97

CHAPTER 18

8259 (Programmable Interrupt Controller)

18.1 Polling and Interrupts

- Whenever more than one I/O devices are connected to a microprocessor based system, any one of the I/O devices may ask service at any time. There are two methods in which the microprocessor can service these I/O devices. One method is to use the **polling routine**, while the other method employs **interrupt**.
- In the polling routine the microprocessor checks whether any of the I/O devices is requesting for service.
- The polling routine is a simple program that keeps a check for the occurrences of interrupt. For e.g. : Let us assume that our polling routine is servicing I/O ports 1, 2, 3,............8. The polling routine will check the status of the I/O ports in a proper sequence.
- The polling routine will first transfer the status of the I/O port 1 to the accumulator. It then checks the contents of accumulator to determine if the service request bit is set. If the bit is set then I/O port 1 service routine is called, otherwise the polling routine will move forward to check if port 2 is requesting service. On completion of the service to port1, the polling routine will test port2. The process is repeated till all the 8 ports are tested and all the I/O ports those are demanding service are processed. On completion of the polling routine, the microprocessor will resume with the execution of the program. Fig. 18.1.1 shows the sequence for polling routine.
- The polling routine has priorities assigned to the different I/O devices. Once the routine begins port1 will always be checked first, then port2 and so on.
- Another way that allows the microprocessor stop with the execution of the program and give service to the I/O devices is **interrupt**. It is an external asynchronous input that informs the microprocessor to complete the instruction that it is currently executing and fetch a new routine in order to offer service to the I/O device. Once the I/O device is serviced, the microprocessor will continue with the execution of its normal program.

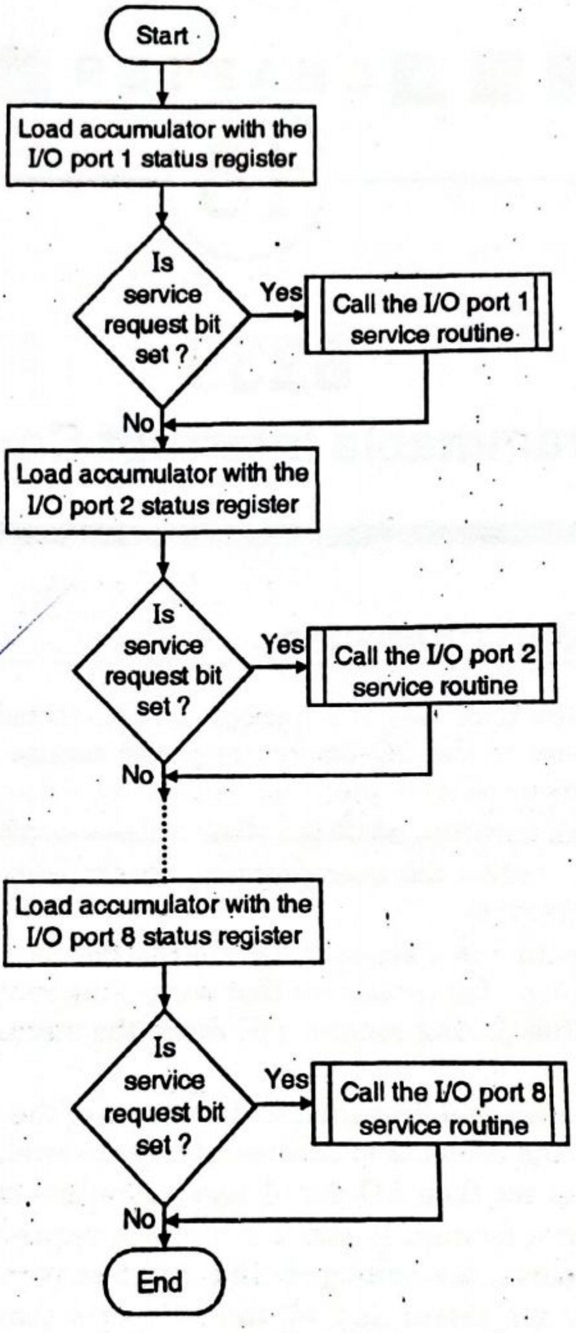

Start
Load accumulator with the I/O port 1 status register
Is service request bit set ?
Yes
Call the I/O port 1 service routine
No
Load accumulator with the I/O port 2 status register
Is service request bit set ?
Yes
Call the I/O port 2 service routine
No
Load accumulator with the I/O port 8 status register
Is service request bit set ?
Yes
Call the I/O port 8 service routine
No
End

