

Segment 8B

8259A

Programmable Interrupt Controller

- ✓ Necessity of 8259A
- ✓ Function of 8259A
- ✓ Connection of 8259A with 8086 microprocessor.
- ✓ Internal architecture of 8259A.
- ✓ Interrupt Operation of 8259A.
- ✓ Programming Modes, ICWs, OCWs
- ✓ Example of Interfacing 8259A with 8086 microprocessor.

Course Instructor

Mohammed Abdul kader
Assistant Professor, EEE, IIUC

Necessity of 8259A

In a system, microprocessor may need to perform the following tasks in an efficient way using interrupt:

- ✓ Read ASCII characters from a keyboard on an interrupt basis.
- ✓ Count interrupts from a timer to produce a real time clock of seconds, minutes and hours.
- ✓ Communicate with an A/D converter.
- ✓ Communicate with a display or printer.
- ✓ Detect several emergency signal like power failure etc on an interrupt basis.

Each of these interrupt applications requires a separate interrupt pin. But, the 8086 has only two interrupt inputs: **NMI and INTR**. If we use NMI for a power failure interrupt, this leaves only one interrupt input for all other applications.

The solution is to use an external device called a priority interrupt controller (PIC) such as Intel 8259A.

Function of 8259A

- ✓ The Programmable Interrupt Controller (PIC) functions as an overall manager in an interrupt-driven system environment.
- ✓ It accepts requests from the peripheral equipment, determines which of the incoming requests is of the highest importance (priority), ascertains whether the incoming request has a higher priority value than the level currently being serviced and issues an interrupt to the CPU based on this determination.
- ✓ Each peripheral device or structure usually has a special program or, routine that is associated with its specific functional or operational requirements; that is referred to as a service routine or service procedure.
- ✓ The 8259A PIC, after issuing an interrupt to the CPU, must somehow input information (interrupt vector number) into the CPU that can point the program counter to the service procedure associated with the requesting device .

Connection of 8259A with 8086 microprocessor (Single Mode)

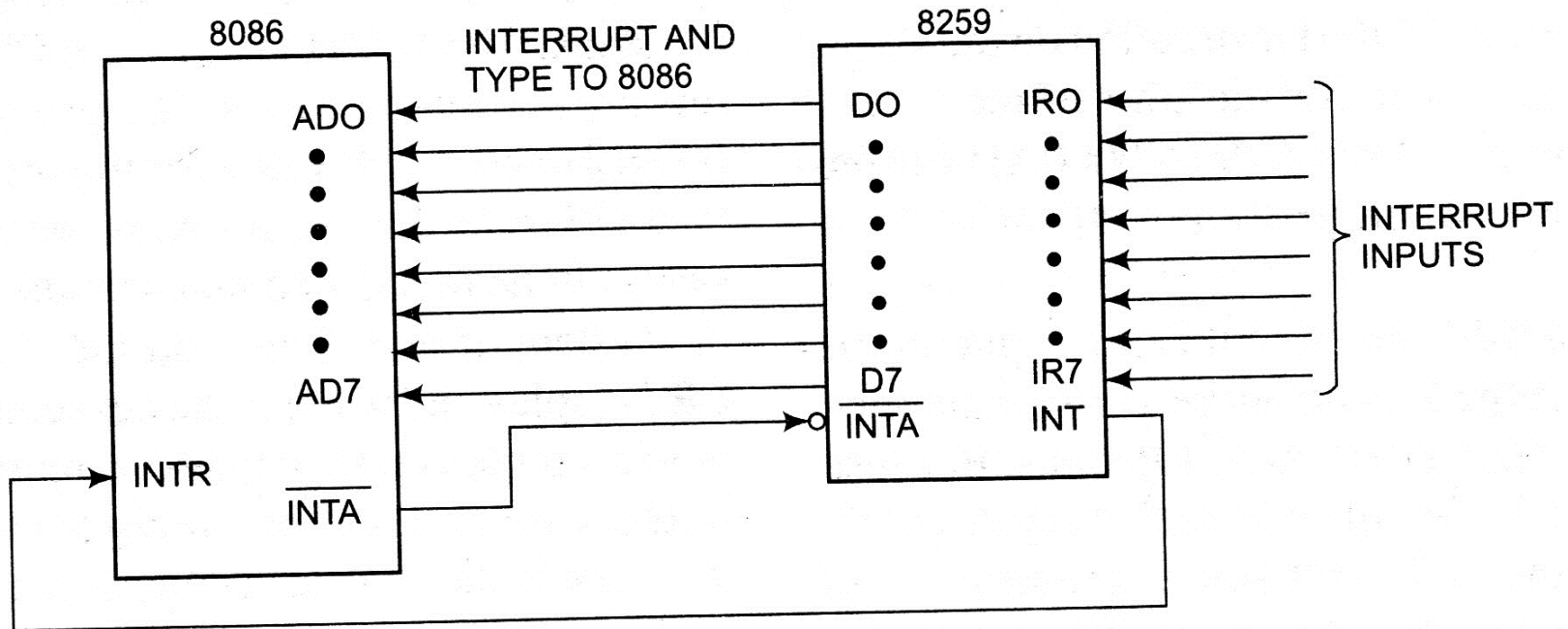


Fig. 1 Block Diagram showing an 8259 connected to an 8086

Connection of 8259A with 8086 microprocessor (Cascade Mode)

