Z8001*/Z8002*

16-Bit Microprocessors

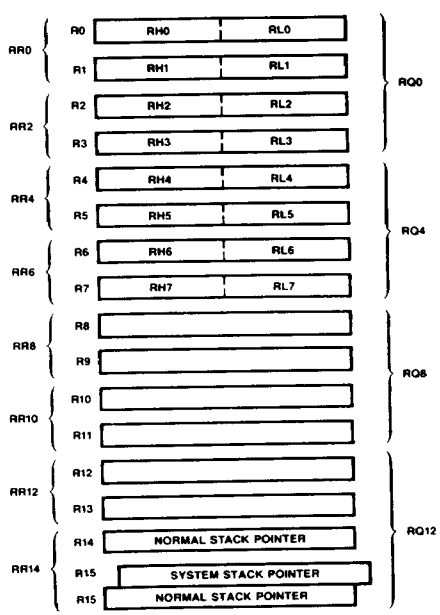
DISTINCTIVE CHARACTERISTICS

- **4, 6, 8 MHz CPU Clock**
  High throughput with low system clock rate for easier system design
- **Powerful General Register Architecture**
  16 general registers provide high throughput in all types of applications.
- **Wide Variety of Data Types**
  Instructions operate on bits, bytes, 16- and 32-bit words for efficient programming of a wide variety of functions.
- **Partitioned for Operating System Protection**
  Hardware bit protects privileged instructions from execution except by operating system.
- **Supports 3 Types of Interrupts**
  Separate pins provided for vectored, non-vectored and non-maskable interrupts
- **Two Compatible CPUs**
  Compact 40-pin Z8002 supports 64KB memory; larger 48-pin Z8001 supports 8MB memory.

GENERAL DESCRIPTION

The Z8001* is a general-purpose 16-bit CPU belonging to the Z8000 family of microprocessors. Its architecture is centered around sixteen 16-bit general registers. The CPU deals with 23-bit address spaces and hence can address directly 8MB of memory. The 23-bit address consists of two components: 7-bit segment number and 16-bit offset. Facilities are provided to maintain three distinct address spaces — code, data and stack. The Z8001 implements a powerful instruction set with flexible addressing modes. These instructions operate on several data types — bit, byte, word (16-bit), long word (32-bit), byte string and word string. The CPU can execute instructions in one of two modes — System and Normal. Sometimes these modes are also known as Privileged and Non-Privileged, respectively. The CPU also contains an on-chip memory refresh facility. The Z8001 is software compatible with the Z8002 microprocessor. The Z8001 is fabricated using silicon-gate N-MOS technology and is packaged in a 48-pin DIP. The Z8001 requires a single +5 power supply and a single phase clock for its operation.

GENERAL REGISTERS

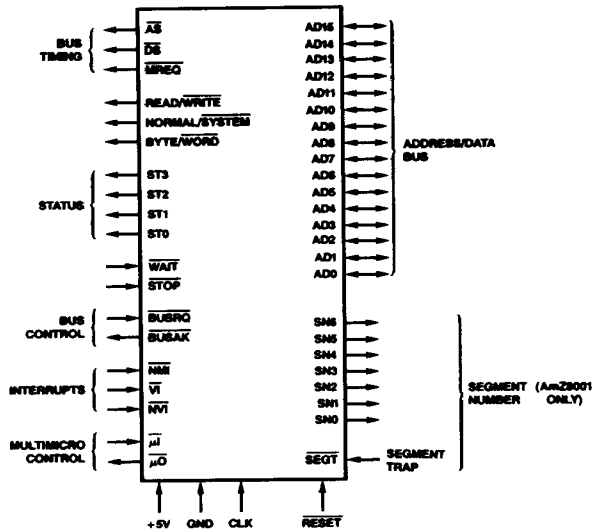


AF002630

*Z8000, Z8001, and Z8002 are trademarks of Zilog, Inc.

Publication # 00971 | Rev. C | Amendment /0
Issue Date: May 1987

LOGIC SYMBOL



LS001272

ORDERING INFORMATION – Z8001

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option
c. Package Type
d. Temperature Range
e. Optional Processing

Z8001 A D C B

e. OPTIONAL PROCESSING
Blank = Standard processing
B = Burn-in

d. TEMPERATURE RANGE*
C = Commercial (0 to +70°C)

c. PACKAGE TYPE
P = 48-Pin Plastic DIP (PD 048)
D = 48-Pin Sidebrazed Ceramic DIP (SD 048)

b. SPEED OPTION
Blank = 4 MHz
A = 6 MHz
A-8 = 8 MHz

a. DEVICE NUMBER/DESCRIPTION
Z8001
16-Bit Segmented Microprocessor

Valid Combinations	
Z8001	PC, DC, DCB
Z8001A	
Z8001A-8	

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order #09275A/0) for electrical performance characteristics.

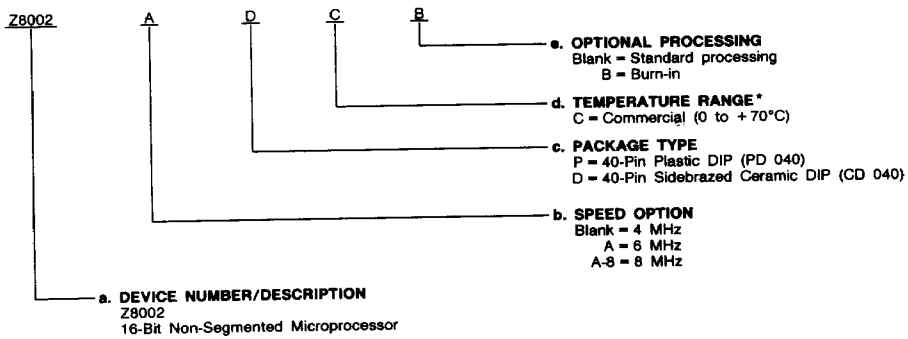
5

ORDERING INFORMATION – Z8002

Commodity Products

AMD commodity products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
Z8002	PC, DC, DCB
Z8002A	
Z8002A-8	

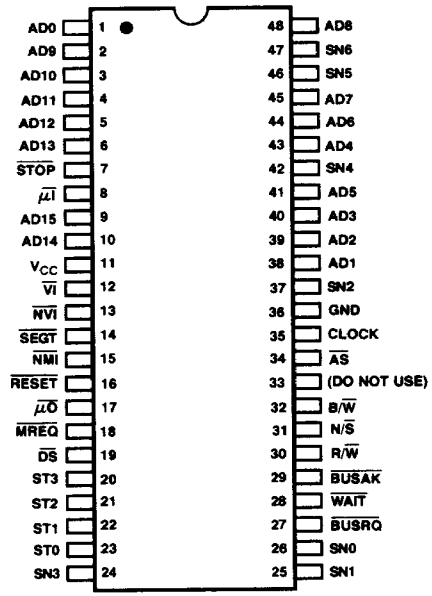
Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

*This device is also available in Military temperature range. See MOS Microprocessors and Peripherals Military Handbook (Order #09275A/0) for electrical performance characteristics.

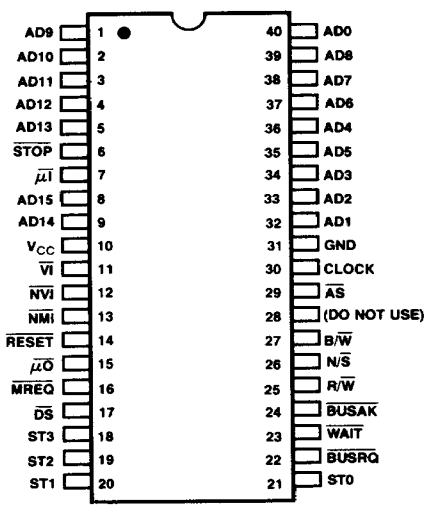
CONNECTION DIAGRAMS
Top View

Z8001



CD005262

Z8002



CD005272

Note: Pin 1 is marked for orientation.

5

Z8001 PIN DESCRIPTION

Pin No.	Name	I/O	Description
11	V_{CC}		+5V Power Supply.
36	V_{SS}		Ground.
1, 38-40, 43, 41, 44, 45, 48, 2-6, 10, 9	AD0-AD15	I/O	Bidirectional, 3-State. Address/Data Bus. This 16-bit multiplexed address/data bus is used for all I/O and memory transactions. HIGH on the bus corresponds to 1 and LOW corresponds to 0. AD0 is the least significant bit position with AD15 is most significant. The $\overline{AS}$ output and $\overline{DS}$ output will indicate whether the bus is used for address offset or data. The status output lines ST0-ST3 will indicate the type of transaction: memory or I/O.
34	$\overline{AS}$	O	3-State. Address Strobe. LOW on this output indicates that the AD0-AD15 bus contains address information. The address information is stable by the time of the LOW-to-HIGH transition of the $\overline{AS}$ output (see timing diagram). The status outputs ST0-ST3 indicate whether the bus contains a memory address or I/O address.
19	$\overline{DS}$	O	3-State. Data Strobe. LOW on this output indicates that the AD0-AD15 bus is being used for data transfer. The R/$\overline{W}$ output indicates the direction of data transfer – read (or in) means data into the CPU and write (or out) means data from the CPU. During a read operation, data can be gated on to the bus where $\overline{DS}$ goes LOW. A LOW-to-HIGH transition on the $\overline{DS}$ output indicates that the CPU has accepted the data (see timing diagram). During a write operation, LOW on the $\overline{DS}$ output indicates that data is set up on the bus. Data will be removed sometime after the LOW-to-HIGH transition of the $\overline{DS}$ output (see timing diagram).
30	R/$\overline{W}$	O	3-State. Read/Write. This output indicates the direction of data flow on the AD0-AD15 bus. HIGH indicates a read operation, i.e., data into the CPU, and LOW indicates a write operation, i.e., data from the CPU. This output is activated at the same time as $\overline{AS}$ going LOW and remains stable for the duration of the whole transaction (see timing diagram).
32	B/$\overline{W}$	O	3-State. Byte/Word. This output indicates the type of data transferred on the AD0-AD15 bus. HIGH indicates byte (8-bit) and LOW indicates word (16-bit) transfer. This output is activated at the same stage as $\overline{AS}$ going LOW and remains valid for the duration of the whole transaction (see timing diagram). The address generated by the CPU is always a byte address. However, the memory is organized as 16-bit words. All instructions and word operands are word aligned and are addressed by even addresses. Thus, for all word transactions with the memory, the least significant address bit will be zero. When addressing the memory for byte transactions, the least significant address bit determines which byte of the memory word is needed; even address specifies the most significant byte, and odd address specifies the least significant byte. In the case of I/O transactions, the address information on the AD0-AD15 bus refers to an I/O port, and B/$\overline{W}$ determines whether a data word or data byte will be transacted. During I/O byte transactions, the least significant address bit A0 determines which half of the AD0-AD15 bus will be used for the I/O transactions. The ST0-ST3 outputs will indicate whether the current transaction is for memory, normal I/O or special I/O.
23-20	ST0-ST3	O	3-State. Status. These four outputs contain information regarding the current transaction in a coded form. The status line codes are shown in the table on the following page.
28	$\overline{WAIT}$	I	Wait. LOW on this input indicates to the CPU that memory or I/O is not ready for the data transfer, and hence, the current transaction should be stretched. The $\overline{WAIT}$ input is sampled by the CPU at certain instances during the transaction (see timing diagram). If $\overline{WAIT}$ input is LOW at these instances, the CPU will go into wait state to prolong the transaction. The wait state will repeat until the $\overline{WAIT}$ input is HIGH at the sampling instant.
31	N/$\overline{S}$	O	3-State. Normal/System Mode. HIGH on this output indicates that the CPU is operating in Normal Mode and LOW indicates operation in System Mode. This output is derived from the Flag Control Word (FCW) register. The FCW register is described under the processor status information section of this document.
18	$\overline{MREQ}$	O	3-State. Memory Request. LOW on this output indicates that a CPU transaction with memory is taking place.
27	$\overline{BUSRQ}$	I	Bus Request. LOW on this input indicates to the CPU that another device (such as DMA) is requesting to take control of the bus. The $\overline{BUSRQ}$ input can be driven LOW anytime. The CPU synchronizes this input internally. The CPU responds by activating $\overline{BUSAK}$ output LOW to indicate that the bus has been relinquished. Relinquishing the bus means that the AD0-AD15, $\overline{AS}$, $\overline{DS}$, B/$\overline{W}$, R/$\overline{W}$, N/$\overline{S}$, ST0-ST3, SN0-SN6 and $\overline{MREQ}$ outputs will be in the high impedance state. The requesting device should control these lines in an identical fashion to the CPU to accomplish transactions. The $\overline{BUSRQ}$ input must remain LOW as long as needed to perform all the transactions and the CPU will keep the $\overline{BUSAK}$ output LOW. After completing the transactions, the device must disable the AD0-AD15, $\overline{AS}$, $\overline{DS}$, B/$\overline{W}$, R/$\overline{W}$, N/$\overline{S}$, ST0-ST3, SN0-SN6 and $\overline{MREQ}$ into the high impedance state and stop driving the $\overline{BUSRQ}$ input LOW. The CPU will make $\overline{BUSAK}$ output HIGH sometime later and take back the bus control.
29	$\overline{BUSAK}$	O	Bus Acknowledge. LOW on this output indicates that the CPU has relinquished the bus in response to a bus request.
15	$\overline{NMI}$	I	Non-Maskable Interrupt. A HIGH-to-LOW transition on this input constitutes non-maskable interrupt request. The CPU will respond with the Non-maskable Interrupt Acknowledge on the ST0-ST3 outputs and will enter an interrupt sequence. The transition on the $\overline{NMI}$ can occur anytime. Of the three kinds of interrupts available, the non-maskable interrupt has the highest priority.
12	$\overline{VI}$	I	Vectored Interrupt. LOW on this input constitutes vectored interrupt request. Vectored interrupt is next lower to the non-maskable interrupt in priority. The NVIE bit in the Flag and Control Word register must be 1 for the vectored interrupt to be honored. The CPU will respond with Vectored Interrupt Acknowledge code on the ST0-ST3 outputs and will begin the interrupt sequence. The $\overline{VI}$ input can be driven LOW anytime and should be held LOW until acknowledged.

