National Semiconductor

NS32082-10 Memory Management Unit

General Description

The NS32082 Memory Management Unit (MMU) provides hardware support for demand-paged virtual memory implementations. The NS32082 functions as a slave processor in Series 32000 microprocessor-based systems. Its specific capabilities include fast dynamic translation, protection, and detailed status to assist an operating system in efficiently managing up to 32 Mbytes of physical memory. Support for multiple address spaces, virtual machines, and program debugging is provided.

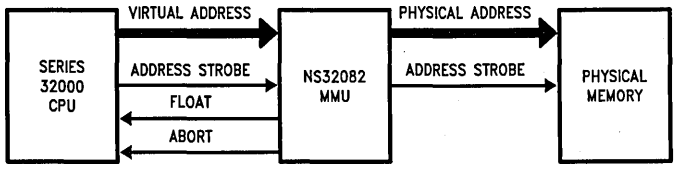
High-speed address translation is performed on-chip through a 32-entry fully associative translation look-aside buffer (TLB), which maintains itself from tables in memory with no software intervention. Protection violations and page faults (references to non-resident pages) are automatically detected by the MMU, which invokes the instruction abort feature of the CPU.

Additional features for program debugging include two breakpoint registers and a breakpoint counter, which provide the programmer with powerful stand-alone debugging capability.

Features

- Totally automatic mapping of 16 Mbyte virtual address space using memory based tables
- On-chip translation look-aside buffer allows 97% of translations to occur in one clock for most applications
- Full hardware support for virtual memory and virtual machines
- Implements "referenced" bits for simple, efficient working set management
- Protection mechanisms implemented via access level checking and dual space mapping
- Program debugging support
- Compatible with NS32016, NS32032 and NS32332 CPUs
- 48-pin dual-in-line package

Conceptual Address Translation Model



TL/EE/8692-1

Table Of Contents

List of Illustrations

List of Illustrations (Continued)

Tables

1.0 Product Introduction

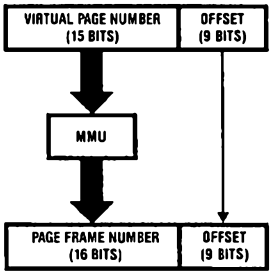
The NS32082 MMU provides hardware support for three basic features of the Series 32000; dynamic address translation, access level checking and software debugging. Dynamic Address Translation is required to implement demand-paged virtual memory. Access level checking is performed during address translation, ensuring that unauthorized accesses do not occur. Because the MMU resides on the local bus and is in an ideal location to monitor CPU activity, debugging functions are also included.

The MMU is intended for use in implementing demand-paged virtual memory. The concept of demand-paged virtual memory is illustrated in *Figure 1-1.* At any point in time, a program sees a uniform addressing space of up to 16 megabytes (the "virtual" space), regardless of the actual size of the memory physically present in the system (the "physical" space). The full virtual space is recorded as an image on a mass storage device. Portions of the virtual space needed by a running program are copied into physical memory when needed.

To make the virtual information directly available to a running program, a mapping must be established between the virtual addresses asserted by the CPU and the physical addresses of the data being referenced.

To perform this mapping, the MMU divides the virtual memory space into 512-byte blocks called "pages." It interprets the 24-bit address from the CPU as a 15-bit "page number" followed by a 9-bit offset, which indicates the position of a byte within the selected page. Similarly, the MMU divides the physical memory into 512-byte frames, each of which can hold a virtual page.

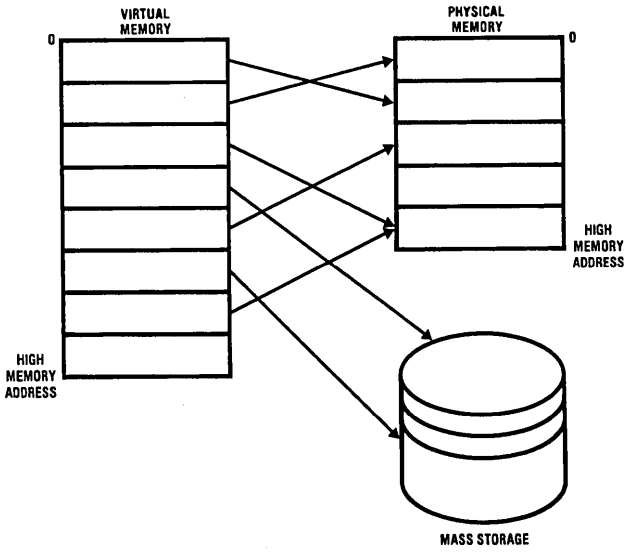
The translation process is therefore modeled as accepting a virtual page number from the CPU and substituting the corresponding physical page frame number for it, as shown in *Figure 1-2.* The offset is not changed. The translated page frame number is 16 bits long, including an additional address bit (A24) intended for physical bank selection. Physical addresses issued by the MMU are 25 bits wide.



TL/EE/8692-3

FIGURE 1-2. NS32082 Address Translation Model

Generally, in virtual memory systems the available physical memory space is smaller than the maximum virtual memory space. Therefore, not all virtual pages are simultaneously resident. Nonresident pages are not directly addressable by the CPU. Whenever the CPU issues a virtual address for a nonresident or nonexistent page, a "page fault" will result. The MMU signals this condition by invoking the Abort feature of the CPU. The CPU then halts the memory cycle,



TL/EE/8692-2

FIGURE 1-1. The Virtual Memory Model

1.0 Product Introduction (Continued)

restores its internal state to the point prior to the instruction being executed, and enters the operating system through the abort trap vector.

The operating system reads from the MMU the virtual address which caused the abort. It selects a page frame which is either vacant or not recently used and, if necessary, writes this frame back to mass storage. The required virtual page is then copied into the selected page frame.

The MMU is informed of this change by updating the page tables (Section 3.2), and the operating system returns control to the aborted program using the RETT instruction. Since the return address supplied by the abort trap is the address of the aborted instruction, execution resumes by retrying the instruction.

This sequence is called paging. Since a page fault encountered in normal execution serves as a demand for a given page, the whole scheme is called demand-paged virtual memory.

The MMU also provides debugging support. It may be programmed to monitor the bus for two virtual or physical addresses in real time. A counter register is associated with one of these, providing a "break-on-N-occurrences" capability.

1.1 PROGRAMMING CONSIDERATIONS

When a CPU instruction is aborted as a result of a page fault, some memory resident data might have been already modified by the instruction before the occurrence of the abort.

This could compromise the restartability of the instruction when the CPU returns from the abort routine.

To guarantee correct results following the re-execution of the aborted instruction, the following actions should not be attempted:

a) No instruction should try to overlay part of a source operand with part of the result. It is, however, permissible to rewrite the result into the source operand exactly if page faults are being generated only by invalid pages and not by write protection violations (for example, the instruction "ABSW X, X", which replaces X with its absolute value). Also, never write to any memory location which is necessary for calculating the effective address of either operand (i.e. the pointer in "Memory Relative" addressing mode; the Link Table pointer or Link Table Entry in "External" addressing mode).

b) No instruction should perform a conversion in place from one data type to another larger data type (Example: MOVWF X, X which replaces the 16-bit integer value in memory location X with its 32-bit floating-point value). The addressing mode combination "TOS, TOS" is an exception, and is allowed. This is because the least-significant part of the result is written to the possibly invalid page before the source operand is affected. Also, integer conversions to larger integers always work correctly in place, because the low-order portion of the result always matches the source value.

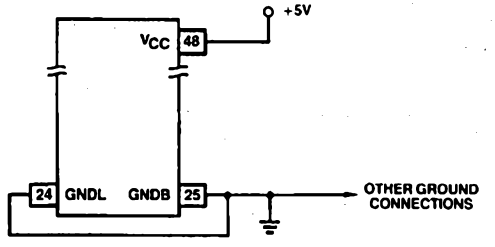
c) When performing the MOVM instruction, the entire source and destination blocks must be considered "operands" as above, and they must not overlap.

2.0 Functional Description

2.1 POWER AND GROUNDING

The NS32082 requires a single 5V power supply, applied on pin 48 (V_{CC}).

Grounding connections are made on two pins. Logic Ground (GNDL, pin 24) is the common pin for on-chip logic, and Buffer Ground (GNDB, pin 25) is the common pin for the output drivers. For optimal noise immunity, it is recommended that GNDL be attached through a single conductor directly to GNDB, and that all other grounding connections be made only to GNDB, as shown below (*Figure 2-1*).



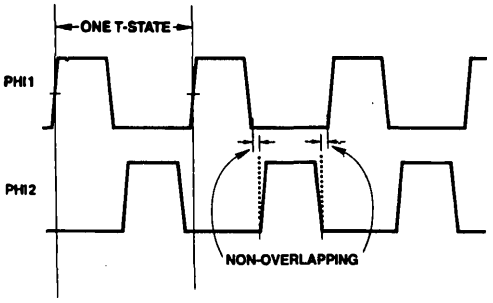
TL/EE/8692-4

FIGURE 2-1. Recommended Supply Connections

2.2 CLOCKING

The NS32082 inputs clocking signals from the NS32201 Timing Control Unit (TCU), which presents two non-overlapping phases of a single clock frequency. These phases are called PHI1 (pin 26) and PHI2 (pin 27). Their relationship to each other is shown in *Figure 2-2*.

Each rising edge of PHI1 defines a transition in the timing state ("T-State") of the MMU. One T-State represents one hardware cycle within the MMU, and/or one step of an external bus transfer. See Section 4 for complete specifications of PHI1 and PHI2.



TL/EE/8692-5

FIGURE 2-2. Clock Timing Relationships

As the TCU presents signals with very fast transitions, it is recommended that the conductors carrying PHI1 and PHI2 be kept as short as possible, and that they not be connected to any devices other than the CPU and MMU. A TTL Clock signal (CTTL) is provided by the TCU for all other clocking.

2.0 Functional Description (Continued)

2.3 RESETTING

The $\overline{RSTI}$ input pin is used to reset the NS32082. The MMU responds to $\overline{RSTI}$ by terminating processing, resetting its internal logic and clearing the appropriate bits in the MSR register.

Only the MSR register is changed on reset. No other program accessible registers, including the TLB are affected.

The $\overline{RST}/\overline{ABT}$ signal is activated by the MMU on reset. This signal should be used to reset the CPU. $\overline{AT}/\overline{SPC}$ is held low for five clock cycles after the rising edge of $\overline{RSTI}$ to indicate to the CPU that the address translation mode must be selected.

The A24/$\overline{HBF}$ signal is sampled by the MMU on the rising edge of $\overline{RSTI}$. It indicates the bus size of the attached CPU. A24/$\overline{HBF}$ must be sampled high for a 16-bit bus and low for a 32-bit bus.

On application of power, $\overline{RSTI}$ must be held low for at least 50 μs after V_{CC} is stable. This is to ensure that all on-chip voltages are completely stable before operation. Whenever a Reset is applied, it must also remain active for not less than 64 clock cycles. The rising edge must occur while PHI1 is high. See *Figures 2-3* and *2-4*.

The NS32201 Timing Control Unit (TCU) provides circuitry to meet the Reset requirements of the NS32082 MMU. *Figure 2-5* shows the recommended connections.

2.4 BUS OPERATION

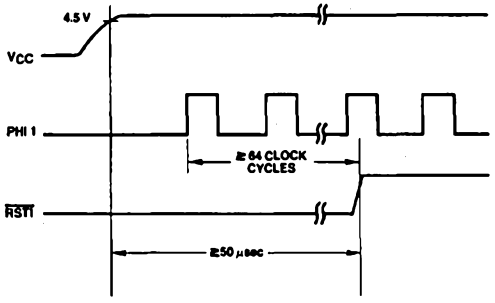
2.4.1 Interconnections

The MMU runs synchronously with the CPU, sharing with it a single multiplexed address/data bus. The interconnections used by the MMU for bus control, when used in conjunction with the NS32016, are shown in *Figure A-1* (Appendix A).

The CPU issues 24-bit virtual addresses on the bus, and status information on other pins, pulsing the signal $\overline{ADS}$ low. These are monitored by the MMU. The MMU issues 25-bit physical addresses on the bus, pulsing the $\overline{PAV}$ line low. The $\overline{PAV}$ pulse triggers the address latches and signals the NS32201 TCU to begin a bus cycle. The TCU in turn generates the necessary bus control signals and synchronizes the insertion of WAIT states, by providing the signal RDY to the MMU and CPU. Note that it is the MMU rather than the CPU that actually triggers bus activity in the system.

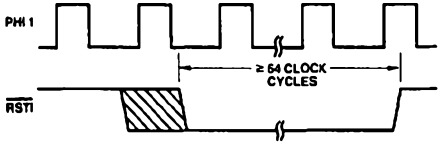
The functions of other interface signals used by the MMU to control bus activity are described below.

The ST0–ST3 pins indicate the type of cycle being initiated by the CPU. ST0 is the least-significant bit of the code. Table 2-1 shows the interpretations of the status codes presented on these lines.



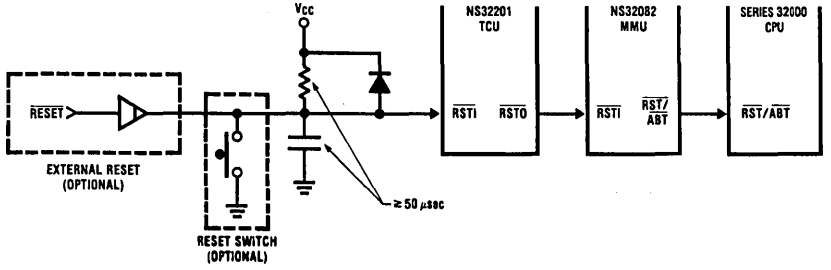
TL/EE/8692-6

FIGURE 2-3. Power-On Reset Requirements



TL/EE/8692-7

FIGURE 2-4. General Reset Timing



TL/EE/8692-8

FIGURE 2-5. Recommended Reset Connections, Memory-Managed System

2.0 Functional Description (Continued)

Status codes that are relevant to the MMU's function during a memory reference are:

1000, 1001 Instruction Fetch status, used by the debugging features to distinguish between data and instruction references.

1010 Data Transfer. A data value is to be transferred.

1011 Read RMW Operand. Although this is always a Read cycle, the MMU treats it as a Write cycle for purposes of protection and breakpointing.

1100 Read for effective address. Data used for address calculation is being transferred.

All other status codes are treated as data accesses if they occur in conjunction with a pulse on the $\overline{ADS}$ pin. Note that these include Interrupt Acknowledge and End of Interrupt cycles performed by the CPU. The status codes 1101, 1110 and 1111 are also recognized by the MMU in conjunction with pulses on the $\overline{SPC}$ line while it is executing Slave Processor instructions, but these do not occur in a context relevant to address translation.

TABLE 2-1. ST0–ST3 Encodings (ST0 Is the Least Significant)

0000 — Idle: CPU Inactive on Bus
0001 — Idle: WAIT Instruction
0010 — (Reserved)
0011 — Idle: Waiting for Slave
0100 — Interrupt Acknowledge, Master
0101 — Interrupt Acknowledge, Cascaded
0110 — End of Interrupt, Master
0111 — End of Interrupt, Cascaded
1000 — Sequential Instruction Fetch
1001 — Non-Sequential Instruction Fetch
1010 — Data Transfer
1011 — Read Read-Modify-Write Operand
1100 — Read for Effective Address
1101 — Transfer Slave Operand
1110 — Read Slave Status Word
1111 — Broadcast Slave ID

The $\overline{DDIN}$ line indicates the direction of the transfer: 0 = Read, 1 = Write.

$\overline{DDIN}$ is monitored by the MMU during CPU cycles to detect write operations, and is driven by the MMU during MMU-initiated bus cycles.

The U/$\overline{S}$ pin indicates the privilege level at which the CPU is making the access: 0 = Supervisor Mode, 1 = User Mode. It is used by the MMU to select the address space for translation and to perform protection level checking. Normally, the U/$\overline{S}$ pin is a direct reflection of the U bit in the CPU's Processor Status Register (PSR). The MOVUS and MOVSU CPU instructions, however, toggle this pin on successive operand accesses in order to move data between virtual spaces.

The MMU uses the $\overline{FLT}$ line to take control of the bus from the CPU. It does so as necessary for updating its internal TLB from the Page Tables in memory, for maintaining the contents of the status bits (R and M) in the Page Table Entries, and for implementing bus timing adjustments needed by the debugging features.

The MMU also aborts invalid accesses attempted by the CPU. This is done by pulsing the $\overline{RST}/\overline{ABT}$ pin low for one clock period. (A pulse longer than one clock period is interpreted by the CPU as a Reset command).

Because the MMU performs only 16-bit transfers, some additional circuitry is needed to interface it to the 32-bit data bus of an NS32032-based system. However, since the MMU never writes to the most-significant word of a Page Table Entry, the only special requirement is that it must be able to read from the top half of the bus. This can be accomplished as shown in *Figure A-2* (Appendix A) by using a 16-bit unidirectional buffer and some gating circuitry that enables it whenever an MMU-initiated bus cycle accesses an address ending in binary "10".

The bus connections required in conjunction with the NS32332 CPU are somewhat more complex (see the NS32332 data sheet), but the sequences of events documented here still hold.

2.4.2 CPU-Initiated Bus Cycles

A CPU-initiated bus cycle is performed in a minimum of five clock cycles (four in the case of the NS32332): T1, TMMU, T2, T3 and T4, as shown in *Figure 2-6*.