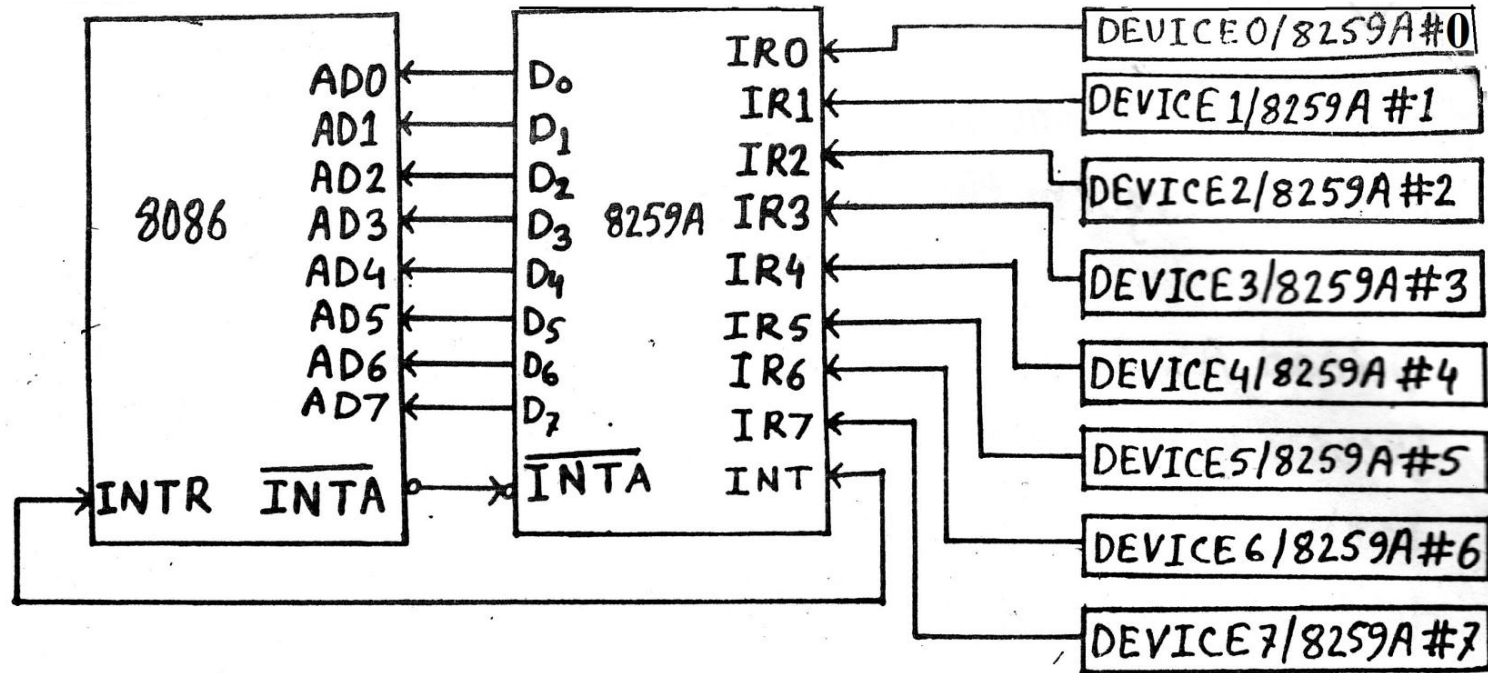


Fig. 2 Block Diagram showing an 8259 connected to an 8086

The 8259A PIC adds eight vectored priority encoded interrupts to the microprocessor. This controller can be expanded without additional hardware, to accept up to 64 interrupt requests. This requires a master 8259A and eight 8259A slaves.

Internal Block Diagram of 8259A

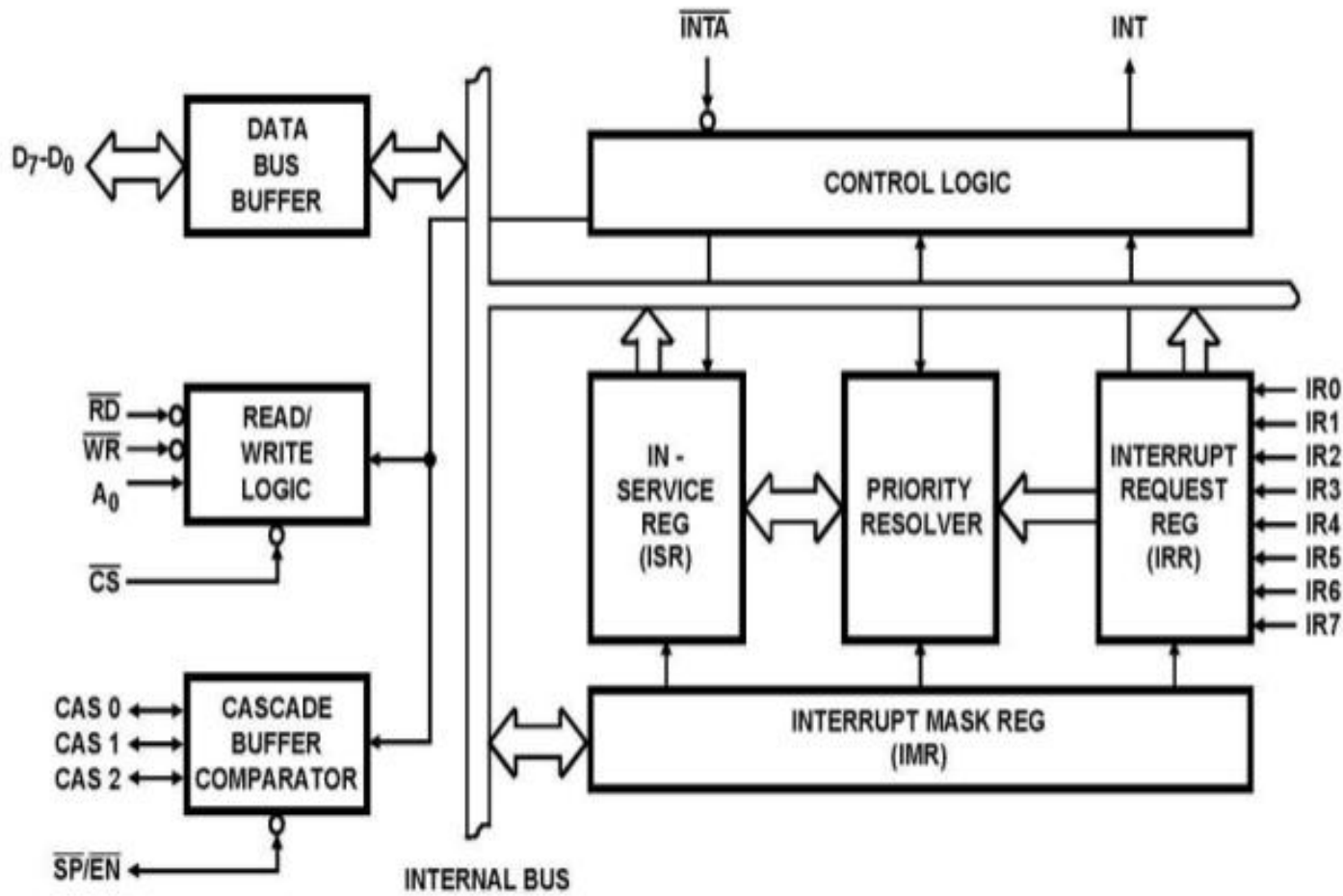


Fig. 3 Internal Block Diagram

\overline{CS}	1	28	Vcc
\overline{WR}	2	27	A0
\overline{RD}	3	26	INTA
D7	4	25	IR7
D6	5	24	IR6
D5	6	23	IR5
D4	7	22	IR4
D3	8	21	IR3
D2	9	20	IR2
D1	10	19	IR1
D0	11	18	IR0
CAS0	12	17	INT
CAS1	13	16	SP/EN
gnd	14	15	CAS2

Fig. 4 Pin Diagram

Internal Block Diagram of 8259A (Cont.)

8-bit data bus:

The 8-bit data bus ($D_7 - D_0$) allows the 8086-

- * to send control words to the 8259A and read a status word from the 8259A.
- * to send interrupt types to the 8086.

The eight data lines are always connected to the lower half of the 8086 data bus because the 8086 expects to receive interrupt types on lower 8-bit data lines.

\overline{RD} , \overline{WR} and \overline{CS} :

The \overline{RD} and \overline{WR} inputs control data transfer when the device is selected by asserting its chip select (\overline{CS}) input low. Usually \overline{RD} and \overline{WR} pins are connected to the system \overline{RD} and \overline{WR} lines. \overline{CS} may be connected to address decoder's output.

Address pin A_0 :

A_0 input of 8259A is used to select one of the two internal addresses in the device. This pin may be connected to any of the system address lines.

Internal Block Diagram of 8259A (Cont.)

Cascade lines (CAS2-CAS0):

The cascade lines (CAS2-CAS0) are used as outputs from the master to the slaves for cascading multiple 8159As in a system. The master outputs a 3-bit slave identification number on these lines. Each slave in a system is assigned a 3-bit ID as part of its initialization. Sending this 3-bit ID number enables the slave.

Slave program/Enable buffer (SP/\overline{EN}):

This pin is a dual function pin, when the 8259A is in buffered mode, this is an output that controls the data bus transceivers in a large microprocessor based system. When the 8259A is not in the buffered mode, this pin programs the device as a master (1) or a slave(0). When we use only one 8159A in our system, the SP/\overline{EN} pin is tied high (1).

INT Pin:

The interrupt output (INT) pin of 8259A is connected to the INTR pin on the microprocessor (8086) when there is only one 8259A in the system. In a system with master and slaves, only master's INT pin is connected to 8086. The slave's INT pins are connected to different IR pins of the master.

Internal Block Diagram of 8259A (Cont.)

Interrupt acknowledge (\overline{INTA}) pin:

This input pin of 8259A is connected to the \overline{INTA} output of the 8086.

