PRELIMINARY

UM82C206

INTEGRATED PERIPHERAL CONTROLLER

FEATURES

- Fully compatible with PC/AT architecture
- Fully compatible with 8237 DMA controller, 8259 interrupt controller, 8254 timer/counter, and 146818 real time clock
- Provides 7 DMA channels, 13 interrupt request channels, 2 timer/counter channels, and a real time clock
- Built in 74LS612 memory mapper for DMA page address
- Provides 114 bytes of CMOS RAM memory
- 8 MHz DMA clock with programmable internal divider for 4 MHz operation
- 16M byte DMA address space
- Programmable wait states for the DMA cycle
- Reduced recovery time (120ns) between I/O operations

GENERAL DESCRIPTION

The UM82C206 Integrated Peripheral Controller includes two 8237 DMA controllers, two 8259 interrupt controllers, one 8254 timer/counter, one MC146818 compatible real time clock,an additional 64 bytes CMOS RAM, one 74LS612 memory mapper, and some top level decoder/configuration logic circuits. It is a single chip integration of all main peripheral parts attached to the X bus of PC/AT architecture. While providing full compatibility with PC/AT architecture, the UM82C206 also offers some enhanced features and improved speed performance. These include an additional 64 bytes of user definable CMOS RAM in real time clock and drastically reduced recovery time for the 8237, 8259 and 8254. Programmable wait state option is provided for the DMA cycles and CPU I/O cycles accessing this chip. This chip also provides programmable 8 or 4 MHz DMA clock selection. The UM82C206 is implemented using advanced 1.5u CMOS design technology and is packaged in an 84-pin PLCC.

PIN CONFIGURATION

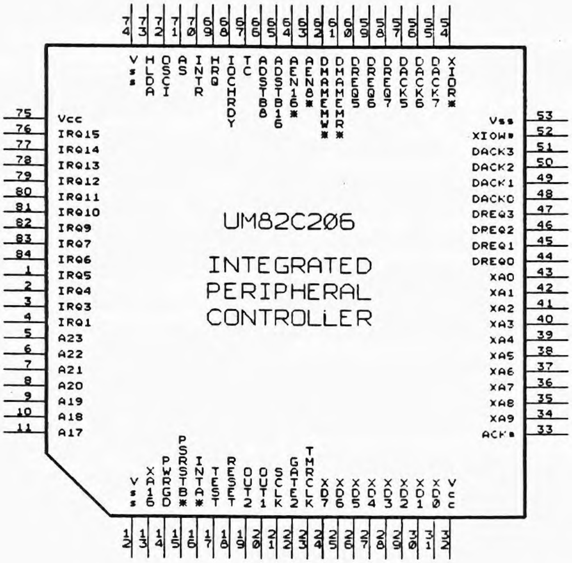


Figure 1. Pin Assignment

BLOCK DIAGRAM

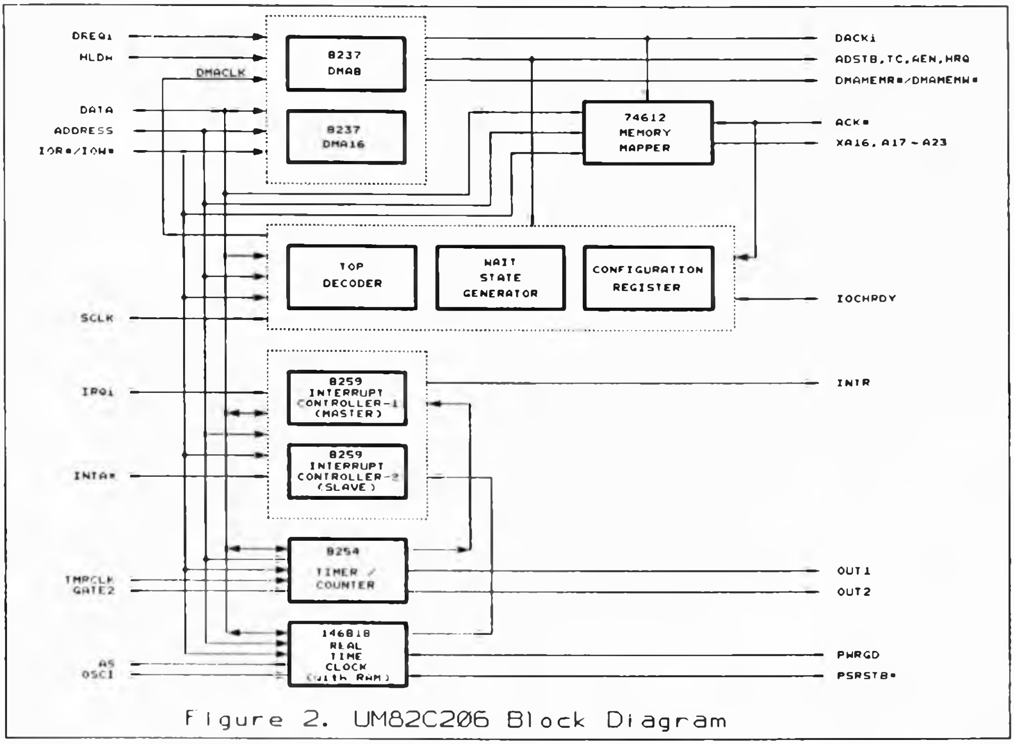


Figure 2. UM82C206 Block Diagram

PIN DESCRIPTION

Pin No.	Symbol	I/O	Description
Clock and Control			
21	SYSCLK	I	CLOCK INPUT is used to generate the timing signals for DMA operation. This pin can be driven to 10 MHz frequency. The internal clock used for DMA operation is either SYSCLK or SYSCLK/2 which is a programmable option.
72	OSCI	I	OSCILLATOR INPUT is used to generate the time base for the time function of real time clock. External square waves of 32.768 KHz may be connected to this pin.
18	RESET	I	RESET is an active high input which affects the following registers: DMA controller: Clears the command, status, request, temporary registers, byte pointer flip flop. Sets the mask register. Following reset, DMA controller is in the idle state. INTERRUPT controller: Clears the edge sense circuit, mask registers, all ICW4 functions. IRQ0 is assigned the highest priority. Slave address is set to 7. Special mask mode is disable and status read is set to IRR.
68	IOCHRDY	I/O	I/O CHANNEL READY is a bidirectional pin. In the input mode, it is used to extend the memory read or write pulses for the DMA controller to access slow memories or I/O devices. It must satisfy setup and hold times with respect to the DMA internal clock in order to work reliably. 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 signal goes high. In the output mode, it is an open drain output and provides an active low output whenever a UM82C206 internal register is accessed. It will remain low for a pre-programmed number of DMA internal clock cycles (as controlled by bits 7 and 6 of UM82C206 configuration register) and then goes high. In this way, IOCHRDY can insert wait states (as counted by DMA internal clock cycles) when CPU accesses the UM82C206 internal registers. This pin must be pulled up by an external resistor. In a PC/AT architecture based design this pin should be wire-ORed to the PC/AT's IOCHRDY signal.

Pin No.	Symbol	I/O	Description
24-31	XD7-XD0	I/O	X DATA BUS are 3-state bidirectional pins which are connected to the XD bus in PC/AT architecture design. During CPU I/O read cycles, these they are output pins to read out the contents of UM82C206 internal registers. During CPU I/O write cycles, these are input pins to let CPU program the contents of UM82C206 internal registers. During DMA cycles, the most significant 8 bits of the address are ouput onto these pins to be strobed into an external latch by ADSTB8 or ADSTB16. During DMA memory-to-memory transfers, data from the memory comes into the DMA controller via these pins and stores in the internal temporary register during read from the memory partial cycle. In the write to memory partial cycle, the data stored in the temporary register will output via these pins again and write into the new memory location. During the interrupt acknowledge cycle, the interrupt controllers output the interrupt vector byte via these pins. These pins are also used as the multiplexed address/data bus for the real time clock and the CMOS RAM accesses.
35-43 34	XA8-XA0 XA9	I/O I	X ADDRESS BUS are connected to the XA bus in PC/AT architecture design. XA8-XA0 pins are bidirectional pins. XA9 is an input only pin. During CPU I/O accesses to the UM82C206, XA9-XA0 are used to address configuration register and the internal registers of 8237s, 8259s, 8254, MC146818, CMOS RAM, 74LS612. During a CPU cycle, XA3-XA0 pins are used by the CPU to address the registers of the DMA controller corresponding to DMA channels 0-3. XA4-XA1 pins are used by the CPU to address the registers of the DMA controller corresponding to DMA channels 5-7. During a DMA cycle, XA7-XA0 pins are outputs and carry address information for DMA channels 0-3. XA8-XA1 pins are outputs and carry address information for DMA channels 5-7.

Pin No.	Symbol	I/O	Description
54	XIOR*	I/O	X I/O READ is a bidirectional active low 3-state pin. In a non DMA or non interrupt cycle, it is an input control signal used by the CPU to read the UM82C206 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 memory transfer.
52	XIOW*	I/O	X I/O WRITE is a bidirectional active low 3 state pin. In a non DMA or non interrupt cycle, it is an input control signal used by the CPU to write the UM82C206 internal registers. In an active DMA cycle, it is an output control signal used by the DMA controller to write data to a peripheral during a DMA read memory transfer.
61	DMAMEMR*	O	DMA MEMORY READ is an active low 3-state output pin used to access data from the selected memory location during DMA read memory or memory-to- memory transfer.
62	DMAMEMW*	O	DMA MEMORY WRITE is an active low 3-state output pin used to write data to the selected memory location during DMA write memory or mcmory-to- memory transfer.
73	HLDA1	I	HOLD ACKNOWLEDGE 1 is an active high signal from the UM82C211C to indicate that the CPU has relinquished control of the system busses.
69	HRQ	O	HOLD REQUEST is an active high output to the UM82C211C 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 to the UM82C211C. After CPU releases the system bus, the UM82C211C then issues a HLDA1 back to the UM82C206 if DMA has been permitted to control the system bus.
44-47 60-58	DREQ0-DREQ3 DREQ5-DREQ7	I	DMA REQUEST is an asynchronous DMA channel request input for each DMA channel. In fixed priority, DREQ0 has the highest priority and DREQ7 has the lowest priority. A periphal device will activate a DREQ line if it needs a DMA service. DACK will acknowledge the recognition of DREQ request. DREQ must be maintained until the corresponding DACK goes active. DREQ will not be recognized while the DMA clock is stopped. Unused DREQ inputs should be kept inactive and the corresponding mask bit should be set to avoid an undesired DMA function. Polarity of DREQ is programmable. Reset initializes these lines to active high. DREQ0-DREQ3 support 8-bit transfers between 8-bit I/O device and 8 or 16-bit system memory. DREQ5-DREQ7 support 16-bit transfers between 16-bit I/O device and 16-bit system memory. DREQ4 is not externally available and is used to cascade DREQ0-DREQ3.