Fig 18.1.1 : Polling sequence

18.2 8259A Programmable Interrupt Controller

- For applications where we require multiple interrupt sources, we need to use an external device called as a priority interrupt controller (PIC).
- By connecting a PIC to the microprocessor we can increase the interrupt handling capacity of the microprocessor.
- 8259A is the commonly used priority interrupt controller.

18.2.1 Features of 8259A

- It is a LSI chip which manages 8 levels of interrupts i.e. it is used to implement 8 level interrupt system.

- It can be cascaded in a master slave configuration to handle upto 64 levels of interrupts.
- It can identify the interrupting device.
- It can resolve the priority of interrupt requests i.e. it, does not require any external priority resolver.
- It can be operated in various priority modes such as fixed priority and rotating priority.
- The interrupt requests are individually maskable.
- The operating modes and masks may be dynamically changed by the software at any time during execution of programs.
- It accepts requests from the peripherals, determines priority of incoming request, checks whether the incoming request has a higher priority value than the level currently being serviced and issues an interrupt signal to the microprocessor.
- It provides 8 bit vector number as an interrupt information.
- It does not require clock signal.
- It can be used in polled as well as interrupt modes.
- The starting address of vector number is programmable.
- It can be used in buffered mode (Buffered mode is applicable for multiprocessor system).

18.3 8259 Block Diagram

Q. Explain 8259 with functional block diagram.

The block diagram of 8259 is as shown in Fig. 18.3.1. It contains following blocks :

• Data bus buffer	• Read/write logic
• Cascade buffer and comparator	• Control logic
• IRR (Interrupt Request Register)	• InSR (In-Service Register)
• Priority resolver	• IMR (Interrupt Mask Register)

(1) ***Data bus buffer*** : It is used to transfer data between microprocessor and internal bus.

(2) ***Read/Write control logic*** : It sets the direction of data bus buffer. It controls all internal read/ write operations. It contains initialization and operation command registers.

(3) ***Cascaded buffer and comparator*** : In master mode, it functions as a cascaded buffer. The cascaded buffers outputs slave identification number on cascade lines. In slave mode, it functions as a comparator. The comparator reads slave identification number from cascade lines and compares this number with its internal identification number. In buffered mode it generates an $\overline{EN}$ signal.

(4) ***Control logic*** : It generates an INT signal. In response to an $\overline{INTA}$ signal, it releases three byte CALL address or one byte Vector number. It controls read/write control logic, cascade buffer/comparator, in service register, priority resolver and IRR.

Interrupt request register (IRR) :

