3. IOP Memory and Interrupt Controllers TABLE OF CONTENTS

3 IOP Memory and Interrupt Controllers

This section continues section 2 with a discussion of how IOP local memory is addressed and how the IOP handles interrupts.

3.1 IOP Memory

IOP memory consists of 16 Kbytes of static RAM and 16 Kbytes of EPROM located on the IOP board. This unit of memory is called local memory. An additional 1 to 3.5 Mbytes of memory are implemented with dynamic RAMs on the Dove system memory board. Schematics of IOP memory logic are contained in Appendix D.

3.1.1 Memory Addressing

Address Bus

The IOP address bus is 24-bits wide; it is addressed by A.23 - A.00. When the IOP 80186 accesses memory, the upper nibbles, A.23 - A.20, are forced high. The other 20 address bits allow the IOP 80186 to address up to 1 Mbyte of memory.

Address Space

The IOP local memory requires zero wait states and stores the 80186 boot programs, device handlers, interrupt vector tables, and local buffers. In contrast, accessing system memory requires one or more wait states. Figure 3.1 illustrates the address ranges of local memory and system memory.

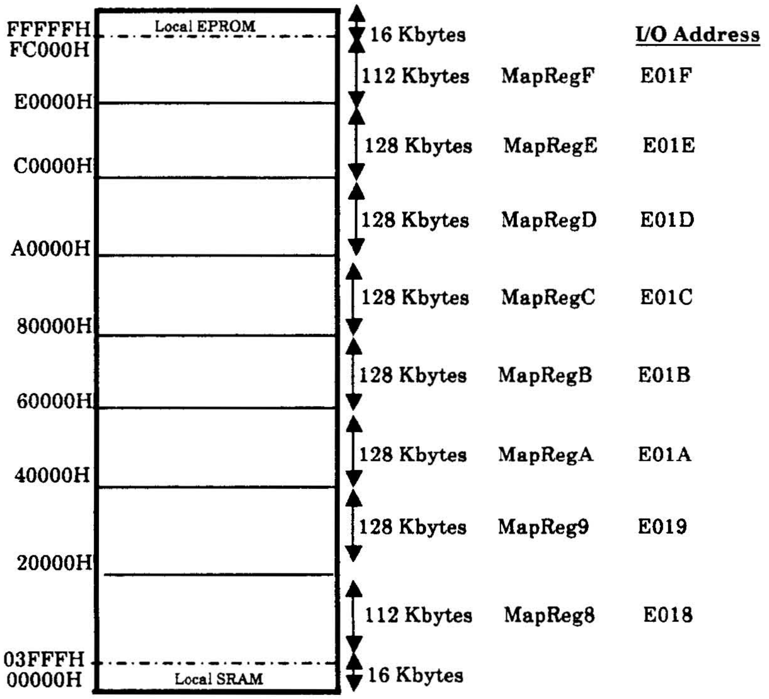


Figure 3.1. IOP address space

3.1.2 Memory Mapping

A memory mapping scheme resides on the system memory board. It contains 1 Mbyte of address space and is divided into 8 "pages." Each page contains 128 Kbytes of memory and has a map register assigned to it, as shown in Figure 3.1. All 80186 addresses are mapped except for the SRAM and EPROM addresses.

Figure 3.2 illustrates the mapping scheme.

