National Semiconductor

PRELIMINARY

NS32203-10 Direct Memory Access Controller

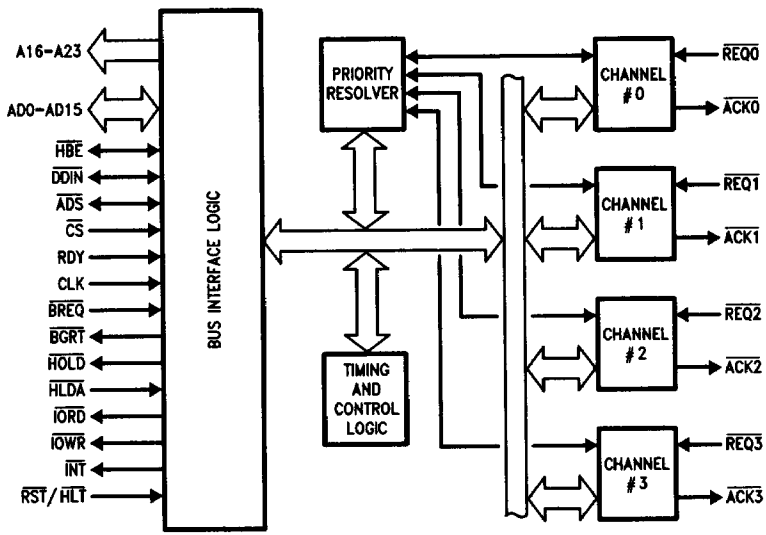
General Description

The NS32203 Direct Memory Access Controller (DMAC) is a support chip for the Series 32000® microprocessor family designed to relieve the CPU of data transfers between memory and I/O devices. The device is capable of packing data received from 8-bit peripherals into 16-bit words to reduce system bus loading. It can operate in local and remote configurations. In the local configuration it is connected to the multiplexed Series 32000 bus and shares with the CPU, the bus control signals from the NS32201 Timing Control Unit (TCU). In the remote configuration, the DMAC, in conjunction with its own TCU, communicates with I/O devices and/or memory through a dedicated bus, enabling rapid transfers between memory and I/O devices. The DMAC provides 4 16-bit I/O channels which may be configured as two complementary pairs to support chaining.

Features

- Direct or Indirect data transfers
- Memory to Memory, I/O to I/O or Memory to I/O transfers
- Remote or Local configurations
- 8-Bit or 16-Bit transfers
- Transfer rates up to 5 Megabytes per second
- Command Chaining on complementary channels
- Wide range of channel commands
- Search capability
- Interrupt Vector generation
- Simple interface with the Series 32000 Family of Microprocessors
- High Speed XMOS™ Technology
- Single +5V Supply
- 48-Pin Dual-In-Line Package

Block Diagram



TL/EE/8701–1

Table of Contents

List of Illustrations

1.0 Product Introduction

The NS32203 Direct Memory Access Controller (DMAC) is specifically designed to minimize the time required for high speed data transfers in a Series 32000-based computer system. It includes a wide variety of options and operating modes to enhance data throughput and system optimization, and to allow dynamic reconfiguration under program control.

The NS32203 can operate in two basic system configurations: local and remote. In the local configuration, the DMAC and the CPU share the same bus (address, data and control) and only one of them can perform data transfers on the bus at any one time. In this configuration, the DMAC and the CPU also share a Timing Control Unit (TCU) and a single set of address latches. Since this configuration yields a minimum part-count system, it offers a good cost/performance trade-off in many situations.

The remote configuration is intended to minimize the CPU bus use. In this configuration, the NS32203 I/O devices and optional buffer memory have their own dedicated bus (remote bus) so that an I/O transfer may be performed without loading the CPU bus (local bus).

Communication between the dedicated bus and the CPU bus may be initiated at any time by either the CPU or the NS32203. The DMAC accesses the CPU bus whenever a data transfer to/from memory or any I/O device residing on this bus is to be performed. The CPU, in turn, accesses the dedicated bus for reading status data or for programming either the DMAC or its I/O devices.

The NS32203 internal organization consists of seven functional blocks as illustrated in the block diagram. Descriptions of these blocks are given below.

DMA Channels. The NS32203 provides four channels. Each channel accepts a request from a peripheral I/O device and informs it when data transfer cycles are about to begin. A set of registers is provided for each channel to control the type of operation for that channel.

Bus Interface Unit. The bus interface unit controls all data transfers between peripheral I/O devices and memory whenever the DMAC is in control of the bus. This unit also controls the transfer of data between the CPU and the DMAC internal registers.

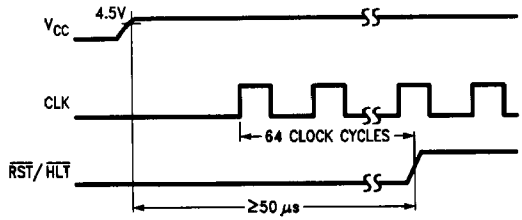
Timing and Control Logic. This block generates all the sequencing and control signals necessary for the operation of the DMAC.

Priority Resolver. This block resolves contentions among channels requesting service simultaneously.

2.0 Functional Description

2.1 RESETTING

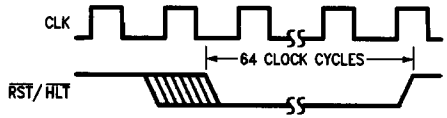
The $\overline{RST}/\overline{HLT}$ line serves both as a reset input for the on-chip logic and as a DMAC HALT input. Resetting is accomplished by pulling $\overline{RST}/\overline{HLT}$ low for at least 64 clock cycles. Upon detecting a Reset, the DMAC terminates any Data transfer in progress, resets its internal logic and enters an inactive state. On application of power, $\overline{RST}/\overline{HLT}$ must be held low for at least 50 μs after V_{CC} is stable. This is to ensure that all on-chip voltages are stable before operation. Whenever reset is applied, the rising edge must occur while the clock signal on the CLK pin is high (see *Figure 2-1* and *2-2*). The NS32201 TCU provides circuitry to meet the reset requirements. *Figure 2-3* shows the recommended connections. The HALT function is accomplished when $\overline{RST}/\overline{HLT}$ is activated for 1 or 2 clock cycles and then released. It can be used to stop any data transfer in progress in case of a bus error. As soon as HALT is acknowledged by the NS32203, the current transfer operation is terminated. See *Figure 4-18*.



TL/EE/8701–2

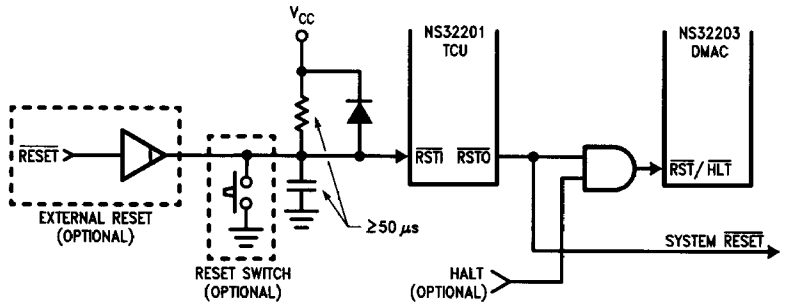
FIGURE 2-1. Power-On Reset Requirements

2.0 Functional Description (Continued)



TL/EE/8701-3

FIGURE 2-2. General Reset Timing



TL/EE/8701-4

FIGURE 2-3. Recommended Reset Connections

2.2 DATA TRANSFER OPERATIONS

After the NS32203 has been initialized by software, it is ready to transfer blocks of data, containing up to 64 kbytes, between memory and I/O devices, without further intervention required of the CPU. Upon receiving a transfer request from an I/O device, the DMAC performs the following operations:

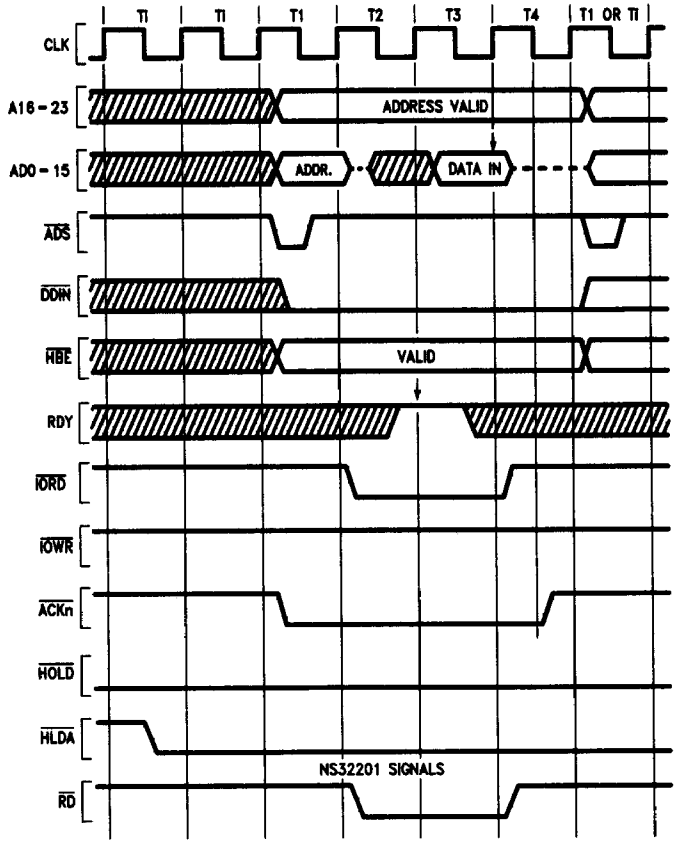
1) Acquires control of the bus
2) Acknowledge the requesting I/O device which is connected to the highest priority channel.
3) Starts executing data transfer cycles according to the values stored into the control registers of the channel being serviced.
4) Terminates data transfers and relinquishes control of the bus as soon as one of the programmed conditions is met.

Each channel can be programmed for indirect or direct data transfers. Detailed descriptions of these transfer types are provided in the following sub-sections.

2.2.1 Indirect Data Transfers

In this mode of operation, each byte or word transfer between source and destination requires at least two bus cycles. The data is first read into the DMAC and subsequently it is written into the destination. The bus cycles in this case are similar to the CPU bus cycles when the MMU is not used. This mode is slower than the direct mode, but is the only one that allows some data manipulation like Byte Search or Word Assembly/Disassembly. *Figure 2-4* and *2-5* show the read and write cycle timing diagrams related to indirect data transfers. If a search operation is specified, extra clock cycles may be added following each read cycle.

