

Priority Interrupt Controller 8259

1. Draw the pin diagram of PIC 8259.

Ans. The following shows the pin details of PIC 8259.

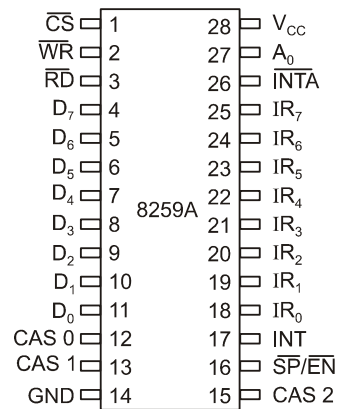


Fig. 9e.1: 8259 pin diagram (Source: Intel Corporation)

2. Draw the functional block diagram of PIC 8259.

Ans. The following shows the functional block diagram of 8259.

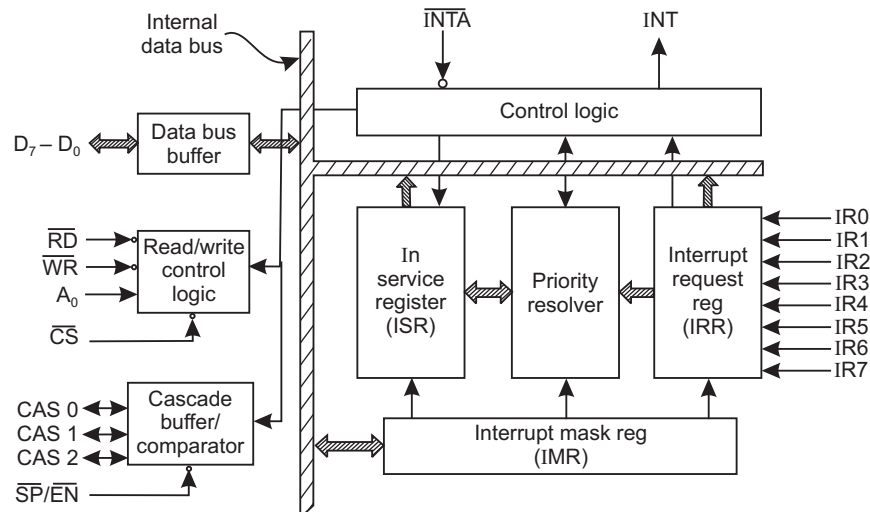


Fig. 9e.2: 8259 Functional block diagram (Source: Intel Corporation)

3. Draw the block schematic showing the interconnections between several I/O devices with PIC 8259, μ P, RAM, ROM, etc.

Ans. The block schematic of the interconnections is shown below:

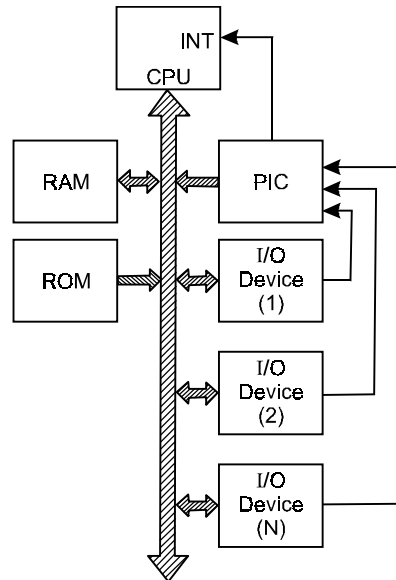


Fig. 9e.3: PIC in an interrupt-driven environment
(Source: Intel Corporation)

4. How many interrupt levels can be handled by 8259?

Ans. A single 8259 can handle up to 8 levels of interrupts.

Several 8259's can be cascaded to handle up to 64 levels of interrupts.

5. In how many interrupt modes can 8259 operate and which command word is utilised for this?

Ans. 8259 can be operated in the following categories of interrupt modes:

- (a) fully nested mode
- (b) rotating priority mode
- (c) special mask mode
- (d) polled mode

Operation Command Word (OCW) is used for 8259 to be operated in the above mentioned modes.

6. How the 8259 is programmed?

Ans. 8259 is programmed by a set of Initialisation Command Words (ICWs). Each 8259 attached to the system must be initialised through ICWs. In all there are four ICWs (ICW1 to ICW4).

7. What are the jobs performed by ICWs?

Ans. ICWs perform the following jobs:

- specifying the vectoring addresses for the individual interrupts.

- specifying single or cascaded mode of operation.
- level or edge triggering mode of operation.

8. What are the jobs performed by OCWs?

Ans. The operation command words (OCWs) are used to operate the PIC 8259 in various interrupt modes like fully nested mode, rotating priority mode, special mask mode and polled mode. The OCWs are also used for masking specific interrupts, status read operations, etc.

9. Describe how the PIC 8259 responds to interrupts?

Ans. PIC 8259 can accept a maximum of 8 interrupts from 8 different I/O devices, resolves the priority with regard to servicing the interrupts and issues an $\overline{\text{INT}}$ output signal, which is connected to INTR input pin (pin 10 of 8085). 8085, in its turn, issues an $\overline{\text{INTA}}$ signal via its pin 11, which is connected to $\overline{\text{INTA}}$ signal of 8259 (pin 26, an input pin for 8259). In response, 8259 puts out a CALL instruction code on the data bus. This is read and decoded by 8085, which then puts out two more $\overline{\text{INTA}}$ s. This is done by 8085 to read the address where the Interrupt Service Subroutine (ISS) is written. Now the PIC 8259 puts out the address of this ISS on the data bus which is eventually read by 8085 and the program jumps to the ISS address as was previously programmed by 8259.