Z8001 PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
13	$\overline{NVI}$	I	Non-Vectored Interrupt. LOW on this input constitutes non-vectored interrupt request. Non-vectored has the lowest priority of the three types of interrupts. The NVIE bit in the Flag and Control Word register must be 1 for this request to be honored. The CPU will respond with Non-Vectored Interrupt Acknowledge code on the ST0-ST3 outputs and will begin the interrupt sequence. The $\overline{NVI}$ input can be driven LOW anytime and should be held LOW until acknowledged.
8	$\overline{\mu I}$	I	Micro-In. This input participates in the resource request daisy chain. See the section on multimicroprocessor support facilities in this document.
17	$\overline{\mu O}$	O	Micro-Out. This output participates in the resource request daisy chain. See the section on multimicroprocessor support facilities in this document.
16	$\overline{RESET}$	I	Reset. LOW on this input initiates a reset sequence in the CPU. See the section on Initialization for details on reset sequence.
35	CLK	I	Clock. All CPU operations are controlled from the signal fed into this input. See DC Characteristics for clock voltage level requirements.
7	$\overline{STOP}$	I	Stop. This active LOW input facilitates one instruction at a time operation. See the section on single stepping.
26, 25, 37, 24, 42, 46, 47	SN0-SN6	O	3-State. Segment Number. These seven outputs contain the segment number part of a memory address. A HIGH on the output corresponds to 1 and a LOW corresponds to 0. SN0 is the least significant bit position and SN6 is the most significant bit position.
14	$\overline{SEGT}$	I	Segment Trap. LOW on this input constitutes a segment trap request. If the line is driven LOW, the CPU will respond with the Segment Trap Acknowledge code on the Status lines and commence a trap sequence. The $\overline{SEGT}$ input may be driven LOW at any time and is customarily held LOW until acknowledged. This input has priority over the interrupts.

Status Line Codes

ST3	ST2	ST1	ST0	
L	L	L	L	Internal Operation
L	L	L	H	Memory Refresh
L	L	H	L	Normal I/O Transaction
L	L	H	H	Special I/O Transaction
L	H	L	L	Segment Trap Acknowledge
L	H	L	H	Non-Maskable Interrupt Acknowledge
L	H	H	L	Non-Vectored Interrupt Acknowledge
L	H	H	H	Vectored Interrupt Acknowledge
H	L	L	L	Memory Transaction for Operand
H	L	L	H	Memory Transaction for Stack
H	L	H	L	Reserved
H	L	H	H	Reserved
H	H	L	L	Memory Transaction for Instruction Fetch (Subsequent Word)
H	H	L	H	Memory Transaction for Instruction Fetch (First Word)
H	H	H	L	Reserved
H	H	H	H	Reserved

Z8002 PIN DESCRIPTION

Pin No.	Name	I/O	Description
10	V_{CC}		+5V Power Supply.
31	V_{SS}		Ground.
40, 32-39, 1-5, 9, 8	AD0-AD15	I/O	Bidirectional, 3-State. Address/Data Bus. This 16-bit multiplexed address/data bus is used for all I/O and memory transactions. HIGH on the bus corresponds to 1 and LOW corresponds to 0. AD0 is the least significant bit position with AD15 is most significant. The $\overline{AS}$ output and $\overline{DS}$ output will indicate whether the bus is used for address offset or data. The status output lines ST0-ST3 will indicate the type of transaction: memory or I/O.
29	$\overline{AS}$	O	3-State. Address Strobe. LOW on this output indicates that the AD0-AD15 bus contains address information. The address information is stable by the time of the LOW-to-HIGH transition of the $\overline{AS}$ output (see timing diagram). The status outputs ST0-ST3 indicate whether the bus contains a memory address or I/O address.
17	$\overline{DS}$	O	3-State. Data Strobe. LOW on this output indicates that the AD0-AD15 bus is being used for data transfer. The R/$\overline{W}$ output indicates the direction of data transfer – read (or in) means data into the CPU and write (or out) means data from the CPU. During a read operation, data can be gated on to the bus where $\overline{DS}$ goes LOW. A LOW-to-HIGH transition on the $\overline{DS}$ output indicates that the CPU has accepted the data (see timing diagram). During a write operation, LOW on the $\overline{DS}$ output indicates that data is set up on the bus. Data will be removed sometime after the LOW-to-HIGH transition of the $\overline{DS}$ output (see timing diagram).
25	R/$\overline{W}$	O	3-State. Read/Write. This output indicates the direction of data flow on the AD0-AD15 bus. HIGH indicates a read operation, i.e., data into the CPU, and LOW indicates a write operation, i.e., data from the CPU. This output is activated at the same time as $\overline{AS}$ going LOW and remains stable for the duration of the whole transaction (see timing diagram).
27	B/$\overline{W}$	O	3-State. Byte/Word. This output indicates the type of data transferred on the AD0-AD15 bus. HIGH indicates byte (8-bit) and LOW indicates word (16-bit) transfer. This output is activated at the same stage as $\overline{AS}$ going LOW and remains valid for the duration of the whole transaction (see timing diagram). The address generated by the CPU is always a byte address. However, the memory is organized as 16-bit words. All instructions and word operands are word aligned and are addressed by even addresses. Thus, for all word transactions with the memory, the least significant address bit will be zero. When addressing the memory for byte transactions, the least significant address bit determines which byte of the memory word is needed; even address specifies the most significant byte, and odd address specifies the least significant byte. In the case of I/O transactions, the address information on the AD0-AD15 bus refers to an I/O port, and B/$\overline{W}$ determines whether a data word or data byte will be transacted. During I/O byte transactions, the least significant address bit A0 determines which half of the AD0-AD15 bus will be used for the I/O transactions. The ST0-ST3 outputs will indicate whether the current transaction is for memory, normal I/O or special I/O.
21-18	ST0-ST3	O	3-State. Status. These four outputs contain information regarding the current transaction in a coded form (see table on previous page).
23	$\overline{WAIT}$	I	Wait. LOW on this input indicates to the CPU that memory or I/O is not ready for the data transfer, and hence, the current transaction should be stretched. The $\overline{WAIT}$ input is sampled by the CPU at certain instances during the transaction (see timing diagram). If $\overline{WAIT}$ input is LOW at these instances, the CPU will go into wait state to prolong the transaction. The wait state will repeat until the $\overline{WAIT}$ input is HIGH at the sampling instant.
26	N/$\overline{S}$	O	3-State. Normal/System Mode. HIGH on this output indicates that the CPU is operating in Normal Mode and LOW indicates operation in System Mode. This output is derived from the Flag Control Word (FCW) register. The FCW register is described under the processor status information section of this document.
16	$\overline{MREQ}$	O	3-State. Memory Request. LOW on this output indicates that a CPU transaction with memory is taking place.
22	$\overline{BUSRQ}$	I	Bus Request. LOW on this input indicates to the CPU that another device (such as DMA) is requesting to take control of the bus. The $\overline{BUSRQ}$ input can be driven LOW anytime. The CPU synchronizes this input internally. The CPU responds by activating $\overline{BUSAK}$ output LOW to indicate that the bus has been relinquished. Relinquishing the bus means that the AD0-AD15, $\overline{AS}$, $\overline{DS}$, B/$\overline{W}$, R/$\overline{W}$, N/$\overline{S}$, ST0-ST3, SN0-SN6 and $\overline{MREQ}$ outputs will be in the high impedance state. The requesting device should control these lines in an identical fashion to the CPU to accomplish transactions. The $\overline{BUSRQ}$ input must remain LOW as long as needed to perform all the transactions and the CPU will keep the $\overline{BUSAK}$ output LOW. After completing the transactions, the device must disable the AD0-AD15, $\overline{AS}$, $\overline{DS}$, B/$\overline{W}$, R/$\overline{W}$, N/$\overline{S}$, ST0-ST3, SN0-SN6 and $\overline{MREQ}$ into the high impedance state and stop driving the $\overline{BUSRQ}$ input LOW. The CPU will make $\overline{BUSAK}$ output HIGH sometime later and take back the bus control.
24	$\overline{BUSAK}$	O	Bus Acknowledge. LOW on this output indicates that the CPU has relinquished the bus in response to a bus request.
13	$\overline{NMI}$	I	Non-Maskable Interrupt. A HIGH-to-LOW transition on this input constitutes non-maskable interrupt request. The CPU will respond with the Non-maskable Interrupt Acknowledge on the ST0-ST3 outputs and will enter an interrupt sequence. The transition on the $\overline{NMI}$ can occur anytime. Of the three kinds of interrupts available, the non-maskable interrupt has the highest priority.
11	$\overline{VI}$	I	Vectored Interrupt. LOW on this input constitutes vectored interrupt request. Vectored interrupt is next lower to the non-maskable interrupt in priority. The VIE bit in the Flag and Control Word register must be 1 for the vectored interrupt to be honored. The CPU will respond with Vectored Interrupt Acknowledge code on the ST0-ST3 outputs and will begin the interrupt sequence. The $\overline{VI}$ input can be driven LOW anytime and should be held LOW until acknowledged.
12	$\overline{NVI}$	I	Non-Vectored Interrupt. LOW on this input constitutes non-vectored interrupt request. Non-vectored has the lowest priority of the three types of interrupts. The NVIE bit in the Flag and Control Word register must be 1 for this request to be honored. The CPU will respond with Non-Vectored Interrupt Acknowledge code on the ST0-ST3 outputs and will begin the interrupt sequence. The $\overline{NVI}$ input can be driven LOW anytime and should be held LOW until acknowledged.
7	$\overline{\mu I}$	I	Micro-In. This input participates in the resource request daisy chain. See the section on multimicroprocessor support facilities in this document.
15	$\overline{\mu O}$	O	Micro-Out. This output participates in the resource request daisy chain. See the section on multimicroprocessor support facilities in this document.
14	$\overline{RESET}$	I	Reset. LOW on this input initiates a reset sequence in the CPU. See the section on Initialization for details on reset sequence.
30	CLK	I	Clock. All CPU operations are controlled from the signal fed into this input. See DC Characteristics for clock voltage level requirements.
6	$\overline{STOP}$	I	Stop. This active LOW input facilitates one instruction at a time operation. See the section on single stepping.

DETAILED DESCRIPTION

The following is a description of the Z8001 and Z8002 CPUs.

General Purpose Registers

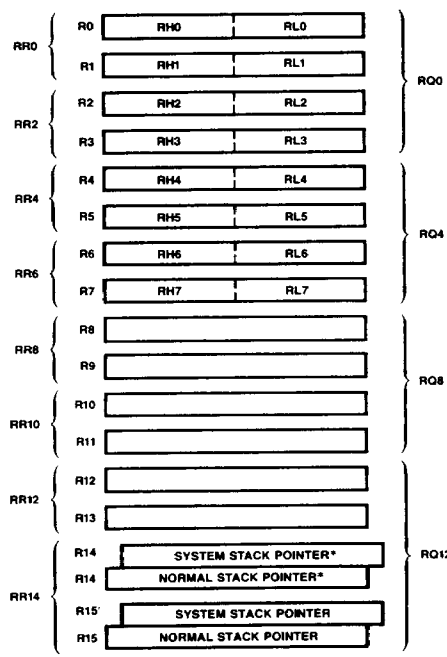
The CPU is organized around sixteen 16-bit general purpose registers R0 through R15 as shown in Figure 1. For byte operations, the first eight registers (R0 through R7) can also be addressed as sixteen 8-bit registers designated as RL0, RH0 and so on to RL7 and RH7. The sixteen registers can also be grouped in pairs RR0, RR2 and so on to RR14 to form eight long word (32-bit) registers. Similarly, the sixteen registers can be grouped in quadruples RQ0, RQ4, RQ8 and RQ12 to form four 64-bit registers.

Stack Pointer (Z8001)

The Z8001 architecture allows stacks to be maintained in memory. Any general-purpose register pair except RR0 can be used as a stack pointer in stack manipulating instructions, such as PUSH and POP. The designated register pair holds a 23-bit segmented address. Certain instructions (such as subroutine call and return) make implicit use of the register pair RR14 as the stack pointer. Two implicit stacks are allowed — normal stack using RR14 as the stack pointer and system stack using RR14′ as the system stack pointer (see Figure 1). If the CPU is operating in the Normal Mode, RR14 is active, and if the CPU is in System Mode, RR14′ will be used instead of RR14. The implied stack pointer is a part of the general registers and hence can be manipulated using the instructions available for register operations.

Stack Pointer (Z8002)

The Z8002 architecture allows stacks to be maintained in the memory. Any general purpose register except R0 can be used as a stack pointer in stack manipulating instructions, such as PUSH and POP. However, certain instructions such as subroutine call and return make implicit use of the register R15 as the stack pointer. Two implicit stacks are maintained — normal stack using R15 as the stack pointer and system stack using R15′ as the system stack pointer (see Figure 1). If the CPU is operating in the Normal Mode, R15 is active, and if the CPU is in System Mode, R15′ will be used instead of R15. The implied stack pointer is a part of the general registers and hence can be manipulated using the instructions available for register operations.



*AmZ8001 only

AF002640

Figure 1. CPU General Registers

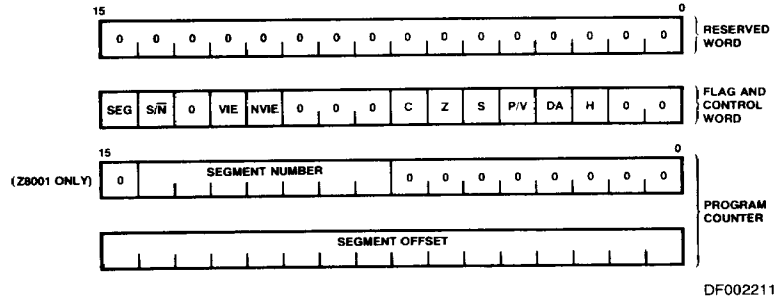


Figure 2. CPU Processor Status

Processor Status

The CPU status consists of the 16-bit flag and control word (FCW) register, and the 16- or 23-bit program counter (see Figure 2). A reserved word is also included for future expansion. The following is a brief description of the FCW bits.

SEG: Segmented/Non-Segmented Bit. Indicates whether the Z8001 is running in segmented or non-segmented mode. 1 indicates segmented; 0 indicates non-segmented. See the section on non-segmented mode, elsewhere in this document. This bit is always 0 in the Z8002.

S/$\overline{N}$: System/Normal – 1 indicates System Mode, and 0 indicates Normal Mode.

VIE: Vectored Interrupt Enable – 1 indicates that Vectored Interrupt requests will be honored.

NVIE: Non-Vectored Interrupt Enable – 1 indicates that Non-vectored interrupt requests will be honored.

C: Carry – 1 indicates that a carry has occurred from the most significant bit position when performing arithmetic operations.

Z: Zero – 1 indicates that the result of an operation is zero.

S: Sign – 1 indicates that the result of an operation is negative; i.e., most significant bit is one.

P/V: Parity/Overflow – 1 indicates that there was an overflow during arithmetic operations. For byte logical operations, this bit indicates parity of the result.

DA: Decimal Adjust – Records byte arithmetic operations.

