Chapter 6. Basic I/O, Memory R/W and Interrupt operation

6.1. Direct Memory Access (DMA)

This is a process where data is transferred between two peripherals directly without the involvement of the microprocessor. Removing the CPU from the path and letting the peripherals device manage the memory bus directly would improve the speed of transfer. Hence, it is a faster scheme and hence used for high speed printers. During DMA transfer, the CPU is idle and no control of memory bus. Data transfer from I/O device to memory or vice-versa is controlled by a DMA controller (e.g. Intel 8237 DMAC) and during DMA transfer it takes over the buses to manage the transfer directly between IO device and memory. Following steps shows the DMA operation in 8085.

• The external DMA controller sends a signal on the HOLD pin to the microprocessor.

• The microprocessor completes the current operation and sends a signal on HLDA and stops using the buses.

• Once the DMA controller is done, it turns off the HOLD signal and the microprocessor takes back control of the buses.

Fig. 6-1 shows how DMA controller operates in microprocessor system.

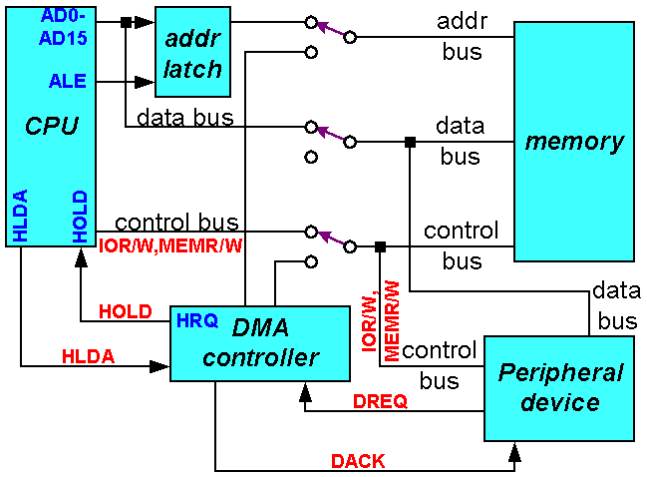

Figure 6-1: DMA controller operating in a microprocessor system

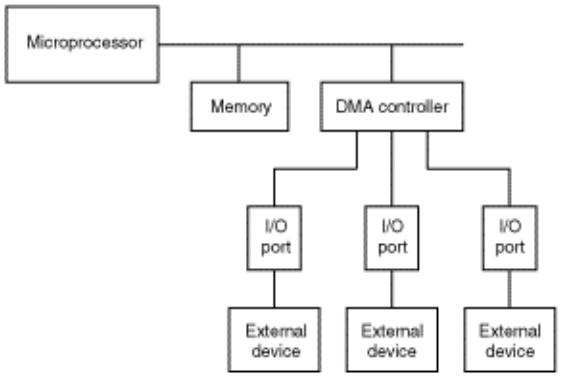

Figure 6-2: A microcomputer with a direct memory-access controller

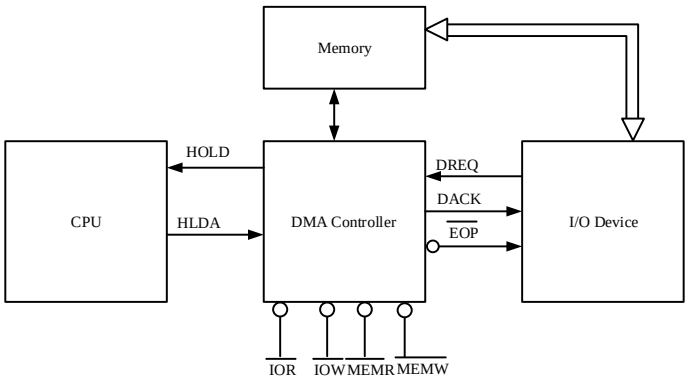

Fig 6-3: Block Diagram of DMA technique

6.2. Basic DMA operation

- Two control signals are used to request and acknowledge a direct memory access (DMA) transfer in the microprocessor-based system.
    1. The HOLD signal as an input (to the processor) is used to request a DMA action.
    2. The HLDA signal as an output that acknowledges the DMA action.
- When the processor recognizes the hold, it stops its execution and enters hold cycles.
- HOLD input has higher priority than INTR.
- The only microprocessor pin that has a higher priority than a HOLD is the RESET pin.
- HLDA becomes active to indicate that the processor has placed its buses at high-impedance state.