10. What are the different functional blocks in 8259?

Ans. PIC 8259 has four different functional blocks viz.,

- (i) Interrupt and Control logic block
- (ii) Data bus buffer
- (iii) Read/Write control logic block and
- (iv) Cascade buffer/comparator section.

11. What the Interrupt and Control Logic Section consist of?

Ans. This section consists of (a) Interrupt Request Register (IRR), (b) In Service Register (ISR), (c) Priority Resolver, (d) Interrupt Mask Register (IMR), (e) Control Logic Block.

12. Write down the sequence of operations for programming 8259.

Ans. 8259 is programmed by issuing initialisation command words and operation command words. Initialisation command words are issued in a sequence. The following is the algorithm for initialising 8259.

1. Write ICW1.
2. Write ICW2.
3. If not in the cascade mode of operation, Go To Step 5.
4. Write ICW3.
5. IF ICW4 is not needed, then Go To Step 7.
6. Write ICW4.
7. Ready to accept interrupt sequence.

The flowchart for the above is shown in Fig. 9e.4.

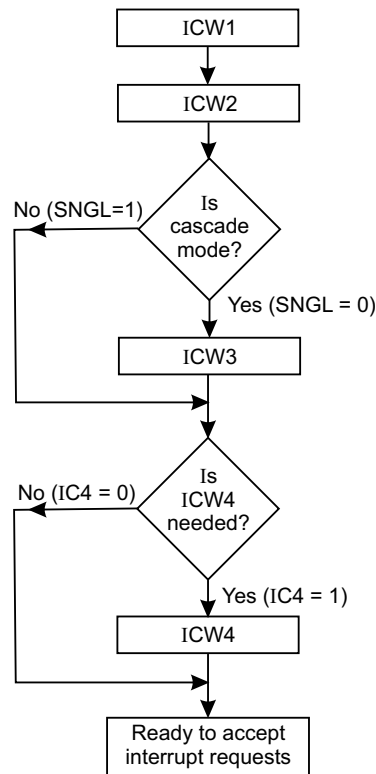


Fig. 9e.4: 8259 initialisation flow chart
(Source: Intel Corporation)

13. Write down the main features of 8259.

Ans. The main features of 8259 are as follows:

1. A single 8259 can handle 8 vectored priority interrupts.
2. 9 numbers of 8259 can be cascaded to have 64 levels of vectored priority interrupts in a μ C system.
3. The priority modes can be changed or reconfigured dynamically at any time during the main program.
4. It can be operated in various interrupt modes—fully nested, rotating priority, special mask and polled.
5. 8259 can be used with either 8080/8085 or 8086/8088 microprocessor.
6. 8259 supports both edge and level triggered mode of interrupts.
7. The CALL address can be programmed to have a spacing of either 4 or 8 memory locations.
8. The data bus is buffered.
9. The AEOI (Automatic End Of Interrupt) can be programmed.

14. Write down the functions of IR0–IR7 pins.

Ans. There are 8 interrupt input lines from external devices with IR0 having the highest and IR7 the lowest priority. These interrupt requests are acknowledged by 8259 by (a) raising the corresponding IR input (i.e., any one of IR0 to IR7) from L to H and holding it high until acknowledged or (b) just by a high level.

15. What is the function of $\overline{\text{SP/EN}}$ pin?

Ans. This is slave program/enable buffer pin of 8259. It has dual functions:

- (a) In the buffered mode it is used as an output to control the buffer transreceivers (EN). It acts as an output pin to enable the data bus buffer of the system.
- (b) In the non-buffered mode, it is used as an input pin to designate the 8259 to operate as a master (SP = 1) or slave (SP = 0).

The buffered or non-buffered mode of operation is determined at the time of initialisation of 8259 via ICW4.

16. Describe the functioning of the pins CAS0 – CAS2.

Ans. In a multiple 8259 structure, these three CAS lines form a private 8259 bus.

For a master 8259, these three pins act as output pins, but act as input pins for a slave 8259.

In a multiple 8259 structure, the master 8259 accepts interrupt requests from slave 8259. The master 8259 then generates a CALL opcode in response to the first $\overline{\text{INTA}}$. The slave 8259 provides the vectoring address. The master, via its three output pins (CAS0 – CAS2), gives out a code to identify the slave 8259. The slave 8259's, connected to the system, each accepts this code from master 8259 and compare it with the code assigned to it during initialisation. The slave, so identified, then puts out the address of the interrupt service subroutine on to DATA BUS during the second and third $\overline{\text{INTA}}$ pulses from the CPU.

The identification of this master and slave 8259's is done by the cascade buffer/comparator block.

17. Discuss the vector data formats when 8085 is interrupted via INTR.

Ans. There are eight interrupt levels (IR0 – IR7) that generate CALL to eight equally spaced locations in the memory. These eight locations can be programmed to be spaced at an interval of either 4 or 8 memory locations.

The first interrupt vector byte released in response to the first $\overline{\text{INTA}}$ is shown in Fig. 9e.5. It is equivalent to the opcode for CALL instruction.

	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
CALL CODE	1	1	0	0	1	1	0	1

Fig. 9e.5: First interrupt vector byte
(Source: Intel Corporation)

In response to the second $\overline{\text{INTA}}$, the second interrupt vector byte released is shown in figure 9e.6. which shows interval spacings of either 4 or 8.

When the interval spacing is 4, 8259 inserts D₀ – D₄ bits automatically as per the levels of interrupts (i.e., IR0 – IR7), while D₅ – D₇ bits are specified during programming of 8259 through ICW1. Again for an interval of 8 between consecutive memory locations, D₀ – D₅ bits are inserted by 8259 automatically, while D₆ – D₇ bits are specified during programming of 8259 through ICW1.