H: Half Carry – 1 indicates that there was a carry from the most significant bit of the lower digit during byte arithmetic.

Data Types

The CPU instructions operate on bits, digits (4 bits), bytes (8 bits), words (16 bits), long words (32 bits), byte strings and word strings type operands. Bits can be set, reset or tested. Digits are used to facilitate BCD arithmetic operations. Bytes are used for characters and small integers. Words are used for integer values and addresses while long words are used for large integer values and addresses. All operands except strings can reside either in memory or general registers. Strings can reside in memory only.

Interrupt and Trap Structure

Interrupt is defined as an external asynchronous event requiring program interruption. For example, interruption is caused by a peripheral needing service. Traps are synchronous events resulting from execution of certain instructions under some defined circumstances. Both interrupts and traps are handled in a similar manner.

The CPU supports three types of interrupts in order of descending priority — non-maskable, vectored and non-vectored. The vectored and non-vectored interrupts can be disabled by appropriate control bits in the FCW. The CPU has four traps — system call, segment trap, unimplemented opcode and privileged instruction. The traps have higher priority than interrupts.

When an interrupt or trap occurs, the current program status is automatically pushed onto the system stack. The program status consists of processor status (i.e., PC and FCW) plus a 16-bit identifier. The identifier contains the reason, source and other coded information relating to the interrupt or trap.

After saving the current program status, the new processor status is automatically loaded from the new program status area located in the memory. This area is designated by the New Program Status Area Pointer (NPSAP) register.

Segmented Addressing (Z8001 Only)

The Z8001 can directly address up to 8MB of memory space, using a 23-bit segmented address. The memory space is divided up into 128 segments, each up to 64kB in size. The upper seven bits of address designate the segment number and are available on the SN0-SN6 outputs during a memory transaction. See the section on memory transactions for details.

The lower sixteen bits of address designate an offset within the segment, relative to the start of the segment, and are available on AD0-AD15 during part of the memory transaction. See the section on memory transactions for details.

The segmented address may be stored as a long word in memory or in a register pair. The segment number and offset can be manipulated separately or together by suitable use of the instruction set.

When the segmented address is contained in code space, a short offset format may be adopted. The segmented address is stored as one word, seven bits of segment number and eight bits of offset. Figure 3 shows the format for segmented addresses.

Addressing Modes

Information contained in the CPU instructions consists of the operation to be performed, the operand type and the location of the operands. Operand locations are designated by general register addresses, memory addresses or I/O addresses. The addressing mode of a given instruction defines the address space referenced and the method to compute the operand address. Addressing modes are explicitly specified or implied in an instruction. Figure 4 illustrates the eight explicit address-

ing modes: Register (R), Immediate (IM), Indirect Register (IR), Direct Address (DA), Indexed (X), Relative Address (RA), Base Address (BA) and Base Indexed (BX).

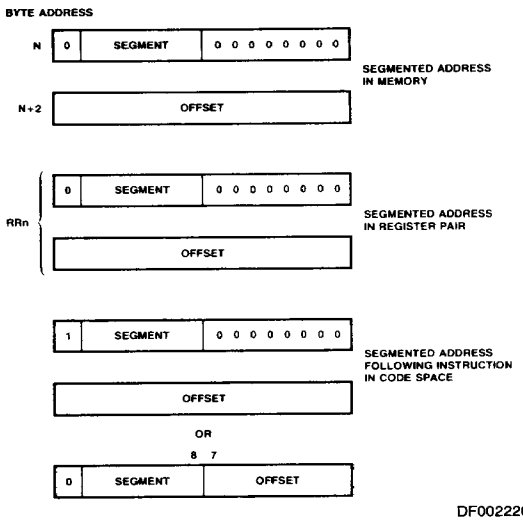


Figure 3. Segmented Address Formats

When an effective segmented address is being computed according to the designated addressing mode, the segment number is not affected by any carry from the 16-bit offset.

Non-Segmented Mode on the Z8001

The Z8001 can execute code designed to run on the non-segmented Z8002. This is achieved by changing the mode of execution of the Z8001 from segmented to non-segmented by writing a 0 to the SEG bit in the FCW. (See the section on processor status.) The change to non-segmented mode sets up a suitable environment for running non-segmented code. However, the environment only exists within the code segment that caused the change of mode from segmented to non-segmented.

SN0-SN6 will continue to indicate the code segment until a reset, interruption or return to segmented mode is encountered.

The effects of the non-segmented mode of operation on the Z8001 are described below:

a) The Z8001 will interpret instruction length like it was a non-segmented Z8002.

b) The Z8001 will implement address computation in an identical manner to the Z8002.

Other CPU functions, such as interrupt and trap handling, reset and stack pointer manipulation, are unaltered. These functions are characterized by the type of CPU, not by the state of the SEG bit in the FCW.

Input/Output

A set of I/O instructions are provided to accomplish byte or word transfers between the CPU and I/O devices. I/O devices are addressed using 16-bit I/O port addresses, and I/O address space is not a part of the memory address space. Two types of I/O instructions are provided, each with its own 16-bit address space. I/O instructions include a comprehensive set of In, Out and Block transfers.

CPU Timing

The CPU accomplishes instruction execution by stepping through a pre-determined sequence of machine cycles, such as memory read, memory write, etc. Each machine cycle requires between three and ten clock cycles. But Requests by DMA devices are granted at machine cycle boundaries. No machine cycle is longer than ten clock cycles, thus assuring fast response to a Bus Request (assuming no extra wait states). The start of a machine cycle is always marked by a LOW pulse on the $\overline{AS}$ output. The status output lines ST0-ST3 indicate the nature of the current cycle in a coded form.

Status Line Codes

Status line coding was listed in the table shown under Pin Description. The following is a detailed description of the status codes.

Internal Operation:

This status code indicates that the CPU is going through a machine cycle for its internal operation. Figure 5 depicts an internal operation cycle. It consists of three clock periods identified as T1, T2 and T3. The $\overline{AS}$ output will be activated with a LOW pulse by the Z8001 to mark the start of a machine cycle. The ST0-ST3 will reflect the code for the internal operation. The $\overline{MREQ}$, $\overline{DS}$ and R/$\overline{W}$ outputs will be HIGH. The N/$\overline{S}$ and SN0-SN6 outputs will remain at the same level as in the previous machine cycle. The CPU will ignore the $\overline{WAIT}$ input during the internal operation cycle. The CPU will drive the AD0-AD15 bus with unspecified information during T1. However, the bus will go into high-impedance during T2 and remain in that state for the remainder of the cycle. The B/$\overline{W}$ output is also activated by the CPU with unspecified information.

Memory Refresh:

This status code indicates that CPU is accessing the memory to refresh. The refresh cycle consists of three clock periods as depicted in Figure 6. The CPU will activate the $\overline{AS}$ output with a LOW pulse to mark the beginning of a machine cycle, and ST0-ST3 outputs will reflect the refresh cycle code. The least significant 9 lines of the AD0-AD15 bus contain the refresh address. Because the memory is word organized, the AD0 will always be LOW. The most significant 7 bus lines are not specified. The $\overline{DS}$ output will remain HIGH for the entire cycle, while R/$\overline{W}$, B/$\overline{W}$, SN0-SN6 and N/$\overline{S}$ outputs will remain at the same level as in the machine cycle prior to refresh. The AD0-AD15 bus will go into high-impedance state during T2 period and remain there for the remainder of the cycle. The CPU will activate the $\overline{MREQ}$ output LOW during the refresh cycle. It should be noted that $\overline{WAIT}$ input is ignored by the CPU for refresh operations.

I/O Transactions:

There are two status line codes used for I/O transaction cycles. The CPU provides two separate I/O spaces and two types of instructions called Normal I/O and Special I/O. Each I/O space is addressed by a 16-bit address called port address. The timing for both types of I/O transactions is essentially identical. A typical I/O cycle consists of four clock periods T1, T2, TWA and T3 as shown in Figure 7. The TWA is the wait state; insertion of one wait state for an I/O cycle is always automatic. Additional wait cycles can be inserted by LOW on the $\overline{WAIT}$ input. The $\overline{WAIT}$ input is sampled during every TW state. If this input is LOW, one more wait state will be inserted. Insertion of wait states continues until $\overline{WAIT}$ input is HIGH. T3 state will follow the last wait state to complete the I/O cycle.

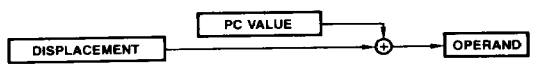
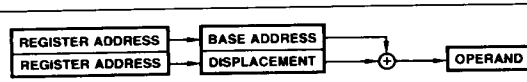
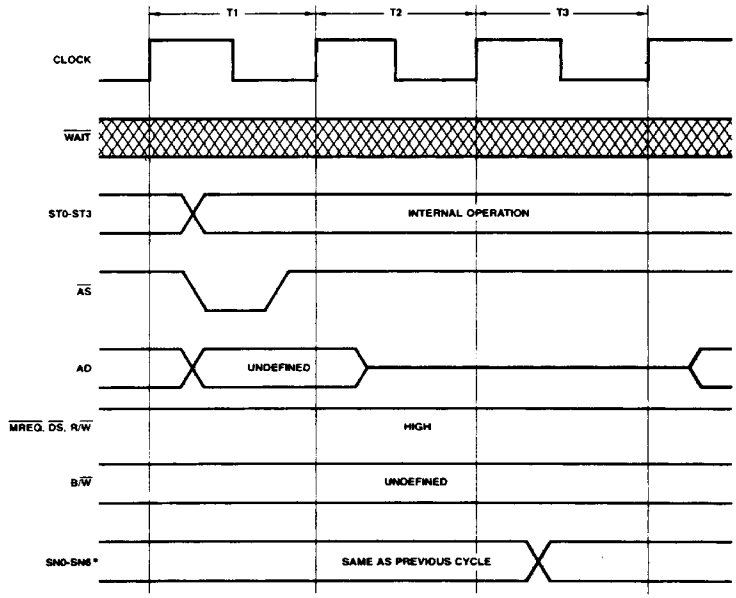
Mode	Operand Addressing			Operand Value
	In the Instruction	In a Register	In Memory	
Register				The content of the register.
Immediate				In the instruction
Indirect Register				The content of the location whose address is in the register.
Direct Address				The content of the locations whose address is in the instruction.
Index				The content of the location whose address is the address in the instruction, offset by the content of the working register.
Relative Address				The content of the locations whose address is the content of the program counter, offset by the displacement in the instruction.
Base Address				The content of the location whose address is the address in the register, offset by the displacement in the instruction.
Base Index				The content of the location whose address is the address in the register, offset by the displacement in the register.

Figure 4. Addressing Modes

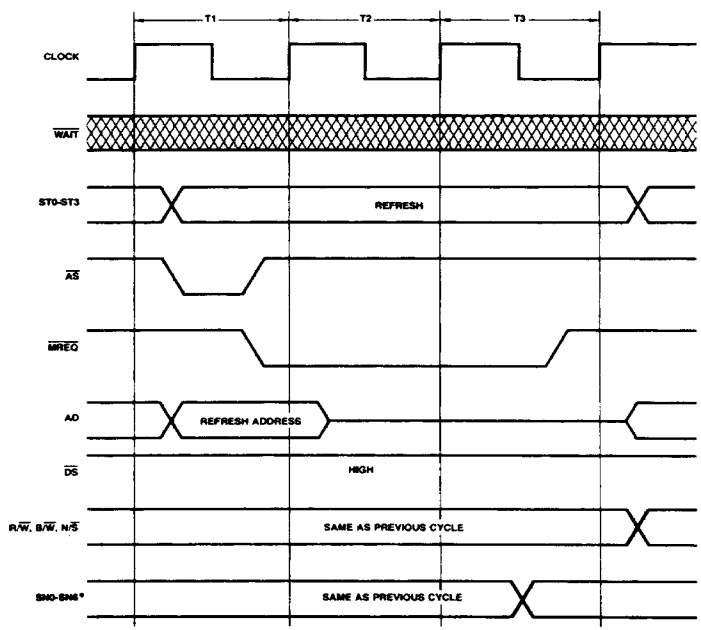
During I/O cycles the ST0-ST3 outputs will reflect the appropriate code depending on the type of instruction being executed (Normal I/O or Special I/O). $\overline{AS}$ output will be pulsed LOW to mark the beginning of the cycle. The CPU drives the AD0-AD15 bus with the 16-bit port address specified by the current instruction. The N/$\overline{S}$ output will be LOW indicating that the CPU is operating in the system mode. It should be recalled that the N/$\overline{S}$ output is derived from the appropriate bit in the FCW register. All I/O instructions are privileged instructions and will be allowed to execute only if the FCW specifies system mode operation. The $\overline{MREQ}$ output will be HIGH. The I/O instructions provide both word or byte transactions. The B/$\overline{W}$ output will be HIGH or LOW depending whether the instruction specifies a byte or word transfer. The SN0-SN6 output will remain at the same level as in the machine cycle prior to the I/O cycle.

Two kinds of I/O transfers should be considered: Data In means reading from the device, and Data Out means writing into the device. For In operations, the R/$\overline{W}$ output will be HIGH. The AD0-AD15 bus will go into high-impedance state during T2. During byte input instructions, the CPU reads either the even or odd half of the Data Bus dependent upon the port address. If the port address is even, the most significant half of the Data Bus is read. If the port address is odd, the least significant half of the Data Bus is read. During word input instructions, the CPU reads all 16 bits of the Data Bus. The CPU will drive the $\overline{DS}$ output LOW to signal to the device that data can be gated onto the bus. The CPU will accept the data during T3, and $\overline{DS}$ output will go HIGH signaling the end of an I/O transaction.