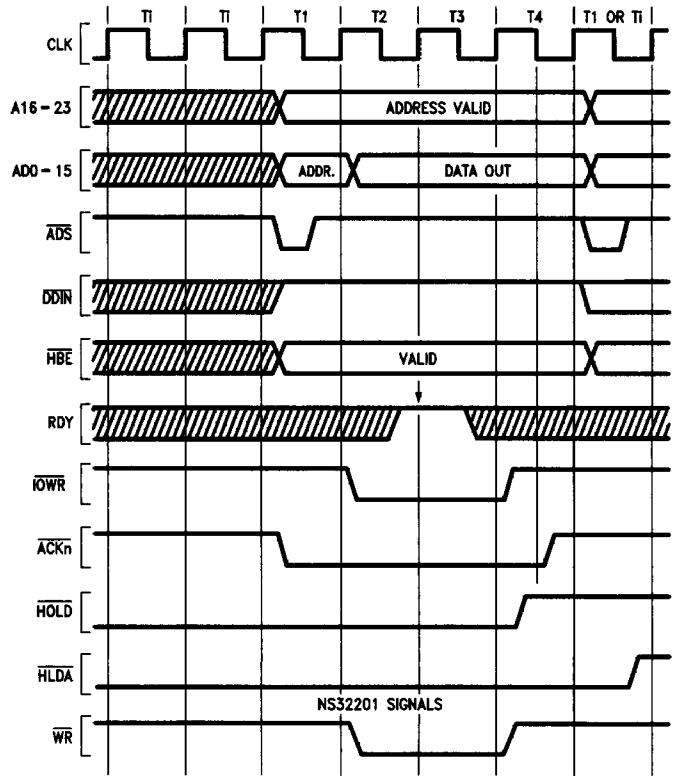
2.0 Functional Description (Continued)



TL/EE/8701–5

FIGURE 2-4. Indirect Read Cycle

2.0 Functional Description (Continued)



TL/EE/8701–6

FIGURE 2-5. Indirect Write Cycle (Single Transfer Mode)

Note: If burst mode is selected, $\overline{HOLD}$ is released at the end of the transfer operation.

2.0 Functional Description (Continued)

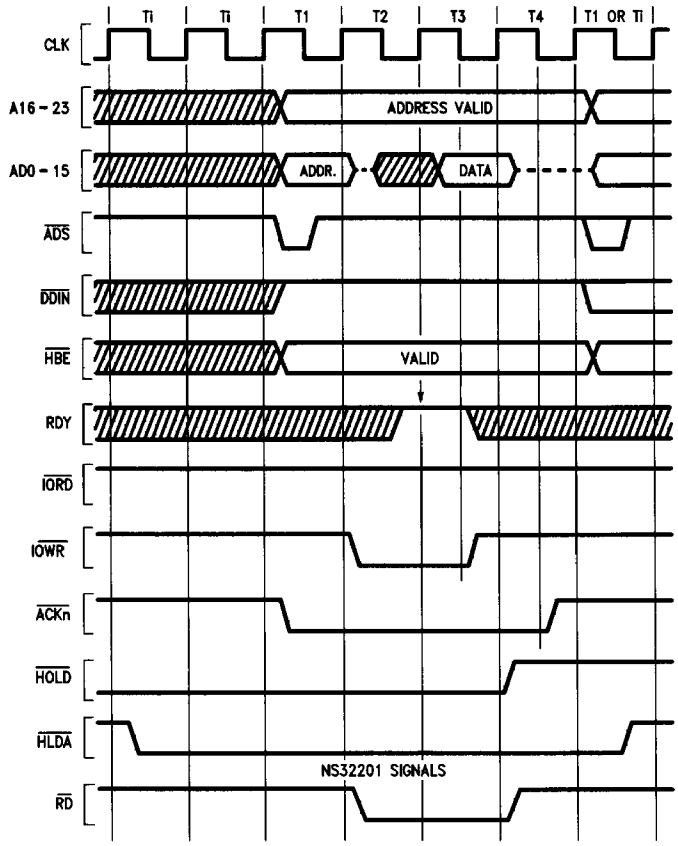
2.2.2 Direct (Flyby) Data Transfers

This mode of operation allows a very high data transfer rate between source and destination. Each data byte or word to be transferred requires only a single bus cycle instead of two separate read and write cycles, which are typical of the indirect mode. The DMAC accomplishes direct data transfers by activating $\overline{IORD}$, during memory write cycles, and $\overline{IOWR}$, during memory read cycles.

An I/O device, in the direct mode, is usually enabled by the proper acknowledge signal ($\overline{ACKn}$) from the DMAC. No search or word assembly/disassembly are possible during direct data transfers. *Figures 2-6* and *2-7* show the timing diagrams of direct memory-to-I/O and I/O-to-memory transfers respectively.

Note 1: In the direct mode each channel can control only one I/O device because the I/O device is hardwired to the $\overline{ACKn}$ output of the corresponding channel. in the indirect mode, a channel can control multiple devices as long as each device is selected through its own address rather than the $\overline{ACKn}$ output. However, the possiblity of selecting a single I/O device by the $\overline{ACKn}$ output is maintained in the indirect mode as well.

Note 2: Whenever the DMAC is either idle or is performing indirect transfers, it generates the $\overline{IORD}$ and $\overline{IOWR}$ signals as a replica of $\overline{RD}$ and $\overline{WR}$. This simplifies the logic required to access I/O devices wired for direct data transfers.



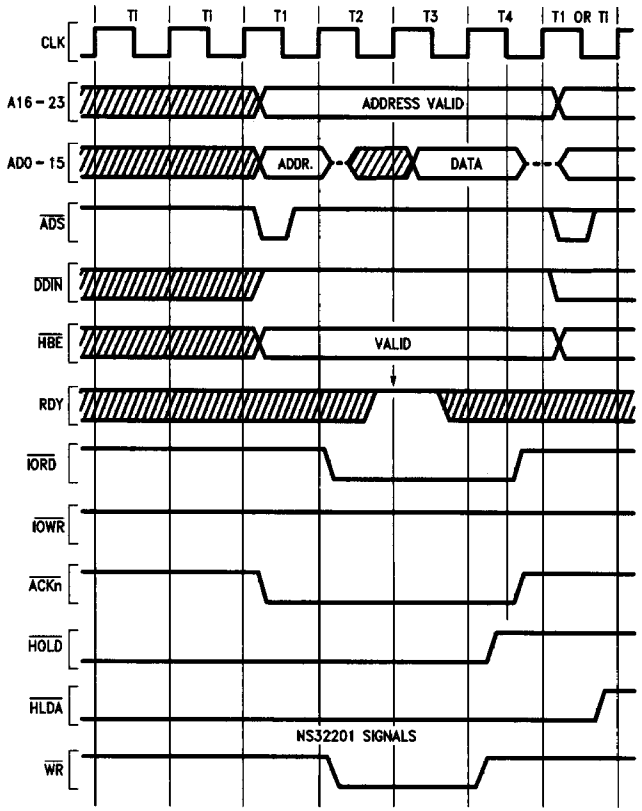
TL/EE/8701-7

FIGURE 2-6. Direct Memory-To-I/O Data Transfer (Single Transfer Mode)

2.0 Functional Description (Continued)

2.3 LOCAL CONFIGURATION

As previously mentioned, in the local configuration the DMAC shares with CPU and MMU the multiplexed address/data bus as well as the control signals from the NS32201 TCU. A typical local configuration is shown in *Figure 2-8*. The DMAC, in the local configuration, must gain control of the bus whenever a data transfer cycle is to be performed, even though it is directed to an I/O device and is related to an indirect data transfer. This causes the system to be quite sensitive to the volume of data handled by the DMAC. Thus, the overall system performance decreases as the volume of data increases. A possible solution to this problem is to use the remote configuration, described in the following section. A significant advantage of the local configuration is its simplicity.



TL/EE/8701-8

FIGURE 2-7. Direct I/O-To-Memory Data Transfer (Single Transfer Mode)

2.0 Functional Description (Continued)

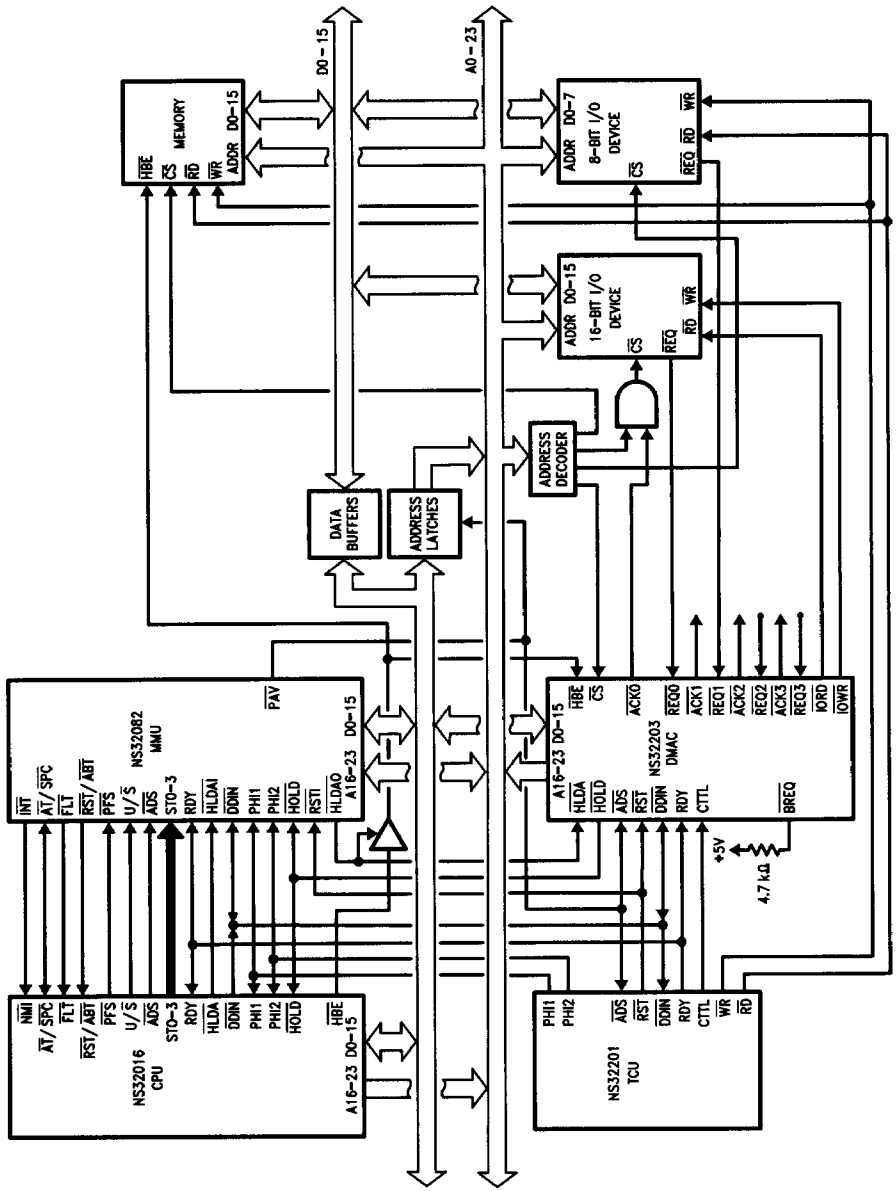


FIGURE 2-8. NS32203 Interconnections In Local Configuration

Note 1: The 16 Bit I/O device is wired for direct transfers.
Note 2: The data buffers should not be enabled during direct data transfers or CPU accesses to the DMAC registers.

