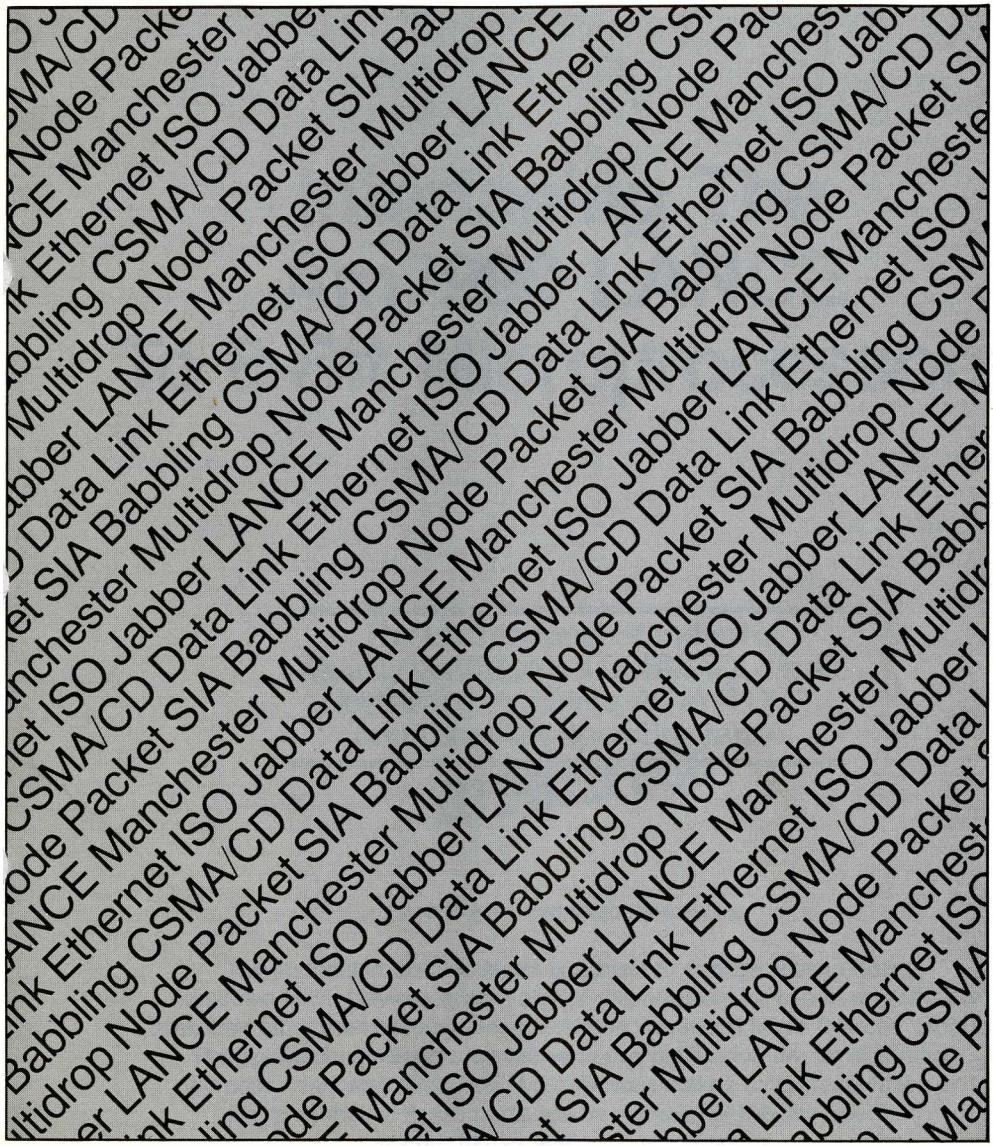
Interfacing The Am7990 LANCE To 8-Bit Microprocessors

Application Note
By Sherman Lee



Order #07422A

ADVANCED MICRO DEVICES

Advanced Micro Devices

Interfacing The Am7990 LANCE To 8-Bit Microprocessors

The International Standard of Quality guarantees a 0.05% AQL on all electrical parameters, AC and DC, over the entire operating range.

INT·STD·500

©1985 Advanced Micro Devices, Inc.

The material in this document is subject to change without notice. Advanced Micro Devices cannot accept responsibility for use of any circuitry described other than circuitry embodied in an Advanced Micro Devices' product. The applications software contained in this publication are for illustration purposes only and Advanced Micro Devices makes no representation or warranty that such programs will be suitable for the use specified without further testing or modification.

Printed in U.S.A.

by Sherman Lee
AMD Headquarters Applications

TABLE OF CONTENTS

FIGURES

SECTION 1
INTRODUCTION

This application note shows the conceptual and realistic implementation of interfaces between the Am7990 Local Area Network Controller for Ethernet (LANCE) and three popular 8-bit microprocessors with byte-wide memory data busses. This paper differs from previous application notes in two significant ways:

First, earlier application notes only showed 8-bit microprocessors interfaced to the LANCE via 16-bit memory data busses. But the reason for choosing an 8-bit over a 16-bit microprocessor in relatively small, cost sensitive systems is that an eight-bit wide memory reduces cost by requiring only half the number of memory devices compared to a 16-bit wide memory. The upcoming 256K x 1 DRAM will bring this cost advantage even to 256K byte systems. This manufacturing cost advantage warrants investigating the interface of the LANCE not only to 8-bit microprocessors but also to a byte-wide memory bus.

Second, many earlier papers describe the microprocessor to LANCE interface on a conceptual level, employing blocks labeled buffers or control. Such a perspective is excellent from a casual reader's standpoint, but somewhat vague and ineffective from an actual designer's viewpoint. So this application note illustrates not only the major function blocks but also the detailed logic implementation and subtle nuances within each critical block for three currently popular 8-bit microprocessors —Z80B, 8088 and 68008.

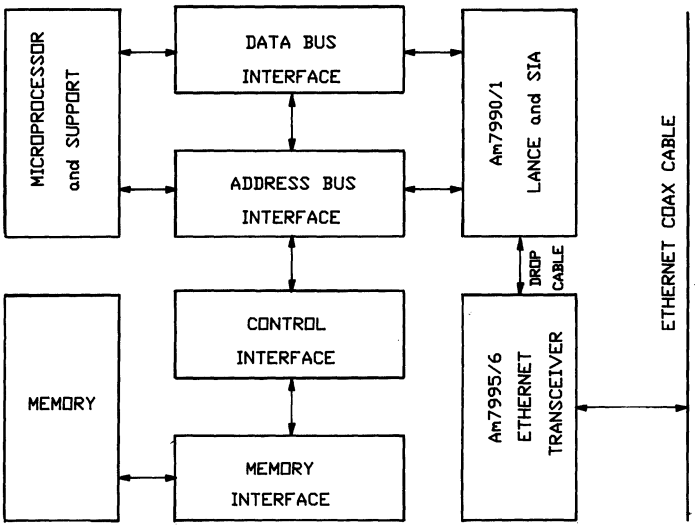
The logic implementation for the Z80B, Am8088, and the MC68008 microprocessors are discussed in the following three sections. The memory and SIA/Transceiver blocks have been deemphasized in order to highlight the important aspects of the LANCE interface. In addition, the logic designs presented have not been minimized in order to clearly show the system partitioning and the logic functionality. An actual design could be optimized by a slight reorganization of the block implementation.

SECTION 2
OVERVIEW

2.1 DESIGN CONCEPTS

The general block diagram (Figure 2-1) shows several major system blocks common to all three designs. As we progress through this paper, the large conceptual blocks will be broken down into their integral sections. This wil show their commonality in design methodology which will be used to relate and create new logic ideas and implementations.

Programmable Array Logic (PAL Devices) are used extensively in the designs in order to provide maximum design flexibility and lowest component count. With the aid of this high level of logic integration, the resultant peripheral interface permits the LANCE to appear as a coprocessor device sitting on the microprocessor's local data bus.



07422A-1

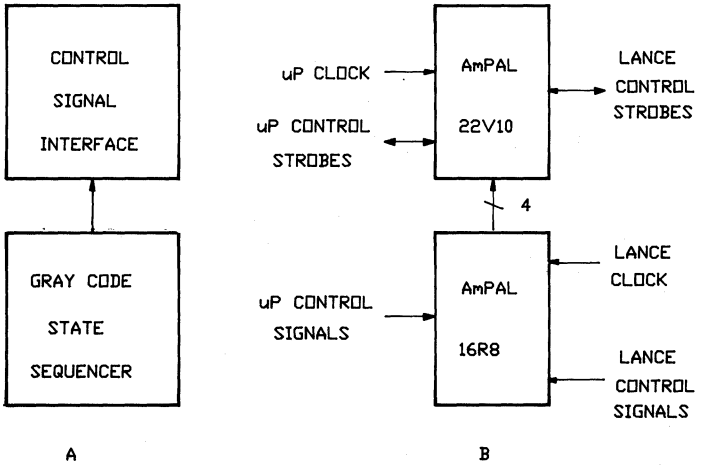
Figure 2-1. Conceptual Block Diagram

2.2 DMA CONTROL SEQUENCER

In order to completely comprehend the implementation of the coprocessor-like interface, it is crucial to understand the Direct Memory Access (DMA) Control Sequencer portion of the Control Bus Interface which converts each word operation into two separate byte transfers. This DMA interface is common to all three designs and consists of two sub-blocks (Figure 2-2a), a Control Signal Interface and a Gray Code State Sequencer, each realized with a PAL device (Figure 2.2b).

The Control Signal Interface consists of an AmPAL22V10 to match and convert the various control strobes required between the LANCE and each individual microprocessor.

The State Sequencer section uses an AmPAL16R8 to generate a four-bit Gray code state sequence (0,1,3,2,6,7, 5,4,C,D,F,E,A,B,9,8,0) to the AmPAL22V10 signal controller. A Gray code sequence was chosen to prevent glitches on the PAL device's gated control output terms during state machine transitions (ref. 1 pp. 364-371). Complete details for this important interface are given for each specific design.



07422A-2

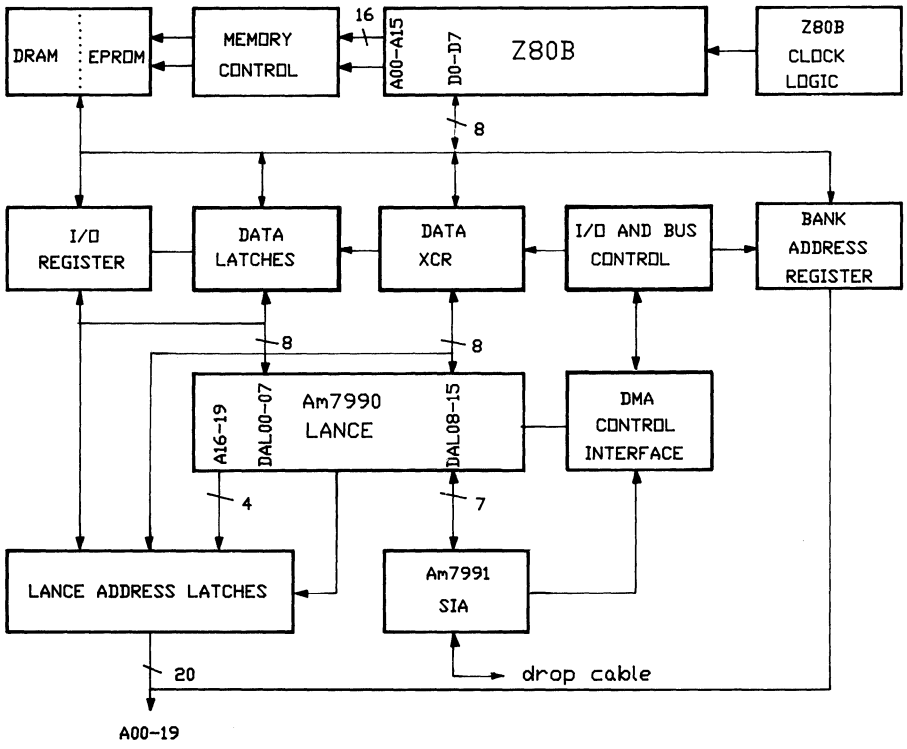
Figure 2-2. DMA Control Sequencer Block Diagrams

SECTION 3
Z80B INTERFACE

The Z80B to Am7990 design can be broken down into three major sections: address, data, and control. The majority of the components used in the interface are bus transceivers, latches, registers and PAL devices (Figure 3-1). Following this discussion, a few insights to memory requirements and boundary conditions are presented.