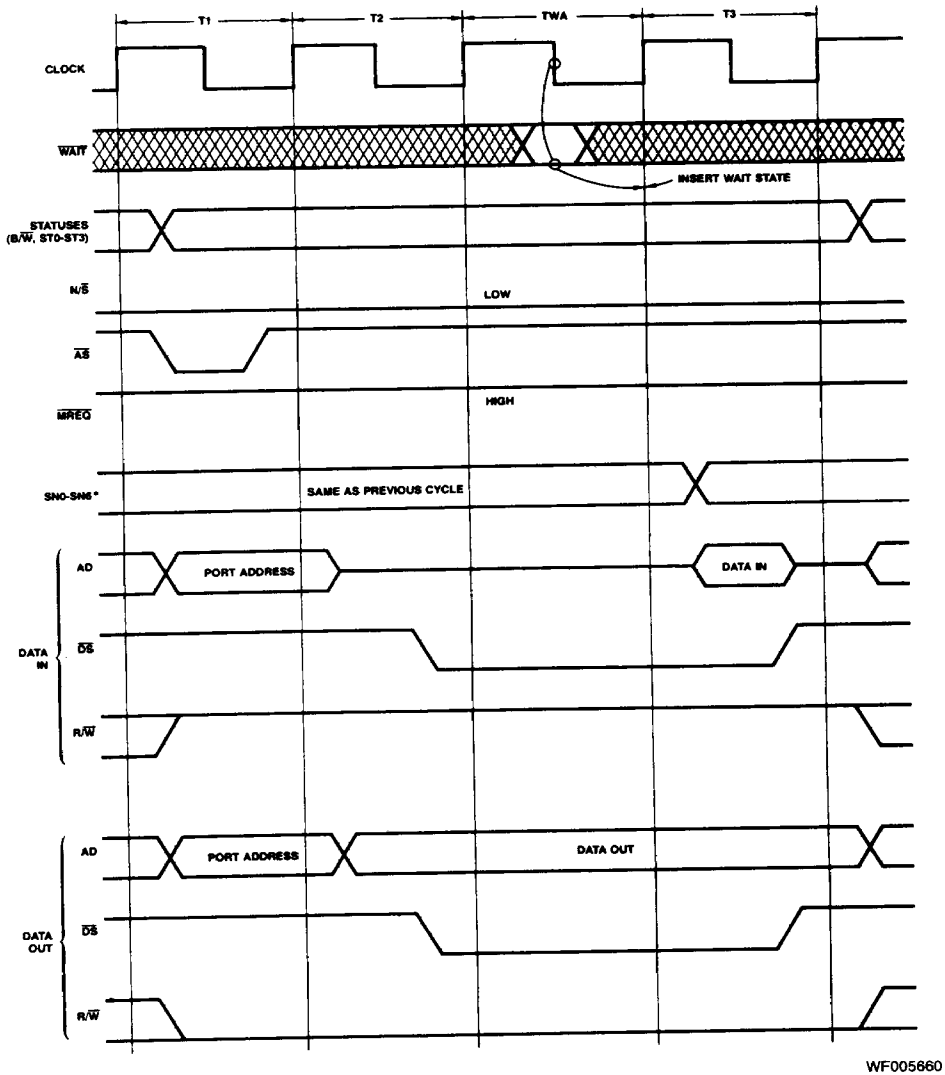
*Z8001 only

Figure 5. Internal Operation Cycle



*Z8001 only

Figure 6. Refresh Cycle



*Z8001 only

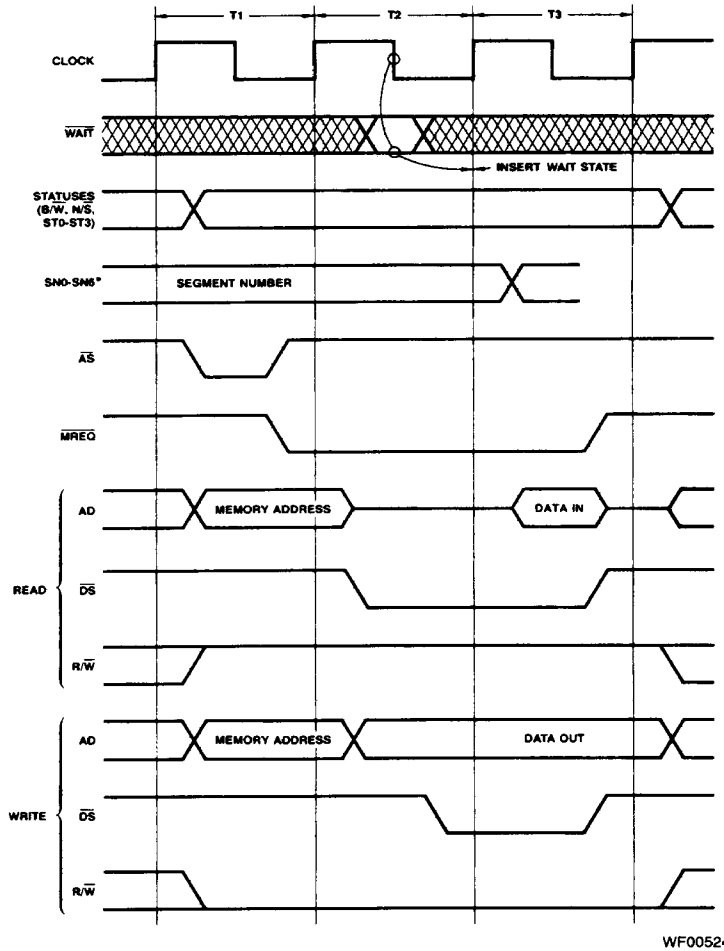
Figure 7. Z8001 I/O Cycle

For Data Out, the R/$\overline{W}$ output will be LOW. The CPU will provide data on the AD0-AD15 bus and activate the $\overline{DS}$ output LOW during T2. During byte output instructions, the CPU duplicates the byte data onto both the high and low halves of the Data Bus, and external logic, using A0, enables the appropriate byte port. During word output instructions the CPU outputs data onto all 16 bits of the Data Bus. The $\overline{DS}$ output goes HIGH during T3 and the cycle is complete.

Memory Transactions:

There are four status line codes that indicate a memory transaction:

a) Memory transaction to read or write an operand;
b) Memory transaction to read from or write into the stack;
c) Memory transaction to fetch the first word of an instruction (sometimes called IF1); and
d) Memory transaction to fetch the subsequent word of an instruction (sometimes called IFN).



*Z8001 only

Figure 8. Memory Transactions

It can be appreciated that all the above transactions essentially fall into two categories: memory read and memory write. In the case of IF1 and IFN cycles, the memory will be read at the address supplied by the program counter. All instructions are multiples of 16-bit words. Words are always addressed by an even address. Thus IF1 and IFN cycles involve performing a memory read for words. On the other hand, a memory transaction for operand and stack operation could be a read or write. Moreover, an operand could be a word or a byte. For stack operation involving the implied stack pointer, the address will be in the appropriate stack pointer register (R15, R15', RR14 or RR14'). For operand transactions, the memory address will come from several sources depending on the instruction and the addressing mode. Memory transaction cycle timing is shown in Figure 8. It typically consists of three clock periods: T1, T2 and T3. Wait states (TW) can be inserted between T2 and T3 by activating the $\overline{WAIT}$ input LOW. The $\overline{WAIT}$ input will be sampled during T2 and during every subsequent TW. The ST0-ST3 outputs will reflect the appropriate code for the current cycle early in T1, and the $\overline{AS}$ output will be pulsed LOW to mark the beginning of the cycle. The N/$\overline{S}$ output will indicate whether the normal or system address space will be used for the current cycle. As shown in the figure, the $\overline{MREQ}$ output will go LOW during T1 to indicate a memory operation.

The segment number becomes valid on the segment lines one clock period before the state of the memory operation and remains valid until the state of T3.

Consider a read operation first. The R/$\overline{W}$ output will be HIGH. The CPU will drive the AD0-AD15 with the appropriate address early in T1. During T2, the bus will go into high-impedance state, and $\overline{DS}$ output will be activated LOW by the CPU. The data can be gated onto the bus when $\overline{DS}$ is LOW. During T1 the B/$\overline{W}$ will also be activated to indicate which byte or word will be transacted. The memory is word organized, and words

are addressed by even addresses. However, when addressing bytes, the memory address may be odd or even, an even address for the most significant byte of a word and the next odd address for the least significant byte of that word. When reading a byte from the memory, the least significant address bit can be ignored and the whole word containing the desired byte is gated onto the bus. The CPU will pick the appropriate byte automatically and will drive the $\overline{DS}$ output HIGH indicating data acceptance.

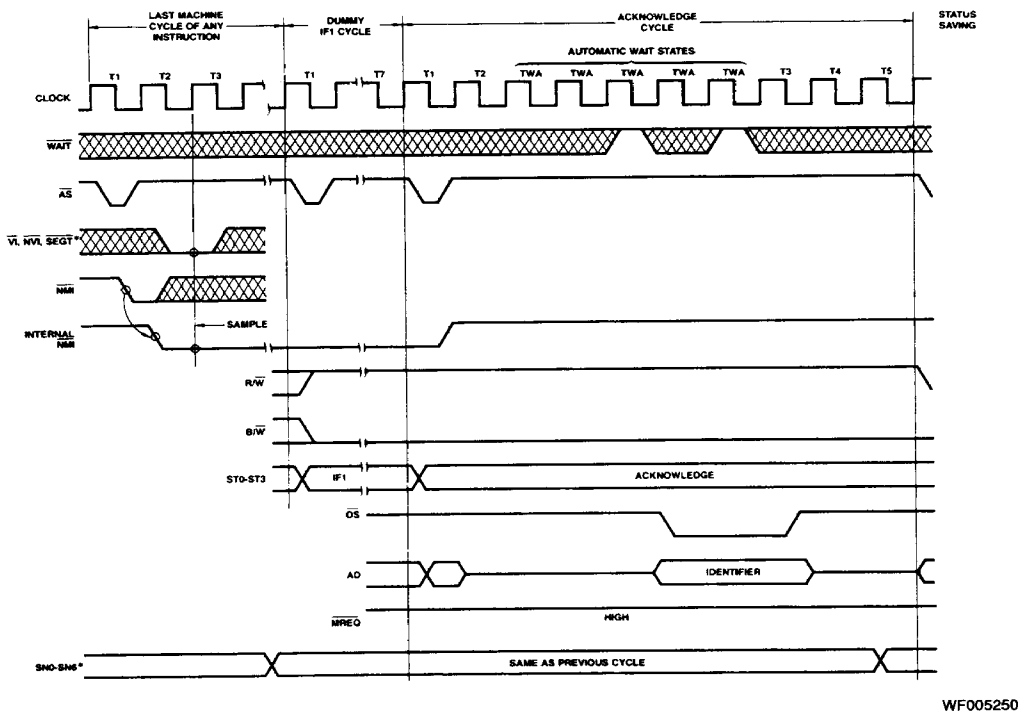
Consider the write operation next. The R/$\overline{W}$ output will be LOW. The CPU removes the address and gates out the data to be written on the bus and activates the $\overline{DS}$ output LOW during T2. If the data to be written is a byte, then the same byte will be on both halves of the bus. The $\overline{DS}$ output will go HIGH during T3, signifying completion of the cycle.

Interrupt and Segment Trap Acknowledge:

There are four status line codes devoted to interrupt and trap acknowledgement. These correspond to non-maskable, vectored and non-vectored interrupts, as well as segment trap. The Interrupt Acknowledge cycle is illustrated in Figure 9. The $\overline{NMI}$ input of the Z8001 is edge detected; i.e., a HIGH-to-LOW input level change is stored in an internal latch. Similar internal storage is not provided for the $\overline{VI}$, $\overline{NVI}$, and $\overline{SEGT}$ inputs. For $\overline{VI}$ and $\overline{NVI}$ inputs to cause an interruption, the corresponding interrupt enable bits in the FCW must be 1. For the following discussion, both the NVIE and VIE bits in the FCW are assumed to be 1.

As shown in the figure, the $\overline{VI}$, $\overline{NVI}$ and $\overline{SEGT}$ input and the internal $\overline{NMI}$ latch output are sampled during T3 of the last machine cycle of an instruction.

A LOW on these signals triggers the corresponding interrupt acknowledge sequence described below. The CPU executes a dummy IF1 cycle prior to entering the actual acknowledge cycle (see memory transactions for IF1 cycle description).



*Z8001 only

Figure 9. Interrupt Acknowledge Cycle

During this dummy IF1 cycle, the program counter is not updated; instead, the implied-system stack pointer (RR14') will be decremented. Following the dummy IF1 cycle is the actual interrupt/trap acknowledge cycle.

The interrupt acknowledge cycle typically consists of 10 clock periods: T1 through T5 and five automatic TW (wait states). As usual, the $\overline{AS}$ output will be pulsed LOW during T1 to mark the beginning of a cycle. The ST0-ST3 outputs will reflect the appropriate interrupt acknowledge code; the $\overline{MREQ}$ output will be HIGH; the N/$\overline{S}$ output remains the same as in the preceding cycle; the R/$\overline{W}$ output will be HIGH; and the B/$\overline{W}$ output will be LOW. The CPU will drive the AD0-AD15 bus with

unspecified information during T1, and the bus will go into the high-impedance state during T2. Three TWA states will automatically follow T2. The $\overline{WAIT}$ input will be sampled during the third TWA state.

If LOW, an extra TW state will be inserted and the $\overline{WAIT}$ will be sampled again during TW. Such insertion of TW states continues until the $\overline{WAIT}$ input is HIGH. After the last TW state, the $\overline{DS}$ output will go LOW and two more automatic wait states (TWA) follow. The interrupting device can gate up to a 16-bit identifier onto the bus when the $\overline{DS}$ output is LOW. The $\overline{WAIT}$ input will be sampled again during the last TWA state. If the $\overline{WAIT}$ input is LOW, one TW state will be inserted, and the $\overline{WAIT}$ will be sampled during TW. Such TW insertion continues until the $\overline{WAIT}$ input is HIGH. After completing the last TW state, T3 will be entered, and the $\overline{DS}$ output will go HIGH. The interrupting device should remove the identifier and cease driving the bus. T4 and T5 states will follow T3 to complete the cycle. Following the interrupt acknowledge cycle will be memory transaction cycles to save the status on the stack. Note that the N/$\overline{S}$ output will be automatically LOW during status saving. The SN0-SN6 outputs are undefined during the acknowledge cycle.

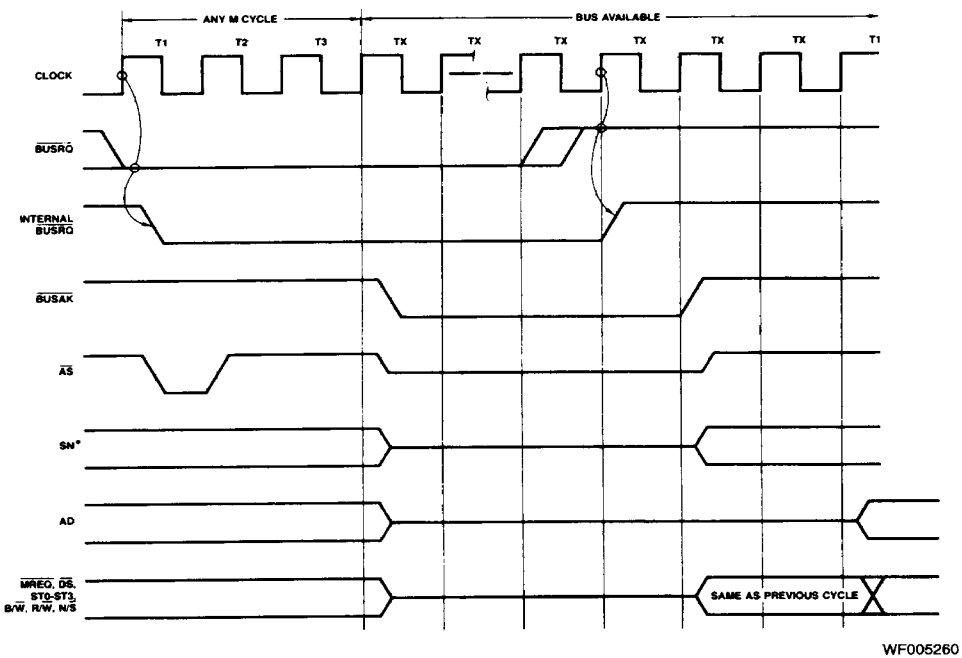
The internal $\overline{NMI}$ latch will be reset to the initial state as $\overline{AS}$ going HIGH in the interrupt acknowledge cycle. The $\overline{VI}$, $\overline{NVI}$ and $\overline{SEGT}$ input should be kept LOW until this time also.