During period T1, the CPU places the virtual address to be translated on the bus, and the MMU latches it internally and begins translation. The MMU also samples the $\overline{DDIN}$ pin, the status lines ST0–ST3, and the U/$\overline{S}$ pin to determine how the CPU intends to use the bus.

During period TMMU the CPU floats its bus drivers and the MMU takes one of three actions:

1) If the translation for the virtual address is resident in the MMU's TLB, and the access being attempted by the CPU does not violate the protection level of the page being referenced, the MMU presents the translated address and generates a $\overline{PAV}$ pulse to trigger a bus cycle in the rest of the system. See *Figure 2-6*.
2) If the translation for the virtual address is resident in the MMU's TLB, but the access being attempted by the CPU is not allowed due to the protection level of the page being referenced, the MMU generates a pulse on the $\overline{RST}/\overline{ABT}$ pin to abort the CPU's access. No $\overline{PAV}$ pulse is generated. See *Figure 2-7*.
3) If the translation for the virtual address is not resident in the TLB, or if the CPU is writing to a page whose M bit is not yet set, the MMU takes control of the bus asserting the $\overline{FLT}$ signal as shown in *Figure 2-8*. This causes the CPU to float its bus and wait. The MMU then initiates a sequence of bus cycles as described in Section 2.4.3.

From state T2 through T4 data is transferred on the bus between the CPU and memory, and the TCU provides the strobes for the transfer. During this time the MMU floats

2.0 Functional Description (Continued)

pins AD0–AD15, and handles pins A16–A24 according to the mode of operation (16-bit or 32-bit) selected during reset (Section 2.3).

In 16-bit bus mode, the MMU drives address lines A16–A24 from TMMU through T4 and they need not be latched externally. This is appropriate for the NS32016 CPU, which uses only AD0–AD15 for data transfers. In 32-bit bus mode, the MMU asserts the physical address on pins A16–A24 only during TMMU, and floats them from T2 through T4 because the CPU uses them for data transfer. In this case the physical address presented on these lines must be latched externally using $\overline{PAV}$.

Whenever the MMU generates an Abort pulse on the $\overline{RST}/\overline{ABT}$ pin, the CPU enters state T2 and then Ti (idle), ending the bus cycle. Since no $\overline{PAV}$ pulse is issued by the MMU, the rest of the system remains unaware that an access has been attempted. The MMU requires that no further memory references be attempted by the CPU for at least two clock cycles after the T2 state, as shown in *Figure 2-7*. This requirement is met by all Series 32000 CPU's. During this time, the RDY line must remain high. This requirement is met by the NS32201 TCU.

2.4.3 MMU-Initiated Cycles

Bus cycles initiated by the MMU are always nested within CPU-initiated bus cycles; that is, they appear after the MMU has accepted a virtual address from the CPU and has set the $\overline{FLT}$ line active. The MMU will initiate memory cycles in the following cases:

1) There is no translation in the MMU's TLB for the virtual address issued by the CPU, meaning that the MMU must reference the Page Tables in memory to obtain the translation.
2) There is a translation for that virtual address in the TLB, but the page is being written for the first time (the M bit in its Level-2 Page Table Entry is 0). The MMU treats this case as if there were no translation in the TLB, and performs a Page Table lookup in order to set the M bit in the Level-2 Page Table Entry as well as in the TLB.

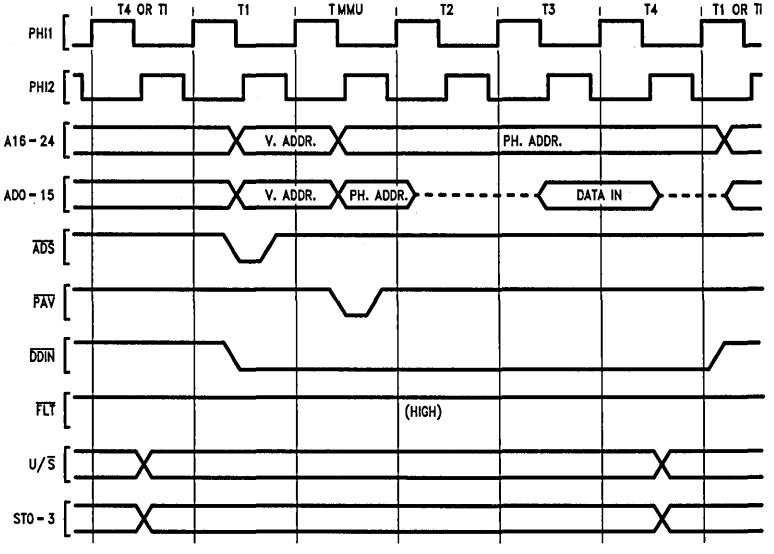
Having made the necessary memory references, the MMU either aborts the CPU access or it provides the translated address and allows the CPU's access to continue to T2.

Figure 2-8 shows the sequence of events in a Page Table lookup. After asserting $\overline{FLT}$, the MMU waits for one additional clock cycle, then reads the Level-1 Page Table Entry and the Level-2 Page Table Entry in four consecutive memory Read cycles. Note that the MMU performs two 16-bit transfers to read each Page Table Entry, regardless of the width of the CPU's data bus. There are no idle clock cycles between MMU-initiated bus cycles unless a bus request is made on the $\overline{HOLD}$ line (Section 2.6).

During the Page Table lookup the MMU drives the $\overline{DDIN}$ signal. The status lines ST0–ST3 and the U/$\overline{S}$ pin are not released by the CPU, and retain their original settings while the MMU uses the bus. The Byte Enable signals from the CPU ($\overline{HBE}$ in 16-bit systems, $\overline{BE0}$–$\overline{BE3}$ in 32-bit systems) should in general be handled externally for correct memory referencing. (The current NS32016 CPU does, however, handle $\overline{HBE}$ in a manner that is acceptable in many systems at clock rates of 12.5 MHz or less.)

In the clock cycle immediately after T4 of the last lookup cycle, the MMU removes the $\overline{FLT}$ signal, issues the translated address, and pulses $\overline{PAV}$ to continue the CPU's access.

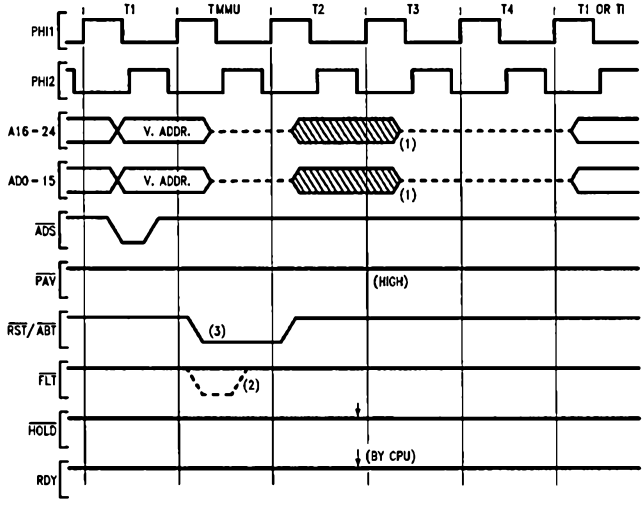
Note that when the MMU sets $\overline{FLT}$ active, the clock cycle originally called TMMU is redesignated Tf. Clock cycles in which the $\overline{PAV}$ pulse occurs are designated TMMU.



TL/EE/8692-9

FIGURE 2-6. CPU Read Cycle; Translation in TLB (TLB Hit)

2.0 Functional Description (Continued)



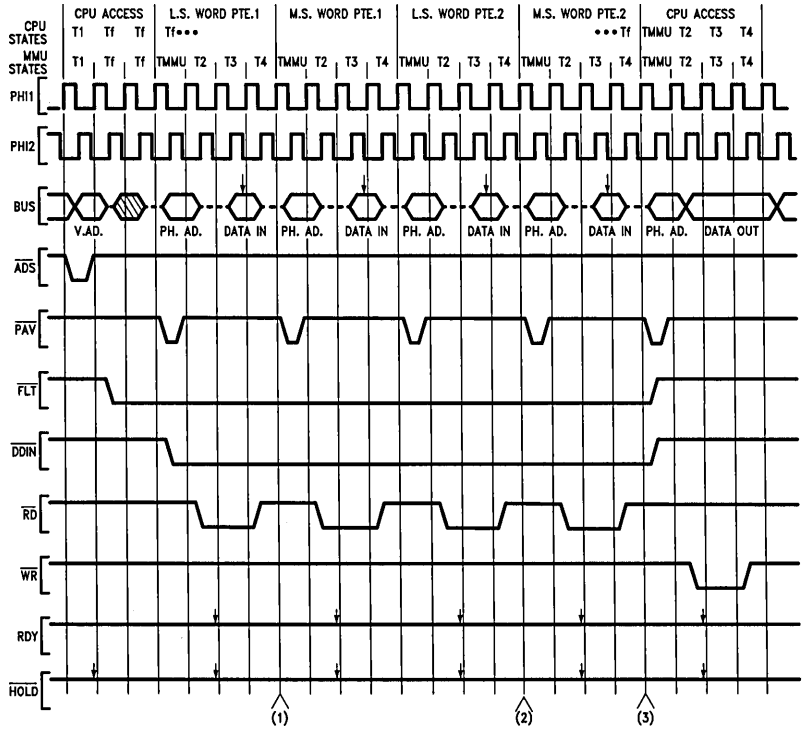
TL/EE/8692-10

Note 1: The CPU drives the bus if a write cycle is aborted.

Note 2: FLT may be pulsed if a breakpoint on physical address is enabled or an execution breakpoint is triggered.

Note 3: If this bus cycle is a write cycle to a write-protected page, FLT is asserted for two clock cycles and the abort pulse is delayed by one clock cycle.

FIGURE 2-7. Abort Resulting from Protection Violation; Translation in TLB



TL/EE/8692-11

Note 1: If the R bit on the Level-1 PTE must be set, a write cycle is inserted here.

Note 2: If either the R or the M bit on the Level-2 PTE must be set, a write cycle is inserted here.

Note 3: If a breakpoint on physical address is enabled, an extra clock cycle is inserted here.

FIGURE 2-8. Page Table Lookup

2.0 Functional Description (Continued)

The Page Table Entries are read starting with the low-order word. If the V bit (bit 0) of the low-order word is zero, or the protection level field PL (bits 1 and 2) indicates that the CPU's attempted access is illegal, the MMU does not generate any further memory cycles, but instead issues an Abort pulse during the clock cycle after T4 and removes the $\overline{FLT}$ signal. The CPU continues to T2 and then becomes idle on the bus, as shown in *Figure 2-9*.

If the R and/or M bit (bit 3 or 4) of the low-order word must be updated, the MMU does this immediately in a single Write cycle, before reading the high-order word of the Page Table Entry. All bits except those updated are rewritten with their original values.

At most, the MMU writes two 16-bit words to memory during a translation: the first to the Level-1 table to update the R bit, and the second to the Level-2 table to update the R and/or M bits.

2.4.4 Cycle Extension

To allow sufficient strobe widths and access time requirements for any speed of memory or peripheral device, the NS32082 provides for extension of a bus cycle. Any type of bus cycle, CPU-initiated or MMU-initiated, can be extended, except Slave Processor cycles, which are not memory or peripheral references.

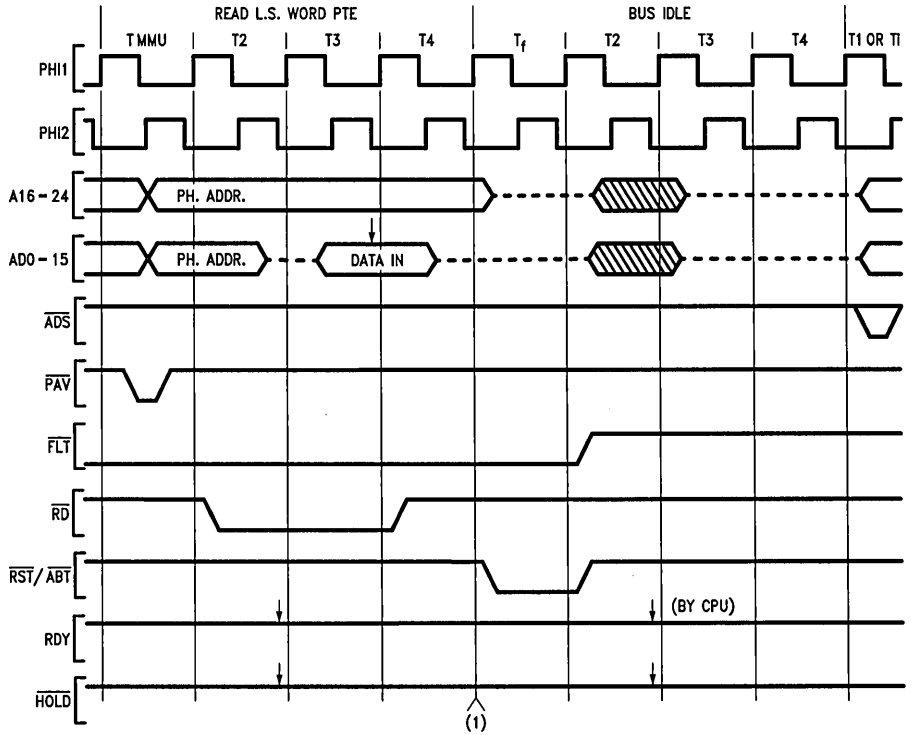
In *Figures 2-6* and *2-8*, note that during T3 all bus control signals are flat. Therefore, a bus cycle can be cleanly extended by causing the T3 state to be repeated. This is the purpose of the RDY (Ready) pin.

Immediately before T3 begins, on the falling edge of clock phase PHI2, the RDY line is sampled by the CPU and/or the MMU. If RDY is high, the next state after T3 will be T4, ending the bus cycle. If it is low, the next state after T3 will be another T3 and the RDY line will be sampled again. RDY is sampled in each following clock period, with insertion of additional T3 states, until it is sampled high. Each additional T3 state inserted is called a "WAIT state."

During CPU bus cycles, the MMU monitors the RDY pin only if the 16-bit mode is selected. This is necessary since the MMU drives the address lines A16–A24, and needs to detect the end of the bus cycle in order to float them.

If the 32-bit mode is selected, the above address lines are floated following the TMMU state. The MMU will be ready to perform another translation after three clock cycles, and the RDY line is ignored.

The RDY pin is driven by the NS32201 Timing Control Unit, which applies WAIT states to the CPU and MMU as requested on its own WAIT request input pins.



TL/EE/8692-12

Note 1: If a breakpoint on physical address is enabled, an extra clock cycle is inserted here.

FIGURE 2-9. Abort Resulting after a Page Table Lookup

2.0 Functional Description (Continued)

2.5 SLAVE PROCESSOR INTERFACE

The CPU and MMU execute four instructions cooperatively. These are LMR, SMR, RDVAL and WRVAL, as described in Section 2.5.2. The MMU takes the role of a Slave Processor in executing these instructions, accepting them as they are issued to it by the CPU. The CPU calculates all effective addresses and performs all operand transfers to and from memory and the MMU. The MMU does not take control of the bus except as necessary in normal operation; i.e., to translate and validate memory addresses as they are presented by the CPU.

The sequence of transfers ("protocol") followed by the CPU and MMU involves a special type of bus cycle performed by the CPU. This "Slave Processor" bus cycle does not involve the issuing of an address, but rather performs a fast data transfer whose purpose is pre-determined by the form of the instruction under execution and by status codes asserted by the CPU.

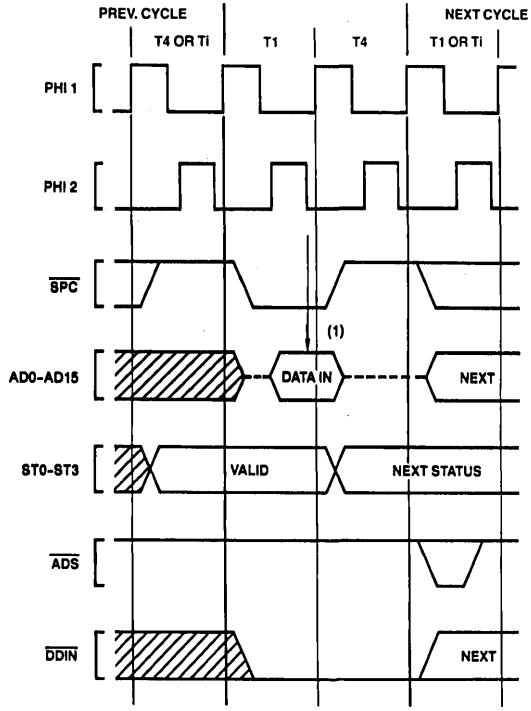
2.5.1 Slave Processor Bus Cycles

The interconnections between the CPU and MMU for Slave Processor communication are shown in *Figures A-1* and *A-2* (Appendix A). The low-order 16 bits of the bus are used for data transfers. The $\overline{SPC}$ signal is bidirectional. It is pulsed by the CPU as a low-active data strobe for Slave Processor transfers, and is also pulsed low by the MMU to acknowledge, when necessary, that it is ready to continue execution of an MMU instruction. Since $\overline{SPC}$ is normally in a high-impedance state, it must be pulled high with a 10 kΩ resistor, as shown. The MMU also monitors the status lines ST0–ST3 to follow the protocol for the instruction being executed.

