NSBMC096™ Memory Controller for the i960CA—User Application Guide

National Semiconductor
Application Note 883
May 1993

TABLE OF CONTENTS

1.0 DISTINCTIVE FEATURES

The NSBMC096 is a memory controller designed specifically for use in i960CA based systems. The specific nature of such systems precludes the NSBMC096 from being a general purpose device. However, within the constraints of the targeted applications, it imposes few architectural restrictions. Every effort has been made to preserve application flexibility while still achieving the principal design objective of providing low cost/high performance glueless memory systems for i960CA Systems.

1.1 Impact on System Design

By supporting both the i960CA bus interface and fast page mode memory protocols, the NSBMC096 enables the system designer to incorporate very large memory sub-systems into high speed processing systems. The fact that little performance penalty is incurred is a key feature. In many application areas such as intelligent peripheral controllers, high speed communication controllers, test instrumentation and high performance work-stations, the use of large memory for code and/or data storage is not only desirable, but indispensable.

Given the low cost per bit of dynamic RAMs, total system cost can be maintained at levels approaching those of currently popular systems that offer performance rates lower by a factor of five or more. Cost/performance ratios of this magnitude are not possible if lower density static memories are substituted, or alternate implementations using power and space consumptive discrete devices are used.

1.2 i960CA Bus Protocol Support

The NSBMC096 implements the bus protocol described in the “i960CA User’s Manual”. Both basic and burst access methods are supported.

The NSBMC096 is functionally equivalent to the V96BMC.
NSBMC096™ is a trademark of National Semiconductor Corporation.
i960® is a registered trademark of Intel Corporation.

NSBMC096 Memory Controller for the i960CA—User Application Guide AN-883

TL/V/11804 RRD-B30M75/Printed in U. S. A.

The interconnect to the processor's bus interface is direct. The processor bus naming convention has been used in designating the pins related to this interface. In order to connect the NSBMC096 to the i960CA one simply connects like named signal pins together. By eliminating the requirement for intermediate control logic, the NSBMC096 memory solution avoids the problems associated with the use of general purpose memory controllers namely: low performance, larger board space requirement, higher heat dissipation and decreased reliability.

1.3 Memory Interface Support

The NSBMC096 supports DRAMs using the fast page mode protocol. By taking advantage of the fast page feature of the DRAMs and row comparison logic, the NSBMC096 is able to achieve static RAM performance from low cost, high density DRAMs. All outputs to the memory array have been designed with high current drive in order to avoid the necessity of external drivers. In addition, care has been taken during design to minimize problems associated with ground bounce and Simultaneously Switching Outputs.

1.4 Bus Buffer Strategies

Although the NSBMC096 does not provide data bus buffering "on-chip", it does generate the required control signals for the buffer components. The use of multi-mode buffer control signals provides the designer with the flexibility to select buffer components which are optimized for the desired cost/performance criteria.

2.0 PROCESSOR PROTOCOL SUPPORT

The three memory and peripheral access methods specified by the i960CA bus interface protocol are:

— Basic Access

— Pipelined

— Burst

The NSBMC096 supports both Basic and Burst access. Pipelined access on the i960CA is unaffected by the external READY input of the processor. Consequently, pipelined operation **cannot** be supported in DRAM systems which must dynamically insert wait states to insure that back to back DRAM access doesn't violate the RAS precharge time (for example). Wait states are dynamically inserted if the processor begins an access when a refresh is in progress.

2.1 Basic Access

The Basic Access method is a conventional processor to memory interface with a synchronous handshake. Using *Figure 2.1* as a reference, a basic access begins when an address strobe (ADS) is asserted with an address in the range of the NSBMC096 (PCLK 0). The access can begin on either the even (A2 = 0) memory bank (Bank A) or the odd (A2 = 1) bank (Bank B). If sufficient RAS precharge time has elapsed, RAS will be asserted on the appropriate bank immediately following the ADS cycle (following PCLK 0). On the same cycle the processor will assert the BLAST signal to indicate that the access is not a burst. Consequently, the NSBMC096 will not begin a RAS strobe on the opposing bank. Either ½ or one cycle following the assertion of RAS, the column address is driven onto the address lines of the selected bank. The time at which the address switches from Row to Column is programmable via bit 18 of the configuration register. After the column address has settled, the CAS strobes are asserted to the appropriate bytes of memory (controlled by the Byte enable outputs of the i960CA). The READY signal is asserted at PCLK 3 to indicate to the processor that data is ready and the current cycle is complete.

If an access request immediately follows a cycle in which RAS has been active, additional wait states will be inserted to guarantee RAS de-assertion for a minimum of 2 clocks. This ensures that the RAS pre-charge time is not violated. Consequently, the second access in *Figure 2.1* has an additional wait state inserted since the RAS strobe doesn't begin as early as it otherwise could.

Bit 20 (Cycle Time) of the configuration register can be used to control the overall access time allotted for RAS/CAS access. The operation depicted in *Figure 2.1* assumes bit 20 is

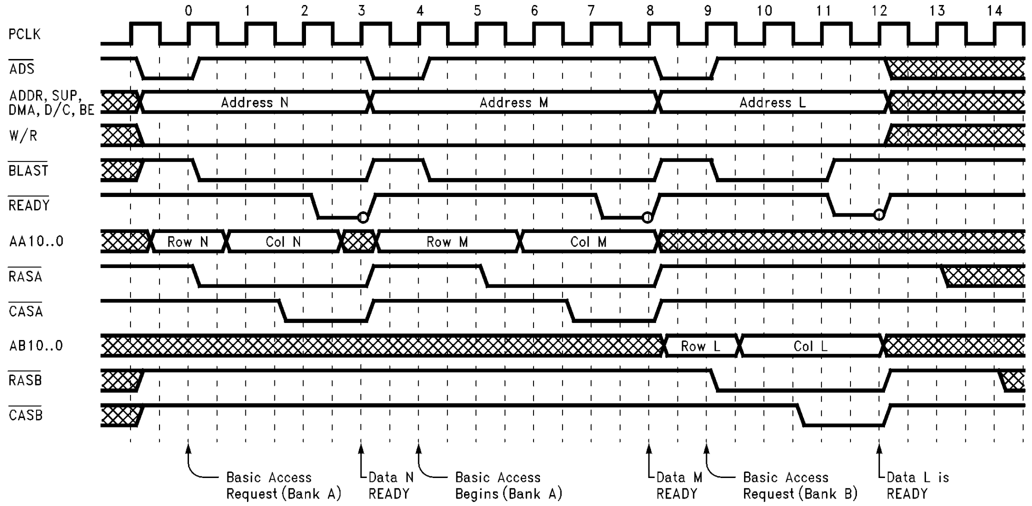


FIGURE 2.1. Basic Access (Minimum Cycle Mode)

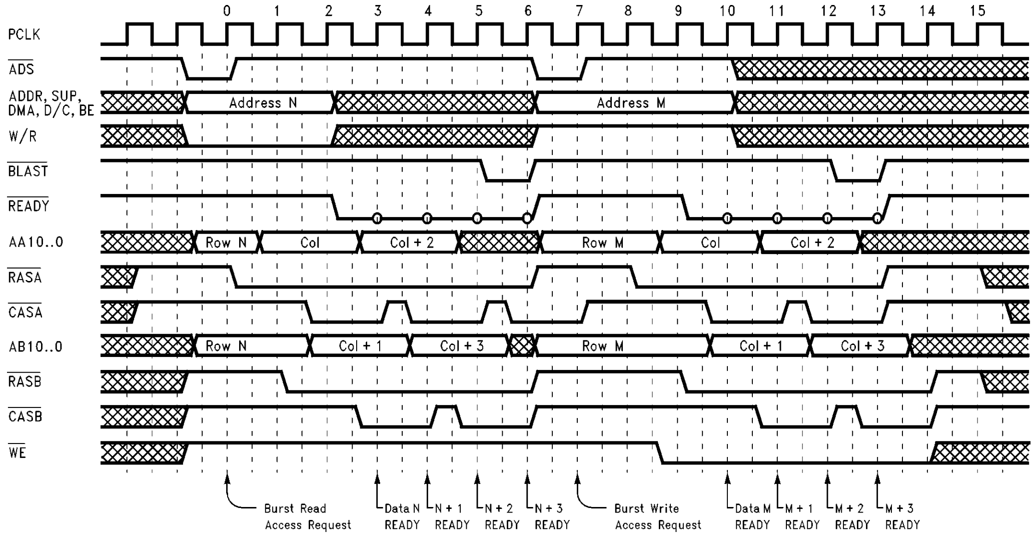
cleared resulting in a 3-cycle RAS to data time and a 1.5 cycle CAS to data time. When bit 20 is set (1), an additional clock cycle is inserted into RAS and CAS. For slower memory and/or higher clock speed, this option can be considered with the consequence of introducing more wait states. Section 2.5 describes this option in greater detail.

