Programmable Interrupt Controller

8259

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
- 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.

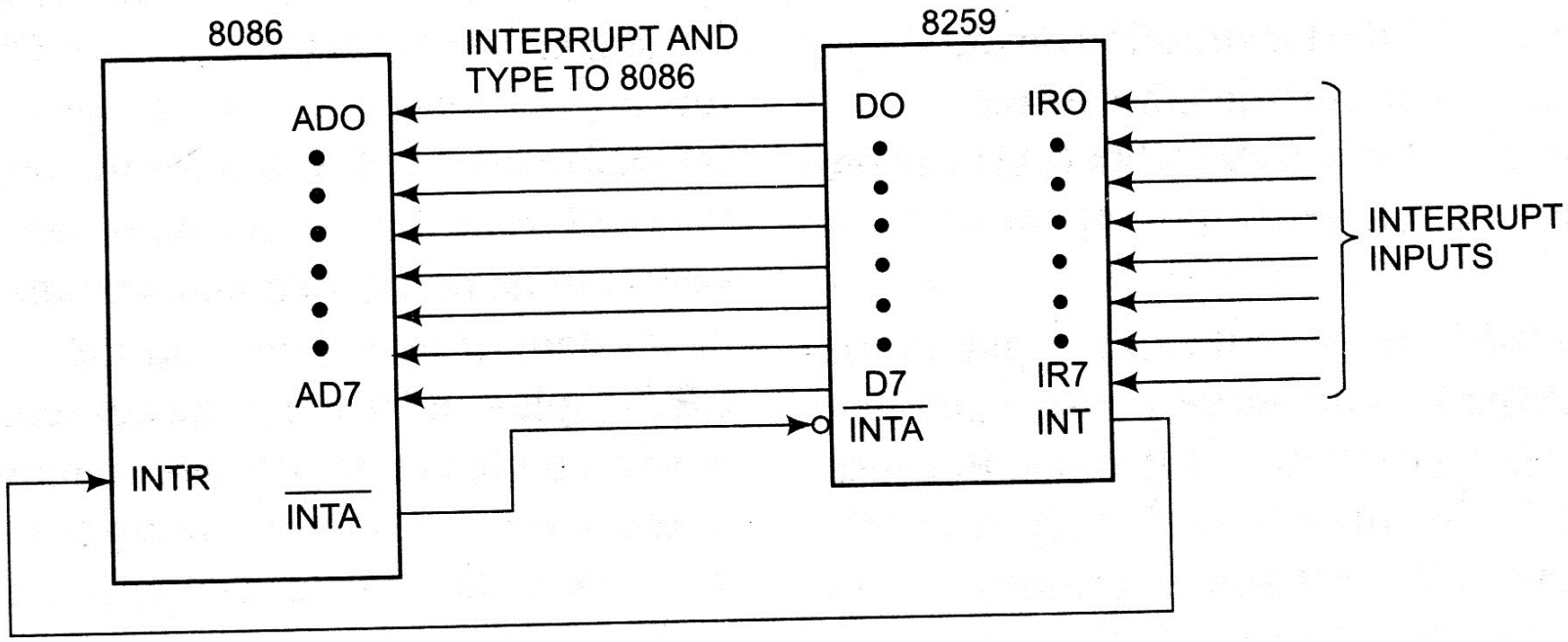
- 8259A is a hardware device to support the interrupt mechanism
- It can support up to 8 vectored priority encoded interrupts to the microprocessor
- Can be expanded (using more 8259) to accept up to 64 interrupt requests using master/slaves configuration

Features of 8259

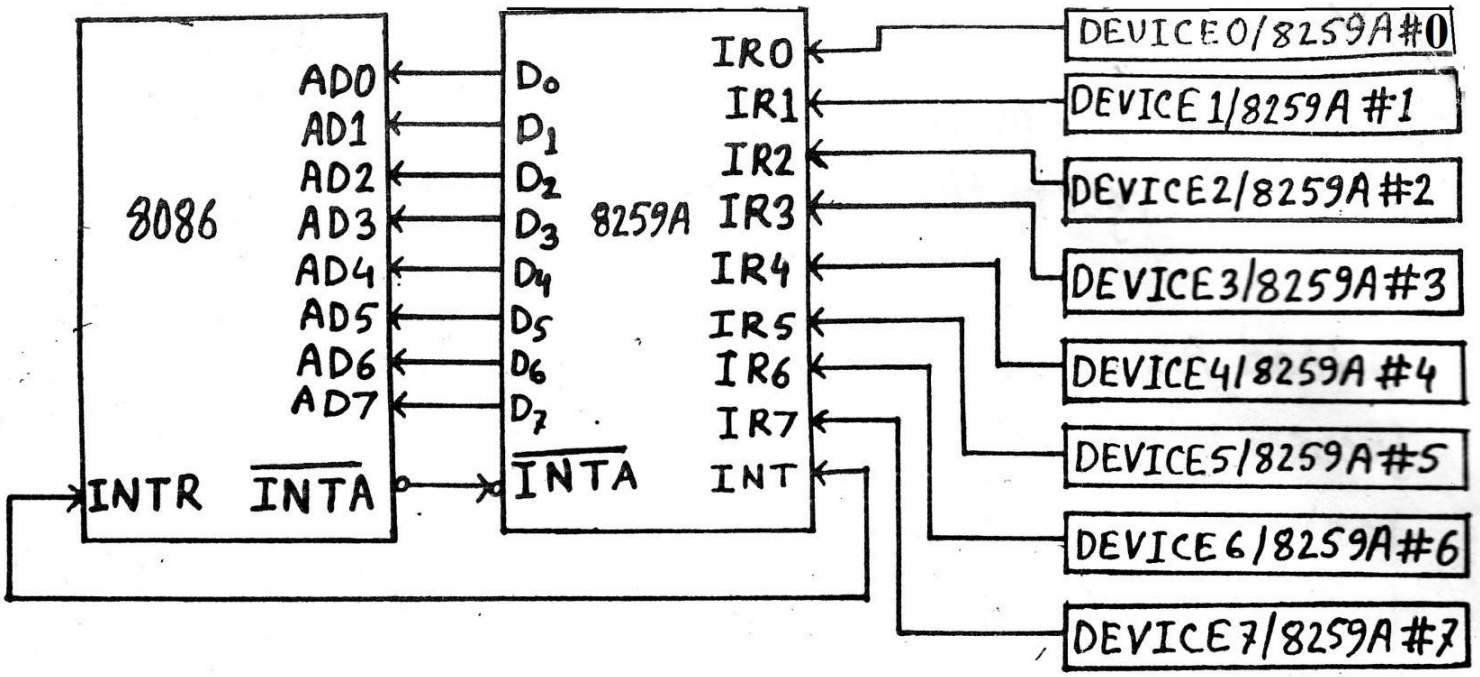
- The 8259 programmable interrupt controller has 8 interrupt pins thus can handle 8 interrupt inputs.
- The priority of interrupts in 8259 can be programmed. The priority of interrupts is decided by the different operating modes.
- We know that a single 8259 can handle 8 interrupt inputs but by cascading multiple 8259, it can handle maximal 64 interrupt inputs.
- 8259 allows individual masking of each generated interrupt using interrupt mask register.

