National Semiconductor

PRELIMINARY

NS32382-10/NS32382-15 Memory Management Units

General Description

The NS32382 Memory Management Unit (MMU) provides hardware support for demand-paged virtual memory implementations. The NS32382 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 4 Gbytes of physical memory. Support for multiple address spaces, virtual machines, and program debugging is provided.

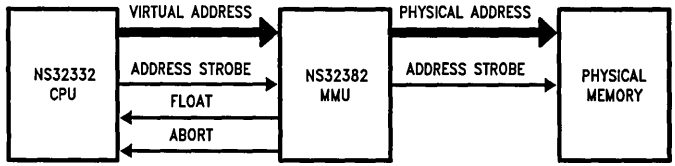
High-speed address translation is performed on-chip through a 32-entry fully associative translation look-aside buffer (TLB), which maintains itself from tables in memory with no software intervention. Protection violations and page faults (references to non-resident pages) are automatically detected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

Additional features for program debugging include three breakpoint registers which provide the programmer with powerful stand-alone debugging capability.

Features

- Compatible with the NS32332 CPU
- Totally automatic mapping of 4 Gbyte 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
- Dedicated 32-bit physical address bus
- Non-cacheable page support
- 125-pin PGA (Pin grid array) package

Conceptual Address Translation Model



TL/EE/9142-1

Table Of Contents

List of Illustrations

List of Illustrations (Continued)

Tables

1.0 Product Introduction

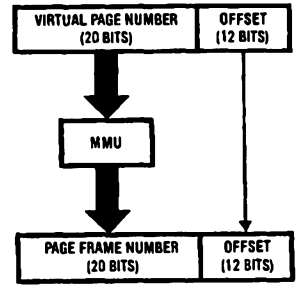
The NS32382 MMU provides hardware support for three basic features of the Series 32000; dynamic address translation, access level checking and software debugging. Dynamic Address Translation is required to implement demand-paged virtual memory. Access level checking is performed during address translation, 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 4 gigabytes (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 4 Kbyte blocks called "pages". It interprets the 32-bit address from the CPU as a 20-bit "page number" followed by a 12-bit offset, which indicates the position of a byte within the selected page. Similarly, the MMU divides the physical memory into 4 Kbyte frames, each of which can hold a virtual page.

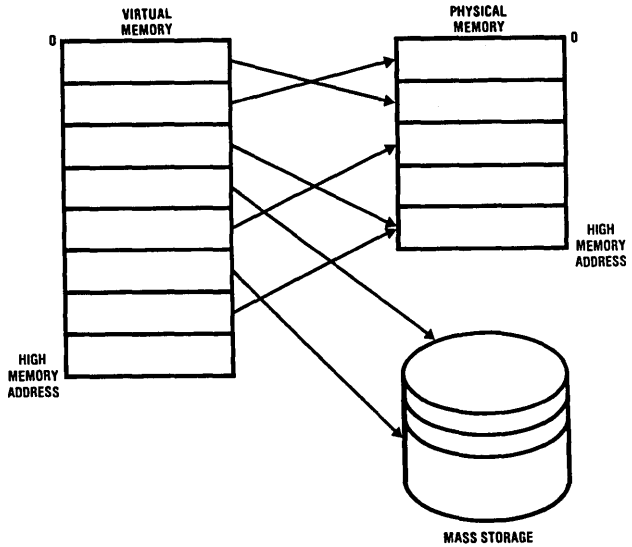
The translation process is therefore modeled as accepting a virtual page number from the CPU and substituting the corresponding physical page frame number for it, as shown in *Figure 1-2.* The offset is not changed. The translated page frame number is 20 bits long. Physical addresses issued by the MMU are 32 bits wide.



TL/EE/9142-3

FIGURE 1-2. NS32382 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, restores its internal state to the point prior to the instruction being executed, and enters the operating system through the abort trap vector.



TL/EE/9142-2

FIGURE 1-1. The Virtual Memory Model

1.0 Product Introduction (Continued)

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 CPU bus for a single or a range of virtual addresses in real time.

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.

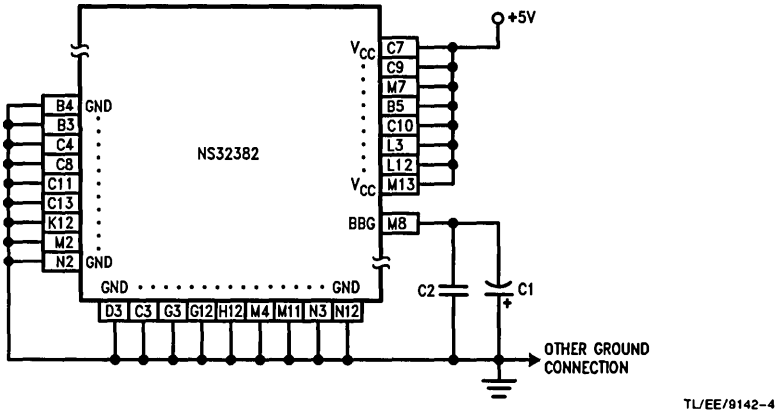
c) When performing the MOVM instruction, the entire source and destination blocks must be considered "operands" as above, and they must not overlap.

2.0 Functional Description

2.1 POWER AND GROUNDING

The NS32382 requires a single 5V power supply, applied on eight (V_{CC}) pins. These pins should be connected together by a power (V_{CC}) plane on the printed circuit board. See *Figure 2-1*.

The grounding connections are made on eighteen (GND) pins.



C1 = 1 μF, Tantalum.

C2 = 1000 pF, low inductance. This should be either a disc or monolithic ceramic capacitor.

FIGURE 2-1. Recommended Supply Connections

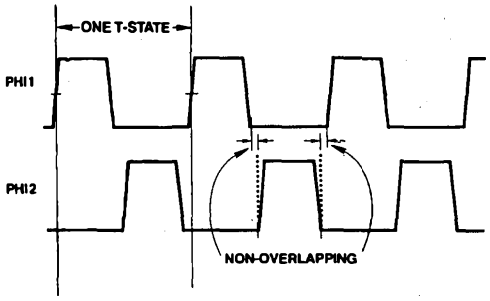
2.0 Functional Description (Continued)

These pins should be connected together by a ground (GND) plane on the printed circuit board.

In addition to V_{CC} and Ground, the NS32382 MMU uses an internally-generated negative voltage (BBG), output of the on-chip substrate voltage generator. It is necessary to filter this voltage externally by attaching a pair of capacitors *(Figure 2-1)* from the BBG pin to ground.

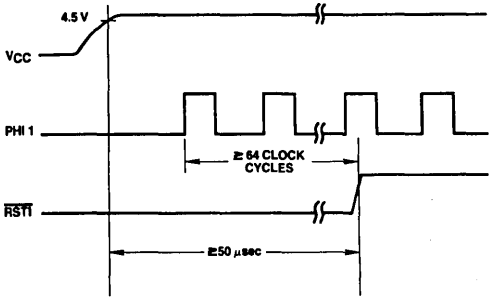
2.2 CLOCKING

The NS32382 inputs clocking signals from the NS32301 Timing Control Unit (TCU), which presents two non-overlapping phases of a single clock frequency. These phases are called PHI1 (pin B8) and PHI2 (pin B9). Their relationship to each other is shown in *Figure 2-2*.



TL/EE/9142-5

FIGURE 2-2. Clock Timing Relationships



TL/EE/9142-6

FIGURE 2-3. Power-On Reset Requirements

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.

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.

2.3 RESETTING

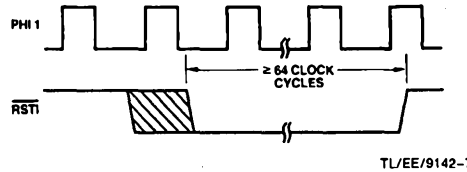
The $\overline{RSTI}$ input pin is used to reset the NS32382. The MMU responds to $\overline{RSTI}$ by terminating processing, resetting its internal logic and clearing the MCR and MSR registers.

Only the MCR and MSR registers are changed on reset. No other program accessible registers are affected.

The $\overline{RST}/\overline{ABT}$ signal is activated by the MMU on reset. This signal should be used to reset the CPU.

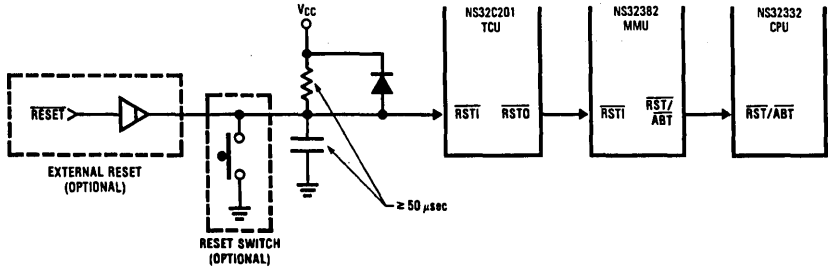
On application of power, $\overline{RSTI}$ must be held low for at least 50 μs after V_{CC} is stable. This is to ensure that all on-chip voltages are completely stable before operation. Whenever a Reset is applied, it must also remain active for not less than 64 clock cycles. See *Figures 2-3* and *2-4*.

The NS32C201 Timing Control Unit (TCU) provides circuitry to meet the Reset requirements of the NS32382 MMU. *Figure 2-5* shows the recommended connections.



TL/EE/9142-7

FIGURE 2-4. General Reset Timing



TL/EE/9142-8

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

2.0 Functional Description (Continued)

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 NS32332, are shown in *Figure A-1* (Appendix A).

The CPU issues 32-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 32-bit physical addresses on the Physical Address bus, pulsing the $\overline{PAV}$ line low. The $\overline{PAV}$ pulse triggers the address latches and signals the NS32C201 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.

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.

The MMU ignores all other status codes. 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 and Operation Word

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, and for maintaining the contents of the status bits (R and M) in the Page Table Entries.

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.)

2.4.2 CPU-Initiated Bus Cycles

A CPU-initiated bus cycle is performed in a minimum of four clock cycles: T1, 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 in the previous T4 cycle to determine how the CPU intends to use the bus.

During period T2 the CPU removes the virtual address from the bus 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 on PA0–PA31 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.

Whenever the MMU generates an Abort pulse on the $\overline{RST}/\overline{ABT}$ pin, the CPU enters state T3 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.

2.0 Functional Description (Continued)

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:

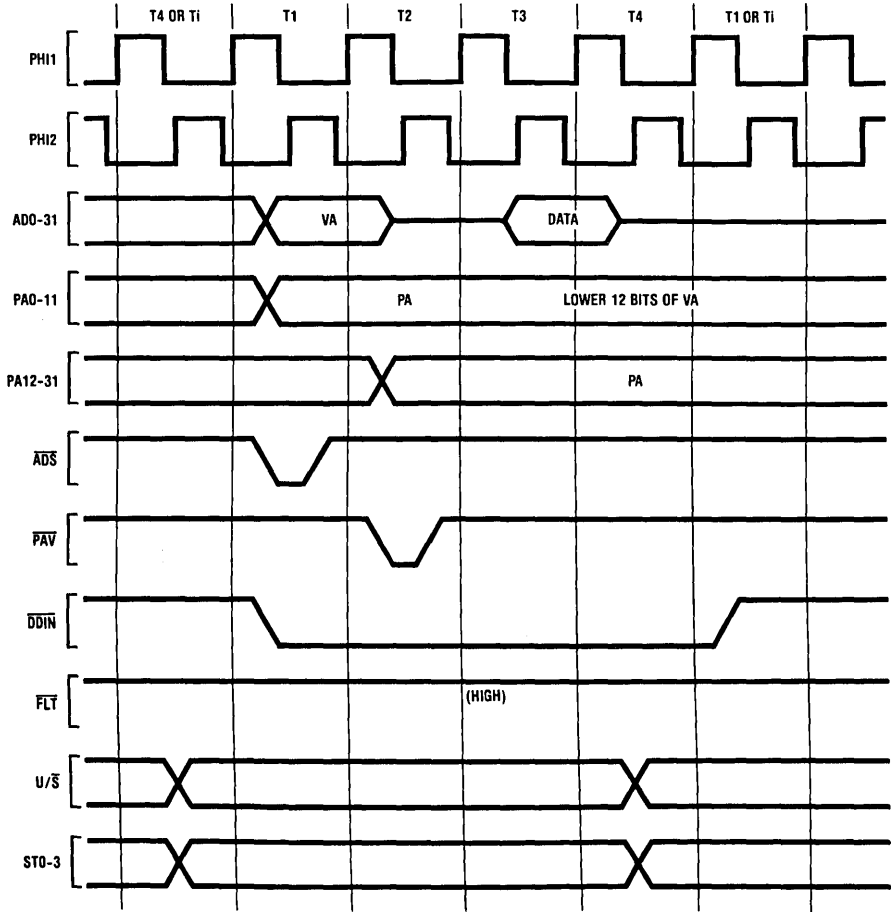
1) There is no translation in the MMU's TLB for the virtual address issued by the CPU, meaning that the MMU must reference the Page Tables in memory to obtain the translation.
2) There is a translation for that virtual address in the TLB, but the page is being written for the first time (the M bit in 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 T3.

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 two consecutive memory Read cycles. 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, $\overline{BE0}$–$\overline{BE3}$, should be handled externally for correct memory referencing.