Status Saving Sequence:

The machine cycles following the interrupt acknowledge cycle push the old status information on the system stack in the following order: program counter, the flag and control word, and the interrupt/trap identifier. Subsequent machine cycles fetch the new program status from the new program status area and then branch to the interrupt/service routine.

Bus Request/Bus Acknowledge Timing:

A LOW on the $\overline{BUSRQ}$ input is an indication to the CPU that another device (such as DMA) is requesting control of the bus. The $\overline{BUSRQ}$ input is synchronized internally at T1 of any machine cycle. (See next paragraph for exception.) The $\overline{BUSAK}$ will go LOW after the last clock period of the machine cycle. The LOW on the $\overline{BUSAK}$ output indicates acknowledgement. When $\overline{BUSAK}$ is LOW, the following outputs will go into the high-impedance state: AD0-AD15, $\overline{AS}$, $\overline{DS}$, $\overline{MREQ}$, ST0-ST3, B/$\overline{W}$, R/$\overline{W}$, SN0-SN6 and N/S. The $\overline{BUSRQ}$ must be held LOW until all transactions are completed. When $\overline{BUSRQ}$ goes HIGH, it is synchronized internally; the $\overline{BUSAK}$ output will go HIGH, and normal CPU operation will resume. Figure 10 illustrates the $\overline{BUSRQ}$/$\overline{BUSAK}$ timing.



*Z8001 only

Figure 10. Bus Request/Acknowledge Cycle

It was mentioned that $\overline{BUSRQ}$ will be honored during any machine cycle with one exception. This exception is during the execution of TSET/TSETB instructions. $\overline{BUSRQ}$ will not be honored once execution of these instructions has started.

Single Stepping

The $\overline{STOP}$ input of the CPU facilitates one instruction at a time or single step operation. Figure 11 illustrates $\overline{STOP}$ input timing. The $\overline{STOP}$ input is sampled on the HIGH-to-LOW transition of the clock input that immediately precedes an IF1 cycle. If the $\overline{STOP}$ is found LOW, Z8001 introduces a memory refresh cycle after T3. Moreover, $\overline{STOP}$ input will be sampled again at T3 in the refresh cycle. If $\overline{STOP}$ is LOW, one more refresh cycle will follow the previous refresh cycle. The $\overline{STOP}$ will be sampled during T3 of the refresh cycle, also. One additional refresh cycle will be added every time $\overline{STOP}$ input is sampled LOW. After completing the last refresh cycle which will occur after $\overline{STOP}$ is HIGH, the CPU will insert two dummy states, T4 and T5, to complete the IF1 cycle and resume its normal operations for executing the instruction. See appropriate sections on memory transactions and memory refresh. It should be noted that refresh cycles will occur even if the refresh facility is disabled during single stepping.

Multimicroprocessor Facilities

The CPU is provided with hardware and software facilities to support multiple microprocessor systems. The $\overline{\mu O}$ and $\overline{\mu I}$ signals of the CPU are used in conjunction with the MBIT, $\overline{MREQ}$, MRES and MSET instructions for this purpose. The $\overline{\mu O}$ output can be activated LOW by using appropriate instruction to signal a request from the CPU for a resource. The $\overline{\mu I}$ input is tested by the CPU before activating the $\overline{\mu O}$ output. LOW at the $\overline{\mu I}$ input indicates that the resource is busy. The CPU can examine the $\overline{\mu I}$ input after activating the $\overline{\mu O}$ output LOW. The $\overline{\mu I}$ will be tested again to see if the requested resource became available.

Initialization

A LOW on the $\overline{Reset}$ input starts the CPU initialization. The initialization sequence is shown in Figure 12. Within five clock periods after the HIGH-to-LOW level change of the $\overline{Reset}$ input, the following will occur:

a) AD0-AD15 bus will be in the high-impedance state.

b) $\overline{AS}$, $\overline{DS}$, $\overline{MREQ}$, $\overline{BUSAK}$ and $\overline{\mu O}$ outputs will be HIGH.

c) ST0-ST3 outputs will be LOW.

d) Refresh will be disabled.

e) R/$\overline{W}$, B/$\overline{W}$ and N/$\overline{S}$ outputs are not affected. For a power-on reset, the state of these outputs is not specified.

f) SN0-SN6 outputs will be LOW.

After the $\overline{Reset}$ input returns HIGH and remains HIGH for three clock periods, two (three for the Z8001) 16-bit memory read operations will be performed as follows. Note that the N/$\overline{S}$ output will be LOW, and ST0-ST3 outputs will reflect IFN code.

a) The contents of the memory location 0002 (segment 0) will be read. This information will be loaded into the FCW of the CPU.
b) The contents of the memory location 0004 (segment 0) will be read. This information will be loaded into the program counter segment number.
c) (Z8001 only.) The contents of the memory location 0006 (segment 0) will be read. This information will be loaded into the program counter offset.

This completes initialization sequence, and an IF1 cycle will follow to fetch the first instruction to begin program execution. See the section on memory transactions for timing.

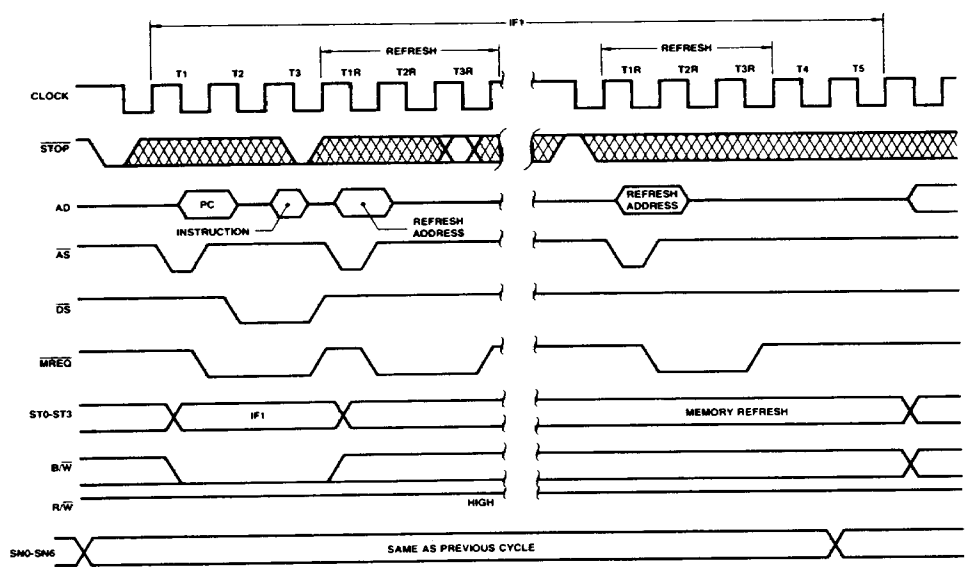
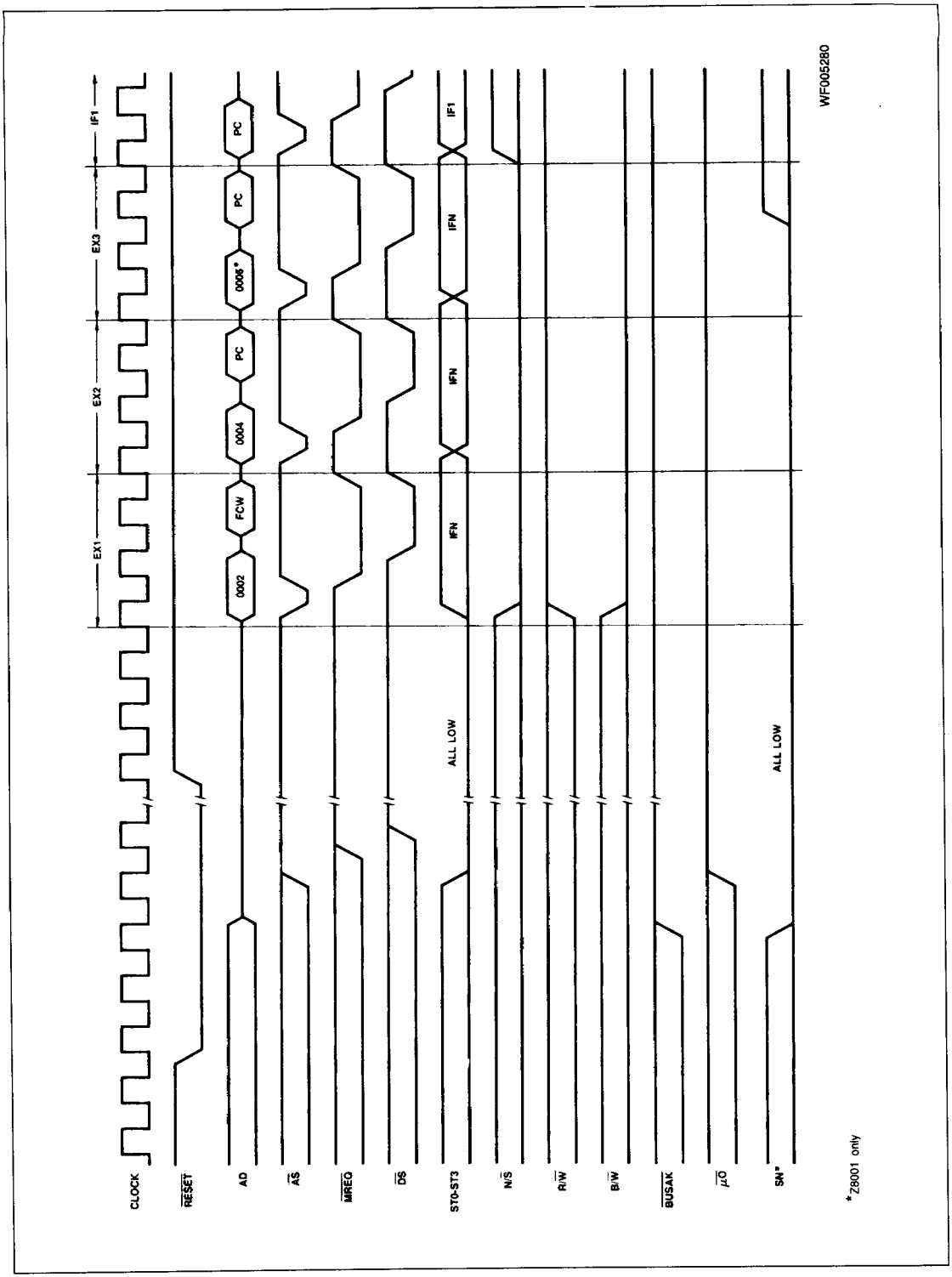


Figure 11. Single Step Timing



Z8001/2 CPU INSTRUCTION SET

ARITHMETIC

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
	ADC ADCB	R, src	R	5	Add with Carry R ← R + src + carry
Add	ADD ADDB ADDL	R, src	R IM IR DA X	4 7 7 9 10	Add R ← R + src
Compare	CP CPB CPL	R, src	R IM IR X	4 7 7 10	Compare with Register R − src
	CP CPB	dst, IM	IR DA X	11 14 15	Compare with Immediate dst − IM
	DAB	dst	R	5	Decimal Adjust
Decrement	DEC DECB	dst, n	R IR DA X	4 11 13 14	Decrement by n dst ← dst − n (n = 1 . . . 16)
Divide	DIV DIVL	R, src	R IM IR DA X	107 107 107 108 109	Divide (signed) Word: $R_n + 1 \leftarrow R_{n,n+1} \div src$ $R_n \leftarrow$ remainder Long Word: $R_n + 2,n+3$ $\leftarrow R_{n...n+3} \div src$ $R_{n,n+1}$ ← remainder
Extend	EXTS EXTSB EXTSL	dst	R	11	Extend Sign Extend sign of low order half of st through high order half of dst
Increment	INC INCB	dst, n	R IR DA X	4 11 13 14	Increment by n dst ← dst + n (n = 1 . . . 16)
Multiply	MULT MULTL	R, src	R IM IR DA X	70 70 70 71 72	Multiply (signed) Word: $R_{n,n+1} \leftarrow R_n + 1 \bullet src$ Long Word: $R_{n...n+3}$ $\leftarrow R_n + 2, + 3 \bullet src$ *Plus seven cycles for each 1 in the multiplicand
Negate	NEG NEGB	dst	R R DA X	7 12 15 16	Negate dst ← 0 − dst
	SBC SBCB	R, src	R	5	Subtract with Carry R ← R − src − carry
Subtract	SUB SUBB SUBL	R, src	R IM IR DA X	4 7 7 9 10	Subtract R ← R − src

LOGICAL

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
AND	AND ANDB	R, scr	R IM IR DA X	4 7 7 9 10	AND R ← R AND src
Complement	COM COMB	dst	R IR DA X	7 12 15 16	Complement dst ← NOT dst
OR	OR ORB	R, src	R IM IR DA X	4 7 7 9 10	OR R ← R OR src
Test	TEST TESTB TESTL	dst	R IR DA X	7 8 11 12	TEST dst OR 0
	TCC TCCB	cc, dst	R	5	Test Condition Code Set LSB if cc is true
XOR	XOR XORB	R, src	R IM IR DA X	4 7 7 9 10	Exclusive OR R ← R XOR src

†Clock cycles for byte or word data, non-segmented addresses. Segmented addresses may require 2 to 4 additional cycles. Some long word data require more cycles.