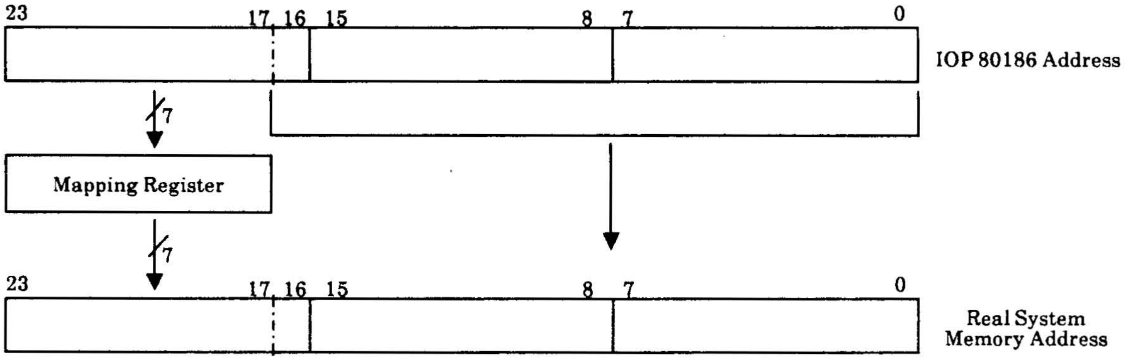


Figure 3.2. IOP memory address space bit assignment

The map registers are addressable by the I/O address E018H to E01FH. The map registers' content and uses are assigned by the IOP operating system (Opie).

3.2 Interrupts

The IOP interrupt controller is programmed by the 80186 to control interrupts to the 80186 from the other IOP subsystems. The interrupt controller accepts interrupt requests from attached I/O devices, determines which requesting device has the highest priority, then activates an interrupt to the 80186 if the selected device has a higher priority than the device currently being serviced.

3.2.1 Hardware

The interrupt controller logic consists of three Intel 8259A chips, an 80186 and part of an 8274. The 82589 controller can handle up to eight vectored priority interrupts for the IOP 80186.

Table 3.1 lists the master 8259's signal names and functions; Table 3.2 describes the possible interrupts. For schematic diagrams, see Appendix D.

For an in depth discussion of the 8259A component, see Volume 1 of the Intel Microsystem Components Handbook.

Table 3.1. Master Interrupt Controller Pin Description

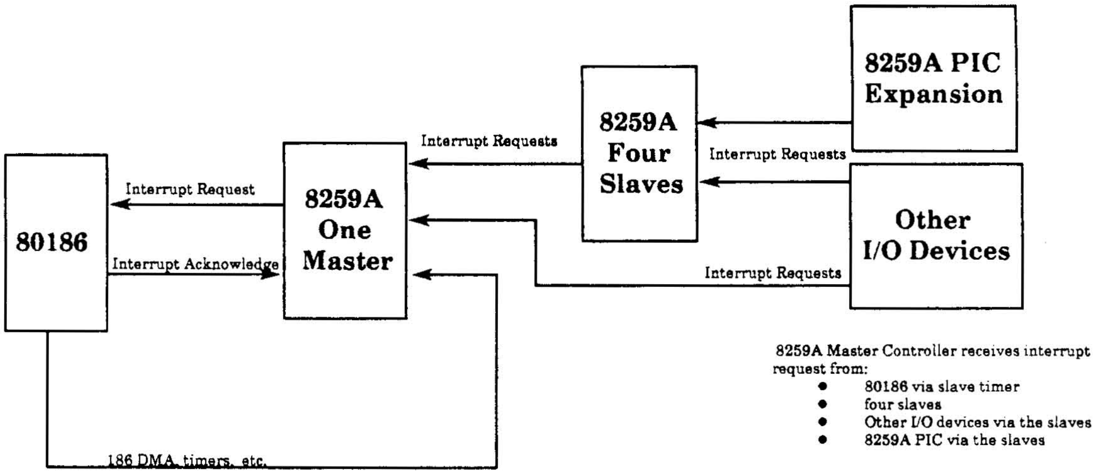
Symbol	Pin No.	Type	Name and Function
VCC	28	I	Supply: +5V Supply
GND	14	I	Ground
CS'	1	I	**Chip Select:**. A low on this pin enables RD' and WR' communication between the CPU and the 8259A. INTA functions are independent of CS.
WR'	2	I	**Write.** A low on this pin when CS is low enables the 8259A to accept command words from the CPU.
RD'	3	I	**Read.** A low on this pin when CS is low enables the 8259A to release status onto the data bus for the CPU.
D7-D0	4-11	I/O	**Bidirectional Data Bus.** Control, status and interrupt-vector information is transferred via this bus.
CAS0-CAS2	12,13,15	I/O	**Cascade Lines.** The CAS lines form a private 8259A bus to control a multiple 8259A structure. These pins are outputs for the master interrupt controller and interrupts for the slave interrupt controller.
SP'/EN'	16	I/O	**Slave Program/Enable Buffer.** This is a dual function pin. When in the buffered mode, it is used as an output to control buffer transceivers (EN). When not in the buffered mode, it can be used as an input to designate a master (SP = 1) or slave (SP = 0).
INT	17	O	**Interrupt.** This pin goes high whenever a valid interrupt request is asserted. It is used to interrupt the CPU; thus it is connected to the CPU's interrupt pin.
IR0-IR7	18-25	I	**Interrupt Requests.** Asynchronous inputs. An interrupt request is executed by raising an IR input (low to high) and holding it high until it is acknowledged (Edge Triggered Mode), or just by a high level on an IR input (Level Triggered Mode).
INTA'	26	I	**Interrupt Acknowledge.** This pin is used to enable 8259A interrupt-vector data onto the data bus by a sequence of interrupt acknowledge pulses issued by the CPU.
A0	27	I	**A0 Address Line.** This pin acts in conjunction with the CS', WR', and RD' pins. It is used by the 8259A to decipher various command words the CPU writes and status the CPU wishes to read. It is connected to the CPU A0 address line A1.