Interrupt request inputs (IR7 – IR0) :

The eight interrupt request inputs (IR7-IR0) are used to request an interrupt by the external devices in case of single 8159A system. In case of multiple 8259A system these input pins (IR7-IR0) of the master is connected to output INT pins of the slaves. Unused IR inputs should be tied to ground so that a noise pulse can not accidentally cause an interrupt. An interrupt signal must remain high on an IR input until after the falling edge of the first \overline{INTA} pulse.

The interrupt Mask Register (IMR):

This register is used to disable (mask) or enable (unmask) individual interrupt inputs. Each bit in this register corresponds to the interrupt input with the same number. We can unmask an interrupt input by sending a command word with a 0 in the bit position that corresponds to that input.

Internal Block Diagram of 8259A (Cont.)

The Interrupt Request Register (IRR):

The IRR keeps track of which interrupt inputs are asking for service. If an interrupt input has an interrupt signal on it, then the corresponding bit in the IRR will be set.

The In Service Register (ISR):

The ISR keeps track of which interrupt inputs are currently being serviced. For each input that is currently being serviced, the corresponding bit will be set in the ISR.

The Priority Resolver:

The Priority Resolver acts as a “judge” that determines if and when an interrupt request on one of the IR inputs gets serviced.

8259A Interrupt Operation

To implement interrupt, **the interrupt enable flip-flop in the microprocessor should be enabled by writing the EI instruction** and the **8259A should be initialized by writing control words** in the control register. The 8259A requires two types of control words:

- (a) Initialization Command Words (ICWS)
- (b) Operational Command Words (OCWs)

ICWs are used to set up the proper conditions and specify RST vector address.

The OCWs are used to perform functions such as masking interrupts, setting up status-read operations etc.

Step-1: The IRR of 8259A stores the request.

Step-2: The priority resolver checks 3 registers-

- * The IRR for interrupt requests.
- * IMR for masking bits and
- *the ISR for interrupt request being served.

It resolves the priority and sets the INT high when appropriate.

8259A Interrupt Operation (Cont.)

Step-3: The MPU acknowledges the interrupt by sending signals in \overline{INTA} .

Step-4: After the \overline{INTA} is received, the appropriate priority bit in the ISR is set to indicate which interrupt level is being served and the corresponding bit in the IRR is reset to indicate that the request for the CALL instruction is placed on the data bus.

Step-5: When MPU decodes the CALL instruction, it places two more \overline{INTA} signals on the data bus.

Step-6: When the 8259A receives the second \overline{INTA} , it places the low-order byte of the CALL address on the data bus. At the 3rd \overline{INTA} , it places the high order byte on the data bus. The CALL address is the vector memory location for the interrupt, this address is placed in the control register during the initialization.

Step-7: During the 3rd \overline{INTA} pulse, the ISR bit is reset either automatically (Automatic-End-of-Interrupt-AEOI) or by a command word that must be issued at the end of the service routine (End of Interrupt-EOI). This option is determined by the initialization command word (ICW).

Step-8: The program sequence is transferred to the memory location specified by the CALL instruction.

Programming the 8259A

- ✓ The 8259A is programmed by **initialization and operation command words**.
- ✓ **Initialization command words (ICWs)** are programmed before the 8259A is able to function in the system and dictate the basic operation of 8259A.
- ✓ **Operation command word (OCWs)** are programmed during the normal course of operation. The OCWs control the operation of the 8259A.

Initialization Command Words (ICWs)

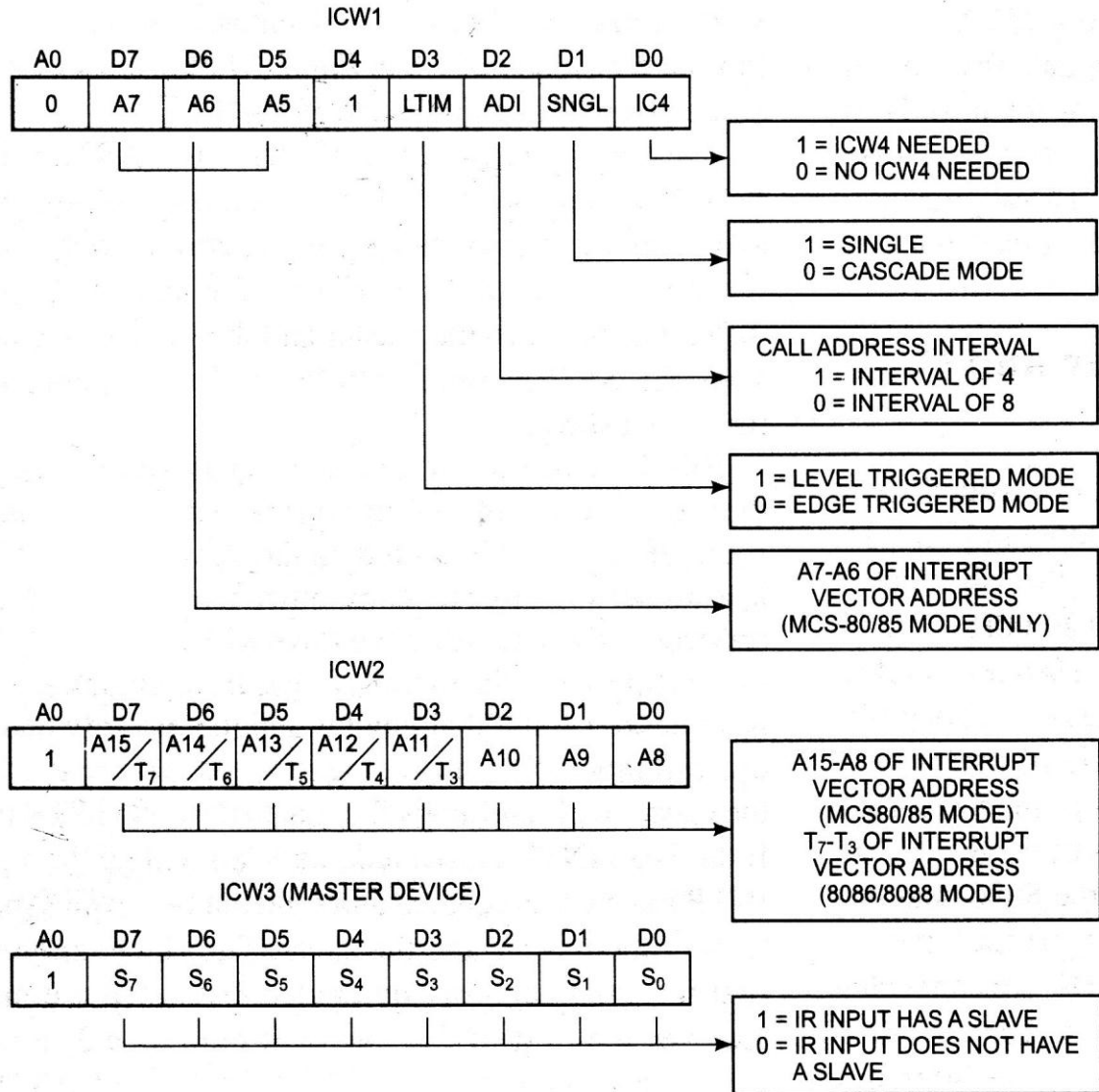
There are four initialization command words (ICWs) for the 8259A that are selected when A0 pin is logic one. If a single 8259A is used in a system, ICW1, ICW2 and ICW4 must be programmed. If cascade mode is used in a system then all four ICWs must be programmed.

