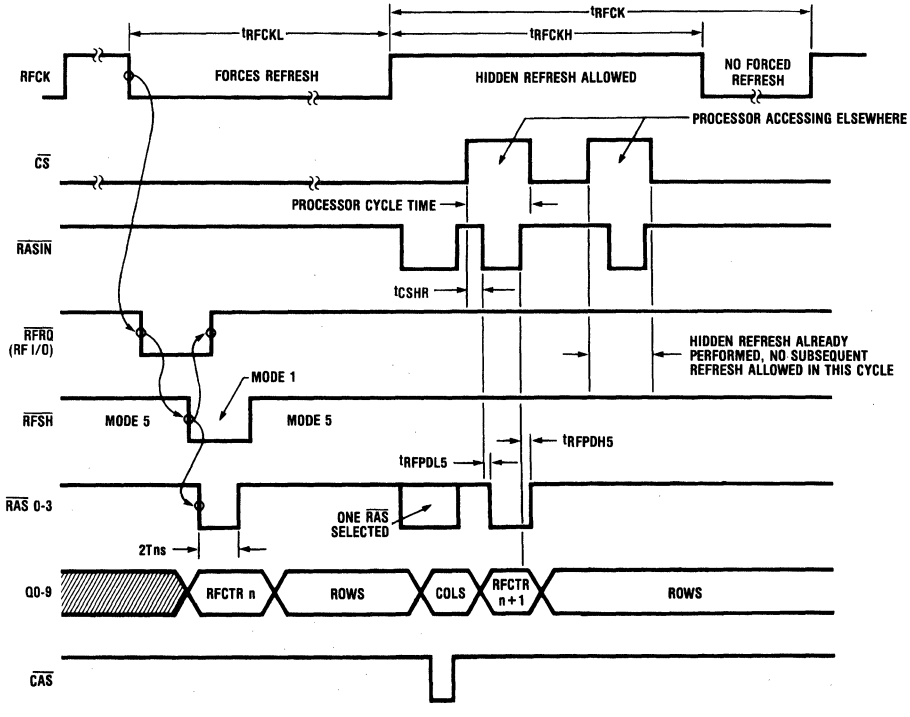
System Characteristics:

1) DRAM used has min t_{RAH} requirement of 15 ns and min t_{ASR} of 0 ns
2) DRAM address is valid from time T_V to the end of the memory cycle
3) four banks of twenty-two 256k memory chips each are being driven

Using the DP8429 (see *Figure 7b*):

1) Tie pin 4 (RAHS) high to guarantee a 15 ns minimum t_{RAH} which is sufficient for the DRAMs being used
2) Generate $\overline{RASIN}$ no earlier than time $T_V + t_{ASRL}$ (see switching characteristics), so that the row address is valid on the DRAM address inputs before $\overline{RAS}$ occurs
3) Tie ADS high since latching the DRAM address on the DP8429 is not necessary
4) Connect the first 20 system address bits to R0-R9 and C0-C9, and bits 21 and 22 to B0 and B1
5) Connect each $\overline{RAS}$ output of the DP8429 to the $\overline{RAS}$ inputs of the DRAMs of one bank of the memory array; connect Q0-Q9 of the DP8429 to A0-A9 of all DRAMs; connect $\overline{CAS}$ of the DP8429 to $\overline{CAS}$ of all the DRAMs

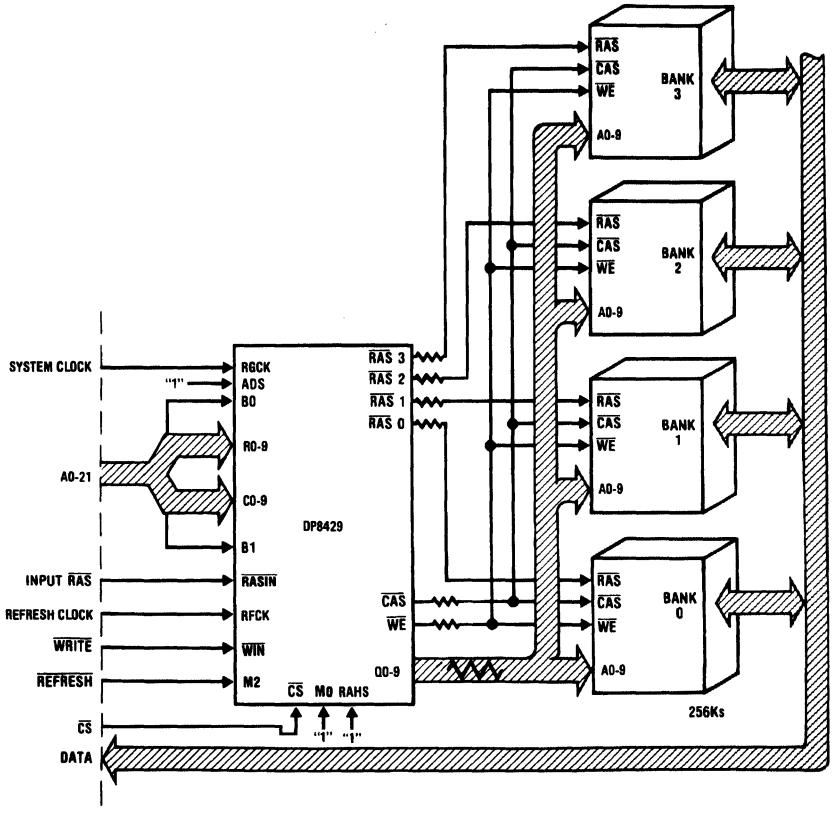
Figure 7c illustrates a similar example using the DP8428 to drive two 32-bit banks.



TL/F/8649-22

FIGURE 7a. Hidden Refreshing (Mode 5) and Forced Refreshing (Mode 1) Timing

DP8428/DP8429 Mode Descriptions (Continued)



TL/F/8649–23

FIGURE 7b. Typical Application of DP8429 Using Modes 5 and 1

Applications

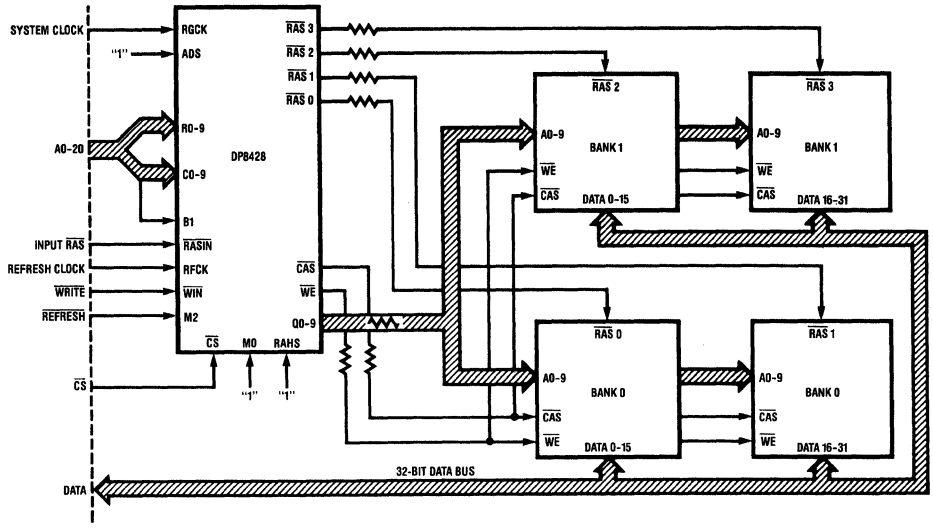
If one desires a memory interface containing the DP8429 that minimizes the number of external components required, modes 5 and 1 should be used. These two modes provide:

1) Automatic access to memory (in mode 5 only one signal, $\overline{\text{RASIN}}$, is required in order to access memory)
2) Hidden refresh capability (refreshes are performed automatically while in mode 5 when non-local accesses are taking place, as determined by $\overline{\text{CS}}$)
3) Refresh request capability (if no hidden refresh took place while RFCK was high, a refresh request is generated at the RFI/O pin when RFCK goes high)
4) Automatic forced refresh (If a refresh request is generated while in mode 5, as described above, external logic should switch the DP8429 into mode 1 to do an automatic forced refresh. No other external control signals need be issued. WAIT states can be inserted into the processor machine cycles if the system tries to access memory while the DP8429 is in mode 1 doing a forced refresh).

Some items to be considered when integrating the DP8429 into a system design are:

1) The system designer should ensure that a DRAM access not be in progress when a refresh mode is entered. Similarly, one should not attempt to start an access while a refresh is in progress. The parameter t_{RFHRL} specifies the minimum time from $\overline{\text{RFSH}}$ high to $\overline{\text{RASIN}}$ going low to initiate an access.
2) One should always guarantee that the DP8429 is enabled for access prior to initiating the access (see t_{CSRL1}).
3) One should bring $\overline{\text{RASIN}}$ low even during non-local access cycles when in mode 5 in order to maximize the chance of a hidden refresh occurring.
4) At lower frequencies (under 10 Mhz), it becomes increasingly important to differentiate between READ and WRITE cycles. $\overline{\text{RASIN}}$ generation during READ cycles can take place as soon as one knows that a processor READ access cycle has started. WRITE cycles, on the other hand, cannot start until one knows that the data to be written at the DRAM inputs will be valid a setup time before $\overline{\text{CAS}}$ (column address strobe) goes true at the DRAM inputs. Therefore, in general, READ cycles can be initiated earlier than WRITE cycles.
5) Many times it is possible to only add WAIT states during READ cycles and have no WAIT states during WRITE cycles. This is because it generally takes less time to write data into memory than to read data from memory.

Applications (Continued)



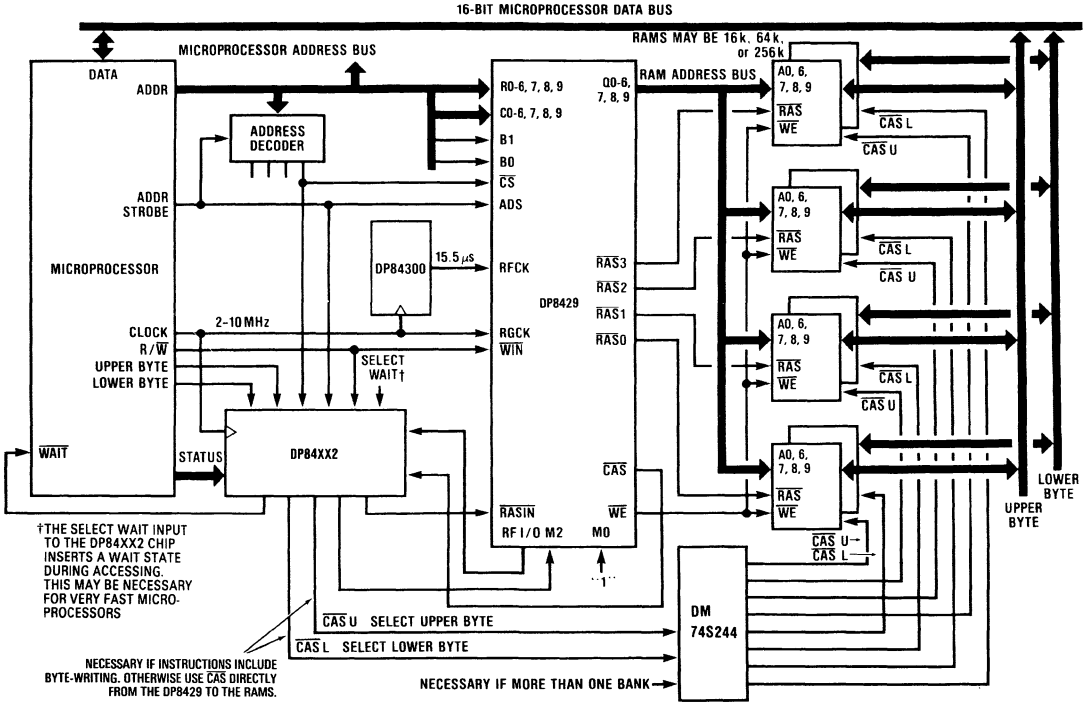
TL/F/8649–24

FIGURE 7c. Typical Application of DP8428 Using Modes 5 and 1

The DP84XX2 family of inexpensive preprogrammed medium Programmable Array Logic devices (PALs) have been developed to provide an easy interface between various microprocessors and the DP84XX family of DRAM controller/drivers. These PALs interface to all the necessary control signals of the particular processor and the DP8429. The PAL controls the operation of the DP8429 in modes 5 and 1, while meeting all the critical timing considerations discussed above. The refresh clock, RFCK, may be divided down from the processor clock using an IC counter such as the DM74LS393 or the DP84300 programmable refresh timer. The DP84300 can provide RFCK periods ranging from 15.4 μs to 15.6 μs based on an input clock of 2 to 10 MHz. *Figure 8* shows a general block diagram for a system using the DP8429 in modes 1 and 5. *Figure 9* shows possible timing diagrams for such a system (using $\overline{\text{WAIT}}$ to prohibit access when refreshing). Although the DP84XX2 PALs are offered as standard peripheral devices for the DP84XX DRAM controller/drivers, the programming equations for these devices are provided so the user may make minor modifications for unique system requirements.

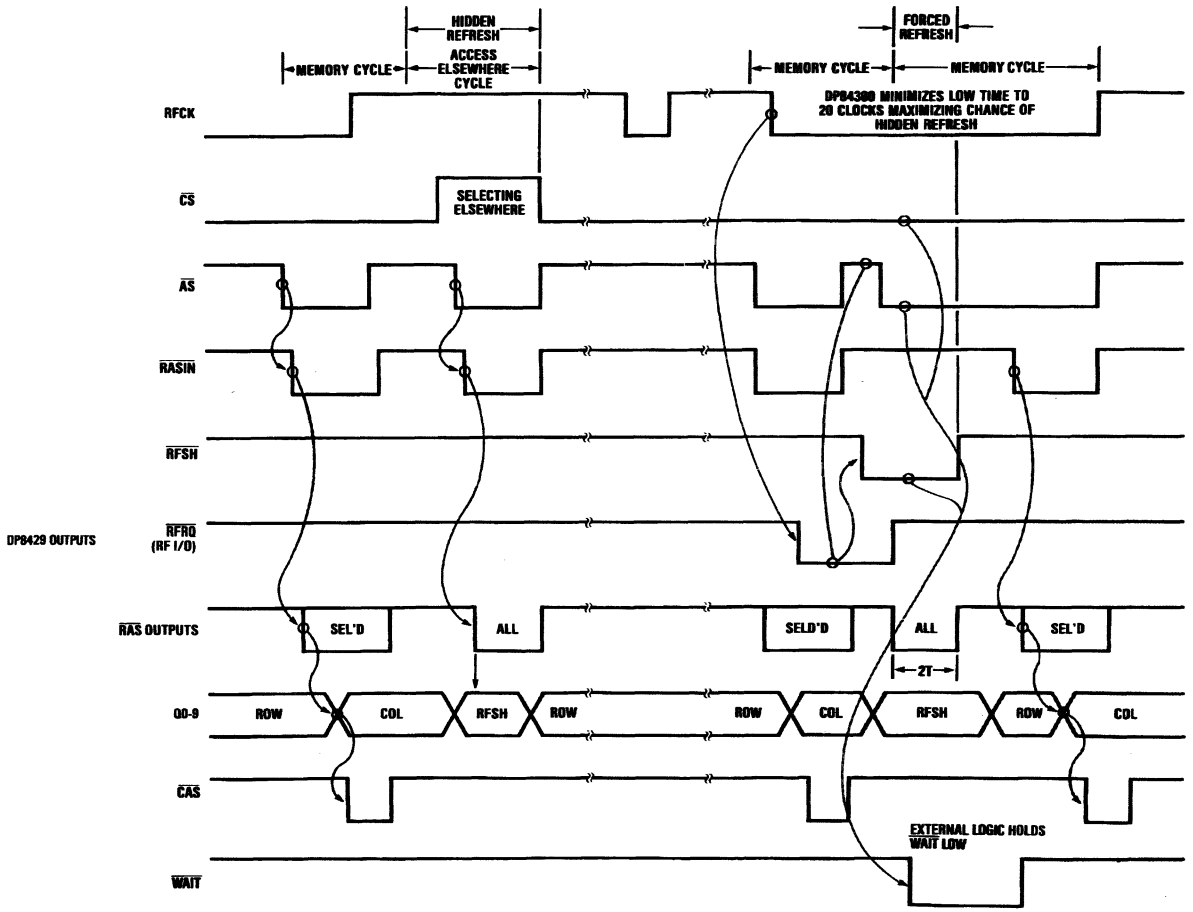
ADVANTAGES OF DP8429 OVER A DISCRETE DYNAMIC RAM CONTROLLER

1) The DP8429 system solution takes up much less board space because everything is on one chip (latches, refresh counter, control logic, multiplexers, drivers, and internal delay lines).
2) Less effort is needed to design a memory system. The DP8429 has automatic modes (1 and 5) which require a minimum of external control logic. Also programmable array logic devices (PALs) have been designed which allow an easy interface to most popular microprocessors (Motorola 68000 family, National Semiconductor 32032 family, Intel 8086 family, and the Zilog Z8000 family).
3) Less skew in memory timing parameters because all critical components are on one chip (many discrete drivers specify a minimum on-chip skew under worst-case conditions, but this cannot be used if more then one driver is needed, such as would be the case in driving a large dynamic RAM array).
4) Our switching characteristics give the designer the critical timing specifications based on TTL output levels (low = 0.8V, high = 2.4V) at a specified load capacitance. All timing parameters are specified on the DP8429:
   A) driving 88 DRAM's over a temperature range of 0–70 degrees centigrade (no extra drivers are needed).
   B) under worst-case driving conditions with all outputs switching simultaneously (most discrete drivers on the market specify worst-case conditions with only one output switching at a time; this is not a true worst-case condition!).



TL/F/8649–26

FIGURE 8. Connecting the DP8429 Between the 16-bit Microprocessor and Memory



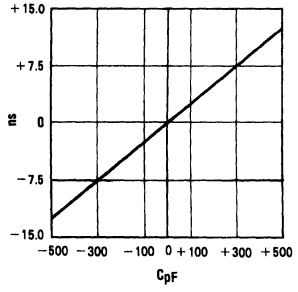
TL/F/8649–27

*T is microprocessor's clock period

FIGURE 9. DP8429 Auto Refresh, Access with $\overline{\text{WAIT}}$ States

Switching Characteristics

All A. C. parameters are specified with the equivalent load capacitances, including traces, of 88 DRAMs organized as 4 banks of 22 DRAMs each. Maximums are based on worst-case conditions including all outputs switching simultaneously. This, in many cases, results in the AC valves shown in the DP84XX DRAM controller data sheet being much looser than true worst case maximum AC delays. The system designer should estimate the DP8429 load in his/her application, and modify the appropriate A. C. parameters using the graph in *Figure 10*. Two example calculations are provided below.



TL/F/8649–28

FIGURE 10. Change in Propagation Delay relative to "true" (application) load minus AC specified data sheet load

Examples

1) A mode 4 user driving 2 banks of DRAM has the following loading conditions:

   $\overline{CAS}$ - 300 pF

   Q0–Q9 - 250 pF

   $\overline{RAS}$ - 150 pF

A.C. parameters should be adjusted in accordance with *Figure 10* and the specifications given for the 88 DRAM load as follows:

max t_{RPDL} = 20 ns − 0 ns = 20 ns (since $\overline{RAS}$ loading is the same as that which is spec'ed)

max t_{CPDL} = 32 ns − 7 ns = 25 ns

max t_{CCAS} = 46 ns − 7 ns = 39 ns

max t_{RCC} = 41 ns − 6 ns = 35 ns

min t_{RHA} is not significantly effected since it does not involve an output transition

Other parameters are adjusted in a similar manner.

2) A mode 5 user driving one bank of DRAM has the following loading conditions:

   $\overline{CAS}$ - 120 pF

   Q0–Q9 - 100 pF

   $\overline{RAS}$ - 120 pF

A. C. parameters should be adjusted as follows:

with RAHS = "1",

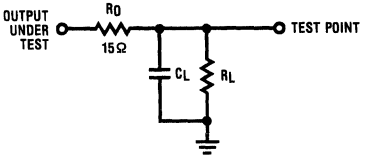
max t_{RICL} = 70 ns − 11 ns = 59 ns

max t_{RCDL} = 55 ns + 1 ns − 11 ns = 45 ns

(the + 1 ns is due to lighter $\overline{RAS}$ loading; the − 11 ns is due to lighter $\overline{CAS}$ loading)

min t_{RAH} = 15 ns + 1 ns = 16 ns

The additional 1 ns is due to the fact that the $\overline{RAS}$ line is driving less (switching faster) than the load to which the 15 ns spec applies. The row address will remain valid for about the same time irregardless of address loading since it is considered to be not valid at the beginning of its transition.



TL/F/8649–29

FIGURE 11. Output Load Circuit

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage, V_{CC}	7.0V
Storage Temperature Range	−65°C to +150°C
Input Voltage	5.5V
Output Current	150 mA
Lead Temp. (Soldering, 10 seconds)	300°C

Operating Conditions

		Min	Max	Units
V_{CC}	Supply Voltage	4.50	5.50	V
T_A	Ambient Temperature	0	+70	°C

Electrical Characteristics V_{CC} = 5.0V ±10%, 0°C ≤ T_A ≤ 70°C unless otherwise noted (Note 2)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_C	Input Clamp Voltage	V_{CC} = Min, I_C = − 12 mA		− 0.8	− 1.2	V
I_{IH}	Input High Current for all Inputs	V_{IN} = 2.5V		2.0	100	μA
I_I RSI	Output Load Current for RFI/O	V_{IN} = 0.5V, Output high		−0.7	−1.5	mA
I_{IL1}	Input Low Current for all Inputs**	V_{IN} = 0.5V		−0.02	−0.25	mA
I_{IL2}	ADS, R/$\overline{C}$, $\overline{CS}$, M2, $\overline{RASIN}$	V_{IN} = 0.5V		−0.05	−0.5	mA
V_{IL}	Input Low Threshold				0.8	V
V_{IH}	Input High Threshold		2.0			V
V_{OL1}	Output Low Voltage*	I_{OL} = 20 mA		0.3	0.5	V
V_{OL2}	Output Low Voltage for RFI/O	I_{OL} = 8 mA		0.3	0.5	V
V_{OH1}	Output High Voltage*	I_{OH} = − 1 mA	2.4	3.5		V
V_{OH2}	Output High Voltage for RFI/O	I_{OH} = − 100 μA	2.4	3.5		V
I_{1D}	Output High Drive Current*	V_{OUT} = 0.8V (Note 3)	−50	− 200		mA
I_{0D}	Output Low Drive Current*	V_{OUT} = 2.4V (Note 3)	50	200		mA
I_{CC}	Supply Current	V_{CC} = Max		150	240	mA

*Except RFI/O

**Except RFI/O, ADS, R/$\overline{C}$, $\overline{CS}$, M2, $\overline{RASIN}$

Switching Characteristics: DP8428 and DP8429

V_{CC} = 5.0V ± 10%, 0°C ≤ T_A ≤ 70°C unless otherwise noted (Notes 2, 4, 5), the output load capacitance is typical for 4 banks of 22 DRAMs each or 88 DRAMs, including trace capacitance.

*** These values are Q0-Q9, C_L = 500 pF; $\overline{RAS0}$–$\overline{RAS3}$, C_L = 150 pF; $\overline{WE}$, C_L = 500 pF; $\overline{CAS}$, C_L = 600 pF; RL = 500Ω unless otherwise noted. See *Figure 11* for test load. Maximum propagation delays are specified with all outputs switching.**

**** Preliminary**

Symbol	Access Parameter	Condition	*CL		**All C_L = 50 pF		Units
			Min	Max	Min	Max	
t_{RICL0}	$\overline{RASIN}$ to $\overline{CAS}$ Low Delay (RAHS = 0)	*Figure 6* DP8428-80/29-80	57	97	42	85	ns
t_{RICL0}	$\overline{RASIN}$ to $\overline{CAS}$ Low Delay (RAHS = 0)	*Figure 6* DP8428-70/29-70	57	87	42	75	ns
t_{RICL1}	$\overline{RASIN}$ to $\overline{CAS}$ Low Delay (RAHS = 1)	*Figure 6* DP8428-80/29-80	48	80	35	68	ns
t_{RICL1}	$\overline{RASIN}$ to $\overline{CAS}$ Low Delay (RAHS = 1)	*Figure 6* DP8428-70/29-70	48	70	35	58	ns
t_{RICH}	$\overline{RASIN}$ to $\overline{CAS}$ High Delay	*Figure 6*		37			ns
t_{RCDL0}	$\overline{RAS}$ to $\overline{CAS}$ Low Delay (RAHS = 0)	*Figure 6* DP8428-80/29-80	43	80			ns
t_{RCDL0}	$\overline{RAS}$ to $\overline{CAS}$ Low Delay (RAHS = 0)	*Figure 6* DP8428-70/29-70	43	72			ns

Switching Characteristics: DP8428 and DP8429 (Continued)

V_{CC} = 5.0V ± 10%, 0°C ≤ T_A ≤ 70°C unless otherwise noted (Notes 2, 4, 5), the output load capacitance is typical for 4 banks of 22 DRAMs each or 88 DRAMs, including trace capacitance.

*** These values are Q0-Q9, C_L = 500 pF; $\overline{RAS}$0–$\overline{RAS}$3, C_L = 150 pF; $\overline{WE}$, C_L = 500 pF; $\overline{CAS}$, C_L = 600 pF; RL = 500Ω unless otherwise noted. See *Figure 11* for test load. Maximum propagation delays are specified with all outputs switching.**

**** Preliminary**

Symbol	Access Parameter	Condition	*CL		**All C_L = 50 pF		Units
			Min	Max	Min	Max	
t_{RCDL1}	$\overline{RAS}$ to $\overline{CAS}$ Low Delay (RAHS = 1)	*Figure 6* DP8428-80/29-80	34	63			ns
t_{RCDL1}	$\overline{RAS}$ to $\overline{CAS}$ Low Delay (RAHS = 1)	*Figure 6* DP8428-70/29-70	34	55			ns
t_{RCDH}	$\overline{RAS}$ to $\overline{CAS}$ High Delay	*Figure 6*		22			ns
t_{RAH0}	Row Address Hold Time (RAHS = 0, Mode 5)	*Figure 6*	25		25		ns
t_{RAH1}	Row Address Hold Time (RAHS = 1, Mode 5)	*Figure 6*	15		15		ns
t_{ASC}	Column Address Set-up Time (Mode 5)	*Figure 6*	0		0		ns
t_{RCV0}	$\overline{RASIN}$ to Column Address Valid (RAHS = 0, Mode 5)	*Figure 6* DP8428-80/29-80		94			ns
t_{RCV0}	$\overline{RASIN}$ to Column Address Valid (RAHS = 0, Mode 5)	*Figure 6* DP8428-70/29-70		85			ns
t_{RCV1}	$\overline{RASIN}$ to Column Address Valid (RAHS = 1, Mode 5)	*Figure 6* DP8428-80/29-80		76			ns
t_{RCV1}	$\overline{RASIN}$ to Column Address Valid (RAHS = 1, Mode 5)	*Figure 6* DP8428-70/29-70		68			ns
t_{RPDL}	$\overline{RASIN}$ to $\overline{RAS}$ Low Delay	*Figures 5a, 5b, 6*		21		18	ns
t_{RPDH}	$\overline{RASIN}$ to $\overline{RAS}$ High Delay	*Figures 5a, 5b, 6*		20		17	ns
t_{ASRL}	Address Set-up to $\overline{RASIN}$ low	*Figures 5a, 5b, 6*	13				ns
t_{APD}	Address Input to Output Delay	*Figures 5a, 5b, 6*		36		25	ns
t_{SPD}	Address Strobe High to Address Output Valid	*Figures 5a, 5b*		48			ns
t_{ASA}	Address Set-up Time to ADS	*Figures 5a, 5b, 6*	5				ns
t_{AHA}	Address Hold Time from ADS	*Figures 5a, 5b, 6*	10				ns
t_{ADS}	Address Strobe Pulse Width	*Figures 5a, 5b, 6*	26				ns
t_{WPD}	$\overline{WIN}$ to $\overline{WE}$ Output Delay	*Figure 5b*		28			ns
t_{CPDL}	$\overline{CASIN}$ to $\overline{CAS}$ Low Delay (R/$\overline{C}$ low, Mode 4)	*Figure 5b*	17	33			ns
t_{CPDH}	$\overline{CASIN}$ to $\overline{CAS}$ High Delay (R/$\overline{C}$ low, Mode 4)	*Figure 5b*	13	33			ns
t_{CPdif}	t_{CPDL} - t_{CPDH}	See Mode 4 Description		13			ns
t_{RCC}	Column Select to Column Address Valid	*Figure 5a*		41			ns
t_{RCR}	Row Select to Row Address Valid	*Figures 5a, 5b*		45			ns
t_{RHA}	Row Address Held from Column Select	*Figure 5a*	7				ns
t_{CCAS}	R/$\overline{C}$ Low to $\overline{CAS}$ Low Delay ($\overline{CASIN}$ Low, Mode 4)	*Figure 5a* DP8428-80/29-80		50			ns
t_{CCAS}	R/$\overline{C}$ Low to $\overline{CAS}$ Low Delay ($\overline{CASIN}$ Low, Mode 4)	*Figure 5a* DP8428-70/29-70		46			ns
t_{DIF1}	Maximum (t_{RPDL} - t_{RHA})	See Mode 4 Description		7			ns
t_{DIF2}	Maximum (t_{RCC} - t_{CPDL})			13			ns

Switching Characteristics: DP8428 and DP8429 (Continued)

V_{CC} = 5.0V ± 10%, 0°C ≤ T_A ≤ 70°C unless otherwise noted (Notes 2, 4, 5). The output load capacitance is typical for 4 banks of 22 DRAMs each or 88 DRAMs, including trace capacitance.

*** These values are Q0-Q9, C_L = 500 pF; $\overline{RAS}$0-$\overline{RAS}$3, C_L = 150 pF; $\overline{WE}$, C_L = 500 pF; $\overline{CAS}$, C_L = 600 pF; RL = 500Ω unless otherwise noted. See *Figure 11* for test load. Maximum propagation delays are specified with all outputs switching.**

****Preliminary**

Symbol	Refresh Parameter	Condition	*CL		**All C_L = 50 pF		Units
			Min	Max	Min	Max	
t_{RC}	Refresh Cycle Period	*Figure 2a*	100				ns
$t_{RASINL,H}$	Pulse Width of $\overline{RASIN}$ during Refresh	*Figure 2a*	50				ns
t_{RFPDL0}	$\overline{RASIN}$ to $\overline{RAS}$ Low Delay during Refresh (Mode 0)	*Figure 2a*		28			ns
t_{RFPDL5}	$\overline{RASIN}$ to $\overline{RAS}$ Low Delay during Hidden Refresh	*Figure 7*		38			ns
t_{RFPDH0}	$\overline{RASIN}$ to $\overline{RAS}$ High Delay during Refresh (Mode 0)	*Figure 2a*		35			ns
t_{RFPDH5}	$\overline{RASIN}$ to $\overline{RAS}$ High Delay during Hidden Refresh	*Figure 7*		44			ns
t_{RFLCT}	$\overline{RFSH}$ Low to Counter Address Valid	*Figures 2a, 3* $\overline{CS}$ = X		38			ns
t_{RFLRL}	$\overline{RFSH}$ Low Set-up to $\overline{RASIN}$ Low (Mode 0), to get Minimum t_{ASR} = 0	*Figure 2a*	12				ns
t_{RFHRL}	$\overline{RFSH}$ High Setup to Access $\overline{RASIN}$ Low	*Figure 3*	25				ns
t_{RFHRV}	$\overline{RFSH}$ High to Row Address Valid	*Figure 3*		43			ns
t_{ROHNC}	$\overline{RAS}$ High to New Count Valid	*Figure 2a*		42			ns
t_{RST}	Counter Reset Pulse Width	*Figure 2a*	60				ns
t_{CTL}	RFI/O Low to Counter Outputs All Low	*Figure 2a*		100			ns
$t_{RFCKL,H}$	Minimum Pulse Width of RFCK	*Figure 7*	100				ns
T	Period of $\overline{RAS}$ Generator Clock	*Figure 3*	30				ns
t_{RGCKL}	Minimum Pulse Width Low of RGCK	*Figure 3*	15				ns
t_{RGCKH}	Minimum Pulse Width High of RGCK	*Figure 3*	15				ns
t_{FRQL}	RFCK Low to Forced $\overline{RFRQ}$ (RFI/O) Low	*Figure 3* C_L = 50 pF RL = 35k		66			ns
t_{FRQH}	RGCK Low to Forced $\overline{RFRQ}$ High	*Figure 3* C_L = 50 pF RL = 35k		55			ns

Switching Characteristics: DP8428 and DP8429 (Continued)

V_{CC} = 5.0V ± 10%, 0°C ≤ T_A ≤ 70°C unless otherwise noted (Notes 2, 4, 5). The output load capacitance is typical for 4 banks of 22 DRAMs each or 88 DRAMs, including trace capacitance.

*** These values are Q0-Q9, C_L = 500 pF; $\overline{RAS0}$–$\overline{RAS3}$, C_L = 150 pF; $\overline{WE}$, C_L = 500 pF; $\overline{CAS}$, C_L = 600 pF; RL = 500Ω unless otherwise noted. See *Figure 11* for test load. Maximum propagation delays are specified with all outputs switching.**

****Preliminary**

Symbol	Refresh Parameter	Condition	*CL		**All C_L = 50 pF		Units
			Min	Max	Min	Max	
t_{RGRL}	RGCK Low to $\overline{RAS}$ Low	*Figure 3*	20	41			ns
t_{RGRH}	RGCK Low to $\overline{RAS}$ High	*Figure 3*	20	48			ns
t_{RQHRF}	$\overline{RFSH}$ Hold Time from RGCK	*Figure 3*	2T				ns
t_{RFRH}	$\overline{RFSH}$ High to $\overline{RAS}$ High (Ending Forced Refresh early)	(See Mode 1 Description)		42			ns
t_{RFSRG}	$\overline{RFSH}$ Low Set-up to RGCK Low (Mode 1)	(See Mode 1 Description) *Figure 3*	12				ns
t_{CSHR}	$\overline{CS}$ High to $\overline{RASIN}$ Low for Hidden Refresh	*Figure 7*	10				ns
t_{CSRL1} for DP8429	$\overline{CS}$ Low to Access $\overline{RASIN}$ Low (Using Mode 5 with Auto Refresh Mode)	*Figure 3*	34				ns
t_{CSRL1} for DP8428	$\overline{CS}$ Low to Access $\overline{RASIN}$ Low (Using Mode 5 with Auto Refresh Mode)	*Figure 3*	5				ns
t_{CSRL0}	$\overline{CS}$ Low to Access $\overline{RASIN}$ Low (Using Modes 4 or 5 with externally controlled Refresh)	(See Mode 5 Description)	5				ns
t_{RKRL}	RFCK High to $\overline{RASIN}$ low for hidden Refresh		50				ns

Input Capacitance T_A = 25°C (Note 2)

Symbol	Parameter	Condition	Min	Typ	Max	Units
C_{IN}	Input Capacitance ADS, R/$\overline{C}$, $\overline{CS}$, M2, $\overline{RASIN}$			8		pF
C_{IN}	Input Capacitance All Other Inputs			5		pF

Note 1: "Absolute Maximum Ratings" are the values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: All typical values are for T_A = 25°C and V_{CC} = 5.0V.

Note 3: This test is provided as a monitor of Driver output source and sink current capability. Caution should be exercised in testing this parameter. In testing these parameters, a 15Ω resistor should be placed in series with each output under test. One output should be tested at a time and test time should not exceed 1 second.

Note 4: Input pulse 0V to 3.0V, $t_R = t_F$ = 2.5 ns, f = 2.5 MHz, t_{PW} = 200 ns. Input reference point on AC measurements is 1.5V Output reference points are 2.4V for High and 0.8V for Low.

Note 5: The load capacitance on RF I/O should not exceed 50 pF.

National Semiconductor

DP8420A/21A/22A microCMOS Programmable 256k/1M/4M Dynamic RAM Controller/Drivers

General Description

The DP8420A/21A/22A dynamic RAM controllers provide a low cost, single chip interface between dynamic RAM and all 8-, 16- and 32-bit systems. The DP8420A/21A/22A generate all the required access control signal timing for DRAMs. An on-chip refresh request clock is used to automatically refresh the DRAM array. Refreshes and accesses are arbitrated on chip. If necessary, a $\overline{WAIT}$ or $\overline{DTACK}$ output inserts wait states into system access cycles, including burst mode accesses. $\overline{RAS}$ low time during refreshes and $\overline{RAS}$ precharge time after refreshes and back to back accesses are guaranteed through the insertion of wait states. Separate on-chip precharge counters for each $\overline{RAS}$ output can be used for memory interleaving to avoid delayed back to back accesses because of precharge. An additional feature of the DP8422A is two access ports to simplify dual accessing. Arbitration among these ports and refresh is done on chip.

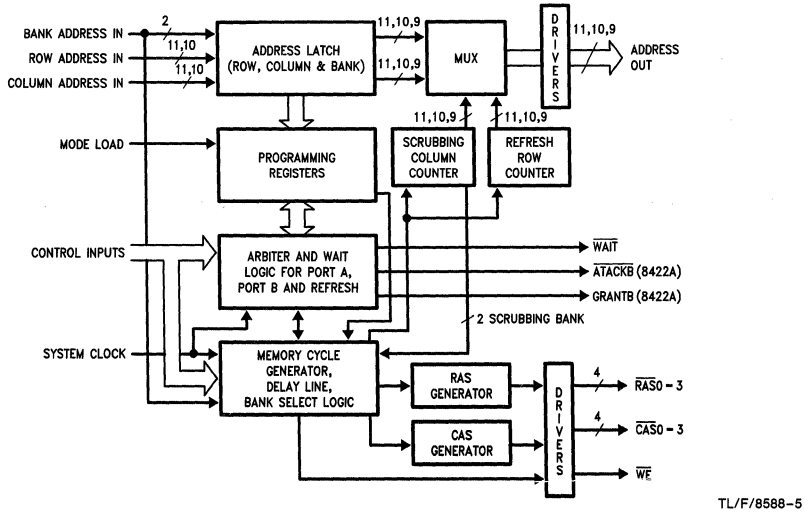
Features

- On chip high precision delay line to guarantee critical DRAM access timing parameters
- microCMOS process for low power
- High capacitance drivers for $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$ and DRAM address on chip
- On chip support for nibble, page and static column DRAMs
- Byte enable signals on chip allow byte writing in a word size up to 32 bits with no external logic
- Selection of controller speeds: 20 MHz and 25 MHz
- On board Port A/Port B (DP8422A only)/refresh arbitration logic
- Direct interface to all major microprocessors (application notes available)
- 4 $\overline{RAS}$ and 4 $\overline{CAS}$ drivers (the $\overline{RAS}$ and $\overline{CAS}$ configuration is programmable)

Control	# of Pins (PLCC)	# of Address Outputs	Largest DRAM Possible	Direct Drive Memory Capacity	Access Ports Available
DP8420A	68	9	256 kbit	4 Mbytes	Single Access Port
DP8421A	68	10	1 Mbit	16 Mbytes	Single Access Port
DP8422A	84	11	4 Mbit	64 Mbytes	Dual Access Ports (A and B)

Block Diagram

DP8420A/21A/22A DRAM Controller



TL/F/8588-5

FIGURE 1

Table of Contents

1.0 Introduction

The DP8420A/21A/22A are CMOS Dynamic RAM controllers that incorporate many advanced features including the capabilities of address latches, refresh counter, refresh clock, row, column and refresh address multiplexor, delay line, refresh/access arbitration logic and high capacitive drivers. The programmable system interface allows any manufacturer's microprocessor or bus to directly interface via the DP8420A/21A/22A to DRAM arrays up to 64 Mbytes in size.

After power up, the DP8420A/21A/22A must first be programmed before accessing the DRAM. The chip is programmed through the address bus.

There are two methods of programming the chip. The first method, mode load only, is accomplished by asserting the signal mode load, $\overline{ML}$. A valid programming selection is presented on the row, column, bank and $\overline{ECAS}$ inputs, then $\overline{ML}$ is negated. When $\overline{ML}$ is negated, the chip is programmed with the valid programming bits on the address bus.

The second method, chip selected access, is accomplished by asserting $\overline{ML}$ and performing a chip selected access. When $\overline{CS}$ and $\overline{AREQ}$ are asserted for the access, the chip is programmed. During this programming access, the programming bits affecting the wait logic become effective immediately, allowing the access to terminate. After the access, $\overline{ML}$ is negated and the rest of the programming bits take effect.

Once the DP8420A/21A/22A has been programmed, a 60 ms initialization period is entered. During this time, the DP8420A/21A/22A controllers perform refreshes to the DRAM array so further DRAM warm up cycles are unnecessary.

The DP8420A/21A/22A can now be used to access the DRAM. There are two modes of accessing with the controller. The two modes are Mode 0, which initiates $\overline{RAS}$ synchronously, and Mode 1, which initiates $\overline{RAS}$ asynchronously.

To access the DRAM using Mode 0, the signal ALE is asserted along with $\overline{CS}$ to ensure a valid VRAM access. ALE asserting sets an internal latch and only needs to be pulsed and not held throughout the entire access. On the next rising clock edge, $\overline{RAS}$ will be asserted for that access. The DP8420A/21A/22A will place the row address on the DRAM address bus, guarantee the programmed value of row address hold time of the DRAM, place the column address on the DRAM address bus, guarantee the programmed value of column address setup time and assert $\overline{CAS}$. $\overline{AREQ}$ can be asserted anytime after the clock edge which starts the access $\overline{RAS}$. $\overline{RAS}$ and $\overline{CAS}$ will extend until $\overline{AREQ}$ is negated.

The other access mode, Mode 1, is asynchronous to the clock. When $\overline{ADS}$ is asserted, $\overline{RAS}$ is asserted. The DP8420A/21A/22A will place the row address on the DRAM address bus, guarantee the programmed value of row address hold time, place the column address on the DRAM address bus, guarantee the programmed value of column address setup time and assert $\overline{CAS}$. $\overline{AREQ}$ can be tied to $\overline{ADS}$ or can be asserted after $\overline{ADS}$ is asserted. $\overline{AREQ}$ negated will terminate the access.

The DP8420A/21A/22A have greatly expanded refresh capabilities compared to other DRAM controllers. There are three modes of refreshing available. These modes are internal automatic refreshing, externally controlled/burst refreshing, and refresh request/acknowledge refreshing. Any of these modes can be used together or separately to achieve the desired results.

When using internal automatic refreshing, the DP8420A/21A/22A will generate an internal refresh request from the refresh request clock. The DP8420A/21A/22A will arbitrate between the refresh requests and accesses. Assuming an access is not currently in progress, the DP8420A/21A/22A will assert the signal $\overline{RFIP}$. On the next positive clock edge, refreshing will begin. If an access had been in progress, the refresh will begin after the access has terminated.

To use externally controlled/burst refresh, the user disables the internal refresh request by asserting the input $\overline{DISRFRSH}$. A refresh can now be externally requested by asserting the input $\overline{RFSH}$. The DP8420A/21A/22A will arbitrate between the external refresh request and accesses. Assuming an access is not currently in progress, the DP8420A/21A/22A will assert the output $\overline{RFIP}$. On the next positive clock edge, refreshing will begin. If an access had been in progress, the refresh would take place after the access has terminated.

With refresh request/acknowledge mode, the DP8420A/21A/22A broadcasts the internal refresh request to the system through the $\overline{RFRQ}$ output pin. External circuitry can determine when to refresh the DRAM through the $\overline{RFSH}$ input.

The controllers have three types of refreshing available: conventional, staggered and error scrubbing. Any refresh control mode can be used with any type of refresh. In a conventional refresh, all of the $\overline{RAS}$ outputs will be asserted and negated at once. In a staggered refresh, the $\overline{RAS}$ outputs will be asserted one positive clock edge apart. Error scrubbing is the same as conventional refresh except that a $\overline{CAS}$ will be asserted during a refresh allowing the system to run that data through an EDAC chip and write it back to memory, if a single bit error has occurred. The refreshes can be extended with the EXTEND REFRESH input, EXTNDRF.

The DP8420A/21A/22A have wait support available as $\overline{DTACK}$ or $\overline{WAIT}$. Both are programmable. $\overline{DTACK}$, Data Transfer ACKnowledge, is useful for processors whose wait signal is active high. $\overline{WAIT}$ is useful for processors whose wait signal is active low. The user can choose either at programming. These signals are used by the on-chip arbitor to insert wait states to guarantee the arbitration between accesses and refreshes or precharge. Both signals are independent of the access mode chosen.

$\overline{DTACK}$ will assert a programmed number of clock edges from the event that starts the access $\overline{RAS}$. $\overline{DTACK}$ will be negated, when the access is terminated, by $\overline{AREQ}$ being negated. $\overline{DTACK}$ can also be programmed to toggle with the $\overline{ECAS}$ inputs during burst/page mode accesses.

$\overline{WAIT}$ is asserted during the start of the access (ALE and $\overline{CS}$, or $\overline{ADS}$ and $\overline{CS}$) and will negate a number of clock edges from the event that starts the access $\overline{RAS}$. After $\overline{WAIT}$ is negated, it will stay negated until the next access. $\overline{WAIT}$ can also be programmed to toggle with $\overline{ECAS}$ inputs during a burst/page mode access.

Both signals can be dynamically delayed further through the $\overline{WAITIN}$ signal to the DP8420A/21A/22A.

The DP8420A/21A/22A have address latches, used to latch the bank, row and column address inputs. Once the address is latched, a column increment feature can be used to increment the column address. The address latches can also be programmed to be fall through.

1.0 Introduction (Continued)

The $\overline{RAS}$ and $\overline{CAS}$ drivers can be configured to drive a one, two or four bank memory array up to 32 bits in width. The $\overline{ECAS}$ signals can then be used to select one of four $\overline{CAS}$ drivers for byte writing with no external logic.

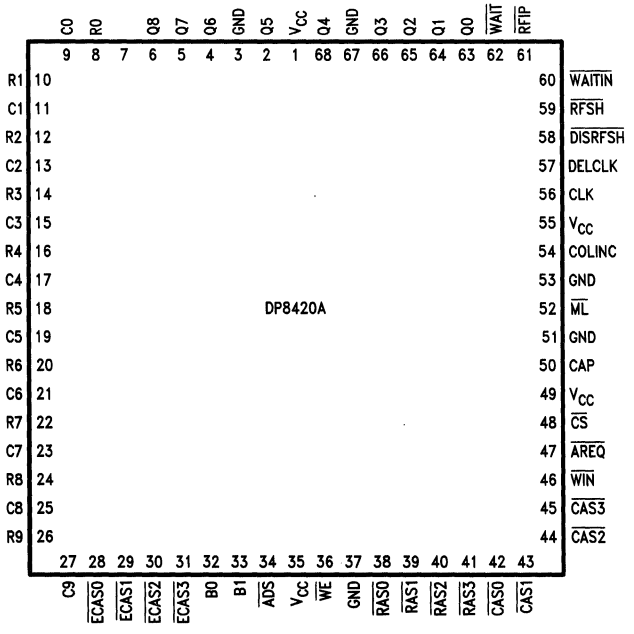
When configuring the DP8420A/21A/22A for more than one bank, memory interleaving can be used. By tying the low order address bits to the bank select lines, B0 and B1, sequential back to back accesses will not be delayed since the DP8420A/21A/22A have separate precharge counters per bank. The DP8420A/21A/22A are capable of performing address pipelining. In address pipelining, the DP8420A/21A/22A guarantee the column address hold time and switch the internal multiplexor to place the row address on the address bus. At this time, another memory access to another bank can be initiated.

The DP8422A has all the features previously mentioned. Unlike the DP8420A/21A, the DP8422A has a second port to allow a second CPU to access the memory array. This port, Port B, has two control signals to allow a CPU to access the DRAM array. These signals are access request for Port B, $\overline{AREQB}$, and Advanced Transfer ACKnowledge for Port B, $\overline{ATACKB}$. Two other signals are used by both Port A and Port B for dual accessing purposes. The signals are lock, $\overline{LOCK}$ and grant Port B, GRANTB. All arbitration for the two ports and refresh is done on-chip by the DP8422A through the insertion of wait states. Since the DP8422A has only one input address bus, the address lines have to be multiplexed externally. The signal GRANTB can be used for this purpose since it is asserted when Port B has access to the DRAM array and negated when Port A has access to the DRAM array. Once a port has access to the array, the other port can be "locked out" by asserting the input $\overline{LOCK}$. $\overline{AREQB}$, when asserted, is used by Port B to request an access. $\overline{ATACKB}$, when asserted, signifies that access $\overline{RAS}$ has been asserted for the requested Port B access. By using $\overline{ATACKB}$, the user can generate an appropriate $\overline{WAIT}$ or $\overline{DTACK}$ like signal for the Port B CPU.

The following explains the terminology used in this data sheet. The terms negated and asserted are used. Asserted refers to a "true" signal. Thus, "$\overline{ECAS0}$ asserted" means the $\overline{ECAS0}$ input is at a logic 0. The term "COLINC asserted" means the COLINC input is at a logic 1. The term negated refers to a "false" signal. Thus, "$\overline{ECAS0}$ negated" means the $\overline{ECAS0}$ input is at a logic 1. The term "COLINC negated" means the input COLINC is at a logic 0. The table shown below clarifies this terminology.

Signal	Action	Logic Level
Active High	Asserted	High
Active High	Negated	Low
Active Low	Asserted	Low
Active Low	Negated	High

Connection Diagrams



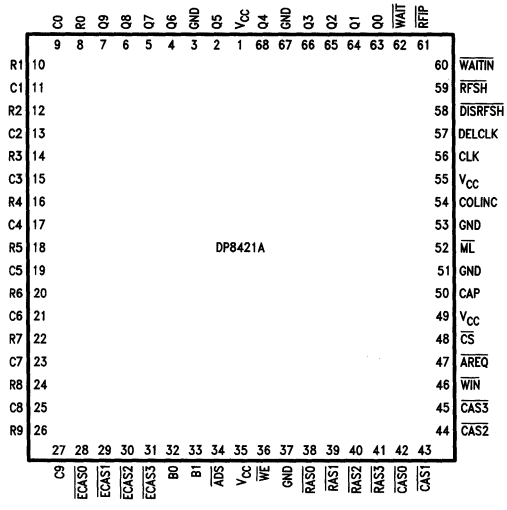
TL/F/8588–4

Top View

FIGURE 2

Order Number DP8420AV-20 or DP8420AV-25
See NS Package Number V68A

Connection Diagrams (Continued)

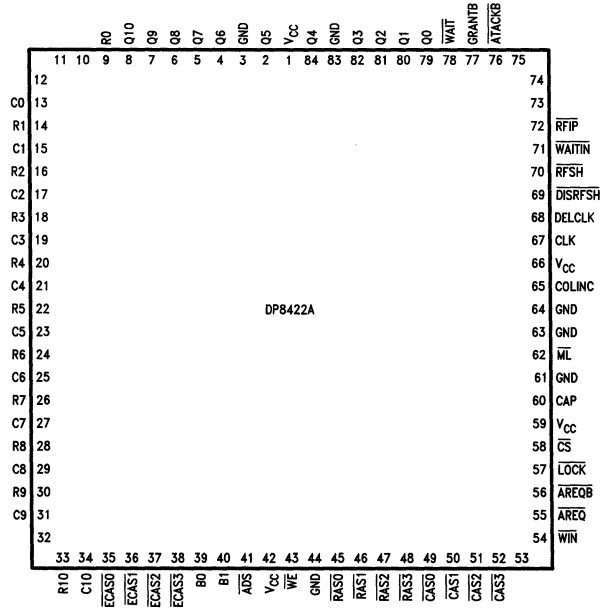


TL/F/8588–3

Top View

FIGURE 3

Order Number DP8421AV-20 or DP8421AV-25
See NS Package Number V68A



TL/F/8588–2

Top View

FIGURE 4

Order Number DP8422AV-20 or DP8422AV-25
See NS Package Number V84A

2.0 Signal Descriptions

Pin Name	Device (If not Applicable to All)	Input/ Output	Description
2.1 ADDRESS, R/W AND PROGRAMMING SIGNALS			
R0–10 R0–9	DP8422A DP8420A/21A	I I	**ROW ADDRESS:** These inputs are used to specify the row address during an access to the DRAM. They are also used to program the chip when $\overline{ML}$ is asserted (except R10).
C0–10 C0–9	DP8422A DP8420A/21A	I I	**COLUMN ADDRESS:** These inputs are used to specify the column address during an access to the DRAM. They are also used to program the chip when $\overline{ML}$ is asserted (except C10).
B0, B1		I	**BANK SELECT:** Depending on programming, these inputs are used to select a group of $\overline{RAS}$ and $\overline{CAS}$ outputs to assert during an access. They are also used to program the chip when $\overline{ML}$ is asserted.
$\overline{ECAS}$0–3		I	**ENABLE $\overline{CAS}$:** These inputs are used to enable a single or group of $\overline{CAS}$ outputs when asserted. In combination with the B0, B1 and the programming bits, these inputs select which $\overline{CAS}$ output or $\overline{CAS}$ outputs will assert during an access. The $\overline{ECAS}$ signals can also be used to toggle a group of $\overline{CAS}$ outputs for page/nibble mode accesses. They also can be used for byte write operations. If $\overline{ECAS}$0 is negated during programming, continuing to assert the $\overline{ECAS}$0 while negating $\overline{AREQ}$ or $\overline{AREQB}$ during an access, will cause the $\overline{CAS}$ outputs to be extended while the $\overline{RAS}$ outputs are negated (the $\overline{ECAS}$n inputs have no effect during scrubbing refreshes).
$\overline{WIN}$		I	**WRITE ENABLE IN:** This input is used to signify a write operation to the DRAM. If $\overline{ECAS}$0 is asserted during programming, the $\overline{WE}$ output will follow this input. This input asserted will also cause $\overline{CAS}$ to delay to the next positive clock edge if address bit C9 is asserted during programming.
COLINC (EXTNDRF)		I I	**COLUMN INCREMENT:** When the address latches are used, and $\overline{RFIP}$ is negated, this input functions as COLINC. Asserting this signal causes the column address to be incremented by one. When $\overline{RFIP}$ is asserted, this signal is used to extend the refresh cycle by any number of periods of CLK until it is negated.
$\overline{ML}$		I	**MODE LOAD:** This input signal, when low, enables the internal programming register that stores the programming information.
2.2 DRAM CONTROL SIGNALS			
Q0–10 Q0–9 Q0–8	DP8422A DP8421A DP8421A	O O O	**DRAM ADDRESS:** These outputs are the multiplexed output of the R0–9, 10 and C0–9, 10 and form the DRAM address bus. These outputs contain the refresh address whenever $\overline{RFIP}$ is asserted. They contain high capacitive drivers with 20Ω series damping resistors.
$\overline{RAS}$0–3		O	**ROW ADDRESS STROBES:** These outputs are asserted to latch the row address contained on the outputs Q0–8, 9, 10 into the DRAM. When $\overline{RFIP}$ is asserted, the $\overline{RAS}$ outputs are used to latch the refresh row address contained on the Q0–8, 9, 10 outputs in the DRAM. These outputs contain high capacitive drivers with 20Ω series damping resistors.
$\overline{CAS}$0–3		O	**COLUMN ADDRESS STROBES:** These outputs are asserted to latch the column address contained on the outputs Q0–8, 9, 10 into the DRAM. These outputs have high capacitive drivers with 20Ω series damping resistors.
$\overline{WE}$ ($\overline{RFRQ}$)		O O	**WRITE ENABLE** or **REFRESH REQUEST:** This output asserted specifies a write operation to the DRAM. When negated, this output specifies a read operation to the DRAM. When the DP8420A/21A/22A is programmed in interleave mode or when $\overline{ECAS}$0 is negated during programming, this output will function as $\overline{RFRQ}$. When asserted, this pin specifies that 13 μs or 15 μs have passed. If $\overline{DISRFSH}$ is negated, the DP8420A/21A/22A will perform an internal refresh as soon as possible. If $\overline{DISRFRSH}$ is asserted, $\overline{RFRQ}$ can be used to externally request a refresh through the input $\overline{RFSH}$. This output has a high capacitive driver and a 20Ω series damping resistor.

2.0 Signal Descriptions (Continued)

Pin Name	Device (If not Applicable to All)	Input/ Output	Description
2.3 REFRESH SIGNALS			
$\overline{RFIP}$		O	**REFRESH IN PROGRESS:** This output is asserted prior to a refresh cycle and is negated when all the $\overline{RAS}$ outputs are negated for that refresh.
$\overline{RFSH}$		I	**REFRESH:** This input asserted with $\overline{DISRFRSH}$ already asserted will request a refresh. If this input is continually asserted, the DP8420A/21A/22A will perform refresh cycles in a burst refresh fashion until the input is negated. If $\overline{RFSH}$ is asserted with $\overline{DISRFSH}$ negated, the internal refresh address counter is cleared (useful for burst refreshes).
$\overline{DISRFSH}$		I	**DISABLE REFRESH:** This input is used to disable internal refreshes and must be asserted when using $\overline{RFSH}$ for externally requested refreshes.
2.4 PORT A ACCESS			
$\overline{ADS}$ (ALE)		I I	**ADDRESS STROBE** or **ADDRESS LATCH ENABLE:** Depending on programming, this input can function as $\overline{ADS}$ or ALE. In mode 0, the input functions as ALE and when asserted along with $\overline{CS}$ causes an internal latch to be set. Once this latch is set an access will start from the positive clock edge of CLK as soon as possible. In Mode 1, the input functions as $\overline{ADS}$ and when asserted along with $\overline{CS}$, causes the access $\overline{RAS}$ to assert if no other event is taking place. If an event is taking place, $\overline{RAS}$ will be asserted from the positive edge of CLK as soon as possible. In both cases, the low going edge of this signal latches the bank, row and column address if programmed to do so.
$\overline{CS}$		I	**CHIP SELECT:** This input signal must be asserted to enable a Port A access.
$\overline{AREQ}$		I	**ACCESS REQUEST:** This input signal in Mode 0 must be asserted some time after the first positive clock edge after ALE has been asserted. When this signal is negated, $\overline{RAS}$ is negated for the access. In Mode 1, this signal must be asserted before $\overline{ADS}$ can be negated. When this signal is negated, $\overline{RAS}$ is negated for the access.
$\overline{WAIT}$ ($\overline{DTACK}$)		O O	**WAIT** or **DTACK:** This output can be programmed to insert wait states into a CPU access cycle. With R7 negated during programming, the output will function as a $\overline{WAIT}$ type output. In this case, the output will be active low to signal a wait condition. With R7 asserted during programming, the output will function as $\overline{DTACK}$. In this case, the output will be negated to signify a wait condition and will be asserted to signify the access has taken place. Each of these signals can be delayed by a number of positive clock edges or negative clock levels of CLK to increase the microprocessor's access cycle through the insertion of wait states.
$\overline{WAITIN}$		I	**WAIT INCREASE:** This input can be used to dynamically increase the number of positive clock edges of CLK until $\overline{DTACK}$ will be asserted or $\overline{WAIT}$ will be negated during a DRAM access.

2.0 Signal Descriptions (Continued)

Pin Name	Device (If not Applicable to All)	Input/ Output	Description
2.5 PORT B ACCESS SIGNALS			
$\overline{AREQB}$	DP8422A only	I	**PORT B ACCESS REQUEST:** This input asserted will latch the row, column and bank address if programmed, and requests an access to take place for Port B. If the access can take place, $\overline{RAS}$ will assert immediately. If the access has to be delayed, $\overline{RAS}$ will assert as soon as possible from a positive edge of CLK.
$\overline{ATACKB}$	DP8422A only	O	**ADVANCED TRANSFER ACKNOWLEDGE PORT B:** This output is asserted when the access $\overline{RAS}$ is asserted for a Port B access. This signal can be used to generate the appropriate $\overline{DTACK}$ or $\overline{WAIT}$ type signal for Port B's CPU or bus.
2.6 COMMON DUAL PORT SIGNALS			
GRANTB	DP8422A only	O	**GRANT B:** This output indicates which port is currently granted access to the DRAM array. When GRANTB is asserted, Port B has access to the array. When GRANTB is negated, Port A has access to the DRAM array. This signal is used to multiplex the signals R0–8, 9, 10; C0–8, 9, 10; B0–1; $\overline{WIN}$; $\overline{LOCK}$ and $\overline{ECAS}$0–3 to the DP8422A when using dual accessing.
$\overline{LOCK}$	DP8422A only	I	**LOCK:** This input can be used by the currently granted port to "lock out" the other port from the DRAM array by inserting wait states into the locked out port's access cycle until $\overline{LOCK}$ is negated.
2.7 POWER SIGNALS AND CAPACITOR INPUT			
V_{CC}		I	**POWER:** Supply Voltage.
GND		I	**GROUND:** Supply Voltage Reference.
CAP		I	**CAPACITOR:** This input is used by the internal PLL for stabilization. The value of the ceramic capacitor should be 0.1 μF and should be connected between this input and ground.

2.8 CLOCK INPUTS

There are two clock inputs to the DP8420A/21A/22A, CLK and DELCLK. These two clocks may both be tied to the same clock input, or they may be two separate clocks, running at different frequencies, asynchronous to each other.

CLK		I	**SYSTEM CLOCK:** This input may be in the range of 0 Hz up to 25 MHz. This input is generally a constant frequency but it may be controlled externally to change frequencies or perhaps be stopped for some arbitrary period of time. This input provides the clock to the internal state machine that arbitrates between accesses and refreshes. This clock's positive edges and negative levels are used to extend the $\overline{WAIT}$ ($\overline{DTACK}$) signals. Ths clock is also used as the reference for the $\overline{RAS}$ precharge time and $\overline{RAS}$ low time during refresh. All Port A and Port B accesses are assumed to be synchronous to the system clock CLK.
DELCLK		I	**DELAY LINE CLOCK:** The clock input DELCLK, may be in the range of 6 MHz to 20 MHz and should be a multiple of 2 (i.e., 6, 8, 10, 12, 14, 16, 18, 20 MHz) to have the DP8420A/21A/22A switching characteristics hold. If DELCLK is not one of the above frequencies the accuracy of the internal delay line will suffer. This is because the phase locked loop that generates the delay line assumes an input clock frequency of a multiple of 2 MHz. For example, if the DELCLK input is at 7 MHz and we choose a divide by 3 (program bits C0–2) this will produce 2.333 MHz which is 16.667% off of 2 MHz. Therefore, the DP8420A/21A/22A delay line would produce delays that are shorter (faster delays) than what is intended. If divide by 4 was chosen the delay line would be longer (slower delays) than intended (1.75 MHz instead of 2 MHz). (See Section 10 for more information.) This clock is also divided to create the internal refresh clock.

3.0 Port A Access Modes

The DP8420A/21A/22A have two general purpose access modes. With one of these modes, any microprocessor can be interfaced to DRAM. A Port A access to DRAM is initiated by two input signals: $\overline{ADS}$ (ALE) and $\overline{CS}$. The access is always terminated by one signal: $\overline{AREQ}$. These input signals should be synchronous to the input clock, CLK. One of these access modes is selected at programming through the B1 input signal. In both modes, once an access has been requested by $\overline{CS}$ and $\overline{ADS}$ (ALE), the DP8422A will guarantee the following:

The DP8420A/21A/22A will have the row address valid to the DRAMs' address bus, Q0–8, 9, 10 given that the row address setup time to the DP8420A/21A/22A was met;

The DP8420A/21A/22A will bring the appropriate $\overline{RAS}$ or $\overline{RAS}$s low;

The DP8420A/21A/22A will guarantee the minimum row address hold time, before switching the internal multiplexor to place the column address on the DRAM address bus, Q0–8, 9, 10;

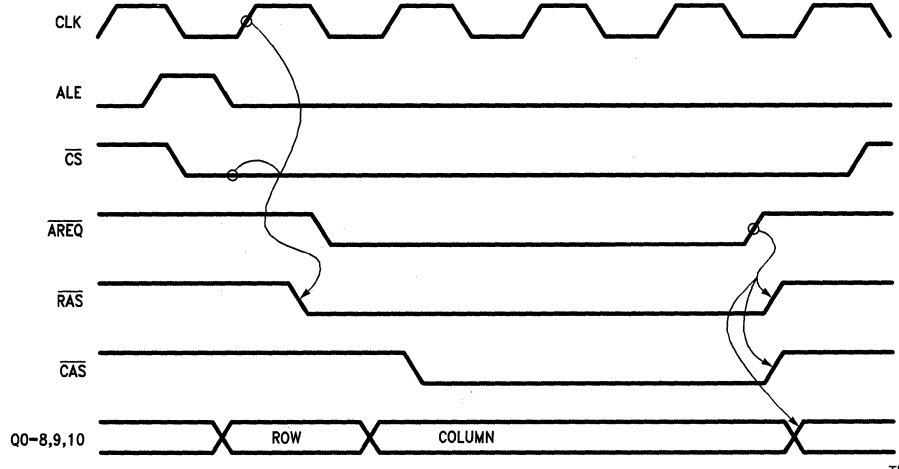
The DP8420A/21A/22A will guarantee the minimum column address setup time before asserting the appropriate $\overline{CAS}$ or $\overline{CAS}$s;

The DP8420A/21A/22A will hold the column address valid the minimum specified column address hold time in address pipelining mode and will hold the column address valid the remainder of the access in non-pipelining mode.

3.1 ACCESS MODE 0

Access Mode 0, shown in *Figure 5a,* is selected by negating the input B1 during programming. This access mode allows accesses to DRAM to always be initiated from the positive edge of the system input clock, CLK. To initiate a Mode 0 access, ALE is pulsed high and $\overline{CS}$ is asserted. Pulsing ALE high and asserting $\overline{CS}$, sets an internal latch which requests an access. If the precharge time from the last access or DRAM refresh had been met and a refresh of DRAM or a Port B access was not in progress, the $\overline{RAS}$ or group of $\overline{RAS}$s would be initiated from the first positive edge of CLK. If a DRAM refresh is in progress or precharge time is required, the controller will wait until these events have taken place and assert $\overline{RAS}$ on the next positive edge of CLK.

Sometime after the first positive edge of CLK after ALE and $\overline{CS}$ have been asserted, the input $\overline{AREQ}$ must be asserted. In single port applications, once $\overline{AREQ}$ has been asserted, $\overline{CS}$ can be negated. Once $\overline{AREQ}$ is negated, $\overline{RAS}$ and $\overline{DTACK}$, if programmed, will be negated. If $\overline{ECAS0}$ is asserted during programming, $\overline{CAS}$ will be negated with $\overline{AREQ}$. If $\overline{ECAS0}$ was negated during programming, a single $\overline{CAS}$ or group of $\overline{CAS}$s will continue to be asserted after $\overline{RAS}$ has been negated given that the appropriate $\overline{ECAS}$s inputs were



TL/F/8588–60

FIGURE 5a. Access Mode 0

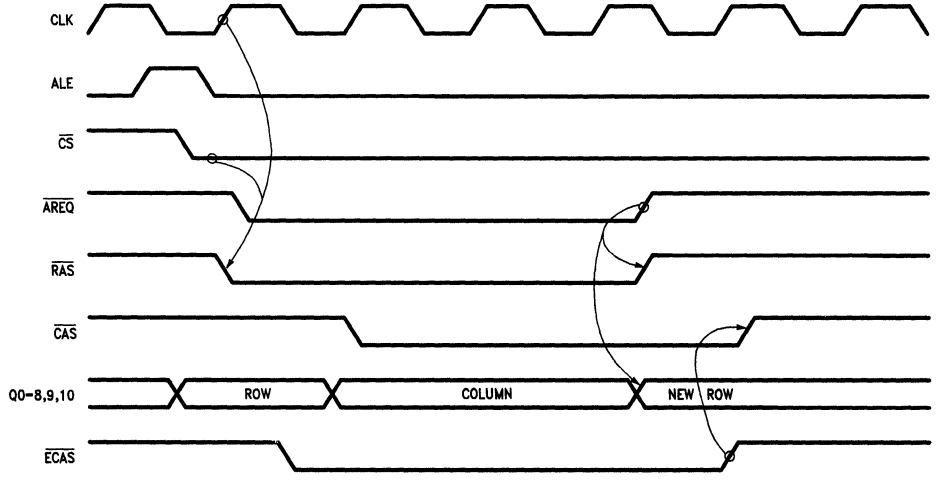
3.0 Port A Access Modes (Continued)

asserted as shown in *Figure 5b*. This allows the DRAM to have data present on the data out bus while gaining $\overline{RAS}$ precharge time. ALE can stay asserted several periods of CLK. However, ALE must be negated before or during the period of CLK in which $\overline{AREQ}$ is negated.

When performing address pipelining, the ALE input cannot be asserted to start another access until $\overline{AREQ}$ has been asserted for at least one clock period of CLK for the present access.

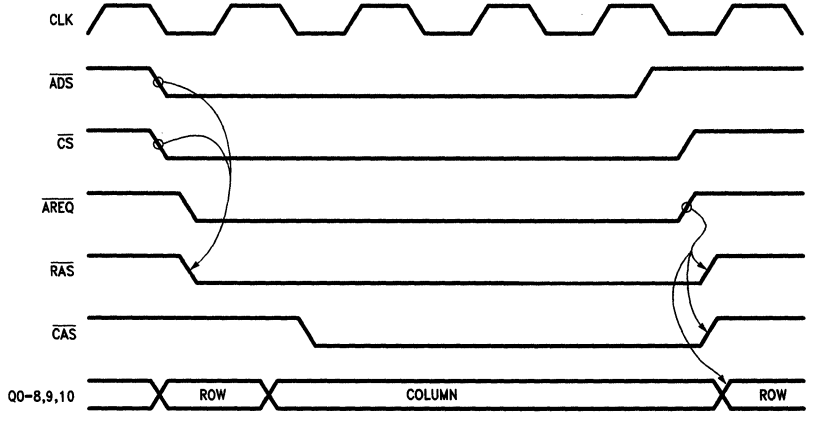
3.2 ACCESS MODE 1

Access Mode 1, shown in *Figure 6a*, is selected by asserting the input B1 during programming. This mode allows accesses, which are not delayed by precharge, Port B access or refresh, to start immediately from the access request input, $\overline{ADS}$. To initiate a Mode 1 access, $\overline{CS}$ is asserted followed by $\overline{ADS}$ asserted. If the programmed precharge time from the last access or DRAM refresh had been met and a refresh of the DRAM or Port B access to the DRAM was not in progress, the $\overline{RAS}$ or group of $\overline{RAS}$s selected by programming and the bank select inputs would be asserted from $\overline{ADS}$ being asserted. If a DRAM refresh or Port B access is in progress or precharge time is required, the controller will wait until these events have taken place and assert $\overline{RAS}$ or the group of $\overline{RAS}$s from the next positive edge of CLK.



TL/F/8588–61

FIGURE 5b. Access Mode 0 Extending $\overline{CAS}$



TL/F/8588–62

FIGURE 6a. Access Mode 1

3.0 Port A Access Modes (Continued)

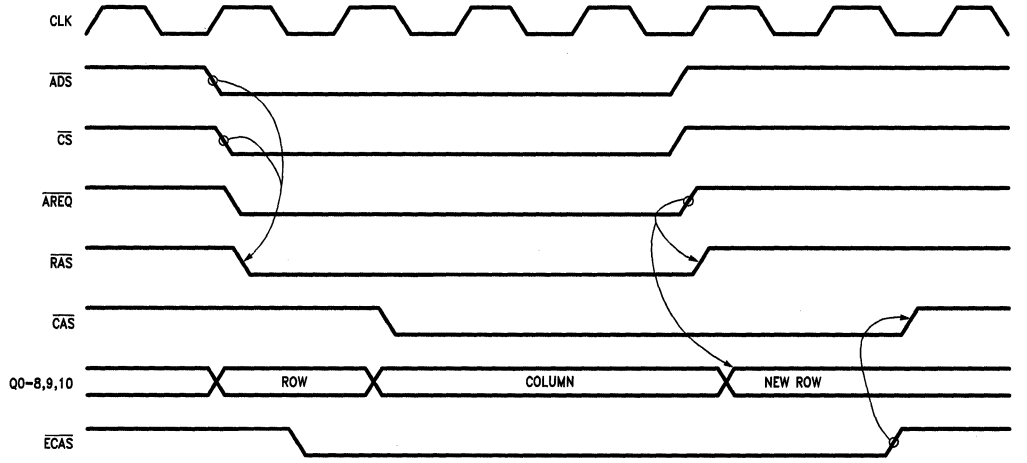
When $\overline{ADS}$ is asserted or sometime after, $\overline{AREQ}$ must be asserted. At this time, $\overline{ADS}$ can be negated and $\overline{AREQ}$ will continue the access. Once $\overline{AREQ}$ is negated, $\overline{RAS}$ and $\overline{DTACK}$, if programmed, will be negated. If $\overline{ECAS}$0 was asserted during programming, $\overline{CAS}$ will be negated with $\overline{AREQ}$. If $\overline{ECAS}$0 was negated during programming, a single $\overline{CAS}$ or group of $\overline{CAS}$s will continue to be asserted after $\overline{RAS}$ has been negated given that the appropriate $\overline{ECAS}$ inputs were asserted as shown in *Figure 6b*. This allows a DRAM to have data present on the data out bus while gaining $\overline{RAS}$ precharge time. $\overline{ADS}$ can continue to be asserted after $\overline{AREQ}$ has been asserted and negated, however a new access would not be started until $\overline{ADS}$ is negated and asserted again. $\overline{ADS}$ and $\overline{AREQ}$ can be tied together in applications not using address pipelining.

If address pipelining is programmed, it is possible for $\overline{ADS}$ to be negated after $\overline{AREQ}$ is asserted. Once $\overline{AREQ}$ is asserted, $\overline{ADS}$ can be asserted again to initiate a new access.

3.3 READ-MODIFY-WRITE CYCLES WITH EITHER ACCESS MODE

There are 2 methods by which this chip can be used to do read-modify-write access cycles. The first method involves doing a late write access where the $\overline{WIN}$ input is asserted some delay after $\overline{CAS}$ is asserted. The second method involves doing a page mode read access followed by a page mode write access with $\overline{RAS}$ held low (see *Figure 5*).

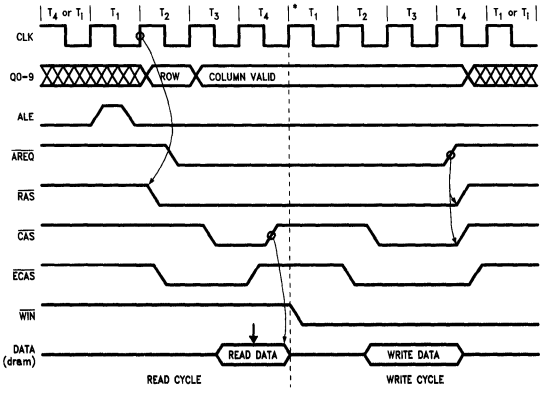
$\overline{CAS}$n must be toggled using the $\overline{ECAS}$n inputs and $\overline{WIN}$ has to be changed from negated to asserted (read to write) while $\overline{CAS}$ is negated. This method is better than changing $\overline{WIN}$ from negated to asserted in a late write access because here a problem may arise with DATA IN and DATA OUT being valid at the same time. This may result in a data line trying to drive two different levels simultaneously. The page mode method of a read-modify-write access allows the user to have transceivers in the system because the data in (read data) is guaranteed to be high impedance during the time the data out (write data) is valid.



TL/F/8588–63

FIGURE 6b. Access Mode 1 Extending $\overline{CAS}$

3.0 Port A Access Modes (Continued)



TL/F/8588-G2

*There may be idle states inserted here by the CPU.

FIGURE 6c. Read-Modify-Write Access Cycle

4.0 Refresh Options

The DP8420A/21A/22A support a wide variety of refresh control mode options including automatic internally controlled refresh, externally controlled/burst refresh, refresh request/acknowledge and any combination of the above. With each of the control modes above, different types of refreshes can be performed. These different types include all $\overline{RAS}$ refresh, staggered refresh and error scrubbing during all $\overline{RAS}$ refresh.

There are three inputs, EXTNDRF, $\overline{RFSH}$ and $\overline{DISRFSH}$, and two outputs, $\overline{RFIP}$ and $\overline{RFRQ}$, associated with refresh. There are also ten programming bits; R0–1, R9, C0–6 and $\overline{ECAS}$0 used to program the various types of refreshing.

The two inputs, $\overline{RFSH}$ and $\overline{DISRFSH}$, are used in the externally controlled/burst refresh mode and the refresh request/acknowledge mode. The output $\overline{RFRQ}$ is used in the refresh requset/acknowledge mode. The input EXTNDRF and the output $\overline{RFIP}$ are used in all refresh modes. Asserting the input EXTNDRF, extends the refresh cycle single or multiple integral clock periods of CLK. The output $\overline{RFIP}$ is asserted one period of CLK before the first refresh $\overline{RAS}$ is asserted. If an access is currently in progress, $\overline{RFIP}$ will be asserted up to one period of CLK before the first refresh $\overline{RAS}$, once $\overline{AREQ}$ or $\overline{AREQB}$ is negated for the access (see *Figure 7a*).

The DP8420A/21A/22A will increment the refresh address counter automatically, independent of the refresh mode used. The refresh address counter will be incremented once all the refresh $\overline{RAS}$s have been negated.

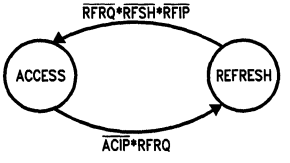
In every combination of refresh control mode and refresh type, the DP8420A/21A/22A is programmed to keep $\overline{RAS}$ asserted a number of CLK periods. The values of $\overline{RAS}$ low time during refresh are programmed with the programming bits R0 and R1.

4.1 REFRESH CONTROL MODES

There are three different modes of refresh control. Any of these modes can be used in combination or singularly to produce the desired refresh results. The three different modes of control are: automatic internal refresh, external/burst refresh and refresh request/acknowledge.

4.1.1. Automatic Internal Refresh

The DP8420A/21A/22A have an internal refresh clock. The period of the refresh clock is generated from the programming bits C0–3. Every period of the refresh clock, an internal refresh request is generated. As long as a DRAM access is not currently in progress and precharge time has been met, the internal refresh request will generate an automatic internal refresh. If a DRAM access is in progress, the DP8420A/21A/22A on-chip arbitration logic will wait until the access is finished before performing the refresh. The refresh/access arbitration logic can insert a refresh cycle between two address pipelined accesses. However, the refresh arbitration logic can not interrupt an access cycle to perform a refresh. To enable automatic internally controlled refreshes, the input $\overline{DISRFSH}$ must be negated.



TL/F/8588-F8

Explanation of Terms

RFRQ = ReFresh ReQuest internal to the DP8420A/21A/22A. RFRQ has the ability to hold off a pending access.

RFSH = Externally requested ReFreSH

RFIP = ReFresh In Progress

ACIP = Port A or Port B (DP8422A only) ACcess In Progress. This means that either RAS is low for an access or is in the process of transitioning low for an access.

FIGURE 7a. DP8420A/21A/22A Access/Refresh Arbitration State Program

4.1.2 Externally Controlled/Burst Refresh

To use externally controlled/burst refresh, the user must disable the automatic internally controlled refreshes by asserting the input $\overline{DISRFSH}$. The user is responsible for generating the refresh request by asserting the input $\overline{RFSH}$. Pulsing $\overline{RFSH}$ low, sets an internal latch, that is used to

4.0 Refresh Options (Continued)

produce the internal refresh request. The refresh cycle will take place on the next positive edge of CLK as shown in *Figure 7b*. If an access to DRAM is in progress or precharge time for the last access has not been met, the refresh will be delayed. Since pulsing $\overline{RFSH}$ low sets a latch, the user does not have to keep $\overline{RFSH}$ low until the refresh starts. When the last refresh $\overline{RAS}$ negates, the internal refresh request latch is cleared.

By keeping $\overline{RFSH}$ asserted past the positive edge of CLK which ends the refresh cycle as shown in *Figure 8*, the user will perform another refresh cycle. Using this technique, the user can perform a burst refresh consisting of any number of refresh cycles. Each refresh cycle during a burst refresh will meet the refresh $\overline{RAS}$ low time and the $\overline{RAS}$ precharge time (programming bits R0–1).

If the user desires to burst refresh the entire DRAM (all row addresses) he could generate an end of count signal (burst refresh finished) by looking at one of the DP8420A/21A/22A high address outputs (Q7, Q8, Q9 or Q10) and the $\overline{RFIP}$ output. The Qn outputs function as a decode of how many row addresses have been refreshed (Q7 = 128 refreshes, Q8 = 256 refreshes, Q9 = 512 refreshes, Q10 = 1024 refreshes).

4.1.3 Refresh Request/Acknowledge

The DP8420A/21A/22A can be programmed to output internal refresh requests. When the user programs $\overline{ECAS0}$ negated during programming and/or address pipelining mode, the $\overline{WE}$ output functions as $\overline{RFRQ}$. $\overline{RFRQ}$ will be asserted by one of two events, either the internal refresh clock has expired which signals that another refresh is needed, or by the signal $\overline{RFSH}$ being pulsed low requesting an external refresh. $\overline{RFRQ}$ will be asserted from a positive edge of CLK. *Figure 9a* shows an example of an external refresh being requested while $\overline{RFRQ}$ is negated. When $\overline{RFRQ}$ is asserted from the expiration of the internal refresh clock signaling a new refresh is needed, it will stay asserted until the $\overline{RFSH}$ is pulsed low with $\overline{DISRFSH}$ asserted. This will cause an externally requested/burst refresh to take place. If $\overline{DISRFSH}$ is negated, an automatic internal refresh will take place as shown in *Figure 9b*.

$\overline{RFRQ}$ will go high and then assert if additional periods of the internal refresh clock have expired and neither an externally controlled refresh nor an automatically controlled internal refresh have taken place as shown in *Figure 9c*. If a time critical event, or long access like page/static column mode access can not be interrupted, $\overline{RFRQ}$ pulsing high can be used to increment a counter. The counter can be used to perform a burst refresh of the number of refreshes missed (through the $\overline{RFSH}$ input).

4.2 REFRESH CYCLE TYPES

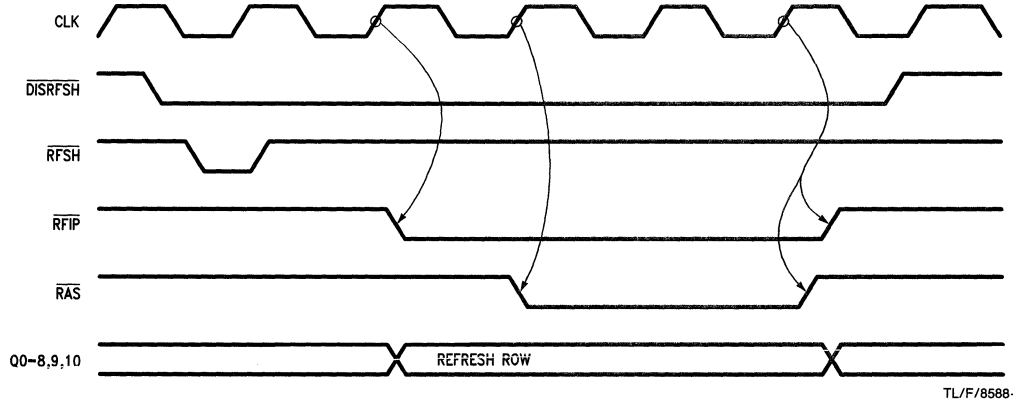
Three different types of refresh cycles are available for use. The three different types are mutually exclusive and can be used with any of the three modes of refresh control. The three different refresh cycle types are: all $\overline{RAS}$ refresh, staggered $\overline{RAS}$ refresh and error scrubbing during all $\overline{RAS}$ refresh. In all refresh cycle types, the $\overline{RAS}$ precharge time is guaranteed: between the previous access $\overline{RAS}$ ending and the refresh $\overline{RAS0}$ starting; between refresh $\overline{RAS3}$ ending and access $\overline{RAS}$ beginning; between burst refresh $\overline{RAS}$s.

4.2.1 Conventional $\overline{RAS}$ Refresh

A conventional refresh cycle causes $\overline{RAS0}$–3 to all assert from the first positive edge of CLK after $\overline{RFIP}$ is asserted as shown in *Figure 10*. $\overline{RAS0}$–3 will stay asserted until the number of positive edges of CLK programmed have passed. On the last positive edge, $\overline{RAS0}$–3, and $\overline{RFIP}$ will be negated. This type of refresh cycle is programmed by negating address bit R9 during programming.

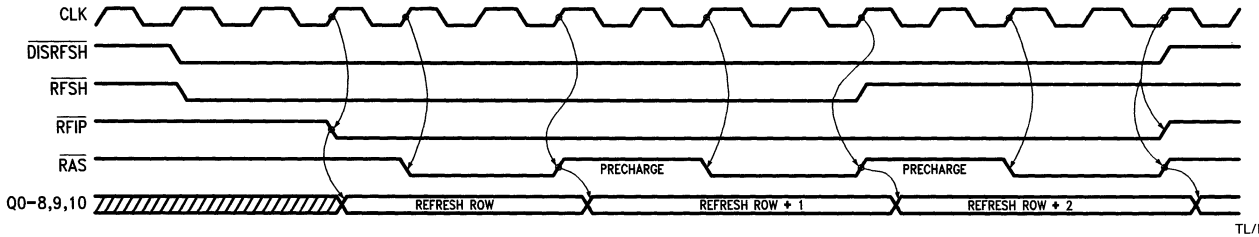
4.2.2 Staggered $\overline{RAS}$ Refresh

A staggered refresh staggers each $\overline{RAS}$ or group of $\overline{RAS}$s by a positive edge of CLK as shown in *Figure 11*. The number of $\overline{RAS}$s, which will be asserted on each positive edge of CLK, is determined by the $\overline{RAS}$, $\overline{CAS}$ configuration mode programming bits C4–C6. If single $\overline{RAS}$ outputs are selected during programming, then each $\overline{RAS}$ will assert on successive positive edges of CLK. If two $\overline{RAS}$ outputs are selected during programming then $\overline{RAS0}$ and $\overline{RAS1}$ will assert on the first positive edge of CLK after $\overline{RFIP}$ is asserted. $\overline{RAS2}$ and $\overline{RAS3}$ will assert on the second positive edge of CLK after $\overline{RFIP}$ is asserted. If all $\overline{RAS}$ outputs were selected during programming, all $\overline{RAS}$ outputs would assert on the first positive edge of CLK after $\overline{RFIP}$ is asserted. Each $\overline{RAS}$ or group of $\overline{RAS}$s will meet the programmed $\overline{RAS}$ low time and then negate.



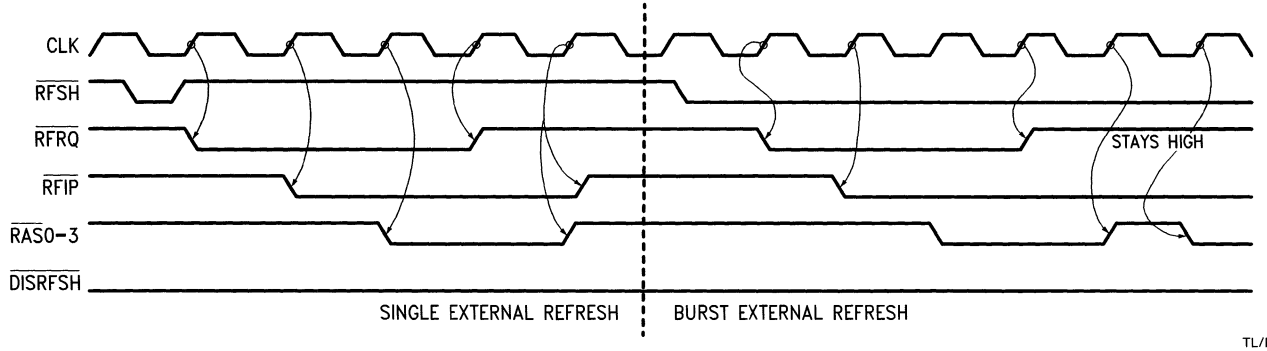
TL/F/8588–64

FIGURE 7b. Single External Refreshes (2 Periods of $\overline{RAS}$ Low during Refresh Programmed)



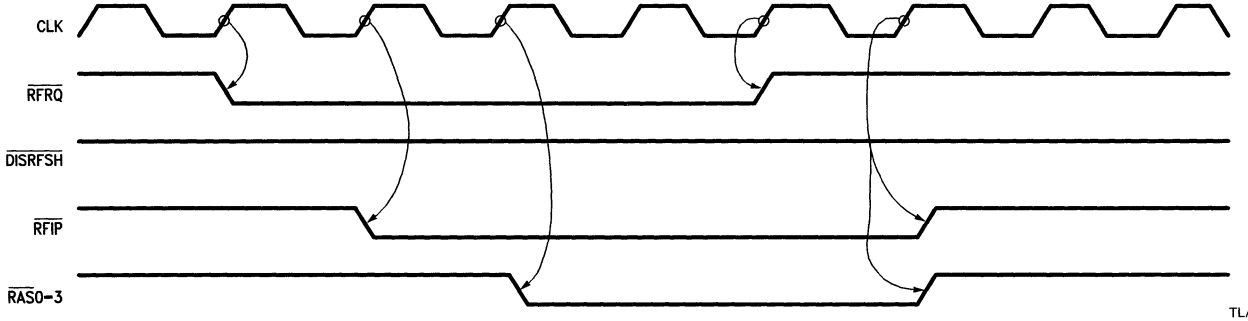
TL/F/8588–65

FIGURE 8. External Burst Refresh (2 Periods of $\overline{RAS}$ Precharge, 2 Periods of Refresh $\overline{RAS}$ Low during Refresh Programmed)



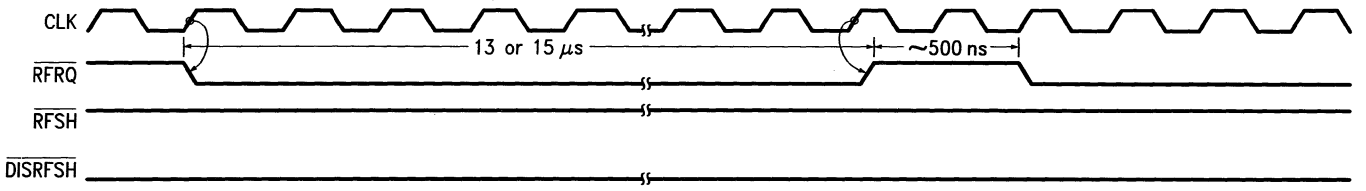
TL/F/8588–66

FIGURE 9a. Externally Controlled Single and Burst Refresh with Refresh Request ($\overline{RFRQ}$) Output (2 Periods of $\overline{RAS}$ Low during Refresh Programmed)



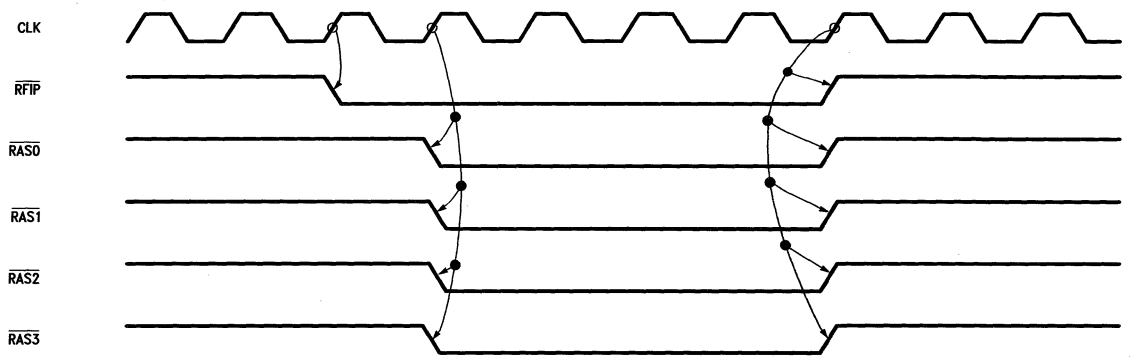
TL/F/8588–67

FIGURE 9b. Automatic Internal Refresh with Refresh Request (3T of $\overline{RAS}$ low during refresh programmed).



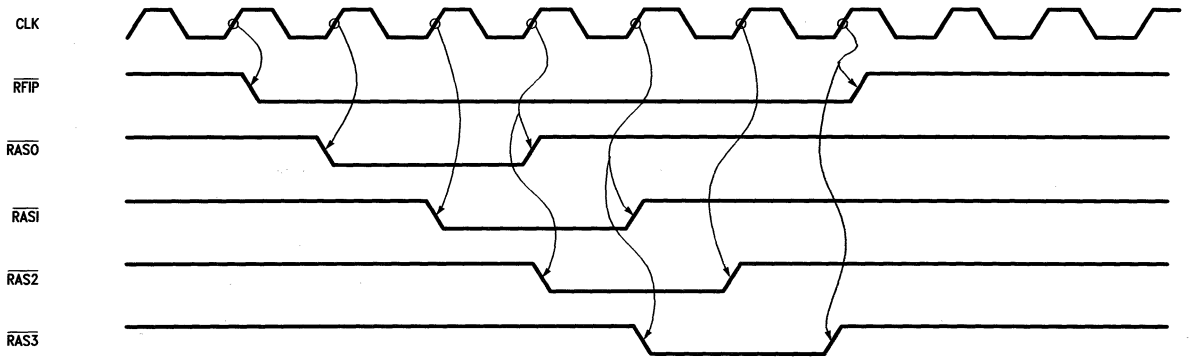
TL/F/8588–68

FIGURE 9c. Refresh Request Timing



TL/F/8588–69

FIGURE 10. Conventional $\overline{RAS}$ Refresh



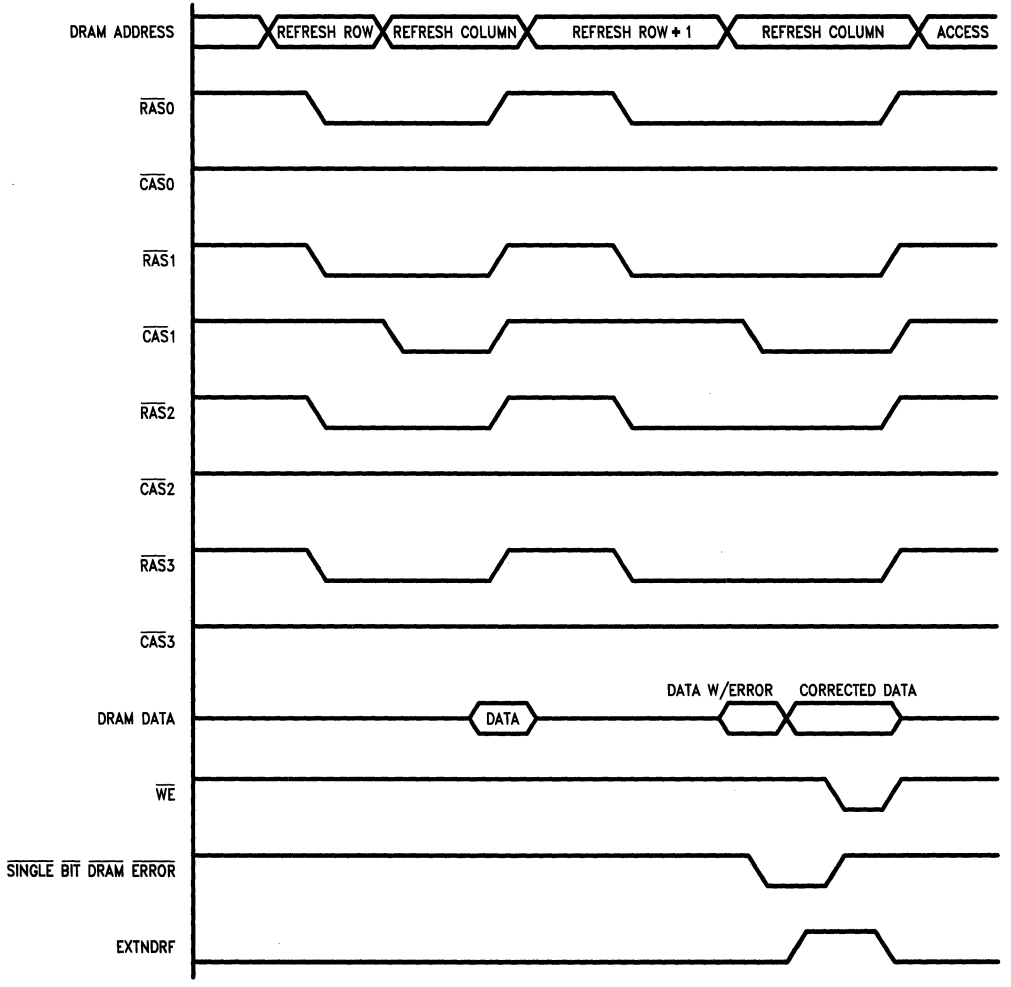
TL/F/8588–70

FIGURE 11. Staggered $\overline{RAS}$ Refresh

4.0 Refresh Options (Continued)

4.2.3 Error Scrubbing during Refresh

The DP8420A/21A/22A support error scrubbing during all $\overline{RAS}$ DRAM refreshes. Error scrubbing during refresh is selected through bits C4–C6 with bit R9 negated during programming. Error scrubbing can not be used with staggered refresh (see Section 9.0). Error scrubbing during refresh allows a $\overline{CAS}$ or group of $\overline{CAS}$s to assert during the all $\overline{RAS}$ refresh as shown in *Figure 12*. This allows data to be read from the DRAM array and passed through an Error Detection And Correction Chip, EDAC. If the EDAC determines that the data contains a single bit error and corrects that error, the refresh cycle can be extended with the input extend refresh, EXTNDRF, and a read-modify-write operation can be performed by asserting $\overline{WE}$. It is the responsibility of the designer to ensure that $\overline{WE}$ is negated. The DP8422A has a 24-bit internal refresh address counter that contains the 11 row, 11 column and 2 bank addresses. The DP8420A/21A have a 22-bit internal refresh address counter that contains the 10 row, 10 column and 2 bank addresses. These counters are configured as bank, column, row with the row address as the least significant bits. The bank counter bits are then used with the programming selection to determine which $\overline{CAS}$ or group of $\overline{CAS}$s will assert during a refresh.



TL/F/8588–46

FIGURE 12. Error Scrubbing during Refresh

4.0 Refresh Options (Continued)

4.3 EXTENDING REFRESH

The programmed number of periods of CLK that refresh $\overline{\text{RAS}}$s are asserted can be extended by one or multiple periods of CLK. Only the all $\overline{\text{RAS}}$ (with or without error scrubbing) type of refresh can be extended. To extend a refresh cycle, the input extend refresh, EXTNDRF, must be asserted before the positive edge of CLK that would have negated all the $\overline{\text{RAS}}$ outputs during the refresh cycle and after the positive edge of CLK which starts all $\overline{\text{RAS}}$ outputs during the refresh as shown in *Figure 13*. This will extend the refresh to the next positive edge of CLK and EXTNDRF will be sampled again. The refresh cycle will continue until EXTNDRF is sampled low on a positive edge of CLK.

4.4 CLEARING THE REFRESH ADDRESS COUNTER

The refresh address counter can be cleared by asserting $\overline{\text{RFSH}}$ while $\overline{\text{DISRFSH}}$ is negated as shown in *Figure 14a*. This can be used prior to a burst refresh of the entire memory array. By asserting $\overline{\text{RFSH}}$ one period of CLK before $\overline{\text{DISRFSH}}$ is asserted and then keeping both inputs asserted, the DP8420A/21A/22A will clear the refresh address counter and then perform refresh cycles separated by the programmed value of precharge as shown in *Figure 14b*. An end-of-count signal can be generated from the Q DRAM address outputs of the DP8420A/21A/22A and used to negate $\overline{\text{RFSH}}$.

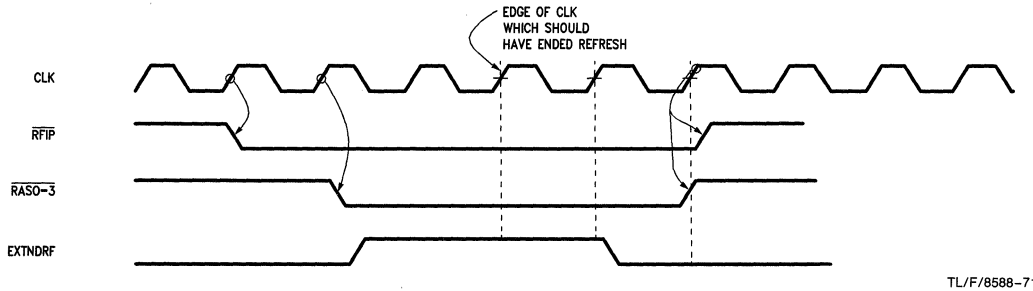


FIGURE 13. Extending Refresh with the Extend Refresh (EXTNDRF) Input

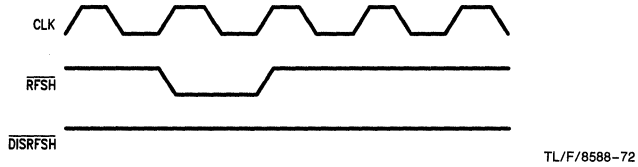


FIGURE 14a. Clearing the Refresh Address Counter

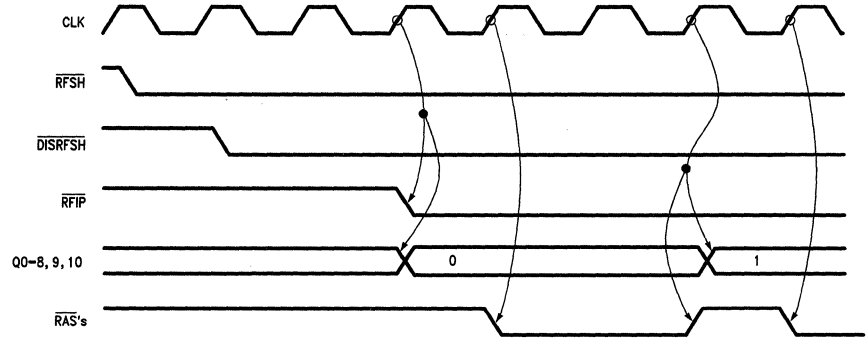


FIGURE 14b. Clearing the Refresh Counter during Burst

4.0 Refresh Options (Continued)

4.5 CLEARING THE REFRESH REQUEST CLOCK

The refresh request clock can be cleared by negating $\overline{\text{DISRFSH}}$ and asserting $\overline{\text{RFSH}}$ for 500 ns, one period of the internal 2 MHz clock as shown in *Figure 15*. By clearing the refresh request clock, the user is guaranteed that an internal refresh request will not be generated for approximately 15 μs, one refresh clock period, from the time $\overline{\text{RFSH}}$ is negated. This action will also clear the refresh address counter.

5.0 Port A Wait State Support

Wait states allow a CPU's access cycle to be increased by one or multiple CPU clock periods. By increasing the CPU's access cycle, all signals associated with that access cycle are extended. The wait or ready input is named differently by CPU manufacturers. However, any CPU's wait or ready input is compatible with either the $\overline{\text{WAIT}}$ or $\overline{\text{DTACK}}$ output of the DP8420A/21A/22A. The CPU samples a wait or ready line to determine if another clock period should be inserted into the access cycle. If another clock period is inserted, the CPU will continue to sample the input every CPU clock period until the input signal changes polarity, allowing the CPU access cycle to terminate. The user determines whether to program $\overline{\text{WAIT}}$ or $\overline{\text{DTACK}}$ (R7) and which value to select for $\overline{\text{WAIT}}$ or $\overline{\text{DTACK}}$ (R2, R3) depending upon the CPU used and where the CPU samples its wait input during an access cycle.

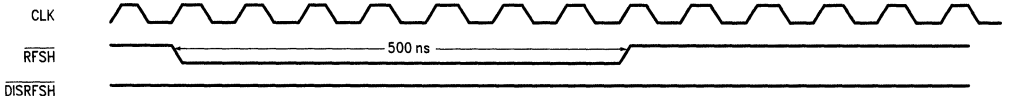
The decision to terminate the CPU access cycle is directly affected by the speed of the DRAMs used. The system designer must ensure that the data from the DRAMs will be present for the CPU to sample or that the data has been written to the DRAM before allowing the CPU access cycle to terminate.

The insertion of wait states also allows a CPU's access cycle to be extended until the DRAM access has taken place. The DP8420A/21A/22A insert wait states into CPU access cycles due to; guaranteeing precharge time, refresh currently in progress, user programmed wait states, the $\overline{\text{WAITIN}}$ signal being asserted and GRANTB not being valid (DP8422A only). If one of these events is taking place and the CPU starts an access, the DP8420A/21A/22A will insert wait states into the access cycle, thereby increasing the length of the CPU's access. Once the event has been completed, the DP8420A/21A/22A will allow the access to take place and stop inserting wait states.

There are six programming bits, R2–R7; an input, $\overline{\text{WAITIN}}$; and an output that functions as $\overline{\text{WAIT}}$ or $\overline{\text{DTACK}}$.

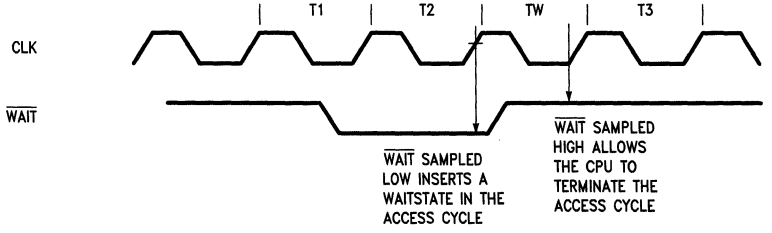
5.1 WAIT TYPE OUTPUT

With the R7 address bit negated during programming, the user selects the $\overline{\text{WAIT}}$ output. As long as $\overline{\text{WAIT}}$ is sampled asserted by the CPU, wait states (extra clock periods) are inserted into the current access cycle as shown in *Figure 16*. Once $\overline{\text{WAIT}}$ is sampled negated, the access cycle is completed by the CPU. $\overline{\text{WAIT}}$ is asserted at the beginning of a chip selected access and is programmed to negate a number of positive edges and/or negative levels of CLK from the event that starts the access. $\overline{\text{WAIT}}$ can also be programmed to function in page/burst mode applications. Once $\overline{\text{WAIT}}$ is negated during an access, and the $\overline{\text{ECAS}}$ inputs are negated with $\overline{\text{AREQ}}$ asserted, $\overline{\text{WAIT}}$ can be programmed to toggle, following the $\overline{\text{ECAS}}$ inputs. Once $\overline{\text{AREQ}}$ is negated, ending the access, $\overline{\text{WAIT}}$ will stay negated until the next chip selected access.



TL/F/8588–75

FIGURE 15. Clearing the Refresh Request Clock Counter



TL/F/8588–76

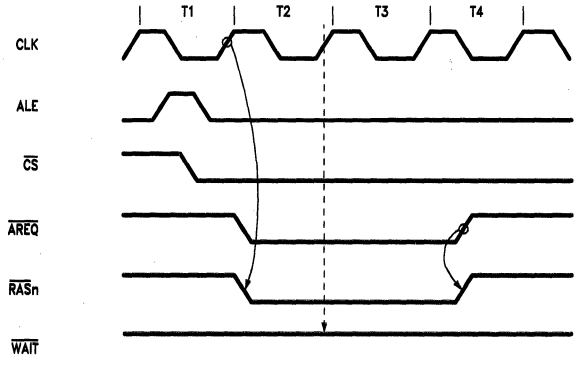
FIGURE 16. $\overline{\text{WAIT}}$ Type Output

5.0 Port A Wait State Support (Continued)

5.1.1. Wait during Single Accesses

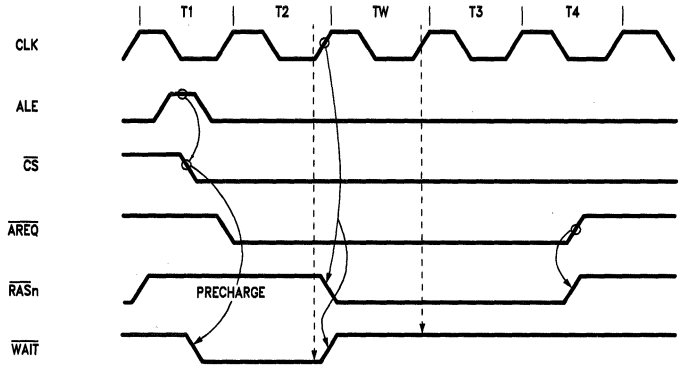
$\overline{WAIT}$ can be programmed to delay a number of positive edges and/or negative levels of CLK. These options are programmed through address bits R2 and R3 at programming time. The user is given four options described below.

0T during non delayed and delayed acceses: $\overline{WAIT}$ will stay negated during a non-delayed access as shown in *Figures 17a* and *17c*. During an access that is delayed, $\overline{WAIT}$ will assert at the start of the access ($\overline{CS}$ and ALE or $\overline{ADS}$) and negate from the positive edge of $\overline{CLK}$ that starts $\overline{RAS}$ for that access as shown in *Figures 17b* and *17d*.



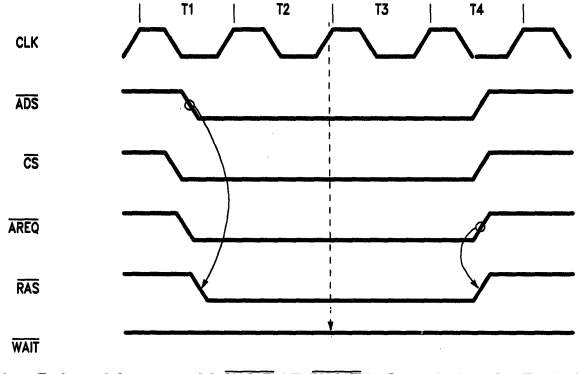
TL/F/8588-77

FIGURE 17a. Mode 0 Non-Delayed Access with $\overline{WAIT}$ 0T ($\overline{WAIT}$ is Sampled at the End of the "T2" Clock State)



TL/F/8588-78

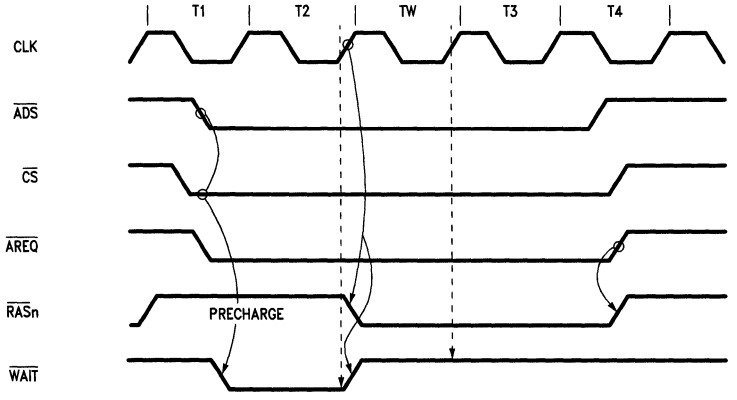
FIGURE 17b. Mode 0 Delayed Access with $\overline{WAIT}$ 0T ("2T" $\overline{RAS}$ Precharge, $\overline{WAIT}$ is Sampled at the End of the "T2" Clock State)



TL/F/8588-79

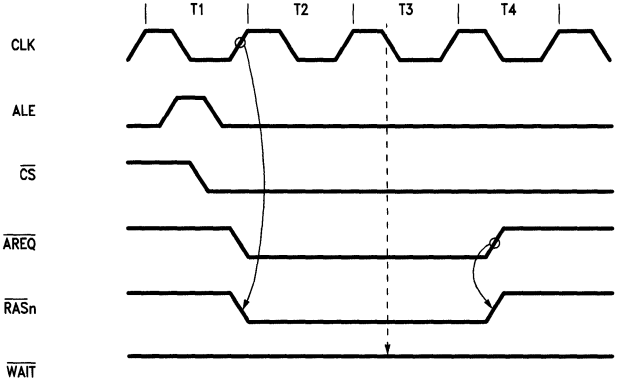
FIGURE 17c. Mode 1 Non-Delayed Access with $\overline{WAIT}$ 0T ($\overline{WAIT}$ is Sampled at the End of the "T2" Clock State)

5.0 Port A Wait State Support (Continued)



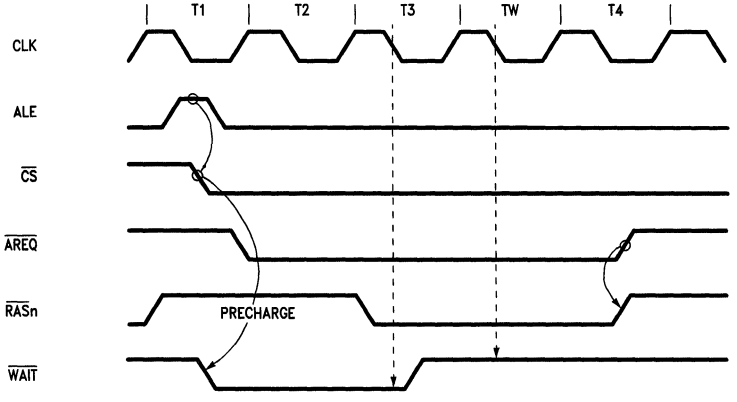
TL/F/8588–80

FIGURE 17d. Mode 1 Delayed Access with WAIT 0T (WAIT is Sampled at the End of the "T2" Clock State)



TL/F/8588–81

FIGURE 18a. Mode 0 Non-Delayed Access with WAIT 0T (WAIT is Sampled at the "T3" Falling Clock Edge)

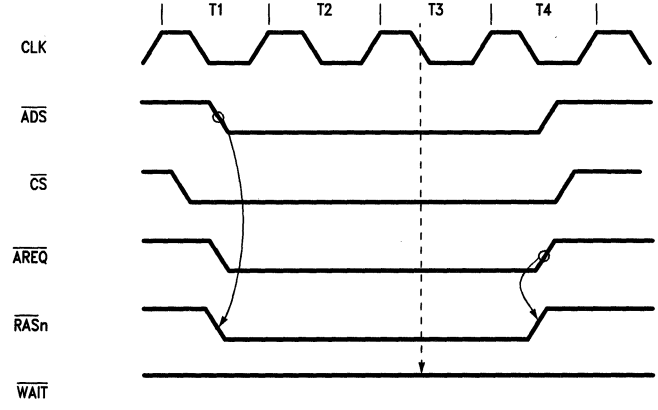


TL/F/8588–82

FIGURE 18b. Mode 0 Delayed Access with WAIT ½T (WAIT is Sampled at the "T3" Falling Clock Edge)

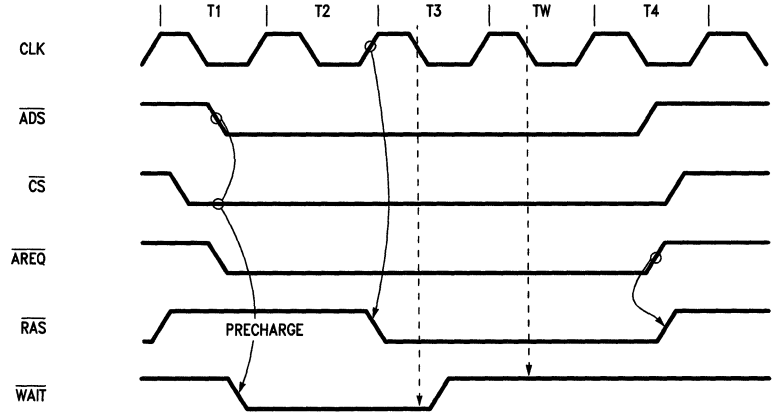
0T during non-delayed accesses and ½T during delayed accesses: WAIT will stay negated during a non-delayed access as shown in *Figures 18a* and *18c*. During an access that is delayed, WAIT will assert at the start of the access (CS and ALE or ADS) and negate on the negative level of CLK after the positive edge of CLK that asserted RAS for that access as shown in *Figures 18b* and *18d*.

5.0 Port A Wait State Support (Continued)



TL/F/8588–83

FIGURE 18c. Mode 1 Non-Delayed Access with $\overline{WAIT}$ 0T ($\overline{WAIT}$ is Sampled at the "T3" Falling Clock Edge)

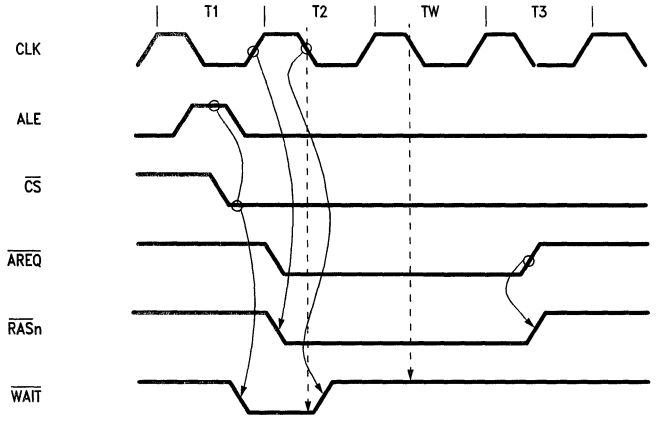


TL/F/8588–84

FIGURE 18d. Mode 1 Delayed Access with $\overline{WAIT}$ ½T ($\overline{WAIT}$ is Sampled at the "T3" Falling Clock Edge)

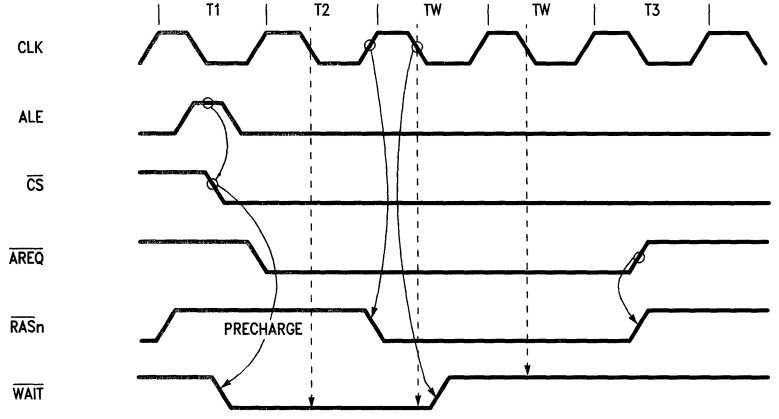
5.0 Port A Wait State Support (Continued)

½T during non-delayed and delayed accesses: if mode 0 is used, $\overline{WAIT}$ will assert when ALE is asserted and $\overline{CS}$ is asserted. $\overline{WAIT}$ will then negate on the negative level of CLK after the positive edge of CLK that asserts $\overline{RAS}$ for the access as shown in *Figure 19a*. In Mode 1, $\overline{WAIT}$ will assert from $\overline{CS}$ asserted and $\overline{ADS}$ asserted. $\overline{WAIT}$ will then negate on the negative level of CLK after $\overline{RAS}$ has been asserted for the access as shown in *Figure 19c*. During delayed accesses in both modes, $\overline{WAIT}$ will assert at the start of the access and negate on the negative level of CLK after the positive edge of CLK that started $\overline{RAS}$ for that access as shown in *Figures 19b* and *19d*.



TL/F/8588–85

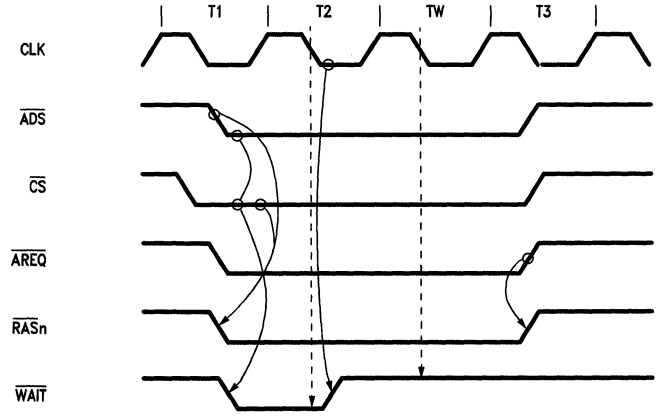
FIGURE 19a. Mode 0 Non-Delayed Access with $\overline{WAIT}$ ½T ($\overline{WAIT}$ is Sampled at the "T2" Falling Clock Edge)



TL/F/8588–86

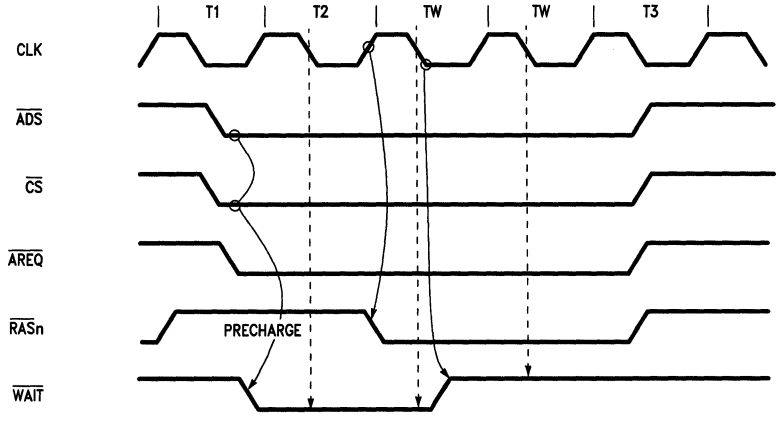
FIGURE 19b. Mode 0 Delayed Access with $\overline{WAIT}$ ½T ($\overline{WAIT}$ is Sampled at the "T2" Falling Clock Edge)

5.0 Port A Wait State Support (Continued)



TL/F/8588–87

FIGURE 19c. MODE 1 Non-Delayed Access with $\overline{WAIT}$ ½T ($\overline{WAIT}$ is Sampled at the "T2" Falling Clock Edge)

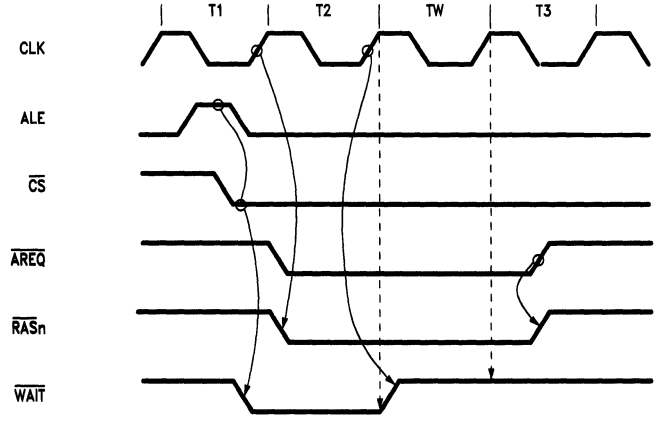


TL/F/8588–88

FIGURE 19d. Mode 1 Delayed Access with $\overline{WAIT}$ ½T ($\overline{WAIT}$ is Sampled at the "T2" Falling Clock Edge)

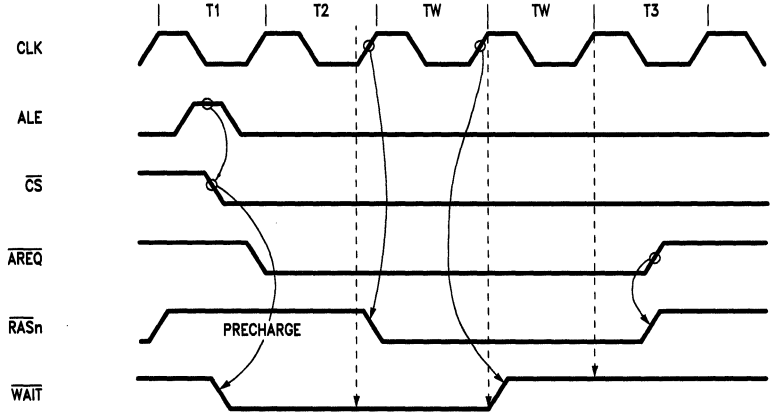
5.0 Port A Wait State Support (Continued)

1T during non-delayed and delayed accesses. In Mode 0, $\overline{WAIT}$ will assert from ALE asserted and $\overline{CS}$ asserted. $\overline{WAIT}$ will negate from the next positive edge of CLK that asserts $\overline{RAS}$ for the access as shown in *Figure 20a*. In Mode 1, $\overline{WAIT}$ will assert from $\overline{ADS}$ asserted and $\overline{CS}$ asserted. $\overline{WAIT}$ will negate from the first positive edge of CLK after $\overline{ADS}$ and $\overline{CS}$ have been asserted as shown in *Figure 20c*. During delayed accesses in both modes, $\overline{WAIT}$ will assert at the beginning of the access and will negate on the first positive edge of CLK after the positive edge of CLK that starts $\overline{RAS}$ for the access as shown in *Figures 20b* and *20d*.



TL/F/8588–89

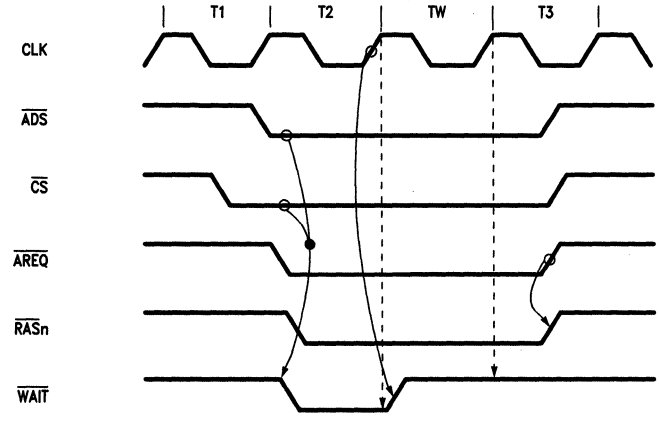
FIGURE 20a. Mode 0 Non-Delayed Access with $\overline{WAIT}$ 1T ($\overline{WAIT}$ Is Sampled at the End of the "T2" Clock State)



TL/F/8588–90

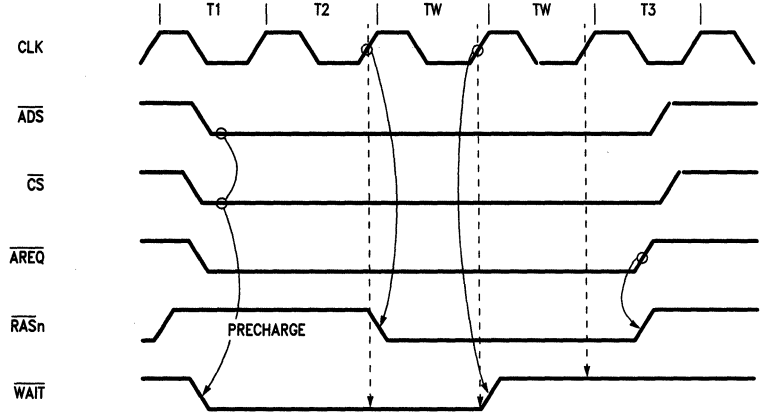
FIGURE 20b. Mode 0 Delayed Access with $\overline{WAIT}$ 1T ($\overline{WAIT}$ Is Sampled at the End of the "T2" Clock State)

5.0 Port A Wait State Support (Continued)



TL/F/8588–91

FIGURE 20c. Mode 1 Non-Delayed Access with WAIT 1T (WAIT is Sampled at the End of the "T2" Clock State)



TL/F/8588–92

FIGURE 20d. Mode 1 Delayed Access with WAIT 1T (WAIT is Sampled at the End of the "T2" Clock State)

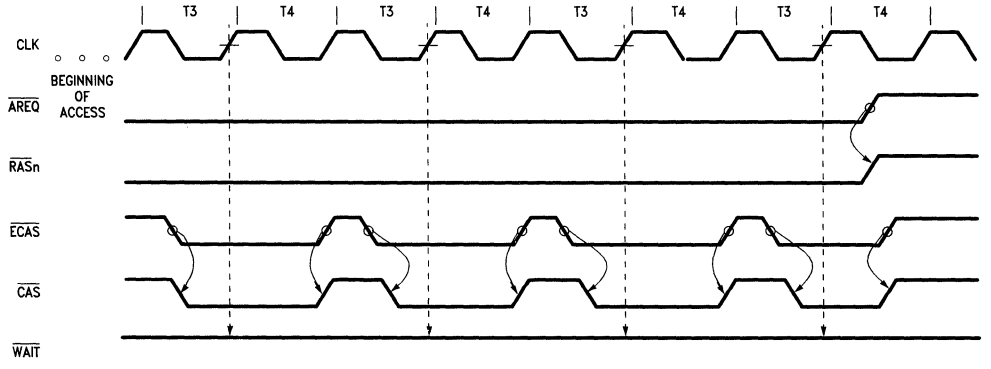
When ending WAIT from a negative level of CLK; if RAS is asserted while CLK is high then WAIT will negate from the negative edge of CLK; if RAS is asserted while CLK is low then WAIT will negate from RAS asserting. When ending WAIT from a positive edge of CLK in Mode 0, the user can think of the positive edge of CLK that starts RAS as 0T and the next positive edge of CLK as 1T. When ending WAIT from a positive edge of CLK in Mode 1, the positive edge of CLK that ADS is setup to can be thought of as 1T in a non-delayed access. In a delayed access, the positive edge of CLK that starts RAS can be thought of as 0T and the next positive edge as 1T.

5.1.2 Wait during Page Burst Accesses

WAIT can be programmed to function differently during page/burst types of accesses. During a page/burst access, the ECAS inputs will be asserted then negated while AREQ is asserted. Through address bits R4 and R5, WAIT can be programmed to assert and negate during this type of access. The user is given four programming options described below.

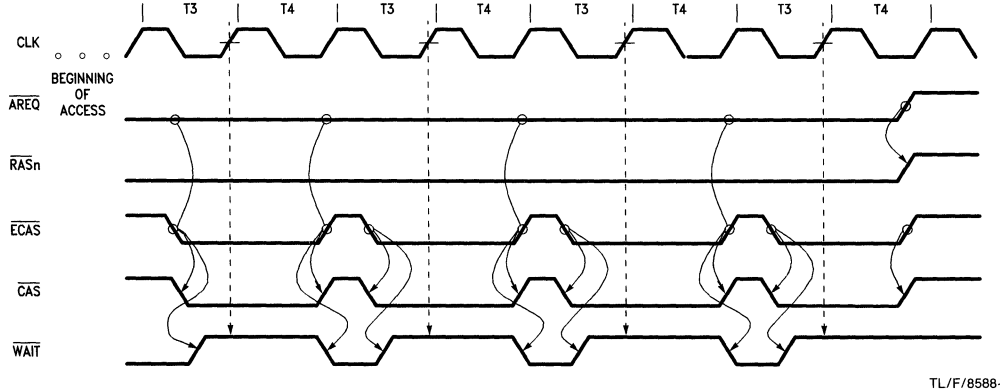
No Wait States: In this case, WAIT will remain negated even if the ECAS inputs are toggled as shown in *Figure 21*.

5.0 Port A Wait State Support (Continued)



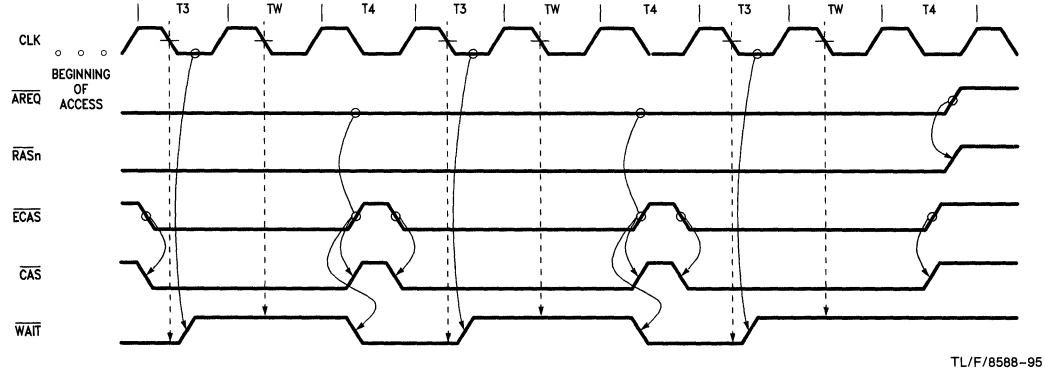
TL/F/8588–93

FIGURE 21. No Wait States during Burst (WAIT is Sampled at the End of the "T3" Clock State)



TL/F/8588–94

FIGURE 22. 0T during Burst (WAIT is Sampled at the End of the "T3" Clock State)



TL/F/8588–95

FIGURE 23. ½T during Burst Access (WAIT is Sampled at the "T3" Falling Clock Edge)

0T: WAIT will be asserted when the ECAS inputs are negated with AREQ remaining asserted. When a single or group of ECAS inputs are asserted, WAIT will be negated as shown in *Figure 22*.

½T: WAIT will be asserted when the ECAS inputs are negated with AREQ remaining asserted. When a single or group of ECAS inputs are asserted again, WAIT will be negated from the first negative level of CLK after a single ECAS or group of ECASs are asserted as shown in *Figure 23*.

5.0 Port A Wait State Support (Continued)

1T: $\overline{WAIT}$ will be asserted when the $\overline{ECAS}$ inputs are negated with $\overline{AREQ}$ remaining asserted. When a single or group of $\overline{ECAS}$ inputs are asserted again, $\overline{WAIT}$ will be negated from the first positive edge of CLK after a single $\overline{ECAS}$ or group of $\overline{ECAS}$s are asserted as shown in *Figure 24*.

When ending $\overline{WAIT}$ from a negative level of CLK; if the $\overline{ECAS}$s are asserted while CLK is high then $\overline{WAIT}$ will negate from the negative edge of CLK, if the $\overline{ECAS}$s are asserted while CLK is low then $\overline{WAIT}$ will negate from the $\overline{ECAS}$s asserting. When ending $\overline{WAIT}$ from a positive edge of CLK, the positive edge of CLK that $\overline{ECAS}$ is setup to can be thought of as 1T.

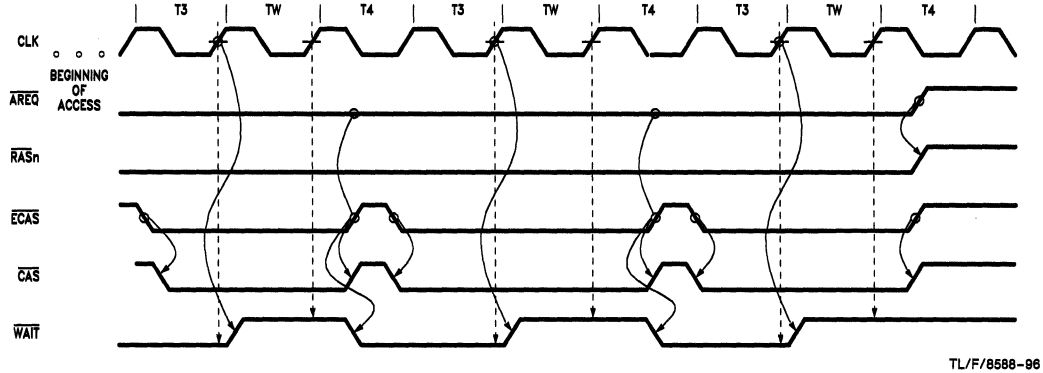
5.2 $\overline{DTACK}$ TYPE OUTPUT

With the R7 address bit asserted during programming, the user selects the $\overline{DTACK}$ type output. As long as $\overline{DTACK}$ is sampled negated by the CPU, wait states are inserted into the current access cycle as shown in *Figure 25*. Once $\overline{DTACK}$ is sampled asserted, the access cycle is completed by the CPU. $\overline{DTACK}$, which is normally negated, is programmed to assert a number of positive edges and/or negative levels from the event that starts $\overline{RAS}$ for the access. $\overline{DTACK}$ can also be programmed to function during page/burst mode accesses. Once $\overline{DTACK}$ is asserted and the $\overline{ECAS}$ inputs are negated with $\overline{AREQ}$ asserted, $\overline{DTACK}$ can be programmed to negate and assert from the $\overline{ECAS}$ inputs toggling to perform a page/burst mode operation. Once $\overline{AREQ}$ is negated, ending the access, $\overline{DTACK}$ will be negated and stays negated until the next chip selected access.

5.2.1 $\overline{DTACK}$ during Single Accesses

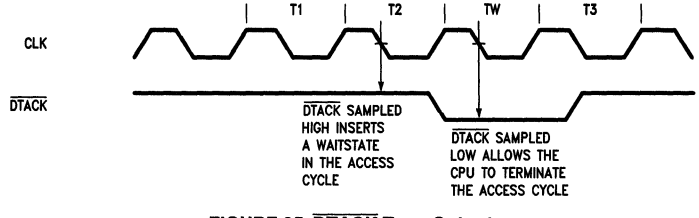
$\overline{DTACK}$ can be programmed to delay a number of positive edges and/or negative levels of CLK. These options are programmed through address bits R2 and R3 at programming time. The user is given four options described by the following.

0T during non-delayed accesses and delayed accesses: in Mode 0, $\overline{DTACK}$ will assert from the positive edge of CLK which starts $\overline{RAS}$ as shown in *Figure 26a*. In Mode 1, $\overline{DTACK}$ will assert from $\overline{ADS}$ and $\overline{CS}$ as shown in *Figure 26c*. During delayed accesses in both modes, $\overline{DTACK}$ will assert from the positive edge of CLK which starts $\overline{RAS}$ for the access as shown in *Figure 26b* and *26d*.



TL/F/8588–96

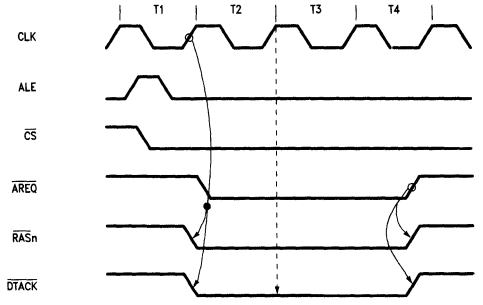
FIGURE 24. 1T during Burst Access ($\overline{WAIT}$ is Sampled at the End of the "T3" Clock State)



TL/F/8588–97

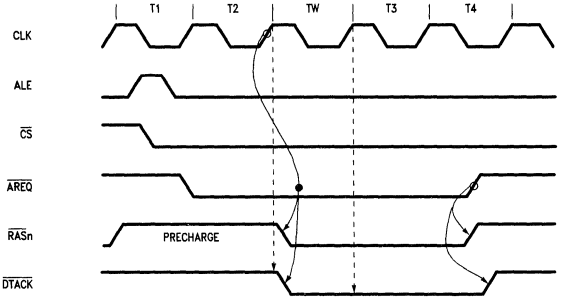
FIGURE 25. $\overline{DTACK}$ Type Output

5.0 Port A Wait State Support (Continued)



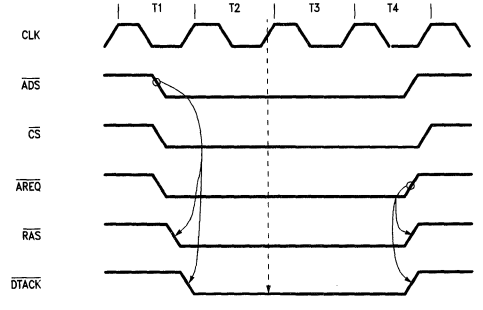
TL/F/8588-98

FIGURE 26a. Mode 0 Non-Delayed Access with DTACK 0T (DTACK is Sampled at the End of the "T2" Clock State)



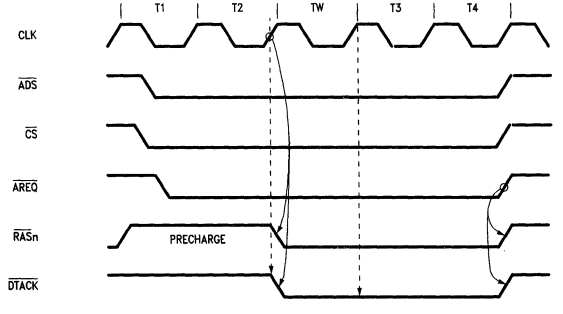
TL/F/8588-99

FIGURE 26b. Mode 0 Delayed Access with DTACK 0T (2T Clock Periods Are Programmed for RAS Precharge, DTACK is Sampled at the End of the "T2" Clock State)



TL/F/8588-A0

FIGURE 26c. Mode 1 Non-Delayed Access with DTACK 0T (DTACK is Sampled at the End of the "T2" Clock State)

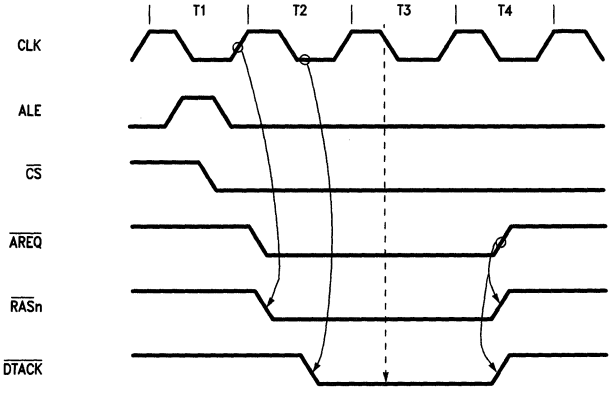


TL/F/8588-A1

FIGURE 26d. Mode 1 Delayed Access with DTACK 0T (DTACK is Sampled at the End of the "T2" Clock State)

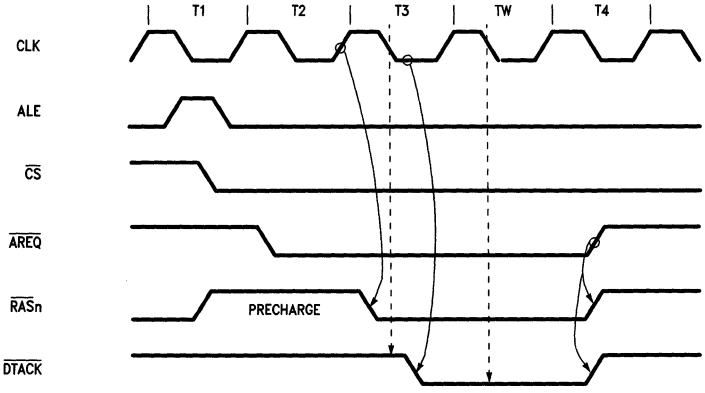
5.0 Port A Wait State Support (Continued)

½T during non-delayed and delayed accesses: In Mode 0, $\overline{DTACK}$ will assert on the negative level of CLK after the positive edge of CLK which starts $\overline{RAS}$ as shown in *Figure 27a*. In Mode 1, $\overline{DTACK}$ will assert from the negative level of CLK after $\overline{ADS}$ has been asserted given that $\overline{RAS}$ is asserted as shown in *Figures 27c* and *27d*. During delayed accesses in both modes, $\overline{DTACK}$ will assert from the negative level of CLK after the positive edge of CLK which starts $\overline{RAS}$ for the access as shown in *Figures 27b* and *27e*.



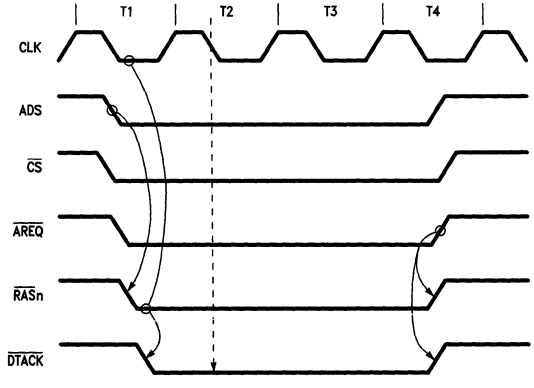
TL/F/8588–A2

FIGURE 27a. Mode 0 Non-Delayed Access with $\overline{DTACK}$ of ½T ($\overline{DTACK}$ is Sampled at the "T3" Falling Clock Edge)



TL/F/8588–A3

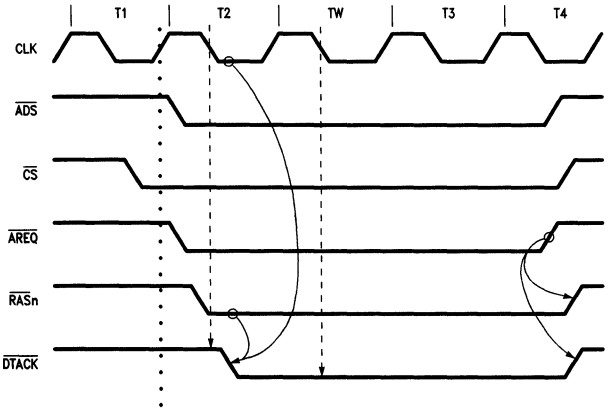
FIGURE 27b. Mode 0 Delayed Access with $\overline{DTACK}$ of ½T ($\overline{DTACK}$ is Sampled at the "T3" Falling Clock Edge)



TL/F/8588–A4

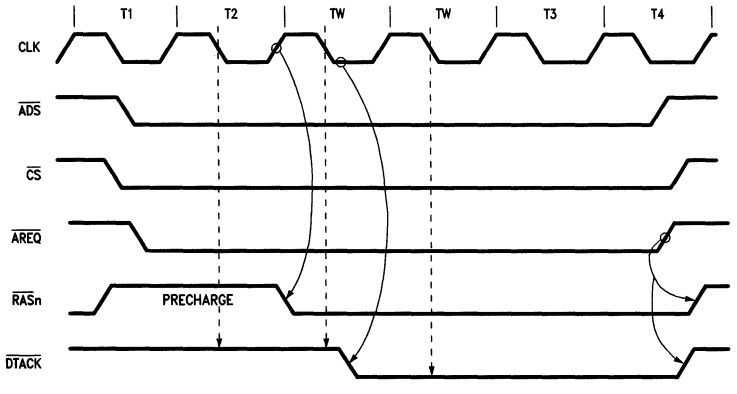
FIGURE 27c. Mode 1 Non-Delayed Access with $\overline{DTACK}$ of ½T ($\overline{DTACK}$ is Sampled at the "T2" Falling Clock Edge)

5.0 Port A Wait State Support (Continued)



TL/F/8588–A5

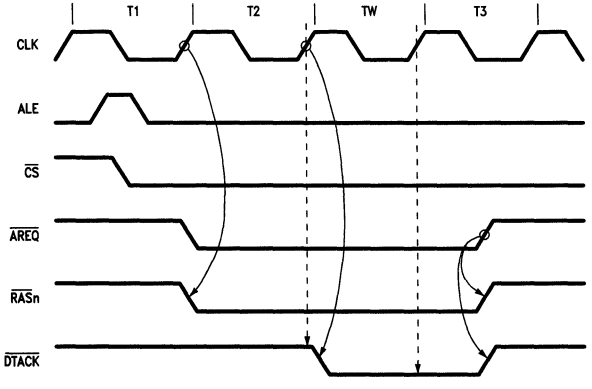
FIGURE 27d. Mode 1 Non-Delayed Access with DTACK of ½T (DTACK is Sampled at the "T2" Falling Clock Edge)



TL/F/8588–A6

FIGURE 27e. Mode 1 Delayed Access with DTACK of ½T (DTACK is Sampled at the "T2" Falling Clock Edge)

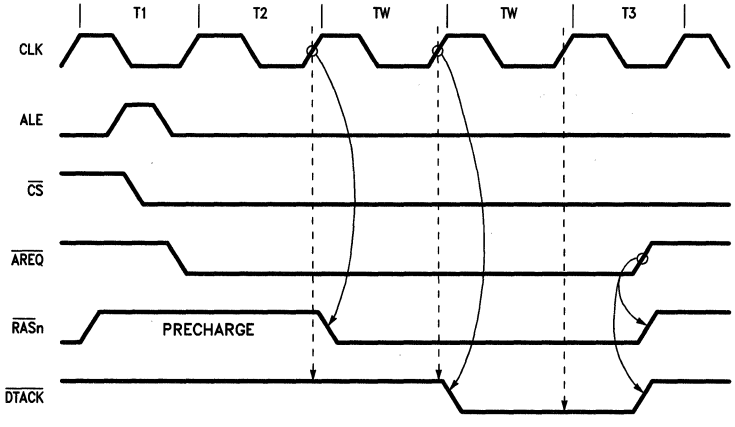
1T during delayed and non-delayed accesses: In Mode 0, DTACK will assert from the first positive edge of CLK after the positive edge of CLK which starts RAS for the access as shown in *Figure 28a*. In Mode 1, DTACK will assert from the positive edge CLK after ADS and CS are asserted as shown in *Figures 28c* and *28d*. During delayed accesses in both modes, DTACK will assert from the first positive edge of CLK after the positive edge of CLK which starts RAS for the access as shown in *Figures 28b* and *28e*.



TL/F/8588–A7

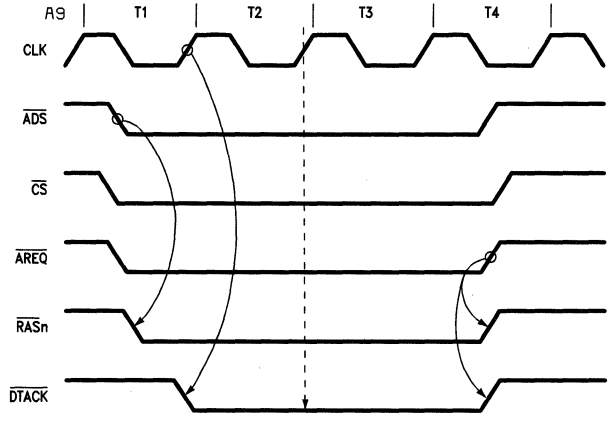
FIGURE 28a. Mode 0 Non-Delayed Access with DTACK of 1T (DTACK is Sampled at the End of the "T2" Clock State)

5.0 Port A Wait State Support (Continued)



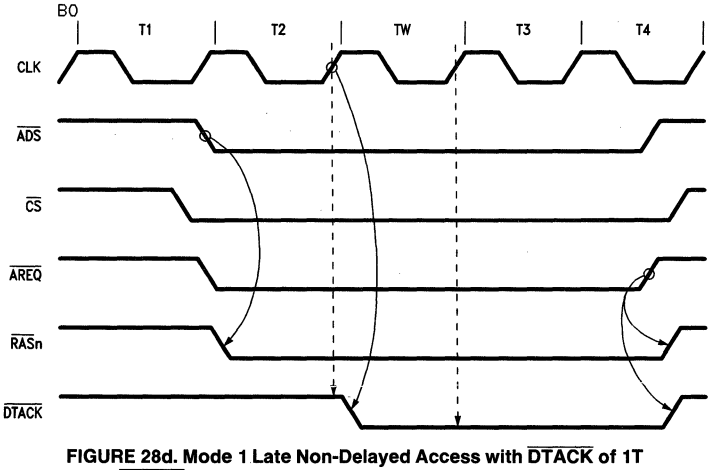
TL/F/8588–A8

FIGURE 28b. Mode 0 Delayed Access with DTACK of 1T (DTACK is Sampled at the End of the "T2" Clock State)



TL/F/8588–A9

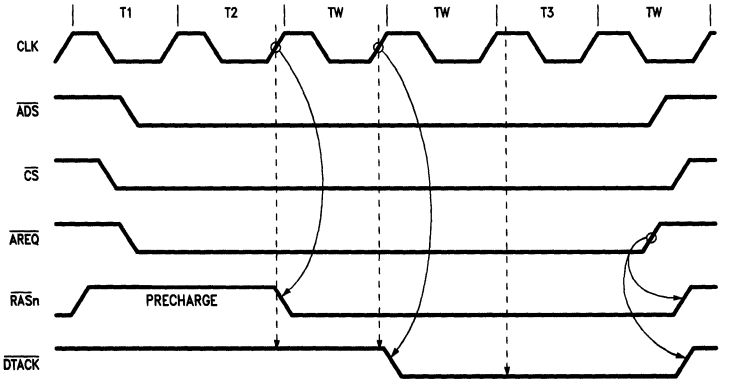
FIGURE 28c. Mode 1 Non-Delayed Access with DTACK of 1T (DTACK is Sampled at the End of the "T2" Clock State)



TL/F/8588–B0

FIGURE 28d. Mode 1 Late Non-Delayed Access with DTACK of 1T (DTACK is Sampled at the End of the "T2" Clock State)

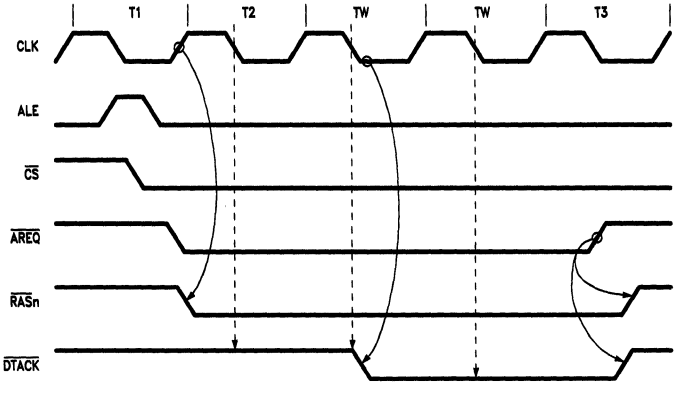
5.0 Port A Wait State Support (Continued)



TL/F/8588–B1

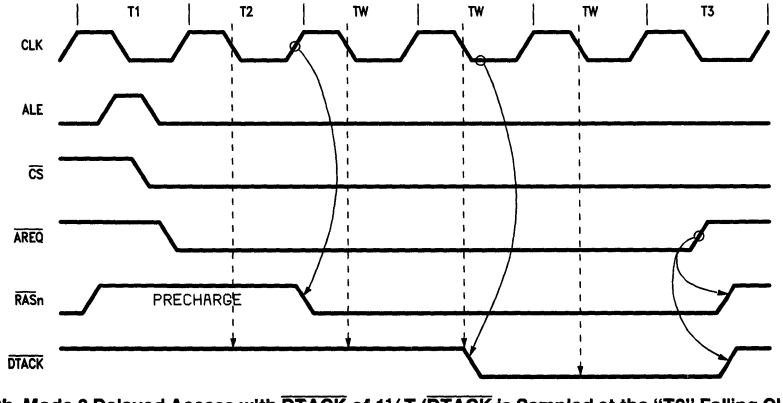
FIGURE 28e. Mode 1 Delayed Access with DTACK of 1T (DTACK is Sampled at the End of the "T2" Clock State)

1½T during delayed and non-delayed accesses: In Mode 0, DTACK will assert from the negative level after the first positive edge of CLK after the positive edge of CLK which starts RAS for the access as shown in *Figure 29a*. In Mode 1, DTACK will assert from the negative level after the first positive edge of CLK after ADS and CS are asserted as shown in *Figures 29c* and *29d*. During delayed accesses in both modes, DTACK will assert from the negative level after the first positive edge of CLK after the positive edge of CLK which starts RAS for the access as shown in *Figures 29b* and *29e*.



TL/F/8588–B2

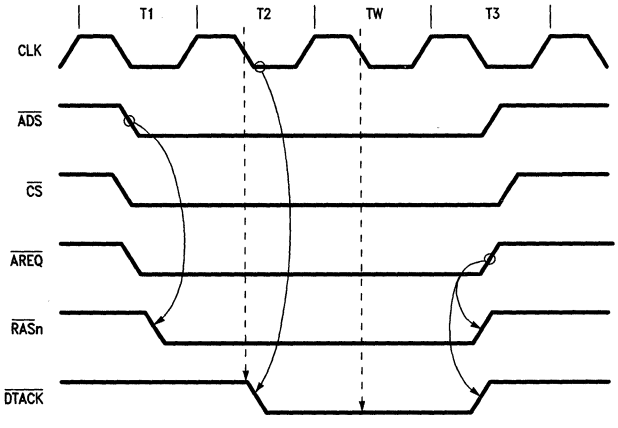
FIGURE 29a. Mode 0 Non-Delayed Access with DTACK of 1½T (DTACK is Sampled at the "T2" Falling Clock Edge)



TL/F/8588–B3

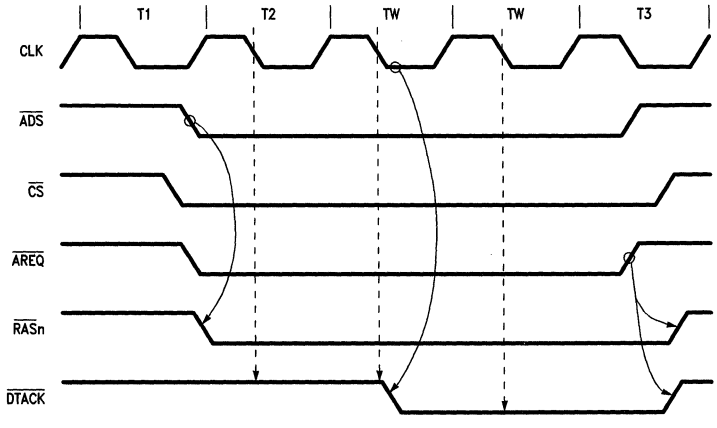
FIGURE 29b. Mode 0 Delayed Access with DTACK of 1½T (DTACK is Sampled at the "T2" Falling Clock Edge)

5.0 Port A Wait State Support (Continued)



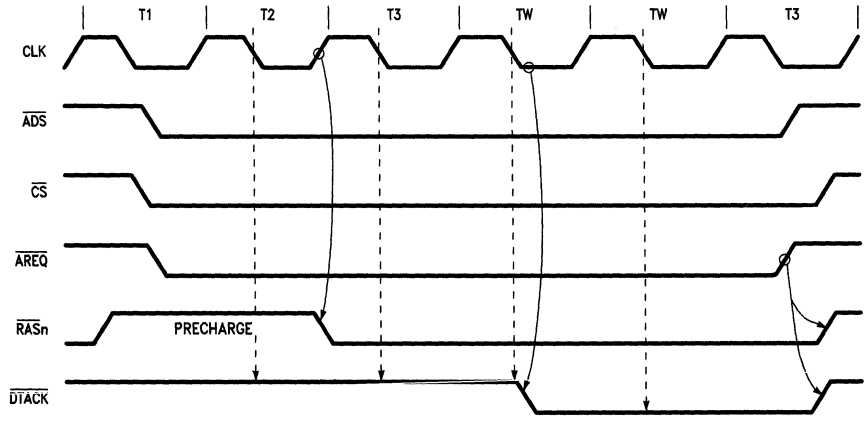
TL/F/8588–B4

FIGURE 29c. Mode 1 Non-Delayed Access with $\overline{\text{DTACK}}$ of 1½T ($\overline{\text{DTACK}}$ is Sampled at the "T2" Falling Clock Edge)



TL/F/8588–B5

FIGURE 29d. Mode 1 Non-Delayed Access with $\overline{\text{DTACK}}$ of 1½T ($\overline{\text{DTACK}}$ is Sampled at the "T2" Falling Clock Edge)



TL/F/8588–B6

FIGURE 29e. Mode 1 Delayed Access with $\overline{\text{DTACK}}$ of 1½T ($\overline{\text{DTACK}}$ is Sampled at the "T2" Falling Clock Edge)

5.0 Port A Wait State Support (Continued)

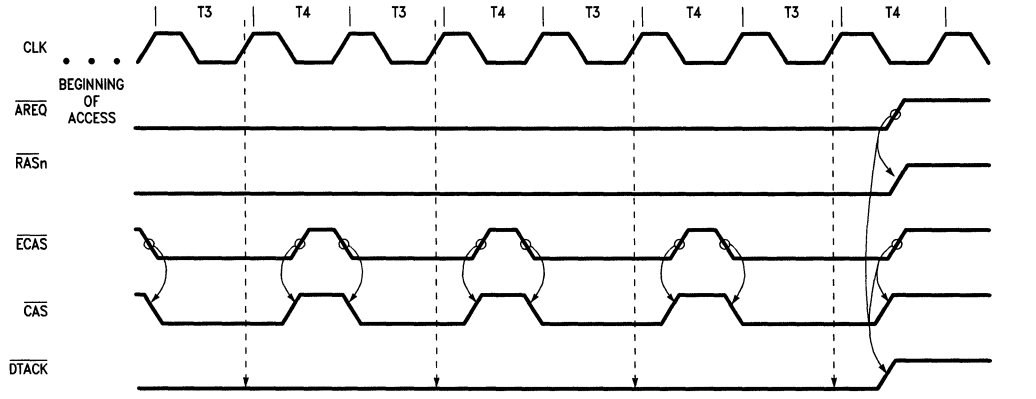
When starting $\overline{DTACK}$ from a negative level of CLK; if $\overline{RAS}$ is asserted while CLK is high then $\overline{DTACK}$ will assert from the negative edge of CLK, if $\overline{RAS}$ is asserted while CLK is low, then $\overline{DTACK}$ will assert from $\overline{RAS}$ asserting. When starting $\overline{DTACK}$ from a positive edge of CLK in Mode 0, the positive edge of CLK that starts $\overline{RAS}$ can be thought of as 0T. In Mode 1 during non-delayed accesses, the positive edge of CLK that $\overline{ADS}$ is setup to can be thought of as 1T. During delayed accesses, the positive edge of CLK that starts $\overline{RAS}$ can be thought of as 0T and the next positive edge of CLK as 1T.

5.2.2 $\overline{DTACK}$ during Page/Burst Accesses

$\overline{DTACK}$ can be programmed to function differently during page/burst types of accesses. During a page/burst access, the $\overline{ECAS}$ inputs will be asserted then negated while $\overline{AREQ}$ remains asserted. Through address bits R4 and R5, $\overline{DTACK}$ can be programmed to negate and assert during this type of access. The user is given four programming options described below.

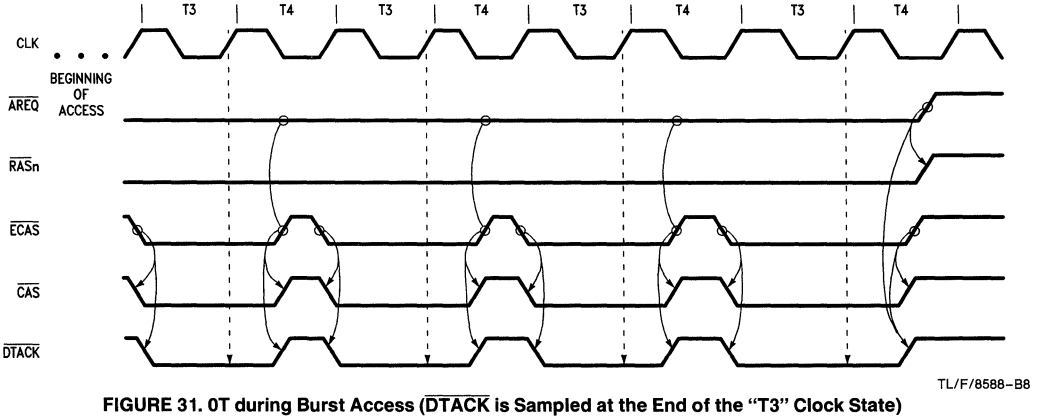
No Wait States: In this case, $\overline{DTACK}$ wll remain asserted even if the $\overline{ECAS}$ inputs are negated with $\overline{AREQ}$ asserted as shown in *Figure 30*.

0T: $\overline{DTACK}$ will be negated when the $\overline{ECAS}$ inputs are negated with $\overline{AREQ}$ asserted. When a single or group of $\overline{ECAS}$ inputs are asserted again, $\overline{DTACK}$ will be asserted as shown in *Figure 31*.



TL/F/8588–B7

FIGURE 30. No Wait States during Burst Access ($\overline{DTACK}$ is Sampled at the End of the "T3" Clock State)



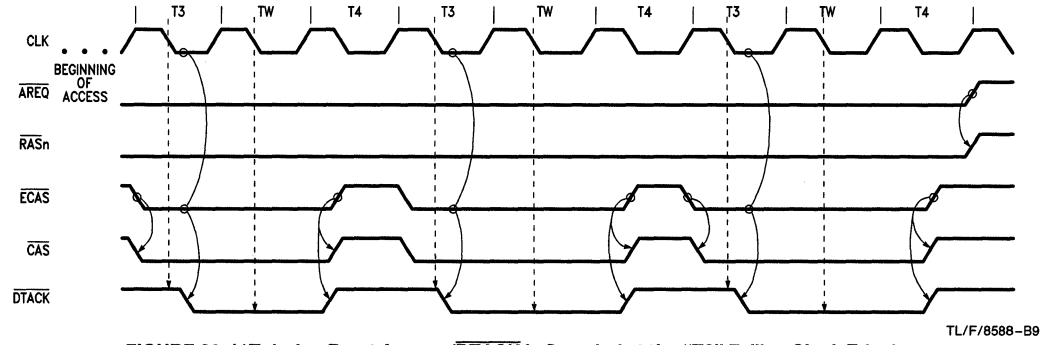
TL/F/8588–B8

FIGURE 31. 0T during Burst Access ($\overline{DTACK}$ is Sampled at the End of the "T3" Clock State)

5.0 Port A Wait State Support (Continued)

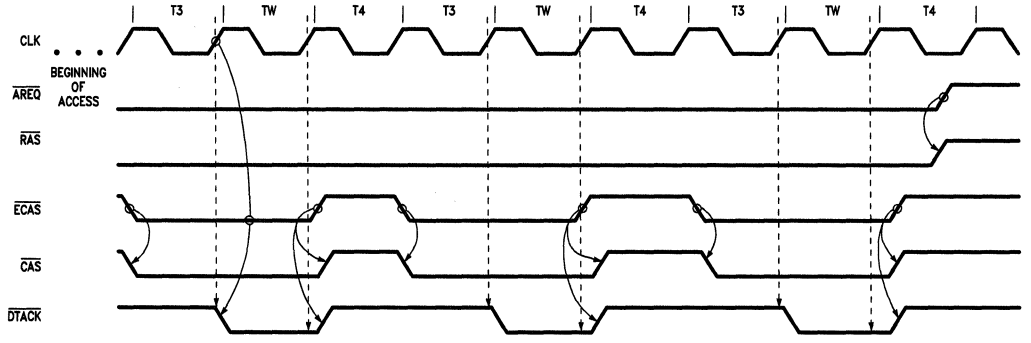
½T: $\overline{DTACK}$ will be negated when the $\overline{ECAS}$ inputs are negated with $\overline{AREQ}$ asserted. When a single or group of $\overline{ECAS}$ inputs are asserted again, $\overline{DTACK}$ will be asserted from the first negative level of CLK after the single or group of $\overline{ECAS}$s are asserted as shown in *Figure 32*.

1T: $\overline{DTACK}$ will be negated when the $\overline{ECAS}$ inputs are negated with $\overline{AREQ}$ asserted. When a single or group of $\overline{ECAS}$ inputs are asserted again, $\overline{DTACK}$ will be asserted from the first positive edge of CLK after the single or group of $\overline{ECAS}$s are asserted as shown in *Figure 33*.



TL/F/8588–B9

FIGURE 32. ½T during Burst Access ($\overline{DTACK}$ is Sampled at the "T3" Falling Clock Edge)



TL/F/8588–C0

FIGURE 33. 1T during Burst Access ($\overline{DTACK}$ is Sampled at the "T3" Falling Clock Edge)

5.0 Port A Wait State Support (Continued)

When starting $\overline{DTACK}$ from a negative level of CLK; if the $\overline{ECAS}$s are asserted while CLK is high then $\overline{DTACK}$ will assert from the negative edge of CLK, if the $\overline{ECAS}$s are asserted while CLK is low then $\overline{DTACK}$ will assert from the $\overline{ECAS}$s asserting. When starting $\overline{DTACK}$ from a positive edge of CLK, the positive edge of CLK that $\overline{ECAS}$ is setup to can be thought of as 1T.

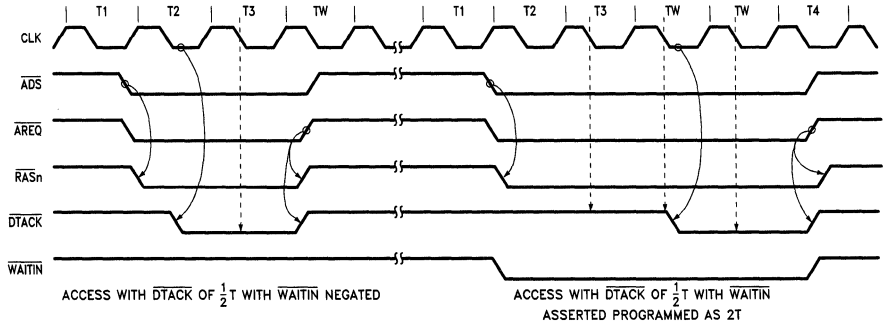
5.3 DYNAMICALLY INCREASING THE NUMBER OF WAIT STATES

The user can increase the number of positive edges of CLK before $\overline{DTACK}$ is asserted or $\overline{WAIT}$ is negated. With the input $\overline{WAITIN}$ asserted, the user can delay $\overline{DTACK}$ asserting or $\overline{WAIT}$ negating either one or two more positive edges of CLK. The number of edges is programmed through address bit R6. If the user is increasing the number of positive edges in a delay that contains a negative level, the positive edges will be met before the negative level. For example if the user programmed $\overline{DTACK}$ of ½T, asserting $\overline{WAITIN}$, programmed as 2T, would increase the number of positive edges resulting in $\overline{DTACK}$ of 2½T as shown in *Figure 34a*. Similarly, $\overline{WAITIN}$ can increase the number of positive edges in a page/burst access. $\overline{WAITIN}$ can be permanently asserted in systems requiring an increased number of wait states. $\overline{WAITIN}$ can also be asserted and negated, depending on the type of access. As an example, a user could invert the $\overline{WRITE}$ line from the CPU and connect the output to $\overline{WAITIN}$. This could be used to perform write accesses with 1 wait state and read accesses with 2 wait states as shown in *Figure 34b*.

5.4 GUARANTEEING $\overline{RAS}$ LOW TIME AND $\overline{RAS}$ PRECHARGE TIME

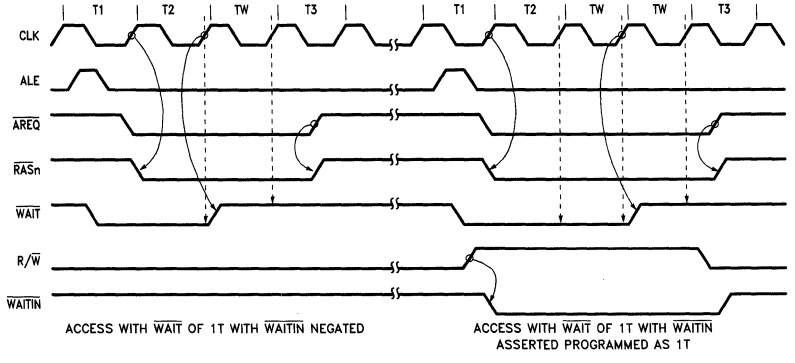
The DP8420A/21A/22A will guarantee $\overline{RAS}$ precharge time between accesses; between refreshes; and between access and refreshes. The programming bits R0 and R1 are used to program combinations of $\overline{RAS}$ precharge time and $\overline{RAS}$ low time referenced by positive edges of CLK. $\overline{RAS}$ low time is programmed for refreshes only. During an access, the system designer guarantees the time $\overline{RAS}$ is asserted through the DP8420A/21A/22A wait logic. Since inserting wait states into an access increases the length of the CPU signals which are used to create $\overline{ADS}$ or ALE and $\overline{AREQ}$, the time that $\overline{RAS}$ is asserted can be guaranteed.

Precharge time is also guaranteed by the DP8420A/21A/22A. Each $\overline{RAS}$ output has a separate positive edge of CLK counter. $\overline{AREQ}$ is negated setup to a positive edge of CLK to terminate the access. That positive edge is 1T. The next positive edge is 2T. $\overline{RAS}$ will not be asserted until the programmed number of positive edges of CLK have passed as shown in *Figures 35, 37a,* and *37b*. Once the programmed precharge time has been met, $\overline{RAS}$ will be asserted from the positive edge of CLK. However, since there is a precharge counter per $\overline{RAS}$, an access using another $\overline{RAS}$ will not be delayed. Precharge time before a refresh is always referenced from the access $\overline{RAS}$ negating before $\overline{RAS0}$ for the refresh asserting. After a refresh, precharge time is referenced from $\overline{RAS3}$ negating, for the refresh, to the access $\overline{RAS}$ asserting.



TL/F/8588–C1

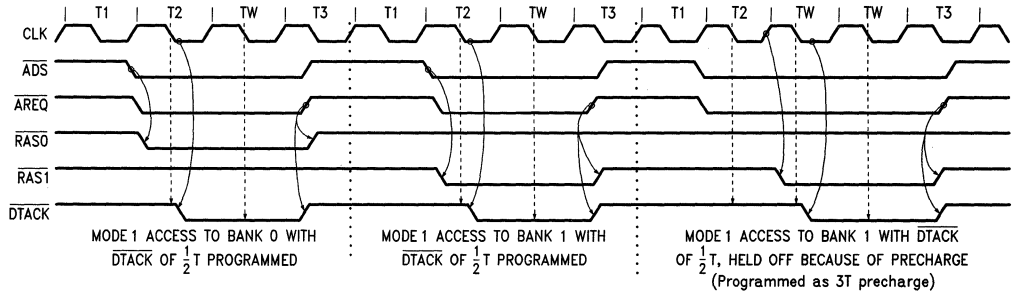
FIGURE 34a. $\overline{WAITIN}$ Example ($\overline{DTACK}$ is Sampled at the "T3" Falling Clock Edge)



TL/F/8588–C2

FIGURE 34b. $\overline{WAITIN}$ Example ($\overline{WAIT}$ is Sampled at the End of "T2")

5.0 Port A Wait State Support (Continued)



TL/F/8588–C3

FIGURE 35. Guaranteeing RAS Precharge (DTACK is Sampled at the "T2" Falling Clock Edge)

6.0 Additional Access Support Features

To support the different modes of accessing, the DP8420A/21A/22A have multiple access features. These features allow the user to take advantage of CPU or DRAM functions. These additional features include: address latches and column increment for page/burst mode support; address pipelining to allow a new access to start to a different bank of DRAM after CAS has been asserted and the column address hold time has been met; and delay CAS, to allow the user with a multiplexed bus to ensure valid data is present before CAS is asserted.

6.1 ADDRESS LATCHES AND COLUMN INCREMENT

The address latches can be programmed, through programming bit B0, to either latch the address or remain permanently in fall-through mode. If the address latches are used to latch the address, the rising edge of ALE in Mode 0 places the latches in fall-through. Once ALE is negated, the address present on the row, column and bank inputs is latched. In Mode 1, the address latches are in fall-through mode until ADS is asserted. ADS asserted latches the address.

Once the address is latched, the column address can be incremented with the input COLINC. With COLINC asserted, the column address is incremented. If COLINC is asserted with all of the bits of the column address asserted, the column address will return to zero. COLINC can be used for sequential accesses of static column DRAMs. COLINC can also be used with the ECAS inputs to support sequential accesses to page mode DRAMs as shown in *Figure 36*. COLINC should only be asserted when the signal RFIP is negated during an access since this input functions as extend refresh when RFIP is asserted. COLINC must be low (negated) when the address is being latched (ADS falling edge in Mode 1).

The address latches function differently with the DP8422A. The DP8422A will latch the address of the currently granted port. If Port A is currently granted, the address will be latched as described in Section 6.1. If Port A is not granted, and requests an access, the address will be latched on the first or second positive edge of CLK after GRANTB has been negated depending on the programming bits R0, R1. For Port B, if GRANTB is asserted, the address will be latched with AREQB asserted. If GRANTB is negated, the address will latch on the first or second positive edge of CLK after GRANTB is asserted depending on the programming bits R0, R1.

6.0 Additional Access Support Features (Continued)

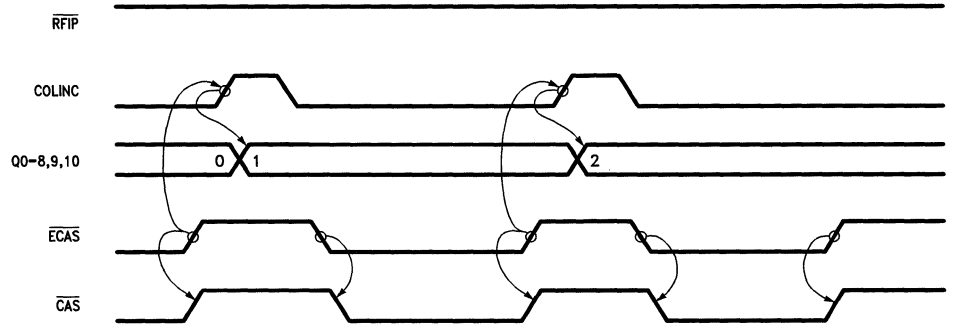
6.2 ADDRESS PIPELINING

Address pipelining is the overlapping of accesses to different banks of DRAM. If the majority of successive accesses are to a different bank, the accesses can be overlapped. Because of this overlapping, the cycle time of the DRAM accesses are greatly reduced. The DP8420A/21A/22A can be programmed to allow a new row address to be placed on the DRAM address bus after the column address hold time has been met. At this time, a new access can be initiated with $\overline{ADS}$ or ALE, depending on the access mode, while $\overline{AREQ}$ is used to sustain the current access. The DP8422A supports address pipelining for Port A only. This mode can not be used with page, static column or nibble modes of operations because the DRAM column address is switched back to the row address after $\overline{CAS}$ is asserted. This mode is programmed through address bit R8 (see *Figures 37a* and *37b*). In this mode, the output $\overline{WE}$ always functions as $\overline{RFRQ}$.

During address pipelining in Mode 0, shown in *Figure 37c*, ALE cannot be pulsed high to start another access until $\overline{AREQ}$ has been asserted for the previous access for at least one period of CLK. $\overline{DTACK}$, if programmed, will be negated once $\overline{AREQ}$ is negated. $\overline{WAIT}$, if programmed to insert wait states, will be asserted once ALE and $\overline{CS}$ are asserted.

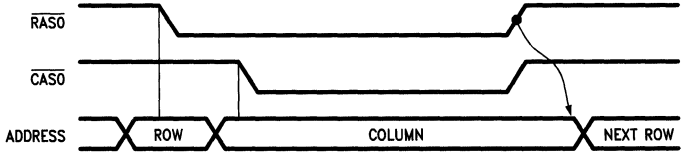
In Mode 1, shown in *Figure 37d*, $\overline{ADS}$ can be negated once $\overline{AREQ}$ is asserted. After meeting the minimum negated pulse width for $\overline{ADS}$, $\overline{ADS}$ can again be asserted to start a new access. $\overline{DTACK}$, if programmed, will be negated once $\overline{AREQ}$ is negated. $\overline{WAIT}$, if programmed, will be asserted once $\overline{ADS}$ is asserted.

In either mode with either type of wait programmed, the DP8420A/21A/22A will still delay the access for precharge if sequential accesses are to the same bank or if a refresh takes place.



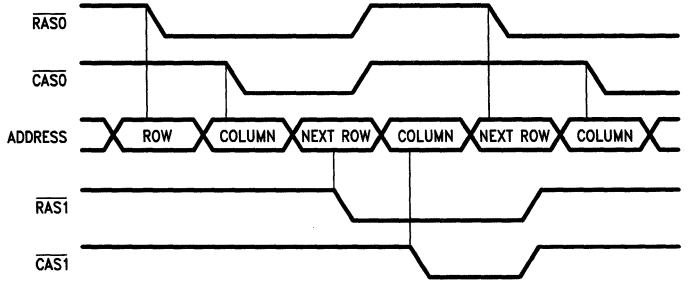
TL/F/8588–C4

FIGURE 36. Column Increment



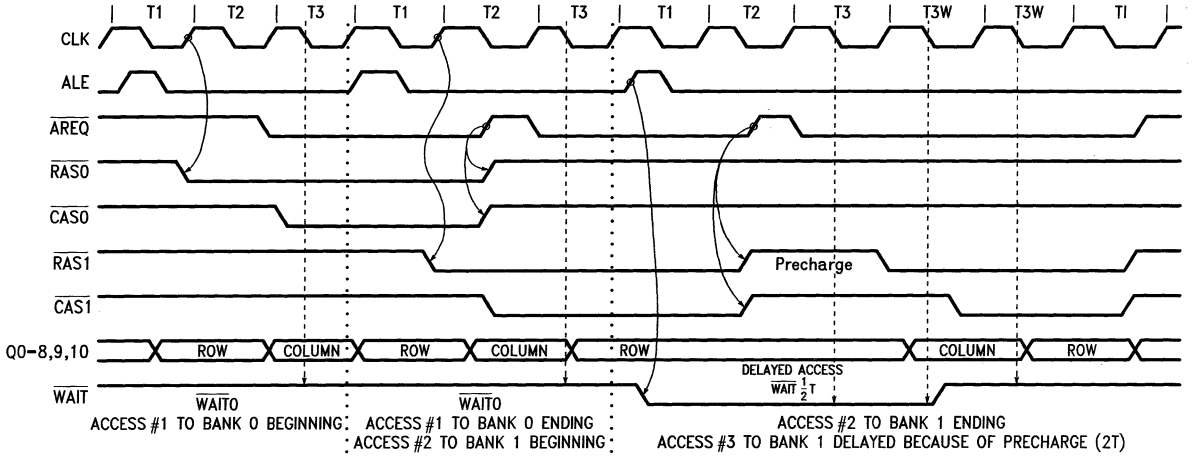
TL/F/8588–G0

FIGURE 37a. Non-Address Pipelined Mode



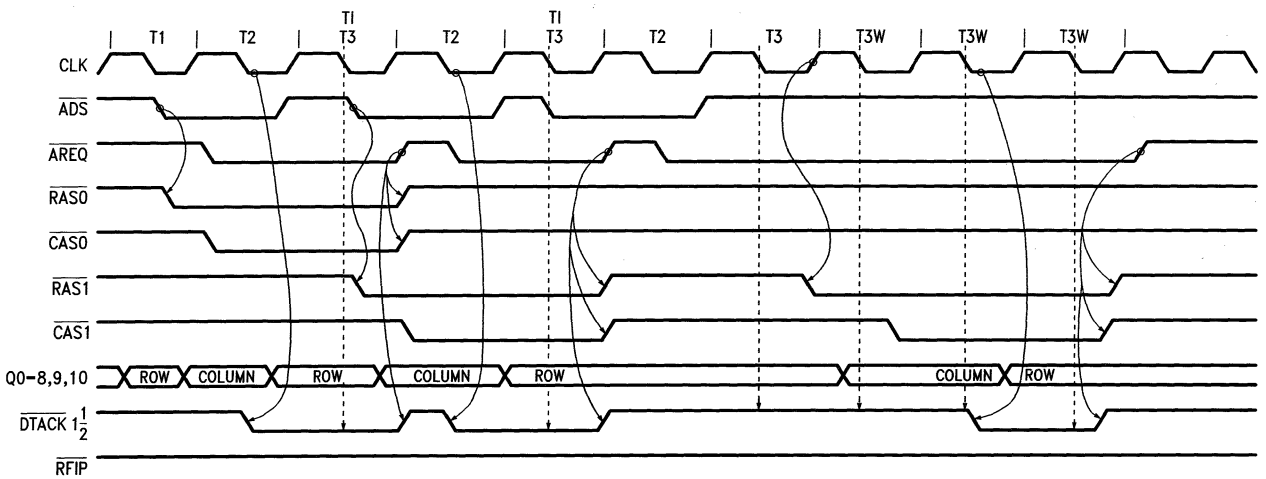
TL/F/8588–G1

FIGURE 37b. Address Pipelined Mode



TL/F/8588–C5

FIGURE 37c. Mode 0 Address Pipelining (WAIT of 0, ½T has Been Programmed. WAIT is Sampled at the "T3" Falling Clock Edge)



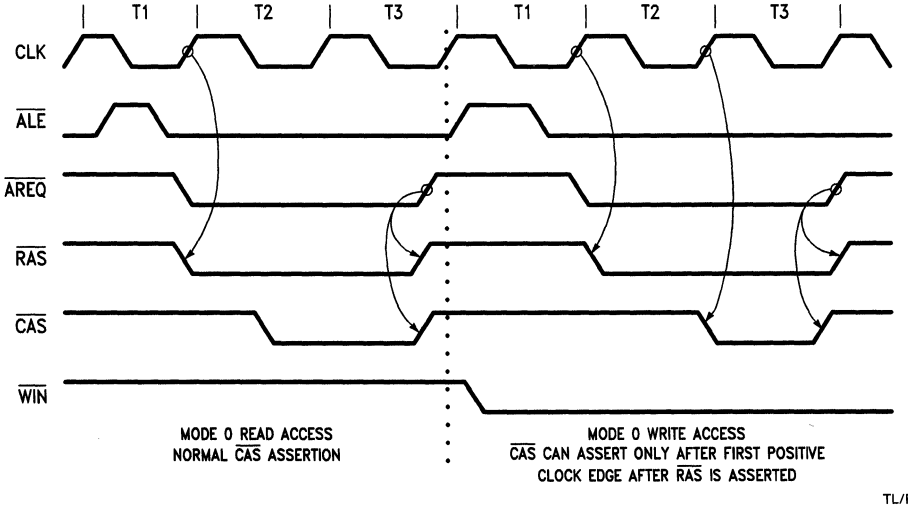
TL/F/8588–C6

FIGURE 37d. Mode 1 Address Pipelining (DTACK 1½T Programmed, DTACK is Sampled at the "T3" Falling Clock Edge)

6.0 Additional Access Support Features (Continued)

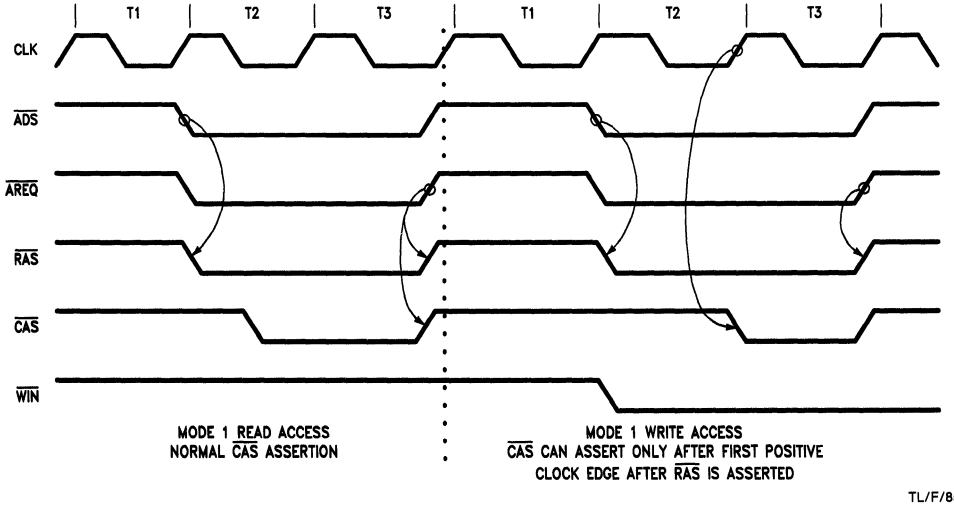
6.3 DELAY $\overline{CAS}$ DURING WRITE ACCESSES

Address bit C9 asserted during programming will cause $\overline{CAS}$ to be delayed until the first positive edge of CLK after $\overline{RAS}$ is asserted when the input $\overline{WIN}$ is asserted. Delaying $\overline{CAS}$ during write accesses ensures that the data to be written to DRAM will be setup to $\overline{CAS}$ asserting as shown in *Figures 38a* and *38b*. If the possibility exists that data still may not be present after the first positive edge of CLK, $\overline{CAS}$ can be delayed further with the $\overline{ECAS}$ inputs. If address bit C9 is negated during programming, read and write accesses will be treated the same (with regard to $\overline{CAS}$).



TL/F/8588–C7

FIGURE 38a. Mode 0 Delay $\overline{CAS}$



TL/F/8588–C8

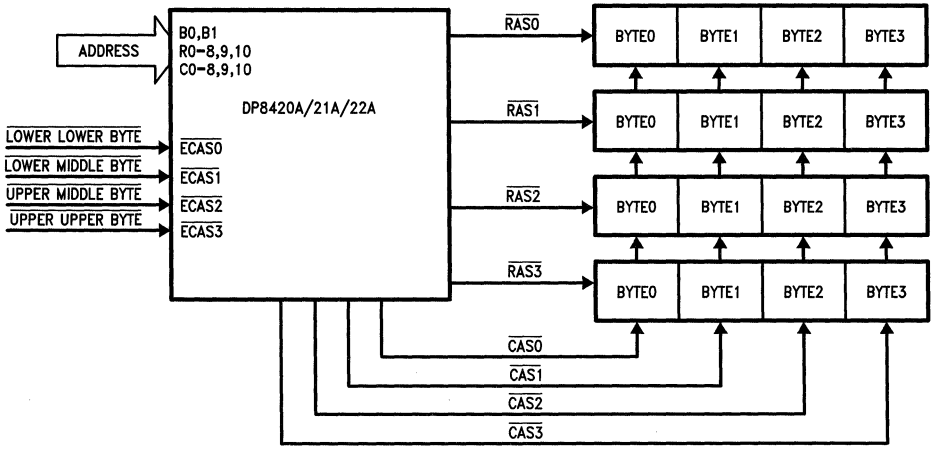
FIGURE 38b. Mode 1 Delay $\overline{CAS}$

7.0 $\overline{RAS}$ and $\overline{CAS}$ Configuration Modes

The DP8420A/21A/22A allow the user to configure the DRAM array to contain one, two or four banks of DRAM. Depending on the functions used, certain considerations must be used when determining how to set up the DRAM array. Programming address bits C4, C5 and C6 along with bank selects, B0–1, and $\overline{CAS}$ enables, $\overline{ECAS}$0–3, determine which $\overline{RAS}$ or group of $\overline{RAS}$s and which $\overline{CAS}$ or group of $\overline{CAS}$s will be asserted during an access. Different memory schemes are described. The DP8420A/21A/22A is specified driving a heavy load of 72 DRAMs, representing four banks of DRAM with 16-bit words and 2 parity bits. The DP8420A/21A/22A can drive more than 72 DRAMs, but the AC timing must be increased. Since the $\overline{RAS}$ and $\overline{CAS}$ outputs are configurable, all $\overline{RAS}$ and $\overline{CAS}$ outputs should be used for the maximum amount of drive.

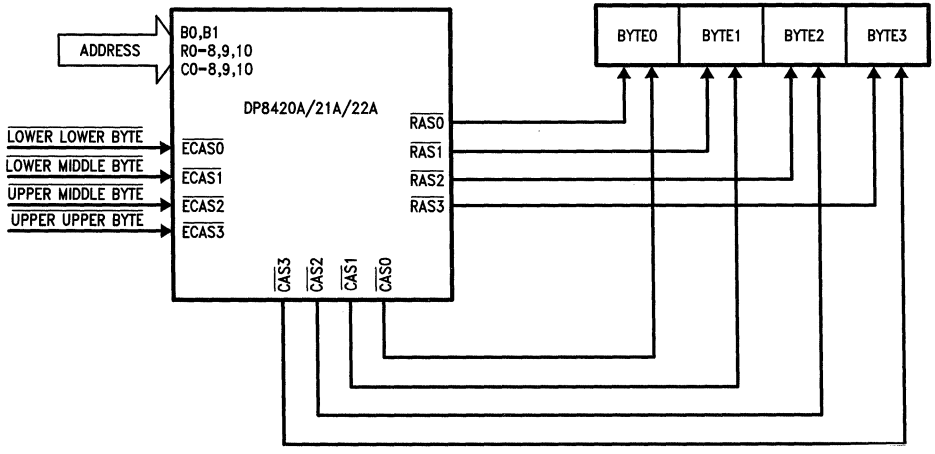
7.1 BYTE WRITING

By selecting a configuration in which all $\overline{CAS}$ outputs are selected during an access, the $\overline{ECAS}$ inputs enable a single or group of $\overline{CAS}$ outputs to select a byte (or bytes) in a word size of up to 32 bits. In this case, the $\overline{RAS}$ outputs are used to select which of up to 4 banks is to be used as shown in *Figures 39a* and *39b*. In systems with a word size of 16 bits, the byte enables can be gated with a high order address bit to produce four byte enables which gives an equivalent to 8 banks of 16-bit words as shown in *Figure 39d*. If less memory is required, each $\overline{CAS}$ should be used to drive each nibble in the 16-bit word as shown in *Figure 39c*.



TL/F/8588–C9

FIGURE 39a. DRAM Array Setup for 32-Bit System (C6, C5, C4 = 1, 1, 0 during Programming)



TL/F/8588–D0

FIGURE 39b. DRAM Array Setup for 32-Bit, 1 Bank System (C6, C5, C4 = 0, 0, 0 Allowing Error Scrubbing or C6, C5, C4 = 0, 1, 1 No Error Scrubbing during Programming)

7.0 $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ Configuration Modes (Continued)

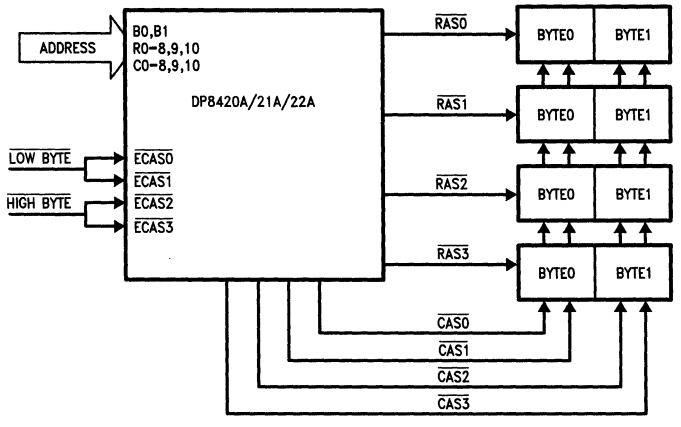


FIGURE 39c. DRAM Array Setup for 16-Bit System (C6, C5, C4 = 1, 1, 0 during Programming)

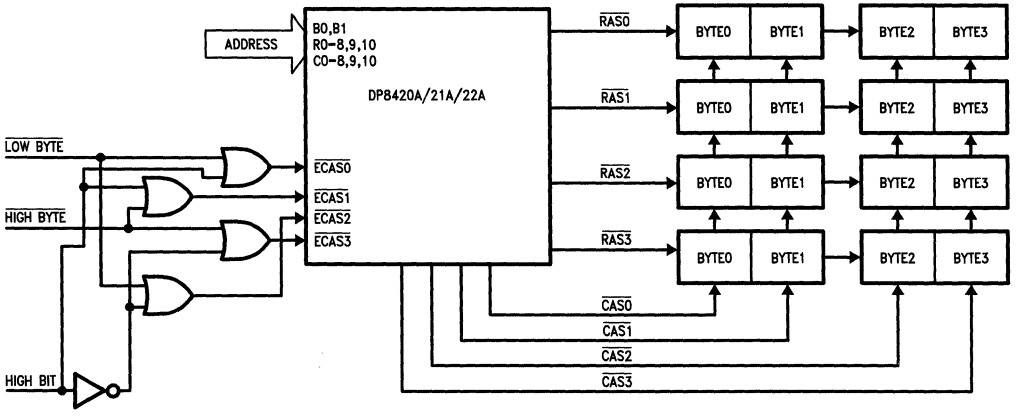


FIGURE 39d. 8 Bank DRAM Array for 16-Bit System (C6, C5, C4 = 1, 1, 0 during Programming)

7.0 $\overline{RAS}$ and $\overline{CAS}$ Configuration Modes (Continued)

7.2 MEMORY INTERLEAVING

Memory interleaving allows the cycle time of DRAMs to be reduced by having sequential accesses to different memory banks. Since the DP8420A/21A/22A have separate precharge counters per bank, sequential accesses will not be delayed if the accessed banks use different $\overline{RAS}$ outputs. To ensure different $\overline{RAS}$ outputs will be used, a mode is selected where either one or two $\overline{RAS}$ outputs will be asserted during an access. The bank select or selects, B0 and B1, are then tied to the least significant address bits, causing a different group of $\overline{RAS}$s to assert during each sequential access as shown in *Figure 40*. In this figure there should be at least one clock period of all $\overline{RAS}$'s negated between different $\overline{RAS}$'s being asserted to avoid the condition of a $\overline{CAS}$ before $\overline{RAS}$ refresh cycle.

7.3 ADDRESS PIPELINING

Address pipelining allows several access $\overline{RAS}$s to be asserted at once. Because $\overline{RAS}$s can overlap, each bank requires either a mode where one $\overline{RAS}$ and one $\overline{CAS}$ are used per bank as shown in *Figure 41a* or where two $\overline{RAS}$s and two $\overline{CAS}$s are used per bank as shown in *Figure 41b*. Byte writing can be accomplished in a 16-bit word system if two $\overline{RAS}$s and two $\overline{CAS}$s are used per bank. In other systems, $\overline{WE}$s (or external gating on the $\overline{CAS}$ outputs) must be used to perform byte writing. If $\overline{WE}$s are used separate data in and data out buffers must be used. If the array is not layed out this way, a $\overline{CAS}$ to a bank can be low before $\overline{RAS}$, which will cause a refresh of the DRAM, not an access. To take full advantage of address pipelining, memory interleaving is used. To memory interleave, the least significant address bits should be tied to the bank select inputs to ensure that all "back to back" sequential accesses are not delayed, since different memory banks are accessed.

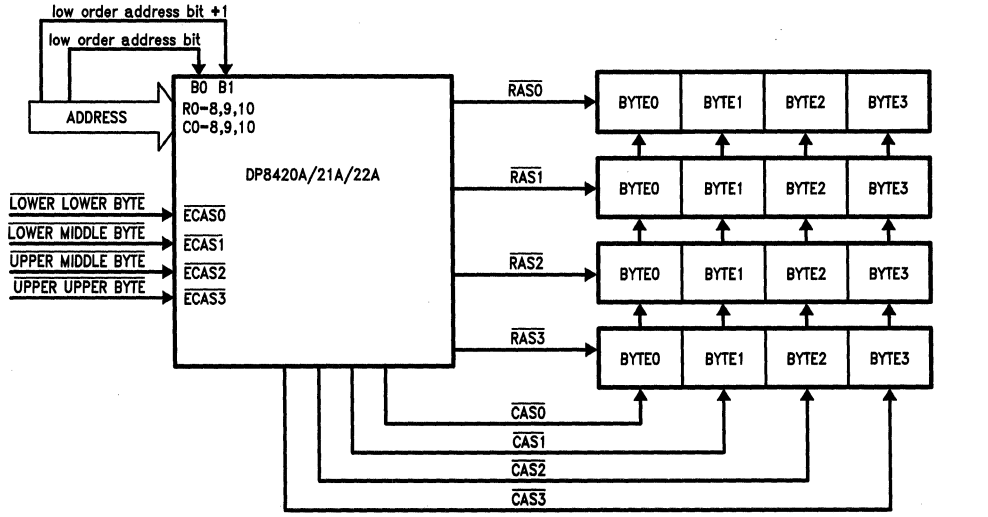
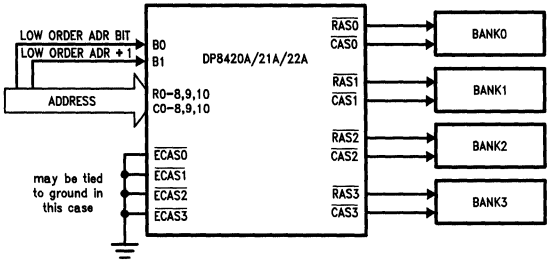


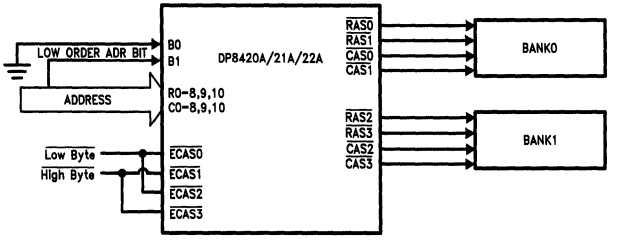
FIGURE 40. Memory Interleaving (C6, C5, C4 = 1, 1, 0 during Programming)

7.0 $\overline{RAS}$ and $\overline{CAS}$ Configuration Modes (Continued)



TL/F/8588–D4

FIGURE 41a. DRAM Array Setup for 4 Banks Using Address Pipelining (C6, C5, C4 = 1, 1, 1 or C6, C5, C4 = 0, 1, 0 (Also Allowing Error Scrubbing) during Programming)

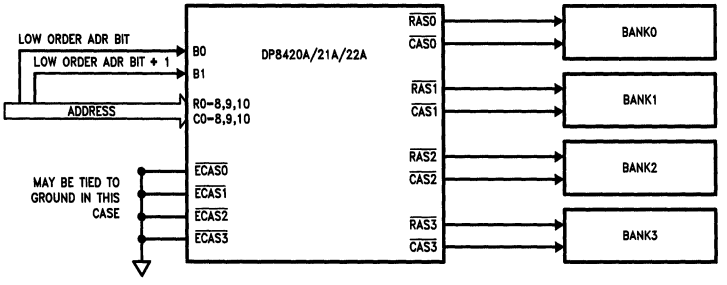


TL/F/8588–D5

FIGURE 41b. DRAM Array Setup for Address Pipelining with 2 Banks (C6, C5, C4 = 1, 0, 1 or C6, C5, C4 = 0, 0, 1 (Also Allowing Error Scrubbing) during Programming)

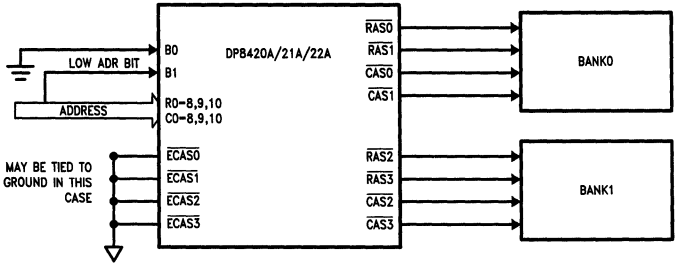
7.4 ERROR SCRUBBING

In error scrubbing during refresh, the user selects one, two or four $\overline{RAS}$ and $\overline{CAS}$ outputs per bank. When performing error detection and correction, memory is always accessed as words. Since the $\overline{CAS}$ signals are not used to select individual bytes, the $\overline{ECAS}$ inputs can be tied low as shown in *Figures 42a* and *42b*.



TL/F/8588–D6

FIGURE 42a. DRAM Array Setup for 4 Banks Using Error Scrubbing (C6, C5, C4 = 0, 1, 0 during Programming)



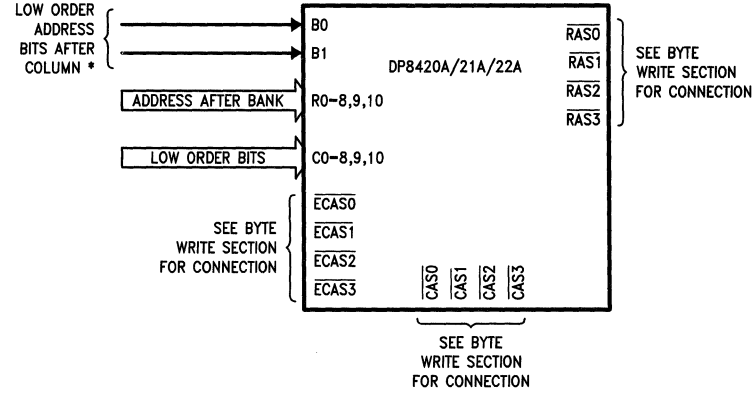
TL/F/8588–D7

FIGURE 42b. DRAM Array Setup for Error Scrubbing with 2 Banks (C6, C5, C4 = 0, 0, 1 during Programming)

7.0 $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ Configuration Modes (Continued)

7.5 PAGE/BURST MODE

In a static column, page or burst mode system, the least significant bits must be tied to the column address in order to ensure that the page/burst accesses are to sequential memory addresses, as shown in *Figure 43.* In a nibble mode system, the least significant bits must be tied to the highest column and row address bits in order to ensure that sequential address bits are the "nibble" bits for nibble mode accesses *(Figure 43).* The $\overline{\text{ECAS}}$ inputs may then be toggled with the DP8420A/21A/22A's address latches in fall-through mode, while $\overline{\text{AREQ}}$ is asserted. The $\overline{\text{ECAS}}$ inputs can also be used to select individual bytes. When using nibble mode DRAMS, the third and fourth address bits can be tied to the bank select inputs to perform memory interleaving. In page or static column modes, the two address bits after the page size can be tied to the bank select inputs to select a new bank if the page size is exceeded.



TL/F/8588–D8

*See table below for row, column & bank address bit map. A0,A1 are used for byte addressing in this example.

Addresses	Nibble Mode*	Page Mode/Static Column Mode Page Size			
		256 Bits/Page	512 Bits/Page	1024 Bits/Page	2048 Bits/Page
Column Address	C9,R9 = A2,A3 C0–8 = X	C0–7 = A2–9 C8–10 = X	C0–8 = A2–10 C9,10 = X	C0–9 = A2–11 C10 = X	C0–10 = A2–12
Row Address	X	X	X	X	X
B0 B1	A4 A5	A10 A11	A11 A12	A12 A13	A13 A14

Assume that the least significant address bits are used for byte addressing. Given a 32-bit system A0,A1 would be used for byte addressing.

X = DON'T CARE, the user can do as he pleases.

*Nibble mode values for R and C assume a system using 1 Mbit DRAMs.

FIGURE 43. Page, Static Column, Nibble Mode System

8.0 Programming and Resetting

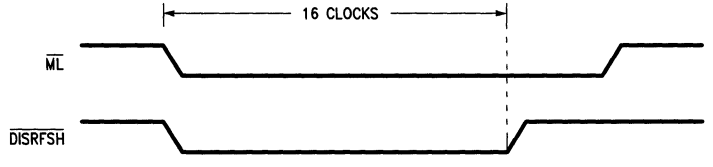
The DP8420A/21A/22A must be programmed by one of two possible programming sequences before it can be used. At power up, the DP8420A/21A/22A programming bits are in an undefined state. All internal latches and flip-flops are cleared. After programming, the DP8420A/21A/22A enters a 60 ms initialization period. During this initialization period, the DP8420A/21A/22A performs refreshes about every 15 μs; this makes further DRAM warmup cycles unnecessary. The chip can be programmed as many times as the user wishes. After the first programming, the 60 ms initialization period will not be entered into unless the chip is reset. During the 60 ms initialization period, $\overline{RFIP}$ is asserted. The actual initialization time period is given by the following formula:

T = 4096*(Clock Divisor Select)
*(Refresh Clock Fine Tune)
/(DELCK Frequency)

8.1 EXTERNAL RESET

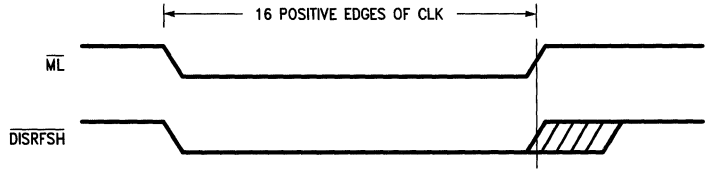
At power up, all internal latches and flip-flops are cleared. The power up state can again be entered by asserting $\overline{ML}$ and $\overline{DISRFSH}$ for 16 positive edges of CLK. After resetting if the user negates $\overline{DISRFSH}$ before negating $\overline{ML}$ as shown in *Figure 44a,* $\overline{ML}$ negated will program the chip. If $\overline{ML}$ is negated before or at the same time as $\overline{DISRFSH}$ as shown in *Figure 44b*, the chip will not be programmed. After the chip is programmed, the 60 ms initialization period will be entered into if this is the first programming after power up or reset.

It is recommended that the user perform a hardware reset of the DP8420A/21A/22A before programming and using the chip.



TL/F/8588–E2

FIGURE 44a. Chip Reset and Programmed



TL/F/8588–E1

FIGURE 44b. Chip Reset but Not Programmed

8.0 Programming and Resetting (Continued)

8.2 PROGRAMMING METHODS

The DP8420A/21A/22A must be programmed by one of two possible programming sequences before it can be used.

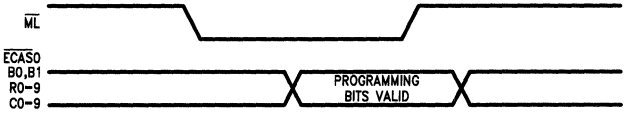
8.2.1 MODE LOAD ONLY PROGRAMMING

MODE LOAD, $\overline{ML}$, asserted enables an internal 23-bit programmable register. To use this method, the user asserts $\overline{ML}$, enabling the internal programming register. After $\overline{ML}$ is asserted, a valid programming selection is placed on the address bus (and $\overline{ECAS0}$), then $\overline{ML}$ is negated. When $\overline{ML}$ is negated, the value on the address bus (and $\overline{ECAS0}$) is latched into the internal programming register and the DP8420A/21A/22A is programmed, as shown in *Figure 45a*. After $\overline{ML}$ is negated, the DP8420A/21A/22A will enter the 60 ms initialization period only if this is the first programming after power up or reset.

Using this method, a set of transceivers on the address bus can be put at TRI-STATE® by the system reset signal. A combination of pull-up and pull-down resistors can be used on the address inputs of the DP8420A/21A/22A to select the programming values, as shown in *Figure 45b*.

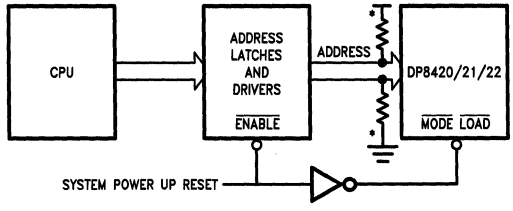
8.2.2 CHIP SELECTED ACCESS PROGRAMMING

The chip can also be programmed by asserting $\overline{ML}$ and performing a chip selected access. $\overline{ADS}$ (or ALE) is disabled internally until after programming. To program the chip using this method, $\overline{ML}$ is asserted. After $\overline{ML}$ is asserted, $\overline{CS}$ is asserted and a valid programming selection is placed on the address bus. When $\overline{AREQ}$ is asserted, the chip is programmed with the programming selection on the address bus. After $\overline{AREQ}$ is negated, $\overline{ML}$ can be negated as shown in *Figure 46a*.



TL/F/8588–D9

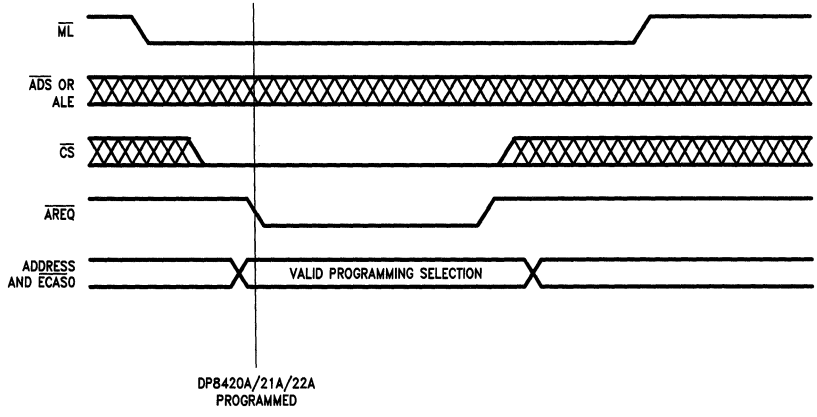
FIGURE 45a. Mode Load Only Programming



TL/F/8588–10

*Pull-Up or Pull-Down Resistors on Each Address Input

FIGURE 45b. Programming during System Reset

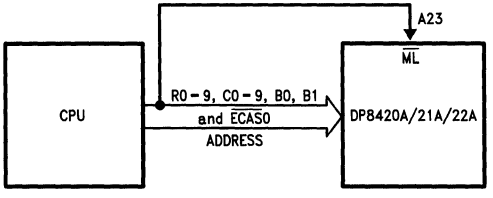


TL/F/8588–E0

FIGURE 46a. $\overline{CS}$ Access Programming

8.0 Programming and Resetting (Continued)

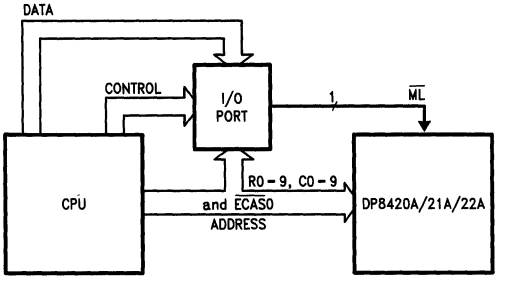
Using this method, various programming schemes can be used. For example if extra upper address bits are available, an unused high order address bit can be tied to the signal $\overline{ML}$. Using this method, one need only write to a page of memory, thus asserting the high order bit and in turn programming the chip as shown in *Figure 46b*.



TL/F/8588–11

FIGURE 46b. Programming the DP8420A/21A/22A through the Address Bus Only

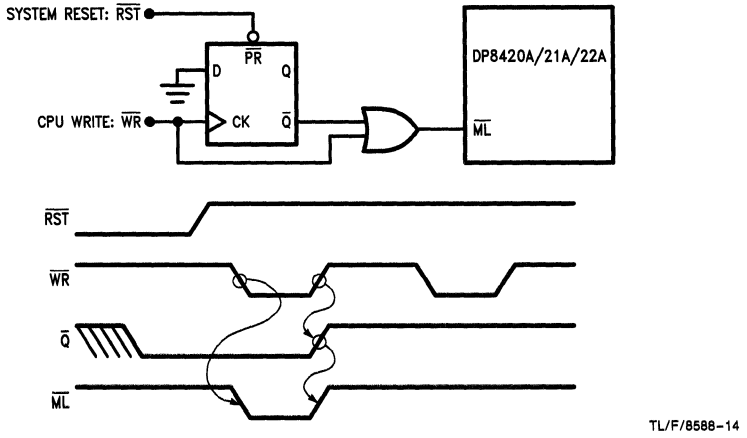
An I/O port can also be used to assert $\overline{ML}$. After $\overline{ML}$ is asserted, a chip selected access can be performed to program the chip. After the chip selected access, $\overline{ML}$ can be negated through the I/O port as shown in *Figure 46c*.



TL/F/8588–12

FIGURE 46c. Programming the DP8420A/21A/22A through the Address Bus and an I/O Port

Another simple way the chip can be programmed is the first write after system reset. This method requires only a flip-flop and an OR gate as shown in *Figure 46d*. At reset, the flip-flop is preset, which pulls the $\overline{Q}$ output low. Since $\overline{WR}$ is negated, $\overline{ML}$ is not enabled. The first write access is used to program the chip. When $\overline{WR}$ is asserted, $\overline{ML}$ is asserted. $\overline{WR}$ negated clocks the flip-flop, negates $\overline{ML}$, and programs the DP8420A/21A/22A with the address and $\overline{ECAS0}$ available at that time. $\overline{CS}$ does not need to be asserted using this method.



TL/F/8588–14

FIGURE 46d. Programming the DP8420A/21A/22A on the First CPU Write after Power Up

8.0 Programming and Resetting (Continued)

8.3 PROGRAMMING BIT DEFINITIONS

Symbol	Description
ECAS0	**Extend CAS/Refresh Request Select**
0	The CASn outputs will be negated with the RASn outputs when AREQ (or AREQB, DP8422A only) is negated. The WE output pin will function as write enable.
1	The CASn outputs will be negated, during an access (Port A (or Port B, DP8422A only)) when their corresponding ECASn inputs are negated. This feature allows the CAS outputs to be extended beyond the RAS outputs negating. Scrubbing refreshes are NOT affected. During scrubbing refreshes the CAS outputs will negate along with the RAS outputs regardless of the state of the ECAS inputs. The WE output will function as ReFresh ReQuest (RFRQ) when this mode is programmed.
B1	**Access Mode Select**
0	**ACCESS MODE 0:** ALE pulsing high sets an internal latch. On the next positive edge of CLK, the access (RAS) will start. AREQ will terminate the access.
1	**ACCESS MODE 1:** ADS asserted starts the access (RAS) immediately. AREQ will terminate the access.
B0	**Address Latch Mode**
0	ADS or ALE asserted for Port A or AREQB asserted for Port B with the appropriate GRANT latch the input row, column and bank address.
1	The row, column and bank latches are fall through.
C9	**Delay CAS during WRITE Accesses**
0	CAS is treated the same for both READ and WRITE accesses.
1	During WRITE accesses, CAS will be asserted by the event that occurs last: CAS asserted by the internal delay line or CAS asserted on the positive edge of CLK after RAS is asserted.
C8	**Row Address Hold Time**
0	Row Address Hold Time = 25 ns minimum
1	Row Address Hold Time = 15 ns minimum
C7	**Column Address Setup Time**
0	Column Address Setup Time = 10 ns minimum
1	Column Address Setup Time = 0 ns minimum
C6, C5, C4	**RAS and CAS Configuration Modes/Error Scrubbing during Refresh**
0, 0, 0	RAS0–3 and CAS0–3 are all selected during an access. ECASn must be asserted for CASn to be asserted. B0 and B1 are not used during an access. Error scrubbing during refresh.
0, 0, 1	RAS and CAS pairs are selected during an access by B1. ECASn must be asserted for CASn to be asserted. B1 = 0 during an access selects RAS0–1 and CAS0–1. B1 = 1 during an access selects RAS2–3 and CAS2–3. B0 is not used during an Access. Error scrubbing during refresh.
0, 1, 0	RAS and CAS singles are selected during an access by B0–1. ECASn must be asserted for CASn to be asserted. B1 = 0, B0 = 0 during an access selects RAS0 and CAS0. B1 = 0, B0 = 1 during an access selects RAS1 and CAS1. B1 = 1, B0 = 0 during an access selects RAS2 and CAS2. B1 = 1, B0 = 1 during an access selects RAS3 and CAS3. Error scrubbing during refresh.
0, 1, 1	RAS0–3 and CAS0–3 are all selected during an access. ECASn must be asserted for CASn to be asserted. B1, B0 are not used during an access. No error scrubbing. (RAS only refreshing)
1, 0, 0	RAS pairs are selected by B1. CAS0–3 are all selected. ECASn must be asserted for CASn to be asserted. B1 = 0 during an access selects RAS0–1 and CAS0–3. B1 = 1 during an access selects RAS2–3 and CAS0–3. B0 is not used during an access. No error scrubbing.

8.0 Programming and Resetting (Continued)

8.3 PROGRAMMING BIT DEFINITIONS (Continued)

Symbol	Description
C6, C5, C4	**$\overline{RAS}$ and $\overline{CAS}$ Configuration Modes** (Continued)
1, 0, 1	$\overline{RAS}$ and $\overline{CAS}$ pairs are selected by B1. $\overline{ECAS}$n must be asserted for $\overline{CAS}$n to be asserted. B1 = 0 during an access selects $\overline{RAS}$0–1 and $\overline{CAS}$0–1. B1 = 1 during an access selects $\overline{RAS}$2–3 and $\overline{CAS}$2–3. B0 is not used during an access. No error scrubbing.
1, 1, 0	$\overline{RAS}$ singles are selected by B0–1. $\overline{CAS}$0–3 are all selected. $\overline{ECAS}$n must be asserted for $\overline{CAS}$n to be asserted. B1 = 0, B0 = 0 during an access selects $\overline{RAS}$0 and $\overline{CAS}$0–3. B1 = 0, B0 = 1 during an access selects $\overline{RAS}$1 and $\overline{CAS}$0–3. B1 = 1, B0 = 0 during an access selects $\overline{RAS}$2 and $\overline{CAS}$0–3. B1 = 1, B0 = 1 during an access selects $\overline{RAS}$3 and $\overline{CAS}$0–3. No error scrubbing.
1, 1, 1	$\overline{RAS}$ and $\overline{CAS}$ singles are selected by B0, 1. $\overline{ECAS}$n must be asserted for $\overline{CAS}$n to be asserted. B1 = 0, B0 = 0 during an access selects $\overline{RAS}$0 and $\overline{CAS}$0. B1 = 0, B0 = 1 during an access selects $\overline{RAS}$1 and $\overline{CAS}$1. B1 = 1, B0 = 0 during an access selects $\overline{RAS}$2 and $\overline{CAS}$2. B1 = 1, B0 = 1 during an access selects $\overline{RAS}$3 and $\overline{CAS}$3. No error scrubbing.
C3	**Refresh Clock Fine Tune Divisor**
0	Divide delay line/refresh clock further by 30 (If DELCLK/Refresh Clock Clock Divisor = 2 MHz = 15 μs refresh period).
1	Divide delay line/refresh clock further by 26 (If DELCLK/Refresh Clock Clock Divisor = 2 MHz = 13 μs refresh period).
C2, C1, C0	**Delay Line/Refresh Clock Divisor Select**
0, 0, 0	Divide DELCLK by 10 to get as close to 2 MHz as possible.
0, 0, 1	Divide DELCLK by 9 to get as close to 2 MHz as possible.
0, 1, 0	Divide DELCLK by 8 to get as close to 2 MHz as possible.
0, 1, 1	Divide DELCLK by 7 to get as close to 2 MHz as possible.
1, 0, 0	Divide DELCLK by 6 to get as close to 2 MHz a possible.
1, 0, 1	Divide DELCLK by 5 to get as close to 2 MHz as possible.
1, 1, 0	Divide DELCLK by 4 to get as close to 2 MHz as possible.
1, 1, 1	Divide DELCLK by 3 to get as close to 2 MHz as possible.
R9	**Refresh Mode Select**
0	$\overline{RAS}$0–3 will all assert and negate at the same time during a refresh.
1	Staggered Refresh. $\overline{RAS}$ outputs during refresh are separated by one positive clock edge. Depending on the configuration mode chosen, either one or two $\overline{RAS}$s will be asserted.
R8	**Address Pipelining Select**
0	Address pipelining is selected. The DRAM controller will switch the DRAM column address back to the row address after guaranteeing the column address hold time.
1	Non-address pipelining is selected. The DRAM controller will hold the column address on the DRAM address bus until the access $\overline{RAS}$s are negated.
R7	**$\overline{WAIT}$ or $\overline{DTACK}$ Select**
0	$\overline{WAIT}$ type output is selected.
1	$\overline{DTACK}$ (Data Transfer ACKnowledge) type output is selected.
R6	**Add Wait States to the Current Access if $\overline{WAITIN}$ is Low**
0	$\overline{WAIT}$ or $\overline{DTACK}$ will be delayed by one additional positive edge of CLK.
1	$\overline{WAIT}$ or $\overline{DTACK}$ will be delayed by two additional positive edges of CLK.

8.0 Programming and Resetting (Continued)

8.3 PROGRAMMING BIT DEFINITIONS (Continued)

Symbol	Description
R5, R4	**$\overline{\text{WAIT}}/\overline{\text{DTACK}}$ during Burst (See Section 5.1.2 or 5.2.2)**
0, 0	NO WAIT STATES; If R7 = 0 during programming, $\overline{\text{WAIT}}$ will remain negated during burst portion of access. If R7 = 1 programming, $\overline{\text{DTACK}}$ will remain asserted during burst portion of access.
0, 1	1T; If R7 = 0 during programming, $\overline{\text{WAIT}}$ will assert when the $\overline{\text{ECAS}}$ inputs are negated with $\overline{\text{AREQ}}$ asserted. $\overline{\text{WAIT}}$ will negate from the positive edge of CLK after the $\overline{\text{ECAS}}$s have been asserted. If R7 = 1 during programming, $\overline{\text{DTACK}}$ will negate when the $\overline{\text{ECAS}}$ inputs are negated with $\overline{\text{AREQ}}$ asserted. $\overline{\text{DTACK}}$ will assert from the positive edge of CLK after the $\overline{\text{ECAS}}$s have been asserted.
1, 0	½T; If R7 = 0 during programming, $\overline{\text{WAIT}}$ will assert when the $\overline{\text{ECAS}}$ inputs are negated with $\overline{\text{AREQ}}$ asserted. $\overline{\text{WAIT}}$ will negate on the negative level of CLK after the $\overline{\text{ECAS}}$s have been asserted. If R7 = 1 during programming, $\overline{\text{DTACK}}$ will negate when the $\overline{\text{ECAS}}$ inputs are negated with $\overline{\text{AREQ}}$ asserted. $\overline{\text{DTACK}}$ will assert from the negative level of CLK after the $\overline{\text{ECAS}}$s have been asserted.
1, 1	0T; If R7 = 0 during programming, $\overline{\text{WAIT}}$ will assert when the $\overline{\text{ECAS}}$ inputs are negated. $\overline{\text{WAIT}}$ will negate when the $\overline{\text{ECAS}}$ inputs are asserted. If R7 = 1 during programming, $\overline{\text{DTACK}}$ will negate when the $\overline{\text{ECAS}}$ inputs are negated. $\overline{\text{DTACK}}$ will assert when the $\overline{\text{ECAS}}$ inputs are asserted.
R3, R2	**$\overline{\text{WAIT}}/\overline{\text{DTACK}}$ Delay Times (See Section 5.1.1 or 5.2.1)**
0, 0	NO WAIT STATES; If R7 = 0 during programming, $\overline{\text{WAIT}}$ will remain high during non-delayed accesses. $\overline{\text{WAIT}}$ will negate when $\overline{\text{RAS}}$ is negated during delayed accesses. NO WAIT STATES; If R7 = 1 during programming, $\overline{\text{DTACK}}$ will be asserted when $\overline{\text{RAS}}$ is asserted.
0, 1	½T; If R7 = 0 during programming, $\overline{\text{WAIT}}$ will negate on the negative level of CLK, after the access $\overline{\text{RAS}}$. 1T; If R7 = 1 during programming, $\overline{\text{DTACK}}$ will be asserted on the positive edge of CLK after the access $\overline{\text{RAS}}$.
1, 0	NO WAIT STATES, ½T; If R7 = 0 during programming, $\overline{\text{WAIT}}$ will remain high during non-delayed accesses. $\overline{\text{WAIT}}$ will negate on the negative level of CLK, after the access $\overline{\text{RAS}}$, during delayed accesses. ½T; If R7 = 1 during programming, $\overline{\text{DTACK}}$ will be asserted on the negative level of CLK after the access $\overline{\text{RAS}}$.
1, 1	1T; If R7 = 0 during programming, $\overline{\text{WAIT}}$ will negate on the positive edge of CLK after the access $\overline{\text{RAS}}$. 1½T; If R7 = 1 during programming, $\overline{\text{DTACK}}$ will be asserted on the negative level of CLK after the positive edge of CLK after the access $\overline{\text{RAS}}$.

8.0 Programming and Resetting (Continued)

8.3 PROGRAMMING BIT DEFINITIONS (Continued)

Symbol	Description
R1, R0	**$\overline{RAS}$ Low and $\overline{RAS}$ Precharge Time**
0, 0	$\overline{RAS}$ asserted during refresh = 2 positive edges of CLK. $\overline{RAS}$ precharge time = 1 positive edge of CLK. $\overline{RAS}$ will start from the first positive edge of CLK after GRANTB transitions (DP8422A).
0, 1	$\overline{RAS}$ asserted during refresh = 3 positive edges of CLK. $\overline{RAS}$ precharge time = 2 positive edges of CLK. $\overline{RAS}$ will start from the second positive edge of CLK after GRANTB transitions (DP8422A).
1, 0	$\overline{RAS}$ asserted during refresh = 2 positive edges of CLK. $\overline{RAS}$ precharge time = 2 positive edges of CLK. $\overline{RAS}$ will start from the first positive edge of CLK after GRANTB transitions (DP8422A).
1, 1	$\overline{RAS}$ asserted during refresh = 4 positive edges of CLK. $\overline{RAS}$ precharge time = 3 positive edges of CLK. $\overline{RAS}$ will start from the second positive edge of CLK after GRANTB transitions (DP8422A).

9.0 Test Mode

Staggered refresh in combination with the error scrubbing mode places the DP8420A/21A/22A in test mode. In this mode, the 24-bit refresh counter is divided into a 13-bit and 11-bit counter. During refreshes both counters are incremented to reduce test time.

10.0 DRAM Critical Timing Parameters

The two critical timing parameters, shown in *Figure 47*, that must be met when controlling the access timing to a DRAM are the row address hold time, tRAH, and the column address setup time, tASC. Since the DP8420A/21A/22A contain a precise internal delay line, the values of these parameters can be selected at programming time. These values will also increase and decrease if DELCLK varies from 2 MHz.

10.1 PROGRAMMABLE VALUES OF tRAH AND tASC

The DP8420A/21A/22A allow the values of tRAH and tASC to be selected at programming time. For each parameter, two choices can be selected. tRAH, the row address hold time, is measured from $\overline{RAS}$ asserted to the row address starting to change to the column address. The two choices for tRAH are 15 ns and 25 ns, programmable through address bit C8.

tASC, the column address setup time, is measured from the column address valid to $\overline{CAS}$ asserted. The two choices for tASC are 0 ns and 10 ns, programmable through address bit C7.

10.2 CALCULATION OF tRAH AND tASC

There are two clock inputs to the DP8420A/21A/22A. These two clocks, DELCLK and CLK can either be tied together to the same clock or be tied to different clocks running asynchronously at different frequencies.

The clock input, DELCLK, controls the internal delay line and refresh request clock. DELCLK should be a multiple of 2 MHz. If DELCLK is not a multiple of 2 MHz, tRAH and tASC will change. The new values of tRAH and tASC can be calculated by the following formulas:

If tRAH was programmed to equal 15 ns then tRAH = 30*(((DELCLK Divisor)* 2 MHz/(DELCLK Frequency))−1) + 15 ns.

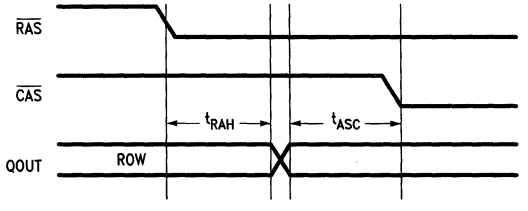
If tRAH was programmed to equal 25 ns then tRAH = 30*(((DELCLK Divisor)* 2 MHz/(DELCLK Frequency))−1) + 25 ns.

If tASC was programmed to equal 0 ns then tASC = 15* ((DELCLK Divisor)* 2 MHz/(DELCLK Frequency)) − 15 ns.

If tASC was programmed to equal 10 ns then tASC = 25* ((DELCLK Divisor)* 2 MHz/(DELCLK Frequency)) − 15 ns.

Since the values of tRAH and tASC are increased or decreased, the time to $\overline{CAS}$ asserted will also increase or decrease. These parameters can be adjusted by the following formula:

Delay to $\overline{CAS}$ = Actual Spec. + Actual tRAH − Programmed tRAH + Actual tASC − Programmed tASC.



TL/F/8588–E3

FIGURE 47. tRAH and tASC

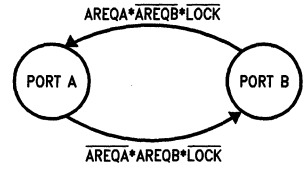
11.0 Dual Accessing Functions (DP8422A)

The DP8422A has all the functions previously described. In addition to those features, the DP8422A also has the capabilities to arbitrate among refresh, Port A and a second port, Port B. This allows two CPUs to access a common DRAM array. DRAM refresh has the highest priority followed by the currently granted port. The ungranted port has the lowest priority. The last granted port will continue to stay granted even after the access has terminated, until an access request is received from the ungranted port (see *Figure 48a*). The dual access configuration assumes that both Port A and Port B are synchronous to the system clock. If they are not synchronous to the system clock they should be externally synchronized (Ex. By running the access requests through several Flip-Flops, see *Figure 50a*).

11.1 PORT B ACCESS MODES (DP8422A)

Port B accesses are initiated from a single input, $\overline{\text{AREQB}}$. When $\overline{\text{AREQB}}$ is asserted, an access request is generated. If GRANTB is asserted and a refresh is not taking place or precharge time is not required, $\overline{\text{RAS}}$ will be asserted when $\overline{\text{AREQB}}$ is asserted. Once $\overline{\text{AREQB}}$ is asserted, it must stay asserted until the access is over. $\overline{\text{AREQB}}$ negated, negates $\overline{\text{RAS}}$ as shown in *Figure 48b*. Note that if $\overline{\text{ECAS}}0 = 1$ during programming the $\overline{\text{CAS}}$ outputs may be held asserted (beyond $\overline{\text{RAS}}$n negating) by continuing to assert the appropriate $\overline{\text{ECAS}}$n inputs (the same as Port A accesses). If Port B is not granted, the access will begin on the first or second positive edge of CLK after GRANTB is asserted (See R0, R1 programming bit definitions) as shown in *Figure 48c*, assuming that Port A is not accessing the DRAM ($\overline{\text{CS}}$, $\overline{\text{ADS}}$/ALE and $\overline{\text{AREQ}}$) and $\overline{\text{RAS}}$ precharge for the particular bank has completed. It is important to note that for GRANTB to transition to Port B, Port A must **not** be requesting an access at a rising clock edge (or locked) and Port B must be requesting an access at that rising clock edge. Port A can request an access through $\overline{\text{CS}}$ and $\overline{\text{ADS}}$/ALE or $\overline{\text{CS}}$ and $\overline{\text{AREQ}}$. Therefore during an interleaved access where $\overline{\text{CS}}$ and $\overline{\text{ADS}}$/ALE become asserted before $\overline{\text{AREQ}}$ from the previous access is negated, Port A will retain GRANTB = 0 whether $\overline{\text{AREQB}}$ is asserted or not.

Since there is no chip select for Port B, $\overline{\text{AREQB}}$ must incorporate this signal. This mode of accessing is similar to Mode 1 accessing for Port A.



TL/F/8588–F9

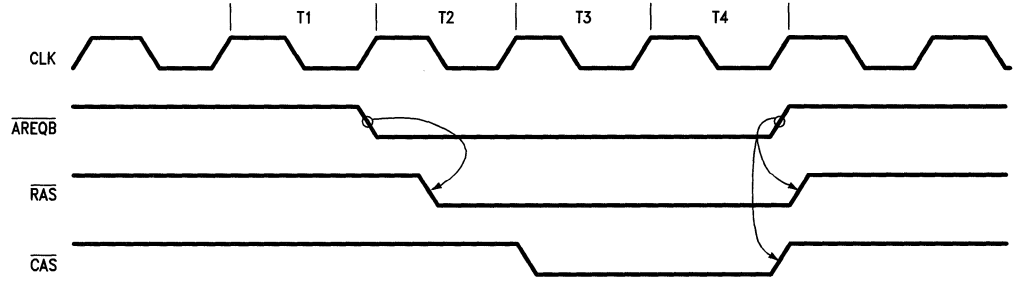
Explanation of Terms

AREQA = Chip Selected access request from Port A

AREQB = Chip Selected access request from Port B

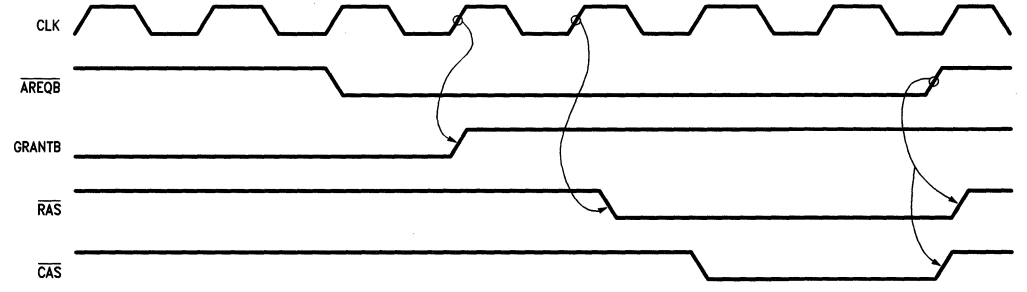
LOCK = Externally controlled LOCKing of the Port that is currently GRANTed.

FIGURE 48a. DP8422A PORT A/PORT B ARBITRATION STATE DIAGRAM. This arbitration may take place during the "ACCESS" or "REFRESH" state (see *Figure 7a*).



TL/F/8588–E4

FIGURE 48b. Access Request for Port B



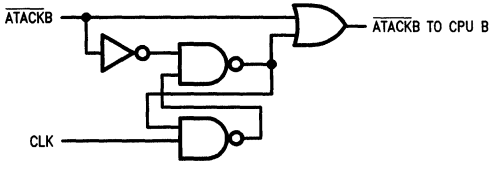
TL/F/8588–E5

FIGURE 48c. Delayed Port B Access

11.0 Dual Accessing Functions (DP8422A) (Continued)

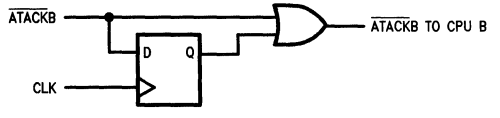
11.2 PORT B WAIT STATE SUPPORT (DP8422A)

Advanced transfer acknowledge for Port B, $\overline{\text{ATACKB}}$, is used for wait state support for Port B. This output will be asserted when $\overline{\text{RAS}}$ for the Port B access is asserted, as shown in *Figures 49a* and *49b*. Once asserted, this output will stay asserted until $\overline{\text{AREQB}}$ is negated. With external logic, $\overline{\text{ATACKB}}$ can be made to interface to any CPU's wait input as shown in *Figure 49c*.



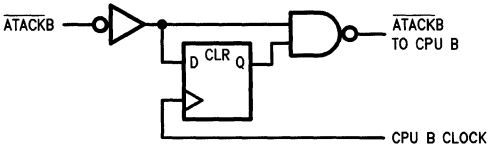
TL/F/8588–30

A) Extend $\overline{\text{ATACK}}$ to ½T (½ Clock) after $\overline{\text{RAS}}$ goes low.



TL/F/8588–31

B) Extend $\overline{\text{ATACK}}$ to 1T after $\overline{\text{RAS}}$ goes low.



TL/F/8588–32

C) Synchronize $\overline{\text{ATACKB}}$ to CPU B Clock. This is useful if CPU B runs asynchronous to the DP8422.

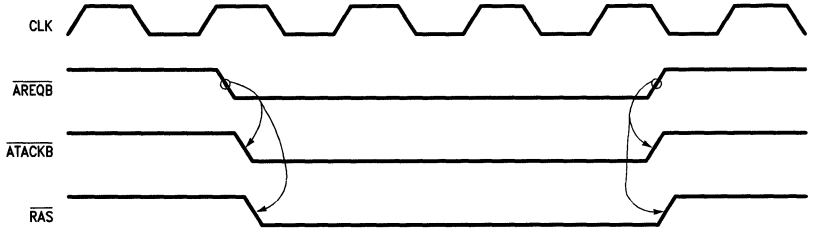
FIGURE 49c. Modifying Wait Logic for Port B

11.3 COMMON PORT A AND PORT B DUAL PORT FUNCTIONS

An input, $\overline{\text{LOCK}}$, and an output, GRANTB, add additional functionality to the dual port arbitration logic. $\overline{\text{LOCK}}$ allows Port A or Port B to lock out the other port from the DRAM. When a Port is locked out of the DRAM, wait states will be inserted into its access cycle until it is allowed to access memory. GRANTB is used to multiplex the input control signals and addresses to the DP8422A.

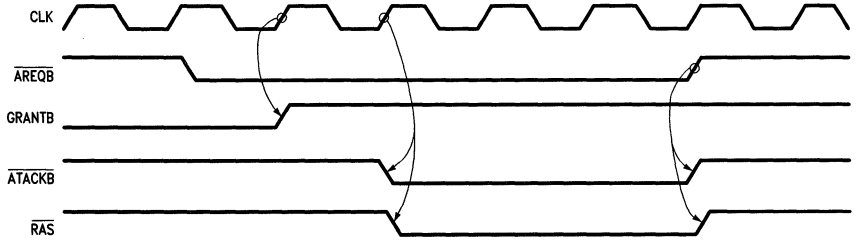
11.3.1 GRANTB Output

The output GRANTB determines which port has current access to the DRAM array. GRANTB asserted signifies Port B has access. GRANTB negated signifies Port A has access to the DRAM array.



TL/F/8588–E6

FIGURE 49a. Non-Delayed Port B Access

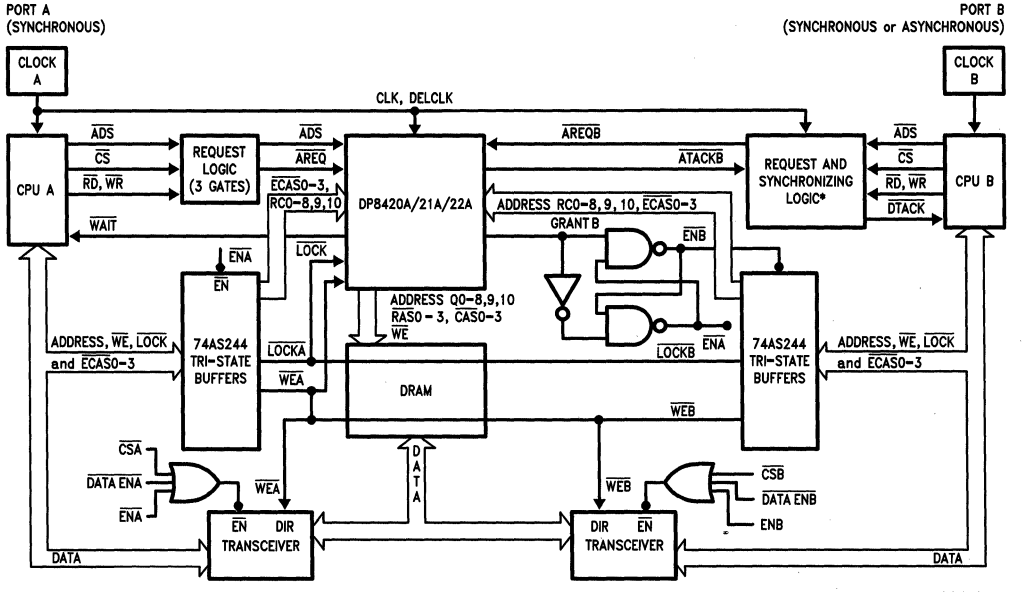


TL/F/8588–E7

FIGURE 49b. Delayed Port B Access

11.0 Dual Accessing Functions (DP8422A) (Continued)

Since the DP8422A has only one set of address inputs, the signal is used, with the addition of buffers, to allow the currently granted port's addresses to reach the DP8422A. The signals which need to be bufferred are R0–10, C0–10, B0–1, $\overline{ECAS0}$–3, $\overline{WE}$, and $\overline{LOCK}$. All other inputs are not common and do not have to be buffered as shown in *Figure 50a.* If a Port, which is not currently granted, tries to access the DRAM array, the GRANTB output will transition from a rising clock edge from $\overline{AREQ}$ or $\overline{AREQB}$ negating and will preceed the $\overline{RAS}$ for the access by one or two clock periods. GRANTB will then stay in this state until the other port requests an access and the currently granted port is not accessing the DRAM as shown in *Figure 50b.*

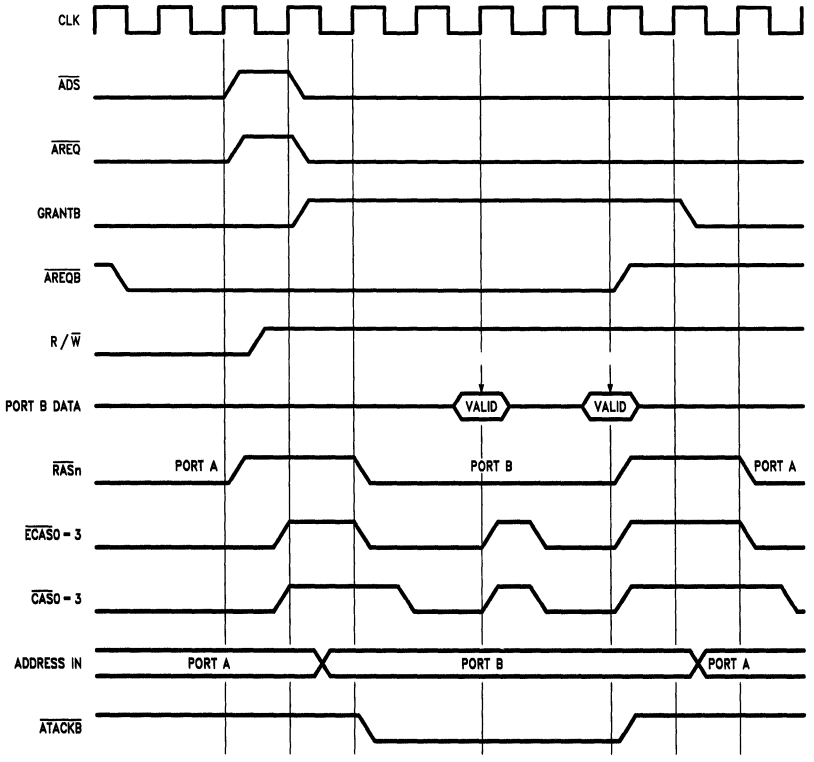


TL/F/8588–55

*If Port B is synchronous the Request Synchronizing logic will not be required.

FIGURE 50a. Dual Accessing with the DP8422A (System Block Diagram)

11.0 Dual Accessing Functions (DP8422A) (Continued)

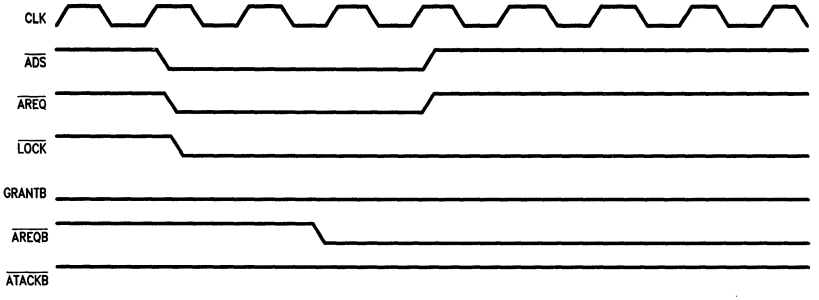


TL/F/8588–29

FIGURE 50b. Wait States during a Port B Access

11.3.2 LOCK Input

When the LOCK input is asserted, the currently granted port can "lock out" the other port through the insertion of wait states to that port's access cycle. LOCK does not disable refreshes, it only keeps GRANTB in the same state even if the other port requests an access, as shown in *Figure 51a*. LOCK can be used by either port.



TL/F/8588–E8

FIGURE 51. LOCK Function

12.0 Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Temperature under Bias ..... 0°C to +70°C
Storage Temperature ..... −65°C to +150°C
All Input or Output Voltage with Respect to GND ..... −0.5V to +7V
Power Dissipation @ 20 MHz 0.5W
ESD Rating to be determined.

13.0 DC Electrical Characteristics T_A = 0°C to +70°C, V_{CC} = 5V ±10%, GND = 0V

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IH}	Logical 1 Input Voltage	Tested with a Limited Functional Pattern	2.0		V_{CC} + 0.5	V
V_{IL}	Logical 0 Input Voltage	Tested with a Limited Functional Pattern	−0.5		0.8	V
V_{OH1}	Q and $\overline{WE}$ Outputs	I_{OH} = −10 mA	V_{CC} − 1.0			V
V_{OL1}	Q and $\overline{WE}$ Outputs	I_{OL} = 10 mA			0.5	V
V_{OH2}	All Outputs except Qs, $\overline{WE}$	I_{OH} = −3 mA	V_{CC} − 1.0			V
V_{OL2}	All Outputs except Qs, $\overline{WE}$	I_{OL} = 3 mA			0.5	V
I_{IN}	Input Leakage Current	V_{IN} = V_{CC} or GND	−10		10	μA
$I_{IL\ ML}$	$\overline{ML}$ Input Current (Low)	V_{IN} = GND			200	μA
I_{CC1}	Standby Current	CLK at 8 MHz (V_{IN} = V_{CC} or GND)		6	15	mA
I_{CC1}	Standby Current	CLK at 20 MHz (V_{IN} = V_{CC} or GND)		8	17	mA
I_{CC1}	Standby Current	CLK at 25 MHz (V_{IN} = V_{CC} or GND)		10	20	mA
I_{CC2}	Supply Current	CLK at 8 MHz (Inputs Active) (I_{LOAD} = 0) (V_{IN} = V_{CC} or GND)		20	40	mA
I_{CC2}	Supply Current	CLK at 20 MHz (Inputs Active) (I_{LOAD} = 0) (V_{IN} = V_{CC} or GND)		40	75	mA
I_{CC2}	Supply Current	CLK at 25 MHz (Inputs Active) (I_{LOAD} = 0) (V_{IN} = V_{CC} or GND)		50	95	mA
C_{IN}*	Input Capacitance	f_{IN} at 1 MHz			10	pF

***Note: C_{IN} is not 100% tested.**

14.0 AC Timing Parameters: DP8420A/DP8421A/DP8422A

Two speed selections are given, the DP8420A/21A/22A-20 and the DP8420A/21A/22A-25. The differences between the two parts are the maximum operating frequencies of the input CLKs and the maximum delay specifications. Low frequency applications may use the "−25" part to gain improved timing.

The AC timing parameters are grouped into sectional numbers as shown below. These numbers also refer to the timing diagrams.