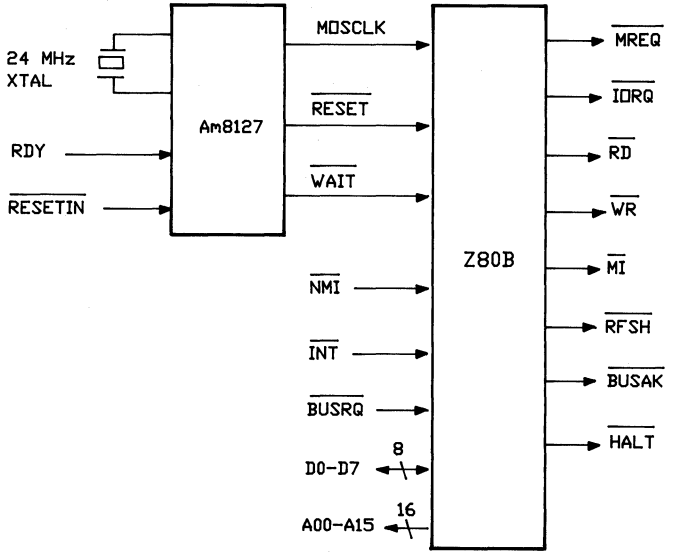
The Z80B is operated at its maximum clock frequency of 6MHz when a 24MHz crystal is connected to the Am8127 Clock Generator (Figure 3-2). The fast operating cycle time allows for minimum interrupt response latency and efficient packet/message handling. For simplicity of design, the Z80 interrupt mode 1 is used for this configuration. An Am9519A-1 Interrupt Controller can easily be added if a more complex interrupt vectoring structure is required.

The control bits in the LANCE's control status register 3 (CSR3) should be programmed to BCON = 1, BSWP = 0 and ACON = 0.



07422A-3

Figure 3-1. Z80B Interface Block Diagram



07422A-4

Figure 3-2. Z80B and Clock Generator

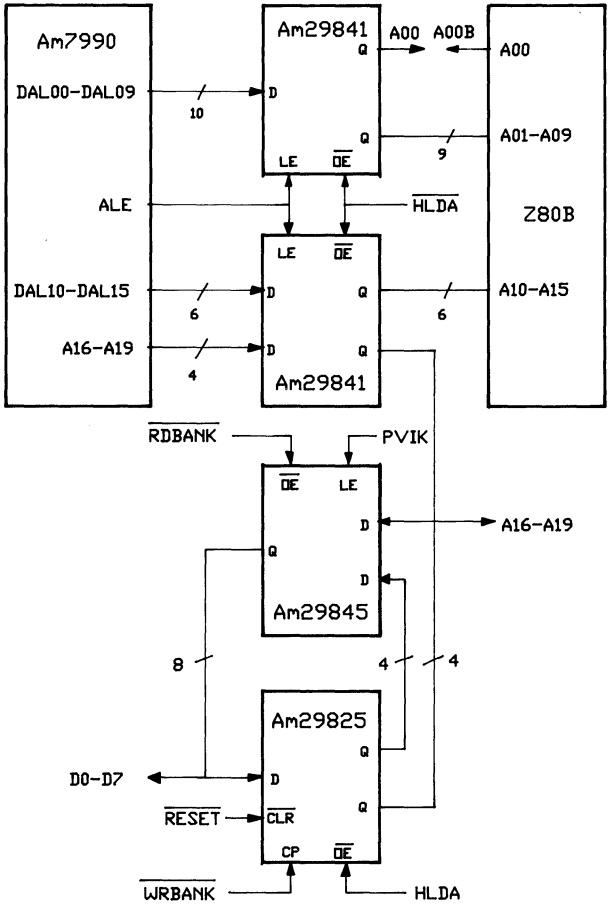
3.1 ADDRESS BUS INTERFACE

A major concern when using a Z80 processor is that the memory addressing range may be insufficient to accommodate the volume of code needed to support the upper-level Ethernet protocols and transmit/receive buffers. If this is the case, a memory-mapper can be added using an Am29705A Dual-Port RAM.

A simpler solution is bank switching using an I/O-mapped output port as an address bank register which contains the higher order bits needed to expand the memory address range. To accomplish this and keep the component count to a minimum, an Am29825 Octal Register is used instead of a 74LS273 and 74LS244 (Figure 3-3). The register clear allows the Z80B to start execution from a known state (Bank 0) after reset. The output enable makes the bank addresses go into a high-impedance state during LANCE DMA operations. The Am29845 provides a readback path for the bank addresses during bank switching operations.

Let's next examine the address generation for LANCE DMA operations. The LANCE multiplexes the lower order addresses (A00-A15) on the DAL lines during the early part of a data transfer cycle. In order to save these addresses, two Am29841 10-bit Latches are used to capture the addresses during ALE high time.

The four extra latches are used to hold the four high order addresses (A16-A19) in order to increase the drive capability of these address lines. The high and low order LANCE addresses are then combined with the bank and Z80B addresses on the system address bus. LANCE address bit 0 is not directly connected to the system address bus, but instead is first taken to the signal control PAL device (as seen in Figure 3.6) for byte/word polarity control, and then connected to the system address bus.



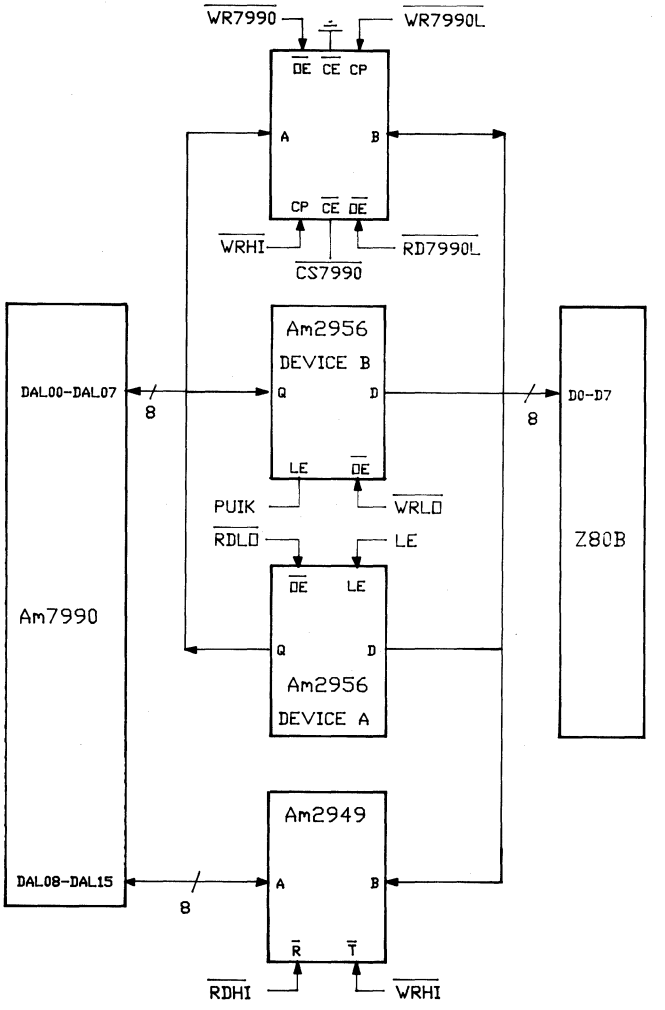
07422A-5

Figure 3-3. Z80B Address Bus Interface

3.2 DATA BUS INTERFACE

The data bus interface performs two major functions. It maps the LANCE's 16-bit data bus into the Z80B's 8-bit wide memory bus. It also acts as a byte funnel, collecting two bytes of data from the 8-bit memory bus and presenting them as a single word to the LANCE's word wide data bus. As a result of these two functions, the data bus interface can be viewed as two independent sections.

When the Z80B accesses the LANCE's CSRs, two devices, an Am2952 Bidirectional I/O Register and the Am2949 Bidirectional Bus Transceiver (Figure 3-4) are required. To the Z80B, the Am2952 appears to be an 8-bit I/O register which acts as a temporary low byte storage buffer during I/O operations to the LANCE. Z80B to LANCE operations are as follows:



07422A-6

Figure 3-4. Z80B Data Bus Interface

Write Operation

1. Write low byte of I/O word to the Am2952.
2. Write high byte of I/O word to LANCE via the Am2949.

Read Operations

1. Read high byte of I/O word from the LANCE via the Am2949.
2. Read low byte of I/O word from Am2952.

This low byte I/O register is a simple and effective method to prevent contamination of I/O data by DMA operations initiated between I/O byte transfers (remember that the Z80 has only byte I/O instructions).

During LANCE DMA transfers, two 8-bit memory data read operations are converted into, what appears to the LANCE as a single 16-bit data read. A single 16-bit LANCE write operation is converted into two 8-bit memory write transfers. The circuit consists of two Am2956 Octal Latches and the Am2949, sharing the first function (Figure 3-4). The functional operation is as follows:

LANCE Read Operation

1. Read and store low byte of data from memory into Am2956 device A.
2. Read high byte of data from memory and present both bytes to the LANCE to complete word read.

LANCE Write Operation

1. Write low byte of data to memory via Am2956 device B.
2. Write high byte of data to memory via Am2949 to complete write cycle.

DMA byte operations enable only the buffer corresponding to the byte of the word being transferred.

The control portion of the data bus interface consists of a single AmPAL22V10, (Figure 3-5b) which generates I/O chip select to the LANCE, I/O strobes for the LANCE's low byte I/O port, and buffer control signals necessary to enable the direction and state of the bus transceiver and latches. A second AmPAL22V10 (Figure 3-5a) generates the control strobes for the address bank register and four spare I/O chip selects. The remaining three outputs of this PAL device are used to synchronize the Z80B INT, NMI and BUSRQ control signals to prevent the occurrence of any potential metastable conditions. Refer to the respective PAL device equations for additional information.

3.3 CONTROL BUS INTERFACE

As mentioned in the overview, the control interface consists of two PAL devices, an AmPAL16R8 and AmPAL22V10. The two PAL devices provide state sequencing strobe conditioning and signal synchronization between the Z80B and the LANCE during both I/O and DMA operations (Figure 3-6).

The AmPAL16R8 generates a Gray code state sequence referenced to the LANCE's 10MHz TCLK and enabled only during a DMA operation when the DAS is active. State variables 1-4 (SQ1-SQ4) are sent to the AmPAL22V10 for strobe conditioning. State variable 0 (SQ0) is used not only to generate the other four state variables, but also to provide a convenient 5MHz 50% duty cycle clock. In addition, this device generates Hold Acknowledge synchronization to the LANCE and the address latches in order to minimize address bus contention.

The AmPAL22V10 signal controller generates all the strobes needed to perform and synchronize I/O and DMA transfers. During I/O operations, it generates LANCE DAS and READ signals from the RD and WR signals of the Z80B. Also, it enables the LANCE's output READY signal to the Z80B's WAIT signal by reclocking it through the Am8127. This synchronizes I/O data transfers.

The AmPAL22V10 really proves its versatility during LANCE DMA operations. The device not only generates the Z80B MREQ, RD, and WR memory control signals, but also controls the LANCE's operational cycle length and A00B polarity for byte/word transfers. Byte operations complete unaffected while word operations are stretched to accommodate for conversion to two squential byte transfers.