- 8259 is programmed in a way that it can handle either edge-triggered or level-triggered interrupt request at a time
- If multiple interrupts are generated, then 8259 holds the status of interrupts that are masked, in-service and pending.
- It reduces the software and real-time overhead generated due to handling multilevel priority interrupts.

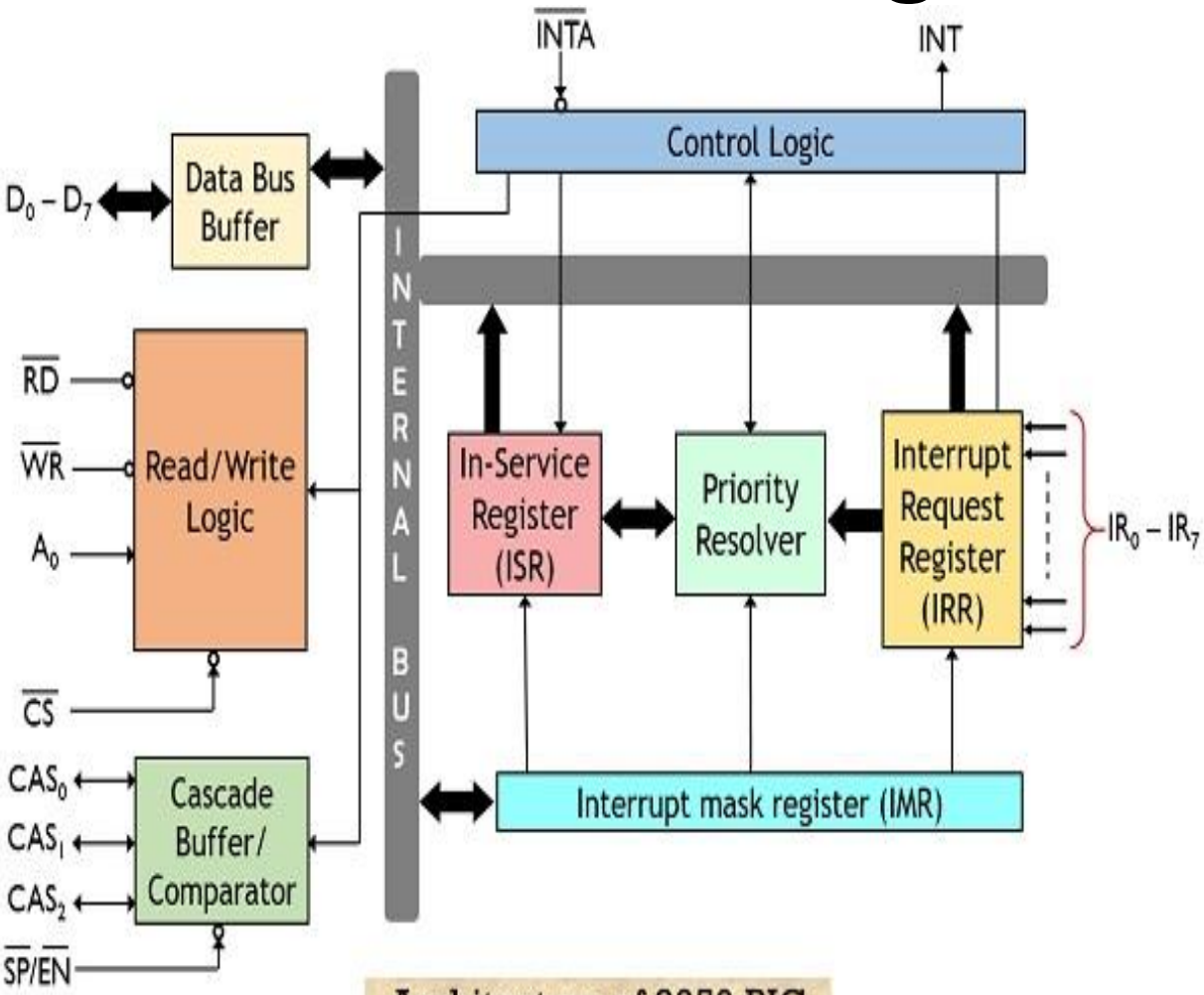
Block Diagram showing an 8259 connected to an 8086

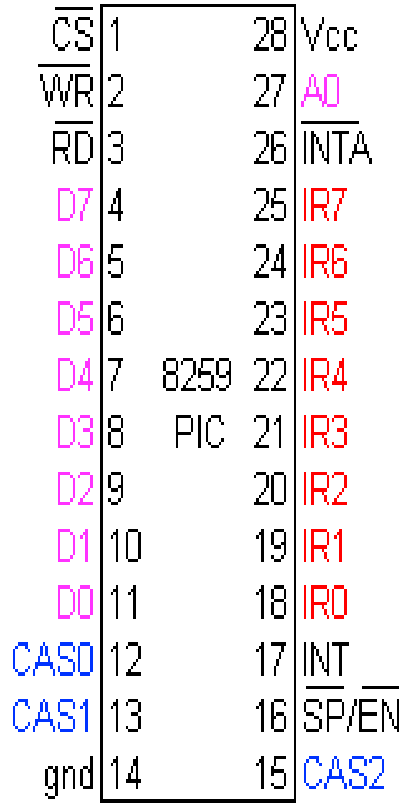


Connection of 8259A with 8086 microprocessor (Cascade Mode)



Block Diagram





Architecture of 8259 PIC

Electronics Desk

- It has an 8-bit of data bus. 8259 never services the interrupt, it simply forwards the interrupts to the microprocessor.
- Thus, the architecture has different units that combinedly functions to increase the interrupts handled by the processor.

- **8-bit data bus:**
- The 8-bit data bus ($D7 - D0$) 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.

- **DATA BUS BUFFER** :This 3-state, bidirectional 8-bit buffer is used to interface the 8259A to the system Data Bus. Control words and status information are transferred through the Data Bus Buffer.

- **READ/WRITE CONTROL LOGIC :** The function of this block is to accept OUTput commands from the CPU. It contains the Initialization Command Word (ICW) registers and Operation Command Word (OCW) registers which store the various control formats for device operation. This function block also allows the status of the 8259A to be transferred onto the Data Bus.