2.2 Burst Access

The burst access method of the i960CA bus protocol is crucial to achieving near theoretical performance. The NSBMC096 fully supports the slave device specification of this access method for both Read and Write access. Although the i960CA restricts bursts to 4 words, the NSBMC096 allows up to 2k bursts (512 words). High speed peripherals which have bus master capability could benefit from the longer bursts. While the NSBMC096 allows bursts to begin at any word boundary, the i960CA always begins a burst on an even word boundary (A2=0).

2.2.1 Interleaved Burst Read/Write Operation

A burst access begins in an identical manner to a basic access when an Address strobe is asserted with a valid address. Following the ADS cycle (PCLK 0 of *Figure 2.2.1*) RAS of the first bank is asserted (always RASA for the i960CA). Unlike the basic access, the BLAST signal remains de-asserted at PCLK 1 resulting in the assertion of RAS for the opposing Bank. Once a burst is established, the READY line remains asserted resulting in zero wait states for the remaining data. With the additional RAS/CAS cycle inserted when the cycle time bit (20) is set, bursts proceed with effectively ½ a wait state (Section 2.5). A burst read cycle always has an unused CAS cycle at the end (CASA following PCLK 5′). This is a consequence of the fact that BLAST is not guaranteed to be valid in time to stop the assertion of CAS. Obviously, data from this CAS strobe is not used.



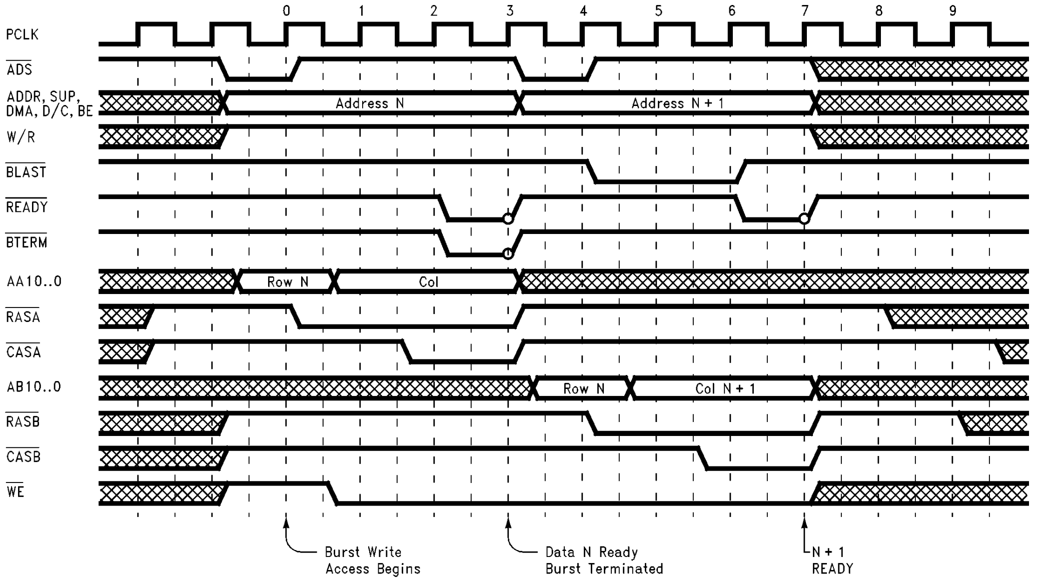
TL/V/11804–2

FIGURE 2.2.1. Burst Read Followed by Burst Write Access

2.2.2 Burst Write Disabled

Burst write should be disabled in higher speed systems using non-latching buffers. The purpose of disabling burst writes under these circumstances is to force the processor to hold the data on the bus long enough to meet the hold time requirements of the DRAM. For optimal performance, burst write should be enabled and latches should be used to hold the data (refer to Section 4.0).

When burst write is disabled (configuration bit 19=1), the NSBMC096 will return a BTERM with READY in response to a burst request (BLAST not asserted). This forces the i960 to re-issue the ADS signal and continue the sequential access as depicted in *Figure 2.2.2* below for a 2 word burst write. The burst write begins with an ADS at PCLK 0. With the absence of BLAST at PCLK 1, RASB would normally be asserted. However, a basic access is performed until PCLK 3 at which time a BTERM is asserted. This forces the processor to issue a new ADS at the second address of the sequence and a new basic access begins on Bank B.

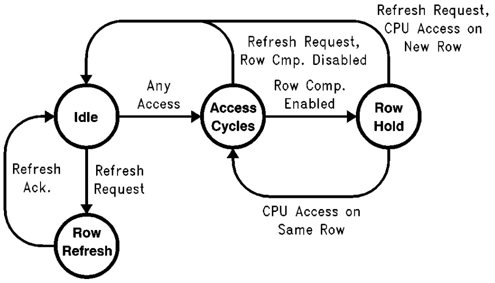


TL/V/11804–3

FIGURE 2.2.2. Burst Write Access with Burst Write Disabled

2.3 Row Compare Mode

To achieve maximum performance, the row comparison feature of the NSBMC096 should be enabled. Row comparison minimizes the time the memory system uses to strobe row addresses into the DRAM by performing a new RAS strobe only if the row changes from that of the preceding access. Consequently there is no delay caused by the RAS-precharge time, for back-to-back accesses within a row. The performance benefits realized are particularly noticeable in code which repetitively references memory within the 2k size. In this section the timing of the various access modes will be explained as they operate with row compare enabled.



TL/V/11804–4

FIGURE 2.3.1. NSBMC096 Simplified State Diagram

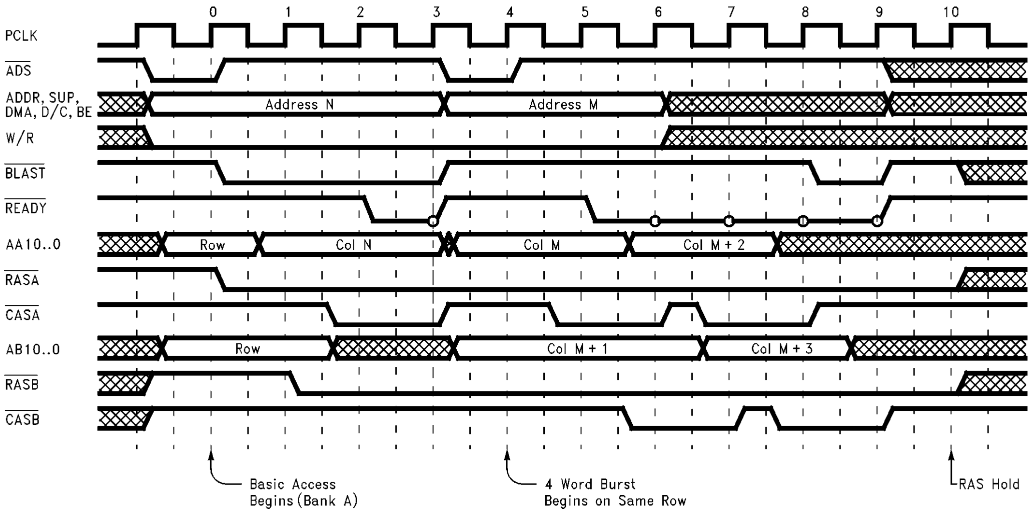
2.3.1 Overview

When row compare is enabled for Instruction and/or Data, The NSBMC096 can exist in one of 4 modes of operation as indicated in the above state diagram. The controller initially begins in the idle state wherein all Memory control signals are de-asserted. Any access to NSBMC096 controlled memory will initiate access cycles by strobing the appropriate RAS and CAS signals. With row compare enabled, the RAS signals remain asserted at the end of an access cycle. The RAS strobes will remain asserted until a refresh request is detected or an access is requested on a new row. If an access begins in the same row the NSBMC096 simply cycles between the "Access Cycles" state and the "Row Hold" state.

In the case that row compare mode is enabled for exclusively Instruction or Data the NSBMC096 will go directly to the "Idle" state or the "Row Hold" state depending on the access type decoded from the D/*C signal. For example, if instruction row compare is enabled and data row compare disabled, then the NSBMC096 will go directly to the "Idle" state following a data access. An instruction access, however, will result in the expected branch to the "Row Hold" state.