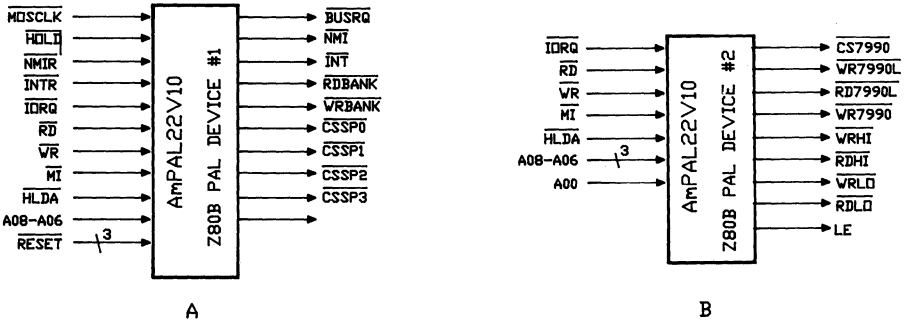
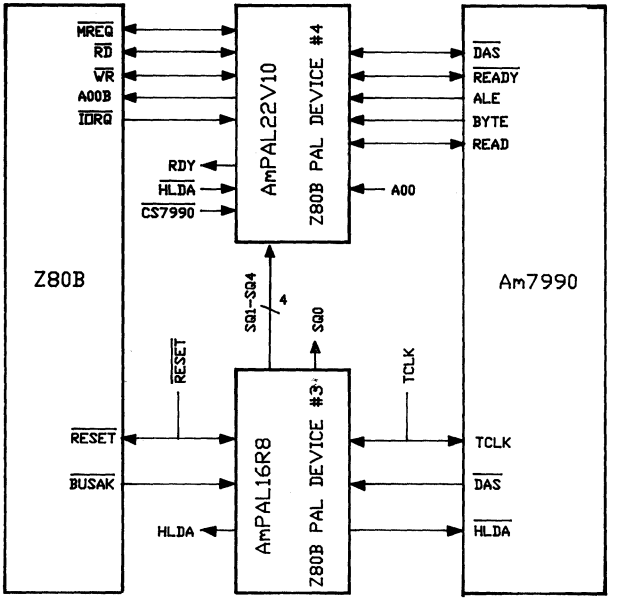


Figure 3-5. Z80B PAL Devices #1 and #2



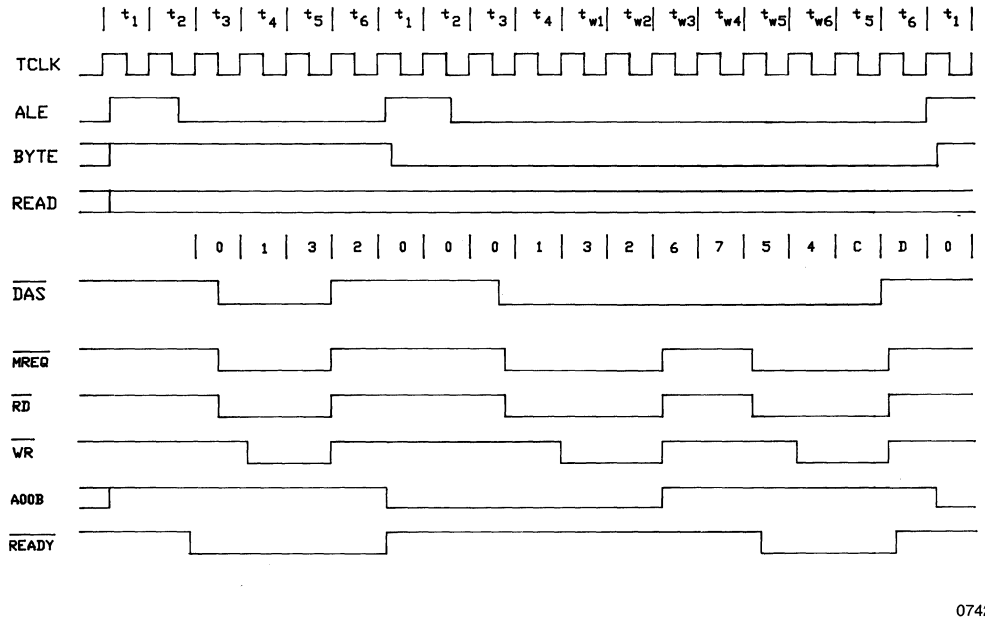
07422A-8

Figure 3-6. Z80B DMA Control Interface

A word transfer appears as two byte transfers. The first transfer cycle begins when DAS is asserted. This causes MREQ and RD to go active for a read cycle or MREQ and WR (during Gray code state 1) to go active for a write cycle. At the end of the first transfer cycle, the memory controls are deactivated and A00B is driven high. Continuing, states 6 and 7 provide an idle period (200ns) for dynamic RAMs to meet RAS precharge.

The second transfer cycle starts in state 5 with MREQ and RD or WR being asserted. The READY signal is also activated in preparation for termination of the current word transfer cycle.

At the end of the second transfer, DAS is negated causing READY, MREQ and RD or WR to be deasserted for the next operation cycle. The MREQ, RD, and WR strobes closely follow those of the Z80B, but these waveforms can be widened or tightened depending upon system memory requirements. Refer to the timing diagram (Figure 3-7) and the PAL device equations (Figures 3-8, 3-9, 3-10, and 3-11). These equations can be a bit tricky.



07422A-9

Figure 3-7. LANCE to Z80B Byte and Word Transfer Cycle Timing Diagram

```
AmPAL22V10
Z80B TO Am7990 INTERFACE PAL DEVICE #1
Z80 SYNCHRONIZATION INTERFACE AND DECODER PAL DEVICE
VERSION 1.0

MOSCLK /NMIR /INTR /HOLD A03 A07 A06 /RD /WR /IORQ /M1 GND
/RESET /HLDA /CSSP0 /CSSP1 /CSSP2 /CSSP3 /RDBANK /WRBANK /NMI /INT /BUSRQ VCC


RDBANK  = /HLDA*/M1*IORQ*A08*A07*/A06*RD        ;READ ADDRESS BANK REGISTER

WRBANK  = /HLDA*/M1*IORQ*A08*A07*/A06*WR        ;WRITE ADDRESS BANK REGISTER

CSSP0   = /HLDA*/M1*IORQ*A08*/A07*A06           ;SPARE I/O CHIP SELECT 0

CSSP1   = /HLDA*/M1*IORQ*A08*/A07*/A06          ;SPARE I/O CHIP SELECT 1

CSSP2   = /HLDA*/M1*IORQ*/A08*A07*A06           ;SPARE I/O CHIP SELECT 2

CSSP3   = /HLDA*/M1*IORQ*/A08*A07*/A06          ;SPARE I/O CHIP SELECT 3

NMI     := NMIR*/RESET                          ;NONMASK INT SYNCHRONIZATION

INT     := INTR*/RESET                          ;INTERRUPT SYNCHRONIZATION

BUSRQ   := HOLD*/RESET                          ;BUS REQUEST SYNCHRONIZATION


DESCRIPTION

GENERATES I/O CHIP SELECTS AND SYNCHRONIZES INPUT Z80B SIGNALS.
```

07422A-10

Figure 3-8. Z80B to Am7990 Interface PAL Device #1

```
AmPAL22V10
Z80B TO Am7990 INTERFACE PAL DEVICE #2
LANCE BUS CONTROL PAL DEVICE
VERSION 1.0

NC A08 A07 A06 A00 /RD /WR /IORQ /M1 /HLDA NC GND
NC NC /RDHI /WRHI /RDLO /WRLO LE /WR7990 /RD7990L /WR7990L /CS7990 VCC

WRHI    = /HLDA*/M1*IORQ*A08*A07*A06*A00*RD     ;I/O READ HIGH BYTE
        + HLDA*A00*WR                           ;DMA MEMORY WRITE HIGH BYTE

RDHI    = /HLDA*/M1*IORQ*A08*A07*A06*/A00*WR    ;I/O WRITE HIGH BYTE
        + HLDA*RD                               ;DMA MEMORY READ HIGH BYTE

WRLO    = HLDA*/A00*WR                          ;DMA MEMORY WRITE LOW BYTE

RDLO    = HLDA*RD                               ;DMA MEMORY READ LOW BYTE

/LE     = /HLDA + A00 + /RD                     ;DMA MEMORY READ LOW BYTE

CS7990  = /HLDA*/M1*IORQ*A08*A07*A06*A00        ;SELECT Am7990 LANCE

RD7990L = /HLDA*/M1*IORQ*A08*A07*A06*/A00*RD    ;READ Am7990 LOW I/O PORT

WR7990L = /HLDA*/M1*IORQ*A08*A07*A06*/A00*WR    ;WRITE Am7990 LOW I/O PORT

WR7990  = /HLDA*/M1*IORQ*A08*A07*A06*A00*WR     ;OUTPUT LOWER BYTE OF I/O PORT


DESCRIPTION

CONTROLS THE DATA PATH INTERFACE BETWEEN THE LANCE AND Z80B.
```

07422A-11

Figure 3-9. Z80B to Am7990 Interface PAL Device #2

```
AmPAL16R8
Z80B TO Am7990 INTERFACE PAL DEVICE #3
DMA GRAY CODE STATE SEQUENCER PAL DEVICE
VERSION 1.0

TCLK /DAS /RESET /BUSAK NC NC NC NC NC GND
GND HOLDA /HLDA NC /SQ4 /SQ3 /SQ2 /SQ1 /SQ0 VCC

SQ0     := /SQ0*DAS*BUSAK*/RESET                          ;GRAY CODE STATE 0

SQ1     := /SQ0*/SQ1*DAS*BUSAK*/RESET                     ;GRAY CODE STATE 1
         +  SQ0* SQ1*DAS*BUSAK*/RESET

SQ2     :=  SQ0* SQ1*/SQ2*DAS*BUSAK*/RESET                ;GRAY CODE STATE 2
         + /SQ0* SQ2*DAS*BUSAK*/RESET
         + /SQ1* SQ2*DAS*BUSAK*/RESET

SQ3     :=  SQ0*/SQ1* SQ2*/SQ3*DAS*BUSAK*/RESET           ;GRAY CODE STATE 3
         + /SQ0* SQ3*DAS*BUSAK*/RESET
         +  SQ1* SQ3*DAS*BUSAK*/RESET
         + /SQ2* SQ3*DAS*BUSAK*/RESET

SQ4     :=  SQ0*/SQ1*/SQ2*/SQ4*DAS*BUSAK*/RESET           ;GRAY CODE STATE 4
         + /SQ0* SQ4*DAS*BUSAK*/RESET
         +  SQ1* SQ4*DAS*BUSAK*/RESET
         +  SQ2* SQ4*DAS*BUSAK*/RESET

HLDA    := BUSAK*/RESET                                   ;HOLD ACK ACTIVE LOW

/HOLDA  := /BUSAK + RESET                                 ;HOLD ACK ACTIVE HIGH


DESCRIPTION

GENERATES GRAY CODE STATE SEQUENCES AND SYNCHRONIZES HOLD ACK.
```

07422A-12

Figure 3-10. Z80B to Am7990 Interface PAL Device #3

```
AmPAL22V10
Z80B TO Am7990 INTERFACE PAL DEVICE #4
SIGNAL CONDITIONER PAL DEVICE
VERSION 1.0

NC /IORQ /CS7990 A00 /HLDA NC BYTE /SQ4 /SQ3 /SQ2 /SQ1 GND
ALE /RD /WR /MREQ A00B RDY NC NC /READY /DAS READ VCC

IF (/HLDA)      READ    = RD                              ;I/O READ STATUS

IF (/HLDA)      DAS     = RD + WR                         ;I/O DATA STROBE

IF (HLDA)       READY   =  BYTE*/ALE                      ;DMA READY SIGNAL
                        + /BYTE*DAS*/SQ1*/SQ2*SQ3
                        + /BYTE*DAS*/SQ2*SQ3*/SQ4
                        + /BYTE*DAS*SQ4

                RDY     := MREQ                           ;CPU WAIT SIGNAL
                        + /CS7990*IORQ
                        +  CS7990*READY

IF (HLDA)       A00B    =  BYTE*A00                       ;DMA A00 POLARITY
                        + /BYTE*SQ3
                        + /BYTE*SQ4

IF (HLDA)       /MREQ   = /DAS                            ;DMA MREQ SIGNAL
                        + SQ2*SQ3*/SQ4

IF (HLDA)       /RD     = /READ                           ;DMA READ STROBE
                        + /DAS
                        + SQ2*SQ3*/SQ4

IF (HLDA)       WR      =  BYTE*/READ*DAS*SQ2             ;DMA WRITE STROBE
                        +  BYTE*/READ*DAS*SQ3
                        +  BYTE*/READ*DAS*SQ4
                        + /BYTE*/READ*DAS*SQ2*/SQ3*/SQ4
                        + /BYTE*/READ*DAS*/SQ1*/SQ2*SQ3
                        + /BYTE*/READ*DAS*/SQ2*SQ3*SQ4

DESCRIPTION

CONVERTS Am7990 CONTROL SIGNALS TO Z80B CONTROL SIGNALS.
```

07422A-13

Figure 3-11. Z80B to Am7990 Interface PAL Device #4

3.4 MEMORY CONSIDERATIONS

When designing the memory control interface, special considerations should be taken since there are two unrelated synchronous events occuring simultaneously. This is especially evident when attempting to design an interface for dynamic RAMs. This interface should be of an asynchronous nature. For the Z80B, this interface is not very complicated. It can be built with one AmPAL22V10 and a delay line (Figure 3-9).