Table 3.2. Description of Possible Interrupts

Interrupt	From	To	Explanation
Parity Check	System Memory	Master IR0	Indicates the occurrence of a system memory parity error.
DebugInIntr	From Debugger	Master IR1	Debugger Interrupt
KbrdInputReq	From Keyboard	Master IR3	Keyboard Char R x D Interrupt
RS232CIntReq	From RS232	Master IR4	RS232 Slave Interrupt
IntFrom186	From 80186	Master IR6	80186 Slave Interrupt
VertRetrIntr	Display		
ENetIntrReq	Ethernet		
RDiskDmaIntr'	Rigid Disk-DMA		
RDiskCtlrIntr	Rigid Disk		
FDCIntrReq	Floppy Disk		
MesaIntrReq	Mesa Processor		
ExpIntrReq	Expansion		
A/MEBIntr'	Dbreak Memory Expansion		
DebugOutIntr	Debugger	Master IR2	Debugger Interrupt

3.2.2 Theory of Operations

The interrupt controller handles multi-level priority interrupts from the slave controller, other I/O devices via the slave, the expansion PIC via the slave, and the 80186. Figure 3.3 illustrates the interrupt flow for the interrupt controller.



8259A Master Controller receives interrupt request from:

- 80186 via slave timer
- four slaves
- Other I/O devices via the slaves
- 8259A PIC via the slaves

Figure 3.3. IOP interrupts

3.2.2.1 Master Interrupt Controller

Interrupt requests for the IOP are funneled down through slave controllers and peripherals to a single master interrupt controller. This controller resolves interrupt priorities and interrupts the IOP 80186.

The interrupt process is as follows:

1) An interrupt request input line has a rising edge transition. That is, the signal goes from logic 0 to logic 1.

   The interrupt controller is edge-sensitive and detects and latches this transition. When this signal occurs, it also causes a bit to be set in the interrupt controller's interrupt request register (IRR) that corresponds to the interrupt level on which the transition occurred.

2) If the interrupt input is not masked (see masking), then the interrupt controller sends an interrupt request to the 80186.

3) When the 80186 accepts the interrupt, it completes the current intruction being executed and then begins the interrupt acknowledge sequence.

   The 80186 responds to an interrupt by executing two interrupt acknowledge cycles separated by two idle states.

4) When the master interrupt controller sees the first interrupt acknowledge cycle, it determines who has interrupt priority by

setting a bit in the in service register (ISR) that corresponds to the highest priority IRR bit that is set.

When the ISR bit is set, all interrupts of lower priority are masked from generating interrupts to the 80186. The Master interrupt controller then sets the cascade lines (CAS0-2) to reflect the interrupt level that is being serviced.

5) The master interrupt controller then waits for the second interrupt acknowledge cycle. When the second acknowledge cycle occurs, the master interrupt controller checks to see if the interrupt being serviced is a general interrupt or a slave controller interrupt.

6) If the interrupt is a general interrupt, then the interrupt controller places an interrupt vector onto the data bus during the second acknowledge cycle. This vector is a unique one byte vector that corresponds to the interrupt being serviced and points to a service routine for a particular interrupt. Each IOP device has a special service routine associated with its functional requirements. (See interrupt pointer table at the end of this section.)