ICW1: Programs the basic operation of 8259A. To program this ICW for 8086-Pentium 4 operation, place a logic 1 in bit **IC4**. Bits **ADI, A7, A6, A5** are don't care for microprocessor operation and only apply to the 8259A when used with an 8-bit 8085 microprocessor. This ICW selects single or cascade operation by programming the **SNGL** bit. The **LTIM** bit determines whether the interrupt request inputs are positive edge-triggered or level-triggered.

Initialization Command Words (Cont.)

ICW2: Selects the vector number used with the interrupt request inputs. For example, if we decide to program the 8259A so it functions at vector locations 08H-0FH, we place 08H in the command word

ICW3 (Master Device): Only used when ICW1 indicates that the system is operated in cascade mode. ICW3 (for master device) indicates where the slave is connected to the master. Suppose, we have two slaves connected to a master using IR0 and IR1. The master is programmed with an ICW3 of 03H.



Initialization Command Words (Cont.)

ICW3 (Slave Device): ICW3 (for slave device) indicates where the slave is connected to the master. Suppose, we have two slaves connected to a master using IR0 and IR1. One slave is programmed with an ICW3 of 01H and other with an ICW3 of 02H.

ICW4 : Programmed for use with the 8086 to Pentium 4 microprocessor, but is not programmed in a system that functions with the 8085 microprocessor. The rightmost bit must be logic 1 to select operation with the 8086-Pentium 4 microprocessor, and the remaining bits are programmed as follows:

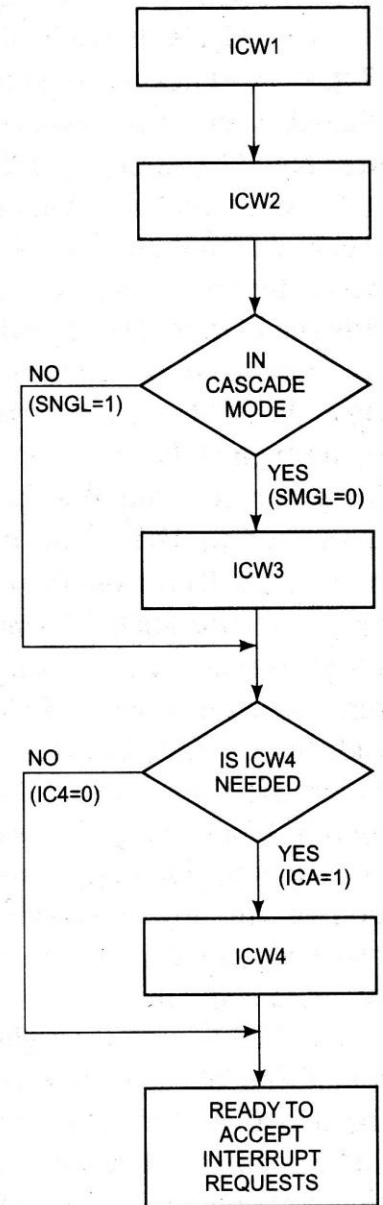
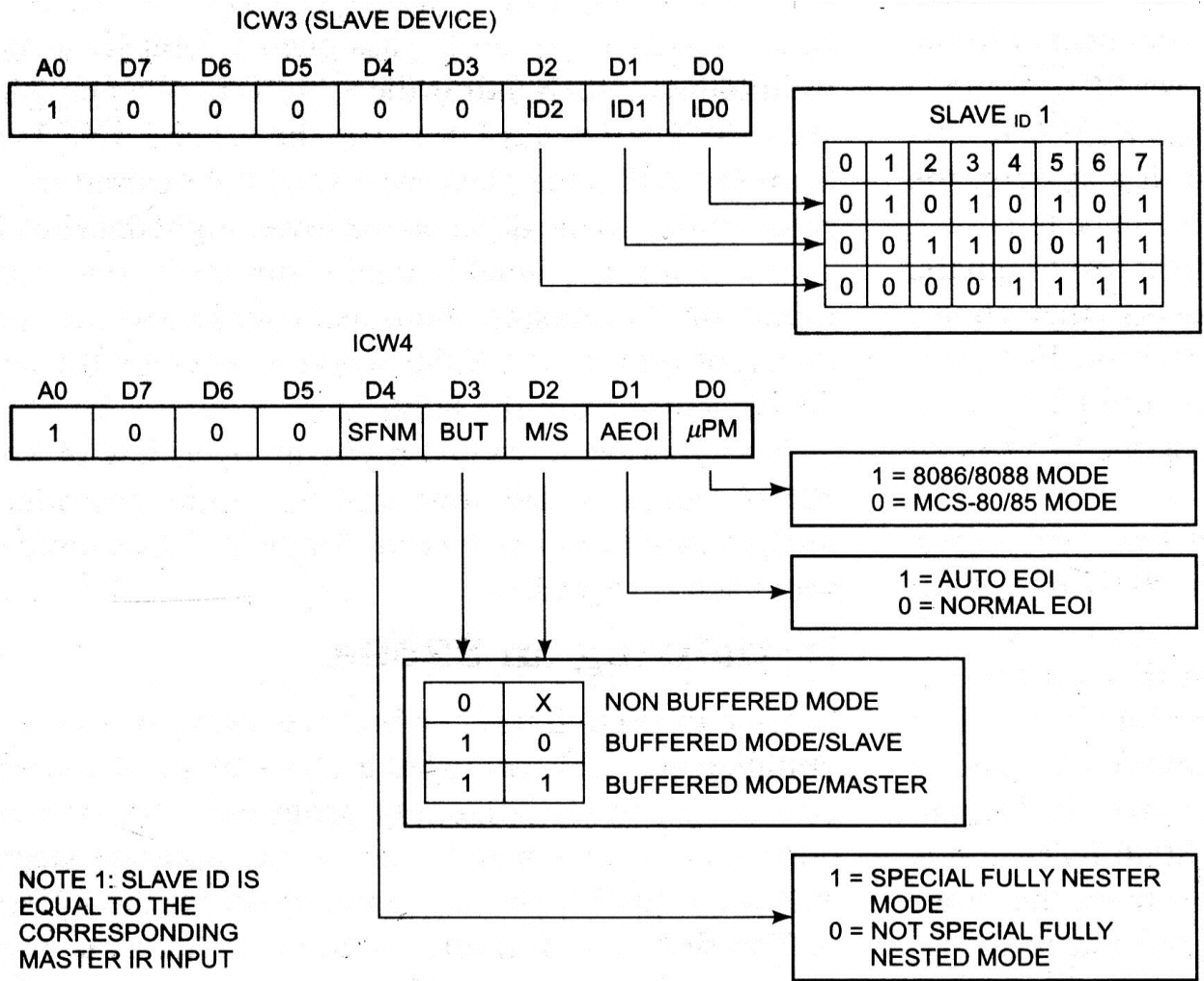
SFNM (Special Fully-Nested Mode)- If SFNM=1, then it selects the special fully-nested mode of operation for the 8259A. At that time, this allows the highest-priority interrupt request from a slave to be recognized by the master while is processing another interrupt from a slave.

Normally (if SFNM=0), only one interrupt request is processed at a time and others are ignored until the process is complete.

BUF and M/S- Buffered and master slave are used together to select buffered operation or nonbuffered operation for the 8259A as master or a slave.

AEOI (Automatic end of interrupt)- Selects automation or normal end of interrupt. The EOI commands of OCW2 are used only if the AEOI mode is selected by ICW4. If AEOI is selected, the interrupt automatically resets the interrupt request bit and does not modify priority. This is the preferred mode of operation for the 8259A and reduces the length of the interrupt service procedure.

