CHIPS PRELIMINARY

82C206
INTEGRATED PERIPHERALS CONTROLLER

- **100% Compatible to IBM™ PC AT**
- **Fully compatible to Intel™'s 8237 DMA controller, 8259 Interrupt controller, 8254 Timer/Counter, and Motorola™'s 146818 Real Time Clock**
- **Offers 7 DMA channels, 13 Interrupt request channels, 2 Timer/Counter channels, and a Real Time Clock**
- **Reduced recovery time (120 ns) between control**
- **114 bytes of CMOS RAM memory**
- **8 MHz DMA clock with programmable internal divider for 4 MHz operation**
- **Programmable wait states for the DMA cycle**
- **16 Mbytes DMA address space**
- **Single chip 84-pin CMOS implementation**

The 82C206 Integrated Peripheral Controller incorporates two 8237 DMA controllers, two 8259 Interrupt controllers, one 8254 Timer/Counter, one MC146818 Real Time Clock, 74LS612 memory mapper, in addition to several other TTL/SSI interface logic chips to offer a single chip integration of all the peripherals attached to the peripheral bus (X-Bus) in the IBM™ PC AT™ While offering a complete compatibility to the IBM PC AT architecture, the chip offers enhanced features and improved speed performance. Tnese include an additional 64 bytes of user RAM for the Real Time Clock, and drastically reduced recovery specifications for the 8237, 8259 and 8254. Variable wait state option is provided for the DMA cycles. Programmable delays are provided for the CPU access to the internal registers of the chip. The chip also provides an option to select 8 or 4 MHz system clock.

The 82C206, along with the CS8220 PC AT Compatible CHIPSet, provides a highly integrated high performance solution for a PC AT compatible implementation.

The 82C206 is implemented using advanced CMOS technology and is packaged in an 84-pin PLCC.

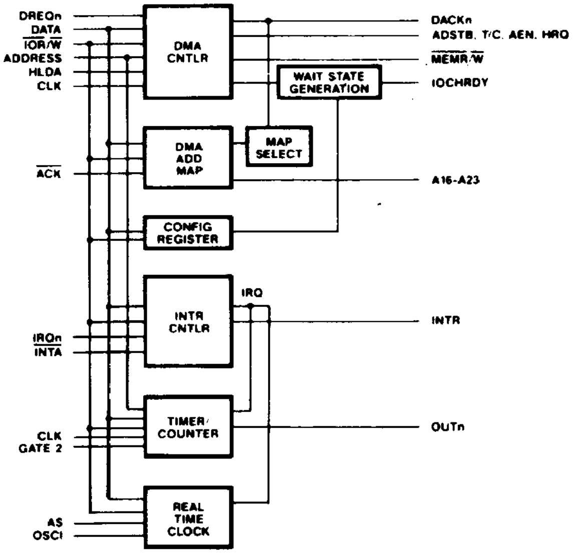
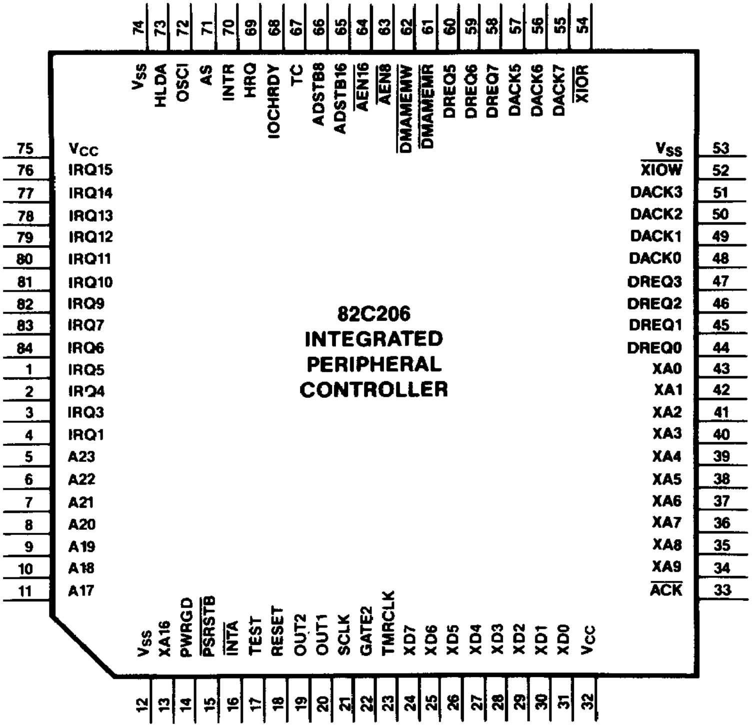


Figure 1. 82C206 Integrated Peripheral Controller Block Diagram


82C206
INTEGRATED
PERIPHERAL
CONTROLLER
74 Vss
73 HLDA
72 OSCI
71 AS
70 INTR
69 HRQ
68 IOCHRDY
67 TC
66 ADSTB8
65 ADSTB16
64 AEN16
63 AEN8
62 DMAMEMW
61 DMAMEMR
60 DREQ5
59 DREQ6
58 DREQ7
57 DACK5
56 DACK6
55 DACK7
54 XIOR
75 Vcc
76 IRQ15
77 IRQ14
78 IRQ13
79 IRQ12
80 IRQ11
81 IRQ10
82 IRQ9
83 IRQ7
84 IRQ6
1 IRQ5
2 IRQ4
3 IRQ3
4 IRQ1
5 A23
6 A22
7 A21
8 A20
9 A19
10 A18
11 A17
53 Vss
52 XIOW
51 DACK3
50 DACK2
49 DACK1
48 DACK0
47 DREQ3
46 DREQ2
45 DREQ1
44 DREQ0
43 XA0
42 XA1
41 XA2
40 XA3
39 XA4
38 XA5
37 XA6
36 XA7
35 XA8
34 XA9
33 ACK
12 Vss
13 XA16
14 PWRGD
15 PSRSTB
16 INTA
17 TEST
18 RESET
19 OUT2
20 OUT1
21 SCLK
22 GATE2
23 TMRCLK
24 XD7
25 XD6
26 XD5
27 XD4
28 XD3
29 XD2
30 XD1
31 XD0
32 Vcc

82C206 Pin Description

Pin No.	Name	Type	Function
24-31	XD7-XD0	I/O	DATA BUS: The Data Bus lines are 3-state bi-directional lines connected to the system data bus (XD bus in a PC/AT design). The outputs are enabled in the program condition during the I/O Read to output the contents of the DMA controller registers (Address register, Status register, the Temporary register or a Word Count register), the three Interrupt Controller registers (Interrupt Request register, In Service register and the Interrupt Mask register), the Timer/Counters registers (namely the contents of these counters or states of the counters), the Real Time Clocks internal registers and page registers of memory mapper. During an I/O write cycle, the outputs are disabled and the CPU can program the DMA Controller registers, the Interrupt Controller registers, the Timer/Counters registers, the DMA Page register and the Real Time Clock registers and internal RAM. During DMA cycles, the most significant 8 bits of the address are output onto the data bus to be strobed into an external latch by ADSTB8 or ADSTB16. During memory-to-memory operations, data from the memory comes into the DMA Controller on the data bus during read from the memory. In the write to memory transfer, the data bus outputs place the data into the new memory location. During the interrupt sequence, the interrupt controllers output the interrupt vector byte on the data bus. Data bus XD7-XD0 also acts as the multiplexed address/data bus for the Real Time Clock.

82C206 Pin Description (Continued)

Pin No.	Name	Type	Function
35-43 34	XA8-XA0 XA9	I/O I	ADDRESS BUS: This is the system address bus used to address various registers of the 82C206. It is tied to the external bus (XA bus) in a PC/AT compatible design. During a non-DMA cycle, A3-A0 act as inputs and are used by the CPU to address the registers of the DMA Controller corresponding to DMA channels 0-3 and A4-A1 address the registers of the DMA Controller corresponding to DMA channels 5-7. In the active DMA cycle, A7-A0 are outputs and carry address information for DMA channels 0-3. Correspondingly, A8-A1 are address outputs for 16 bit DMA channels 5-7. During program condition, A9-A0 are used to address configuration register and the internal registers of DMA Controller, INT Controller, Timer, RTC and Memory Mapper.
18	RESET	I	RESET: Reset is active high input which affects the following registers: DMA Controllers: Clears the Command, Status, DMA Request, Temporary register, First/Last flip-flop; sets the Mask register. Following reset, the DMA controller is in an idle state. INTERRUPT Controllers: Clears the edge sense circuit, the interrupt mask register, all ICW4 functions, IRQ0 is assigned highest priority, slave address is set to 7, special mask mode is disabled and status read is set to IRR.
54	$\overline{\text{XIOR}}$	I/O	I/O READ: I/O Read is a bidirectional active low three-state line. In an idle cycle (non DMA, non-interrupt), it is an input control signal used by the CPU to read the 82C206 internal registers. In an active DMA cycle, it is an output control signal used by the DMA Controller to access data from a peripheral during a DMA write transfer.
52	$\overline{\text{XIOW}}$	I/O	I/O WRITE: I/O Write is a bidirectional active low three-state signal. In an idle cycle (non-DMA, non-interrupt) it is an input control signal used by the CPU to load information to the 82C206 internal registers. In an active DMA cycle, it is an output control signal used by the DMA Controller to load data to the peripheral during a DMA Read transfer.
61	$\overline{\text{DMAMEMR}}$	O	DMA MEMORY READ: $\overline{\text{DMAMEMR}}$ is an active low three-state output used to access data from the selected memory location during DMA Read or memory-to-memory transfer.

82C206 Pin Description (Continued)

Pin No.	Name	Type	Function
62	DMAMEMW	O	DMA MEMORY WRITE: DMAMEMW is an active low three-state output used to write data to the selected memory location during DMA write or a memory-to-memory transfer. In a PC/AT compatible design, this signal is connected to XMEMW.
21	SCLK	I	CLOCK INPUT: The Clock Input is used to generate the timing signals which control DMA operations. This input may be driven from DC to 10 MHz. The Clock may be stopped in either state for standby operation. The internal clock used for DMAC is either the SCLK or SCLK/2 depending on the setting of DMA CLOCK SELECT bit in the configuration register.
68	IOCHRDY	I/O	I/O CHANNEL READY: In the input mode, a low on IOCHRDY causes the internal DMA ready signal to go low asynchronously. When IOCHRDY goes high, one DMA Clock cycle will elapse before internal DMA Ready goes up. This signal is used to extend memory read and write pulses for the DMA controllers to accommodate slow memories or I/O devices. IOCHRDY must satisfy set-up and hold times with respect to DMACLK in order to work reliably. In the output mode, this pin is an open drain output and provides an active low output whenever any 82C206 register is addressed for read or write. This output will remain low for a programmed number of DMACLK cycles (as configured by bits 6 and 7 of 82C206 configuration register) and then goes high, if pulled up by an external register. IOCHRDY provides a means of introducing a programmed number of wait-states (as counted by DMACLK cycles) for I/O read/write cycles to 82C206. In a PC/AT architecture based design this pin should be wire-ored to PC/AT's IOCHRDY signal.
73	HLDA	I	HOLD ACKNOWLEDGE: The active high Hold Acknowledge from the CPU indicates that it has relinquished control of the system busses.

82C206 Pin Description (Continued)

Pin No.	Name	Type	Function
44- 47 60- 58	DREQ0- DREQ3 DREQ5- DREQ7	I	DMA REQUEST: The DMA Request (DREQ) are individual asynchronous channel request inputs used by peripheral circuits to obtain DMA service. In Fixed Priority, DREQ0 has the highest priority and DREQ7 has the lowest priority. A request is generated by activating the DREQ line of a channel. DACK will acknowledge the recognition of DREQ signal. Polarity of DREQ is programmable. Reset initializes these lines to active high. DREQ must be maintained until the corresponding DACK goes active. DREQ will not be recognized while the clock is stopped. Unused DREQ inputs should be pulled High or Low (inactive) and the corresponding mask bit set. DREQ0-DREQ3 support 8-bit transfers between 8-bit I/O and 8 or 16-bit system memory. DREQ5-DREQ7 support 16-bit data transfers between 16-bit peripheral and 16-bit system memory. DREQ4 is not available as it is used to cascade DREQ0-DREQ3.
67	TC	O	TERMINAL COUNT: Terminal Count (TC) is an active high signal. Information concerning the completion of DMA services is available at the TC output pin. A pulse is generated by the DMA Controller when terminal count (TC) for any channel is reached, except for channel 0 in memory-to-memory mode. During memory-to-memory transfers TC will be output when the TC for channel 1 occurs. When a TC pulse occurs, the DMA Controller will terminate the service, and if auto-initialize is enabled, the base registers will be written to the current registers of that channel. The mask bit and TC bit in the status word will be set for the currently active channel unless the channel is programmed for auto-initialize. In that case, the mask bit remains clear.

82C206 Pin Description (Continued)

Pin No.	Name	Type	Function
69	HRQ	O	HOLD REQUEST: The Hold Request (HRQ) output is used to request control of the system bus. When a DREQ occurs and the corresponding mask bit is clear, or a software DMA request is made, the DMA Controller issues HRQ. The HLDA signal then informs the controller when access to the system busses is permitted. For stand-alone operation where the DMA Controller always controls the busses, HRQ may be tied to HLDA. This will result in one SO state before the transfer.
48- 51 57- 55	DACK0- DACK3 DACK5- DACK7	O	DMA ACKNOWLEDGE: DMA Acknowledge is used to notify the individual peripherals when one has been granted a DMA cycle. The active polarity of these lines is programmable. Reset initializes them to active low. Because these signals are used internally for cascading the DMA channels and for DMA page register selection, these signals must be programmed to be active low.
66	ADSTB8	O	ADDRESS STROBE: This is an active high signal used to control latching of the upper address byte (A8-A15) for 8-bit peripherals. It will drive directly the strobe input of external transparent octal latches. During block operations, ADSTB8 will only be issued when the upper address byte must be updated, thus speeding operation through elimination of S1 states. ADSTB8 is active for DMA channels 0-3.
64	$\overline{\text{AEN16}}$	0	ADDRESS ENABLE for 16-BIT DMA TRANSFERS: This signal enables the 8-bit latch containing the upper 8 address bits (A9-A16) on to system address bus. It is inactive when external bus master controls the system bus (MASTER=0). This signal is active low.
65	ADSTB16	O	ADDRESS STROBE for 16-BIT TRANSFERS (channels 5-7). This is an active high signal used to control latching of the upper address byte A9-A16 for 16-bit DMA transfers. Its function is just like ADSTB8.
63	$\overline{\text{AEN8}}$	O	ADDRESS ENABLE for 8-BIT DMA TRANSFERS: This signal is the output enable for the 8-bit latch containing upper 8 address bits (A8-A15). It enables A8-A15 system address bus. It is inactive when external bus master controls the system bus (MASTER=0). This signal is active low.

82C206 Pin Description (Continued)

Pin No.	Name	Type	Function
33	$\overline{\text{ACK}}$ (MSE)	I	MODULE SELECT ENABLE: When high, it enables the chip select function on one of the modules (DMA Controller, INT Controler, TIMER, RTC, DMA Page register or the Configuration register) for the programming function, i.e. CPU read or write of the command, status or other register of various modules of the 82C206. When low, the 82C206 is essentially disconnected from the system bus. The 82C206 at this time could be performing an active DMA or an interrupt cycle. In a PC/AT compatible design, this pin is tied to $\overline{\text{ACK}}$ signal.
11-5 13	A23-A17 XA16	O O	DMA PAGE REGISTER ADDRESS: XA16 and A17-A23 are 3-state output pins. XA16 is the least significant bit of the DMA page register and is used for DMA transfers for 8-bit peripherals only (channel 0-3). XA16 is not used for DMA transfers to 16-bit peripherals (channel 5-7) as XA9-XA16 is provided by demultiplexing the data bus. A17-A23 are the upper 7 bits of the DMA page register.
76-82 83, 84 1-3 4	IRQ15-IRQ9 IRQ7, IRQ6 IRQ5-IRQ3 IRQ1	I I	INTERRUPT REQUESTS: Asynchronous inputs. An interrupt request is executed by raising an IRQ input low to high and holding it high until it is acknowledged (edge triggered mode) or just a high level on an IRQ input (level triggered mode).
16	$\overline{\text{INTA}}$	I	INTERRUPT ACKNOWLEDGE: This pin is used to enable the interrupt controllers interrupt vector data on to the data bus by a sequence of interrupt acknowledge pulses issued by the CPU.
70	INTR	O	INTERRUPT: This pin goes high whenever a valid interrupt request is asserted. It is used to interrupt the CPU, and is usually connected to the CPU's interrupt pin.
23	TMRCLK	I	TIMER CLOCK: Clock input for Counter 0, Counter 1 and Counter 2.
22	GATE2	I	GATE 2: Gate input for Counter 2. In a PC/AT compatible design, the Counter 2 is used for tone generation for speaker. In this design, the GATE 2 input is driven by Bit 0 of I/O Port 61H (called TIM2GATE SPK).
20	OUT1	O	OUT 1: Output of Timer 1. In a PC/AT compatible design, Timer 1 is programmed as a rate generation to produce 15 μsec period signal used for interrupt request for refresh cycles.
19	OUT 2	O	OUT 2: Output of Timer 2. In a PC/AT compatible design, OUT 2 is used to drive the speaker.