2.3.2 Row Hit

The timing diagram of *Figure 2.3.2* provides a more detailed explanation of the NSBMC096 when a "Row Hit" access is performed. Prior to PCLK 0 the controller is "Idle". Following PCLK 0 the controller asserts RASA for the requested access. Since row compare is enabled RASB is asserted at PCLK 1 in spite of the fact that no access to Bank B is required. On a row compare enabled access the NSBMC096 always strobes RAS for both banks of memory so that they will be synchronized for future access. When the burst access begins at PCLK 4, the controller issues the first CAS strobe on that cycle. Consequently, the first datum is ready at PCLK 6 (one wait state for the first access). If row compare was not enabled, both RAS signals would be de-asserted by PCLK 10. However, only an access to a new row or a refresh request will cause them to de-assert.

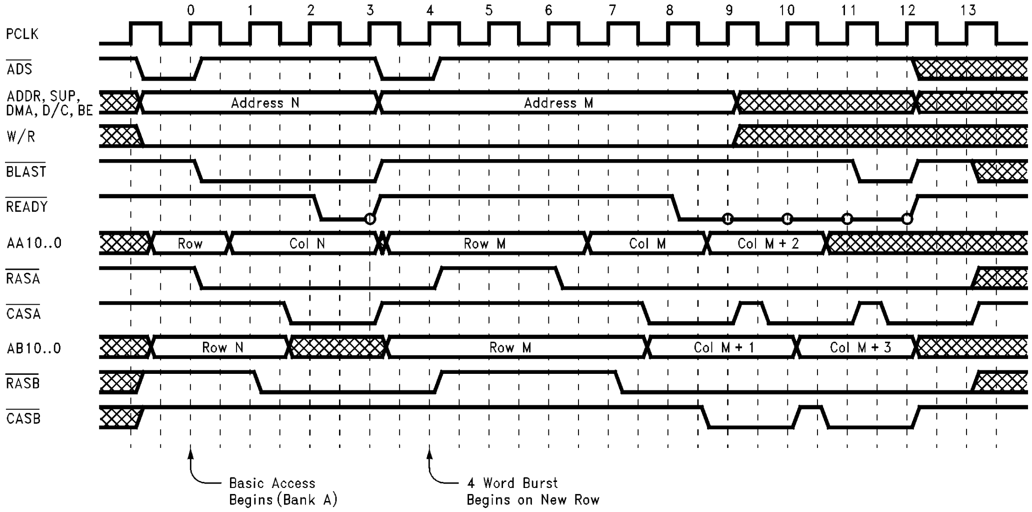


TL/V/11804–5

FIGURE 2.3.2. NSBMC096 with Row Compare Enabled: Row "Hit"

2.3.3 Row Miss

When an access begins on a new row, the controller will first return to idle mode before beginning the new RAS. It remains in idle state for 2 PCLK cycles so that the RAS precharge time required by DRAMs will be inserted. Consequently, a row miss results in additional wait states. *Figure 2.3.2* contrasts that of *2.3.1* by illustrating the row miss condition on the second access. The basic access at PCLK 0 proceeds as in *Figure 2.3.1.* However, address M (PCLK 4) is in a new row and the state changes to "Idle" before proceeding with the access cycles. With a new row in place, the DRAMs are ready for an access within row M at PCLK 13 ("Row Hold" state).

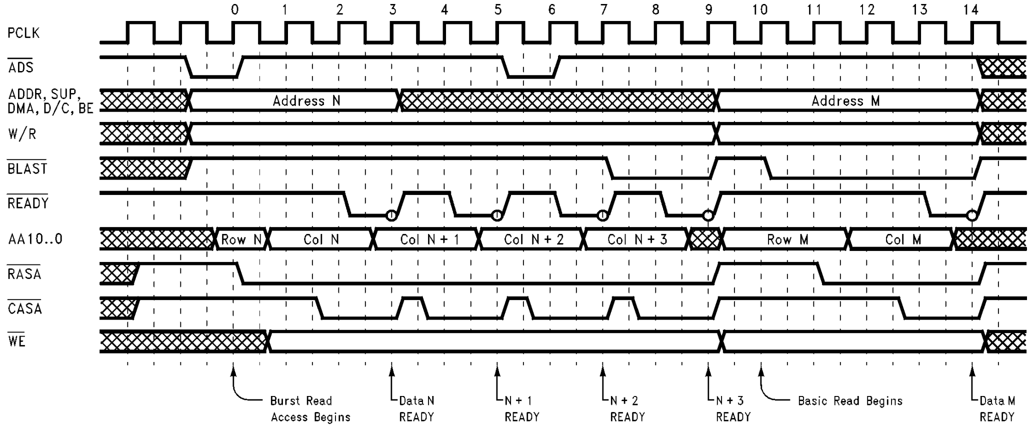


TL/V/11804-6

FIGURE 2.3.3. NSBMC096 with Row Compare Enabled: Row "Miss"

2.4 Non-Interleave Operation

The NSBMC096 can restrict access to a single bank of DRAM to achieve non-interleave operation if bit 17 of the configuration register is set to "1". By supporting non-interleave memory, the system integrator can assemble a processor card with only 8 memory devices. That same card can then be upgraded to a higher performance interleaved system by simply adding the additional memory devices to provide dual banks. Some restrictions apply to non-interleave operation. Only a single block of one bank can be populated since there is no advantage to expanding memory depth without going to interleaved operation. Burst sizes of up to 4 words are supported, whereas interleave mode supports 512 word bursts (2 kB). Row compare should not be enabled when non-interleave mode is active. As *Figure 2.4* indicates, non-interleave mode uses the Bank A side only. Consequently, the Bank B signals should not be connected. Note also that read and write cycles are identical with the exception of the Write Enable (WE) to the DRAM (in interleave mode the READY signal is asserted at the beginning of CAS for write cycles and at the end of CAS for read cycles).

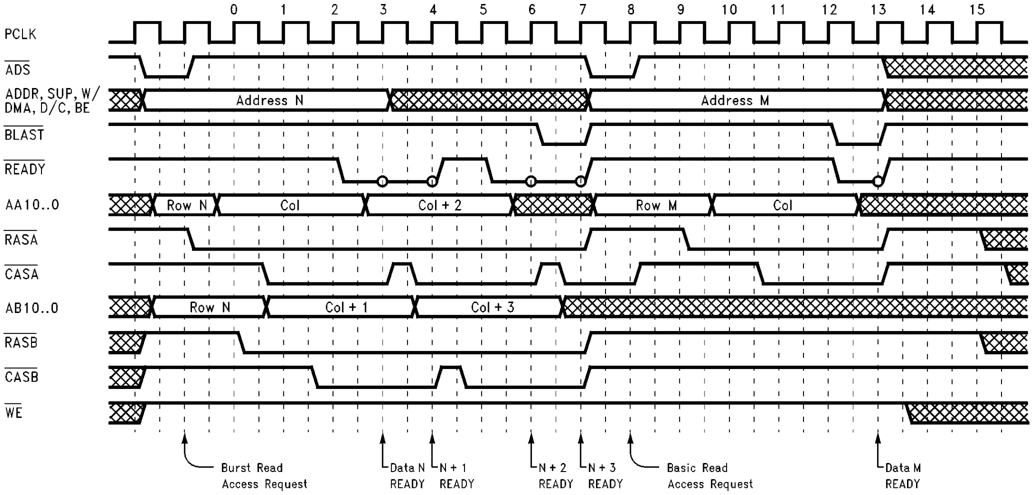


TL/V/11804-7

FIGURE 2.4. Non-Interleave Read/Write Access

2.5 Extended Cycle Mode

Up until this point the timing diagrams have depicted the minimum cycle mode of operation (Configuration Bit 20 = 0). However, when Row and/or Column access delay becomes critical (with slower DRAMs or higher clock rates) then an additional wait state can be inserted to extend the RAS/CAS cycle. The diagram below documents the case of a basic access and a burst read/write (row compare disabled). The results when row compare is enabled are similar to those in Section 2.3 except that the CAS strobes are 2.5 cycles wide as opposed to 1.5 cycles.



TL/V/11804–8

FIGURE 2.5. Basic and Burst Read Access for Extended Cycle Mode

3.0 MEMORY INTERFACE

In this chapter the NSBMC096/memory interface is explained. More specifically, how the processor address is translated into a memory address. This section also provides the calculations required to determine the DRAM speed requirements for a given memory configuration based on the NSBMC096 controller. The access speed required for system DRAM is a function of processor clock speed, processor data setup/hold times, bus buffer delays, NSBMC096 I/O delays and operating mode.

3.1 Address Multiplexing

The Processor address is translated into a Row/Column address as listed in Table I. During the Row phase of an access the memory address lines (MA11..0) are derived from the addresses under the ‘‘Row’’ heading and from the ‘‘Col’’ section of the table during the column phase. After the initial access of a burst the memory address is automatically incremented for the column. Table I contains the address mapping for normal interleave operation. Tables II–V reveal that the mapping changes according to the memory block size configured when non-interleave operation is selected.