Pin No.	Symbol	I/O	Description
67	TC	O	TERMINAL COUNT is an active high signal. It indicates the completion of DMA services. A pulse is generated by the DMA controller when terminal count for any channel is reached except for channel 0 in memory-to-memory transfer mode. During memory-to-memory transfer terminal count will be generated when the terminal count 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 register will be set for the currently active channel unless the channel is programmed for auto-initialization. In that case, the mask bit remains clear.
48-51 57-55	DACK0-DACK3 DACK5-DACK7	O	DMA ACKNOWLEDGE is used to notify the individual peripherals when one has been granted a DMA cycle. Because these signals are used internally for cascading the DMA channels and for DMA page register selection, they must be programmed to active low and cannot be changed. Reset initializes them to active low.
66	ADSTB8	O	ADDRESS STROBE 8 is an active high output. It is used to latch the upper address byte XA8-XA15 for 8-bit peripheral devices. During DMA block transfers, ADSTB8 will only be issued when the upper address byte must be updated, thus speeding transfer through elimination of S1 states of DMA cycles. ADSTB8 is active for DMA channels 0-3.
65	ADSTB16	O	ADDRESS STROBE 16 is an active high output. It is used to latch the upper address byte XA9-XA16 for 16-bit peripheral devices. During DMA block transfers, ADSTB16 will only be issued when the upper address byte must be updated, thus speeding transfer through elimination of S1 states of DMA cycles. ADSTB16 is active for DMA channels 5-7.
63	AEN8*	O	ADDRESS ENABLE 8 is an active low output. It is used to enable the latch of the upper address byte XA8-XA15 for 8-bit peripheral devices. It is inactive when external bus master controls the system bus. AEN8* is active for DMA channels 0-3.
64	AEN16*	O	ADDRESS ENABLE 16 is an active low output. It is used to enable the latch of the upper address byte XA9-XA16 for 16-bit peripheral devices. It is inactive when external bus master controls the system bus. AEN16* is active for DMA channels 5-7.

Pin No.	Symbol	I/O	Description
33	ACK• (MSE)	I	MODULE SELECT ENABLE is a two purpose input. When high, it enables the chip select function on one of the modules of UM82C206 for the CPU programming functions. When low, the UM82C206 is essentially disconnected from the system bus and is capable of performing an active DMA or an interrupt cycle. In a PC/AT architecture design, it is tied to ACK• signal of main board.
11-5 13	A23-A17 XA16	O	A23-A17 and XA16 are 3-state output pins. A23- A17 are the upper 7 bits of the DMA page register. XA16 is the least significant bit of the DMA page register and is used for DMA transfers for 8-bit peripheral devices only. XA16 is not used for 16-bit DMA transfers as XA16-XA9 being provided by demultiplexing the data bus.
76- 82 83 84 1- 3 4	IRQ15- IRQ9 IRQ7 IRQ6 IRQ5- IRQ3 IRQ1	I	INTERRUPT REQUESTS are asynchronous inputs. When 8259 is operating in edge triggered mode, an interrupt request is executed by raising an IRQ input low to high and holding it high until it is acknowledged by CPU. When 8259 is operating in level triggered mode, an interrupt request is executed by raising an IRQ input high and holding it high until it is acknowledged by CPU.
16	INTA•	I	INTERRUPT ACKNOWLEDGE is an active low input. It is used to enable the interrupt controllers to output the vector data on to the data bus by an interrupt acknowledge sequence from the CPU.
70	INTR	O	INTERRUPT REQUEST is an active high output pin. It is connected to the CPU's interrupt pin and is used to interrupt the CPU when an interrupt request occurs.
23	TMRCLK	I	TIMER CLOCK is an input clock for 8254 counter 0, counter 1 and counter 2. In PC/AT architecture design, it is approximately 1.19 MHz.
22	GATE2	I	GATE 2 is a gate input for 8254 counter 2. In PC/AT architecture design, the counter 2 is used for tone generation for speaker. It is driven by bit 0 of I/O port 61h.
20	OUT1	O	OUT 1 is an output of 8254 counter 1. In PC/AT architecture design, the counter 1 is programmed as a rate generator to produce a 15 usec period signal for DRAM refresh.
19	OUT2	O	OUT 2 is an output of 8254 counter 2. In PC/AT architecture design, counter 2 is used for tone generation for speaker.

Pin No.	Symbol	I/O	Description
71	AS	I	ADDRESS STROBE is an active high input. It is pulsed by UM82C211C when CPU accesses the real time clock or CMOS RAM of the UM82C206. The falling edge of this pulse latches the address from the XD bus.
15	PSRSTB*	I	POWER SUPPLY STROBE is an active low input. It is used to establish the condition of the control registers of real time clock when power is applied to the device. In PC/AT architecture design, it should be tied to the battery back-up circuit. When 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 bit is cleared to zero.
14	PWRG	I	POWER GOOD is an active high input and is connected to the power good of the power supply in PC/AT architecture design. It must be high for bus cycles in which the CPU accesses the real time clock. When it is low, all address, data, data strobe and R/W pins are disconnected from the processor.
17	TEST	I	TEST is an active high input to enable the chip testing for production. It should be tied low for normal operation.
32,75	VDD		POWER SUPPLY
12,53, 74	VSS		GROUND

FUNCTIONAL/REGISTER DESCRIPTION

TOP LEVEL DECODER AND CONFIGURATION REGISTER

The UM82C206 top level decoder provides 8 separate enables to various subsystems of the device. Table 1. contains a truth table for the top level decoder. The enabling of the UM82C206 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 XIOR* signal is also asserted. The decoder is enabled by signals ACK*, XA9, XA8. To enable any internal subsystem ACK* must be '1' and both XA9 and XA8 must be '0'.

Table 1. UM82C206 Internal Decode

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	DMA8
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	DMA16
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

Configuration Register

Index register port : 22H

Data register port : 23H

Index : 01H

<table>
<tr><th>Bits</th><th>Function</th></tr>
<tr><td>7-6</td><td>These bits contain the information of wait states inserted when the CPU accesses the registers of UM82C206. Wait states are counted as SYSCLK clock cycles and are not affected by the DMA clock selection.
<table>
<tr><th>7</th><th>6</th><th>Register I/O R/W wait states</th></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>2</td></tr>
<tr><td>1</td><td>0</td><td>3</td></tr>
<tr><td>1</td><td>1</td><td>4 (default)</td></tr>
</table></td></tr>
<tr><td>5-4</td><td>These bits contain the information of wait states inserted in 16 bit DMA cycles. Further control of the DMA cycle length is available through the use of the IOCHRDY pin of the UM82C206. During DMA cycle this pin is used as an input to the wait state generation logic to extend the cycle if necessary.
<table>
<tr><th>5</th><th>4</th><th>16 bit DMA wait states</th></tr>
<tr><td>0</td><td>0</td><td>1 (default)</td></tr>
<tr><td>0</td><td>1</td><td>2</td></tr>
<tr><td>1</td><td>0</td><td>3</td></tr>
<tr><td>1</td><td>1</td><td>4</td></tr>
</table></td></tr>
<tr><td>3-2</td><td>These bits contain the information of wait states inserted in 8 bit DMA cycles. Further control of the DMA cycle length is available through the use of the IOCHRDY pin of the UM82C206. During DMA cycle this pin is used as an input to the wait state generation logic to extend the cycle if necessary.
<table>
<tr><th>3</th><th>2</th><th>8 bit DMA wait states</th></tr>
<tr><td>0</td><td>0</td><td>1 (default)</td></tr>
<tr><td>0</td><td>1</td><td>2</td></tr>
<tr><td>1</td><td>0</td><td>3</td></tr>
<tr><td>1</td><td>1</td><td>4</td></tr>
</table></td></tr>
<tr><td>1</td><td>EMR bit enables the early DMAMEMR* function. In IBM PC/AT design DMAMEMR* is delayed one clock cycle later than XMEMR*. If set to 1, it will start DMAMEMR* at the same time as XMEMR*. If set to 0, it will start DMAMEMR* as an IBM PC/AT design (default).</td></tr>
<tr><td>0</td><td>CLK bit selects the DMA clock. If this bit set to 0, the SYSCLK input is divided by two and is used to drive both the 8-bit and 16-bit DMA subsystems (default). If this bit is set to 1, the SYSCLK will directly drive the DMA subsystems. 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</td></tr>
</table>

DMA SUBSYSTEM

The UM82C206 contains two 8237 DMA controllers. Each controller is a four channel DMA device which will generate the memory addresses and control signals necessary to transfer data between a peripheral device and memory directly. The two DMA controllers are internally cascaded to provide four DMA channels for transfers to 8-bit peripherals (DMA8), and three channels for transfers to 16-bit peripherals (DMA16). The channel 0 of DMA16 provides the cascade interconnection of the two DMA devices, thereby maintaining PC/AT compatibility. Hereinafter, the description of the DMA subsystem pertains to both DMA8 and DMA16 unless otherwise noted.

DMA I/O Address Map

The I/O address map of the DMA subsystem of UM82C206 is listed in Table 2. The mapping is fully compatible with PC/AT architecture.

Table 2. DMA Subsystem I/O Address Map

ADDRESS		OPERATION		BYTE	REGISTER
DMA8	DMA16	XIOR*	XIOW*	POINTER	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

DMA8	DMA16	XIOR*	XIOW*	POINTER	FUNCTION
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	X	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

DMA Operation

During normal operation of the UM82C206, the DMA subsystem will be in either the idle condition, the program condition or the active condition. When DMA controller is in the idle condition, it only executes the SI idle state cycles. The DMA controller will remain in the idle condition unless it has been initialized to work and one of the DMA request pins has been asserted. In that case, the DMA controller will exit the idle condition and enter the active condition. The DMA controller will also exit the idle condition and enter the program condition when CPU attempts to access its internal registers.

Idle Condition

When no peripherals request service, the DMA subsystem will enter the idle condition and perform only the SI idle states. During this time the UM 82C206 will sample the DREQ input pins every clock cycle to determine if any peripheral is requesting a DMA service. The internal select from the top level decoder and HLDA1 input pin will also sample at the same time to determine if the CPU is attempting to access the internal registers. With either of the above two conditions satisfied, the DMA subsystem will exit the idle condition and enter the program condition or the active condition. Note that the program condition has priority over the active condition since a CPU cycle has already started before DMA has been granted use of the bus.

Program Condition

The DMA subsystem will enter the program condition whenever HLDA1 input pin is inactive and an internal select from top level decoder is active. During this time, the address lines XA0-XA3 become inputs if DMA8 is selected, or XA1-XA4 become inputs if DMA16 is selected. These address inputs are used to decode the DMA controller registers which are to be accessed. The XIOR* and XIOW* are used to select and time the CPU reads or writes. When DMA16 is selected, the XA0 is not used to decode and is ignored. Due to the large number and size of the internal registers of the DMA subsystem, an internal byte pointer flip/flop is used to supplement the addressing of the 16-bit word count and address registers. This byte pointer is used to determine the upper or lower byte of the word count and address registers. This byte pointer flip/flop is cleared by hardware RESET or a master clear command. It may also be set or cleared by the CPU's 'set byte pointer flip/flop' or clear byte pointer flip/flop' commands. There are special commands supported by the DMA subsystem in the program condition. These commands do not use the data bus but are derived from a set of addresses, the internal select and XIOR* or XIOW*. These commands are listed at the end of table 2. Erratic operation of the UM82C206 can occur if a request for service occurs on an unmasked DMA channel which is being programmed. The channel should be masked or the DMA should be disabled to prevent the UM82C206 from attempting to service a peripheral with a channel which is only partially programmed.