1–36 Common parameters to all modes of operation

50–56 Difference parameters used to calculate;
$\overline{RAS}$ low time,
$\overline{RAS}$ precharge time,
$\overline{CAS}$ high time and
$\overline{CAS}$ low time

100–121 Common dual access parameters used for Port B accesses and inputs and outputs used only in dual accessing

200–212 Refresh parameters

300–315 Mode 0 access parameters used in both single and dual access applications

400–416 Mode 1 access parameters used in both single and dual access applications

450–455 Special Mode 1 access parameters which supersede the 400–416 parameters when dual accessing

500–506 Programming parameters

Unless otherwise stated V_{CC} = 5.0V ±10%, 0 < T_A < 70°C, the output load capacitance is typical for 4 banks of 18 DRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:

C_L = 50 pF loads on all outputs except
C_L = 150 pF loads on Q0–8, 9, 10 and $\overline{WE}$; or

C_H = 50 pF loads on all outputs except
C_H = 125 pF loads on $\overline{RAS}$0–3 and $\overline{CAS}$0–3 and
C_H = 380 pF loads on Q0–8, 9, 10 and $\overline{WE}$.

14.0 AC Timing Parameters: DP8420A/DP8421A/DP8422A (Continued)

Number	Symbol	Common Parameter Description	8420A/21A/22A-20				8420A/21A/22A-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
1	fCLK	CLK Frequency	0	20	0	20	0	25	0	25
2	tCLKP	CLK Period	50		50		40		40	
3, 4	tCLKPW	CLK Pulse Width	15		15		12		12	
5	fDCLK	DELCLK Frequency	5	20	5	20	5	20	5	20
6	tDCLKP	DELCLK Period	50	200	50	200	50	200	50	200
7, 8	tDCLKPW	DELCLK Pulse Width	15		15		12		12	
9a	tPRASCAS0	$\overline{RAS}$ Asserted to $\overline{CAS}$ Asserted (tRAH = 15 ns, tASC = 0 ns)	30		30		30		30	
9b	tPRASCAS1	$\overline{RAS}$ Asserted to $\overline{CAS}$ Asserted (tRAH = 15 ns, tASC = 10 ns)	40		40		40		40	
9c	tPRASCAS2	($\overline{RAS}$ Asserted to $\overline{CAS}$ Asserted (tRAH = 25 ns, tASC = 0 ns)	40		40		40		40	
9d	tPRASCAS3	($\overline{RAS}$ Asserted to $\overline{CAS}$ Asserted (tRAH = 25 ns, tASC = 10 ns)	50		50		50		50	
10a	tRAH	Row Address Hold Time (tRAH = 15)	15		15		15		15	
10b	tRAH	Row Address Hold Time (tRAH = 25)	25		25		25		25	
11a	tASC	Column Address Setup Time (tASC = 0)	0		0		0		0	
11b	tASC	Column Address Setup Time (tASC = 10)	10		10		10		10	
12	tPCKRAS	CLK High to $\overline{RAS}$ Asserted following Precharge		27		32		22		26
13	tPARQRAS	$\overline{AREQ}$ Negated to $\overline{RAS}$ Negated		38		43		31		35
14	tPENCL	$\overline{ECAS}$0–3 Asserted to $\overline{CAS}$ Asserted		23		31		20		27
15	tPENCH	$\overline{ECAS}$0–3 Negated to $\overline{CAS}$ Negated		25		33		20		27
16	tPARQCAS	$\overline{AREQ}$ Negated to $\overline{CAS}$ Negated		60		68		47		54
17	tPCLKWH	CLK to $\overline{WAIT}$ Negated		39		39		31		31
18	tPCLKDL0	CLK to $\overline{DTACK}$ Asserted (Programmed as $\overline{DTACK}$ of 1/2, 1, 1½ or if $\overline{WAITIN}$ is Asserted)		33		33		28		28
19	tPEWL	$\overline{ECAS}$ Negated to $\overline{WAIT}$ Asserted during a Burst Access		44		44		36		36
20	tSECK	$\overline{ECAS}$ Asserted Setup to CLK High to Recognize the Rising Edge of CLK during a Burst Access	24		24		19		19	

14.0 AC Timing Parameters: DP8420A/DP8421A/DP8422A (Continued)

Unless otherwise stated V_{CC} = 5.0V ±10%, 0°C < T_A < 70°C, the output load capacitance is typical for 4 banks of 18 DRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:

C_L = 50 pF loads on all outputs except
C_L = 150 pF loads on Q0–8, 9, 10 and $\overline{WE}$; or

C_H = 50 pF loads on all outputs except
C_H = 125 pF loads on $\overline{RAS}$0–3 and $\overline{CAS}$0–3 and
C_H = 380 pF loads on Q0–8, 9, 10 and $\overline{WE}$.

Number	Symbol	Common Parameter Description	8420A/21A/22A-20				8420A/21A/22A-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
21	tPEDL	$\overline{ECAS}$ Asserted to $\overline{DTACK}$ Asserted during a Burst Access (Programmed as $\overline{DTACK}$0)		48		48		38		38
22	tPEDH	$\overline{ECAS}$ Negated to $\overline{DTACK}$ Negated during a Burst Access		49		49		38		38
23	tSWCK	$\overline{WAITIN}$ Asserted Setup to CLK	5		5		5		5	
24	tPWINWEH	$\overline{WIN}$ Asserted to $\overline{WE}$ Asserted		34		44		27		37
25	tPWINWEL	$\overline{WIN}$ Negated to $\overline{WE}$ Negated		34		44		27		37
26	tPAQ	Row, Column Address Valid to Q0–8, 9, 10 Valid		29		38		26		35
27	tPCINCQ	COLINC Asserted to Q0–8, 9, 10 Incremented		34		43		30		39
28	tSCINEN	COLINC Asserted Setup to $\overline{ECAS}$ Asserted to Ensure tASC = 0 ns	18		19		17		19	
29a	tSARQCK1	$\overline{AREQ}$, $\overline{AREQB}$ Negated Setup to CLK High with 1 Period of Precharge	46		46		37		37	
29b	tSARQCK2	$\overline{AREQ}$, $\overline{AREQB}$ Negated Setup to CLK High with >1 Period of Precharge Programmed	19		19		15		15	
30	tPAREQDH	$\overline{AREQ}$ Negated to $\overline{DTACK}$ Negated		34		34		27		27
31	tPCKCAS	CLK High to $\overline{CAS}$ Asserted when Delayed by $\overline{WIN}$		31		39		25		32
32	tSCADEN	Column Address Setup to $\overline{ECAS}$ Asserted to Guarantee tASC = 0	14		15		14		16	
33	tWCINC	COLINC Pulse Width	20		20		20		20	
34a	tPCKCL0	CLK High to $\overline{CAS}$ Asserted following Precharge (tRAH = 15 ns, tASC = 0 ns)		81		89		72		79
34b	tPCKCL1	CLK High to $\overline{CAS}$ Asserted following Precharge (tRAH = 15 ns, tASC = 10 ns)		91		99		82		89
34c	tPCKCL2	CLK High to $\overline{CAS}$ Asserted following Precharge (tRAH = 25 ns, tASC = 0 ns)		91		99		82		89
34d	tPCKCL3	CLK High to $\overline{CAS}$ Asserted following Precharge (tRAH = 25 ns, tASC = 10 ns)		101		109		92		99
35	tCAH	Column Address Hold Time (Interleave Mode Only)	32		32		32		32	
36	tPCQR	$\overline{CAS}$ Asserted to Row Address Valid (Interleave Mode Only)		90		90		90		90

14.0 AC Timing Parameters: DP8420A/DP8421A/DP8422A (Continued)

Unless otherwise stated V_{CC} = 5.0V ±10%, 0°C < T_A < 70°C, the output load capacitance is typical for 4 banks of 18 DRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:

C_L = 50 pF loads on all outputs except
C_L = 150 pF loads on Q0–8, 9, 10 and $\overline{WE}$; or

C_H = 50 pF loads on all outputs except
C_H = 125 pF loads on $\overline{RAS}$0–3 and $\overline{CAS}$0–3 and
C_H = 380 pF loads on Q0–8, 9, 10 and $\overline{WE}$.

Number	Symbol	Difference Parameter Description	8420A/21A/22A-20				8420A/21A/22A-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
50	tD1	($\overline{AREQ}$ or $\overline{AREQB}$ Negated to $\overline{RAS}$ Negated) Minus (CLK High to $\overline{RAS}$ Asserted)		16		16		14		14
51	tD2	(CLK High to Refresh $\overline{RAS}$ Negated) Minus (CLK High to $\overline{RAS}$ Asserted)		13		13		11		11
52	tD3a	($\overline{ADS}$ Asserted to $\overline{RAS}$ Asserted (Mode 1)) Minus ($\overline{AREQ}$ Negated to $\overline{RAS}$ Negated)		4		4		4		4
53	tD3b	(CLK High to $\overline{RAS}$ Asserted (Mode 0)) Minus ($\overline{AREQ}$ Negated to $\overline{RAS}$ Negated)		4		4		4		4
54	tD4	($\overline{ECAS}$ Asserted to $\overline{CAS}$ Asserted) Minus ($\overline{ECAS}$ Negated to $\overline{CAS}$ Negated)	−7	7	−7	7	−7	7	−7	7
55	tD5	(CLK to Refresh $\overline{RAS}$ Asserted) Minus (CLK to Refresh $\overline{RAS}$ Negated)		6		6		6		6
56	tD6	($\overline{AREQ}$ Negated to $\overline{RAS}$ Negated) Minus ($\overline{ADS}$ Asserted to $\overline{RAS}$ Asserted ((Mode 1))		12		12		10		10

14.0 AC Timing Parameters: DP8420A/DP8421A/DP8422A (Continued)

Unless otherwise stated V_{CC} = 5.0V ±10%, 0°C < T_A < 70°C, the output load capacitance is typical for 4 banks of 18 DRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:

C_L = 50 pF loads on all outputs except

C_L = 150 pF loads on Q0–8, 9, 10 and $\overline{WE}$; or

C_H = 50 pF loads on all outputs except

C_H = 125 pF loads on $\overline{RAS}$0–3 and $\overline{CAS}$0–3 and

C_H = 380 pF loads on Q0–8, 9, 10 and $\overline{WE}$.

Number	Symbol	Common Dual Access Parameter Description	8420A/21A/22A-20				8420A/21A/22A-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
100	tHCKARQB	$\overline{AREQB}$ Negated Held from CLK High	3		3		3		3	
101	tSARQBCK	$\overline{AREQB}$ Asserted Setup to CLK High	8		8		7		7	
102	tPAQBRASL	$\overline{AREQB}$ Asserted to $\overline{RAS}$ Asserted		43		48		37		41
103	tPAQBRASH	$\overline{AREQB}$ Negated to $\overline{RAS}$ Negated		41		46		32		36
105	tPCKRASG	CLK High to $\overline{RAS}$ Asserted for Pending Port B Access		55		60		44		48
106	tPAQBATKBL	$\overline{AREQB}$ Asserted to $\overline{ATACKB}$ Asserted		57		57		45		45
107	tPCKATKB	CLK High to $\overline{ATACKB}$ Asserted for Pending Access		67		67		51		51
108	tPCKGH	CLK High to GRANTB Asserted		40		40		32		32
109	tPCKGL	CLK High to GRANTB Negated		35		35		29		29
110	tSADDCKG	Row Address Setup to CLK High That Asserts $\overline{RAS}$ following a GRANTB Change to Ensure tASR = 0 ns for Port B	11		15		11		16	
111	tSLOCKCK	$\overline{LOCK}$ Asserted Setup to CLK Low to Lock Current Port	5		5		5		5	
112	tPAQATKBH	$\overline{AREQ}$ Negated to $\overline{ATACKB}$ Negated		26		26		21		21
113	tPAQBCASH	$\overline{AREQB}$ Negated to $\overline{CAS}$ Negated		59		67		47		54
114	tSADAQB	Address Valid Setup to $\overline{AREQB}$ Asserted	7		11		7		12	
116	tHCKARQG	$\overline{AREQ}$ Negated Held from CLK High	5		5		5		5	
117	tWAQB	$\overline{AREQB}$ High Pulse Width to Guarantee tASR = 0 ns	31		35		26		31	
118a	tPAQBCAS0	$\overline{AREQB}$ Asserted to $\overline{CAS}$ Asserted (tRAH = 15 ns, tASC = 0 ns)		103		111		87		94
118b	tPAQBCAS1	$\overline{AREQB}$ Asserted to $\overline{CAS}$ Asserted (tRAH = 15 ns, tASC = 10 ns)		113		121		97		104
118c	tPAQBCAS2	$\overline{AREQB}$ Asserted to $\overline{CAS}$ Asserted (tRAH = 25 ns, tASC = 0 ns)		113		121		97		104
118d	tPAQBCAS3	$\overline{AREQB}$ Asserted to $\overline{CAS}$ Asserted (tRAH = 25 ns, tASC = 10 ns)		123		131		107		114
120a	tPCKCASG0	CLK High to $\overline{CAS}$ Asserted for Pending Port B Access (tRAH = 15 ns, tASC = 0 ns)		113		121		96		103
120b	tPCKCASG1	CLK High to $\overline{CAS}$ Asserted for Pending Port B Access (tRAH = 15 ns, tASC = 10 ns)		123		131		106		113
120c	tPCKCASG2	CLK High to $\overline{CAS}$ Asserted for Pending Port B Access (tRAH = 25 ns, tASC = 0 ns)		123		131		106		113
120d	tPCKCASG3	CLK High to $\overline{CAS}$ Asserted for Pending Port B Access (tRAH = 25 ns, tASC = 10 ns)		133		141		116		123
121	tSBADDCKG	Bank Address Valid Setup to CLK High That Starts $\overline{RAS}$ for Pending Port B Access	10		10		10		10	

14.0 AC Timing Parameters: DP8420A/DP8421A/DP8422A (Continued)

Unless otherwise stated V_{CC} = 5.0V ±10%, 0°C < T_A < 70°C, the output load capacitance is typical for 4 banks of 18 DRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:
C_L = 50 pF loads on all outputs except
C_L = 150 pF loads on Q0–8, 9, 10 and $\overline{WE}$; or
C_H = 50 pF loads on all outputs except
C_H = 125 pF loads on $\overline{RAS}$0–3 and $\overline{CAS}$0–3 and
C_H = 380 pF loads on Q0–8, 9, 10 and $\overline{WE}$.

Number	Symbol	Refresh Parameter Description	8420A/21A/22A-20				8420A/21A/22A-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
200	tSRFCK	$\overline{RFSH}$ Asserted Setup to CLK High	27		27		22		22	
201	tSDRFCK	$\overline{DISRFSH}$ Asserted Setup to CLK High	28		28		22		22	
202	tSXRFCK	EXTENDRF Setup to CLK High	15		15		12		12	
204	tPCKRFL	CLK High to $\overline{RFIP}$ Asserted		39		39		31		31
205	tPARQRF	$\overline{AREQ}$ Negated to $\overline{RFIP}$ Asserted		62		62		50		50
206	tPCKRFH	CLK High to $\overline{RFIP}$ Negated		65		65		51		51
207	tPCKRFRASH	CLK High to Refresh $\overline{RAS}$ Negated		35		40		29		33
208	tPCKRFRASL	CLK High to Refresh $\overline{RAS}$ Asserted		28		33		23		27
209a	tPCKCL0	CLK High to $\overline{CAS}$ Asserted during Error Scrubbing (tRAH = 15 ns, tASC = 0 ns)		82		90		73		80
209b	tPCKCL1	CLK High to $\overline{CAS}$ Asserted during Error Scrubbing (tRAH = 15 ns, tASC = 10 ns)		92		100		83		90
209c	tPCKCL2	CLK High to $\overline{CAS}$ Asserted during Error Scrubbing (tRAH = 25 ns, tASC = 0 ns)		92		100		83		90
209d	tPCKCL3	CLK High to $\overline{CAS}$ Asserted during Error Scrubbing (tRAH = 25 ns, tASC = 10 ns)		102		110		83		100
210	tWRFSH	$\overline{RFSH}$ Pulse Width	15		15		15		15	
211	tPCKRQL	CLK High to $\overline{RFRQ}$ Asserted		46		46		40		40
212	tPCKRQH	CLK High to $\overline{RFRQ}$ Negated		50		50		40		40

14.0 AC Timing Parameters: DP8420A/DP8421A/DP8422A (Continued)

Unless otherwise stated V_{CC} = 5.0V ±10%, 0°C < T_A < 70°C, the output load capacitance is typical for 4 banks of 18 DRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:
C_L = 50 pF loads on all outputs except
C_L = 150 pF loads on Q0–8, 9, 10 and $\overline{WE}$; or
C_H = 50 pF loads on all outputs except
C_H = 125 pF loads on $\overline{RAS}$0–3 and $\overline{CAS}$0–3 and
C_H = 380 pF loads on Q0–8, 9, 10 and $\overline{WE}$.

Number	Symbol	Mode 0 Access Parameter Description	8420A/21A/22A-20				8420A/21A/22A-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
300	tSCSCK	$\overline{CS}$ Asserted to CLK High	14		14		13		13	
301a	tSALECKNL	ALE Asserted Setup to CLK High Not Using On-Chip Latches or if Using On-Chip Latches and B0, B1, Are Constant, Only 1 Bank	16		16		15		15	
301b	tSALECKL	ALE Asserted Setup to CLK High, if Using On-Chip Latches if B0, B1 Can Change, More Than One Bank	29		29		29		29	
302	tWALE	ALE Pulse Width	18		18		13		13	
303	tSBADDCK	Bank Address Valid Setup to CLK High	20		20		18		18	
304	tSADDCK	Row, Column Valid Setup to CLK High to Guarantee tASR = 0 ns	11		15		11		16	
305	tHASRCB	Row, Column, Bank Address Held from ALE Negated (Using On-Chip Latches)	10		10		8		8	
306	tSRCBAS	Row, Column, Bank Address Setup to ALE Negated (Using On-Chip Latches)	3		3		2		2	
307	tPCKRL	CLK High to $\overline{RAS}$ Asserted		27		32		22		26
308a	tPCKCL0	CLK High to $\overline{CAS}$ Asserted (tRAH = 15 ns, tASC = 0 ns)		81		89		72		79
308b	tPCKCL1	CLK High to $\overline{CAS}$ Asserted (tRAH = 15 ns, tASC = 10 ns)		91		99		82		89
308c	tPCKCL2	CLK High to $\overline{CAS}$ Asserted (tRAH = 25 ns, tASC = 0 ns)		91		99		82		89
308d	tPCKCL3	CLK High to $\overline{CAS}$ Asserted (tRAH = 25 ns, tASC = 10 ns)		101		109		92		99
309	tHCKALE	ALE Negated Hold from CLK High	0		0		0		0	
310	tSWINCK	$\overline{WIN}$ Asserted Setup to CLK High to Guarantee $\overline{CAS}$ is Delayed	−21		−21		−16		−16	
311	tPCSWL	$\overline{CS}$ Asserted to $\overline{WAIT}$ Asserted		26		26		22		22
312	tPCSWH	$\overline{CS}$ Negated to $\overline{WAIT}$ Negated		30		30		25		25
313	tPCLKDL1	CLK High to $\overline{DTACK}$ Asserted (Programmed as $\overline{DTACK0}$)		40		40		32		32
314	tPALEWL	ALE Asserted to $\overline{WAIT}$ Asserted ($\overline{CS}$ is Already Asserted)		35		35		29		29
315		$\overline{AREQ}$ Negated to CLK High That Starts Access $\overline{RAS}$ to Guarantee tASR = 0 ns (Non-Interleaved Mode Only)	41		45		34		39	
316	t_{PCKCV0}	CLK High to Column Address Valid (t_{RAH} = 15 ns, t_{ASC} = 0 ns)		78		87		66		75

14.0 AC Timing Parameters: DP8420A/DP8421A/DP8422A (Continued)

Unless otherwise stated V_{CC} = 5.0V ±10%, 0°C < T_A < 70°C, the output load capacitance is typical for 4 banks of 18 DRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:
C_L = 50 pF loads on all outputs except
C_L = 150 pF loads on Q0–8, 9, 10 and $\overline{WE}$; or
C_H = 50 pF loads on all outputs except
C_H = 125 pF loads on $\overline{RAS}$0–3 and $\overline{CAS}$0–3 and
C_H = 380 pF loads on Q0–8, 9, 10 and $\overline{WE}$.

Number	Symbol	Mode 1 Access Parameter Description	8420A/21A/22A-20				8420A/21A/22A-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
400a	tSADSCK1	$\overline{ADS}$ Asserted Setup to CLK High	15		15		13		13	
400b	tSADSCKW	$\overline{ADS}$ Asserted Setup to CLK (to Guarantee Correct $\overline{WAIT}$ or $\overline{DTACK}$ Output; Doesn't Apply for $\overline{DTACK0}$)	31		31		25		25	
401	tSCSADS	$\overline{CS}$ Setup to $\overline{ADS}$ Asserted	6		6		5		5	
402	tPADSRL	$\overline{ADS}$ Asserted to $\overline{RAS}$ Asserted		30		35		25		29
403a	tPADSCL0	$\overline{ADS}$ Asserted to $\overline{CAS}$ Asserted (tRAH = 15 ns, tASC = 0 ns)		86		94		75		82
403b	tPADSCL1	$\overline{ADS}$ Asserted to $\overline{CAS}$ Asserted (tRAH = 15 ns, tASC = 10 ns)		96		104		85		92
403c	tPADSCL2	$\overline{ADS}$ Asserted to $\overline{CAS}$ Asserted (tRAH = 25 ns, tASC = 0 ns)		96		104		85		92
403d	tPADSCL3	$\overline{ADS}$ Asserted to $\overline{CAS}$ Asserted (tRAH = 25 ns, tASC = 10 ns)		106		114		95		102
404	tSADDADS	Row Address Valid Setup to $\overline{ADS}$ Asserted to Guarantee tASR = 0 ns	9		13		9		14	
405	tHCKADS	$\overline{ADS}$ Negated Held from CLK High	0		0		0		0	
406	tSWADS	$\overline{WAITIN}$ Asserted Setup to $\overline{ADS}$ Asserted to Guarantee $\overline{DTACK0}$ Is Delayed	0		0		0		0	
407	tSBADAS	Bank Address Setup to $\overline{ADS}$ Asserted	11		11		11		11	
408	tHASRCB	Row, Column, Bank Address Held from $\overline{ADS}$ Asserted (Using On-Chip Latches)	10		10		10		10	
409	tSRCBAS	Row, Column, Bank Address Setup to $\overline{ADS}$ Asserted (Using On-Chip Latches)	3		3		2		2	
410	tWADSH	$\overline{ADS}$ Negated Pulse Width	12		16		12		17	
411	tPADSD	$\overline{ADS}$ Asserted to $\overline{DTACK}$ Asserted (Programmed as $\overline{DTACK0}$)		43		43		35		35
412	tSWINADS	$\overline{WIN}$ Asserted Setup to $\overline{ADS}$ Asserted (to Guarantee $\overline{CAS}$ Delayed during Writes Accesses)	−10		−10		−10		−10	
413	tPADSWL0	$\overline{ADS}$ Asserted to $\overline{WAIT}$ Asserted (Programmed as $\overline{WAIT0}$, Delayed Access)		35		35		29		29
414	tPADSWL1	$\overline{ADS}$ Asserted to $\overline{WAIT}$ Asserted (Programmed $\overline{WAIT}$ 1/2 or 1)		35		35		29		29
415	tPCLKDL1	CLK High to $\overline{DTACK}$ Asserted (Programmed as $\overline{DTACK0}$, Delayed Access)		40		40		32		32
416		$\overline{AREQ}$ Negated to $\overline{ADS}$ Asserted to Guarantee tASR = 0 ns (Non Interleaved Mode Only)	38		42		31		36	
417	tPADSCV0	$\overline{ADS}$ Asserted to Column Address Valid (t_{RAH} = 15 ns, t_{ASC} = 0 ns)		83		92		69		78

14.0 AC Timing Parameters: DP8420A/DP8421A/DP8422A (Continued)

Unless otherwise stated V_{CC} = 5.0V ±10%, 0°C < T_A < 70°C, the output load capacitance is typical for 4 banks of 18 DRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:
C_L = 50 pF loads on all outputs except
C_L = 150 pF loads on Q0–8, 9, 10 and $\overline{WE}$; or
C_H = 50 pF loads on all outputs except
C_H = 125 pF loads on $\overline{RAS}$0–3 and $\overline{CAS}$0–3 and
C_H = 380 pF loads on Q0–8, 9, 10 and $\overline{WE}$.

Number	Symbol	Mode 1 Dual Access Parameter Description	8420A/21A/22A-20				8420A/21A/22A-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
450	tSADDCKG	Row Address Setup to CLK High That Asserts $\overline{RAS}$ following a GRANTB Port Change to Ensure tASR = 0 ns	11		15		11		16	
451	tPCKRASG	CLK High to $\overline{RAS}$ Asserted for Pending Access		48		53		38		42
452	tPCLKDL2	CLK to $\overline{DTACK}$ Asserted for Delayed Accesses (Programmed as $\overline{DTACK}$0)		53		53		43		43
453a	tPCKCASG0	CLK High to $\overline{CAS}$ Asserted for Pending Access (tRAH = 15 ns, tASC = 0 ns)		101		109		86		93
453b	tPCKCASG1	CLK High to $\overline{CAS}$ Asserted for Pending Access (tRAH = 15 ns, tASC = 10 ns)		111		119		96		103
453c	tPCKCASG2	CLK High to $\overline{CAS}$ Asserted for Pending Access (tRAH = 25 ns, tASC = 0 ns)		111		119		96		103
453d	tPCKCASG3	CLK High to $\overline{CAS}$ Asserted for Pending Access (tRAH = 25 ns, tASC = 10 ns)		121		129		106		113
454	tSBADDCKG	Bank Address Valid Setp to CLK High That Asserts $\overline{RAS}$ for Pending Access	5		5		4		4	
455	tSADSCK0	$\overline{ADS}$ Asserted Setup to CLK High	12		12		11		11	

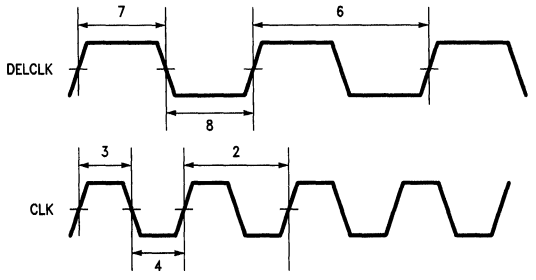
Number	Symbol	Programming Parameter Description	8420A/21A/22A-20				8420A/21A/22A-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
500	tHMLADD	Mode Address Held from $\overline{ML}$ Negated	8		8		7		7	
501	tSADDML	Mode Address Setup to $\overline{ML}$ Negated	6		6		6		6	
502	tWML	$\overline{ML}$ Pulse Width	15		15		15		15	
503	tSADAQML	Mode Address Setup to $\overline{AREQ}$ Asserted	0		0		0		0	
504	tHADAQML	Mode Address Held from $\overline{AREQ}$ Asserted	51		51		38		38	
505	tSCSARQ	$\overline{CS}$ Asserted Setup to $\overline{AREQ}$ Asserted	6		6		6		6	
506	tSMLARQ	$\overline{ML}$ Asserted Setup to $\overline{AREQ}$ Asserted	10		10		10		10	

Note 1: "Absolute Maximum Ratings" are the values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: Input pulse 0V to 3V; tR = tF = 2.5 ns. Input reference point on AC measurements is 1.5V. Output reference points are 2.4V for High and 0.8V for Low.

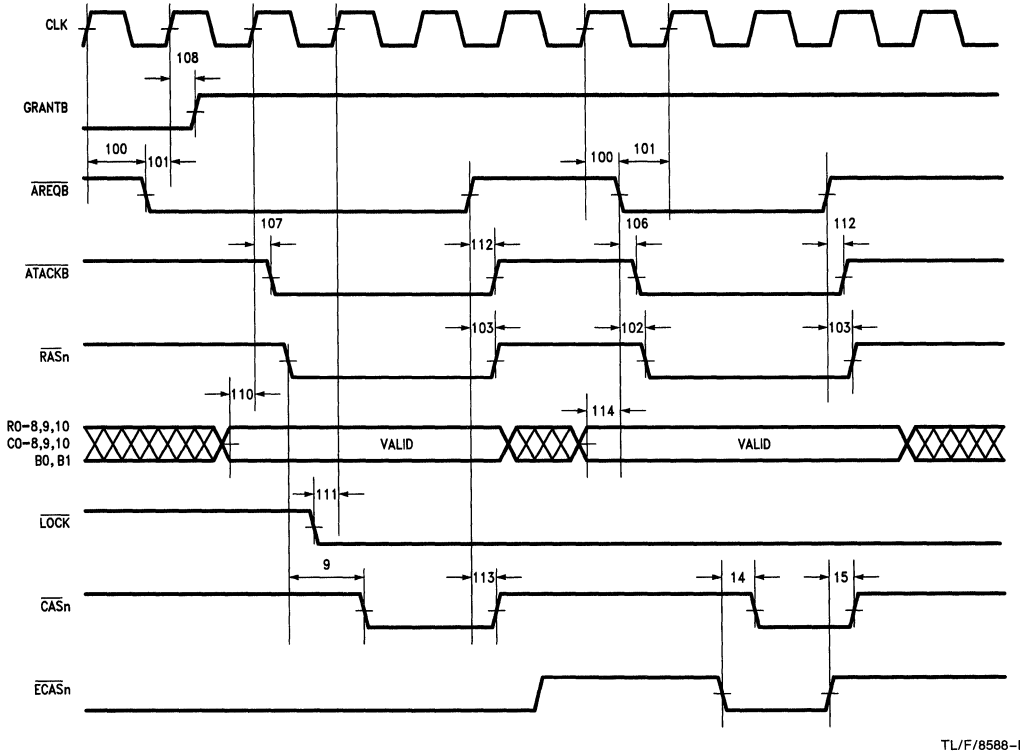
Note 3: AC Production testing is done at 50 pF.

14.0 AC Timing Parameters: DP8420A/DP8421A/DP8422A (Continued)



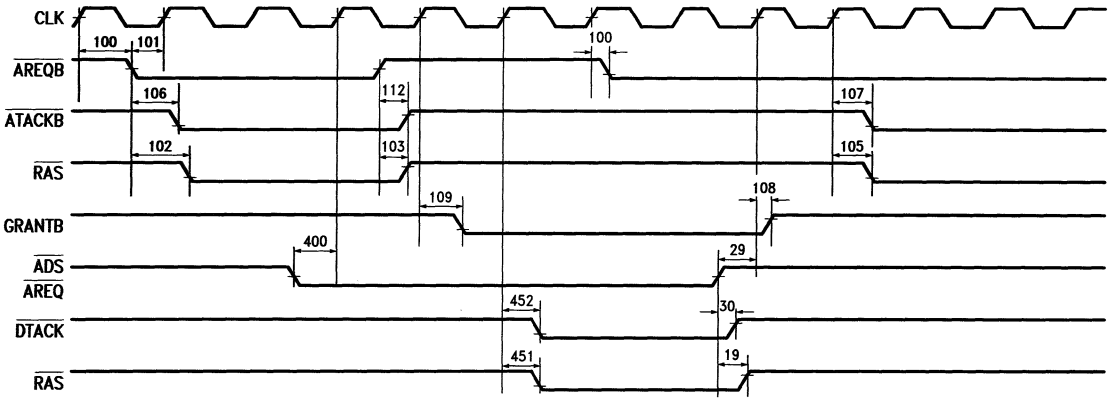
TL/F/8588–E9

FIGURE 52. Clock, DELCLK Timing



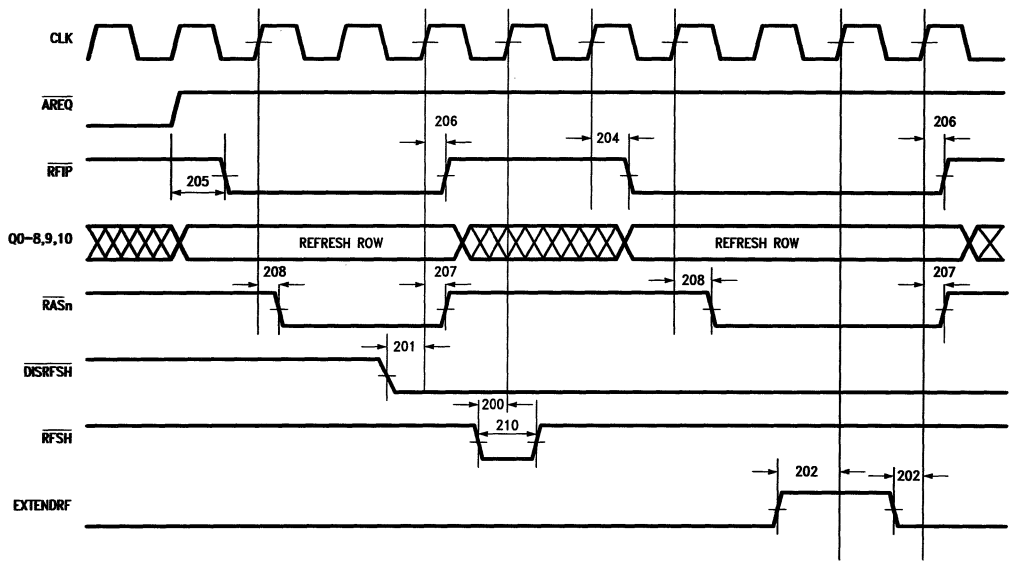
TL/F/8588–F0

FIGURE 53. 100: Dual Access Port B



TL/F/8588–F1

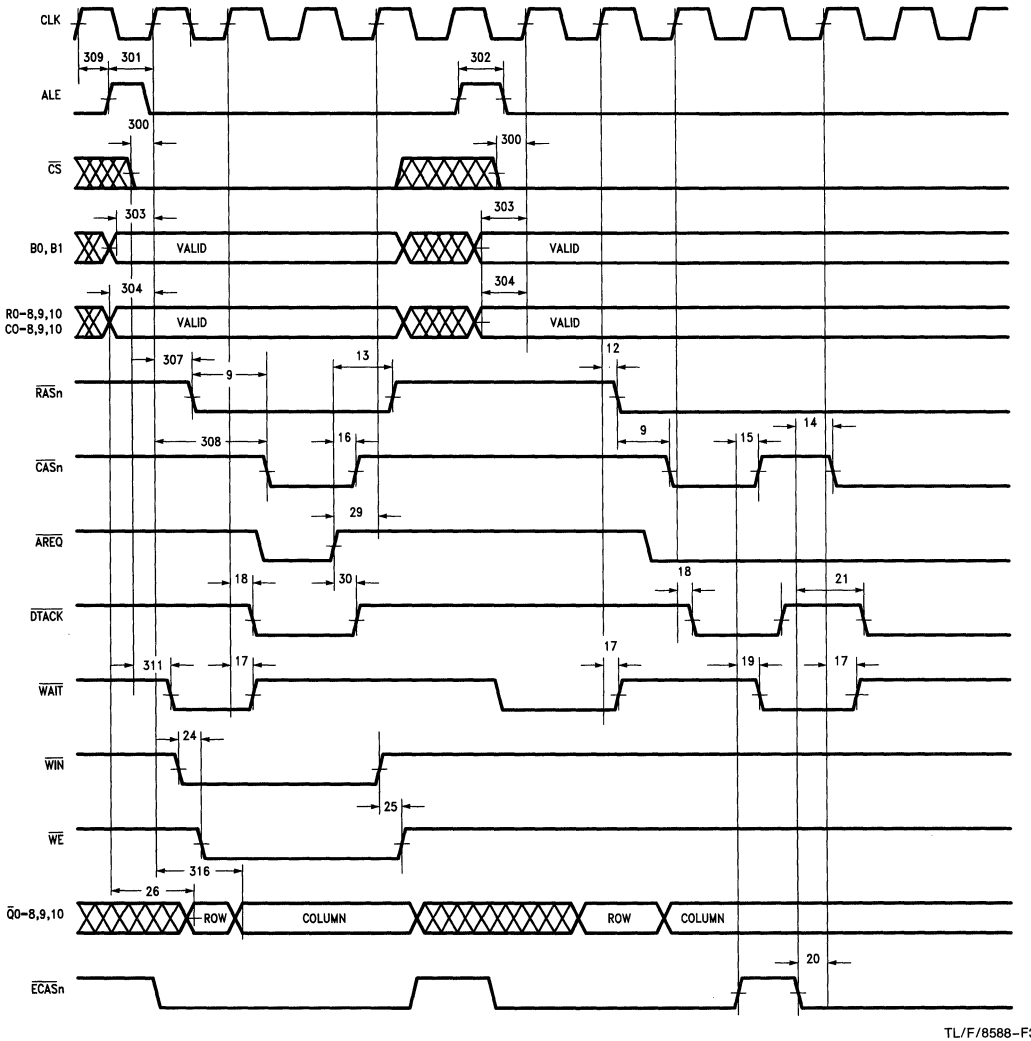
FIGURE 54. 100: Port A and Port B Dual Access



TL/F/8588–F2

FIGURE 55. 200: Refresh Timing

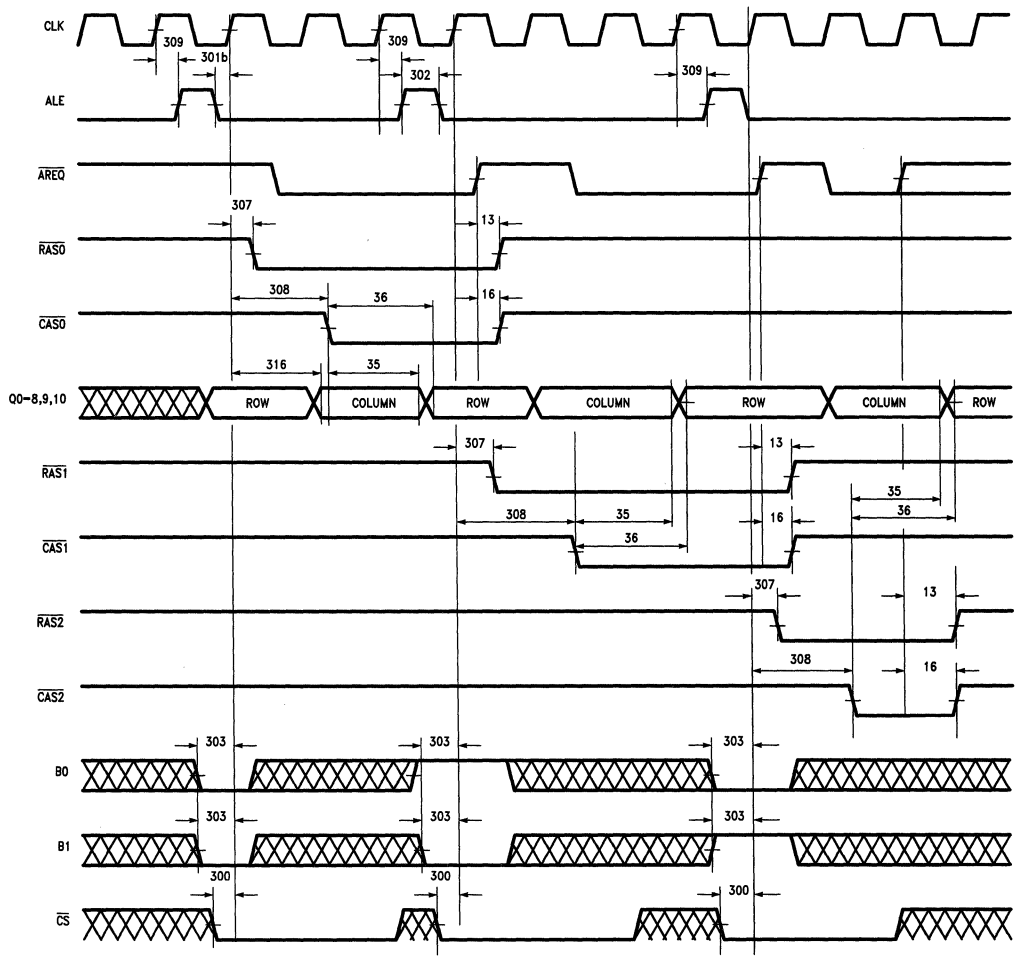
14.0 AC Timing Parameters: DP8420A/DP8421A/DP8422A (Continued)



TL/F/8588-F3

FIGURE 56. 300: Mode 0 Timing

14.0 AC Timing Parameters: DP8420A/DP8421A/DP8422A (Continued)

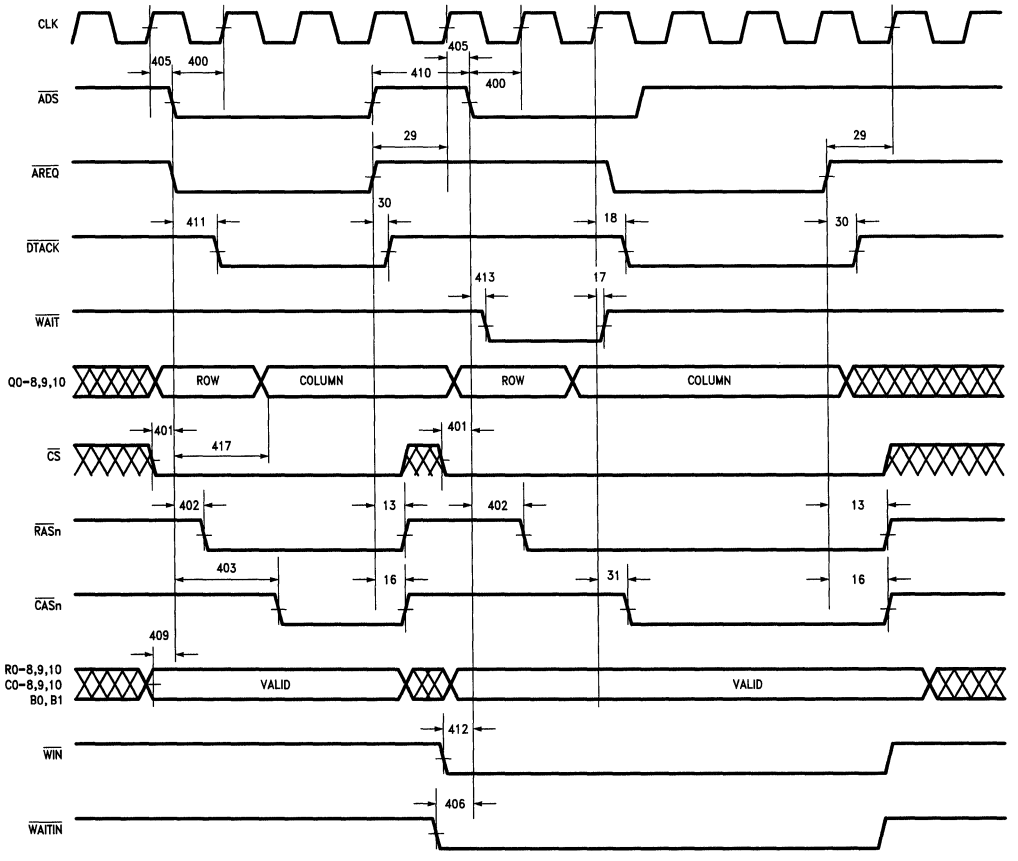


TL/F/8588–F4

(Programmed as C4 = 1, C5 = 1, C6 = 1)

FIGURE 57. 300: Mode 0 Interleaving

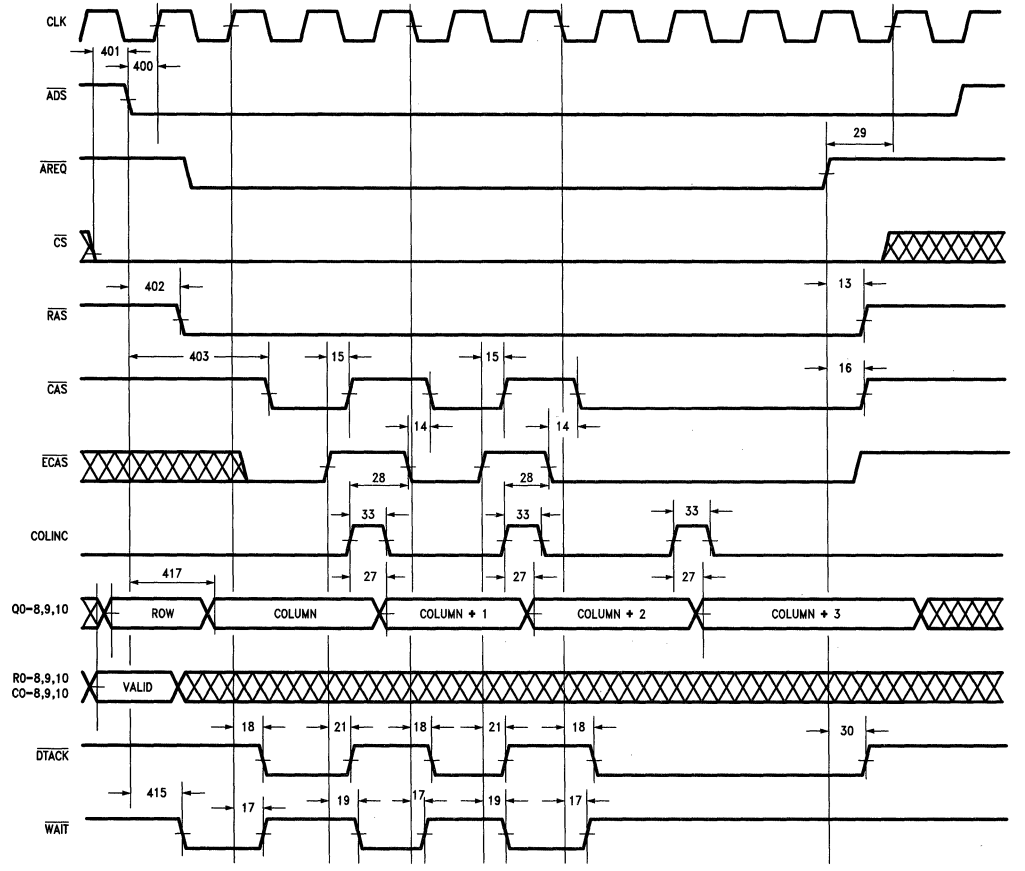
14.0 AC Timing Parameters: DP8420A/DP8421A/DP8422A (Continued)



TL/F/8588–F5

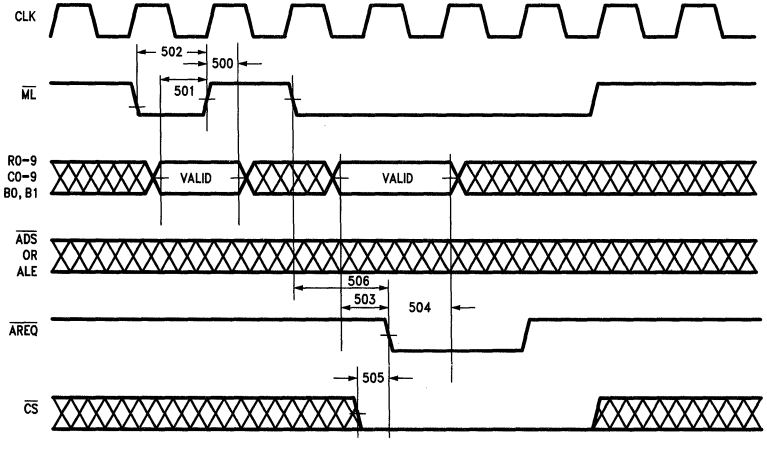
FIGURE 58. 400: Mode 1 Timing

14.0 AC Timing Parameters: DP8420A/DP8421A/DP8422A (Continued)



TL/F/8588-F6

FIGURE 59. 400: COLINC Page/Static Column Access Timing



TL/F/8588-F7

FIGURE 60. 500: Programming

15.0 Functional Differences between the DP8420A/21A/22A and the DP8420/21/22

1. **Extending the Column Address Strobe ($\overline{CAS}$) after $\overline{AREQ}$ Transitions High**

   The DP8420A/21A/22A allows $\overline{CAS}$ to be asserted for an indefinite period of time beyond $\overline{AREQ}$ (or $\overline{AREQB}$, DP8422A only. Scrubbing refreshes are not affected.) being negated by continuing to assert the appropriate $\overline{ECAS}$ inputs. This feature is allowed as long as the $\overline{ECAS}$0 input was negated during programming. The DP8420/21/22 does not allow this feature.

2. **Dual Accessing**

   The DP8420A/21A/22A asserts $\overline{RAS}$ either one or two clock periods after GRANTB has been asserted or negated depending upon how the R0 bit was programmed during the mode load operation. The DP8420/21/22 will always start $\overline{RAS}$ one clock period after GRANTB is asserted or negated. The above statements assume that $\overline{RAS}$ precharge has been completed by the time GRANTB is asserted or negated.

3. **Refresh Request Output ($\overline{RFRQ}$)**

   The DP8420A/21A/22A allows $\overline{RFRQ}$ (refresh request) to be output on the $\overline{WE}$ output pin given that $\overline{ECAS}$0 was negated during programming or the controller was programmed to function in the address pipelining (memory interleaving) mode. The DP8420/21/22 only allows $\overline{RFRQ}$ to be output during the address pipelining mode.

4. **Clearing the Refresh Request Clock Counter**

   The DP8420A/21A/22A allows the internal refresh request clock counter to be cleared by negating $\overline{DISRFSH}$ and asserting $\overline{RFSH}$ for at least 500 ns. The DP8420/21/22 clears the internal refresh request clock counter if $\overline{DISRFSH}$ remains low for at least 500 ns. Once the internal refresh request clock counter is cleared the user is guaranteed that an internally generated $\overline{RFRQ}$ will not be generated for at least 13 μs–15 μs (depending upon how programming bits C0, 1, 2, 3 were programmed).

16.0 DP8420A/21A/22A User Hints

1. All inputs to the DP8420A/21A/22A should be tied high, low or the output of some other device.

   Note: One signal is active high. COLINC (EXTNDRF) should be tied low to disable.

2. Each ground on the DP8420A/21A/22A must be decoupled to the closest on-chip supply (V_{CC}) with 0.1 μF ceramic capacitor. This is necessary because these grounds are kept separate inside the DP8420A/21A/22A. The decoupling capacitors should be placed as close as possible with short leads to the ground and supply pins of the DP8420A/21A/22A.

3. The output called "CAP" should have a 0.1 μF capacitor to ground.

4. The DP8420A/21A/22A has 20Ω series damping resistors built into the output drivers of $\overline{RAS}$, $\overline{CAS}$, address and $\overline{WE}$/$\overline{RFRQ}$. Space should be provided for external damping resistors on the printed circuit board (or wire-wrap board) because they may be needed. The value of these damping resistors (if needed) will vary depending upon the output, the capacitance of the load, and the characteristics of the trace as well as the routing of the trace. The value of the damping resistor also may vary between the wire-wrap board and the printed circuit board. To determine the value of the series damping resistor it is recommended to use an oscilloscope and look at the furthest DRAM from the DP8420A/21A/22A. The undershoot of $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$ and the addresses should be kept to less than 0.5V below ground by varying the value of the damping resistor. The damping resistors should be placed as close as possible with short leads to the driver outputs of the DP8420A/21A/22A.

5. The circuit board must have a good V_{CC} and ground plane connection. If the board is wire-wrapped, the V_{CC} and ground pins of the DP8420A/21A/22A, the DRAM associated logic and buffer circuitry must be soldered to the V_{CC} and ground planes.

6. The traces from the DP8420A/21A/22A to the DRAM should be as short as possible.

7. $\overline{ECAS}$0 should be held low during programming if the user wishes that the DP8420A/21A/22A be compatible with a DP8420/21/22 design.

8. Parameter Changes due to Loading

   All A.C. parameters are specified with the equivalent load capacitances, including traces, of 64 DRAMs organized as 4 banks of 18 DRAMs each. Maximums are based on worst-case conditions. If an output load changes then the A.C. timing parameters associated with that particular output must be changed. For example, if we changed our output load to

   C = 250 pF loads on $\overline{RAS}$0–3 and $\overline{CAS}$0–3

   C = 760 pF loads on Q0–9 and $\overline{WE}$

   we would have to modify some parameters (not all calculated here)

   $308a clock to $\overline{CAS}$ asserted

   (t_{RAH} = 15 ns, t_{ASC} = 0 ns)

   A ratio can be used to figure out the timing change per change in capacitance for a particular parameter by using the specifications and capacitances from heavy and light load timing.

$$\text{Ratio} = \frac{\$308a\ \text{w/Heavy Load} - \$308a\ \text{w/Light Load}}{C_H(\overline{CAS}) - C_L(\overline{CAS})}$$

$$= \frac{79\ \text{ns} - 72\ \text{ns}}{125\ \text{pF} - 50\ \text{pF}} = \frac{7\ \text{ns}}{75\ \text{pF}}$$

$$\$308a\ (\text{actual}) = (\text{capacitance difference} \times \text{ratio}) + \$308a\ (\text{specified})$$

$$= (250\ \text{pF} - 125\ \text{pF})\frac{7\ \text{ns}}{75\ \text{pF}} + 79\ \text{ns}$$

$$= 11.7\ \text{ns} + 79\ \text{ns}$$

$$= 90.7\ \text{ns @ 250 pF load}$$

9. It is recommended that the user perform a hardware reset of the DP8420A/21A/22A before programming and using the chip. A hardware reset consists of asserting both $\overline{ML}$ and $\overline{DISRFSH}$ for a minimum of 16 positive edges of CLK, see Section 8.1.

National Semiconductor

PRELIMINARY

NS32CG821 microCMOS Programmable 1M Dynamic RAM Controller/Driver

General Description

The NS32CG821 dynamic RAM controller provides a low cost, single chip interface between dynamic RAM and the NS32CG16. The NS32CG821 generates all the required access control signal timing for DRAMs. An on-chip refresh request clock is used to automatically refresh the DRAM array. Refreshes and accesses are arbitrated on chip. If necessary, a $\overline{WAIT}$ output inserts wait states into memory access cycles, including burst mode accesses. $\overline{RAS}$ low time during refreshes and $\overline{RAS}$ precharge time after refreshes and back to back accesses are guaranteed through the insertion of wait states. Separate on-chip precharge counters for each $\overline{RAS}$ output can be used for memory interleaving to avoid delayed back to back accesses because of precharge.

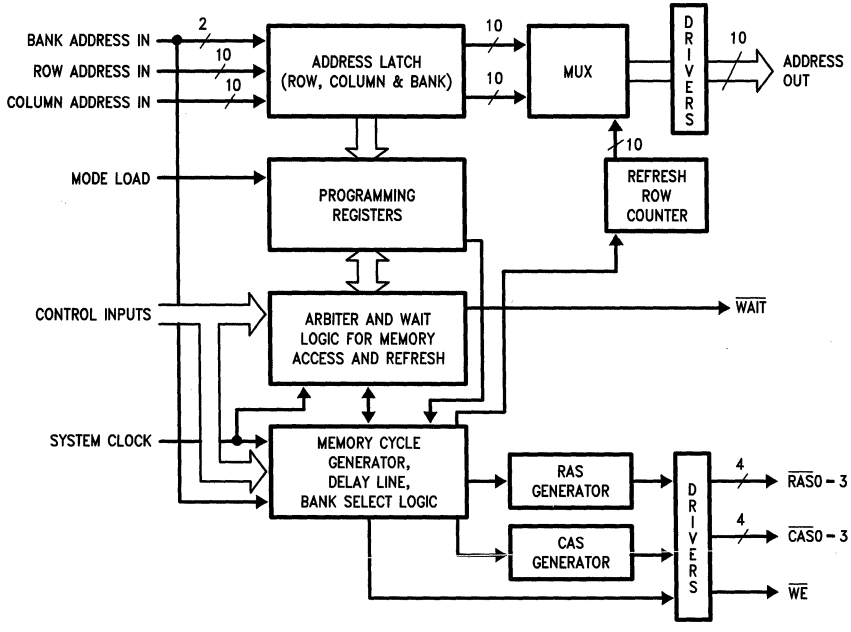
Features

- Allows zero wait state operation
- On chip high precision delay line to guarantee critical DRAM access timing parameters
- microCMOS process for low power
- High capacitance drivers for $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$ and DRAM address on chip
- On chip support for page and static column DRAMs
- Byte enable signals on chip allow byte writing with no external logic
- Selection of controller speeds: 20 MHz and 25 MHz
- On board access refresh arbitration logic
- Direct interface to the NS32CG16 microprocessor
- 4 $\overline{RAS}$ and 4 $\overline{CAS}$ drivers (the $\overline{RAS}$ and $\overline{CAS}$ configuration is programmable)

Control	# of Pins (PLCC)	# of Address Outputs	Largest DRAM Possible	Direct Drive Memory Capacity
NS32CG821	68	10	1 Mbit	8 Mbytes

Block Diagram

NS32CG821 DRAM Controller



TL/F/10126–1

FIGURE 1

Table of Contents

1.0 Introduction

The NS32CG821 is a CMOS Dynamic RAM controller that incorporates many advanced features including the capabilities of address latches, refresh counter, refresh clock, row, column and refresh address multiplexer, delay line, refresh/access arbitration logic and high capacitive drivers. The programmable system interface allows the NS32CG16 microprocessor to directly interface via the NS32CG821 to DRAM arrays up to 8 Mbytes in size. (See *Figure 3* for an example.)

After power up, the NS32CG821 must first be programmed before accessing the DRAM. The chip is programmed through the address bus.

There are two methods of programming the chip. The first method, mode load only, is accomplished by asserting the signal mode load, $\overline{ML}$. A valid programming selection is presented on the row, column, bank and $\overline{ECAS0}$ inputs, then $\overline{ML}$ is negated. When $\overline{ML}$ is negated, the chip is programmed with the valid programming bits on the address bus.

The second method, chip selected access, is accomplished by asserting $\overline{ML}$ and performing a chip selected access. When $\overline{CS}$ and $\overline{TSO}$ are asserted for the access, the chip is programmed. During this programming access, the programming bits affecting the wait logic become effective immediately, allowing the access to terminate. After the access, $\overline{ML}$ is negated and the rest of the programming bits take effect.

Once the NS32CG821 has been programmed, a 60 ms initialization period is entered. During this time, the NS32CG821 controller performs refreshes to the DRAM array so further DRAM warm up cycles are unnecessary.

To access the DRAM, the signal ALE is asserted along with $\overline{CS}$ to ensure a valid DRAM access. ALE asserting sets an internal latch and only needs to be pulsed and not held throughout the entire access. Once $\overline{CS}$ and ALE are both asserted, $\overline{WAIT}$ is asserted, unless $\overline{WAIT}$ is programmed as OT and a non-delayed access occurs, and follows $\overline{CS}$ until the rising clock edge. This is not a problem since the 32CG16 will not process the $\overline{WAIT}$ signal until the end of state T2. $\overline{WAIT}$ only has to guarantee that it meets the setup time to this edge of CLK on which it is sampled. On the next rising clock edge, $\overline{RAS}$ will be asserted for that access. The NS32CG821 will place the row address on the DRAM address bus, guarantee the programmed value of row address hold time of the DRAM, place the column address on the DRAM address bus, guarantee the programmed value of column address setup time and assert $\overline{CAS}$. $\overline{TSO}$ can be asserted anytime after the clock edge which starts the access $\overline{RAS}$. $\overline{RAS}$ and $\overline{CAS}$ will extend until $\overline{TSO}$ is negated.

The NS32CG821 has greatly expanded refresh capabilities compared to other DRAM controllers.

When using internal automatic refreshing, the NS32CG821 will generate an internal refresh request from the refresh request clock. The NS32CG821 will arbitrate between the refresh requests and accesses. Assuming an access is not currently in progress, a refresh will occur and on the next positive clock edge, refreshing will begin. If an access had been in progress, the refresh will begin after the access has terminated.

The controller has two types of refreshing available: conventional and staggered. Any refresh control mode can be used with internal refreshing. In a conventional refresh, all of the $\overline{RAS}$ outputs will be asserted and negated at once. In a staggered refresh, the $\overline{RAS}$ outputs will be asserted one positive clock edge apart.

The NS32CG821 has wait support available as programmable $\overline{WAIT}$, which connects directly to the NS32CG16 $\overline{CWAIT}$ pin. This signal is used by the on-chip arbitor to insert wait states to guarantee the arbitration between accesses and refreshes or precharge.

$\overline{WAIT}$ is asserted during the start of the access (ALE and $\overline{CS}$) and will negate a number of clock edges from the event that starts the access $\overline{RAS}$. After $\overline{WAIT}$ is negated, it will stay negated until the next access. $\overline{WAIT}$ can also be programmed to toggle with $\overline{ECAS}$ inputs during a burst/page mode access.

$\overline{WAIT}$ can be dynamically delayed further through the $\overline{WAITIN}$ signal to the NS32CG821.

The NS32CG821 has address latches, used to latch the bank, row and column address inputs. Once the address is latched, a column increment feature can be used to increment the column address. The address latches can also be programmed to be fall through.

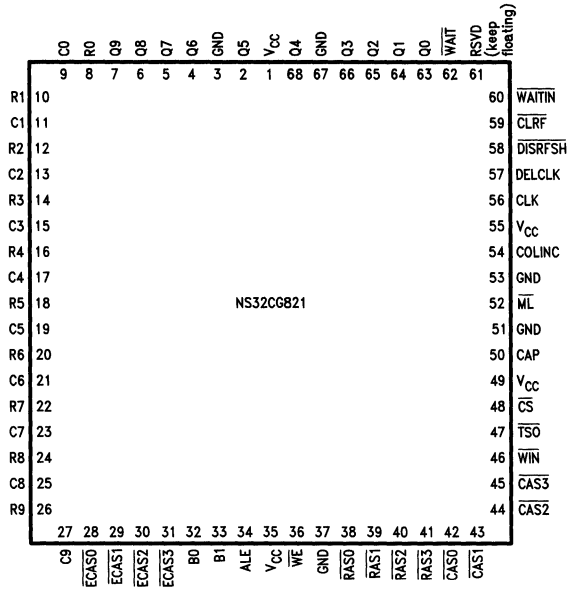
The $\overline{RAS}$ and $\overline{CAS}$ drivers can be configured to drive a one, two, four or eight bank memory array. The $\overline{ECAS}$ signals can then be used to select one of four $\overline{CAS}$ drivers for byte writing with no external logic.

When configuring the NS32CG821 for more than one bank, memory interleaving can be used. By tying the low order address bits to the bank select lines, B0 and B1, sequential back to back accesses will not be delayed since the NS32CG821 has separate precharge counters per bank.

The following explains the terminology used in this data sheet. The terms negated and asserted are used. Asserted refers to a "true" signal. Thus, "$\overline{ECAS0}$ asserted" means the $\overline{ECAS0}$ input is at a logic 0. The term "COLINC asserted" means the COLINC input is at a logic 1. The term negated refers to a "false" signal. Thus, "$\overline{ECAS0}$ negated" means the $\overline{ECAS0}$ input is at a logic 1. The term "COLINC negated" means the input COLINC is at a logic 0. The table shown below clarifies this terminology.

Signal	Action	Logic Level
Active High	Asserted	High
Active High	Negated	Low
Active Low	Asserted	Low
Active Low	Negated	High

Connection Diagram

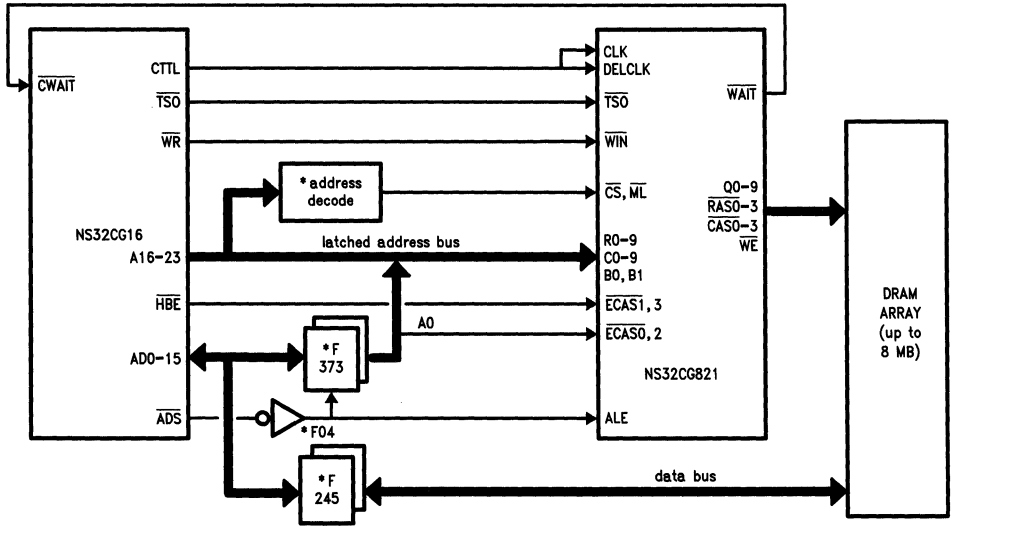


TL/F/10126–2

Top View

FIGURE 2

Order Number NS32CG821V-20 or NS32CG821V-25
See NS Package Number V68A



TL/F/10126–48

*Standard components in any NS32CG16 design

FIGURE 3. NS32CG16–NS32CG821 Connection Diagram (Note 1)

Note 1: This is only one possible way to connect the NS32CG821. See Sections 7 and 8 for additional options and configurations for hookup.

2.0 Signal Descriptions

Pin Name	Input/ Output	Description
2.1 ADDRESS, R/W AND PROGRAMMING SIGNALS		
R0–9	I	**ROW ADDRESS:** These inputs are used to specify the row address during an access to the DRAM. They are also used to program the chip when $\overline{ML}$ is asserted (except R10).
C0–9	I	**COLUMN ADDRESS:** These inputs are used to specify the column address during an access to the DRAM. They are also used to program the chip when $\overline{ML}$ is asserted (except C10).
B0, B1	I	**BANK SELECT:** Depending on programming, these inputs are used to select a group of $\overline{RAS}$ and $\overline{CAS}$ outputs to assert during an access. They are also used to program the chip when $\overline{ML}$ is asserted.
$\overline{ECAS}$0–3	I	**ENABLE $\overline{CAS}$:** These inputs are used to enable a single or group of $\overline{CAS}$ outputs when asserted. In combination with the B0, B1 and the programming bits, these inputs select which $\overline{CAS}$ output or $\overline{CAS}$ outputs will assert during an access. The $\overline{ECAS}$ signals can also be used to toggle a group of $\overline{CAS}$ outputs for page mode accesses. They also can be used for byte write operations.
$\overline{WIN}$	I	**WRITE ENABLE IN:** This input is used to signify a write operation to the DRAM. The $\overline{WE}$ output follows this input. This input asserted will also cause $\overline{CAS}$ to delay to the next positive clock edge if address bit C9 is asserted during programming.
COLINC	I	**COLUMN INCREMENT:** When the address latches are used, this input functions as COLINC. Asserting this signal causes the column address to be incremented by one.
$\overline{ML}$	I	**MODE LOAD:** This input signal, when low, enables the internal programming register that stores the programming information.
2.2 DRAM CONTROL SIGNALS		
Q0–9	O	**DRAM ADDRESS:** These outputs are the multiplexed output of the R0–9 and C0–9 and form the DRAM address bus. These outputs contain the refresh address whenever refreshing is taking place. They contain high capacitive drivers with 20Ω series damping resistors.
$\overline{RAS}$0–3	O	**ROW ADDRESS STROBES:** For an access, these outputs are asserted to latch the row address contained on the outputs Q0–9 into the DRAM. For refreshing, the $\overline{RAS}$ outputs are used to latch the refresh row address contained on the Q0–9 outputs in the DRAM. These outputs contain high capacitive drivers with 20Ω series damping resistors.
$\overline{CAS}$0–3	O	**COLUMN ADDRESS STROBES:** These outputs are asserted to latch the column address contained on the outputs Q0–9 into the DRAM. These outputs have high capacitive drivers with 20Ω series damping resistors.
$\overline{WE}$	O	**WRITE ENABLE:** This output asserted specifies a write operation to the DRAM. When negated, this output specifies a read operation to the DRAM. This output has a high capacitive driver and a 20Ω series damping resistor.
2.3 REFRESH SIGNALS		
$\overline{CLRF}$	I	**CLEAR REFRESH:** This pin, in conjunction with $\overline{DISRFSH}$ is used to clear the internal refresh counter.
$\overline{DISRFSH}$	I	**DISABLE REFRESH:** When asserted with $\overline{ML}$ asserted for 16 positive edges of clock, the entire chip is reset and when negated with $\overline{CLRF}$ asserted clears the internal refresh address counter.
2.4 MEMORY ACCESS		
ALE	I	**ADDRESS LATCH ENABLE:** When ALE asserted along with $\overline{CS}$ causes an internal latch to be set. Once this latch is set and precharge time has been met an access will start from the positive clock edge of CLK as soon as possible. If Address Latch ($B_0 = 0$) is programmed, the low going edge of this signal latches the bank, row, and column address.
$\overline{CS}$	I	**CHIP SELECT:** This input signal must be asserted to enable an access.
$\overline{TSO}$	I	**TIMING STATE OUTPUT:** This input signal must be asserted some time after the first positive clock edge after ALE has been asserted. When this signal is negated, $\overline{RAS}$ is negated for the access.

2.0 Signal Descriptions (Continued)

Pin Name	Input/ Output	Description
2.4 MEMORY ACCESS (Continued)		
$\overline{WAIT}$	O	**WAIT:** This output can be programmed to insert wait states into a CPU access cycle. This signal can be delayed by a number of positive clock edges or negative clock levels of CLK, depending on how it is programmed, to increase the microprocessor's access cycle through the insertion of wait states.
$\overline{WAITIN}$	I	**WAIT INCREASE:** This input can be used to dynamically increase the number of positive clock edges of CLK until $\overline{WAIT}$ will be negated during a DRAM access.
2.5 POWER SIGNALS AND CAPACITOR INPUT		
V_{CC}	I	**POWER:** Supply Voltage.
GND	I	**GROUND:** Supply Voltage Reference.
CAP	I	**CAPACITOR:** This input is used by the internal PLL for stabilization. The value of the ceramic capacitor should be 0.1 μF and should be connected between this input and ground.
2.6 CLOCK INPUTS There are two clock inputs to the NS32CG821, CLK and DELCLK. These two clocks may both be tied to the same clock input, or they may be two separate clocks, running at different frequencies, asynchronous to each other.		
CLK	I	**SYSTEM CLOCK:** This input may be in the range of 0 Hz up to 25 MHz. This input is generally a constant frequency but it may be controlled externally to change frequencies or perhaps be stopped for some arbitrary period of time. This input provides the clock to the internal state machine that arbitrates between accesses and refreshes. This clock's positive edges and negative levels are used to extend the $\overline{WAIT}$ signal. This clock is also used as the reference for the $\overline{RAS}$ precharge time and $\overline{RAS}$ low time during refresh. All memory accesses are assumed to be synchronous to the system clock CLK.
DELCLK	I	**DELAY LINE CLOCK:** The clock input DELCLK, may be in the range of 6 MHz to 20 MHz and should be a multiple of 2 (i.e., 6, 8, 10, 12, 14, 16, 18, 20 MHz) to have the NS32CG821 switching characteristics hold. If DELCLK is not one of the above frequencies the accuracy of the internal delay line will suffer. This is because the phase locked loop that generates the delay line assumes an input clock frequency of a multiple of 2 MHz. For example, if the DELCLK input is at 7 MHz and we choose a divide by 3 (program bits C0–2) this will produce 2.333 MHz which is 16.667% off of 2 MHz. Therefore, the NS32CG821 delay line would produce delays that are shorter (faster delays) than what is intended. If divide by 4 was chosen the delay line would be longer (slower delays) than intended (1.75 MHz instead of 2 MHz). (See Section 10 for more information.) This clock is also divided to create the internal refresh clock.

3.0 Memory Access

An access to DRAM is initiated by two input signals: ALE and $\overline{CS}$. The access is always terminated by one signal: $\overline{TSO}$. These input signals should be synchronous to the input clock, CLK. Once an access has been requested by $\overline{CS}$ and ALE, the NS32CG821 will guarantee the following:

The NS32CG821 will have the row address valid to the DRAMs' address bus, Q0–9 given that the row address set-up time to the NS32CG821 was met;

The NS32CG821 will bring the appropriate $\overline{RAS}$ or $\overline{RAS}$s low;

The NS32CG821 will guarantee the minimum row address hold time, before switching the internal multiplexer to place the column address on the DRAM address bus, Q0–9;

The NS32CG821 will guarantee the minimum column address setup time before asserting the appropriate $\overline{CAS}$ or $\overline{CAS}$s;

The NS32CG821 will hold the column address valid the minimum specified column address hold time.

The memory access shown in *Figure 4* is selected by negating the input B1 during programming. This access mode al-

3.0 Memory Access (Continued)

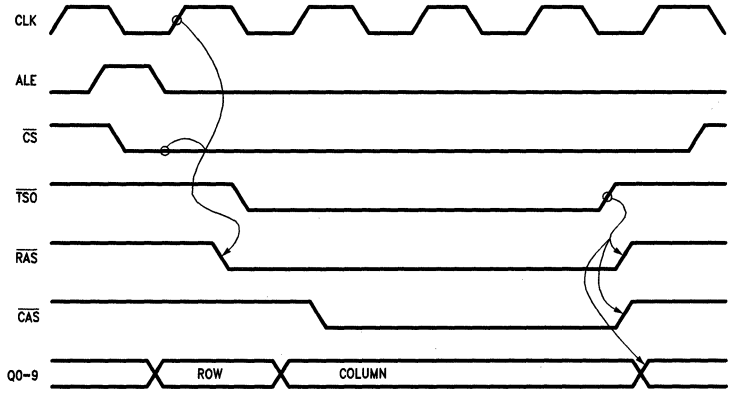


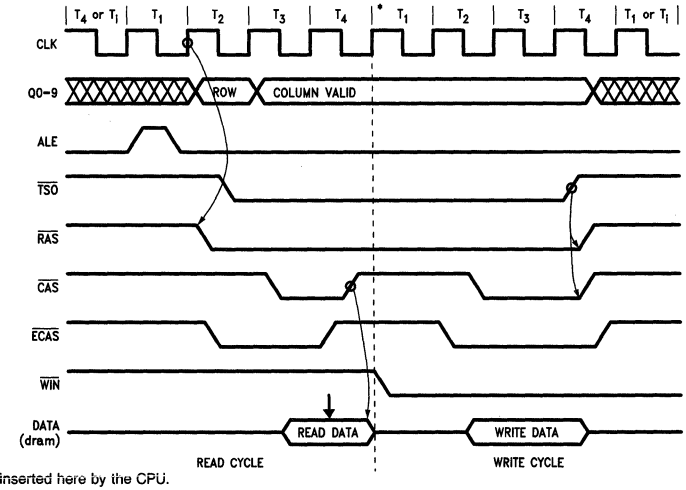
FIGURE 4. Memory Access

lows accesses to DRAM to always be initiated from the positive edge of the system input clock, CLK. To initiate an access, ALE is pulsed high and CS is asserted. Pulsing ALE high and asserting CS, sets an internal latch which requests an access. If the precharge time from the last access or DRAM refresh had been met and a refresh of DRAM was not in progress, the RAS or group of RASs would be initiated from the first positive edge of CLK. If a DRAM refresh is in progress or precharge time is required, the controller will wait until these events have taken place and assert RAS on the next positive edge of CLK.

Once ALE and CS are both asserted, WAIT is asserted, unless WAIT is programmed as OT and a non-delayed access occurs, and follows CS until the rising clock edge. This is not a problem since the 32CG16 will not process the WAIT signal until the end of state T2. WAIT only has to guarantee that it meets the setup time to this edge of CLK on which it is sampled.

Sometime after the first positive edge of CLK after ALE and CS have been asserted, the input TSO must be asserted. Once TSO has been asserted, CS can be negated. Once TSO is negated, RAS and CAS will be negated. ALE can stay asserted several periods of CLK. However, ALE must be negated before or during the period of CLK in which TSO is negated.

There are 2 methods by which this chip can be used to do read-modify-write access cycles. The first method involves doing a late write access where the WIN input is asserted some delay after CAS is asserted. The second method involves doing a page mode read access followed by a page mode write access with RAS held low (see *Figure 5*).



*There may be idle states inserted here by the CPU.

FIGURE 5. Read-Modify-Write Access Cycle

3.0 Memory Access (Continued)

$\overline{CAS}$n must be toggled using the $\overline{ECAS}$n inputs and $\overline{WIN}$ has to be changed from negated to asserted (read to write) while $\overline{CAS}$ is negated. This method is better than changing $\overline{WIN}$ from negated to asserted in a late write access because here a problem may arise with DATA IN and DATA OUT being valid at the same time. This may result in a data line trying to drive two different levels simultaneously. The page mode method of a read-modify-write access allows the user to have transceivers in the system because the data in (read data) is guaranteed to be high impedance during the time the data out (write data) is valid.

4.0 Refresh Options

The NS32CG821 supports automatic internally controlled refresh. Different types of refreshes can be performed. These different types include all $\overline{RAS}$ refresh and staggered refresh.

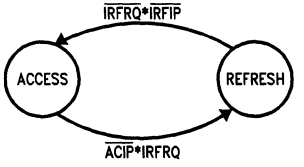
There are two inputs, $\overline{DISRFSH}$ and $\overline{CLRF}$, associated with refresh. There are also ten programming bits; R0–1, R9, C0–6 and $\overline{ECAS}$0 used to program the various types of refreshing.

The NS32CG821 will increment the refresh address counter automatically. The refresh address counter will be incremented once all the refresh $\overline{RAS}$s have been negated.

In every combination of internal refresh and refresh type, the NS32CG821 is programmed to keep $\overline{RAS}$ asserted a number of CLK periods. The values of $\overline{RAS}$ low time during refresh are programmed with the programming bits R0 and R1.

4.1 AUTOMATIC INTERNAL REFRESH

The NS32CG821 has an internal refresh clock. The period of the refresh clock is generated from the programming bits C0–3. Every period of the refresh clock, an internal refresh request is generated. As long as a DRAM access is not currently in progress and precharge time has been met, the internal refresh request will generate an automatic internal refresh. If a DRAM access is in progress, the NS32CG821 on-chip arbitration logic will wait until the access is finished before performing the refresh. The refresh/access arbitration logic can insert a refresh cycle between two accesses. If the two accesses are back to back, the arbitration logic can insert a refresh cycle into the beginning of the next access. The CPU will wait to complete that access until the refresh cycle is completed. However, the refresh arbitration logic can not interrupt an access cycle in progress to perform a refresh. To enable automatic internally controlled refreshes, the input $\overline{DISRFSH}$ must be negated.



TL/F/10126–6

Explanation of Terms

IRFRQ = Internal ReFresh ReQuest of the NS32CG821. IRFRQ has the ability to hold off a pending access.

IRFIP = Internal ReFresh In Progress

ACIP = ACcess In Progress. This means that either RAS is low for an access or is in the process of transitioning low for an access.

FIGURE 6. NS32CG821 Access/Refresh Arbitration State Program

4.2 REFRESH CYCLE TYPES

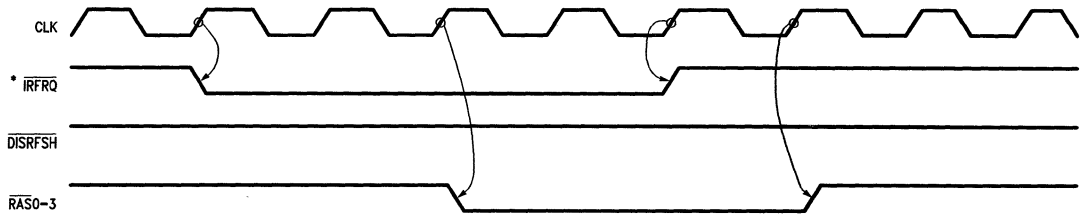
Two different types of refresh cycles are available for use. The two different types are mutually exclusive and can be used with the internal refresh control. The two different refresh cycle types are: all $\overline{RAS}$ refresh and staggered $\overline{RAS}$ refresh. In all refresh cycle types, the $\overline{RAS}$ precharge time is guaranteed: between the previous access $\overline{RAS}$ ending and the refresh $\overline{RAS}$0 starting; between refresh $\overline{RAS}$3 ending and access $\overline{RAS}$ beginning; between burst refresh $\overline{RAS}$s.

4.2.1 Conventional $\overline{RAS}$ Refresh

A conventional refresh cycle causes $\overline{RAS}$0–3 to all assert from the first positive edge of CLK after refresh begins as shown in *Figure 8*. $\overline{RAS}$0–3 will stay asserted until the number of positive edges of CLK programmed have passed. On the last positive edge, $\overline{RAS}$0–3 will be negated and the refresh cycle will end. This type of refresh cycle is programmed by negating address bit R9 during programming.

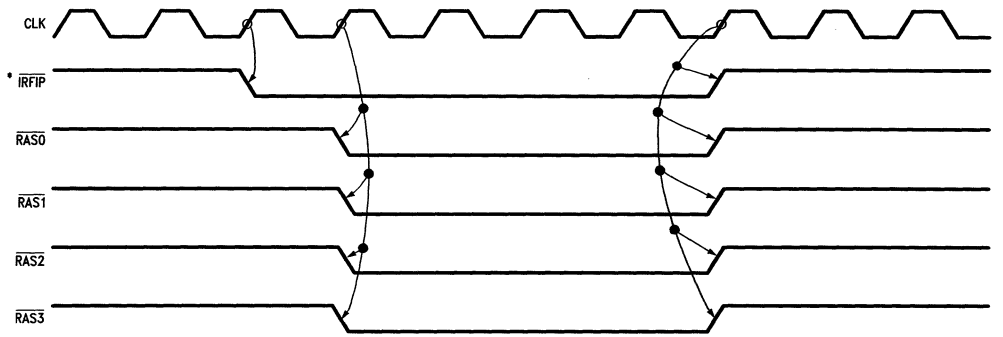
4.2.2 Staggered $\overline{RAS}$ Refresh

A staggered refresh staggers each $\overline{RAS}$ or group of $\overline{RAS}$s by a positive edge of CLK as shown in *Figure 9*. The number of $\overline{RAS}$s, which will be asserted on each positive edge of CLK, is determined by the $\overline{RAS}$, $\overline{CAS}$ configuration mode programming bits C4–C6. If single $\overline{RAS}$ outputs are selected during programming, then each $\overline{RAS}$ will assert on successive positive edges of CLK. If two $\overline{RAS}$ outputs are selected during programming then $\overline{RAS}$0 and $\overline{RAS}$1 will assert on the first positive edge of CLK after refresh cycle begins. $\overline{RAS}$2 and $\overline{RAS}$3 will assert on the second positive edge of CLK after refresh cycle begins. If all $\overline{RAS}$ outputs were selected during programming, all $\overline{RAS}$ outputs would assert on the first positive edge of CLK after refresh cycle begins. Each $\overline{RAS}$ or group of $\overline{RAS}$s will meet the programmed $\overline{RAS}$ low time and then negate.



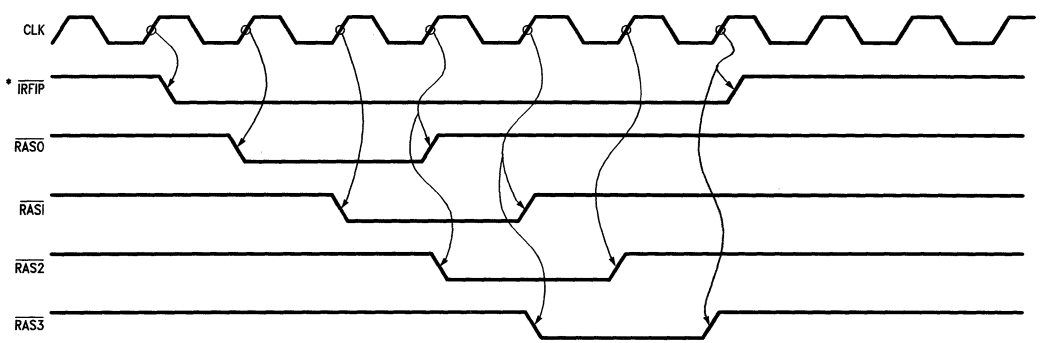
TL/F/10126–7

FIGURE 7. Automatic Internal Refresh with Internal Refresh Request (3T of $\overline{RAS}$ low during refresh programmed)



TL/F/10126–8

FIGURE 8. Conventional $\overline{RAS}$ Refresh



TL/F/10126–9

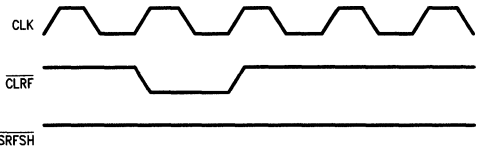
FIGURE 9. Staggered $\overline{RAS}$ Refresh

*$\overline{IRFRQ}$ and $\overline{IRFIP}$ are internal variables, they are only shown in the timing diagrams to make the diagrams easier to understand.

4.0 Refresh Options (Continued)

4.3 CLEARING THE REFRESH ADDRESS COUNTER

The refresh address counter can be cleared by asserting $\overline{CLRF}$ while $\overline{DISRFSH}$ is negated as shown in *Figure 10*. This can be used prior to a burst refresh of the entire memory array. An end-of-count signal can be generated from the Q DRAM address outputs of the NS32CG821 and used to negate $\overline{CLRF}$.



TL/F/10126–10

FIGURE 10. Clearing the Refresh Address Counter

5.0 Wait State Support

Wait states allow a CPU's access cycle to be increased by one or multiple CPU clock periods. By increasing the CPU's access cycle, all signals associated with that access cycle are extended. The CPU samples a wait line to determine if another clock period should be inserted into the access cycle. If another clock period is inserted, the CPU will continue to sample the input every CPU clock period until the input signal changes polarity, allowing the CPU access cycle to terminate. The user determines which value to select for $\overline{WAIT}$ depending upon the CPU speed used and where the user wants the CPU to sample its wait input during an access cycle.

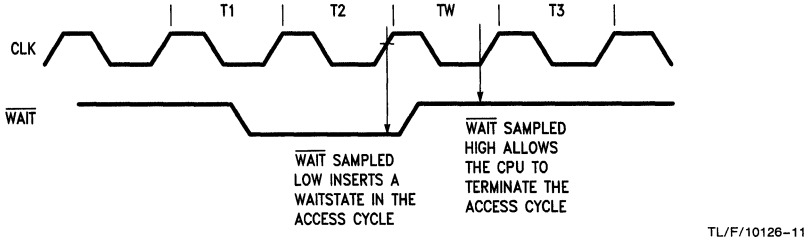
The decision to terminate the CPU access cycle is directly affected by the speed of the DRAMs used. The system designer must ensure that the data from the DRAMs will be present for the CPU to sample or that the data has been written to the DRAM before allowing the CPU access cycle to terminate.

The insertion of wait states also allows a CPU's access cycle to be extended until the DRAM access has taken place. The NS32CG821 inserts wait states into CPU access cycles due to; guaranteeing precharge time, refresh currently in progress, user programmed wait states, and the $\overline{WAITIN}$ signal being asserted. If one of these events is taking place and the CPU starts an access, the NS32CG821 will insert wait states into the access cycle, thereby increasing the length of the CPU's access. Once the event has been completed, the NS32CG821 will allow the access to take place and stop inserting wait states.

There are six programming bits, R2–R7; an input, $\overline{WAITIN}$; and an output that functions as $\overline{WAIT}$.

5.1 WAIT OUTPUT

If $\overline{WAIT}$ is sampled asserted by the CPU, wait states (extra clock periods) are inserted into the current access cycle as shown in *Figure 11*. Once $\overline{WAIT}$ is sampled negated, the access cycle is completed by the CPU. $\overline{WAIT}$ is asserted at the beginning of a chip selected access and is programmed to negate a number of positive edges and/or negative levels of CLK from the event that starts the access. $\overline{WAIT}$ can also be programmed to function in page/burst mode applications. Once $\overline{WAIT}$ is negated during an access, and the $\overline{ECAS}$ inputs are negated with $\overline{TSO}$ asserted, $\overline{WAIT}$ can be programmed to toggle, following the $\overline{ECAS}$ inputs. Once $\overline{TSO}$ is negated, ending the access, $\overline{WAIT}$ will stay negated until the next chip selected access.



TL/F/10126–11

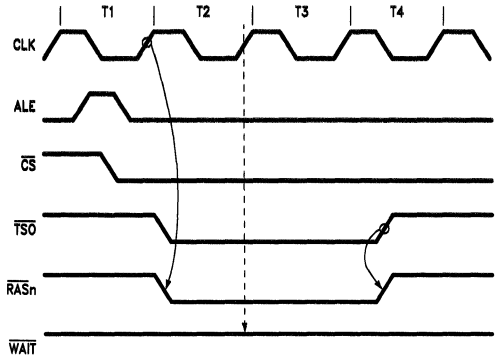
FIGURE 11. $\overline{WAIT}$ Type Output

5.0 Wait State Support (Continued)

5.1.1 Wait during Single Accesses

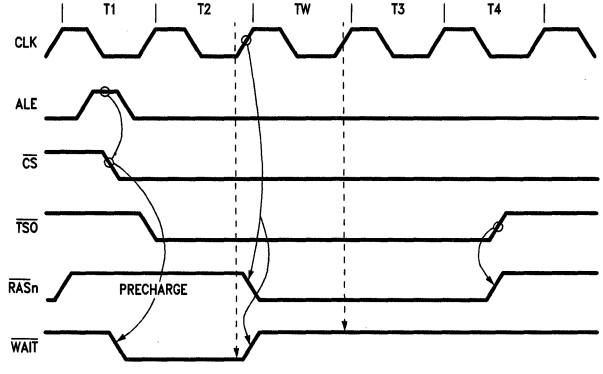
$\overline{WAIT}$ can be programmed to delay a number of positive edges and/or negative levels of CLK. These options are programmed through address bits R2 and R3 at programming time. The user is given four options described below.

0T during non delayed and delayed acceses: $\overline{WAIT}$ will stay negated during a non-delayed access as shown in *Figure 12*. During an access that is delayed, $\overline{WAIT}$ will assert at the start of the access ($\overline{CS}$ and ALE) and negate from the positive edge of $\overline{CLK}$ that starts $\overline{RAS}$ for that access as shown in *Figure 13*.



TL/F/10126–12

FIGURE 12. Non-Delayed Access with $\overline{WAIT}$ 0T ($\overline{WAIT}$ is Sampled at the End of the "T2" Clock State)

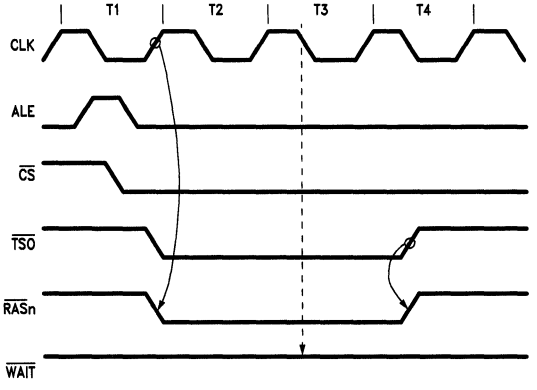


TL/F/10126–13

FIGURE 13. Delayed Access with $\overline{WAIT}$ 0T ("2T" $\overline{RAS}$ Precharge, $\overline{WAIT}$ is Sampled at the End of the "T2" Clock State)

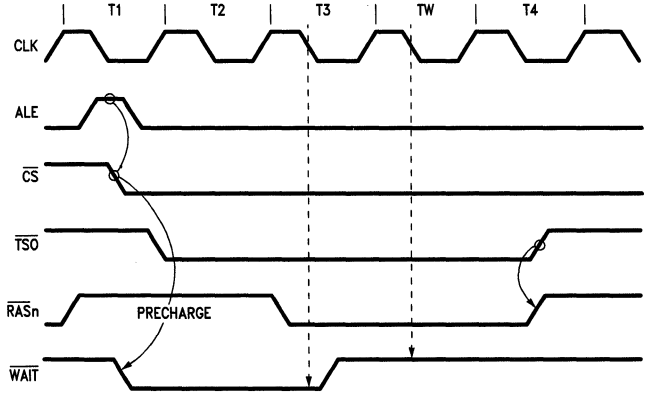
5.0 Wait State Support (Continued)

0T during non-delayed accesses and ½T during delayed accesses: $\overline{WAIT}$ will stay negated during a non-delayed access as shown in *Figure 14*. During an access that is delayed, $\overline{WAIT}$ will assert at the start of the access ($\overline{CS}$ and ALE) and negate on the negative level of CLK after the positive edge of CLK that asserted $\overline{RAS}$ for that access as shown in *Figure 15*.



TL/F/10126–14

FIGURE 14. Non-Delayed Access with $\overline{WAIT}$ 0T ($\overline{WAIT}$ is Sampled at the "T3" Falling Clock Edge)

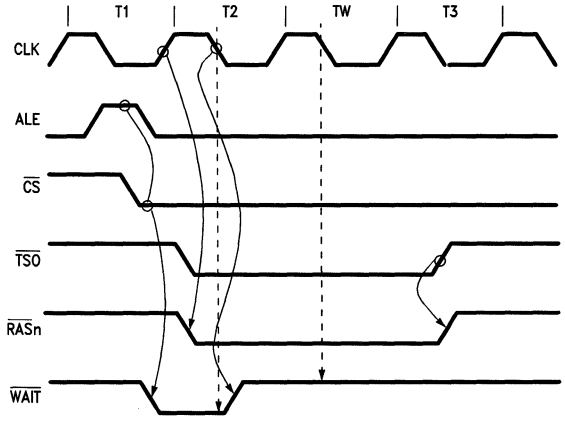


TL/F/10126–15

FIGURE 15. Delayed Access with $\overline{WAIT}$ ½T ($\overline{WAIT}$ is Sampled at the "T3" Falling Clock Edge)

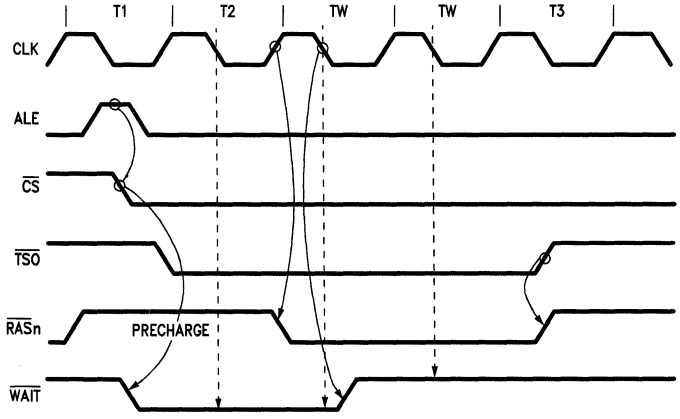
5.0 Wait State Support (Continued)

½T during non-delayed and delayed accesses: $\overline{WAIT}$ will assert when ALE is asserted and $\overline{CS}$ is asserted. $\overline{WAIT}$ will then negate on the negative level of CLK after the positive edge of CLK that asserts $\overline{RAS}$ for the access as shown in *Figure 16*. During delayed accesses in both modes, $\overline{WAIT}$ will assert at the start of the access and negate on the negative level of CLK after the positive edge of CLK that started $\overline{RAS}$ for that access as shown in *Figure 17*.



TL/F/10126–16

FIGURE 16. Non-Delayed Access with $\overline{WAIT}$ ½T ($\overline{WAIT}$ is Sampled at the "T2" Falling Clock Edge)

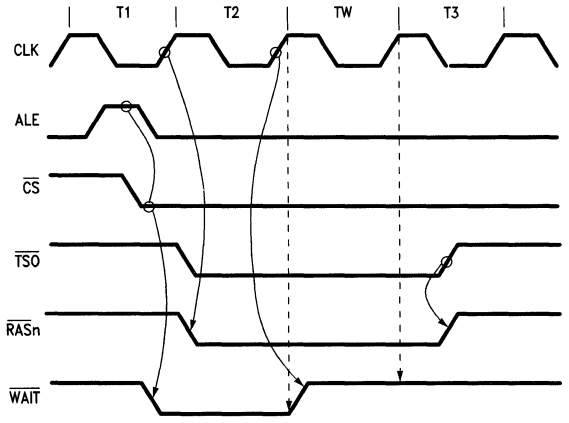


TL/F/10126–17

FIGURE 17. Delayed Access with $\overline{WAIT}$ ½T ($\overline{WAIT}$ is Sampled at the "T2" Falling Clock Edge)

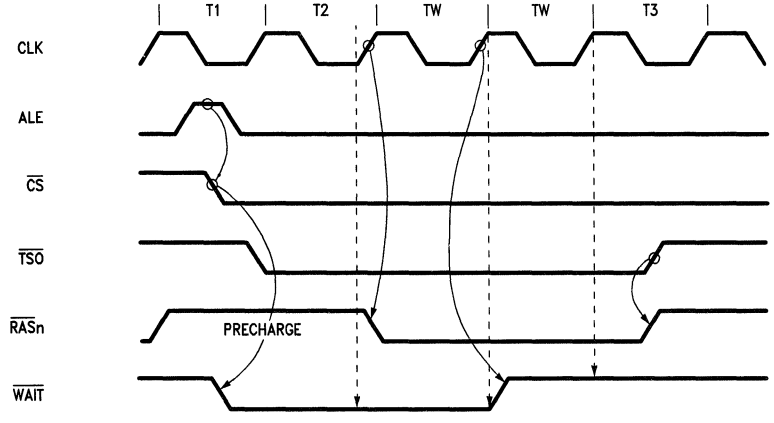
5.0 Wait State Support (Continued)

1T during non-delayed and delayed accesses. $\overline{WAIT}$ will assert from ALE asserted and $\overline{CS}$ asserted. $\overline{WAIT}$ will negate from the next positive edge of CLK that asserts $\overline{RAS}$ for the access as shown in *Figure 18*. During delayed accesses, $\overline{WAIT}$ will assert at the beginning of the access and will negate on the first positive edge of CLK after the positive edge of CLK that starts $\overline{RAS}$ for the access as shown in *Figure 19*.



TL/F/10126–18

FIGURE 18. Non-Delayed Access with $\overline{WAIT}$ 1T ($\overline{WAIT}$ is Sampled at the End of the "T2" Clock State)



TL/F/10126–19

FIGURE 19. Delayed Access with $\overline{WAIT}$ 1T ($\overline{WAIT}$ is Sampled at the End of the "T2" Clock State)

5.0 Wait State Support (Continued)

When ending $\overline{WAIT}$ from a negative level of CLK; if $\overline{RAS}$ is asserted while CLK is high then $\overline{WAIT}$ will negate from the negative edge of CLK; if $\overline{RAS}$ is asserted while CLK is low then $\overline{WAIT}$ will negate from $\overline{RAS}$ asserting. When ending $\overline{WAIT}$ from a positive edge of CLK, the user can think of the positive edge of CLK that starts $\overline{RAS}$ as 0T and the next positive edge of CLK as 1T. In a delayed access, the positive edge of CLK that starts $\overline{RAS}$ can be thought of as 0T and the next positive edge as 1T.

5.1.2 Wait during Page Burst Accesses

$\overline{WAIT}$ can be programmed to function differently during page/burst types of accesses. During a page/burst access, the $\overline{ECAS}$ inputs will be asserted then negated while $\overline{TSO}$ is asserted. Through address bits R4 and R5, $\overline{WAIT}$ can be programmed to assert and negate during this type of access. The user is given four programming options described below.

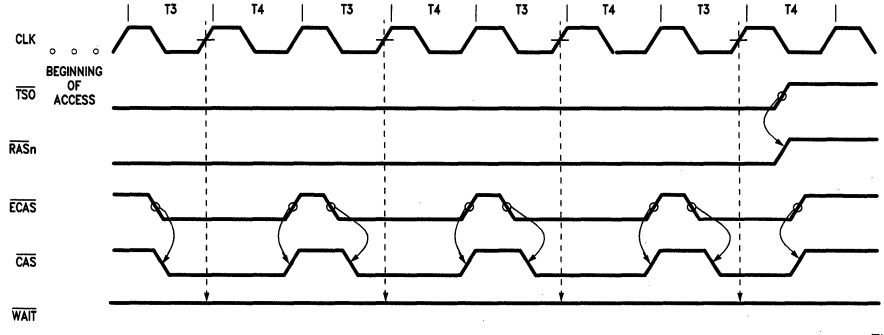
No Wait States: In this case, $\overline{WAIT}$ will remain negated even if the $\overline{ECAS}$ inputs are toggled as shown in *Figure 20*.

0T: $\overline{WAIT}$ will be asserted when the $\overline{ECAS}$ inputs are negated with $\overline{TSO}$ remaining asserted. When a single or group of $\overline{ECAS}$ inputs are asserted, $\overline{WAIT}$ will be negated as shown in *Figure 21*.

½T: $\overline{WAIT}$ will be asserted when the $\overline{ECAS}$ inputs are negated with $\overline{TSO}$ remaining asserted. When a single or group of $\overline{ECAS}$ inputs are asserted again, $\overline{WAIT}$ will be negated from the first negative level of CLK after a single $\overline{ECAS}$ or group of $\overline{ECAS}$s are asserted as shown in *Figure 22*.

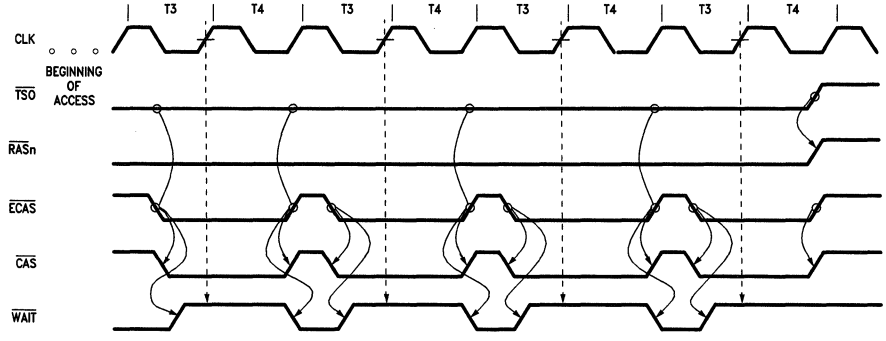
1T: $\overline{WAIT}$ will be asserted when the $\overline{ECAS}$ inputs are negated with $\overline{TSO}$ remaining asserted. When a single or group of $\overline{ECAS}$ inputs are asserted again, $\overline{WAIT}$ will be negated from the first positive edge of CLK after a single $\overline{ECAS}$ or group of $\overline{ECAS}$s are asserted as shown in *Figure 23*.

When ending $\overline{WAIT}$ from a negative level of CLK; if the $\overline{ECAS}$s are asserted while CLK is high then $\overline{WAIT}$ will negate from the negative edge of CLK, if the $\overline{ECAS}$s are asserted while CLK is low then $\overline{WAIT}$ will negate from the $\overline{ECAS}$s asserting. When ending $\overline{WAIT}$ from a positive edge of CLK, the positive edge of CLK that $\overline{ECAS}$ is setup to can be thought of as 1T.



TL/F/10126–20

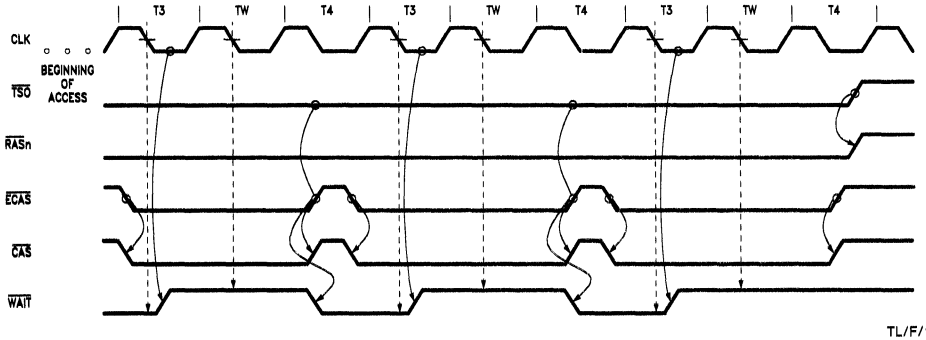
FIGURE 20. No Wait States during Burst ($\overline{WAIT}$ is Sampled at the End of the "T3" Clock State)



TL/F/10126–21

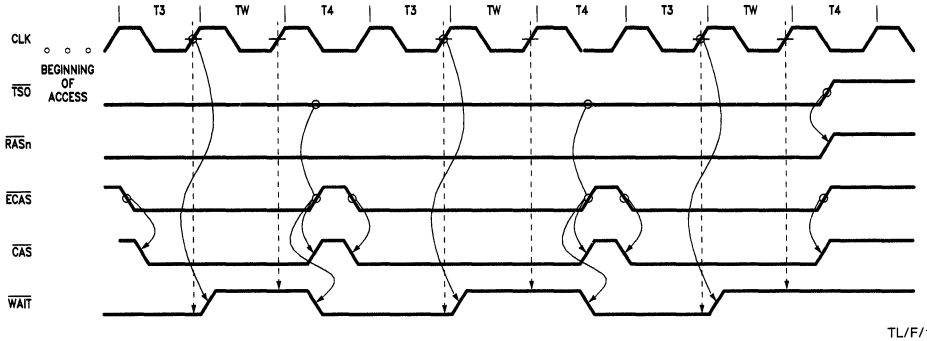
FIGURE 21. 0T during Burst ($\overline{WAIT}$ is Sampled at the End of the "T3" Clock State)

5.0 Wait State Support (Continued)



TL/F/10126-22

FIGURE 22. ½T during Burst Access (WAIT is Sampled at the "T3" Falling Clock Edge)



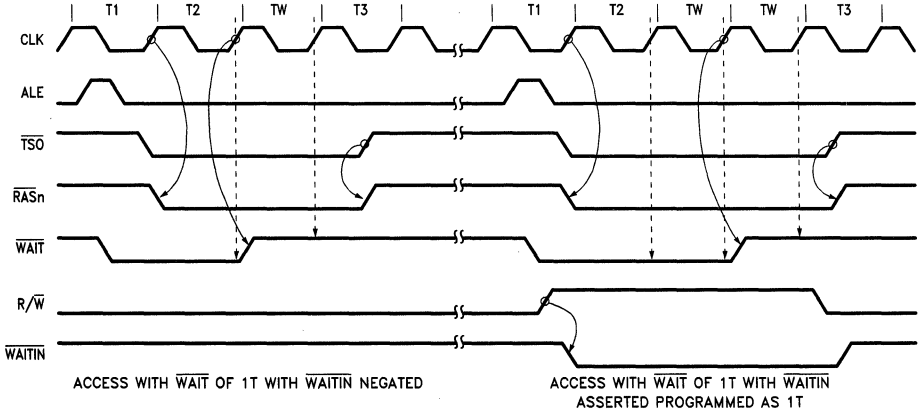
TL/F/10126-23

FIGURE 23. 1T during Burst Access (WAIT is Sampled at the End of the "T3" Clock State)

5.2 DYNAMICALLY INCREASING THE NUMBER OF WAIT STATES

The user can increase the number of positive edges of CLK before WAIT is negated. With the input WAITIN asserted, the user can delay WAIT negating either one or two more positive edges of CLK. The number of edges is programmed through address bit R6. If the user is increasing the number of positive edges in a delay that contains a negative level, the positive edges will be met before the negative level.

WAITIN can increase the number of positive edges in a page/burst access. WAITIN can be permanently asserted in systems requiring an increased number of wait states. WAITIN can also be asserted and negated, depending on the type of access. As an example, a user could connect the DDIN output from the NS32CG16 to the WAITIN input. This could be used to perform write accesses with 1 wait state and read accesses with 2 wait states as shown in *Figure 24*.



TL/F/10126-24

FIGURE 24. WAITIN Example (WAIT is Sampled at the End of "T2")

5.0 Wait State Support (Continued)

5.3 GUARANTEEING $\overline{RAS}$ LOW TIME AND $\overline{RAS}$ PRECHARGE TIME

The NS32CG821 will guarantee $\overline{RAS}$ precharge time between accesses; between refreshes; and between access and refreshes. The programming bits R0 and R1 are used to program combinations of $\overline{RAS}$ precharge time and $\overline{RAS}$ low time referenced by positive edges of CLK. $\overline{RAS}$ low time is programmed for refreshes only. During an access, the system designer guarantees the time $\overline{RAS}$ is asserted through the NS32CG821 wait logic. Since inserting wait states into an access increases the length of the CPU signals which are used to create ALE and $\overline{TSO}$, the time that $\overline{RAS}$ is asserted can be guaranteed.

Precharge time is also guaranteed by the NS32CG821. Each $\overline{RAS}$ output has a separate positive edge of CLK counter. $\overline{TSO}$ is negated setup to a positive edge of CLK to terminate the access. That positive edge is 1T. The next positive edge is 2T. $\overline{RAS}$ will not be asserted until the programmed number of positive edges of CLK have passed. Once the programmed precharge time has been met, $\overline{RAS}$ will be asserted from the positive edge of CLK. However, since there is a precharge counter per $\overline{RAS}$, an access using another $\overline{RAS}$ will not be delayed. Precharge time before a refresh is always referenced from the access $\overline{RAS}$ negating before $\overline{RAS}$0 for the refresh asserting. After a refresh, precharge time is referenced from $\overline{RAS}$3 negating, for the refresh, to the access $\overline{RAS}$ asserting.

6.0 Additional Access Support Features

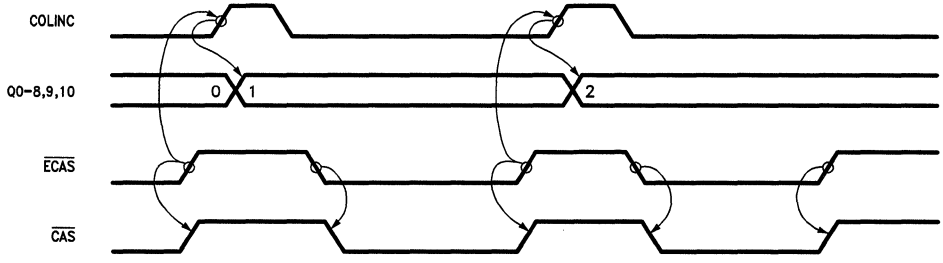
To support the different modes of accessing, the NS32CG821 have multiple access features. These features allow the user to take advantage of CPU or DRAM functions. These additional features include: address latches and column increment for page/burst mode support; and delay $\overline{CAS}$, to allow the user with a multiplexed bus to ensure valid data is present before $\overline{CAS}$ is asserted.

6.1 ADDRESS LATCHES AND COLUMN INCREMENT

The address latches can be programmed, through programming bit B0, to either latch the address or remain permanently in fall-through mode. If the address latches are used to latch the address, the rising edge of ALE places the latches in fall-through. Once ALE is negated, the address present on the row, column and bank inputs is latched.

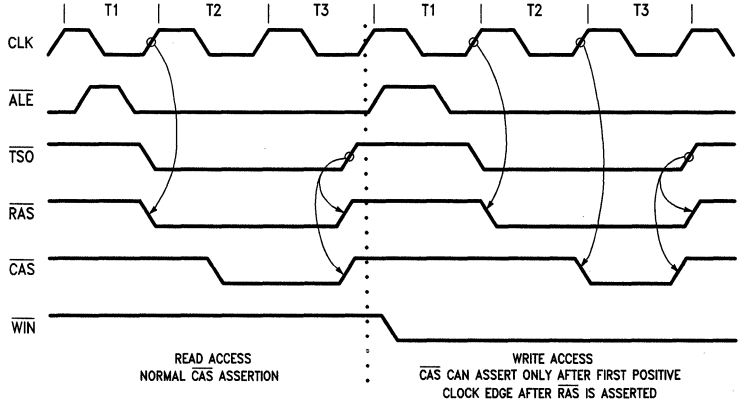
Once the address is latched, the column address can be incremented with the input COLINC. With COLINC asserted, the column address is incremented.

If COLINC is asserted with all of the bits of the column address asserted, the column address will return to zero. COLINC can be used for sequential accesses of static column DRAMs. COLINC can also be used with the $\overline{ECAS}$ inputs to support sequential accesses to page mode DRAMs as shown in *Figure 25*. COLINC should only be asserted during an access.



TL/F/10126–25

FIGURE 25. Column Increment



TL/F/10126–26

FIGURE 26. Delay $\overline{CAS}$

6.0 Additional Access Support Features (Continued)

6.2 DELAY $\overline{CAS}$ DURING WRITE ACCESSES

Address bit C9 asserted during programming will cause $\overline{CAS}$ to be delayed until the first positive edge of CLK after $\overline{RAS}$ is asserted when the input $\overline{WIN}$ is asserted. Delaying $\overline{CAS}$ during write accesses ensures that the data to be written to DRAM will be setup to $\overline{CAS}$ asserting as shown in *Figure 26*. If the possibility exists that data still may not be present after the first positive edge of CLK, $\overline{CAS}$ can be delayed further with the $\overline{ECAS}$ inputs. If address bit C9 is negated during programming, read and write accesses will be treated the same (with regard to $\overline{CAS}$).

7.0 $\overline{RAS}$ and $\overline{CAS}$ Configuration Modes

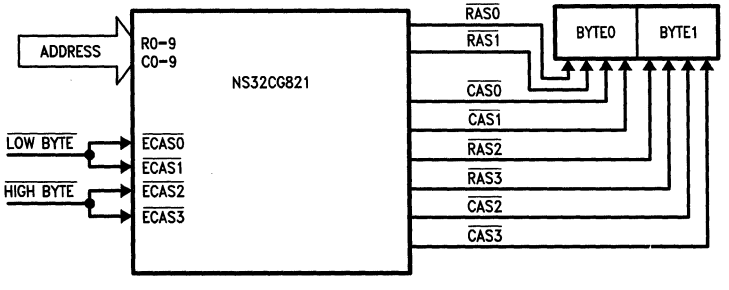
The NS32CG821 allow the user to configure the DRAM array to contain one, two, four or eight banks of DRAM. Depending on the functions used, certain considerations must be used when determining how to set up the DRAM array. Programming address bits C4, C5 and C6 along with bank selects, B0–1, and $\overline{CAS}$ enables, $\overline{ECAS}$0–3, determine which $\overline{RAS}$ or group of $\overline{RAS}$s and which $\overline{CAS}$ or group of $\overline{CAS}$s will be asserted during an access. Different memory schemes are described. The NS32CG821 is specified driving a heavy load of 72 DRAMs, representing four banks of DRAM with 16-bit words and 2 parity bits. The NS32CG821 can drive more than 72 DRAMs, but the AC timing must be increased. Since the $\overline{RAS}$ and $\overline{CAS}$ outputs are configurable, all $\overline{RAS}$ and $\overline{CAS}$ outputs should be used for the maximum amount of drive.

7.1 BYTE WRITING

By selecting a configuration in which all $\overline{CAS}$ outputs are selected during an access, the $\overline{ECAS}$ inputs enable a single or group of $\overline{CAS}$ outputs to select a byte (or bytes) in a word. In this case, the $\overline{RAS}$ outputs are used to select which of up to 4 banks is to be used as shown in *Figure 29*. In systems with a word size of 16 bits, the byte enables can be gated with a high order address bit to produce four byte enables which gives an equivalent to 8 banks of 16-bit words as shown in *Figure 30*. If less memory is required, each $\overline{CAS}$ should be used to drive each nibble in the 16-bit word as shown in *Figures 27* and *28*.

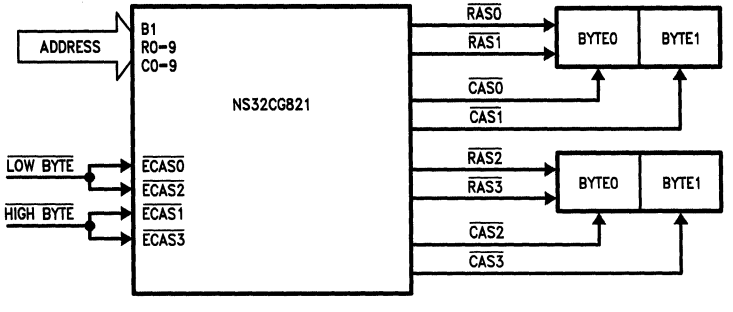
7.2 MEMORY INTERLEAVING

Memory interleaving allows the cycle time of DRAMs to be reduced by having sequential accesses to different memory banks. Since the NS32CG821 have separate precharge counters per bank, sequential accesses will not be delayed if the accessed banks use different $\overline{RAS}$ outputs. To ensure different $\overline{RAS}$ outputs will be used, a mode is selected where either one or two $\overline{RAS}$ outputs will be asserted during an access. The bank select or selects, B0 and B1, are then tied to the least significant address bits, causing a different



TL/F/10126–27

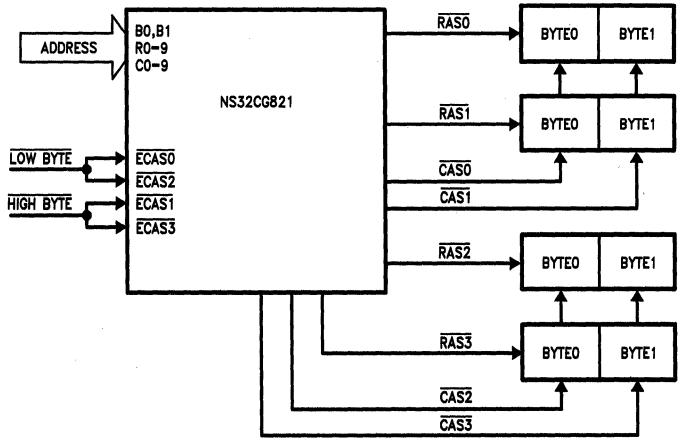
FIGURE 27. 1 Bank DRAM Array Setup for 16-Bit System (C_6, C_5, C_4 = 011 during Programming)



TL/F/10126–28

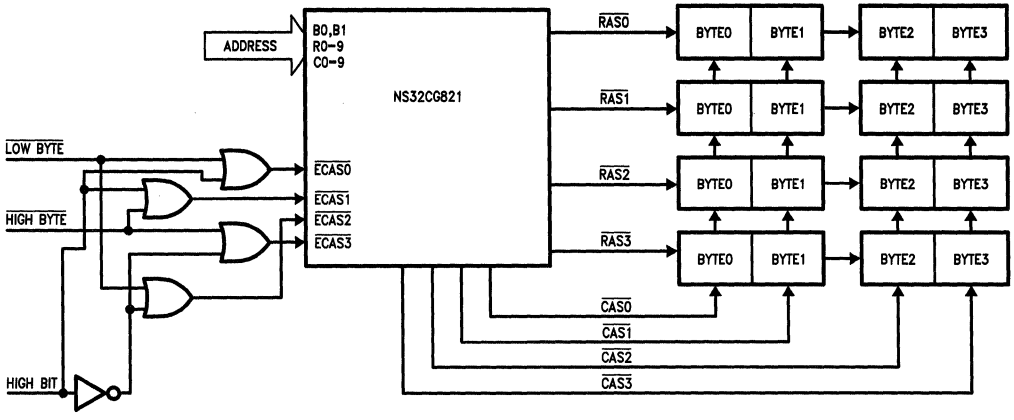
FIGURE 28. 2 Banks DRAM Array Setup for 16-Bit System (C_6, C_5, C_4 = 101 during Programming)

7.0 $\overline{RAS}$ and $\overline{CAS}$ Configuration Modes (Continued)



TL/F/10126–29

FIGURE 29. 4 Banks Array Setup for 16-Bit System (C_6, C_5, C_4 = 110 during Programming)



TL/F/10126–30

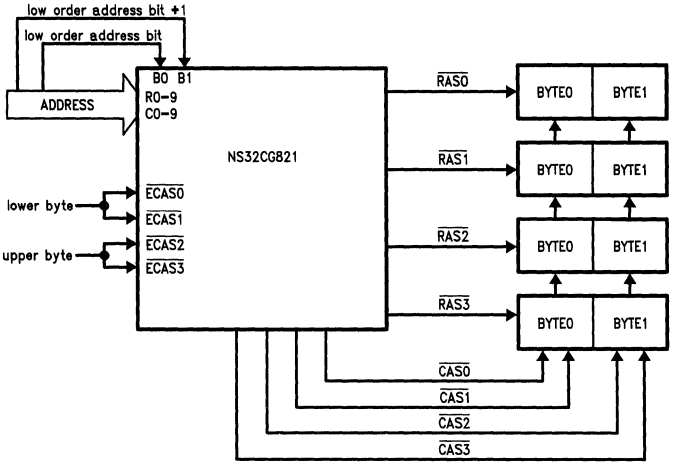
FIGURE 30. 8 Bank DRAM Array for 16-Bit System (C_6, C_5, C_4 = 1, 1, 0 during Programming)

7.0 $\overline{RAS}$ and $\overline{CAS}$ Configuration Modes (Continued)

group of $\overline{RAS}$s to assert during each sequential access as shown in *Figure 31*. In this figure there should be at least one clock period of all $\overline{RAS}$'s negated between different $\overline{RAS}$'s being asserted to avoid the condition of a $\overline{CAS}$ before $\overline{RAS}$ refresh cycle.

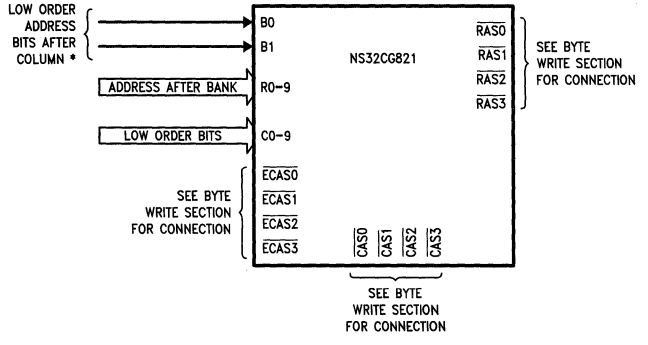
7.3 PAGE/BURST MODE

In a static column, page or burst mode system, the least significant bits must be tied to the column address in order to ensure that the page/burst accesses are to sequential memory addresses, as shown in *Figure 32*. The $\overline{ECAS}$ inputs may then be toggled with the NS32CG821's address latches in fall-through mode, while $\overline{TSO}$ is asserted. The $\overline{ECAS}$ inputs can also be used to select individual bytes. In page or static column modes, the two address bits after the page size can be tied to the bank select inputs to select a new bank if the page size is exceeded.



TL/F/10126–31

FIGURE 31. Memory Interleaving (C6, C5, C4 = 1, 1, 0 during Programming)



TL/F/10126–32

*See table below for row, column & bank address bit map. A0 is used for byte addressing in this example.

Addresses	Page Mode/Static Column Mode Page Size		
	256 Bits/Page	512 Bits/Page	1024 Bits/Page
Column Address	C0–7 = A1–8 C8–9 = X	C0–8 = A1–9 C9 = X	C0–9 = A1–10
Row Address	X	X	X
B0 B1	A9 A10	A10 A11	A11 A12

X = DON'T CARE, the user can do as he pleases.

FIGURE 32. Page, Static Column, Mode System

8.0 Programming and Resetting

The NS32CG821 must be programmed by one of two possible programming sequences before it can be used. Once the chip is programmed, the bits effecting the wait logic become effective immediately, thus allowing the programming bus cycle to end. At power up, the NS32CG821 programming bits are in an undefined state. All internal latches and flip-flops are cleared. After programming, the NS32CG821 enters a 60 ms initialization period. During this initialization period, the NS32CG821 performs refreshes about every 15 μs; this makes further DRAM warmup cycles unnecessary. The chip can be programmed as many times as the user wishes. After the first programming, the 60 ms initialization period will not be entered into unless the chip is reset. During the 60 ms initialization period, internal refreshes are taking place and the CPU is not allowed to enter into a memory access cycle. If a memory access is attempted, the NS32CG821 will send out wait states into the processor until initialization is complete. The actual initialization time period is given by the following formula:

T = 4096*(Clock Divisor Select)
*(Refresh Clock Fine Tune)
/(DELCK Frequency)

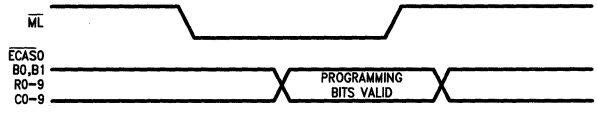
8.1 MODE LOAD ONLY PROGRAMMING

MODE LOAD, $\overline{ML}$, asserted enables an internal 23-bit programmable register. To use this method, the user asserts $\overline{ML}$, enabling the internal programming register. After $\overline{ML}$ is asserted, a valid programming selection is placed on the address bus (and $\overline{ECAS0}$), then $\overline{ML}$ is negated. When $\overline{ML}$ is negated, the value on the address bus (and $\overline{ECAS0}$) is latched into the internal programming register and the NS32CG821 is programmed, as shown in *Figure 33*. After $\overline{ML}$ is negated, the NS32CG821 will enter the 60 ms initialization period only if this is the first programming after power up or reset.

Using this method, a set of transceivers on the address bus can be put at TRI-STATE® by the system reset signal. A combination of pull-up and pull-down resistors can be used on the address inputs of the NS32CG821 to select the programming values, as shown in *Figure 34*.

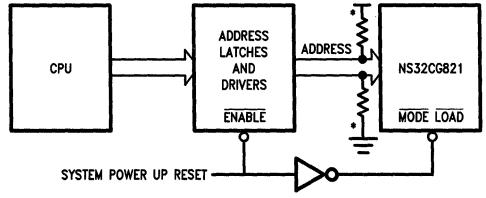
8.2 CHIP SELECTED ACCESS PROGRAMMING

The chip can also be programmed by asserting $\overline{ML}$ and performing a chip selected access. ALE is disabled internally until after programming. To program the chip using this method, $\overline{ML}$ is asserted. After $\overline{ML}$ is asserted, $\overline{CS}$ is asserted and a valid programming selection is placed on the address bus. When $\overline{TSO}$ is asserted, the chip is programmed with the programming selection on the address bus. After $\overline{TSO}$ is negated, $\overline{ML}$ can be negated as shown in *Figure 35*.



TL/F/10126–33

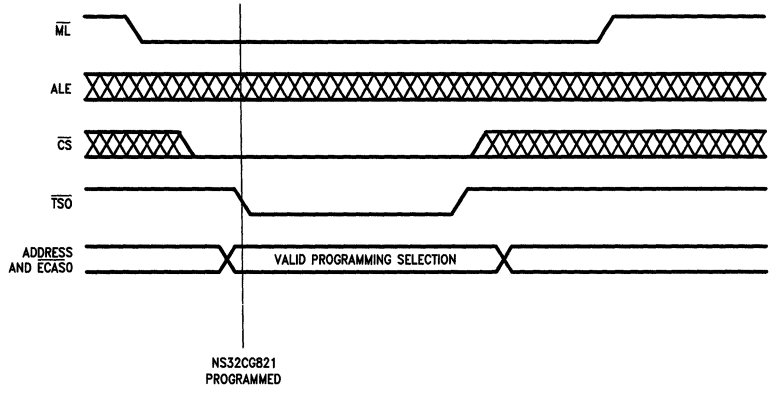
FIGURE 33. Mode Load Only Programming



TL/F/10126–34

*Pull-Up or Pull-Down Resistors on Each Address Input

FIGURE 34. Programming during System Reset

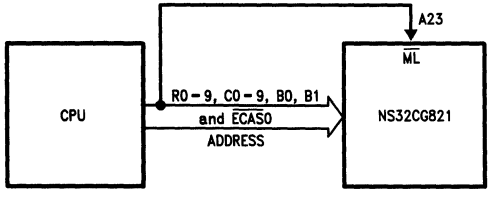


TL/F/10126–35

FIGURE 35. $\overline{CS}$ Access Programming

8.0 Programming and Resetting (Continued)

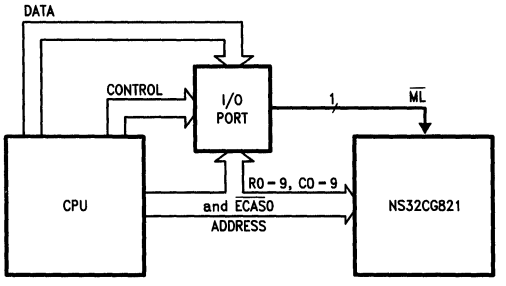
Using this method, various programming schemes can be used. For example if extra upper address bits are available, an unused high order address bit can be tied to the signal $\overline{ML}$. Using this method, one need only write to a page of memory, thus asserting the high order bit and in turn programming the chip as shown in *Figure 36*.



TL/F/10126–36

FIGURE 36. Programming the NS32CG821 through the Address Bus Only

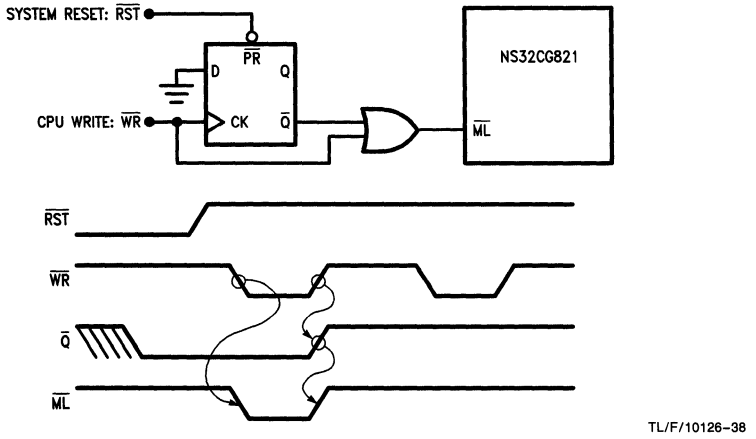
An I/O port can also be used to assert $\overline{ML}$. After $\overline{ML}$ is asserted, a chip selected access can be performed to program the chip. After the chip selected access, $\overline{ML}$ can be negated through the I/O port as shown in *Figure 37*.



TL/F/10126–37

FIGURE 37. Programming the NS32CG821 through the Address Bus and an I/O Port

Another simple way the chip can be programmed is the first write after system reset. This method requires only a flip-flop and an OR gate as shown in *Figure 38*. At reset, the flip-flop is preset, which pulls the $\overline{Q}$ output low. Since $\overline{WR}$ is negated, $\overline{ML}$ is not enabled. The first write access is used to program the chip. When $\overline{WR}$ is asserted, $\overline{ML}$ is asserted. $\overline{WR}$ negated clocks the flip-flop, negates $\overline{ML}$, and programs the NS32CG821 with the address and $\overline{ECAS0}$ available at that time. $\overline{CS}$ does not need to be asserted using this method.



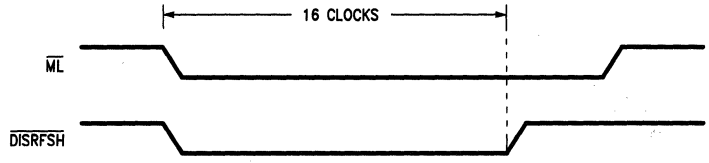
TL/F/10126–38

FIGURE 38. Programming the NS32CG821 on the First CPU Write after Power Up

8.0 Programming and Resetting (Continued)

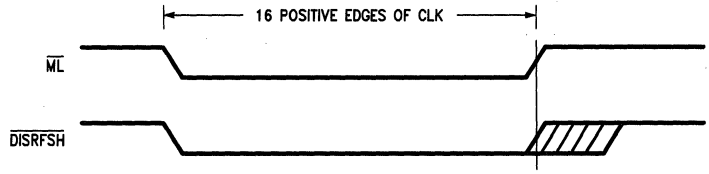
8.3 EXTERNAL RESET

At power up, all internal latches and flip-flops are cleared. The power up state can again be entered by asserting $\overline{ML}$ and $\overline{DISRFSH}$ for 16 positive edges of CLK. After resetting if the user negates $\overline{DISRFSH}$ before negating $\overline{ML}$ as shown in *Figure 39*, $\overline{ML}$ negated will program the chip. If $\overline{ML}$ is negated before or at the same time as $\overline{DISRFSH}$ as shown in *Figure 40*, the chip will not be programmed. After the chip is programmed, the 60 ms initialization period will be entered into if this is the first programming after power up or reset.



TL/F/10126–39

FIGURE 39. Chip Reset and Programmed



TL/F/10126–40

FIGURE 40. Chip Reset but Not Programmed

8.0 Programming and Resetting (Continued)

8.4 PROGRAMMING BIT DEFINITIONS (Notes 1 and 2)

Symbol	Description
B0	**Address Latch Mode**
0	ALE asserted latches the input row, column and bank address.
1	The row, column and bank latches are fall through.
C9	**Delay $\overline{CAS}$ during WRITE Accesses**
0	$\overline{CAS}$ is treated the same for both READ and WRITE accesses.
1	During WRITE accesses, $\overline{CAS}$ will be asserted by the event that occurs last: $\overline{CAS}$ asserted by the internal delay line or $\overline{CAS}$ asserted on the positive edge of CLK after $\overline{RAS}$ is asserted.
C8	**Row Address Hold Time**
0	Row Address Hold Time = 25 ns minimum
1	Row Address Hold Time = 15 ns minimum
C7	**Column Address Setup Time**
0	Column Address Setup Time = 10 ns minimum
1	Column Address Setup Time = 0 ns minimum
C6, C5, C4	**$\overline{RAS}$ and $\overline{CAS}$ Configuration Modes**
0, 1, 1	$\overline{RAS}$0–3 and $\overline{CAS}$0–3 are all selected during an access. $\overline{ECAS}$n must be asserted for $\overline{CAS}$n to be asserted. B1, B0 are not used during an access.
1, 0, 1	$\overline{RAS}$ and $\overline{CAS}$ pairs are selected by B1. $\overline{ECAS}$n must be asserted for $\overline{CAS}$n to be asserted. B1 = 0 during an access selects $\overline{RAS}$0–1 and $\overline{CAS}$0–1. B1 = 1 during an access selects $\overline{RAS}$2–3 and $\overline{CAS}$2–3. B0 is not used during an access.
1, 1, 0	$\overline{RAS}$ singles are selected by B0–1. $\overline{CAS}$0–3 are all selected. $\overline{ECAS}$n must be asserted for $\overline{CAS}$n to be asserted. B1 = 0, B0 = 0 during an access selects $\overline{RAS}$0 and $\overline{CAS}$0–3. B1 = 0, B0 = 1 during an access selects $\overline{RAS}$1 and $\overline{CAS}$0–3. B1 = 1, B0 = 0 during an access selects $\overline{RAS}$2 and $\overline{CAS}$0–3. B1 = 1, B0 = 1 during an access selects $\overline{RAS}$3 and $\overline{CAS}$0–3.
C3	**Refresh Clock Fine Tune Divisor**
0	Divide delay line/refresh clock further by 30 (If DELCLK/Refresh Clock Clock Divisor = 2 MHz = 15 μs refresh period).
1	Divide delay line/refresh clock further by 26 (If DELCLK/Refresh Clock Clock Divisor = 2 MHz = 13 μs refresh period).
C2, C1, C0	**Delay Line/Refresh Clock Divisor Select**
0, 0, 0	Divide DELCLK by 10 to get as close to 2 MHz as possible.
0, 0, 1	Divide DELCLK by 9 to get as close to 2 MHz as possible.
0, 1, 0	Divide DELCLK by 8 to get as close to 2 MHz as possible.
0, 1, 1	Divide DELCLK by 7 to get as close to 2 MHz as possible.
1, 0, 0	Divide DELCLK by 6 to get as close to 2 MHz as possible.
1, 0, 1	Divide DELCLK by 5 to get as close to 2 MHz as possible.
1, 1, 0	Divide DELCLK by 4 to get as close to 2 MHz as possible.
1, 1, 1	Divide DELCLK by 3 to get as close to 2 MHz as possible.
R9	**Refresh Mode Select**
0	$\overline{RAS}$0–3 will all assert and negate at the same time during a refresh.
1	Staggered Refresh. $\overline{RAS}$ outputs during refresh are separated by one positive clock edge. Depending on the configuration mode.
R6	**Add Wait States to the Current Access if $\overline{WAITIN}$ is Low.**
0	$\overline{WAIT}$ will be delayed by one additional positive edge of CLK.
1	$\overline{WAIT}$ will be delayed by two additional positive edges of CLK.

8.0 Programming and Resetting (Continued)

8.4 PROGRAMMING BIT DEFINITIONS (Continued)

Symbol	Description
R5, R4	**$\overline{\text{WAIT}}$ during Burst (See Section 5.1.2)**
0, 0	NO WAIT STATES; $\overline{\text{WAIT}}$ will remain negated during burst portion of access.
0, 1	1T; $\overline{\text{WAIT}}$ will assert when the $\overline{\text{ECAS}}$ inputs are negated with $\overline{\text{TSO}}$ asserted. $\overline{\text{WAIT}}$ will negate from the positive edge of CLK after the $\overline{\text{ECAS}}$s have been asserted.
1, 0	½T; $\overline{\text{WAIT}}$ will assert when the $\overline{\text{ECAS}}$ inputs are negated with $\overline{\text{TSO}}$ asserted. $\overline{\text{WAIT}}$ will negate on the negative level of CLK after the $\overline{\text{ECAS}}$s have been asserted.
1, 1	0T; $\overline{\text{WAIT}}$ will assert when the $\overline{\text{ECAS}}$ inputs are negated. $\overline{\text{WAIT}}$ will negate when the $\overline{\text{ECAS}}$ inputs are asserted.
R3, R2	**$\overline{\text{WAIT}}$ Delay Times (See Section 5.1.1)**
0, 0	NO WAIT STATES; $\overline{\text{WAIT}}$ will remain high during non-delayed accesses. $\overline{\text{WAIT}}$ will negate when $\overline{\text{RAS}}$ is negated during delayed accesses.
0, 1	½T; $\overline{\text{WAIT}}$ will negate on the negative level of CLK, after the access $\overline{\text{RAS}}$.
1, 0	NO WAIT STATES, ½ T; $\overline{\text{WAIT}}$ will remain high during non-delayed accesses. $\overline{\text{WAIT}}$ will negate on the negative level of CLK, after the access $\overline{\text{RAS}}$, during delayed accesses.
1, 1	1T; $\overline{\text{WAIT}}$ will negate on the positive edge of CLK after the access $\overline{\text{RAS}}$.
R1, R0	**$\overline{\text{RAS}}$ Low and $\overline{\text{RAS}}$ Precharge Time**
0, 0	$\overline{\text{RAS}}$ asserted during refresh = 2 positive edges of CLK. $\overline{\text{RAS}}$ precharge time = 1 positive edge of CLK.
0, 1	$\overline{\text{RAS}}$ asserted during refresh = 3 positive edges of CLK. $\overline{\text{RAS}}$ precharge time = 2 positive edges of CLK.
1, 0	$\overline{\text{RAS}}$ asserted during refresh = 2 positive edges of CLK. $\overline{\text{RAS}}$ precharge time = 2 positive edges of CLK.
1, 1	$\overline{\text{RAS}}$ asserted during refresh = 4 positive edges of CLK. $\overline{\text{RAS}}$ precharge time = 3 positive edges of CLK.

Note 1: During programming $\overline{\text{ECAS}}_0$, B_1, R_7 have to be set to low, and R_8 has to be set high.

Note 2: $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ configuration modes C_6, C_5, C_4 = 000, 001, 010, 100 and 111 are reserved.

9.0 DRAM Critical Timing Parameters

The two critical timing parameters, shown in *Figure 41*, that must be met when controlling the access timing to a DRAM are the row address hold time, tRAH, and the column address setup time, tASC. Since the NS32CG821 contain a precise internal delay line, the values of these parameters can be selected at programming time. These values will also increase and decrease if DELCLK varies from 2 MHz.

9.1 PROGRAMMABLE VALUES OF tRAH AND tASC

The NS32CG821 allow the values of tRAH and tASC to be selected at programming time. For each parameter, two choices can be selected. tRAH, the row address hold time, is measured from $\overline{RAS}$ asserted to the row address starting to change to the column address. The two choices for tRAH are 15 ns and 25 ns, programmable through address bit C8.

tASC, the column address setup time, is measured from the column address valid to $\overline{CAS}$ asserted. The two choices for tASC are 0 ns and 10 ns, programmable through address bit C7.

9.2 CALCULATION OF tRAH AND tASC

There are two clock inputs to the NS32CG821. These two clocks, DELCLK and CLK can either be tied together to the same clock or be tied to different clocks running asynchronously at different frequencies.

The clock input, DELCLK, controls the internal delay line and refresh request clock. DELCLK should be a multiple of 2 MHz. If DELCLK is not a multiple of 2 MHz, tRAH and tASC will change. The new values of tRAH and tASC can be calculated by the following formulas:

If tRAH was programmed to equal 15 ns then tRAH = 30*(((DELCLK Divisor)* 2 MHz/(DELCLK Frequency))−1) + 15 ns.

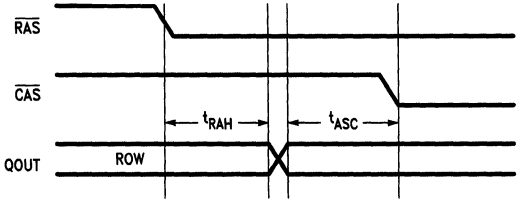
If tRAH was programmed to equal 25 ns then tRAH = 30*(((DELCLK Divisor)* 2 MHz/(DELCLK Frequency))−1) + 25 ns.

If tASC was programmed to equal 0 ns then tASC = 15* ((DELCLK Divisor)* 2 MHz/(DELCLK Frequency)) − 15 ns.

If tASC was programmed to equal 10 ns then tASC = 25* ((DELCLK Divisor)* 2 MHz/(DELCLK Frequency)) − 15 ns.

Since the values of tRAH and tASC are increased or decreased, the time to $\overline{CAS}$ asserted will also increase or decrease. These parameters can be adjusted by the following formula:

Delay to $\overline{CAS}$ = Actual Spec. + Actual tRAH − Programmed tRAH + Actual tASC − Programmed tASC.



TL/F/10126–41

FIGURE 41. tRAH and tASC

10.0 Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Temperature under Bias ..... 0°C to +70°C
Storage Temperature ..... −65°C to +150°C
All Input or Output Voltage with Respect to GND ..... −0.5V to +7V
Power Dissipation @ 20 MHz 0.5W
ESD Rating to be determined.

11.0 DC Electrical Characteristics T_A = 0°C to +70°C, V_{CC} = 5V ±10%, GND = 0V

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IH}	Logical 1 Input Voltage	Tested with a Limited Functional Pattern	2.0		V_{CC} + 0.5	V
V_{IL}	Logical 0 Input Voltage	Tested with a Limited Functional Pattern	−0.5		0.8	V
V_{OH1}	Q and $\overline{WE}$ Outputs	I_{OH} = −10 mA	V_{CC} − 1.0			V
V_{OL1}	Q and $\overline{WE}$ Outputs	I_{OL} = 10 mA			0.5	V
V_{OH2}	All Outputs except Qs, $\overline{WE}$	I_{OH} = −3 mA	V_{CC} − 1.0			V
V_{OL2}	All Outputs except Qs, $\overline{WE}$	I_{OL} = 3 mA			0.5	V
I_{IN}	Input Leakage Current	V_{IN} = V_{CC} or GND	−10		10	μA
$I_{IL\ ML}$	$\overline{ML}$ Input Current (Low)	V_{IN} = GND			200	μA
I_{CC1}	Standby Current	CLK at 8 MHz (V_{IN} = V_{CC} or GND)		6	15	mA
I_{CC1}	Standby Current	CLK at 20 MHz (V_{IN} = V_{CC} or GND)		8	17	mA
I_{CC1}	Standby Current	CLK at 25 MHz (V_{IN} = V_{CC} or GND)		10	20	mA
I_{CC2}	Supply Current	CLK at 8 MHz (Inputs Active) (I_{LOAD} = 0) (V_{IN} = V_{CC} or GND)		20	40	mA
I_{CC2}	Supply Current	CLK at 20 MHz (Inputs Active) (I_{LOAD} = 0) (V_{IN} = V_{CC} or GND)		40	75	mA
I_{CC2}	Supply Current	CLK at 25 MHz (Inputs Active) (I_{LOAD} = 0) (V_{IN} = V_{CC} or GND)		50	95	mA
C_{IN}*	Input Capacitance	f_{IN} at 1 MHz			10	pF

***Note: C_{IN} is not 100% tested.**

12.0 AC Timing Parameters: NS32CG821

Two speed selections are given, the NS32CG821-20 and the NS32CG821-25. The differences between the two parts are the maximum operating frequencies of the input CLKs and the maximum delay specifications. Low frequency applications may use the "−25" part to gain improved timing.

The AC timing parameters are grouped into sectional numbers as shown below. These numbers also refer to the timing diagrams.

1–36 Common parameters to all modes of operation

50–56 Difference parameters used to calculate;
$\overline{RAS}$ low time,
$\overline{RAS}$ precharge time,
$\overline{CAS}$ high time and
$\overline{CAS}$ low time

200–212 Refresh parameters

300–315 Memory access parameters used in both single and dual access applications

500–506 Programming parameters

Unless otherwise stated V_{CC} = 5.0V ±10%, 0 < T_A < 70°C, the output load capacitance is typical for 4 banks of 18 DRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:

C_L = 50 pF loads on all outputs except
C_L = 150 pF loads on Q0–9 and $\overline{WE}$; or

C_H = 50 pF loads on all outputs except
C_H = 125 pF loads on $\overline{RAS}$0–3 and $\overline{CAS}$0–3 and
C_H = 380 pF loads on Q0–9 and $\overline{WE}$.

12.0 AC Timing Parameters: NS32CG821 (Continued)

Unless otherwise stated V_{CC} = 5.0V ±10%, 0°C < T_A < 70°C, the output load capacitance is typical for 4 banks of 18 DRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:

C_L = 50 pF loads on all outputs except
C_L = 150 pF loads on Q0–9 and $\overline{WE}$; or

C_H = 50 pF loads on all outputs except
C_H = 125 pF loads on $\overline{RAS}$0–3 and $\overline{CAS}$0–3 and
C_H = 380 pF loads on Q0–9 and $\overline{WE}$.

Number	Symbol	Common Parameter Description	NS32CG821-20				NS32CG821-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
1	fCLK	CLK Frequency	0	20	0	20	0	25	0	25
2	tCLKP	CLK Period	50		50		40		40	
3, 4	tCLKPW	CLK Pulse Width	15		15		12		12	
5	fDCLK	DELCLK Frequency	5	20	5	20	5	20	5	20
6	tDCLKP	DELCLK Period	50	200	50	200	50	200	50	200
7, 8	tDCLKPW	DELCLK Pulse Width	15		15		12		12	
9a	tPRASCAS0	$\overline{RAS}$ Asserted to $\overline{CAS}$ Asserted (tRAH = 15 ns, tASC = 0 ns)	30		30		30		30	
9b	tPRASCAS1	$\overline{RAS}$ Asserted to $\overline{CAS}$ Asserted (tRAH = 15 ns, tASC = 10 ns)	40		40		40		40	
9c	tPRASCAS2	($\overline{RAS}$ Asserted to $\overline{CAS}$ Asserted (tRAH = 25 ns, tASC = 0 ns)	40		40		40		40	
9d	tPRASCAS3	($\overline{RAS}$ Asserted to $\overline{CAS}$ Asserted (tRAH = 25 ns, tASC = 10 ns)	50		50		50		50	
10a	tRAH	Row Address Hold Time (tRAH = 15)	17		15		17		15	
10b	tRAH	Row Address Hold Time (tRAH = 25)	27		25		27		25	
11a	tASC	Column Address Setup Time (tASC = 0)	2		0		2		0	
11b	tASC	Column Address Setup Time (tASC = 10)	12		10		12		10	
12	tPCKRAS	CLK High to $\overline{RAS}$ Asserted following Precharge		27		32		22		26
13	tPARQRAS	$\overline{TSO}$ Negated to $\overline{RAS}$ Negated		38		43		31		35
14	tPENCL	$\overline{ECAS}$0–3 Asserted to $\overline{CAS}$ Asserted		23		31		20		27
15	tPENCH	$\overline{ECAS}$0–3 Negated to $\overline{CAS}$ Negated		25		33		20		27
16	tPARQCAS	$\overline{TSO}$ Negated to $\overline{CAS}$ Negated		60		68		47		54
17	tPCLKWH	CLK to $\overline{WAIT}$ Negated		39		39		31		31
19	tPEWL	$\overline{ECAS}$ Negated to $\overline{WAIT}$ Asserted during a Burst Access		42		42		34		34
20	tSECK	$\overline{ECAS}$ Asserted Setup to CLK High to Recognize the Rising Edge of CLK during a Burst Access	24		24		19		19	
23	tSWCK	$\overline{WAITIN}$ Asserted Setup to CLK	5		5		5		5	
24	tPWINWEH	$\overline{WIN}$ Asserted to $\overline{WE}$ Asserted		39		49		31		41
25	tPWINWEL	$\overline{WIN}$ Negated to $\overline{WE}$ Negated		39		49		31		41
26	tPAQ	Row, Column Address Valid to Q0–9 Valid		29		38		26		35
27	tPCINCQ	COLINC Asserted to Q0–9 Incremented		34		43		30		39
28	tSCINEN	COLINC Asserted Setup to $\overline{ECAS}$ Asserted to Ensure tASC = 0 ns	16		17		15		17	

12.0 AC Timing Parameters: NS32CG821 (Continued)

Unless otherwise stated V_{CC} = 5.0V ±10%, 0°C < T_A < 70°C, the output load capacitance is typical for 4 banks of 18 DRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:
C_L = 50 pF loads on all outputs except
C_L = 150 pF loads on Q0–9 and $\overline{WE}$; or
C_H = 50 pF loads on all outputs except
C_H = 125 pF loads on $\overline{RAS}$0–3 and $\overline{CAS}$0–3 and
C_H = 380 pF loads on Q0–9 and $\overline{WE}$.

Number	Symbol	Common Parameter Description	NS32CG821-20				NS32CG821-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
29a	tSARQCK1	$\overline{TSO}$ Negated Setup to CLK High with 1 Period of Precharge	43		43		34		34	
29b	tSARQCK2	$\overline{TSO}$ Negated Setup to CLK High with >1 Period of Precharge Programmed	19		19		15		15	
31	tPCKCAS	CLK High to $\overline{CAS}$ Asserted when Delayed by $\overline{WIN}$		31		39		25		32
32	tSCADEN	Column Address Setup to $\overline{ECAS}$ Asserted to Guarantee tASC = 0	14		15		14		16	
33	tWCINC	COLINC Pulse Width	20		20		20		20	
34a	tPCKCL0	CLK High to $\overline{CAS}$ Asserted following Precharge (tRAH = 15 ns, tASC = 0 ns)		81		89		72		79
34b	tPCKCL1	CLK High to $\overline{CAS}$ Asserted following Precharge (tRAH = 15 ns, tASC = 10 ns)		91		99		82		89
34c	tPCKCL2	CLK High to $\overline{CAS}$ Asserted following Precharge (tRAH = 25 ns, tASC = 0 ns)		91		99		82		89
34d	tPCKCL3	CLK High to $\overline{CAS}$ Asserted following Precharge (tRAH = 25 ns, tASC = 10 ns)		101		109		92		99
35	tCAH	Column Address Hold Time (Interleave Mode Only)	32		32		32		32	
36	tPCQR	$\overline{CAS}$ Asserted to Row Address Valid (Interleave Mode Only)		90		90		90		90

12.0 AC Timing Parameters: NS32CG821 (Continued)

Unless otherwise stated V_{CC} = 5.0V ±10%, 0°C < T_A < 70°C, the output load capacitance is typical for 4 banks of 18 DRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:
C_L = 50 pF loads on all outputs except
C_L = 150 pF loads on Q0–9 and $\overline{WE}$; or
C_H = 50 pF loads on all outputs except
C_H = 125 pF loads on $\overline{RAS}$0–3 and $\overline{CAS}$0–3 and
C_H = 380 pF loads on Q0–9 and $\overline{WE}$.

Number	Symbol	Difference Parameter Description	NS32CG821-20				NS32CG821-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
50	tD1	($\overline{TSO}$ Negated to $\overline{RAS}$ Negated) Minus (CLK High to $\overline{RAS}$ Asserted)		16		16		14		14
51	tD2	(CLK High to Refresh $\overline{RAS}$ Negated) Minus (CLK High to $\overline{RAS}$ Asserted)		13		13		11		11
53	tD3b	(CLK High to $\overline{RAS}$ Asserted Minus ($\overline{TSO}$ Negated to $\overline{RAS}$ Negated)		4		4		4		4
54	tD4	($\overline{ECAS}$ Asserted to $\overline{CAS}$ Asserted) Minus ($\overline{ECAS}$ Negated to $\overline{CAS}$ Negated)	−7	7	−7	7	−7	7	−7	7
55	tD5	(CLK to Refresh $\overline{RAS}$ Asserted) Minus (CLK to Refresh $\overline{RAS}$ Negated)		5		5		5		5

Unless otherwise stated V_{CC} = 5.0V ±10%, 0°C < T_A < 70°C, the output load capacitance is typical for 4 banks of 18 DRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:
C_L = 50 pF loads on all outputs except
C_L = 150 pF loads on Q0–9 and $\overline{WE}$; or
C_H = 50 pF loads on all outputs except
C_H = 125 pF loads on $\overline{RAS}$0–3 and $\overline{CAS}$0–3 and
C_H = 380 pF loads on Q0–9 and $\overline{WE}$.

Number	Symbol	Refresh Parameter Description	NS32CG821-20				NS32CG821-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
207	tPCKRFRASH	CLK High to Refresh $\overline{RAS}$ Negated		35		40		29		33
208	tPCKRFRASL	CLK High to Refresh $\overline{RAS}$ Asserted		28		33		23		27

12.0 AC Timing Parameters: NS32CG821 (Continued)

Unless otherwise stated V_{CC} = 5.0V ±10%, 0°C < T_A < 70°C, the output load capacitance is typical for 4 banks of 18 DRAMs per bank, including trace capacitance (see Note 2).

Two different loads are specified:
C_L = 50 pF loads on all outputs except
C_L = 150 pF loads on Q0–9 and $\overline{WE}$; or
C_H = 50 pF loads on all outputs except
C_H = 125 pF loads on $\overline{RAS}$0–3 and $\overline{CAS}$0–3 and
C_H = 380 pF loads on Q0–9 and $\overline{WE}$.

Number	Symbol	Memory Access Parameter Description	NS32CG821-20				NS32CG821-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
300	tSCSCK	$\overline{CS}$ Asserted to CLK High	14		14		13		13	
301a	tSALECKNL	ALE Asserted Setup to CLK High Not Using On-Chip Latches or if Using On-Chip Latches and B0, B1, Are Constant, Only 1 Bank	16		16		15		15	
301b	tSALECKL	ALE Asserted Setup to CLK High, if Using On-Chip Latches if B0, B1 Can Change, More Than One Bank	29		29		29		29	
302	tWALE	ALE Pulse Width	18		18		13		13	
303	tSBADDCK	Bank Address Valid Setup to CLK High	20		20		18		18	
304	tSADDCK	Row, Column Valid Setup to CLK High to Guarantee tASR = 0 ns	11		15		11		16	
305	tHASRCB	Row, Column, Bank Address Held from ALE Negated (Using On-Chip Latches)	10		10		8		8	
306	tSRCBAS	Row, Column, Bank Address Setup to ALE Negated (Using On-Chip Latches)	3		3		2		2	
307	tPCKRL	CLK High to $\overline{RAS}$ Asserted		27		32		22		26
308a	tPCKCL0	CLK High to $\overline{CAS}$ Asserted (tRAH = 15 ns, tASC = 0 ns)		81		89		72		79
308b	tPCKCL1	CLK High to $\overline{CAS}$ Asserted (tRAH = 15 ns, tASC = 10 ns)		91		99		82		89
308c	tPCKCL2	CLK High to $\overline{CAS}$ Asserted (tRAH = 25 ns, tASC = 0 ns)		91		99		82		89
308d	tPCKCL3	CLK High to $\overline{CAS}$ Asserted (tRAH = 25 ns, tASC = 10 ns)		101		109		92		99
309	tHCKALE	ALE Negated Hold from CLK High	0		0		0		0	
310	tSWINCK	$\overline{WIN}$ Asserted Setup to CLK High that starts access $\overline{RAS}$ to Guarantee $\overline{CAS}$ is Delayed	−21		−21		−16		−16	
311	tPCSWL	$\overline{CS}$ Asserted to $\overline{WAIT}$ Asserted		26		26		22		22
312	tPCSWH	$\overline{CS}$ Negated to $\overline{WAIT}$ Negated		26		26		22		22
314	tPALEWL	ALE Asserted to $\overline{WAIT}$ Asserted ($\overline{CS}$ is Already Asserted)		48		48		39		39
315		$\overline{TSO}$ Negated to CLK High That Starts Access $\overline{RAS}$ to Guarantee tASR = 0 ns	41		45		34		39	
316	$t_{tPCKCV0}$	CLK to Column Addr. Valid (t_{RAH} = 15 ns, t_{ASC} = 0 ns)		78		87		66		75

12.0 AC Timing Parameters: NS32CG821 (Continued)

Unless otherwise stated V_{CC} = 5.0V ±10%, 0°C < T_A < 70°C, the output load capacitance is typical for 4 banks of 18 DRAMs per bank, including trace capacitance (see Note 2).

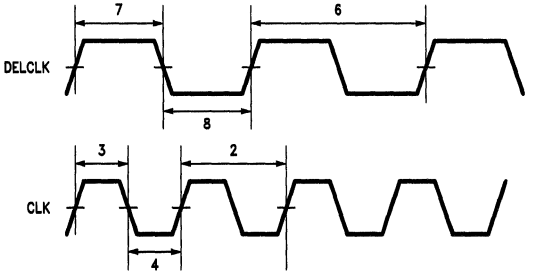
Two different loads are specified:
C_L = 50 pF loads on all outputs except
C_L = 150 pF loads on Q0–9 and $\overline{WE}$; or
C_H = 50 pF loads on all outputs except
C_H = 125 pF loads on $\overline{RAS}$0–3 and $\overline{CAS}$0–3 and
C_H = 380 pF loads on Q0–9 and $\overline{WE}$.

Number	Symbol	Programming Parameter Description	NS32CG821-20				NS32CG821-25			
			C_L		C_H		C_L		C_H	
			Min	Max	Min	Max	Min	Max	Min	Max
500	tHMLADD	Mode Address Held from $\overline{ML}$ Negated	6		6		5		5	
501	tSADDML	Mode Address Setup to $\overline{ML}$ Negated	6		6		6		6	
502	tWML	$\overline{ML}$ Pulse Width	15		15		15		15	
503	tSADAQML	Mode Address Setup to $\overline{TSO}$ Asserted	0		0		0		0	
504	tHADAQML	Mode Address Held from $\overline{TSO}$ Asserted	39		39		29		29	
505	tSCSARQ	$\overline{CS}$ Asserted Setup to $\overline{TSO}$ Asserted	6		6		6		6	
506	tSMLARQ	$\overline{ML}$ Asserted Setup to $\overline{TSO}$ Asserted	10		10		10		10	

Note 1: "Absolute Maximum Ratings" are the values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

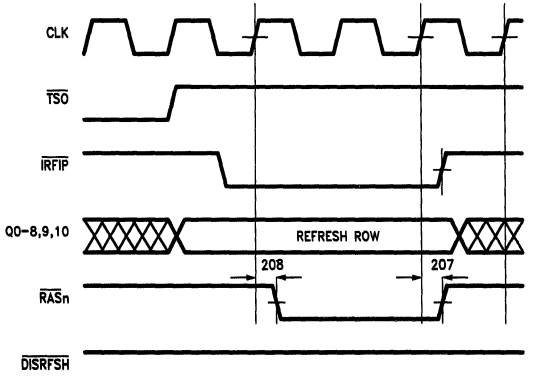
Note 2: Input pulse 0V to 3V; tR = tF = 2.5 ns. Input reference point on AC measurements is 1.5V. Output reference points are 2.4V for High and 0.8V for Low.

Note 3: AC Production testing is done at 50 pF.



TL/F/10126–42

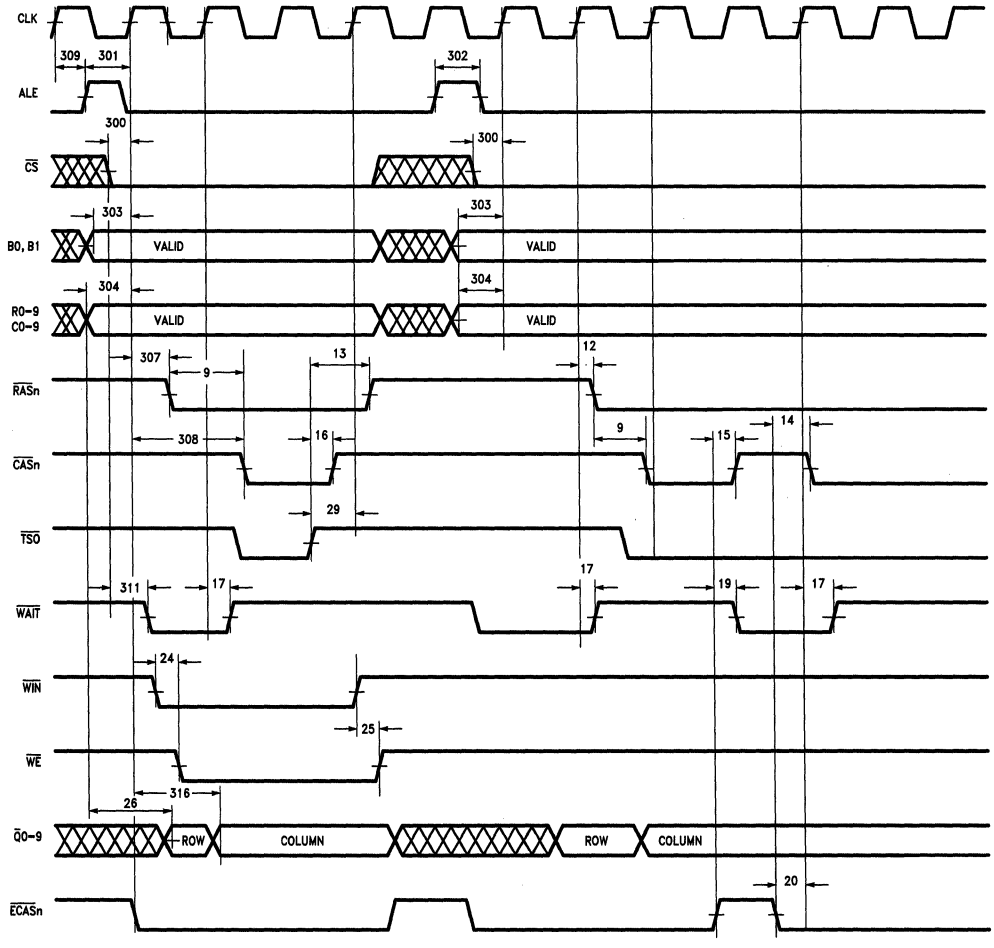
FIGURE 42. Clock, DELCLK Timing



TL/F/10126–43

FIGURE 43. 200: Refresh Timing

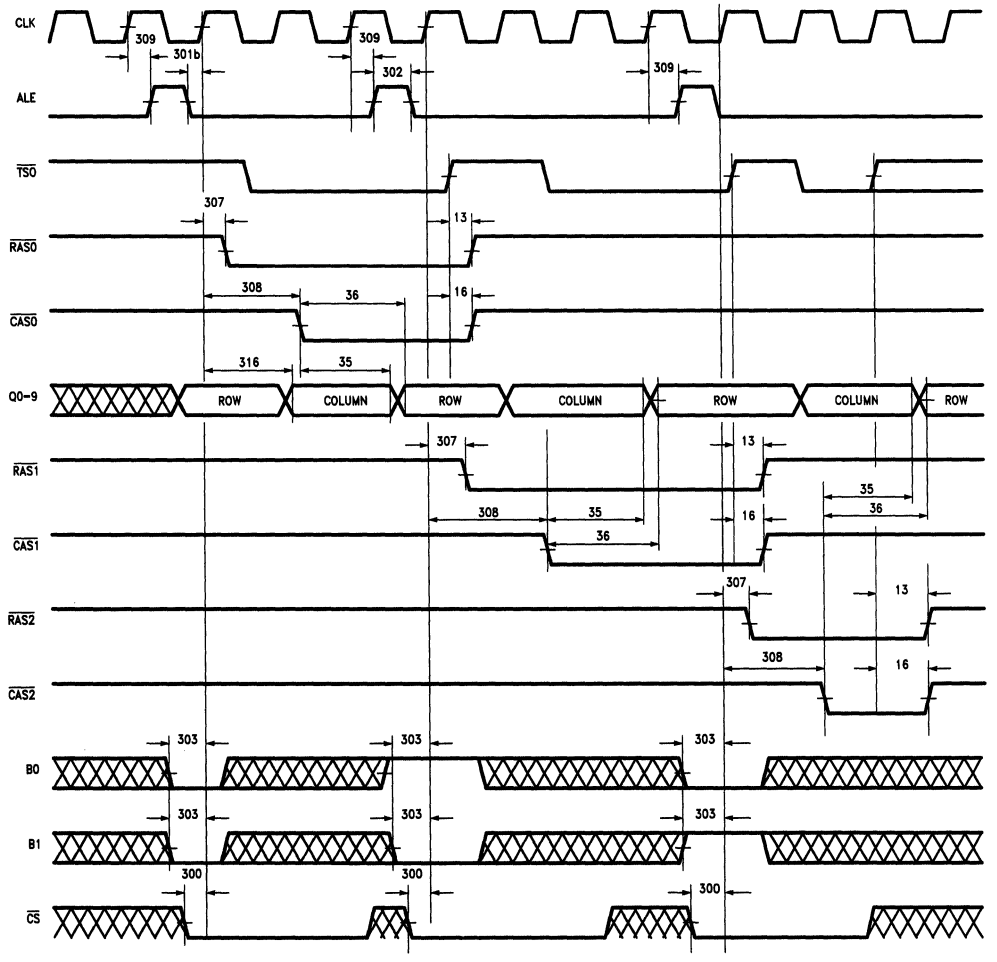
12.0 AC Timing Parameters: NS32CG821 (Continued)



TL/F/10126–44

FIGURE 44. 300: Memory Access Timing

12.0 AC Timing Parameters: NS32CG821 (Continued)

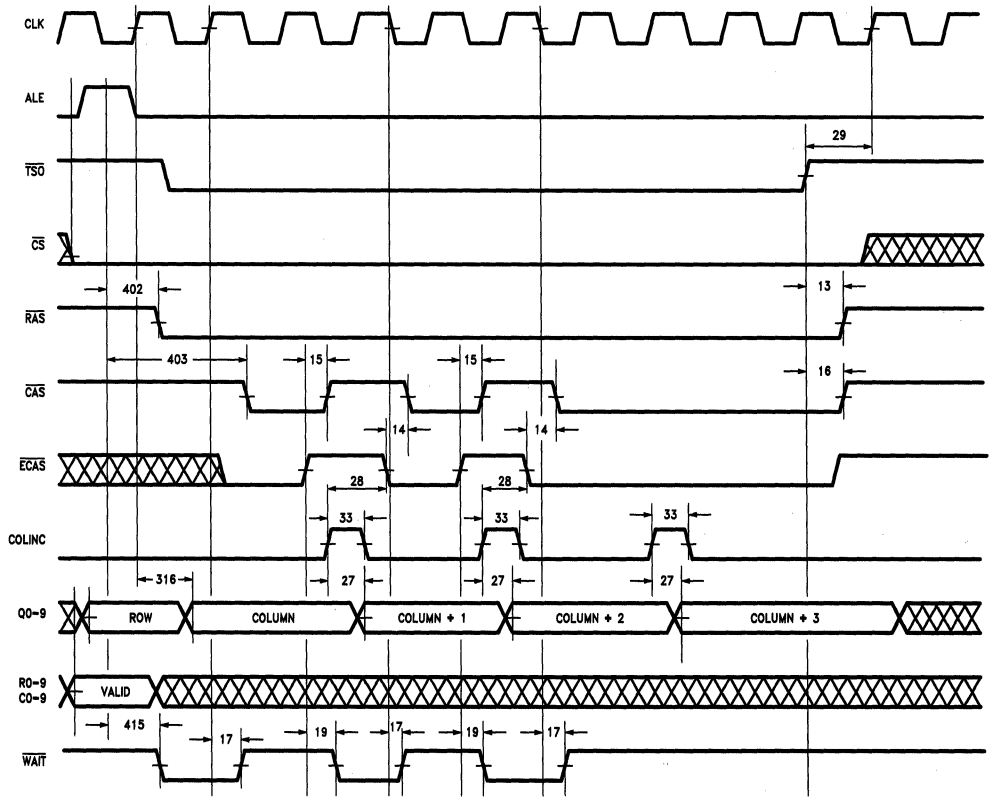


TL/F/10126–45

(Programmed as C4 = 1, C5 = 1, C6 = 1)

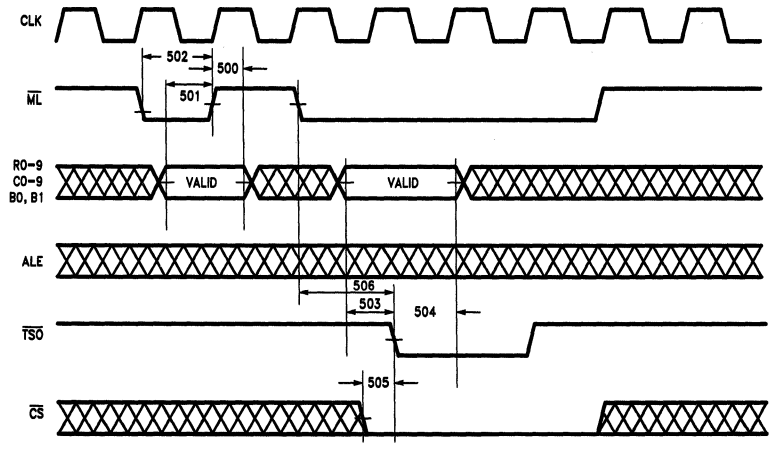
FIGURE 45. 300: Access Interleaving

12.0 AC Timing Parameters: NS32CG821 (Continued)



TL/F/10126–46

FIGURE 46. 400: COLINC Page/Static Column Access Timing



TL/F/10126–47

FIGURE 47. 500: Programming

13.0 NS32CG821 User Hints

1. All inputs to the NS32CG821 should be tied high, low or the output of some other device.

   Note: One signal is active high. COLINC should be tied low to disable.

2. Each ground on the NS32CG821 must be decoupled to the closest on-chip supply (V_{CC}) with 0.1 μF ceramic capacitor. This is necessary because these grounds are kept separate inside the NS32CG821. The decoupling capacitors should be placed as close as possible with short leads to the ground and supply pins of the NS32CG821.
3. The output called "CAP" should have a 0.1 μF capacitor to ground.
4. The NS32CG821 has 20Ω series damping resistors built into the output drivers of $\overline{RAS}$, $\overline{CAS}$, address and $\overline{WE}$. Space should be provided for external damping resistors on the printed circuit board (or wire-wrap board) because they may be needed. The value of these damping resistors (if needed) will vary depending upon the output, the capacitance of the load, and the characteristics of the trace as well as the routing of the trace. The value of the damping resistor also may vary between the wire-wrap board and the printed circuit board. To determine the value of the series damping resistor it is recommended to use an oscilloscope and look at the furthest DRAM from the NS32CG821. The undershoot of $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$ and the addresses should be kept to less than 0.5V below ground by varying the value of the damping resistor. The damping resistors should be placed as close as possible with short leads to the driver outputs of the NS32CG821.
5. The circuit board must have a good V_{CC} and ground plane connection. If the board is wire-wrapped, the V_{CC} and ground pins of the NS32CG821, the DRAM associated logic and buffer circuitry must be soldered to the V_{CC} and ground planes.
6. The traces from the NS32CG821 to the DRAM should be as short as possible.
7. PARAMETER CHANGES DUE TO LOADING

   All A.C. parameters are specified with the equivalent load capacitances, including traces, of 64 DRAMs organized as 4 banks of 18 DRAMs each. Maximums are based on worst-case conditions. If an output load changes then the A.C. timing parameters associated with that particular output must be changed. For example, if we changed our output load to

   C = 250 pF loads on $\overline{RAS}$0–3 and $\overline{CAS}$0–3

   C = 760 pF loads on Q0–9 and $\overline{WE}$

   we would have to modify some parameters (not all calculated here)

   $308a Clock to $\overline{CAS}$ asserted

   (t_{RAH} = 15 ns, t_{ASC} = 0 ns)

   A ratio can be used to figure out the timing change per change in capacitance for a particular parameter by using the specifications and capacitances from heavy and light load timing.

$$\text{Ratio} = \frac{\$308a\ \text{w/heavy load} - \$308a\ \text{w/light load}}{C_H(\overline{CAS}) - C_L(\overline{CAS})}$$

$$= \frac{79\ \text{ns} - 72\ \text{ns}}{125\ \text{pF} - 50\ \text{pF}} = \frac{7\ \text{ns}}{75\ \text{pF}}$$

$$\$308a\ (\text{actual}) = (\text{capacitance difference} \times \text{ratio}) + \$308a\ (\text{specified})$$

$$= \left(250\ \text{pF} - 125\ \text{pF}\right)\frac{7\ \text{ns}}{75\ \text{pF}} + 79\ \text{ns}$$

$$= 11.7\ \text{ns} + 79\ \text{ns}$$

$$= 90.7\ \text{ns @ 250 pF load}$$

National Semiconductor

PRELIMINARY

DP8520A/DP8521A/DP8522A microCMOS Programmable 256k/1M/4M Video RAM Controller/Drivers

General Description

The DP8520A/21A/22A video RAM controllers provide a low cost, single chip interface between video RAM and all 8-, 16- and 32-bit systems. The DP8520A/21A/22A generate all the required access control signal timing for VRAMs. An on-chip refresh request clock is used to automatically refresh the VRAM array. Refreshes and accesses are arbitrated on chip. If necessary, a $\overline{WAIT}$ or $\overline{DTACK}$ output inserts wait states into system access cycles, including burst mode accesses. $\overline{RAS}$ low time during refreshes and $\overline{RAS}$ precharge time after refreshes and back to back accesses are guaranteed through the insertion of wait states. Separate on-chip precharge counters for each $\overline{RAS}$ output can be used for memory interleaving to avoid delayed back to back accesses because of precharge. An additional feature of the DP8522A is two access ports to simplify dual accessing. Arbitration among these ports and refresh is done on chip.

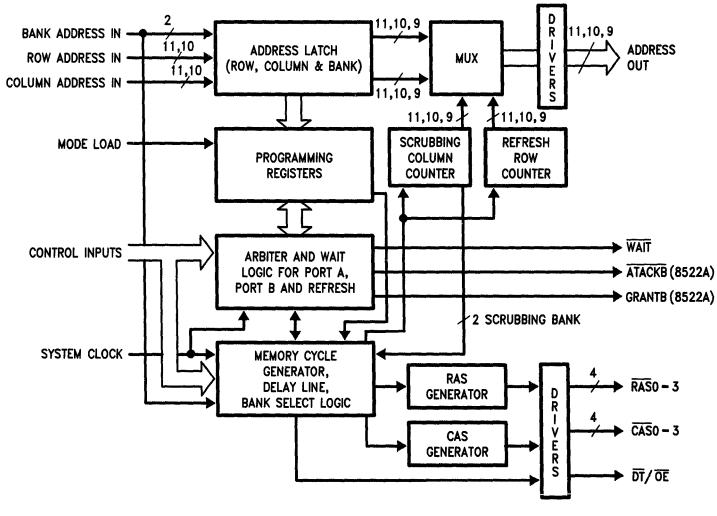
Features

- On chip high precision delay line to guarantee critical VRAM access timing parameters
- microCMOS process for low power
- High capacitance drivers for $\overline{RAS}$, $\overline{CAS}$, $\overline{DT}/\overline{OE}$ and VRAM address on chip
- On chip support for nibble, page and static column VRAMs
- Byte enable signals on chip allow byte writing in a word size up to 16 bits with no external logic
- Selection of controller speeds: 20 MHz and 25 MHz
- On board Port A/Port B (DP8522A only)/refresh arbitration logic
- Direct interface to all major microprocessors (application notes available)
- 4 $\overline{RAS}$ and 4 $\overline{CAS}$ drivers (the $\overline{RAS}$ and $\overline{CAS}$ configuration is programmable)

Control	# of Pins (PLCC)	# of Address Outputs	Largest VRAM Possible	Direct Drive Memory Capacity	Access Ports Available
DP8520A	68	9	256 kbit	4 Mbytes	Single Access Port
DP8521A	68	10	1 Mbit	16 Mbytes	Single Access Port
DP8522A	84	11	4 Mbit	64 Mbytes	Dual Access Ports (A and B)

Block Diagram

DP8520A/21A/22A VRAM Controller



TL/F/9338–5

FIGURE 1

National Semiconductor

29F68
Dynamic RAM Controller

General Description

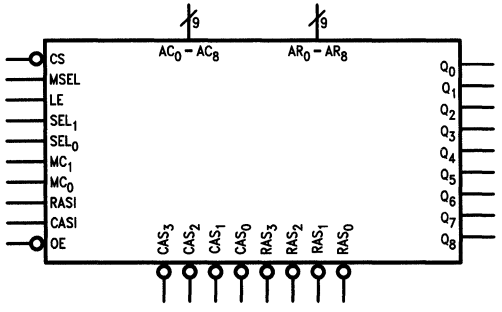
The 29F68 is a high-performance memory controller, replacing many SSI and MSI devices by grouping several unique functions. It provides two 9-bit address latches and two 9-bit counters for row and column address generation during refresh. A 2-bit bank select latch for row and column address generation during refresh, and a 2-bit bank select latch for the two high order address bits are provided to select one of the four RAS and CAS outputs.

The 29F68 is functionally equivalent to AMD's Am2968 and Motorola's MC74F2968.

Features

- High-performance memory controller
- Replaces many SSI and MSI devices by grouping several unique functions
- Functionally equivalent to AMD's Am2968 and Motorola's MC74F2968
- Provides control for 16K, 64K, or 256K dynamic RAM systems
- Outputs directly drive up to 88 DRAMs
- Highest order two address bits select one of four banks of RAMs
- Chip Select for easy expansion
- Provides memory refresh with error correction mode

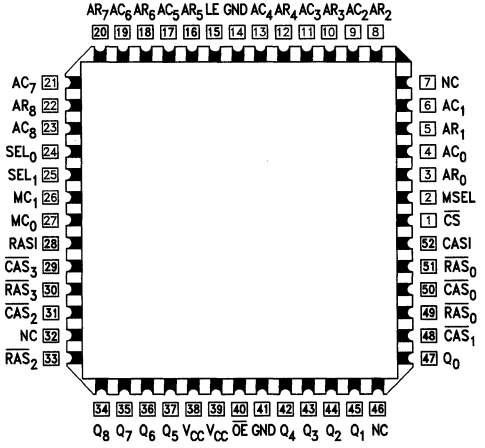
Logic Symbol



TL/F/9608–1

Connection Diagram

Pin Assignment for LCC and PCC



TL/F/9608–2

National Semiconductor

54F/74F968
1 Mbit Dynamic RAM Controller

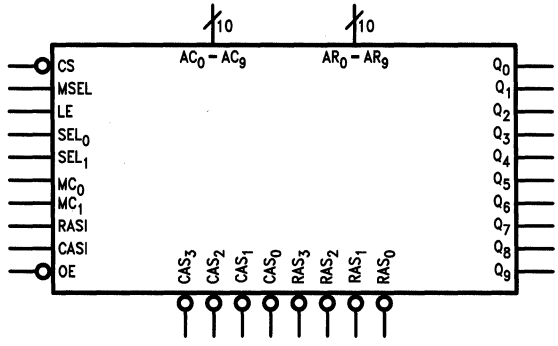
General Description

The 'F968 is a high performance memory controller, replacing many SSI and MSI devices by grouping several unique functions. It provides two 10-bit address latches and two 10-bit counters for row and column address generation during refresh. A 2-bit bank select latch for row and column address generation during refresh and a 2-bit bank select latch for the two high order address bits are provided to select one of the four RAS and CAS outputs.

Features

- Provides control for 16K, 64K, 256K or 1 Mbit DRAM systems
- Outputs directly drive up to 88 DRAMs
- Chip select for easy expansion
- Provides memory refresh with error correction mode
- 52-pin plastic leaded chip carrier

Logic Symbol



TL/F/9604–1

Precautions to Take When Driving Memories

National Semiconductor
Application Note 305
Mike Evans

As memory prices continue their relentless reduction of cost per bit, more and more systems designers are incorporating memories into their designs. In general these memories comprise a number of dynamic RAMs, such as the 64k x 1. In this x 1 configuration, the number of RAMs required is a multiple of the bus width. Most new system designs use 16-bit microprocessors, so that a typical memory will comprise from 16 to 64 DRAMs, thus providing from 64k to 256k addressing capability. This means the memory drivers have to drive upwards of 16 RAMs. The drivers may be part of an integrated circuit dynamic RAM controller such as the DP8408A/DP8409A, or they may be on a separate chip such as the DP84240/DP84244 octal memory drivers. The recommendations in this article are valid for any type of memory driver. The purpose of the article is to forewarn new designers using memories of problems they will encounter if adequate precautions are not taken.

A typical configuration of a 16-bit wide memory is shown in *Figure 1*. Each driver address output goes to every dynamic RAM, as does $\overline{WE}$. $\overline{CAS}$ outputs go to half the number of RAMs assuming byte writing is required. $\overline{RAS}$ outputs each go only to one bank. Note that these loads are not true for the data inputs and outputs. Each data I/O only connects to its respective bit, so the loading is only one RAM per bank for data. In general, this is why buffers are not required on the data bus when interfacing to memory. Data In of the RAMs can be linked directly to Data Out for any one bit, and also to the corresponding bit on the data bus. This is true for normal read and write operations, but if read-modify-write cycles are employed, the Data Out signals must be buffered from the data bus.

Using this typical memory configuration may not be as simple as it seems. Without care and attention, problems can arise for the unprepared, and there are two areas in particular which may cause memory errors or memory damage: one is voltage overshoot caused by inductive traces and high capacitive loads, the other is switching spikes caused by switching high capacitive loads.

OVERSHOOT AND UNDERSHOOT

(Undershoot is Negative Overshoot)

When a system requires a number of dynamic RAMs, the result is high capacitance loads, caused by a combination of RAM input capacitance and trace capacitance. Each dynamic RAM has a specified input capacitance of 10 pF maximum, but most dynamic RAMs are closer to 2 to 3 pF. Very few actually get close to 10 pF, even under worst case conditions of high temperature and V_{CC}. It is safe, therefore, to assume a much lower average input capacitance when using 16 or more RAMs.

In fact, the input capacitance of most inputs is due more to the package than the input gating, because the silicon gate inputs of the transistors in today's market have such high impedance. A typical maximum would be 2.5 pF. Control inputs such as $\overline{RAS}$ and $\overline{CAS}$ connect to more than one transistor input. For example, on the National Semiconductor 64k x 1 dynamic RAM, the NMC4164, $\overline{RAS}$ goes to two transistors and $\overline{CAS}$ to four. In general, this is true for most manufacturers' RAMs, so a more typical maximum input capacitance would be 3 pF for $\overline{RAS}$ and 3.5 pF for $\overline{CAS}$. RAM input currents are so small as to be negligible. The input current is quoted as 10 μA maximum, but again most RAMs are much less than this in a typical memory. Driving DRAMs, therefore, is not a problem of DC drive capability, but rather a problem of capacitance drive capability.

Driving DRAM input capacitance is further compounded by printed circuit traces, and even more so by wire-wrapping. Both can be represented by a transmission line with distributed capacitance and inductance. Thus, the total load is equivalent to a complex impedance comprising the distributed trace inductance, and a capacitance comprising distributed trace capacitance and RAM input capacitance as shown in *Figure 2a*.

The effect is an overshoot or undershoot at the dynamic RAM inputs that occurs each time a memory driver changes state, as shown in *Figure 2b*. As the driver output changes state, the load capacitance cannot be instantaneously charged or discharged because the current available is limited both by the driver transistor impedance, and the equivalent series resistance from the supply rail through the chip to the trace resistance. This current will be similar in value to the quoted short circuit current of the driver stage; therefore there is a spike of current that lasts as long as it takes to change the voltage of all the capacitances. For the driver stages of the DP8408A/DP8409A, or the DP84240/DP84244, the typical short circuit current is 100 mA per stage. This is true for either direction, so that the high-to-low transition takes roughly the same time as the low-to-high transition, minimizing skew times on all the driver outputs, as they transition in either direction. Assuming the output low voltage, V_{OL}, is 0.2V and the output high voltage, V_{OH}, is 3.2V, and that the charge/discharge current is constant at I_{SC}, then the current spike will exist for a time, T, where,

$$T = C_{IL} \times (V_{OH} - V_{OL})/I_{SC}$$
$$= 500\ pF \times 3.0V/100\ mA = 15\ ns$$

C_L (500 pF) is the load capacitance of typically 64 to 88 dynamic RAMs, in other words, four banks comprising 16 data bits and possibly six check bits if error correction is required.

In fact, due to the trace inductance, the rate of change of current will not be a step function, so that the current waveform looks like a spike. Even so, the rapid rate of change of current, di/dt, into the trace inductance L, will create a potentially excessive voltage "e" across this inductance. As an example, if the current changes from 0 to 100 mA in 6 ns, and the composite trace inductance is 0.3 μH, then the voltage across this inductance is "e,"where,

$$e = L\ di/dt$$
$$= 0.3\ \mu H \times 100\ mA/6\ ns = 5V$$

In other words, at this rate of change in current, even a small inductance can be dangerous for two reasons. First, the dynamic RAMs at the far end of the trace could be destroyed, unless they have clamping diodes to V_{CC} and GND (most do not), or second, the returning voltage may exceed the threshold it has just passed causing a second

and then third change of state. If this sudden glitch occurs on a control signal input such as $\overline{RAS}$, the memory contents may be inadvertently changed.

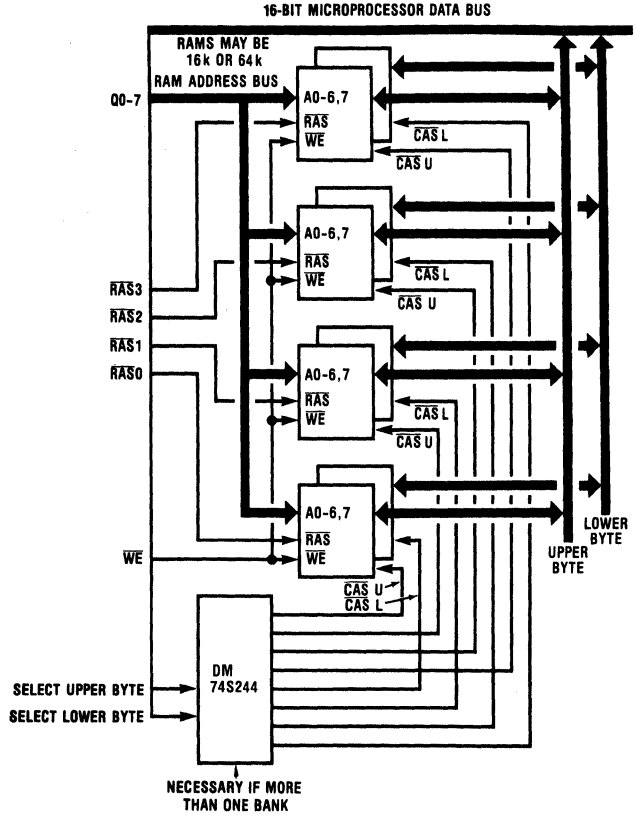
It is therefore necessary to remove the spike. The most common approach is to insert a damping resistor in the path between the driver and the RAMs, fairly close to the driver, as shown by R_D in *Figure 2a*. The best value for the resistor is the critical value giving a critically damped transition. Too high a value will cause overdamping which results in a slow transition. This slow edge may create excessive skew problems and slow down the memory cycle, or even worse, the edge may be slow enough that the RAM cycle never begins internally. If the damping resistor value is too low, the undershoot or overshoot may not be removed. It is therefore recommended that the resistor be determined on the first prototypes (not wire-wrapped prototypes because the value will be different due to the larger distributed inductance and capacitance). Also, the values may be different for the control lines, particularly $\overline{CAS}$. If there are a number of banks, and a $\overline{RAS}$ is used to select each bank, then the damping resistor in this line will be higher.

Typical values for the damping resistors will be between 15Ω and 100Ω, the lower the loading, the higher the values.

Some IC manufacturers offer octal memory drivers with on-chip series resistors fixed at ≈25Ω. Unless this is the critical value required for all the lines, problems will arise. The DP8400 family has been designed with equivalent internal values of approximately 10Ω, allowing for any external value of damping resistor.

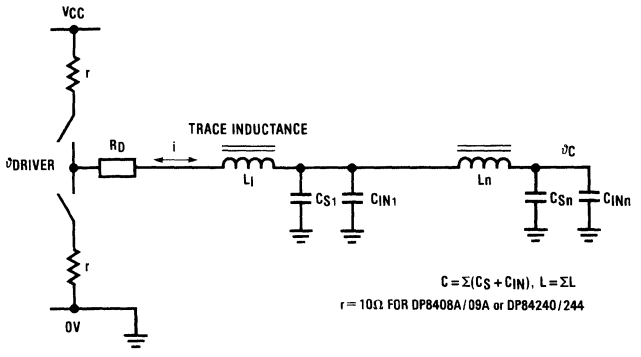
SWITCHING CURRENT SPIKES

Another major undesirable effect of the fast current spikes is the effect on the V_{CC} and GND pins. The worst case is when all eight or nine address outputs switch in the same direction at the same time, as shown in *Figure 3a*. If each driver can source or sink 100 mA, then a current of approximately 1A could enter or exit the driver chip in a period of 20 ns. The resistance and inductance of the V_{CC} and GND lines to the chip can cause excessive drops during this switching time (see waveforms in *Figure 3a*), which may, in turn, upset latches either in the DP8408A/DP8409A, or externally. A ceramic capacitor connected across V_{CC} and GND pins will largely remove the spike. A 1 μF multilayer ceramic is recommended. This should be fitted as close as possible to the pins in order to reduce lead inductance. The DP8408A/DP8409A pin configuration facilitates this with



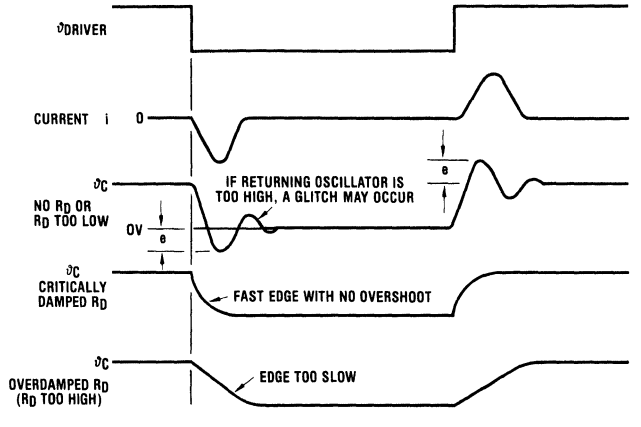
TL/F/5031–1

FIGURE 1. Typical 16-Bit Memory with Byte Write Address



TL/F/5031–2

FIGURE 2a. Complex Load Impedance Caused by Distributed Trace Inductance L and Capacitance C_S, and RAM Input Capacitance C_{IN}



TL/F/5031–3

FIGURE 2b. Timing Waveforms Showing the Effect of Variations of R_D on Signals Appearing at the RAM

GND and V_{CC} pins 0.2" apart so that the ceramic capacitor can be fitted as close to the chip as possible. The second GND pin should also be decoupled. These GND and V_{CC} pins are located in the center of the package to reduce bonding lead lengths. In fact, the lead resistance is five times lower than if the supply pins were in the corners. An example of how this spike can be reduced would be the previous example of a 1A change in supply current switching in 20 ns with a 1 μF ceramic capacitor decoupling GND and V_{CC}. The voltage drop "v" is 1A×20 ns/1 μF, or 20 mV.

If the decoupling capacitor was 0.01 μF, the drop would be 2V. Tantalum or other types of capacitors are lower frequency capacitors and have only a small effect in reducing the voltage spike. Ceramic capacitors are high frequency, and multilayer capacitors with lower inductance have a greater effect in reducing the voltage spike and are therefore recommended. As a further recommendation, the dynamic RAMs should be similarly decoupled with approximately a 0.1 μF ceramic capacitor on each RAM. Wire-wrapped boards, in particular, need special attention.

There are some other precautions that may be considered when driving memories. First, be aware that IC sockets increase load capacitance and inductance, so it becomes a matter of the importance of removability of chips, and maintainability. Also, shorter, thicker trace lengths will reduce the load, and good GND and V_{CC} connections will help reduce the voltage spikes around the memory board. For wire-wrapped designs, GND and V_{CC} should be multiwired.

With proper decoupling and correct selection of damping resistors, integrated circuit dynamic RAM controllers will function as expected to ease the burden of the system designer.

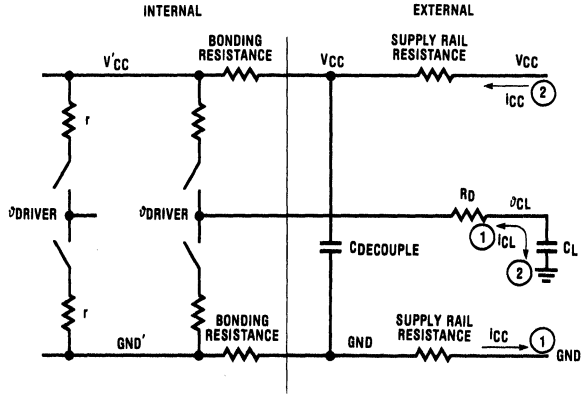


FIGURE 3a. Effect of Switching All Outputs Simultaneously in the Same Direction

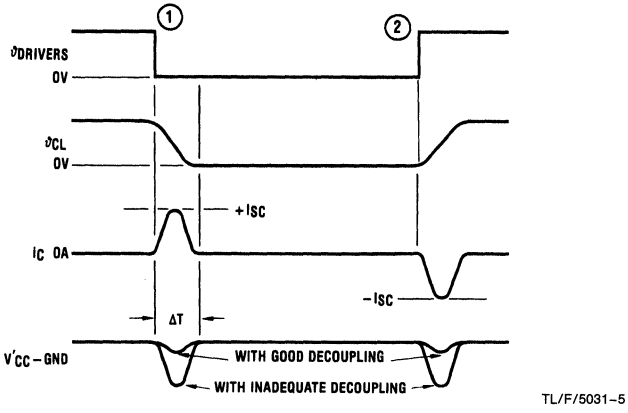


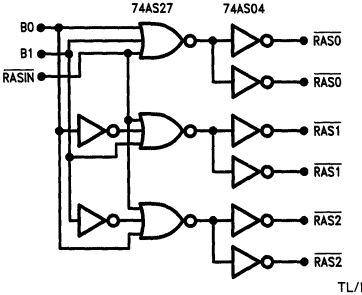
FIGURE 3b. Timing Waveforms Showing Internal Supply Rail Drops During Output Switching

DP8408A/09A/17/18/19/ 28/29 Application Hints

National Semiconductor
Application Brief #1
Tim Garverick
Webster Meier

The DP8408A, DP8409A dynamic RAM controllers have been well received by dynamic memory users because they perform functions formerly requiring multiple integrated circuit chips. These controllers are designed to be suitable for a variety of DRAM control methods. As a result of the many combinations of ways in which inputs to these chips may be varied, it was inevitable that certain conditions exist that would cause the DP8408A, DP8409A to respond in an undesirable way. Feedback from customers using these chips has resulted in thorough investigations of such conditions. The following are constraints on the use of the DRAM controllers which are not addressed in their data sheets. The majority of customers will find that most of the items on this list are not pertinent to their particular application, and those that are impose minimal restrictions.

1) The on-chip refresh counter resets when the RFI/O pin goes low for a refresh request in mode 5 if this pin is excessively loaded with capacitance. The data sheet suggests that this pin not be loaded with greater than 50 pF. Since RFI/O, in most cases, needs only to drive a low capacitance in a refresh control circuit, this limit is not unreasonable.
2) When the DP8408A, DP8409A is in a refresh mode, the RFI/O pin indicates that the on-chip refresh counter has reached its end-of-count. This end-of-count is selectable as 127, 255 or 511 (511 is available only on the DP8409A) to accommodate 16k, 64k or 256k DRAMS, respectively. Although the end-of-count may be chosen to be any of these, the counter always counts to 511 (255 for the DP8408A) before rolling over to zero.
3) When going from mode 0, 1 or 2 (refresh) to mode 5 of the DP8408A, if $\overline{CASIN}$ and R/$\overline{C}$ are both low, a glitch occurs on the $\overline{CAS}$ output. Since neither of these inputs is used in these modes, one or both should be held high.
4) Most DRAMs specify 0 ns row address set-up time to $\overline{RAS}$. In order to guarantee this, the row address to the DP8408A, DP8409A must be valid 10 ns before $\overline{RASIN}$ transitions low to initiate an access. In terms of the data sheet parameters, maximum ($t_{APD} - t_{RPDL}$) = 10 ns.
5) When changing modes from refresh to access, again sufficient time must be allowed for the row address to be valid before $\overline{RAS}$ occurs. In this case, the address outputs of the DP8408A, DP8409A are changing from the refresh counter to the row address inputs. In order for the row address to be set up a minimum of 0 ns before $\overline{RAS}$ goes low, $\overline{RASIN}$ should not go low until 30 ns after the change from refresh to access mode.
6) Both the low and high pulse widths of $\overline{RAS}$ have minimum requirements during refresh. When in mode 0, the $\overline{RASIN}$ to $\overline{RAS}$ low delay is longer than the $\overline{RASIN}$ to $\overline{RAS}$ high delay. In terms of the data sheet parameters, maximum ($t_{RFPDL} - t_{RFPDH}$) = 25 ns. Thus, the minimum low pulse width of $\overline{RAS}$ in mode 0 equals the $\overline{RASIN}$ low pulse width minus 25 ns. The minimum high pulse width of $\overline{RAS}$ in mode 0 equals the $\overline{RASIN}$ high pulse width.
7) The fastest memory access may be accomplished using mode 4 and external delay lines (see App. Brief #9).
8) In the data sheet, it is specified that $\overline{CS}$ should go low 30 ns (t_{CSLR}) before $\overline{RASIN}$ goes low to initiate an access in mode 5. This is to prevent the possibility of a glitch on the $\overline{RAS}$ outputs, resulting from the DP8409A interpreting the $\overline{RASIN}$ as a hidden refresh. For the same reason, $\overline{CS}$ should be held low for a minimum of 15 ns after $\overline{RASIN}$ returns high, ending the access in mode 5.
9) If the DP8409A is being used in mode 5 and $\overline{CS}$ = 1, and if $\overline{RASIN}$ goes low within 15 ns before RFCK (R/$\overline{C}$) goes low, up to a 15 ns glitch may occur on the refresh request pin, RFI/O. However, since $\overline{CS}$ is high, a hidden refresh will occur as it normally would with RFCK high. If the glitch on RFI/O were detected and interpreted as a forced refresh request, no forced refresh would be allowed by the DP8409A since a hidden refresh was allowed. This would not cause any problem, however, since the hidden refresh has taken care of the refresh requirement for that period of RFCK. Also, this forced refresh request could not be detected if the system does not check RFI/O for a low state while $\overline{RASIN}$ is low (i.e., an access is taking place).
10) At CPU clock frequencies of 10 MHz and above it is suggested that the hidden refresh capability of the DRAM controller (DP8409/17/19/29) be disabled. The main reason for this suggestion is to satisfy the parameter "t_{RKRL}" (RFCK high to $\overline{RASIN}$ low for hidden refresh) which is given as a minimum of 50 ns in the DP8417/19/29 data sheets. Disabling hidden refresh also eliminates the need of meeting the parameter of "t_{CSRL1}" ($\overline{CS}$ low to access $\overline{RASIN}$ low using Mode 5 with hidden refresh capability) which is given as a minimum of 34 ns in the DP8417/19/29 data sheets. In order to eliminate hidden refresh the "$\overline{CS}$" pin of the DRAM controller should be permanently grounded on the DRAM controller, and the "$\overline{CS}$" that previously went to the DRAM controller should be "ORed" with "$\overline{RASIN}$" (the "OR" gate's output becoming the new "$\overline{RASIN}$" input to the DRAM controller).
11) If the user desires to improve the DRAM controller "$\overline{RASIN}$ to $\overline{RAS}$ out" time ("tRPDL") external logic may be used to create multiple "$\overline{RAS}$s". The circuit shown below requires only several 74XX oxide isolated type IC's (74AS27 and 74AS04) to accomplish this aim. To use this circuit $\overline{RASIN}$ should transition low during refreshes.



TL/F/5033–1

DP8408A/9A Fastest DRAM Access Mode

National Semiconductor
Application Brief
Tim Garverick
Rusty Meier

If one desires the fastest possible operation of the DP8408A/9A multi-mode dynamic RAM controller/driver in accessing DRAMs, mode 4, externally controlled access mode should be considered.

In using mode 4 there are three input signals which must be considered:

1) $\overline{\text{RASIN}}$—generates $\overline{\text{RAS}}$
2) R/$\overline{\text{C}}$—switches between rows and columns on the address outputs
3) $\overline{\text{CASIN}}$—generates $\overline{\text{CAS}}$

In producing these signals a delay will be needed between $\overline{\text{RASIN}}$ and R/$\overline{\text{C}}$ and between R/$\overline{\text{C}}$ and $\overline{\text{CASIN}}$. **(Note:** In mode 4 external generation of $\overline{\text{CASIN}}$ can produce $\overline{\text{CAS}}$ faster than automatic generation of $\overline{\text{CAS}}$.)

Two important parameters have been added to the DP8408A/9A data sheets that help one compute the minimum acceptable delays between the above-mentioned signals. These parameters are:

1) t_{DIF1} = MAXIMUM ($t_{RPDL} - t_{RHA}$) = 13 ns
   where t_{RPDL} = $\overline{\text{RASIN}}$ to $\overline{\text{RAS}}$ delay
   t_{RHA} = row address held from column select
2) t_{DIF2} = MAXIMUM ($t_{RCC} - t_{CPDL}$) = 13 ns
   where t_{RCC} = column select to column address valid
   t_{CPDL} = $\overline{\text{CASIN}}$ to $\overline{\text{CAS}}$ delay

These parameters are specified as being less than what would be calculated using the min/max values given for t_{RCC}, t_{CPDL}, t_{RPDL} and t_{RHA} in the DP8408A/9A specification sheets, because on-chip delays track over temperature and supply variations.

The equation for the delay between $\overline{\text{RASIN}}$ and R/$\overline{\text{C}}$ that guarantees the specified DRAM t_{RAH} is:

min delay required = $t_{DIF1} + t_{RAH}$
= 13 ns + t_{RAH}
where t_{RAH} = DRAM minimum row address hold time from $\overline{\text{RAS}}$

The equation for the delay between R/$\overline{\text{C}}$ and $\overline{\text{CASIN}}$ that guarantees the specified DRAM t_{ASC} is:

min delay required = $t_{DIF2} + t_{ASC}$
= 13 ns + t_{ASC}
where t_{ASC} = DRAM minimum column address set-up time to $\overline{\text{CAS}}$

To produce the above-mentioned delays between signals, a ±2 ns resolution delay line can be used as follows:

(assuming t_{RAH} = 20 ns, t_{ASC} = 0 ns)
$\overline{\text{RASIN}}$ to R/$\overline{\text{C}}$ delay = 13 ns + 20 ns
= 33 ns
R/$\overline{\text{C}}$ to $\overline{\text{CASIN}}$ delay = 13 ns + 0 ns
= 13 ns

Thus, R/$\overline{\text{C}}$ must follow $\overline{\text{RASIN}}$ by a minimum of 33 ns and $\overline{\text{CASIN}}$ must follow R/$\overline{\text{C}}$ by a minimum of 13 ns. With a delay line of ± 2 ns resolution, the $\overline{\text{RASIN}}$ to R/$\overline{\text{C}}$ and R/$\overline{\text{C}}$ to $\overline{\text{CASIN}}$ delays can be typicals of 35 ns and 15 ns, respectively. (See *Figures 1* and *2*.)

This scheme will provide a maximum $\overline{\text{RASIN}}$ to $\overline{\text{CAS}}$ delay of:

35 ns + 15 ns + 2 ns (resolution uncertainty)
+ MAXIMUM (t_{CPDL}) = 52 ns + MAXIMUM (t_{CPDL})

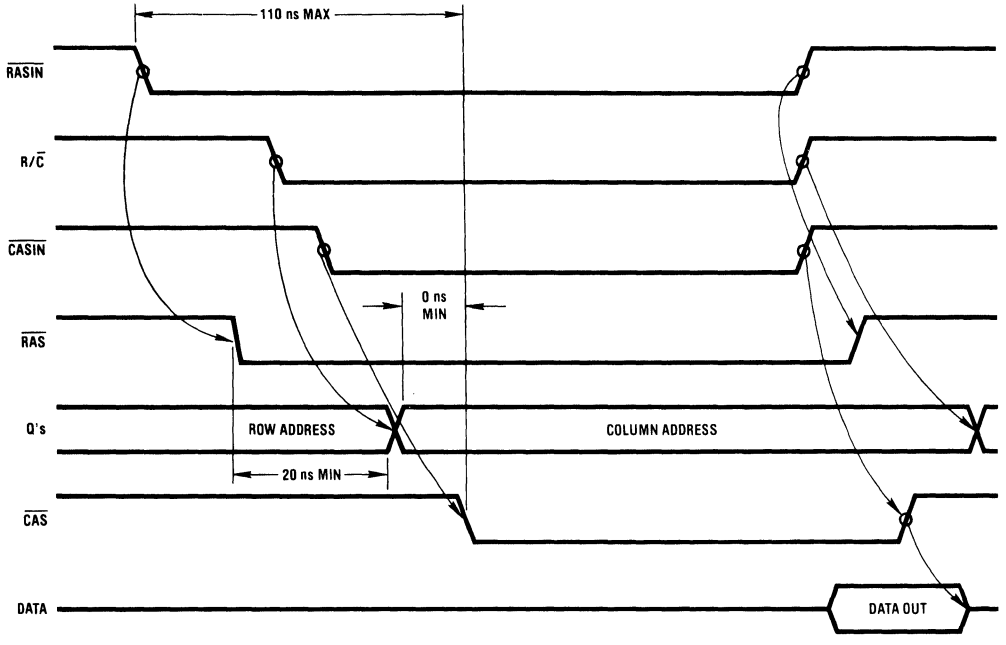
For the DP8408/9-2, MAXIMUM (t_{CPDL}) = 58 ns.

For the DP8408A/9A (no dash), MAXIMUM (t_{CPDL}) = 68 ns (not 58 ns as indicated in data sheets up to November 1982).

The fastest mode 4 accesses (with the assumed delay line and DRAM parameters) are therefore, 110 ns and 120 ns, respectively, for the -2 and non-dash parts.

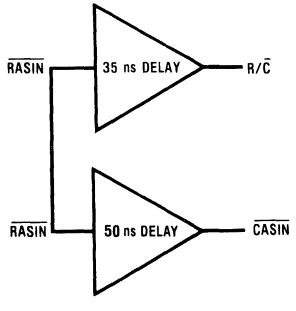
The maximum $\overline{\text{RASIN}}$ to $\overline{\text{CAS}}$ delay (t_{RICL}) in mode 5 (auto mode) for the DP8408/9-2 (which guarantees a min t_{RAH} of 20 ns) is 130 ns. The maximum t_{RICL} in mode 5 for the DP8408A/9A (no dash) is 160 ns.

Thus, it is shown that if the features offered by the DP8408A/9A automatic modes can be sacrificed, mode 4 (externally controlled access) may be used to obtain the fastest memory access.



TL/F/8403–1

FIGURE 1. Mode 4 Timing Relationships



TL/F/8403–2

FIGURE 2. Mode 4 Externally Generated Signals

Section 2

Error Detection and Correction

Section 2 Contents

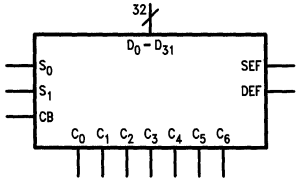
National Semiconductor

54F/74F420
Parallel Check Bit/Syndrome Bit Generator

General Description

The 'F420 is a parallel check bit/syndrome bit generator. The 'F420 utilizes a modified hamming code to generate 7 check bits from a 32-bit dataword, in 15 ns, when operated in the check bit generate mode. When operated in the syndrome generate mode, the check bits and data bits read from memory are utilized in a parity summer to generate syndrome bits upon error detection. The maximum error count detectable is 2. A single error detect can occur in 18 ns; a double error detect in 22 ns. The syndrome bit generation can be output in 15 ns (maximum).

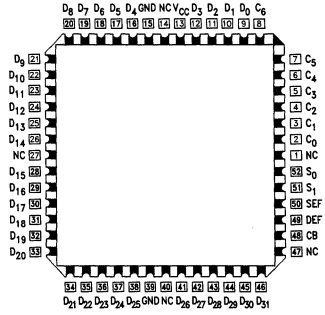
Logic Diagram



TL/F/9542–1

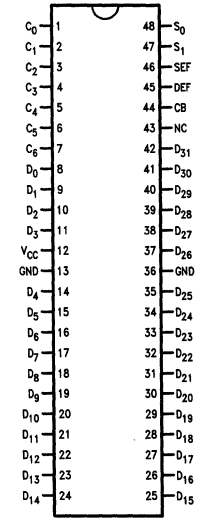
Connection Diagrams

Pin Assignment for LCC and PCC



TL/F/9542–2

Pin Assignment for DIP and Flatpak



TL/F/9542–3

Unit Loading/Fan Out: See Section 1 for U.L. definitions

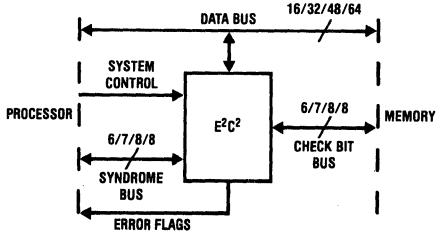
Pin Names	Description	54F/74F	
		U.L. HIGH/LOW	Input I_{IH}/I_{IL} Output I_{OH}/I_{OL}
C_0–C_6	Check Bit/Syndrome Bus Inputs/ Outputs	3.5/1.083 150/40 (33.3)	70 μA/−0.65 mA −3 mA/24 mA (20 mA)
D_0–D_{31}	Data Bit Bus	1.0/1.0	20 μA/−0.6 mA
CB	Check Bit Control	1.0/1.0	20 μA/−0.6 mA
DEF	Double Error Flag	50/33.3	−1 mA/20 mA
SEF	Single Error Flag	50/33.3	−1 mA/20 mA
S_0, S_1	Mode Control	1.0/1.0	20 μA/−0.6 mA

National Semiconductor

DP8400-2—E²C² Expandable Error Checker/Corrector

General Description

The DP8400-2 Expandable Error Checker and Corrector (E^2C^2) aids system reliability and integrity by detecting errors in memory data and correcting single or double-bit errors. The E^2C^2 data I/O port sits across the processor-memory data bus as shown, and the check bit I/O port connects to the memory check bits. Error flags are provided, and a syndrome I/O port is available. Fabricated using high speed Schottky technology in a 48-pin dual-in-line package, the DP8400-2 has been designed such that its internal delay times are minimal, maintaining maximum memory performance.



TL/F/6899–1

For a 16-bit word, the DP8400-2 monitors data between the processor and memory, with its 16-bit bidirectional data bus connected to the memory data bus. The DP8400-2 uses an encoding matrix to generate 6 check bits from the 16 bits of data. In a WRITE cycle, the data word and the corresponding check bits are written into memory. When the same location of memory is subsequently read, the E^2C^2 generates 6 new check bits from the memory data and compares them with the 6 check bits read from memory to create 6 syndrome bits. If there is a difference (causing some syndrome bits to go high), then that memory location contains an error and the DP8400-2 indicates the type of error with 3 error flags. If the error is a single data-bit error, the DP8400-2 will automatically correct it.

The DP8400-2 is easily expandable to other data configurations. For a 32-bit data bus with 7 check bits, two DP8400-2s can be used in cascade with no other ICs. Three DP8400-2s can be used for 48 bits, and four DP8400-2s for 64 data bits, both with 8 check bits. In all these configurations, single and double-error detection and single-error correction are easy to implement.

When the memory is more unreliable, or better system integrity is preferred, then in any of these configurations, double-error correction can be performed. One approach requires a further memory WRITE-READ cycle using complemented data and check bits from the DP8400-2. If at least one of the two errors is a hard error, the DP8400-2 will correct both errors. This implementation requires no more memory check bits or DP8400-2s than the single-error correct configurations.

The DP8400-2 has a separate syndrome I/O bus which can be used for error logging or error management. In addition, the DP8400-2 can be used in BYTE-WRITE applications (for up to 72 data bits) because it has separate byte controls for the data buffers. In 16 or 32-bit systems, the DP8400-2 will generate and check system byte parity, if required, for integrity of the data supplied from or to the processor. There are three latch controls to enable latching of data in various modes and configurations.

Operational Features

- Fast single and double-error detection
- Fast single-error correction
- Double-error correction after catastrophic failure with no additional ICs or check bits
- Functionally expandable to 100% double-error correct capability
- Functionally expandable to triple-error detect
- Directly expandable to 32 bits using 2 DP8400-2s only
- Directly expandable to 48 bits using 3 DP8400-2s only
- Directly expandable to 64 bits using 4 DP8400-2s only
- Expandable to and beyond 64 bits in fast configuration with extra ICs
- 3 error flags for complete error recording
- 3 latch enable inputs for versatile control
- Byte parity generating and checking
- Separate byte controls for outputting data in BYTE-WRITE operation
- Separate syndrome I/O port accessible for error logging and management
- On-chip input and output latches for data bus, check bit bus and syndrome bus
- Diagnostic capability for simulating check bits
- Memory check bit bus, syndrome bus, error flags and internally generated syndromes available on the data bus
- Self-test of E^2C^2 on the memory card under processor control
- Full diagnostic check of memory with the E^2C^2
- Complete memory failure detectable
- Power-on clears data and syndrome input latches

Timing Features

16-BIT CONFIGURATION

WRITE Time: 29 ns from data-in to check bits valid

DETECT Time: 21 ns from data-in to Any Error (AE) flag set

CORRECT Time: 44 ns from data-in to correct data out

Timing Features (Continued)

32-BIT CONFIGURATION

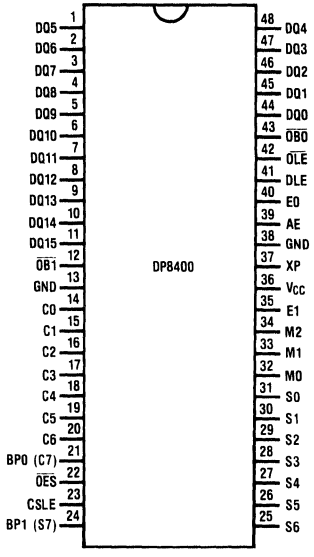
WRITE Time: 49 ns from data-in to check bits valid

DETECT Time: 46 ns from data-in to Any Error (AE) flag set

CORRECT Time: 84 ns from data-in to correct data out

DP8400-2 Connection Diagram

Dual-In-Line Package

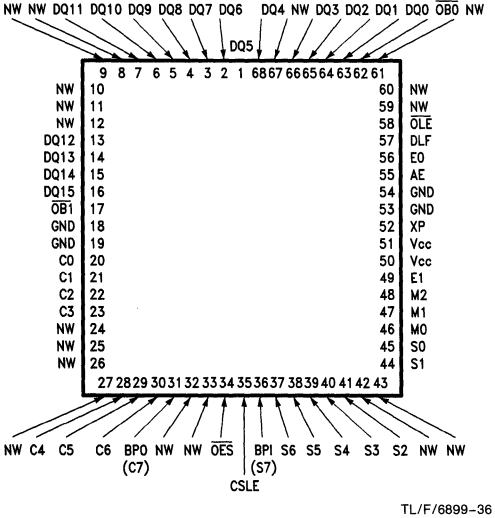


TL/F/6899–2

Top View

Order Number DP8400V-2, DP8400N-2, or DP8400D-2
See NS Package V68, N48A or D48A

Chip Carrier Package



TL/F/6899–36

Top View

Pin Descriptions

Pin #	Description
1	DQ5
2	DQ6
3	DQ7
4	DQ8
5	DQ9
6	DQ10
7	DQ11
13	DQ12
14	DQ13
15	DQ14
16	DQ15
17	$\overline{OB1}$
18	GND
19	GND
20	C0
21	C1
22	C2
23	C3
28	C4
29	C5
30	C6
31	BP0 (C7)
34	$\overline{OES}$
35	CSLE
36	BP1 (S7)
37	S6
38	S5
39	S4
40	S3
41	S2
44	S1
45	S0
46	M0
47	M1
48	M2
49	E1
50	V_{CC}
51	V_{CC}
52	XP
53	GND
54	GND
55	AE
56	E0
57	DLF
58	$\overline{OLE}$
62	$\overline{OB0}$
63	DQ0
64	DQ1
65	DQ2
66	DQ3
68	DQ4

Note: Pins 8, 9, 10, 11, 12, 24, 25, 26, 27, 32, 33, 42, 43, 59, 60, 61, and 67 are all **NW**.

Pin Definitions See *Figure 1* for abbreviations

V_{CC}, GND, GND: 5.0V ±5%. The 3 supply pins have been assigned to the center of the package to reduce voltage drops, both DC and AC. Also there are two ground pins to reduce the low-level noise. The second ground pin is located two pins from V_{CC}, so that decoupling capacitors can be inserted directly next to these pins. It is important to adequately decouple this device, due to the high switching currents that will occur when all 16 data bits change in the same direction simultaneously. A recommended solution would be a 1 μF multilayer ceramic capacitor in parallel with a low-voltage tantalum capacitor, both connected close to pins 36 and 38 to reduce lead inductance.

DQ0–DQ15: Data I/O port. 16-bit bidirectional data bus which is connected to the input of DIL0 and DIL1 and the output of DOB0 and DOB1, with DQ8–DQ15 also to CIL.

C0–C6: Check-bit I/O port. 7-bit bidirectional bus which is connected to the input of the CIL and the output of the COB. COB is enabled whenever M2 is low.

S0–S6: Syndrome I/O port. 7-bit bidirectional bus which is connected to the input of the SIL and the output of the SOB.

DLE: Input data latch enable. When high, DIL0 and DIL1 outputs follow the input data bus. When low, DIL0 and DIL1 latch the input data.

CSLE: Input check bit and syndrome latch enable. When high, CIL and SIL follow the input check and syndrome bits. When low, CIL and SIL latch the input check and syndrome bits. If $\overline{OES}$ is low, SIL remains latched.

$\overline{OLE}$: Output latch enable. $\overline{OLE}$ enables the internally generated data to DOL0, and DOL1, COL and SOL when low, and latches when high.

XP: Multi-expansion, which feeds into a three-level comparator. With XP at 0V, only 6 or 7 check bits are available for expansion up to 40 bits, allowing byte parity capability. With XP open or at V_{CC}, expansion beyond 40 bits is possible, but byte parity capability is no longer available. When XP is at V_{CC}, CG6 and CG7, the internally generated upper two check bits, are set low. When XP is open, CG6 and CG7 are set to word parity.

BP0 (C7): When XP is at 0V, this pin is byte-0 parity I/O. In the Normal WRITE mode, BP0 receives system byte-0 parity, and in the Normal READ mode outputs system byte-0 parity. When XP is open or at V_{CC}, this pin becomes C7 I/O, the eighth check bit for the memory check bits, for 48-bit expansion and beyond.

BP1 (S7): When XP is at 0V, this pin is byte-1 parity I/O. In the Normal WRITE mode, BP1 receives system byte-1 parity, and in the Normal READ mode outputs system byte-1 parity. When XP is open or at V_{CC}, this pin becomes S7 I/O, the eight syndrome bit for 48-bit expansion and beyond.

AE: Any error. In the Normal READ mode, when low, AE indicates no error and when high, indicates that an error has occurred. In any WRITE mode, AE is permanently low.

E0: In the Normal READ mode, E0 is high for a single-data error, and low for other conditions. In the Normal WRITE mode, E0 becomes $\overline{PE0}$ and is low if a parity error exists in byte-0 as transmitted from the processor.

E1: In the Normal READ mode, E1 is high for a single-data error or a single check bit error, and low for no error and double-error. In the Normal WRITE mode, E1 becomes $\overline{PE1}$ and is low if a parity error exists in byte-1 as transmitted from the processor.

$\overline{OB0}$, $\overline{OB1}$: Output byte-0 and output byte-1 enables. These inputs, when low, enable DOL0 and DOL1 through DOB0 and DOB1 onto the data bus pins DQ0–DQ7 and DQ8–DQ15. When $\overline{OB0}$ and $\overline{OB1}$ are high the DOB0, DOB1 outputs are TRI-STATE®.

$\overline{OES}$: Output enables syndromes. I/O control of the syndrome latches. When high, SOB is TRI-STATE and external syndromes pass through the syndrome input latch with CSLE high. When $\overline{OES}$ is low, SOB is enabled and the generated syndromes appear on the syndrome bus, also CSLE is inhibited internally to SIL.

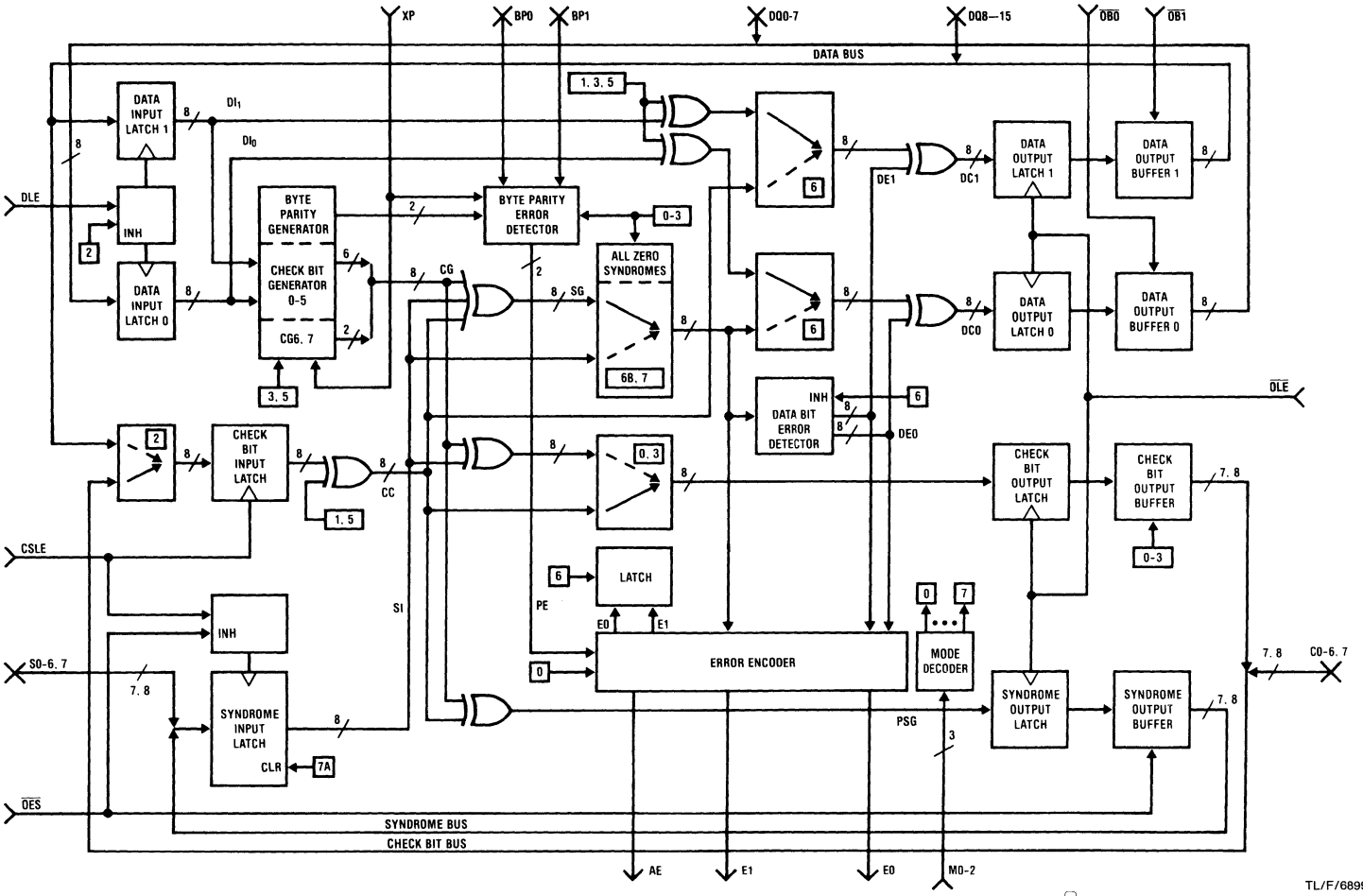
M0, M1, M2: Mode control inputs. These three controls define the eight major operational modes of the DP8400-2. Table III depicts the modes.

System Write (*Figure 2a*)

The Normal WRITE mode is mode 0 of Table III. Referring to the block diagram in *Figure 9a* and the timing diagram of *Figure 9b*, the 16 bits of data from the processor are enabled into the data input latches, DIL0 and DIL1, when the input data latch enable (DLE) is high. When this goes low, the input data is latched. The check bit generator (CG) then produces 6 parity bits, called check bits. Each parity bit monitors different combinations of the input data-bits. In the 16-bit configuration, assuming no syndrome bits are being fed in from the syndrome bus into the syndrome input latch, the 6 check bits enter the check bit output latch (COL), when the output latch enable $\overline{OLE}$ is low, and are latched in when $\overline{OLE}$ goes high. Whenever M2 (READ/$\overline{WRITE}$) is low, the check bit output buffer COB always enables the COL contents onto the external check bit bus. Also the data error decoder (DED) is inhibited during $\overline{WRITE}$ so no correction can take place. Data output latches DOL0 and DOL1, when enabled with $\overline{OLE}$, will therefore see the contents of DIL0 and DIL1. If valid system data is still on the data bus, a memory WRITE will write to memory the data on the data bus and the check bits output from COB. If the system has vacated the data bus, output enables ($\overline{OB0}$ and $\overline{OB1}$) must be set low so that the original data word with its 6 check bits can be written to memory.

System Read

There are two methods of reading data: the error monitoring method (*Figure 2b*), and the always correct method (*Figure 2c*). Both require fast error detection, and the second, fast correction. With the first method, the memory data is only monitored by the E^2C^2, and is assumed to be correct. If there is an error, the Any Error flag (AE) goes high, requiring further action from the system to correct the data. With the always correct method, the memory data is assumed to be possibly in error. Memory data is removed and the corrected, or already correct, data is output from the E^2C^2 by enabling $\overline{OB1}$ and $\overline{OB0}$. To detect an error (referring to *Figures 10a* and *10b*) first DLE and CSLE go high to enter data bits and check bits from memory into DIL0, DOL1 and CIL. The 6 check bits generated in CG from DIL0 and DOL1 are then compared with CIL to generate syndromes on the internal syndrome bus (SG). Any bit or bits of SG that go high indicate an error to the error encoder (EE).



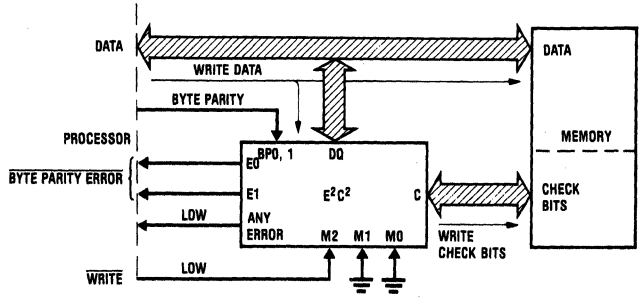
DIL	Data Input Latch	SG	Syndrome Generator	SOL	Syndrome Output Latch
CG	Check Bit Generator	DED	Data Error Detector	DOB0, 1	Data Output Buffer Bytes 0, 1
CIL	Check Bit Input Latch	DE0, 1	Data Error Bytes 0, 1	COB	Check Bit Output Buffer
CC	Check Bit Complementor	PE	Parity Error	SOB	Syndrome Output Buffer
SIL	Syndrome Input Latch	DOL0, 1	Data Output Latch Bytes 0, 1	EE	Error Encoder
PSG	Partial Syndrome Generator	COL	Check Bit Output Latch	DC0, 1	Data Corrector Bytes 0, 1

(n) mode of operation signifies active signal

FIGURE 1. DP8400-2 Block Diagram

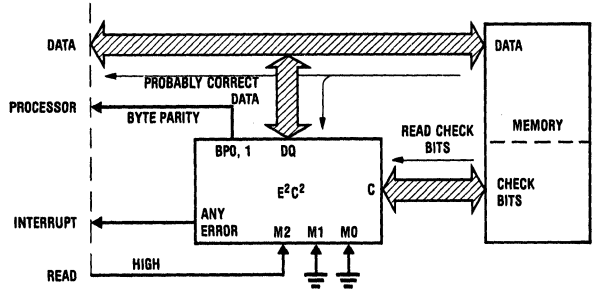
DP8400-2

System Diagrams—Modes of Operation



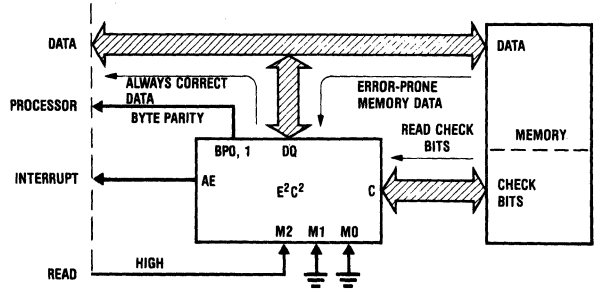
TL/F/6899–4

FIGURE 2a. Normal WRITE Mode with E^2C^2



TL/F/6899–5

FIGURE 2b. Normal READ Mode, Error Monitoring Method with E^2C^2



TL/F/6899–6

FIGURE 2c. Normal READ Mode, Always Correct Method with E^2C^2

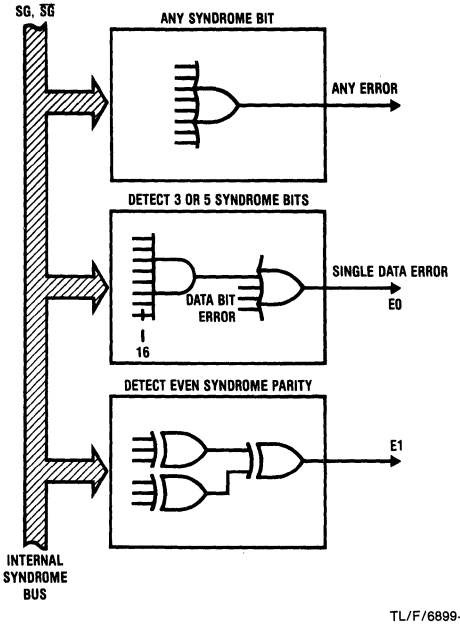
System Read (Continued)

If data correction is required $\overline{OB0}$ and $\overline{OB1}$ must be set low (after memory data has been disabled) to enable data output buffers DOB0 and DOB1. The location of any data bit error is determined by the data error decoder (DED), from the syndrome bits. The bit in error is complemented in the DOL for correction. The other 15 bits from DED pass the DIL contents directly to the DOL, so that DOL now contains corrected data.

Error Determination

The three error flags, for a 16-bit example, are decoded from the internally generated syndromes as shown in *Figure 3*. First, if any error has occurred, the generated check bits will be different from the memory check bits, causing some of the syndrome bits to go high. By OR-ing the syndrome bits, the output will be an indication of any error.

If there is a single-data error, then (from the matrix in Table IV) it can be seen that any data error causes either 3 or 5 syndrome bits to go high. 16 AND gates decode which bit is in error and the bit in error is XOR-ed with the corresponding bit of the DIL to correct it, whereas the other 15 decoder outputs are low, causing the corresponding 15 bits in DIL to transfer to DOL directly. DOL now contains corrected data. The 16 AND gate outputs are OR-ed together causing E0 to go high, so that E0 is the single-data-error indication. If the error is a double-error, then either 2, 4 or 6 of the syndrome bits will be high. The syndromes for two errors (including one or two check bit errors) are the two sets of syndromes for each individual error bit, XOR-ed together. By performing a parity check on the syndrome bits, flag E1 will indicate even/odd parity. If there is still an error, but it is not one of these errors, then it is a detectable triple-bit error. Some triple-bit errors are not detectable as such and may be interpreted as single-bit errors and falsely corrected as single-data errors. This is true for all standard ECC circuits using a Modified Hamming-code matrix. The DP8400-2 is capable, with its Rotational Syndrome Word Generator matrix, of determining all triple-bit errors using twice as many DP8400-2s and twice as many check bits.



TL/F/6899–7

FIGURE 3. Error Encoder

Error Flags

Three error flags are provided to allow full error determination. Table I shows the error flag outputs for the different error types in Normal READ mode. If there is an error, then ANY ERROR will go high, at a time t_{DEV} (*Figure 10b*) after data and check bits are presented to the DP8400-2. The other two error flags E0 and E1 become valid t_{DE0} and t_{DE1} later.

The error flags differentiate between no error, single check bit error, single data-bit error, double-bit error. Because the DP8400-2 can correct double errors, it is important to know that two errors have occurred, and not just a multiple-error indication. The error flags will remain valid as long as DLE and CSLE are low, or if DLE is high, and data and check bits remain valid.

Byte Parity Support

Some systems require extra integrity for transmission of data between the different cards. To achieve this, individual byte parity bits are transmitted with the data bits in both directions. The DP8400-2 offers byte parity support for up to 40 data bits. If the processor generates byte parity when transferring information to the memory, during the WRITE cycle, then each byte parity bit can be connected to the corresponding byte parity I/O pin on the DP8400-2, either BP0 or BP1. The DP8400-2 develops its own internal byte parity bits from the two bytes of data from the processor, and compares them with BP0 and BP1 using an exclusive-OR for both parities. The output of each exclusive-OR is fed to the error flags E0 and E1 as $\overline{PE0}$ and $\overline{PE1}$, so that a byte parity error forces its respective error flag low, as in Table II. These flags are only valid for the Normal WRITE (mode 0) and XP at 0V. The DP8400-2 checks and generates even byte parity.

When transferring information from the memory to the processor, the DP8400-2 receives the memory data, and outputs the corresponding byte parity bits on BP0 and BP1 to the processor. The processor block can then check data integrity with its own byte parity generator. If in fact memory data was in error, the DP8400-2 derives BP0 and BP1 from the corrected data, so when corrected data is output from the DP8400-2, the processor will not detect a byte parity error.

During the read mode, DP8400-2 corrects single data bit error and also its parity.

TABLE I. Error Flags After Normal Read (Mode 4)

AE	E1	E0	Error Type
0	0	0	No error
1	1	0	Single check bit error
1	1	1	Single-data error
1	0	0	Double-bit error
All Others			Invalid conditions

TABLE II. Error Flags after Normal Write (Mode 0)

AE	E1 ($\overline{PE1}$)	E0 ($\overline{PEJ}$)	Error Type
0	1	1	No parity error
0	1	0	Parity error, byte 0
0	0	1	Parity error, byte 1
0	0	0	Parity error, bytes 0, 1

TABLE III. DP8400-2 Modes of Operation

Mode	M2 (R/W)	M1	M0	$\overline{OES}$	Operation
0	0	0	0	X	Normal WRITE DIL → DOL, CG → COL → COB
1	0	0	1	X	Complement WRITE $\overline{DIL}$ → DOL, $\overline{CIL}$ → COL → COB
2	0	1	0	X	Diagnostic WRITE, DLE inhibited DQ8–DQ15 ⊕ CG → SOL → SOB DQ8–DQ15 → CIL → COL → COB
3	0	1	1	X	Complement data-only WRITE $\overline{DIL}$ → DOL, (CG0, 1, 4, 5, $\overline{CG2}$, $\overline{CG3}$) → COL → COB
4	1	0	0	X	Normal READ DIL ⊕ DE → DOL, CIL → COL
5	1	0	1	X	Complement READ $\overline{DIL}$ ⊕ DE → DOL, $\overline{CIL}$ → COL
6A	1	1	0	0	READ generated syndromes, check bit bus, error flags, SG0–SG6 → DQ0–DQ6, CIL0–CIL6 → DQ8–DQ14, E1 → DQ7, E0 → DQ15
6B	1	1	0	1	READ syndrome bus, check bit bus, error flags, SIL0–SIL6 → DQ0–DQ6, CIL0–CIL6 → DQ8–DQ14, E1 → DQ7, E0 → DQ15
7A	1	1	1	0	Generated syndromes replace with zero 0 → SIL → SG, CIL → COL, DIL ⊕ DE → DOL
7B	1	1	1	1	Generated syndromes replace SIL → SG, CIL → COL, DIL ⊕ DE → DOL

TABLE IV. Data-In To Check Bit Generate, Or Data Bit Error To Syndrome-Generate Matrix (16-Bit Configuration)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
	GENERATE CHECK BITS →																DQ0-15
0	0	0	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0
1	0	0	0	1	0	0	1	0	1	1	0	1	0	1	1	1	1
2	1	0	0	1	1	0	0	0	1	0	1	0	1	1	1	1	2*
3	0	1	1	0	0	0	0	1	1	1	1	0	1	0	1	1	3*
4	1	1	0	0	0	1	0	1	1	0	0	1	0	1	0	1	4
5	1	1	1	0	1	1	1	0	1	0	0	0	1	1	1	0	5
	4	8	9	7	5	1	3	9	E	B	D	3	C	7	F	F	0
	3	3	2	0	2	3	2	1	3	0	0	1	2	3	2	1	1

GENERATED SYNDROMES

GENERATED CHECK BITS

HEXADECIMAL EQUIVALENT OF SYNDROME BITS

*C2, C3 generate odd parity

TL/F/6899–8

Modes of Operation

There are three mode-control pins, M2, M1 and M0, offering 8 major modes of operation, according to Table III.

M2 is the READ/$\overline{\text{WRITE}}$ control. In normal operation, mode 0 is Normal WRITE and mode 4 is Normal READ. By clamping M0 and M1 low, and setting M2 low during WRITE and high during READ, the DP8400-2 is very easy to use for normal operation. The other modes will be covered in later sections.

16-BIT CONFIGURATION

The first two rows on top of the check bit generate matrix (Table IV) indicate the data position of DQ0 to DQ15. The left side of the matrix, listed 0 to 5, corresponds to syndromes S0 to S5. S0 is the least significant syndrome bit. There are two rows of hexadecimal numbers below the matrix. They are the hex equivalent of the syndrome patterns. For example, syndrome pattern in the first column of the matrix is 001011. Its least significant four bits (0010) equal hexadecimal 4, and the remaining two bits (11) equal hexadecimal 3.

Check bit generation is done by selecting different combinations of data bits and generating parities from them. Each row of the check bit generate matrix corresponds to the generation of a check bit numbered on the right hand side of the matrix, and the ones in that row indicate the selection of data bits.

The following are the check bit generate equations for 16-bit wide data words:

CG0 = DQ2 ⊕ DQ3 ⊕ DQ4 ⊕ DQ5 ⊕ DQ6 ⊕ DQ7 ⊕ DQ9 ⊕ DQ10 ⊕ DQ11 ⊕ DQ13 ⊕ DQ14 ⊕ DQ15

CG1 = DQ3 ⊕ DQ6 ⊕ DQ8 ⊕ DQ9 ⊕ DQ11 ⊕ DQ13 ⊕ DQ14 ⊕ DQ15

*CG2 = DQ0 ⊕ DQ3 ⊕ DQ4 ⊕ DQ8 ⊕ DQ10 ⊕ DQ12 ⊕ DQ13 ⊕ DQ14 ⊕ DQ15 ⊕ 1

*CG3 = DQ1 ⊕ DQ2 ⊕ DQ7 ⊕ DQ8 ⊕ DQ9 ⊕ DQ10 ⊕ DQ12 ⊕ DQ14 ⊕ DQ15 ⊕ 1

CG4 = DQ0 ⊕ DQ1 ⊕ DQ5 ⊕ DQ7 ⊕ DQ8 ⊕ DQ11 ⊕ DQ13 ⊕ DQ15

CG5 = DQ0 ⊕ DQ1 ⊕ DQ2 ⊕ DQ4 ⊕ DQ5 ⊕ DQ6 ⊕ DQ8 ⊕ DQ12 ⊕ DQ13 ⊕ DQ14

*CG2 and CG3 are odd parities.

The following error map (Table V) depicts the relationship between all possible error conditions and their associated syndrome patterns. For example, if a syndrome pattern is S0–5 = 111101, data bit 14 is in error.

Figure 4 shows how to connect one DP8400-2 in a 16-bit configuration, in order to detect and correct single or double-bit errors. For a Normal WRITE, processor data is presented to the DP8400-2, where it is fed through DIL0 and DIL1 to the check bit generator. This generates 6 parity bits from different combinations of data bits, according to Table IV. The numbers in the row below the table are the hexadecimal equivalent of the column bits (with bits 6, 7 low). A "1" in any row indicates that the data bit in that column is connected to the parity generator for that row. For example, check bit 1 generates parity from data bits 3, 6, 8, 9, 11, 13, 14, and 15.

Check bits 0, 1, 4, 5, and 6 generate even parity, and check bits 2 and 3 generate odd parity. This is done to insure that a total memory failure is detected. If all check bits were even parity, then all zeroes in the data word would generate all check bits zero and a total memory failure would not be detected when a memory READ was performed. Now all-zero-data bits produce C2 and C3 high and a total memory failure will be detected. When reading back from the same location, the memory data bits (possibly in error) are fed to the same check bit generator, where they are compared to the memory check bits (also possibly in error) using 6 exclusive-OR gates. The outputs of the XORs are the syndrome bits, and these can be determined according to Table IV for one data bit error. For example, an error in bit 2 will produce the syndrome word 101001 (for S5 to S0 respectively). The syndrome word is decoded by the error encoder to the error flags, and the data-error decoder to correct a single data bit error. Assuming the memory data has been latched in the DIL, by making DLE go low, memory data can be disabled. Then by setting $\overline{\text{OB0}}$ and $\overline{\text{OB1}}$ low, corrected data will appear on the data bus. The syndromes are available as outputs on pins S0–5 when $\overline{\text{OES}}$ is low. It is also possible to feed in syndromes to SIL when $\overline{\text{OES}}$ is high and CSLE goes high. This can be useful when using the Error Management Unit shown in *Figure 4*. C6 and S6 are not used for 16 bits. It is safe therefore to make C6 appear low, through a 2.7 kΩ resistor to ground. The same applies for S6 if syndromes are input to the DP8400-2. If $\overline{\text{OES}}$ is permanently low, S6 may be left open.

Any 16-bit memory correct system using the DP8400-2 without syndrome inputs must keep the $\overline{\text{OES}}$ pin grounded, then all the syndrome I/O pins may be left open. The reason for this is that the DP8400-2 resets the syndrome input latch at power up. If the $\overline{\text{OES}}$ pin is grounded, the syndrome input latch will remain reset for normal operations.

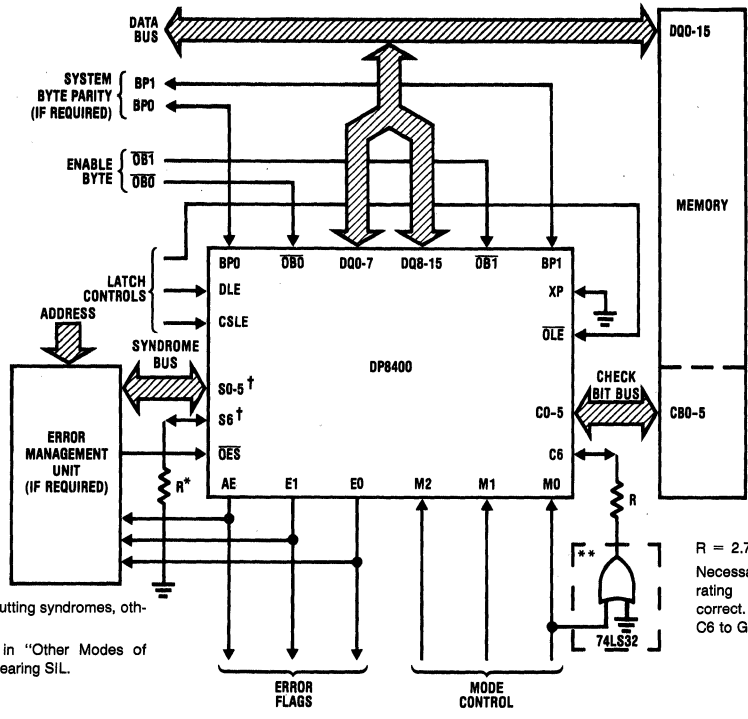
The parameter t_{NMR} (see *Figure 10b*), new mode recognized time, is measured from M2 (changing from READ to WRITE) to the valid check bits appearing on the check bit bus, provided the $\overline{\text{OLE}}$ was held low.

The parameter t_{MCR} (see *Figure 10b*), mode change recognized time, is measured from M2 (changing from WRITE to

TABLE V. Syndrome Decode To Bit In Error For 16-Bit Data Word

		S0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Syndrome		**S1**	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
Bits		**S2**	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
		S3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
S5	**S4**																	
0	0		NE	C0	C1	D	C2	D	D	3	C3	D	D	9	D	10	T	D
0	1		C4	D	D	11	D	T	T	D	D	7	T	D	T	D	D	15
1	0		C5	D	D	6	D	4	T	D	D	2	T	D	12	D	D	14
1	1		D	5	T	D	0	D	D	13	1	D	D	T	D	T	8	D

NE = no error Cn = check bit n in error T = three errors detected Number = single data bit in error D = two bits in error



* Necessary when inputting syndromes, otherwise leave open.

† Refer-to-discussion in "Other Modes of Operation" under Clearing SIL.

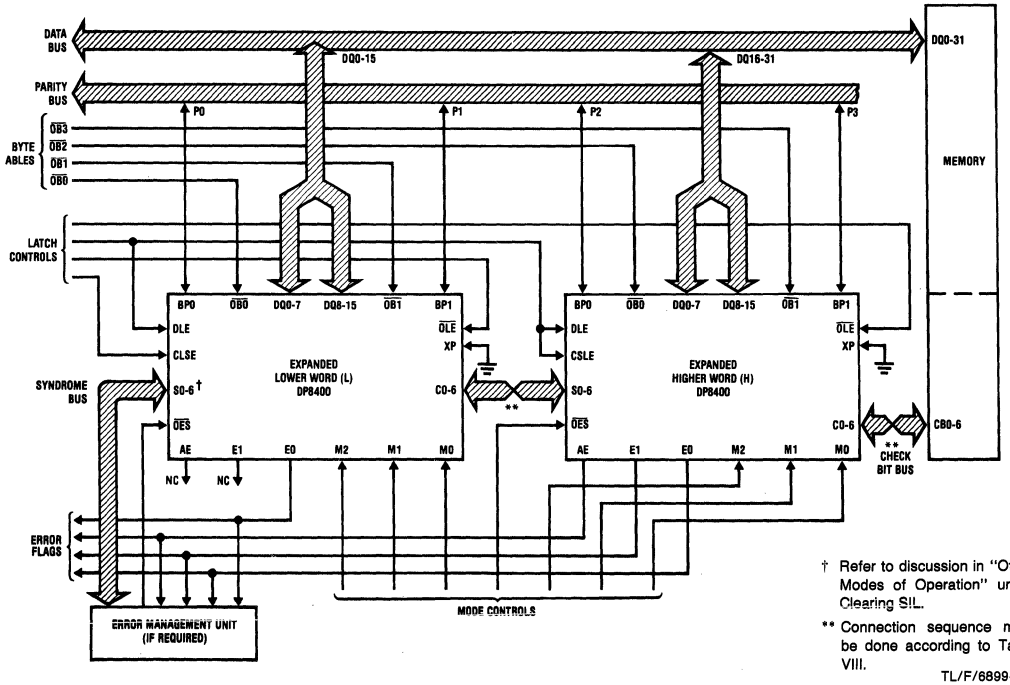
**20 ns max tpd1, 0

R = 2.7 kΩ

Necessary only when incorporating double complement correct. Otherwise connect C6 to GND through R.

TL/F/6899–9

FIGURE 4. 16-Bit Configuration Using One DP8400-2



† Refer to discussion in "Other Modes of Operation" under Clearing SIL.

** Connection sequence must be done according to Table VIII.

TL/F/6899–10

FIGURE 5. 32-Bit Error Detection and Correction

Modes of Operation (Continued)

READ) when both E1 and E2 become invalid. This is required when a memory correcting system employs the DP8400-2 with byte parity checking. The E1 and E2 pins flag the byte parity error in a memory WRITE cycle. When the DP8400-2 switches to a subsequent memory READ cycle, it requires t_{MCR} for E1 and E2 to be switched to flag any READ error(s).