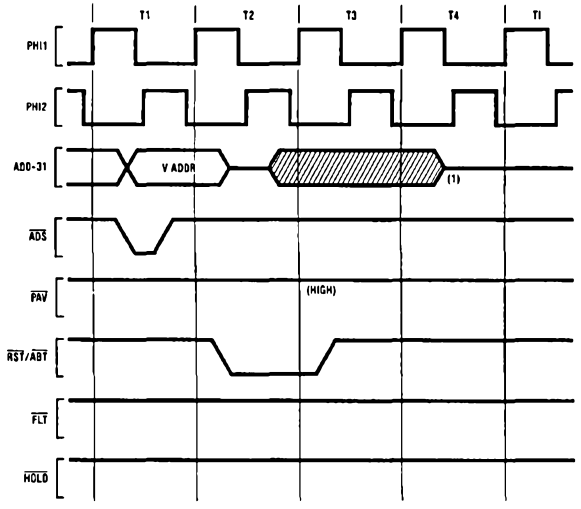
In the clock cycle immediately after T4 of the last lookup cycle, the MMU issues the translated address and pulses $\overline{MADS}$. In the subsequent cycle it removes $\overline{FLT}$ and pulses $\overline{PAV}$ to continue the CPU's access.



TL/EE/9142-9

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

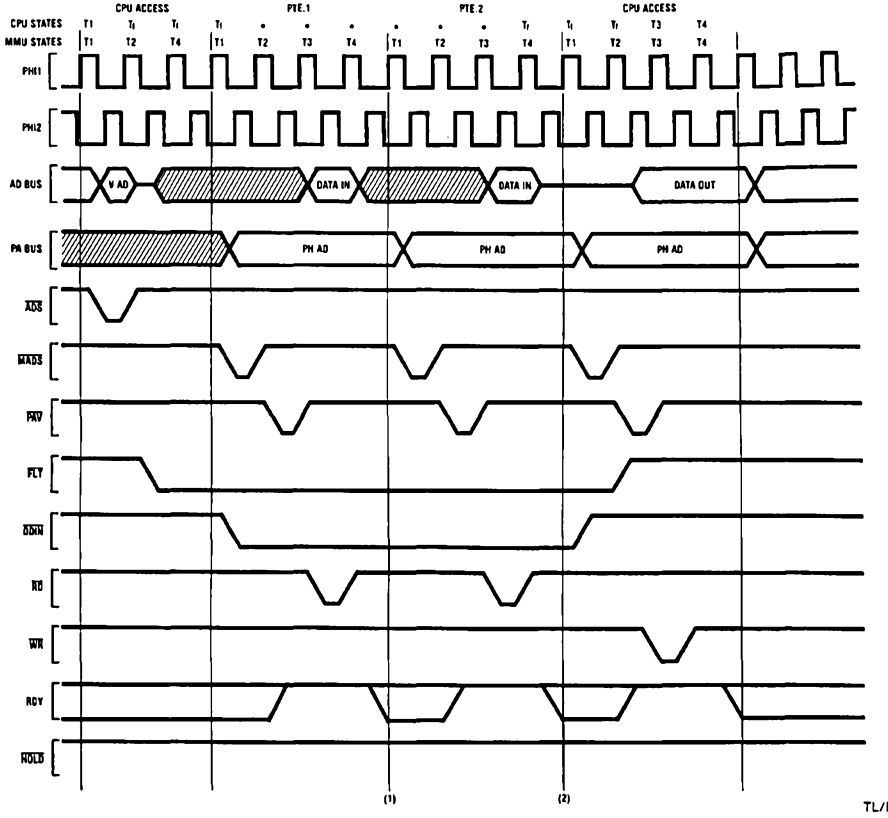
2.0 Functional Description (Continued)



TL/EE/9142-10

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

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



TL/EE/9142-11

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.

FIGURE 2-8. Page Table Lookup

2.0 Functional Description (Continued)

If the V bit (Bit 0) in any of the Page Table Entries 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.

If the R and/or M bit (bit 7 or 8) must be updated, the MMU does this immediately in a single Write cycle. All bits except those updated are rewritten with their original values.

At most, the MMU writes two double 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.

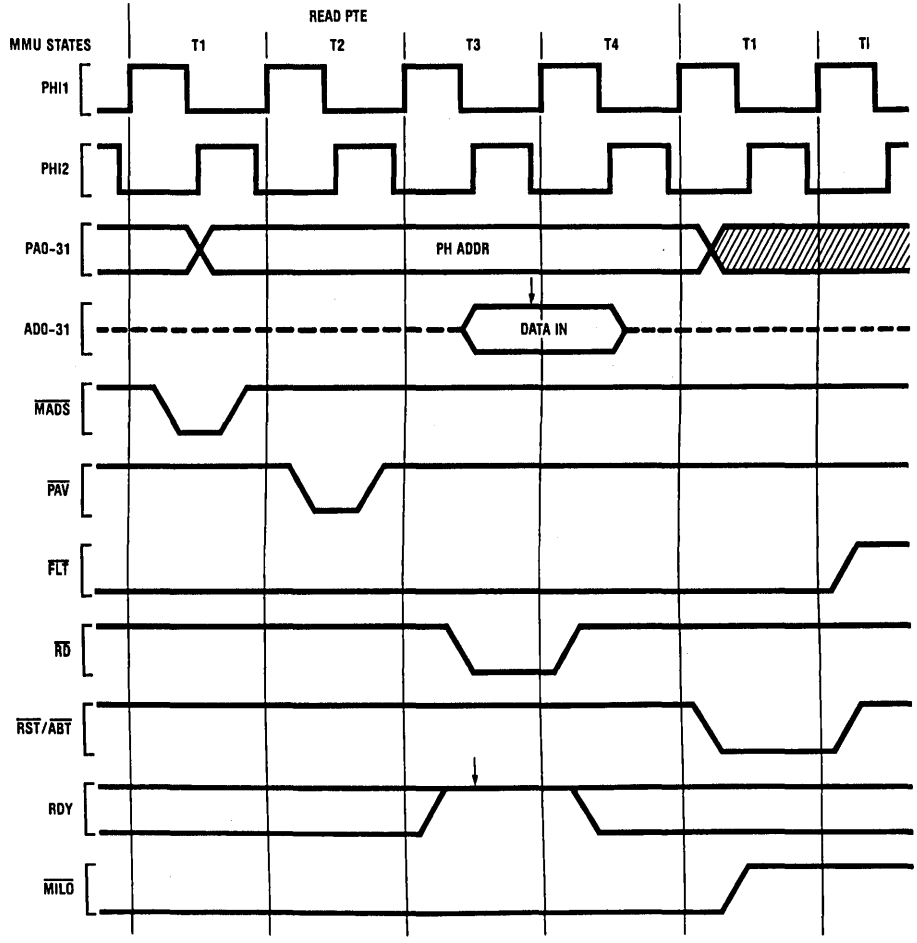
2.4.4 Cycle Extension

To allow sufficient strobe widths and access time requirements for any speed of memory or peripheral device, the NS32382 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.

In the middle of T3, on the falling edge of clock phase PHI1, 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".

The RDY pin is driven by the NS32C201 Timing Control Unit, which applies WAIT states to the CPU and MMU as requested on its own WAIT request input pins.



TL/EE/9142-12

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

2.0 Functional Description (Continued)

2.4.5 Bus Retry

The Bus Retry input signal ($\overline{BRT}$) provides a system with the capability of repeating a bus cycle upon the occurrence of a "soft" or correctable error. The system first determines that a correctable error has occurred and then activates the $\overline{BRT}$ input. The MMU then samples this input on the falling edge of PHI1 in both T3 and T4 of a bus cycle. A valid bus retry will be issued as a result of a low being sampled in both T3 and T4.

If the MMU gets a Bus Retry when it is controlling the bus, it will re-run the bus cycle until $\overline{BRT}$ is deactivated.

Any Pending Hold request will not be acknowledged by the MMU if a bus retry is detected and during Hold Acknowledge, the MMU will not recognize the Bus Retry signal.

2.4.6 Bus Error

The Bus Error input signal $\overline{BER}$ will be activated (low) when a "hard" or uncorrectable error occurs within the system (e.g. bus timeout, double ECC error). $\overline{BER}$ will be sampled on the falling edge of PHI1 in T4. If the MMU detects Bus Error while it is controlling the bus, it will store the virtual address which caused the error in the BEAR (Bus Error Address Register), and set the ME bit in the MSR to indicate MMU ERROR. An abort signal $\overline{ABT}$ will be generated and further memory accesses by the MMU will be inhibited. The 32382 then returns bus control to the CPU by releasing the $\overline{FLT}$ signal, ($\overline{FLT}$ = 1). Any pending Hold request will not be acknowledged by the MMU if a bus error is detected.

If the Bus Error signal is received when the CPU is controlling the bus, the MMU will store the virtual address in BEAR, and set the CE bit in the MSR to indicate CPU ERROR.

During the Hold Acknowledge, the MMU will ignore the $\overline{BER}$ signal.

2.4.7 Interlocked Bus Transfers

Both the 32332 CPU and the 32382 MMU are capable of executing interlocked cycles to access a stream of data from memory without intervention from other devices.

Before executing an interlocked access, the 32332 CPU performs a dummy read with Read-Modify-Write status (1011). The MMU handles the dummy read as if it were a real RMW access. The TLB entries will be searched and page table look-up will be performed if a miss occurs. The access level is checked and the CPU will be aborted if write privilege is not currently assigned. The Reference (R) and the Modify (M) bits in the first and second level PTEs, as well as those in the Translation look-aside Buffer, will be updated. By executing the dummy read, the CPU is assured of no MMU intervention when the actual interlocked access is performed.

The 32382 MMU executes interlocked Read-Modify-Write memory cycles to access Page Table Entries (PTEs) and update the Reference (R) and Modify (M) bit in the PTEs when necessary. If the R and/or M bit(s) do not require updating, the write portion of the RMW cycle will not be executed. The memory cycles to access PTEs during execution of RDVAL and WRVAL instructions are not interlocked since R and M bits are not updated.

During interlocked access cycles, the $\overline{MILO}$ signal from the MMU will be asserted. $\overline{MILO}$ has the same timing as $\overline{ILO}$ from the CPU. $\overline{MILO}$ is asserted in the clock cycle immediately before the Read-Modify-Write access and de-activated in the clock cycle following T4 of the write cycle.

The write portion of the Read-Modify-Write access will not be executed if any one of the following conditions occurs:

(1) A bus error has occurred in the read portion of the interlocked access.

(2) The R and/or M bit(s) in the PTE(s) do not require updating.

(3) A protection violation has occurred.

(4) An invalid PTE is detected.

If a bus retry is encountered in an interlocked access, $\overline{MILO}$ will continue to be asserted, and the access will be retried.

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

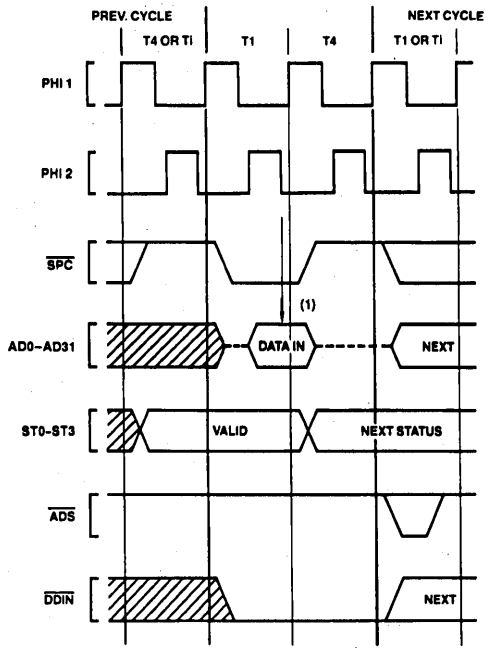
The interconnections between the CPU and MMU for Slave Processor communication are shown in *Figure A-1* (Appendix A). The $\overline{SPC}$ signal is pulsed by the CPU as a low-active data strobe for Slave Processor transfers. Since $\overline{SPC}$ is normally in a high-impedance state, it must be pulled 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 double-word (32 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.

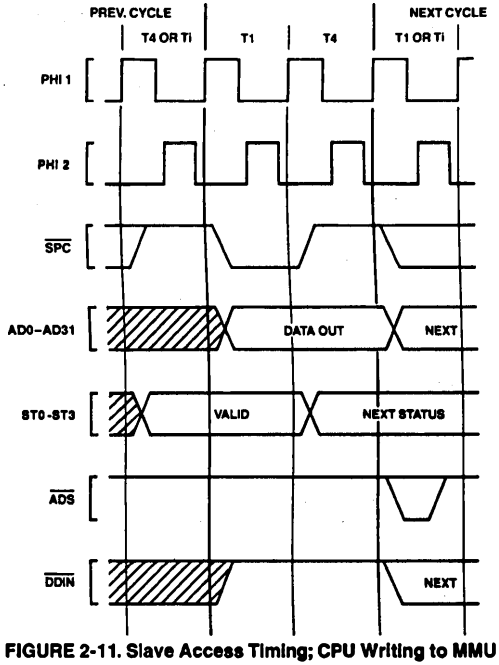
2.0 Functional Description (Continued)



TL/EE/9142–13

Note 1: CPU samples Data Bus here.

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



TL/EE/9142–14

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

2.0 Functional Description (Continued)

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 the ID Byte and the Operation Word, using Slave Processor bus cycles. While applying status code IIII, the CPU transfers the ID byte on bits AD24–AD31, the operation word on bits AD8–AD23 in a swapped order of bytes and a non-used byte XXXXXXX1 (X = Don't Care) on bits AD0–AD7.