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
4	A7	A6	A5	1	0	1	0	0
5	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

Fig. 9e.6: Second interrupt vector byte
(Source: Intel Corporation)

The following figure shows the format of the byte released by 8259 in response to the third $\overline{\text{INTA}}$.

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈

Fig. 9e.7: Third interrupt vector byte
(Source: Intel Corporation)

This byte is completely programmable and is specified by ICW2.

18. Discuss the Initialisation Command Word 1 (ICW1).

Ans. Whenever a Write Command is received with $A_0 = 0$ and $D_4 = 1$, it is interpreted by 8259 to be an initialisation command word ICW1. During ICW1, the following occur:

- The edge sense circuit is reset, which means that following initialisation, an interrupt request (IR) input must make a L to H transition to generate an interrupt.
- The interrupt mask register (IMR) is cleared, i.e., all interrupts are now disabled.
- IR7 input is assigned priority 7 (lowest).
- The slave mode address is set to 7.
- The special mask mode is cleared and Status Read is set to IRR.
- If D_0 bit in ICW1 (IC_4) is set to 0, then all functions selected in ICW4 are set to zero.

The format of the byte to be followed for ICW1 is shown below:

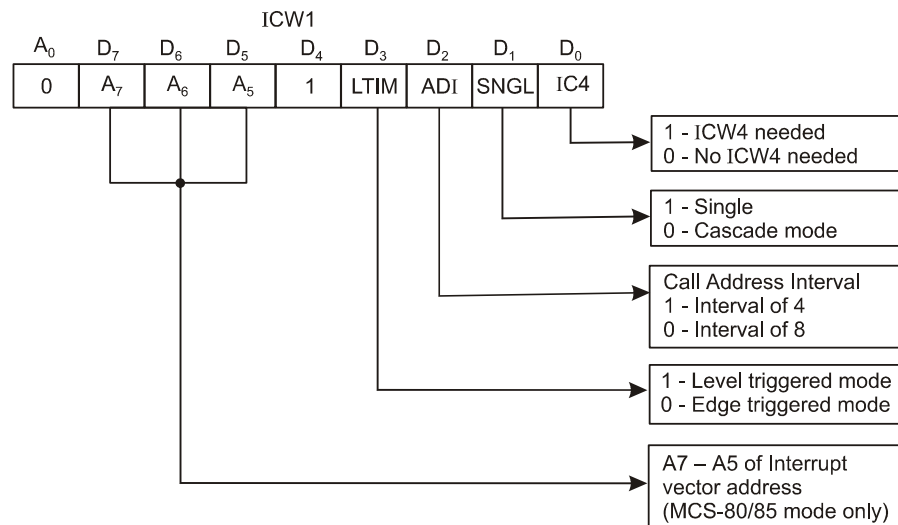


Fig. 9e.8: Initialisation command word 1 (Source: Intel Corporation)

- **Bit D₀ (IC4):** It indicates whether ICW4 is needed or not. If it is '1', then ICW4 is needed and if '0', then ICW4 is not needed.
- **Bit D₁ (SNGL):** If this bit is '0', then only one 8259 is in the system and if it is '1' then additional 8259's are there in the system.
- **Bit D₂ (ADI):** ADI stands for 'address interval'. If this bit is '0' then call address interval is 8 and if '1' then call address interval becomes 4.
- **Bit D₃ (LTIM):** This bit determines recognition of the interrupts either in level triggered or edge triggered mode. If this bit = '0', then it is edge triggered mode and if this bit is = '1' then the input interrupts will be recognised if they are in the level triggered mode.
- **D₅ - D₇:** These are A₅ - A₇ bits as shown under ICW1. For an interval spacing of 4, A₀ - A₄ bits are automatically inserted by 8259 while A₀ - A₅ are inserted automatically for an interval of 8. A₅ - A₇ bits are programmable as set by the bits D₅ - D₇ of ICW1.

19. Discuss the Initialisation Command Word 2 (ICW2).

Ans. The initialisation command word 2 i.e., ICW2 is shown below:

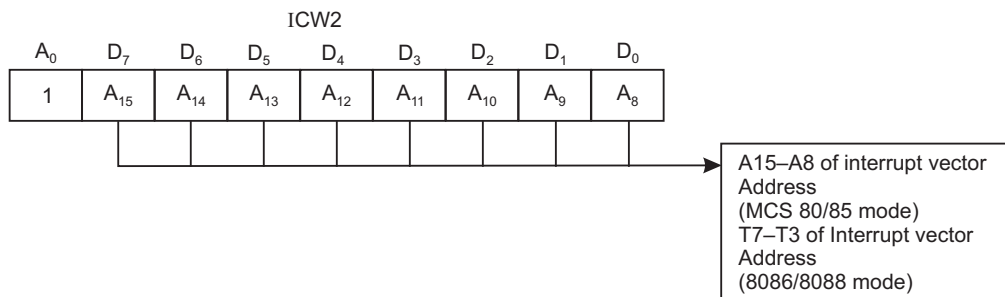


Fig. 9e.9: Initialisation command word 2 (Source: Intel Corporation)

If $A_0 = 1$ of the write command word is issued to 8259, following ICW1, it is interpreted as ICW2. ICW2 is used to load the high order byte of the interrupt vector address of all the interrupts. This byte is common for all the interrupts.

20. Discuss the Initialisation Command Word 3 (ICW3).

Ans. ICW3 can have two modes of operations: Master Mode ICW3 and Slave Mode ICW3. ICW3 is required only if several 8259's are used in the μC system in a cascaded form.