Active Condition

The DMA subsystem will enter the active condition whenever a software request occurs or a DMA request occurrs on an unmasked channel which has already been programmed. When a DREQ occurs and the corresponding mask bit is clear, or a software DMA request is made, the DMA subsystem issues HRQ to the UM82C211C. After CPU releases the system bus, the UM82C211C then issues a HLDA1 back to the UM82C206 if DMA has been permitted to control the system bus. After being granted control of the bus, the DMA subsystem will then begin a DMA transfer cycle. Take DMA read cycle as an example. After receiving a DREQ, the UM82C206 will issue a HRQ to the UM82C211C. Until a HLDA1 is returned, the DMA subsystem will remain in an idle condition. On the next clock cycle the DMA will exit idle and enter S0 state. During S0 the DMA will resolve priority and issue DACK on the highest priority channel which is requesting service. The DMA will then enter S1 state where the multiplexed addresses are output and latched. The DMA will then enter S2 State where the UM82C206 asserts DMAMEMR* command. The DMA will then enter S3 state where the UM82C206 asserts XIOW* command. The DMA will then remain in S3 until the wait state counter has expired and IOCHRDY is high. Note that at least one additional S3 will occur unless compressed timing is programmed. Once a ready condition is detected, the DMA will enter S4 where DMAMEMR* and XIOW* are deasserted. In compressed mode and demand mode, subsequent transfers will begin in S2 unless the intermediate addresses require updating. In these subsequent transfers the lower addresses are changed in S2.

Transfer Modes

There are four transfer modes supported by the DMA. They are single transfer mode, block transfer mode, demand transfer mode and cascade mode. The DMA can be programmed on a channel by channel basis to operate in one of these four modes.

Single Transfer Mode - In this mode the DMA will execute only one cycle at a time. DREQ must be held active until DACK becomes active in order to be recognized. If DREQ is held active throughout the single transfer, the UM82C206 will deassert HRQ and release the bus to the system once the transfer is complete. After HLDA1 has gone inactive the UM82C206 will again assert HRQ and execute another transfer on the same channel unless a request from a higher priority channel has been received. In single transfer mode the CPU is ensured of at least one full machine cycle execution between DMA transfers. Following each transfer the word count register is decreased and the address register is increased or decrased depending on the DEC bit of mode register. When the word count decrements from 0000H to FFFFH the terminal count bit in the status register is set and a pulse is output to TC pin. If the autoinitialization is selected, the channel will reinitialize itself for the next service. Otherwise, the DMA will set the cor_ responding DMA request bit mask and suspend transferring on that channel.

Block Transfer Mode - In this mode the DMA will begin transfers in response to either a DREQ or a software request. If DREQ starts the transfers, it need only be held active until DACK becomes active. The transfers will continue until the word count decrements from 0000H to FFFFH, at which time TC pin is pulsed and the terminal count bit of status register is set. Again, an autoinitialization will occur at the end of the last service if the channel has been programmed to do so.

Demand Transfer Mode - In this 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 capacity. 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 register. Once DREQ has been deasserted, higher priority channels are allowed to intervene. Reaching terminal count will result in the generation of a pulse on TC pin, the setting of the terminal count bit in the status register and autoinitialization if programmed to do so.

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 HLDA1 signals of the slave DMA devices. Once the master has received a HLDA1 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 HLDA1 from the CPU and DREQ on active channel. This prevents conflicts between the DMA devices.

Figure 3. shows the cascade interconnection for two levels of DMA devices. Note that channel 0 of DMA16 is internally connected for cascade mode to DMA8. Additional devices can be cascaded to the available channels in either DMA8 or DMA16 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 DMA8 and DMA16 to function correctly, the active low state of DACK* should not be modified. The first level device's DMA request mask bits will prevent second level cascaded devices from generating unwanted hold requests during the initialization process.

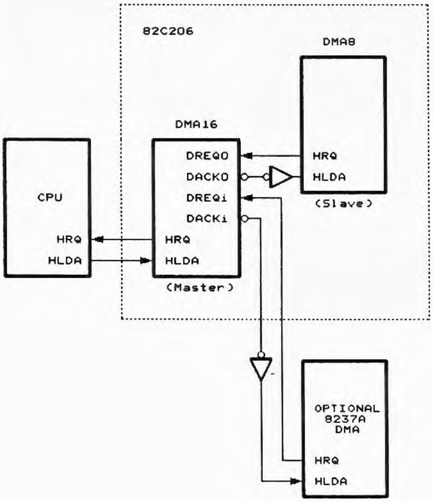


Figure 3. Cascade Mode Interconnect

Transfer Types

Single transfer mode, block transfer mode and demand transfer mode can perform any of the three transfer types. These three transfer types are read , write and verify transfers.

Read Transfers - These transfers move data from memory to an I/O peripheral by generating the memory address and asserting DMAMEMR* and XIOW* during the same transfer cycle.

Write Transfers - These transfers move data from an I/O peripheral to memory by generating the memory address and asserting DMAMEMW* and XIOR* during the same transfer cycle.

Verify Transfers - The verify transfers are pseudo transfers. In this type of transfer the DMA will operate as in read or write transfers by generating HRQ, DACK, memory addresses and respond to the terminal count. But it does not activate the memory and I/O command signals. Since no transfer actually takes place IOCHRDY is also ignored during verify transfers.

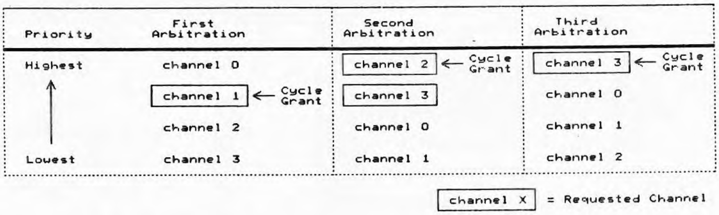
Memory-to-Memory Transfers - In addition to the above three transfer types, there is also a memory-to-memory transfer which can only be used on DMA channel 0 and channel 1. The memory-to-memory transfer is used to move a block of memory from one location in memory to another. DMA channels 0 and 1 may be programmed to operate as memory-to-memory channels by setting a bit in the DMA command register. Once programmed to do so the transfer can be started by generating either a software or an external request to channel 0. During the transfer, channel 0 provides the address for the source block during the memory read portion of the transfer, channel 1 generates the address for the destination block during the memory write portion of the same transfer. During the read portion transfer, a byte of data is latched in the internal temporary register of DMA. The contents of this register are then output on the XD0-XD7 output pins during the write portion of the transfer and subsequently written to memory location. Channel 0 may be programmed to maintain the same source address on every transfer. This allows the CPU to initialize large blocks of memory with the same value. The DMA subsystem will continue performing transfers until channel 1 reaches the terminal count.

Autoinitialization

The mode register of each DMA channel contains a bit which will cause the channel to reinitialize after reaching terminal count. During autoinitialization , the base address and base word count registers, which were originally programmed by the CPU, are reloaded into the current address and current word count registers. 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 remain cleared upon reaching terminal count. This allows the DMA to continue operation without CPU intervention. In 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.

DREQ Priority

The UM82C206 supports two types of priority schemes which are software programmable. They are fixed priority and rotating priority. Fixed priority assigns priority based on channel position. In this method channel 0 is assigned the highest priority and channel 3 is the lowest priority. After the recognition of any one channel for service, the other channels are prevented from interfering with that service until it is completed. In rotating priority, 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 4. In instances where multiple requests occur at the same time, the UM82C206 will issue a HRQ but will not freeze the priority logic until HLDA1 is returned. Once HLDA1 becomes active the priority logic is frozen and DACK is asserted on the highest requesting channel. Priority will not be reevaluated until HLDA1 has been deactivated.



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

channel X = Requested Channel

Figure 4. Rotating Priority Scheme

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 S1 state, 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 DMA8 is used for 8-bit transfers and DMA16 is used for 16-bit transfers, a 1-bit skew occurs in the intermediate address fields. DMA8 will therefore output addresses A8-A15 on the data bus at this time whereas DMA16 will output A9-A16. A separate set of latch and enable signals are provided for both DMA8 and DMA16 to accommodate the address skew.

During 8-bit DMA transfers, in which DAM8 is active, the UM82C206 will output the lower 8-bits 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, issused to control the output drivers of the external latch. A16-A23 are also generated at this time from a DMA page register in the UM82C206. Note that A16 is output on the XA16 pin of the device.

During 16-bit DMA transfers, in which DAM16 is active, the UM82C206 will output the lower 8-bits of address on XA1-XA8. The intermediate 8-bits of address A9-A16 will be output on XD0-XD7 and ADSTB16 will be asserted for one DMA clock cycle. The falling edge of ADSTB16 is used to latch the intermediate addresses A9-A16. An enable signal, AEN16, issued to control the output drivers of the external latch. A17-A23 are also generated at this time from a DMA page register in the UM82C206. Note that XA0 and XA16 remain 3-state during 16-bit DMA transfers.

The DMA page registers are a set of 16 8-bit registers in the UM82C206 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 maintaim PC/AT compatibility. Each DMA channel has a page register associated with it with the exception of channel 0 of DMA16 which is used for internal cascading to DMA8. Assignment of each of these registers is shown in Table 3. along with its CPU I/O read/write address.

During demand and block transfers, the UM82C206 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 UM82C206 will only update the latch contents when necessary. The UM82C206 will therefore only execute S1 state when necessary and improve the overall system throughput.

Table 3. DMA Page Register I/O Address Map

I/O ADDRESS	TYPE	REGISTER FUNCTION
080H	R/W	Unused
081H	R/W	DMA8 channel 2 (DACK2)
082H	R/W	DMA8 channel 3 (DACK3)
083H	R/W	DMA8 channel 1 (DACK1)
084H	R/W	Unused
085H	R/W	Unused
086H	R/W	Unused
087H	R/W	DMA8 channel 0 (DACK0)
088H	R/W	Unused
089H	R/W	DMA16 channel 2 (DACK6)
08AH	R/W	DMA16 channel 3 (DACK7)
08BH	R/W	DMA16 channel 1 (DACK5)
08CH	R/W	Unused
08DH	R/W	Unused
08EH	R/W	Unused
08FH	R/W	DRAM refresh cycle

Compressed Timing

The DMA subsystem in the UM82C206 can be programmed to transfer a word in as few as 2 DMA clock cycles. Normal transfers require 4 DMA clock cycles since S3 is executed twice due to the one wait state insertion. In systems capable of supporting higher throughput, the UM82C206 can be programmed to omit one S3 and assert both commands in S2. S2 begins the cycle by generat- ing the address and asserting both commands. One S3 cycle is executed and the cycle terminates in S4. If compressed timing is selected, TC 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. 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 UM82C206 will generate TC 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.