LOAD AND EXCHANGE

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Clear	CLR CLRB	dst	R IR DA X	7 8 11 12	Clear dst←0
Exchange	EX EXB	R, src	R IR DA X	6 12 15 16	Exchange R↔src
Load	LD LDB LDL	R, src	R IM IR DA X RA BA BX	3 7 7 9 10 14 14 14	Load into Register R←src
	LD LDB LDL	dst, R	IR DA X RA BA BX	8 11 12 14 14 14	Load into Memory (Store) dst←R
	LD LDB	dst, IM	R IR DA X	5 11 14 15	Load Immediate into Memory dst←IM
	LDA	R, src	DA X RA BA BX	12 13 15 15 15	Load Address R←source address
	LDAR	R, src	RA	15	Load Address Relative R←source address
	LDK	R, src	IM	5	Load Constant R←n (n = 0...15)
	LDM	R, src, n	IR DA X	11 + 3n 14 + 3n 15 + 3n	Load Multiple R←src (n consecutive words] (n = 1...16)
	LDM	dst, R, n	IR DA X	11 + 3n 14 + 3n 15 + 3n	Load Multiple (Store Multiple) dst←R (n consecutive words) (n = 1...16)
	LDR LDRB LDRL	R, src	RA	14	Load Relative R←src (range −32768... + 32767)
	LDR LDRB LDRL	dst, R	RA	14	Load Relative (Store Relative) dst←R (range −32768... + 32767)
Pop	POP POPL	dst, R	R IR DA X	8 12 16 16	Pop dst←IR Autoincrement contents of R
Push	PUSH PUSHL	IR, src	R IM IR DA X	9 12 13 14 14	Push Autodecrement contents of R IR←src

BIT MANIPULATION

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Test	BIT BITB	dst, b	R IR DA X	4 8 10 11	Test Bit Static Z flag←NOT dst bit specified by b
	BIT BITB	dst, R	R	10	Test Bit Dynamic Z flag←NOT dst bit specified by contents of R
Reset	RES RESB	dst, b	R IR DA X	4 11 13 14	Reset Bit Static Reset dst bit specified by b
	RES RESB	dst, R	R	10	Reset Bit Dynamic Reset dst bit specified by contents of R
Set	SET SETB	dst, b	R IR DA X	4 11 13 14	Set Bit Static Set dst bit specified by b
	SET SETB	dst, R	R	10	Set Bit Dynamic Set dst bit specified by contents of R
Test and Set	TSET TSETB	dst	R IR DA X	7 11 14 15	Test and Set S flag←MSB of dst dst←all 1s

ROTATE AND SHIFT

	Mnemonics	Operand	Addr. Modes	Clock Cycles†	Operation
Rotate	RLDB	R, src	R	9	Rotate Digit Left
	RRDB	R, src	R	9	Rotate Digit Right
	RL RLB	dst, n	R R	6	Rotate Left by n bits (n = 1, 2)
	RLC RLCB	dst, n	R R	6	Rotate Left through Carry by n bits (n = 1, 2)
	RR RRB	dst, n	R R	6	Rotate Right by n bits (n = 1, 2)
	RRC RRCB	dst, n	R R	6	Rotate Right through Carry by n bits (n = 1, 2)
Shift	SDA SDAB SDAL	dst, R	R	15 + 3n	Shift Dynamic Arithmetic Shift dst left or right by contents of R
	SDL SDLB SDLL	dst, R	R	15 + 3n	Shift Dynamic Logical Shift dst left or right by contents of R
	SLA SLAB SLAL	dst, n	R	13 + 3n	Shift Left Arithmetic by n bits
	SLL SLLB SLLL	dst, n	R	13 + 3n	Shift Left Logical by n bits
	SRA SRAB SRAL	dst, n	R	13 + 3n	Shift Right Arithmetic by n bits
	SRL SRLB SRLL	dst, n	R	13 + 3n	Shift Right Logical by n bits

†Clock cycles for byte or word data, non-segmented addresses. Segmented addresses may require 2 to 4 additional cycles. Some long word data require more cycles.

BLOCK TRANSFER AND STRING MANIPULATION (AUTO INCREMENT/DECREMENT AND REPEAT)

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Compare	CPD CPDB	R_X, src, R_Y, cc	IR	20	Compare and Decrement R_X – src Autodecrement src address $R_Y \leftarrow R_Y - 1$
	CPDR CPDRB	R_X, src, R_Y, cc	IR	11 + 9n	Compare, Decrement and Repeat R_X – src Autodecrement src address $R_Y \leftarrow R_Y - 1$ Repeat until cc is true or $R_Y = 0$
	CPI CPIB	R_X, src, R_Y, cc	IR	20	Compare and Increment R_X – src Autoincrement src address $R_Y \leftarrow R_Y - 1$
	CPIR CPIRB	R_X, src, R_Y, cc	IR	11 + 9n	Compare, Increment and Repeat R_X – src Autoincrement src address $R_Y \leftarrow R_Y - 1$ Repeat until cc is true or $R_Y = 0$
	CPSD CPSDB	dst, src, R, cc	IR	25	Compare String and Decrement dst – src Autodecrement dst and src addresses R ← R – 1
	CPSDR CPSDRB	dst, src, R, cc	IR	11 + 14n	Compare String, Decr. and Repeat dst – src Autodecrement dst and src addresses R ← R – 1 Repeat until cc is true or R = 0
	CPSI CPSIB	dst, src, R, cc	IR	25	Compare String and Increment dst – src Autoincrement dst and src addresses R ← R – 1
	CPSIR CPSIRB	dst, src, R, cc	IR	11 + 14n	Compare String, incr. and Repeat dst – src Autoincrement dst and src addresses R ← R – 1 Repeat until cc is true or R = 0
Load	LDD LDDB	dst, src, R	IR	20	Load and Decrement dst ← src Autodecrement dst and src addresses R ← R – 1
	LDDR LDDRB	dst, src, R	IR	11 + 9n	Load, Decrement and Repeat dst ← src Autodecrement dst and src addresses R ← R – 1 Repeat until R = 0

BLOCK TRANSFER AND STRING MANIPULATION (Cont.)

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Load	LDI LDIB	dst, src, R	IR	20	Load and Increment dst ← src Autoincrement dst and src addresses R ← R – 1
	LDIR LDIRB	dst, src, R	IR	11 + 9n	Load, Increment and Repeat dst ← src Autoincrement dst and src addresses R ← R – 1 Repeat until R = 0
Translate	TRDB	dst, src, R	IR	25	Translate and Decrement dst ← src (dst) Autodecrement dst address R ← R – 1
	TRDRB	dst, src, R	IR	11 + 14n	Translate, Decrement and Repeat dst ← src (dst) Autodecrement dst address R ← R – 1 Repeat until R = 0
	TRIB	dst, src, R	IR	25	Translate and Increment dst ← src (dst) Autoincrement dst address R ← R – 1 0
	TRIRB	dst, src, R	IR	11 + 14n	Translate, Increment and Repeat dst ← src (dst) Autoincrement dst address R ← R – 1 Repeat until R = 0
Translate and Test	TRTDB	src 1, src 2, R	IR	25	Translate and Test, Decrement RH1 ← src 2 (src 1) Autodecrement src 1 address R ← R – 1
	TRTDRB	src 1, src 2, R	IR	11 + 14n	Translate and Test, Decrement and Repeat RH1 ← src 2 (src 1) Autodecrement src 1 address R ← R – 1 Repeat until R = 0 or RH1 = 0
	TRTIB	src 1, src 2, R	IR	25	Translate and Test, Increment RH1 ← src 2 (src 1) Autoincrement src 1 address R ← R – 1
	TRTIRB	src 1, src 2, R	IR	11 + 14n	Translate and Test, Increment and Repeat RH1 ← src 2 (src 1) Autoincrement src 1 address R ← R – 1 Repeat until R = 0 or RH1 = 0

†Clock cycles for byte or word data, non-segmented addresses. Segmented addresses may require 2 to 4 additional cycles. Some long word data require more cycles.

INPUT/OUTPUT

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Input	IN* INB*	R, src	IR DA	10 12	Input R ← src
	IND* INDB*	dst, src, R	IR	21	Input and Decrement dst ← src Autodecrement dst address R ← R − 1
	INDR* INDRB*	dst, src, R	IR	11 + 10n	Input, Decrement and Repeat dst ← src Autodecrement dst address R ← R − 1 Repeat until R = 0
	INI* INIB*	dst, src, R	IR	21	Input and Increment dst ← src Autoincrement dst address R ← R − 1
	INIR* INIRB*	dst, src, R	IR	11 + 10n	Input, Increment and Repeat dst ← src Autoincrement dst address R ← R − 1 Repeat until R = 0
Output	OUT* OUTB*	dst, R	IR DA	10 12	Output dst ← R
	OUTD* OUTDB*	dst, src, R	IR	21	Output and Decrement dst ← src Autodecrement src address R ← R − 1
	OTDR* OTDRB*	dst, src, R	IR	11 + 10n	Output and Decrement dst ← src Autodecrement src address R ← R − 1 Repeat until R = 0
	OUTI* OUTIB*	dst, src, R	IR	21	Output and Increment dst ← src Autoincrement src address R ← R − 1
	OTIR* OTIRB*	dst, src, R	IR	11 + 10n	Output, Increment and Repeat dst ← src Autoincrement src address R ← R − 1 Repeat until R = 0
Special Input (identical to input but different status code)	SIN* SINB*	R, src	DA	12	Special Input R ← src
	SIND* SINDB*	dst, src, R	IR	21	Special Input and Decrement dst ← src Autodecrement dst address R ← R − 1
	SINDR* SINDRB*	dst, src, R	IR	11 + 10n	Special Input, Decr. and Repeat dst ← src Autodecrement dst address R ← R − 1 Repeat until R = 0
	SINI* SINIB*	dst, src, R	IR	21	Special Input and Increment dst ← src Autoincrement dst address R ← R − 1
	SINIR* SINIRB*	dst, src, R	IR	11 + 10n	Special Input, Incr. and Repeat dst ← src Autoincrement dst address R ← R − 1 Repeat until R = 0

INPUT/OUTPUT (Cont.)

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Special Output (identical to output, but different status code)	SOUT* SOUTB*	dst, src	DA	12	Special Output dst ← src
	SOUTD* SOUTDB*	dst, src R	IR	21	Special Output and Decrement dst ← src Autodecrement src address R ← R − 1
	SOTDR* SOTDRB*	dst, src, R	IR	11 + 10n	Special Output, Decr. and Repeat dst ← src Autodecrement src address R ← R − 1 Repeat until R = 0
	SOUTI* SOUTIB*	dst, src R	IR	21	Special Output and Increment dst ← src Autoincrement src address R ← R − 1
	SOTIR* SOTIRB*	dst, src R	R	11 + 10n	Special Output, Incr. and Repeat dst ← src Autoincrement src address R ← R − 1 Repeat until R = 0

CPU CONTROL

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Interrupts	DI*	int	–	7	Disable Interrupt (Any combination of NVI, VI)
	EI*	int	–	7	Enable Interrupt (Any combination of NVI, VI)
Halt	HALT*	–	–	6 + 3n	HALT
Control Words	LDCTL*	CTLR, src	R	7	Load into Control Register CTLR ← src
	LDCTL*	dst, CTLR	R	7	Load into Control Register dst ← CTLR
	LDCTLB	dst, FLGR	R	7	Load into Flag Byte Register FLGR ← src
	LDPS*	src	IR DA X	12 16 17	Load Program Status PS ← src
Multi Micro	MBIT*	–	–	7	Test Multi-Micro Bit Set S if μI is HIGH; reset S if μI is LOW
	MREQ*	dst	R	12 + 7n	Multi-Micro Request
	MRES*	–	–	5	Multi-Micro Reset
	MSET*	–		5	Multi-Micro Set
NOP	NOP			7	No Operation
Flags	RESFLG	flag		7 7	Reset Flag (Any combination of C, Z, S, P/V)
	SETFLG	flag		7	Set Flag (Any combination of C, Z, S, P/V)
	COMFLG	flags	–	7	Complement Flag (Any combination of C, Z, S, P/V)

*Privileged instructions. Executed in system mode only.

†Clock cycles for byte or word data, non-segmented addresses. Segmented addresses may require 2 to 4 additional cycles. Some long word data require more cycles.

PROGRAM CONTROL

	Mnemonics	Operands	Addr. Modes	Clock Cycles†	Operation
Call	CALL	dst	IR DA X	10 12 13	Call Subroutine Autodecrement SP @ SP ← PC PC ← dst
	CALR	dst	RA	10	Call Relative Autodecrement SP @SP ← PC PC ← PC + dst (range −4094 to +4096)
	SC	src	IM	33	System Call Autodecrement SP @ SP ← old PS Push Instruction PS ← System Call PS
Jump	DJNZ DBJNZ	R, dst	RA	11	Decrement and Jump if Non-Zero R ← R − 1 IF R = 0: PC ← PC + dst (range −254 to 0)
	IRET*	–	–	13	Interrupt Return PS ← @SP Autoincrement SP
	JP	cc, dst	IR DA X	7 – 7 8	Jump Conditional If cc is true: PC ← dst
	JR	cc, dst	RA	8	Jump Conditional Relative If cc is true: PC ← PC + dst (range −256 to +254)
	RET	cc	–	10	Return Conditional If cc is true: PC ← @SP Autodecrement SP
Return	IRET*	–	–	13	Interrupt Return PS ← @SP Autoincrement SP

*Privileged instructions. Executed in system mode only.

†Clock cycles for byte or word data, non-segmented addresses. Segmented addresses may require 2 to 4 additional cycles. Some long word data require more cycles.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Voltage at any Pin
Relative to V_{SS} ..... −0.3 to +7.0V
Power Dissipation 2.5W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... 5 V ± 5%

Operating ranges define those limits between which the functionality of the device is guaranteed.