82C206 Pin Description (Continued)

Pin No.	Name	Type	Function
71	AS	I	ADDRESS STROBE: Address strobe is a positive pulse whose falling edge latches the address from the XD bus.
72	OSCI	I	OSCILLATOR INPUT: The time base for the time functions is connected to this pin. External square waves of 32.768 KHz may be connected to this input.
15	$\overline{\text{PSRSTB}}$	I	This input is used to establish the condition of the control registers when power is applied to the device. In a PC/AT compatible design, this pin should be tied to the battery back-up circuit. When $\overline{\text{PSRSTB}}$ and TEST are both low, the following occurs: (a) Periodic Interrupt Enable (PIE) bit is cleared to zero. (b) Alarm Interrupt Enable (AIE) bit is cleared to zero. (c) Update ended Interrupt Enable (UIE) bit is cleared to zero. (d) Update ended Interrupt Flag (UF) bit is cleared to zero. (e) Interrupt Request status Flag (IRQF) is cleared to zero. (f) Periodic Interrupt Flag (PF) bit is cleared to zero. (g) The part is not accessible. (h) Alarm interrupt Flag (AF) bit is cleared to zero. (i) Square Wave output enable list is cleared to zero.
14	PWRGD	I	PWRGOOD: The Power Good pin must be high for bus cycles in which the CPU accesses the RTC. When PWRGD is low, all address, data, data strobe and R/W pins are disconnected from the processor.
17	TEST	I	TEST: Test is an active high input. It initializes various internal registers so that the test program starts in a known state. It should be tied low for normal operation.
32, 75	V_{CC}	—	Power Supply
12, 53, 74	V_{SS}	—	Ground

82C206—INTEGRATED PERIPHERALS CONTROLLER

The 82C206 is a LSI implementation of the standard peripherals required to implement an IBM PC/AT system board. This device contains the equivalent of two 8237A DMA Controllers, a 74LS612 Mapper, two 8259A Interrupt Controllers, an 8254 Counter/Timer, and a MC146818 Real Time Clock with RAM. The 82C206 provides all of the standard peripherals required for a system board implementation except the keyboard interface controller. Figure 1 illustrates the subsystems contained within the 82C206.

Two DMA Controllers are provided and connected in such a way as to provide the user with four channels of DMA (DMA1) for 8-bit transfers and three channels of DMA (DMA2) for 16-bit transfers (the first 16-bit DMA channel is used for cascading). Included as part of the DMA subsystem is the Page Register (DMAPAGE) device which is used to supplement the DMA and drive the upper address lines when required.

Sixteen channels of interrupt are provided in the 82C206. These channels are partitioned into two cascaded controllers (INTC1,INTC2) with 8 inputs each. Of these 16 channels, three are connected internally to various devices, allowing 13 user definable channels of interrupt. The three internally connected channels are as follows:

Channel 0 — Counter/Timer Counter 0 Interrupt
Channel 2 — Cascade to Slave Interrupt Controller (INTC2)
Channel 8 — Real Time Clock Interrupt

The remaining 13 channels may be defined and utilized as necessary to meet the users specific system requirements.

A Counter/Timer (CTC) subsystem is provided which contains three independent counters. All three counters are driven from a clock input pin which is independent from the other clock inputs to the device. Counter 0 is connected to Interrupt 0 of INTC1. It is intended to be used as a multi-level interrupt to the system for such tasks as time keeping and task switching. Counter 1 may be programmed to generate pulses or square waves for use by external devices. The third channel (Counter 2) is a full function Counter/Timer which has a gate input for controlling the internal counter. This channel can be used as an interval counter, a timer, or as a gated rate/pulse generator.

A Real Time Clock (RTC) is included in the 82C206 for maintaining the time and date. This subsystem also contains 114 bytes of RAM in addition to the Clock/Calendar. The Clock/Calendar information and RAM are kept active by connecting the device to an external battery when system power is turned off.

To interconnect and control all of these major subsystems a top level control section is employed which is divided into subsystems for purposes of discussion.

The first section is the Clock and Wait State Control section. This subsystem controls the generation of DMA wait states and the negation of IOCHRDY (if programmed to do so) during CPU access of the device. The last subsystem is the Top Level Decode.

In order to accommodate over 200 registers in the 82C206 and maintain I/O decode compatibility with the IBM PC/AT, a multilevel decode scheme is employed. The Top Level Decode subsystem performs the function of generating enables to the various subsystems. Control and direction of the XD0-XD7 data bus buffers are also handled by this subsystem.

Each of these subsystems will now be described separately.

Top Level Decode

The 82C206 Top Level Decode provides 8 separate enables to various subsystems of the device. Figure 2 contains a truth table for the Top Level Decoder. The enabling of the 82C206 XD0-XD7 output buffers is also controlled by this section. The output buffers are enabled whenever an enable is generated to an internal subsystem *and* the $\overline{\text{XIOR}}$ signal is asserted.

ACK	XA9	XA8	XA7	XA6	XA5	XA4	XA3	XA2	XA1	XA0	ADDRESS RANGE(Hex)	SELECTED DEVICE
1	0	0	0	0	0	0	X	X	X	X	000-00F	DMA1
1	0	0	0	0	1	0	0	0	0	X	020-021	INTC1
1	0	0	0	0	1	0	0	0	1	X	022-023	CONFIG
1	0	0	0	1	0	0	0	0	X	X	040-043	CTC
1	0	0	0	1	1	1	0	0	0	1	071	RTC
1	0	0	1	0	0	0	X	X	X	X	080-08F	DMAPAGE
1	0	0	1	0	1	0	0	0	0	X	0A0-0A1	INTC2
1	0	0	1	1	0	X	X	X	X	X	0C0-0DF	DMA2
0	X	X	X	X	X	X	X	X	X	X	DISABLED	
X	1	X	X	X	X	X	X	X	X	X	DISABLED	
X	X	1	X	X	X	X	X	X	X	X	DISABLED	

Figure 2. 82C206 Internal Decode

The decoder is enabled by three signals. These three signals are ACK, XA9 and XA8. To enable any internal device ACK must be "1" and both XA9 and XA8 must be "0".

The decode scheme employed in the 82C206 is designed to comply with the IBM PC/AT requirements and is more fully decoded. If the user wishes to take advantage of the areas which are unused by inserting additional peripherals in the I/O map, he may do so since the subsystems in the 82C206 will not respond to the unused address spaces established by the Top Level Decoder. The extra peripherals may be tied directly to the XD0-XD7 data lines since the 82C206 output buffers are not enabled unless an internal subsystem is enabled.

Clock and Wait State Control

The Clock and Wait State Control subsystem performs four functions, control of the DMA command width, control of the CPU read or write cycle length, and selection of the DMA clock rate. All of these functions are user selectable by writing to the Configuration Register located at address 023H.

Writing and reading this register is accomplished by first writing a 01H to location 022H to select the 82C206 Configuration Register, and then performing either a read or write to location 023H.

Configuration Register (023H) (Index 01H)

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
RW1	RW0	16W1	16W0	8W1	8W0	EMR	CLK

RW1-RW0—When the higher speed CPU's are accessing the 82C206, the cycle can be extended by programming up to four wait states into the Configuration Register. This will cause the 82C206 to assert a not ready condition on IOCHRDY (low) whenever a valid decode from the Top Level Decoder is detected and either XIOR or XIOW is asserted. IOCHRDY will remain low for the number of wait states programmed into the Configuration Register bits 6 and 7.

RW1	RW0	Read/Write Cycle Wait States
0	0	1
0	1	2
1	0	3
1	1	4

Wait states are in increments of one SCLK cycle and are not affected by the DMA Clock Divider.

16W1-16W0—Wait states can be independently controlled for both 8-bit and 16-bit DMA cycles. This allows the user to tailor the DMA cycle more closely to the application.

16W1	16W0	16-Bit DMA Wait States
0	0	1
0	1	2
1	0	3
1	1	4

8W1-8W0—Wait states may be inserted in 8-bit DMA cycles by programming these two bits in the Configuration Register.

8W1	8W0	8-Bit DMA Wait States
0	0	1
0	1	2
1	0	3
1	1	4

Further control of the cycle length is available through the use of the IOCHRDY pin on the 82C206. During DMA this pin is used as an input to the wait state generation logic to extend the cycle if necessary. This input is driven low (0) by the peripheral to extend the cycle. The cycle can then be completed by releasing IOCHRDY and allowing it to return high (1).

EMR—This bit enables the extended $\overline{\text{DMAMEMR}}$ function. Normally the assertion of $\overline{\text{DMAMEMR}}$ is delayed one clock cycle later than $\overline{\text{XIOR}}$ in the IBM PC/AT implementation. This may not be desirable in some systems. A "1" programmed into this bit position will start $\overline{\text{DMAMEMR}}$ at the same time as $\overline{\text{XIOR}}$.

CLK—This bit allows the user to insert a divider between the DMA Controller subsystems and the SCLK input pin, or connect the two directly. When this bit position contains a "0", the SCLK input is divided by two and is used to drive both the 8-bit and 16-bit DMA subsystems. A "1" in this position bypasses the divider and uses the SCLK input directly. Whenever the state of this bit is changed, an internal synchronizer controls the actual switching of the clock to prevent a short clock pulse from causing a DMA malfunction.

The Configuration Register contents are preloaded by RESET to an initial value of 0C0 hex. This value establishes a default which is IBM PC/AT compatible and corresponds to:

Read/Write cycles	—4 wait states
16-bit DMA transfers	—1 wait state
8-bit DMA transfers	—1 wait state

$\overline{\text{DMAMEMR}}$ delayed 1 DMA clock cycle later than $\overline{\text{XIOR}}$

DMA clock is equal to SCLK/2

DMA FUNCTIONAL DESCRIPTION

The equivalent of two 8237A DMA Controllers is implemented in the 82C206. Each controller is a four channel DMA device which will generate the memory addresses and control signals necessary to transfer information between a peripheral device and memory directly. This allows high speed information transfer with little CPU intervention.

The two DMA Controllers are internally cascaded to provide four DMA channels for transfers to 8-bit peripherals (DMA1), and three channels for transfers to 16-bit peripherals (DMA2). DMA2 Channel 0 privides the cascade interconnection of the two DMA devices, thereby maintaining IBM PC/AT compatibility.

DMA cycle length control is provided internally in the 82C206 allowing independent control for both 8-bit and 16-bit cycles. This is done through the programmable registers which can extend command signals or insert wait states.

Each DMA Channel has a pair of 16-bit counters and a reload register for each counter. The 16-bit counters allow the DMA to transfer blocks as large as 65536 words. The register associated with each counter allows the channel to reinitialize without reprogramming.

From this point on the description of the DMA subsystem pertains to both DMA1 and DMA2 unless otherwise noted.

DMA Operation

During normal operation of the 82C206, the DMA subsystem will be in either the Idle condition, the Program condition or the Active condition. In the Idle condition the DMA controller will be executing cycles consisting of only one state. The idle state SI is the default condition and the DMA will remain in this condition unless the device has been initalized and one of the DMA requests is active or the CPU attempts to access one of the internal registers.

When a DMA request becomes active the device enters the Active condition and issues a hold request to the system. Once in the Active condition the 82C206 will generate the necessary memory addresses and command signals to accomplish a memory-to I/O, I/O-to-memory, or a memory-to-memory transfer. Memory-to-I/O and I/O-to-memory transfers take place in one cycle while memory-to-memory transfers require two cycles. During transfers between memory and I/O, data is presented on the system bus by either memory or the requesting device and the transfer is completed in one cycle. Memory-to-memory transfers however, require the DMA to store data from the read operation in an internal register. The contents of this register is then written to memory on the subsequent cycle.

During transfers between memory and I/O, two commands are activated during the same cycle. In the case of a memory-to-I/O transfer, the 82C206 will assert both DMAMEMR and XIOW allowing data to be transfered directly to the requesting device from memory. Note that 82C206 does not latch data from nor drive data out on this type of cycle.

The number of clock cycles required to transfer a word of data may be varied by programing the DMA or, optionally extended by the peripheral device. During an Active cycle the DMA will sequence through a series of states. Each state will be one DMA clock cycle in length and the number of states in a cycle will vary depending on how the device is programmed and what type of cycle is being performed. The states are labeled S0–S4 and will be explained in detail in the section entitled Active Condition.

Idle Condition

When no device is requesting service the DMA is in an Idle condition which maintains the state machine in the SI state. During this time the 82C206 will sample the DREQ input pins every clock cycle. The internal select from the top level decoder and HLDA are also sampled at the same time to determine if the CPU is attempting to access the internal registers. When either of the above two situations occurs, the DMA will exit the Idle condition. Note that the Program condition has priorit over the Active condition since a CPU cycle has already started.

Program Condition

The Program condition is entered whenever HLDA is inactive and an internal select is active. The internal select is derived from the top level decode described previously. During this time address lines XA0-XA3 become inputs if DMA1 is selected, or XA1-XA4 become inputs if DMA2 is selected. Note, when DMA2 is selected XA0 is ignored. These address inputs are used to select the DMA controller registers which are to be read or written. Figure 3 lists the register address assignment. Due to the large number of internal registers in the DMA subsystem, an internal flip-flop is used to supplement the addressing of the count and address registers. This bit is used to select between the high and low bytes of these registers. The flip-flop will toggle each time a read or write occurs to any of the word count or address registers in the DMA. This internal flip-flop will be cleared by hardware RESET or a Master Clear command and may be set or cleared by the CPU issuing the appropriate command.

Special commands are supported by the DMA subsystem in the Program condition to control the device. These commands do not make use of the data bus but are derived from a set of addresses, the internal select and XIOW or

ADDRESS		OPERATION			
DMA1	DMA2	$\overline{\text{XIOR}}$	$\overline{\text{XIOW}}$	Flip-Flop	Register Function
000h	0C0h	0	1	0	Read Channel 0 Current Address Low Byte
		0	1	1	Read Channel 0 Current Address High Byte
		1	0	0	Write Channel 0 Base and Current Address Low Byte
		1	0	1	Write Channel 0 Base and Current Address High Byte
001h	0C2h	0	1	0	Read Channel 0 Current Word Count Low Byte
		0	1	1	Read Channel 0 Current Word Count High Byte
		1	0	0	Write Channel 0 Base and Current Word Count Low Byte
		1	0	1	Write Channel 0 Base and Current Word Count High Byte
002h	0C4h	0	1	0	Read Channel 1 Current Address Low Byte
		0	1	1	Read Channel 1 Current Address High Byte
		1	0	0	Write Channel 1 Base and Current Address Low Byte
		1	0	1	Write Channel 1 Base and Current Address High Byte
003h	0C6h	0	1	0	Read Channel 1 Current Word Count Low Byte
		0	1	1	Read Channel 1 Current Word Count High Byte
		1	0	0	Write Channel 1 Base and Current Word Count Low Byte
		1	0	1	Write Channel 1 Base and Current Word Count High Byte
004h	0C8h	0	1	0	Read Channel 2 Current Address Low Byte
		0	1	1	Read Channel 2 Current Address High Byte
		1	0	0	Write Channel 2 Base and Current Address Low Byte
		1	0	1	Write Channel 2 Base and Current Address High Byte
005h	0CAh	0	1	0	Read Channel 2 Current Word Count Low Byte
		0	1	1	Read Channel 2 Current Word Count High Byte
		1	0	0	Write Channel 2 Base and Current Word Count Low Byte
		1	0	1	Write Channel 2 Base and Current Word Count High Byte
006h	0CCh	0	1	0	Read Channel 3 Current Address Low Byte
		0	1	1	Read Channel 3 Current Address High Byte
		1	0	0	Write Channel 3 Base and Current Address Low Byte
		1	0	1	Write Channel 3 Base and Current Address High Byte
007h	0CEh	0	1	0	Read Channel 3 Current Word Count Low Byte
		0	1	1	Read Channel 3 Current Word Count High Byte
		1	0	0	Write Channel 3 Base and Current Word Count Low Byte
		1	0	1	Write Channel 3 Base and Current Word Count High Byte
008h	0D0h	0	1	X	Read Status Register
		1	0	X	Write Command Register
009h	0D2h	0	1	X	Read DMA Request Register
		1	0	X	Write DMA Request Register
00Ah	0D4h	0	1	X	Read Command Register
		1	0	X	Write Single Bit DMA Request Mask Register
00Bh	0D6h	0	1	X	Read Mode Register
		1	0		Write Mode Register
00Ch	0D8h	0	1	X	Set Byte Pointer Flip-Flop
		1	0	X	Clear Byte Pointer Flip-Flop
00Dh	0DAh	0	1	X	Read Temporary Register
		1	0	X	Master Clear
00Eh	0DCh	0	1	X	Clear Mode Register Counter
		1	0	X	Clear All DMA Request Mask Register Bits
00Fh	0DEh	0	1	X	Read All DMA Request Mask Register Bits
		1	0	X	Write All DMA Request Mask Register Bits

Figure 3. DMA Registers

$\overline{\text{XIOR}}$. These commands are Master Clear, Clear Mask Register, Clear Mode Register Counter, Set and Clear Byte Pointer Flip-Flop.