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.

The CPU then waits for the MMU to signal the completion of the instruction by pulsing $\overline{\text{SDONE}}$ low.

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 $\overline{\text{SDONE}}$ low) that it is ready to present the specified register's contents to the CPU. Upon receiving this "Done" pulse, the CPU reads the contents of the selected register in one Slave Processor bus cycle, 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 first performs the effective address calculation and obtains the address to be validated. It then issues the ID Byte and the Operation Word to the MMU. It initiates 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 $\overline{\text{SDONE}}$ for two clock cycles to acknowledge that the instruction may continue execution and an MMU status read is required.

TABLE 2-2. LMR Instruction Protocol

CPU Action	Status	MMU Action
Issues ID Byte and Operation Word, pulsing $\overline{\text{SPC}}$.	1111	Accepts and decodes instruction.
Accesses memory for effective address calculation and operand fetching or instruction prefetching.	XXXX	Translates CPU addresses.
Issues operand value to MMU, pulsing $\overline{\text{SPC}}$.	1101	Accepts operand value from bus; places it into referenced MMU register.
Waits for $\overline{\text{SDONE}}$ pulse from MMU.	0011	Sends completion signal by pulsing $\overline{\text{SDONE}}$ low.

TABLE 2-3. SMR Instruction Protocol

CPU Action	Status	MMU Action
Issues ID Byte and Operation Word, pulsing $\overline{\text{SPC}}$.	1111	Accepts and decodes instruction.
Accesses memory for effective address calculation or instruction prefetching.	XXXX	Translates CPU addresses.
Waits for $\overline{\text{SDONE}}$ pulse from MMU.	0011	Sends completion signal by pulsing $\overline{\text{SDONE}}$ low.
Reads results from MMU, pulsing $\overline{\text{SPC}}$.	1101	Presents data value from referenced MMU register on bus.

TABLE 2-4. RDVAL/WRVAL Instruction Protocol

CPU Action	Status	MMU Action
Performs effective address calculation and obtains address to be validated.	XXXX	Translates CPU addresses.
Issues ID Byte and operation word, pulsing $\overline{\text{SPC}}$.	1111	Accepts and decodes instruction.
CPU may prefetch instructions.	XXXX	
Performs dummy one-byte memory read from operand's location.	1010	Translates CPU address, using User-Mode mapping, and performs requested test on the address presented by the CPU. Aborts the CPU if there is no protection violation and the level-1 page table entry is invalid. Aborts on protection violations are temporarily suppressed.
Waits for $\overline{\text{SDONE}}$ pulse from MMU	XXXX	Pulses $\overline{\text{SDONE}}$ low for two clock cycles.
Sends $\overline{\text{SPC}}$ 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.

2.0 Functional Description (Continued)

The CPU then reads a status word from the MMU. 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.

2.6 BUS ACCESS CONTROL

The NS32382 MMU has the capability of relinquishing its access to the bus upon request from a DMA device. It does this by using $\overline{\text{HOLD}}$, $\overline{\text{HLDAI}}$ and $\overline{\text{HLDAO}}$.

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

Requests for DMA are presented in parallel to both the CPU and MMU on the $\overline{\text{HOLD}}$ 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 $\overline{\text{HLDA}}$ signal to its own $\overline{\text{HLDAO}}$ pin. When the MMU grants the bus, it does so by activating its $\overline{\text{HLDAO}}$ pin directly, and the CPU is not involved. $\overline{\text{HLDAI}}$ in this case is ignored.

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

2.7 BREAKPOINTING

The MMU provides the ability to monitor references to memory locations in real time, generating a Breakpoint trap on occurrence of any type of reference made by a program to a specified virtual address or range of addresses.

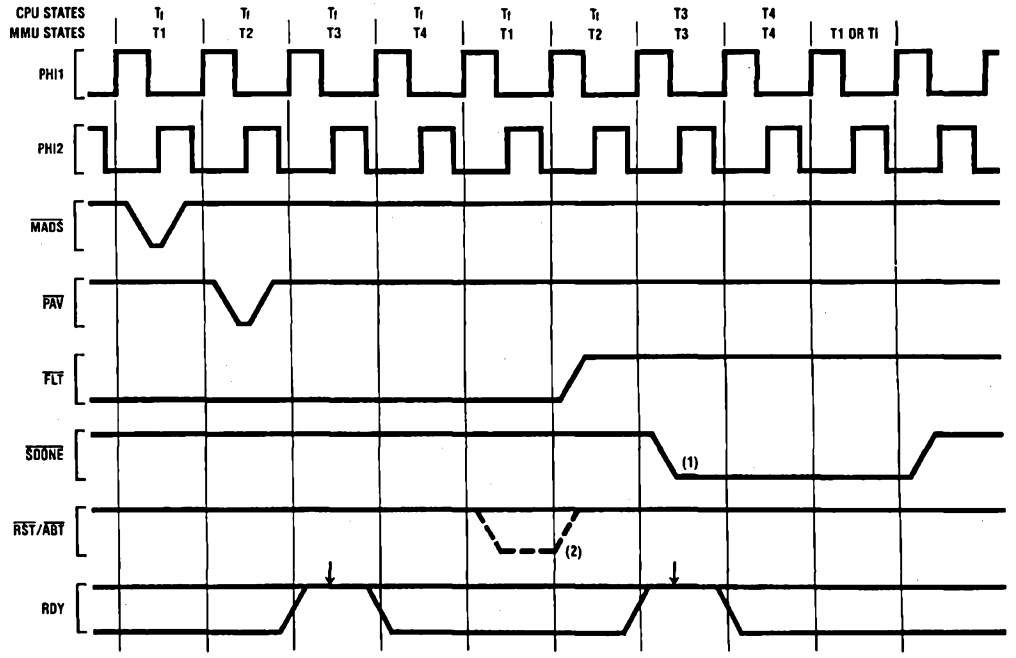
Breakpoint monitoring is enabled and regulated by the setting of appropriate bits in the BAR, BMR, BDR, MCR and MSR registers. See Sections 3.7 through 3.11.

The MMU compares the 32-bit address stored in the BAR register with the virtual address from the CPU. Selected bits can be masked off by the data pattern stored in the BMR register. Only those bit positions which are set in the BMR register will be used in the comparison process, bit positions which are cleared become "Don't Cares".

If a Breakpoint condition is detected, an abort will be issued to the CPU and the BP bit in the MSR register will be set. The virtual address that triggered the Breakpoint is then stored in the BDR register.

The dummy read addresses generated by the CPU during RDVAL/WRVAL operations, are not subject to Breakpoint address comparison. See Section 2.5.2.

When a Breakpoint is enabled, the NS32332 burst cycles should be inhibited by keeping the $\overline{\text{BIN}}$ signal high. The reason being that the CPU addresses are not incremented during burst. It is therefore possible for the CPU to skip over the address specified in the BAR register during burst cycle.



TL/EE/9142-15

Note 1: If there is a protection violation or an invalid Level-2 PTE then $\overline{\text{SDONE}}$ is issued two clock cycles earlier in T1.

Note 2: If there is no protection violation and the Level-1 PTE is not valid, an abort is generated and $\overline{\text{SDONE}}$ is not pulsed.

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

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 3.14, 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 LEVEL-1 Page Tables referenced by the MMU for address translation. See Section 3.3.

IVAR0, IVAR1—Invalidate Virtual Address Registers. These WRITE-ONLY registers are used to remove invalid Page Table Entries from the Translation Buffer.

TEAR—Translation Exception Address Registers. This register contains the virtual address which caused the translation exception.

BEAR—Bus Error Address Register. This register contains the virtual address which triggered the bus error.

BAR—Breakpoint Address Register. Used to hold a virtual address for breakpoint address comparison.

BMR—Breakpoint Mask Register. The contents of this register indicate which bit positions of the virtual address are to be compared.

BDR—Breakpoint Data Register. This register contains the virtual address that triggered a breakpoint.

MCR—Memory Management Control Register. Contains the control field for selecting the various features provided by the MMU.

MSR—Memory Management Status Register. Contains basic status fields for all MMU functions. See Section 3.11.

3.2 MEMORY MANAGEMENT FUNCTIONS

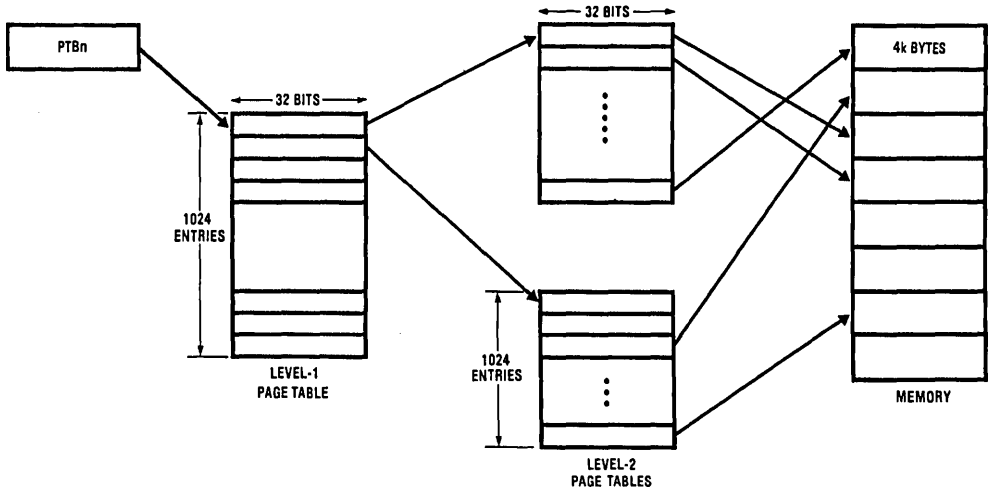
The NS32382 uses sets of tables in physical memory (the "Page Tables") to define the mapping from virtual to physical addresses. These tables are found by the MMU using one of its two Page Table Base registers: PTB0 or PTB1. Which register is used depends on the currently selected address space. See Section 3.2.2.

3.2.1. Page Tables 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 1024 32-bit Page Table Entries (PTE's) and therefore occupies 4 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 20-bit PFN field, providing bits 12-31 of the physical address. The remaining bits (0-11) are assumed zero, placing a Level-2 Page Table always on a 4 Kbyte (page) boundary.



TL/EE/9142-18

FIGURE 3-1. Two-Level Page Tables

3.0 Architectural Description (Continued)

Level-2 Page Tables contain 1024 32-bit Page Table entries, and so occupy 4 Kbytes (1 page). Each Level-2 Page Table Entry points to a final 4 Kbyte physical page frame. In other words, its PFN provides the Page Frame Number portion (bits 12-31) 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 MCR (Sec. 3.10) 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 MCR 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 MCR, 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 MCR (Sec. 3.10).

3.2.3 Page Table Entry Formats

Figure 3-2 shows the formats of Level-1 and Level-2 Page Table Entries (PTE's).

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.

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

NU Not Used. These bits are reserved by National for future enhancements. Their values should be set to zero.

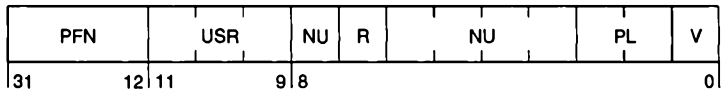
CI Cache Inhibit. This bit appears only in Level-2 PTE's. It is used to specify non-cacheable pages.

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.

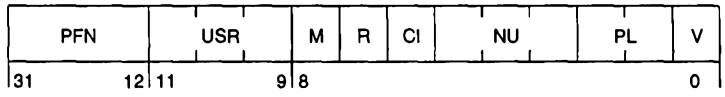
R = 1 => The page has been referenced since the R bit was last cleared.

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

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.



First Level PTE



Second Level PTE

FIGURE 3-2. Page Table Entries (PTE's)

3.0 Architectural Description (Continued)

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 unaltered.

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

They can be used by the user software.

PFN Page Frame Number. This 20-bit field provides bits 12-31 of the physical address. See *Figure 3-3*.

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*.

