Section 2. System Board

© Copyright IBM Corp. 1984, 1990

Description

This section describes the system board components and how they interact. Hardware descriptions and programming information are provided to acquaint hardware designers and programmers with the operation of the system.

System Microprocessor

The 80286 microprocessor has the following:

- 10-MHz clock operation
- 24-bit address
- 16-bit data interface
- Instruction set, including string I/O
- Hardware integer multiply and divide operations
- Two operating modes
  - 8086-compatible Real Address mode
  - Protected Virtual Address mode
- 16MB (MB = 1,048,576 or 2^{20} bytes) of physical address space.

Real Address Mode

In the Real Address mode, the system microprocessor address space is a contiguous array of up to 1MB. The system microprocessor addresses memory by generating 20-bit physical addresses.

The segment portion of the pointer is interpreted as the upper 16 bits of a 20-bit segment address. The lower 4 bits of the 20-bit segment address are always 0. Therefore, segment addresses begin on multiples of 16 bytes.

All segments in the Real Address mode are 64KB in size and can be read, written, or executed. An exception or interrupt can occur if data operands or instructions attempt to wrap around the end of a segment (for example, a word with its least significant byte at offset hex FFFF and its most significant byte at hex 0000). If, in the Real Address mode, the information contained in the segment does not use the full 64KB, the unused end of the segment can be overlaid by another segment to reduce physical memory requirements.

Protected Virtual Address Mode

The Protected Virtual Address mode (Protected mode) offers extended physical and virtual memory address space, memory protection mechanisms, and new operations to support operating systems and virtual memory.

The Protected mode provides a 1-gigabyte virtual address space per task mapped into a 16MB physical address space. The virtual address space can be larger than the physical address space, because any use of an address that does not map to a physical memory location will cause a restartable exception.

As in the Real Address mode, the Protected mode uses 32-bit pointers, consisting of 16-bit selector and offset components. The selector specifies an index into a memory-resident table rather than the upper 16 bits of a real memory address. The 24-bit base address of the desired segment is obtained from the tables in memory. The 16-bit offset is added to the segment base address to form the physical address. The tables are automatically referred to by the system microprocessor whenever a segment register is loaded with a selector. All instructions that load a segment register refer to the memory-based tables without additional program support. The memory-based tables contain 8-byte values.

Performance

The 80286 microprocessor operates at 10 MHz for a clock cycle time of 100 nanoseconds. The system inserts one wait state whenever it accesses system board RAM, which results in a 300-nanosecond, 16-bit memory cycle time. The system inserts a minimum of one wait state whenever it does a system board I/O operation, which results in a minimum I/O cycle time of 300-nanoseconds. The system inserts two wait states whenever it accesses system board ROM, which results in a 400-nanosecond 16-bit ROM cycle time.

The bandwidth for memory refresh is approximately 3.0%.

The bus cycle time for memory operations is 1 microsecond for direct memory access (DMA) operations, with the DMA controller operating at 5 MHz with no additional wait states inserted. I/O operations and system board memory operations have one wait state inserted.

Adapter Card Channel

The system board has a 112-pin connector in interface with the adapter card unit. The adapter card unit contains three I/O channel connectors to support two 11-inch adapter boards and one 9.5-inch adapter board.

Connector

The adapter card unit connector on the system board contains the I/O channel signals, power supply voltages, and power supply control (-FAULT).

The I/O channel signals are designed to provide sufficient power for the adapters assuming two low-power Schottky (LS) loads per slot. IBM adapters typically use only one load per adapter.

The pin numbering and signal assignments for the system board are:

Figure 2-1 (Page 1 of 2). Pin Numbers and Signal Assignments for Adapter Card Unit Connector

Pin	Signal	Pin	Signal
1	Ground	57	+36 V
2	Ground	58	Ground
3	-I/O CH CK	59	SD15
4	RESET DRV	60	SD14
5	Ground	61	SD13
6	+5 V	62	SD12
7	IRQ9	63	SD11
8	DRQ2	64	SD10
9	-12 V	65	SD9
10	-OWS	66	SD8
11	+12 V	67	Ground
12	+12 V	68	-MEMW
13	Ground	69	-MEMR
14	I/O CH RDY	70	LA17
15	AEN	71	LA18
16	-SMEMW	72	LA19
17	-SMEMR	73	LA20
18	-IOW	74	LA21
19	-IOR	75	LA22
20	Ground	76	LA23
21	-DACK3	77	-SBHE
22	DRQ3	78	Ground
23	-DACK1	79	SA19
24	DRQ1	80	SA18
25	-REFRESH	81	SA17
26	CLK	82	SA16
27	Ground	83	Ground

Figure 2-1 (Page 2 of 2). Pin Numbers and Signal Assignments for Adapter Card Unit Connector

Pin	Signal	Pin	Signal
28	IRQ7	84	SA15
29	IRQ6	85	SA14
30	IRQ5	86	SA13
31	IRQ4	87	SA12
32	IRQ3	88	Ground
33	-DACK2	89	SA11
34	TC	90	SA10
35	BALE	91	SA9
36	Ground	92	SA8
37	OSC	93	Ground
38	-MEMCS16	94	SA7
39	-I/O CS16	95	SA6
40	IRQ15	96	SA5
41	IRQ14	97	SA4
42	-FAULT	98	Ground
43	IRQ11	99	SA3
44	IRQ10	100	SA2
45	-DACK0	101	SA1
46	DRQ0	102	SA0
47	-DACK5	103	Ground
48	DRQ5	104	SD0
49	Ground	105	SD1
50	-DACK6	106	SD2
51	DRQ6	107	SD3
52	-DACK7	108	Ground
53	DRQ7	109	SD4
54	-MASTER	110	SD5
55	Ground	111	SD6
56	+36 V	112	SD7

Signal Description

The following is a description of the I/O channel signals. All lines are TTL-compatible. The (I), (O), or (I/O) notation refers to input, output, or input and output.

SA0—SA19 (O): Address bits 0 through 19 are used to address memory and I/O devices within the system.

These 20 address lines, in addition to LA17 through LA23, allow access of up to 16MB of memory. SA0 through SA19 are gated on the system bus when BALE is high and are latched on the falling edge of BALE. These signals are generated by the microprocessor or DMA controller. They also may be driven by other microprocessors or DMA controllers that reside on the I/O channel. Only the lower 16 signals are used in I/O addressing, and all 16 should be decoded by I/O devices. SA0 is the least significant and SA19 is the most significant.

LA17—LA23 (O): These signals (unlatched) are used to address memory and I/O devices within the system. They give the system up to 16MB of addressability. These signals are valid when BALE is high. LA17 through LA23 are not latched during microprocessor cycles and therefore do not stay valid for the whole cycle. Their purpose is to generate memory decodes for one wait state memory cycles. These decodes should be latched by I/O adapters on the falling edge of BALE. These signals also may be driven by other microprocessors or DMA controllers that reside on the I/O channel.

AEN (O): The address enable signal is used to de-gate the microprocessor and other devices from the I/O channel to allow DMA transfers to take place. When this signal is active, the DMA controller has control of the address bus, data bus, and Read and Write command signals, (MEMR, SMEMR, IOR, and MEMW, SMEMW, IOW). When this signal is inactive, the microprocessor has control. This signal should be part of the adapter-select decode to prevent incorrect adapter selects during DMA operations.

BALE (O) (Buffered): The buffered address latch enable signal is provided by the bus controller and is used on the system board to latch valid addresses and memory decodes from the microprocessor.

Addresses are latched with the falling edge of BALE. BALE is forced high during DMA cycles. It is available to the I/O channel as an indicator of a valid microprocessor or DMA address when used with AEN.

CLK (O): System clock has a frequency of 10 MHz and a 50% duty cycle. This signal should only be used for synchronization. It is not intended for uses requiring a fixed frequency.

SD0—SD15 (I/O): Data bits 0 through 7 provide data bus bits 0 to 7 for the microprocessor, memory, and I/O devices.

All 8-bit devices on the I/O channel should use SD0 through SD7 for communications to the microprocessor. The 16-bit devices will use SD0 through SD15. To support 8-bit devices, the data on SD8 through SD15 will be gated to SD0 through SD7 during 8-bit transfers to these devices; 16-bit microprocessor transfers to 8-bit devices will be converted to two 8-bit transfers.