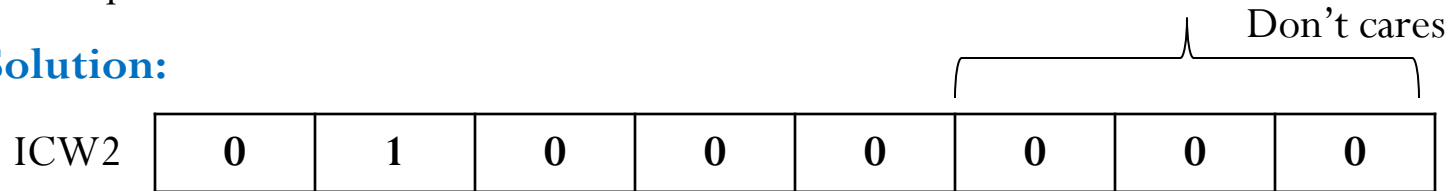
Initialization Command Words (Cont.)



Problems related to ICWs

Problem-1: Determine the control word ICW2 required for the PIC so that inputs IRQ0 to IRQ7 correspond to type numbers 40H to 47H. From which memory locations will the processor fetch the interrupt vectors?

Solution:

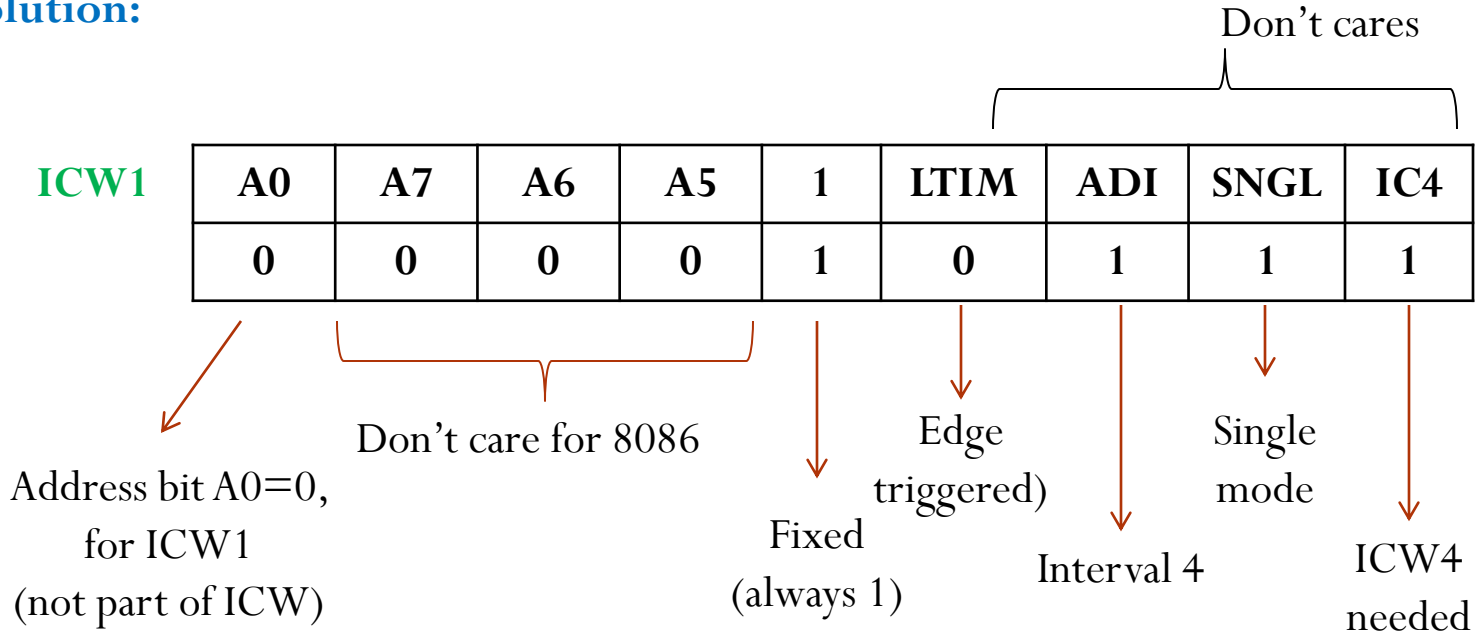


Input	Type Number output by PIC	Interrupt vector location
IRQ0	40H	00100:00103H (4X40=100H)
IRQ1	41H	00104:00107H
IRQ2	42H	00108:0010BH
IRQ3	43H	0010C:0010FH
IRQ4	44H	00110:00113H
IRQ5	45H	00114:00117H
IRQ6	46H	00118:011BH
IRQ7	47H	0011C:0011FH

Problems related to ICWs

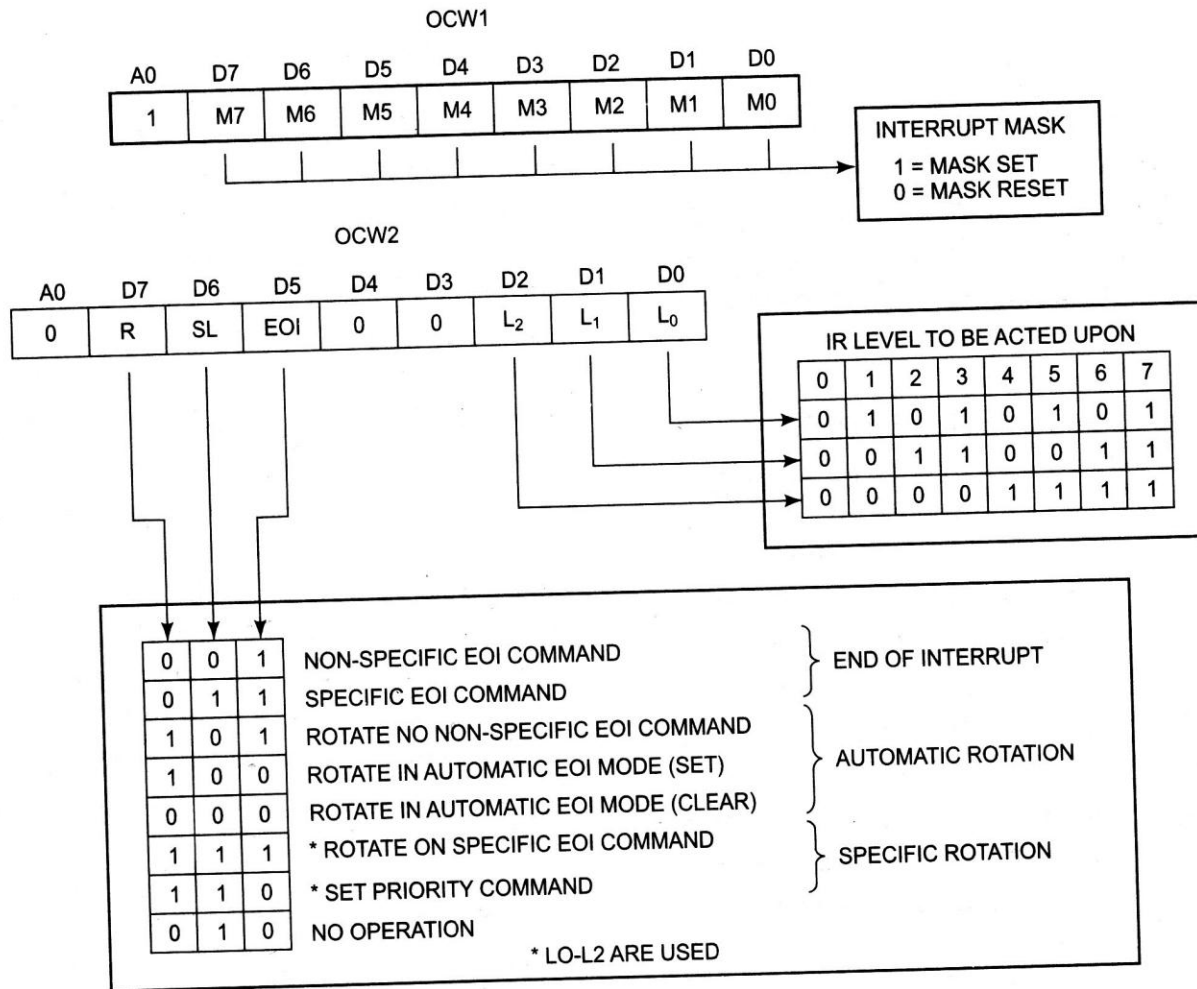
Problem-2: Determine the value for ICW1 for 8086 mode, if it is triggered by rising edge trigger and there is a single PIC. Also use interval 4 and note that ICW4 is needed .