   If the interrupt comes from a slave interrupt controller, then the master interrupt controller does nothing during the second acknowledge cycle, thus allowing the slave controller to generate the interrupt vector.

3.2.2.2 Slave Interrupt Controllers

The slave interrupt controllers accept and process the interrupt requests the same as the master interrupt controller accesses and processes them. However, an interrupt generated by a slave controller goes to the master controller instead of going directly to the 80186.

The interrupt process for a slave interrupt controller is as follows:

1) The slave controller accepts an interrupt request and sends the request to the master interrupt controller.

2) The master controller then processes this request as if it were a normal interrupt request. In this case, the slave, instead of the master, generates the interrupt vector for the 80186.

3) The 80186 acknowledges the interrupt and generates two acknowledge cycles. After the first interrupt acknowledge pulse occurs, the master interrupt controller sets the cascade lines (CAS0-2). This signal ensures that the slave corresponding to the interrupt request level being serviced is selected.

4) When the cascade lines indicate that a slave's interrupt level is being serviced, the slave waits for the second interrupt acknowledge to occur.

   When the interrupt acknowledge occurs, the slave controller places the appropriate vector on the data bus for the 80186 to read.

One other difference should be noted. When the master interrupt controller is programmed in Special Fully Nested Mode (SFNM), the behavior of the ISR bits is modified for slave interrupt levels only.

When an ISR bit is set for an interrupt that has a slave on it, the ISR bit will not mask further interrupt requests for that level. That is, an interrupt request for a slave that already has its ISR bit set is processed and passed on to the 80186 as if no ISR bit were set. Interrupt requests from lower priority levels are still masked.

Interrupt Pointer Table

During the second interrupt acknowledge cycle, the 80186 reads the interrupt vector. The low 1K (0-3FF) of the IOP local RAM is used by the 80186 as a table of instruction pointers and code sements for the various interrupts.

Figure 3.4 illustrates the interrupt pointer table. Each entry in this table is two words long. The first word is the instruction pointer to the service routine. The second word contains the code segment where the service routine resides.

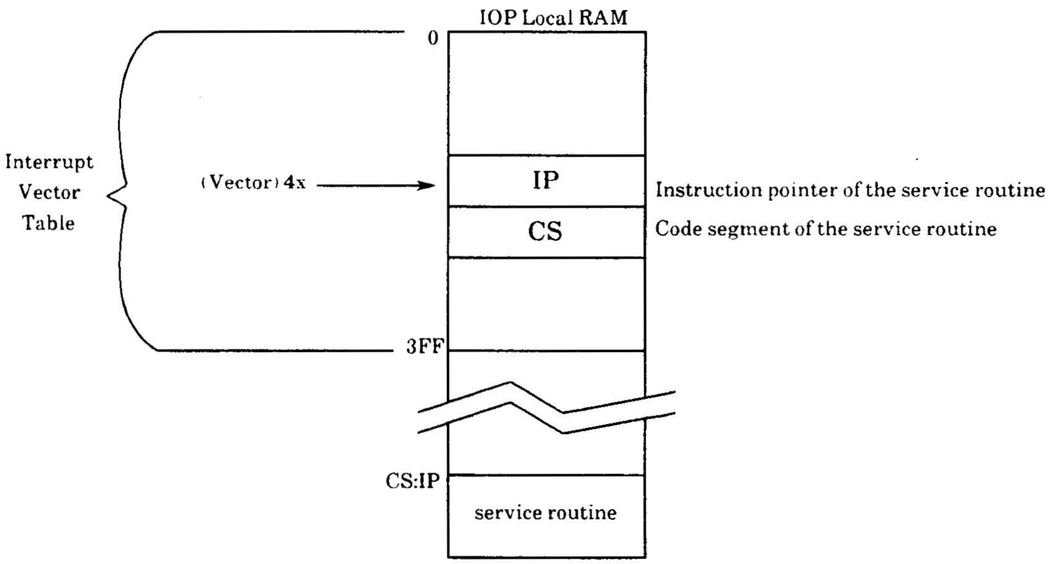

IOP Local RAM
0
Interrupt Vector Table
(Vector) 4x
IP
CS
Instruction pointer of the service routine
Code segment of the service routine
3FF
CS:IP
service routine

Figure 3.4. Interrupt pointer table

The 80186 saves the flag register, code segment register, and the instruction pointer register on the stack.