TABLE I. Interleave Mode (Any DRAM Size)

Row	Col	Memory Address
A25	A26	MA11
A23	A24	MA10
A21	A22	MA9
A19	A20	MA8
A18	A10	MA7
A17	A9	MA6
A16	A8	MA5
A15	A7	MA4
A14	A6	MA3
A13	A5	MA2
A12	A4	MA1
A11	A3	MA0

TABLE II. Non-Interleave 64 MB Block (DRAM Size = 3)

Row	Col	Memory Address
A25	A4	MA11
A23	A24	MA10
A22	A22	MA9
A19	A20	MA8
A18	A10	MA7
A17	A9	MA6
A16	A8	MA5
A15	A7	MA4
A14	A6	MA3
A13	A5	MA2
A12	A3	MA1
A11	A2	MA0

TABLE III. Non-Interleave 16 MB Block (DRAM Size = 2)

Row	Col	Memory Address
*	*	MA11
A23	A4	MA10
A21	A22	MA9
A19	A20	MA8
A18	A10	MA7
A17	A9	MA6
A16	A8	MA5
A15	A7	MA4
A14	A6	MA3
A13	A5	MA2
A12	A3	MA1
A11	A2	MA0

TABLE IV. Non-Interleave 4 MB Block (DRAM Size = 1)

Row	Col	Memory Address
*	*	MA11
*	*	MA10
A21	A4	MA9
A19	A20	MA8
A18	A10	MA7
A17	A9	MA6
A16	A8	MA5
A15	A7	MA4
A14	A6	MA3
A13	A5	MA2
A12	A4	MA1
A11	A3	MA0

TABLE V. Non-Interleave 1 MB Block (DRAM Size = 0)

Row	Col	Memory Address
*	*	MA11
*	*	MA10
*	*	MA9
A19	A4	MA8
A18	A10	MA7
A17	A9	MA6
A16	A8	MA5
A15	A7	MA4
A14	A6	MA3
A13	A5	MA2
A12	A3	MA1
A11	A2	MA0

3.2 Memory Refresh

A RAS only refresh cycle, is performed at a rate determined by an internal refresh counter. When the counter decrements to its terminal count value, the NSBMC096 will start the refresh memory cycle as soon as an idle period is available. The refresh counter, however, does not pause for refresh memory cycle completion. It operates continuously and independently to guarantee the overall device refresh rate. The refresh address generator always maintains a full 12-bit refresh address. The refresh row address is automatically incremented between refresh cycles. An outstanding refresh request has priority over the initiation of a new access. The refresh period is derived from the system clock and the value programmed into configuration bits [11..6] as:

$$\frac{\text{(Programmed Value + 1)}}{\text{PCLK Frequency (MHz)}} - {}^{*}16\ (\mu\text{s per row})$$

The valid range for the programmed value is from 1 to 63. If the value 0 is used refresh cycle execution will be disabled.

3.3 Memory Performance Requirements

When selecting memory for use with the NSBMC096 it is important that the performance of the devices be examined with regard to the rate at which the system is to operate. The NSBMC096 only supports page mode devices. However, since these are the most prevalent types of devices on the market, this is not a constraint.

Particular attention must be paid to the page mode behavior of the memory devices. Most DRAM manufacturers offer the fast page feature on memories of a megabit or greater. The characteristics of the fast page mode devices are generally those required for reliable operation at speeds of 25 MHz and above. The following sections present some of the critical performance requirements of memory devices as a function of system operating frequency.

3.3.1 RAS Pre-Charge Time

A minimum of two PCLK cycles are provided between successive RAS cycles. Therefore the required RAS pre-charge time is given by:

$$t_{RP} \leq 2(t_{PC}) + t_{RHL} - t_{RLH}$$

Where: t_{RP} = DRAM RAS Pre-Charge Time

t_{PC} = PCLK Cycle Time

t_{RHL} = RAS High to Low Delay

t_{RLH} = RAS Low to High Delay

Because $t_{RHL} > t_{RLH}$ always, this requirement reduces to: $t_{RP} \leq 2(t_{PC})$

3.3.2 Access Time from RAS

It is possible to control the required RAS access time of the DRAM memory using the NSBMC096 Configuration register (Bit 20). The basic RAS access time is calculated as:

$$t_{RAC} \leq 3 * t_{PC} - t_{RHL} - t_{BUF} - t_{SU}$$

Where: t_{RAC} = DRAM RAS Access Time

t_{PC} = PCLK Cycle Time

t_{RHL} = RAS High to Low Delay

t_{BUF} = Buffer Delay

t_{SU} = i960CA Data Setup Time

When configuration bit 20 is set to 1, an additional PCLK cycle is inserted into the RAS cycle and t_{RAC} is increased by an additional t_{PC} period.

3.3.3 CAS Pre-Charge Time

The CAS pre-charge time during page mode access is one of the factors that determines the page mode cycle time. The maximum permissible value is given as:

$$t_{CP} \leq t_{PCH} + t_{CHL} - t_{CLH}$$

Where: t_{CP} = DRAM CAS Pre-Charge Time

t_{PCH} = PCLK High Time (t_{PC}/2 Approx.)

t_{CHL} = CAS High to Low Delay

t_{CLH} = CAS Low to High Delay

Because $t_{CHL} > t_{CLH}$ at all times, this requirement reduces to: $t_{CP} \leq t_{PCH}$.

3.3.4 Access Time from CAS

The required CAS access time is given as:

$$t_{CAC} \leq 1.5 * t_{PC} - t_{CHL} - t_{BUF} - t_{SU}$$

Where: t_{CAC} = DRAM Page Mode CAS Access Time

t_{PC} = PCLK Cycle Time

t_{CHL} = CAS High to Low Delay

t_{BUF} = Buffer Delay

t_{SU} = i960CA Data Setup Time

3.4 Write Cycle Restrictions

During a burst write cycle the data hold time of the DRAM must be considered. Typically the DRAM requires a non-zero data hold time following the assertion of the CAS strobe for an early write cycle. Two alternatives exist in meeting the hold time specification:

1. Configure the NSBMC096 for burst write disable, or
2. use registers or latches on the data into the DRAM.

The method chosen will depend on desired cost/performance ratio for a given application. These methods are further described in Section 5.2 sub-sections 1 through 5.

4.0 TIMER FUNCTIONS

In addition to memory support, the NSBMC096 also contains several features which further enhance its potential to maximize system integration. The first of these additional features is a 24-Bit programmable interval timer designed to divide the system clock (PCLK) by a programmable division factor. This functionality is usually required in most microprocessor applications and would otherwise require additional components.

A Bus Watch Timer, also integrated into the NSBMC096, is necessary to avoid bus ‘‘freezing’’ when an access is initiated in a region of address space which doesn’t reach the READY condition.

4.1 Interrupt Timer Operations

The interrupt timer function of the NSBMC096 consists of a 24-bit down counter that automatically re-loads itself on terminal count with the count value contained in configuration register bits 32–55. Consequently, the timer output signal (TINT) is pulsed at a constant frequency as it is primarily intended for use as a clock tic interrupt generator. Both edge and level sensitive interrupt modes are supported. When enabled as an edge sensitive interrupt, TINT pulses low for one PCLK cycle. If level sensitive operation is desired then the output will be asserted low until it is explicitly acknowledged. Acknowledgement is accomplished as a ‘‘Special Operation’’ (Appendix A).

During system boot and initialization a 24-bit count value should be loaded into the counter register and the counter enabled using the ‘‘Special Operation’’ functions dedicated to that purpose (refer to Appendix A for the special operation codes). The interrupt rate can be changed at any time by simply storing a new value into the ‘‘Timer Count Value’’(TCV) field of the configuration register. Since the configuration register is accessed in Byte quantities, the 24-bit TCV must be updated in 3 steps. When a new value is placed in the most significant byte of the TCV, the TCV is immediately loaded into the counter and the new rate then takes effect. Consequently, the TCV should be modified by first updating the lower Bytes so that they will be in place when the counter re-load is forced.

If the counter is disabled for a period of time, then it will reload the TCV so that, when next enabled, the first interrupt will be seen after the full count period.