-DACK0 to -DACK3 and -DACK5 to -DACK7 (O): -DMA acknowledge 0 to 3 and 5 to 7 are used to acknowledge DMA requests (DRQ0 to DRQ3 and DRQ5 to DRQ7). They are active low.

DRQ0—DRQ3 and DRQ5—DRQ7 (I): DMA requests 0 through 3 and 5 through 7 are asynchronous channel requests used by peripheral devices and the I/O channel microprocessors to gain DMA services. They are prioritized with DRQ0 being the highest and DRQ7 being the lowest. A request is generated by bringing a DRQ line to an active level. A DRQ line must be held high until the corresponding DMA acknowledge line goes active. DRQ0 through DRQ3 will perform 8-bit DMA transfers and DRQ5 through DRQ7 will perform 16-bit transfers. DRQ4 is used on the system board and is not available on the I/O channel.

-I/O CH CK (I): The I/O channel check signal generates an NMI to provide parity information about memory or a device on the I/O channel. It is driven active to indicate an uncorrectable error and held active for at least two clock cycles.

I/O CH RDY (I): The I/O channel ready signal is normally active (ready) and is pulled inactive (not ready) by a memory or I/O device to lengthen I/O or memory cycles. It allows slower devices to attach to the I/O channel with a minimum of difficulty. Any slow device using this signal should drive it inactive immediately after detecting a valid address and a Read or Write command. For every clock cycle this signal is inactive, one wait state is added. This signal should not be held inactive longer than 17 clock cycles.

-IOR (O): The I/O read signal instructs an I/O device to drive its data onto the data bus. This signal is driven by the microprocessor or the DMA controller.

-IOW (O): The I/O write signal instructs an I/O device to read the data on the data bus. This signal is driven by the microprocessor or the DMA controller.

IRQ3–IRQ7, IRQ9–IRQ11, and IRQ14–IRQ15 (I): Interrupt request signals 3 through 7, 9 through 11, and 14 through 15 are used to signal the microprocessor that an I/O device requires attention. They are prioritized with IRQ9 through IRQ11 and IRQ14 through IRQ15 having highest priority (IRQ9 is the highest) and IRQ3 through IRQ7 having the lowest (IRQ7 is the lowest). An interrupt request is generated when an IRQ line is raised from low to high. The line must be held high until the microprocessor acknowledges the interrupt request (Interrupt Service routine). Interrupts 12 and 13 are used on the system board and are not available on the I/O channel. Interrupt 8 is used for the real-time clock.

-SMEMR (O) -MEMR (I/O): These signals instruct memory devices to drive data onto the data bus. -SMEMR is active only when the memory decode is within the low 1MB of memory space. -MEMR is active on all memory read cycles. -MEMR can be driven by any microprocessor or DMA controller in the system. -SMEMR is derived from -MEMR and the decode of the low 1MB of memory. When a microprocessor on the I/O channel wishes to drive -MEMR, it must have the address lines valid on the bus for one clock period before driving -MEMR active. Both signals are active low.

-SMEMW (O) -MEMW (I/O): These signals instruct the memory devices to store the data present on the data bus. -SMEMW is active only when the memory decode is within the low 1MB of the memory space. -MEMW is active on all memory write cycles. -MEMW can be driven by any microprocessor or DMA controller in the system. -SMEMW is derived from -MEMW and the decode of the low 1MB of memory. When a microprocessor on the I/O channel wishes to drive -MEMW, it must have the address lines valid on the bus for one system clock period before driving -MEMW active. Both signals are active low.

-REFRESH (I/O): The memory refresh signal indicates a refresh cycle and can be driven by a microprocessor on the I/O channel.

OSC (O): The oscillator signal is a high-speed clock with a 70 ns period (14.31818 MHz) and a 50% duty cycle. This signal is not synchronous with the system clock.

RESET DRV (O): The reset drive signal is used to reset or initialize system logic upon power on or during a low line-voltage.

TC (O): Terminal Count provides a pulse when the terminal count for any DMA channel is reached.

-MASTER (I): This signal is used with the DRQ line to gain control of the system. A processor or DMA controller on the I/O channel can issue a DRQ to a DMA channel in cascade mode and receive a -DACK. Upon receiving the -DACK, an I/O microprocessor can pull -MASTER low, which will allow it to control the system address, data, and control lines (a condition known as tri-state). After -MASTER is low, the I/O microprocessor must wait one system clock period before driving the address and data lines, and two clock periods before issuing a Read or Write command. If the signal is held low for more than 15 microseconds, system memory can be lost because of the lack of refresh.

-MEM CS16 (I): The MEM 16 chip select signal indicates to the system that the present data transfer is a one wait state 16-bit memory cycle. It must be derived from the decode of LA17 through LA23. -MEM CS16 should be driven with an open-collector or tri-state driver capable of sinking 20mA.

-I/O CS16 (I): The -I/O 16-bit chip select signal indicates to the system that the present data transfer is a 16-bit, one wait state, I/O cycle. It is derived from an address decode. -I/O CS16 is active low and should be driven with an open-collector or tri-state driver capable of sinking 20mA.

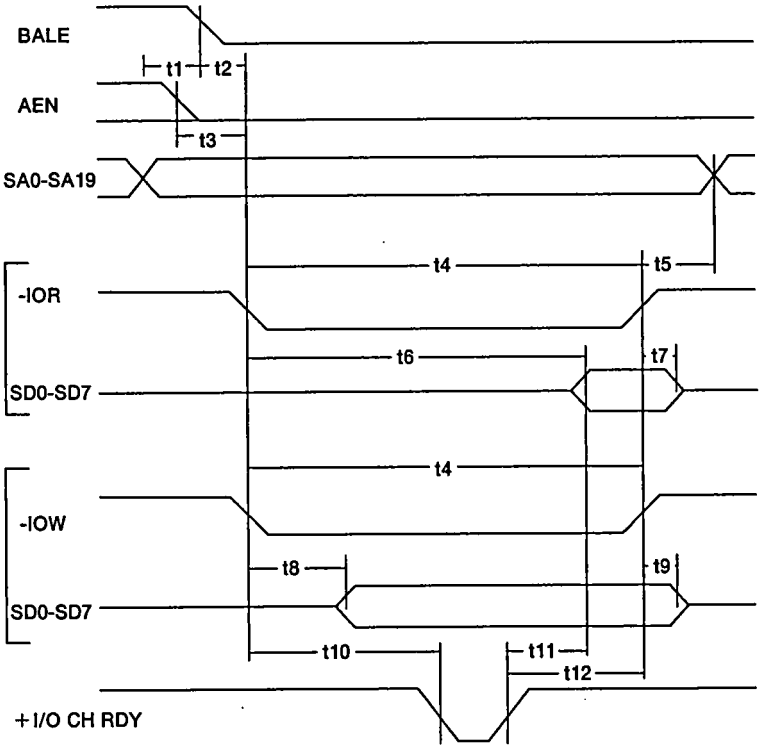
-0WS (I): The zero wait state signal tells the microprocessor that it can complete the present bus cycle without inserting any additional wait states. In order to run a memory cycle to a 16-bit device without wait states, -0WS is derived from an address decode. In order to run a memory cycle to an 8-bit device with a minimum of two wait states, -0WS should be driven active one system clock after the Read or Write command is active and gated with the address decode for the device. Memory Read and Write commands to an 8-bit device are active on the falling edge of the system clock. -0WS is active low and should be driven with an open collector or tri-state driver capable of sinking 20mA.

-SBHE (I/O): The system bus high enable signal indicates a transfer of data on the high-order byte of the data bus, SD8 - SD15. Sixteen-bit devices use SBHE to condition data bus buffers tied to SD8 through SD15.