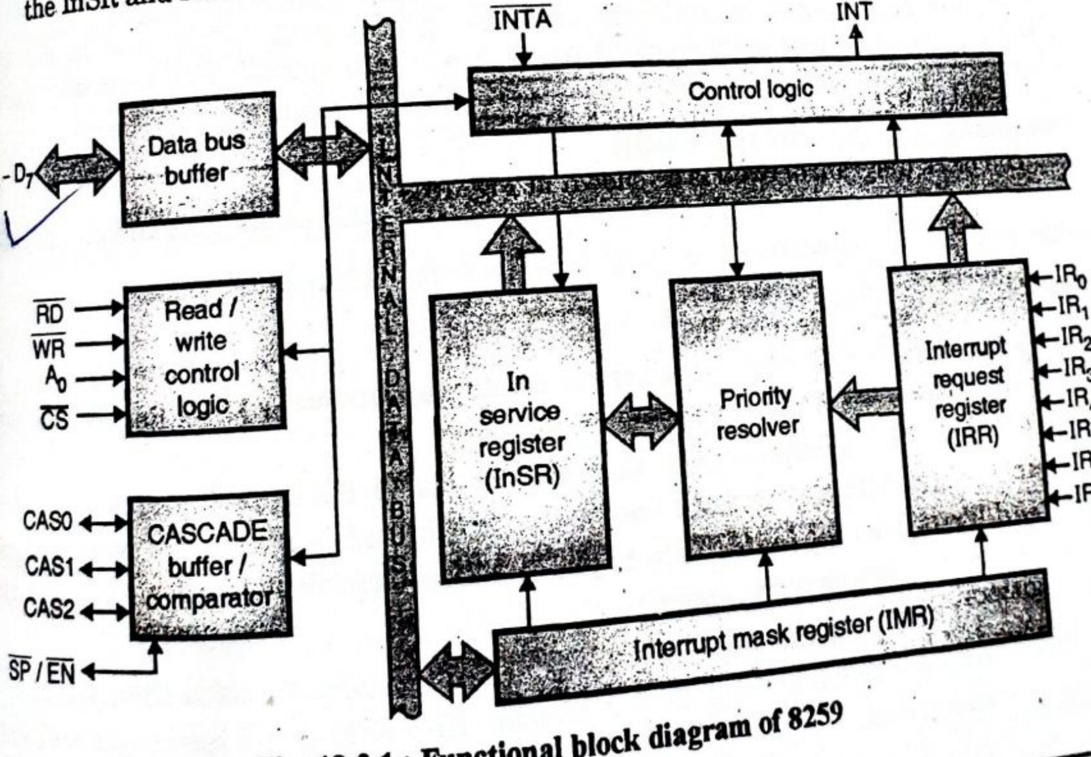
Q. Explain how will you read the IRR.

It is used to store all pending interrupt requests. Each bit of this register is set at the rising edge or at the high level of the corresponding interrupt request line. The microprocessor can read contents of this register by issuing appropriate, command word.

In service register (InSR) :

It is used to store all interrupt levels currently being serviced. Each bit of this register is set by priority resolver and reset by End of interrupt command word. The microprocessor can read contents of this register by issuing appropriate command word.

(7) *Priority resolver* : It determines the priorities of the bit set in the IRR. To make decision, the priority resolver looks at the ISR. If the higher priority bit in the InSR is set then it ignores the new request. If the priority resolvers finds that the new interrupt has a higher priority than the highest priority interrupt currently being serviced and the new interrupt is not in service, then it will set appropriate bit in the InSR and send the INT signal to the microprocessor for new interrupt request.



INTA
INT
Control logic
D0 - D7
Data bus buffer
RD
WR
A0
CS
Read / write control logic
INTERNAL DATA BUS
In service register (InSR)
Priority resolver
Interrupt request register (IRR)
IR0
IR1
IR2
IR3
IR4
IR5
IR6
IR7
CAS0
CAS1
CAS2
CASCADE buffer / comparator
SP / EN
Interrupt mask register (IMR)

Fig. 18.3.1 : Functional block diagram of 8259

(8) *Interrupt mask register (IMR)* :

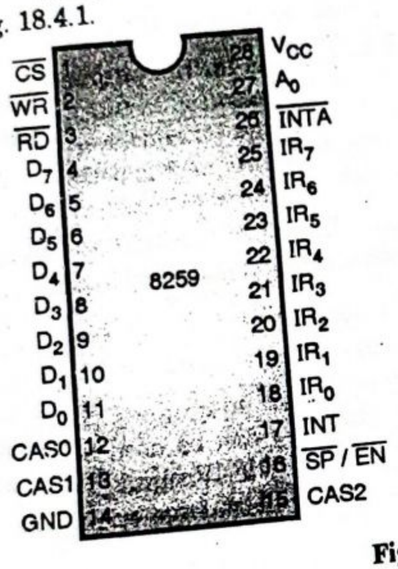
Q. Explain how will you read the IMR.

It is a programmable register.It is used to Mask unwanted interrupt request, by writing appropriate command word. The microprocessor can read contents of this register without issuing any command word.

18.4 Pin Configuration of 8259

Q. Explain in detail 8259 with pin configuration.

The pin configuration of 8259 programmable interrupt controller is as shown in Fig. 18.4.1.