The 82C206 will enable programming whenever HLDA has been inactive for one DMA clock cycle. It is the responsibility of the system to ensure that programming and HLDA are mutually exclusive. Erratic operation of the 82C206 can occur if a request for service occurs on an unmasked channel which is being programmed. The channel should be masked or the DMA disabled to prevent the 82C206 from attempting to service a device with a channel which is partially programmed.

Active Condition

The 82C206 DMA subsystem enters the Active condition whenever a software request occurs or a DMA request on an unmasked channe! occurs and the device is not in the Program condition. The 82C206 will then begin a DMA transfer cycle.

In a read cycle for example, after receiving a DREQ, the 82C206 will issue a HRQ to the system. Until a HLDA is returned the DMA will remain in an idle condition. On the next clock cycle the DMA will exit Idle and enter state S0. During S0 the device will resolve priority and issue DACK on the highest priority channel requesting service. The DMA will then proceed to state S1 where the multiplexed addresses are output and latched. State S2 is then entered, at which time the 82C206 will assert $\overline{\text{DMAMEMR}}$. The device then transitions into S3 where the $\overline{\text{XIOW}}$ command is asserted. The 82C206 DMA will then remain in S3 until the Wait State Counter has decremented to zero *and* IOCHRDY is true. Note that at least one additional S3 will occur unless Compressed Timing is selected. Once a ready condition is detected, the DMA will enter S4 where both commands are deasserted. In Burst Mode and Demand Mode (discussed below), subsequent cycles will begin in S2 unless the intermediate addresses require updating. In these subsequent cycles the lower addresses are changed in S2.

The DMA can be programmed on a channel by channel basis to operate in one of four modes. The four modes are listed below.

Single Transfer Mode—This mode directs the DMA to execute only one transfer cycle at a time. DREQ must be held active until DACK becomes active. If DREQ is held active throughout the cycle, the 82C206 will deassert HRQ and release the bus once the transfer is complete. After HLDA has gone inactive the 82C206 will again assert HRQ and execute another cycle on the same channel unless a request from a higher priority channel has been received. In this mode the CPU is ensured of being allowed to execute at least one bus cycle between transfers.

Following each transfer the word count is decremented and the address is incremented or decremented. When the word count decrements from 0000h to FFFFh the terminal count bit in the status register is set and a T/C pulse is generated. If the autoinitialization option has been enabled, the channel will reinitialize itself. If Autoinitialize is not selected the DMA will set the DMA request bit mask and suspend transfering on that channel.

Block Transfer Mode—When Block Transfer Mode is selected, the 82C206 will begin transfers in response to either a DREQ or a software request and will continue until a terminal count (FFFFh) is reached, at which time T/C is pulsed and the status register terminal count bit is set. In this mode DREQ need only be held active until DACK is asserted. Autoinitialization is optional in this mode also.

Demand Transfer Mode—In Demand Transfer mode the DMA will begin transfers in response to the assertion of DREQ and will continue until either terminal count is reached or DREQ becomes inactive. This mode is normally used for peripherals which have limited buffering ability. The peripheral can initiate a transfer and continue until its buffer capacity is exhausted. The peripheral may then re-establish service by again asserting DREQ. During idle periods between transfers the CPU is released to operate and can monitor the operation by reading intermediate values from the address and word count registers. Once DREQ has been deasserted, higher priority channels are allowed to intervene. Reaching terminal count will result in the generation of a T/C pulse, the setting of the terminal count bit in the status register and autoinitialization (if enabled).

Cascade Mode—This mode is used to interconnect more than one DMA controller, to extend the number of DMA channels while preserving the priority chain. In Cascade mode the master DMA controller does not generate address or control signals. The DREQ and DACK signals of the master are used to interface the HRQ and HLDA signals of the slave DMA devices. Once the master has received a HLDA from the CPU in response to a DREQ caused by the HRQ from a slave DMA controller, the master DMA controller will ignore all inputs except HLDA from the CPU and DREQ on the active channel. This prevents conflicts between the DMA devices.

Figure 4 shows the cascade interconnection for two levels of DMA devices. Note that Channel 0 of DMA2 is internally connected for Cascade mode to DMA1. Additional devices can be cascaded to the available channels in either DMA1 or DMA2 since cascade is not limited to two levels of DMA controllers.

When programming cascaded controllers, begin with the device which is actually generating HRQ to the system (first level device) and then proceed to the second level devices. RESET causes the DACK outputs to become active low and are placed in the inactive state. To allow the internal cascade between DMA1 and DMA2 to function correctly, the active low state of DACK should not be modified. This is because the 82C206 has an inverter between DACK0 of DMA2 and HLDA of DMA1. The first level device's DMA request mask bits will prevent second level cascaded devices from generating unwanted hold requests during the initialization process.

DMA Transfers

Four types of transfer modes are provided in the 82C206 DMA subsystem. These transfer types are:

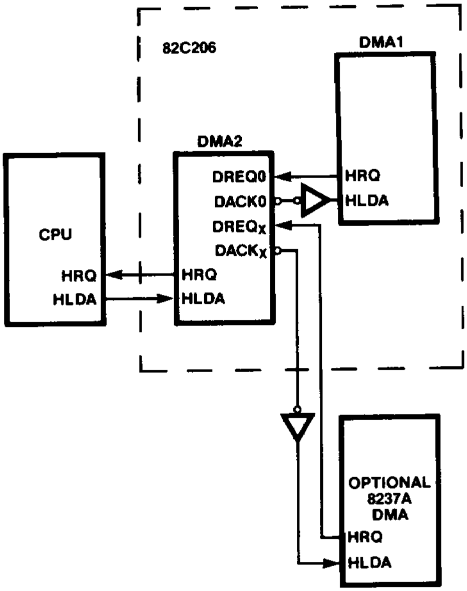


Figure 4. Cascade Mode Interconnect

Read Transfer—Read transfers move data from memory to an I/O device, by generating the memory address and asserting $\overline{\text{DMAMEMR}}$ and $\overline{\text{XIOW}}$ during the same cycle.

Write Transfer—Write transfers move data from an I/O device to memory by generating the memory address and asserting $\overline{\text{XIOR}}$ and $\overline{\text{DMAMEMW}}$.

Memory-to-Memory Transfer—The memory-to-memory transfer is used to move a block of memory from one location in memory to another. DMA channels 0 and 1 may be programmed to operate as memory-to-memory channels by setting a bit in the Command Register. Once programmed to perform a memory-to-memory transfer the process can be started by generating either a software or an external request to channel 0. Once the transfer is initiated, Channel 0 provides the address for the source block during the memory read portion of the cycle. Channel 1 generates the address for the memory write cycle. During the read cycle, a byte of data is latched in the internal Temporary Register of the 82C206. The contents of this register are then output on the XD0-7 data lines during the write portion of the cycle and subsequently written to memory. Channel 0 may be programmed to maintain the same source address on every cycle. This allows the CPU to initalize large blocks of memory with the same value. The 82C206 will continue performing transfer cycles until Channel 1 reaches terminal count.

Verify Transfer—The verify transfer is a pseudo-transfer which is useful for diagnostics. In this type of transfer the DMA will operate as if it is performing a Read or Write Transfer by generating HRQ, addresses and DACK but will do so without asserting a command signal. Since no transfer actually takes place IOCHRDY is ignored during Verify transfer cycles.

Autoinitialization

Each of the four DMA channel Mode Registers contains a bit which will cause the channel to reinitialize after reaching terminal count. During this process, referred to as Autoinitialization, the Base Address and Base Word Count Registers, which were originally written by the CPU, are reloaded into the Current Address and Current Word Count Registers (both the base and current registers are loaded during a CPU write cycle). The base registers remain unchanged during DMA Active cycles and can only be changed by the CPU. If the channel has been programmed to autoinitialize, the request mask bit will not be set upon reaching terminal count. This allows the DMA to continue operation without CPU intervention.

During memory-to-memory transfers the Word Count Registers of both Channel 0 and Channel 1 must be programmed with the same starting value for full autoinitialization. If Channel 0 reaches terminal count before Channel 1, then Channel 0 will reload the starting address and word count and continue transferring data from the beginning of the source block. Should Channel 1 reach terminal count first, it will reload the current registers and Channel 0 will remain uninitialized.

DREQ Priority

The 82C206 supports two schemes for establishing DREQ priority. The first is fixed priority which assigns priority based on channel position. In this method Channel 0 is assigned the highest priority. Priority assignment then progresses downward through the channels in order with Channel 3 receiving the lowest priority.

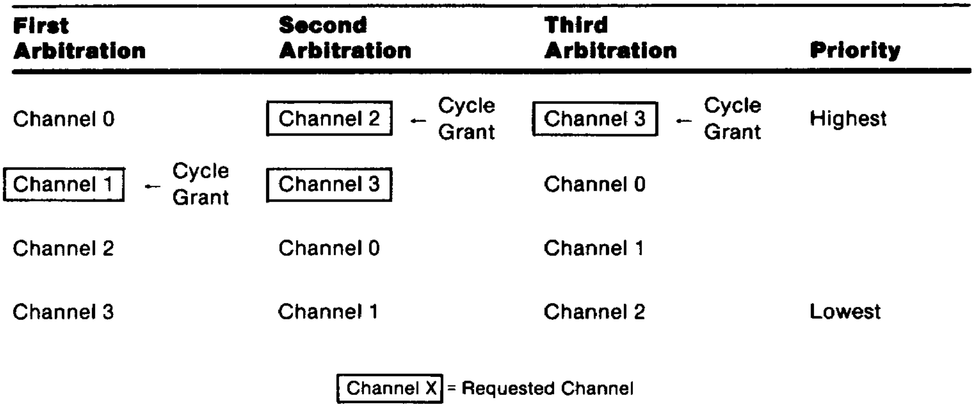
The second type of priority assignment is rotating priority. In this scheme the ordering of priority from Channel 0 to Channel 3 is maintained but the actual assignment of priority changes. The channel most recently serviced will be assigned the lowest priority and, since the order of priority assignment remains fixed, the remaining three channels rotate accordingly. The rotating priority assignment is illustrated in Figure 5.

In instances where multiple requests occur at the same time the 82C206 will issue a HRQ but will not freeze the priority logic until HLDA is returned. Once HDLA becomes active the priority logic is frozen and DACK is asserted on the highest requesting channel. Priority will not be re-evaluated until HLDA has been deactivated.

Address Generation

Eight intermediate bits of the address are multiplexed onto the data lines during Active cycles of the DMA. This reduces the number of pins required by the DMA subsystem. During state S1, the intermediate addresses are output on data lines XD0-XD7. These addresses should be externally latched and used to drive the system address bus. Since DMA1 is used to transfer 8-bit data and DMA2 is used to transfer 16-bit data, a one bit skew occurs in the intermediate address fields. DMA1 will therefore output addresses A8-A15 on the data bus at this time whereas DMA2 will output A9-A16. A separate set of latch and enable signals are provided for both DMA1 and DMA2 to accommodate the address skew.

During 8-bit DMA cycles, in which DMA1 is active, the 82C206 will output the lower 8-bits



First Arbitration		Second Arbitration		Third Arbitration		Priority
Channel 0		Channel 2	← Cycle Grant	Channel 3	← Cycle Grant	Highest
Channel 1	← Cycle Grant	Channel 3		Channel 0		
Channel 2		Channel 0		Channel 1		
Channel 3		Channel 1		Channel 2		Lowest

Channel X = Requested Channel

Figure 5. Rotating Priority Scheme

of address on XA0-XA7. The intermediate 8-bits of address will be output on XD0-XD7 and ADSTB8 will be asserted for one DMA clock cycle. The falling edge of ADSTB8 is used to latch the intermediate addresses A8-A15. An enable signal, AEN8, is used to control the output drivers of the external latch. A16-A23 are also generated at this time from a DMA Page Register in the 82C206. Note that A16 is output on the XA16 pin of the device.

16-bit DMA cycles from DMA2 require the 82C206 to output the lower 8-bits of the address on XA1-XA8. The intermediate addresses A9-A16 are output on XD0-XD7. Control for a separate latch is provided by signals ADSTB16 and AEN16. The DMA Page Register now generates A17-A23. During 16-bit DMA transfers XA0 and XA16 remain inactive.

The DMA Page Register is a set of 16 8-bit registers in the 82C206 which are used to generate the high order addresses during DMA cycles. Only 8 of the registers are actually used but all 16 were included to maintain IBM PC/AT compatibility. Each DMA channel has a register associated with it with the exception of Channel 0 of DMA2 which is used for internal cascading to DMA1. Assignment of each of these registers is shown in Figure 6 along with its Read/Write address.

Address	Register Function
080h	Unused
081h	8-bit DMA Channel 2 (DACK2)
082h	8-bit DMA Channel 3 (DACK3)
083h	8-bit DMA Channel 1 (DACK1)
084h	Unused
085h	Unused
086h	Unused
087h	8-bit DMA Channel 0 (DACK0)
088h	Unused
089h	16-bit DMA Channel 2 (DACK6)
08Ah	16-bit DMA Channel 3 (DACK7)
08Bh	16-bit DMA Channel 1 (DACK5)
08Ch	Unused
08Dh	Unused
08Eh	Unused
08Fh	Refresh Cycle

Figure 6.
DMA Address Extension Register Map

During Demand and Block Transfers, the 82C206 generates multiple sequential transfers. For most of these transfers the information in the external address latches will remain the same, eliminating the need to be relatched. Since the need to update the latches occurs only when a carry or borrow from the lower 8-bits of the Address Counter exists, the 82C206 will only update the latch contents when necessary. The 82C206 will therefore, only execute S1 cycles when necessary, resulting in an overall through-put improvement.

Compressed Timing

The DMA subsystem in the 82C206 can be programmed to transfer a word in as few as 3 DMA clock cycles. The normal DMA cycle consists of three states: S2, S3, and S4 (this assumes Demand or Block Transfer Mode). Normal transfers require 4 DMA clock cycles since S3 is executed twice due to the 1 wait state insertion. In systems capable of supporting higher through-put, the 82C206 can be programmed to omit one S3 and assert both commands in S2. S2 begins the cycle by generating the address and asserting both commands. One S3 cycle is executed and the cycle terminates in S4. If Compressed Timing is selected, T/C will be output in S2 and S1 cycles will be executed as necessary to update the address latch. Note that Compressed Timing is not allowed for memory-to-memory transfers.

Register Description

Current Address Register

Each DMA channel has a 16-bit Current Address Register which holds the address used during transfers. Each channel can be programmed to increment or decrement this register whenever a transfer is completed. This register can be read or written by the CPU in consecutive 8-bit bytes. If Autoinitialization is selected, this register will be reloaded from the Base Address Register upon reaching terminal count in the Current Word Count Register. Channel 0 can be prevented from incrementing or decrementing by setting the Address Hold Bit in the Command Register.