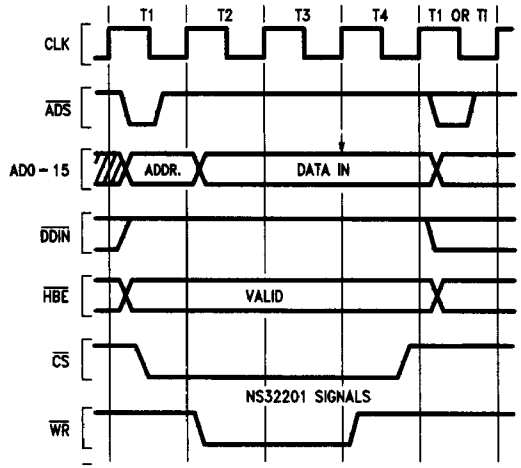
2.0 Functional Description (Continued)

2.4 REMOTE CONFIGURATION

The remote configuration is intended to minimize CPU Bus usage. In this configuration, the DMAC, buffer memory and I/O devices reside on a dedicated bus. Communication between the dedicated bus and the CPU bus is achieved by means of TRI-STATE buffers. Whenever the CPU needs to access the dedicated bus, it issues a bus request to the NS32203 by activating the $\overline{\text{BREQ}}$ signal. As the dedicated bus becomes idle, the DMAC pulls off the bus and acknowledges the CPU request by activating $\overline{\text{BGRT}}$. This output is also used as a control signal for the interconnection logic of the two buses.

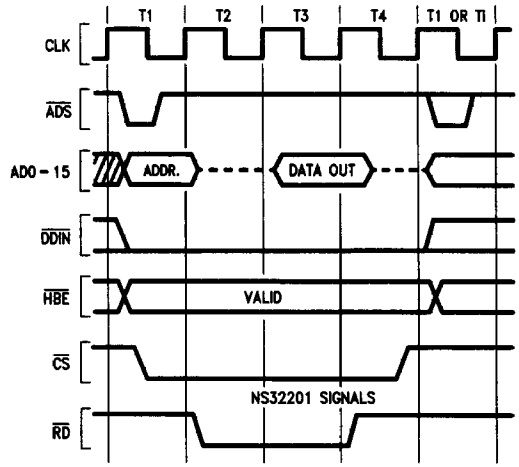
The CPU can either be interrupted by $\overline{\text{BGRT}}$ or it can poll $\overline{\text{BGRT}}$ to determine when the dedicated bus can be accessed. The DMAC, in turn, before accessing the CPU bus, has to gain control of it. This is accomplished through the usual request-acknowledge mechanism performed by means of the $\overline{\text{HOLD}}$ and $\overline{\text{HLDA}}$ signals.

Figure A-1 in Appendix A shows an interconnection diagram of a basic remote configuration. Both TCUs are clocked by the same clock signal. They are synchronized during reset by the $\overline{\text{RWEN}}$/SYNC signal so that their output clocks are in phase. *Figures 2-9* and *2-10* show the timing diagrams for read and write accesses to the NS32203 internal registers.



TL/EE/8701-10

FIGURE 2-9. Write to NS32203 Internal Registers



TL/EE/8701-11

FIGURE 2-10. Read from NS32203 Internal Registers

2.0 Functional Description (Continued)

2.5 DATA SOURCE (DESTINATION) ATTRIBUTES

Two types of data source (destination) are recognized: I/O device and memory. If the source (destination) is an I/O device, its address register is not changed after a data transfer; if it is memory, its address register is either incremented or decremented after any data transfer, according to the value of the corresponding direction bit. In the remote configuration, any data source (destination) may reside either on the CPU bus or on the dedicated bus. If it resides on the dedicated bus, the NS32203 does not activate the $\overline{HOLD}$ request line when an access to the source (destination) is performed, unless a direct transfer with a data destination (source) residing on the CPU bus is required.

Data can be transferred in either 8 bit or 16 bit units. The DMAC always considers the memory to be 16 bits wide. Thus, if an 8 bit transfer is specified, address bit A0 will determine the byte of the data-bus where the transfer takes place. If A0 = 0, the transfer occurs on the low order byte. If A0 = 1, it occurs on the high order byte. Different transfer widths can be specified for source and destination. However, some limitations exist in specifying these transfer widths when certain operations must be performed. These limitations are explained below.

1) If a transfer block has an odd number of bytes or is not word aligned, an 8 bit width for source and destination should be selected.
2) 16-bit I/O transfers can not be specified with 8 bit memory transfers.
3) Memory to memory transfers should have the same width.

Note 1: If source and destination are both memory, DMAC transfers can only be performed in indirect mode.

Note 2: If source and destination are both I/O devices and direct mode is being used, the source device is accessed by $\overline{IORD}$ and $\overline{ACK}$n; the destination device is accessed by $\overline{WR}$ (from the NS32201) and $\overline{CS}$ (from the address decoder). This allows a one direction data transfer only from one I/O device (source) to another. If data is to be transferred in both directions in direct mode between two I/O devices, two channels must be used (one for each direction of transfer), and extra hardware is required to control the read and write signals to the two I/O devices.

Note 3: When an 8-bit transfer is related to an I/O device, the other half of the 16-bit data bus is considered as DON'T CARE, and the $\overline{HBE}$/ signal may be activated.

2.6 WORD ASSEMBLY/DISASSEMBLY

This feature is automatically enabled when indirect transfers are selected, with data transferred between an 8-bit wide I/O device and a 16-bit I/O device or memory. For every 16-bit I/O device or memory access, the DMAC accesses the 8-bit I/O device twice, assembling two data bytes into a 16-bit word or breaking a 16-bit word into two data bytes, depending on the direction of transfer. The word assembly/disassembly feature allows a significant increase in the transfer speed and minimizes the CPU bus usage when the transfer occurs between an 8-bit I/O device residing on the dedicated bus, and a 16-bit I/O device or memory residing on the CPU bus. Word assembly/disassembly is not possible during direct data transfers.

Note: Requests from other channels are not acknowledged in the middle of a word assembly/disassembly. If this is unacceptable, 8 bit transfers should be specified for both source and destination.

2.7 AUTO TRANSFER

The NS32203 initiates a data transfer as a result of a request from an I/O device. In some cases a data transfer may be necessary without the corresponding request signal being asserted. This can happen, for example, when a block of data is to be moved from one memory region to another. In such cases, the auto transfer mode can be selected by setting an appropriate bit in the command register. The DMAC will initiate a data transfer regardless of the $\overline{REQ}$n signal for that channel.

Note: For proper operation, when auto transfer is required, the low order byte of the command register (containing the auto-transfer enable bit) should be written into after the other registers controlling the channel operation have been initialized.

2.8 SEARCH

The NS32203 provides a search capability that can be used to detect the occurrence of a certain data pattern. The search is performed by comparing each data byte with the search register, in conjunction with the mask register. An appropriate bit in the command register indicates whether the search continues 'UNTIL' a match occurs, or 'WHILE' a match exists. The search operation does not necessarily involve a data transfer. The DMAC allows a block of data to be searched without requiring any data transfer between source and destination. When performing a search, the user can specify whether or not the matched byte will be transferred. If 'INCLUSIVE SEARCH' is specified (INC = 1), the matched byte will be transferred, and the channel parameters will be updated accordingly. In this case, if a 16 bit word has been read from the data source and the search condition is satisfied by the low order byte, then the high order byte is transferred as well. If 'EXCLUSIVE SEARCH' is specified (INC = 0), the transfer will terminate with the last byte before the search condition was satisfied, and the parameters will point to the last transferred byte.

Search is not possible during direct transfers.

2.9 INTERRUPTS

The NS32203 provides interrupt circuitry that can be used to generate an interrupt whenever a data transfer is completed or a search condition is met. If an NS32202 ICU is used, the $\overline{INT}$ signal from the DMAC should be connected to an interrupt input of the ICU. When an interrupt occurs and the corresponding interrupt acknowledge (INTA) or return from interrupt (RETI) cycle is executed by the CPU, the NS32203 supplies its own vector as if it were a cascaded ICU. For such operation the virtual address of the interrupt vector register should be placed in the ICU cascade table, described in the NS32016 and NS32202 data sheets. See section 3.1.2.

2.10 TRANSFER MODES

When the NS32203 is in the inactive state and a channel requests service, the DMAC gains control of the bus and enters the active state. It is in this state that the data transfer takes place in one of the following modes:

SINGLE TRANSFER MODE

In single transfer mode, the NS32203 makes a single byte or word transfer for each $\overline{HOLD}$/$\overline{HLDA}$ handshake sequence.

In this case the request signal from the I/O device is edge sensitive, that is, a single transfer is performed each time a

2.0 Functional Description (Continued)

falling edge on $\overline{REQn}$ occurs. To perform multiple transfers, it is therefore necessary to temporarily deassert $\overline{REQn}$ after each transfer is initiated. If auto transfer mode is selected, the bus is released between two transfers for at least one clock cycle.

BURST (DEMAND) TRANSFER MODE

In burst transfer mode the DMAC will continue making data transfers until $\overline{REQn}$ goes inactive. Thus, the I/O device requesting service may suspend data transfer by bringing $\overline{REQn}$ inactive. Service may be resumed by asserting $\overline{REQn}$ again. If the auto transfer mode is selected, the DMAC will perform a single burst of data transfers until the end-transfer condition is reached.