Data is transferred between the CPU and the MMU with Slave Processor bus cycles, illustrated in *Figures 2-10* and *2-11*. Each bus cycle transfers one byte or one word (16 bits) to or from the MMU.

Slave Processor bus cycles are performed by the CPU in two clock periods, which are labeled T1 and T4. During T1, the CPU activates $\overline{SPC}$ and, if it is writing to the MMU, it presents data on the bus. During T4, the CPU deactivates $\overline{SPC}$ and, if it is reading from the MMU, it latches data from the bus. The CPU guarantees that data written to the MMU is held through T4 to provide for the MMU's hold time requirements. The CPU also guarantees that the status code on ST0–ST3 becomes valid, at the latest, during the clock period preceding T1. The status code changes during T4 to anticipate the next bus cycle, if any.

Note that Slave Processor bus cycles are never extended with WAIT states. The RDY line is not sampled.

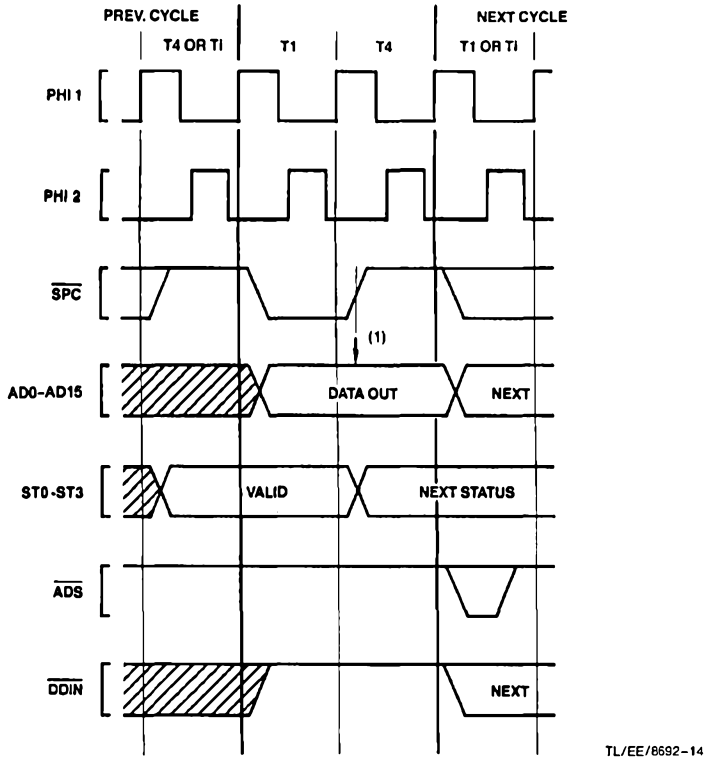


TL/EE/8692–13

Note 1: CPU samples Data Bus here.

FIGURE 2-10. Slave Access Timing; CPU Reading from MMU

2.0 Functional Description (Continued)



Note 1: MMU samples Data Bus here.

FIGURE 2-11. Slave Access Timing; CPU Writing to MMU

2.5.2 Instruction Protocols

MMU instructions have a three-byte Basic Instruction field consisting of an ID byte followed by an Operation Word. See *Figure 3-10* for the MMU instruction encodings. The ID Byte has three functions:

1) It identifies the instruction as being a Slave Processor instruction.
2) It specifies that the MMU will execute it.
3) It determines the format of the following Operation Word of the instruction.

The CPU initiates an MMU instruction by issuing first the ID Byte and then the Operation Word, using Slave Processor bus cycles. The ID Byte is sent on the least-significant byte of the bus, in conjunction with status code 1111 (Broadcast ID Byte). The Operation Word is sent on the entire 16-bit data bus, with status code 1101 (Transfer Operation Word / Operand). The Operation Word is sent with its bytes swapped; i.e., its least-significant byte is presented to the MMU on the most-significant half of the 16-bit bus.

Other actions are taken by the CPU and the MMU according to the instruction under execution, as shown in Tables 2-2, 2-3 and 2-4.

In executing the LMR instruction (Load MMU Register, Table 2-2), the CPU issues the ID Byte, the Operation Word, and then the operand value to be loaded by the MMU. The register to be loaded is specified in a field within the Operation Word of the instruction.

In executing the SMR instruction (Store MMU Register, Table 2-3), the CPU also issues the ID Byte and the Operation Word of the instruction to the MMU. It then waits for the MMU to signal (by pulsing $\overline{SPC}$ low) that it is ready to present the specified register's contents to the CPU. Upon receiving this "Done" pulse, the CPU reads first a "Status Word" (dictated by the protocol for Slave Processor instructions) which the MMU provides as a word of all zeroes. The CPU then reads the contents of the selected register in two successive Slave Processor bus cycles, and places this result value into the instruction's destination (a CPU general-purpose register or a memory location).

In executing the RDVAL (Read-Validate) or WRVAL (Write-Validate) instruction, the CPU again issues the ID Byte and the Operation Word to the MMU. However, its next action is to initiate a one-byte Read cycle from the memory address whose protection level is being tested. It does so while presenting status code 1010; this being the only place that this status code appears during a RDVAL or WRVAL instruction. This memory access triggers a special address translation from the MMU. The translation is performed by the MMU using User-Mode mapping, and any protection violation occurring during this memory cycle does not cause an Abort. The MMU will, however, abort the CPU if the Level-1 Page Table Entry is invalid.

Upon completion of the address translation, the MMU pulses $\overline{SPC}$ to acknowledge that the instruction may continue execution.

2.0 Functional Description (Continued)

TABLE 2-2. LMR Instruction Protocol

CPU Action	Status	MMU Action
Issues ID Byte of instruction, pulsing $\overline{SPC}$.	1111	Accepts ID Byte.
Sends Operation Word of Instruction, pulsing $\overline{SPC}$.	1101	Decodes instruction.
Issues low-order word of new register value to MMU, pulsing $\overline{SPC}$.	1101	Accepts word from bus; places it into low-order half of referenced MMU register.
Issues high-order word of new register value to MMU, pulsing $\overline{SPC}$.	1101	Accepts word from bus; places it into high-order half of referenced MMU register.

TABLE 2-3. SMR Instruction Protocol

CPU Action	Status	MMU Action
Issues ID Byte of Instruction, pulsing $\overline{SPC}$.	1111	Accepts ID Byte.
Sends Operation Word of instruction, pulsing $\overline{SPC}$.	1101	Decodes instruction.
Waits for Done pulse from MMU.	xxxx	Sends Done pulse on $\overline{SPC}$.
Pulses $\overline{SPC}$ and reads Status Word from MMU.	1110	Presents Status Word (all zeroes) on bus.
Pulses $\overline{SPC}$, reading low-order word of result from MMU.	1101	Presents low-order word of referenced MMU register on bus.
Pulses $\overline{SPC}$, reading high-order word of result from MMU.	1101	Presents high-order word of referenced MMU register on bus.

TABLE 2-4. RDVAL/WRVAL Instruction Protocol

CPU Action	Status	MMU Action
Issues ID Byte of instruction, pulsing $\overline{SPC}$.	1111	Accepts ID Byte.
Sends Operation Word of instruction, pulsing $\overline{SPC}$.	1101	Decodes instruction.
Performs dummy one-byte memory read from operand's location.	1010	Translates CPU's address, using User-Mode mapping, and performs requested test on the address presented by the CPU. Aborts the CPU if the level-1 page table entry is invalid. Starts a Memory Cycle from the Translated Address if the translation is successful. Aborts on protection violations are temporarily suppressed.
Waits for Done pulse from MMU	xxxx	Sends Done pulse on $\overline{SPC}$.
Sends $\overline{SPC}$ pulse and reads Status Word from MMU; places bit 5 of this word into the F bit of the PSR register.	1110	Presents Status Word on bus, indicating in bit 5 the result of the test.

If the translation is successful the MMU will also start a dummy memory cycle from the translated address. See *Figure 2-12*. Note that, during this time the CPU will monitor the RDY line. Therefore, for proper operation, the RDY line must be kept high if the memory cycle is not performed.

The CPU then reads from the MMU a Status Word. Bit 5 of this Status Word indicates the result of the instruction:

0 if the CPU in User Mode could have made the corresponding access to the operand at the specified address (Read in RDVAL, Write in WRVAL),

1 if the CPU would have been aborted for a protection violation.

Bit 5 of the Status Word is placed by the CPU into the F bit of the PSR register, where it can be tested by subsequent instructions as a condition code.

Note: The MMU sets the R bit on RDVAL; R and M bits on WRVAL.

2.6 BUS ACCESS CONTROL

The NS32082 MMU has the capability of relinquishing its access to the bus upon rquest from a DMA device. It does this by using $\overline{HOLD}$, $\overline{HLDAI}$ and $\overline{HLDAO}$.

Details on the interconnections of these pins are provided in *Figures A-1* and *A-2* (Appendix A).

Requests for DMA are presented in parallel to both the CPU and MMU on the $\overline{HOLD}$ pin of each. The component that currently controls the bus then activates its Hold Acknowledge output to grant bus access to the requesting device. When the CPU grants the bus, the MMU passes the CPU's $\overline{HLDA}$ signal to its own $\overline{HLDAO}$ pin. When the MMU grants the bus, it does so by activating its $\overline{HLDAO}$ pin directly, and the CPU is not involved. $\overline{HLDAI}$ in this case is ignored.

Refer to *Figures 4-14, 4-15* and *4-16* for details on bus granting sequences.

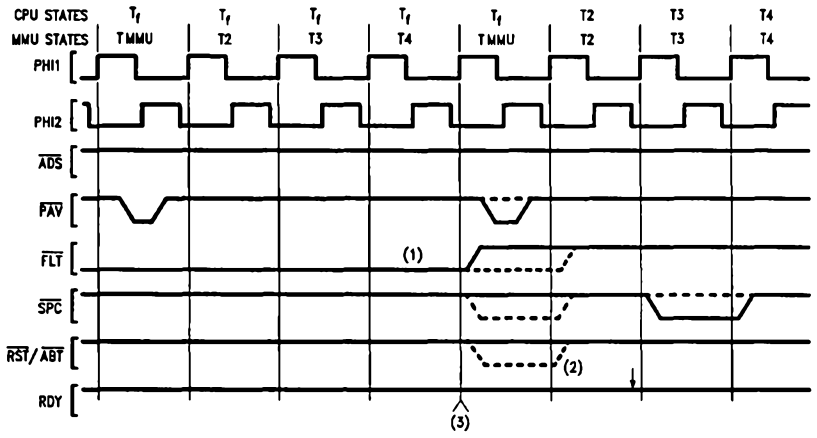
2.7 BREAKPOINTING

The MMU provides the ability to monitor references to two memory locations in real time, generating a Breakpoint trap on occurrence of any specified type of reference to either location made by a program. In addition, a Breakpoint trap may be inhibited until a specified number of such references have been performed.

Breakpoint monitoring is enabled and regulated by the setting of appropriate bits in the MSR and BPR0-1 registers. See Sections 3.5 and 3.7.

A Breakpoint trap is signalled to the CPU as either a Non-Maskable Interrupt or an Abort trap, depending on the setting of the AI bit in the MSR register.

2.0 Functional Description (Continued)



TL/EE/8692-15

Note 1: $\overline{FLT}$ is asserted if the translation is not in the TLB or a WRVAL instruction is executed and the M Bit is not set.

Note 2: If the Level-1 PTE is not valid, an abort is generated, $\overline{SPC}$ is issued in TMMU and $\overline{FLT}$ is deasserted in T_2.

Note 3: If a protection violation occurs or the Level-2 PTE is invalid, an Idle State is inserted here, $\overline{PAV}$ is not pulsed and $\overline{SPC}$ is pulsed during this Idle State.

FIGURE 2-12. $\overline{FLT}$ Deassertion During RDVAL/WRVAL Execution

The MSR register also indicates which breakpoint register triggered the break, and the direction (read or write) and type of memory cycle that was detected. The breakpoint address is not placed into the EIA register, as this register holds the addresses of address translation errors only. The breakpoint address is, however, available in the indicated Breakpoint register.

On occurrence of any trap generated by the MMU, including the Breakpoint trap, the BEN bit in the MSR register is immediately cleared, disabling any further Breakpoint traps.

Enabling breakpoints may cause variations in the bus timing given in the previous sections. Specifically:

1) While either breakpoint is enabled to monitor physical addresses, the MMU inserts an additional clock period into all bus cycles by asserting the $\overline{FLT}$ line for one clock. See *Figure 2-13.*

2) If the CPU initiates an instruction prefetch from a location at which a breakpoint is enabled on Execution, the MMU asserts the $\overline{FLT}$ line to the CPU, performs the memory cycle itself, and issues an edited instruction word to the CPU. See *Figure 2-14* and Section 2.7.1.

Note: Instructions which use two operands, a read-type and a write-type (e.g., MOVD 0(r1).0(r2), with the first operand valid and protected to allow user reads, and the second operand either invalid (page fault) or write protected, cause a read-type break event to occur for the first operand regardless of the outcome of the instruction. Each time the instruction is retried, the read-event is recorded. Hence, the breakpoint count register may reflect a different count than a casual assumption would lead one to. The same effect can occur on a RMW type operand with read only protection.

2.7.1 Breakpoints on Execution

The Series 32000 CPUs have an instruction prefetch which requires synchronization with execution breakpoints. In consideration of this, the MMU only issues an execution breakpoint when an instruction is prefetched with a nonsequential status code and the conditions specified in a breakpoint register are met. This guarantees that the instruction prefetch queue is empty and there are not pending instructions in the pipeline. There are three cases to consider:

Case 1: A nonsequential instruction prefetch is made to a breakpointed address.

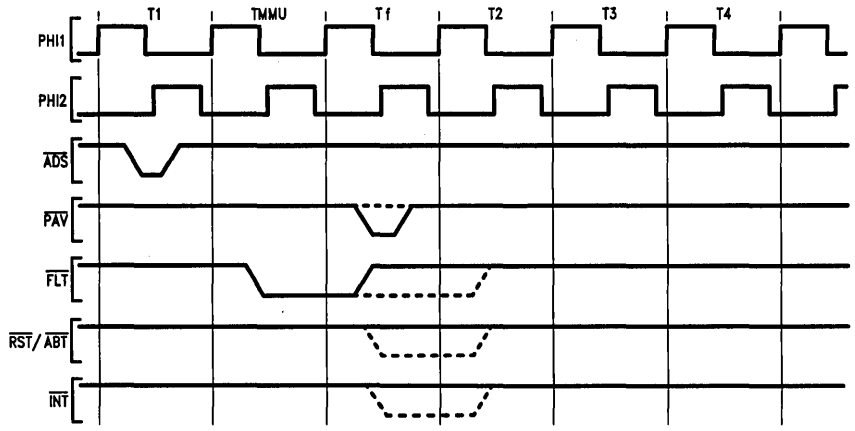
Response: The queue is necessarily empty. The breakpoint is issued.

Case 2, 3: A sequential prefetch is made to a breakpointed address OR a prefetch is made to an even address and the breakpoint is on the next odd address.

Response: In these cases, there may be instructions pending in the queue which must finish before the breakpoint is fired. Instead of putting the opcode byte (the one specified by the breakpointed address) in the queue, a DIA instruction is substituted for it. DIA is a single byte instruction which branches to itself, causing a queue flush. When the DIA executes, the breakpoint address is again issued, this time with nonsequential fetch status and the problem is reduced to case 1.

Note: Execution breakpoints cannot be used when the MMU is connected to either an NS32032 or an NS32332 CPU.

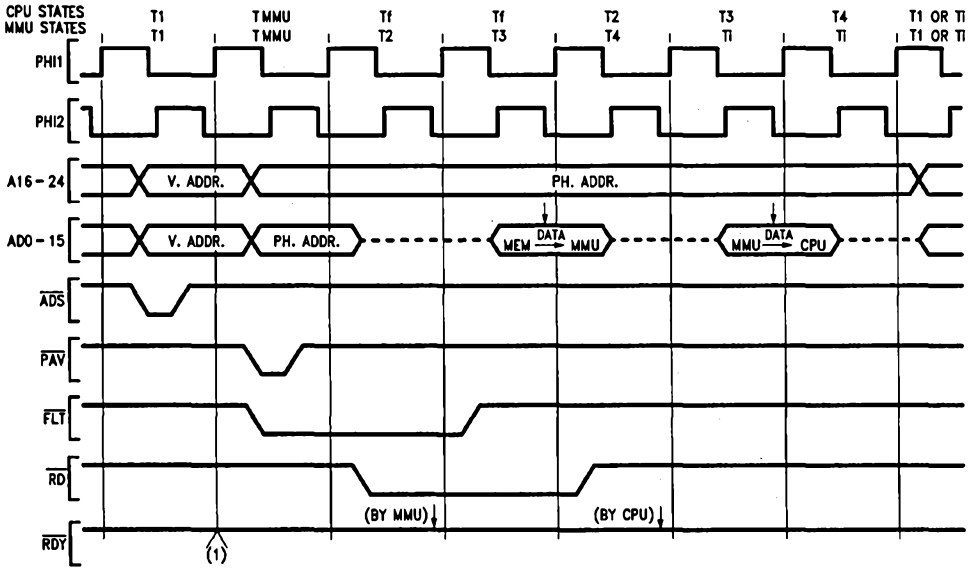
2.0 Functional Description (Continued)



TL/EE/8692-16

Note: If a breakpoint condition is met and abort on breakpoint is enabled, the bus cycle is aborted. In this case $\overline{FLT}$ is stretched by one clock cycle.