Solution:



Operation Command Words (OCWs)

The operational command words (OCWs) are used to direct the operation of the 8259A once it is programmed with ICW. The ICWs are selected when A0= 0 level, except for OCW1, which is selected for A0=1.



Operation Command Words (OCWs)

OCW1: Used to set and read the interrupt mask register. When a mask bit is set, it will turn off (mask) the corresponding interrupt input. The mask register is read when OCW1 is read. Because the state of mask bits is unknown when the 8259A is first initialize, OCW1 must be programmed after programming the ICW upon initialization.

OCW2: Programmed only when the AEOI mode is not selected for the 8259A. In this case, this OCW selects the way that the 8259A responds to an interrupt. The modes are listed as follow-

Nonspecific End-of-Interrupt (Nonspecific EOI Command)- A command sent by the interrupt service procedure to signal the end of interrupt. When this command is sent to the 8259A, it resets the highest priority ISR bit (lowest numbered IS bit, suppose IR4 and IR6 are now in service, IR4 has the highest priority than IR6). This allows the interrupt to take action again or a lower priority interrupt to take effect.

Specific EOI Command: A command that allows a specific interrupt request to be reset. The exact position is determined with bits L2-L0 of OCW2.

Operation Command Words (OCWs)

Rotate on Nonspecific EOI Command- A command that functions exactly like the Nonspecific End-of-Interrupt command, except that it rotates the interrupt priorities after resetting the interrupt status register bit. **The level reset by this command becomes the lowest-priority interrupt.** For example, if IR4 was just received by this command, it becomes the lowest-priority interrupt input and IR5 becomes the highest priority.

Rotate on automatic EOI:

Rotate on automatic EOI mode(Set)- Once this command is sent to PIC, it will automatically cause the PIC to perform a **rotate on non-specific EOI command** during INTA bus cycles. This command must only be sent to the 8259A once if this mode is desired.

Rotate on automatic EOI mode(Clear)- To disable the rotate on automatic EOI mode this clear command should be sent to 8259A.

Rotate on specific EOI: Functions as the specific EOI, except that it selects rotating priority.

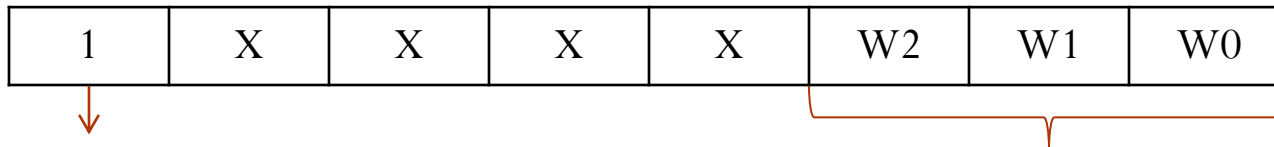
Set priority- Allows the programmer to set the lowest priority interrupt input using the L2-L0 bits, thus fixing all other priority.

Operation Command Words (Cont.)

OCW3: Selects the register to be read, the operation of the special mask register and the poll command.

Reading register- Both the interrupt request register (IRR) and in-service register (ISR) are read by programming OCW3. [Note- Interrupt mask register (IMR) is read through OCW1, to read the IMR A0=1]. To read either IRR or ISR, A0=0. Bit position D0 and D1 of OCW3 select which register (IRR or ISR) is read.

Poll Mode- In this mode, the INT output of PIC is inhibited and the device is used as a prioritized poller. Performing an I/O read instruction from the PIC (either port address) returns the status word shown in fig. The rightmost three bits of the poll word indicate the active interrupt request with the highest priority. The leftmost bit indicates whether there is an interrupt and must be checked to determine whether the right three bit contains valid information.

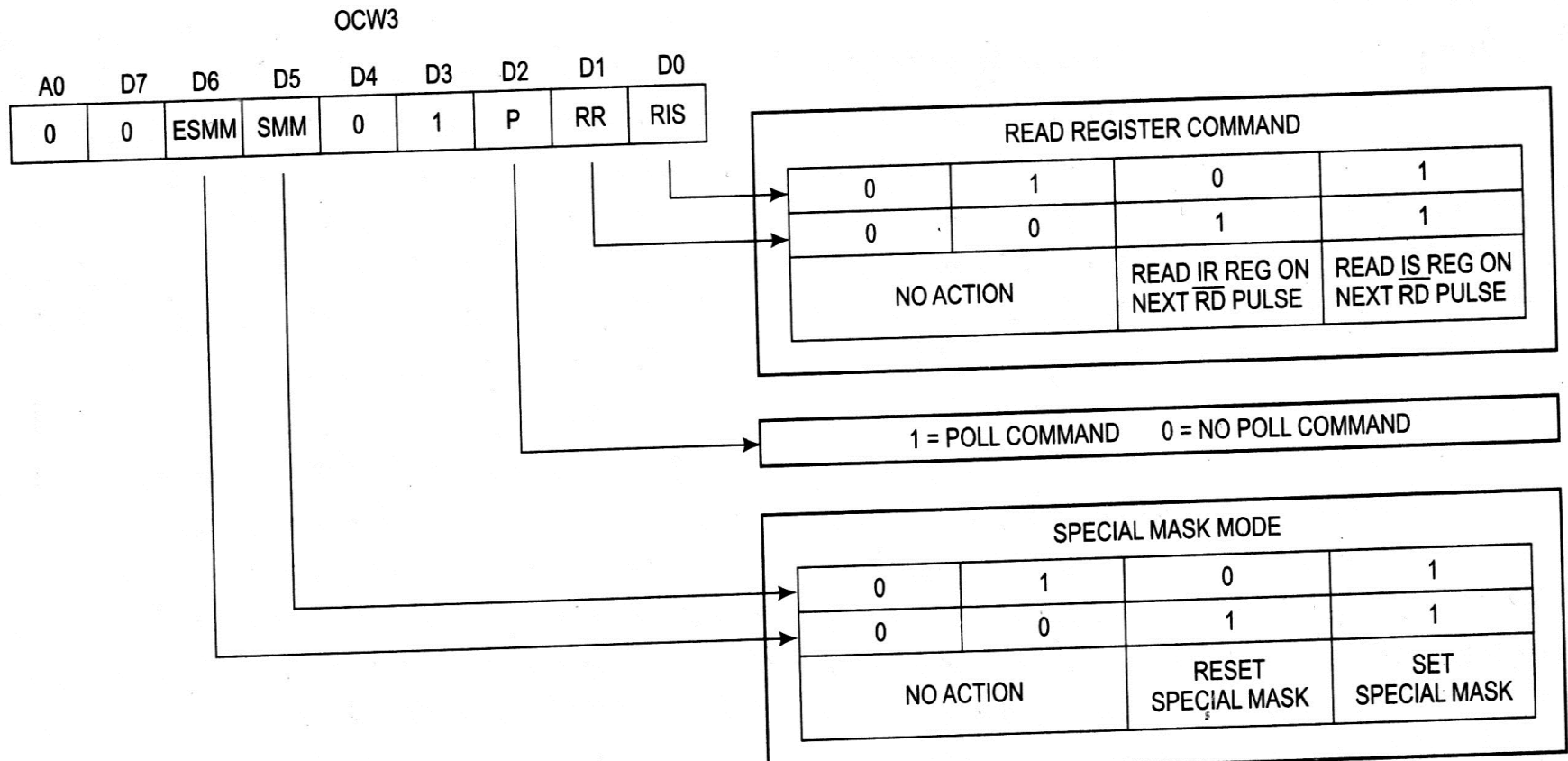


1 signifies on active input

Binary code of highest priority level requesting service.