Note 1: In either of the transfer modes described above, data transfers can only occur as long as the byte count is not zero or a search condition is not met. Whenever any of these conditions occur, the NS32203 terminates the current operation and releases the bus for at least one clock cycle.

Note 2: Whenever the DMAC releases $\overline{HOLD}$, it waits for $\overline{HLDA}$ to go inactive for at least one clock cycle before reasserting $\overline{HOLD}$ again to continue the transfer operation.

2.11 CHAINING

The NS32203 provides a chaining feature that allows the four DMAC channels to be regarded as two complementary pairs. Channels 0 and 1 form the first pair, while channels 2 and 3 form the second pair. Each pair is programmed independently by setting the corresponding bit in the configuration register. When two channels are complementary, only the even channel can perform transfer operations, while the odd one serves as temporary storage for the new control values and parameters loaded for the chaining operation. If an operation is being performed by the even channel of a pair and an end-condition is reached, the channel is not returned to the inactive state; rather, a new set of control values with or without parameters is loaded from the complementary channel and a new operation is started. During the reload operation the bus is released for at least two clock cycles. At the end of the second operation the channel returns to the inactive state, unless a new set of values has been loaded into the complementary channel by the CPU.

The chaining feature can be used to transfer blocks of data to/from non-contiguous memory segments. For example, the CPU can load channel 0 and 1 with control values and parameters for the first two blocks. After the operation for the first block is completed by channel 0, the control values and parameters stored in channel 1 are transferred to channel 0, during an update cycle, and a second operation is started. The CPU, being notified by an interrupt, can load channel 1 registers with control values and parameters for the third data block.

Note 1: Whenever a reload operation occurs, the register values of the complementary channel are affected. Thus, the CPU must always load a new set of values into the complementary channel if another chaining operation is required.

Note 2: When the chain option is selected, the CPU must be given the opportunity to acquire the bus for enough time between DMAC operations, in order for the complementary channel to be updated.

2.12 CHANNEL PRIORITIES

The NS32203 has four I/O channels, each of which can be connected to an I/O device. Since no dependency exists between the different I/O devices, a priority level is assigned to each I/O channel, and a priority resolver is provided to resolve multiple requests activated simultaneously.

The priority resolver checks the priorities on every cycle. If a channel is being serviced and a higher priority request is received, the channel operation is suspended and control passes to the higher priority channel, unless the lock bit for the lower priority channel is set. If the lock bit is set, that channel operation is continued until completion before control passes to the higher priority channel. The bus is always released for at least two clock cycles when control passes from one channel to another.

Two types of priority encodings are available as software selectable options.

The first is fixed priority which fixes the channels in priority order based on the decreasing values of their numbers. Channel 3 has the lowest priority, while channel 0 has the highest.

The second option is variable priority. The last channel that receives service becomes the lowest priority channel among all other channels with variable priority, while the channels which previously had lower priority will get their priorities increased. If variable priority is selected for all four channels, any I/O device requesting service is guaranteed to be acknowledged after no more than three higher priority services have occurred. This prevents any channel from monopolizing the system. Priority types can be intermixed for different channels.

As an example, let channels 0, 2 and 3 have variable priority and channel 1 fixed priority. Channel 2 receives service first, followed by channel 0. The priority levels among all channels will change as follows.

Priority	Initial Order	Next Order	Final Order
High 3	ch.0	ACK → ch.0	ch.3
2	ch.1	ch.1	ch.1 → fixed priority
1	ACK → ch.2	ch.3	ch.2
Low 0	ch.3	ch.2	ch.0

Whenever the PT bit (priority type) in the command register is changed, the priority levels of all the channels are reset to the initial order. If only one channel has variable priority, then no change in priority will occur from the initial order.

Note: If the lock bit is not set, three idle states are inserted between the write cycle of a previous burst indirect transfer and the next read cycle.

3.0 Architectural Description

The NS32203 has 128 8-bit registers that can be addressed either individually or in pairs, using the 7 least significant bits of the address bus and the high byte enable signal $\overline{HBE}$. Seventy-one of these registers are reserved, while the rest are accessible by the CPU for read/write operations. *Figure 3-1* shows the NS32203 internal registers together with their address offsets. Detailed descriptions of these registers are given in the following sections.

3.1 GLOBAL REGISTERS

The global registers consist of one configuration, one status and two interrupt vector registers. They are shared by all channels, and they control the overall operation of the NS32203.

3.1.1 CONF—Configuration Register

This register controls the hardware configuration of the NS32203 as well as the chaining feature.

4

3.0 Architectural Description (Continued)

The CONF register format is shown below:

7	6	5	4	3	2	1	0
XXXXX					C1	C0	CNF

CNF — Configuration Bit. Determines whether the NS32203 is in local or remote configuration.

CNF = 0 = > Local Configuration

CNF = 1 = > Remote Configuration

C0 — Chaining bit for channels 0 and 1. Determines whether or not channel 0 and 1 are complementary.

C0 = 0 = > Channels not complementary

C0 = 1 = > Channel 1 complementary to channel 0

C1 — Chaining bit for channels 2 and 3. Determines whether or not channels 2 and 3 are complementary.

C1 = 0 = > Channels not complementary

C1 = 1 = > Channel 3 complementary to channel 2

XXXXX — Reserved. These bits should be set to 0.

At reset, all CONF bits are reset to zero.

Note: The CNF bit should never be set by the software if the DMAC is wired for local configuration, otherwise bus conflicts will result.

	23 16	15 8	7 0	
Channel 0 Control Registers	COM(H) (02_{16})	COM(M) (01_{16})	COM(L) (00_{16})	Command
			SRCH (04_{16})	Search Pattern
			MSK (08_{16})	Search Mask
Channel 0 Parameter Registers	SRC(H) ($0E_{16}$)	SRC(M) ($0D_{16}$)	SRC(L) ($0C_{16}$)	Source Address
	DST(H) (12_{16})	DST(M) (11_{16})	DST(L) (10_{16})	Destination Address
		LNGT(H) (15_{16})	LNGT(L) (14_{16})	Block Length
Channel 1 Control Registers	COM(H) (22_{16})	COM(M) (21_{16})	COM(L) (20_{16})	Command
			SRCH (24_{16})	Search Pattern
			MSK (28_{16})	Search Mask
Channel 1 Parameter Registers	SRC(H) ($2E_{16}$)	SRC(M) ($2D_{16}$)	SRC(L) ($2C_{16}$)	Source Address
	DST(H) (32_{16})	DST(M) (31_{16})	DST(L) (30_{16})	Destination Address
		LNGT(H) (35_{16})	LNGT(L) (34_{16})	Block Length
Channel 2 Control Registers	COM(H) (42_{16})	COM(M) (41_{16})	COM(L) (40_{16})	Command
			SRCH (44_{16})	Search Pattern
			MSK (48_{16})	Search Mask
Channel 2 Parameters Registers	SRC(H) ($4E_{16}$)	SRC(M) ($4D_{16}$)	SRC(L) ($4C_{16}$)	Source Address
	DST(H) (52_{16})	DSC(M) 51_{16})	DST(L) (50_{16})	Destination Address
		LNGT(H) (55_{16})	LNGT(L) (54_{16})	Block Length
Channel 3 Control Registers	COM(H) (62_{16})	COM(M) (61_{16})	COM(L) (60_{16})	Command
			SRCH (64_{16})	Search Pattern
			MSK (68_{16})	Search Mask
Channel 3 Parameter Registers	SRC(H) ($6E_{16}$)	SRC(M) ($6D_{16}$)	SRC(L) ($6C_{16}$)	Source Address
	DST(H) (72_{16})	DST(M) (71_{16})	DST(L) (70_{16})	Destination Address
		LNGT(H) (75_{16})	LNGT(L) (74_{16})	Block Length
Global Registers			CONF (78_{16})	Configuration
			SVCT ($5C_{16}$)	Software Vector
			HVCT ($7C_{16}$)	Hardware Vector
		STAT(H) ($7F_{16}$)	STAT(L) ($7E_{16}$)	Status

FIGURE 3-1. NS32203 Internal Registers

3.0 Architectural Description (Continued)

3.1.2 HVCT — Hardware Vector Register

This register contains the interrupt vector byte that is supplied to the CPU during an interrupt acknowledge (INTA) or return from interrupt (RETI) cycle. The HVCT register format is shown below.

7	6	5	4	3	2	1	0
BIAS					E	CN	

CN — Channel number. Represents the number of the interrupting channel

E — Error code. Determines whether a normal operation completion or an error condition has occurred on the interrupting channel.

E = 0 => Normal Operation Completion

E = 1 => A second interrupt was generated by the same channel before the first interrupt was serviced.

BIAS — Programmable bias. This field is programmed by writing the pattern BBBBB000 into the HVCT register.

The NS32203 always interprets a read of the HVCT register as either an interrupt acknowledge (INTA) cycle or a return from interrupt (RETI) cycle. Since these cycles cause internal changes to the DMAC, normal programs should never read the HVCT register (see next section). The DMAC distinguishes an INTA cycle from a RETI cycle by the state of an internal flip-flop, called Interrupt Service Flip-Flop, that toggles every time the HVCT register is read. This flip-flop is cleared on reset or when the HVCT register is written into. When an interrupt is acknowledged by the CPU, the $\overline{INT}$ signal is deasserted unless another interrupt from a lower priority channel is pending. In this case the $\overline{INT}$ signal is deasserted when the acknowledge cycle for the second interrupt is performed.

For this reason, if the $\overline{INT}$ signal is connected to an interrupt input of the NS32202 ICU, the triggering mode of that interrupt position should be 'low level'.

Furthermore, if that ICU interrupt input is programmed for cascaded operation and nesting of interrupts from other devices connected to the ICU is to be allowed, then the ICU interrupt input connected to the DMAC should be masked off during the interrupt service routine, before the CPU interrupt is reenabled. This is because the DMAC does not provide interrupt nesting capability.

An interrupt from a certain channel can be acknowledged only after the return from interrupt from a previously acknowledged interrupt is performed.

3.1.3 SVCT — Software Vector Register

The SVCT register is an image of the HVCT register. It is a read-only register used for diagnostics. It allows the programmer to read the interrupt vector without affecting the interrupt logic of the NS32203. The format of the SVCT register is the same as that of the HVCT register.