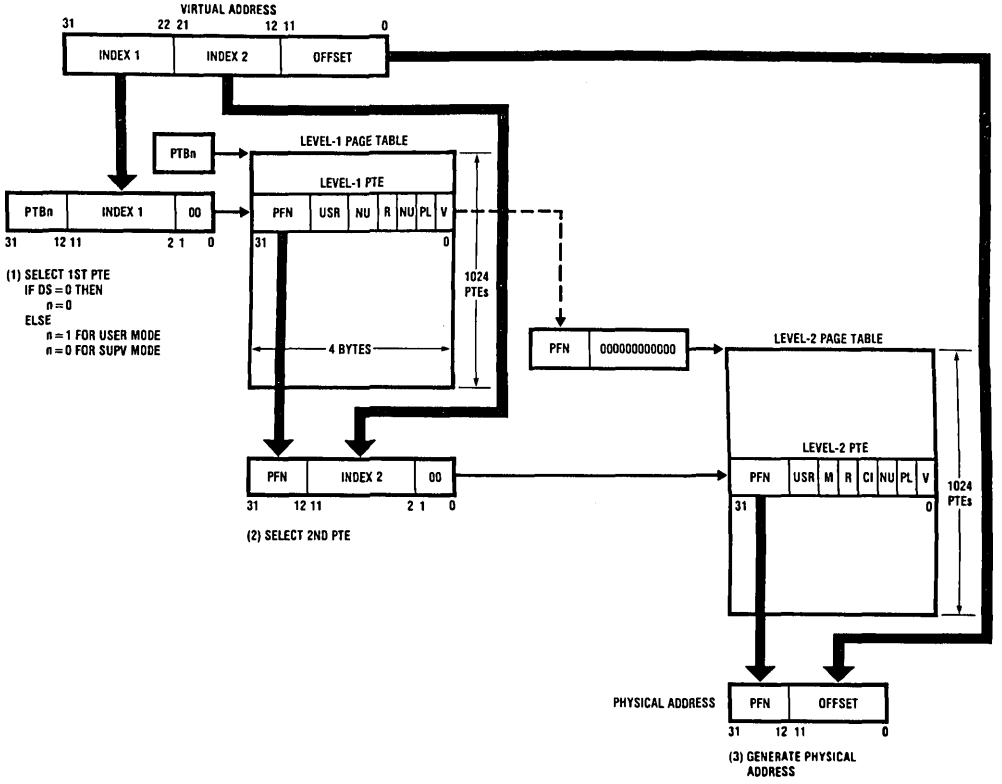
Bits 12–31 of the virtual address hold the 20-bit Page Number, which in the course of the translation is replaced with the 20-bit Page Frame Number of the physical address. The virtual Page Number field is further divided into two fields, INDEX 1 and INDEX 2.

Bits 0–11 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 0–11 of the final physical address.

The 10-bit INDEX 1 field of the virtual address is used as an index into the Level-1 Page Table, selecting one of its 1024 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 field of that entry gives the base address of the selected Level-2 Page Table.

The INDEX 2 field of the virtual address (10 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 field of the selected entry provides 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/9142–20

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 12 bits are permanently zero, so that each register always points to a 4 Kbyte boundary in memory.

The PTBn registers may be loaded or stored using the MMU Slave Processor instructions LMR and SMR (Section 3.14).

3.4 INVALIDATE VIRTUAL ADDRESS REGISTERS (IVAR0, IVAR1)

The Invalidate Virtual Address registers are write-only registers. When a virtual address is written to IVAR0 or IVAR1 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.

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 MCR TU and/or TS bits) does not purge any entries from the TLB.

The format of the $IVAR_n$ registers is shown in *Figure 3-5.*

3.5 TRANSLATION EXCEPTION ADDRESS REGISTER (TEAR)

The TEAR Register is loaded when a translation exception occurs. It contains the 32-bit virtual address which caused the translation exception and is a read-only register. TEAR has the same format as the IVARn registers of *Figure 3-5.*

For more details on the updating of TEAR, refer to the note at the end of Section 3.11.

3.6 BUS ERROR ADDRESS REGISTER (BEAR)

The BEAR Register is loaded when a CPU or MMU bus error occurs. It contains the 32-bit virtual address which triggered the bus error and is a read-only register. BEAR has the same format as the IVARn registers of *Figure 3-5.*

3.7 BREAKPOINT ADDRESS REGISTER (BAR)

The Breakpoint Address Register is used to hold a virtual address for breakpoint address comparison during instruction and operand accesses. It is 32 bits in length and its format is shown in *Figure 3-6.*

3.8 BREAKPOINT MASK REGISTER (BMR)

The Breakpoint Mask Register provides corresponding bit positions for each of the virtual address bits that are to be compared when the Breakpoint Address Compare Function is enabled. Bits which are set in this register are used for matching virtual address bits while bits which are cleared are treated as "don't cares". This allows a breakpoint to be generated upon an access to any location within a block of addresses. The BMR Register format is shown in *Figure 3-6.*

3.9 BREAKPOINT DATA REGISTER (BDR)

The Breakpoint Data Register holds the virtual address that triggered the breakpoint.

It is a read-only register and its format is shown in *Figure 3-6.*

3.10 MEMORY MANAGEMENT CONTROL REGISTER (MCR)

The MCR Register controls the various features provided by the MMU. It is 32 bits in length and has the format shown in *Figure 3-7.* All bits will be cleared on reset. The bits 8 to 31 are RESERVED for future use and must be loaded with zeros.

When MCR is read as a 32-bit word, bits 8 to 31 will be returned as zeros. Details on the MCR bits are given below.

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.

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.

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.

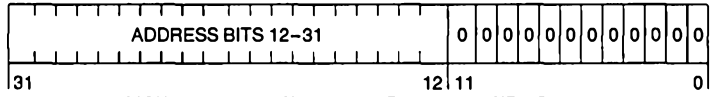


FIGURE 3-4. Page Table Base Registers (PTB0, PTB1)

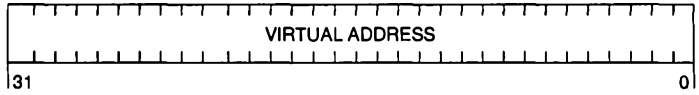


FIGURE 3-5. Address Registers (IVAR0, IVAR1, TEAR, BEAR)

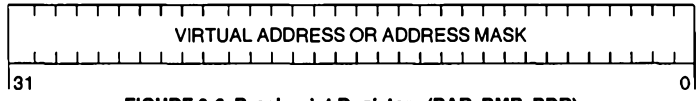


FIGURE 3-6. Breakpoint Registers (BAR, BMR, BDR)

3.0 Architectural Description (Continued)

AO Access Level Override. This bit may be set to temporarily cause User Mode accesses to be given Supervisor Mode privilege. See Section 3.13.

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.

BE Break on Execution. If BE is 1, a break is generated when the instruction at the breakpoint address is fetched. If BE is 0, this condition is disabled.

BAS Breakpoint Address Space. This bit selects the address space for breakpointing.

BAS = 0 Selects Address Space 0 (PTB0).

BAS = 1 Selects Address Space 1 (PTB1).

3.11 MEMORY MANAGEMENT STATUS REGISTER (MSR)

The Memory Management Status Register provides status information for translation exceptions as well as bus errors.

When either a translation exception or a bus error occurs, the corresponding bits in the MSR are updated.

The MSR register can be loaded with an LMR instruction. Its format is shown in *Figure 3-8*. Bits 19 through 31 are reserved for future use and are returned as zeros when read. Bits 8 and 18 are also reserved.

Upon reset, all MSR bits are cleared to zero. Details on the function of each bit are given below.

TEX Translation Exception. This 2-bit field specifies the cause of the current address translation exception. Combinations appearing in this field are summarized below.

00 No Translation Exception

01 First Level PTE Invalid

10 Second Level PTE Invalid

11 Protection Violation

Note: During address translation, if a protection violation and an invalid PTE are detected at the same time, the TEX field is set to indicate a protection violation.

DDT Data Direction. This bit indicates the direction of the transfer that the CPU was attempting when the translation exception occurred.

DDT = 0 = > Read Cycle

DDT = 1 = > Write Cycle

UST User/Supervisor. This is the state of the U/$\overline{S}$ pin from the CPU during the access cycle that triggered the translation exception.

STT CPU Status. This 4-bit field is set on an address translation exception to the value of the CPU Status Bus (ST0–ST3).

BP Break. This bit is set to indicate that a breakpoint condition has been detected by the MMU.

CE CPU Error. This bit is set when a bus error occurs while the CPU is in control of the bus.

ME MMU Error. This bit is set when a bus error occurs while the MMU is in control of the bus.

DDE Data Direction. This bit indicates the direction of the transfer that the CPU was attempting when the bus error occurred.

DDE = 0 = > Read Cycle

DDE = 1 = > Write Cycle

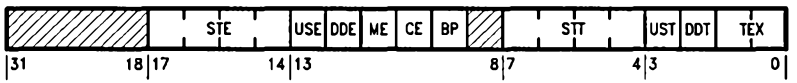
USE User/Supervisor. This is the state of the U/$\overline{S}$ pin from the CPU during the access cycle that triggered the bus error.

STE CPU Status. This 4-bit field is set to the value of the CPU status bus (ST0–ST3) when a bus error is detected.

Note: The MSR and TEAR registers are updated whenever a translation exception occurs, regardless of whether a CPU abort will result. As a consequence, after an abort is recognized, MSR and TEAR may be overwritten with new data and thus the original contents may be lost. This happens if the CPU, while executing the abort routine, performs instruction prefetch cycles from an invalid page. To ensure correct operation the reading of MSR and TEAR should be performed before any instruction prefetch crosses a page boundary, unless the next page is valid. This may place some restrictions in the relocation of the abort routine.

TL/EE/9142-24

FIGURE 3-7. Memory Management Control Register (MCR)



TL/EE/9142-25

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

3.0 Architectural Description (Continued)

3.12 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 (20 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 (20 bits) is loaded from the PFN field of the Level-2 Page Table Entry.

The CI and M bits are 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.)

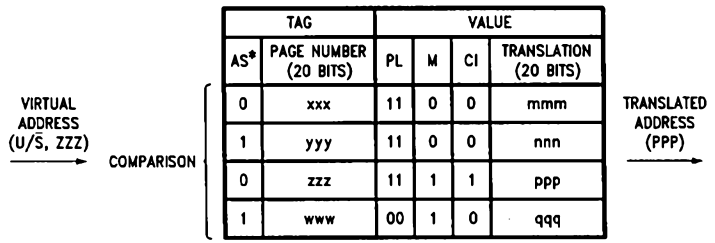
When a virtual address is presented to the MMU for translation, the high-order 20 bits (page number) and the Address Space qualifier are compared associatively to the corresponding fields in all entries of the TLB. When the Tag portion of a TLB entry completely matches the input values, the Value portion 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 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 one of the IVARn registers 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 MCR TU and TS bits) does not affect the contents of the TLB.

3.13 ADDRESS TRANSLATION ALGORITHM

The MMU either translates the 32-bit virtual address to a 32-bit physical address or reports a translation error. This process is described algorithmically in the following pages. See also *Figure 3-3.*



TL/EE/9142-26

FIGURE 3-9. TLB Model

*AS represents the virtual address space qualifier.

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/S is high)
     W (1 if DDIN input is high)
     VA  Virtual address consisting of:
        INDEX_1 (from pins A31-A22)
        INDEX_2 (from pins A21-A12)
        OFFSET  (from pins A11-A0)
     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 MCR.A0  (U from U/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 PA0-PA31;
     CI Cache Inhibit Signal
     Abort pulse on RST/ABT pin.
Uses:
     MCR        Control 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 and R
     PTEP_1     Pointer, holding address of PTE_1
     PTE_2      Level-2 Page Table Entry:
                       fields PFN, PL, V, M, R and CI
     PTEP_2     Pointer, holding address of PTE_2
IF ( (MCR.TU == 0) AND (U == 1) )  OR  ( (MCR.TS == 0) AND (U ==0) )  If translation not enabled then echo
     THEN  {  PA(0:31) = VA(0:31) ; CINH PIN = 0 ; DONE  }  ;              virtual address as physical address.
     ,

IF (MCR.DS == 1) AND (U == 1)                                        If Dual Space mode and CPU in User Mode
   THEN {  PTEP_1(31:12) = PTB1(31:12) ;                                  then form Level-1 PTE address
           PTEP_1(11:2) = VA.INDEX_1 ; PTEP_1(1:0)=0  }                         from PTB1 register,
   ELSE {  PTEP_1(31:12) = PTB0(31:12) ;                                  else form Level-1 PTE address
           PTEP_1(11:2) = 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,

