In the Fully Nested Mode, the highest level in the ISR would necessarily correspond to the last interrupt acknowledged and serviced. In such a case, a non-specific EOI command may be issued by the CPU.

However, if the FNM is not used, the 8259 may not be able to determine the last interrupt acknowledged. In such a case, a specific EOI command will have to be issued by the CPU.

It should be noted that in the cascade mode, the EOI command must be issued twice, once for the master and once for the slave.

Automatic End Of Interrupt (AEOI)

If the AEOI mode is set, the 8259 will perform a non-specific EOI on its own on the trailing edge of the third $\overline{\text{INTA}}$ pulse. The AEOI mode can only be used for a master 8259 and not for a slave.

b) Special Fully Nested Mode (SFNM) :

In the FNM, on the acknowledgement of an interrupt, further interrupts from the same level are disabled. However, in large systems which use cascaded 8259s and where the interrupt levels within each slave have to be considered. An interrupt input to a slave, in turn causes the slave to place an interrupt request to the master on one of the master's inputs. Further interrupts to the slave will cause the slave to place requests to the master on the same input to the master, but these will not be recognised because further interrupts on the same input level are disabled by the master.

The Special Fully Nested Mode (SFNM) is used to avoid this problem. The SFNM is set up ICW4 during initialisation. It is similar to the FNM except for the following differences:

(i) When an interrupt request from a slave is being serviced, the slave is allowed to place further requests (these requests are of a higher priority than the request currently being serviced). These interrupts are recognised by the master and it initiates interrupt requests to the CPU.

(ii) Before exiting from the interrupt service routine, a non-specific EOI must be sent to the slave and its ISR must be read to determine if it was the only interrupt to the slave. If the ISR is empty, a non-specific EOI command can be sent to the master. If it is not empty, it implies that the same IR level input to the master is to be serviced again due to more than one interrupt being presented to the slave, and an EOI must not be sent to the master.

c) Rotating Priority Mode :

The Rotating priority mode can be set in (i) Automatic rotation, and (ii) Specific rotation.

(i) Automatic Rotation

In this mode, a device, after being serviced, receives the lowest priority. Assuming that IR_3 has just been serviced, it will receive the seventh priority.

IR_0	IR_1	IR_2	IR_3	IR_4	IR_5	IR_6	IR_7
4	5	6	7	0	1	2	3

(ii) Specific Rotation

In the Automatic Rotation mode, the interrupt request last serviced is assigned the lowest priority, whereas in the Specific Rotation mode, the lowest priority can be assigned to any interrupt input (IR_0 to IR_7) thus fixes all other priorities.

For example if the lowest priority is assigned to IR_2, other priorities are as shown below.

IR_0	IR_1	IR_2	IR_3	IR_4	IR_5	IR_6	IR_7
5	6	7	0	1	2	3	4

d) Special Mask Mode :

If any interrupt is in service then the corresponding bit is set in ISR and the lower priority interrupts are inhibited. Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control, for example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion. In these cases we have to go for special mask mode.

In the special mask mode it inhibits further interrupts at that level and enables interrupts from all other levels (lower as well as higher) that are not masked. Thus any interrupt may be selectively enabled by loading the mask register.

e) Poll Mode :

In this mode the INT output is not used. The microprocessor checks the status of interrupt requests by issuing poll command. The microprocessor reads contents of 8259A after issuing poll command. During this read operation the 8259A provides polled word and sets ISR bit of highest priority active interrupt request FORMAT.

I	X	X	X	X	W_2	W_1	W_0

$I = 1$	$\rightarrow$	One or more interrupt requests activated.
$I = 0$	$\rightarrow$	No interrupt request activated.
$W_2\ W_1\ W_0$	$\rightarrow$	Binary code of highest priority active interrupt request.

6.5.5 Programming the 8259A

The 8259A requires two types of command words. Initialization Command Words (ICWs) and Operational Command Words (OCWs).

The 8259A can be initialized with four ICWs; the first two are compulsory, and the other two are optional based on the modes being used. These words must be issued in a given sequence. After initialization, the 8259A can be set up to operate in various modes by using three different OCWs; however, they no longer need to be issued in a specific sequence.

Flow chart :

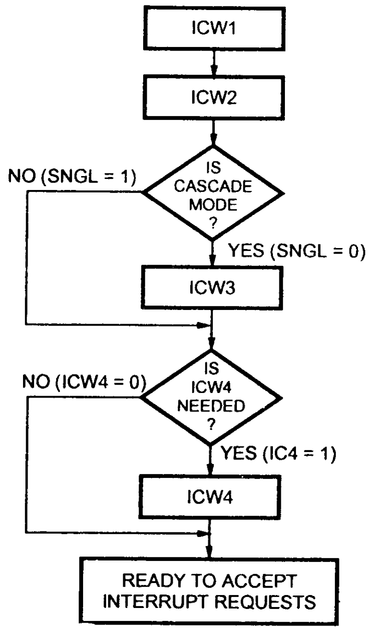

Fig. 6.7 8259 A initialization flowchart

Initialization Command Word 1 (ICW1)

Fig. 6.8 shows the Initialization Command Word 1 (ICW1).