3.1.4 STAT — Status Register

The status register contains status information of the NS32203, and can be used when the interrupts are not enabled. Each set bit is automatically cleared when a read operation is performed. The format of this register is shown in the following figure.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ME	CH	MN	TC	ME	CH	MN	TC	ME	CH	MN	TC	ME	CH	MN	TC
channel #3				channel #2				channel #1				channel #0			

The status of each channel is defined in a four-bit field as described below:

TC — Transfer Complete.

Indicates the completion of a channel operation, regardless of the state of the length register or whether a match/no match condition occurred.

MN — Match/No Match Bit.

This bit is set when a match/no match condition occurs.

CH — Channel Halted.

Set when a channel operation is halted by pulling the $\overline{RST}/\overline{HLT}$ pin.

ME — Multiple events. This bit is set when more than one of the above conditions have occurred.

Note: If an interrupt is enabled, the corresponding bit in the status register is not cleared upon read, unless the interrupt is acknowledged.

3.2 CONTROL REGISTERS

Each of the four channels has three control registers, consisting of a 24-bit command register, an 8-bit search register and an 8-bit mask register.

3.2.1 COM — Command Register

The command register controls the operation of the associated channel. It is divided into three separately addressable parts: COM(L), COM(M) and COM(H). The format of each part and bit functions are shown below.

COM(L) — Command Register (Low-Byte)

7	6	5	4	3	2	1	0
AT	LK	PT	UW	INC	DI	CC	

CC — Command Code

CC = 00 => Channel Disabled.

CC = 01 => Search

CC = 10 => Data Transfer

CC = 11 => Data Transfer and Search

DI — Direct/Indirect Transfers

DI = 0 => Indirect Transfers

DI = 1 => Direct Transfers

INC — Inclusive/Exclusive Search

INC = 0 => Exclusive Search

INC = 1 => Inclusive Search

UW — Search type

UW = 0 => Search UNTIL

UW = 1 => Search WHILE

PT — Priority type

PT = 0 => Fixed

PT = 1 => Variable

LK — Priority lock

LK = 0 => Priority Unlocked

LK = 1 => Priority Locked

3.0 Architectural Description (Continued)

AT — Auto transfer
AT = 0 = > Auto Transfer Disabled
AT = 1 = > Auto Transfer Enabled

At Reset, the CC bits in COM(L) are cleared, disabling the channel.

Note: The CC bits can be cleared by software during an indirect data transfer to stop the transfer. This, however, should not be done during direct data transfers. See section 3.3.3.

COM(M) - Command Register (Middle-Byte)

7	6	5	4	3	2	1	0
DD	DW	DL	DT	SD	SW	SL	ST

ST — Source Type
ST = 0 = > I/O Device
ST = 1 = > Memory

SL — Source Location
(Effective only in the remote configuration)
SL = 0 = > Local
SL = 1 = > Remote

SW — Source Width
SW = 0 = > 8 Bits
SW = 1 = > 16 Bits

SD — Source Direction
SD = 0 = > Up
SD = 1 = > Down

DT — Destination Type
DT = 0 = > I/O Device
SD = 1 = > Memory

DL — Destination Location
(Effective only in the remote configuration)
DL = 0 = > Local
DL = 1 = > Remote

DW — Destination Width
DW = 0 = > 8 Bits
DW = 1 = > 16 Bits

DD — Destination Direction.
DD = 0 = > Up
DD = 1 = > Down

COM(H) - Command Register (High-Byte)

7	6	5	4	3	2	1	0
HLI	MNI	TCI	AMN		ATC	DM	X

X — Reserved. (Should be set to 0)

TM — Transfer Mode
DM = 0 = > Single Transfer
DM = 1 = > Burst Transfer

ATC — Action after Transfer Complete
ATC = 0 = > Disable Channel
ATC = 1 = > Load Control Values and Parameters from Complementary Channel and Continue

AMN — Action after Match/No Match
AMN = 00 = > Disable Channel
AMN = 01 = > Continue
AMN = 10 = > Load Control Values from Complementary Channel and Continue
AMN = 11 = > Load Control Values and Parameters from Complementary Channel and Continue

TCI — Interrupt Mask on "Transfer Complete"
TCI = 0 = > No Interrupt
TCI = 1 = > Interrupt

MNI — Interrupt Mask on "Match/No Match"
MNI = 0 = > No Interrupt
MNI = 1 = > Interrupt

HLI — Interrupt Mask on "Channel Halted"
HLI = 0 = > No Interrupt
HLI = 1 = > Interrupt

3.2.2 SRCH — Search Register

This 8-bit register holds the value to be compared with the data transferred during the channel operation.

3.2.3 MSK — Mask Register

The 8-bit mask register determines which bits of the transferred data are compared with corresponding search register bits. If a mask register bit is set to 0, the corresponding search register bit is ignored in the compare operation. At reset, all the MSK bits are set to 0.

3.3 PARAMETER REGISTERS

Each channel has three parameter registers, consisting of a 24-bit source address register, a 24-bit destination address register and a 16-bit block length register.

3.3.1 SRC — Source Address Register

The source address register points to the physical address of the data source. When the data source is an I/O device, the register does not change during the transfer operation. When the data source is memory, the register is incremented or decremented by either one or two after each transfer.

3.3.2 DST — Destination Address Register

The destination address register points to the physical address of the data destination. When the data destination is an I/O device, the register does not change during the transfer operation. When the data destination is memory, the register is incremented or decremented by either one or two after each transfer.

3.3.3 LNGT — Block Length Register

The block length register holds the number of bytes in the block to be transferred. It is decremented by either one or two after each transfer.

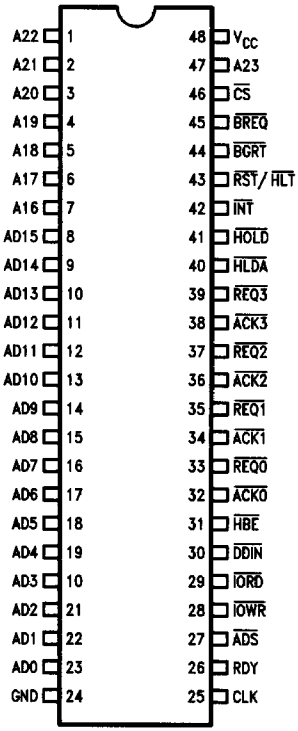
Note: A direct data transfer can be stopped by writing zeroes into the LNGT register. The number of bytes transferred can be determined in this case, from the value of either the SRC or the DST register.

4.0 Device Specifications

4.1. NS32203 PIN DESCRIPTIONS

The following is a brief description of all NS32203 pins. The descriptions reference portions of the Functional Description, Section 2.0.

Connection Diagram



TL/EE/8701-12

Top View

FIGURE 4-1. NS32203 Dual-In-Line Package

Order Number NS32203D or NS32203N
See NS Package Number D48A or N48A

4.1.1 SUPPLIES

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

Ground (GND): Ground reference for on-chip logic.

4.1.2 INPUT SIGNALS

Reset/Halt ($\overline{RST}/\overline{HLT}$): Active low. If held active for 1 or 2 clock cycles and released, this signal halts the DMAC operation on the active channel. If held longer, it resets the DMAC. Section 2.1.

Chip Select ($\overline{CS}$): When low, the device is selected, enabling CPU access to the DMAC internal registers.

Ready (RDY): Active high. When inactive, the DMA Controller extends the current bus cycle for synchronization with slow memory or peripherals. Upon detecting RDY active, the DMAC terminates the bus cycle.

Channel Request 0–3 ($\overline{REQ0}$ - $\overline{REQ3}$): Active low. These lines are used by peripheral devices to request DMAC service.

Bus Request ($\overline{BREQ}$): Used only in the remote configuration. This signal, when asserted, forces the DMAC to stop transferring data and to release the bus. It must be activated by the CPU before any CPU access to the remote bus is performed. In the local configuration this signal should be connected to V_{CC} via a 4.7k resistor. Section 2.4.

Hold Acknowledge ($\overline{HLDA}$): Active low. When asserted, indicates that control of the system bus has been relinquished by the current bus master and the DMAC can take control of the bus.

Clock (CLK): Clock signal supplied by the CTTL output of the NS32201 TCU.

4.1.3 OUTPUT SIGNALS

Address Bits 16–23 (A16–A23): Most significant 8 bits of the address bus.

Hold Request ($\overline{HOLD}$): Active low. Used by the DMAC to request control of the system bus.

Channel Acknowledge 0–3 ($\overline{ACK0}$ - $\overline{ACK3}$): These lines indicate that a channel is active. When a channel's request is honored, the corresponding acknowledge line is activated to notify the peripheral device that it has been selected for a transfer cycle. Section 2.2.2.

Bus Grant ($\overline{BGRT}$): Used only in the remote configuration. This signal is used by the DMAC to inform the CPU that the remote bus has been relinquished by the DMAC and can be accessed by the CPU. Section 2.4.

I/O Read ($\overline{IORD}$): Active low. Enables data to be read from a peripheral device. Section 2.2.2.

I/O Write ($\overline{IOWR}$): Active low. Enables data to be written to a peripheral device. Section 2.2.2.

Interrupt ($\overline{INT}$): Active low. Used to generate an interrupt request when a programmed condition has occurred. Section 2.9.

4.1.4 INPUT/OUTPUT SIGNALS

Address/Data 0–15 (AD0–AD15): Multiplexed Address/Data bus lines. Also used by the CPU to access the DMAC internal registers.

High Byte Enable ($\overline{HBE}$): Active low. Enables data transfers on the most significant byte of the data bus.

Address Strobe ($\overline{ADS}$): Active low. Controls address latches and indicates the start of a bus cycle.

Data Direction In ($\overline{DDIN}$): Active low. Status signal indicating the direction of data flow in the current bus cycle.

4.0 Device Specifications (Continued)

4.2 ABSOLUTE MAXIMUM RATINGS

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

Temperature Under Bias	0°C to +70°C
Storage Temperature	−65°C to +150°C
All Input or Output Voltages with Respect to GND	−0.5V to +7V
Power Dissipation	1.1 Watt

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 $+70°C$, $V_{CC} = 5V \pm 5\%$, 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_{OH}	High Level Output Voltage	$I_{OH} = -400\ \mu A$	2.4			V
V_{OL}	Low Level Output Voltage	$I_{OL} = 2$ mA			0.45	V
I_I	Input Load Current	$0 < V_{IN} \leq V_{CC}$	−20		20	μA
I_L	Leakage Current Output and I/O Pins in TRI-STATE/Input Mode	$0.4 \leq V_{IN} \leq V_{CC}$	−20		20	μA
I_{CC}	Active Supply Current	$I_{OUT} = 0$, $T_A = 25°C$		180	300	mA

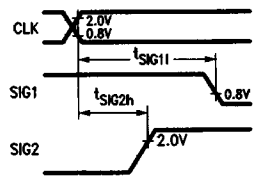
4.4 SWITCHING CHARACTERISTICS

4.4.1 Definitions

All the timing specifications given in this section refer to 0.8V and 2.0V on all the input and output signals as illustrated in *Figures 4-2* and *4-3*, unless specifically stated otherwise.