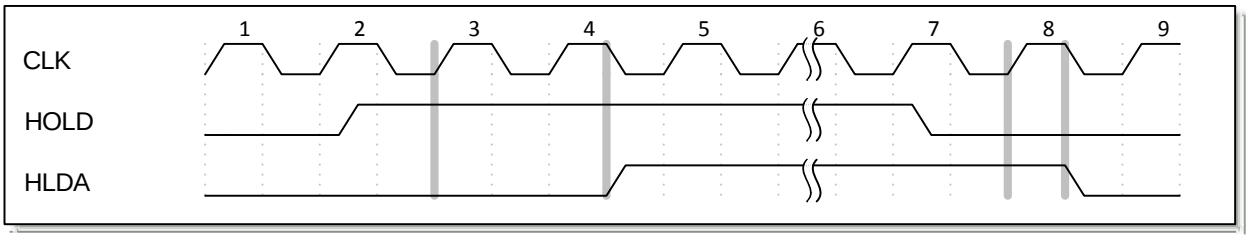

Figure 6-4: DMA timing diagram

6.3. 8237 DMA Controller

Features

- It is a 4-channel interface, which allows data transfer between memory and up to 4 I/O devices, bypassing CPU.
- A maximum of 16KB of data can be transferred by DMA sequentially at a time
- Initialization of the DMAC is done under program control for each channel.
- DMAC can be operated in three modes:
    - DMA Read (reading from memory, writing into peripheral)
    - DMA Write (Writing into memory, reading from peripheral)
    - DMA verify

  Priority for each of the 4-channels can be set in (a) fixed priority, (b) rotating priority

    - A terminal count register exists for each of 4 channels. The number of bytes of data to be transferred is stored in the D_{13}-D_0 positions of the 16-bit terminal count register.

The 8237A Multimode Direct Memory Access (DMA) Controller is a peripheral interface circuit for microprocessor systems. It is designed to improve system performance by allowing external devices to directly transfer information from the system memory. Memory-to-memory transfer capability is also provided. The 8237A offers a wide variety of programmable control features to enhance data throughput and system optimization and to allow dynamic reconfiguration under program control.

The 8237A is designed to be used in conjunction with an external 8-bit address latch. It contains four independent channels and may be expanded to any number of channels by cascading additional controller chips. The three basic transfer modes allow programmability of the types of DMA service by the user. Each channel can be individually programmed to Auto initialize to its original condition following an End of Process (EOP). Each channel has a full 64K address and word count capability

6.4. DMA Controller 8237 Interfacing

Figure 6-5 shows a convenient method for configuring a DMA system with the 8237A controller and an 8080A/8085AH microprocessor system.