FIGURE 2-13. Bus Timing with Breakpoint on Physical Address Enabled



TL/EE/8692-17

Note 1: If a breakpoint on physical address is enabled, an extra clock cycle is inserted here.

FIGURE 2-14. Execution Breakpoint Timing; Insertion of DIA Instruction

3.0 Architectural Description

3.1 PROGRAMMING MODEL

The MMU contains a set of registers through which the CPU controls and monitors management and debugging functions. These registers are not memory-mapped. They are examined and modified by executing the Slave Processor instructions LMR (Load Memory Management Register) and SMR (Store Memory Management Register). These instructions are explained in Section 3.11, along with the other Slave Processor instructions executed by the MMU.

A brief description of the MMU registers is provided below. Details on their formats and functions are provided in the following sections.

PTB0, PTB1—Page Table Base Registers. They hold the physical memory addresses of the Page Tables referenced by the MMU for address translation. See Section 3.3.

EIA—Error/Invalidate Register. Dual-function register, used to display error addresses and also to purge cached translation information from the TLB. See Section 3.4.

BPR0, BPR1—Breakpoint Registers. Specify the conditions under which a breakpoint trap is generated. See Section 3.5.

BCNT—Breakpoint Counter Register. 24-bit counter used to count BPR0 events. Allows the breakpoint trap from the BPR0 register to be inhibited until a specified number of events have occurred. See Section 3.6.

MSR—Memory Management Status Register. Contains basic control and status fields for all MMU functions. See Section 3.7.

3.2 MEMORY MANAGEMENT FUNCTIONS

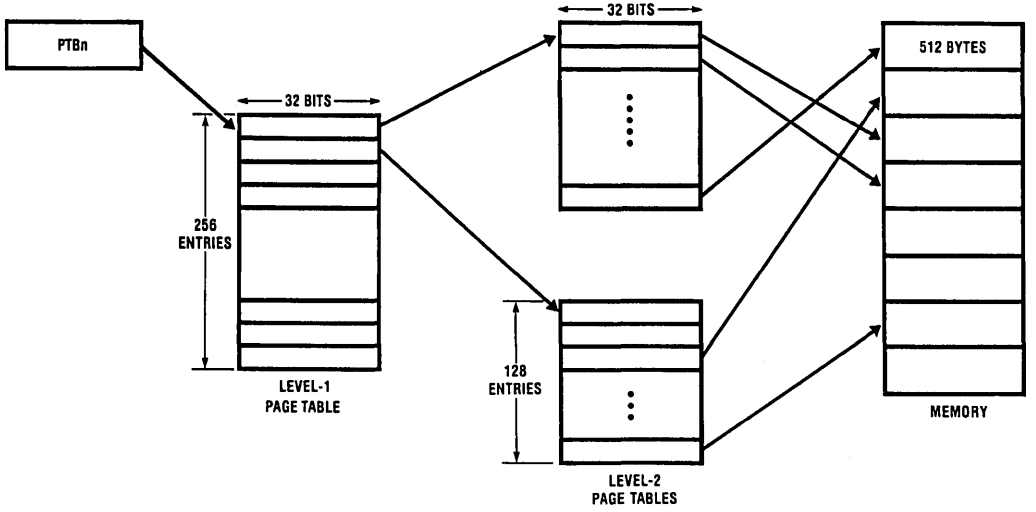
The NS32082 uses sets of tables in physical memory (the "Page Tables") to define the mapping from virtual to physical addresses. These tables are found by the MMU using one of its two Page Table Base registers: PTB0 or PTB1. Which register is used depends on the currently selected address space. See Section 3.2.2.

3.2.1. Page Table Structure

The page tables are arranged in a two-level structure, as shown in *Figure 3-1*. Each of the MMU's PTBn registers may point to a Level-1 page table. Each entry of the Level-1 page table may in turn point to a Level-2 page table. Each Level-2 page table entry contains translation information for one page of the virtual space.

The Level-1 page table must remain in physical memory while the PTBn register contains its address and translation is enabled. Level-2 Page Tables need not reside in physical memory permanently, but may be swapped into physical memory on demand as is done with the pages of the virtual space.

The Level-1 Page Table contains 256 32-bit Page Table Entries (PTE'S) and therefore occupies 1 Kbyte. Each entry of the Level-1 Page Table contains fields used to construct the physical base address of a Level-2 Page Table. These fields are a 15-bit PFN field, providing bits 9-23 of the physical address, and an MS bit providing bit 24. The remaining bits (0–8) are assumed zero, placing a Level-2 Page Table always on a 512-byte (page) boundary.



TL/EE/8692-18

FIGURE 3-1. Two-Level Page Tables

3.0 Architectural Description (Continued)

Level-2 Page Tables contain 128 32-bit Page Table entries, and so occupy 512 bytes (1 page). Each Level-2 Page Table Entry points to a final 512-byte physical page frame. In other words, its PFN and MS fields provide the Page Frame Number portion (bits 9-24) of the translated address (*Figure 3-3*). The OFFSET field of the translated address is taken directly from the corresponding field of the virtual address.

3.2.2 Virtual Address Spaces

When the Dual Space option is selected for address translation in the MSR (Sec. 3.7) the MMU uses two maps: one for translating addresses presented to it in Supervisor Mode and another for User Mode addresses. Each map is referenced by the MMU using one of the two Page Table Base registers: PTB0 or PTB1. The MMU determines the CPU's current mode by monitoring the state of the U/$\overline{S}$ pin and applying the following rules.

1) While the CPU is in Supervisor Mode (U/$\overline{S}$ pin = 0), the CPU is said to be presenting addresses belonging to Address Space 0, and the MMU uses the PTB0 register as its reference for looking up translations from memory.
2) While the CPU is in User Mode (U/$\overline{S}$ pin = 1), and the MSR DS bit is set to enable Dual Space translation, the CPU is said to be presenting addresses belonging to Address Space 1, and the MMU uses the PTB1 register to look up translations.
3) If Dual Space translation is not selected in the MSR, there is no Address Space 1, and all addresses presented in both Supervisor and User modes are considered by the MMU to be in Address Space 0. The privilege level of the CPU is used then only for access level checking.

Note: When the CPU executes a Dual-Space Move instruction (MOVUSi or MOVSUi), it temporarily enters User Mode by switching the state of the U/$\overline{S}$ pin. Accesses made by the CPU during this time are treated by the MMU as User-Mode accesses for both mapping and access level checking. It is possible, however, to force the MMU to assume Supervisor-Mode privilege on such accesses by setting the Access Override (AO) bit in the MSR (Sec. 3.7).

3.2.3 Page Table Entry Formats

Figure 3-2 shows the formats of Level-1 and Level-2 Page Table Entries (PTE's). Their formats are identical except for the "M" bit, which appears only in a Level-2 PTE.

The bits are defined as follows:

V Valid. The V bit is set and cleared only by software.

V = 1 => The PTE is valid and may be used for translation by the MMU.

V = 0 => The PTE does not represent a valid translation. Any attempt to use this PTE will cause the MMU to generate an Abort trap. While V = 0, the operating system may use all other bits except the PL field for any desired function.

PL Protection Level. This two-bit field establishes the types of accesses permitted for the page in both User Mode and Supervisor Mode, as shown in Table 3-1.

The PL field is modified only by software. In a Level-1 PTE, it limits the maximum access level allowed for all pages mapped through that PTE.

TABLE 3-1. Access Protection Levels

Mode	U/$\overline{S}$	Protection Level Bits (PL)			
		00	01	10	11
User	1	no access	no access	read only	full access
Supervisor	0	read only	full access	full access	full access

R Referenced. This is a status bit, set by the MMU and cleared by the operating system, that indicates whether the page mapped by this PTE has been referenced within a period of time determined by the operating system. It is intended to assist in implementing memory allocation strategies. In a Level-1 PTE, the R bit indicates only that the Level-2 Page Table has been referenced for a translation, without necessarily implying that the translation was successful. In a Level-2 PTE, it indicates that the page mapped by the PTE has been successfully referenced.

R = 1 => The page has been referenced since the R bit was last cleared.

R = 0 => The page has not been referenced since the R bit was last cleared.

Note: The RDVAL and WRVAL instructions set the Level-1 and Level-2 bits for the page whose protection level is tested. See Sections 2.5.2 and 3.11.

M Modified. This is a status bit, set by the MMU whenever a write cycle is successfully performed to the page mapped by this PTE. It is initialized to zero by the operating system when the page is brought into physical memory.

M = 1 => The page has been modified since it was last brought into physical memory.

M = 0 => The page has not been modified since it was last brought into physical memory.

In Level-1 Page Table Entries, this bit position is undefined, and is altered in an undefined manner by the MMU while the V bit is 1.

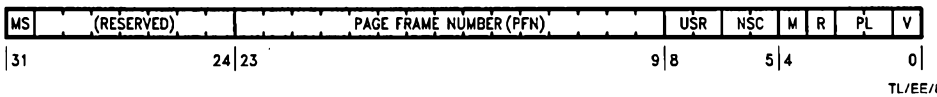
Note: The WRVAL instruction sets the M bit for the page whose protection level is tested. See Sections 2.5.2 and 3.11.

NSC Reserved. These bits are ignored by the MMU and their values are not changed.

They are reserved by National, and therefore should not be used by the user software.

USR User bits. These bits are ignored by the MMU and their values are not changed.

They can be used by the user software.



TL/EE/8692-19

FIGURE 3-2. A Page Table Entry

3.0 Architectural Description (Continued)

PFN Page Frame Number. This 15-bit field provides bits 9-23 of the Page Frame Number of the physical address. See *Figure 3-3*.

MS Memory System. This bit represents the most significant bit of the physical address, and is presented by the MMU on pin A24. This bit is treated by the MMU no differently than any other physical address bit, and can be used to implement a 32-Mbyte physical addressing space if desired.

3.2.4 Physical Address Generation

When a virtual address is presented to the MMU by the CPU and the translation information is not in the TLB, the MMU performs a page table lookup in order to generate the physical address.

The Page Table structure is traversed by the MMU using fields taken from the virtual address. This sequence is diagrammed in *Figure 3-3*.

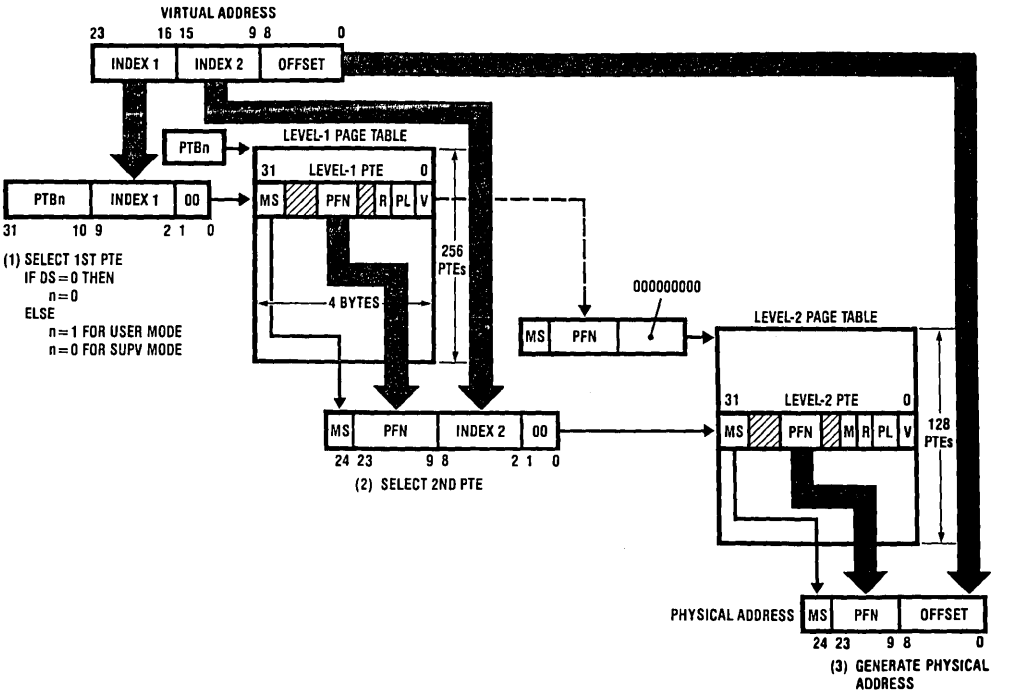
Bits 9–23 of the virtual address hold the 15-bit Page Number, which in the course of the translation is replaced with the 16-bit Page Frame Number of the physical address. The virtual Page Number field is further divided into two fields, INDEX 1 and INDEX 2.

Bits 0–8 constitute the OFFSET field, which identifies a byte's position within the accessed page. Since the byte position within a page does not change with translation, this value is not used, and is simply echoed by the MMU as bits 0–8 of the final physical address.

The 8-bit INDEX 1 field of the virtual address is used as an index into the Level-1 Page Table, selecting one of its 256 entries. The address of the entry is computed by adding INDEX 1 (scaled by 4) to the contents of the current Page Table Base register. The PFN and MS fields of that entry give the base address of the selected Level-2 Page Table.

The INDEX 2 field of the virtual address (7 bits) is used as the index into the Level-2 Page Table, by adding it (scaled by 4) to the base address taken from the Level-1 Page Table Entry. The PFN and MS fields of the selected entry provide the entire Page Frame Number of the translated address.

The offset field of the virtual address is then appended to this frame number to generate the final physical address.



TL/EE/8692-20

FIGURE 3-3. Virtual to Physical Address Translation

3.0 Architectural Description (Continued)

3.3 PAGE TABLE BASE REGISTERS (PTB0, PTB1)

The PTBn registers hold the physical addresses of the Level-1 Page Tables.

The format of these registers is shown in *Figure 3-4*. The least-significant 10 bits are permanently zero, so that each register always points to a 1 Kbyte boundary in memory.

The PTBn registers may be loaded or stored using the MMU Slave Processor instructions LMR and SMR (Section 3.11).

3.4 ERROR/INVALIDATE ADDRESS REGISTER (EIA)

The Error/Invalidate Address register is a dual-purpose register.

1) When it is read using the SMR instruction, it presents the virtual address which last generated an address translation error.
2) When a virtual address is written into it using the LMR instruction, the translation for that virtual address is purged, if present, from the TLB. This must be done whenever a Page Table Entry has been changed in memory, since the TLB might otherwise contain an incorrect translation value.

The format of the EIA register is shown in *Figure 3-5*. When a translation error occurs, the cause of the error is reported by the MMU in the appropriate fields of the MSR register (Section 3.7). The ADDRESS field of the EIA register holds the virtual address at which the error occurred, and the AS bit indicates the address space that was in use.

In writing a virtual address to the EIA register, the virtual address is specified in the low-order 24 bits, and the AS bit specifies the address space. A TLB entry is purged only if it matches both the ADDRESS and AS fields.

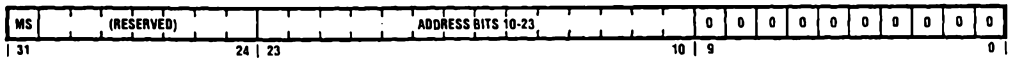
Another technique for purging TLB entries is to load a PTBn register. This automatically purges all entries associated with the addressing space mapped by that register. Turning off translation (clearing the MSR TU and/or TS bits) does not purge any entries from the TLB.

3.5 BREAKPOINT REGISTERS (BPR0, BPR1)

The Breakpoint registers BPR0 and BPR1 specify the addresses and conditions on which a Breakpoint trap will be generated. They are each 32 bits in length and have the format shown in *Figure 3-6*. All implemented bits of BPR0 and BPR1 are readable and writable.

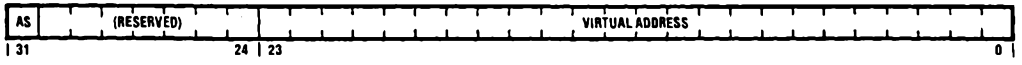
Bits 0 through 23 and bit 31 (AS) specify the breakpoint address. This address may be either virtual or physical, as specified in the VP bit.

Bits 24 and 25 are not implemented. Bit 26 (CE) is not implemented in register BPR1.



TL/EE/8692-21

FIGURE 3-4. Page Table Base Registers (PTB0, PTB1)



TL/EE/8692-22

FIGURE 3-5. EIA Register

TL/EE/8692-23

FIGURE 3-6. Breakpoint Registers (BPR0, BPR1)

3.0 Architectural Description (Continued)

Bits 26 through 30 specify the breakpoint conditions. Breakpoint conditions define how the breakpoint address is compared and which conditions permit a break to be generated. A Breakpoint register can be selectively disabled by setting all of these bits to zero.

AS Address Space. This bit depends on the setting of the VP bit. For virtual addresses, this bit contains the AS (Address Space) qualifier of the virtual address (Section 3.2.2). For physical addresses, this bit contains the MS (Memory System) bit of the physical address.

VP Virtual/Physical. If VP is 0, the breakpoint address is compared against each referenced virtual address. If VP is 1, the breakpoint address is compared against each physical address that is referenced by the CPU (i.e. after translation).