Signal Timings

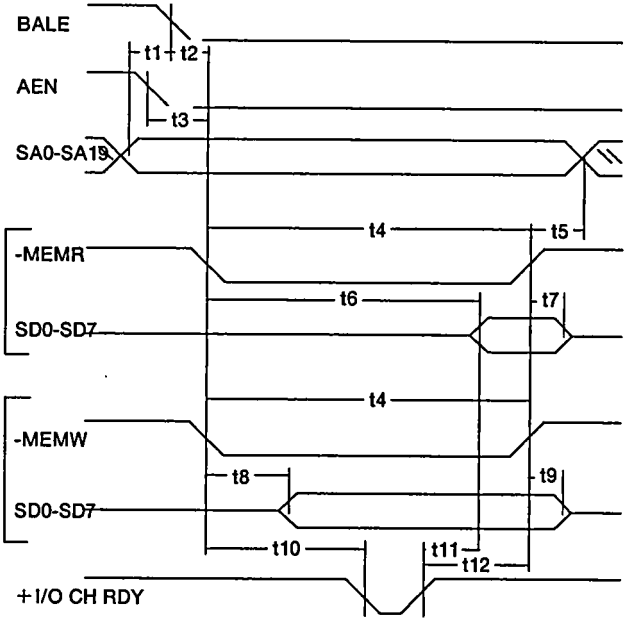
The following diagrams show the signal timings for I/O and memory operations.

8-Bit I/O Bus Cycles



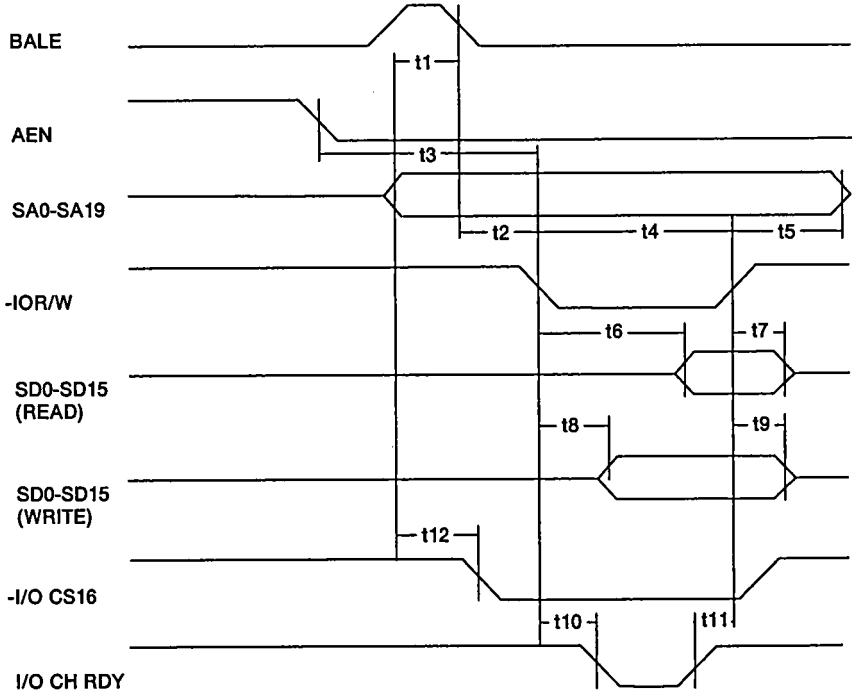
Symbol	Description	Min (ns)	Max (ns)
t1	Address valid to BALE inactive	10	
t2	BALE inactive to Command active	25	
t3	Command active from AEN inactive	150	
t4	Command pulse width	415	
t5	Address hold from Command inactive	20	
t6	Data valid from Read active		372
t7	Data hold from Read inactive	0	
t8	Data valid from Write active		55
t9	Data hold from Write inactive	15	
t10	I/O CH RDY inactive from Command active		285
t11	Read Data valid from I/O CH RDY active		0
t12	Command inactive from I/O CH RDY active	130	

8-Bit Memory Bus Cycles



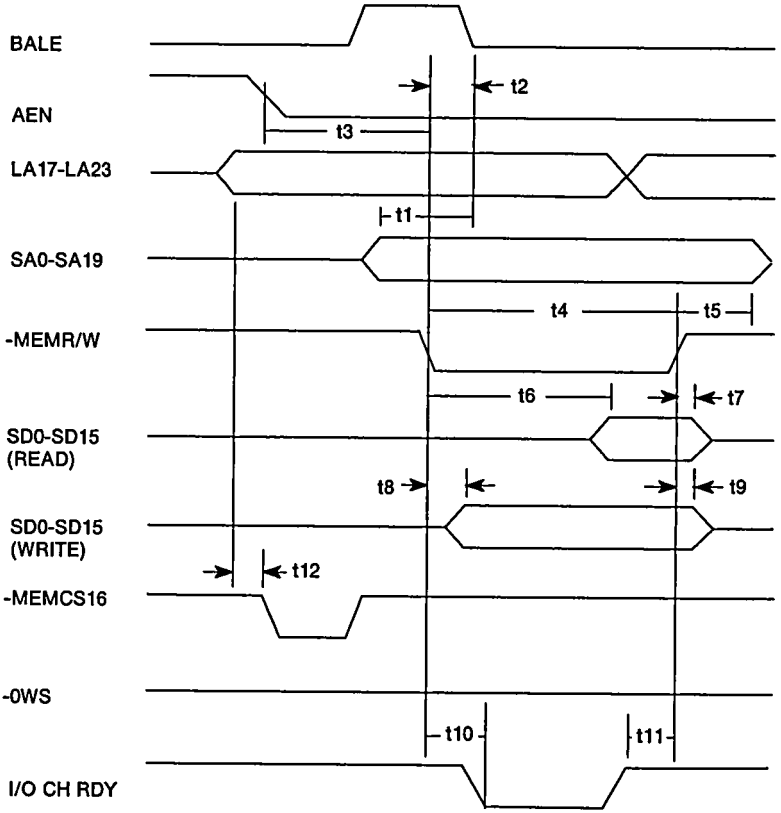
Symbol	Description	Min (ns)	Max (ns)
t1	Address valid to BALE inactive	10	
t2	BALE inactive to Command active	25	
t3	Command active from AEN inactive	150	
t4	Command pulse width	415	
t5	Address hold from Command inactive	20	
t6	Data valid from Read active		372
t7	Data hold from Read inactive	0	
t8	Data valid from Write active		55
t9	Data hold from Write inactive	15	
t10	I/O CH RDY inactive from Command active		285
t11	Read Data valid from I/O CH RDY active		0
t12	Command inactive from I/O CH RDY active	130	

16-Bit I/O Bus Cycles



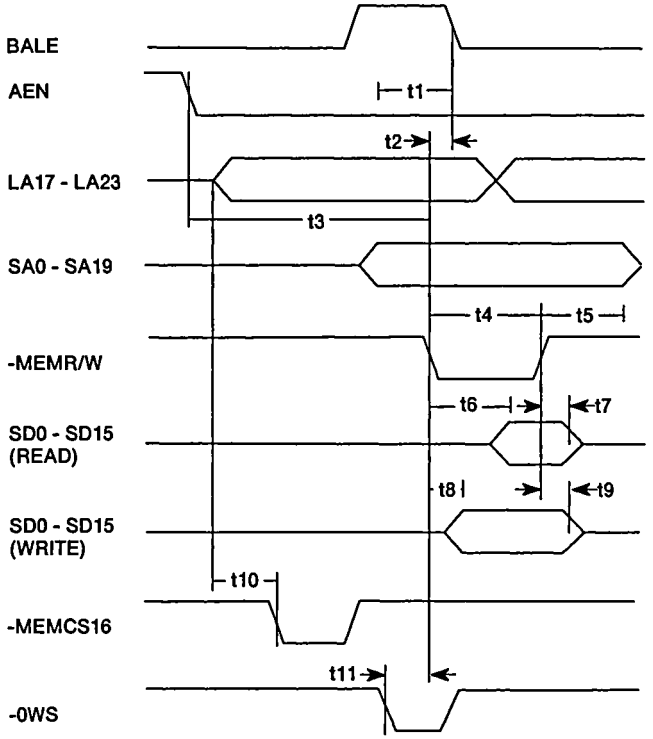
Symbol	Description	Min (ns)	Max (ns)
t1	Address valid to BALE inactive	10	
t2	BALE inactive to Command active	25	
t3	Command active from AEN inactive	150	
t4	Command pulse width	115	
t5	Address hold from Command inactive	20	
t6	Data valid from Read active		72
t7	Data hold from Read inactive	0	
t8	Data valid from Write active		55
t9	Data hold from Write inactive	15	
t10	I/O CH RDY inactive from Command active		20
t11	Command Inactive from I/O CH RDY active	130	
t12	-I/O CS16 active from SA0-SA19 valid		62