An additional memory timing constraint, the dynamic RAM refresh cycle time, must be examined. To meet the worst case condition, the following inequality must hold:

$$2ms/128 \geq (16m+4) \times 100ns + (19+n) \times p$$

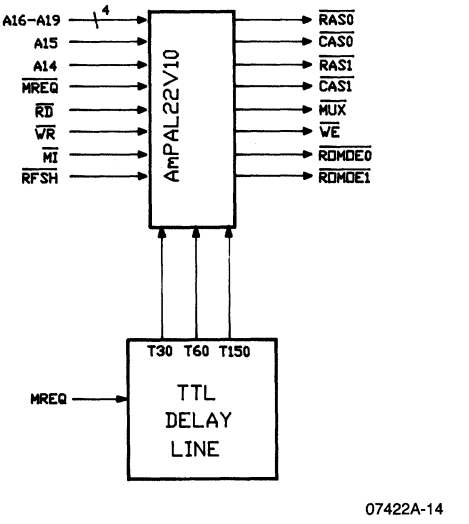
where:

- m equals the number of TCLKs per byte transfer cycle. The 4 refers to the 4 TCLKs for DMA synchronization,
- n equals the number of Z80B memory waits per EX (SP),HL instruction,
- p equals the Z80B clock period in nanoseconds.

For n = 0 and p = 165ns, this implies that $m \leq 7.5$

Therefore, each DMA byte transfer must be less than 8 TCLKs in order to provide sufficient dynamic RAM refresh. The design discussed uses only 6 TCLKs.

For n = 0 and p = 250ns, this implies that $m \leq 6.5$ which means that at 4Mhz, only 6 TCLKs are allowed.

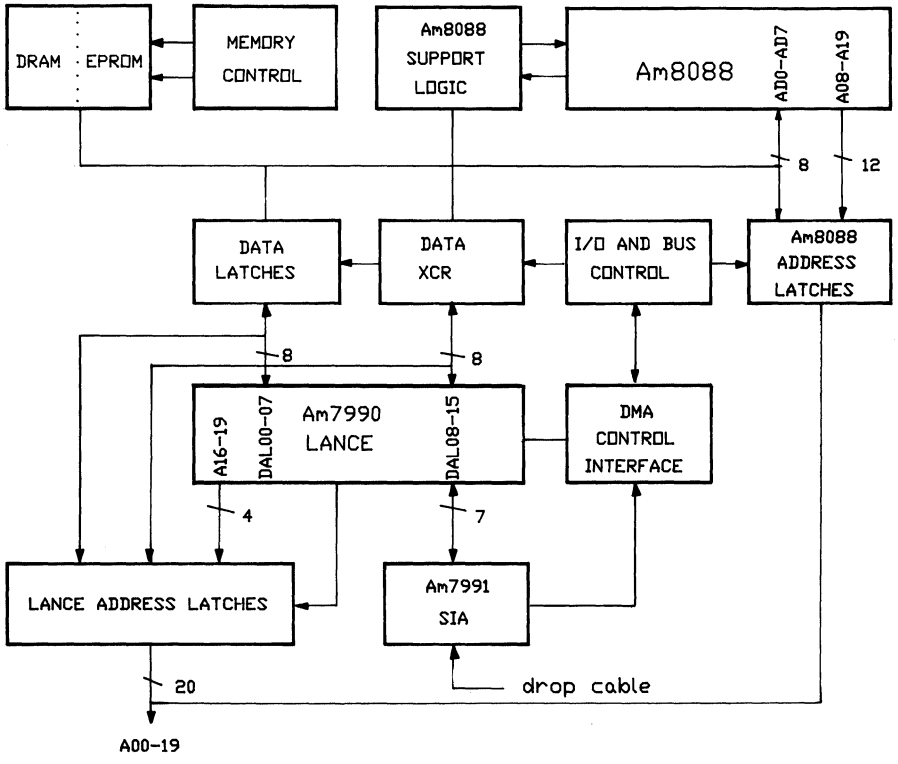


07422A-14

Figure 3-12. Z80B Memory Interface

SECTION 4
Am8088 INTERFACE

The Am8088 to Am7990 interface is the simplest of all three designs examined here. It can be divided into three major sections: address, data, and control. Most of the devices used in the interface are bus transceivers, latches, registers, and control PAL devices (Figure 4-1). A few helpful suggestions are given to aid in the design of the memory interface at the end of this section.

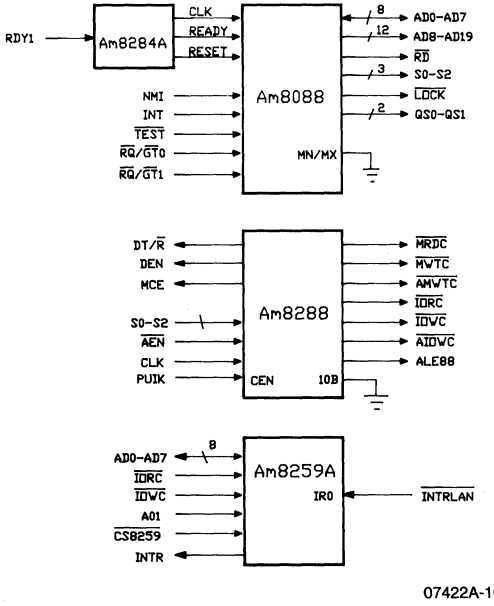


07422A-15

Figure 4-1. Am8088 Interface Block Diagram

The Am8088 is operated in MAX mode (Figure 4-2) and is surrounded by its usual interface devices: the Am8284A Clock Generator, the Am8288 Bus Controller, and the Am8259A Interrupt Controller (ref #3 for more system information). The Am8088's clock frequency can be either 5, 8, or 10 MHz depending on the particular Am8088 (i.e. -2 or -1) and peripheral devices chosen. The faster clock frequencies allow quicker interrupt response and more efficient packet/message/status handling. The Am8259A generates a vector to an interrupt handling routine after receiving and acknowledging an interrupt request from the LANCE.

The control bits in the LANCE's control status register 3 (CSR3) should be programmed to BCON = 1, BSWP = 0 and ACON = 0.



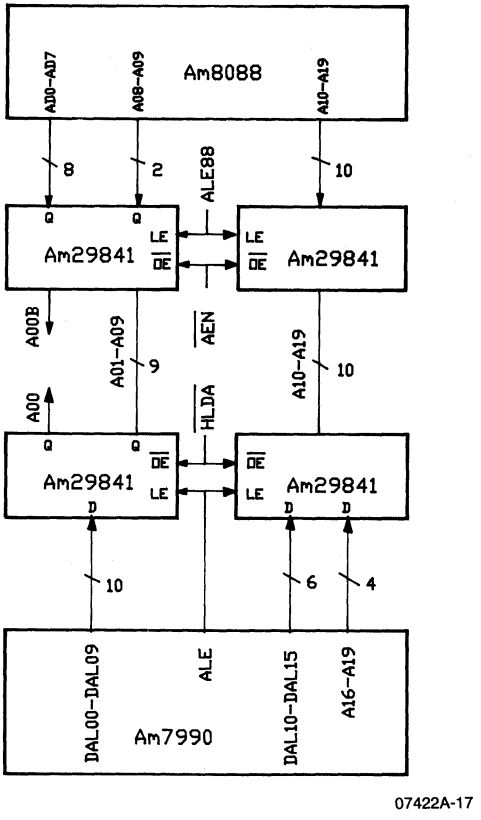
07422A-16

Figure 4-2. Am8088 and Support Logic

4.1 ADDRESS BUS INTERFACE

The Am8088, unlike the Z80 and MC68008, time multiplexes the lower eight bits of address (A00-A07) on the address/data (AD) lines, and multiplexes the high order four bits of address (A16-A19) with control status information during the first part of its instruction and data cycles. In order to retain these addresses and lower the device count, two Am29841 10-bit latches are used to hold all 20 bits of Am8088 address (A00-A19) when the Am8088 ALE is active (Figure 4-3).

Let's now examine address generation for LANCE DMA operations. The LANCE multiplexes the lower order addresses (A00-A15) on the DAL lines during the early part of a data transfer cycle. In order to save these addresses, another set of two Am29841's are used to capture the addresses during



07422A-17

Figure 4-3. Am8088 Address Bus Interface

LANCE ALE high time. The four extra latches are used to store the four high order addresses (A16-A19) in order to improve the drive capability of these address lines. The high and low order LANCE addresses are then connected to the demultiplexed Am8088 address lines on the system address bus. LANCE address bit 0 is not directly connected to the system address bus, but instead is first taken to the signal control PAL devices (Figure 4-6) for byte/word polarity control and then connected to the system address bus. The output enable controls on each set of latches permit the correct set of addresses to be present on the system address bus at the appropriate time.

4.2 DATA BUS INTERFACE

The data bus interface performs two major functions. It maps the LANCE's 16-bit data bus into the Am8088's 8-bit wide memory bus, and it acts as a byte funnel collecting two bytes of data from the 8-bit memory bus and presenting it as a single word to the LANCE's word wide data bus. With the aid of the LOCK instruction prefix and word I/O instructions, it is possible to combine both functions into the same set of buffers/latches and eliminate the necessity for a low byte I/O port.

Am8088 accesses to the LANCE's CSR's are carried out by three devices, an Am8287 bus transceiver and two Am8282 octal latches (Figure 4-4). To the Am8088, these three devices appear to be a single word I/O port with the Am8287 connected to the high byte of the word and the two Am8282's to the low byte. Am8088 to LANCE I/O operations are as follows:

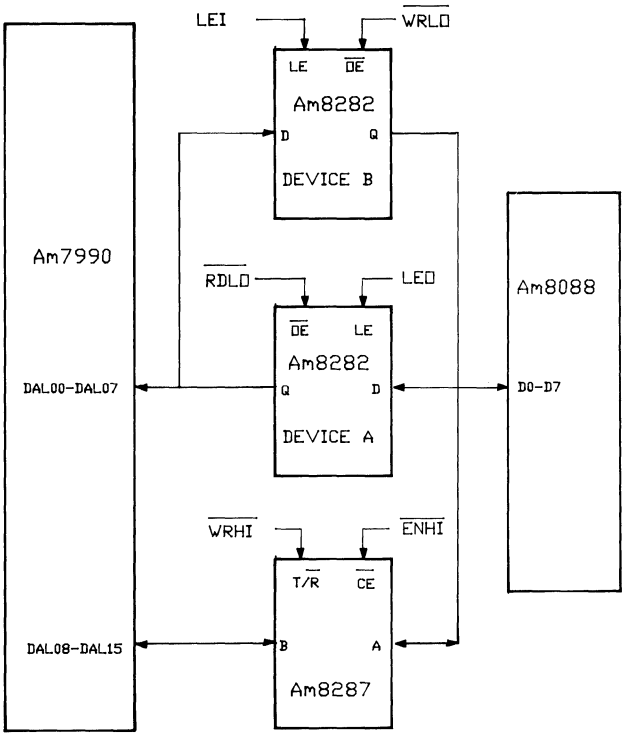
Write Operation

1. Locked I/O word write to even Am7990 I/O address.

Read Operation

1. Locked I/O word read from odd Am7990 I/O address.

The Locked I/O instruction provides mutually exclusive access to the transceiver and latches for both I/O and DMA operations, thereby, preventing any destructive data contamination. The circuit used for I/O accesses is also used for DMA operations.



07422A-18

Figure 4-4. Am8088 Data Bus Interface

During LANCE DMA transfers, two 8-bit memory data read operations are converted into, what appears to the LANCE as a single 16-bit data read operation. A 16-bit LANCE write operation is converted into two 8-bit memory write transfers. The functional sequence is as follows:

LANCE Read Operation

1. Read and store low byte of data from memory into Am8282 device A.
2. Read high byte of data from memory and present both bytes to the LANCE to complete word read.

LANCE Write Operation

1. Write low byte of data to memory via Am8282 device B.
2. Write high byte of data to memory via Am8287 to complete write cycle.

DMA byte operations enable only the buffer corresponding to the byte of the word being transferred.

The control portion of the data bus interface consists of a single AmPAL22V10. This PAL device (Figure 4-5b) generates I/O Chip Select to both the LANCE and the Am8259, and provides the I/O and DMA bus buffer control signals necessary to enable the direction and state of the bus transceiver and latches. Refer to the PAL device equations for detailed information.

4.3 CONTROL BUS INTERFACE

As described in the overview, the control interface consists of two PAL devices, an AmPAL16R8 and AmPAL22V10. The two PAL devices provide state sequencing strobe conditioning and signal synchronization between the Am8088 and the LANCE during both I/O and DMA operations (Figure 4-6).