- The RD‘ and WR’ inputs control data transfer when the device is selected by asserting its chip select(CS‘) input low. Usually RD‘ and WR’ pins are connected to the system RD‘ and WR’ lines. CS’ may be connected to address decoder’s output.
- **Address pin $A0$** :
- $A0$ 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.

- **Cascade lines (CAS2-CAS0):**
- The cascade lines (CAS2-CAS0) are used as outputs from the master to the slaves for cascading multiple 8259A’ s 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.

INTERRUPT REQUEST REGISTER (IRR) AND IN-SERVICE REGISTER (ISR)

- The interrupts at the IR input lines are handled by two registers in cascade, the Interrupt Request Register (IRR) and the In-Service (ISR).
- The IRR is used to store all the interrupt levels which are requesting service; and the ISR is used to store all the interrupt levels which are being serviced.

PRIORITY RESOLVER

- This logic block determines the priorites of the bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during INTA pulse.
- INTERRUPT MASK REGISTER (IMR) The IMR stores the bits which mask the interrupt lines to be masked. The IMR operates on the IRR. Masking of a higher priority input will not affect the interrupt request lines of lower quality

The events occuring in an 8086 system:

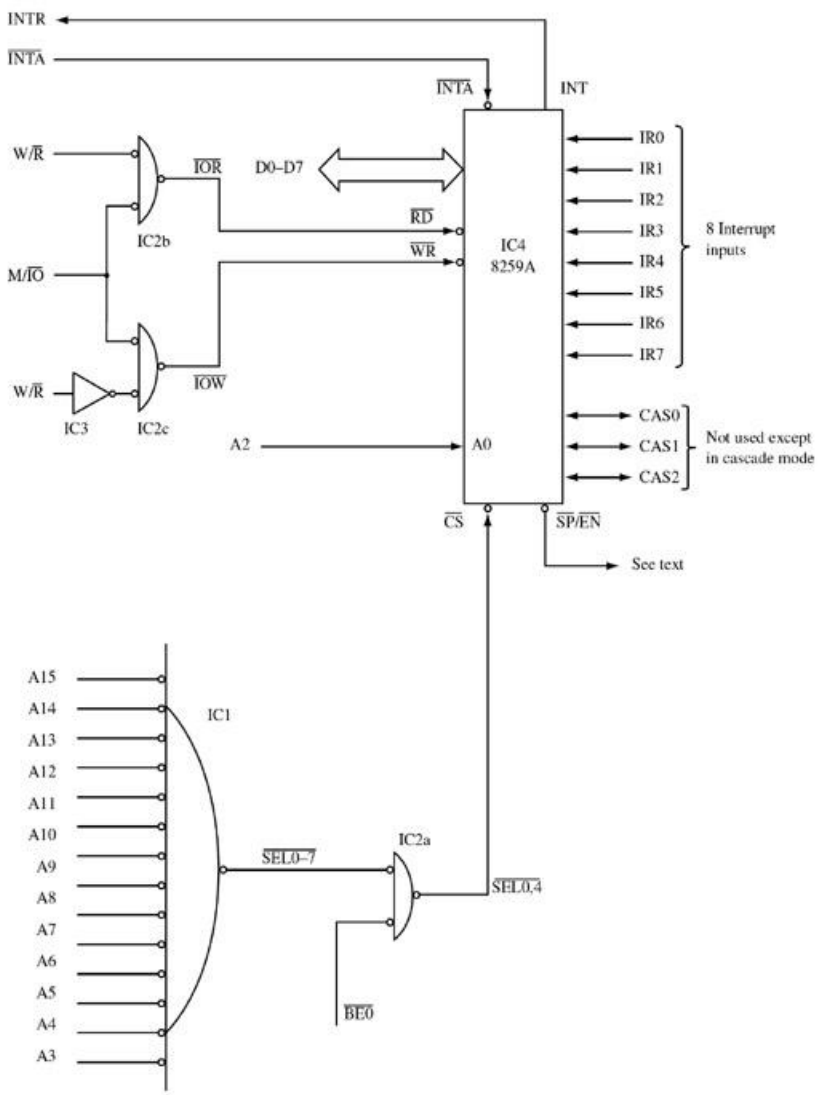
- One or more of the INTERRUPT REQUEST lines (IR7 –0) are raised high, setting the corresponding IRR bit(s).
- The 8259A evaluates these requests, and sends an INT to the CPU, if appropriate.
- The CPU acknowledges the INT and responds with an INTA pulse.
- Upon receiving an INTA from the CPU group, the highest priority ISR bit is set and the corresponding IRR bit is reset. The 8259A does not drive the Data Bus during this cycle.
- The 8086 will initiate a second INTA pulse. During this pulse, the 8259A releases an 8-bit pointer onto the Data Bus where it is read by the CPU.
- This completes the interrupt cycle. In the AEOI mode the ISR bit is reset at the end of the second INTA pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