4.2 Bus Watch Timer

Bus ‘‘Freezing’’ is the condition wherein a processor initiates a data access and expects to get a READY reply although none is generated. This situation can occur with the i960CA if external READY is enabled in some region of memory that is not decoded by a peripheral or memory controller. The NSBMC096 generates a READY status for memory access that it controls. It also asserts READY when it’s configuration registers are accessed. However, access to an address range out of the NSBMC096’s jurisdiction can be problematic if there are un-implemented ‘‘Holes’’ in the address space. The 5-bit Bus Watch Timer can detect the bus freeze condition and force READY and/or an interrupt by monitoring the processors DEN (data enable) output. When the processor initiates an access it asserts the DEN signal and rescinds it only when the access has completed. In the absence of an internal or external READY condition, DEN will remain asserted indefinitely. When DEN is asserted the Bus Watch Timer begins to count down from it’s initialized value. If it reaches terminal count then a READY and/or interrupt (via the BERR output) will be generated. After DEN is rescinded, the counter is re-loaded to be prepared for the next access. The BERR signal can be used as a level or edge sensitive interrupt depending on the state of configuration bit 29 (Appendix A).

5.0 SYSTEM INTERCONNECT

This section describes the connection of the NSBMC096 to both the host i960CA processor, and the DRAM memory devices. It describes some of the possible buffer strategies for interconnecting the i960CA data bus to the memory array inputs and outputs.

5.1 Signal Description

The NSBMC096 signals can be subdivided into 3 distinct categories: processor interface, memory interface and buffer control signals. A complete listing of these signals and the device pin-outs is presented in the NSBMC096 data sheet in addition to Appendices 2 and 3 of this document.

What follows is a brief functional description of the major signal groups.

5.1.1 Processor Interface

The signals in this group are assigned the same names as their counterparts on the i960CA. They are designed to be connected directly to the i960CA and the use of multiple (up to 4) NSBMC096s in a system is fully supported. The outputs on the reply signals (READY and BTERM) have been designed so that they may be simply wire ‘‘OR’’ed together and connected **directly** to the processor interface. The outputs are tristate not open collector and require only a nominal pull-up to V_{CC} of approximately 4.7 kΩ.

Note: Devices that generate READY/BTERM from a totem pole output require the use of a logic gate such as a 74F08 to combine the multiple sources together.

5.1.2 Memory Interface

The memory block controlled by the NSBMC096 is organized as two banks of 32 bits each. Parity is not directly supported by the controller since the data bus is not connected to the NSBMC096. However, parity checker/generator buffers can be used for this purpose.

The memory interface outputs support high current drivers that will drive loads to 320 pF per bank. Given a nominal input load of 7 pF per device, and a trace capacitance of 18.5 pF/ft., 36 memory devices per bank (byte parity included) are easily supported. These outputs however must be externally matched to the input impedance of the DRAM memory array. A passive serial or parallel terminating network is all that is required. In practice, a series resistor of between 15Ω and 33Ω is optimal for most applications.

5.1.3 Buffer Control Signals

The transfer of Instructions and Data from the memory subsystem to the processor data bus is facilitated by buffers controlled by the NSBMC096. The 4 signals provided for this purpose operate in multiple modes according to the Buffer Mode Field of the configuration register.

Two of these signals provide low true transparent latch enable controls for use during data transfers from the i960CA to memory. They are designed to operate with 74FCT543 style transparent latches to provide additional data hold time during write operations at high PCLK speeds or with slow memories at lower speeds.

The functions performed by the remaining two signals change according to programmed mode. The signal names, as they appear on the logic symbol, reflect the functions performed in operational mode 0. Table VI shows the control signals and their assigned names in the various operating modes.

TABLE VI. Mode Dependent Buffer Control Signals

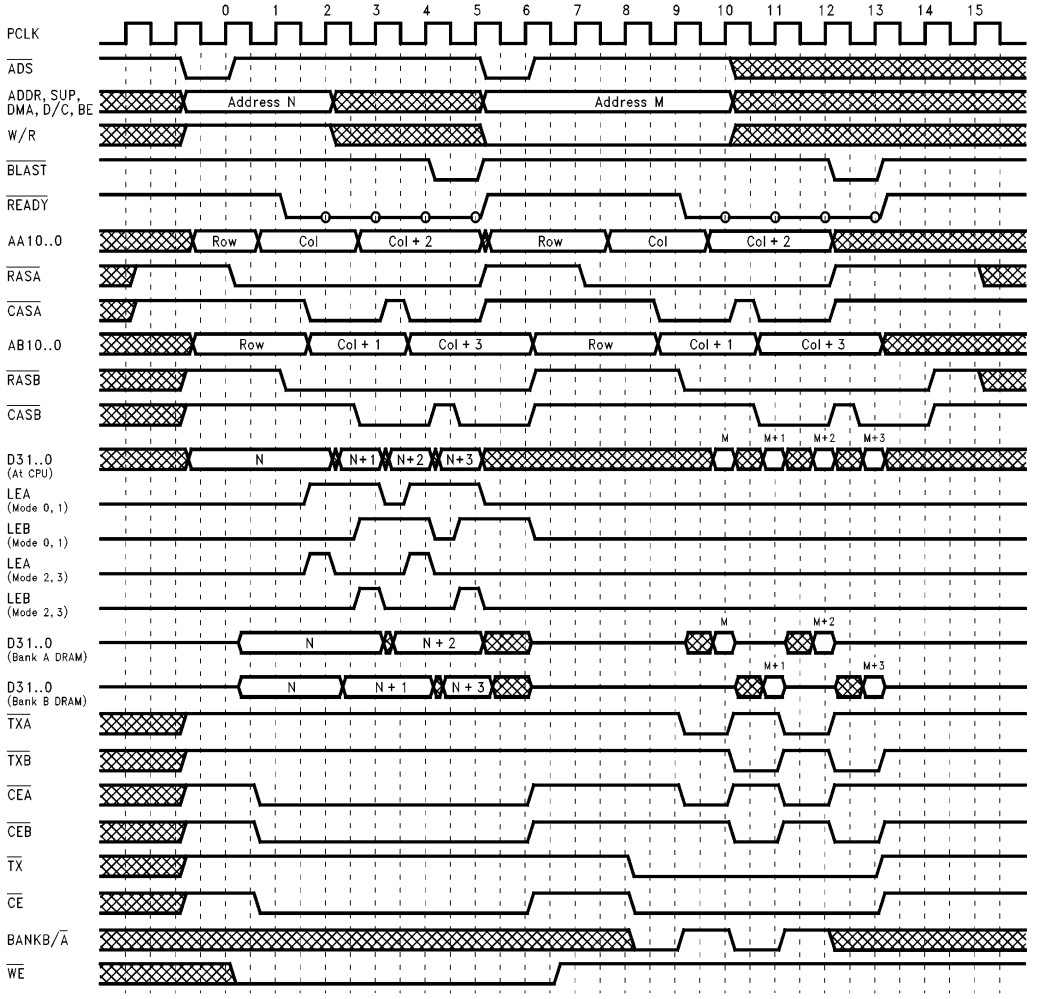
Mode 0 (Default)	Mode 1	Mode 2	Mode 3
*TxA	*CeA	*Tx	*Ce
*TxB	*CeB	BankB/*A	BankB/*A
*LEA	*LEA	*LEA	*LEA
*LEB	*LEB	*LEB	*LEB

Modes 0 and 1 are primarily designed for use with bit wide memories. Since these memories do not have output enables, separate buffers for each bank are required to multiplex between the two banks. Modes 2 and 3 are designed to take advantage of the output enables on nibble wide memories.

Modes 1 and 3 can be used with buffers which have a chip enable and a direction control. The direction controls are connected to the memory write enables, and the chip enables are connected to the appropriate chip enable signal (*CeA/B). When devices with select and direction controls are used, care should be taken to connect the 1/0 ports to the bus so that the low true write enables signals (*MWeA/B) set the data transfer direction from the processor into the memories (e.g., for 74F245's connect port A to the memories).

For modes 2 and 3 the NSBMC096 generates a signal called BANKB/A. The high level of this signal indicates that BANK B must be enabled starting at the **next** rising edge of PCLK. Conversely, the low level indicates that BANK A must be enabled starting at the next rising edge of PCLK. In order for the memory chip enables for each bank to operate with correct timing, the memory output enable (derived from BANKB/*A) delay from PCLK must be kept to a minimum. Section 5.2 provides more details on the use of the BANKB/*A signal.

Figure 5.1.3 provides a composite view of the buffer controls displayed for each of the modes.



TL/V/11804–9

FIGURE 5.1.3. Buffer Control Timing for All Buffer Control Modes

5.1.3.1 Control Signal Interpretation—Mode 0

This mode is designed to be used with bit wide memories and buffers which have separate output enables for each direction (e.g., 74FCT543). The bus transmit enable controls for each bank (*TxA/B) are asserted low for read operations from memory, while the memory bank write enables (*MWeA/B) are asserted low for write operations.