Expanded Operation

32-BIT CONFIGURATION

Figure 5 shows how to connect two DP8400-2s in cascade to detect single and double-bit errors, and to correct single-data errors. The same circuit will also correct double-bit errors once a double-error has been detected, provided at least one error is a hard error. The lower chip L is in effect a slave to the higher chip H, which controls the memory check bits and error reporting. The check bit bus of L is reordered and connected to the syndrome bus of H, as shown in *Figure 5.*

In a Normal WRITE mode, referring to *Figures 13a, 13b,* and *13c,* the 7 check bits generated from the lower 16 bits (CGL) are transferred via the COL to the COB of L, provided $\overline{OLE}$ is high and M2 (R/$\overline{W}$) of L is low. These partial check bits from L then appear at SIL of H, so that with CSLE high, they combine with the 6 check bits generated in H with an overlap of one bit, to produce 7 check bits. With M2 (R/$\overline{W}$) of H low, these 7 check bits are output from COB to memory.

A READ cycle may consist of DETECT ONLY or DETECT THEN CORRECT, depending on the system approach. In both approaches, L writes its partial check bits, CGL, to H as in WRITE mode. H develops the syndrome bits from CGL, CGH and the 7 check bits read from memory in CIL. H then outputs from its error encoder (EE) if there is an error. If corrected data is required, H already knows if it has a single-data error from its syndrome bits, but if not, it must transfer partial syndromes back to L. These partial syndromes PSH, (CGH XOR-ed with CIL), are stored in SOL of H. L must therefore change modes from WRITE to READ, while H outputs the partial syndromes from its SOB by setting $\overline{OES}$ low. The partial syndromes are fed into CIL of L and XOR-ed with CGL to produce syndrome bits at SGL. The data error decoder, DED, then corrects the error in L. The DED of H will already have corrected an error in the higher 16 bits. Only one error in 32 bits can be corrected as a single-data error, the chip with no error does not change the contents of its DIL when it is enabled in DOL. Table VI shows the 3 error flags of H, which become valid during the DETECT cycle. E0 of L becomes valid during the CORRECT cycle, so that the 4 flags provide complete error reporting.

TABLE VI. Error Flags After Normal READ (32-Bit Configuration)

AE (H)	E1 (H)	E0 (H)	E0 (L)*	Error Type
0	0	0	0	No error
1	1	0	0	Single-check bit error
1	1	1	0	Single-data bit error (H)
1	1	0	1	Single-data bit error (L)
1	0	0	0	Double-bit error
All Others				Invalid conditions

*E0 (L) is valid after transfer of partial syndromes from higher to lower

Equations for 32-bit expansion:

$$t_{DCB32} = t_{DCB16} + t_{SCB16}$$

$$t_{DEV32} = t_{DCB16} + t_{SEV16}$$

$$t_{DCD32}\ (\text{High Chip}) = t_{DCB16} + t_{SCD16}$$

$$t_{DCD32}\ (\text{Low Chip}) = t_{DCB16} + t_{BR}{}^{*} + t_{CCD16}$$

*t_{BR}: Bus reversing time (25 ns)

32-BIT MATRIX

Table VII shows a 32-bit matrix using two DP8400-2s in cascade as in *Figure 5.* This is one of 12 matrices that work for 32 bits. The matrix for bits 0 to 15 (lower chip) is the matrix of Table IV for 16-bit configuration, with row 6 always "0". The matrix for bits 16 to 31 (higher chip) uses the same row combinations but interchanged, for example, the 3rd row (row 2) of L matrix is the same as the 6th row (row 5) of the H matrix. This means row 5 of H is in fact check bit 2 of H. Thus, the 6th row (row 5) combines generated check bit 5 (CG5) of L and generated check bit 2 of H. Check bit 5 of L therefore connects to the syndrome bit 2 (CG2) of H, and the composite generated check bit is written to check bit 2 of memory. Thus C2 performs a parity check on bits 0, 1, 2, 4, 5, 6, 8, 12, 13, 14, of L, and bits 16, 19, 20, 24, 26, 28, 29, 30, 31, of H. CG2 and CG3 generate odd parity, so that CG5 of L generates even parity which combines with CG2 of H generating odd parity. CG3 of L and CG3 of H both generate odd parity causing C3 to memory to represent even parity. Only 6 check bits are generated in each chip, the 7th (CG6) is always zero with XP grounded. Thus CG6 of L combines with CG0 of H so that C0 to memory is the parity of bits 18, 19, 20, 21, 22, 23, 25, 26, 27, 29, 30, 31. Similarly C6 to memory is only CG2 of L. The 7 composite generated check bits of H can now be written to memory.

When reading data and check bits from memory, CG6–CG0 of L are combined with CG6–CG0 of H in the same combination as WRITE. Memory check bits are fed into C6–C0 of H and compared with the 7 combined parity bits in H, to

TABLE VII. Data Bit Error To Syndrome-Generate Matrix (32-Bit Configuration)

	L															H																
										1	1	1	1	1	1																	DQ0-31
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	
0	0	0	1	1	1	1	1	0	1	1	1	0	1	1	1		0	0	0	1	0	0	1	0	1	1	0	1	0	1	1	1	1
1	0	0	0	1	0	0	1	0	1	1	1	0	1	0	1	1	1	1	1	0	1	1	1	0	1	0	0	0	1	1	1	0	5
*2	1	0	0	1	1	0	0	0	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
SYNDROMES *3	0	1	1	0	0	0	0	1	1	1	1	0	1	0	1	1	0	1	1	0	0	0	0	1	1	1	1	0	1	0	1	1	3
4	1	1	0	0	0	1	0	1	1	0	0	1	0	1	0	1	1	1	0	0	0	1	0	1	1	0	0	1	0	1	0	1	4
5	1	1	1	0	1	1	1	0	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	0	1	0	1	0	1	1	1	1	2
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0
	4	8	9	7	5	1	3	9	E	B	D	3	C	7	F	F	2	A	A	1	2	2	3	8	B	9	8	1	A	3	B	9	HEX
	3	3	2	0	2	3	2	1	3	0	0	1	2	3	2	1	3	1	4	6	6	5	4	5	3	4	6	5	2	7	6	7	

GENERATED CHECK BITS

TL/F/6899–11

*CG2, CG3 generate odd parity

Expanded Operation (Continued)

TABLE VIII. Check Bit Port To Syndrome Port Interconnections For Expansion To 32 Bits

		LS		LC	HS		HC		
Syndrome I/O to Management	S0	0		0	1		1	C0	Check Bit I/O to Memory
	S1	1		1	5		5	C1	
	S2	2		2	6		6	C2	
	S3	3		3	3		3	C3	
	S4	4		4	4		4	C4	
	S5	5		5	2		2	C5	
	S6	6		6	0		0	C6	

TABLE IX. Syndrome Decode To Bit In Error For 32-Bit Data Word

Syndrome Bits			S0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
			S1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
			S2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
			S3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
S6	S5	S4																	
0	0	0		NE	C0	C1	D	C2	D	D	3	C3	D	D	9	D	10	T	D
0	0	1		C4	D	D	11	D	T	T	D	D	7	17	D	T	D	D	15
0	1	0		C5	D	D	6	D	4	T	D	D	2	28	D	12	D	D	14
0	1	1		D	5	16	D	0	D	D	13	1	D	D	24	D	T	8	D
1	0	0		C6	D	D	22	D	T	T	D	D	25	18	D	T	D	D	T
1	0	1		D	27	21	D	T	D	D	T	23	D	D	T	D	T	T	D
1	1	0		D	19	20	D	T	D	D	T	26	D	D	30	D	T	T	D
1	1	1		T	D	D	29	D	T	T	D	D	31	T	D	T	D	D	T

NE = no error Cn = check bit n in error T = three errors detected
Number = single data bit in error D = two bits in error

produce 7 syndrome bits S6–S0. H can now determine if there is any error, and if it has a single-data error, it can locate it and correct it without transferring partial syndromes to L. As an example of a DETECT cycle, CG5 of L combines with CG2 of H and is compared in H with memory check bit 2.

If L is now set to mode 4, Normal READ, and $\overline{OES}$ of H is set low, the partial syndromes of H (CG6–CG0 of H XOR-ed with C6–C0 of H) are transferred and shifted to L. L receives these partial syndromes (S6–S0 of H) as check bit inputs C2, C1, C4, C3, C5, C0, C6 respectively, and compares them with CG6–CG0 respectively, to produce syndrome bits S6–S0. L now decodes these syndromes to correct any single-data error in data bits 0 to 15. For example, partial syndrome bit 2 of H combines with generated check bit 5 of L to produce syndrome bit 5 in L. An error in data bit 10 will create syndrome bits in L as 0001101 from S6–S0, and these will appear on S6–S0 of L with $\overline{OES}$ low. An error in H will appear as per the H matrix. For example, an error in bit 16 will cause S6–S0 of L to be 0110010.

If $\overline{OES}$ of L is set low, this syndrome combination appears on pins S6 to S0. For errors in bits 0 to 15, the syndrome outputs will be according to Table VII. For errors in bits 16 to 31, the syndrome outputs from L will still be according to Table VII due to the shifting of partial syndrome bits from H to L. The syndrome outputs from L are unique for each of the possible 32 bits in error.

If there is a check bit error, only one syndrome bit will be high. For example, if C5 is in error, then S1 of L will be high. For double-errors, an even number of syndrome bits will be high, derived from XOR-ing the two single-bit error syndromes. As mentioned previously, this is only one of the 12 approaches to connecting two chips for 32 bits, 6 of which are mirror images.

Table VIII depicts the exact connection for 32-bit expansion. LS equals syndrome bits of L. LC equals check bits of L. HS equals syndrome bits of H. HC equals check bits of H. Syndrome bits S0 to S6 of L are connected to system syndrome bits S0 to S6. LC and HS columns are lined together showing the check bit port of L connected to the syndrome port of H in the exact sequence as shown in Table VIII. For example, check bit C0 of L is connected to the syndrome bit S1 of H, and check bit C6 of L is connected to the syndrome bit S0 of H. Check bits of H are connected to the system check bits in the order shown. Check bit C1 of H is connected to the system check bit C0.

EXPANSION FOR DATA WORDS REQUIRING 8 CHECK BITS

For 16-bit and 32-bit configurations, XP is set permanently low. In 48-bit or 64-bit configurations, XP is either set permanently to V_{CC} or left open, according to Table X, to provide 8 check bits and syndrome bits.

TABLE X. XP: Expansion Status

XP	Status	Data Bus
0V	BP0 and BP1 are byte parity I/O CG6 = 0	< 40 Bits
Open	No byte parity I/O, CG6 and CG7 = word parity	≥ 40 Bits
V_{CC}	No byte parity I/O, CG6 and CG7 = 0	≥ 40 Bits

48-BIT EXPANSION

Three DP8400-2s are required for 48 bits, with the higher chip using all 8 of its check bits to the memory. No byte parity is available for 48 to 64 bits. XP of all three chips must be at V_{CC}. The three chips are connected in cascade as in

Expanded Operation (Continued)

TABLE XI. Check Bit Port To Syndrome Port Interconnections For Expansion To 48 Bits

		LL S		LL C	LH S		LH C	HL S		HL C		
Syndrome I/O to Management	S0	0		0	1		1	6		6	C0	Check Bit I/O to Memory
	S1	1		1	5		5	1		1	C1	
	S2	2		2	6		6	4		4	C2	
	S3	3		3	3		3	7		7	C3	
	S4	4		4	4		4	2		2	C4	
	S5	5		5	2		2	3		3	C5	
	S6	6		6	0		0	5		5	C6	
	S7	7		7	7		7	0		0	C7	

For example: S0 of LL is connected to system syndrome S0. C0 of LL is connected to S1 of LH. C1 of LH is connected to S6 of HL. C6 of HL is connected to system check bit C0.

TABLE XII. Syndrome Decode To Bit In Error For 48-Bit Data Word

			S0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Syndrome			S1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
Bits			S2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
			S3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
S7	S6	S5	S4																
0	0	0	0	NE	C0	C1	D	C2	D	D	3	C3	D	D	9	D	10	T	D
0	0	0	1	C4	D	D	11	D	T	T	D	D	7	17	D	T	D	D	15
0	0	1	0	C5	D	D	6	D	4	T	D	D	2	28	D	12	D	D	14
0	0	1	1	D	5	16	D	0	D	D	13	1	D	D	24	D	T	8	D
0	1	0	0	C6	D	D	22	D	T	T	D	D	25	18	D	T	D	D	T
0	1	0	1	D	27	21	D	32	D	D	T	23	D	D	T	D	T	T	D
0	1	1	0	D	19	20	D	33	D	D	T	26	D	D	30	D	T	T	D
0	1	1	1	44	D	D	29	D	T	40	D	D	31	T	D	T	D	D	T
1	0	0	0	C7	D	D	T	D	T	43	D	D	T	T	D	T	D	D	T
1	0	0	1	D	T	35	D	T	D	D	T	T	D	D	T	D	T	T	D
1	0	1	0	D	T	41	D	39	D	D	T	T	D	D	T	D	T	T	D
1	0	1	1	42	D	D	T	D	T	47	D	D	T	T	D	T	D	D	T
1	1	0	0	D	T	38	D	37	D	D	T	T	D	D	T	D	T	T	D
1	1	0	1	36	D	D	T	D	T	45	D	D	T	T	D	T	D	D	T
1	1	1	0	34	D	D	T	D	T	T	D	D	T	T	D	T	D	D	T
1	1	1	1	D	T	46	D	T	D	D	T	T	D	D	T	D	T	T	D

NE = no error Cn = check bit n in error T = three errors detected
Number = single data bit in error D = two bits in error

Figure 6, but with the HH chip removed. The error flags are as Table XV, but with AE (HH) and E1 (HH) becoming AE (HL) and E1 (HL), and E0 (HH) removed.

48-BIT MATRIX

The matrix for 48 bits is that for 64 bits shown (in Table XVI) but only using bits 0 to 47. This is one of many matrices for 48-bit expansion using the basic 16-bit matrix. The matrix shown uses 2 zeroes for CG6 and CG7, for all three chips, with XP set to V_{CC}. Other matrices may use CG6 and CG7 as word parity with XP open.

64-BIT EXPANSION

There are two basic methods of expansion to 64 bits, both requiring 8 check bits to memory, and four DP8400-2s. One is the cascade method of *Figure 6*, requiring no extra ICs. With this method partial check bits have to be transferred through three chips in the WRITE or DETECT mode, and partial syndrome bits transferred back through three chips in CORRECT mode. This method is similar to *Figure 5*, 32-bit approach. The connections between the check bit bus and syndrome bus for each of the chip pairs are shown in Table XIII.

The error flags of HH are valid during the DETECT cycle as in Table XV, and the other error flags are valid during the CORRECT cycle.

A faster method of 64-bit expansion shown in *Figure 7* requires a few extra ICs, but can WRITE in 50 ns, DETECT in 42 ns or DETECT THEN CORRECT in 90 ns. In the WRITE mode, all four sets of check bits are combined externally in the 8 74S280 parity generators. These generate 8 composite check bits from the system data, which are then enabled to memory. In the DETECT mode, again 8 composite check bits are generated, from the memory data this time, and compared with the memory check bits to produce 8 external syndrome bits. These syndrome bits may be OR-ed to determine if there is any error. By making the 74S280 outputs $\overline{\text{SYNDROMES}}$, then any bit low causes the 74S30 NAND gate to go high, giving any error indication. To correct the error, these syndrome bits are fed re-ordered into SIL of each DP8400-2 now set to mode 7B. This enables the syndromes directly to SG and then DED of each chip. One chip

Expanded Operation (Continued)

will output corrected data, while the other three output non-modified data (but still correct).

Equations for fast 64-bit expansion:

$t_{DCB64} = t_{DCB16} + t_{pd}$ (74S280) + t_{pd} (74S240)

$t_{DEV64} = t_{DCB16} + t_{pd}$ (74S280) + t_{pd} (74S30)

$t_{DCB64} = t_{DCB16} + t_{pd}$ (74S280) + t_{pd} (74ALS533) + t_{SCD16}

64-BIT MATRIX

With the 64-bit matrix shown in Table XVI, it is necessary to set at least one chip with CG6, CG7 non-zero. The highest chip, connected to data bits 48 to 63, has XP set open, so that its CG6 and CG7 are word parity. The syndrome word of the highest chip will now have either 5 or 7 syndrome bits high, but inside the chip CG6 and CG7 remove two of these in a READ so that the chip sees the normal 3 or 5 syndrome bits.

TABLE XIII. Check Bit Port To Syndrome Port Interconnections For Expansion To 64 Bits

		LL S		LL C	LH S		LH C	HL S		HL C	HH S		HH C		
Syndrome I/O to Management	S0	0		0	1		1	6		6	7		7	C0	Check Bit I/O to Memory
	S1	1		1	5		5	1		1	0		0	C1	
	S2	2		2	6		6	4		4	1		1	C2	
	S3	3		3	3		3	7		7	2		2	C3	
	S4	4		4	4		4	2		2	3		3	C4	
	S5	5		5	2		2	3		3	4		4	C5	
	S6	6		6	0		0	5		5	5		5	C6	
	S7	7		7	7		7	0		0	6		6	C7	

For example: S0 of LL is connected to system syndrome S0. C0 of LL is connected to S1 of LH. C1 of LH is connected to S6 of HL. C6 of HL is connected to S7 of HH. C7 of HH is connected to system check bit C0.

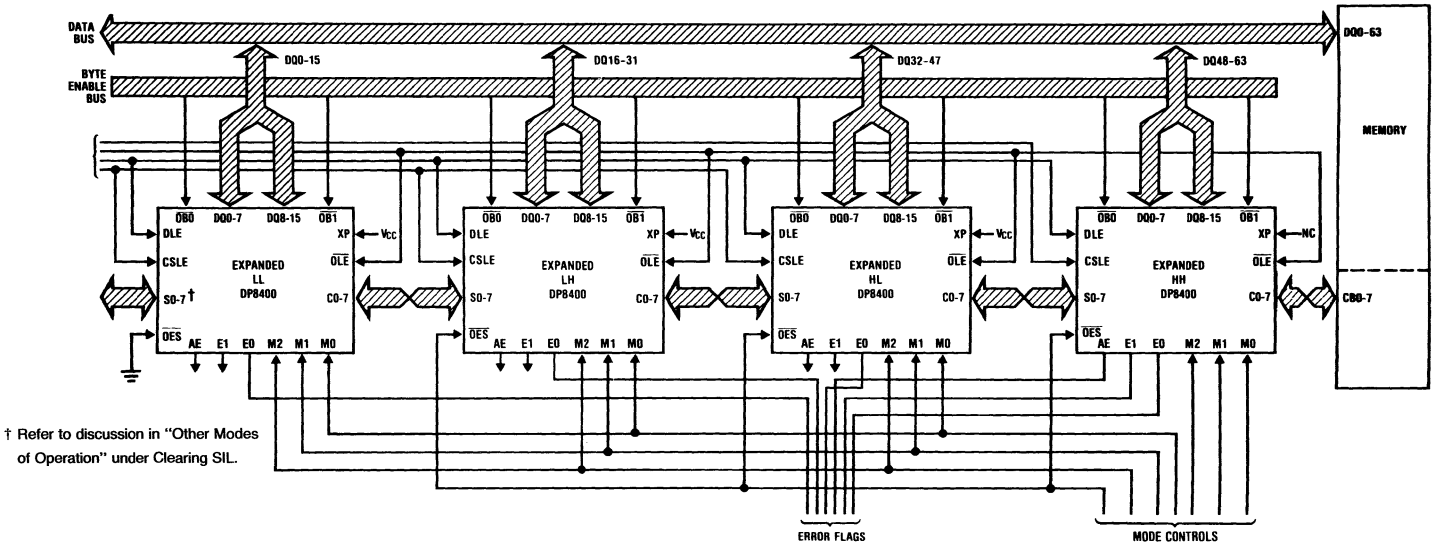
TABLE XIV. Syndrome Decode To Bit In Error For 64-Bit Data Word

			S0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Syndrome			**S1**	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
Bits			**S2**	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
			S3	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
S7	**S6**	**S5**	**S4**																
0	0	0	0	NE	C0	C1	D	C2	D	D	3	C3	D	D	9	D	10	T	D
0	0	0	1	C4	D	D	11	D	T	T	D	D	7	17	D	T	D	D	15
0	0	1	0	C5	D	D	6	D	4	T	D	D	2	28	D	12	D	D	14
0	0	1	1	D	5	16	D	0	D	D	13	1	D	D	24	D	T	8	D
0	1	0	0	C6	D	D	22	D	T	T	D	D	25	18	D	T	D	D	T
0	1	0	1	D	27	21	D	32	D	D	T	23	D	D	T	D	T	T	D
0	1	1	0	D	19	20	D	33	D	D	T	26	D	D	30	D	T	T	D
0	1	1	1	44	D	D	29	D	T	40	D	D	31	T	D	T	D	D	T
1	0	0	0	C7	D	D	T	D	T	43	D	D	T	T	D	T	D	D	51
1	0	0	1	D	T	35	D	T	D	D	57	T	D	D	58	D	T	T	D
1	0	1	0	D	T	41	D	39	D	D	59	T	D	D	T	D	T	T	D
1	0	1	1	42	D	D	55	D	T	47	D	D	T	T	D	T	D	D	63
1	1	0	0	D	T	38	D	37	D	D	54	T	D	D	52	D	T	T	D
1	1	0	1	36	D	D	50	D	T	45	D	D	60	T	D	T	D	D	62
1	1	1	0	34	D	D	53	D	T	T	D	D	48	T	D	T	D	D	61
1	1	1	1	D	49	46	D	T	D	D	T	T	D	D	T	D	56	T	D

NE = no error Cn = check bit n in error T = three errors detected
Number = single data bit in error D = two bits in error

TABLE XV. Error Flags After Normal READ (Any 64-Bit Configuration)

AE (HH)	E1 (HH)	E0 (HH)	E0 (HL)	E0 (LH)	E0 (LL)	Error Type
0	0	0	0	0	0	No error
1	1	0	0	0	0	Single-check bit error
1	1	1	0	0	0	Single-data bit error in HH
1	1	0	1	0	0	Single-data bit error in HL
1	1	0	0	1	0	Single-data bit error in LH
1	1	0	0	0	1	Single-data bit error in LL
1	0	0	0	0	0	Double-error



TL/F/6899–12

FIGURE 6. Cascade Expansion Using No Extra ICs (64-Bit Configuration)

TABLE XVI. Data Bit Error To Syndrome-Generate Matrix (64-Bit Configuration)

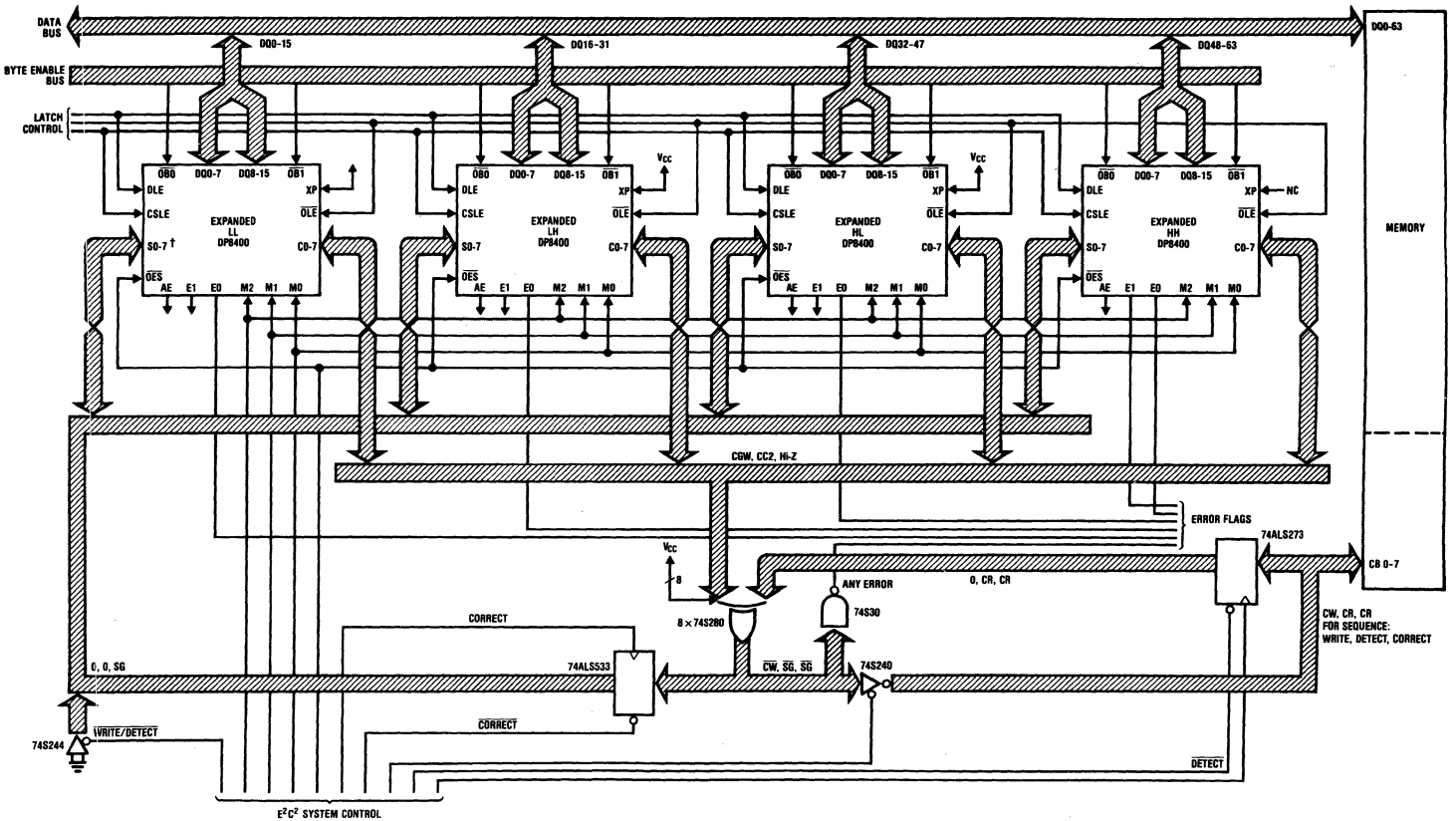
	LL								LH								HL								HH								
	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	31	30	29	28	27	26	25	24	47	46	45	44	43	42	41	40	
	39	38	37	36	35	34	33	32	23	22	21	20	19	18	17	16	55	54	53	52	51	50	49	48	63	62	61	60	59	58	57	56	
0	0	0	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0	0	0	1	0	0	1	0	1	1	1	0	1	0	1	1	7
1	0	0	0	1	0	0	1	0	1	1	0	1	0	1	1	1	1	1	1	0	1	1	1	0	0	0	0	1	0	0	1	0	0
2	1	0	0	1	1	0	0	0	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	1
3	0	1	1	0	0	0	0	1	1	1	1	0	1	0	1	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	2
4	1	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	1	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	3
5	1	1	1	0	1	1	1	0	1	0	0	1	1	1	0	0	0	1	1	1	0	0	0	0	0	1	1	0	0	0	0	1	4
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	1	1	1	0	1	1	1	5
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	1	1	1	0	1	1	1	6
	4	8	9	7	5	1	3	9	E	B	D	3	C	7	F	F	2	A	A	1	2	2	3	8	B	9	8	1	A	3	B	9	HEX
	3	3	2	0	2	3	2	1	3	0	0	1	2	3	2	1	3	1	4	6	6	5	4	5	3	4	6	5	2	7	6	7	

SYNDROMES

DQ0-63

GENERATED CHECK BITS

TL/F/6899–13



TL/F/6899–14

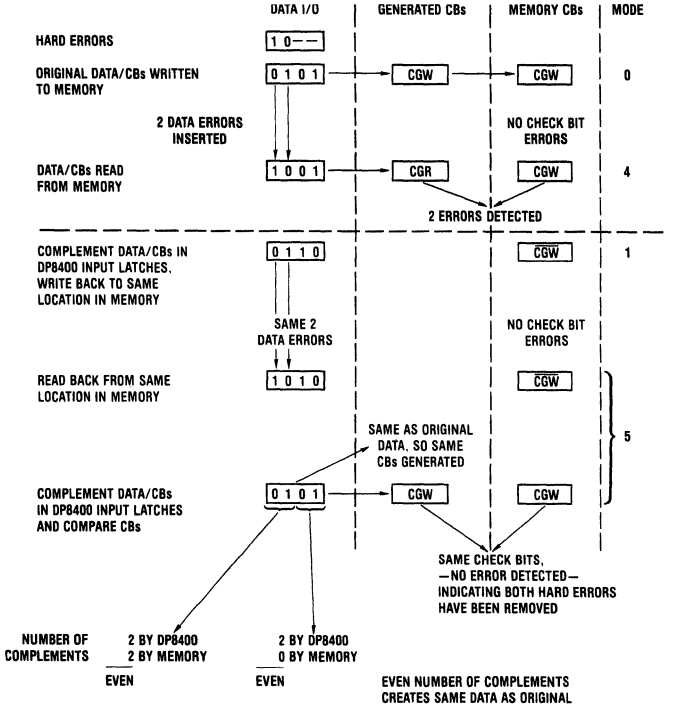
FIGURE 7. Parallel Expansion (Fast 64-Bit Configuration)

Other Modes Of Operation

DOUBLE ERROR CORRECTION, USING THE DOUBLE-COMPLEMENT APPROACH

The DP8400-2 can be made to correct two errors, using no extra ICs or check bits, if at least one of the two errors detected is a hard error. This does require an extra memory WRITE and READ. Nevertheless, if a permanent failure exists, and an additional error occurs (creating two errors), both errors can be corrected, thereby saving a system crash.

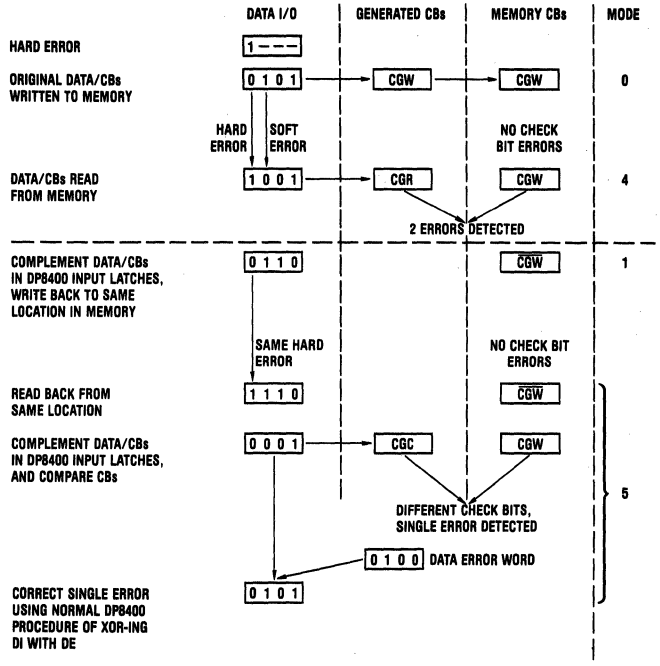
Once a double error has been detected, the system puts the DP8400-2 in COMPLEMENT mode by setting M0 high. First a WRITE cycle is required and M2 is set low, putting the chip in mode 1, Table III, (COMPLEMENT WRITE), so that the contents of DIL are complemented into DOL, and the contents of CIL complemented into COL. $\overline{OB0}$ and $\overline{OB1}$ are set low so that complemented data and check bits can be written back to the same location of memory. Writing back complemented data to a location with a hard error forces the error to repeat itself. For example, if cell N of a particular location is jammed permanently high, and a low is written to it, a high will be read. However, when the data is complemented a low is again written, so that a high is read back for the second time. After a second READ (this second READ is a COMPLEMENT READ) of the location, data and check bits from the memory are recomplemented, so that bit N now contains a low. In other words, the error in bit N has corrected itself, while the other bits are true again. If there are two hard errors in a location, both are automatically corrected and the DP8400-2 detects no error on COMPLEMENT READ, as in *Figure 8a*. *Figure 8b* also shows that if one error is soft, the hard error will disappear on the second READ and the DP8400-2 corrects the soft error as a single-error. Therefore, in both cases, the DOL contains corrected data, ready to be enabled by $\overline{OB0}$ and $\overline{OB1}$. A WRITE to memory at this stage removes the complemented data written at the start of the sequence.



TL/F/6899–15

FIGURE 8a. Double Error Correct Complement Hard Error Method — 2 Hard Errors In Data Bits

Other Modes of Operation (Continued)



TL/F/6899–16

FIGURE 8b. Double Error Correct Complement Hard Error Method — 1 Hard Error, 1 Soft Error In Data Bits

The examples shown in *Figures 8a* and *8b* are for 4 data bits. This approach will work for any number of data bits, but for simplicity these examples show how complementing twice corrects two errors in the data bits. The double COMPLEMENT approach also works for any two errors providing at least one is hard. In other words, one data-bit error and one check bit error, or two check bit errors are also corrected if one or both are hard. At the end of the COMPLEMENT READ cycle, the error flags indicate whether the data was correctable or not, as shown in Table XVII. If both the errors were soft, then the data was not correctable and the error flags indicate this.

This approach is ideal where double errors are rare but may occur. To avoid a system crash, a double-error detect now causes the system to enter a subroutine to set the DP8400-2 in COMPLEMENT mode. This method is also useful in bulk-memory applications, where RAMs are used with known cell failures, and is applicable in 16, 32, 48 or 64-bit configuration. In the 16-bit configuration, modes 1 and 5 of Table III are used. In the 32-bit expanded configuration, modes 1, 5 and 5 are used for the highest chip, and modes 3, 3 and 4 for the lower chip for WRITE, DETECT, and CORRECT. With the lower chip it is necessary to wrap around DOL (after latching its contents in mode 3), back to DIL and perform a Normal READ in mode 4 in the lower chip.

TABLE XVII. ERROR FLAGS AFTER COMPLEMENT READ (MODE 5)

AE	E1	E0	Error Type
0	0	0	Two hard errors
1	1	0	One hard error, one soft check bit error
1	1	1	One hard error, one soft data bit error
1	0	0	Two soft errors, not corrected

Other Modes of Operation (Continued)

DOUBLE-ERROR CORRECT WITH ERROR LOGGING

Figures 4 and *5* show the E²C² syndrome port connected to an error management unit (EMU). This scheme stores syndromes and the address of locations that fail, thereby logging the errors. Subsequent errors in a memory location that has already stored syndromes in the EMU, can then be removed by injecting the stored syndromes of the first error. To save the addresses and syndromes when power to the EMU is removed, it is necessary to be able to transfer information via the E²C² syndrome port to the processor data bus. This is also useful for logging the errors in the processor. Transfer in the opposite direction is also necessary.

DATA BUS TO SYNDROME BUS TRANSFER

This is necessary when transferring syndrome information to the error management unit, which is connected to the external syndrome bus. First, data to make CG = 0 (all data bits high) must be latched in DIL. Then in mode 2, data is fed to CIL, XOR-ed with 0, and output via SOL with $\overline{OES}$ low to the syndrome bus. Data is therefore fed directly from the system to the syndrome bus, and this cycle may be repeated as long as DLE is kept low, forcing CG to remain zero.

SYNDROME BUS TO DATA BUS TRANSFER

This is important when information in the error logger or error management unit has to be read. The DP8400-2 is set to mode 6B with $\overline{OES}$ high, and with $\overline{OB0}$, $\overline{OB1}$ and $\overline{OLE}$ low. If CSLE is high, the syndrome bus and check bit bus data appear on the lower and upper bytes of the data bus to be read by the system. Also E1 and E0 values that were valid when mode 6 was entered, appear on DQ7 and DQ15.

FULL DIAGNOSTIC CHECK OF MEMORY

Using mode 2, it is possible to transfer the upper byte of the data bus directly to the CIL, with CSLE high, without affecting DIL. These simulated check bits then appear on the check bit bus with $\overline{OLE}$ low, which also causes the previously latched contents of DIL to transfer to DOL. By enabling $\overline{OB0}$ and $\overline{OB1}$ data can be written to memory with the simulated check bits. A Normal READ cycle can then aid the system in determining that the memory bits are functioning correctly, since the processor knows the check bits and data it sent to the E²C². Another solution is to put the E²C² in mode 6 and read the memory check bits directly back to the processor.

SELF-TEST OF THE E²C² ON-CARD

Again using mode 2, data written from the processor data bus upper byte to CIL may be stored in CIL, by taking CSLE low. Next, a mode 0 WRITE can be performed and the user generated data can be latched in the DP8400-2 input latches (DLE held low). Now the user may perform a normal mode 4 READ. This will in effect be a Diagnostic READ of the user generated data and check bits without using the external memory. Thus by reading corrected data in mode 4, and by reading the generated syndromes, and error flags E0 and E1, the DP8400-2 can be tested completely on-card without involving memory.

MONITORING GENERATED SYNDROMES AND MEMORY CHECK BITS

Mode 6A enables SG0–SG6 onto DQ0–DQ6, and CIL0–CIL6 onto DQ8–DQ14, provided $\overline{OLE}$, $\overline{OB0}$ and $\overline{OB1}$ are low. Also the two error flags, E1 and E0 (latched from the previous READ mode), appear on DQ7 and DQ15. This may be used for checking the internal syndromes, for reading the memory check bits, or for diagnostics by checking the latched error flags.

CLEARING SIL

In the 16-bit only configuration, or the lower chip of expanded configurations, and in various modes of operation in the higher expanded chips, it is required that SIL be maintained at zero. At power-up initialization, both SIL and DIL are reset to all low. If $\overline{OES}$ is kept low, SIL will remain reset because CSLE is inhibited to SIL. Another method is to keep $\overline{OES}$ always high and the syndrome bus externally set low, or set low whenever CSLE can be used to clear SIL.

Mode 7A also forces the SIL to be cleared whenever CSLE occurs, and also these zero syndromes go to the internal syndrome bus SG. This puts the DP8400-2 in a PASS-THROUGH mode where the DIL contents pass to DOL and CIL contents to COL, if $\overline{OLE}$ is low.

POWER-UP INITIALIZATION OF MEMORY

Both SIL and DIL are reset low at power-up initialization. This facilitates writing all zeroes to the memory data bits to set up the memory. The check bits corresponding to all-zero data will appear on the check bit bus if the DP8400-2 is set to mode 0 and $\overline{OLE}$ is set low. All-zero data appears on the data bus when $\overline{OB0}$ and $\overline{OB1}$ are also set low. The system can now write zero-data and corresponding check bits to every memory location.

BYTE WRITING

Figure 14a shows the block diagram of a 16-bit memory correction system consisting of a DP8400-2 error correction chip and a DP8409A DRAM controller chip. There are 12 control signals associated with the interface. Six of the signals are standard DP8400-2 input signals, three are standard DP8409A input signals, and three are buffer control signals. The buffer control signals, $\overline{PBUF0}$ and $\overline{PBUF1}$, control when data words or bytes from the DP8400-2/memory data bus are gated to the processor bus and when data words or bytes from the processor are gated to the DP8400-2/memory data bus.

When the processor is reading or writing bytes to memory, words will always be read or written by the DP8400-2 and DP8409A error correction and DRAM controller section. The High Byte Enable and Address Data Bit Zero signals from the processor should control the byte transfers via the ocal bus transceiver signals $\overline{PBUF0}$ and $\overline{PBUF1}$. The buffer control signal, $\overline{DOUTB}$, controls when data from memory is gated onto the DP8400-2/memory data bus.

Figure 14b shows the timing relationships of the 12 control signals, along with the DP8400-2/memory data bus and some of the DRAM control signals ($\overline{RAS}$ and $\overline{CAS}$). RGCK is the $\overline{RAS}$ generator clock of the DP8409A which is used in Mode 1 (Auto Refresh mode), along with being the system clock.

Other Modes of Operation (Continued)

Having two separate byte enable pins, $\overline{OB0}$ and $\overline{OB1}$, it is easy to implement byte writing using the DP8400-2. First it is necessary to read from the location to which the byte is to be written. To do this the DP8400-2 is put in normal Read mode (Mode 4), which will detect and correct a single bit error. $\overline{WIN}$ is kept high and $\overline{RASIN}$ is pulled low, causing the DP8409A, now in Mode 5 (Auto Access mode), to start a read memory cycle. The data word and check bits from memory are then enabled onto the DP8400-2/memory data bus by pulling $\overline{DOUTB}$ low. The data and check bits are valid on the bus after the $\overline{RASIN}$ to $\overline{CAS}$ time (t_{RAC}) plus the column access time (t_{CAC}) of the particular memories used. DLE, CSLE can then be pulled low in order to latch the memory data into the input latches of the DP8400-2. $\overline{OLE}$ can be pulled low to enable the corrected memory word, or the original memory word if no error was present, into the data output latches. Following this, $\overline{DOUTB}$ can be pulled high to disable the memory data from the DP8400-2/memory data bus. The corrected memory word will be available at the data output latches "t_{DCD16}" after the memory word was available at the data input latches. Once the corrected data is available at the output latches $\overline{OLE}$ can be pulled high to latch the corrected data. Also DLE and CSLE can be pulled high in order to enable the input data latches again.

Now the DP8400-2 can be put into a write cycle (Mode 0 = M2 = Low). At this time the byte to be written to memory and the other byte from memory can be enabled onto the DP8400-2/memory data bus ($\overline{OB0}$, $\overline{PBUF1}$ or $\overline{OB1}$, $\overline{PBUF0}$ go low). DLE, CSLE can now transition low to latch the new memory word into the data input latch. $\overline{OLE}$ is pulled low to enable the output latches. When the new checkbits are valid, t_{DCB16} after the data word is valid on the DP8400-2/memory data bus, $\overline{OLE}$ and $\overline{DLE}$ can be pulled high to latch the new memory word into the output latches, and then $\overline{WIN}$ can be pulled low to write the data into memory. $\overline{RASIN}$ should be held low long enough to cause the new data and check bits to be stored into memory ($\overline{WIN}$ data hold time).

Also a READ-MODIFY-WRITE cycle was performed, taking approximately 40% longer than a normal memory WRITE cycle. A READ and then a WRITE memory cycle could have been used in the above example but it would have taken longer.

Buffers are used in this system (74ALS244) to keep the Data Out and Data In of the memory IC's from conflicting with each other during Read-Modify-Write cycles.

A byte READ from memory is no different from a normal READ. This approach may be used for a 16-bit processor using byte writing, or an 8-bit processor using a 16-bit memory to reduce the memory percentage of check bits, or with memory word sizes greater than two bytes.

An APP NOTE (App Note 387) has been written detailing an Error Correcting Memory System using the DP8409A or DP8419 (Dynamic RAM Controller) and the DP8400-2 interfaced to a National Semiconductor Series 32000 CPU. See this App Note for further system details and considerations.

BEYOND SINGLE-ERROR CORRECT

With the advent of larger semiconductor memories, the frequency of the soft errors will increase. Also some memory system designers may prefer to buy less expensive memories with known cell, row or column failures, thus, more hard errors. All this means that double-error correct, triple-error detect capability, and beyond will become increasingly important. The DP8400-2 can correct two errors, provided one or both are hard errors, with no extra components, using the double complement approach. There are two other approaches to enhance reliability and integrity. One is to use the error management unit to log errors using the syndrome bus, and then to output these syndromes, when required, back to the DP8400-2.

DOUBLE SYNDROME DECODING

The other approach takes advantage of the Rotational Syndrome Word Generator matrix. This matrix is an improvement of the Modified Hamming-code, so that if, on a second DP8400-2, the data bus is shifted or rotated by one bit, and 2 errors occur, the syndromes for this second chip will be different from the first for any 2 bits in error. Both chips together output a unique set of syndromes for any 2 bits in error. This can be decoded to correct any 2-bit error. This is not possible with other Modified Hamming-code matrices.

Absolute Maximum Ratings (Note 1)

Storage Temperature Range	−65°C to +150°C
Supply Voltage, V_{CC}	7V
Input Voltage	5.5V
Output Sink Current	50 mA
Maximum Power Dissipation at 25°C	
Molded Package	3269 mW
Lead Temperature (Soldering, 10 seconds)	300°C

*Derate molded package 26.2 mW/°C above 25°C.

Operating Conditions

	Min	Max	Units
V_{CC}, Supply Voltage	4.75	5.25	V
T_A, Ambient Temperature	0	70	°C

Electrical Characteristics (Note 2) V_{CC} = 5V ±5%, T_A = 0°C to 70°C unless otherwise noted

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IL}	Input Low Threshold				0.8	V
V_{IH}	Input High Threshold		2.0			V
V_C	Input Clamp Voltage	V_{CC} = Min, I_C = −18 mA		−0.8	−1.5	V
I_{IH}	Input High Current	V_{IN} = 2.7V		1	160	μA
I_{IH} (XP)	Input High Current	V_{CC} = Max, XP = 5.25V		2.5	4.5	mA
I_{IL} (XP)	Input Low Current	V_{CC} = Max, XP = 0V		−2.5	−4.5	mA
I_{IL} (BP0/C7)	Input Low Current	V_{CC} = Max, V_{IN} = 0.5V		−100.0	−500	μA
I_{IL} (BP1/S7)	Input Low Current	V_{CC} = Max, V_{IN} = 0.5V		−100.0	−500	μA
I_{IL} (CSLE)	Input Low Current	V_{CC} = Max, V_{IN} = 0.5V		−150.0	−750	μA
I_{IL} (DLE)	Input Low Current	V_{CC} = Max, V_{IN} = 0.5V		−200.0	−1000	μA
I_{IL}	Input Low Current	V_{CC} = Max, V_{IN} = 0.5V		−50.0	−250	μA
I_I	Input High Current (Max)	V_{IN} = 5.5V (Except XP Pin)			1.0	mA
V_{OL}	Output Low Voltage	I_{OL} = 8 mA (Except BP0, BP1) I_{OL} = 4 mA (BP0, BP1 Only)		0.3 0.3	0.5 0.5	V V
V_{OH}	Output High Voltage	I_{OH} = −100 μA I_{OH} = −1 mA	2.7 2.4	3.2 3.0		V V
I_{OS}	Output Short Current (Note 3)	V_{CC} = Max		−150	−250	mA
I_{CC}	Supply Current	V_{CC} = Max		220	300	mA
C_{IN} (I/O)	Input Capacitance All Bidirectional Pins	Note 4		8.0		pF
C_{IN}	Input Capacitance All Unidirectional Input Pins	Note 4		5.0		pF

Note 1: "Absolute Maximum Ratings" are the values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: All typical values are for T_A = 25°C and V_{CC} = 5.0V

Note 3: Only one output at a time should be shorted.

Note 4: Input capacitance is guaranteed by periodic testing. F test = 10 kHz at 300 mV, T_A = 25°C.

Note 5: All switching parameters measured from 1.5V of input to 1.5V of output. Input pulse amplitude 0V to 3V, $t_r = t_f$ = 2.5 ns.

DP8400-2 Switching Characteristics (Note 5)

V_{CC} = 5.0V ±5%, T_A = 0°C to 70°C, C_L = 50 pF

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{DCB16}	Data Input Valid to Check Bit Valid	*Figure 9b*		29	40	ns
t_{DEV16}	Data Input to Any Error Valid	*Figures 10b, 11b* (Note 1)		21	31	ns
t_{DCD16}	Data Input Valid to Corrected Data Valid	*Figure 10b*, $\overline{OB0}$, $\overline{OB1}$ Low		44	61	ns
t_{DSI}	Data Input Set-Up Time Before DLE, CSLE H to L	*Figures 10b, 13d*	10	5		ns
t_{DHI}	Data Input Hold Time After DLE, CSLE H to L	*Figures 10b, 13d*	10	5		ns
t_{DSO}	Data Input Set-Up Time Before $\overline{OLE}$ L to H	*Figure 10b*	10	5		ns
t_{DHO}	Data Input Hold Time After $\overline{OLE}$ L to H	*Figure 10b*	10	5		ns
t_{DE0}	Data in Valid to E0 Valid	*Figures 9b, 10b, 13d*		36	55	ns
t_{DE1}	Data in Valid to E1 Valid	*Figures 9b, 10b, 13d*		43	55	ns
t_{IEV}	DLE, CSLE High to Any Error Flag Valid (Input Data Previously Valid)	*Figure 10b*		28	45	ns
t_{IEX}	DLE, CSLE High to Any Error Flag Invalid	*Figures 9b, 10b*		38	60	ns
t_{ILE}	DLE, CSLE High Width to Guarantee Valid Data Latched	*Figures 10b, 13d*	20			ns
t_{OLE}	$\overline{OLE}$ Low Width to Guarantee Valid Data Latched	*Figure 13d*	20			ns
t_{ZH}	High Impedance to Logic 1 from $\overline{OB0}$, $\overline{OB1}$, $\overline{OES}$ M2 H to L	*Figures 9b, 10b, 13d*		22	36	ns
t_{HZ}	Logic 1 to High Impedance from $\overline{OB0}$, $\overline{OB1}$, $\overline{OES}$, M2 L to H	*Figures 9b, 10b, 13d,*		38	55	ns
t_{ZL}	High Impedance to Logic 0 from $\overline{OB0}$, $\overline{OB1}$, $\overline{OES}$ M2 H to L	*Figures 9b, 10b, 13d*		19	35	ns
t_{LZ}	Logic 0 to High Impedance from $\overline{OB0}$, $\overline{OB1}$, $\overline{OES}$, M2 H to L	*Figures 9b, 10b, 13d*		15	25	ns
t_{PPE}	Byte Parity Input Valid to Parity Error Flags Valid	*Figure 9b*		16	27	ns
t_{DPE}	Data In Valid to Parity Error Flags Valid	*Figures 9b, 13d*		27	55	ns
t_{DCP}	Data in Valid to Corrected Byte Parity Output Valid	*Figure 9b*		44	61	ns

Note 1: This parameter value holds given that an error occurred. In the case of no error, t_{DEV16} will be max of 80 ns.

DP8400-2 Switching Characteristics (Continued) (Note 5)

V_{CC} = 5.0V ±5%, T_A = 0°C to 70°C, C_L = 50 pF

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{NMR}	New Mode Recognize Time	*Figure 10b*		22	35	ns
t_{CDV}	Mode Valid to Complement Data Valid	*Figure 11b*		34	50	ns
t_{CCV}	Mode Valid to Complement Check Bit Valid	*Figure 11b*		30	45	ns
t_{SCB}	Syndrome Input Valid to Check Bit Valid	*Figure 13d*		20	35	ns
t_{SEV}	Syndrome Input Valid (CGL) to Any Error Valid	*Figure 13d*		17	27	ns
t_{SCD}	Syndrome Inputs Valid to Corrected Data Valid	*Figure 13d*		35	50	ns
t_{DSB}	Data Input Valid to Syndrome Bus Valid	*Figure 13d*, $\overline{OES}$ Low		28	46	ns
t_{CSB}	Check Bit Inputs Valid to Syndrome Bus Valid	*Figure 13d*, $\overline{OES}$ Low		19	32	ns
t_{CEV}	Check Bit Inputs Valid (PSH) to Any Error Valid	*Figure 13d*		17	30	ns
t_{CCD}	Check Bit Input Valid (PSH) to Corrected Data Valid	*Figure 13d*		30	45	ns
t_{DCB32}	Data Input Valid to Check Bit Valid	*Figure 13d*		49	75	ns
t_{DEV32}	Data Input Valid to Any Error Valid	*Figure 13d*		46	67	ns
t_{DCD32}	Data Input Valid to Corrected Data Out	*Figure 13d*, $\overline{OB0}$, $\overline{OB1}$ Low		84	110	ns

Note 1: "Absolute Maximum Ratings" are the values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

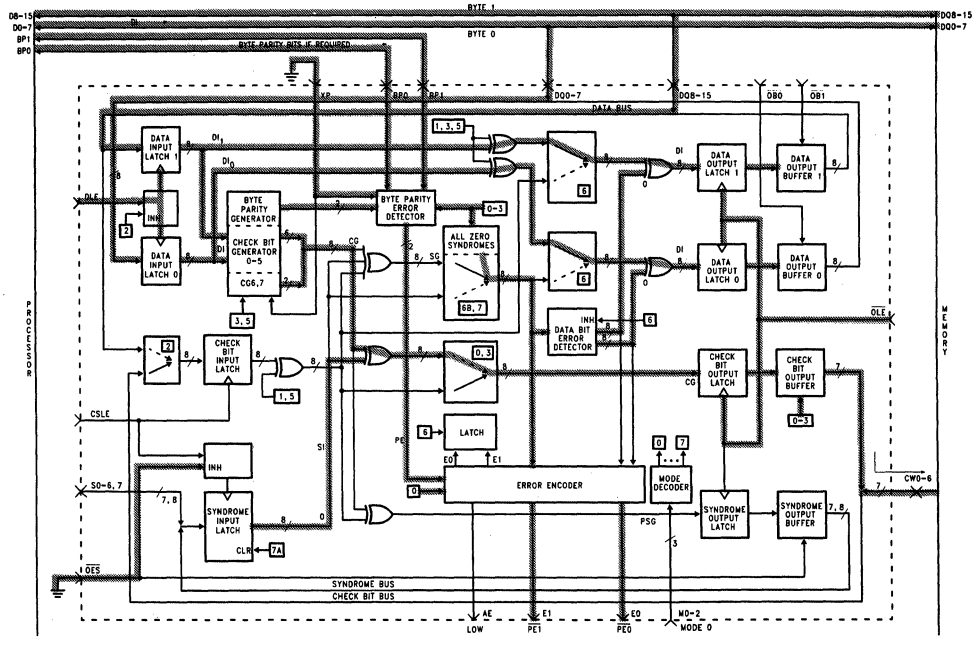
Note 2: All typical values are for $T_A = 25°C$ and $V_{CC} = 5.0V$.

Note 3: Only one output at a time should be shorted.

Note 4: Input capacitance is guaranteed by periodic testing. F test = 10 kHz at 300 mV, $T_A = 25°C$.

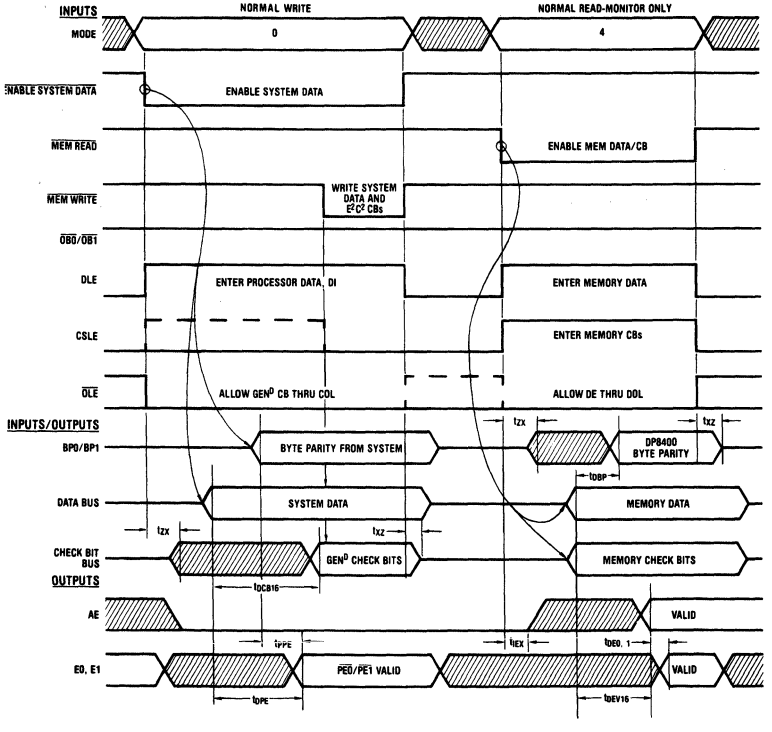
Note 5: All switching parameters measured from 1.5V of input to 1.5V of output. Input pulse amplitude 0V to 3V, $t_r = t_f = 2.5$ ns.

Typical Applications



TL/F/6899–17

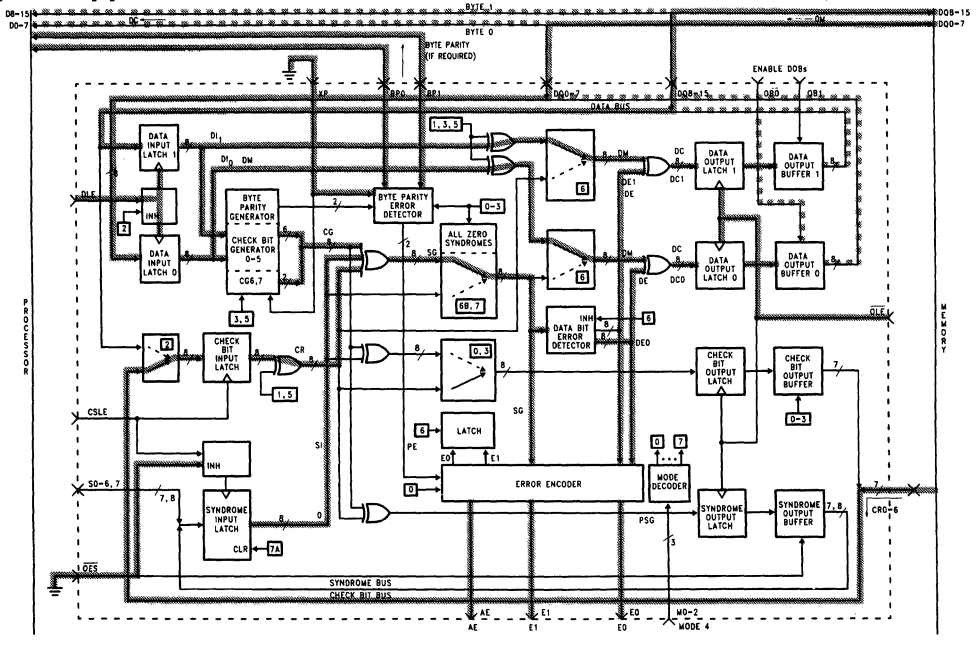
FIGURE 9a. DP8400-2 16-Bit Configuration, Normal WRITE with Byte Parity Error Detect If Required



TL/F/6899–18

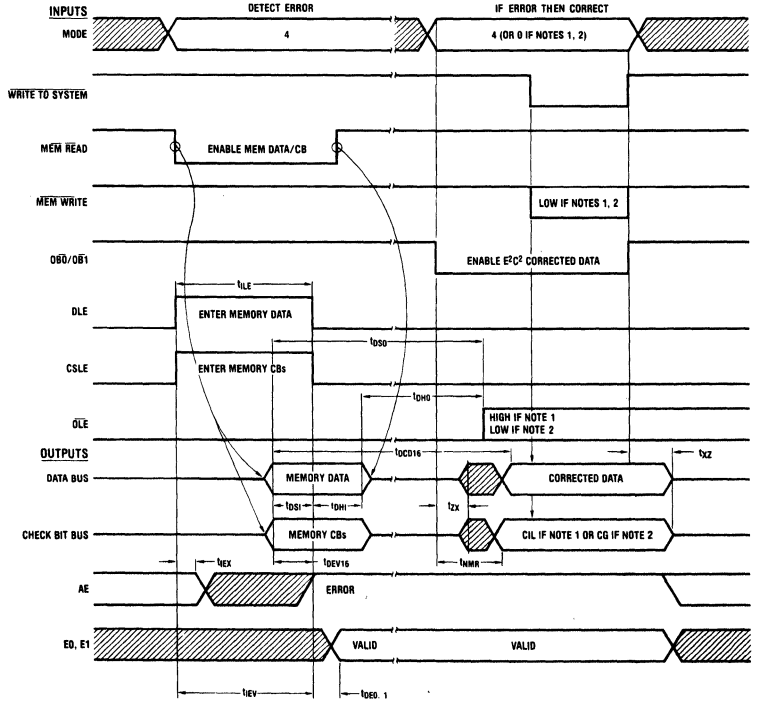
FIGURE 9b. DP8400-2 16-Bit Configuration, Normal WRITE and Normal READ Timing Diagram

Typical Applications (Continued)



TL/F/6899–19

FIGURE 10a. DP8400-2 16-Bit Configuration, Normal READ — Detect Error (And Correct if Required---)



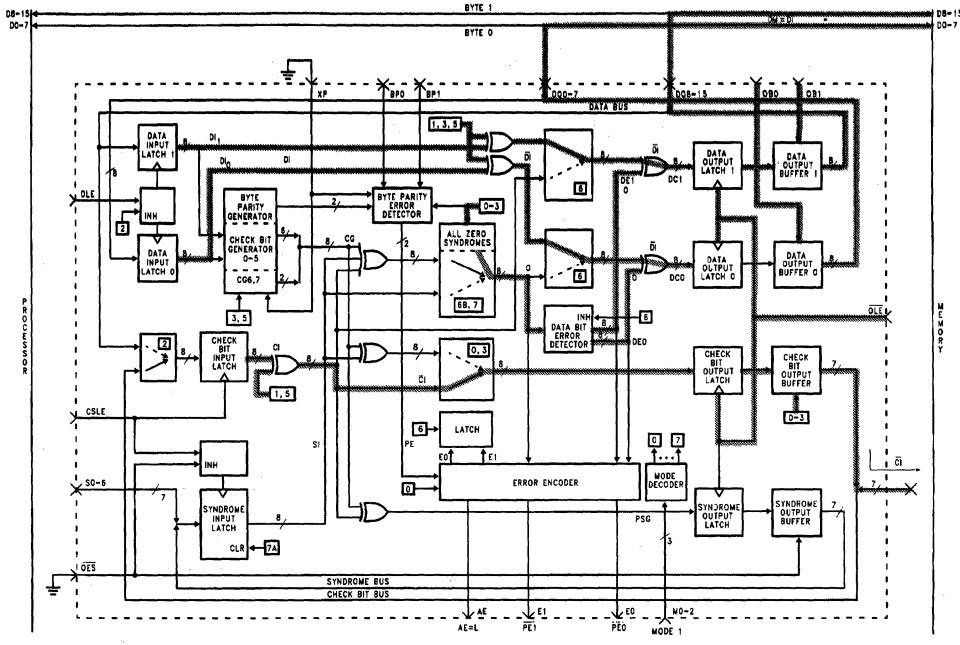
TL/F/6899–20

Note 1: If rewriting correct data and CBs to same location and single data error was detected.

Note 2: If rewriting correct data and CBs to same location and single check bit was detected.

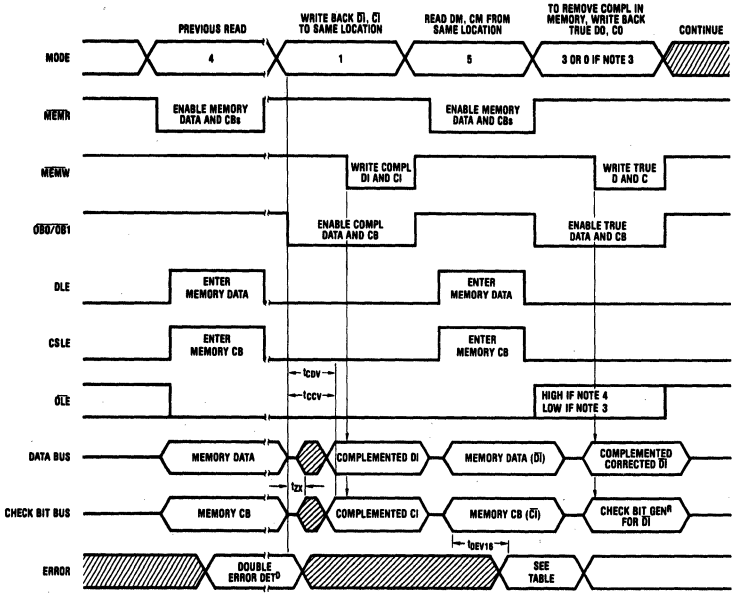
FIGURE 10b. DP8400-2 16-Bit Configuration, DETECT THEN CORRECT Timing Diagram

Typical Applications (Continued)



TL/F/6899–21

FIGURE 11a. DP8400-2 16-Bit Configuration, COMPLEMENT WRITE



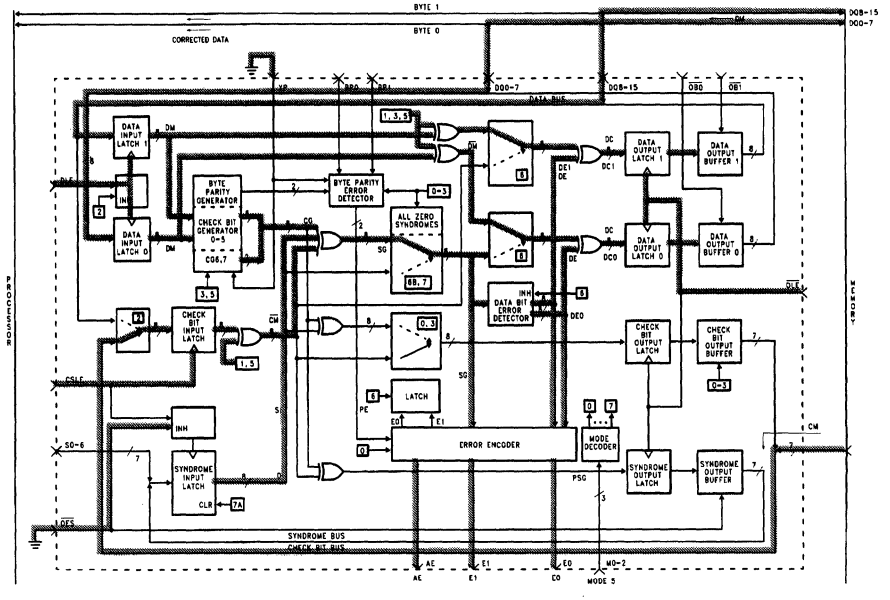
TL/F/6899–22

Note 3: If rewriting corrected data and CBs back to same location and 1 soft data bit error was detected.

Note 4: If rewriting corrected data and CBs back to same location and 2 hard errors or 1 soft check bit was detected.

FIGURE 11b. DP8400-2 16-Bit Configuration, Detect 2 Errors, COMPLEMENT WRITE, COMPLEMENT READ, Output Corrected Data Timing Diagram

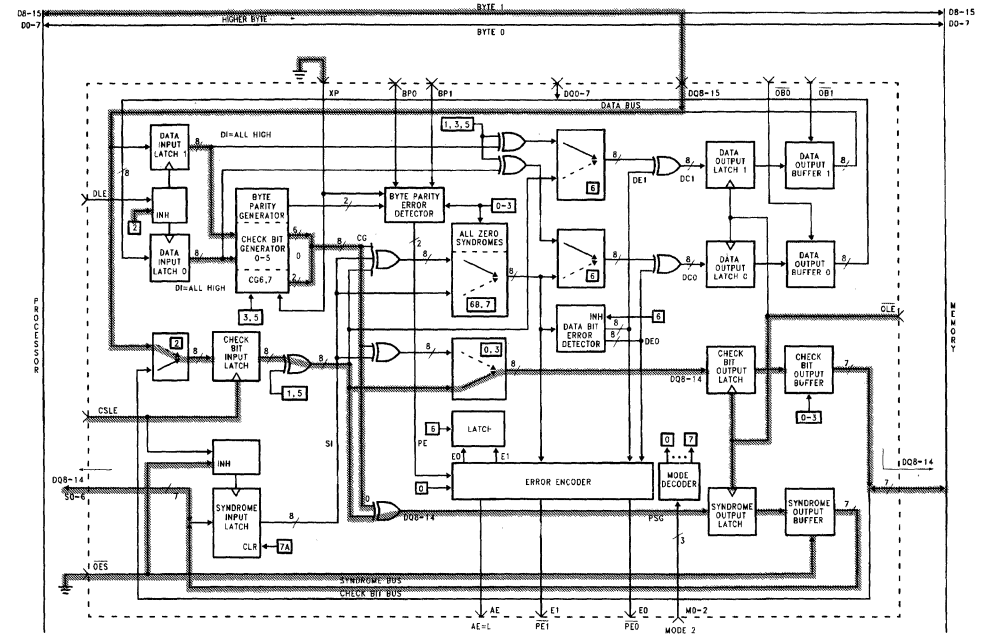
Typical Applications (Continued)



TL/F/6899–23

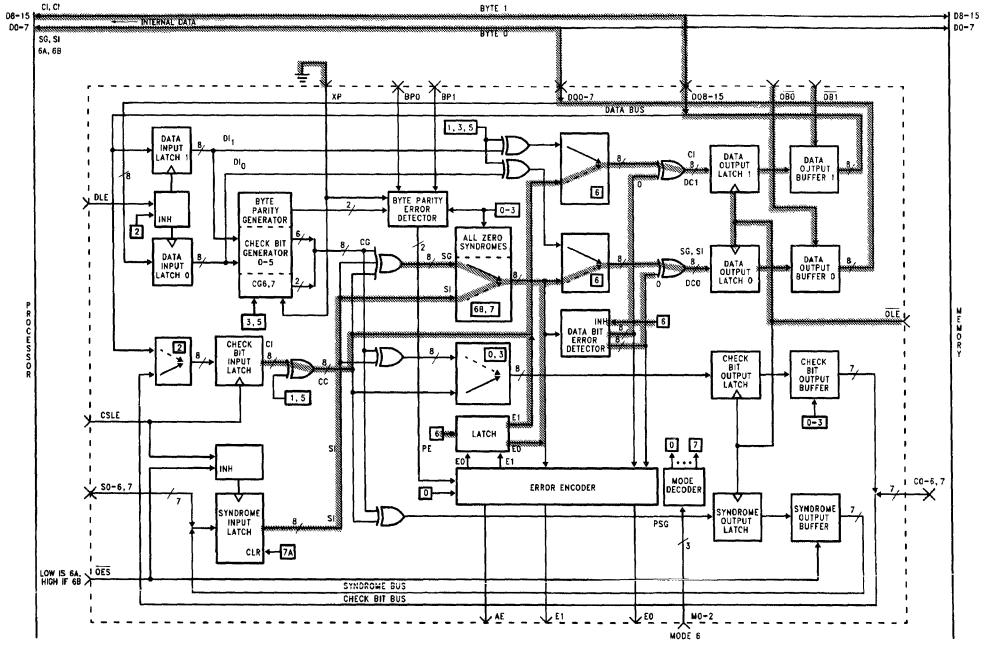
FIGURE 11c. DP8400-2 16-Bit Configuration, COMPLEMENT READ and Output Corrected if One or Two Hard Errors

Typical Applications (Continued)



TL/F/6899–24

FIGURE 12a. DP8400-2 16-Bit Configuration, Diagnostic WRITE, READ. Data Bus to Check Bit Bus or Syndrome Bus (Providing DI = HIGH in Previous Cycle to Set CG = All Zero For Transfer to S)



TL/F/6899–25

FIGURE 12b. DP8400-2 16-Bit Configuration, Monitor on Data Bus — Memory Check Bits

Typical Applications (Continued)

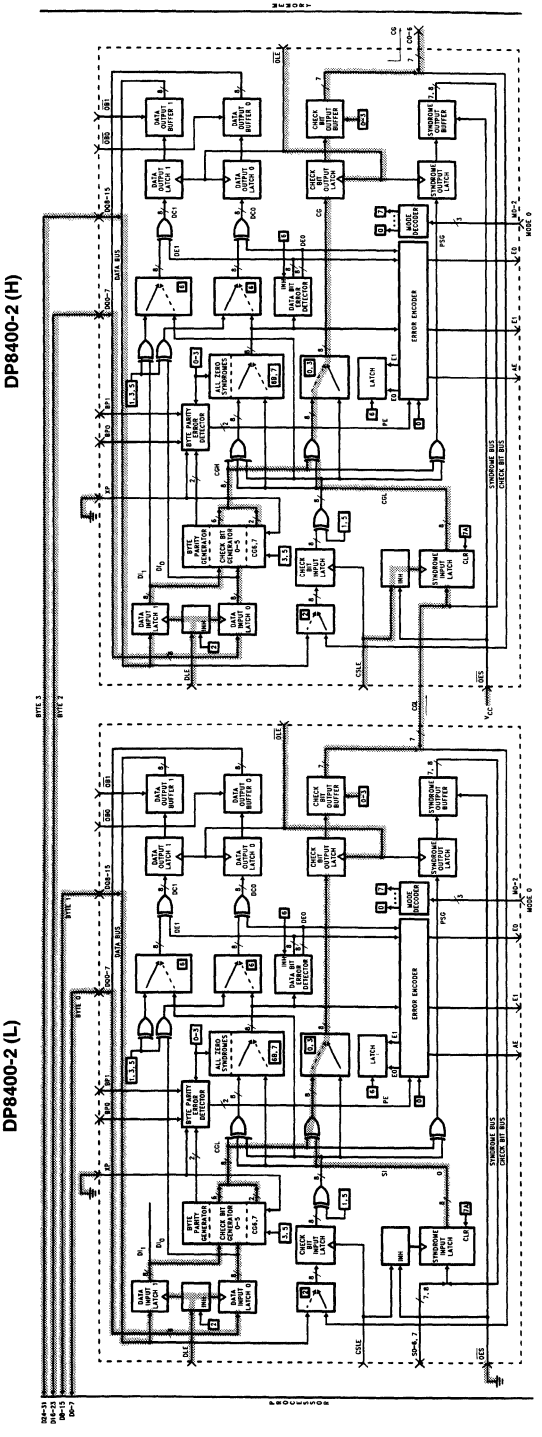


FIGURE 13a. DP8400-2 32-Bit Configuration, WRITE

TL/F/6899–26

Typical Applications (Continued)

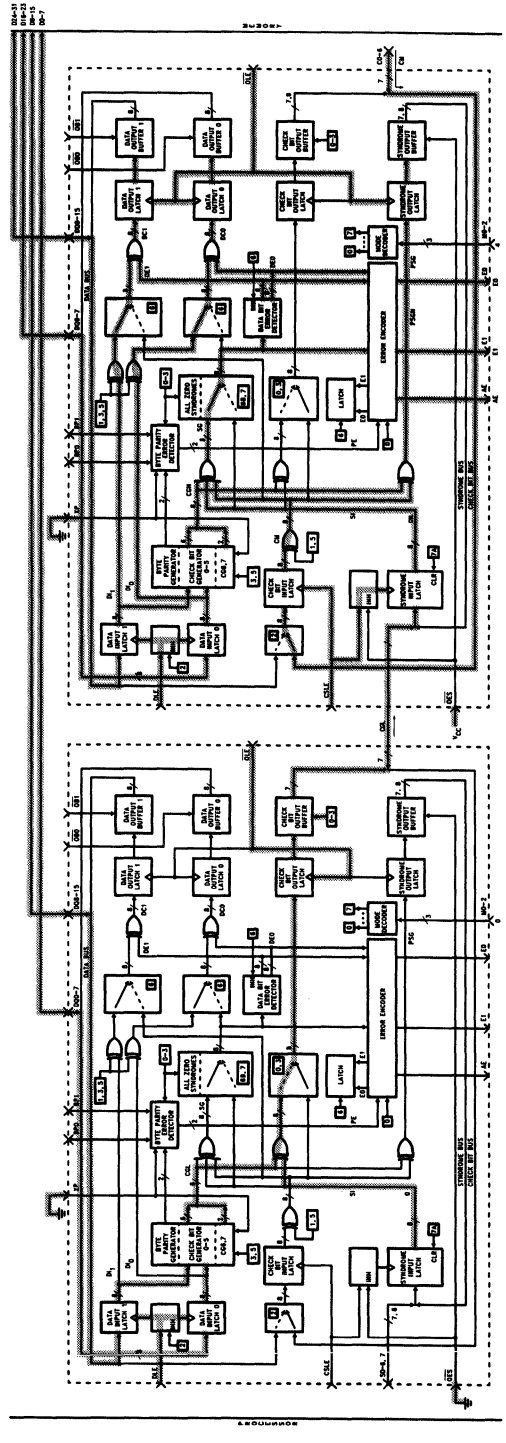


FIGURE 13b. DP8400-2 32-Bit Configuration, READ Detect Error Only

TL/F/6899–27

Typical Applications (Continued)

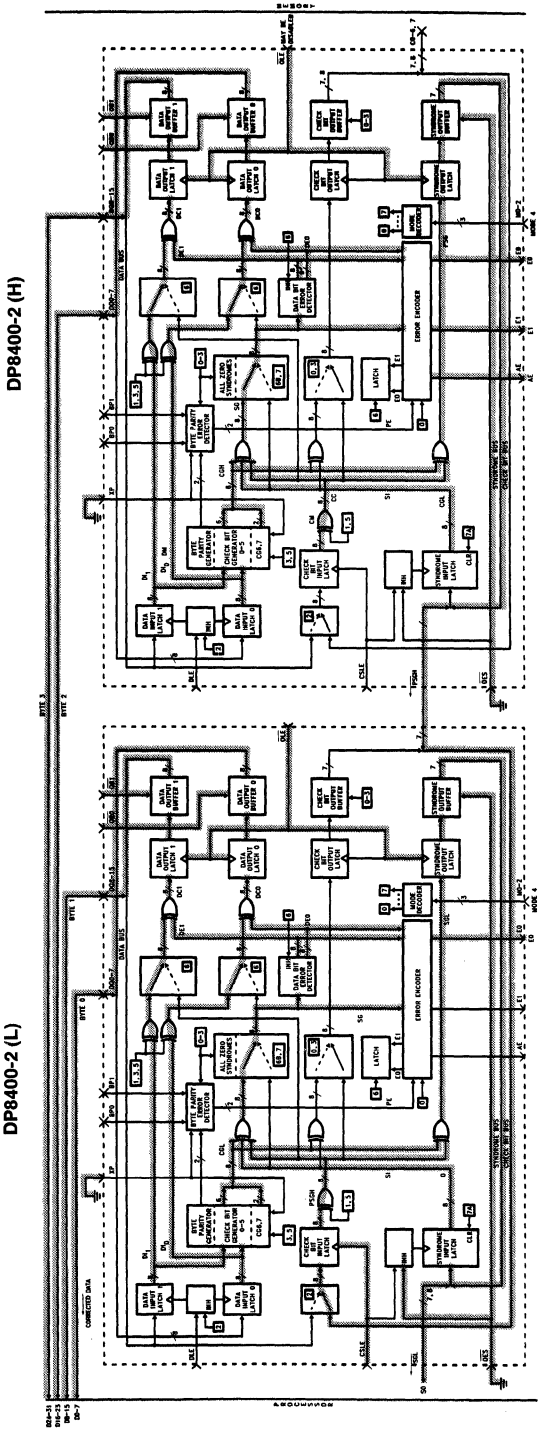
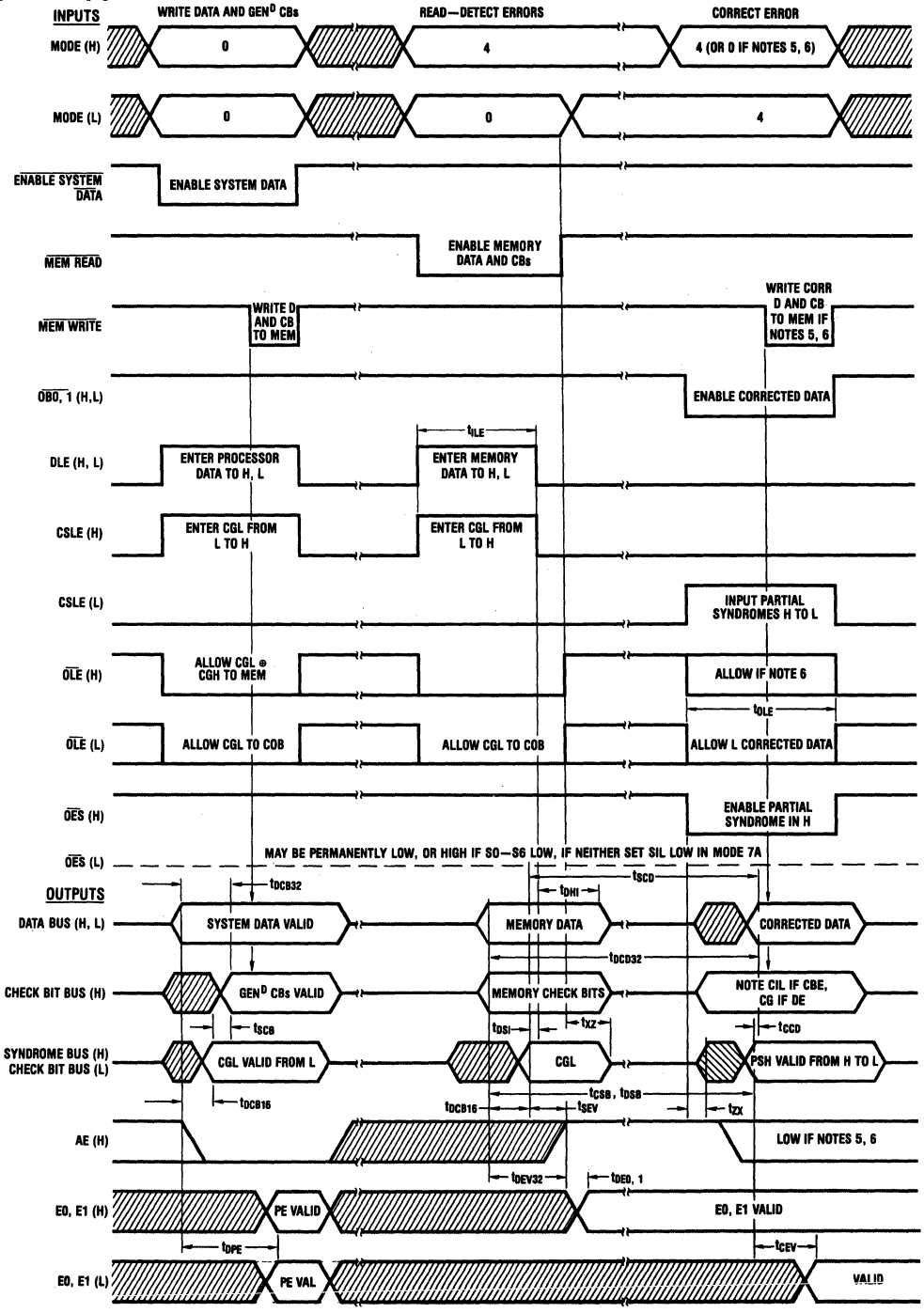


FIGURE 13c. DP8400-2 32-Bit Configuration, READ Correct Data

TL/F/6899–28

Typical Applications (Continued)

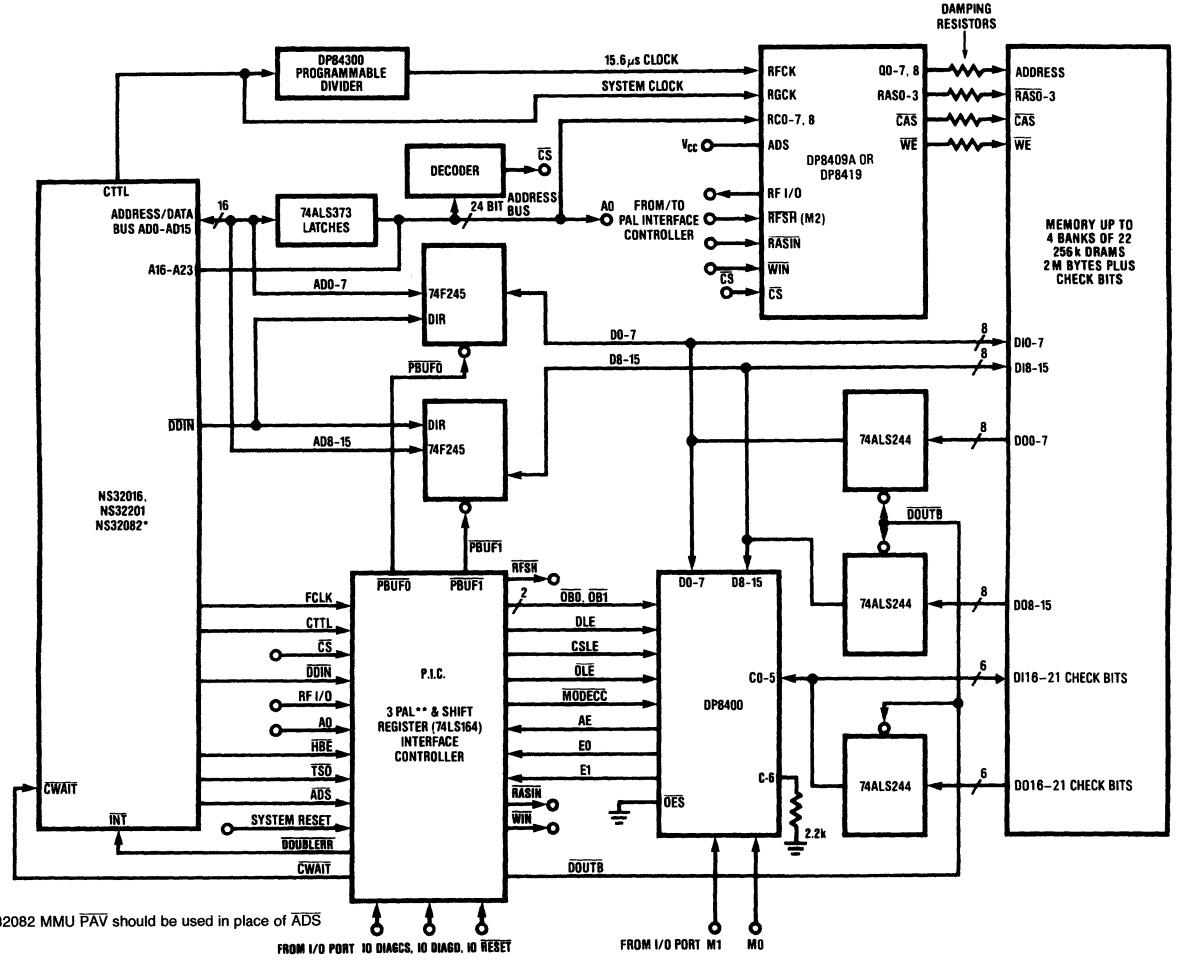


Note 5: If rewriting corrected data and CBs back to same location and single data error was detected. TL/F/6899–29

Note 6: If rewriting corrected data and CBs back to same location and single check bit error was detected.

FIGURE 13d. DP8400-2 32-Bit Configuration, WRITE, DETECT and CORRECT Timing Diagram

NS32016, DP8400-2, DP8409A or DP8419 Error Correcting Memory System

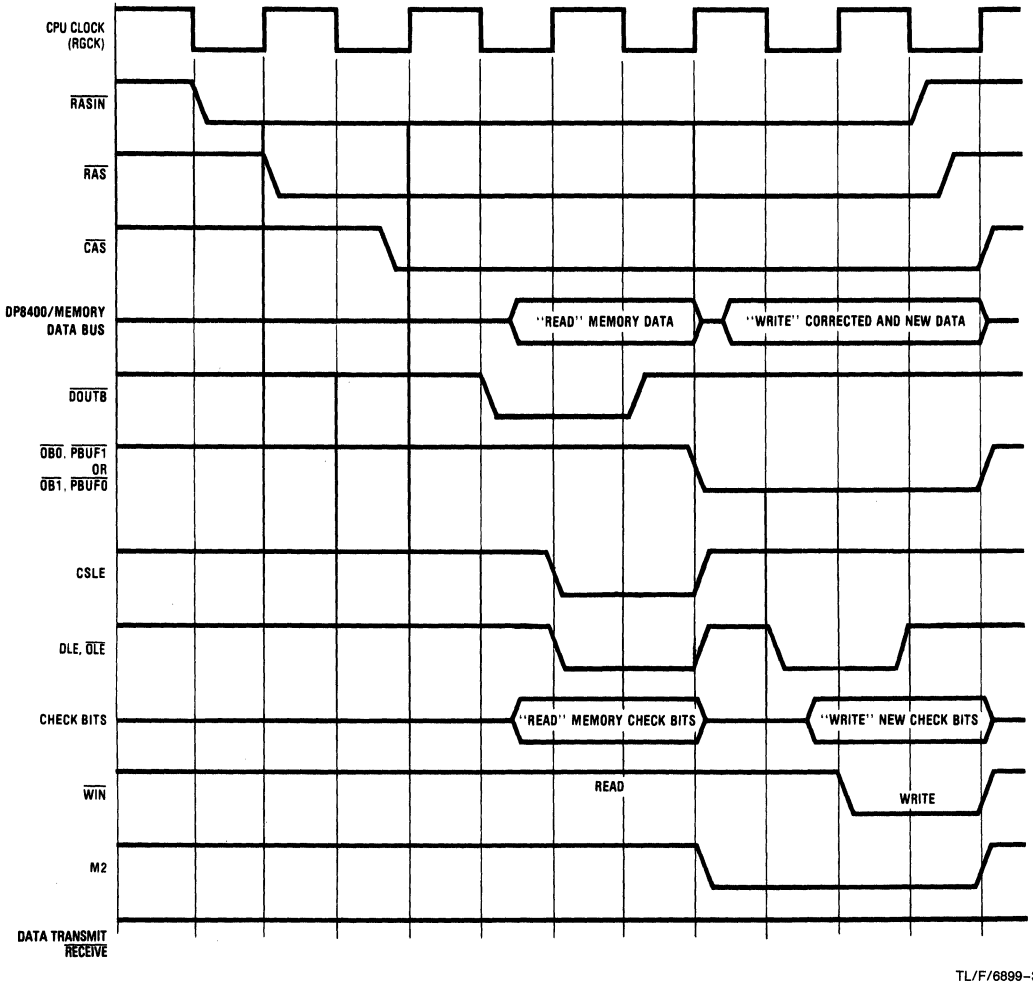


IF system contains NS32082 MMU $\overline{PAV}$ should be used in place of $\overline{ADS}$

TL/F/6899–30

FIGURE 14a. DP8400-2/8409A System Interface Block Diagram (See Figure 14b for Byte Write Control Timing)

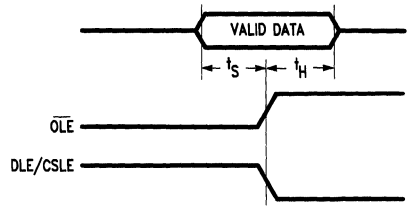
Typical Applications (Continued)



TL/F/6899–31

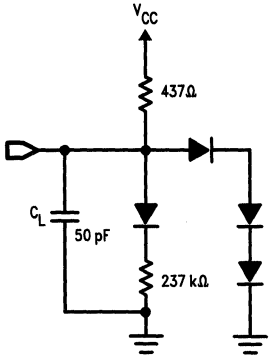
FIGURE 14b. DP8400-2 16-Bit Configuration, Byte Write Timing

Typical Applications (Continued)



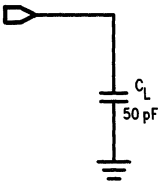
TL/F/6899–32

FIGURE 15. Timing Waveform for Set-Up and Hold Time



TL/F/6899–33

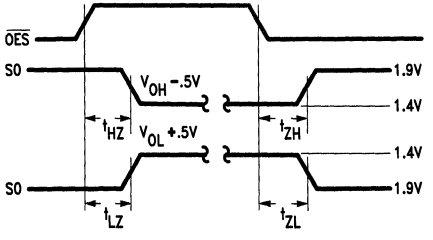
(a) TRI-STATE LOAD



TL/F/6899–34

(b) TEST LOAD

FIGURE 16. Loading Circuit



TL/F/6899–35

FIGURE 17. TRI-STATE Measurement

National Semiconductor

PRELIMINARY

DP8402A/DP8403/DP8404/DP8405 32-Bit Parallel Error Detection and Correction Circuits (EDAC's)

General Description

The DP8402A, DP8403, DP8404 and DP8405 devices are 32-bit parallel error detection and correction circuits (EDACs) in 52-pin DP8402A and DP8403 or 48-pin DP8404 and DP8405 600-mil packages. The EDACs use a modified Hamming code to generate a 7-bit check word from a 32-bit data word. This check word is stored along with the data word during the memory write cycle. During the memory read cycle, the 39-bit words from memory are processed by the EDACs to determine if errors have occurred in memory.

Single-bit errors in the 32-bit data word are flagged and corrected.

Single-bit errors in the 7-bit check word are flagged, and the CPU sends the EDAC through the correction cycle even though the 32-bit data word is not in error. The correction cycle will simply pass along the original 32-bit data word in this case and produce error syndrome bits to pinpoint the error-generating location.

Double bit errors are flagged but not corrected. These errors may occur in any two bits of the 39-bit word from memory (two errors in the 32-bit data word, two errors in the 7-bit check word, or one error in each word). The gross-error condition of all lows or all highs from memory will be detected. Otherwise, errors in three or more bits of the 39-bit word are beyond the capabilities of these devices to detect.

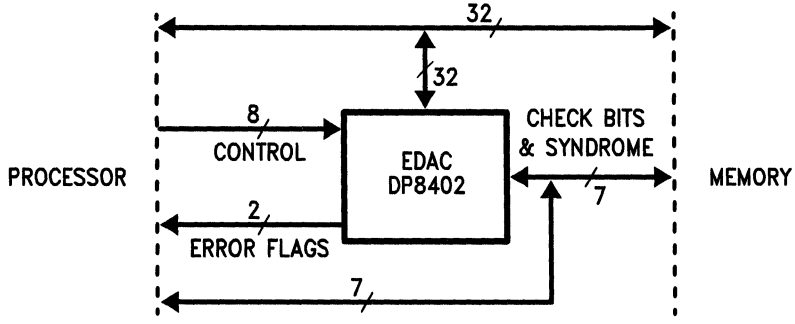
Read-modify-write (byte-control) operations can be performed with the DP8402A and DP8403 EDACs by using output latch enable, $\overline{\text{LEDBO}}$, and the individual $\overline{\text{OEB0}}$ thru $\overline{\text{OEB3}}$ byte control pins.

Diagnostics are performed on the EDACs by controls and internal paths that allow the user to read the contents of the DB and CB input latches. These will determine if the failure occurred in memory or in the EDAC.

Features

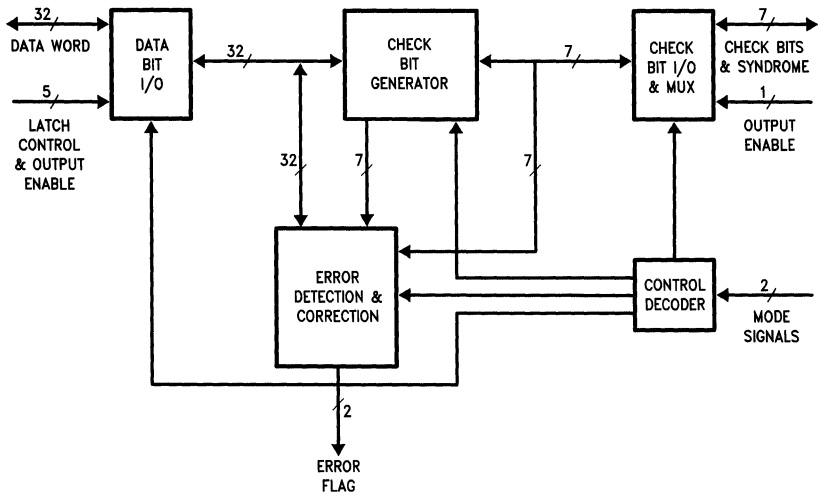
- Detects and corrects single-bit errors
- Detects and flags double-bit errors
- Built-in diagnostic capability
- Fast write and read cycle processing times
- Byte-write capability . . . DP8402A and DP8403
- Fully pin and function compatible with TI's SN74ALS632A thru SN74ALS635 series

System Environment



TL/F/8535–1

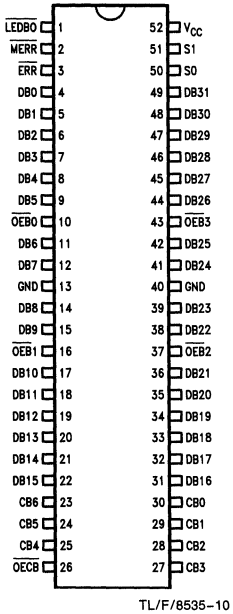
Simplified Functional Block and Connection Diagrams



TL/F/8535–2

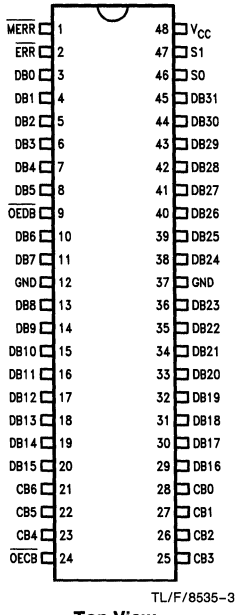
Device	Package	Byte-Write	Output
DP8402A	52-pin	yes	TRI-STATE®
DP8403	52-pin	yes	Open-Collector
DP8404	48-pin	no	TRI-STATE
DP8405	48-pin	no	Open-Collector

Dual-In-Line Packages



TL/F/8535–10

Top View

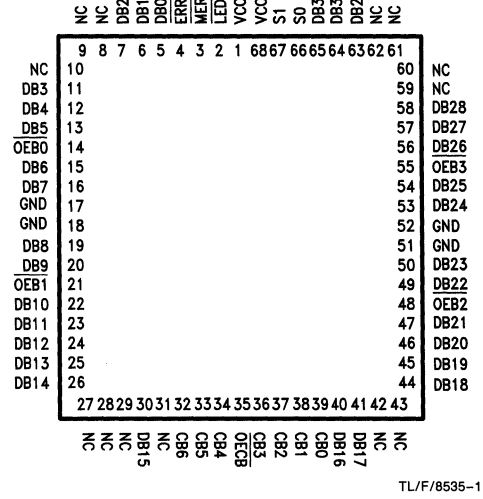


TL/F/8535–3

Top View

Order Number DP8402AD,
DP8403D, DP8404D or DP8405D
See NS Package Number D48A or D52A

Plastic Chip Carrier



TL/F/8535–11

Top View

Order Number DP8402AV
See NS Package Number V68A

Mode Definitions

MODE	PIN NAME S1	S0	DESCRIPTION MODE	OPERATION
0	L	L	WRITE	Input dataword and output checkword
1	L	H	DIAGNOSTICS	Input various data words against latched checkword/output valid error flags.
2	H	L	READ & FLAG	Input dataword and output error flags
3	H	H	CORRECT	Latched input data and checkword/output corrected data and syndrome code

Pin Definitions

S0, S1	Control of EDAC mode, see preceding Mode Definitions
DB0 thru DB31	I/O port for 32 bit dataword.
CB0 thru CB6	I/O port for 7 bit checkword. Also output port for the syndrome error code during error correction mode.
$\overline{OEB0}$ thru $\overline{OEB3}$ (DP8402A, DP8403)	Dataword output buffer enable. When high, output buffers are at TRI-STATE. Each pin controls 8 I/O ports. $\overline{OEB0}$ controls DB0 thru DB7, $\overline{OEB1}$ controls DB8 thru DB15, $\overline{OEB2}$ controls DB16 thru DB23 and $\overline{OEB3}$ controls DB24 thru DB31.
$\overline{LEDBO}$ (DP8402A, DP8403)	Data word output Latch enable. When high it inhibits input to the Latch. Operates on all 32 bits of the dataword.
$\overline{OEDB}$ (DP8404, DP8405)	TRI-STATE control for the data I/O port. When high output buffers are at TRI-STATE.
$\overline{OECB}$	Checkword output buffer enable. When high the output buffers are in TRI-STATE mode.
$\overline{ERR}$	Single error output flag, a low indicates at least a single bit error.
$\overline{MERR}$	Multiple error output flag, a low indicates two or more errors present.

PCC Pin Definitions DP8402A

pin 1	V_{CC}	pin 35	$\overline{OECB}$
2	$\overline{LEDBO}$	36	CB3
3	$\overline{MERR}$	37	CB2
4	$\overline{ERR}$	38	CB1
5	DB0	39	CB0
6	DB1	40	DB16
7	DB2	41	DB17
8	NC	42	NC
9	NC	43	NC
10	NC	44	DB18
11	DB3	45	DB19
12	DB4	46	DB20
13	DB5	47	DB21
14	$\overline{OEBO}$	48	$\overline{OEB2}$
15	DB6	49	DB22
16	DB7	50	DB23
17	GND	51	GND
18	GND	52	GND
19	DB8	53	DB24
20	DB9	54	DB25
21	$\overline{OEB1}$	55	$\overline{OEB3}$
22	DB10	56	DB26
23	DB11	57	DB27
24	DB12	58	DB28
25	DB13	59	NC
26	DB14	60	NC
27	NC	61	NC
28	NC	62	NC
29	NC	63	DB29
30	DB15	64	DB30
31	NC	65	DB31
32	CB6	66	S0
33	CB5	67	S1
34	CB4	68	V_{CC}

TABLE I. Write Control Function

Memory Cycle	EDAC Function	Control S1	S0	Data I/O	DB Control $\overline{OEBn}$ or $\overline{OEDB}$	DB Output Latch DP8402A, DP8403 $\overline{LEDBO}$	Check I/O	CB Control $\overline{OECB}$	Error Flags $\overline{ERR}$	$\overline{MERR}$
Write	Generate check word	L	L	Input	H	X	Output check bits†	L	H	H

†See Table II for details on check bit generation.

Memory Write Cycle Details

During a memory write cycle, the check bits (CB0 thru CB6) are generated internally in the EDAC by seven 16-input parity generators using the 32-bit data word as defined in Table 2. These seven check bits are stored in memory along with the original 32-bit data word. This 32-bit word will later be used in the memory read cycle for error detection and correction.

TABLE II. Parity Algorithm

Check Word	32-Bit Data Word																													
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CB0	X		X	X		X					X		X	X	X			X			X		X	X	X	X		X				X
CB1				X		X		X		X		X		X	X	X				X		X		X		X		X			X	X
CB2	X		X			X	X		X			X	X			X	X		X			X	X		X		X	X	X		X
CB3			X	X	X				X	X	X				X	X			X	X	X				X	X	X			X	X
CB4	X	X							X	X	X	X	X	X			X	X							X	X	X	X	X	X		
CB5	X	X	X	X	X	X	X	X									X	X	X	X	X	X	X	X								
CB6	X	X	X	X	X	X	X	X																	X	X	X	X	X	X	X	X

The seven check bits are parity bits derived from the matrix of data bits as indicated by "X" for each bit.

Check bits 0, 1, 2 are odd parity or the exclusive NORing of the "X"ed bits for the particular check bit. Check bits 3, 4, 5, 6 are even parity or the exclusive ORing of the "X"ed bits for the particular check bit.

Memory Read Cycle (Error Detection & Correction Details)

During a memory read cycle, the 7-bit check word is retrieved along with the actual data. In order to be able to determine whether the data from the memory is acceptable to use as presented on the bus, the error flags must be tested to determine if they are at the high level.

The first case in Table III represents the normal, no-error conditions. The EDAC presents highs on both flags. The next two cases of single-bit errors give a high on $\overline{MERR}$ and a low on $\overline{ERR}$, which is the signal for a correctable error, and the EDAC should be sent through the correction cycle. The last three cases of double-bit errors will cause the EDAC to signal lows on both $\overline{ERR}$ and $\overline{MERR}$, which is the interrupt indication for the CPU.

TABLE III. Error Function

Total Number of Errors		Error Flags		Data Correction
32-Bit Data Word	7-Bit Check Word	$\overline{ERR}$	$\overline{MERR}$	
0	0	H	H	Not applicable
1	0	L	H	Correction
0	1	L	H	Correction
1	1	L	L	Interrupt
2	0	L	L	Interrupt
0	2	L	L	Interrupt

The DP8402 check bit syndrome matrix can be seen in TABLE II. The horizontal rows of this matrix generate the check bits by selecting different combinations of data bits, indicated by "X"s in the matrix, and generating parity from them. For instance, parity check bit "0" is generated by EXCLUSIVE NORing the following data bits together; 31, 29, 28, 26, 21, 19, 18, 17, 14, 11, 9, 8, 7, 6, 4, and 0. For example, the data word "00000001H" would generate the check bits CB6–0 = 48H (Check bits 0, 1, 2 are odd parity and check bits 3, 4, 5, 6 are even parity).

During a WRITE operation (mode 0) the data enters the DP8402 and check bits are generated at the check bit input/output port. Both the data word and the check bits are then written to memory.

During a READ operation (mode 2, error detection) the data and check bits that were stored in memory, now possibly in error, are input through the data and check bit I/O ports. New check bits are internally generated from the data word. These new check bits are then compared, by an EXCLUSIVE NOR operation, with the original check bits that were stored in memory. The EXCLUSIVE NOR of the original check bits, that were stored in memory, with the new check bits is called the syndrome word. If the original check bits are the same as the new check bits, a no error condition, then a syndrome word of all ones is produced and both error flags ($\overline{ERR}$ and $\overline{MERR}$) will be high. The DP8402 matrix encodes errors as follows:

TABLE IV. Read, Flag, and Correct Function

Memory Cycle	EDAC Function	Control S1	Control S0	Data I/O	DB Control $\overline{OEBn}$ or $\overline{OEDB}$	DB Output Latch DP8402A, DP8403 $\overline{LEDBO}$	Check I/O	CB Control $\overline{OECB}$	Error Flags $\overline{ERR}$	Error Flags $\overline{MERR}$
Read	Read & flag	H	L	Input	H	X	Input	H	Enabled†	
Read	Latch input data and check bits	H	H	Input data latched	H	L	Input check word latched	H	Enabled†	
Read	Output corrected data & syndrome bits	H	H	Output corrected data word	L	X	Output syndrome bits‡	L	Enabled†	

†See Table III for error description.

‡See Table V for error location.

Memory Read Cycle (Error Detection & Correction Details) (Continued)

1) Single data bit errors cause 3 or 5 bits in the syndrome word to go low. The columns of the check bit syndrome matrix (TABLE II) are the syndrome words for all single bit data errors in the 32 bit word (also see TABLE V). The data bit in error corresponds to the column in the check bit syndrome matrix that matches the syndrome word. For instance, the syndrome word indicating that data bit 31 is in error would be (CB6-CB0) = "0001010", see the column for data bit 31 in TABLE II, or see TABLE V. During mode 3 (S0 = S1 = 1) the syndrome word is decoded, during single data bit errors, and used to invert the bit in error thus correcting the data word. The corrected word is made available on the data I/O port (DB0 thru DB31), the check word I/O port (CB0 thru CB6) presents the 7-bit syndrome error code. This syndrome error code can be used to locate the bad memory chip.

2) A single check bit error will cause that particular check bit to go low in the syndrome word.

3) A double bit error will cause an even number of bits in the syndrome word to go low. The syndrome word will then be the EXCLUSIVE NOR of the two individual syndrome words corresponding to the 2 bits in error. The two-bit error is not correctable since the parity tree can only identify single bit errors.

If any of the bits in the syndrome word are low the "$\overline{\text{ERR}}$" flag goes low. The "$\overline{\text{MERR}}$" (dual error) flag goes low during any double bit error conditions. (See Table III).

Three or more simultaneous bit errors can cause the EDAC to believe that no error, a correctable error, or an uncorrectable error has occurred and will produce erroneous results in all three cases. It should be noted that the gross-error conditions of all lows and all highs will be detected.

TABLE V. Syndrome Decoding

Syndrome Bits							Error
6	5	4	3	2	1	0	
L	L	L	L	L	L	L	unc
L	L	L	L	L	L	H	2-bit
L	L	L	L	L	H	L	2-bit
L	L	L	L	L	H	H	unc
L	L	L	L	H	L	L	2-bit
L	L	L	L	H	L	H	unc
L	L	L	L	H	H	L	unc
L	L	L	L	H	H	H	2-bit
L	L	L	H	L	L	L	2-bit
L	L	L	H	L	L	H	unc
L	L	L	H	L	H	L	DB31
L	L	L	H	L	H	H	2-bit
L	L	L	H	H	L	L	unc
L	L	L	H	H	L	H	2-bit
L	L	L	H	H	H	L	2-bit
L	L	L	H	H	H	H	DB30
L	L	H	L	L	L	L	2-bit
L	L	H	L	L	L	H	unc
L	L	H	L	L	H	L	DB29
L	L	H	L	L	H	H	2-bit
L	L	H	L	H	L	L	DB28
L	L	H	L	H	L	H	2-bit
L	L	H	L	H	H	L	2-bit
L	L	H	L	H	H	H	DB27
L	L	H	H	L	L	L	DB26
L	L	H	H	L	L	H	2-bit
L	L	H	H	L	H	L	2-bit
L	L	H	H	L	H	H	DB25
L	L	H	H	H	L	L	2-bit
L	L	H	H	H	L	H	DB24
L	L	H	H	H	H	L	unc
L	L	H	H	H	H	H	2-bit

Syndrome Bits							Error
6	5	4	3	2	1	0	
L	H	L	L	L	L	L	2-bit
L	H	L	L	L	L	H	unc
L	H	L	L	L	H	L	DB7
L	H	L	L	L	H	H	2-bit
L	H	L	L	H	L	L	DB6
L	H	L	L	H	L	H	2-bit
L	H	L	L	H	H	L	2-bit
L	H	L	L	H	H	H	DB5
L	H	L	H	L	L	L	DB4
L	H	L	H	L	L	H	2-bit
L	H	L	H	L	H	L	2-bit
L	H	L	H	L	H	H	DB3
L	H	L	H	H	L	L	2-bit
L	H	L	H	H	L	H	DB2
L	H	L	H	H	H	L	unc
L	H	L	H	H	H	H	2-bit
L	H	H	L	L	L	L	DB0
L	H	H	L	L	L	H	2-bit
L	H	H	L	L	H	L	2-bit
L	H	H	L	L	H	H	unc
L	H	H	L	H	L	L	2-bit
L	H	H	L	H	L	H	DB1
L	H	H	L	H	H	L	unc
L	H	H	L	H	H	H	2-bit
L	H	H	H	L	L	L	2-bit
L	H	H	H	L	L	H	unc
L	H	H	H	L	H	L	unc
L	H	H	H	L	H	H	2-bit
L	H	H	H	H	L	L	unc
L	H	H	H	H	L	H	2-bit
L	H	H	H	H	H	L	2-bit
L	H	H	H	H	H	H	CB6

Syndrome Bits							Error
6	5	4	3	2	1	0	
H	L	L	L	L	L	L	2-bit
H	L	L	L	L	L	H	unc
H	L	L	L	L	H	L	unc
H	L	L	L	L	H	H	2-bit
H	L	L	L	H	L	L	unc
H	L	L	L	H	L	H	2-bit
H	L	L	L	H	H	L	2-bit
H	L	L	L	H	H	H	unc
H	L	L	H	L	L	L	unc
H	L	L	H	L	L	H	2-bit
H	L	L	H	L	H	L	2-bit
H	L	L	H	L	H	H	DB15
H	L	L	H	H	L	L	2-bit
H	L	L	H	H	L	H	unc
H	L	L	H	H	H	L	DB14
H	L	L	H	H	H	H	2-bit
H	L	H	L	L	L	L	unc
H	L	H	L	L	L	H	2-bit
H	L	H	L	L	H	L	2-bit
H	L	H	L	L	H	H	DB13
H	L	H	L	H	L	L	2-bit
H	L	H	L	H	L	H	DB12
H	L	H	L	H	H	L	DB11
H	L	H	L	H	H	H	2-bit
H	L	H	H	L	L	L	2-bit
H	L	H	H	L	L	H	DB10
H	L	H	H	L	H	L	DB9
H	L	H	H	L	H	H	2-bit
H	L	H	H	H	L	L	DB8
H	L	H	H	H	L	H	2-bit
H	L	H	H	H	H	L	2-bit
H	L	H	H	H	H	H	CB5

Syndrome Bits							Error
6	5	4	3	2	1	0	
H	H	L	L	L	L	L	unc
H	H	L	L	L	L	H	2-bit
H	H	L	L	L	H	L	2-bit
H	H	L	L	L	H	H	DB23
H	H	L	L	H	L	L	2-bit
H	H	L	L	H	L	H	DB22
H	H	L	L	H	H	L	DB21
H	H	L	L	H	H	H	2-bit
H	H	L	H	L	L	L	2-bit
H	H	L	H	L	L	H	DB20
H	H	L	H	L	H	L	DB19
H	H	L	H	L	H	H	2-bit
H	H	L	H	H	L	L	DB18
H	H	L	H	H	L	H	2-bit
H	H	L	H	H	H	L	2-bit
H	H	L	H	H	H	H	CB4
H	H	H	L	L	L	L	2-bit
H	H	H	L	L	L	H	DB16
H	H	H	L	L	H	L	unc
H	H	H	L	L	H	H	2-bit
H	H	H	L	H	L	L	DB17
H	H	H	L	H	L	H	2-bit
H	H	H	L	H	H	L	2-bit
H	H	H	L	H	H	H	CB3
H	H	H	H	L	L	L	unc
H	H	H	H	L	L	H	2-bit
H	H	H	H	L	H	L	2-bit
H	H	H	H	L	H	H	CB2
H	H	H	H	H	L	L	2-bit
H	H	H	H	H	L	H	CB1
H	H	H	H	H	H	L	CB0
H	H	H	H	H	H	H	none

CB X = error in check bit X

DB Y = error in data bit Y

2-bit = double-bit error

unc = uncorrectable multibit error

TABLE VI. Read-Modify-Write Function

MEMORY CYCLE	EDAC FUNCTION	CONTROL S1	CONTROL S0	BYTEn†	$\overline{OEB}$n†	DB OUTPUT LATCH $\overline{LEDBO}$	CHECK I/O	CB CONTROL	ERROR FLAG $\overline{ERR}$ $\overline{MERR}$
Read	Read & Flag	H	L	Input	H	X	Input	H	Enabled
Read	Latch input data & check bits	H	H	Input data latched	H	L	Input check word latched	H	Enabled
Read	Latch corrected data word into output latch	H	H	Output data word latched	H	H	Hi-Z	H	Enabled
							Output Syndrome bits	L	
Modify /write	Modify appropriate byte or bytes & generate new check word	L	L	Input modified BYTE0	H	H	Output check word	L	H H
				Ouput unchanged BYTE0	L				

†$\overline{OEB}$0 controls DB_0–DB_7 (BYTE0), $\overline{OEB}$1 controls DB_8–DB_{15} (BYTE1), $\overline{OEB}$2 controls DB16–DB23 (BYTE2), $\overline{OEB}$3 controls DB24–DB31 (BYTE3).

Read-Modify-Write (Byte Control) Operations

The DP8402A and DP8403 devices are capable of byte-write operations. The 39-bit word from memory must first be latched into the DB and CB input latches. This is easily accomplished by switching from the read and flag mode (S1 = H, SO = L) to the latch input mode (S1 = H, S0 = H). The EDAC will then make any corrections, if necessary, to the data word and place it at the input of the output data latch. This data word must then be latched into the output data latch by taking $\overline{LEDBO}$ from a low to a high.

Byte control can now be employed on the data word through the $\overline{OEB}$0 through $\overline{OEB}$3 controls. $\overline{OEB}$0 controls DB0–DB7 (byte 0), $\overline{OEB}$1 controls DB8–DB15 (byte 1), $\overline{OEB}$2 controls DB16–DB23 (byte 2), and $\overline{OEB}$3 controls DB24–DB31 (byte 3). Placing a high on the byte control will disable the output and the user can modify the byte. If a low is placed on the byte control, then the original byte is allowed to pass onto the data bus unchanged. If the original data word is altered through byte control, a new check word must be generated before it is written back into memory. This is easily accomplished by taking control S1 and S0 low. Table VI lists the read-modify-write functions.

Diagnostic Operations

The DP8402A thru DP8405 are capable of diagnostics that allow the user to determine whether the EDAC or the memory is failing. The diagnostic function tables will help the user to see the possibilities for diagnostic control.

In the diagnostic mode (S1 = L, S0 = H), the checkword is latched into the input latch while the data input remains transparent. This lets the user apply various data words against a fixed known checkword. If the user applies a diagnostic data word with an error in any bit location, the $\overline{ERR}$ flag should be low. If a diagnostic data word with two errors in any bit location is applied, the $\overline{MERR}$ flag should be low. After the checkword is latched into the input latch, it can be verified by taking $\overline{OECB}$ low. This outputs the latched checkword. With the DP8402A and DP8403, the diagnostic data word can be latched into the output data latch and verified. It should be noted that the DP8404 and DP8405 do not have this pass-through capability because they do not contain an output data latch. By changing from the diagnostic mode (S1 = L, S0 = H) to the correction mode (S1 = H, S0 = H), the user can verify that the EDAC will correct the diagnostic data word. Also, the syndrome bits can be produced to verify that the EDAC pinpoints the error location. Table VII DP8402A and DP8403 and Table VIII DP8404 and DP8405 list the diagnostic functions.

TABLE VII. DP8402A, DP8403 Diagnostic Function

EDAC FUNCTION	CONTROL S1	CONTROL S0	DATA I/O	DB BYTE CONTROL $\overline{OEBn}$	DB OUTPUT LATCH $\overline{LEDBO}$	CHECK I/O	CB CONTROL $\overline{OECB}$	ERROR FLAGS $\overline{ERR}$	ERROR FLAGS $\overline{MERR}$
Read & flag	H	L	Input correct data word	H	X	Input correct check bits	H	H	H
Latch input check word while data input latch remains transparent	L	H	Input diagnostic data word†	H	L	Input check bits latched	H	Enabled	
Latch diagnostic data word into output latch	L	H	Input diagnostic data word†	H	H	Output latched check bits	L	Enabled	
						Hi-Z	H		
Latch diagnostic data word into input latch	H	H	Input diagnostic data word latched	H	H	Output syndrome bits	L	Enabled	
						Hi-Z	H		
Output diagnostic data word & syndrome bits	H	H	Output diagnostic data word	L	H	Output syndrome bits	L	Enabled	
						Hi-Z	H		
Output corrected diagnostic data word & output syndrome bits	H	H	Output corrected diagnostic data word	L	L	Output syndrome bits	L	Enabled	
						Hi-Z	H		

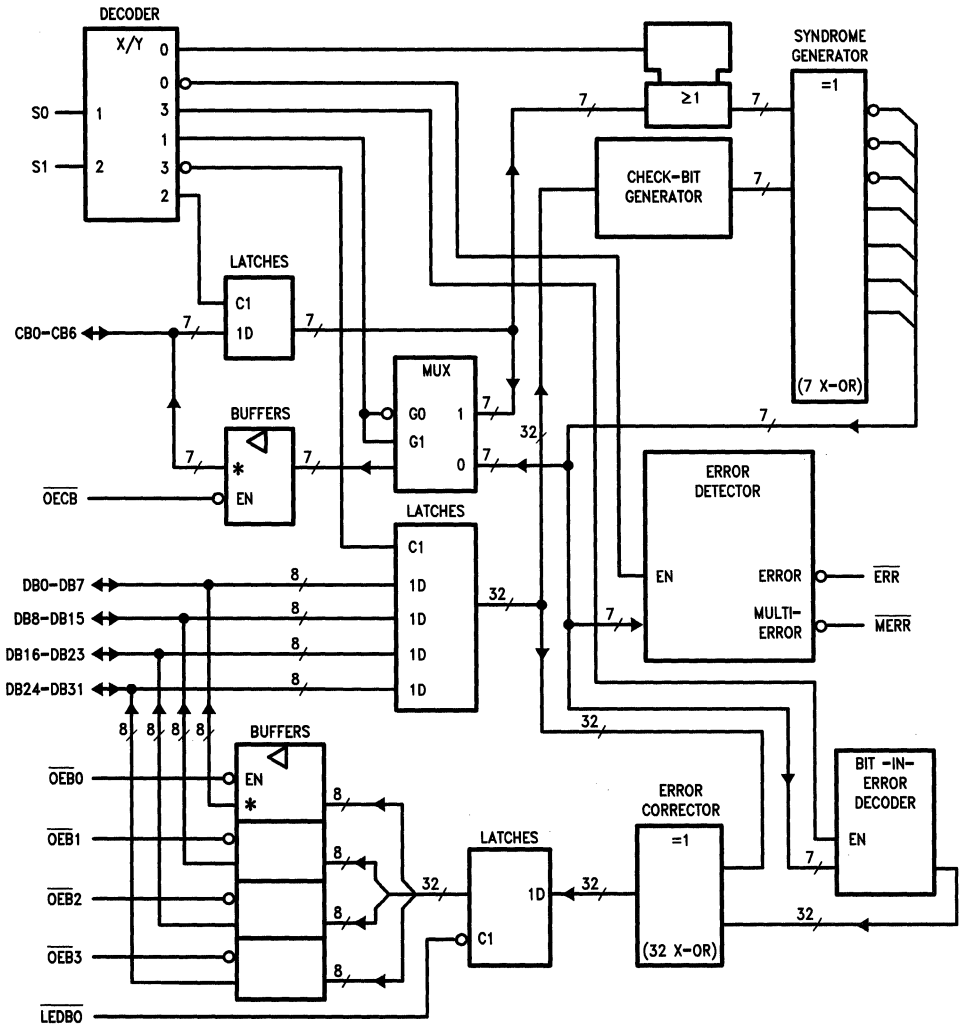
†Diagnostic data is a data word with an error in one bit location except when testing the $\overline{MERR}$ error flag. In this case, the diagnostic data word will contain errors in two bit locations.

TABLE VIII. DP8404, DP8405 Diagnostic Function

EDAC FUNCTION	CONTROL S1	CONTROL S0	DATA I/O	DB CONTROL $\overline{OEDB}$	CHECK I/O	DB CONTROL $\overline{OECB}$	ERROR FLAGS $\overline{ERR}$	ERROR FLAGS $\overline{MERR}$
Read & flag	H	L	Input correct data word	H	Input correct check bits	H	H	H
Latch input check bits while data input latch remains transparent	L	H	Input diagnostic data word†	H	Input check bits latched	H	Enabled	
Output input check bits	L	H	Input diagnostic data word†	H	Output input check bits	L	Enabled	
Latch diagnostic data into input latch	H	H	Input diagnostic data word latched	H	Output syndrome bits	L	Enabled	
					Hi-Z	H		
Output corrected diagnostic data word	H	H	Output corrected diagnostic data word	L	Output syndrome bits	L	Enabled	
					Hi-Z	H		

†Diagnostic data is a data word with an error in one bit location except when testing the $\overline{MERR}$ error flag. In this case, the diagnostic data word will contain errors in two bit locations.

DP8402A, DP8403 Logic Diagram (Positive Logic)

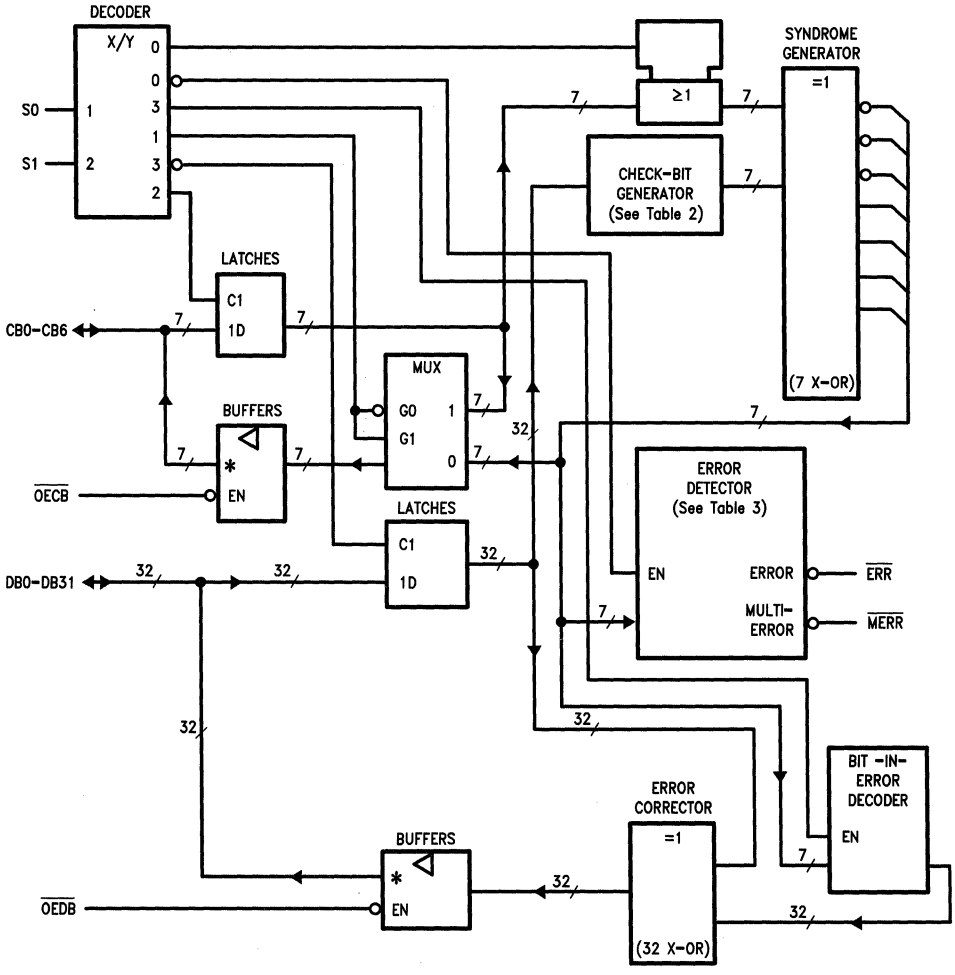


DP8402A HAS TRI-STATE (▽) CHECK-BIT AND DATA OUTPUTS.
DP8403 HAS OPEN-COLLECTOR (⏢) CHECK-BIT AND DATA OUTPUTS.

TL/F/8535-4

2

DP8404, DP8405 Logic Diagram (Positive Logic)



DP8404 HAS TRI-STATE (▽) CHECK-BIT AND DATA OUTPUTS.
DP8405 HAS OPEN-COLLECTOR (◇) CHECK-BIT AND DATA OUTPUTS.

TL/F/8535–5

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Over Operating Free-Air Temperature Range (unless otherwise noted)

Supply Voltage, V_{CC} (See Note 1) 7V

Input Voltage: CB and DB 5.5V

All Others 7V

Operating Free-Air Temperature: Military −55°C to +125°C

Commercial 0° to +70°C

Storage Temperature Range −65°C to +150°C

Recommended Operating Conditions

Symbol	Parameter	Conditions	Military			Commercial			Units
			Min	Typ	Max	Min	Typ	Max	
V_{CC}	Supply Voltage		4.5	5	5.5	4.5	5	5.5	V
V_{IH}	High-Level Input Voltage		2			2			V
V_{IL}	Low-Level Input Voltage				0.8			0.8	V
I_{OH}	High-Level Output Current	$\overline{ERR}$ Or $\overline{MERR}$			−0.4			−0.4	mA
		DB Or CB DP8402A, DP8404			−1			−2.6	
I_{OL}	Low-Level Output Current	$\overline{ERR}$ Or $\overline{MERR}$			4			8	mA
		DB or CB			12			24	
t_w	Pulse Duration	$\overline{LEDBO}$ Low	25			25			ns
t_{su}	Setup Time	(1) Data And Check Word Before S0 ↑ (S1 = H)	15			10			ns
		(2) SO High Before $\overline{LEDBO}$ ↑ (S1 = H)†	45			45			
		(3) $\overline{LEDBO}$ High Before The Earlier of S0 ↓ or S1 ↓ †	0			0			
		(4) $\overline{LEDBO}$ High Before S1 ↑ (S0 = H)	0			0			
		(5) Diagnostic Data Word Before S1 ↑ (S0 = H)	15			10			
		(6) Diagnostic Check Word Before The Later Of S1 ↓ or S0 ↑	15			10			
		(7) Diagnostic Data Word Before $\overline{LEDBO}$ ↑ (S1 = L and S0 = H)‡	25			20			
t_h	Hold Time	(8) Read-Mode, S0 Low And S1 High	35			30			ns
		(9) Data And Check Word After S0 ↑ (S1 = H)	20			15			
		(10) Data Word After S1 ↑ (S0 = H)	20			15			
		(11) Check Word After The Later of S1 ↓ or S0 ↑	20			15			
		(12) Diagnostic Data Word After $\overline{LEDBO}$ ↑ (S1 = L And S0 = H)‡	0			0			
t_{corr}	Correction Time (see *Figure 1*)*		65			58			ns
T_A	Operating Free-Air Temperature		−55		125	0		70	°C

*This specification may be interpreted as the maximum delay to guarantee valid corrected data at the output and includes the t_{su} setup delay.

†These times ensure that corrected data is saved in the output data latch.

‡These times ensure that the diagnostic data word is saved in the output data latch.

DP8402A, DP8404 Electrical Characteristics

Over Recommended Operating Free-Air Temperature Range (unless otherwise noted)

Symbol	Parameter	Test Conditions	Military			Commercial			Units
			Min	Typ†	Max	Min	Typ†	Max	
V_{IK}		V_{CC} = 4.5V, I_I = −18 mA			−1.5			−1.5	V
V_{OH}	All outputs	V_{CC} = 4.5V to 5.5V, I_{OH} = − 0.4 mA	V_{CC}−2			V_{CC}−2			V
	DB or CB	V_{CC} = 4.5V, I_{OH} = −1 mA	2.4	3.3					
		V_{CC} = 4.5V, I_{OH} = −2.6 mA				2.4	3.2		
V_{OL}	$\overline{ERR}$ or $\overline{MERR}$	V_{CC} = 4.5V, I_{OL} = 4 mA		0.25	0.4		0.25	0.4	V
		V_{CC} = 4.5V, I_{OL} = 8 mA					0.35	0.5	
	DB or CB	V_{CC} = 4.5V, I_{OL} = 12 mA		0.25	0.4		0.25	0.4	
		V_{CC} = 4.5V, I_{OL} = 24 mA					0.35	0.5	
I_I	S0 or S1	V_{CC} = 5.5V, V_I = 7V			0.1			0.1	mA
	All others	V_{CC} = 5.5V, V_I = 5.5V			0.1			0.1	
I_{IH}	S0 or S1	V_{CC} = 5.5V, V_I = 2.7V			20			20	μA
	All others‡				20			20	
I_{IL}	S0 or S1	V_{CC} = 5.5V, V_I = 0.4V			−0.4			−0.4	mA
	All others‡				−0.1			−0.1	
I_O§		V_{CC} = 5.5V, V_O = 2.25V	−30		−112	−30		−112	mA
I_{CC}		V_{CC} = 5.5V, (See Note 1)		150	250		150	250	mA

DP8403, DP8405 Electrical Characteristics

Over Recommended Operating Free-Air Temperature Range (unless otherwise noted)

Symbol	Parameter	Test Conditions	Military			Commercial			Units
			Min	Typ†	Max	Min	Typ†	Max	
V_{IK}		V_{CC} = 4.5V, I_I = −18 mA			−1.5			−1.5	V
V_{OH}	$\overline{ERR}$ or $\overline{MERR}$	V_{CC} = 4.5V to 5.5V, I_{OH} = −0.4 mA	V_{CC}−2			V_{CC}−2			V
I_{OH}	DB or CB	V_{CC} = 4.5V, V_{OH} = 5.5V			0.1			0.1	mA
V_{OL}	$\overline{ERR}$ or $\overline{MERR}$	V_{CC} = 4.5V, I_{OL} = 4 mA		0.25	0.4		0.25	0.4	V
		V_{CC} = 4.5V, I_{OL} = 8 mA					0.35	0.5	
	DB or CB	V_{CC} = 4.5V, I_{OL} = 12 mA		0.25	0.4		0.25	0.4	
		V_{CC} = 4.5V, I_{OL} = 24 mA					0.35	0.5	
I_I	S0 or S1	V_{CC} = 5.5V, V_I = 7V							mA
	All others	V_{CC} = 5.5V, V_I = 5.5V							
I_{IH}	S0 or S1	V_{CC} = 5.5V, V_I = 2.7V							μA
	All others‡								
I_{IL}	S0 or S1	V_{CC} = 5.5V, V_I = 0.4V							mA
	All others‡								
I_O§	$\overline{ERR}$ or $\overline{MERR}$	V_{CC} = 5.5V, V_O = 2.25V	−30		−112	−30		−112	mA
I_{CC}		V_{CC} = 5.5V, (See Note 1)		150			150		mA

†All typical values are at V_{CC} = 5V, T_A = +25°C.

‡For I/O ports (Q_A through Q_H), the parameters I_{IH} and I_{IL} include the off-state output current.

§The output conditions have been chosen to produce a current that closely approximates one half of the true short-circuit output current, I_{OS}.

Note 1: I_{CC} is measured with S0 and S1 at 4.5V and all CB and DB pins grounded.

DP8402A Switching Characteristics

V_{CC} = 4.5V to 5.5V, C_L = 50 pF, T_A = Min to Max (unless otherwise noted)

Symbol	From (Input)	To (Output)	Test Conditions	Military		Commercial		Units
				Min	Max	Min	Max	
t_{pd}	DB and CB	$\overline{ERR}$	S1 = H, S0 = L, R_L = 500Ω	10	43	10	40	ns
	DB	$\overline{ERR}$	S1 = L, S0 = H, R_L = 500Ω	10	43	10	40	
t_{pd}	DB and CB	$\overline{MERR}$	S1 = H, S0 = L, R_L = 500Ω	15	67	15	55	ns
	DB	$\overline{MERR}$	S1 = L, S0 = H, R_L = 500Ω	15	67	15	55	
t_{pd}	S0 ↓ and S1 ↓	CB	R1 = R2 = 500Ω	10	60	10	48	ns
t_{pd}	DB	CB	S1 = L, S0 = L, R1 = R2 = 500Ω	10	60	10	48	ns
t_{pd}	$\overline{LEDB0}$ ↓	DB	S1 = X, S0 = H, R1 = R2 = 500Ω	7	35	7	30	ns
t_{pd}	S1 ↑	CB	S0 = H, R1 = R2 = 500Ω	10	60	10	50	ns
t_{en}	$\overline{OECB}$ ↓	CB	S0 = H, S1 = X, R1 = R2 = 500Ω	2	30	2	25	ns
t_{dis}	$\overline{OECB}$ ↑	CB	S0 = H, S1 = X, R1 = R2 = 500Ω	2	30	2	25	ns
t_{en}	$\overline{OEB0}$ thru $\overline{OEB3}$ ↓	DB	S0 = H, S1 = X, R1 = R2 = 500Ω	2	30	2	25	ns
t_{dis}	$\overline{OEB0}$ thru $\overline{OEB3}$ ↑	DB	S0 = H, S1 = X, R1 = R2 = 500Ω	2	30	2	25	ns

DP8403 Switching Characteristics

V_{CC} = 4.5V to 5.5V, C_L = 50 pF, T_A = Min to Max (unless otherwise noted)

Symbol	From (Input)	To (Output)	Test Conditions	Military			Commercial			Units
				Min	Typ†	Max	Min	Typ†	Max	
t_{pd}	DB and CB	$\overline{ERR}$	S1 = H, S0 = L, R_L = 500Ω		26			26		ns
	DB	$\overline{ERR}$	S1 = L, S0 = H, R_L = 500Ω		26			26		
t_{pd}	DB and CB	$\overline{MERR}$	S1 = H, S0 = L, R_L = 500Ω		40			40		ns
			S1 = L, S0 = H, R_L = 500Ω		40			40		
t_{pd}	S0 ↓ and S1 ↓	CB	R_L = 680Ω		40			40		ns
t_{pd}	DB	CB	S1 = L, S0 = L, R_L = 680Ω		40			40		ns
t_{pd}	$\overline{LEDB0}$ ↓	DB	S1 = X, S0 = H, R_L = 680Ω		26			26		ns
t_{pd}	S1 ↑	CB	S0 = H, R_L = 680Ω		40			40		ns
t_{PLH}	$\overline{OECB}$ ↑	CB	S1 = X, S0 = H, R_L = 680Ω		24			24		ns
t_{PHL}	$\overline{OECB}$ ↓	CB	S1 = X, S0 = H, R_L = 680Ω		24			24		ns
t_{PLH}	$\overline{OEB0}$ thru $\overline{OEB3}$ ↑	DB	S1 = X, S0 = H, R_L = 680Ω		24			24		ns
t_{PHL}	$\overline{OEB0}$ thru $\overline{OEB3}$ ↓	DB	S1 = X, S0 = H, R_L = 680Ω		24			24		ns

†All typical values are at V_{CC} = 5V, T_A = +25°C.

2

DP8404 Switching Characteristics, V_{CC} = 4.5V to 5.5V, C_L = 50 pF, T_A = Min to Max

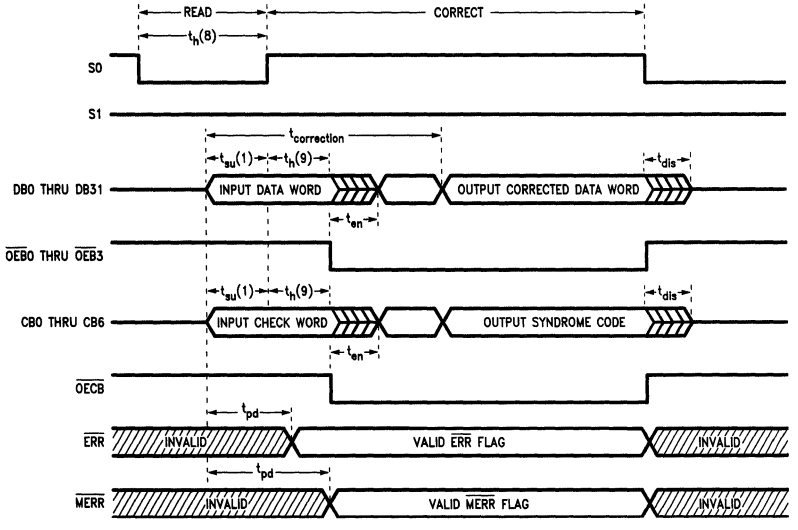
Symbol	From (Input)	To (Output)	Test Conditions	Military			Commercial			Units
				Min	Typ†	Max	Min	Typ†	Max	
t_{pd}	DB and CB	$\overline{ERR}$	S1 = H, S0 = L, R_L = 500Ω		26			26		ns
			S1 = L, S0 = H, R_L = 500Ω		26			26		
t_{pd}	DB and CB	$\overline{MERR}$	S1 = H, S0 = L, R_L = 500Ω		40			40		ns
			S1 = L, S0 = H, R_L = 500Ω		40			40		
t_{pd}	S0 ↓ and S1 ↓	CB	R1 = R2 = 500Ω		35			35		ns
t_{pd}	DB	CB	S1 = L, S0 = L, R1 = R2 = 500Ω		35			35		ns
t_{pd}	S1 ↑	CB	S0 = H, R1 = R2 = 500Ω		35			35		ns
t_{en}	$\overline{OECB}$ ↓	CB	S1 = X, S0 = H, R1 = R2 = 500Ω		18			18		ns
t_{dis}	$\overline{OECB}$ ↑	CB	S1 = X, S0 = H, R1 = R2 = 500Ω		18			18		ns
t_{en}	$\overline{OECB}$ ↓	DB	S1 = X, S0 = H, R1 = R2 = 500Ω		18			18		ns
t_{dis}	$\overline{OECB}$ ↑	DB	S1 = X, S0 = H, R1 = R2 = 500Ω		18			18		ns

DP8405 Switching Characteristics, V_{CC} = 4.5V to 5.5V, C_L = 50 pF, T_A = Min to Max

Symbol	From (Input)	To (Output)	Test Conditions	Military			Commercial			Units
				Min	Typ†	Max	Min	Typ†	Max	
t_{pd}	DB and CB	$\overline{ERR}$	S1 = H, S0 = L, R_L = 500Ω		26			26		ns
	DB	$\overline{ERR}$	S1 = L, S0 = H, R_L = 500Ω		26			26		
t_{pd}	DB and CB	$\overline{MERR}$	S1 = H, S0 = L, R_L = 500Ω		40			40		ns
			S1 = L, S0 = H, R_L = 500Ω		40			40		
t_{pd}	S0 ↓ and S1 ↓	CB	R_L = 680Ω		40			40		ns
t_{pd}	DB	CB	S1 = L, S0 = L, R_L = 680Ω		40			40		ns
t_{pd}	S1 ↑	DB	S0 = H, R_L = 680Ω		40			40		ns
t_{PLH}	$\overline{OECB}$ ↑	CB	S1 = X, S0 = H, R_L = 500Ω		24			24		ns
t_{PHL}	$\overline{OECB}$ ↓	CB	S1 = X, S0 = H, R_L = 680Ω		24			24		ns
t_{PLH}	$\overline{OEDB}$ ↑	DB	S1 = X, S0 = H, R_L = 680Ω		24			24		ns
t_{PHL}	$\overline{OEDB}$ ↓	DB	S1 = X, S0 = H, R_L = 680Ω		24			24		ns

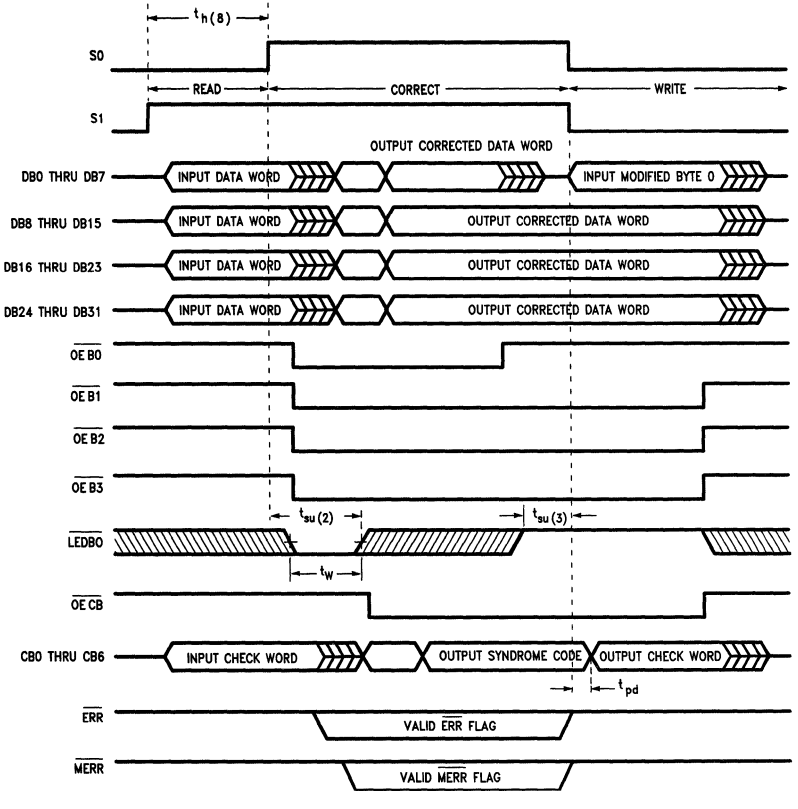
†All typical values are at V_{CC} = 5V, T_A = +25°C.

Switching Waveforms



TL/F/8535–6

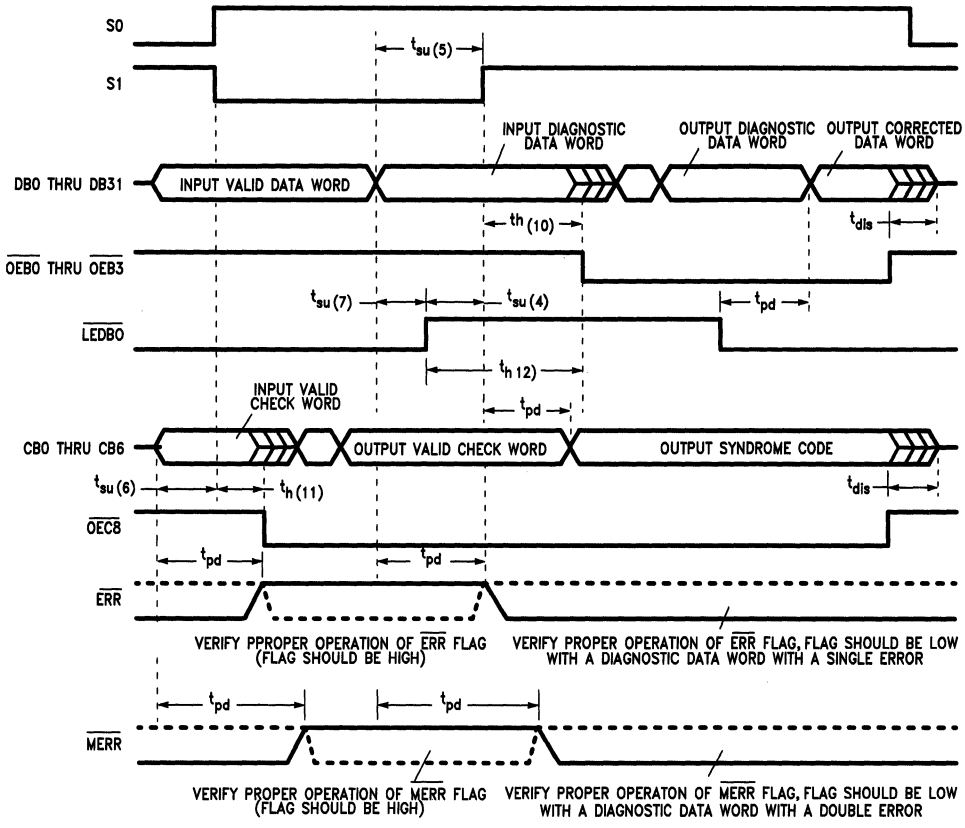
FIGURE 1. Read, Flag, and Correct Mode



TL/F/8535–7

FIGURE 2. Read, Correct Modify Mode

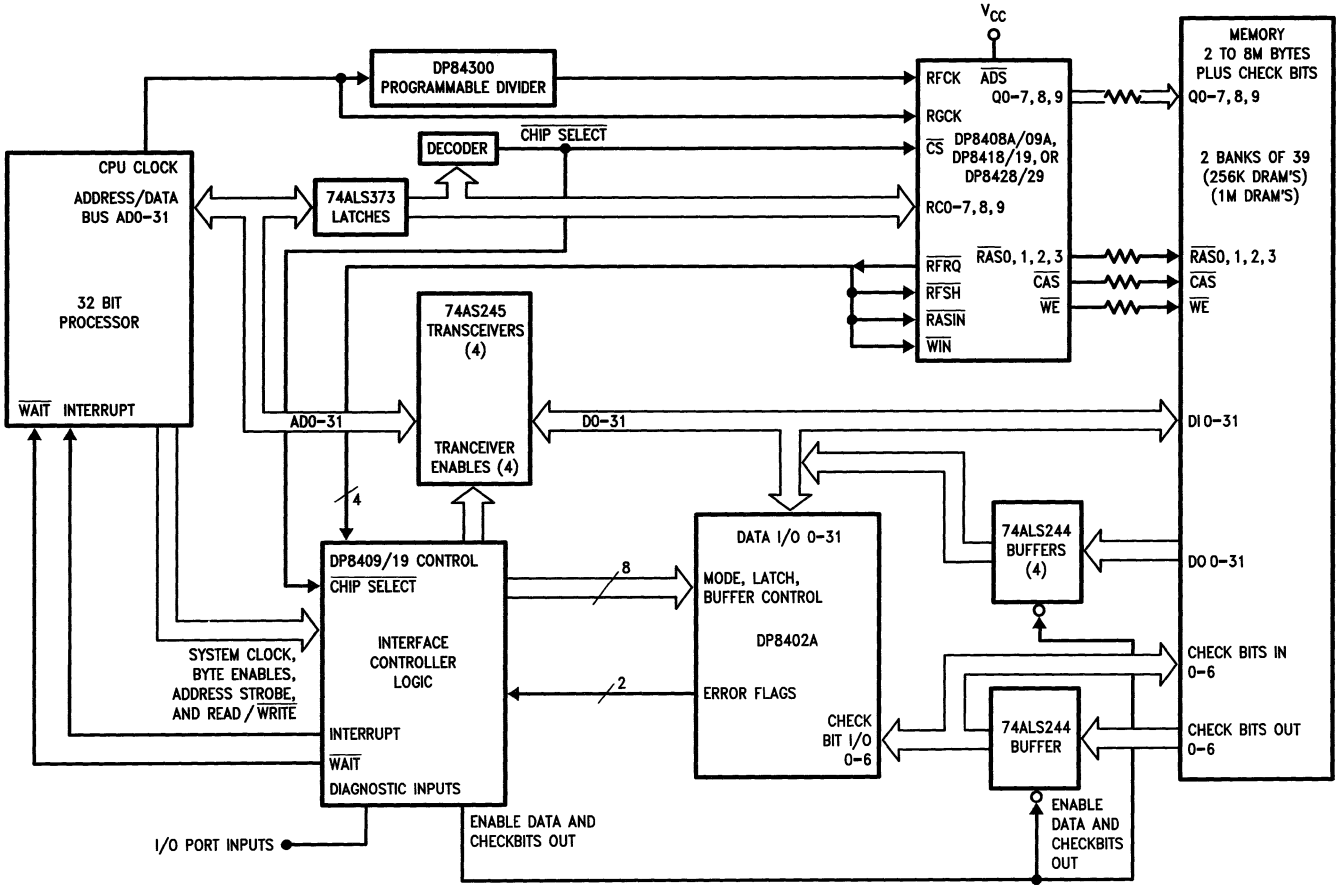
Switching Waveforms (Continued)



TL/F/8535–8

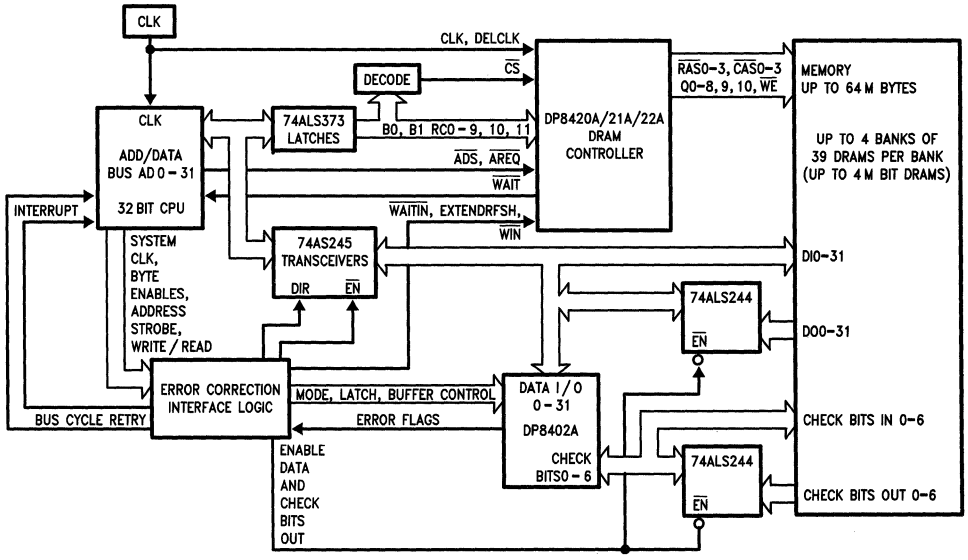
FIGURE 3. Diagnostic Mode

DP8402A Interfaced to the DP8418/19/28/29 System Diagram



TL/F/8535–9

DP8402A Interfaced to the DP8420A/21A/22A System Diagram



TI/F/8535–12

National Semiconductor

54F/74F632
32-Bit Parallel Error Detection and Correction Circuit

General Description

The 'F632 device is a 32-bit parallel error detection and correction circuit (EDAC) in a 52-pin or 68-pin package. The EDAC uses a modified Hamming code to generate a 7-bit check word from a 32-bit data word. This check word is stored along with the data word during the memory write cycle. During the memory read cycle, the 39-bit words from memory are processed by the EDAC to determine if errors have occurred in memory.

Single-bit errors in the 32-bit data word are flagged and corrected.

Single-bit errors in the 7-bit check word are flagged, and the CPU sends the EDAC through the correction cycle even though the 32-bit data word is not in error. The correction cycle will simply pass along the original 32-bit data word in this case and produce error syndrome bits to pinpoint the error-generating location.

Dual-bit errors are flagged but not corrected. These errors may occur in any two bits of the 39-bit word from memory (two errors in the 32-bit data word, two errors in the 7-bit check word, or one error in each word). The gross-error condition of all LOWs or all HIGHs from memory will be detected. Otherwise, errors in three or more bits of the 39-bit word are beyond the capabilities of these devices to detect.

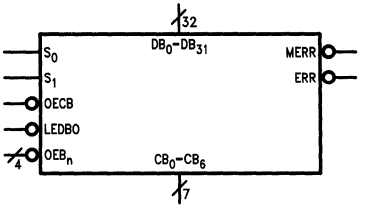
Read-modify-write (byte-control) operations can be performed by using output latch enable, $\overline{\text{LEDBO}}$, and the individual $\overline{\text{OEB}}_0$ through $\overline{\text{OEB}}_3$ byte control pins.

Diagnostics are performed on the EDACs by controls and internal paths that allow the user to read the contents of the Data Bit and Check Bit input latches. These will determine if the failure occurred in memory or in the EDAC.

Features

- Detects and corrects single-bit errors
- Detects and flags dual-bit errors
- Built-in diagnostic capability
- Fast write and read cycle processing times
- Byte-write capability

Logic Symbol



TL/F/9579–1

Unit Loading/Fan Out: See Section 1 for U.L. definitions

Pin Names	Description	54F/74F	
		U.L. HIGH/LOW	Input I_{IH}/I_{IL} Output I_{OH}/I_{OL}
CB_0–CB_6	Check Word Bit, Input	3.5/1.083	70 μA/−650 μA
	or TRI-STATE® Output	150/40 (33.3)	−3 mA/24 mA (20 mA)
DB_0–DB_{31}	Data Word Bit, Input	3.5/1.083	70 μA/−650 μA
	or TRI-STATE Output	150/40 (33.3)	−3 mA/24 mA (20 mA)
$\overline{OEB}_0$–$\overline{OEB}_3$	Output Enable Data Bits	1.0/1.0	20 μA/−0.6 mA
$\overline{LEDBO}$	Output Latch Enable Data Bit	1.0/1.0	20 μA/−0.6 mA
$\overline{OECB}$	Output Enable Check Bit	1.0/1.0	20 μA/−0.6 mA
S_0, S_1	Select Pins	1.0/1.0	20 μA/−0.6 mA
$\overline{ERR}$	Single Error Flag	50/33.3	−1 mA/20 mA
$\overline{MERR}$	Multiple Error Flag	50/33.3	−1 mA/20 mA

Expanding the Versatility of the DP8400

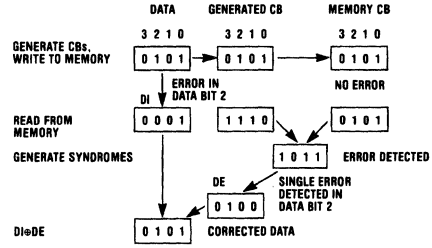
National Semiconductor
Application Note 306
Mike Evans

BASIC OPERATION OF THE DP8400

Introducing error correction capabilities to a memory incurs some penalties–extra memory, additional access times, and extra control circuitry. The DP8400 has been designed to minimize the last two, and for some data word widths, less extra memory is required than for other error correction circuits.

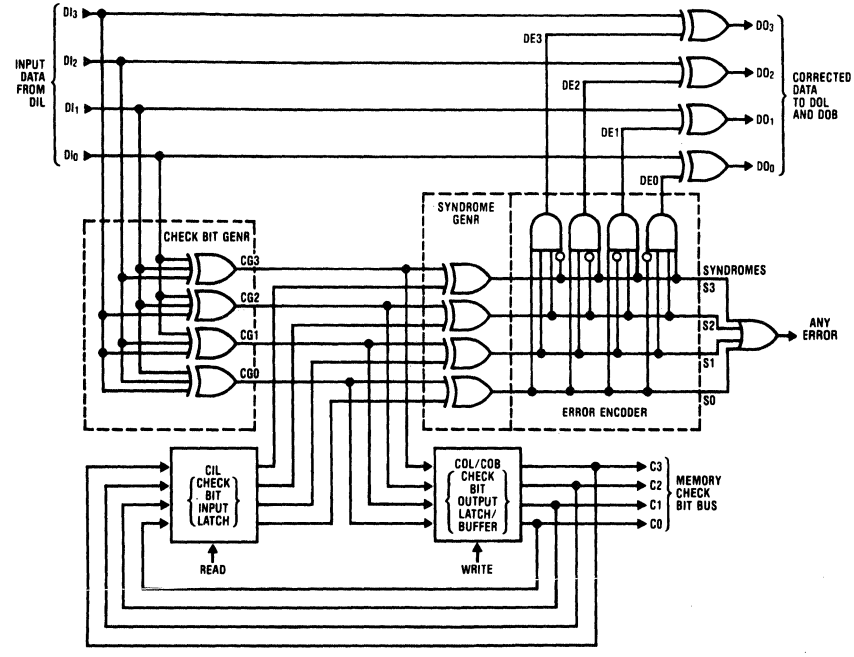
In systems using error correction, extra memory is needed for check bits, which are merely parity bits, each derived from different combinations of the data bits. If a single error does occur, the error correction circuit can determine which bit is in error and then complement that bit, to re-create the original data word. As the memory data word widens, the ratio of check bits to memory data bits is reduced. As a rough guide, starting with four data bits and four check bits, one additional check bit is required each time the data word doubles.

A circuit diagram of how the DP8400 generates the check bits in a write cycle and corrects errors in a read cycle is shown in *Figure 1a*, which uses four data bits and four check bits. A 4-bit example is shown in *Figure 1b*. In a write cycle, the data input latch, DIL, receives the system data and generates four parity bits or check bits, which pass through the check bit output latch, COL, and buffer, to be written to the selected memory location with the system data. This delays every write cycle, but fortunately the DP8400 takes only 30 ns extra to generate the (six) check bits. When this location is subsequently read, the four memory data bits pass through DIL to generate four new check bits. The four memory check bits pass through the check bit input latch, CIL, and are fed into four Exclusive-OR gates with the four generated check bits. The outputs of these gates are called syndrome bits, and obviously, if there are no errors, the two sets of check bits will be the same and no syndrome bits will go high. If there is an error in the check bits, only the corresponding syndrome bit will go high; in this case the data bits are still correct. If one of the data bits is in error, three syndrome bits will go high (in the case of DP8400, three or five will go high), and the syndrome word is unique for any of the bits in error. The four AND-gates decode which bit is in error and complement it out of the second set of Exclusive-OR gates. The other three output bits remain the same as the input bits, so the corrected word is now available to the system.



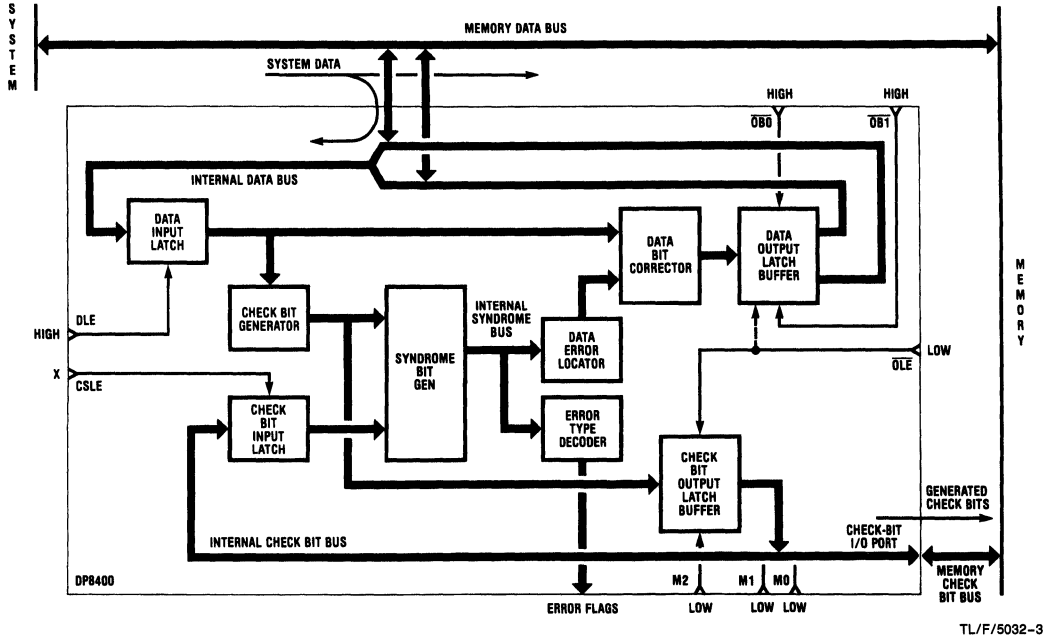
TL/F/5032–1

FIGURE 1b. Example of Single Error Correction



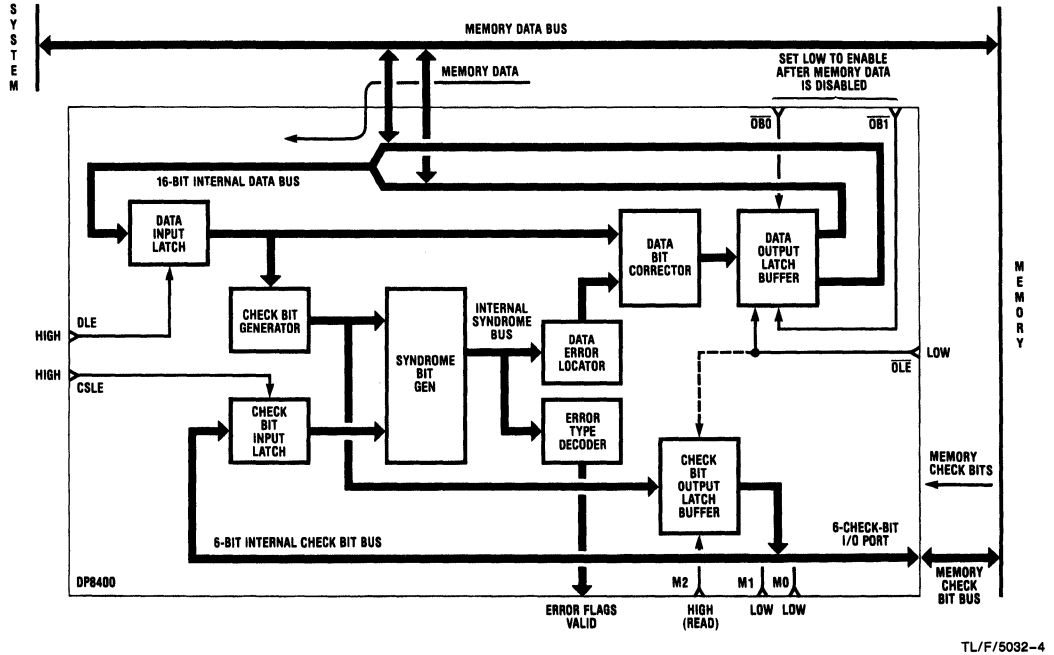
TL/F/5032–2

FIGURE 1a. Error Correction 4-Bit Functional Diagram



TL/F/5032-3

FIGURE 2a. DP8400 Write To Memory Cycle



TL/F/5032-4

FIGURE 2b. DP8400 Read From Memory Cycle

In the case of the DP8400 with 16 data bits and 6 check bits, there are 16 AND-gates to decode the 6 syndrome bits to determine the data bit in error. Table I shows the DP8400 matrix, called a Nelson Code, which has some unique features concerned with double soft error correction. For the purposes of this description, the matrix may be considered to be a form of Modified Hamming Code. The matrix has two functions: horizontally it tells us the value of the generated check bits for any data word when writing to memory, and vertically it tells us the syndrome word for any data bit in error. In a write cycle to memory, a '1' in any row indicates that the data bit in that column helps generate the parity bit in that row. For example, check bit 1 checks the parity of data bits 3, 6, 8, 9, 11, 13, 14 and 15, and generates even parity for those data bits. In a read cycle from memory, three or five of the six syndrome bits will go high for a single data bit error, and the columns represent the syndrome word, so the data bit in error is the number at the top of the column for that syndrome word. The 16 AND-gates each decode one of the 16 syndrome words shown in the columns of Table I, to locate the error. If there is a data bit error, one of the outputs of the 16 AND-gates will go high, to complement the data bit in error.

If two errors have occurred, the syndrome word is simply the Exclusive-OR of the syndrome words of the two individual bits in error, whether data or check bits, and is always even parity. First, if two check bits are in error, the corresponding two syndrome bits will go high. Second, for one data bit and one check bit error, then either two, four or six syndrome bits will go high. Finally, if two data bits are in error, again two, four or six syndrome bits go high. Thus a parity on the syndromes will indicate any two errors. This is important because if we know there are two errors, the DP8400 can attempt to correct them. The third error flag, E1, is the parity of the syndrome bus and check bit error. The DP8400 provides three error flags AE (Any error), E0 and E1, as shown in Table II, so that the exact nature of the error can be determined.

CONFIGURATION AND CONTROL OF THE DP8400

The DP8400 has a 16-bit data I/O port and an 8-bit check bit I/O port (6 bits used with 16 data bits) for applications with memories used with 16-bit microprocessors. The 16-bit data I/O port sits on the memory data bus, and the 6 check bit I/O port connects directly to the check bit section of memory. In other words, each memory location now contains 16 data bits with 6 check bits. The DP8400 is expandable to beyond 80 data bits, each additional 16 data bits requiring an additional DP8400 without the need for extra logic circuitry. 32-bit wide memory busses are also a popular width for minicomputers. In addition, 16-bit microprocessor systems may use 32-bit memory, because this larger memory data width requires only 7 check bits, a lower percentage overhead of check bits to data bits.

Figures 2a and *2b* show a simplified block diagram of the DP8400 with its control signals. The numerous control signals provide ease of use in the many varied applications of this chip. There are three latch enable signals DLE, CSLE and $\overline{\text{OLE}}$. Whenever DLE is high, data on the data I/O port D0–15 is entered into the data input latch DIL, and is latched in as DLE goes low. This allows either processor or memory data to be present on the data bus for only 3 ns prior to, and held over for 10 ns after DLE goes low. The data can then be removed if desired. Similarly, CSLE, when high, allows check bits on the check bit I/O port and external data on the syndrome I/O port to enter the check bit and syndrome input latches (CIL and SIL), respectively. These are latched in as CSLE goes low. (In 16-bit operation, $\overline{\text{OES}}$, Output Enable Syndromes, will be set low permanently, inhibiting CSLE to SIL, which remains in the power-up reset condition so that it does not affect the simplified block diagram.) $\overline{\text{OLE}}$, when set low, allows internal information into the data and check bit output latches (and the syndrome output latch, not shown). As $\overline{\text{OLE}}$ goes high, this information becomes latched. For some less complex designs, DLE, CSLE and $\overline{\text{OLE}}$ may be linked together. Providing $\overline{\text{OLE}}$ was low to allow corrected data into DOL, then $\overline{\text{OB0}}$ and $\overline{\text{OB1}}$, when set low, enable the two data output buffers to present corrected data to the system. Data is enabled or disabled within 15 ns of these inputs going low or high, respectively.

The DP8400 has three mode pins, M2, M1 and M0, which offer eight major modes of operation, designated 0 to 7. The most important two are Normal Write and Normal Read, and for these M1 and M0 are set low. M2 is READ/$\overline{\text{WRITE}}$ so Normal Write is mode 0 and Normal Read is mode 4. Other modes are used for the Double Complement Correct approach (Modes 1, 3 and 5) and for diagnostics (Modes 2 and 6). Mode 7 used when expanded to more than 16 data bits and fast correction times are required.

NORMAL OPERATION WITH A 16 DATA BIT MEMORY

The basic requirements for normal operation of the DP8400 are that it generates check bits, detect errors and correct them with minimum delays, and that it be easy to use. In normal operation M1 and M0 are set low. *Figure 2a* shows how the DP8400 generates check bits when writing data to memory. DLE may be kept high, $\overline{\text{OLE}}$ low, CSLE low, and M2 low so that the DP8400 is in Mode 0. System data is presented to the data I/O port on pins D0–15, and enters DIL, where it connects to the check bit generator CG. The six generated check bits pass through COL and are enabled (with M2 low) onto the check bit I/O port. The six generated check bits will appear 30 ns after the 16 data bits are presented to the data I/O port. A write to memory will now store the 16 data bits and 6 corresponding check bits in the selected location of memory. The write cycle is therefore slowed down by 30 ns, which in most memory systems is not significant.

Figure 2b shows the paths when reading from memory, with DLE set high to enter the memory data bits into DIL, and CSLE also set high to enter memory check bits into CIL. M2 is set high so that the DP8400 is in Mode 4. The Any Error flag, AE, becomes valid 35 ns after memory data and check bits are valid. Error flags E1 and E0 become valid approximately 15 ns later. Thus, if AE is low, no further operations are necessary. For fast 16-bit microprocessor systems, it may be necessary to introduce a wait state every read cycle to first determine if an error exists. If no error is detected the wait state is removed and the read cycle continues.

If an error is detected, then the error flags E1 and E0 must be examined to determine the required action. If the error is a single data bit error, DOL will by now contain corrected data. If there is no check bit error, then COL, which follows CIL when in Mode 4, now contains the original check bits. By taking $\overline{\text{OLE}}$ high, corrected data bits are latched in DOL, and correct check bits in COL. The memory is now disabled, so that $\overline{\text{OB0}}$ and $\overline{\text{OB1}}$ can be set low to enable corrected data onto the data bus, and M2 set low to enable the contents of COL onto the check bit bus. A write to the same location of memory will therefore remove the data bit error if

it was a soft error. The microprocessor can read the corrected data once the wait signal is removed.

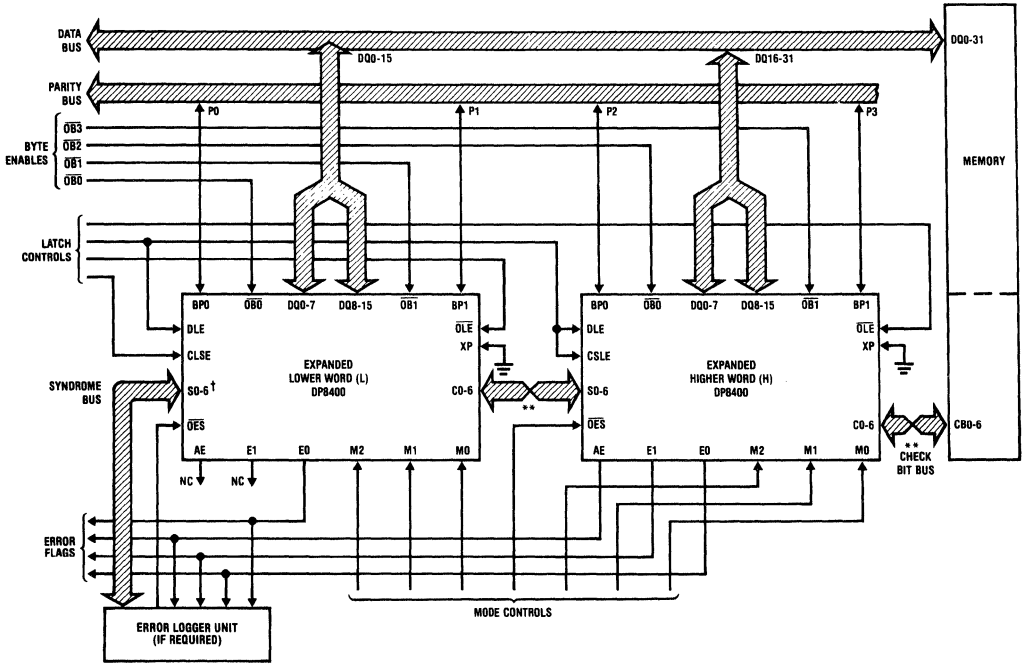
If the error is a single check bit error, DLE should be set low. DOL contains the contents of DIL, still correct data. Memory can now be disabled so that $\overline{OB0}$ and $\overline{OB1}$, when set low, output correct data, and M2 when set low, allows the generated check bits from DIL to be output on the check bit I/O port. A write to the same location of memory will remove the check bit error if it was a soft error. The microprocessor now reads this correct data when the wait signal is removed. If a double bit error is detected, then other approaches may be taken, as described in the data sheet and later in this application note.

The primary features of the DP8400 are discussed in the data sheet; there are, however, a number of other features that become very useful once a designer becomes acquainted with error correcting techniques.

These include: expansion beyond 16 data bits, diagnostic routines, error logging (allowing some double error correction), and a novel approach offering fast correction of any double error. This application note discusses how the DP8400 has been designed to function in all of these applications, making it the most versatile and comprehensive error correction chip available.

ERROR CHECKING AND CORRECTING FOR WIDER-THAN-16-BITS DATA WIDTHS

At present, most 16-bit microprocessor systems use a 16-bit wide main memory, partly for simplicity, and also because main memories, in general, have not become large enough in size to justify otherwise. The data sheet shows how to accomplish this with one DP8400, utilizing the matrix of Table I. It is fairly easy to use a memory of twice the microprocessor data width to reduce total chip count when incorporating error correction capability. One example would be a complex 8-bit microprocessor using large main memory. If the memory data width is kept at eight bits, then five check bits are required for error correction for each byte of data. If four banks of memory are required, each bank comprising 13 chips, then 52 total memory chips are required and only 62% of the memory is used for system data. If the memory data width is increased to 16 bits for the same microprocessor-based system, then six check bits are required.



TL/F/5032–5

†Refer to discussion in "Other Modes of Operation" under Clearing SIL.

** Connection sequence must be done according to Table VIII.

FIGURE 3. 32-Bit Error Detection and Correction

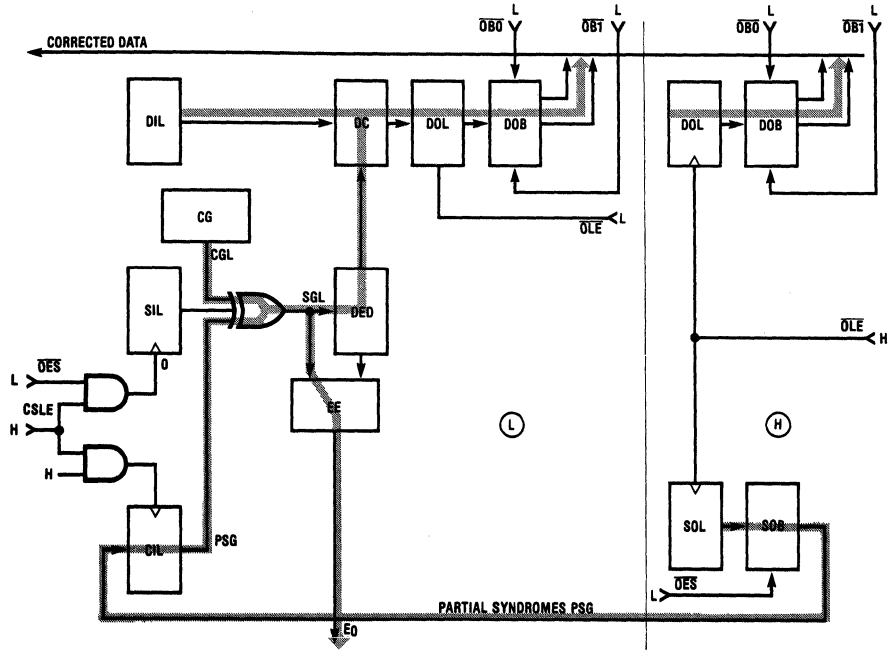
The memory now comprises two banks each of 22 chips, totaling 44 memory chips–a savings of eight memory chips. This saving is offset somewhat by the need to incorporate byte-writing capability, which does require extra components and slows down the memory write cycle. One DP8400 is still needed, using all 16 bits, and two bidirectional buffers are also required.

As a second example, using a 16-bit microprocessor with a memory of eight banks, each comprising 16 bits of data and six check bits, the total is 8 × 22 or 176 memory chips. Once the memory is widened to 32 data bits with seven check bits, only four banks are required, and the total number of memory chips reduces to 4 × 39, or 156—a savings of 20 memory chips. This is offset a little by the fact that an extra DP8400 is required, and slightly slower memory write and read cycles are necessary. In some cases, therefore, widening the memory data bus becomes more practical for large memories.

Saving memory chips is just one reason why there is a need to be able to expand the DP8400 beyond 16 data bits. Most minicomputers now use 32-bit wide data busses, and soon there will be some 32-bit microprocessors. Other systems use 24 bits, 48 bits, 52 bits, 64 bits or a variety of other data widths. The DP8400 has been configured to be expandable to any data width, even beyond 80 bits, merely by inserting an additional DP8400 for each 16-bit increment in memory data.

A section of the chip shown in the data sheet Block Diagram comprises the syndrome input and output latches, SIL and-SOL, and a dedicated syndrome I/O port. This port has a number of uses not normally needed in simple 16-bit single error correction applications.

One use of this syndrome port is for data widths wider than 16 bits. Only one DP8400 is required with 16 data bits or less, but if a system uses more than 16 memory data bits, additional DP8400s are required. For example, two DP8400s, one with its 16-bit data port connected to the lower word, and the other to the higher word, can be configured to generate check bits, and detect and correct errors for a 32-bit memory as shown in *Figure 3*. For writing to memory, both chips will still generate six check bits from the two words of 16 bits. But with more than 26 total data bits, seven check bits are required. Therefore, it is necessary to combine the two sets of check bits to produce seven composite check bits to be written to memory as shown in the flow path depicted in *Figure 4a*. This is achieved by outputting the six generated check bits from the lower word DP8400 (designated L), and inputting them to H, the higher word DP8400. The syndrome port of H is available to receive these check bits from L, to be loaded into SIL of H, provided CSLE is high. The six outputs from SIL combine with the six check bits generated in H to create seven composite check bits, and this 7-bit combination is output on the check bit port to the memory check bits. Table II shows one of twelve possible ways to combine the two sets of check bits. Note that the lower word matrix for bits 0 and 15 is identical to Table I with the addition of all "0"s for the seventh check bit. The higher word matrix for bits 16 to 31 uses the same rows but in a different order, implying that the check bits from L must be cross-connected to H. For example, memory check bit 5 is generated from check bit 1 of L and check bit 5 of H. Both chips are therefore set to normal write mode when generating check bits.



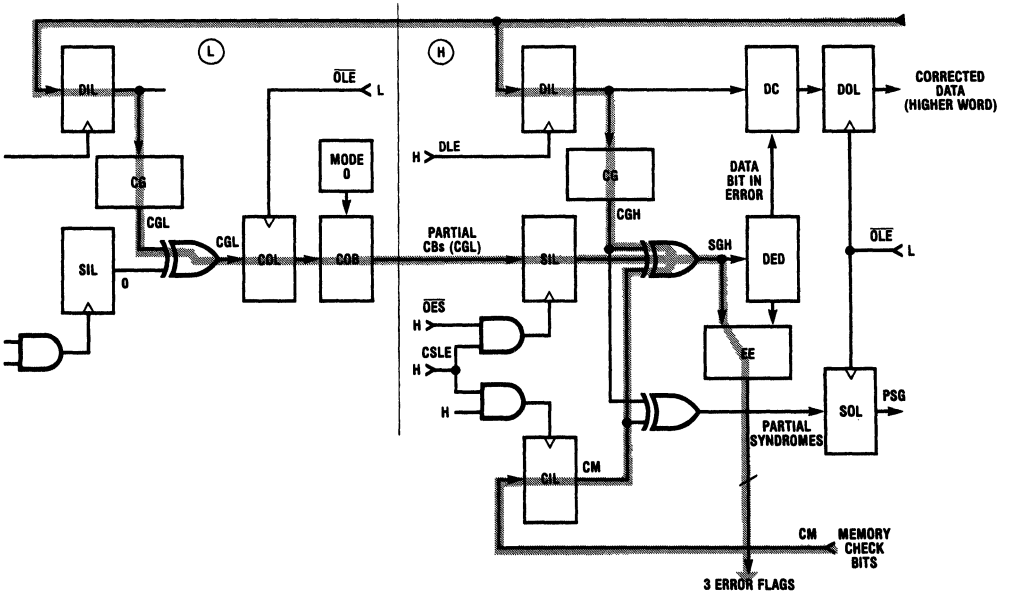
TL/F/5032–6

FIGURE 4a. E²C² 32-Bit Configuration, Error-Correct Flow Path

When reading from memory, the two chips first need to detect for an error. *Figure 4b* shows the flow path through the chips. L is set to normal write mode and H to normal read mode. Memory data is supplied to both chips so that L generates six check bits from the lower word data bits, and feeds them to SIL of H, the same as for writing. H also generates its own check bits which combine with those from L, and these seven composite check bits are compared with the seven memory check bits fed into CIL of H. This combining, plus comparison of check bits, is equivalent to seven 3-input Exclusive-OR gates. The output of these Exclusive-OR gates are the seven syndrome bits, and these can be decoded to determine the type of error. First, if there is no error, error flag AE of H will remain inactive because memory data is correct, provided $\overline{OLE}$ is kept low, and DOL of both L and H will contain correct data. Second, if there is a memory check bit error, only one of the seven syndromes will go high and the three error flags of H will indicate a check bit error as in Table III. Note that memory data is still correct, and with $\overline{OLE}$ low, DOL of both L and H contain correct data. Third, if there is a single data error in bits 16-31, the syndromes of H are such that the data error locator will locate the error and correct it, so again DOL of both L and H contain correct data. This is because the seventh syndrome bit is low for an error in the higher word, so that we have a six syndrome bit word as in Table I, to be decoded as normal to correct the error. In each of these three cases, DOL of both L and H contained correct data, and the common condition for these is either that AE(H) is "0", or E1(H) is "1".

The fourth case is more complex. In the previous three cases, correct data has been available in both DOL about 50 ns after memory data became valid. Now with a single data error in bits 0–15, AE(H) is a "1", E1(H) a "1", and EO(H) a "0", but L does not have sufficient information to locate the error. It is first necessary to feed back the partially generated syndromes of H back to L, and this is achieved by reversing the direction of the common bus. First L is placed in normal read mode so that L's generated check bits become disabled. Next, the partial syndromes in H are enabled onto the bus by setting $\overline{OES}$ of H low, so that its syndrome I/O port outputs the combined Exclusive-OR of CG(H) and CIL(H), which is transferred to CIL of L. These partial syndromes then combine with CG(L) to generate valid syndrome bits in L, demonstrated by the flow path of *Figure 4c*. If there is, in fact, a data bit error in bits 0–15, the seventh syndrome bit will go low, allowing the remaining six bits to be decoded to locate the error as per the columns of Table II. This switching around of the common bus, therefore, takes more time to correct the error in L, equivalent to a total time of approximately 100 ns. The fifth kind of error is identified as a double error. In this case, the error flags indicate the double error and the system can take the necessary action.

A logical approach when using two DP8400s would be to first see if there is any need to reverse the common bus by monitoring AE(H), and when it is low, to output directly from DOL of both chips by setting $\overline{OB0}$ and $\overline{OB1}$ of each low. The System Data Valid flag should be set active at this time. If the AE(H) output is high and the error flags do not indicate a double error, then the common bus should be switched around and the System Data Valid signal set true. If the error is a double error, the user may utilize a number of alternatives, including the Double Complement Correct method.



TL/F/5032–7

FIGURE 4b. E²C² 32-Bit Configuration, Detect Flow Path

TABLE I. Data In to Check Bit Generate, or Data Bit Error to Syndrome-Generate Matrix (16-Bit Configuration)

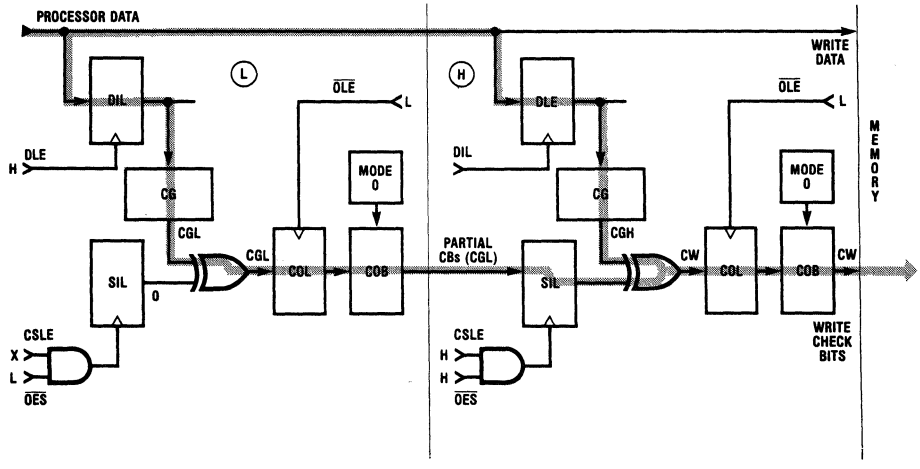
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
		GENERATE CHECK BITS																DQ0–15
	0	0	0	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0
	1	0	0	0	1	0	0	1	0	1	1	0	1	0	1	1	1	1
GENERATED	2	1	0	0	1	1	0	0	0	1	0	1	0	1	1	1	1	2*
SYNDROMES	3	0	1	1	0	0	0	0	1	1	1	1	0	1	0	1	1	3*
	4	1	1	0	0	0	1	0	1	1	0	0	1	0	1	0	1	4
	5	1	1	1	0	1	1	1	0	1	0	0	0	1	1	1	0	5
		4	8	9	7	5	1	3	9	E	B	D	3	C	7	F	F	0
		3	3	2	0	2	3	2	1	3	0	0	1	2	3	2	1	1
		HEXADECIMAL EQUIVALENT OF SYNDROME BITS																

*C2, C3 generate odd parity.

TABLE II. Data Bit Error to Syndrome-Generate Matrix (32-Bit Configuration)

		L															H																
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	DQ0–31
	0	0	0	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0	0	0	1	0	0	1	0	1	1	0	1	1	0	1	1	1
	1	0	0	0	1	0	0	1	0	1	1	0	1	0	1	1	1	1	1	1	0	1	1	1	0	1	0	0	0	1	1	1	0	5
	*2	1	0	0	1	1	0	0	0	1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6
SYNDROMES	*3	0	1	1	0	0	0	0	1	1	1	1	0	1	0	1	1	0	1	1	0	0	0	0	1	1	1	1	0	1	0	1	1	3
	4	1	1	0	0	0	1	0	1	1	0	0	1	0	1	0	1	1	1	0	0	0	1	0	1	1	0	0	1	0	1	0	1	4
	5	1	1	1	0	1	1	1	0	1	0	0	0	1	1	1	0	1	0	0	1	1	0	0	0	1	0	1	0	1	1	1	1	2
	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	1	1	1	0	1	1	0
		4	8	9	7	5	1	3	9	E	B	D	3	C	7	F	F	2	A	A	1	2	2	3	8	B	9	8	1	A	3	B	9	HEX
		3	3	2	0	2	3	2	1	3	0	0	1	2	3	2	1	3	1	4	6	6	5	4	5	3	4	6	5	2	7	6	7	1

*CG2, CG3 generate odd parity.



TL/F/5032–8

FIGURE 4c. E²C² 32-Bit Configuration, Write Flow Path

TABLE III. Error Flags After Normal Read (32-Bit Configuration)

AE (H)	E1 (H)	E0(H)	E0(L)*	Error Type
0	0	0	0	No Error
1	1	0	0	Single check bit error
1	1	1	0	Single data bit error (H)
1	1	0	1	Single data bit error (L)
1	0	0	0	Double bit error
	All Others			Invalid conditions

*E0 (L) is valid after transfer of partial syndromes from higher to lower.

This approach to wider data width error detection and correction is termed the cascade configuration, and it requires only the one additional DP8400. The cascade approach can be used with up to five DP8400s controlling 80 data bits. The advantage is that only one additional DP8400 is required per 16 data bits, although write and read times become progressively slower as the number of DP8400s is increased. This is because of the time taken for the generated check bits to ripple through from the lowest to highest chips when writing and detecting, and then ripple back the other way for correcting.

In many memory systems, speed is of utmost importance and for faster systems, it is possible to connect the DP8400s in a parallel configuration using additional ICs. Application Note AN-308 describes this approach in detail.

The user may, therefore, select one of these approaches (or a combination of both) for systems using memory data widths of more than 16 bits.

DIAGNOSTIC CAPABILITIES OF THE DP8400

The DP8400 has been designed with system fault diagnois in mind. In fact, it is possible under microprocessor control with the DP8400 in site on the memory board to fully test every gate inside the DP8400 activated in normal operation, and also to diagnose all memory check bits. The DP8400 has two main diagnostic modes–modes 2 and 6. In other words, with M1 set high and M0 set low, information can be written to or read from the chip.

Mode 6 allows the memory check bits to be read onto the higher byte bits 8–14, and syndromes to be read on the lower byte bits 0–6, as shown in *Figure 5a*. The remaining two bits, 7 and 15, are the error flags E1 and E0 that were valid when mode 6 was entered. The syndrome bits will be the internally generated syndromes if $\overline{OES}$ is low (mode 6A), or external syndromes input on the syndrome I/O port if $\overline{OES}$ is high (mode 6B). The external syndromes could be obtained from an error logger/syndrome injector unit—this is an error logger with the capability of injecting syndromes back to the DP8400. Therefore, by being able to read the externally stored syndromes, the microprocessor can monitor or store the syndromes whenever needed.

Mode 2 transfers system data from the higher byte into CIL, instead of DIL, to simulate check bits. This can be used in three ways. First, as shown in *Figure 5b*, the simulated check bits can be latched in CIL by taking CSLE low. If the DP8400 is now set to normal read, mode 4, and new data is presented then, provided DLE is high and CSLE is kept low, the DP8400 will perform a normal read operation as if it were reading memory check bits. The results of this simulated read may be checked by enabling DOL to see if an error (if inserted) was corrected. Or as a further check, by entering mode 6, the predicted generated syndromes and error flags may be checked. Second, also while in mode 2, the simulated check bits appear at the check bit port (from the data bus higher byte) available to be written to the check bit portion of memory as shown in *Figure 4c*. $\overline{OLE}$ is set high before the original simulated check bits are removed and then memory data is subsequently placed on the data bus. A write to memory will now write known data and simulated check bits to the selected location. By writing known data to the memory check bits in mode 2, and then reading the memory check bits in mode 6, each check bit in each location can be validated. Third, it is possible in mode 2 with $\overline{OES}$ low to transfer data from the higher byte to the syndrome I/O port, also shown in *Figure 5c*. But first the generated check bits must be all low. This is attained by previously loading all "1"s into DIL in an earlier cycle. This is useful when using an error logger in conjunction with the DP8400 to feed the syndrome word into the logger whenever an error occurs.

ERROR LOGGING WITH SYNDROME INJECTION CAPABILITY

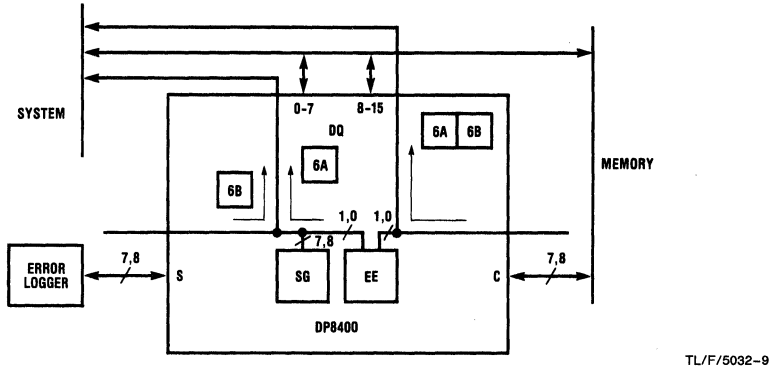
An important application of the dedicated syndrome I/O port is for error logging. This is because the internally generated syndromes derived during reading are available on this port, provided $\overline{OES}$ is set low. These syndromes indicate the exact location of a single error, whether it is in the data bits or check bits; they are therefore useful to be stored for error logging. Every time an error occurs when indicated by error flag AE, the syndromes corresponding to this error can be logged.

The syndrome word can be fed from SOL via the Syndrome Output Buffer onto the external syndrome bus. An Error Logger connected to this bus, as shown in *Figure 6*, will store the syndrome word in the same location as the corresponding address of each error that ocurs. An intelligent error logger will differentiate between new errors and ones that have occurred previously, by logging only new errors and ignoring ones that have already occurred. An easy way to determine this would be to compare the incoming memory address with the address of errors contained in the logger. If a match is not found and an error occurs, the new address and corresponding syndromes are logged. If a match is found, then whether an error occurs or not,

no further action is necessary. Tag bits may be provided to indicate whether the error is hard or soft.

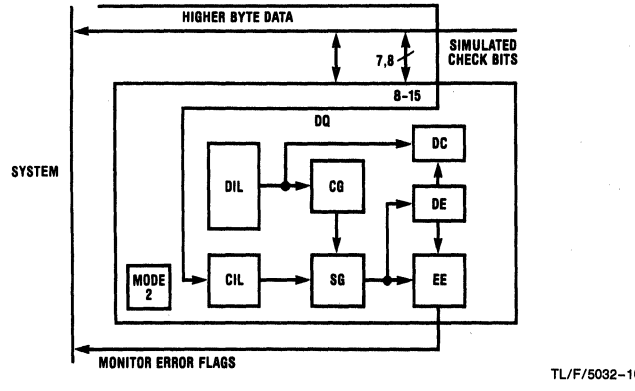
For example, if an error has already been logged at a particular address and that address is re-written to, then if the error repeats subsequently, it is a hard error, and if not, it is a soft error. So, if a tag bit is set when a write occurs to a previously logged address and a subsequent error is detected at that address, a second tag bit is set indicating a hard error. A better approach would be to have the DP8400 correct and rewrite to the same location all in the same cycle, as soon as a single error is detected. The first error detected in a location is classified as a soft error until it recurs, and if an error does recur, a tag bit is set to indicate a hard error. It is assumed here that multiple soft errors will not occur in the same location.

Now that the error logger contains error information, it is necessary for the microprocessor to retrieve it. The DP8400 makes this easy, because the external syndrome bus data can be transferred to the data bus as described for operation in mode 6. If the error logger is made capable of outputting stored syndromes, and subsequently outputting the corresponding address one byte at a time, then all the relevant information can be retrieved by the microprocessor. The user may choose to store this in nonvolatile memory in the event of a power failure. When power returns, it will be desirable to restore this information back to the error logger, and this can be achieved by first loading DIL with all "1"s to create all generated check bits low. Now the addresses and syndromes can be loaded from the higher byte of the microprocessor through the syndrome I/O port one byte at a time, with DP8400 in mode 2, to the error logger.



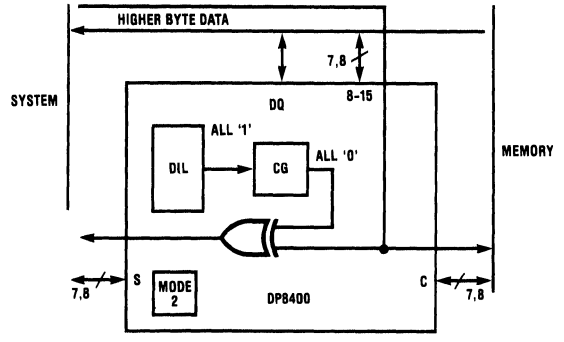
TL/F/5032–9

FIGURE 5a. Read Internal Generated Syndromes and Check Bit Port (Mode 6A) or Read Syndrome Port and Check Bit Port (Mode 6B)



TL/F/5032–10

FIGURE 5b. Diagnostic Read – Compare Simulated Check Bits with Check Bits Generated from Data Stored in Previous Cycle

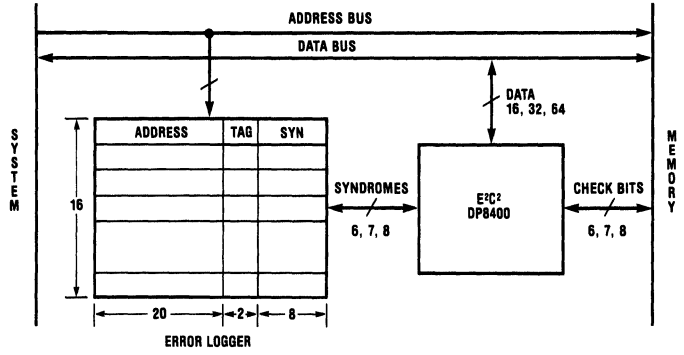


TL/F/5032–11

1) DIAGNOSTIC WRITE: WRITE HIGHER DATA BYTE TO CHECK BIT BUS (MODE 2)

2) TRANSFER HIGHER DATA BYTE TO SYNDROME BUS (MODE 2, PREVIOUS CYCLE LATCHED ALL '1's IN DIL TO MAKE CG = 0)

FIGURE 5c. DP8400: Mode 2



TL/F/5032–12

FIGURE 6. Error Logger Connected to DP8400 Syndrome Port

CORRECTING DOUBLE ERRORS USING THE ERROR LOGGER

It is possible to take the error logging function one stage further. As described so far, the error logger has been storing single errors (data bit or check bit). What if a double error is detected? If it is detected without any previous history at that address, one solution would be to perform a Double Complement to attempt to correct both errors. If this is not done, no useful information can be obtained. If both errors are corrected, the error logger records the syndromes of both, and tags whether they were both hard, or one hard and one soft. But, if there is a previous history at this address of a single error, then it is fair to assume that the second error has subsequently occurred. In this case, if the error logger could be made to inject the syndromes of the first error into the DP8400, the DP8400 would correct this error so that its DOL would then contain data with one error (if both errors are data bit errors). It is necessary at this point to wrap-around DOL back to DIL and allow the DP8400 to correct the second error. This approach is much faster than the Double Complement approach and at the same time offers full error logging capability.

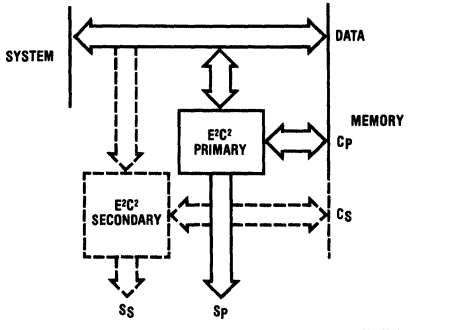
ANY DOUBLE ERROR CORRECTION USING THE DOUBLE SYNDROME DECODE APPROACH

The data sheet shows how the DP8400 can perform double error correction using the Double Complement Approach, provided at least one of the errors was hard. For very large memories, this may not be adequate, as some systems will require total double error correction capability–quickly, without having to wait two additional memory cycles. Some of these systems will also require triple error detect capability. Fortunately, the matrix of the DP8400 has been configured to allow both of these capabilities. Most modern error detection/correction matrices use a modified version of Hamming's original code. The Hamming code allows single errors to be corrected, however, two errors may not be detected as such. For 16 data bits, five check bits are required. Modified Hamming codes allow double error detect capability, as well, by arranging that the Exclusive-OR of the syndrome words of any two bits in error produces an even parity syndrome word. A parity check on the syndrome bus will, therefore, indicate two errors (or no error, but in this case, the Any Error flag will be inactive). For 16 data bits, six check bits are required for single/double error detect and single error correction capabilities.

The DP8400 has a matrix that goes one step further by using a version of the Nelson code. This costs no additional on-chip gates to those required for a Modified Hamming code. To be able to correct any two errors, it is necessary to be able to determine their location, and no present version of the Modified Hamming code is able to do this. There are matrices that do exist that can generate 12 check bits from 16 data bits (or 14 check bits from 32 data bits) for writing, and then generate 12 (or 14) syndrome bits when reading, so that the location of both errors can be determined and corrected. But, because most applications do not require this degree of integrity and associated expense, they are not very popular. It would be ideal if two DP8400s could be configured as in *Figure 7a*, with each generating a different set of check bits and a different set of syndrome bits so that the double syndrome word could be unique and decodable for any two bits in error. Fortunately, National Semiconductor has achieved this by incorporating a feature called the Rotational Syndrome Word Generator, which uses rotated data to the secondary DP8400.

The primary DP8400 generates check bits when writing, and syndrome bits when reading, as in a normal 16-bit system. But the data port of the secondary DP8400 receives data shifted by a number of bits, usually one bit. In other words, for this secondary chip, system data bit 0 connects to DQ1, system data bit 1 to DQ2, etc. Each DP8400 has its own dedicated six memory check bits, which are obviously different from each other due to the data shifting on the secondary DP8400. The Nelson code is such that during a read, not only does each DP8400 generate a different set of syndrome bits, but the double syndrome word (comprising 12 bits for 16 data bits) is unique for any two bits in error. It is necessary to be able to output these syndromes as they occur and to do this, $\overline{OES}$ of both chips is set low during the time memory data is valid.

Now that we have a unique double syndrome word for any two bits in error, it is necessary to decode it to correct both errors. The easiest way to do this is to connect the double syndrome word to the address inputs of a registered PROM (a PROM with latchable data out) as shown in *Figure 7b*. In this example, 12 syndrome bits require 4k addressing capability, and 32k registered PROMs will be made available soon. Some of the addresses of the RPROM will be used for double errors and each address will be unique for any two bits in error. The corresponding data out could, therefore, contain one of the syndrome words. Double errors may be caused by two data bit errors, a data bit and primary check bit error, a data bit and secondary check bit error, a primary and secondary check bit error, or two errors in either primary or secondary check bits. In these cases, if the RPROM address stores the syndrome word for one of the two errors, this will be available at the output of the RPROM when enabled.



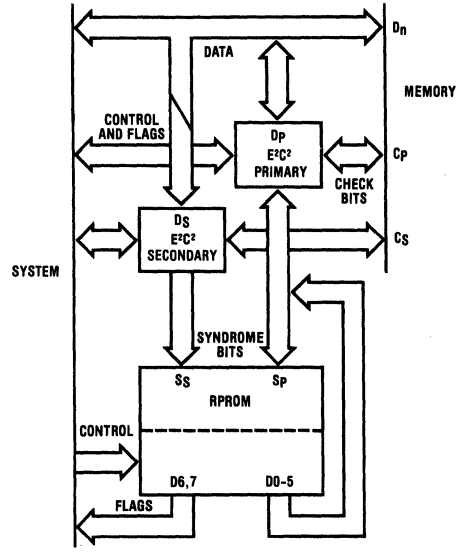
TL/F/5032–13

FIGURE 7a. 2 Different Generators

First of all, this data must be latched in the RPROM register, and then the $\overline{OES}$ input to each DP8400 must be set high to deactivate the two syndrome output buffers. Next, the RPROM data must be enabled onto the primary syndrome bus so the primary DP8400 can enter this syndrome word, representing one of the two bits in error with CSLE high. At the same time, the primary DP8400 must be set to mode 7 so that the syndrome word appears on the internal syndrome bus, replacing the generated syndromes. If $\overline{OLE}$ is now set from low to high, DOL will contain either one or no error, depending on where the two errors were located. In other words, the DP8400 has just corrected one of the errors. By setting $\overline{OLE}$ low, then disabling memory and enabling $\overline{OB0}$ and $\overline{OB1}$ of the primary DP8400, this data is output on the data bus and back into the DIL with DLE high. There is now only one data error, and this can be corrected by setting the DP8400 to normal read, mode 4.

Thus, both errors have been corrected at a fairly fast rate. For example, for a 50 ns RPROM, the total time to generate double syndromes, feed back a one-error syndrome word to the primary DP8400, correct it, wraparound, and correct again, may take less than 120 ns total.

Only a few of the addresses in the RPROM are required for double errors. Some double syndrome words represent single errors and triple errors. All single bit errors also produce a unique double syndrome word different from all double bit errors.



TL/F/5032–14

FIGURE 7b

In fact, nearly all triple bit errors produce unique double syndrome words different from single and double bit errors. Those that do not produce unique double syndrome words, duplicate syndrome words of other single, double, and triple bit errors; however, these comprise only about 5 percent of the total. We can say, therefore, that this approach will correct not only all double bit errors, but will detect 95 percent of all triple bit errors. Note that with error correction systems utilizing the modified Hamming code, the majority of triple bit errors are interpreted as single bit errors and falsely corrected as such. It is up to the designer to determine the chances of three errors occurring in a memory location, and the (likely) consequences that they will be falsely corrected. If this condition is undesirable, then the Double Syndrome Decode Method offers greatly enhanced integrity; in fact, if the three errors detected do have a unique double syndrome word, they can be corrected. As stated, no presently used Modified Hamming code offers a unique double syndrome word for multiple errors; this is only possible with a Nelson code. This example was largely for 16 data bits, but the idea will work for other data widths.

In the 16-bit example, the RPROM has to output only six bits representing the syndrome bits of a bit in error. This leaves two spare bits which can be used as flags, and the user can program his RPROM accordingly. One solution is to use these flags to indicate the type of action required—whether to correct at all, correct once, or correct twice by wrapping around.

BLOCK DIAGRAM OF THE DP8400

This Application Note discusses first the single error correction, showing a simplified block diagram of the chip for both a write cycle to generate check bits, and a read cycle to detect errors and correct single bit errors.

The most important requirement when accessing memory is that these operations be performed with minimal memory delays. The DP8400, therefore, has been structured internally to minimize series propagation delays through the chip. A full block diagram of the DP8400 is shown, and first impressions are that there might be excessive delays in the various paths due to the additional blocks that have been added to the basic functional block diagram. In fact, this is not the case, because the DP8400 has been configured in bipolar Schottky logic and uses the AND-OR-INVERT gate in many of the blocks. This type of gate structure is used in multiplexers, Exclusive-OR gates and fall-through latches. It is possible, therefore, to combine these functions into one wide gate, reducing the propagation delays through some of these blocks to that of one gate. For example, the check bit output latch COL receives its input from an Exclusive-OR gate followed by a multiplexer. These three functions can be combined into one wide gate, and this greatly reduces the time taken to generate check bits.

THE DP8400—A VERSATILE ERROR CHECKER/ CORRECTOR FOR ALL APPLICATIONS

It was shown earlier how the DP8400 was able to detect single and double errors, and correct single errors. For 8- and 16-bit systems, these could easily be accomplished with a minimum of extra circuitry. The DP8400 can also be used in complex high integrity systems. In fact, investigations are still progressing as to its immense capabilities. It is the only error correction circuit capable of these features, and yet it still provides very fast throughput. For these reasons, the DP8400 should become the industry standard error correction chip for the foreseeable future.

DP8400s in 64-Bit Expansion

National Semiconductor
Application Note 308
Chuck Pham

The purpose of this Application Note is to provide memory designers with detailed information on the DP8400 parallel expansion method. This method allows fast check bit generation, error detection, and error correction. A thorough understanding of the 16-bit implementation is a prerequisite. Included in this note are the following: error correction expansion matrix; detailed steps for check bit generation, error detection and error correction; an example of a single error correction; and the detailed wiring diagram for the 64-bit configuration.

The Error Correction Expansion Matrix

For a 16-bit word, the DP8400 reads data between the processor and memory, with its 16-bit bidirectional data bus connected to the memory data bus. The DP8400 uses an encoding matrix to generate six check bits from the 16 bits of data. This 16-bit matrix contains 16 unique syndrome patterns corresponding to each error location which allows the DP8400's Data Error Decoder (DED) to identify the data error location.

The DP8400 is easily expandable to other data configurations. For a 32-bit data word with seven check bits, two DP8400s are used. Three DP8400s can be used for 48 bits, four DP8400s for 64 bits, and five DP8400s for 80 bits, all with eight check bits. In order to expand the DP8400, additional check bits are required to provide the unique characteristic of the single data error syndrome. For expansion beyond 24 bits, check bits 6 and 7 (C6 and C7) are used. Note that these check bits can be configured to be always either zero or word parity, depending on the input voltage level of the Expansion Pin (XP). By rearranging all eight check bits (C0–C7) of each DP8400, we can obtain many different matrices that meet the above requirement. One of these is shown in Table I. For illustration, this matrix will be used throughout this application note to clarify the E^2C^2 expansion concept.

Check Bit Generation, Error Detection And Error Correction

CHECK BIT GENERATION (*Figure 1*)

In the Check Bit Generation mode, all four DP8400s are set to mode 0, normal write. The 64 bits of data from the system data bus are enabled into the Data Input Latches (DIL) of each DP8400. The individual Check Bit Generation (CG) of the four DP8400s then produce eight parity bits, or partial check bits, derived from the input data. (Note that all the syndrome input latches should be cleared so that only the partial check bits will pass through the Check Bit Output Latches/Buffers (COL and COB). In the normal write mode, the COBs are always enabled onto each check bit port. This allows the partial check bits to be combined externally in

TABLE I. Data Bit Error to Syndrome-Generate Matrix, 64-Bit Configuration

The partial code of device 0:

Error Locations (Data Bit Numbers)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	0	1	1	1	1	1	1	0	1	1	1	0	1	1	1	C0
0	0	0	1	0	0	1	0	1	1	0	1	0	1	1	1	C1
1	0	0	1	1	0	0	0	1	0	1	0	1	1	1	1	C2
0	1	1	0	0	0	0	1	1	1	1	0	1	0	1	1	C3
1	1	0	0	0	1	0	1	1	0	0	1	0	1	0	1	C4
1	1	1	0	1	1	1	0	1	0	0	0	1	1	1	0	C5
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C7

The partial code of device 1:

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
0	0	0	1	0	0	1	0	1	1	0	1	0	1	1	1	C1
1	1	1	0	1	1	1	0	1	0	0	0	1	1	1	0	C5
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C6
1	1	0	0	0	1	0	1	1	0	0	1	0	1	0	1	C4
0	1	1	0	0	0	0	1	1	1	1	0	1	0	1	1	C3
1	0	0	1	1	0	0	0	1	0	1	0	1	1	1	1	C2
0	0	1	1	1	1	1	1	0	1	1	1	0	1	1	1	C0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C7

The partial code of device 2:

32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C6
0	1	1	0	0	0	0	1	1	1	1	0	1	0	1	1	C3
1	1	1	0	1	1	1	0	1	0	0	0	1	1	1	0	C5
1	1	0	0	0	1	0	1	1	0	0	1	0	1	0	1	C4
1	0	0	1	1	0	0	0	1	0	1	0	1	1	1	1	C2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C7
0	0	0	1	0	0	1	0	1	1	0	1	0	1	1	1	C1
0	0	1	1	1	1	1	1	0	1	1	1	0	1	1	1	C0

The partial code of device 3:

48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	
1	1	0	0	0	1	0	1	1	0	0	1	0	1	0	1	C4
1	1	1	0	1	1	1	0	1	0	0	0	1	1	1	0	C5
0	1	1	0	0	0	0	1	1	1	1	0	1	0	1	1	C3
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	C6
1	0	0	1	1	0	0	0	1	0	1	0	1	1	1	1	C2
0	0	1	1	1	1	1	1	0	1	1	1	0	1	1	1	C0
0	0	0	1	0	0	1	0	1	1	0	1	0	1	1	1	C1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	C7

Check Bit Generation, Error Detection And Error Correction (Continued)

the eight 74S280s' parity generators/checkers to produce eight composite check bits. Table II. shows how these check bits are generated.

Table II. Composite Check Bits Generation

Ccomp. 0 = C(0)0 ⊕ C(1)1 ⊕ C(2)6 ⊕ C(3)4
Ccomp. 1 = C(0)1 ⊕ C(1)5 ⊕ C(2)3 ⊕ C(3)5
Ccomp. 2 = C(0)2 ⊕ C(1)6 ⊕ C(2)5 ⊕ C(3)3
Ccomp. 3 = C(0)3 ⊕ C(1)4 ⊕ C(2)4 ⊕ C(3)6
Ccomp. 4 = C(0)4 ⊕ C(1)3 ⊕ C(2)2 ⊕ C(3)2
Ccomp. 5 = C(0)5 ⊕ C(1)2 ⊕ C(2)7 ⊕ C(3)0
Ccomp. 6 = C(0)6 ⊕ C(1)0 ⊕ C(2)1 ⊕ C(3)1
Ccomp. 7 = C(0)7 ⊕ C(1)7 ⊕ C(2)0 ⊕ C(3)7

Notes:

Ccomp: composite check bit.

C(X)N: the partial check bit N of device X.

(Refer to Table I. for clarification).

To aid in fast error detection during memory read cycles, these composite check bits are complemented and written into memory along with the system data. If the system data has vacated the data bus, the Output Enables ($\overline{OB0}$ and $\overline{OB1}$) must be set low so that the original data word with its eight composite check bits can be written into memory.

DETECTION MODE (*Figure 2*)

In the Detection mode, again all the DP8400s are set to mode 0, normal write, then the partial check bits derived from the memory data bits are generated in a manner similar to that described for the check bit generation mode. These partial check bits are then associatively compared with the memory check bits in the eight 74S280s to produce eight external $\overline{\text{Composite Syndrome}}$ bits. As explained in the check bit generation mode, the composite check bits are complemented before being written into memory. This shows why complemented Composite Syndrome bits are produced instead of true composite syndromes. Then, if any bits on the $\overline{\text{Composite Syndrome}}$ bus go low, this will cause the 74S30 NAND gate to go high, giving the Any Error indication. If there is no error, all $\overline{\text{Composite Syndrome}}$ bits remain high. These $\overline{\text{Syndrome}}$ bits are also latched into the 74ALS533 Octal D-type Transparent Latch (with inverted output). The composite syndromes are then fed into the syndrome ports of the DP8400s in different combinations for each, for error-type determination and/or error correction.

CORRECTION MODE: (*Figure 3*)

Upon receiving the Any Error indication during the detection mode, it takes an additional step to determine the error type and to correct a single data error. All the DP8400s should be set to mode 7B (which is mode 7 with $\overline{OES}$ high), this mode enables the external syndromes directly to the Syndrome Generator (SG) and then the Data Error Decoder (DED) of each chip. For a single data error, the input syndrome will be unique for that error location; consequently, only one DP8400 can decode that error location and correct that bit. The other three do not indicate an error and do not change their data output latch contents. This corrected data can be output to the system data bus by means of $\overline{OB0}$ and $\overline{OB1}$. The DP8400 that decodes the data error location will indicate a single data error, while all others indicate a check bit error. If there was a single check bit error or a double bit error, then all the DP8400s will indicate a check bit error or a double bit error, respectively, through their error flags.

AN EXAMPLE OF A SINGLE DATA ERROR CORRECTION

Assuming all zero data is to be written into memory, we obtain the following set of partial check bits for all DP8400s:

C0 = 0 C4 = 0
C1 = 0 C5 = 0
C2 = 1 C6 = 0
C3 = 1 C7 = 0

Note that each DP8400 contains the basic 16-bit matrix (C0—C5). Therefore, the first six partial check bits are the same for all devices; only C6 and C7 are different. With the 64-bit configuration using the above 64-bit matrix, C6 = C7 = 0 (by connecting XP directly to V_{CC}) for the devices 0, 1, and 2; and C6 = C7 = word parity (by leaving XP pin floating) for the device 3. However, with all zero data, word parity is also zero (even parity). Therefore, the above partial check bits are obtained.

Using the formulas given in Table II, the composite check bits are as follows:

Ccomp. 0 = 0 ⊕ 0 ⊕ 0 ⊕ 0 = 0
Ccomp. 1 = 0 ⊕ 0 ⊕ 1 ⊕ 0 = 1
Ccomp. 2 = 1 ⊕ 0 ⊕ 0 ⊕ 1 = 0
Ccomp. 3 = 1 ⊕ 0 ⊕ 0 ⊕ 0 = 1
Ccomp. 4 = 0 ⊕ 1 ⊕ 1 ⊕ 1 = 1
Ccomp. 5 = 0 ⊕ 1 ⊕ 0 ⊕ 0 = 1
Ccomp. 6 = 0 ⊕ 0 ⊕ 0 ⊕ 0 = 0
Ccomp. 7 = 0 ⊕ 0 ⊕ 0 ⊕ 0 = 0

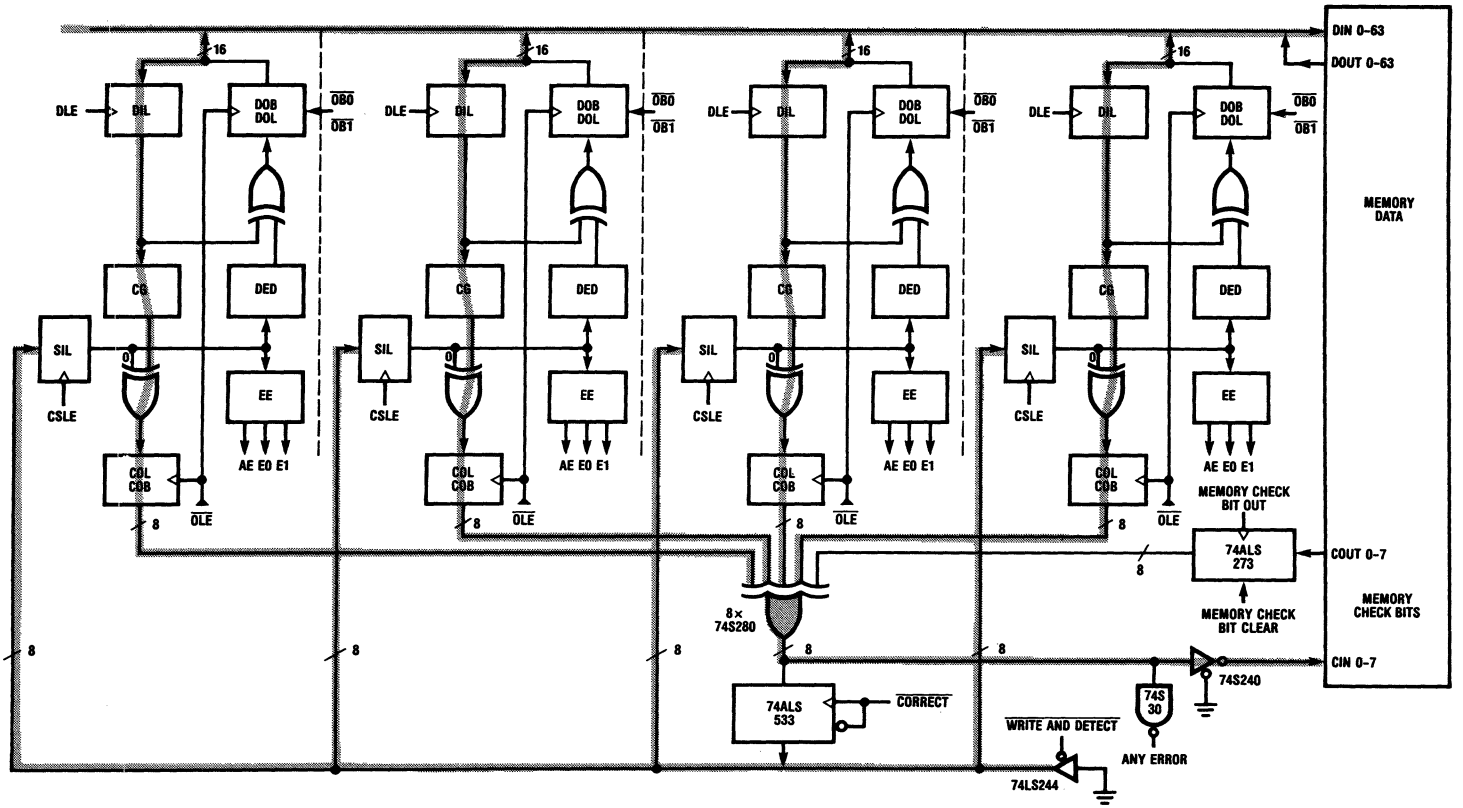
Note that these composite check bits are complemented before they are written into memory. Thus, the memory check bits read later from memory are 1100 0101.

If an error has occurred in the data position 35 which is bit 3 of device 2, then the partial check bits C(3) N produced during the detection mode are as follows:

C(3)0 = 1 C(4) = 0
C(3)1 = 1 C(5) = 0
C(3)2 = 0 C(6) = 0
C(3)3 = 1 C(7) = 0

The partial check bits of other devices are unchanged. Consequently, the newly generated composite check bits (Ccomp) and the total syndrome bits are:

Bit #	Newly Generated Composite Check Bits		Memory Check Bits		$\overline{\text{Composite Syndrome}}$
0	0	⊕	1	=	1
1	1	⊕	0	=	1
2	0	⊕	1	=	1
3	1	⊕	0	=	1
4	0	⊕	0	=	0
5	1	⊕	0	=	1
6	1	⊕	1	=	0
7	1	⊕	1	=	0



TL/F/5039–1

FIGURE 1. E²C² Simplified Block Diagram—
64-bit Parallel Expansion, Check-Bit Generation

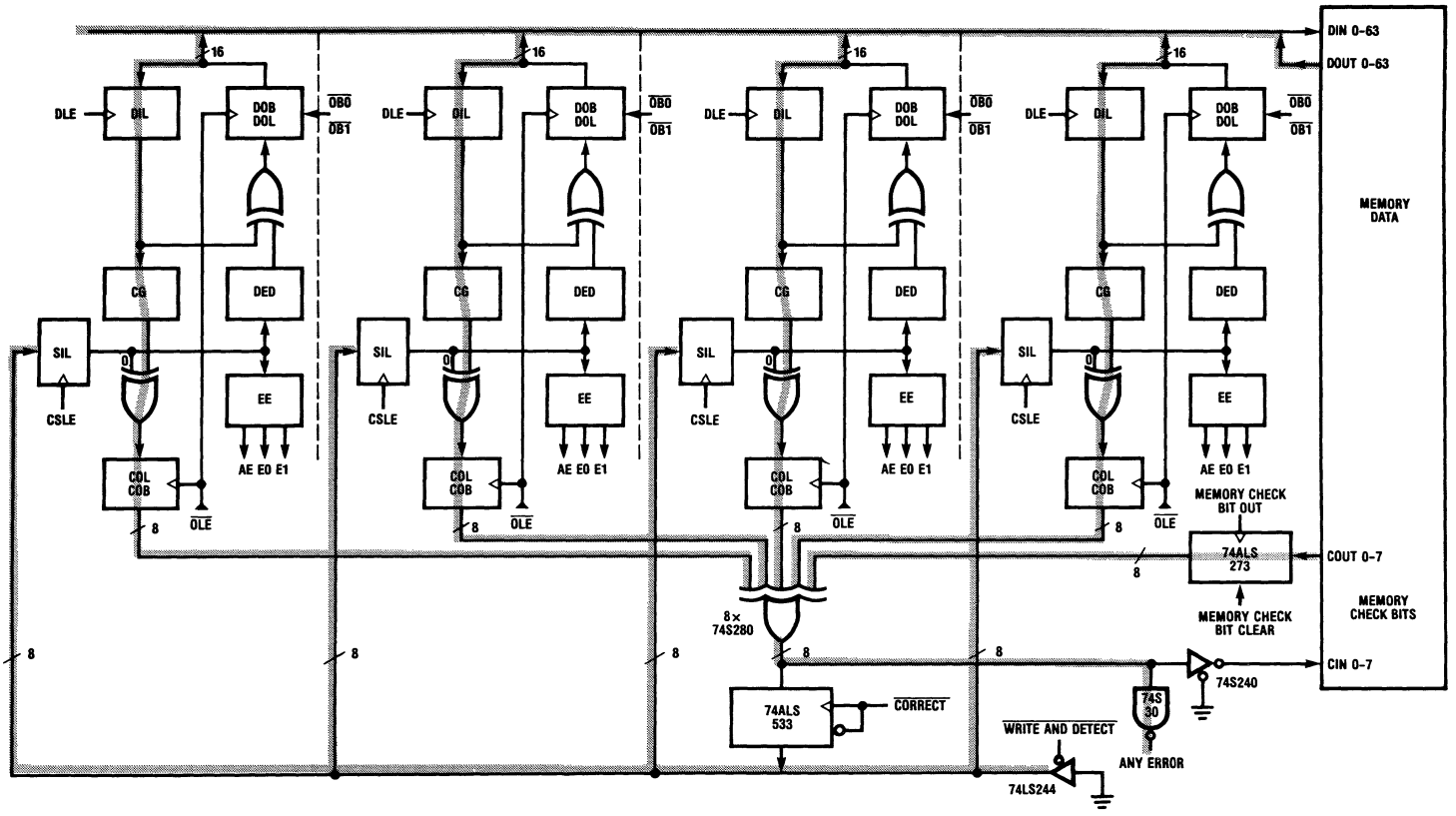


FIGURE 2. E²C² Simplified Block Diagram — 64-Bit Parallel Expansion, Error Detection

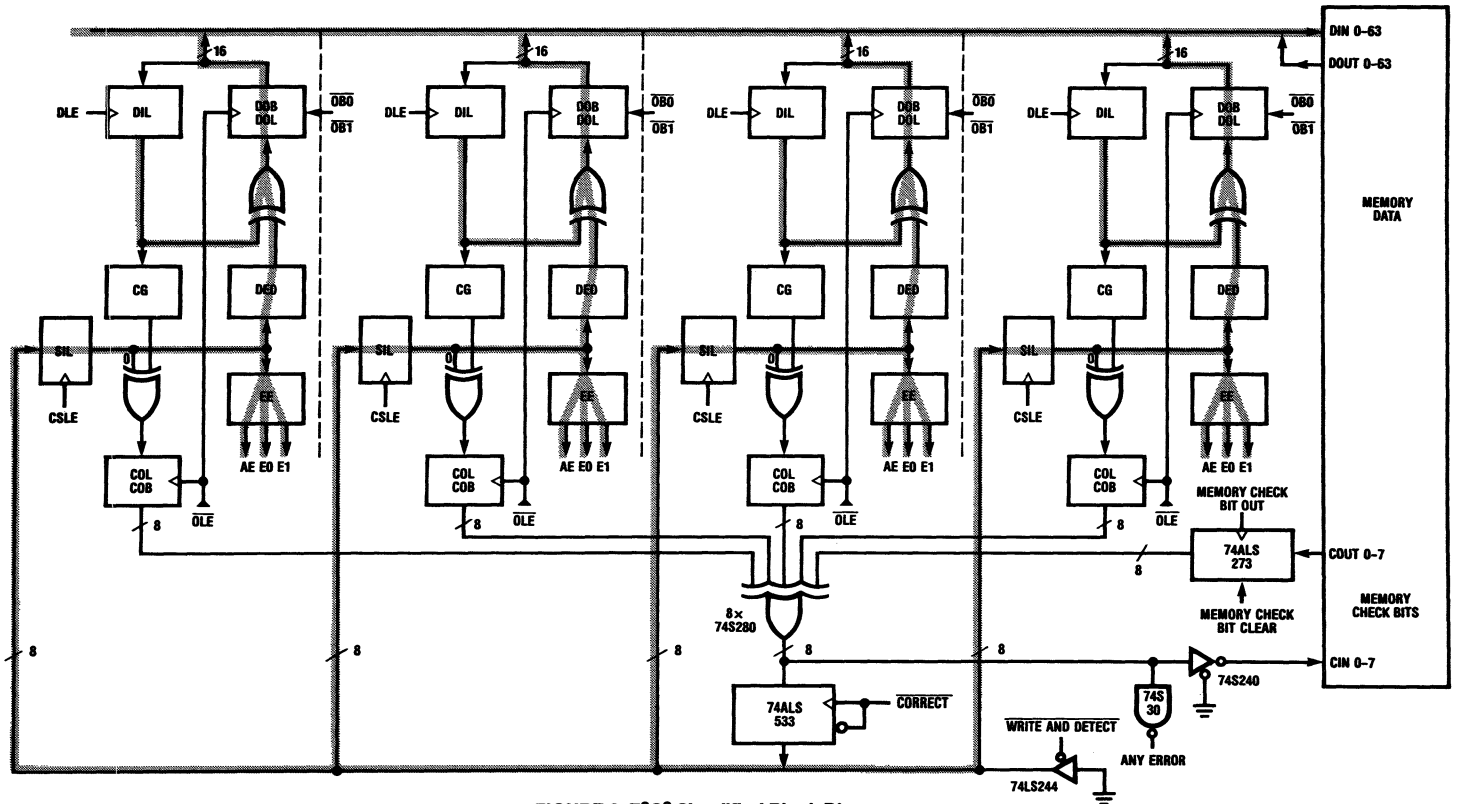
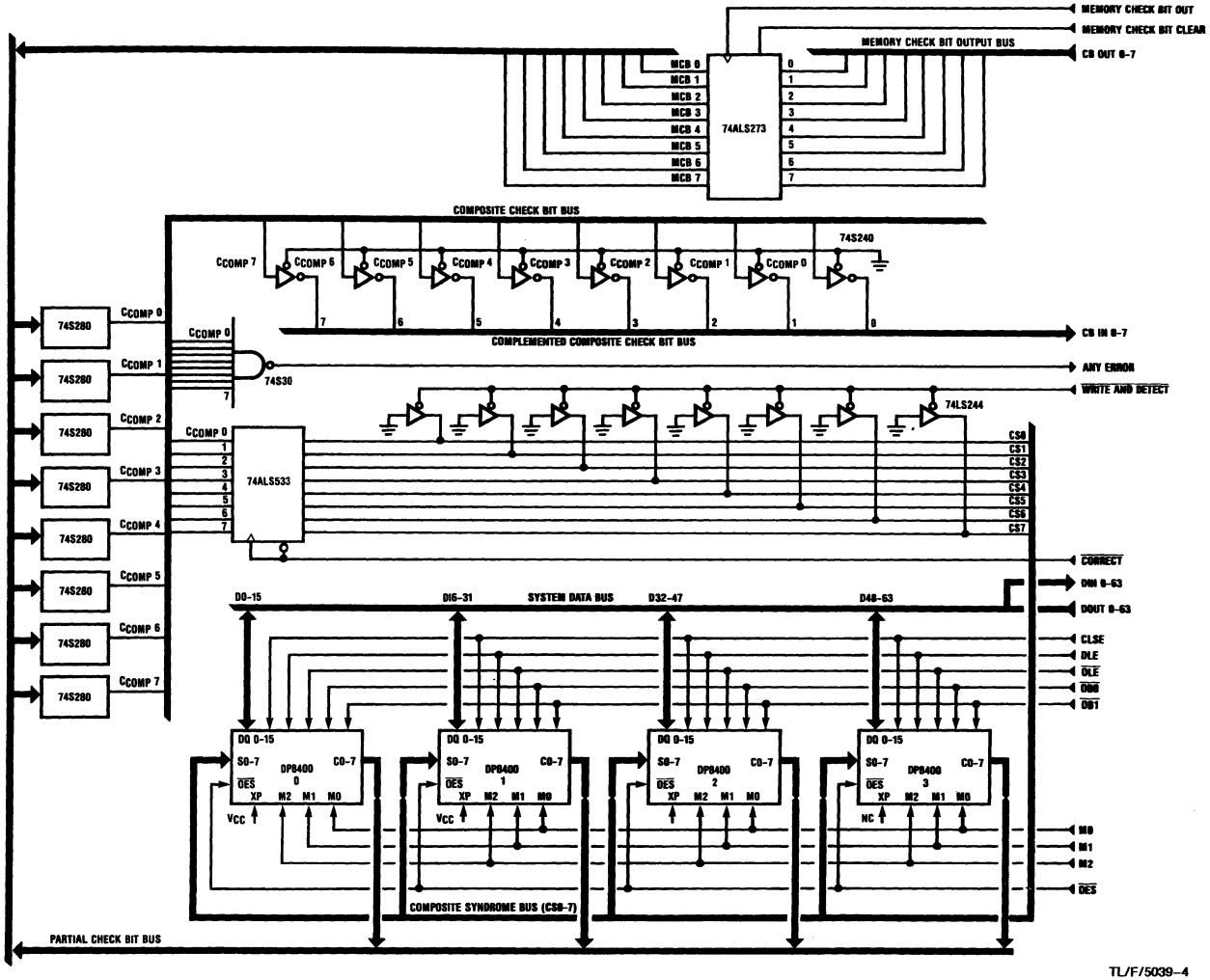


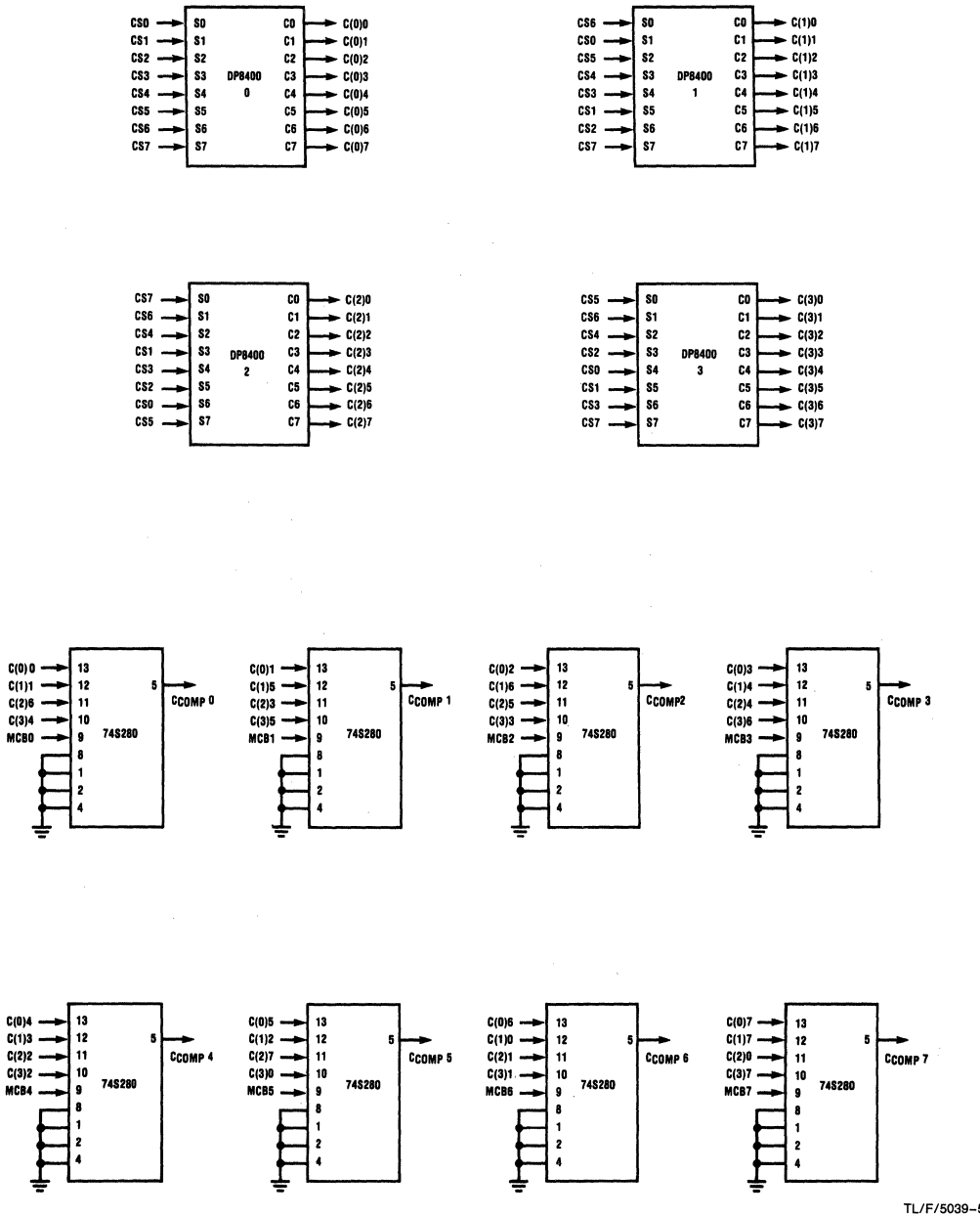
FIGURE 3. E^2C^2 Simplified Block Diagram— 64-Bit Parallel Expansion, Error Determination and Correction



TL/F/5039–4

FIGURE 4. E²C² 64-Bit Parallel Expansion, Detailed Block Diagram

Check Bit Generation, Error Detection And Error Correction (Continued)



TL/F/5039–5

FIGURE 5. E²C² 64-Bit Parallel Expansion

Check Bit Generation, Error Detection And Error Correction (Continued)

The composite syndrome 11010000 is that of the error location 35. Since the syndrome is unique and fed reordered to each DP8400, only device 2 will recognize this syndrome pattern and complement its data bit 3. Then the corrected data can be output to the system data bus when $\overline{OB0}$ and $\overline{OB1}$ of all four DP8400s go low. Devices 0, 1, and 3 all output the same data they received from memory. Only device 2 changes its (erroneous) data. Refer to *Figure 6* below for the timing diagrams of a memory write and memory read cycle (detect then correct).

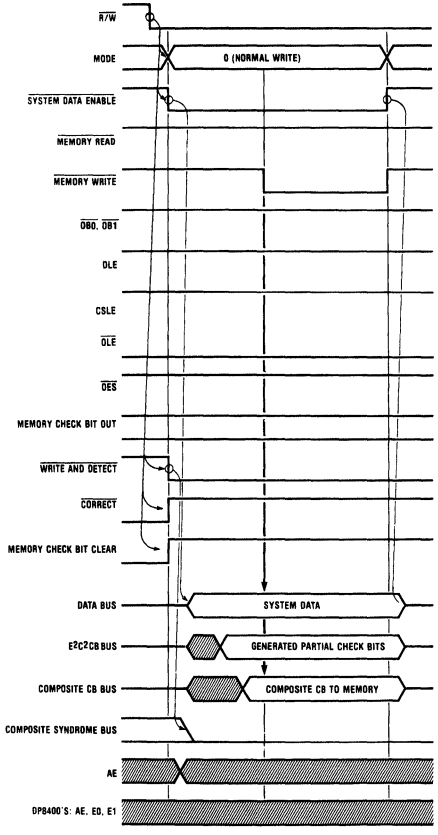
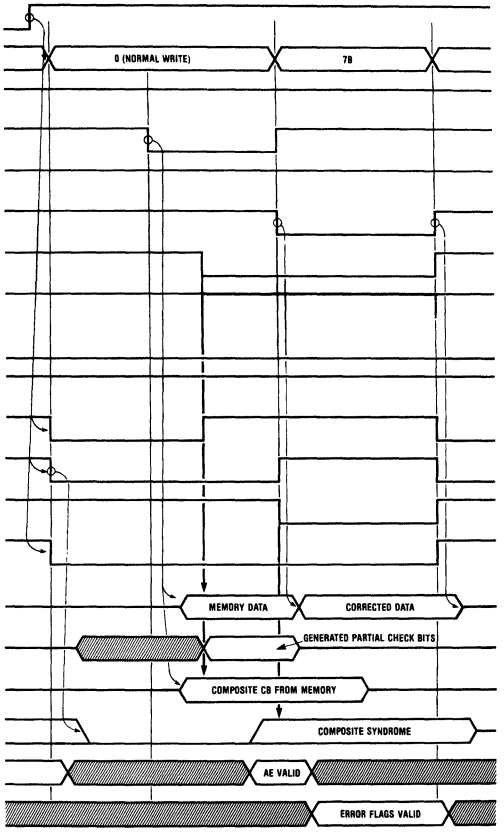


FIGURE 6A. E²C² 64-Bit Parallel Expansion Memory Write Cycle



TL/F/5039–6

FIGURE 6B. E²C² 64-Bit Parallel Expansion Memory Read Cycle (Detect Then Correct)

Section 3

Microprocessor Applications for the DP8408A/09A/17/18/19/28/29

Section 3 Contents

National Semiconductor

Selection Guide

One of the great strengths of the DP8400 DRAM Interface Family is its General purpose open-architecture approach. Applications and hardware support for all the major 8-, 16-, and 32-bit microprocessors (not just National's) are provided through the DP84XX2 Family. Each of these devices has been tailored to provide a general purpose but efficient interface between the DP8409A, 8417, 8418, 8419, 8428, 8429 DRAM controller/drivers and each of the major cpu's.

Each device uses a 20 pin standard PAL device such as the PAL16R4A as its building block. Programming equations have been written and hard programmed into each device which supply all the control signals needed to perform memory read, write, refresh, and arbitration. In order to allow for system customization, the programming equations for each device are printed in each data sheet.

Microprocessor to DRAM Controller Interface
a SELECTION GUIDE

Device #	Microprocessor Supported	DRAM Cont./ Drivers Supported	Max. Prop Delay		V_{CC}	Typ. I_{CC}	Process	Operating Temp.	Package	Page No.
			"A" PAL	"B" PAL						
DP84412	NS32008/16/32	DP8409A, 17, 18, 19, 28, 29	25 ns	15 ns	+5V ±10%	120 mA	Junction Isolated (S) or Oxide Isolated (ALS)	0°–70°C	20J, N, V	3-24
DP84512	NS32332	DP8417, 18, 19, 28, 29	25 ns	15 ns	+5V ±10%	120 mA		0°–70°C	20J, N, V	3-64
DP84322	68000/08/10 (≤10 MHz)	DP8409A, 17, 18, 19, 28, 29	25 ns	15 ns	+5V ±10%	120 mA		0°–70°C	20J, N, V	3-9
DP84422	68000/08/10 (≥10 MHz)	DP8409A, 17, 18, 19, 28, 29	25 ns	15 ns	+5V ±10%	120 mA		0°–70°C	20J, N, V	3-37
DP84522	68020	DP8417, 18, 19, 28, 29	25 ns	15 ns	+5V ±10%	120 mA		0°–70°C	20J, N, V	3-65
DP84432	8086/88/186/188	DP8409A, 17, 18, 19, 28, 29	25 ns	15 ns	+5V ±10%	120 mA		0°–70°C	20J, N, V	3-51
DP84532	80286	DP8409A, 17, 18, 19, 28, 29	25 ns	15 ns	+5V ±10%	120 mA		0°–70°C	20J, N, V	3-81

National Semiconductor

PRELIMINARY

DP84300 Programmable Refresh Timer

General Description

The DP84300 programmable refresh timer is a logic device which produces the desired refresh clock required by all dynamic memory systems.

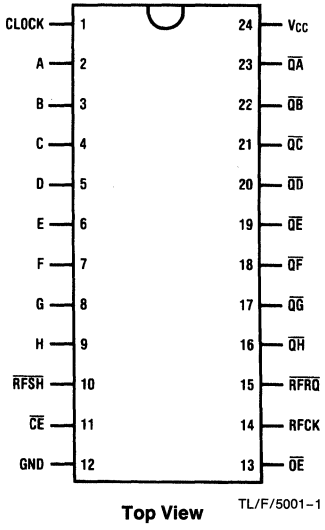
Additional circuitry has been included in the device to minimize logic required by memory systems to perform refresh control.

Features

- One chip solution to produce RFCK timing for the DP8408A, DP8409A, DP8417, DP8418, DP8419, DP8428, DP8429 dynamic RAM controllers
- Programmable refresh clock timer allows for a maximum refresh period with most system clocks
- Timing is completely synchronous with the input clock, preventing race conditions present in some memory controllers
- Includes a refresh request output, simplifying the design of refresh logic in discrete controllers

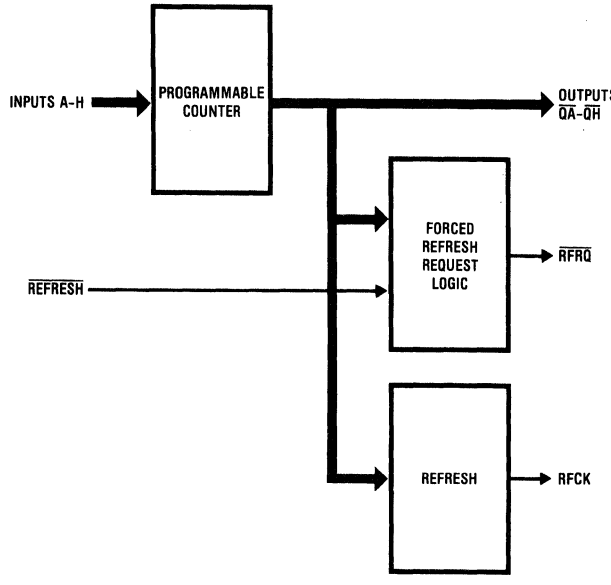
Connection & Block Diagrams

Dual-In-Line Package



Top View TL/F/5001–1

Order Number DP84300N
See NS Package Number N24C



TL/F/5001–2

FIGURE 1

Recommended Operating Conditions (Commercial)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

	Min	Typ	Max	Units
V_{CC}, Supply Voltage	4.75	5.00	5.25	V
I_{OH}, High Level Output Current			−3.2	mA
I_{OL}, Low Level Output Current			16	mA
T_A, Operating Free Air Temperature	0		75	°C

Electrical Characteristics over recommended operating temperature range

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IH}	High Level Input Voltage		2			V
V_{IL}	Low Level Input Voltage				0.8	V
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA			−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, V_{IH} = 2V, V_{IL} = 0.8V, I_{OH} = Max	2.4			V
V_{OL}	Low Level Output Voltage	V_{CC} = Min, V_{IH} = 2V, V_{IL} = 0.8V, I_{OL} = Max			0.5	V
I_{OZH}	Off-State Output Current High Level Voltage Applied	V_{CC} = Max, V_{IH} = 2V, V_O = 2.4V, V_{IL} = 0.8V			100	μA
I_{OZL}	Off-State Output Current Low Level Voltage Applied	V_{CC} = Max, V_{IH} = 2V, V_O = 0.4V, V_{IL} = 0.8V			−100	μA
I_I	Input Current at Maximum Input Voltage	V_{CC} = Max, V_I = 5.5V			1.0	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.4V			25	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V			−250	μA
I_{OS}	Short Circuit Output Current	V_{CC} = Max	−30		−130	mA
I_{CC}	Supply Current	V_{CC} = Max		150	180	mA

DP84300 Switching Characteristics over recommended ranges of temperature and V_{CC}

Symbol	Parameter		Conditions R_L = 667Ω	Commercial T_A = 0°C to +75°C V_{CC} = 5.0V ±5%			Units
				Min	Typ	Max	
t_{PD}	Clock to Output		C_L = 45 pF		35	50	ns
t_{PZX}	Pin 13 to Output Enable				20	35	ns
t_{PXZ}	Pin 13 to Output Disable		C_L = 5 pF		20	35	ns
t_{PZX}	Input to Output Enable		C_L = 45 pF		35	45	ns
t_{PXZ}	Input to Output Disable		C_L = 5 pF		35	45	ns
t_W	Width of Clock	High		25			ns
		Low		35			ns
t_{SU}	Set-Up Time			50			ns
t_H	Hold Time			0	−15		ns
f_{MAX}	Maximum Frequency			12.5			MHz

Mnemonic Description

INPUT SIGNALS

CLOCK — Provides a time base for the programmable divider.

A-H — Program inputs A through H. These inputs select the number of clock cycles that will produce one refresh period. These inputs are binary encoded, with input A the LSB, and H the MSB. Additionally, all zeros produce the maximum count of 256, and an input of one will reset the counter to one.

$\overline{\text{REFRESH}}$ — This input is used to reset the refresh request output ($\overline{\text{RFRQ}}$).

$\overline{\text{OE}}$ — Output enable. Places the outputs in TRI-STATE®.

$\overline{\text{CE}}$ — Counter enable. This input, when low, enables the timer clock and, when high, stalls the timer.

OUTPUT SIGNALS

$\overline{\text{QA}}$-$\overline{\text{QH}}$ — Refresh timer outputs $\overline{\text{QA}}$ through $\overline{\text{QH}}$. Timer starts at programmed input and counts down to one.

$\overline{\text{RFRQ}}$ — Refresh request. This output goes low on the rising edge of the refresh clock (RFCK). The first input clock edge after the $\overline{\text{REFRESH}}$ input is set low clears this output.

RFCK — Refresh clock. The period of the clock is determined by setting conditions on input pins A through H. This output is low for 20 clocks, and high for the remainder of the period.

Functional Description

The DP84300 block diagram is shown in *Figure 1*. This circuit is basically an 8-bit programmable counter. The user selects the number of input clock cycles required per refresh period and sets the binary equivalent on inputs A through H. A signal of that period is produced at the refresh clock (RFCK) output. This output stays low for 20 clock cycles, and goes high for the balance of the period.

When used with the DP8409A dynamic RAM controller, this duty cycle allows the DP8409A the maximum probability to perform a hidden refresh, while still allowing ample time for the DP8409A to perform a forced refresh when needed.

An additional output is provided to ease the design of systems that don't use the DP8409A. This output is called refresh request ($\overline{\text{RFRQ}}$). Refresh request becomes true at the rising edge of refresh clock, and becomes false on the first rising edge of the input clock after a refresh.

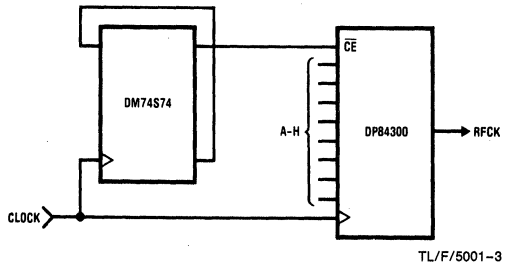
In systems where a divisor of more than 256 is needed, an expansion input ($\overline{\text{CE}}$) has been provided. When this input is high, all counter-related timing is suspended. This excluded actions due to the $\overline{\text{REFRESH}}$ input. The circuits in *Figures 2a* and *2b* show how to expand the range of the timer by 2x or by up to 4096 clock cycles. *Figures 3a* and *3b* show two typical applications using the DP84300.

By using the clock enable input, it is also possible to change the duty cycle of the refresh clock. The circuits in *Figures 4a* and *4b* show how this may be done.

To reset the counter to a known state, select an input divisor of one. On the next clock edge the counter will reset to one. On the next clock edge whatever input divisor that is present on input A-H will be loaded into the counters.