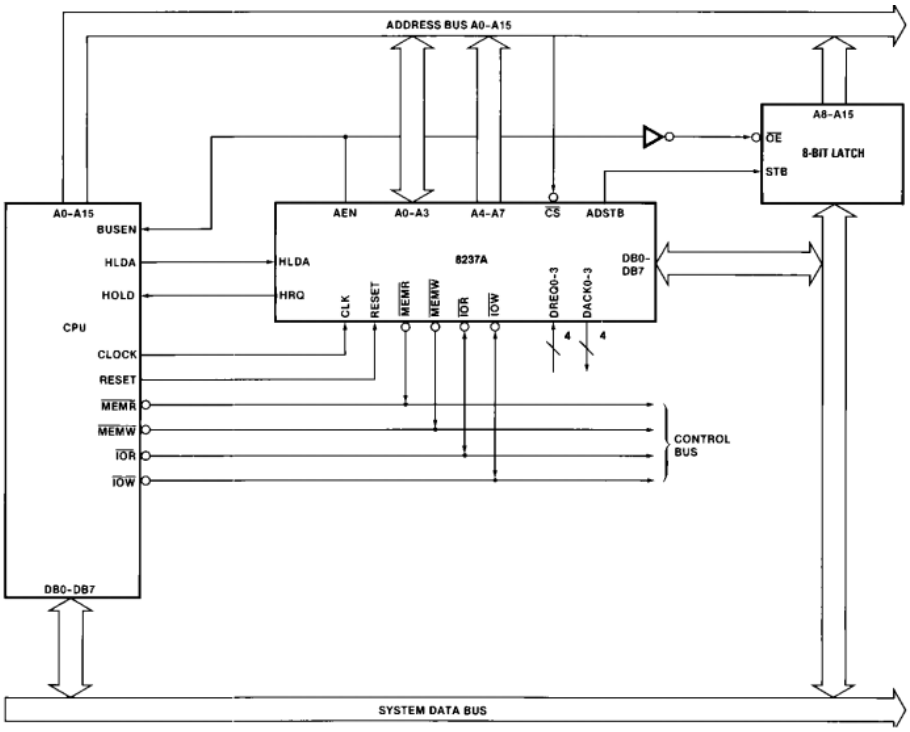

Fig. 6-5: Microprocessor and 8237 Interfacing

The multimode DMA controller issues a HRQ to the processor whenever there is at least one valid DMA request from a peripheral device. When the processor replies with a HLDA signal, the 8237A takes control of the address bus, the data bus and the control bus. The address for the first transfer operation comes out in two bytes -- the least significant 8 bits on the eight address outputs and the most significant 8 bits on the data bus. The contents of the data bus are then latched into an 8-bit latch to complete the full 16 bits of the address bus. The 8282 is a high speed, 8-bit, three-state latch in a 20-pin package. After the initial transfer takes place, the latch is updated only after a carry or borrow is generated in the least significant address byte. Four DMA channels are provided when one 8237A is used.

6.5. Interrupt

Interrupt is a mechanism by which an I/O or an instruction can suspend the normal execution of processor and get itself serviced. Generally, a particular task is assigned to that interrupt signal. In the microprocessor based system the interrupts are used for data transfer between the peripheral devices and the microprocessor.

Interrupt Service Routine (ISR)

A small program or a routine that when executed services the corresponding interrupting source is called as an ISR.

Maskable/Non-Maskable Interrupt

An interrupt that can be disabled by writing some instruction is known as Maskable Interrupt otherwise it is called Non-Maskable Interrupt.

There are 6 pins available in 8085 for interrupt:

1. TRAP
2. RST 7.5
3. RST6.5
4. RST5.5
5. INTR
6. INTA

Execution of Interrupts

When there is an interrupt requests to the Microprocessor then after accepting the interrupts Microprocessor send the INTA (active low) signal to the peripheral. The vectored address of particular interrupt is stored in program counter. The processor executes an interrupt service routine (ISR) addressed in program counter.

There are two types of interrupts used in 8085 Microprocessor:

1. Software Interrupts
2. Hardware Interrupts

Software Interrupts

A software interrupts is a particular instructions that can be inserted into the desired location in the program. There are eight Software interrupts in 8085 Microprocessor. From RST0 to RST7.

1. RST0
2. RST1
3. RST2
4. RST3
5. RST4
6. RST5

7. RST6
8. RST7

They allow the microprocessor to transfer program control from the main program to the subroutine program. After completing the subroutine program, the program control returns back to the main program.
We can calculate the vector address of these interrupts using the formula given below:

Vector Address = Interrupt Number * 8

For Example:

RST2: vector address=2*8 = 16 = 0010H

RST1: vector address=1*8 = 08 = 0008H

RST3: vector address=3*8 = 24= 0018H

Vector address table for the software interrupts:

Interrupt	Vector Address
RST0	0000_H
RST1	0008_H
RST2	0010_H
RST3	0018_H
RST4	0020_H
RST5	0028_H
RST6	0030_H
RST7	0038_H

Hardware Interrupt

The interrupt where the CPU pins are used to receive the interrupt request are called hardware interrupts. There are 6 interrupt pins in the microprocessor used as Hardware Interrupts given below:

1. TRAP
2. RST7.5
3. RST6.5
4. RST5.5
5. INTR

Note: *INTA is not an interrupt. INTA is used by the Microprocessor for sending the acknowledgement. TRAP has highest priority and RST7.5 has second highest priority and so on.*

The Vector address of these interrupts are given below:

Interrupt	Vector Address
RST7.5	$003C_H$
RST6.5	0034_H
RST5.5	$002C_H$
TRAP	0024_H

TRAP

It is non maskable edge and level triggered interrupt. TRAP has the highest priority and vectored interrupt. Edge and level triggered means that the TRAP must go high and remain high until it is acknowledged. In case of sudden power failure, it executes an ISR and send the data from main memory to backup memory.
As we know that TRAP cannot be masked but it can be delayed using HOLD signal. This interrupt transfers the microprocessor's control to location 0024H. TRAP interrupts can only be masked by reseting the microprocessor. There is no other way to mask it.

RST7.5

It has the second highest priority. It is maskable and edge level triggered interrupt. The vector address of this interrupt is 003CH. Edge sensitive means input goes high and no need to maintain high state until it is recognized.
It can also be reset or masked by reseting microprocessor. It can also be resetted by DI instruction.

RST6.5 and RST5.5

These are level triggered and maskable interrupts. When RST6.5 pin is at logic 1, INTE flip-flop is set. RST 6.5 has third highest priority and RST 5.5 has fourth highest priority. It can be masked by giving DI and SIM instructions or by reseting microprocessor.

INTR

It is level triggered and maskable interrupt. The following sequence of events occurs when INTR signal goes high:

1. The 8085 checks the status of INTR signal during execution of each instruction.
2. If INTR signal is high, then 8085 complete its current instruction and sends active low interrupt acknowledge signal, if the interrupt is enabled.
3. On receiving the instruction, the 8085 save the address of next instruction on stack and execute received instruction.

It has the lowest priority. It can be disabled by resetting the microprocessor or by DI and SIM instruction.

6.6. SIM (Set interrupt mask) and RIM (Read Interrupt Mask) Instruction

SIM Instruction

This is a multipurpose instruction and used to implement the 8085 interrupts 7.5, 6.5, 5.5, and serial data output. The instruction interprets the accumulator contents as shown in fig. 6.6

- Bit 0 is the mask for RST 5.5, bit 1 is the mask for RST 6.5 and bit 2 is the mask for RST 7.5.
  - ✓ If the mask bit is 0, the interrupt is available.
  - ✓ If the mask bit is 1, the interrupt is masked.
- Bit 3 (Mask Set Enable - MSE) is an enable for setting the mask.
  - ✓ If it is set to 0 the mask is ignored and the old settings remain.
  - ✓ If it is set to 1, the new setting are applied.
- Bit 4 of the accumulator in the SIM instruction allows explicitly resetting the RST 7.5 memory even if the microprocessor did not respond to it.
- Bit 5 is not used by the SIM instruction
- Bit 6 & Bit 7 is used for extra functionality such as serial data transmission.