Command Register Format (Read/Write)

Bits	Function
7	DAK - Writing a "0" to this bit makes DACK an active low out- pin. Writing a "1" to this bit makes DACK an active high output pin.
6	DRQ - Writing a "0" to this bit makes DREQ an active high input pin. Writing a "1" to this bit makes DREQ an active low input pin.
5	EW - Writing a "1" to this bit enables Extended Write feature. It causes the write command to be asserted one DMA cycle earlier during a transfer. Thus read and write commands both begin in state S2 when enabled.
4	RP - Writing a "1" to this bit selects a Rotating Priority scheme for honoring DMA requests. The default condition is fixed priority.
3	CT - Writing a "1" to this bit enables the Compressed Timing. The default condition causes the DMA to operate with normal timing.
2	CD - Controller Disable. Writing a "1" to this bit disables the DMA subsystem (DMA8 or DMA16). This function is normally used whenever the CPU needs to reprogram one of the channels to prevent DMA cycles from occuring.
1	AH - Writing a "1" to these bit enables the address hold feature in Channel 0 when performing memory-to-memory transfer.
0	M-M - Writing a "1" to this bit 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 Mode Register gets written. The remaining six bits control the mode of the selected channel. Each channel 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, bit 0 and 1 will both be 1.

Mode Register Format (Read/Write)

<table>
<tr><th>Bits</th><th>Function</th></tr>
<tr><td>7-6</td><td>These bits contain the information of mode selection for each channel :
<table>
<tr><th>7</th><th>6</th><th>Function</th></tr>
<tr><td>M1</td><td>M0</td><td>Mode Select</td></tr>
<tr><td>0</td><td>0</td><td>Demand Mode</td></tr>
<tr><td>0</td><td>1</td><td>Single Cycle Mode</td></tr>
<tr><td>1</td><td>0</td><td>Block Mode</td></tr>
<tr><td>1</td><td>1</td><td>Cascade Mode</td></tr>
</table></td></tr>
<tr><td>5</td><td>DEC - Writing a "1" to this bit DECrements the address after each transfer.</td></tr>
<tr><td>4</td><td>AI - Writing a "1" to this bit enable Auto_Initialization function.</td></tr>
<tr><td>3-2</td><td>These bits control the type of transfer which is to be performed.
<table>
<tr><th>3</th><th>2</th><th>Function</th></tr>
<tr><td>TT1</td><td>TT2</td><td>Transfer Type Select</td></tr>
<tr><td>0</td><td>0</td><td>Verify Transfer</td></tr>
<tr><td>0</td><td>1</td><td>Write Transfer</td></tr>
<tr><td>1</td><td>0</td><td>Read Transfer</td></tr>
<tr><td>1</td><td>1</td><td>Illegal</td></tr>
</table></td></tr>
<tr><td>1-0</td><td>These bits determine which channel's Mode Register will be written. Read back of a mode register will cause these bits to both be "1".
<table>
<tr><th>1</th><th>0</th><th>Function</th></tr>
<tr><td>CS1</td><td>CS0</td><td>Channel Selection</td></tr>
<tr><td>0</td><td>0</td><td>select Channel 0</td></tr>
<tr><td>0</td><td>1</td><td>select Channel 1</td></tr>
<tr><td>1</td><td>0</td><td>select Channel 2</td></tr>
<tr><td>1</td><td>1</td><td>select Channel 3</td></tr>
</table></td></tr>
</table>

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 Register 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.

Request Register Write Format

<table>
<tr><th>Bits</th><th>Function</th></tr>
<tr><td>7-3</td><td>Don't care.</td></tr>
<tr><td>2</td><td>RB - Writing a "1" to this bit sets the Request Bit.</td></tr>
<tr><td>1-0</td><td>RS1-RS0 - Channel Request Select. These bits determine which channel's Request bit will be set.
<table>
<tr><th>1</th><th>0</th><th>Function</th></tr>
<tr><td>RS1</td><td>RS0</td><td>Channel Selection</td></tr>
<tr><td>0</td><td>0</td><td>select Channel 0</td></tr>
<tr><td>0</td><td>1</td><td>select Channel 1</td></tr>
<tr><td>1</td><td>0</td><td>select Channel 2</td></tr>
<tr><td>1</td><td>1</td><td>select Channel 3</td></tr>
</table>
</td></tr>
</table>

Request Register Read Format

Bits	Function
7-4	Always reads "1".
3-0	RC3-RC0 - These bits contain the state of the request bit associated with each Request Channel. 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.

Request Mask Register Set/Reset Format

Bits	Function
7-3	Don't care.
2	MB - Writing a "1" to this bit set the request Mask Bit and inhibits external requests.
1-0	MS1-MS0 - Channel request Mask Select. These bits determine which channel's Request Mask bit will be set.

1	0	Function
MS1	MS0	Channel Selection
0	0	select Channel 0
0	1	select Channel 1
1	0	select Channel 2
1	1	select Channel 3

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.

Request Mask Register Read/Write Format

Bits	Function
7-4	Always reads "1".
3-0	MB3-MB0 - These bits contain the state of the request Mask Bit associated with each request channel. The bit position corresponds to the channel number.

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.

Status Register Format (Read)

Bits	Function
7-4	DRQ3-DRQ0 - These bits show the status of each channel request, and are not affected by the state of the Mask Register bits. Reading "1" means "request" occurs, and bits 7,6,5,4 represent channels 3,2,1,0 respectively.These bits can be cleared by RESET, Master Clear or the pending request being deasserted.
3-0	TC3-TC0 - These bits indicate which channel has reached Terminal Count reading "1". These bits can be cleared by RESET, Master Clear or each time a Status Read takes place. The channel number corresponds to the bit position.

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.

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 XIOR* or XIOW*. For these special commands, the data bus is ignored by the 82C206 whenever an XIOW* activated command is issued. Data returned on XIOR* activated commands is undefined.

1. 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.

2. 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.

3. 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 Request Mask Register is set. Immediately following Master Clear or RESET, the DMA will be in the Idle Condition.

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

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

INTERRUPT SUBSYSTEM

INTERRUPT CONTROLLLER FUNCTIONAL DESCRIPTION

The programmable interrupt controllers in the UM82C206 function as a system wide interrupt manager in an iPAX86 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 recofigured 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 UM82C206. Each of the interrupt controllers is equivalent to an 8259A device operating in iPAX86 Mode. The two devices are interconnected and must be programmed to operate in Cascade Mode (see Figure 5) 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. Table 4 lists the 16 interrupt channels and their interrupt request source.

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 02H (0A0H).

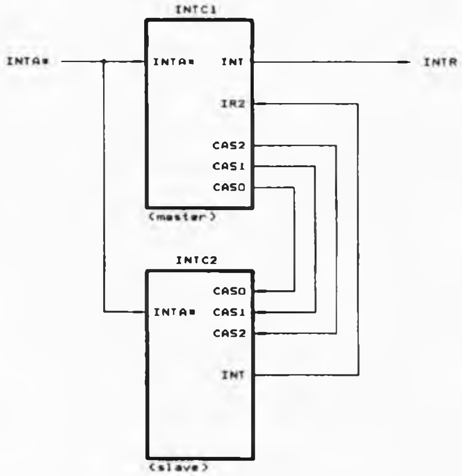


Figure 5. Internal Cascade Interconnect

Table 4. Interrupt Request Source

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

Controller Operation

Figure 6 is a block diagram of the major elements in the 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 Request (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.

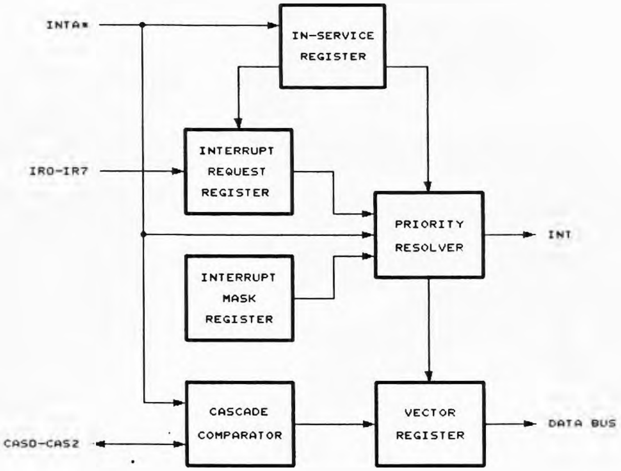


Figure 6. Interrupt Controller Block Diagram

Interrupt Sequence

The UM82C206 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 UM82C206 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 , see Figure 7). The events which occur during an interrupt sequence are as follows:

1 - One or more of the interrupt requests (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.

5 - The CPU will execute a second INTA cycle, during which the UM82C206 will drive an 8-bit vector onto the data pins XD7-XD0, which is read by the CPU. The format of this vector is shown in Table 5. Note that V7-V3 in Table 5 are programmable by writing to Initialization 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 to allows further interrupts. 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.

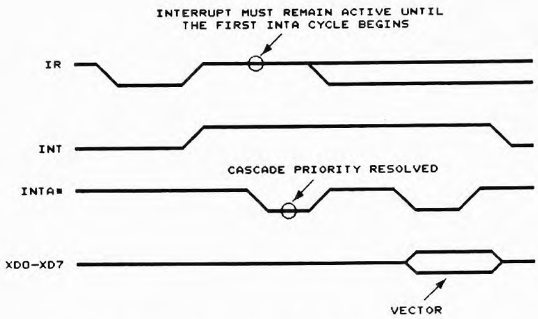

INTERRUPT MUST REMAIN ACTIVE UNTIL THE FIRST INTA CYCLE BEGINS
IR
INT
CASCADE PRIORITY RESOLVED
INTA*
XD0-XD7
VECTOR

Figure 7. Interrupt Sequence

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). The UM82C206 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 an 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 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 service routine will only be acknowledged if the CPU has internally re-enabled interrupts.

Specific Rotation Mode - Specific Rotatior 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 specificed)

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 (IR3 is highest priority request being serviced)

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

(SPEcific Rotation command issued with Channel 4 specificed)

After Rotation

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

Programming The Interrupt Controller

Two type of commands are used to control the UM82C206 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 docs 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 - IR0 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 8 for a flow chart of the initialization sequence. The initialization sequence can be terminated at any point (all 4 bytes must be written for the controller to be properly initialized) by writing to address 02H(0A0H) with a 0 in data bit 4. Note, this will cause OCW2 or OCW3 to be written.