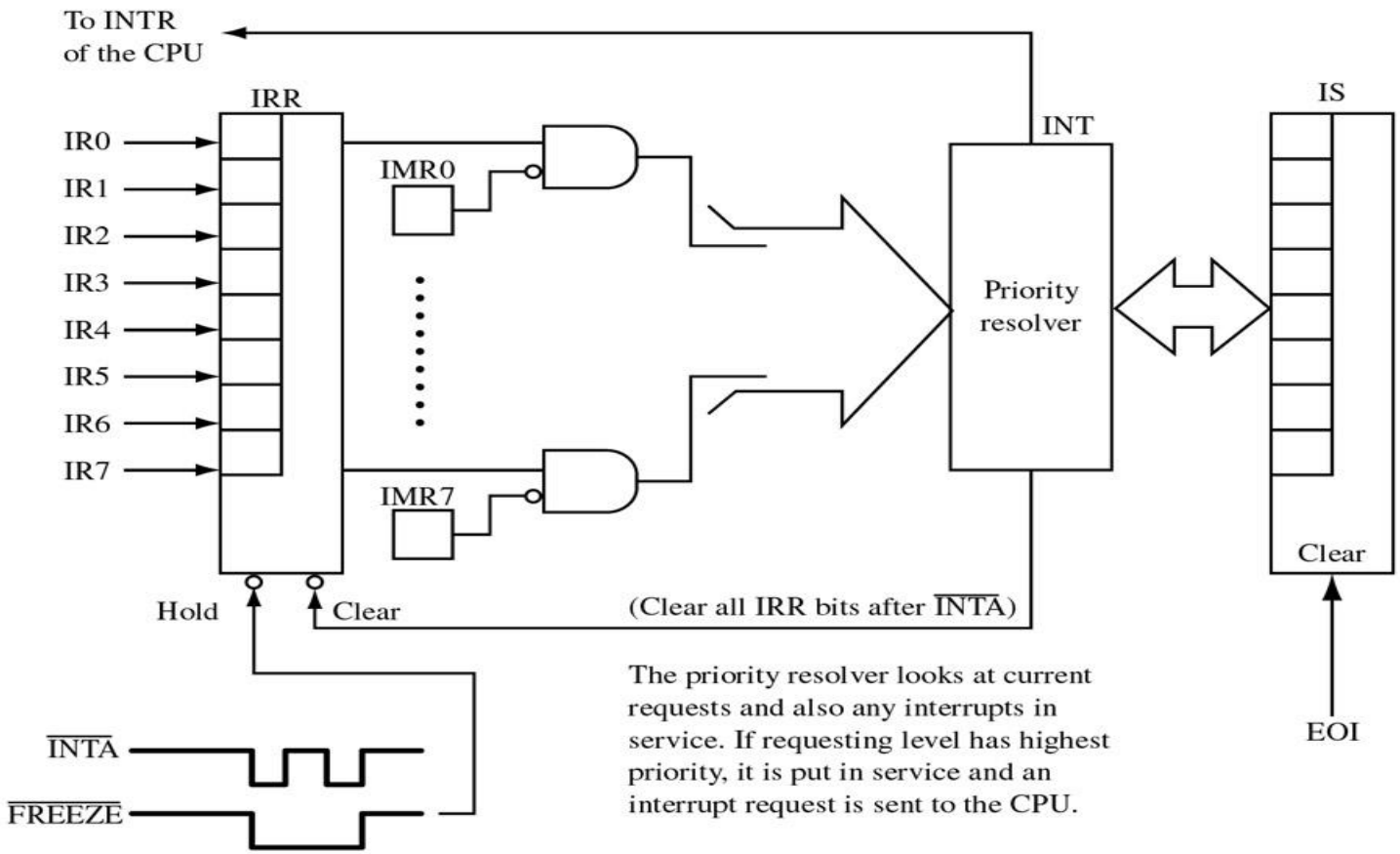
PROGRAMMING THE 8259A

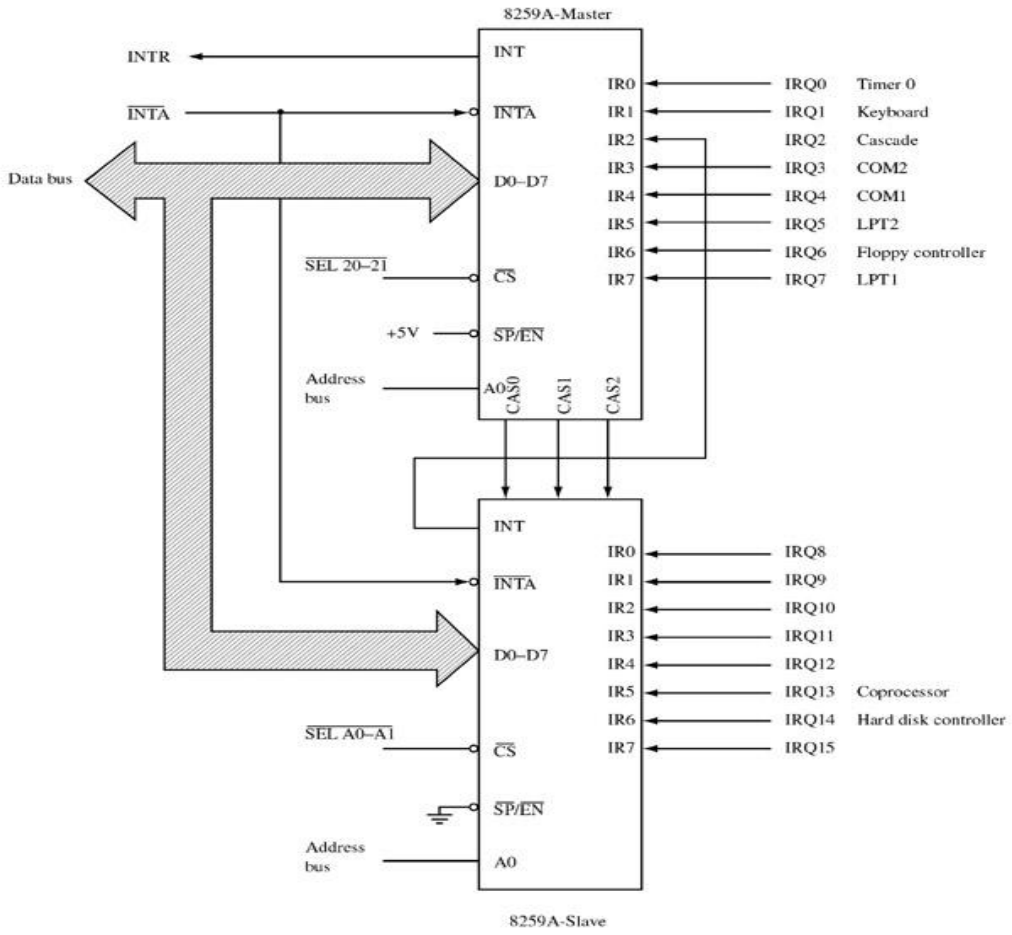
- The 8259A accepts two types of command words generated by the CPU:

1. Initialization Command Words (ICWs): Before normal operation can begin, each 8259A in the system must be brought to a starting point by a sequence of 2 to 4 bytes timed by WR pulses.
2. Operation Command Words (OCWs): These are the command words which command the 8259A to operate in various interrupt modes. These modes are:
   a. Fully nested mode
   b. Rotating priority mode
   c. Special mask mode
   d. Polled mode

The OCWs can be written into the 8259A anytime after initialization.


INTR
INTA
INTA
INT
W/R
IOR
D0–D7
RD
WR
IC4
8259A
IR0
IR1
IR2
IR3
IR4
IR5
IR6
IR7
8 Interrupt inputs
IC2b
M/IO
W/R
IOW
IC3
IC2c
A2
A0
CAS0
CAS1
CAS2
Not used except in cascade mode
CS
SP/EN
See text
A15
A14
A13
A12
A11
A10
A9
A8
A7
A6
A5
A4
A3
IC1
SEL0–7
IC2a
SEL0,4
BE0


To INTR
of the CPU
IRR
IR0
IR1
IR2
IR3
IR4
IR5
IR6
IR7
IMR0
IMR7
INT
Priority
resolver
IS
Clear
EOI
Hold
Clear
(Clear all IRR bits after INTA)
INTA
FREEZE
The priority resolver looks at current requests and also any interrupts in service. If requesting level has highest priority, it is put in service and an interrupt request is sent to the CPU.