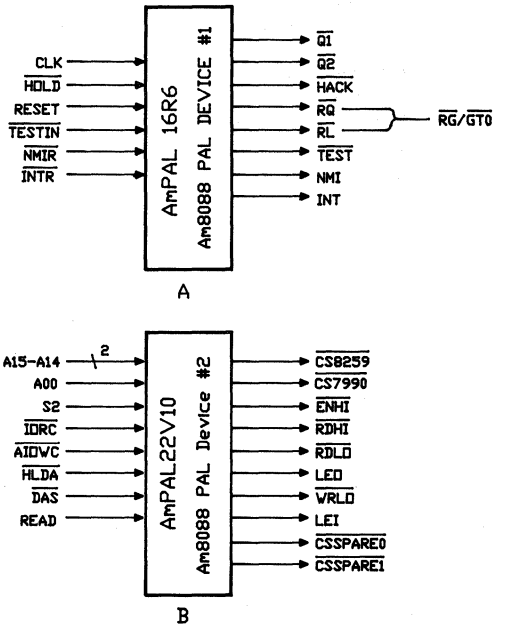


Figure 4-5. Am8088 PAL Devices #1 and #2

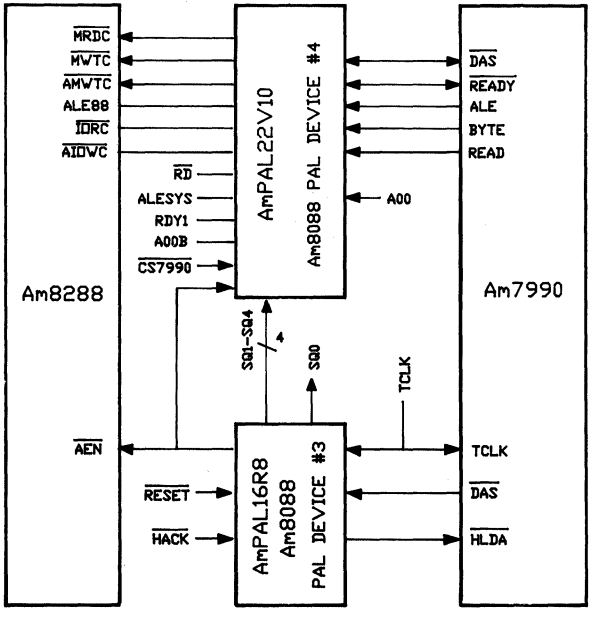


Figure 4-6. Am8288 DMA Control Interface

Since the LANCE data interface resides on the Am8088's local bus, it is necessary to produce the request/grant (RQ/GT) protocol in order to gain control of the local data and address busses. To accomplish this task, a HOLD/HLDA to RQ/GT convertor was implemented in an AmPAL16R6 (Figure 4-5a). This PAL device takes HOLD from the LANCE and, with a few registers and outputs, generates the Am8088's RQ/GT line (with two outputs RQ and RL). In addition, this PAL device synchronizes the TEST, INT, and NMI signals to meet the setup and hold times required by the Am8088. (See the RQ/GT timing diagram (Figure 4-7) and PAL device equations for more details.)

The AmPAL16R8 generates a Gray code state sequence referenced to the LANCE's 10MHz TCLK and enabled only during a DMA operation when the DAS is active. State variables 1-4 (SQ1-SQ4) are sent to AmPAL22V10 for strobe conditioning. State variable 0 (SQ0) is used to generate the other four state variables, and also provides a convenient 5MHz 50% duty cycle clock. In addition, this device generates hold acknowledge synchronization to the LANCE and the address latches to minimize address bus contention.

The AmPAL22V10 signal controller generates all the strobes needed to perform and synchronize I/O and DMA transfers. During I/O operations, it generates LANCE DAS and READ signals from the IORC and AIOWC signals of the Am8288. It also converts the LANCE's READY signal to the Am8088's READY signal via reclocking through the Am8284A in order to synchronize I/O data transfers.

The AmPAL22V10 really proves its versatility during LANCE DMA operations. The device not only generates the Am8088 RD signal and Am8288 MRDC, MWTC, and AMWTC memory control signals, but also controls the LANCE's operational cycle length and A00B polarity for byte/word transfers. Byte operations proceed unaffected while word operations are stretched to accommodate for conversion to two sequential byte transfers.

A word transfer appears as two byte transfers. The first transfer cycle begins when DAS is asserted. This causes MRDC and RD to go active for a read cycle or AMWTC and MWTC to go active for a write cycle. At the end of the first transfer cycle, the memory controls are deactivated and A00B is driven high. Continuing, states 6 and 7 provide an idle period (200ns) to meet the RAS precharge requirement of dynamic RAMs.

The second transfer cycle starts in state 5 with MRDC and RD or AMWTC and MWTC being asserted. The READY signal is also activated in preparation for termination of the current word transfer cycle. At the end of the second transfer, DAS is negated which causes READY, MRDC, and RD or AMWTC and MWTC to be deasserted for the next operation cycle. The MRDC, RD, AMWTC, and MWTC strobes closely follow those of the Am8088 and Am8288, but these waveforms can be modified depending upon system memory timing requirements. Refer to the DMA timing diagram (Figure 4-8) and the PAL device equations (Figures 4-9, 4-10, 4-11, and 4-12). The equations can be a bit difficult to understand at first.

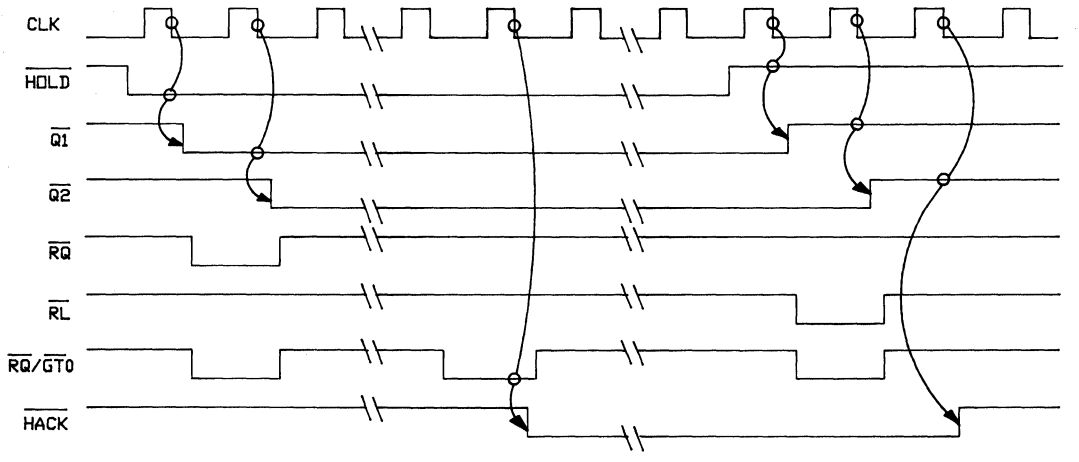
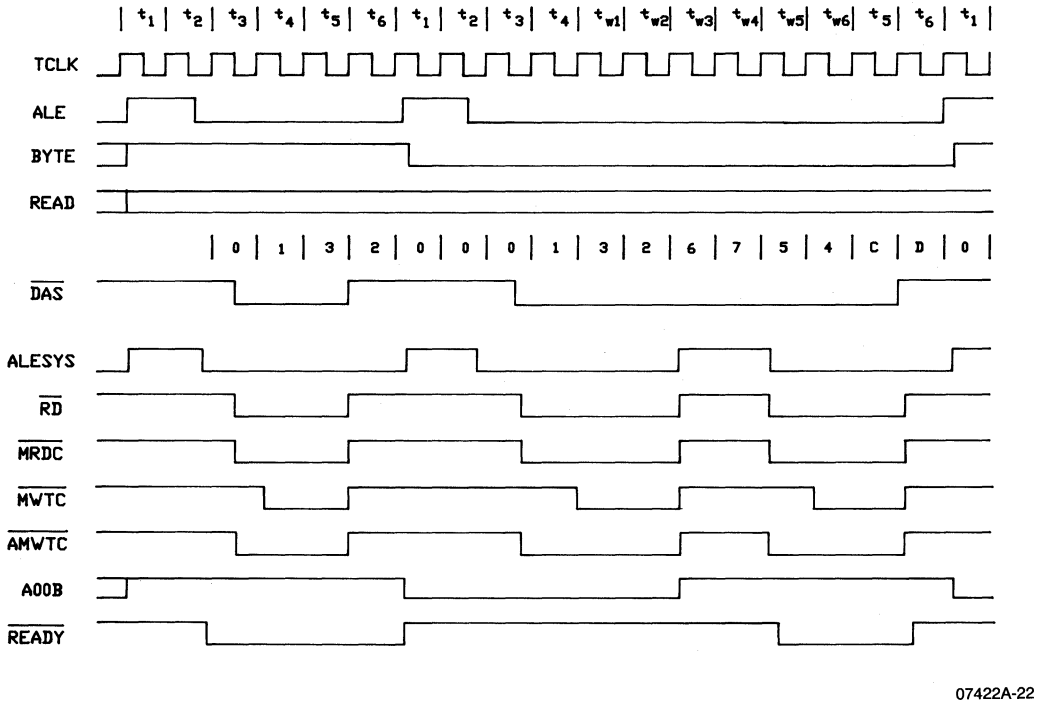


Figure 4-7. $\overline{RQ}$/ $\overline{GT}$— $\overline{HOLD}$/ $\overline{HACK}$ Converter Timing Diagram



07422A-22

Figure 4-8. LANCE to Am8088 Byte and Word Transfer Timing Diagram

```
AmPAL16R6
Am8088 TO Am7990 INTERFACE PAL DEVICE #1
DMA ARBITRATION AND SYNCHRONIZATION PAL DEVICE
VERSION 1.1

/CLK RESET /NMIR INTR /HOLD /TESTIN NC NC NC GND
GND /RQ /HACK /Q1 /Q2 NMI INT /TEST /RL VCC

TEST    := TESTIN*/RESET                        ;TEST INPUT SYNCHRONIZATION

/NMI    := /NMIR + RESET                        ;NONMASK INT SYNCHRONIZATION

/INT    := /INTR + RESET                        ;INTERRUPT SYNCHRONIZATION

Q1      := HOLD*/RESET                          ;HOLD STATE 1

Q2      := Q1*/RESET                            ;HOLD STATE 2

HACK    := Q2*HACK*/RESET                       ;HOLD ACK GENERATION
         + Q2*RQ*RL*/RESET

IF (Q1*/Q2*/RESET)  RQ = Q1*/Q2*/RESET          ;DMA REQUEST OUTPUT

IF (/Q1*Q2*/RESET)  RL = /Q1*Q2*/RESET          ;DMA RELEASE OUTPUT


DESCRIPTION

CONVERTS HOLD/HOLD ACK TO REQUEST/GRANT DMA REQUEST.  ALSO SYNCHRONIZES
INTERRUPT AND TEST INPUT SIGNALS TO THE Am8033.
```

07422A-23

Figure 4-9. Am8088 to Am7990 Interface PAL Device #1

```
AmPAL22V10
Am8088 TO Am7990 INTERFACE PAL DEVICE #2
PERIPHERAL CHIP SELECT DECODER AND LANCE DATA BUS CONTROL PAL DEVICE
VERSION 1.0

NC A15 A14 A00 /HLDA READ /DAS /IORC /AIOWC S2 NC GND
NC LE1 /WRLO LE0 /RDLO /ENHI /WRHI /CS7990 /CS8259 /CSSPARE1 /CSSPARE0 VCC

CS8259  = /HLDA*/S2*/A15*A14                    ;Am8259 CHIP SELECT

CS7990  = /HLDA*/S2*/A15*/A14*A00               ;Am7990 CHIP SELECT

WRHI    = /HLDA*/S2*/A15*/A14*A00*IORC          ;I/O HIGH BYTE READ
        +  HLDA*/READ                           ;DMA HIGH BYTE WRITE

ENHI    = /HLDA*/S2*/A15*/A14*A00*IORC          ;I/O HIGH BYTE READ
        + /HLDA*/S2*/A15*/A14*A00*AIOWC         ;I/O HIGH BYTE WRITE
        +  HLDA*DAS*A00                         ;DMA HIGH BYTE TRANSFER

RDLO    = /HLDA*/S2*/A15*/A14*A00*AIOWC         ;I/O LOW BYTE WRITE
        +  HLDA*READ*DAS                        ;DMA LOW BYTE READ

WRLO    = /HLDA*/S2*/A15*/A14*/A00*IORC         ;I/O LOW BYTE READ
        +  HLDA*/READ*DAS*/A00                  ;DMA LOW BYTE WRITE

LE0     = /HLDA*/S2*/A15*/A14*/A00*AIOWC        ;I/O LOW BYTE READ
        +  HLDA*READ*DAS*/A00                   ;DMA LOW BYTE WRITE

LE1     = /HLDA*/S2*/A15*/A14*A00*IORC          ;I/O LOW BYTE READ
        +  HLDA*/READ*DAS*/A00                  ;DMA LOW BYTE WRITE

CSSPARE0= /HLDA*/S2*A15*/A14                    ;SPARE I/O CHIP SELECT 0

CSSPARE1= /HLDA*/S2*A15*A14                     ;SPARE I/O CHIP SELECT 1


DESCRIPTION

GENERATES PERIPHERAL CHIP SELECTS AND CONTROLS LANCE DATA BUS INTERFACE
BUFFERS.
```