Table 5. Interrupt Vector Byte

	D7	D6	D5	D4	D3	D2	D1	D0
IR7	V7	V6	V5	V4	V3	1	1	1
IR6	V7	V6	V5	V4	V3	1	1	0
IR5	V7	V6	V5	V4	V3	1	0	1
IR4	V7	V6	V5	V4	V3	1	0	0
IR3	V7	V6	V5	V4	V3	0	1	1
IR2	V7	V6	V5	V4	V3	0	1	0
IR1	V7	V6	V5	V4	V3	0	0	1
IR0	V7	V6	V5	V4	V3	0	0	0

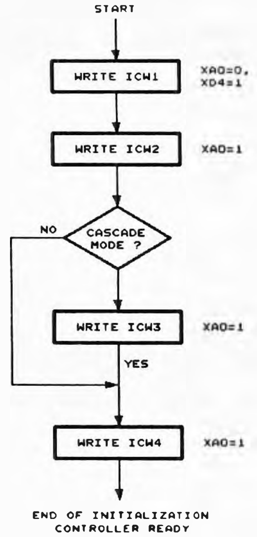


Figure 8. Initialization Sequence

ICW1 - Address 020H (0A0H)
- Write Only

Bits	Function
7-5	Don't care.
4	Must set "1" for ICW1, Since ICW1, OCW2 and OCW3 share the same address,(020H, 0A0H).
3	LTM - Bit 3 selects Level Triggered Mode or Edge triggered Mode input to the IR. If a "1" is written to LTM, a high level on the IR input will generate an interrupt request and the IR must be removed prior to EOI to prevent another interrupt. In Edge Triggered Mode, a low to high transition will generate an interrupt request. In either mode, IR must be held high until the first INTA cycle is started in order to generate the proper vector. IR7 vector will be generated if the IR input is deasserted early.
2	Don't care.
1	SM - This bit 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.
0	Don't care.

ICW2 - Address 021H (0A1H)
- Write Only

Bits	Function
7-3	V7-V3 - These bits are the upper 5 bits of the interrupt vector and are programmable by the CPU. INTC1 and INTC2 need not be programmed with the same value in ICW2, usually INTC1 is programmed with a value of 08H ,and INTC2 is programmed with a value of 70H.
2-0	The lower three bits of the vector are generated by the Priority Resolver during INTA (ses Table 5).

ICW3 - Format for INTC1 - Address 021H
- Write Only

Bits	Function
7-0	S7-S0 - Selects which IR inputs have Slave Mode controllers connected. ICW3 in INTC1 must be written with a 04H(IRQ2) for INTC2 to function correctly.

ICW3-Format for INTC1-Address 0A1H
-Write Only

Bits	Function
7-3	Don't care.
2-0	ID2-ID0 - Determines the Slave Mode address the controller will respond to during the cascade INTA sequence. ICW3 in INTC2 should be written with a 02H(IRQ2 of INTC1) for Cascade Mode operation.

ICW4 - Address 021H (0A1H)
- Write Only

Bits	Function
7-5	Don't care.
4	EMI - this bit 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 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.
3-2	Don't care.
1	AEOI - Auto End Of Interrupt is enabled when ICW4 is written with a one 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.
0	Don't care.

Operational Command Words

Operational Command Words (OCWs) allow the UM82C206 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)
- Read/Write

OCW2 - Address 020H (0A0H)
- Write Only

<table>
<tr><th>Bits</th><th>Function</th></tr>
<tr><td>7-5</td><td>R,SL,EOI - These bits selects operational function. Writing a "1" in bit 7 causes one of the rotate functions to be selected.
Writing a "1" in bit 6 causes a specific or immediate function to occur. All specific commands require L2-L0 to be valid except no operation.
Writing a "1" in bit 5 causes a function related to EOI to occur.
<table>
<tr><th>7</th><th>6</th><th>5</th><th>Function</th></tr>
<tr><td>R</td><td>SL</td><td>EOI</td><td>operational function</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Clear rotate in auto EOI mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Non-specific EOI Command</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>No operation</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Specific EOI Command *</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Set rotate in auto EOI mode</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Rotate on non-specific EOI Command</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Set priority Command *</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Rotate on specific EOI Command *</td></tr>
</table>
* L2-L0 are used by these commands.</td></tr>
<tr><td>4-3</td><td>These bits must be set "0" to indicate that OCW2 is selected, because ICW1, OCW2 and OCW3 share the same address, (020H, 0A0H).</td></tr>
<tr><td>2-0</td><td>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.</td></tr>
</table>

OCW3 - Address 020H (0A0H)
- Write Only

<table>
<tr><th>Bits</th><th>Function</th></tr>
<tr><td>7</td><td>This bit must be set "0".</td></tr>
<tr><td>6-5</td><td>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 - During Special Mask Mode, writing a 1 to any bit position of OCW1 inhibits interrupts and a 0 enables interrupts on the associated channel by causing the Priority Resolver to ignore the condition of the ISR.
<table>
<tr><th>6</th><th>5</th><th>Function</th></tr>
<tr><td>ESMM</td><td>SMM</td><td>Mask mode enable/select</td></tr>
<tr><td>0</td><td>X</td><td>No operation</td></tr>
<tr><td>1</td><td>0</td><td>Reset special mask mode to normal mask mode</td></tr>
<tr><td>1</td><td>1</td><td>Set special mask mode</td></tr>
</table>
</td></tr>
<tr><td>4-3</td><td>These bits must be set "0" to indicate that OCW3 is selected, because ICW1, OCW2 and OCW3 share the same address, (020H, 0A0H).</td></tr>
<tr><td>2</td><td>PM - Writing a "1" to this bit of OCW3 enables Polled Mode. Writing OCW3 with poll mode acts like the first INTA cycle, freezing all interrupt request lines and resolving priority. The next read operation to the controller acts like a second INTA cycle and polled vector is output to data bus. The format of polled vector is described later (see Poll Mode Read).</td></tr>
<tr><td>1-0</td><td>RR - Writing a "1" to this bit enable the contents of IRR or ISR (determined by RIS) to be placed on XD7-XD0 when reading the Status Port at address 020H (0A0H). Asserting PM forces RR reset.
<table>
<tr><th>1</th><th>0</th><th>Function</th></tr>
<tr><td>RR</td><td>RIS</td><td>select next read register</td></tr>
<tr><td>0</td><td>X</td><td>No operation</td></tr>
<tr><td>1</td><td>0</td><td>Read IRR on next read</td></tr>
<tr><td>1</td><td>1</td><td>Read ISR on next read</td></tr>
</table>
</td></tr>
</table>

IRR, ISR and Poll Vector

IRR, ISR and Poll Vector share the same address (020H, 0A0H). The selection of the registers depends on the programming of ITC. If the latest OCW3 issued poll command, (PM=1), the poll vector is selected for the next read. Before another poll command is issued, subsequent read to the address will select IRR or ISR depending on the latest OCW3, if RR=1 and RIS=0, ISR is selected. Note that poll command is cleared after the first read to the ITC. After initialization (ICW1 or RESET), IRR is selected.

IRR-Address 020H(0A0H)

Bits	Function
7-0	IR7-IR0 - These bits corresponds to the interrupt request bit of Interrupt Request Register. A "1" on these bits indicate that an interrupt request is pending on the corresponding line.

ISR-Address 020H(0A0H)

Bits	Function
7-0	IS7-IS0 - These bits correspond to the interrupt service bit of Interrupt Service Register. A "1" on these bits indicate that an interrupt is being serviced on the corresponding line. EOI will clear corresponding IS bit of ISR.

Poll Vector-Address 020H(0A0H)

Bits	Function
7	INT - A "1" in these bits indicates that a pending interrupt is polled. If there is no pending interrupt request or the request is removed before the poll command, this bit is 0.
6-3	Don't care.
2-0	V2-V0 - These bits are the binary encoding of the highest priority level pending interrupt request being polled. If no pending interrupt has been polled, all three bits are equal to 1.

Many registers share the same I/O address of INTC. The following table summarizes the address of each register.

WRITE			
A0	XD4	XD3	Register Selected
0	0	0	OCW2
0	0	1	OCW3
0	1	X	ICW1
1	X	X	ICW2,ICW3,ICW4 during initialization sequence
			OCW1 (Mask Register) after initialization sequence

READ	
A0	Register Selected
0	IRR, ISR or Poll Vector
1	OCW1 (Mask Register)

COUNTER/TIMER SUBSYSTEM

COUNTER/TIMER FUNCTIONAL DESCRIPTION

The UM82C206 contains a 8254 compatible counter/timer(CTC). The counter/ timer can be used to generate accurate time delays under software control. It contains 3 16-bit counters (counters 0-2) which can be programmed to count in binary or binary-coded decimal (BCD). Each counter operates inde- pendently of the other and can be programmed for operation as a timer or a counter.

All counters in this subsystem are controlled by a common control logic as shown in Figure 9. The control logic decodes and generates the necessary commands 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 because their gate is hardwired to VCC internally. Counter 2 can be programmed to operate in any of the six modes as listed below:

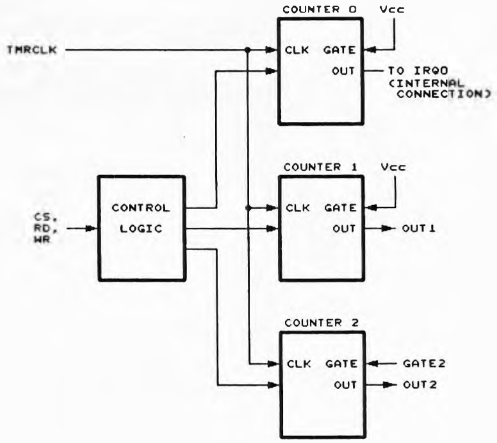


Figure 9. Counter/Timer Block Diagram

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 this subsystem are driven from a common clock input pin (TMRCLK) which is different from other clock inputs to the UM82C206. Counter 0's output (OUT0) is internally connected to IR0 of INTC1 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 external devices. Counter 2 is a full function counter/timer. It can be used as an interval timer, a counter, or as a gated rate/pulse generator. In PC/AT compatible design, counter 0 is used as a system timer, counter 1 is used as a DRAM refresh rate generator, and counter 2 is used for speaker sound generation.

Counter Description

Each counter in this subsystem contains a control register, a status register, a 16-bit counting element, a pair of 8-bit counter input latchs, and a pair of 8-bit counter output latches. Each counter shares the same clock input (TMRCLK). GATE0, GATE1 and OUT0 are not externally accessible. This is fully compatible with PC/AT. Output of OUT0 is dependent on the counter mode. (see Mode Definitions)

The control register stores the mode and command information used to control the counter. It may be loaded by writing a byte to the write control word at port 043H. The status register allows the software to monitor counter condition and read back the contents of the control register.

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

The counting element is indirectly loaded by writing one or two bytes (optional) to the counter input latches, which are in turn loaded into the counting element. Thus the counting element can be loaded or reloaded in one TMRCLK cycle. The counting element is also read indirectly by reading the contents of the counter output latches. The counter output latches are transparent latches which can be read while transparent or latched (see latch counter command).