Operation Command Words (Cont.)

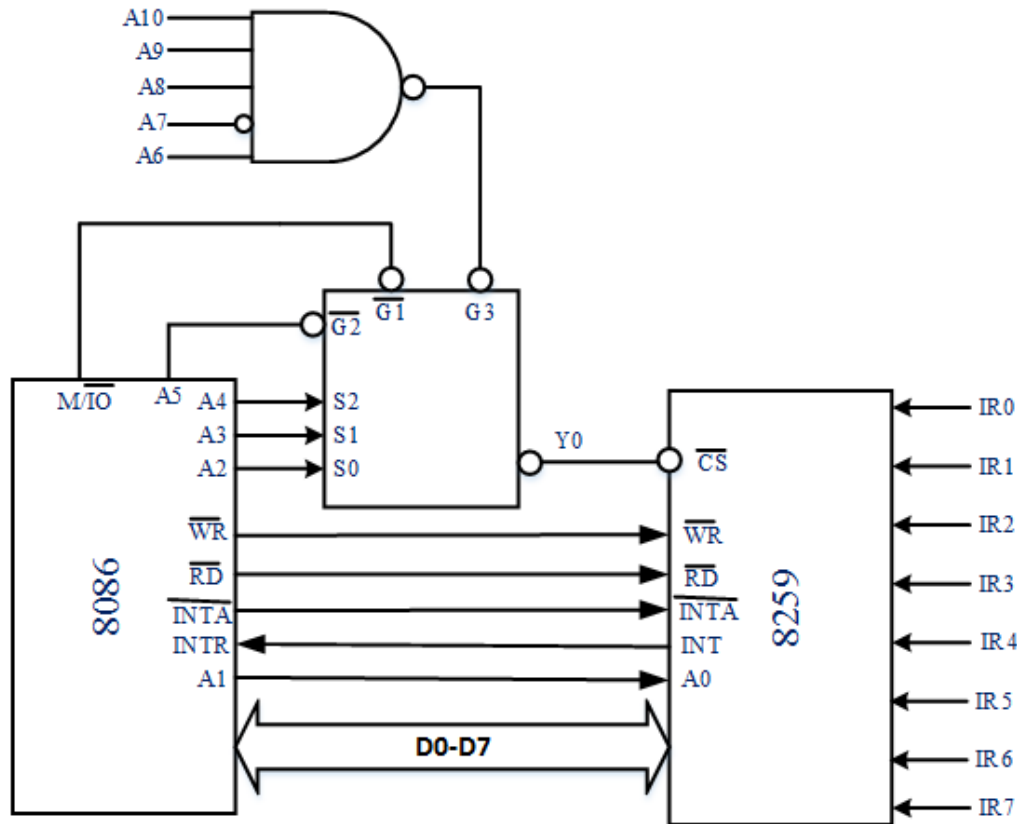
Special Mask Mode- As we have seen, the PIC normally inhibits interrupt requests of equal or lower priority than that of currently in service. In the special mask mode, this is altered to allow interrupts on all inputs except the input currently in service.



Interfacing and Programming 8259A

Problem-3: Show 8259A interfacing connections with 8086 at the address 074x. Write an ALP (Assembly language procedure) to initialize the 8259A in single level triggered mode with call address interval of 4, non-buffered on special fully nested mode. Then set the 8259A to operate with IR6 masked, IR4 as bottom priority level with rotate on specific EOI mode. Set special mask mode of 8259A. Also, read IRR and ISR into registers BH and BL respectively. Base address=80H.

Solution:

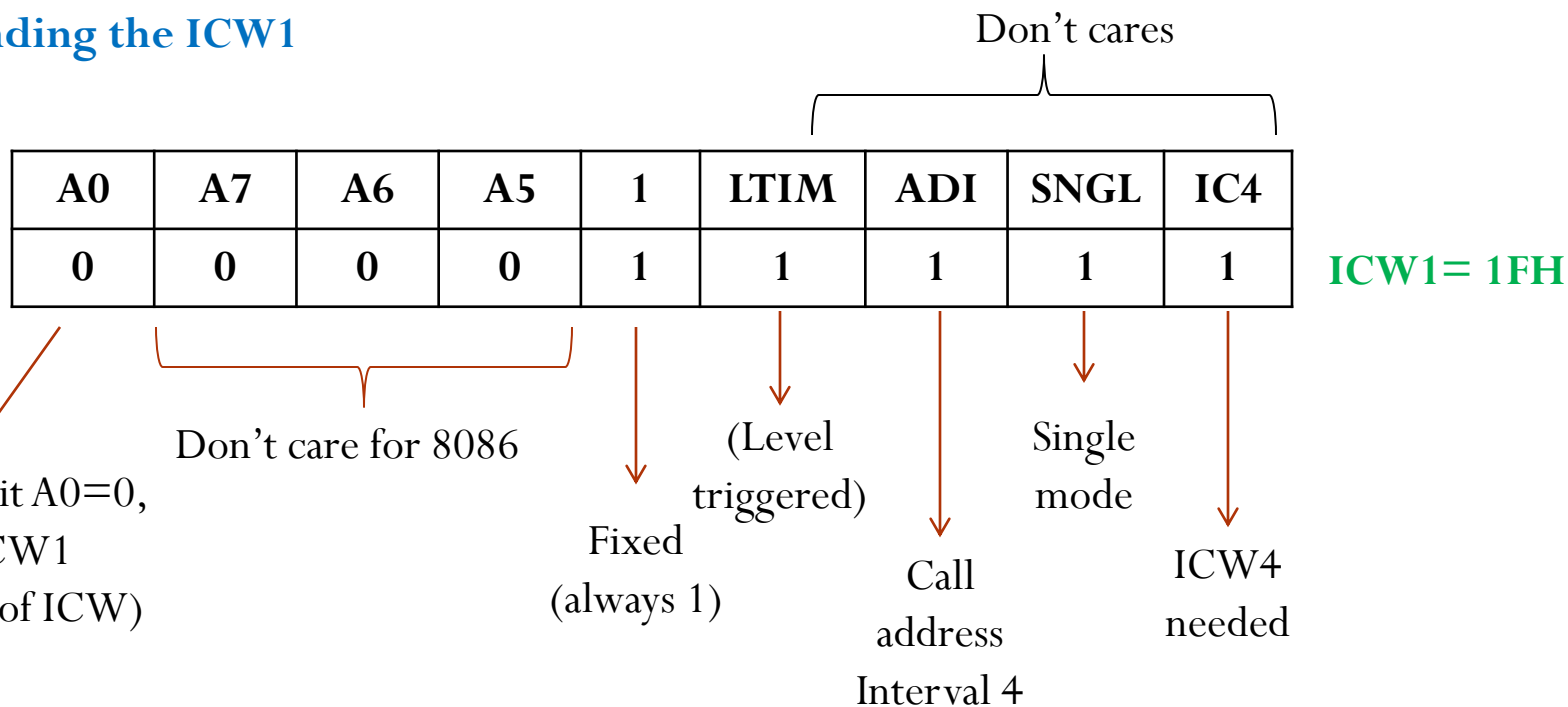


Solution of Problem 3(Cont.)