5.1.3.2 Control Signal Interpretation—Mode 1

When configured in this mode, the NSBMC096 asserts the separate data bus chip enable controls for each bank during both read and write operations. The signals *MWeA/B are asserted low only during write operations and determine the direction of data transfers. This configuration mode is appropriate for the direct control of 74F245 style devices that have a single enable and a direction control.

5.1.3.3 Control Signal Interpretation—Mode 2

This configuration supports nibble wide memories and 74F543 style buffers that have output enable controls. The NSBMC096 generates a single transmit enable for data access. The transmit enables are asserted low for read operations to either bank. The memory bank write enables *MWeA/B are asserted low for write operations to either bank and are used as in Mode 0. The BANKB/*A signal is used to select which bank is to be next accessed for read operations.

5.1.3.4 Control Signal Interpretation—Mode 3

If this mode is selected, a single chip enable is generated for data access to both banks. It is asserted low for both read and write operations to either bank. The BANKB/*A signal is used to select which bank is read. Both memory bank write enables are asserted low for write operations to either bank.

5.2 Bus Buffer Interconnect Strategies

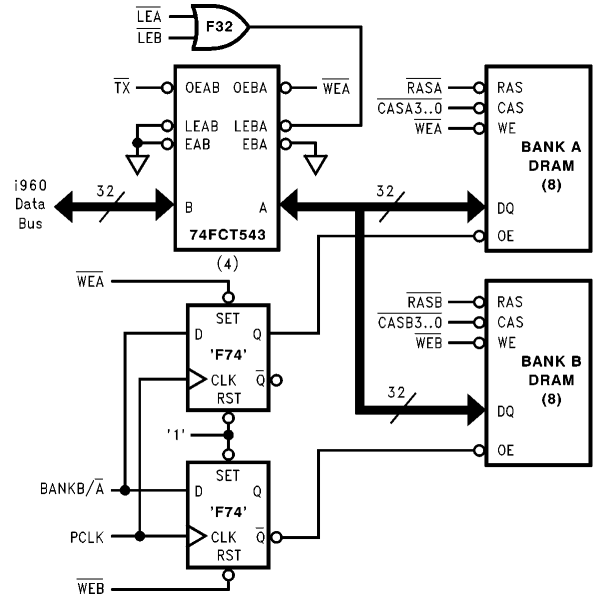
The size, and cost of the memory system design is a function of the types of DRAM chosen and the desired system performance. The sections following expand some of the possible configurations and illustrate them with appropriate diagrams. Not all of these configurations will work at higher clock rates in all modes. Therefore, care should be taken to select memory and buffer components to match the speed requirements. Appendix C elaborates on some of the necessary calculations.

5.2.1 Nibble Wide Memories with “543” Buffers

If nibble wide DRAM devices are used in conjunction with 74FCT543 octal buffers, it is possible to design a low cost system with minimal component count. If 256k by 4-bit (1 Mbit) devices are used, a memory size of 2 MB results. Since the NSBMC096 accommodates devices up to 64 MB (16M by 4-Bit) in size, the system memory size may be easily expanded using the larger devices.

Figure 5.2.1 shows the connection of the NSBMC096 control signals to the data bus buffers and memory. The NSBMC096 should be configured for buffer mode 2.

The output enables of the DRAM are driven by signals derived from the BANKB/*A signal. The write enable strobes from the NSBMC096 are used to disable the output enable during write cycles.



TL/V/11804–10

FIGURE 5.2.1. Circuit for Nibble Wide DRAM Using “543” Buffers and Mode 2 Controls

5.2.2 SIMM Modules with "543" Style Buffers

Unfortunately standard configurations of SIMM modules lack an output enable signal and rely instead on the CAS signal to gate the output buffers. In the previous example, the DRAM output enables were used to provide multiplexing between the two banks of memory for data read cycles. However, when using SIMMs or bit wide DRAMs, it is necessary to individually buffer each of the two memory banks to provide the multiplexing function. An illustration of this is seen in *Figure 5.2.2.* The NSBMC096 must be configured for buffer mode 0 in order to generate the required TXA and TXB signals.

It is important to configure the RAS signals and distribute them properly depending on the type of memories used. When using 8-bit SIMMs or Bit wide DRAMs the "DRAM Size" field (bits 12–14) of the configuration register should be set to 0 through 3 (depending on memory size). This insures that all 4 RAS signals of a bank operate in unison. This allows one RAS signal to drive each Byte of DRAMs (8 or 9 component loads) so that the load is distributed. If 32-bit SIMMs are used then one RAS signal per module is used and the DRAM size code should be set to between 4 and 7. The additional RAS signals can then be used to expand memory size.

The disadvantage of this buffer configuration is that a larger number of buffers are required than in the previous example. One benefit of using SIMM modules is that memory expansion can be accomplished by simply replacing the modules with higher density ones. Using 32-bit modules, only 2 are required for interleave operation.

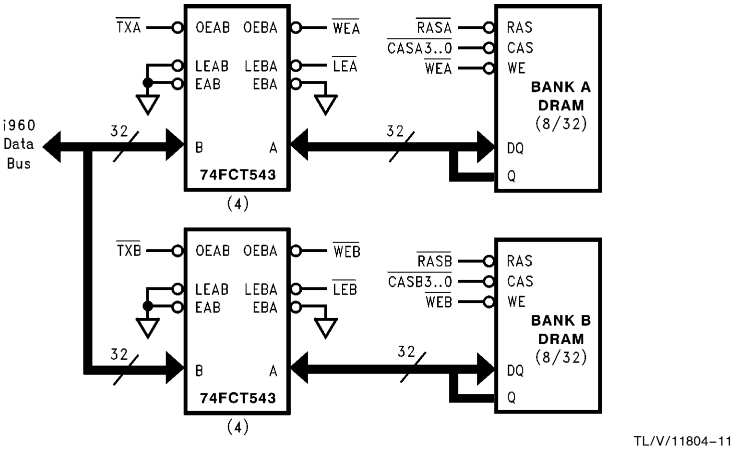


FIGURE 5.2.2. Data Buffering for SIMM and Bit Wide DRAM Using "543" Latches

5.2.3 Buffers with Direction Control

Many popular buffers such as the ubiquitous "245" have a common enable signal and a direction control. These buffers are easily used in place of the "543" components of the last two sections. However, for *Figure 5.2.1* buffer mode 3 should be used with the enable of the "245" driven from the CE signal and the direction controlled by one of the write enables. The same replacement can be made in *Figure 5.2.2* with CEA and CEB driving the enables for the respective sets of "245s". Since "245" devices do not provide latching, it may be necessary to disable burst write operation. Consequently the "543" option is preferred.

5.2.4 Unbuffered Data Bus

The most efficient use of board real estate can be realized by directly connecting the DRAM data I/O to the processor. However, data bus loading should be considered. For example, a single block of nibble wide DRAMs results in only 2 DRAM data loads per processor data signal. However, a fully expanded system with 4 blocks of DRAM places 8 loads on the data bus. Only nibble wide DRAMs can be used as they are equipped with an output enable for multiplexing between the two banks of interleave memory. The output enables can be driven as depicted in *Figure 5.2.1*. As explained previously, it is probably necessary to disable burst write, since the data is not latched, to improve hold time.

5.2.5 Expanding Memory Size

When using nibble wide memories it is possible to populate a single NSBMC096 with up to 4 blocks of memory devices. Expansion in this manner is accomplished using a pair of RAS signals for each block of memory as depicted in *Figure 5.2.3*. Using 1 megabit (256k by 4) devices the base memory size (only block 0 installed) is 2 MB (configure the "DRAM size" bit field to 4). If 4, 6 or 8 MB is required, then the additional memory devices can be installed.