The 80186 multiplies the vector by four to obtain an index into the vector table. It then reads the instruction pointer into its IP register and the code segment into its CS register.

The 80186 then continues executing the service routine pointed to by the code segment and instruction pointer.

Masking Interrupts

Interrupts may be masked in two ways. One: When a higher priority interrupt is accepted, the interrupt sets the ISR bit, which then masks all interrupts of lower priority from being generated to the 80186. Two: Software setting of the mask register in the interrupt controller prevents or masks an interrupt from ever setting its corresponding IRR bit.

Cascade Lines

Cascade lines behave much the same as address lines; the slave controllers constantly monitor the lines, waiting to see their "address" or interrupt level appear on the lines.

3.2.3 Programmer Interface

This section describes the registers and the relevant timing for the interrupt controller.

3.2.3.1 Registers

The six registers used for interrupts to the IOP are illustrated as follows:

- Figure 3.5 illustrates the content of the first interrupt vector byte.
- Figure 3.6 illustrates the content of the second interrupt vector byte.
- Figure 3.7 illustrates the content of the third interrupt vector byte.
- Figure 3.8 illustrates the content of the interrupt vector byte for the iAPX 86 system mode.
- Figure 3.9 illustrates the initialization sequence.
- Figure 3.10 illustrates the initialization command word format.
- Figure 3.11 illustrates the operation command word format.

	D7	D6	D5	D4	D3	D2	D1	D0
Call Code	1	1	0	0	1	1	0	1

Figure 3.5. Content of first interrupt vector byte

IR	Interval = 4							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	A5	1	1	1	0	0
6	A7	A6	A5	1	1	0	0	0
5	A7	A6	A5	1	0	1	0	0
4	A7	A6	A5	1	0	0	0	0
3	A7	A6	A5	0	1	1	0	0
2	A7	A6	A5	0	1	0	0	0
1	A7	A6	A5	0	0	1	0	0
0	A7	A6	A5	0	0	0	0	0

IR	Interval = 8							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	1	1	1	0	0	0
6	A7	A6	1	1	0	0	0	0
5	A7	A6	1	0	1	0	0	0
4	A7	A6	1	0	0	0	0	0
3	A7	A6	0	1	1	0	0	0
2	A7	A6	0	1	0	0	0	0
1	A7	A6	0	0	1	0	0	0
0	A7	A6	0	0	0	0	0	0

Figure 3.6. Content of second interrupt vector byte

D7	D6	D5	D4	D3	D2	D1	D0
A15	A14	A13	A12	A11	A10	A9	A8

Figure 3.7. Content of third interrupt vector byte

IR	D7	D6	D5	D4	D3	D2	D1	D0
7	T7	T6	T5	T4	T3	1	1	1
6	T7	T6	T5	T4	T3	1	1	0
5	T7	T6	T5	T4	T3	1	0	1
4	T7	T6	T5	T4	T3	1	0	0
3	T7	T6	T5	T4	T3	0	1	1
2	T7	T6	T5	T4	T3	0	1	0
1	T7	T6	T5	T4	T3	0	0	1
0	T7	T6	T5	T4	T3	0	0	0