4, 6MHz Devices
DC CHARACTERISTICS over operating range unless otherwise specified

Parameters	Description	Test Conditions	Min	Max	Units
V_{CH}	Clock Input High Voltage	Driven by External Clock Generator	$V_{CC}-0.4$	$V_{CC}+0.3$	Volts
V_{CL}	Clock Input Low Voltage	Driven by External Clock Generator	−0.3	0.45	Volts
V_{IH}	Input High Voltage		2.0	$V_{CC}+0.3$	Volts
V_{IH} $\overline{NMI}$, Reset	Input High Voltage		2.4	$V_{CC}+0.3$	Volts
V_{IL}	Input Low Voltage		−0.3	0.8	Volts
V_{OH}	Output High Voltage	$I_{OH} = -250\mu A$	2.4		Volts
V_{OL}	Output Low Voltage	$I_{OL} = +2.0mA$		0.4	Volts
I_{IL}	Input Leakage Except $\overline{SEGT}$ Pin	$0.4 \leq V_{IN} \leq +2.4V$		±10	μA
I_{IL} on $\overline{SEGT}$	Input Leakage on $\overline{SEGT}$ Pin		−100	100	μA
I_{OL}	Output Leakage	$0.4 \leq V_{OUT} \leq +2.4V$		±10	μA
ICC	V_{CC} Supply Current	Commercial		300	mA

8MHz Devices
DC CHARACTERISTICS over operating range unless otherwise specified

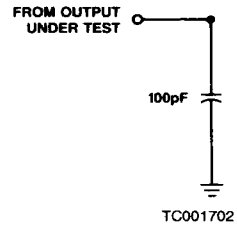
Parameters	Description	Test Conditions	Min	Max	Units
V_{CH}	Clock Input High Voltage	Driven by External Clock Generator	$V_{CC}-0.4$	$V_{CC}+0.3$	Volts
V_{CL}	Clock Input Low Voltage	Driven by External Clock Generator	−0.3	0.45	Volts
V_{IH}	Input High Voltage		2.0	$V_{CC}+0.3$	Volts
V_{IH} $\overline{NMI}$, Reset	Input High Voltage		2.4	$V_{CC}+0.3$	Volts
V_{IL}	Input Low Voltage		−0.3	0.8	Volts
V_{OH}	Output High Voltage	$I_{OH} = -250\mu A$	2.4		Volts
V_{OL}	Output Low Voltage	$I_{OL} = +2.0mA$		0.4	Volts
I_{IL}	Input Leakage Except $\overline{SEGT}$ Pin	$0.4 \leq V_{IN} \leq +2.4V$		±10	μA
I_{IL} on $\overline{SEGT}$	I_{IL} on $\overline{SEGT}$ Pin	$0.4 \leq V_{IN} \leq 2.4$ V	−100	100	μA
I_{OL}	Output Leakage	$0.4 \leq V_{OUT} \leq +2.4V$		±10	μA
I_{CC}	V_{CC} Supply Current	Commercial		400	mA

See section 6 for Thermal Characteristics Information.

Standard Test Conditions

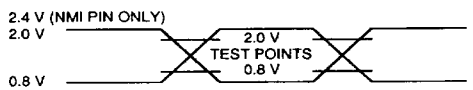
The characteristics below apply for the following test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

SWITCHING TEST CIRCUIT



TC001702

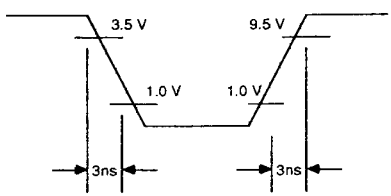
SWITCHING TEST WAVEFORMS
Input/Output



LS003010

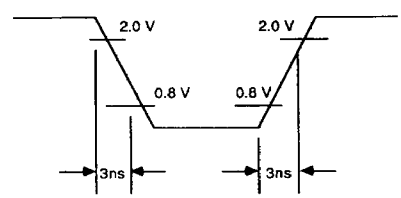
AC testing outputs are driven at 2.0 V for a logical 1 and 0.5 V for a logical 0. The clock is driven at V_{CC} −0.4 V and 0.45 V. Timing measurements are made at 2.0 V for a logical 1 and 0.5 V for a logical 0.

AC Clock Input



WF024270

AC Input (Except Clock)



WF024280

SWITCHING CHARACTERISTICS over operating ranges unless otherwise specified

Number	Parameters	Description	4MHz Devices		6MHz Devices		Units
			Min	Max	Min	Max	
1	TcC	Clock Cycle Time	250	2000	165	2000	ns
2	TwCh	Clock Width (HIGH)	105	1895	70	1930	ns
3	TwCl	Clock Width (LOW)	105	1895	70	1930	ns
4	TfC	Clock Fall Time		20		10	ns
5	TrC	Clock Rise Time		20		15	ns
6†	TdC(SNv)	Clock ↑ to Segment Number Valid (50pF Load)		130		110	ns
7†	TdC(SNn)	Clock ↑ to Segment Number Not Valid	20		10		ns
8	TdC(Bz)	Clock ↑ to Bus Float		65		55	ns
9	TdC(A)	Clock ↑ to Address Valid		100		75	ns
10	TdC(Az)	Clock ↑ to Address Float		65		55	ns
11	TdA(DR)*	Address Valid to Read Data Required Valid		475		305	ns
12	TsDI(C)	Data In to Clock ↓ Set-up Time	30		20		ns
13	TdDS(A)*	$\overline{DS}$ ↑ to Address Active	80		45		ns
14	TdC(DW)	Clock ↑ to Write Data Valid		100		75	ns
15	ThDI(DS)	Data In to $\overline{DS}$ ↑ Hold Time	0		0		ns
16	TdDO(DS)*	Data Out Valid to $\overline{DS}$ ↑ Delay	295		195		ns
17	TdA(MR)*	Address Valid to $\overline{MREQ}$ ↓ Delay	55		35		ns
18	TdC(MR)	Clock ↓ to $\overline{MREQ}$ ↓ Delay		80		70	ns
19	TwMRh*	$\overline{MREQ}$ Width (HIGH)	210		135		ns
20	TdMR(A)*	$\overline{MREQ}$ ↓ to Address Not Active	70		35		ns
21	TdDO(DSW)*	Data Out Valid to $\overline{DS}$ ↓ (Write) Delay	55		35		ns
22	TdMR(DR)*	$\overline{MREQ}$ ↓ to Read Data Required Valid		370		230	ns
23	TdC(MR)	Clock ↓ to $\overline{MREQ}$ ↑ Delay		80		60	ns
24	TdC(ASf)	Clock ↑ to $\overline{AS}$ ↓ Delay		80		60	ns
25	TdA(AS)*	Address Valid to $\overline{AS}$ ↑ Delay	55		35		ns
26	TdC(ASr)	Clock ↓ to $\overline{AS}$ ↑ Delay		90		80	ns
27	TdAS(DR)*	$\overline{AS}$ ↑ to Read Data Required Valid		360		220	ns
28	TdDS(AS)*	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay	70		35		ns
29	TwAS*	$\overline{AS}$ Width (LOW)	85		55		ns
30	TdAS(A)*	$\overline{AS}$ ↑ to Address Not Active Delay	70		45		ns
31	TdAz(DSR)	Address Float to $\overline{DS}$ (Read) ↓ Delay	0		0		ns
32	TdAS(DSR)*	$\overline{AS}$ ↑ to $\overline{DS}$ (Read) ↓ Delay	80		55		ns
33	TdDSR(DR)*	$\overline{DS}$ (Read) ↓ to Read Data Required Valid		205		130	ns
34	TdC(DSr)	Clock ↓ to $\overline{DS}$ ↑ Delay		70		65	ns
35	TdDS(DW)*	$\overline{DS}$ ↑ to Write Data and STATUS Not Valid	75		45		ns
36	TdA(DSR)*	Address Valid to $\overline{DS}$ (Read) ↓ Delay	180		110		ns
37	TdC(DSR)	Clock ↑ to $\overline{DS}$ (Read) ↓ Delay		120		85	ns
38	TwDSR*	$\overline{DS}$ (Read) Width (LOW)	275		185		ns
39	TdC(DSW)	Clock ↓ to $\overline{DS}$ (Write) ↓ Delay		95		80	ns
40	TwDSW*	$\overline{DS}$ (Write) Width (LOW)	185		110		ns
41	TdDSI(DR)*	$\overline{DS}$ (Input) ↓ to Read Data Required Valid		330		210	ns

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

Number	Parameters	Description	4MHz Devices		6MHz Devices		Units
			Min	Max	Min	Max	
42	TdC(DSI)	Clock ↓ to $\overline{DS}$ (I/O) ↓ Delay		120		90	ns
43	TwDS*	$\overline{DS}$ (I/O) Width (LOW)	410		255		ns
44	TdAS(DSA)*	$\overline{AS}$ ↑ to $\overline{DS}$ (Acknowledge) ↓ Delay	1065		690		ns
45	TdC(DSA)	Clock ↑ to $\overline{DS}$ (Acknowledge) ↓ Delay		120		85	ns
46	TdDSA(DR)*	$\overline{DS}$ (Acknowledge) ↓ to Read Data Required Delay		455		295	ns
47	TdC(S)	Clock ↑ to Status Valid Delay		110		85	ns
48	TdS(AS)*	Status Valid to AS ↑ Delay	50		30		ns
49	TsR(C)	$\overline{RESET}$ to Clock ↑ Set-up Time	180		70		ns
50	ThR(C)	$\overline{RESET}$ to Clock ↑ Hold Time	0		0		ns
51	TwNMI	$\overline{NMI}$ Width (LOW)	100		70		ns
52	TsNMI(C)	$\overline{NMI}$ to Clock ↑ Set-up Time	140		70		ns
53	TsVI(C)	$\overline{VI}$, $\overline{NVI}$ to Clock ↑ Set-up Time	110		50		ns
54	ThVI(C)	$\overline{VI}$, $\overline{NVI}$ to Clock ↑ Hold Time	20		20		ns
55†	TsSGT(C)	$\overline{SEGT}$ to Clock ↑ Set-up Time	70		55		ns
56†	ThSGT(C)	$\overline{SEGT}$ to Clock ↑ Hold Time	0		0		ns
57	TsMI(C)	$\overline{MI}$ to Clock ↑ Set-up Time	180		140		ns
58	ThMI(C)	$\overline{MI}$ to Clock ↑ Hold Time	0		0		ns
59	TdC(MO)	Clock ↑ to $\overline{MO}$ Delay		120		85	ns
60	TsSTP(C)	$\overline{STOP}$ to Clock ↓ Set-up Time	140		100		ns
61	ThSTP(C)	$\overline{STOP}$ to Clock ↓ Hold Time	0		0		ns
62	TsWT(C)	$\overline{WAIT}$ to Clock ↓ Set-up Time	50		30		ns
63	ThWT(C)	$\overline{WAIT}$ to Clock ↓ Hold Time	10		10		ns
64	TsBRQ(C)	$\overline{BUSRQ}$ to Clock ↑ Set-up Time	90		80		ns
65	ThBRQ(C)	$\overline{BUSRQ}$ to Clock ↑ Hold Time	10		10		ns
66	TdC(BAKr)	Clock ↑ to $\overline{BUSAK}$ ↑ Delay		100		75	ns
67	TdC(BAKf)	Clock ↑ to $\overline{BUSAK}$ ↓ Delay		100		75	ns
68	TwA	Address Valid Width	150		95		ns
69	TdDS(S)	$\overline{DS}$ ↑ to STATUS Not Valid	80		55		ns

*Clock-cycle-time-dependent characteristics. These numbers are computed assuming the clock characteristics are at the limits given in parameters 1 through 5. For other clock frequencies, these parameters can be derived from other specs and the clock characteristics. See tables on following pages.

†Z8001 and Z8001A only.

SWITCHING CHARACTERISTICS over operating ranges unless otherwise specified (Cont'd.)

Number	Parameters	Description	8MHz Devices		Units
			Min	Max	
1	TcC	Clock Cycle Time	125	2000	ns
2	TwCh	Clock Width (HIGH)	55	2000	ns
3	TwCl	Clock Width (LOW)	55	2000	ns
4	TfC	Clock Fall Time		10	ns
5	TrC	Clock Rise Time		10	ns
6†	TdC(SNv)	Clock ↑ to Segment Number Valid (50pF Load)		100	ns
7†	TdC(SNn)	Clock ↑ to Segment Number Not Valid	10		ns
8	TdC(Bz)	Clock ↑ to Bus Float		50	ns
9	TdC(A)	Clock ↑ to Address Valid		65	ns
10	TdC(Az)	Clock ↑ to Address Float		45	ns
11	TdA(DR)*	Address Valid to Read Data Required Valid		225	ns
12	TsDI(C)	Data In to Clock ↓ Set-up Time	15		ns
13	TdDS(A)*	$\overline{DS}$ ↑ to Address Active	40		ns
14	TdC(DW)	Clock ↑ to Write Data Valid		65	ns
15	ThDI(DS)	Data In to $\overline{DS}$ ↑ Hold Time	0		ns
16	TdDO(DS)*	Data Out Valid to $\overline{DS}$ ↑ Delay	150		ns
17	TdA(MR)*	Address Valid to $\overline{MREQ}$ ↓ Delay	30		ns
18	TdC(MR)	Clock ↓ to $\overline{MREQ}$ ↓ Delay		55	ns
19	TwMRh*	$\overline{MREQ}$ Width (HIGH)	105		ns
20	TdMR(A)*	$\overline{MREQ}$ ↓ to Address Not Active	35		ns
21	TdDO(DSW)*	Data Out Valid to $\overline{DS}$ ↓ (Write) Delay	30		ns
22	TdMR(DR)*	$\overline{MREQ}$ ↓ to Read Data Required Valid		175	ns
23	TdC(MR)	Clock ↓ to $\overline{MREQ}$ ↑ Delay		55	ns
24	TdC(ASf)	Clock ↑ to $\overline{AS}$ ↓ Delay		55	ns
25	TdA(AS)*	Address Valid to $\overline{AS}$ ↑ Delay	30		ns
26	TdC(ASr)	Clock ↓ to $\overline{AS}$ ↑ Delay		65	ns
27	TdAS(DR)*	$\overline{AS}$ ↑ to Read Data Required Valid		170	ns
28	TdDS(AS)*	$\overline{DS}$ ↑ to $\overline{AS}$ ↓ Delay	35		ns
29	TwAS*	$\overline{AS}$ Width (LOW)	45		ns
30	TdAS(A)*	$\overline{AS}$ ↑ to Address Not Active Delay	30		ns
31	TdAz(DSR)	Address Float to $\overline{DS}$ (Read) ↓ Delay	0		ns
32	TdAS(DSR)*	$\overline{AS}$ ↑ to $\overline{DS}$ (Read) ↓ Delay	30		ns
33	TdDSR(DR)*	$\overline{DS}$ (Read) ↓ to Read Data Required Valid		115	ns
34	TdC(DSr)	Clock ↓ to $\overline{DS}$ ↑ Delay		65	ns
35	TdDS(DW)*	$\overline{DS}$ ↑ to Write Data and STATUS Not Valid	40		ns
36	TdA(DSR)*	Address Valid to $\overline{DS}$ (Read) ↓ Delay	85		ns
37	TdC(DSR)	Clock ↑ to $\overline{DS}$ (Read) ↓ Delay		70	ns
38	TwDSR*	$\overline{DS}$ (Read) Width (LOW)	140		ns
39	TdC(DSW)	Clock ↓ to $\overline{DS}$ (Write) ↓ Delay		65	ns
40	TwDSW*	$\overline{DS}$ (Write) Width (LOW)	85		ns
41	TdDSI(DR)*	$\overline{DS}$ (Input) ↓ to Read Data Required Valid		135	ns
42	TdC(DSf)	Clock ↓ to $\overline{DS}$ (I/O) ↓ Delay		85	ns
43	TwDS*	$\overline{DS}$ (I/O) Width (LOW)	200		ns
44	TdAS(DSA)*	$\overline{AS}$ ↑ to $\overline{DS}$ (Acknowledge) ↓ Delay	520		ns
45	TdC(DSA)	Clock ↑ to $\overline{DS}$ (Acknowledge) ↓ Delay		65	ns

†Z8001A-8 only.