16-Bit Memory Bus Cycles (1 Wait State)



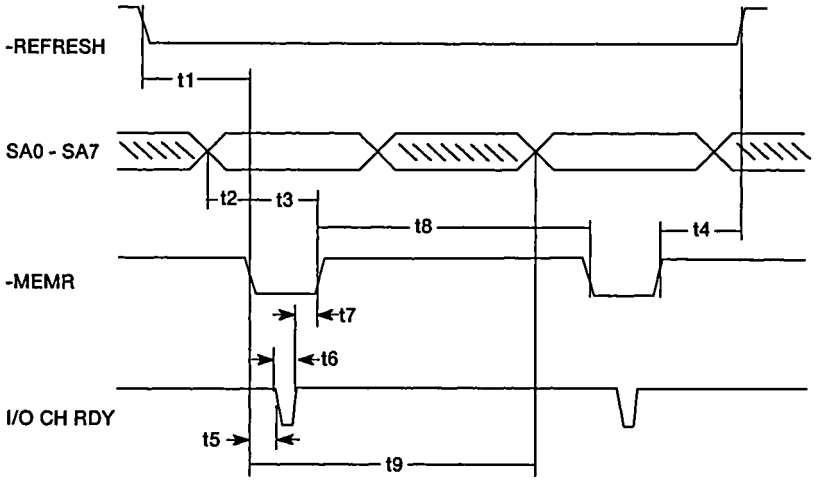
Symbol	Description	Min (ns)	Max (ns)
t1	Address valid to BALE inactive	10	
t2	Command active to BALE inactive	25	
t3	Command active from AEN inactive	150	
t4	Command pulse width	165	
t5	Address hold from Command inactive	20	
t6	Data valid from Read active		122
t7	Data hold from Read inactive	0	
t8	Data valid from Write active		35
t9	Data hold from Write inactive	15	
t10	I/O CH RDY inactive from Command active		35
t11	Command Inactive from I/O CH RDY active	130	
t12	-MEMCS16 active from LA17-LA23 valid		43

16-Bit Memory Bus Cycles (0 Wait State)



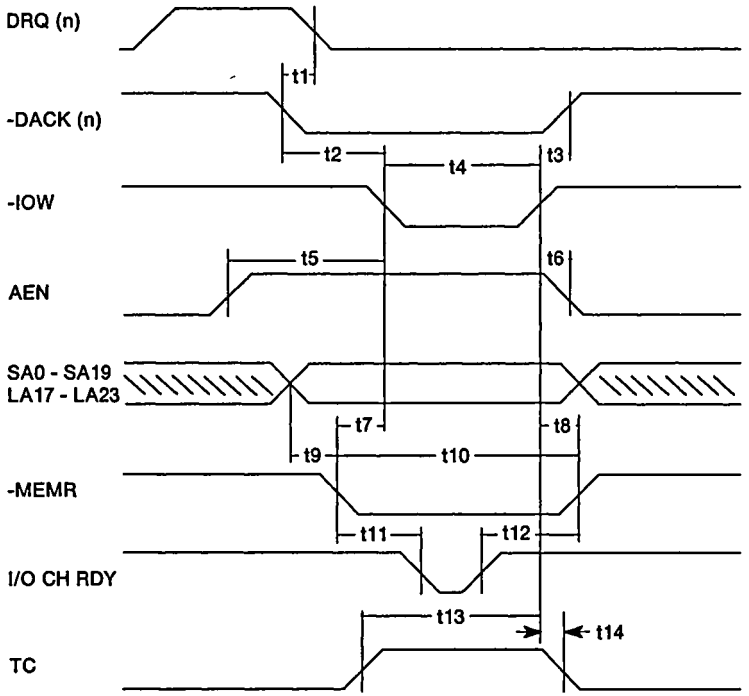
Symbol	Description	Min (ns)	Max (ns)
t1	Address valid to BALE inactive	10	
t2	Command active to BALE inactive	25	
t3	Command active from AEN inactive	150	
t4	Command pulse width	65	
t5	Address hold from Command inactive	20	
t6	Data valid from Read active		22
t7	Data hold from Read inactive	0	
t8	Data valid from Write active		35
t9	Data hold from Write inactive	15	
t10	-MEMCS16 active from LA17-LA23 valid		43
t11	-0WS active to -MEMR/W active	12	

Memory Refresh



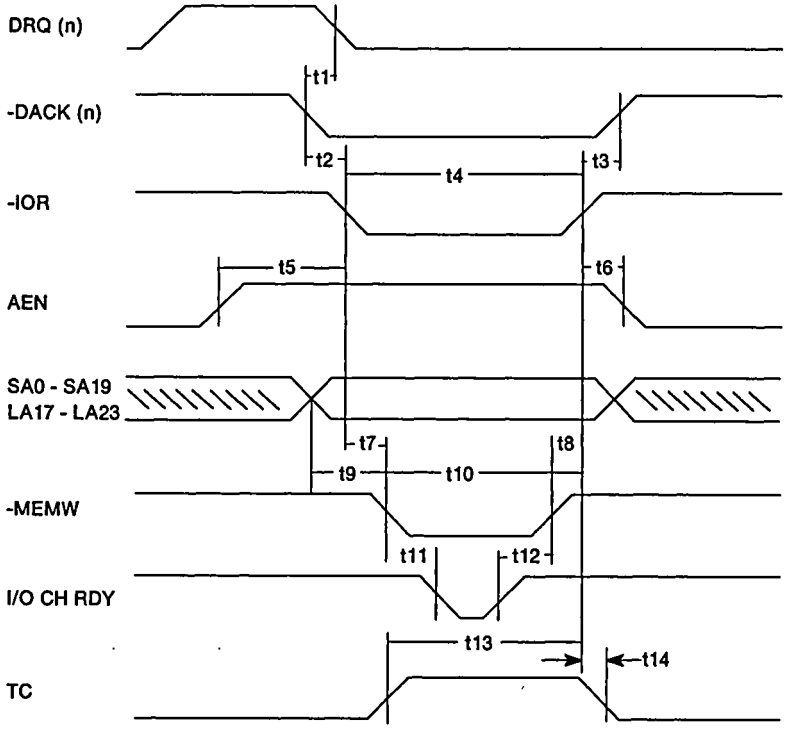
Symbol	Description	Min (ns)	Max (ns)
t1	-REFRESH active to -MEMR active	130	
t2	Address valid to -MEMR active	40	
t3	-MEMR pulse width	150	
t4	-MEMR inactive to -REFRESH inactive	10	
t5	-MEMR active to I/O CH RDY inactive		60
t6	I/O CH RDY pulse width		600
t7	-MEMR inactive from I/O CH RDY active	0	
t8	-MEMR inactive to next -MEMR active	200	
t9	Next address valid from -MEMR active	300	

DMA Read



Symbol	Description	Min (ns)	Max (ns)
t1	-DACK active to DRQ inactive	0	
t2	-DACK active to -IOW active	260	
t3	-IOW inactive to -DACK inactive	0	
t4	-IOW pulse width	360	
t5	AEN active to -IOW active	400	
t6	-IOW inactive to AEN inactive	0	
t7	-IOW active from -MEMR active	0	
t8	-IOW inactive to -MEMR inactive	0	
t9	Address valid to -MEMR active	130	
t10	-MEMR pulse width	360	
t11	-MEMR active to I/O CH RDY inactive		75
t12	-MEMR inactive from I/O CH RDY active	200	
t13	TC active setup to -IOW inactive	310	
t14	TC inactive from -IOW inactive	0	

DMA Write



Symbol	Description	Min (ns)	Max (ns)
t1	-DACK active to DRQ inactive	0	
t2	-DACK active to -IOR active	0	
t3	-IOR inactive to -DACK inactive	0	
t4	-IOR pulse width	560	
t5	AEN active to -IOR active	250	
t6	-IOR inactive to AEN inactive	0	
t7	-MEMW active from -IOR active	70	
t8	-MEMW inactive to -IOR inactive	0	
t9	Address valid to -MEMW active	130	
t10	-MEMW pulse width	330	
t11	-MEMW active to I/O CH RDY inactive		75
t12	-MEMW inactive from I/O CH RDY active	200	
t13	TC active setup to -IOR inactive	310	
t14	TC inactive from -IOR inactive	0	

DMA Controller

The direct memory access (DMA) controller allows I/O devices to transfer data directly to and from memory. This allows higher system microprocessor throughput by freeing the system microprocessor of I/O tasks.