TABLE I. Divider Constants for Generation of a 15.5 μs Clock

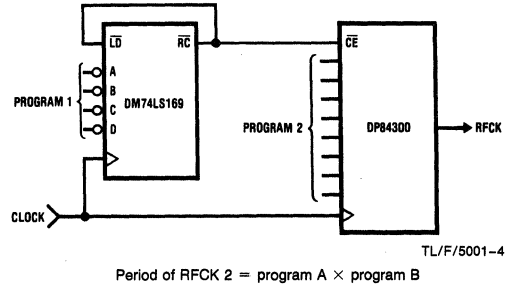
CPU Clock Frequency	Divisor Input	Actual Period of Output	% Chance of Hidden Refresh
2 MHz	31	15.5 μs	35%
3 MHz	46	15.3 μs	56%
4 MHz	62	15.5 μs	67%
5 MHz	77	15.6 μs	74%
6 MHz	93	15.5 μs	78%
7 MHz	109	15.6 μs	81%
8 MHz	124	15.5 μs	83%
9 MHz	140	15.6 μs	85%
10 MHz	155	15.5 μs	87%



TL/F/5001–3

Period of RFCK = 2x program input

FIGURE 2a. Expansion of Clock Divisor by 2x



TL/F/5001–4

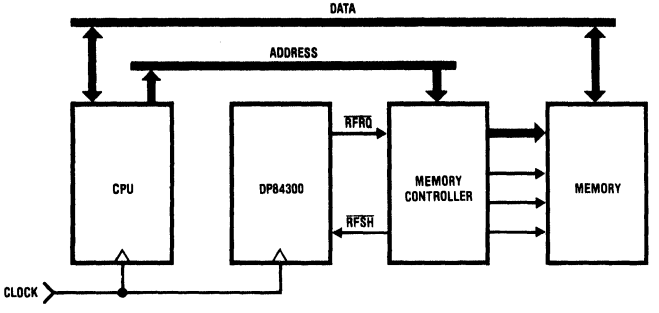
Period of RFCK 2 = program A × program B

$\overline{\text{RFCK}}$ is low for 20x program 1 clocks

Maximum period of RFCK is 4096 clocks

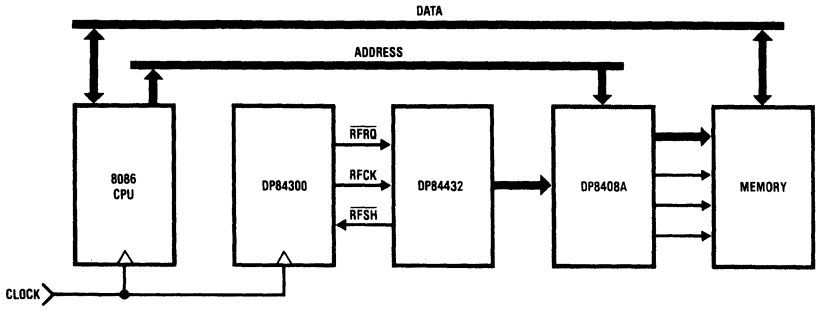
FIGURE 2b. Typical Expansion for the DP84300

Functional Description (Continued)



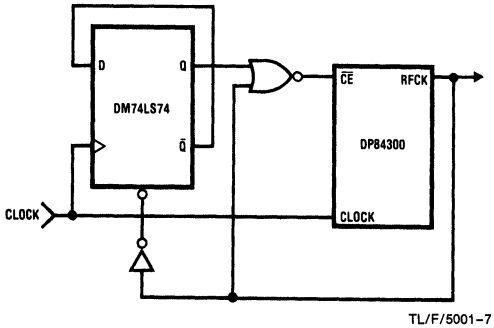
TL/F/5001–5

FIGURE 3a. Dynamic Memory System Using DP84300



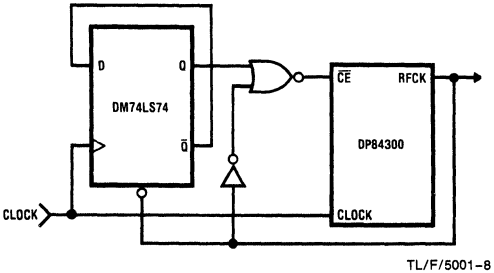
TL/F/5001–6

FIGURE 3b. 8086 System Using Dynamic RAMs DP8408A, DP84300, and DP84432



TL/F/5001–7

FIGURE 4a. Circuit for Extending RFCK Low to 40 Clocks

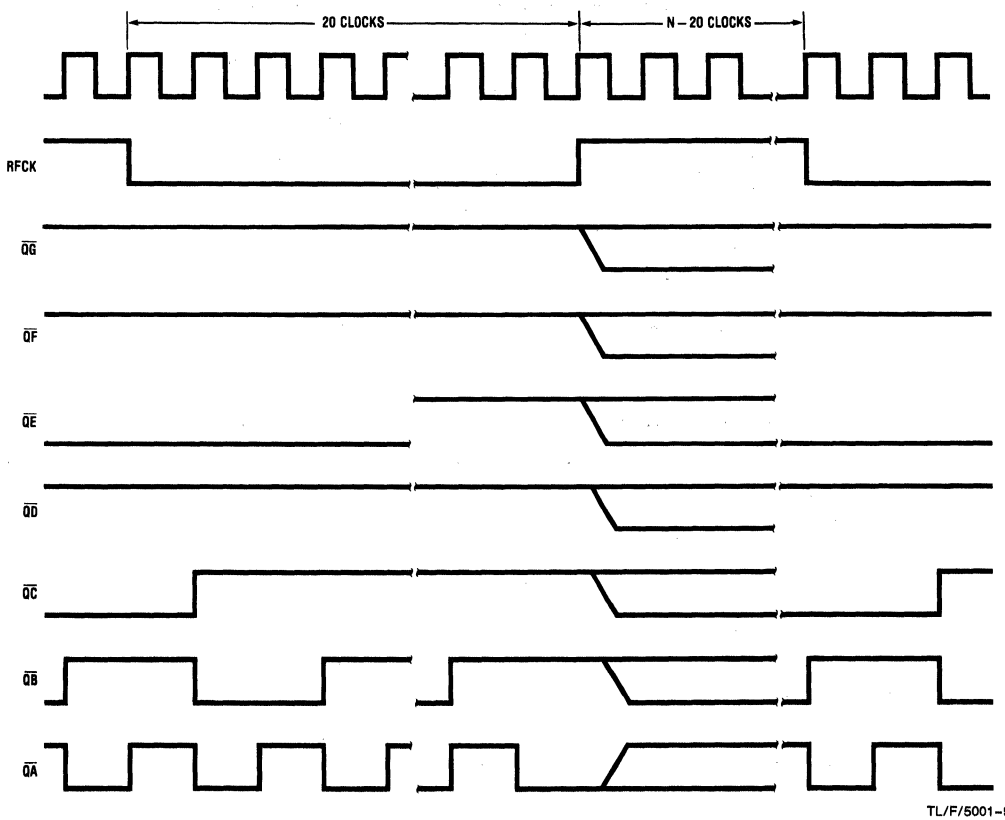


TL/F/5001–8

FIGURE 4b. Circuit for Extending RFCK High by 2x

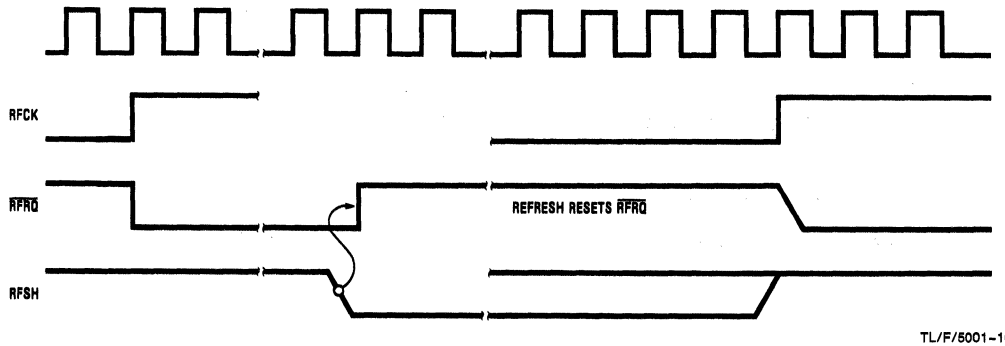
Timing Diagrams

Refresh Timer Outputs



TL/F/5001–9

$\overline{\text{REFRESH REQUEST}}$ ($\overline{\text{RFRQ}}$) Output Timing



TL/F/5001–10

National Semiconductor

DP84322 Dynamic RAM Controller Interface Circuit for the 68000 CPU

General Description

The DP84322 dynamic RAM controller interface is a Programmable Array Logic (PAL®) device which allows for easy interface between the DP8409A, 17, 18, 19, 28, 29 dynamic RAM Controllers and the 68000/008/010 microprocessors.

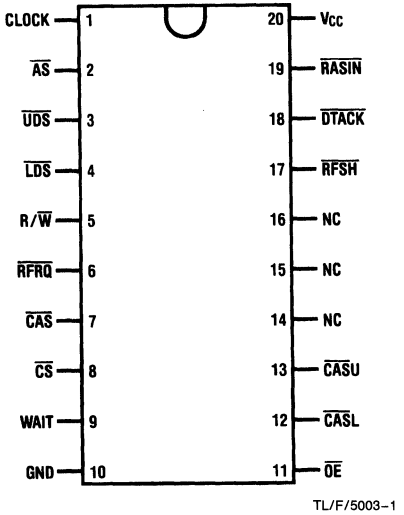
The DP84322 supplies all the control signals needed to perform memory read, write and refresh. Logic is included for inserting a wait state when using fast CPUs.

Features

- Provides 3-chip solution for the 68000 CPU and dynamic RAM interface (DP84300, DP84322, & DP8409A)
- Works with all 68000 speed versions
- Possibility of operation at 8 MHz with no wait states
- Performs hidden refresh
- $\overline{\text{DTACK}}$ is automatically inserted for both memory access and memory refresh
- Performs forced refresh using typically 4 CPU clocks
- Standard National Semiconductor PAL part (DMPAL16R4)
- PAL logic equations can be modified by the user for his specific application and programmed into any of the PAL in the National Semiconductor PAL family, including the new high speed PALs.

Connection and Block Diagrams

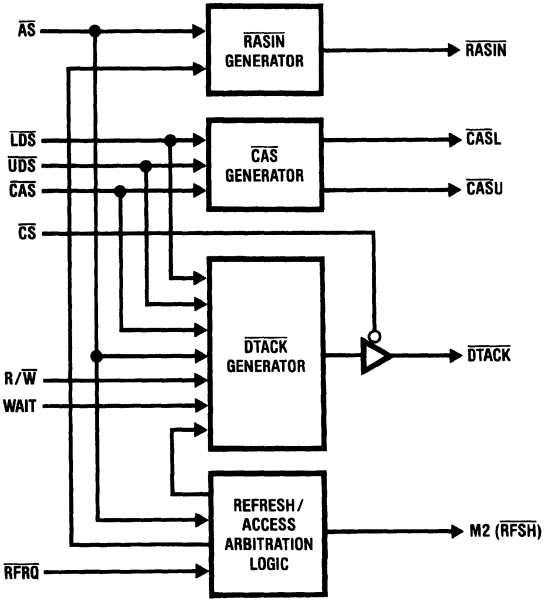
Dual-In-Line Package



TL/F/5003–1

Top View

Order Number DP84322J or DP84322N
See NS Package Number J20A or N20A



TL/F/5003–2

Recommended Operating Conditions (Commercial)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

	Min	Typ	Max	Units
V_{CC}, Supply Voltage	4.75	5.00	5.25	V
I_{OH}, High Level Output Current			−3.2	mA
I_{OL}, Low Level Output Current			24 (Note 2)	mA

	Min	Typ	Max	Units
T_A, Operating Free Air Temperature	0		75	°C

Electrical Characteristics over recommended operating temperature range

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IH}	High Level Input Voltage		2			V
V_{IL}	Low Level Input Voltage				0.8	V
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA			−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, V_{IH} = 2V, V_{IL} = 0.8V, I_{OH} = Max	2.4			V
V_{OL}	Low Level Output Voltage	V_{CC} = Min, V_{IH} = 2V, V_{IL} = 0.8V, I_{OL} = Max			0.5	V
I_{OZH}	Off-State Output Current High Level Voltage Applied	V_{CC} = Max, V_{IH} = 2V, V_O = 2.4V, V_{IL} = 0.8V			100	μA
I_{OZL}	Off-State Output Current Low Level Voltage Applied	V_{CC} = Max, V_{IH} = 2V, V_O = 0.4V, V_{IL} = 0.8V			−100	μA
I_I	Input Current at Maximum Input Voltage	V_{CC} = Max, V_I = 5.5V			1.0	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.4V			25	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V			−250	μA
I_{OS}	Short Circuit Output Current	V_{CC} = Max	−30		−130	mA
I_{CC}	Supply Current	V_{CC} = Max		150	225(1)	mA

Switching Characteristics over recommended ranges of temperature and V_{CC} (Note 3)

Symbol	Parameter		Test Conditions R_L = 667Ω	Commercial T_A = 0°C to +75°C V_{CC} = 5.0V ±5%			Units
				Min	Typ	Max	
t_{PD}	Input to Output		C_L = 50 pF		15	25	ns
t_{PD}	Clock to Output				10	15	ns
t_{PZX}	Pin 11 to Output Enable				10	20	ns
t_{PXZ}	Pin 11 to Output Disable		C_L = 5 pF		11	20	ns
t_{PZX}	Input to Output Enable		C_L = 50 pF		10	25	ns
t_{PXZ}	Input to Output Disable		C_L = 5 pF		13	25	ns
t_w	Width of Clock	High		15			ns
		Low		15			ns
t_{su}	Set-Up Time			25			ns
t_h	Hold Time			0	−10		ns

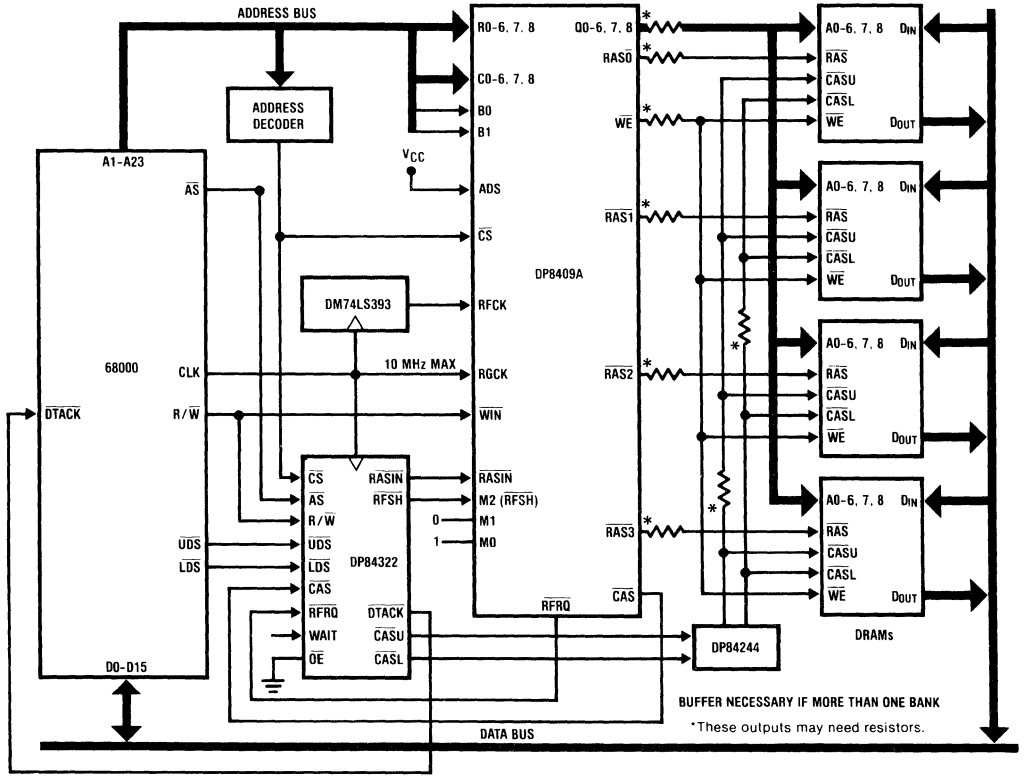
Note 1: I_{CC} = max at minimum temperature.

Note 2: One output at a time; otherwise 16 mA.

Note 3: If a PAL16R4B PAL is used, the Switching Characteristics will improve correspondingly.

System Block Diagram

DP84322 and DP8409A for 68000 CPU



TL/F/5003–3

Mnemonic Description

INPUT SIGNALS

CLOCK — The clock signal determines the timing of the outputs and should be connected directly to the 68000 clock input.

$\overline{AS}$ — Address Strobe from the 68000 CPU. This input is used to generate $\overline{RASIN}$ to the DP8409A.

$\overline{UDS}$, $\overline{LDS}$ — Upper and lower data strobe from the 68000 CPU. These inputs, together with $\overline{AS}$, R/$\overline{W}$, provide $\overline{DTACK}$ to the 68000.

R/$\overline{W}$ — Read/write from the 68000 CPU, when WAIT = 0. Selects processor speed when WAIT = 1 ("1" = 4 to 6 MHz, "0" = 8 MHz).

$\overline{CAS}$ — Column Address Strobe from the DP8409A. This input, together with $\overline{LDS}$ and $\overline{UDS}$, provides two separate $\overline{CAS}$ outputs for accessing upper and lower memory data bytes.

$\overline{CS}$ — Chip Select. This input enables $\overline{DTACK}$ output. $\overline{CS}$ = 0, $\overline{DTACK}$ output is enabled; $\overline{CS}$ = 1, $\overline{DTACK}$ output is TRI-STATE®.

$\overline{RFRQ}$ — Refresh Request. This input requests the DP84322 for a forced refresh.

WAIT — This input allows the necessary wait state to be inserted for memory access cycles.

OUTPUT SIGNALS

$\overline{RASIN}$ — This output provides a memory cycle start signal to the DP8409A and provides $\overline{RAS}$ timing during hidden refresh.

$\overline{CASU}$, $\overline{CASL}$ — These signals are the separate $\overline{CAS}$ outputs needed for byte writing.

$\overline{DTACK}$ — This output is used to insert wait states into the 68000 memory cycles when selected and during a forced refresh cycle where the CPU attempts to access the memory. This output is enabled when $\overline{CS}$ input is low and at TRI-STATE when $\overline{CS}$ is high.

$\overline{RFSH}$ — This output controls the mode of the DP8409A. It always goes low for 4 CPU clock periods when $\overline{AS}$ is inactive and a forced refresh is requested through $\overline{RFRQ}$ input. This allows the DP8409A to perform an automatic forced refresh.

Functional Description

MEMORY ACCESS

As a 68000 bus cycle begins, a valid address is output on the address bus A1–A23. This address is decoded to provide Chip Select ($\overline{CS}$) to the DP8409A. After the address becomes valid, $\overline{AS}$ goes low and it is used to set $\overline{RASIN}$ low from the DP84322 interface circuit. Note that $\overline{CS}$ must go low for a minimum of 10 ns before the assertion of $\overline{RASIN}$ for a proper memory access. As an example, with a 8 MHz 68000, the address is valid for at least 30 ns before $\overline{AS}$ goes active. $\overline{AS}$ then has to ripple through the DP84322 to produce $\overline{RASIN}$. This means the address is valid for a minimum of 40 ns before $\overline{RASIN}$ goes low, and the decoding of $\overline{CS}$ should take less than 30 ns. At this speed the DM74LS138 or DM74LS139 decoders can be selected to guarantee the 10 ns minimum required by $\overline{CS}$ set-up time going low before the access $\overline{RASIN}$ goes low (t_{CSRL} of the DP8409A). This is important because a false hidden refresh may take place when the minimum t_{CSRL} is not met. Typically $\overline{RASIN}$ occurs at the end of S2. Subsequently, selected $\overline{RAS}$ output, row to column select and then $\overline{CAS}$ will automatically follow $\overline{RASIN}$ as determined by mode 5 of the DP8409A. Mode 5 guarantees a 30 ns minimum for row address hold time (t_{RAH}) and a minimum of 8 ns column address set-up time (t_{ASC}). If the system requires instructions that use byte writing, then $\overline{CASU}$ and $\overline{CASL}$ are needed for accessing upper and lower memory data bytes, and they are provided by the DP84322. In the DP84322, $\overline{LDS}$ and $\overline{UDS}$ are gated with $\overline{CAS}$ from the DP8409A to provide $\overline{CASL}$ and $\overline{CASU}$, therefore designers need not be concerned about delaying $\overline{CAS}$ during write cycles to assure valid data being written into memory. The 8 MHz 68000 specifies during a write cycle that data output is valid for a minimum of 30 ns before $\overline{DS}$ goes active. Thus, $\overline{CASL}$ and $\overline{CASU}$ will not go low for at least 40 ns after the output data becomes stable, guaranteeing the 68000 valid data is written to memory.

Furthermore, the gating of $\overline{UDS}$, $\overline{LDS}$ and $\overline{CAS}$ allows the DP84322 interface controller to support the test and set instruction (TAS). The 68000 utilizes the read-modify-write cycle to execute this instruction. The TAS instruction provides a method of communication between processors in a multiple processor system. Because of the nature of this instruction, in the 68000, this cycle is indivisible and the Address Strobe $\overline{AS}$ is asserted throughout the entire cycle, however $\overline{DS}$ is asserted twice for two accesses: a read then a write. The dynamic RAM controller and the DP84322 respond to this read-modity-write instruction as follows (refer to the TAS instruction timing diagram for clarification). First, the selected $\overline{RAS}$ goes low as a result of $\overline{AS}$ going low, and this $\overline{RAS}$ output will remain low throughout the entire cycle. Then the DP84322's selected $\overline{CAS}$ output ($\overline{CASL}$ or $\overline{CASU}$) goes low to read the specified data byte. After this read, $\overline{DS}$ goes high causing the selected $\overline{CAS}$ to go high. A few clocks later R/$\overline{W}$ goes low and then $\overline{DS}$ is reasserted. As $\overline{DS}$ goes low, the selected $\overline{CAS}$ goes low strobing the CPU's modified data into memory, after which the cycle is ended when $\overline{AS}$ goes high.

The two $\overline{CAS}$ outputs from the DP84322, however, can only drive one memory bank. For additional driving capability, a memory driver such as the DP84244 should be added to drive loads of up to 500 pF.

Since this DP84322 interface circuit is designed to operate with all of the 68000 speed versions, a status input called WAIT is used to distinguish the 8 MHz from the others. The WAIT input should be set low for 6 MHz or less allowing full speed of operation with no wait states. Data Transfer Acknowledge input ($\overline{DTACK}$) of the 68000 at these speeds is automatically inserted during S2 for every memory transaction cycle and is then negated at the end of that cycle when $\overline{UDS}$ and/or $\overline{LDS}$ go high. For the 8 MHz 68000 however, a wait state is required for every memory transaction cycle. At these speeds, the WAIT input is set high, selecting the DP8409A's $\overline{CAS}$ output to generate $\overline{DTACK}$ and again $\overline{DTACK}$ is negated at the end of the cycle when $\overline{UDS}$ or $\overline{LDS}$ goes high. Note that $\overline{DTACK}$ output is enabled only when the DP8409A's $\overline{CS}$ is low. Therefore when the 68000 is accessing I/O or ROM (in other words, when the DP8409A is not selected), the DP84322's $\overline{DTACK}$ output goes high impedance logic '1' through the external pull-up resistor and it is now up to the designer to supply $\overline{DTACK}$ for a proper bus cycle.

The following table indicates the maximum memory speed in terms of the DRAM timing parameters: t_{CAC} (access-time from $\overline{CAS}$) and t_{RP} ($\overline{RAS}$ precharge time) required by different 68000 speed versions:

Microprocessor Clock	Maximum t_{CAC}	Minimum t_{RP}	Minimum t_{RAS}
8 MHz	125 ns	140 ns	220 ns
6 MHz	90 ns	170 ns	290 ns
4 MHz	270 ns	280 ns	450 ns

Pin 5 (R/$\overline{W}$ input to the DP84322) is not used as R/$\overline{W}$ when the WAIT input is high. Therefore, when WAIT is high and pin 5 is low, this is configured for the 8 MHz 68000. The dynamic RAM controller in this configuration operates in mode 5 and mode 1.

When both WAIT and pin 5 are high, this is configured for 4 MHz and 6 MHz 68000, allowing only two microprocessor clocks for memory refresh. Furthermore, the designer can use the DP8408A because the dynamic RAM controller now operates in mode 0 and mode 5 or mode 6. In addition, the programmable refresh timer, DP84300, should be used to determine the refresh rate (RFCK) and to provide the refresh request ($\overline{RFRQ}$) input to the DP84322. The refresh timer can provide over two hundred different divisors. $\overline{RFRQ}$ is given at the beginning of every RFCK cycle and remains active until M2 goes low for memory refresh. The DP84322 samples $\overline{RFRQ}$ when $\overline{AS}$ is high, then sets M2 low for two microprocessor clocks, taking the DP8408A or DP8409A to the external control refresh mode. $\overline{RASIN}$ for this refresh is also issued by the DP84322. If a memory access is pending, $\overline{RASIN}$ for this access will not be given until it is delayed for approximately one microprocessor clock, allowing $\overline{RAS}$ precharge time for the dynamic RAMs.

The following table indicates different memory speeds in terms of the DRAM parameters required by 4 MHz and 6 MHz 68000:

Microprocessor Clock	Maximum t_{CAC}	Minimum t_{RAS}	Minimum t_{RP}	Minimum t_{RAH}
4 MHz	290 ns	200 ns	225 ns	20 ns
6 MHz	110 ns	125 ns	140 ns	20 ns

DP8408A, DP8409A operate in mode 6 and mode 0.

Functional Description (Continued)

When WAIT = 1, pin 5 = 0 (8 MHz), the PAL controller supports read and write cycles with one inserted wait state, forced refresh with five wait states inserted if $\overline{CS}$ is valid, and hidden refresh. This PAL mode does not support the TAS instruction.

When WAIT = pin 5 = 1 (4–6 MHz), the PAL controller supports read and write cycles with no wait states inserted, and forced refresh with two wait states inserted if $\overline{CS}$ is valid. This PAL mode does not support the TAS instruction and only supports hidden refresh when used in mode 5 with the DP8409A controller.

The DP84322 can possibly be operated at 8 MHz with no wait states (WAIT = "0") given the following conditions:

FAST PAL (PAL16R4A)

S2 + S3 + S4 + S5 = 250 ns

$\overline{RASIN}$ delay = 60 ns ($\overline{AS}$ low max.)

+ 25 ns (Fast PAL delay) = 85 ns max.

$\overline{RASIN}$ to $\overline{CAS}$ delay DP8409-2 = 130 ns max.

External $\overline{CASH,L}$ generation using 74S02 and 74S240

7.5 ns (74S02) + 10 ns (74S240) − 7.5 ns (less load on 8409 $\overline{CAS}$ line) = 10 ns max.

Transceiver delay (74LS245) = 12 ns max.

68000 data setup into S6 = 40 ns min.

$\therefore$ Minimum t_{CAC} = 53 ns

= 250 − 85 − 130 − 10 − 12 + 40

Minimum t_{RAS} = 240 ns

Minimum t_{RP} = 150 ns

Minimum t_{RAH} = 20 ns

REFRESH CYCLE

Since the access sequence timing is automatically derived from $\overline{RASIN}$ in mode 5, R/$\overline{C}$ and $\overline{CASIN}$ are not used and now become Refresh Clock (RFCK) and $\overline{RAS}$-generator clock (RGCK) respectively. The Refresh Clock RFCK may be divided down from RGCK, which is the microprocessor clock, using the DM74LS393 or DM74LS390. RFCK provides the refresh time interval and RGCK the fast clock for all-$\overline{RAS}$ refresh if forced refreshing is necessary. The DP8409A offers both hidden refresh in mode 5 and forced refresh in mode 1 with priority placed on hidden refreshing. Assume 128 rows are to be refreshed, then a 16 μs maximum clock period is needed for RFCK to distribute refreshing of all the rows over the 2 ms period.

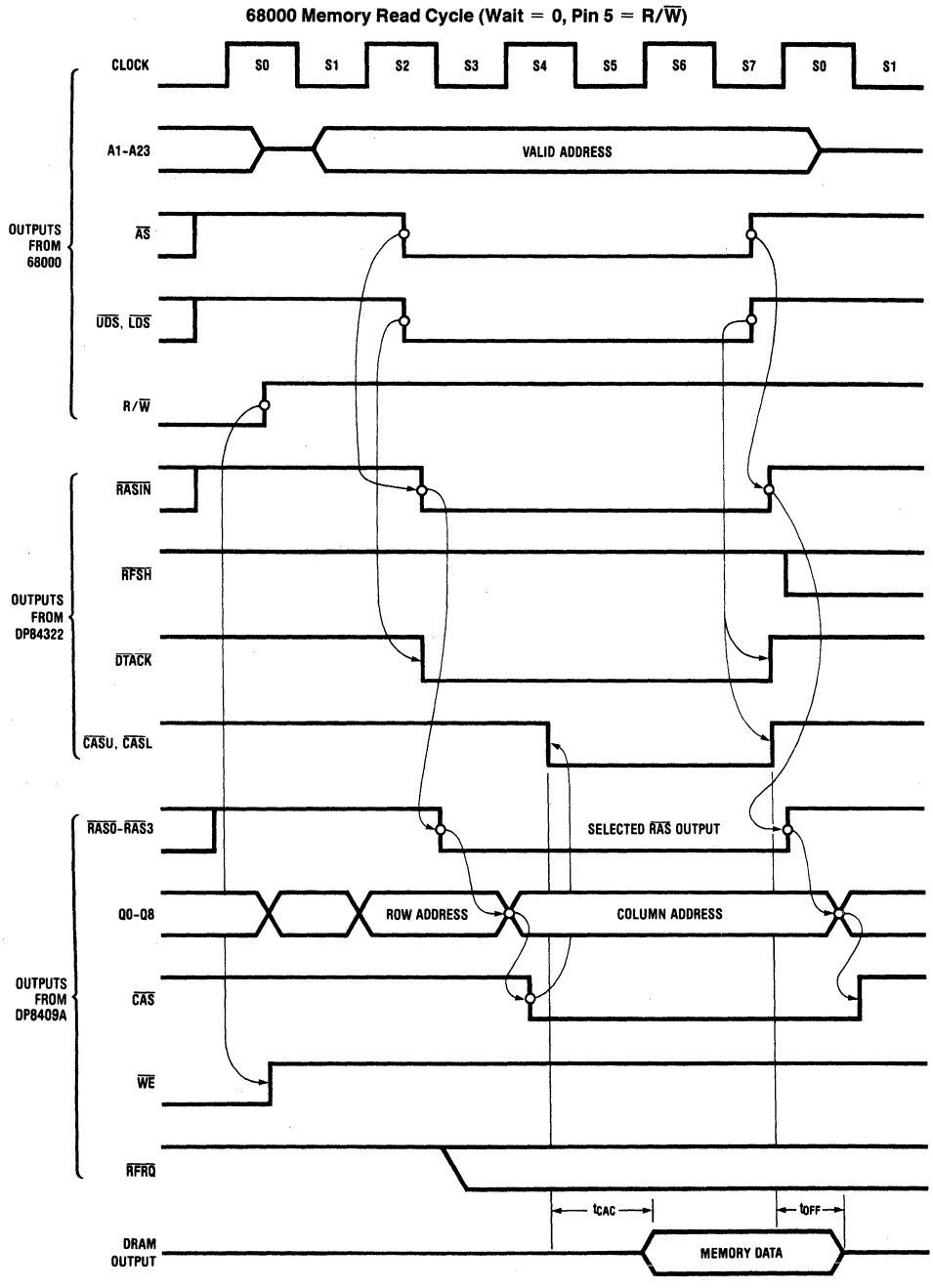
The DP8409A provides hidden refreshing in mode 5 when the refresh clock (RFCK) is high and the microprocessor is not accessing RAM. In other words, when the DP8409A's chip select is inactive because the microprocessor is accessing elsewhere, all four $\overline{RAS}$ outputs follow $\overline{RASIN}$, strobing the contents of the on-chip refresh counter to every memory bank. $\overline{RASIN}$ going high terminates the hidden refresh and also increments the refresh counter, preparing it for the next refresh cycle. Once a hidden refresh has taken place, a forced refresh will not be requested by the DP8409A for the current RFCK cycle.

However, if the microprocessor continuously accessed the DP8409A and memory while RFCK was high, a hidden refresh could not have taken place and now the system must force a refresh. Immediately after RFCK goes low, the Refresh Request signal ($\overline{RFRQ}$) from the DP8409A goes low, indicating a forced refresh is necessary. First, when $\overline{RFRQ}$ goes low any time during S2 to S7, the controller interface circuit waits until the end of the current memory access cycle and then sets M2 ($\overline{RFSH}$) low. This refresh takes four microprocessor clocks to complete. If the current cycle is another memory cycle, the 68000 will automatically be put in four wait states. Alternately, when $\overline{RFRQ}$ goes low while $\overline{AS}$ is high during S0 to S1, M2 is now set low at S2. Therefore, it requires an additional microprocessor clock for this refresh. Once the DP8409A is in mode 1 forced refresh, all the $\overline{RAS}$ outputs remain high until two RGCK trailing edges after M2 goes low, when all $\overline{RAS}$ outputs go low. This allows a minimum of one and a half clock periods of RGCK for $\overline{RAS}$ precharge time. As specified in the DP8409A data sheet, the $\overline{RAS}$ outputs remain low for two clock periods of RGCK. The refresh counter is incremented as the $\overline{RAS}$ outputs go high. Once the forced refresh has ended, M2 is brought high, the DP8409A back to mode 5 auto access. Note that $\overline{RASIN}$ for the pending access is not given until it has been delayed for a full microprocessor clock, allowing $\overline{RAS}$ precharge time for the coming access.

If the 68000 bus is inactive (i.e., the 68000's instruction queue is full, or the 68000 is executing internal operations such as a multiply instruction, or the 68000 is in halt state . . .) and a refresh has been requested, a refresh will also take place because $\overline{RFRQ}$ is continuously sampled while $\overline{AS}$ is high. Therefore, refreshing under these conditions will be transparent to the microprocessor. Consequently, the system throughput is increased because the DP84322 allows refreshing while the 68000 bus is inactive.

The 84322 is a standard National Semiconductor PAL part (DMPAL16R4). The user can modify the PAL equations to support his particular application. The 84322 logic equations function table (functional test), and logic diagram can be seen at the end of this data sheet.

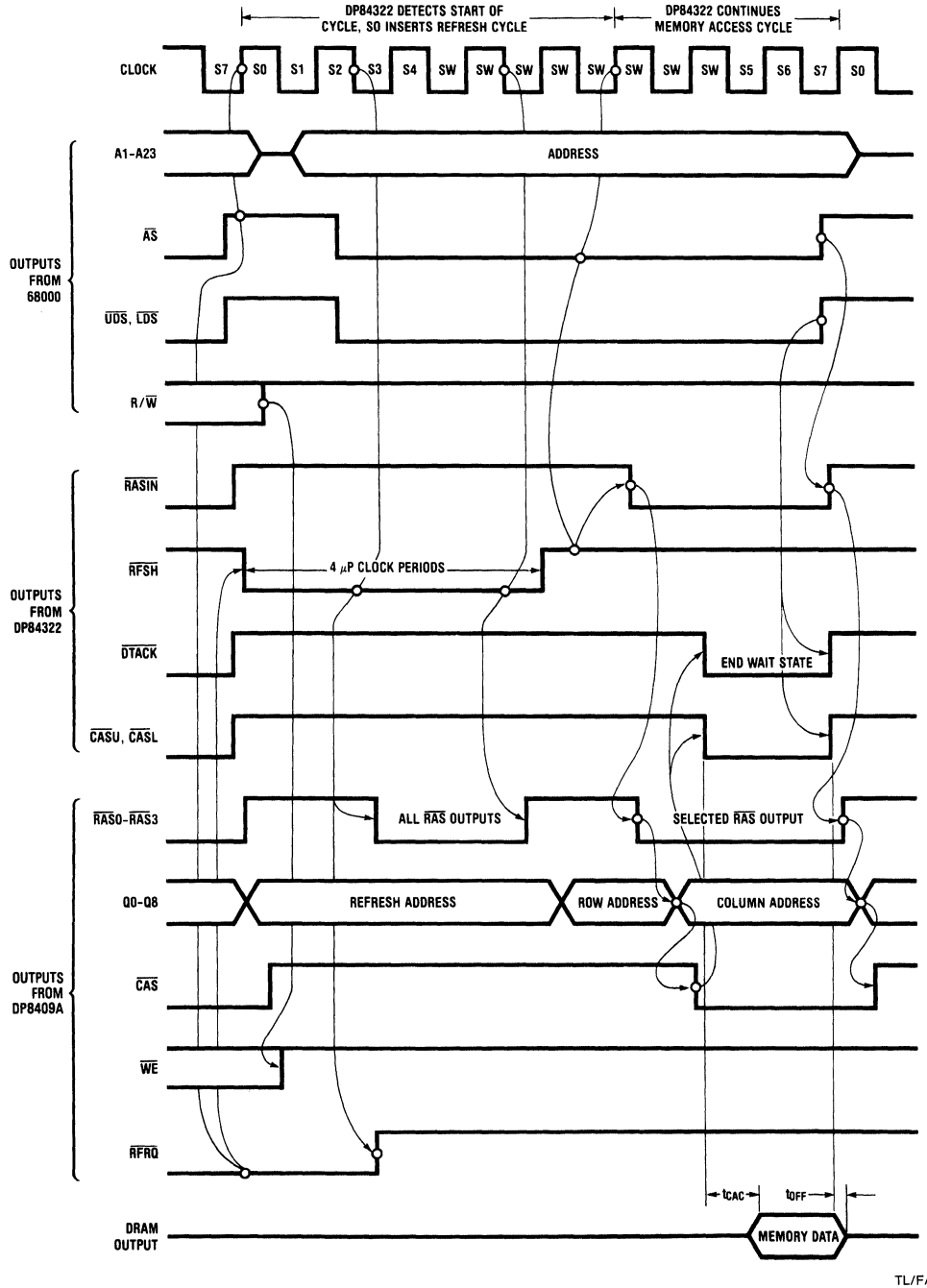
System Timing Diagrams



TL/F/5003–4

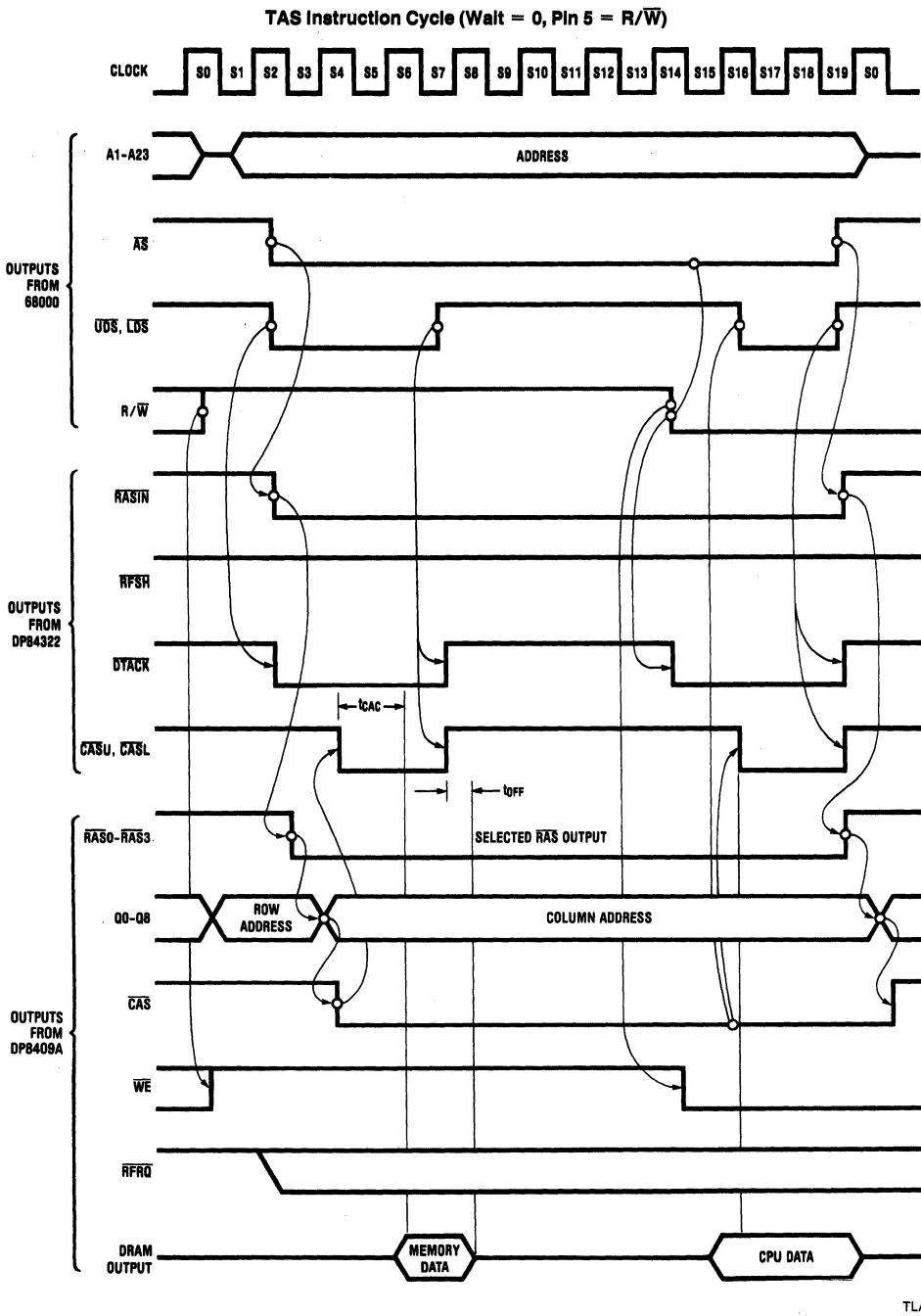
System Timing Diagrams (Continued)

68000 Memory Read Cycle and Forced Refresh (Wait = 0, Pin 5 = R/$\overline{W}$)
(4 Wait Clock Periods Inserted for Forced Refresh)



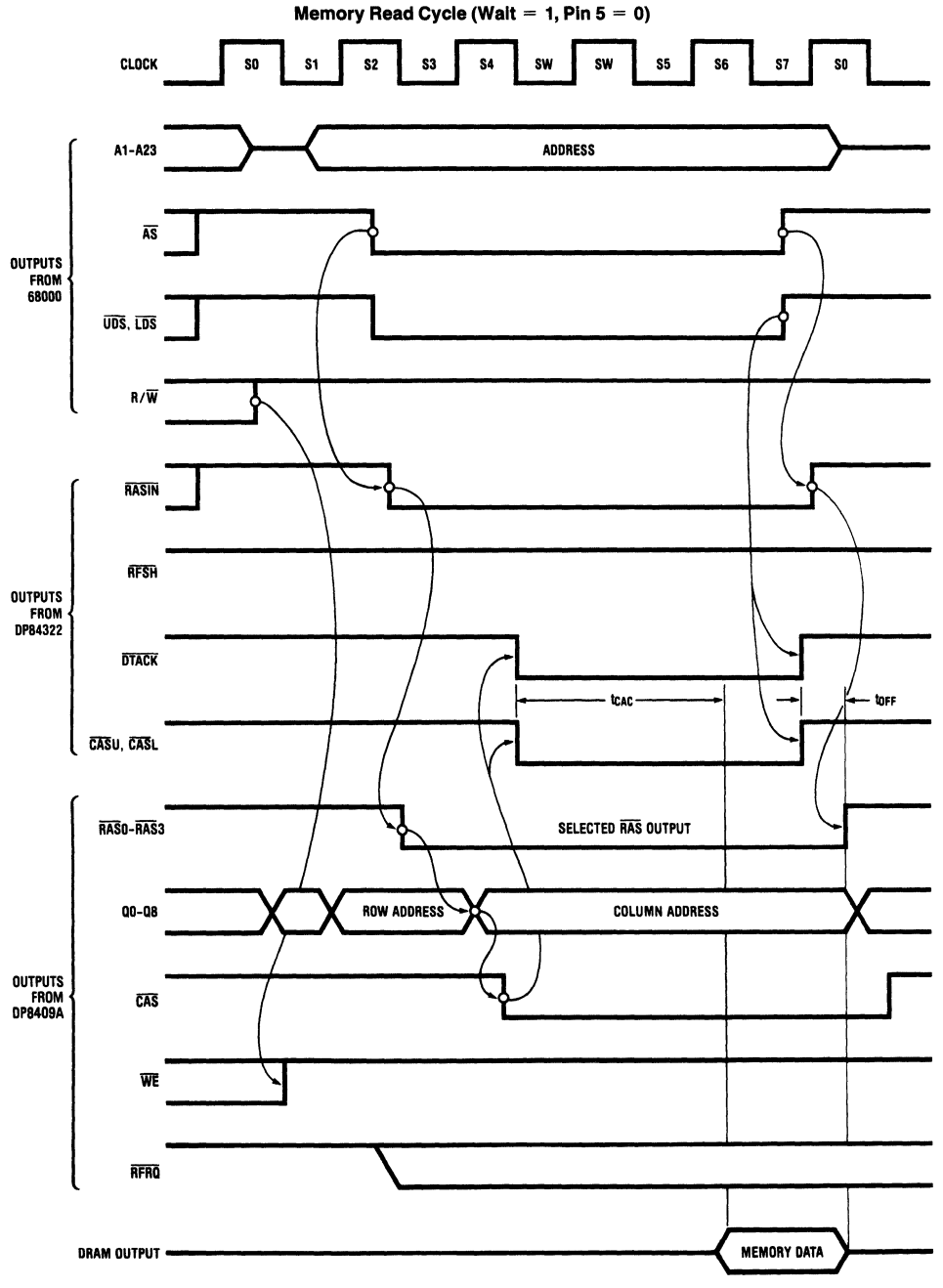
TL/F/5003–5

System Timing Diagrams (Continued)



TL/F/5003–6

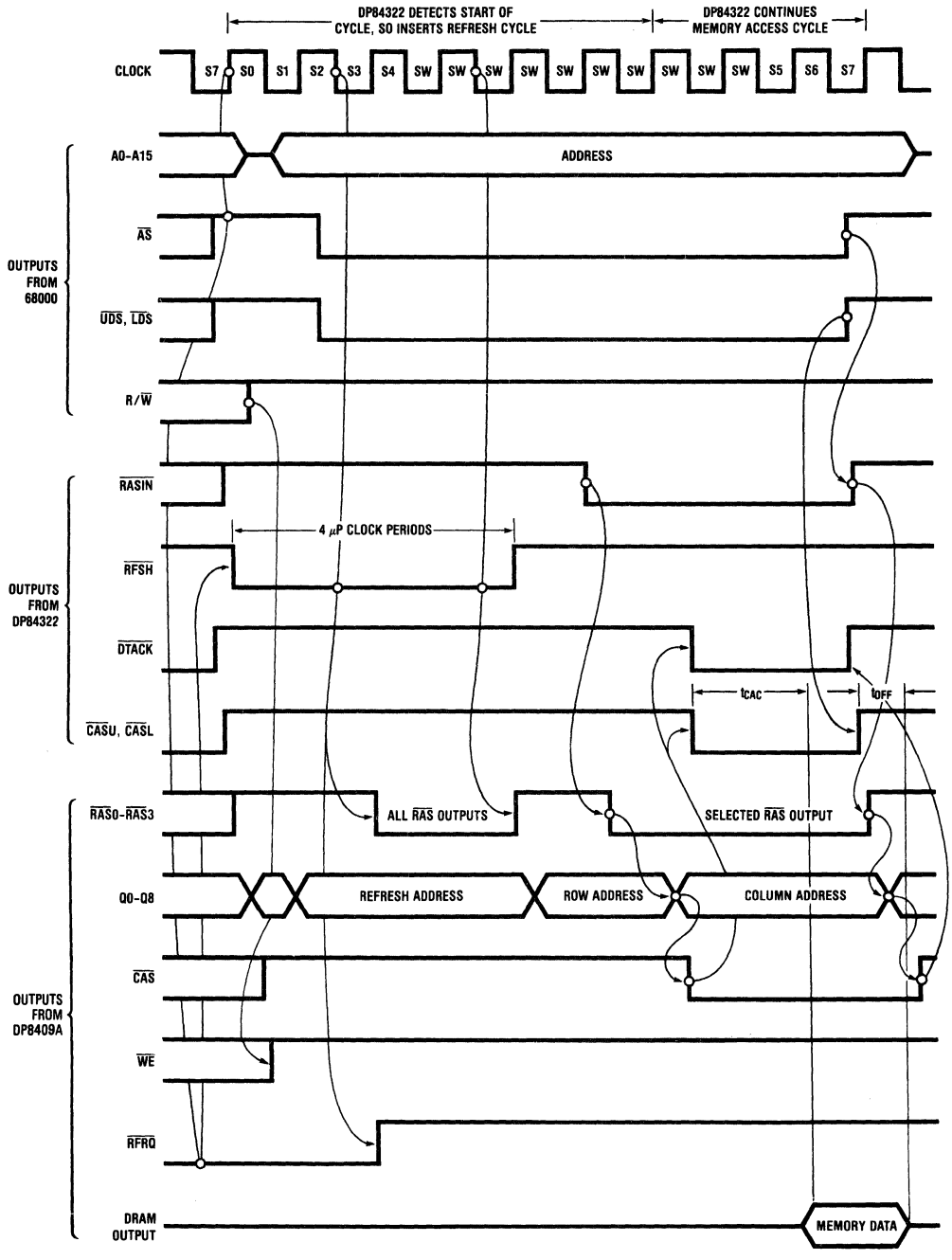
System Timing Diagrams (Continued)



TL/F/5003–7

System Timing Diagrams (Continued)

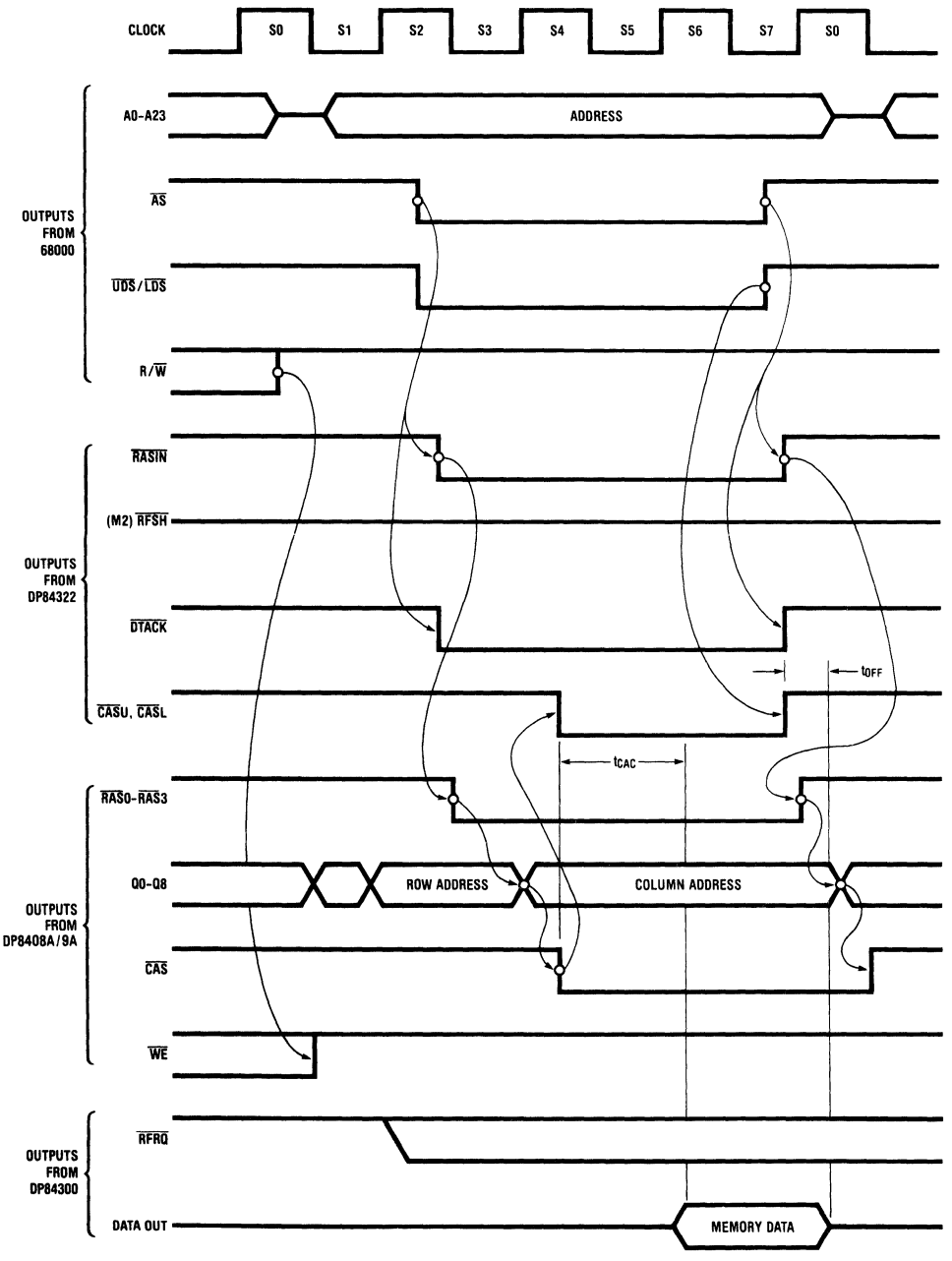
Memory Read Cycle and Forced Refresh (Wait = 1, Pin 5 = 0)



TL/F/5003–8

System Timing Diagrams (Continued)

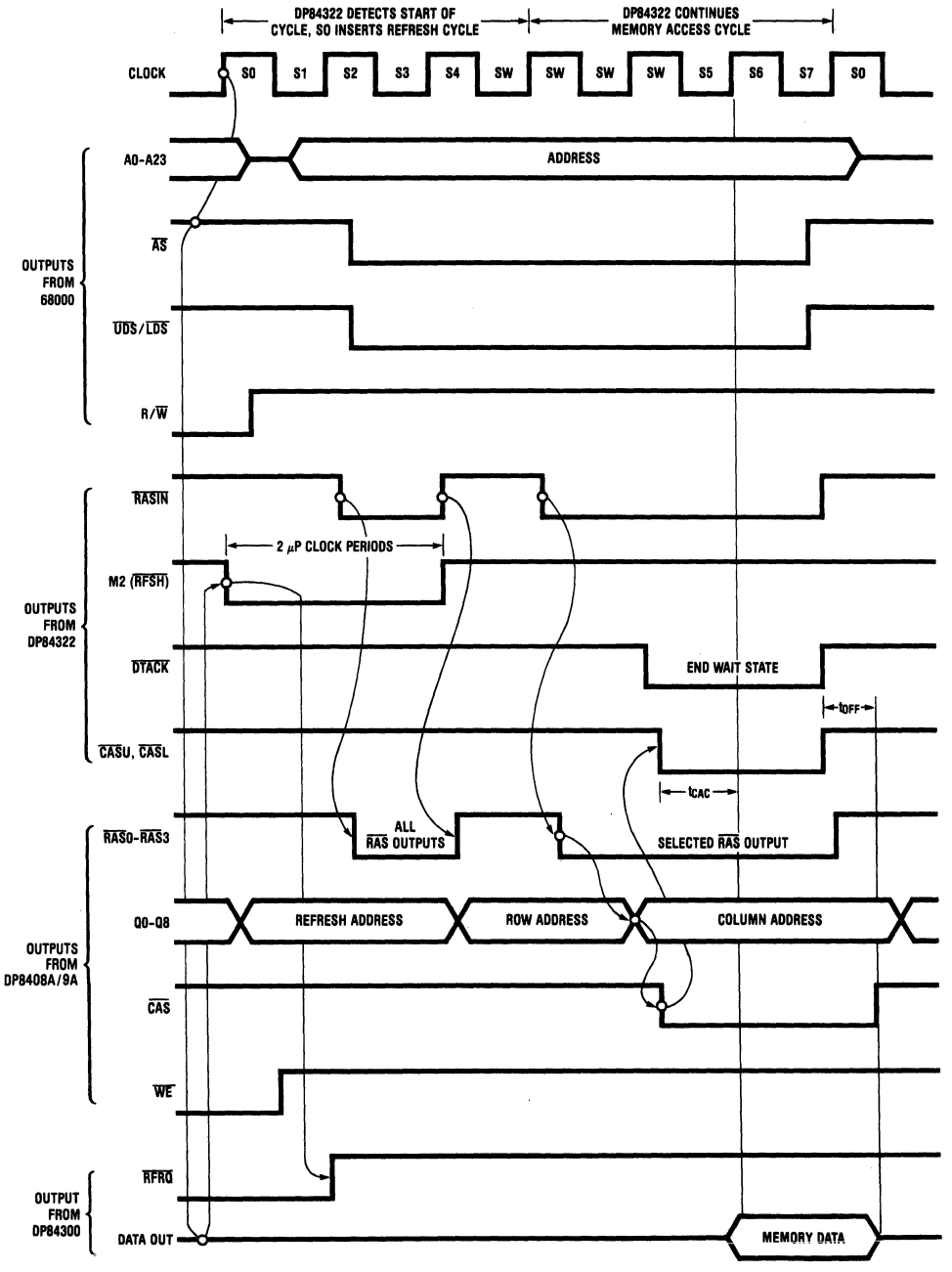
68000 Memory Read Cycle (Wait and Pin 5 = 1)



TL/F/5003–10

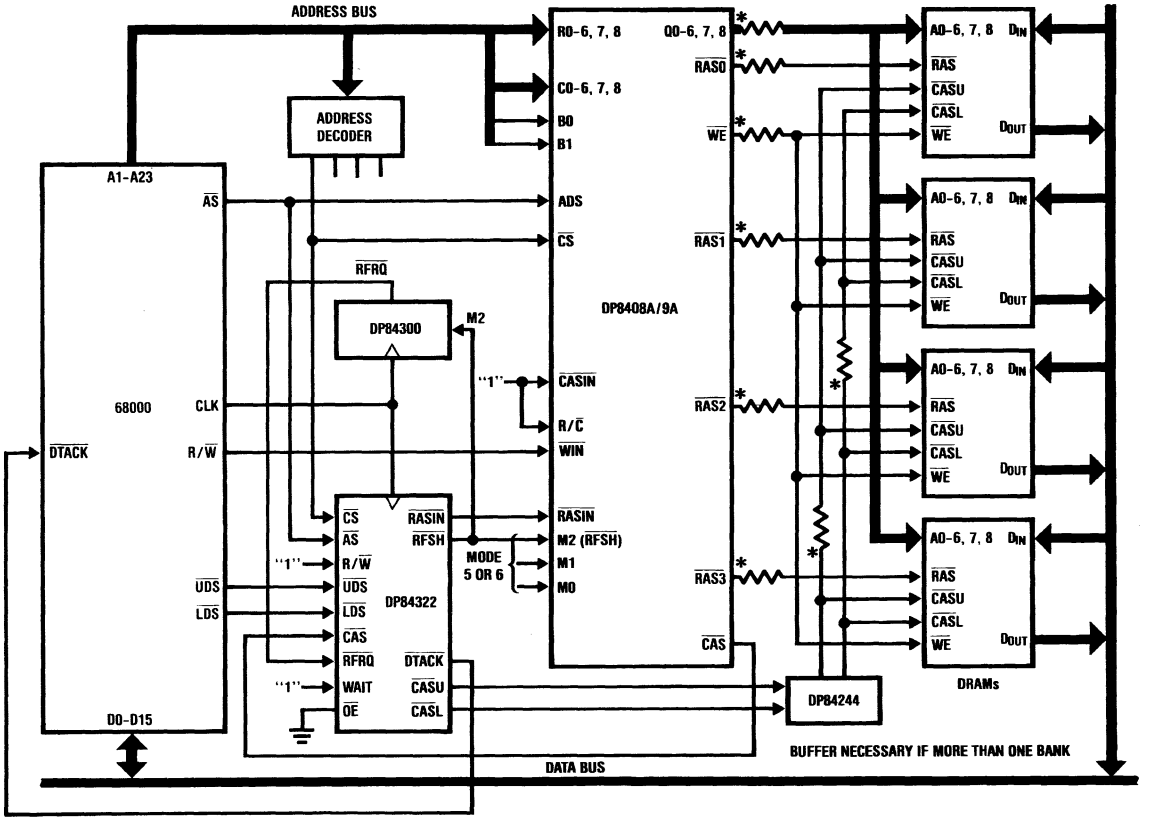
System Timing Diagrams (Continued)

68000 Memory Read Cycle and Memory Refresh (Wait and Pin 5 = 1)



TL/F/5003–11

DP8408A, DP8409A and 68000 Interface



TL/F/5003–9

*These outputs may need resistors.

PAL Boolean Equations

PAL16R4 DP84322 Dynamic RAM Controller Interface for the MC68000-DP8409A Memory System

CK /AS /UDS /LDS R /RFRQ /CAS /CS WAIT GND /OE /CL /CU /C /B /A /RFSH /DTACK /RASIN VCC

IF (VCC) RASIN = AS • /RFSH • /A + RFSH • R • A • WAIT

IF (CS) DTACK = /R • CAS • WAIT + UDS • /A • /B • /WAIT + LDS • /A • /B • /WAIT + AS • /R • /A • /B • /WAIT + AS • /RFSH • R • /A • /B • WAIT

RFSH: = /AS • RFRQ + RFSH • /R • /C • WAIT + RFSH • R • /A • WAIT + RFSH • /C • /WAIT

A: = RFSH

B: = A

C: = B

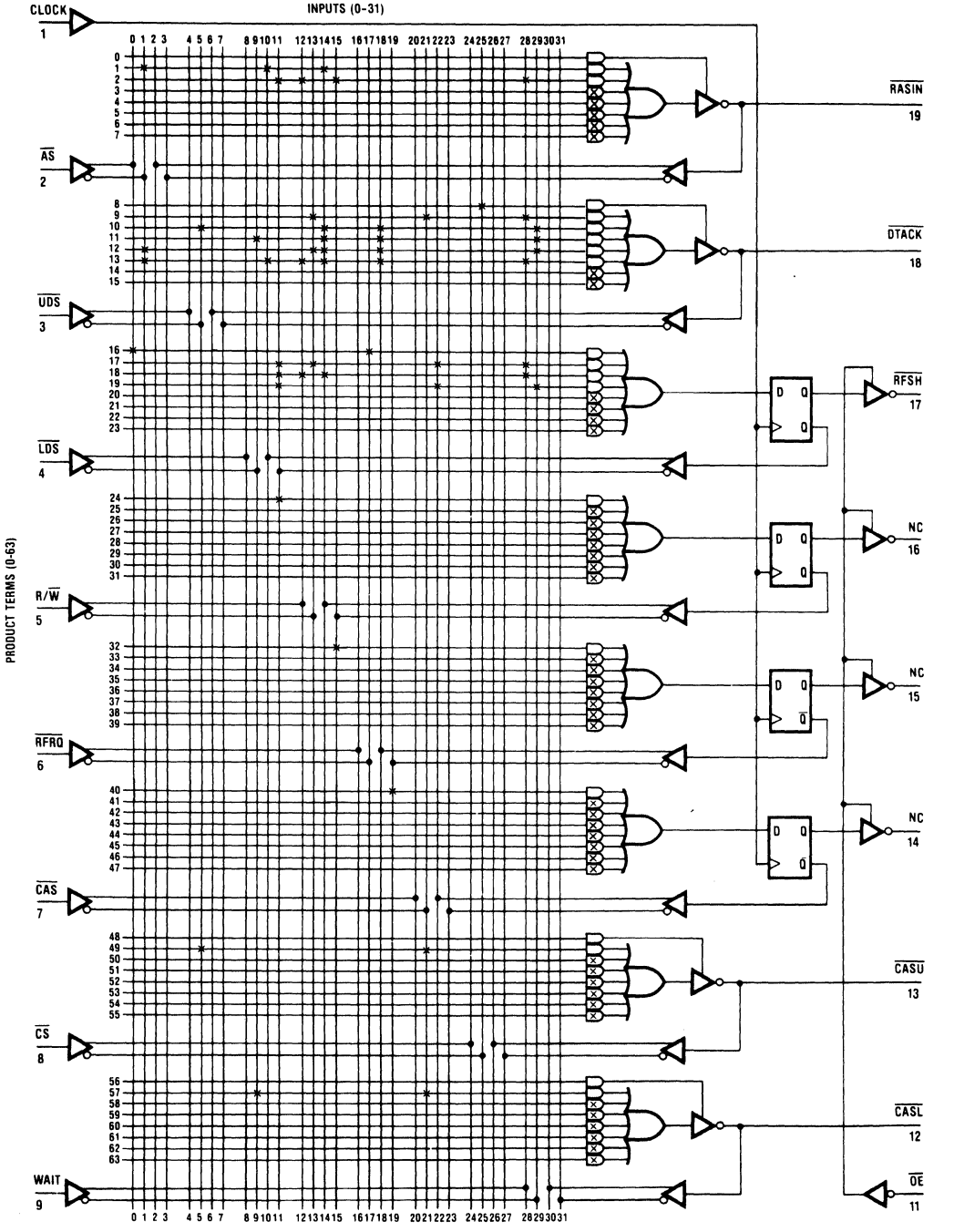
IF (VCC) CU = UDS • CDS

IF (VCC) CL = LDS • CAS

Function Table

CK	$\overline{AS}$	$\overline{UDS}$	$\overline{LDS}$	R	$\overline{RFRQ}$	$\overline{CAS}$	$\overline{CS}$	WAIT	$\overline{OE}$	$\overline{CL}$	$\overline{CU}$	$\overline{C}$	$\overline{B}$	$\overline{A}$	$\overline{RFSH}$	$\overline{DTACK}$	$\overline{RASIN}$
C	H	L	L	H	H	H	H	L	L	H	H	X	X	X	X	X	H
C	H	L	L	H	H	L	H	L	L	L	L	X	X	X	X	X	H
C	H	L	H	H	H	L	H	L	L	H	L	X	X	X	X	X	H
C	H	H	L	H	H	L	H	L	L	L	H	X	X	X	X	X	H
C	H	H	H	H	H	H	H	L	L	H	H	H	H	H	H	Z	H
C	L	L	H	H	H	H	L	L	L	H	H	H	H	H	H	L	L
C	L	L	H	H	H	L	L	L	L	H	L	H	H	H	H	L	L
C	L	H	H	H	H	L	L	L	L	H	H	H	H	H	H	H	L
C	L	H	H	L	H	L	L	L	L	H	H	H	H	H	H	L	L
C	L	L	H	L	H	L	L	L	L	H	L	H	H	H	H	L	L
C	H	H	H	L	H	H	L	L	L	H	H	H	H	H	H	H	H
C	H	H	H	L	L	H	L	L	L	H	H	H	H	H	L	H	H
C	H	H	H	L	L	H	L	L	L	H	H	H	H	L	L	H	H
C	L	H	L	L	H	H	L	L	L	H	H	L	L	L	L	H	H
C	L	H	L	L	H	H	L	L	L	H	H	L	L	L	H	H	H
C	L	H	L	L	H	H	L	L	L	H	H	L	L	H	H	H	L
C	L	H	L	L	H	L	L	L	L	L	H	L	H	H	H	L	L
C	L	H	L	L	H	L	L	L	L	L	H	H	H	H	H	L	L
C	H	H	H	L	H	L	L	L	L	H	H	H	H	H	H	H	H
C	H	H	H	L	L	H	L	H	L	H	H	H	H	H	L	H	H
C	H	H	H	L	L	H	L	H	L	H	H	H	H	L	L	H	H
C	L	L	L	L	H	H	L	H	L	H	H	H	L	L	L	H	H
C	L	L	L	L	H	H	L	H	L	H	H	L	L	L	L	H	H
C	L	L	L	L	H	H	L	H	L	H	H	L	L	L	H	H	H
C	L	L	L	L	H	H	L	H	L	H	H	L	L	H	H	H	L
C	L	L	L	L	H	L	L	H	L	L	L	L	H	H	H	L	L
C	H	H	H	L	H	L	L	H	L	H	H	H	H	H	H	L	H
C	H	H	H	L	H	H	H	H	L	H	H	H	H	H	H	Z	H
C	H	H	H	H	L	H	L	H	L	H	H	H	H	H	L	H	H
C	H	H	H	H	L	H	L	H	L	H	H	H	H	L	L	H	L
C	L	L	H	H	H	H	L	H	L	H	H	H	L	L	H	H	H
C	L	L	H	H	H	H	L	H	L	H	H	L	L	H	H	H	L
C	L	L	H	H	H	L	L	H	L	H	L	L	H	H	H	L	L
C	H	H	H	H	H	L	L	H	H	H	H	Z	Z	Z	Z	H	H

DP84322 Logic Diagram PAL 16R4



TL/F/5003-12

National Semiconductor

DP84412 Dynamic RAM Controller Interface Series Circuit for the Series 32000® CPU

General Description

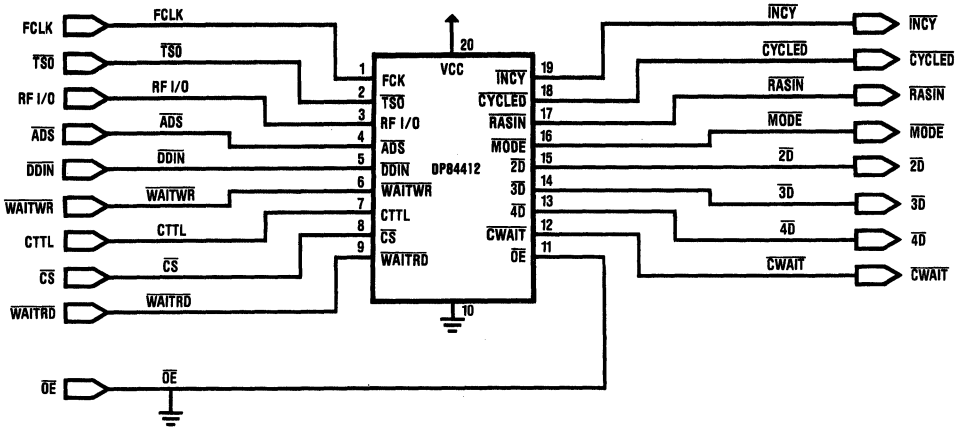
The DP84412 is a new Programmable Array Logic (PAL®) device, that replaces the DP84312, designed to allow an easy interface between the National Semiconductor Series 32000 family of processors and the National Semiconductor DP8409A, DP8429, or DP8419 DRAM controller.

The new DP84412 supplies all the control signals needed to perform memory read, write and refresh and work with the National Semiconductor Series 32000 family of processors up to 10 MHz. Logic is also included to insert WAIT states, if wanted, into the microprocessor READ or WRITE cycles when using fast CPUs.

Features

- Provides a 3-chip solution for the Series 32000 family, dynamic RAM interface (DP8409A or DP8419, DP84412, and clock divider).
- Works with all Series 32000 family speed versions up to 10 MHz.
- Operation of Series 32000 processor at 10 MHz with no WAIT states.
- Controls DP8409A or DP8419 Mode 5 accesses, hidden refreshes and Mode 1 Forced Refreshes automatically.
- Inserts WAIT states in READ or WRITE cycles automatically depending on whether $\overline{\text{WAITRD}}$ or $\overline{\text{WAITWR}}$ are low, or if $\overline{\text{CS}}$ becomes active during a forced Refresh cycle.
- Uses a standard National Semiconductor PAL part (DMPAL16R6A).
- The PAL logic equations can be modified by the user for his specific application and programmed into any of the PALs in the National Semiconductor family, including the new very high speed PALs ("B" PAL parts).

Connection Diagram



TL/F/8397–1

Order Number DP84412J or DP84412N
See NS Package Number N20A or J20A

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

	Operating	Programming		Operating	Programming
Supply Voltage, V_{CC}	7V	12V	Off-State Output Voltage	5.5V	12V
Input Voltage	5.5V	12V	Storage Temperature Range	−65°C to +150°C	

Recommended Operating Conditions

Symbol	Parameter		Commercial			Units
			Min	Typ	Max	
V_{CC}	Supply Voltage		4.75	5	5.25	V
t_w	Width of Clock	Low	15	10		ns
		High	15	10		
t_{su}	Setup Time from Input or Feedback to Clock		25	16		ns
t_h	Hold Time		0	−10		ns
T_A	Operating Free-Air Temperature		0	25	75	°C
T_C	Operating Case Temperature					°C

Electrical Characteristics Over Recommended Operating Temperature Range

Symbol	Parameter	Test Conditions		Min	Typ	Max	Units
V_{IH}	High Level Input Voltage			2			V
V_{IL}	Low Level Input Voltage					0.8	V
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA			−0.8	−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, V_{IL} = 0.8V, V_{IH} = 2V	I_{OH} = −3.2 mA COM	2.4	2.8		V
V_{OL}	Low Level Output Voltage	V_{CC} = Min, V_{IL} = 0.8V, V_{IH} = 2V	I_{OL} = 24 mA COM		0.3	0.5	V
I_{OZH}	Off-State Output Current	V_{CC} = Max, V_{IL} = 0.8V, V_{IH} = 2V	V_O = 2.4V			100	μA
I_{OZL}			V_O = 0.4V			−100	μA
I_I	Maximum Input Current	V_{CC} = Max, V_I = 5.5V				1	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.4V				25	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V			−0.02	−0.25	mA
I_{OS}	Output Short-Circuit Current	V_{CC} = 5V	V_O = 0V	−30	−70	−130	mA
I_{CC}	Supply Current	V_{CC} = Max			120	180	mA

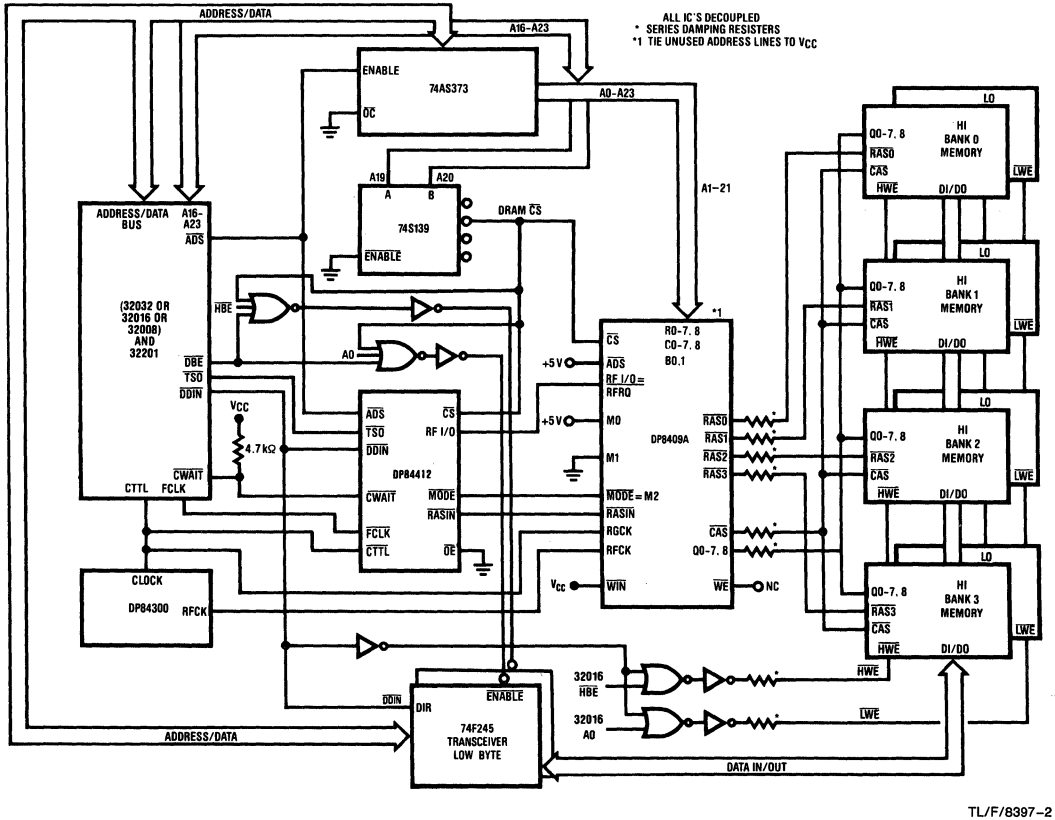
Switching Characteristics Over Recommended Ranges of Temperature and V_{CC}

V_{CC} = 5V ±10%. Commercial: T_A = 0°C to 75°C, V_{CC} = 5V ±5%

Symbol	Parameter	Test Conditions R1, R2	Commercial			Units
			Min	Typ	Max	
t_{PD}	Input or Feedback to Output	CL = 50 pF		15	25	ns
t_{CLK}	Clock to Output or Feedback			10	15	ns
t_{PZX}	Pin 11 to Output Enable			10	20	ns
t_{PXZ}	Pin 11 to Output Disable	C_L = 5 pF		11	20	ns
t_{PZX}	Input to Output Enable	C_L = 50 pF		10	25	ns
t_{PXZ}	Input to Output Disable	C_L = 5 pF		13	25	ns
f_{MAX}	Maximum Frequency		25	30		ns

V_{CC} = Max at minimum temperature.

PAL For Series 32000 Family Systems



TL/F/8397–2

Mnemonic Description

INPUTS SIGNALS

1) "FCLK" Fast clock from the NS32201 TCU clock chip, this signal runs at twice the speed of the system clock.
2) "$\overline{TSO}$" From the NS32201 TCU clock chip, this signal indicates the start of the "T2" state and goes high at the beginning of the "T4" state.
3) "RFI/O" $\overline{RFRQ}$ (refresh request) in mode 5. From 8409A, an active low signal.
4) "$\overline{ADS}$" From the Series 32000 CPU, address strobe. If the system includes the MMU (NS32082) then $\overline{PAV}$ should be connected to this input.
5) "$\overline{DDIN}$" Used to differentiate between READ and WRITE cycles, and to allow $\overline{CS}$ READ cycles to start early.
6) "$\overline{WAITWRITE}$" This signal is used to add a WAIT state into a $\overline{CS}$ WRITE access cycle, and delay $\overline{RASIN}$ until the end of the "T2" clock period.
7) "CTTL" From the NS32201 TCU clock chip, this signal runs at the system clock frequency.
8) "$\overline{CS}$" From decoder chip (chip select) (active low).
9) "$\overline{WAITREAD}$" Used to insert 1 wait state into the Series 32000 READ bus cycle. The wait state allows the use of memory with longer access times (t_{CAC}). An active low signal.
10) "$\overline{OE}$" This input enables the outputs of the "D-Flip Flop" outputs of the PAL.

OUTPUTS SIGNALS

1) "$\overline{MODE}$" This pin goes to M2 on the DP8409A to change from mode 5 to mode 1 (only used for forced refresh).
2) "$\overline{2DLY}$" Delay used internal to the PAL.
3) "$\overline{3DLY}$" Delay used internal to the PAL.
4) "$\overline{4DLY}$" Delay used internal to the PAL.
5) "$\overline{RASIN}$" To the 8409A (creates RASs). Goes low earlier for READ cycles than WRITE cycles.
6) "$\overline{CYCLED}$" Goes active low once a hidden refresh (non $\overline{CS}$ cycle) or DRAM access has been performed. $\overline{CYCLED}$ always goes low at the beginning of the "T3" processor state. This signal goes high (reset) by the end of the processor bus cycle as indicated by $\overline{TSO}$ being high.
7) "$\overline{CWAIT}$" This output inserts "WAIT" or "HOLD" states into the NS32016 machine cycles (only WAIT states are used in this application). This output is in "not enabled" condition when $\overline{CS}$ is high (not chip selected).
8) "$\overline{INCYCLE}$" This signal goes active from the CPU $\overline{ADS}$ signal. This signal indicates that the processor is doing an access somewhere in the system. This signal stays low for several T states of the access cycle.

Functional Description

The following description applies to both the DP8409A and the DP8419 dynamic RAM controllers.

A memory cycle starts when chip select ($\overline{CS}$) and address strobe ($\overline{ADS}$) are true. $\overline{RASIN}$ is supplied from the DP84412 to the DP8409A dynamic RAM controller, which then supplies a $\overline{RAS}$ signal to the selected dynamic RAM bank. After the necessary row address hold time, the DP8409A switches the address outputs to the column address. The DP8409A then supplies the required $\overline{CAS}$ signal to the DRAM. In order to do byte operations it is suggested that the user provide external logic, as shown in the system block diagram, to produce a HIGH WRITE ENABLE and/or a LOW WRITE ENABLE. To differentiate between a READ and a WRITE, the $\overline{DDIN}$ signal from the CPU is used. $\overline{DDIN}$ is also supplied to the external WRITE ENABLE logic.

A refresh cycle is started by one of two conditions. The refresh cycle caused by the first condition is called a hidden refresh. This occurs when refresh clock (RFCK) is high, $\overline{CS}$ is not true, and $\overline{RASIN}$ goes true. Here the CPU is accessing something else in the system and the DRAM can be refreshed at that time, thereby being transparent to the CPU. The second type of refresh is called forced refresh. This occurs if no hidden refresh was performed while RFCK was high. When RFCK transitions low a refresh request ($\overline{RFRQ}$) is generated. If there is not a DRAM access in progress the DP84412 will force a refresh by putting the DP8409A into mode 1 (automatic forced refresh mode). If the CPU tries to access the DRAM during a forced refresh cycle WAIT states will be inserted into its cycles until the forced refresh is over and the DRAM $\overline{RAS}$ precharge time has been met. Then the pending DRAM access will be allowed to take place.

The DP84412 also allows forced refreshes to take place during long accesses of other devices. For instance, if EEPROM takes several microseconds to write to, the DRAM will still be refreshed while that access is in progress.

In a standard memory cycle, the access can be slowed down by one clock cycle to accommodate slower memories or allow time to generate parity. This is accomplished by inserting a WAIT state into the processor access cycle. The DP84412 can insert WAIT states into either READ or WRITE cycles, or both. The extra WAIT state will not appear during the hidden refresh cycle, so faster devices on the CPU bus will not be affected.

System Interface Description

All members of the Series 32000 family of processors are able to use the DP84412.

The DP84412 differentiates between READ and WRITE cycles, allowing the $\overline{RASIN}$ signal to start earlier during a READ cycle compared to a WRITE cycle.

$\overline{RASIN}$ during a READ cycle will always start at the beginning of the "T2" processor cycle. The user must also guarantee that $\overline{CS}$ is valid a minimum of 30 ns before $\overline{RASIN}$ becomes valid. The worst case would be at 10 MHz where FCLK preceeds PHI1 by a maximum of 10 ns. $\overline{RASIN}$ can occur a minimum of approximately 8 ns after FCLK. Therefore $\overline{CS}$ must occur a minimum of 32 ns (30 ns + 2 ns) before the rising edge of PHI1 at 10 MHz.

The user may want to tie $\overline{CS}$ low on the DP8409A/19 (disable HIDDEN REFRESH) and use the system transceivers to select the DRAM. In this case one only needs to concern himself with the 10 ns address setup time to $\overline{RASIN}$.

System Interface Description (Continued)

The DP84412 can be used in a system with the MMU (NS32082) but the signal $\overline{PAV}$ would be connected to the $\overline{ADS}$ input instead of $\overline{ADS}$.

Several other critical parameters in this application that involve the input signals $\overline{DDIN}$, $\overline{CWAIT}$, $\overline{TSO}$, and FCLK. These parameters become most critical at 10 MHz where it is suggested that they are directly connected to the corresponding pins of the Series 32000 family ICs.

This section of the data sheet goes through the calculation of the "tRAC" ($\overline{RAS}$ access time) and "tCAC" ($\overline{CAS}$ access time) required by the DRAM for the Series 32000 family CPUs to operate at a particular clock frequency without introducing wait states into the processor access cycles. Both "tRAC" and "tCAC" must be considered in determining what speed DRAM can be used in a particular system design. The DRAM chosen must meet both the "tRAC" and "tCAC" parameters calculated. In order to determine the "tRAC" and "tCAC" needed the DP8419 and fast PALs ("B" type PALs) timing parameters were used. If the user is using the DP8408A/09A or a slower PAL device he should substitute their respective delays into the equations below.

Most all of the calculations contained in this note use "RAHS" = 1 (15 ns guaranteed minimum row address hold time). Calculations only used "RAHS" = 0 (25 ns guaranteed minimum row address hold time) when the calculated access time from RAS exceeded 200 ns. This is because DRAMs can be found with row access times up to 150 ns that require only 15 ns row address hold times.

EXAMPLE DRAM TIMING CALCULATIONS

A) 8 MHz Series 32000 CPU, No Wait states

#1) $\overline{RASIN}$ = T1 − 2 ns (FCLK to PHI1 skew) + 12 ns ("B" PAL clocked output) = 125 − 2 + 12 = 135 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum (DP8419)

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 80 ns (DP8419 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMs speced in data sheet) = 77 ns

#4) 74F245 transceiver delay = 7 ns maximum

#5) CPU data setup time to "T4" = data setup to PHI2 T.E. + maximum PHI2 F.E. to PHI1 R.E. = 15 + 5 = 20 ns minimum

"tRAC" = T1 + T2 + T3 − #1 − #2 − #4 − #5
= 125 + 125 + 125 − 135 − 20 − 7 − 20 = 193 ns

"tCAC" = T1 + T2 + T3 − #1 − #3 − #4 − #5
= 125 + 125 + 125 − 135 − 77 − 7 − 20 = 136 ns

Therefore the DRAM chosen should have a "tRAC" less then or equal to 193 ns and a "tCAC" less than or equal to 136 ns. Standard 150 ns DRAMs meet this criteria.

The minimum $\overline{RAS}$ PRECHARGE TIME will be approximately one and one half clock periods = 125 + 62 = 187 ns.

The minimum $\overline{CAS}$ PRECHARGE TIME will be approximately one and one half clock periods plus 35 ns (minimum $t_{RICL} - t_{RICH}$ for the DP8409-2) = 125 + 62 + 35 = 222 ns.

The minimum $\overline{RAS}$ PULSE WIDTH will be approximately two clock periods − 5 ns (maximum $t_{RPDL} - t_{RPDH}$ for the DP8409-2) = 250 − 5 = 245 ns.

The minimum $\overline{CAS}$ PULSE WIDTH will be approximately two clock periods − 70 ns (maximum $t_{RICL} - t_{RICH}$ for the DP8409-2) = 250 − 70 = 180 ns.

The smallest pulse widths are generated during WRITE cycles since $\overline{RASIN}$ during WRITE cycles starts later than $\overline{RASIN}$ during READ cycles.

If one inserted a WAIT state in READ cycles the DRAM column access times and the $\overline{RAS}$ pulse width would be increased by one clock period (125 ns in this case). A WAIT state in WRITE cycles would just increase the $\overline{RAS}$ pulse width by one clock period.

B) 10 MHz Series 32000, No Wait States

#1) $\overline{RASIN}$ low = T1 − 2 ns (FCLK − PHI1 skew) + 12 ns ("B" PAL clocked output) = 100 − 2 + 12 = 110 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 80 ns maximum (DP8419 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMs speced in data sheet) = 77 ns

#4) 74F245 transceiver delay = 7 ns maximum

#5) CPU data setup time to "T4" = data setup to PHI2 T.E. + maximum PHI2 F.E. to PHI1 R.E. = 15 + 5 = 15 ns minimum

"tRAC" = T1 + T2 + T3 − #1 − #2 − #4 − #5
= 100 + 100 + 100 − 110 − 20 − 7 − 15 = 148 ns

"tCAC" = T1 + T2 + T3 − #1 − #3 − #4 − #5
= 100 + 100 + 100 − 110 − 77 − 7 − 15 = 91 ns

Therefore the DRAM chosen should have a "tRAC" less then or equal to 148 ns and a "tCAC" less than or equal to 91 ns. Standard 120 ns DRAMs meet this criteria.

The minimum $\overline{RAS}$ PRECHARGE TIME will be approximately one and one half clock periods = 100 + 50 = 150 ns.

The minimum $\overline{CAS}$ PRECHARGE TIME will be approximately one and one half clock periods plus 35 ns (minimum $t_{RICL} - t_{RICH}$ for the DP8409-2) = 100 + 50 + 35 = 185 ns.

The minimum $\overline{RAS}$ PULSE WIDTH will be approximately two clock periods − 5 ns (maximum $t_{RPDL} - t_{RPDH}$ for the DP8409-2) = 200 − 5 = 195 ns.

The minimum $\overline{CAS}$ PULSE WIDTH will be approximately two clock periods − 70 ns (maximum $t_{RICL} - t_{RICH}$ for the DP8409-2) = 200 − 70 = 130 ns.

The smallest pulse widths are generated during WRITE cycles since $\overline{RASIN}$ during WRITE cycles starts later than $\overline{RASIN}$ during READ cycles.

If one inserted a WAIT state in READ cycles the DRAM column access times and the $\overline{RAS}$ pulse width would be increased by one clock period (100 ns in this case). A WAIT state in WRITE cycles would just increase the $\overline{RAS}$ pulse width by one clock period.

SUGGESTIONS

It is suggested that the DP8409A could be used up to 8 MHz. Above 8 MHz one should use the DP8409-2 or the DP8419. Also, fast PALs ("A" or "B" parts) should be used at 8 MHz and above.

INTERPRETING THE DP84412 PAL EQUATIONS

The boolean equations for the DP84412 were written using the standard PALASM™ format. In other words the equation: "IF (V_{CC}) RASIN = INCY*$\overline{MODE}$*4D*DDIN" will mean;

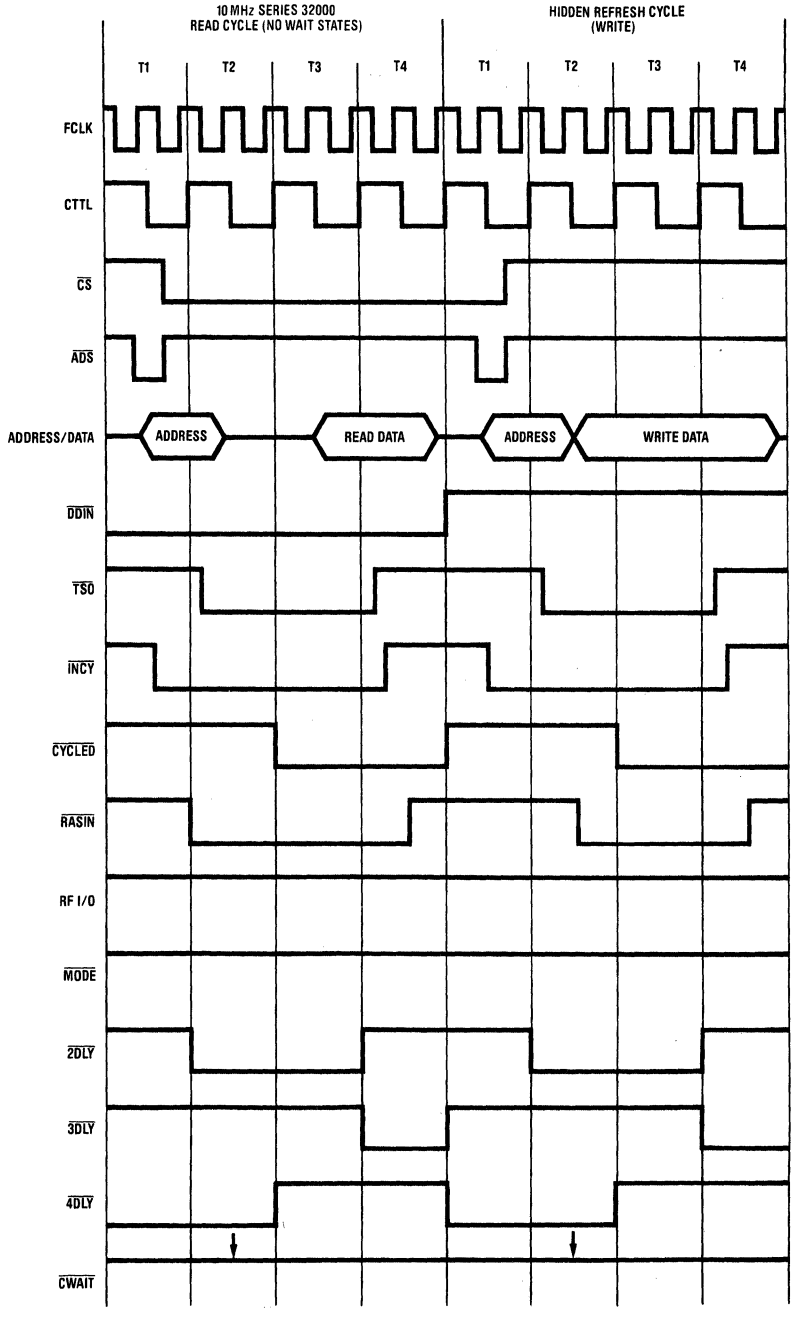
The output "$\overline{RASIN}$" (see pin list for DP84412) will be active low (inverted RASIN) when the output "$\overline{INCY}$" is low (making INCY high) AND the output "$\overline{MODE}$" is high AND the output "$\overline{4D}$" is low (making 4D high) AND the input $\overline{DDIN}$ is low (making DDIN high).

PAL Boolean Equations

```
PAL16R6A     ;FAST PAL
NEW PAL FOR THE NATIONAL SEMICONDUCTOR NS32016, 32008, 32032
NATIONAL SEMICONDUCTOR (WORKS UP 10 MHz)

FCLK /TSO RFIO /ADS /DDIN /WAITWR CTTL /CS /WAITRD GND
/OE /CWAIT /4DLY /3DLY /2DLY /MODE /RASIN /CYCLED /INCY VCC

RASIN := INCY*/CYCLED*/MODE*/CTTL*DDIN+      ;Start /RASIN fast during
                                             ; "READ" cycle
         INCY*/MODE*2DLY*/WAITWR+            ; ''WRITE'' cycle without WAIT states
         /CS•INCY•/MODE•2DLY+                ; Hidden Refresh /RASIN
         CS•INCY•/MODE•2DLY•WAITWR•/CTTL+     ; ''WRITE'' cycle with WAIT states
         RASIN•INCY•/MODE•2DLY               ; continue /RASIN

CYCLED := /MODE*2DLY*/WAITWR*/DDIN*/CTTL+          ;No WAITS inserted
          /MODE*2DLY*/WAITRD*/DDIN*/CTTL+          ;No WAITS inserted
          /MODE*2DLY*4DLY*/WAITRD*/DDIN*/CTTL+     ;WAIT in READ cycle
          /MODE*2DLY*4DLY*/WAITWR*/DDIN*/CTTL+     ;WAIT in WRITE cycle
          CYCLED*TSO*/MODE+
          CYCLED*/MODE*CTTL

MODE := /RFIO*/INCY*2DLY*/CTTL+              ;forced refresh during idle
                                             ;states, in long cycles,
        MODE*3DLY+                           ;or at the end of a cycle
        MODE*4DLY+
        MODE*CTTL

2DLY := MODE*/4DLY*/CTTL+
        2DLY*CTTL+
        INCY*/CYCLED*/MODE*/3DLY*4DLY*/CTTL+
        CS*DDIN*WAITRD*INCY*/MODE*2DLY*/3DLY*/4DLY+    ;extend 2DLY if
        CS*/DDIN*WAITWR*INCY*/MODE*2DLY*/3DLY*/4DLY    ;  WAIT states
                                                       ;    are wanted

3DLY := 2DLY*/4DLY*/CTTL+
        3DLY*CTTL

4DLY := 3DLY*/CTTL+
        4DLY*CTTL+
        /INCY*/MODE*/CTTL+
        INCY*/MODE*/2DLY*/CTTL

IF (VCC) INCY = ADS*/MODE+
           CS*TSO*/CYCLED*/MODE*/2DLY*4DLY+    ;Start INCY for CS
           INCY*/CYCLED+                       ;access after forced
           INCY*2DLY                           ;refresh

IF (CS) CWAIT=CS*TSO*/CYCLED*/MODE*/2DLY*4DLY+  ;for Access during
                                                ;forced refresh
   CS*TSO*MODE+                                 ;during forced refresh
   CS*INCY*/CYCLED*DDIN*WAITRD*/MODE*2DLY*/3DLY*4DLY+
                                                ; /CS READ cycle with
                                                ; WAIT states
   CS*INCY*/CYCLED*/DDIN*WAITWR*/MODE*2DLY*/3DLY*4DLY
                                                ; /CS WRITE cycle with
                                                ; WAIT states
```

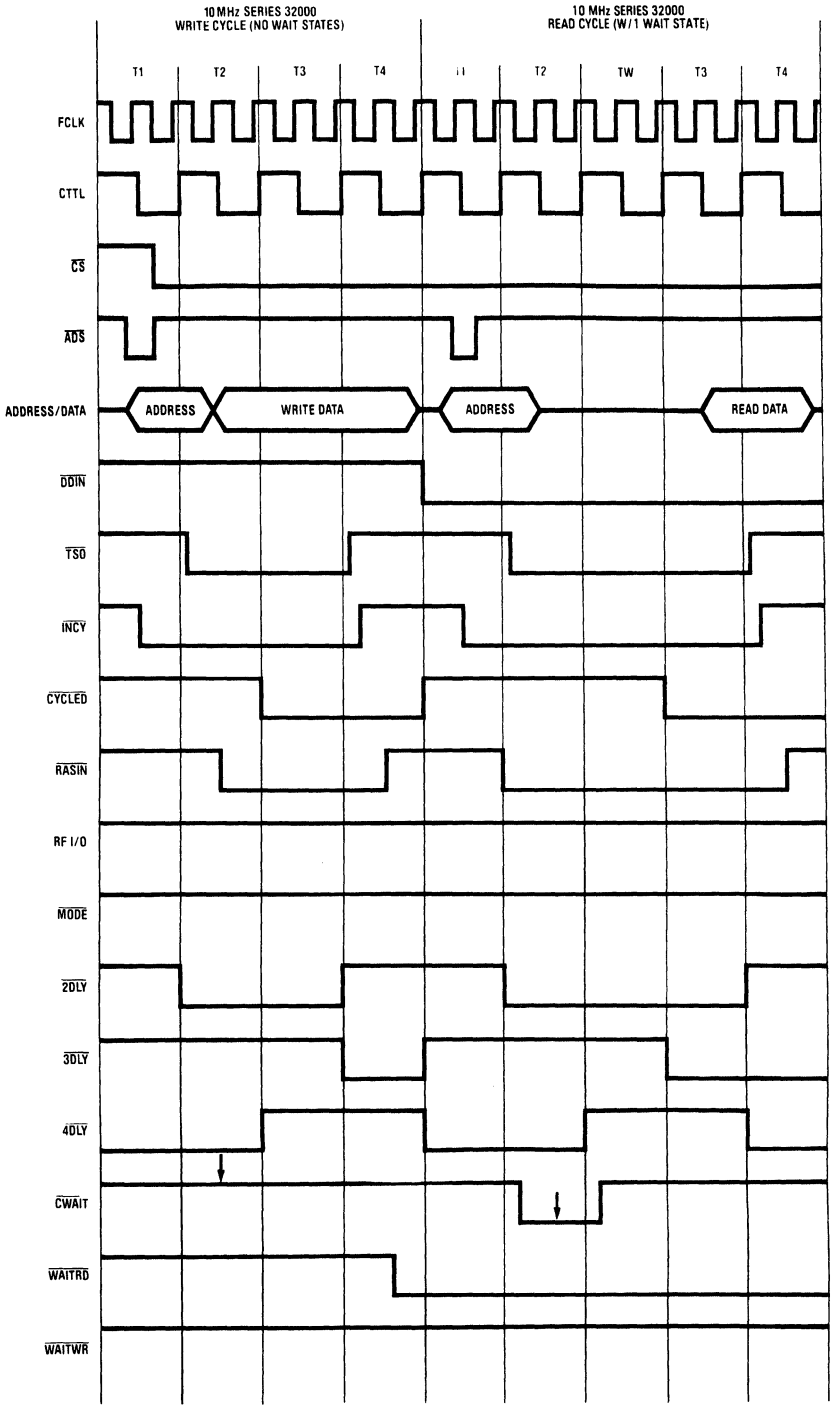
FIGURE 1. Equations for the Series 32000 Family Interface PAL

System Timing Diagrams



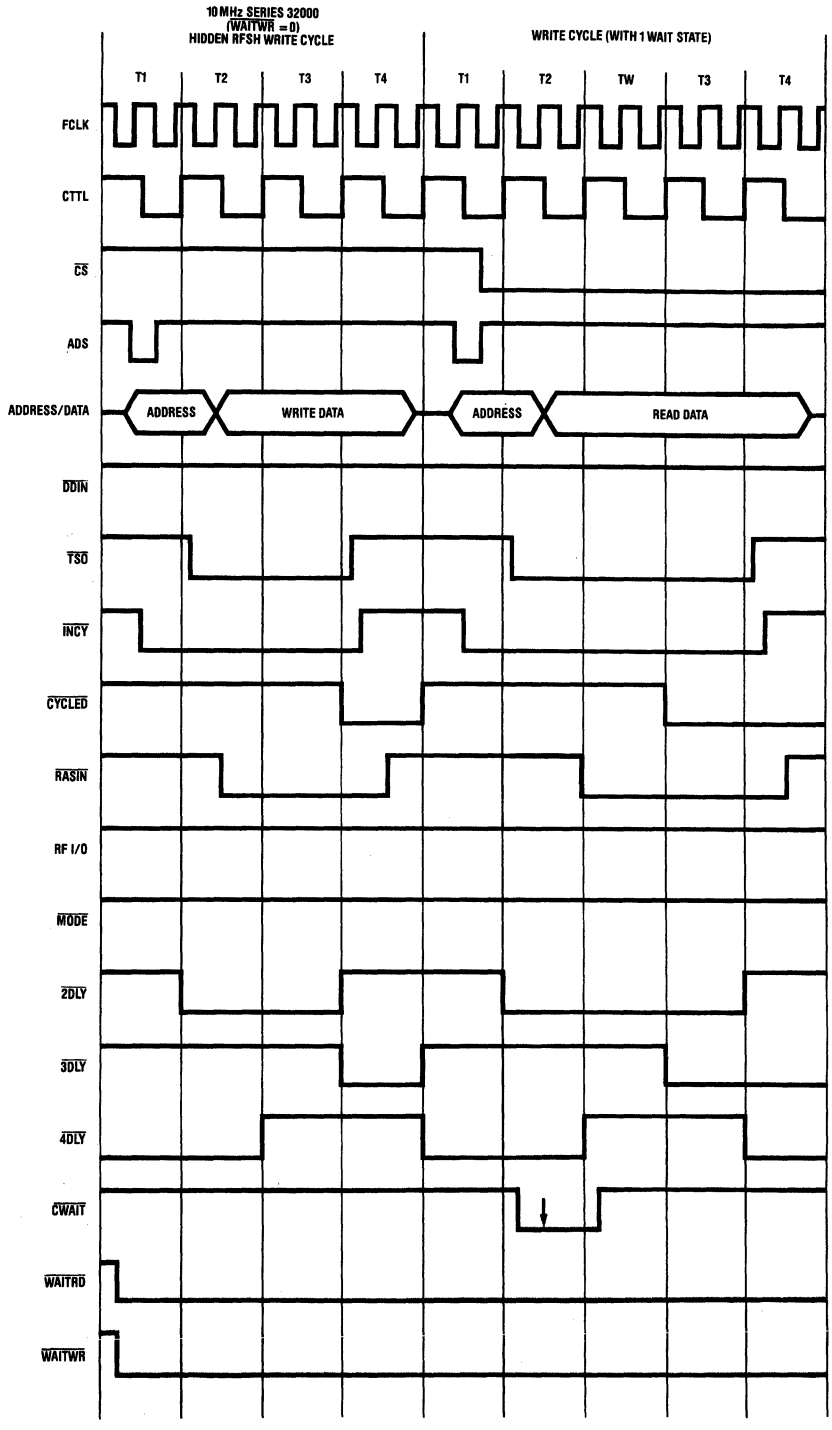
TL/F/8397–3

System Timing Diagrams (Continued)



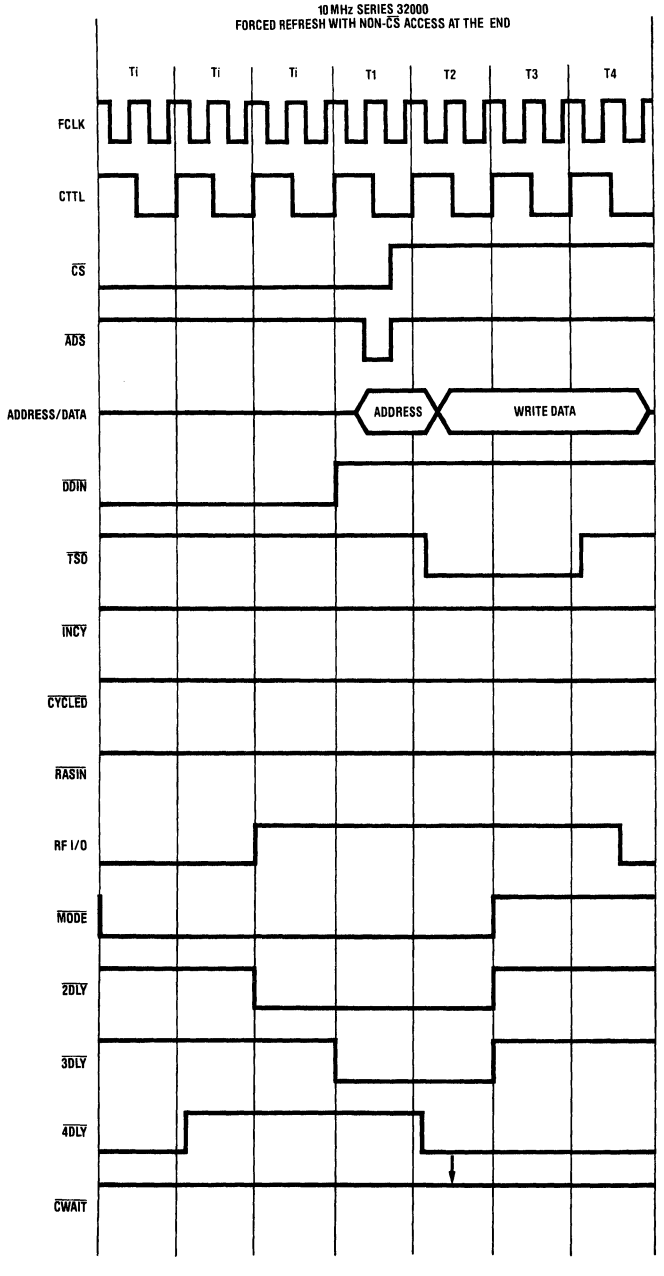
TL/F/8397–4

System Timing Diagrams (Continued)



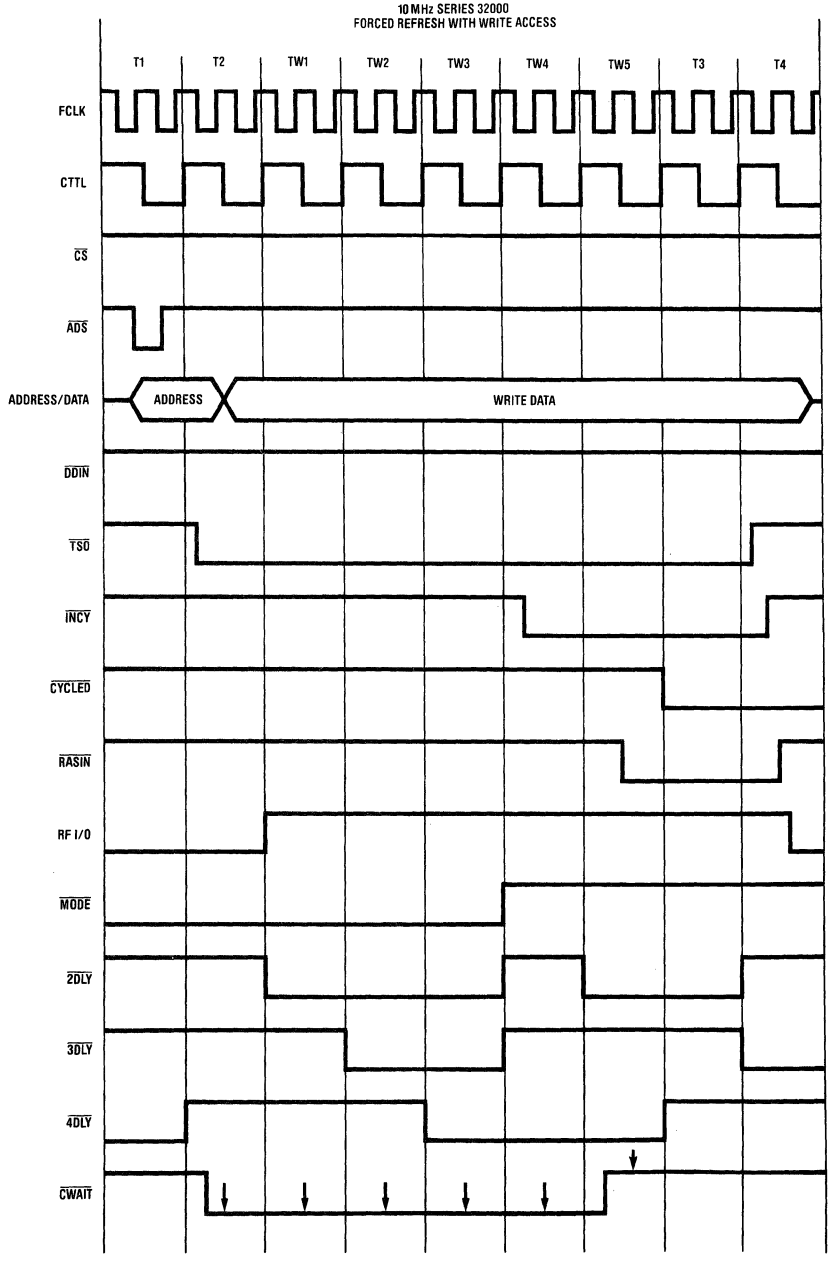
TL/F/8397–5

System Timing Diagrams (Continued)



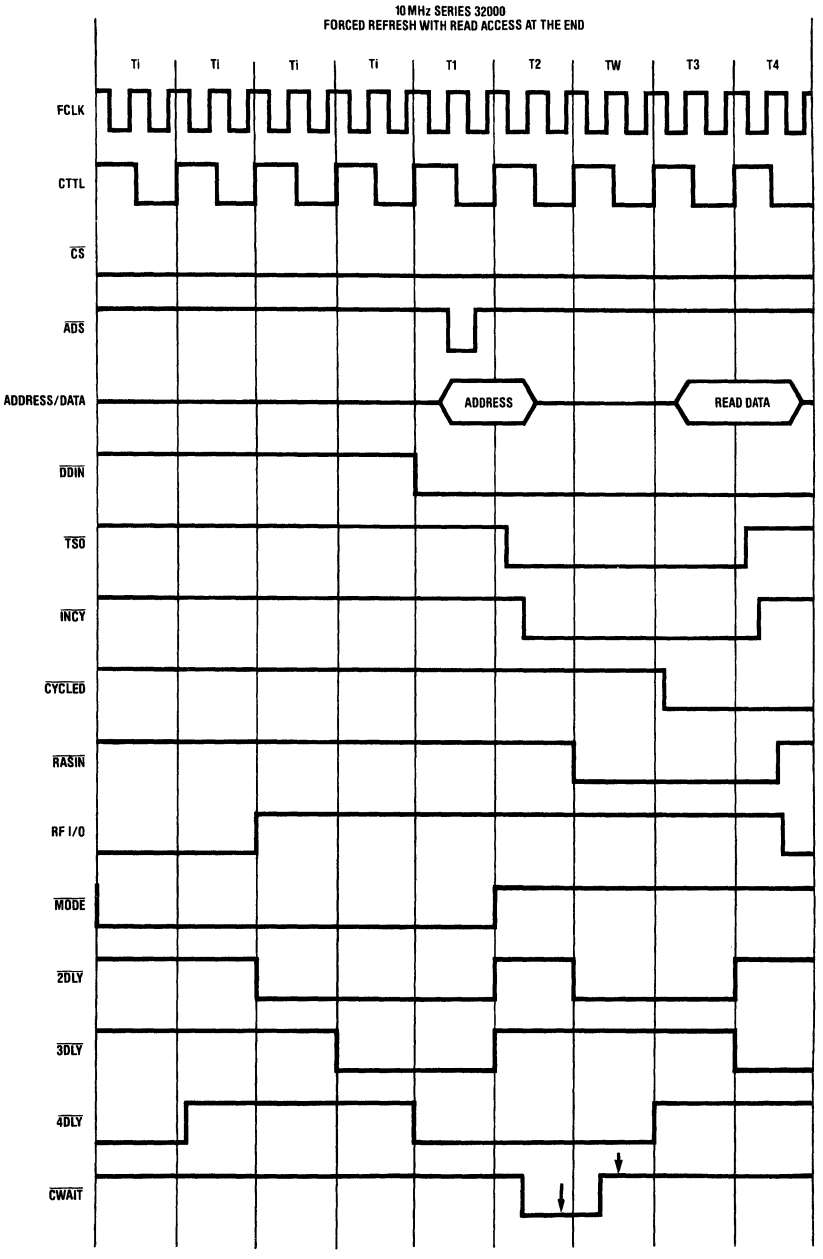
TL/F/8397–6

System Timing Diagrams (Continued)



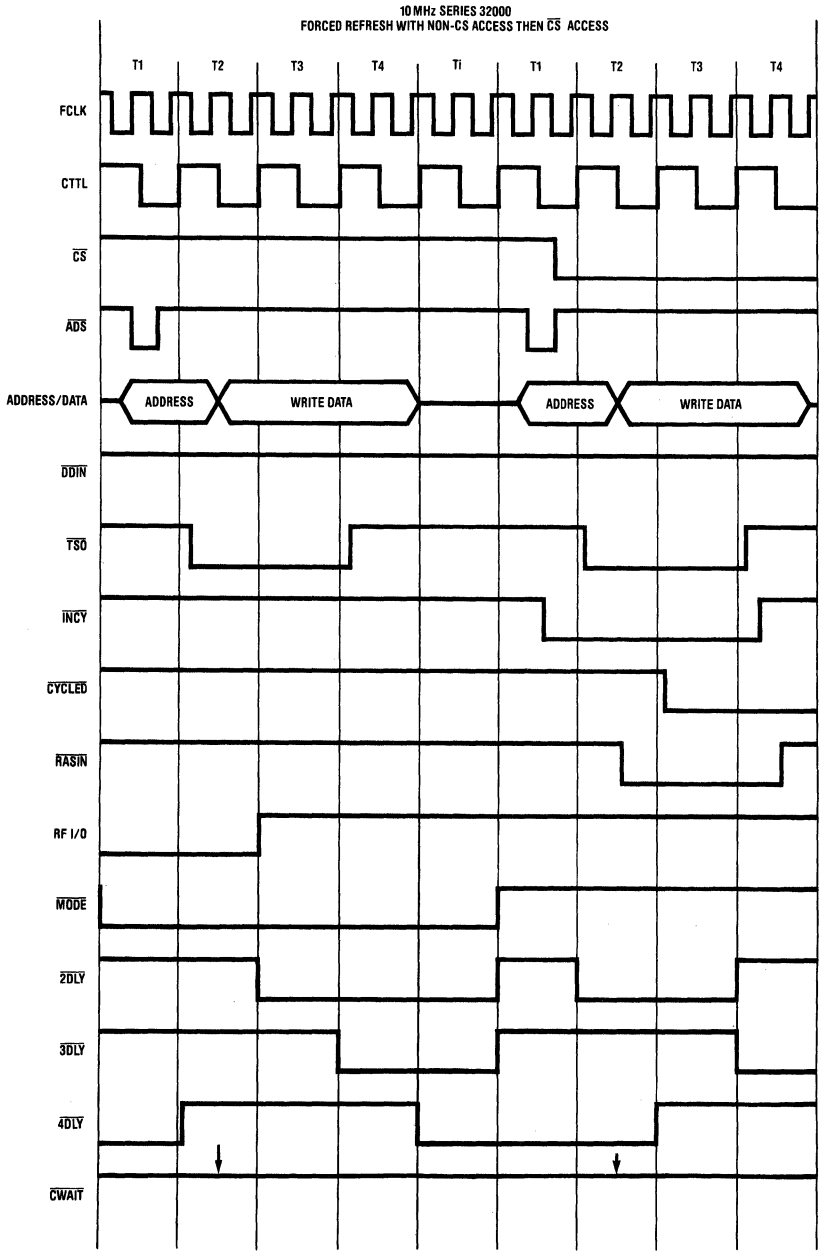
TL/F/8397–7

System Timing Diagrams (Continued)



TL/F/8397–8

System Timing Diagrams (Continued)



TL/F/8397–9

National Semiconductor

PRELIMINARY

DP84422 Dynamic RAM Controller Interface Circuit for the 68000/008/010 CPU(s)

General Description

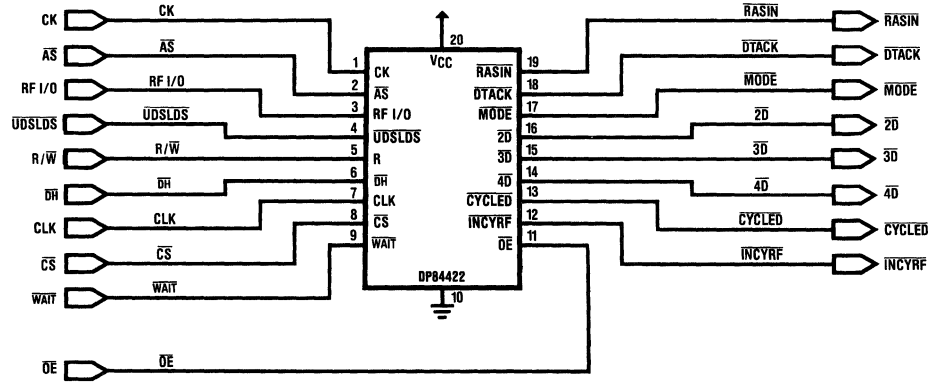
The DP84422 is a new Programmable Array Logic (PAL®) device, that replaces the DP84322, designed to allow an easy interface between the Motorola 68000 family of processors and the National Semiconductor DP8409A, DP8429, or DP8419 DRAM controller.

The new DP84422 supplies all the control signals needed to perform memory read, write, read modify write (as in the Test and Set, "TAS", instruction), and refresh and work with the 68000 family of processors up to 12.5 MHz. Logic is also included to insert WAIT states, if wanted, into the microprocessor READ or WRITE cycles when using fast CPUs.

Features

- Provides a 3-chip solution for the 68000 family, dynamic RAM interface (DP8409A or DP8419, DP84422, and clock divider).
- Works with all 68000 family speed versions up to 12.5 MHz.—(68008; 68000; and 68010).
- Operation of 68000 processor at 10 MHz with no WAIT states.
- Controls DP8409A or DP8419 Mode 5 accesses, hidden refreshes and Mode 1 Forced Refreshes automatically.
- Inserts WAIT states in READ or WRITE cycles automatically depending on when $\overline{WAIT}$ is low, or if chip select becomes active during a forced Refresh cycle.
- Uses a standard National Semiconductor PAL part (DMPAL16R4A).
- The PAL logic equations can be modified by the user for his specific application and programmed into any of the PALs in the National Semiconductor family, including the new very high speed PALs ("B" PAL parts).

Connection Diagram



TL/F/8398–1

Order Number DP84422J or DP84422N
See NS Package J20A or N20A

3

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

	Operating	Programming
Supply Voltage, V_{CC}	7V	12V
Input Voltage	5.5V	12V

	Operating	Programming
Off-State Output Voltage	5.5V	12V
Storage Temperature Range	−65°C to +150°C	

Recommended Operating Conditions

Symbol	Parameter		Commercial			Units
			Min	Typ	Max	
V_{CC}	Supply Voltage		4.75	5	5.25	V
t_w	Width of Clock	Low	15	10		ns
		High	15	10		
t_{su}	Setup Time from Input or Feedback to Clock		25	16		ns
t_h	Hold Time		0	−10		ns
T_A	Operating Free-Air Temperature		0	25	75	°C
T_C	Operating Case Temperature					°C

Electrical Characteristics Over Recommended Operating Temperature Range

Symbol	Parameter	Test Conditions		Min	Typ	Max	Units
V_{IH}	High Level Input Voltage			2			V
V_{IL}	Low Level Input Voltage					0.8	V
V_{IC}	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA			−0.8	−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min, V_{IL} = 0.8V, V_{IH} = 2V	I_{OH} = −3.2 mA COM	2.4	2.8		V
V_{OL}	Low Level Output Voltage	V_{CC} = Min, V_{IL} = 0.8V, V_{IH} = 2V	I_{OL} = 24 mA COM		0.3	0.5	V
I_{OZH}	Off-State Output Current	V_{CC} = Max, V_{IL} = 0.8V, V_{IH} = 2V	V_O = 2.4V			100	μA
I_{OZL}			V_O = 0.4V			−100	μA
I_I	Maximum Input Current	V_{CC} = Max, V_I = 5.5V				1	mA
I_{IH}	High Level Input Current	V_{CC} = Max, V_I = 2.4V				25	μA
I_{IL}	Low Level Input Current	V_{CC} = Max, V_I = 0.4V			−0.02	−0.25	mA
I_{OS}	Output Short-Circuit Current	V_{CC} = 5V	V_O = 0V	−30	−70	−130	mA
I_{CC}	Supply Current	V_{CC} = Max			120	180	mA

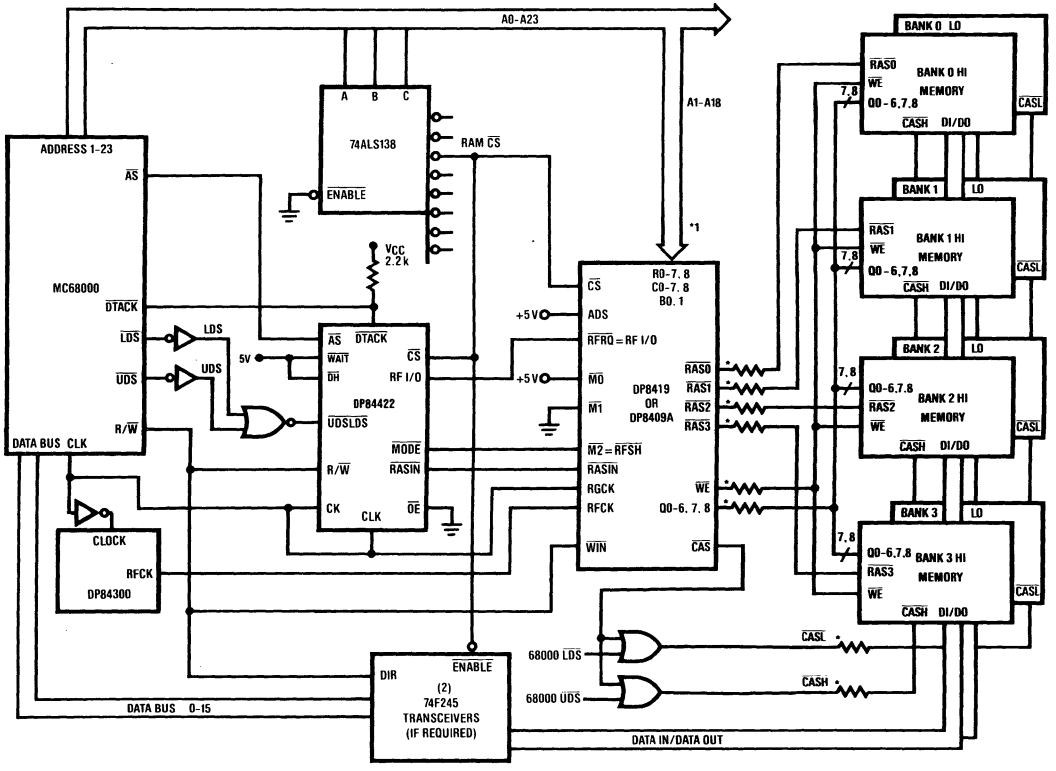
Switching Characteristics Over Recommended Ranges of Temperature and V_{CC}

V_{CC} = 5V ± 10% Commercial: T_A = 0 to 75°C, V_{CC} = 5V ± 5%

Symbol	Parameter	Test Conditions R1, R2	Commercial			Units
			Min	Typ	Max	
t_{PD}	Input or Feedback to Output	CL = 50 pF		15	25	ns
t_{CLK}	Clock to Output of Feedback			10	15	ns
t_{PZX}	Pin 11 to Output Enable			10	20	ns
t_{PXZ}	Pin 11 to Output Disable	C_L = 5 pF		11	20	ns
t_{PZX}	Input to Output Enable	C_L = 50 pF		10	25	ns
t_{PXZ}	Input to Output Disable	C_L = 5 pF		13	25	ns
f_{MAX}	Maximum Frequency		25	30		ns

V_{CC} = Max. at minimum temperature

ALL IC'S DECOUPLED
* SERIES DAMPING RESISTERS
*1 TIE UNUSED ADDRESS LINES TO V_{CC}
This circuit provides direct support of the 68000 Test and Set Instruction using PAGE MODE DRAMs.



TL/F/8398–2

Mnemonic Description

INPUT SIGNALS

1) "CLK", "CK" This is the 68000 CPU clock.
2) "$\overline{AS}$" This is the 68000 address strobe pin. This signal also tells when the 68000 is in a cycle.
3) "$\overline{CS}$" This is the chip select signal for the DP8409A.
4) "R" This is the READ/$\overline{WRITE}$ pin from the 68000.
5) "RFIO" This is the RFIO, used as refresh request, from the DP8409A.
6) "$\overline{WAIT}$" This pin allows the insertion of 1 WAIT state in a $\overline{CS}$ Access cycle if low. As an example; if the user wants 1 WAIT state in READ accesses but 0 WAIT states in WRITE accesses he can invert the "R/$\overline{W}$" input to this input.
7) "$\overline{UDSLDS}$" This input was produced by inverting the two terms $\overline{UDS}$ and $\overline{LDS}$ and then logically "NOR"ing them together. This input is low whenever one or both $\overline{UDS}$ or $\overline{LDS}$ are low. This pin is used in order to support the 68000 "TAS" instruction. This signal is used in the "$\overline{DTACK}$" PAL output.
8) "$\overline{DH}$" This input allows the user to disable the DP8409A/19 hidden refresh, when low, provided he also ties "$\overline{CS}$" low on the DP8409A/19. When this input is low "$\overline{RASIN}$" is only brought low when a "$\overline{CS}$" access ("$\overline{CS}$" input to PAL low) is in progress
9) "$\overline{OE}$" Must be tied low to enable DP84422 outputs.

OUTPUT SIGNALS

1) "$\overline{CYCLED}$" This signal goes low once a hidden refresh or an access has been done as indicated by $\overline{2DLY}$ and $\overline{3DLY}$ being low. This signal goes high once the cycle is over as indicated by $\overline{AS}$ going high. See also "$\overline{DH}$ input
2) "$\overline{RASIN}$" This signal goes low following $\overline{AS}$ during an access or hidden refresh. See also "$\overline{DH}$" input.
3) "$\overline{DTACK}$" This signal causes WAIT states to be inserted into the 68000 processor cycles if it is not low a setup time before S4 falling clock edge.
4) "$\overline{INCYRF}$" This signal indicates that an access has been requested during a forced refresh cycle. This signal is used to insert WAIT states during the forementioned condition or to prevent a "non-$\overline{CS}$" access cycle from automatically starting.
5) "$\overline{MODE}$" This signal is used to pull the DP8409A pin M2 low in order to go to mode 1 to do a forced refresh.
6) "$\overline{2DLY}$" This signal is an internal delay.
7) "$\overline{3DLY}$" This signal is an internal delay.
8) "$\overline{4DLY}$" This signal is an internal delay.

Functional Description

The following description applies to both the DP8409A, DP8429, and the DP8419 dynamic RAM controllers.

A memory cycle starts when chip select ($\overline{CS}$) and address strobe ($\overline{AS}$) are true. $\overline{RASIN}$ is supplied from the DP84422 to the DP8409A dynamic RAM controller, which then supplies a $\overline{RAS}$ signal to the selected dynamic RAM bank. After the necessary row address hold time, the DP8409A switches the address outputs to the column address. The DP8409A then supplies the required $\overline{CAS}$ signal to the DRAM. In order to do byte operations it is suggested that the user provide external logic, as shown in the system block diagram, to produce a HIGH CAS and a LOW CAS. To differentiate between a READ and a WRITE, the R/$\overline{W}$ signal from the CPU is used.

A refresh cycle is started by one of two conditions. The refresh cycle caused by the first condition is called a hidden refresh. This occurs when refresh clock (RFCK) is high, $\overline{CS}$ is not true, and $\overline{RASIN}$ goes low. Here the CPU is accessing something else in the system and the DRAM can be refreshed at that time, thereby being transparent to the CPU. The second type of refresh is called forced refresh. This occurs if no hidden refresh was performed while RFCK was high. When RFCK transitions low a refresh request (RFRQ) is generated. If there is not a DRAM access in progress the DP84422 will force a refresh by putting the DP8409A into mode 1 (automatic forced refresh mode). If the CPU tries to access the DRAM during a forced refresh cycle WAIT states will be inserted into its cycles until the forced refresh is over and the DRAM $\overline{RAS}$ precharge time has been met. Then the pending DRAM access will be allowed to take place.

The DP84422 also allows forced refreshes to take place during long accesses of other devices. For instance, if EEPROM takes several microseconds to write to, the DRAM will still be refreshed while that access is in progress.

In a standard memory cycle, the access can be slowed down by one clock cycle to accommodate slower memories or allow time to generate parity. This is accomplished by inserting a WAIT state into the processor access cycle. The DP84422 can insert WAIT states into either READ cycles, WRITE cycles, READ MODIFY WRITE cycles, or both READ and WRITE cycles or the READ and WRITE portion of a READ MODIFY WRITE cycle. The extra WAIT state will not appear during the hidden refresh cycle, so faster devices on the CPU bus will not be affected.

During a Test and Set instruction $\overline{CAS}$ is generated twice while $\overline{RAS}$ is low. In order for this instruction to execute properly Page Mode DRAMs must be used.

System Interface Description

All members of the Motorola 68000 family of processors are able to use the DP84422.

$\overline{RASIN}$ during a READ cycle will always start at the beginning of the "S3" processor cycle. The user must guarantee that $\overline{CS}$ is valid a minimum of 34 ns before $\overline{RASIN}$ becomes valid, unless the PAL "$\overline{DH}$" input is low and the DP8409A/19 "$\overline{CS}$" input is tied low (hidden refresh disabled).

System Interface Description (Continued)

Several critical parameters in this application involve the input system CLOCK and the ADDRESS STROBE, $\overline{AS}$. These parameters become most critical at higher frequencies (10 MHz and above) where it is suggested that they are directly connected to the corresponding pins of the Motorola 68000 family ICs.

This section of the data sheet goes through the calculation of the "t_{RAC}" ($\overline{RAS}$ access time) and "t_{CAC}" ($\overline{CAS}$ access time) required by the DRAM for the 68000 family CPUs to operate at a particular clock frequency without introducing wait states into the processor access cycles. Both "t_{RAC}" and "t_{CAC}" must be considered in determining what speed DRAM can be used in a particular system design. The DRAM chosen must meet both the "t_{RAC}" and "t_{CAC}" parameters calculated. In order to determine the "t_{RAC}" and "t_{CAC}" needed the DP8419 and fast PALs ("B" type PALs) timing parameters were used. If the user is using the DP8408A/09A or a slower PAL device he should substitute their respective delays into the equation below.

Most all of the calculations contained in this note use "RAHS" = 1 (15 ns guaranteed minimum row address hold time). Calculations only used "RAHS" = 0 (25 ns guaranteed minimum row address hold time) when the calculated access time from RAH exceeded 200 ns. This is because DRAMs can be found with row access times up to 150 ns that require only 15 ns row address hold times.

The calculated "t_{RAC}" and "t_{CAC}" may differ from the actual system values depending upon the external circuitry used to produce "$\overline{CASH}$" and "$\overline{CASL}$". The DP8409A/19 "$\overline{RASIN}$–$\overline{CAS}$" low will be approximately 10–15 ns less than the value given in the data sheet because of the small loading on the DP8409A/19 "$\overline{CAS}$" output. The external circuitry needed to produce "$\overline{CASH, L}$" should be loaded such that the column address (from DP8409A/19 is valid when "$\overline{CASH, L}$" goes low. For this reason "$\overline{RASIN}$–$\overline{CASH, L}$" may be longer than the value used in the "t_{RAC}, t_{CAC}" calculations, and therefore may give a smaller "t_{RAC}, t_{CAC}" then was calculated.

EXAMPLE DRAM TIMING CALCULATIONS

A) 8 MHz 68000, No WAIT States

#1) $\overline{RASIN}$ low = S0 + S1 + $\overline{AS}$ low (maximum) + "B" PAL combinational output delay maximum = 125 + 60 + 15 = 220 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 80 ns (DP8419 $\overline{RASIN}$ – $\overline{CAS}$ low) – 3 ns (load of 72 DRAMs instead of 88 DRAMs speced in data sheet) = 77 ns

#4) 74F245 transceiver delay = 7 ns maximum

#5) CPU data setup time = 15 ns minimum

"t_{RAC}" = (S0 + S1) + (S2 + S3) + (S4 + S5) + S6 (min) – #1 – #2 – #4 – #5
= 125 + 125 + 125 + 55 – 200 – 20 – 7 – 15
= 188 ns

"t_{CAC}" = (S0 + S1) + (S2 + S3) + (S4 + S5) + S6 (min) – #1 – #3 – #4 – #5
= 125 + 125 + 125 + 55 – 200 – 77 – 7 – 15
= 131 ns

Therefore the DRAM chosen should have a "t_{RAC}" less than or equal to 188 ns and a "t_{CAC}" less than or equal to 131 ns. Standard 150 ns DRAMs meet this criteria.

The minimum $\overline{RAS}$ PRECHARGE TIME will be approximately one and one half clock periods = 125 + 55 = 180 ns.

The minimum $\overline{CAS}$ PRECHARGE TIME will be approximately one and one half clock periods plus 35 ns (minimum $t_{RICL} - t_{RICH}$ for the DP8409-2) = 125 + 55 + 35 = 215 ns.

The minimum $\overline{RAS}$ PULSE WIDTH will be approximately two clock periods – 5 ns (maximum $t_{RPDL} - t_{RPDH}$ for the DP8409-2) = 250 – 5 = 245 ns.

The minimum $\overline{CAS}$ PULSE WIDTH will be approximately two clock periods – 70 ns (maximum $t_{RICL} - t_{RICH}$ for the DP8409-2) = 250 – 70 = 180 ns.

The smallest pulse widths are generated during WRITE cycles since $\overline{RASIN}$ during WRITE cycles starts later than $\overline{RASIN}$ during READ cycles.

If one inserted a WAIT state in READ cycles the DRAM column access times, the $\overline{CAS}$ pulse width, and the $\overline{RAS}$ pulse width would be increased by one clock period (125 ns in this case). A WAIT state in WRITE cycles would just increase the $\overline{RAS}$ and $\overline{CAS}$ precharge by one clock period.

B) 10 MHz 68000, No WAIT states

#1) $\overline{RASIN}$ low = S0 + S1 + $\overline{AS}$ low (maximum) + "B" PAL combinational output delay maximum = 100 + 55 + 15 = 170 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 80 ns (DP8419 $\overline{RASIN}$ – $\overline{CAS}$ low) – 3 ns (load of 72 DRAMs instead of 88 DRAMs speced in data sheet) = 77 ns

#4) 74F245 transceiver delay = 7 ns maximum

#5) CPU data setup time = 10 ns minimum

"t_{RAC}" = (S0 + S1) + (S2 + S3) + (S4 + S5) + S6 (min) – #1 – #2 – #4 – #5
= 100 + 100 + 100 + 45 – 170 – 20 – 7 – 10
= 138 ns

"t_{CAC}" = (S0 + S1) + (S2 + S3) + (S4 + S5) – S6 (min) – #1 – #3 – #4 – #5
= 100 + 100 + 100 + 45 – 170 – 77 – 7 – 10
= 81 ns

Therefore the DRAM chosen should have a "t_{RAC}" less than or equal to 138 ns and a "t_{CAC}" less than or equal to 81 ns. Standard 120 ns DRAMs meet this criteria.

The minimum $\overline{RAS}$ PRECHARGE TIME will be approximately one and one half clock periods = 100 + 45 = 145 ns.

The minimum $\overline{CAS}$ PRECHARGE TIME will be approximately one and one half clock periods plus 35 ns (minimum $t_{RICL} - t_{RICH}$ for the DP8419) = 100 + 45 + 35 = 180 ns.

The minimum $\overline{RAS}$ PULSE WIDTH will be approximately two clock periods – 5 ns (maximum $t_{RPDL} - t_{RPDH}$ for the DP8419) = 200 – 5 = 195 ns.

The minimum $\overline{CAS}$ PULSE WIDTH will be approximately two clock periods – 50 ns (maximum $t_{RICL} - t_{RICH}$ for the DP8419) = 200 – 50 = 150 ns.

The smallest pulse widths are generated during WRITE cycles since $\overline{RASIN}$ during WRITE cycles starts later than $\overline{RASIN}$ during READ cycles.

If one inserted a WAIT state in READ cycles the DRAM column access times, the $\overline{CAS}$ pulse width, and the $\overline{RAS}$ pulse width would be increased by one clock period (100 ns in this case). A WAIT state in WRITE cycles would just increase the $\overline{RAS}$ and $\overline{CAS}$ precharge by one clock period.

Interpreting the DP84422 PAL Equations

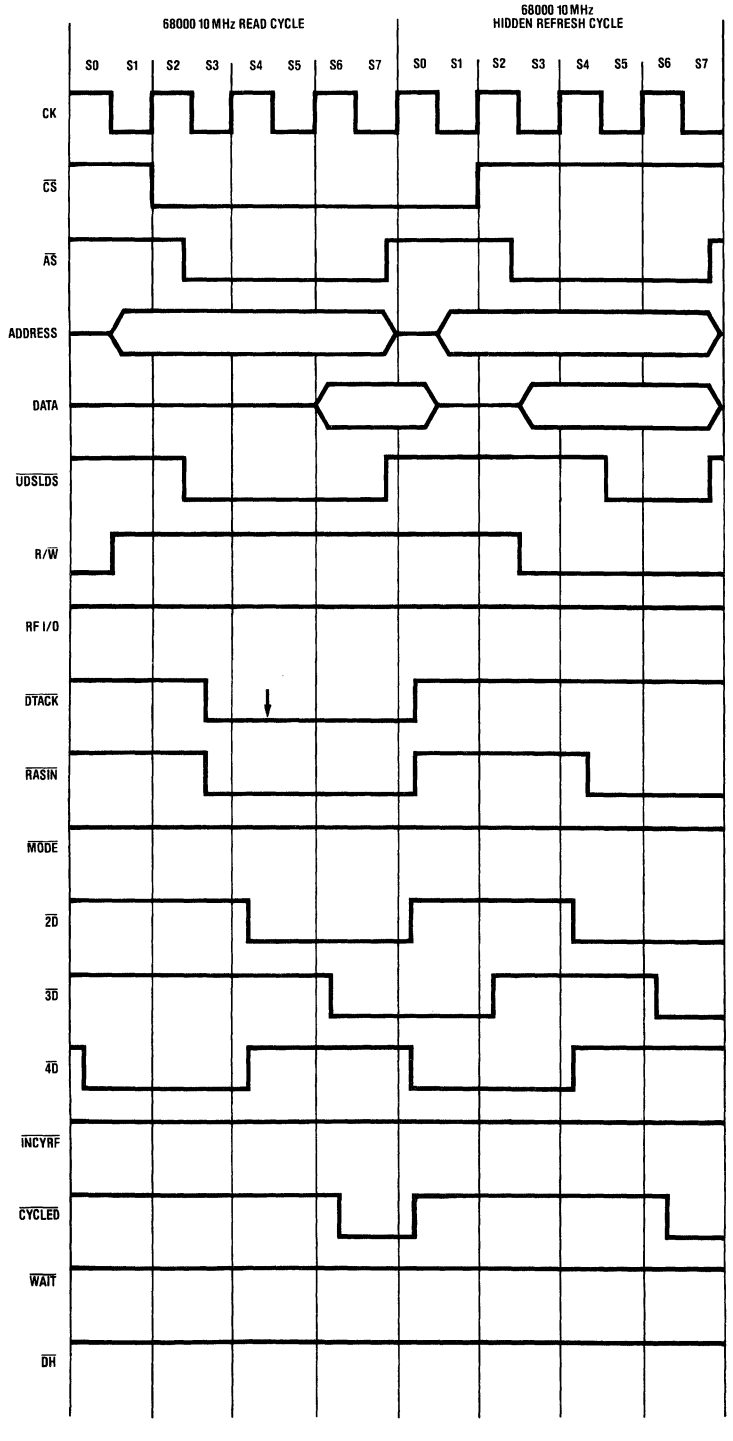
The boolean equations for the DP84422 were written using the standard PALASM™ format. In other words the equation: "IF (VCC) RASIN = INCY *$\overline{\text{MODE}}$*4D*$\overline{\text{R}}$" will mean;

The output "$\overline{\text{RASIN}}$" (see pin list for DP84422) will be active low (inverted $\overline{\text{RASIN}}$) when the output "$\overline{\text{INCY}}$" is low (making INCY high) AND the output "$\overline{\text{MODE}}$" is high AND the output "$\overline{\text{4D}}$" is low (making 4D high) and the input R/$\overline{\text{W}}$ is low (making $\overline{\text{R}}$ high).

```
PAL16R4A ; FAST PAL
NEW PAL FOR THE MOTOROLA 68000 PROCESSOR
(WORKS UP TO 12.5MHZ)

CK /AS RFIO /UDSLDS R /DH CLK /CS /WAIT GND
/OE /INCYRF /CYCLED /4DLY /3DLY /2DLY /MODE /DTACK /RASIN VCC

IF (VCC) RASIN =
 CS*/INCYRF*AS*/MODE*4DLY*/CYCLED*/CLK+          ;Start RASIN
 /CS*/INCYRF*AS*/MODE*2DLY*/CYCLED*/DH+          ;RASIN for Hidden RFSH
 CS*INCYRF*AS*/MODE*4DLY*/CYCLED*/CLK+           ;Start RASIN after RFSH
 CS*RASIN*/MODE*AS+                              ;Hold RASIN valid
 RASIN*/MODE*2DLY                                ;Hold RASIN valid

IF (VCC) CYCLED =/MODE*2DLY*3DLY*/4DLY+          ;Start "CYCLED", does not allow
                 CYCLED*AS+                      ;  glitch after refresh
                 /MODE*CYCLED*/CLK+              ;End on rising edge of CLK
                 /CS*AS*/MODE*/2DLY*/3DLY*/4DLY  ;Start during long accesses of other
                                                 ; devices

IF (VCC) INCYRF =MODE*AS+                        ;Set Access during Refresh
                 INCYRF*4DLY*AS                  ;Hold it while 4DLY is low

IF (CS) DTACK = AS*/WAIT*/R*/MODE*/CLK+          ;0 WAIT's for WRITE
        AS*WAIT*/R*/MODE*2DLY*/CLK+              ;1 WAIT for WRITE
        UDSLDS*/WAIT*R*/MODE*/CLK+               ;0 WAIT's for READ
        UDSLDS*WAIT*R*/MODE*2DLY*/CLK+           ;1 WAIT for READ
        DTACK*2DLY*/MODE+                        ;Continue DTACK
        DTACK*AS*RASIN*/MODE*/CYCLED+            ;Continue DTACK
        DTACK*AS*/R*/MODE                        ;Continue DTACK in RMW
                                                 ;  cycle
MODE : = /RFIO*/AS*/CYCLED*/RASIN+               ;For IDLE states or beginning
                                                 ;  states of 68000 cycle
        /CS*/RFIO*AS*CYCLED*/2DLY*/3DLY*/RASIN+  ;For RFSH during long cycles
                                                 ;  of other devices
        MODE*/3DLY+
        MODE*/4DLY
2DLY : = MODE*/4DLY+
        /INCYRF*AS*/CYCLED*/MODE*/3DLY*4DLY+     ;Start 2DLY
        CS*INCYRF*AS*/CYCLED*/MODE*/3DLY*4DLY+   ;Start 2DLY after RFSH
        /MODE*2DLY*/3DLY+
        CS*WAIT*AS*/MODE*2DLY*3DLY*/4DLY+        ;Make 2DLY longer
        CS*AS*/R*CYCLED*/MODE*/2DLY*/3DLY*/4DLY  ;Start second 2DLY for
                                                 ;the TAS instruction
3DLY : = 2DLY*/4DLY
4DLY : = 3DLY+
         /AS*/MODE+
         /CS*/RFIO*AS*CYCLED*/2DLY*/3DLY*/RASIN*/MODE ;Need for beginning of forced refresh to
                                                      ; inhibit "2DLY"
```

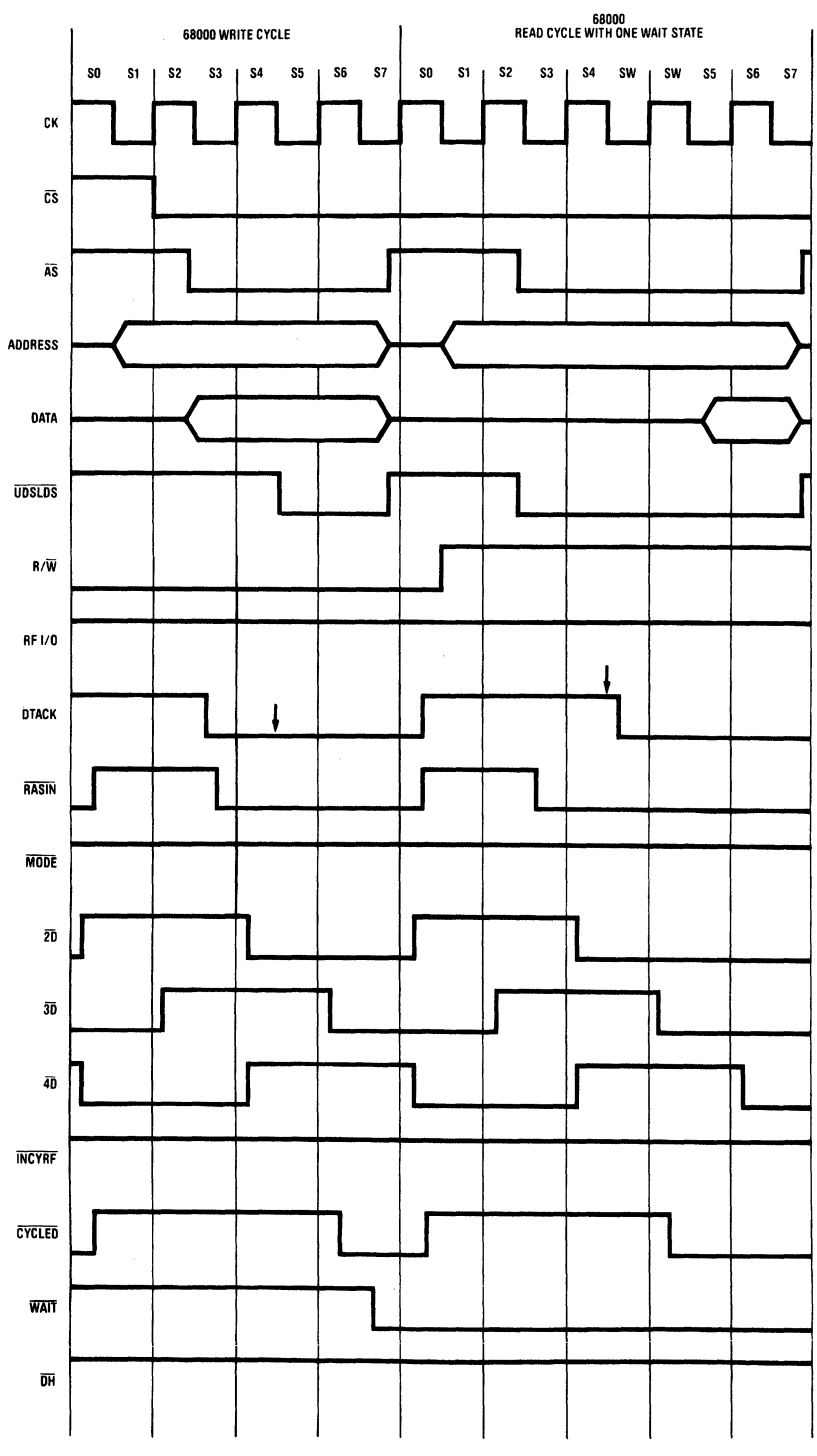
FIGURE 1. Equations for New 68000 PAL That Supports the 68000 "TAS" Instruction

System Timing Diagrams



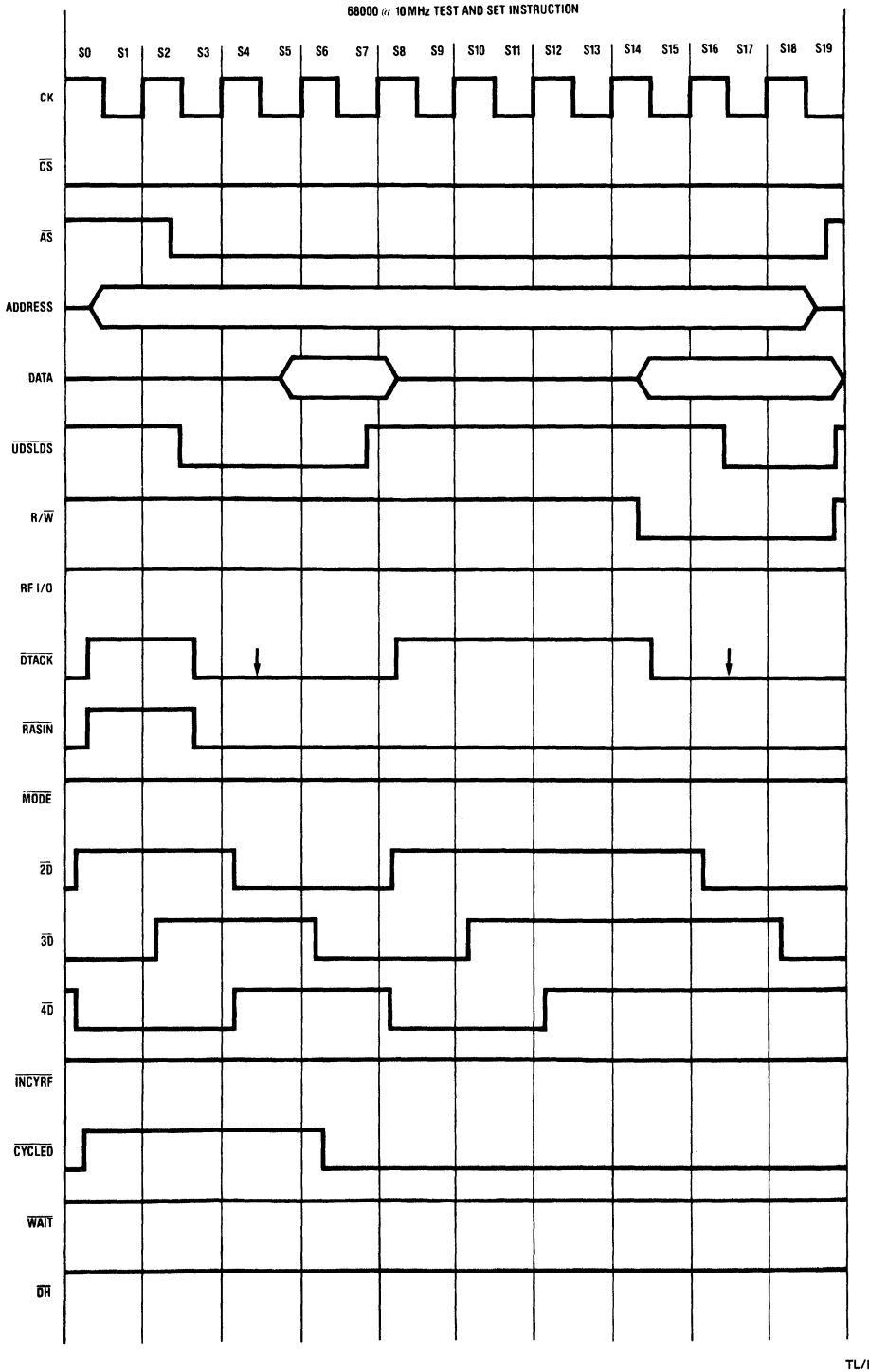
TL/F/8398–3

System Timing Diagrams (Continued)



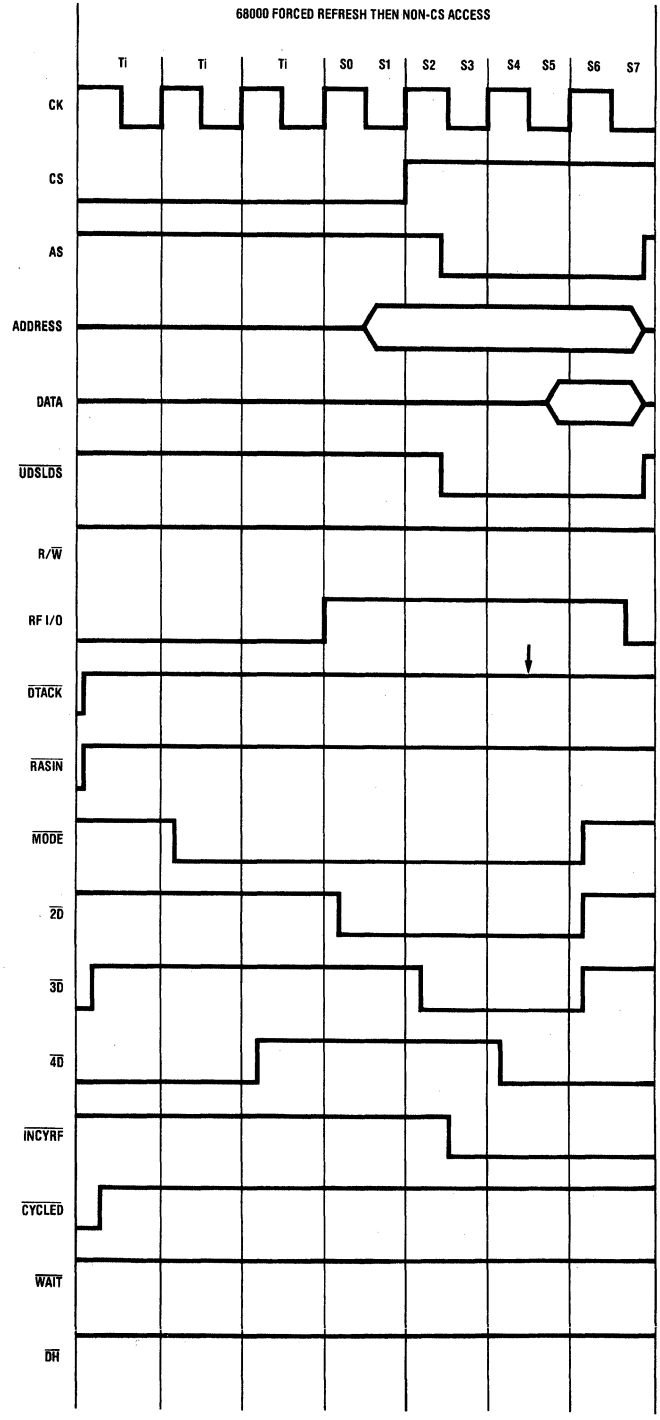
TL/F/8398–4

System Timing Diagrams (Continued)



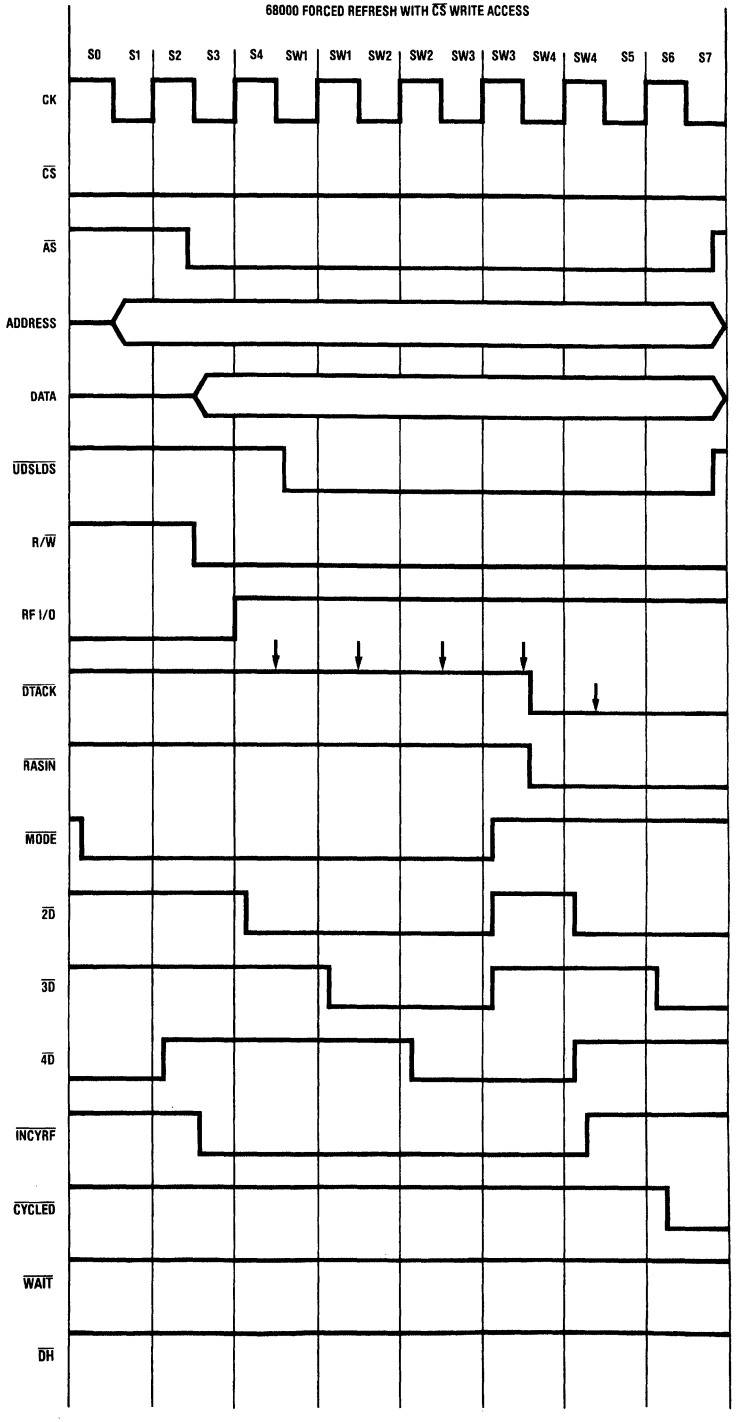
TL/F/8398–5

System Timing Diagrams (Continued)



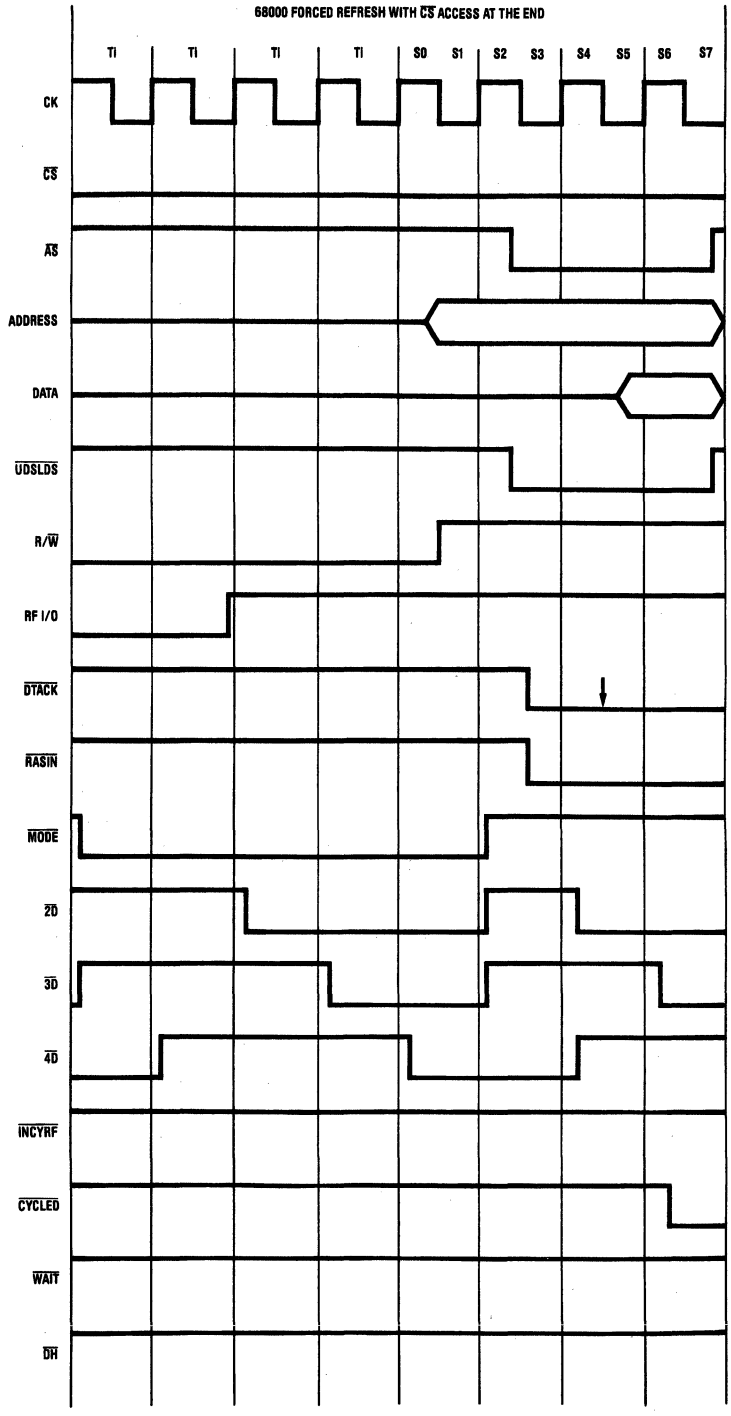
TL/F/8398–6

System Timing Diagrams (Continued)



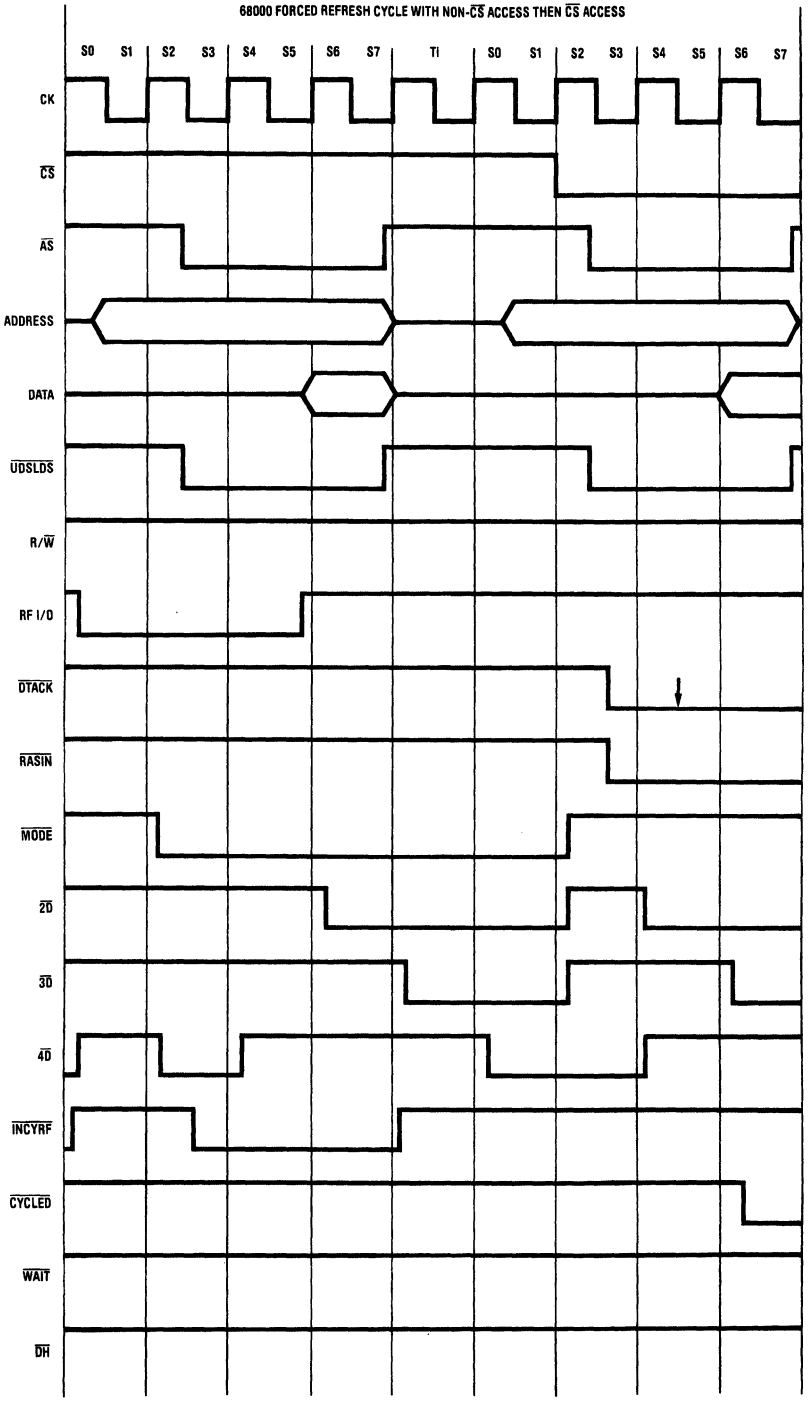
TL/F/8398–7

System Timing Diagrams (Continued)



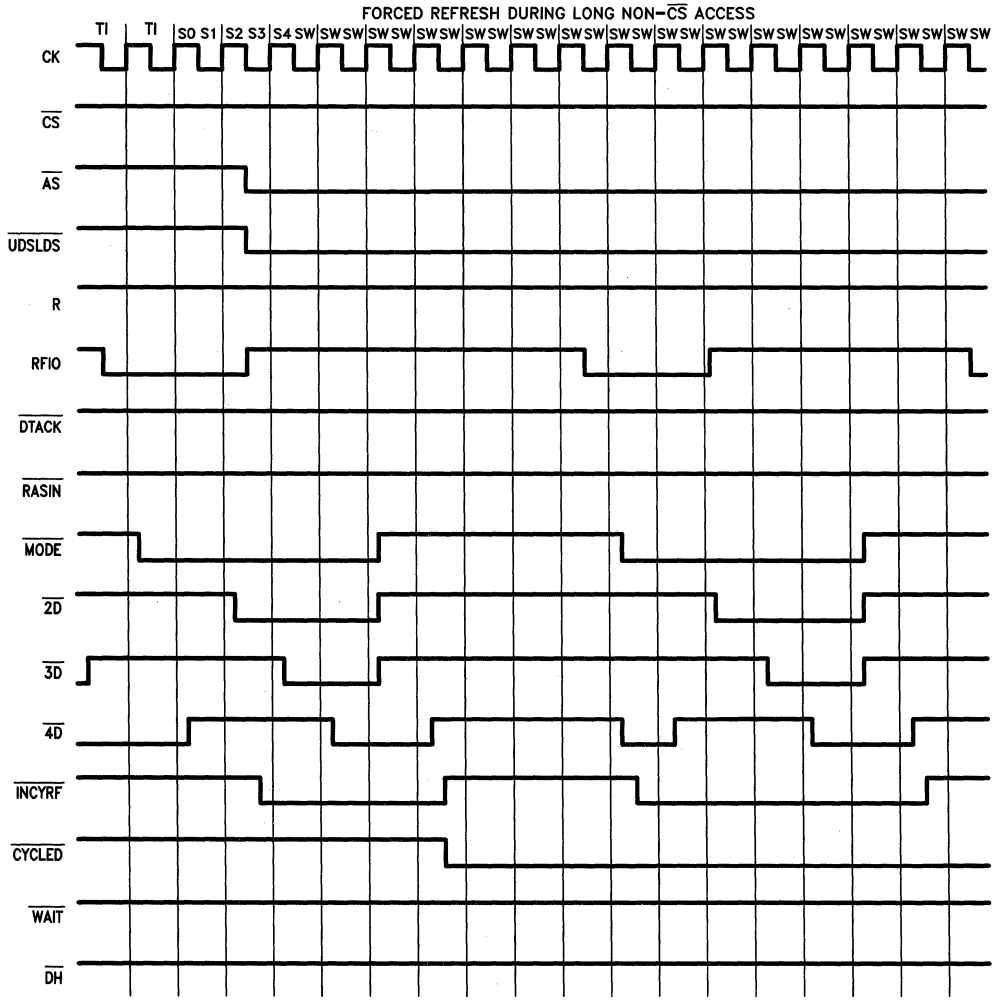
TL/F/8398–8

System Timing Diagrams (Continued)



TL/F/8398–9

System Timing Diagrams (Continued)



TL/F/8398–10

National Semiconductor

PRELIMINARY

DP84432 Dynamic RAM Controller Interface Circuit for the 8086/8088/80186/80188 CPU's

General Description

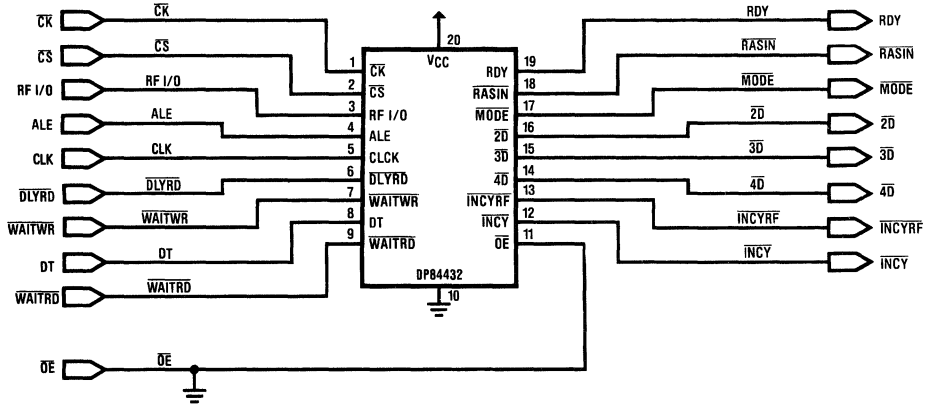
The DP84432 is a new Programmable Array Logic (PAL®) device, that replaces the DP84332, designed to allow an easy interface between the Intel 8088, 8086, 80188, 80186 CPU's and the National Semiconductor DP8409A, DP8429, or DP8419 DRAM controller.

The new DP84432 supplies all the control signals needed to perform memory read, write and refresh and work with the Intel processors up to 10 MHz. Logic is also included to insert WAIT states, if wanted, into the microprocessor READ or WRITE cycles when using fast CPU's.

Features

- Provides a 3-chip solution for the 8086 family, dynamic RAM interface (DP8409A or DP8419, DP84432, and clock divider)
- Works with all 8086 family speed versions up to 10 MHz
- Operation of 8086, 8088, 80186, 80188 at 10 MHz with no WAIT states
- Controls DP8409A or DP8419 Mode 5 accesses, hidden refreshes and Mode 1 Forced Refreshes automatically
- Inserts WAIT states in READ or WRITE cycles automatically depending on whether WAITRD or WAITWR are low, or if $\overline{CS}$ becomes active during a forced Refresh cycle
- Uses a standard National Semiconductor PAL part (DMPAL16R4A)
- The PAL logic equations can be modified by the user for his specific application and programmed into any of the PALs in the National Semiconductor family, including the new very high speed PALs ("B" PAL parts)

Connection Diagram



TL/F/8399–1

Order Number DP84432N or DP84432J
See NS Package Number N20A or J20A

3

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

	Operating	Programming
Supply Voltage, V_{CC}	7V	12V
Input Voltage	5.5V	12V
Off-State Output Voltage	5.5V	12V
Storage Temperature Range	−65°C to +150°C	

DP84432 Recommended Operating Conditions

Symbol	Parameter		Commercial			Units
			Min	Typ	Max	
V_{CC}	Supply Voltage		4.75	5	5.25	V
t_w	Width of Clock	Low	15	10		ns
		High	15	10		
t_{su}	Setup Time from Input or Feedback to Clock		25	16		ns
t_h	Hold Time		0	−10		ns
T_A	Operating Free-Air Temperature		0	25	75	°C
T_C	Operating Case Temperature					°C

Electrical Characteristics Over Recommended Operating Temperature Range

Symbol	Parameter	Test Conditions		Min	Typ	Max	Units
V_{IH}	High Level Input Voltage			2			V
V_{IL}	Low Level Input Voltage					0.8	V
V_{IC}	Input Clamp Voltage	V_{CC} = Min., I_I = −18 mA			−0.8	−1.5	V
V_{OH}	High Level Output Voltage	V_{CC} = Min. V_{IL} = 0.8V V_{IH} = 2V	I_{OH} = −3.2 mA COM	2.4	2.8		V
V_{OL}	Low Level Output Voltage	V_{CC} = Min. V_{IL} = 0.8V V_{IH} = 2V	I_{OL} = −24 mA COM		0.3	0.5	V
I_{OZH}	Off-State Output Current	V_{CC} = Max V_{IL} = 0.8V V_{IH} = 2V	V_O = 2.4V			100	μA
I_{OZL}			V_O = 0.4V			−100	μA
I_I	Maximum Input Current	V_{CC} = Max., V_I = 5.5V				1	mA
I_{IH}	High Level Input Current	V_{CC} = Max., V_I = 2.4V				25	μA
I_{IL}	Low Level Input Current	V_{CC} = Max., V_I = 0.4V			−0.02	−0.25	mA
I_{OS}	Output Short-Circuit Current	V_{CC} = 5V V_O = 0V		−30	−70	−130	mA
I_{CC}	Supply Current†	V_{CC} = Max.			120	180	mA

Switching Characteristics Over Recommended Ranges of Temperature and V_{CC}

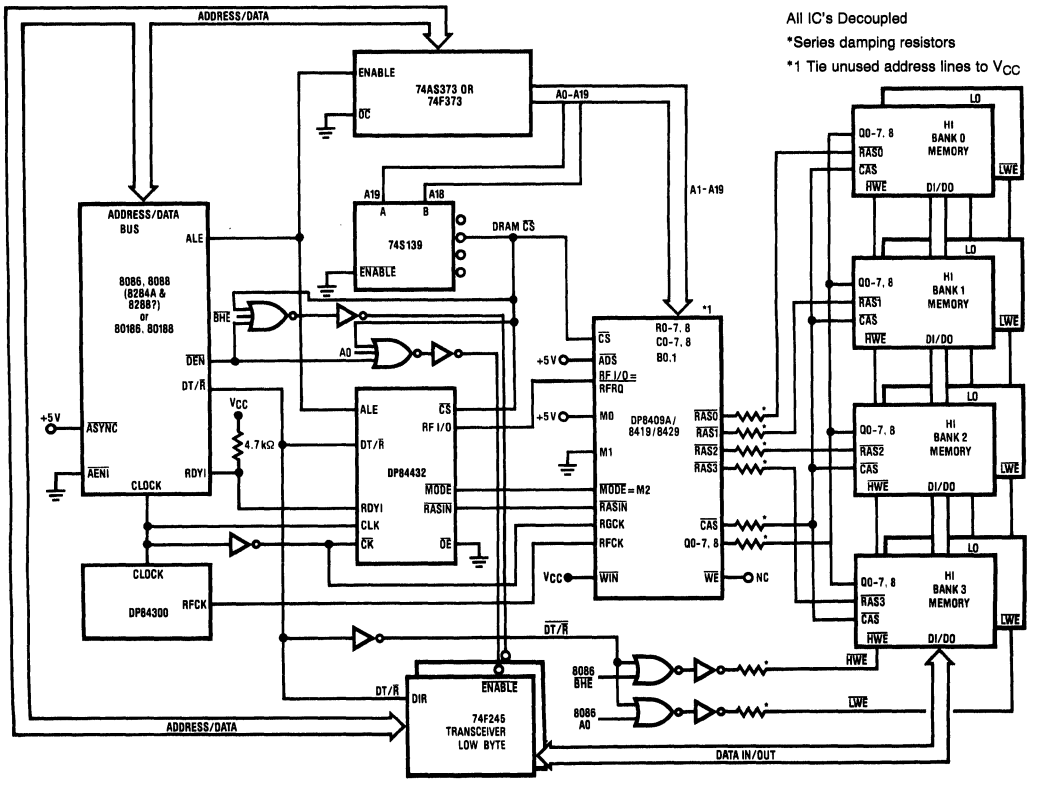
V_{CC} = 5V ± 10%. Commercial: T_A = 0 to 75°C, V_{CC} = 5V ± 5%

Symbol	Parameter	Test Conditions R1,R2	Commercial			Units
			Min	Typ	Max	
t_{PD}	Input or Feedback to Output	CL = 50 pF		15	25	ns
t_{CLK}	Clock to Output or Feedback			10	15	ns
t_{PZX}	Pin 11 to Output Enable			10	20	ns
t_{PXZ}	Pin 11 to Output Disable	C_L = 5 pF		11	20	ns
t_{PZX}	Input to Output Enable	C_L = 50 pF		10	25	ns
t_{PXZ}	Input to Output Disable	C_L = 5 pF		13	25	ns
f_{MAX}	Maximum Frequency		25	30		ns

V_{CC} = Max. at minimum temperature.

Block Diagram

8086 System Block Diagram



TL/F/8399–2

Mnemonic Description

INPUT SIGNALS

1) "$\overline{\text{CLOCK}}$" Inverted clock from 8284A or 8288. "$\overline{\text{CLOCK}}$" should be delayed from CLOCK (pin 5).

2) "$\overline{\text{CS}}$" From decoder chip (chip select) (active low).

3) "ALE" From 8086 (active high).

4) "RFI/O" $\overline{\text{RFRQ}}$ (refresh request) in mode 5. From 8409A, an active low signal.

5) "CLOCK" The non-inverted clock directly from the 8284A. This signal should be unbuffered to this input so as not to incur any extra delay in the $\overline{\text{RASIN}}$ generation time.

6) "$\overline{\text{DELAYREAD}}$" This input signal allows the user to delay when the $\overline{\text{RASIN}}$ signal becomes valid to the DP8409A during a READ cycle of the 8086. This input should be low when using the DP8409A unless an external delay line is used to guarantee a 30 ns $\overline{\text{CS}}$ to $\overline{\text{RASIN}}$ delay (for DP8409A or 15 ns for DP8419) or if the user can afford to disable the hidden refresh by permanently tying $\overline{\text{CS}}$ low on the DP8409A.

7) "$\overline{\text{WAITWRITE}}$" This signal is used to delay when $\overline{\text{RASIN}}$ becomes valid during an 8086 WRITE cycle and also adds a WAIT state into a $\overline{\text{CS}}$ WRITE access cycle. One may want to delay when $\overline{\text{RASIN}}$ becomes valid during a WRITE cycle when generating a parity bit for each byte. This would allow time to generate parity and be assured that the data and parity bit were both written to memory.

8) "DT/$\overline{\text{R}}$" or "$\overline{\text{S1}}$" Used to differentiate between READ and WRITE cycles, and to allow $\overline{\text{CS}}$ READ cycles to start early. If the system is not a minimum mode 8086 or 8088 system then the status signal "$\overline{\text{S1}}$" should be used instead of "DT/R" so that the DP84432 knows immediately whether the CPU is doing a READ or a WRITE access cycle.

Mnemonic Description (Continued)

9) "$\overline{\text{WAITREAD}}$" Used to insert 1 wait state into the 8086 $\overline{\text{READ}}$ bus cycle. The wait state following bus cycle "T3" allows the use of memory with longer access times (t_{CAC}). An active low signal.

10) "$\overline{\text{OE}}$" This input enables the outputs of the "D-Flip Flop" outputs of the PAL.

OUTPUTS SIGNALS

1) "$\overline{\text{MODE}}$" This pin goes to M2 on the DP8409A to change from mode 5 to mode 1 (only used for forced refresh).

2) "$\overline{\text{2DLY}}$" Delay used internal to the PAL.

3) "$\overline{\text{3DLY}}$" Delay used internal to the PAL.

4) "$\overline{\text{4DLY}}$" Delay used internal to the PAL.

5) "$\overline{\text{RASIN}}$" To the 8409A (creates $\overline{\text{RAS}}$'s).

6) "RDY1" To the 8284A or 8288 to insert wait states into the 8086 bus cycles (active low).

7) "$\overline{\text{INCYCLE}}$ $\overline{\text{REFRESH}}$" This signal is used in the *Figure 1* PAL to detect that an access cycle was started during a DRAM refresh cycle. This allows the PAL to determine, later in the cycle, whether to restart the "INCYCLE" signal or not. If the CPU is not accessing the DRAM, as determined by "$\overline{\text{CS}}$" being low, then "INCYCLE" is not restarted.

8) "$\overline{\text{INCYCLE}}$" This signal goes active from the CPU ALE signal. This signal indicates that the processor is doing an access somewhere in the system. This signal stays low for several T states of the access cycle.

Functional Description

The following description applies to both the DP8409A and the DP8419 dynamic RAM controllers.

A memory cycle starts when chip select ($\overline{\text{CS}}$) and address latch enable (ALE) are true. $\overline{\text{RASIN}}$ is supplied from the DP84432 to the DP8409A dynamic RAM controller, which then supplies a $\overline{\text{RAS}}$ signal to the selected dynamic RAM bank. After the necessary row address hold time, the DP8409A switches the address outputs to the column address. The DP8409A then supplies the required $\overline{\text{CAS}}$ signal to the DRAM. In order to do byte operations it is suggested that the user provide external logic, as shown in the system block diagram, to produce a HIGH WRITE ENABLE or a LOW WRITE ENABLE. To differentiate between a READ and a WRITE, the DT/$\overline{\text{R}}$ (or status signal "$\overline{\text{S1}}$" in a maximum mode 8086 or 8088 system or in a 80186, 8188 system) signal from the CPU is inverted and also supplied to the external WRITE ENABLE logic.

A refresh cycle is started by one of two conditions. The refresh cycle caused by the first condition is called a hidden refresh. This occurs when refresh clock (RFCK) is high, $\overline{\text{CS}}$ is not true, and $\overline{\text{RASIN}}$ goes true. Here the CPU is accessing something else in the system and the DRAM can be refreshed at that time, thereby being transparent to the CPU. The second type of refresh is called forced refresh. This occurs if no hidden refresh was performed while RFCK was high. When RFCK transitions low a refresh request ($\overline{\text{RFRQ}}$) is generated. If there is not a DRAM access in progress the DP84432 will force a refresh by putting the DP8409A into mode 1 (automatic forced refresh mode). If the CPU tries to access the DRAM during a forced refresh cycle WAIT states will be inserted into its cycles until the forced refresh is over and the DRAM $\overline{\text{RAS}}$ precharge time has been met. Then the pending DRAM access will be allowed to take place.

The DP84432 also allows forced refreshes to take place during long accesses of other devices. For instance, if EEPROM takes several microseconds to write to, the DRAM will still be refreshed while that access is in progress.

In a standard memory cycle, the access can be slowed down by one clock cycle to accommodate slower memories or allow time to generate parity. This is accomplished by inserting a WAIT state into the processor access cycle. The DP84432 can insert WAIT states into either READ or WRITE cycles, or both. The extra WAIT state will not appear during the hidden refresh cycle, so faster devices on the CPU bus will not be affected.

SYSTEM INTERFACE DESCRIPTION

The 80186 or 80188 will be able to use the DP84432 but it will be necessary to invert "ALE" of the 80186 or 80188 and logically NOR it with the "CLOCK" signal. This fix makes the 80186 or 80188 "ALE" signal appear to be similar to the 8086 or 8088 "ALE" signal. The 8088 will be able to use this PAL, but the 8088 will not need the logic necessary to produce $\overline{\text{LWE}}$, $\overline{\text{HWE}}$. The 80286 can not use this PAL because it's WAIT state logic is different. (See DP84532 data sheet).

The DP84432 differentiates between READ and WRITE cycles, allowing the $\overline{\text{RASIN}}$ signal to start earlier during a READ cycle compared to a WRITE cycle.

$\overline{\text{RASIN}}$ during a READ cycle can start during T1 or T2 of a processor cycle depending on whether the DELAYREAD input is set low or high. If DELAYREAD is false the user will need to use an external delay line to guarantee that CS will be valid a minimum of 30 ns before $\overline{\text{RASIN}}$ becomes true. If the user is willing to give up hidden refreshes ($\overline{\text{CS}}$ tied permanently low on DP8409A) he must only guarantee that the addresses are valid at the inputs of the DP8409A by a minimum of 10 ns before $\overline{\text{RASIN}}$ becomes valid.

This section of the data sheet goes through the calculation of the "tRAC" ($\overline{\text{RAS}}$ access time) and "tCAC" ($\overline{\text{CAS}}$ access time) required by the DRAM for the iAPX 86/88/186/188 family CPUs to operate at a particular clock frequency without introducing wait states into the processor access cycles. Both "tRAC" and "tCAC" must be considered in determining what speed DRAM can be used in a particular system design. The DRAM chosen must meet both the "tRAC" and "tCAC" parameters calculated. In order to determine the "tRAC" and "tCAC" needed the DP8419 and fast PALs ("B" type PALs) timing parameters were used. If the user is using the DP8408A/09A or a slower PAL device he should substitute their respective delays into the equations below.

Most all of the calculations contained in this note use "RAHS" = 1 (15 ns guaranteed minimum row address hold time). Calculations only used "RAHS" = 0 (25 ns guaranteed minimum row address hold time) when the calculated access time from RAS exceeded 200 ns. This is because DRAMs can be found with row access times up to 150 ns that require only 15 ns row address hold times.

Functional Description (Continued)

EXAMPLE DRAM TIMING CALCULATIONS

A) iAPX 86/88 8 MHz, No WAIT states, "/DLYRD" = low

#1) $\overline{RASIN}$ low = 1 system clock period + 15 ns ("B" PAL combinational output delay) = 125 + 15 = 140 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 80 ns (DP8419 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMs speced in data sheet) = 77 ns maximum (using 15 ns minimum row address hold time)

#4) 74F245 transceiver delay = 7 ns maximum

#5) CPU data setup time to "T4" = 20 ns minimum

"t_{RAC}" = T1 + T2 + T3 − #1 − #2 − #4 − #5
= 125 + 125 + 125 − 140 − 20 − 7 − 20 = 188 ns

"t_{CAC}" = T1 + T2 + T3 − #1 − #3 − #4 − #5
= 125 + 125 + 125 − 140 − 77 − 7 − 20 = 131 ns

Therefore the DRAM chosen should have a "t_{RAC}" less then or equal to 188 ns and a "t_{CAC}" less then or equal to 131 ns. Standard 150 ns DRAMs meet this criteria.

The minimum $\overline{RAS}$ PRECHARGE TIME will be approximately two clock periods = 125 + 125 = 250 ns.

The minimum $\overline{CAS}$ PRECHARGE TIME will be approximately two clock periods plus 50 ns (minimum t_{RICL}-t_{RICH} for the DP8409-2) = 125 + 125 + 50 = 300 ns.

The minimum $\overline{RAS}$ PULSE WIDTH will be approximately two clock periods − 5 ns (t_{RPDL}-t_{RPDH} for the DP8409-2) = 125 + 125 − 5 = 245 ns.

The minimum $\overline{CAS}$ PULSE WIDTH will be approximately two clock periods − 70 ns (maximum t_{RICL}-t_{RICH} for the DP8409-2) = 125 + 125 − 70 = 180 ns.

The above times are assuming the use of the DP8409-2 and a fast ("A" part) PAL. The smallest pulse widths are generated during WRITE cycles since $\overline{RASIN}$ during WRITE cycles starts later than $\overline{RASIN}$ during READ cycles.

B) 80186, 8 MHz, "$\overline{DLYRD}$" = HIGH, No Hidden Refresh ($\overline{CS}$ = Low), No Wait States

Minimum $\overline{RASIN}$ = 55 ns (min clk low) + 1 ns (min PAL delay) = 68 ns

Maximum Address Valid = 44 ns (ADD valid max) + 8 ns (74F373) = 52 ns

#1) $\overline{RASIN}$ low = Maximum clock high + 15 ns ("B" PAL combinational output delay) = 70 + 15 = 85 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 97 ns (DP8419 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMs speced in data sheet) = 94 ns maximum (using 25 ns minimum row address hold time)

#4) 74F245 Transceiver delay = 7 ns maximum

#5) CPU data setup time to "T4" = 20 ns minimum

"t_{RAC}" = T1 + T2 + T3 − #1 − #2 − #4 − #5
= 125 + 125 + 125 − 85 − 20 − 7 − 20 = 243 ns

"t_{CAC}" = T1 + T2 + T3 − #1 − #3 − #4 − #5
= 125 + 125 + 125 − 85 − 94 − 7 − 20 = 169 ns

Therefore the DRAM chosen should have a "t_{RAC}" less then or equal to 243 ns and a "t_{CAC}" less then or equal to 169 ns. Standard 200 ns DRAMs meet this criteria.

The minimum $\overline{RAS}$ PRECHARGE TIME will be approximately one clock period + 55 ns (minimum clock low) − 15 ns (maximum DP84432 clocked output delay for ending $\overline{RASIN}$) = 125 + 55 − 15 = 165 ns.

The minimum $\overline{CAS}$ PRECHARGE TIME will be approximately one clock period + 55 ns (minimum clock low) − 15 ns (maximum clocked output delay for ending $\overline{RASIN}$) + 35 ns (minimum t_{RICL}-t_{RICH} for the DP8409-2) = 125 + 55 − 15 + 35 = 200 ns.

The minimum $\overline{RAS}$ PULSE WIDTH will be approximately two clock periods − 5 ns (t_{RPDL}-t_{RPDH} for the DP8409-2) = 125 + 125 − 5 = 245 ns.

The minimum $\overline{CAS}$ PULSE WIDTH will be approximately two clock periods − 70 ns (maximum t_{RICL}-t_{RICH} for the DP8409-2) = 125 + 125 − 70 = 180 ns.

C) 8086, 8 MHz, $\overline{CS}$ Tied Low (no hidden refresh), $\overline{DLYRD}$ = HIGH, No Delay Line Needed, No Wait States

Minimum $\overline{RASIN}$ = 69 ns (min clk low) + 13 ns (min PAL delay) = 82 ns

Maximum Address Valid = 60 ns (ADD valid max) + 8 ns (74F373) = 68 ns

The address must be valid a minimum of 10 ns before $\overline{RASIN}$ goes valid at the inputs of the DP8409A or DP8419, which it will be given the ICs used in this example.

#1) $\overline{RASIN}$ low = Maximum clock high + 15 ns ("B" PAL combinational output delay) = 82 + 15 = 97 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 97 ns (DP8419 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMs speced in data sheet) = 94 ns maximum (using 25 ns minimum row address hold time)

#4) 74F245 Transceiver delay = 7 ns maximum

#5) CPU data setup time to "T4" = 20 ns minimum

"t_{RAC}" = T1 + T2 + T3 − #1 − #2 − #4 − #5
= 125 + 125 + 125 − 97 − 20 − 7 − 20 = 231 ns

"t_{CAC}" = T1 + T2 + T3 − #1 − #3 − #4 − #5
= 125 + 125 + 125 − 97 − 94 − 7 − 20 = 157 ns

Therefore the DRAM chosen should have a "t_{RAC}" less then or equal to 231 ns and a "t_{CAC}" less then or equal to 157 ns. Standard 200 ns DRAMs meet this criteria.

The minimum $\overline{RAS}$ PRECHARGE TIME will be approximately one clock period + 69 ns (minimum clock low) + 15 ns (maximum DP84432 clocked output delay for ending $\overline{RASIN}$) = 125 + 69 − 15 = 179 ns.

The minimum $\overline{CAS}$ PRECHARGE TIME will be approximately one clock period + 69 ns (minimum clock low) − 15 ns (maximum clocked output delay for ending $\overline{RASIN}$) + 35 ns (minimum t_{RICL}-t_{RICH} for the DP8409-2) = 125 + 69 − 15 + 35 = 214 ns.

The minimum $\overline{RAS}$ PULSE WIDTH will be approximately two clock periods − 5 ns (t_{RPDL}-t_{RPDH} for the DP8409-2) = 125 + 125 − 5 = 245 ns.

The minimum $\overline{CAS}$ PULSE WIDTH will be approximately two clock periods − 70 ns (maximum t_{RICL}-t_{RICH} for the DP8409-2) = 125 + 125 − 70 = 180 ns.

Functional Description (Continued)

D) 8086, 10 MHz, $\overline{CS}$ Tied Low (no hidden refresh), $\overline{DLYRD}$ = HIGH, No Delay Line Needed, No Wait States

MINIMUM $\overline{RASIN}$ = 52 ns (min clk low) + 13 ns (min PAL delay) = 65 ns

MAXIMUM ADDRESS VALID = 50 ns (ADD valid max) + 8 ns (74F373) = 58 ns

The address must be valid a minimum of 10 ns before $\overline{RASIN}$ goes valid at the inputs of the DP8409A or DP8419. As an example use two 74ALS04 inverters to guarantee a minimum delay of 4 ns, therefore MINIMUM $\overline{RASIN}$ = 69 ns

#1) $\overline{RASIN}$ low = Maximum clock high + 15 ns ("B" PAL combinational output delay) = 61 + 15 = 76 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 80 ns (DP8419 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMs speced in data sheet) = 77 ns maximum (using 15 ns minimum row address hold time)

#4) 74F245 Transceiver delay = 7 ns maximum

#5) CPU data setup time to "T4" = 5 ns minimum

"t_{RAC}" = T1 + T2 + T3 − #1 − #2 − #4 − #5

= 100 + 100 + 100 − 76 − 20 − 7 − 5 = 192 ns

"t_{CAC}" = T1 + T2 + T3 − #1 − #3 − #4 − #5

= 100 + 100 + 100 − 76 − 77 − 7 − 5 = 135 ns

Therefore the DRAM chosen should have a "t_{RAC}" less then or equal to 192 ns and a "t_{CAC}" less then or equal to 135 ns. Standard 150 ns DRAMs meet this criteria.

The minimum $\overline{RAS}$ PRECHARGE TIME will be approximately one clock period + 69 ns (minimum clock low) − 15 ns (maximum DP84432 clocked output delay for ending $\overline{RASIN}$) = 125 + 69 − 15 = 179 ns.

The minimum $\overline{CAS}$ PRECHARGE TIME will be approximately one clock period + 69 ns (minimum clock low) − 15 ns (maximum clocked output delay for ending $\overline{RASIN}$) + 35 ns (minimum t_{RICL}-t_{RICH} for the DP8409-2) = 125 + 69 − 15 + 35 = 214 ns.

The minimum $\overline{RAS}$ PULSE WIDTH will be approximately two clock periods − 5 ns (t_{RPDL}-t_{RPDH} for the DP8409-2) = 125 + 125 − 5 = 245 ns.

The minimum $\overline{CAS}$ PULSE WIDTH will be approximately two clock periods − 70 ns (maximum t_{RICL}-t_{RICH} for the DP8409-2) = 125 + 125 − 70 = 180 ns.

SUGGESTIONS

It is suggested that the DP8409A is to be used up to 8 MHz. Above 8 MHz one should use the DP8409-2 or the DP8419. Also fast PALs ("A" parts) should be used at 8 MHz and above. If very fast PALs are used ("B" parts) the access will be 10 ns faster than calculated in the above sections.

These suggestions occur because of DRAM parameters that must be met, such as:

1) $\overline{CAS}$ ACCESS TIME—time from $\overline{CAS}$ valid until data is available at the DRAM outputs.

2) $\overline{RAS}$ PRECHARGE TIME—minimum amount of time from $\overline{RAS}$ high until $\overline{RAS}$ transitions low again.

3) $\overline{CAS}$ PRECHARGE TIME—minimum amount of time from $\overline{CAS}$ high until $\overline{CAS}$ transitions low again.

4) $\overline{RAS}$ PULSE WIDTH—minimum $\overline{RAS}$ valid time during an access, this usually occurs during a WRITE operation since $\overline{RASIN}$ is generated later then in a READ operation.

5) $\overline{CAS}$ PULSE WIDTH—minimum $\overline{CAS}$ valid time during an access.

6) DATA IN SETUP TIME—the data, during a DRAM WRITE access cycle, must be valid at the DRAM inputs when WRITE ENABLE or $\overline{CAS}$ transitions low, whichever occurs last.

For instance, during a WRITE operation, one does not want $\overline{CAS}$ to go valid until the data to be written is setup at the inputs to the dynamic RAM. Therefore an 8086 running at 5 MHz should use a DP8409A and a slower DP84432 PAL.

EXAMPLE: 8086, 5 MHz, DP8409A, DP84432 (fast PAL "A" part)

MINIMUM $\overline{RASIN}$ = 3 ns (min clk inversion) + 7 ns (min fast PAL clocked output) + 13 ns (min combinational fast PAL output) = 23 ns into the T2 CPU cycle.

MINIMUM $\overline{CAS}$ = MINIMUM $\overline{RASIN}$ + MINIMUM $\overline{RASIN}$ TO $\overline{CAS}$ TIME = 23 + 95 = 118 ns

MINIMUM DATA VALID during an 8086 WRITE at 5 MHz = 110 ns

MINIMUM DATA VALID at DRAM input = MINIMUM DATA VALID + MINIMUM TRANSCEIVER DELAY (74F245) = 110 + 7 = 117 ns

Therefore, worst case, the data could be valid 1 ns before $\overline{CAS}$ becomes valid at the DRAM inputs. Most DRAMs specify 0 ns setup time so this is OK, but if the DP8409A is driving less then the full load specified in the data sheet $\overline{CAS}$ could become valid before the data was available at the DRAM inputs. Therefore the user may want to use a slower PAL or adjust the PAL equations to start the WRITE later in the access cycle. For example, the second equation in the $\overline{RASIN}$ term could be adjusted as follows to accomplish a later $\overline{RASIN}$ during WRITE cycles:

change "$\overline{CS}$*$\overline{INCY}$*$\overline{MODE}$*2D*$\overline{WAITWR}$" to

"$\overline{CS}$*$\overline{INCY}$*$\overline{MODE}$*2D*$\overline{WAITWR}$*$\overline{CLK}$"

At higher frequencies one generally wants to generate WRITE as the DP84432 does in order to guarantee that the $\overline{CAS}$ pulse width is great enough.

INTERPRETING THE DP84432 PAL EQUATIONS

The boolean equations for the DP84432 were written using the standard PALASM™ format. In other words the equation:

"IF (VCC) RASIN = INCY*$\overline{MODE}$*4D*$\overline{DT}$" will mean;

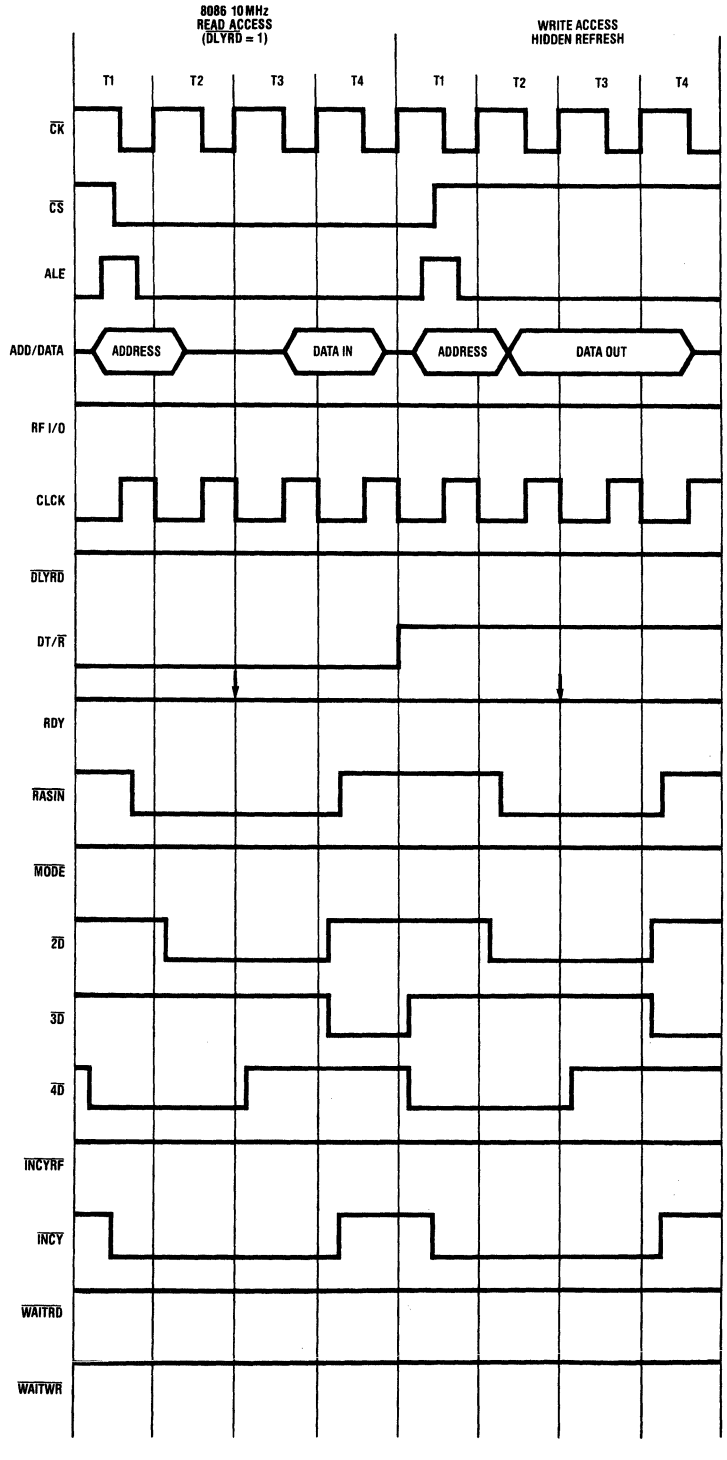
The output "$\overline{RASIN}$" (see pin list for DP84432) will be active low (inverted RASIN) when the output "$\overline{INCY}$" is low (making INCY high) AND the output "$\overline{MODE}$" is high AND the output "$\overline{4D}$" is low (making 4D high) AND the input DT/R is low (making $\overline{DT/R}$ high).

PAL Boolean Equations

```
    PAL16R4A     ;FAST PAL
    NEW PAL FOR INTEL PROCESSORS 8086, 8088, 80186, 80188
    NATIONAL SEMICONDUCTOR (WORKS UP TO 10 MHz)

    /CK /CS RF10 ALE CLK /DLYRD /WAITWR DT /WAITRD GND /OE
    /INCY /INCYRF /4D /3D /2D /MODE /RASIN RDY VCC

IF (VCC) RASIN =
  INCY*/MODE*4D*/DT*/DLYRD*CLK+              ;Start RASIN, early READ
  CS*INCY*/MODE*2D*/WAITWR+                  ;Start RASIN, early WRITE
  CS*INCY*/INCYRF*/ALE*/MODE*/3D*/DT*/CLK+   ;Start READ
  CS*INCY*/MODE*2D*DT*WAITWR*CLK+            ;Late WRITE
  /CS*INCY*/MODE*2D+                         ;Hidden RFSH
  RASIN*INCY*/ALE*/MODE*/3D*4D+              ;Continue RASIN
  RASIN*/MODE*2D                             ;Continue RASIN

IF (VCC) INCYRF = ALE*MODE+                  ;Start INCYCLE in REFRESH
                  INCYRF*MODE+               ;Continue INCY in REFRESH
                  INCYRF*4D*/CLK             ;Continue INCY in REFRESH

MODE := /RF10*/INCY*/2D+                     ;Forced RFSH at beginning
                                             ; of a cycle, during IDLE
        MODE*/3D+                            ; states, or during long
        MODE*/4D                             ; accesses of other devices

2D := MODE*/4D+
  INCY*/MODE*4D+
  CS*/DT*WAITRD*INCY*/MODE*2D*/3D*/4D+       ;Extend for "CS READ" cycle
  CS*DT*WAITWR*INCY*/MODE*2D*/3D*/4D         ;Extend for "CS WRITE" cycle

3D := 2D*/4D

4D := 3D+
      /INCY*/MODE+
      INCY*/MODE*/2D

IF (VCC) INCY = ALE*/MODE+                   ;Start INCY for access
         INCY*/INCYRF*/MODE*/3D*4D+          ;Continue INCY during access
         INCY*/MODE*2D+                      ;End INCY during access
         CS*INCYRF*/MODE*/2D*/3D*4D+         ;Start INCY after REFRESH
         CS*INCY*/MODE*/3D*4D*/RDY           ;Continue INCY after REFRESH

IF (CS) /RDY = CS*INCYRF*/2D*/3D*4D+         ;Access at end of RFSH cycle
        CS*/RDY*/MODE*/2D*/3D*4D+            ;Continue RDY after RFSH
        CS*MODE+                             ;Continue RDY after RFSH
        CS*/DT*WAITRD*INCY*/MODE*2D*/3D*4D+  ;WAIT for "CS READ"
        CS*DT*WAITWR*INCY*/MODE*2D*/3D*4D    ;WAIT for "CS WRITE"
```

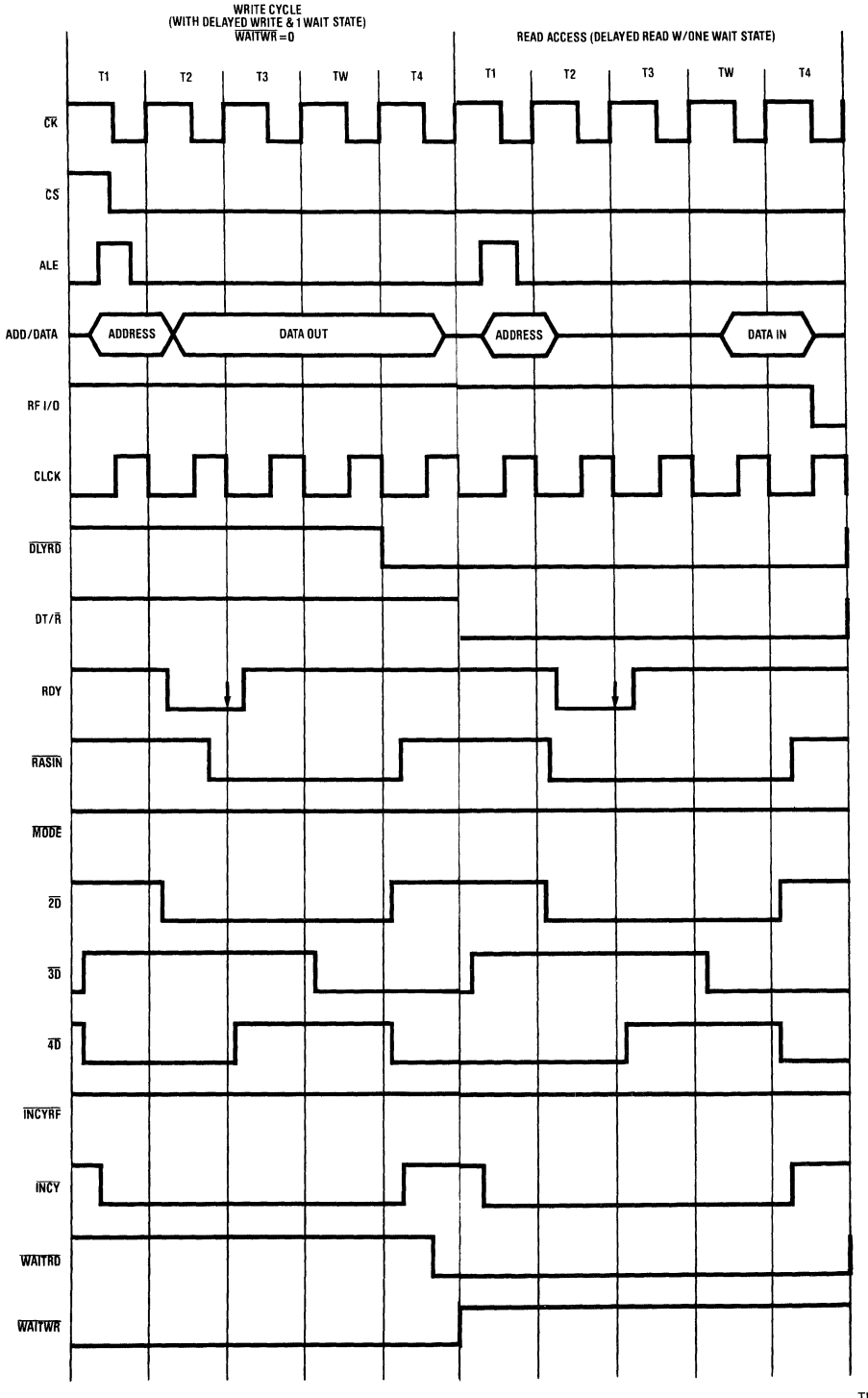
FIGURE 1. Equations for the DP84432 Standard Interface PAL (Works in Minimum or Maximum Mode)

System Timing Diagrams



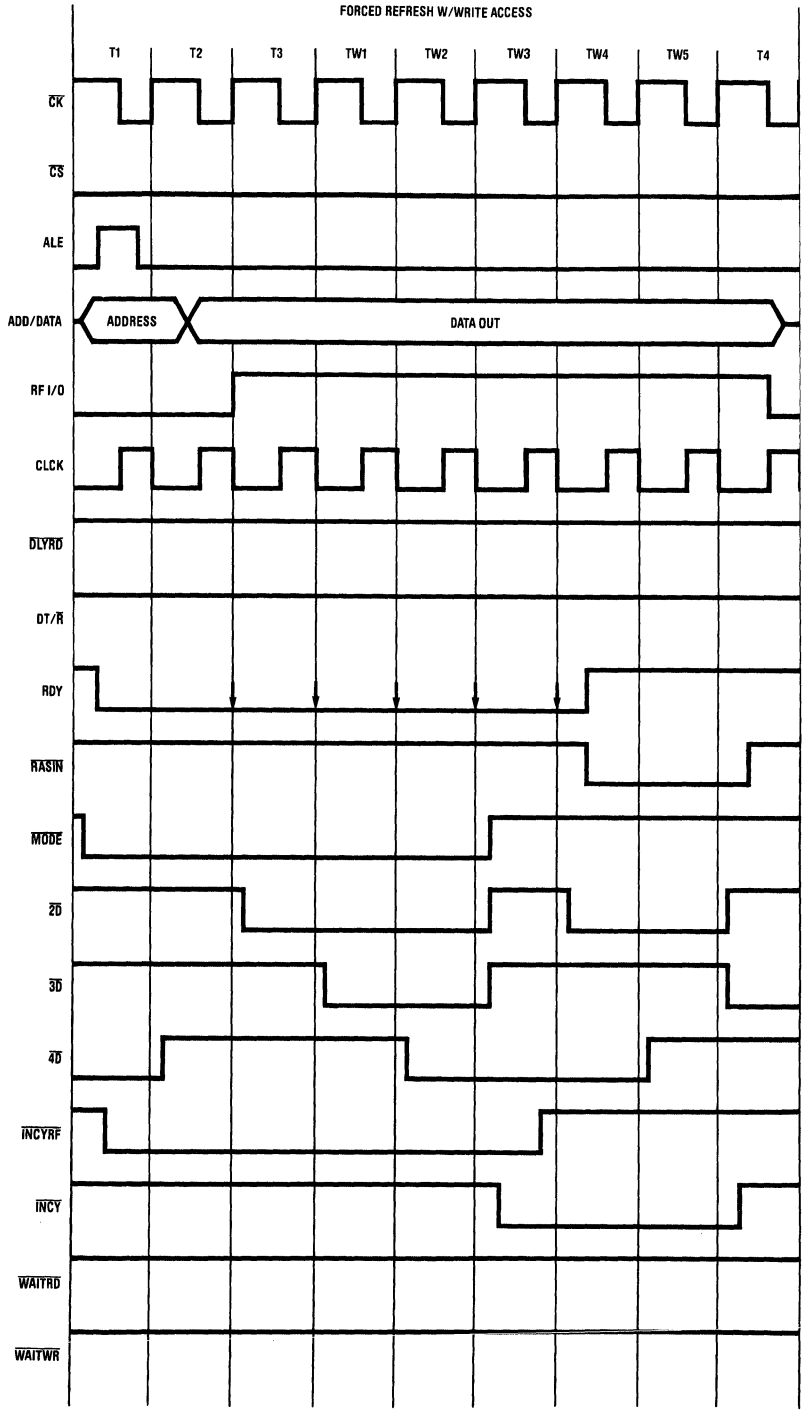
TL/F/8399–3

System Timing Diagrams (Continued)



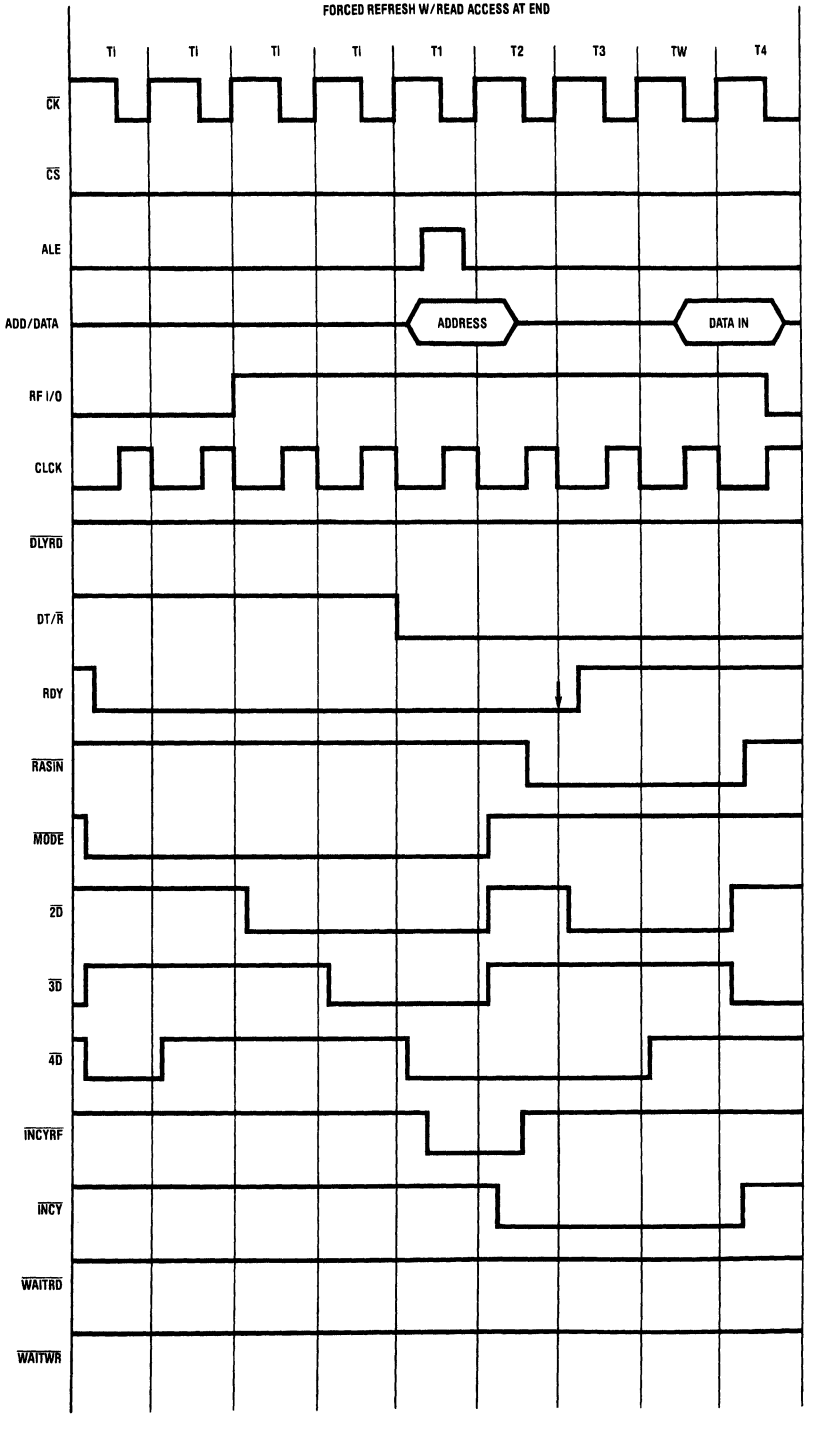
TL/F/8399–5

System Timing Diagrams (Continued)



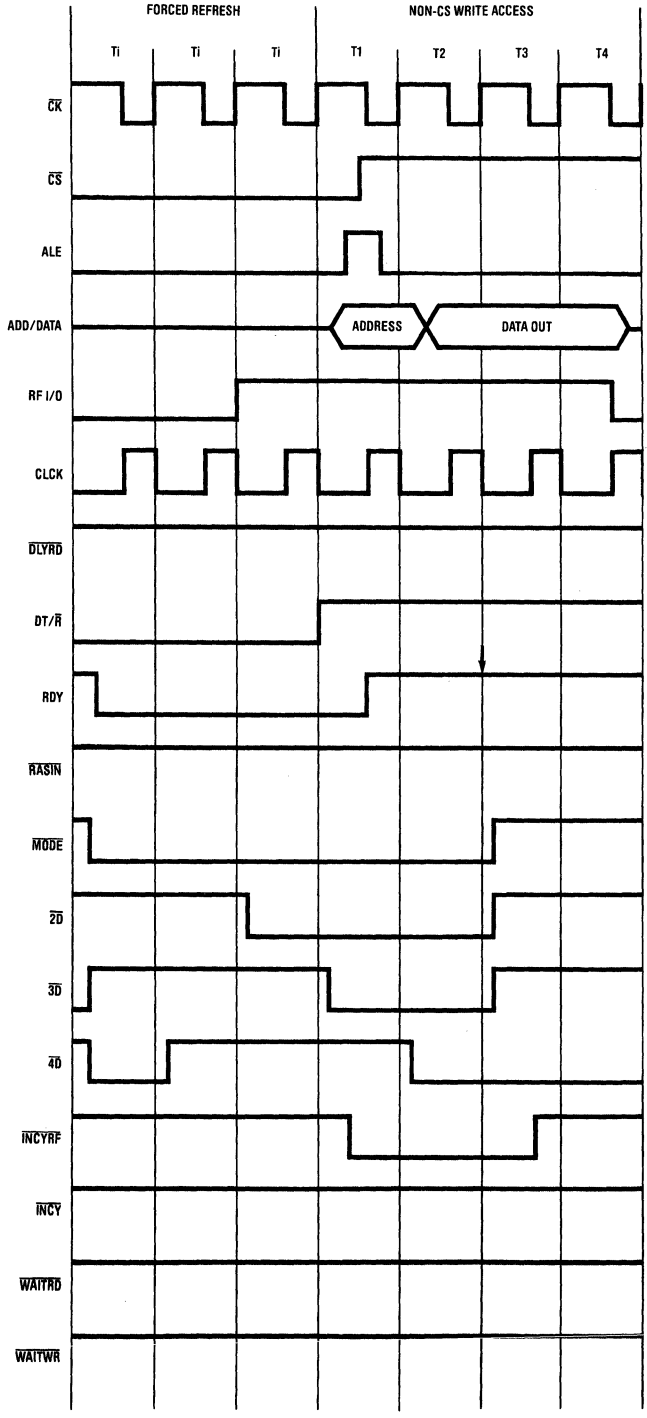
TL/F/8399–6

System Timing Diagrams (Continued)



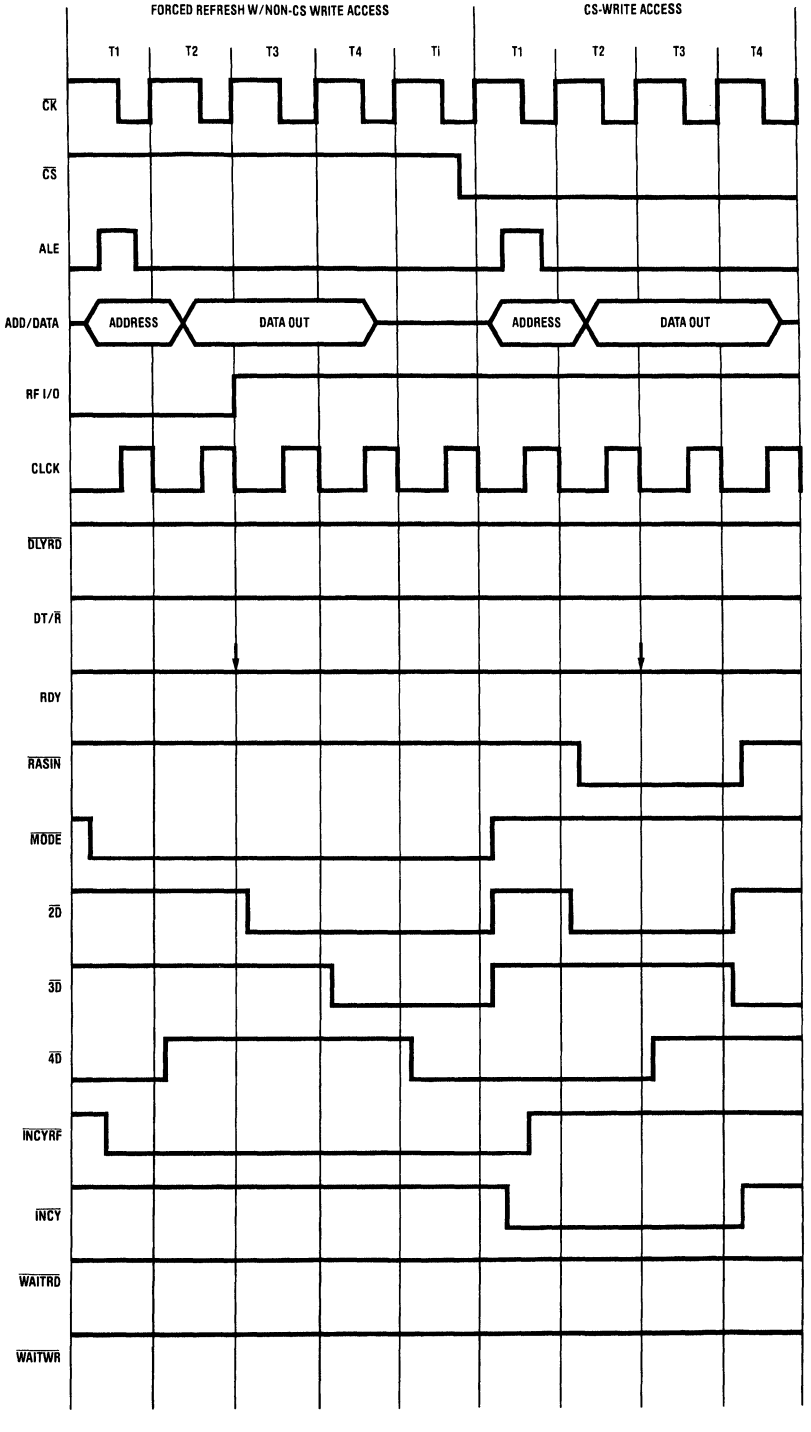
TL/F/8399–7

System Timing Diagrams (Continued)



TL/F/8399–8

System Timing Diagrams (Continued)



TL/F/8399–9

National Semiconductor

PRELIMINARY

DP84512 Dynamic RAM Controller Interface Circuit for the NS32332

General Description

This is a PAL (Programmable Array Logic) device designed to allow an easy interface between the National Semiconductor NS32332 microprocessor and the National Semiconductor DP8417/18/19/28/29 dynamic RAM controller.

This PAL supplies all the control signals needed to perform memory read, burst read, write, and refresh operations up to a frequency of 15 MHz.

Features

- Provides a 3-chip solution for the NS32332/DP8418 (or DP8428) dynamic RAM interface (1 PAL, DP8418 and clock divider)
- Works with all speed versions of the NS32332 up to 15 MHz
- Allows operation of NS32332 at 12 MHz with no WAIT states with standard 120 ns 256k or 1M DRAMs
- Controls DP8417/18/19/28/29 mode 5 accesses and mode 0 forced refreshes automatically
- Allows READ accesses in burst mode
- CPU WAIT states are automatically inserted during contention between DRAM accesses and DRAM refreshes
- Uses standard National Semiconductor PALs (i.e., DMPAL16R4A, the user may want to use faster versions of these PALs at higher CPU operating frequencies)
- The PAL programming equations are provided with comments for easy user modification to his specific requirements

National Semiconductor

PRELIMINARY

DP84522 Dynamic RAM Controller Interface Circuit for the 68020 CPU

General Description

This is a Programmable Array Logic (PAL®) device designed to allow an easy interface between the 68020 microprocessor and the National Semiconductor DP8417, DP8418, DP8419, DP8428 or DP8429 dynamic memory controllers.

This PAL supplies all the control signals needed to perform memory read, write, and refresh operations up to a frequency of 16.7 MHz.

Features

- Provides a 3 or 4 chip solution for the 68020/DP8418 (or DP8428) dynamic RAM interface (1 or 2 PALs, DP8418, and clock divider)
- Works with all speed versions of the 68020 up to 16.7 MHz
- Allows operation of 68020 at 12.5 MHz with 1 WAIT state with standard 120 ns 256k DRAMs
- Controls DP8418/28 mode 5 accesses and mode 1 or 0 forced refreshes automatically
- Allows memory interleaving if desired
- CPU WAIT states are automatically inserted during contention between memory interleaving/DRAM accesses/ DRAM refreshes
- Uses standard National Semiconductor PALs (i.e., DMPAL16R4A; the user may want to use faster versions of these PALs at higher CPU operating frequencies)
- The PAL programming equations are provided with comments for easy user modification to his specific requirements

Functional Description

The following description applies only to the DP8418 or DP8428 since "RFI/O" going low initiates a mode 0 externally controlled forced refresh. This forced refresh resets the internal refresh request logic on the DP8418 or DP8428, but will not reset the internal logic on the DP8409A.

A memory cycle starts when chip select ($\overline{CS}$) and the address strobe ($\overline{AS}$) become true. $\overline{RASIN}$ is supplied from the PALs to the DP8418 DRAM controller, which then supplies $\overline{RAS}$ to the selected $\overline{RAS}$ bank. After the necessary row address hold time, the DP8418 switches the address outputs to the column address. The DP8418 then supplies the required $\overline{CAS}$ signal to the DRAM.

The first PAL (PAL #1) supports byte operations by producing four WRITE enables, one for each possible byte of the 32 bit word (upper, upper middle, lower middle, and lower write enable). These WRITE enables are produced externally from the 68020 "DATA STROBE" and "READ/WRITE" outputs. Since it is possible that all WRITE cycles may be LATE WRITE cycles ("WRITE ENABLE" occurring after "COLUMN ADDRESS STROBE") memory buffers should be used instead of transceivers to separate the data in from the data out of the DRAMs.

The second PAL (PAL #2) supports byte operations by producing four COLUMN ADDRESS STROBES, one for each of the possible bytes of the 32-bit word. This PAL terminates the DP8418 "$\overline{RASIN}$" input early but holds the DRAM data valid by latching the byte "$\overline{CAS}$'s" externally. This method of supporting byte writes allows transceivers to be used, or to directly connect the DRAM data in and data out pins to the 68020 data bus I/O pins.

Hidden REFRESH cycles are not allowed in this set of PALs because of the need for adequate $\overline{RAS}$ precharge times in all circumstances and the desire not to be inserting WAIT states into access cycles of other system elements.

These PALs perform externally controlled forced refreshes automatically (mode 0). A refresh cycle occurs when the DP8418 input RFCK transitions low and the RFIO signal goes low requesting a refresh cycle. The PAL responds by pulling RFSH low (M2 and M0) if there are no current DRAM accesses in progress. If a DRAM access is in progress the PAL waits until the current access is completed before performing the forced refresh cycle. If an access is requested during the forced refresh cycle WAIT states are automatically inserted into the access cycle until the refresh cycle is completed and adequate $\overline{RAS}$ precharge has been completed. The pending DRAM access cycle is then performed. The input signal "$\overline{NOWAIT}$" allows one to vary the amount of time required to do a refresh (see the description of the "$\overline{NOWAIT}$" input in the pin description section). In one of the timing diagrams the "$\overline{RFSH}$" output was tied to the "$\overline{NOWAIT}$" input to decrease the length of the refresh cycle but still insert one wait state in normal DRAM access cycles (see *Figure 5*).

The first PAL (PAL #1) supports memory interleaving to guarantee adequate $\overline{RAS}$ precharge time during two consecutive accesses to the same DRAM bank. This is performed by looking at the lower address bit or bits, A2 and/or A3. If the processor is sequentially accessing the DRAM each $\overline{RAS}$ output will have plenty of precharge time. But if the system tries to access the same bank twice in a row the access will be delayed until adequate $\overline{RAS}$ precharge time has been met. During this time WAIT states will automatically be inserted into the pending access cycle.

The second PAL (PAL #2) guarantees adequate $\overline{RAS}$ precharge time (one and one half system clock periods) by ending the DP8418 "$\overline{RASIN}$" input early. The DRAM data is held valid by externally latching the DRAM "$\overline{CAS}$" input as explained earlier. This has the additional benefit of sim-

Functional Description (Continued)

plifying the memory interface of the 68020 by eliminating the external components needed for interleaving, though as one approaches 16.7 MHz the interleaving circuitry may again become necessary to guarantee adequate "$\overline{RAS}$" precharge time.

For PAL #1 an external "D type" flip-flop or another PAL could be used for the support of memory interleaving. If one is not using memory interleaving (10 MHz or below) the "PREVO" input can be used for some other function and the equations of "$\overline{RASIN}$" that employ "PREVO" can be adjusted.

The PAL equations for this interface are written in the National Semiconductor PLAN™ format, which differs from the standard PALASM™ format.

EXAMPLE: PLAN FORMAT

```
/RASIN := RFSH*/2D*/AS
```

This translates as, "$\overline{RASIN}$" is low after the rising edge of the input clock given that "$\overline{RFSH}$" was high and "$\overline{2D}$" was low and "$\overline{AS}$" was low a setup time before the clock transitions high (here $\overline{RASIN}$, $\overline{2D}$, and $\overline{RFSH}$ are outputs of the PAL and $\overline{AS}$ is an input).

EXAMPLE: PALASM FORMAT

```
RASIN := /RFSH*2D*AS
```

The above expression means the same as the PLAN format expression except it is written in PALASM format. In other words; "$\overline{RASIN}$" will go low after the rising edge of the clock given that "$\overline{RFSH}$" was high, "$\overline{2D}$" was low, and "$\overline{AS}$" was low a setup time before the clock transitions high (here $\overline{RASIN}$, $\overline{2D}$, and $\overline{RFSH}$ are outputs of the PAL and $\overline{AS}$ is an input).

Depending on the specific type of PALs and logic used the user can calculate the speed requirements for the DRAM at the specified processor frequency as follows:

CALCULATION OF DRAM "t_{RAC}" $\overline{RAS}$ ACCESS TIME AND "t_{CAC}" $\overline{CAS}$ ACCESS TIME REQUIRED FOR A 12.5 MHz 68020, 1 WAIT STATE, MICROPROCESSOR SYSTEM

#1) $\overline{RASIN}$ generation time = "S0" + "S1" + 1 combinational output delay of the PAL generating the "$\overline{RASIN}$" output (assume DMPAL16R4B) = 80 ns + 15 ns = 95 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ out delay of the DP8418 = 20 ns maximum (used to determine "t_{RAC}")

#3) $\overline{RASIN}$ to $\overline{CAS}$ out delay of the DP8418 DRAM contoller driving a load of 2 banks of 256k DRAMs, each bank containing 36 (32 DRAMs plus byte parity) = 72 DRAMs

Since this is under the specified load in the data sheet (approximately 88 DRAMs) approximately 3 ns can be subtracted from the data sheet number, giving 80 ns − 3 ns = 77 ns maximum (used to determine "t_{CAC}")

#4) 74AS244 buffer delay = 7 ns maximum

#5) Data setup time required from the falling edge of "S4" clock = 10 ns maximum

A normal 12.5 MHz 68020 access cycle (with 1 WAIT state inserted) contains 4 clock periods of 80 ns per period.

The required DRAM "t_{RAC}" (row access time) can be calculated from

S0 + S1 + S2 + S3 + SW1 + SW2 + S4
(minimum 1/2 period) − #1 − #2 − #4 − #5
= 40 + 40 + 40 + 40 + 40 + 40 + 35
− 95 − 20 − 7 − 10 = 143 ns

The required DRAM "t_{CAC}" (column access time) can be calculated from

S0 + S1 + S2 + S3 + SW1 + SW2 + S4
(minimum 1/2 period) − #1 − #3 − #4 − #5
= 40 + 40 + 40 + 40 + 40 + 40 + 35
− 95 − 77 − 7 − 10 = 86 ns

The DRAMs selected for this system must satisfy both the "t_{RAC} and t_{CAC}" requirements. Therefore the DRAMs must have a "t_{RAC}" (row access time) less then or equal to 143 ns and a "t_{CAC}" (column access time) less than or equal to 86 ns to be used in this system, under worst case conditions, for a 1 WAIT state 12.5 MHz 68020 system. Common 120 ns 64k or 256k DRAMs meet this specification. If one is using PAL #2, producing external "CAS's" and not using any external transceivers he could possibly use 150 ns DRAMs in the above example.

If one is using PAL #2, the calculated "t_{RAC}" and "t_{CAC}" may differ from the actual system values, depending upon the external circuitry used to produce the byte "$\overline{CAS}$'s". The DP8418 "$\overline{RASIN}$–$\overline{CAS}$" low will be approximately 10–15 ns less than the value given in the data sheet because of the small loading on the DP8418 "$\overline{CAS}$" output. The external circuitry needed to produce the byte "$\overline{CAS}$'s" should be loaded such that the column address (from DP8418) is valid when "$\overline{CAS}$" goes low. For this reason "$\overline{RASIN}$-byte $\overline{CAS}$" may be longer than the value used in the "t_{RAC}, t_{CAC}" calculations, and therefore may give a smaller "t_{RAC}, t_{CAC}" than was calculated.

68020 PAL Inputs and Outputs

(Pin number of the PAL on the left)

PAL #1 Inputs

1) "CK"	This is the system clock.
2) "$\overline{AS}$"	Address strobe from 68020.
3) "$\overline{RFRQ}$"	This is the refresh request from the DP8418.
4) "$\overline{CS}$"	This is the chip select (see system block diagram).
5) "R"	READ/$\overline{WRITE}$ output pin from the 68020.
6) "CLK"	The system clock.
7) "PREVO"	This output holds the previously accessed DRAM "$\overline{RAS}$" bank.
8) "B0"	This input is the address bit "A2" and is used to determine which "$\overline{RAS}$" bank the system is accessing.
9) "$\overline{NOWAIT}$"	This PAL always inserts one WAIT state into every 68020 access cycle. This input,

68020 PAL Inputs and Outputs (Continued)

if low, allows the DRAM to be accessed with no wait states inserted into the access cycle. This input also, if low during a refresh, shortens the length of the refresh cycle by one clock period. This causes the $\overline{RAS}$ pulse width (during a refresh) and the $\overline{RAS}$ precharge time (after a refresh) to be shorter.

11) "$\overline{OE}$" This input enables the PAL outputs.

PAL #1 Outputs

19) "$\overline{XDLY}$" This signal is used to guarantee one period of "$\overline{RFSH}$" high to "$\overline{RASIN}$" low time and to guarantee two periods of RAS precharge time in consecutive accesses to the same DRAM bank.

18) "$\overline{RASIN}$" This signal causes $\overline{RAS}$ (or $\overline{RAS}$s) to go low during a DRAM access or refresh.

17) "$\overline{RFSH}$" This signal initiates a DRAM Refresh.

16) "$\overline{1DLY}$" A delay that is used internally.

15) "$\overline{2DLY}$" A delay that is used internally.

14) "$\overline{RFREQ}$" Refresh request (from the DP8418) synchronized to the system clock.

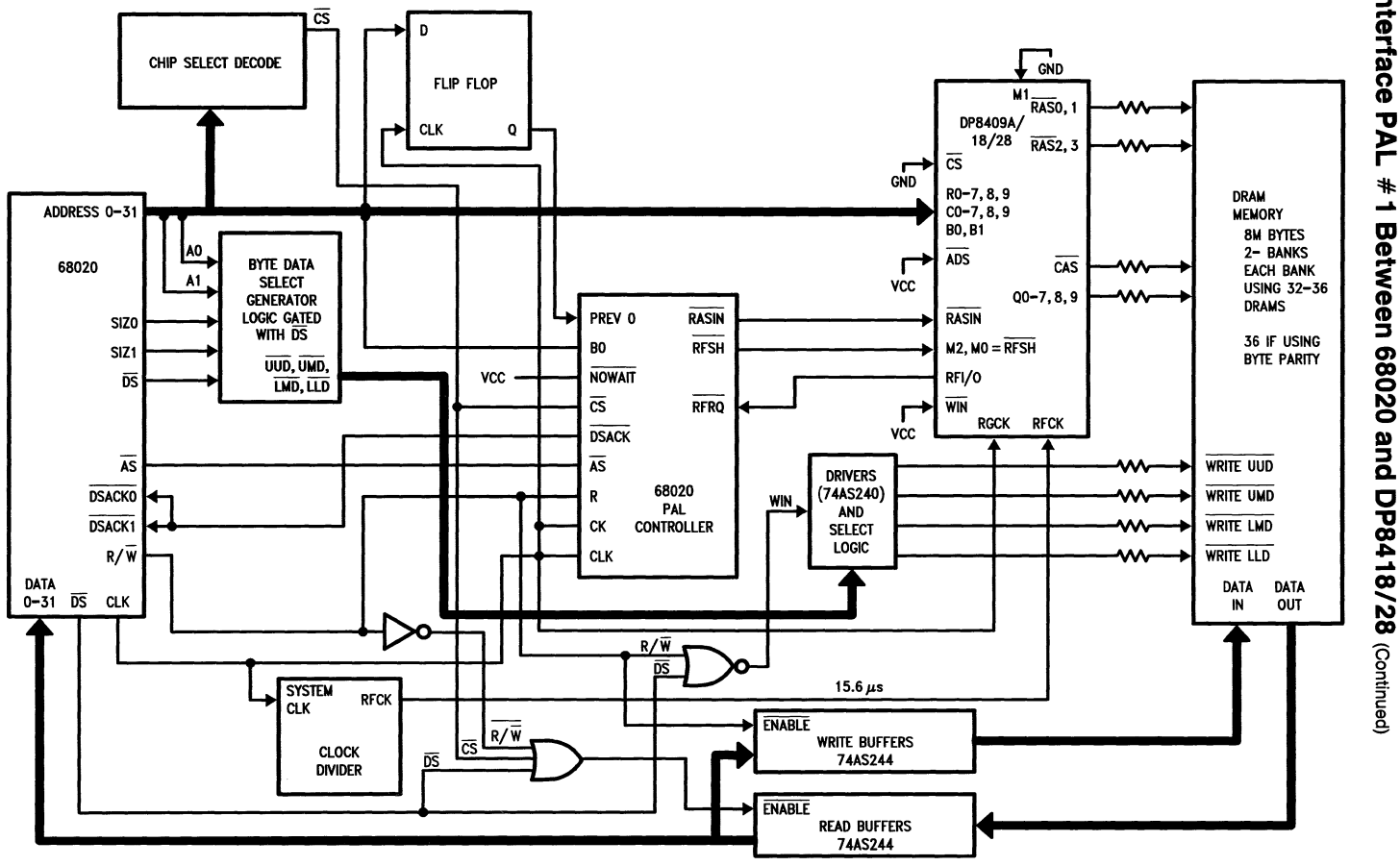
13) "$\overline{RFREQCK}$" This input synchronizes "$\overline{RFREQ}$" to the falling edge of the input system clock "CLK" and is used in arbitrating between refreshes and accesses (see "$\overline{RASIN}$" equations).

12) "$\overline{DSACK}$" This output goes to the 68020 "$\overline{DSACK0}$, $\overline{1}$" data acknowledge input. This output allows WAIT states to be inserted into DRAM access cycles during access/refresh/RAS precharge contention.

Interface PAL # 1 Boolean Equations

This PAL will work up to 16.7 MHz with the 68020. This PAL uses mode 0 (M0 = M1 = M2 = low) for doing externally controlled forced refreshes, guaranteeing more than 2.5 periods of RGCK RAS pulse width ("NOWAIT" = high). If "NOWAIT" is low the refresh is shortened by one clock period. This PAL will only work with the DP8417/18/19/28/29 since it uses mode 0 to reset the RFSH request (RFIO) signal.