PTEP_2(31:11) = PTE_1.PFN ;                                                    and form Level-2 PTE address.
PTEP_2(11:2) = 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(31:11) = PTE_2.PFN ; PA(11:0) = VA.OFFSET ; CINH = PTE_2.CI ;                 and generate physical address.
DONE ;
```

3.0 Architectural Description (Continued)

3.14 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 3-2 shows the encodings of the "short" field for selecting the various MMU internal registers.

TABLE 3-2. "Short" Field Encodings

"Short" Field	Register
0000	BAR
0001	RESERVED
0010	BMR
0011	BDR
0110	BEAR
1001	MCR
1010	MSR
1011	TEAR
1100	PTB0
1101	PTB1
1110	IVAR0
1111	IVAR1

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.

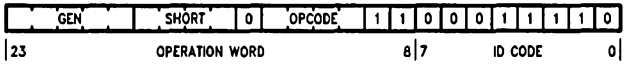
Note: The Series 32000 Dual-Space Move instructions (MOVSUi and MOVUSi), although they involve memory management action, are not Slave Processor instructions. The CPU implements them by switching the state of its U/$\overline{S}$ pin at appropriate times to select the desired mapping and protection from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

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

4.0 Device Specifications

4.1 NS32382 PIN DESCRIPTIONS

The following is a brief description of all NS32382 pins. The descriptions reference portions of the Functional Description, Section 2.0.



TL/EE/8142–27

FIGURE 3-10. MMU Slave Instruction Format

4.0 Device Specifications (Continued)

4.1.1 Supplies

Power (V_{CC}): Eight pins, connected to the +5V supply.

Back Bias Generator (BBG): Output of on-chip substrate voltage generator.

Ground (GND): Eighteen pins, connected to ground.

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.

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

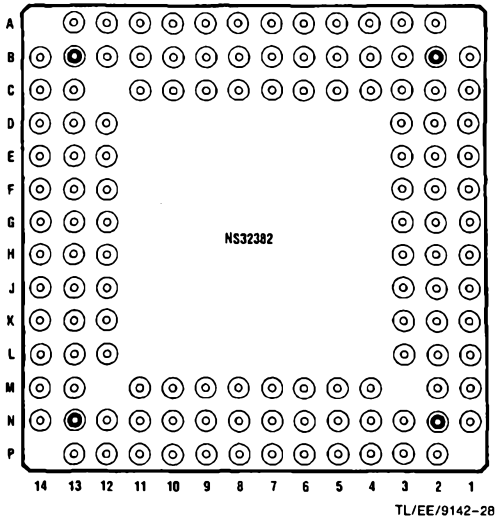
Status Lines (ST0–ST3): Status code input from the CPU. Active from T4 of previous bus cycle through T3 of current bus cycle. Section 2.4.

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.

Bus Error ($\overline{BER}$): Active low. When active, indicates that an error occurred during a bus cycle. Not applicable for slave cycles.

Connection Diagram



Bottom View

FIGURE 4-1. Pin Grid Array Package

Order Number NS32382U-10 or NS32382U-15
See NS Package Number U125A

NS32382 Pinout Descriptions
125 Pin Grid Array

Desc	Pin	Desc	Pin	Desc	Pin	Desc	Pin
NC	A2	V_{CC}	C7	AD22	H1	PA4	M9
$\overline{SPC}$	A3	GND	C8	AD21	H2	PA7	M10
NC	A4	V_{CC}	C9	AD20	H3	GND	M11
$\overline{SDONE}$	A5	V_{CC}	C10	GND	H12	V_{CC}	M13
$\overline{MILO}$	A6	GND	C11	PA22	H13	PA13	M14
$\overline{HLDAI}$	A7	GND	C13	PA21	H14	NC	N1
$\overline{RSTI}$	A8	CINH	C14	AD19	J1	GND	N2
$\overline{BER}$	A9	AD29	D1	AD18	J2	GND	N3
$\overline{BRT}$	A10	AD31	D2	AD17	J3	AD9	N4
$\overline{RST}/\overline{ABT}$	A11	GND	D3	PA20	J12	AD5	N5
ST0	A12	$\overline{ADS}$	D12	PA19	J13	AD2	N6
ST1	A13	RESERVED	D13	PA18	J14	AD0	N7
NC	B1	PA31	D14	AD14	K1	PA0	N8
NC	B2	AD27	E1	AD15	K2	PA3	N9
GND	B3	AD30	E2	AD16	K3	PA6	N10
GND	B4	U/$\overline{S}$	E3	GND	K12	PA9	N11
V_{CC}	B5	PA30	E12	PA17	K13	GND	N12
$\overline{HOLD}$	B6	PA29	E13	PA16	K14	NC	N13
RDY	B7	PA28	E14	AD13	L1	PA12	N14
PHI2	B8	AD25	F1	AD12	L2	AD11	P2
PHI1	B9	AD26	F2	V_{CC}	L3	AD10	P3
$\overline{PAV}$	B10	AD28	F3	V_{CC}	L12	AD8	P4
$\overline{FLT}$	B11	PA27	F12	PA14	L13	AD6	P5
ST2	B12	PA26	F13	PA15	L14	AD4	P6
ST3	B13	PA25	F14	NC	MI	AD1	P7
RESERVED	B14	AD23	G1	GND	M2	PA1	P8
NC	C1	AD24	G2	GND	M4	PA2	P9
$\overline{MADS}$	C2	GND	G3	AD7	M5	PA5	P10
GND	C3	GND	G12	AD3	M6	PA8	P11
GND	C4	PA24	G13	V_{CC}	M7	PA10	P12
$\overline{DDIN}$	C5	PA23	G14	BBG	M8	PA11	P13
$\overline{HLDAO}$	C6						

4.0 Device Specifications (Continued)

Bus Retry ($\overline{BRT}$): Active low. When active, the MMU will re-execute the last bus cycle. Not applicable for slave cycles.

Slave Processor Control ($\overline{SPC}$): Active low. Used as a data strobe for slave processor transfers.

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 to abort the current CPU instruction.

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

Physical Address Valid ($\overline{PAV}$): Active low. Pulse generated during T2 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.

Cache Inhibit (CINH): This output signal reflects the state of the CI bit in the second level Page Table Entry (PTE). It is used to specify non-cacheable pages. During MMU generated bus cycles and when the MMU is in No-Translation mode, CINH will be held low.