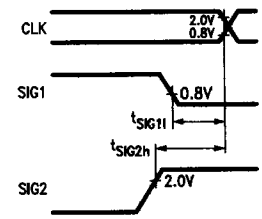
ABBREVIATIONS:

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



TL/EE/8701–13

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



TL/EE/8701–14

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

4.0 Device Specifications (Continued)

4.4.2 Timing Tables

4.4.2.1 Output Signals: Internal Propagation Delays, NS32203-10

Maximum Times Assume Capacitive Loading of 100 pF.

Name	Figure	Description	Reference/ Conditions	NS32203-10		Units
				Min	Max	
t_{ALv}	4-7	Address Bits 0–15 Valid	After R.E., CLK T1		50	ns
t_{ALh}	4-9	Address Bits 0–15 Hold Time	After R.E., CLK T2	5		ns
t_{AHv}	4-7	Address Bits 16–23 Valid	After R.E., CLK T1		50	ns
t_{AHh}	4-7	Address Bits 16–23 Hold	After R.E., CLK T1 or Ti	5		ns
t_{ALADSs}	4-8	Address Bits 0–15 Set Up	Before $\overline{ADS}$ T.E.	25		ns
t_{AHADSs}	4-8	Address Bits 16–23 Set Up	Before $\overline{ADS}$ T.E.	25		ns
t_{ALADSh}	4-9	Address Bits 0–15 Hold Time	After $\overline{ADS}$ T.E.	15		μs
t_{ALf}	4-8	Address Bits 0–15 Floating	After R.E., CLK T2		25	ns
t_{Dv}	4-7	Data Valid (Write Cycle)	After R.E., CLK T2		50	ns
t_{Dh}	4-7	Data Hold (Write Cycle)	After R.E., CLK T1 or Ti	0		ns
t_{DOv}	4-5	Data Valid (Reading DMAC Registers)	After R.E., CLK T3		50	
t_{DOh}	4-5	Data Hold (Reading DMAC Registers)	After R.E., CLK T4	10		
t_{HBEv}	4-7	$\overline{HBE}$ Signal Valid	After R.E., CLK T1		50	ns
t_{HBEh}	4-7	$\overline{HBE}$ Signal Hold	After R.E., CLK T1 or Ti	0		ns
t_{DDINv}	4-8	$\overline{DDIN}$ Signal Valid	After R.E., CLK T1		65	ns
t_{DDINh}	4-8	$\overline{DDIN}$ Signal Hold	After R.E., CLK T1 or Ti	0		ns
t_{ADSa}	4-7	$\overline{ADS}$ Signal Active	After R.E., CLK T1		35	ns
t_{ADSia}	4-7	$\overline{ADS}$ Signal Inactive	After R.E., CLK T1		40	ns
t_{ADSw}	4-7	$\overline{ADS}$ Pulse Width	at 0.8V (Both Edges)	30		ns
t_{ALz}	4-12. 4-13	AD0–AD15 Floating	After R.E., CLK Ti		55	ns
t_{AHz}	4-12, 4-13	A16–A23 Floating	After R.E., CLK Ti		55	ns
t_{ADSz}	4-12, 4-13	$\overline{ADS}$ Floating	After R.E., CLK Ti		55	ns
t_{HBEz}	4-12, 4-13	$\overline{HBE}$ Floating	After R.E., CLK Ti		55	ns
t_{DDINz}	4-12, 4-13	$\overline{DDIN}$ Floating	After R.E., CLK Ti		55	ns
t_{HLDa}	4-11	$\overline{HOLD}$ Signal Active	After R.E., CLK Ti		50	ns
t_{HLDia}	4-12	$\overline{HOLD}$ Signal Inactive	After R.E., CLK Ti or T4		50	ns
t_{INTa}	4-19, 4-21	$\overline{INT}$ Signal Active	After R.E., CLK Ti		40	ns
t_{ACKa}	4-16, 4-17, 4-7	$\overline{ACKn}$ Signal Active	After R.E., CLK T1		50	ns
t_{ACKia}	4-16, 4-17, 4-7	$\overline{ACKn}$ Signal Inactive	After F.E., CLK T4		35	ns

4.0 Device Specifications (Continued)

Name	Figure	Description	Reference/ Conditions	NS32203-10		Units
				Min	Max	
t_{BGRTa}	4-13	$\overline{BGRT}$ Signal Active	After R.E., CLK		65	ns
t_{BGRTia}	4-14	$\overline{BGRT}$ Signal Inactive	After R.E., CLK		65	ns
t_{IORDa}	4-8, 4-9	$\overline{IORD}$ Active	After R.E., CLK T2		40	ns
t_{IORDia}	4-8	$\overline{IORD}$ Inactive (During Indirect Transfers)	After R.E., CLK T4		40	ns
t_{IORDia}	4-9	$\overline{IORD}$ Inactive (During Direct Transfers)	After F.E., CLK T4		40	ns
t_{IOWRa}	4-7, 4-10	$\overline{IOWR}$ Active	After R.E., CLK T2		40	ns
t_{IOWRia}	4-7	$\overline{IOWR}$ Inactive (During Indirect Transfers)	After R.E., CLK T4		40	ns
$t_{IOWRdia}$	4-10	$\overline{IOWR}$ Inactive (During Direct Transfers)	After F.E., CLK T3		40	ns
4.4.2.2 Input Signal Requirements: NS32203-10						
t_{PWR}	4-22	Power Stable to $\overline{RST}/\overline{HLT}$ R.E.	After V_{CC} Reaches 4.75V	50		µs
t_{RSTw}	4-23	$\overline{RST}/\overline{HLT}$ Pulse Width (Resetting the DMAC)	at 0.8V (Both Edges)	64		tCp
t_{RSTs}	4-24	$\overline{RST}/\overline{HLT}$ Set Up Time (Resetting the DMAC)	Before F.E., CLK	15		ns
t_{HLTs}	4-18	$\overline{RST}/\overline{HLT}$ Setup Time (Halting a DMAC Transfer)	Before R.E., CLK T3	25		ns
t_{HLTh}	4-19	$\overline{RST}/\overline{HLT}$ Hold Time (Halting a DMAC Transfer)	After R.E., CLK T4	10		ns
t_{DIs}	4-6	Data in Setup Time	Before R.E., CLK T3	15		ns
t_{DIh}	4-6	Data in Hold	After R.E., CLK T4	3		ns
t_{DIs}	4-6	Data in Setup Time (Writing to DMAC Registers)	After R.E., CLK T3	15		ns
t_{DIh}	4-6	Data in Hold (Writing to DMAC Registers)	After R.E., CLK T4	3		ns
t_{HLDAs}	4-11, 4-12	$\overline{HOLDA}$ Setup Time	Before R.E., CLK	25		ns
t_{HLDAh}	4-11	$\overline{HLDA}$ Hold Time	After R.E., CLK	10		ns
t_{RDYs}	4-15	RDY Setup Time	Before R.E., CLK T2 or T3	20		ns
t_{RDYh}	4-15	RDY Hold Time	After R.E., CLK T3	5		ns
t_{REQs}	4-16, 4-17	$\overline{REQn}$ Setup Time	Before R.E., CLK	50		ns
t_{REQh}	4-16, 4-17	$\overline{REQn}$ Hold Time	After R.E., CLK	10		
t_{BREQs}	4-13	$\overline{BREQ}$ Setup Time	Before R.E., CLK	25		ns

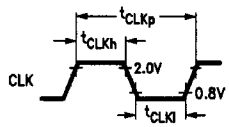
4.0 Device Specifications (Continued)

Name	Figure	Description	Reference/ Conditions	NS32203-10		Units
				Min	Max	
t_{BREQh}	4-13	$\overline{BREQ}$ Hold Time	After R.E., CLK	10		ns
$t_{ALADSis}$	4-6	Address Bits 0–5 Setup	Before $\overline{ADS}$ T.E.	20		ns
$t_{ALADSih}$	4-6	Address Bits 0–5 Hold	After $\overline{ADS}$ T.E.	20		ns
t_{HBEs}	4-6	$\overline{HBE}$ Setup Time	Before R.E., CLK T1	10		ns
t_{HBEih}	4-6	$\overline{HBE}$ Hold Time	After R.E., CLK T4	40		ns
t_{ADSs}	4-6	$\overline{ADS}$ L.E. Setup Time	Before R.E., CLK T1	40		ns
t_{ADSiw}	4-6	$\overline{ADS}$ Pulse Width	$\overline{ADS}$ L.E. to $\overline{ADS}$ T.E.	35		ns
t_{CSs}	4-6	$\overline{CS}$ Setup Time	Before R.E., CLK T1	15		ns
t_{CSh}	4-6	$\overline{CS}$ Hold Time	After R.E., CLK T4	40		ns
t_{DDINs}	4-6	$\overline{DDIN}$ Setup Time	Before R.E., CLK T2	30		ns
t_{DDINh}	4-6	$\overline{DDIN}$ Hold Time	After R.E., CLK T4	40		ns

4.4.2.3 Clocking Requirements: NS32203-10

Name	Figure	Description	Reference/ Conditions	NS32203-10		Units
				Min	Max	
t_{CLKh}	4-4	Clock High Time	At 2.0V (Both Edges)	42		ns
t_{CLK1}	4-4	Clock Low Time	At 0.8V (Both Edges)	42		ns
t_{CLKp}	4-4	Clock Period	R.E., CLK to Next R.E. CLK	100		ns