The DMA controller is software programmable. The system microprocessor can address the DMA controller and read or modify the internal registers to define the DMA modes, transfer addresses, transfer counts, channel masks, and page registers.

The functions of the DMA controller can be grouped in two categories; program condition and DMA transfer. During program condition, the DMA registers can be programmed or read.

Program condition commences when the system microprocessor refers to the DMA controller within a specific address range. (See "DMA I/O Address Map" on page 2-23.) The DMA controller needs a minimum of 1.0 microseconds (one wait state) to complete any program command generated by the system microprocessor.

Single transfers require a minimum of 1.0 microseconds to transfer either a byte or a word.

The DMA controller supports:

- Register/Program compatibility with the IBM Personal Computer AT® DMA channels (8237 Compatible mode).
- 16MB (24-bit) address capability for memory.
- Seven independent DMA channels capable of transferring data between memory and I/O devices.
- Serial DMA operation with overlapped read and write cycles for each transfer operation.
- Each channel fixed to byte or word transfer.
- Sharing of the system bus interface and control logic.

Personal Computer AT is a registered trademark of International Business Machines Corporation.

Data Transfers Between Memory and I/O Devices

The DMA controller performs serial transfers with a minimum of five 200-nanosecond clock cycles for memory read to I/O, or I/O read to memory write operations. No burst mode or memory-to-memory transfers are supported.

Byte Pointer

A byte pointer allows 8-bit ports to access consecutive bytes of registers greater than 8 bits. These registers are the Memory Address registers (3 bytes), the Transfer Count registers (2 bytes), and the I/O Address registers (2 bytes).

DMA Channels

DMA channels 0 through 3 support 8-bit data transfers between 8-bit I/O adapters and 8-bit or 16-bit system memory. Each channel can transfer data throughout the 16MB system-address space in 64KB blocks. Logic is included on the system board containing the equivalent of two 8237 controller chips.

Figure 2-2 shows DMA channel assignments.

Figure 2-2. DMA Channel Assignments

Channel	Assignment
DRQ0	Unused
DRQ1	Unused
DRQ2	Diskette
DRQ3	Fixed Disk
DRQ4	Cascade
DRQ5	Unused
DRQ6	Unused
DRQ7	Unused

Address Generation for DMA Channels 0 through 3

Figure 2-3 shows address generation for DMA channels 0 through 3.

Figure 2-3. DMA Address Generation for Channels 0-3

Source	DMA Page Registers	Controller
Address	A23---------A16	A15---------A0

Note: The byte high enable (BHE) addressing signal is generated by inverting address line A0.

Address Generation for DMA Channels 5 through 7

DMA channel 4 is used to cascade channels 5 through 7 to the microprocessor. Channels 5, 6, and 7 support 16-bit data transfers between 16-bit I/O adapters and 16-bit system memory. These channels can transfer data throughout the 16MB system-address space in 128KB blocks. Channels 5, 6, and 7 cannot transfer data on odd-byte boundaries.

Figure 2-4 shows address generation for the DMA channels 5 through 7.

Figure 2-4. DMA Address Generation for Channels 5-7

Source	DMA Page Registers	Controller
Address	A23---------A17	A16---------A1

Note: The addressing signals, BHE and A0, are forced to a logical 0.

Page Register Addresses

Figure 2-5 shows the addresses for the page registers.

Figure 2-5. Page Register Addresses

Page Register	I/O Hex Address
DMA Channel 0	0087
DMA Channel 1	0083
DMA Channel 2	0081
DMA Channel 3	0082
DMA Channel 5	008B
DMA Channel 6	0089
DMA Channel 7	008A
Refresh	008F

Addresses for all DMA channels do not increase or decrease through page boundaries (64KB for channels 0 through 3, and 128KB for channels 5 through 7).

DMA channels 5 through 7 perform 16-bit data transfers. Access is only to 16-bit devices (I/O or memory) during the DMA cycles of channels 5 through 7. Access to the DMA controller, which controls these channels, is through I/O addresses hex 0C0 through 0DF.

All DMA memory transfers made with channels 5 through 7 must occur on even-byte boundaries. When the base address for these channels is programmed, the real address divided by 2 is the data written to the base address register. Also, when the base word count for channels 5 through 7 is programmed, the count is the number of 16-bit words to be transferred. Therefore, DMA channels 5 through 7 can transfer 65,536 words, or 128KB maximum, for any selected page of memory. These DMA channels divide the 16MB memory space into 128KB pages. When the DMA page registers for channels 5 through 7 are programmed, data bits D7 through D1 contain the most-significant 7 address bits (A23 through A17) of the desired memory space. Data bit D0 of the page registers for channels 5 through 7 is not used in the generation of the DMA memory address.

At power-on time, all internal locations, especially the mode registers, should be loaded with some valid value. This is done even if some channels are unused.

DMA I/O Address Map

Figure 2-6. DMA I/O Addresses for Memory Addresses, Word Counts, and Command/Status Registers

Address (hex)	Description	Bit Description	Byte Pointer
0000	Channel 0, Memory Address Register	00-15	Yes
0001	Channel 0, Transfer Count Register	00-15	Yes
0002	Channel 1, Memory Address Register	00-15	Yes
0003	Channel 1, Transfer Count Register	00-15	Yes
0004	Channel 2, Memory Address Register	00-15	Yes
0005	Channel 2, Transfer Count Register	00-15	Yes
0006	Channel 3, Memory Address Register	00-15	Yes
0007	Channel 3, Transfer Count Register	00-15	Yes
0008	Channel 0-3, Rd Status/Wrt Command Reg	00-07	
0009	Channel 0-3, Write Request Register	00-02	
000A	Channel 0-3, Write Single Mask Reg Bit	00-02	
000B	Channel 0-3, Mode Register (Write)	00-07	
000C	Channel 0-3, Clear Byte Pointer (Write)	N/A	
000D	Channel 0-3, Master Clear (W)/Temp (R)	00-07	
000E	Channel 0-3, Clear Mask Register (Write)	00-03	Yes
000F	Channel 0-3, Write All Mask Register Bits	00-03	Yes
0081	Channel 2, Page Table Address Register **	00-07	
0082	Channel 3, Page Table Address Register **	00-07	
0083	Channel 1, Page Table Address Register **	00-07	
0087	Channel 0, Page Table Address Register **	00-07	
0089	Channel 6, Page Table Address Register **	00-07	
008A	Channel 7, Page Table Address Register **	00-07	
008B	Channel 5, Page Table Address Register **	00-07	
008F	Channel 4, Pg Tbl Address/Refresh Register	00-07	
00C0	Channel 4, Memory Address Register	00-15	Yes
00C2	Channel 4, Transfer Count Register	00-15	Yes
00C4	Channel 5, Memory Address Register	00-15	Yes
00C6	Channel 5, Transfer Count Register	00-15	Yes
00C8	Channel 6, Memory Address Register	00-15	Yes
00CA	Channel 6, Transfer Count Register	00-15	Yes
00CC	Channel 7, Memory Address Register	00-15	Yes
00CE	Channel 7, Transfer Count Register	00-15	Yes
00D0	Channel 4-7, Read Status/Write Command Register	00-07	
00D2	Channel 4-7, Write Request Register	00-02	
00D4	Channel 4-7, Write Single Mask Register Bit	00-02	
00D6	Channel 4-7, Mode Register (Write)	00-07	
00D8	Channel 4-7, Clear Byte Pointer (Write)	N/A	
00DA	Channel 4-7, Master Clear (W)/Temp (R)	00-07	
00DC	Channel 4-7, Clear Mask Register (Write)	00-03	
00DE	Channel 4-7, Write All Mask Register Bits	00-03	

* Dependent upon the function used.

** Upper byte of Memory Address register.

DMA Registers

All system microprocessor access to the DMA must be 8-bit I/O instructions. Figure 2-7 lists the name and size of the DMA registers.