Slave Done ($\overline{SDONE}$): Active low. Used by the MMU to inform the CPU of the completion of a slave instruction. It floats when it is not active.

MMU Address Strobe ($\overline{MADS}$): Active low. This signal is asserted in T1 of an MMU initiated cycle. It indicates that the physical address is available on the physical address bus. $\overline{MADS}$ is floated during hold acknowledge.

MMU Interlock ($\overline{MILO}$): Active low. This signal is asserted by the MMU when it performs a read-modify-write operation to up-date the R and/or the M bit in the Page Table Entry (PTE). It is inactive during Hold Acknowledge.

Physical Address Bus (PA0–PA31): These 32 signal lines carry the physical address. They float during Hold Acknowledge.

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/Data 0–31 (AD0–AD31): 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	2.5W

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} − 0.5		V_{CC} + 0.5	V
V_{CL}	Low Level Clock Voltage	PHI1, PHI2 Pins Only	−0.5		0.3	V
V_{CRT}	Clock Input Ringing Tolerance	PHI1, PHI2 Pins Only	−0.5		0.5	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}	$\overline{SPC}$ Input Current (Low)	V_{IN} = 0.4V, $\overline{SPC}$ 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_{OUT} ≤ V_{CC}	−20		20	μA
I_{CC}	Active Supply Current	I_{OUT} = 0, T_A = 25°C		350	500	mA

4.0 Device Specifications (Continued)

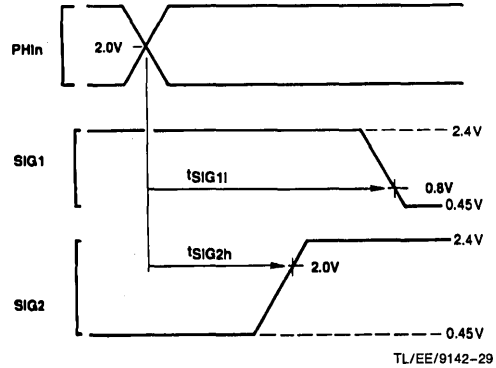
4.4 SWITCHING CHARACTERISTICS

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.

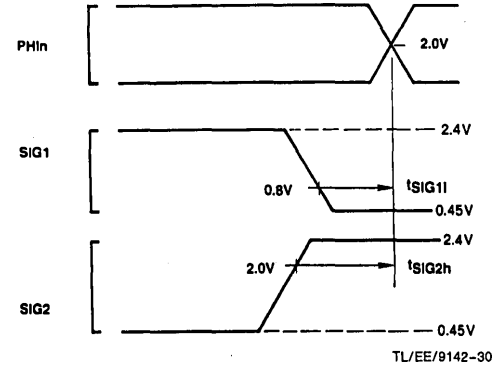
ABBREVIATIONS:

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



TL/EE/9142-29

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



TL/EE/9142-30

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, NS32382-10, NS32382-15.
Maximum times assume capacitive loading of 50 pF.

Name	Figure	Description	Reference/Conditions	NS32382-10		NS32382-15		Units
				Min	Max	Min	Max	
t_{PALv}	4-4	PA0-11 Valid ($\overline{FLT}$ = 1)	After R.E., PHI1 T1		75		50	ns
t_{PAHv}	4-4	PA12-31 Valid ($\overline{FLT}$ = 1)	After R.E., PHI1 T2		30		20	ns
t_{PAVa}	4-4	$\overline{PAV}$ Signal Active	After R.E., PHI1 T2		25		17	ns
t_{PAVia}	4-4	$\overline{PAV}$ Signal Inactive	After R.E., PHI2 T2		40		27	ns
t_{PAVw}	4-4	$\overline{PAV}$ Pulse Width	At 0.8V (Both Edges)	35		22		ns
t_{PALh}	4-4	PA0-11 Hold ($\overline{FLT}$ = 1)	After R.E., PHI1 (Next) T1	0		0		ns
t_{PAHh}	4-4	PA12-31 Hold ($\overline{FLT}$ = 1)	After R.E., PHI1 (Next) T2	0		0		ns
t_{CIv}	4-4, 4-15,	CINH Signal Valid ($\overline{FLT}$ = 1) ($\overline{FLT}$ = 0)	After R.E., PHI1 T2 After R.E., PHI1 T1		40		27	ns
t_{CIh}	4-4	CINH Signal Hold ($\overline{FLT}$ = 1)	After R.E., PHI1 (Next) T2	0		0		
t_{DDINv}	4-5, 4-7, 4-15	$\overline{DDIN}$ Signal Valid ($\overline{FLT}$ = 0)	After R.E., PHI1 T1		35		25	ns
t_{DDINh}	4-5	$\overline{DDIN}$ Signal Hold ($\overline{FLT}$ = 0)	After R.E., PHI1 (Next) T1	0		0		ns
t_{Dv}	4-6	AD0-AD31 Valid (Memory Write)	After R.E., PHI1 T2		50		38	ns
t_{Dh}	4-6	AD0-AD31 Hold (Memory Write)	After R.E., PHI1 (Next) T1	0		0		ns
t_{MAv}	4-6	PA0-31 Valid ($\overline{FLT}$ = 0)	After R.E., PHI1 T1		30		20	ns
t_{MAh}	4-6	PA0-31 Hold ($\overline{FLT}$ = 0)	After R.E., PHI1 (Next) T1	0		0		ns

4.0 Device Specifications (Continued)

4.4.2.1 Output Signals: Internal Propagation Delays, NS32382-10, NS32382-15.
Maximum times assume capacitive loading of 50 pF. (Continued)

Name	Figure	Description	Reference/Conditions	NS32382-10		NS32382-15		Units
				Min	Max	Min	Max	
t_{MADSa}	4-6, 15	$\overline{MADS}$ Signal Active ($\overline{FLT}$ = 0)	After R.E., PHI1 T1		25		17	ns
t_{MADSia}	4-6	$\overline{MADS}$ Signal Inactive	After R.E., PHI2 T1	5	35	5	25	ns
t_{MADSw}	4-6	$\overline{MADS}$ Pulse Width	At 0.8V (Both Edges)	35		22		ns
t_{DDINf}	4-7, 4-9, 11	$\overline{DDIN}$ Floating	After R.E., PHI1 T3 After R.E., PHI1 T1		25		25	ns
t_{MILOa}	4-5, 4-15	$\overline{MILO}$ Signal Active	After R.E., PHI1 T4		50		38	
t_{MILOia}	4-7, 4-15	$\overline{MILO}$ Signal Inactive	After R.E., PHI1 T1 or Ti		50		38	ns
t_{ABTa}	4-8	$\overline{RST}/\overline{ABT}$ Signal Active (Abort)	After R.E., PHI1 T1 or T2		50		40	ns
t_{ABTia}	4-8	$\overline{RST}/\overline{ABT}$ Signal Inactive (Abort)	After R.E., PHI1 T2 or T3	2	50	2	40	ns
t_{ABTw}	4-8	$\overline{RST}/\overline{ABT}$ Pulse Width (Abort)	At 0.8V (Both Edges)	60		40		ns
t_{FLTa}	4-5	$\overline{FLT}$ Signal Active	After R.E., PHI1 T2		50		40	ns
t_{FLTia}	4-7, 4-9	$\overline{FLT}$ Signal Inactive	After R.E., PHI1 T2		40		30	ns