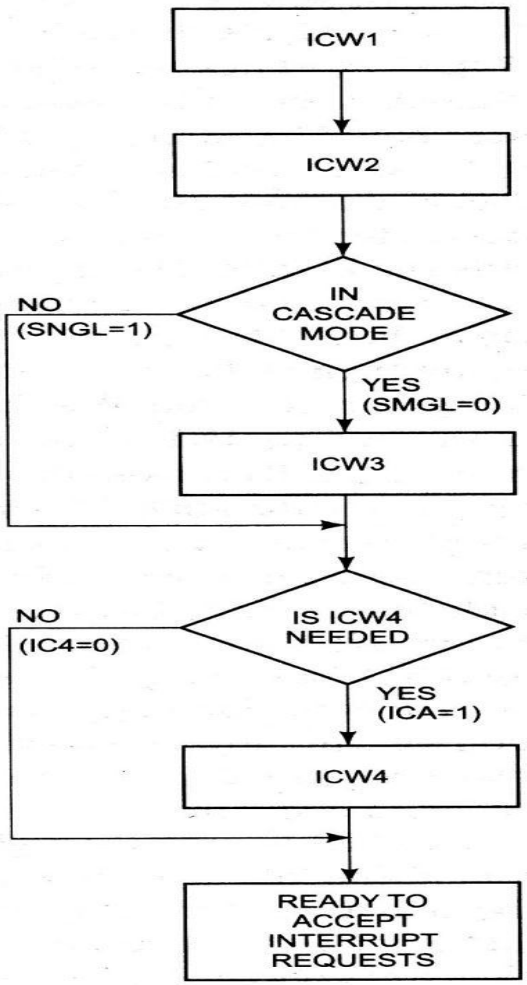


Example of two cascaded PICs

INITIALIZATION COMMAND WORDS

- Before it starts functioning, the 8259A must be initialized by writing two to four command words into the respective command word registers. These are called as initialized command words.
- If A0 = 0 and D4 = 1, the control word is recognized as ICW1. It contains the control bits for edge/level triggered mode, single/cascade mode, call address interval and whether ICW 4 is required or not.
- If A0=1, the control word is recognized as ICW 2. The ICW 2 stores details regarding interrupt vector addresses.

Initialisation sequence of 8259A



- ICW1 starts the intialization sequence during which the following automatically occur.

a. The edge sense circuit is reset, which means that following initialization, an interrupt request (IR) input must make a low-to-high transistion to generate an interrupt.

b. The Interrupt Mask Register is cleared.

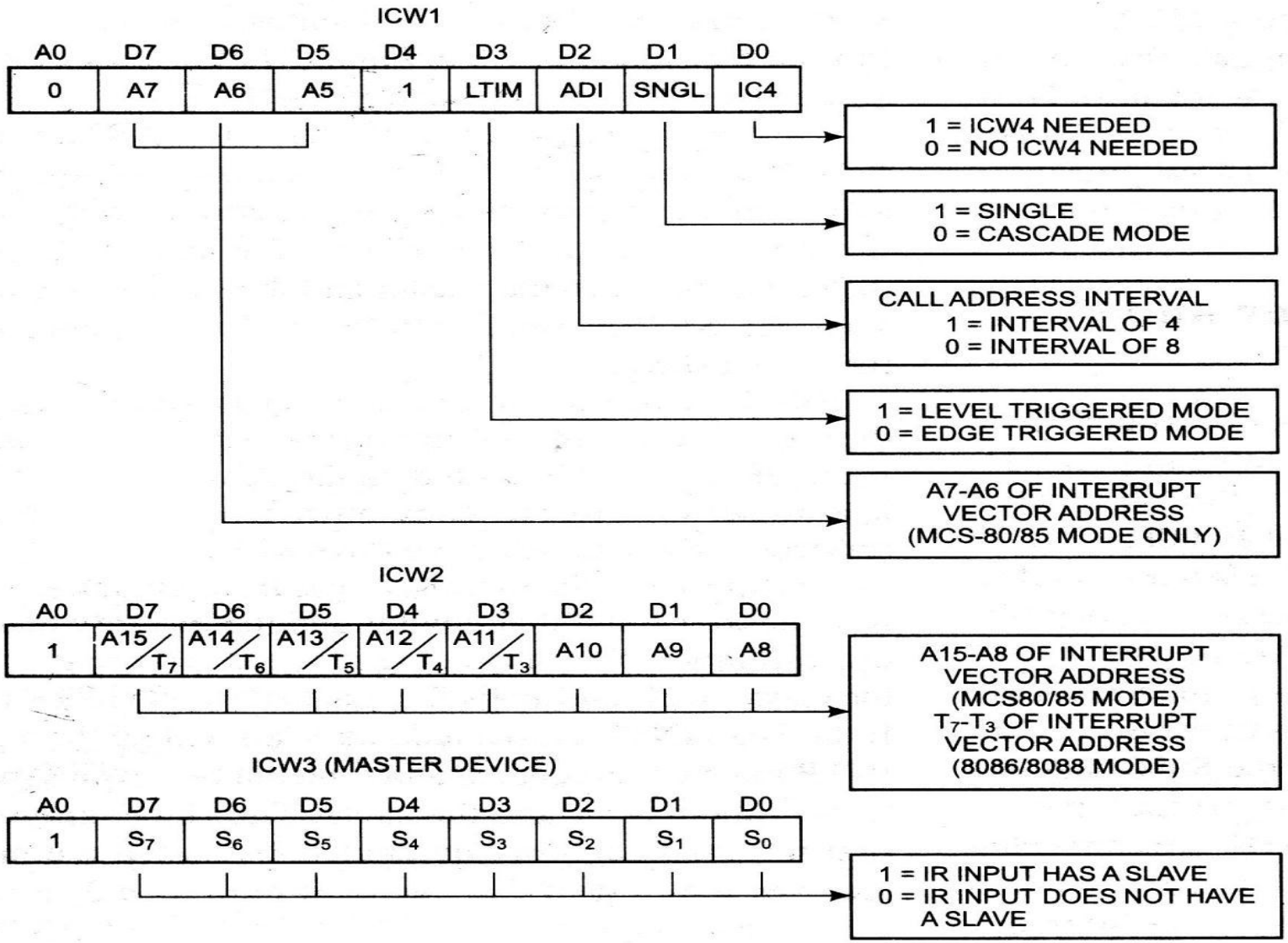
c. IR7 input is assigned priority 7.

d. The slave mode address is set to 7.

e. Special Mask Mode is cleared and Status Read is set to IRR.

f. If IC4 = 0, then all functions selected in ICW4 are set to zero. (Non-Buffered mode*, no AutoEOI, MCS-80, 85 system)

INITIALIZATION COMMAND WORDS



ICW3 (SLAVE DEVICE)