The format of the byte (both for Master Mode ICW3 and Slave Mode ICW3) are shown below:

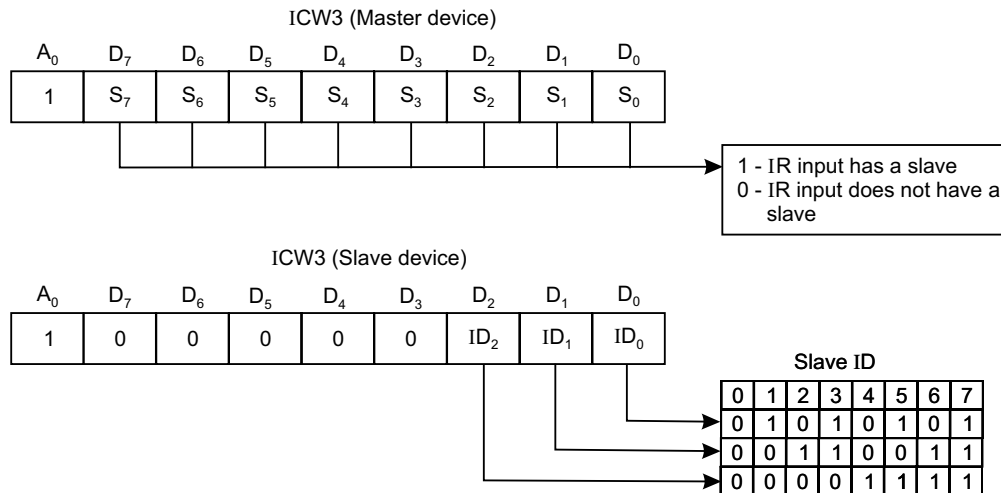


Fig. 9e.10: Initialisation command word 3
(Source: Intel Corporation)

Master Mode ICW3: For a 8259 to be treated as a master, we must have $\overline{SP}/\overline{EN}$ pin = 1 in a non-buffered environment and $M/S = 1$ in ICW4 in a buffered environment.

Then each bit in ICW3 is used to indicate to the master whether it has a slave 8259 attached to it on its corresponding interrupt request (IR) input pin. A '1' indicates the presence of a slave 8259 corresponding to that input and a '0' indicates the absence of a slave.

If now a particular slave 8259 raises its INTR output, then the master generates a CALL instruction opcode and puts out the slave identification number on its output $CAS0 - CAS2$ lines. This number goes to the slave 8259's via their $CAS0 - CAS2$ pins (these pins act as input pins for slave 8259's). Thus the number is compared with the individual slave identification number loaded during initialisation. The slave which initially placed the INTR output is thus identified and hence it releases the vector address during the second and third \overline{INTA} cycles.

Slave Mode ICW3: For a 8259 to be treated as a slave, we must have $\overline{SP}/\overline{EN}$ pin = 0 in a non-buffered environment and $M/S = 0$ in a ICW4 in a buffered environment.

Bits $D_0 - D_2$ of ICW3 (in Slave Mode) assign the slave identification code (Slave ID). The slave ID is equivalent to the master IR input to which the INTR output of the slave is connected. The slave ID compares this number with its own $CAS0 - CAS2$ inputs so as to release the address vector.

21. Describe the Initialisation Command Word 4 (ICW4).

Ans. The format of ICW4 is shown below. ICW4 is loaded only if D_0 bit of ICW1 (IC4) is set.

As shown in Fig. 9e.11, ICW4 is loaded only if D_0 bit of ICW1 (IC4) is set. The format of ICW4 is shown below.

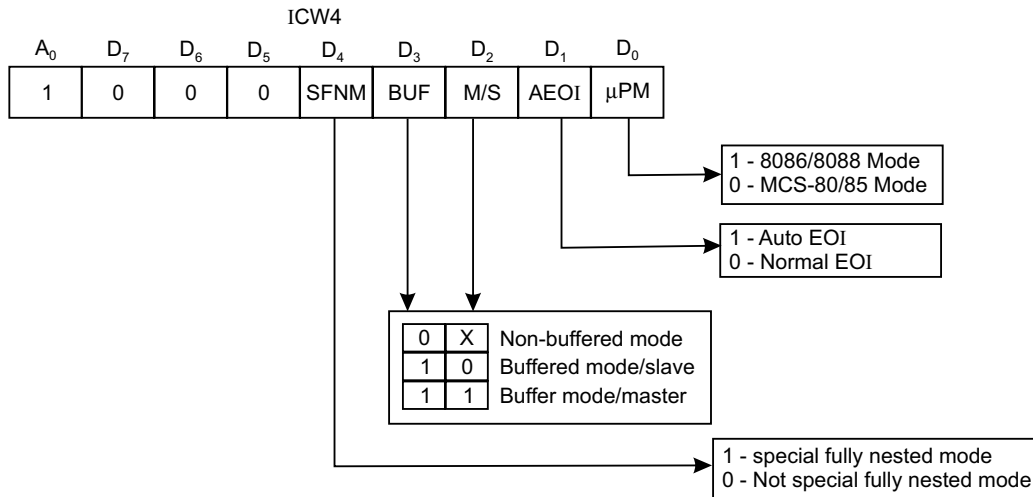


Fig. 9e.11: Initialisation command word 4
(Source: Intel Corporation)

The bit positions D_0 to D_4 are now explained:

μ PM: This corresponds to D_0 bit position and differentiates between 8086/8088 mode and MCS-80/85 mode.

AEOI: It stands for 'automatic end of interrupt'. If AEOI bit (bit D_1) = 1, then it is in auto EOI mode and if it is = 0, it is in normal EOI mode.

M/S: In the buffered mode, if M/S = 1, then 8259 is initialised as a master and if M/S = 0, then 8259 acts as a slave.