Figure 2-7. DMA Registers

Register	Size (bits)	Quantity of Registers	Allocation
Memory Address	16	8	1 per Channel
Transfer Count	16	8	1 per Channel
Page	8	8	1 per Channel
Mask	4	2	1 for Channels 7 - 4 1 for Channels 3 - 0
Mode	8	8	1 per Channel
Status	8	2	1 for Channel 7 - 4 1 for Channel 3 - 0

Memory Address Register

Each DMA channel has a 16-bit Memory Address register. This register holds the value of the address used during DMA transfers. The address is incremented or decremented after each transfer and the intermediate values of the address are stored in this register during the transfer. This register is written or read by the microprocessor in successive 8-bit bytes. Auto-initialization will restore this register to its original value.

The contents of the corresponding DMA page register is the most significant byte of the DMA memory address.

Transfer Count Register

Each DMA channel has a 16-bit Transfer Count register that is loaded by the system microprocessor. The transfer count determines the number of transfers to be executed by the DMA channel before reaching terminal count. The number of transfers is always one more than the count specifies. For example, if the count is 0, the DMA does one transfer. When the value in the register goes from hex 0000 to FFFF, a terminal count pulse is generated by the DMA channel. The Transfer Count register can be read by the system microprocessor in successive I/O bytes when the DMA controller is in the program condition. Auto-initialization will restore the register to its original value.

Mask Register

Each DMA channel has a corresponding mask bit that, when set, disables the DMA from servicing the requesting device. Each mask bit can be set or cleared by the system microprocessor. A system reset or DMA master clear sets all mask bits to 1. A Clear Mask Register command sets all mask bits to 0. This register can be programmed using the 8237 Compatible mode commands.

Figure 2-8. Set/Clear Single Mask Bit Using 8237 Compatible Mode

Bit	Function
7 - 3	Reserved (Must be set to 0)
2	0 = Clear Mask Bit 1 = Set Mask Bit
1, 0	00 = Select Channel 0 or 4 01 = Select Channel 1 or 5 10 = Select Channel 2 or 6 11 = Select Channel 3 or 7

Figure 2-9. DMA Mask Register Write Using 8237 Compatible Mode

Bit	Function
7 - 4	Reserved (Must be set to 0)
3	0 = Clear Channel 3 or 7 Mask Bit 1 = Set Channel 3 or 7 Mask Bit
2	0 = Clear Channel 2 or 6 Mask Bit 1 = Set Channel 2 or 6 Mask Bit
	0 = Clear Channel 1 or 5 Mask Bit 1 = Set Channel 1 or 5 Mask Bit
0	0 = Clear Channel 0 or 4 Mask Bit 1 = Set Channel 0 or 4 Mask Bit

Mode Register

Each DMA channel has a Mode register that identifies the type of operation that takes place when that channel is activated. The Mode register is programmed by the system microprocessor and the contents are reformatted and stored internally in the DMA controller. This register can only be written.

Figure 2-10. 8237 Compatible Mode Register

Bit	Function
7-6	Reserved (Must be set to 0)
5	00 = Address Increment 01 = Address Decrement
4	00 = Auto-initialize Disable 01 = Auto-initialize Enable
3, 2	00 = Verify Operation 01 = Write Operation 10 = Read Operation 11 = Reserved
1, 0	00 = Select Channel 0 or 4 01 = Select Channel 1 or 5 10 = Select Channel 2 or 6 11 = Select Channel 3 or 7

Status Register

There are two Status registers containing information about the status of the devices. This information tells which channels have reached terminal count. Bits 3 through 0 are set every time a terminal count is reached by a corresponding channel. All bits are cleared by reset or following a system microprocessor Status Read command. When these registers are written, they are Command registers as shown in Figure 2-12 on page 2-27.

Figure 2-11. Status Register (Read)

Bit	Function
7	Channel 3 or 7 Request
6	Channel 2 or 6 Request
5	Channel 1 or 5 Request
4	Channel 0 Request
3	Channel 3 or 7 Terminal Count
2	Channel 2 or 6 Terminal Count
1	Channel 1 or 5 Terminal Count
0	Channel 0 Terminal Count

Figure 2-12. Command Register (Write)

Bit	Function
7	0 = DACK Active Low 1 = DACK Active High
6	0 = DRQ Active High 1 = DRQ Active Low
5	0 = Late Write 1 = Extended Write
4	0 = Fixed Priority 1 = Rotating Priority
3	0 = Normal Timing 1 = Compressed Timing
2	0 = Controller Enable 1 = Controller Disable
1	Reserved
0	Reserved (Must be 0)

Figure 2-13 shows an example of programming DMA channel 2.

Figure 2-13. DMA Channel 2 Programming Example

Program Step	OUT to ADRS Data
Set Channel Mask Bit	(000AH) x6H
Clear Byte Pointer	(000CH) xxH
Write Memory Address	(0004H) xxH
Write Memory Address	(0004H) xxH
Write Page Table Address	(0081H) xxH
Clear Byte Pointer	(000CH) xxH
Write Register Count	(0005H) xxH
Write Register Count	(0005H) xxH
Write Mode Register	(000BH) xxH
Clear Channel 2 Mask Bit	(000AH) x2H

Values inside ( ) are addresses, x represents data.

Interrupts

The system provides 15 levels of system interrupts. Any or all of the interrupts can be masked, including the non-maskable interrupt. The system board uses the logic equivalent of two Intel 8259A interrupt controllers.

Non-Maskable Interrupt

The non-maskable interrupt (NMI) signals the system microprocessor that a channel check has occurred. The NMI masks all other interrupts and the IRET instruction restores the interrupt flag to the state it was in prior to the interrupt. A system reset resets the NMI.

A NMI request from channel check is subject to mask control from the NMI mask bit at I/O address hex 0070. This address is shared with the address for the RT/CMOS RAM. (See "RTC/CMOS RAM I/O Operations" on page 3-159.) The power-on default of the NMI mask is 1 (NMI disabled). Prior to enabling the NMI after a power-on reset (write to address hex 0070 with bit 7 equal to 0), the channel check state is initialized by the POST.

Warning: When writing to address hex 0070 to enable or disable an NMI, a read to address hex 0071 must be accessed immediately. Failure to do this can cause unreliable operation of the Real-Time Clock/CMOS RAM.

Interrupt Assignments

Figure 2-14 shows the interrupt levels and their functions. The interrupt levels are listed in the order of priority. The highest priority is the NMI and the lowest is IRQ7.

Figure 2-14. Interrupt Level Assignments by Priority

Level		Function
NMI		Channel Check
IRQ0		Timer
IRQ1		Keyboard
IRQ2		Cascade Interrupt Control to IRQ8 - IRQ15
	IRQ8	Real-Time Clock
	IRQ9	Redirect Cascade
	IRQ10	Reserved
	IRQ11	Reserved
	IRQ12	Mouse
	IRQ13	Reserved
	IRQ14	Fixed Disk
	IRQ15	Reserved
IRQ3		Serial Alternate
IRQ4		Serial Primary
IRQ5		Reserved
IRQ6		Diskette
IRQ7		Parallel Port/Audio Card

IRQ 8 through 15 are cascaded through IRQ 2.

Hardware interrupt IRQ9 is defined as the replacement interrupt level for the cascade level IRQ2. Program interrupt sharing should be implemented on IRQ2, interrupt hex 0A. The following processing occurs to maintain compatibility with the IRQ2 used by IBM Personal Computer products:

1. A device drives the interrupt request active on IRQ2 of the channel.
2. This interrupt request is mapped in hardware to IRQ9 input on the second interrupt controller.

3. When the interrupt occurs, the system microprocessor passes control to the IRQ9 (interrupt hex 71) interrupt handler.
4. This interrupt handler performs an end of interrupt (EOI) to the second interrupt controller and passes control to IRQ2 (interrupt hex 0A) interrupt handler.
5. This IRQ2 interrupt handler causes the device to reset the interrupt request prior to performing an EOI to the master interrupt controller that finishes servicing the IRQ2 request.

Note: Prior to programming the interrupt controllers, interrupts should be disabled by executing a CLI instruction. This includes the Mask register, end of interrupts, initialization control words, and operational control words.

System Timers

The system has three programmable timers: channel 0, channel 1, and channel 2. Channels 0, 1, and 2 are similar to channels 0, 1, and 2 of the IBM Personal Computer, IBM Personal Computer XT™, and the IBM Personal Computer AT.

Figure 2-15 is a block diagram of the timers.