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	ID2	ID1	ID0

SLAVE $_{ID}$ 1

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	BUT	M/S	AEOI	μPM

1 = 8086/8088 MODE
0 = MCS-80/85 MODE

1 = AUTO EOI
0 = NORMAL EOI

0	X	NON BUFFERED MODE
1	0	BUFFERED MODE/SLAVE
1	1	BUFFERED MODE/MASTER

1 = SPECIAL FULLY NESTER MODE
0 = NOT SPECIAL FULLY NESTED MODE

NOTE 1: SLAVE ID IS EQUAL TO THE CORRESPONDING MASTER IR INPUT

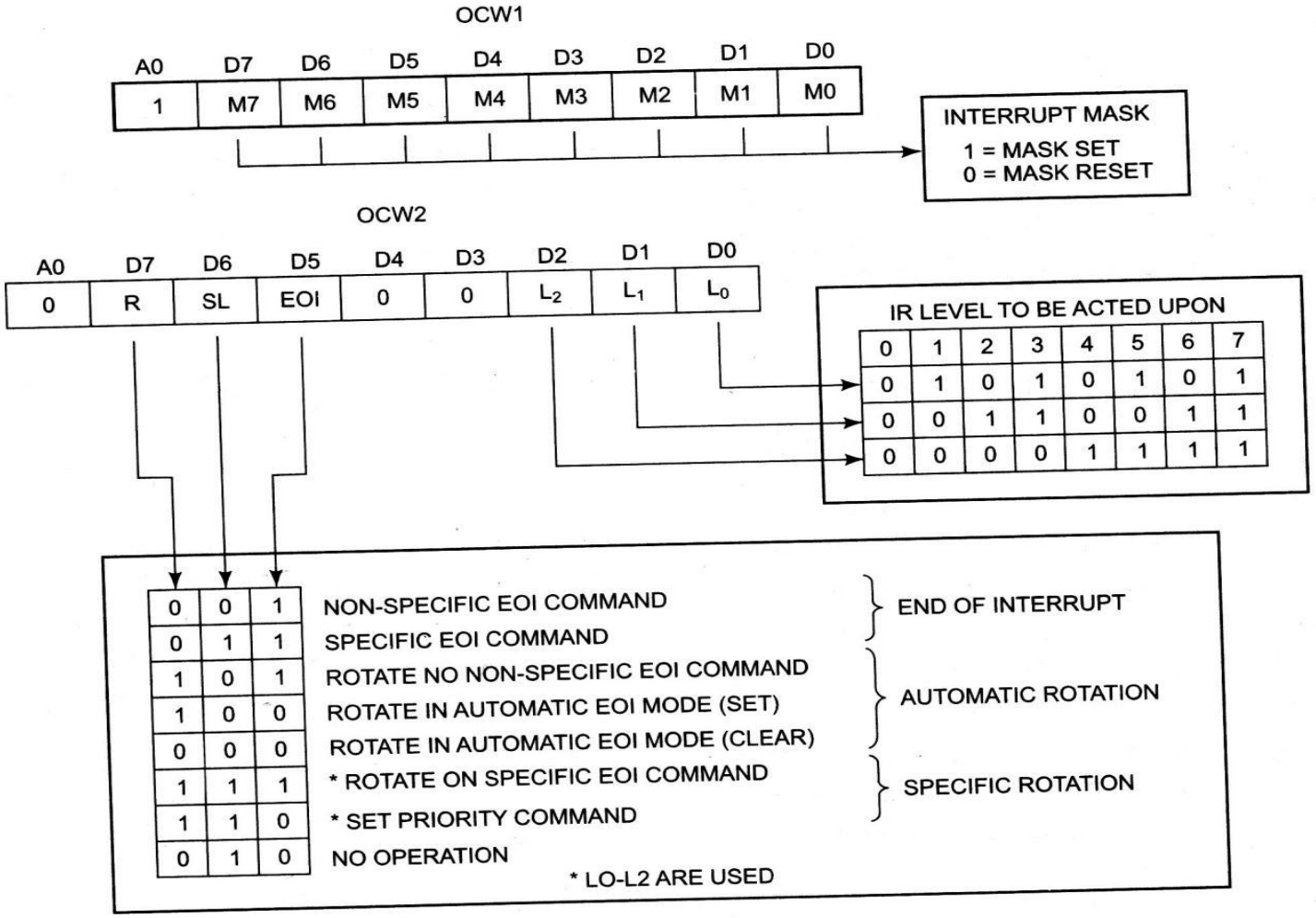
- **SFNM:** If SFNM=1, allows the highest-priority interrupt request from a slave to be recognized by the master while is processing another interrupt from a slave.
- Normally (if SFNM=0), only one interrupt request is processed at a time and others are ignored until the process is complete.
- **If BUF = 1**, the buffered mode is selected. In the buffered mode, SP/EN acts as enable output and the master/slave is determined using the M/S bit of ICW 4.
- **Buffered mode of 8259** − The bus driving buffers are needed in the data bus when 8259 is used in a huge system. The problem is solved by configuring 8259 for buffer mode of operation.SP/EN pin is used as an output to enable data bus.
- **Non-Buffered mode** − Used when the output port enables the Buffer.

- M/S: If M/S = 1, 8259A is a master. If M/S =0, 8259A is slave. If BUF = 0, M/S is to be neglected.
- AEOI: If AEOI = 1, the automatic end of interrupt mode is selected. The major drawback with this mode is that the ISR doesn’t have info on which IR is served.

Operation Command Words:

- Once 8259A is initialized using the previously discussed command words for initialisation, it is ready for its normal function, i.e. for accepting the interrupts but 8259A has its own way of handling the received interrupts called as modes of operation.
- These modes of operations can be selected by programming, i.e. writing three internal registers called as operation command words.

- **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.


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
0 1 1 SPECIFIC EOI COMMAND
1 0 1 ROTATE NO NON-SPECIFIC EOI COMMAND
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
1 1 0 * SET PRIORITY COMMAND
0 1 0 NO OPERATION
* LO-L2 ARE USED
END OF INTERRUPT
AUTOMATIC ROTATION
SPECIFIC ROTATION

- In **OCW 2** the three bits, R, SL and EOI control the end of interrupt, the rotate mode and their combinations.
- **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.
- **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 sent to 8259A.