In the non-buffered mode, M/S pin has no significance. In this case, the characteristics of 8259 (i.e., whether 8259 is a master or a slave) is determined by $\overline{SP}/\overline{EN}$ pin.

BUF: This bit position (D_3 bit) determines buffered/non-buffered mode of operation. If BUF = 1, then it is buffered mode of operation and the $\overline{SP}/\overline{EN}$ pin is used as an output to enable the data bus buffer of the system.

SFNM: This stands for Special Fully Nested Mode (bit D_4). If SFNM = 1, then this mode is programmed.

22. Describe the Operation Command Words (OCWs).

Ans. There are three OCWs. These OCWs may be required to change the manner in which the interrupts are to be processed. For this to be achieved, the OCWs may be loaded any time after initialisation is over.

The format for the OCW1 is shown in Fig. 9e.12.

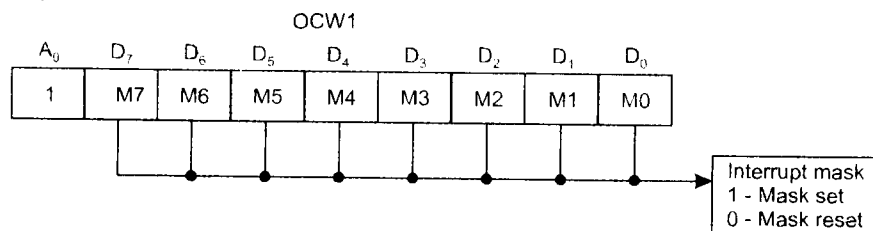


Fig. 9e.12: Format of operation command word 1
(Source: Intel Corporation)

OCW1: It is used to enable/disable a particular interrupt request by programming the Interrupt Mask Register (IMR). If $M = 1$, then the corresponding interrupt is masked and $M = 0$ indicates its unmasked condition.

A write command with $A_0 = 1$ is interpreted as OCW1, and written after ICW2.

OCW2: The format for OCW2 is shown below:

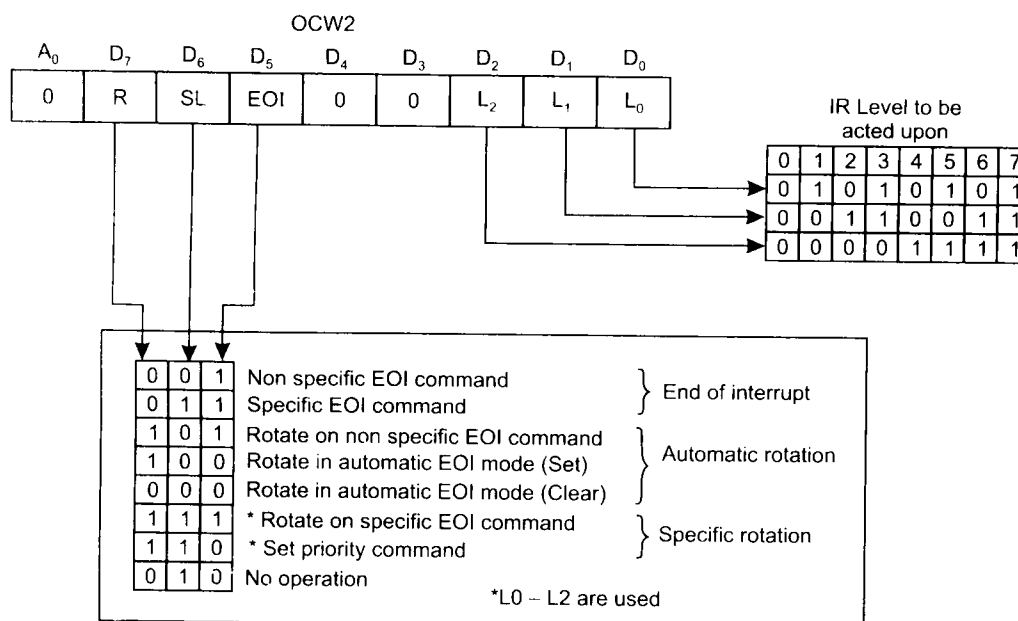


Fig. 9e.13: Format of operation command word 2
(Source: Intel Corporation)

A write command with $A_0 = 0$ and $D_4 D_3 = 00$ is interpreted as OCW2. 'R' and 'SL' stand for Rotate and Select Level respectively. The bits corresponding to R, SL and EOI control the Rotate and End of Interrupt modes. Bits $L_2 - L_0$ specify the interrupt level which is to be acted upon when SL is in active condition.

OCW3: The jobs performed by OCW3 are as follows:

(a) to read the status of registers.

(b) set/reset the Special Mask and Polled modes.

The format of OCW3 is shown Fig. 9e.14.