07422A-24

Figure 4-10. Am8088 to Am7990 Interface PAL Device #2

```
AmPAL16R8
Am8088 TO Am7990 INTERFACE PAL DEVICE #3
DMA GRAY CODE STATE SEQUENCER PAL DEVICE
VERSION 1.0

TCLK /DAS RESET /HACK NC NC NC NC NC GND
GND /AEN /HLDA NC /SQ4 /SQ3 /SQ2 /SQ1 /SQ0 VCC

SQ0     := /SQ0*DAS*HACK*/RESET                          ;GRAY CODE STATE 0

SQ1     := /SQ0*/SQ1*DAS*HACK*/RESET                     ;GRAY CODE STATE 1
         +  SQ0* SQ1*DAS*HACK*/RESET

SQ2     :=  SQ0* SQ1*/SQ2*DAS*HACK*/RESET                ;GRAY CODE STATE 2
         + /SQ0* SQ2*DAS*HACK*/RESET
         + /SQ1* SQ2*DAS*HACK*/RESET

SQ3     :=  SQ0*/SQ1* SQ2*/SQ3*DAS*HACK*/RESET           ;GRAY CODE STATE 3
         + /SQ0* SQ3*DAS*HACK*/RESET
         +  SQ1* SQ3*DAS*HACK*/RESET
         + /SQ2* SQ3*DAS*HACK*/RESET

SQ4     :=  SQ0*/SQ1*/SQ2*/SQ4*DAS*HACK*/RESET           ;GRAY CODE STATE 4
         + /SQ0* SQ4*DAS*HACK*/RESET
         +  SQ1* SQ4*DAS*HACK*/RESET
         +  SQ2* SQ4*DAS*HACK*/RESET

HLDA    :=  HACK*/RESET                                  ;HOLD ACK ACTIVE LOW

AEN     := /HACK + RESET                                 ;HOLD ACK ACTIVE HIGH


DESCRIPTION

GENERATES GRAY CODE STATE SEQUENCES AND SYNCHRONIZES HOLD ACK.
```

07422A-25

Figure 4-11. Am8088 to Am7990 Interface PAL Device #3

```
AmPAL22V10
Am8038 TO Am7990 INTERFACE PAL DEVICE #4
SIGNAL CONDITIONER PAL DEVICE
VERSION 1.0

ALE88 /IORC /CS7990 A00 /AEN /AIOWC BYTE /SQ4 /SQ3 /SQ2 /SQ1 GND
ALE /RD /MWTC /MRDC A00B RDY1 /AMWTC ALESYS /READY /DAS READ VCC

IF (AEN)        READ     = RD                                ;I/O READ STATUS

IF (AEN)        DAS      = IORC + AIOWC                      ;I/O DATA STROBE

IF (/AEN)       READY    =  BYTE*/ALE                        ;DMA READY SIGNAL
                         + /BYTE*DAS*/SQ1*/SQ2*SQ3
                         + /BYTE*DAS*/SQ2*SQ3*/SQ4
                         + /BYTE*DAS*SQ4

IF (CS7990)     RDY1     = READY                             ;I/O WAIT SIGNAL

IF (/AEN)       A00B     =  BYTE*A00                         ;DMA A00 POLARITY
                         + /BYTE*SQ3
                         + /BYTE*SQ4

IF (/AEN)       /MRDC    = /READ                             ;DMA MEMORY READ
                         + /DAS                              ;STROBE
                         + SQ2*SQ3*/SQ4

IF (/AEN)       /RD      = /READ                             ;DMA READ STROBE
                         + /DAS
                         + SQ2*SQ3*/SQ4

IF (/AEN)       /MWTC    = READ                              ;DMA MEMORY WRITE
                         + /DAS                              ;STROBE
                         + SQ2*SQ3*/SQ4

IF (/AEN)       AMWTC    =  BYTE*/READ*DAS*SQ2               ;DMA ADVANCED MEMORY
                         +  BYTE*/READ*DAS*SQ3               ;WRITE STROBE
                         +  BYTE*/READ*DAS*SQ4
                         + /BYTE*/READ*DAS*SQ2*/SQ3*/SQ4
                         + /BYTE*/READ*DAS*/SQ1*/SQ2*SQ3
                         + /BYTE*/READ*DAS*/SQ2*SQ3*SQ4

                ALESYS   = ALE                               ;ADRESS LATCH ENABLE
                         + ALE88                             ;GENERATION
                         + SQ2*SQ3*/SQ4


DESCRIPTION

CONVERTS Am7990 CONTROL SIGNALS TO Am8083 CONTROL SIGNALS.
```

07422A-26

Figure 4-12. Am8088 to Am7990 Interface PAL Device #4

4.4 MEMORY CONSIDERATIONS

When considering the memory architecture for this LANCE system, it is important to remember why the Am8088 processor was chosen. An overriding concern was to keep the memory cost low by using an 8-bit memory data path. Keeping this in mind, it is quite obvious that an odd number (usually one) of each type of byte-wide memory devices would be prudent. In most small or cost-sensitive systems, a single EPROM and a single bank of dynamic RAM is the minimum achievable.

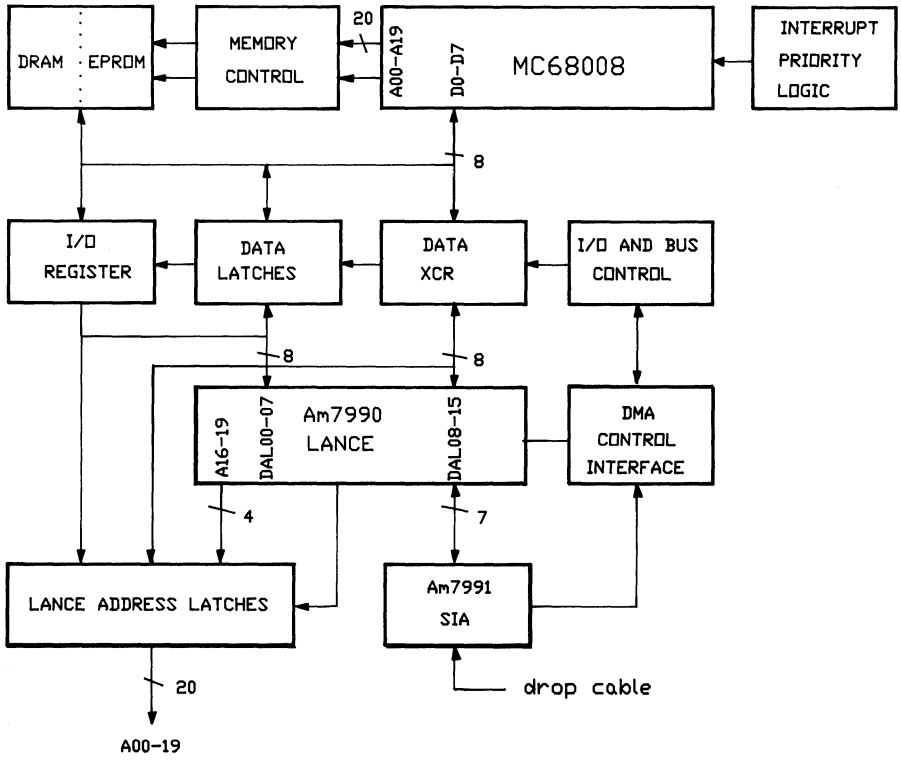
The designer should use the higher density ROMs or EPROMS (i.e. Am27128, Am27256, or Am27512) in order to provide an upgrade path for more complex software functions and to permit the faster Am8088's to run without memory waits. For dynamic RAMs, a memory bank (64K x 8 bits) of eight (or nine) 64K x 1 DRAMS could be implemented with the Am2968/69/70 family of dynamic RAM controllers. If additional buffer space is required for increased packet/message buffering, the memory is quadrupled by substituting 256Kx1 DRAMs for the 64Kx1 DRAMs Reference suggested memory organization diagram. Here are three suggestions for dynamic RAM refresh:

1. The processor periodically accessing memory locations initiated by a Non-Maskable Interrupt.
2. Using an external DMA controller (i.e. an Am9517A) to periodically access and refresh memory (as done on the IBM PC).
3. Using a stand alone dynamic RAM controller. Modifying the state sequence to match its particualr requirements.

Whichever choice is made, the interface described is flexible enough to be tailored to fit the interface constraints.

SECTION 5
MC68008 INTERFACE

The MC68008 to Am7990 design is very similar to the Z80B interface described earlier and is also partitioned into three major sections: address, data, and control. Most of the components used in the interface are bus transceivers, latches, registers and PAL devices (Figure 5-1). The end of this Section makes a few suggestions in regards to fast memory requirements.

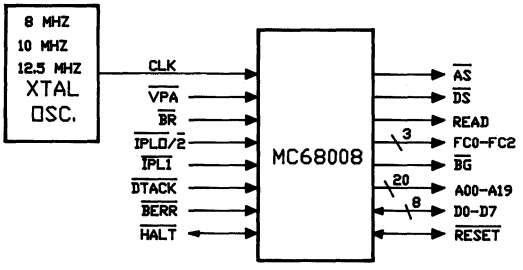


07422A-27

Figure 5-1. MC68008 Interface Block Diagram

The MC68008 can operate at 8, 10, or 12.5 MHz clock frequencies depending on the speed selection of the MC68008 and the external crystal oscillator used (Figure 5-2). The simplest and most economical method is to operate the MC68008 at 10MHz using the LANCE's TCLK as reference. Higher clock rates allow for minimum interrupt response latency and more efficient packet/message/status handling. There should not be any difficulties with interrupt response time even with an 8MHz device. LANCE is not dependent on interrupt response time for packet reception or transmission.

The control bits in the LANCE's control status register 3 (CSR3) should be programmed to BCON=0, BSWP=1 and ACON=0.



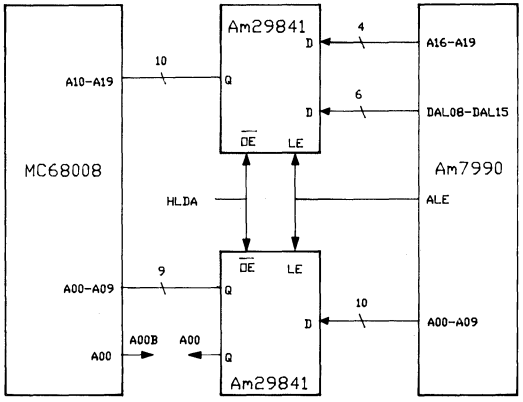
07422A-28

Figure 5-2. MC68008 and Clock

5.1 ADDRESS BUS INTERFACE

The MC68008 has the simplest address interface because it does not multiplex its address. Therefore, all 20-bits of address can be taken directly to the system address bus.

Now, let's examine the address generation for LANCE DMA operations. The LANCE multiplexes the lower order addresses (A00-A15) on the DAL lines during the early part of a data transfer cycle. In order to save these addresses, two Am29841 10-bit Latches are used to capture the addresses during ALE High time (Figure 5-3). The four extra latches are used to hold the four high order addresses (A16-A19) in order to enhance the drive capability of these address lines. The high and low order LANCE addresses are then connected to the MC68008 addresses on the system address bus. LANCE address bit 0 is not directly connected to the system address bus, but instead is first taken to the signal control PAL device (Figure 5-6) for byte/word polarity control and then connected to the system address bus.