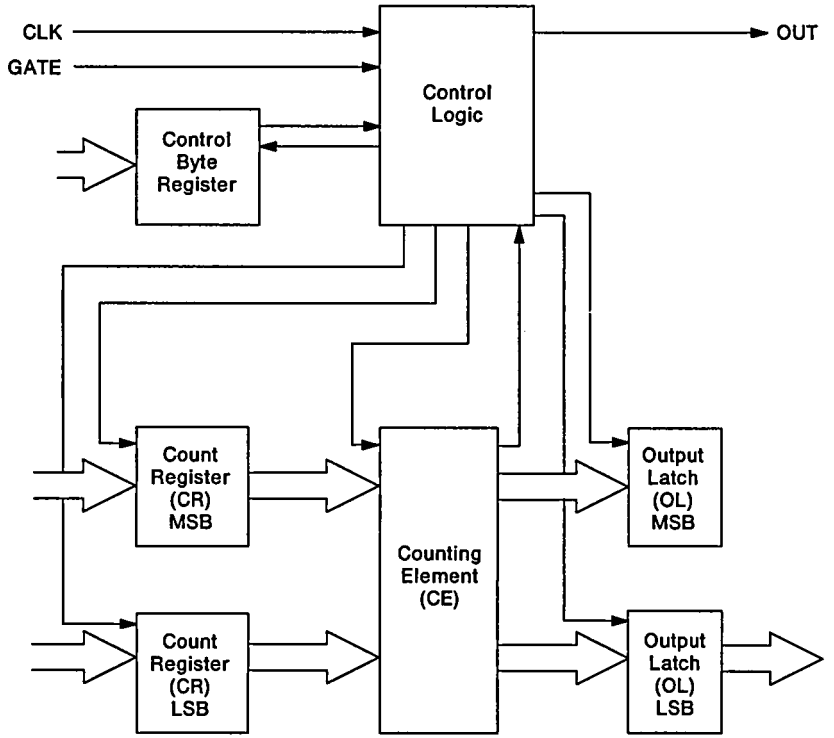


Figure 2-15. System Timer Block Diagram

Personal Computer XT is a trademark of International Business Machines Corporation.

Channel 0 - System Timer

- GATE 0 is always enabled.
- CLK IN 0 is driven by 1.190 MHz.
- CLK OUT 0 drives the interrupt controller chip, IRQ 0.

Channel 1 - Refresh Request Generator

- GATE 1 is always enabled.
- CLK IN 1 is driven by 1.190 MHz.
- CLK OUT 1 is request-refresh cycle.

  Note: Channel 1 is programmed as a rate generator to produce a 15-microsecond signal.

Channel 2 - Tone Generation for Speaker

- GATE 2 is controlled by bit 0 (PPI) of port hex 0061.
- CLK IN 2 is driven by 1.190 MHz.
- CLK OUT 2 is connected to the input port hex 0061, bit 5. CLK OUT 2 is also logically ANDed with port hex 0061, bit 1 to produce the speaker data enable signal.

Timers 0, 1, and 2

Each timer is independent. Timers 0, 1, and 2 are 16-bit down counters that can be preset. They can count in binary or binary-coded decimal (BCD).

Programming the System Timers

The system treats the programmable interval timer as an arrangement of four external I/O ports. Three ports are treated as count registers and one as control register for mode programming. Timers are programmed by writing a control word and then an initial count. All control words are written into the control word registers, which are located at I/O address hex 0043 for timers 0, 1, and 2. Initial counts are written into the count registers, not the control word registers. The format of the initial count is determined by the control word used.

The count is written to the count register. It is then transferred to the counting element, according to the mode definition. When the count is read, the data is presented at the output latch.

Counter Write Operations

The control word must be written before the initial count is written.

The count must follow the count format specified in the control word.

A new initial count can be written to the counters at any time without affecting the programmed mode of the counter. Counting is affected as described in the mode definitions. The new count must follow the programmed count format.

Counter Read Operations

The counters can be read using the Counter Latch command. (See "Counter Latch Command" on page 2-35.)

If the counter is programmed for 2-byte counts, 2 bytes must be read. The 2 bytes need not be read consecutively; read, write, or programming operations of other counters can be inserted between them.

Note: If the counters are programmed to read or write 2-byte counts, the program must not transfer control between writing the first and second byte to another routine that also reads or writes into the same counter. This will cause an incorrect count.

Registers

Figure 2-16. System Timer/Counter Registers

I/O Address (hex)	Register
0040	Read/Write Timer/Counter 0
0041	Read/Write Timer/Counter 1
0042	Read/Write Timer/Counter 2
0043	Write Control Byte for Counters 0, 1, or 2

Channel 0 (Hex 0040) and Channel 2 (Hex 0042) Count Registers

The control byte is written to address hex 0043, indicating the format of the count (least-significant byte only, most-significant byte only, or least-significant byte followed by most-significant byte). This procedure must be done before writing the count to I/O address hex 0040 for channel 0 or hex 0042 for channel 2.

Control Byte Register - Channel 0, 1, or 2 (Hex 0043)

This is a write-only register. Figure 2-17 through Figure 2-20 describe the format for the control byte (I/O address hex 0043) for counters 0, 1, and 2.

Figure 2-17. SC - Select Counter, I/O Address Hex 0043

Bit 7 SC1	Bit 6 SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Reserved

Figure 2-18. RW - Read/Write Counter, I/O Address Hex 0043

Bit 5 RW1	Bit 4 RW0	
0	0	Counter Latch Command
0	1	Read/Write Counter Bits 0 - 7 only
1	0	Read/Write Counter Bits 8 - 15 only
1	1	Read/Write Counter Bits 0 - 7 first, then Bits 8 - 15

Figure 2-19. M - Counter Mode, I/O Address Hex 0043

Bit 3 M2	Bit 2 M1	Bit 1 M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

Don't care bits (X) should be set to 0.

Figure 2-20. Binary Coded Decimal (BCD)

Bit 0 BCD	
0	Binary Counter 16 Bits
1	Binary Coded Decimal Counter (4 Decades)

Counter Latch Command

The Counter Latch command is written to the control byte register. The SC0 and SC1 bits select the counter, and bits 5 and 4 distinguish this command from a control byte. Figure 2-21 shows the format of the Counter Latch command.

Figure 2-21. Counter Latch Command

Bit	Function
7	SC1 - Specifies the counter to be latched
6	SC0 - Specifies the counter to be latched
5, 4	0 - Specifies the Counter Latch command
3 - 0	0 - Reserved

The count is latched into the selected counter output latch at the time the Counter Latch command is received. This count is held in the latch until it is read by the system microprocessor (or until the counter is reprogrammed). After the count is read by the system microprocessor, it is automatically unlatched and the output latch returns to following the counting element. Counter Latch commands do not affect the programmed mode of the counter in any way. All subsequent latch commands to a given counter, issued before the count is read, are ignored. A read cycle to the counter latch returns the value latched by the first Counter Latch command.

System Timer Modes

The following definitions are used when describing the timer modes.

CLK pulse The rising edge, then the falling edge of the CLK input to a counter.

Trigger The rising edge of the GATE input to a counter.

Counter load The transfer of a count from the counter register to the counting element.

Mode 0 - Interrupt on Terminal Count

Event counting can be done using mode 0. Counting is enabled when GATE is set to 1, and disabled when GATE is set to 0. If GATE is 1 when the control byte and initial count are written to the counter, the sequence for mode 0 is:

1. The control byte is written to the counter, and OUT goes low.
2. The initial count is written.
3. Initial count is loaded on the next CLK pulse. The count is not decremented by this CLK pulse.

   The count is decremented until the counter reaches 0. For an initial count of *n*, the counter reaches 0 after $n+1$ CLK pulses.
4. OUT goes high.

OUT remains high until a new count or new mode 0 control byte is written into the counter.

If GATE is equal to 0 when an initial count is written to the counter, it is loaded on the next CLK pulse even though counting is not enabled. After GATE enables counting, OUT goes high *n* CLK pulses later.

If a new count is written to a counter while counting, it is loaded on the next CLK pulse, and counting continues from the new count. If a 2-byte count is written to the counter, the following occurs:

1. The first byte written to the counter disables the counting. OUT goes low immediately and there is no delay for the CLK pulse.
2. When the second byte is written to the counter, the new count is loaded on the next CLK pulse. OUT goes high when the counter reaches 0.