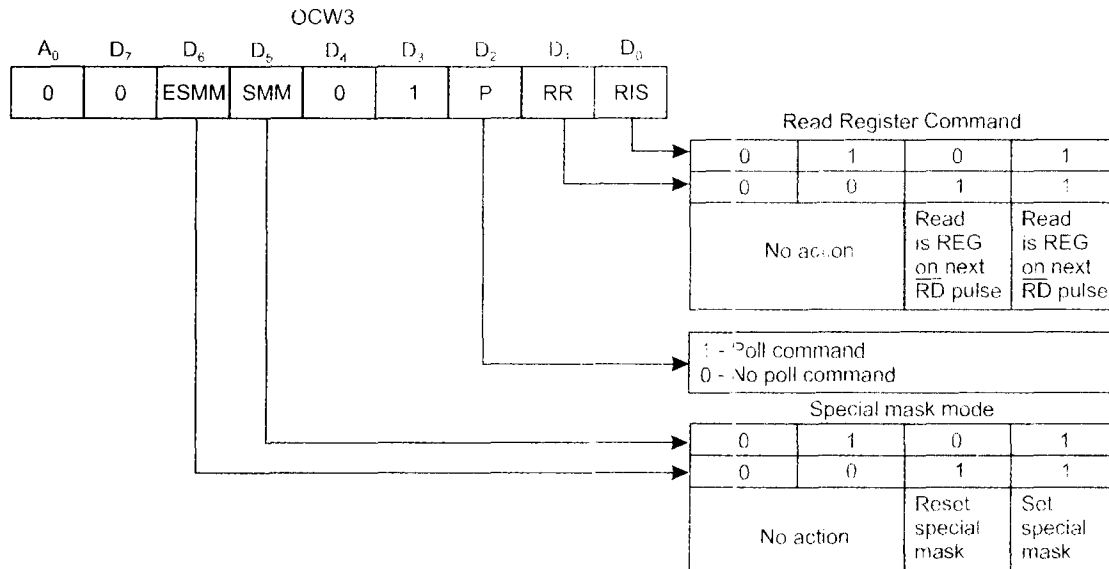


Fig. 9e.14: Format of operation command word 3
(Source: Intel Corporation)

23. Describe the FNM (Fully Nested Mode).

Ans. This mode is auto-set after initialisation is over—i.e., it is a default mode. This mode can only be changed through Operation Command Words (OCWs).

IR0 is assigned the highest priority (priority 0) and IR7 the lowest priority (priority 7). When 8259 acknowledges an interrupt request via its INTR pin, it finds out the highest priority and the corresponding bit in the In Service Register (ISR) is set.

The resetting of this bit in ISR can occur in one of the following ways:

- When the CPU issues an EOI (End of Interrupt) through OCW before coming out of ISR.
- If AEOI (Automatic End of Request) mode is set in ICW4 during initialisation, the corresponding ISR bit is automatically reset—it occurs on the trailing edge of the last (third) INTA pulse.

When a particular ISR bit is set, (a) all lower level interrupts and (b) same level interrupts remain in the inhibited condition.

But a higher level interrupt will force the 8259 to generate an INTR, but the same will be acknowledged if the Interrupt Enable F/F has already been enabled via software while the interrupt service subroutine is in progress.

24. Describe the EOI command.

Ans. This command stands for End of Interrupt (EOI).

The interrupt service bit (that is being currently serviced) can be reset by an End of Interrupt Command. This is issued by the CPU, usually just before coming out of the interrupt service routine.

There are two ways in which the EOI can be exerted—in the Fully Nested Mode (FNM) and non FNM.

In the FNM, a *non-specific* EOI command is issued by the CPU. This is an OUT instruction by the CPU to 8259. This is derived by $A_0 = 0$, $D_7 D_6 D_5 D_4 D_3 = 00100$ with $D_2 D_1 D_0$ can have any value. This is apparent from the format of the operation command word 2 i.e., OCW2. This OUT instruction is issued by CPU before exiting from the interrupt service subroutine. On receiving this instruction, 8259 resets the highest level of interrupt (i.e., the current one that which is being serviced).

In non FNM, 8259 cannot determine the last interrupt acknowledged. Thus in this case, the CPU will have to issue a *specific* EOI command signalling out the specific interrupt service bit that is to be resetted. This is done under OCW2 with $A_0 = 0$, $D_7 D_6 D_5 D_4 D_3 = 01100$ and $D_2 - D_0$ specifies the level on which the EOI command is to act.

25. In the cascade mode, how many EOI commands are to be issued?

Ans. In such a case, two EOI commands must be issued—one for the master and one for the slave.

26. Describe the AEOI command.

Ans. This mode can only be used for a master 8259 and is set by ICW4. 8259 performs a non-specific EOI on the trailing edge of the third (i.e., last) \overline{INTA} pulse.

27. Explain Special Fully Nested Mode (SFNM).

Ans. In the cascaded mode of operation, if a slave receives a higher priority interrupt request than one which is in service (through the same slave), it would not be recognised by the master. This is because the master ISR bit is already in the set condition, thereby it ignores all requests of equal or lower priority. The higher priority interrupt won't be serviced until after the master ISR bit is reset by an EOI command. This is most likely to happen after the completion of the lower priority routine.

This is where the SFNM comes into. It is meant only for the master and done during master initialisation (through ICW4). In this mode, the master will ignore interrupt requests of lower priority, but responds to requests of equal or higher priority.

The following are the differences between FNM and SFNM:

- (a) In SFNM, the slave is allowed to place an interrupt request (of higher priority than the one currently being serviced). The master recognises this higher level interrupt, which in its turn places this interrupt request to the CPU.
- (b) In SFNM, the software must determine if any other slave interrupts are pending before issuing an EOI command to the slave and then reading its ISR (In Service Register). If the ISR contains all zeroes, then no interrupt from the slave is in service and an EOI command can be sent to the master. If the ISR is not all zeros, an EOI command should not be sent to the master. Clearing the master ISR bit with an EOI command while there are still slave interrupts in service would allow lower priority interrupt to be recognised by the master.

28. Mention the types of Rotating Priority Mode of interrupt.

Ans. The Rotating Priority Mode can be set in

- (a) Automatic Rotation
- (b) Specific Rotation.

29. Discuss the Automatic Rotation.

Ans. In situations where several communicating channels are connected to μC system, all the channels should be accorded equal priority in sharing information with the μC .