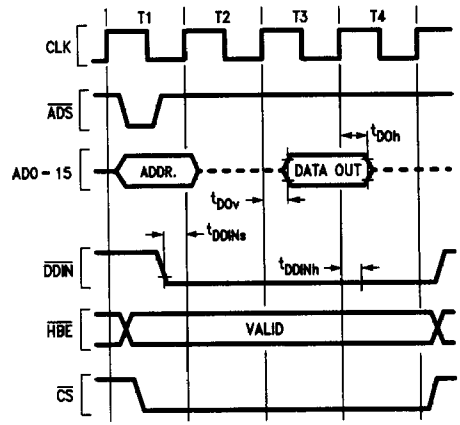
4.4.3 Timing Diagrams



TL/EE/8701–17

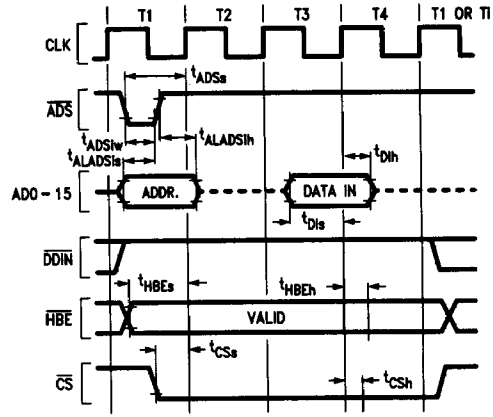
FIGURE 4-4. Clock Timing

4.0 Device Specifications (Continued)



TL/EE/8701–16

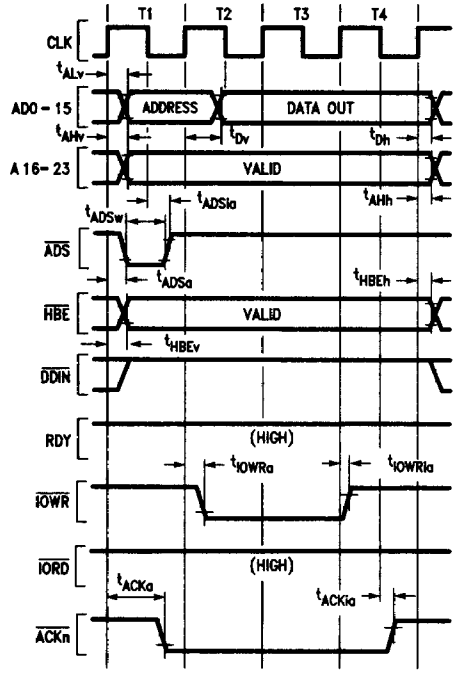
FIGURE 4-5. Read from DMAC Registers



TL/EE/8701–15

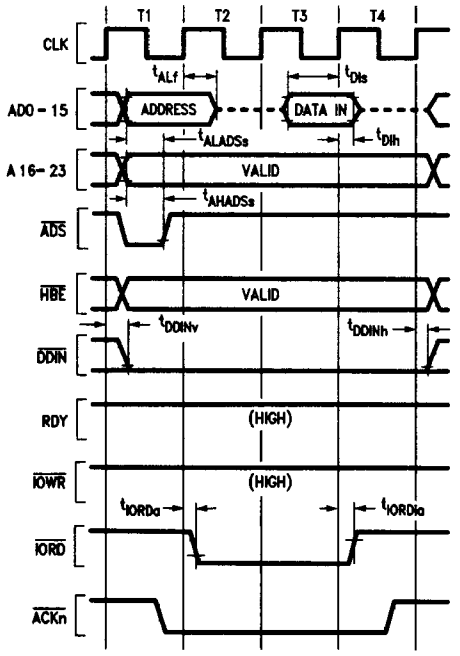
FIGURE 4-6. Write to DMAC Registers

4.0 Device Specifications (Continued)



TL/EE/8701–18

FIGURE 4-7. Indirect Write Cycle

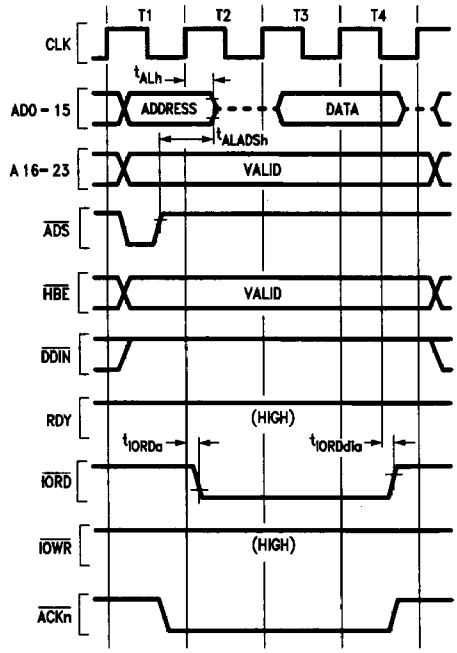


TL/EE/8701–19

FIGURE 4-8. Indirect Read Cycle

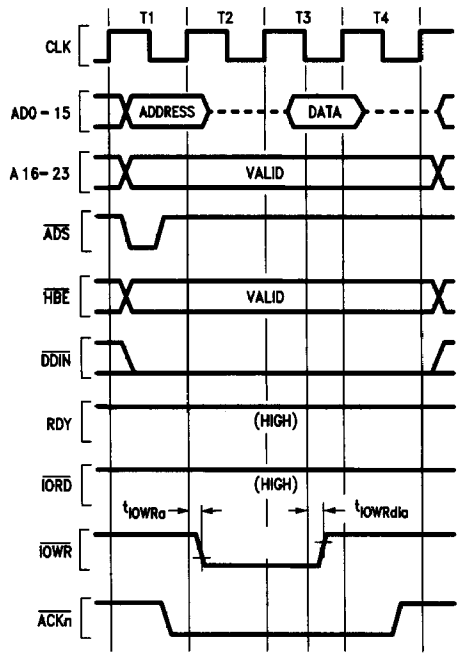
4

4.0 Device Specifications (Continued)



TL/EE/8701–20

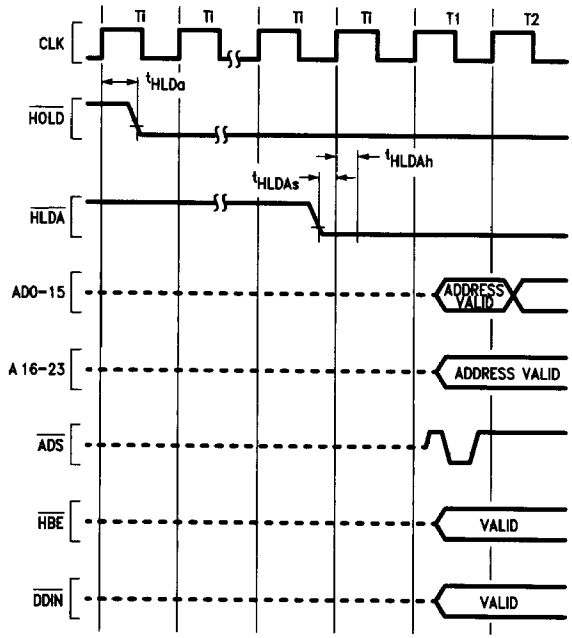
FIGURE 4-9. Direct I/O to Memory Transfer



TL/EE/8701–21

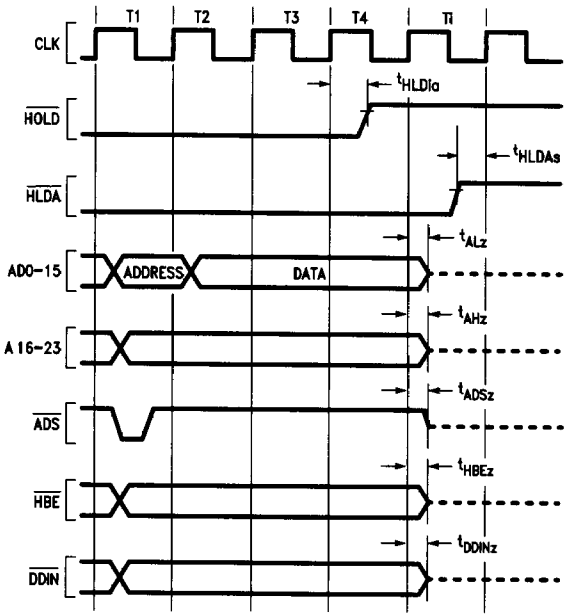
FIGURE 4-10. Direct Memory to I/O Transfer

4.0 Device Specifications (Continued)



TL/EE/8701-22

FIGURE 4-11. $\overline{\text{HOLD}}/\overline{\text{HOLDA}}$ Sequence Start



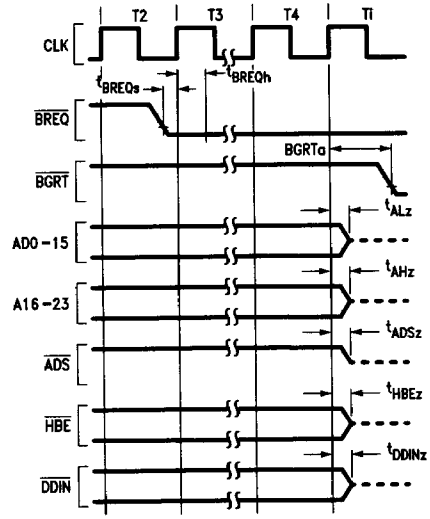
TL/EE/8701-23

FIGURE 4-12. $\overline{\text{HOLD}}/\overline{\text{HOLDA}}$ Sequence End

Note 1: DMAC in local configuration.

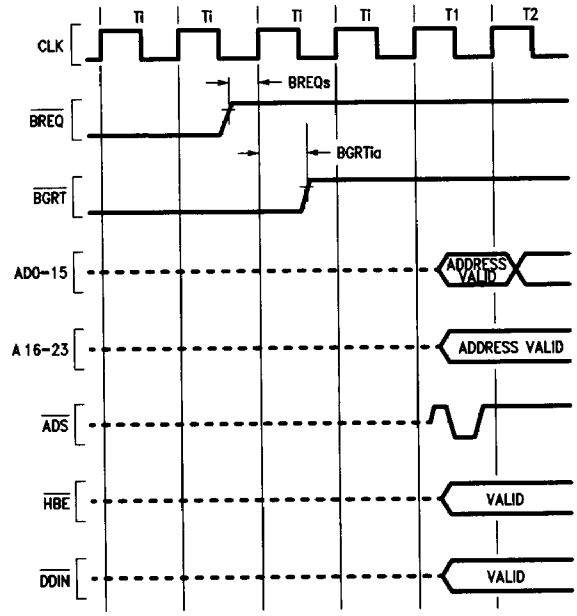
Note 2: The $\overline{\text{HOLD}}/\overline{\text{HOLDA}}$ sequence shown above is related to the single transfer mode. In burst transfer mode $\overline{\text{HOLD}}$ is deactivated two cycles later.