BE Break on Execution. If BE is 1, a break is generated immediately before the instruction at the breakpoint address is executed. While this option is enabled, the breakpoint address must be the address of the first byte of an instruction. If BE is 0, this condition is disabled.

Note: This option cannot be used in systems based on any CPU with a 32-bit wide bus.

The BE bit should only be set when the CPU has a 16-bit bus (i.e. NS32016, NS32C016). In other systems, use instead the BPT instruction placed in memory, to signal a break.

BR Break on Read. If BR is 1, a break is generated when data is read from the breakpoint address. Instruction fetches do not trigger a Read breakpoint. If BR is 0, this condition is disabled.

BW Break on Write. If BW is 1, a break is generated when data is written to the breakpoint address or when data is read from the breakpoint address as the first part of a read-modify-write access. If BW is 0, this condition is disabled.

CE Counter Enable. This bit is implemented only in the BPR0 register. If CE is 1, no break is generated unless the Breakpoint Count register (BCNT, see below) is zero. The BCNT register decrements when the condition for the breakpoint in register BPR0 is met and the BCNT register is not already zero. If CE is 0, the BCNT register is disabled, and breaks from BPR0 occur immediately.

Note 1: The bits BR, BW and CE should not all be set. The counting performed by the MMU becomes inaccurate, and in Abort Mode (MSR AI bit set), it can trap a program in such a way as to make it impossible to retry the breakpointed instruction correctly.

Note 2: An execution breakpoint should not be counted (BE and CE bits both set) if it is placed at an address that is the destination of a branch, or if it follows a queue-flushing instruction. See Table 3-2. The counting performed by the MMU will be inaccurate if interrupts occur during the fetch of that address.

TABLE 3-2. Instructions Causing Non-Sequential Fetches

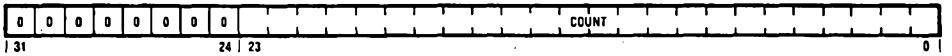
Branch	
ACBi	Add, Compare and Branch: unless result is zero
BR	Branch (Unconditional)
BSR	Branch to Subroutine
Bcond	Branch (Conditional): only if condition is met
CASEi	Case Branch
CXP	Call External Procedure
CXPD	Call External Procedure with Descriptor
DIA	Diagnose
JSR	Jump to Subroutine
JUMP	Jump
RET	Return from Subroutine
RXP	Return from External Procedure
BPT	Breakpoint Trap
FLAG	Trap on Flag
RETI	Return from Interrupt: if MSR loaded properly by supervisor
RETT	Return from Trap: if MSR loaded properly by supervisor
SVC	Supervisor Call

Also all traps or interrupts not generated by the MMU.

Branch to Following Instruction	
BICPSRi	Bit Clear in PSR
BISPSRi	Bit Set in PSR
LMR	Load Memory Management Register
LPRi	Load Processor Register: unless UPSR is the register specified
MOVSUi	Move Value from Supervisor to User Space
MOVUSi	Move Value from User to Supervisor Space
WAIT	Wait: fetches next instruction before waiting

3.6 BREAKPOINT COUNT REGISTER (BCNT)

The Breakpoint Count register (BCNT) permits the user to specify the number of breakpoint conditions given by register BPR0 that should be ignored before generating a Breakpoint trap. The BCNT register is 32 bits in length, containing a counter in its low-order 24 bits, as shown in *Figure 3-7.* The high-order eight bits are not used.



TL/EE/8692-24

FIGURE 3-7. Breakpoint Count Register (BCNT)

3.0 Architectural Description (Continued)

The BCNT register affects the generation of Breakpoint traps only when it is enabled by the CE bit in the BPR0 register. When the BPR0 breakpoint condition is encountered, and the BPR0 CE bit is 1, the contents of the BCNT register are checked against zero. If the BCNT contents are zero, a breakpoint trap is generated. If the contents are not equal to zero, no breakpoint trap is generated and the BCNT register is decremented by 1.

If the CE bit in the BPR0 register is 0, the BCNT register is ignored and the BPR0 condition breaks the program execution regardless of the BCNT register's contents. The BCNT register contents are unaffected.

3.7 MEMORY MANAGEMENT STATUS REGISTER (MSR)

The Memory Management Status Register (MSR) provides overall control and status fields for both address translation and debugging functions. The format of the MSR register is shown in *Figure 3-8*.

The MSR fields relevant to either of the above functions are described in the following sub-sections.

3.7.1 MSR Fields for Address Translation.

Control Functions

The address translation control bits in the MSR, ad exception of the R bit, are both readable (using the SMR instruction) and writable (using LMR).

R Reset. When read, this bit's contents are undefined. Whenever a "1" is written into it, MSR status fields TE, B, TET, ED, BD, EST and BST are cleared to all zeroes. (The BN bit is not affected.)

TU Translate User-Mode Addresses. While this bit is "1", the MMU translates all addresses presented while the CPU is in User Mode. While it is "0", the MMU echoes all User-Mode virtual addresses without performing translation or access level checking. This bit is cleared by a hardware Reset.

Note: Altering the TU bit has no effect on the contents of the TLB.

TS Translate Supervisor-Mode Addresses. While this bit is "1", the MMU translates all addresses presented while the CPU is in Supervisor Mode. While it is "0", the MMU echoes all Supervisor-Mode virtual addresses without translation or access level checking. This bit is cleared by a hardware Reset.

Note: Altering the TS bit has no effect on the contents of the TLB.

DS Dual-Space Translation. While this bit is "1", Supervisor Mode addresses and User Mode addresses are translated independently of each other, using separate mappings. While it is "0", both Supervisor Mode addresses and User Mode addresses are translated using the same mapping. See Section 3.2.2.

AO Access Level Override. This bit may be set to temporarily cause User Mode accesses to be given Supervisor Mode privilege. See Section 3.10.

Status Fields

The MSR status fields may be read using the MSR instruction, but are not writable. Instead, all status fields (except the BN bit) may be cleared by loading a "1" into the R bit using the LMR instruction.

TE Translation Error. This bit is set by the MMU to indicate that an address translation error has occurred. This bit is cleared by a hardware reset.

TET Translation Error Type. This three-bit field shows the reason(s) for the last address translation error reported by the MMU. The format of the TET field is shown below.

IL2	IL1	PL

PL Protection Level error. The access attempted by the CPU was not allowed by the protection level assigned to the page it attempted to access (forbidden by either of the Page Table Entry PL fields).

IL1 Invalid Level 1. The Level-1 Page Table Entry was invalid (V bit = 0).

IL2 Invalid Level 2. The Level-2 Page Table Entry was invalid (V bit = 0).

These error indications are not mutually exclusive. A protection level error and an invalid translation error can be reported simultaneously by the MMU.

ED Error Direction. This bit indicates the direction of the transfer that the CPU was attempting on the most recent address translation error.

ED=0=>Write cycle.

ED=1=>Read cycle.

EST Error Status. This 3-bit field is set on an address translation error to the low-order three bits of the CPU status bus. Combinations appearing in this field are summarized below.

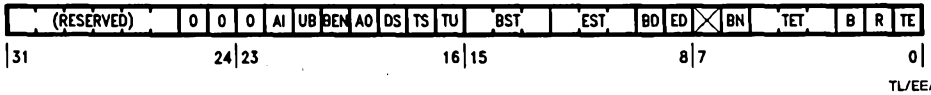
000 Sequential instruction fetch

001 Non-sequential instruction fetch

010 Operand transfer (read or write)

011 The Read action of a read-modify-write transfer (operands of access class "rmw" only: See the Series 32000 Instruction Set Reference Manual for further details).

100 A read transfer which is part of an effective address calculation (Memory Relative or External mode)



TL/EE/8692-25

Note: In some Series 32000 documentation, the bits TE, R and B are jointly referenced with the keyword "ERC".

FIGURE 3-8. Memory Management Status Register (MSR)

3.0 Architectural Description (Continued)

3.7.2 MSR Fields for Debugging

Control Functions

Breakpoint control bits in the MSR are both readable (using the SMR instruction) and writable (using LMR).

BEN Breakpoint Enable. Setting this bit enables both Breakpoint Registers (BPR0, BPR1) to monitor CPU activity. This bit is cleared by a hardware reset or whenever a Breakpoint trap or an address translation error occurs. If only one breakpoint register must be enabled, the other register should be disabled by clearing all of its control bits (bits 26–31) to zeroes.

Note: When the BEN bit is set (using the LMR instruction), the MMU enables breakpoints only after two non-sequential instruction fetch cycles have been completed by the CPU. See Section 3.9.

UB User-Only Breakpointing. When this bit is set in conjunction with the BEN bit, it limits the Breakpoint Registers to monitor addresses only while the CPU is in User Mode.

AI Abort/Interrupt. This bit selects the action taken by the MMU on a breakpoint. While AI is "0" the MMU generates a pulse on the $\overline{INT}$ pin (this can be used to generate a non-maskable interrupt). While AI is "1" the MMU generates an Abort pulse instead.

Status Fields

The MSR status fields may be read using the SMR instruction, but are not writable. Instead, all status fields (except the BN bit) may be cleared by loading a "1" into the R bit using the LMR instruction. See Section 3.7.1.

B Break. This bit is set to indicate that a breakpoint trap has been generated by the MMU.

BN Breakpoint Number. The BN bit contains the register number for the most recent breakpoint trap generated by the MMU. If BN is 1, the breakpoint was triggered by the BPR1 register. If BN is 0, the breakpoint was triggered by the BPR0 register. If both registers trigger a breakpoint simultaneously, the BN bit is set to 1.

BD Break Direction. This bit indicates the direction of the transfer that the CPU was attempting on the access that triggered the most recent breakpoint trap. It is loaded from the complement of the $\overline{DDIN}$ pin.

BD = 0 => Write cycle.

BD = 1 => Read cycle.

BST Breakpoint Status. This 3-bit field is loaded on a Breakpoint trap from the low-order three bits of the CPU status bus. Combinations appearing in this field are summarized below.

- 000 No break has occurred since the field was last reset.
- 001 Instruction fetch
- 010 Operand transfer (read or write)
- 011 The Read action of a read-modify-write transfer (operands of access class "rmw" only: See the Series 32000 Instruction Set Reference Manual for further details).
- 100 A read transfer which is part of an effective address calculation (Memory Relative or External mode)

Note: The BST field encodings 000 and 001 differ from those of the EST field (Section 3.7.1) because the MMU inserts a DIA instruction into the instruction stream in implementing Execution breakpoints (Section 2.7.1). One side effect of this is that a breakpoint trap is never triggered directly by a sequential instruction fetch cycle.

3.8 TRANSLATION LOOKASIDE BUFFER (TLB)

The Translation Lookaside Buffer is an on-chip fully associative memory. It provides direct virtual to physical mapping for the 32 most recently used pages, requiring only one clock period to perform the address translation.

The efficiency of the MMU is greatly increased by the TLB, which bypasses the much longer Page Table lookup in over 97% of the accesses made by the CPU.

Entries in the TLB are allocated and replaced by the MMU itself; the operating system is not involved. The TLB entries cannot be read or written by software; however, they can be purged from it under program control.

Figure 3-9 models the TLB. Information is placed into the TLB whenever the MMU performs a lookup from the Page Tables in memory. If the retrieved mapping is valid (V = 1 in both levels of the Page Tables), and the access attempted is permitted by the protection level, an entry of the TLB is loaded from the information retrieved from memory. The recipient entry is selected by an on-chip circuit that implements a Least-Recently-Used (LRU) algorithm. The MMU places the virtual page number (15 bits) and the Address Space qualifier bit into the Tag field of the TLB entry.

The Value portion of the entry is loaded from the Page Tables as follows:

The Translation field (16 bits) is loaded from the MS bit and PFN field of the Level-2 Page Table Entry.

The M bit is loaded from the Level-2 Page Table Entry.

The PL field (2 bits) is loaded to reflect the net protection level imposed by the PL fields of the Level-1 and Level-2 Page Table Entries.

(Not shown in the figure are additional bits associated with each TLB entry which flag it as full or empty, and which select it as the recipient when a Page Table lookup is performed.)

When a virtual address is presented to the MMU for translation, the high-order 15 bits (page number) and the Address Space qualifier are compared associatively to the corresponding fields in all entries of the TLB. When the Tag portion of a TLB entry completely matches the input values, the Value portion is produced as output. If the protection level is not violated, and the M bit does not need to be changed, then the physical address Page Frame number is output in the next clock cycle. If the protection level is violated, the MMU instead activates the Abort output. If no TLB entry matches, or if the matching entry's M bit needs to be changed, the MMU performs a page-table lookup from memory.

Note that for a translation to be loaded into the TLB it is necessary that the Level-1 and Level-2 Page Table Entries be valid (V bit = 1). Also, it is guaranteed that in

3.0 Architectural Description (Continued)

the process of loading a TLB entry (during a Page Table lookup) the Level-1 and Level-2 R bits will be set in memory if they were not already set. For these reasons, there is no need to replicate either the V bit or the R bit in the TLB entries.

Whenever a Page Table Entry in memory is altered by software, it is necessary to purge any matching entry from the TLB, otherwise the MMU would be translating the corresponding addresses according to obsolete information. TLB entries may be selectively purged by writing a virtual address to the EIA register using the LMR instruction. The TLB entry (if any) that matches that virtual address is then purged, and its space is made available for another translation. Purging is also performed by the MMU whenever an address space is remapped by altering the contents of the PTB0 or PTB1 register. When this is done, the MMU purges all the TLB entries corresponding to the address space mapped by that register. Turning translation on or off (via the MSR TU and TS bits) does not affect the contents of the TLB.

Note: If the value in the PTB0 register must be changed, it is strongly recommended that the translation be disabled before loading the new value, otherwise the purge performed may be incomplete. This is due to instruction prefetches and/or memory read cycles occurring during the LMR instruction which may restore TLB entries from the old map.

3.9 ENTRY/RE-ENTRY INTO PROGRAMS UNDER DEBUGGING

Whenever the MSR is written, breakpoints are disabled. After two non-sequential instruction fetch cycles have completed, they are again enabled if the new BEN bit value is '1'. The recommended sequence for entering a program under test is:

```
LMR    MSR, New_Value
RETT   n        ; or RETI
```

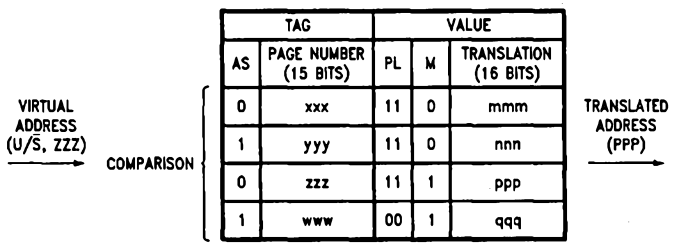
executed with interrupts disabled (CPU PSR I bit off).

This feature allows a debugger or monitor program to return control to a program being debugged without the risk of a false breakpoint trap being triggered during the return.

The LMR instruction performs the first non-sequential fetch cycle, in effect branching to the next sequential instruction. The RETT (or RETI) instruction performs the second non-sequential fetch as its last memory reference, branching to the first (next) instruction of the program under debug. The non-sequential fetch caused by the RETT instruction, which might not have occurred otherwise, is not monitored.

3.10 ADDRESS TRANSLATION ALGORITHM

The MMU either translates the 24-bit virtual address to a 25-bit physical address or reports a translation error. This process is described algorithmically in the following pages. See also *Figure 3-3*.



TL/EE/8692-26

FIGURE 3-9. TLB Model

MMU Page Table Lookup and Access Validation Algorithm

Legend:

x = y	x is assigned the value y
x == y	Comparison expression, true if x is equal to y
x AND y	Boolean AND expression, true only if assertions x and y are both true
x OR y	Boolean inclusive OR expression, true if either of assertions x and y is true
;	Delimiter marking end of statement
{ . . . }	Delimiters enclosing a statement block
item(i)	Bit number i of structure "item"
item(i:j)	The field from bit number i through bit number j of structure "item"
item.x	The bit or field named "x" in structure "item"
DONE	Successful end of translation; MMU provides translated address
ABORT	Unsuccessful end of translation; MMU aborts CPU access

This algorithm represents for all cases a valid definition of address translation.
Bus activity implied here occurs only if the TLB does not contain the mapping,
or if the reference requires that the MMU alter the M bit of the Page Table Entry.
Otherwise, the MMU provides the translated address in one clock period.

Input (from CPU):

- U (1 if U/$\overline{S}$ is high)
- W (1 if $\overline{DDIN}$ input is high)
- VA Virtual address consisting of:
  - INDEX_1 (from pins A23-A16)
  - INDEX_2 (from pins AD15-AD9)
  - OFFSET (from pins AD8-AD0)
- ACCESS_LEVEL The access level of a reference is a 2-bit value synthesized by the MMU from CPU status:
  - bit 1 = U AND NOT MSR.A0 (U from U/$\overline{S}$ input pin)
  - bit 0 = 1 for Write cycle, or Read cycle of an "rmw" class operand access
    0 otherwise.

Output:

- PA Physical Address on pins A24-A16, AD15-AD0;
  - or
- Abort pulse on $\overline{RST}$/$\overline{ABT}$ pin.