07422A-29

Figure 5-3. MC68008 Address Bus Interface

5.2 DATA BUS INTERFACE

The data bus interface is similar to the Z80's but has the buffer assignments reversed because of the difference in definition of high and low bytes. It performs two functions. It convert the LANCE's 16-bit data bus into the MC68008's 8-bit wide memory bus and it acts as a byte funnel; collecting two bytes of data from the 8-bit memory bus and presenting them as a single word to the LANCE's word wide data bus. As a result of these two functions, the data bus interface can be viewed as two independent sections.

The operation of the first section is active when the MC68008 accesses the LANCE's CSRs. This section consists of two devices, an Am2952 Bidirectional I/O Register and the Am2947 Bidirectional Bus Transceiver (Figure 5-4). To the MC68008, the Am2952 appears as an 8-bit I/O register which acts as a temporary high byte storage buffer during transfer to the LANCE. MC68008 to LANCE operations are as follows:

Write Operation

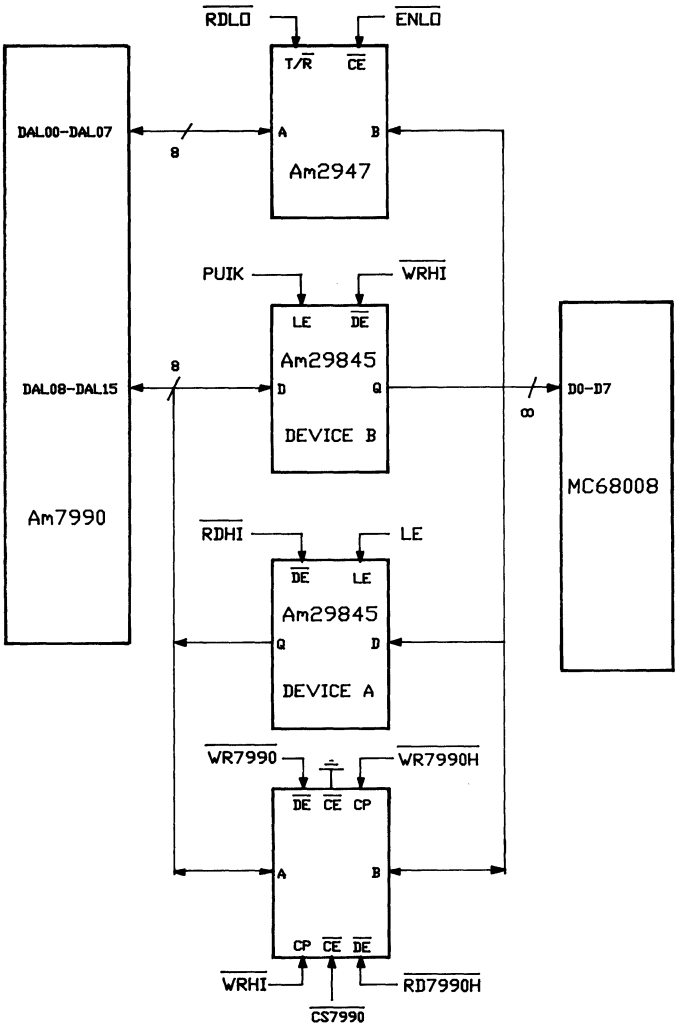
1. Write high byte (A0=0) of word to the Am2952.
2. Write low byte (A0=1) of word to LANCE via the Am2947.

Read Operations

1. Read low byte (A0=1) of word from the LANCE via the Am2947.
2. Read high byte (A0=0) of word from Am2952.

This high byte register is a simple and effective method to prevent contamination of data by DMA operations initiated between byte transfers since the MC68008 has no capabilities to prevent operations between word transfer instructions.

The second section is operational during LANCE DMA transfers. Its purpose is to convert two 8-bit memory data read operations into what appears to the LANCE as a single 16-bit data read operation, and to convert a 16-bit LANCE write operation into two 8-bit memory write transfers. The circuit consists



07422A-30

Figure 5-4. MC68008 Data Bus Interface

of two Am29845 Octal Latches, and the Am2947, sharing the first function, (Figure 5-4). The functional sequence is as follows:

LANCE Read Operation

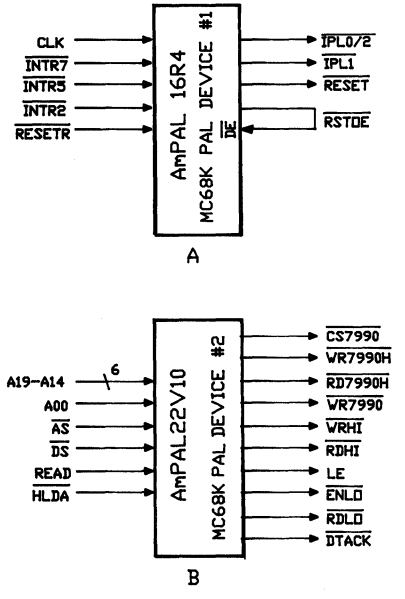
1. Read and store high byte (A0=0) of data from memory into Am29845 device A.
2. Read low byte (A0=1) of data from memory and present both bytes to the LANCE to complete word read.

LANCE Write Operation

1. Write high byte (A0=0) of data to memory via Am29845 device B.
2. Write low byte (A0=1) of data to memory via Am2947 to complete write cycle.

DMA byte operations enable only the buffer corresponding to the byte of the word being transferred.

The control portion of the data bus interface consists of a single AmPAL22V10, (Figure 5-5b) which generates I/O Chip Select to the LANCE, I/O strobes for the LANCE's low byte I/O port, and buffer control signals necessary to enable the direction and state of the bus transceiver and latches. In addition, this PAL device signals a data acknowledge (DTACK) for reading and writing of the high byte I/O port. Refer to the respective AmPAL equations for additional information.



07422A-31

Figure 5-5. MC68008 PAL Devices #1 and #2

5.3 CONTROL BUS INTERFACE

An AmPAL16R4 converts the interrupt requests for the LANCE from the prioritized interrupt protocol used by the MC68008 (Figure 5-5a). Additionally, this PAL device synchronizes and controls the RESET output to the MC68008 and the remainder of the LANCE system. See the PAL device equations for implementation details.

As discussed in the overview, the control interface consists of two PAL devices, an AmPAL16R8 and AmPAL22V10. The two PAL devices provide state sequencing strobe conditioning and signal synchronization between the MC68008 and the LANCE during both processor and DMA operations (Figure 5-6).

The AmPAL16R8 generates a Gray code state sequence referenced to the LANCE's 10MHz TCLK and enabled only during a DMA operation when the DAS is active. State variables 1-4 (SQ1-SQ4) are sent to the AmPAL22V10 for strobe conditioning. State variable 0 (SQ0) is used to generate the other four state variables, and also provides a handy 5MHz 50% duty cycle clock. In addition, hold acknowledge synchronization to the LANCE and to the address latches is generated in this device in order to minimize address bus contention.

The AmPAL22V10 signal controller generates all the strobes needed to perform and synchronize I/O and DMA transfers. During I/O operations, it generates LANCE DAS signal from the DS signal of the MC68008. Also, its reclocks the LANCE's output READY signal to the MC68008's DTACK signal in order to synchronize data transfers.

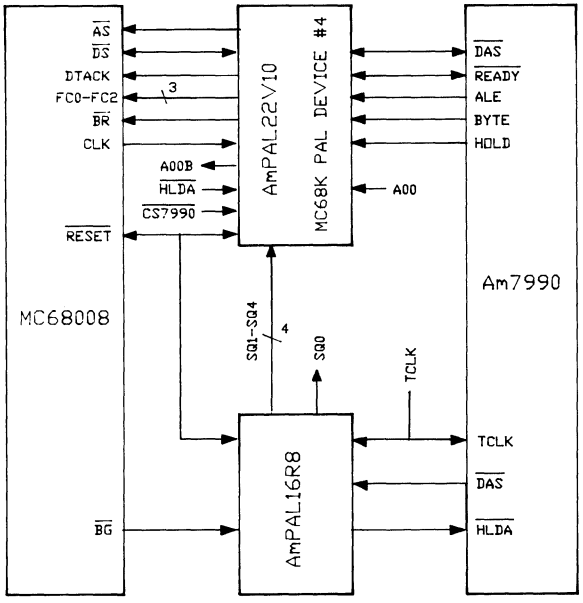
The AmPAL22V10 really proves its capabilities during LANCE DMA operations. It first synchronizes the LANCE's HOLD to generate the MC68008's BR signal. Then the device not only generates the MC68008 AS, DS, and FC0-FC2 memory control signals, but also controls the LANCE's operational cycle length and A00B polarity for byte/word transfers. Byte operations complete undisturbed while word operations are stretched to accommodate the conversion to two consecutive byte transfers.

A word transfer appears as two byte transfers. The first transfer cycle begins when DAS is asserted. This causes DS to go active for a read cycle or a write cycle. READ is directly driven to the bus by the LANCE. At the end of the first transfer cycle, AS and DS are deactivated and A00B is driven high. Continuing, states 6 and 7 provide an idle period (200ns) for dynamic RAMs to meet RAS precharge.

The second transfer cycle starts in state 5 with AS and DS being asserted. The READY signal is also activated in preparation for terminating the current word transfer cycle. At the end of the second transfer, DAS is negated causing READY and DS to be deasserted for the next operation cycle. AS and DS waveforms can be customized to different system memory requirements. Refer to the the timing diagram (Figure 5-7) and PAL device equations (Figures 5-8, 5-9, 5-10, and 5-11).

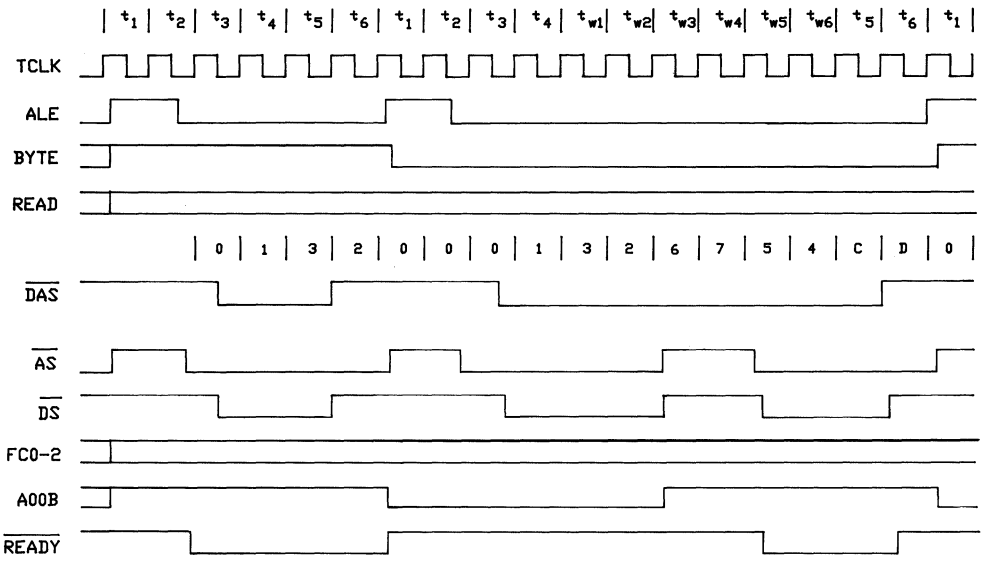
5.4 MEMORY CONSIDERATIONS

Most suggestions are similar to those discussed in the Am8088 Section, so refer to that Section. The only exception is the memory and I/O mapping. The MC68008 has no I/O-mapped peripheral capabilities, so all I/O peripherals must be memory-mapped. (See suggested memory map diagram.) One last detail, the high speed MC68008s must use the faster, higher density EPROMs to run without memory waits and thereby achieve maximum performance.