Figure 3.8. Content of interrupt vector byte for iAPX 86 System Mode

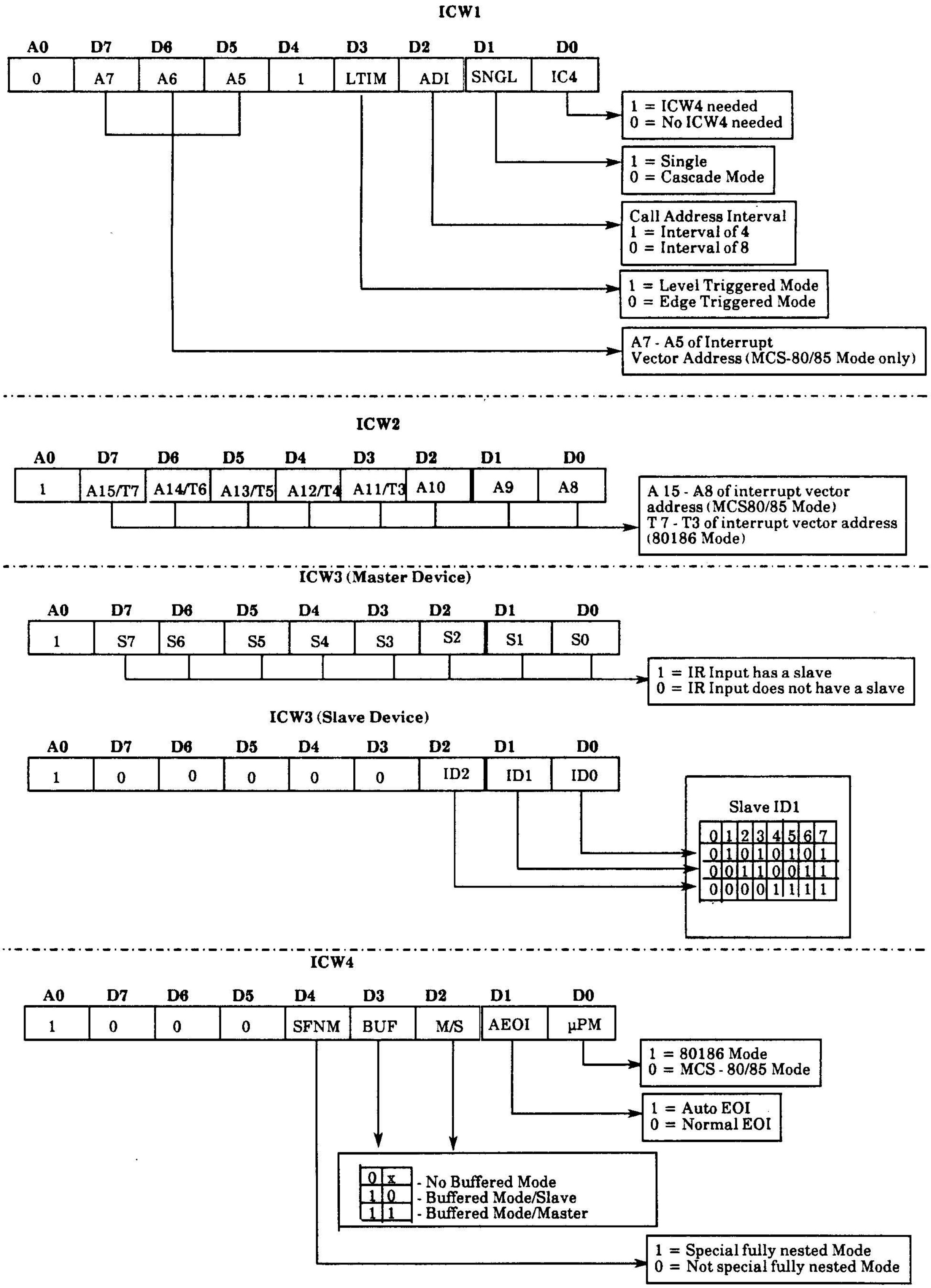
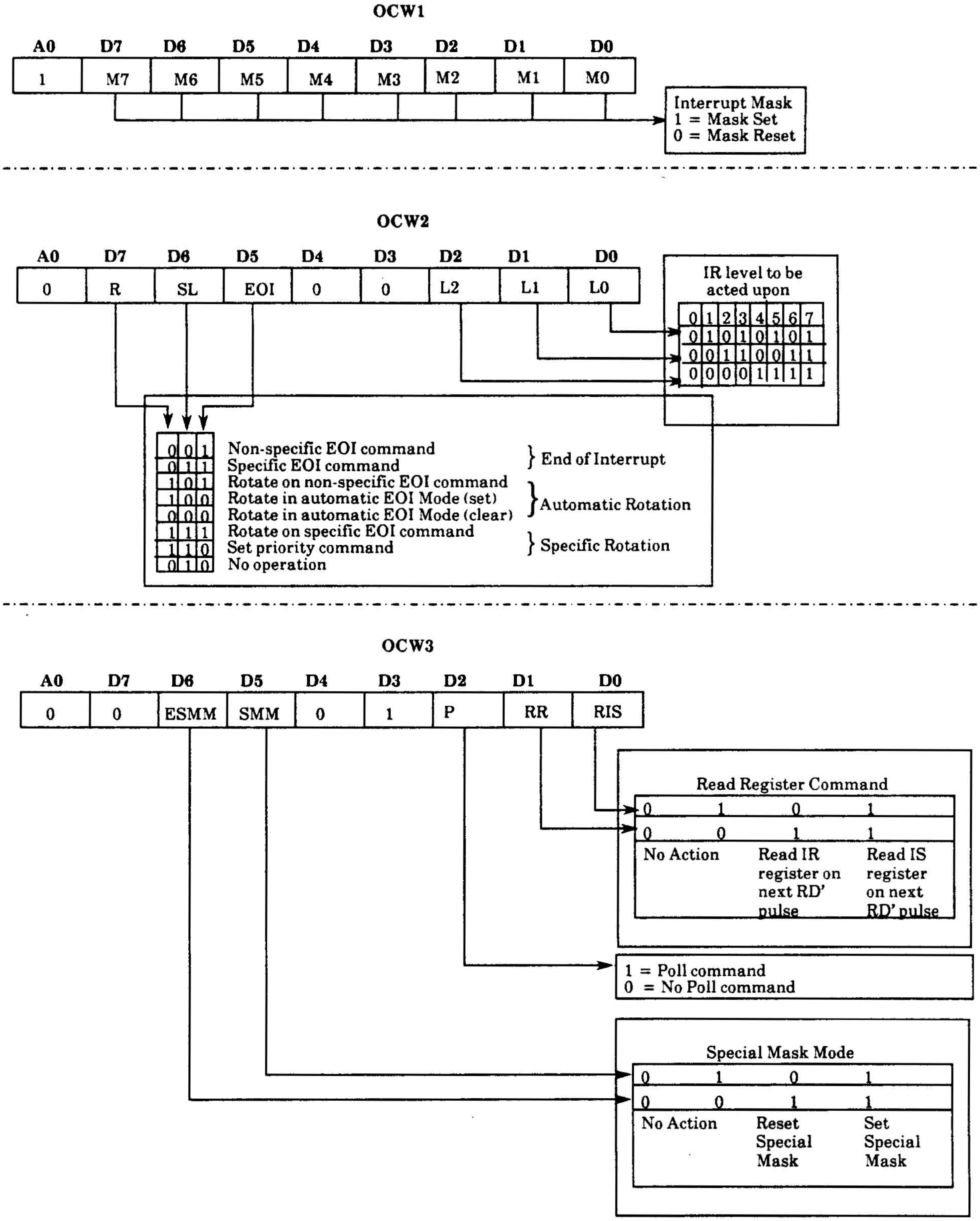