Current Word Count Register

Each channel has a Current Word Count Register which determines the number of transfers to perform. The actual number of transfers performed will be one greater than the value programmed into the register. The register is decremented after each transfer until it goes from zero to FFFFh. When this roll-over occurs the 82C206 will generate T/C and either suspend operation on that channel and set the appropriate Request Mask Bit or Autoinitialize and continue.

Base Address Register

Associated with each Current Address Register is a Base Address Register. This is a write only register which is loaded by the CPU when writing to the Current Address Register. The purpose of this register is to store the initial value of the Current Address Register for Autoinitialization. The contents of this register are loaded into the Current Address Register whenever terminal count is reached and the Autoinitialize Bit is set.

Base Word Count Register

This register preserves the initial value of the Current Word Count Register. It is also a write only register which is loaded by writing to the Current Word Count Register. This register is loaded into the Current Word Count Register during Autoinitialization.

Command Register

This register controls the overall operation of a DMA subsystem. The register can be read or written by the CPU and is cleared by either RESET or a Master Clear command.

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
DAK	DRQ	EW	RP	CT	CD	AH	M-M

DAK—DACK active level is determined by bit 7. Programming a 1 in this bit position makes DACK an active high signal.

DRQ—DREQ active level is determined by bit 6. Writing a 1 in this bit position causes DREQ to become active low.

EW—Extended Write is enabled by writing a 1 to bit 5, causing the write commands to be asserted one DMA cycle earlier during a transfer. The read and write commands both begin in state S2 when enabled.

RP—Writing a 1 to bit 4 causes the 82C206 to utilize a rotating priority scheme for honoring DMA requests. The default condition is fixed priority.

CT—Compressed timing is enabled by writing a 1 to bit 3 of this register. The default 0 condition causes the DMA to operate with normal timing.

CD—Bit 2 is the master disable for the DMA controller. Writing a 1 to this location disables the DMA subsystem (DMA1 or DMA2). This function is normally used whenever the CPU needs to reprogram one of the channels to prevent DMA cycles from occuring.

AH—Writing a 1 to bit 1 enables the address hold feature in Channel 0 when performing memory-to-memory transfers.

M-M—A 1 in the bit 0 position enables Channel 0 and Channel 1 to be used for memory-to-memory transfers.

Mode Register

Each DMA channel has a Mode Register associated with it. All four Mode Registers reside at the same I/O address. Bits 0 and 1 of the Write Mode Register command determine which channel's Mode Register gets written. The remaining six bits control the mode of the selected channel. Each channel's Mode Register can be read by sequentially reading the Mode Register location. A Clear Mode Register Counter command is provided to allow the CPU to restart the mode read process at a known point. During mode read operations, bits 0 and 1 will both be 1.

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
M1	M0	DEC	AI	TT1	TT0	CS1	CS0

(Read/Write Register)

M1-M0—Mode selection for each channel is accomplished by bits 6 and 7.

M1	M0	MODE
0	0	Demand Mode
0	1	Single Cycle Mode
1	0	Block Mode
1	1	Cascade Mode

DEC—Determines direction of the address counter. A one in bit 5 decrements the address after each transfer.

AI—The Autoinitialization function is enabled by writing a 1 in bit 4 of the Mode Register.

TT1-TT0—Bits 2 and 3 control the type of transfer which is to be performed.

TT1	TT0	TYPE
0	0	Verify Transfer
0	1	Write Transfer
1	0	Read Transfer
1	1	Illegal

CS1-CS0—Channel Select bits 1 and 0 determine which channel's Mode Register will be written. Read back of a mode register will result in bits 1 and 0 both being ones.

CS1	**CS0**	**CHANNEL**
0	0	Channel 0 select
0	1	Channel 1 select
1	0	Channel 2 select
1	1	Channel 3 select

Request Register

This is a four bit register used to generate software requests (DMA service can be requested either externally or under software control). Request Register bits can be set or reset independently by the CPU. The Request Mask has no effect on software generated requests. All four bits are read in one operation and appear in the lower four bits of the byte. Bits 4 through 7 are read as ones. All four request bits are cleared to zero by RESET.

msb lsb

b7	b6	b5	b4	b3	b2	b1	b0
X	X	X	X	X	RB	RS1	RS0

(Write Operation)

RB—The request bit is set by writing a 1 to bit 2. RS1-RS0 select which bit (channel) is to be manipulated.

RS1-RS0—Channel Select 0 and 1 determine which channel's Mode Register will be written. Read back of the mode register will result in bits 0 and 1 both being ones.

RS1	RS0	CHANNEL
0	0	Channel 0 select
0	1	Channel 1 select
1	0	Channel 2 select
1	1	Channel 3 select

Format for the Request Register read operation is shown below.

msb lsb

b7	b6	b5	b4	b3	b2	b1	b0
1	1	1	1	RC3	RC2	RC1	RC0

(Read Operation)

RC3-RC0—During a Request Register read, the state of the request bit associated with each channel is returned in bits 0 through 3 of the byte. The bit position corresponds to the channel number.

Request Mask Register

The Request mask register is a set of four bits which are used to inhibit external DMA requests from generating transfer cycles. This register can be programmed in two ways. Each channel can be independently masked by writing to the Write Single Mask Bit location. The data format for this operation is shown below.

msb lsb

b7	b6	b5	b4	b3	b2	b1	b0
X	X	X	X	X	MB	MS1	MS0

(Set/Reset Operation)

MB—Bit 2 sets or resets the request mask bit for the channel selected by MS1 and MS0. Writing a 1 in this bit position sets the mask, inhibiting external requests.

MS1-MS0—These two bits select the specific mask bit which is to be set or reset.

MS1	MS0	CHANNEL
0	0	Channel 0 select
0	1	Channel 1 select
1	0	Channel 2 select
1	1	Channel 3 select

Alternatively all four mask bits can be programmed in one operation by writing to the Write All Mask Bits address. Data format for this and the Read All Mask Bits function is shown below.

msb lsb

b7	b6	b5	b4	b3	b2	b1	b0
X	X	X	X	MB3	MB2	MB1	MB0

(Read/Write Operation)

MB3-MB0—Each bit position in the field represents the mask bit of a channel. The mask bit number corresponds to the channel number associated with the mask bit.

All four mask bits are set following a RESET or a Master Clear command. Individual channel mask bits will be set as a result of terminal count being reached, if Autoinitialize is disabled. The entire register can be cleared, enabling all four channels, by performing a Clear Mask Register operation.

Status Register

The status of all four channels can be determined by reading the Status Register. Information is available to determine if a channel has reached terminal count and whether an external service request is pending. Bits 0-3 of this register are cleared by RESET, Master Clear or each time a Status Read takes place. Bits 4-7 are cleared by RESET, Master Clear or the pending request being deasserted. Bits 4-7 are not affected by the state of the Mask Register Bits. The channel number corresponds to the bit position.

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
DRQ3	DRQ2	DRQ1	DRQ0	TC3	TC2	TC1	TC0

(Read Only Register)

Temporary Register

The Temporary Register is used as a temporary holding register for data during memory-to-memory transfers. The register is loaded during the first cycle of a memory-to-memory transfer from XD0-XD7. During the second cycle of the transfer, the data in the Temporary Register is output on the XD0-XD7 pins. Data from the last memory-to-memory transfer will remain in the register unless a RESET or Master Clear occurs.

Special Commands

Five Special Commands are provided to make the task of programming the device easier. These commands are activated as a result of a specific address and assertion of either a $\overline{\text{XIOR}}$ or $\overline{\text{XIOW}}$. Information on the data lines is ignored by the 82C206 whenever an $\overline{\text{XIOW}}$ activated command is issued, thus data returned on $\overline{\text{XIOR}}$ activated commands is invalid.

Clear Byte Pointer Flip-Flop—This command is normally executed prior to reading or writing to the address or word count registers. This initializes the flip-flop to point to the low byte of the register and allows the CPU to read or write the register bytes in correct sequence.

Set Byte Pointer Flip-Flop—Setting the Byte Pointer Flip-Flop allows the CPU to adjust the pointer to the high byte of an address or word count register.

Master Clear—This command has the same effect as a hardware RESET. The Command Register, Status Register, Request Register, Temporary Register, Mode Register Counter and Byte Pointer Flip-Flop are cleared and the Request Mask Register is set. Immediately following Master Clear or RESET, the DMA will be in the Idle Condition.

Clear Request Mask Register—This command enables all four DMA channels to accept requests by clearing the mask bits in the register.

Clear Mode Register Counter—In order to allow access to four Mode Registers while only using one address, an additional counter is used. After clearing the counter all four Mode Registers may be read by doing successive reads to the Read Mode Register address. The order in which the registers will be read is Channel 0 first, Channel 3 last.

INTERRUPT CONTROLLER FUNCTIONAL DESCRIPTION

The programmable interrupt controllers in the 82C206 function as a system wide interrupt manager in an iAPX86 system. They accept requests from peripherals, resolve priority on pending interrupts and interrupts in service, issue an interrupt request to the CPU, and provide a vector which is used as an index by the CPU to determine which interrupt service routine to execute.

A variety of priority assignment modes are provided,which can be reconfigured at any time during system operation, allowing the complete interrupt subsystem to be restructured, based on the system environment.

Overview

Two interrupt controllers, INTC1 and INTC2, are included in the 82C206. Each of the interrupt controllers is equivalent to an 8259A device operating in iAPX86 Mode. The two devices are interconnected and must be programmed to operate in Cascade Mode (see Figure 7) for proper operation of all 16 interrupt channels. INTC1 is located at addresses 020H-021H and is configured for Master operation (defined below) in Cascade Mode. INTC2 is a Slave device (defined below) and is located at 0A0H-0A1H. The Interrupt Request output signal from INTC2 (INT) is internally connected to the interrupt request input Channel 2 (IR2) of INTC1. The address decoding and Cascade interconnection matches that of the IBM PC/AT.

Two additional interconnections are made to the interrupt request inputs of the interrupt controllers. The output of Timer 0 in the Counter/Timer subsystem is connected to Channel 0 (IR0) of INTC1. Interrupt request from the Real Time Clock is connected to Channel 0 (IR0) of INTC2. Figure 8 lists the 16 interrupt channels and their interrupt request source.

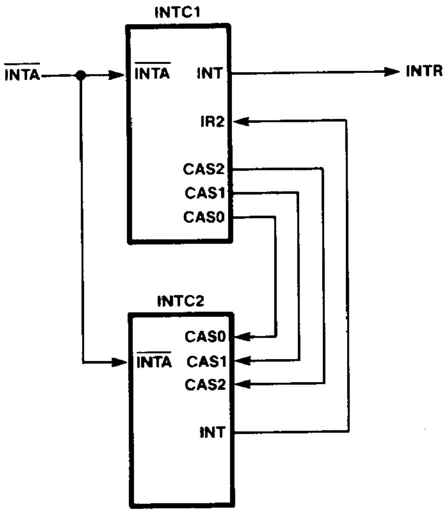


Figure 7. Internal Cascade Interconnect

Description of the Interrupt Subsystem will pertain to both INTC1 and INTC2 unless otherwise noted. Wherever register addresses are used, the address for the INTC1 register will be listed first and the address for the INTC2 register will follow in parenthesis. Example 020H (0A0H)

Controller Operation

Figure 9 is a block diagram of the major elements in the interrupt controller. The Interrupt Request Register (IRR) is used to store requests from all of the channels which are requesting service. Interrupt Request Register bits are labeled using the Channel Name IR7-IR0. The In-Service Register (ISR) contains all the channels which are currently being serviced (more than one channel can be in service at a time). In-Service Register bits are labeled IS7-IS0 and correspond to IR7-IR0. The Interrupt Mask Register (IMR) allows the CPU to disable any or all of the interrupt channels. The Priority Resolver evaluates inputs from the above three registers, issues an interrupt request, and latches the corresponding bit into the In-Service Register. During interrupt acknowledge cycles, a master controller outputs a code to the slave device which is compared in the Cascade Buffer/Comparator with a three bit ID code previously written. If a match occurs in the slave controller, it will generate an interrupt vector. The contents of the Vector Register are used to provide the CPU with an interrupt vector during Interrupt Acknowledge (INTA) cycles.

Interrupt Sequence

The 82C206 allows the CPU to perform an indirect jump to a service routine in response to a request for service from a peripheral device. The indirect jump is based on a vector which is provided by the 82C206 on the second of two CPU generated INTA cycles (the first INTA cycle is used for resolving priority and the second cycle is for transferring the vector to the CPU). The events which occur during an interrupt sequence are as follows:

1—One or more of the interrupt requests

Controller Number	Channel Name	Interrupt Request Source
INTC1	IR0	Counter/Timer Out0
INTC1	IR1	IRQ1 Input Pin
INTC1	IR2	INTC2 Cascade Interrupt
INTC1	IR3	IRQ3 Input Pin
INTC1	IR4	IRQ4 Input Pin
INTC1	IR5	IRQ5 Input Pin
INTC1	IR6	IRQ6 Input Pin
INTC1	IR7	IRQ7 Input Pin
INTC2	IR0	Real Time Clock IRQ
INTC2	IR1	IRQ9 Input Pin
INTC2	IR2	IRQ10 Input Pin
INTC2	IR3	IRQ11 Input Pin
INTC2	IR4	IRQ12 Input Pin
INTC2	IR5	IRQ13 Input Pin
INTC2	IR6	IRQ14 Input Pin
INTC2	IR7	IRQ15 Input Pin

Figure 8. Interrupt Request Source

(IR7-IR0) becomes active, setting the corresponding IRR bit(s).

2—The interrupt controller resolves priority based on the state of the IRR, IMR and ISR and asserts the INTR output if appropriate.

3—The CPU accepts the interrupt and responds with an INTA cycle.

4—During the first INTA cycle, the highest priority ISR bit is set and the corresponding IRR bit is reset. The internal Cascade address is generated and the XD7-XD0 outputs remain tri-stated.

5—The CPU will execute a second INTA cycle, during which the 82C206 will drive an 8-bit vector onto the data pins XD7-XD0, which is in turn latched by the CPU. The format of this vector is shown in Figure 12. Note that V7-V3 in Figure 12 are programmable by writing to Ininialization Control Word 2 (see Initialization Command Words section below).

6—At the end of the second INTA cycle, the ISR bit will be cleared if the Automatic End Of Interrupt mode is selected (see End Of Interrupt section below). Otherwise, the ISR bit must be cleared by an End Of Interrupt (EOI) command from the CPU at the end of the interrupt service routine.

If no interrupt request is present at the beginning of the first INTA cycle (i.e., a spurious interrupt) INTC1 will issue an interrupt level 7 vector during the second INTA cycle.

End Of Interrupt

EOI is defined as the condition which causes an ISR bit to be reset. Determination of which ISR bit is to be reset can be done by a CPU command (specific EOI) or, the Priority Resolver can be instructed to clear the highest priority ISR bit (non-specific EOI).

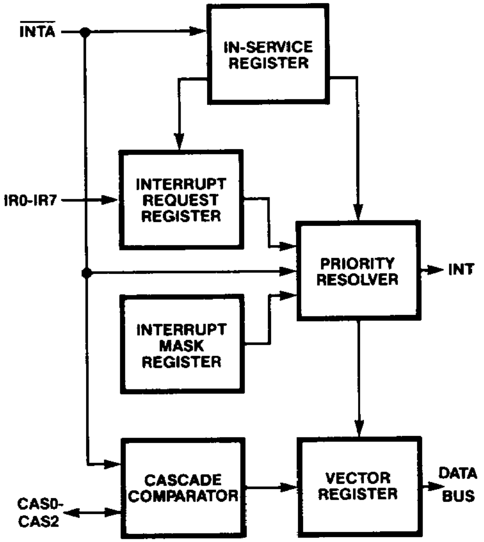


Figure 9. Interrupt Controller Block Diagram

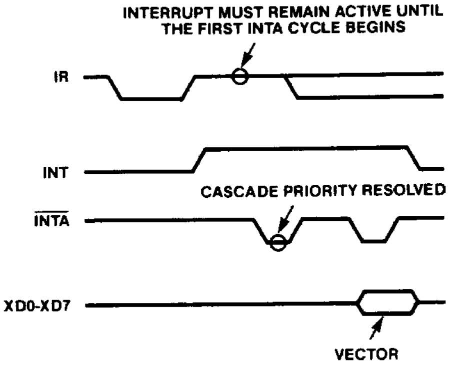

INTERRUPT MUST REMAIN ACTIVE UNTIL THE FIRST INTA CYCLE BEGINS
IR
INT
CASCADE PRIORITY RESOLVED
INTA
XD0-XD7
VECTOR

Figure 10. Interrupt Sequence

The 82C206 can determine the correct ISR bit to reset when operated in modes which do not alter the fully nested structure, since the current highest priority ISR bit is necessarily the last level acknowledged and serviced. In conditions where the fully nested structure is not preserved, a specific EOI must be generated at the end of the interrupt service routine. An ISR bit that is masked, in Special Mask Mode by a IMR bit, will not be cleared by a non-specific EOI command. The interrupt controller can optionally generate an Automatic End Of Interrupt (AEOI) on the trailing edge of the second INTA cycle.