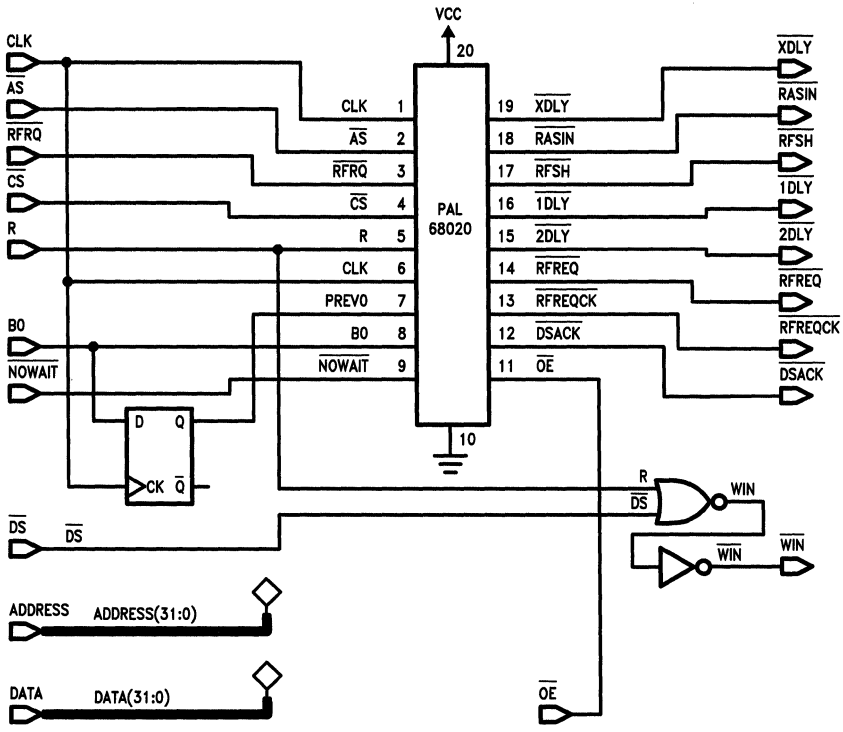
```
DMPAL16R4A
CK   AS   RFRQ   CS   R   CLK   PREVO   BO   /NOWAIT   GND
/OE  /DSACK  /RFREQCK  /RFREQ  /2DLY  /1DLY  /RFSH  /RASIN  /XDLY  VCC

IF (VCC) /XDLY = RFSH*/2DLY*RASIN*PREVO*BO      ;Same bank interleave
     +RFSH*/2DLY*RASIN*/PREVO*/BO               ;Same bank interleave
     +/RFSH*1DLY*/2DLY*/RASIN*NOWAIT*/CLK       ;"/XDLY" low during RFSH
     +/RFSH*1DLY*2DLY*RASIN*RFREQ*/NOWAIT       ;"/XDLY" low during RFSH
     +/XDLY*/RFSH*RFREQ                         ;Hold "/XDLY" low
     +/XDLY*RASIN*/AS*CLK                       ;Hold "/XDLY" low

IF (VCC) /RASIN = /RFSH*RFREQ*/1DLY             ;RFSH "/RASIN"
  +/RASIN*/RFSH*/2DLY*XDLY                      ;Hold "/RASIN" low
  +RFSH*RFREQCK*/AS*/CS*PREVO*/BO*CLK           ;Start "/RASIN"
  +RFSH*RFREQCK*/AS*/CS*/PREVO*BO*CLK           ;Start "/RASIN"
  +RFSH*RFREQCK*/AS*/CS*2DLY*XDLY*CLK           ;After idle states
  +/RASIN*RFSH*/AS*/CS                          ;Hold "/RASIN" low
  +/RASIN*/CLK                                  ;Hold "/RASIN" low

IF (VCC) /RFREQCK = /RFREQ*/CLK                 ;Start from falling clock
     +/RFREQCK*/RFREQ                           ;  edge

IF (/CS) /DSACK = /CS*RFSH*/RASIN*NOWAIT*/CLK   ;One WAIT state
  +/DSACK*/CS*RFSH*/RASIN*/AS                   ;Hold "/DSACK" low
  +/CS*RFSH*/AS*/NOWAIT*XDLY                    ;No WAIT state in access

/RFSH := /RFREQ*RASIN*/1DLY*/2DLY               ;Start RFSH
     +/RFREQ*RASIN*1DLY                         ;Start RFSH
     +/RFSH*/RFREQ                              ;Hold RFSH low
     +/RFSH*/RASIN                              ;Hold RFSH low
     +/RFSH*/1DLY                               ;Hold RFSH low

/1DLY := /RFSH*2DLY*/RFREQ                  ;Start "/1DLY" during RFSH
     +/RFSH*/1DLY*2DLY*XDLY                 ;Continue "/1DLY" during RFSH
     +RFSH*/RASIN                           ;Start "/1DLY" during /RASIN

/2DLY := /RFSH*/1DLY                        ;Start "/2DLY" during RFSH
     +/RFSH*2DLY*/RFREQ*/NOWAIT             ;Shorten RFSH
     +RFSH*/RASIN*/1DLY                     ;Start "/2DLY" during /RASIN
     +RFSH*/RASIN*/NOWAIT                   ;Shorten access

/RFREQ := /RFRQ                             ;Synchronize to system clock
```



TL/F/8589-1

FIGURE 1. 68020 System Diagram for PAL # 1

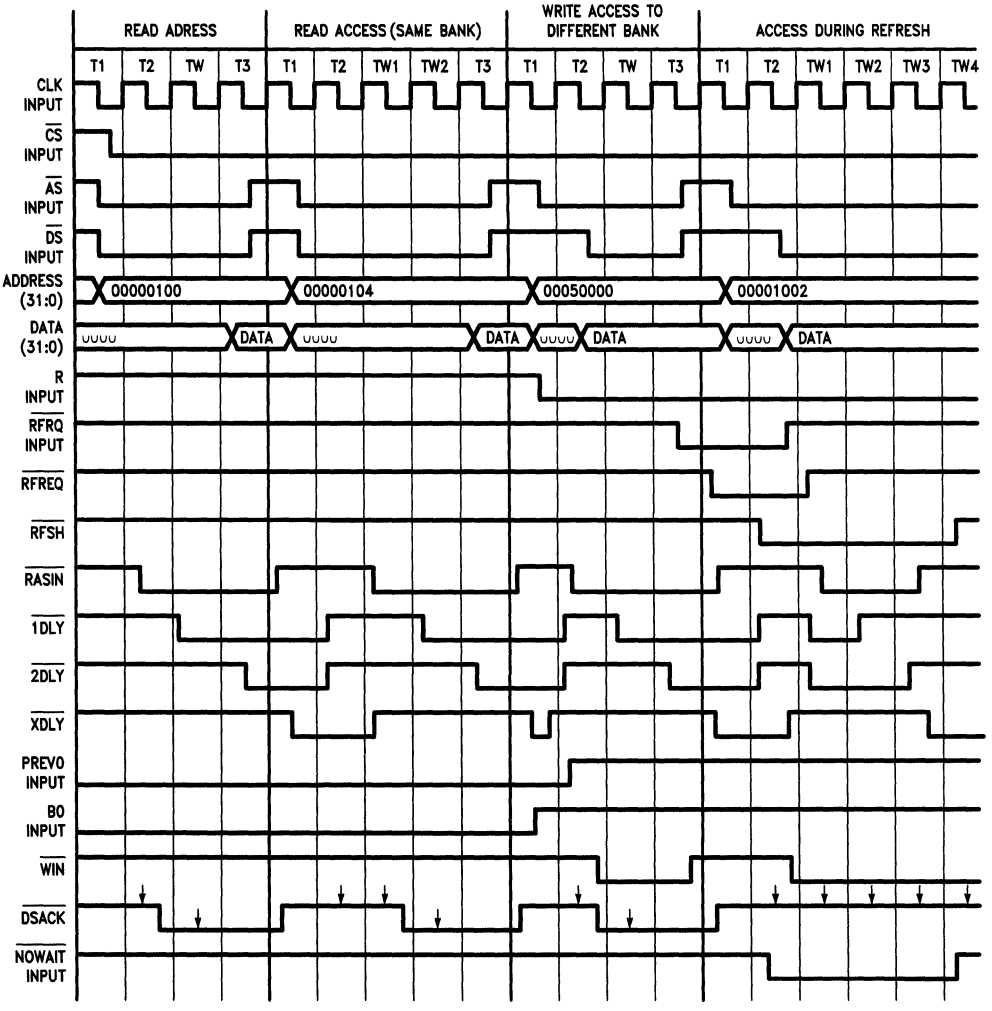
Interface PAL # 1 Between 68020 and DP8418/28 (Continued)



TL/F/8589–2

FIGURE 2. PAL # 1 Simulation Diagram

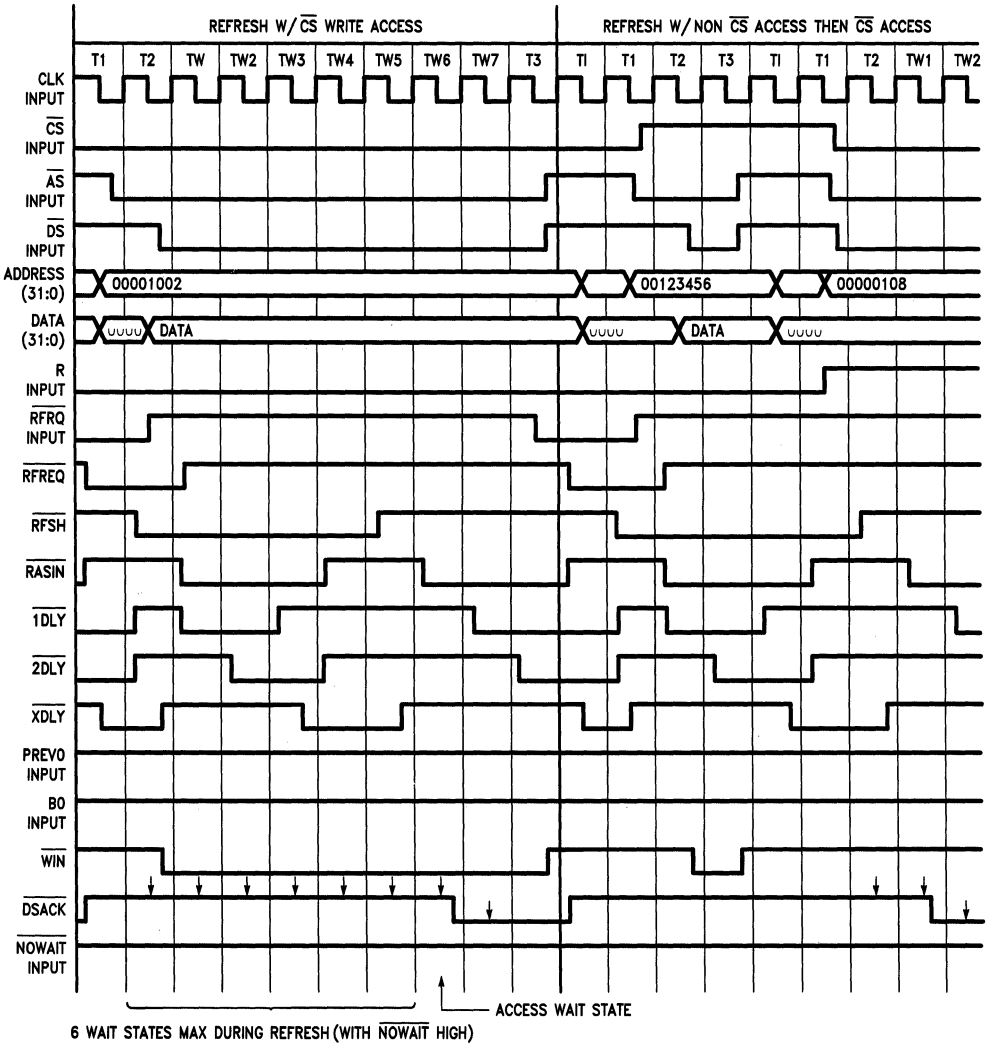
Interface PAL # 1 System Timing Diagram



TL/F/8589–3

FIGURE 3

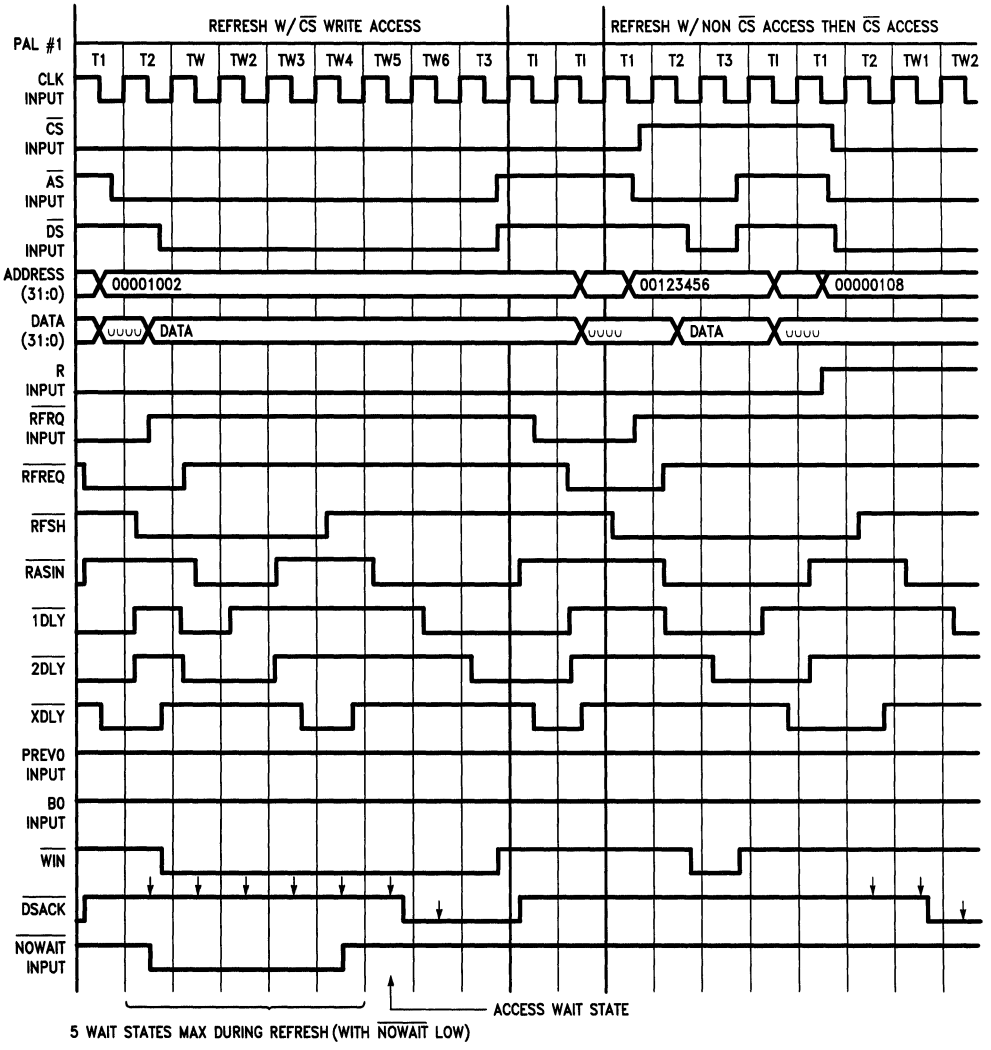
Interface PAL # 1 System Timing Diagram (Continued)



TL/F/8589–4

FIGURE 4

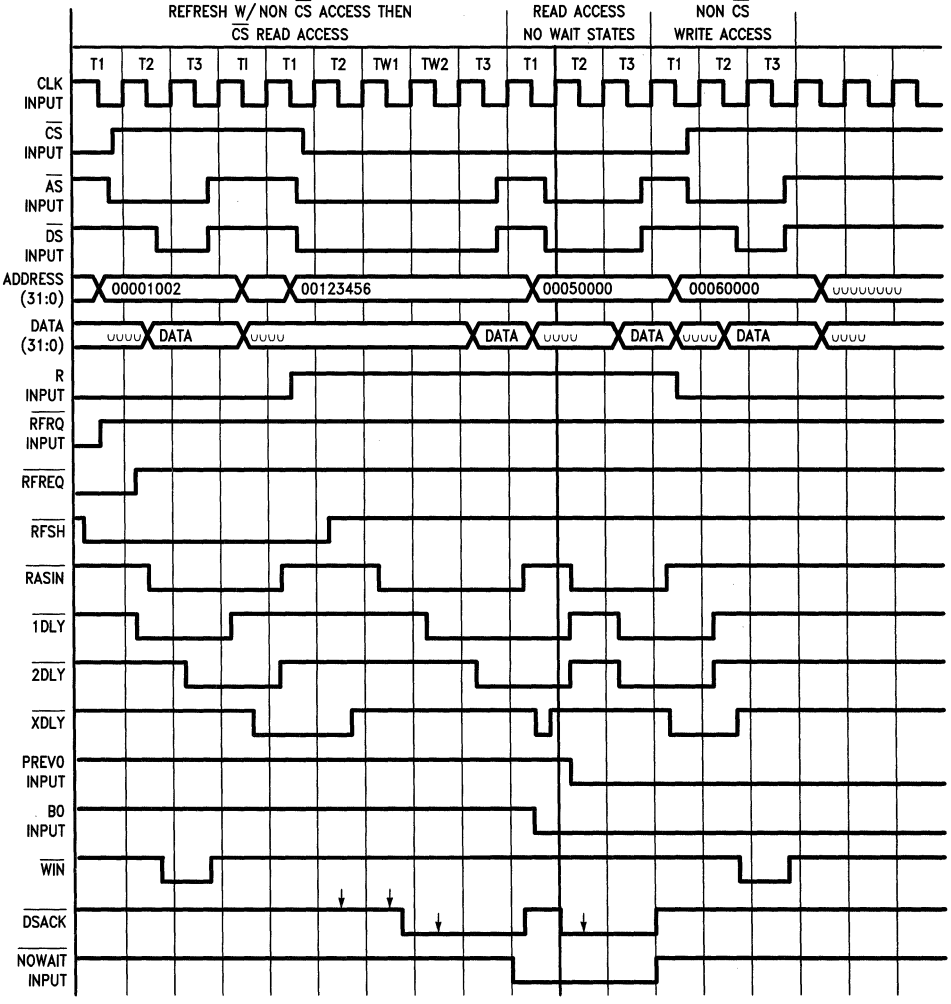
Interface PAL # 1 System Timing Diagram (Continued)



TL/F/8589–5

FIGURE 5

Interface PAL #1 System Timing Diagram (Continued)



TL/F/8589-6

FIGURE 6

Interface PAL #2 Boolean Equations

This PAL is similar to PAL #1 but ends "$\overline{\text{RASIN}}$" one half period earlier than PAL #1 and relies on the external generation of byte "$\overline{\text{CAS}}$'s" to hold the data valid during 68020 READ access cycles.

```
DMPAL16R4A
CK  /AS  /RFRQ  /CS  R  CLK  NC1  NC2  /NOWAIT  GND
/OE  /DSACK  /RFREQCK  /RFREQ  /2DLY  /1DLY  /RFSH  /RASIN  /XDLY  VCC

IF (VCC) /XDLY = RFSH*/2DLY*/AS                ;"/XDLY" during access
     +/RFSH*1DLY*/2DLY*/RASIN*NOWAIT*/CLK      ;"/XDLY" during RFSH
     +/RFSH*1DLY*2DLY*RASIN*RFREQ*/NOWAIT      ;"/XDLY" during RFSH
     +/RFSH*/XDLY*RFREQ                        ;Hold "/XDLY" low
     +/XDLY*RASIN*/AS*CLK                      ;Hold "/XDLY" low

IF (VCC) /RASIN = /RFSH*RFREQ*/1DLY          ;RFSH "/RASIN"
  +/RFSH*/RASIN*/2DLY*XDLY                   ;Hold in RFSH
  +/RFSH*/RASIN*/CLK                         ;Hold in RFSH
  +RFSH*RFREQCK*/AS*/CS*XDLY*CLK             ;Start "/RASIN"
  +RFSH*RFREQCK*/AS*/CS*2DLY*XDLY*CLK        ;After idle states
  +RFSH*/RASIN*/AS*/CS*XDLY                  ;Hold "/RASIN" low
  +RFSH*/RASIN*CLK                           ;Hold "/RASIN" low

IF (VCC) /RFREQCK = /RFREQ*/CLK              ;Start from falling clock
     +/RFREQCK*/RFREQ                        ;  edge

IF (VCC) /DSACK = /CS*RFSH*/RASIN*NOWAIT*/CLK   ;One WAIT state
  +/DSACK*/CS*RFSH*/RASIN*/AS                   ;Hold "/DSACK" low
  +/CS*RFSH*/AS*/NOWAIT*XDLY                    ;No WAIT state in access

/RFSH := /RFREQ*RASIN*/1DLY*/2DLY           ;Start RFSH
     +/RFREQ*RASIN*1DLY                     ;Start RFSH
     +/RFSH*/RFREQ                          ;Hold RFSH low
     +/RFSH*/RASIN                          ;Hold RFSH low
     +/RFSH*/1DLY                           ;Hold RFSH low

/1DLY := /RFSH*2DLY*/RFREQ                ;Start "/1DLY" during RFSH
     +/RFSH*/1DLY*2DLY*XDLY               ;Continue "/1DLY" during RFSH
     +RFSH*/RASIN                         ;Start "/1DLY" during /RASIN

/2DLY := /RFSH*/1DLY                      ;Start "/2DLY" during RFSH
     +/RFSH*2DLY*/RFREQ*/NOWAIT           ;Shorten RFSH
     +RFSH*/RASIN*/1DLY                   ;Start "/2DLY" during /RASIN
     +RFSH*/RASIN*/NOWAIT                 ;Shorten access

/RFREQ := /RFRQ                           ;Synchronize to system clock
```

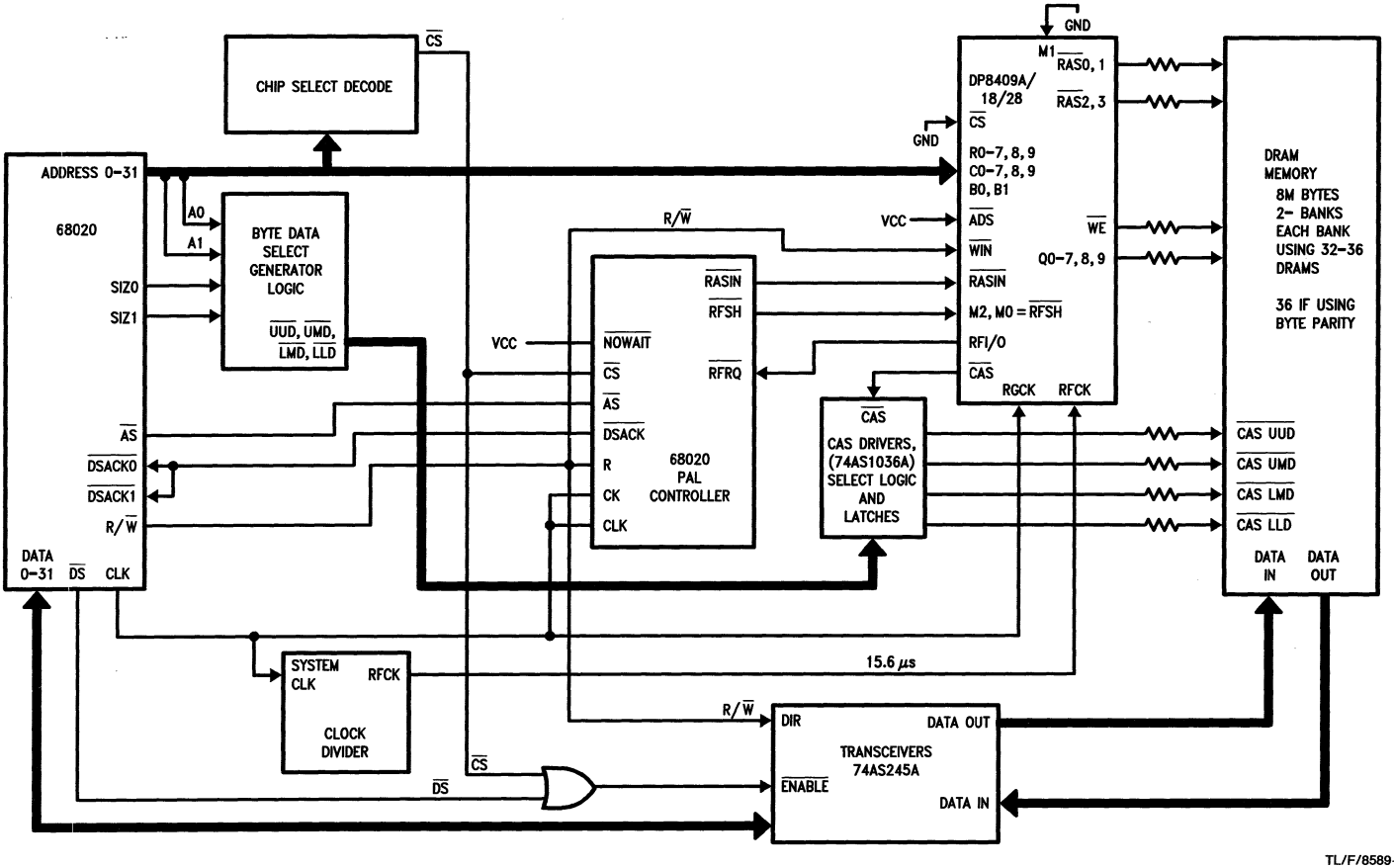
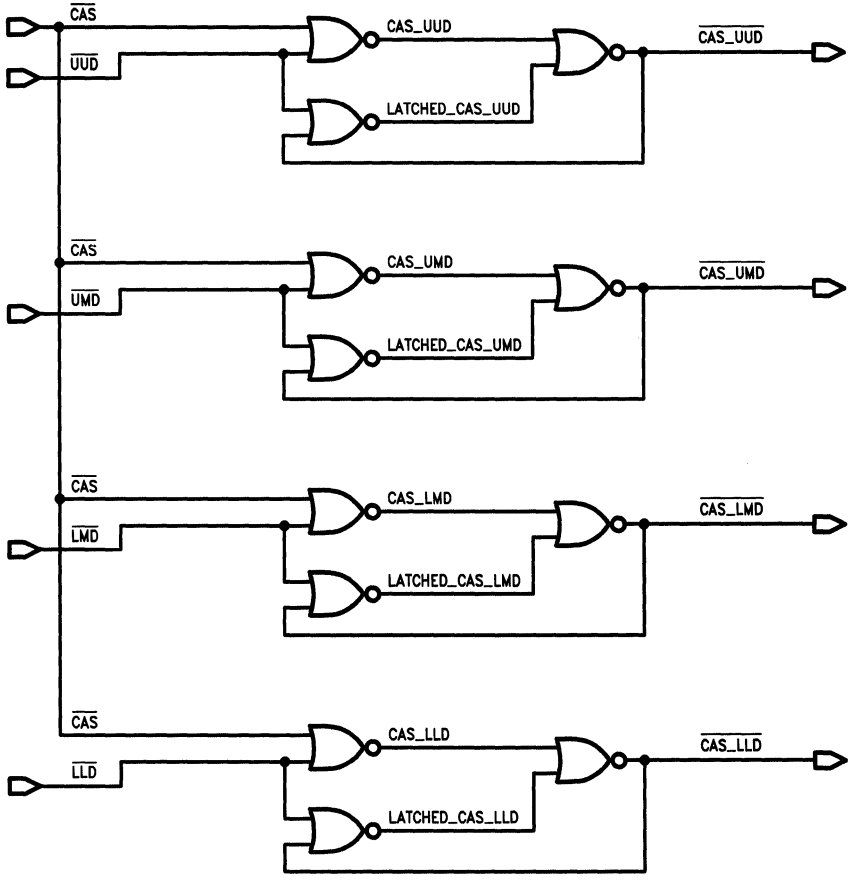

Interface PAL #2 68020 System Diagram
CHIP SELECT DECODE
CS
ADDRESS 0-31
68020
A0
A1
BYTE DATA SELECT GENERATOR LOGIC
UUD, UMD, LMD, LLD
SIZ0
SIZ1
AS
DSACK0
DSACK1
R/W
DATA 0-31
DS
CLK
SYSTEM CLK
RFCK
CLOCK DIVIDER
15.6 μs
R/W
VCC
NOWAIT
CS
AS
DSACK
R
CK
CLK
68020 PAL CONTROLLER
RASIN
RFSH
RFRQ
GND
DP8409A/ 18/28
M1
RAS0, 1
RAS2, 3
CS
R0-7, 8, 9
C0-7, 8, 9
B0, B1
ADS
WIN
RASIN
M2, M0 = RFSH
RFI/O
CAS
RGCK
RFCK
WE
Q0-7, 8, 9
CAS DRIVERS, (74AS1036A) SELECT LOGIC AND LATCHES
DRAM MEMORY
8M BYTES
2- BANKS
EACH BANK
USING 32-36
DRAMS
36 IF USING
BYTE PARITY
CAS UUD
CAS UMD
CAS LMD
CAS LLD
DATA IN
DATA OUT
DIR
DATA OUT
TRANSCEIVERS 74AS245A
ENABLE
DATA IN
TL/F/8589-7

FIGURE 7

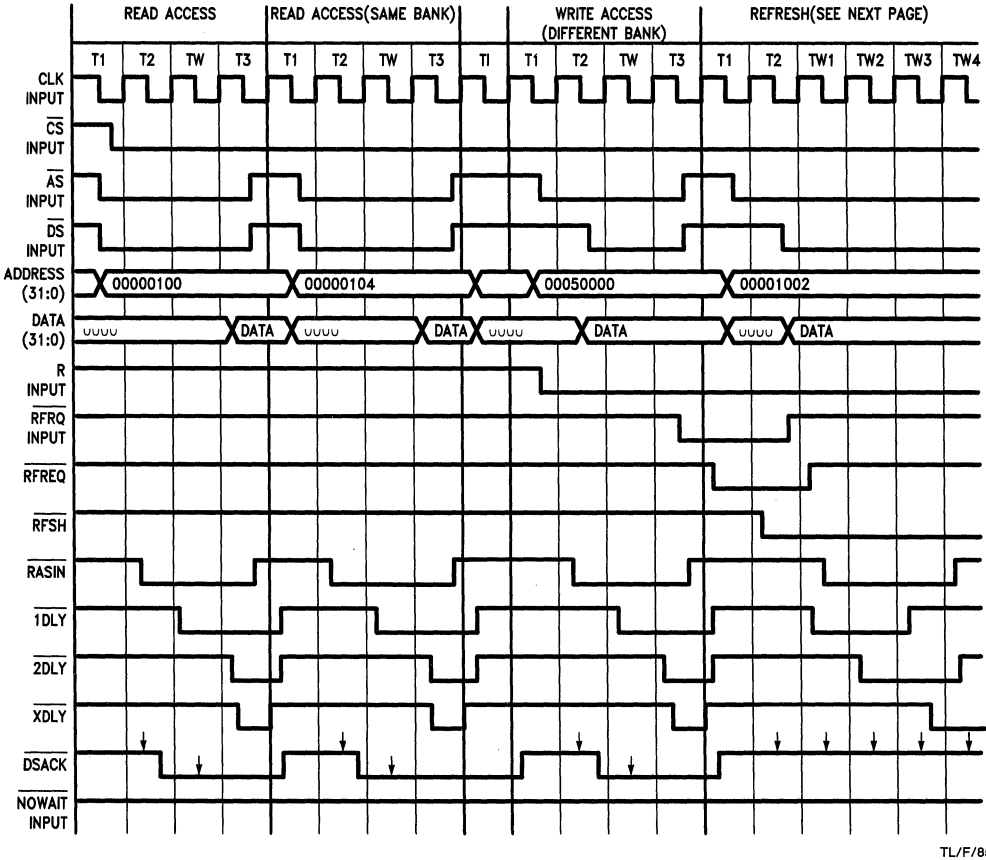
Interface PAL #2 CAS Driver, Select Logic, and Latches



TL/F/8589–8

FIGURE 8

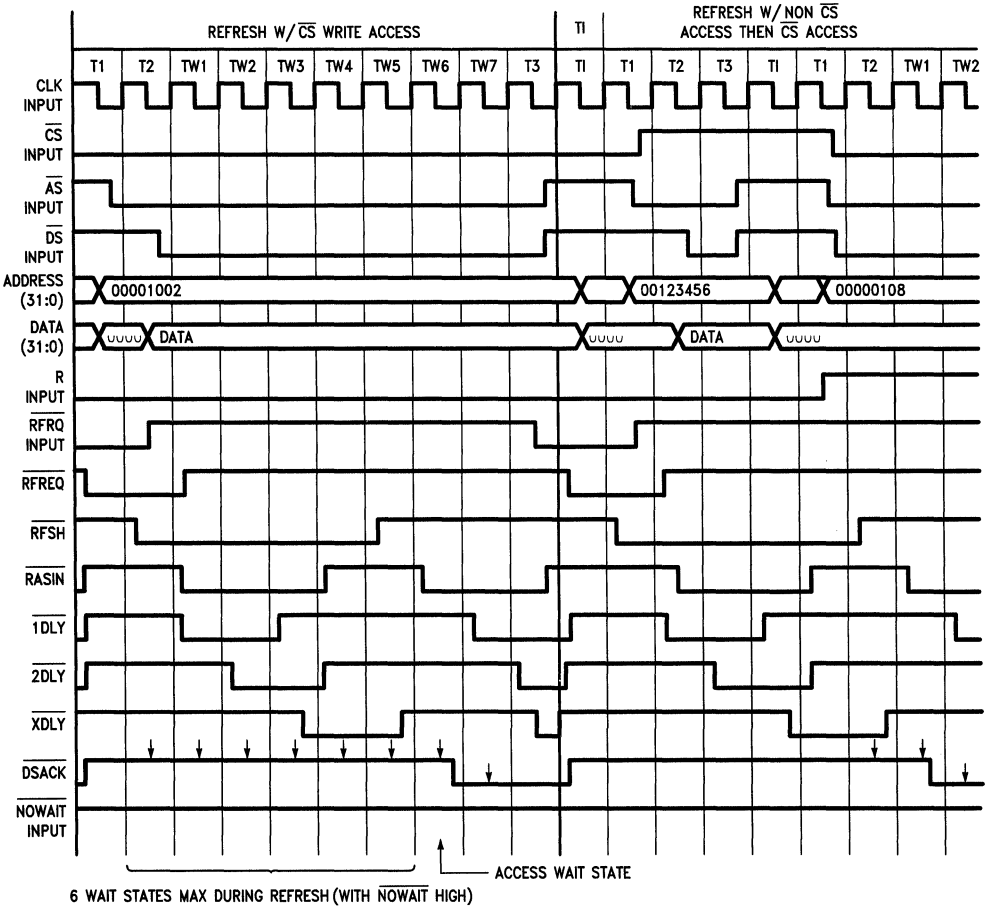
Interface PAL #2 68020 System Timing Diagram



TL/F/8589-9

FIGURE 9

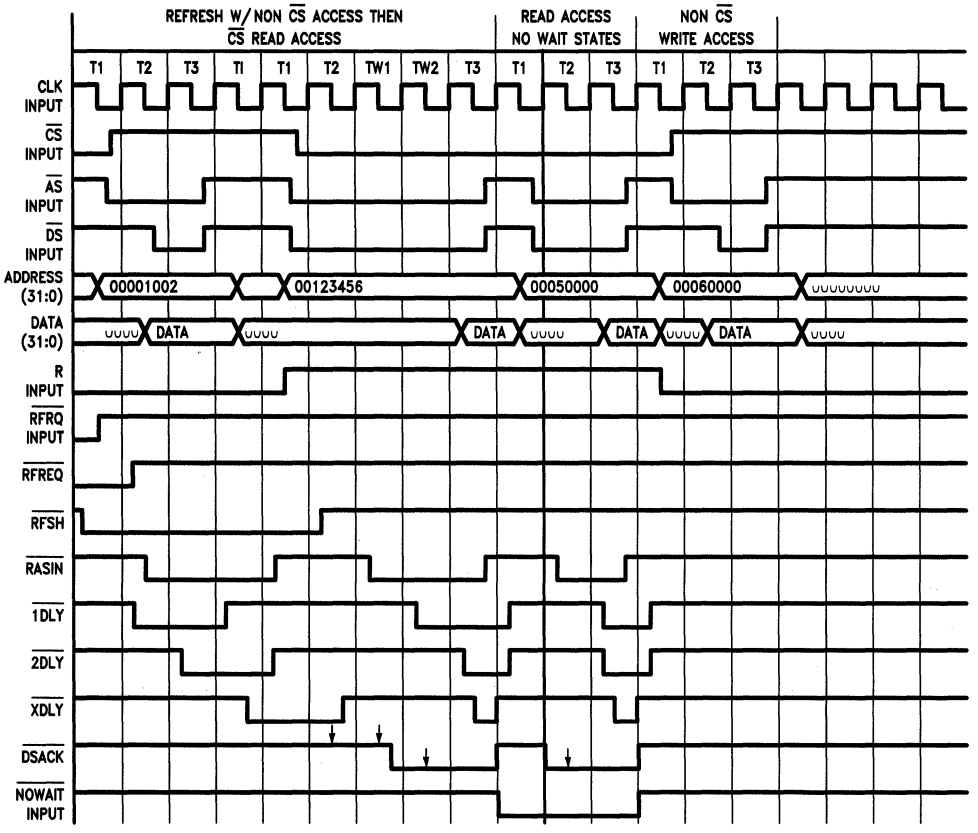
Interface PAL #2 Between 68020 and DP8418 (Continued)



TL/F/8589–10

FIGURE 10. System Timing

Interface PAL #2 Between 68020 and DP8418 (Continued)



TL/F/8589–11

FIGURE 11. System Timing

National Semiconductor

PRELIMINARY

DP84532 Dynamic RAM Controller Interface Circuit for the iAPX 286 CPU

General Description

This is a Programmable Array Logic (PAL®) device designed to allow an easy interface between the 80286 microprocessor and the National Semiconductor DP8419/29 or DP8409A dynamic memory controllers.

This PAL supplies all the control signals needed to perform memory read, write, and refresh operations up to a frequency of 16 MHz.

Features

- Provides a 3- or 4-chip solution for the 80286/DP8419 (or DP8409A/29) dynamic RAM interface (1 or 2 PALs, DP8419, and clock divider)
- Works with all speed versions of the 80286 up to 10 MHz
- Allows operation of 80286 at 8 MHz with no WAIT states with standard 120 ns 256k DRAMs
- Controls DP8409A/19 mode 5 accesses and mode 0 or 1 forced refreshes automatically
- Allows memory interleaving if desired
- CPU WAIT states are automatically inserted during contention between memory interleaving/DRAM accesses/DRAM refreshes
- Uses standard National Semiconductor PALs (i.e., DMPAL16R6A and DMPAL16R4A; the user may want to use faster versions of these PALs at higher CPU operating frequencies)
- The PAL programming equations are provided with comments for easy user modification to his specific requirements

Functional Description

The following description applies to both the DP8409A, DP8419 and DP8429.

A memory cycle starts when chip select ($\overline{CS}$) and the status ($\overline{S0}$*S1) become true. $\overline{RASIN}$ is supplied from the PALs to the DP8419 DRAM controller, which then supplies $\overline{RAS}$ to the selected $\overline{RAS}$ bank. After the necessary row address hold time, the DP8419 switches the address outputs to the column address. The DP8419 then supplies the required $\overline{CAS}$ signal to the DRAM. In order to do byte operations a HIGH WRITE ENABLE and a LOW WRITE ENABLE are produced from PAL #2. All WRITE cycles are LATE WRITE cycles, to assure that valid data is written to the DRAMs. A $\overline{WRITE}$ strobe is produced by PAL #1 to assure enough $\overline{WIN}$ pulse width and to guarantee that valid data is latched into the DRAMs when writing to them. Memory buffers are used externally, to separate the data in from the data out of the DRAMs during LATE WRITE cycles.

Hidden REFRESH cycles are not allowed in this set of PALs because of the need for adequate $\overline{RAS}$ precharge times in all circumstances and the desire not to be inserting WAIT states into access cycles of other system elements.

These PALs perform automatic or externally controlled forced refreshes (mode 0 or 1). A refresh cycle occurs when the DP8419 input RFCK transitions low and the RFIO signal goes low requesting a refresh cycle. The PAL responds by pulling $\overline{RFSH}$ (M2, and/or MO depending on whether mode 1 or 0 is desired) low if there are no current DRAM accesses in progress. If a DRAM access is in progress the PAL waits until the current access is completed before performing the forced refresh cycle. If an access is requested during the forced refresh cycle WAIT states are automatically inserted into the access cycle until the refresh cycle is completed and adequate $\overline{RAS}$ precharge has been completed. The pending DRAM access cycle is then performed.

In order to guarantee adequate $\overline{RAS}$ precharge time during two consecutive accesses to the same DRAM bank, memory interleaving is performed by looking at the two lower address bits, A1 and A2. If the processor is sequentially accessing the DRAM, each $\overline{RAS}$ output will have plenty of precharge time. But if the system tries to access the same bank twice in a row the access will be delayed until adequate $\overline{RAS}$ precharge time has been met. During this time WAIT states will automatically be inserted into the pending access cycle.

The 8 MHz 80286 has two "T" states ("T_S" and "T_C"), it is possible for these PALs to get one clock phase out of sync with the 80286 CPU during access cycles pending while performing a refresh cycle. The two 8 MHz "T" states of the CPU contain four 16 MHz clock periods ("CLK" output of 82284 clock generator). This 2X clock is the clock the interface described herein uses. In other words, the PALs produce a $\overline{RASIN}$ output that is low for three 16 MHz clock periods for the 8 MHz 80286. Since WAIT states insert two 16 MHz clock periods and $\overline{RASIN}$ can start one clock period after $\overline{RFSH}$ transitions high, it is possible for $\overline{RASIN}$ to start one period early and go high one period before the access cycle ends, thus not holding the data valid during a READ access cycle long enough. To counteract this problem the term "ALE" is used in several of the PAL equations ($\overline{RASIN}$, $\overline{1DLY}$, and 2DLY) to sync the $\overline{RASIN}$ output to the access cycle. See the timing diagrams (*Figure 6*) and PAL equations for some further insight into the potential problems. This synchronization could also have been done externally by holding $\overline{CAS}$ low until either $\overline{MWTC}$ or $\overline{MRDC}$ go high, thus holding the READ data valid until the access cycle is over.

Two PALs were designed for this PAL interface. PAL #2 is used mostly for the support of memory interleaving. If one is not using memory interleaving (6 MHz or below) PAL #2 can be omitted and the PAL #1 "$\overline{PRECH}$" input can be tied high. The high and low memory write strobes can be produced externally.

Functional Description (Continued)

The PAL equations for this interface are written in the National Semiconductor PLAN™ format, which differs from the standard PALASM™ format.

EXAMPLE: PLAN FORMAT

```
/RASIN := RFSH*/2D*ALE
```

This translates as, "$\overline{\text{RASIN}}$" is low after the rising edge of the input clock given that "$\overline{\text{RFSH}}$" was high and "$\overline{\text{2D}}$" was low and "ALE" was high a setup time before the clock transitions high (here $\overline{\text{RASIN}}$, $\overline{\text{2D}}$, and $\overline{\text{RFSH}}$ are outputs of the PAL and ALE is an input).

EXAMPLE: PALASM FORMAT

```
RASIN := /RFSH*2D*ALE
```

The above expression means the same as the PLAN format expression except it is written in PALASM format. In other words, "$\overline{\text{RASIN}}$" will go low after the rising edge of the clock given that "$\overline{\text{RFSH}}$" was high, "$\overline{\text{2D}}$" was low, and "ALE" was high a setup time before the clock transitions high (here $\overline{\text{RASIN}}$, $\overline{\text{2D}}$, and $\overline{\text{RFSH}}$ are outputs of the PAL and ALE is an input).

Depending on the specific type of PALs and logic used, the user can calculate the speed requirements for the DRAM at the specified processor frequency as follows:

CALCULATION OF DRAM "t_{RAC}" $\overline{\text{RAS}}$ ACCESS TIME AND "t_{CAC}" $\overline{\text{CAS}}$ ACCESS TIME REQUIRED FOR AN 8 MHz 80286, NO WAIT STATE, MICROPROCESSOR SYSTEM

#1) $\overline{\text{RASIN}}$ generation time = one period of the system clock + one 74AS244 gate delay (the system clock is inverted to the PALs) + one clocked output delay of the PAL generating the $\overline{\text{RASIN}}$ output (assume DMPAL16R6B) = 62.5 ns + 4.5 ns + 12 ns = 79 ns maximum

#2) $\overline{\text{RASIN}}$ to $\overline{\text{RAS}}$ out delay of the DP8419 = 20 ns maximum (used to determine "t_{RAC}")

#3) $\overline{\text{RASIN}}$ to $\overline{\text{CAS}}$ out delay of the DP8419 DRAM controller driving a load of 4 banks of 256k DRAMs, each bank containing 18 (16 DRAMs plus byte parity) = 72 DRAMs

Since this is under the specified load in the data sheet (approximately 88 DRAMs) approximately 3 ns can be subtracted from the data sheet number, giving 80 ns − 3 ns = 77 ns maximum (used to determine "t_{CAC}").

#4) 74AS244 buffer delay = 7 ns maximum

#5) Data setup time required from the end of "T_C" phase 2 clock cycle = 10 ns minimum

A normal 8 MHz 80286 access cycle contains 4 clock periods (16 MHz) of 62.5 ns per period = 250 ns

The required DRAM "t_{RAC}" (row access time) can be calculated from 250 ns − #1 − #2 − #4 − #5 = 134 ns

The required DRAM "t_{CAC}" (column access time) can be calculated from 250 ns − #1 − #3 − #4 − #5 = 77 ns.

The DRAMs selected for this system must satisfy both the "t_{RAC}" and "t_{CAC}" requirements. Therefore the DRAMs must have a "t_{RAC}" (row access time) less than or equal to 134 ns and a "t_{CAC}" (column access time) less than or equal to 77 ns to be used in this system, under worst case conditions, for a no WAIT state, 8 MHz 80286 system. Common 120 ns 256k DRAMs meet this specification.

Other Options

In the system block diagram, buffers (74AS244s) were used to isolate the data in from the data out of the DRAM. This is needed because all WRITE accesses are late WRITEs (READ-MODIFY-WRITE cycles). In this system a HIGH and LOW WRITE enable were produced. The user could just as well have produced a HIGH and LOW CAS. In producing a HIGH and LOW CAS, the user would need the $\overline{\text{WRITE}}$ output of PAL #1 (to bring CAS low during a WRITE), A0 and $\overline{\text{BHE}}$ (for byte WRITEs), and the DT/$\overline{\text{R}}$ signal (for determining whether the access is a READ or WRITE access). Also, by generating a HIGH and LOW CAS, the system can use transceivers instead of buffers in the DRAM data path. The only problem with this approach is that $\overline{\text{RASIN}}$ to $\overline{\text{CAS}}$ out may take a little longer since $\overline{\text{CAS}}$ goes through some external logic.

80286 PAL Inputs and Outputs

(Pin number of the PAL on the left)

PAL #1 INPUTS

1) "$\overline{CLK}$" This is the inverted system clock ("CLK") of the 82284 clock generator.

2) "$\overline{CS}$" This is the latched chip select (see system block diagram).

3) "$\overline{RFRQ}$" Refresh request from the DP8419.

4) "$\overline{S0}$" Status pin from the 80286.

5) "$\overline{S1}$" Status pin from the 80286.

6) "ALE" Address latch enable from 82288.

7) "NC1" No contact.

8) "$\overline{PRECH}$" This signal indicates that a back-to-back access cycle, to the same DRAM bank, is taking place. In this situation, the PAL controller will delay "$\overline{RASIN}$" until adequate $\overline{RAS}$ precharge time has been guaranteed, and also insert WAIT states into the present access cycle to accommodate the extra precharge time.

9) "NC2" No contact.

11) "$\overline{OE}$" This input enables the PAL outputs.

PAL #1 OUTPUTS

19) "$\overline{CYREQ}$" This signal indicates that an access was requested during a Refresh or during the precharge time of the previous access.

18) "$\overline{RFREQ}$" This output guarantees that the refresh request occurs within 15 ns after the system clock. This is necessary in order for the refresh/access arbitration to work correctly.

17) "$\overline{RASIN}$" This signal causes $\overline{RAS}$ (or $\overline{RAS}$s) to go low during a DRAM access or refresh.

16) "$\overline{RFSH}$" This signal initiates a DRAM Refresh.

15) "$\overline{1DLY}$" A delay that is used internally.

14) "$\overline{2DLY}$" A delay that is used internally.

13) "$\overline{SYNRDY}$" This output goes to the 82284 clock generator synchronous ready input. This output inserts WAIT states into DRAM access cycles during access/refresh/RAS precharge contention.

12) "$\overline{WRITE}$" This output produces a WRITE strobe for the DRAMs.

PAL #2 INPUTS

1) "ALE" This is the address latch enable input from the 82288.

2) "B0" This is the "A1" address bit from the 80286.

3) "B1" This is the "A2" address bit from the 80286.

4) "$\overline{RASIN}$" This signal causes $\overline{RAS}$ (or $\overline{RAS}$s) to go low during a DRAM access or refresh.

5) "$\overline{1DLY}$" This is a delay used internal to the PALs.)

6) "$\overline{RFSH}$" This signal initiates a DRAM Refresh.

7) "$\overline{WRITE}$" This output produces a WRITE strobe for the DRAMs.

8) "A0" This is the "A0" address bit from the 80286 and is used during byte read or byte write situations.

9) "$\overline{BHE}$" This is the high byte enable signal from the 80286 and is used during byte read or byte write situations.

11) "$\overline{OE}$" This input enables the PAL outputs.

PAL #2 OUTPUTS

19) "$\overline{PRECH}$" This signal indicates that a back-to-back access cycle, to the same DRAM bank, is taking place. In this situation, the PAL controller will delay "$\overline{RASIN}$" until adequate $\overline{RAS}$ precharge time has been guaranteed, and also insert WAIT states into the present access cycle to accommodate the extra precharge time.

18) "NC" No contact to this pin.

17) "$\overline{PREV0}$" Latches if the previous access was to Bank 0.

16) "$\overline{PREV1}$" Latches if the previous access was to Bank 1.

15) "$\overline{PREV2}$" Latches if the previous access was to Bank 2.

14) "$\overline{PREV3}$" Latches if the previous access was to Bank 3.

13) "$\overline{WINLOW}$" This is the low byte DRAM write input.

12) "$\overline{WINHIGH}$" This is the high byte DRAM write input.

Equations for PALs to Interface the DP8419 to the 80286

These PALs work up to 10 MHz and use mode 0 for doing externally controlled forced refreshes, guaranteeing 3 periods (of 2X clock from 82284) of RGCK RAS pulse width. This set of PALs will only work for the DP8419 since they use mode 0 forced refresh to reset the refresh request (RFIO) signal.

```
PAL #1
DMPAL16R6A
/CLK  /CS  /RFRQ  /S0  /S1  ALE  NC1  /PRECH  NC2  GND
/OE  /WRITE  /SYNRDY  /2DLY  /1DLY  /RFSH  /RASIN  /RFREQ  /CYREQ  VCC

IF (VCC) /CYREQ = /CS*/RFSH*S0*/S1                ;Read access during RFSH
           +/CS*/RFSH*/S0*S1                      ;Write access during RFSH
           +/CYREQ*/RFSH                          ;Hold "/CYREQ" during RFSH
           +/CYREQ*RASIN                          ;Hold "/CYREQ"
           +/CYREQ*1DLY                           ;Hold "/CYREQ"
           +/CS*RASIN*1DLY*/2DLY*S0*/S1           ;Precharge needed during access
           +/CS*RASIN*1DLY*/2DLY*/S0*S1           ;Precharge needed during access

IF (VCC) /WRITE = /S0*S1*/CS*ALE                  ;Write access
           +/WRITE*1DLY                           ;Hold "/WRITE" low
           +/WRITE*/RFSH                          ;Hold "/WRITE" low
           +/WRITE*/RASIN                         ;Hold "/WRITE" low

/RFREQ := /RFRQ

/RASIN := /RFSH*/1DLY*/ALE                        ;RFSH "/RASIN" except if "ALE"
  +/RFSH*/1DLY*/RASIN                             ;Keep "/RASIN" low during RFSH
  +RFSH*/S0*S1*PRECH*RFREQ*/ALE*/CS               ;WRITE access
  +RFSH*S0*/S1*PRECH*RFREQ*/ALE*/CS               ;READ access
  +/RASIN*RFSH*2DLY                               ;Hold "/RASIN" low
  +RFSH*/CYREQ*1DLY*2DLY*/ALE                     ;"/RASIN" after precharge delay
                                                  ;  or RFSH

/RFSH := /RFREQ*RASIN*1DLY*/2DLY                  ;Start RFSH after access
  +/RFREQ*RASIN*/1DLY*/2DLY                       ;Start RFSH after access
  +/RFREQ*RASIN*1DLY*2DLY*CYREQ                   ;Start RFSH after idle states
  +/RFSH*/1DLY                                    ;Hold RFSH low
  +/RFSH*/2DLY                                    ;Hold RFSH low
  +/RFSH*/RFREQ                                   ;Hold RFSH low

/1DLY := /RFSH*2DLY*/RFREQ                        ;"/1DLY" during RFSH
  +/RFSH*/RASIN*2DLY*/1DLY                        ;Hold "/1DLY" low
  +/RFSH*/RASIN*/1DLY*ALE                         ;Hold "/1DLY" low if "ALE"
  +RFSH*/RASIN                                    "/1DLY" during access

/2DLY := /RFSH*/RASIN                             ;"/2DLY" during RFSH
  +/RFSH*/2DLY*ALE                                ;Hold "/2DLY" low if "ALE"
  +RFSH*/RASIN*/1DLY                              ;"/2DLY" during access
  +RFSH*/1DLY*/2DLY*/PRECH*RASIN*RFREQ            ;Hold "/2DLY" low for precharge

/SYNRDY := /CS*/RASIN*1DLY*RFSH                   ;"/SYNRDY" during an access
```

Equations for PALs to Interface the DP8419 to the 80286 (Continued)

PAL #2

```
DMPAL16R4A
ALE  BO  B1  /RASIN  /1DLY  /RFSH  /WRITE  AO  /BHE  GND
/OE  /WINHIGH  /WINLOW  /PREV3  /PREV2  /PREV1  /PREV0  NC  /PRECH  VCC

IF (VCC) /PRECH = RFSH*/BO*/B1*/PREV0*RASIN*/1DLY  ;Need precharge during
          +RFSH*BO*/B1*/PREV1*RASIN*/1DLY           ;  present access if
          +RFSH*/BO*B1*/PREV2*RASIN*/1DLY           ;  previous access bank =
          +RFSH*BO*B1*/PREV3*RASIN*/1DLY            ;  present access bank

/PREV0 := /BO*/B1                              ;Previous access to bank 0

/PREV1 := BO*/B1                               ;Previous access to bank 1

/PREV2 := /BO*B1                               ;Previous access to bank 2

/PREV3 := BO*B1                                ;Previous access to bank 3

IF (VCC) /WINLOW = RFSH*/RASIN*/1DLY*/AO*/WRITE    ;"/WRITE" during access

IF (VCC) /WINHIGH = RFSH*/RASIN*/1DLY*/BHE*/WRITE  ;"/WRITE" during access
```

Equations for PALs to Interface the DP8409A or DP8419 to the 80286

These PALs work up to 10 MHz with the DP8419 and up to a frequency where the minimum RGCK high or low pulse width (of 82284 2X clock) is equal to or greater than 35 ns for the DP8409A. These PALs only guarantee 2 system clock periods of RAS low during refresh and 2 periods of RAS precharge time (of 82284 2X clock) between consecutive accesses to the same RAS bank.

PAL #1

```
DMPAL16R6A
/CLK  /CS  /RFRQ  /S0  /S1  ALE  NC1  /PRECH  NC2 GND
/OE  /WRITE  /SYNRDY  /2DLY  /1DLY  /RFSH  /RASIN  /RFREQ  /CYREQ  VCC

IF (VCC) /CYREQ = /CS*/RFSH*S0*/S1          ;Read access during RFSH
           +/CS*/RFSH*/S0*S1                ;Write access during RFSH
           +/CYREQ*/RFSH                    ;Hold "/CYREQ" during RFSH
           +/CYREQ*RASIN                    ;Hold "/CYREQ"
           +/CYREQ*1DLY                     ;Hold "/CYREQ"
           +/CS*RASIN*1DLY*/2DLY*S0*/S1     ;Precharge needed during access
           +/CS*RASIN*1DLY*/2DLY*/S0*S1     ;Precharge needed during access

IF (VCC) /WRITE = /S0*S1*/CS*ALE            ;Write access
           +/WRITE*1DLY                     ;Hold "/WRITE" low
           +/WRITE*/RFSH                    ;Hold "/WRITE" low
           +/WRITE*/RASIN                   ;Hold "/WRITE" low

/RFREQ := /RFRQ

/RASIN := RFSH*/S0*S1*PRECH*RFREQ*/ALE*/CS      ;WRITE access
     +RFSH*S0*/S1*PRECH*RFREQ*/ALE*/CS          ;READ access
     +/RASIN*RFSH*2DLY                          ;Hold "/RASIN" low
     +RFSH*/CYREQ*1DLY*2DLY*/ALE                ;"/RASIN" after precharge
                                                ;  delay or RFSH

   /RFSH := /RFREQ*RASIN*1DLY*/2DLY            ;Start RFSH after access
        +/RFREQ*RASIN*/1DLY*/2DLY               ;Start RFSH after access
        +/RFREQ*RASIN*1DLY*2DLY*CYREQ           ;Start RFSH after idle states
        +/RFSH*/1DLY                            ;Hold RFSH low
        +/RFSH*/2DLY                            ;Hold RFSH low
        +/RFSH*/RFRQ                            ;Hold RFSH low
```

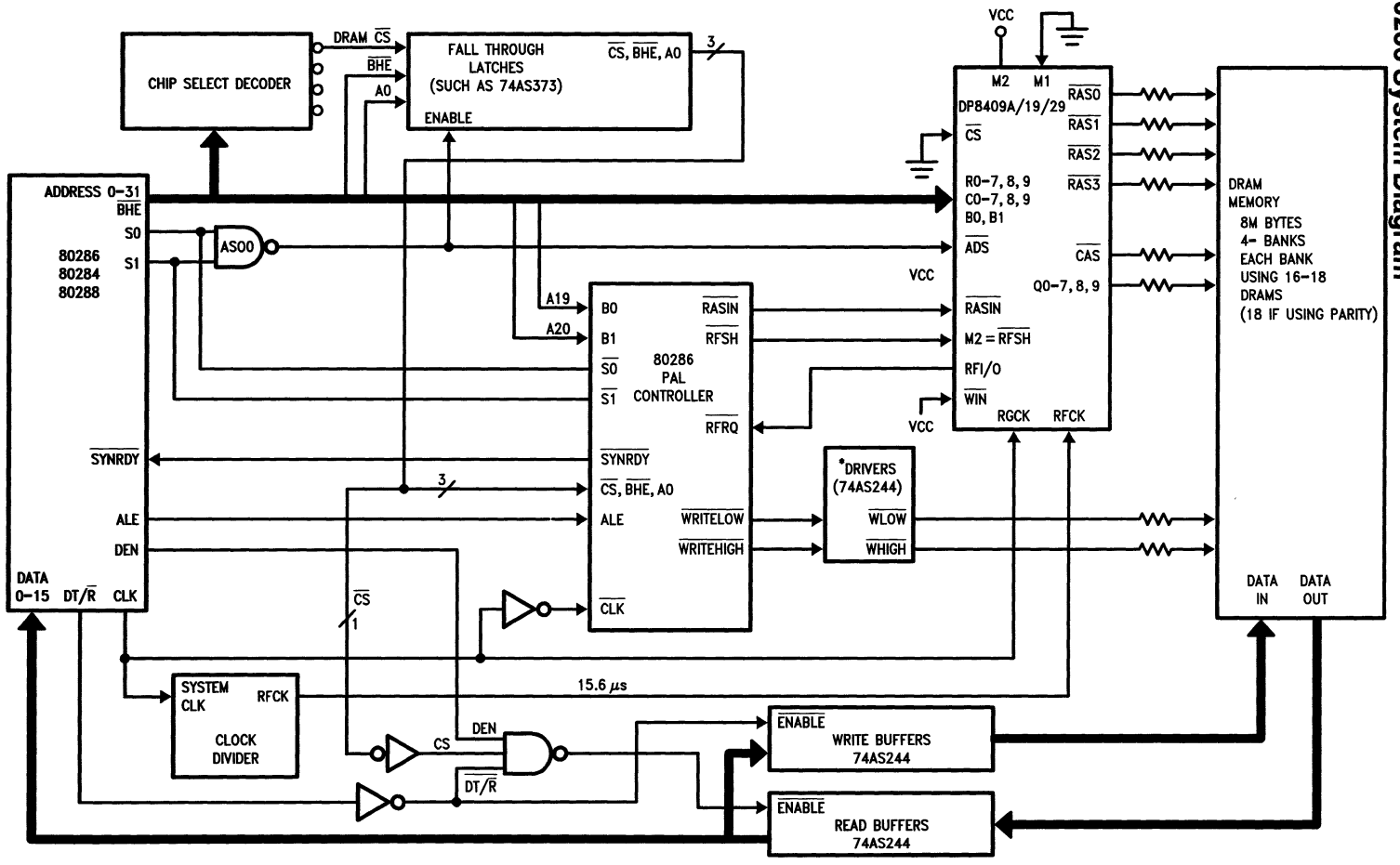
Equations for PALs to Interface the DP8409A or DP8419 to the 80286

(Continued)

```
/1DLY := /RFSH*2DLY*/RFRQ                        ;"/1DLY" during RFSH
      +/RFSH*/1DLY*2DLY                          ;Hold "/1DLY" low
      +RFSH*/RASIN                               ;"/1DLY" during access

/2DLY := /RFSH*/1DLY*/ALE                        ;"/2DLY" during RFSH
         +/RFSH*/2DLY*ALE                        ;Hold "/2DLY" low if "ALE"
         +RFSH*/RASIN*/1DLY                      ;"/2DLY" during access
         +RFSH*/1DLY*/2DLY*/PRECH*RASIN*RFREQ    ;Hold "/2DLY" low for precharge

/SYNRDY := /CS*/RASIN*1DLY*RFSH                  ;"/SYNRDY" during an access
```

If only 2 banks of DRAM were to be used the PAL interface would require only 1 PAL. The two inputs "$\overline{\text{PRECHOUT}}$ and NC" (pin #8 and #9) could be changed to "B0" and "$\overline{\text{PREVB0}}$" to allow interleaving of bank 0 and 1. "$\overline{\text{PREVB0}}$" could be produced externally using a "D" type flip-flop with "ALE" as its clock and "B0" as its input. The equations for "$\overline{\text{RASIN}}$" and "$\overline{\text{2DLY}}$" will have to be changed as follows:

```
/RASIN := /RFSH*/1DLY*/ALE                    ;RFSH "/RASIN"
  +/RFSH*/1DLY*/RASIN                         ;Hold "/RASIN" low in RFSH
  +RFSH*/S0*S1*B0*/PREVB0*RFREQ*/ALE*/CS      ;WRITE access
  +RFSH*/S0*S1*/B0*PREVB0*RFREQ*/ALE*/CS      ;WRITE access
  +RFSH*S0*/S1*B0*/PREVB0*RFREQ*/ALE*/CS      ;READ access
  +RFSH*S0*/S1*/B0*PREVB0*RFREQ*/ALE*/CS      ;READ access
  +/RASIN*RFSH*2DLY                           ;Hold ""/RASIN'' low
  +RFSH*/CYREQ*1DLY*2DLY*/ALE                 ;"/RASIN" after precharge
                                              ;  delay or RFSH

/2DLY := /RFSH*/1DLY                             ;"/2DLY" during RFSH
   +/RFSH*/2DLY*ALE                              ;Hold "/2DLY" low if "ALE"
   +RFSH*/RASIN*/1DLY                            ;"/2DLY" during access
   +RFSH*/1DLY*/2DLY*/B0*/PREVB0*RASIN*RFREQ     ;Hold "/2DLY" low for precharge
```



*Drivers may not be needed depending on the load being driven.

TL/F/8590–1

FIGURE 1

80286 System Diagram (Continued)

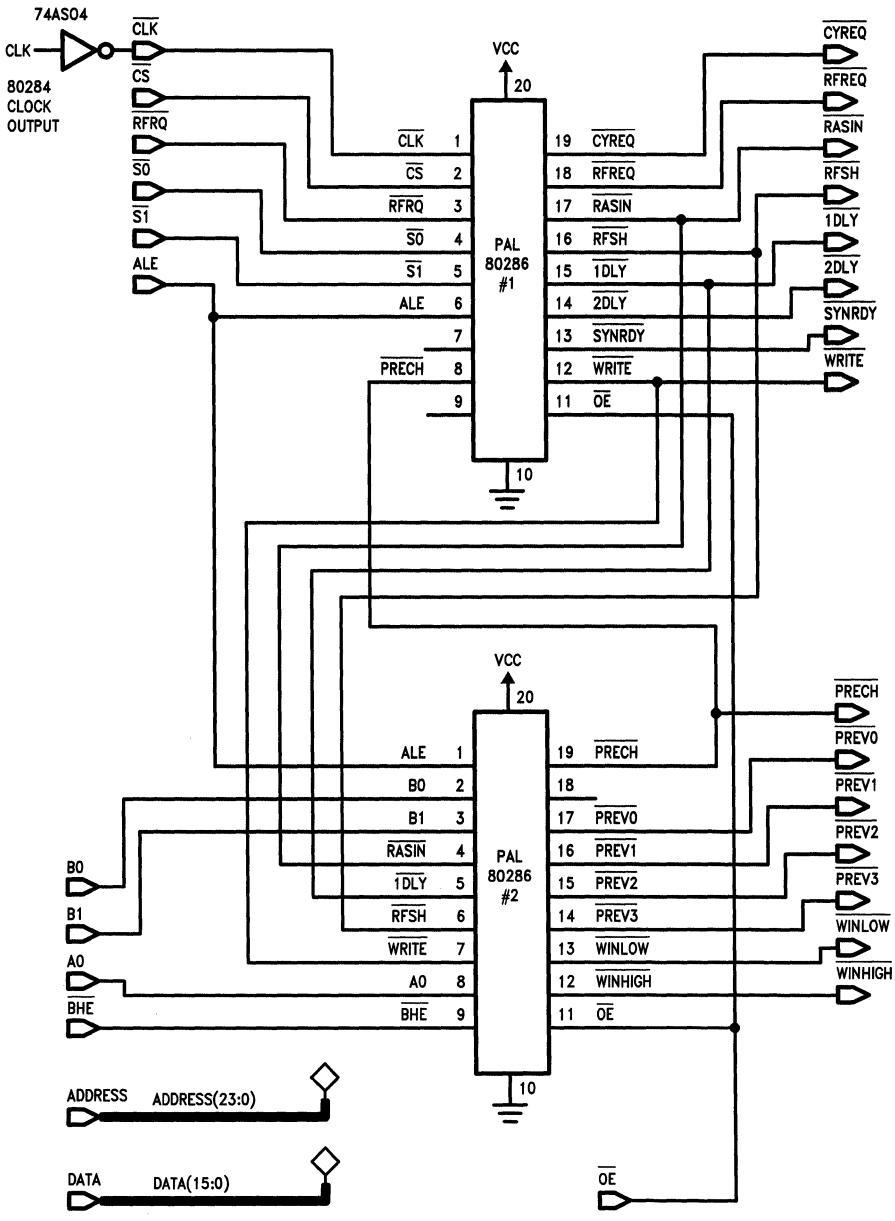
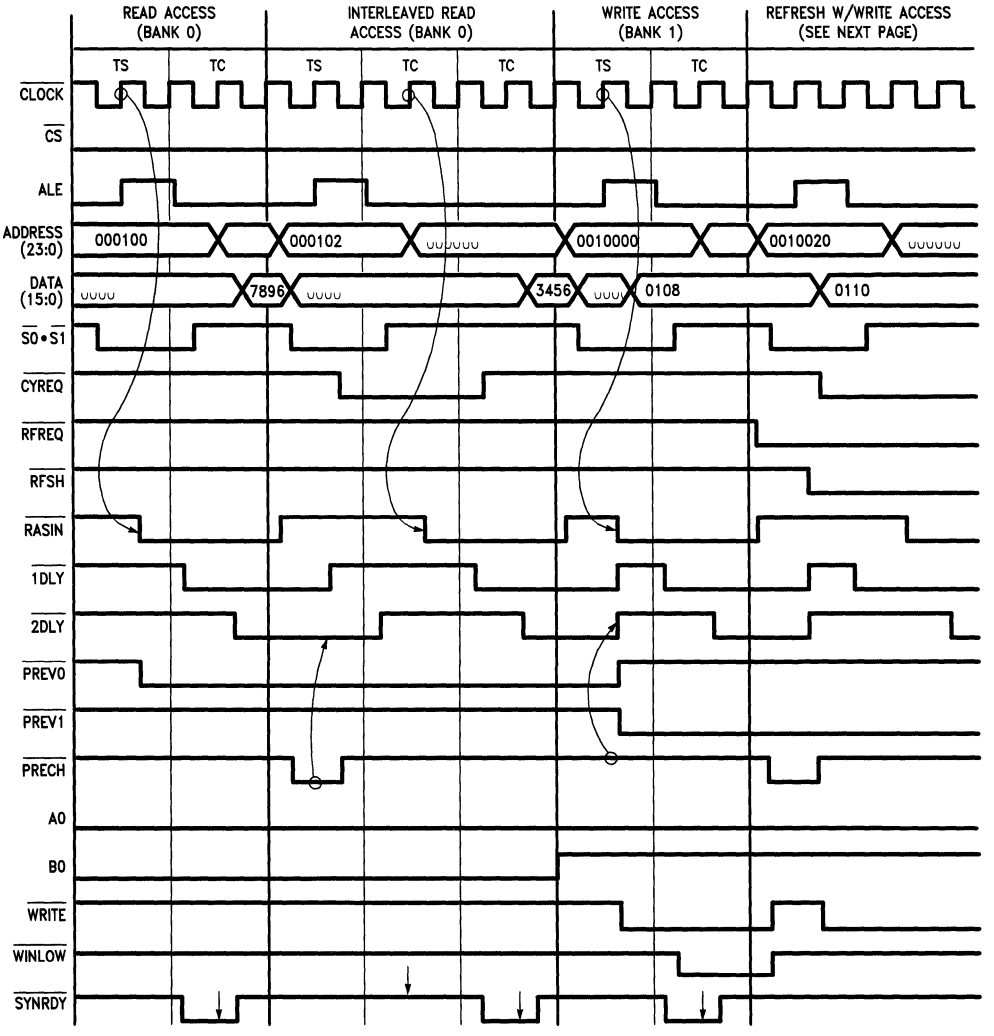


FIGURE 2. 80286 PALs

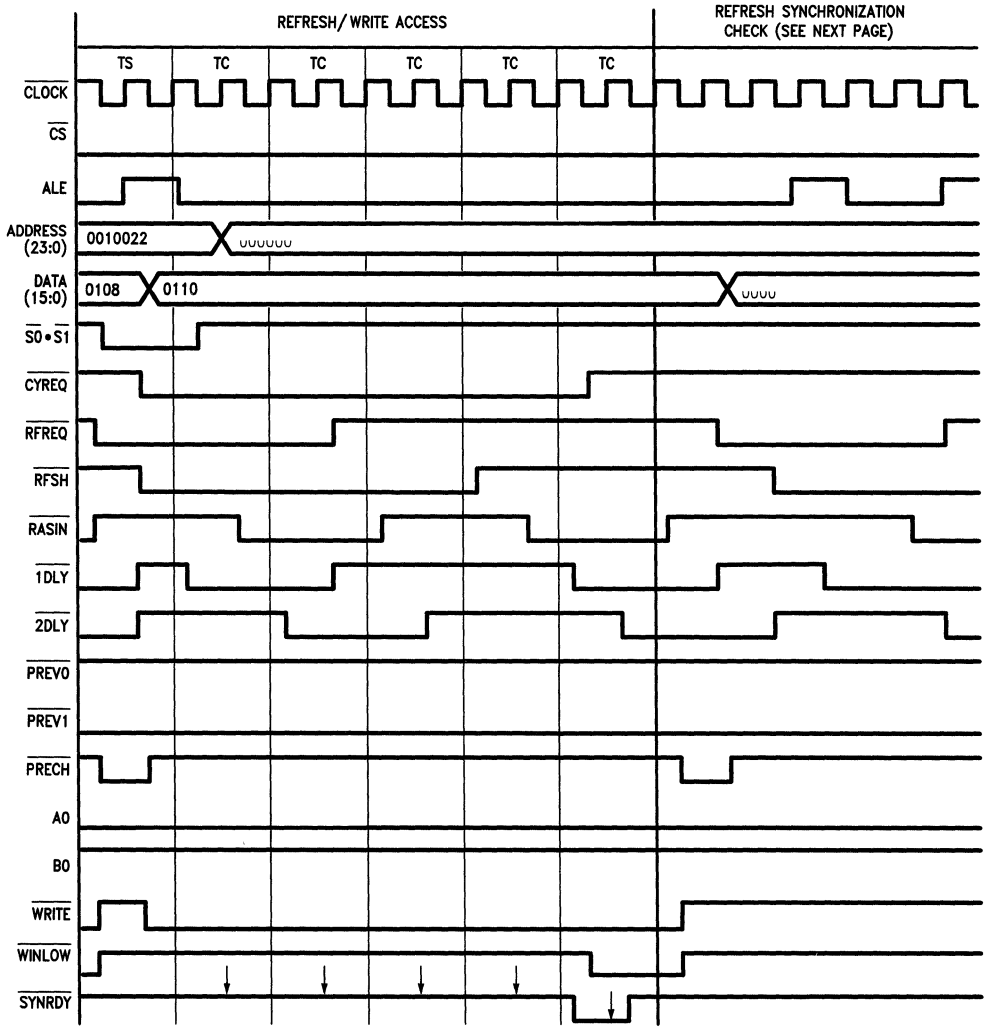
System Timing Diagram



TL/F/8590–3

FIGURE 3

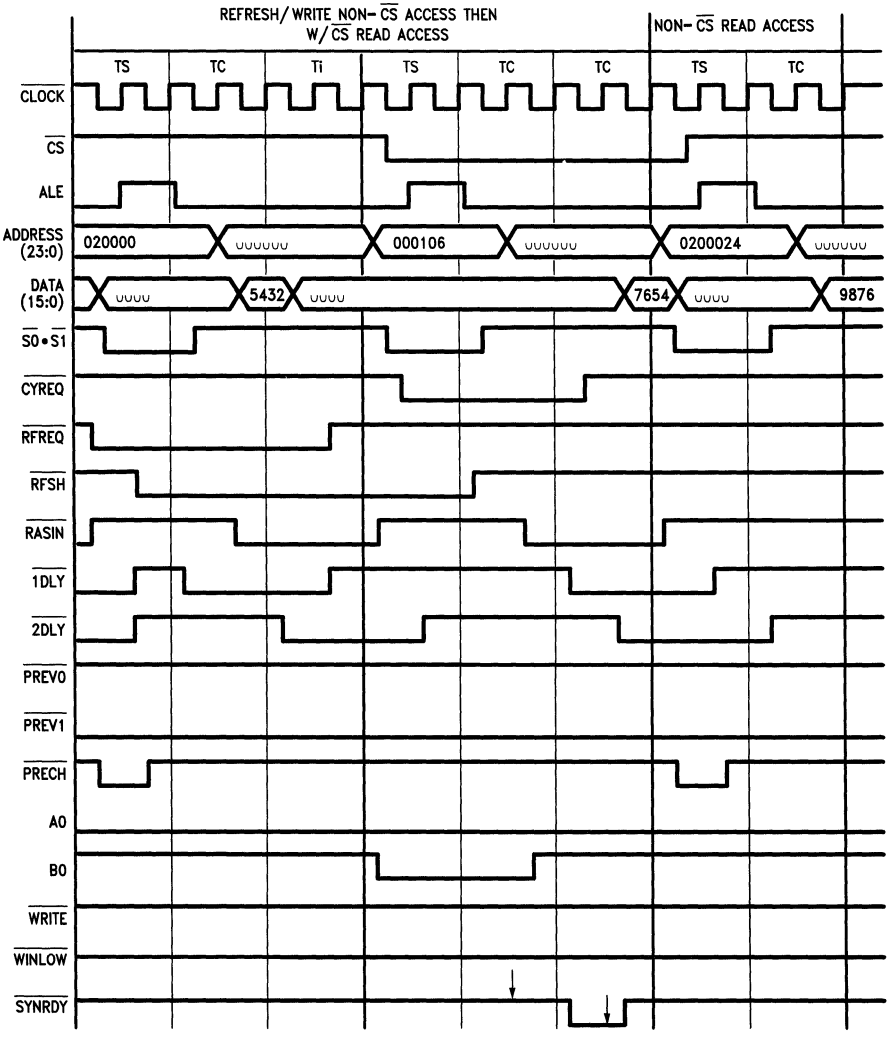
System Timing Diagram (Continued)



TL/F/8590–4

FIGURE 4

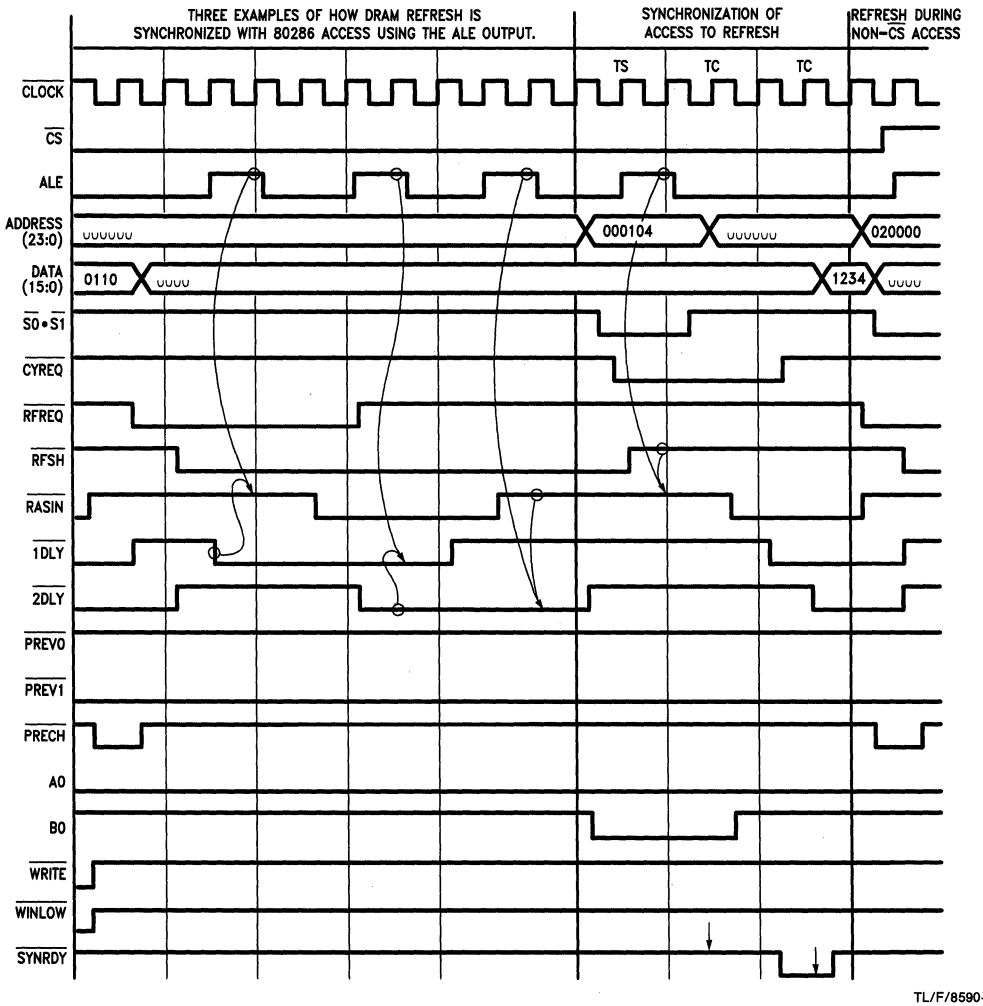
System Timing Diagram (Continued)



TL/F/8590–5

FIGURE 5

System Timing Diagram (Continued)



TL/F/8590–6

FIGURE 6

Interfacing the DP8408A/09A To Various Microprocessors

National Semiconductor
Application Note 309
Chuck Pham, Webster (Rusty) Meier

High storage density and low cost have made dynamic RAMs the designer's choice in most memory applications. However, the major drawback of dynamic RAMs is the complex timing involved. First, a $\overline{RAS}$ must occur with the row address previously set up on the multiplexed address bus. After the row address has been held for some minimum time after $\overline{RAS}$ (namely the row address hold time of the dynamic RAMs, t_{RAH}), the column address is set up and then $\overline{CAS}$ occurs. In addition, refreshing must be done periodically to keep all memory cells charged.

With the introduction of the DP8408A Dynamic RAM Controller/Driver, the above complexities are simplified. The DP8408A is housed in a 48-pin package with eight multiplexed address outputs (Q0–7) and six control outputs ($\overline{RAS}$0–3, $\overline{CAS}$, $\overline{WE}$). It consists of two 8-bit address latches and an 8-bit refresh counter. All the output drivers are capable of driving 500 pF loads.

The following discussion demonstrates a typical application of the DP8408A Dynamic RAM Controller/Driver in Z8000™- and Z80®-based systems. The DP8408A basically has six modes of operation: Externally Controlled Refresh, Externally Controlled All-$\overline{RAS}$ Write, Externally Controlled Access, Auto Access (slow t_{RAH}), Auto Access (fast t_{RAH}) and Set End of Count.

The DP8408A, operating in the auto access mode, requires only $\overline{RASIN}$ to initiate a memory access cycle because all the dynamic RAM's control signals are automatically delayed from this input. (Refer to *Figure 1* for the auto access timing sequence.)

In the following applications, the DP8408A operates in either mode 5 or mode 6 Auto Access and mode 1 or 2 Externally Controlled Refresh to provide minimum additional logic.

The DP8408A and Z8000 Interface

MEMORY ACCESS CYCLE

Figure 2a shows the detailed block diagram of Z8000 and the DP8408A interface. Consider a memory cycle of the Z8000; first, the memory address is output on the Address and Data multiplexed bus (AD0–15) during T1 and is latched to the DP8408A by $\overline{AS}$. Simultaneously, $\overline{MREQ}$ goes low and is used to provide $\overline{RASIN}$ to initiate a memory transaction cycle. Then the selected $\overline{RAS}$ output, row address hold time (t_{RAH}), column address set up time (t_{ASC}) and $\overline{CAS}$ output will follow $\overline{RASIN}$ as determined by the auto access modes. A maximum of one wait state is required for 6 MHz and 10 MHz CPUs. This wait state is automatically inserted by the $\overline{CAS}$ output of the DP8408A. For systems using byte-writing, the DM74S158 provides two separate $\overline{CAS}$ outputs for accessing the low and high byte of memory. Note that $\overline{DS}$ from the Z8000 is also gated with the DP8408A's CAS output to generate $\overline{CASL}$ and $\overline{CASH}$. This guarantees the valid data from the Z8000 being written into memory during memory write cycles. Refer to *Figure 3* for the detailed memory transaction cycle timing.

The following formula allows the designer to determine the proper memory speed in terms of t_{CAC} (access time from $\overline{CAS}$):

$$t_{CAC}\ \text{max.} = 3 \times t_{cC} - \text{tdc(MR)} - t_{RICL} - t_{CASdly} - t_{sDR}(C) - 15.$$

The Z8000 parameters:

t_{cC}: clock cycle time

$t_{sDR}(C)$: read data to clock ↓ set up time

tdc(MR): clock to $\overline{MREQ}$ delay

The DP8408A, 74S158 and 74LS245 parameters:

t_{RICL}: RASIN to CAS delay

t_{CASdly}: the propagation delay of the 74S158

15 ns: the propagation delay of the 74LS245 (at 50 pF load)

For the 10 MHz CPU and the DP8408A:

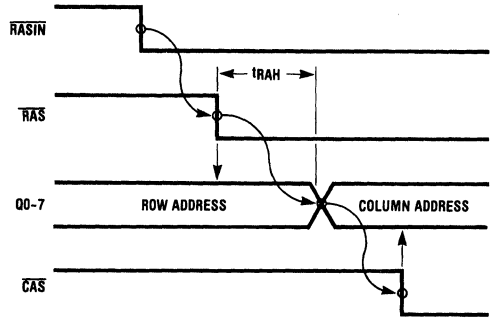
t_{CAC} max. = 300 − 40 − 131 − 14 − 10 − 15 = 90 ns.

- t_{RICL} max. (mode 6) = 131 ns at 15 pF load.
- t_{CASdly} max. = 14 ns at 50 pF load.

Since $\overline{MREQ}$ is connected directly to $\overline{RASIN}$, t_{RP} ($\overline{RAS}$ precharge time) and t_{RAS} ($\overline{RAS}$ pulse width) are determined by $\overline{MREQ}$ high and low, respectively.

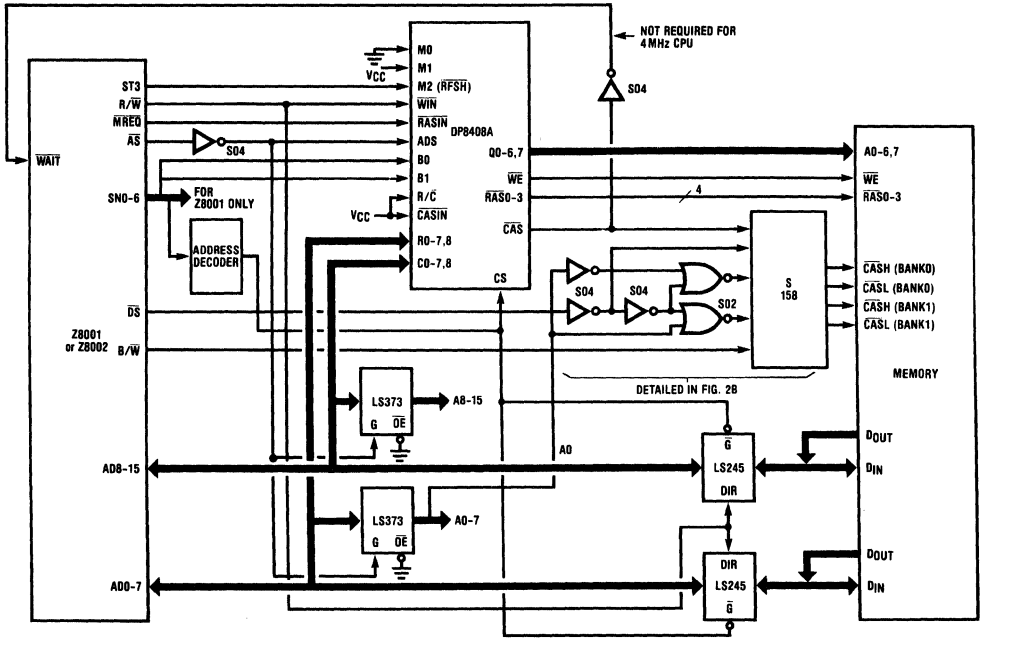
MEMORY REFRESH CYCLE

The Z8000 CPU contains a refresh rate counter for automatic memory refresh. This counter should be programmed during the processor initialization to determine the refresh rate. Since memory refresh is automatically inserted by the Z8000, there is no additional refresh arbitration logic allowed. The CPU's STATUS 3 (ST3) output can be directly connected to the M2 ($\overline{RFSH}$) pin of the DP8408A. During the memory refresh cycle, ST3 goes low, setting the DP8408A in the external control refresh mode (mode 2). Then all four $\overline{RAS}$ outputs will follow $\overline{MREQ}$ to strobe the DP8408A's refresh address to all memory banks (the Z8000 refresh address is ignored). As $\overline{MREQ}$ goes high again, the DP8408A increments its refresh counter, preparing it for the next refresh cycle. Refer to *Figure 4* for the refresh cycle timing. Note that ST3 also goes low during the internal cycle, I/O reference cycle and interrupt acknowledge cycle, but the memory will not be refreshed because $\overline{MREQ}$ is not active during these cycles. The DP8408A on-chip refresh counter will not be incremented when M2 goes low unless $\overline{MREQ}$ is inserted.



TL/F/5040–1

FIGURE 1. Auto Access Timing Sequence (Mode 5 or Mode 6)



TL/F/5040–2

FIGURE 2a. Z8000 and DP8408A Interface

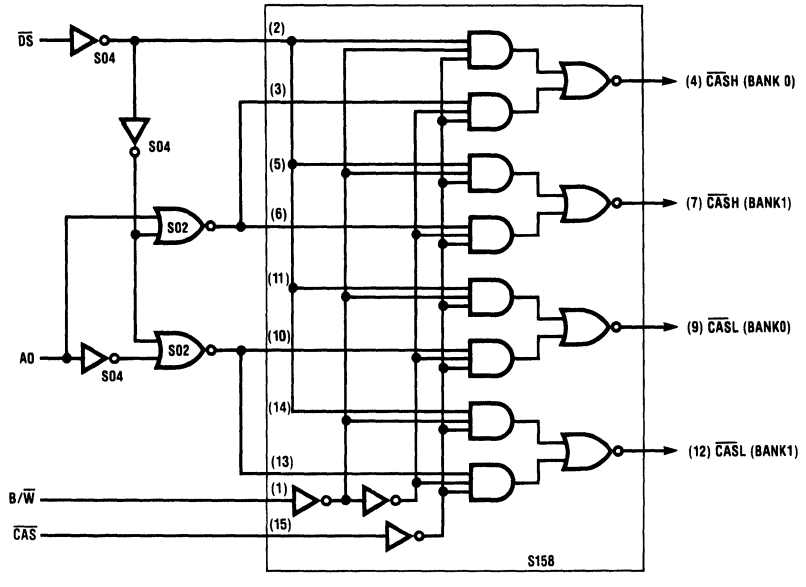


FIGURE 2b. CASH and CASL Decoder

When the processor is in either halt state (by executing the privileged HALT instruction) or single-stepping mode (when STOP) input is low), it introduces memory refresh cycles. However, care should be taken when the CPU is in either a WAIT state or a Bus Acknowledge cycle, that the dynamic RAM refresh will not take place. If these conditions occur over a long period of time, a burst refresh is recommended. This can be done by toggling RASIN while keeping M2 low, until all the rows of the dynamic RAM have been refreshed, then the CPU can resume its operations.

The DP8408A and Z80A® Interface

INSTRUCTION FETCH CYCLE

Figure 5 shows the detailed interconnections between the DP8408A, the Z80 and the Dynamic RAMs. *Figure 6* shows the timing during an M1 cycle (op code fetch). The program counter is output on the address bus at the beginning of the M1 cycle. One-half clock later MREQ goes active. This input is used to provide RASIN to the DP8408A to access the dynamic memory. Subsequently, the selected RAS output, Row to Column Select and then CAS output will automatically follow RASIN as determined by the Auto Access modes of the DP8408A. The RD line also goes active to indicate a memory read cycle is in progress. After t_{CAC} (access time from CAS), read data becomes valid. This data is sampled on the rising edge of T3, then both MREQ and RD go inactive. Immediately following this, RFSH goes low, putting the DP8408A in the Externally Controlled Refresh mode. The MREQ goes active causing all four RAS outputs to go active to perform a refresh to all the banks of the dynamic RAMs. Note that during memory refresh cycles, the refresh address from the CPU is output on the address bus. However, the contents of the DP8408A on-chip refresh counter are used instead to provide the row address to the dynamic memory array. Since the Z80 provides only a 7-bit refresh address, it is an advantage to use the DP8408A 8-bit refresh counter to support 64k dynamic RAMs directly. The DP8408A refresh counter is incremented as MREQ returns high, ending the memory refresh. The RFSH goes inactive returning the DP8408A back to the Auto Access mode, preparing it for the next access cycle.

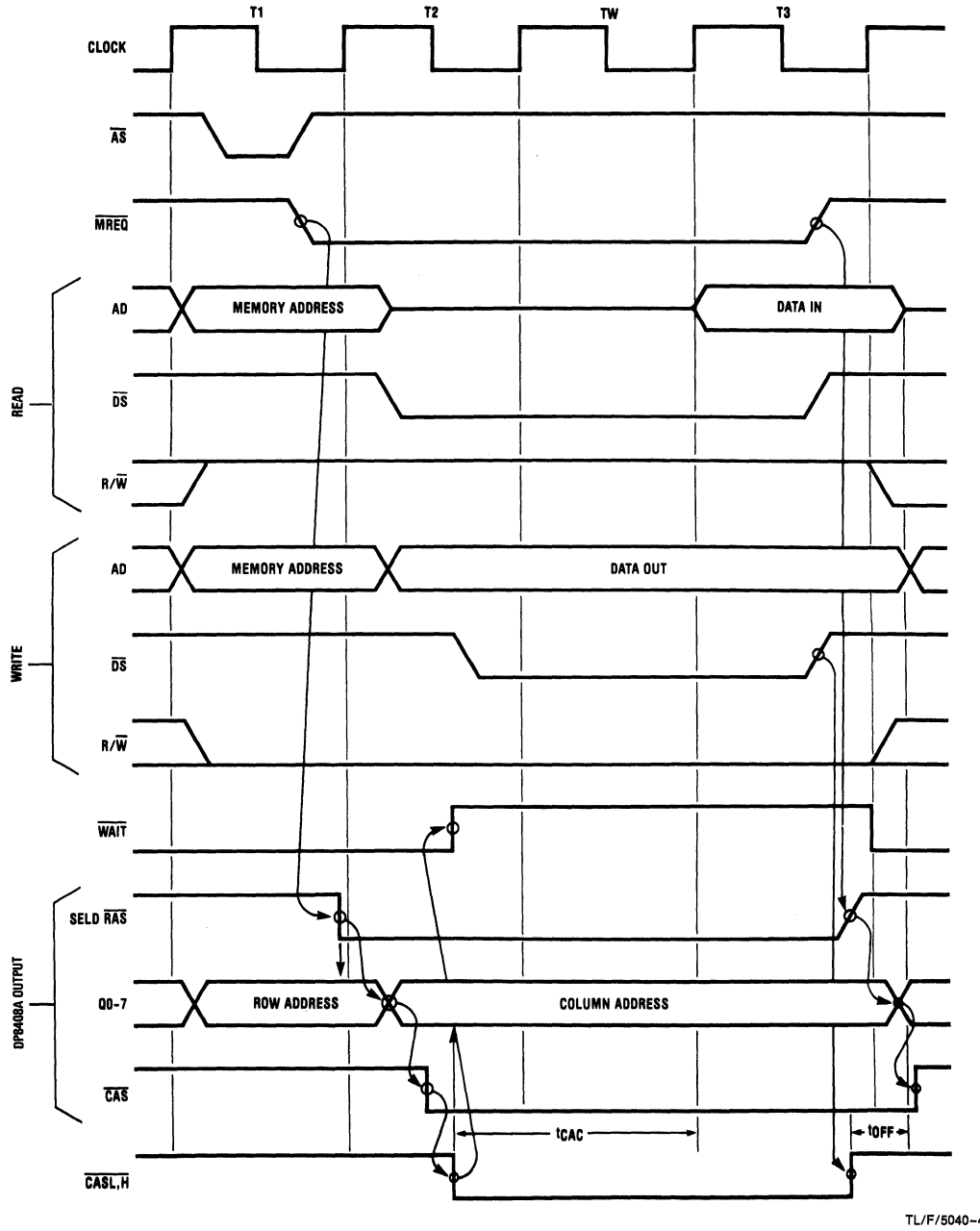
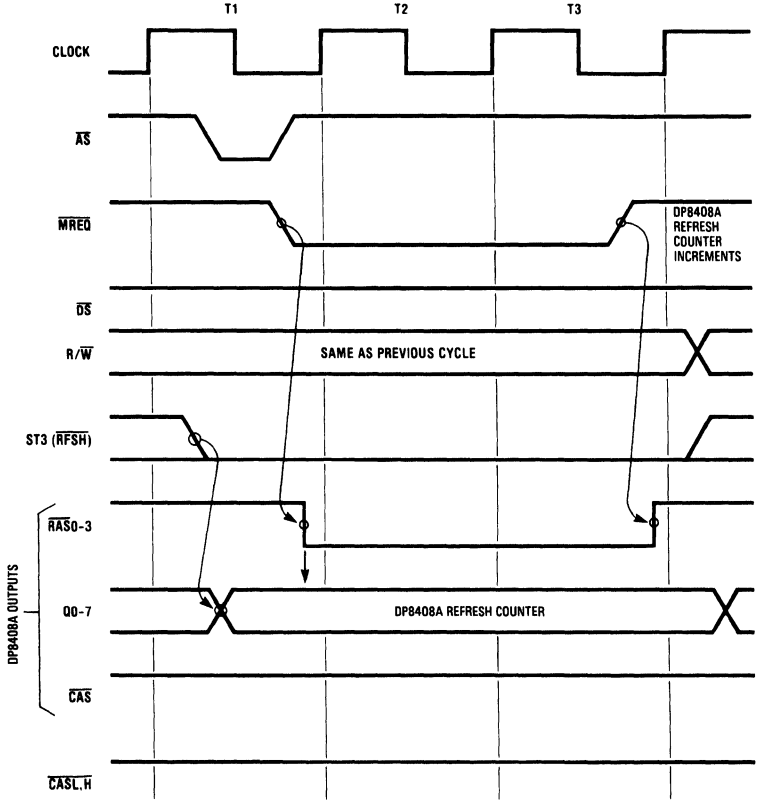
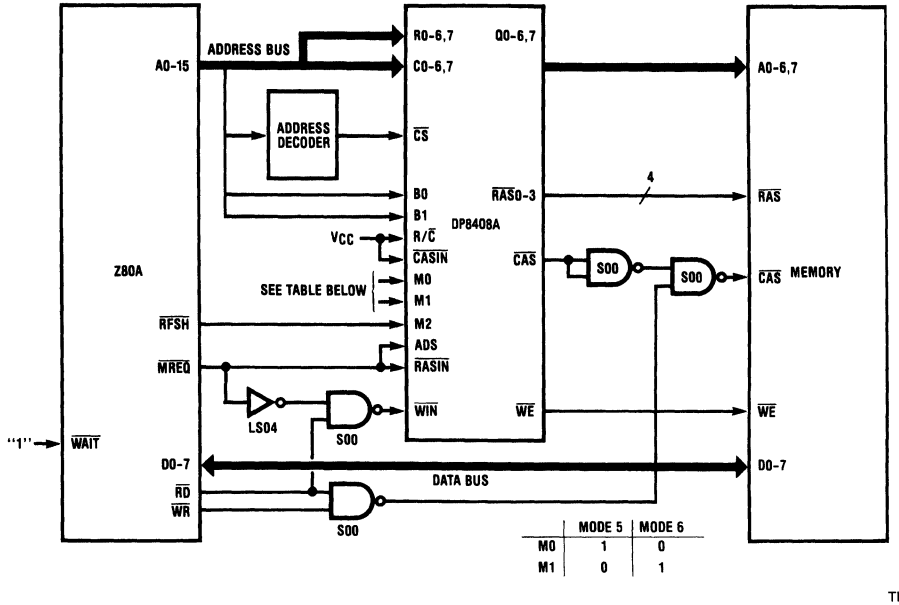


FIGURE 3. Memory Transaction Cycles



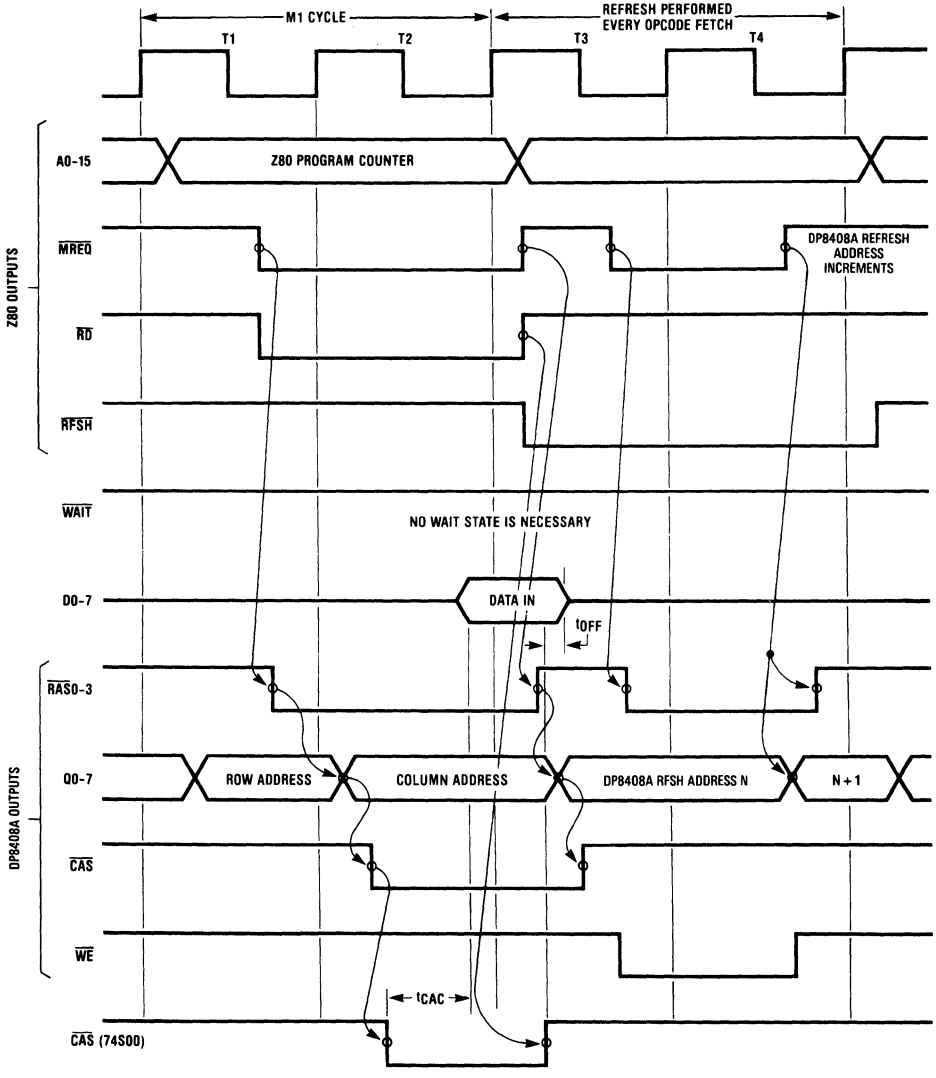
TL/F/5040-5

FIGURE 4. Memory Refresh Cycle



TL/F/5040-6

FIGURE 5. DP8408A and Z80A Interface



TL/F/5040–7

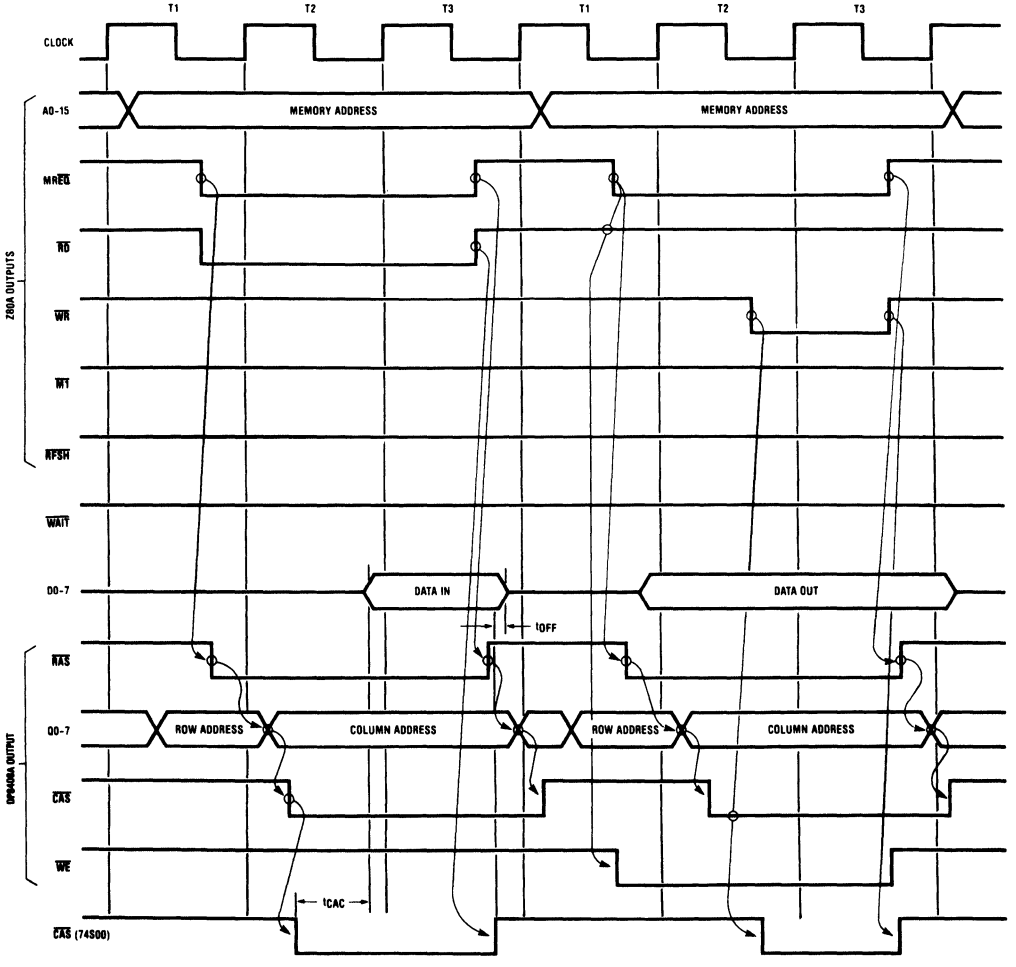
FIGURE 6. Z80A Op Code Fetch Cycle Showing Memory Refresh

MEMORY ACCESS CYCLE

Figure 7 shows the timing of the memory read and memory write cycle other than for the M1 op code fetch cycle. Similar to the op code fetch cycle, $\overline{MREQ}$ is used to provide $\overline{RASIN}$. $\overline{MREQ}$ goes active after the address to the memory has had time to stabilize. Again, $\overline{RAS}$ output, Row to Column Select and then $\overline{CAS}$ output will automatically follow $\overline{RASIN}$ to access the specified memory location. For a memory read cycle, both $\overline{MREQ}$ and $\overline{RD}$ go active, and as a result, $\overline{WIN}$ remains high (refer to *Figure 5*), which allows a memory read operation to occur. On the other hand, only $\overline{MREQ}$ goes active during a write cycle, which forces $\overline{WIN}$ low, indicating an early write cycle. It should be noted that the $\overline{CAS}$ output to the memory array will not go low until $\overline{WR}$ goes low during memory write cycles as this guarantees the valid CPU data will be written into memory.

It is worth mentioning that the Z80 CPU provides powerful block transfer instructions. An example is the LDIR (load, increment and repeat); using only this instruction, the programmer can move any block of data from the location pointed to by the D and E registers. This operation is repeated until the byte counter (B and C registers) reaches zero. Thus, this single instruction can move any block of data from one location to any other. Due to the fact that this instruction is refetched after each data byte transfer, the memory refresh cycle always takes place even though a transfer of up to 64k bytes of data may be performed. Furthermore, when the CPU has executed the software HALT instruction and is waiting for an interrupt before normal CPU operations can resume, the CPU executes NOP instructions to maintain memory refresh activity.

However, care should be taken when the CPU is in either WAIT state or a Bus Acknowledge cycle, the dynamic RAM refresh will not take place. If these conditions occur long enough, a burst refresh is recommended, and it can be done by toggling $\overline{RASIN}$ while keeping M2 low until all the rows of the dynamic RAM have been refreshed before the CPU can resume its operation.



TL/F/5040–8

FIGURE 7. Z80A Memory Read and Memory Write Cycle

The following formulas allow designers to select the appropriate dynamic memory, based on different CPU and DP8408A speed versions, to allow the CPU full speed of operation:

max. t_{CAC}: $1.5 \times t_{Cmin} - t_{DL\phi}(MR) - tRICL - t_{CASDLY} - t_{S\phi}(D)$

min. t_{RP}: $tw(MRH) = tw(\phi H) + t_f - 20$

min. t_{RAS}: $tw(MRL) - 20 = t_C - 50$

Dynamic RAM Parameters:

t_{CAC}: access time from $\overline{CAS}$

t_{RP}: $\overline{RAS}$ precharge time

t_{RAS}: $\overline{RAS}$ pulse width

Z80 Parameters:

t_C: clock period

$tw(\phi H)$: clock pulse width, clock high

tf: clock fall time

$t_{DL\phi}(MR)$: $\overline{MREQ}$ delay from falling edge of clock, $\overline{MREQ}$ low

$t_{S\phi}(D)$: Data set up time to rising edge of clock during M1 cycle

DP8408A and 74S00 Parameters:

t_{RICL}: $\overline{RASIN}$ to $\overline{CAS}$ output delay

t_{CASDLY}: propagation delay of the two 74S00 NAND gates

For example, if the Z80A (4 MHz) and the DP8408A are used, then:

max. t_{CAC}: $1.5(250) - 85 - 132 - 13 - 50 = 95$ ns

min. t_{RP}: $110 + 20 - 20 = 110$ ns

min. t_{RAS}: $t_C - 50 = 200$ ns

t_{RICL} max.

(mode 6): 132 ns at 15 pF load

t_{CASDLY} max.: 13 ns at 50 pF load

Therefore, in this case, the designer should choose a dynamic memory which has maximum t_{CAC} of 95 ns, minimum t_{RP} of 110 ns and minimum t_{RAS} of 200 ns.

DP8409A and MC68B09E Interface

DP8409A OVERVIEW

The DP8409A Dynamic RAM Controller/Driver is designed to control all multiplexed-address dynamic RAMs. It consists of two 9-bit address latches and a 9-bit refresh counter, thus allowing control of all 16k, 64k, and the coming generation 256k dynamic RAMs. More important, all the DP8409A outputs are capable of driving 500 pF loads.

The DP8409A basically has eight modes of operation: Externally Controlled Refresh, Automatic Forced Refresh, Internal Auto Burst Refresh, All $\overline{RAS}$ Auto Write, Externally Controlled Access, Auto Access (slow t_{RAH} and with hidden refresh), Fast Auto Access (fast t_{RAH}) and Set End of Count. Of all these modes, Auto Access (mode 5) and Auto Forced Refresh (mode 1) are the most popular and will be used throughout this application. Mode 5 requires only $\overline{RASIN}$ to initiate a memory access cycle, because all the dynamic RAM's control signals are automatically delayed from this input, as shown in *Figure 1*. To attain maximum system throughput, it is obviously advantageous to perform refreshes without interrupting the system. The DP8409A can do this by monitoring the $\overline{CS}$ input to see if it is high. If $\overline{CS}$ is high, the RAMs are not being accessed. If $\overline{CS}$ is high for one cycle, the DP8409A performs a hidden refresh during this cycle, and stops in time for the system to start another access. But if a hidden refresh does not occur in a specific time slot, a refresh must be forced and this can be done by using Mode 1, Automatic Forced Refresh.

To perform automatic forced refresh, the DP8409A must receive two clock signals: the refresh period clock, RFCK, and RGCK, the $\overline{RAS}$-generator clock; RGCK can be the microprocessor clock. It takes approximately four RGCK clock periods to perform this automatic forced refresh. The DP8409A gives preference to hidden refresh using RFCK as a level reference. The refresh time slot commences as RFCK goes high. If $\overline{CS}$ goes high while RFCK is high, the refresh counter is enabled in the address outputs. All four $\overline{RAS}$ outputs follow $\overline{RASIN}$; so to perform a hidden refresh, $\overline{RASIN}$ must be set low and the refresh counter gets incremented as $\overline{RASIN}$ goes high. The DP8409A allows only one such hidden refresh to occur with a clock cycle of RFCK to minimize power consumption.

If a hidden refresh does not occur the DP8409A must force a refresh before RFCK begins a new cycle on a low-to-high transition. Therefore, as RFCK goes low (and a hidden refresh has not occurred), RF I/O (Refresh Request) goes low, requesting that a refresh be performed. When the system acknowledges the request, it sets M2 low, and prevents further access to the DP8409A. Then two RGCK negative edges after M2 has gone low, all four $\overline{RAS}$ outputs go low and remain low for two RGCK clock periods. After all four $\overline{RAS}$ outputs have gone low, M2 can go high any time to end the Automatic Forced Refresh. The DP8409A allows only one automatic refresh to occur within a clock cycle of RFCK.

MEMORY ACCESS

The MC68B09E starts a memory access cycle when E goes low, then the memory address becomes valid on the Address Bus A0–15. This address is decoded to provide Chip Select ($\overline{CS}$) to the DP8409A. Then Q goes high and sets $\overline{RASIN}$ low from the PAL® Control Logic as shown in *Figure 12*. Note that $\overline{CS}$ must go low for a minimum of 10 ns before the assertion of $\overline{RASIN}$ for a proper memory access. This is important because a false hidden refresh may take place when this 10 ns minimum setup time is not met. $\overline{RASIN}$ goes low initiating the auto access sequence as shown in *Figure 1*. Mode 5 guarantees a 30 ns minimum for row address hold time and a minimum of 8 ns column address set up time. $\overline{RASIN}$ remains low until E goes low at the end of the current access cycle. Using the 16R6A Programmable Array Logic (25 ns PAL), the maximum access time from $\overline{CAS}$ of the selected dynamic RAM is determined as follows:

Max. t_{CAC}: $3 \times 125 - 25 - 160 - 40 = 150$ ns 8409A

t_{CAC}: $3 \times 125 - 25 - 130 - 40 = 180$ ns 8409A-2

Q high to

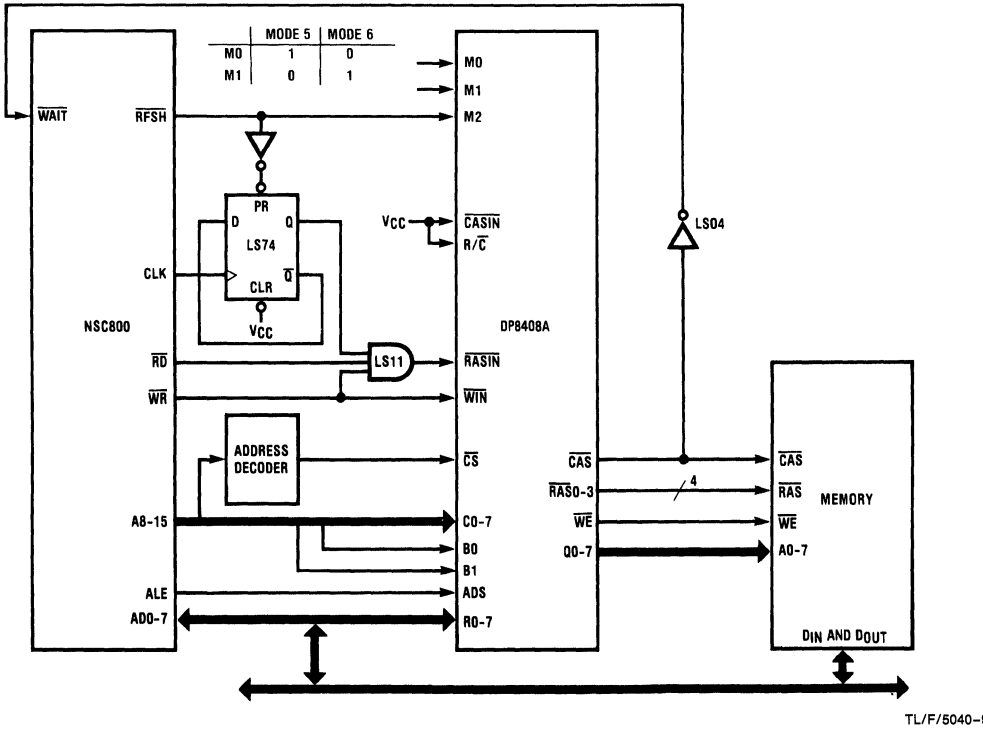
E low: 3×125 ns (8 MHz clock) $= 375$ ns

Q high to
$\overline{RASIN}$ low: 25 ns (16R6 A PAL Parameter)
$\overline{RASIN}$ to $\overline{CAS}$
Output low: 160 ns (DP8409A's t_{RICL}, Mode 5, at 500 pF load)
130 ns (DP8409A-2's t_{RICL})
Read data setup time (before E going low): 40 ns

MEMORY REFRESH

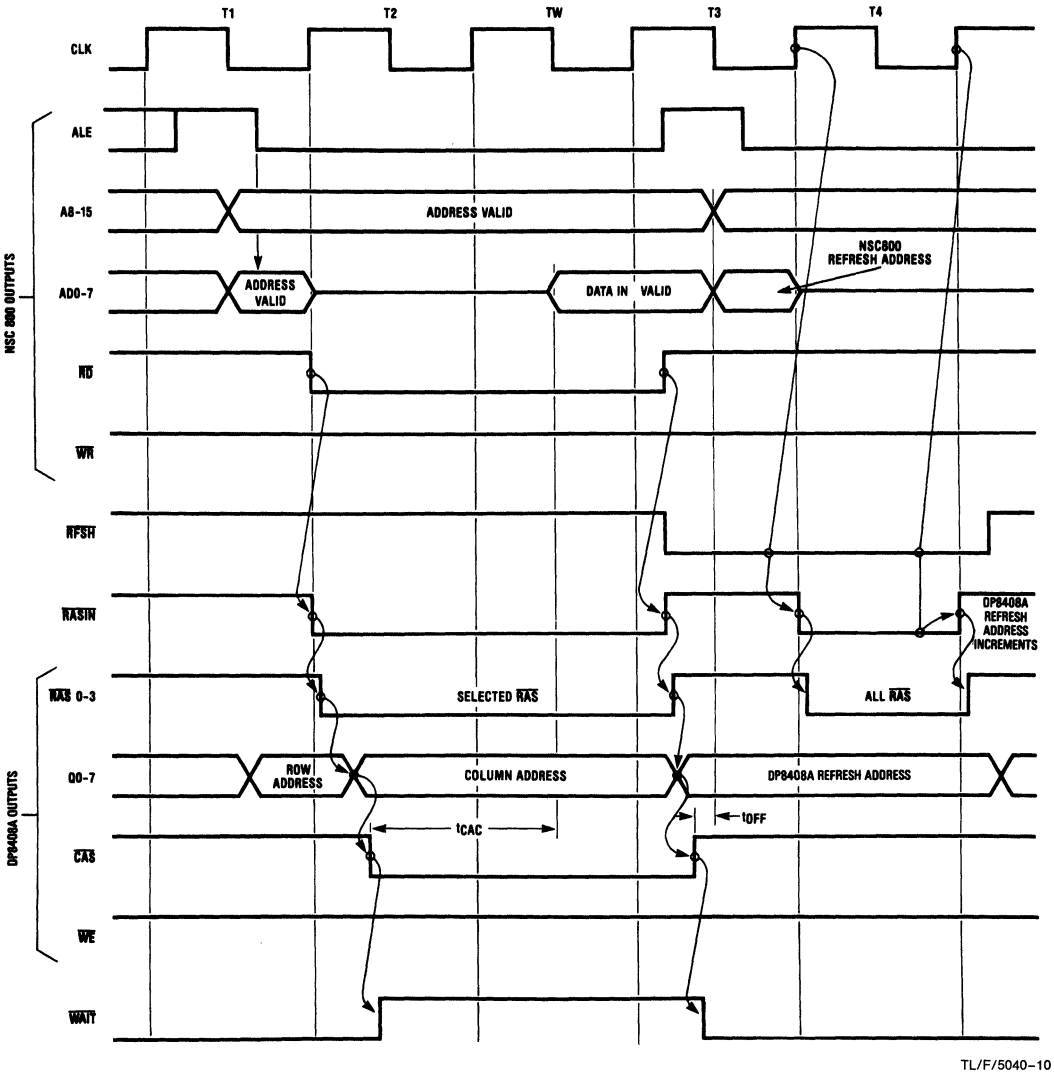
As described above, $\overline{RASIN}$ goes active when Q and/or E are high. This scheme, therefore, maximizes chances for hidden refresh because $\overline{CS}$ is high during nondynamic memory cycle. For example, when the CPU is executing internal operation or the CPU is accessing ROM or I/O, $\overline{CS}$ is high during these times. The DP8409A therefore performs a hidden refresh as $\overline{RASIN}$ goes low, assuming that RFCK is high.

However, if no hidden refresh occurs while RFCK was high, RF I/O goes low immediately after the RFCK high-to-low transition requests a forced refresh. The PAL Control Logic samples RF I/O, when E and Q are high and low respectively, to set M2 (RFSH) low, as shown in *Figure 13*. Once M2 has gone low, a forced refresh automatically occurs (as described in the DP8409A Overview). M2 remains low for four system clock periods to allow for this forced refresh. If the current microprocessor cycle is a nondynamic memory cycle ($\overline{CS}$ is high), this refresh is transparent to the microprocessor and $\overline{STRETCH}$ remains high (E and Q are not stretched). Nevertheless, if the current cycle is a dynamic memory access cycle, $\overline{STRETCH}$ goes low stretching E and Q for a maximum of four system clocks. $\overline{RASIN}$ for the pending access will be issued a full system clock after M2 has gone high; this is to allow some $\overline{RAS}$ precharge time for the dynamic RAM. After this, memory will be accessed in the manner as described in the Memory Access Cycle.



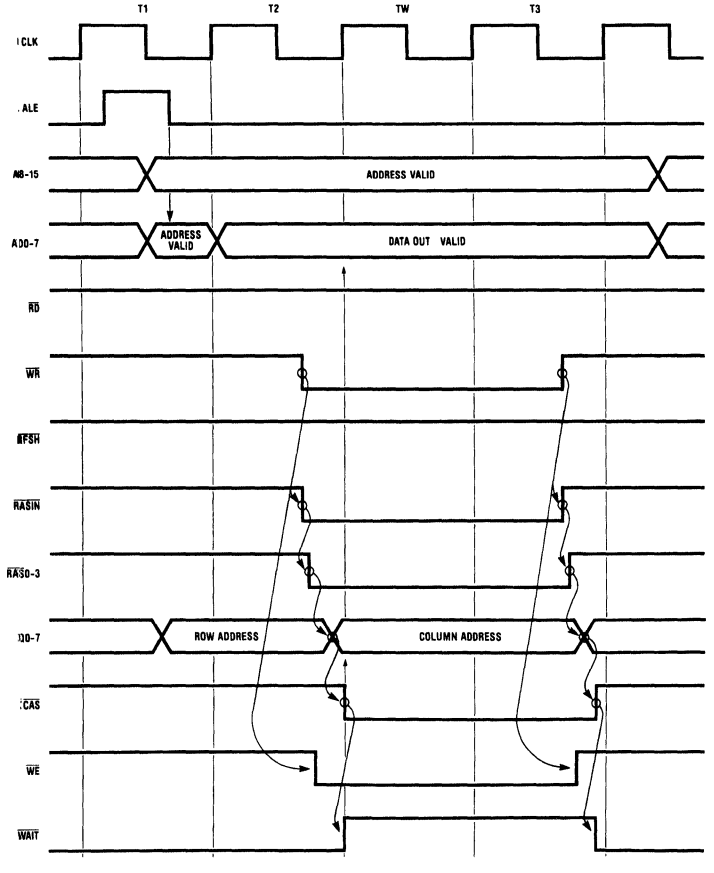
TL/F/5040–9

FIGURE 8. NSC800 and DP8408A Interface



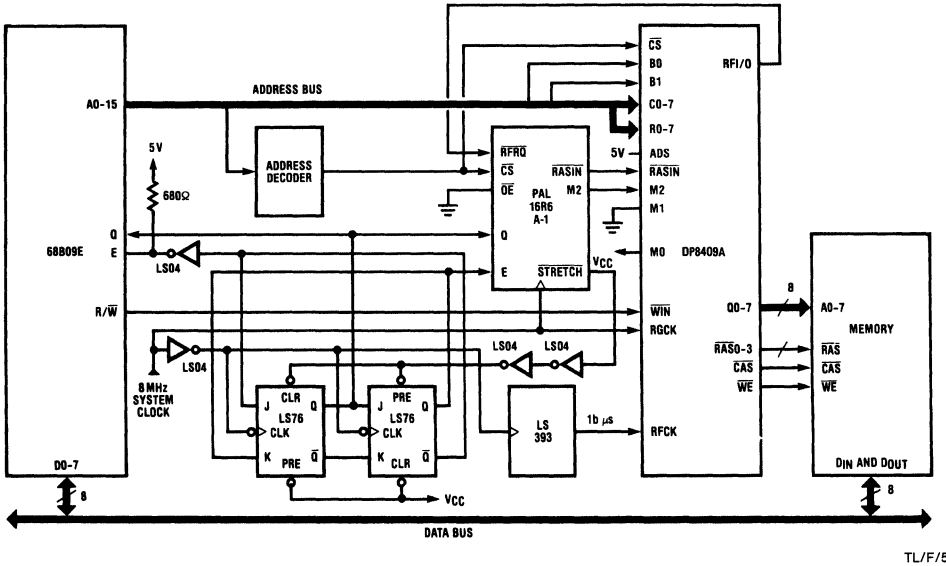
TL/F/5040–10

FIGURE 9. NSC800 Op Code Fetch Cycle Showing Memory Refresh



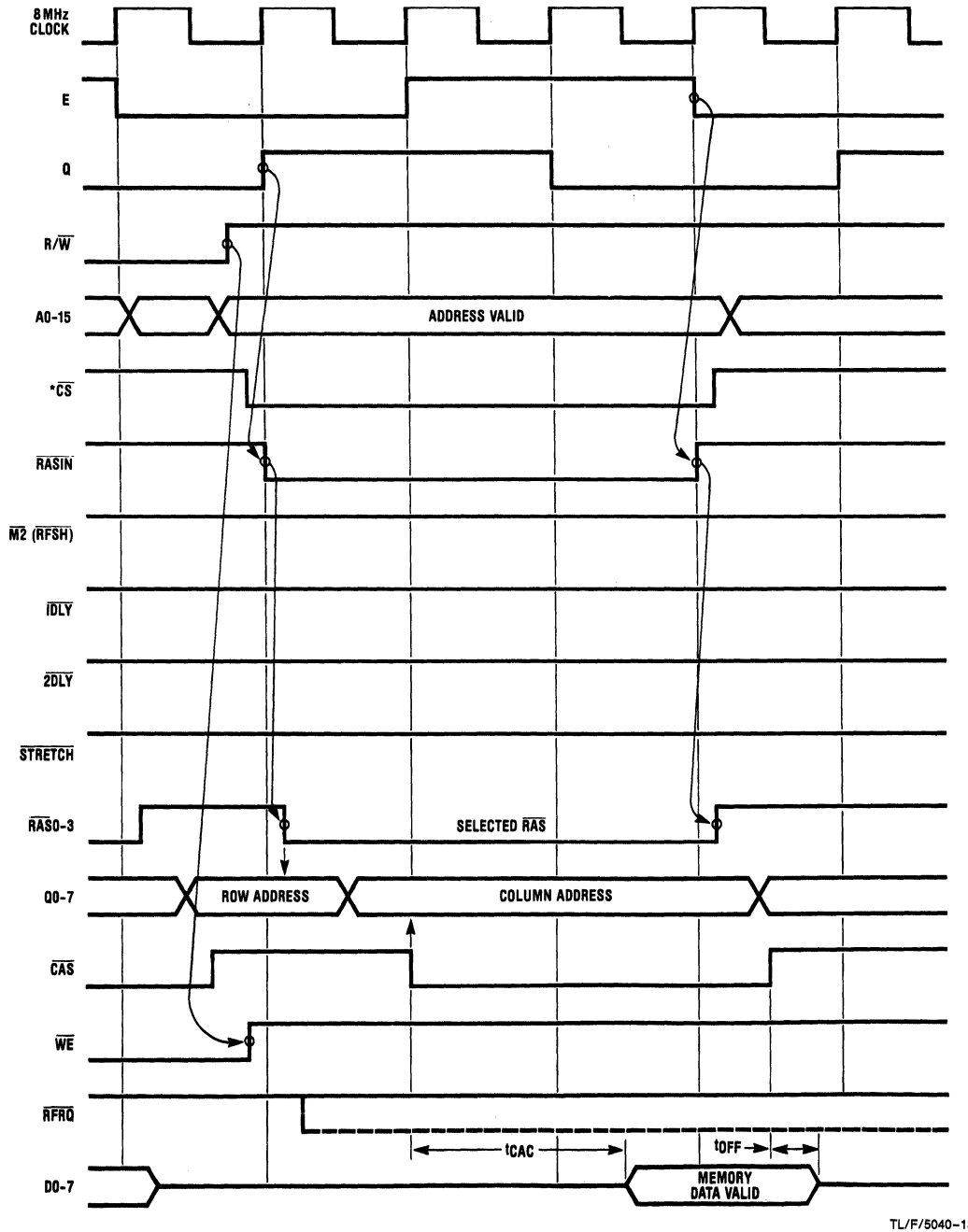
TL/F/5040–11

FIGURE 10. NSC800 Memory Write Cycle



TL/F/5040–12

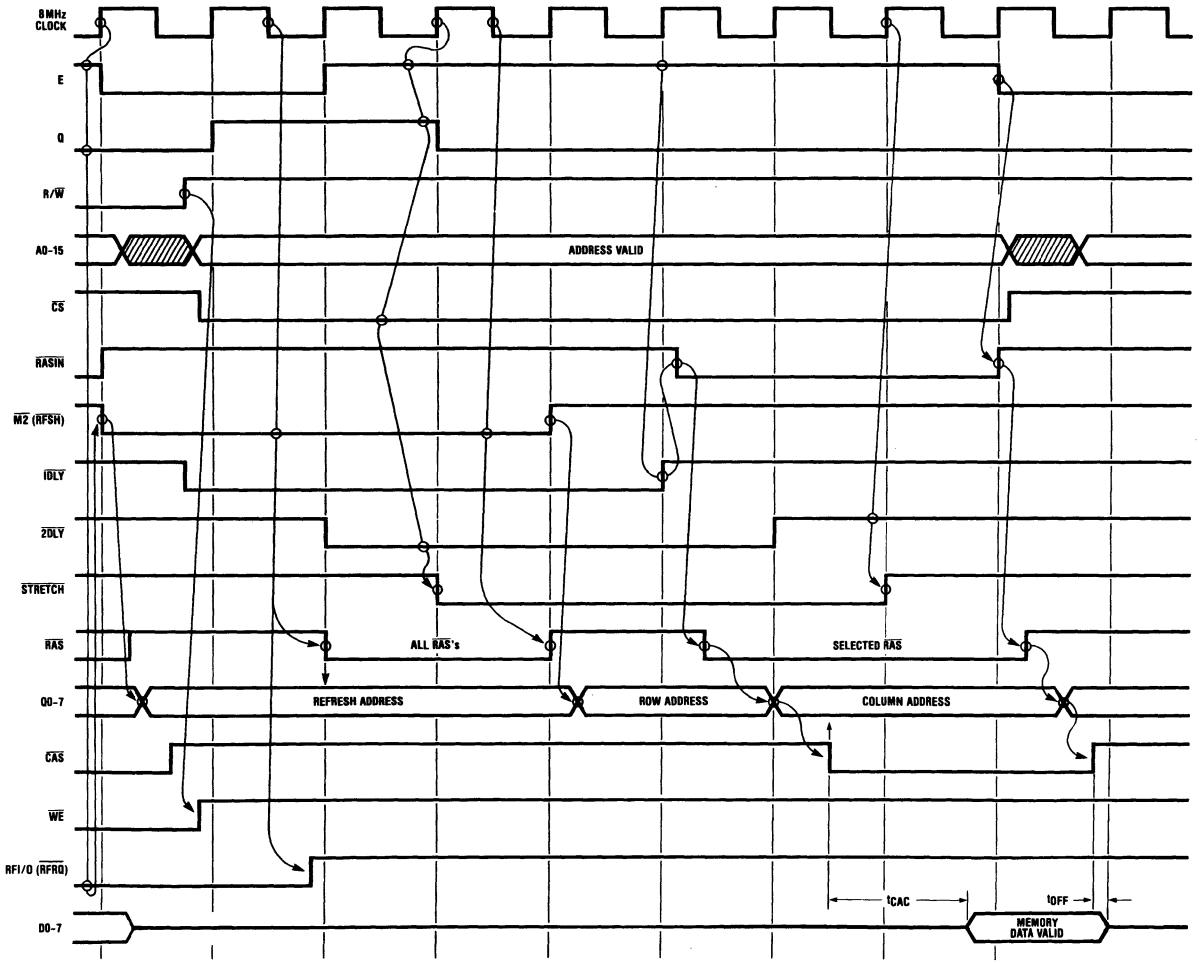
FIGURE 11. MC68B09E and DP8409A Interface



TL/F/5040–13

*If CS is high throughout this cycle (RFCK is also high), hidden refresh occurs instead of a memory access.

FIGURE 12. MC68B09E Memory Read Cycle

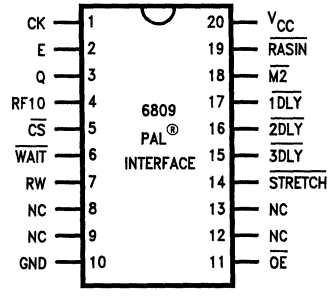


TL/F/5040–14

FIGURE 13. MC68B09E Forced Refresh and Memory Read Cycle

```
PAL16R6A
User Part #
6809/8409A Interface PAL
National Semiconductor
CK E Q RFIO /CS /WAIT RW A B GND /OE C D /STRETCH
/3DLY /2DLY /1DLY /M2 /RASIN VCC
If (VCC) RASIN=CS*E*/M2*/1DLY +
                CS*Q*/M2
M2:=E*/RFIO*/Q +
     M2*/3DLY
1DLY:=M2
2DLY:=1DLY
3DLY:=2DLY
STRETCH:=CS*2DLY*E +
          CS*WAIT*E*Q*RW
;DESCRIPTION:
;The above equations are written in standard PALASM™ format.
;Also included in the logic is a ""/WAIT"" (active low) input. This
;input will allow the insertion of one WAIT state in a READ
;access cycle if it is tied low. If WAIT states are wanted in
;both READ and WRITE access cycles the ""RW"" input in the STRETCH
;equation should be deleted.
;The user should make sure that CS is valid at the DP8409A input a
;minimum of 30 ns before RASIN is valid. If the user does not
;care about the HIDDEN REFRESH feature of the DP8409A, CS can be
;tied permanently low. In this configuration the RASIN term can
;transition whenever is convenient.
```

Dual-In-Line Package



TL/F/5040–15

Top View

DP8400/8419 Error Correcting Dynamic RAM Memory System for the Series 32000®

National Semiconductor
Application Note 387
Webster (Rusty) Meier

INTRODUCTION

Three PAL's® (Programmable Array Logic devices) were used in this application in order to interface between the NS32016, DP8419 and the DP8400 to produce an error correcting memory system for the Series 32000 microprocessor family. The PAL Interface Controller (hereafter referred to as P.I.C.) takes care of all interfacing logic, no extra control logic is needed.

FEATURES

- The P.I.C. controls the following types of cycles:
  - A) READ cycles with no errors detected, ALWAYS CORRECT MODE (1 WAIT state inserted).
  - B) READ cycles with single error detected, the correct data will be written back to memory and given to the CPU. One WAIT state is inserted into the READ cycle and one WAIT state is inserted into the next access cycle (and the access is delayed) if it immediately follows the READ cycle.
  - C) READ cycles with more then one error detected. In this case the processor is interrupted and appropriate action can be taken.
  - D) WRITE cycles (no WAIT states).
  - E) BYTE WRITE cycles, or READ MODIFY WRITE cycles (3 WAIT states inserted). If more then one error is detected in the READ portion of this cycle the processor will be interrupted so appropriate action can be taken.
  - F) DRAM REFRESH cycles (may cause a maximum of 5 WAIT states to be inserted into an access cycle if the access occurs while the refresh is taking place).
- All single bit errors are automatically corrected and rewritten back to memory.
- All double bit errors are detected and cause a system interrupt.
- Can directly drive up to 2M bytes of Dynamic RAM (4 banks of 22 256k DRAMS, each bank being 16 data bits plus 6 check bits).
- The P.I.C. allows full use of the DP8400 and all its modes of operation, including:
  - A) The DIAGNOSTIC modes (can do a diagnostic test of the DP8400 without needing to use external memory).
  - B) The COMPLEMENT modes (useful for doing the DOUBLE COMPLEMENT METHOD to try to correct 2 errors).
- The P.I.C. interfaces between the DP8409A or DP8419 Dynamic RAM controller, the DP8400 Expandable Error Checker and Corrector, the NS32016 processor, the NS32201 Timing Control Unit, and the NS32082 Memory Management Unit (if used in the system).
- Provides outputs to interrupt the CPU and to insert WAIT states if needed.
- This interface uses PAL's whose equations and timing are given, allowing the user to customize the interface to his own requirements (even a different processor family) if he so desires.
- Can work at 10 MHz (using the new DP8419, DP8400-2, and common 120 ns 64k DRAMs). Operation at higher frequencies is possible.

DESCRIPTION

The P.I.C. consists of 3 PAL's and one 74LS164 parallel output serial shift register (see P.I.C. logic diagram). If greater speed is needed for the shift register (CPU clock speed is over 6 MHz) one could use some similar type of shift register in a faster type of logic ("AS, ALS, F"), or could make one out of D flip-flops (74AS174).

If one is using a CPU other then the Series 32000 and does not have a fast clock (FCLK, twice system clock frequency) he could substitute a 5 or 10 tap delay line for the shift register.

The P.I.C. uses a shift register as an aid in determining the state of the CPU and where it is in an access cycle. When either of the two outputs, "$\overline{\text{RASIN}}$" or "$\overline{\text{RFSH}}$", go true the shift register is enabled and begins producing a series of delays. These delays, along with specific signals from the CPU, are used in the interface to determine the state of the CPU and create the appropriate control signals for the DP8400, the DP8409A/DP8419, and the processor. Other CPUs should be able to customize this interface to their requirements by adjusting the appropriate equations.

The logic in the upper right hand corner of the P.I.C. logic diagram may not be needed (74LS374's, 74LS244, 74LS240's LED's and several SSI gates). The logic allows the latching of the DRAM bank (BA17, BA18), the syndrome (S0–S7), and the error flags (AE, E0, E1) during an error condition. The latched data will be displayed on the LED's (until the I/O RESET signal is applied) and can be read from the data bus by the CPU. The address in error could also be latched by this same logic, if desired.

The 2 input AND gate (U5) in the upper left of the P.I.C. logic diagram holds $\overline{\text{CS}}$ low until after $\overline{\text{RASIN}}$ goes high on the DP8409A/19. This is particularly useful for READ cycles with one ERROR where $\overline{\text{RASIN}}$ is extended beyond the end of the current cycle, perhaps into another access cycle.

In this application double bit errors, in the dynamic RAM, generate an interrupt to the CPU. All single bit errors are automatically corrected and rewritten back to memory.

During a SYSTEM RESET the internal flip-flops of PAL #1 are set to a refresh state by making the RESET input look like a refresh request (External logic was used to "NOR" the DP8409A/19 RFI/O input with a system RESET input to produce the PAL #1 RFI/O input).

The P.I.C. performs HIDDEN REFRESHES (CPU not accessing the Dynamic RAM controlled by the DP8409A, indicated by "/CS" being high) assuming a 4 "T" state processor access cycle.

The P.I.C. allows the full use of the DP8400 and all its modes of operation. For example, the DP8400 has excellent diagnostic capabilities included in modes "2" and "6". These modes allow one to perform a complete diagnostic test of the DP8400 without using the external memory. This is possible using an I/O port to control "M1 and M0" of the DP8400, along with the diagnostic control signals "DIAGCS and DIAGD" as follows:

1) The user can set the I/O signals "M1" and "DIAGCS" both high and perform a mode 2 DIAGNOSTIC WRITE to the DP8400 with user generated CHECK bits on the high byte of the data bus. The CHECK bits will be latched into the DP8400 (CSLE held low) until the user sets the I/O signal "DIAGCS" low.
2) The user can then set the I/O signals "M1" low and "DIAGD" high and perform a mode 0 WRITE, latching the user generated data in the DP8400 input latches (DLE held low).
3) Next, the user can perform a normal mode 4 READ. This will in effect be a diagnostic READ of the user generated data and check bits without using the external memory. In this way the DP8400 can be completely checked out during system initialization.
4) The syndromes, check bits, and error flags can also be read, provided $\overline{\text{ODLE}}$, $\overline{\text{OB0}}$, and $\overline{\text{OB1}}$ are low, using mode 6A or by reading the latches.
5) When the diagnostics are completed the user can return the DP8400 to normal functioning by resetting the I/O port outputs to the original DP8400 operating mode values ("M0, M1, DIAGCS, DIAGD" all low, and "$\overline{\text{I/O RESET}}$" high).

Using the I/O port signal "M0" the user could perform the DOUBLE COMPLEMENT METHOD to try to correct a DOUBLE bit error in the DRAM (see DP8400 data sheet for further information on the DOUBLE COMPLEMENT METHOD).

Another I/O port output, "I/O $\overline{\text{RESET}}$", allows the outputs "$\overline{\text{DOUBLERROR}}$" and "$\overline{\text{ERROR}}$" in PAL #3 to be reset. The signal "$\overline{\text{ERRLAT}}$" is used in this interface to latch the SYNDROME, DRAM bank, and ERROR flags during a CPU READ access with a single, double, or triple bit error. The CPU can READ these latched error signals by performing a memory READ from a specific memory location. (An OFF BOARD CHIP SELECT, "$\overline{\text{CS-OFFB}}$".) This READ will gate the latched error condition to the CPU data bus via the 74LS244 buffer and the signal $\overline{\text{SYNDROME-DATA}}$ (see the upper right hand corner of the P.I.C. controller logic diagram).

The PAL equations that follow are in the National Semiconductor PLAN™ format, which differs from the standard PALASM™ format.

EXAMPLE: PLAN FORMAT

"$\overline{\text{RASIN}}$:= RFSH * $\overline{\text{2D}}$ * ODLE"

This translates as, "$\overline{\text{RASIN}}$" is low after the rising edge of the input clock given that "$\overline{\text{RFSH}}$" was high and "2D" was low and "$\overline{\text{ODLE}}$" was high a setup time before the clock transitions high (here $\overline{\text{RASIN}}$, $\overline{\text{RFSH}}$, and $\overline{\text{ODLE}}$ are outputs of the PAL and 2D is an input).

EXAMPLE: PALASM FORMAT

"RASIN := $\overline{\text{RFSH}}$ * $\overline{\text{2D}}$ * $\overline{\text{ODLE}}$"

The above expression means the same as the PLAN format expression except it is written in PALASM format. In other words "$\overline{\text{RASIN}}$" will go low after the rising edge of the clock given that "$\overline{\text{RFSH}}$" was high, "2D" was low and "$\overline{\text{ODLE}}$" was high a setup time before the clock transitions high (here $\overline{\text{RASIN}}$, $\overline{\text{RFSH}}$, and $\overline{\text{ODLE}}$ are outputs and 2D is an input).

Depending on the Specific type of PAL's and logic used the user can calculate the speed requirements for the DRAM at the specified processor frequency as follows:

Here both "t_{RAC}" and "t_{CAC}" must be calculated and considered in determining what speed DRAM can be used in a particular system design. The DRAM chosen must meet both the "t_{RAC}" and "t_{CAC}" parameters calculated.

EXAMPLE SYSTEM, 10 MHz, DP8400-2, DP8419, FAST "A" PART PALs

#1) $\overline{\text{RASIN}}$ low = T1 − 2 ns (F_{CLK} − PHI1 skew) + 15 ns ("A" PAL clocked output) = 100 − 2 + 15 = 113 ns maximum

#2) $\overline{\text{RASIN}}$ to $\overline{\text{RAS}}$ low = 20 ns maximum (DP8419)

#3) $\overline{\text{RASIN}}$ to $\overline{\text{CAS}}$ low = 80 ns (DP8419 $\overline{\text{RASIN}}$-$\overline{\text{CAS}}$ low maximum)

#4) 74F244 transceiver delay = 7 ns maximum

#5) DP8400-2 data setup time to "CSLE, DLE" = 10 ns maximum

#6) Minimum "CSLE, DLE" delay into "T3" = Minimum "A" PAL delay − minimum FCLK to PHI1 skew = 8 − 2 = 6 ns minimum

"t_{RAC}" = T1 + T2 + TW − #1 − #2 − #4 − #5 + #6
= 100 + 100 + 100 − 113 − 20 − 7 − 10 + 6
= 156 ns

"t_{CAC}" = T1 + T2 + TW − #1 − #3 − #4 − #5 + #6
= 100 + 100 + 100 − 113 − 80 − 7 − 10 + 6
= 96 ns

Therefore the DRAM chosen should have a "t_{RAC}" less than or equal to 156 ns and a "t_{CAC}" less than or equal to 96 ns. Standard 150 ns DRAMs meet this criteria.

Approximately 150 ns minimum RAS precharge time.

Approximately 200 ns minimum CAS precharge time.

Approximately 230 ns minimum RAS pulse width.

Approximately 180 ns minimum CAS pulse width.

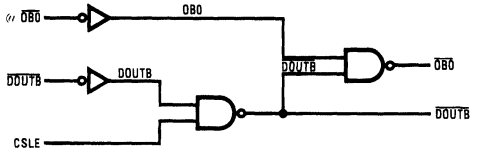
One must also consider the WRITE command to $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ lead times when choosing DRAMs for this system. During a READ access cycle, with a single bit error, a READ-MODIFY-WRITE access is performed. Here, the WRITE command to $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ lead times are one half period in length. This may present a problem to systems operating at frequencies of 10 MHz or greater. One can alleviate this problem by inserting an extra WAIT state into READ access cycles (see Use of P.I.C. at higher operating frequencies, #3) or by using external drivers from the PAL "WE" output to the DRAM "WE" input (thereby speeding up the $\overline{\text{WIN}}$ to $\overline{\text{WE}}$ delay and guaranteeing a greater $\overline{\text{WE}}$ to $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ lead time).

USE OF THE P.I.C. AT HIGHER FREQUENCIES

1) If one is using this interface above 4-6 MHz he should consider using the fast PAL's* (example "PAL16R8A" instead of "PAL16R8"), a fast shift register (example 74F164), external fast logic (such as "AS, ALS, or F" type 74XX series) or the faster "B" type PALs to produce outputs "$\overline{\text{DOUTB}}$, $\overline{\text{OB0}}$, $\overline{\text{OB1}}$" to the DP8400, and the new DP8400-2 error correction chip. The fast PAL's* have an input to output maximum time of 25 ns, and 15 ns if it is a registered output. The slow PAL's* have an input to output maximum time of 35 ns, and 25 ns if it is a registered output.

One needs to produce "$\overline{DOUTB}$, $\overline{OB0}$, $\overline{OB1}$" faster at higher CPU speeds to guarantee that the CPU reads valid data during a READ access cycle. To do this he could use external fast logic as shown in the following figure.

Using the above example we can calculate (assuming a 10 MHz 32000 series processor) the time required to have valid data at the CPU data input pins.



TL/F/8400–1

@$\overline{OB1}$ would have the same configuration as $\overline{OB0}$

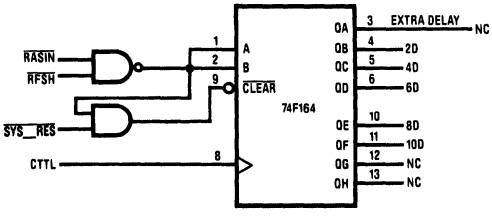
13 ns (maximum time of CSLE into state T3 assuming fast "A" PAL) +9 ns (maximum 74ALS00 propagation delay) + 9 ns (max 74ALS00 prop delay) +36 ns (maximum DP8400-2 "$\overline{OB0}$, $\overline{OB1}$" to output valid delay) + 7 ns (maximum 74F245 propagation delay) + 20 ns (data setup time required for the series 32000 with respect to the CTTL-clock) = 94 ns ***This value must not exceed 100 ns for a 10 MHz processor.

The delay of "$\overline{DOUTB}$" is to allow the DP8400 data, check bit and syndrome latches "DLE, and CSLE" to latch the data and check bits before turning off the DRAM output buffers.

The delay of "$\overline{OB0}$ and $\overline{OB1}$" allow the DRAM output buffers to turn off before the DP8400 starts driving the DP8400 memory data bus. In general the DRAM output buffers should turn off much faster then the DP8400 output buffers can turn on, so the user may want to allow "$\overline{OB0}$, $\overline{OB1}$" to become valid at the same time as "$\overline{DOUTB}$" transitions high.

2) In order to allow the use of slower DRAMs at higher CPU speeds one may want to slow down access cycles by adding an extra WAIT state.

To do this one could replace the 74LS164 IC with the following circuit:



TL/F/8400–2

Here "CTTL" was used instead of "FCLK" with a 74F164.

The "$\overline{RFSH}$" PAL equation must be adjusted to keep "$\overline{RFSH}$" 5 clock periods long, as follows:

$\overline{RFSH}$: = $\overline{RFIO}$*INCY*$\overline{2D}$
+$\overline{RFSH}$*$\overline{RFIO}$
+$\overline{RFSH}$*$\overline{6D}$
+$\overline{RFSH}$*CTTL

If WAIT states are also wanted in WRITE access cycles the "$\overline{CWAIT}$" equations must include the following term:

+RFSH * $\overline{INCY}$ * $\overline{TSO}$ * DDIN * $\overline{2D}$

If one wants to keep WRITE cycles without WAIT states inserted then the "$\overline{RASIN}$" equations must be modified for HIDDEN REFRESH and WRITE cycles as follows:

+RFSH * $\overline{RASIN}$ * $\overline{INCY}$ * $\overline{2D}$

3) Another possibility for this interface at higher frequencies would be to adjust READ access cycles by adding another WAIT state to them, as well as adjusting BYTE WRITE cycles.

Using this method one would need another stage for the shift register or use a 74F164 and use CTTL as its clock instead of FCLK. If one looks at the above figure, using the 74F164, for reference the extra stage "10D" would be used. This would allow one to make the READ access cycle one "T" state longer by adjusting the READ and READ with error "$\overline{RASIN}$" equations.

To make the READ access cycle one "T" state longer another WAIT state would have to be added to READ cycles (making a total of 2 WAIT states) and the latch signals "$\overline{ODLE}$" and "CSLE" must be adjusted by delaying them back ½ "T" state (allowing a ½ cycle longer access time). This also has the advantage of allowing the other ½ cycle of time to get the data valid at the inputs of the Series 32000 CPU.

The BYTE WRITE access cycle could also be adjusted by delaying the signals "$\overline{ODLE}$" and "CSLE" by ½ cycle. No other equations need to be touched. This would allow an extra ½ cycle for access time during BYTE WRITE access cycles.

This would allow a standard 150 ns to possibly 200 ns DRAM in a 10 MHz system [80.5 ns + ½ "T" state (50 ns) = 130.5 ns column access time (t_{CAC})] but would sacrifice by having 2 WAIT states in READ access cycles.

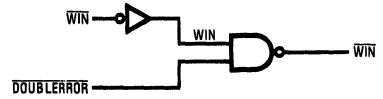
4) One also must be careful to make sure that $\overline{CS}$ is low, during an access, a minimum of 30 ns (DP8409A, 15 ns DP8419) before $\overline{RASIN}$ transitions low. If this is a problem one could tie $\overline{CS}$ permanently low (disabling hidden REFRESH) and use the system transceivers to select the memory system.

OTHER OPTIONS

If one is using the NS32082 Memory Management unit in a Series 32000 system he should connect the output "$\overline{PAV}$" (Physical Address Valid) to the P.I.C. instead of the address strobe output "$\overline{ADS}$".

An output for the BUS PARITY ERROR in a data transfer from the CPU to memory could also be detected, from the error flags and "AE" of the DP8400, and used to interrupt the CPU. However, the P.I.C. does not make use of that feature of the DP8400, though it would be very easy to add.

If one does not want to WRITE corrected data to memory in case of a DOUBLE BIT error, in READ access cycle, he could disable the WRITE signal, "$\overline{WIN}$", during a DOUBLE BIT error as follows:



TL/F/8400–3

NS32016, DP8400, DP8409A PALs Inputs and Outputs

PIN NUMBER OF THE PAL ON THE LEFT

PAL #1 Inputs

1) "FCLK" Fast Clock (twice "CTTL" frequency) from NS32201.
2) "CTTL" Output clock from NS32201.
3) "$\overline{CS}$" Chip Select for the Dynamic RAM controlled by the DP8409A and DP8400.
4) "$\overline{DDIN}$" Data Direction in, from NS32016, indicates the direction of the data transfer during a bus cycle.
5) "RFIO" Refresh request output from the DP8409A, also is used as a reset input to set PAL to a known state.
6) "$\overline{INCY}$" Output from PAL #2 indicating that the NS32016 is in an access cycle.
7) "$\overline{AOHBE}$" If address bit 0 and high byte enable (from NS32016) are both low this input is high. Used to determine when byte operations are in progress.
8) "2D" "RASIN" or "RFSH" delayed by 2 periods of FCLK. This output is from the external shift register.
9) "$\overline{ERRLAT}$" Output from PAL #3 indicating that any error, "AE", was valid during a READ access cycle.
11) "$\overline{OE}$" Enables PAL outputs.
12) "4D" "2D" delayed by 2 periods of RFCK, also an output of the external shift register.
18) "6D" "4D" delayed by 2 periods of RGCK, also an output of the external shift register.
19) "8D" "6D" delayed by 2 periods of RGCK, also an output of the external shift register.

PAL #1 Outputs

17) "$\overline{RASIN}$" Input to DP8409A
16) "$\overline{RFSH}$" Input to DP8409A, causes the DP8409A to enter mode 1 to do a refresh.
15) "$\overline{WIN}$" This output is used as an input to the DP8409A. It causes a WRITE to the DRAM.
14) "$\overline{CYCLED}$" This output is used in many other equasions and functions as a signal that the particular access cycle is midway to completion.

PAL #2 Inputs

1) "$\overline{RFSH}$" Output from PAL #1 that indicates whether the DRAMs are being refreshed.
2) "$\overline{RASIN}$" Output from PAL #1.
3) "AO" Output from NS32016, address bit 0.
4) "$\overline{HBE}$" Output from NS32016, high byte enable.
5) "$\overline{DDIN}$" Data Direction in, from NS32016.
6) "$\overline{ADS}$" Address strobe from NS32016.
7) "$\overline{TSO}$" Output from NS32016.
8) "2D" Output from the shift register.
9) "$\overline{CS}$" Chip select for the DRAM.
11) "$\overline{CYCLED}$" Output from PAL #1.
13) "$\overline{ODLE}$" Output Latch Enable to the DP8400 (Output from PAL #3).

PAL #2 Outputs

19) "$\overline{PBUF1}$" This signal enables the high byte of the processor, through the CPU transceiver, onto the DP8400/ Memory data bus.
18) "$\overline{OB1}$" Controls DP8400 output buffer for byte "1".
17) "$\overline{OB0}$" Controls DP8400 output buffer for byte "0".
16) "$\overline{PBUF0}$" This signal enables the low byte of the processor, through the CPU transceiver, onto the DP8400/ Memory data bus.
15) "$\overline{DOUTB}$" Controls memory buffers that interface between the DRAM and the DP8400 memory data bus.
14) "$\overline{INCY}$" Output indicating that the NS32016 is in an access cycle.
12) "$\overline{CWAIT}$" Output to NS32016 that causes WAIT states to be inserted into the NS32016 bus cycles.

PAL #3 Inputs

1) "FCLK" Fast clock from NS32201.
2) "CTTL" System clock from the NS32201.
3) "DIAGCS" Enable input from I/O port for diagnostics to enable "CSLE", check bit syndrome latch enable.
4) "DIAGD" Enable input from I/O port for diagnostics to enable "DLE", data latch enable.
5) "$\overline{RESET}$" Reset input from I/O port to reset PAL error latches.
6) "CSRASIN" Output from the PAL #1 logically "NOR"ed with the DRAM Chip Select signal. This indicates the beginning of a selected DRAM access cycle.
7) "AE" Output from DP8400 indicating an error.
8) "E01" This is the "E0" and "E1" error flags, of the DP8400, logically "NOR"ed together.
9) "$\overline{DOUTB}$" Controls memory buffers that interface between the DRAM and the DP8400/memory data base.
11) "$\overline{OE}$" Enables the PAL outputs.
12) "$\overline{AOHBE}$" If address bit 0 AND high byte enable (from NS32016) are both low this input is high. Used to determine when byte operations are in progress.
19) "$\overline{DDIN}$" Data Direction in, from NS32016.

PAL #3 Outputs

18) "$\overline{ODLE}$" Output that controls both the DP8400 Data latch and output latches. This output goes directly to both the "DLE" and $\overline{OLE}$ pin of the DP8400.
17) "$\overline{CSLE}$" Output that controls the DP8400 Check bit Syndrome latch. This output goes directly to the "CSLE" pin of the DP8400, it is only inverted so the PAL programmer will program it correctly.

NS32016, DP8400, DP8409A PALs Inputs and Outputs (Continued)

16) "$\overline{\text{MODECC}}$" Output that is used as an input to the DP8400. This signal controls whether the DP8400 is in READ or WRITE Mode.

15) "$\overline{\text{DOUBLERR}}$" Used to interrupt the system when a double bit error has been detected during a READ cycle.

14) "$\overline{\text{ERRLAT}}$" Used in the PAL controller to indicate that an error has occurred during a $\overline{\text{CS}}$ READ cycle or a $\overline{\text{CS}}$ BYTE WRITE cycle, as indicated by "AE" being valid. This signal can be used to latch the DRAM bank in error, the SYNDROME of the error, the ERROR flags, and the DRAM address (of the data in error) when a DRAM error occurs.

13) "$\overline{\text{ERROR}}$" This output is used to display the DRAM bank in error, the syndrome of the error, and the error flags of the DP8400 when a single, double, or triple bit error occurs. The preceding error condition is held in an external error register (74LS374's). The contents of the registers are displayed on LED's to help the user diagnose where a DRAM problem may reside in the memory system.

PAL NUMBER 1

```
PAL16R4A
FCLK CTTL /CS /DDIN RFIO /INCY /AOHBE 2D /ERRLAT GND
/OE 4D NC /CYCLED /WIN /RFSH /RASIN 6D 8D VCC
/RASIN : = RFSH*/INCY*/4D*/CTTL*ERRLAT          ;Start /RASIN
    +RFSH*/RASIN*/INCY*/4D                      ;WRITE or hidden RFSH
    +RFSH*/CS*/RASIN*/INCY*/DDIN*/6D            ;READ cycle
    +RFSH*/CS*/RASIN*/INCY*DDIN*/AOHBE*WIN      ;BYTE WRITE cycle
    +RFSH*/CS*/RASIN*/INCY*DDIN*/AOHBE*CTTL     ;Extend BYTE WRITE
    +RFSH*/CS*/RASIN*/DDIN*/ERRLAT*/8D          ;READ w/error

/RFSH : = /RFIO*INCY*RASIN                  ;RFSH in idle states or in long
        +/RFSH*/RFIO                        ; accesses of other devices or
        +/RFSH*/8D                          ; at the beginning of an access
        +/RFSH*CTTL

/WIN : =
      RFSH*/CS*/RASIN*/ERRLAT*6D*/CTTL*/DDIN    ;READ w/error
      +/WIN*RFSH*/RASIN*/ERRLAT*6D              ;READ w/error continue
      +RFSH*/CS*/RASIN*DDIN*2D*CTTL*AOHBE       ;WRITE
      +/WIN*RFSH*/CS*/RASIN*DDIN*2D*AOHBE       ;WRITE continue
      +RFSH*/CS*/RASIN*DDIN*/AOHBE*/CYCLED*/CTTL ;BYTE WRITE
      +/WIN*RFSH*/CS*/RASIN*DDIN*/AOHBE*6D      ;BYTE WRITE continue

/CYCLED : =
    RFSH*/RASIN*/CS*DDIN*4D*/AOHBE*/CTTL        :BYTE WRITE
    +RFSH*/RASIN*/CS*DDIN*/AOHBE*4D*/CYCLED     ;BYTE WRITE
    +RFSH*/RASIN*/CS*/DDIN*2D*/CTTL             :READ, READ w/error
    +RFSH*/RASIN*/CS*/DDIN*4D*/CYCLED           ;READ, READ w/error
    +RFSH*/RASIN*/CS*DDIN*2D*AOHBE              ;WRITE
    +RFSH*/RASIN*CS*2D*/CTTL                    ;HIDDEN REFRESH
    +RFSH*/CYCLED*/ERRLAT                       :Finish for READ w/error
    +RFSH*/CYCLED*CTTL                          ;Finish
```

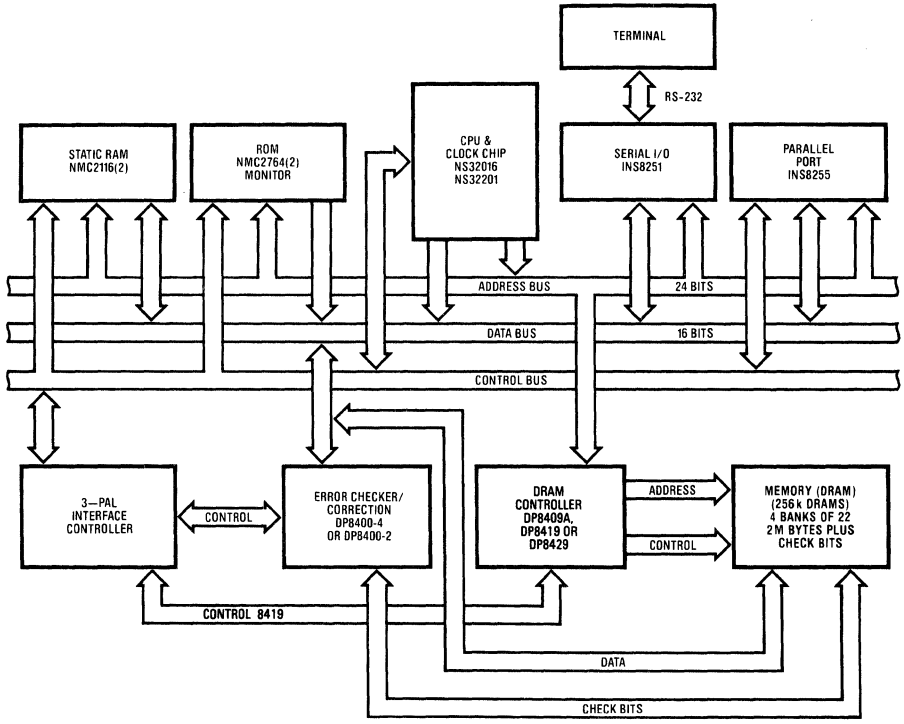
PAL NUMBER 2

```
PAL16L8A
/RFSH /RASIN A0 /HBE /DDIN /ADS /TS0 2D /CS GND
/CYCLED /CWAIT /ODLE /INCY /DOUTB /PBUF0 /OB0 /OB1 /PBUF1 VCC

IF (VCC) /PBUF1 =
   RFSH*/CS*/INCY*/DDIN*2D*/HBE                    ;READ or READ w/error
   +RFSH*/CS*/INCY*DDIN*A0*/HBE*DOUTB*/ODLE*2D     ;BYTE WRITE high
   +RFSH*/CS*/INCY*DDIN*2D*/OB0*A0*/HBE            ;BYTE WRITE cont
   +RFSH*/CS*/INCY*DDIN*/A0*/HBE*DOUTB             ;word WRITE

IF (VCC) /OB1 =
   RFSH*/CS*/INCY*/DDIN*2D*/CYCLED*DOUTB           ;READ or READ w/error
   +RFSH*/CS*/INCY*DDIN*/A0*HBE*2D*/ODLE*DOUTB     ;BYTE WRITE low
   +RFSH*/CS*/INCY*DDIN*/A0*HBE*2D*/OB1*DOUTB      ;BYTE WRITE cont
   +RFSH*/OB1*DOUTB*2D                             ;READ w/error hold

IF (VCC) /OB0 =
   RFSH*/CS*/INCY*/DDIN*2D*/CYCLED*DOUTB        ;READ or READ w/error
   +RFSH*/CS*/INCY*DDIN*A0*/HBE*2D*/ODLE*DOUTB  ;BYTE WRITE high
   +RFSH*/CS*/INCY*DDIN*A0*/HBE*2D*/OB0*DOUTB   ;BYTE WRITE cont
   +RFSH*/OB0*DOUTB*2D                          ;READ w/error hold

IF (VCC) /PBUF0 =
   RFSH*/CS*/INCY*/DDIN*2D*/A0                  ;READ or READ w/error
  +RFSH*/CS*/INCY*DDIN*/A0*HBE*DOUTB*/ODLE*2D   ;BYTE WRITE low
  +RFSH*/CS*/INCY*DDIN*2D*/A0*HBE*/OB1          ;BYTE WRITE cont
  +RFSH*/CS*/INCY*DDIN*/A0*/HBE*DOUTB           ;word WRITE

IF (VCC) /DOUTB =
    RFSH*/CS*/INCY*/DDIN*2D*CYCLED              ;READ or READ w/error
    +RFSH*/CS*/INCY*DDIN*/A0*HBE*2D*ODLE*OB1    ;BYTE WRITE low
    +RFSH*/CS*/INCY*DDIN*A0*/HBE*2D*ODLE*OB0    ;BYTE WRITE high

IF (VCC) /INCY = RFSH*/ADS*/2D*CYCLED       ;Start INCY
      +RFSH*/CS*/TS0*/2D                    ;Start INCY for access
                                            ; after forced refresh
                                            ; or READ w/error
      +RFSH*/INCY*CYCLED                    ;Continue
      +RFSH*/INCY*/TS0*/CS                  ;Continue for /CS access

IF (/CS) /CWAIT =
    /RFSH*/TS0                              ;Access in RFSH
    +RFSH*/TS0*RASIN                        ;Access after forced RFSH
    +RFSH*/INCY*/TS0*/DDIN*/2D              ;READ cycle
    +RFSH*/INCY*/TS0*DDIN*/A0*HBE*CYCLED    ;BYTE WRITE
    +RFSH*/INCY*/TS0*DDIN*A0*/HBE*CYCLED    ;BYTE WRITE
    +RFSH*/TS0*/CYCLED*/2D*RASIN            ;WAIT after READ w/error
```

PAL NUMBER 3

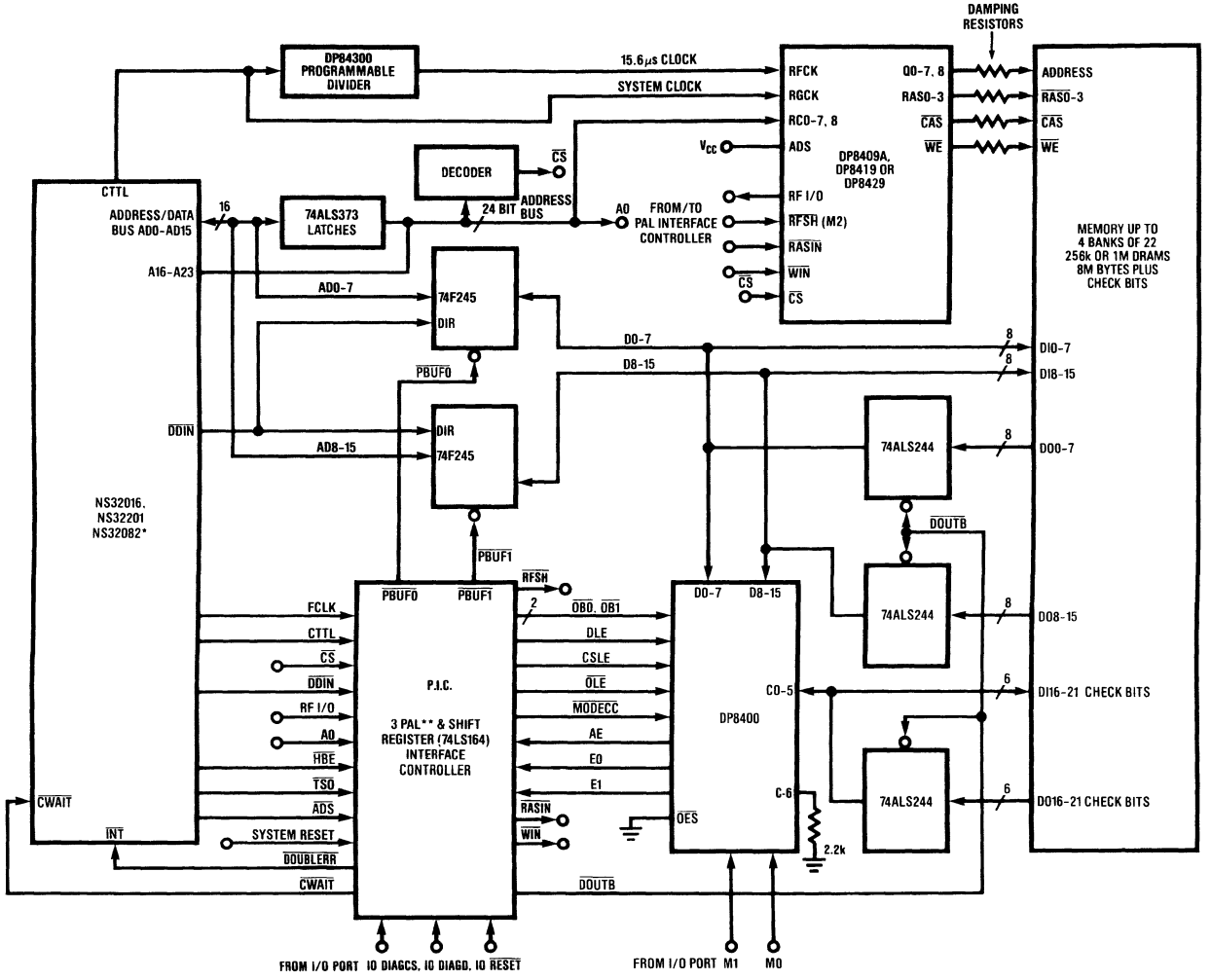
```
PAL16R6A
FCLK CTTL DIAGCS DIAGD /RESET CSRASIN AE E01 /DOUTB GND
/OE /AOHBE /ERROR /ERRLAT /DOUBLERR /MODECC /CSLE /ODLE /DDIN VCC

/ODLE : = CSRASIN*/DDIN*/DOUTB*/CTTL              ;Read
        + CSRASIN*/DDIN*/MODECC*CSLE*ODLE         ;Read with error
        + CSRASIN*DDIN*/AOHBE*/DOUTB*/CTTL        ;Byte Write
        + /ODLE*CSRASIN*DDIN*/AOHBE*CTTL          ;Continue during Byte Write
        + CSRASIN*DDIN*/AOHBE*/MODECC*CSLE*ODLE   ;Byte Write
        + CSRASIN*DDIN*AOHBE*/CTTL                ;Word Write
        + /ODLE*CSRASIN*CTTL                      ;Hold "/ODLE"
        + /ODLE*DIAGD                             ;Hold "/ODLE" for
                                                  ;  diagnostics

/CSLE : = CSRASIN*/DDIN*/DOUTB*/CTTL              ;Read
        + CSRASIN*/DDIN*/MODECC                   ;Read with error
        + CSRASIN*DDIN*/AOHBE*/DOUTB*/CTTL        ;Byte Write
        + CSRASIN*DDIN*/AOHBE*/MODECC             ;Byte Write
        + CSRASIN*DDIN*AOHBE*/CTTL                ;Word WRITE
        + /CSLE*CSRASIN*CTTL                      ;Hold "/CSLE"
        + /CSLE*DIAGCS                            ;Hold "/CSLE" for
                                                  ;  diagnostics

/MODECC :=
    CSRASIN*/ODLE*/DDIN*/CTTL          ;READ or Write w/error
   + CSRASIN*/ODLE*DDIN*/AOHBE*/CTTL   ;BYTE WRITE
   + CSRASIN*DDIN*AOHBE                ;WORD WRITE
   + /MODECC*CSRASIN                   ;Hold "/MODECC"

/DOUBLERR :=
  /DIAGCS*/DIAGD*RESET*CSRASIN*/ODLE*/CTTL*AE*E01  ;Double bit error
                                                   ;  during READs
                                                   ;  or BYTE WRITEs
  + /DOUBLERR*RESET                                ;Hold "/DOUBLERR"

/ERRLAT :=
    /DIAGCS*/DIAGD*CSRASIN*/ODLE*/CTTL*AE      ;Any Error during
                                               ;  READ or BYTE WRITE
    + /ERRLAT*CSRASIN                          ;Continue "/ERRLAT" during
                                               ;  READ or during BYTE WRITE

/ERROR  :=  /DIAGD*/DIAGCS*RESET*CSRASIN*/ERRLAT ;Store error syndrome
                                                 ;  and RAS bank and
                                                 ;  error flags
             + /ERROR*RESET                      ;Hold until RESET

;The output, "/CSLE", is shown inverted so the PAL will be
;programmed correctly, in other words, "/CSLE" goes low after the
;rising edge of FCLK given that one of its input equations was
;low a setup time before FCLK transitioned high. The output,
;"/CSLE", should go straight to the pin "CSLE" of the DP8400.
```

DP8400, DP8409A, NS32016 Error Correcting Dynamic RAM Computer System



TL/F/8400–4

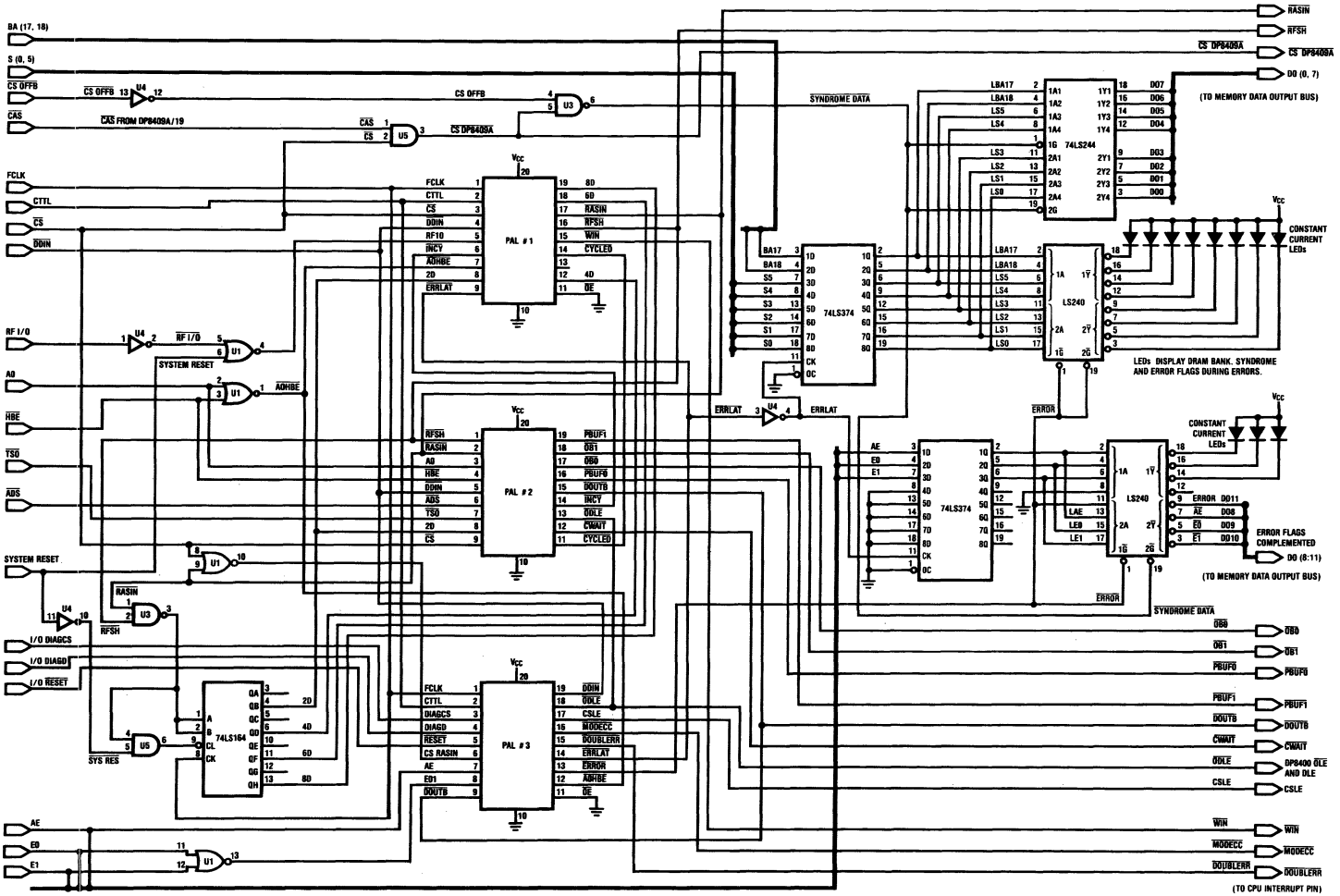
NS32016, DP8400, DP8409A or DP8418 Error Correcting Memory System



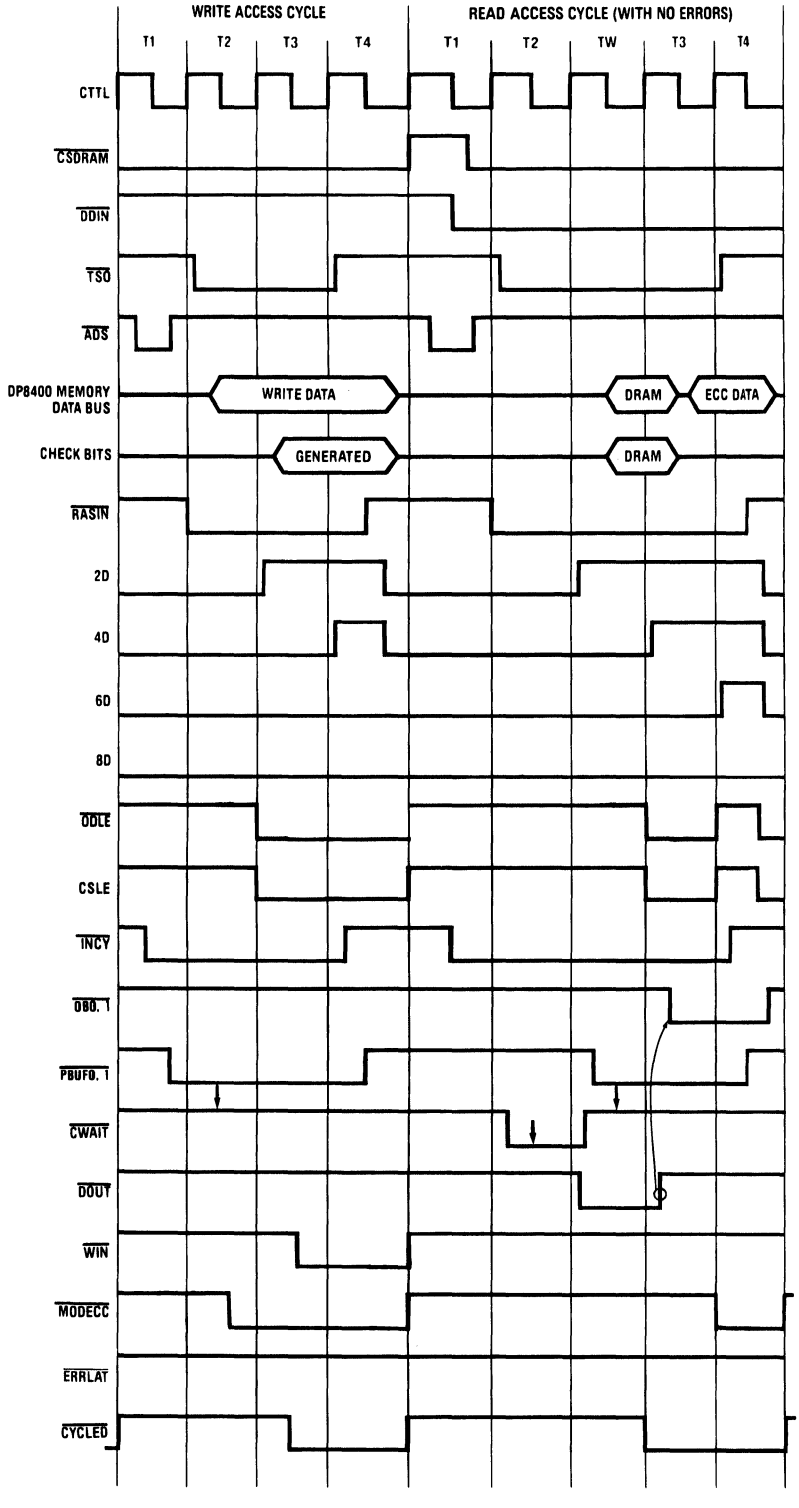
TL/F/8400–5

*IF system contains NS32082 MMU $\overline{PAV}$ should be used in place of $\overline{ADS}$

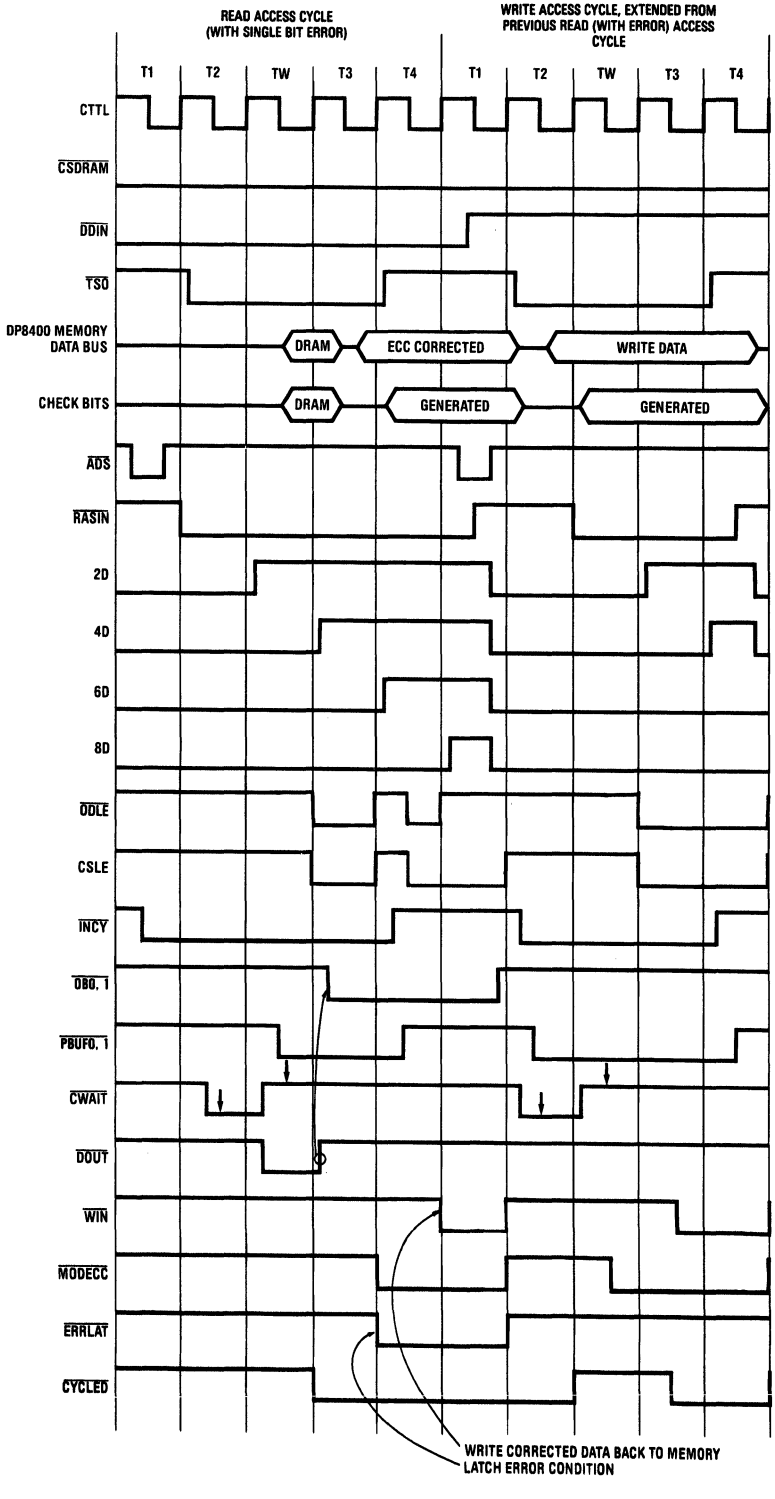
P.I.C. 3 PAL and Shift Register Interface Controller



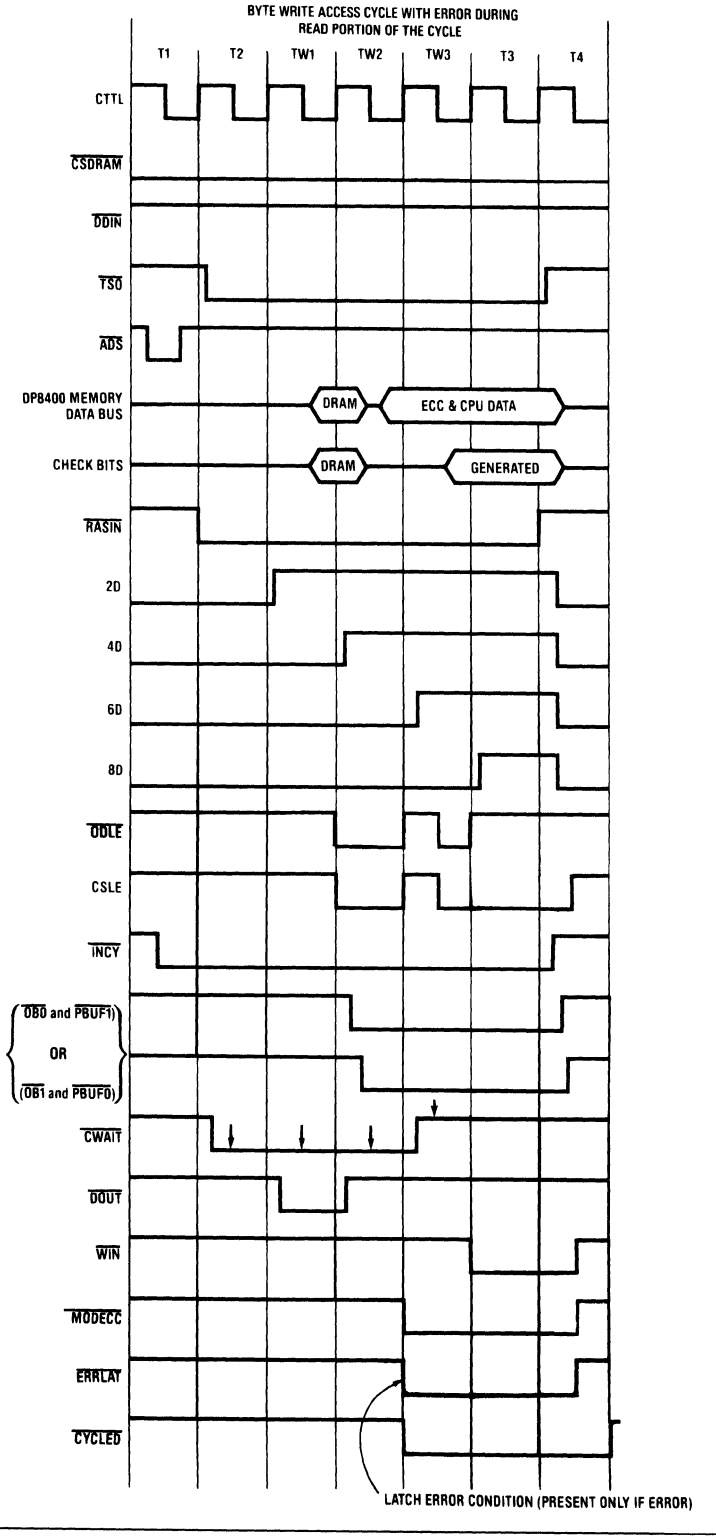
TL/F/8400–6


WRITE ACCESS CYCLE
READ ACCESS CYCLE (WITH NO ERRORS)
T1
T2
T3
T4
T1
T2
TW
T3
T4
CTTL
CSDRAM
DDIN
TSO
ADS
DP8400 MEMORY DATA BUS
WRITE DATA
DRAM
ECC DATA
CHECK BITS
GENERATED
DRAM
RASIN
2D
4D
6D
8D
ODLE
CSLE
INCY
OBO. 1
PBUFO. 1
CWAIT
DOUT
WIN
MODECC
ERRLAT
CYCLED

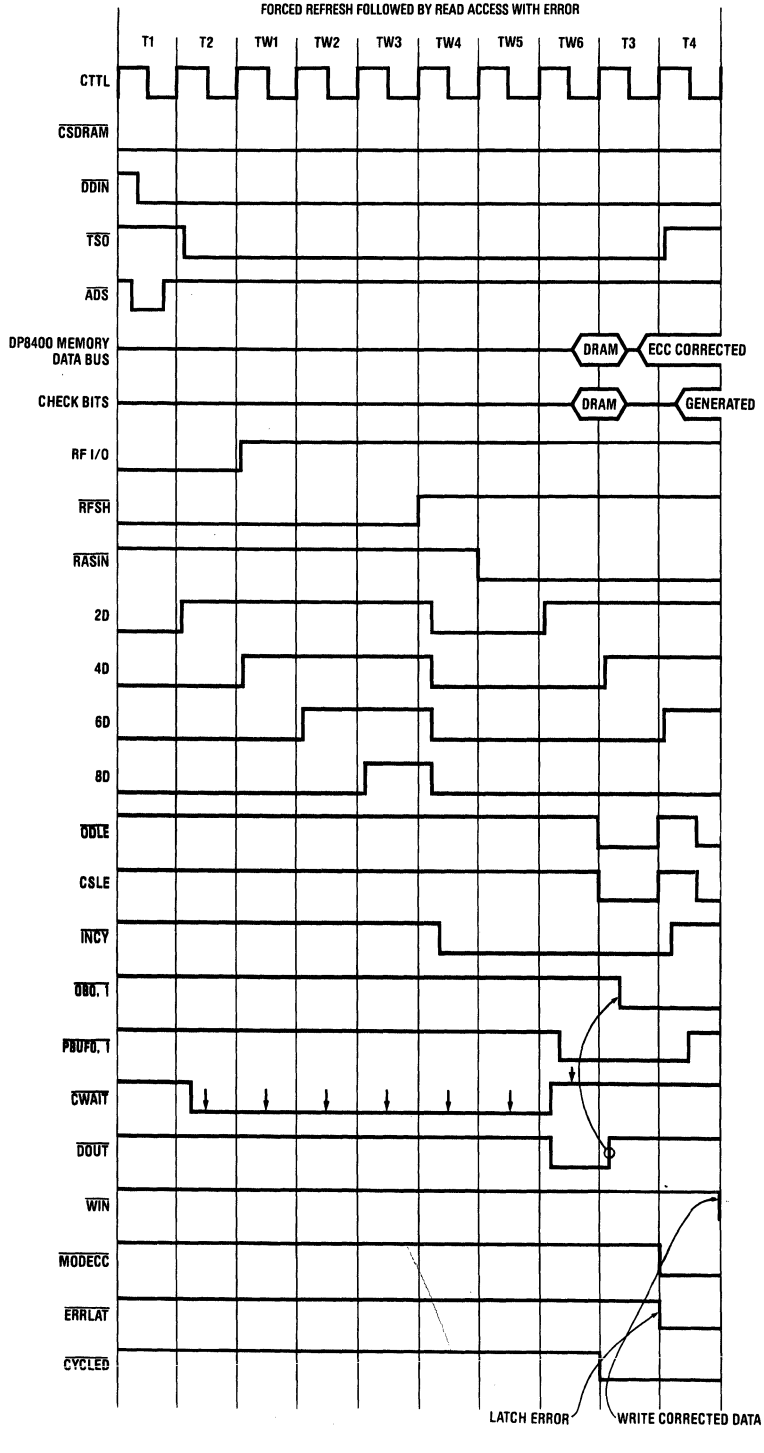
TL/F/8400–7


READ ACCESS CYCLE
(WITH SINGLE BIT ERROR)
WRITE ACCESS CYCLE, EXTENDED FROM
PREVIOUS READ (WITH ERROR) ACCESS
CYCLE
T1
T2
TW
T3
T4
T1
T2
TW
T3
T4
CTTL
CSDRAM
DDIN
TSO
DP8400 MEMORY
DATA BUS
DRAM
ECC CORRECTED
WRITE DATA
CHECK BITS
DRAM
GENERATED
GENERATED
ADS
RASIN
2D
4D
6D
8D
ODLE
CSLE
INCY
OBO, 1
PBUF0, 1
CWAIT
DOUT
WIN
MODECC
ERRLAT
CYCLED
WRITE CORRECTED DATA BACK TO MEMORY
LATCH ERROR CONDITION

TL/F/8400–8

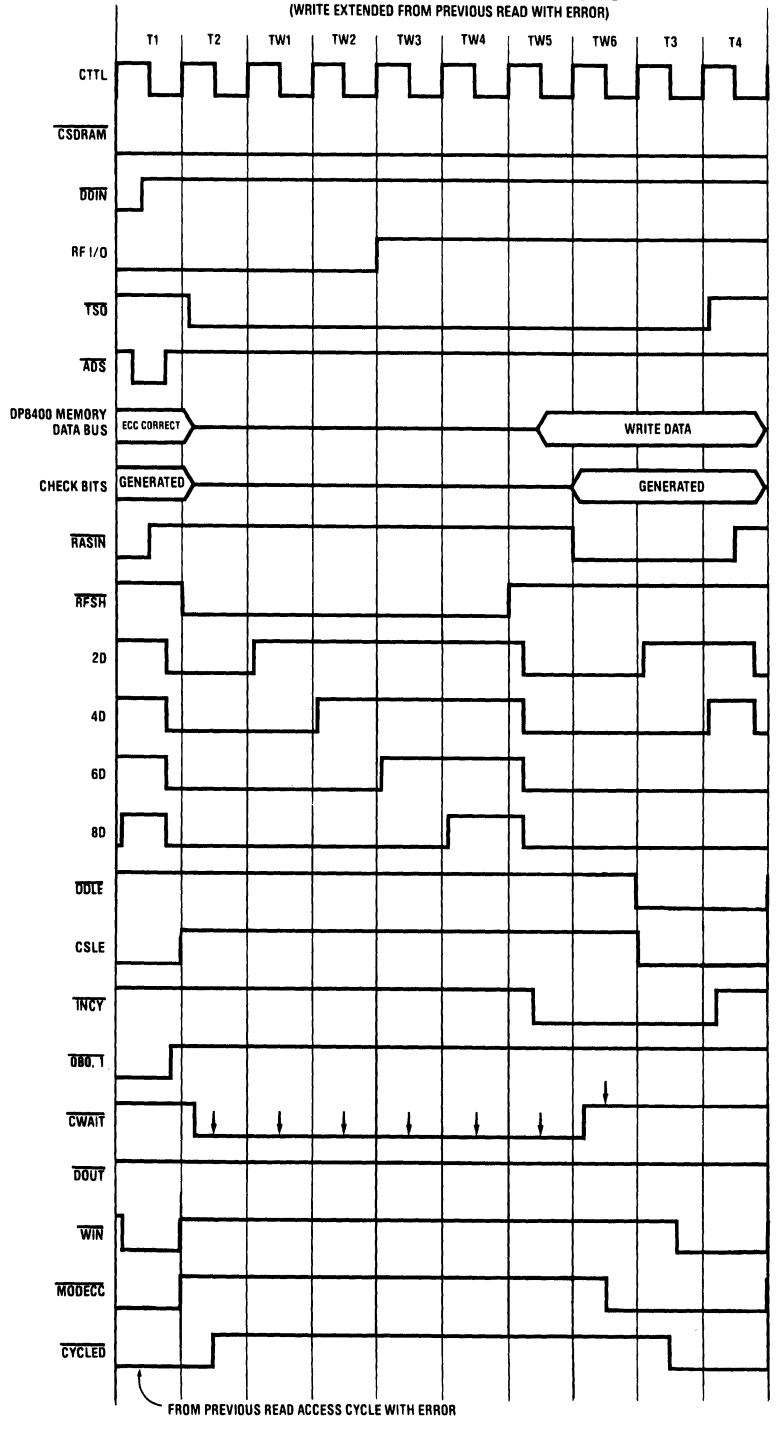

BYTE WRITE ACCESS CYCLE WITH ERROR DURING READ PORTION OF THE CYCLE
T1
T2
TW1
TW2
TW3
T3
T4
CTTL
CSDRAM
DDIN
TSO
ADS
DP8400 MEMORY DATA BUS
DRAM
ECC & CPU DATA
CHECK BITS
DRAM
GENERATED
RASIN
2D
4D
6D
8D
ODLE
CSLE
INCY
OB0 and PBUF1
OR
OB1 and PBUF0
CWAIT
DOUT
WIN
MODECC
ERRLAT
CYCLED
LATCH ERROR CONDITION (PRESENT ONLY IF ERROR)

TL/F/8400–9


FORCED REFRESH FOLLOWED BY READ ACCESS WITH ERROR
T1
T2
TW1
TW2
TW3
TW4
TW5
TW6
T3
T4
CTTL
CSDRAM
DDIN
TSO
ADS
DP8400 MEMORY DATA BUS
DRAM
ECC CORRECTED
CHECK BITS
DRAM
GENERATED
RF I/O
RFSH
RASIN
2D
4D
6D
8D
DDLE
CSLE
INCY
DBO.1
PBUF0.1
CWAIT
DOUT
WIN
MODECC
ERRLAT
CYCLED
LATCH ERROR
WRITE CORRECTED DATA

TL/F/8400–10

FORCED REFRESH WITH WRITE ACCESS AT THE SAME TIME
(WRITE EXTENDED FROM PREVIOUS READ WITH ERROR)



TL/F/8400–11

Determining the Speed of the Dynamic RAM Needed When Interfacing the DP8419-80 to Most Major Microprocessors

National Semiconductor
Application Note 411
Webster (Rusty) Meier Jr.

INTRODUCTION

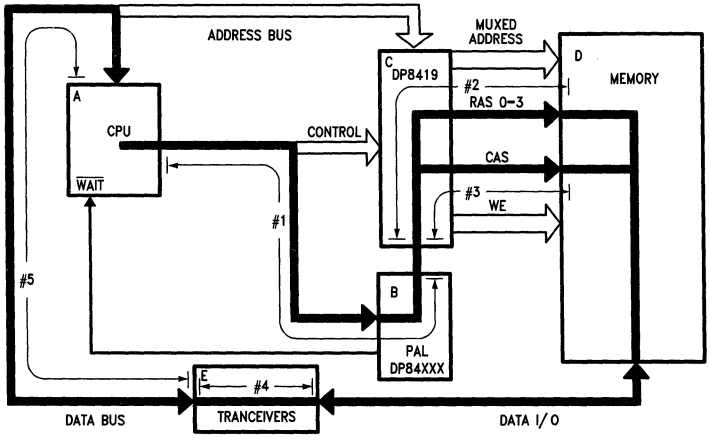
This application note looks at the individual delay elements of a CPU to memory access path for a typical memory system utilizing the DP8419-80 DRAM controller. In the final analysis the reader should be equipped with all the necessary equations to easily calculate the slowest/cheapest allowable DRAMS for no wait state CPU operation when using the DP8419-80 in his/her system.

Equations for calculating the maximum allowable DRAM "tRAC" ($\overline{RAS}$ access time) and "tCAC" ($\overline{CAS}$ access time) specifications for a particular microprocessor to operate at its maximum clock frequency without wait states are provided. Table I and *Figure 3* at the end of this application note give potential DP8419-80 users an illustration of what speed DRAM they may typically need to use in order to achieve no wait state operation with a particular microprocessor. It is important to note that even better performance can be achieved by using the faster DP8419-70.

THE 5 FUNCTIONAL BLOCKS

**Figure 1* illustrates the five functional blocks and the five main delay segments of our DP8419-80 based system example. For this particular example, the following functional block descriptions apply:

Functional Block	Functional Block Description
A)	The CPU issues an access request to the PAL then reads or writes data to or from the DRAMs;
B)	The PAL provides the refresh access arbitration logic which holds off a CPU access during a DRAM refresh and DRAM refresh during a CPU access. The PAL also provides the $\overline{RASIN}$ signal to the DP8419-80;
C)	The DP8419-80 generates the control signal timing required by the DRAMs. It also automatically multiplexes the row and column addresses during access, provides the refresh address during refresh and provides the on board capacitive drive for the direct interface with the DRAM array;
D)	The DRAM provides or stores data in response to the DP8419-80's control signal; and,
E)	The tranceivers isolate the DRAMs from the data bus when they are not being accessed in addition to passing data between the CPU and memory during read and write cycles.



TL/F/8595–1

DP84412: 32008/016/032 - DP8409A/18/19/28/29 Interface PAL
DP84512: NS32332 - DP8417/18/19/28/29 Interface PAL
DP84322/422: 68000/008/010 - DP8409A/18/19/28/29 Interface PALS
DP84522: 68020 - DP8418/19/28/29 Interface PAL
DP84432: 8086/88/186/188 - DP8409A/18/19/28/29 Interface PAL
DP84532: 80286 - DP8418/19/28/29 Interface PALS

FIGURE 1. Delay Elements of the CPU to Memory Data Access Path (DP8409A or DP8417/18/19/28/29 System)

Figure 2 may prove to be a helpful reference. It illustrates a hypothetical system timing pattern for memory accessing for a 4T state microprocessor.

DELAY SEGMENTS

Delay segments #1 through #5 are also shown in *Figure 1*. Delay segment #1 represents the timing delay from when the CPU initiates an access to the point where $\overline{RASIN}$ is issued by the PAL to the DP8419-80;

Delay segment #2 represents the $\overline{RASIN}$ to $\overline{RAS}$ out delay of the DP8419-80 DRAM controller;

Delay segment #3 represents the $\overline{RASIN}$ to $\overline{CAS}$ out delay of the DP8419-80 DRAM controller;

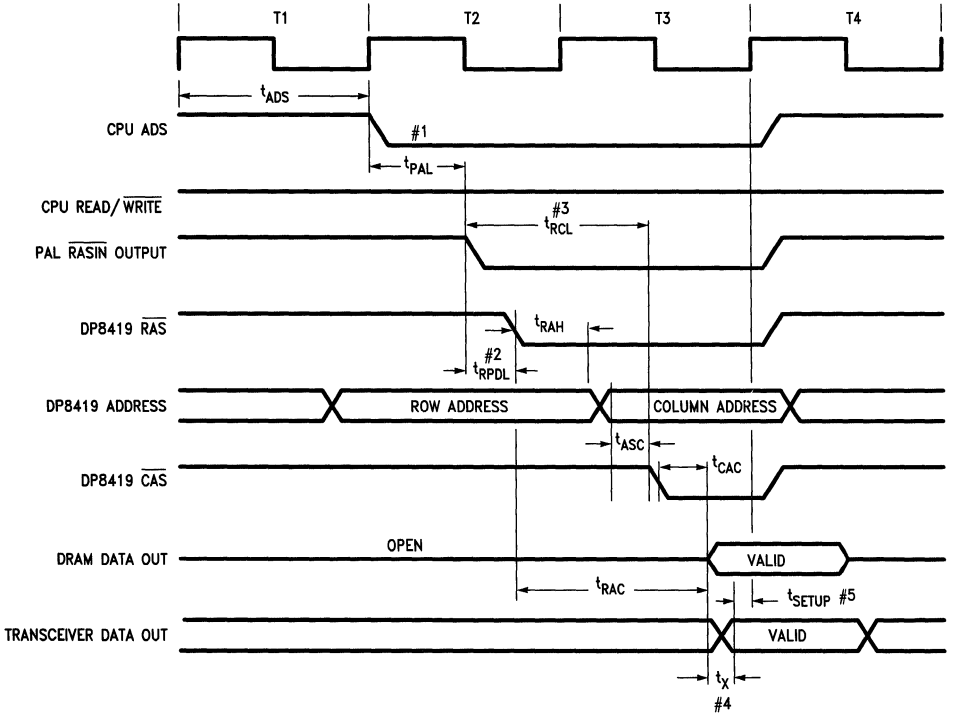
Delay segment #4 represents the inherent delay of the CPU/memory bus transceivers;

Delay segment #5 represents the required CPU data setup time.

The unique equations for determining the values of delay segments #1 through #5 for each of the major microprocessors are provided as the primary content of this application note.

Both "tRAC" and "tCAC" must be considered in determining what speed DRAM can be used in a particular system design. The DRAM chosen must meet both the "tRAC" and "tCAC" parameters calculated. If more information is desired on how "tRAC" and "tCAC" were calculated for a particular microprocessor, the reader should consult the microprocessor data sheet and the PAL data sheet for the particular microprocessor (ie. DP84412 Series 32000 processors, DP84422 68000 family processors. DP84522 68020 family processors, DP84432 iAPX88/86/188/186 processor, DP84532 iAPX286).

Most of the calculations contained in this application note use "RAHS" = 1 (15 ns guaranteed minimum row address hold time). Calculations only used "RAHS" = 0 (25 ns guaranteed minimum row address hold time) when the calculated access time from $\overline{RAS}$ equaled or exceeded 200 ns. This is because DRAMs can be found with $\overline{RAS}$ access times up to 150 ns that require only 15 ns row address hold times.



TL/F/8595-2

FIGURE 2. System Access Timing (4T State Microprocessor Example)

tRAC/tCAC CALCULATIONS FOR THE MAJOR MICROPROCESSORS

I) Series 32000 "tRAC" and "tCAC" Calculations

Series 32000 8 MHz No Wait State Calculations

#1) $\overline{RASIN}$ low = T1 − 2 ns (FCLK − PHI1 skew) + 12 ns ("B" PAL clocked output) = 125 − 2 + 12 = 135 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 80 ns (DP8419-80 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMs speced in data sheet) = 77 ns

#4) 74F245 transceiver delay = 7 ns maximum

#5) CPU data setup time to "T4" = 20 ns minimum

"tRAC" = T1 + T2 + T3 − #1 − #2 − #4 − #5
= 25 + 125 + 125 − 135 − 20 − 7 − 20
= 193 ns

"tCAC" = T1 + T2 + T3 − #1 − #3 − #4 − #5
= 125 + 125 + 125 − 135 − 77 − 7 − 20
= 136 ns

Therefore the DRAM chosen should have a "tRAC" less than or equal to 193 ns and a "tCAC" less than or equal to 136 ns. Standard 150 ns DRAMs meet this criteria.

Series 32000 10 MHz No Wait State Calculations

#1) $\overline{RASIN}$ low = T1 − 2 ns (FCLK − PHI1 skew) + 12 ns ("B" PAL clocked (output) = 100 − 2 + 12 = 110 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 80 ns (DP8419 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMs speced in data sheet) = 77 ns

#4) 74F245 transceiver delay = 7 ns maximum

#5) CPU data setup time to "T4" = 15 ns minimum

"tRAC" = T1 + T2 + T3 − #1 − #2 − #4 − #5
= 100 + 100 + 100 − 110 − 20 − 7 − 15
= 148 ns

"tCAC" = T1 + T2 + T3 − #1 − #3 − #4 − #5
= 100 + 100 + 100 − 100 − 77 − 7 − 15
= 91 ns

Therefore the DRAM chosen should have a "tRAC" less than or equal to 148 ns and a "tCAC" less than or equal to 91 ns. Standard 120 ns DRAMs meet this criteria.

II) 68000 Family "tRAC" and "tCAC" Calculations

68000 Family 8 MHz No Wait State Calculations

#1) $\overline{RASIN}$ low = S0 + S1 + $\overline{AS}$ low (maximum) + "B" PAL combinational output delay maximum = 125 + 60 + 15 = 200 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 80 ns (DP8419-80 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMS speced in data sheet) = 77 ns

#4) 74F245 transceiver delay = 7 ns maximum

#5) CPU data setup time = 15 ns minimum

"tRAC" = (S0 + S1) + (S2 + S3) + (S4 + S5) + S6 (minimum) − #1 − #2 − #4 − #5
= 125 + 125 + 125 + 55 − 200 − 20 − 7 − 15
= 188 ns

"tCAC" = (S0 + S1) + (S2 + S3) + (S4 + S5) + S6 (minimum) − #1 − #3 − #4 − #5
= 125 + 125 + 125 + 55 − 200 − 77 − 7 − 15
= 131 ns

Therefore the DRAM chosen should have a "tRAC" less than or equal to 188 ns and a "tCAC" less than or equal to 131 ns. Standard 150 ns DRAMs meet this critieria.

68000 Family 9 MHz No Wait State Calculations

#1) $\overline{RASIN}$ low = S0 + S1 + $\overline{AS}$ low (maximum) + "B" PAL combinational output delay maximum = 111 + 55 + 15 = 181 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 80 ns (DP8419-80 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMs speced in data sheet) = 77 ns

#4) 74F245 transceiver delay = 7 ns maximum

#5) CPU data setup time = 10 ns minimum

"tRAC" = (S0 + S1) + (S2 + S3) + (S4 + S5) + S6 (minimum) − #1 − #2 − #4 − #5
= 111 + 111 + 111 + 45 − 181 − 20 − 7 − 10
= 160 ns

"tCAC" = (S0 + S1) + (S2 + S3) + (S4 + S5) + S6 (minimum) − #1 − #3 − #4 − #5
= 111 + 111 + 111 + 45 − 181 − 77 − 7 − 10
= 103 ns

Therefore the DRAM chosen should have a "tRAC" less than or equal to 160 ns and a "tCAC" less than or equal to 103 ns. Standard 150 ns DRAMs meet this criteria.

68000 Family 10 MHz No Wait State Calculations

#1) $\overline{RASIN}$ low = S0 + S1 + $\overline{AS}$ low (maximum) + "B" PAL combinational output delay = 100 + 55 + 15 = 170 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 80 ns (DP8419-80 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMs speced in data sheet) = 77 ns

#4) 74F245 transceiver delay = 7 ns maximum

#5) CPU data setup time = 10 ns minimum

"tRAC" = (S0 + S1) + (S2 + S3) + (S4 + S5) + S6 (minimum) − #1 − #2 − #4 − #5
= 100 + 100 + 100 + 45 − 170 − 20 − 7 − 10
= 138 ns

"tCAC" = (S0 + S1) + (S2 + S3) + (S4 + S5) + S6 (minimum) − #1 − #3 − #4 − #5
= 100 + 100 + 100 + 45 − 170 − 77 − 7 − 10
= 81 ns

Therefore the DRAM chosen have a "tRAC" less than or equal to 138 ns and a "tCAC" less than or equal to 81 ns. Standard 120 ns DRAMs meet this criteria.

68000 Family 11 MHz No Wait State Calculations

#1) $\overline{RASIN}$ low = S0 + S1 + $\overline{AS}$ low (maximum) + "B" PAL combinational output delay maximum = 91 + 55 + 15 = 161 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 80 ns (DP8419-80 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMs speced in data sheet) = 77 ns

#4) 74F245 transceiver delay = 7 ns maximum

#5) CPU data setup time = 10 ns minimum

"tRAC" = (S0 + S1) + (S2 + S3) + (S4 + S5) + S6 (minimum) − #1 − #2 − #4 − #5
= 91 + 91 + 91 + 35 − 161 − 20 − 7 − 10
= 110 ns

"tCAC" = (S0 + S1) + (S2 + S3) + (S4 + S5) + S6 (minimum) − #1 − #3 − #4 − #5
= 91 + 91 + 91 + 35 − 161 − 77 − 7 − 10
= 53 ns

Therefore the DRAM chosen should have a "tRAC" less than or equal to 110 ns and a "tCAC" less than or equal to 53 ns. Standard 100 ns DRAMs meet this criteria.

68000 Family 12 MHz No Wait State Calculations

#1) $\overline{RASIN}$ low = S0 + S1 + $\overline{AS}$ low (maximum) + "B" PAL combinational output delay maximum = 83 + 55 + 15 = 153 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 80 ns (DP8419-80 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMs speced in data sheet) = 77 ns

#4) 74F245 transceiver delay = 7 ns maximum

#5) CPU data setup time = 10 ns minimum

"tRAC" = (S0 + S1) + (S2 + S3) + (S4 + S5) + S6 (minimum) − #1 − #2 − #4 − #5
= 83.3 + 83.3 + 83.3 + 35 − 153 − 20 − 7 − 10 = 95 ns

"tCAC" = (S0 + S1) + (S2 + S3) + (S4 + S5) + S6 (minimum) − #1 − #3 − #4 − #5
= 83.3 + 83.3 + 83.3 + 35 − 153 − 77 − 7 − 10 = 38 ns

Therefore the DRAM chosen should have a "tRAC" less than or equal to 95 ns and a "tCAC" less than or equal to 38 ns.

III) 68020 "TRAC" AND "TCAC" Calculations

68020 6 MHz No Wait State Calculations

#1) $\overline{RASIN}$ low = S0 + S1 + "B" PAL combinational output delay maximum = 167 + 15 = 182 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 80 ns (DP8419-80 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMS speced in data sheet) = 77 ns

#4) 74F244 transceiver delay = 7 ns maximum

#5) CPU data setup time = 10 ns minimum

"tRAC" = (S0 + S1) + (S2 + S3) + S4 (minimum) − #1 − #2 − #4 − #5
= 167 + 167 + 75 − 182 − 20 − 7 − 10 = 190 ns

"tCAC" = (S0 + S1) + (S2 + S3) + S4 (minimum) − #1 − #3 − #4 − #5
= 167 + 167 + 75 − 182 − 77 − 7 − 10 = 133 ns

Therefore the DRAM chosen should have a "tRAC" less than or equal to 190 ns and a "tCAC" less than or equal to 133 ns. Standard 150 ns DRAMs meet this criteria.

68020 7 MHz No Wait State Calculations

#1) $\overline{RASIN}$ low = S0 + S1 + "B" PAL combinational output delay maximum = 143 + 15 = 158 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 80 ns (DP8419-80 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMS speced in data sheet) = 77 ns

#4) 74F244 transceiver delay = 7 ns maximum

#5) CPU data setup time = 10 ns minimum

"tRAC" = (S0 + S1) + (S2 + S3) + S4 (minimum) − #1 − #2 − #4 − #5
= 143 + 143 + 60 − 158 − 20 − 7 − 10 = 151 ns

"tCAC" = (S0 + S1) + (S2 + S3) + S4 (minimum) − #1 − #3 − #4 − #5
= 143 + 143 + 60 − 158 − 77 − 7 − 10 = 94 ns

Therefore the DRAM chosen should have a "tRAC" less than or equal to 151 ns and a "tCAC" less than or equal to 94 ns. Standard 150 ns DRAMs meet this criteria.

68020 8 MHz No Wait State Calculations

#1) $\overline{RASIN}$ low = S0 + S1 + "B" PAL combinational output delay maximum = 125 + 15 = 140 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 80 ns (DP8419-80 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMS speced in data sheet) = 77 ns

#4) 74F244 transceiver delay = 7 ns maximum

#5) CPU data setup time = 10 ns minimum

"tRAC" = (S0 + S1) + (S2 + S3) + S4 (minimum) − #1 − #2 − #4 − #5
= 125 + 125 + 55 − 140 − 20 − 7 − 10 = 128 ns

"tCAC" = (S0 + S1) + (S2 + S3) + S4 (minimum) − #1 − #3 − #4 − #5
= 125 + 125 + 55 − 140 − 77 − 7 − 10 = 71 ns

Therefore the DRAM chosen should have a "tRAC" less than or equal to 128 ns and a "tCAC" less than or equal to 71 ns. Standard 120 ns DRAMs meet this criteria.

68020 9 MHz No Wait State Calculations

#1) $\overline{RASIN}$ low = S0 + S1 + "B" PAL combinational output delay maximum = 111 + 15 = 131 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to CAS low = 80 ns (DP8419-80 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMS speced in data sheet) = 77 ns

#4) 74F244 transceiver delay = 7 ns maximum

#5) CPU data setup time = 10 ns minimum

"tRAC" = (S0 + S1) + (S2 + S3) + S4 (minimum) − #1 − #2 − #4 − #5
= 111 + 111 + 50 − 131 − 20 − 7 − 10
= 104 ns

"tCAC" = (S0 + S1) + (S2 + S3) + S4 (minimum) − #1 − #3 − #4 − #5
= 111+ 111 + 50 − 131 − 77 − 7 − 10
= 47 ns

Therefore the DRAM chosen should have a "tRAC" less than or equal to 104 ns and a "tCAC" less than or equal to 47 ns.

IV) 68020 with 1 Wait State Inserted "tRAC" and "tCAC" Calculations

68020 10 MHz (1 Wait State) Calculations

#1) $\overline{RASIN}$ low = S0 + S1 + "B" PAL combinational output delay maximum = 100 + 15 = 115 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to CAS low = 80 ns (DP8419-80 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMS speced in data sheet) = 77 ns

#4) 74F244 transceiver delay = 7 ns maximum

#5) CPU data setup time = 10 ns minimum

"tRAC" = (S0 + S1) + (S2 + S3) + (SW + SW) + S4 (minimum) − #1 − #2 − #4 − #5
= 100 + 100 + 100 + 45 − 115 − 20 − 7 − 10
= 193 ns

"tCAC" = (S0 + S1) + (S2 + S3) + (SW + SW) + S4 (minimum) − #1 − #3 − #4 − #5
= 100 + 100 + 100 + 45 − 115 − 77 − 7 − 10
= 136 ns

Therefore the DRAM chosen should have a "tRAC" less than or equal to 193 ns and a "tCAC" less than or equal to 136 ns. Standard 150 ns DRAMs meet this criteria.

68020 12 MHz (1 Wait State) Calculations

#1) $\overline{RASIN}$ low = S0 + S1 + "B" PAL combinational output delay maximum = 80 + 15 = 95 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 80 ns (DP8419-80 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMS speced in data sheet) = 77 ns

#4) 74F244 transceiver delay = 7 ns maximum

#5) CPU data setup time = 10 ns minimum

"tRAC" = (S0 + S1) + (S2 + S3) + (SW + SW) + S4 (minimum) − #1 − #2 − #4 − #5
= 83.3 + 83.3 + 83.3 + 35 − 95 − 20 − 7 − 10
= 153 ns

"tCAC" = (S0 + S1) + (S2 + S3) + (SW + SW) + S4 (minimum) − #1 − #3 − #4 − #5
= 83.3 + 83.3 + 83.3 + 35 − 95 − 77 − 7 − 10
= 96 ns

Therefore the DRAM chosen should have a "tRAC" less than or equal to 153 ns and a "tCAC" less than or equal to 96 ns. Standard 150 ns DRAMs meet this criteria.

68020 14 MHz (1 Wait State) Calculations

#1) $\overline{RASIN}$ low = S0 + S1 + "B" PAL combinational output delay maximum = 72 + 15 = 87 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 80 ns (DP8419-80 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMs speced in data sheet) = 77 ns

#4) 74F244 transceiver delay = 7 ns maximum

#5) CPU data setup time = 5 ns minimum

"tRAC" = (S0 + S1) + (S2 + S3) + (SW + SW) + S4 (minimum) − #1 − #2 − #4 − #5
= 72 + 72 + 72 + 30 − 87 − 20 − 7 − 5
= 127 ns

"tCAC" = (S0 + S1) + (S2 + S3) + (SW + SW) + S4 (minimum) − #1 − #3 − #4 − #5
= 72 + 72 + 72 + 30 − 87 − 77 − 7 − 5
= 70 ns

Therefore the DRAM chosen should have a "tRAC" less than or equal to 127 ns and a "tCAC" less than or equal to 70 ns. Standard 120 ns DRAMs meet this criteria.

68020 16 MHz (1 Wait State) Calculations

#1) $\overline{RASIN}$ low = S0 + S1 + "B" PAL combinational output delay maximum = 62.5 + 15 = 77.5 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 80 ns (DP8419-80 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMS speced in data sheet) = 77 ns

#4) 74F244 transceiver delay = 7 ns maximum

#5) CPU data setup time = 5 ns minimum

"tRAC" = (S0 + S1) + (S2 + S3) + (SW + SW) + S4 (minimum) − #1 − #2 − #4 − #5
= 62.5 + 62.5 + 62.5 + 25 − 77.5 − 20 − 7 − 5
= 103 ns

"tCAC" = (S0 + S1) + (S2 + S3) + (SW + SW) + S4 (minimum) − #1 − #3 − #4 − #5
= 62.5 + 62.5 + 62.5 + 25 − 77.5 − 77 − 7 − 5
= 46 ns

Therefore the DRAM chosen should have a "tRAC" less than or equal to 103 ns and a "tCAC" less than or equal to 46 ns.

V) iAPX 86/88/186/188 Family "tRAC" and "tCAC" Calculations

iAPX 86/88 8 MHz No Wait State Calculations

#1) $\overline{\text{RASIN}}$ low = Maximum clock high + 15 ns ("B" PAL combinational output delay) = 82 + 15 = 97 ns maximum

#2) $\overline{\text{RASIN}}$ to $\overline{\text{RAS}}$ low = 20 ns maximum

#3) $\overline{\text{RASIN}}$ to $\overline{\text{CAS}}$ low = 97 ns (DP8419-80 $\overline{\text{RASIN}}$ − $\overline{\text{CAS}}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMs speced in data sheet) = 94 ns maximum (using 25 ns minimum row address hold time)

#4) 74F245 transceiver delay = 7 ns maximum

#5) CPU data setup time to "T4" = 20 ns minimum

"tRAC" = T1 + T2 + T3 − #1 − #2 − #4 − #5
= 125 + 125 + 125 − 97 − 20 − 7 − 20
= 231 ns

"tCAC" = T1 + T2 + T3 − #1 − #3 − #4 − #5
= 125 + 125 + 125 − 97 − 94 − 7 − 20
= 157 ns

Therefore the DRAM chosen should have a "tRAC" less than or equal to 231 ns and a "tCAC" less than or equal to 157 ns. Standard 200 ns DRAMs meet this criteria.

iPX 186/188 8 MHz No Wait State Calculations

#1) $\overline{\text{RASIN}}$ low = Maximum clock high + 15 ns ("B" PAL combinational output delay) = 70 + 15 = 85 ns maximum

#2) $\overline{\text{RASIN}}$ to $\overline{\text{RAS}}$ low = 20 ns maximum

#3) $\overline{\text{RASIN}}$ to $\overline{\text{CAS}}$ low = 97 ns (DP8419-80 $\overline{\text{RASIN}}$ − $\overline{\text{CAS}}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMs speced in data sheet) = 94 ns maximum (using 25 ns minimum row address hold time)

#4) 74F245 transceiver delay = 7 ns maximum

#5) CPU data setup time to "T4" = 20 ns minimum

"tRAC" = T1 + T2 + T3 − #1 − #2 − #4 − #5
= 125 + 125 + 125 − 85 − 20 − 7 − 20
= 243 ns

"tCAC" = T1 + T2 + T3 − #1 − #3 − #4 − #5
= 125 + 125 + 125 − 85 − 94 − 7 − 20
= 169 ns

Therefore the DRAM chosen should have a "tRAC" less than or equal to 243 ns and a "tCAC" less than or equal to 169 ns. Standard 200 ns DRAMs meet this criteria.

iAPX 86/88 10 MHz No Wait State Calculations

#1) $\overline{\text{RASIN}}$ low = Maximum clock high + 15 ns ("B" PAL combinational output delay) = 61 + 15 = 76 ns maximum

#2) $\overline{\text{RASIN}}$ to $\overline{\text{RAS}}$ low = 20 ns maximum

#3) $\overline{\text{RASIN}}$ to $\overline{\text{CAS}}$ low = 80 ns (DP8419-80 $\overline{\text{RASIN}}$ − $\overline{\text{CAS}}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMs speced in data sheet) = 77 ns maximum (using 15 ns minimum row address hold time)

#4) 74F245 transceiver delay = 7 ns maximum

#5) CPU data setup time to "T4" = 5 ns minimum

"tRAC" = T1 + T2 + T3 − #1 − #2 − #4 − #5
= 100 + 100 + 100 − 76 − 20 − 7 − 5
= 192 ns

"tCAC" = T1 + T2 + T3 − #1 − #3 − #4 − #5
= 100 + 100 + 100 − 76 − 77 − 7 − 5
= 135 ns

Therefore the DRAM chosen should have a "tRAC" less than or equal to 192 ns and a "tCAC" less than or equal to 135 ns. Standard 150 ns DRAMs meet this criteria.

VI) iAPX 286 "tRAC" and "tCAC" Calculations

6 MHz iAPX 286, 12 MHz Clock, No Wait State Calculations

#1) $\overline{\text{RASIN}}$ low = T1 + 74AS04 gate delay + "B" PAL clocked output delay = 83.3 + 4.5 + 12 = 100 ns maximum

#2) $\overline{\text{RASIN}}$ to $\overline{\text{RAS}}$ low = 20 ns maximum

#3) $\overline{\text{RASIN}}$ to $\overline{\text{CAS}}$ low = 80 ns (DP8419-80 $\overline{\text{RASIN}}$ − $\overline{\text{CAS}}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMs speced in data sheet) = 77 ns maximum (using 15 ns minimum row address hold time)

#4) 74F244 transceiver delay = 7 ns maximum

#5) CPU data setup time to "T4" = 20 ns minimum

"tRAC" = T1 + T2 + T3 + T4 − #1 − #2 − #4 − #5
= 83.3 + 83.3 + 83.3 + 83.3 − 100 − 20 − 7 − 20 = 186 ns

"tCAC" = T1 + T2 + T3 + T4 − #1 − #3 − #4 − #5
= 83.3 + 83.3 + 83.3 + 83.3 − 100 − 77 − 7 − 20 = 129 ns

Therefore the DRAM chosen should have a "tRAC" less than or equal to 186 ns and a "tCAC" less than or equal to 129 ns. Standard 150 ns DRAMs meet this criteria.

7 MHz iAPX 286, 14 MHz Clock, No Wait State Calculations

#1) $\overline{\text{RASIN}}$ low = T1 + 74AS04 gate delay + "B" PAL clocked output delay = 71.4 + 4.5 + 12 = 88 ns maximum

#2) $\overline{\text{RASIN}}$ to $\overline{\text{RAS}}$ low = 20 ns maximum

#3) $\overline{\text{RASIN}}$ to $\overline{\text{CAS}}$ low = 80 ns (DP8419-80 $\overline{\text{RASIN}}$ − $\overline{\text{CAS}}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMs speced in data sheet) = 77 ns maximum (using 15 ns minimum row address hold time)

#4) 74F244 transceiver delay = 7 ns maximum

#5) CPU data setup time to "T4" = 10 ns minimum

"tRAC" = T1 + T2 + T3 + T4 − #1 − #2 − #4 − #5
= 71.4 + 71.4 + 71.4 + 71.4 − 88 − 20 − 7 − 10 = 160 ns

"tCAC" = T1 + T2 + T3 + T4 − #1 − #3 − #4 − #5
= 71.4 + 71.4 + 71.4 + 71.4 − 88 − 77 − 7 − 10 = 103 ns

Therefore the DRAM chosen should have a "tRAC" less than or equal to 160 ns and a "tCAC" less than or equal to 103 ns. Standard 150 ns DRAMs meet this criteria.

8 MHz IAPX 286, 16 MHz Clock, No Wait State Calculations

#1) $\overline{RASIN}$ low = T1 + 74AS04 gate delay + "B" PAL clocked output delay = 62.5 + 4.5 + 12 = 79 ns maximum

#2) $\overline{RASIN}$ to $\overline{RAS}$ low = 20 ns maximum

#3) $\overline{RASIN}$ to $\overline{CAS}$ low = 80 ns (DP8419-80 $\overline{RASIN}$ − $\overline{CAS}$ low) − 3 ns (load of 72 DRAMs instead of 88 DRAMs speced in data sheet) = 77 ns maximum (using 15 ns minimum row address hold time)

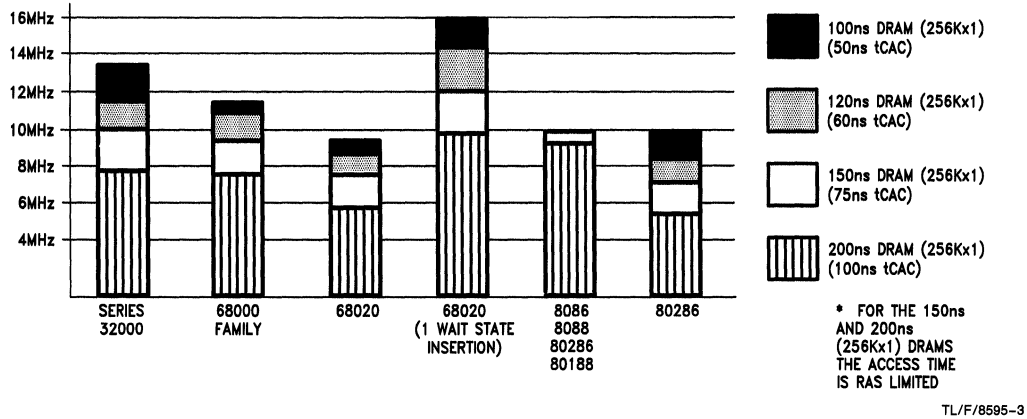
#4) 74F244 transceiver delay = 7 ns maximum

#5) CPU data setup time to "T4" = 10 ns minimum

"tRAC" = T1 + T2 + T3 + T4 − #1 − #2 − #4 − #5
= 62.5 + 62.5 + 62.5 + 62.5 − 79 − 20 − 7 − 10 = 134 ns

"tCAC" = T1 + T2 + T3 + T4 − #1 − #3 − #4 − #5
= 62.5 + 62.5 + 62.5 + 62.5 − 79 − 77 − 7 − 10 = 77 ns

Therefore the DRAM chosen should have a "tRAC" less than or equal to 134 ns and a "tCAC" less than or equal to 77 ns. Standard 120 ns DRAMs meet this criteria.



TL/F/8595–3

FIGURE 3

Note 1: The data presented in this figure is based on typical examples. Faster "no wait state" CPU performance is possible with several of the microprocessors shown above via the use of the DP8419-70 instead of the DP8419-80; the elimination of Data Bus Transceivers; a more tailored PAL (Refresh Access Arbitrator) approach; faster support logic; lower than the 15Ω damping resistor specified in the DP8419-80 data sheet; or, less than the specified capacitive load driven directly by the DP8419 (88 DRAMs).

Dual Port Interface for the DP8417/18/19/28/29 DRAM Controller

National Semiconductor
Application Note 436
Webster (Rusty) B. Meier

INTRODUCTION

This application note describes a general purpose dual port interface to the DP8417/18/19/28/29 DRAM controller. A PAL® (Programmable Array Logic) device is used to implement this interface. The PAL contains the logic necessary to arbitrate between the three ports (Refresh, Port A, and Port B), provide WAIT states to Port A or B when necessary, and an output to multiplex the Port A or B addresses to the DRAM controller.

FEATURES

- Provides a versatile dual port interface to the DP8417/18/19/28/29 DRAM controller
- Provides arbitration circuitry between DRAM refresh cycles, Port A accesses, and Port B accesses
- Allows for Port A and Port B to be synchronous or asynchronous to the input system clock
- Guarantees a minimum of one and one half system clock periods of RAS precharge time between grants to any two ports
- Provides WAIT state logic to both PORT A and Port B to handle contention problems between ports
- Differentiates between READ and WRITE accesses for Port A allowing Port A WRITE accesses to begin later than READ accesses

DESCRIPTION

This hardware arbitrates access to the dynamic RAM controlled by the DP8419 (or any of the related family members: DP8417/18/19/28/29) to either:

1) A Refresh cycle, "$\overline{\text{GRNTRF}}$"

2) Port A, "$\overline{\text{GRNTA}}$"

3) Port B, "$\overline{\text{GRNTB}}$"

Refresh always has the highest priority and will always occur immediately upon a refresh request ($\overline{\text{RFRQ}}$) given that an access by Port A or B is not currently in progress. Port A has a higher priority than Port B though the scheme used attempts to give both ports a more equal priority. The arbiter does this by leaving Port A or Port B granted, after an access by that particular port, as long as no other ports are currently trying to access the DRAM. This scheme is used because data tends to be transferred in bursts from a particular port.

Once a port is granted, subsequent requests by that port immediately access the DRAM, until another port gains access to the DRAM (see *Figure 11* of the timing waveforms for Port A).

The term "$\overline{\text{WINA}}$" (write enable for Port A) is used to cause "$\overline{\text{RASIN}}$" to be generated later for a WRITE access than a READ access. This may be necessary to guarantee that valid data is written to the DRAM during WRITE accesses. If Port B is asynchronous this input is not needed because Port B requests are delayed through the external synchronization circuitry. If Port B is synchronous both ports should mux to the "$\overline{\text{WIN}}$" input, and use this input in generating the "$\overline{\text{RASIN}}$" output of the PAL.

This arbiter guarantees one and one half system clock periods of RAS precharge between accesses of different ports. It is up to the user to guarantee the precharge time between consecutive accesses from the same port. This arbiter assumes a minimum of one period high time between access requests from a particular port.

Hidden Refresh is not supported in any of the following dual port schemes for several reasons:

1) If "$\overline{\text{CS}}$", of the DP8419, is not permanently tied low the user must guarantee a "$\overline{\text{CS}}$-$\overline{\text{RASIN}}$" minimum time of 34 ns for the DP8419. This could slow down the access time of several of the dual port schemes presented.

2) In order to do hidden refresh a port must be granted during a non-$\overline{\text{CS}}$ access cycle. When the port is granted during a non-$\overline{\text{CS}}$ access cycle the other port may be requesting the dual ported memory also and have to wait for it. A possible problem is that the non-$\overline{\text{CS}}$ access may not even be causing a hidden refresh at that time so in essence the other port is being slowed down for no reason (i.e. a hidden or forced refresh may have already been done during that period of the refresh clock).

If either Port A or B tries to access the DRAM during a refresh WAIT states will automatically be inserted into that port's access cycle. Also if one of the ports tries to access the DRAM while the other port is, WAIT states will automatically be inserted into the appropriate port's access cycle. The user may want to change the "$\overline{\text{WAIT}}$" state equations depending upon the processor or bus being interfaced to.

The DUAL PORT ARBITER gives access to the refresh cycle via the M2 ($\overline{\text{RFSH}}$) pin of the DP8419. The $\overline{\text{GRNTB}}$ output of the DUAL PORT CONTROLLER acts as a multiplexor signal to enable either PORT A or PORT B. Once enabled the Port selected will enable its addresses, write enable, $\overline{\text{LOCK}}$ control signal, and data to the DP8419 and its controlled memory. The user must be careful to assure that a particular port will not be locked ("$\overline{\text{LOCK}}$" low while "$\overline{\text{GRNTA}}$ or B" is low) for more than 15.6 μs (RFCK period) or the system may miss a refresh.

The Dual Port scheme presented assumes that all "PORT REQUEST" inputs are synchronous to the system clock input to the PAL (i.e. "PORT REQUESTs" occur following a rising edge of the system clock). If a specific "PORT REQUEST" is asynchronous to the system clock it has to be synchronized to the system clock by running it through two flip-flops (see "AREQB" and "ARFRQ" in the system block diagram). The two "$\overline{\text{RFRQ}}$" synchronizing flip-flops are needed for the PAL refresh logic to work correctly.

The Dual Port scheme presented does not assume the use of any specific processor. Therefore, the user may require some external logic to interface the Dual Port PAL to a specific microprocessor or bus.

Figures 1-5 show several suggestions for circuits used to generate "$\overline{\text{REQA}}$" for different CPU's. The PAL equations were designed assuming a National Semiconductor Series 32000® CPU on Port A. In the "$\overline{\text{RASIN}}$" equations for Port A WRITE cycles were started one half period later than READ

cycles and both READ and WRITE accesses were ended one half period after "$\overline{\text{REQA}}$" went high (this is to make up for WRITE accesses starting one half period after "$\overline{\text{REQA}}$"). The user may wish to modify these equations (and possibly the "$\overline{\text{WAITA}}$" equations) depending upon the specific CPU being used.

EXAMPLE: DETERMINING THE REQUIRED MEMORY SPEED ("t_{RAC}" AND "t_{CAC}") FOR A SERIES 32000 TO RUN AT 10 MHz WITHOUT WAIT STATES

Assume the Series 32000 is synchronously interfaced to Port A.

#1) $\overline{\text{RASIN}}$ low = T1 + 6 ns (PHI1 to CTTL Rising edge maximum) + 12 ns ("B" PAL clocked output) + 15 ns ("B" PAL combinational output) = 100 + 6 + 12 + 15 = 133 ns maximum

#2) $\overline{\text{RASIN}}$ to $\overline{\text{RAS}}$ low = 20 ns maximum

#3) $\overline{\text{RASIN}}$ to $\overline{\text{CAS}}$ low = 70 ns (DP8419-70) − 3 ns (72 DRAMs instead of 88 DRAMs spec'd in data sheet) = 67 ns maximum

#4) 74F245 transceiver delay = 7 ns maximum

#5) CPU data setup time to "T4" clock cycle = 15 ns maximum

"t_{RAC}" = T1 + T2 + T3 − #1 − #2 − #4 − #5
= 100 + 100 + 100 − 133 ns − 20 − 7 − 15 = 125 ns

"t_{CAC}" = T1 + T2 + T3 − #1 − #3 − #4 − #5
= 100 + 100 + 100 − 133 ns − 67 − 7 − 15 = 78 ns

Therefore the DRAM chosen should have a "t_{RAC}" less than or equal to 125 ns and a "t_{CAC}" less than or equal to 78 ns. Standard 120 ns DRAMs meet this criteria.

The following is an example of how to interpret the PAL equations correctly. These equations are presented in the format specified by the National Semiconductor PLAN format. CAUTION, this format differs from the much used PALASM format.

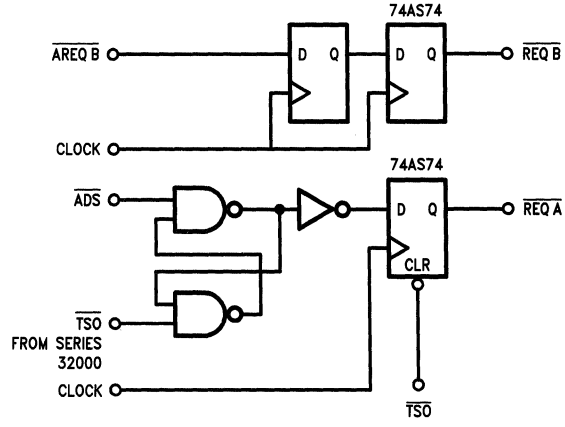
EXAMPLE: $\overline{\text{GRNTRF}}$:= $\overline{\text{RFRQ}}$*GRNTA*GRNTB

This reads, the active low flip-flop output "$\overline{\text{GRNTRF}}$" is low following the rising edge of the input clock given that, the active low input "$\overline{\text{RFRQ}}$" is low AND the active low output "$\overline{\text{GRNTA}}$" is high AND the active low output "$\overline{\text{GRNTB}}$" is high a setup time before the input clock transitions high. (Notice that $\overline{\text{RFRQ}}$ is interpreted as being low.)

POSSIBLE MODIFICATIONS TO THIS APPLICATION

In this application "$\overline{\text{REQB}}$" is synchronized to the falling edge of the system clock input of the PAL. Generating "$\overline{\text{REQB}}$" from the falling clock edge allows minimum delay from the asynchronous request to the synchronized request producing "$\overline{\text{GRNTB}}$" and or "$\overline{\text{RASIN}}$". Producing "$\overline{\text{REQB}}$" in this way also delays "$\overline{\text{RASIN}}$" during a port B access because of the effect of the "$\overline{\text{GTOA}}$" term. In order to calculate the t_{RAC} and t_{CAC} of the DRAM (see Series 32000 example above) the delay to "$\overline{\text{RASIN}}$" low would be: "$\overline{\text{AREQB}}$" low (asynchronous request B) + SYNCHRONIZATION delay (2 flip-flops) + 3 input NAND gate delay of "$\overline{\text{GTOA}}$" + PAL delay for "$\overline{\text{RASIN}}$".

If "$\overline{\text{REQB}}$" is synchronized to the rising edge of the system clock there is a potential danger of getting glitches on the "$\overline{\text{RASIN}}$" output of the PAL as a result of the "$\overline{\text{GTOA,B}}$" terms. The glitches are possible under the condition of both "$\overline{\text{REQA}}$" and "$\overline{\text{REQB}}$" going low during a single clock period. For example, if Port B is currently granted ("$\overline{\text{GRNTB}}$" low) and "$\overline{\text{REQA}}$" goes low more than one inverter gate delay before "$\overline{\text{REQB}}$" goes low the "$\overline{\text{GTOA}}$" term will initially be high, then go low, then back high. This could cause a small glitch at the beginning of "$\overline{\text{RASIN}}$". This glitch can be avoided by guaranteeing that either the requests are separated by at least a three input NAND gate delay (as is the case in this application note) or that when two requests happen within one clock period they happen within one inverter gate delay of each other. The circuits shown below, in *Figure 1*, could be used to guarantee that when two requests happen within one clock they occur within one gate delay of each other.

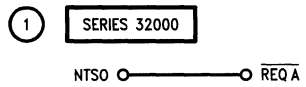


TL/F/8678–1

FIGURE 1. Alternative Request Generating Circuits

IDEAS ON GENERATING "$\overline{\text{REQA}}$" FOR SEVERAL DIFFERENT MICROPROCESSORS.

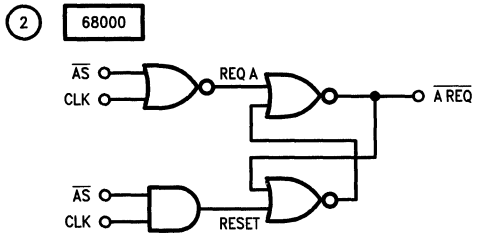
*$\overline{\text{REQA}}$, $\overline{\text{REQB}}$, $\overline{\text{RFRQ}}$ should have a minimum setup time of approximately 20 ns before the rising edge of the system clock.



TL/F/8678–2

FIGURE 2. Series 32000 "$\overline{\text{REQA}}$"

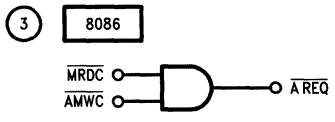
Minimum of 2 periods $\overline{\text{RAS}}$ precharge between successive accesses.



TL/F/8678–3

FIGURE 3. 68000 "$\overline{\text{REQA}}$"

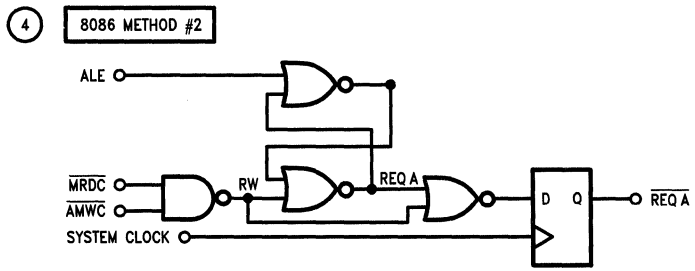
Minimum of 1½ periods of $\overline{\text{RAS}}$ precharge.



TL/F/8678–4

FIGURE 4. 8086 "$\overline{\text{REQA}}$" Method #1

Minimum of 2 periods of $\overline{\text{RAS}}$ precharge.



TL/F/8678–5

FIGURE 5. 8086 "$\overline{\text{REQA}}$" Method #2

(For faster speed, minimum of 1 period of $\overline{\text{RAS}}$ precharge.)

DUAL PORT PAL #1 INPUTS

1) "CLOCK" System clock.
2) "$\overline{\text{REQA}}$" A synchronous access request from Port A.
3) "$\overline{\text{WINA}}$" WRITE ENABLE from Port A. This input is used to delay "$\overline{\text{RASIN}}$" during WRITE accesses.
4) "$\overline{\text{REQB}}$" A synchronous chip selected access request form Port B. "$\overline{\text{AREQB}}$" is run through two flip-flops to get "$\overline{\text{REQB}}$". Chip Select for Port B is assumed to be included within this input.
5) "$\overline{\text{RFRQ}}$" A synchronous refresh request.
6) "$\overline{\text{LOCK}}$" The "$\overline{\text{LOCK}}$" input is an active low signal that is driven by either Port A or Port B. This input, when low, causes the arbiter to keep the currently granted Port granted until the "$\overline{\text{LOCK}}$" input goes high. This input is useful in implementing atomic operations such as semaphores that are useful in multiuser/multitasking operating systems.
7) "$\overline{\text{GTOA}}$" This input is generated externally using the three signals $\overline{\text{REQA}}$, $\overline{\text{REQB}}$, and $\overline{\text{LOCK}}$ with some discrete logic. This input indicates that the arbiter will switch to Port A, given that Port B is currently granted. This input is needed to guarantee that when the arbiter switches control of the DRAM from Port B to Port A that $\overline{\text{GRNTB}}$ goes invalid before $\overline{\text{REQB}}$ is able to start another access (see the $\overline{\text{RASIN}}$ output term "PORTB RASIN" in PAL equations).

8) "GTOB" This input is generated externally using the three signals REQA, REQB, and LOCK with some discrete logic. This input indicates that the arbiter will switch to Port B given that Port A is currently granted. This input is needed to guarantee that when the arbiter switches control of the DRAM from Port A to Port B that GRNTA goes invalid before REQA is able to start another access (see the RASIN output term "PORTA RASIN" in PAL equations).
9) "CLK" This is the system clock input that may be used in the PAL equations (i.e. "WAIT").
10) "CSA" This input is the chip select input for Port A. It is used, along with "REQA", to request and cause an access to the DRAM.

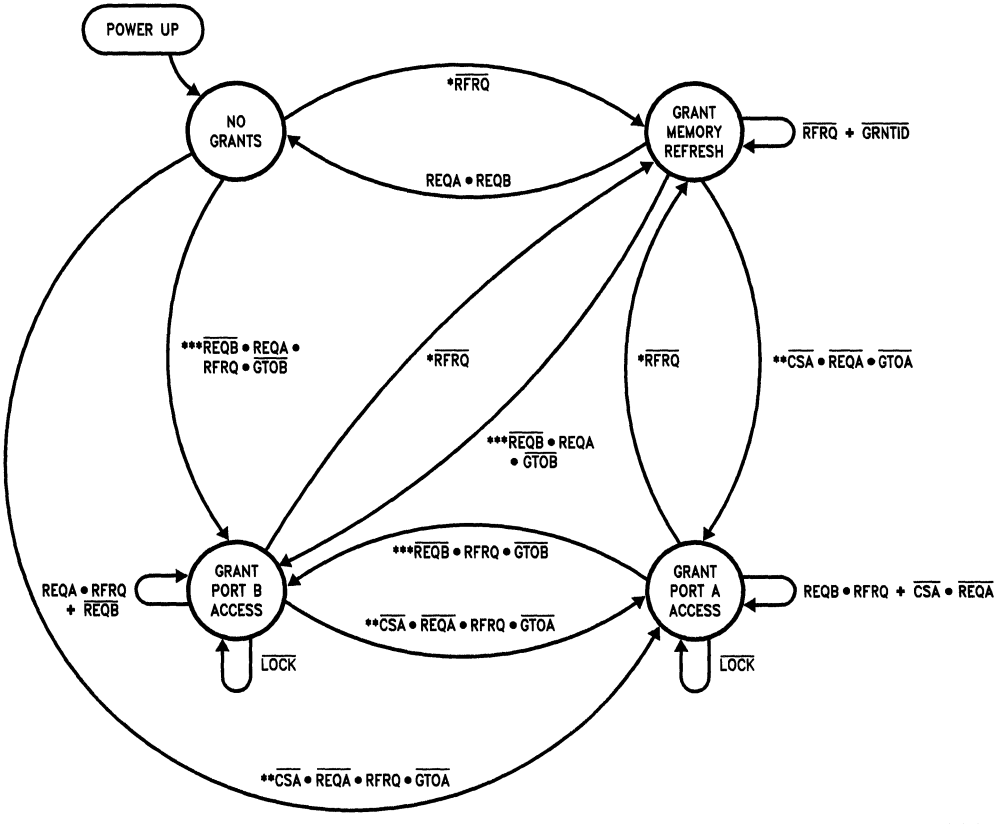
DUAL PORT PAL #1, OUTPUTS

NOTE: All outputs are active low.