A write command issued to the 8259 with $A_0 = 0$ and $D_4 = 1$ is interpreted as ICW1, which starts the initialization sequence.

It specifies

1. Single or multiple 8259As in the system.
2. 4 or 8-bit interval between the interrupt vector locations.
3. The address bits A_7 - A_5 of the CALL instruction.
4. Edge triggered or level triggered interrupts.
5. ICW4 is needed or not.

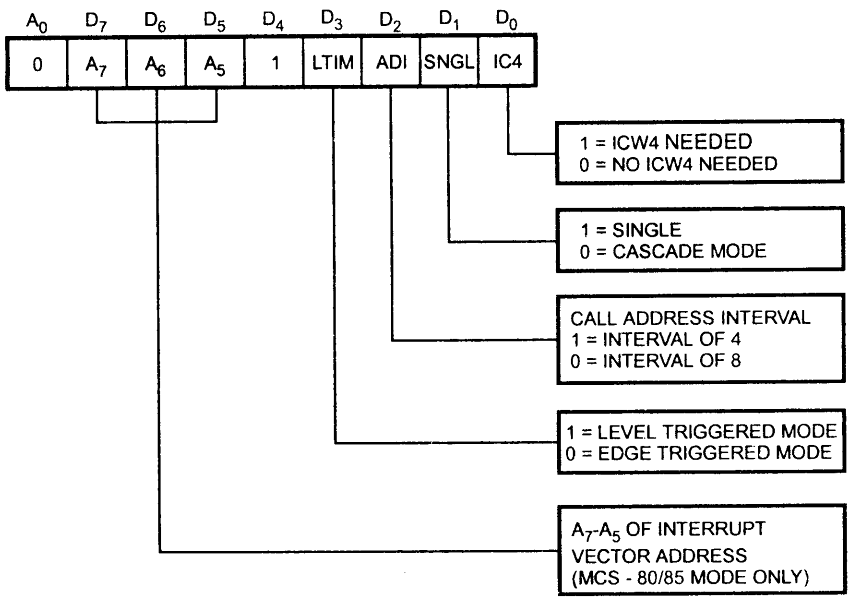

Fig. 6.8 Initialization command word 1 (ICW1)

Initialization Command Word 2 (ICW2)

Fig 6.9 shows the Initialization Command Word 2 (ICW2).

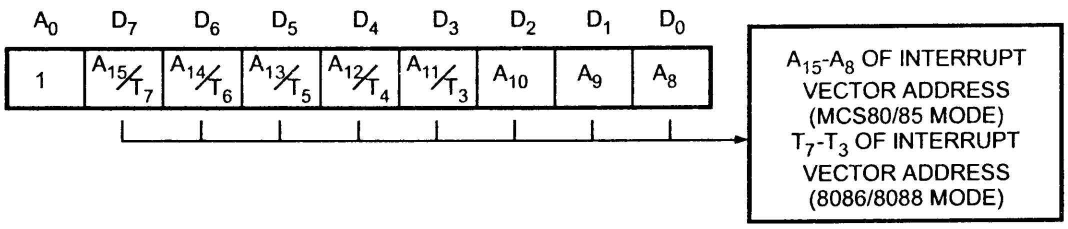
A_0 D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
1 A_{15}/T_7 A_{14}/T_6 A_{13}/T_5 A_{12}/T_4 A_{11}/T_3 A_{10} A_9 A_8
A_{15}-A_8 OF INTERRUPT VECTOR ADDRESS (MCS80/85 MODE) T_7-T_3 OF INTERRUPT VECTOR ADDRESS (8086/8088 MODE)

Fig. 6.9 Initialization command word 2 (ICW2)

A write command following ICW1, with A0 = 1 is interpreted as ICW2. This is used to load the high order byte of the interrupt vector address of all the interrupts.

Initialization Command Word 3 (ICW3)

ICW3 is required only if there is more than one 8259 in the system and if they are cascaded. An ICW3 operation loads a slave register in the 8259. The format of the byte to be loaded as an ICW3 for a master 8259 or a slave is shown in the Fig. 6.10. For master, each bit in ICW3 is used to specify whether it has a slave 8259 attached to it on its corresponding IR (Interrupt Request) input. For slave, bits D_0-D_2 of ICW3 are used to assign a slave identification code (slave ID) to the 8259.