4.0 Device Specifications (Continued)



TL/EE/8701-24

FIGURE 4-13. Bus Request/Grant Sequence Start



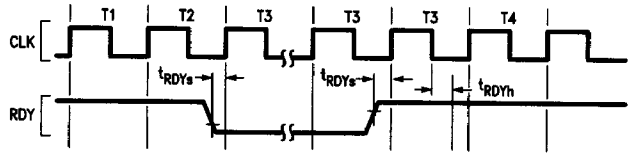
TL/EE/8701-25

FIGURE 4-14. Bus Request/Grant Sequence End

Note 1: DMAC in remote configuration.

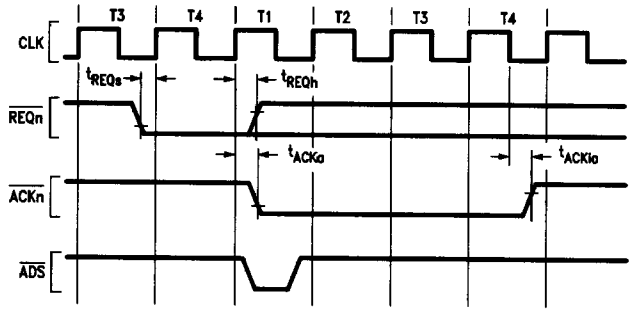
Note 2: If $\overline{\text{BREQ}}$ is asserted in the middle of a DMAC transfer, the transfer will always be completed.

4.0 Device Specifications (Continued)



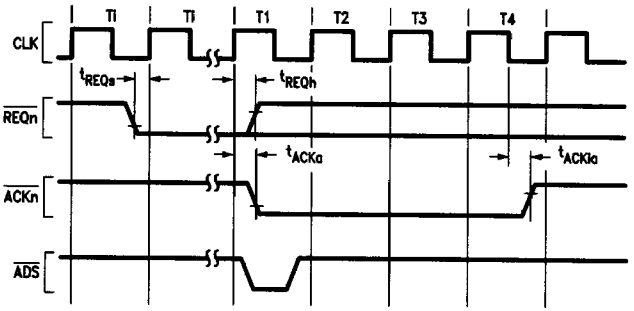
TL/EE/8701–26

FIGURE 4-15. Ready Sampling



TL/EE/8701–27

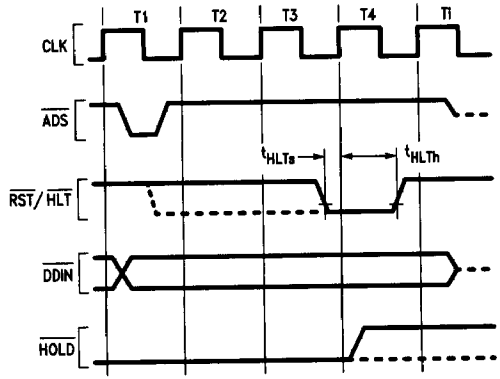
FIGURE 4-16. REQn/ACKn Sequence (DMAC Initially Not Idle)



TL/EE/8701–28

FIGURE 4-17. REQn/ACKn Sequence (DMAC Initially Idle)

4.0 Device Specifications (Continued)

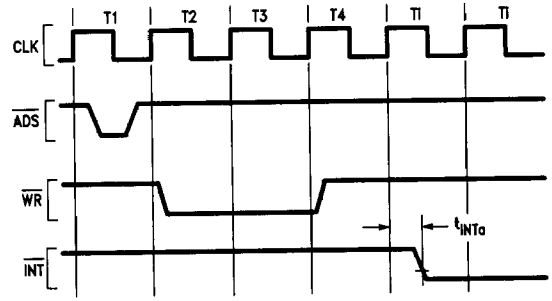


TL/EE/8701-29

FIGURE 4-18. Halted Cycle

Note 1: Halt may occur in previous T-States. It must be applied for 1 or 2 clock cycles.

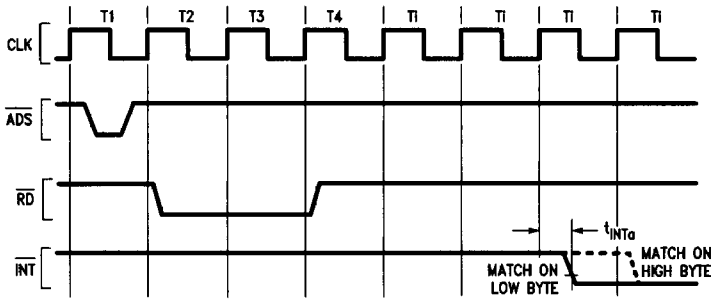
Note 2: If BREQ is asserted in the middle of a DMAC transfer, the transfer will always be completed.



TL/EE/8701-30

FIGURE 4-19. Interrupt on Transfer Complete

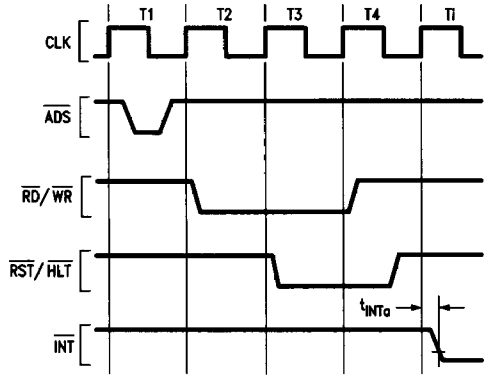
4.0 Device Specifications (Continued)



TL/EE/8701–31

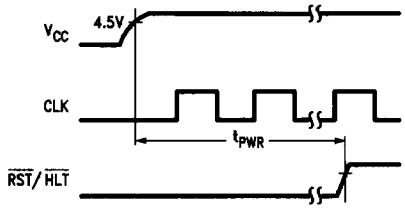
FIGURE 4-20. Interrupt on Match/No Match

Note: If inclusive search is specified a write cycle is performed before $\overline{INT}$ is activated.



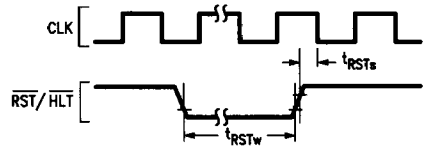
TL/EE/8701–32

FIGURE 4-21. Interrupt on Halt



TL/EE/8701–33

FIGURE 4-22. Power on Reset



TL/EE/8701–34

FIGURE 4-23. Non Power on Reset

4

Appendix A. Interfacing Suggestions

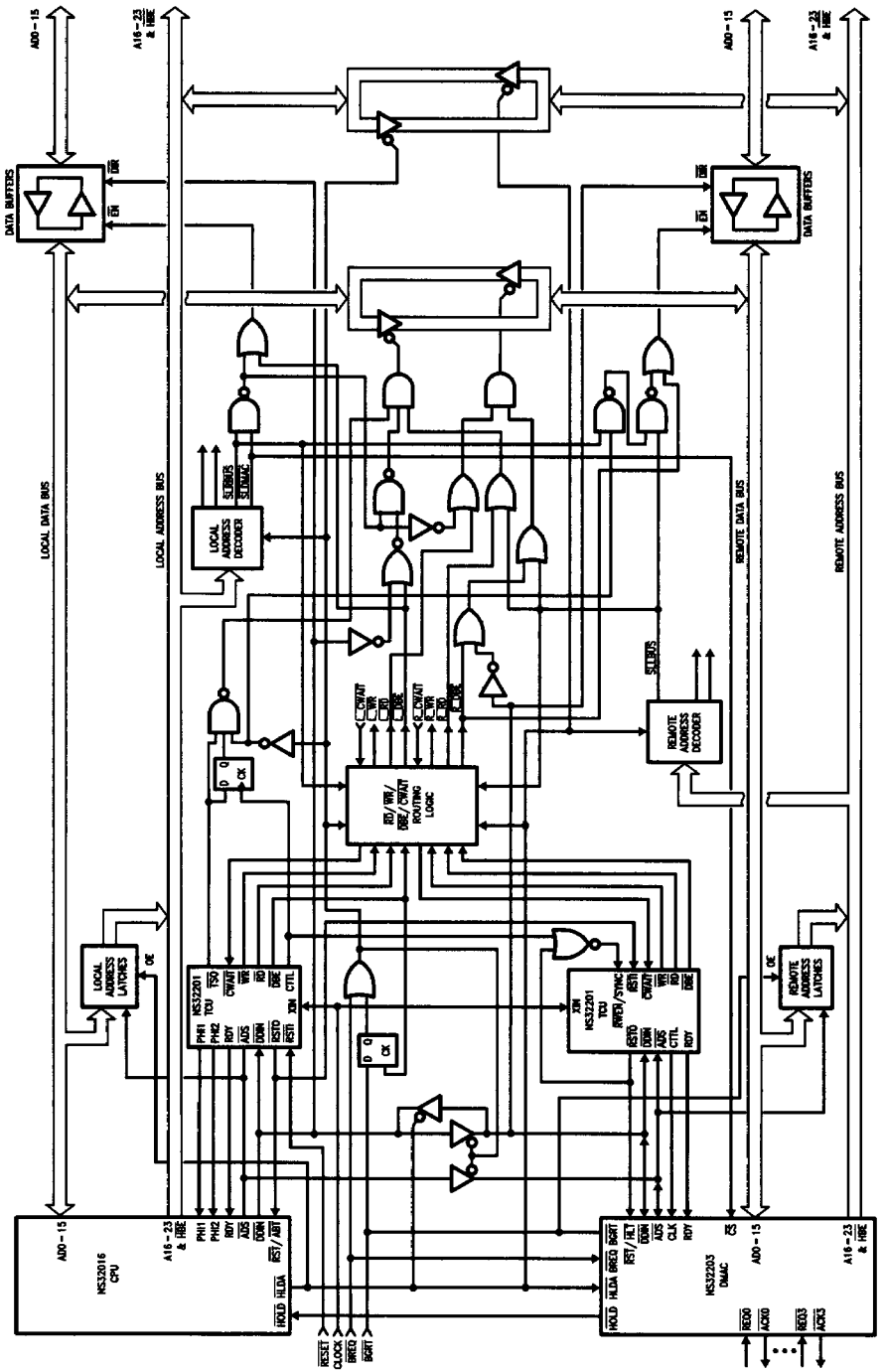


FIGURE A-1. NS32203 Interconnections in Remote Configuration.

Note: This logic does not support direct (flyby) DMAC transfers.