t_{Df}	4-12	Data Bits Floating (Slave Processor Read)	After R.E., PHI1 T4		25		18	
t_{Dv}	4-12	AD0-AD31 Valid (CPU Slave Read)	After R.E., PHI1 T1		50		38	ns
t_{Dh}	4-12	AD0-AD31 Hold (CPU Slave Read)	After R.E., PHI1 T4	4		3		ns
t_{SDNa}	4-14	$\overline{SDONE}$ Signal Active	After R.E., PHI2		50		35	ns
t_{SDNia}	4-14	$\overline{SDONE}$ Signal Inactive	Ater R.E., PHI1		50		35	ns
t_{SDNw}	4-14	$\overline{SDONE}$ Pulse Width	At 0.8V (Both Edges)	25	90	17	60	ns
t_{SDNdw}	4-14	$\overline{SDONE}$ Double Pulse Width	At 0.8V (Both Edges)	225	275	140	180	ns
t_{SDNf}	4-14	$\overline{SDONE}$ Signal Floating	After R.E., PHI2		40		25	ns
t_{HLDAOa}	4-15	$\overline{HLDAO}$ Signal Active ($\overline{FLT}$ = 0)	After R.E., PHI1 Ti		60		40	ns
$t_{HLDAOia}$	4-15	$\overline{HLDAO}$ Signal Inactive ($\overline{FLT}$ = 0)	After R.E., PHI1 T4		60		40	ns
t_{MADSz}	4-15	$\overline{MADS}$ Signal Floated by $\overline{HOLD}$	After R.E., PHI1 Ti		40		25	ns
t_{PAVz}	4-15	$\overline{PAV}$ Signal Floated by $\overline{HOLD}$	After R.E., PHI1 Ti		40		25	ns
t_{PAVr}	4-15	$\overline{PAV}$ Return from Floating (Caused by $\overline{HOLD}$)	After R.E., PHI1 T1		40		25	ns
t_{Dz}	4-15	AD0-AD31 Floating (Caused by $\overline{HOLD}$)	After R.E., PHI1 Ti		25		18	ns
t_{MAz}	4-15	PA0-31 Floated by $\overline{HOLD}$	After R.E., PHI1 Ti		25		18	ns
t_{DDINz}	4-15	$\overline{DDIN}$ Signal Floated by $\overline{HOLD}$	After R.E., PHI1 Ti		40		25	ns
t_{CIz}	4-15	CINH Signal Floated by $\overline{HOLD}$	After R.E., PHI1 Ti		25		18	ns
t_{MILOia}	4-15	$\overline{MILO}$ Signal Inactive by $\overline{HOLD}$ ($\overline{FLT}$ = 0)	After R.E., PHI1 Ti		50		38	ns

4.0 Device Specifications (Continued)

4.4.2.1 Output Signals: Internal Propagation Delays, NS32382-10, NS32382-15.
Maximum times assume capacitive loading of 50 pF. (Continued)

Name	Figure	Description	Reference/Conditions	NS32382-10		NS32382-15		Units
				Min	Max	Min	Max	
t_{MILOa}	4–15	$\overline{MILO}$ Signal Active ($\overline{FLT}$ = 0)	After R.E., PHI1 T4		50		38	ns
t_{HLDAOa}	4–16	$\overline{HLDAO}$ Signal Active ($\overline{FLT}$ = 1)	After R.E., PHI1 Ti		45		30	ns
$t_{HLDAOia}$	4–16	$\overline{HLDAO}$ Signal Inactive ($\overline{FLT}$ = 1)	After R.E., PHI1 Ti or T4		45		30	ns
t_{MADSz}	4–16	$\overline{MADS}$ Signal Floated by $\overline{HLDAI}$ ($\overline{FLT}$ = 1)	After R.E., PHI1 Ti		25		18	ns
t_{MADSr}	4–16	$\overline{MADS}$ Return from Floating ($\overline{FLT}$ = 1)	After R.E., PHI1 Ti or T4		30		20	ns
t_{PAVz}	4–16	$\overline{PAV}$ Signal Floated $\overline{HLDAI}$ ($\overline{FLT}$ = 1)	After R.E., PHI1 Ti		25		18	ns
t_{PAVr}	4–16	$\overline{PAV}$ Return from Floating ($\overline{FLT}$ = 1)	After R.E., PHI1 Ti or T4		30		20	ns
t_{Dz}	4–16	AD0–AD31 Signals Floating ($\overline{FLT}$ = 1)	After R.E., PHI1 Ti		25		18	ns
t_{Dr}	4–16	AD0–AD31 Return from Floating ($\overline{FLT}$ = 1)	After R.E., PHI1 Ti or T4		30		20	ns
t_{MAz}	4–16	PA0–31 Signals Floated by $\overline{HLDAI}$ ($\overline{FLT}$ = 1)	After R.E., PHI1 T1		25		18	ns
t_{MAr}	4–16	PA0–31 Return from Floating ($\overline{FLT}$ = 1)	After R.E., PHI1 Ti or T4		30		20	ns
t_{Clz}	4–16	CINH Signal Floated by $\overline{HLDAI}$ ($\overline{FLT}$ = 1)	After R.E., PHI1 Ti		25		18	ns
t_{Clr}	4–16	CINH Return from Floating ($\overline{FLT}$ = 1)	After R.E., PHI1 Ti or T4		30		20	ns
t_{RSTOa}	4–18	$\overline{RST}/\overline{ABT}$ Signal Active (Reset)	After R.E., PHI2 Ti		50		40	ns
t_{RSTOia}	4–18	$\overline{RST}/\overline{ABT}$ Signal Inactive (Reset)	After R.E. PHI2 Ti		50		40	ns
t_{RSTOw}	4–18	$\overline{RST}/\overline{ABT}$ Pulse Width (Reset)	At 0.8V (Both Edges)	64		64		t_{cp}

4.4.2.2 Input Signal Requirements: NS32382-10, NS32382-15

Name	Figure	Description	Reference/Conditions	NS32382-10		NS32382-15		Units
				Min	Max	Min	Max	
t_{Dis}	4–5	Input Data Setup ($\overline{FLT}$ = 0)	Before F.E., PHI2 T3	12		10		ns
t_{Dih}	4–5	Input Data Hold ($\overline{FLT}$ = 0)	After R.E., PHI1 T4	3		3		ns
t_{RDYs}	4–5	RDY Setup	Before F.E., PHI1 T3	20		12		ns
t_{RDYh}	4–5	RDY Hold	After R.E., PHI2 T3	4		3		ns
t_{SPCs}	4-12	$\overline{SPC}$ Input Setup	Before F.E., PHI2 T1	45		35		ns
t_{SPCh}	4-12	$\overline{SPC}$ Input Hold	After R.E., PHI1 T4	0		0		ns
t_{USs}	4–4, 4-12	U/$\overline{S}$ Setup	Before F.E., PHI2 T4	25		20		ns
t_{USh}	4–4, 4–12	U/$\overline{S}$ Hold	After R.E., PHI1 (Next) T4	0		0		ns
t_{STs}	4–4, 4–12	ST0–3 Setup	Before F.E., PHI2 T4	40		25		ns
t_{STh}	4–4, 4–12	ST0–3 Hold	After R.E., PHI1 (Next) T4	0		0		ns