Priority Assignment

Assignment of priority is based on an interrupt channel's position relative to the other channels in the interrupt controller. After the initialization sequence, IR0 has the highest priority, IR7 has the lowest, and priority assignment is fixed (Fixed Priority Mode). Priority assignment can be rotated either manually (Specific Rotation Mode) or automatically (Automatic Rotation Mode) by programming Operational Command Word 2 (OCW2).

Fixed Priority Mode—This is the default condition which exists unless rotation (either manual or automatic) is enabled, or the controller is programmed for Polled Mode. In Fixed Priority Mode, interrupts are fully nested with priority assigned as shown:

	Lowest						Highest	
Priority Status	7	6	5	4	3	2	1	0

Nesting allows interrupts of a higher priority to generate interrupt requests prior to the completion of the interrupt in service. When an interrupt is acknowledged, priority is resolved, the highest priority request's vector is placed on the bus and the ISR bit for that channel is set. This bit remains set until an EOI (automatic or CPU generated) is issued to that channel. While the ISR bit is set, all interrupts of equal or lower priority are inhibited. Note that a higher priority interrupt which occurs during an interrupt service routine, will only be acknowledged if the CPU has internally re-enabled interrupts.

Specific Rotation Mode—Specific Rotation allows the system software to re-assign priority levels by issuing a command which redefines the highest priority channel.

Before Rotation

	Lowest							Highest
Priority Status	7	6	5	4	3	2	1	0

(Specific Rotation command issued with Channel 5 specified)

After Rotation

	Lowest							Highest
Priority Status	5	4	3	2	1	0	7	6

Automatic Rotation Mode—In applications where a number of equal priority peripherals are requesting interrupts, Automatic Rotation may be used to equalize the priority assignment. In this mode a peripheral, after being serviced, is assigned the lowest priority. All peripherals connected to the controller will be serviced at least once in 8 interrupt requests to the CPU from the controller. Automatic rotation will occur,if enabled, due to the occurrance of EOI (automatic or CPU generated).

Before Rotation (IR4 is highest priority request being serviced)

ISR Status Bit	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
	0	1	0	1	0	0	0	0
	Lowest							Highest
Priority Status	7	6	5	4	3	2	1	0

After Rotation (IR4 service completed)

ISR Status Bit	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
	0	1	0	0	0	0	0	0
	Lowest							Highest
Priority Status	4	3	2	1	0	7	6	5

Programming The Interrupt Controller

Two types of commands are used to control the 82C206 interrupt controllers, Initialization Command Words (ICWs) and Operational Command Words (OCWs).

Initialization Command Words

The initialization process consists of writing a sequence of 4 bytes to each interrupt controller. The initialization sequence is started by writing the first Initialization Command Word (ICW1) to address 020H (0A0H) with a 1 on bit 4 of the data byte. The interrupt controller interprets this as the start of an initialization sequence and does the following:

1—The Initialization Command Word Counter is reset to zero.
2—ICW1 is latched into the device
3—Fixed Priority Mode is selected
4—IR7 is assigned the highest priority
5—The Interrupt Mask Register is cleared
6—The Slave Mode Address is set to 7
7—Special Mask Mode is disabled
8—The IRR is selected for Status Read operations

The next three I/O writes to address 021H (0A1H) will load ICW2-ICW4. See Figure 11 for a flow chart of the initialization sequence. The initialization sequence can be terminated

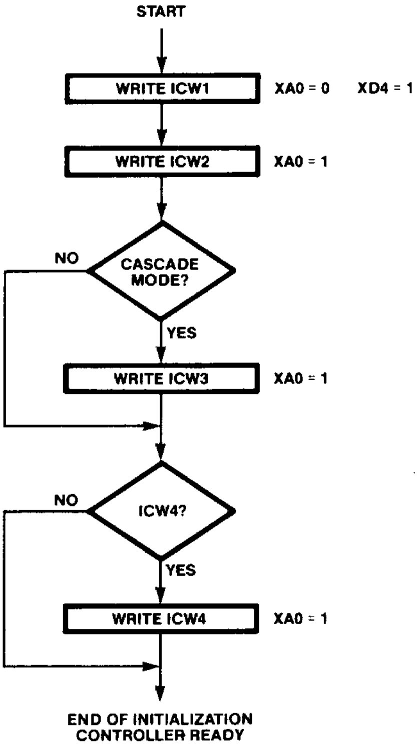


Figure 11. Initialization Sequence

	D7	D6	D5	D4	D3	D2	D1	D0
IR7	V7	V6	V5	V4	V3	1	1	1
IR7	V7	V6	V5	V4	V3	1	1	0
IR7	V7	V6	V5	V4	V3	1	0	1
IR7	V7	V6	V5	V4	V3	0	1	1
IR7	V7	V6	V5	V4	V3	0	1	0
IR7	V7	V6	V5	V4	V3	0	0	1
IR7	V7	V6	V5	V4	V3	0	0	0

Figure 12. Interrupt Vector Byte

at any point (all 4 bytes must be written for the controller to be properly initialized) by writing to address 020H (0A0H) with a 0 in data bit 4. Note, this will cause OCW2 or OCW3 to be written.

ICW1—Address 020H (0A0H)

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
X	X	X	SI	LTM	X	SM	X

(Write Only Register)

SI—Bit 4 indicates to the interrupt controller that an Initialization Sequence is starting and must be a 1 to write ICW1.

LTM—Bit 3 selects level or edge triggered inputs to the IRR. If a 1 is written to LTM, a 'high' level on the IRR input will generate an interrupt request. The IR must be active until the first INTA cycle is started to generate the proper interrupt vector (an IR7 vector will be generated if the IRR input is deasserted early) and the IR must be removed prior to EOI to prevent a second interrupt from occuring.

SM—Bit 1 selects between Single Mode and Cascade Mode. Single Mode is used whenever only one interrupt controller (INTC1) is used and is not recommended for this device. Cascade Mode allows the two interrupt controllers to be connected through IR2 of INTC1. INTC1 will allow INTC2 to generate its own interrupt vectors if Cascade Mode is selected and the highest priority IR pending is from an INTC2 input. INTC1 and INTC2 must be programmed for Cascade Mode for both devices to operate.

ICW2—Address 021H (0A1H)

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
V7	V6	V5	V4	V3	X	X	X

(Write Only Register)

V7-V3—These bits are the upper 5 bits of the interrupt vector and are programmable by the CPU. The lower three bits of the vector are generated by the Priority Resolver during INTA (see Figure 12). INTC1 and INTC2 need not be programmed with the same value in ICW2.

ICW3 Format for INTC1—Address 021H

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
S7	S6	S5	S4	S3	S2	S1	S0

(Write Only Register)

S7-S0—Select which IR inputs have Slave Mode controllers connected. ICW3 in INTC1 must be written with a 04H for INTC2 to function.

ICW3 Format for INTC2—Address 0A1H

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	ID2	ID1	ID0

(Write Only Register)

ID2-ID0—Determine the Slave Mode address the controller will respond to during the cascaded INTA sequence. ICW3 in INTC2 should be written with a 02H for Cascade Mode operation. Note, b7-b3 should be zero.

ICW4—Address 021H (0A1H)

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
X	X	X	EMI	X	X	AEOI	X

(Write Only Register)

EMI—Bit 4 will Enable Multiple Interrupts from the same channel in Fixed Priority Mode. This allows INTC2 to fully nest interrupts, when Cascade Mode with Fixed Priority Mode are both selected, without being blocked by INTC1. Correct handling of this mode requires the CPU to issue a non-specific EOI command to INTC2 and check its In-Service Register for zero, when exiting an interrupt service routine. If zero, a non-specific EOI command should be sent to INTC1. If non-zero, no command is issued.

AEOI—Auto End Of Interrupt is enabled when ICW4 is written with a zero in bit 1. The interrupt controller will perform a non-specific EOI on the trailing edge of the second INTA cycle. Note, this function should not be used in a device with fully nested interrupts unless the device is a cascade Master.

Operational Command Words

Operational Command Words (OCWs) allow the 82C206 interrupt controllers to be controlled or reconfigured at any time while operating. Each interrupt has 3 OCWs which can be programmed to affect the proper operating configuration and a Status Register to monitor controller operation.

Operational Command Word 1 (OCW1) is located at address 021h (0A1h) and may be written any time the controller is not in Initialization Mode. Operational Command Words 2 and 3 (OCW2,OCW3) are located at address 020H (0A0H). Writing to address 020H (0A0H) with a 0 in bit 4 will place the controller in operational mode and load OCW2 (if data bit 3 = 0) or OCW3 (if data bit 3 = 1).

OCW1—Address 021H (0A1H)

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
M7	M6	M5	M4	M3	M2	M1	M0

(Read/Write Register)

M7-M0—These bits control the state of the Interrupt Mask Register. Each Interrupt Request can be masked by writing a 1 in the appropriate bit position (M0 controls IR0 etc.). Setting an IMR bit has no affect on lower priority requests. All IMR bits are cleared by writing ICW1.

OCW2—Address 020h (0A0h)

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
R	SL	EOI	SI	2/3	L2	L1	L0

(Write Only Register)

R—This bit in conjunction with SL and EOI selects operational function. Writing a 1 in bit 7 causes one of the rotate functions to be selected.

R	SL	EOI	Function
1	0	0	Rotate on auto EOI enable*
1	0	1	Rotate on non-specific EOI
1	1	0	Specific Rotate Command
1	1	1	Rotate on specific EOI

***This function is disabled by writing a zero to all three bit positions.**

SL—This bit in conjunction with R and EOI selects operational function. Writing a 1 in this bit position causes a specific or immediate function to occur. All specific commands require L2-L0 to be valid except no operation.

R	SL	EOI	Function
0	1	0	No operation
0	1	1	Specific EOI Command
1	1	0	Specific Rotate Command
1	1	1	Rotate on specific EOI

EOI—This bit in conjunction with R and SL selects operational function. Writing a 1 in this bit position causes a function related to EOI to occur.

R	SL	EOI	Function
0	0	1	Non-specific EOI Command
0	1	1	Specific EOI Command
1	0	1	Rotate on non-specific EOI
1	1	1	Rotate on specific EOI

SI—Writing a 0 in this bit position takes the interrupt controller out of initialize mode and writes OCW2 or OCW3.

2/3—If the I/O write places a 0 in bit 4 (SI), then writing a 0 in bit 3 (2/3) selects OCW2 and writing a 1 will select OCW3.

L2-L0—These three bits are internally decoded to select which interrupt channel is to be affected by the Specific command. L2-L0 must be valid during three of the four specific cycles (see SL above).

OCW3—Address 020H (0A0H)

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
0	ESMM	SMM	SI	2/3	PM	RR	RIS

(Write Only Register)

ESMM—Writing a 1 in this bit position enables the Set/Reset Special Mask Mode function controlled by bit 5 (SMM). ESMM allows the other functions in OCW3 to be accessed and manipulated without affecting the Special Mask Mode state.

SMM—If ESMM and SMM both are written with a 1 the Special Mask Mode is enabled. Writing a 1 to ESMM and a 0 to SMM disables Special Mask Mode. During Special Mask Mode, writing a 1 to any bit position inhibits interrupts and a 0 enables interrupts on the associated channel by causing the Priority Resolver to ignore the condition of the ISR.

SI—See SI above.

2/3—See 2/3 above.

PM—Polled Mode is enabled by writing a 1 to bit 2 of OCW3, causing the 82C206 to perform the equivalent of an INTA cycle during the next I/O read operation to the controller. The byte read during this cycle will have bit 7 set if an interrupt is pending. If bit 7 of the byte is set, the level of the highest pending request will be encoded on bits 2-0. The IRR will remain frozen until the read cycle is completed at which time the PM bit is reset.

RR—When the RR bit (bit 1) is 1, reading the Status Port at address 020h (0A0h) will cause the contents of IRR or ISR (determined by RIS) to be placed on XD7-XD0. Asserting PM forces RR reset.

RIS—This bit selects between the IRR and the ISR during Status Read operations if RR = 1.

COUNTER/TIMER FUNCTIONAL DESCRIPTION

The Counter/Timer (CTC) in the 82C206 is general purpose, and can be used to generate accurate time delays under software control. The CTC contains 3 16-bit counters (Counter 0-3) which can be programmed to count in binary or binary coded decimal (BCD). Each counter operates independently of the other two and can be programmed for operation as a timer or a counter.

All three of the counters shown in Figure 13 are controlled from a common set of control logic. The Control Logic decodes control in-

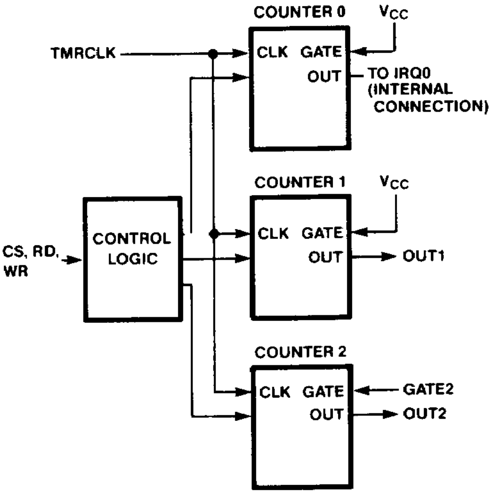


Figure 13. Counter/Timer Block Diagram

formation written to the CTC and provides the controls necessary to load, read, configure and control each counter. Counter 0 and Counter 1 can be programmed for all six modes, but Mode 1 and Mode 5 have limited usefulness due to the lack of an external hardware trigger signal. Counter 2 can be operated in any of six modes listed below.

Mode 0	Interrupt on terminal count
Mode 1	Hardware retriggerable one-shot
Mode 2	Rate generator
Mode 3	Square wave generator
Mode 4	Software triggered strobe
Mode 5	Hardware retriggerable strobe

All three counters in the CTC are driven from a common clock input pin (TMRCLK) which is independent from other clock inputs to the 82C206. Counter 0's output (Out0) is connected to IR0 of INTC1 (see Interrupt Controller Functional Description) and may be used as an interrupt to the system for time keeping and task switching. Counter 1 may be programmed to generate pulses or square waves for use by external devices. The third counter (Counter 2) is a full function Counter/Timer. This channel can be used as an interval timer, a counter, or as a gated rate/pulse generator.

Counter Description

Each counter in the CTC contains a Control Register, a Status Register, a 16-bit Counting Element (CE), a pair of 8-bit Counter Input Latches (CIL,CIH),and a pair of 8-bit Counter Output Latches (COL,COH). Each counter also has a clock input for loading and decrementing the CE, a mode defined GATE input for controlling the counter (only GATE2 is externally accessible), and an OUT signal (OUT0 is not externally accessible). The OUT signal's state and function are controlled by the Counter Mode and condition of the CE (see Mode Definitions).

The Control Register stores the mode and command information used to control the counter. The Control Register may be loaded by writing a byte, containing a pointer to the desired counter, to the Write Control Word address (043h). The remaining bits in the byte contain the mode, the type of command, and count format information.

The Status register allows the software to monitor counter condition and read back the contents of the Control Register.

The Counting Element is a loadable 16-bit synchronous down counter. The CE is loaded or decremented on the falling edge of TMRCLK. The CE contains the maximum count when a 0 is loaded; which is equivalent to 65536 in binary operation or 10000 in BCD. The CE does not stop when it reaches 0. In Modes 2 and 3 the CE will be reloaded and in all other modes it will wrap around to FFFF in binary operation or 9999 in BCD.