SWITCHING CHARACTERISTICS (Cont'd.)

Number	Parameters	Description	8MHz Devices		Units
			Min	Max	
46	TdDSA(DR)*	$\overline{DS}$ (Acknowledge) ↓ to Read Data Required Delay		235	ns
47	TdC(S)	Clock ↑ to Status Valid Delay		75	ns
48	TdS(AS)*	Status Valid to $\overline{AS}$ ↑ Delay	25		ns
49	TsR(C)	$\overline{RESET}$ to Clock ↑ Set-up Time	70		ns
50	ThR(C)	$\overline{RESET}$ to Clock ↑ Hold Time	0		ns
51	TwNMI	$\overline{NMI}$ Width (LOW)	50		ns
52	TsNMI(C)	$\overline{NMI}$ to Clock ↑ Set-up Time	70		ns
53	TsVI(C)	$\overline{VI}$,$\overline{NVI}$, to Clock ↑ Set-up Time	50		ns
54	ThVI(C)	$\overline{VI}$,$\overline{NVI}$ to Clock ↑ Hold Time	20		ns
55†	TsSGT(C)	$\overline{SEGT}$ to Clock ↑ Set-up Time	45		ns
56†	ThSGT(C)	$\overline{SEGT}$ to Clock ↑ Hold Time	0		ns
57	TsMI(C)	$\overline{MI}$ to Clock ↑ Set-up Time	90		ns
58	ThMI(C)	$\overline{MI}$ to Clock ↑ Hold Time	0		ns
59	TdC(MO)	Clock ↑ to $\overline{MO}$ Delay		65	ns
60	TsSTP(C)	$\overline{STOP}$ to Clock ↓ Set-up Time	75		ns
61	ThSTP(C)	$\overline{STOP}$ to Clock ↓ Hold Time	0		ns
62	TsWT(C)	$\overline{WAIT}$ to Clock ↓ Set-up Time	25		ns
63	ThWT(C)	$\overline{WAIT}$ to Clock ↓ Hold Time	10		ns
64	TsBRQ(C)	$\overline{BUSRQ}$ to Clock ↑ Set-up Time	60		ns
65	ThBRQ(C)	$\overline{BUSRQ}$ to Clock ↑ Hold Time	10		ns
66	TdC(BAKr)	Clock ↑ to $\overline{BUSAK}$ ↑ Delay		60	ns
67	TdC(BAKf)	Clock ↑ to $\overline{BUSAK}$ ↓ Delay		60	ns
68	TwA*	Address Valid Width	90		ns
69	TdDS(S)*	$\overline{DS}$ ↓ to STATUS Not Valid	45		ns

*Clock-cycle-time-dependent characteristics. These numbers are computed assuming the clock characteristics are at the limits given in parameters 1 through 5. For other clock frequencies, three parameters can be derived from other specs and the clock characteristics. See following table.

†Z8001A-8 only.

CLOCK-CYCLE-TIME-DEPENDENT CHARACTERISTICS

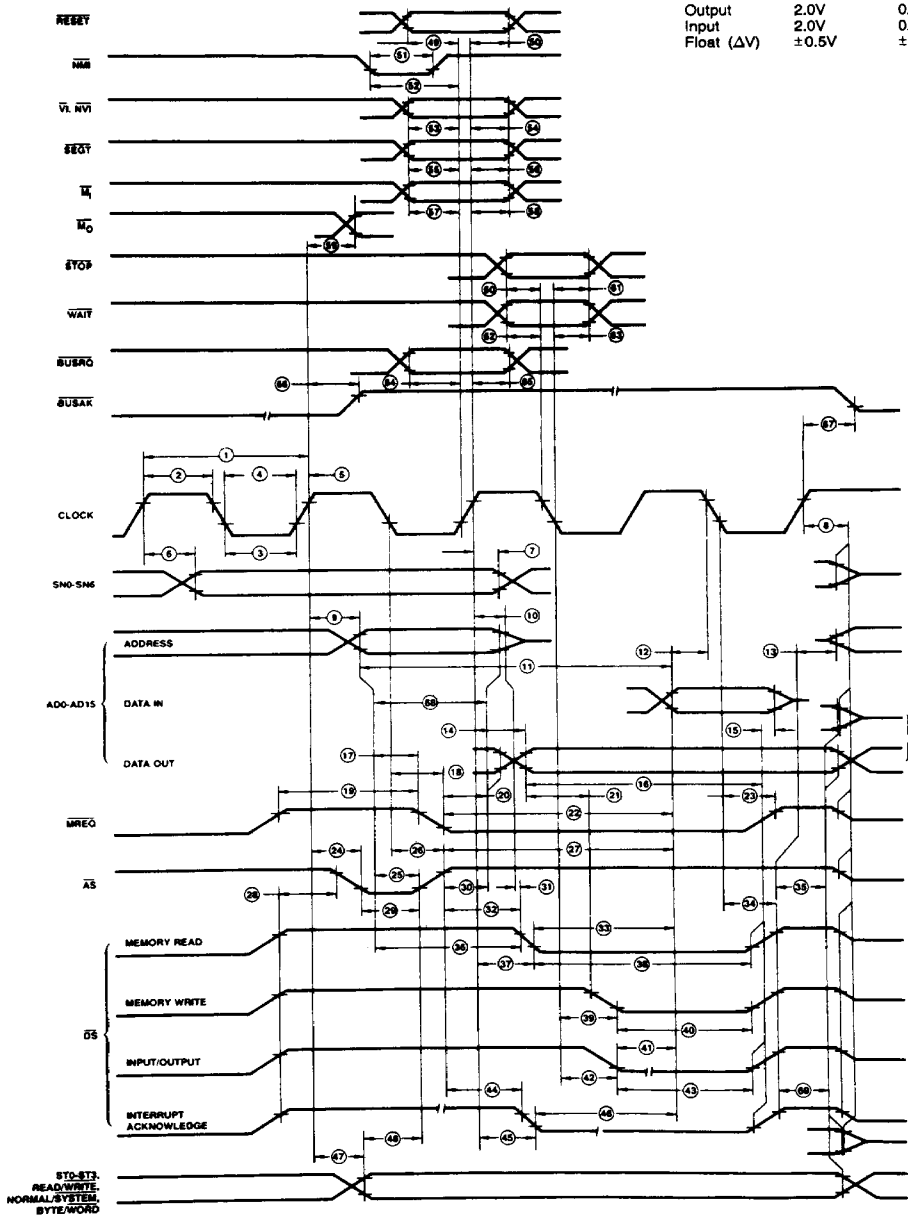
The parameters listed below are also shown in the switching specification. However, they are dependent on the actual values of the clock periods. The equations below define that dependence, so the correct limit for these parameters may be determined for any system, regardless of the actual clock characteristics.

No.	Parameters	4MHz Devices	6MHz Devices	8MHz
11	TdA(DR)	2TcC + TwCH – 125ns	2TcC + TwCh – 95ns	2TcC + TwCh – 80ns
13	TdDS(C)	TwCl – 25ns	TWCl – 30ns	TwCl – 15ns
16	TdDO(DS)	TcC + TwCh – 60ns	TcC+ TwCh – 40ns	TcC + TwCh – 30ns
17	TdA(MR)	TwCh – 50ns	TwCh – 35ns	TwCh – 25ns
19	TwMRh	TcC – 40ns	TcC – 30ns	TcC – 20ns
20	TdMR(A)	TwCl – 35ns	TwCl – 35ns	TwCl – 20ns
21	TdDO(DSW)	TwCh – 50ns	TwCh – 35ns	TwCh – 25ns
22	TdMR(DR)	2TcC – 125ns	2TcC – 105ns	2TcC – 75ns
25	TdA(AS)	TwCh – 50ns	TwCh – 35ns	TwCh – 25ns
27	TdAS(DI)	2TcC – 140ns	2TcC – 115ns	2TcC – 80ns
28	TdDS(AS)	TwCl – 35ns	TwCl – 35ns	TwCl – 20ns
29	TwAS	TwCh – 20ns	TwCh – 15ns	TwCh – 10ns
30	TdAS(A)	TwCl – 35ns	TwCl – 40ns	TwCl – 25ns
32	TdAS(DSR)	TwCl – 25ns	TwCl – 35ns	TwCl – 25ns
33	TdDSR(DR)	TcC + TwCh – 150ns	TcC+ TwCh – 105ns	TcC + TwCh – 65ns
35	TdDS(DW)	TwCl – 30ns	TwCl – 25ns	TwCl – 15ns
36	TdA(DSR)	TcC – 70ns	TcC – 55ns	TcC – 40ns
38	TwDSR	TcC + TwCh – 80ns	TcC+ TwCh – 50ns	TcC + TwCh – 40ns
40	TwDSW	TcC – 65ns	TcC – 55ns	TcC – 40ns
41	TdDSI(DR)	2TcC – 170ns	2TcC – 130ns	2TcC – 115ns
43	TwDS	2TcC – 90ns	2TcC – 75ns	2TcC – 50ns
44	TdAS(DSA)	4TcC + TwCh – 40ns	4TcC – TwCl – 40ns	4TcC + TwCl – 35ns
46	TdDSA(DR)	2TcC + TwCh – 150ns	2TcC+ TwCh – 105ns	2TcC + TwCh – 70ns
48	TdS(AS)	TwCh – 55ns	TwCh – 40ns	TwCh – 30ns
68	TwA	TcC – 90ns	TcC – 70ns	Tcc – 35ns
69	TdDS(S)	TwCl – 25ns	TwCl – 15ns	TwCl – 10ns

Z8001 TIMING DIAGRAM

Timing Measurements are made at the following voltages.

	HIGH	LOW
Clock	4.0V	0.8V
Output	2.0V	0.8V
Input	2.0V	0.8V
Float (ΔV)	±0.5V	±0.5V



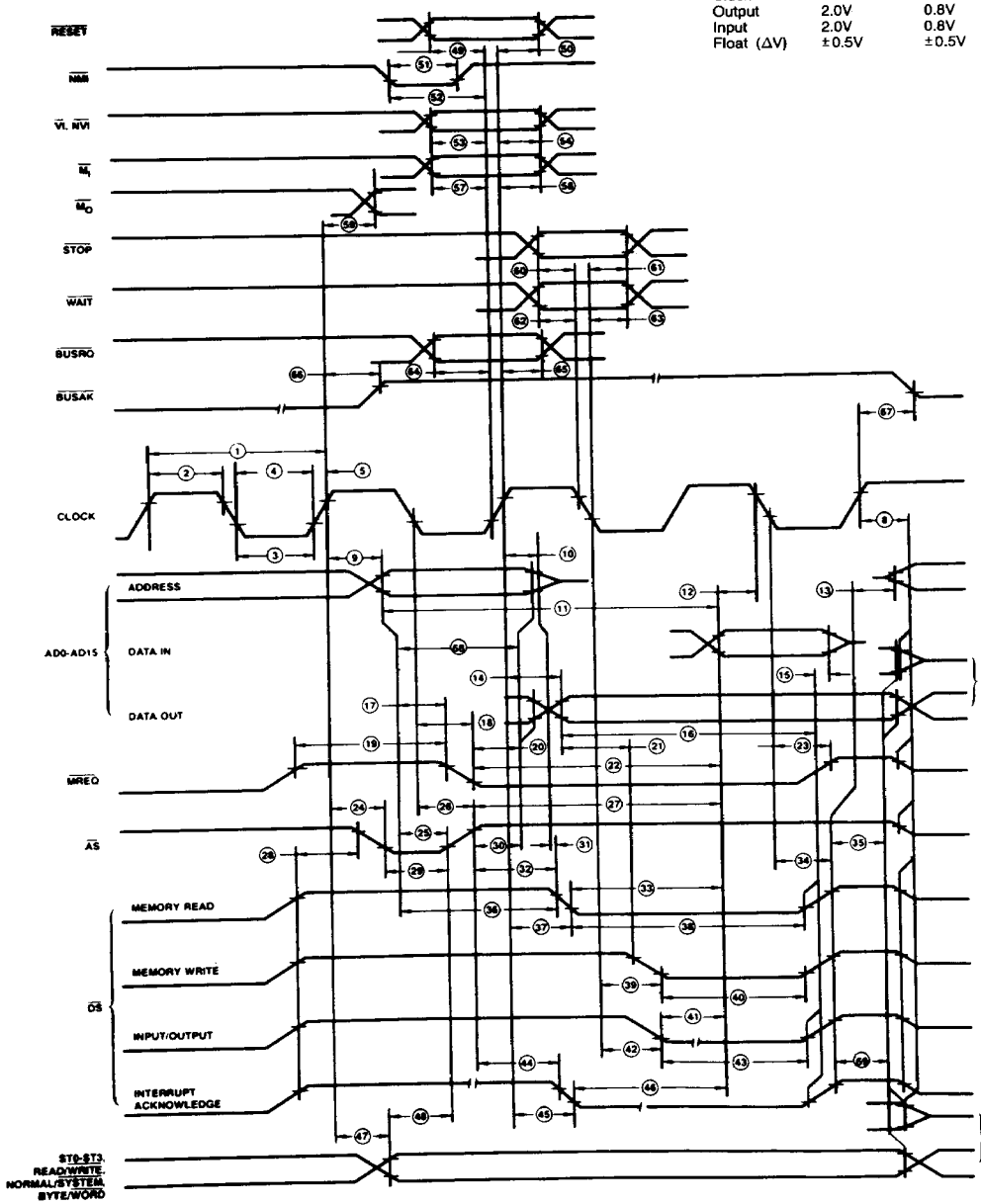
WF005290

This composite timing diagram does not show actual timing sequences. Refer to this diagram only for the detailed timing relationships of individual edges. Use the preceding illustrations as an explanation of the various timing sequences.

Z8002 TIMING DIAGRAM

Timing Measurements are made at the following voltages.

	HIGH	LOW
Clock	4.0V	0.8V
Output	2.0V	0.8V
Input	2.0V	0.8V
Float (ΔV)	±0.5V	±0.5V



WF005300

This composite timing diagram does not show actual timing sequences. Refer to this diagram only for the detailed timing relationships of individual edges. Use the preceding illustrations as an explanation of the various timing sequences.