Thus when a peripheral is serviced, all other equal priority peripherals should be given a chance to be serviced before the original peripheral is serviced a second time around. This is accomplished by automatically assigning a peripheral the lowest priority after being serviced. Thus a device, presently being serviced, would have to wait until all other devices are serviced.

Automatic rotation is of two types:

- (a) Rotate on non-specific EOI Command.
- (b) Rotate on automatic EOI Mode.

Rotate on non-specific EOI Command: When the rotate on NSEOI command is issued, the highest ISR bit is reset and the corresponding IR level is assigned the lowest priority.

Let IR0 has the highest and IR7 the lowest priority. Let also that IR6 and IR4 are in service with IR4 accorded the highest priority. Bit 4 in the ISR is reset when a NSEOI command is executed. After this, IR4 becomes the lowest priority and IR5 becomes the highest priority. The situations are explained in the following figure with the left side indicating the situation before the command is executed and the right side after the command execution.

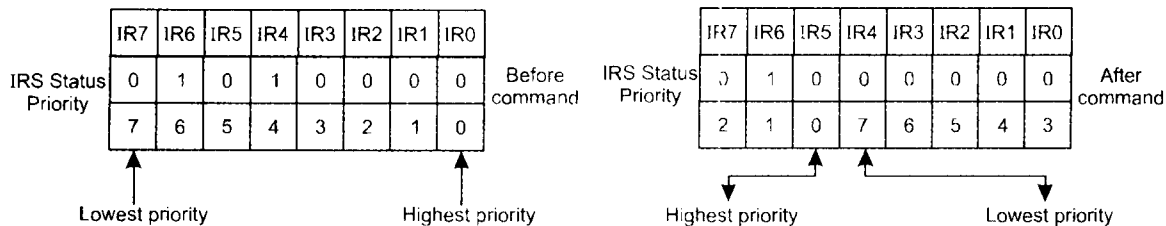


Fig. 9e.15: Rotation on non-specific EOI command

Rotate in Automatic EOI Mode: This mode works much like the rotate on NSEOI command. The main difference between the two lies in the priority routine done automatically after the last INTA pulse of an interrupt request. To enter or exit from this mode, a rotate-in-automatic EOI Set Command and rotate-in-automatic EOI Clear Command is provided.

30. Explain Specific Rotation.

Ans. In this mode, the lowest priority can be assigned to any of the IR levels (between 0 and 7) as specified by OCW2.

This mode is set by CPU by issuing an OUT instruction in the following manner:

$$A_0 = 0$$

$$D_7 D_6 D_5 D_4 D_3 = 1 1 0 0 0$$

$D_2 D_1 D_0$ bits specify which interrupt level is to be accorded lowest priority. This mode is independent of EOI command.

31. Describe Special Mask Mode.

Ans. The special mask mode enables interrupts from all levels except the level presently in service. This is done by masking the level that is in service and then issuing the special mask mode command. Once the special mask mode is set, it remains in effect until reset.

The Special Mask Mode can be set by making ESSM and SMM bits '1' in OCW3. When a mask bit is set in OCW1, all further interrupts at that level are inhibited, while interrupts on all other levels that have not been masked by OCW1 (both lower and higher) are enabled. Thus it is possible to selectively enable interrupts by programming the mask register.

The special mask mode is cleared by loading OCW3 as follows:

ESSM = 1
and SMM = 0

32. Describe the Polled Mode interrupt scheme.

Ans. In this mode, the interrupting devices seeking services from 8085 are polled one after another to detect which device has sought for interrupt request. The INT output of 8259 is either not connected to INTR input of 8085 or the interrupts are disabled by software means.

Bit D_2 (i.e., P) in OCW3 is set to '1' in the Polled mode. 8259 is then read by masking its \overline{RD} and \overline{CS} pins '0'. The ISR bit is set corresponding to the highest level interrupt in the IRR. A byte is put out on the data bus as shown below:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	–	–	–	–	W_2	W_1	W_0

Fig. 9e.16: Polled mode output word
(Source: Intel Corporation)

If $D_7 = 1$, then it implies that an interrupt needs servicing, while $D_2 - D_0$ bits (i.e., $W_2 - W_0$) correspond to the highest priority interrupt level which is requesting service.

Since INTR line is not in use in the polled mode, hence more than one 8259 may be connected in the master mode. Hence it is possible to have more than sixty four levels of interrupts in this mode.

33. On which registers status read operations can be done?

Ans. Status read operations can be done on Interrupt Request Register (IRR), In-Service Register (ISR) and Interrupt Mask Register (IMR).

34. How IRR status read operation is done?

Ans. A particular 8259 can be set up for an Interrupt Request Register (IRR) read operation by inserting in OCW3 RR (read register) = 1 and RIS (Read ISR) = 0. Following this, $\overline{RD} = 0$ and $\overline{CS} = 0$ are made on the 8259, which thereby puts out the contents of IRR on the data bus.

In the non-polled mode of 8259, A_0 should be made '0' so as to put out the contents of IRR status word after the IRR has been set for status read operation.

35. Discuss the ISR status read operation.

Ans. A particular 8259 can be set up for an In-Service Register (ISR) read operation by inserting in OCW3 RR = 1 and RIS = 1. Following this $\overline{RD} = 0$ and $\overline{CS} = 0$ are made on the 8259,

which thereby puts out the contents of ISR on the data bus.

In the non-pollled mode of 8259, A_0 should be made '0' so as to put out the contents of ISR status word after the ISR has been set for status read operation.

36. How IMR status read operation is done?

Ans. An Interrupt Mask Register (IMR) status read operation on a 8259 is done with $A_0 = 1$, $\overline{RD} = 0$, $\overline{CS} = 0$. This causes the contents of IMR to be put out on the data bus. For an IMR status read operation, OCW3 is not needed.