The CE is indirectly loaded by writing one or two bytes (optional) to the Counter Input Latches, which are in turn loaded into the CE. This allows the CE to be loaded or reloaded in one TMRCLK cycle.

The CE is also read indirectly by reading the contents of the Counter Output Latches. COL and COH are transparent latches which can be read while transparent or latched (see Latch Counter Command).

Programming The CTC

After power-up the condition of CTC Control Registers, counter registers, CE, and the output of all counters is undefined. Each counter must be programmed before it can be used.

Counters are programmed by writing a Control Word and then an initial count. The Control Register of a counter is written by writing to the Control Word address (see Figure 14). The Control Word is a write only location.

Control Word—(043H)

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
F3	F2	F1	F0	M2	M1	M0	BCD

(Write Only Register)

F3-F0—Bits 7-4 determine the command to be performed.

M2-M0—Bits 3-1 determine the counter's mode during Read/Write Counter Commands (see Read/Write Counter Command) or select the counter during a Read-Back Command (see Read-Back Command). Bits 3-1 become "don't care" during Latch Counter Commands.

Address	Function
040h	Counter 0 Read/Write
041h	Counter 1 Read/Write
042h	Counter 2 Read/Write
043h	Control Register Write Only

Figure 14. Counter/Timer Address Map

BCD—Bit 0 selects binary coded decimal counting format during Read/Write Counter Commands. Note, during Read-Back Command this bit must be 0.

Read/Write Counter Command

When writing to a counter, two conventions must be observed:

1—Each counters Control Word must be written before the initial count is written.

2—Writing the initial count must follow the

F3	F2	F1	F0	Command
0	0	0	0	Latch Counter 0 (see Counter Latch Command)
0	0	0	1	Read/Write Counter 0 LSB Only
0	0	1	0	Read/Write Counter 0 MSB Only
0	0	1	1	Read/Write Counter 0 LSB then MSB
0	1	0	0	Latch Counter 1 (see Counter Latch Command)
0	1	0	1	Read/Write Counter 1 LSB Only
0	1	1	0	Read/Write Counter 1 MSB Only
0	1	1	1	Read/Write Counter 1 LSB then MSB
1	0	0	0	Latch Counter 2 (see Counter Latch Command)2
1	0	0	1	Read/Write Counter 2 LSB Only
1	0	1	0	Read/Write Counter 2 MSB Only
1	0	1	1	Read/Write Counter 2 LSB then MSB
1	1	X	X	Read-Back Command (see Counter Read-Back Command)

MSB = most significant byte
LSB = least significant byte

format specified in the Control Word (least significant byte only, most significant byte only, or least significant byte and then most significant byte).

A new initial count can be written into the counter at any time after programming without rewriting the Control Word providing the programmed format is observed.

During Read/Write Counter Commands M3-M0 are defined as follows:

M2	M1	M0	Function
0	0	0	Select Mode 0
0	0	1	Select Mode 1
X	1	0	Select Mode 2
X	1	1	Select Mode 3
1	1	0	Select Mode 4
1	1	1	Select Mode 5

Latch Counter Command

When a Latch Counter Command is issued, the counter's output latches (COL,COH) latch the current state of the CE. COL and COH remain latched until read by the CPU, or the counter is reprogrammed. The output latches then return to a "transparent" condition. In this condition the latches are enabled and the contents of the CE may be read directly.

Latch Counter Commands may be issued to more than one counter before reading the first counter to which the command was issued. Also, multiple Latch Counter Commands issued to the same counter without reading the counter will cause all but the first command to be ignored.

Read-Back Command

The Read-Back Command allows the user to check the count value, Mode, and state of the OUT signal and Null Count Flag of the selected counter(s).

The format of the Read-Back Command is:

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
1	1	LC	LS	C2	C1	C0	0

LC—Writing a 0 in bit 5 causes the selected counter(s) to latch the state of the CE in COL and COH.

LS—Writing a 0 in bit 4 causes the selected counter(s) to latch the current condition of it's Control Register, Null Count and Output into the Status Register. The next read of the Counter will result in the contents of the Status Register being read (see Status Read).

C2-C0—Writing a 1 in bit 3 causes Counter 3 to latch one or both of the registers specified by LC and LS. The same is true for bits 2 and 1 except that they enable Counters 1 and 0 respectively.

Each counter's latches remain latched until either the latch is read or the counter is reprogrammed.

If LS = LC = 0, status will be returned on the next read from the counter. The next one or two reads (depending on whether the counter is programmed to transfer one or two bytes) from the counter result in the count being returned.

Status Byte

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
OUT	NC	F1	F0	M2	M1	M0	BCD

OUT—Bit 7 contains the state of the OUT signal of the counter

NC—Bit 6 contains the condition of the Null Count Flag. This flag is used to indicate that the contents of the CE are valid. NC will be set to a 1 during a write to the Control Register or the counter. NC is cleared to a 0 whenever the counter is loaded from the counter input registers.

F1-F0—Bits 5-4 contain the F1 and F0 Command bits which were written to the Command Register of the counter during initialization. This information is useful in determining whether the high byte, the low byte or both must be transferred during counter read/write operations.

M2-M1—These bits reflect the mode of the counter and are interpreted in the same manner as in Write Command operations.

BCD—Bit 0 indicates the CE is operating in BCD format.

Counter Operation

Due to the previously stated restrictions in Counter 0 and Counter 1, Counter 2 will be used as the example in describing counter operation, but the description of Mode 0, 2, 3 and 4 is relevant to all counters.

The following terms are defined for describing CTC operation.

TMRCLK pulse—A rising edge followed by a falling edge of the 82C206 TMRCLK input.

trigger—The rising edge of the GATE2 input.

counter load—The transfer of the 16-bit value in CIL and CIH to the CE.

initialized—A Control Word written and the Counter Input Latches loaded.

Counter 2 operates in one of the following modes.

Mode 0—Interrupt on terminal count

Writing the Control Word causes OUT2 to go low and remain low until the CE reaches 0, at which time it goes back high and remains high until a new count or Control Word is written. Counting is enabled when GATE2 = 1. Disabling the count has no effect on OUT2.

The CE is loaded with the first TMRCLK pulse after the Control Word and initial count are loaded. When both CIL and CIH are written, the CE is loaded after CIH is written (see Write Operations). This TMRCLK pulse does not decrement the count, so for an initial count of N, OUT2 does not go high until N+1 TMRCLK pulses after initialization. Writing a new initial count to the counter reloads the CE on the next TMRCLK pulse and counting continues from the new count.

If an initial count is written with GATE2 = 0, it will still be loaded on the next TMRCLK pulse but counting does not begin until GATE2 = 1. OUT2 therefore, goes high N TMRCLK pulses after GATE2 = 1.

Mode 1—Hardware retriggerable one-shot

Writing the Control Word causes OUT2 to go high initially. Once initialized the counter is armed and a trigger causes OUT2 to go low on the next TMRCLK pulse. OUT2 then remains low until the counter reaches 0. An initial count of N results in a one-shot pulse N TMRCLK cycles long.

Any subsequent triggers while OUT2 is low cause the CE to be reloaded, extending the length of the pulse. Writing a new count to CIL and CIH will not affect the current one-shot unless the counter is retriggered.

Mode 2—Rate generator

Mode 2 functions as a divide-by-N counter, with OUT2 as the carry. Writing the Control Word during initialization sets OUT2 high.

When the initial count is decremented to 1, OUT2 goes low on the next TMRCLK pulse. The following TMRCLK pulse returns OUT2 high, reloads the CE and the process is repeated. In Mode 2 the counter continues counting (if GATE2 = 1) and will generate an OU T2 pulse every N TMRCLK cycles. Note that a count of 1 is illegal in Mode 2.

GATE2 = 0 disables counting and forces OUT2 high immediately. A trigger reloads the the CE on the next TMRCLK pulse. Thus GATE2 can be used to synchronize the counter to external events.

Writing a new count while counting does not affect current operation unless a trigger is received. Otherwise, the new count will be loaded at the end of the current counting cycle.

Mode 3—Square wave generator

Mode 3 is similar to Mode 2 in every respect except for the duty cycle of OUT2. OUT2 is set high initially and remains high for the first half of the count. When the first half of the initial count expires, OUT2 goes low for the remainder of the count.

If the counter is loaded with an even count, the duty cycle of OUT2 will be 50% (high = low = N/2). For odd count values, OUT2 is high one TMRCLK cycle longer than it is low. Therefore, high = (N+1)/2 and low = (N-1)/2.

Mode 4—Software triggered strobe

Writing the Control Word causes OUT2 to go high initially. Expiration of the initial count causes OUT2 to go low for one TMRCLK cycle. GATE2 = 0 disables counting but has no effect on OUT2. Also, a trigger will not reload the CE.

The counting sequence is started by writing the initial count. The CE is loaded on the TMRCLK pulse after initialization. The CE begins decrementing one TMRCLK pulse later. OUT2 will go low for one TMRCLK cycle, (N+1) cycles after the initial count is written.

If a new initial count is written during a counting sequence, it is loaded into the CE on the next TMRCLK pulse and the sequence continues from the new count. This allows the sequence to be "retriggerable" by software.

Mode 5—Hardware triggered strobe

Writing the Control Word causes OUT2 to go high initially. Counting is started by trigger. The expiration of the initial count causes OUT2 to go low for one TMRCLK cycle. GATE2 = 0 disables counting.

The CE is loaded on the TMRCLK pulse after a trigger. Since loading the CE inhibits decrementing, OUT2 will go low for one TMRCLK cycle, (N+1) TMRCLK cycles after the trigger.

If a new count is loaded during counting, the current counting sequence will not be affected unless a trigger occurs. A trigger causes the counter to be reloaded from CIL and CIH, making the counter "retriggerable".

Mode	**Condition**		
	Low	Rising	High
0	Disables Counting	—	Enables Counting
1	—	a) Initiates Counting b) Resets Out Pin	—
2	a) Disables Counting b) Forces Out Pin High	Initates Counting	Enables Counting
3	a) Disables Counting b) Forces Out Pin High	Initiates Counting	Enables Counting
4	Disables Counting	—	Enables Counting
5	—	Initiates Counting	—

Figure 15. Gate Pin Function

GATE2

In Modes 0, 2, 3 and 4 GATE2 is level sensitive and is sampled on the rising edge of TMRCLK. In Modes 1, 2, 3 and 5 the GATE2 input is rising-edge sensitive. This rising edge sets an internal flip-flop whose output is sampled on the next rising edge of TMRCLK. The flop-flop resets immediately after being sampled. Note that in Modes 2 and 3 the GATE2 input is both edge and level sensitive.

REAL TIME CLOCK FUNCTIONAL DESCRIPTION

This section of the 82C206 combines a complete time-of-day clock with alarm, one hundred year calendar, a programmable periodic interrupt, and 114 bytes of low power static RAM. Provisions are made to enable the device to operate in a low power (battery powered) mode and protect the contents of both the RAM and clock during system power-up and power-down.

Register Access

Reading and writing to the 128 locations in the Real Time Clock is accomplished by first placing the Index Address of the location you wish to access on the data input pins XD0-XD6 and then strobing the AS input pin. The address will then be latched into the Index Address Register on the falling edge of AS. The Index Address Register is then used as a pointer to the specific byte in the Real Time Clock, which may be read or written to by asserting $\overline{\text{XIOR}}$ or $\overline{\text{XIOW}}$ with with an address on the XA9-XA0 inputs of 071H.

Since AS will most likely be generated by an I/O operation which will result in the assertion of $\overline{\text{XIOW}}$, it is recommended that an address of 070H be applied to the XA9-XA0 inputs during this time. This will prevent the modification of other registers in the 82C206.

Address Map

Figure 16 illustrates the internal register/RAM organization of the Real Time Clock portion of the 82C206. The 128 addressable locations in the Real Time Clock are divided into 10 bytes which normally contain the time, calendar, and alarm data, four control and status bytes and 114 general purpose RAM bytes. All 128 bytes are readable by the CPU. The CPU may also write to all locations except Registers C, D, Bit 7 of Register A and Bit 7 of the Seconds Byte which is always 0.

Index	Function
00	SECONDS
01	SECONDS ALARM
02	MINUTES
03	MINUTES ALARM
04	HOURS
05	HOURS ALARM
06	DAY OF WEEK
07	DATE OF MONTH
08	MONTH
09	YEAR
0A	REGISTER A
0B	REGISTER B
0C	REFISTER C
0D	REGISTER D
0E	USER RAM
0F	USER RAM
.	
.	
.	
7E	USER RAM
7F	USER RAM

Figure 16. Address Map for Real Time Clock

Time Calendar and Alarm Bytes

The CPU can obtain the time and calendar information by reading the appropriate locations in the Real Time Clock. Initalization of the time, calendar and alarm information is accomplished by writing to these locations. Information is stored in these locations in binary-coded decimal (BCD) format.

Before initalization of the internal registers can be performed, the SET bit in Register B should be set to a "1" to prevent Real Time Clock updates from occuring. The CPU then initalizes the first 10 locations in BCD format. The SET bit should then be cleared to allow updates. Once initalized and enabled, the Real Time Clock will perform Clock/Calendar updates at a 1 Hz rate.

Index Register Address	Function	BCD Range
0	Seconds	00-59
1	Seconds Alarm	00-59
2	Minutes	00-59
3	Minutes Alarm	00-59
4	Hours (12 hour mode) Hours (24 hour mode)	01-12 (AM) 81-92 (PM) 00-23
5	Hours Alarm (12 hour mode) Hours Alarm (24 hour mode)	01-12 (AM) 81-92 (PM) 00-23
6	Day of Week	01-07
7	Day of Month	01-31
8	Month	01-12
9	Year	00-99

Figure 17. Time, Calendar, Alarm Data Format

Figure 17 shows the format for the ten clock, calendar and alarm locations. The 24/12 bit in Register B determines whether the hour locations will be updated using a 1-12 or 0-23 format. After initalization the 24/12 bit cannot be changed without reinitalizing the hour locations. In 12 hour format the high order bit of the hours byte in both the time and alarm bytes will indicate PM when it is a "1".

During updates, which occur once per second, the 10 bytes of time, calendar and alarm information are unavailable to be read or written by the CPU for a period of 2ms. These 10 locations cannot be written during this time. Information read while the Real Time Clock is performing an update will be undefined. The Update Cycle section shows how to avoid Update Cycle/CPU contention problems.

The alarm bytes can be programmed to generate an interrupt at a specific time or they can be programmed to generate a periodic interrupt. To generate an interrupt at a specific time, the user need only program the time that the interrupt is to occur into the 3 alarm bytes. Alternately, a periodic interrupt can be generated by setting the high order two bits in an alarm register to a "1", which turns that byte into a "don't care". For instance, an interrupt can be generated every hour by programming a C0H into Register 5, or an interrupt can be generated once a second by programming the same value into all three alarm registers.

Static RAM

The 114 bytes of RAM from Index Address 0EH to 7FH are not affected by the Real Time Clock. These bytes are accessable during the update cycle and may be used for whatever the designer wishes. Typical applications will use this as nonvolatile storage for configuration and calibration parameters since this device is normally battery powered when the system in turned off.

Control and Status Registers

The 82C206 contains four registers used to control the operation and monitor the status of the Real Time Clock. These registers are located at Index Address 0AH-0DH and are accessable by the CPU at all times.

REGISTER A (0AH)

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
UIP	DV2	DV1	DV0	RS3	RS2	RS1	RS0

(Read/Write register except UIP)

UIP—Update in progress flag is a status bit used to indicate when an update cycle is about to take place. A "1" indicates that an update cycle is taking place or is imminent. UIP will go active (High) 244us prior to the start of an update cycle and will remain active for an additional 2ms while the update is taking place. The UIP bit is read only and is not affected by Reset. Writing a "1" to the SET bit in Register B will clear the UIP status bit.

DV2-DV0—These three bits are used to control the Divider/Prescaler on the Real Time Clock. While the 82C206 can operate at frequencies higher than 32.768 Khz, this is not recommended for battery powered operation due to the increased power consumption at these higher frequencies.

DV2	**DV1**	**DVO**	**OSCI Freq.**	**Mode**
0	0	0	4.194304MHz	Operate
0	0	1	1.048576MHz	Operate
0	1	0	32.768KHz	Operate
1	1	X	Reset Divider	

Divider Options

RS3-RS0—These four bits control the Periodic Interrupt rate. The Periodic interrupt is derived from the Divider/Prescaler in the Real Time Clock and is separate from the Alarm Interrupt. Both the alarm and periodic interrupts do however, use the same interrupt channel in the Interrupt Controller. Use of the Periodic Interrupt allows the generation of interrupts at rates higher than once per second. Below are the interrupt rates for which the Real Time Clock can be programmed.