Uses:

- MSR Status Register:
  fields TU, TS and DS

MMU Page Table Lookup and Access Validation Algorithm (Continued)

```
     PTB0       Page Table Base Register 0
     PTB1       Page Table Base Register 1
     PTE_1      Level-1 Page Table Entry:
                        fields PFN, PL, V, R and MS
     PTEP_1     Pointer, holding address of PTE_1
     PTE_2      Level-2 Page Table Entry:
                        fields PFN, PL, V, M, R and MS
     PTEP_2     Pointer, holding address of PTE_2
IF ( (MSR.TU == 0) AND (U == 1) )  OR  ( (MSR.TS == 0) AND (U ==0) )  If translation not enabled then echo
     THEN  {  PA(0:23) = VA(0:23) ; PA(24) = 0 ; DONE  }  ;                  virtual address as physical address.

IF (MSR.DS == 1) AND (U == 1)                                          If Dual Space mode and CPU in User Mode
   THEN {  PTEP_1(24) = PTB1.MS ; PTEP_1(23:10) = PTB1(23:10) ;             then form Level-1 PTE address
           PTEP_1(9:2) = VA.INDEX_1 ; PTEP_1(1:0)=0  }                             from PTB1 register,
   ELSE {  PTEP_1(24) = PTB0.MS ; PTEP_1(23:10) = PTB0(23:10) ;             else form Level-1 PTE address
           PTEP_1(9:2) = VA.INDEX_1; PTEP_1(1:0) = 0                               from PTB0 register.
        } ;

                                                  - - - LEVEL 1 PAGE TABLE LOOKUP - - -

IF ( ACCESS_LEVEL > PTE_1.PL ) OR (PTE_1.V == 0)                       If protection violation or invalid Level-2 page
   THEN ABORT ;                                                            table then abort the access.

IF PTE_1.R == 0 THEN PTE_1.R = 1 ;                                     Otherwise, set Reference bit if not already set,
PTE_1(4) = (undefined value);                                                  (the M bit position may be garbaged)

PTEP_2(24) = PTE_1.MS ; PTEP_2(23:9) = PTE_1.PFN ;                             and form Level-2 PTE address.
PTEP_2(8:2) = VA.INDEX_2 ; PTEP_2(1:0) = 0 ;

                                                  - - - LEVEL 2 PAGE TABLE LOOKUP - - -

IF ( ACCESS_LEVEL > PTE_2.PL ) OR ( PTE_2.V ==0 )                            If protection violation or invalid page
   THEN ABORT ;                                                              then abort the access.

IF PTE_2.R == 0 THEN PTE_2.R == 1 ;                                    Otherwise, set Referenced bit if not already set,
IF ( W == 1) AND ( PTE_2.M == 0 ) THEN PTE_2.M = 1 ;                         if Write cycle set Modified bit if not
                                                                                already set,
PA(24) = PTE_2.MS ; PA(23:9) = PTE_2.PFN ; PA(8:0) = VA.OFFSET ;                  and generate physical address.
DONE ;
```

3.0 Architectural Description (Continued)

3.11 INSTRUCTION SET

Four instructions of the Series 32000 instruction set are executed cooperatively by the CPU and MMU. These are:

LMR	Load Memory Management Register
SMR	Store Memory Management Register
RDVAL	Validate Address for Reading
WRVAL	Validate Address for Writing

The format of the MMU slave instructions is shown in *Figure 3-10.* Table 3-3 shows the encodings of the "short" field for selecting the various MMU internal registers.

TABLE 3-3. "Short" Field Encodings

"Short" Field	Register
0000	BPR0
0001	BPR1
1010	MSR
1011	BCNT
1100	PTB0
1101	PTB1
1111	EIA

Note: All other codes are illegal. They will cause unpredictable registers to be selected if used in an Instruction.

For reasons of system security, all MMU instructions are privileged, and the CPU does not issue them to the MMU in User Mode. Any such attempt made by a User-Mode program generates the Illegal Operation trap, Trap (ILL). In addition, the CPU will not issue MMU instructions unless its CFG register's M bit has been set to validate the MMU instruction set. If this has not been done, MMU instructions are not recognized by the CPU, and an Undefined Instruction trap, Trap (UND), results.

The LMR and SMR instructions load and store MMU registers as 32-bit quantities to and from any general operand (including CPU General-Purpose Registers).

The RDVAL and WRVAL instructions probe a memory address and determine whether its current protection level would allow reading or writing, respectively, if the CPU were in User Mode. Instead of triggering an Abort trap, these instructions have the effect of setting the CPU PSR F bit if the type of access being tested for would be illegal. The PSR F bit can then be tested as a condition code.

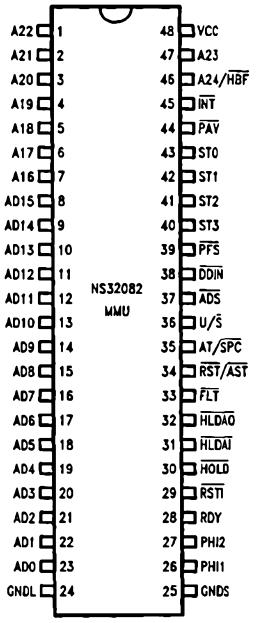
Note: The Series 32000 Dual-Space Move instructions (MOVSUi and MOVUSi), although they involve memory management action, are not Slave Processor instructions. The CPU implements them by switching the state of its U/$\overline{S}$ pin at appropriate times to select the desired mapping and protection from the MMU.

For full architectural details of these instructions, see the Series 32000 Instruction Set Reference Manual.

4.0 Device Specifications

4.1 NS32082 PIN DESCRIPTIONS

The following is a brief description of all NS32082 pins. The descriptions reference portions of the Functional Description, Section 2.0.



TL/EE/8692-28

Top View

Order Number NS16082D
See NS Package Number D48A

FIGURE 4-1. Dual-In-Line Package Connection Diagram

4.1.1 Supplies

Power (V_{CC}): +5V positive supply. Section 2.1.

Logic Ground (GNDL): Ground reference for on-chip logic. Section 2.1.

Buffer Ground (GNDB): Ground reference for on-chip drivers connected to output pins. Section 2.1.

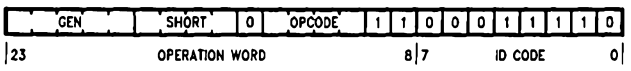
4.1.2 Input Signals

Clocks (PHI1, PHI2): Two-phase clocking signals. Section 2.2.

Ready (RDY): Active high. Used by slow memories to extend MMU originated memory cycles. Section 2.4.4.

Hold Request ($\overline{HOLD}$): Active low. Causes a release of the bus for DMA or multiprocessing purposes. Section 2.6.

Hold Acknowledge In ($\overline{HLDAI}$): Active low. Applied by the CPU in response to $\overline{HOLD}$ input, indicating that the CPU has released the bus for DMA or multiprocessing purposes. Section 2.6.



TL/EE/8692-27

FIGURE 3-10. MMU Slave Instruction Format

4.0 Device Specifications (Continued)

Reset Input ($\overline{RSTI}$): Active low. System reset. Section 2.3.

Status Lines (ST0–ST3): Status code input from the CPU. Active from T4 of previous bus cycle through T3 of current bus cycle. Section 2.4.

Program Flow Status ($\overline{PFS}$): Active low. Pulse issued by the CPU at the beginning of each instruction.

User/Supervisor Mode (U/$\overline{S}$): This signal is provided by the CPU. It is used by the MMU for protection and for selecting the address space (in dual address space mode only). Section 2.4.

Address Strobe Input ($\overline{ADS}$): Active low. Pulse indicating that a virtual address is present on the bus.

4.1.3 Output Signals

Reset Output/Abort ($\overline{RST}$/$\overline{ABT}$): Active Low. Held active longer than one clock cycle to reset the CPU. Pulsed low during T2 or TMMU to abort the current CPU Instruction.

Interrupt Output ($\overline{INT}$): Active low. Pulse used by the debug functions to inform the CPU that a break condition has occurred.

Float Output ($\overline{FLT}$): Active low. Floats the CPU from the bus when the MMU accesses page table entries or performs a physical breakpoint check. Section 2.4.3.

Physical Address Valid ($\overline{PAV}$): Active low. Pulse generated during TMMU indicating that a physical address is present on the bus.

Hold Acknowledge Output ($\overline{HLDAO}$): Active low. When active, indicates that the bus has been released.

4.1.4 Input-Output Signals

Data Direction In ($\overline{DDIN}$): Active low. Status signal indicating direction of data transfer during a bus cycle. Driven by the MMU during a page-table lookup.

Address Translation/Slave Processor Control ($\overline{AT}$/$\overline{SPC}$): Active low. Used by the CPU as the data strobe output for Slave Processor transfers; used by the MMU to acknowledge completion of an MMU instruction. Section 2.3 and 2.5. Held low during reset to select the address translation mode on the CPU.

M.S. Bit of Physical Address/High Byte Float (A24/$\overline{HBF}$): Most significant bit of physical address. Sampled on the rising edge of the reset input to select 16 or 32-bit bus mode. This pin outputs a low level if address translation is not enabled. It is floated during T2–T4 if 32-bit bus mode is selected.

Address Bits 16–23 (A16–A23): High order bits of the address bus. These signals are floated by the MMU during T2–T4 if 32-bit bus mode is selected.

Address/Data 0–15 (AD0–AD15): Multiplexed Address/Data Information. Bit 0 is the least significant bit.

4.2 ABSOLUTE MAXIMUM RATINGS

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

Temperature Under Bias	0°C to +70°C
Storage Temperature	−65°C to +150°C
All Input or Output Voltages with Respect to GND	−0.5V to +7V
Power Dissipation	1.5W

Note: *Absolute maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended; operation should be limited to those conditions specified under Electrical Characteristics.*

4.3 ELECTRICAL CHARACTERISTICS T_A = 0 to +70°C, V_{CC} = 5V ±5%, GND = 0V

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IH}	High Level Input Voltage		2.0		V_{CC} + 0.5	V
V_{IL}	Low Level Input Voltage		−0.5		0.8	V
V_{CH}	High Level Clock Voltage	PHI1, PHI2 pins only	V_{CC} − 0.35		V_{CC} + 0.5	V
V_{CL}	Low Level Clock Voltage	PHI1, PHI2 pins only	−0.5		0.3	V
V_{CLT}	Low Level Clock Voltage, Transient (ringing tolerance)	PHI1, PHI2 pins only	−0.5		0.6	V
V_{OH}	High Level Output Voltage	I_{OH} = −400 μA	2.4			V
V_{OL}	Low Level Output Voltage	I_{OL} = 2 mA			0.45	V
I_{ILS}	$\overline{AT}$/$\overline{SPC}$ Input Current (low)	V_{IN} = 0.4V, $\overline{AT}$/$\overline{SPC}$ in input mode	0.05		1.0	mA
I_I	Input Load Current	0 ≤ V_{IN} ≤ V_{CC}, All inputs except PHI1, PHI2, $\overline{AT}$/$\overline{SPC}$	−20		20	μA
I_L	Leakage Current (Output and I/O Pins in TRI-STATE/Input Mode)	0.4 ≤ V_{IN} ≤ V_C	−20		30	μA
I_{CC}	Active Supply Current	I_{OUT} = 0, T_A = 25°C		200	300	mA

4.0 Device Specifications (Continued)

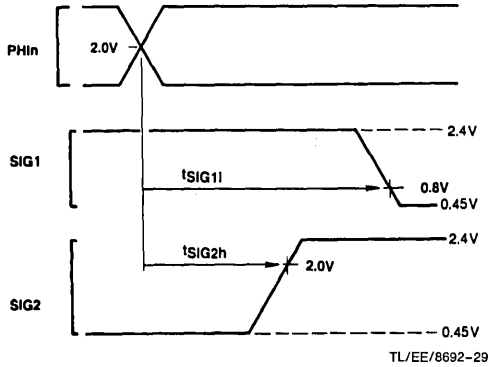
4.4 SWITCHING CHARACTERISTICS

4.4.1 Definitions

All the timing specifications given in this section refer to 2.0V on the rising or falling edges of the clock phases PHI1 and PHI2, and 0.8V or 2.0V on all other signals as illustrated in *Figures 4-2* and *4-3*, unless specifically stated otherwise.

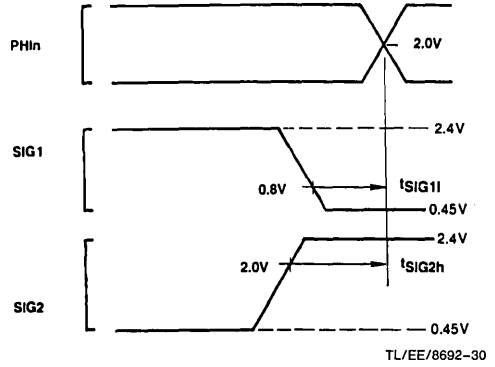
ABBREVIATIONS:

L.E. — leading edge R.E. — rising edge
T.E. — trailing edge F.E. — falling edge



TL/EE/8692-29

FIGURE 4-2. Timing Specification Standard (Signal Valid after Clock Edge)



TL/EE/8692-30

FIGURE 4-3. Timing Specification Standard (Signal Valid before Clock Edge)

4.4.2 Timing Tables

4.4.2.1 Output Signals: Internal Propagation Delays, NS32082-10.
Maximum times assume capacitive loading of 100 pF.

Name	Figure	Description	Reference/Conditions	NS32082-10		Units
				Min	Max	
t_{ALv}	4-4	Address Bits 0-15 Valid	After R.E., PHI1 TMMU or T1		40	ns
t_{ALh}	4-4	Address Bits 0-15 Hold	After R.E., PHI1 T2	5		ns
t_{AHv}	4-4, 4-6	Address Bits 16-24 Valid	After R.E., PHI1 TMMU or T1		40	ns
t_{AHh}	4-4	Address Bits 16-24 Hold	After R.E., PHI1 T2	5		ns
t_{ALPAVs}	4-5	Address Bits 0-15 Set Up	Before $\overline{PAV}$ T.E.	25		ns
t_{AHPAVs}	4-5	Address Bits 16-24 Set Up	Before $\overline{PAV}$ T.E.	25		ns
t_{ALPAVh}	4-5	Address Bits 0-15 Hold	After $\overline{PAV}$ T.E.	15		ns
t_{AHPAVh}	4-5	Address Bits 16-24 Hold	After $\overline{PAV}$ T.E.	15		ns
t_{ALf}	4-10	AD0-AD15 Floating	After R.E., PHI1 T2		25	ns
t_{AHf}	4-7, 4-10	A16-A24 Floating	After R.E., PHI1 T2 or T1		25	ns
t_{ALz}	4-15, 4-16	AD0-AD15 Floating (Caused by $\overline{HOLD}$)	After R.E., PHI1 Ti		25	ns
t_{AHZ}	4-15, 4-16	A16-A24 Floating (Caused by $\overline{HOLD}$)	After R.E., PHI1 Ti		25	ns
t_{ALr}	4-15, 4-16	AD0-AD15 Return from Floating (Caused by $\overline{HOLD}$)	After R.E., PHI1 T1		50	ns

4.0 Device Specifications (Continued)

4.4.2.1 Output Signals: Internal Propagation Delays, NS32082-10. (Continued)

Name	Figure	Description	Reference/Conditions	NS32082-10		Units
				Min	Max	
t_{AHr}	4–15, 4–16	A16–A24 Return from Floating (Caused by $\overline{HOLD}$)	After R.E., PHI1 T1		50	ns
t_{Dv}	4–6	Data Valid (Memory Write)	After R.E., PHI1 T2		50	ns
t_{Dh}	4–6	Data Hold (Memory Write)	After R.E., PHI1 next T1 or Ti	0		ns
t_{Df}	4–11	Data Bits Floating (Slave Processor Read)	After R.E., PHI1 T1 or Ti		10	ns
t_{Dv}	4–11	Data Valid (Slave Processor Read)	After R.E., PHI1 T1		50	ns
t_{Dh}	4–11	Data Hold (Slave Processor Read)	After R.E., PHI1 next T1 or Ti	0		ns
t_{DDINv}	4–5, 4–7	$\overline{DDIN}$ Signal Valid	After R.E., PHI1 T1 or T_{MMU}		50	ns
t_{DDINh}	4–5	$\overline{DDIN}$ Signal Hold	After R.E., PHI1 T1 or Ti	0		ns
t_{DDINf}	4–7	$\overline{DDIN}$ Signal Floating	After R.E., PHI1 T2		25	ns
t_{DDINz}	4–16	$\overline{DDIN}$ Signal Floating (Caused by $\overline{HOLD}$)	After R.E., PHI1 Ti		50	ns
t_{DDINr}	4–16	$\overline{DDIN}$ Return from Floating (Caused by $\overline{HOLD}$)	After R.E., PHI1 T1 or Ti		50	ns
t_{DDINAf}	4–9	$\overline{DDIN}$ Floating after Abort ($\overline{FLT}$ = 0)	After R.E., PHI2 T2		25	ns
t_{PAVa}	4–4	$\overline{PAV}$ Signal Active	After R.E., PHI1 T_{MMU} or T1		35	ns
t_{PAVia}	4–4	$\overline{PAV}$ Signal Inactive	After R.E., PHI2 T_{MMU} or T1		40	ns
t_{PAVw}	4–4	$\overline{PAV}$ Pulse Width	At 0.8V (Both Edges)	30		ns
t_{PAVdz}	4–14, 4–15	$\overline{PAV}$ Floating Delay	After $\overline{HLDAI}$ F.E.		25	ns
t_{PAVdr}	4–14, 4–15	$\overline{PAV}$ Return from Floating	After $\overline{HLDAI}$ R.E.		25	ns
t_{PAVz}	4–16	$\overline{PAV}$ Floating (Caused by $\overline{HOLD}$)	After R.E., PHI2 T4		30	ns
t_{PAVr}	4–16	$\overline{PAV}$ Return from Floating (Caused by $\overline{HOLD}$)	After R.E., PHI2 Ti		30	ns
t_{FLTa}	4–5, 4–10	$\overline{FLT}$ Signal Active	After R.E., PHI1 T_{MMU}		55	ns
t_{FLTia}	4–7, 4–10	$\overline{FLT}$ Signal Inactive	After R.E., PHI1 T_{MMU}, T_f or T2		35	ns
t_{ABTa}	4–8, 4–10	Abort Signal Active	After R.E., PHI1 T_{MMU} or T1		55	ns
t_{ABTia}	4–8, 4–10	Abort Signal Inactive	After R.E., PHI1 T2		55	ns
t_{ABTw}	4–8, 4–10	Abort Pulse Width	At 0.8V (Both Edges)	70		ns
t_{INTa}	4–4, 4–10	$\overline{INT}$ Signal Active	After R.E., PHI1 T_{MMU} or T_f		55	ns
t_{INTia}	4–4, 4–10	$\overline{INT}$ Signal Inactive	After R.E., PHI1 T2		55	ns
t_{INTw}	4–10	$\overline{INT}$ Pulse Width	At 0.8V (Both Edges)	70		ns
t_{SPCa}	4–13	$\overline{SPC}$ Signal Active	After R.E., PHI1 T1		40	ns
t_{SPCia}	4–13	$\overline{SPC}$ Signal Inactive	After R.E., PHI1 T4		40	ns