$D_0 - D_7$	I/O	Data bus
$\overline{RD}$	I	I/O read
$\overline{WR}$	I	I/O write
$\overline{CS}$	I	Chip select
INT	O	Interrupt request (To 8085)
$\overline{INTA}$	I	Interrupt acknowledge (From 8085)
A_0	I	Address line
$IR_0 - IR_7$	I	Interrupt request (To 8259)
$\overline{SP} / \overline{EN}$	I/O	Slave program / Enable buffer
CAS0 – CAS2	I/O	Cascade lines

Fig. 18.4.1 : Pin diagram of 8259

Symbol	Description
(1) $D_0 - D_7$	These are bi-directional, tristate, buffered, non multiplexed data lines. These lines are connected to the system data lines directly (in non-buffered mode) or through data buffers (in buffered mode).
(2) $\overline{RD}$ *(Read)*	It is an active low control input line. It is used to read contents of internal registers. It is connected to $\overline{\textbf{IOR}}$ or $\overline{\textbf{MEMR}}$ of the system bus.
(3) $\overline{WR}$ *(Write)*	It is an active low control input line. It is used to write data into registers. It is connected to $\overline{\textbf{IOWR}}$ or $\overline{\textbf{MEMWR}}$ of the system bus.
(4) A_0	It is an address input line. It is used to select appropriate control register. It is connected to one of the address lines of the system address bus.
(5) $\overline{CS}$	It is an active low chip select input line. It is used to select 8259 A chip. This signal is generated by address decoder.
(6) CAS0 – CAS2	These are bi-directional 3 bit cascade lines. These lines are used in cascade mode only. In master mode these lines function as output lines. In this mode, the PIC places 3 bit slave identification number on cascade lines. In slave mode, these lines function as input lines. In this mode, the PIC reads 3 bit slave identification number from master PIC via cascade lines.

Symbol	Description
(7) $\overline{SP}$ / $\overline{EN}$ *(Slave Program / Enable)*	It is an active low bi-directional control line. In non buffered mode, it functions as $\overline{\textbf{SP}}$ input line. In this mode, $\overline{\textbf{SP}}$ is used to distinguish between master and slave PIC's. i.e. $\overline{\textbf{SP}}$ pin of master PIC is connected to V_{CC} while $\overline{\textbf{SP}}$ pin of slave PIC's is grounded. In buffered mode it functions as an $\overline{\textbf{EN}}$ output line. In this mode it is used to enable data buffers.
(8) *INT*	It is an interrupt output line. It goes high whenever a valid interrupt (unmasked and highest priority) request is activated. It must be connected to INTR input of the microprocessor.
(9) $\overline{INTA}$	It is an interrupt acknowledge input line. This signal is generated by the microprocessor. The 8259A accepts two $\overline{\textbf{INTA}}$ pulses to release one byte vector number. The 8259 A does not work without $\overline{\textbf{INTA}}$ pulses in vectored mode.
(10) $IR_0 - IR_7$	These are asynchronous interrupt request input lines. These signals are generated by peripherals. They can be used either in edge triggered or level triggered mode. The IR input should make low to high transition in level as well as edge triggered mode.

18.5 Priority Modes

Q. Explain various priority modes of 8259 A.

The various priority modes of 8259A are :

- Fully nested mode.
- Special fully nested mode.
- Rotating priority mode.
- Special masked mode.

18.5.1 Fully nested mode (FNM)

- After initialization, the 8259A operates in fully nested mode i.e. it is default priority mode. It continues to operate in this mode until the mode is changed through OCWs. It is also called as default mode.
- In this mode, IR_0 has highest priority and IR_7 has lowest priority. When the interrupt is acknowledged, it sets the corresponding bit of ISR.
- This bit will prevent all interrupts of the same or lower level, however it will accept higher priority interrupt requests.
- The bit in the InSR will remain set until an EOI (End of Interrupt) command is issued by the microprocessor at the end of ISR (Interrupt Service Routine).
- If the AEOI (Automatic End of Interrupt) bit is set, the bit in the InSR resets at the trailing edge of last $\overline{\text{INTA}}$

End of Interrupt (EOI)

Q. Explain the importance of EOI command of 8259.

(1) The InSR bit can be reset by on EOI command that is issued by the microprocessor before existing from the interrupt routine.