Rate Selection				Time Base	
RS3	RS2	RS1	RS0	4.194304 MHz 1.048576 MHz	32.768 KHz
0	0	0	0	None	None
0	0	0	1	30.517 μs	3.90526 ms
0	0	1	0	61.035 μs	7.8125 ms
0	0	1	1	122.070 μs	122.070 μs
0	1	0	0	244.141 μs	244.141 μs
0	1	0	1	488.281 μs	488.281 μs
0	1	1	0	976.562 μs	976.562 μs
0	1	1	1	1.953125 ms	1.953125 ms
1	0	0	0	3.90625 ms	3.90625 ms
1	0	0	1	7.8125 ms	7.8125 ms
1	0	1	0	15.625 ms	15.625 ms
1	0	1	1	31.25 ms	31.25 ms
1	1	0	0	62.5 ms	62.5 ms
1	1	0	1	125 ms	125 ms
1	1	1	0	250 ms	250 ms
1	1	1	1	500 ms	500 ms

Periodic Interrupt Rate

REGISTER B (0BH)

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
SET	PIE	AIE	UIE	0	024/12DSE		

(Read/Write Register)

SET—Writing a "0" to this bit enables the Update Cycle and allows the Real Time Clock to function normally. When set to a "1" the Update Cycle is inhibited and any cycle in progress is aborted. The SET bit is not affected by the RESET input pin.

PIE—The Periodic Interrupt Enable Bit controls the generation of interrupts based on the value programmed into the RS3-RS0 bits of Register A. This allows the user to disable this function without affecting the programmed rate. Writing a "1" to this bit enables the generation of periodic interrupts. This bit is cleared to a "0" by Reset.

AIE—The generation of alarm interrupts is enabled by setting this bit to a "1". Once this bit is enabled the Real Time Clock will generate an alarm whenever a match occurs between the programmed alarm and clock information. If the don't care condition is programmed into one or more of the Alarm Registers, this will enable the generation of periodic interrupts at rates of one second or greater. This bit is cleared by Reset.

24/12—The 24/12 control bit is used to establish the format of both the Hours and Hours Alarm bytes. If this bit is a "1", the Real Time Clock will interpret and update the the information in these two bytes using the 24 hour mode. This bit can be read or written by the CPU and is not affected by Reset.

DSE—The Real Time Clock can be instructed to handle daylight savings time changes by setting this bit to a "1". This enables two exceptions to the normal time keeping sequence to occur. On the last Sunday in April AM. Set-
ting this bit to a "0" disables the execution of these two exceptions. PSRSTB has no affect on this bit.

REGISTER C (0CH)

msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
IRQF	PF	AF	UF	0	0	0	0

(Read only register)

IRQF—The Interrupt Request Flag bit is set to a "1" when any of the conditions which can cause an interrupt is true and the interrupt enable for that condition is true. The condition which causes this bit to be set, also generates an interrupt. The logic expression for this flag is:

IRQF = PF & PIE
+ AF & AIE
+ UF & UIE

This bit and all other active bits in this register are cleared by reading the register or by activating the PSRSTB/ input pin. Writing to this register has no affect on the contents.

PF—The Periodic Interrupt Flag is set to a "1" when a transition, which is selected by RS3-RS0, occurs in the divider chain. This bit will become active, independent of the condition of the PIE control bit. The PF bit will then generate an interrupt a nd set IRQF if PIE is a "1".

AF—A "1" appears in the AF bit whenever a match has occured between the time registers and alarm registers during an update cycle. This flag is also independent of it's enable (AIE) and will generate an interrupt if AIE is true.

REGISTER D (0DH)

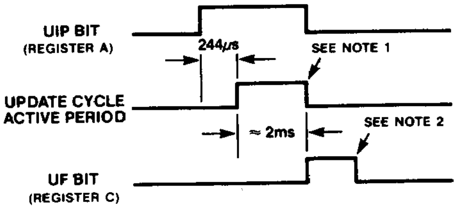
msb							lsb
b7	b6	b5	b4	b3	b2	b1	b0
VRT	0	0	0	0	0	0	0

(Read only register)

VRT—The Valid RAM and Time Bit indicates the condition of the contents of the Real Time Clock. This bit is cleared to a "0" whenever the PS input pin is LOW. This pin is normally derived from the power supply which supplies Vcc to the device and will allow the user to determine whether the registers have been initalized since power was applied to the device. $\overline{\text{PSRSTB}}$ has no affect on this bit and it can only be set by reading Register D.All unused register bits will be "0" when read and are not writable.

Update Cycle

During normal operation the Real Time Clock will perform an update cycle once every second. The performance of an update cycle is contingent upon the divider bits DV2-DV0 not being cleared, and the SET bit in Register B cleared. The function of the update cycle is to increment the clock/calendar registers and compare them to the Alarm Registers. If a match occurs between the two sets of registers, an alarm is issued and an interrupt will be issued if the alarm and interrupt control bits are enabled.



NOTE:
1. REGISTERS 0-9 ARE UNAVAILABLE TO BE READ OR WRITTEN DURING THIS TIME.
2. UF BIT CLEARED BY CPU READ OF REGISTER C.

Figure 18. Update Cycle

During the time that an update is taking place, the lower 10 registers are unavailable to the CPU. This is done to prevent the possible corruption of data in the registers or the reading of incorrect data. To avoid contention problems between the Real Time Clock and the CPU, a flag is provided in Register A to alert the user of an impending update cycle. This Update In Process Bit (UIP) is asserted 244μs before the actual start of the cycle and is maintained until the cycle is complete. Once the cycle is complete the UIP bit will be cleared and the Update Flag (UF) in Register C will be set. Figure 18 illustrates the update cycle. CPU access is always allowed to Registers A through D during update cycles.

Two methods for reading and writing to the Real Time Clock are recommended. Both of these methods will allow the user to avoid contention between the CPU and the Real Time Clock for access to the time and date information.

The first method is to read Register A, determine the state of the UIP bit and if it is "0", perform the read or write operation. For this method to work successfully the entire read or write operation (including any interrupt service routines which might occur) must not require longer than 244μs to complete from the beginning of the read of Register A to the completion of the last read or write operation to the Clock Calendar Registers.

The second method of accessing the lower 10 registers is to read Register C once and disregard the contents. Then subsequently continue reading this register until the UF bit is a "1". This bit will become true immediately after an update has been completed. The user then has until the start of the next update cycle to complete a read or write operation.

Power-Up/Down

Most applications will require the Real Time Clock to remain active whenever the system power is turned off. To accomplish this the user must provide an alternate source of power to the 82C206. This alternate source of power is normally provided by connecting a battery to the Vcc supply pin of the device. A means should be provided to switch from the system power supply to the battery. A circuit such as the one shown in Figure 19 may be used to eliminate power drain on the battery when the entire 82C206 is active. The circuit shown here will allow for reliable transitions between system and battery power without undue battery power drain.

The user should also ensure that the Vin maximum specification is never exceeded when powering the system up or down. Failure to observe this specification may result in damage to the device.

A pin is provided on the device to protect the contents of the Real Time Clock and reduce power consumption whenever the system is powered down. This pin (PWRGD) should be low whenever the system power supply is not within specifications for proper operation of the system. This signal may be generated by circuitry in either the power supply or on the system board. The PWRGD input will disable all unnecessary inputs during the time the system is powered down to prevent noise on the inactive pins from causing increased Icc. This pin must therefore be inactive for the remainder of the device to operate properly when system power is applied.

One pin is provided to initalize the device whenever power is applied to the 82C206. This pin (PSRSTB) will not alter the RAM or Clock/Calendar contents but it will initalize the necessary control register bits. (See Pin Description for a list of the control register bits affected by PSRSTB) Assertion of PSRSTB disables the generation of interrupts and sets a flag indicating that the contents of the device may not be valid. A recommended circuit for controlling the PSRSTB input is also shown in Figure 19.

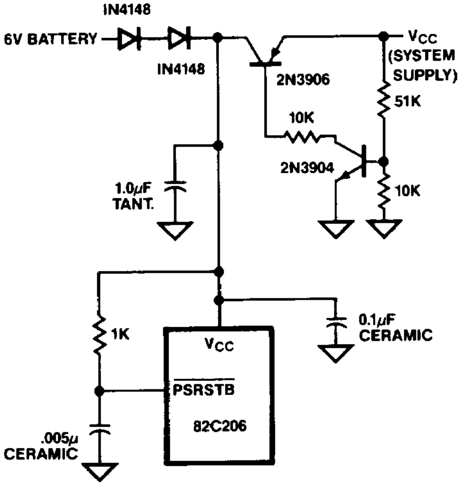


Figure 19. Power Conversion and Reset Circuitry

82C206 Absolute Maximum Ratings

Parameter	Symbol	Min.	Max.	Units
Supply Voltage	V_{CC}	—	7.0	V
Input Voltage	V_I	-0.5	5.5	V
Output Voltage	V_O	-0.5	5.5	V
Operating Temperature	T_{op}	-25	85	C
Storage Temperature	T_{stg}	-40	125	C

NOTE: Permanent device may occur of Absolute Maximum Ratings are exceeded. Functional operation should be restricted to the conditions described under Operating Conditions.

82C206 Operating Conditions

Parameter	Symbol	Min.	Max.	Units
Supply Voltage	V_{CC}	4.75	5.25	V
Ambient Temperature	T_A	0	70	C

DC Characteristics (T_A = 0-70° C, V_{CC} = 5 ± 5%)

Symbol	Parameter	Min.	Max.	Units	Test Conditions
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}+0.5	V	
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 2.0 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -2.0 mA
I_{OL}	Input Current	-10	10	μA	V_{IN} = V_{CC} to 0V
I_{OL}	Output Leakage Current	-10	10	μA	V_{OU} = V_{CC} to 0.45
I_{CC}	V_{CC} Supply Current		30	mA	CLK Freq = 8 MHz
I_{CCSB}	V_{CC} Standby Supply Current		10	μA	CLK Freq = DC

Capacitance (T_A = 25°C, V_{CC} = GND = 0V)

Symbol	Parameter	Min.	Max.	Units	Test Conditions
C_{IN}	Input Capacitance		10	pF	FC = 1 MHz
$C_{I/O}$	I/O Capacitance		20	pF	Unmeasured pins
C_{OUT}	Output Capacitance		20	pF	Returned to V_{SS}

AC Characteristics (T_A = 0–70°C, V_{CC} = 5V + 5%)

Symbol	Parameter	6 MHz Min.	Max.	8 MHz Min.	Max.	Units
t1	Address Setup to Command Active	25		25		nsec
t2	Command Active Period	250		200		nsec
t3	Address Hold Time from Command Inactive	0		0		nsec
t4	Data Valid Delay	200		160		nsec
t5	Data Hold Time from $\overline{\text{XIOR}}$ Inactive	10		10		nsec
t6	XD0-XD7 Active from $\overline{\text{XIOR}}$	5	40	5	40	nsec
t7	Data Setup to $\overline{\text{XIOW}}$ Inactive	200		160		nsec
t8	Data Hold Time from $\overline{\text{XIOW}}$ Inactive	0		0		nsec
t9	Command Recovery Time	150		120		nsec
t10	Interrupt Request width (Low)	120		100		nsec
t11	Interrupt Request width (High)	250		200		nsec
t12	INT Output Delay		400		300	nsec
t20	Real Time Clock Cycle Time		500		500	nsec
t21	AS Pulse Width	200		160		nsec
t22	Data Valid Setup to AS Inactive	200		160		nsec
t23	Data Hold Time from AS Inactive	0		0		nsec
t24	OSCI Period	500		500		nsec
t25	OSCI High Time	200		200		nsec
t26	OSCI Low Time	200		200		nsec
t27	PSRSTB High Delay from Vcc	5		5		μsec
t28	PSRSTB Low Pulse Width	5		5		μsec
t29	VRT Bit Valid Delay		2		2	μsec
t40	TMRCLK Period	200	DC	125	DC	nsec
t41	TMRCLK Low Time	80		50		nsec
t42	TMRCLK High Time	80		50		nsec
t43	GATE2 Setup to TMRCLK	80		50		nsec
t44	GATE2 Hold Time from TMRCLK	80		50		nsec
t45	GATE2 Low Time	80		50		nsec
t46	GATE2 High Time	80		50		nsec
t47	OUT2 Delay from TMRCLK		200		120	nsec
t48	OUT2 Delay from GATE2		200		120	nsec

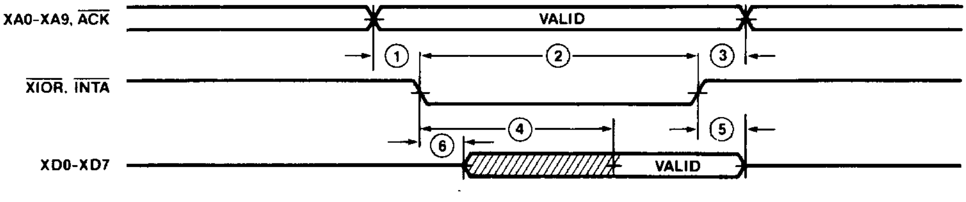
AC Characteristics (T_A = 0–70°C, V_{CC} = 5V + 5%) (Continued)

Symbol	Parameter	6 MHz		8 MHz		Units
		Min.	**Max.**	**Min.**	**Max.**	
t50	SCLK Period (1 X SCLK)	186		125		nsec
t50A	SCLK Period (2 X SCLK)	93		62		nsec
t51	SCLK Low Time (1 X SCLK)	75		43		nsec
t51A	SCLK Low Time (2 X SCLK)	32		22		nsec
t52	SCLK High Time (1 X SCLK)	82		55		nsec
t52A	SCLK High Time (2 X SCLK)	40		27		nsec
t53	DREQx Setup to SCLK	0		0		nsec
t54	HRQ Valid From SCLK		120		75	nsec
t55	HLDA Setup to SCLK	75		45		nsec
t56	AENx Valid Delay from SCLK		175		105	nsec
t57	AENx Invalid Delay from SCLK	TBD	130	TBD	80	nsec
t58	ADSTBx Valid Delay from SCLK		80		50	nsec
t59	ADSTBx Invalid Delay from SCLK		120		120	nsec
t60	XD0-XD7 Active Delay from SCLK		110		60	nsec
t61	XD0-XD7 Valid Setup to ADSTBx Low	80		65		nsec
t62	XD0-XD7 Hold Time from ADSTBx Low	25		25		nsec
t63	XD0-XD7 Tristate Delay from SCLK		170		135	nsec
t64	Address Valid Delay from SCLK		110		60	nsec
t65	Address Hold Time from DMAMEMR High	66		50		nsec
t66	Address Tristate Delay from SCLK		90		55	nsec
t67	DACKx Delay from SCLK		170		105	nsec
t68	Command Enable Delay from SCLK		150		90	nsec
t69	Command Active Delay from SCLK		190		120	nsec
t70	Write Cmnd Inactive Delay from SCLK		130		80	nsec
t71	Address Hold Time from Write High	116		75		nsec
t72	Command Tristate Delay from SCLK		120		75	nsec

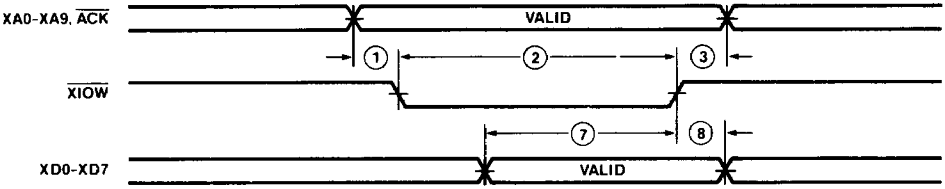
AC Characteristics (T_A = 0-70°C, V_{CC} = 5V + 5%) (Continued)

Symbol	Parameter	6 MHz Min.	Max.	8 MHz Min.	Max.	Units
t73	Read Cmnd Inactive Delay from SCLK		190		115	nsec
t74	TC Delay from SCLK		100		60	nsec
t75	XD0-XD7 Setup to Read Cmnd Inactive	155		90		nsec
t76	XD0-XD7 Hold from Read Cmnd Inactive	0		0		nsec
t77	XD0-XD7 Valid Delay from SCLK		190		120	nsec
t78	XD0-XD7 Hold from Write Inactive	15		15		nsec
t79	IOCHRDY Input Setup to SCLK	50		35		nsec
t80	IOCHRDY Input Hold Time from SCLK	35		20		nsec