07422A-32

Figure 5-6. MC68008 DMA Control Interface



07422A-33

Figure 5-7. LANCE to MC68008 Byte and Word Transfer Timing Diagram

```
AmPAL16R4
MC68008 TO Am7990 INTERFACE PAL DEVICE #1
INTERRUPT PRIORITY ENCODER AND RESET SYNCHRONIZER PAL DEVICE
VERSION 1.0

CLK /RESETR /INTR2 /INTR5 /INTR7 NC NC NC NC GND
/OERST /RSTOE NC /RESET NC NC NC /IPL02 /IPL1 VCC

RESET   := RESETR                       ;RESET OUTPUT

RSTOE    = RESETR                       ;RESET OUTPUT CONTROL

IPL02    = INTR7 + INTR5                ;INTERRUPT PRIORITY LEVEL 0 AND 2

IPL1     = INTR7 + /INTR5*INTR2         ;INTERRUPT PRIORITY LEVEL 1


DESCRIPTION

PRIORITIZES INTERRUPTS AND SYNCHRONIZES RESET TO MC68008.
```

07422A-34

Figure 5-8. MC68008 to Am7990 Interface PAL Device #1

```
AmPAL22V10
MC68008 TO Am7990 INTERFACE PAL DEVICE #2
PERIPHERAL CHIP SELECT DECODER AND LANCE DATA BUS CONTROL PAL DEVICE
VERSION 1.0

A19 A18 A17 A16 A15 A14 A00 /HLDA READ /DS /AS GND
NC /DTACK /WRHI LE0 /RDHI /ENLO /RDLO /CS7990 /WR7990 /RD7990H /WR7990H VCC

CS7990   = /HLDA*AS*/A19*/A18*A17*A16*/A15*/A14*A00      ;Am7990 CHIP SELECT

WR7990   = /HLDA*AS*/A19*/A18*A17*A16*/A15*/A14*A00
         * /READ*DS                                      ;LOWER BYTE OF I/O PORT

RD7990H  = /HLDA*AS*/A19*/A18*A17*A16*/A15*/A14*/A00
         * READ*DS                                       ;RD Am7990 HI I/O PORT

WR7990H  = /HLDA*AS*/A19*/A18*A17*A16*/A15*/A14*/A00
         * /READ*DS                                      ;WR Am7990 HI I/O PORT

RDLO     = /HLDA*AS*/A19*/A18*A17*A16*/A15*/A14*/READ    ;I/O LOW BYTE READ
         +  HLDA*READ                                    ;DMA LOW BYTE WRITE

ENLO     = /HLDA*AS*/A19*/A18*A17*A16*/A15*/A14*A00*DS   ;I/O LOW BYTE TRANSFER
         +  HLDA*DS*A00                                  ;DMA LOW BYTE TRANSFER

RDHI     = HLDA*READ*DS                                  ;DMA HIGH BYTE READ

WRHI     = HLDA*/READ*DS*/A00                            ;DMA HIGH BYTE WRITE

LE0      = HLDA*READ*DAS*/A00                            ;DMA HIGH BYTE WRITE

IF (/HLDA*AS*/A19*/A18*A17*A16*/A15*/A14*/A00) DTACK = DAS      ;DATA ACK STRBE


DESCRIPTION

GENERATES PERIPHERAL CHIP SELECTS AND CONTROLS LANCE DATA BUS INTERFACE
BUFFERS.
```

07422A-35

Figure 5-9. MC68008 to Am7990 Interface PAL Device #2

```
AmPAL16R8
MC68008 TO Am7990 INTERFACE PAL DEVICE #3
DMA GRAY CODE STATE SEQUENCER PAL DEVICE
VERSION 1.0

TCLK /DAS /RESET /BG NC NC NC NC NC GND
GND NC /HLDA NC /SQ4 /SQ3 /SQ2 /SQ1 /SQ0 VCC

SQ0     := /SQ0*DAS*BG*/RESET                          ;GRAY CODE STATE 0

SQ1     := /SQ0*/SQ1*DAS*BG*/RESET                     ;GRAY CODE STATE 1
         +  SQ0* SQ1*DAS*BG*/RESET

SQ2     :=  SQ0* SQ1*/SQ2*DAS*BG*/RESET                ;GRAY CODE STATE 2
         + /SQ0* SQ2*DAS*BG*/RESET
         + /SQ1* SQ2*DAS*BG*/RESET

SQ3     :=  SQ0*/SQ1* SQ2*/SQ3*DAS*BG*/RESET           ;GRAY CODE STATE 3
         + /SQ0* SQ3*DAS*BG*/RESET
         +  SQ1* SQ3*DAS*BG*/RESET
         + /SQ2* SQ3*DAS*BG*/RESET

SQ4     :=  SQ0*/SQ1*/SQ2*/SQ4*DAS*BG*/RESET           ;GRAY CODE STATE 4
         + /SQ0* SQ4*DAS*BG*/RESET
         +  SQ1* SQ4*DAS*BG*/RESET
         +  SQ2* SQ4*DAS*BG*/RESET

HLDA    := BG*/RESET                                   ;HOLD ACK ACTIVE LOW


DESCRIPTION

GENERATES GRAY CODE STATE SEQUENCES AND SYNCHRONIZES HOLD ACK.
```

07422A-36

Figure 5-10. MC68008 to Am7990 Interface PAL Device #3

```
AmPAL22V10
MC68008 TO Am7990 INTERFACE PAL DEVICE #4
SIGNAL CONDITIONER PAL DEVICE
VERSION 1.0

CLK /HOLD /CS7990 A00 /HLDA /RESET BYTE /SQ4 /SQ3 /SQ2 /SQ1 GND
ALE FC0 FC1 FC2 A00B /DTACK /DS /BR /READY /DAS /AS VCC

IF (/HLDA)      DAS      = DS                                ;I/O DATA STROBE

IF (HLDA)       READY    =  BYTE*/ALE                        ;DMA READY SIGNAL
                         + /BYTE*DAS*/SQ1*/SQ2*SQ3
                         + /BYTE*DAS*/SQ2*SQ3*/SQ4
                         + /BYTE*DAS*SQ4

IF (CS7990)     DTACK   := READY*/RESET                      ;I/O DATA ACK SIGNAL

IF (HLDA)       A00B     =  BYTE*A00                         ;DMA A00 POLARITY
                         + /BYTE*SQ3
                         + /BYTE*SQ4

IF (HLDA)       /DS      = /DAS + SQ2*SQ3*/SQ4               ;DMA DATA STROBE

IF (HLDA)       AS       = ALE + SQ2*SQ3*/SQ4                ;DMA ADDRESS STROBE

IF (HLDA)       FC0      = HLDA                              ;STATUS BIT 0

IF (HLDA)       FC1      = /HLDA                             ;STATUS BIT 1

IF (HLDA)       FC2      = /HLDA                             ;STATUS BIT 2

                BR      := HOLD*/RESET                       ;DMA BUS REQUEST


DESCRIPTION

CONVERTS Am7990 CONTROL SIGNALS TO MC68008 CONTROL SIGNALS.
```

07422A-37

Figure 5-11. MC68008 to Am7990 Interface PAL Device #4

SECTION 6
CONCLUSIONS

Three relatively straightforward designs were presented and discussed. Each interface started from a conceptual level design and each was in turn expanded to more detailed blocks progressively specifying more function and implementation. The design concepts defined are of sufficient scope to encompass a wide spectrum of LANCE systems. Additionally, the logic implementations are reasonably flexible and can be customized to accommodate nearly all 8-bit LANCE interfaces and designs.

SECTION 7
REFERENCES

1. Friedman and Menon, "Theory and Design of Switching Circuits," Computer Science Press, Inc., 1975
2. Advanced Micro Devices Analog and Communication Products Data Book, 1983
3. Advanced Micro Devices MOS Microprocessor and Peripherals Data Book, 1983
4. Advanced Micro Devices Bipolar Microprocessor and Interface Data Book, 1983
5. Advanced Micro Devices Programmable Array Logic Handbook, 1983
6. Advanced Micro Devices MOS and Bipolar Memory Data Book, 1983
7. Advanced Micro Devices Am7990/1 Technical Manual, November 1982
8. Advanced Micro Devices Local Area Network Controller Am7990 (LANCE) Technical Manual, 1985
9. Mostek MK68590 Technical Manual, November 1982
10. Motorola MC68008 Advanced Information Specification, November 1982
11. Zilog 1982/83 Data Book

The International Standard of Quality guarantees a 0.05% AQL on all electrical parameters, AC and DC, over the entire operating range.

ADVANCED MICRO DEVICES
DOMESTIC SALES OFFICES

ALABAMA	(205) 882-9122
ARIZONA,	
Tempe	(602) 242-4400
Tucson	(602) 792-1200
CALIFORNIA,	
El Segundo	(213) 640-3210
Newport Beach	(714) 752-6262
San Diego	(619) 560-7030
Sunnyvale	(408) 720-8811
Woodland Hills	(818) 992-4155
COLORADO	(303) 691-5100
CONNECTICUT,	
Southbury	(203) 264-7800
FLORIDA,	
Altamonte Springs	(305) 339-5022
Clearwater	(813) 530-9971
Ft Lauderdale	(305) 484-8600
Melbourne	(305) 254-2915
GEORGIA	(404) 449-7920
ILLINOIS	(312) 773-4422
INDIANA	(317) 244-7207
KANSAS	(913) 451-3115
MARYLAND	(301) 796-9310
MASSACHUSETTS	(617) 273-3970
MINNESOTA	(612) 938-0001
NEW JERSEY	(201) 299-0002
NEW YORK,	
Liverpool	(315) 457-5400
Poughkeepsie	(914) 471-8180
Woodbury	(516) 364-8020
NORTH CAROLINA,	
Charlotte	(704) 525-1875
Raleigh	(919) 847-8471
OREGON	(503) 245-0080
OHIO,	
Columbus	(614) 891-6455
PENNSYLVANIA,	
Allentown	(215) 398-8006
Willow Grove	(215) 657-3101
TEXAS,	
Austin	(512) 346-7830
Dallas	(214) 934-9099
Houston	(713) 785-9001
WASHINGTON	(206) 455-3600
WISCONSIN	(414) 782-7748

INTERNATIONAL SALES OFFICES

BELGIUM,		
Bruxelles	TEL:	(02) 771 99 93
	FAX:	762-3716
	TLX:	61028
CANADA, Ontario,		
Kanata	TEL:	(613) 592-0060
Willowdale	TEL:	(416) 224-5193
	FAX:	(416) 224-0056
FRANCE,		
Paris	TEL:	(01) 687.36.66
	FAX:	6862185
	TLX:	20253
GERMANY,		
Hannover area	TEL:	(05143) 50 55
	FAX:	5553
	TLX:	925287
München	TEL:	(089) 41 14-0
	FAX:	406490
	TLX:	523883
Stuttgart	TEL:	(0711) 62 33 77
	FAX:	625187
	TLX:	721882
HONG KONG,		
Kowloon	TEL:	3-695377
	FAX:	1234276
	TLX:	50426
ITALY, Milano	TEL:	(02) 3390541
	FAX:	3498000
	TLX:	315286
JAPAN, Tokyo	TEL:	(03) 345-8241
	FAX:	3425196
	TLX:	J24064 AMDTKOJ
LATIN AMERICA,		
Ft. Lauderdale	TEL:	(305) 484-8600
	FAX:	(305) 485-9736
SWEDEN, Stockholm	TEL:	(08) 733 03 50
	FAX:	7332285
	TLX:	11602
UNITED KINGDOM,		
Manchester area	TEL:	(0925) 828008
	FAX:	827693
	TLX:	628524
London area	TEL:	(04862) 22121
	FAX:	22179
	TLX:	859103

NORTH AMERICAN REPRESENTATIVES

CALIFORNIA	
I^2 INC	OEM (408) 988-3400
	DISTI (408) 498-6868
CONNECTICUT	
SCIENTIFIC COMPONENTS	(203) 272-2963
IDAHO	
INTERMOUNTAIN TECH MKGT	(208) 322-5022
INDIANA	
SAI MARKETING CORP	(317) 241-9276
IOWA	
LORENZ SALES	(319) 377-4666
MICHIGAN	
SAI MARKETING CORP	(313) 227-1786
NEBRASKA	
LORENZ SALES	(402) 475-4660
NEW JERSEY	
TAI CORPORATION	(609) 933-2600
NEW MEXICO	
THORSON DESERT STATES	(505) 293-8555
NEW YORK	
NYCOM, INC	(315) 437-8343
OHIO	
Dayton	
DOLFUSS ROOT & CO	(513) 433-6776
Strongsville	
DOLFUSS ROOT & CO	(216) 238-0300
PENNSYLVANIA	
DOLFUSS ROOT & CO	(412) 221-4420
UTAH	
R^2 MARKETING	(801) 595-0631

Advanced Micro Devices reserves the right to make changes in its product without notice in order to improve design or performance characteristics. The performance characteristics listed in this document are guaranteed by specific tests, guard banding, design and other practices common to the industry. For specific testing details, contact your local AMD sales representative. The company assumes no responsibility for the use of any circuits described herein.

ADVANCED MICRO DEVICES 901 Thompson Pl., P.O. Box 3453, Sunnyvale, CA 94088, USA
TEL: (408) 732-2400 • TWX: 910-339-9280 • TELEX: 34-6306 • TOLL FREE: (800) 5388450

© 1985 Advanced Micro Devices, Inc.
Printed in U.S.A. CBM-WCP-10M-12/85-0