Programming The Counter/Timer

After system reset the contents of control registers, counter registers, counting elements, and the output of all counters are undefined. Each counter must be programmed before it can be used. Each counter is programmed by writing its control register with a control word and then giving an initial count to its counting element. Table 6 lists the I/O address map used by the counter/timer subsystem.

Table 6. Counter/Timer I/O Address Map

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

Read/Write Counter Command

Each counter has a write only control register. This control register is written with a control word to the I/O address 043H. The control word format is described here.

Control Word Format (Write Only)

Bits	Function		
7-6	SC1-SC0 - Select which counter this control word is written to.		
	7	6	
	SC1	SC0	Function
	0	0	select counter 0
	0	1	select counter 1
	1	0	select counter 2
	1	1	reserved for read-back command
5-4	RW1-RW0 - Determine the counter read/write word size.		
	5	4	
	RW1	RW0	Function
	0	0	reserved for counter latch command
	0	1	read/write LSB only
	1	0	read/write MSB only
	1	1	read/write LSB first, then MSB

			MSB = most significant byte
			LSB = least significant byte

<table>
<tr><th>Bits</th><th>Function</th></tr>
<tr><td>3-1</td><td>M2-M0 - Select the counter operating mode.
<table>
<tr><th>3</th><th>2</th><th>1</th><th></th></tr>
<tr><td>M2</td><td>M1</td><td>M0</td><td>Function</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>select mode 0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>select mode 1</td></tr>
<tr><td>X</td><td>1</td><td>0</td><td>select mode 2</td></tr>
<tr><td>X</td><td>1</td><td>1</td><td>select mode 3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>select mode 4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>select mode 5</td></tr>
</table>
</td></tr>
<tr><td>0</td><td>BCD - During read/write counter commands control word writing, a "1" selects binary coded decimal counting format. A "0" selects binary counting format. During read-back command word writing, this bit must be 0.</td></tr>
</table>

When programming to a counter, the below sequences must be followed: First, each counter's control register must be written with a control word before the initial count is written. Second, writing the initial count must follow the 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.

Counter Latch Command

When a counter latch command is issued, the counter's output latches latch the current state of the counting element. The counter's output latches remain latched until read by the CPU or the counter is reprogrammed. After that the output latches then return to a "transparent" condition. Counter latch commands may be issued to more than one counter before reading the first counter to which this commannd was issued. Also, multiple counter latch commands issued to the same counter without reading the counter will cause all but the first command to be ignored. Below describes the counter latch command format.

Counter Latch Command Format (Write Only)

Bits	Function
7-6	SC1-SC0 - Select which counter is being latched.

7	6	
SC1	SC0	Function
0	0	select counter 0
0	1	select counter 1
1	0	select counter 2
1	1	reserved for read-back command

Bits	Function
5-4	These two bits must be zero for the counter latch command.
3-0	These four bits are don't care bits.

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 described below.

Read-Back Command Format (Write Only)

Bits	Function
7-6	These two bits must be "1" for the read-back command.
5	LCOUNT - A "0" in this bit will latch the count of the counting element of the selected counter(s).
4	LSTATUS - A "0" in this bit will latch the status information of the selected counter(s).
3-1	C2-0 - These three bits select which counter(s) the read-back command is applied to.

3	2	1	
C2	C1	C0	Function
0	X	X	select counter 2
X	0	X	select counter 1
X	X	0	select counter 0

Each counter's latches remain latched until either the latch is read or the counter is reprogrammed. If both LSTATUS and LCOUNT are "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. Multiple read-back commands issued to the same counter without reading the counter will cause all but the first command to be ignored. The status read from each counter is described below.

Status Format

Bits	Function
7	OUT - This contains the state of the OUT signal of the counter.
6	NC - This contains the condition of the null count flag. This flag is used to indicate that the contents of the counting element are valid. It will be set to a "1" during a write to the control register or the counter. It is cleared to a "0" whenever the counter is loaded from the counter input register.
5-4	RW1-0 - These two bits indicate the counter read/write word size. This information is useful in determining where the high byte, the low byte or both must be transferred during counter read/write operations.
3-1	M2-0 - These bits reflect the operating mode of the counter and are interpreted in the same manner as in the write control word format.
0	BCD - This bit indicates the counting element is operating in binary format or BCD format.

Counter Operation

Because counter 0 and counter 1 have limitations in some of their operating modes, we will use counter 2 to describe the various counter operating modes. But the description of modes 0, 2, 3 and 4 is suitable for all counters. The following terms are defined for describing counter/timer operation.

TMRCLK pulse - A rising edge followed by a falling edge of the UM82C206 TMRCLK input.

trigger - The rising edge of the GATE2 input.

counter load - The transfer of the 16-bit value in counter input latches to to the counting element.

initialized - A control word written and the counter input latches loaded.

Counter 2 can operate in one of the following modes:
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 triggered strobe

Mode 0 - Interrupt on terminal count

Mode 0 is usually used for event counting. After the counter being written with the control word, OUT2 of the counter goes low and remains low until the counting element 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 and disabled when GATE2 = 0. GATE2 has no effect on OUT2.

The counting element is loaded at the first TMRCLK pulse after the control word and initial count are loaded. When both initial count bytes are required, the counting element is loaded after the high byte is written. 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 counting element 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 progress until GATE2 = 1. When GATE2 goes high, OUT2 will go high after N TMRCLK pulses later.

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 counting element to be reloaded, extending the length of the pulse. Writing a new count to counter input latches will not affect the current one-shot pulse unless the counter is retriggered. In the latter case, the counting element is loaded with the new count and the one-shot pulse continues until the new count expires.

Mode 2 - Rate generator

This mode functions as a divide-by-N counter. After writing the control word during initialization the counter's OUT2 is set to 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 OUT2 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 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 golow 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".

Table 7. Gate Pin Function

Mode	Gate		
	Low	Rising	High
0	Disables Counting	---	Enables Counting
	---	a) Initiates Counting	---
	---	b) Resets Out Pin	---
2	a) Disables Counting	Initiates Counting	Enables Counting
	b) Forces Out Pin High		
3	a) Disables Counting	Initiates Counting	Enables Counting
	b) Forces Out Pin High		
4	Disables Counting	---	Enables Counting
5	---	Initiates Counting	---

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 flip-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 SUBSYSTEM

REAL TIME CLOCK FUNCTIONAL DESCRIPTION

This subsystem of the UM82C206 integrates a complete time-of-day real time clock with alarm, one hundred year calendar, a programmable periodic interrupt, and 114 bytes of CMOS static RAM. The UM82C206 is designed to operate in a low power (battery powered) mode and protects the contents of both the CMOS static RAM and clock from change during system power up and down.

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 UM82C206. This alternate source of power is normally provided by connecting a battery to the Vcc pin to switch from the system power supply to the battery. A circuit implementing such a function is shown in Figure 10. It is used to eliminate power drain on the battery when the entire UM82C206 is active. It will also make a clean and reliable transition between system and battery power without drawing too much battery power.

The PWRGD pin is provided on the device to protect the contents of RAM and the real time clock. It is also used to reduce power consumption whenever the system is powered down. This pin should be low whenever the system power supply is not within specifications for proper operation of the system. This pin may be driven by circuitry in either the power supply or on the system board. When the PWRGD input is low, it will disable all unnecessary inputs and outputs. In this way it will prevent noise on the inactive pins and reduce leakage current when the system is powered down. This pin must therefore be at high level for the remainder of the device to operate properly when system power is applied.

The PSRSTB* pin is provided to initialize the device whenever power is applied to the UM82C206. This pin will not alter the RAM or real time clock contents but it will initialize the necessary control register bits. A low on PSRSTB* pin 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 shown in Figure 10.

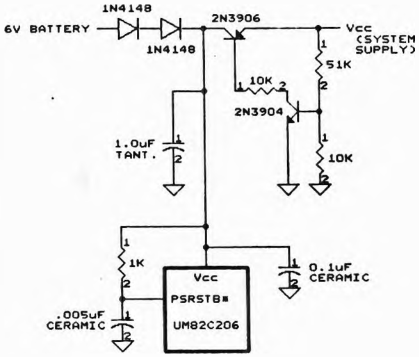


Figure 1Ø. Power Conversion and Reset Circuitry

Register Access

Reading and writing to the 128 locations in this subsystem 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 this subsystem, which may be read or written by asserting XIOR* or XIOW* with an address of 071H on the XA9-XA0 input pins.

In PC/AT compatible design, the AS is generated by an I/O write operation to port 070H. To avoid the unintentional change of the contents of real time clock and CMOS RAM, it is recommended that an address of 070H be applied to the XA9-XA0 inputs of UM82C206 during the AS asserted time.

Address Map

Table 8 illustrates the internal register/RAM organization of the real time clock subsystem of the UM82C206. The 128 addressable locations in this subsystem are divided into 10 bytes which normally contain the time, calendar, alarm setting and four control and status bytes and 114 general purpose CMOS RAM bytes. All 128 bytes are readable by the CPU. The CPU may also write to all locations except registers 0CH, 0DH, bit 7 of register 0AH and bit 7 of the register 00H which is always 0.

Table 8. Address Map for Real Time Clock Subsystem

Index	Function
00H	SECONDS
01H	SECONDS ALARM
02H	MINUTES
03H	MINUTES ALARM
04H	HOURS
05H	HOURS ALARM
06H	DAY OF WEEK
07H	DAY OF MONTH
08H	MONTH
09H	YEAR
0AH	REGISTER A
0BH	REGISTER B
0CH	REGISTER C
0DH	REGISTER D
0EH	USER RAM
0FH	USER RAM
· · ·	
7EH	USER RAM
7FH	USER RAM

Time/Calender and Alarm Bytes

The CPU can obtain the time and calendar information by reading the appropriate locations in the real time clock. Initialization of the time, calendar and alarm information is accomplished by writing to these bytes. Data stored in these locations are in binary-coded decimal (BCD) format.

Before initialization of the internal registers of the real time clock, the SET bit in register 0BH should be set to a "1" to prevent real time clock from updating. The CPU then initializes the first 10 locations in BCD format. The SET bit should then be cleared to allow updating. After initialized and enabled, the real time clock will perform clock/calendar updates at a 1.024 KHz rate in PC/AT compatible design.

Table 9. Time, Calendar, Alarm Data Format

Index Register Address	Function	BCD Range
00H	Seconds	00-59
01H	Seconds of Alarm	00-59
02H	Minutes	00-59
03H	Minutes of Alarm	00-59
04H	Hours (12 hour mode) Hours (24 hour mode)	01-12 (AM) 81-92 (PM) 00-23
05H	Hours of Alarm (12 hour mode) Hours of Alarm (24 hour mode)	01-12 (AM) 81-92 (PM) 00-23
06H	Day of Week	01-07
07H	Day of Month	01-31
08H	Month	01-12
09H	Year	00-99

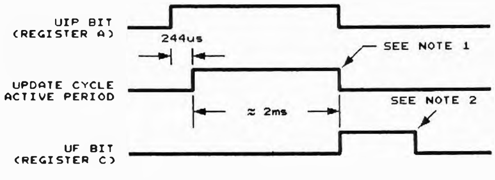
The alarm bytes can be programmed to generate an interrupt at a specific time or they can be programmed to generate a periodic interrupt. To generate an interrupt at a specific time, the user need only program the time that the interrupt is to occur into the 3 alarm bytes. Alternately, a periodic interrupt 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 once a second by programming the same value into all three alarm registers.