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

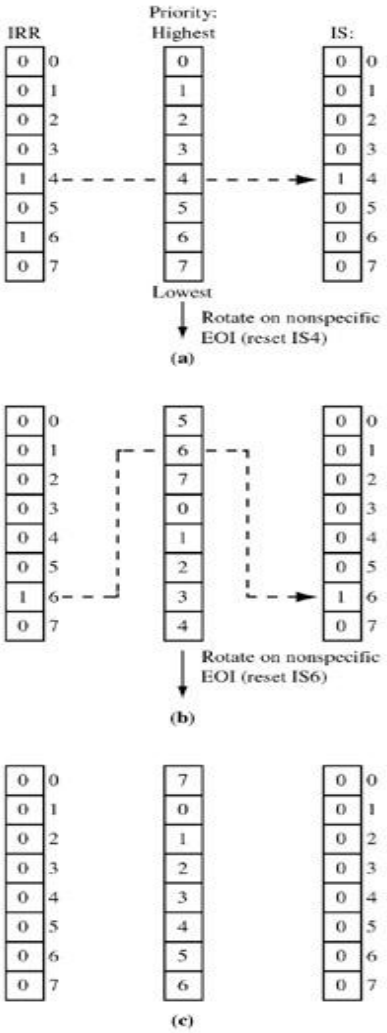
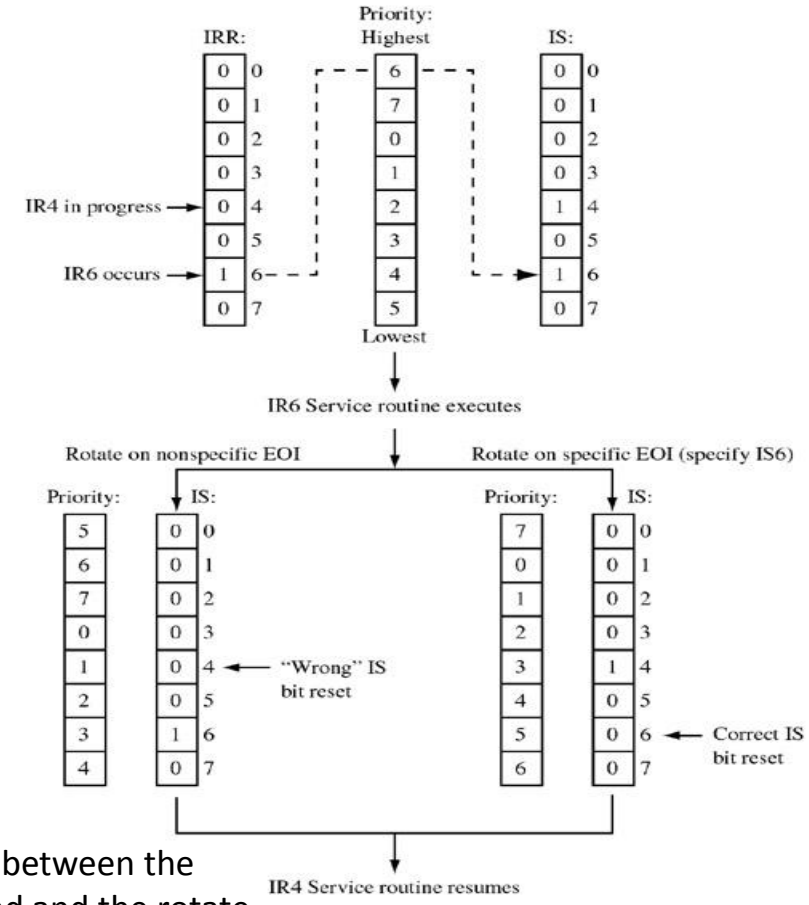


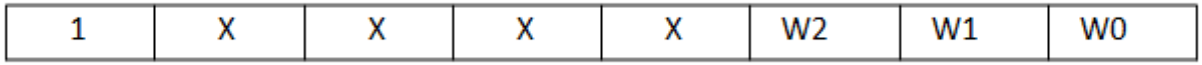
FIGURE (a) Simultaneous interrupt requests arrive on IR4 and IR6. IR4 has highest priority and its IS bit is set as the IR4 service routine is put in service. (b) The IR4 service routine issues a rotate-on-nonspecific-EOI command, resetting IS4 and assigning it lowest priority. IR6 is now placed in service. (c) The IR6 service routine issues a rotate-on-nonspecific-EOI command, resetting IS6 and assigning it lowest priority.



Example illustrating the difference between the rotate-on-nonspecific-EOI command and the rotate-on-specific-EOI command.

- **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.

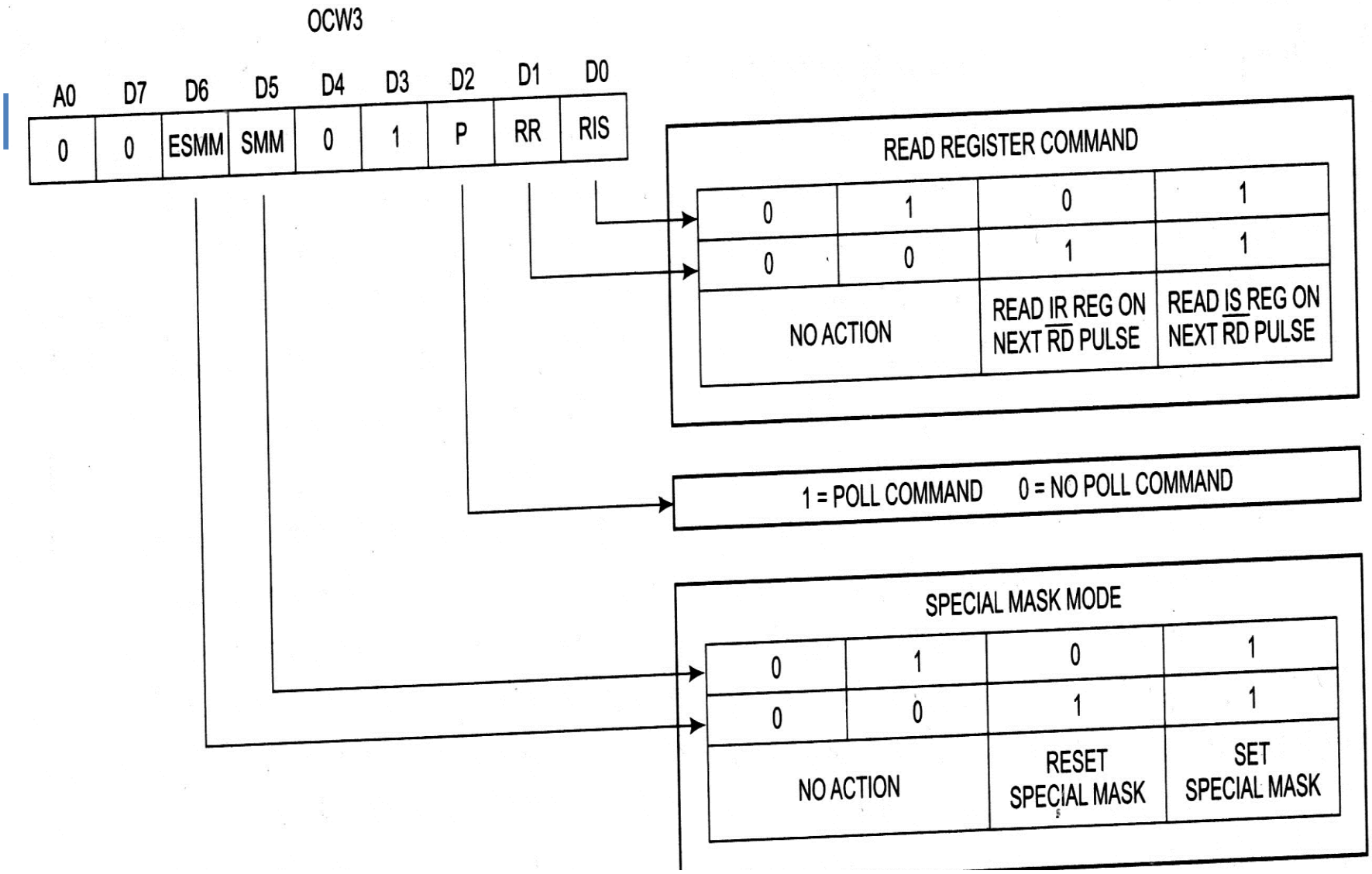
- **Poll Mode-** In this mode, the INT output of PIC is inhibited and the device is used as a prioritized poller. Performing an I/O read instruction from the PIC (either port address) returns the status word shown in fig. The rightmost three bits of the poll word indicate the active interrupt request 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.