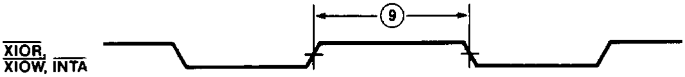
82C206 TIMING DIAGRAMS



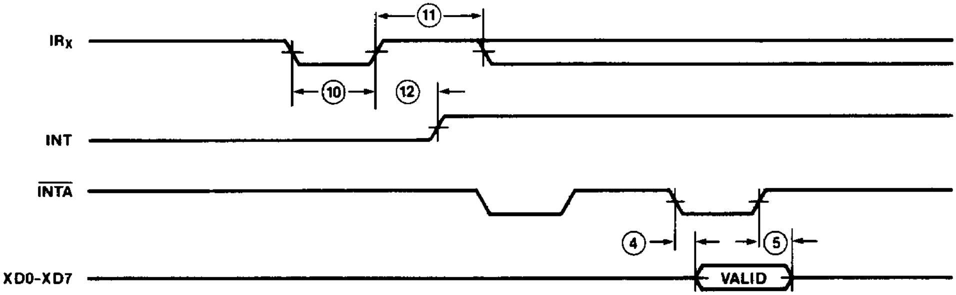
PERIPHERAL READ/INTA CYCLE



PERIPHERAL WRITE CYCLE

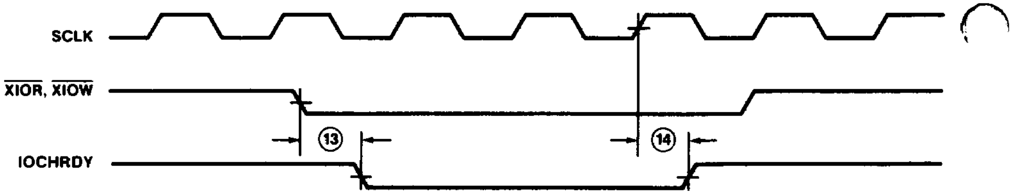


COMMAND RECOVERY

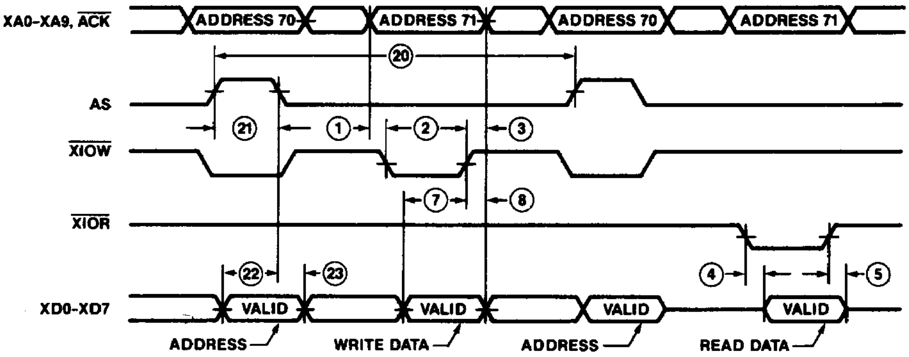


INTA SEQUENCE

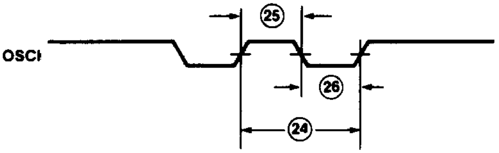
82C206 TIMING DIAGRAMS

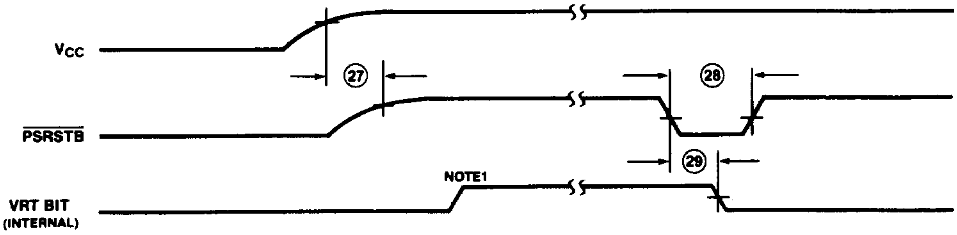


IOCHRDY OUTPUT



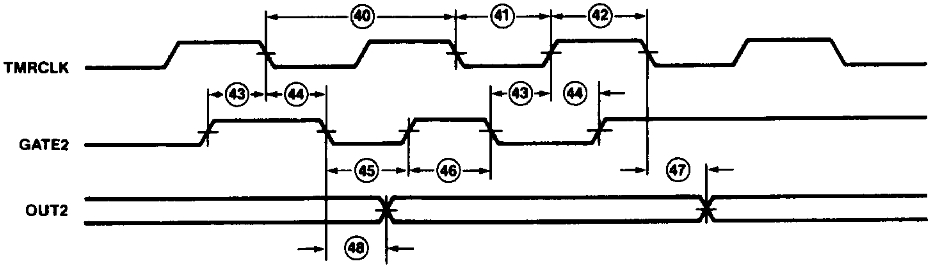
REAL TIME CLOCK ACCESS CYCLE



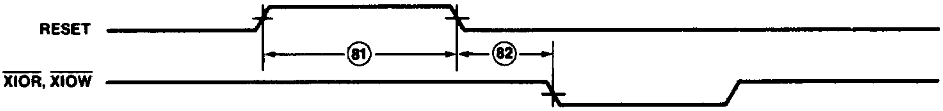


REAL TIME CLOCK POWER-UP SEQUENCE

82C206 TIMING DIAGRAMS

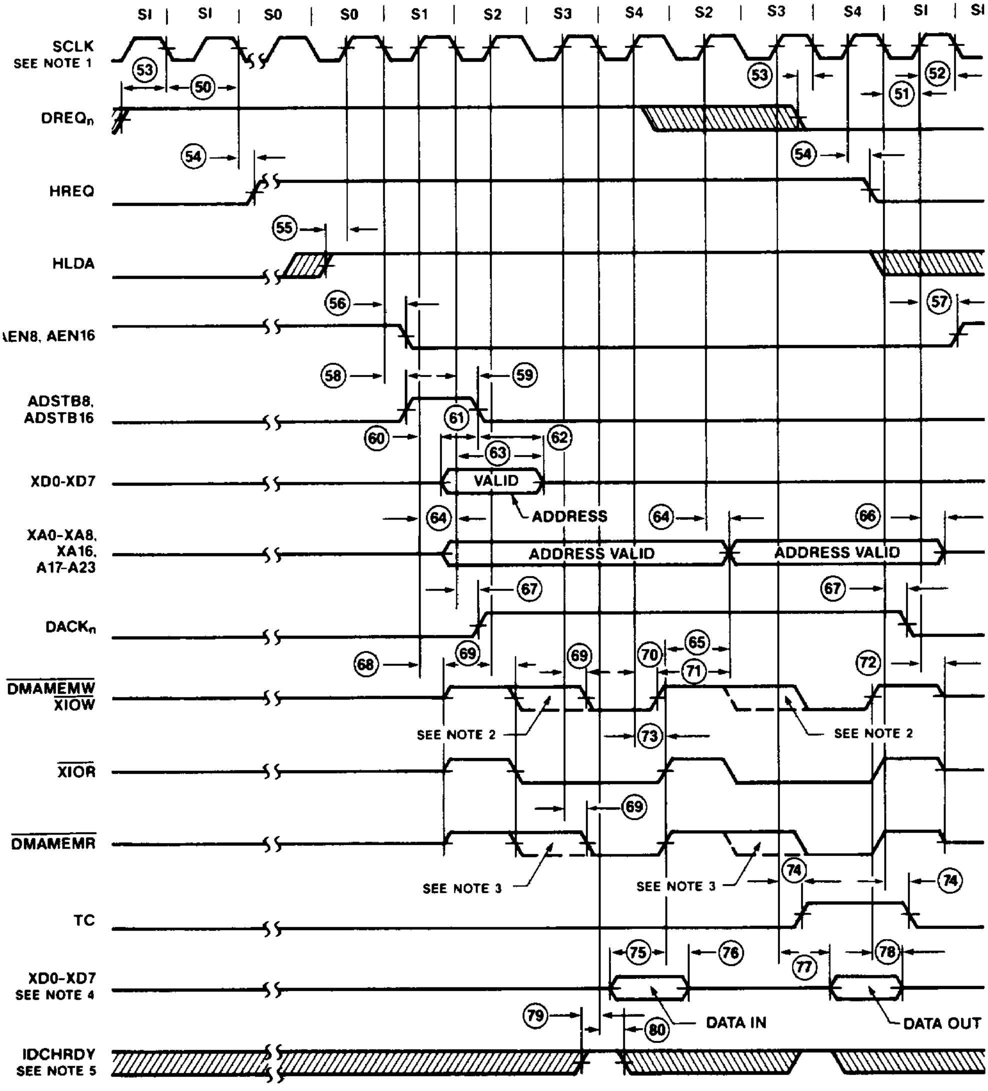


COUNTER/TIMER PARAMETERS



DMA RESET

82C206 TIMING DIAGRAMS



NOTES:

1. All timings referenced to SCLK are independent of the state of the clock select bit in the configuration register. SCLK shown in this diagram is the undivided clock directly from the input
2. Extended Write mode selected
3. Extended Read mode selected
4. Data Bus during memory to Memory Transfer
5. IOCHRDY Input Timing

Load Circuit Measurement Conditions

Parameter	Output Type	Symbol	C_L(pF)	R_1 (Ω)	R_L (Ω)	SW_1	SW_2
Propagation Delay Time	Totem pole 3-state Bidirectional	t_{PLH} t_{PHL}	50	—	1.0K	OFF	ON
Propagation Delay Time	Open drain or Open Collector	t_{PLH} t_{PHL}	50	0.5K	—	ON	OFF
Disable Time	3-state Bidirectional	t_{PLZ} t_{PHZ}	5	0.5K	1.0K	ON OFF	ON
Enable Time	3-state Bidirectional	t_{PZL} t_{PZH}	50	0.5K	1.0K	ON OFF	ON ON

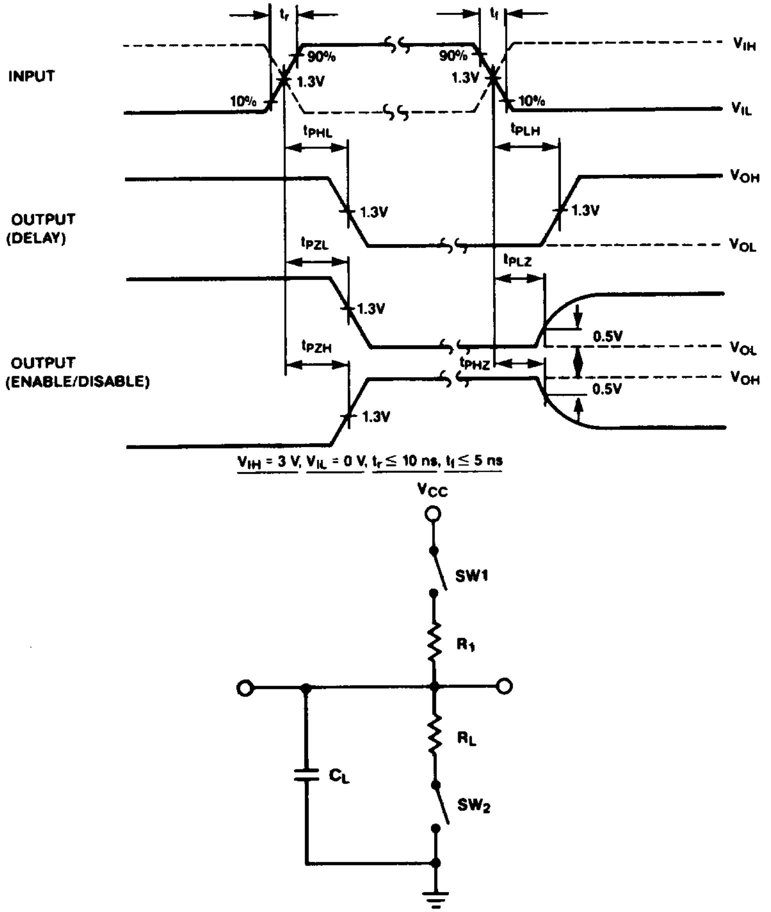
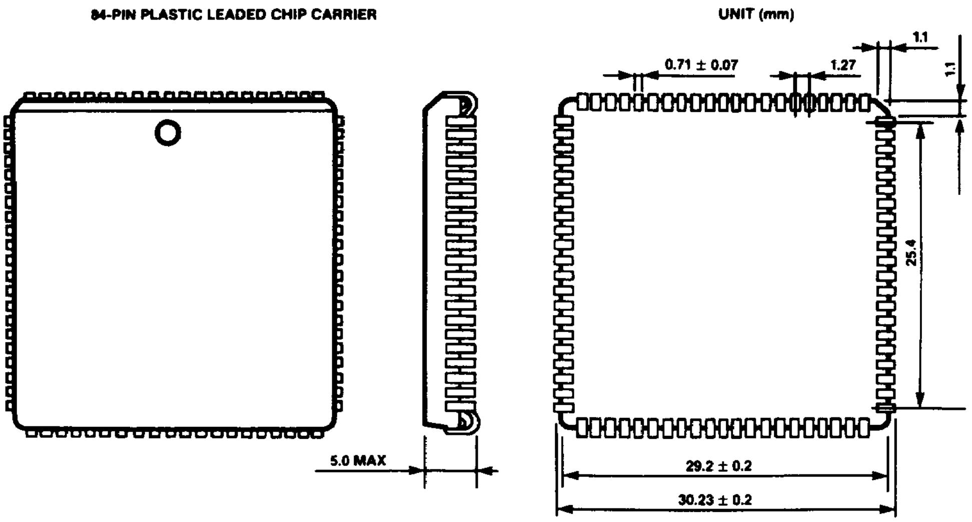


Figure 10. Load Circuit and AC Characteristics Measurement Waveform



Ordering Information

Order Number	Package Type	Operating Range
P82C206	PLCC-84	C (Note 1)

NOTE:

1. PLCC-84 Plastic Leaded Chip Carrier 84-Pin Package
   C = Commercial Range, 0-70°C, ±5% Supply Voltage

CHIPS

Chips and Technologies, Incorporated
3050 Zanker Road, San Jose, CA 95134 408-434-0600 Telex 272929 CHIP UR

IBM, AT, XT, PS/2, Micro Channel, Personal System/2, Enhanced Graphics Adapter, Color Graphics Adapter, IBM Color Display, IBM Monochrome Display are trademarks of International Business Machines.

Intel, iAPX 386 are trademarks of Intel Corporation.

Motorola is a trademark of Motorola.

Hercules Graphics is a trademark of Hercules Computer Technology.

Lotus is a trademark of Lotus Corporation.

Microsoft is a trademark of Microsoft.

CHIPSet, CHIPSpak, CHIPSport, CHIPS*LINK*, NEAT are trademarks of Chips and Technologies, Inc.

THESE DATA SHEETS ARE PROVIDED FOR THE GENERAL INFORMATION OF THE CUSTOMER. CHIPS AND TECHNOLOGIES, INC. RESERVES THE RIGHT TO MODIFY THESE PARAMETERS AS NECESSARY AND CUSTOMER SHOULD ENSURE THAT IT HAS THE MOST RECENT REVISION OF THE DATA SHEET. THE CUSTOMER SHOULD BE ON NOTICE THAT THE FIELD OF PERSONAL COMPUTERS IS THE SUBJECT OF MANY PATENTS HELD BY DIFFERENT PARTIES. CUSTOMERS SHOULD ENSURE THAT THEY TAKE APPROPRIATE ACTION SO THAT THEIR USE OF THE PRODUCTS DOES NOT INFRINGE ANY PATENTS. IT IS THE POLICY OF CHIPS AND TECHNOLOGIES, INC. TO RESPECT THE VALID PATENT RIGHTS OF THIRD PARTIES AND NOT TO INFRINGE OR ASSIST OTHERS TO INFRINGE SUCH RIGHTS.