Table 9 shows the format for the ten clock, calendar and alarm registers. The 24/12 bit in Register 0BH determines whether the hour locations will be updated using a 1-12 or 0-23 format. After initialization the 24/12 bit cannot be changed without reinitializing the hour locations. In 12 hour format the bit 7 of the hours byte in both the time and alarm bytes will indicate PM when it is a "1".

Update Cycle

During normal operation the real time clock will perform an update cycle, assuming one of the proper time bases is chosen, the divider bits DV2-DV0 isn't reset and the SET bit in register 0BH is cleared. The function of the update cycle is to increment the clock/calendar registers and compare them to the alarm registers. If a match or don't care condition occurs between the two sets of registers, an alarm is issued and an interrupt control bits are enabled.

During an update cycle, the lower 10 registers are not accessible by the CPU. By this way it can prevent the possible corruption of data in the real time clock registers or the reading of incorrect data. To avoid contention between the



NOTE:
1. REGISTER 0-9 ARE UNAVAILABLE FOR READ OR WRITE THIS TIME.
2. UF BIT CLEARED BY READING REGISTER C.

Figure 11. Update Cycle

real time clock and the CPU, a flag is provided in register 0AH to alert the user of an update in progress cycle. This update in progress bit (UIP) is asserted 244us 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 0CH will be set. Figure 11 illustrates the update cycle. CPU access is always allowed to registers 0AH through 0DH during update cycles.

Two recommended methods can be used for reading and writing to the real time clock in a PC/AT compatible design. Both of them will allow the user to avoid contention between the CPU and the real time clock for access to the time and date data.

The first method is to read register 0AH, 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 244us to complete from the beginning of the read of register 0AH 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 0CH 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 cycle has been completed. The user then has to complete a read or write operation before the next update cycle.

Control and Status Registers

The UM82C206 contains four registers used to control the operation and monitor the status of the real time clock. The CPU can access these registers at any time with index address at 0AH-0DH.

REGISTER 0AH

Index register port : 70H
Data register port : 71H
Index : 0AH (Read/Write register except UIP)

REGISTER 0AH

Bits	Function
7	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 occurring. The UIP bit is read only and is not affected by reset. Writing a "1" to the SET bit in register 0BH will clear the UIP status bit.
6-4	DV2-DV0 - These three bits are used to control the Divider/Prescaler on the real time clock. While the UM82C206 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.

Bits	Function

6	5	4	Divider Options	
DV2	DV1	DV0	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	

3-0	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 separated 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.

3	2	1	0	Periodic Interrupt Rate	
RS3	RS2	RS1	RS0	4.194304MHz time base 1.048576MHz time base	32.768KHz time base
0	0	0	0	None	None
0	0	0	1	30.517 us	3.90625 ms
0	0	1	0	61.035 us	7.8125 ms
0	0	1	1	122.070 us	122.070 us
0	1	0	0	244.141 us	244.141 us
0	1	0	1	488.281 us	488.281 us
0	1	1	0	976.562 us	976.562 us
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		31.25 ms	31.25 ms
1	1		1	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

REGISTER 0BH

Index register port : 70H
Data register port : 71H
Index : 0BH (Read/Write register)

Bits	Function
7	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.
6	PIE - The Periodic Interrupt Enable Bit controls the generation of interrupts based on the value programmed into the RS3-RS0 bits of Register 0AH. 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 "0" by Reset.
5	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.
4	UIE - The update ended interrupt enable bit is used to enable the update end flag (UF) bit in register 0CH to generate an interrupt. A "1" in this bit enables the interrupt generating. A "0" disables it. This bit is cleared by reset. It is also cleared when the SET bit goes high.
3	SQWE - The square wave enable bit is always fixed to 0. It will disable the square wave generation.
2	DM - The data mode bit is always fixed to 0. It will always select the BCD format for real time clock.
1	24/12 - The 24/12 control bit is used to establish the format of both the Hour and Hour Alarm bytes. If this bit is a "1", the Real Time Clock will interpret and update 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.
0	DSE - The Real Time Clock can be instructed to handle daylight savings time changes by setting this bit to a "1". This enables two execeptions to the normal time keeping sequence to occur. On the first Sunday in April the time increments from 1:59:59 AM to 3:00:00 AM. On the last sunday in October when the time first reaches 1:59:59 AM it changes to 1:00:00 AM. Setting this bit to a "0" disables the execution of these two exceptions. RESET has no effet on this bit.

REGISTER 0CH

Index register port : 70H
Data register port : 71H

Bits	Function
7	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 effect on the contents.
6	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 and set IRQF if PIE is a "1".
5	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.
4	UF - A "1" appears in the UF bit whenever an update cycle is ended. This flag is also independent of it's enable (UIE) and will generate an interrupt if UIE is true.
3-0	Not used - All unused bits will be "0" when read and are not writeable.

REGISTER 0DH

Index register port : 70H
Data register port : 71H
Index : 0DH (Read Only register)

Bits	Function
7	VRT - The valid CMOS RAM and time bit indicates the condition of the contents of the RAM and real time clock. This bit is cleared to a "0" whenever the PSRSTB* input pin is low. This 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 initialized since power was applied to the device. Reset has no effect on this bit and it can only be set by reading register 0DH.
6-0	Not used - All unused bits will be "0" when read and are not writeable.

CMOS Static RAM

The 114 bytes of RAM from index address 0EH to 7FH are not affected by the real time clock. They are accessible during the update cycle and may be used for whatever the designer wishes. Typical applications will use these as nonvolatile storage for system configuration parameters. They are normally battery powered when the system is turned off.

ABSOLUTE MAXIMUM RATINGS*

Parameter	Symbol	Min.	Max.	Unit
Supply Voltage	VCC	-0.3	+6.7	V
Input Voltage	Vi	-0.3	+6.7	V
Output Voltage	Vo	-0.3	Vcc+0.3	V
Operating Temperature	Top	-20	+70	˚C
Storage Temperature	Tstg	-55	+125	˚C

*Comments

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are a stress ratings only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

DC ELECTRICAL CHARACTERISTICS
(Vcc = 4.75 to 5.25V, Ta = -20 to +70˚C)

Symbol	Parameter	Min.	Max.	Unit	Test Condition	Note
ViL	Input Low Voltage	-0.3	+0.8	V		
ViH	Input High Voltage	+2.2	Vcc+0.3	V		
VoL	Output Low Voltage (Iol=8mA)	-	0.5	V		
VoH	Output High Voltage (Ioh=-2mA)	3.5	-	V		
IiL	Input Leakage Current	-	1	uA		
Iozi	Output High-Z Leakage Current	-	1	uA		
Iccsb	Standby Power Supply Current	-	10	uA		

AC ELECTRICAL CHARACTERISTICS (8 MHz)
(Vcc = 4.75 to 5.25V, Ta = -20 to +70°C)

Symbol	Description	Min.	Typ.	Max.	Unit	Test Condition	Note
t1	Address setup to command active	25			ns		
t2	Command active period			200	ns		
t3	Address hold time from command inactive	0			ns		
t4	Data valid delay	160			ns		
t5	Data hold time from XIOR* inactive	10			ns		
t6	XD0-XD7 active from XIOR*	5		40	ns		
t7	Data setup to XIOW* inactive	160			ns		
t8	Data hold time from XIOW* inactive	0			ns		
t9	Command recovery time	120			ns		
t10	Interrupt request low width	100			ns		
t11	Interrupt request high width	200			ns		
t12	INTR output delay			300	ns		
t20	Real time clock cycle time			500	ns		
t21	AS pulse width	160			ns		
t22	Data valid setup to AS inactive	160			ns		
t23	Data hold time from AS inactive	0			ns		
t24	OSCI period	500			ns		
t25	OSCI high time	200			ns		
t26	OSCI low time	200			ns		
t27	PSRSTB high delay from VCC	5			us		
t28	PSRSTB low pulse width	5			us		
t29	VRT bit valid delay			2	ns		
t40	TMRCLK period	125			ns		

Symbol	Description	Min.	Typ.	Max.	Unit	Test Condition	Note
t41	TMRCLK low time	50			ns		
t42	TMRCLK high time	50			ns		
t43	GATE2 setup to TMRCLK	50			ns		
t44	GATE2 hold time from TMRCLK	50			ns		
t45	GATE2 low time	50			ns		
t46	GATE2 high time	50			ns		
t47	OUT2 delay from TMRCLK			120	ns		
t48	OUT2 delay from GATE2			120	ns		
t50	SYSCLK period (DMA clock = SYSCLK)	125			ns		
t50a	SYSCLK period (DMA clock = SYSCLK/2)	62			ns		
t51	SYSCLK low time (DMA clock = SYSCLK)	43			ns		
t51a	SYSCLK low time (DMA clock = SYSCLK/2)	22			ns		
t52	SYSCLK high time (DMA clock = SYSCLK)	55			ns		
t52a	SYSCLK high time (DMA clock = SYSCLK/2)	27			ns		
t53	DREQi setup to SYSCLK	0			ns		
t54	HRQ valid from SYSCLK			75	ns		
t55	HLDA1 setup to SYSCLK	45			ns		
t56	AENi valid delay from SYSCLK			105	ns		

Symbol	Description	Min.	Typ.	Max.	Unit	Test Condition	Note
t57	AENi invalid delay from SYSCLK			80	ns		
t58	ADSTBi valid delay from SYSCLK			50	ns		
t59	ADSTBi invalid delay from SYSCLK			120	ns		
t60	XD0-XD7 active delay from SYSCLK			60	ns		
t61	XD0-XD7 valid setup to ADSTBi low	65			ns		
t62	XD0-XD7 hold time from ADSTBi low	25			ns		
t63	XD0-XD7 tristate delay from SYSCLK			135	ns		
t64	Address valid delay from SYSCLK			60	ns		
t65	Address hold time from DMAMEMR high	50			ns		
t66	Address tristate delay from SYSCLK			55	ns		
t67	DACKi delay from SYSCLK			105	ns		
t68	Command enable delay from SYSCLK			90	ns		
t69	Command active delay from SYSCLK			120	ns		
t70	Write command inactive delay from SYSCLK			80	ns		
t71	Address hold time from write command high	75			ns		
t72	Command tristate delay from SYSCLK			75	ns		
t73	Read command inactive delay from SYSCLK TC delay from SYSCLK			115	ns		
t74	TC delay from SYSCLK			60	ns		
t75	XD0-XD7 setup to read command inactive	90			ns		

Symbol	Description	Min.	Type	Max.	Unit	Test Condition	Note
t76	XD0-XD7 hold time from read command inactive	0			ns		
t77	XD0-XD7 valid delay from SYSCLK			120	ns		
t78	XD0-XD7 hold from write command inactive	15			ns		
t79	IOCHRDY input setup to SYSCLK	35			ns		
t80	IOCHRDY input hold time from SYSCLK	20			ns		