(2) In the FNM, the highest level in the InSR would correspond to the last interrupt that is acknowledged and serviced in such a case, a non-specific EOI command can be issued

(3) If the fully nested mode is not used, the 8259 PIC may not be able to determine the last interrupt that is acknowledged. In such a case a specific EOI command needs to be issued.

(4) In the cascade mode, the EOI command should be issued twice once for master and once for salve.

(A) *Nonspecific EOI command* :

This command informs the 8259A that the current interrupt service routine has been completed. It resets current bit (highest priority bit) of the InSR. This command is independent of the interrupt level and is thus called a nonspecific EOI. It must be used in fully nested mode.

(B) *Specific EOI command* :

If the fully nested mode is not used, the 8259A may not be able to tell which interrupt was just acknowledged. In such a case a specific EOI command must be issued. This command resets a bit of InSR, which is specified by $L_2L_1L_0$.

Automatic end of interrupt (AEOI)

In this mode the 8259 will perform a non-specific EOI on its own and on the trailing edge third $\overline{\text{INTA}}$ pulse. It can be used only for master and not for salve.

18.5.2 Special Fully Nested Mode (SFNM)

- In the fully nested mode, on the acknowledgement of an interrupt, the further interrupts of the same level are disabled.
- Consider a large system that uses cascaded 8259s and where the interrupt levels within each salve have to be considered. An interrupt request input to the salve causes the salve to place an interrupt request to the master and on one of the master's inputs.
- Interrupts to the same salve will cause the salve to place the request to the master on same input to the master. However, these interrupts will not be recognized.
- This is because further interrupts to the same level are disabled by the master as its InSR bit is set.
- The special fully nested mode is used to prevent the problem.

18.5.2(A) Difference between Special Fully Nested Mode And Fully Nested Mode

(1) Whenever an interrupt request form a salve is being serviced, the slave is allowed to place further requests and if these requests are of a higher priority than the request that is currently being serviced. These interrupts are recognized by the master. It initializes interrupt requests to the microprocessor unit.

(2) Before termination form the ISR, a non-specific EOI must be sent to the slave. Its InSR must be read to determine if it was the only interrupt to the slave. If the InSR is empty, a non-specific EOI command can be sent to the master. If the InSR is not empty, then the same IR level input to the master can be reserviced. This is because there are multiple interrupts on the slave EOI must not be sent to the master.

18.5.3 Rotating Priority Mode

Q. Explain rotating priority mode in case of 8259 PIC.

The rotating priority mode can be set as :

- Automatic rotation
- Specific rotation

18.5.3(A) Automatic Rotation

- In this mode, a device after being serviced becomes the lowest priority, and consecutive next interrupt becomes highest priority.
- The device that has been just serviced will receive the seventh priority. Here IR_4 has just been serviced, hence it becomes lowest priority and IR5 becomes highest priority.

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

18.5.3(B) Specific Rotation

- In the automatic rotation mode, the interrupt request that is serviced last is assigned the lowest priority.
- In the specific rotation mode, lowest priority can be assigned to any interrupt input (IR_0 to IR_7) by a specific rotation command.
- e.g. if lowest priority is assigned to IR_3, all other interrupt priorities are shown.

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

18.5.4 Special Masked Mode

- If an interrupt is in service, then the corresponding bit in the InSR is set and the lower priority interrupts are inhibited.
- Sometimes it is desired for the interrupt service routine to dynamically after the systems interrupt priority structure. In such cases we have to used special masked mode.
- Special masked mode inhibits further interrupts at that level and enables interrupts form all other levels that are not masked. Thus, as interrupt can be enabled by loading the mask register.

18.5.5 Operating Modes

8259 has two operating modes viz. interrupt driven and polling mode. In interrupt driven mode, 8259 interrupts the processor with the INT pin whenever it gets an interrupt.

Polled mode

Q. Explain POLL mode of 8259.

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

I	X	X	X	X	X	W_2	W_1	W_0

I = 1 One or more interrupt requests activated
I = 0 No interrupt request activated

Binary code of highest priority active interrupt request

18.6 Programming the 8259A

Q. Explain initialisation command words of 8259 IC.

The 8259 can be programmed through a sequence of simple I/O operations. It accepts two types of command words. They are :

- Initialization command words (ICWs)
- Operation command words (OCWs)