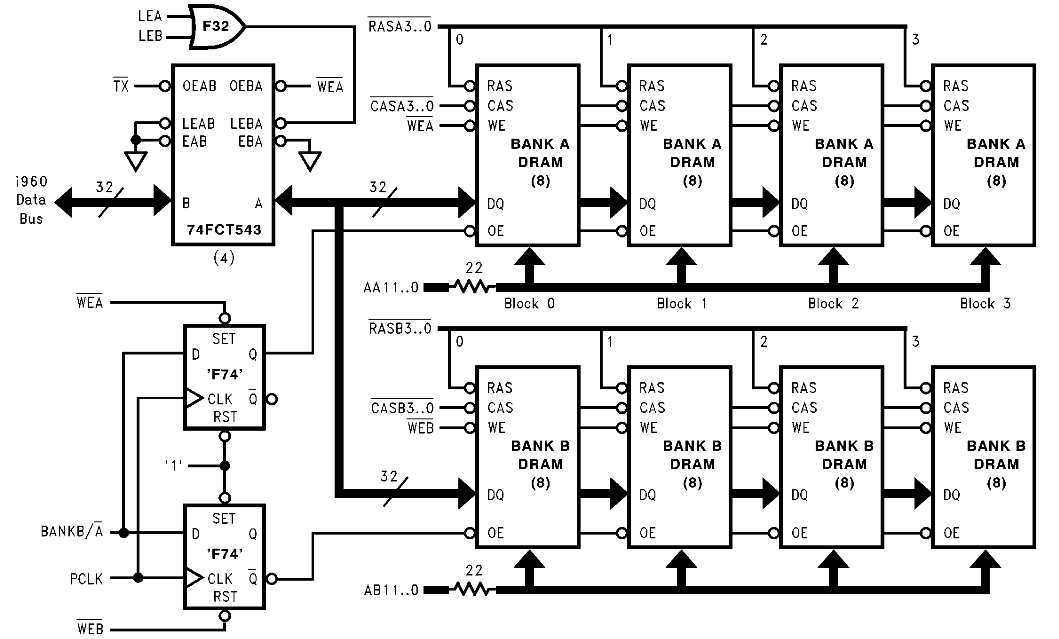


FIGURE 5.2.3. Employing Multiple RAS Outputs for Memory Expansion

6.0 SOFTWARE DESIGN

The ease with which the NSBMC096 may be integrated into a circuit design has been illustrated in the foregoing sections. There are no restrictions, placed on software architecture, that are imposed by the NSBMC096 with the exception of the requirement for proper initialization.

6.1 General Considerations

If performance is to be optimized, single external data access should be minimized where possible in favor of data bursts. In order to do this, call tree analysis and subroutine flow tracing should be performed with a view to maximizing data locality of reference. While an optimizing compiler can go a long way in maximizing local variable placement, array access, and loop constructs, the responsibility of global or external data placement is the province of the programmer. By supporting burst access to data memory, the NSBMC096 promotes the use of locality of reference augmented by register "spill/fill" as a means of enhancing program performance.

6.2 i960CA Configuration

In order for the NSBMC096 to interact correctly with the i960CA it is imperative that the processors "Memory Region Configuration" be properly initialized for the region in which the NSBMC096 resides. The following parameters should be used:

Burst	— Enabled
External Ready	— Enabled
Pipe-Lining	— Disabled
N_{RAD}	— 0
N_{RDD}	— 0
N_{WAD}	— 0
N_{WDD}	— 0
N_{XDA}	— 0
Bus Width	— 32-Bit

6.3 Software Configuration of the NSBMC096

Configuration of the NSBMC096 occurs when supervisor write cycles are initiated within the address range FF0F0000 to FF0FFFFF. Each byte of the configuration register is mapped into a 1k region of the processors address space. The lower bits of the address (bits 2..9) are used as the "data" for the configuration register. The address required to load a configuration byte is built in the following manner:

Store Address	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
	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	ID2	ID1	ID0	C2	C1	C0	D7	D6	D5	D4	D3	D2	D1	D0	X	X
	Configuration Base Address																NSBMC096 ID#			Conf. Byte Select			Configuration Data									

TL/V/11804–14

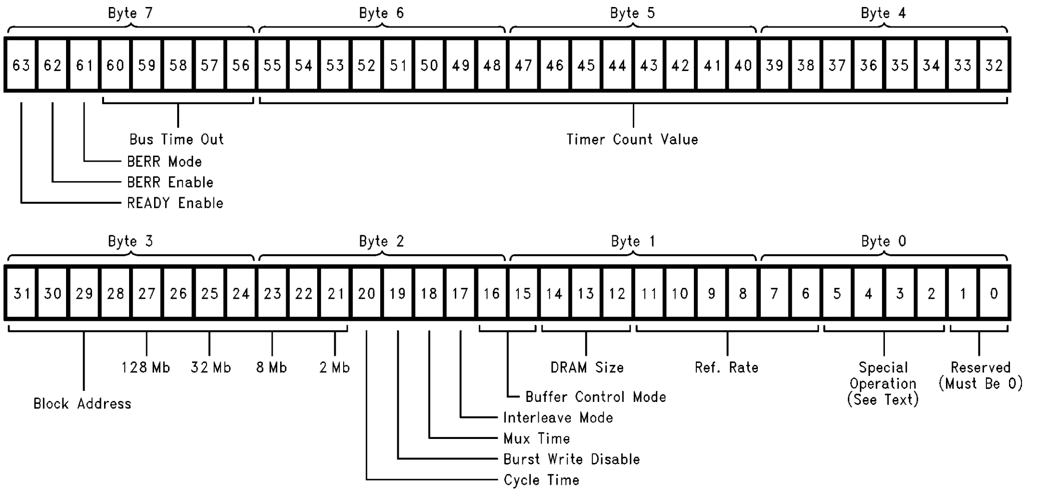
Where: NSBMC096 ID# = the ID code placed on the ID0..2 pins of the NSBMC096
Conf. Byte Select = one of 8 bytes of the 64-bit configuration word
Configuration Data = data to be written into the selected byte of the configuration

The following program could be used to configure the NSBMC096:

```
          # Register Usage
          #        r0                    _ Base Address of NSBMC096
          #        r1                    _ Low order Configuration Word
          #        r2                    _ High order Configuration Word
          #        r3                    _ Temp Register
          #        r4                    _ Count Value
          #        r5                    _ Configuration Store Address

          # Initialize Constants
          lda      FF0F0000H, r0         # Base Address of NSBMC096 (ID=0)
          lda      00055500, r1          # Low order Configuration Word
          lda      D907A120, r2          # High order Configuration Word
          call     cfgword               # Configure the first 32 bits
          mov      r2, r1
          lda      1000H, r3
          or       r3, r0, r0            # Base address of config bytes 4-7
          call     cfgword               # Configure the second 32 bits
          b        systemboot            _ proceed with system boot
cfgword                                  # Configure 4 bytes of the NSBMC096
          xor      r4, r4, r4            # Count Value = 0
loop:
          mov      r1, r3                # Configuration Word
          shlo     3, r4, r5             # Calculate base of a configuration byte
          extract  r5, 8, r3             # Extract required field
          shlo     2, r3, r3             # Convert byte address to word address
          shlo     10, r4, r5            # Calc. byte select portion of address
          or       r0, r5, r5            # Or in Base Address
          or       r3, r5, r5            # Or in the configuration data
          st       r5, r5                # Configure a byte
          cmpinci  4, r4, r4             # Increment Loop Count
          bl       loop                  # Branch until all 4 bytes complete
          ret
```

7.0 APPENDIX A—CONFIGURATION REGISTER SUMMARY



TL/V/11804–13

Bits 2–5 Special Operations

5	4	3	2	
0	0	0	0	Access other Configuration Bits 0–1, 6, 7
0	1	0	0	Instruction Access Row Compare Disable (Default)
0	1	1	0	Instruction Access Row Compare Enable
0	1	0	1	Data Access Row Compare Disable (Default)
0	1	1	1	Data Access Row Compare Enable
1	0	0	0	Acknowledge Timer Interrupt
1	0	1	0	Enable Timer Output for Level Sense Interrupt
1	1	0	0	Disable All Timer Interrupts (Default)
1	1	1	0	Enable Timer Output for Edge Sense Interrupt

Bits 6–11 Refresh Rate

Refresh Rate = (PCLK Frequency)/(16 * Programmed Value + 1)

Bits 12–14 DRAM Size

Memory Size Code	Memory Block Size	Max Banks	Memory Types
0 0 0	2M	1	256K x 1
0 0 1	8M	1	1M x 1
0 1 0	32M	1	4M x 1
0 1 1	128M	1	16M x 1, 16M x 4
1 0 0	2M	4	64K x 4
1 0 1	8M	4	256K x 4
1 1 0	32M	4	1M x 4
1 1 1	128M	4	16M x 4

Bits 15–16 Buffer Mode

Mode Bits	Signal 1	Signal 2	Signal 3	Signal 4
00	TXA	TXB	LEA	LEB
01	CEA	CEB	LEA	LEB
10	TX	BankB/*A	LEA	LEB
11	CE	BankB/*A	LEA	LEB

Bit 17 Interleave Mode

0 = Interleave Operation

1 = Non-Interleave Operation

Bit 18 Row to Column Multiplex

0 = ½ PCLK Cycle RAS Hold

1 = 1 PCLK Cycle RAS Hold

Bit 19 Burst Write Disable

0 = Burst Write Enable

1 = Burst Write Disable

Bit 20 Cycle Time

0 = Minimum RAS/CAS

1 = Maximum RAS/CAS

Bits 32–55 Timer Count Value

TINT Output Rate = PCLK/(Timer Count Value)