Mode 1 - Hardware Retriggerable One-Shot

The counter is armed by writing the control byte and initial count to the counter. When a trigger occurs the counter is loaded. The sequence for mode 1 is:

1. OUT is high.
2. The control byte and initial count are written to the counter.
3. On the CLK pulse following a trigger, OUT goes low and begins the one-shot pulse.
4. The count is decremented until the counter reaches 0.
5. OUT goes high.

OUT remains high until the CLK pulse after the next trigger.

For an initial count of *n*, a one-shot pulse is *n* CLK pulses long. The one-shot pulse repeats the same count of *n* for the next triggers. OUT remains low *n* CLK pulses following any trigger. GATE does not affect OUT. The current one-shot pulse is not affected by a new count written to the counter unless the counter is retriggered. If the counter is retriggered, the new count is loaded and the one-shot pulse continues.

Note: Mode 1 is only valid on counter 2.

Mode 2 - Rate Generator

This mode causes the counter to perform a divide-by-*n* function. Counting is enabled when GATE is set to 1, and disabled when GATE is set to 0. The sequence for mode 2 is:

1. OUT is high.
2. The control byte and initial count are written to the counter.
3. Initial count is loaded on the next CLK pulse.
4. The count is decremented until the counter reaches 1.
5. OUT goes low for one CLK pulse.
6. OUT goes high.
7. The counter reloads the initial count.
8. The process is repeated.

If GATE goes low during the OUT pulse, OUT goes high. On the next CLK pulse, a trigger reloads the counter with the initial count. OUT

goes low *n* CLK pulses after the trigger. This allows the GATE input to be used to synchronize the counter.

OUT goes low *n* CLK pulses after writing the initial count. This allows software synchronization of the counter.

The current counting sequence is not affected by a new count being written to the counter. If the counter receives a trigger after a new count is written and before the end of the current count, the new count is loaded on the next CLK pulse and counting continues from the new count. If the trigger is not received by the counter, the new count is loaded following the current counting cycle.

Mode 3 - Square Wave

Mode 3 is similar to mode 2 except for the duty cycle of OUT. Counting is enabled when GATE is set to 1, and disabled when GATE is set to 0. An initial count of *n* results in OUT being a square wave. The period of the square wave is *n* CLK pulses. If OUT is low and GATE goes low, OUT goes high. On the next CLK pulse, a trigger reloads the counter with the initial count.

After writing a control byte and initial count, the counter is loaded on the next CLK pulse. This allows software synchronization of the counter.

The current counting sequence is not affected by a new count being written to the counter. If the counter receives a trigger after a new count is written, and before the end of the current count half-cycle of the square wave, the new count is loaded on the next CLK pulse, and counting continues from the new count. If the trigger is not received by the counter, the new count is loaded following the current half-cycle.

The implementation of mode 3 differs, depending on whether the count written is an odd or even number. If the count is even, the sequence for mode 3 is:

1. OUT is high.
2. The initial count is loaded on the first CLK pulse.
3. The count is decremented by 2 on succeeding CLK pulses.
4. The count is decremented until the counter reaches 0.
5. OUT changes state.
6. The counter is reloaded with the initial count.

7. The process repeats indefinitely.

If the count is odd, the sequence for mode 3 is:

1. OUT is high.
2. The initial count minus 1 is loaded on the first CLK pulse.
3. The count is decremented by 2 on succeeding CLK pulses.
4. The count is decremented until the counter reaches 0.
5. One CLK pulse after the count reaches 0, and OUT goes low.
6. The counter is reloaded with the initial count minus 1.
7. The count is decremented by 2 on succeeding CLK pulses.
8. The count is decremented until the counter reaches 0.
9. OUT goes high.
10. The process repeats indefinitely.

Mode 3, using an odd count, causes OUT to go high for a count of $(n+1)/2$ and low for a count of $(n-1)/2$.

Mode 3 can operate such that OUT is initially set low, when the control byte is written. For this condition, the sequence for mode 3 is:

1. OUT is low.
2. The count decrements to half of the initial count.
3. OUT goes high.
4. The count is decremented until the counter reaches 0.
5. OUT goes low.
6. The process repeats indefinitely.

This process results in a square wave with a period of *n* CLK pulses.

Note: If it is required that OUT be high after the control byte is written, the control byte must be written twice. This applies only to mode 3.

Mode 4 - Software Retriggerable Strobe

Counting is enabled when GATE is 1, and disabled when GATE is 0. Counting is triggered when an initial count is written. The sequence for mode 4 is:

1. OUT is high.
2. The control byte and initial count are written to the counter.
3. The initial count is loaded on the next CLK pulse. The count is not decremented by this clock pulse.
4. The count is decremented until the counter reaches 0. For an initial count of *n*, the counter reaches 0 after $n+1$ CLK pulses.
5. OUT goes low for one CLK pulse.
6. OUT goes high.

GATE should not go low one half CLK pulse before or after OUT goes low. If this occurs, OUT remains low until GATE transitions high.

If a new count is written to a counter while counting, it is loaded on the next CLK pulse. Counting then continues from the new count. If a 2-byte count is written, the following occurs:

1. The first byte written to the counter does not affect counting.
2. When the second byte is written to the counter, the new count is loaded on the next CLK pulse.

The mode 4 sequence can be retriggered by software. The period from when the new count of *n* is written to when OUT strobes low is $(n+1)$ pulses.

Mode 5 - Hardware Retriggerable Strobe

Counting is triggered by the rising edge of GATE. The sequence for mode 5 is:

1. OUT is high.
2. The control byte and initial count are written to the counter. Counting is triggered by the rising edge of GATE.
3. The counter is loaded on the next CLK pulse after the trigger. The count is not decremented by this CLK pulse.
4. The count is decremented until the counter reaches 0.
5. OUT goes low for one CLK pulse. This occurs $(n+1)$ CLK pulses after the trigger.

6. OUT goes high.

The counting sequence can be retriggered. OUT strobes low $(n+1)$ pulses after the trigger. GATE does not affect OUT.

The current counting sequence is not affected by a new count being written to the counter. If the counter receives a trigger after a new count is written and before the end of the current count, the new count is loaded on the next CLK pulse and counting continues from the new count.

Note: Mode 5 is valid only on counter 2.

Operations Common to All Modes

Control bytes written to a counter cause the control logic to reset. OUT goes to a known state. This does not take a CLK pulse.

The falling edge of the CLK pulse is when new counts are loaded and counters are decremented.

Counters do not stop when they reach 0. In modes 0, 1, 4, and 5, the counter wraps around to the highest count, and continues counting. Modes 2 and 3 are periodic; the counter reloads itself with the initial count and continues from there.

The GATE is sampled on the rising edge of the CLK pulse.

Figure 2-22 shows the minimum and maximum initial counts for the counters.

Figure 2-22. Minimum and Maximum Initial Counts, Counters 0 and 2

Mode	Min Count	Max Count
0	1	$0 = 2^{16}$ (Binary Counting) or 10^4 (BCD Counting)
1	1	$0 = 2^{16}$ (Binary Counting) or 10^4 (BCD Counting)
2	2	$0 = 2^{16}$ (Binary Counting) or 10^4 (BCD Counting)
3	2	$0 = 2^{16}$ (Binary Counting) or 10^4 (BCD Counting)
4	1	$0 = 2^{16}$ (Binary Counting) or 10^4 (BCD Counting)
5	1	$0 = 2^{16}$ (Binary Counting) or 10^4 (BCD Counting)

Speaker

The system board generates a signal to an amplifier driving the display speaker. The audio signal is carried through the power/audio cable connector from the system to the display.

The audio signal on the system board combines audio from three sources:

- System timer
- 2400 bps modem card
- Audio card.

The system timer circuit on the system board allows the speaker to be driven three different ways:

- A direct program control register bit can be toggled to generate a pulse train.
- The clock input to the timer can be modulated with a program-controlled I/O port bit.
- The output from channel 2 of the timer can be programmed to generate a waveform to the speaker.

Figure 2-23. Speaker Tone Generation

Channel 2	Tone
Gate 2	Controlled by I/O Port Bit 1
Clock In 2	1.190 MHz OSC
Clock Out 2	Used to drive speaker

All three methods can be performed simultaneously. (See "System Timers" on page 2-31.)