1) "RASIN" This is the RASIN input to the DP8419 for Port A, Port B, and refresh.
2) "GRNTA" This output is the grant output for Port A.
3) "GRNTB" This output functions as the grant output for both Port A (high) and Port B (low).
4) "GRNTRF" Goes to DP8419 M2 (RFSH) input. This causes an automatic forced refresh cycle.
5) "GRNT1D" Goes low one period after "GRNTA", "GRNTB", or "GRNTRF" go low. This output is used to guarantee that one period is allowed after arbitration before a "RASIN" is generated during a port access. This allows the particular port's address, write enable signal, and lock input to become valid before an access is started. This output also allows the PAL to determine when a particular port has been granted for several system clock periods. This information allows the arbiter to immediately generate "RASIN" for any subsequent memory accesses since the address is already muxed to the DRAM controller (see *Figure 11* for the timing waveforms for Port A).
6) "WAITA" This output functions as a WAIT input for Port A.
7) "GTORFSH" This input is generated internally and indicates that the arbiter will give access control over to the refresh Port at the next rising clock edge.
8) "XACKB" This output is generated external to the PAL and functions as a transfer acknowledge for Port B.

```
DUAL PORT PAL #1
PAL16R4B

CLOCK  /REQA  /WINA  /REQB  /RFRQ  /LOCK  /GTOA  /GTOB  CLK  GND
/OE  /CSA  /WAITA  /GRNT1D  /GRNTRF  /GRNTB  /GRNTA  /RASIN  /GTORFSH  VCC

/GRNTA  := /CSA*/REQA*GRNTRF*RFRQ*GRNTB          ;Start GRNTA
           +/LOCK*/GRNTA                         ;Continue GRNTA
           +/CSA*/REQA*RFRQ*/GRNTRF*GRNT1D       ;RFSH_TO_PORTA
           +/CSA*/GTOA*/*GRNTB*RFRQ*RASIN        ;PORTB_TO_PORTA
           +/CSA*/REQA*/GRNTA                    ;Hold GRNTA
           +/GRNTA*REQB*RFRQ                     ;Hold GRNTA

/GRNTB  := REQA*GRNTA*RFRQ*GRNTRF*/REQB          ;Start GRNTB
           +/LOCK*/GRNTB                         ;Continue GRNTB
           +/GTOB*/GRNTA*RFRQ*RASIN              ;PORTA_TO_PORTB
           +REQA*RFRQ*/GRNTRF*/REQB*GRNT1D       ;RFSH_TO_PORTB
           +/REQB*/GRNTB                         ;Hold GRNTB
           +/GRNTB*REQA*RFRQ                     ;Hold GRNTB
           +/GRNTB*CSA*RFRQ                      ;Hold GRNTB

/GRNTRF := GRNTA*GRNTB*/RFRQ                     ;Start GRNTRF
           +/GRNTRF*/RFRQ                        ;Continue GRNTRF
           +REQA*/GRNTA*LOCK*/RFRQ               ;PORTA_TO_RFSH
           +REQB*/GRNTB*LOCK*/RFRQ               ;PORTB_TO_RFSH
           +/GRNTRF*/GRNT1D                      ;Hold GRNTRF

/GRNT1D := /GRNTA*GTOB*GTORFSH                   ;GRNT1D for PORTA
           +/GRNTB*GTOA*GTORFSH                  ;GRNT1D for PORTB
           +/GRNTRF*/RFRQ                        ;GRNT1D for RFSH

IF (VCC) /GTORFSH =
          REQA*/GRNTA*LOCK*/RFRQ                 ;PORTA_TO_RFSH
          +REQB*/GRNTB*LOCK*/RFRQ                ;PORTB_TO_RFSH
```

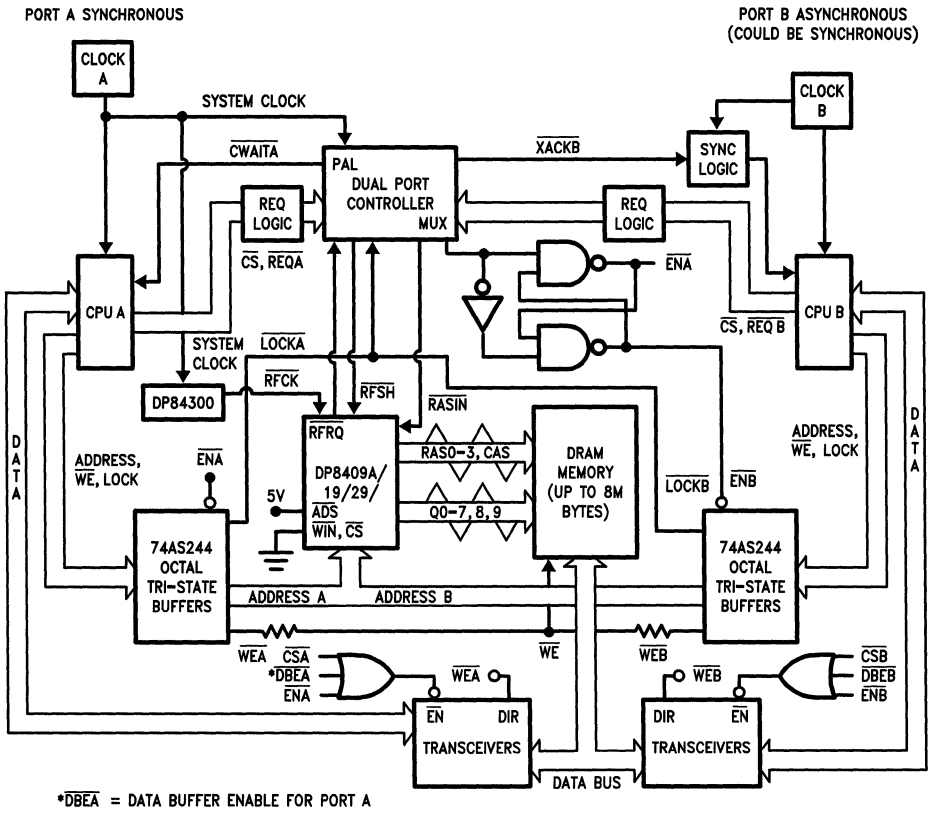
```
DUAL PORT PAL #1,
PAL16R4B (Continued)
IF (VCC) /RASIN =
/CSA*/REQA*/GRNTA*/GRNT1D*GTOB*GTORFSH*WINA          ;PORTA READ RASIN
+/CSA*/REQA*/GRNTA*/GRNT1D*GTOB*GTORFSH*/WINA*/CLK   ;PORTA WRITE RASIN
+/CSA*/REQA*/GRNTA*/GRNT1D*GTOB*GTORFSH*/WINA*/RASIN ;PORTA WRITE RASIN
+/GRNTA*/GRNT1D*/RASIN*CLK                          ;Hold PORTA RASIN
+/REQB*/GRNTB*/GRNT1D*GTOA*GTORFSH                  ;PORTB RASIN
+/GRNTRF*/GRNT1D*GTORFSH                            ;RFSH RASIN

IF (VCC) /WAITA = /CSA*/REQA*GRNTA                 ;Start WAITA
        +/CSA*/REQA*/WAITA*CLK                     ;WAIT until one half period after GRNTA
```



TL/F/8678–6

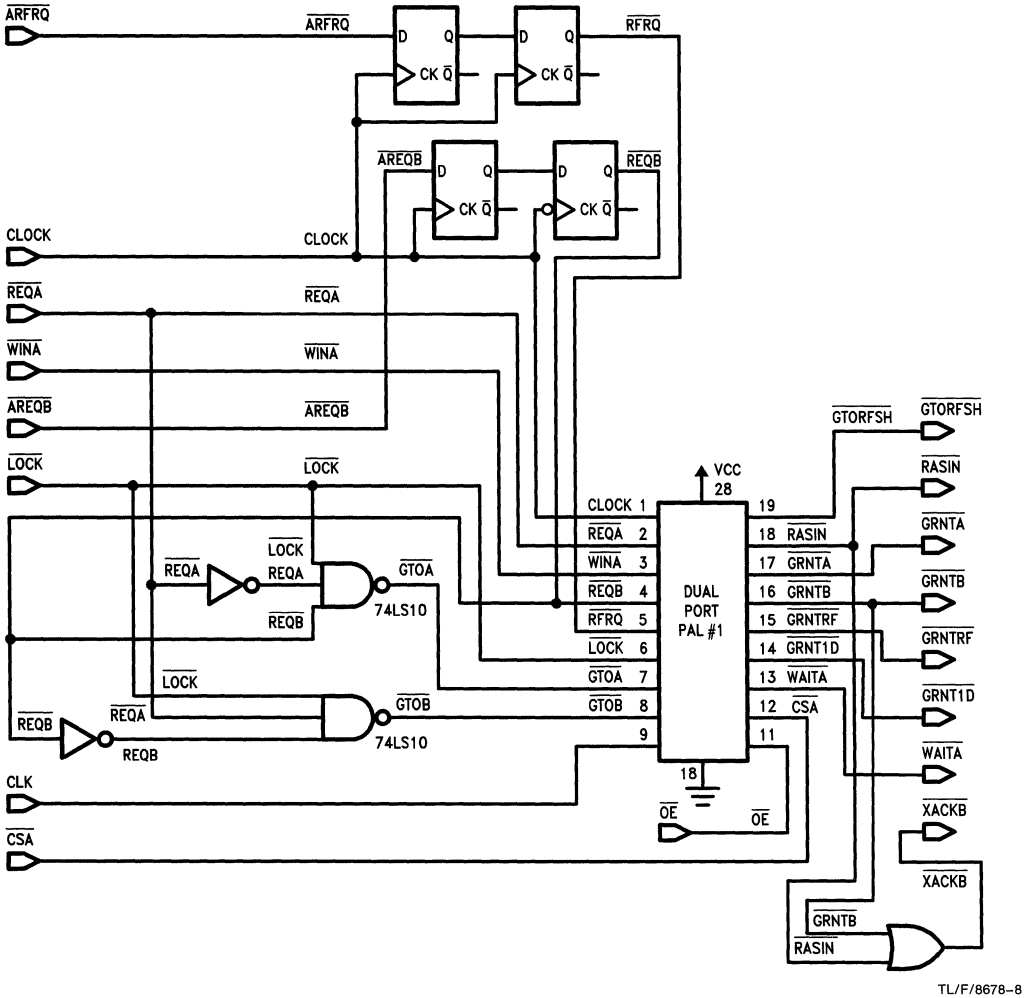
FIGURE 6. Dual Port State Diagram

*Refresh has the highest priority
**Port A has the second highest priority
***Port B has the lowest priority



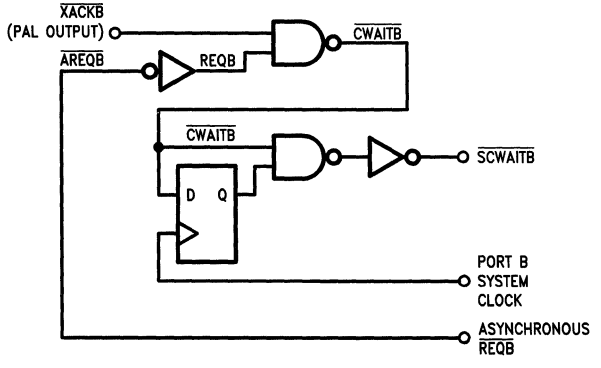
TL/F/8678–7

FIGURE 7. Dual Port Interface



TL/F/8678–8

FIGURE 8. Dual Port PAL Controller Diagram



TL/F/8678–9

FIGURE 9. Asynchronous Port B transfer acknowledge ("XACKB") synchronizes circuit to produce "CWAITB" synchronous with the Port B clock "CTTL" ("SCWAITB")

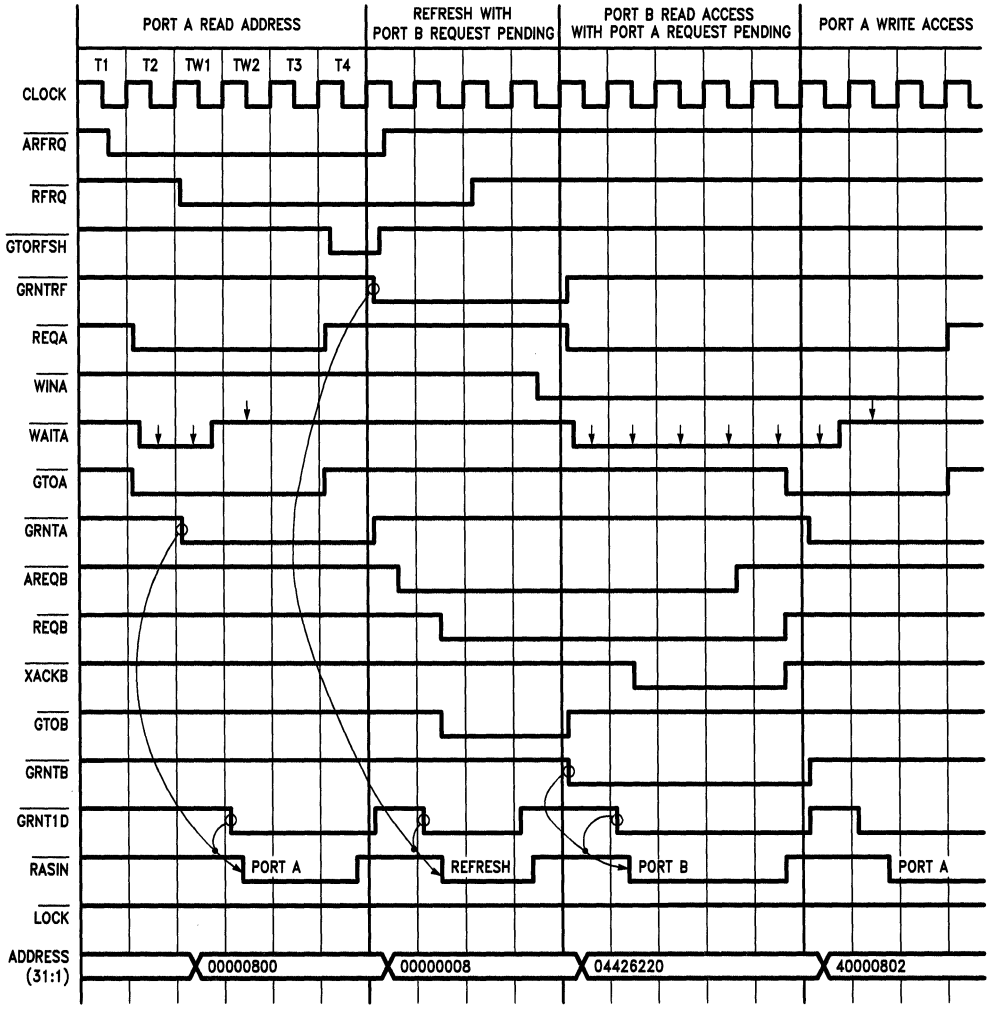
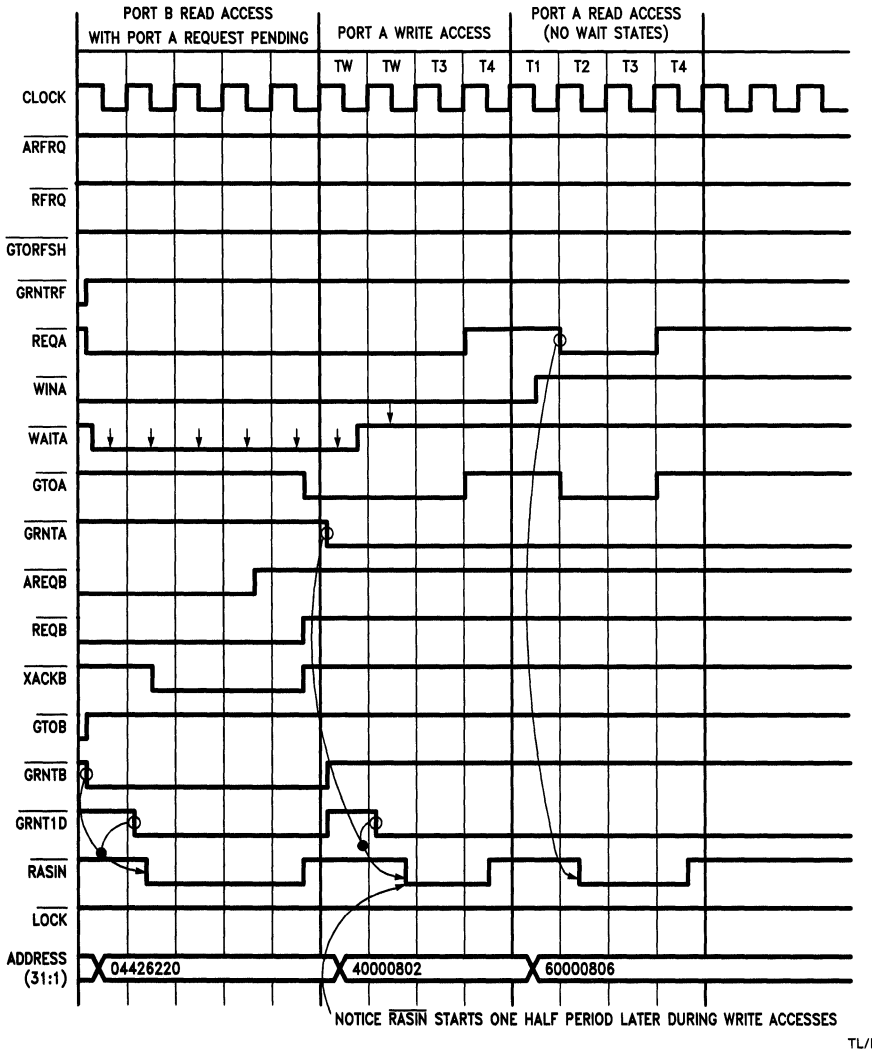


FIGURE 10. Dual Port Timing



TL/F/8678–11

FIGURE 11. Dual Port Timing

Section 4

Microprocessor Applications for the DP8420A/21A/22A

Section 4 Contents

Explanation of National Semiconductor "PLAN™" Software for Programming PAL®s

National Semiconductor
Application Brief 36
Webster (Rusty) Meier Jr. and Joe Tate

INTRODUCTION

The National Semiconductor PLAN software provides interactive design and development tools for system designers who use programmable logic devices (PALs).

The PLAN software package permits a designer to create, with the use of an architecture specific language, an easily read and understood text file to describe a circuit design, select the appropriate device to accommodate the described logic, assign a pin-list to the device, document the design, generate test vectors for the selected devices, and format data to facilitate device programming and functional testing.

PLAN allows the user to assign the device type (type of PAL), generate a pin list to accommodate the equations, and enter the boolean equations that define the PALs operation. The boolean equations are written in the sum-of-products form with architecture defining commands as required.

The sum-of-products formatted boolean equations begin with a symbol representing a device output followed by either the equality ("=") or clock (":=") operator. The sign of the output symbol defines the state of the device output when the equation is satisfied. If the output symbol is preceded by a complement sign ("/") the equation output will be low when the input states defining one of the product terms (of that particular output) are true.

The sum-of-products boolean equations define an output to be true, given that one of the product terms in that output's equations is true. The symbols that make up the sum-of-products refer to the state of the input (or output). If a complement sign precedes the input (or output) symbol, in the equation, it means that the input (or output) must be low to be true (logic one). Notice that it does not matter whether the input (or output) ***pin*** has a complement sign preceding it or not, anytime a complement sign precedes an input (or output) ***symbol in an equation*** that ***symbol*** must be low to be true (logic one).

Consider the following example (use the PAL pinout and equations listed below to identify the input and output pin names);

EXAMPLE EQUATIONS:

```
IF (VCC) /DC = /AS*/CS*/DB*/CLK
             + /AS*/CS*/DC*CLK
```

This example reads: the output "/DC" will transition low given that one of the following conditions are valid;

1. the input "/AS" is low AND the input "/CS" is low AND the output "/DB" is low and the input "CLK" is low, OR
2. the input "/AS" is low AND the input "/CS" is low AND the output "/DC" is low and the input "CLK" is high

PLAN equations can be converted to PALASM™ format very easily. First, the ***outputs*** of the boolean equations should be complemented. Second, any ***symbols*** in the boolean equations that are complemented (have "/" preceding the symbol) in the ***pin*** list should be complemented in the boolean equations. As an example the above mentioned sample equation has been converted to PALASM below;

```
IF (VCC) DC = AS*CS*DB*/CLK
            + AS*CS*DC*CLK
```

Notice that "CLK" does not have a complement sign preceding it in the ***pin*** list and therefore has the same representation in both the PLAN and the PALASM equations.

EXAMPLE NATIONAL SEMICONDUCTOR PLAN FORMAT PAL EQUATIONS

```
PAL16R4D
BCLK /CS /AS NC1 /DTACK /EXST /ADDW CLK NC2
GND
/OE /STERM /DC NC3 /DB /DA NC4 /ENCAS /AREQ
VCC

IF (VCC) /AREQ = /AS*/CS*CLK
               +/AREQ*/CS*/CLK

IF (VCC) /ENCAS = /AREQ*/CS*DC
                +/AREQ*/CS*/CLK

IF (VCC) /DC = /AS*/CS*/DB*/CLK
             +/AS*/CS*/DC*CLK

IF (VCC) /STERM = /AS*/CS*/DA*DB*/CLK*/ADDW
                +/AS*/CS*/DTACK*DB*CLK*ADDW
                +/EXST*/CLK
                +/STERM*CLK

/DA := /AREQ*/CS*/DTACK*DB*/ADDW

/DB := /AREQ*/CS*/DTACK*/DA*DB*/ADDW
     +/AREQ*/CS*/DTACK*DB*/ADDW
```

Interfacing the DP8420A/21A/22A to the NS32008/016/C016/032/132

National Semiconductor
Application Note 542
Joe Tate and Rusty Meier

INTRODUCTION

This application note explains interfacing the DP8420A/21A/22A to the National Semiconductor 32C016. Two different designs are shown and explained. It is assumed the reader is familiar with the NS32C016 access cycles and the DP8420A/21A/22A modes of operation. This application note is written for the NS32C016, but is also valid for the NS32008/016/032/132.

DESIGN DESCRIPTION

This design is a simple circuit to interface the DP8420A/21A/22A to the NS32C016 and up to 32 Mbytes of DRAM. The DP8420A/21A/22A is operated in mode 0. An access cycle begins when the 32C016 asserts the $\overline{ADS}$ signal and places a valid address on the bus. The $\overline{ADS}$ signal places a group of 74AS373 fall-through latches in fall-through mode and $\overline{ADS}$ negated latches the address to guarantee the address is valid throughout the entire access. The $\overline{ADS}$ signal is inverted to produce the signal ALE to the DP8420A/21A/22A. On the next rising clock edge, after the ALE signal is asserted, the DP8420A/21A/22A will assert $\overline{RAS}$. After guaranteeing the row address hold time, tRAH, the DP8420A/21A/22A will place the column address on the DRAM address bus, guarantee the column address set-up time and assert $\overline{CAS}$. The transceivers are enabled by $\overline{CS}$ and $\overline{AS}$. After tCAC, the DRAM will place the data on the bus. The DP8420A/21A/22A will also take care of refresh access arbitration and will hold off the access by asserting the $\overline{CWAIT}$ signal to the NS32C201 TCU.

Timing parameters are referenced to the numbers shown in the DP8420A/21A/22A data sheet. Times beginning with a "$" refer to the DP8420A/21A/22A data sheet. Times beginning with a "#" refer to the NS32C016 data sheet. Times beginning with a "!" refer to the NS32C201 data sheet in the 1986 Series 32000® data book. Equations given allow the user the calculation timing based on his frequency and application. The clock to the DELCLK has been chosen to be a multiple of 2MHz. If you do not have a clock, which is a multiple of 2 MHz, the $\overline{ADS}$ to $\overline{CAS}$ time must be recalculated.

DESIGN TIMING PARAMETERS

Clock Period = Tcp10 = 100 ns @ 10 MHz
= Tcp15 = 66 ns @ 15 MHz

$300: $\overline{CS}$ asserted to CLK High
= T1 − (PHI1 to address + AS373 in to Out + AS138 Decoder + CTTL to PHI1 Max + Inverter)
= Tcp − #tALV − tphl − tphl − !tPCr
= 100 ns − 50 ns − 6 ns − 9 ns − 2 ns

= 33 ns @ 10 MHz

= 66 ns − 35 ns − 6 ns − 9 ns − 2 ns

= 14 ns @ 15 MHz

$301b: ALE Setup to CLK High
= T1 − Inverter Max − PHI1 to $\overline{ADS}$ − CTTL to PHI1
= Tcp − tplh − #tADSa − !tPCr
= 100 ns − 5 ns − 35 ns − 2 ns

= 55 ns @ 10 MHz

= 66 ns − 5 ns − 26 ns − 2 ns

= 33 ns @ 15 MHz

$302: ALE Pulse Width
= T1 − Inverter Max − PHI1 to $\overline{ADS}$ − CTTL to PHI1
= #tADSw

= 30 ns @ 10 MHz

= 25 ns @ 15 MHz

$303 & $304: Address Setup to CLK
= T1 − PHI1 to Address + AS373 in to out + CTTL to PHI1 Max)
= Tcp − #tADSa − tphl − !tPCr
= 100 ns − 50 ns − 6 ns − 2 ns

= 42 ns @ 10 MHz

= 66 ns − 35 ns − 6 ns − 2 ns

= 23 ns @ 15 MHz

$309: ALE Negated Held from CLK High
= Min CLK to $\overline{ADS}$ + Min Inverter − CTTL to PHI1 Max
= Min CLK to $\overline{ADS}$ + 1 ns − 2 ns

= Min CLK to $\overline{ADS}$ − 1 ns @ 10 MHz

= Min CLK to $\overline{ADS}$ − 1 ns @ 15 MHz

* no time is specified for CLK to $\overline{ADS}$ min.*

$310: $\overline{WIN}$ Setup to CLK High to Guarantee $\overline{CAS}$ is Delayed
= T1 + T2 − PHI1 to CTTL R.E. − DDIN Signal Valid − 74AS04
= 2Tcp − !tPCr − #tDDINv − tphl
= 200 ns − 2 ns − 45 ns − 5 ns

= 148 @ 10 MHz

= 66 ns + 66 ns − 2 ns − 38 ns − 5 ns

= 87 ns @ 15 MHz

$315: $\overline{AREA}$ Negated to CLK High that Starts Access $\overline{RAS}$

= T4 + T1 − PHI1 to CTTL R.E. − PHI1 to TSO High

= 2Tcp − !tPCr − !tTr

= 200 ns − 2 ns − 18 ns

= 180 ns @ 10 MHz

= 66 ns + 66 ns − 2 ns − 10 ns

= 120 ns @ 15 MHz

$\overline{CWAIT}$ TIMING

!tCWs(W): $\overline{CWAIT}$ Setup for WAIT STATES

= Min PHI1 Pulse Width + Min Clock Overlap − PHI1 to TSO − 74AS32

= !Tclw(1) + !tnOVL − !tTf − tphl

= 40 ns + 0 ns − 12 ns − 5 ns

= 23 ns @ 10 MHz

= 27 ns + 0 ns − 6 ns − 5 ns

= 16 ns @ 15 MHz

*parameters $311 & 312 & 314 ensure $\overline{WAIT}$ will already be asserted

!tCWs(W): $\overline{CWAIT}$ Setup for Termination of Access

= Clock Period − Max to PHI to CTTL − CLK to Wait High − 74AS32

= !TCP − !tTCr − $17 − tphl

= 100 ns − 6 ns − 31 ns − 5 ns

= 58 ns @ 10 MHz

= 66 ns − 2 ns − 31 ns − 5 ns

= 28 ns

tRAC AND tCAC TIMING FOR DRAMs

Timing diagrams are supplied on page 8. Since systems and DRAM times vary, the user is encouraged to change the following equations to match his system requirements. Timing has been supplied for systems with 0 or 1 wait states. If DELCLK is not a multiple of 2 MHz the CLK to $\overline{CAS}$ delay must be recalculated.

0 Wait States

tRAC = T2 + T3 − Max Clock Skew − CLK to $\overline{RAS}$ − Transceiver Delay − Data Setup

= 2TCP − !tPCr − $307 − tphl − #tDIs

= 200 ns − 6 ns − 26 ns − 7 ns − 15 ns

= 146 ns @ 10 MHz

= 2Tcp − !tPCr − $307 − tphl − #tDIs

= 132 ns − 2 ns − 26 ns − 7 ns − 10 ns

= 87 ns @ 15 MHz

1 Wait State

tRAC = T2 + TW + T3 − Max Clock Skew − CLK to RAS − Transceiver Delay − Data Setup

= 3Tcp − !tPCr − $307 − tphl − #tDIs

= 300 ns − 6 ns − 26 ns − 7 ns − 15 ns

= 246 ns @ 10 MHz

= 198 ns − 2 ns − 26 ns − 7 ns − 10 ns

= 153 ns @ 15 MHz

0 Wait States

tCAC = T2 + T3 − Max Clock Skew − CLK to RAS − Transceiver Delay − Data Setup

= 2Tcp − !tPCr − $308a − tphl − #tDIs

= 200 ns − 6 ns − 79 ns − 7 ns − 15 ns

= 93 ns @ 10 MHz

= 132 ns − 2 ns − 79 ns − 7 ns − 10 ns

= 34 ns @ 15 MHz

1 Wait States

tCAC = T2 + T3 − Max Clock Skew − CLK to $\overline{CAS}$ − Transceiver Delay − Data Setup

= 3Tcp − !tPCr − $308a − tphl − #tDIs

= 300 ns − 6 ns − 79 ns − 7 ns − 15 ns

= 193 ns @ 10 MHz

= 198 ns − 2 ns − 79 ns − 7 ns − 10 ns

= 100 ns @ 15 MHz

$\overline{RAS}$ Precharge Parameters

$29b: $\overline{AREQ}$ Negated Setup to CLK

= Clock Period − Clock Skew PHI1 to CTTL − PHI1 to TSO

= TCP − !tPCr − !tTr

= 100 ns − 2 ns − 18 ns

= 80 ns @ 10 MHz

= 66 ns − 2 ns − 10 ns

= 54 ns @ 15 MHz

tRP = Programmed Clocks − Clock Skew PHI1 to CTTL − PHI1 to TSO − [($\overline{AREQ}$ to $\overline{RAS}$ Negated) − (CLK to $\overline{RAS}$ Asserted)]

= 2TCP − !tPCr − !tTr − $50

= 200 ns − 2 ns − 18 ns − 16 ns

= 164 ns @ 10 MHz

RAS Low During Refresh

tRAS = Programmed Clocks − [(CLK High to Refresh $\overline{RAS}$ Asserted) − (CLK High to Refresh $\overline{RAS}$ Negated)]

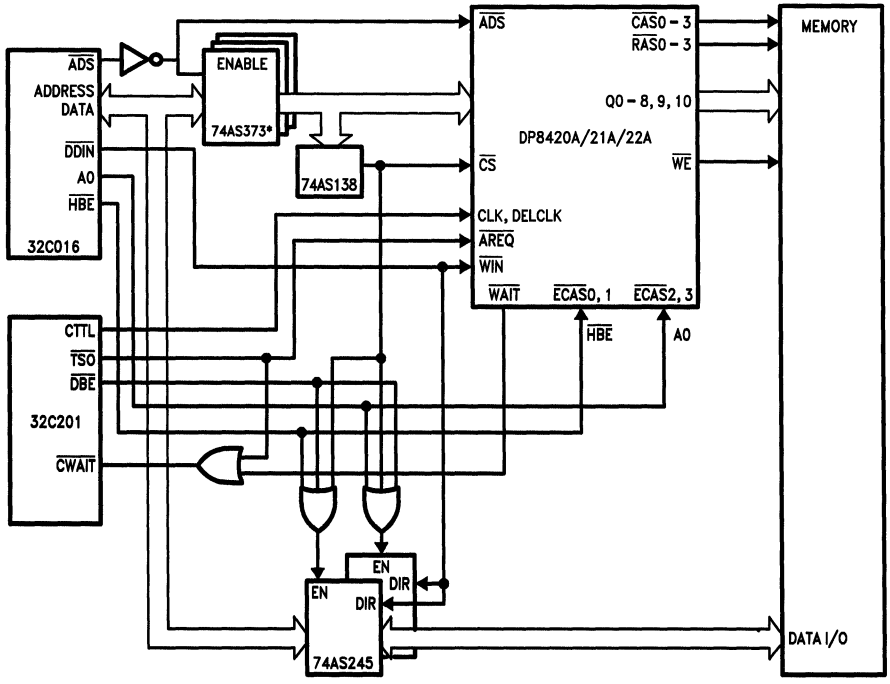
= 2TCP − $55

= 200 ns − 6 ns

= 194 ns @ 10 MHz

= 132 ns − 6 ns

= 126 ns @ 15 MHz



*LATCHES ARE NOT NEEDED IF DP8420A/21A/22A INTERNAL LATCHES ARE USED.

TL/F/9736–1

*Latches are not needed if DP8420A/21A/22A internal latches are used.

32C016 up to 15 MHz

Design Programming Bits

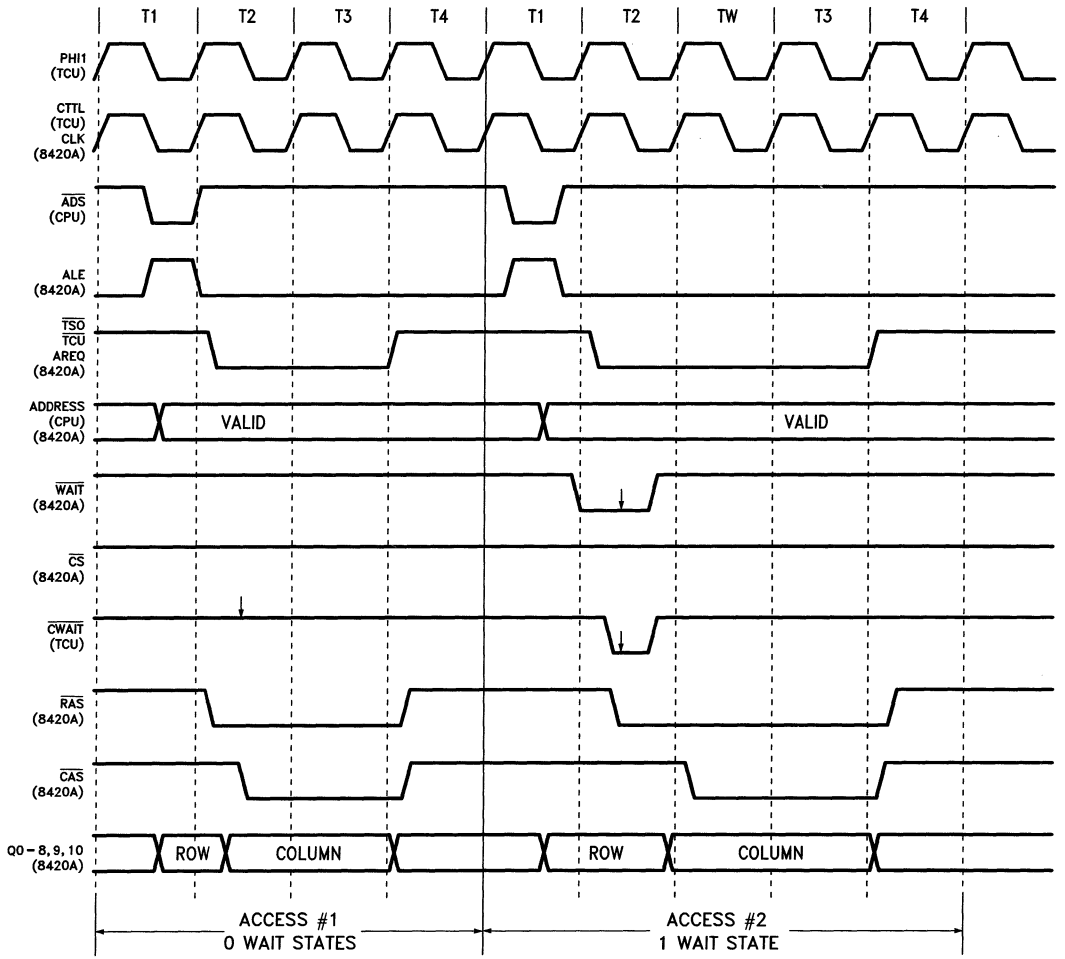
Bits	Description	Value
R0, R1	$\overline{RAS}$ Low During Refresh = 2T $\overline{RAS}$ Precharge Time = 2T	R0 = 0 R1 = 1
R2, R3	$\overline{WAIT}$ Generation Mode during Non-Burst Access	R2 = u R3 = u
R4, R5	$\overline{WAIT}$ During Burst	R4 = 0 R5 = 0
R6	ADD Wait States with $\overline{WAITIN}$	R6 = x
R7	$\overline{WAIT}$ Mode Selected	R7 = 0
R8	Non-Interleaved Mode	R8 = 1
R9	Staggered or all $\overline{RAS}$ Refresh	R9 = u
C0, C1, C2	Divisior for DELCLK *Use a Multiple of 2 MHz External Clock	C0 = * C1 = * C2 = *
C3	+30 REFRESH	C3 = *
C4, C5, C6	$\overline{RAS}$, $\overline{CAS}$ Configuration Mode **Choose an all $\overline{CAS}$ Mode, Tie a $\overline{CAS}$ to Each Nibble	C4 = ** C5 = ** C6 = **
C7	Select 0 ns Column Address Setup	C7 = 1
C8	Select 15 ns Row Address Hold	C8 = 1
C9	$\overline{CAS}$ is Delayed During Writes	C9 = 1
B0	Latches are Fall-Through	B0 = 1
B1	Access Mode 0	B1 = 0
$\overline{ECAS0}$	Non-Extend $\overline{CAS}$ Mode	$\overline{ECAS0}$ = 0

x = don't care

u = user defined

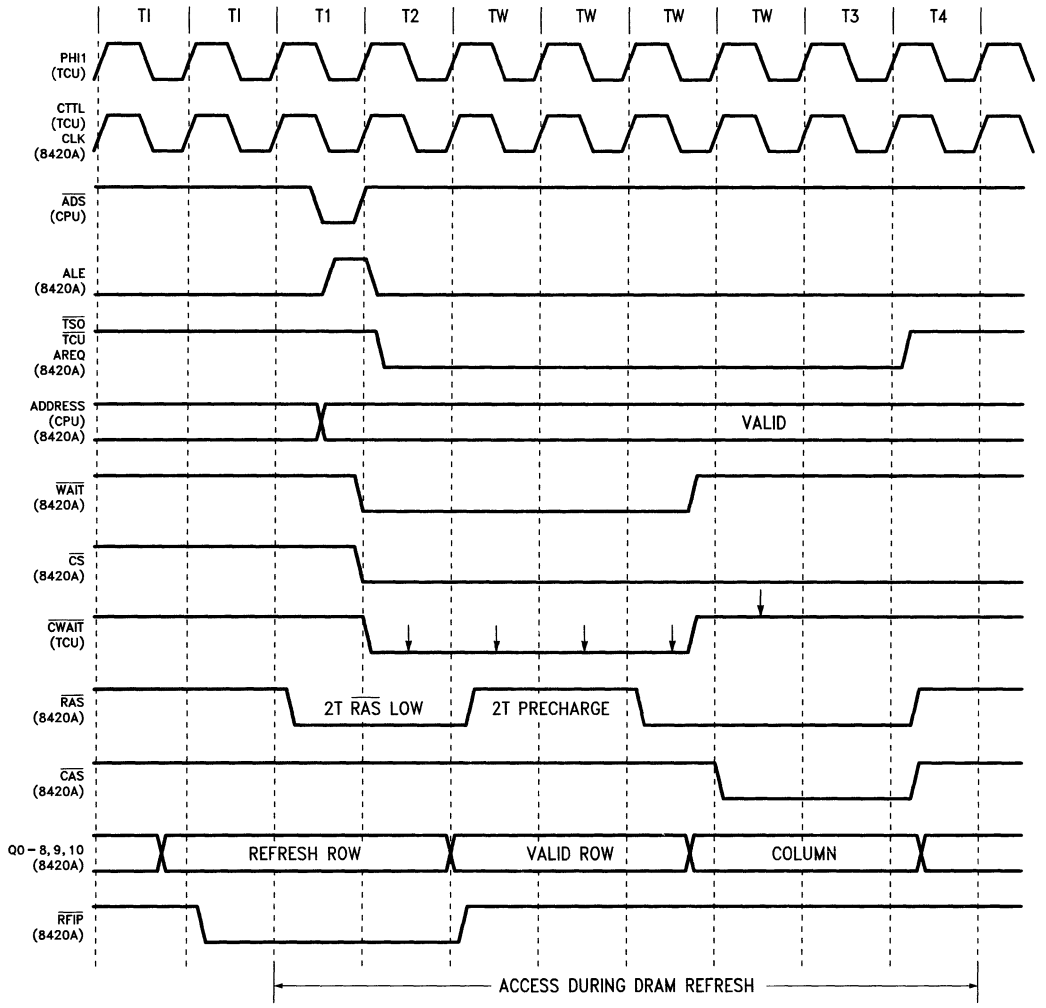
R2 = 0	R3 = 1	for 0 WAIT STATES
R2 = 1	R3 = 0	for 1 WAIT STATE
R9 = 0		all $\overline{RAS}$ refresh
R9 = 1		staggered refresh

Design Timing #1



TL/F/9736–2

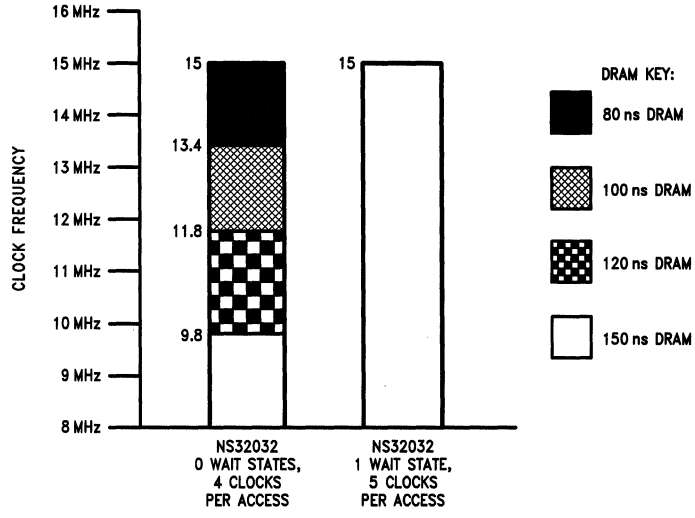
Design Timing #2



TL/F/9736-3

Access #3 Access/Refresh Arbitration

DRAM Speed Versus Processor Speed (DRAM Speed References the RAS Access Time, t_{RAC}, of the DRAM Using DP8422A-25 Timing Specifications)



TL/F/9736–4

Interfacing the DP8420A/21A/22A to the National Semiconductor NS32332

National Semiconductor
Application Note 543
Webster (Rusty) Meier Jr. and
Joe Tate

I INTRODUCTION

This application note describes how to interface the National Semiconductor NS32332 microprocessor to the DP8422A DRAM controller (also applicable to DP8420A/21A). There are four designs shown in this application note. The differences between these designs are as follows:

1. Design #1 can be used up to 14 MHz, has no wait states in normal accesses and no wait states in burst accesses, does not contain an MMU unit, is programmed with $\overline{\text{DTACK}}$0 out of the DP8422A, and has the $\overline{\text{1W}}$ input of the PAL tied high,
2. Design #2 can be used up to 15 MHz, has one wait state in normal accesses and no wait states in burst accesses, does not contain an MMU unit, is programmed with $\overline{\text{DTACK}}$1 out of the DP8422A, and has the $\overline{\text{1W}}$ input of the PAL tied high,
3. Design #3 can be used up to 14 MHz, has one wait state in normal accesses and no wait states in burst accesses, does contain an MMU unit, is programmed with $\overline{\text{DTACK}}$0 out of the DP8422A, and has the $\overline{\text{1W}}$ input of the PAL tied high,
4. Design #4 can be used up to 15 MHz, has two wait states in normal accesses and no wait states in burst accesses, does contain an MMU unit, is programmed with $\overline{\text{DTACK}}$1 out of the DP8422A, and has the $\overline{\text{1W}}$ input of the PAL tied high,

An extra wait state can also be added to any of the four above designs by tying the $\overline{\text{1W}}$ input low. It is assumed that the reader is already familiar with NS32332 and the DP8422A modes of operation.

II DESCRIPTION OF FOUR DESIGNS, ALLOWING UP TO 15 MHz OPERATION WITH 0, 1, OR 2 WAIT STATES IN NORMAL ACCESSES, NO WAIT STATES IN BURST ACCESSES AND AN OPTIONAL MMU (MEMORY MANAGEMENT UNIT, NS32382)

These four designs are all similar. Taken together they allow the user to design a 32332 DRAM system with 0, 1, or 2 wait states during an access. This system can be designed with or without the NS32382 MMU. These designs are shown driving two banks of DRAM, each bank being 32 bits in width, giving a maximum memory capacity of 32 Mbytes (using 4 Mbit x 1 DRAMs). By choosing a different $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ configuration mode (see programming mode bits section of DP8422A data sheet), this application could support 4 banks of DRAM, giving a memory capacity of 64 Mbytes (using 4 Mbit x 1 DRAMs).

Note: When driving 64 Mbytes, the timing calculations will have to be adjusted to the greater capacitive load.

The memory banks are interleaved on every four word (32-bit word) boundary. This means that the address bit (A4) is tied to the bank select input of the DP8422A (B1). If the majority of accesses made by the NS32332 are sequential, the NS32332 can be doing burst accesses most of the time. Each burst of four words can alternate memory banks, allowing one memory bank to be precharging ($\overline{\text{RAS}}$ precharge) while the other bank is being accessed. This is a higher performance memory system than a non-interleaved memory system (bank select on the higher address bits). Each back to back memory access to the same memory bank will generally require extra wait states to be inserted into the CPU access cycles to guarantee the $\overline{\text{RAS}}$ precharge time.

The logic shown in this application note forms a complete NS32332 memory sub-system, no other logic is needed. This sub-system automatically takes care of:

A. arbitration between Port A, Port B, and refreshing the DRAM;
B. the isertion of wait states to the processor (Port A and Port B) when needed (i.e., if $\overline{\text{RAS}}$ precharge is needed, refresh is happening during a memory access, the other Port is currently doing an access . . . etc.);
C. performing byte writes and reads to the 32-bit words in memory.

By making use of the enable input on the 74AS373 latch, this application can easily be used in a dual access application. The addresses and chip select are TRI-STATE® through this latch, the write input ($\overline{\text{WIN}}$), lock input ($\overline{\text{LOCK}}$), and $\overline{\text{ECAS}}$0–3 inputs must also be able to be TRI-STATE (a 74AS244 could be used for this purpose). By multiplexing the above inputs (through the use of the above parts and similar parts for Port B), the DP8422A can be used in a dual access applications. All the timing (see TIMING section of this application note) will remain the same whether single or dual accessing is implemented.

If an MMU (NS32382) is used the signal "$\overline{\text{PAV}}$" should be input to the PAL "$\overline{\text{ADS}}$" input instead of the NS32332 $\overline{\text{ADS}}$ input. If wanted the user could input the $\overline{\text{MADS}}$ signal to the PAL (using the "NC1" input), allowing the access cycle to be started one clock earlier. When the PAL senses the $\overline{\text{MADS}}$ input transitioning low it can insert one less wait state into that particular access.

The PAL output term $\overline{\text{D1}}$ and the input term $\overline{\text{1W}}$ can be deleted from the PAL if the user is not interested in adding an extra wait state to any of the four designs.

III NS32332 DESIGN, UP TO 15 MHz WITH 0, 1, OR 2 WAIT STATES IN NORMAL ACCESSES AND NO WAIT STATES IN BURST ACCESSES, PROGRAMMING MODE BITS

Programming Bits	Description
R0 = 0	$\overline{RAS}$ Low Two Clocks, $\overline{RAS}$
R1 = 1	Precharge of Two Clocks
R2 = X	Choose depending on whether design
R3 = X	1–4 is chosen. Choose R2,3 = 0,0 if $\overline{DTACK0}$ is wanted. Choose R2,3 = 1,0 if $\overline{DTACK1}$ is wanted ($\overline{DTACK}$ low first rising CLK edge after access $\overline{RAS}$ is low).
R4 = 0	No WAIT states during burst accesses
R5 = 0	
R6 = 0	If $\overline{WAITIN}$ = 0, add one clock to $\overline{DTACK}$. Since we are not using the $\overline{WAITIN}$ input it should be tied high on the DP8422A.
R7 = 1	Select $\overline{DTACK}$
R8 = 1	Non-interleaved mode
R9 = X	
C0 = X	Select based upon the input
C1 = X	"DELCLK" frequency. Example: if the
C2 = X	input clock frequency is 14 MHz then
C3 = X	choose C0,1,2, = 1,1,0 (divide by seven, this will give a frequency of 2 MHz).
C4 = 0	$\overline{RAS}$ groups selected by "B1". This
C5 = 0	mode allows two $\overline{RAS}$ outputs to go
C6 = 1	low during an access, and allows byte writing 32-bit words.
C7 = 1	Column address setup time of 0 ns
C8 = 1	Row address hold time of 15 ns
C9 = 1	Delay $\overline{CAS}$ during write accesses to one clock after $\overline{RAS}$ transitions low
B0 = 1	Fall-thru latches
B1 = 0	Access mode 0
$\overline{ECAS0}$ = 0	$\overline{CAS}$ not extended beyond $\overline{RAS}$

0 = program with low voltage level

1 = program with high voltage level

X = program with either high or low voltage level (don't care condition)

IV NS32332, DESIGN #2 (NO MMU UNIT) AT 15 MHz WITH ONE WAIT STATE IN NORMAL ACCESSES, DESIGN #4 (HAS MMU UNIT) AT 15 MHz WITH TWO WAIT STATES IN NORMAL ACCESSES. DESIGNS #2 AND #4 HAVE A TOTAL OF THREE CLOCK PERIODS TO ACCESS THE DRAM IN NORMAL ACCESSES AND HAVE ZERO WAIT STATES DURING BURST ACCESSES.

1. Maximum time to latch enable valid:

   $\overline{ADS}$ makes the 74AS373 fall-thru at 17 ns (max) from PHI1 CLOCK low + 5 ns (74AS04) + 2 ns (PHI1 to CTTL clock skew) = 24 ns

   ***Note:** $\overline{MADS}$ and $\overline{PAV}$ are valid 17 ns maximum from PHI1 rising clock edge if NS32382 is used

2. Maximum time to address valid from CTTL CLOCK (NS32332 spec) = 20 ns (address valid from PHI1 clock) + 2 ns (PHI1 to CTTL clock skew) = 22 ns

3. Maximum time to latched address valid from CTTL CLOCK:

   11.5 ns (74AS373 enable time maximum) + 24 ns (#1) = 35.5 ns

4. Minimum ALE high setup time to $\overline{CLOCK}$ high (DP8422A-25 needs 15 ns):

   66.6 ns (one clock period) − 17 ns (NS32332 max time to $\overline{ADS}$ low from PHI1) − 2 ns (PHI1 to CTTL clock skew) − 15 ns (PAL16R6B combinational output) − 5 ns (74AS04) = 27.6 ns

5. Minimum address setup time to CLOCK high (DP8422A-25 needs 18 ns):

   66.6 ns (one clock period) − 35.5 ns (#3) = 31.1 ns

6. Minimum $\overline{CS}$ setup time to CLOCK high (DP8422A-25 needs 13 ns):

   66.6 ns (one clock period) − 35.5 ns (#3) − 9 ns (Max 74AS138 decoder) = 22.1 ns

7. Determining t_{RAC} ($\overline{RAS}$ access time required by the DRAM):

   199.8 ns (three clock periods to do access) − 2 ns (NS32C201 PHI1 to CTTL clock skew) − 7 ns (data set-up time) − 7 ns (74F245) − 26 ns (CLK to $\overline{RAS}$ low on DP8422A-25) = 157.8 ns. Therefore the t_{RAC} of the DRAM must be 157.8 ns or less.

8. Determining t_{CAC} ($\overline{CAS}$ access time) and column address access time required by the DRAM:

   199.8 ns − 2 ns − 7 ns − 7 ns − 12 ns (74AS32, 6 ns, plus 6 ns extra, taken from lab data on the 74AS32, requried to drive a 22Ω damping resistor and an equivalent load capacitance of 150 pF, approximately 16 DRAM $\overline{CAS}$ inputs per $\overline{CASG}$n output) − 72 ns (CLK to $\overline{CAS}$ low on DP8422A-25) = 99.8 ns. Therefore the t_{CAC} of the DRAM must be 99.8 ns or less.

9. Determining the nibble mode access time required by the DRAM:

   66.6 ns (T3 clock) − 2 ns (clock skew) − 12 ns (PAL16R6B, $\overline{CASEN}$ clocked output) − 12 ns (74AS32, see #8 description) − 7 ns (74F245) − 7 ns (data setup) = 26.6 ns
   Therefore the nibble mode access time needed by the DRAM must be 26.6 ns or less.

10. Maximum time to $\overline{DTACK0}$ low (PAL16R6B needs 15 ns setup to CTTL):

    66.6 ns (one clock) − 33 ns ($\overline{DTACK0}$ low from CLK high on DP8422A-25) = 33.6 ns
    **Note that $\overline{DTACK1}$ may be used from some of the designs, it occurs at 28 ns maximum from the CLK input.

11. Minimum RDY setup time to RDY being sampled (12 ns to the PHI1 falling edge is needed by the NS32332):

    27.3 ns (minimum PHI1 high pulse width) − 2 ns (NS32C201 PHI1 to CTTL clock skew) − 12 ns (PAL16R6B clocked output maximum) = 13.3 ns

***Note:** Calculations can be performed for different frequencies and the other designs (#1 and #3) by substituting the appropriate values into the above equations. Design numbers 1 and 3 have only two clock periods to perform an access, therefore the t_{RAC} and t_{CAC} calculations would be affected by having one less clock period during an access.

V NS32332 DESIGN, PAL EQUATIONS WRITTEN IN NATIONAL SEMICONDUCTOR PLAN FORMAT

```
PAL16R6B
CTTL /CS /ADS /BOUT /DTACK EXRDY /1W RESET
NC1 GND
/OE /EN_TRAN NC2 /AREQ /CASEN /D2 /D1 RDY
/ADSL VCC
IF (VCC) /ADSL = /ADS
                 +/ADSL*AREQ*/RESET
                 +/ADSL*/BOUT*/CS
                 +/ADSL*/CASEN*/CS
IF (VCC) /EN=_TRAN = /AREQ*/CS*/CASEN
                     +/EN_TRAN*/BOUT
RDY := /CS*/ADSL*D1*/1W
       +/CS*/ADSL*DTACK*D2*1W
/D1 := /DTACK*D1*/1W*/CS*/RESET
       +*/EXRDY*CS*/1W
/D2 := /CS*/D1*/1W*/RESET
       +/CS*/DTACK*D2*1W*/RESET
       +CS*/EXRDY*1W*/RESET
/CASEN := /ADSL*D2*/RESET
          +AREQ*/RESET
/AREQ := /ADSL*/RESET
```

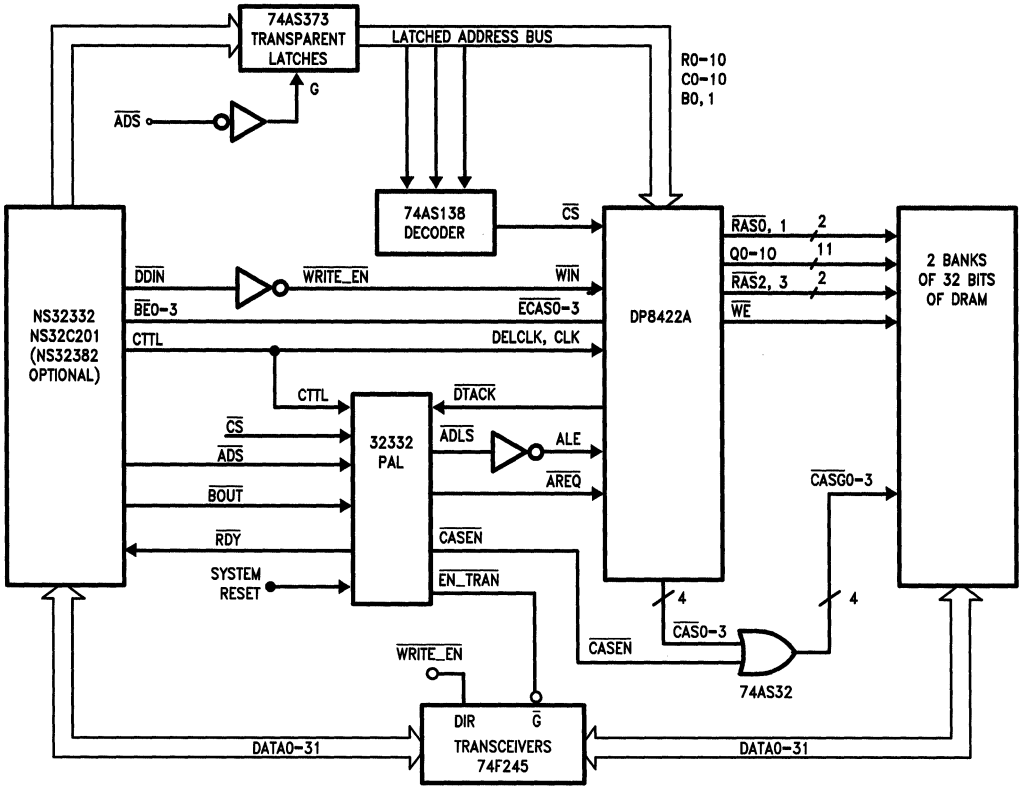
Key: Reading PAL® Equations Written in PLAN™

```
EXAMPLE EQUATIONS: /CASEN := /ADSL*D2*/RESET
                             +AREQ*/RESET
```

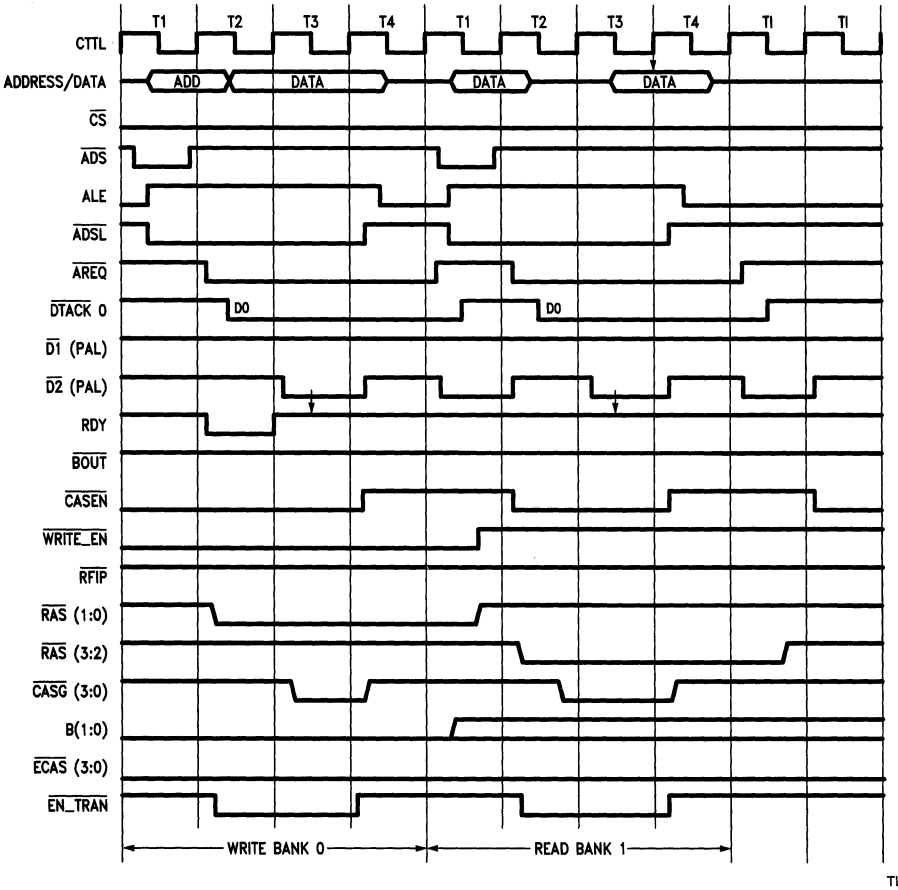
This example reads: the output "/CASEN" will transition low on the next rising "CTTL" clock edge (given that one of the following conditions are valid a setup time before "CTTL" transitions high);

1. the output "/ADSL" is low AND the output "/D2" is high AND the input "RESET" is low, OR
2. the output "/AREQ" is high AND the input "RESET" is low

NS32332 Design, up to 15 MHz, May Include MMU; 0, 1, or 2 Wait States per Normal Access, No Wait States in Burst Accesses (74AS373 latches are not needed if address and bank inputs meet setup and hold times of ALE and DP8422A is programmed to latch the addresses)

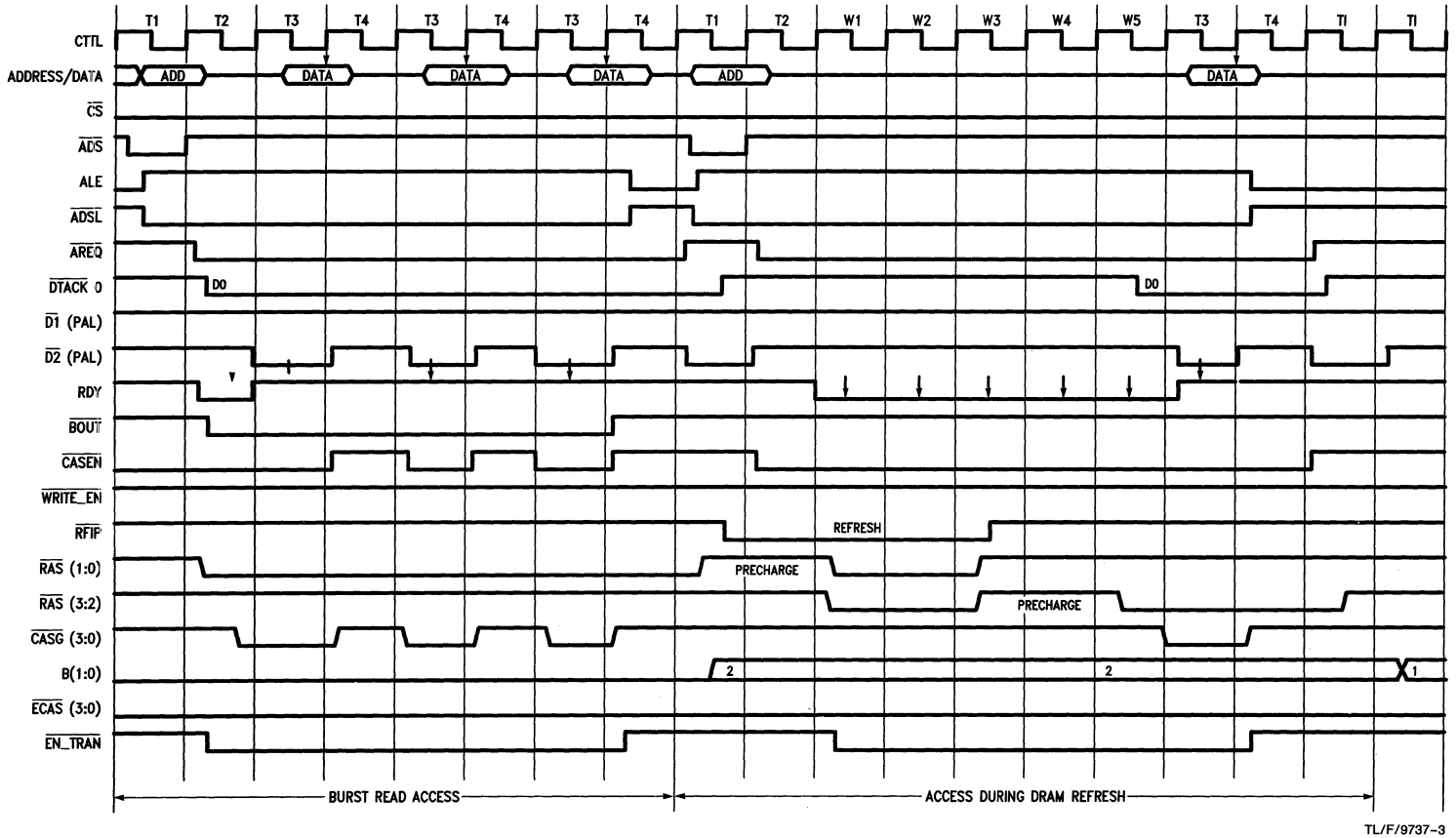


TL/F/9737–1



TL/F/9737–2

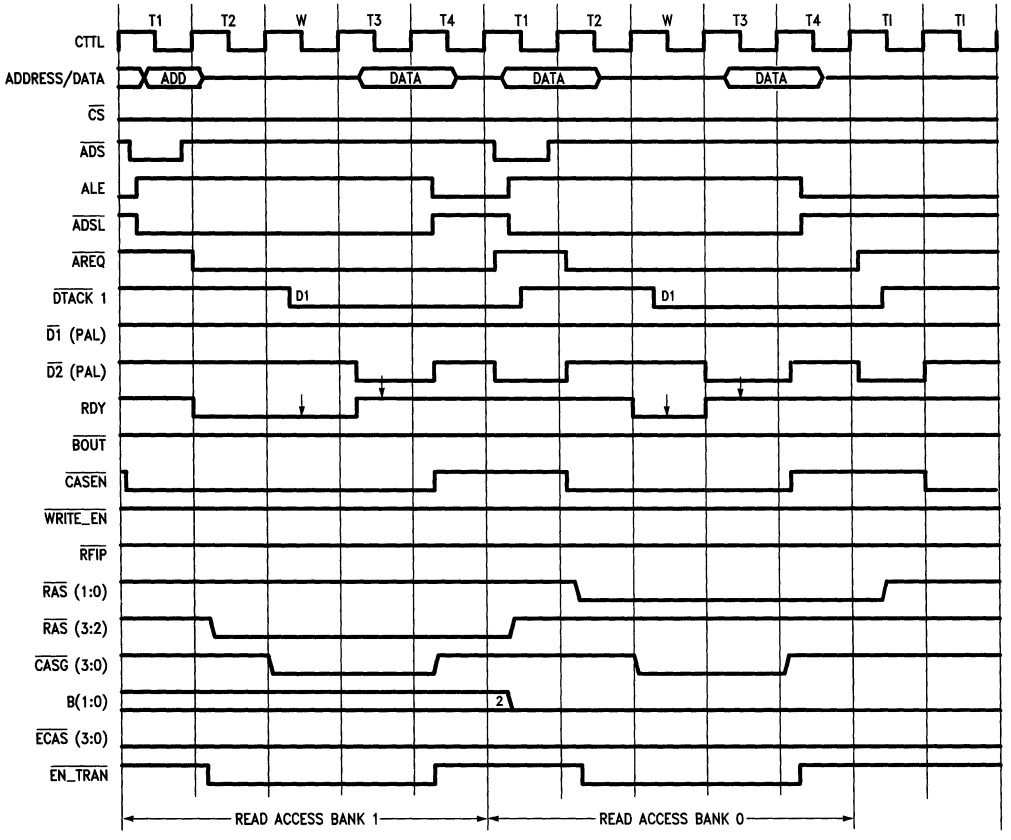
Design # 1: 32332 with No Wait States (No MMU, PAL Input $\overline{1W}$ Tied High)



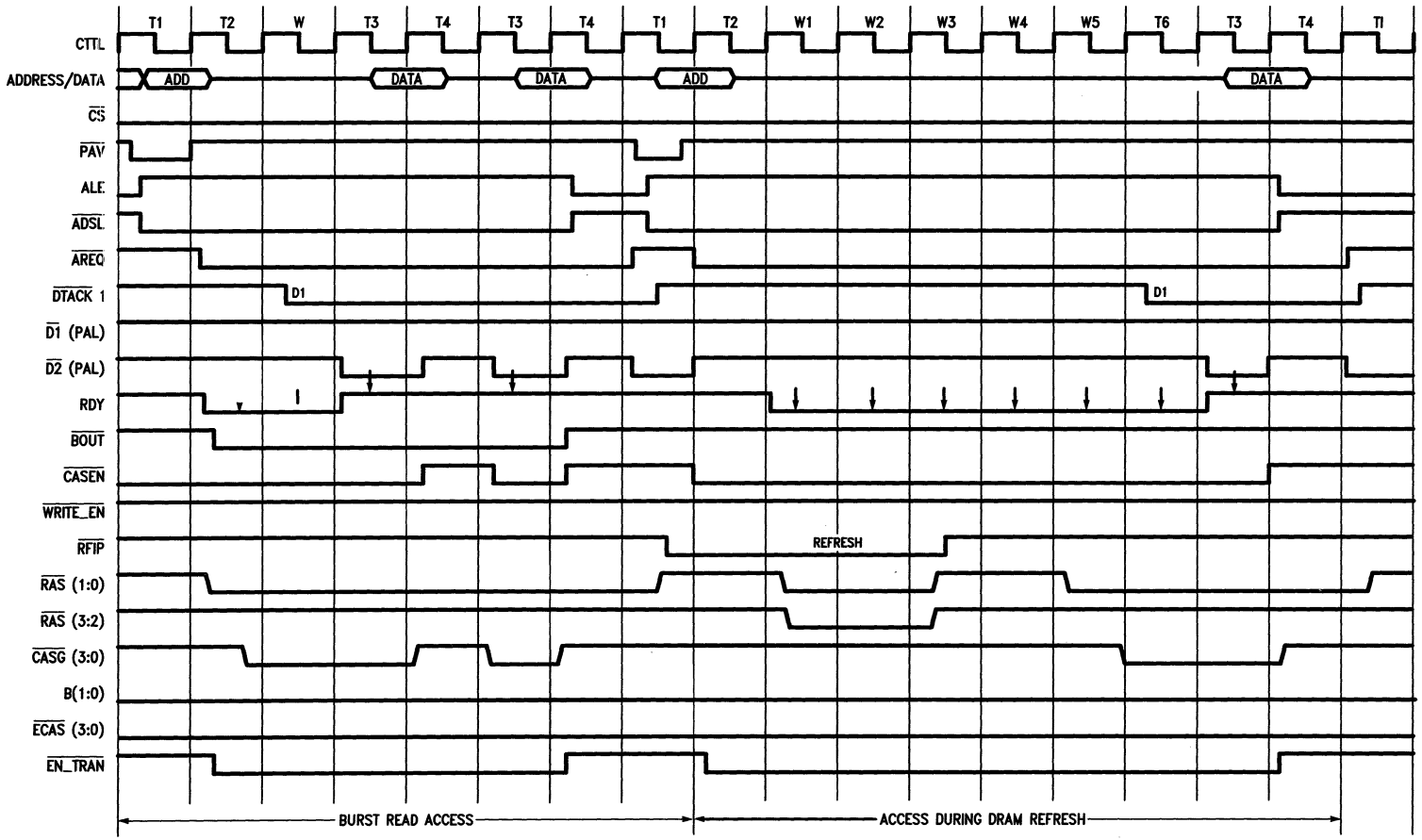
TL/F/9737–3

Design # 1: 32332 with No Wait States (No MMU, PAL Input $\overline{1W}$ Tied High)

Design #2: 32332 with One Wait State per Access (Non-Burst, $\overline{\text{DTACK}}$1 Programmed)
No MMU, PAL Input $\overline{\text{1W}}$ Tied High

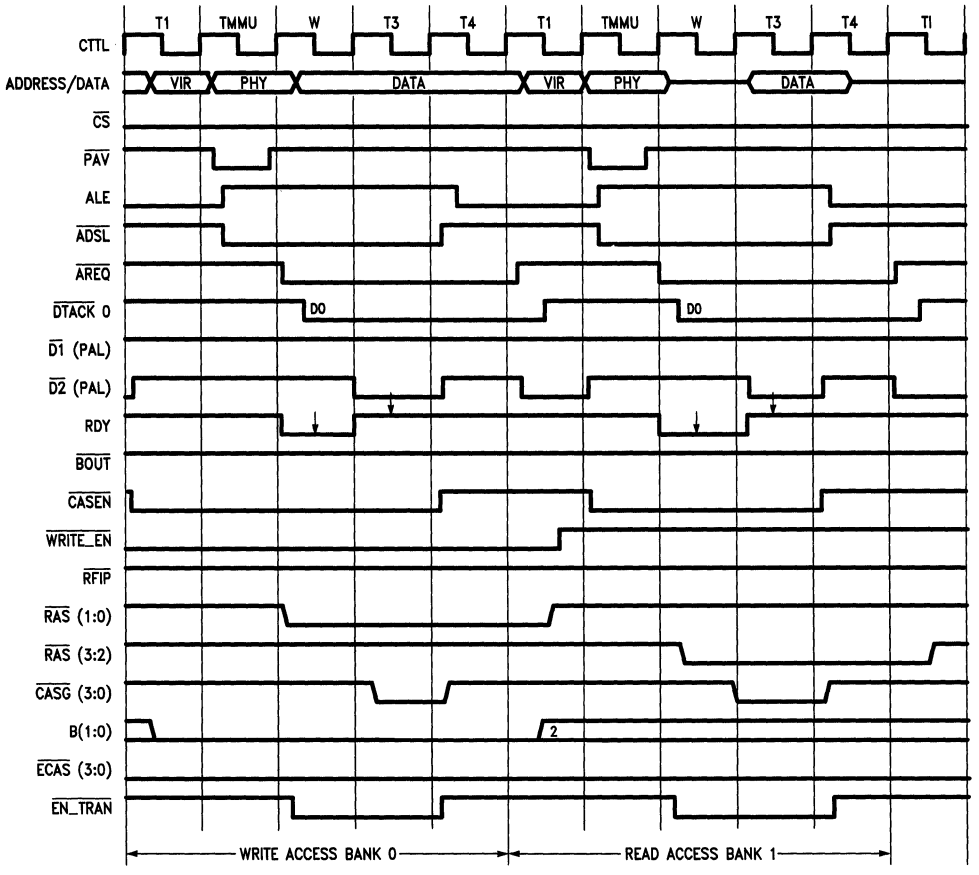


TL/F/9737–4



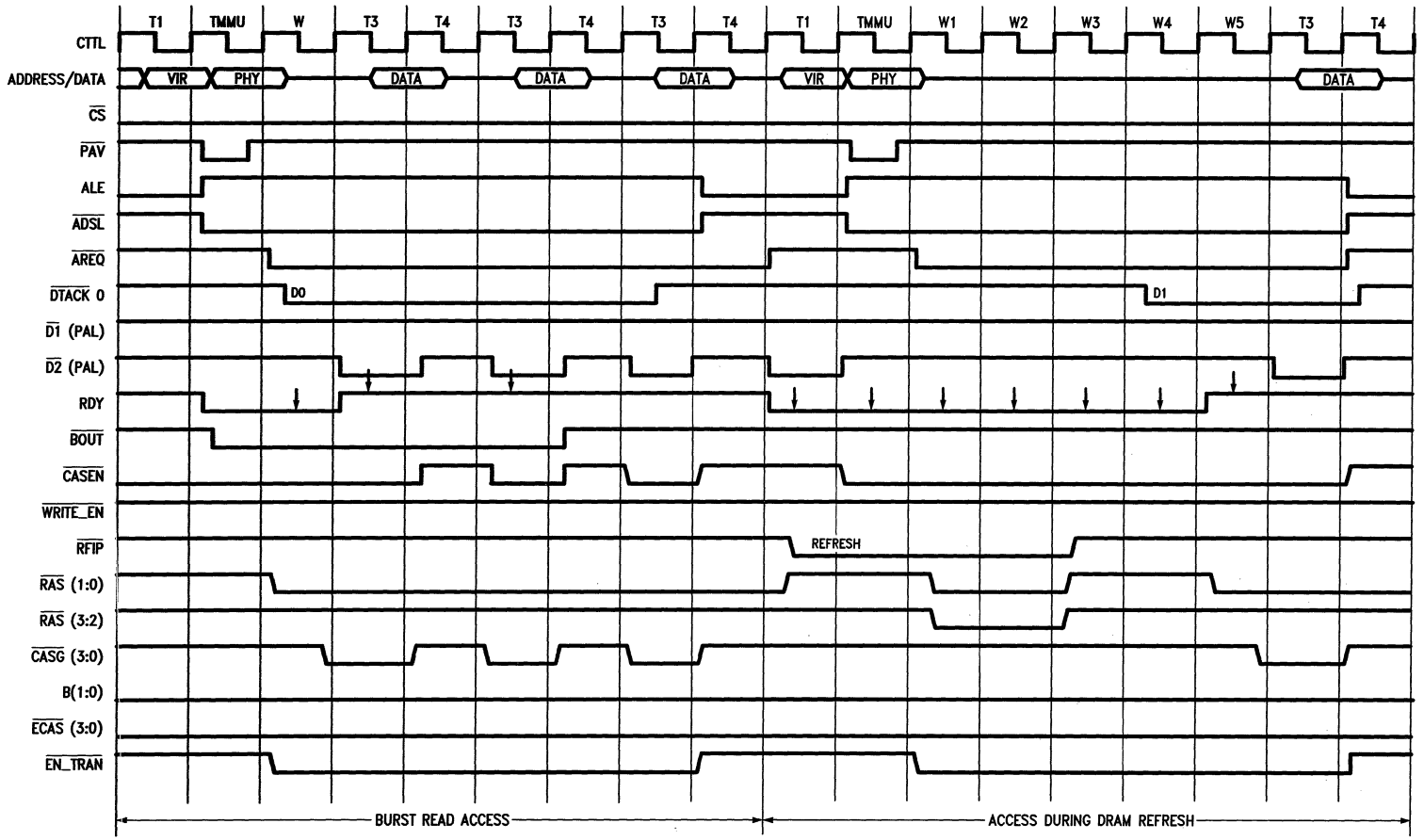
TL/F/9737–5

Design #2: 32332 with One Wait State per Access (Non-Burst, DTACK1 Programmed), No MMU, PAL Input 1W Tied High



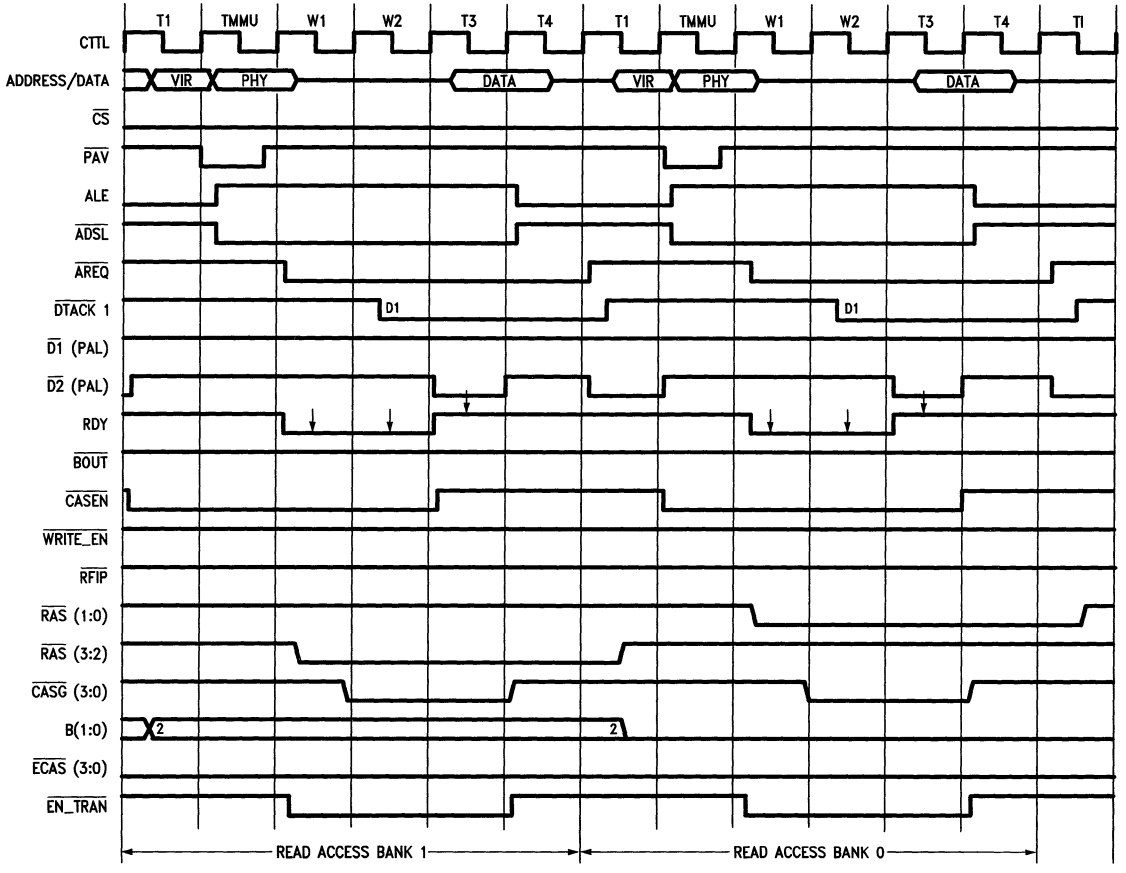
TL/F/9737–6

Design #3: 32332 with MMU and One Wait State per Access (Non-Burst), PAL Input $\overline{1W}$ Tied High



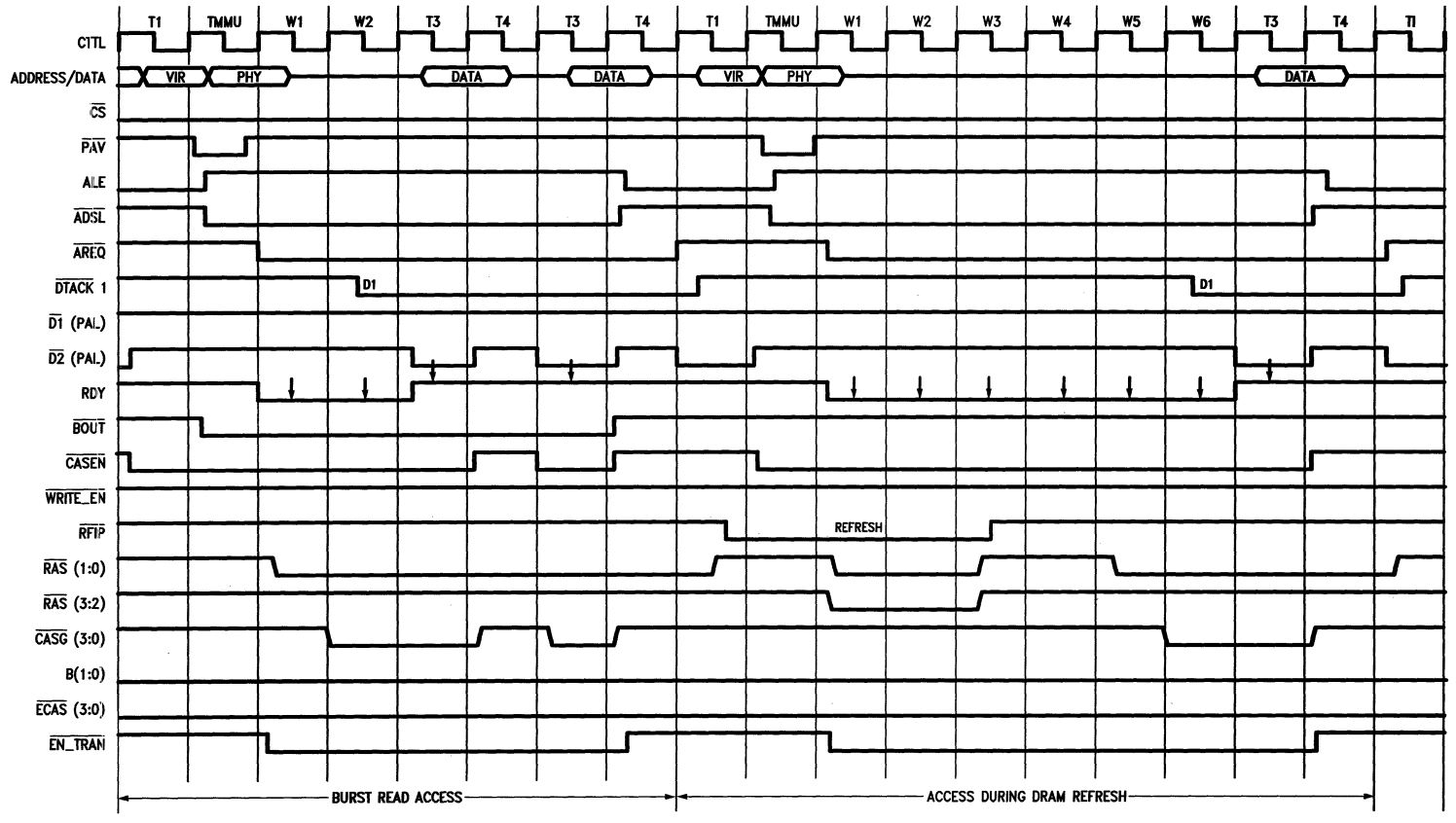
TL/F/9737–7

Design #3: 32332 with MMU and One Wait State per Access (Non-Burst), PAL Input $\overline{1W}$ Tied High



TL/F/9737–8

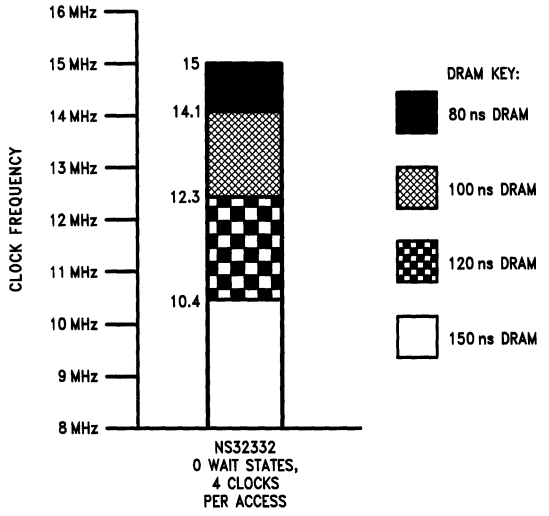
Design #4: 32332 with MMU and Two Wait States per Access (Non-Burst, $\overline{\text{DTACK}}$1 Programmed), PAL Input $\overline{\text{1W}}$ Tied High



TL/F/9737–9

Design #4: 32332 with MMU and Two Wait States per Access (Non-Burst, $\overline{\text{DTACK}}$1 Programmed), PAL Input $\overline{\text{1W}}$ Tied High

Dram Speed Vs. Processor Speed, (Dram Speed References the RAS Access Time, tRAC, of the Dram, using DP8422A-25 Timing Specifications)



TL/F/9737–10

Interfacing the DP8420A/21A/22A to the National Semiconductor NS32532

National Semiconductor
Application Note 541
Webster (Rusty) Meier Jr. and Joe Tate

1.0 INTRODUCTION

This application note describes how to interface the National Semiconductor NS32532 microprocessor to the DP8422A DRAM controller (also applicable to DP8420A/21A). It is assumed that the reader is already familiar with NS32532 and the DP8422A modes of operation.

2.0 DESCRIPTION OF DESIGN, ALLOWING UP TO 25 MHz OPERATION WITH 2 OR 3 WAIT STATES IN NORMAL ACCESSES AND 1 WAIT STATE DURING BURST ACCESSES

This design drives two banks of DRAM, each bank being 32 bits in width, giving a maximum memory capacity of 32 Mbytes (using 4 Mbit x 1 DRAMs). By choosing a different $\overline{RAS}$ and $\overline{CAS}$ configuration mode (see programming mode bits section of DP8422A data sheet) this application could support 4 banks of DRAM, giving a memory capacity of 64 Mbytes (using 4 Mbit x 1 DRAMs, NOTE that when driving 64 Mbytes the timing calculations will have to be adjusted to the greater capacitive load).

The memory banks are interleaved on every four word (32-bit word) boundary. This means that the address bit (A4) is tied to the bank select input of the DP8422A (B1). If the majority of accesses made by the NS32532 are sequential, the NS32532 can be doing burst accesses most of the time. Each burst of four words can alternate memory banks, allowing one memory bank to be precharging ($\overline{RAS}$ precharge) while the other bank is being accessed. This is a higher performance memory system than a non-interleaved memory system (bank select on the higher address bits). Each separate memory access to the same memory bank will generally require extra wait states to be inserted into the CPU access cycles to allow for the $\overline{RAS}$ precharge time.

This design supports the NS32532 burst access operations. To support these operations it is assumed that nibble mode DRAMs will be used. (See the timing calculations, Section IV).

The logic shown in this application note forms a complete NS32532 memory sub-system, no other logic is needed. This sub-system automatically takes care of:

A arbitration between Port A, Port B, and refreshing the DRAM;

B the insertion of wait states to the processor (Port A and Port B) when needed (i.e., if $\overline{RAS}$ precharge is needed, refresh is happening during a memory access, the other Port is currently doing an access . . . etc);

C performing byte writes and reads to the 32-bit words in memory.

The Confirm Bus Cycle ($\overline{CONF}$) signal is input to the DP8422A Chip Select ($\overline{CS}$) input. Therefore the $\overline{CONF}$ signal disables the current access, from the DP8422A, if the NS32532 has cancelled it ($\overline{CONF}$ high setup to $\overline{ADS}$ transitioning low). The PAL starts an access via the DP8422A by pulling the $\overline{ADS}$, $\overline{AREQ}$ inputs low. These inputs are brought low given that $\overline{CS}$ and Begin Memory Transaction ($\overline{BMT}$) are both low.

By making use of the enable input on the 74AS244 buffer, this application can easily be used in a dual access application. The addresses and chip select are TRI-STATE® through this buffer, the write input ($\overline{WIN}$), lock input ($\overline{LOCK}$), and $\overline{ECAS}0-3$ inputs must also be able to be TRI-STATE (another 74AS244 could be used for this purpose). By multiplexing the above inputs (through the use of the above parts and similar parts for Port B) the DP8422A can be used in a dual access application. If this design is used in a dual access application at 25 MHz the t_{RAC} and t_{CAC} (required $\overline{RAS}$ and $\overline{CAS}$ access time required by the DRAM) will have to be recalculated since the time to $\overline{RAS}$ and $\overline{CAS}$ is longer for the dual access application (see TIMING section of this application note).

3.0 NS32532 DESIGN, UP TO 25 MHz WITH 2 OR 3 WAIT STATES DURING NORMAL ACCESSES AND 1 WAIT STATE DURING BURST ACCESSES, PROGRAMMING MODE BITS

Programming Bits	Description
R0 = 1	$\overline{RAS}$ low four clocks, $\overline{RAS}$
R1 = 1	precharge of three clocks.
R2 = 1	$\overline{DTACK}$1 is chosen. $\overline{DTACK}$ low
R3 = 0	first rising CLK edge after access $\overline{RAS}$ is low.
R4 = 0	No Wait states during burst
R5 = 0	accesses.
R6 = 0	If $\overline{WAITIN}$ = 0, add one clock to $\overline{DTACK}$. Since we are not using the $\overline{WAITIN}$ input it should be tied high on the DP8422A.
R7 = 1	Select $\overline{DTACK}$
R8 = 1	Non-Interleaved Mode
R9 = X	
C0 = X	Select based upon the input
C1 = X	"DELCLK" frequency. Example: if
C2 = X	the input clock frequency is 20 MHz then choose C0,1,2 = 0,0,0 (divide by ten, this will give a frequency of 2 MHz). If using the DP8422A over 20 MHz do an initial divide by two externally and then run that output into the DELCLK input and choose the correct divider.
C3 = X	
C4 = 0	$\overline{RAS}$ groups selected by "B1". This
C5 = 0	mode allows two $\overline{RAS}$ outputs to
C6 = 1	go low during an access, and allows byte writing 32-bit words.
C7 = 1	Column address setup time of 0 ns.
C8 = 1	Row address hold time of 15 ns.
C9 = 1	Delay $\overline{CAS}$ during write accesses to one clock after $\overline{RAS}$ transitions low.
B0 = 1	Fall-thru latches
B1 = 1	Access Mode 1
$\overline{ECAS}$0 = 0	$\overline{CAS}$ not extended beyond $\overline{RAS}$.

0 = program with low voltage level

1 = program with high voltage level

X = program with either high or low voltage level (don't care condition)

NS32532 TIMING CALCULATIONS FOR DESIGN AT 25 MHz WITH 3 WAIT STATES DURING THE NORMAL ACCESSES AND 1 WAIT STATE DURING BURST ACCESSES

1 Minimum $\overline{ADS}$ low setup time to $\overline{CLOCK}$ high for $\overline{DTACK}$ logic to work correctly (DP8422A-25 needs 25 ns):

40 ns (one clock period) − 8 ns (PAL16R4D clocked output maximum) = 32 ns

2a Minimum time to $\overline{ADS}$ low = 40 ns (one clock period) + 2 ns (minimum clocked output delay of PAL16R4D PAL) = 42 ns minimum

2b Minimum address setup time to $\overline{ADS}$ low (DP8422A-25 needs 14 ns):

42 ns (#2a above) − 8 ns (max time to address valid from BLCK high) − 6.2 ns (74AS244 buffer delay max) = 27.8 ns

3a Minimum $\overline{CS}$ setup time to CLK high (PAL16R4D needs 10 ns):

40 ns (one clock period) − 8 ns (maximum time to address valid from BCLK high) − 6.2 ns (74AS244 buffer delay maximum) − 9 ns (max 74AS138 decoder) = 16.8 ns

3b Minimum $\overline{CS}$ setup time to $\overline{ADS}$ low (DP8422A-25 needs 5 ns):

a Minimum time to $\overline{ADS}$ low (see #2a from above) = 42 ns

b Maximum time to $\overline{CONF}$ ($\overline{CONF}$ is tied to $\overline{CS}$ of the DP8422A) = 20 ns (one half clock period) + 9 ns ($\overline{CONF}$ low from falling clock edge) = 29 ns maximum

Therefore:

42 ns (minimum time to $\overline{ADS}$ low) − 29 ns (maximum time to $\overline{CONF}$ low) = 13 ns

4 Determining t_{RAC} during a normal access ($\overline{RAS}$ access time needed by the DRAM):

160 ns (four clock periods to do the access) − 8 ns (PAL16R4D clocked output) − 29 ns ($\overline{ADS}$ to $\overline{RAS}$ low) − 10 ns (NS32532 data setup time) − 7 ns (74F245) = 106 ns

Therefore the t_{RAC} of the DRAM must be 106 ns or less.

5 Determining t_{CAC} during a normal access ($\overline{CAS}$ access time) and column address access time needed by the DRAM:

160 ns − 8 ns − 10 ns − 7 ns − 75 ns ($\overline{ADS}$ to $\overline{CAS}$ low on DP8422A-25, 50 pF spec) − 12 ns [74AS32, 6 ns, plus 6 ns extra, taken from lab data on the 74AS32, for drving a 22Ω damping resistor and 150 pF of capacitance associated with driving 16 DRAM $\overline{CAS}$ inputs (per $\overline{CAS}$ output)] = 48 ns

Therefore the t_{CAC} of the DRAM must be 48 ns or less.

6 Determining the nibble mode access time needed during a burst access:

80 ns (two clock periods to do the burst) − 20 ns (one half clock period during which $\overline{CAS}$ is high from the previous access) − 10 ns (PAL16R4D combinational output from CLK input falling edge, $\overline{ENCAS}$) − 12 ns (74AS32 delay to produce $\overline{CAS}$ from the $\overline{ENCAS}$ input, see description from #5) − 10 ns (NS32532 data setup time) − 7 ns (74F245) = 21 ns

Therefore the nibble mode access time of the DRAM must be 21 ns or less.

7 Maximum time to $\overline{DTACK}$1 low (PAL16R6D needs 10 ns setup to BCLK):

40 ns (one clock) − 28 ns ($\overline{DTACK}$1 low from CLK high on DP8422A-25) = 12 ns

8 Minimum $\overline{RDY}$ setup time to BCLK (19 ns to BCLK rising edge is needed by the NS32532):

40 ns (one clock period) − 8 ns (PAL16R4D clocked output maximum) = 32 ns

Note: Calculations can be performed for different frequencies by substituting the appropriate values into the above equations.

5.0 NS32532 DESIGN, PAL EQUATIONS WRITTEN IN NATIONAL SEMICONDUCTOR PLAN FORMAT

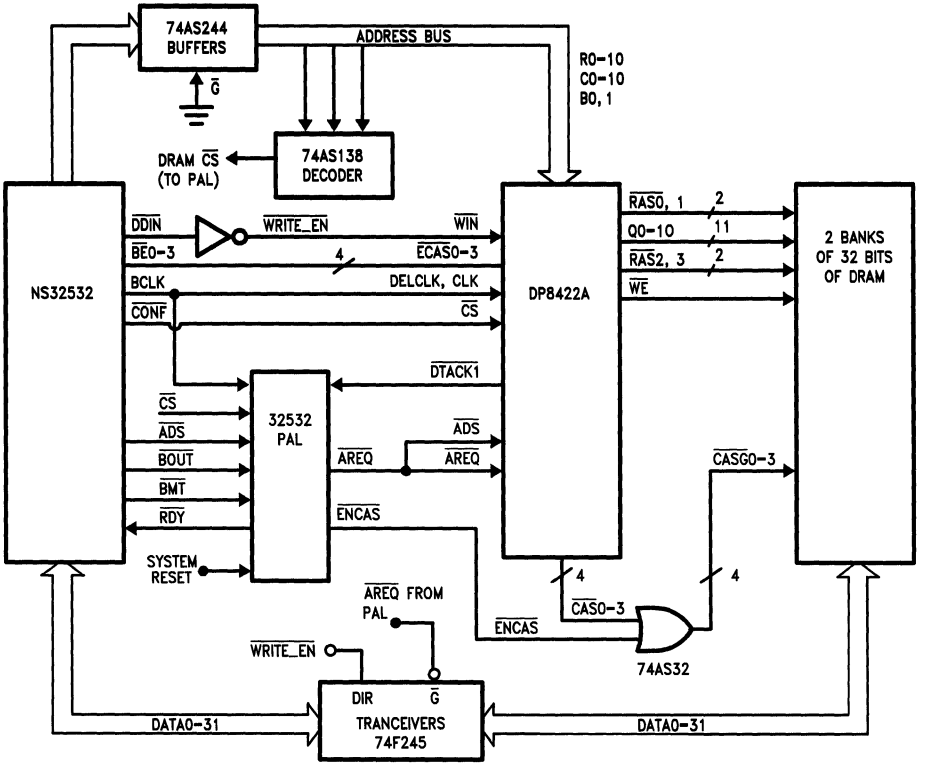
```
PAL16R4D
BCLK /CS /ADS /BOUT /DTACK /EXRDY /3W CLK /BMT GND
/OE NC1 /DB NC2 /RDY /DA /AREQ /ENCAS /ADSL VCC

IF (VCC) /ADSL = /ADS
                 +/ADSL*/CLK
                 +/ADSL*/BOUT*/CS
                 +/ADSL*RDY*/CS

IF (VCC) /ENCAS = /AREQ*/CS*DB
                  +/AREQ*/CLK*/CS

IF (VCC) /DB = /AREQ*/BOUT*/RDY*/CLK
             +/AREQ*/BOUT*/DB*CLK

/AREQ := /ADSL*/BMT*/CS
       +/ADSL*/AREQ*/CS

/DA := /ADSL*/AREQ*/DTACK*DA*/3W*/CS

/RDY := /AREQ*/DTACK*/ADSL*/DA*/CS*/3W
      +/AREQ*/DTACK*/ADSL*RDY*/CS*3W
      +/EXRDY
```

Key: Reading PAL equations written in PLAN

```
EXAMPLE EQUATIONS: IF (VCC) /DB = /AREQ*/BOUT*/RDY*/CLK
                                  +/AREQ*/BOUT*/DB*CLK
```

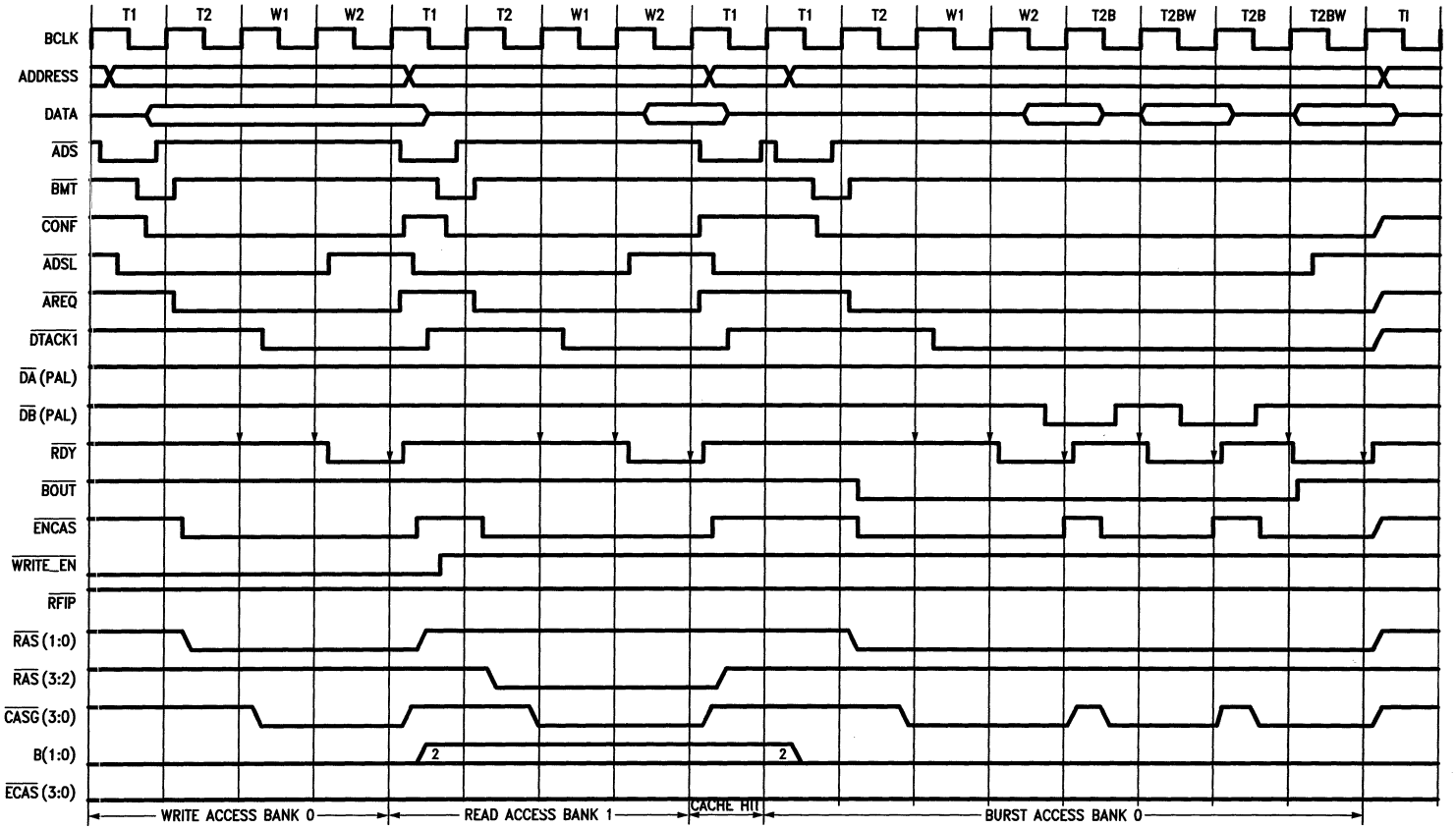
This example reads: the output "/DB" will transition low given that one of the following conditions are valid;
1) the output "/AREQ" is low AND the input "/BOUT" is low AND the output "/RDY" is low and the input "CLK" is low, OR
2) the output "/AREQ" is low AND the input "/BOUT" is low AND the output "/DB" is low and the input "CLK" is high.

NS32532 Design, up to 25 MHz, with 2 or 3 Wait States in Normal Accesses and 1 Wait State in Burst Accesses



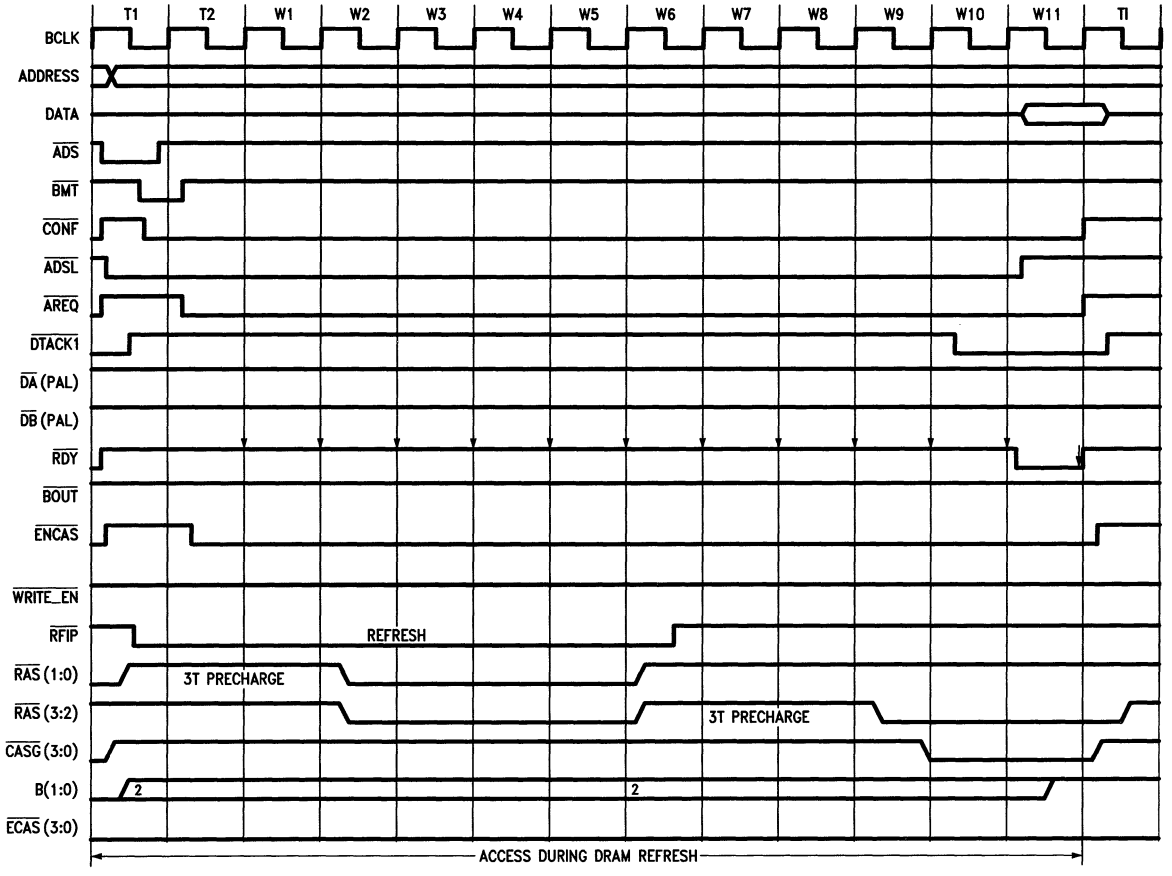
TL/F/9735-1

NS32532 with 2 Wait States per Access (One Wait State during Burst Accesses), $\overline{3W}$ Input of PAL is Tied High



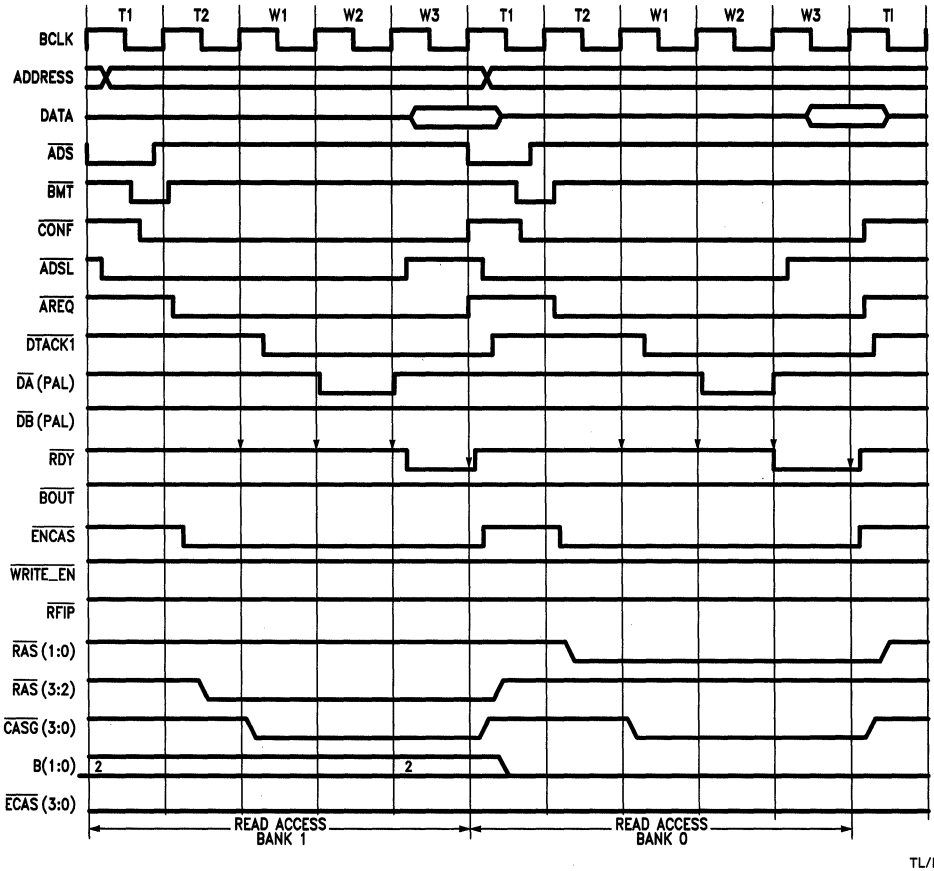
TL/F/9735–2

NS32532 with 2 Wait States per Access (One Wait State during Burst Accesses), $\overline{3W}$ Input of PAL is Tied High



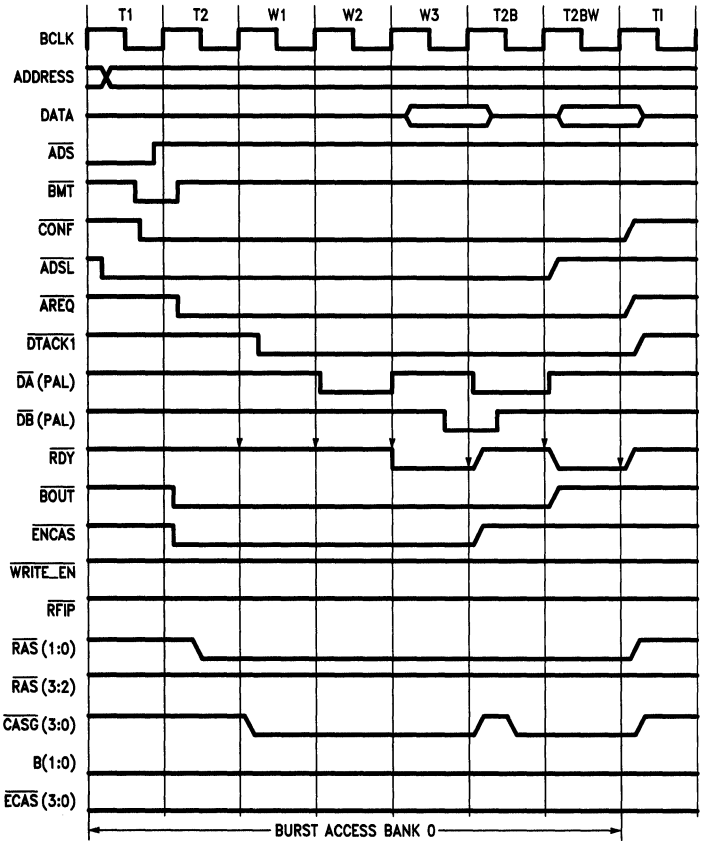
TL/F/9735–3

NS32532 with 3 Wait States per Access (One Wait State during Burst Accesses), $\overline{3W}$ Input of PAL is Tied Low



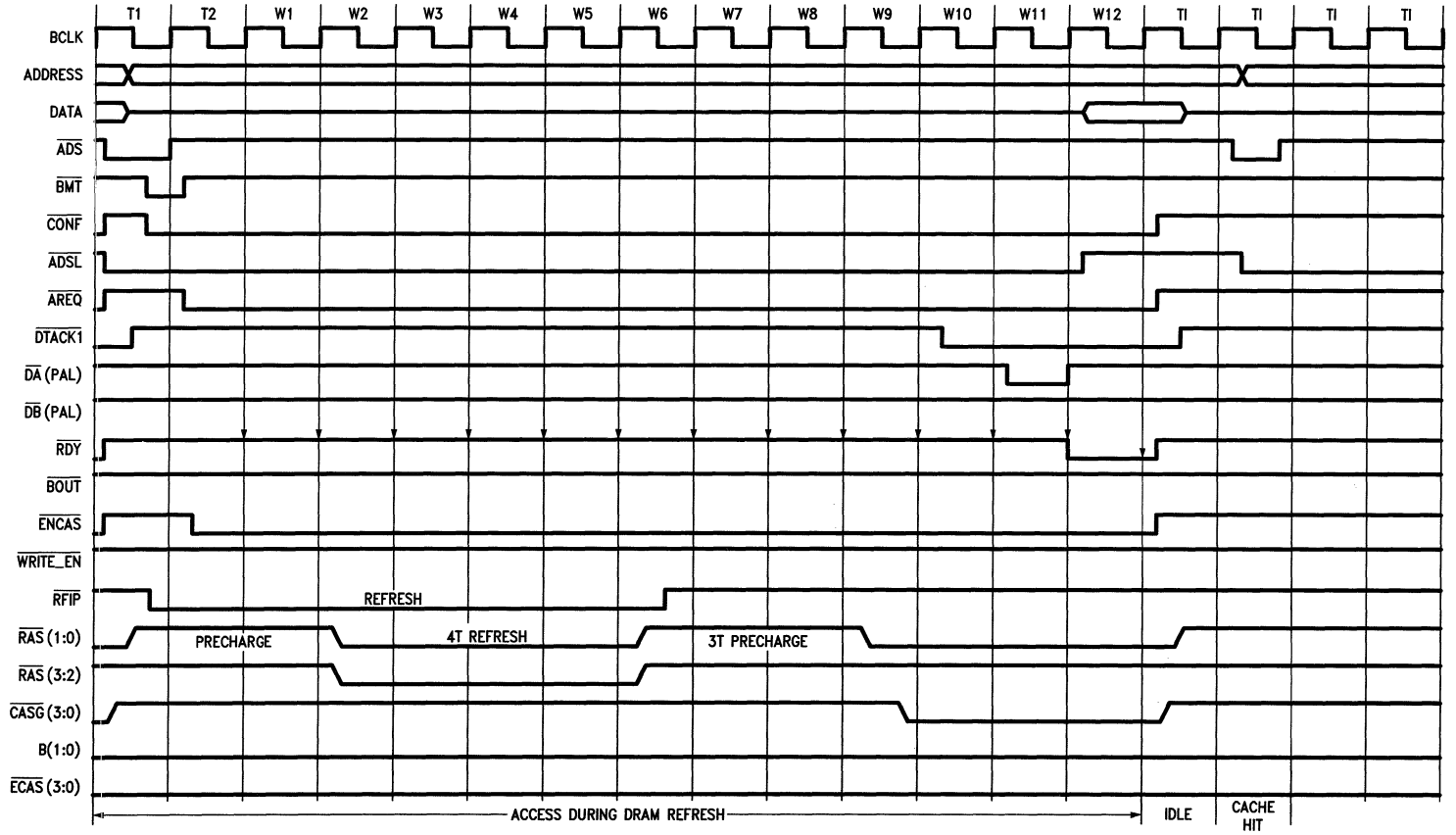
TL/F/9735–4

NS32532 with 3 Wait States per Access (One Wait State during Burst Accesses), $\overline{3W}$ Input of PAL is Tied Low



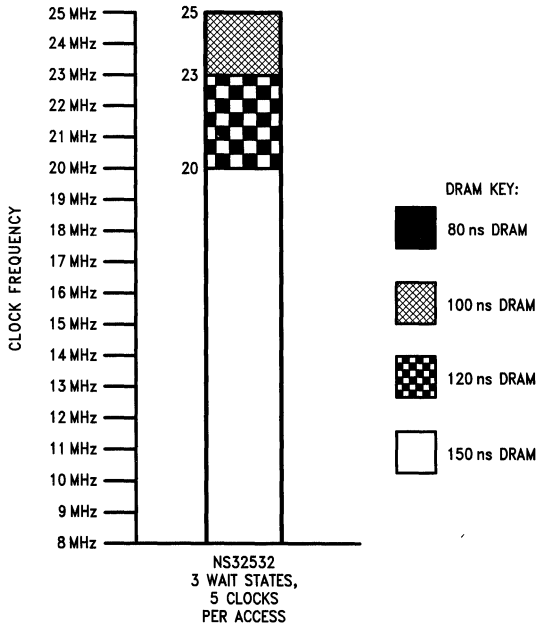
TL/F/9735–5

NS32532 with 3 Wait States per Access, $\overline{3W}$ Input of PAL is Tied Low



TL/F/9735–6

DRAM Speed Vs. Processor Speed, (DRAM Speed References the RAS Access Time, t_{RAC}, of the DRAM, using DP8422A-25 Timing Specificatons)



TL/F/9735–7

A Dual Access NS32532 Error Detecting and Correcting Memory System

National Semiconductor
Application Note 540
Webster (Rusty) Meier, Jr. and Joe Tate

I. INTRODUCTION

This appendix describes how to interface two NS32532 microprocessors, both synchronous to the same system clock, to a DP8422A DRAM controller and a 74F632 EDAC chip. It is assumed that the reader is already familiar with NS32532, the DP8422A, and the 74F632 modes of operation. The National Semiconductor DP8420A can be used in place of the 74F632, though its timing is slower.

This application note supports the following types of memory accesses:

1. Read accesses with 6 wait states inserted (8 clock periods total in the synchronous mode read access), any single bit errors are automatically corrected before sending the data to the CPU (EDAC unit in always correct mode);
2. Write accesses with 3 wait states inserted (5 clock periods total in the synchronous mode write access);
3. Byte write accesses with 7 wait states inserted (9 clock periods total in the synchronous mode byte write access);
4. Burst read accesses with 3 wait states in the burst portion of the access (4 clock periods total per synchronous mode burst read memory access);
5. Scrubbing during DRAM refreshes (6 clock periods total during the refresh if no errors, 8 clock periods total during the refresh if any errors), any single bit errors are corrected. The corrected word is then written back to the DRAM.

II. DESCRIPTION OF 25 MHz DUAL ACCESS NS32532 SYSTEM INTERFACED TO THE DP8422A AND THE 74F632

This design allows two NS32532 microprocessors to access a common error corrected dynamic memory system. The error corrected memory system is implemented using the 74F632 EDAC chip in the always correct mode. Whichever NS32532 accessed the memory last has a higher priority. Both NS32532s are interfaced to the DRAM and allow the DRAM system to support burst mode accesses.

This design is very similar to the 68030 dual access EDAC design and the PALs will be very similar. The reader can refer to that design to see the timing waveforms, block diagrams, and simulations. The only necessary changes to the 68030 design are that the $\overline{\text{STERMA,B}}$ PAL outputs will have to be modified to support the NS32532 $\overline{\text{RDY}}$ inputs, and the $\overline{\text{AREQ}}$, $\overline{\text{AREQB}}$ outputs of the PALs will have to take $\overline{\text{ADS}}$, $\overline{\text{CS}}$, and $\overline{\text{CONF}}$ into consideration (not just $\overline{\text{ADS}}$, $\overline{\text{CS}}$ as in the 68030 design). The user should also be careful not to violate the DP8422A parameter #416 ($\overline{\text{AREQ}}$ negated to $\overline{\text{ADS}}$ asserted to guarantee tASR = 0 ns) when generating the $\overline{\text{AREQ}}$, $\overline{\text{AREQB}}$ PAL outputs.

During read accesses the data is always processed through the EDAC chip (always correct type of system). If a single bit error occurs during a read access this design guarantees correct data to the CPU, but does not write the corrected data back to the DRAM. Single bit soft errors in memory are only corrected (written back to memory) during scrubbing type refreshes. The memory is scrubbed often enough that the probability of accumulating two soft errors in memory is very unlikely.

During read accesses the data is always processed through the 74F632 EDAC chip (i.e., the EDAC data buffers are enabled to provide the data to the CPU). The 74F632 is always put into latch and correct mode during read accesses, even though the data from the memory may be correct. This allows $\overline{\text{CAS}}$ to be toggled early (before the CPU has sampled the data), during burst mode accesses, to start accessing the next word of the burst access.

This design drives two banks of DRAM, each bank being 39 bits in width (32 data bits plus 7 check bits) giving a maximum memory capacity of 32 Mbytes of error corrected memory (using 4 Mbit x 1 DRAMs). By choosing a different $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ configuration mode (see programming mode bits section of DP8422A data sheet) this application can support 4 banks of DRAM, giving a memory capacity of 64 Mbytes (using 4 Mbit x 1 DRAMs, NOTE that when driving 64 Mbytes the timing calculations will have to be adjusted to the greater capacitive load).

The memory banks are interleaved on every four word (32-bit word) boundary. This means that the address bit (A4) is tied to the bank select input of the DP8422A (B1).

Address bits A3,2 are tied to the least significant column address inputs (C1,0) to support burst accesses using nibble mode DRAMs.

Address bits A1,0 are used to produce the four byte select data strobes, used in byte reads and writes. If the majority of accesses made by the NS32532 are sequential, the NS32532 can be doing burst accesses most of the time. Each burst of four words can alternate memory banks (address bit A4 tied to DP8422A pin B1), allowing one memory bank to be precharging ($\overline{\text{RAS}}$ precharge) while the other bank is being accessed. This is a higher performance memory system than a non-interleaved memory system (bank select on the higher address bits). Each separate memory access to the same memory bank will generally require extra wait states to be inserted into the CPU access cycles to allow for the $\overline{\text{RAS}}$ precharge time.

The logic shown in this application note forms a complete NS32532 memory sub-system, no other logic is needed. This sub-system automatically takes care of:

A. arbitration between Port A, Port B, and refreshing the DRAM;
B. the insertion of wait states to the processor (Port A and Port B) when needed (i.e., if $\overline{\text{RAS}}$ precharge is needed, refresh is happening during a memory access, the other Port is currently doing an access . . . etc.);
C. performing bytes write and reads to the 32-bit words in memory;
D. normal and burst access operations.

By making use of the enable input on the 74AS244 buffer, this application allows dual access applications. The addresses and chip select are TRI-STATE through this buffer, the write input ($\overline{\text{WIN}}$), lock input ($\overline{\text{LOCK}}$) and $\overline{\text{ECAS0–3}}$ inputs must also be able to be TRI-STATE (another 74AS244 could be used for this purpose). By multiplexing the above inputs (through the use of the above parts and similar parts for Port B) the DP8422A allows dual accessing to be performed.

III. NS32532 25 MHz DUAL ACCESS EDAC DESIGN: THE EDAC ERROR MONITORING METHOD IN CONJUNCTION WITH THE NS32532 BUS RETRY FEATURE

The NS32532 dual access EDAC system design can use the error monitoring method in conjunction with the NS32532 bus retry feature, instead of the always correct method (design shown in the NS32532 application note). The error monitoring method can yield a slight improvement in system performance.

By using the error monitoring method of error correction single read accesses or the first read access during a burst access can be shortened by one clock period, allowing a synchronous read access to have only 5 wait states inserted, 7 clock periods total (compared to 6 wait states, 8 clock periods total when doing the always correct method). All other types of accesses (burst reads, bytes writes, word writes, refresh scrubbing) will execute in the same number of clock cycles, and in the same manner as described in this application note.

Read accesses can save one wait state because the data from the DRAM memory is assumed to be correct in the error monitoring system design. Therefore the DRAM data is given directly to the CPU instead of running it through the EDAC chip as was done in the always correct method.

In order to do this design it is required that the bus retry feature of the NS32532 and registered transceivers be employed.

The bus retry feature of the NS32532 involves pulling the NS32532 input signal "$\overline{BRT}$" low before the end of state T2 or T2B. Given that this is done the NS32532 will complete the bus cycle normally, but will ignore the data read in the case of a read cycle. The CPU will then wait for $\overline{BRT}$ to transition high before repeating the bus cycle (unless that access is not currently needed by the CPU). This feature is useful for the case where an error is detected in the DRAM data. In this case $\overline{BRT}$ is brought low until the data from the DRAM is corrected (by the EDAC chip) and written back to the DRAM. $\overline{BRT}$ is then brought high to continue CPU processing.

Registered transceivers are necessary (in place of the 74F245's shown in the block diagram) during burst mode read accesses because $\overline{CAS}$ transitions high before the CPU has sampled the DRAM data. The registered transceivers hold the data valid until the CPU samples it during these cases.

A read, read with a single bit error, and burst read access timing are shown at the end of this appendix implementing the error monitoring method. The user can see how these access cycles differ from the always correct method access cycles shown in the 68030 dual access EDAC application note.

IV. NS32532 25 MHz DUAL ACCESS DESIGN, PROGRAMMING MODE BITS

Programming Bits	Description
R0 = 1 R1 = 1	$\overline{RAS}$ low four clocks, $\overline{RAS}$ precharge of three clocks
R2 = 1 R3 = 0	$\overline{DTACK1}$ is chosen. $\overline{DTACK}$ low first rising CLK edge after access $\overline{RAS}$ is low.
R4 = 0 R5 = 0	No WAIT states during burst accesses
R6 = 0	If $\overline{WAITIN}$ = 0, add one clock to $\overline{DTACK}$. $\overline{WAITIN}$ may be tied high or low in this application depending upon the number of wait states the user desires to insert into the access.
R7 = 1	Select $\overline{DTACK}$
R8 = 1 R9 = X	Non-interleaved Mode
C0 = 1 C1 = X C2 = X	Selected based upon the input "DELCLK" frequency. Example: if the input clock frequency is 20 MHz, then choose C0,1,2 = 0,0,0 (divide by ten, this will give a frequency of 2 MHz). If DELCLK of the DP8422A is over 20 MHz do an initial divide by two externally and then run that output into the DELCLK input and choose the correct divider.
C3 = X C4 = 0 C5 = 0 C6 = 1	$\overline{RAS}$ groups select by "B1". This mode allows two $\overline{RAS}$ outputs to go low during an access, and allows byte writing in 32-bit words.
C7 = 1	Column address setup time of 0 ns
C8 = 1	Row address hold time of 15 ns
C9 = 1	Delay $\overline{CAS}$ during write accesses to one clock after $\overline{RAS}$ transitions low
B0 = 1	Fall-thru latches
B1 = 1	Access mode 1
$\overline{ECAS}$0 = 0	Non-extend $\overline{CAS}$ mode

0 = Program with low voltage level
1 = Program with high voltage level
X = Program with either high or low voltage level (don't care condition)

V. NS32532 25 MHz WORST CASE TIMING CALCULATIONS

The worst case access is an access from Port B. This occurs because the time to $\overline{RAS}$ and $\overline{CAS}$ low is longer for the Port B access than a Port A access, a refresh with scrubbing access, or an access which has been delayed from starting (due to refresh, $\overline{RAS}$ precharge time, or the other Port accessing memory).

A. Worst case time to $\overline{RAS}$ low from the beginning of an access cycle:

   40 ns (T1 clock period of NS32532) + 10 ns (PAL16R4D maximum combinational output delay to produce $\overline{AREQB}$) + 41 ns (DP8422A-25 parameter #102, $\overline{AREQ}$ to $\overline{RAS}$ delay maximum) = 91 ns

B. Worst case time to $\overline{CAS}$ low from the beginning of an access cycle:

   40 ns + 10 ns + 94 ns (DP8422A-25 parameter #118a, $\overline{AREQB}$ to $\overline{CAS}$ delay maximum) = 144 ns

C. Worst case time to DRAM data valid:

   144 ns (from "B" above, maximum time to $\overline{CAS}$) + 50 ns ($\overline{CAS}$ access time "tCAC" for a typical 100 ns DRAM) = 194 ns

D. Worst case time to data valid on the EDAC data bus:

   194 ns (from "C" above) + 7 ns (74AS244 maximum delay) = 201 ns

E. Worst case time until the error flags are valid from the 74F632:

   201 ns (from "D" above) + 31 ns (74F632 maximum time to error flags valid) = 232 ns

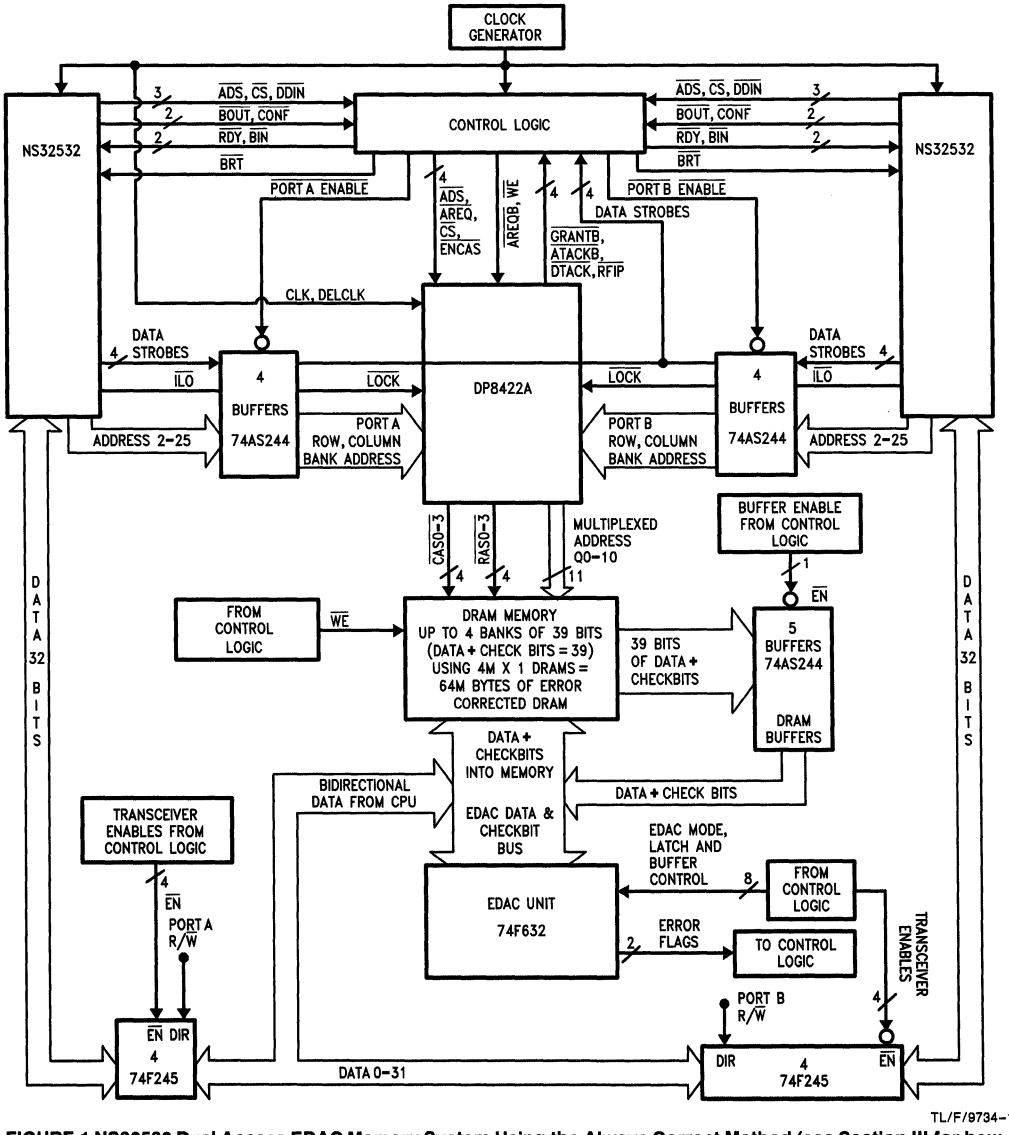
F. Worst case time until corrected data is valid from the 74F632:

   201 ns (from "D" above) + 28 ns (74F632 maximum time from data in to correct data out) = 229 ns

G. Worst case time until corrected data is available at the CPU:

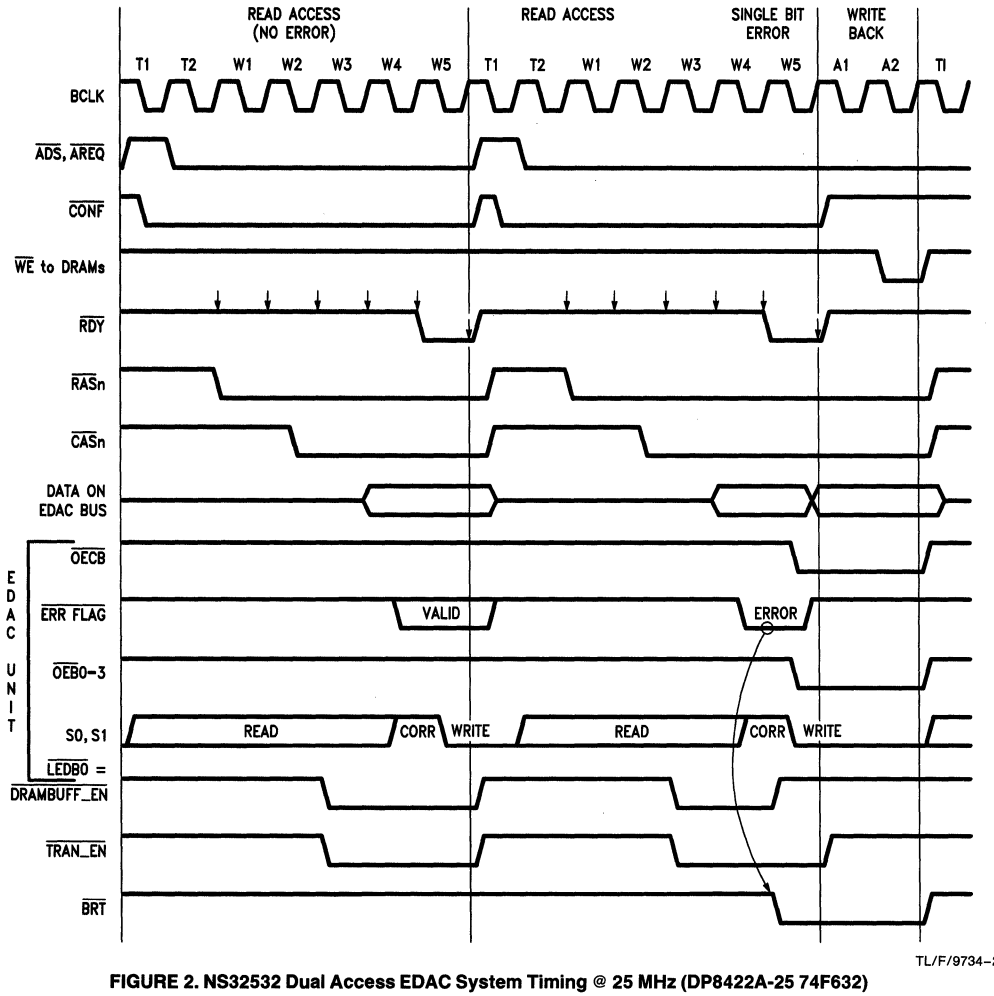
   229 ns (from "F" above) + 7 ns (74F245 maximum delay) = 236 ns

EDAC Memory—Control Logic in this System Needs the Following:
3 PALS and Some Logic Gates



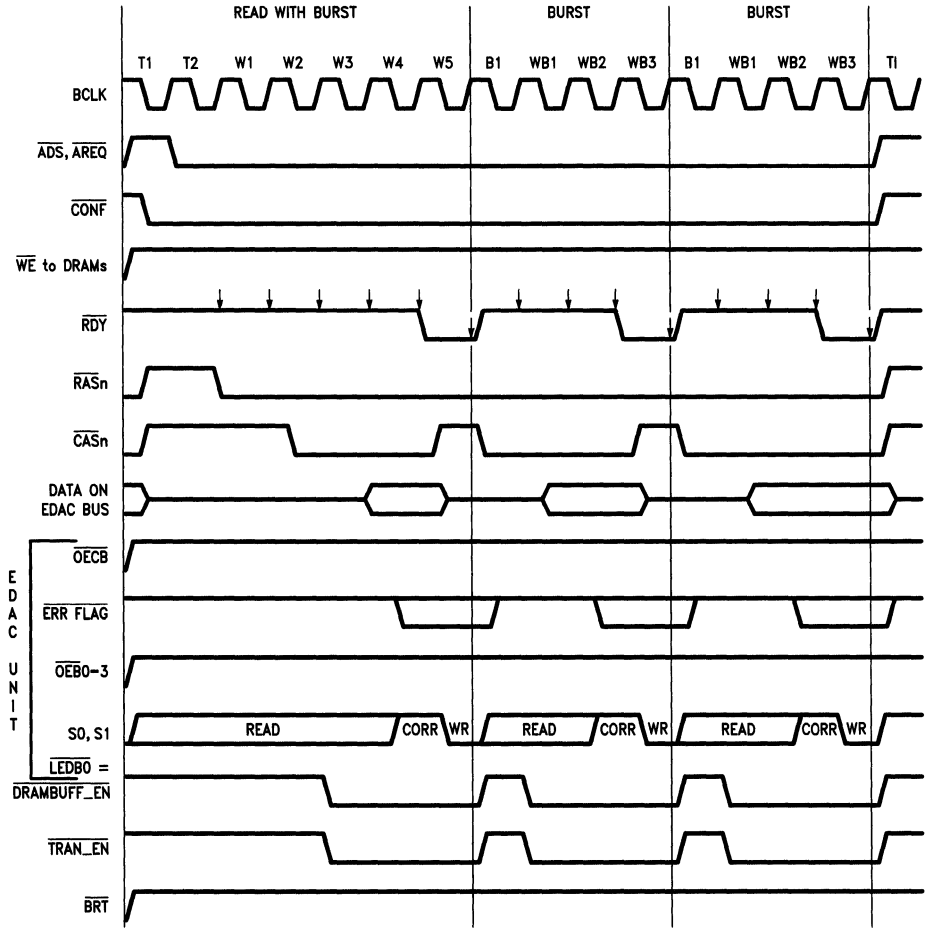
TL/F/9734-1

FIGURE 1.NS32532 Dual Access EDAC Memory System Using the Always Correct Method (see Section III for how Block Diagram Changes for Error Monitoring Method)



TL/F/9734–2

FIGURE 2. NS32532 Dual Access EDAC System Timing @ 25 MHz (DP8422A-25 74F632)
Error Monitoring Method Using the Bus Retry Input of the NS32532 and Registered Transceivers



TL/F/9734–3

FIGURE 3. NS32532 Dual Access EDAC System Timing @ 25 MHz (DP8422A-25, 74F632) Error Monitoring Method Using the Bus Retry Input of the NS32532 and Registered Transceivers

Interfacing the DP8420A/21A/22A to the 68000/008/010

National Semiconductor
Application Note 538
Joe Tate and Rusty Meier

INTRODUCTION

This application note explains interfacing the DP8420A/21A/22A DRAM controller to the 68000. Three different designs are shown and explained. It is assumed that the reader is familiar with the 68000 access cycles and the DP8420A/21A/22A modes of operation. This application note also applies to the 68010.

DESIGN #1 DESCRIPTION

Design #1 is a simple circuit to interface the 68000 to the DP8420A/21A/22A and up to 32 Mbytes of DRAM. The DP8420A/21A/22A is operated in Mode 1. An access cycle begins when the 68000 places a valid address on the address bus and asserts the address strobe ($\overline{AS}$). Chip select ($\overline{CS}$) is generated by a 74AS138 decoder. If a refresh or Port B access (DP8422A only) is not in progress, the DP8420A/21A/22A will assert the proper $\overline{RAS}$ depending on the bank select inputs (B0, B1). After guaranteeing the programmed value of row address hold time the DP8420A/21A/22A will switch the DRAM address (Q0–8, 9, 10) to the column address and assert $\overline{CAS}$. By this time, the 74AS245's have been enabled and the DRAMs place their data on the data bus. The DP8420A/21A/22A also asserts $\overline{DTACK}$ which is used to generate $\overline{DTACK}$ to the 68000 to complete the access.

If a refresh or Port B access had been in progress, the DP8420A/21A/22A would have delayed the 68000's access by inserting wait states into the access cycle until the refresh or Port B access was complete and the programmed amount of precharge time was met. This circuit can run up to 10 MHz with 0 wait states, with two or more banks. For 10 MHz, zero wait states with one bank, see design #2.

Timing parameters are referenced to the numbers shown in the DP8420A/21A/22A data sheet timing parameters. Numbered times starting with a "$" refer to the DP8420A/21A/22A timing parameters. Numbered times starting with "#" refer to the 68000 data sheet. Equations have been given to allow the user to calculate timing based on his frequency and application. The clock is at 10 MHz, a multiple of 2 MHz, allowing it to be tied directly to DELCLK. If DELCLK is not a multiple of 2 MHz, $\overline{ADS}$ to $\overline{CAS}$ must be recalculated.

DESIGN #1 TIMING AT 10 MHz AND 8 MHz

Clock Period = Tcp10 = 100 ns @ 10 MHz
= Tcp8 = 125 ns @ 8 ns

$400b: $\overline{ADS}$ Asserted Setup to CLK High
= Clock Period − CLK High to $\overline{AS}$ Asserted
= Tcp10 − #9
= 100 ns − 55 ns
= 45 ns @ 10 MHz
= Tcp8 − #9
= 125 ns − 60 ns
= 65 ns @ 8 MHz

$401: $\overline{CS}$ Setup to $\overline{ADS}$ Asserted
= 68000 Address to $\overline{AS}$ Max
− 74AS138 Decoder
= #11 − Tphl Max
= 20 ns − 9 ns
= 11 ns @ 10 MHz
= #11 − Tphl
= 30 ns − 9 ns
= 21 ns @ 8 MHz

$407 & $404: Address Valid Setup to $\overline{ADS}$ Asserted
= 68000 Address to $\overline{AS}$ Max
= #11 Max
= 20 ns @ 10 MHz
= #11 Max
= 30 ns @ 8 MHz

$405: $\overline{ADS}$ Negated Held from CLK High
= 68000 CLK High to $\overline{AS}$ Asserted Min
= #10 Min
= 0 ns @ 10 MHz
= #10 Min
= 0 ns @ 8 MHz

#47: DTACK Setup Time
= ½ Clock Period
− Clock to $\overline{DTACK}$ Asserted
= ½ Tcp10 − $18
= 50 ns − 28 ns
= 22 ns @ 10 MHz **Using 8420-25
= ½ Tcp8 − $18
= 62.5 ns − 33 ns
= 29.5 ns @ 8 MHz **Using 8420-25

$\overline{RAS}$ LOW DURING REFRESH

tRAS = Programmed Clock
− [(CLK High to Refresh $\overline{RAS}$ Asserted)
− (CLK High to Refresh $\overline{RAS}$ Negated)]
= Tcp10 + Tcp10 − $55
= 100 ns + 100 ns − 6 ns
= 194 ns @ 10 MHz
= Tcp8 + Tcp8 − $55
= 125 ns + 125 ns − 6 ns
= 244 ns @ 8 MHz

$\overline{RAS}$ PRECHARGE PARAMETERS**

tRP = (Programmed Clocks − 1)
− [($\overline{AREQ}$ to $\overline{RAS}$ Negated)
− (CLK to $\overline{RAS}$ Asserted)]
= Tcp10 − $50
= 100 ns − 16 ns
= 84 ns @ 10 MHz
= Tcp8 − $50
= 125 ns − 16 ns
= 109 ns @ 8 MHz

**To gain more precharge program 3t or use design #2.

tRAC AND tCAC FOR DRAMs

Timing is supplied for the system shown in *Figure 1. (see Figures 2, 3* and *4).* Since systems and DRAM times vary, the user is encouraged to change the following equations to match his system requirements. Timing has been supplied for systems with 0 or 1 wait state. If DELCLK is not a multiple of 2 MHz, the timing for tRAH and tASC will increase or decrease according to the equations given in the data sheet. The $\overline{ADS}$ to $\overline{RAS}$ and $\overline{ADS}$ to $\overline{CAS}$ will also have to be changed depending on the capacitance of the DRAM array.

0 Wait States

tRAC = s2 + s3 + s4 + s5 + s6 − CLK to $\overline{AS}$ Asserted Max − $\overline{ADS}$ Asserted to $\overline{RAS}$ Asserted − 74AS245 Delay Max − 68000 Data Setup Min
= 2½ Tcp10 − #9 − $402 − Tphl Max − #27
= 250 ns − 55 ns − 35 ns − 7 ns − 10 ns
= 143 ns @ 10 MHz Using 8420-20 w/Heavy Load
= 2½ Tcp8 − #9 − $402 − Tphl Max − #27
= 312.5 ns − 60 ns − 35 ns − 7 ns − 15 ns
= 195 ns @ 8 MHz Using 8420-20 w/Heavy Load

1 Wait State

tRAC = s2 + s3 + s4 + sw + sw + s5 + s6 − CLK to $\overline{AS}$ Asserted Max − $\overline{ADS}$ Asserted to $\overline{RAS}$ Asserted − 74AS245 Delay Max − 68000 Data Setup Min
= 3½ Tcp10 − #9 − $402 − Tphl Max − #27
= 350 ns − 55 ns − 35 ns − 7 ns − 10 ns
= 243 ns @ 10 MHz Using 8420-20 w/Heavy Load
= 3½ Tcp8 − #9 − $402 − Tphl Max − #27
= 437.5 ns − 60 ns − 35 ns − 7 ns − 15 ns
= 320 ns @ 8 MHz Using 8420-20 w/Heavy Load

0 Wait States

tCAC = s2 + s3 + s4 + s5 + s6 − CLK to $\overline{AS}$ Asserted Max − $\overline{ADS}$ Asserted to $\overline{CAS}$ Asserted − 74AS245 Delay Max − 68000 Data Setup Min
= 2½ Tcp10 − #9 − $403a − Tphl Max − #27
= 250 ns − 55 ns − 94 ns − 7 ns − 10 ns
= 84 ns @ 10 MHz Using 8420-20 w/Heavy Load
= 2½ Tcp8 − #9 − $403a − Tphl Max − #27
= 312.5 ns − 60 ns − 94 ns − 7 ns − 15 ns
= 136 ns @ 8 MHz Using 8420-20 w/Heavy Load

1 Wait State

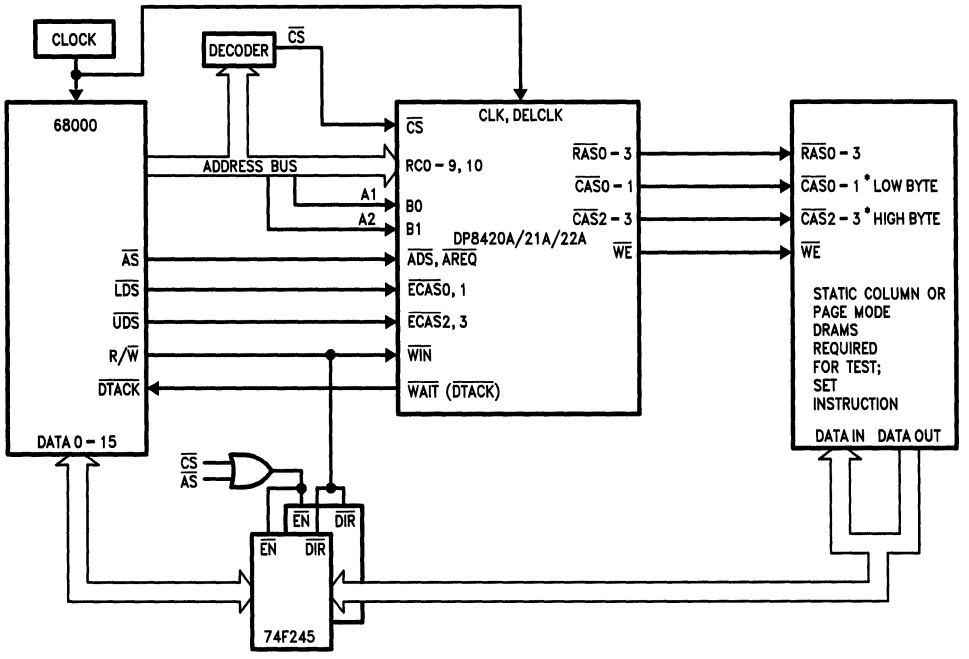
tCAC = s2 + s3 + s4 + sw + sw + s6 − CLK to $\overline{AS}$ Asserted Max − $\overline{ADS}$ Asserted to $\overline{CAS}$ Asserted − 74AS245 Delay Max − 68000 Data Setup Min
= 3½ Tcp10 − #9 − $403a − Tphl Max − #27
= 350 ns − 55 ns − 94 ns − 7 ns − 10 ns
= 184 ns @ 10 MHz Using 8420-20 w/Heavy Load
= 3½ Tcp8 − #9 − $403a − Tphl Max − #27
= 437.5 ns − 60 ns − 94 ns − 7 ns − 15 ns
= 261 ns @ 8 MHz Using 8420-20 w/Heavy Load

Design #1 Programming Bits

Bits	Description	Value
R0, R1	$\overline{RAS}$ Low Time During REFRESH = 2T $\overline{RAS}$ Precharge Time = 2T	R0 = 0 R1 = 1
R2, R3	$\overline{DTACK}$ Generation Modes for Non-Burst Accesses	R2 = s R3 = s
R4, R5	$\overline{DTACK}$ Generation Modes for Burst Accesses	R4 = s R5 = s
R6	Add Wait States with $\overline{WAITIN}$	R6 = s
R7	$\overline{DTACK}$ Mode Select	R7 = 1
R8	Non Interleaved Mode	R8 = 1
R9	Staggered or All RAS REFRESH	R9 = u
C0, C1, C2	Divisor for DELCLK	C0 = s C1 = s C2 = s
C3	+30 REFRESH	C3 = 0
C4, C5, C6	$\overline{RAS}$, $\overline{CAS}$ Configuration Mode *Choose All $\overline{CAS}$ Mode	C4 = u C5 = u C6 = u
C7	Select 0 ns Column Address Setup	C7 = 1
C8	Select 15 ns Row Address Setup	C8 = 1
C9	CAS is Delayed to the Next Rising CLK Edge During Writes	C9 = 1
B0	The Row/Column Bank Latches Are Fall Through Mode	B0 = 1
B1	Access Mode 1	B1 = 1
$\overline{ECAS0}$	$\overline{CAS}$ Not Extended Beyond $\overline{RAS}$	$\overline{ECAS0}$ = 0

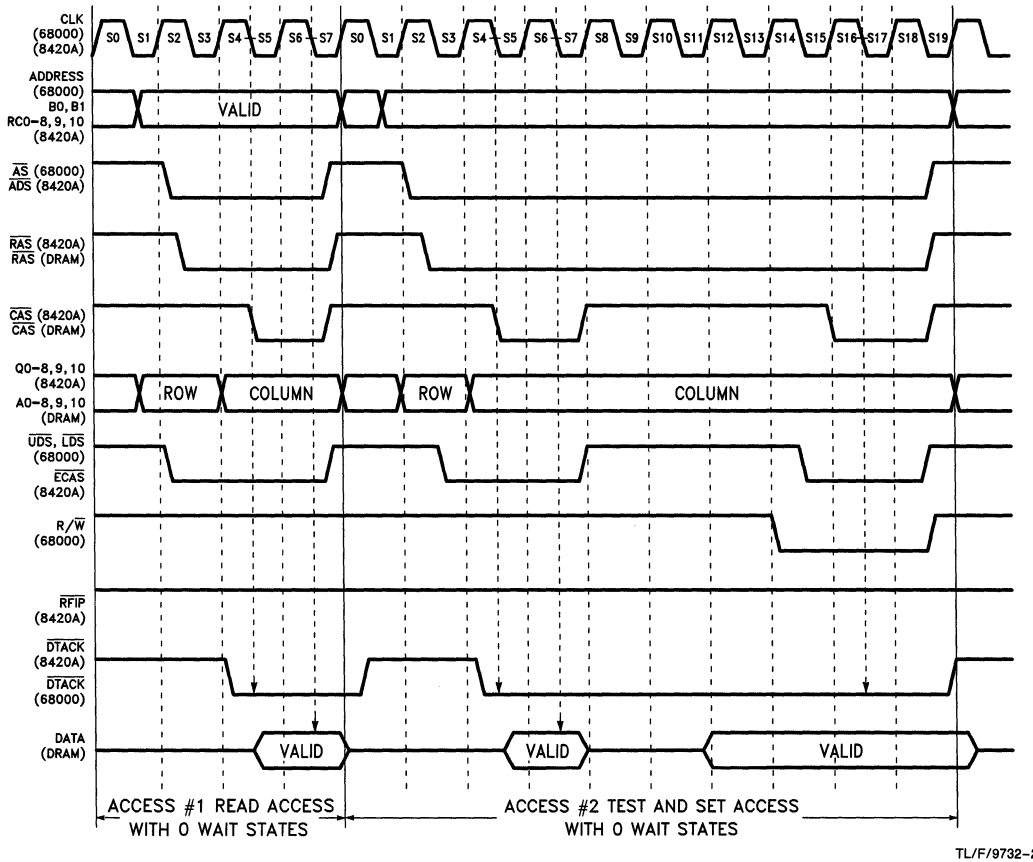
u = user defined **s = system dependent**

R2 = 1	R3 = 0		for 0 WAIT STATES
R2 = 1	R3 = 0	R6 = 0	for 1 WAIT STATE
C0 = 1	C1 = 0	C2 = 1	for 10 MHz
C0 = 0	C1 = 0	C2 = 1	for 8 MHz
R4 = 0	R5 = 0		for 0 WAIT STATES during write portion of test and set
R4 = 1	R5 = 1		for 1 WAIT STATE during write portion of test and set



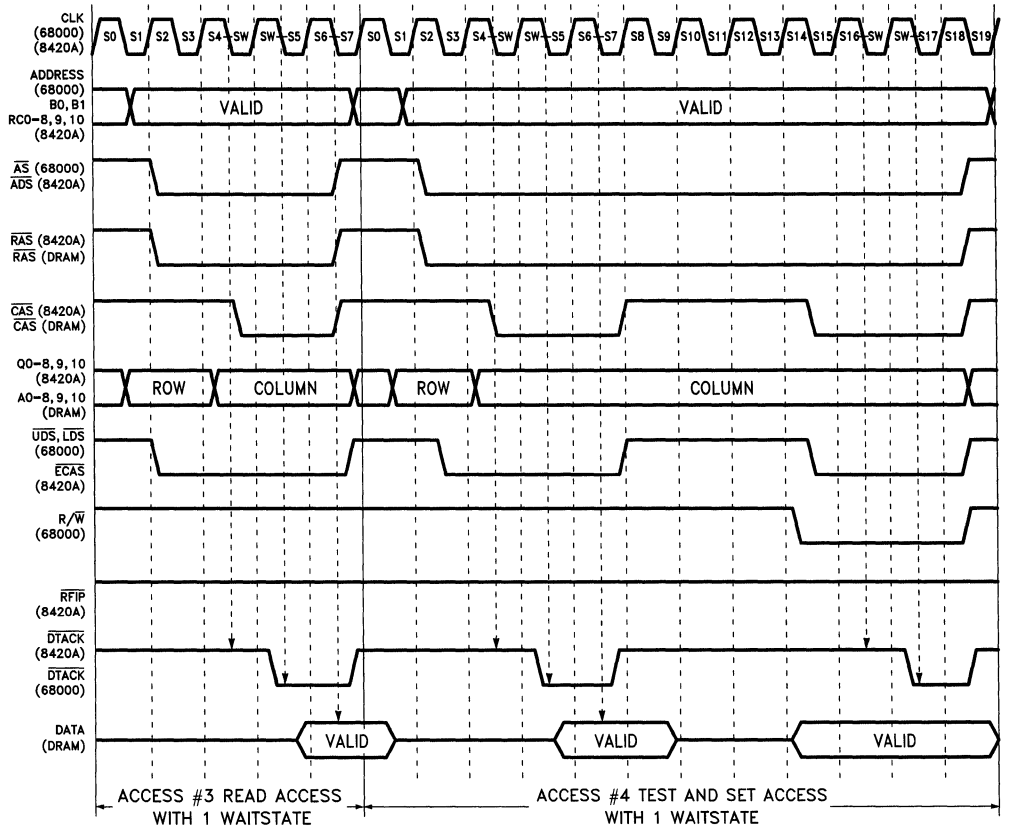
TL/F/9732–1

FIGURE 1. 68000 Design # 1



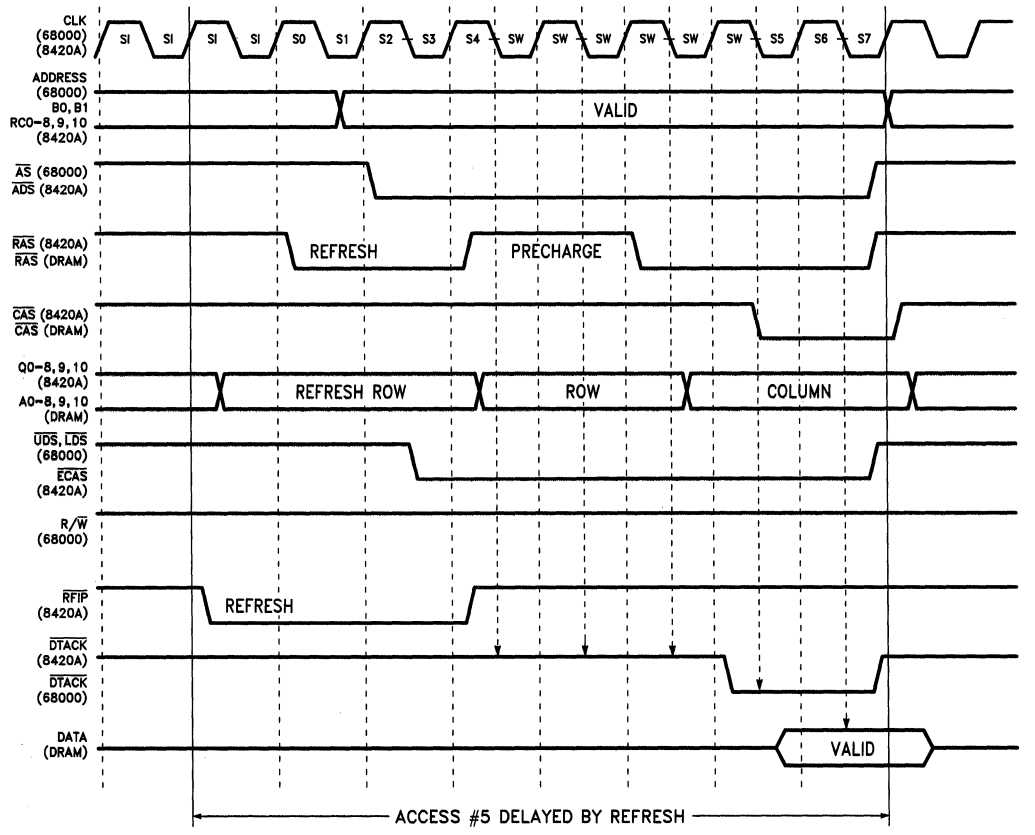
TL/F/9732–2

FIGURE 2. 68000 Design #1 Timing



TL/F/9732-3

FIGURE 3. 68000 Design # 1 Timing



TL/F/9732–4

FIGURE 4. 68000 Design # 1 Timing

DESIGN #2 DESCRIPTION

Design #2 differs from Design #1 in that the 68000 can be run up to 12.5 MHz. This design can also run with no wait states at 10 MHz if only one bank of DRAM is being used. A latch must be used with the 68000 address strobe to guarantee the address setup to $\overline{ADS}$ asserted requirement of the DP8420A/21A/22A. Again, the DP8420A/21A/22A is operated in Mode 1.

An access cycle begins when the 68000 places a valid address on the address bus at the beginning of processor state s1. At processor state s2, the 68000 asserts the address strobe, $\overline{AS}$. This signal is qualified with CLK low to set a latch. The output of this latch produces the signal $\overline{ADS}$ to the DP8420A/21A/22A. When the signal $\overline{ADS}$ is asserted on the DP8420A/21A/22A, the chip will assert $\overline{RAS}$. After guaranteeing the row address hold time, the 8420A/21A/22A will place the column address to the DRAM address bus. After guaranteeing the column address setup time, the DP8420A/21A/22A will assert $\overline{CAS}$. After time tCAC has passed, the DRAM will place its data on the data bus. The 8420A/21A/22A will assert the $\overline{DTACK}$ output allowing the bus cycle to end.

If a refresh of a Port B access had been in progress, the access would have been delayed by inserting wait states in the Port A access cycle.

DESIGN #2 TIMING AT 12.5 MHz

Clock Period = Tcp12 80 ns @ 12.5 MHz

$400b: $\overline{ADS}$ Asserted Setup to CLK High

= Clock Period + ½ Clock Period + 74AS04 Delay Min + 74AS04 Delay Min − Clock to AS Asserted Max − 74AS04 Delay Min − 74AS02 Delay Max − 74AS02 Delay Max

= Tcp12 + ½ Tcp12 + Tphl Min + Tphl Min − #9 − Tphl Min − Tphl Max − Tphl Max

= 80 ns + 40 ns + 1 ns + 1 ns − 55 ns − 1 ns − 4.5 ns − 4.5 ns

= 57 ns @ 12.5 MHz

$401: $\overline{CS}$ Setup to $\overline{ADS}$ Asserted

= Clock Period + 74AS04 Delay Min + 74AS04 Delay Min + 74AS02 Delay Min + 74AS02 Delay Min − 74AS04 Delay Min − Clock to ADR Max − 74AS138 Delay Max

= Tcp12 + Tphl Min + Tphl Min + Tphl Min + Tphl Min − Tphl Min − #6 − Tphl Max

= 80 ns + 1 ns + 1 ns + 1 ns + 1 ns − 1 ns − 55 ns − 9 ns

= 19 ns @ 12.5 MHz

$407 & $404: Address Valid to $\overline{ADS}$ Asserted

= Clock Period + 74AS04 Delay Min + 74AS04 Delay Min + 74AS02 Delay Min + 74AS02 Delay Min − Clock to ADR Max − 74AS04 Min

= Tcp12 + Tphl + Tphl + Tphl + Tphl − #6 − Tphl

= 80 ns + 1 ns + 1 ns + 1 ns + 1 ns − 55 ns − 1 ns

= 28 ns @ 12.5 MHz

$405: $\overline{ADS}$ Negated Held from CLK High

= Min 74AS04 + Min 74AS02 + Min 74AS02 + Min 74AS04 − Min 74AS04

= Tphl + Tphl + Tphl + Tphl − Tphl

= 1 ns + 1 ns + 1 ns + 1 ns − 1 ns

= 3 ns @ 12.5 MHz

#47: DTACK Setup Time

= 1 Clock Period − CLOCK skew (74AS04) − Max Clock to DTACK

= Tcp12 − Tphl Max − $18

= 80 ns − 5 ns − 28 ns

= 47 ns @ 12.5 MHz

$\overline{RAS}$ LOW DURING REFRESH

tRAS = Programmed Clock − [(CLK High to Refresh $\overline{RAS}$ Asserted) − (CLK High to Refresh $\overline{RAS}$ Negated)]

= Tcp12 + Tcp12 − $55

= 80 ns + 80 ns − 6 ns

= 154 ns @ 12.5 MHz

$\overline{RAS}$ PRECHARGE PARAMETERS

tRP = Programmed Clocks − Clock to $\overline{AS}$ Negated − [($\overline{AREQ}$ to $\overline{RAS}$ Negated) − (CLK to $\overline{RAS}$ Asserted)]

= Tcp12 + Tcp12 − $50

= 80 ns + 80 ns − 16 ns

= 144 ns @ 12.5 MHz

$29b: $\overline{AREQ}$ Negated Setup to CLK

= Clock Period + Min CLOCK Skew 74AS04 − Max 74AS02 − Max 74AS02

= Tcp12 + Tphl + Tphl − Tphl

= 80 ns + 1 ns − 4.5 ns − 4.5 ns

= 72 ns @ 12.5 MHz

tRAC AND tCAC FOR DRAMs

Timing is supplied for the system shown in *Figure 5*. (See *Figures 6*). Since systems and DRAM times vary, the user is encouraged to change the following equations to match his system. Timing has been suppiled for systems with 0 wait states and 1 bank of DRAM and 1 wait state and 4 banks of DRAM. If DELCLK is not a multiple of 2 MHz, the times of tRAH and tASC will increase or decrease according to the equations given in the data sheet. The $\overline{ADS}$ to $\overline{RAS}$ and $\overline{ADS}$ to $\overline{CAS}$ will also have to be changed depending on the capacitance of the DRAM array.

tRAC

0 wait states * does not use transceivers *

tRAC = s2 + s3 + s4 + s5 + s6 − 74AS02 Max − 74AS02 Max − Clock to AS Max − $\overline{ADS}$ to $\overline{RAS}$ − Data Setup

= 2½ Tcp12 − Tphl − Tphl − #9 − $402 − #27

= 200 − 4.5 ns − 4.5 ns − 55 ns − 25 ns − 10 ns

= 101 ns @ 12.5 MHz **Using 8420-25 w/Light Load

1 wait state * uses transceivers *

tRAC = s2 + s3 + s4 + sw + sw + s5 + s6 − 74AS02 Max − 7AS02 Max − Clock to $\overline{AS}$ Max − $\overline{ADS}$ to $\overline{RAS}$ − 74AS245 Delay − Data Setup

= 3½ Tcp12 − Tphl − Tphl − #9 − $402 − Tphl − #27

= 280 ns − 4.5 ns − 4.5 ns − 55 ns − 29 ns − 7 ns − 10 ns

= 170 ns @ 12.5 MHz

tCAC

0 wait states * does not use transceivers *

tCAC = s2 + s3 + s4 + s5 + s6 − 74AS02 Max − 74AS02 Max − Clock to $\overline{AS}$ Max − $\overline{ADS}$ Asserted to $\overline{CAS}$ − Data Setup

= 2½ Tcp12 − Tphl − Tphl − #9 − $403a − #27

= 200 ns − 4.5 ns − 4.5 ns − 55 ns − 75 ns − 10 ns

= 51 ns @ 12.5 MHz *Using 8420-25 w/Light Load

1 wait state * uses transceivers *

tCAC = s2 + s3 + s4 + sw + sw + s5 + s6 − 74AS02 Max Delay − 74AS02 Max Delay − Clock to $\overline{AS}$ Max − $\overline{ADS}$ Asserted to $\overline{CAS}$ − 74AS245 Data Setup

= 3½ Tcp12 − Tphl − Tphl − #9 − $403a − Tphl − #27

= 280 ns − 4.5 ns − 4.5 ns − 55 ns − 75 ns − 7 ns − 10 ns

= 124 ns @ 12.5 MHz

DESIGN #2, 0 WAIT STATES DURING WRITE ACCESS

Design #2 can be modified to allow 0 wait states during writes. To accomplish this, the chip must be programmed with the same value except that bits R2, R3 and R6 are changed to:

R2 = 0 DTACK of 0T from RAS
R3 = 0
R6 = 0 Hold off DTACK 1 extra clock period

The hardware must be modifed. The signal R/W from the 68000 is inverted and tied to the 8420 signal $\overline{WAITIN}$. This ensures that a wait state will only be asserted during read accesses (see *Figure 6*).

0 waits during write access timing

$\overline{RAS}$ Low Time

tRP = Max $\overline{AS}$ Low − ½ Clock Period − 74AS02 Delay − 74AS02 Delay + 74AS02 Delay + 74AS02 Delay − [($\overline{ADS}$ Asserted to $\overline{RAS}$) − ($\overline{AREQ}$ Negated to $\overline{RAS}$ Negated)]

= #14 − ½ Tcp12 − Tphl − Tphl + Tphl + Tphl − $52

= 160 ns − 40 ns − 0 ns

= 120 ns @ 12.5 MHz

$\overline{CAS}$ Low Time

tCP = s2 + s3 + s4 + s5 + s6 − Max CLK to $\overline{AS}$ − 74AS02 − 74AS02 − Max $\overline{AS}$ to $\overline{CAS}$ + Min CLK to $\overline{DS}$ + Min ECAS to CAS

= 2½ Tcp12 − #9 − Tphl − Tphl − $403a + #12 + $14

= 200 ns − 55 ns − 4.5 ns − 4.5 ns − 82 ns + 0 ns + 0 ns

= 54 ns @ 12.5 MHz

Design #2 Programming Bits

Bits	Description	Value
R0, R1	$\overline{RAS}$ Low Time = 2T $\overline{RAS}$ Precharge Time = 2T	R0 = 0 R1 = 1
R2, R3	$\overline{DTACK}$ Generation Modes for Non-Burst Accesses	R2 = 0 R3 = 1
R4, R5	$\overline{DTACK}$ Generation Modes for Burst Accesses	R4 = 0 R5 = 1
R6	Add Wait States with $\overline{WAITIN}$	R6 = 0
R7	$\overline{DTACK}$ Mode Select	R7 = 1
R8	Non Interleaved Mode	R8 = 1
R9	Staggered or All RAS REFRESH	R9 = u
C0, C1, C2	Divisor for DELCLK	C0 = u C1 = u C2 = u
C3	+30 REFRESH	C3 = 0
C4, C5, C6	$\overline{RAS}$, $\overline{CAS}$ Configuration Mode *Choose All $\overline{CAS}$ Mode	C4 = u C5 = u C6 = u
C7	Select 15 ns Column Address Setup	C7 = 1
C8	Select 15 ns Row Address Setup	C8 = 1
C9	CAS is Delayed to the Next Rising CLK Edge During Writes	C9 = 1
B0	The Row/Column Bank Latches Are Fall Through Mode	B0 = 1
B1	Access Mode 1	B1 = 1
$\overline{ECAS0}$	$\overline{CAS}$ Not Extended Beyond $\overline{RAS}$	$\overline{ECAS0}$ = 0

u = user defined

*see previous page for 0 WAIT STATES during writes

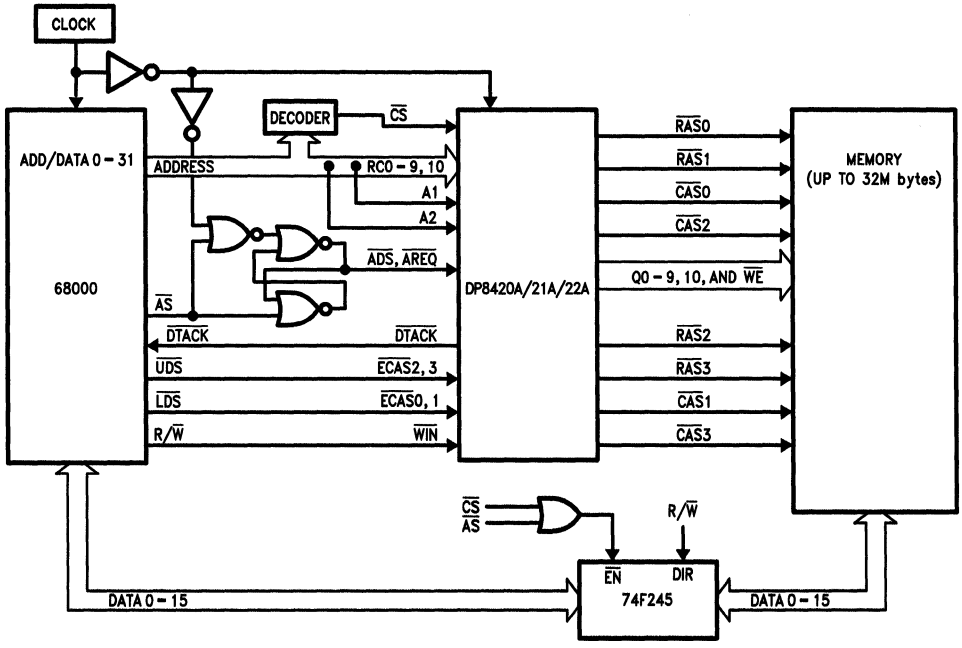
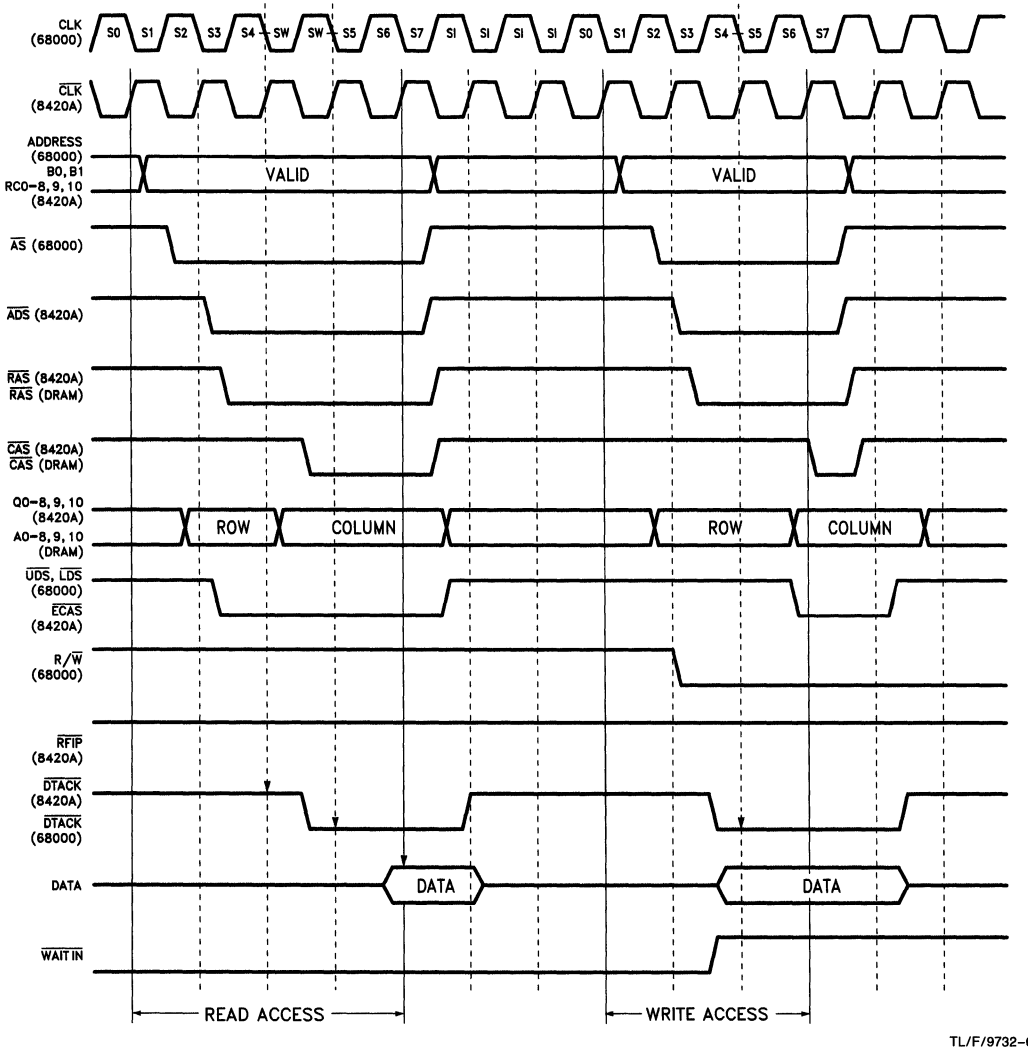


FIGURE 5. 68000 Design #2 up to 12.5 MHz



TL/F/9732–6

FIGURE 6. Design #2 Timing with Zero Wait States during Writes

DESIGN #3 DESCRIPTION

Design #3 is a simple circuit to interface the 68000 running @ 16 MHz to the DP8420A/21A/22A and up to 32 Mbytes of DRAM. The DP8420A/21A/22A is operated in Mode 1. An access cycle begins when the 68000 places a valid address on the address bus and asserts $\overline{AS}$. $\overline{AS}$ is then clocked with a 74AS74 flip-flop. The output of the flip-flop is used to produce $\overline{ADS}$ to the DP8420A/21A/22A.

Chip Select ($\overline{CS}$) is generated by a 74AS138 decoder. If a refresh or Port B access had been in progress, the 8420A/21A/22A would hold off the access by inserting wait states in the access cycle. The DP8420A/21A/22A will place the row address on the DRAM's address bus and assert $\overline{RAS}$. After guaranteeing the row address hold time, tRAH, the DP8420A/21A/22A will place the column address on the DRAM's address bus and assert $\overline{CAS}$.

DESIGN #3 TIMING AT 16.667 MHz

Clock Period = Tcp16 = 60 ns @ 16.667 MHz

$400b: $\overline{ADS}$ Asserted Setup to CLK High
= Clock Period − 74AS74 Delay Max
= Tcp16 − Tphl
= 60 ns − 9 ns
= 51 ns @ 16.667 MHz

$401: $\overline{CS}$ Asserted Setup to $\overline{ADS}$ Asserted
= 1½ Clock Periods + Min 74AS74 Delay − Max Clock to Address − 74AS138 Delay
= 1½ Tcp16 + Tphl − #6 − Tphl
= 90 ns + 4.5 ns + 50 ns − 9 ns
= 35.5 ns @ 16.667 MHz

$407 & $404: Address Valid Setup to $\overline{ADS}$ Asserted
= 1½ Clock Periods + Min 74AS74 Delay − Max Clock to Address
= 1½ Tcp16 + Tphl − #6
= 90 ns + 4.5 ns − 50 ns
= 44.5 ns @ 16.667 MHz

$405: $\overline{ADS}$ Negated Held from CLK High
= Min 74AS74 Delay
= 4.5 ns @ 16.667 MHz

#47: DTACK Setup Time
= Clock Period − 74AS74 Delay Max
= Tcp16 − Tphl
= 60 ns − 9 ns
= 51 ns @ 16.667 MHz

$\overline{RAS}$ LOW DURING REFRESH

tRAS = Programmed Clocks − [(CLK High to Refresh $\overline{RAS}$ Asserted) − (CLK High to Refresh $\overline{RAS}$ Negated)]
= Tcp16 + Tcp16 + Tcp16 + Tcp16 − $55
= 240 ns − 6 ns
= 234 ns @ 16.667 MHz

tRP = (Programmed Clocks − 1) − [($\overline{AREQ}$ to $\overline{RAS}$ Negated) − (CLK to $\overline{RAS}$ Asserted)]
= Tcp16 + Tcp16 − $50
= 120 ns − 16 ns
= 104 ns @ 16.667 MHz

$\overline{RAS}$ PRECHARGE PARAMETERS

tRP = Programmed Clocks − Clock to $\overline{AS}$ Negated − [($\overline{AREQ}$ to $\overline{RAS}$ Negated) − (CLK to $\overline{RAS}$ Asserted)]

tRAC AND tCAC FOR DRAMs

Timing is supplied for the system shown in *Figure 7*. Since system and DRAM times vary, the user is encouraged to change the following equations to match his system requirements. Timing has been supplied for systems with 2 wait states. If DELCLK is not a multiple of 2 MHz, the timing for tRAH and tASC will increase or decrease according to the times given in the data sheet. The $\overline{ADS}$ to $\overline{RAS}$ and $\overline{ADS}$ to $\overline{CAS}$ will also have to be changed depending on the capacitance of the DRAM array.

1 wait state * using 1 BANK with no transceivers

tRAC = s4 + sw + sw + s5 + s6 − 74AS74 Delay − $\overline{ADS}$ to $\overline{RAS}$ − Data Setup
= 2½ Tcp16 − Tphl − $402 − #27
= 150 ns − 9 ns − 25 ns − 10 ns
= 106 ns @ 16.667 MHz Using 8420-25 w/Light Load

2 wait states * uses 4 banks with tranceivers *

tRAC = s4 + sw + sw + sw + sw + s5 + s6 − 74AS74 Delay − $\overline{ADS}$ to $\overline{RAS}$ − Data Setup − Transceivers
= 3½ Tcp16 − Tphl − $402 − #27 − Tphl
= 210 ns − 9 ns − 29 ns − 10 ns − 7 ns
= 155 ns @ 16.667 MHz

1 wait state * using 1 BANK with no transceivers

tCAC = s4 + sw + sw + s5 + s6
− 74AS74 Delay − $\overline{ADS}$ to $\overline{CAS}$
− Data Setup
= 2½ Tcp16 − Tphl − $403a − #27
= 150 ns − 9 ns − 75 ns − 10 ns
= 56 ns @ 16 MHz

2 wait states * using 4 banks with transceivers *

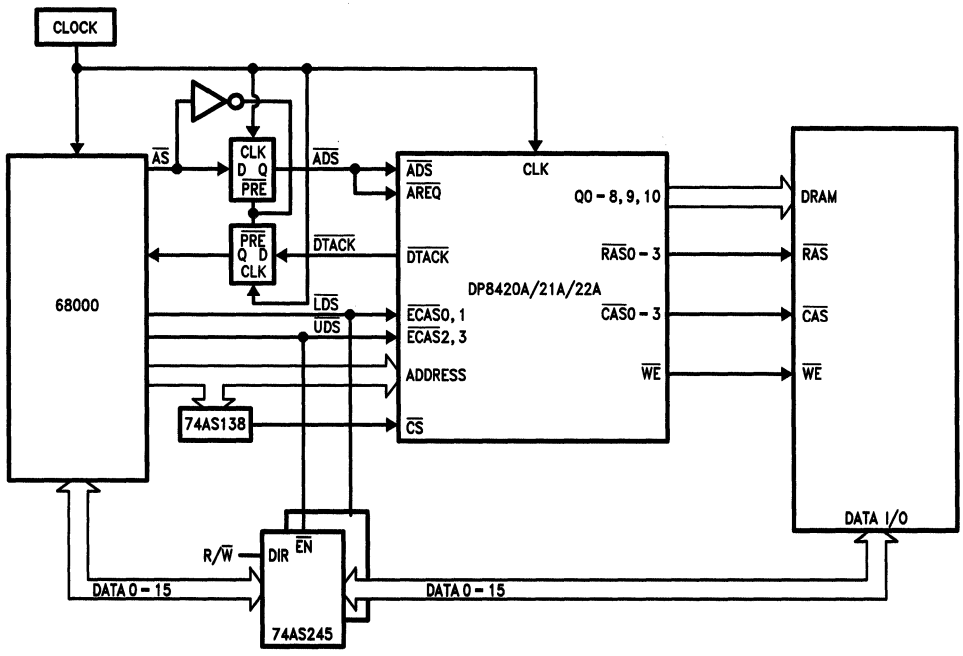
tCAC = s4 + sw + sw + sw + sw + s5 + s6
− 74AS74 Delay − $\overline{ADS}$ to $\overline{CAS}$
− Data Setup − Transceiver
= 3½ Tcp16 − Tphl − $403a
− #27 − Tphl
= 210 ns − 9 ns − 82 ns
− 10 ns − 7 ns
= 102 ns @ 16 MHz

Design #3 Programming Bits

Bits	Description	Value
R0, R1	$\overline{RAS}$ Low Time = 2T $\overline{RAS}$ Precharge Time = 2T	R0 = 0 R1 = 1
R2, R3	$\overline{DTACK}$ Generation Modes for Non-Burst Accesses	R2 = 1 R3 = 0
R4, R5	$\overline{DTACK}$ Generation Modes for Burst Accesses	R4 = u R5 = u
R6	Add Wait States with $\overline{WAITIN}$	R6 = u
R7	$\overline{DTACK}$ Mode Select	R7 = 1
R8	Non Interleaved Mode	R8 = 1
R9	Staggered or All RAS REFRESH	R9 = u
C0, C1, C2	Divisor for DELCLK (+8 for 16 MHz)	C0 = 0 C1 = 1 C2 = 0
C3	+30 REFRESH	C3 = 0
C4, C5, C6	$\overline{RAS}$, $\overline{CAS}$ Configuration Mode *Choose All $\overline{CAS}$ Mode	C4 = u C5 = u C6 = u
C7	Select 15 ns Column Address Setup	C7 = 1
C8	Select 15 ns Row Address Setup	C8 = 1
C9	CAS is Delayed to the Next Rising CLK Edge During Writes	C9 = 1
B0	The Row/Column Bank Latches Are Fall Through Mode	B0 = 1
B1	Access Mode 1	B1 = 1
$\overline{ECAS0}$	$\overline{CAS}$ Not Extended Beyond $\overline{RAS}$	$\overline{ECAS0}$ = 0

u = user defined

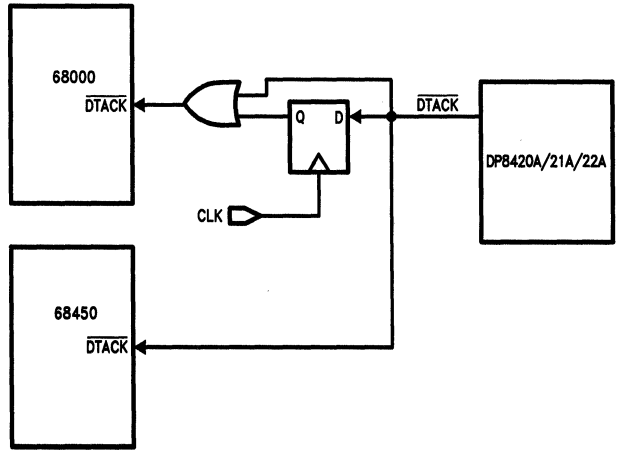
*see previous page for 0 WAIT STATES during writes



TL/F/9732-7

FIGURE 7. 68000 Design #3, Works up to 16 MHz

Additional Circuitry for Design #1
Using the 68450 DMA Controller



TL/F/9732-8

Because the 68450 samples $\overline{DTACK}$ on a positive edge of CLK and the 68000 samples $\overline{DTACK}$ on the negative edge, additional circuitry must be added to produce the two $\overline{DTACK}$ signals. The $\overline{DTACK}$s must be produced different to ensure $\overline{RAS}$ low time after an access delayed by a refresh. The programming bits must also be changed as follows:

```
For 0 WAITSTATES

R2 = 0 R3 = 1 FOR /DTACK OF 1/2

For 1 WAITSTATE

R2 = 0 R3 = 1 R6 = 0 FOR /DTACK OF 1 1/2
```

Tie the DP8420 signal $\overline{WAITIN}$ low for 1 waitstate and high for 0 waitstates. All timing except for the following should still apply. Times with a "#" refer to the 68000 data sheet. Times with a "!" refer to the 68450 data sheet and times with a "$" refer to the DP8420A/21A/22A data sheet.

$47: $\overline{DTACK}$ Setup Time

= ½ CLOCK Period − 74AS74 CLOCK to Q − 74AS32

= ½ Tcp10 − Tphl − Tphl

= 50 ns − 9 ns − 6 ns

= 35 ns @ 10 MHz

= ½ Tcp8 − Tphl − Tphl

= 62.5 ns − 9 ns − 6 ns

= 47 ns @ 8 MHz

!6: $\overline{DTACK}$ Setup Time (68450)

= ½ CLOCK Period − CLOCK to $\overline{DTACK}$

= ½ Tcp10 − $18

= 50 ns − 28 ns

= 22 ns @ 10 MHz

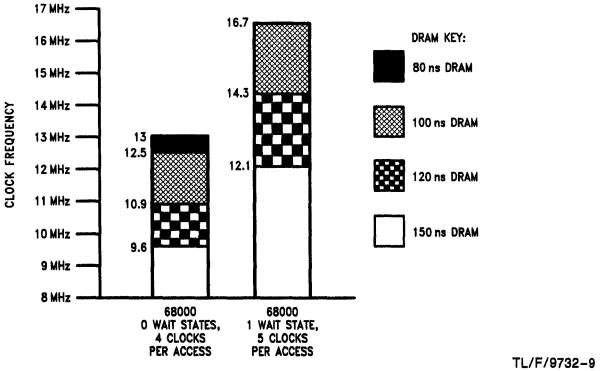
= ½ Tcp8 − $18

= 62.5 ns − 33 ns

= 29 ns @ 8 MHz

All other 68450 times are the same as the 68000.

DRAM Speed Versus Processor Speed,
(DRAM Speed References the RAS Access Time, tRAC, of the DRAM.
Using DP8422A-25 Timing Specifications)



TL/F/9732–9

Interfacing the DP8422A to the 68000-16 (Zero Wait State Burst Mode Access)

National Semiconductor
Application Note 615
Lawson H.C. Chang

INTRODUCTION

This application note describes interfacing the DP8422A DRAM controller (also applicable to DP8420A/21A) to the 68000 (16 MHz) with slower memories. This design is based upon burst mode access by holding $\overline{RAS}$ low and toggling $\overline{CAS}$. It is assumed that the user is familiar with the DP8422A and 68000 mode operations.

DESIGN DESCRIPTION

This design consists of the DP8422A DRAM controller, a PAL (20R4D), and a page detector (ALS6311). This design accommodates four banks of DRAM, each bank being 16 bits in width, giving maximum memory capacity of either 2 Mbytes (using 256k x 4 light load DRAMs) or 8 Mbytes (using 1M x 1 DRAMs). The schematic diagram of interfacing DP8422A to the 68000 is shown in *Figure 1*. The DP8422A is operated in Mode 1. An access cycle begins when the 68000 places a valid address on the address bus and asserts the Address Strobe ($\overline{AS}$) if a refresh or Port B access (DP8422A only) is not in progress. The proper $\overline{RAS}$ and $\overline{CAS}$ will be asserted respectively, depending upon programming bits C6, C5, and C4 for $\overline{RAS}$ and $\overline{CAS}$ configuration after guaranteeing the programmed value of row address hold time and the column address setup time.

The High Speed Access ($\overline{HSA}$) output signal of page detector indicates whether the current access is in the same page as previous access or not. $\overline{ADS}$ (AREQ) is kept low if the current access is in the page, otherwise $\overline{ADS}$ ($\overline{AREQ}$) will be forced to go high to terminate the burst access. Internal refresh logic automatically generates refresh request every 15 μs. Since the 256k x 4 DRAM data input and output signals can be controlled by the Output Enable ($\overline{OE}$).

Transceivers (F245) could be eliminated from this design if 256k x 4 DRAM light load DRAM were used. The timing diagrams are shown in *Figure 2* and *Figure 3*.

DP8422A PROGRAMMING BITS

u = user defined

Programming Bits		Description
R0,R1	= 0,1	$\overline{RAS}$ high and low times
R2,R3	= u,u	$\overline{DTACK}$ generation mode for nonburst access
R4,R5	= u,u	$\overline{DTACK}$ generation mode for burst access
R6	= 0	Add wait states if $\overline{WAITIN}$ is low
R7	= 1	$\overline{DTACK}$ mode select
R8	= 1	Noninterleave mode
R9	= u	All $\overline{RAS}$'s or staggered refresh select
C0,C1,C2	= 0,1,0	Refresh clock divisor select
C3	= 0	Refresh clock divider select
C4,C5,C6	= u,u,u	$\overline{RAS}$ and $\overline{CAS}$ configuration mode
C7	= 1	tASC mode select
C8	= 1	tRAH mode select
C9	= u	Delay $\overline{CAS}$ during write access mode select

Programming Bits		Description
B0	= 1	Fall through mode
B1	= 1	Mode 1 access
$\overline{ECAS0}$	= 1	Extend $\overline{CAS}$ and refresh request

DESIGN TIMING PARAMETERS

Timing parameters are referenced to the numbers shown in the DP8422A data sheet timing parameters. Numbered times starting with a "$" refer to DP8422A timing parameters. Numbered times starting with a "#" refer to 68000 timing parameters.

16 MHz Tcp = 62.5 ns

$400b: $\overline{ADS}$ asserted setup to CLK

Tcp − PAL tCLK max.
= 62.5 ns − 8 ns
= 54.5 ns

$401: $\overline{CS}$ setup to $\overline{ADS}$ asserted

2 Tcp − #6 $\overline{CLK}$ to Address valid − PAL tp max.
= 125 ns − 55 ns − 10 ns
= 60 ns

#47: $\overline{DTACK}$ (68000) low setup to CLK low

Tcp − $18 CLK to $\overline{DTACK}$ (DP8422A) asserted − PAL tp max.
= 62.5 ns − 28 ns − 10 ns
= 24.5 ns (DP8422A-25)
= 62.5 ns − 33 ns − 10 ns
= 19.5 ns (DP8422A-20)

I. LIGHT LOAD TIMING

tRAC (nonburst access):

3 Tcp − PAL tCLK max. − $402 $\overline{ADS}$ low to $\overline{RAS}$ low − #27 data setup − F245 Transceiver tp max.
= 187.5 ns − 8 ns − 25 ns − 7 ns − 6 ns
= 141.5 ns (DP8422A-25)
= 187.5 ns − 8 ns − 30 ns − 7 ns − 6 ns
= 136.5 ns (DP8422A-20)

tCAC (nonburst access):

3 Tcp − PAL tCLK max. − $403a $\overline{ADS}$ low to $\overline{CAS}$ low − #27 data setup − F245 Transceiver tp max.
= 187.5 ns − 8 ns − 75 ns − 7 ns − 6 ns
= 91.5 ns (DP8422A-25)
= 187.5 ns − 8 ns − 86 ns − 7 ns − 6 ns
= 80.5 ns (DP8422A-20)

tAA (nonburst access):

3 Tcp − PAL tCLK max. − $417 $\overline{ADS}$ low to Column Address valid − #27 data setup − F245 Transceiver tp max.
= 187.5 ns − 8 ns − 69 ns − 7 ns − 6 ns
= 97.5 ns (DP8422A-25)
= 187.5 ns − 8 ns − 83 ns − 7 ns − 6 ns
= 83.5 ns (DP8422A-20)

tCAC (burst access):

2.5 Tcp−#9 $\overline{CLK}$ high to $\overline{DS}$ low max.−PAL tp max.−$14 $\overline{ECAS}$ low to $\overline{CAS}$ low max.−#27 data setup−F245 Transceiver tp max.

= 156 ns−40 ns−10 ns−20 ns−7−6 ns

= 73 ns (DP8422A-25)

= 156 ns−40 ns−10 ns−23 ns−7 ns−6 ns

= 70 ns (DP8422A-20)

tAA (burst access):

3 Tcp−#6 $\overline{CLK}$ low to Address valid max.−$26 Address valid to Q max. − #27 data setup−F245 Transceiver tp max.

= 187.5 ns−5 ns−26 ns−7 ns−6 ns

= 98.5 ns (DP8422A-25)

= 187.5 ns−5 ns−29 ns−7 ns−6 ns

= 95.5 ns (DP8422A-20)

II. HEAVY LOAD TIMING

tRAC (nonburst access):

3 Tcp−PAL tCLK max.−$402 $\overline{ADS}$ low to $\overline{RAS}$ low−#27 data setup−F245 Transceiver tp max.

= 187.5 ns−8 ns−29 ns−7 ns−6 ns

= 137.5 ns (DP8422A-25)

= 187.5 ns−8 ns−35 ns−7 ns−6 ns

= 131.5 ns (DP8422A-20)

tCAC (nonburst access):

3 Tcp−PAL tCLK max.−$403a $\overline{ADS}$ low to $\overline{CAS}$ low−#27 data setup−F245 Transceiver tp max.

= 187.5 ns−8 ns−82 ns−7 ns−6 ns

= 84.5 ns (DP8422A-25)

= 187.5 ns−8 ns−94 ns−7 ns−6 ns

= 72.5 ns (DP8422A-20)

tAA (nonburst access):

3 Tcp−PAL tCLK max.−$417 $\overline{ADS}$ low to Column Address valid−#27 data setup−F245 Transceiver tp max.

= 187.5 ns−8 ns−78 ns−7 ns−6 ns

= 88.5 ns (DP8422A-25)

= 187.5 ns−8 ns−92 ns−7 ns−6 ns

= 74.5 ns (DP8422A-20)

tCAC (burst access):

2.5 Tcp−#9 $\overline{CLK}$ high to $\overline{DS}$ low max.−PAL tp max.−$14 $\overline{ECAS}$ low to $\overline{CAS}$ low max.−#27 data setup−F245 Transceiver tp max.

= 156 ns−40 ns−10 ns−27 ns−7 ns−6 ns

= 66 ns (DP8422A-25)

= 156 ns−40 ns−10 ns−31 ns−7 ns−6 ns

= 62 ns (DP8422A-20)

tAA (burst access):

3 Tcp−#6 $\overline{CLK}$ low to Address valid max.−$26 Address valid to Q max.− #27 data setup−F245 Transceiver tp max.

= 187.5 ns−5 ns−35 ns−7 ns−6 ns

= 89.5 ns (DP8422A-25)

= 187.5 ns−5 ns−38 ns−7 ns−6 ns

= 86.5 ns (DP8422A-20)

68KPAL (PAL20R4D) EQUATIONS

The Boolean entry operators are listed as:

":="	Replaced by (after clock)
"="	Equality
"*"	AND
"+"	OR
"/"	Complement
"~"	Active low

The brief explanation of PAL output signals

CS~	This combinational output signal is Chip Select.
CSD~	This sequential output signal is Chip Select Delayed by one clock.
ADS~	This sequential output signal is Address Strobe (also used as an Access Request, $\overline{AREQ}$, to DP8422A).
READY~	This combinational output signal is Data Ready.
ECASU~	This combinational output signal is to select upper byte.
ECASL~	This combinational output signal is to select lower byte.

Inputs: CLK, A21, A22, FC2, FC1, FC0, UDS~, LSD~, RFRQ~, AS~, DTACK~, HSA~

Outputs: /CS~ = /A21*/A22*/FC2*/FC1*FC0 +
/A21*/A22*/FC2*FC1*/FC0+
/A21*/A22*FC2*/FC1*FC0+
/A21*/A22*FC2*FC1*/FC0

/CSD~ := /CS~

/ADS~ := /HSA~*/CSD~*RFRQ~ +
/AS~*/RFRQ~

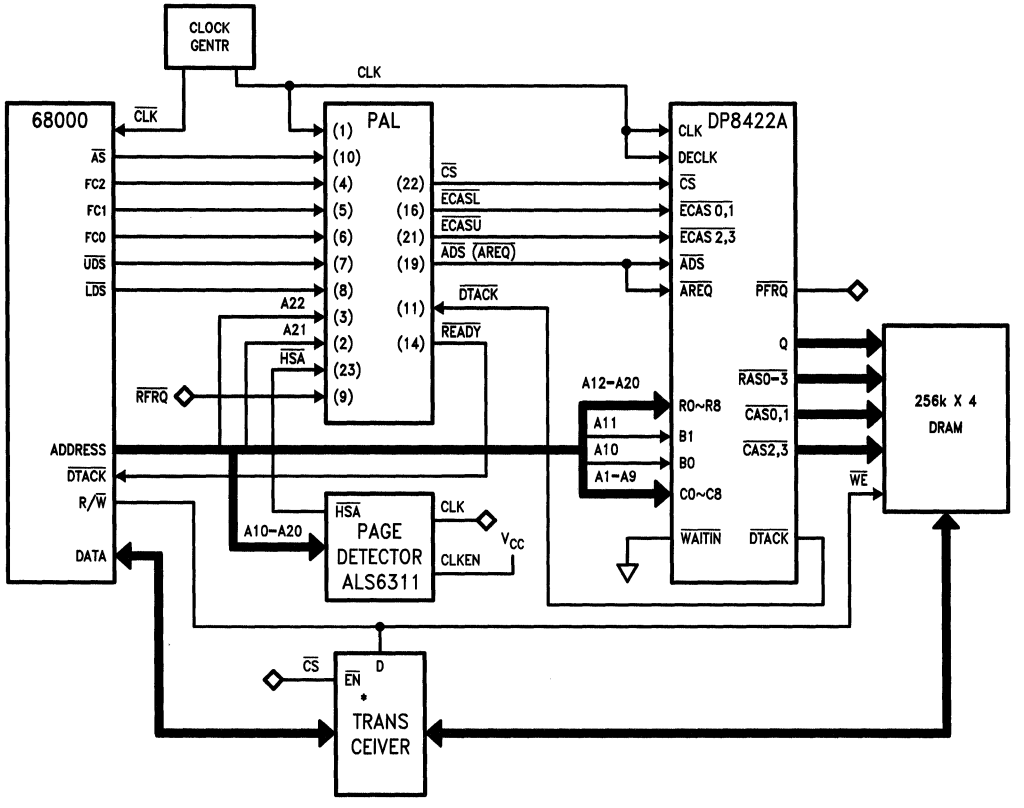
/READY~ = /DTACK~*/AS~

/ECASU~ = /ADS~*/UDS~*/CSD~
*/HSA~

/ECASL~ = /ADS~*/LDS~*/CSD~
*/HSA~

Note: Address inputs such as A21 and A22, are system dependent.

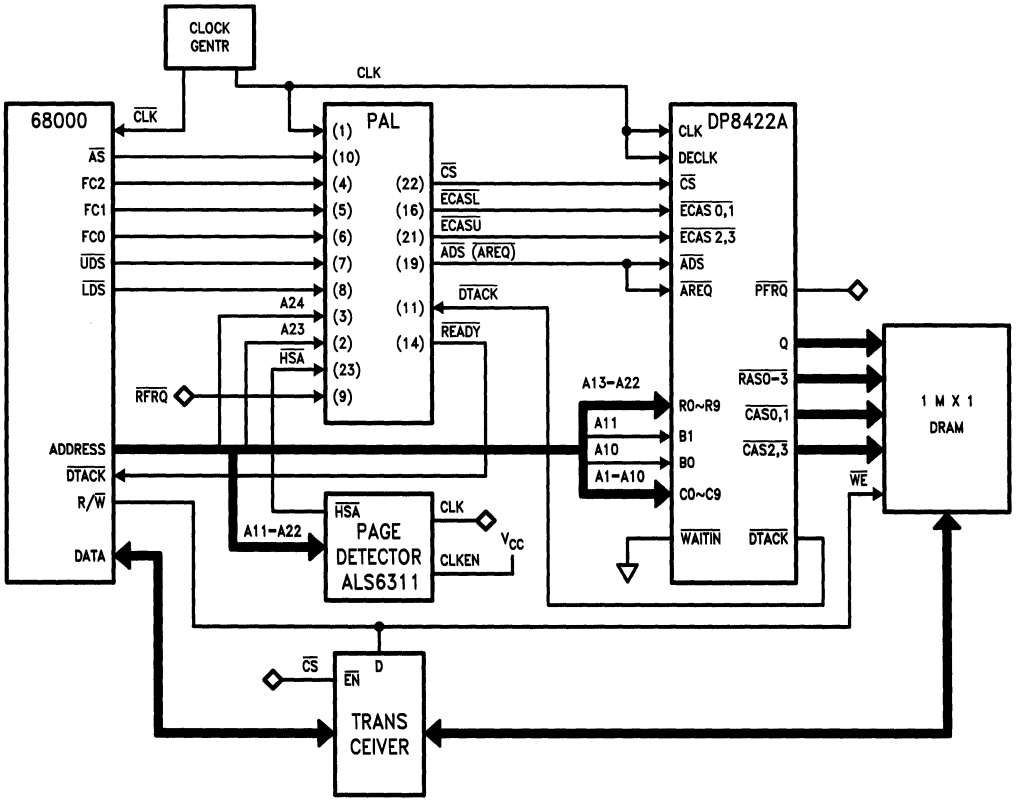
4



TL/F/10438–1

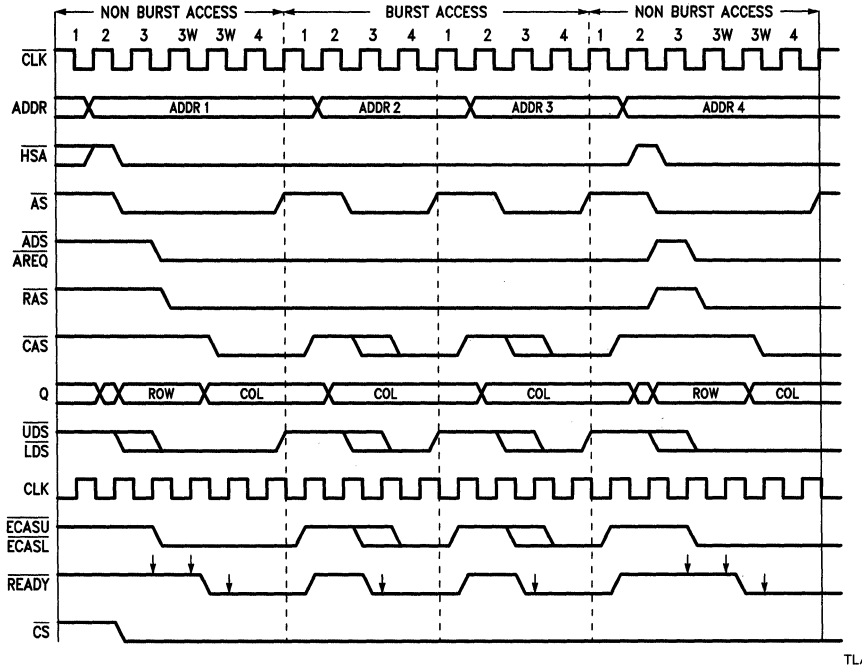
*Transceivers could be eliminated from this design.

FIGURE 1.1 Schematic Diagram of Interfacing DP8422A/68000 Burst Access to 256k x 4 DRAM



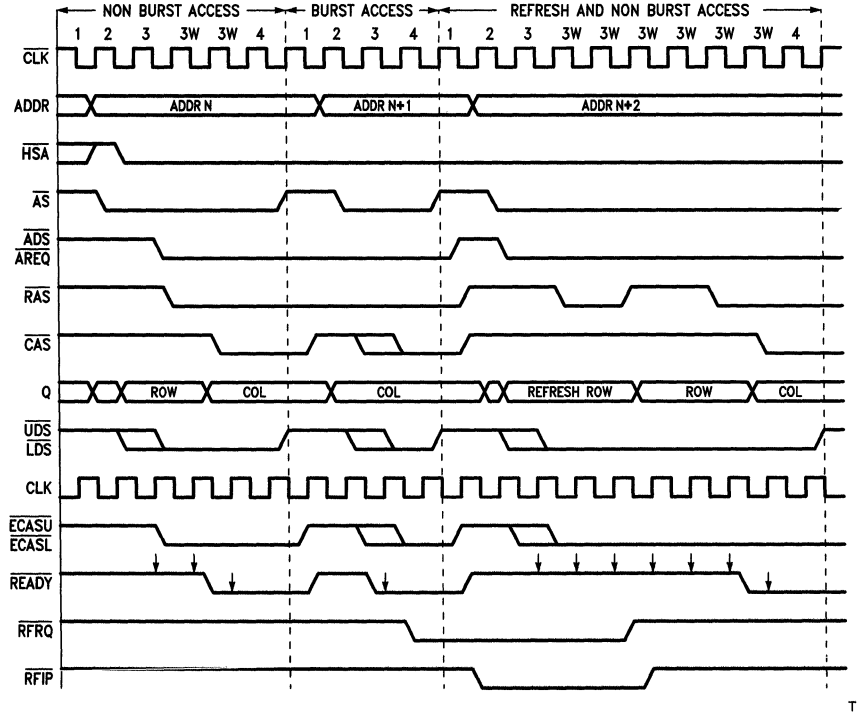
TL/F/10438–2

FIGURE 1.2 Schematic Diagram of Interfacing DP8422A/68000 Burst Access to 1M x 1 DRAM



TL/F/10438–3

FIGURE 2. Timing Diagram of Burst and Nonburst Access



TL/F/10438–4

FIGURE 3. Timing Diagram of Refresh, Burst and Nonburst Access

Interfacing the DP8420A/21A/22A to the 68020

National Semiconductor
Application Note 539
Joe Tate and Rusty Meier

INTRODUCTION

This application note explains interfacing the DP8420A/21A/22A DRAM controller to the 68020 microprocessor. Three different designs are shown and explained. It is assumed that the reader is already familiar with the 68020 access cycles and the DP8420A/21A/22A modes of operation.

DESIGN #1 DESCRIPTION

Design #1 is a simple circuit to interface the 68020 to the DP8420A/21A/22A and up to 64 Mbytes of DRAM. The DP8420A/21A/22A is operated in Mode 1. An access cycle begins when the 68020 places a valid address on the address bus and asserts the address strobe ($\overline{AS}$). Chip select ($\overline{CS}$) is generated by a 74AS138 decoder. If a refresh or Port B access (DP8422A only) is not in progress, the DP8420A/21A/22A will assert the proper $\overline{RAS}$ depending on the bank select inputs (B0, B1). After guaranteeing the programmed value of row address hold time, the DP8420A/21A/22A will switch the DRAM address (Q0–8, 9, 10) to the column address and assert $\overline{CAS}$. By this time, the 74AS245s have been enabled and the DRAMs place their data on the data bus. The DP8420A/21A/22A also asserts $\overline{DTACK}$ which is used to generate $\overline{DSACK}$0,1 to the 68020.

If a refresh had been in progress, the DP8420A/21A/22A would have delayed the 68020's access by inserting wait states into the access cycle until the refresh was complete and the programmed amount of precharge time was met. This circuit can run up to 16 MHz with one wait state. However, the timing parameters become close to the minimums for the DP8420A/21A/22A parameters. $\overline{ADS}$ asserted to CLK high ($400b), $\overline{CS}$ setup to $\overline{ADS}$ asserted ($401) and $\overline{ADS}$ negated held from CLK ($405). Problems can also occur if the loading on the clocks generated from the 74AS74 cause too much skew between CLK and $\overline{CLK}$. The clock must be inverted to guarantee timing parameters. A solution to this problem is to invert the CLOCK to the 68020 with a 74AS04.

Since the 68020 address strobe can end late in the access, a problem with $\overline{RAS}$ precharge can occur in back-to-back accesses. In these accesses, the DP8420A/21A/22A will guarantee the precharge time by inserting wait states. To reduce this problem, memory interleaving should be used by tying the low order address bits to the bank selects.

Timing parameters are referenced to the numbers shown in the DP8420A/21A/22A data sheet. Numbered times starting with a "$" refer to the DP8420A/21A/22A timing parameters. Numbered times starting with "#" refer to the Motorola 68020 data sheet. Equations have been given to allow the user to calculate timing based on his frequency and application. The clock has been chosen at a multiple of 2 MHz only to allow the user to hook the system clock to the PLL delay line clock (DELCLK). If you are running at a frequency that is not a multiple of 2 MHz, it is recommended that you use a clock which is a multiple of 2 MHz for DELCLK. If DELCLK is not a multiple of 2 MHz, $\overline{ADS}$ to $\overline{CAS}$ must be recalculated.

DESIGN #1 TIMING AT 16 MHz AND 12 MHz

Clock Period = Tcp16 = 62.5 ns @ 16 MHz
= Tcp12 = 83 ns @ 12 MHz

$400b: $\overline{ADS}$ Asserted Setup to CLK High
= Clock Period − 68020 Clock to $\overline{AS}$ Low Max
= Tcp16 − #9 Max
= 62.5 ns − 30 ns

= 32 ns @ 16 MHz

= Tcp12 − #9 Max
= 83 ns −40 ns

= 43 ns @ 12 MHz

$401: $\overline{CS}$ Setup to $\overline{ADS}$ Asserted
= 68020 Address to $\overline{AS}$ Maximum − 74AS138 Decoder Maximum
= #11 Max − Tphl Max
= 15 ns − 9 ns

= 6 ns @ 16 MHz

= #11 Max − Tphl Max
= 20 ns − 9 ns

= 11 ns @ 12 MHz

$407 & 404: Address Valid Setup to $\overline{ADS}$ Asserted
= 68020 Address to $\overline{AS}$ Maximum
= #11 Max

= 15 ns @ 16 MHz

= #11 Max

= 20 ns @ 12 MHz

$405: $\overline{ADS}$ Negated Held from CLK High
= 68020 Minimum Clock to $\overline{AS}$
= #9 Min

= 3 ns @ 16 MHz

= #9 Min

= 3 ns @ 12 MHz

#47A: $\overline{DSACK}$0, 1 Setup Time
= ½ Clock Period − Max 74AS74 Delay − Max 74AS32 Delay
= 31 ns − 9 ns − 5 ns

= 17 ns @ 16 MHz

= ½ Tcp12 − Tphl Max − Tphl Max
= 41 ns − 9 ns − 5 ns

= 27 ns @ 12 MHz

#47B: $\overline{DSACK0}$, 1 Hold Time
= ½ Clock Period + Min 74AS74 Delay + Min 74AS32 Delay
= ½ Tcp16 + Tphl Min + Tphl Min
= 31 ns + 5 ns + 1 ns

= 37 ns @ 16 MHz

= ½ Tcp12 + Tphl Min + Tphl Min
= 41 ns − 9 ns − 5 ns

= 47 ns @ 12 MHz

$\overline{RAS}$ Low during REFRESH

tRAS = Programmed Clocks − [(CLK High to Refresh $\overline{RAS}$ Asserted) − (CLK High to Refresh $\overline{RAS}$ Negated)]
= Tcp16 + Tcp16 − $55
= 62.5 ns + 62.5 ns −6 ns

= 119 ns @ 16 MHz

= Tcp12 + Tcp12 − $55
= 83.3 ns + 83.3 ns − 6 ns

= 160 ns @ 12 MHz

$\overline{RAS}$ Precharge Parameters**

$29b: $\overline{AREQ}$ Negated Setup to CLK High
= CLOCK Period − CLOCK Low to 68020 $\overline{AS}$ Negated
= Tcp16 − #12
= 62.5 ns − 30 ns

= 32 ns @ 16 MHz

= Tcp12 − #12
= 83 ns − 40 ns

= 43 ns @ 12 MHz

tRP = s5 + s0 + s1 + s2 − 68020 CLK Low to $\overline{AS}$ Negated − [($\overline{AREQ}$ to $\overline{RAS}$ Negated) − (CLK to $\overline{RAS}$ Asserted)]
= 2 Tcp16 − #12 − $50
= 125 ns − 30 ns − 16 ns

= 79 ns @ 16 MHz w/8420A/21A/22A

= 81 ns @ 16 MHz w/8420A/21A/22A

= 2 Tcp12 − #12 − $50
= 166 ns − 40 ns − 16 ns

= 120 ns @ 12 MHz

****Note:** To gain more precharge program 3T.

tRAC AND tCAC TIMING FOR DRAMs

Timing is supplied for the system shown in *Figure 1 (See Figures 2, 3)*. Since systems and DRAM times vary, the user is encouraged to change the following equations to match his system requirements. Timing has been supplied for systems with 0, 1 or 2 wait states. If the DELCLK is not a multiple of 2 MHz, the timing for tRAH and tASC will increase. Because tRAH and tASC will increase, $\overline{ADS}$ to $\overline{RAS}$ and $\overline{ADS}$ to $\overline{CAS}$ will also increase and must be changed according to the equations given in the data sheet. The $\overline{ADS}$ to $\overline{RAS}$ and $\overline{ADS}$ to $\overline{CAS}$ will also have to be changed depending on the capacitance of the DRAM array.

0 Wait States

tRAC = s1 + s2 + s3 + s4 − 68020 CLK to $\overline{AS}$ Max − 74AS245 Delay Max − 68020 Data Setup Min − $\overline{ADS}$ Asserted to $\overline{RAS}$ Asserted
= Tcp16 + Tcp16 − #9 Max − Tphl Max − #27 Min − $402 Max
= 62.5 ns + 62.5 ns − 30ns − 7 ns − 5 ns −35 ns

= 48 ns @ 16 MHz w/8420A-20 Heavy Load

= 58 ns @ 16 MHz w/8420A-25 Light Load

= Tcp12 + Tcp12 − #9 Max − Tphl Max − #27 Min − $402 Max
= 83 ns + 83 ns − 40 ns − 7 ns − 10 ns − 35 ns

= 74 ns @ 12 MHz w/8420A-20 Heavy Load

= 84 ns @ 12 MHz w/8420A-25 Light Load

1 Wait State

tRAC = s1 + s2 + sw + sw + s3 + s4 − 68020 CLK to $\overline{AS}$ Max − 74AS245 Delay Max − 68020 Data Setup Min − $\overline{ADS}$ Asserted to $\overline{RAS}$ Asserted
= Tcp16 + Tcp16 + Tcp16 − #9 Max − Tphl Max − #27 Min − $402 Max
= 62.5 ns + 62.5 ns + 62.5 ns − 30 ns − 7 ns − 5 ns −35 ns

= 110 ns @ 16 MHz w/8420A-20 Heavy Load

= 120 ns @ 16 MHz w/8420A-25 Light Load

= Tcp12 + Tcp12 + Tcp12 − #9 Max − Tphl Max − #27 Min − $402 Max
= 83 ns + 83 ns + 83 ns −40 ns − 7 ns − 10 ns − 35 ns

= 157 ns @ 12 MHz w/8420A-20 Heavy Load

= 167 ns @ 12 MHz w/8420A-25 Light Load

2 Wait States

tRAC = s1 + s2 + sw + sw + sw + sw + s3 + s4 − 68020 CLK to $\overline{AS}$ Max − 74AS245 Delay Max − 68020 Data Setup Min − $\overline{ADS}$ Asserted to $\overline{RAS}$ Asserted
= Tcp16 + Tcp16 + Tcp16 + Tcp16 − 9 Max − Tphl Max − #27 Min − $402 Max
= 62.5 ns + 62.5 ns + 62.5 ns + 62.5 ns − 30 ns − 7 ns −5 ns −35 ns

= 173 ns @ 16 MHz w/8420A-20 Heavy Load

= 183 ns @ 16 MHz w/8420A-25 Light Load

= Tcp12 + Tcp12 + Tcp12 + Tcp12 −9 Max − Tphl Max − #27 Min − $402 Max
= 83 ns + 83 ns + 83 ns + 83 ns −40 ns −7 ns − 10 ns −35 ns

= 240 ns @ 12 MHz w/8420A-20 Heavy Load

= 250 ns @ 12 MHz w/8420A-25 Light Load

0 Wait State

tCAC = s1 + s2 + s3 + s4 − 68020 CLK to $\overline{AS}$ Max − 74AS245 Delay Max − 68020 Data Setup Min − $\overline{ADS}$ Asserted to $\overline{CAS}$ Asserted
= Tcp16 + Tcp16 − #9 Max − Tphl Max − #27 Min − $403a Max
= 62.5 ns + 62.5 ns − 30 ns − 7 ns −5 ns − 94 ns

= −11 ns @ 16 MHz w/8420A-20 Heavy Load

= 8 ns @ 16 MHz w/8420A-25 Light Load

= Tcp12 + Tcp12 − #9 Max − Tphl Max − #27 Min − $403a Max
= 83 ns + 83 ns −40 ns − 7 ns − 10 ns − 94 ns

= 15 ns @ 12 MHz w/8420A-20 Heavy Load

= 34 ns @ 12 MHz w/8420A-25 Light Load

1 Wait State

tCAC = s1 + s2 + sw + sw + s3 + s4 − 68020 CLK to $\overline{AS}$ Max − 74AS245 Delay Max − 68020 Data Setup Min − $\overline{ADS}$ Asserted to $\overline{CAS}$ Asserted
= Tcp16 + Tcp16 + Tcp16 − #9 Max − Tphl Max − #27 Min − $403a Max
= 62.5 ns + 62.5 ns + 62.5 ns − 30 ns − 7 ns − 5 ns − 94 ns

= 51 ns @ 16 MHz w/8420A-20 Heavy Load

= 70.5 ns @ 16 MHz w/8420A-25 Light Load

= Tcp12 + Tcp12 + Tcp12 − #9 Max − Tphl Max − #27 Min − $403a Max
= 83 ns + 83 ns + 83 ns − 40 ns − 7 ns − 10 ns − 94 ns

= 98 ns @ 12 MHz w/8420A-20 Heavy Load

= 134 ns @ 12 MHz w/8420A-25 Light Load

2 Wait States

tCAC = s1 + s2 + sw + sw + sw + sw + s3 + s4 − 68020 CLK to $\overline{AS}$ Max − 74AS245 Delay Max − 68020 Data Setup Min − $\overline{ADS}$ Asserted to $\overline{CAS}$ Asserted
= Tcp16 + Tcp16 + Tcp16 + Tcp16 − #9 Max − Tphl Max − #27 Min − $403a Max
= 62.5 ns + 62.5 ns + 62.5 ns +62.5 ns − 30 ns − 7 ns − 5 ns − 94 ns

= 114 ns @ 16 MHz w/8420A-20 Heavy Load

= 133 ns @ 16 MHz w/8420A-25 Light Load

= Tcp12 + Tcp12 + Tcp12 + Tcp12 − #9 Max − Tphl Max − #27 Min − $403a Max
= 83 ns + 83 ns + 83 ns + 83 ns − 40 ns − 7 ns − 10 ns − 74 ns

= 181 ns @ 12 MHz w/8420A-20 Heavy Load

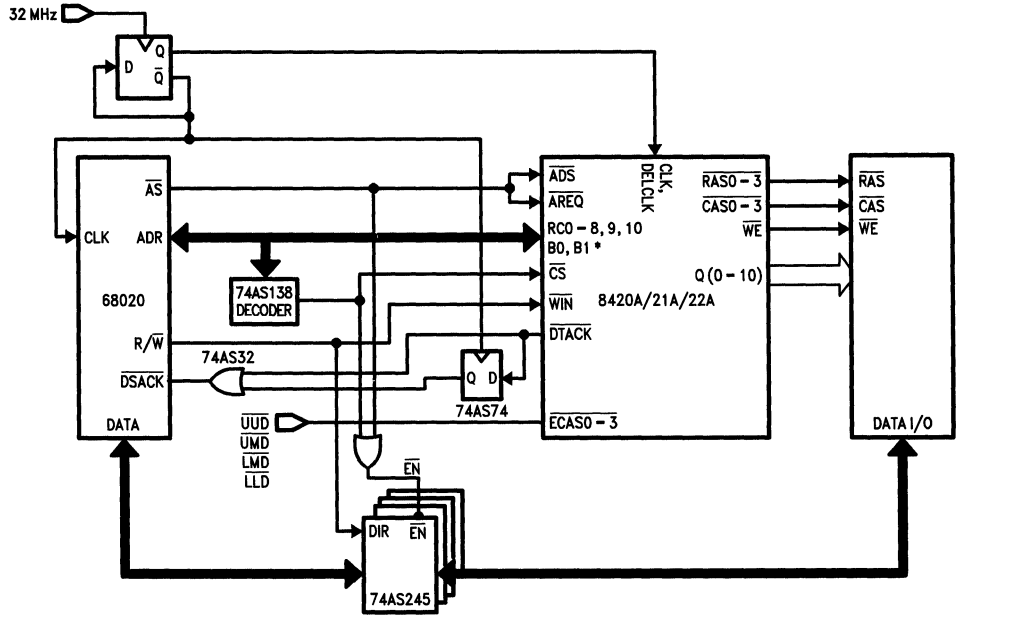
= 200 ns @ 12 MHz w/8420A-25 Light Load

Design #1 Programming Bits

Bits	Description	Value
R0, R1	$\overline{RAS}$ Low during REFRESH = 2T $\overline{RAS}$ Precharge Time = 2T	R0 = 0 R1 = 1
R2, R3	$\overline{DTACK}$ Generation Modes for Non-Burst Accesses (½T after $\overline{RAS}$)	R2 = 0 R3 = 1
R4, R5	$\overline{DTACK}$ during Burst Mode	R4 = x R5 = x
R6	Add Wait States with WAITIN	R6 = x
R7	$\overline{DTACK}$ Mode Selected	R7 = 1
R8	Non-Interleaved Mode	R8 = 1
R9	Staggered or All REFRESH	R9 = u
C0, C1, C2	Divisor for DELCLK	C0 = s C1 = s C2 = s
C3	+30 REFRESH	C3 = 0
C4, C5, C6	$\overline{RAS}$, $\overline{CAS}$ Configuration Mode *Choose An All $\overline{CAS}$ Mode	C4 = u C5 = u C6 = u
C7	Select 0 ns Column Address Setup	C7 = 1
C8	Select 15 ns Row Address Hold	C8 = 1
C9	CAS is Delayed to Next Rising CLOCK during Writes	C9 = 1
B0	The Row/Column Bank Latches Are in Fall Through Mode	B0 = 1
B1	Access Mode 1	B1 = 1
$\overline{ECAS0}$	$\overline{CAS}$ not extended beyond $\overline{RAS}$	$\overline{ECAS0}$ = 0

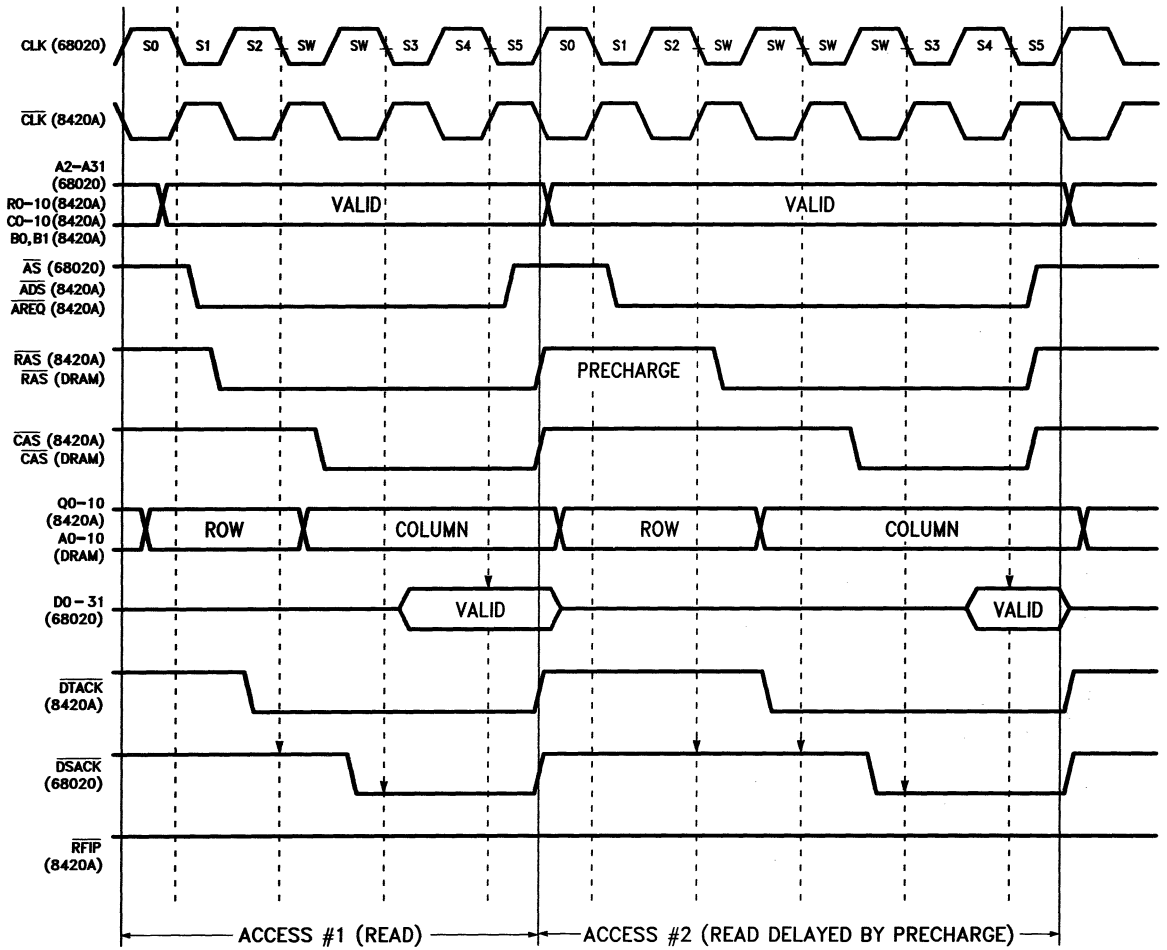
x = don't care
u = user defined
s = system dependent

s @ 16 MHz	s @ 12 MHz
C0 = 0 C1 = 1 C2 = 0	C0 = 0 C1 = 0 C2 = 1



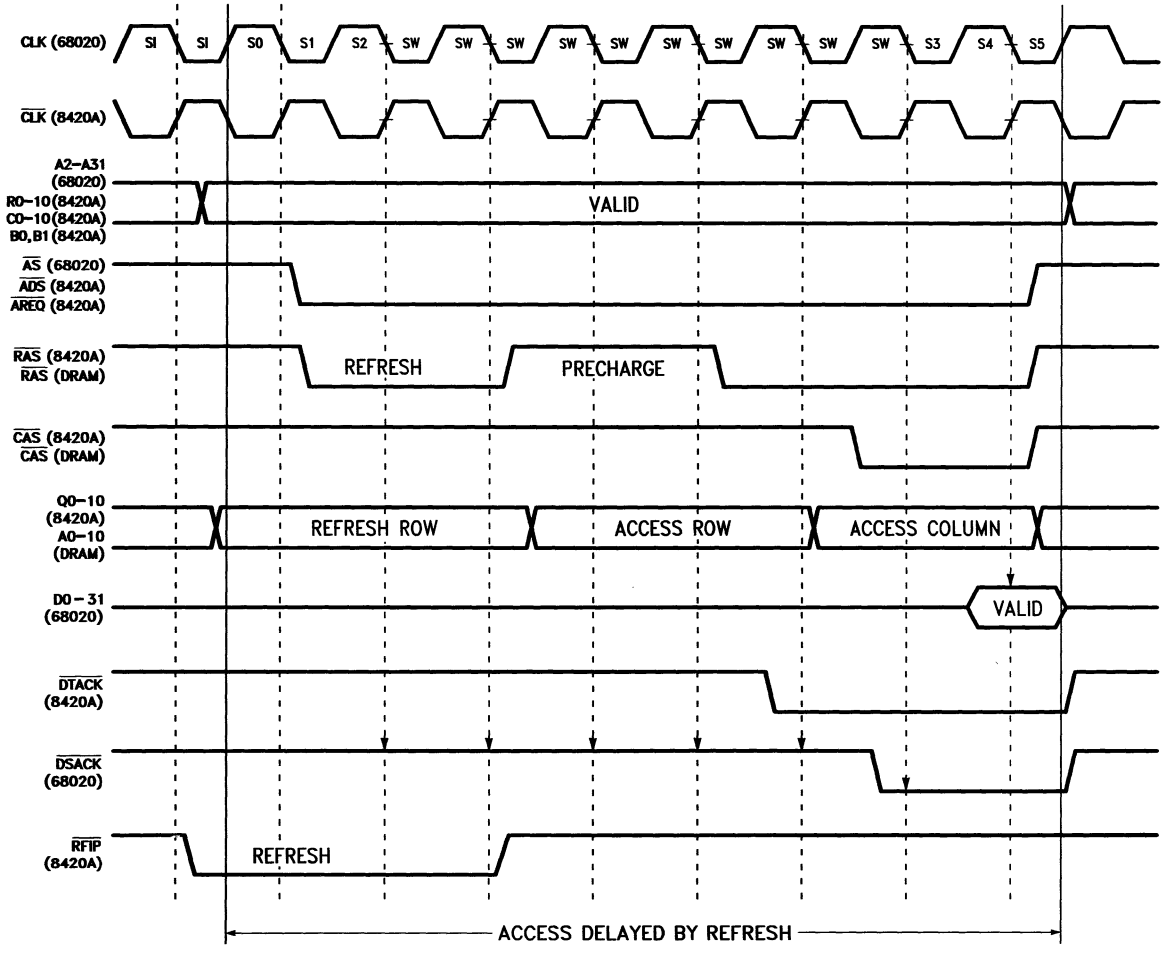
*Tie least significant ADR bits (A2, A3) to B0, B1.

FIGURE 1. Design # 1, 68020 up to 16 MHz with 1 Wait State



TL/F/9733–2

FIGURE 2. Design # 1 Timing



TL/F/9733–3

FIGURE 3. 68020 Design #1 Timing

DESIGN #2 DESCRIPTION

Design #2 is a modification of design #1. This design allows a DRAM array up to 64 Mbytes. However, driving an array with greater capacitance than specified in the data sheet requires derating the $\overline{ADS}$ to $\overline{RAS}$ and $\overline{ADS}$ to $\overline{CAS}$ times. Smaller DRAM arrays can derate times by interpolating times in the DP8420A/21A/22A data sheet timing parameters.

This design differs from design #1 in that a latch was added to produce $\overline{ADS}$ and $\overline{AREQ}$. This latch asserts $\overline{ADS}$ and $\overline{AREQ}$ at the beginning of the 68020 "S2" clock. This latch was added to increase the time from $\overline{ADS}$ asserted to CLK ($400b), $\overline{CS}$ setup to $\overline{ADS}$ asserted ($401) and $\overline{ADS}$ negated held from CLK high ($405). The DP8420/21/22 is operated in Mode 1. An access cycle begins when the 68020 places a valid address on the address bus and asserts $\overline{ADS}$. If the address is in the address space of the DRAM, the 74AS138 decoder asserts $\overline{CS}$. During the next positive clock level, the latch is set which produces $\overline{ADS}$ and $\overline{AREQ}$. If a refresh or Port B access is not in progress, the DP8420A/21A/22A will assert the proper $\overline{RAS}$ depending on programming and the bank select inputs (B0, B1). After guaranteeing the programmed value of row address hold time, the DP8420A/21A/22A will switch the DRAM address (Q0–8, 9, 10) to the column address and assert $\overline{CAS}$. By this time the 74AS245s have been enabled and the DRAMs place their data on the data bus. The DP8420A/21A/22A also asserts $\overline{DTACK}$ which is used to generate $\overline{DSACK}$0,1 to the 68020. When the 68020 negates $\overline{AS}$, the latch is cleared and the access is terminated.

If a refresh or Port B access had been in progress, the DP8420A/21A/22A would have delayed the 68020 access by inserting wait states into the access cycle until the refresh was complete and the programmed amount of precharge was met. This circuit can run up to 20 MHz with 2 wait states. It is suggested that the least significant address bits be tied to the bank select inputs (B0, B1). This will reduce the chance of having to insert wait states to guarantee $\overline{RAS}$ precharge time. To keep the delays as specified in the data sheet, it is recommended that DELCLK is a multiple of 2 MHz.

DESIGN #2 TIMING AT 20 MHz AND 16.667 MHz

Clock Period = Tcp20 = 50 ns @ 20 MHz
= Tcp16 = 60 ns @ 16.667 MHz

$400b: $\overline{ADS}$ Asserted to CLK High
= Clock Period − 74AS04 Maxhl − 74AS02 Maxlh − 74AS02 Maxhl
= Tcp20 − Tphl Max − Tplh Max − Tphl Max
= 50 ns − 4 ns − 4.5 ns − 4.5 ns
= 37 ns @ 20 MHz
= Tcp16 − Tphl Max − Tplh Max − Tphl Max
= 60 ns − 4 ns − 4.5 ns − 4.5 ns
= 47 ns @ 16.667 MHz

$401: $\overline{CS}$ Setup to $\overline{ADS}$ Asserted
= Clock Period + 74AS04 Minhl + 74AS02 Minlh + 74AS02 Minhl 68020 Clock to Address Max − 74AS138 Decoder Maxhl
= Tcp20 + Tphl Min + Tplh Min + Tphl Min − #6 Max − Tphl Max
= 50 ns + 1 ns + 1 ns + 1 ns − 25 ns − 9 ns
= 19 ns @ 20 MHz
= Tcp16 + Tphl Min + Tplh Min + Tphl Min − #6 Max − Tphl Max
= 60 ns + 1 ns + 1 ns − 30 ns − 9 ns
= 24 ns @ 16.667 MHz

$404 & $407: Address Setup to $\overline{ADS}$ Asserted
= Clock Period + 74AS04 Minhl + 74AS02 Minlh + 74AS02 Minhl 68020 Clock to Address Max
= Tcp20 + Tphl Min + Tplh Min + Tphl Min − #6 Max
= 50 ns + 1 ns + 1 ns + 1 ns − 25 ns
= 28 ns @ 20 MHz
= Tcp16 + Tphl Min + Tplh Min + Tphl Min − #6 Max
= 60 ns + 1 ns + 1 ns + 1 ns − 30 ns
= 33 ns @ 16.667 MHz

$405: $\overline{ADS}$ Negated Held from CLK High
= 74AS04 Minhl + 74AS02 Minlh + 74AS02 Minhl
= Tphl Min + Tplh Min + Tphl Min
= 1 ns + 1 ns + 1 ns
= 3 ns @ 20 MHz
= Tphl Min + Tplh Min + Tphl Min
= 1 ns + 1 ns + 1 ns
= 3 ns @ 16.667 MHz

#47A: $\overline{DSACK}$0,1 Setup Time
= ½ Clock Period − Max 74AS74 Delay − Max 74AS32 Delay ½ Tcp20 − Tphl Max − Tphl Max
= 25 ns − 9 ns − 5 ns
= 11 ns @ 20 MHz
= ½ Tcp16 − Tphl Max − Tphl Max
= 30 ns − 9 ns − 5 ns
= 16 ns @ 16.667 MHz

#47B: $\overline{DSACK}$0,1 Hold Time
= ½ Clock Period + Min 74AS74 Delay + Min 74AS32 Delay
= ½ Tcp20 + Tphl Min + Tphl Min
= 25 ns + 5 ns + 1 ns
= 31 ns @ 20 MHz
= 30 ns − 5 ns − 1 ns
= 36 ns @ 16.667 MHz

$\overline{RAS}$ Low during REFRESH

tRAS = Programmed Clocks
− [(CLK High to Refresh $\overline{RAS}$ Asserted)
− (CLK High to Refresh $\overline{RAS}$ Negated)]
= Tcp20 + Tcp20 + Tcp20
+ Tcp20 − $55
= 200 ns −6 ns

= 194 ns @ 20 MHz

= Tcp16 + Tcp16 − $55
= 120 ns − 6 ns

= 114 ns @ 16.667 MHz

$\overline{RAS}$ Precharge Parameters

$29b: $\overline{AREQ}$ Negated Setup to CLK High
= CLOCK Period − Max 74AS04 − Max 74AS02 − Max 74AS02 − Max 74AS02
= 50 ns − 5 ns − 4.5 ns − 4.5 ns − 4.5 ns

= 31.5 ns @ 20 MHz

=60 ns − 5 ns − 4.5 ns − 4.5 ns − 4.5 ns

= 41.5 ns @ 16.667 MHz

tRP = Programmed Clocks − Max 74AS04 − Max 74AS02 − Max 74AS02 [($\overline{AREQ}$ to $\overline{RAS}$ Negated) − (CLK to $\overline{RAS}$ Asserted)]
= 3 Tcp20 − Tphl − Tplh − Tphl − Tplh − $50
= 150 ns − 4 ns − 4.5 ns − 4.5 ns − 16 ns

= 116.5 ns @ 20 MHz

= 2 Tcp16 − Tphl − Tplh − Tphl − Tplh − $50
= 120 ns − 4 ns − 4.5 ns − 4.5 ns − 4.5 ns − $50

= 86.5 ns @ 16.667 MHz

*To gain more precharge @ 16.667 MHz program 3T precharge.

tRAC AND tCAC TIMING FOR DRAMs

Timing is supplied for the system shown in *Figure 4 (See Figures 5, 6)*. Since systems and DRAM times vary, the user is encouraged to change the following equations to match his system requirements. Timing has been supplied for systems with 1, 2 or 3 wait states. If the DELCLK is not a multiple of 2 MHz, the timing for tRAH and tASC will change. Because tRAH and tASC will change, $\overline{ADS}$ to $\overline{RAS}$ and $\overline{ADS}$ to $\overline{CAS}$ will also vary and must be changed according to the equations given in the data sheet. The $\overline{ADS}$ to $\overline{RAS}$ and $\overline{ADS}$ to $\overline{CAS}$ will also have to be changed depending on the capacitance of the DRAM array.

1 Wait State

tRAC = s2 + s3 + sw + sw + s4
− 74AS04 Maxhl − 74AS02 Maxlh
− 74AS02 Maxhl − 74AS245 Delay Max − 68020 Data Setup Min
− $\overline{ADS}$ Asserted to $\overline{RAS}$ Asserted
= ½ Tcp20 + Tcp20 + Tcp20 − Tphl Max − Tplh Max − Tphl Max
− #27 Min − $402 Max
= 25 ns + 50 ns + 50 ns − 4 ns − 4.5 ns − 4.5 ns − 7 ns − 5 ns − 35 ns

= 65 ns @ 20 MHz w/8420A-20 Heavy Load

= 75 ns @ 20 MHz w/8420A-25 Light Load

= ½ Tcp16 + Tcp16 + Tcp16 − Tphl Max − Tplh Max − Tphl Max − #27 Min − $402 Max
= 30 ns + 60 ns + 60 ns −4 ns − 4.5 ns − 4.5 ns − 7 ns − 5 ns −35 ns

= 85 ns @ 16.667 MHz w/8420A-20 Heavy Load

= 95 ns @ 16.667 MHz w/8420A-25 Light Load

2 Wait States

tRAC = s2 + sw + sw + sw + sw + s3 + s4
− 74AS04 Maxhl − 74AS02 Maxlh
− 74AS02 Maxhl − 74AS245 Delay Max
− 68020 Data Setup Min
− $\overline{ADS}$ Asserted to $\overline{RAS}$ Asserted
= ½ Tcp20 + Tcp20 + Tcp20 + Tcp20 − Tphl Max − Tplh Max − Tphl Max − #27 Min − $402 Max
= 25 ns + 50 ns + 50 ns + 50 ns − 4 ns − 4.5 ns − 4.5 ns − 7 ns − 5 ns − 35 ns

= 115 ns @ 20 MHz w/8420 − 20 Heavy Load

= 125 ns @ 20 MHz w/8420 − 25 Light Load

= ½ Tcp16 + Tcp16 + Tcp16 + Tcp16 − Tphl Max − Tplh Max − Tphl Max − #27 Min − $402 Max
= 30 ns + 60 ns + 60 ns + 60 ns − 4 ns − 4.5 ns − 4.5 ns − 7 ns − 5 ns − 35 ns

= 150 ns @ 16.667 MHz w/8420 − 20 Heavy Load

= 160 ns @ 16.667 MHz w/8420 − 25 Light Load

3 Wait States

tRAC = s2 + sw + sw + sw + sw + sw
+ s3 + s4 + sw − 74AS04 Maxhl
− 74AS02 Max − 74AS02 Maxhl
− 74AS245 Max Delay − 68020
Data Setup Min − $\overline{ADS}$
Asserted to $\overline{RAS}$ Asserted

= ½ Tcp20 + Tcp20 + Tcp20 + Tcp20
+ Tcp20 − Tphl Max − Tplh Max
− Tphl Max − #27 Min − $402 Max

= 25 ns + 50 ns + 50 ns + 50 ns
− 50 ns − 4 ns − 4.5 ns − 4.5 ns
− 7 ns − 5 ns − 35 ns

= 165 ns @ 20 MHz w/8420 − 20 Heavy Load

= 175 ns @ 20 MHz w/8420 − 25 Light Load

= ½ Tcp16 + Tcp16 + Tcp16 + Tcp16
+ Tcp16 − Tphl Max − Tplh Max
− Tphl Max − #27 Min − $402 Max

= 30 ns + 60 ns + 60 ns + 60 ns − 60 ns
− 4 ns − 4.5 ns − 4.5 ns
− 7 ns − 5 ns − 35 ns

= 210 ns @ 16.667 MHz w/8420 − 20 Heavy Load

= 220 ns @ 16.667 MHz w/8420 − 25 Light Load

1 Wait State

tCAC = s2 + s3 + sw + sw + s4
− 74AS04 Maxhl − 74AS04 Maxlh
− 74AS02 Maxhl − 74AS245 Delay
Max − 68020 Data Setup Min
− $\overline{ADS}$ Asserted to $\overline{CAS}$ Asserted

= ½ Tcp20 + Tcp20 + Tcp20 − Tphl Max
− Tplh Max − Tphl Max
− #27 Min − $403a Max

= 25 ns + 50 ns + 50 ns − 4 ns − 4.5 ns
− 4.5 ns − 7 ns − 5 ns − 94 ns

= 6 ns @ 20 MHz w/8420 − 20 Heavy Load

= 35 ns @ 20 MHz w/8420 − 25 Light Load

= ½ Tcp16 + Tcp16 + Tcp16 − Tphl Max
− Tplh Max − Tphl Max −
#27 Min − $403a Max

= 30 ns + 60 ns + 60 ns − 4 ns − 4.5 ns
− 4.5 ns − 7 ns − 5 ns − 94 ns

= 26 ns @ 16.667 MHz w/8420 − 20 Heavy Load

= 55 ns @ 16.667 MHz w/8420 − 25 Light Load

2 Wait States

tCAC = s2 + sw + sw + sw + sw + s3 + s4
+ 74AS04 Maxhl − 74AS04 Maxlh
− 74AS02 Maxhl − 74AS245 Delay Max
− 68020 Data Setup Min
− $\overline{ADS}$ Asserted to $\overline{CAS}$ Asserted

= ½ Tcp20 + Tcp20 + Tcp20 + Tcp20
− Tphl Max − Tplh Max − Tphl Max
− #27 Min − $403a Max

= 25 ns + 50 ns + 50 ns + 50 ns
− 4 ns − 4.5 ns − 4.5 ns
− 7 ns − 5 ns − 94 ns

= 56 ns @ 20 MHz w/8420 − 20 Heavy Load

= 85 ns @ 20 MHz w/8420 − 25 Light Load

= ½ Tcp16 + Tcp16 + Tcp16 + Tcp16
− Tphl Max − Tplh Max − Tphl Max
− #27 Min − $403a Max

= 30 ns + 60 ns + 60 ns + 60 ns
− 4 ns − 4.5 ns − 4.5 ns
− 7 ns − 5 ns − 94 ns

= 91 ns @ 16.667 MHz w/8420 − 20 Heavy Load

= 120 ns @ 16.667 MHz w/8420 − 25 Light Load

3 Wait States

tCAC = s2 + sw + sw + sw + sw + sw
+ s3 + s4 + sw − 74AS04 Maxhl
− 74AS02 Max − 74AS02 Maxhl
− 74AS245 Max Delay − 68020
Data Setup Min − $\overline{ADS}$
Asserted to $\overline{CAS}$ Asserted

= ½ Tcp20 + Tcp20 + Tcp20 + Tcp20
+ Tcp20 − Tphl Max − Tplh Max
− Tphl Max − #27 Min − $403a Max

= 25 ns + 50 ns + 50 ns + 50 ns
− 50 ns − 4 ns − 4.5 ns − 4.5 ns
− 7 ns − 5 ns − 94 ns

= 106 ns @ 20 MHz w/8420 − 20 Heavy Load

= 135 ns @ 20 MHz w/8420 − 25 Light Load

= ½ Tcp16 + Tcp16 + Tcp16 + Tcp16
+ Tcp16 − Tphl Max − Tplh Max
− Tphl Max − #27 Min − $403a Max

= 30 ns + 60 ns + 60 ns + 60 ns − 60 ns
− 4 ns − 4.5 ns − 4.5 ns
− 7 ns − 5 ns − 94 ns

= 151 ns @ 16.667 MHz w/8420 − 20 Heavy Load

= 180 ns @ 16.667 MHz w/8420 − 25 Light Load

Programming Bits for Design #2

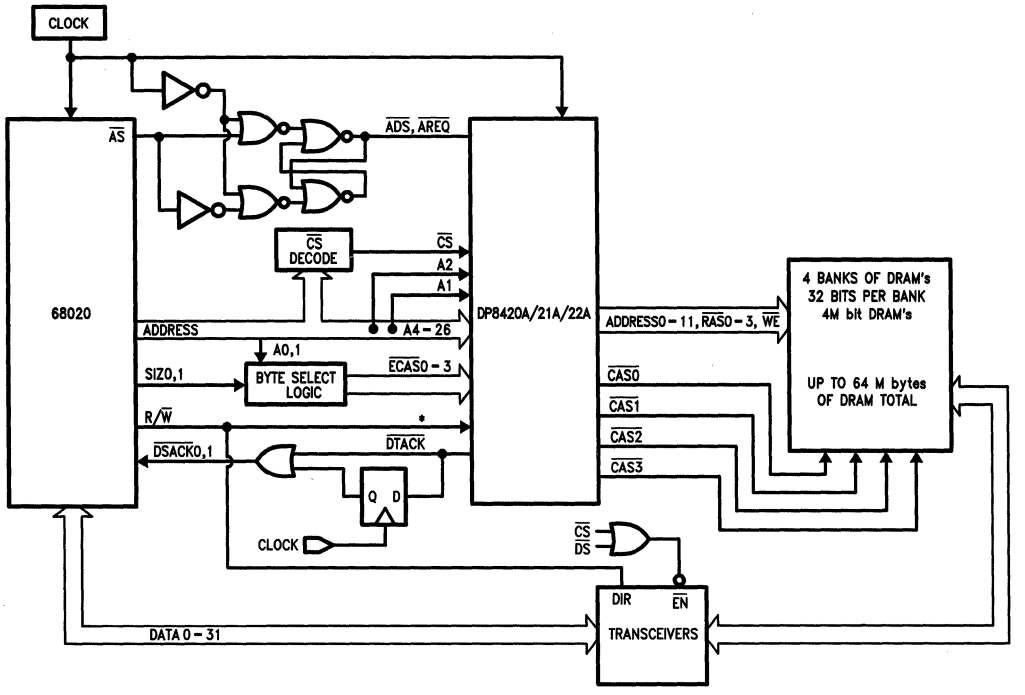
Bits	Description	Value
R0, R1	$\overline{RAS}$ Low during REFRESH $\overline{RAS}$ Precharge Time	R0 = s R1 = s
R2, R3	$\overline{DTACK}$ Generation Modes for Non-Burst (1T after $\overline{RAS}$)	R2 = 1 R3 = 0
R4, R5	$\overline{DTACK}$ during Burst Mode	R4 = x R5 = x
R6	Add Wait States with WAITIN	R6 = 0
R7	$\overline{DTACK}$ Mode Selected	R7 = 1
R8	Non-Interleaved Mode	R8 = 1
R9	Staggered or All REFRESH	R9 = u
C0, C1, C2	Divisor for DELCLK	C0 = s C1 = s C2 = s
C3	+30 Fine Tune	C3 = s
C4, C5, C6	$\overline{RAS}$, $\overline{CAS}$ Configuration Mode *Choose an All CAS Mode	C4 = u C5 = u C6 = u
C7	Select 0 ns Column Address Setup	C7 = 1
C8	Select 15 ns Row Address Hold	C8 = 1
C9	CAS is Delayed to Next Rising CLOCK Edge during Writes	C9 = 1
B0	The Row/Column Bank Latches are in Fall Through Mode	B0 = 1
B1	Access Mode 1	B1 = 1
$\overline{ECAS0}$	$\overline{CAS}$ not extended beyond $\overline{RAS}$	$\overline{ECAS0}$ = 0

x = don't care

u = user defined

s = system dependent

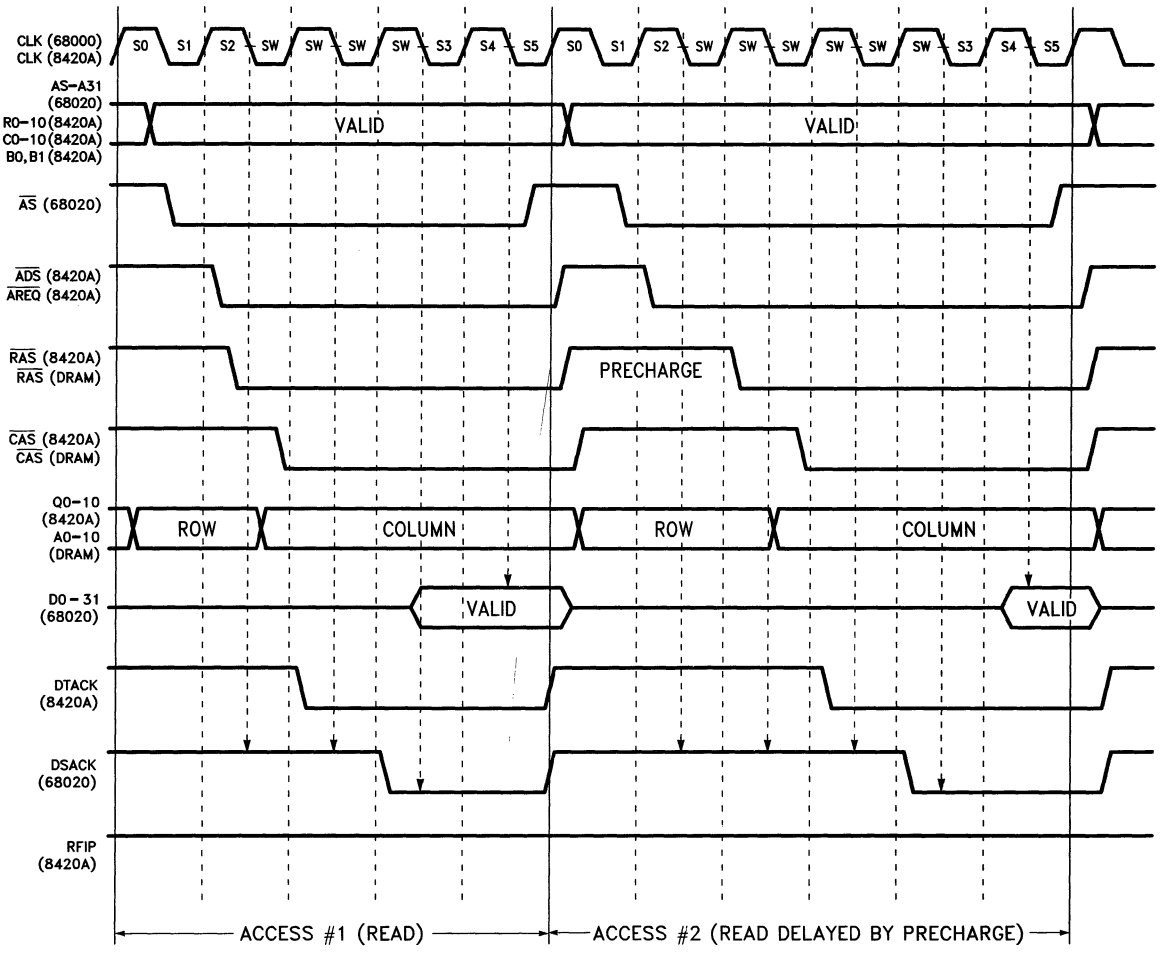
s @ 16.667 MHz		s @ 20 MHz	
R0 = 0	C0 = 0	R0 = 1	C0 = 0
R1 = 1	C1 = 1	R1 = 1	C1 = 0
	C2 = 0		C2 = 0
	C3 = 0		C3 = 0



TL/F/9733–4

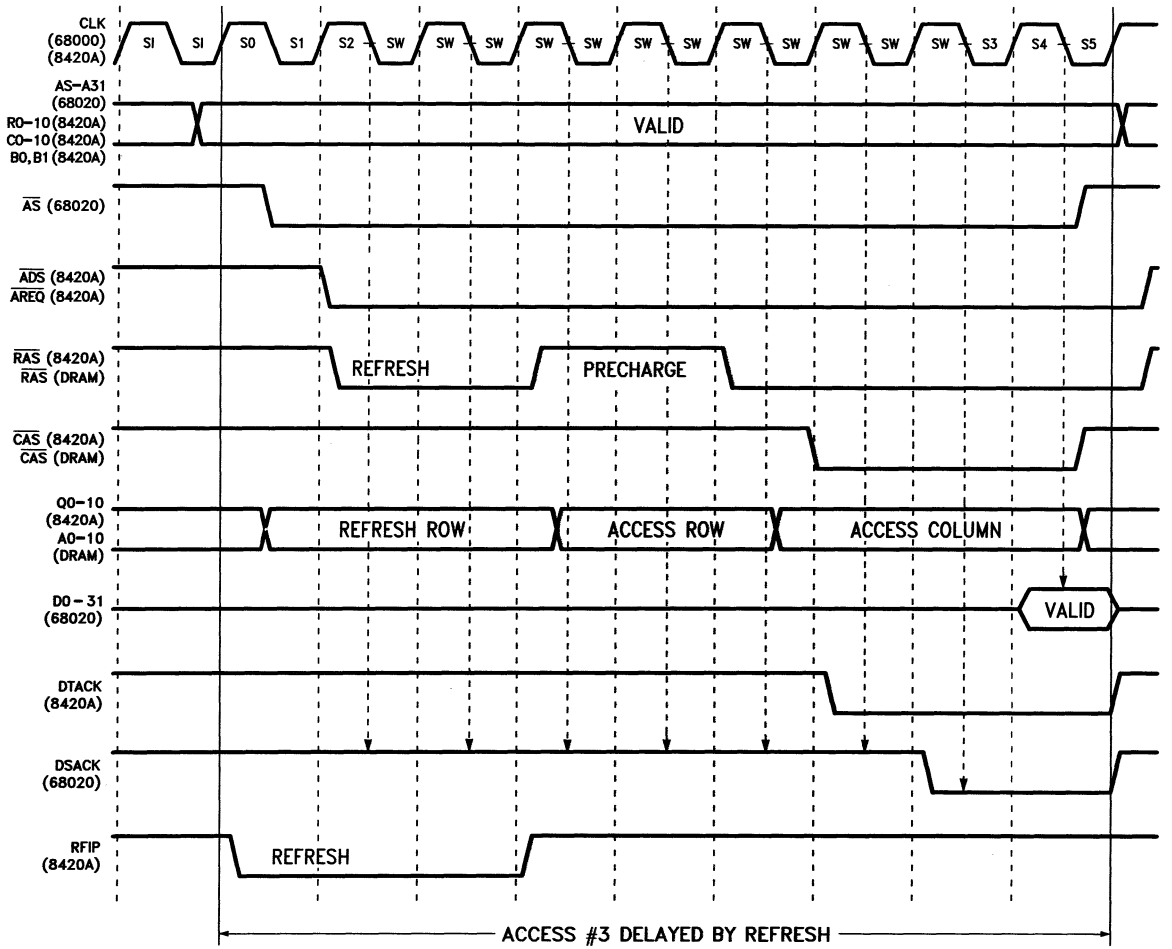
*For higher clock frequencies, the $\overline{ADS}$ input and the R/$\overline{W}$ line from the 68020 should be connected logically by an OR gate for input to the $\overline{WIN}$ input on the DP8420A/21A/22A to avoid a late write during a read access.