t_{Dis}	4–13	Data In Setup (Slave Processor Write)	Before F.E., PHI2 T1	40		22		ns

4.0 Device Specifications (Continued)

4.4.2.2 Input Signal Requirements: NS32382-10, NS32382-15 (Continued)

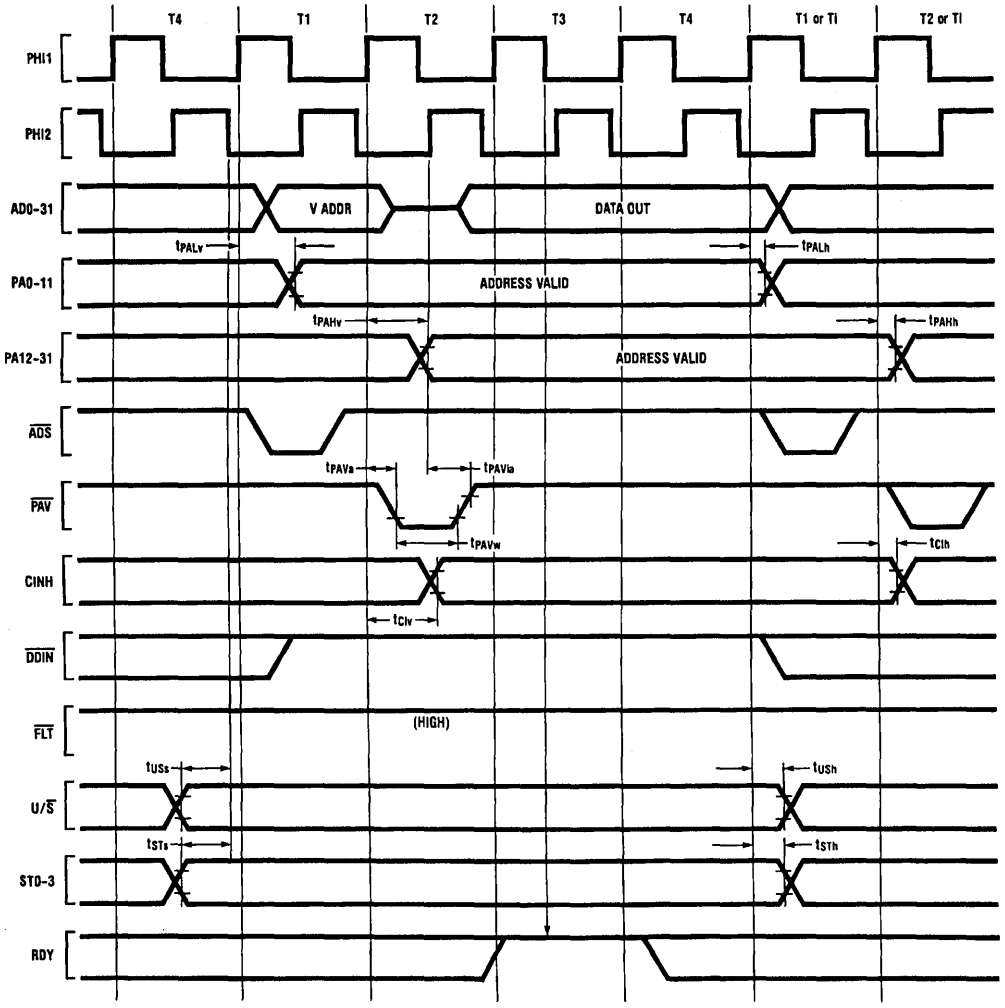
Name	Figure	Description	Reference/Conditions	NS32382-10		NS32382-15		Units
				Min	Max	Min	Max	
t_{DIh}	4-13	Data In Hold (Slave Processor Write)	After R.E., PHI1 (Next) Ti	3		3		ns
t_{HOLDs}	4-15	$\overline{HOLD}$ Setup ($\overline{FLT}$ = 0)	Before F.E., PHI2 T3	15		15		ns
t_{HOLDh}	4-15	$\overline{HOLD}$ Hold ($\overline{FLT}$ = 0)	After R.E., PHI1 T4	0		0		ns
t_{HLDAis}	4-16	$\overline{HLDAI}$ Signal Setup ($\overline{FLT}$ = 1)	Before F.E., PHI2 Ti	25		15		ns
t_{HLDAih}	4-16	$\overline{HLDAI}$ Signal Hold ($\overline{FLT}$ = 1)	After R.E., PHI1 Ti or T4	0		0		ns
t_{BRTs}	4-10	$\overline{BRT}$ Signal Setup ($\overline{FLT}$ = 0)	Before F.E., PHI1 T3 or T4	25		14		ns
t_{BRTh}	4-10	$\overline{BRT}$ Signal Hold ($\overline{FLT}$ = 0)	After R.E., PHI2 T3 or T4	0		0		ns
t_{BERs}	4-11	$\overline{BER}$ Signal Setup ($\overline{FLT}$ = 0)	Before F.E., PHI1 T4	25		14		ns
t_{BERh}	4-11	$\overline{BER}$ Signal Hold ($\overline{FLT}$ = 0)	After R.E., PHI2 T4	0		0		ns
t_{RSTIs}	4-18	Reset Input Setup	Before F.E., PHI1 Ti	20		10		ns
t_{RSTIw}	4-18	Reset Input Width	At 0.8V (Both Edges)	64		64		t_{cp}

4.4.2.3 Clocking Requirements: NS32382-10, NS32382-15

Name	Figure	Description	Reference/ Conditions	NS32382-10		NS32382-15		Units
				Min	Max	Min	Max	
t_{Cp}	4-17	Clock Period	R.E., PHI1, PHI2 to Next R.E., PHI1, PHI2	100	250	66	250	ns
$t_{CLw(1,2)}$	4-17	PHI1, PHI2 Pulse Width	At 2.0V on PHI1, PHI2 (Both Edges)	0.5 t_{cp} −10 ns		0.5 t_{cp} −6 ns		
$t_{CLh(1,2)}$	4-17	PHI1, PHI2 High Time	At V_{CC} − 0.9V on PHI1, PHI2 (Both Edges)	0.5 t_{cp} −15 ns		0.5 t_{cp} −10 ns		
t_{CLl}	4-17	PHI1, PHI2 Low Time	At 0.8V on PHI1, PHI2 (Both Edges)	0.5 t_{Cp} −5 ns		0.5 t_{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	−2	5	ns
t_{nOVLas}		Non-Overlap Asymmetry ($t_{nOVL(1)} - t_{nOVL(2)}$)	At 0.8V on PHI1, PHI2	−4	4	−3	3	ns
t_{CLhas}		PHI1, PHI2 Asymmetry $t_{CLh(1)} - t_{CLh(2)}$)	At V_{CC} − 0.9V on PHI1, PHI2	−5	5	−3	3	ns

4.0 Device Specifications (Continued)

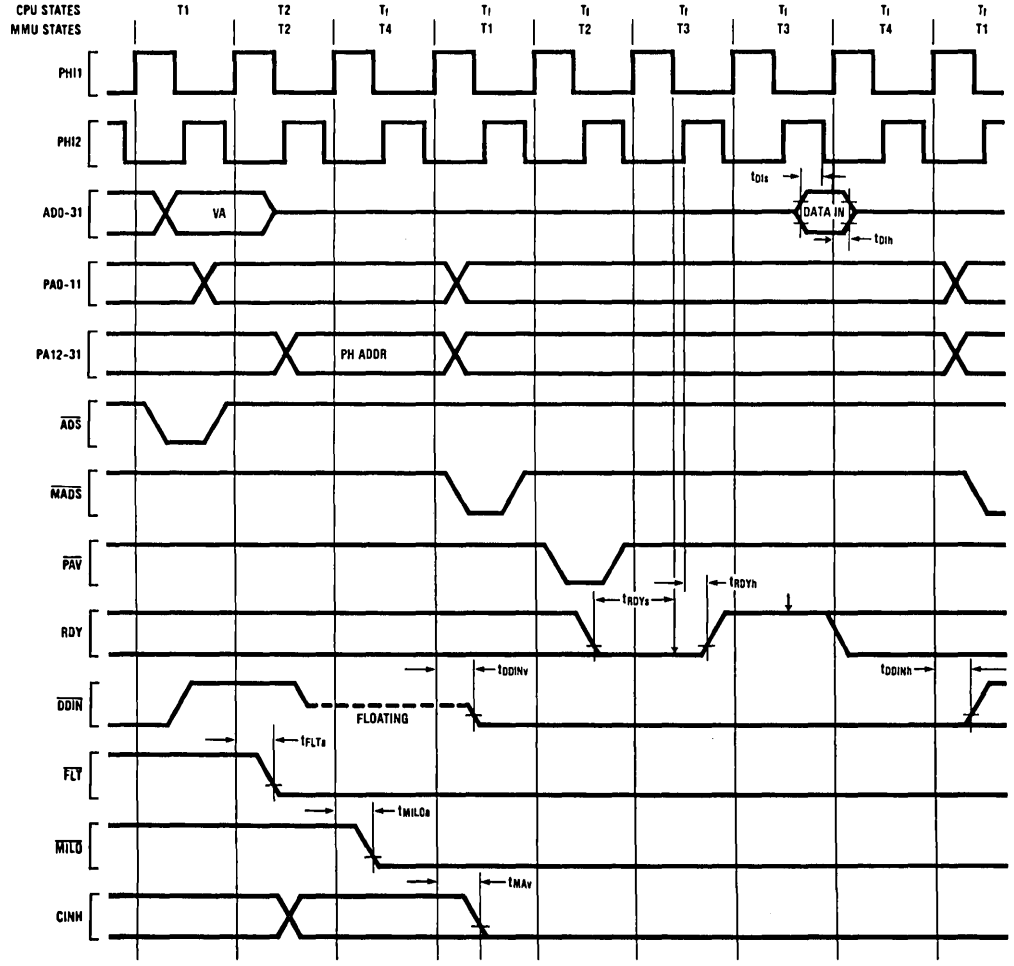
4.4.3 Timing Diagrams



TL/EE/9142-31

FIGURE 4-4. CPU Write Cycle Timing; Translation in TLB

4.0 Device Specifications (Continued)

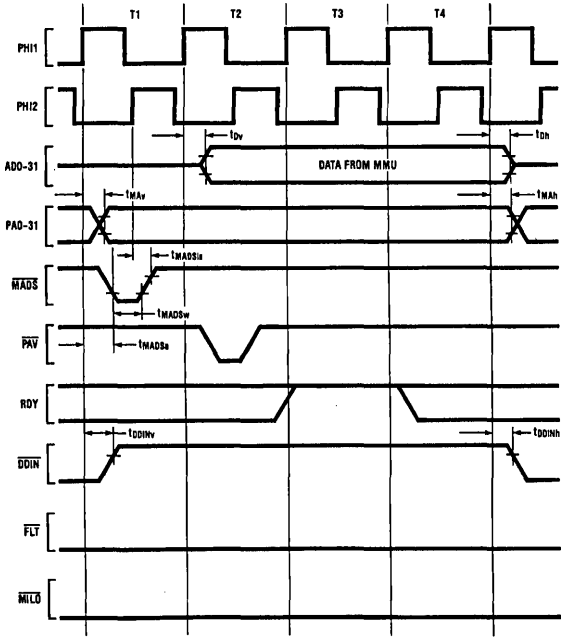


TL/EE/9142-32

FIGURE 4-5. MMU Read Cycle Timing (1-Wait State); After a TLB Miss

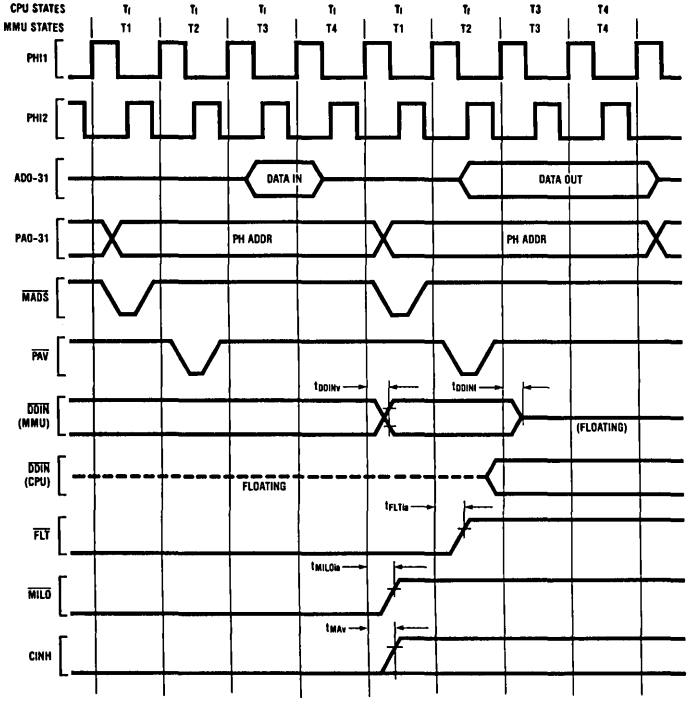
Note: After $\overline{FLT}$ is deasserted, $\overline{DDIN}$ may be driven temporarily by both CPU and MMU. This, however, does not cause any conflict. Since CPU and MMU force $\overline{DDIN}$ to the same logic level.

4.0 Device Specifications (Continued)



TL/EE/9142-33

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



TL/EE/9142-34

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

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

4.0 Device Specifications (Continued)

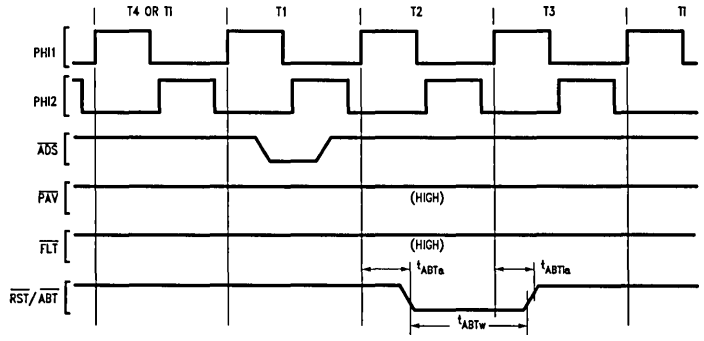


FIGURE 4-8. Abort Timing ($\overline{FLT}$ = 1)

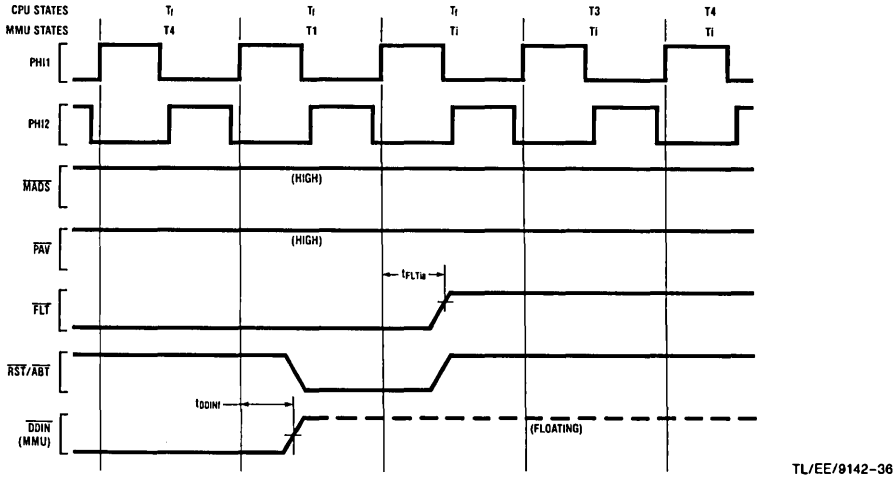
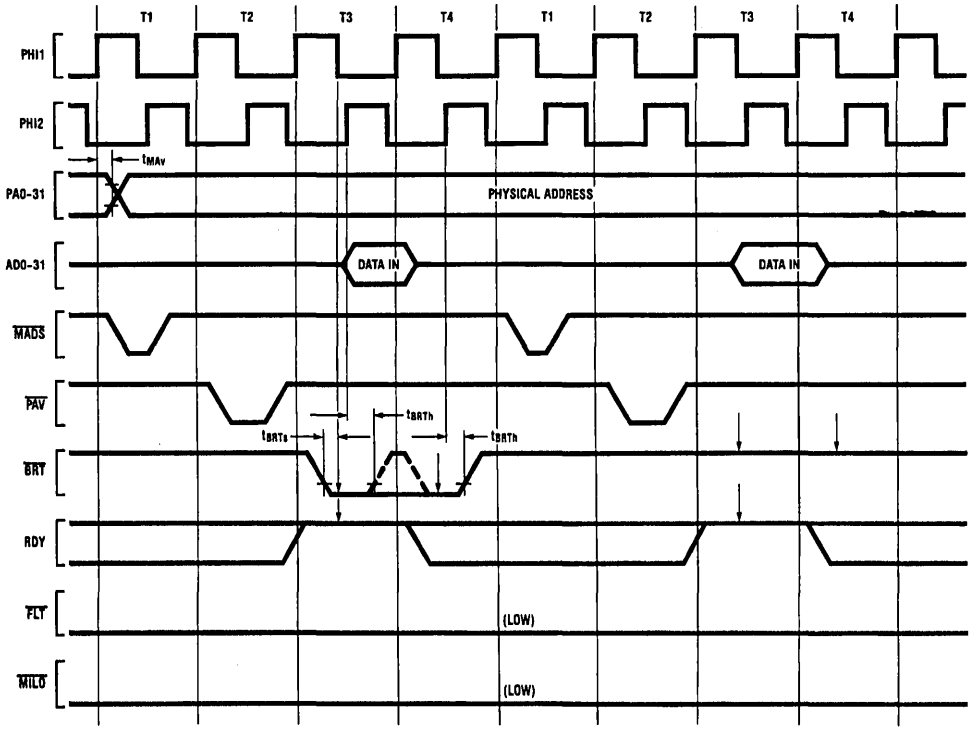


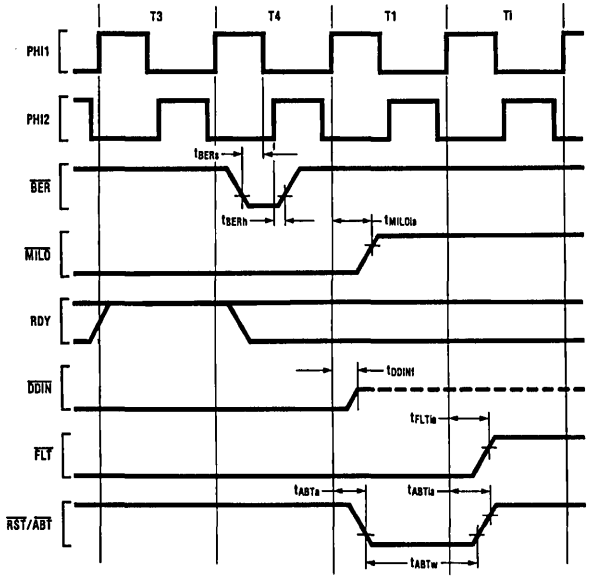
FIGURE 4-9. Abort Timing ($\overline{FLT}$ = 0)

4.0 Device Specifications (Continued)



TL/EE/9142-37

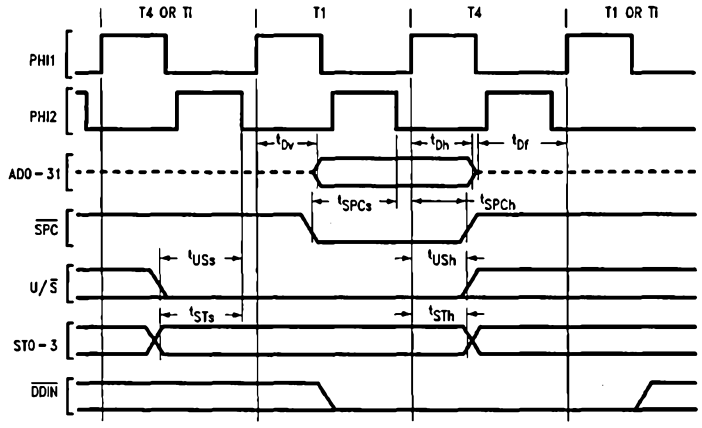
FIGURE 4-10. MMU Bus Retry Timing



TL/EE/9142-53

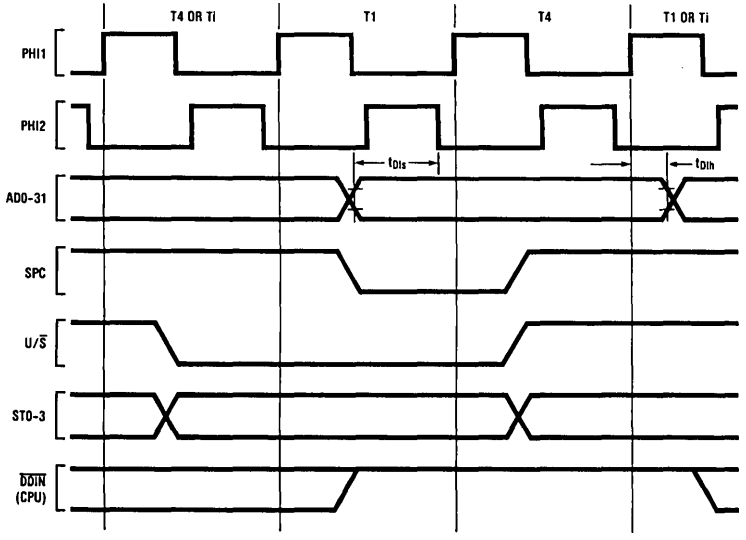
FIGURE 4-11. Bus Error Timing

4.0 Device Specifications (Continued)



TL/EE/9142-38

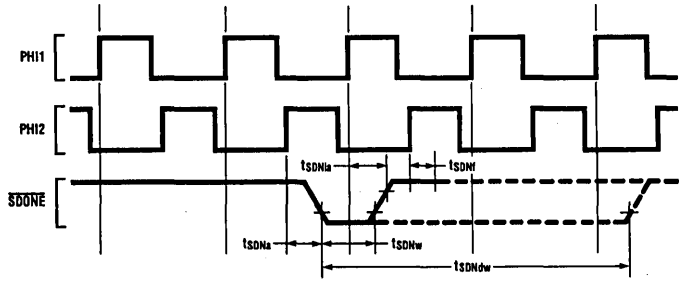
FIGURE 4-12. Slave Access Timing; CPU Reading from MMU



TL/EE/9142-39

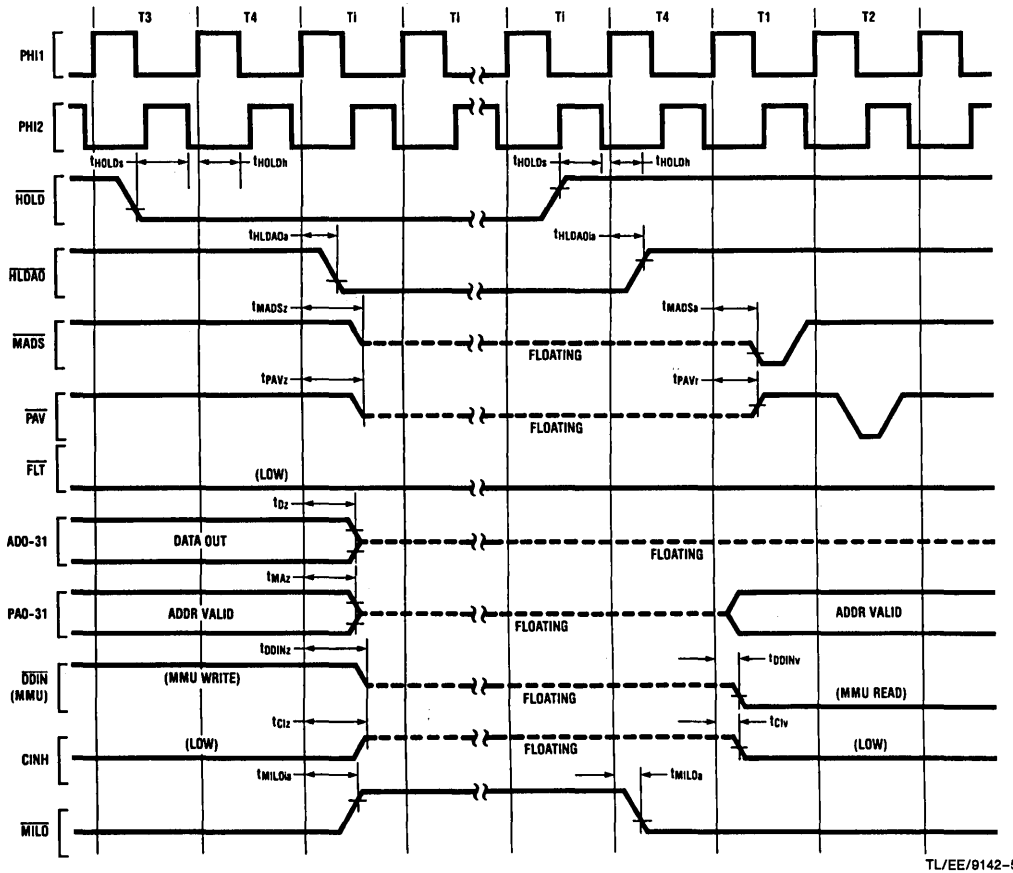
FIGURE 4-13. Slave Access Timing; CPU Writing to MMU

4.0 Device Specifications (Continued)



TL/EE/9142–40

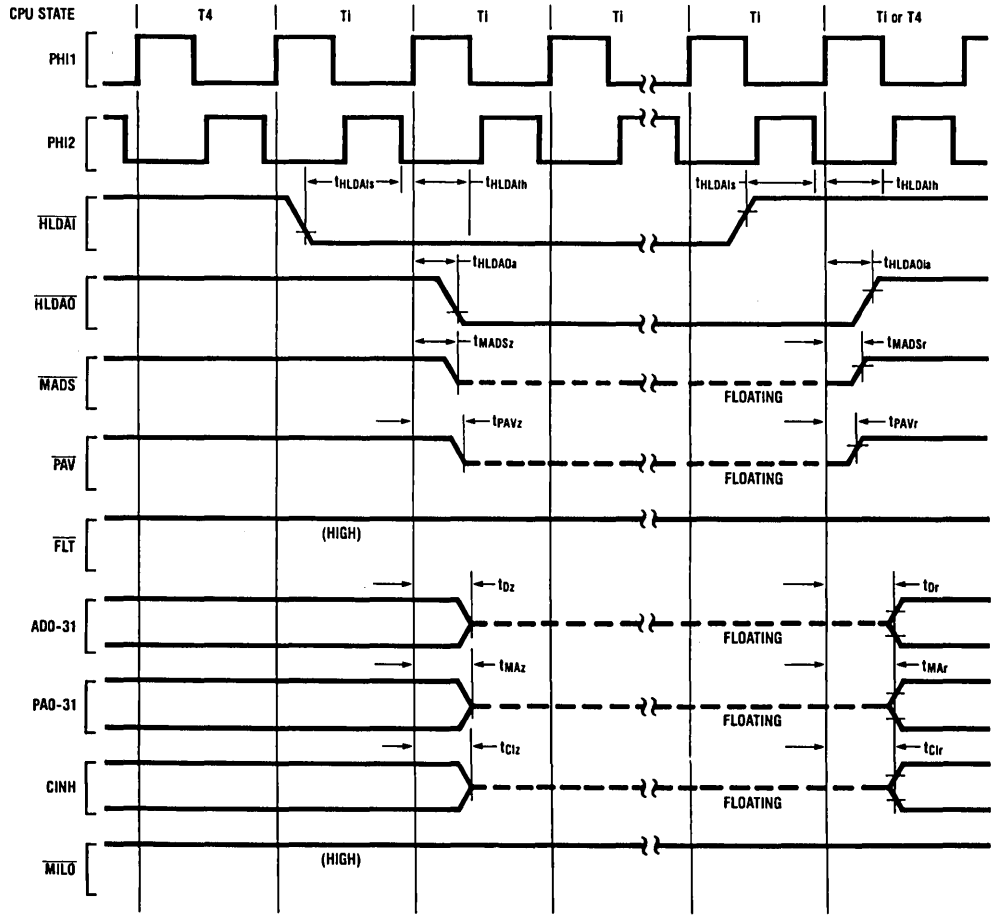
FIGURE 4-14. SDONE Timing



TL/EE/9142–50

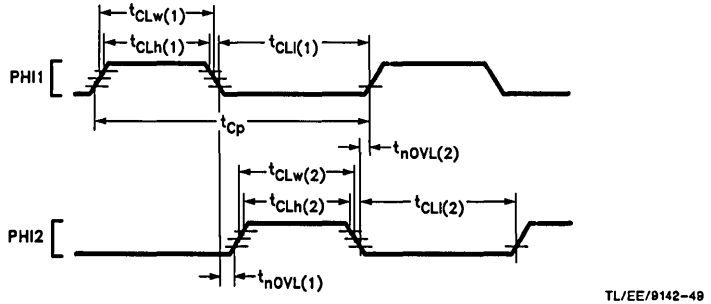
FIGURE 4-15. Hold Timing (FLT = 0)

4.0 Device Specifications (Continued)



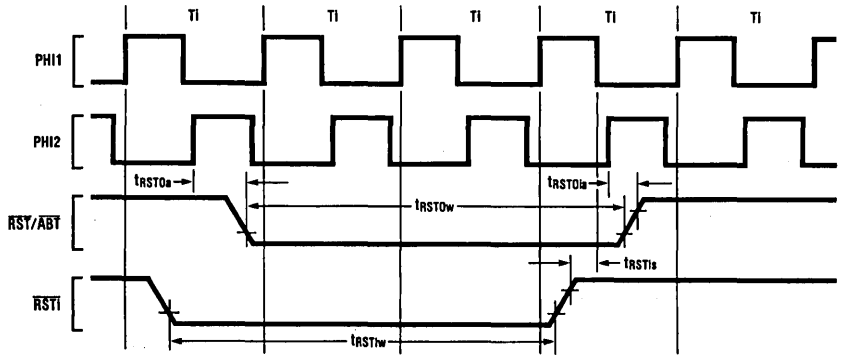
TL/EE/9142-51

FIGURE 4-16. Hold Timing ($\overline{FLT}$ = 1)



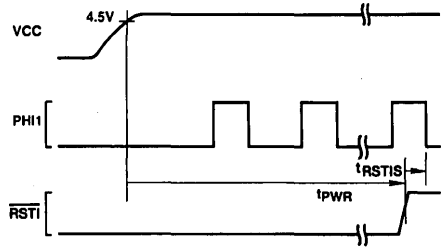
TL/EE/9142-49

FIGURE 4-17. Clock Waveforms



TL/EE/9142-45

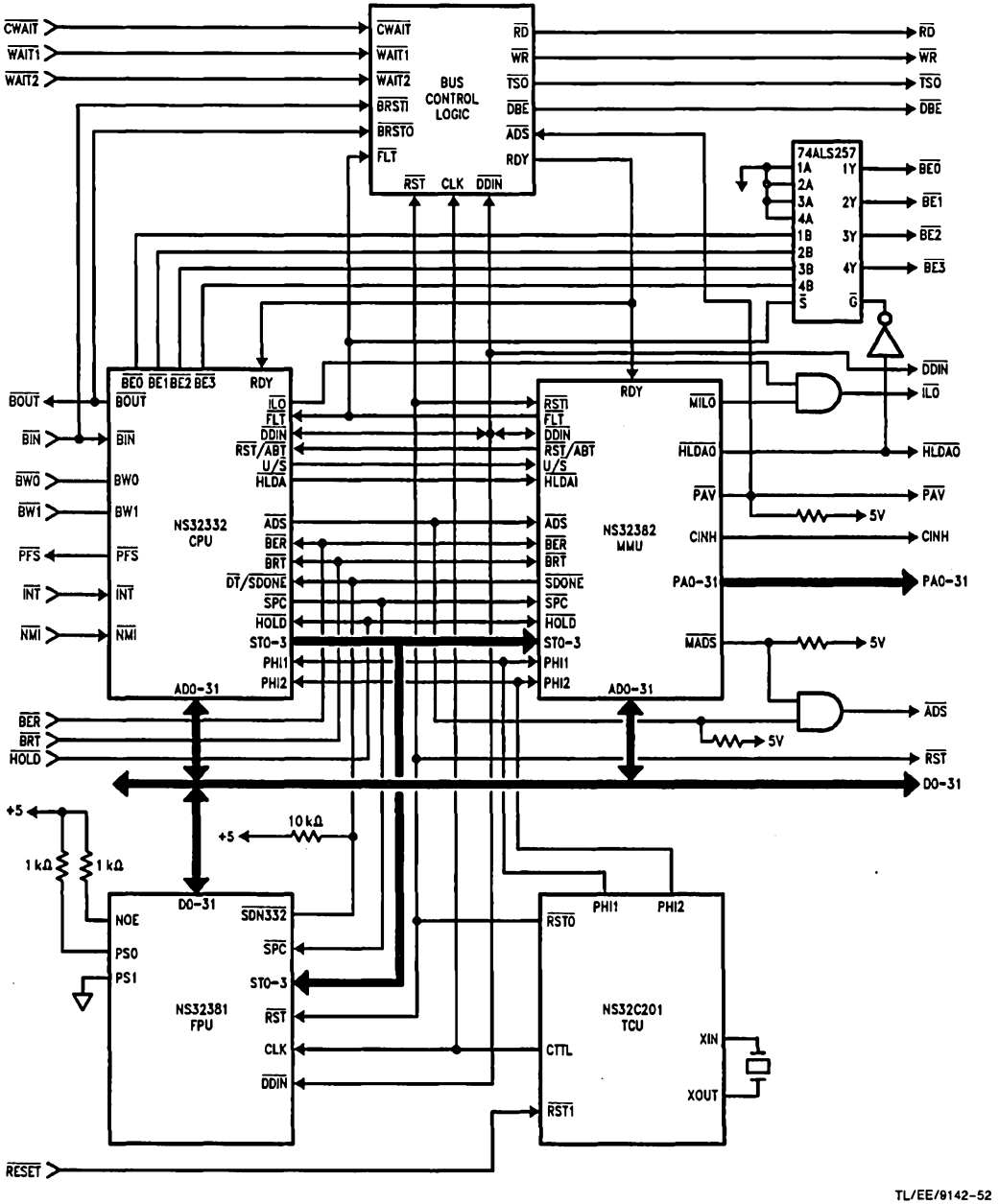
FIGURE 4-18. Non Power-On Reset Timing



TL/EE/9142-46

FIGURE 4-19. Power-On Reset

Appendix A: Interfacing Suggestions



TL/EE/9142-52

FIGURE A-1. System Connection Diagram