4.0 Device Specifications (Continued)

4.4.2.1 Output Signals: Internal Propagation Delays, NS32082-10. (Continued)

Name	Figure	Description	Reference/Conditions	NS32082-10		Units
				Min	Max	
t_{SPCf}	4–13	$\overline{SPC}$ Signal Floating	After F.E., PHI1 T4		25	ns
t_{SPCw}	4–13	$\overline{SPC}$ Pulse Width	At 0.8V (Both Edges)	70		ns
t_{HLD0da}	4–14	$\overline{HLDA0}$ Assertion Delay	After $\overline{HLDAI}$ F.E.		50	ns
$t_{HLD0dia}$	4–14, 4–15	$\overline{HLDA0}$ Deassertion Delay	After $\overline{HLDAI}$ R.E.		50	ns
t_{HLD0a}	4–15, 4–16	$\overline{HLDA0}$ Signal Active	After R.E., PHI1 Ti		30	ns
t_{HLD0ia}	4–16	$\overline{HLDA0}$ Signal Inactive	After R.E, PHI1 Ti		30	ns
t_{ATa}	4–18	$\overline{AT}/\overline{SPC}$ Signal Active	After R.E., PHI1		35	ns
t_{ATia}	4–18	$\overline{AT}/\overline{SPC}$ Signal Inactive	After R.E., PHI1		35	ns
t_{ATf}	4–18	$\overline{AT}/\overline{SPC}$ Signal Floating	After F.E., PHI1		25	ns
t_{RST0a}	4–18	$\overline{RST}/\overline{ABT}$ Asserted (Low)	After R.E. PHI1		30	ns
t_{RST0ia}	4–18	$\overline{RST}/\overline{ABT}$ Deasserted (High)	After R.E. PHI1 Ti		30	ns

4.4.2.2 Input Signal Requirements: NS32082-10

Name	Figure	Description	Reference/Conditions	NS32082-10		Units
				Min	Max	
t_{DIs}	4–5	Data In Set Up (Memory Read)	Before F.E., PHI2 T3	15		ns
t_{DIh}	4–5	Data In Hold (Memory Read)	After R.E., PHI1 T4	3		ns
t_{DIs}	4–12	Data In Set Up (Slave Processor Write)	Before F.E., PHI2 T1	20		ns
t_{DIh}	4–12	Data In Hold (Slave Processor Write)	After R.E., PHI1 T4	3		ns
t_{RDYs}	4–5	RDY Signal Set Up	Before F.E., PHI2 T2 or T3	15		ns
t_{RDYh}	4–5	RDY Signal Hold	After F.E., PHI1 T3	5		ns
t_{USs}	4–4, 4-11	U/$\overline{S}$ Signal Set Up	Before F.E., PHI2 T4 or Ti	35		ns
t_{USh}	4–4, 4–11	U/$\overline{S}$ Signal Hold	After R.E., PHI1 Next T4	0		ns

4.0 Device Specifications (Continued)

4.4.2.2 Input Signal Requirements: NS32082-10 (Continued)

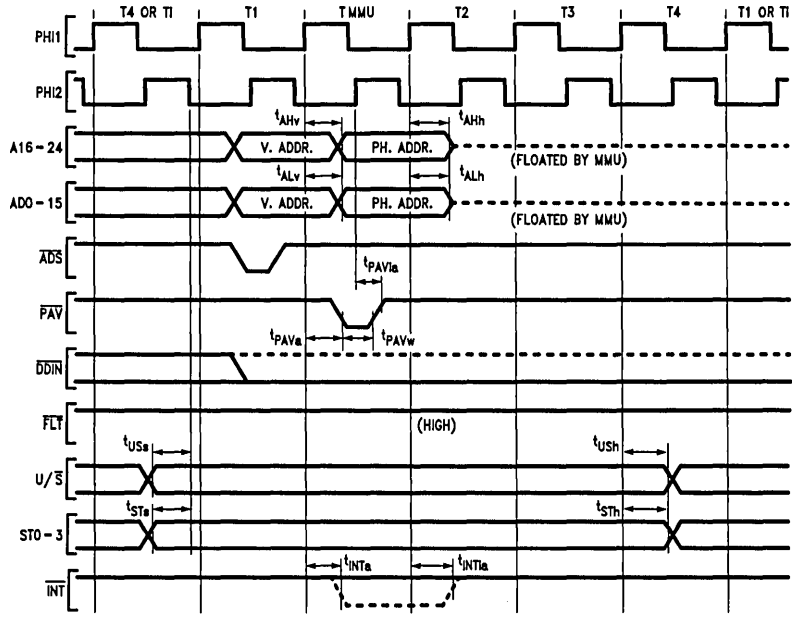
Name	Figure	Description	Reference/Conditions	NS32082-10		Units
				Min	Max	
t_{STs}	4–4, 4–11	Status Signals Set Up	Before F.E., PHI2 T4 or Ti	35		ns
t_{STh}	4–4, 4–11	Status Signals Hold	After R.E., PHI1 Next T4	0		ns
t_{SPCs}	4-11	$\overline{SPC}$ Input Set Up	Before F.E., PHI2 T1	45		ns
t_{SPCh}	4-11	$\overline{SPC}$ Input Hold	After R.E., PHI1 T4	0		ns
t_{HLDs}	4-16	$\overline{HOLD}$ Signal Set Up	Before F.E., PHI2 T4 or Ti	25		ns
t_{HLDh}	4-16	$\overline{HOLD}$ Signal Hold	After F.E., PHI2 T4 or Ti	0		ns
t_{HLDIs}	4-15	$\overline{HLDAI}$ Signal Set Up	Before F.E., PHI2 Ti	20		ns
t_{HLDIh}	4-15	$\overline{HLDAI}$ Signal Hold	After F.E., PHI2 Ti	0		ns
t_{HBFs}	4-18	A24/$\overline{HBF}$ Signal Set Up	Before F.E., PHI2	10		ns
t_{HBFh}	4-18	A24/$\overline{HBF}$ Signal Hold	After F.E., PHI2	0		ns
t_{RSTIs}	4-18	Reset Input Set Up	Before F.E., PHI1	20		ns
t_{PWR}	4-19	Power Stable to $\overline{RSTI}$ R.E.	After V_{CC} Reaches 4.5V	50		μs
t_{RSTIw}		$\overline{RSTI}$ Pulse Width	At 0.8V (Both Edges)	64		t_{cp}

4.4.2.3 Clocking Requirements: NS32082-10

Name	Figure	Description	Reference/ Conditions	NS32082-10		Units
				Min	Max	
t_{Cp}	4–17	Clock Period	R.E., PHI1, PHI2 to Next R.E., PHI1, PHI2	100	250	ns
t_{CLw}	4–17	PHI1, PHI2 Pulse Width	At 2.0V on PHI1, PHI2 (Both Edges)	$0.5t_{Cp}$ − 10 ns		
t_{CLh}	4–17	PHI1, PHI2 High Time	At V_{CC} − 0.9V on PHI1, PHI2 (Both Edges)	$0.5t_{Cp}$ − 15 ns		
t_{CLl}	4–17	PHI1, PHI2 Low Time	At 0.8V on PHI1, PHI2	$0.5t_{Cp}$ − 5 ns		
$t_{nOVL\,(1,2)}$	4–17	Non-overlap Time	0.8V on F.E. PHI1, PHI2 to 0.8V on R.E., PHI2, PHI1	−2	5	ns
t_{nOVLas}		Non-overlap Asymmetry ($t_{nOVL(1)} - t_{nOVL(2)}$)	At 0.8V on PHI1, PHI2	−4	4	ns
t_{CLwas}		PHI1, PHI2 Asymmetry $t_{CLw(1)} - t_{CLw(2)}$	At 2.0V on PHI1, PHI2	−5	5	ns

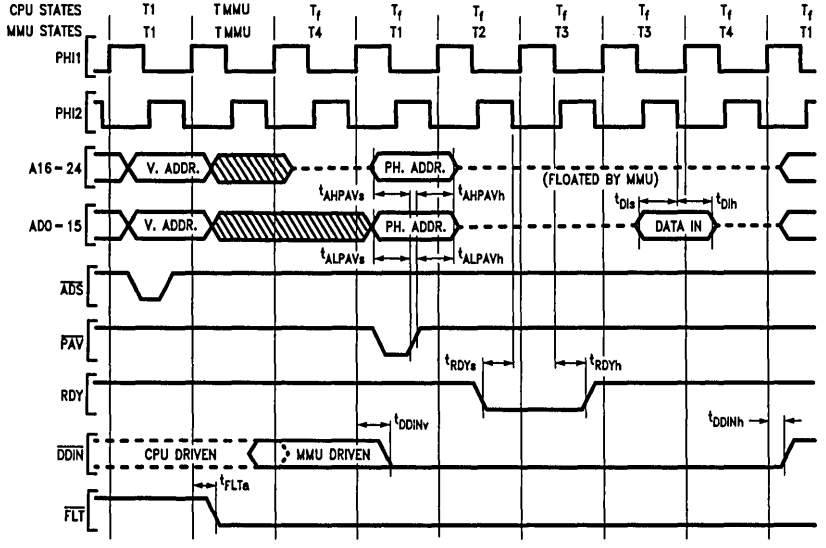
4.0 Device Specifications (Continued)

4.4.3 Timing Diagrams



TL/EE/8692-31

FIGURE 4-4. CPU Read (Write) Cycle Timing (32-Bit Mode); Translation in TLB

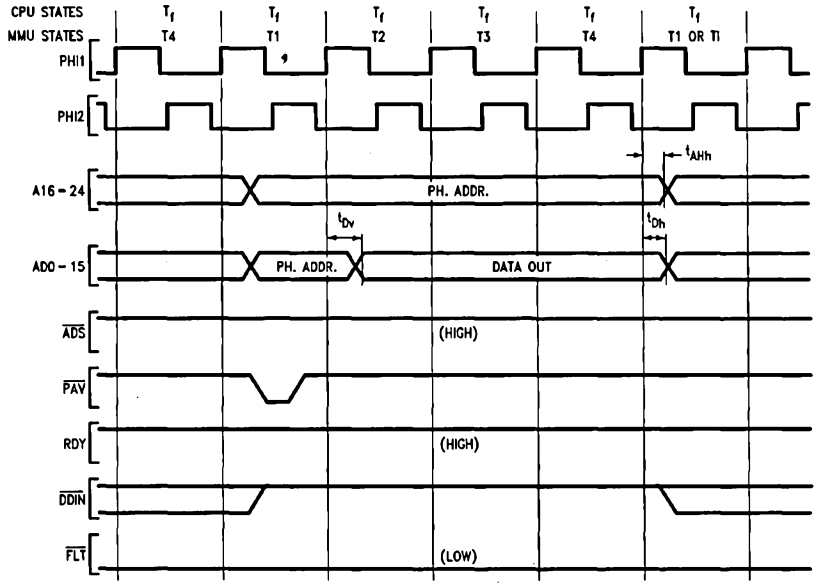


TL/EE/8692-32

FIGURE 4-5. MMU Read Cycle Timing (32-Bit Mode); After a TLB Miss

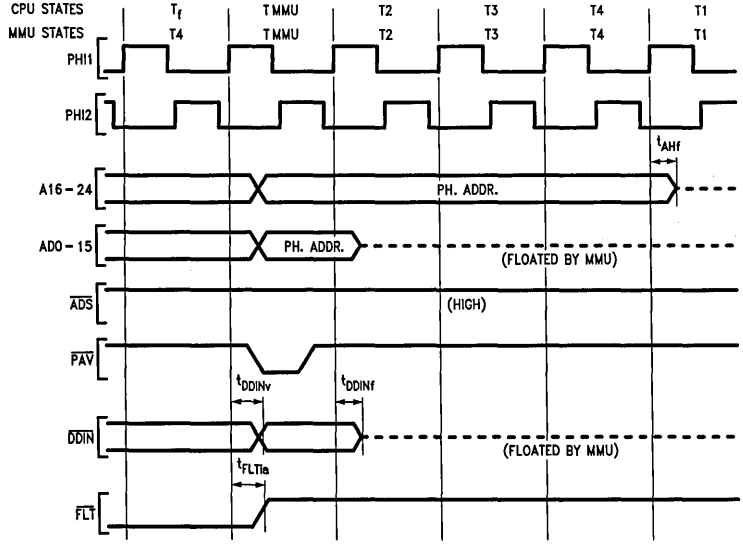
Note: After $\overline{FLT}$ is asserted, $\overline{DDIN}$ may be driven temporarily by both CPU and MMU. This, however, does not cause any conflict, since both CPU and MMU force $\overline{DDIN}$ to the same logic level.

4.0 Device Specifications (Continued)



TL/EE/8692-33

FIGURE 4-6. MMU Write Cycle Timing; after a TLB Miss

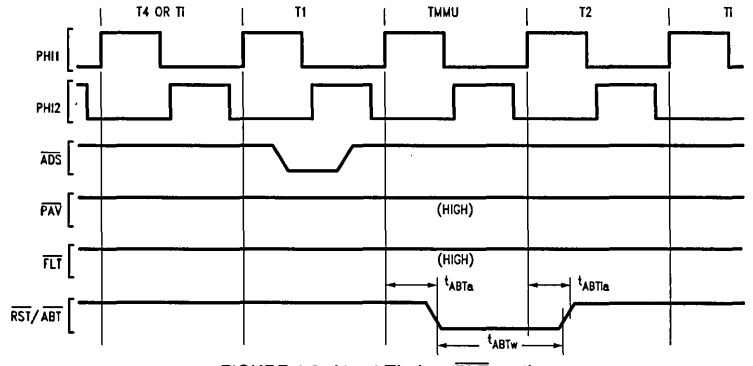


TL/EE/8692-34

FIGURE 4-7. $\overline{FLT}$ Deassertion Timing

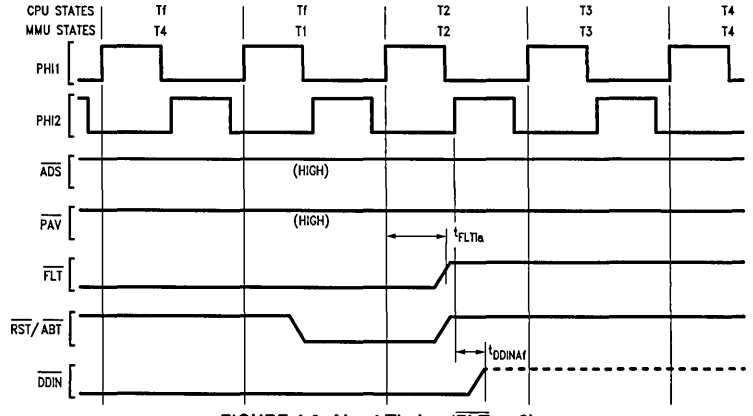
Note: After $\overline{FLT}$ is deasserted, $\overline{DDIN}$ may be driven temporarily by both CPU and MMU. This, however, does not cause any conflict. Since CPU and MMU force $\overline{DDIN}$ to the same logic level.