Step-1: Finding Address

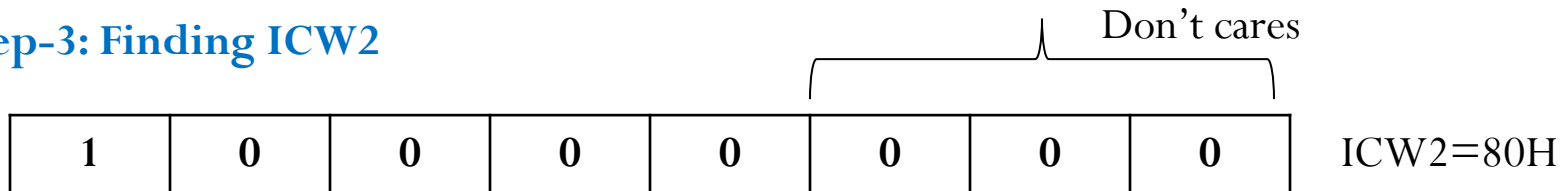
A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Address of Port
0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0740H (ICW1, OCW2, OCW3)
0	0	0	0	0	1	1	1	0	1	0	0	0	0	1	0	0742H (ICW2, ICW4, OCW1)

Step-2: Finding the ICW1



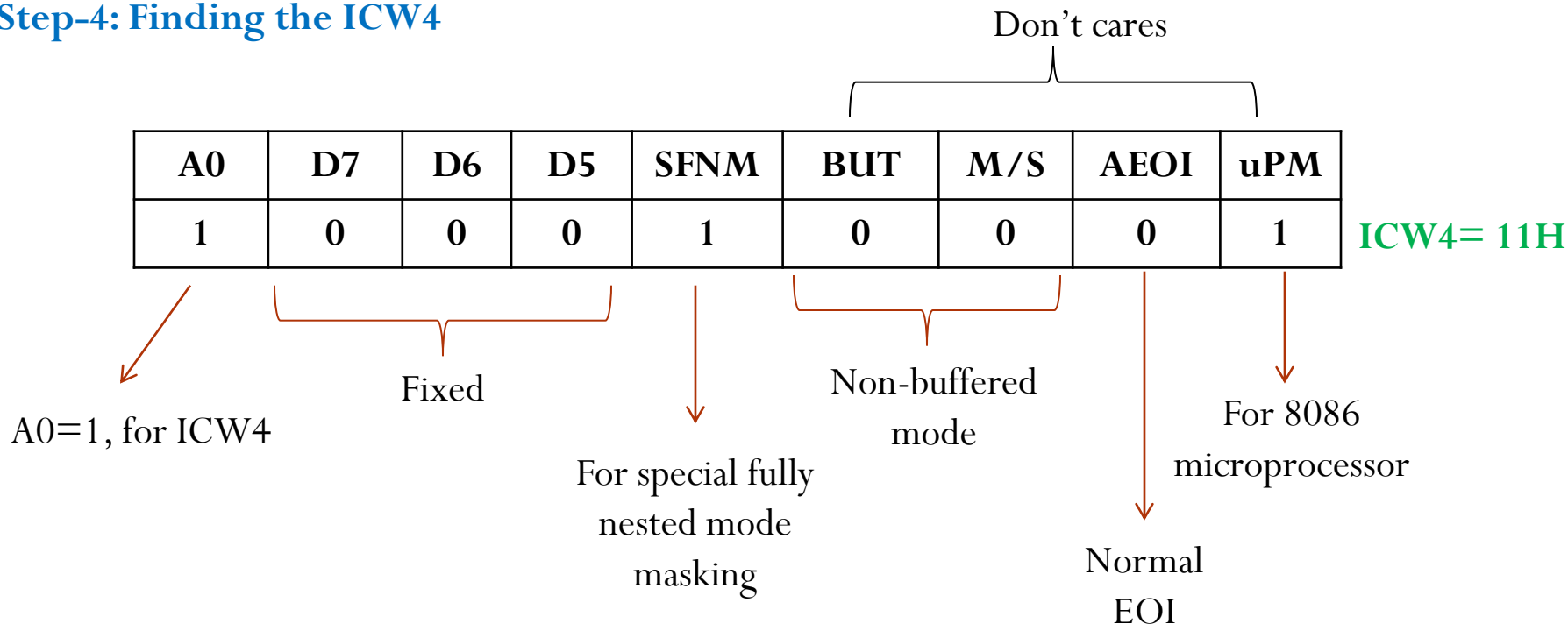
Solution of Problem 3(Cont.)

Step-3: Finding ICW2



ICW3 is not needed as 8259A is set in single mode.

Step-4: Finding the ICW4



Solution of Problem 3(Cont.)

Step-5: Finding OCW1

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	0	1	0	1	0	0	0	0

OCW1= 40H

↓
IR6 is masked

Step-6: Finding the OCW2

A0	D7	D6	D5	D4	D3	D2	D1	D0
10	1	1	1	0	0	1	0	0

OCW2= E4H

↙
A0=0, for OCW2

┌──────────┴──────────┐
Rotate on Specific
EOI command

┌──────────┴──────────┐
Fixed

┌──────────┴──────────┐
Bottom priority
level set at IR4

Solution of Problem 3(Cont.)

Step-8: Finding OCW3

The OCW3 sets the special mask mode and reads ISR and IRR using the following control words-

For reading IRR:

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	1	0	1	0	1	0

OCW3= 6AH

fixed

Special masked mode

Fixed

No poll command

Read IRR

For reading ISR:

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	1	0	1	0	1	1

OCW3= 6BH

fixed

Special masked mode

Fixed

No poll command

Read ISR

Solution of Problem 3(Cont.)

Assembly Language Procedure

INTERRUPT PROC NEAR

MOV AL, AFH ; Loading ICW1 to AL

MOV DX, 0740H ; Loading Address of ICW1 to DX (Variable port addressing)

OUT DX, AL ; Sending ICW1 to port (address: 0740H) of 8259A

MOV DX, 0742H ; address of ICW2

MOV AL, 80H ; Loading ICW2 to AL which select the vector address

MOV DX, AL ; Sending ICW2 to port (address: 0742H) of 8259A

MOV AL, 11H ; Loading ICW4 to AL

OUT DX, AL ; Sends ICW4 to 0742H

MOV AL, 40H ; Loading OCW1 to AL

OUT DX, AL ; Sends OCW1 to 0742H

MOV AL, E4H ; Loading OCW2 to AL

MOV DX, 0740H ; Address of OCW2

Solution of Problem 3(Cont.)

Assembly Language Procedure (Cont.)

MOV DX, AL ; Sending OCW2 to 0740H address.

MOV AL, 6AH ; Loading OCW3 for reading IRR

OUT DX, AL; Sending OCW3 to 0740H address.

IN AL, DX ; Reading IRR and store to AL

MOV BH, AL; Store IRR into BH

MOV AL, 6BH ; Loading OCW3 for reading ISR

OUT DX, AL; Sending OCW3 to 0740H address.

IN AL, DX ; Reading ISR and store to AL

MOV BL, AL; Store ISR into BL

RET

INTERRUPT ENDP

References:

1. Microprocessor and Interfacing by Douglas V Hall
2. The Intel Microprocessors by Barry B. Brey
3. Microcomputers and Microprocessors By John Uffenbeck
4. 8259A Lecture Slides, Available on:
www.slideshare.net/meghadityaroychaudhury/8259-a-12246982