37. Which status read operation is performed by default after initialisation of 8259?

Ans. The default status read operation is the IRR status read, after the initialisation of 8259.

38. Mention when status read operation is not possible.

Ans. When OCW3 is set in the polled mode with $P = 1$ and $RR = 1$, status read operation is not possible.

39. Discuss the Default IR7 routine.

Ans. An interrupt via the INTR input will be treated as a valid interrupt if it remains high until after the falling edge of the first \overline{INTA} pulse. Then a valid IR7 input occurs resulting in a *normal* CALL to the IR7 routine. If it is otherwise, a *default* CALL to the IR7 routine will be generated.

A normal IR7 operation sets the ISR bit while a default IR7 operation does not do so. For this IR7 is normally used for RET instructions. If IR7 is to be utilised for other purposes, then the default IR7 operation is first to be checked. This is done by a status read operation of ISR at the beginning of interrupt service routine for IR7. If after this read operation, IR7 input = 1, then it is a valid interrupt, otherwise not.

40. Distinguish between NSEOI, SEOI and AEOI.

Ans. These three stand for non-specific end of interrupt, specific end of interrupt and automatic end of interrupt.

An NSEOI command sent from the μP lets the 8259 know when a service routine has been completed, but without any specification of its *exact* interrupt level. The 8259 determines the interrupt level (the highest priority interrupt in service) and resets the correct bit in the ISR.

The NSEOI is best suited when servicing is always at the highest priority level. When 8259 receives a NSEOI command, it simply resets the highest priority ISR bit.

The main advantage of this mode is that it is not necessary to specify the IR level. NSEOI command is not suited for the following two cases:

- (a) Using a set priority command within an interrupt service routine.
- (b) Using a Special Mask Mode.

A SEOI command sent from μP to 8259 lets it know when a service routine of a *particular* interrupt level is completed. A SEOI command resets a specific (particular) ISR bit—thus any one of the eight IR levels can be specified.

A SEOI command is needed when 8259 is unable to determine the IR level.

A SEOI command is best suited for situations in which priorities of the interrupt levels are changed during an interrupt routine (Specific Rotation).

The AEOI mode scores over the EOI modes in that no command has to be issued in

AEIOI mode. Thus AEIOI mode greatly simplifies programming and lowers code requirements within interrupt routines.

AEIOI mode should be used continuously because the ISR bit of a routine presently in service is reset right after its acknowledgement. It thus leaves behind nothing in the ISR about which particular bit is being serviced. If any interrupt request occurs during this time, it will be serviced (provided all interrupts are enabled) regardless of its priority—whether low or high.

Another peculiar problem—called ‘over nesting’ may happen in this case. It occurs when an IR input keeps as interrupting its own routine. It results in unnecessary stack pushes which could fill up the entire stack in a worst case situation.

41. Describe how several PICs are cascaded together.

Ans. Several PICs can be cascaded together—in all 9—with one PIC acting as the master and the rest eight as slaves and shown in Fig. 9e.17. \overline{SP} pin of the master is connected to V_{cc} , whereas for slave PICs, their \overline{SP} pins are connected to ground. The INT outputs of the slave PICs are connected to one of IR0-IR7 pins of the master. The registers within the PICs are allocated separate addresses by using separate \overline{CS} signals. Initialisation of each of the PICs are done separately.

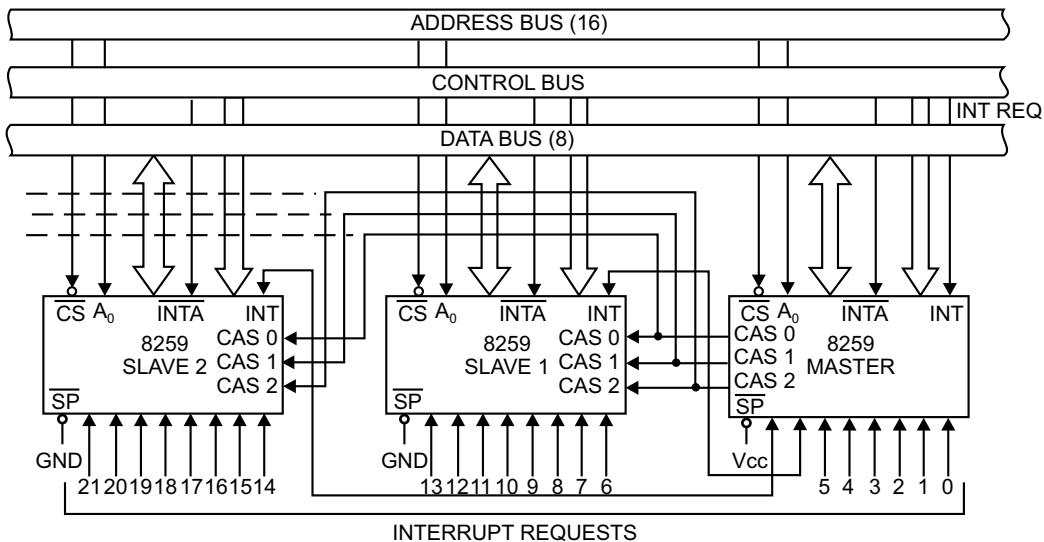


Fig. 9e. 17: Cascading of 8259 PICs

When an interrupt comes, it activates one of the IR input lines of a slave PIC and this in turn activates one of the IR lines of the master (via the INT output pin of the slave PIC which has been interrupted). This in turn interrupts the CPU. The INTA output from the master enables the corresponding CAS0-CAS2 lines of the slave PIC—this releases the vector address of the data bus in the second and third INTA cycles.