4.0 Device Specifications (Continued)



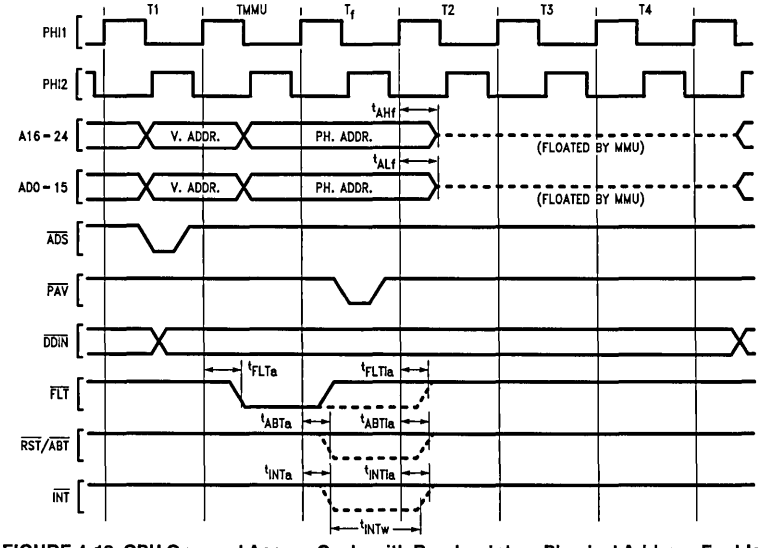
TL/EE/8692-35

FIGURE 4-8. Abort Timing ($\overline{FLT}$ = 1)



TL/EE/8692-36

FIGURE 4-9. Abort Timing ($\overline{FLT}$ = 0)

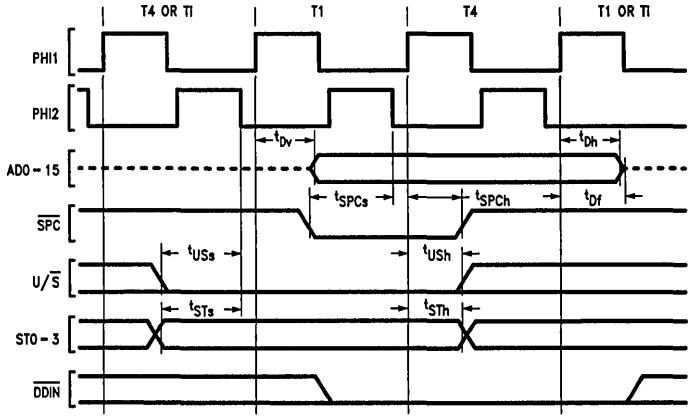


TL/EE/8692-37

FIGURE 4-10. CPU Operand Access Cycle with Breakpoint on Physical Address Enabled

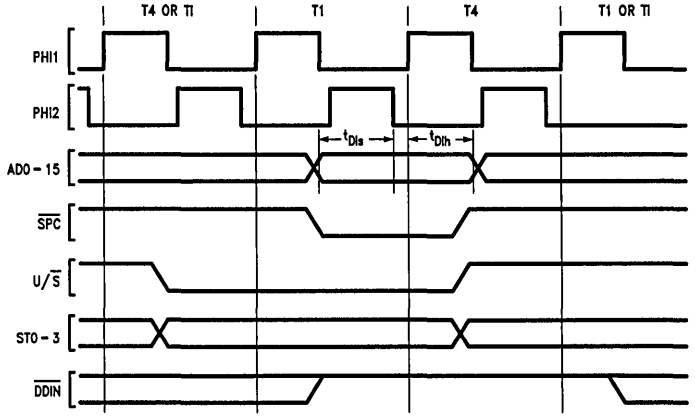
Note: If a breakpoint condition is met and abort on breakpoint is enabled, the bus cycle is aborted. In this case $\overline{FLT}$ is stretched by one clock cycle.

4.0 Device Specifications (Continued)



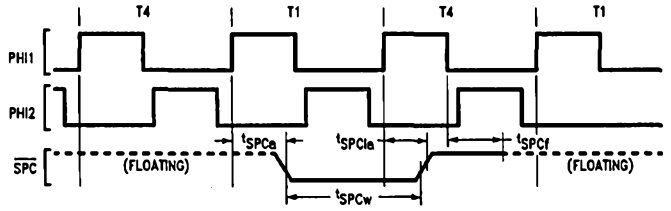
TL/EE/8692-38

FIGURE 4-11. Slave Access Timing; CPU Reading from MMU



TL/EE/8692-39

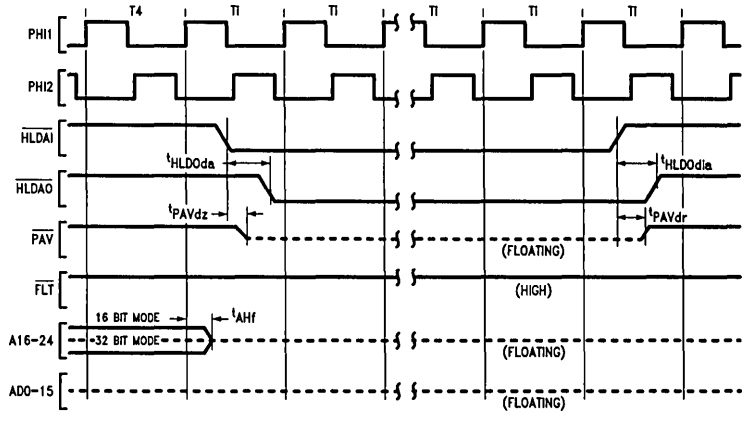
FIGURE 4-12. Slave Access Timing; CPU Writing to MMU



TL/EE/8692-40

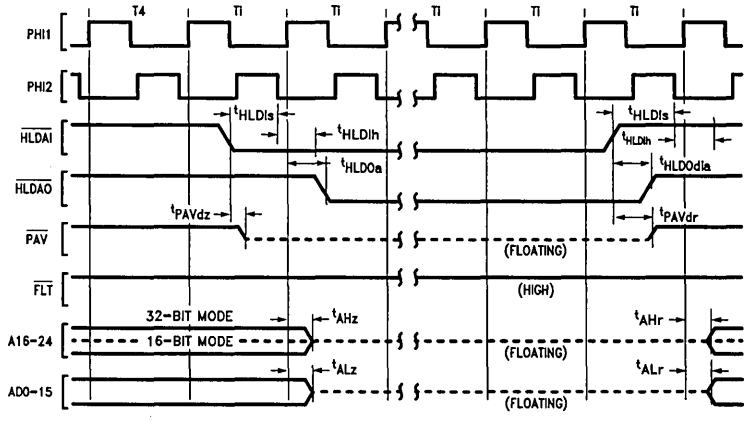
FIGURE 4-13. $\overline{SPC}$ Pulse from the MMU

4.0 Device Specifications (Continued)



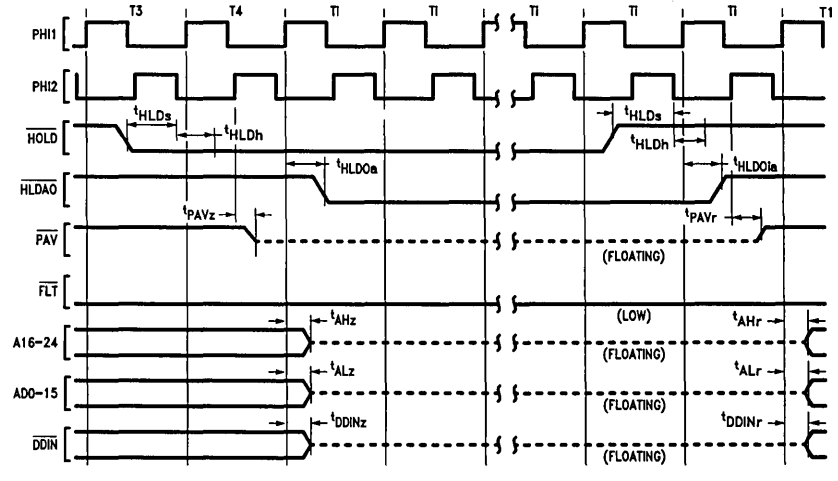
TL/EE/8692-41

FIGURE 4-14. Hold Timing ($\overline{FLT}$ = 1); SMR Instruction Not Being Executed



TL/EE/8692-42

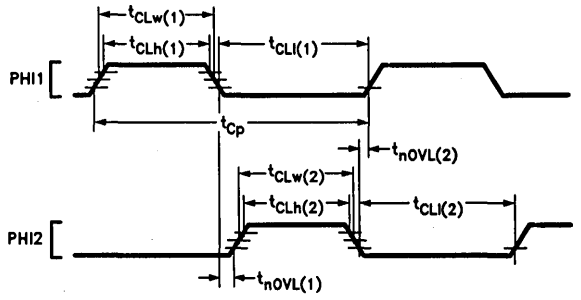
FIGURE 4-15. Hold Timing ($\overline{FLT}$ = 1); SMR Instruction Being Executed



TL/EE/8692-43

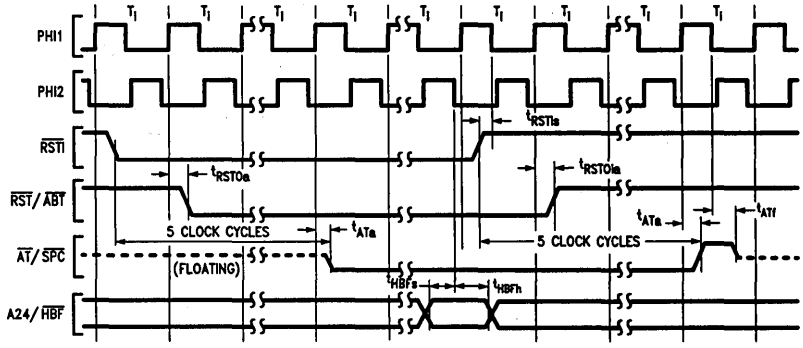
FIGURE 4-16. Hold Timing ($\overline{FLT}$ = 0)

4.0 Device Specifications (Continued)



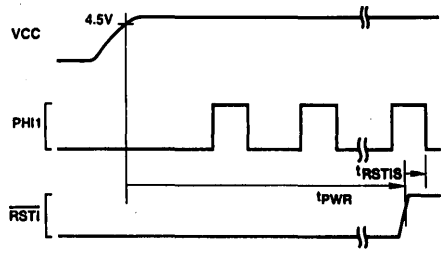
TL/EE/8692-49

FIGURE 4-17. Clock Waveforms



TL/EE/8692-45

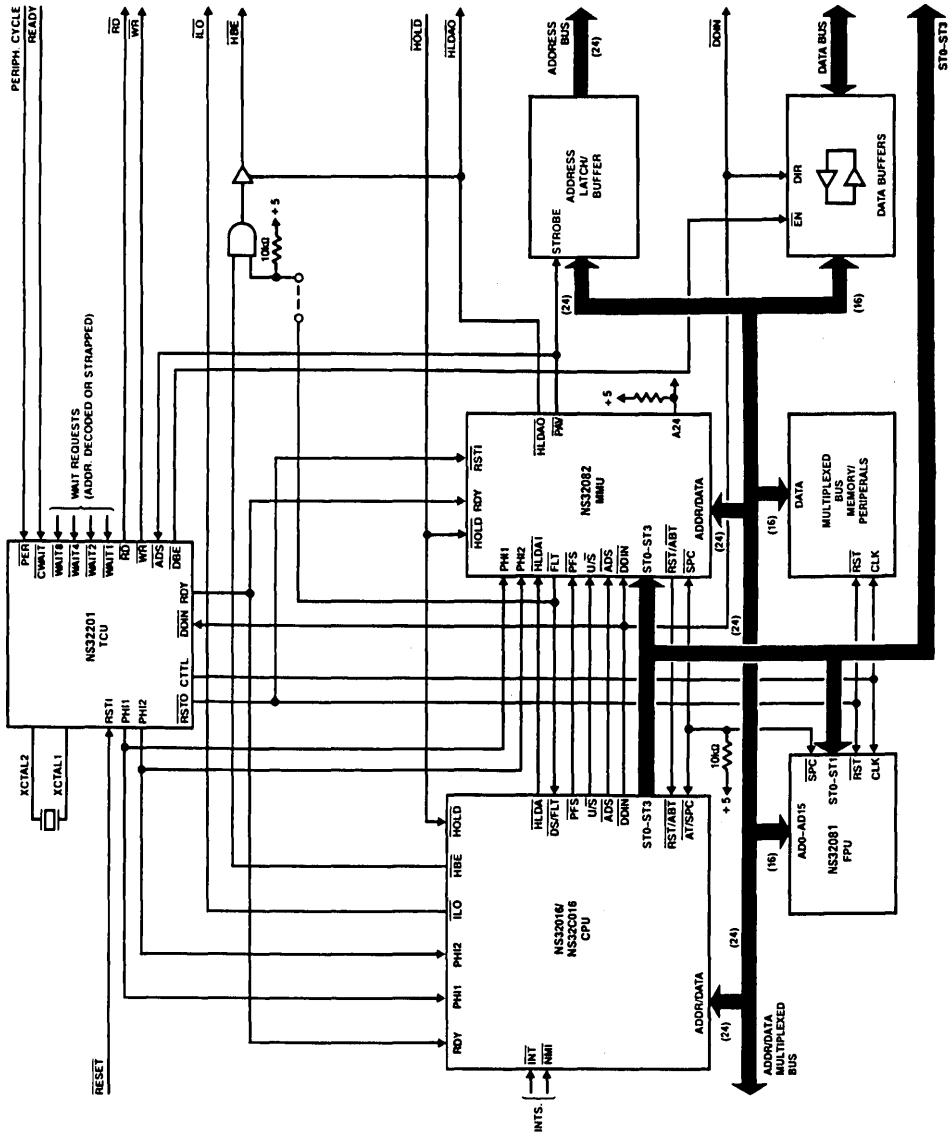
FIGURE 4-18. Reset Timing



TL/EE/8692-46

FIGURE 4-19. Power-On Reset

Appendix A: Interfacing Suggestions

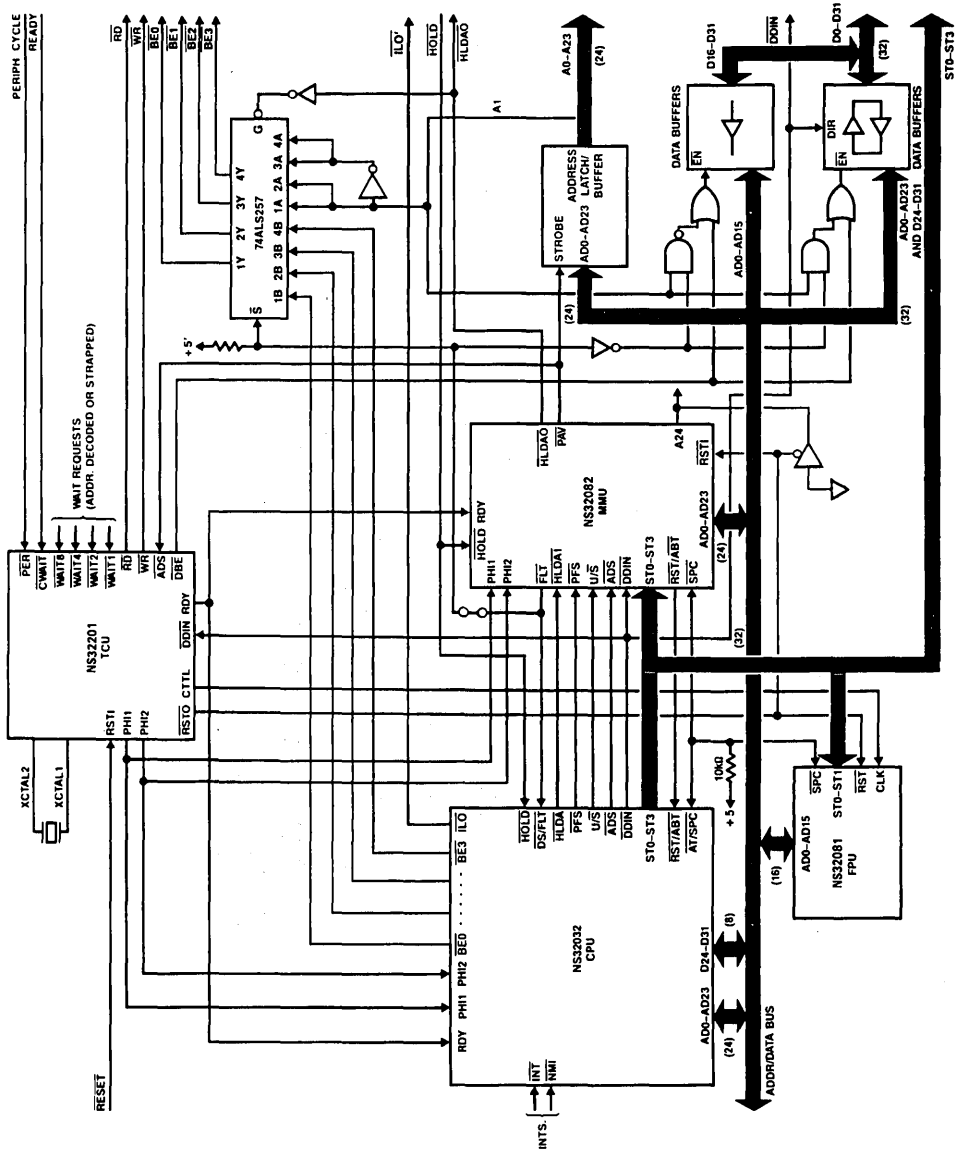


TL/EE/8692-47

FIGURE A-1. System Connection Diagram

Note: The "AND" gate on the $\overline{HBE}$ line is not needed when an NS32016 is used.

Appendix A: Interfacing Suggestions (Continued)



TL/EE/8692-48

FIGURE A-2. System Connection Diagram