Bits 56–60 Bus Time Out

Number of PCLK cycles for a bus error

Bit 61 BERR Mode

0 = Edge Sense Interrupt

1 = Level Sense Interrupt

Bit 62 BERR Enable

0 = Disable

1 = Enable

Bit 63 READY Enable

0 = READY not asserted for bus error

1 = READY asserted when bus error occurs

8.0 APPENDIX B—PIN LIST SORTED BY SIGNAL NAME

Name	QFP Pin
A2	121
A3	122
A4	123
A5	124
A6	125
A7	126
A8	127
A9	128
A10	129
A11	130
A12	131
A13	132
A14	1
A15	2
A16	3
A17	5
A18	8
A19	6
A20	7
A21	9
A22	11
A23	12
A24	10
A25	14
A26	13
A27	15
A28	20
A29	21
A30	22
A31	19
AA0	53
AA1	54
AA2	55
AA3	56
AA4	59
AA5	60
AA6	61
AA7	62
AA8	65
AA9	66

Name	QFP Pin
AA10	67
AA11	68
AB0	87
AB1	88
AB2	89
AB3	90
AB4	93
AB5	94
AB6	95
AB7	96
AB8	99
AB9	100
AB10	101
AB11	102
ADS	34
BE0	29
BE1	32
BE2	35
BE3	36
BERR	27
BLAST	31
BTERM	37
CASA0	71
CASA1	72
CASA2	73
CASA3	74
CASB0	105
CASB1	106
CASB2	107
CASB3	108
D/*C	23
DEN	30
ID0	39
ID1	40
ID2	41
LEA	43
LEB	44
MWEA	82
MWEB	118
PCLK	25

Name	QFP Pin
RASA0	77
RASA1	78
RASA2	79
RASA3	80
RASB0	111
RASB1	112
RASB2	113
RASB3	114
READY	38
Reserved	42
RESET	120
SUP	24
TINT	26
TXA	45
TXB	46
V_{CC}	4
V_{CC}	47
V_{CC}	57
V_{CC}	63
V_{CC}	69
V_{CC}	75
V_{CC}	81
V_{CC}	91
V_{CC}	97
V_{CC}	103
V_{CC}	109
V_{CC}	115
V_{SS}	33
V_{SS}	48
V_{SS}	58
V_{SS}	64
V_{SS}	70
V_{SS}	76
V_{SS}	86
V_{SS}	92
V_{SS}	98
V_{SS}	104
V_{SS}	110
V_{SS}	119
W/*R	28

9.0 APPENDIX C—DESIGN EXAMPLE

The purpose of this Appendix is to demonstrate how to configure the NSBMC096 for a specific application and to verify timing requirements. For the purposes of this example, the following assumptions will be made:

System Clock:	25 MHz		
Memory Size:	2 MB using 16k–256k *4 DRAMs expandable to 8 MB using 64 DRAMs		
Memory Configuration:	Minimum Cycle Time, Burst Write Capabilities, 2-Way Interleave, 512 Refresh cycles in 8 ms.		
Timer Output:	50 Hz, Edge Sensitive Interrupt		
74FCT543 Prop. Delay:	2.5 ns Minimum, 8.5 ns Max		
74F32 Prop. Delay:	3 ns Minimum, 6.5 ns Max		
DRAM:		100 ns	80 ns
	RAS Access Time t_{RAC}	100 ns	80 ns
	CAS Access Time t_{RAC}	25 ns	20 ns
	RAS Precharge Time t_{RP}	70 ns	60 ns
	CAS Precharge Time t_{RP}	10 ns	10 ns
	Row Address Hold t_{RAH}	15 ns	10 ns
	Column Address Setup t_{ASC}	0 ns	0 ns
	Data Hold Time t_{DH}	20 ns	15 ns

DATA HOLD TIME (Burst Write)

If no latches are used for the data bus:

$t_{DH}(\text{DRAM}) \leq t_{PCL}-t_{CHL}(\text{NSBMC096})$

$\leq 20-19 = 1$ ns (Worst Case)

Obviously more hold time is required. Using the circuit in *Figure 5.2.1* to provide latching, the latch enable signals LEA and LEB track the timing of CAS with very low skew. Consequently, the latch enable signals instruct the latches to ‘‘Hold’’ just as CAS is being asserted. This guarantees a minimum hold time of $t_{PC}/2$ (less some possible clock skew):

$t_{DH}(\text{DRAM}) \leq t_{PC}/2 + t_{P(min)}$ (F32) $+ t_{P(min)}$ (IDT74FCT543)—Clock Skew

$\leq 20 + 3 + 2.5 - 1 = 24.5$ ns (Worst Case)

This is sufficient for 80 ns and 100 ns DRAM.

RAS Pre-Charge Time

From 3.3.1:

$t_{RP} \leq 2(t_{PC}) = 80$ ns (Worst Case)

Sufficient for 80 ns and 100 ns DRAM.

Access Time from RAS

From 3.3.2:

$t_{RAC} \leq 3{*}t_{PC} - t_{RHL(max)}$ (NSBMC096) $- t_{SU}$(i960CA)   For Minimum Cycle Mode

$\leq 120 - 24 - 6.5 - 5 = 84.5$ ns (Worst Case)

Sufficient for 80 ns DRAM. Simply add t_{PC} for Maximum Cycle mode resulting in 124.5 ns.

CAS Pre-Charge Time

From 3.3.3:

$t_{CP} \le t_{PCH} \cong 20$ ns

This is sufficient for both 80 ns and 100 ns DRAM.

Access Time from CAS

From 3.3.4:

$t_{CAS} \le 1.5 * t_{PC} - t_{CHL(max)} - t_{BUF(max)}$
(IDT74FCT543)—t_{SU} (i960CA)
$\le 60 - 19 - 8.5 - 5 = 27.5$ ns

This is compatible with both 80 ns and 100 ns DRAMs.

Row Address Hold Time

For Mux Time config bit = 0:

$t_{RAH} \le t_M - 4 = 16$ ns

For Mux Time config bit = 1:

$t_{RAH} \le t_M - 4 = 36$ ns

Both modes are compatible with 80 ns and 100 ns DRAMs.

Column Address Setup Time

For Mux Time config bit = 0:

$t_{RAH} \le t_{PC} + t_{CHL} - t_{CAV} = 40 + 19 - 23 = 36$ ns

For Mux Time config bit = 1:

$t_{RAH} \le t_{PCH} + t_{CHL} - t_{CAV} = 40 + 19 - 23 = 16$ ns

Both modes are compatible with 80 ns and 100 ns DRAMs. Chose config bit 18 = 1.

Refresh Rate

For 512 Rows in 8 ms: Refresh 1 row every 16 μs.

Refresh Configuration Constant 25 MHz *16 μs/16 − 1 = 24 = 0 x 18

Timer Count Value

For a 50 Hz Timer interrupt: TVC = 25 MHz/50 Hz = 500000 = 0 x 07A120

Bus Watch Timer

Time out if DEN Asserted for longer than 1 μs: Bus Time Out = 1 μs = 25 cycles = 0 x 19

Configuration

For each of the fields in the configuration register, the following values have been derived for this example: Ref. Rate = 0 x 18, DRAM size = 5, Buffer Mode = 2, Interleave Mode = 0, Mux Time = 1, Burst Write Disable = 0, Cycle Time = 0, Block Address = 0, Timer Count Value = 0 x 07A120, BERR mode = 0, BERR Enable = 1, BERR READY enable = 1. The resulting 64-bit configuration is as follows:

0xD907A12000055500

NSBMC096 Memory Controller for the i960CA—User Application Guide

AN-883

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, CA 95052-8090
Tel: 1(800) 272-9959
TWX: (910) 339-9240

National Semiconductor GmbH
Livry-Gargan-Str. 10
D-82256 Fürstenfeldbruck
Germany
Tel: (81-41) 35-0
Telex: 527649
Fax: (81-41) 35-1

National Semiconductor Japan Ltd.
Sumitomo Chemical
Engineering Center
Bldg. 7F
1-7-1, Nakase, Mihama-Ku
Chiba-City,
Ciba Prefecture 261
Tel: (043) 299-2300
Fax: (043) 299-2500

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductores Do Brazil Ltda.
Rue Deputado Lacorda Franco
120-3A
Sao Paulo-SP
Brazil 05418-000
Tel: (55-11) 212-5066
Telex: 391-1131931 NSBR BR
Fax: (55-11) 212-1181

National Semiconductor (Australia) Pty, Ltd.
Building 16
Business Park Drive
Monash Business Park
Nottinghill, Melbourne
Victoria 3168 Australia
Tel: (3) 558-9999
Fax: (3) 558-9998

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.