TIMING WAVEFORMS

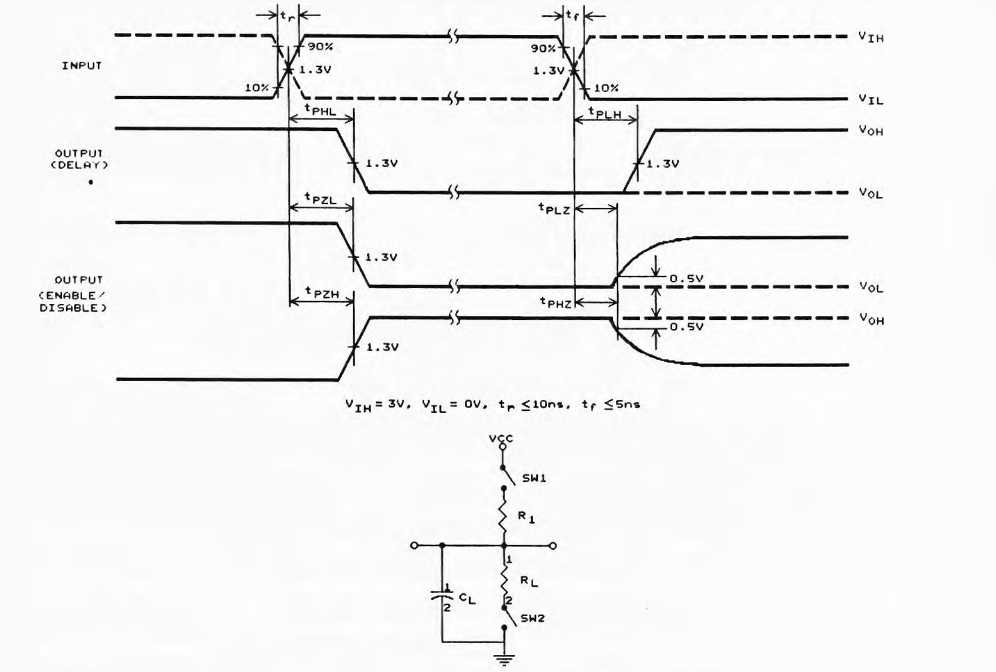
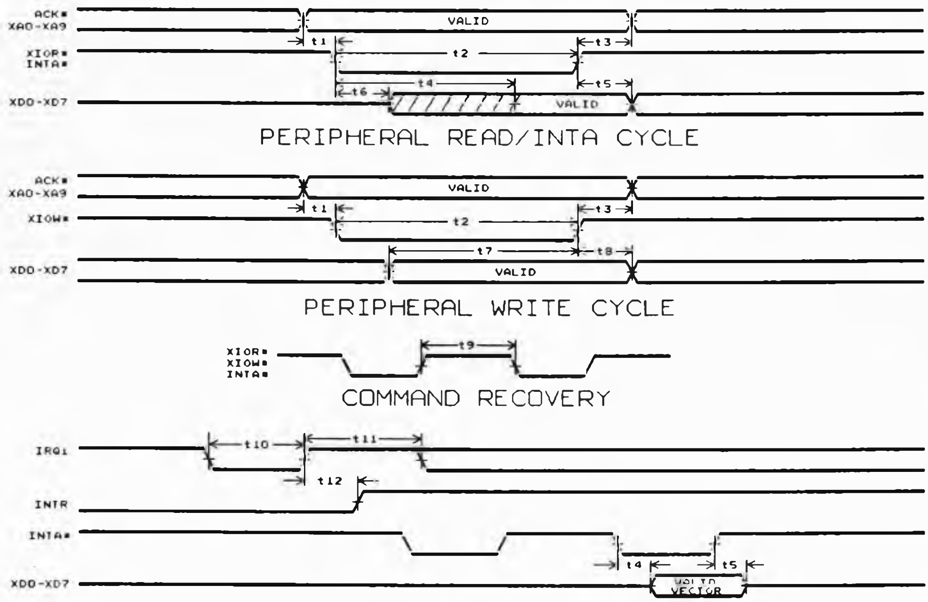


Figure 12. AC Characteristics Measurement Waveform and Load Circuit

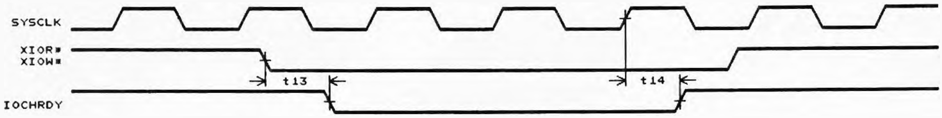


PERIPHERAL READ/INTA CYCLE

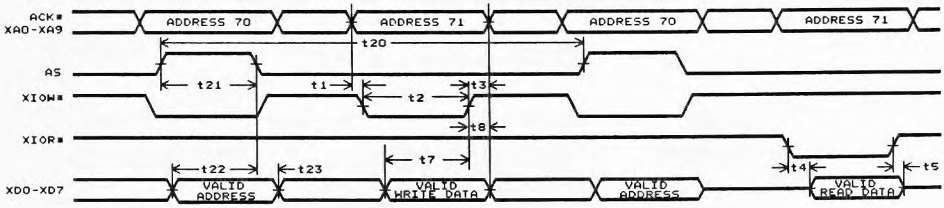
PERIPHERAL WRITE CYCLE

COMMAND RECOVERY

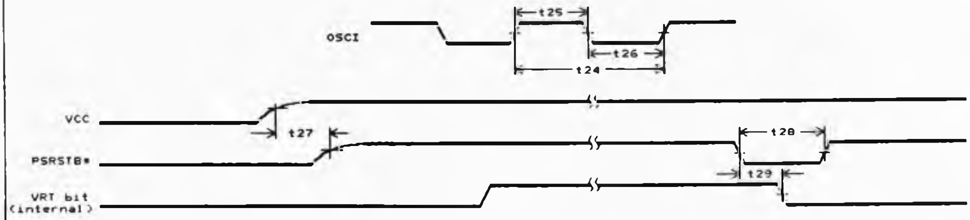
INTA SEQUENCE



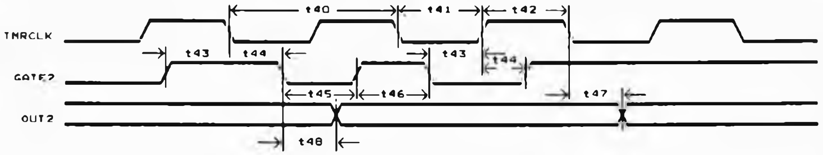
IOCHRDY OUTPUT



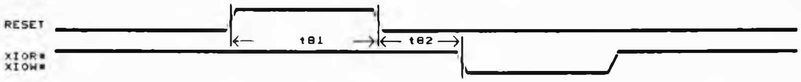
REAL TIME CLOCK ACCESS CYCLE



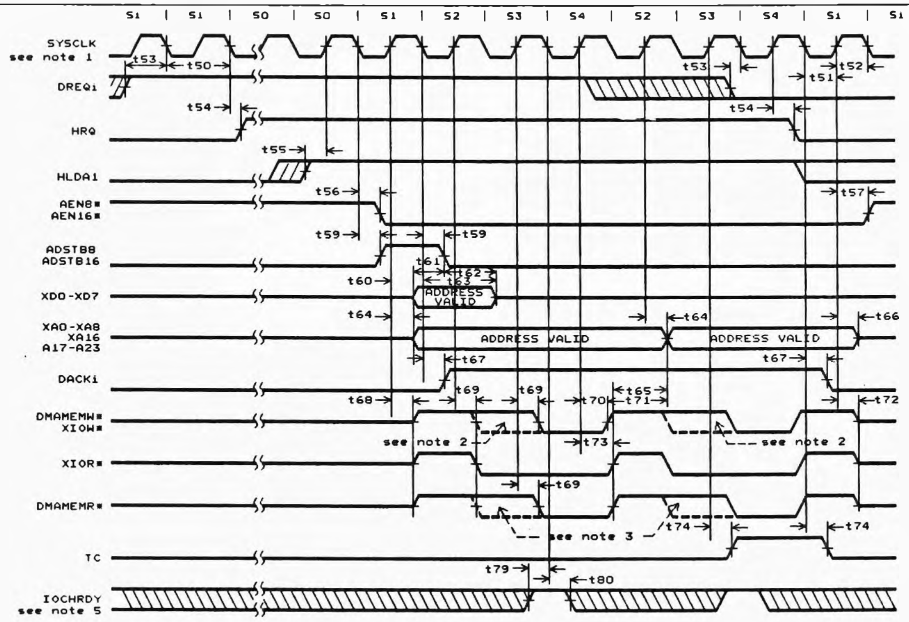
REAL TIME CLOCK POWER-UP SEQUENCE



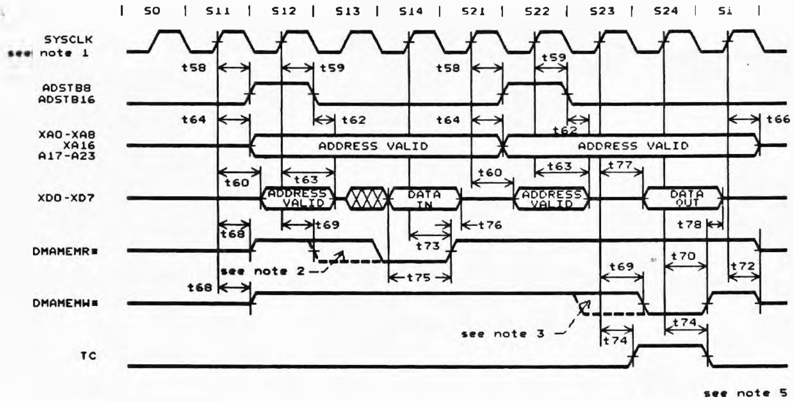
COUNTER/TIMER PARAMETERS



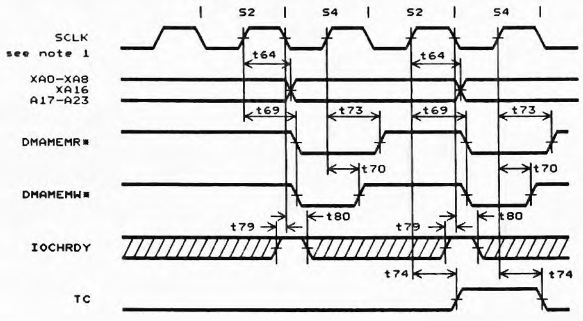
DMA RESET



DMA Transfer Timing



Memory - to - Memory Transfer



Compressed Transfer

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. IOCHRDY Input Timing.

5. DMA wait states are added between S3,S4 in normal timing; between S2,S4 in compressed timing; between S13, S14 and S23, S24 in memory-to-memory timing.