ICW1
A0 D7 D6 D5 D4 D3 D2 D1 D0
0 A7 A6 A5 1 LTIM ADI SNGL IC4
1 = ICW4 needed
0 = No ICW4 needed
1 = Single
0 = Cascade Mode
Call Address Interval
1 = Interval of 4
0 = Interval of 8
1 = Level Triggered Mode
0 = Edge Triggered Mode
A7 - A5 of Interrupt
Vector Address (MCS-80/85 Mode only)
ICW2
A0 D7 D6 D5 D4 D3 D2 D1 D0
1 A15/T7 A14/T6 A13/T5 A12/T4 A11/T3 A10 A9 A8
A 15 - A8 of interrupt vector
address (MCS80/85 Mode)
T 7 - T3 of interrupt vector address
(80186 Mode)
ICW3 (Master Device)
A0 D7 D6 D5 D4 D3 D2 D1 D0
1 S7 S6 S5 S4 S3 S2 S1 S0
1 = IR Input has a slave
0 = IR Input does not have a slave
ICW3 (Slave Device)
A0 D7 D6 D5 D4 D3 D2 D1 D0
1 0 0 0 0 0 ID2 ID1 ID0
Slave ID1
0 1 2 3 4 5 6 7
0 1 0 1 0 1 0 1
0 0 1 1 0 0 1 1
0 0 0 0 1 1 1 1
ICW4
A0 D7 D6 D5 D4 D3 D2 D1 D0
1 0 0 0 SFNM BUF M/S AEOI μPM
1 = 80186 Mode
0 = MCS - 80/85 Mode
1 = Auto EOI
0 = Normal EOI
0 x - No Buffered Mode
1 0 - Buffered Mode/Slave
1 1 - Buffered Mode/Master
1 = Special fully nested Mode
0 = Not special fully nested Mode