FIGURE 4. 68020 Design #2



TL/F/9733–5

FIGURE 5. 68020 Design #2 Timing



TL/F/9733–6

FIGURE 6. 68020 Design #2 Timing

DESIGN #3 DESCRIPTION

Design #3 uses two 68020s sharing a common DRAM array. Port A's interface is the same as design #1.

The two processors share the same CLK. By using the same CLK, the request from Port B do not have to be synchronized to the system CLK.

In this design, an access begins from Port A when the 68020 asserts $\overline{AS}$. Assuming the DP8422 has granted access to Port A through GRANTB negated, $\overline{AS}$ will assert $\overline{RAS}$. After guaranteeing the programmed value of tRAH, the DP8422 will switch the Q outputs to the column address tASC before asserting $\overline{CAS}$. By this time the 74AS245s have been enabled and the DRAM places its data on the data bus. The cycle is terminated by the DP8422 asserting $\overline{DTACK}$. The 68020 will then sample the data from the data bus and negate $\overline{AREQ}$. $\overline{AREQ}$ negated will cause $\overline{RAS}$ to be negated.

If at any time during Port A's access Port B had requested an access by asserting $\overline{AREQB}$, Port B's 68020 would have been delayed by keeping $\overline{ATACKB}$ negated. This would have inserted wait states into Port B's access. After Port A's access terminates, GRANTB is asserted to allow Port B's address through the mux. On the next rising CLOCK edge, $\overline{RAS}$ will be asserted. Again, after guaranteeing the necessary address parameters, $\overline{CAS}$ will be asserted.

Refresh will happen after the current access is completed and precharge time has been met. During this refresh, all accesses will be held off.

Since back-to-back accesses can cause precharge delays, it is recommended that the low order address bits be tied to the bank select inputs.

DESIGN #3 TIMING DESCRIPTION

Clock Period = Tcp16 = 65 ns @ 16 MHz
= Tcp12 = 83 ns @ 12 MHz

Port A Timing

$400b: $\overline{ADS}$ Asserted Setup to CLK High
= Clock Period − 68020 Clock to $\overline{AS}$ Low Max
= Tcp16 − #9 Max
= 62.5 ns − 30 ns

= 32.5 ns @ 16 MHz

= Tcp12 − #9 Max
= 83 ns − 40 ns

= 43 ns @ 12 MHz

$401: $\overline{CS}$ Setup to $\overline{ADS}$ Asserted
= 68020 $\overline{AS}$ Address to $\overline{AS}$ Max + 74AS244 Min − 74AS138 Decoder Max
= #11 Max + Tphl Min + Tphl Max
= 15 ns + 2 ns − 9 ns

= 8 ns @ 16 MHz

= #11 Max + Tphl Min + Tphl Max
= 20 ns + 2 ns − 9 ns

= 13 ns @ 12 MHz

$404 & 407: Address Valid Setup to $\overline{ADS}$ Asserted
= 68020 Address to $\overline{AS}$ + Min 74AS244 − Max 74AS244
= #11 Max + Tphl Min + Tphl Max
= 15 ns + 2 ns − 6 ns

= 11 ns @ 16 MHz

= #11 Max + Tphl Min − Tphl Max
= 20 ns + 2 ns − 6 ns

= 16 ns @ 12 MHz

$405: $\overline{ADS}$ Negated Held from CLK
= 68020 Min CLOCK to $\overline{AS}$
= #9 Min
= 3 ns

= 3 ns @ 16 MHz

= #9 Min + Tphl Min
= 3 ns + 2 ns

= 3 ns @ 12 MHz

#47A: $\overline{DSACK}$0,1 Setup Time
= ½ CLOCK Period − 74AS74 Delay Max − 74AS32 Delay Max
= ½ Tcp16 − Tphl Max − Tphl Max
= 31 ns − 9 ns − 5 ns

= 18 ns @ 16 MHz

* Requires External Flip-Flop and OR Gate.

= ½ CLOCK Period − CLK to $\overline{DTACK}$ Asserted
= ½ Tcp − $18 Max
= 41 ns − 33 ns

= 8 ns @ 12 MHz

*Program as DTACK of 1½ No External Flip-Flop Required.

#47B: $\overline{DSACK}$0,1 Hold Time
= ½ Clock Period + Min 74AS74 + Min 74AS32
= ½ Tcp16 + Tphl Min + Tphl Min
= 31 ns + 5 ns + 1 ns

= 37 ns @ 16 MHz

*Requires External Flip Flop and OR Gate.

= ½ CLOCK Period + Min CLK to $\overline{DTACK}$ Asserted
= 41 ns + 0 ns

= 41 ns @ 12 MHz

*Program as $\overline{DTACK}$ of 1½.

#450 & $454: Address Setup to CLK High
= CLOCK Period − CLK High to GRANTB Negated − 74AS04 Delay Max − 74AS00 Delay Max − 74AS00 Delay Max − 74AS244 Delay
= Tcp16 − $109 Max − Tplh − Tphl − Tphl − Tzh
= 62.5 ns − 26 ns − 5 ns − 4.5 ns − 4.5 ns − 9 ns

= 13.5 ns @ 16 MHz

= tcp12 − $109 Max − Tplh − Tphl − Tphl − Tzh
= 83 ns − 32 ns − 5 ns − 4.5 ns − 4.5 ns − 9 ns

= 34 ns @ 12 MHz

Port B Timing

$100: $\overline{\text{AREQB}}$ Held Negated from CLK High
= CLK to $\overline{\text{AS}}$ Min + 74AS32 Min
= #9 Min + Tphl Min
= 3 ns + 2 ns

= 5 ns @ 16 MHz

= #9 Min + Tphl Min
= 3 ns + 2 ns

= 5 ns @ 12 MHz

$101: $\overline{\text{AREQB}}$ Asserted Setup to CLK High
= Clock Period − CLK to $\overline{\text{AS}}$ Max − 74AS32 Max
= Tcp16 − #9 Max − Tphl Max
= 62.5 ns − 30 ns − 6 ns

= 26.5 ns @ 16 MHz

= Tcp12 − #9 Max − Tphl Max
= 83 ns − 40 ns − 6 ns

= 37 ns @ 12 MHz

$110: Row Address Setup to CLK High
= CLOCK Period − CLK High to GRANTB Asserted − 74AS04 Delay Max − 74AS00 Delay Max − 74AS244 Delay
= Tcp16 − $108 − Tplh − Tphl − Tzh
= 60 ns − 30 ns − 5 ns − 4.5 ns − 9 ns

= 11.5 ns @ 16 MHz

= Tcp12 − $108 − Tplh − Tphl − Tzh
= 80 ns − 30 ns − 5 ns − 4.5 ns − 9 ns

= 31.5 ns @ 20 MHz

$114: Address Valid Setup to $\overline{\text{AREQB}}$ Asserted
= Min Address to $\overline{\text{AS}}$ − 74AS244 Max + 74AS244 Min + 74AS32 Min
= #11 − Tphl Max + Tphl Min + Tphl Min
= 15 ns − 6 ns + 2 ns + 2 ns

= 13 ns @ 16 MHz

= #11 − Tphl Max + Tphl Min + Tphl Min
= 20 ns − 6 ns + 2 ns + 2 ns

= 18 ns @ 12 MHz

$117: $\overline{\text{AREQ}}$ Negated Pulse Width
= $\overline{\text{AS}}$ Negated Width
= #15 Min

= 40 ns @ 16 MHz

= #15 Min

= 50 ns @ 12 MHz

#47a $\overline{\text{DSACK}}$0,1 Setup Time
= ½ CLOCK Period − Max 74AS74 − Max 74AS32
= ½ Tcp16 + Min Tphl + Min Tphl
= 31 ns − 9 ns − 5 ns

= 17 ns @ 16 MHz

= ½ Tcp12 − Tphl Max − Tphl Max
= 41 ns − 9 ns − 5 ns

= 27 ns @ 12 MHz

#47B $\overline{\text{DSACK}}$0,1 Hold Time
= ½ CLOCK Period + Min 74AS74 + Min 74AS32
= ½ Tcp16 + Min Tphl + Min Tphl
= 31 ns + 5 ns + 1 ns

= 37 ns @ 16 MHz

= ½ Tcp12 + Min Tphl + Min Tphl
= 41 ns + 5 ns + 1 ns

= 47 ns @ 12 MHz

$\overline{\text{RAS}}$ Low during REFRESH

tRAS = Programmed CLOCKs − [(CLK High to Refresh $\overline{\text{RAS}}$ Asserted) − (CLK High to Refresh $\overline{\text{RAS}}$ Negated)]
= Tcp16 + Tcp16 − $55
= 62.5 ns + 62.5 ns −6 ns

= 119 ns @ 16 MHz

= Tcp12 + Tcp12 − $55
= 83.3 ns +83.3 ns − 6 ns

= 160 ns @ 12 MHz

$\overline{\text{RAS}}$ Precharge Parameters

$29b: $\overline{\text{AREQ}}$ Negated Setup to CLK High
= CLOCK Period − CLOCK Low to $\overline{\text{AS}}$ Negated
= 62.5 ns − 30 ns

= 32 ns @ 16 MHz

= Tcp12 − #12
= 83 ns − 40 ns

= 43 ns @ 12 MHz

tRP = ss + s0 + s1 + s2 − 68020 CLK Low to $\overline{AS}$ Asserted − [($\overline{AREQ}$ to $\overline{RAS}$ Negated) − (CLK to $\overline{RAS}$ Asserted)]

= 2 Tcp16 − # 12 − $50

= 125 ns − 30 ns − 16 ns

= 79 ns @ 16 MHz w/8420A-20

= 81 ns @ 16 MHz w/8420A-25

= 2 Tcp12 − #12 − $50

= 166 ns − 30 ns − 16 ns

= 120 ns @ 12 MHz

tRAC AND tCAC TIMING FOR DRAMs

Timing is supplied for the system shown in *Figure 7 (See Figure 8)*. Since systems and DRAM times vary, the user is encouraged to change the following equations to match his system requirements. Timing has been supplied for systems with 1 or 2 wait states. If DELCLK is not a multiple of 2 MHz, the timing for tRAH and tASC will change and must be calculated and changed according to the equations in the DP8420A/21A/22A data sheet.

Port A

1 Wait State

tRAC = s1 + s2 + sw + sw + s3 + s4 − 68020 CLK to $\overline{AS}$ Max − 74AS244 Delay Max − 74AS245 Delay Max − 68020 Data Setup Min − $\overline{ADS}$ Asserted to $\overline{RAS}$ Asserted

= Tcp16 + Tcp16 + Tcp16 − #9 Max − Tphl − Tphl − #27 Min − $402 Max

= 62.5 ns + 62.5 ns + 62.5 ns − 30 ns − 6.2 ns − 7 ns − 5 ns − 29 ns

= 110 ns @ 16 MHz w/8420A-25 Heavy Load

= Tcp12 + Tcp12 + Tcp12

= #9 Max − Tphl − Tphl − #27 Min − $402 Max

= 83.3 ns + 83.3 ns + 83.3 ns − 40 ns − 6.2 ns − 7 ns − 10 ns −35 ns

= 163 ns @ 12 MHz w/8420A-20 Heavy Load

2 Wait States

tRAC = s1 + s2 + sw + sw + s3 + s4 − 68020 CLK to $\overline{AS}$ Max − 74AS244 Delay Max − 74AS245 Delay Max − 68020 Data Setup Min − $\overline{ADS}$ Asserted to $\overline{RAS}$ Asserted

= Tcp16 + Tcp16 + Tcp16 + Tcp16 − #9 Max − Tphl − Tphl #27 Min − $402 Max

= 62.5 ns + 62.5 ns + 62.5 ns + 62.5 ns − 30 ns − 6.2 ns − 7 ns −5 ns − 29 ns

= 172.8 ns @ 16 MHz w/8420A-25 Heavy Load

= Tcp12 + Tcp12 + Tcp12 + Tcp12 − #9 Max − Tphl − Tphl − #27 Min − $402 Max

= 83.3 ns + 83.3 ns + 83.3 ns + 83.3 ns − 40 ns − 6.2 ns −7 ns − 10 ns −35 ns

= 240 ns @ 12 MHz w/8420A-20 Heavy Load

Port B

1 Wait State

tRAC = s1 + s2 + sw + sw + s3 + s4 − 68020 CLK to $\overline{AS}$ Max − 74AS32 Delay Max − 74AS244 Delay Max − 74AS245 Delay Max − 68020 Data Setup Min − $\overline{AREQB}$ Asserted to $\overline{RAS}$ Asserted

= Tcp16 + Tcp16 + Tcp16 − #9 Max − Tphl Max − Tphl Max − Tphl Max − #27 Min − $102 Max

= 62.5 ns + 62.5 ns + 62.5 ns − 30 ns − 6.2 ns − 5 ns − 7 ns − 5 ns − 34 ns

= 110 ns @ 16 MHz w/8420A-25 Heavy Load

= Tcp12 + Tcp12 + Tcp12 − #9 Max − Tphl Max − Tphl Max − Tphl Max − #27 Min − $102 Max

= 83.3 ns + 83.3 ns + 83.3 ns − 40 ns − 6.2 ns − 5 ns − 7 ns − 10 ns − 42 ns

= 139 ns @ 12 MHz w/8420A-20 Heavy Load

2 Wait States

tRAC = s1 + s2 + sw + sw + sw + sw − 68020 CLK to $\overline{AS}$ Max − 74AS32 Delay Max − 74AS244 Delay Max − 74AS245 Delay Max − 68020 Data Setup Min − $\overline{AREQB}$ Asserted to $\overline{RAS}$ Aserted

= Tcp16 + Tcp16 + Tcp16 + Tcp16 − #9 Max − Tphl Max − Tphl Max − Tphl Max − #27 Min − $102 Max

= 62.5 ns + 62.5 ns + 62.5 ns + 62.5 ns − 30 ns − 6.2 ns − 5 ns − 7 ns − 5 ns − 34 ns

= 167 ns @ 16 MHz w/8420A-25 Heavy Load

= Tcp12 + Tcp12 + Tcp12 + Tcp12 − #9 Max − Tphl Max − Tphl Max − Tphl Max − #27 Min − $102 Max

= 83.3 ns + 83.3 ns + 83.3 ns + 83.3 ns −40 ns − 6.2 ns − 5 ns − 7 ns − 10 ns −42 ns

= 223 ns @ 12 MHz w/8420A-20 Heavy Load

Port A

1 Wait State

tCAC = s1 + s2 + sw + sw + s3 + s4 − 68020 CLK to $\overline{AS}$ Max − 74AS244 Delay Max − 74AS245 Delay Max − 68020 Data Setup Min − $\overline{ADS}$ Asserted to $\overline{CAS}$ Asserted

= Tcp16 + Tcp16 + Tcp16 − #9 Max − Tphl − Tphl − #27 Min − $403a Max

= 62.5 ns + 62.5 ns + 62.5 ns − 30 ns − 6.2 ns − 7 ns −5 ns − 82 ns

= 57 ns @ 16 MHz w/8420A-25 Heavy Load

= Tcp12 + Tcp12 + Tcp12 = #9 Max − Tphl − Tphl − #27 Min − $403a Max

= 83.3 ns + 83.3 ns + 83.3 ns −40 ns − 6.2 ns − 7 ns − 10 ns − 94 ns

= 92 ns @ 12 MHz w/8420A-20 Heavy Load

2 Wait States

tCAC = s1 + S2 + sw + sw + sw + sw + s3 + s4 − 68020 CLK to $\overline{AS}$ Max − 74AS244 Delay Max − 74AS245 Delay Max − 68020 Data Setup Min − $\overline{ADS}$ Asserted to $\overline{CAS}$ Asserted

= Tcp16 + Tcp16 + Tcp16 + Tcp16 − #9 Max − Tphl − Tphl − #27 Min − $403a Max

= 62.5 ns + 62.5 ns + 62.5 ns + 62.5 ns − 30 ns − 6.2 ns − 7 ns − 5 ns − 82 ns

= 119 ns @ 16 MHz w/8420A-25 Heavy Load

= Tcp12 + Tcp12 + Tcp12 + Tcp12 − #9 Max − Tphl − Tphl − #27 Min − $403a Max

= 83.3 ns + 83.3 ns + 83.3 ns + 83.3 ns − 40 ns − 6.2 ns − 5 ns − 7 ns − 10 ns − 94 ns

= 171 ns @ 12 MHz w/8420A-20 Heavy Load

Port B

1 Wait State

tCAC = s1 + s2 + sw + sw + s3 + s4 − 68020 CLK to $\overline{AS}$ Max − 74AS32 Delay Max − 74AS244 Delay Max − 74AS245 Delay Max − 68020 Data Setup Min − $\overline{AREQB}$ Asserted to $\overline{CAS}$ Asserted

= Tcp16 + Tcp16 + Tcp16 − #9 Max − Tphl − Tphl − Tphl − #27 Min − $118

= 62.5 ns + 62.5 ns + 62.5 ns − 30 ns − 5 ns − 6.2 ns − 7 ns −5 ns − 88 ns

= 46 ns @ 16 MHz w/8420A-25 Heavy Load

= Tcp12 + Tcp12 + Tcp12 = #9 Max − Tphl − Tphl − Tphl − #27 Min − $118a

= 83.3 ns + 83.3 ns + 83.3 ns −40 ns − 5 ns − 6.2 ns − 7 ns − 10 ns − 103 ns

= 78 ns @ 12 MHz w/8420A-20 Heavy Load

2 Wait States

tCAC = s1 + s2 + sw + sw + sw + sw + s3 + s4 − 68020 CLK to $\overline{AS}$ Max − 74AS32 Delay Max − 74AS244 Delay Max − 74AS245 Delay Max − 68020 Data Setup Min − $\overline{AREQB}$ Asserted to $\overline{CAS}$ Asserted

= Tcp16 + Tcp16 + Tcp16 + Tcp16 − #9 Max − Tphl − Tphl − Tphl − #27 Min − $118a Max

= 62.5 ns + 62.5 ns + 62.5 ns − 30 ns − 5 ns − 6.2 ns − 5 ns − 7 ns − 5 ns − 84 ns

= 109 ns @ 16 MHz w/8420A-25 Heavy Load

= Tcp12 + Tcp12 + Tcp12 + Tcp12 − #9 Max − Tphl − Tphl − Tphl − #27 Min − $118 Max

= 83.3 ns + 83.3 ns + 83.3 ns + 83.3 ns − 40 ns − 5 ns − 6.2 ns − 7 ns − 10 ns − 103 ns

= 162 ns @ 12 MHz w/8420A-20 Heavy Load

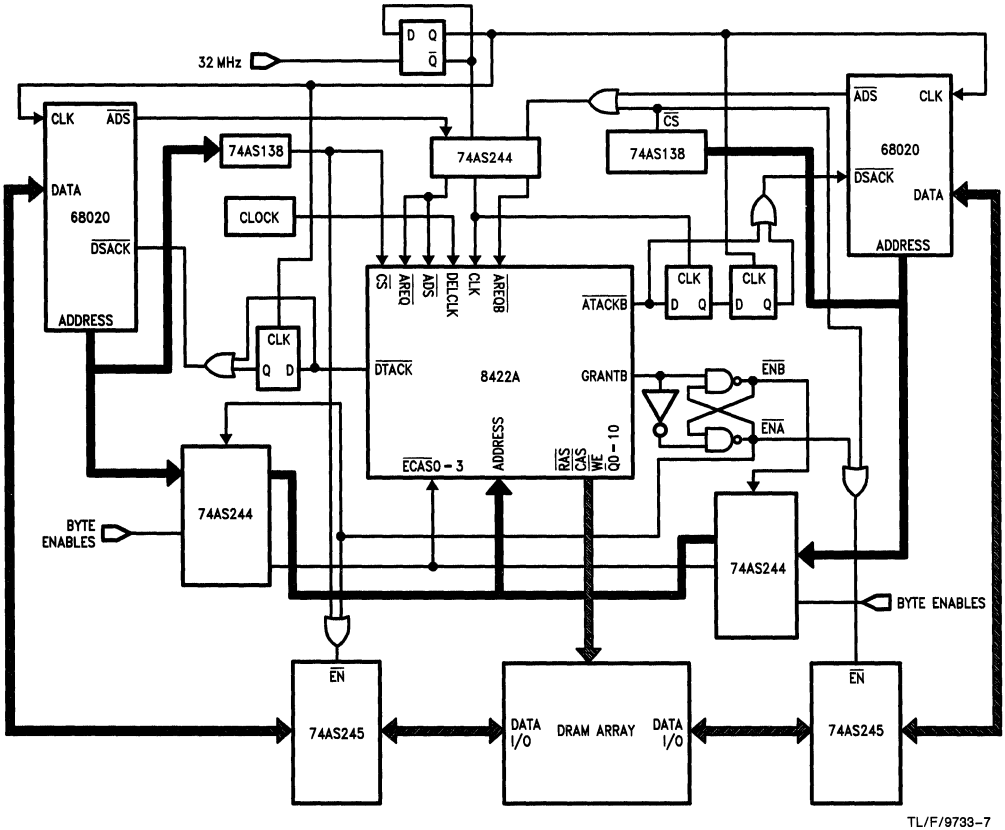


FIGURE 7. 68020 Dual Port System up to 16 MHz

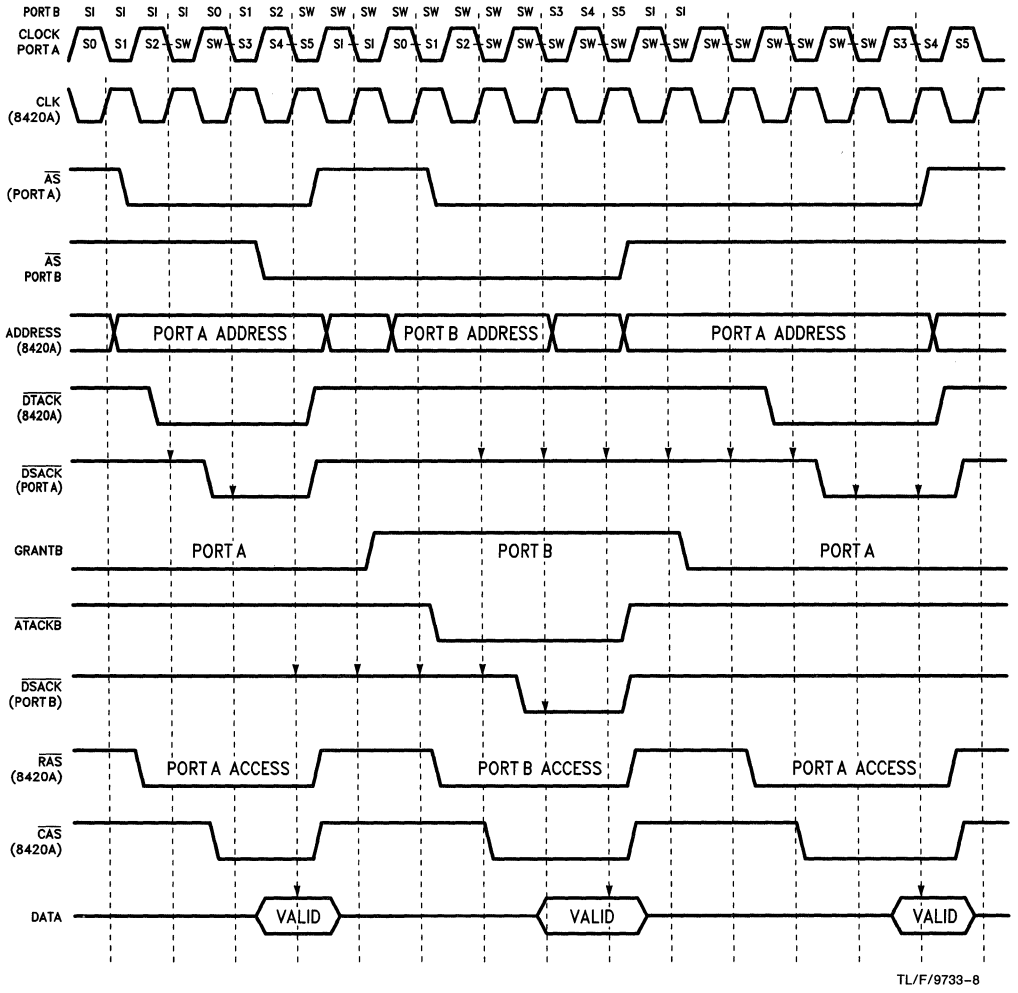
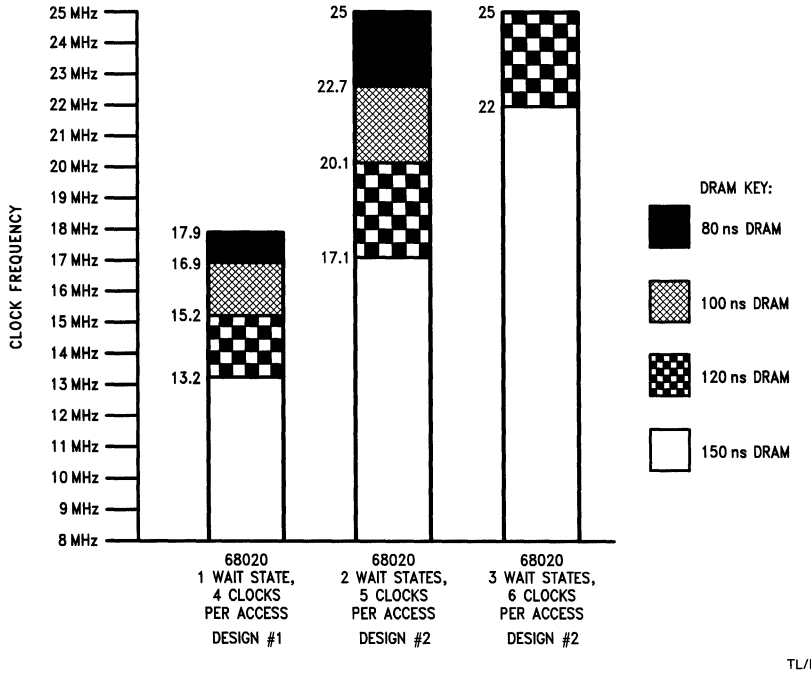


FIGURE 8. 68020 Design #3 Timing, Delayed Dual Port Accesses

DRAM Speed Versus Processor Speed. (DRAM Speed References the RAS Access Time, t_{RAC}, of the DRAM, Using DP8422A-25 Timing Specifications.)



TL/F/9733–9

Interfacing the DP8422A to the 68020 (Zero Wait State Burst Mode Access)

National Semiconductor
Application Note 616
Lawson H. C. Chang

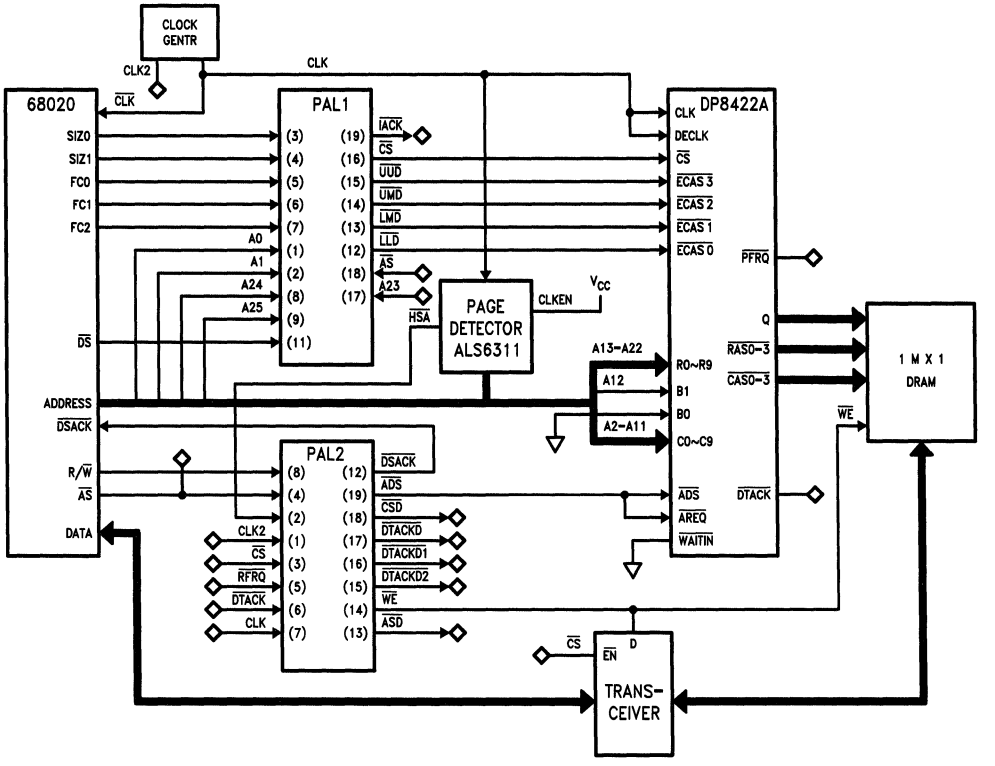
INTRODUCTION

This application note describes interfacing the DP8422A DRAM controller (also applicable to DP8420A/21A) to the 68020 with slower memories. This design is based upon burst mode access by holding $\overline{RAS}$ low and toggling $\overline{CAS}$. It is assumed that the user is familiar with the DP8422A and 68020 mode operations.

DESIGN DESCRIPTION

This design consists of the DP8422A DRAM controller, two PALs (16R4D and 16L8D), and a page detector (ALS6311). This design accommodates two banks of DRAM, each bank being 32 bits in width, giving maximum memory capacity of either 8 Mbytes (using 1M x 1 DRAMs) or 32 Mbytes (using 4M x 1 DRAMs). This design is based on 1M x 1 DRAM running at 16 MHz clock. The schematic diagram of interfacing DP8422A to the 68020 is shown in *Figure 1*. The DP8422A is operated in Mode 1. An access cycle begins when the 68020 places a valid address on the address bus and asserts the Address Strobe ($\overline{AS}$) if a refresh or Port B access (DP8422A only) is not in progress. The proper $\overline{RAS}$ and $\overline{CAS}$ will be asserted respectively, depending upon programming bits C6, C5, and C4 for $\overline{RAS}$ and $\overline{CAS}$ configuration after guaranteeing the programmed value of row address hold time and the column address setup time.

The High Speed Access ($\overline{HSA}$) output signal of page detector indicates whether the current access is in the same page as previous access or not. $\overline{ADS}$ ($\overline{AREQ}$) is kept low if the current access is in the page, otherwise $\overline{ADS}$ ($\overline{AREQ}$) will be forced to go high to terminate the burst access. Internal refresh logic automatically generates refresh request every 15 μs. The timing diagrams are shown in *Figure 2* and *Figure 3*.



TL/F/10440–1

FIGURE 1. Schematic Diagram of Interfacing DP8422A/68020 Burst Access

DP8422A PROGRAMMING BITS

u = User Defined

Programming Bits		Description
R0, R1	= 0, 1	$\overline{RAS}$ High and Low Times
R2, R3	= 0, 0	$\overline{DTACK}$ Generation Mode for Nonburst Access
R4, R5	= 0, 0	$\overline{DTACK}$ Generation Mode for Burst Access
R6	= 0	Add Wait States if $\overline{WAITIN}$ is Low
R7	= 1	$\overline{DTACK}$ Mode Select
R8	= 1	Noninterleave Mode
R9	= u	All $\overline{RAS}$'s or Staggered Refresh Select
C0, C1, C2	= 0, 1, 0	Refresh Clock Divisor Select
C3	= 0	Refresh Clock Divider Select
C4, C5, C6	= 0, 0, 1	$\overline{RAS}$ and $\overline{CAS}$ Configuration Mode
C7	= 1	t_{ASC} Mode Select
C8	= 1	t_{RAH} Mode Select
C9	= u	Delay $\overline{CAS}$ during Write Access Mode Select
B0	= 1	Fall through Mode
B1	= 1	Mode 1 Access
$\overline{ECAS0}$	= 1	Extend $\overline{CAS}$ and Refresh Request

DESIGN TIMING PARAMETERS

Timing parameters are referenced to the numbers shown in the DP8422A data sheet timing parameters. Numbered times starting with a "$" refer to DP8422A timing parameters. Numbered times starting with a "#" refer to 68020 timing parameters.

16 MHz t_{CP} = 62.5 ns

$400: $\overline{ADS}$ Asserted Setup to CLK
½ t_{CP} − PAL t_{CLK} Max.
= 31.25 ns − 8 ns
= 23.25 ns

$401: $\overline{CS}$ Setup to $\overline{ADS}$ Asserted
2 t_{CP} + PAL t_{CLK} Min. − #6 $\overline{CLK}$ to Address Valid − PAL t_P Max.
= 125 ns + 5.5 ns − 30 ns − 10 ns
= 90.5 ns

$416: $\overline{AREQ}$ Negated to $\overline{ADS}$ Asserted
t_{CP} − (PAL t_{CLK} Max. − PAL t_{CLK} Min.)
= 62.5 ns − 2.5 ns
= 60 ns

#47: $\overline{DTACK}$ (68020) Low Setup to CLK Low
½ t_{CP} − PAL t_{CLK} Max. − PAL t_p Max.
= 31.25 ns − 8 ns − 10 ns
= 13.5 ns

t_{RAC} (Nonburst Access):
3 t_{CP} − PAL t_{CLK} Max. − PAL t_p Max. − $402 $\overline{ADS}$ Low to $\overline{RAS}$ Low − #27 Data Setup − F245 Transceiver t_p Max.
= 187.5 ns − 8 ns − 10 ns − 29 ns − 5 ns − 6 ns
= 129.5 ns (-25 Part)
= 187.5 ns − 8 ns − 10 ns − 35 ns − 5 ns − 6 ns
= 123.5 ns (-20 Part)

t_{CAC} (Nonburst Access):
3 t_{CP} − PAL t_{CLK} Max. − PAL t_p Max. − $403a $\overline{ADS}$ Low to $\overline{CAS}$ Low − #27 Data Setup − F245 Transceiver t_p Max.
= 187.5 ns − 8 ns − 10 ns − 82 ns − 5 ns − 6 ns
= 76.5 ns (-25 Part)
= 187.5 ns − 8 ns − 10 ns − 94 ns − 5 ns − 6 ns
= 64.5 ns (-20 Part)

t_{AA} (Nonburst Access):
3 t_{CP} − PAL t_{CLK} Max. − PAL t_p Max. − $417 $\overline{ADS}$ Low to Column Address Valid − #27 Data Setup − F245 Transceiver t_p Max.
= 187.5 ns − 8 ns − 10 ns − 78 ns − 5 ns − 6 ns
= 80.5 ns (-25 Part)
= 187.5 ns − 8 ns − 10 ns − 92 ns − 5 ns − 6 ns
= 66.5 ns (-20 Part)

t_{CAC} (Burst Access):
2 t_{CP} − #9 $\overline{CLK}$ High to $\overline{DS}$ Low Max. − PAL t_p Max. − $14 $\overline{ECAS}$ Low to $\overline{CAS}$ Low Max. − #27 Data Setup − F245 Transceiver t_p Max.
= 125 ns − 30 ns − 10 ns − 27 ns − 5 ns − 6 ns
= 47 ns (-25 Part)
= 125 ns − 30 ns − 10 ns − 31 ns − 5 ns − 6 ns
= 43 ns (-20 Part)

t_{AA} (Burst Access):
3.5 t_{CP} − #6 $\overline{CLK}$ Low to Address Valid Max. − $26 Address Valid to Q Max. − #27 Data Setup − F245 Transceiver t_p Max.
= 218.75 ns − 30 ns − 35 ns − 5 ns − 6 ns
= 142.75 ns (-25 Part)
= 218.75 ns − 30 ns − 38 ns − 5 ns − 6 ns
= 139.75 ns (-20 Part)

68020PAL EQUATIONS

The Boolean entry operators are listed as

":=" Replaced by (After Clock)
"=" Equality
"*" AND
"+" OR
"/" Complement
"~" Active low

The brief explanation of PAL output signals

CS~ This combinational output signal is Chip Select.

CSD~ This sequential output signal is Chip Select Delayed by one clock.

ADS~ This sequential output signal is Address Strobe (also used as an Access Request, / AREQ, to DP8422A).

DTACKD~ This sequential output signal is Data Transfer Delayed by one clock.

DTACKD1~ This sequential output signal is Data Transfer Delayed by two clocks.

DTACKD2~ This sequential output signal is Data Transfer Delayed by three clocks.

WE~ This sequential output signal is Write Enable to DRAM.

DSACK~ This combinational output signal is Data Transfer and Size Acknowledge.

UUD~ This combinational output signal is to select upper upper byte.

UMD~ This combinational output signal is to select upper middle byte.

LMD~ This combinational output signal is to select lower middle byte.

LLD~ This combinational output signal is to select lower lower byte.

IACK~ This combinational output signal is interrupt acknowledge.

(I) PAL1 (PAL16L8D) EQUATIONS

Inputs: A0, A1, A23, A24, A25, FC2, FC1, FC0, SIZ0, SIZ1, DS~, AS~

Outputs:

/IACK~ = /FC2 + /FC1 + /FC0

/CS~ = /A23 * /A24 * /A25 * /FC2 * /FC1 * FC0 + /A23 * /A24 * /A25 * /FC2 * FC1 * /FC0 + /A23 * /A24 * /A25 * FC2 * /FC1 * FC0 + /A23 * /A24 * /A25 * FC2 * FC1 * /FC0

/UUD~ = /A0 * /A1 * /DS~ * /AS~

/UMD~ = /SIZ0 * /A1 * /DS~ * /AS~ + A0 * /A1 * /DS~ * /AS~ + SIZ1 * /A1 * /DS~ * /AS~

/LMD~ = /A0 * /A1 * /DS~ * /AS~ + /A1 * /SIZ0 * /SIZ1 * /DS~ * /AS~ + SIZ1 * SIZ0 /A1 * /DS~ * /AS~ + /SIZ0 * /A1 * A0 * /DS~ * /AS~

/LLD~ = A0 * SIZ0 * SIZ1 * /DS~ * /AS~ + /SIZ0 * /SIZ1 * /DS~ * /AS~ + A0 * A1 * /DS~ * /AS~ + A1 * SIZ1 * /DS~ * /AS~

(II) PAL2 (PAL16R6D) EQUATIONS

Inputs: CLK2, CS~, HSA~, AS~, RFRQ~, DTACK~, CLK, RW~

/ADS~ = /HSA~ * /ASD~

/DSACK~ = /DTACK~ * /DTACKD~

/DTACKD~ := /DTACK~ * /CLK + /DTACKD~ * CLK

/DTACKD1~ := /DTACKD~ * /CLK + /DTACKD1~ * CLK

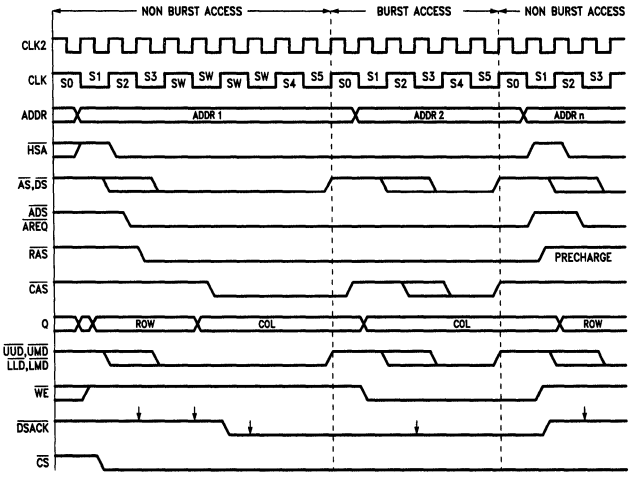
/DTACKD2~ := /DTACKD1~ * /CLK + /DTACKD2~ * CLK

/ASD~ := /CSD~ * CLK /AS~ + /CSD~ * /HSA~ * /AS~ + /CSD~ * /RFRQ~ * /AS~ + /RFRQ~ * CSD~ * /AS~ + /RFRQ~ * CLK * /AS~ + /RFRQ~ * /HSA~ * /AS~ + /CSD~ * CLK * RFRQ~ + /CSD~ * /HSA~ * RFRQ~

/WE~ := /RW~ * AS~ * CLK + /WE~ * /CLK

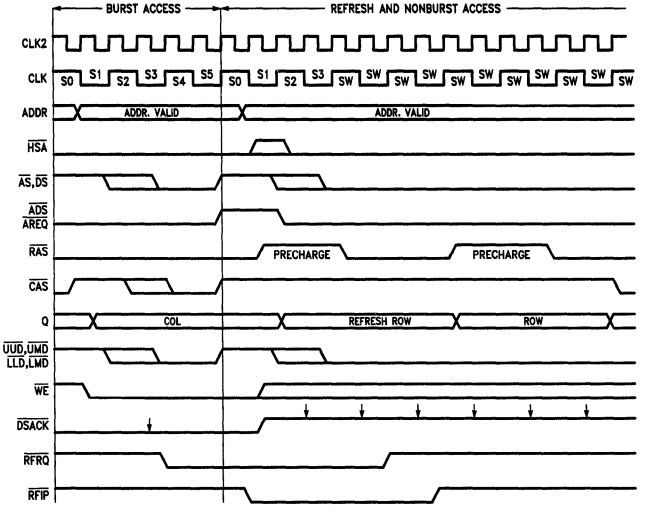
/CSD~ := /CS~ * /CLK + /CSD~ * CLK

Note: PAL1 address inputs, such as A23, A24 and A25, are system memory size dependent.



TL/F/10440–2

FIGURE 2. Timing Diagram of Burst and Nonburst Access



TL/F/10440–3

FIGURE 3. Timing Diagram of Refresh, Burst and Nonburst Access.

Interfacing the DP8422A to an Asynchronous Port B in a Dual 68020 System

National Semiconductor
Application Note 617
Chris Koehle

INTRODUCTION

This application note explains interfacing the DP8422A DRAM controller to two 68020 microprocessors that are running at the same frequency, but asynchronously to each other. This application note is a supplement to AN-539 (Interfacing the DP8420A/21A/22A to the 68020) and is intended to show synchronization logic and timing requirements for a Port B CPU that is running asynchronous to the DP8422A. It is assumed that the reader is already familiar with the 68020 access cycles and the DP8422A modes of operation.

DESIGN DESCRIPTION

This design shows all of the logic necessary to interface an asynchronous 20 MHz 68020 to the DP8422A Port B control inputs. This design is a worst case example and includes some logic that would not be needed in slower systems (i.e., 68000 @ 10 MHz). In our example, a Port B access begins when the asynchronous 68020 places a valid address on the address bus and asserts the address strobe, $\overline{AS}$. In order to synchronize this signal to the DP8422A, it is run through two DQ flip-flops, which are clocked by Port A's input clock. The first DQ is run by an inverted clock in order to reduce the synchronization delay time by ½T. Once $\overline{AREQB}$ is asserted, the access request is latched on the DP8422A and an access will start. If GRANTB is asserted, signaling Port B control of the access port, and a refresh is not in progress, the DP8422A will assert an access $\overline{RAS}$.

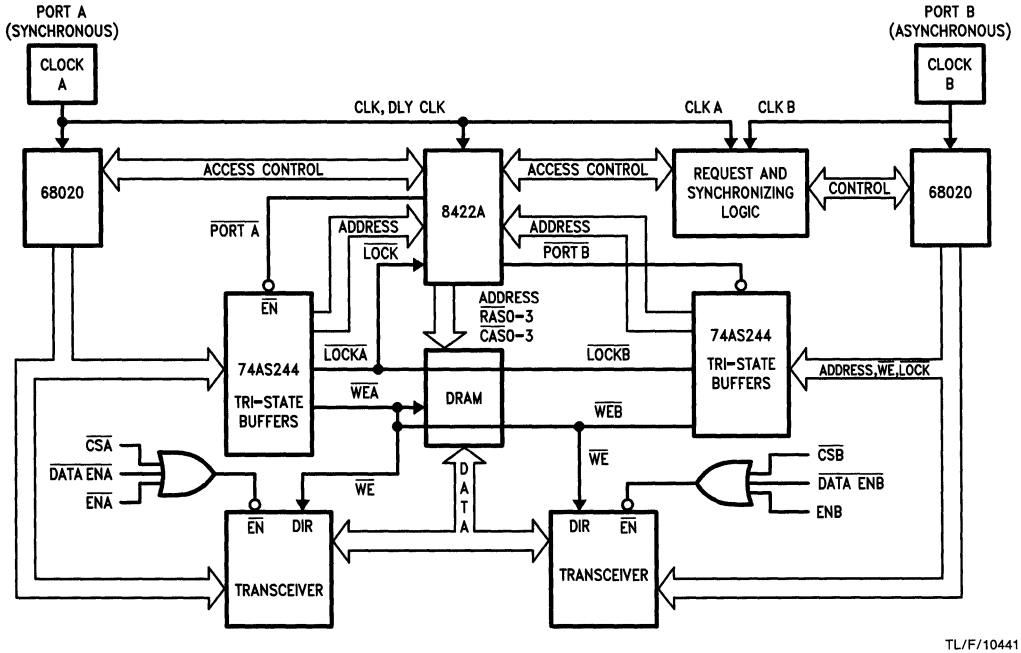
The transfer acknowledge signal, $\overline{ATACKB}$ is also asserted from $\overline{AREQB}$ asserting (or from the same edge of clock that starts $\overline{RAS}$ for a delayed access). $\overline{AREQB}$ also runs through two DQ flip-flops before being input as $\overline{DSACK}$ to the 68020B. These two flip-flops serve a dual purpose, one being to synchronize the $\overline{ATACKB}$ signal to the 68020B clock, and the other is to provide 1T of delay for the $\overline{ATACKB}$ signal.

Once $\overline{DSACK}$ is sampled as asserted by the 68020, $\overline{AS}$ is negated, signifying an end to the present access. $\overline{AS}$ negating is used to negate the $\overline{DSACK}$ input by presetting the two flip-flops connected to this input. This is done to guarantee that the 68020B $\overline{DSACK}$ signal is negated prior to the next access request being valid. The "Q" output that is connected to the $\overline{DSACK}$ input on the 68020 is also run through 1½T of delay (2DQ's) before it is used to preset the flip-flop that drives $\overline{AREQB}$. The preset is used to negate the $\overline{AREQB}$ signal and to hold it negated for the required $\overline{AREQB}$ negated pulse width as specified by the DP8422A. $\overline{AS}$ is also used to preset this signal so $\overline{AREQB}$ does not get asserted in instances where there are not back-to-back accesses. The 1½T delay is used so that the $\overline{AREQB}$ signal is negated after data is sampled by the 68020. In addition to the parameters calculated in AN-539, these additional parameters are included to show how the synchronizing logic still meets the necessary setup times required by the system (the "$" symbol refers to a DP8422A parameter, and the "#" symbol refers to a 68020 parameter).

$101 $\overline{AREQB}$ Asserted Setup to CLK High
(The DP8422A-25 needs 7 ns)
$= 1\ T_{CP} - t_{PHL}$ (74F74 DQ Flip-Flop)
$= 50$ ns $- 8$ ns
$= 42$ ns

#47a $\overline{DSACK}$ Asserted Setup Time
(68020 needs 5 ns)
$\frac{1}{2}\ T_{CP} - t_{PHL}$ (74F74)
25 ns − 8 ns
17 ns

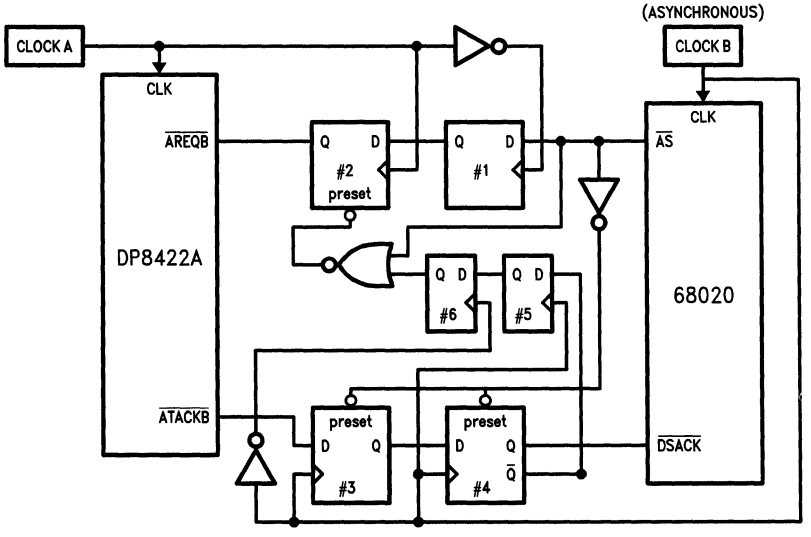
$\overline{AS}$ negated pulse width guarantees the DP8422A meets $\overline{AREQB}$ negated pulse width ($117) through the preset of DQ flip-flop which is connected to $\overline{AREQB}$ input.



TL/F/10441–1

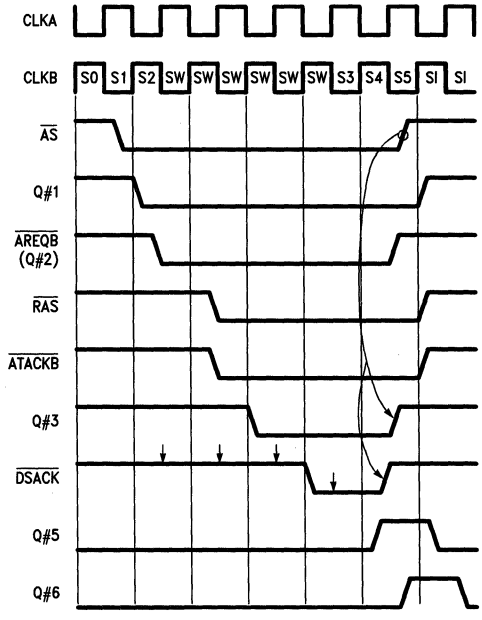
FIGURE 1. Dual Porting with the DP8422A with an Asynchronous Port B

Synchronizing Logic for a Dual 68020 Asynchronous System



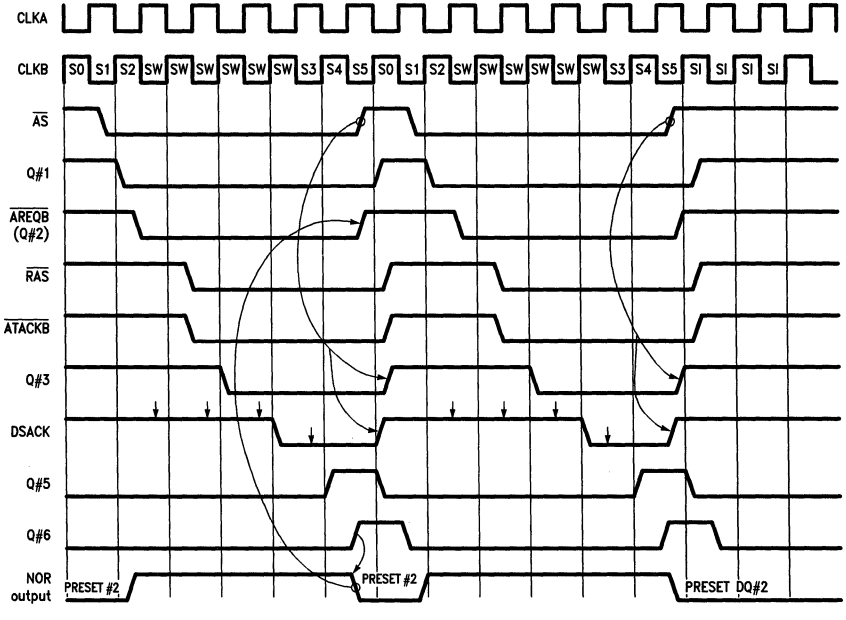
TL/F/10441–2

68020 Asynchronous Port B—Single Access



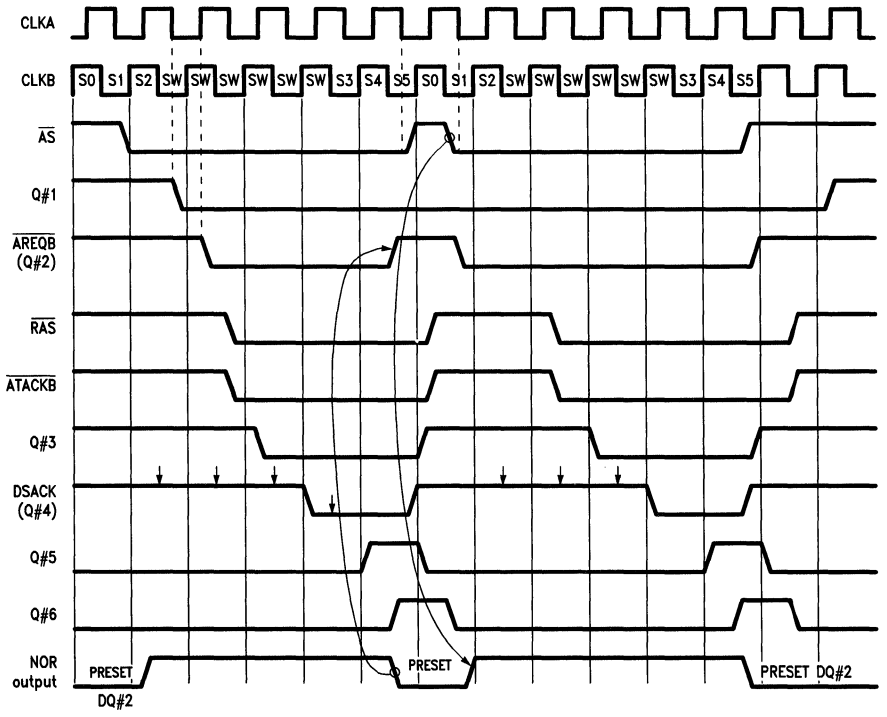
TL/F/10441–3

Back to Back Accesses where DQ Catches AS Negated Pulse in between Accesses



TL/F/10441–4

Back-to-Back Accesses where DQ # 1 Does Not Catch $\overline{AS}$ Negated Pulse in between Accesses



TL/F/10441–5

Interfacing the DP8420A/21A/22A to the 68030 Microprocessor

National Semiconductor
Application Note 537
Webster (Rusty) Meier Jr. and Joe Tate

I. INTRODUCTION

This application note describes how to interface the 68030 microprocessor to the DP8422A DRAM controller (also applicable to DP8420A/21A). It is assumed that the reader is already familiar with 68030 and the DP8422A modes of operation.

II. DESCRIPTION OF DESIGN, ALLOWING UP TO 25 MHz OPERATION WITH 2, 3 OR 4 WAIT STATES IN NORMAL ACCESSES AND 1 OR 2 WAIT STATES DURING BURST ACCESSES

This design drives two banks of DRAM, each bank being 32 bits in width, giving a maximum memory capacity of 32 Mbytes (using 4 Mbit x 1 DRAMs). By choosing a different $\overline{RAS}$ and $\overline{CAS}$ configuration mode (See Programming Mode Bits Section of DP8422A Data Sheet) this application can support 4 banks of DRAM, giving a memory capacity of 64 Mbytes (using 4 Mbit x 1 DRAMs, NOTE that when driving 64 Mbytes the timing calculations will have to be adjusted to the greater capacitive load).

The memory banks are interleaved on every four word (32-bit word) boundry. This means that the address bit (A4) is tied to the bank select input of the DP8422A (B1). Address bits A3, 2 are tied to the highest row and column address inputs to support nibble mode burst accesses (using nibble mode DRAMs). Address bits A1, 0 are used to produce the four byte select data strobes, used in byte reads and writes. If the majority of accesses made by the 68030 are sequential, the 68030 can be doing burst accesses most of the time. Each burst of four words can alternate memory banks, allowing one memory bank to be precharging ($\overline{RAS}$ precharge) while the other bank is being accessed. This is a higher performance memory system then a non-interleaved memory system (bank select on the higher address bits). Each separate memory access to the same memory bank will generally require extra wait states to be inserted into the CPU access cycles to allow for the $\overline{RAS}$ precharge time.

This design supports the 68030 in its synchronous mode of operation. All DRAM accesses are terminated through the 68030 $\overline{STERM}$ input. The burst mode transfer operations are also supported using the synchronous mode of operation. To support these operations nibble mode DRAMs must be used. Nibble mode DRAMs are necessary to support wrap-around during a burst access.

This application allows 2, 3 or 4 wait states to be inserted in normal synchronous accesses and 1 or 2 wait states to be inserted during burst accesses of the 68030. The number of wait states can be adjusted through the $\overline{WAITIN}$ input of the DP8422A and the $\overline{ADDW}$ input of the PAL®, see the table below (the first two rows of the table can also be seen in the timing simulations that appear at the end of this application note):

$\overline{WAITIN}$	$\overline{ADDW}$	Number of Wait States In	
		Normal Access	Burst Access
0	0	4	2
0	1	3	1
1	0	3	2
1	1	2	1

The PAL has an input called "$\overline{EXST}$" that allows the user to setup the $\overline{STERM}$ for any other system peripheral to this PAL. This PAL will synchronize $\overline{EXST}$ to the system clock by gating it with a low logic level on the CLK input (for more information see the PAL equations in Section V). This allows one device to produce $\overline{STERM}$ to the 68030.

The logic shown in this application note forms a complete 68030 memory sub-system, no other logic is needed. This sub-system automatically takes care of:

a. Arbitration between Port A, Port B, and refreshing the DRAM;

b. The insertion of wait states to the processor (Port A and Port B) when needed (i.e, if $\overline{RAS}$ precharge is needed, refresh is happening during a memory access, the other Port is currently doing an access . . . etc.);

c. Performing byte writes and reads to the 32-bit words in memory;

d. Normal and burst access operations.

By making use of the enable input on the 74AS244 buffer, this application can easily be used in a dual access application. The addresses and chip select are tri-stated through this buffer, the write input ($\overline{WIN}$), lock input ($\overline{LOCK}$), and $\overline{ECAS}0-3$ inputs must also be able to be tri-stated (another 74AS244 could be used for this purpose). By multiplexing the above inputs (through the use of the above parts and similar parts for Port B) the DP8422A can be used in a dual access application. If this design is used in a dual access application at 25 MHz the t_{RAC} and t_{CAC} (require $\overline{RAS}$ and $\overline{CAS}$ access time required by the DRAM) will have to be recalculated since the time to $\overline{RAS}$ and $\overline{CAS}$ is longer for the dual access application (See TIMING Section of this application note).

III. 68030 DESIGN, UP TO 25 MHz WITH 2, 3 OR 4 WAIT STATES DURING NORMAL ACCESSES AND 1 OR 2 WAIT STATES DURING BURST ACCESSES, PROGRAMMING MODE BITS

Programming

Bits	Description
R0 = 1	$\overline{RAS}$ low four clocks, $\overline{RAS}$ precharge of three clocks
R1 = 1	
R2 = 1	$\overline{DTACK}$1 is chosen. $\overline{DTACK}$ low first rising CLK edge after access $\overline{RAS}$ is low.
R4 = 0	No WAIT states during burst accesses
R5 = 0	
R6 = 0	If $\overline{WAITIN}$ = 0, add one clock to $\overline{DTACK}$. $\overline{WAITIN}$ may be tied high or low in this application depending upon the number of wait states the user desires to insert into the access.
R7 = 1	Select $\overline{DTACK}$
R8 = 1	Non-Interleaved Mode
R9 = X	
C0 = X C1 = X C2 = X	Select based upon the input "DELCLK" frequency. Example: if the input clock frequency is 20 MHz then choose C0, 1, 2 = 0, 0, 0 (divide by ten, this will give a frequency of 2 MHz). If using the DP8422A over 20 MHz do an initial divide by two externally and then run that output into the DELCLK input and choose the correct divider.
C3 = X	
C4 = 0 C5 = 0 C6 = 1	$\overline{RAS}$ groups selected by "B1". This mode allows two $\overline{RAS}$ outputs to go low during an access, and allows byte writing in 32-bit words.
C7 = 1	Column address setup time of 0 ns.
C8 = 1	Row address hold time of 15 ns.
C9 = 1	Delay $\overline{CAS}$ during write accesses to one clock after $\overline{RAS}$ transitions low
B0 = 1	Fall-thru latches
B1 = 1	Access mode 1
$\overline{ECAS0}$ = 0	$\overline{CAS}$ not extended beyond $\overline{RAS}$

0 = Program with low voltage level

1 = Program with high voltage level

X = Program with either high or low voltage level (don't care condition)

IV. 68030 TIMING CALCULATIONS FOR DESIGN AT 25 MHz WITH 4 WAIT STATES DURING THE NORMAL ACCESSES AND 2 WAIT STATES DURING BURST ACCESSES

1. Minimum $\overline{ADS}$ low setup time to $\overline{CLOCK}$ high for $\overline{DTACK}$ logic to work correctly (DP8422A-25 needs 25 ns):

   40 ns (one clock period) − 10 ns (PAL16R4D combinational output max) = 30 ns

2A. Minimum address setup time to $\overline{ADS}$ low (DP8422A-25 needs 14 ns):

   40 ns (one clock period) − 20 ns (assumed 68030 max time to address valid from CLK high) − 6.2 ns (74AS244 buffer delay max) + 2.5 ns (minimum PAL16R4D combinational output delay) = 16.3 ns

2B. Minimum address setup time to CLK high (used in #3B calculation below):

   40 ns (one clock period) − 20 ns (assumed 68030 max time to address valid from CLK high) − 6.2 ns (74AS244 buffer delay max) = 13.8 ns

3A. Minimum $\overline{CS}$ setup time to $\overline{ADS}$ low (DP8422A-25 needs 5 ns):

   16.3 ns (#2A) − 9 ns (max 74AS138 decoder) = 7.3 ns

3B. Minimum $\overline{CS}$ setup time to CLK high (PAL equations need 0 ns):

   13.8 ns (#2B) − 9 ns (max 74AS138 decoder) = 4.8 ns

4. Determining t_{RAC} during a normal access ($\overline{RAS}$ access time needed by the DRAM):

   180 ns (four and one half clock periods to do the access) − 10 ns (PAL16R4D combinational output, $\overline{ADS}$) − 29 ns ($\overline{ADS}$ to $\overline{RAS}$ low) − 5 ns (68030 data setup time) − 7 ns (74F245) = 129 ns.

   Therefore the t_{RAC} of the DRAM must be 129 ns or less.

5. Determining t_{CAC} during a normal access ($\overline{CAS}$ access time) and column address access time needed by the DRAM:

   180 ns − 10 ns − 5 ns − 7 ns − 75 ns ($\overline{ADS}$ to $\overline{CAS}$ low on DP8422A-25, 50 pF spec) −12 ns [74AS32, 6 ns plus 6 ns extra, taken from lab data on the 74AS32, for driving a 22Ω damping resister and 150 pF of capacitance associated with driving 16 DRAM $\overline{CAS}$ inputs (per $\overline{CAS}$ output)] = 71 ns.

   Therefore the t_{CAC} of the DRAM must be 71 ns or less.

6. Determining the nibble mode access time needed during a burst access:

   120 ns (three clock periods to do the burst) − 20 ns (one half clock period during which $\overline{CAS}$ is high from the previous access) − 20 ns (the data is sampled on a faling clock edge) − 10 ns (PAL16R4D combinational output from CLK input falling edge, $\overline{ENCAS}$) − 12 ns (74AS32 delay to produce $\overline{CAS}$ from the $\overline{ENCAS}$ input, see discription from #5) − 5 ns (68030 data setup time) − 7 ns (74F245) = 46 ns.

   Therefore the nibble mode access time of the DRAM must be 46 ns or less.

7. Maximum time to $\overline{DTACK}$2 low (PAL16RD needs 10 ns setup to CLK):

   40 ns (One clock) − 28 ns ($\overline{DTACK}$2 low from CLK high on DP8422A-25) = 12 ns

8. Minimum $\overline{STERM}$ setup time to CLK (0 ns to CLK rising edge is needed by the 68030):

   20 ns (one half clock period) − 10 ns (PAL16R4D combinational output maximum) = 10 ns

** **Note:** That calculations can be performed for different frequencies and/or different combinations of wait states by substituting the appropriate values into the above equations.

V. 68030 Design, PAL Equations Written In National Semiconductor PLAN™ Format

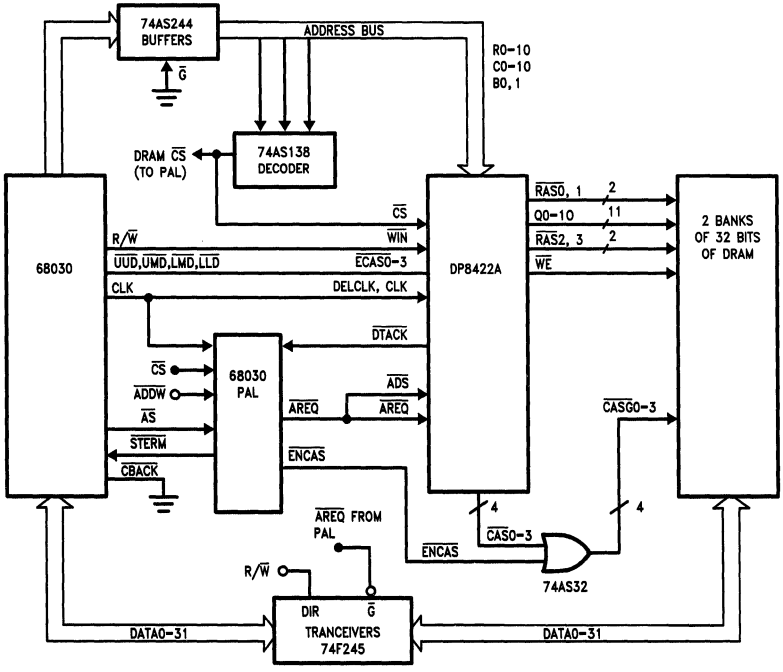
```
PAL16R4D
BCLK /CS /AS NC1 /DTACK /EXST /ADDW CLK NC2 GND
/OE  /STERM /DC NC3 /DB /DA NC4 /ENCAS /AREQ VCC

IF (VCC) /AREQ = /AS * /CS * CLK
               + /AREQ * /CS * /CLK

IF (VCC) /ENCAS = /AREQ * /CS * DC
                + /AREQ * /CS * /CLK

IF (VCC) /DC =   /AS * /CS * /DB * /CLK
               + /AS * /CS * /DC * CLK

IF (VCC) /STERM = /AS * /CS * /DA * DB * /CLK * /ADDW
                + /AS * /CS * /DTACK * DB * /CLK * ADDW
                + /EXST * /CLK
                + /STERM * CLK

*** ''IF (/CS) /STERM'' could be used if the user desires to wire ''OR''
        /STERM outputs together from other peripherals.
/DA: = /AREQ * /CS * /DTACK * DB * /ADDW
/DB: = /AREQ * /CS * /DTACK * /DA * DB * /ADDW
     + /AREQ * /CS * /DTACK * DB * ADDW
```

KEY: READING PAL EQUATIONS WRITTEN IN PLAN

```
EXAMPLE EQUATIONS: IF (VCC) /DC = /AS * /CS * /DB * /CLK
                                + /AS * /CS * /DC * CLK
```

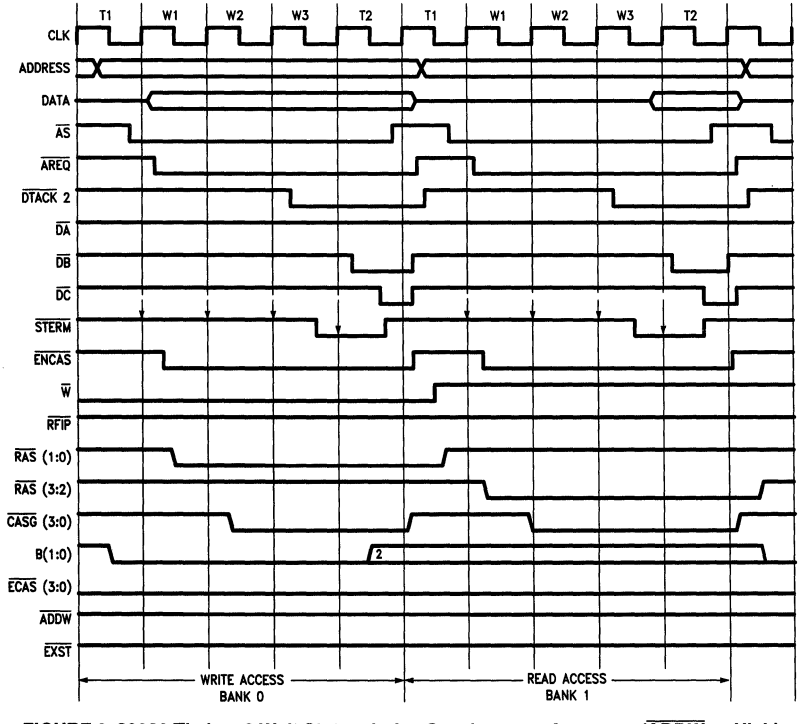
This example reads: the output "$\overline{DC}$" will transition low given that one of the following conditions are valid:

1. The input "$\overline{AS}$" low AND the input "$\overline{CS}$" is low AND the output "$\overline{DB}$" is low and the input "CLK" is low, OR
2. The input "$\overline{AS}$" is low AND the input "$\overline{CS}$" is low AND the output "$\overline{DC}$" is low and the input "CLK" is high.



TL/F/9731-1

FIGURE 1. 68030 Design, up to 25 MHz, with 2, 3 or 4 Wait States in Normal Accesses and 1 or 2 Wait State in Burst Accesses



TL/F/9731-2

FIGURE 2. 68030 Timing, 3 Wait States during Synchronous Accesses ($\overline{ADDW}$ = High)

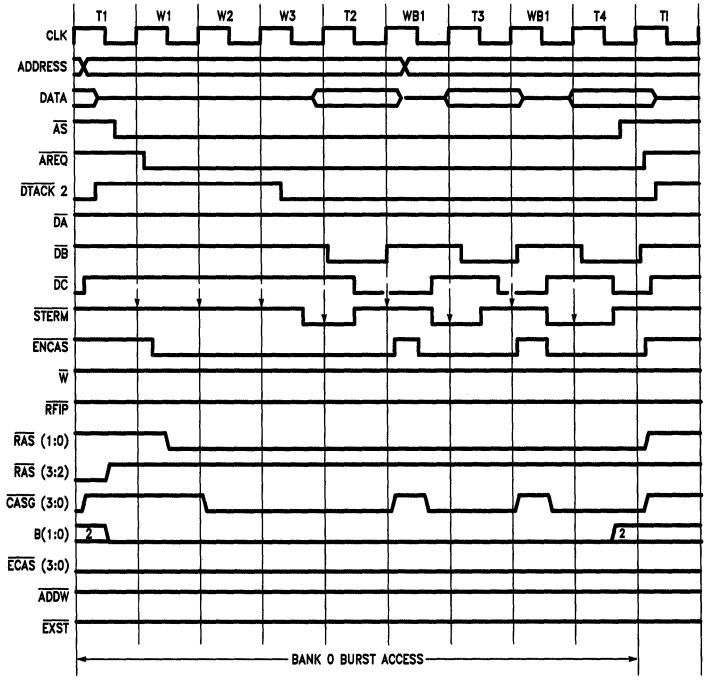


FIGURE 3. 68030 Timing, 3 Wait States during Synchronous Accesses and 1 Wait State during Burst Accesses ($\overline{ADDW}$ = High)

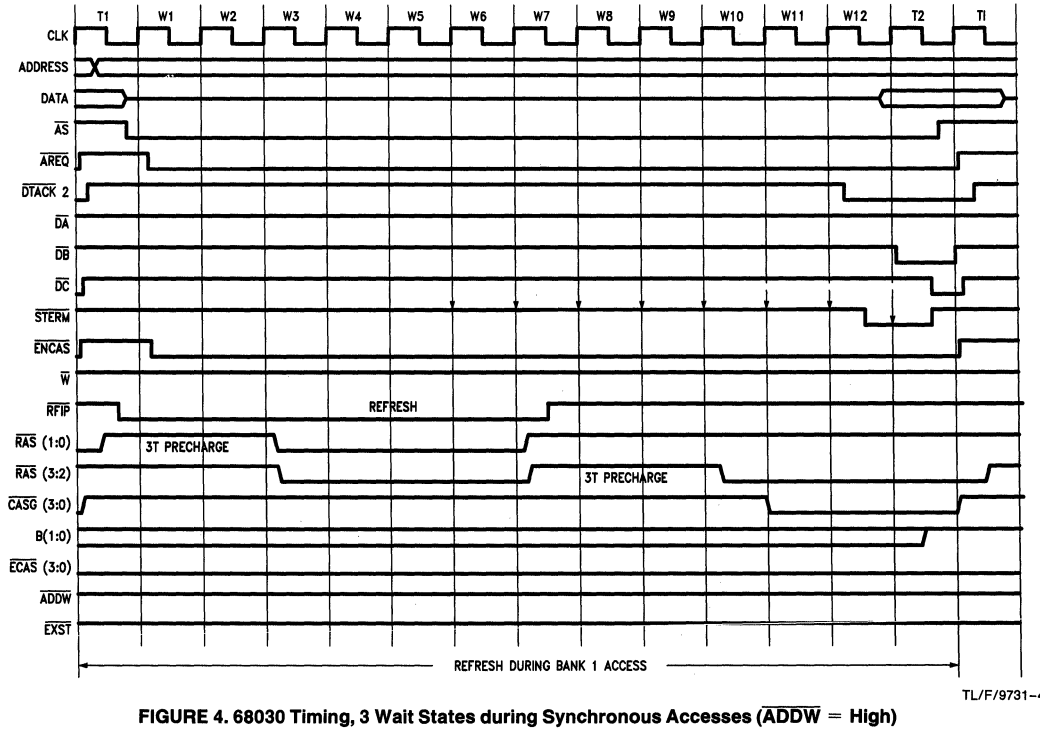
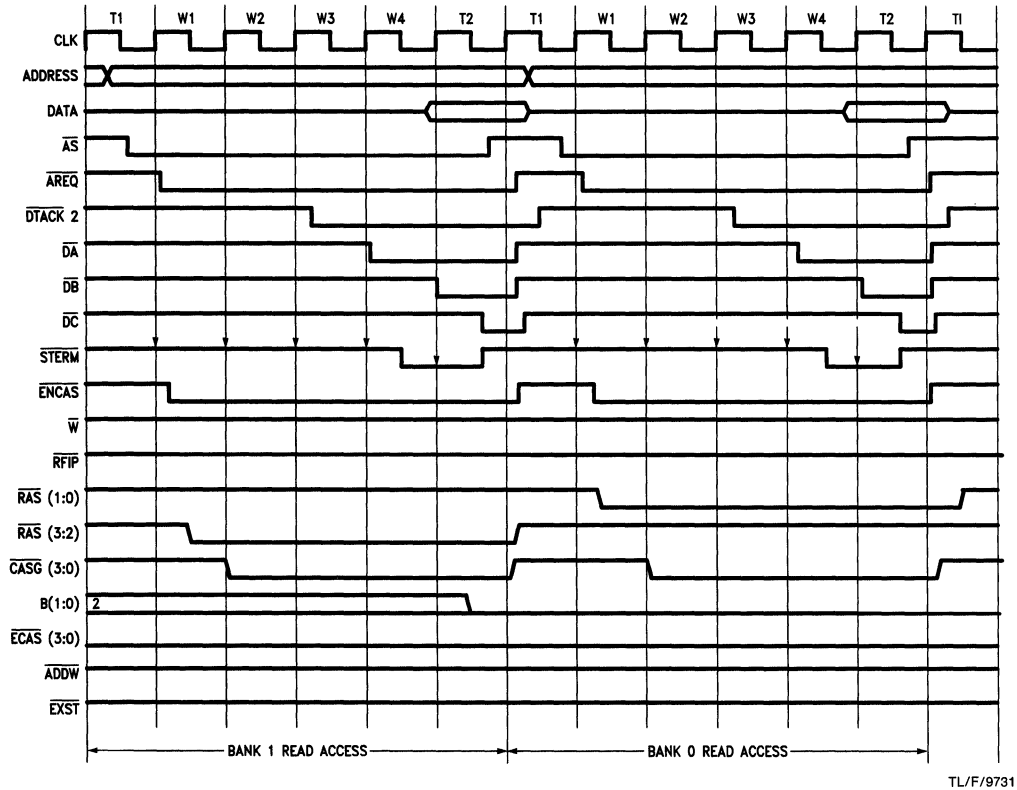
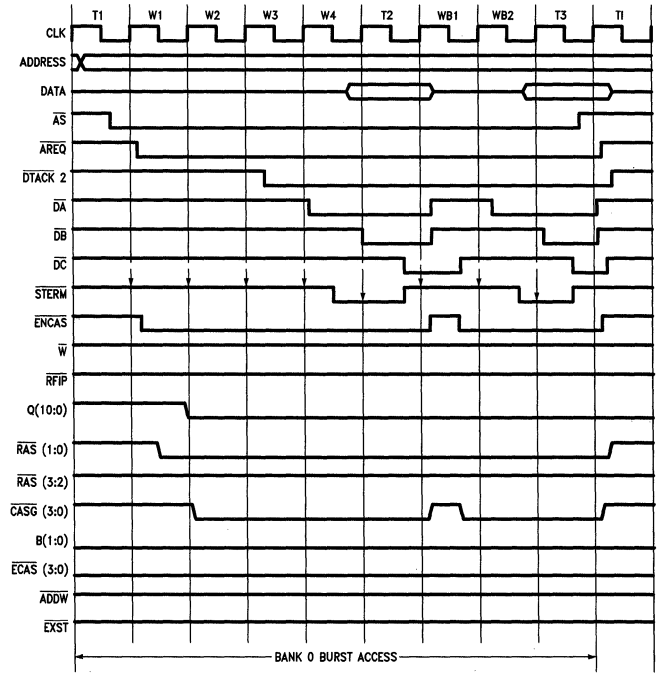


FIGURE 4. 68030 Timing, 3 Wait States during Synchronous Accesses ($\overline{ADDW}$ = High)



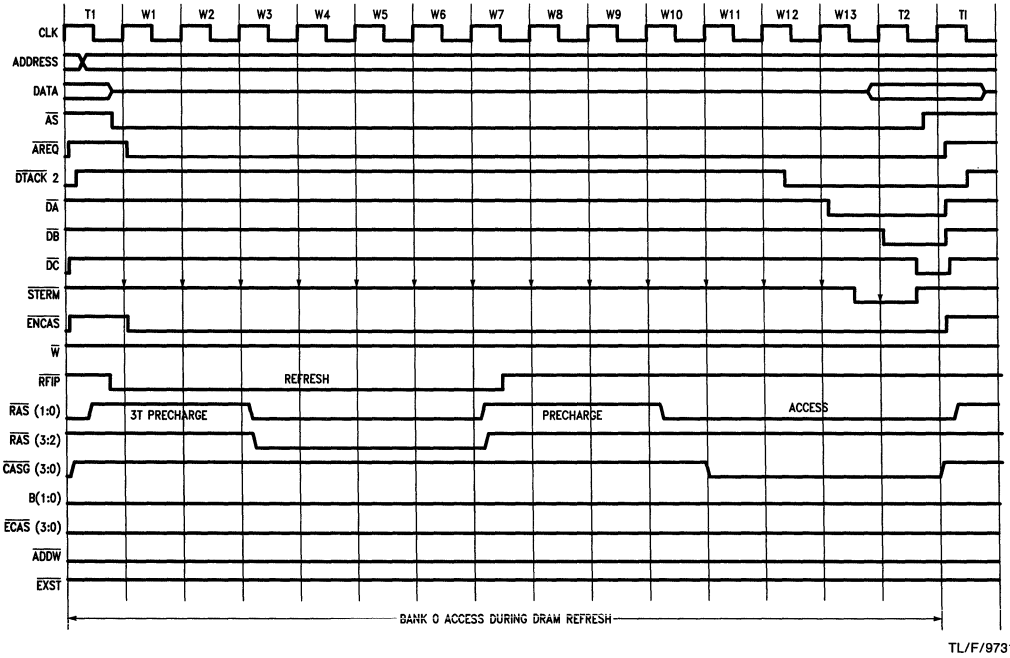
TL/F/9731–5

FIGURE 5. 68030 Timing, 4 Wait States during Synchronous Accesses ($\overline{ADDW}$ = Low)



TL/F/9731–6

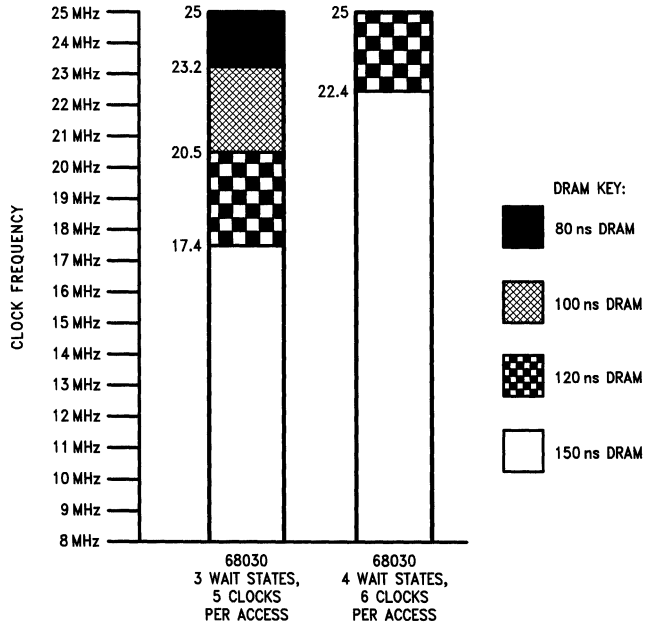
FIGURE 6. 68030 Timing, 4 Wait States during Synchronous Accesses and 2 Wait States during Burst Accesses ($\overline{ADDW}$ = Low)



TL/F/9731–7

FIGURE 7. 68030 Timing, 4 Wait States during Synchronous Accesses ($\overline{ADDW}$ = Low)

DRAM Speed Versus Processor Speed (DRAM Speed References the RAS Access Time, tRAC, of the DRAM Using DP8422A-25 Timing Specifications)



TL/F/9731–8

A Dual Access DP8422A/ 68030/74F632 Error Detecting and Correcting Memory System

National Semiconductor
Application Note 535
Webster (Rusty) Meier Jr. and
Joe Tate

I INTRODUCTION

This application note describes how to interface two 68030 microprocessors, both synchronous to the same system clock, to a DP8422A DRAM controller and a 74F632 EDAC chip. It is assumed that the reader is already familiar with 68030, the DP8422A, and the 74F632 modes of operation. The National Semiconductor DP8402A can be used in place of the 74F632 though its timing is slower.

This application note supports the following types of memory accesses:

1. Read accesses with 6 wait states inserted (8 clock periods total in the synchronous mode read access), any single bit errors are automatically corrected before sending the data to the CPU (EDAC unit in always correct mode/ error monitoring mode is also described);
2. Write accesses with 3 wait states inserted (5 clock periods total in the synchronous mode write access);
3. Byte write accesses with 7 wait states inserted (9 clock periods total in the synchronous mode byte write access);
4. Burst read accesses with 3 wait states in the burst portion of the access (4 clock periods total per synchronous mode burst read memory access);
5. Scrubbing during DRAM refreshes (6 clock periods total during the refresh if no errors, 8 clock periods total during the refresh if any errors), any single bit errors are corrected. The corrected word is then written back to the DRAM.

II DESCRIPTION OF 25 MHz DUAL ACCESS 68030 SYSTEM INTERFACED TO THE DP8422A AND THE 74F632

This design allows two 68030 microprocessors to access a common error corrected dynamic memory system. The error corrected memory system is implemented using the 74F632 EDAC chip in the always correct mode. Whichever 68030 accessed the memory last has a higher priority. Both 68030s are interfaced to the DRAM in the synchronous mode of operation (the accesses are terminated with the 68030 $\overline{\text{STERM}}$ input). This allows the DRAM system to support burst mode accesses.

During read accesses the data is always processed through the EDAC chip (always correct type of system). If a single bit error occurs during a read access this design guarantees correct data to the CPU, but does not write the corrected data back to the DRAM. Single bit soft errors in memory are only corrected (written back to memory) during scrubbing type refreshes. The memory is scrubbed often enough that the probability of accumulating two soft errors in memory is very unlikely.

During read accesses the data is always processed through the 74F632 EDAC chip (i.e., the EDAC data buffers are enabled to provide the data to the CPU). The 74F632 is always put into latch and correct mode during read accesses, even though the data from the memory may be correct. This allows $\overline{\text{CAS}}$ to be toggled early (before the CPU has sampled the data), during burst mode accesses, to start accessing the next word of the burst access.

This design drives two banks of DRAM, each bank being 39 bits in width (32 data bits plus 7 check bits) giving a maximum memory capacity of 32 Mbytes of error corrected memory (using 4 M-bit x 1 DRAMs). By choosing a different $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ configuration mode (see programming mode bits section of DP8422A data sheet) this application can support 4 banks of DRAM, giving a memory capacity of 64 Mbytes (using 4M-bit x 1 DRAMs, NOTE that when driving 64 Mbytes the timing calculations will have to be adjusted to the greater capacitive load).

The memory banks are interleaved on every four word (32-bit word) boundary. This means that the address bit (A4) is tied to the bank select input of the DP8422A (B1).

Address bits A3,2 are tied to the highest row and column address inputs (R9, C9 for 1 Mbit DRAMs) to support burst accesses using nibble mode DRAMs. Nibble mode DRAMs must be used! The reason for this is that nibble mode DRAMs support address wrap-around during a burst access. Address wrap-around is needed during an internal cache miss where the 68030 starts a burst memory access on a non-page boundary (i.e., the first of a 4 word burst may have the least significant address bits, "A3,A2" = 10). Given this condition, the CPU expects word 2, word 3, word 0, word 1. On incrementing from word 3 to word 0 the address bit A4 must not change (the nibble page must remain the same). Nibble mode DRAMs support the address wrap-around feature.

Address bits A1, A0 are used to produce the four byte select data strobes, used in byte reads and writes. If the majority of accesses made by the 68030 are sequential, the 68030 can be doing burst accesses most of the time. Each burst of four words can alternate memory banks (address bit A4 tied to DP8422A pin B1), allowing one memory bank to be precharging ($\overline{\text{RAS}}$ precharge) while the other bank is being accessed. This is a higher performance memory system than a non-interleaved memory system (bank select on the higher address bits). Each separate memory access to the same memory bank will generally require extra wait states to be inserted into the CPU access cycles to allow for the $\overline{\text{RAS}}$ precharge time.

The logic shown in this application note forms a complete 68030 memory sub-system, no other logic is needed. This sub-system automatically takes care of:

A. arbitration between Port A, Port B, and refreshing the DRAM;

B. the insertion of wait states to the processor (Port A and Port B) when needed (i.e., if $\overline{\text{RAS}}$ precharge is needed, refresh is happening during a memory access, the other Port is currently doing an access . . . etc.);

C. performing byte writes and reads to the 32-bit words in memory;

D. normal and burst access operations.

By making use of the enable input on the 74AS244 buffer, this application allows dual access applications. The addresses and chip select are TRI-STATE® through this buffer, the write input ($\overline{WIN}$), lock input ($\overline{LOCK}$), and $\overline{ECAS0-3}$ inputs must also be able to be TRI-STATE (another 74AS244 could be used for this purpose). By multiplexing the above inputs (through the use of the above parts and similar parts for Port B) the DP8422A allows dual accessing to be performed.

III ANOTHER OPTION FOR A 68030 25 MHz DUAL ACCESS EDAC DESIGN: THE EDAC ERROR MONITORING METHOD IN CONJUNCTION WITH THE 68030 ASYNCHRONOUS LATE RETRY FEATURE

The 68030 dual access EDAC system design could use the error monitoring method in conjunction with the 68030 asynchronous late retry feature, instead of the always correct method (design shown in this application note). The error monitoring method can yield a slight improvement in system performance.

By using the error monitoring method of error correction single read accesses or the first read access during a burst access can be shortened by one clock period, allowing a synchronous read access to have only 5 wait states inserted, 7 clock periods total (compared to 6 wait states, 8 clock periods total when doing the always correct method). All other types of accesses (burst reads, byte writes, word writes, refresh scrubbing) will execute in the same number of clock cycles, and in the same manner as described in this application note.

Read accesses can save one wait state because the data from the DRAM memory is assumed to be correct in the error monitoring system design. Therefore the DRAM data is given directly to the CPU instead of running it through the EDAC chip as was done in the always correct method.

In order to do this design it is required that the asynchronous late retry feature of the 68030 and registered transceivers (74F646) be employed.

The asynchronous late retry feature of the 68030 involves pulling the 68030 input signals "$\overline{BERR}$ and $\overline{HALT}$" both low before the falling clock edge of the last clock cycle of the access. Given that this is done the 68030 will suspend all bus activity until $\overline{HALT}$ is brought high and then will retry the aborted bus cycle (unless that access is not currently needed by the CPU). This feature is useful for the case where an error is detected in the DRAM data. In this case $\overline{BERR}$ and $\overline{HALT}$ are brought low until the data from the DRAM is corrected (by the EDAC chip) and written back to the DRAM. $\overline{BERR}$ and $\overline{HALT}$ are then brough high to continue CPU processing.

Registered transceivers (74F646) are necessary during burst mode read accesses because $\overline{CAS}$ transitions high before the CPU has sampled the DRAM data. The registered transceivers hold the data valid until the CPU samples it during these cases.

A read, read with a single bit error, and burst read access timing are shown at the end of this application note implementing the error monitoring method. The user can see how these access cycles differ from the always correct method access cycles.

IV 68030 25 MHz DUAL ACCESS DESIGN, PROGRAMMING MODE BITS

Programming Bits	Description
R0 = 1 R1 = 1	$\overline{RAS}$ low four clocks, $\overline{RAS}$ precharge of three clocks
R2 = 1 R3 = 0	$\overline{DTACK}$ 1 is chosen. $\overline{DTACK}$ low first rising CLK edge after access $\overline{RAS}$ is low.
R4 = 0 R5 = 0	No WAIT states during burst accesses
R6 = 0	If $\overline{WAITIN}$ = 0, add one clock to $\overline{DTACK}$. $\overline{WAITIN}$ may be tied high or low in this application depending upon the number of wait states the user desires to insert into the access.
R7 = 1	Select $\overline{DTACK}$
R8 = 1 R9 = X	Non-interleaved mode
C0 = X C1 = X C2 = X	Select based upon the input "DELCLK" frequency. Example: if the input clock frequency is 20 MHz then choose C0,1,2 = 0,0,0 (divide by ten, this will give a frequency of 2 MHz). If DELCLK of the DP8422A is over 20 MHz do an initial divide by two externally and then run that output into the DELCLK input and choose the correct divider.
C3 = X C4 = 0 C5 = 0 C6 = 1	$\overline{RAS}$ groups selected by "B1". This mode allows two $\overline{RAS}$ outputs to go low during an access, and allows byte writing in 32-bit words.
C7 = 1	Column address setup time of 0 ns
C8 = 1	Row address hold time of 15 ns
C9 = 1	Delay $\overline{CAS}$ during write accesses to one clock after $\overline{RAS}$ transitions low
B0 = 1	Fall-thru latches
B1 = 1	Access mode 1
$\overline{ECAS0}$ = 0	Non-extend $\overline{CAS}$

0 = Program with low voltage level

1 = Program with high voltage level

X = Program with either high or low voltage level (don't care condition)

V 68030 25 MHz WORST CASE TIMING CALCULATIONS

The worst case access is an access from Port B. This occurs because the time to $\overline{RAS}$ and $\overline{CAS}$ low is longer for the Port B access than; a Port A access, a refresh with scrubbing access, or an access which has been delayed from starting (due to refresh, $\overline{RAS}$ precharge time, or the other Port accessing memory).

A. Worst case time to $\overline{RAS}$ low from the beginning of an access cycle:

40 ns (T1 clock period of 68030) + 10 ns (PAL16R4D maximum combinational output delay to produce $\overline{AREQB}$) + 41 ns (DP8422A-25 parameter #102, $\overline{AREQ}$ to $\overline{RAS}$ delay maximum) = 91 ns

B. Worst case time to $\overline{CAS}$ low from the beginning of an access cycle:

40 ns + 10 ns + 94 ns (DP8422A-25 parameter #118a, $\overline{AREQB}$ to $\overline{CAS}$ delay maximum) = 144 ns

C. Worst case time to DRAM data valid:

144 ns (from "B" above, maximum time to $\overline{CAS}$) + 50 ns ($\overline{CAS}$ access time "t_{CAC}" for a typical 100 ns DRAM) = 194 ns

D. Worst case time to data valid on the EDAC data bus:

194 ns (from "C" above) + 7 ns (74AS244 maximum delay) = 201 ns

E. Worst case time until the error flags are valid from the 74F632:

201 ns (from "D" above) + 31 ns (74F632 maximum time to error flags valid) = 232 ns

F. Worst case time until corrected data is valid from the 74F632:

201 ns (from "D" above) + 28 ns (74F632 maximum time from data in to corrected data out) = 229 ns

G. Worst case time until corrected data is available at the CPU:

229 ns (from "F" above) + 7 ns (74F245 maximum delay) = 236 ns

VI 68030 25 MHz DUAL ACCESS DESIGN, TIMING CALCULATIONS

1. Minimum $\overline{ADS}$ low setup time to $\overline{CLOCK}$ high for $\overline{DTACK}$ logic to work correctly (DP8422A-25 needs 25 ns, parameter #400b):

   40 ns (one clock period) − 10 ns (PAL16R4D combinational output maximum that produces $\overline{AREQ}$, $\overline{ADS}$) = 30 ns

2a. Minimum address setup time to $\overline{ADS}$ low (DP8422A-25 needs 14 ns, parameter #404):

   40 ns (one clock period) − 20 ns (assumed 68030 max time to address valid from CLK high) − 6.2 ns (74AS244 buffer delay max) + 2.5 ns (minimum PAL 16R4D combinational output delay that produces $\overline{AREQ}$, $\overline{ADS}$) = 16.3 ns

2b. Minimum address setup time to CLK high (used in #3B calculation below):

   40 ns (one clock period) − 20 ns (assumed 68030 max time to address valid from CLK high) − 6.2 ns (74AS244 buffer delay max) = 13.8 ns

3a. Minimum $\overline{CS}$ setup time to $\overline{ADS}$ low (DP8422A-25 needs 5 ns, parameter #401):

   16.3 ns (#2a) − 9 ns (max 74AS138 decoder) = 7.3 ns

3b. Minimum $\overline{CS}$ setup time to CLK high (PAL equations need 0 ns):

   13.8 ns (#2b) − 9 ns (max 74AS138 decoder) = 4.8 ns

4. Determining t_{RAC} during a normal access ($\overline{RAS}$ access time needed by the DRAM):

   200 ns (five and one-half clock periods to get data from the DRAM to the 74F632 data inputs) − 3 ns (74F632 data setup time to mode input S0 high) + 2.5 ns (minimum PAL16R4D combinational output delay for "S0") − 84 ns (from "A" of worst case times, from the beginning of the access to $\overline{RAS}$ low) − 6.2 ns (74F244 DRAM buffer delay maximum) = 129.3 ns

   Therefore the t_{RAC} of the DRAM must be 129.3 ns or less.

5. Determining t_{CAC} during a normal access ($\overline{CAS}$ access time) and column address access time needed by the DRAM:

   220 ns (five and one-half clock periods to get data from the DRAM to the 74F632 data inputs) − 3 ns (74F632 data setup time to mode input S0 high) + 2.5 ns (minimum PAL16R4D combinational output delay for "S0") − 138 ns (from "B" of worst case times, from the beginning of the access to $\overline{CAS}$ low) − 6.2 ns (74F244 DRAM buffer delay maximum) = 75.3 ns

   Therefore the t_{CAC} of the DRAM must be 75.3 ns or less.

6. Determining the nibble mode access time needed during a burst access:

   100 ns (two and one-half clock periods to do the burst) − 8 ns (PAL16R4D clocked output delay maximum for $\overline{ENCAS}$ output) − 27 ns (DP8422A-25 $\overline{ECASn}$ to $\overline{CASn}$ asserted maximum, parameter #14) − 3 ns (74F632 data setup time to mode input S0 high) + 2.5 ns (minimum PAL16R4D combinational output delay for "S0") − 6.2 ns (74F244 DRAM buffer delay maximum) = 58.3 ns

   Therefore the nibble mode access time of the DRAM must be 58.3 ns or less

7. Maximum time to $\overline{DTACK1}$ low (PAL16R4D needs 10 ns setup to CLK):

   40 ns (One clock) − 28 ns ($\overline{DTACK2}$ low from CLK high on DP8422A-25, parameter #18) = 12 ns

8. Minimum $\overline{STERM}$ setup time to CLK (0 ns to CLK rising edge is needed by the 68030):

   20 ns (one-half clock period) − 10 ns (PAL16R4D combinational output maximum) = 10 ns

**Note: That calculations can be performed for different frequencies and/or different combinations of wait states by substituting the appropriate values into the above equations.

VII 68030 25 MHz DUAL ACCESS EDAC SYSTEM DESIGN, PAL EQUATIONS WRITTEN IN NATIONAL SEMICONDUCTOR PLAN™ FORMAT

DP1
PAL16R4D
BCLK CLK $\overline{CSASA}$ $\overline{CSB}$ $\overline{ASB}$ $\overline{DTACK}$ $\overline{ATACKB}$ $\overline{WCBREQ}$ $\overline{RFIP}$ GND $\overline{OE}$ $\overline{RAS0}$ $\overline{COUNT}$ $\overline{ENCAS}$ $\overline{D3}$ $\overline{D2}$ $\overline{D1}$ $\overline{AREQB}$ $\overline{AREQ}$ VCC

IF (VCC) $\overline{AREQ}$ = $\overline{CSASA}$ * CLK
+ $\overline{AREQ}$ * $\overline{CSASA}$
+ $\overline{AREQ}$ * $\overline{CLK}$

IF (VCC) $\overline{AREQB}$ = $\overline{CSB}$ * $\overline{ASB}$ * CLK
+ $\overline{AREQB}$ * $\overline{CSB}$ * $\overline{ASB}$
+ $\overline{AREQB}$ * $\overline{CLK}$

IF (VCC) $\overline{COUNT}$ = $\overline{AREQ}$ * $\overline{DTACK}$ * $\overline{CSASA}$
+ $\overline{AREQB}$ * $\overline{ATACKB}$ * $\overline{ASB}$
+ $\overline{RFIP}$ * $\overline{RAS0}$

$\overline{D1}$:= $\overline{AREQ}$ * $\overline{DTACK}$
+ $\overline{ATACKB}$ * $\overline{AREQB}$
+ $\overline{RFIP}$ * $\overline{RAS0}$

$\overline{D2}$:= $\overline{D1}$ * D3 * $\overline{COUNT}$
+ D3 * $\overline{AREQ}$ * $\overline{DTACK}$ * RFIP

$\overline{D3}$:= $\overline{D2}$ * $\overline{COUNT}$

$\overline{ENCAS}$: = $\overline{WCBREQ}$
+ D1
+ $\overline{D2}$
+ D3
+ $\overline{RFIP}$

DP2
PAL16L8D
BCLK R $\overline{WORD}$ GRANTB $\overline{RFIP}$ $\overline{SERR}$ $\overline{D2}$ $\overline{D5}$ $\overline{D6}$ GND $\overline{OE}$ $\overline{STERMB}$ $\overline{STERMA}$ $\overline{OECB}$ $\overline{OEB}$ $\overline{TRAN\_EN}$ S1 S0 $\overline{EXRF}$ VCC

IF (VCC) $\overline{EXRF}$ = $\overline{RFIP}$ * S1 * $\overline{D2}$ * D5 * D6 * SERR
+ $\overline{EXRF}$ * $\overline{RFIP}$ * S1 * D5 * D6
+ $\overline{RFIP}$ * $\overline{D5}$ * $\overline{SERR}$

IF (VCC) $\overline{S0}$ = $\overline{R}$ * $\overline{WORD}$ * RFIP
+ $\overline{D2}$ * D5
+ $\overline{S0}$ * BCLK
+ $\overline{D5}$ * $\overline{BCLK}$
+ $\overline{S0}$ * $\overline{D5}$
+ $\overline{S0}$ * $\overline{D6}$
+ $\overline{S1}$ * $\overline{SERR}$ * $\overline{RFIP}$

IF (VCC) $\overline{S1}$ = $\overline{R}$ * $\overline{WORD}$ * RFIP
+ $\overline{D5}$ * $\overline{BCLK}$
+ $\overline{S1}$ * $\overline{D5}$
+ $\overline{S1}$ * $\overline{D6}$ * $\overline{R}$ * WORD
+ $\overline{S1}$ * $\overline{D6}$ * $\overline{RFIP}$
+ $\overline{S1}$ * $\overline{SERR}$ * $\overline{RFIP}$

IF (VCC) $\overline{TRAN\_EN}$ = R * $\overline{D5}$ * $\overline{BCLK}$ * RFIP
+ $\overline{TRAN\_EN}$ * R * $\overline{D5}$ * D6 * RFIP
+ R * $\overline{D5}$ * $\overline{STERMA}$ * RFIP
+ R * $\overline{D5}$ * $\overline{STERMB}$ * RFIP
+ $\overline{R}$ * $\overline{WORD}$ * $\overline{S1}$ * RFIP
+ $\overline{R}$ * WORD * $\overline{D5}$ * $\overline{BCLK}$ * RFIP
+ $\overline{TRAN\_EN}$ * $\overline{R}$ * WORD * $\overline{D5}$ * RFIP
+ $\overline{TRAN\_EN}$ * $\overline{R}$ * WORD * $\overline{D6}$ * RFIP

IF (VCC) $\overline{OEB}$ = R * $\overline{D5}$ * $\overline{BCLK}$
+ $\overline{OEB}$ * R * $\overline{D5}$
+ $\overline{RFIP}$ * $\overline{D5}$ * $\overline{BCLK}$ * $\overline{SERR}$
+ $\overline{OEB}$ * $\overline{RFIP}$ * $\overline{D5}$ * $\overline{SERR}$
+ $\overline{OEB}$ * $\overline{RFIP}$ * $\overline{D6}$ * $\overline{SERR}$
+ $\overline{R}$ * WORD * $\overline{D5}$ * $\overline{BCLK}$
+ $\overline{OEB}$ * $\overline{R}$ * WORD * $\overline{D5}$
+ $\overline{OEB}$ * $\overline{R}$ * WORD * $\overline{D6}$

IF (VCC) $\overline{OECB}$ = $\overline{R}$ * $\overline{WORD}$ * RFIP * $\overline{S1}$
+ $\overline{RFIP}$ * $\overline{D5}$ * $\overline{BCLK}$ * $\overline{SERR}$
+ $\overline{OECB}$ * $\overline{RFIP}$ * $\overline{D5}$ * $\overline{SERR}$
+ $\overline{OECB}$ * $\overline{RFIP}$ * $\overline{D6}$ * $\overline{SERR}$
+ $\overline{R}$ * WORD * $\overline{D5}$ * $\overline{BCLK}$
+ $\overline{OECB}$ * $\overline{R}$ * WORD * $\overline{D5}$
+ $\overline{OECB}$ * $\overline{R}$ * WORD * $\overline{D6}$

IF (VCC) $\overline{STERMA}$ = R * RFIP * $\overline{D5}$ * D6 * $\overline{GRANTB}$ * $\overline{BCLK}$
+ $\overline{STERMA}$ * R * RFIP * $\overline{D5}$ * $\overline{GRANTB}$ * BCLK
+ $\overline{R}$ * $\overline{WORD}$ * RFIP * $\overline{D2}$ * D6 * $\overline{GRANTB}$ * $\overline{BCLK}$
+ $\overline{STERMA}$ * $\overline{R}$ * $\overline{WORD}$ * RFIP * $\overline{D2}$ * D6 * $\overline{GRANTB}$ * BCLK
+ $\overline{R}$ * WORD * RFIP * $\overline{D5}$ * $\overline{D6}$ * $\overline{GRANTB}$ * $\overline{BCLK}$
+ $\overline{STERMA}$ * $\overline{R}$ * WORD * RFIP * $\overline{D6}$ * $\overline{GRANTB}$ * BCLK

IF (VCC) $\overline{STERMB}$ = R * RFIP * $\overline{D5}$ * D6 * GRANTB * $\overline{BCLK}$
+ $\overline{STERMB}$ * R * RFIP * $\overline{D5}$ * GRANTB * BCLK
+ $\overline{R}$ * $\overline{WORD}$ * RFIP * $\overline{D2}$ * D6 * $\overline{GRANTB}$ * $\overline{BCLK}$
+ $\overline{STERMB}$ * $\overline{R}$ * $\overline{WORD}$ * RFIP * $\overline{D2}$ * D6 * GRANTB * BCLK
+ $\overline{R}$ * WORD * RFIP * $\overline{D5}$ * $\overline{D6}$ * GRANTB * $\overline{BCLK}$
+ $\overline{STERMB}$ * $\overline{R}$ * WORD * RFIP * $\overline{D6}$ * GRANTB * BCLK

DP3
PAL16R4D
BCLK CLK $\overline{S0}$ $\overline{S1}$ $\overline{ERR}$ $\overline{MERR}$ $\overline{COUNT}$ $\overline{D2}$ $\overline{D3}$ GND $\overline{OE}$ $\overline{OECB}$ $\overline{BERR}$ $\overline{D6}$ $\overline{D5}$ $\overline{D4}$ $\overline{WE}$ $\overline{SERR}$ $\overline{LEDB0}$ VCC

IF (VCC) $\overline{LEDB0}$ = $\overline{D2}$ * $\overline{S0}$ * S1 * $\overline{CLK}$
+ $\overline{LEDB0}$ * $\overline{D3}$ * $\overline{S0}$
+ $\overline{LEDB0}$ * $\overline{CLK}$

IF (VCC) $\overline{SERR}$ = $\overline{D4}$ * S0 * S1 * $\overline{COUNT}$ * $\overline{ERR}$ * CLK
+ $\overline{SERR}$ * $\overline{COUNT}$

IF (VCC) $\overline{BERR}$ = D4 * S0 * S1 * $\overline{COUNT}$ * $\overline{MERR}$ * CLK
+ $\overline{BERR}$ * $\overline{COUNT}$

$\overline{WE}$:= $\overline{S1}$ * $\overline{D2}$ * D3 * $\overline{COUNT}$ * $\overline{OECB}$

$\overline{D4}$:= $\overline{D3}$ * $\overline{COUNT}$

$\overline{D5}$:= $\overline{D4}$ * $\overline{COUNT}$

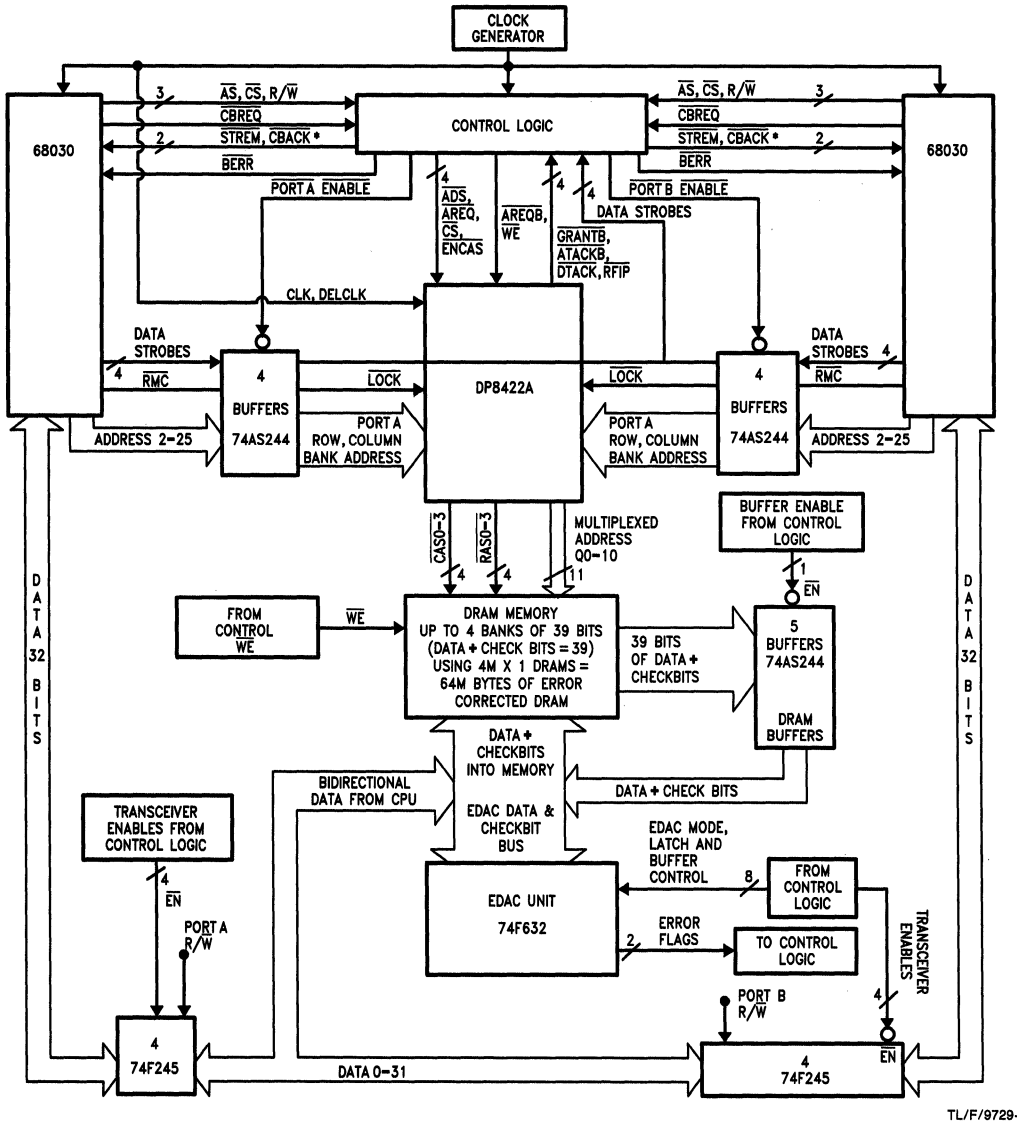
$\overline{D6}$:= $\overline{D5}$ * $\overline{COUNT}$

Key: Reading PAL equations written in PLAN
EXAMPLE EQUATIONS:

IF (VCC) $\overline{AREQ}$ = $\overline{CSASA}$ * CLK
+ $\overline{AREQ}$ * $\overline{CSASA}$
+ $\overline{AREQ}$ * $\overline{CLK}$

This example reads: the output "$\overline{AREQ}$" will transition low given that one of the following conditions are valid;

1. the input "$\overline{CSASA}$" is low AND the input "CLK" is high, OR
2. the output "$\overline{AREQ}$" is low AND the input "$\overline{CSASA}$" is low, OR
3. the output "$\overline{AREQ}$" is low AND the input "CLK" is low.

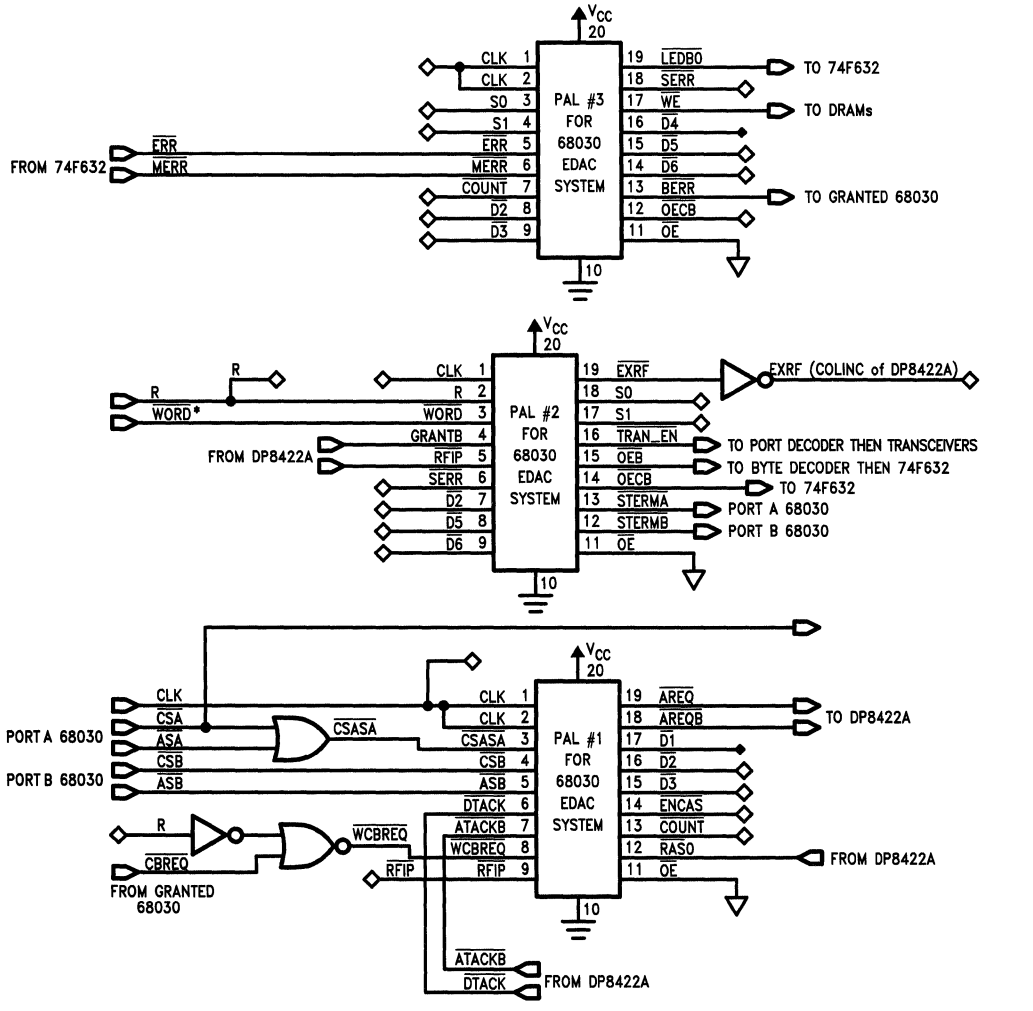


TL/F/9729-1

Control logic in this system needs the following: 3 PAL®s and some logic gates

*$\overline{\text{CBACK}}$ is tied low back to 68030

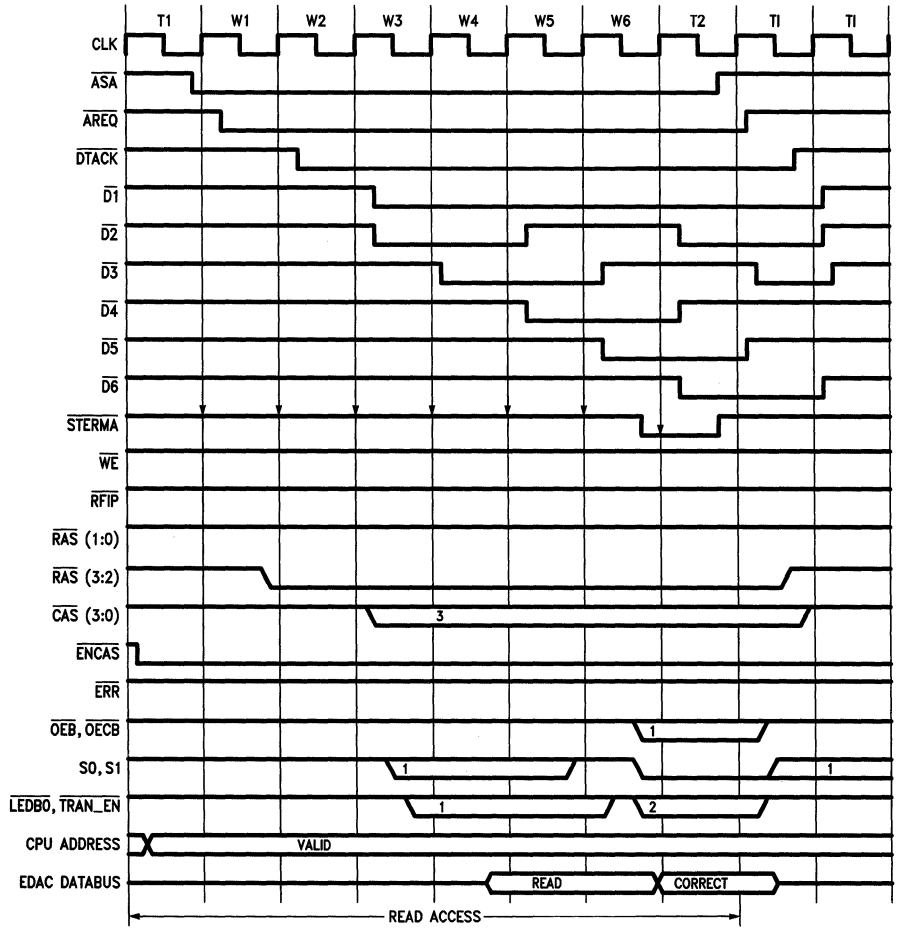
FIGURE 1. Block Diagram of Dual Access 68030 Error Detecting and Correcting (74F632) Memory System



TL/F/9729–2

*If $\overline{\text{WORD}}$ is low then 32 bits are being accessed from the memory system.
If $\overline{\text{WORD}}$ is high then less than 32 bits are being accessed from the memory system.

FIGURE 2. Control Logic for 68030 Dual Access EDAC Memory System



TL/F/9729–3

FIGURE 3. 68030 EDAC Read Access Timing

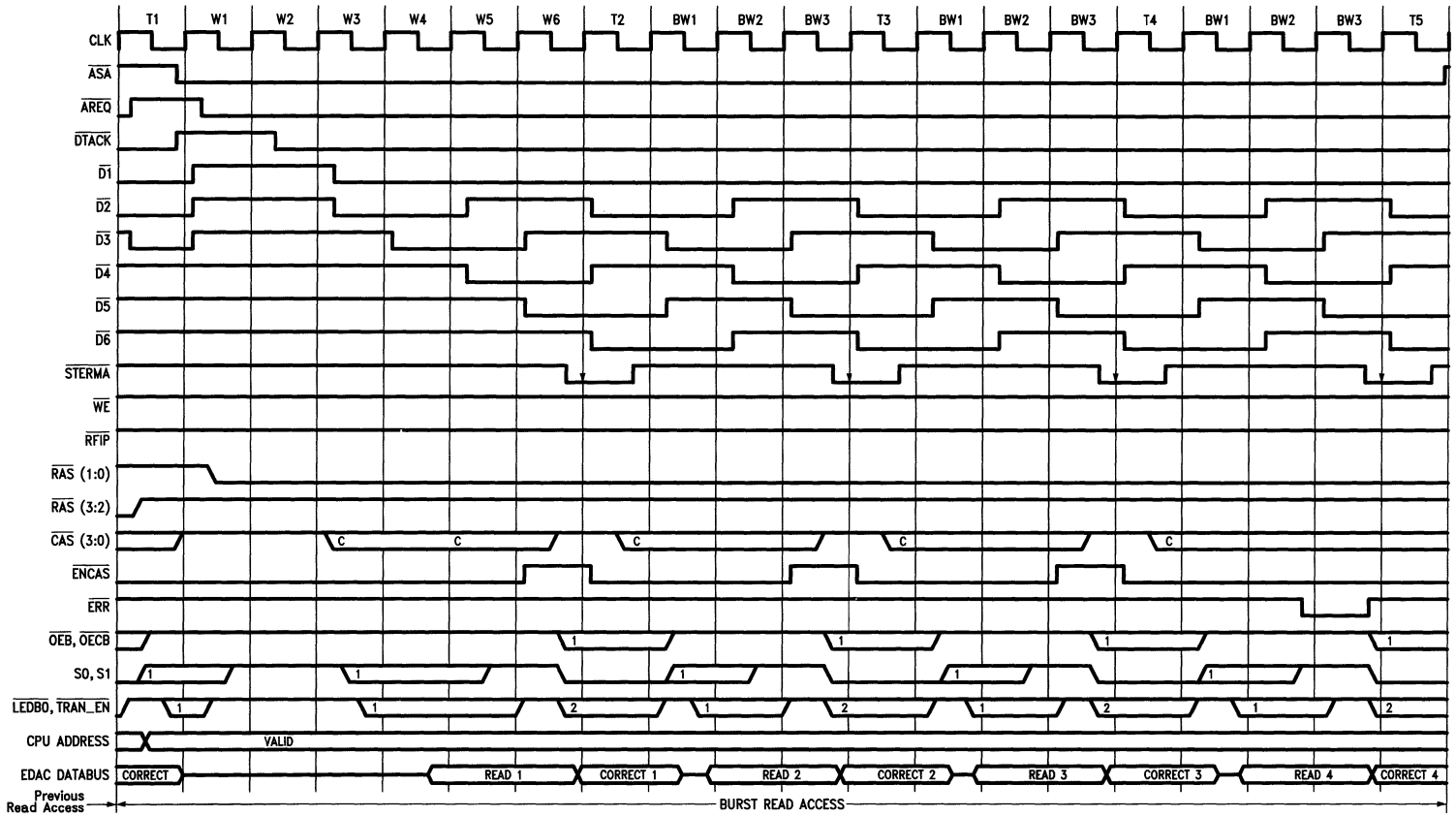
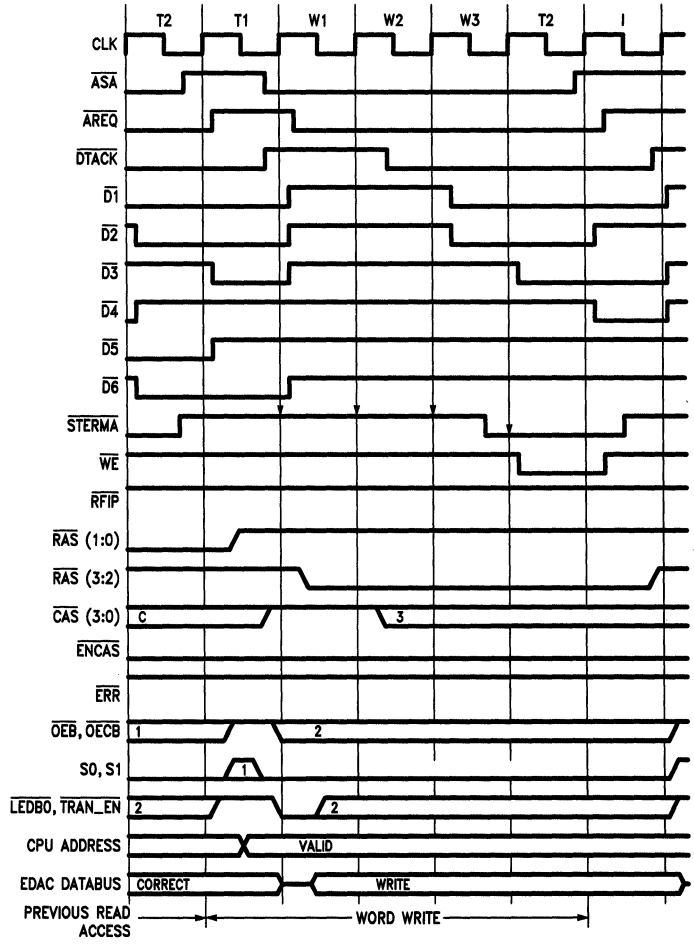
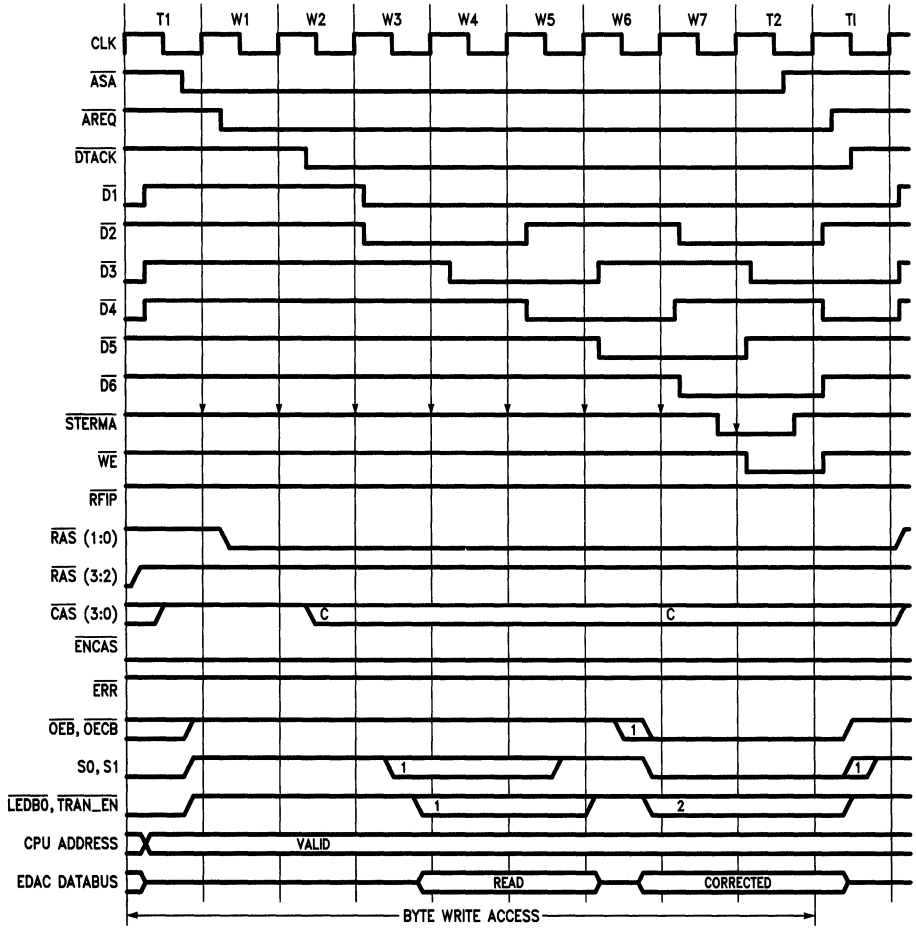


FIGURE 4. 68030 EDAC Burst Read Access Timing



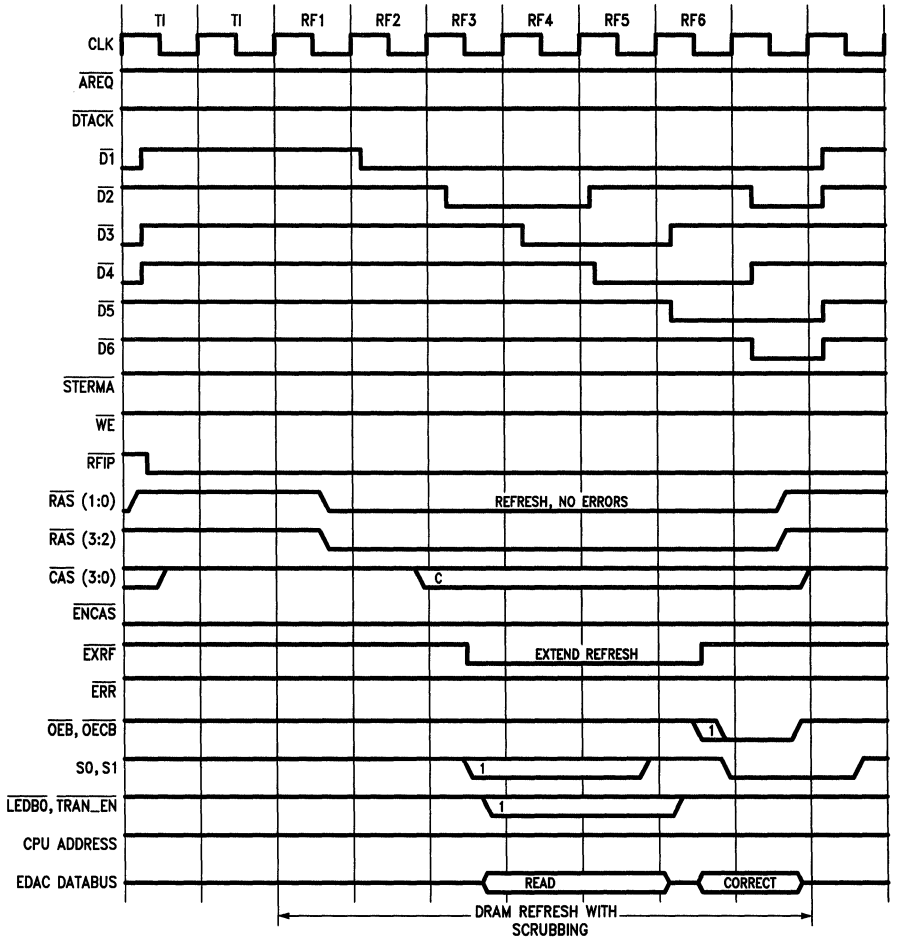
TL/F/9729–5

FIGURE 5. 68030 EDAC Word Write Access Timing



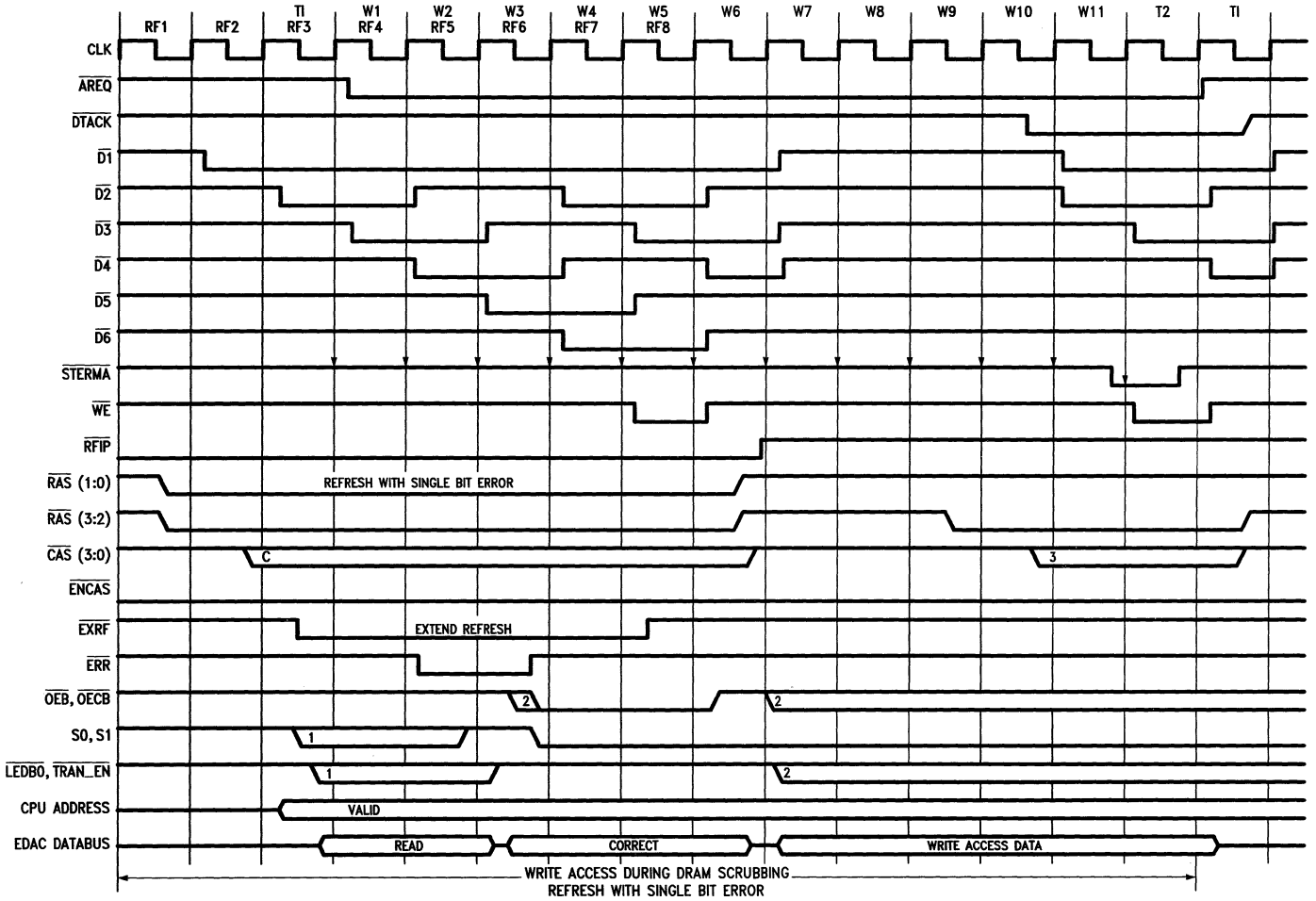
TL/F/9729–6

FIGURE 6. 68030 EDAC Byte Write Access Timing



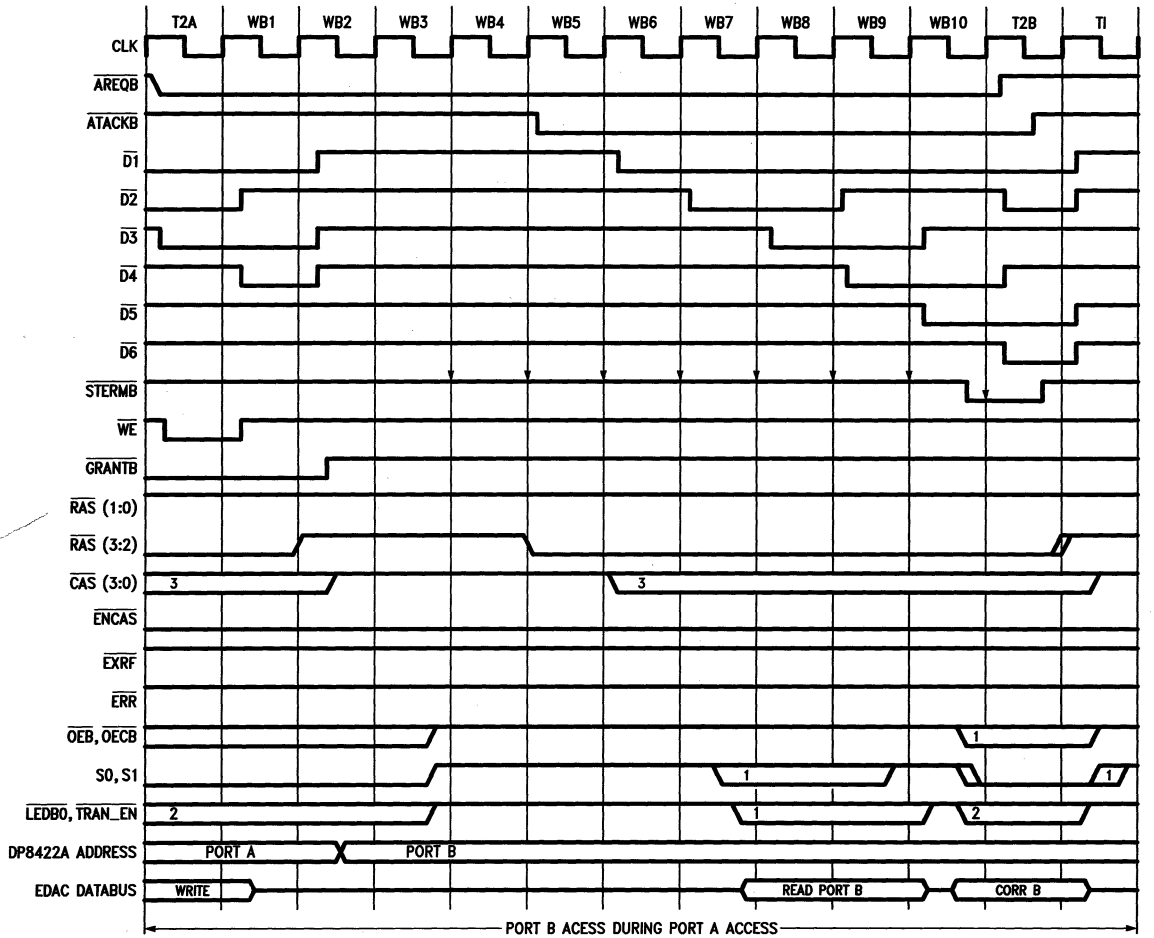
TL/F/9729–7

FIGURE 7. 68030 EDAC DRAM Refresh with Scrubbing



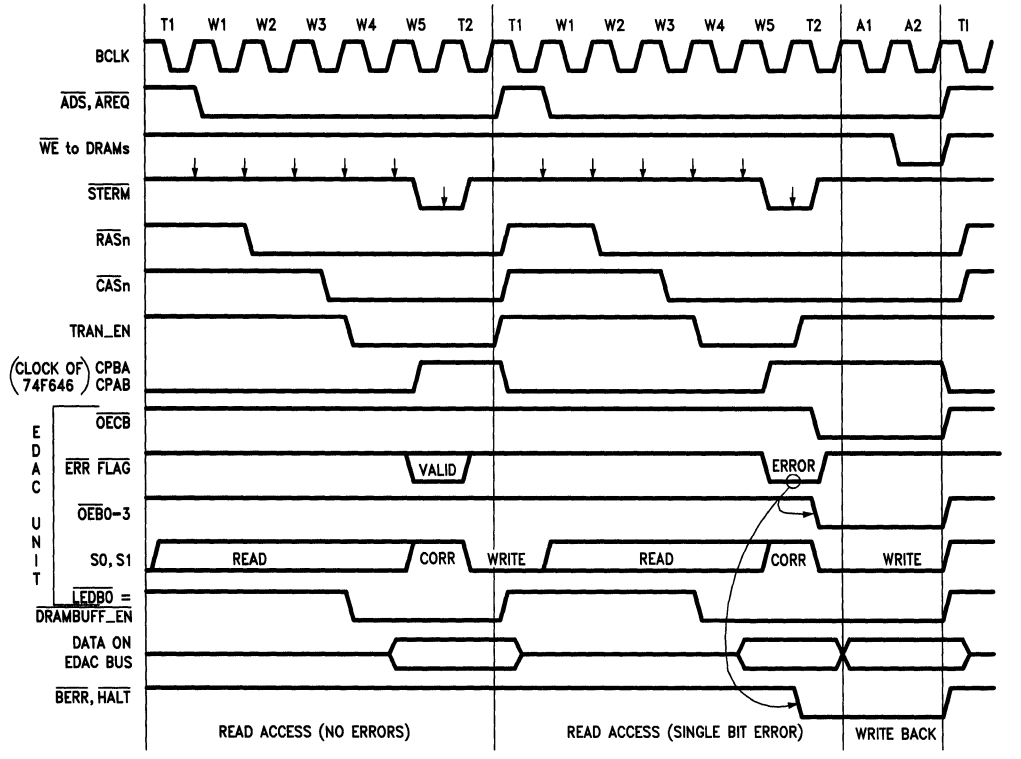
TL/F/9729–8

FIGURE 8. 68030 EDAC Write Access during Refresh Timing



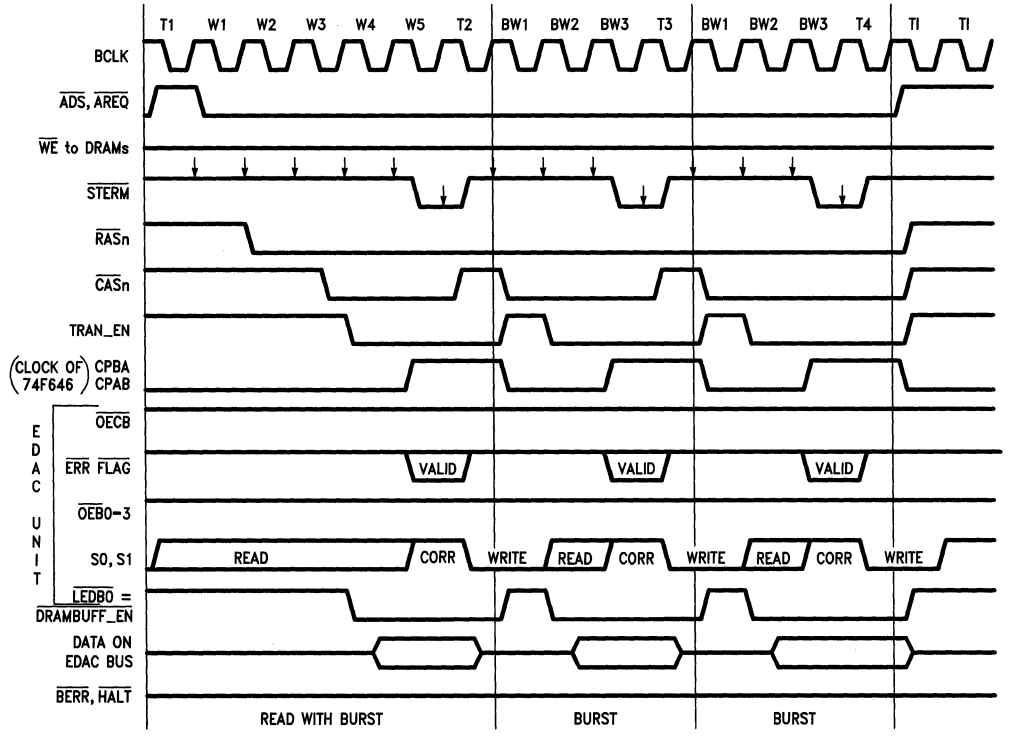
TL/F/9729–9

FIGURE 9. 68030 EDAC Port B Access during Port A Access



TL/F/9729–10

FIGURE 10. 68030 EDAC Error Monitoring Method Using the Asynchronous Late Retry Feature of the 68030



TL/F/9729–11

FIGURE 11. 68030 EDAC Error Monitoring Method Using the Asynchronous Late Retry Feature of the 68030

Interfacing the DP8420A/21A/22A to the 8086/186/88/188 Microprocessor

National Semiconductor
Application Note 544
Webster (Rusty) Meier Jr.

I INTRODUCTION

This application note describes how to interface the 80186 microprocessor to the DP8422A DRAM controller (also applicable to DP8420A/21A). It is assumed that the reader is already familiar with 80186 and the DP8422A modes of operation. This application note will also allow the 8086/88/188 to interface to the DP8420A/21A/22A.

II DESCRIPTION OF DESIGN, 8086/88/186/188 OPERATING AT UP TO 16 MHz (UP TO 12.5 MHz WITH 0 WAIT STATES)

The block diagram of this design is shown driving four banks of DRAM, each bank being 16 bits in width, giving a maximum memory capacity of up to 32 Mbytes (using 4 M-bit × 1 DRAMs).

The memory banks are interleaved on word (16-bit word) boundries. This means that the address bits (A1,2) is tied to the bank select inputs of the DP8422A (B0,1).

Address bit A0 is used, along with Bus High Enable ($\overline{BHE}$), to produce the two byte select data strobes. These byte selects (A0, $\overline{BHE}$) are used in byte reads and writes as well as selects for the transceivers.

This application allows 0 or more wait states to be inserted in normal accesses of the 8086/186/88/188. The number of wait states can be adjusted through the $\overline{WAITIN}$ input of the DP8422A.

The logic shown in this application note forms a complete 8086/186/88/188 memory sub-system, no other logic is needed. This sub-system automatically takes care of:

A. arbitration between Port A, Port B and refreshing the DRAM;

B. the insertion of wait states to the processor (Port A and Port B) when needed (i.e., if $\overline{RAS}$ precharge is needed, refresh is happening during a memory access, the other Port is currently doing an access . . . etc);

C. performing byte writes and reads to the 16-bit words in memory.

If the system uses the 8086/88 the "ALE" output can be directly input to the DP8420A/21A/22A, the 74AS08 "AND" gate and the two 74AS04 inverters on the "ALE" output are not needed.

By using the "output control" pins of some external latches (74AS373's), this application can easily be used in a dual access application. The addresses could be tri-stated through these latches, the write input ($\overline{WIN}$), lock input ($\overline{LOCK}$), and $\overline{ECAS0\text{-}3}$ inputs must also be able to be tri-stated (a 74AS244 could be used for this purpose). By multiplexing the above inputs (through the use of the above parts and similar parts for Port B) the DP8422A can be used in a dual access application. If this design is used in a dual access application the t_{RAC} and t_{CAC} (required $\overline{RAS}$ and $\overline{CAS}$ access time required by the DRAM) will have to be recalculated since the time to $\overline{RAS}$ and $\overline{CAS}$ is longer for the dual access application (see TIMING section of this application note).

III 8086/186/88/188 DESIGN, 10 MHz WITH 0 WAIT STATES DURING NORMAL ACCESSES, PROGRAM MODE BITS

Programming Bits	Description
R0 = 0 R1 = 1	$\overline{RAS}$ low two clocks, $\overline{RAS}$ precharge of two clocks. If more $\overline{RAS}$ precharge is desired the user should program three periods of $\overline{RAS}$ precharge.
R2 = 0 R3 = 0	$\overline{WAIT}$ zero is chosen. $\overline{WAIT}$ follows the access $\overline{RAS}$ low.
R4 = 0 R5 = 0	No WAIT states during burst accesses
R6 = 0	If $\overline{WAIT}$ = 0, add one clock to $\overline{WAIT}$. $\overline{WAITIN}$ may be tied high or low in this application depending upon the number of wait states the user desires to insert into the access
R7 = 0	Select $\overline{WAIT}$
R8 = 1	Non-interleaved Mode
R9 = X	
C0 = X C1 = X C2 = X	Select based upon the input "DELCLK" frequency. Example: if the input clock frequency is 10 MHz then choose C0,1,2 = 1,0,1 (divide by five, this will give a frequency of 2 MHz).
C3 = X	
C4 = 0 C5 = 1	$\overline{RAS}$ banks selected by "B0,1". This mode allows one $\overline{RAS}$
C6 = 1	output to go low during an access, and allows byte writing in 16-bit words.
C7 = 1	Column address setup time of 0 ns.
C8 = 1	Row address hold time of 15 ns
C9 = 1	Delay $\overline{CAS}$ during write accesses to one clock after $\overline{RAS}$ transitions low
B0 = 1	Fall through latches.
B1 = 0	Access mode 0
$\overline{ECAS0}$ = 0	$\overline{CAS}$ not extended beyond $\overline{RAS}$

0 = Program with low voltage level
1 = Program with high voltage level
X = Program with either high or low voltage level (don't care condition)

IV 8086/186/88/188 TIMING CALCULATIONS FOR DESIGN AT 10 MHz WITH NO WAIT STATES DURING NORMAL ACCESSES

1. Minimum ALE high setup time to CLOCK high (DP8422A-20 needs 16 ns, #301a):

   100 ns (one clock period) − 9 ns (maximum delay through two 74AS04S) − 6 ns (74AS08 max delay) = 85 ns

2. Minimum address setup time to CLK high (DP8422A-20 needs 20 ns, #303):

   100 ns (one clock period) − 50 ns (min address valid delay, TCLAV parameter in 80C186 data sheet) − 6 ns (74AS373 max delay) + 1 ns (74ALS04B min delay) = 45 ns

3. Minimum $\overline{CS}$ setup time to clock high (DP8422-20 needs 14 ns, #300):

   45 ns (#2 above) − 10 ns (max 74ALS138 decoder) = 35 ns

4. Determining t_{RAC} during a normal access ($\overline{RAS}$ access time needed by the DRAM):

   200 ns (two clock periods to do the access) − 32 ns (CLK to $\overline{RAS}$ low max, DP8422-20 #307) − 15 ns (8086/186/88/188 data setup time, TDVCL) − 8 ns (74AS245A max delay) − 5 ns (74AS04 max delay, clock skew) = 140 ns

   Therefore the t_{RAC} of the DRAM must be 140 ns or less.

5. Determining t_{CAC} during a normal access ($\overline{CAS}$ access time) and column address access time needed by the DRAM:

   200 ns − 89 ns (CLK to $\overline{CAS}$ low on DP8422A-20, #308a) − 15 ns − 8 ns − 5 ns = 83 ns

   Therefore the t_{CAC} of the DRAM must be 83 ns or less.

6. Minimum SRDY (Synchronous ReaDY) setup time to SYSCLK low (CLK to DP8422A is inverted from SYSCLK), 8086/186/88/188 SRDY input needs 15 ns, TSRYCL:

   100 ns (one clock period) − 39 ns (DP8422A-20 max delay to $\overline{WAIT}$ 0 high after arbitration, parameter #17) = 61 ns

Note: Calculations can be performed for different frequencies, different logic (ALS or CMOS ... etc), or the DP8422A-25, and/or different combinations of wait states by substatuting the appropriate values into the above equations.

V 8086/186/88/188 TIMING CALCULATIONS FOR DESIGN AT 16 MHz WITH ONE WAIT STATE DURING NORMAL ACCESSES (THE $\overline{WAITIN}$ INPUT OF THE DP8422A SHOULD BE TIED LOW)

1. Minimum ALE high setup time to CLOCK high (DP8422A-20 needs 16 ns, #301a):

   62.5 ns (one clock period) − 9 ns (maximum delay through two 74AS04s) − 6 ns (74AS08 max delay) = 47.5 ns

2. Minimum address setup time to CLK high (DP8422A-20 needs 20 ns, #303):

   62.5 ns (one clock period) − 33 ns (min address valid delay, TCLAV parameter in 80C186 data sheet) − 6 ns (74AS373 max delay) + 1 ns (74ALS04B min delay) = 24.5 ns

3. Minimum $\overline{CS}$ setup time to clock high (DP8422A-20 needs 14 ns, #300):

   24.5 ns (#2 above) − 10 ns (max 74ALS138 decoder) = 14.5 ns

4. Determining t_{RAC} during a normal access ($\overline{RAS}$ access time needed by the DRAM):

   182.5 ns (three clock periods to do the access) − 32 ns (CLK to $\overline{RAS}$ low max, DP8422A-20 #307) − 15 ns (8086/186/88/188 data setup time, TDVCL) − 8 ns (74S245A max delay) − 5 ns (74AS04 max delay, clock skew) = 122.5 ns

   Therefore the t_{RAC} of the DRAM must be 122.5 ns or less.

5. Determining t_{CAC} during a normal access ($\overline{CAS}$ access time) and column address access time needed by the DRAM:

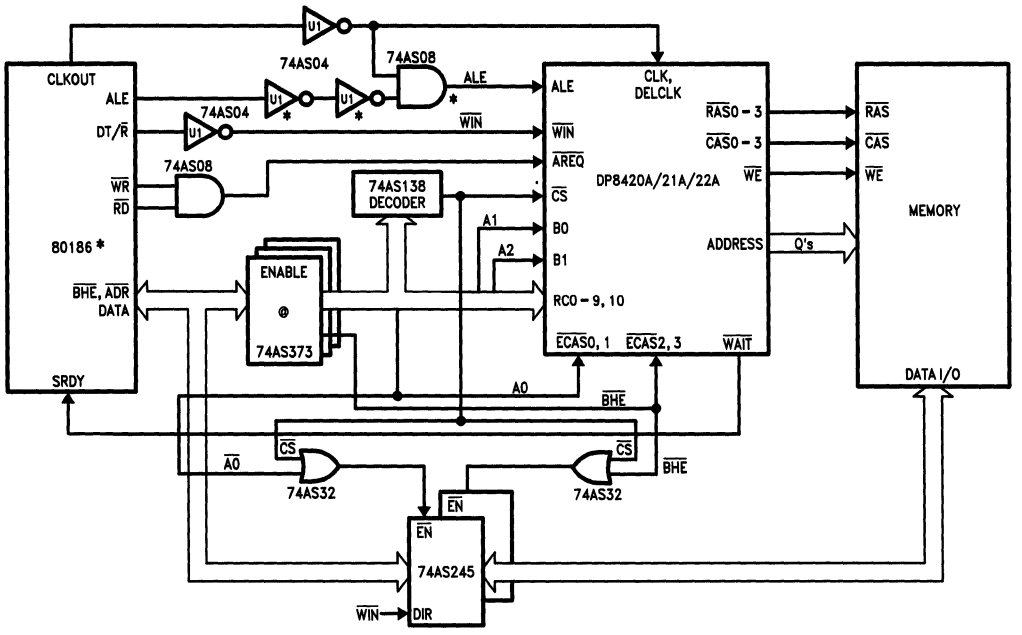
   182.5 ns − 89 ns (CLK to $\overline{CAS}$ low on DP8422A-20, #308a) − 15 ns − 8 ns − 5 ns = 65.6 ns

   Therefore the t_{CAC} of the DRAM must be 65.5 ns or less.

6. Minimum SRDY (Synchronous ReaDY) setup time to SYSCLK low (CLK to DP8422A is inverted from SYSCLK), 8086/186/88/188 SRDY input needs 15 ns, TSRYCL:

   62.5 ns (one clock period) − 39 ns (DP8422A-20 max delay to $\overline{WAIT}$ 1 high, parameter #17) = 23.5 ns

Note: Calculations can be performed for different frequencies, different logic (ALS or CMOS ... etc), the DP8422A-25 and/or different combinations of wait states by substatuting the appropriate values into the above equations.

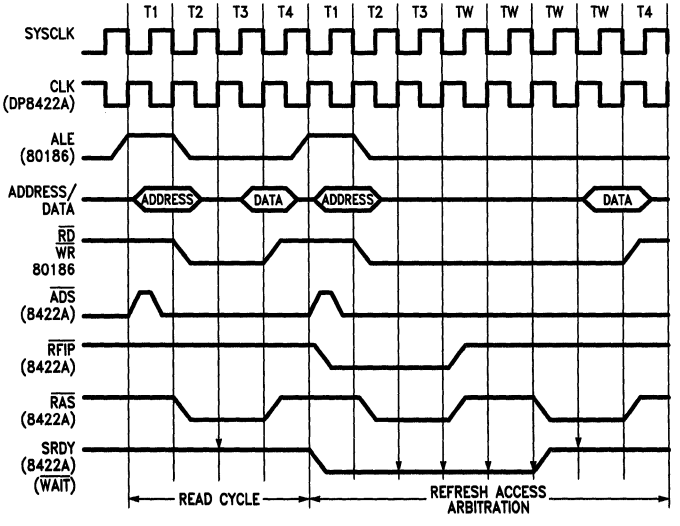


TL/F/9738–1

@May not be needed in all memory applications

*If using the 8086/88 the two inverters (74AS04) and the "AND" gate (74AS08) are not needed, ALE from the 8086/88 can be directly connected to the DP8420A/21A/22A ALE input.

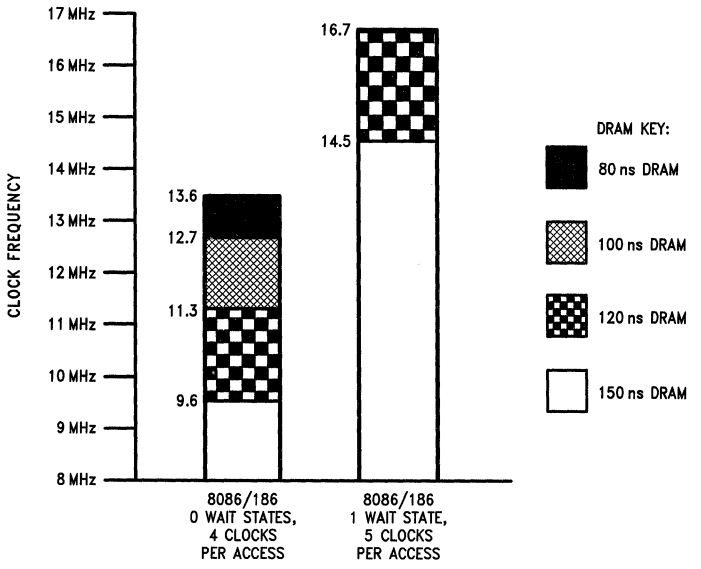
FIGURE 1. 80186 Block Diagram



TL/F/9738–2

FIGURE 2. 80186 Timing

DRAM Speed Vs. Processor Speed, (DRAM Speed References the $\overline{RAS}$ Access Time, t_{RAC}, of the DRAM, using DP8422A-25 Timing Specifications)



TL/F/9738–3

Interfacing the DP8420A/21A/22A to the 80286

National Semiconductor
Application Note 545
Webster (Rusty) Meier Jr. and Joe Tate

INTRODUCTION

This application note describes how to interface the 80286 microprocessor to the DP8422A DRAM controller (also applicable to DP8420A/21A). There are three designs contained within this application note. The designs differ in terms of the maximum allowable frequency of operation. Design #1 can be used up to 16 MHz (80286-8) with one wait state. Design #2 can be used up to 20 MHz (80286-10) with one wait state. Design #3 can be used up to 25 MHz (80286-12) with one wait state. It is assumed that the reader is already familiar with 80286 access cycles and the DP8422A modes of operation.

DESCRIPTION OF DESIGN #1, ALLOWS UP TO 16 MHz OPERATION (CLOCK OUTPUT OF THE 82284) WITH NO WAIT STATES USING THE 80286-8

Design #1 (see *Figures 1 and 2*) consists of the DP8422A DRAM controller and several logic gates. These parts interface to the 80286 as shown in the block diagram. It accommodates two banks of DRAM, each bank being 16 bits in width, giving a maximum memory capacity of 16 Mbytes (using 4M-bit X 1 DRAMs). By choosing a different $\overline{RAS}$ and $\overline{CAS}$ configuration mode (see programming mode bits section of 8422A data sheet) this application could support 4 banks of DRAM, giving a memory capacity of 32 Mbytes (using 4M-bit X 1 DRAMs).

The memory banks are interleaved. This means that the least significant address bit (A1) is tied to the bank select input of the DP8422A (B1). Because the majority of accesses made by the 80286 will be sequential in nature, one memory bank can be precharging ($\overline{RAS}$ precharge) while the other bank is being accessed. The interleaved memory system has higher system performance than a non-interleaved memory system. In a non-interleaved memory system, each sequential access will generally be to the same memory bank thereby requiring extra wait states to be inserted into the CPU access cycles to allow for the $\overline{RAS}$ precharge time.

The user can choose non-address pipelined mode for this design as long as the parameter "$\overline{AREQ}$ negated to CLK high minimum to guarantee tASR = 0 ns" is guaranteed (45 ns minimum for the 8422A-20, 39 ns for the 8422A-25). At 16 MHz, the user must choose address pipelined mode since it is not possible to meet the above parameter (62.5 ns one clock − 25 ns $\overline{MRDC}$,$\overline{MWRC}$ max valid − 5.5 ns 74AS08 max delay + 1 ns min 74AS00 $\overline{CLOCK}$ delay = 33.5 ns which is less than the 39 ns the DP8422A-25 needs). When using the DP8422A in address pipelined mode, the DRAMs chosen should need a minimum column address hold time of 32 ns or less.

The logic shown in this application note forms a complete 80286 memory sub-system, no other logic is needed. This sub-system automatically takes care of:

A. Arbitration between Port A, Port B, and refreshing the DRAM;
B. The insertion of wait states to the processor (Port A and Port B) when needed (i.e., if $\overline{RAS}$ precharge is needed, refresh is occurring during a memory access, the other Port is currently doing an access . . . etc.);
C. Performing byte writes and reads to the 16-bit words in memory.

Since the $\overline{WE}$ output of the DP8422A becomes refresh request ($\overline{RFRQ}$) if the chip is programmed in address pipelining mode, the $\overline{WIN}$ signal was buffered to provide $\overline{WE}$ to the DRAMs.

The gates labeled "U1" should both be in the same package (74AS00) so that their delays cancel, see the TIMING section for how these delays cancel.

The ready logic can be made faster by programming $\overline{DTACK0}$ into the DP8422A and running this through a fast bipolar flip-flop clocked by $\overline{CLOCK}$.

By making use of the enable input on the 74AS373 latch, this application can easily be used in a dual access application. The addresses and chip select are TRI-STATE® through this latch, the write input ($\overline{WIN}$), lock input ($\overline{LOCK}$), and $\overline{ECAS0-3}$ inputs must also be able to be TRI-STATE (a 74AS244 could be used for this purpose). By multiplexing the above inputs, (through the use of the above parts and similar parts for Port B) the DP8422A can be used in a dual access application. All the timing (see TIMING section of this application note) will remain the same whether single or dual accessing is implemented.

DESCRIPTION OF DESIGN #2, ALLOWS UP TO 20 MHz OPERATION (CLOCK OUTPUT OF THE 82284) WITH NO WAIT STATES USING THE 80286-10

Design #2 (see *Figures 3, 4, 5*) is basically the same as Design #1 except for the following changes:

A. The circuit that produces $\overline{SRDY}$, the gate signal for the 74AS373 transparent latches, and $\overline{AREQ}$ has been changed to using $\overline{DTACK0}$, several 74AS374 flip-flops, and some logic gates. This was needed for speed in producing the gating signal of the 74AS373 (so as to get adequate address and chip select setup time),
B. The output "D1" which is used to produce the $\overline{AREQ}$ input was gated with $\overline{CS}$. This was done to guarantee that the $\overline{DTACK2}$ output becomes defined after power up,
C. This design will work at 20 MHz,
D. This design has zero wait states inserted in normal sequential accesses, multiple wait states may be inserted on multiple accesses to the same memory bank, during DRAM refreshing, or during accesses from Port B if dual accessing is used (DP8422A only).

In the 80286 READY LOGIC, the 74AS374 flip-flop that produces the D2 output has some gating at its inputs. This gating is used to synchronize the D2 output to the 80286 PCLK so as to end the access at the correct time.

The user can choose non-address pipelined mode for this design as long as the parameter "$\overline{AREQ}$ negated to CLK high minimum to guarantee tASR = 0 ns" is guaranteed

(45 ns minimum for the 8422A-20, 39 ns for the 8422A-25). At 20 MHz, the user should choose address pipelined mode since it is not possible to meet the above parameter (50 ns one clock − 8 ns 74AS374 max delay − 4.5 ns 74AS00 max delay + 1 ns min 74AS00 $\overline{CLOCK}$ delay = 38.5 ns which is less than the 39 ns the DP8422A-25 needs). The user should also keep in mind that when using the DP8422A in address pipelined mode the DRAMs chosen need a minimum column address hold time of 32 ns.

DESCRIPTION OF DESIGN #3, ALLOWS UP TO 25 MHz OPERATION (CLOCK OUTPUT OF THE 82284) WITH ONE WAIT STATE USING THE 80286-12

Design #3 (see *Figures 6, 7, 8*) is very similar to Design #2 except for the following changes:

A. The circuit that produces $\overline{SRDY}$, the gate signal for the 74AS373 transparent latches, and $\overline{AREQ}$ has been changed to use $\overline{DTACK2}$, a 74AS175 flip-flop, and some logic gates. This was needed because the 84288-12 was not known to be available, and also for speed in producing the gating signal of the 74AS373,

B. The $\overline{AREQ}$ input was gated with $\overline{CS}$ using gate "U2". This was done to guarantee that the $\overline{DTACK2}$ output becomes defined after power up,

C. This design will work at 25 MHz and possibly beyond, if the 80286 is ever produced at faster speeds (see the TIMING section for Design #3),

D. This design has one wait state inserted in normal sequential accesses, multiple wait states may be inserted on multiple accesses to the same memory bank, during DRAM refreshing, or during accesses from Port B if dual accessing is used (DP8422A only).

The user can choose non-address pipelined mode for this design as long as the parameter "$\overline{AREQ}$ negated to CLK high minimum to guarantee tASR = 0 ns" is guaranteed (45 ns minimum for the 8422A-20, 39 ns for the 8422A-25). At 25 MHz, the user must choose address pipelined mode since it is not possible to meet the above parameter (40 ns one clock − 7.5 ns 74AS175 max delay − 1 ns min 74AS00 $\overline{CLOCK}$ delay = 33.5 ns which is less than the 39 ns the DP8422A-25 needs). The user should also keep in mind that when using the DP8422A in address pipelined mode the DRAMs chosen need a minimum column address hold time of 32 ns.

In the 80286 READY LOGIC, the 74AS175 flip-flop that produces the D4 output has some gating at its inputs. This gating is used to synchronize the D4 output to the 80286 PCLK so as to end the access at the correct time.

80286 DESIGN #1, UP TO 16 MHz WITH NO WAIT STATES, PROGRAMMING MODE BITS

Programming Bits	Description
R0 = 0	$\overline{RAS}$ low two clocks, $\overline{RAS}$ precharge of
R1 = 1	two clocks. It should be noted that the user should choose R0,1 = 11 when operating above 16 MHz to allow enough $\overline{RAS}$ precharge time
R2 = 1	$\overline{DTACK}$ low one clock from $\overline{RAS}$ low
R3 = 0	
R4 = 0	No WAIT states during burst accesses
R5 = 0	
R6 = 1	If $\overline{WAITIN}$ = 0, add two clocks to $\overline{DTACK}$
R7 = 1	Select $\overline{DTACK}$
R8 = X	The user may choose address pipelined mode (R8 = 0) remember to choose DRAMs with column address hold times of 32 ns or less, or non-address pipelined mode (R8 = 1), at clock frequencies below 16 MHz.
R9 = X	
C0 = X	Select based upon the input clock
C1 = X	frequency. Example: if the input clock
C2 = X	frequency is 12 MHz then choose C0,1,2 = 0,0,1 (divide by six, this will give a frequency of 2 MHz).
C3 = X	
C4 = 1	$\overline{RAS}$ and $\overline{CAS}$ groups selected by "B1".
C5 = 1	This mode allows two $\overline{RAS}$ and two
C6 = 1	$\overline{CAS}$ outputs to go low during an access, and allows byte writing in 16-bit words.
C7 = 1	Column address setup time of 0 ns.
C8 = 1	Row address hold time of 15 ns.
C9 = 1	Delay $\overline{CAS}$ during write accesses to one clock after $\overline{RAS}$ transitions low
B0 = 1	Fall-thru latches
B1 = 0	Access mode 0
$\overline{ECAS0}$ = 0	Non-extend $\overline{CAS}$ mode

0 = Program with low voltage level

1 = Program with high voltage level

X = Program with either high or low voltage level (don't care condition)

80286 DESIGN #2, UP TO 20 MHz WITH NO WAIT STATES, PROGRAMMING MODE BITS

Programming Bits	Description
R0 = 1	$\overline{RAS}$ low four clocks, $\overline{RAS}$ precharge of three clocks
R1 = 1	
R2 = 0	$\overline{DTACK}$ low from $\overline{RAS}$ low
R3 = 0	
R4 = 0	No WAIT states during burst accesses
R5 = 0	
R6 = 0	If $\overline{WAITIN}$ = 0, add one clock to $\overline{DTACK}$. Since we are not using the $\overline{WAITIN}$ input it should be tied high on the DP8422A.
R7 = 1	Select $\overline{DTACK}$
R8 = X	The user may choose address pipelined mode (R8 = 0), remember to choose DRAMs with column address hold times of 32 ns or less or non-address pipelined mode (R8 = 1), at clock frequencies below 20 MHz
R9 = X	
C0 = X	Select based upon the input clock
C1 = X	frequency. Example: if the input clock
C2 = X	frequency is 16 MHz then choose C0,1,2 = 0,1,0 (divide by eight, this will give a frequency of 2 MHz).
C3 = X	
C4 = 1	$\overline{RAS}$ and $\overline{CAS}$ groups selected by "B1".
C5 = 1	This mode allows two $\overline{RAS}$ and $\overline{CAS}$
C6 = 1	outputs to go low during an access, and allows byte writing in 16-bit words.
C7 = 1	Column address setup time of 0 ns.
C8 = 1	Row address hold time of 15 ns.
C9 = 1	Delay $\overline{CAS}$ during write accesses to one clock after $\overline{RAS}$ transitions low
B0 = 1	Fall-thru latches
B1 = 0	Access mode 0
$\overline{ECAS}$0 = 0	Non-extend $\overline{CAS}$ mode

0 = Program with low voltage level

1 = Program with high voltage level

X = Program with either high or low voltage level (don't care condition)

80286 DESIGN #3, UP TO 25 MHz WITH ONE WAIT STATE, PROGRAMMING MODE BITS

Programming Bits	Description
R0 = 1	$\overline{RAS}$ low four clocks, $\overline{RAS}$ precharge of three clocks
R1 = 1	
R2 = 1	$\overline{DTACK}$ low one clock from $\overline{RAS}$ low
R3 = 0	
R4 = 0	No WAIT states during burst accesses
R5 = 0	
R6 = 0	If $\overline{WAITIN}$ = 0, add one clock to $\overline{DTACK}$. Since we are using $\overline{DTACK2}$ the $\overline{WAITIN}$ input should be tied low on the DP8422A.
R7 = 1	Select $\overline{DTACK}$
R8 = X	The user must choose address pipelined mode (R8 = 0), at clock frequencies above 20 MHz. Also remember to choose DRAMs with column address hold times of 32 ns or less or non-address pipelined mode (R8 = 1), at clock frequencies below 20 MHz
R9 = X	
C0 = X	Select based upon the input clock
C1 = X	frequency. Example: if the input clock
C2 = X	frequency is 12 MHz then choose C0,1,2 = 0,0,1 (divide by six, this will give a frequency of 2 MHz). For a CPU frequency of 24 MHz the clock could be divided by two initially to give a 12 MHz input to the DELCLK input of the DP8422A.
C3 = X	
C4 = 1	$\overline{RAS}$ and $\overline{CAS}$ groups selected by "B1".
C5 = 0	This mode allows two $\overline{RAS}$ and two
C6 = 1	$\overline{CAS}$ outputs to go low during an access, and allows byte writing in 16-bit words.
C7 = 1	Column address setup time of 0 ns.
C8 = 1	Row address hold time of 15 ns.
C9 = 1	Delay $\overline{CAS}$ during write accesses to one clock after $\overline{RAS}$ transitions low
B0 = 1	Fall-thru latches
B1 = 0	Access mode 0
$\overline{ECAS}$0 = 0	Non-extend $\overline{CAS}$ mode

0 = Program with low voltage level

1 = Program with high voltage level

X = Program with either high or low voltage level (don't care condition)

80286, DESIGN #1, ACCESS MODE 0, 16 MHz TIMING (80286-8) WITH NO WAIT STATES DURING SEQUENTIAL ACCESSES

1. Maximum time to address valid:

   $\overline{MRDC}$, $\overline{MWTC}$ make 74AS373 fall-thru at 25 ns (max) from CLOCK low + 5.5 ns (74AS08) + 11.5 ns (74AS373 enable time) = 42 ns

2a. Maximum time to ALE high from CLOCK:

   40 ns (max status valid) + 4.5 ns (74AS00) = 44.5 ns

2b. Maximum time to ALE high from $\overline{CLOCK}$ (DP8422A CLK):

   40 ns (max status valid) + 0 ns (74AS00 delays of $\overline{CLOCK}$ and ALE cancel out) = 40 ns

3. Minimum ALE high setup time to $\overline{CLOCK}$ high (DP8422A-20 needs 16 ns):

   62.5 ns (one clock period) − 40 ns (#2b, the 74AS00 U1 delays in the $\overline{CLOCK}$ and the ALE path will cancel out) = 22.5 ns

4. Minimum address setup time to $\overline{CLOCK}$ high (DP8422A-20 needs 20 ns):

   62.5 ns (one clock period) + 1 ns (min 74AS00 delay, $\overline{CLOCK}$) − 42 ns (#1) = 21.5 ns

5a. Maximum $\overline{CS}$ valid time from $\overline{CLOCK}$ high:

   60 ns (maximum address valid from phase two of previous clock) + 22 ns (maximum prop delay of 74ALS138) − 1 ns (minimum 74AS00 $\overline{CLOCK}$ delay) = 81 ns,

   THIS IS EQUAL TO 18.5 NS FROM PHASE ONE OF THE CURRENT "TS" INVERTED CLOCK ($\overline{CLOCK}$, 81 ns − 62.5 ns = 18.5 ns).

5b. Maximum time to active high gating signal to 74AS373 transparent latch from $\overline{CLOCK}$:

   25 ns ($\overline{MRDC}$, $\overline{MWRC}$ maximum delay to inactive) + 5.5 ns (maximum 74AS08 delay) − 1 ns (minimum 74AS00 $\overline{CLOCK}$ delay) = 29.5 ns,

   AS CAN BE SEEN $\overline{CS}$ PRECEDES THIS ACTIVE HIGH GATING SIGNAL, THEREFORE THE MAXIMUM TIME TO THE GATING SIGNAL DETERMINES THE MAXIMUM TIME TO A VALID CHIP SELECT.

5c. Minimum $\overline{CS}$ setup time to $\overline{CLOCK}$ high (DP8422A-20 needs 14 ns):

   62.5 ns (one clock period) − 29.5 ns (#5b) − 11.5 ns (maximum enable time to valid output for 74AS373) = 21.5 ns

6. Determining tRAC ($\overline{RAS}$ access time needed by the DRAM):

   250 ns (TS + TC) − 62.5 ns (one clock) − 4.5 ns (74AS04 delay) − 10 ns (data setup time) − 7.5 ns (74AS245) − 32 ns (CLK to $\overline{RAS}$ low on DP8422A-25) = 133.5 ns

   Therefore the tRAC of the DRAM must be 133.5 ns or less.

7. Determining tCAC ($\overline{CAS}$ access time) and column address access time needed by the DRAM:

   250 ns − 62.5 − 4.5 − 10 ns − 7.5 ns − 89 ns (CLK to $\overline{CAS}$ low on DP8422A-25) = 76.5 ns

   Therefore the tCAC of the DRAM must be 76.5 ns or less.

8. Maximum time to $\overline{DTACK1}$ low:

   62.5 ns (One clock) + 4.5 ns (max 74AS04 $\overline{CLOCK}$ delay) +41 ns ($\overline{DTACK0}$ low from CLK high) = 108 ns

9. Minimum $\overline{DTACK1}$ setup time to $\overline{SRDY}$ being sampled (15 ns is needed by the 80286):

   125 ns (two clock periods) − 108 ns (#8) = 17 ns

*****IF FASTER SPEEDS ARE DESIRED THE USER CAN USE THE DP8422A-25.**

80286, DESIGN #2, ACCESS MODE 0, 20 MHz TIMING (80286-10) WITH ZERO WAIT STATES DURING SEQUENTIAL ACCESSES

1. Maximum time to address valid:

   The "G" input makes the 74AS373 fall-thru at 8 ns (max) from $\overline{CLOCK}$ high (8 ns max delay of 74AS374 low to high) + 11.5 ns (74AS373 enable time) = 19.5 ns

2a. Maximum time to ALE high from CLOCK:

   28 ns (max status valid) + 4.5 ns (74AS00) = 32.5 ns

2b. Maximum time to ALE high from $\overline{CLOCK}$ (DP8422A CLK):

   28 ns (max status valid) + 0 ns (74AS00 delays of $\overline{CLOCK}$ and ALE cancel out) = 28 ns

3. Minimum ALE high setup time to $\overline{CLOCK}$ high (DP8422A-25 needs 15 ns):

   50 ns (one clock period) − 28 ns (#2b, the 74AS00 U1 delays in the $\overline{CLOCK}$ and the ALE path will cancel out) = 22 ns

4. Minimum address setup time to $\overline{CLOCK}$ high (DP8422A-25 needs 18 ns):

   50 ns (one clock period) − 19.5 ns (#1) = 30.5 ns

5a. Maximum $\overline{CS}$ valid time from $\overline{CLOCK}$ high:

   47 ns (maximum address valid from phase two of previous clock) + 22 ns (maximum prop delay of 74ALS138) − 1 ns (minimum 74AS00 $\overline{CLOCK}$ delay) = 68 ns,

   THIS IS EQUAL TO 18 NS FROM PHASE ONE OF THE CURRENT "TS" INVERTED CLOCK ($\overline{CLOCK}$).

5b. Maximum time to $\overline{AREQ}$ (active high gating signal to 74AS373 transparent latch from $\overline{CLOCK}$:

   8 ns (max delay of gate (D1) output of 74AS374 low to high from $\overline{CLOCK}$) + 4.5 ns (74AS00, $\overline{AREQ}$ is output) = 12.5 ns

   AS CAN BE SEEN $\overline{CS}$ SUCCEEDS THIS ACTIVE HIGH GATING SIGNAL, THEREFORE THE MAXIMUM TIME TO CHIP SELECT DETERMINES THE MAXIMUM TIME TO A VALID LATCHED CHIP SELECT.

5c. Minimum $\overline{CS}$ setup time to $\overline{CLOCK}$ high (DP8422A-25 needs 13 ns):

   50 ns (one clock period) − 18 ns (#5a) − 11.5 ns (maximum enable time to valid output for 74AS373) = 20.5 ns

6. Determining tRAC ($\overline{RAS}$ access time needed by the DRAM):

   200 ns (TS + TC) − 50 ns (one clock) − 4.5 ns (74AS04 delay, $\overline{CLOCK}$) − 8 ns (data setup time) − 7.5 ns (74AS245) − 26 ns (CLK to $\overline{RAS}$ low on DP8422A-25) = 104 ns

   Therefore the tRAC of the DRAM must be 104 ns or less.

7. Determining tCAC ($\overline{CAS}$ access time) and column address access time needed by the DRAM:

   200 ns − 50 − 4.5 − 8 ns − 7.5 ns − 79 ns (CLK to $\overline{CAS}$ low on DP8422A-25) = 51 ns

   Therefore the tCAC of the DRAM must be 51 ns or less.

8. Minimum setup time of D0 low (from $\overline{DTACK0}$) to $\overline{CLOCK}$ (74AS374 needs 3 ns):

   50 ns (one clock) − 33 ns ($\overline{DTACK0}$ low from CLK high) − 4.5 ns (max delay of 74AS02) = 12.5 ns

9. Minimum PCLK setup to $\overline{CLOCK}$ (rising edge of 74AS374, needs 3 ns):

   50 ns (one clock) − 35 ns (max PCLK delay) − 5 ns (max 74AS04 delay) − 4.5 ns (max delay of 74AS02) = 5.5 ns

10. Minimum $\overline{SRDY}$ setup time to clock where it is sampled (15 ns is needed by the 80286):

    50 ns (one clock period) − 8 ns (max delay low to high of "Q" output of 74AS374) − 4.5 ns (max delay of 74AS00) − 4.5 ns (max delay of 74AS00, $\overline{CLOCK}$) = 33 ns

80286, DESIGN #3, ACCESS MODE 0, 25 MHz TIMING (80286-12) WITH ONE WAIT STATE DURING SEQUENTIAL ACCESSES

1. Maximum time to address valid:

   The "G" input makes the 74AS373 fall-thru at 7.5 ns (max) from $\overline{CLOCK}$ high (7.5 ns max delay of 74AS175 low to high) + 11.5 ns (74AS373 enable time) = 19 ns

2a. Maximum time to ALE high from CLOCK:

   22 ns (max status valid) + 4.5 ns (74AS00) = 26.5 ns

2b. Maximum time to ALE high from $\overline{CLOCK}$ (DP8422A CLK):

   22 ns (max status valid) + 0 ns (74AS00 delays of $\overline{CLOCK}$ and ALE cancel out) = 22 ns

3. Minimum ALE high setup time to $\overline{CLOCK}$ high (DP8422A-25 needs 15 ns):

   40 ns (one clock period) − 22 ns (#2b, the 74AS00 U1 delays in the $\overline{CLOCK}$ and the ALE path will cancel out) = 18 ns

4. Minimum address setup time to $\overline{CLOCK}$ high (DP8422A-25 needs 18 ns):

   40 ns (one clock period) − 19 ns (#1) = 21 ns

5a. Maximum $\overline{CS}$ valid time from $\overline{CLOCK}$ high:

   37 ns (maximum address valid from phase two of previous clock) + 14 ns (maximum prop delay of 74ALS139) − 1 ns (minimum 74AS00 $\overline{CLOCK}$ delay) = 50 ns,

   THIS IS EQUAL TO 10 NS FROM PHASE ONE OF THE CURRENT "TS" INVERTED CLOCK ($\overline{CLOCK}$).

5b. Maximum time to active high gating signal to 74AS373 transparent latch from $\overline{CLOCK}$:

   7.5 ns (max delay of gate ($\overline{D3}$) output of 74AS175 low to high from $\overline{CLOCK}$)

   AS CAN BE SEEN $\overline{CS}$ SUCCEEDS THE ACTIVE HIGH GATING SIGNAL, THEREFORE THE MAXIMUM TIME TO $\overline{CS}$ VALID DETERMINES THE MAXIMUM TIME TO A VALID CHIP SELECT.

5c. Minimum $\overline{CS}$ setup time to $\overline{CLOCK}$ high (DP8422A-25 needs 13 ns):

   40 ns (one clock period) − 10 ns (#5a) − 11.5 ns (maximum enable time to valid output for 74AS373) = 18.5 ns

6. Determining tRAC ($\overline{RAS}$ access time needed by the DRAM):

   240 ns (TS + TC + TC) − 40 ns (one clock) − 4.5 ns (74AS04 delay, $\overline{CLOCK}$) − 8 ns (data setup time) − 7.5 ns (74AS245) − 26 ns (CLK to $\overline{RAS}$ low on DP8422A-25) = 154 ns

   Therefore the tRAC of the DRAM must be 154 ns or less.

7. Determining tCAC ($\overline{CAS}$ access time) and column address access time needed by the DRAM:

   240 ns − 40 − 4.5 − 8 ns − 7.5 ns − 79 ns (CLK to $\overline{CAS}$ low on DP8422A-25) = 101 ns

   Therefore the tCAC of the DRAM must be 101 ns or less.

8. Minimum setup time of $\overline{D2}$ low (from $\overline{DTACK2}$) to $\overline{CLOCK}$ (74AS175 needs 3 ns):

   40 ns (one clock) − 28 ns ($\overline{DTACK2}$ low from CLK high) − 4.5 ns (max delay of 74AS02) = 7.5 ns

9. Minimum $\overline{SRDY}$ setup time to clock where it is sampled (15 ns is needed by the 80286):

   40 ns (one clock period) − 10 ns (max delay high to low of $\overline{Q}$ output of 74AS175) − 4.5 ns (max delay of 74AS00, $\overline{CLOCK}$) = 25.5 ns

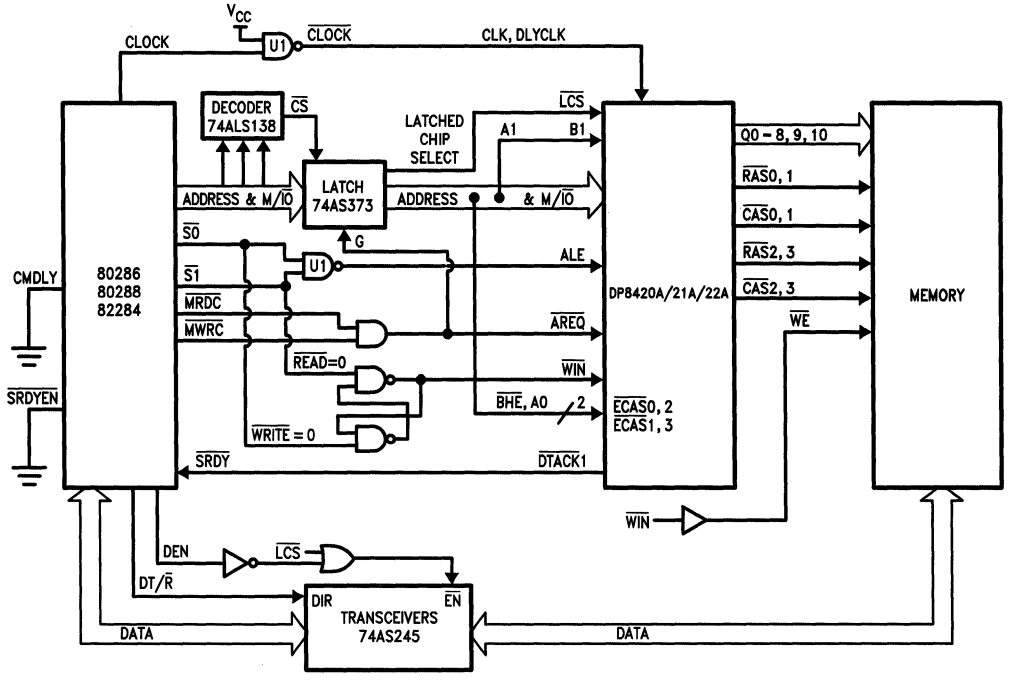


FIGURE 1. 80286 Block Diagram of Design # 1, Mode 0, up to 16 MHz

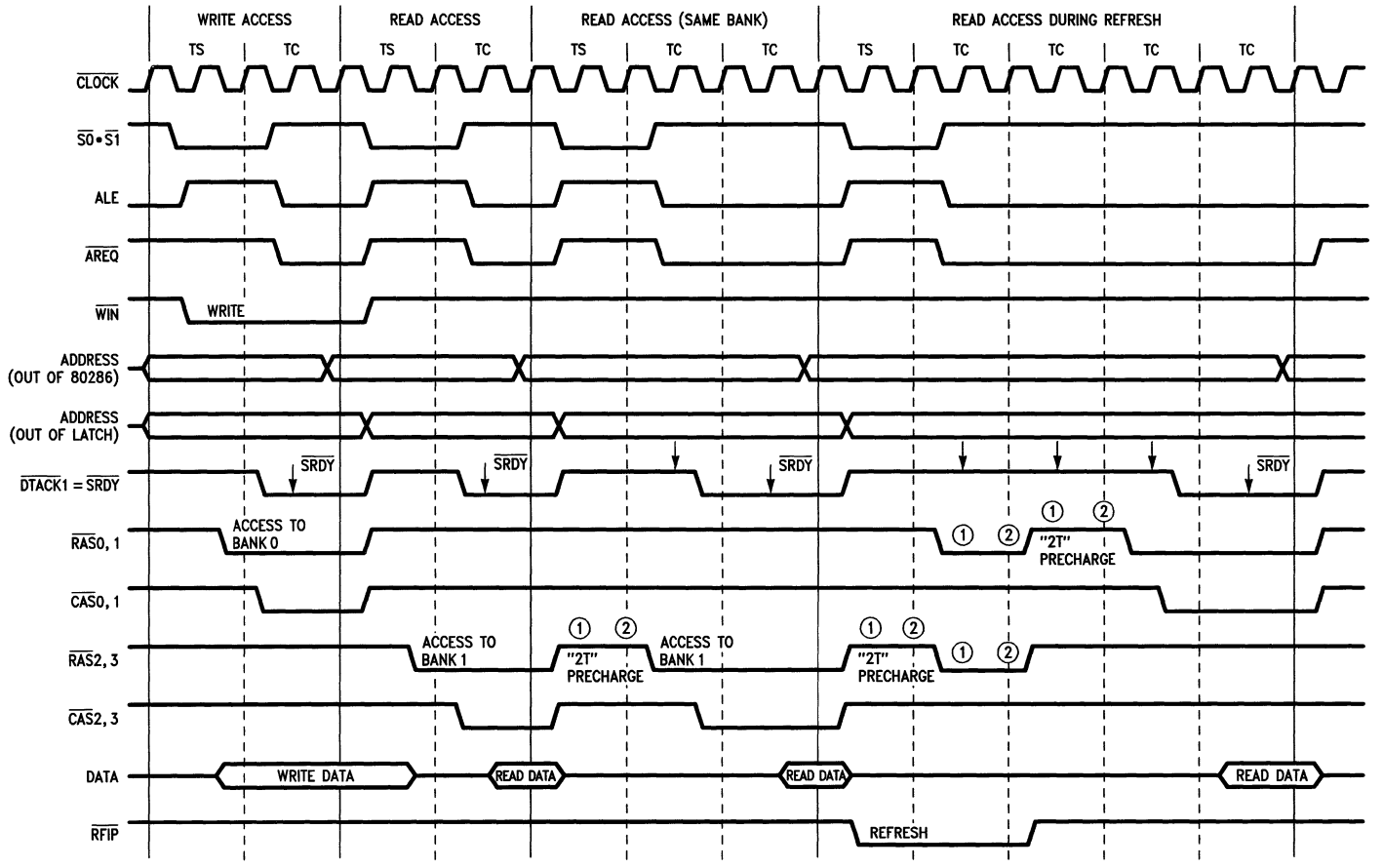
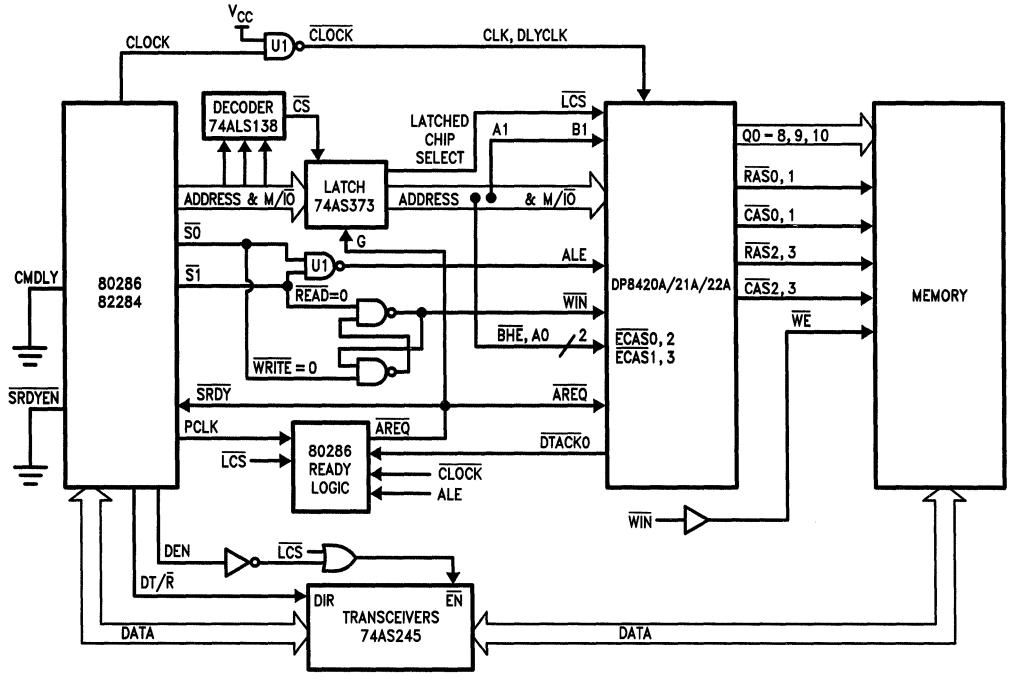
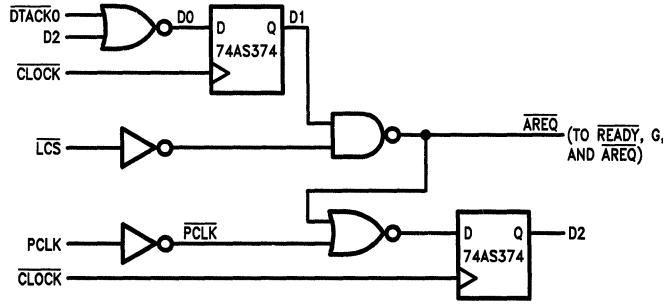


FIGURE 2. 80286 Design # 1 Timing, up to 16 MHz (80286-8), No Wait States



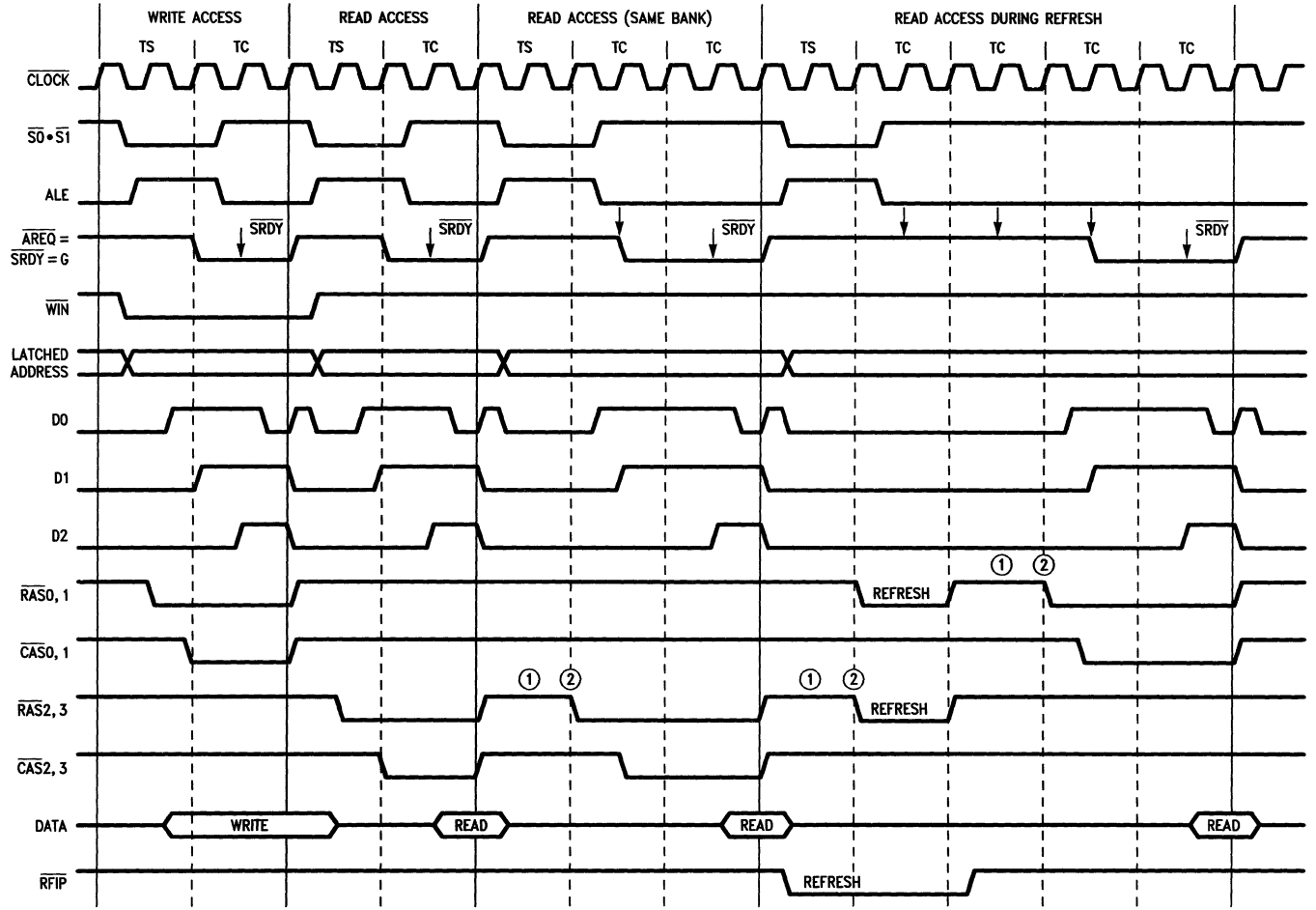
TL/F/9739–4

FIGURE 3. 80286 Block Diagram of Design #2, Mode 0, up to 20 MHz



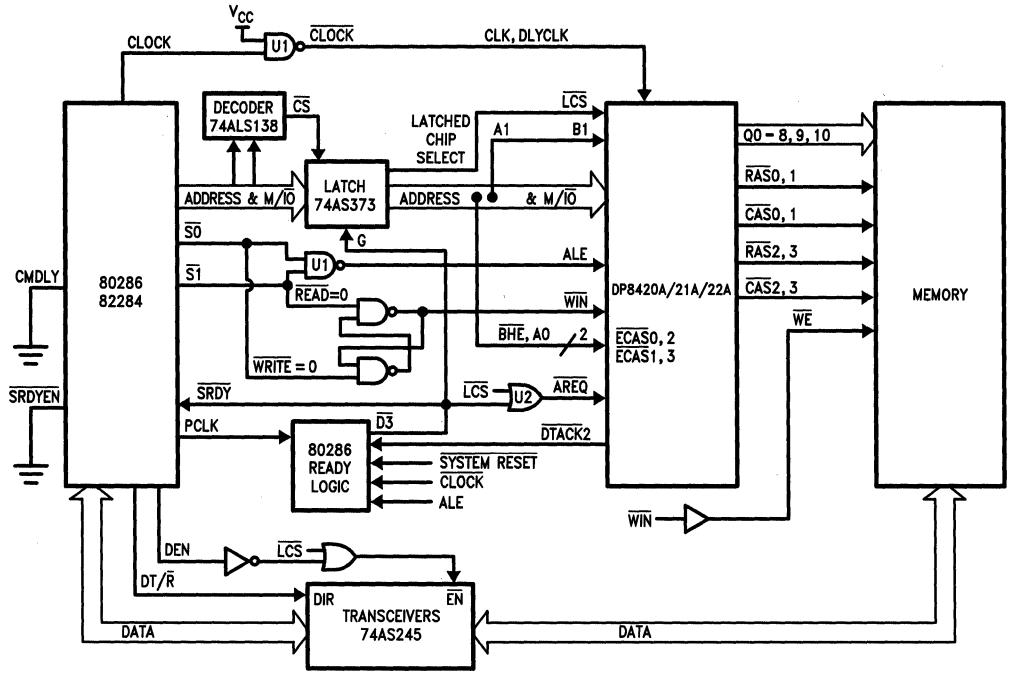
TL/F/9739–5

FIGURE 4. 80286 Ready Logic for Design #2



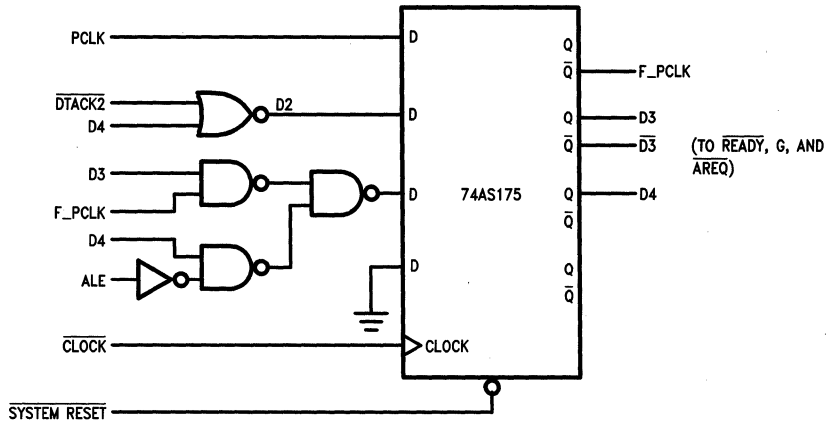
TL/F/9739–6

FIGURE 5. 80286 Design #2 Timing, up to 20 MHz (80286-10), No Wait States



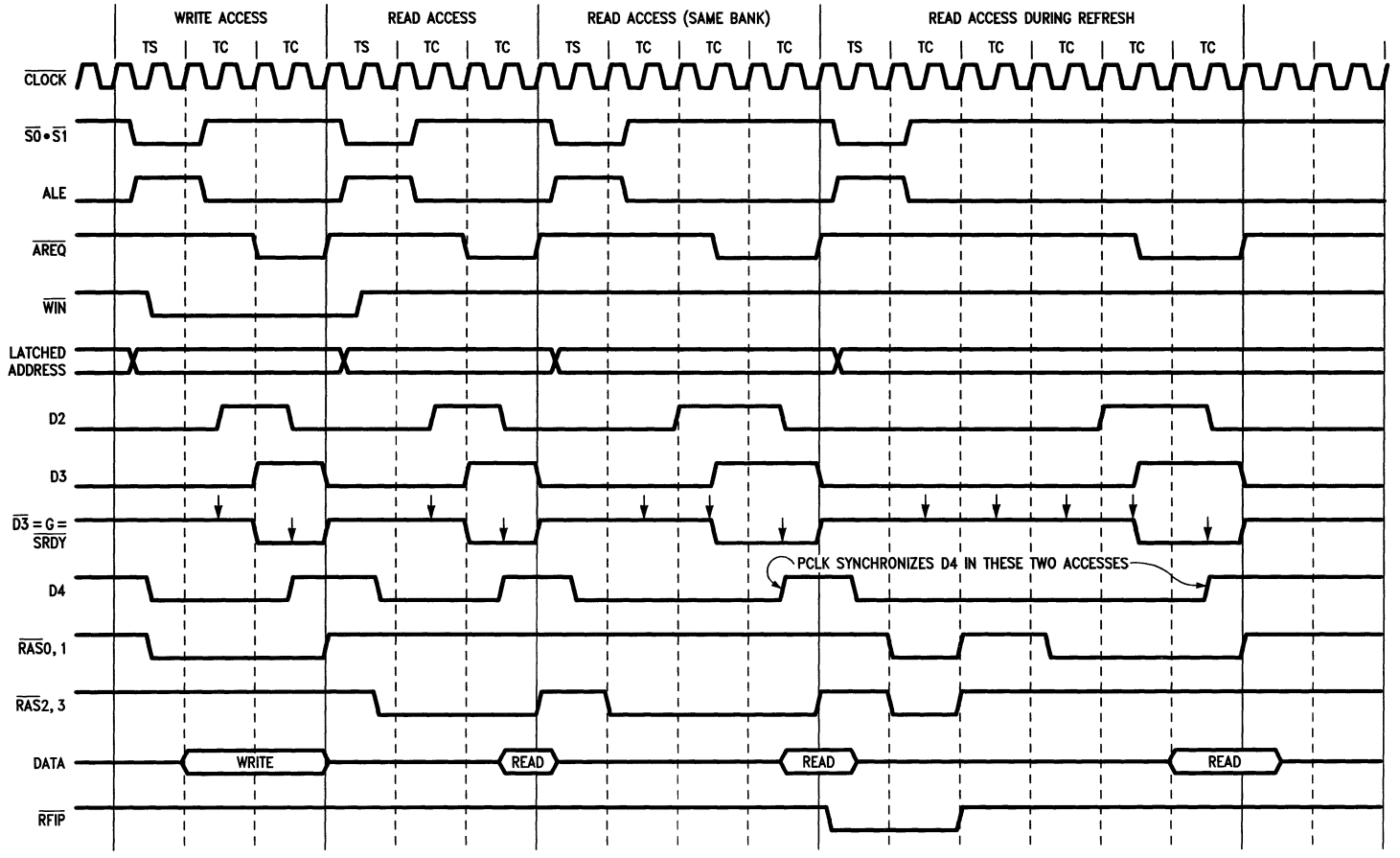
TL/F/9739–8

FIGURE 6. 80286 System Block Diagram of Design #3, Mode 0, up to 25 MHz



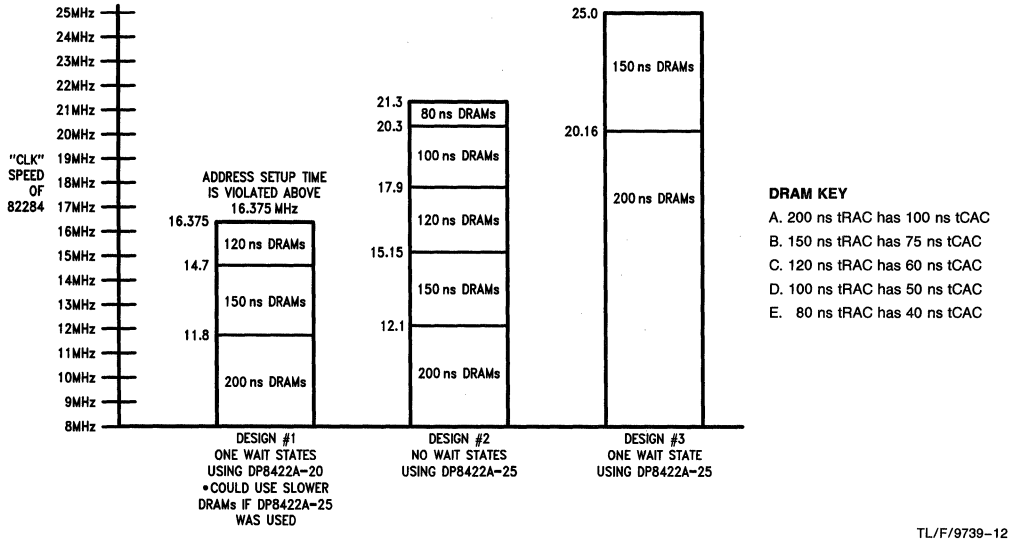
TL/F/9739–9

FIGURE 7. 80286 Ready Logic for Design #3



TL/F/9739–10

FIGURE 8. 80286 Design #3 Timing, up to 25 MHz (80286-12), with One Wait State per Access



TL/F/9739–12

FIGURE 9. 80286, DRAM Speed vs. Processor Speed (the Processor Speed is Referencing the "CLK" Output of 82284, the DRAM Speed is Referencing the $\overline{RAS}$ Access Time "tRAC" of the DRAM)

Interfacing the DP8420A/21A/22A to the 80286 above 25 MHz, including No Wait States in Burst Mode

National Semiconductor
Application Note 618
Webster (Rusty) Meier Jr.

I. INTRODUCTION

This application note describes how to interface the 80286 microprocessor to the DP8422A DRAM controller (also applicable to DP8420A/21A) at clock frequencies above 25 MHz. It is assumed that the reader is already familiar with 80286 and the DP8422A modes of operation.

II. DESCRIPTION OF DESIGN #1, 80286 OPERATING AT UP TO 40 MHz WITH ONE WAIT STATED (80286-20)

The block diagram of this design is shown driving two banks of DRAM, each bank being 16 bits in width, giving a maximum memory capacity of up to 4 Mbytes (using 1 Mbit x 1 DRAMs). This memory could easily be expanded up to 32 Mbytes using four banks of 4 Mbit DRAMs.

The memory banks are interleaved on word (16-bit word) boundaries. This means that the address bit (A1) is tied to the bank select input of the DP8422A (B1).

Address bit A0 is used, along with Bus High Enable (BHE), to produce the two byte select ECAS~0,1 strobes. These byte select strobes (ECAS~0,1) enable the CAS~ outputs which are used in byte reads and writes.

If the majority of accesses made by the 80286 are sequential, the 80286 can alternate memory banks, allowing one memory bank to be precharging (RAS~ precharge) while the other banks are being accessed. Each separate memory access to the same memory bank will require extra wait states to be inserted into the CPU access cycles to allow for the RAS~ precharge time.

This application inserts 1 wait state in normal accesses of the 80286. The number of wait states can be adjusted through the WAITIN input of the DP8422A.

The logic shown in this application note forms a complete 80286 memory sub-system, no other logic is needed. This sub-system automatically takes care of:

A. Arbitration between Port A, Port B, and refreshing the DRAM;

B. The insertion of wait states to the processor (Port A and Port B) when needed (i.e., if RAS~ precharge is needed, refresh is happening during a memory access, the other Port is currently doing an access . . . etc);

C. Performing byte writes and reads to the 16-bit words in memory.

It is important that the 74AS00 NAND gates (U1) be in the same package so these delays (CLK~, S01) track each other.

By using the "output control" pins of some external latches (74AS373's), this application can easily be used in a dual access application. The addresses could be tri-stated through these latches, the write input (WIN~), lock input (LOCK~), and ECAS~0-3 inputs must also be able to be tri-stated (a 74AS244 could be used for this purpose). By multiplexing the above inputs (through the use of the above parts and similar parts for Port B) the DP8422A can be used in a dual access application. If this design is used in a dual access application the tRAC and tCAC (required RAS and CAS access time required by the DRAM) will have to be recalculated since the time to RAS and CAS is longer for the dual access application (see TIMING section of this application note).

Also, throughout this application note the symbol '~' has been used to denote and active low signal. For example RAS~0 refers to the active low RAS0 output of the DP8421A. For even higher system performance an 'E' speed PAL can be used.

III. DESCRIPTION OF DESIGN #2, 80286 OPERATING AT UP TO 40 MHz (80286-20) WITH ZERO WAIT STATES USING PAGE MODE DRAMs

This design is very similar with respect to design #1 except for the following differences.

The memory banks are interleaved on page (1024 word) boundaries. This means that the address bit (A11) is tied to the bank select input of the DP8421A (B1).

Address bit A0 is used, along with Bus High Enable (BHE), to produce the two byte select ECAS~0,1 strobes. These byte select strobes (ECAS~0,1) are logically "ORed" with the DP8421A CAS~ outputs to produce the byte selecting CAS~ inputs to the DRAMs.

If the majority of accesses made by the 80286 are sequential and within a page, the 80286 in conjunction with the page detector (74ALS6311) allow zero wait state accessing. Each in-page memory access is completed using page mode (toggling the CAS~ inputs).

As in design #1 it is important that the 74AS00 NAND gates (U1) be in the same package so the delays (CLK~, S01) track each other. For even higher system performance an 'E' speed PAL could be used.

IV. 80286 DESIGNS #1 AND #2 PROGRAMMING MODE BITS

Programming Bits	Description
R0 = 0 R1 = 1	RAS~ low two clocks, RAS~ precharge of two clocks. If more RAS~ precharge is desired the user should program three periods of RAS~ precharge.
R2 = 0 R3 = 1	DTACK~ ½ is chosen. DTACK~ follows the access RAS~ low.
R4 = 0	No WAIT states during burst accesses.
R5 = 0	
R6 = 0	If WAITIN~ = 0, add one clock to DTACK~. WAITIN~ may be tied high or low in this application depending upon the number of wait states the user desires to insert into the access.
R7 = 1	Select DTACK~.
R8 = 1 R9 = X	Non-interleaved Mode.
C0 = X C1 = X C2 = X C3 = X	Select based upon the input "DELCLK" frequency. Example: if the input clock frequency is 16 MHz then choose C0, 1, 2 = 0, 1, 0 (divide by eight, this will give a frequency of 2 MHz).
C4 = 1 C5 = 0 C6 = 1	RAS banks selected by "B1". This mode allows two RAS~ outputs to go low during an access, and allows byte writing in 16 bit words.
C7 = 1	Column address setup time of 0 ns.
C8 = 1	Row address hold time of 15 ns.
C9 = 1	Delay CAS~ during write accesses to one clock after RAS~ transitions low.
B0 = 1	Fall through latches.
B1 = 1	Access mode 1.
ECAS~0 = 1	Allow CAS~ to be extended after RAS~ transitions high. Also, allow the WE~ output to be used as a refresh request (RFRQ~) output indicator.

0 = Program with low voltage level
1 = Program with high voltage level
X = Program with either high or low voltage level (don't care condition)

V. 80286 TIMING CALCULATIONS FOR DESIGNS #1 AND #2 AT 32 MHz (80286-16) WITH ONE WAIT STATE DURING NORMAL ACCESSES AND ZERO WAIT STATES IN PAGE MODE ACCESSES (DESIGN #2 ONLY). THE WAITIN~ INPUT OF THE DP8422A SHOULD BE TIED LOW.

1. Minimum S01 high setup time to CLK~ high ('D' speed PAL needs 8 ns):

   31.25 ns (one clock period, 32 MHz) − 20 ns (maximum 80286 S0~, S1~ delay, #12a) − 1 ns (maximum skew between CLK~ and S0~, S1~ since both gates are in the same package) = 10.25 ns.

2. Maximum address valid time (with respect to CLK~ high during phase 1 in Ts):

   62.5 ns (two clocks 32 MHz) − 31 ns (80286 address valid delay from previous clock period, #15) + 1 ns (minimum CLK~ valid delay, 74AS00) = −1.25 ns (before CLK~ high phase 1 Ts).

3. Minimum address setup time to ADS~ low (DP8421A-25 needs 14 ns, #404):

   31.25 ns (one clock period) + 1.25 ns (from #2 calculation above) + 2 ns (minimum ADS~ valid delay from CLK~ high, beginning of phase 2 in Ts) = 34.5 ns address setup.

4. Minimum CS setup time to ADS~ low (DP8421A-25 needs 5 ns, #401): 34.5 ns (#3 above) − 10 ns (max 74ALS138 decoder) = 24.5 ns.

5. Determining tRAC during a normal access (RAS~ access time needed by the DRAM):

   156.25 ns (five clock (CLK) periods to do the access) −4.5 ns (max delay 74AS00 for CLK~) − 8 ns (max 'D' speed PAL clocked output delay for ADS~ from CLK~) − 29 ns (ADS~ to RAS~ low max, DP8421A-25 #402) − 7 ns (80286 data setup time #8) − 7 ns (74F245 max delay) = 100.75 ns.

   Therefore the tRAC of the DRAM must be 100.75 ns or less.

6. Determining tCAC during a normal access (CAS~ access time) and column address access time needed by the DRAM:

   156.25 ns (five clock (CLK) periods to do the access) −4.5 ns (max delay 74AS00 for CLK~) − 8 ns (max 'D' speed PAL clocked output delay for ADS~ from CLK~) − 82 ns (ADS~ to RAS~ low max, DP8421A-25 #402) − 7 ns (80286 data setup time #8) − 7 ns (74F245 max delay) = 47.75 ns.

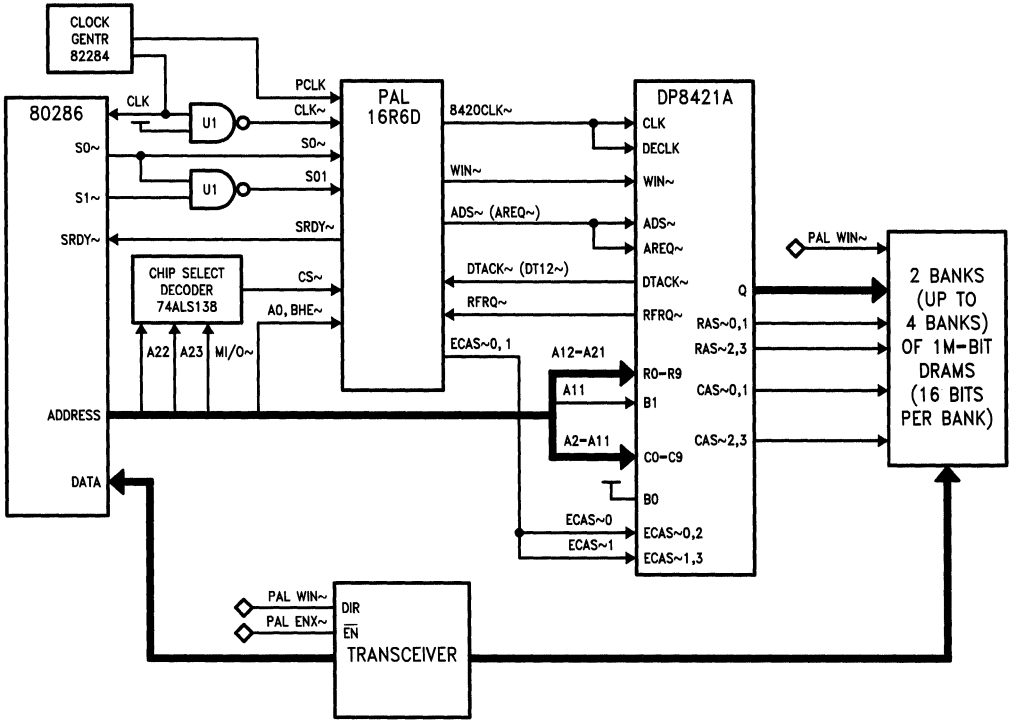
   Therefore the tCAC and the column address access time of the DRAM must be 47.75 ns or less.

7. Determining the column address setup time to CAS~0–3 low (0 ns needed by the DRAMs) during burst mode accesses for zero wait states (DESIGN #2 ONLY):

   31.25 ns (phase 1 in Ts) + 1.25 ns (#2 above, address valid with respect to CLK~ beginning of phase 1 in Ts) + 2 ns (minimum 'D' speed PAL clocked output delay from CLK~, ECAS~0,1) + 2 ns (74AS32 min delay to CAS~0–3 low) = 36.5 ns.

   This gives 1.5 ns column address setup time to CAS~0–3 low (36.5 ns − 35 ns 8421A-25 column address input to output valid, #26).

Interfacing the 80286 to the 8421A



TL/F/10442–1

8. Determining the tCAC (CAS~ access time) needed during burst mode accesses for zero wait states (DESIGN #2 ONLY):

   93.75 ns (three clocks of CLK) − 4.5 ns (74AS00 max delay, CLK~) − 8 ns ('D' speed PAL clocked output delay from CLK~, ECAS~0,1) − 10 ns (74AS32 max delay to CAS~0-3 low) − 7 ns (80286 data setup time #8) − 7 ns (74F245 max delay) = 57.25 ns tCAC needed.

9. Determining the column address access time needed during burst mode accesses for zero wait states (DESIGN #2 ONLY):

   57.25 ns (#8 above, tCAC needed by the DRAM) + 1.5 ns (#7 above, column address setup time to CAS~0-3 low) = 58.75 ns.

10. Minimum SRDY~ (Synchronous ReaDY) setup time to CLK low (80286 SRDY input needs 15 ns, #11):

    62.5 ns (two clock periods) − 4.5 ns (74AS00 max delay, CLK~) − 10 ns ('D' speed PAL combinational output max delay to SRDY~ low) = 48 ns.

Note: Calculations can be performed for different frequencies, different logic (ALS or CMOS ... etc), and/or different combinations of wait states by substituting the appropriate values into the above equations.

VI. 80286 PAL INPUT AND OUTPUT DESCRIPTIONS FOR DESIGNS #1 AND #2

Inputs:

CLK~	The inverted clock (CLK) of the 80286.
PCLK	The half speed clock of the 80286, produced by the 82284.
S01	The 80286 S0~ 'NANDed' with S1~.
S0~	The S0~ output of the 80286.
WIN~	The 80286 S0~ input low latched throughout the access cycle.
CS~	The DRAM chip select generated from the 80286 addresses.
DT12~	The DTACK~ output of the 8421A.
A0	The least significant address bit (low byte enable) from the 80286.
BHE~	The high byte enable from the 80286.
RFRQ~	The refresh request output from the 8421A.
HSA~	The High Speed Access output (comparison equal) from the 74ALS6311.
OE~	Output enable of the PAL®.

Outputs:

ECAS~0	The low byte CAS~ enable, this output also toggles during page mode accesses in design #2.
ECAS~1	The high byte CAS~ enable, this output also toggles during page mode accesses in design #2.
SRDY~	This is the ready input to the 80286, it is used to insert wait states into 80286 access cycles.
8420CLK~	This is the CLOCK and DELCLK input to the 8421A. This clock runs at half of the 80286 CLK frequency.
ADS~	This is the ADS~ and AREQ~ inputs to the 8421A. In design #2 this input stays low thru multiple accesses as long as the accesses are within a page.
NOACC~	This PAL output is low at the end of an 80286 access and stays low until the next access starts.
LREQ~	In Design #2 this output latches that an access request occurred (from the 80286) during an out-of-page access or refresh request during page mode accessing.
WIN~	The latched S0~ output from the 80286.
ENX~	The PAL output used to enable the data transceivers.

80286 PAL EQUATIONS WRITTEN IN NATIONAL SEMICONDUCTOR PLAN™ FORMAT (DESIGN #1)

1. Up to 40 MHz (80286-20)

```
PAL16R6D
CLK~ PCLK S01 S0~ CS~ DT12~ A0 BHE~ NC3 GND
OE~ ECAS~1 WIN~ ENX~ SRDY~ ADS~ NOACC~
8420CLK~ ECAS~0 VCC

If (VCC) /ECAS~0 =  /CS~*S01*S0~*/A0*8420CLK~
                                          ;READ
                    +/CS~*/ADS~*/DT12~
                    */A0*8420CLK~         ;WRITE
                    +/ECAS~0*/ADS~
                    +/ECAS~0*/SRDY~

If (VCC) /ECAS~1 =  /CS~*S01*S0~*
                    /BHE~*8420CLK~        ;READ
                    +/CS~*/ADS~*/DT12~*
                    /BHE~*8420CLK~        ;WRITE
                    +/ECAS~1*/ADS~
                    +/ECAS~1*/SRDY~

/8420CLK~ := /PCLK
/NOACC~   := /SRDY~*/ADS~
             +/NOACC~*/PCLK
             +/NOACC~*CS~*/ADS~
             +/NOACC~*/S01
/ADS~     := /CS~*S01*PCLK
             +/ADS~*SRDY~
/SRDY~    := /CS~*/ADS~*/DT12~*NOACC~*
             /PCLK
             +/SRDY~*/ADS~*NOACC~
/ENX~     := /CS~*/ADS~
/WIN~     := /S0~*S01
             +/WIN~*NOACC~
             +/WIN~*/PCLK
```

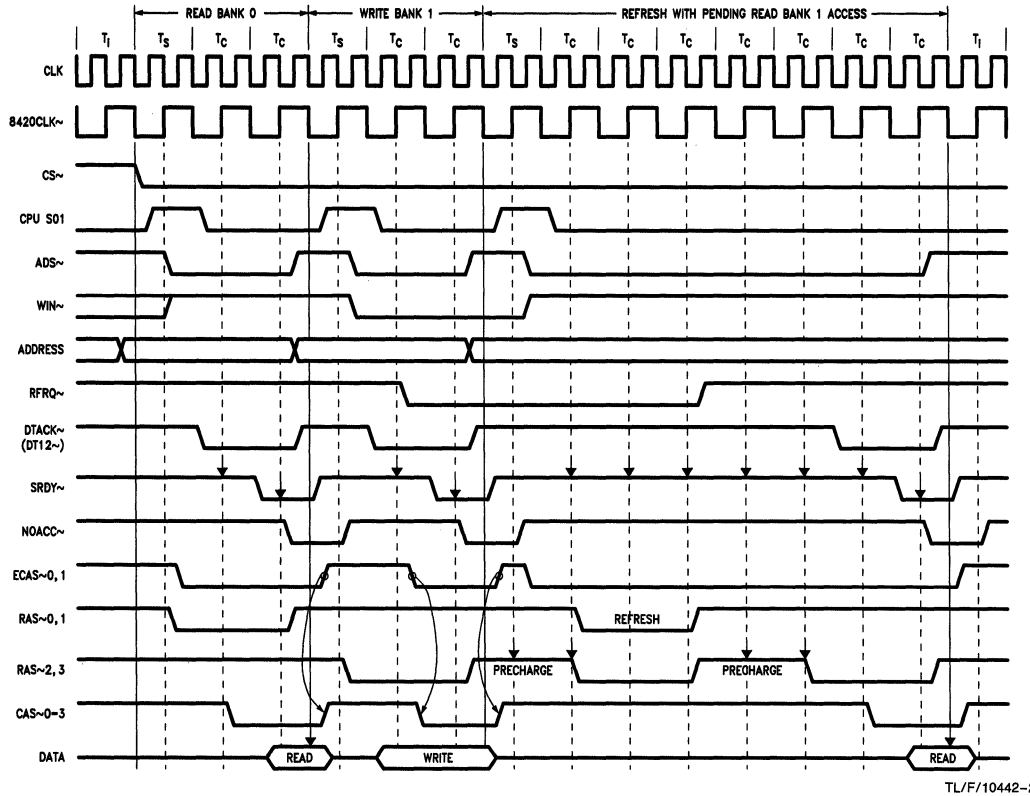
80286 PAGE MODE PAL EQUATIONS WRITTEN IN NATIONAL SEMICONDUCTOR PLAN FORMAT (DESIGN #2)

2. Up to 40 MHz (80286-20)

```
PAL16R6D
CLK~ PCLK S01 WIN~ CS~ DT12~ RFRQ~ HSA~
A0 GND
OE~ BHE~ ADS~ LREQ~ NOACC~ 8420CLK~
ECAS~1 ECAS~0 SRDY~ VCC

If (VCC)/SRDY~ =  /CS~*/ADS~*
                  /DT12~*NOACC~*8420CLK~
                  +/SRDY~*/ADS~*NOACC~
                  +/SRDY~*/ADS~*8420CLK~
```

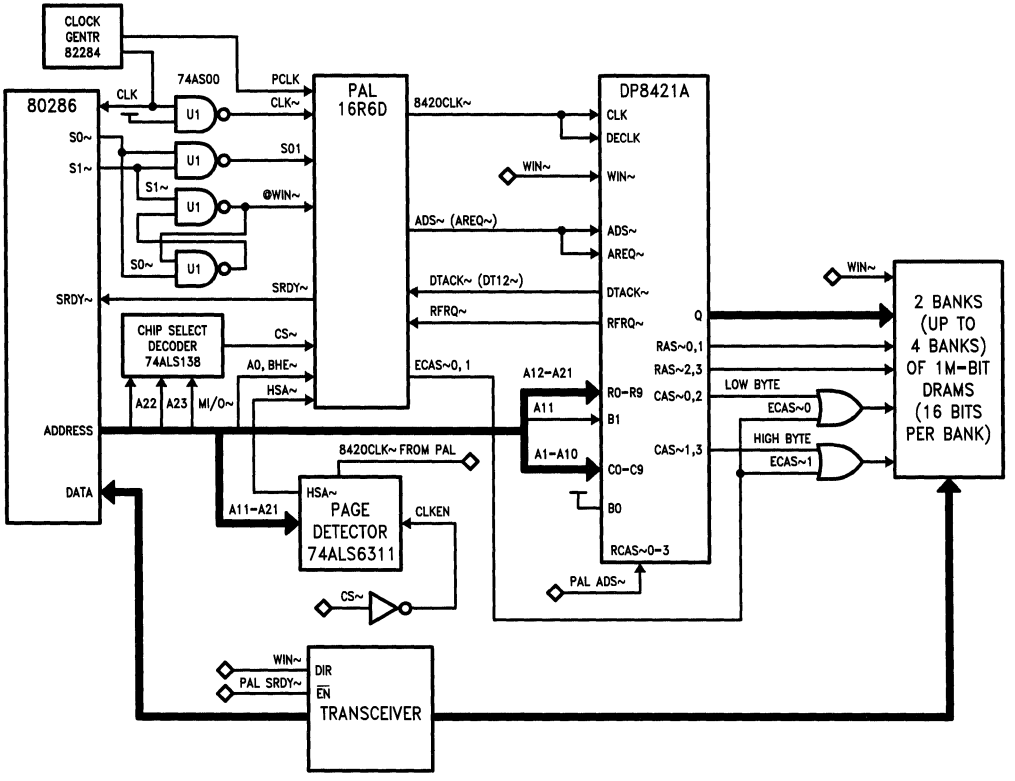
```
/ECAS~0 :=  /CS~*S01*WIN~*/A0*/HSA~*PCLK                          ;READ WITH ADS~ LOW
          + /CS~*S01*WIN~*/A0*HSA~*ADS~*PCLK                      ;READ WITH ADS~ HIGH
          + /CS~*/LREQ~*/A0*/HSA~*WIN~*PCLK                       ;READ DELAYED ACCCESS
          + /CS~*/ADS~*/SRDY~*NOACC~*/A0*PCLK
          + /ECAS~0*/ADS~*NOACC~
/ECAS~1 :=  /CS~*S01*WIN~*/BHE~*/HSA~*PCLK                        ;READ WITH ADS~ LOW
          + /CS~*S01*WIN~*/BHE~*HSA~*ADS~*PCLK                    ;READ WITH ADS~ HIGH
          + /CS~*LREQ~*/BHE~*/HSA~*WIN~*PCLK                      ;READ DELAYED ACCESS
          + /CS~*/ADS~*/SRDY~*NOACC~*/BHE~*PCLK
          + /ECAS~1*/ADS~*NOACC~
/8420CLK~ := /PCLK
/NOACC~ :=  /SRDY~*/ADS~
          + /NOACC~*/PCLK
          + /NOACC~*CS~*/ADS~
          + /NOACC~*/S01
/LREQ~:=  /CS~*S01*HSA~*/ADS~
          + /CS~*S01*/RFRQ~*/ADS~
          + /LREQ~*ADS~
/ADS~: =  CS~*S01*ADS~*RFRQ~*PCLK
          + /LREQ~*/HSA~*PCLK
          + /ADS~*NOACC~
          + /ADS~*/NOACC~*RFRQ~*/HSA~
          + /ADS~*/NOACC~*/PCLK
```

80286/DP8421A Page Mode Timing (Design # 1)



TL/F/10442–2

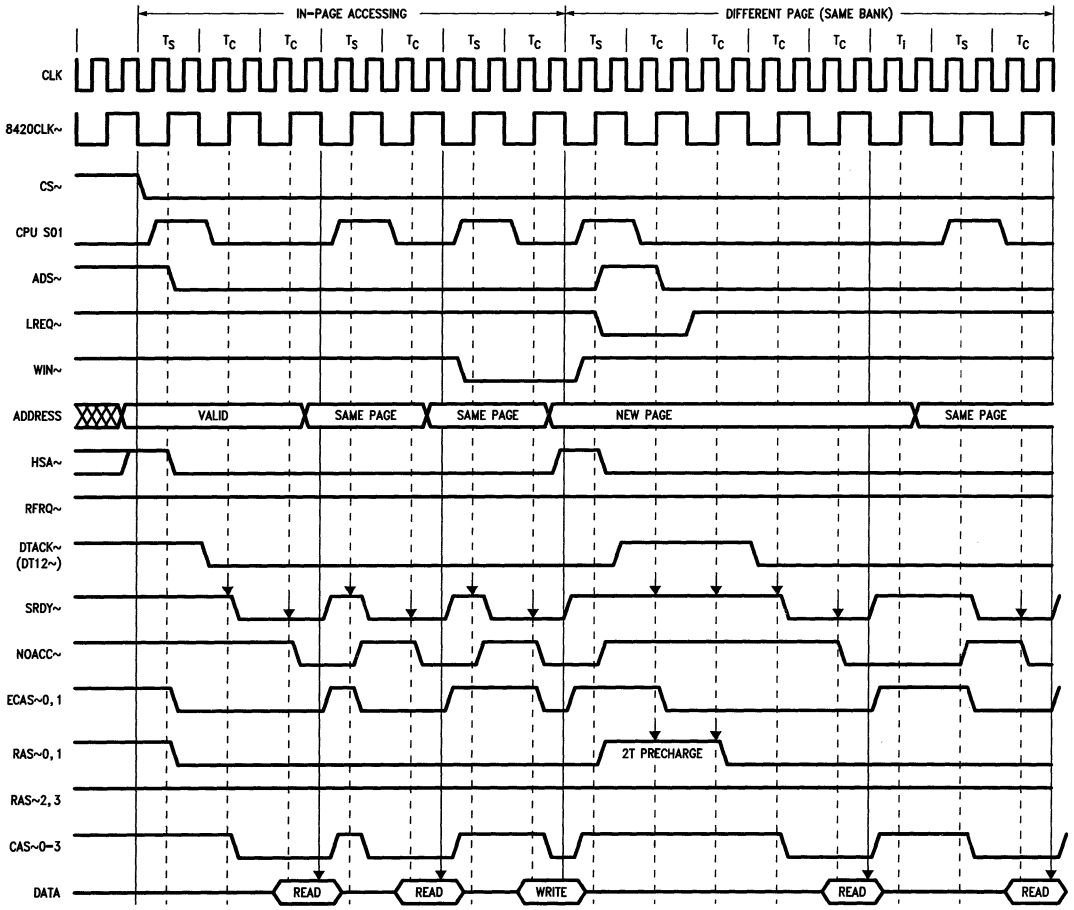
Interfacing 80286/8421A Using Page Mode Accessing (Design #2)



TL/F/10442-3

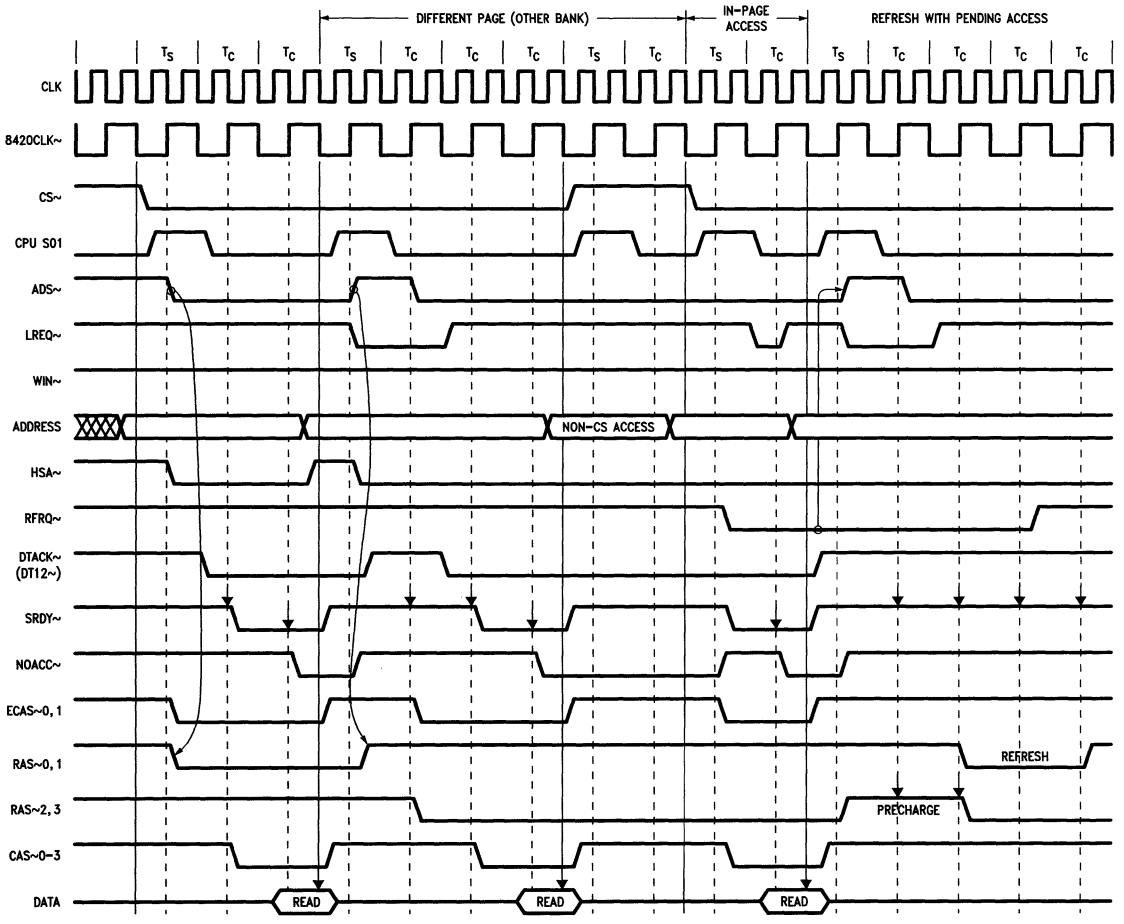
@At high frequencies (CLK > 32 MHz) the WIN~ input may need to be sampled by a flip-flop (clocked by 8420CLK~) before being input to the PAL to meet the setup requirements of the PAL inputs. This would have the effect of delaying ECAS~0,1 becoming valid by one clock period (CLK~) during read accesses, this would not affect the performance of this interface.

80286/DP8421A Page Mode Timing (Design #2)



TL/F/10442–4

80286/DP8421A Page Mode Timing (Design #2)



TL/F/10442-5

Interfacing the DP8420A/21A/22A to the 80386

National Semiconductor
Application Note 536
Webster (Rusty) Meier Jr. and Joe Tate

INTRODUCTION

This application note describes how to interface the 80386 microprocessor to the DP8422A DRAM controller (also applicable to DP8420A/21A). The 80386 is interfaced with the DP8422A in both address pipelined (Design #1) and non-address pipelined (Design #2) mode up to 50 MHz (80386-25). It is assumed that the reader is already familiar with 80386 access cycles and the DP8422A modes of operation.

I. DESCRIPTION OF DESIGN #1A and 1B, THE 80386 IN ADDRESS PIPELINED MODE, ALLOWING OPERATION UP TO 50 MHZ (80386-25) WITH ONE WAIT STATE PER ACCESS. (40 MHZ, TWO WAIT STATES PER ACCESS AT 50 MHZ)

The #1 Designs (80386 in address pipelined mode) consist of the DP8422A DRAM controller, a single PAL® (PAL16R8D), and several logic gates. These parts interface to the 80386 as shown in the block diagrams. This design accommodates two banks of DRAM, each bank being 32 bits in width, giving a maximum memory capacity of 32 Mbytes (using 4 Mbit x 1 DRAMs). See *Figures 1–5*.

Design #1A differs from #1B in terms of the maximum operating frequency. Design #1A can operate up to 40 MHz, Design #1B can operate up to 50 MHz. Designs #1A and B allow 1 wait state per access for read cycles and 2 wait states per access for write cycles in address pipelined mode.

The memory banks are two way interleaved. This means that the least significant address bit (A2) is tied to the bank select input of the DP8422A (B1). Because the majority of accesses made by the 80386 will be sequential, one memory bank can be precharging ($\overline{RAS}$ precharge) while the other bank is being accessed. This allows the memory system to be much higher performance than a non-interleaved memory system. In a non-interleaved memory system, each successive access will generally be to the same memory bank thereby requiring extra wait states to be inserted into the CPU access cycles to allow for the $\overline{RAS}$ precharge time.

In Designs 1A and 1B, the PAL (PAL16R8D) is used primarily to support the address pipelining capability of the 80386 (next address input, NA#). Since the NA# input is only allowed to drop low at the end of the current access no address latches are needed in the system. If address buffers were desired they could be used, but the DP8422A–25 would have to be used in order to meet the bank address and chip select setup times (see "80386 32 MHz Timing Calculations" section). An input is provided ($\overline{EXT\_NA}$) on the PAL for other system peripherals to have their $\overline{NA}$ inputs synchronized to the system clock (up to 50 MHz).

Designs 1A and 1B have one wait state during successive address pipelined accesses to alternating memory banks. During accesses to the same memory bank multiple wait states will be inserted to guarantee $\overline{RAS}$ precharge.

If the user desires two wait states during successive address pipelined accesses (an extra wait state per access), this can be accomplished by running $\overline{RAS0}$ and $\overline{RAS2}$ through a flip-flop (clocked by CLKA) before allowing them to be input to the PAL in Design #1B. This will delay $\overline{NA}$ and $\overline{READY}$ by one CLKA clock period. In Design #1A the $\overline{WAITIN}$ input could be tied low and programmed to add 1 clock to the $\overline{DTACK}$ output.

If the user wants to do dual accessing with the DP8422A DRAM controller, address buffers (74AS244s) must be added to the address, $\overline{ECAS0–3}$, $\overline{LOCK}$, and $\overline{WIN}$ inputs. For the 32 MHz system (80386–16), the system diagram will remain unchanged, but the user will need to use the faster DP8422A–25 part.

For higher frequency dual access memory systems (above 32 MHz), these designs will have to be modified as above. Also, CLKA should be inverted (use $\overline{1Q}$ output from 74AS175). This will cause $\overline{RAS}$ to be started one half CLKA clock period later, allowing extra address and chip select setup time to the DP8422A.

II. DESCRIPTION OF DESIGN #2, THE 80386 IN NON-ADDRESS PIPELINED MODE, ALLOWING OPERATION UP TO 40 MHZ (80386-20) WITH TWO WAIT STATES PER ACCESS (50 MHZ WITH THREE WAIT STATES)

Design #2 (80386 not using address pipelined mode) consists of the DP8422A DRAM controller, several flip-flops (74AS175), and several logic gates. These parts interface to the 80386 as shown in the block diagrams. This design accommodates two banks of DRAM, each bank being 32 bits in width, giving a maximum memory capacity of 32 Mbytes (using 4 Mbit x 1 DRAMs). See *Figures 6–10*.

The memory banks are two way interleaved. This means that the least significant address bit (A2) is tied to the bank select input of the DP8422A (B1). Because the majority of accesses made by the 80386 will be sequential, one memory bank can be precharging ($\overline{RAS}$ precharge) while the other bank is being accessed. This allows the memory system to be much higher performance than a non-interleaved memory system. In a non-interleaved memory system, each successive access will generally be to the same memory bank thereby requiring extra wait states to be inserted into the CPU access cycles to allow for the $\overline{RAS}$ precharge time.

Design #2 has two wait states during successive accesses to alternating memory banks. During accesses to the same memory bank multiple wait states will be inserted to guarantee $\overline{RAS}$ precharge.

If the user desires three wait states during successive accesses (an extra wait state per access), this can be accomplished by pulling $\overline{WAITIN}$ low during accesses. $\overline{WAITIN}$ should be programmed to add one clock period (CLKA) to the $\overline{DTACK}$ output. This will delay $\overline{READY}$ by one CLKA clock period.

If the user wants to do dual accessing with the DP8422A DRAM controller address buffers (74AS244s) must be added to the address, $\overline{ECAS}$0–3, $\overline{LOCK}$ and $\overline{WIN}$ inputs. For the 32 MHz system (80386-16), the system diagram will remain unchanged, but the user will need to use the faster DP8422A–25 part.

For higher frequency dual access memory systems (above 32 MHz), design #2 will have to be modified as above. Also, CLKA should be inverted (use $\overline{1Q}$ output from 74AS175). This will cause $\overline{RAS}$ to be started one half CLKA clock period later, allowing extra address and chip select setup time to the DP8422A.

III. COMMON DESIGN FEATURES

The logic shown in these applications form a complete 80386 memory sub-system, no other logic is needed. This sub-system automatically takes care of:

a. arbitration between Port A, Port B, and refreshing the DRAM;

b. the insertion of wait states to the processor (Port A) when needed (i.e., one wait state during address pipelined accesses (#1 Designs), two wait states during non-address pipelined accesses (Design #2), multiple wait states if an access takes place during a refresh operation or if $\overline{RAS}$ precharge is needed . . . etc.);

c. enabling address pipelining on the 80386 through the NA# input (#1 Designs only), and

d. performing byte writes and reads to the 32-bit words in memory.

The timing calculations for two designs (Designs #1 and #2) are included in this application note with the DP8422A interfaced to the 80386-16 running at 32 MHz and the 80386-20 running at 40 MHz.

Since the DP8420A/21A/22A has a column address hold time of 32 ns the minimum time between two accesses (to guarantee 0 ns row address setup time) is 150 ns (equivalent to three clock periods at 20 MHz, 150 ns).

When using the DP8420A/21A/22A at 20 MHz the user should program three clock periods of precharge. This is because two clock periods of precharge at 20 MHz will only guarantee 81 ns of $\overline{RAS}$ precharge (2 x 50 ns $- t_{D1}$, (parameter #50 "14 ns") – clock (20 MHz) to $\overline{AREQ}$ high, (approximately 5 ns for both design #1 and #2)).

In Design #2 the four gates "A1, A2, B, C" are not necessary if the system designer already has some means of correctly enabling the data transceivers. Also, in Design #2 the NOR gate that produces $\overline{READY}$ will not be needed in many systems, the $\overline{3Q}$ output of the 74AS175 could be used instead ($\overline{READY}$2.5). Though, this output would not allow quite as much $\overline{READY}$ setup time as the output of the NOR gate.

Because of the way ALE is generated to the DP8422A, two pulses of ALE may be generated during each access (see timing diagrams 3, 4, 5, 8, 9 and 10). This is not detailed in the DP8420A/21A/22A data sheet but this is permissible as long as no glitches happen after $\overline{AREQ}$ transitions low for that access. Therefore, this is a valid way of providing ALE to the DP8422A.

IV. 80386 DESIGN #1 AND #2, PROGRAMMING MODE BITS

Programming Bits	Description
R0 = 1 R1 = 1	$\overline{RAS}$ low four clocks, $\overline{RAS}$ precharge of three clocks
R2 = X R3 = X	
R4 = 0 R5 = 0 R6 = X R7 = X	No WAIT states during burst accesses
R8 = 0 R9 = X	Interleaved Mode (requires DRAMs with a column address hold time of 32 ns or less)
C0 = X C1 = X C2 = X	Select based upon the input clock frequency. Example: if the input clock frequency is 16 MHz then choose C0, 1, 2 = 0, 1, 0 (divide by eight, this will give a frequency of 2 MHz).
C3 = X	
C4 = 1 C5 = 0 C6 = 1	$\overline{RAS}$ and $\overline{CAS}$ groups selected by "B1". This mode allows two $\overline{RAS}$ and two $\overline{CAS}$ outputs to go low during an access.
C7 = 1	Column address setup time of 0 ns.
C8 = 1	Row address hold time of 15 ns.
C9 = 1	Delay $\overline{CAS}$ during write accesses to one clock after $\overline{RAS}$ transitions low
B0 = 1	Fall-thru latches
B1 = 0	Access mode 0
$\overline{ECAS}$0 = 0	Non-extend $\overline{CAS}$

0 = Program with low voltage level
1 = Program with high voltage level
X = Program with either high or low voltage level (don't care condition)

V. 80386 #1 DESIGNS (AND #2 DESIGN), 32 MHZ TIMING CALCULATIONS, WITH ONE WAIT STATE PER ACCESS IN ADDRESS PIPELINED MODE, TWO WAIT STATES PER ACCESS IN NON-ADDRESS PIPELINED MODE (DP8422A-20 USES THE 16 MHZ CLOCK)

Note: Design #2 timing calculations are the same as Design #1 except for those calculations involving "CLKA", Timing calculations involving the minimum or maximum delay or skew of CLKA with respect to CLK2 should be recalculated substituting the 74AS175 parameters for the PAL parameters.

1. Maximum time to address valid (with respect to 16 MHz clock):

   40 ns (maximum time to address valid) − 2 ns ("D" speed PAL minimum delay, because 16 MHz clock is from PAL with minimum delay as skew from 32 MHz CPU clock) = 38 ns

2. Maximum time to ALE high (with respect to 16 MHz clock):

   35 ns (maximum $\overline{ADS}$ valid) + 4.5 ns (74AS02 maximum delay) − 2 ns ("D" speed PAL minimum clock delay, for skew between 32 MHz and 16 MHz clock) = 37.5 ns

   The CLKA delay to ALE through 74AS02 is similar: 31 ns (one half clock period, 16 MHz) + 4.5 ns (74AS02 maximum delay) + 1.5 ns (PAL estimated skew between low to high and high to low clock to output delay) = 37 ns

3. Minimum ALE high setup time to CLKA high (DP8422A-20 needs 16 ns):

   62.5 ns (one clock period, 16 MHz) − 37.5 ns (#2) = 25 ns

4. Minimum address setup time to CLKA high (DP8422-20 needs 20 ns):

   62.5 ns (Once clock period, 16 MHz) − 38 ns (#1) = 24.5 ns

5. Minimum $\overline{CS}$ setup time to CLKA high (DP8422-20 needs 14 ns):

   24.5 ns (#4) − 9 ns (74AS138 decoder) = 15.5 ns

6. Determining t_{RAC} ($\overline{RAS}$ access time needed by the DRAM):

   250 ns (four clock periods at 16 MHz) −62.5 ns (one clock period) −8 ns (PALmaximum delay low to high from CLK2 clock, clock skew) −10 ns (data setup time) −7 ns (74AS245) −32 ns (CLK to $\overline{RAS}$ low) = 130.5 ns

   Therefore the t_{RAC} of the DRAM must be 130.5 ns or less.

7. Determining t_{CAC} ($\overline{CAS}$ access time needed by the DRAM):

   250 ns −62.5 ns −8 ns −10 ns −7 ns −89 ns (CLK to $\overline{CAS}$ low) = 73.5 ns

   Therefore the t_{CAC} of the DRAM must be 73.5 ns or less. COMMON 120 ns DRAMS WILL MEET THIS t_{RAC} AND t_{CAC} PARAMETER.

8. Minimum setup of $\overline{DTACK0}$ to the PAL16R8D, DESIGN #1 ONLY, (need 10 ns):

   62.5 ns (one clock period) −8 ns (PAL maximum delay low to high from clock, clock skew 32 MHz vs 16 MHz) −41 ns (clock to $\overline{DTACK0}$ valid from DP8422A-20) = 13.5 ns

9A. Minimum $\overline{READY}$ setup time to $\overline{READY}$ being sampled (20 ns is needed by the 80386) DESIGN #1A and #1B:

   31.25 ns (one half clock period) −8 ns (maximum "D" PAL clocked output delay) = 23.25 ns

9B. Minimum $\overline{READY}$ setup time to $\overline{READY}$ being sampled (20 ns is needed by the 80386) DESIGN #2:

   IF 74AS02 IS USED TO PRODUCE $\overline{READY}$:

   31.25 ns (Last one half clock period of T2) + 13 ns [17.5 ns (see in note below) −4.5 ns (max 74AS02 Delay)] = 44.25 ns

   IF $\overline{3Q}$ OUTPUT OF 74AS175 IS USED FOR $\overline{READY}$:

   31.25 ns (Last one half clock period of T2) −10 ns (max delay of 74AS175) = 21.25 ns

Note: $\overline{DTACK1.5}$ setup to 74AS175 input, used to generate $\overline{DTACK2.5}$ (74AS175 needs 3 ns):

   62.5 ns (one clock period at 16 MHz) −45 ns [7.5 ns (CLKA max delay) + 33 ns (DP8422A-20 $\overline{DTACK1.5}$ max delay) + 4.5 ns (74AS02 max delay)] = 17.5 ns setup to mid T2 of last access clock period.