- **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


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 REG ON NEXT RD PULSE
READ IS REG 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

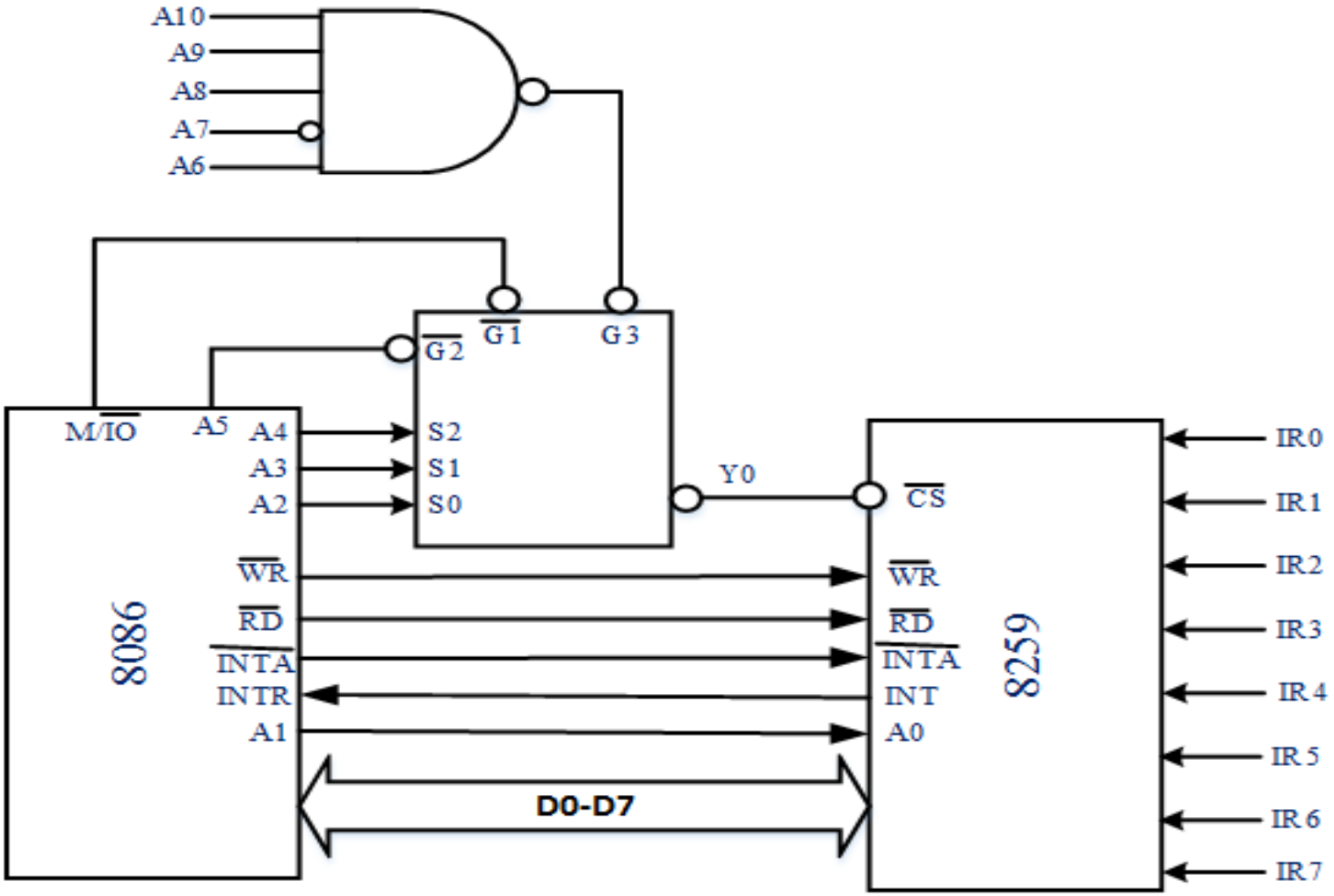
- **Problem-1:** Determine the control word ICW2 required for the PIC so that inputs IRQ0 to IRQ7 correspond to type numbers 40H to 47H. From which memory locations will the processor fetch the interrupt vectors?
  - ICW2:

0	1	0	0	0	0	0	0

- **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 .

A0	A7	A6	A5	1	LTIM	ADI	SNGL	IC4
0	0	0	0	1	0	1	1	1

- **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.


A10
A9
A8
A7
A6
G2
G1
G3
M/IO
A5
A4
A3
A2
S2
S1
S0
Y0
CS
WR
RD
INTA
INTR
A1
8086
WR
RD
INTA
INT
A0
8259
IR0
IR1
IR2
IR3
IR4
IR5
IR6
IR7
D0-D7

Address of Port
0740H (ICW1, OCW2, OCW3)
0742H (ICW2,ICW4,OCW1)

- **Step-2: Finding the ICW1=1FH**

A0	A7	A6	A5	1	LTIM	ADI	SNGL	IC4
0	0	0	0	1	1	1	1	1

Step 3:Finding ICW2=80H

Step 4: Finding the ICW4=11H

A0	D7	D6	D5	SFNM	BUF	M/S	AEOI	uPM
1	0	0	0	1	0	0	0	1

- **Step-5:** Finding OCW1=40H

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

- **Step-6: Finding the OCW2=E4H**

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

- **Step-7: Finding the OCW3**
- **For reading IRR=6AH**
- **For reading ISR=6BH**

```
INTERRUPT PROC  NEAR
MOV AL, 1FH             ; 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
```

```
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
```