Figure 3.9. Initialization command word format



OCW1

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	M7	M6	M5	M4	M3	M2	M1	M0

Interrupt Mask
1 = Mask Set
0 = Mask Reset

OCW2

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	R	SL	EOI	0	0	L2	L1	L0

IR level to be acted upon

0	1	2	3	4	5	6	7
0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	0	0	0	1	1	1	1

0	0	1	Non-specific EOI command	End of Interrupt
0	1	1	Specific EOI command	
1	0	1	Rotate on non-specific EOI command	Automatic Rotation
1	0	0	Rotate in automatic EOI Mode (set)	
0	0	0	Rotate in automatic EOI Mode (clear)	
1	1	1	Rotate on specific EOI command	Specific Rotation
1	1	0	Set priority command	
0	1	0	No operation	

OCW3

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	ESMM	SMM	0	1	P	RR	RIS

Read Register Command

0	1	0	1
0	0	1	1
No Action		Read IR register on next RD' pulse	Read IS register on next RD' pulse

1 = Poll command
0 = No Poll command

Special Mask Mode

0	1	0	1
0	0	1	1
No Action		Reset Special Mask	Set Special Mask

Figure 3.10. Operation command word format

3.2.3.2 Timing

As discussed in section 3.2.2.1, when the interrupt controller sends an interrupt to the IOP 80186, the 80186 responds with two acknowledge cycles. Figure 3.10 illustrates the timing for the interrupt acknowledge cycle.

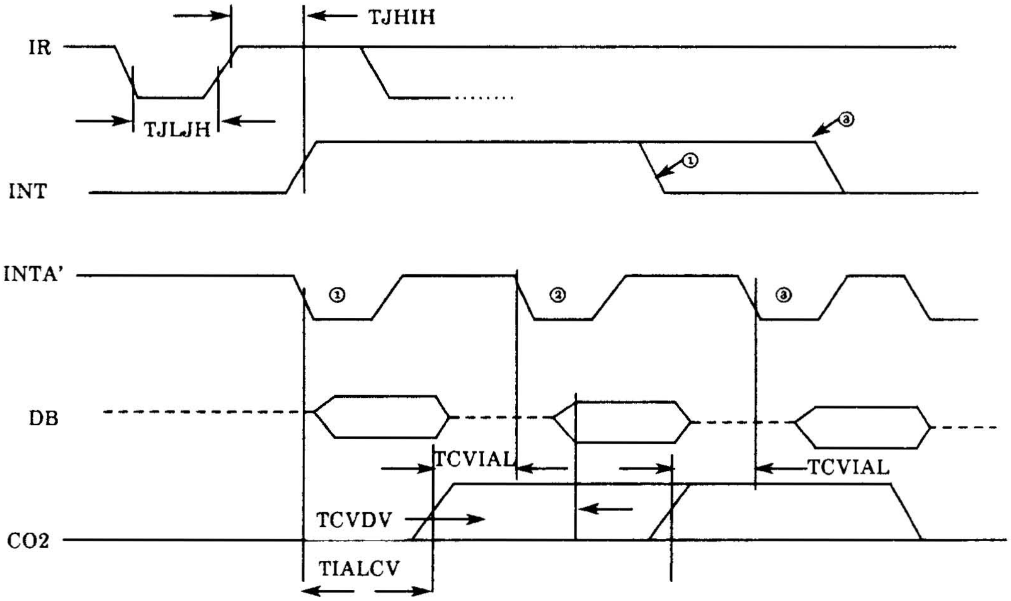


Figure 3.10. Interrupt controller timing diagram INTA' cycle