10. Minimum $\overline{NA}$ setup time to $\overline{NA}$ being sampled, Design #1 only (10 ns is needed by the 80386):

   31 ns (one clock period at 32 MHz) −8 ns (maximum "D" PAL clocked output delay) = 23 ns

VI. 80386 #1 DESIGNS (AND #2 DESIGN), 40 MHZ TIMING CALCULATIONS, WITH ONE WAIT STATE PER ACCESS IN ADDRESS PIPELINED MODE, TWO WAIT STATES PER ACCESS IN NON-ADDRESS PIPELINED MODE (DP8422A-25 USES THE 20 MHZ CLOCK)

***Note:** Design #2 timing calculations are the same as Design #1 except for those calculations involving "CLKA". CLKA is produced by a 74AS175 in Design #2 instead of a "D" speed PAL (Design #1). Therefore the timing calculations involving the minimum or maximum delay or skew of CLKA with respect to CLK2 should be recalculated substituting the 74AS175 parameters for the PAL parameters.

1. Maximum time to address valid (with respect to 20 MHz clock):

   32 ns (maximum time to address valid) −2 ns ("D" speed PAL minimum delay, because 20 MHz clock is from PAL with minimum delay as skew from 40 MHz CPU clock) = 30 ns

2. Maximum time to ALE high (with respect to 20 MHz clock):

   30 ns (maximum $\overline{ADS}$ valid) +4.5 ns (74AS02 maximum delay) −2 ns (PAL minimum clock delay, for skew between 40 MHz and 20 MHz clock) = 32.5 ns

   The CLKA delay to ALE through 74AS02 is: 25 ns (one half clock period, 20 MHz) +4.5 ns (74AS02 maximum delay) +1.5 ns (PAL skew between low to high and high to low clock to output delay) = 31 ns

3. Minimum ALE high setup time to CLKA high (DP8422A-25 needs 15 ns):

   50 ns (one clock period, 20 MHz) −32.5 ns (#2) = 17.5 ns

4. Minimum address setup time to CLKA high (DP8422A-25 needs 18 ns):

   50 ns (one clock period, 20 MHz) −30 ns (#1) = 20 ns

5. Minimum $\overline{CS}$ setup time to CLKA high (DP8422A-25 needs 13 ns):

   20 ns (#4) −6 ns (74AS139 two to four decoder) = 14 ns

6. Determining t_{RAC} ($\overline{RAS}$ access time needed by the DRAM):

   200 ns (four clock periods at 20 MHz) −50 ns (one clock period) −8 ns (PAL maximum delay low to high from CLK2 clock, clock skew) −10 ns (data setup time) −7 ns (74AS245) −26 ns(CLK to $\overline{RAS}$ low) = 99 ns
   Therefore the t_{RAC} of the DRAM must be 99 ns or less.

7. Determining t_{CAC} ($\overline{CAS}$ access time needed by the DRAM):

   200 ns −50 ns −8 ns −10 ns −7 ns −79 ns (CLK to $\overline{CAS}$ Low) = 46 ns
   Therefore the t_{CAC} of the DRAM must be 46 ns or less.

8. Minimum setup of $\overline{DTACK0}$ to the PAL16R8D, DESIGN #1 ONLY, (need 8 ns):

   50 ns (one clock period) −8 ns (PAL maximum delay low to high from clock, clock skew 20 MHz vs 40 MHz) −33 ns (clock to $\overline{DTACK0}$ valid from DP8422A-25) = 9 ns

9A. Minimum $\overline{READY}$ setup time to $\overline{READY}$ being sampled (11 ns is needed by the 80386) DESIGN #1:

   25 ns (one half clock period) −8 ns (maximum "D" PAL clocked output delay) = 17 ns

9B. Minimum $\overline{READY}$ setup time to $\overline{READY}$ being sampled (11 ns is needed by the 80386) DESIGN #2:

   IF 74AS02 IS USED TO PRODUCE $\overline{READY}$:

   25 ns (Last one half clock period of T2) +5.5 ns [10 ns (see in note below) −4.5 ns (max 74AS02 delay)] = 30.5 ns

   IF $\overline{3Q}$ OUTPUT OF 74AS175 IS USED FOR $\overline{READY}$:
   25 ns (Last one half clock period of T2) −10 ns (max delay of 74AS175) = 15 ns

Note: $\overline{DTACK}$ 1.5 setup to 74AS175 input, used to generate $\overline{DTACK2.5}$ (74AS175 needs 3 ns):

   50 ns (one clock period of 20 MHz) −40 ns [7.5 ns (CLKA max delay) +28 ns (DP8422-25 $\overline{DTACK1.5}$ max delay) +4.5 ns (74AS02 max delay)] = 10 ns setup to mid T2 of last access clock period.

10. Minimum $\overline{NA}$ setup time to $\overline{NA}$ being sampled, Design #1 only (8 ns is needed by the 80386):

   25 ns (one clock period at 40 MHz) −8 ns (maximum "D" PAL clocked output delay) = 17 ns

VII. 80386 DESIGN # 1, PAL EQUATIONS WRITTEN IN NATIONAL SEMICONDUCTOR PLAN™ FORMAT

```
DESIGN #1A UP TO 40 MHz (80386-20)
PAL16R8D
CLK2 82384CLK ADS CS DTACK0 EXT_NA W B1 RESET GND
OE BOWE ADS1D CLKA READ READY NA ADS3D B1WE VCC

B1WE :=ADS3D*ADS1D*READ*DTACK0*B1
      +B1WE*CLKA*READ*RESET

BOWE :=ADS3D*ADS1D*READ*DTACK0*B1
      +BOWE*CLKA*READ*RESET

ADS3D :=ADS1D*CLKA*RESET
       +ADS3D*CLKA*RESET

NA :=DTACK0*ADS1D*CLKA*RESET*W
    +DTACK0*ADS1D*ADS3D*BOWE*CLKA*RESET
    +DTACK0*ADS1D*ADS3D*B1WE*CLKA*RESET
    +EXT_NA*CLKA*RESET
    +NA*CLKA*RESET

READY :=NA*ADS3D*ADS1D*DTACK0*CLKA*RESET
       +READY*ADS3D*DTACK0*RESET

ADS1D :=ADS*CLKA*CS
       +ADS1D*NA*RESET
       +ADS1D*CLKA*RESET

READ :=CS*W*ADS1D*CLKA*RESET
      +READ*ADS1D*RESET
      +READ*CLKA*RESET

CLKA :=82384CLK

DESIGN #1B UP TO 50 MHz (80386-25)
PAL16R8D
CLK2 82384CLK ADS CS RAS0 RAS2 W NO_NA RESET GND
OE BOWE ADS1D CLKA READ READY NA ADS3D B1WE VCC

B1WE :=ADS3D*ADS1D*READ*RAS2*NO_NA
      +B1WE*CLKA*READ*RESET

BOWE :=ADS3D*ADS1D*READ*RAS0*NO_NA
      +BOWE*CLKA*READ*RESET

ADS3D :=ADS1D*CLKA*RESET
       +ADS3D*CLKA*RESET

NA :=RAS0*ADS1D*CLKA*RESET*NO_NA*W
    +RAS0*ADS1D*ADS3D*BOWE*CLKA*RESET
    +RAS0*ADS1D*ADS3D*B1WE*CLKA*RESET
    +RAS2*ADS1D*CLKA*RESET*NO_NA*W
    +RAS2*ADS1D*ADS3D*BOWE*CLKA*RESET
    +RAS2*ADS1D*ADS3D*B1WE*CLKA*RESET
    +NA*CLKA*RESET

READY :=NA*ADS3D*ADS1D*RAS0*CLKA*RESET*NO_NA
       +READY*ADS3D*RAS0*RESET*NO_NA
       +NA*ADS3D*ADS1D*RAS2*CLKA*RESET*NO_NA
       +READY*ADS3D*RAS2*RESET*NO_NA

ADS1D :=ADS*CLKA*CS
       +ADS1D*NA*RESET
       +ADS1D*CLKA*RESET

READ :=CS*W*ADS1D*CLKA*RESET
      +READ*ADS1D*RESET
      +READ*CLKA*RESET

CLKA :=82384CLK
```

Logic Needed for "NO_NA" Term in Design #1B

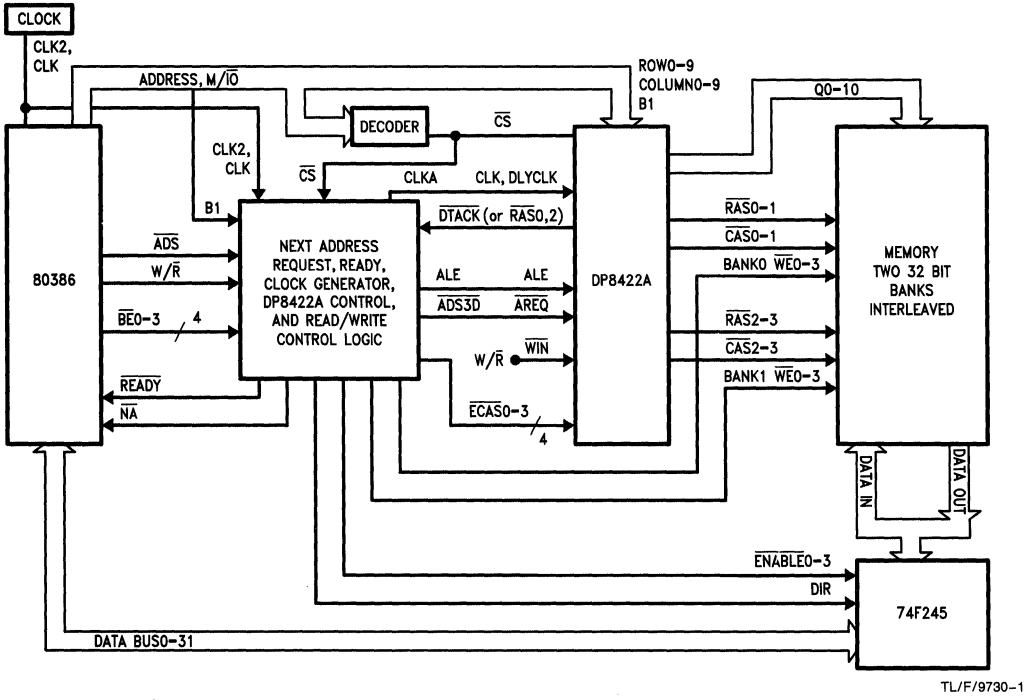
TL/F/9730–12

Key: Reading PAL Equations Written in PLAN

EXAMPLE EQUATIONS: $\overline{READ}$: = CS_RD*ADS1D*$\overline{CLKA}$
+$\overline{READ}$*$\overline{ADS1D}$
+$\overline{READ}$*CLKA

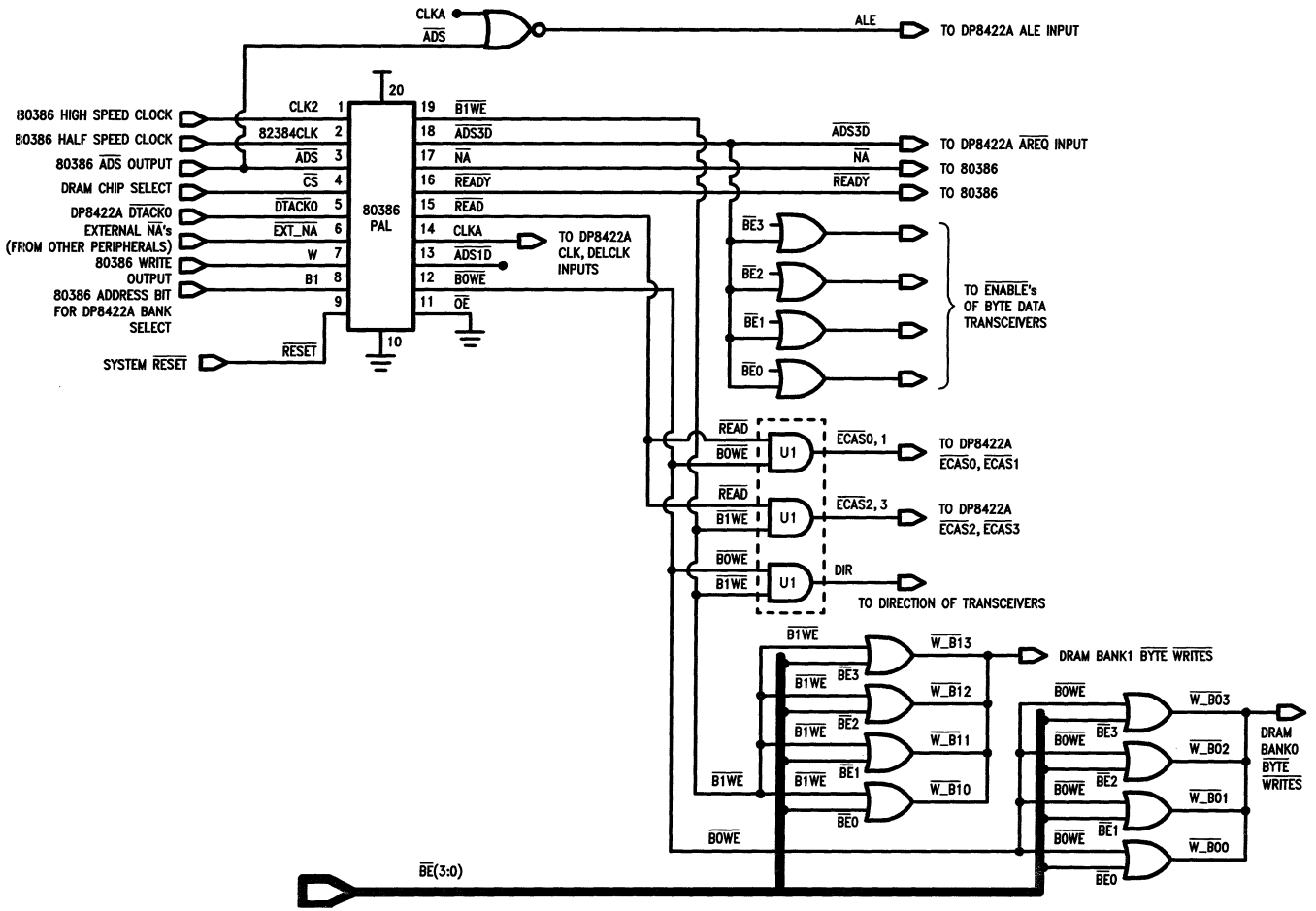
This example reads: the output "$\overline{READ}$" will transition low on the next rising "CLK2" clock edge (given that one of the following conditions are valid, a setup time before "CLK2" transitions high);

1. the input "CS_RD" is high AND the input "$\overline{ADS1D}$" is high AND the input "CLKA" is low, OR
2. the output "$\overline{READ}$" is low AND the input "$\overline{ADS1D}$" is low, OR
3. the output "$\overline{READ}$" is low AND the input "CLKA" is high



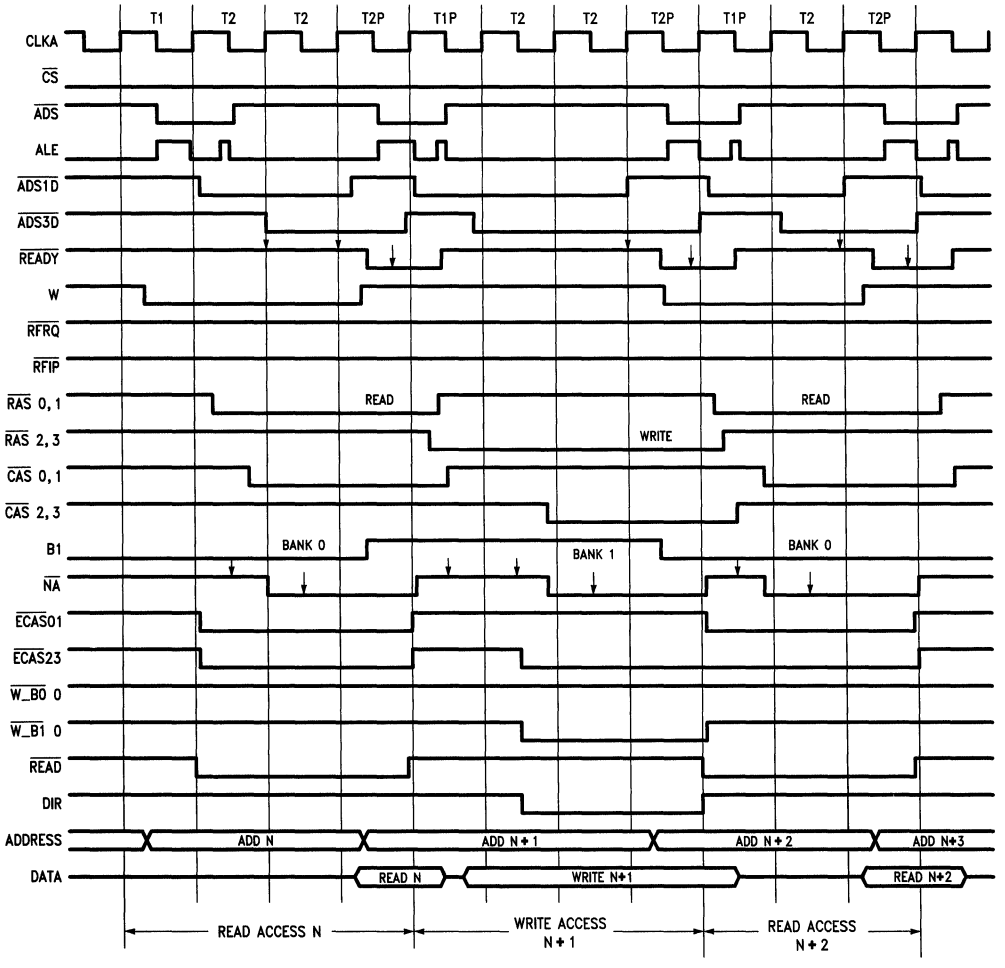
TL/F/9730-1

FIGURE 1. 80386 Design #1 (A and B), System Block Diagram for Address Pipelined Mode Operation at up to 50 MHz (DP8422A Uses Half Speed Clock, CLKA)



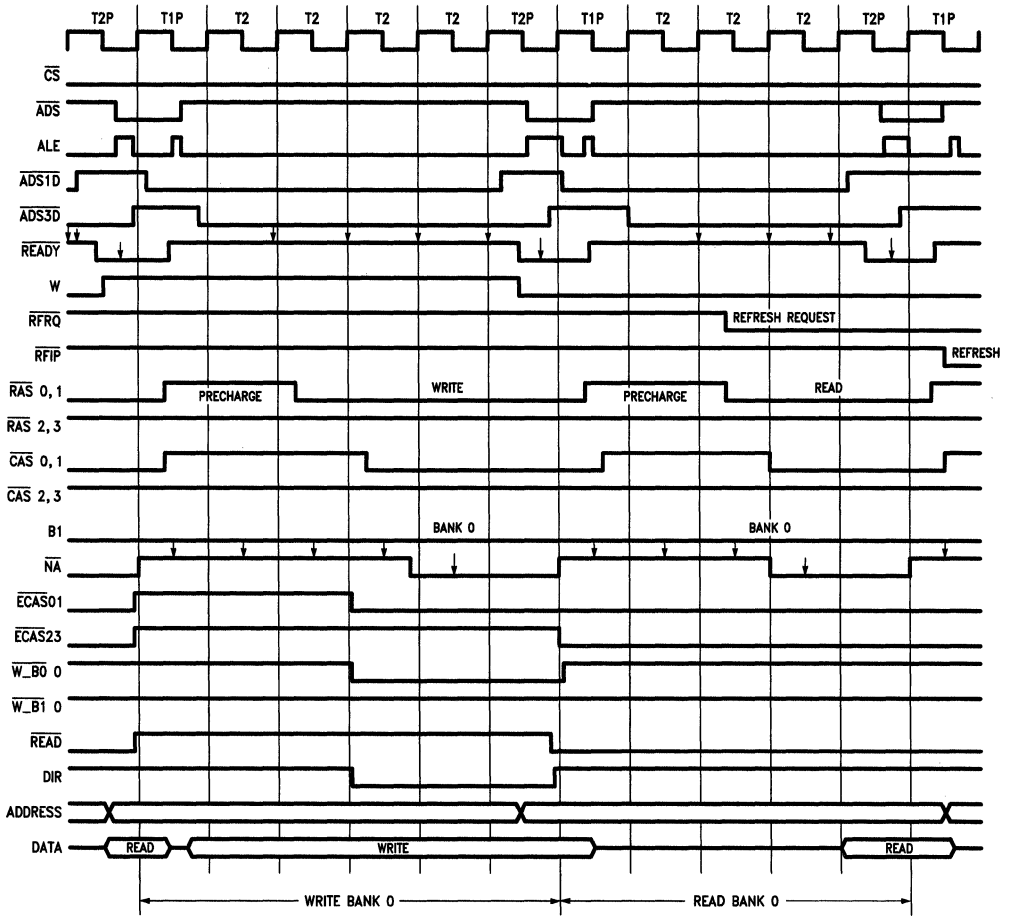
TL/F/9730–2

FIGURE 2. 80386/DP8422A Interface Control Logic for Design # 1A



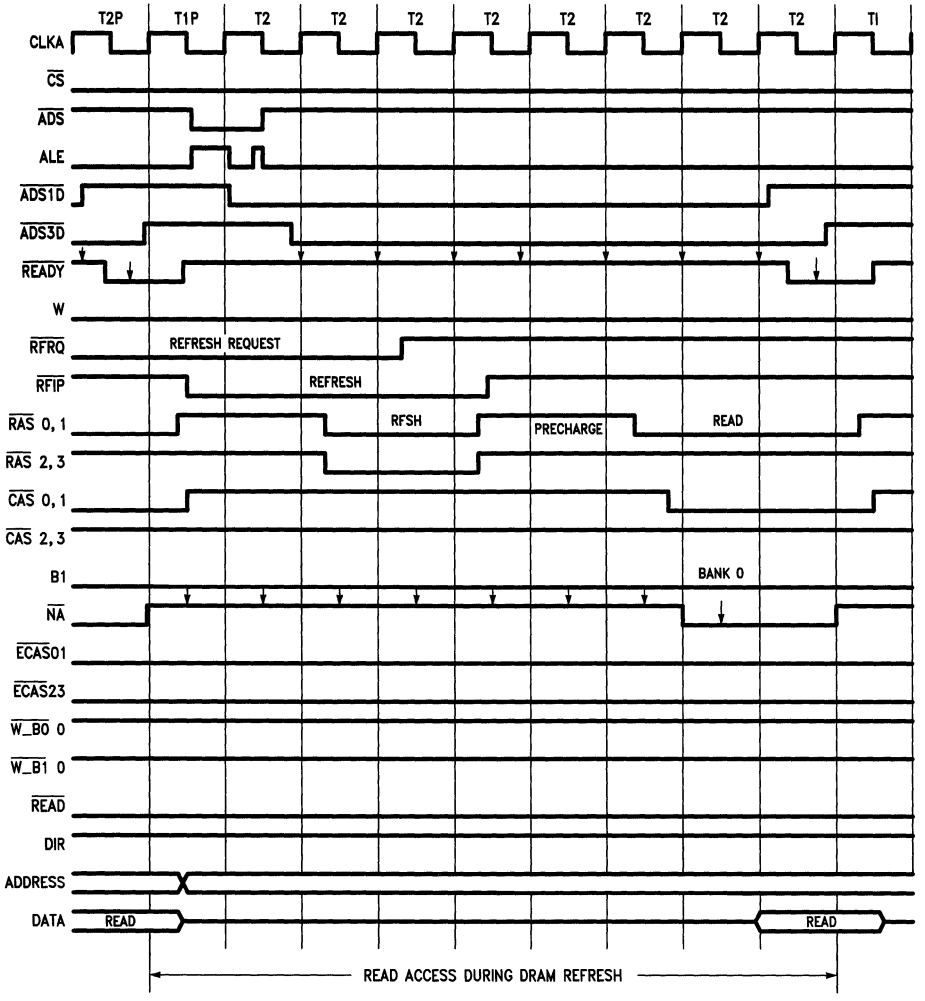
TL/F/9730–3

FIGURE 3. 80386 Design # 1 (A and B) Address Pipelined Timing



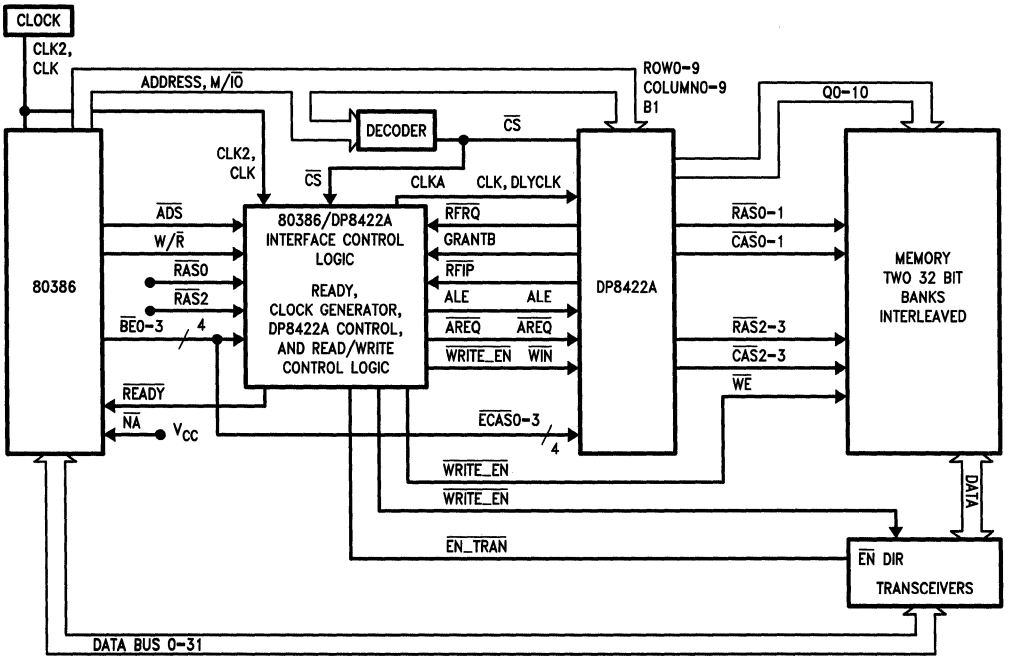
TL/F/9730–4

FIGURE 4. 80386 Design #1 Address Pipelined Timing



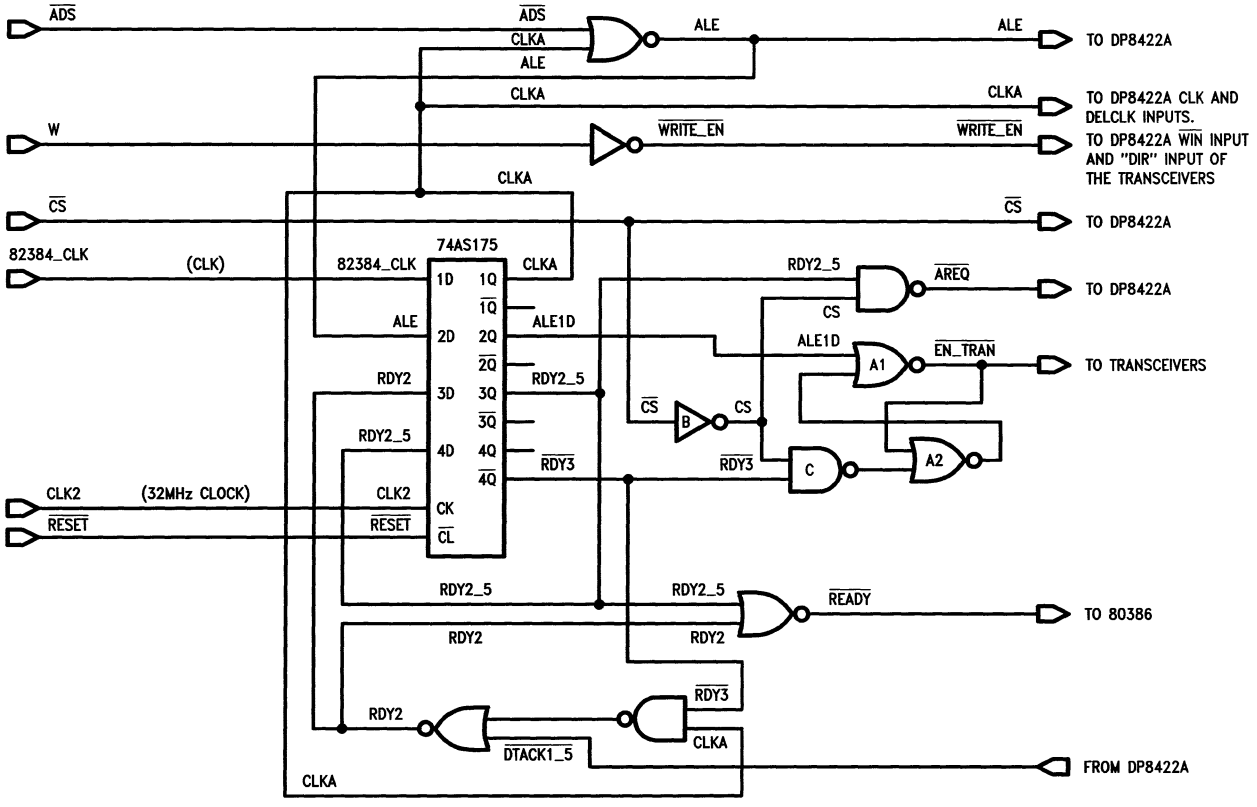
TL/F/9730–5

FIGURE 5. 80386 Design #1 Address Pipelined Timing



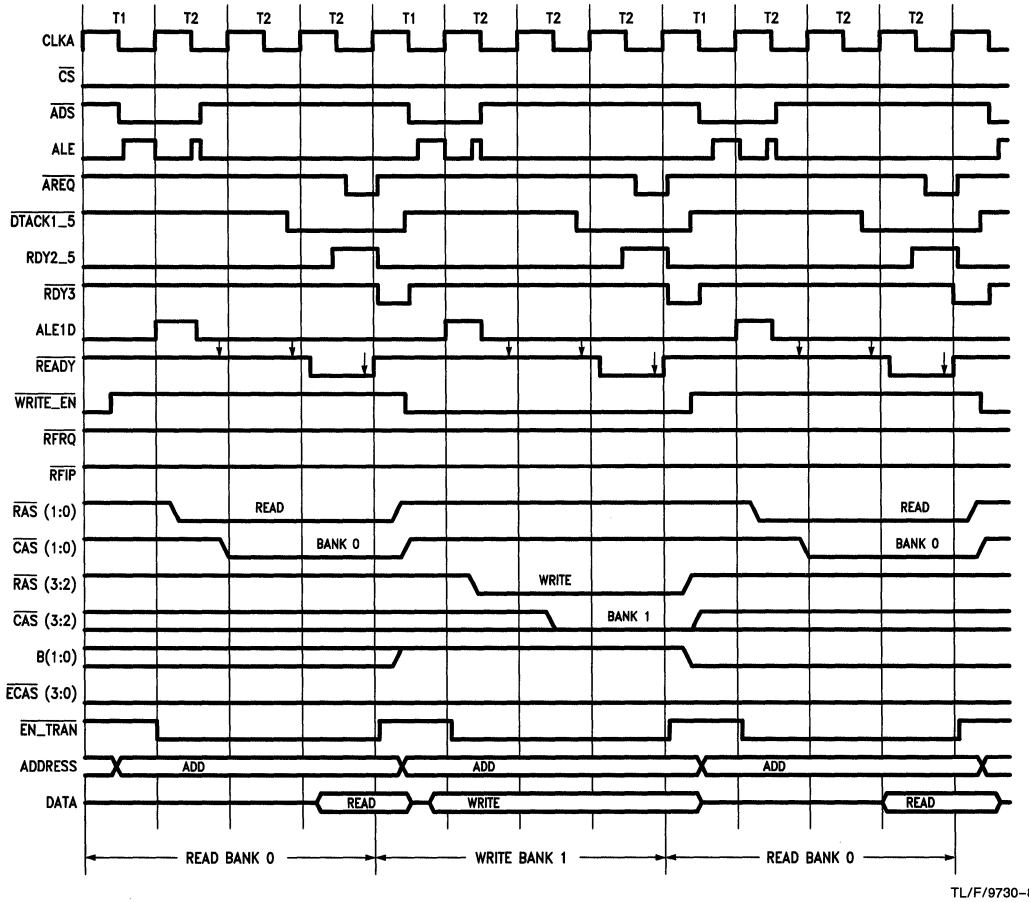
TL/F/9730-6

FIGURE 6. 80386 Design #2, System Block Diagram for Non-Address Pipelined Mode Operation at up to 50 MHz (DP8422A Uses One Half Speed Clock)



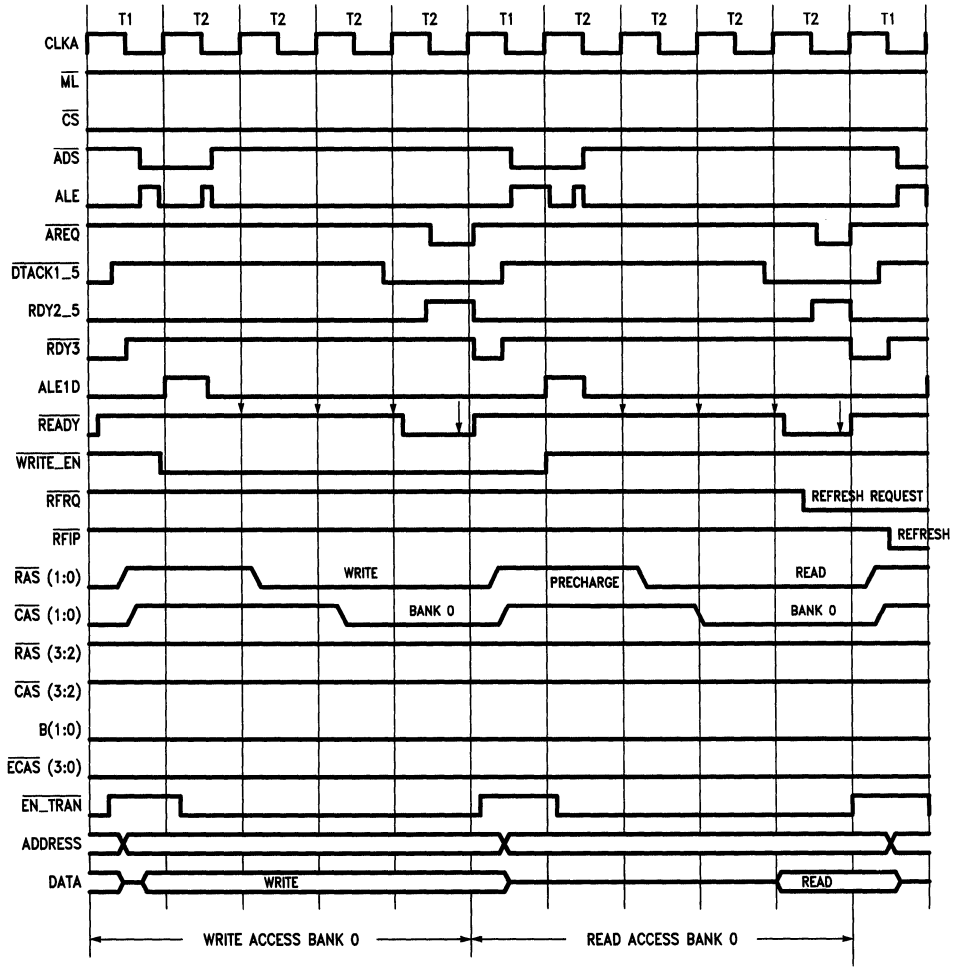
TL/F/9730–7

FIGURE 7. 80386/DP8422 Interface Control Logic for Design #2



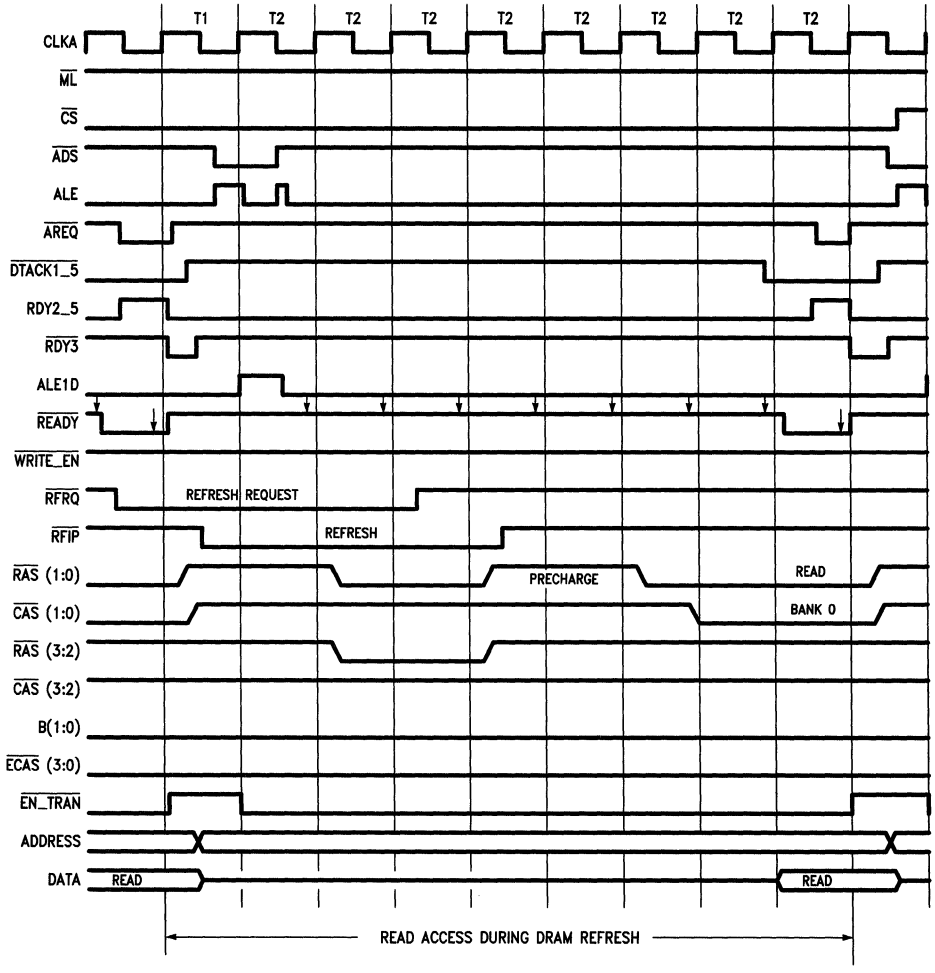
TL/F/9730–8

FIGURE 8. 80386 Design #2 Non-Address Pipelined Timing



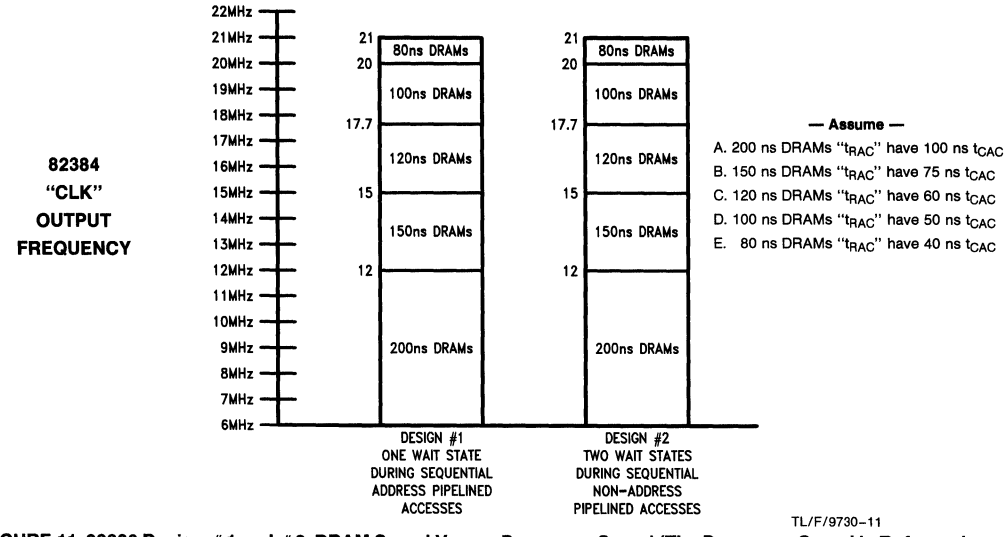
TL/F/9730–9

FIGURE 9. 80386 Design #2 Non-Address Pipelined Timing



TL/F/9730–10

FIGURE 10. 80386 Design #2 Non-Address Pipelined Timing



TL/F/9730–11

FIGURE 11. 80386 Design #1 and #2, DRAM Speed Verses Processor Speed (The Processor Speed is Referencing the 82384 "CLK" Output, The DRAM Speed is Referencing the $\overline{RAS}$ Access Time "t_{RAC}" of the DRAM)

Interfacing the DP8420A/21A/22A to the 80386 (Zero Wait State Burst Mode Access)

National Semiconductor
Application Note 619
Lawson H. C. Chang

INTRODUCTION

This application note describes how to interface the 80386 microprocessor to the DP8422A DRAM controller (also applicable to DP8420A/21A) with burst mode access. The 80386 is running at 16 MHz, 20 MHz, or 25 MHz speed. It is assumed that the reader is familiar with the 80386 and DP8422A modes operation.

DESCRIPTION

Two designs in this application note are provided to support page mode access in interfacing the DP8422A to the 80386 microprocessor. The DP8422A is operated in Mode 1 in both designs. An access cycle begins when the 80386 places a valid address on the address bus and asserts the Address Strobe (/ADS) if a refresh or Port B (DP8422A only) access is not in progress. During the burst access all /RAS's are kept low while toggling /CAS's. The burst access can be terminated when out of page signal is detected. The High Speed Access (/HSA) output signal of page detector (ALS6311) is used as an out of page signal to indicate whether the current access is in the same page as previous access or not. In other words, the row and bank select addresses have been changed from one access to the next.

I. Design #1 Description

This design simply consists of a DP8422A DRAM controller, a page detector (ALS6311), and two PALs (386PALN1 and 386PALN2). THE 386PALN1 is used to generate /CAS's and /WE signals. Where the 386PALN2 is to generate / ADS (or /AREQ), /NA, and /READY signals. This design can accommodate two banks of DRAM, 32 bits in each bank, giving a maximum memory capacity of 8 Mbytes (1M x 1 DRAMs) or 32 Mbytes (4M x 1 DRAMs). The schematic diagram is shown in *Figure 1*.

II. Design #2 Description

This design consists of the DP8422A DRAM controller, a page detector (ALS6311), a count up and down counter (F169), two 20R4D PALs, and two 16R4D PALs. The count up and down counter is to hold the number of refresh being missed. The maximum missed refreshes are six to guarantee /RAS pulse width maximum timing (t_{RASP} = 100 μs). The external refresh control logic forces DRAM controller to initiate refresh as soon as the 80386 is not accessing the memory. 386PAL1 is used to generate /ECAS(3:0) and /BED(3:0) signals. 386PAL2 is to generate /ADS, /AREQ, /NA, and some intermediate signals. 386PAL3 is to generate /MOE, /RFSHCK, and /RFIPDn signals. 386PAL4 is to generate /WE, /READY, and some intermediate signals. The schematic diagram is shown in *Figure 5*. Two designs, based upon the load capacity, are described in the following:

A. Design #2 Description for Light Load

This design interface the DP8422A to the 80386 that can accommodate two banks of DRAM, each bank being 32 bits in width, giving a maximum memory capacity of 2 Mbytes (256k by 4 DRAM). During read or write burst access cycles, zero wait state can be achieved when the 80386 is running up to 25 MHz. /MOE is tied to /OE of DRAM for /OE controlled write access. Transceivers were eliminated in this design for gaining speed. During nonburst or initial access cycles that one, two, or three wait states are required depending upon the speed of the system clock.

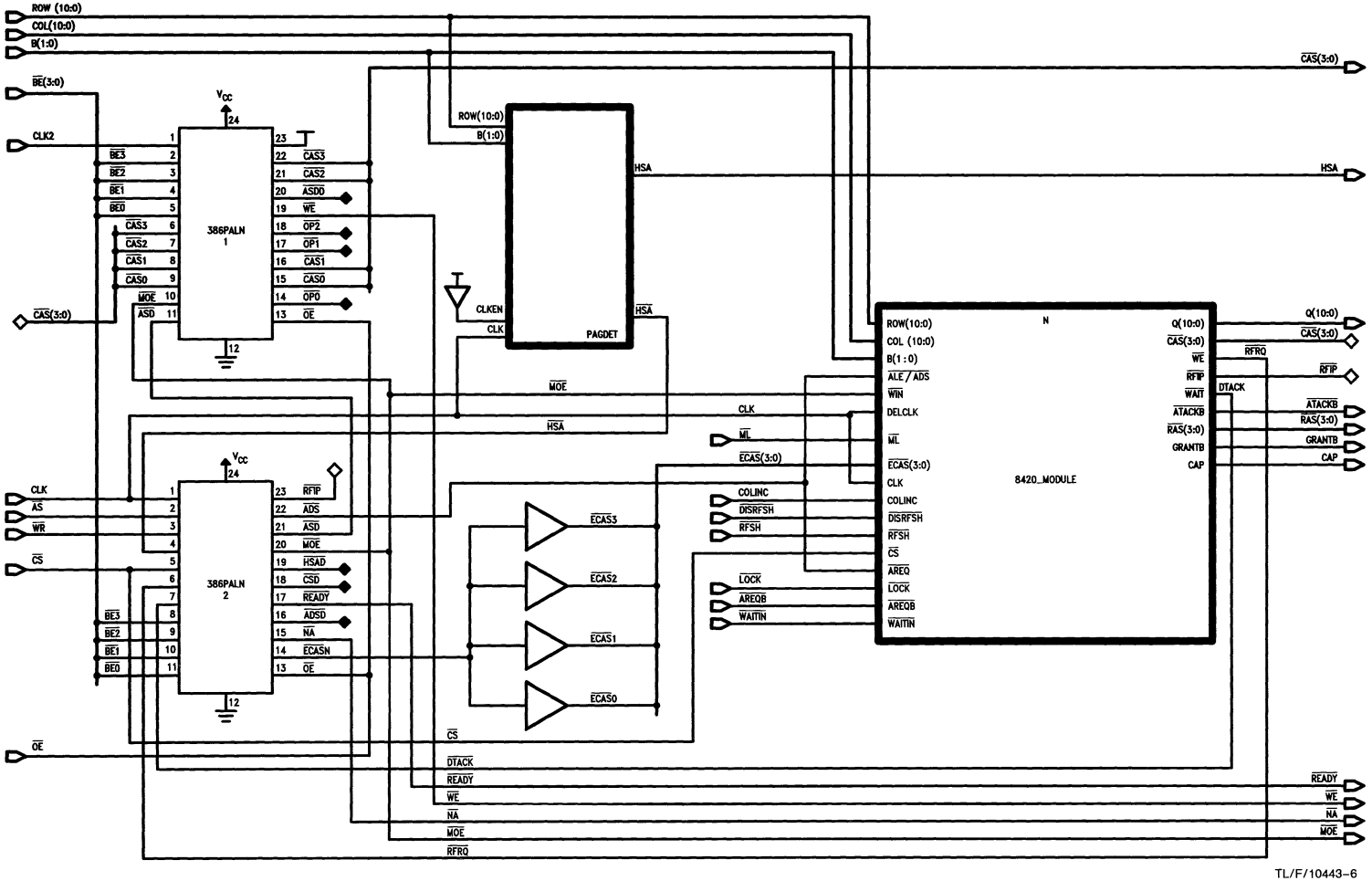
B. Design #2 Description for Heavy Load

This design is to interface the DP8422A to the 80386 and up to 8 Mbytes (1 Mbits DRAM) or 32 Mbytes (4 Mbits DRAM) memory. Zero wait state can be achieved during read burst access cycle. During write burst access cycles, one wait state has to be inserted to the 80386 bus cycle in order to guarantee data valid before /CAS going low and column address hold time after /CAS going low. One, two, or three wait states are required for microprocessor to read or write valid data during nonburst or initial access cycles. The number of required wait states depends upon the speed of the system clock.

PROGRAMMING MODE BITS FOR DESIGN #1 AND #2

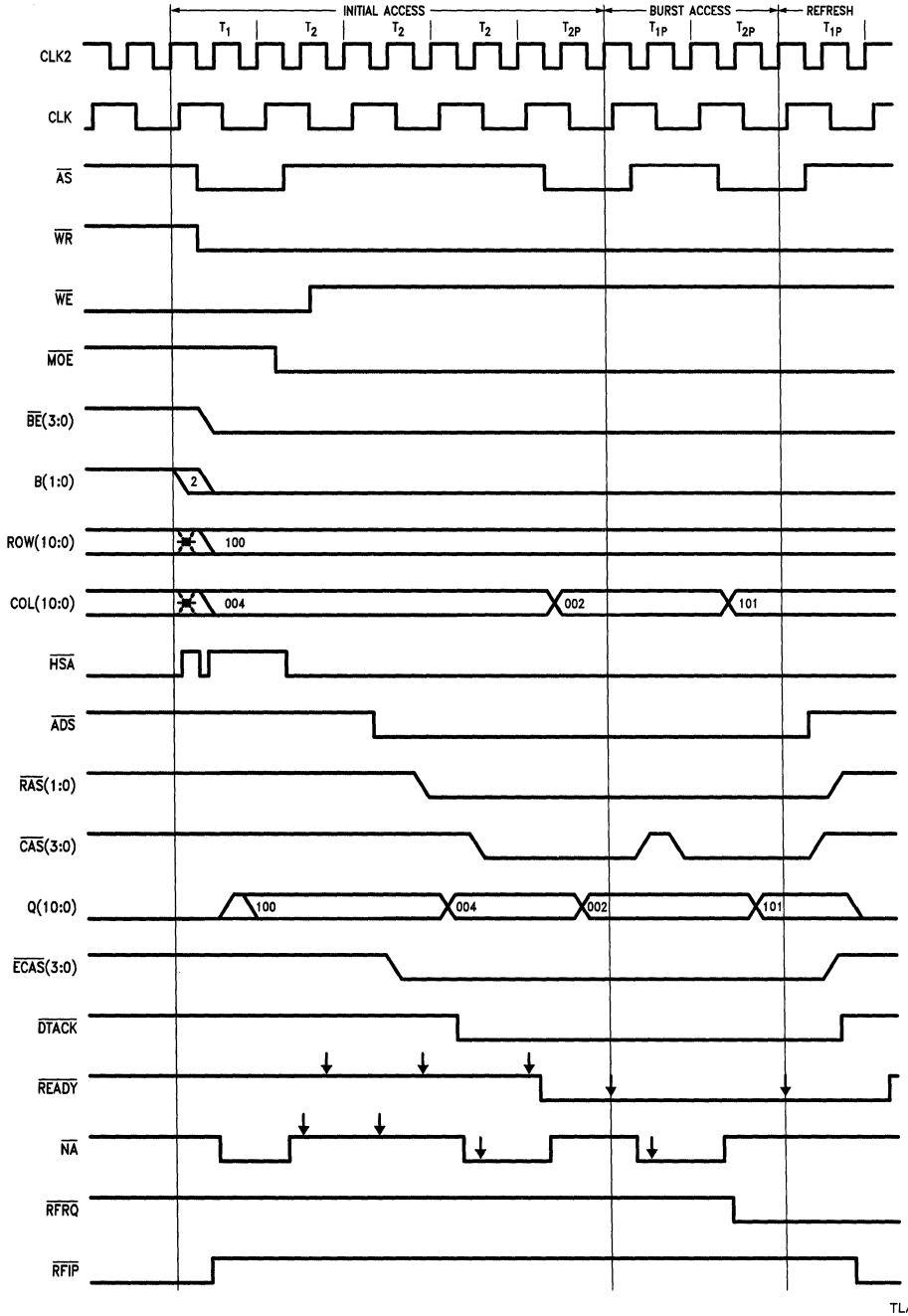
u = User Define x = Don't Care

Programming Bits		Description
R9	= u	Stagger or /RAS Refresh
R8	= 1	Noninterleaved Mode
R7	= 1	/DTACK is selected
R6	= x	/WAITIN Controlled /DTACK High
R5, R4	= 1, 1	No Wait State during Burst Mode
R3, R2	= u, u	Wait State during Nonburst Mode
R1, R0	= u, u	/RAS Low and Precharge Time
C9	= 0	/CAS is same for READ & WRITE
C8, C7	= 1, 1	t_{RAH} 15 ns and t_{ASC} 0 ns
C6, C5, C4	= u u, u	/RAS and /CAS Configuration
C3	= 0	Refresh Clock Divider
C2, C1, C0	= u, u, u	Refresh Clock Divisor Select
B1	= 1	Mode 1 Selected
B0	= 1	Fall through Selected
/ECAS0	= 1	Extend /CAS and Refresh Request



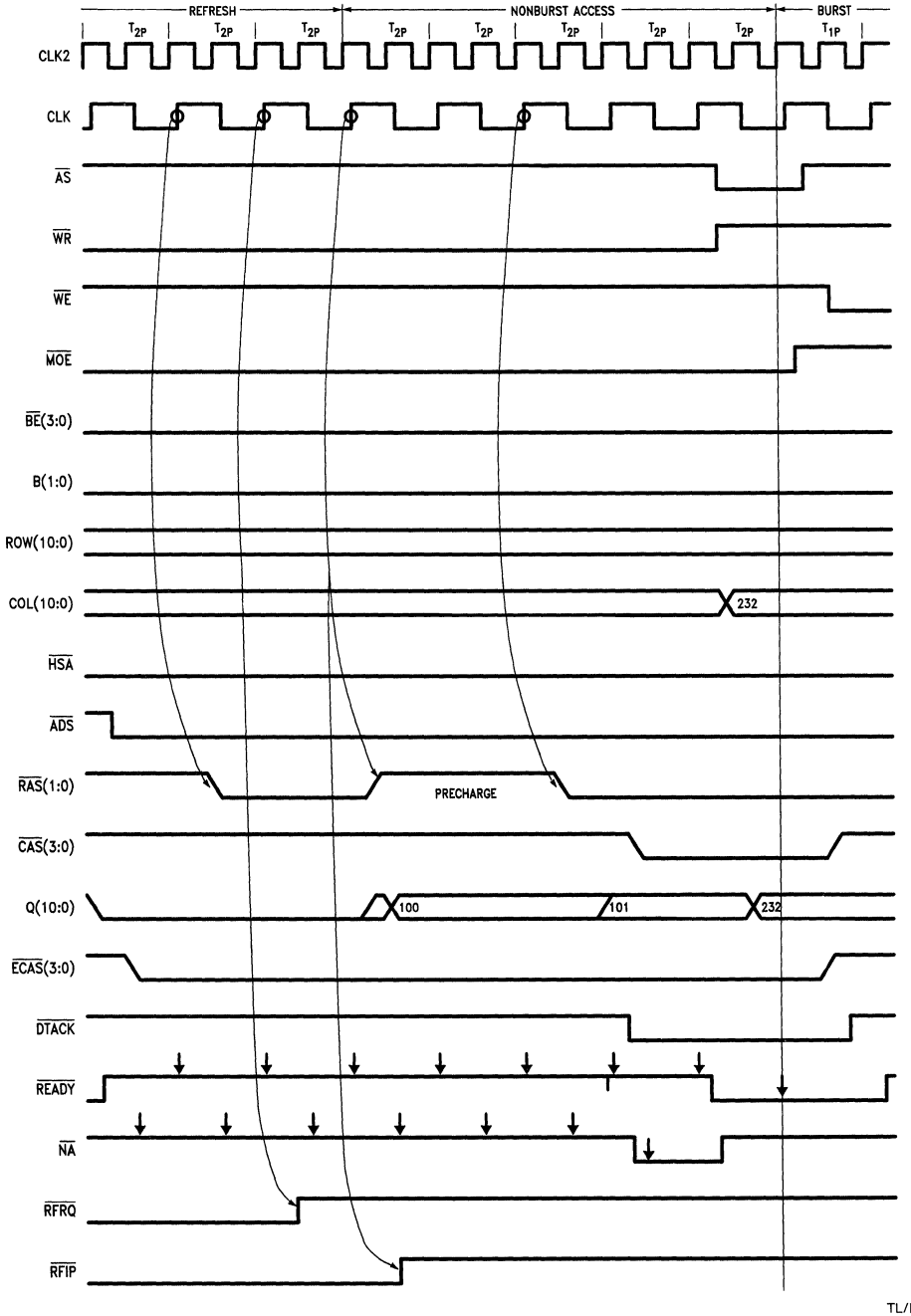
TL/F/10443–6

FIGURE 1. Schematic Diagram of Interfacing the DP8422A to the 80386 for Design # 1



TL/F/10443–7

FIGURE 2. Timing Diagram of Initial and Burst Access for Design #1



TL/F/10443–8

FIGURE 3. Timing Diagram of Refresh Cycle for Design # 1

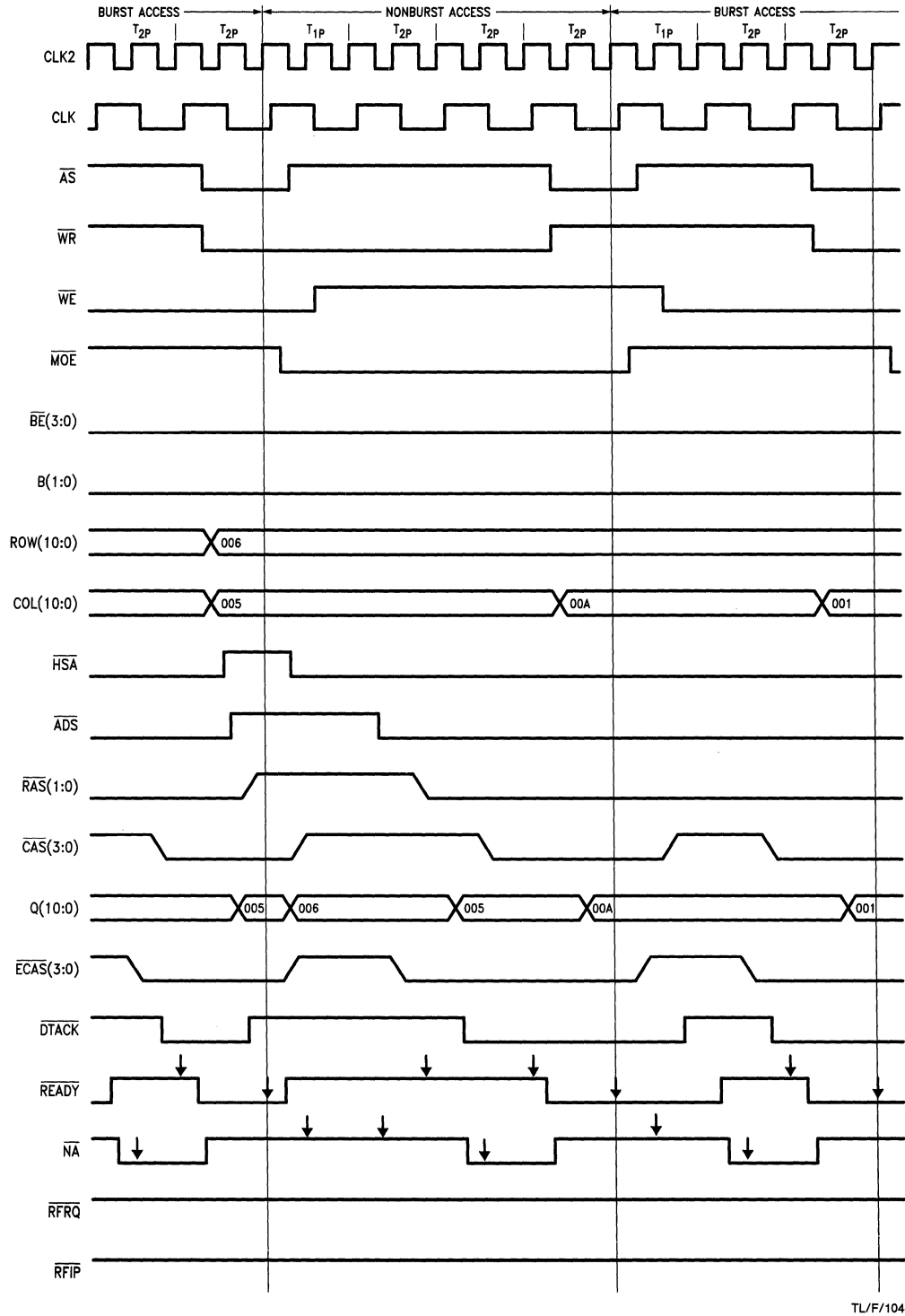
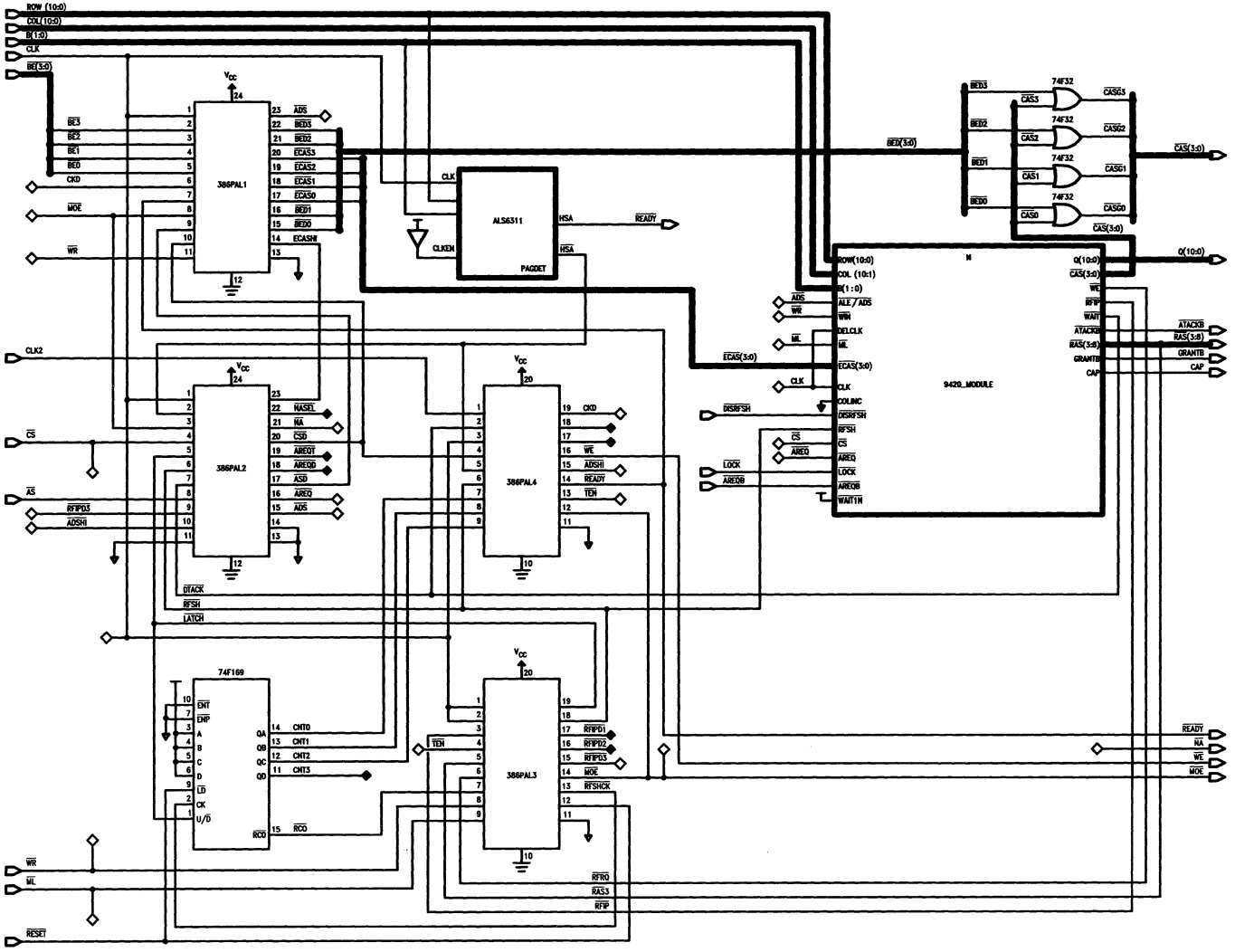


FIGURE 4. Timing Diagram of Burst and Non-Burst Access for Design # 1



TL/F/10443–1

FIGURE 5. Schematic Diagram of Interfacing the DP8422A to the 80386 for Design #2

DESIGN TIMING PARAMETERS

AC timing parameters are referred to under the heavy load when using n M x 1 or n K x 1 DRAMs, and under light load when using n M x 4 or n K x 4 DRAMs.

Times that begin with a "$" refer to DP8422A data sheet July 1988 and a "#" refer to Intel 1989 Microprocessor and Peripheral Handbook. The timing diagrams are shown in *Figure 2* through *Figure 4* and *Figure 6* through *Figure 9*. The simulation timing is based on 10 MHz clock. It may use E speed PAL for 25 MHz design.

I. Timing Calculation for Design #1

25 MHz t_{CP} = 40 ns with light load

$400b: /ADS Asserted Setup to CLK

t_{CP} − PAL20R4E CLK t_p max. − PAL20R4E t_p max.
= 40 ns − 7 ns − 8 ns = 25 ns (@ 25 MHz)

$401: /CS Setup to /ADS Asserted

2 t_{CP} + PAL20R6E CLK t_p min. + PAL20R6E t_p min. − #6 Address Valid − Decoder t_p max.
= 80 ns + 4 ns + 6 ns − 21 ns − 9 ns
= 40 ns (@ 25 MHz)

$416: /AREQ Negated to /ADS Asserted

2 t_{CP} + PAL20R6E CLK t_p min. + Skew of CLK2 and CLK min. − #6 Address Valid − /HSA t_p max.
= 80 ns + 4 ns + 3.5 ns − 21 ns − 14 ns
= 52.5 ns (@ 25 MHz)

a. Address pipelined burst mode access with 0 wait state:

t_{CAC} = 2 t_{CP} − PAL20R4E t_{CLK} max. − PAL20R4E t_p max. − #21 (Data Setup) − ½ t_{CP} − Skew of CLK2 and CLK max. − Transceiver t_p max.
= 80 ns − 7 ns − 8 ns − 7 ns − 20 ns − 10 ns − 6 ns
= 22 ns (DP8422A-25 Part)

t_{AA} = 3 t_{CP} − $26 (Address to Q Valid) − #6 (Address Valid) − #21 (Data Setup) − Transceiver t_p max.
= 120 ns − 26 ns − 21 ns − 6 ns
= 60 ns (DP8422A-25 Part)

t_{OEA} = 2 t_{CP} − PAL20R6E CLK Out t_p max. − #21 (Data Setup) − Transceiver t_p max.
= 80 ns − 7 ns − 11 ns − 6 ns
= 56 ns (DP8422A-25 Part)

b. Address pipelined nonburst mode access with 3 wait states and initial access with 4 wait states.

t_{RAC} = 4 t_{CP} − PAL20R6E t_{CLK} max. − PAL20R6E t_p max. − $402 (/ADS low to /RAS low) − #21 (Data Setup) − Skew of CLK2 and CLK t_p max. − Transceiver t_p max.
= 160 ns − 7 ns − 8 ns − 29 ns − 7 ns − 10 ns − 6 ns
= 93 ns (DP8422A-25 Part)

t_{CAC} = 4 t_{CP} − PAL20R6E t_{CLK} max. − PAL20R6E t_p max. − $403 (/ADS low to /CAS low) − #21 (Data Setup) − Skew of CLK2 and CLK t_p max. — Transceiver t_p max.
= 160 ns − 7 ns − 8 ns − 82 ns − 7 ns − 10 ns − 6 ns
= 40 ns (DP8422A-25 Part)

t_{AA} = 4 t_{CP} − PAL20R6E t_{CLK} max. − PAL20R6E t_p max. − $417 (/ADS low to Column Address valid) − #21 (Data Setup) − Skew of CLK2 and CLK t_p max. − Transceiver t_p max.
= 160 ns − 7 ns − 8 ns − 78 ns − 7 ns − 10 ns − 6 ns
= 44 ns (DP8422A-25 Part)

t_{OEA} = 5 t_{CP} − PAL20R6E t_{CLK} max. − #21 (Data Setup) − Skew CLK2 and CLK t_p max. − Transceiver t_p max.
= 200 ns − 7 ns − 7 ns − 10 ns − 6 ns
= 170 ns (DP8422A-25 Part)

II. Timing Calculation for Design #2

$400b: /ADS Asserted Setup to CLK

t_{CP} − PAL16R4D CLK t_p max. − PAL20R4D t_p max.
= 62.5 ns − 8 ns − 10 ns = 44.5 ns (@ 16 MHz)
= 50 ns − 8 ns − 10 ns = 32 ns (@ 20 MHz)
= 40 ns − 8 ns − 10 ns = 22 ns (@ 25 MHz)

$401: /CS Setup to /ADS Asserted

3 t_{CP} + PAL20R4D CLK t_p min. + PAL20R4D t_p min. − #6 Address Valid − Decoder t_p max.
= 187.5 ns + 5.5 ns + 7.1 ns − 36 ns − 9 ns
= 155 ns (@ 16 MHz)
= 150 ns + 5.5 ns + 7.1 ns − 30 ns − 9 ns
= 123.6 ns (@ 20 MHz)
= 120 ns + 5.5 ns + 7.1 ns − 21 ns − 9 ns
= 102.6 ns (@ 25 MHz)

$416: /AREQ Negated to /ADS Asserted

2 t_{CP} + PAL20R4D CLK t_p min. + Skew of CLK2 and CLK min. − #6 Address Valid − PAL20R4D t_p max.
= 125 ns + 4 ns + 3.5 ns − 36 ns − 10 ns
= 86.5 ns (@ 16 MHz)
= 100 ns + 4 ns + 3.5 ns − 30 ns − 10 ns
= 67.5 ns (@ 20 MHz)
= 80 ns + 4 ns + 3.5 ns − 21 ns − 10 ns
= 56.5 ns (@25 MHz)

A. Design #2 Light Load Timing Calculation (No Transceivers):

1. 16 MHz t_{CP} = 62.5 ns with light load

a. Address pipelined burst mode access with 0 wait state.

t_{CAC} = 2 t_{CP} − PAL20R4D t_p max. − 74F32 t_p max. − #21 (Data Setup) − ½ t_{CP}
= 125 ns − 10 ns − 6 ns − 11 ns − 31 ns
= 67 ns (DP8422A-20 and DP8422A-25 Part)

t_{AA} = 3 t_{CP} − $26 (Address to Q Valid) − #6 (Address Valid) − #21 (Data Setup)
= 187.5 ns − 26 ns − 36 ns − 11 ns
= 114.5 ns (DP8422A-25 Part)
= 187.5 ns − 29 ns − 36 ns − 11 ns
= 111.5 ns (DP8422A-20 Part)

t_{OEA} = 2 t_{CP} − PAL16R4D CLK Out t_p max. − #21 (Data Setup)
= 125 ns − 8 ns − 11 ns
= 106 ns (DP8422A-20 and DP8422A-25 Part)

b. Address pipelined nonburst mode access with 2 wait states.

t_{RAC} = 3 t_{CP} − $307 (CLK High to /RAS Low) − #21 (Data Setup)
= 187.5 ns − 22 ns − 11 ns
= 154 ns (DP8422A-25 Part)
= 187.5 ns − 27 ns − 11 ns
= 149.5 ns (DP8422A-20 Part)

t_{CAC} = 3 t_{CP} − $308 (CLK High to /CAS Low) − #21 (Data Setup) − 74F32 t_p max.
= 187.5 ns − 72 ns − 11 ns − 6 ns
= 98.5 ns (DP8422A-25 Part)
= 187.5 ns − 81 ns − 11 ns − 6 ns
= 89.5 ns (DP8422A-20 Part)

t_{AA} = 3 t_{CP} − $316 (CLK High to Column Address Valid) − #21 (Data Setup)
= 187.5 ns − 66 ns − 11 ns
= 110.5 ns (DP8422A-25 Part)
= 187.5 ns − 78 ns − 11 ns
= 98.5 ns (DP8422A-20 Part)

t_{OEA} = 4 t_{CP} − PAL16R4D CLK out t_p max. − #21 (Data Setup)
= 250 ns − 8 ns − 11 ns
= 231 ns (DP8422A-20 and DP8422A-25 Part)

c. Initial Access with 3 Wait States.

t_{RAC} = 3 t_{CP} − PAL20R4D CLK t_p max. − $402 (/ADS Low to /RAS Low) − #21 (Data Setup)
= 187.5 ns − 8 ns − 25 ns − 11 ns
= 143.5 ns (DP8422A-25 Part)
= 187.5 ns − 8 ns − 30 ns − 11 ns
= 138.5 ns (DP8422A-20 Part)

t_{CAC} = 3 t_{CP} − PAL20R4D CLK t_p max. − $403 (/ADS Low to /CAS Low) − #21 (Data Setup) − 74F32 t_p Max.
= 187.5 ns − 8 ns − 75 ns − 11 ns − 6 ns
= 87.5 ns (DP8422A-25 Part)
= 187.5 ns − 8 ns − 86 ns − 11 ns − 6 ns
= 76.5 ns (DP8422A-20 Part)

t_{AA} = 3 t_{CP} − PAL20R4D CLK t_p max. − $417 (/ADS Low to Column Address Valid) − #21 (Data Setup)
= 187.5 ns − 8 ns − 69 ns − 11 ns
= 99.5 ns (DP8422A-25 Part)
= 187.5 ns − 8 ns − 83 ns − 11 ns
= 85.5 ns (DP8422A-20 Part)

t_{OEA} = 3 t_{CP} − PAL16R4D CLK Out t_p max. − #21 (Data Setup)
= 187.5 ns − 8 ns −11 ns
= 168.5 ns (DP8422A-20 and DP8422A-25 Part)

2. 20 MHz t_{CP} = 50 ns with light load

a. Address pipelined burst mode access with 0 wait state.

t_{CAC} = 2 t_{CP} − PAL20R4D t_p max. − 74F32 t_p max. − #21 (Data Setup) − ½ t_{CP}.
= 100 ns − 10 ns − 6 ns − 11 ns − 25 ns
= 48 ns (DP8422A-20 and DP8422A-25 Part)

t_{AA} = 3 t_{CP} − $26 (Address to Q Valid) − #6 (Address Valid) − #21 (Data Setup)
= 150 ns − 26 ns − 30 ns − 11 ns
= 83 ns (DP8422A-25 Part)
= 150 ns − 29 ns − 30 ns − 11 ns
= 80 ns (DP8422A-20 Part)

t_{OEA} = 2 t_{CP} − PAL16R4D CLK out t_p max − #21 (Data Setup)
= 100 ns − 8 ns − 11 ns
= 81 ns (DP8422A-20 and DP8422A-25 Part)

b. Address pipelined nonburst mode access with 2 wait states.

t_{RAC} = 3 t_{CP} − $307 (CLK High to /RAS Low) − #21 (Data Setup)
= 150 ns − 22 ns − 11 ns
= 117 ns (DP8422A-25 Part)
= 150 ns − 27 ns − 11 ns
= 112 ns (DP8422A-20 Part)

t_{CAC} = 3 t_{CP} − $308 (CLK High to /CAS Low) − #21 (Data Setup) − 74F32 t_p max.
= 150 ns − 72 ns − 11 ns − 6 ns
= 61 ns (DP8422A-25 Part)
= 150 ns − 81 ns − 11 ns − 6 ns
= 52 ns (DP8422A-20 Part)

t_{AA} = 3 t_{CP} − $316 (CLK High to Column Address Valid) − #21 (Data Setup)
= 150 ns − 66 ns − 11 ns
= 73 ns (DP8422A-25 Part)
= 150 ns − 78 ns − 11 ns
= 61 ns (DP8422A-20 Part)

t_{OEA} = 4 t_{CP} − PAL16R4D CLK out t_p max. − #21 (Data Setup)
= 200 ns − 8 ns − 11 ns
= 181 ns (DP8422A-20 and DP8422A-25 Part)

c. Initial access with 3 wait states.

t_{RAC} = 3 t_{CP} − PAL20R4D CLK t_p max. − $402 (/ADS Low to /RAS Low) − #21 (Data Setup)
= 150 ns − 8 ns − 25 ns − 11 ns
= 106 ns (DP8422A-25 Part)
= 150 ns − 8 ns − 30 ns − 11 ns
= 101 ns (DP8422A-20 Part)

t_{CAC} = 3 t_{CP} − PAL20R4D CLK t_p max. − $403 (/ADS Low to /CAS Low) − #21 (Data Setup) − 74F32 t_p max.
= 150 ns − 8 ns − 75 ns − 11 ns − 6 ns
= 50 ns (DP8422A-25 Part)
= 150 ns − 8 ns − 86 ns − 11 ns − 6 ns
= 39 ns (DP8422A-20 Part)

t_{AA} = 3 t_{CP} − PAL20R4D CLK t_p max. $417 (/ADS Low to Column Address Valid) − #21 (Data Setup)
= 150 ns − 8 ns − 69 ns − 11 ns
= 62 ns (DP8422A-25 Part)
= 150 ns − 8 ns − 83 ns − 11 ns
= 48 ns (DP8422A-20 Part)

t_{OEA} = 3 t_{CP} − PAL16R4D CLK Out t_p max. − #21 (Data Setup)
= 150 ns − 8 ns − 11 ns
= 131 ns (DP8422A-20 and DP8422A-25 Part)

3. 25 MHz t_{CP} = 40 ns with light load

a. Address pipelined burst mode access with 0 wait state.

t_{CAC} = 2 t_{CP} − PAL20R4D t_p max. − 74F32 t_p max. − #21 (Data Setup) − ½ t_{CP}.
= 80 ns − 10 ns − 6 ns − 7 ns − 20 ns
= 37 ns (DP8422A-25 Part)

t_{AA} = 3 t_{CP} − $26 (Address to Q Valid) − #6 (Address Valid) − #21 (Data Setup)
= 120 ns − 26 ns − 21 ns − 7 ns
= 83 ns (DP8422A-25 Part)

t_{OEA} = 2 t_{CP} − PAL16R4D CLK Out t_p max. − #21 (Data Setup)
= 80 ns − 8 ns − 11 ns
= 65 ns (DP8422A-25 Part)

b. Address pipelined nonburst mode access with 2 wait states.

t_{RAC} = 3 t_{CP} − $307 (CLK High to /RAS Low) − #21 (Data Setup)
= 120 ns − 22 ns − 7 ns
= 91 ns (DP8422A-25 Part)

t_{CAC} = 3 t_{CP} − $308 (CLK High to /CAS Low) − #21 (Data Setup) − 74F32 t_p max.
= 120 ns − 72 ns − 7 ns − 6 ns
= 35 ns (DP8422A-25 Part)

t_{AA} = 3 t_{CP} − $316 (CLK High to Column Address Valid) − #21 (Data Setup)
= 120 ns − 66 ns − 7 ns
= 47 ns (DP8422A-25 Part)

t_{OEA} = 4 t_{CP} − PAL16R4D CLK Out t_p max. − #21 (Data Setup)
= 160 ns − 8 ns − 7 ns
= 145 ns (DP8422A-25 Part)

c. Initial access with 4 wait states.

t_{RAC} = 4 t_{CP} − PAL20R4D CLK t_p max. − $402 (/ADS Low to /RAS Low) − #21 (Data Setup)
= 160 ns − 8 ns − 25 ns − 7 ns
= 120 ns (DP8422A-25 Part)

t_{CAC} = 4 t_{CP} − PAL20R4D CLK t_p max. − $403 (/ADS Low to /CAS Low) − #21 (Data Setup) − 74F32 t_p max.
= 160 ns − 8 ns − 75 ns − 7 ns − 6 ns
= 64 ns (DP8422A-25 Part)

t_{AA} = 4 t_{CP} − PAL20R4D CLK t_p max. − $417 (/ADS Low to Column Address Valid) − #21 (Data Setup)
= 160 ns − 8 ns − 69 ns − 7 ns
= 76 ns (DP8422A-25 Part)

t_{OEA} = 4 t_{CP} − PAL16R4D CLK Out t_p max. − #21 (Data Setup)
= 160 ns − 8 ns − 7 ns
= 145 ns (DP8422A-25 Part)

B. Design #2 Heavy Load Timing Calculation:

1. 16 MHz t_{CP} = 62.5 ns with Heavy Load

a. Address pipelined burst mode access with 0 wait state.

t_{CAC} = 2 t_{CP} − PAL20R4D t_p max. − 74F32 t_p max. − #21 (Data Setup) − ½ t_{CP} Transceiver t_p max.
= 125 ns − 10 ns − 8 ns − 11 ns − 31 ns − 7 ns
= 48 ns (DP8422A-20 and DP8422A-25 Part)

t_{AA} = 3 t_{CP} − $26 (Address to Q Valid) − #6 (Address Valid) − #21 (Data Setup) − Transceiver t_p max.
= 187.5 ns − 35 ns − 36 ns − 11 ns − 7 ns
= 98.5 ns (DP8422A-25 Part)
= 187.5 ns − 38 ns − 36 ns − 11 ns − 7 ns
= 95.5 ns (DP8422A-20 Part)

t_{OEA} = 2 t_{CP} − PAL16R4D CLK Out t_p Max. − #21 (Data Setup) − Transceiver t_p max.
= 125 ns − 8 ns − 11 ns − 7 ns
= 89 ns (DP8422A-20 and DP8422A-25 Part)

b. Address pipelined nonburst mode access with 2 wait states.

t_{RAC} = 3 t_{CP} − $307 (CLK High to /RAS Low) − #21 (Data Setup) − Transceiver t_p max.
= 187.5 ns − 26 ns − 11 ns − 7 ns
= 143.5 ns (DP8422A-25 Part)
= 187.5 ns − 32 ns − 11 ns − 7 ns
= 137.5 ns (DP8422A-20 Part)

t_{CAC} = 3 t_{CP} − $308 (CLK High to /CAS Low) − #21 (Data Setup) − 74F32 t_p max. − Transceiver t_p max.
= 187.5 ns − 72 ns − 11 ns − 8 ns − 7 ns
= 89.5 ns (DP8422A-25 Part)
= 187.5 ns − 81 ns − 11 ns − 8 ns − 7 ns
= 80.5 ns (DP8422A-20 Part)

t_{AA} = 3 t_{CP} − $316 (CLK High to Column Address Valid) − #21 (Data Setup) − Transceiver t_p max.
= 187.5 ns − 75 ns − 11 ns − 7 ns
= 94.5 ns (DP8422A-25 Part)
= 187.5 ns − 87 ns − 11 ns − 7 ns
= 82.5 ns (DP8422A-20 Part)

t_{OEA} = 4 t_{CP} − PAL16R4D CLK Out t_p max. − #21 (Data Setup) − Transceiver t_p max.
= 250 ns − 8 ns − 11 ns − 7 ns
= 224 ns (DP8422A-20 and DP8422A-25 Part)

c. Initial Access with 3 Wait States.

t_{RAC} = 3 t_{CP} − PAL20R4D CLK t_p max. − $402 (/ADS Low to /RAS Low) − #21 (Data Setup) − Transceiver t_p max.
= 187.5 ns − 8 ns − 29 ns − 11 ns − 7 ns
= 132.5 ns (DP8422A-25 Part)
= 187.5 ns − 8 ns − 35 ns − 11 ns − 7 ns
= 126.5 ns (DP8422A-20 Part)

t_{CAC} = 3 t_{CP} − PAL20R4D CLK t_p max. − $403 (/ADS Low to /CAS Low) − #21 (Data Setup) − 74F32 t_p max. − Transceiver t_p max.
= 187.5 ns − 8 ns − 75 ns − 11 ns − 8 ns − 7 ns
= 78.5 ns (DP8422A-25 Part)
= 187.5 ns − 8 ns − 86 ns − 11 ns − 8 ns − 7 ns
= 67.5 ns (DP8422A-20 Part)

t_{AA} = 3 t_{CP} − PAL20R4D CLK t_p max. − $417 (/ADS Low to Column Address Valid) − #21 (Data Setup) − Transceiver t_p max.
= 187.5 ns − 8 ns − 78 ns − 11 ns − 7 ns
= 83.5 ns (DP8422A-25 Part)
= 187.5 ns − 8 ns − 92 ns − 11 ns − 7 ns
= 69.5 ns (DP8422A-20 Part)

t_{OEA} = 3 t_{CP} − PAL16R4D CLK Out t_p max. − #21 (Data Setup) − Transceiver t_p max.
= 187.5 ns − 8 ns − 11 ns − 7 ns
= 161.5 ns (DP8422A-20 and DP8422A-25 Part)

2. 20 MHz t_{CP} = 50 ns with heavy load

a. Address pipelined burst mode access with 0 wait state.

t_{CAC} = 2 t_{CP} − PAL20R4D t_p max. − 74F32 t_p max. − #21 (Data Setup) − ½ t_{CP} − Transceiver t_p max.
= 100 ns − 10 ns − 8 ns − 11 ns − 25 ns − 7 ns
= 29 ns (DP8422A-20 and DP8422A-25 Part)

t_{AA} = 3 t_{CP} − $26 (Address to Q Valid) − #6 (Address Valid) − #21 (Data Setup) − Transceiver t_p max.
= 150 ns − 35 ns − 30 ns − 11 ns − 7 ns
= 67 ns (DP8422A-25 Part)
= 150 ns − 38 ns − 30 ns − 11 ns − 7 ns
= 64 ns (DP8422A-20 Part)

t_{OEA} = 2 t_{CP} − PAL16R4D CLK Out t_p max. − #21 (Data Setup) − Transceiver t_p max.
= 100 ns − 8 ns − 11 ns − 7 ns
= 74 ns (DP8422A-20 and DP8422A-25 Part)

b. Address pipelined nonburst mode access with 2 wait states

t_{RAC} = 3 t_{CP} − $307 (CLK High to /RAS Low) − #21 (Data Setup) − Transceiver t_p max.
= 150 ns − 26 ns − 11 ns − 7 ns
= 106 ns (DP8422A-25 Part)
= 150 ns − 32 ns − 11 ns − 7 ns
= 100 ns (DP8422A-20 Part)

t_{CAC} = 3 t_{CP} − $308 (CLK High to /CAS Low) − #21 (Data Setup) − 74F32 t_p max. − Transceiver t_p max.
= 150 ns − 72 ns − 11 ns − 8 ns − 7 ns
= 52 ns (DP8422A-25 Part)
= 150 ns − 81 ns − 11 ns − 8 ns − 7 ns
= 43 ns (DP8422A-20 Part)

t_{AA} = 3 t_{CP} − $316 (CL High to Column Address Valid) − #21 (Data Setup) − Transceiver t_p max.
= 150 ns − 75 ns − 11 ns − 7 ns
= 57 ns (DP8422A-25 Part)
= 150 ns − 87 ns − 11 ns − 7 ns
= 45 ns (DP8422A-20 Part)

t_{OEA} = 4 t_{CP} − PAL16R4D CLK Out t_p max. − #21 (Data Setup) − Transceiver t_p max.
= 200 ns − 8 ns − 11 ns − 7 ns
= 174 ns (DP8422A-20 and DP8422A-25 Part)

c. Initial access with 4 wait states.

t_{RAC} = 4 t_{CP} − PAL20R4D CLK t_p max. − $402 (/ADS Low to /RAS Low) − #21 (Data Setup) − Transceiver t_p max.
= 200 ns − 8 ns − 29 ns − 11 ns − 7 ns
= 145 ns (DP8422A-25 Part)
= 200 ns − 8 ns − 35 ns − 11 ns − 7 ns
= 139 ns (DP8422A-20 Part)

t_{CAC} = 4 t_{CP} − PAL20R4D CLK t_p max. − $403 (/ADS Low to /CAS Low) − #21 (Data Setup) − 74F32 t_p max. − Transceiver t_p max.
= 200 ns − 8 ns − 75 ns − 11 ns − 8 ns − 7 ns
= 91 ns (DP8422A-25 Part)
= 200 ns − 8 ns − 86 ns − 11 ns − 8 ns − 7 ns
= 80 ns (DP8422A-20 Part)

t_{AA} = 4 t_{CP} − PAL20R4D CLK t_p max. − $417 (/ADS Low to Column Address Valid) − #21 (Data Setup) − Transceiver t_p max.
= 200 ns − 8 ns − 78 ns − 11 ns − 7 ns
= 96 ns (DP8422A-25 Part)
= 200 ns − 8 ns − 92 ns − 11 ns − 7 ns
= 82 ns (DP8422A-20 Part)

t_{OEA} = 4 t_{CP} − PAL16R4D CLK Out t_p max. − #21 (Data Setup) − Transceiver t_p max.
= 200 ns − 8 ns − 11 ns − 7 ns
= 174 ns (DP8422A-20 and DP8422A-25 Part)

PAL EQUATIONS

The Boolean entry operators are listed as:

"=" Equality
":=" Replaced by (After Clock)
"*" AND
"+" OR
"/" Complement
"N" Active Low

The brief explanation of PAL output signals

CAS(3:0)~ These combinational output signals are Column Address Strobes

ASDD~ This sequential output signal is ASD~ Delayed by one CLK2 clock.

HSAD~ This sequential output signal is HSA~ Delayed by one clock.

ADSD~ This sequential output signal is ADS~ Delayed by one clock.

BED(3:0)~ These combinational output signals are used to toggle CAS of DRAM directly during burst and nonburst access cycles.

ECAS~(3:0) These sequential output signals are used to hold CAS low during the burst access except design #2 heavy load burst write access cycles.

ADS~ This combinational output signal is Address Strobe to the DP8422A.

AREQ~ This combinational output signal is Access Request to the DP8422A.

NASEL~ This combinational output signal selects Next Address from either initial or noninitial access cycles.

NA~ This combinational output signal is Next Address to the 80386.

CSD~ This sequential output signal is Chip Select Delayed by one clock.

AREQT~ This sequential output signal is Access Request Transition that holds HSA~ (High Speed Access) one clock during CSD~ is low.

AREQD~ This sequential output signal is Access Request Delayed by one clock.

ASD~ This sequential output signal is Address Strobe Delayed by one clock.

WE~ This combinational output signal is Write Enable to the DRAM.

LATCH~ This combinational output signal is to hold external refresh request.

RFSH~ This combinational output signal is complement of LATCH~.

RFSHCK~ This combinational output signal is used to clock count up and down counter.

RFIPD1~ This sequential output signal is Refresh In Progress Delayed by one clock.

RFIPD2~ This sequential output signal is Refresh In Progress Delayed by two clocks.

RFIPD3~ This sequential output signal is Refresh in Progress Delayed by three clocks.

MOE~ This sequential output signal is Memory Output Enable to control the data output of the DRAM.

TEN~ This combinational output signal indicates that six missed refreshes are reached.

CKD This combinational output signal is normal CLK delayed by PAL.

READY~ This sequential output signal is delayed DTACK~ by one clock.

ADSHI~ This sequential output signal indicates whether the 80386 is accessing the DRAM or the DP8422A is refreshing the DRAM.

Ia. 386PALN1 (PAL20R4E) for Design #1

Inputs: CLK2, BE3~, BE2~, BE1~, BE0~, CAS~3, CAS~2, CAS~1, CAS~0, MOE~, ASD~;

Outputs:

```
/CAS3~   = (/ASDD~ * /CAS~3) + (ASD~ * /CAS~3) + (MOE~ * /CAS~3) + (BE3~ * /CAS~3);
/CAS2~   = (/ASDD~ * /CAS~2) + (ASD~ * /CAS~2) + MOE~ * /CAS~2) + (BE2~ * /CAS~2);
/CAS1~   = (/ASDD~ * /CAS~1) + (ASD~ * /CAS~1) + MOE~ * /CAS~1) + (BE1~ * /CAS~1);
/CAS0~   = (/ASDD~ * /CAS~0) + (ASD~ * /CAS~0) + MOE~ * /CAS~0) + (BE0~ * /CAS~0);
/ASDD~   := /ASD~;
/WE~     := MOE~;
```

Ib. 386PALN2 (PAL20R6E) for Design # 1

Inputs: CLK, BE3~, BE2~, BE1~, BE0~, CS~, HSA~, RFRQ~, RFIP~, DTACK~, AS~, WR~;

Outputs:

```
/ADS~    = (/HSAD~ * RFRQ~ * /HHA~) + (ASD~ * /RFRQ~);
/NA~     = (READY~ * /DTACK~ * RFRQ~ * RFIP~) + (READY~ * MOE~ * RFRQ~ * RFIP~) + (/READY~ * RFRQ~ * /MOE~ * /ASD~ * RFIP~);
/ECAS~N  = (ASD~ * /HSAD~ * RFRQ~) + (BE3~ * BE2~ * BE1~ * BE0~ * /HSAD~ * RFRQ~) + (/MOE~ * /HSAD~ * RFRQ~) + (CSD~ * /HSAD~ * RFRQ~) + (ASD~ * /RFRQ~);
/CSD~    := /CS~;
/ASD~    := /AS~;
/MOE~    := /WR~;
/HSAD~   := /HSA~ * /CSD~;
/READY~  := /DTACK~;
/ADSD~   := /ADS~;
```

II a. 386PAL1 (PAL20R4D) for Design #2 Light Load

Inputs: CLK, BE3~, BE2~, BE1~, BE0~, READY~, CKD, MOE~, ASD~, CSD~, WR~, ADS~;

Outputs:

```
/BED3~   = (READY~ * /ECAS~3) + (/ASD~ * /ECAS~3) + (/CKD * /ECAS~3);
```

```
/BED2~    = (READY~ * /ECAS~2) + (/ASD~ *
            /ECAS~2) + (/CKD * /ECAS~2);
/BED1~    = (READY~ * /ECAS~1) + (/ASD~ *
            /ECAS~1) + (/CKD * /ECAS~1);
/BED0~    = (READY~ * /ECAS~0) + (/ASD~ *
            /ECAS~0) + (/CKD * /ECAS~0);
/ECAS~3   := (/BE3~ * /CSD~ * READY~)
             + (/BE3~ * /CSD~ * /ADS~);
/ECAS~2   := (/BE2~ * /CSD~ * READY~)
             + (/BE2~ * /CSD~ * /ADS~);
/ECAS~1   := (/BE1~ * /CSD~ * READY~)
             + (/BE1~ * /CSD~ * /ADS~);
/ECAS~0   := (/BE0~ * /CSD~ * READY~)
             + (/BE0~ * /CSD~ * /ADS~);
/ECASHI   := (/ECAS~3 + /ECAS~2
             + /ECAS~1 + /ECAS~0;
```

IIb. 386PAL1 (PAL20R4D) for Design #2 Heavy Load

Inputs: CLK, BE3~, BE2~, BE1~, BE0~, READY~, CKD, MOE~, ASD~, CSD~, WR~, ADS~;

Outputs:

```
/BED3~    = (READY~ * /ECAS~3) + (ASD~ *
            /ECAS~3) + (/CKD * /ECAS~3)
             + (MOE~ * ECAS~3);
/BED2~    = (READY~ * /ECAS~2) + (ASD~*
            /ECAS~2) + (/CKD * /ECAS~2)
             + (MOE~ * ECAS~2);
/BED1~    = (READY~ * /ECAS~1) + (ASD~ *
            /ECAS~1) + (/CKD * /ECAS~1)
             + (MOE~ * ECAS~1);
/BED0~    = (READY~ * /ECAS~0) + (ASD~ *
            /ECAS~0) + (/CKD * /ECAS~0)
             + (MOE~ * ECAS~0);
/ECAS~3:= (/BE3~ * /CSD~ * READY~) +
          (/BE3~ * /CSD~ * READY~ * /ASD~)
          +
          (/BE3~ * /CSD~ * READY~ * /WR~)
          +
          (/BE3~ * /CSD~ * /ADS~ * READY~)
          +
          (/BE3~ * /CSD~ * /ADS~) +
          (/BE3~ * /CSD~ * /ADS~ * /WR~);
/ECAS~2:= (/BE2~ * /CSD~ * READY~) +
          (/BE2~ * /CSD~ * READY~ * /ASD~)
          +
          (/BE2~ * /CSD~ * READY~ * /WR~)
          +
          (/BE2~ * /CSD~ * /ADS~ * READY~)
          +
          (/BE2~ * /CSD~ * /ADS~) +
          (/BE2~ * /CSD~ * /ADS~ * /WR~);
/ECAS~1:= (/BE1~ * /CSD~ * READY~) +
          (/BE1~ * /CSD~ * READY~ * /ASD~)
          +
          (/BE1~ * /CSD~ * READY~ * /WR~)
          +
          (/BE1~ * /CSD~ * /ADS~ * READY~)
          +
          (/BE1~ * /CSD~ * /ADS~) +
          (/BE1~ * /CSD~ * /ADS~ * /WR~);
/ECAS~0:= (/BE0~ * /CSD~ * READY~) +
          (/BE0~ * /CSD~ * READY~ * /ASD~)
          +
          (/BE0~ * /CSD~ * READY~ * /WR~)
          +
          (/BE0~ * /CSD~ * /ADS~ * READY~)
          +
          (/BE0~ * /CSD~ * /ADS~) +
          (/BE0~ * /CSD~ * /ADS~ * /WR~);
/ECASHI:= /ECAS~3 + /ECAS~2 + /ECAS~1 +
          /ECAS~0;
```

IIc. 386PAL2 (PAL20R4D) for Design #2 Light and Heavy Load

Inputs: CLK, CS~, HSA~, LATCH~, RFSH~, DTACK~, AS~, RFIPD3~, MOE~, ECASHI;

Outputs:

```
/ADS~     = (/CS~ * /AS~) + (/AREQT~ *
            /LATCH~);
/AREQ~    = (/HSA~ * CSD~ * /AREQT~ * /CS~
            * AS~) + (/HSA~ * CSD~ *
            /AREQT~ * /LATCH~) + (/RFSH~ *
            /CS~ * AS~) + (/RFSH~ *
            /LATCH~);
/NASEL~   = (/CS~ * /DTACK~ * /AS~) + (/NA-
            SEL~ * CS~);
/NA~      = (NASEL~ * /HSA~ * /CSD~ *
            /AREQT~) + (/NASEL~ * /ECASHI *
            MOE~) + (/NASEL~ * /AREQD~ *
            /MOE~ * RFSH~ * RFIPD3~);
/CSD~     := /CS~;
/AREQT~:= /HSA~ * /CSD~;
/AREQD~:= /AREQ~;
/ASD~     := /AS~;
```

IId. 386PAL3 (PAL16R4D) for Design #2 Light and Heavy Load

Inputs: CLK, CK, RCO~, ML~, RAS~3, RFRQ~, RFIP~, TEN~, WR~;

Outputs:

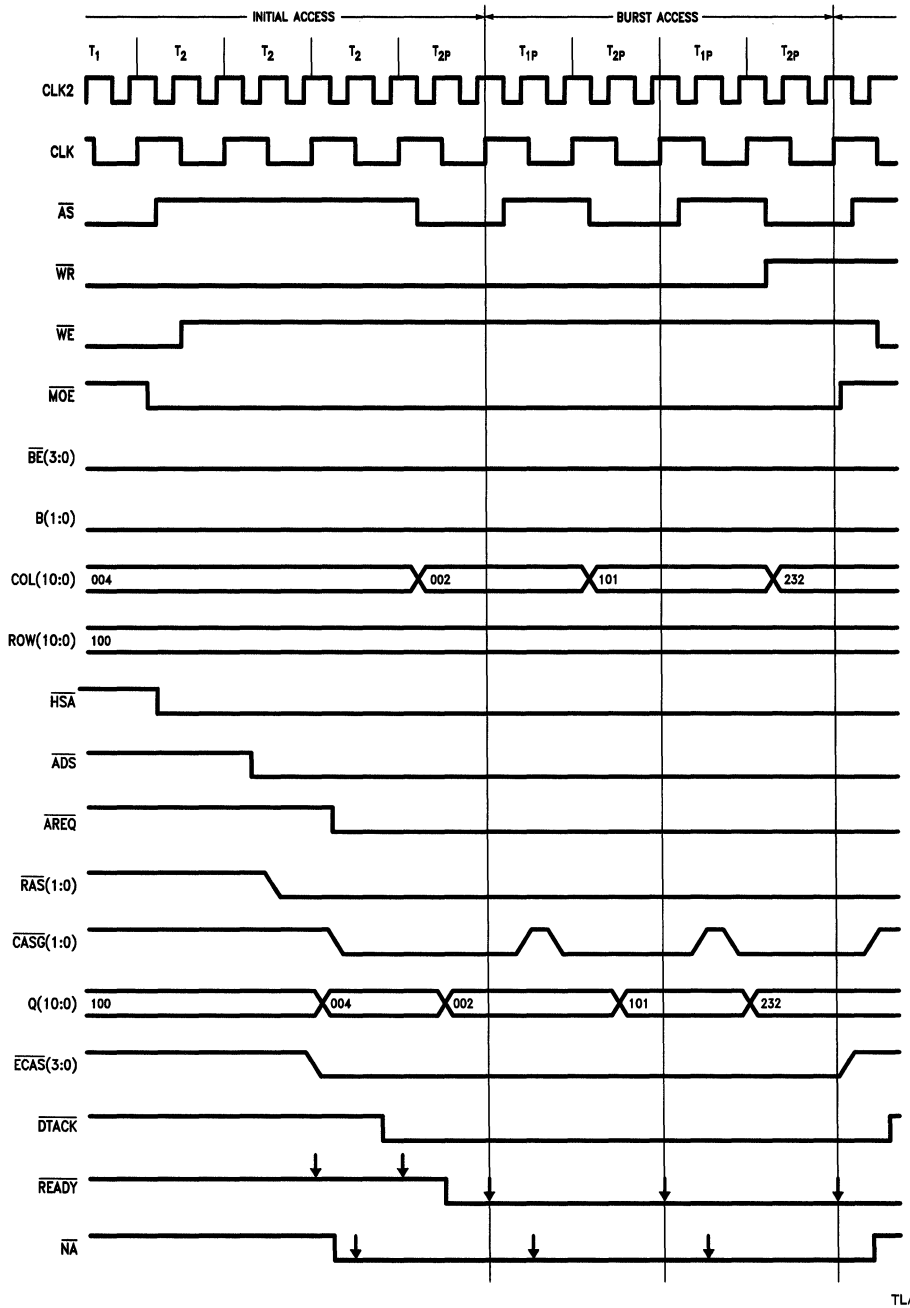
```
/LATCH~   = /RCO + RFSH~;
/RFSHCK~  = /CK + (/LATCH~ * /RFRQ~) +
            (/LATCH * /RAS~3);
/RFSH~    = /TEN~ + LATCH~;
/RFIPD1~  := /RFIP~;
/RFIPD2~  := /RFIPD1~;
/RFIPD3~  := /RFIPD2~;
/MOE~     := /WR~;
```

IIe. 386PAL4 (PAL16R4D) for Design #2 Light and Heavy Load

Inputs: CLK2, DTACK~, CSD~, HSA~, RFSH~, CK, CNT0, CNT1, CNT2, MOE~;

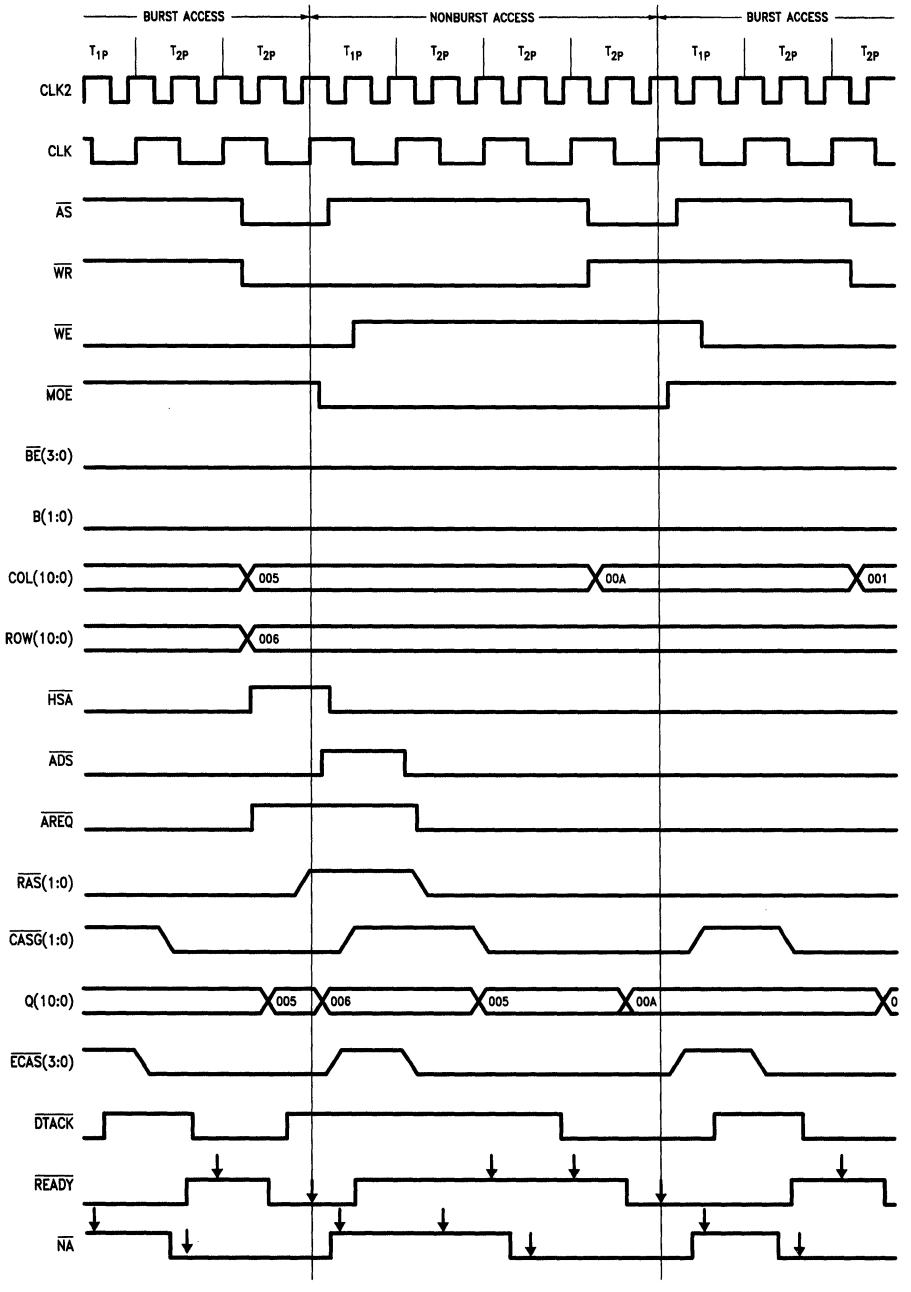
Outputs:

```
/TEN~     = /CNT0 * CNT1 * /CNT2;
/CKD      = /CK;
/READY~   := (/DTACK~ * CK) + (/READY~ * /CK);
/ADSHI~   := (/CSD~ * /HSA~ * RFSH~) +
             /RFSH~;
/WE~      := /MOE~;
```



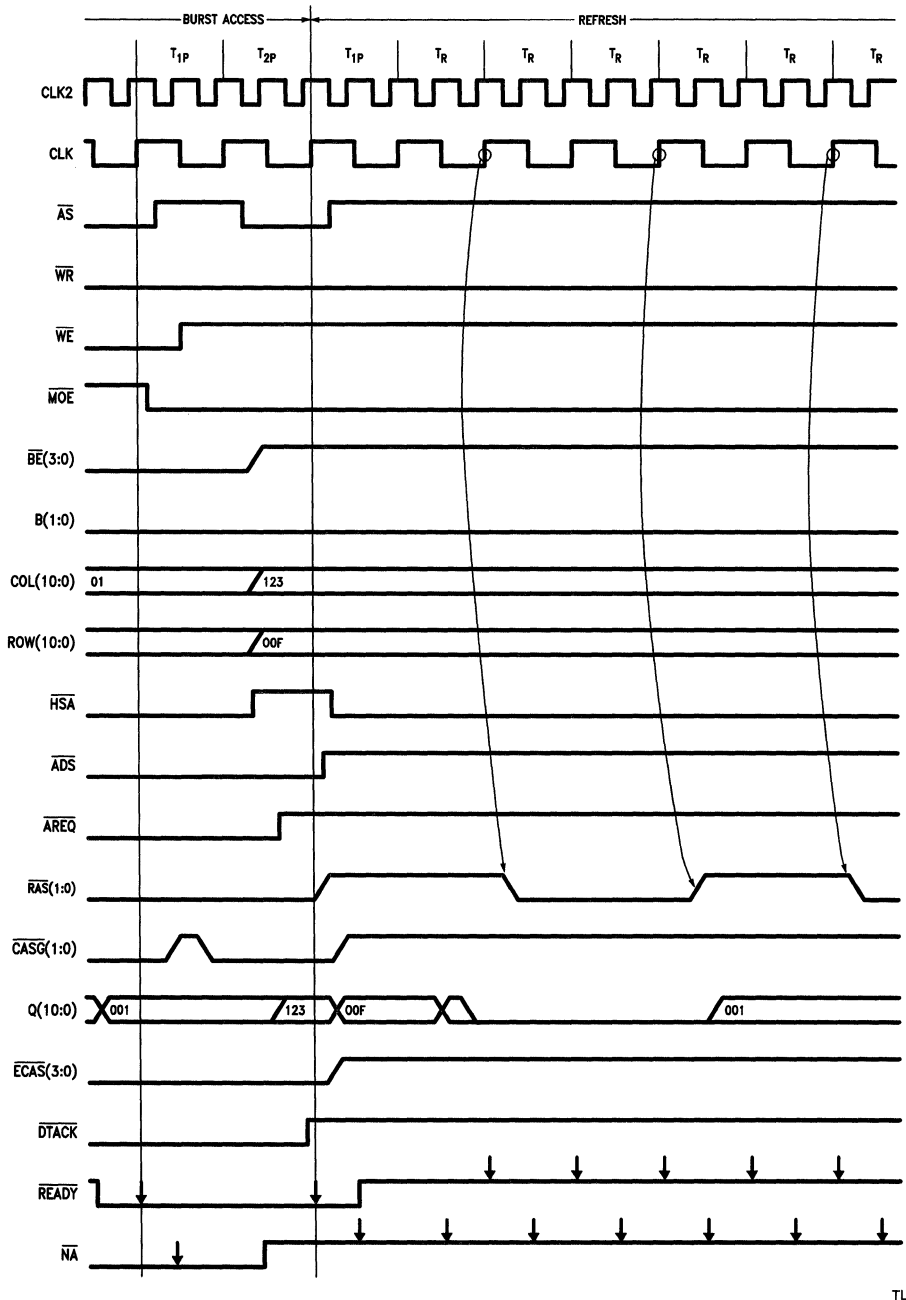
TL/F/10443–2

FIGURE 6. Timing Diagram of Initial and Burst Access Cycles for Design #2



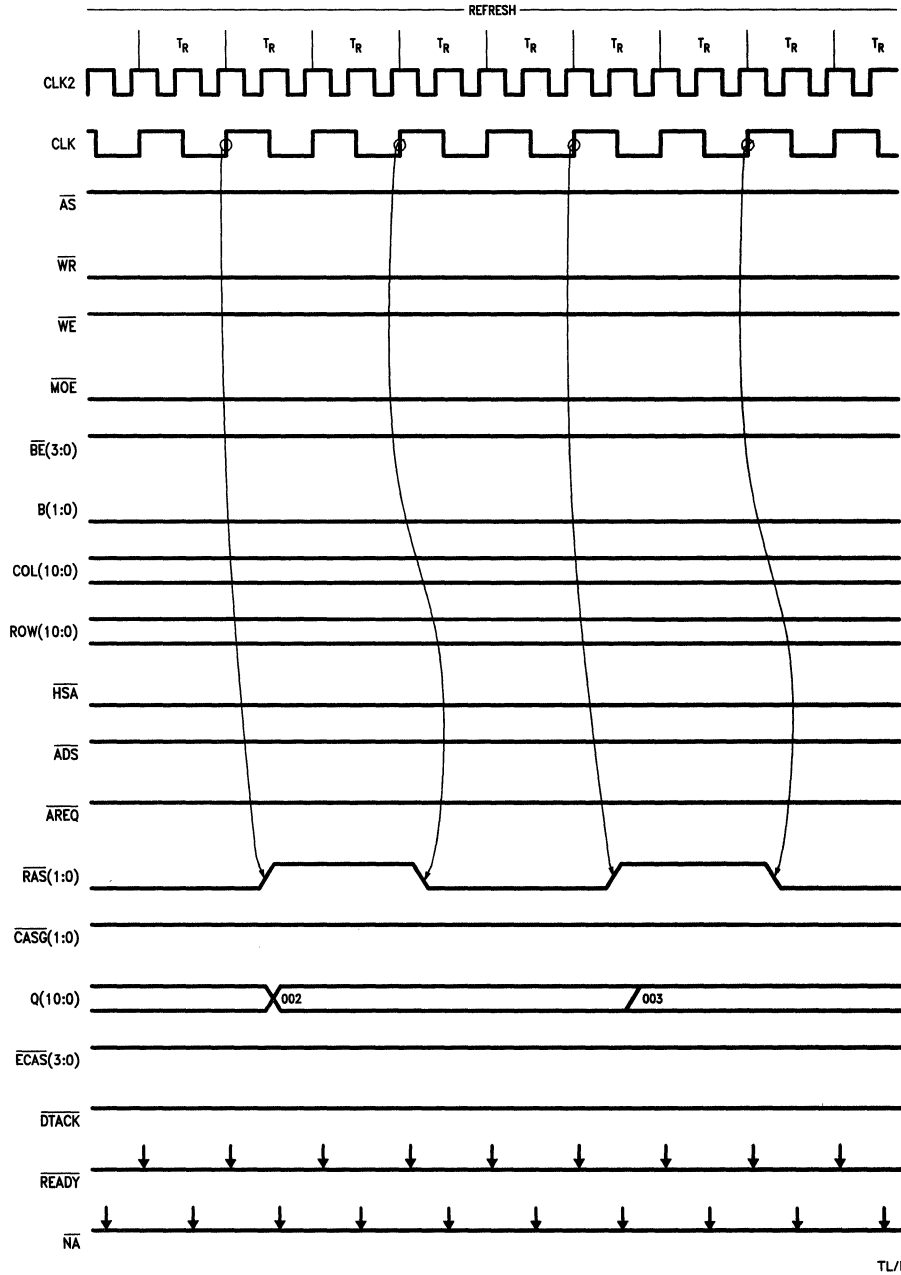
TL/F/10443–3

FIGURE 7. Timing Diagram of Non-Burst and Burst Access Cycles for Design #2



TL/F/10443–4

FIGURE 8. Timing Diagram of Refresh Cycles following Burst Access Cycles for Design #2



TL/F/10443–5

FIGURE 9. Timing Diagram of Refresh Cycles for Design #2

Interfacing the DP8420A/21A/22A to the 29000 Utilizing the Burst Access Mode

National Semiconductor
Application Note 602
Webster (Rusty) Meier Jr.

INTRODUCTION

This application note describes how to interface the 29000 microprocessor to the DP8422A DRAM controller (also applicable to DP8420A/21A). The DP8422A supports the 29000 in the burst access mode. It is assumed that the reader is already familiar with 29000 access cycles and the DP8422A modes of operation.

IA. Description of Designs #1 and #2, the 29000 in burst access mode on the instruction bus, allowing operation up to 25 MHz with four wait states in normal accesses and one or two wait states in burst accesses

The two designs of this application note consist of the DP8422A DRAM controller, a single PAL® (PAL16L8D), several flip-flops, and several logic gates. These parts interface to the 29000 as shown in the block diagrams. This design accommodates two banks of DRAM, each bank being 32 bits in width, giving a maximum memory capacity of 32 Mbytes (using 4M-bit X 1 DRAMs).

This memory design supports burst accesses by the 29000 to the instruction bus but could also support burst accessing to the data bus given a few minor changes. Design #1 inserts 4 wait states in normal accesses and 2 wait states in burst accesses. Design #2 inserts 4 wait states in normal accesses and 1 wait state in burst accesses. When idle states occur during a burst access the next occurring access will only have one wait state.

This application allows page mode DRAMs to be used by holding $\overline{RAS}$ low after an access is completed ($\overline{IRDY}$ low) until either a refresh request or a new access request ($\overline{IREQ}$) is generated. If $\overline{IBREQ}$ (Instruction Burst Request) has transitioned low the next sequential word is accessed from the DRAM via a page mode access. This access involves incrementing the column address (DP8422A COLINC input high) and toggling the $\overline{CAS}$ outputs via the PAL $\overline{ECAS}$ output. The instruction burst acknowledge ($\overline{IBACK}$) input is driven low except in the case of a refresh request ($\overline{RFRQ}$) or a new access ($\overline{IREQ}$) being requested.

If the user wants to do dual accessing with the DP8422A DRAM controller, address buffers (74AS244s) must be added to the address, $\overline{ECAS}$0–3, $\overline{LOCK}$, and $\overline{WIN}$ inputs.

Note that the DP8422A DRAM controller could be used to allow instruction memory accessing thru Port A. The instruction memory could be loaded thru a data memory interface to Port B.

The logic shown in these applications form a complete 29000 memory sub-system, no other logic is needed. This sub-system automatically takes care of:

A. arbitration between Port A, and refreshing the DRAM:

B. the insertion of wait states to the processor (Port A) when needed;

C. remaining in burst access mode unless a refresh request ($\overline{RFRQ}$) or new instruction request is generated;

D. incrementing the column address during burst mode accesses; and

E. guaranteeing the $\overline{RAS}$ and $\overline{CAS}$ low and precharge times.

The timing calculations for this design are included in this application note.

When using the DP8420A/21A/22A at or above 20 MHz the user should program three clock periods of precharge. This is because two clock periods of precharge at 20 MHz will only guarantee 77 ns of $\overline{RAS}$ precharge (2 × 50 ns − tD1, parameter #50 "14 ns" − clock (20 MHz) to $\overline{AREQ}$ high "9 ns").

IB. High Performance NO WAIT State 29000 System

A higher performance 29000 system (similar to the ones discussed in this application note) could be designed using the DP8422A by accessing four $\overline{RAS}$ banks at once, this could allow zero wait state burst accesses. An example of this method using 2 DRAM banks is shown at the end of this application note. If 2 DRAM controllers were used (4 BANKs) burst accesses could execute in zero wait states. The COLINC inputs to the two DRAM controllers would be staggered in time as well as the four $\overline{ECAS}$ outputs (B0$\overline{ECAS}$–B3$\overline{ECAS}$) from the 29000 control logic block.

II. 29000 Programming mode bits

Programming Bits	Description
R0 = 1 R1 = 1	$\overline{RAS}$ low four clocks, $\overline{RAS}$ precharge of three clocks
R2 = 1 R3 = 0	$\overline{DTACK}$ will be asserted on the positive edge of CLK following the access $\overline{RAS}$
R4 = 0 R5 = 0 R6 = X R7 = 1	No WAIT states during burst accesses $\overline{DTACK}$ output
R8 = 1 R9 = X	Non-address pipelining to support burst accessing
C0 = X C1 = X C2 = X	Select based upon the input clock frequency. Example: if the DELCLK frequency is 20 MHz then choose C0,1,2 = 0,0,0 (divide by ten, this will give a frequency of 2 MHz).
C3 = X	
C4 = 0 C5 = 0 C6 = 1	$\overline{RAS}$ and $\overline{CAS}$ groups selected by "B1". This mode allows two $\overline{RAS}$ and four $\overline{CAS}$ outputs to go low during an access.
C7 = 1 C8 = 1	Column address setup time of 0 ns. Row address hold time of 15 ns
C9 = 1	Delay $\overline{CAS}$ during write accesses to one clock after $\overline{RAS}$ transitions low.
B0 = 0 B1 = 1	Latching of the address inputs, needed for the burst accessing capability of the 29000 (COLINC input in particular). Access mode 1

0 = Program with low voltage level
1 = Program with high voltage level
X = Program with either low voltage level (don't care condition)

III. 29000 25 MHz timing calculations for Design #1 and Design #2, with four wait states per normal access and one or two wait states per burst access

1. Maximum time to address valid (with respect to 25 MHz clock):

   14 ns (29000 data sheet #6)

2. Maximim time to $\overline{ADS}$ low (with respect to 25 MHz clock):

   9 ns (74AS374 clock to Q, tPHL)

3. Minimum $\overline{ADS}$ low setup time to CLK high (DP8422-25 needs 25 ns, #400b):

   40 ns (one clock period, 25 MHz) − 9 ns (#2) = 31 ns

4. Minimum address setup time to $\overline{ADS}$ low (DP8422-25 needs 14 ns, #404):

   40 ns (One clock period, 25 MHz) + 2 ns (minimum 74AS374 clock to Q) − 14 ns (#1) = 28 ns

5. Minimum $\overline{CS}$ setup time to CLKA high (DP8422-25 needs 5 ns, #401):

   24 ns (#4) − 9 ns (74AS138 decoder = 15 ns

6. Determining tRAC ($\overline{RAS}$ access time needed by the DRAM):

   200 ns (five clock periods at 25 MHz) − 40 ns (one clock period, T1) − 9 ns (#2) − 6 ns (29000 data setup time, #9a) − 7 ns (74F245) − 29 ns (CLK to $\overline{RAS}$ low) = 109 ns

   Therefore the tRAC of the DRAM must be 109 ns or less.

7. Determining tCAC ($\overline{CAS}$ access time needed by the DRAM) and tAA (column address access time):

   200 ns − 40 ns − 9 ns − 6 ns − 7 ns − 82 ns (CLK to $\overline{CAS}$ low) = 56 ns

   Therefore the tCAC and tAA (access time from the column address) of the DRAM must be 56 ns or less. COMMON 100 ns DRAMS WILL MEET THIS tRAC, tAA, AND tCAC PARAMETER.

8. Minimum setup of $\overline{DTACK}$1 to the 74AS374 ONLY, (need 2 ns):

   40 ns (one clock period) − 28 ns (DP8422A-25 $\overline{DTACK}$1 delay, #18) = 12 ns

9. Minimum $\overline{IRDY}$ setup time to $\overline{IRDY}$ being sampled (12 ns is needed by the 29000, #9):

   40 ns (one clock period) − 9 ns (maximum 74AS374 clock to Q output delay) = 31 ns

10a. Determining tCAC during a burst access for Design #1:

120 ns (3 clock periods) − 20 ns (one half clock period during which $\overline{CAS}$ is high) − 10 ns (PAL16L8 maximum propagation delay) − 12 ns (74AS32 propagation delay driving ≈ 125 pF with damping resistor and other associated delays) − 6 ns (data setup time) − 7 ns (74F245) = 65 ns

Therefore the tCAC of the DRAM must be ≤ 65 ns

10b. Determining tCAC during a burst access for Design #2:

80 ns (2 clock periods) − 20 ns (one half clock period during which $\overline{CAS}$ is high) − 10 ns (PAL16L8 maximum propagation delay) − 12 ns (74AS32 propagation delay driving ≈ 125 pF with damping resistor and other associated delays) − 6 ns (data setup time) − 7 ns (74F245) = 25 ns

Therefore the tCAC of the DRAM must be ≤ 25 ns

11a. Determining tAA during a burst access for Design #1: 160 ns (4 clock periods, because $\overline{IRDY}$ inverted is COLINC) − 39 ns (DP8422A COLINC to Q's valid, #27) = 121 ns ≥ tAA.

11b. Determining tAA during a burst access for Design #2: 120 ns (3 clock periods, because $\overline{IRDY}$ inverted is COLINC) − 39 ns (DP8422A COLINC to Q's valid, #27) = 81 ns ≥ tAA.

IV. 29000 PAL equations, Design #1, 2 wait states during burst accesses. Written in National Semiconductor PLAN™ format

PAL16L8D

CLK $\overline{CS}$ $\overline{IREQ}$ $\overline{IBREQ}$ $\overline{DTACK2}$ $\overline{AREQ}$ $\overline{IRDY}$ $\overline{IRDY3}$ $\overline{RFRQD}$
GND $\overline{RESET}$ $\overline{IBACK}$ $\overline{ECAS}$ $\overline{ECAS1}$ $\overline{DOACC}$ $\overline{ENDACC}$ $\overline{IREQH}$
$\overline{IRDY1}$ $\overline{AREQ1}$ VCC

If (VCC) $\overline{IBACK}$ = $\overline{AREQ}$*RFRQD*IREQH*$\overline{DTACK2}$
+$\overline{AREQ}$*RFRQD*$\overline{IREQH}$*IREQ
+$\overline{IBACK}$*IREQH*RFRQD

If (VCC) $\overline{ECAS}$ = $\overline{AREQ}$*ECAS1*$\overline{DOACC}$*RESET
+$\overline{AREQ}$*$\overline{DOACC}$*$\overline{IRDY3}$*$\overline{CLK}$*RESET
+$\overline{AREQ}$*$\overline{DOACC}$*$\overline{IRDY1}$*RESET
+$\overline{ECAS}$*ECAS1*CLK*RESET
+$\overline{ECAS}$*$\overline{CLK}$*RESET

If (VCC) $\overline{ECAS1}$ = $\overline{IRDY}$*$\overline{CLK}$
+$\overline{ECAS1}$*CLK

If (VCC) $\overline{DOACC}$ = $\overline{AREQ}$*RFRQD*$\overline{IBREQ}$*RESET
+$\overline{CS}$*$\overline{IREQ}$*IREQH*$\overline{AREQ}$*RESET
+$\overline{DOACC}$*IRDY*RESET

If (VCC) $\overline{ENDACC}$ = $\overline{IREQ}$*$\overline{IREQH}$
+$\overline{RFRQD}$*$\overline{IRDY}$
+$\overline{RFRQD}$*$\overline{AREQ}$*$\overline{IREQH}$
+$\overline{ENDACC}$*$\overline{AREQ}$
+$\overline{RESET}$

If (VCC) $\overline{IREQH}$ = $\overline{AREQ}$*IREQ
+$\overline{IREQH}$*$\overline{AREQ}$*RESET

If (VCC) $\overline{IRDY1}$ = $\overline{DTACK2}$*$\overline{AREQ}$*IRDY*IRDY3*IREQH
+$\overline{AREQ}$*IRDY*IRDY3*$\overline{IREQH}$*CLK*$\overline{DOACC}$
+$\overline{IRDY1}$*IRDY*IRDY3*$\overline{AREQ}$

If (VCC) $\overline{AREQ1}$ = $\overline{CS}$*$\overline{IREQ}$*IRDY*ENDACC*RESET
+$\overline{AREQ1}$*ENDACC*RESET
+$\overline{AREQ1}$*$\overline{DOACC}$*RESET

29000 PAL equations, Design #2, 1 wait state during burst accesses. Written in National Semiconductor PLAN format

PAL16L8D

CLK $\overline{CS}$ $\overline{IREQ}$ $\overline{IBREQ}$ $\overline{DTACK2}$ $\overline{AREQ}$ $\overline{IRDY}$ $\overline{IRDY3}$ $\overline{RFRQD}$
GND $\overline{RESET}$ $\overline{IBACK}$ $\overline{ECAS}$ $\overline{ECAS1}$ $\overline{DOACC}$ $\overline{ENDACC}$ $\overline{IREQH}$
$\overline{IRDY1}$ $\overline{AREQ1}$ VCC

If (VCC) $\overline{IBACK}$ = $\overline{AREQ}$*RFRQD*IREQH*$\overline{DTACK2}$
+$\overline{AREQ}$*RFRQD*$\overline{IREQH}$*IREQ
+$\overline{IBACK}$*IREQH*RFRQD

If (VCC) $\overline{ECAS}$ = $\overline{AREQ}$*ECAS1*$\overline{DOACC}$*RESET
+$\overline{AREQ}$*$\overline{DOACC}$*$\overline{IRDY3}$*$\overline{CLK}$*RESET
+$\overline{AREQ}$*$\overline{DOACC}$*$\overline{CLK}$*RESET
+$\overline{ECAS}$*ECAS1*CLK*RESET
+$\overline{ECAS}$*$\overline{CLK}$*RESET

If (VCC) $\overline{ECAS1}$ = $\overline{IRDY}$*$\overline{CLK}$
+$\overline{ECAS1}$*CLK

If (VCC) $\overline{DOACC}$ = $\overline{AREQ}$*RFRQD*$\overline{IBREQ}$*RESET
+$\overline{CS}$*$\overline{IREQ}$*IREQH*$\overline{AREQ}$*RESET
+$\overline{DOACC}$*IRDY*RESET

If (VCC) $\overline{ENDACC}$ = $\overline{IREQ}$*$\overline{IREQH}$
+$\overline{RFRQD}$*$\overline{IRDY}$
+$\overline{RFRQD}$*$\overline{AREQ}$*$\overline{IREQH}$
+$\overline{ENDACC}$*$\overline{AREQ}$
+$\overline{RESET}$

If (VCC) $\overline{IREQH}$ = $\overline{AREQ}$*IREQ
+$\overline{IREQH}$*$\overline{AREQ}$*RESET

If (VCC) $\overline{IRDY1}$ = $\overline{DTACK2}$*$\overline{AREQ}$*IRDY*$\overline{IREQH}$*CLK*$\overline{DOACC}$
+$\overline{AREQ}$*IRDY*$\overline{IREQH}$*CLK*$\overline{DOACC}$
+$\overline{IRDY1}$*IRDY*$\overline{AREQ}$

If (VCC) $\overline{AREQ1}$ = $\overline{CS}$*$\overline{IREQ}$*IRDY*ENDACC*RESET
+$\overline{AREQ1}$*ENDACC*RESET
+$\overline{AREQ1}$*$\overline{DOACC}$*RESET

Key: Reading PAL equations written in PLAN

EXAMPLE EQUATIONS: $\overline{READ}$: = CS_RD*ADS1D*$\overline{CLKA}$
+$\overline{READ}$*$\overline{ADS1D}$
+$\overline{READ}$*CLKA

This example reads: the output "$\overline{READ}$" will transition low on the next rising "CLK" clock edge (given that one of the following conditions are valid a setup time before "CLK" transitions high);

1. the input "CS_RD" is high AND the input "$\overline{ADS1D}$" is high AND the input "CLKA" is low, OR
2. the output "$\overline{READ}$" is low AND the input "$\overline{ADS1D}$" is low, OR
3. the output "$\overline{READ}$" is low AND the input "CLKA" is high

V. 29000 application note PAL and 74AS374 outputs

What follows is a brief explanation of the PAL and 74AS374 outputs:

$\overline{AREQ}1$ This combinational output of the PAL is sampled at the next rising clock edge (74AS374) to provide the $\overline{ADS}$ and $\overline{AREQ}$ outputs that drive the DP8422A DRAM accesses. This output is held low to allow burst accessing until either a new access is requested by the 29000 ($\overline{IREQ}$) or a refresh is requested ($\overline{RFRQ}$).

$\overline{IRDY}1$ This combinational output of the PAL is sampled at the next rising clock edge (74AS374) to provide the $\overline{IRDY}$ output that terminates each 29000 access.

$\overline{IREQH}$ This combinational output of the PAL is used internal to the PAL as an indication of $\overline{IREQ}$ having transitioned high. It is useful in determining when the 29000 is terminating a burst access to request an access to another page ($\overline{IREQH}$ = $\overline{IREQ}$ = low).

$\overline{ENDACC}$ This combinational output of the PAL is used internal to the PAL as an indication of when to terminate a burst, or single, access. It indicates a new page access ($\overline{IREQ}$), a refresh request ($\overline{RFRQ}$), or a hardware reset ($\overline{RESET}$) operation. Accesses are only terminated after $\overline{IRDY}$ is issued or during idle states when no accesses are pending.

$\overline{DOACC}$ This combinational output of the PAL is used internal to the PAL to keep track of requested accesses, single ($\overline{IREQ}$) or burst ($\overline{IBREQ}$ and $\overline{IBACK}$).

$\overline{ECAS}1$ This combinational output of the PAL is used internal to the PAL to allow the $\overline{ECAS}$ output to have minimum delay following the CLK high and low.

$\overline{ECAS}$ This combinational output of the PAL is used to toggle the $\overline{CAS}$ outputs of the DP8422A during burst accessing.

$\overline{IBACK}$ This combinational output of the PAL is used as an input to the 29000 to indicate when the DP8422A is available to support burst accessing. This output is pulled high during refresh requests ($\overline{RFRQ}$) and out of page accesses ($\overline{IREQH}$ = $\overline{IREQ}$ = low).

$\overline{AREQ}$ See $\overline{AREQ}1$ explanation.

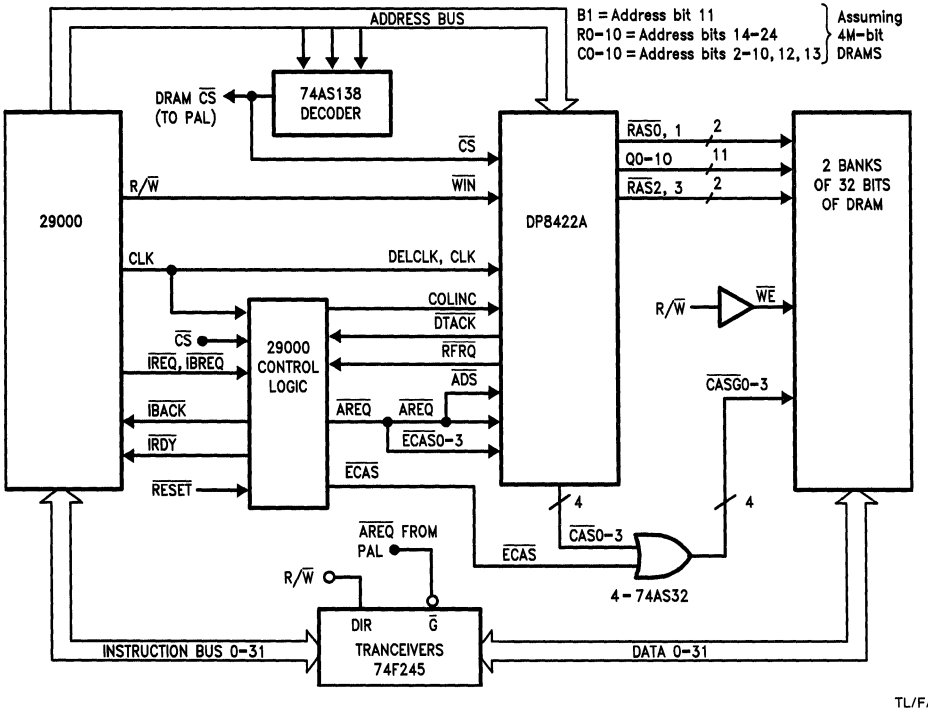
$\overline{IRDY}$ See $\overline{IRDY}1$ explanation.

$\overline{IRDY}3$ This output (74AS374) is used as a state input to the PAL. This term is $\overline{IRDY}$ delayed by one clock period.

$\overline{RFRQD}$ This clocked output is used to synchronize the DP8422A $\overline{RFRQ}$ output.

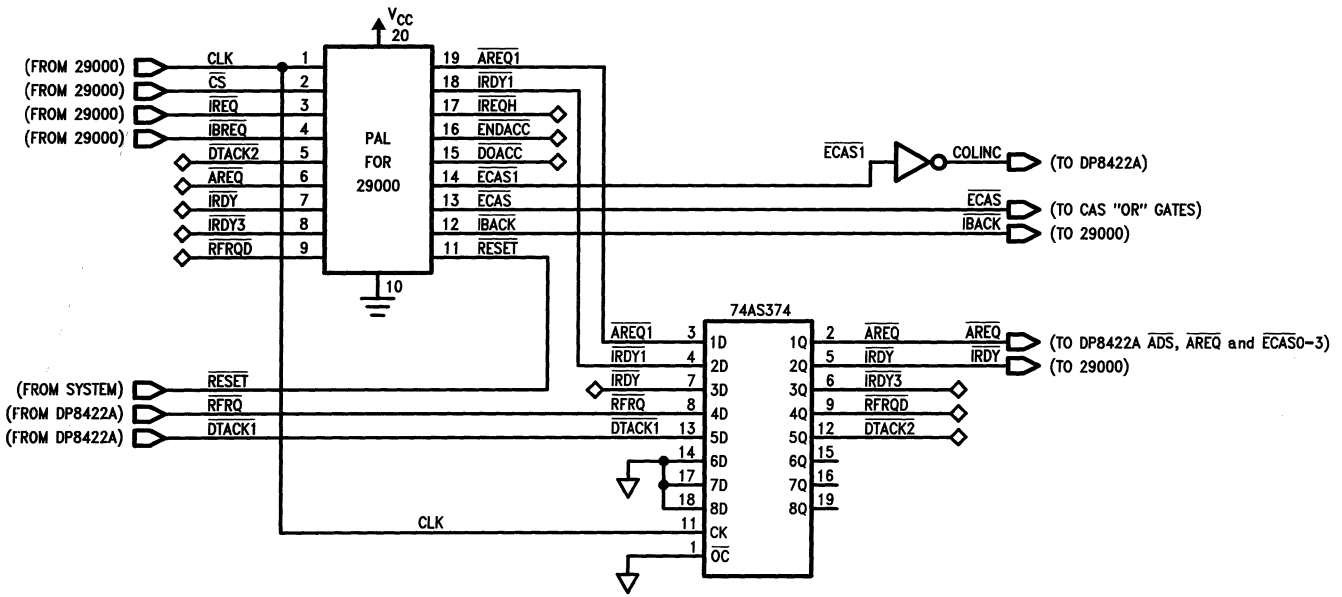
$\overline{DTACK}2$ This clocked output is generated from the DP8422A $\overline{DTACK}1$ output and is synchronized to the next rising clock edge.

Block Diagram of 29000/DP8422A Design at 25 MHz (Instruction memory interface to Port A, Port B could be used as a data memory interface to load the instructions into the DRAM)



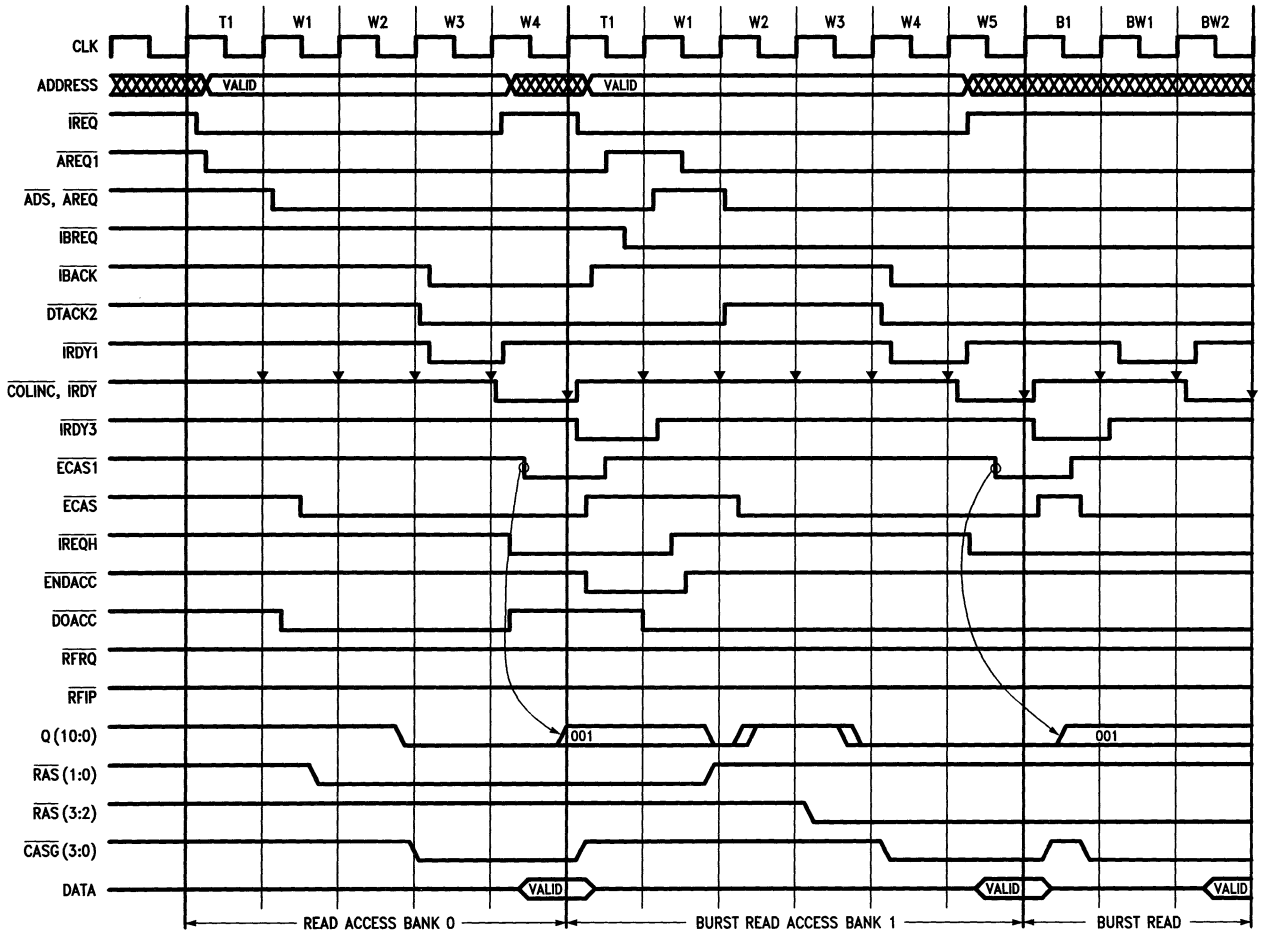
TL/F/10394-1

29000 Control Logic Block Diagram



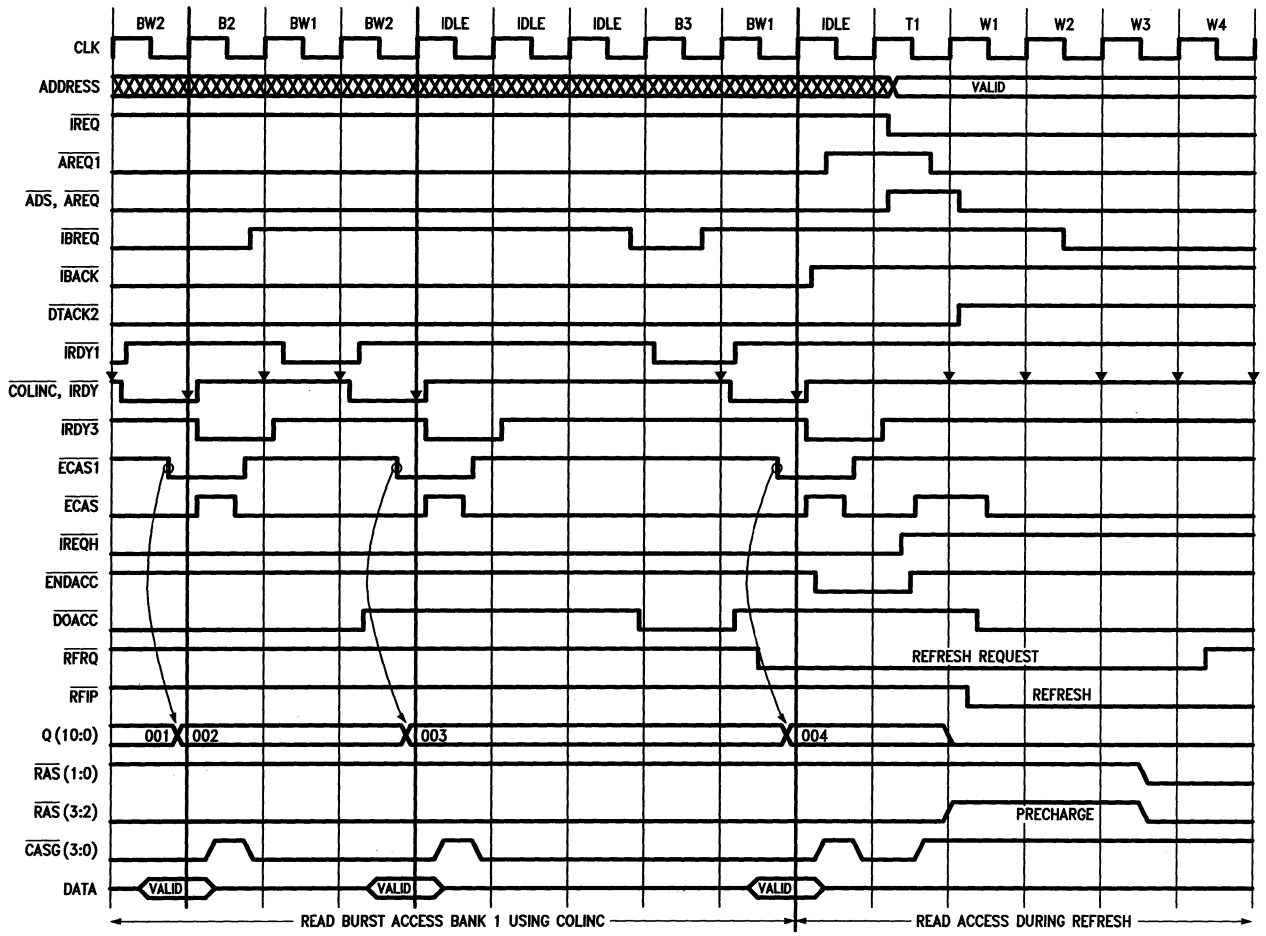
TL/F/10394–2

Design # 1 Timing



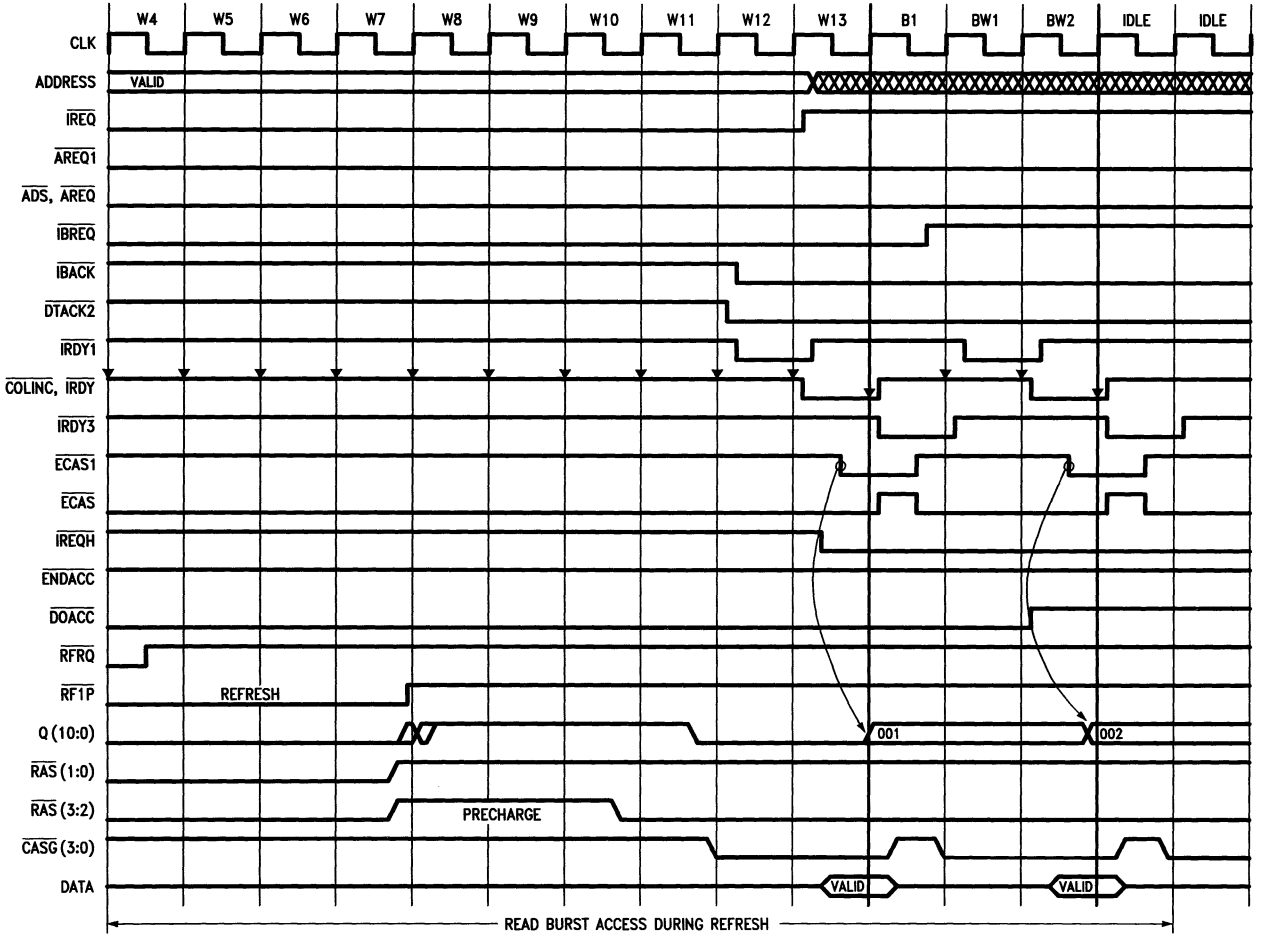
TL/F/10394–3

Design #1 Timing (Continued)



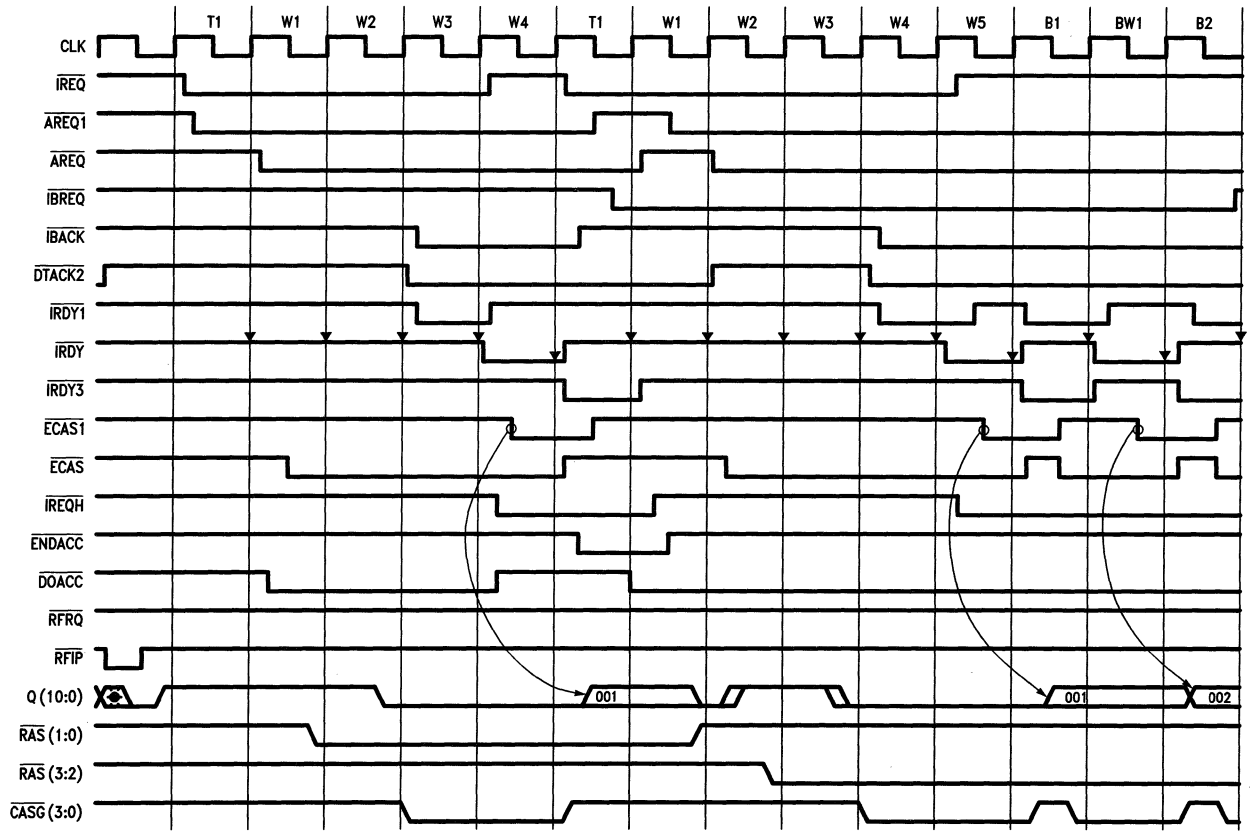
TL/F/10394–4

Design #1 Timing (Continued)



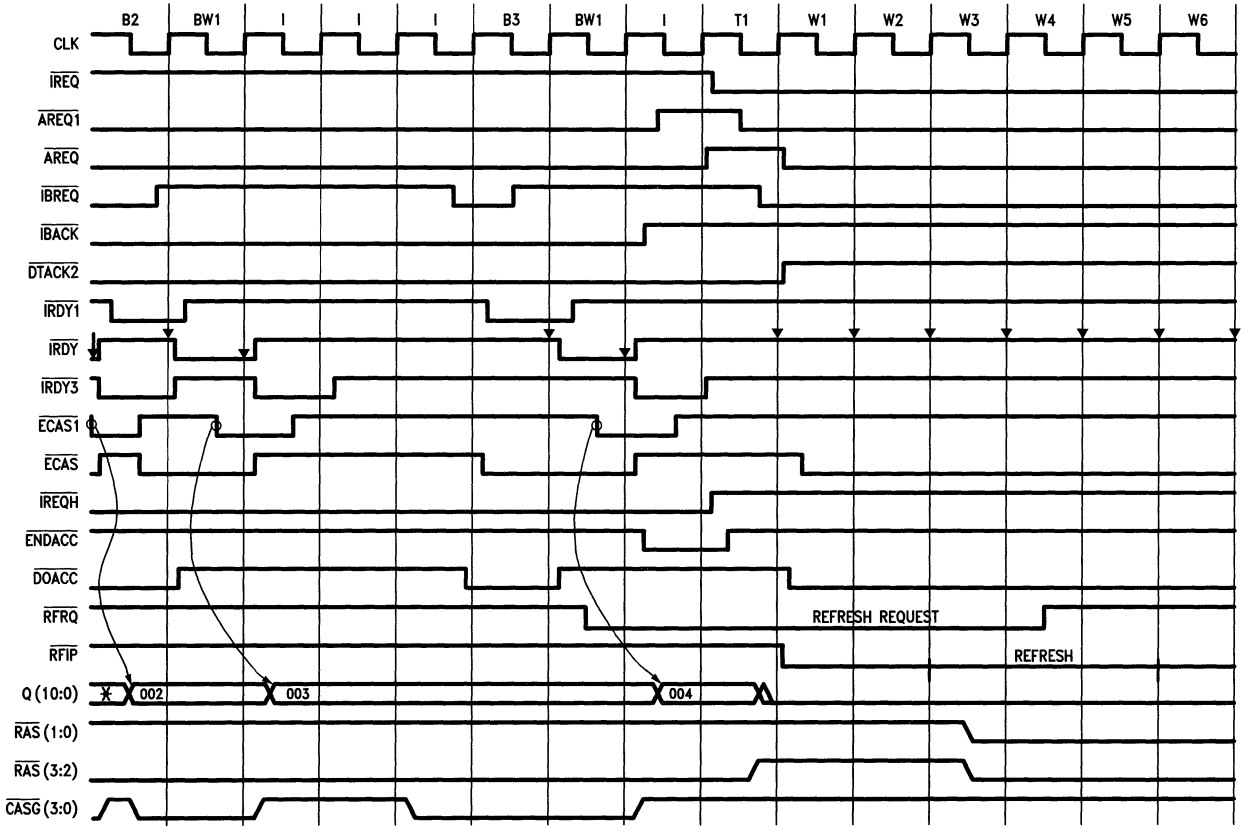
TL/F/10394–5

Design #2 Timing



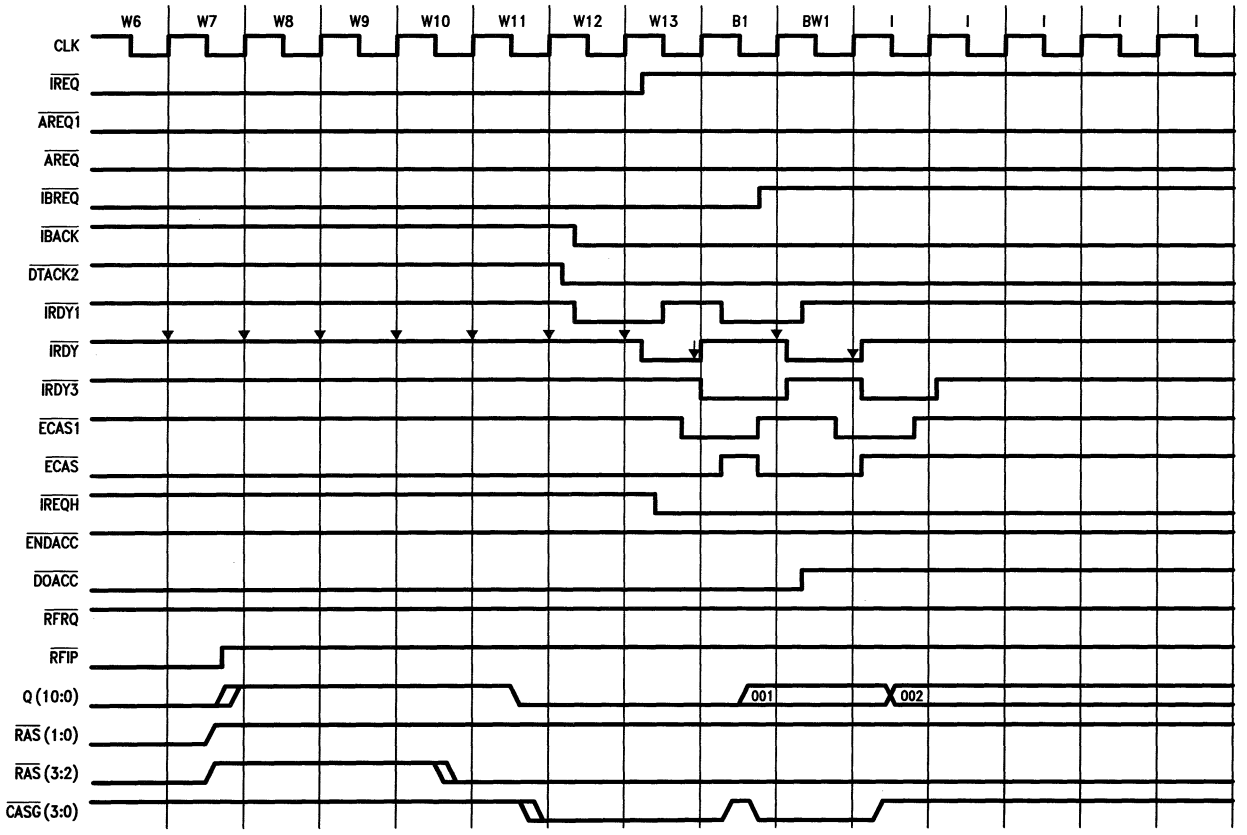
TL/F/10394–9

Design #2 Timing (Continued)



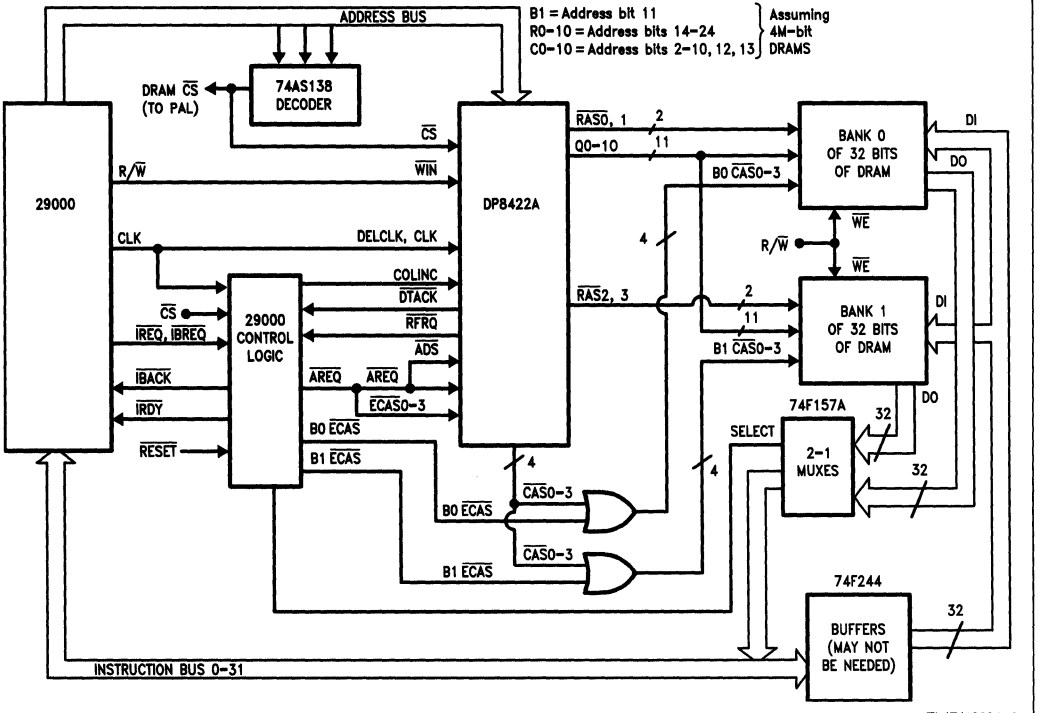
TL/F/10394–10

Design #2 Timing (Continued)



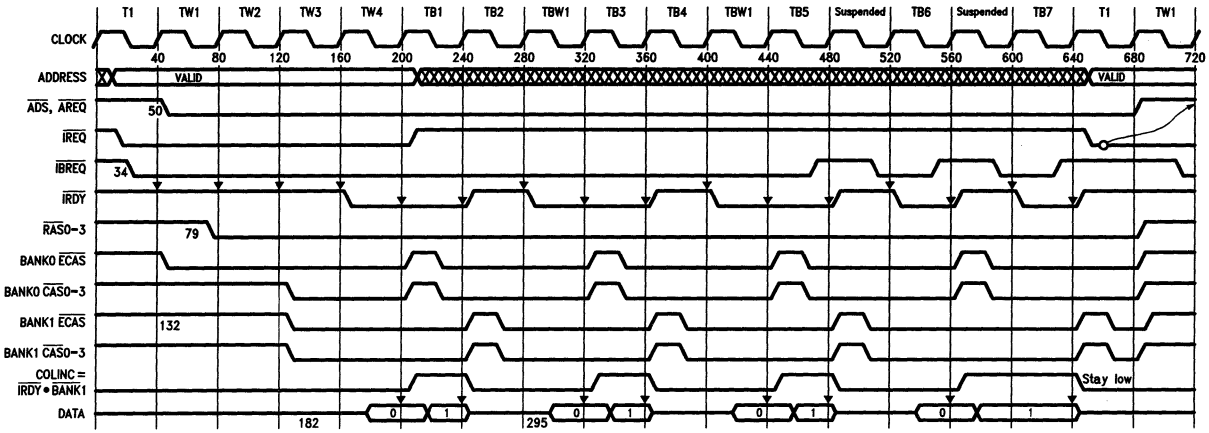
TL/F/10394–11

High Performance 2900 System Block Diagram Using 2 DRAM Banks, This System Could Be of Still Higher Performance by Extending it to 4 DRAM Banks



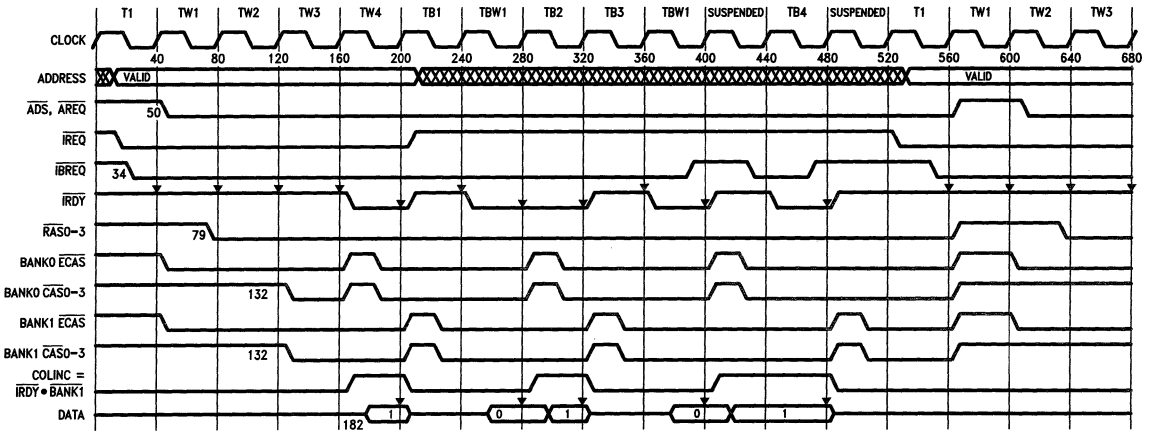
TL/F/10394–8

29000 Mode 1,2 Bank System High Performance (Start Access in Bank 0)



TL/F/10394-6

2900 Mode 1 2 Bank System High Performance (Start Access in Bank 1) (Interleaving Ability Not as Good)



TL/F/10394-7

Interfacing the DP8420A/21A/22A to the Z280/Z80000/Z8000 Microprocessor

National Semiconductor
Application Note 546
Webster (Rusty) Meier, Jr. and Joe Tate

I INTRODUCTION

This application note describes how to interface the Z280 microprocessor to the DP8422A DRAM controller (also applicable to DP8420A/21A). It is assumed that the reader is already familiar with Z280 and the DP8422A modes of operation. The interface to the Z80000 and Z8000 is similar to the interface described in this application note.

II DESCRIPTION OF DESIGN, ALLOWING OPERATION AT 10 MHz (AND ABOVE) WITH 1 WAIT STATE IN NORMAL ACCESSES AND 1 WAIT STATE DURING BURST ACCESSES

The block diagram of this design is shown driving two banks of DRAM, each bank being 16 bits in width, giving a maximum memory capacity of up to 16 Mbytes (using 4 M-bit × 1 DRAMs). By choosing a different $\overline{RAS}$ and $\overline{CAS}$ configuration mode (see programming mode bits section of DP8422A data sheet) this application can support 4 banks of DRAM, giving a memory capacity of up to 32 Mbytes (using 4 M-bit × 1 DRAMs).

The memory banks are interleaved on every four word (16-bit word) boundry. This means that the address bit (A3) is tied to the bank select input of the DP8422A (B1).

Address bits A2,1 are tied to the most significant row and column address inputs (R9,C9 for 1 Mbit DRAMs) to support burst accesses using static column mode DRAMs. Since this application assumes the use of static column DRAMs the column address strobe ($\overline{CAS}$) is left low throughout the entire burst access. If the user desires to use nibble mode or page mode DRAMs the $\overline{CAS}$ outputs must be toggled, the $\overline{ECAS}$ inputs the DP8422A can be used for this purpose ($\overline{DS}$ of the Z280 could be "OR"ed with the current $\overline{ECAS}$ inputs). If nibble mode DRAMs are used the COLINC input of the DP8422A need not be driven.

Address bit A0 is used to produce the two byte select data strobes along with the byte/word signal (B/W). These byte selects (Byte 0 $\overline{ECAS}$ and Byte 1 $\overline{ECAS}$) are used in byte reads and writes as well as selects for the transceivers.

If the majority of accesses made by the Z280 are sequential, the Z280 can be doing burst accesses most of the time. Each burst of four words can alternate memory banks, allowing one memory bank to be precharging ($\overline{RAS}$ precharge) while the other bank is being accessed (Bank select, B1, tied to address A3). This is a higher performance memory system then a non-interleaved memory system (bank select on the higher address bits). Each separate memory access to the same memory bank may require extra wait states to be inserted into the CPU access cycles to allow for the $\overline{RAS}$ precharge time, if two periods or more of $\overline{RAS}$ precharge were programmed.

This application allows 1 or more wait states to be inserted in normal accesses and 1 or more wait states to be inserted during burst accesses of the Z280. The number of wait states can be adjusted through the $\overline{WAITIN}$ input of the DP8422A.

The logic shown in this application note forms a complete Z280 memory sub-system, no other logic is needed. This sub-system automatically takes care of:

A. arbitration between Port A, Port B, and refreshing the DRAM;

B. the insertion of wait states to the processor (Port A and Port B) when needed (i.e., if $\overline{RAS}$ precharge is needed, refresh is happening during a memory access, the other Port is currently doing an access . . .etc);

C. performing byte writes and reads to the 16-bit words in memory;

D. normal and burst access operations.

The external wait logic (U1, U2, U3, U4; see *Figure 1*) is needed to support burst accesses of the Z280. During burst accesses the Z280 $\overline{WAIT}$ input is sampled every falling clock edge. What is worse is that the $\overline{WAIT}$ input needs one half clock period setup time and the $\overline{DS}$ signal (used to toggle $\overline{ECAS}$0–3 and thereby toggle the DP8422A $\overline{WAIT}$ output) takes close to one half of a clock period to transition high. This leaves no time for the DP8422A $\overline{WAIT}$ output to transition between states. The external flip-flop is used to provide extra fast response time for normal access wait states and to toggle when doing a burst mode access. If the user is not going to do burst accesses the $\overline{WAIT}$ output can be tied directly to the $\overline{WAIT}$ input of the Z280 (U1, U2, U3, U4 would not be needed). Also all this logic could easily be put into a PAL® if desired.

By using the "output control" pins of some external latches (74ALS373's), this application can easily be used in a dual access application. The addresses could be TRI-STATE® through these latches, the write input ($\overline{WIN}$), lock input ($\overline{LOCK}$), and $\overline{ECAS}$0–3 inputs must also be able to be TRI-STATE (a 74AS244 could be used for this purpose). By multiplexing the above inputs (through the use of the above parts and similar parts for Port B) the DP8422A can be used in a dual access application. If this design is used in a dual access application the t_{RAC} and t_{CAC} (required $\overline{RAS}$ and $\overline{CAS}$ access time required by the DRAM) will have to be recalculated since the time to $\overline{RAS}$ and $\overline{CAS}$ is longer for the dual access application (see TIMING section of this application note).

III Z280 DESIGN, 10 MHz WITH 1 WAIT STATE DURING NORMAL ACCESSES AND 1 WAIT STATE DURING BURST ACCESSES, PROGRAMMING MODE BITS

Programming Bits	Description
R0 = 0	$\overline{RAS}$ low two clocks, $\overline{RAS}$ precharge
R1 = 1	of two clocks, this setup will only guarantee 93.5 ns $\overline{RAS}$ precharge (at 10 MHz) from refresh $\overline{RAS}$ high to access $\overline{RAS}$ low. If more $\overline{RAS}$ precharge is desired the user should program three periods of $\overline{RAS}$ precharge.
R2 = 0	$\overline{DTACK}$ one half is chosen. $\overline{DTACK}$
R3 = 1	low first rising CLK edge after access $\overline{RAS}$ is low.
R4 = 0	No WAIT states during burst accesses
R5 = 0	
R6 = 0	If $\overline{WAITIN}$ = 0, add one clock to $\overline{DTACK}$. $\overline{WAITIN}$ may be tied high or low in this application depending upon the number of wait states the user desires to insert into the access.
R7 = 1	Select $\overline{DTACK}$
R8 = 1	Non-interleaved Mode
R9 = X	
C0 = X	Select based upon the input
C1 = X	"DELCLK" frequency. Example: if the
C2 = X	input clock frequency is 10 MHz then choose C0,1,2 = 1,0,1 (divide by five, this will give a frequency of 2 MHz).
C3 = X	
C4 = 0	$\overline{RAS}$ groups selected by "B1". This
C5 = 0	mode allows two $\overline{RAS}$ outputs to go
C6 = 1	low during an access, and allows byte writing in 16- or 32-bit words.
C7 = 1	Column address setup time of 0 ns
C8 = 1	Row address hold time of 15 ns
C9 = 1	Delay $\overline{CAS}$ during write accesses to one clock after $\overline{RAS}$ transitions low
B0 = 0	Latches latch on ALE input low
B1 = 0	Access mode 0
$\overline{ECAS}$0 = 0	$\overline{CAS}$n not extended beyond $\overline{RAS}$n

0 = Program with low voltage level
1 = Program with high voltage level
X = Program with either high or low voltage level (don't care condition)

IV Z280 TIMING CALCULATIONS FOR DESIGN AT 10 MHz WITH 1 WAIT STATE DURING NORMAL ACCESSES AND 1 WAIT STATE DURING BURST ACCESSES

1. Minimum ALE high setup time to CLOCK high if using the on-chip latches and more then one $\overline{RAS}$ bank (DP8422A-20 needs 29 ns, #301b):

   100 ns (one clock period) − 20 ns ($\overline{AS}$ valid maximum delay, #3 of Z280 data sheet) − 11 ns (74ALS04B max delay) = 69 ns

2A. Minimum address setup time to ALE low (DP8422A-20 needs 3 ns, #306):

   25 ns (address setup to $\overline{AS}$ high, #20 Z280 data sheet) + 1 ns (74ALS04B min delay) = 26 ns

2B. Minimum address hold time to ALE low (DP8422A-20 needs 10 ns, #305):

   20 ns (address hold from $\overline{AS}$ high, #22 of Z280 data sheet) + 1 ns (74ALS04B min delay) = 21 ns

2C. Minimum address setup to CLOCK high (DP8422A-20 needs bank address setup to CLOCK of 20 ns, #303):

   100 ns (one clock period) − 20 ns (max clock to address valid, Z280 data sheet #2) = 80 ns

3. Minimum $\overline{CS}$ setup time to clock high (DP8422A-20 needs 14 ns, #300): 80 ns (#2C above) − 22 ns (max 74ALS138 decoder) = 58 ns

4. Determining t_{RAC} during a normal access ($\overline{RAS}$ access time needed by the DRAM):

   250 ns (two and one half clock periods to do the access) − 32 ns (CLK to $\overline{RAS}$ low max, DP8422A-20 #307) − 30 ns (Z280 data setup time, #9) − 10 ns (74ALS245A max delay) = 178 ns

   Therefore the t_{RAC} of the DRAM must be 178 ns or less. (One can see that if zero wait states would have been programmed the t_{RAC} would have been 84 ns (using DP8422A-25, has faster CLK to $\overline{RAS}$ low of 26 ns) 184−100 (one clock)).

5. Determining t_{CAC} during a normal access ($\overline{CAS}$ access time) and column address access time needed by the DRAM:

   250 ns − 89 ns (CLK to $\overline{CAS}$ low on DP8422A-20, #308a) − 30 ns − 10 ns = 121 ns

   Therefore the t_{CAC} of the DRAM must be 121 ns or less.

6. Determining the column address access time needed during a static column mode burst access:

   20 ns (two clocks to do the access, Ex. mid T3 to mid TBW to mid T4) − 35 ns ($\overline{DS}$ high, Z280 parameter #8) − 43 ns (COLINC asserted to address outputs of DP8420A-20 incremented, #27) − 30 ns (Z280 data setup time, #9) − 10 ns (74ALS245A max delay) = 82 ns

   Therefore the column address access time of the DRAM must be 82 ns or less. (One can see that if zero wait states would have been programmed the column address access time would have been less then 0 ns (82 − 100 (one clock))).

7. Maximum time to $\overline{DTACK}$ one half low (74ALS374 D type flip-flop needs 10 ns setup to CLK):

   100 ns (One clock, mid T2 in mid TW) − 33 ns ($\overline{DTACK}$ one half low from CLK high on DP8422A-20, #18) − 12 ns (max delay on 74ALS02 = 55 ns

8. Minimum $\overline{WAIT}$ setup time to CLK low (Z280 $\overline{WAIT}$ input needs 50 ns, #14):

   100 ns (one clock period) − 16 ns (74ALS374 max delay) − 14 ns (74ALS08 max delay) = 70 ns

9. Minimum $\overline{RAS}$ precharge (DP8422A programmed with 2 clock periods of $\overline{RAS}$ precharge):

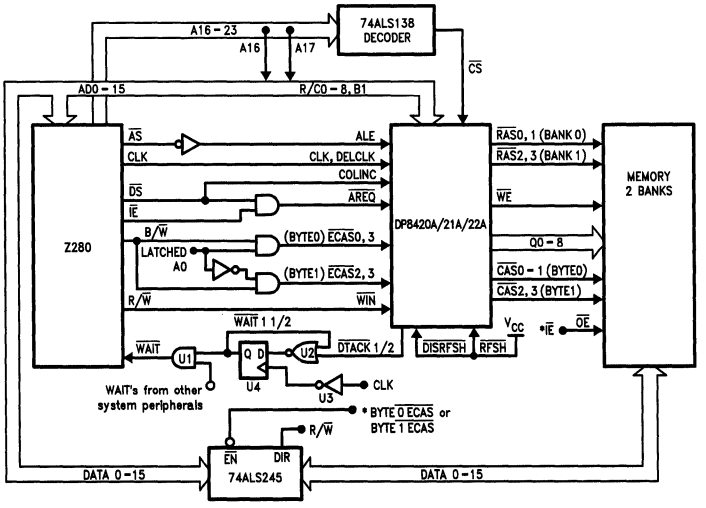
   Since the $\overline{AREQ}$ input of the DP8422A will go high from $\overline{DS}$ and $\overline{IE}$ both being high the $\overline{AREQ}$ high setup to clock rising edge (DP8422A parameter #29b, 19 ns) parameter is violated. This means that the rising clock edge following $\overline{AREQ}$ high may or may not be counted.

Since that first rising clock edge could be counted, and would give less $\overline{RAS}$ precharge time, we must assume this condition in the calculation of the minimum $\overline{RAS}$ precharge. Therefore:

200 ns (2 clock periods) − 50 ns (half clock period before both $\overline{IE}$ and $\overline{DS}$ transition high) − 35 ns ($\overline{IE}$ and $\overline{DS}$ high, Z280 parameters #8 and #19) − 5.5 ns (74AS08 max delay) − 16 ns (DP8422A $\overline{RAS}$ high to $\overline{RAS}$ low difference parameter #50) = 93.5 ns

Therefore, the user should guarantee that the DRAM he is using needs a $\overline{RAS}$ precharge time of 93.5 ns or less. If more $\overline{RAS}$ precharge time is needed the user should program the DP8422A with 3 periods of $\overline{RAS}$ precharge (R0, R1) during programming.

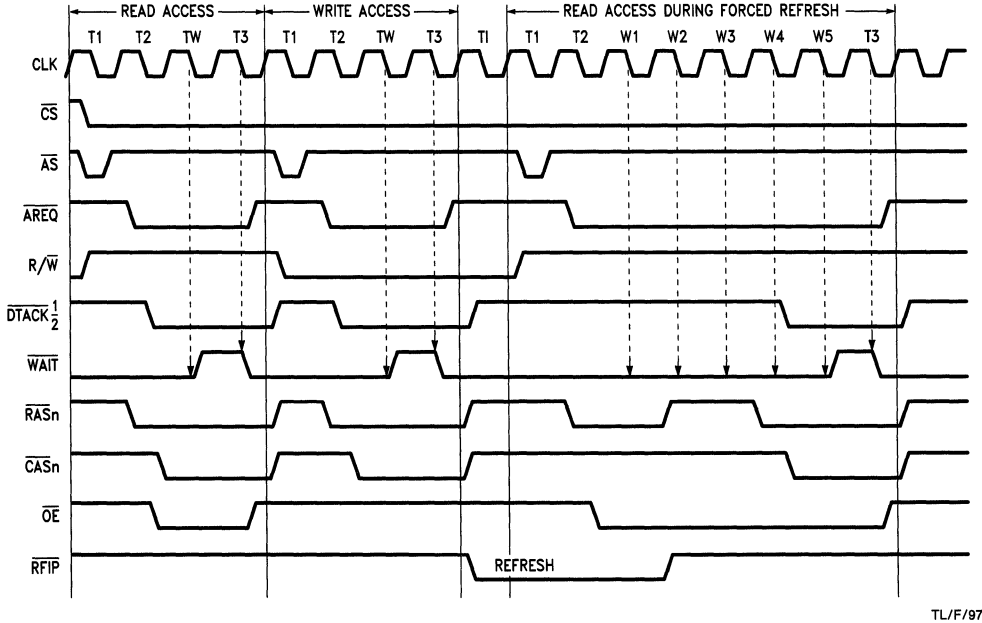
Note: Calculations can be performed for different frequencies and/or different combinations of wait states by substatuting the appropriate values into the above equations.



TL/F/9740–1

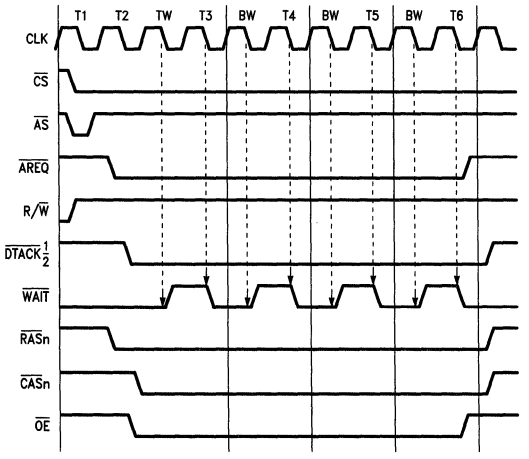
*The user may want to gate $\overline{CS}$ ("OR" Gate) with the signals that produce $\overline{OE}$ to the DRAMs and $\overline{EN}$ to the transceivers

FIGURE 1. 10 MHz Z280 Design (Z-bus Interface), 1 Wait State in Normal Accesses, 1 Wait State in Burst Accesses



TL/F/9740–2

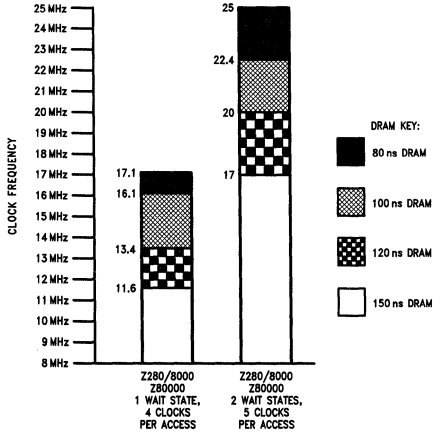
FIGURE 2. Z280 Access Cycles and Refresh (1 Wait State during Normal Access Cycles)



TL/D/9740-4

FIGURE 3. Z280 Burst Access Cycle (1 Wait State in Normal and Burst Accesses)

DRAM Speed vs Processor Speed (DRAM Speed References the RAS Access Time, t_{RAC}, of the DRAM. Using DP8422A-25 Timing Specifications)



TL/D/9740-5

Interfacing the Dual Port DP8422A to the TMS320C30 and the VME Bus

National Semiconductor
Application Note 642
Lawson Chang

INTRODUCTION

This application note describes how to interface the TMS320C30 Digital Signal Processor (DSP) and the VME bus to the DP8422A DRAM Controller. This system is running at 16 MHz. It is assumed that the reader is familiar with the DP8422A, TMS320C30, and VME bus operations.

DESCRIPTION

This design consists of Port A of the DP8422A interfaced to the primary bus of TMS320C30 DSP and Port B interfaced to the VME bus. The DP8422A is operated in access Mode 1 and uses the interleaving capability on chip. A Port A access cycle begins when the TMS320C30 places a valid address on the address bus and asserts the strobe (/STRB) signal, only if a refresh or Port B (VME bus) access is not in progress. GRANTB of the DP8422A indicates which port is currently granted to do an access. Port A is the default port upon power up. This design accommodates 4 banks of DRAM (256K x 4), 32 bits in each bank, giving maximum memory capacity of 4M bytes. The schematic diagram is shown in *Figure 1* and simple timing diagrams are shown in *Figures 2* and *3*.

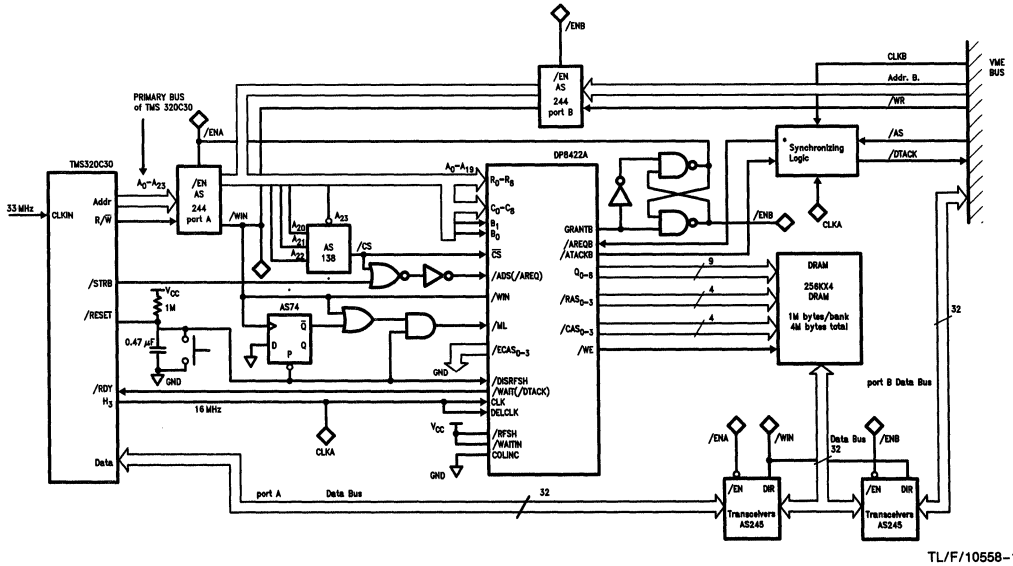
PROGRAMMING MODE AND BITS

Programming the DP8422A is on the first TMS320C30 DSP write after power up. 60 ms initialization period is needed right after this chip access write access programming.

Programming Bits

u = user defined, x = don't care.

R0, R1 = u, u
R2, R3 = u, u
R4, R5, R6 = x, x, x
R7 = 1
R8 = 0
R9 = u
C0, C1, C2 = 0 1 0 (16 MHz)
C3 = 0
C4, C5, C6 = u, u, u (or 0, 1, 1)
C7, C8 = u, u
C9 = 1
B0 = 0
B1 = 1



TL/F/10558–1

FIGURE 1. Interfacing DP8422A/TMS320C30 Schematic Diagram (Interleave Mode)

*Please refer to Interfacing the DP8422A to an Asychronous Port B in a Dual 68020 System application note.

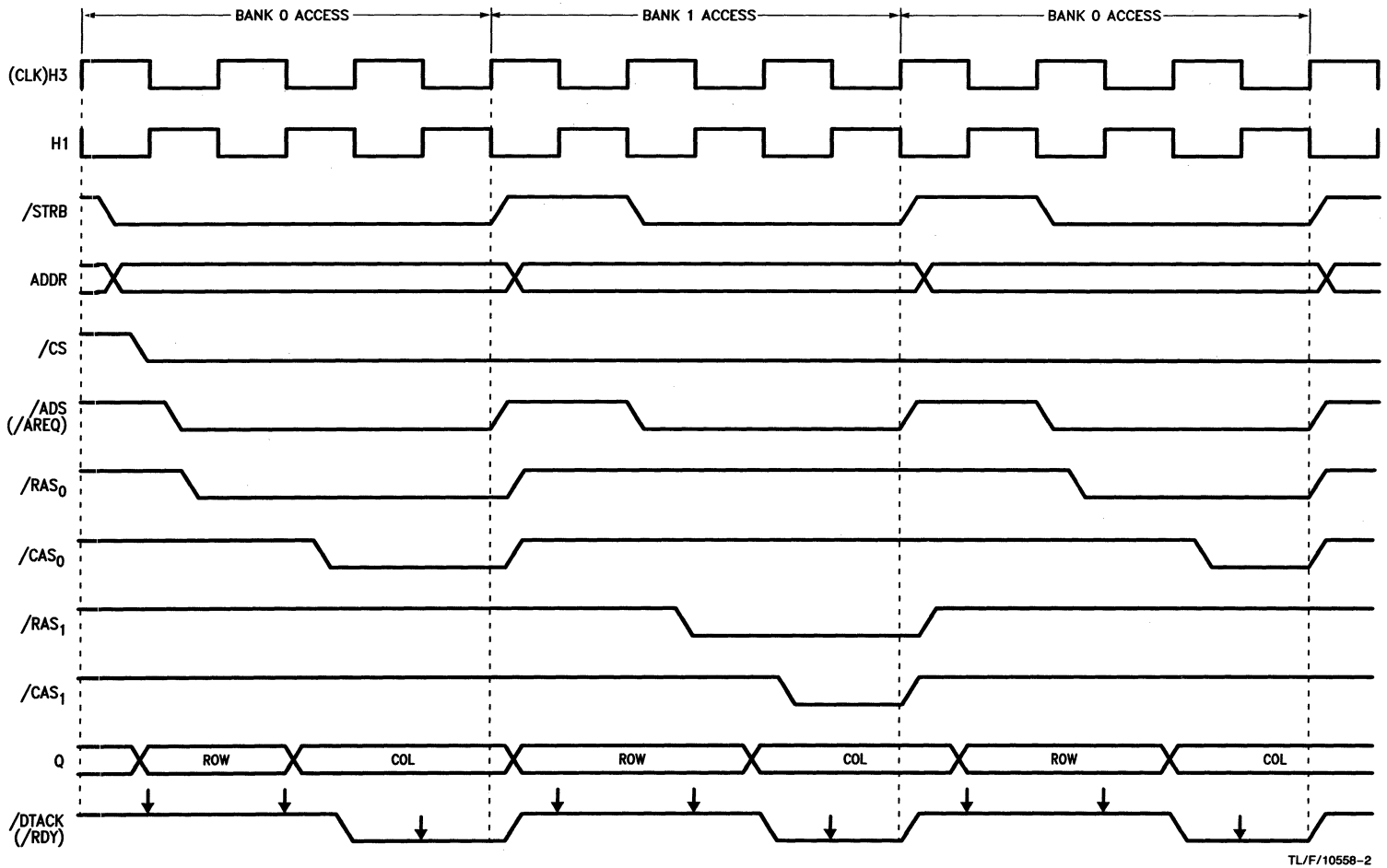
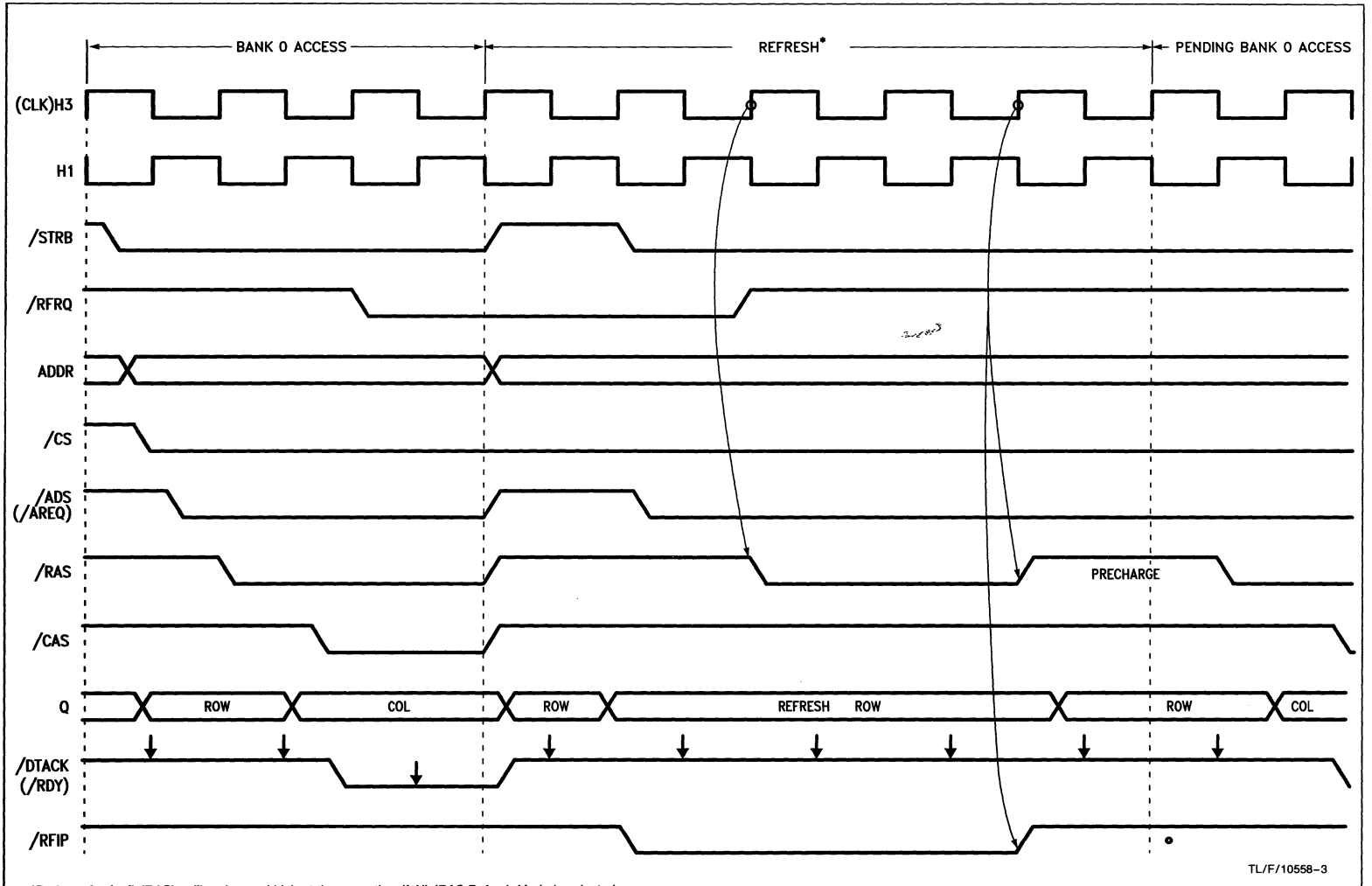


FIGURE 2. Timing Diagram for Interleave Access between Bank 0 and Bank 1



*During refresh all /RAS's will go low and high at the same time if All /RAS Refresh Mode is selected.

FIGURE 3. Timing Diagram for Refresh after an Access

Section 5

Microprocessor Application for the NS32CG821

Section 5 Contents

Interfacing the NS32CG821 to the NS32CG16

National Semiconductor
Application Note 576
Chris Koehle
Rich Levin

INTRODUCTION

This application note explains how to interface the NS32CG821 to the NS32CG16 microprocessor. It is assumed that the reader is familiar with the NS32CG16 access cycles and operation of the NS32CG821.

DESIGN DESCRIPTION

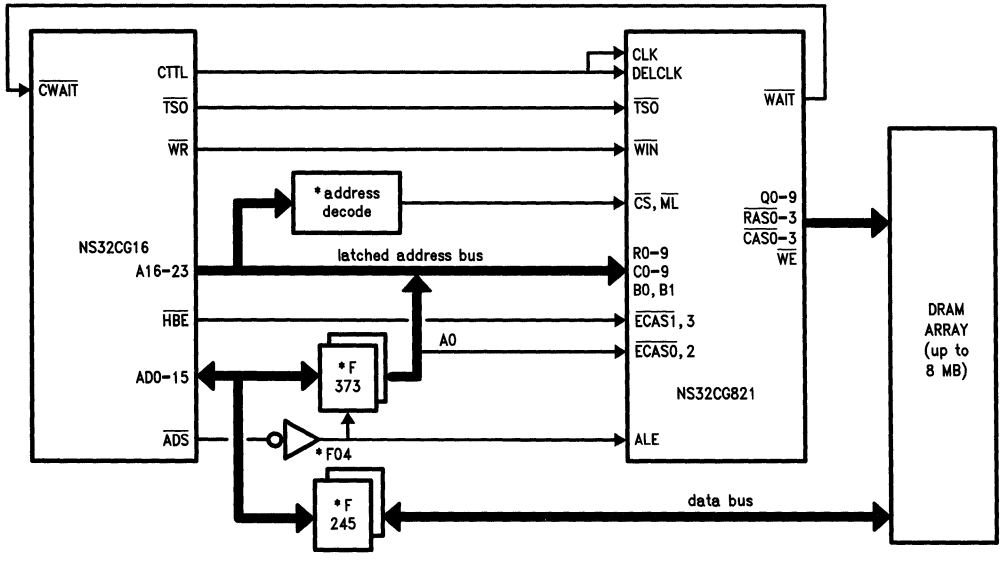
This design is a simple circuit to interface the NS32CG821 to the NS32CG16 and up to 8 Mbytes of DRAM. An access cycle begins when the NS32CG16 asserts the $\overline{ADS}$ signal and places a valid address on the bus. The $\overline{ADS}$ places a pair of 74F373 fall-through latches in fall-through mode and on the negating edge of $\overline{ADS}$ latches the address to guarantee that the address is valid throughout the entire access. The $\overline{ADS}$ signal is inverted to produce the signal ALE to the NS32CG821. On the next rising clock edge, after the ALE signal is asserted, the NS32CG821 will assert $\overline{RAS}$. After guaranteeing the row address hold time, t_{RAH}, the NS32CG821 will place the column address on the DRAM address bus, guarantee the column address setup time and assert $\overline{CAS}$. During read cycles, the DRAM will place valid data on the bus after the DRAM, t_{CAC}, timing has been met. During write cycles, $\overline{CAS}$ will be delayed until after T3re, to ensure that the CPU's write data is valid before $\overline{CAS}$ is asserted.

The N32CG821 will also take complete care of the DRAM's refresh needs. There is an internal 15 microsecond timer, and a refresh address counter. Refresh access arbitration will be controlled by an internal state machine. It will allow current cycles to complete before starting the refresh cycle. If a refresh cycle is in progress the NS32CG16 will be held off completing the access by asserting the $\overline{CWAIT}$ signal to the NS32CG16.

During programming of the chip, it is recommended that the user gate $\overline{ML}$ (Mode Load) and ($\overline{TSO}$) (Timing State Output) for the connection onto the $\overline{ML}$ pin of the NS32CG821. This is to ensure that the chip will be programmed while a valid access address is present.

Timing parameters are referenced to the numbers shown in the NS32CG821 data sheet, and are included in each equation in *italics* to indicate the target specifications that need to be satisfied. Times that begin with a "$" refer to the NS32CG821 data sheet unless otherwise stated times use "NS32CG821-20" part's parameters with heavy loading; these times are generally worse than the "NS32CG821-25" part. Times that begin with a "#" refer to the NS32CG16 data sheet. Equations are provided so that the user can calculate timing based on their frequency and application.

NS32CG16-NS32CG821 Connection Diagram



TL/F/10130–1

*Standard components in any NS32CG16 design.

DESIGN TIMING PARAMETERS

Timing diagrams are supplied further on in this document.

Clock Period = $T_{cp}10$ = 100 ns @ 10 MHz
= $T_{cp}15$ = 66 ns @ 15 MHz

$300: $\overline{CS}$ Asserted to CLK High (only A20–A23 are used for decode)

14 ns Min @ NS32CG821-20, 13 ns Min @ NS32CG821-25

= T1 − (CTTL to address valid + B-PAL delay)
= T_{cp} − #t_{AHv} − t_{Bpal}
= 100 ns − 40 ns − 15 ns

= 45 ns @ 10 MHz

= 66 ns − 30 ns − 15 ns

= 21 ns @ 15 MHz

$301a: ALE Setup to CLK High

16 ns Min @ NS32CG821-20,
15 ns Min @ NS32CG821-25

= T1 − Inverter Max − CTTL to $\overline{ADS}$
= T_{cp} − t_{plh} − #t_{ADSa}
= 100 ns − 5 ns − 35 ns

= 60 ns @ 10 MHz

= 66 ns − 5 ns − 26 ns

= 35 ns @ 15 MHz

$302: ALE Pulse Width

18 ns Min @ NS32CG821-20,
13 ns Min @ NS32CG821-25

= #t_{ADSw}

= 30 ns @ 10 MHz
= 25 ns @ 15 MHz

$303 & $304: Address Setup to CLK

20 ns Min @ NS32CG821-20,
18 ns Min @ NS32CG821-25

= T1 − CTTL to Address − F373 In to Out
= T_{cp} − t_{AHv} − t_{phl}
= 100 ns − 40 ns − 6 ns

= 54 ns @ 10 MHz

= 66 ns − 30 ns − 6 ns

= 30 ns @ 15 MHz

$310: $\overline{WIN}$ Setup to CLK High that starts the access $\overline{RAS}$ (to guarantee $\overline{CAS}$ is delayed)

−21 ns Min @ NS32CG821-20 &
−16 ns Min @ NS32CG821-25

= T1 − (time until $\overline{WR}$ active)
= T_{cp} − (T_{cp} + t_{WRa}
= 100 ns − (100 ns + 20 ns)

= −20 ns @ 10 MHz

= 66 ns − (66 ns + 15 ns)

= −15 ns @ 15 MHz

Timing parameter to guarantee $\overline{RAS}$ Precharge with 2 Clock Precharge

$29b: $\overline{TSO}$ negated Setup to CLK High with > 1 Period of Precharge

19 ns Min @ NS32CG821-20,
15 ns Min @ NS32CG821-25

= T4 − CTTL to $\overline{TSO}$ Inactive
= T_{cp} − #TSO_{ia}
= 100 ns −12 ns

= 88 ns @ 10 MHz

= 66 ns −10 ns

= 56 ns @ 15 MHz

$\overline{CWAIT}$: Setup for Wait States, before T3re

#tCWs, 20 ns Min @ 10 MHz
& 15 MHz for NS32CG16

= T1 + T2 − (Time in T1 until $\overline{CS}$ Asserted + $\overline{CS}$ to $\overline{WAIT}$ Asserted)
= $2T_{cp}$ − (#T_{AHv} + t_{Bpal} + $311)
= 200 ns − (40 ns + 15 ns + 26 ns)

= 119 ns @ 10 MHz

= 133 ns − (30 ns + 15 ns + 22 ns)

= 66 ns @ 15 MHz

$\overline{CWAIT}$: Hold, after T3re

#tCWh, 5 ns Min @ 10 MHz
& 15 MHz for NS32CG16

= T3re to $\overline{WAIT}$ Negated
= $17 Min

= 7 ns @ NS32CG821-20
= 7 ns @ NS32CG821-25

$\overline{CWAIT}$: Setup for Termination of Access

#tCWs, 20 ns Min
@ 10 MHz & 15 MHz
for NS32CG16

= T3 − CLK to $\overline{WAIT}$ Negated
= T_{cp} − $17 Max
= 100 ns − 39 ns

= 61 ns @ 10 MHz **NS32CG821-20 Part**

= 66 ns − 39 ns

= 27 ns @ 15 MHz **NS32CG821-20 Part**

DRAM SPECIFIC TIMING WHEN USING THE NS32CG16 @ 10 MHz

Since systems and DRAM times vary, the user is encouraged to change the following equations to match their system requirements. Timing has been supplied for systems with 0 or 1 wait states. The times assume worst case load. As such, the time will improve with lower circuit loads.

Zero Wait States Analysis for a 10 MHz NS32CG16

t_{RAC} = T2 + T3 − CLK to $\overline{RAS}$ − Transceiver Delay − Data Setup
= $2T_{cp}$ − \$307 − t_{phl} − #t_{Dls}
= 200 ns − 32 ns − 7 ns − 18 ns
= 143 ns @ 10 MHz **NS32CG821-20 Part**

t_{AA} = T2 + T3 − CLK to Column Address Valid − Transceiver Delay − Data Setup
= $2T_{cp}$ − \$316act − t_{phl} − #t_{Dls}
= 200 ns − 87 ns − 7 ns − 18 ns
= 88 ns @ 10 MHz **NS32CG821-20 Part**

t_{CAC} = T2 + T3 − CLK to $\overline{CAS}$ − Transceiver Delay − Data Setup
= $2T_{cp}$ − \$308act − t_{phl} − #t_{Dls}
= 200 ns − 89 ns − 7 ns − 18 ns
= 86 ns @ 10 MHz **NS32CG821-20 Part**

One Wait State Analysis for a 10 MHz NS32CG16

t_{RAC} = T2 + T_w + T3 − CLK to $\overline{RAS}$ − Transceiver Delay − Data Setup
= $3T_{cp}$ − \$307 − t_{phl} − #t_{Dls}
= 300 ns − 32 ns − 7 ns − 18 ns
= 243 ns @ 10 MHz **NS32CG821-20 Part**

t_{AA} = T2 + T_w + T3 − CLK to ColumnAddress Valid − Transceiver Delay − Data Setup
= $3T_{cp}$ − \$316act − t_{phl} − #t_{Dls}
= 300 ns − 87 ns − 7 ns − 18 ns
= 188 ns @ 10 MHz **NS32CG821-20 Part**

t_{CAC} = T2 + T_w + T3 − CLK to $\overline{CAS}$ − Transceiver Delay − Data Setup
= $3T_{cp}$ − \$308act − t_{phl} − #t_{Dls}
= 300 ns − 89 ns − 7 ns − 18 ns
= 186 ns @ 10 MHz **NS32CG821-20 Part**

$\overline{RAS}$ Precharge Parameter (for both zero and one wait state)

t_{RP} $\overline{RAS}$ Precharge Time
see DRAM data sheet (usually < 100 ns)
= T4 + T1 − CTTL to $\overline{TSO}$ inactive + [(CLK high to $\overline{RAS}$ asserted) − ($\overline{TSO}$ negated to $\overline{RAS}$ negated)]
= $2T_{cp}$ − #t_{TSOia} + \$53
= 200 ns − 12 ns + 4 ns
= 192 ns @ 10 MHz **for NS32CG16**

$\overline{RAS}$ Low During Refresh (for both zero and one wait state)

t_{RAS} = Programmed Clocks − [(CLK High to Refresh $\overline{RAS}$ asserted) − (CLK High to Refresh $\overline{RAS}$ negated)]
= $2T_{cp}$ − \$55
= 200 ns − 5 ns
= 195 ns @ 10 MHz **for NS32CG16**

DRAM SPECIFIC TIMING WHEN USING THE NS32CG16 @ 15 MHz

The input DELCLK controls the internal delay line and should be a multiple of 2 MHz. Since DELCLK is 15 MHz (when directly connected to CTTL) and is not a multiple of 2 MHz, t_{RAH} and t_{ASC} will vary from the programmed times according to the equations listed below.

In addition, please note the following pertaining to the timing equations:

1. Times for t_{RAH} and t_{ASC} at light loads are specified 2 ns longer than for normal-heavy loads. (See data sheet specifications.)
2. Light load is defined as 4 banks of four x 4 DRAMs
3. When using normal-heavy loads at 15 MHz a DELCLK divisor of 8 is used and when using light loads at 15 MHz a DELCLK divisor of 7 is used.

t_{RAHact} = 30 * [(DELCLK divisor * 2 MHz/DELCLK freq) − 1] ns + 15 ns (+2 ns for light load only)
= 30 * [(8 * 2 MHz/15 MHz) − 1]ns + 15 ns
= 17 ns @ 15 MHz Normal-Heavy Loads
= 30 * [(7 * 2 MHz/15 MHz) − 1]ns + 15 ns + 2 ns
= 15 ns @ 15 MHz Light Loads

t_{ASCact} = 15 * [DELCLK Divisor * 2 MHz/DELCLK Freq]ns − 15 ns (+2 ns for light load only)
= 15 * [8 * 2 MHz/15 MHz]ns − 15 ns
= 1 ns @ 15 MHz Normal-Heavy Loads
= 15 * [(7 * 2 MHz/15 MHz] ns − 15 ns + 2 ns
= 1 ns @ 15 MHz Light Loads

PARAMETER ADJUSTMENTS FOR 15 MHz DELCLK DUE TO CHANGED t_{RAH} and t_{ASC}

\$308act CLK High to $\overline{CAS}$ = Data Sheet Spec + (Actual t_{RAH} − Spec t_{RAH}) + (Actual t_{ASC} − Spec t_{ASC})
= 79 ns + (17 ns − 15 ns) + (1 ns − 0 ns)
= 82 ns @ 15 MHz Heavy Load **NS32CG821-25 Part**
= 89 ns + (17 ns − 15 ns) + (1 ns − 0 ns)
= 92 ns @ 15 MHz Heavy Load **NS32CG821-20 Part**
= 72 ns + (15 ns − 17 ns) + (1 ns − 2 ns)
= 69 ns @ 15 MHz Light Load **NS32CG821-25 Part**
= 81 ns + (15 ns − 17 ns) + (1 ns − 2 ns)
= 78 ns @ 15 MHz Light Load **NS32CG821-20 Part**

5

$316act CLK High to Column Address Valid

= Data sheet Spec + (Actual t_{RAH} − Specified t_{RAH})

= 75 ns + (17 ns − 15 ns)

= 77 ns @ 15 MHz Normal-Heavy Load **NS32CG821-20 Part**

= 87 ns + (17 ns − 15 ns)

= 89 ns @ 15 MHz Normal-Heavy Load **NS32CG821-20 Part**

= 66 ns + (15 ns − 17 ns)

= 64 ns @ 15 MHz Light Load **NS32CG821-25 Part**

= 78 ns + (15 ns − 17 ns)

= 76 ns @ 15 MHz Light Load **NS32CG821-20 Part**

ZERO WAIT STATES ANALYSIS FOR A 15 MHz NS32CG16

t_{RAC} = T2 + T3 − CLK to $\overline{RAS}$ − Transceiver Delay − Data Setup

= $2T_{cp}$ − $307 − t_{phl} − #t_{DIs}

= 133 ns − 26 ns − 7 ns − 15 ns

= 85 ns Normal-Heavy Load **NS32CG821-25 Part**

= 133 ns − 22 ns − 7 ns − 15 ns

= 89 ns Light Load **NS32CG821-25 Part**

t_{AA} = T2 + T3 − CLK to Column Address Valid − Transceiver Delay − Data Setup

= $2T_{cp}$ − $316act − t_{phl} − #t_{DIs}

= 133 ns − 77 ns − 7 ns − 15 ns

= 34 ns Normal-Heavy Load **NS32CG821-25 Part**

= 133 ns − 64 ns − 7 ns − 15 ns

= 47 ns Light Load **NS32CG821-25 Part**

t_{CAC} = T2 + T3 − CLK to $\overline{CAS}$ − Transceiver Delay − Data Setup

= $2T_{cp}$ − $308act − t_{phl} − #t_{DIs}

= 133 ns − 82 ns − 7 ns − 15 ns

= 29 ns Normal-Heavy Load **NS32CG821-25 Part**

= 133 ns − 69 ns − 7 ns − 15 ns

= 42 ns Light Load **NS32CG821-25 Part**

ONE WAIT STATE ANALYSIS FOR A 15 MHz NS32CG16

t_{RAC} = T2 + T_w + T3 − CLK to $\overline{RAS}$ − Transceiver Delay − Delay Setup

= $3T_{cp}$ − $307 − t_{phl} − #t_{DIs}

= 200 ns − 26 ns − 7 ns − 15 ns

= 152 ns Normal-Heavy Load **NS32CG821-25 Part**

= 200 ns − 22 ns − 7 ns − 15 ns

= 156 ns Light Load **NS32CG821-25 Part**

t_{AA} = T2 + T_w + T3 − CLK to Column Address Valid − Transceiver Delay − Data Setup

= $3T_{cp}$ − $316act − t_{phl} − #t_{DIs}

= 200 ns − 77 ns − 7 ns − 15 ns

= 101 ns Normal-Heavy Load **NS32CG821-25 Part**

= 200 ns − 64 ns − 7 ns − 15 ns

= 114 ns, Light Load **NS32CG821-25 Part**

t_{CAC} = T2 + T_w + T3 − CLK to $\overline{CAS}$ − Transceiver Delay − Data Setup $3T_{cp}$ − $308act − t_{phl} − #t_{DIs}

= 200 ns − 82 ns − 7 ns − 15 ns

= 96 ns Normal-Heavy Load **NS32CG821-25 Part**

= 200 ns − 69 ns − 7 ns − 15 ns

= 109 ns Light Load **NS32CG821-25 Part**

$\overline{RAS}$ PRECHARGE PARAMETER (FOR BOTH ZERO AND ONE WAIT STATE)

t_{RP} $\overline{RAS}$ Precharge Time

see DRAM data sheet (usually < 100 ns)

= T4 + T1 − CTTL to $\overline{TSO}$ Inactive + [(CLK High to $\overline{RAS}$ Asserted) − ($\overline{TSO}$ Negated to $\overline{RAS}$ Negated)]

= $2T_{cp}$ − t_{TSOia} + $53

= 133 ns − 10 ns + 4 ns

= 127 ns @ 15 MHz for NS32CG16

$\overline{RAS}$ LOW DURING REFRESH (FOR BOTH ZERO AND ONE WAIT STATE)

t_{RAS} = Programmed Clocks − [(CLK High to Refresh $\overline{RAS}$ Asserted) − (CKL High to Refresh $\overline{RAS}$ Negated)]

= $2T_{cp}$ − $55

= 133 ns − 6 ns

= 127 ns @ 15 MHz for NS32CG16

Programming Bits*

Bit	Value	Description
R1, R0	1, 0	$\overline{RAS}$ Low during Refresh = 2T $\overline{RAS}$ Precharge Time = 2T
R3, R2	0, 0 1, 1	No Wait States during Non-Delayed Access One Wait State during Non-Delayed Access
R5, R4	0, 0	No Wait States during Burst
R6	User Defined	Add Wait States with $\overline{WAITIN}$
R9	User Defined	Staggered or all $\overline{RAS}$ Refresh
C0, C1, C2	**	Divisor for DELCLK
C3	***	Time between Refreshes
C6, C5, C4	User Defined	Depends on User's DRAM Configurations
C7	1	Choose t_{ASC} = 0 ns
C8	1	Choose t_{RAH} = 15 ns
C9	1	Delay $\overline{CAS}$ for Write Accesses
B0	1	Address Latches are Fall Through

* $\overline{ECASO}$, B1, and R7 must be programmed low and R8 must be programmed high for operation of chip.

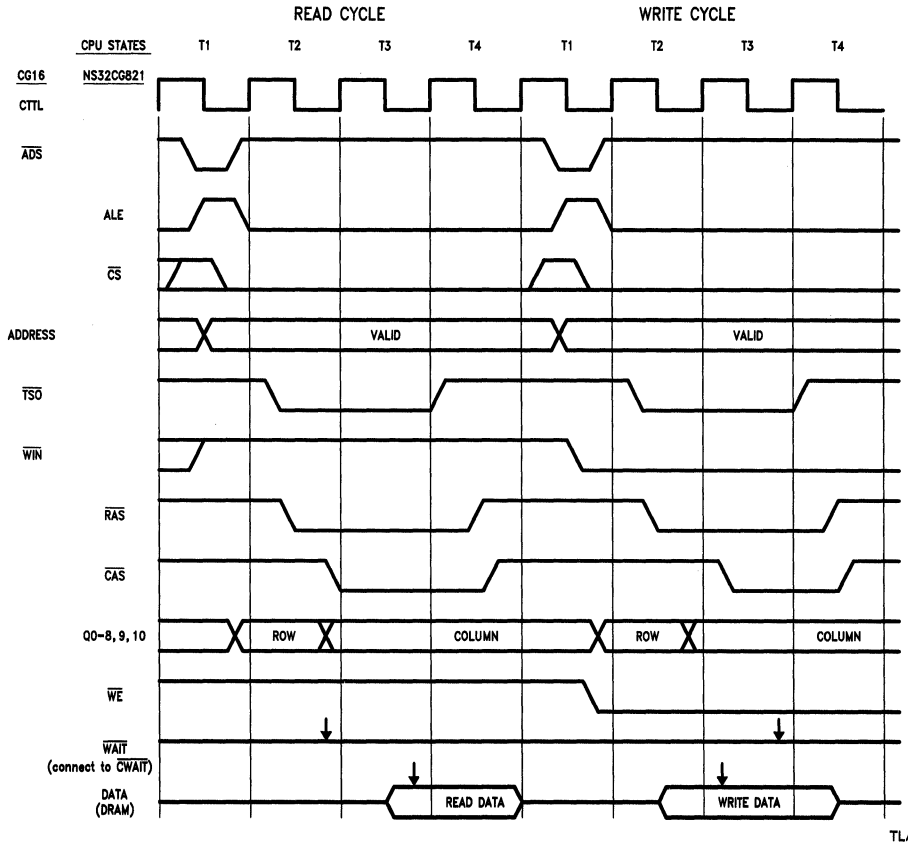
**Choose C2, C1, C0 =

1, 0, 1 for NS32CG16 @ 10 MHz
0, 1, 0 for NS32CG16 @ 15 MHz, w/Heavy Load
0, 1, 1 for NS32CG16 @ 15 MHz, w/Light Load

*** Choose C3 =

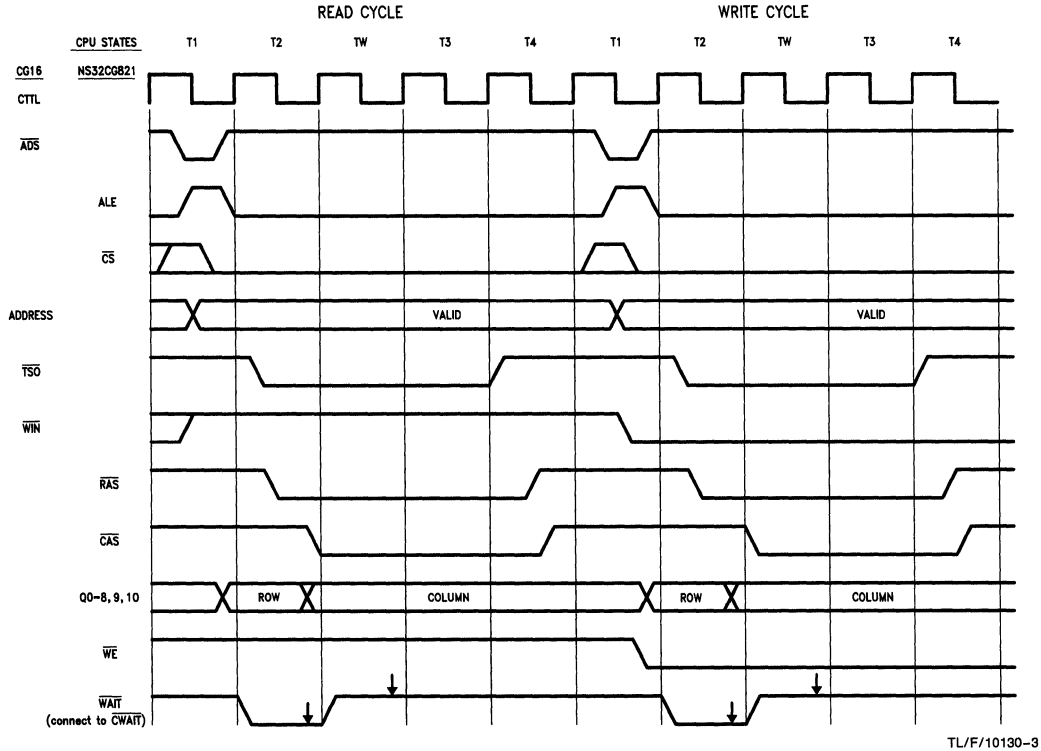
0 for NS32CGG16 @ 10 MHz
1 for NS32CG16 @ 15 MHz Normal–Heavy Load
0 for NS32CG16 @ 15 MHz Light Load

NS32CG16–NS32CG821 Dram Timing for 0 Wait States



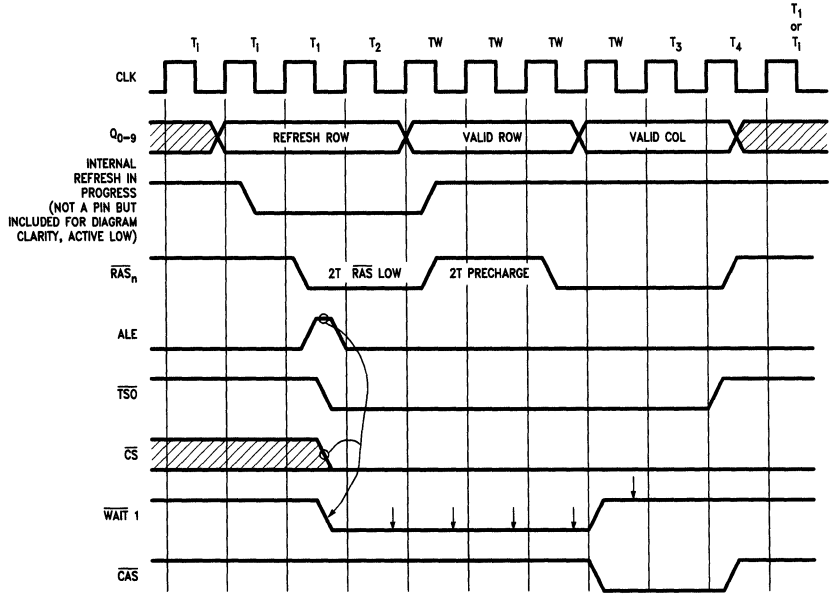
TL/F/10130–2

NS32CG16–NS32CG821 DRAM Timing with One Wait State



TL/F/10130–3

Access Request during an Internal Refresh Cycle (for One Wait State during Access)



TL/F/10130–4

Section 6

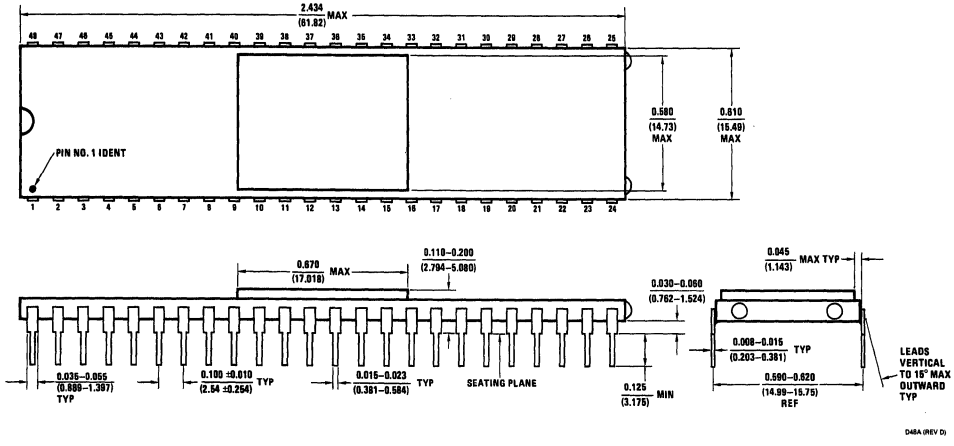
Physical Dimensions/ Appendices

Section 6 Contents

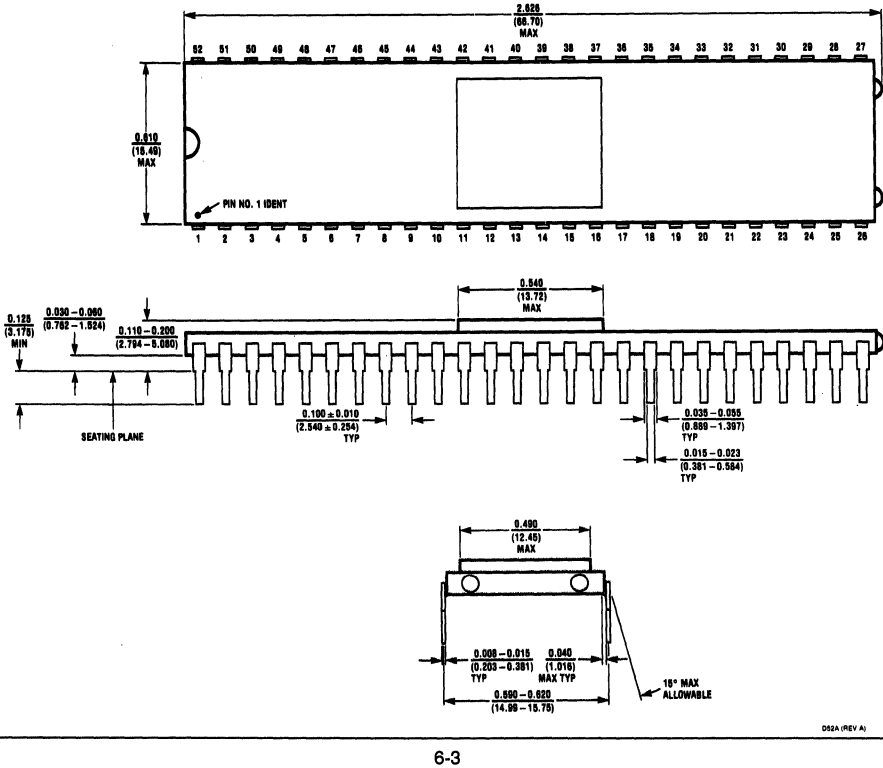
National Semiconductor

All dimensions are in inches (millimeters)

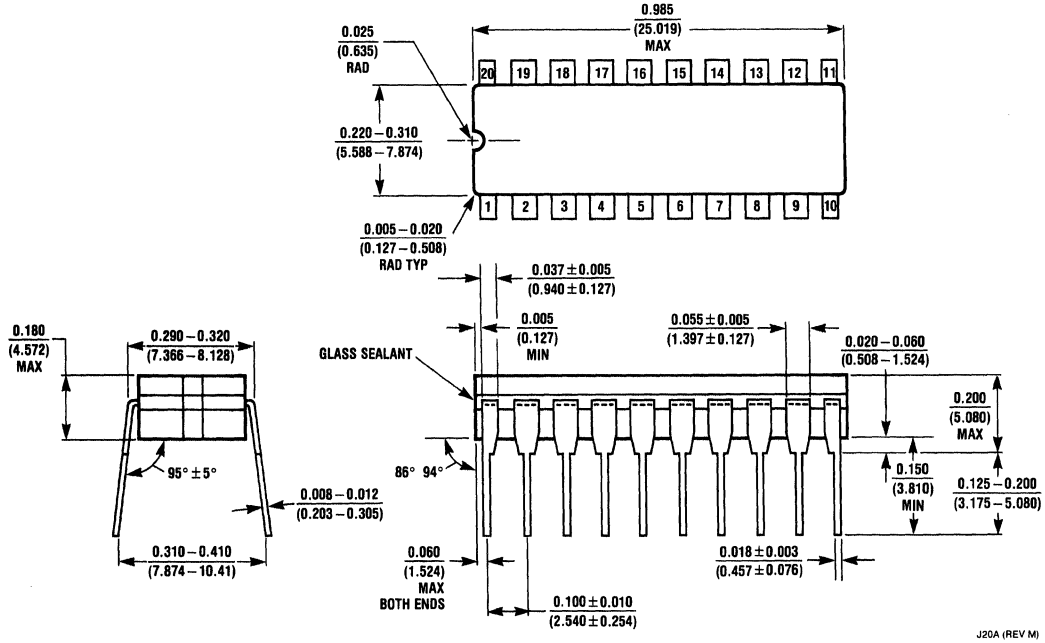
48 Lead Hermetic Dual-In-Line Package (D)
NS Package Number D48A



52 Lead Sidebraze Hermetic Dual-In-Line Package (D)
NS Package Number D52A

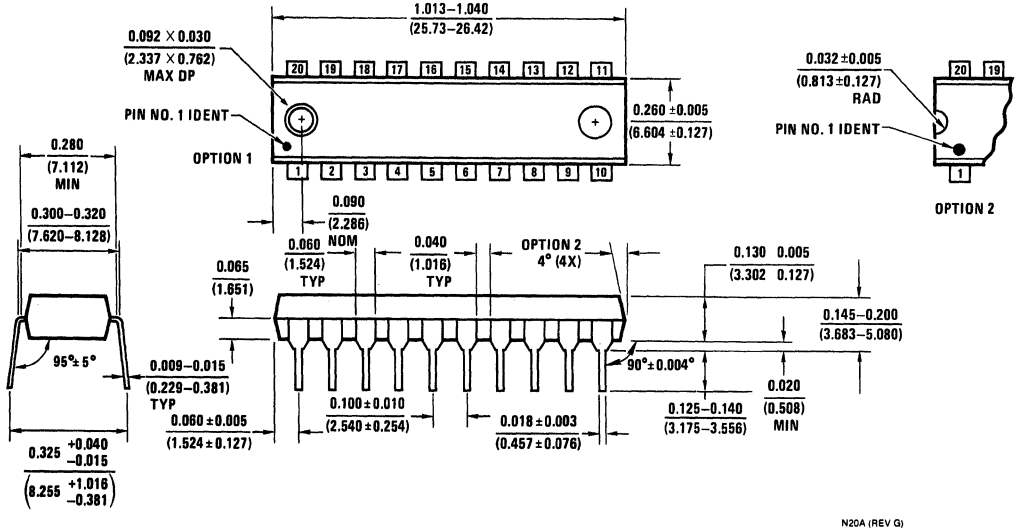


20 Lead Ceramic Dual-In-Line Package (J)
NS Package Number J20A



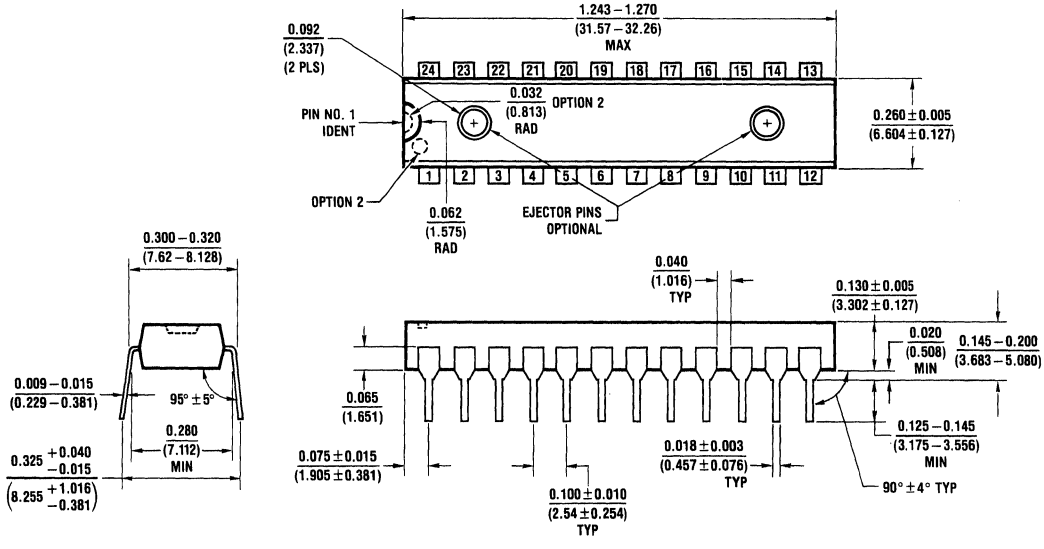
J20A (REV M)

20 Lead Molded Dual-In-Line Package (N)
NS Package Number N20A



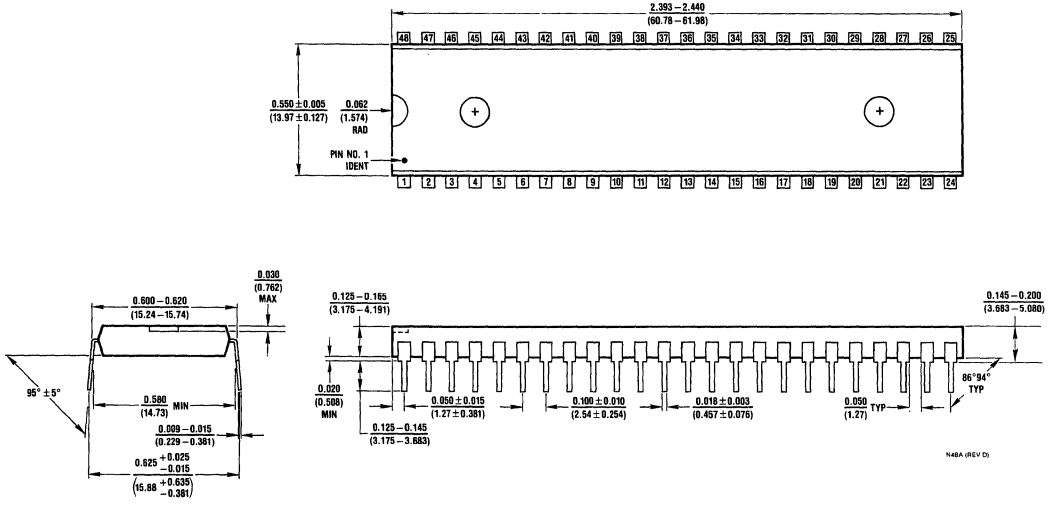
N20A (REV G)

24 Lead Skinny Dual-In-Line Package (0.300" Centers Molded) (N)
NS Package Number N24C



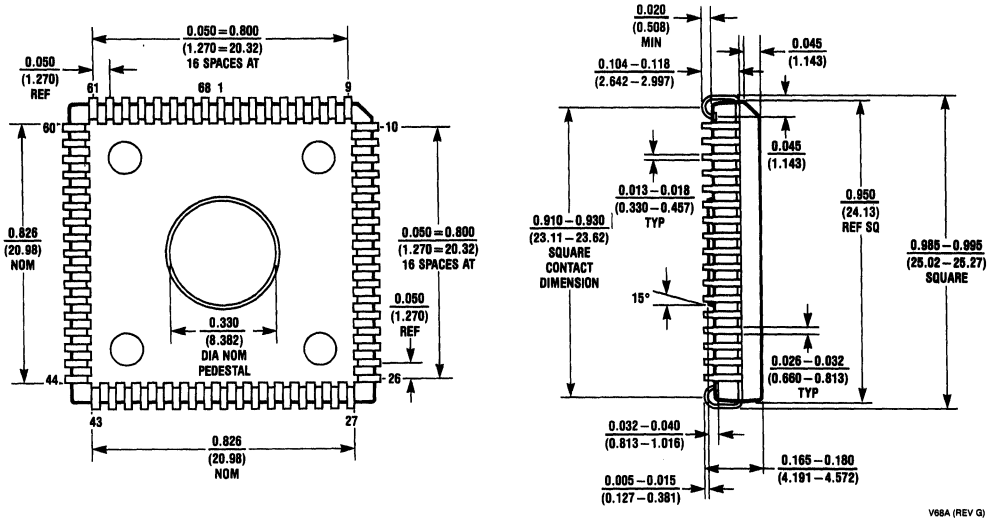
N24C (REV F)

48 Lead Molded Dual-In-Line Package (N)
NS Package Number N48A



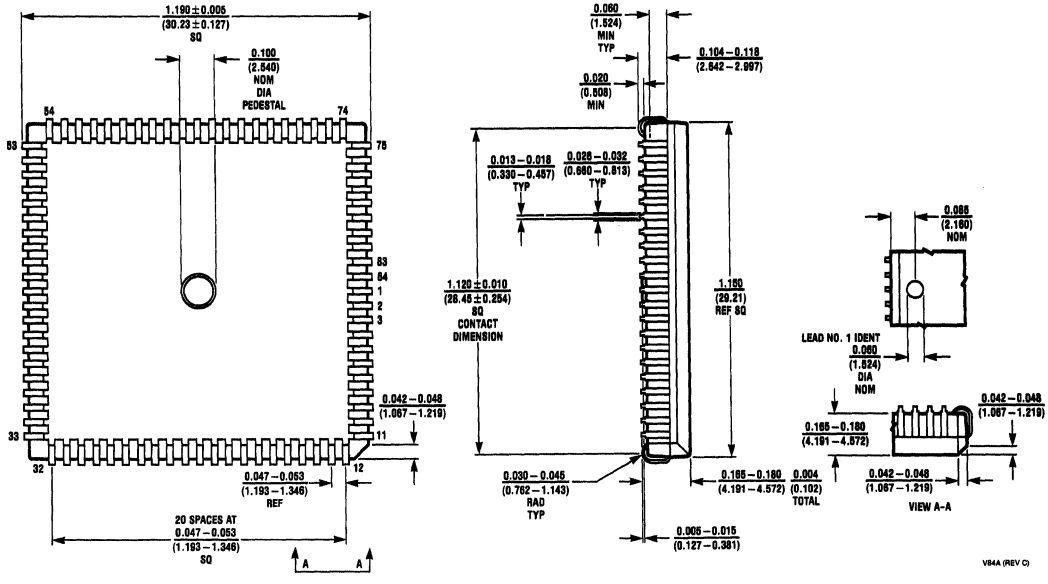
N48A (REV D)

68 Lead Plastic Chip Carrier (V)
NS Package Number V68A



V68A (REV G)

84 Lead Plastic Chip Carrier (V)
NS Package Number V84A



V84A (REV C)

NOTES

National Semiconductor

Bookshelf of Technical Support Information

National Semiconductor Corporation recognizes the need to keep you informed about the availability of current technical literature.

This bookshelf is a compilation of books that are currently available. The listing that follows shows the publication year and section contents for each book.

Please contact your local National sales office for possible complimentary copies. A listing of sales offices follows this bookshelf.

We are interested in your comments on our technical literature and your suggestions for improvement.

Please send them to:

Technical Communications Dept. M/S 16300
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090

ALS/AS LOGIC DATABOOK—1987

Introduction to Bipolar Logic • Advanced Low Power Schottky • Advanced Schottky

ASIC DESIGN MANUAL/GATE ARRAYS & STANDARD CELLS—1987

SSI/MSI Functions • Peripheral Functions • LSI/VLSI Functions • Design Guidelines • Packaging

CMOS LOGIC DATABOOK—1988

CMOS AC Switching Test Circuits and Timing Waveforms • CMOS Application Notes • MM54HC/MM74HC MM54HCT/MM74HCT • CD4XXX • MM54CXXX/MM74CXXX • Surface Mount

DATA ACQUISITION LINEAR DEVICES—1989

Active Filters • Analog Switches/Multiplexers • Analog-to-Digital Converters • Digital-to-Analog Converters Sample and Hold • Temperature Sensors • Voltage Regulators • Surface Mount

DATA COMMUNICATION/LAN/UART DATABOOK—1989

LAN IEEE 802.3 • High Speed Serial/IBM Data Communications • ISDN Components • UARTs Modems • Transmission Line Drivers/Receivers

DISCRETE SEMICONDUCTOR PRODUCTS DATABOOK—1989

Selection Guide and Cross Reference Guides • Diodes • Bipolar NPN Transistors Bipolar PNP Transistors • JFET Transistors • Surface Mount Products • Pro-Electron Series Consumer Series • Power Components • Transistor Datasheets • Process Characteristics

DRAM MANAGEMENT HANDBOOK—1989

Dynamic Memory Control • Error Detection and Correction • Microprocessor Applications for the DP8408A/09A/17/18/19/28/29 • Microprocessor Applications for the DP8420A/21A/22A Microprocessor Applications for the NS32CG821

EMBEDDED SYSTEM PROCESSOR DATABOOK—1989

Embedded System Processor Overview • Central Processing Units • Slave Processors • Peripherals Development Systems and Software Tools

F100K DATABOOK—1989

Family Overview • F100K Datasheets • 11C Datasheets • 10K and 100K Memory Datasheets Design Guide • Circuit Basics • Logic Design • Transmission Line Concepts • System Considerations Power Distribution and Thermal Considerations • Testing Techniques • Quality Assurance and Reliability

FACT™ ADVANCED CMOS LOGIC DATABOOK—1989

Description and Family Characteristics • Ratings, Specifications and Waveforms
Design Considerations • 54AC/74ACXXX • 54ACT/74ACTXXX

FAST® ADVANCED SCHOTTKY TTL LOGIC DATABOOK—Rev. 1—1988

Circuit Characteristics • Ratings, Specifications and Waveforms • Design Considerations • 54F/74FXXX

FAST® APPLICATIONS HANDBOOK—REPRINT

Reprint of 1987 Fairchild FAST Applications Handbook
Contains application information on the FAST family: Introduction • Multiplexers • Decoders • Encoders
Operators • FIFOs • Counters • TTL Small Scale Integration • Line Driving and System Design
FAST Characteristics and Testing • Packaging Characteristics • Index

GENERAL PURPOSE LINEAR DEVICES DATABOOK—1989

Continuous Voltage Regulators • Switching Voltage Regulators • Operational Amplifiers • Buffers • Voltage Comparators
Instrumentation Amplifiers • Surface Mount

GRAPHICS HANDBOOK—1989

Advanced Graphics Chipset • DP8500 Development Tools • Application Notes

INTERFACE DATABOOK—1988

Transmission Line Drivers/Receivers • Bus Transceivers • Peripheral Power Drivers • Display Drivers
Memory Support • Microprocessor Support • Level Translators and Buffers • Frequency Synthesis • Hi-Rel Interface

LINEAR APPLICATIONS HANDBOOK—1986

The purpose of this handbook is to provide a fully indexed and cross-referenced collection of linear integrated circuit applications using both monolithic and hybrid circuits from National Semiconductor.

Individual application notes are normally written to explain the operation and use of one particular device or to detail various methods of accomplishing a given function. The organization of this handbook takes advantage of this innate coherence by keeping each application note intact, arranging them in numerical order, and providing a detailed Subject Index.

LS/S/TTL DATABOOK—1989

Contains former Fairchild Products
Introduction to Bipolar Logic • Low Power Schottky • Schottky • TTL • TTL—Low Power

MASS STORAGE HANDBOOK—1989

Rigid Disk Pulse Detectors • Rigid Disk Data Separators/Synchronizers and ENDECs
Rigid Disk Data Controller • SCSI Bus Interface Circuits • Floppy Disk Controllers • Disk Drive Interface Circuits
Rigid Disk Preamplifiers and Servo Control Circuits • Rigid Disk Microcontroller Circuits • Disk Interface Design Guide

MEMORY DATABOOK—1988

PROMs, EPROMs, EEPROMs • Flash EPROMs and EEPROMs • TTL I/O SRAMs
ECL I/O SRAMs • ECL I/O Memory Modules

MICROCONTROLLER DATABOOK—1989

COP400 Family • COP800 Family • COPS Applications • HPC Family • HPC Applications
MICROWIRE and MICROWIRE/PLUS Peripherals • Microcontroller Development Tools

MICROPROCESSOR DATABOOK—1989

Series 32000 Overview • Central Processing Units • Slave Processors • Peripherals
Development Systems and Software Tools • Application Notes • NSC800 Family

PROGRAMMABLE LOGIC DATABOOK & DESIGN MANUAL—1989

Product Line Overview • Datasheets • Designing with PLDs • PLD Design Methodology • PLD Design Development Tools
Fabrication of Programmable Logic • Application Examples

REAL TIME CLOCK HANDBOOK—1989

Real Time Clocks and Timer Clock Peripherals • Application Notes

RELIABILITY HANDBOOK—1986

Reliability and the Die • Internal Construction • Finished Package • MIL-STD-883 • MIL-M-38510
The Specification Development Process • Reliability and the Hybrid Device • VLSI/VHSIC Devices
Radiation Environment • Electrostatic Discharge • Discrete Device • Standardization
Quality Assurance and Reliability Engineering • Reliability and Documentation • Commercial Grade Device
European Reliability Programs • Reliability and the Cost of Semiconductor Ownership
Reliability Testing at National Semiconductor • The Total Military/Aerospace Standardization Program
883B/RETS™ Products • MILS/RETS™ Products • 883/RETS™ Hybrids • MIL-M-38510 Class B Products
Radiation Hardened Technology • Wafer Fabrication • Semiconductor Assembly and Packaging
Semiconductor Packages • Glossary of Terms • Key Government Agencies • AN/ Numbers and Acronyms
Bibliography • MIL-M-38510 and DESC Drawing Cross Listing

SPECIAL PURPOSE LINEAR DEVICES DATABOOK—1989

Audio Circuits • Radio Circuits • Video Circuits • Motion Control Circuits • Special Function Circuits
Surface Mount

TELECOMMUNICATIONS—1987

Line Card Components • Integrated Services Digital Network Components • Modems
Analog Telephone Components • Application Notes

National Semiconductor

National Semiconductor Corporation
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090
Tel: (408) 721-5000
TWX: (910) 339-9240

SALES OFFICES (Continued)

INTERNATIONAL OFFICES

Electronica NSC de Mexico SA
Juventino Rosas No. 118-2
Col Guadalupe Inn
Mexico, 01020 D.F. Mexico
Tel: 52-5-524-9402

National Semicondutores Do Brasil Ltda.
Av. Brig. Faria Lima, 1383
6.0 Andor-Conj. 62
01451 Sao Paulo, SP, Brasil
Tel: (55/11) 212-5066
Fax: (55/11) 211-1181 NSBR BR

National Semiconductor GmbH
Industriestrasse 10
D-8080 Furstenfeldbruck
West Germany
Tel: (0-81-41) 103-0
Telex: 527-649
Fax: (08141) 103554

National Semiconductor (UK) Ltd.
The Maple, Kembrey Park
Swindon, Wiltshire SN2 6UT
United Kingdom
Tel: (07-93) 61-41-41
Telex: 444-674
Fax: (07-93) 69-75-22

National Semiconductor Benelux
Vorstlaan 100
B-1170 Brussels
Belgium
Tel: (02) 6-61-06-80
Telex: 61007
Fax: (02) 6-60-23-95

National Semiconductor (UK) Ltd.
Ringager 4A, 3
DK-2605 Brondby
Denmark
Tel: (02) 43-32-11
Telex: 15-179
Fax: (02) 43-31-11

National Semiconductor S.A.
Centre d'Affaires-La Boursidiere
Bâtiment Champagne, B.P. 90
Route Nationale 186
F-92357 Le Plessis Robinson
France
Tel: (1) 40-94-88-88
Telex: 631065
Fax: (1) 40-94-88-11

National Semiconductor (UK) Ltd.
Unit 2A
Clonskeagh Square
Clonskeagh Road
Dublin 14
Tel: (01) 69-55-89
Telex: 91047
Fax: (01) 69-55-89

National Semiconductor S.p.A.
Strada 7, Palazzo R/3
I-20089 Rozzano
Milanofiori
Italy
Tel: (02) 8242046/7/8/9
Twx: 352647
Fax: (02) 8254758

National Semiconductor S.p.A.
Via del Cararaggio, 107
00147 Rome
Italy
Tel: (06) 5-13-48-80
Fax: (06) 5-13-79-47

National Semiconductor (UK) Ltd.
P.O. Box 29
N-1321 Stabekk
Norway
Tel: (2) 12-53-70
Fax: (2) 12-53-75

National Semiconductor AB
Box 1009
Grosshandlarvagen 7
S-12123 Johanneshov
Sweden
Tel: (08) 7228050
Fax: (08) 7229095

National Semiconductor GmbH
Calle Agustin de Foxa, 27 (9°D)
28036 Madrid
Spain
Tel: (01) 733-2958
Telex: 46133
Fax: (01) 733-8018

National Semiconductor Switzerland
Alte Winterthurerstrasse 53
Postfach 567
Ch-8304 Wallisellen-Zurich
Switzerland
Tel: (01) 830-2727
Telex: 828-444
Fax: (01) 830-1900

National Semiconductor
Kauppakartanonkatu 7 A22
SF-00930 Helsinki
Finland
Tel: (90) 33-80-33
Telex: 126116
Fax: (90) 33-81-30

National Semiconductor
Postbus 90
1380 AB Weesp
The Netherlands
Tel: (0-29-40) 3-04-48
Telex: 10-956
Fax: (0-29-40) 3-04-30

National Semiconductor Japan Ltd.
Sanseido Bldg. 5F
4-15 Nishi Shinjuku
Shinjuku-ku
Tokyo 160 Japan
Tel: 3-299-7001
Fax: 3-299-7000

National Semiconductor Hong Kong Ltd.
Suite 513, 5th Floor,
Chinachem Golden Plaza,
77 Mody Road, Tsimshatsui East,
Kowloon, Hong Kong
Tel: 3-7231290
Telex: 52996 NSSEA HX
Fax: 3-3112536

National Semiconductor (Australia) PTY, Ltd.
1st Floor, 441 St. Kilda Rd.
Melbourne, 3004
Victory, Australia
Tel: (03) 267-5000
Fax: 61-3-2677458

National Semiconductor (PTE), Ltd.
200 Cantonment Road 13-01
Southpoint
Singapore 0208
Tel: 2252226
Telex: RS 33877

National Semiconductor (Far East) Ltd.
Taiwan Branch
P.O. Box 68-332 Taipei
7th Floor, Nan Shan Life Bldg.
302 Min Chuan East Road,
Taipei, Taiwan R.O.C.
Tel: (86) 02-501-7227
Telex: 22837 NSTW
Cable: NSTW TAIPEI

National Semiconductor (Far East) Ltd.
Korea Branch
13th Floor, Dai Han Life Insurance
63 Building,
60, Yoido-dong, Youngdeungpo-ku,
Seoul, Korea 150-763
Tel: (02) 784-8051/3, 785-0696/8
Telex: 24942 NSPKLO
Fax: (02) 784-8054

 TL2675 RRD-RRD60M099/Printed in U.S.A.