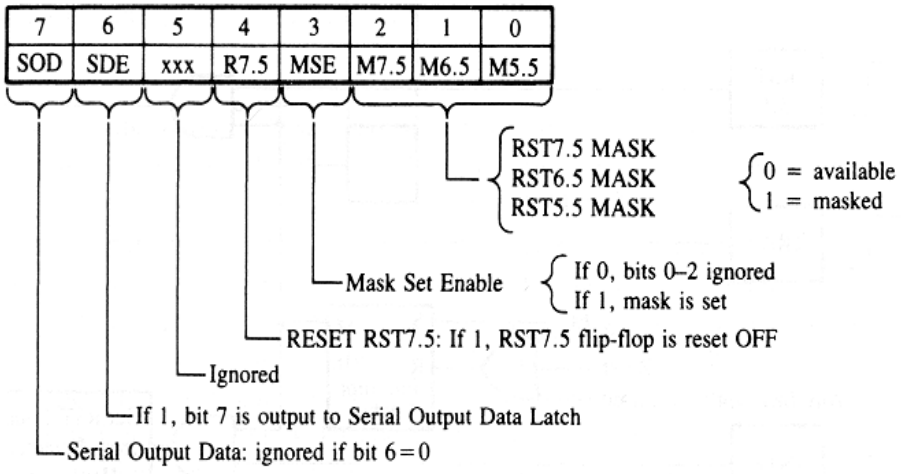

Figure 6-6: Format of SIM instruction

Example: Set the interrupt masks so that RST5.5 is enabled, RST6.5 is masked, and RST7.5 is enabled. First, determine the contents of the accumulator as:

0	**0**	**0**	**0**	**1**	**0**	**1**	**0**

Contents of accumulator are: 0A H

- Enable 5.5 bit 0 = 0
- Disable 6.5 bit 1 = 1
- Enable 7.5 bit 2 = 0
- Allow setting the masks bit 3 = 1
- Don't reset the flip flop bit 4 = 0
- Bit 5 is not used bit 5 = 0
- Don't use serial data bit 6 = 0
- Serial data is ignored bit 7 = 0

The second step is to set the interrupt as instruction below

```
EI              ; Enable interrupts including INTR
MVI A, 0A       ; Prepare the mask to enable RST 7.5, and 5.5, disable 6.5
SIM             ; Apply the settings RST masks
```

RIM Instruction

This is a multipurpose instruction used to read the status of interrupts 7.5, 6.5, 5.5 and read serial data input bit. The instruction loads eight bits in the accumulator with the interpretations shown in fig. 6-6.

- Bits 0-2 show the current setting of the mask for each of RST 7.5, RST 6.5 and RST 5.5 .They return the contents of the three mask flip flops.
- Bit 3 shows whether the maskable interrupt process is enabled or not. It can be used by a program to determine whether or not interrupts are enabled.
- Bits 4-6 show whether or not there are pending interrupts on RST 7.5, RST 6.5, and RST 5.5 .
- Bit 7 is used for Serial Data Input. The RIM instruction reads the value of the SID pin on the microprocessor and returns it in this bit.

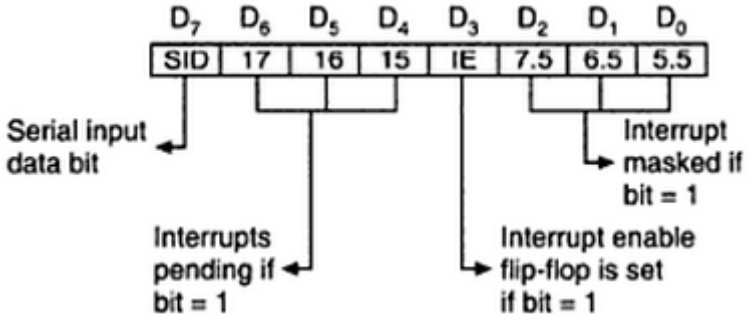

Figure 6-7: Format of RIM instruction

6.7. The 8259A Programmable Interrupt Controller

The 8259A is a programmable interrupt controller designed to work with Intel microprocessor 8085, 8086, and 8088. The 8259A interrupt controller can

1. Manage eight interrupts according to the instructions written into its control registers. This is equivalent to providing eight interrupt pins on the microprocessor in place of one INTR (8085) pin.

2. Vector an interrupt request anywhere in the memory map. However, all eight interrupts are spaced at the interval of either four or eight locations. This eliminates the major drawback of the 8085 interrupts in which all interrupts are vectored to memory location on page 00H.

3. Resolves eight levels of interrupt priorities in a variety of modes, such as fully nested mode, automatic rotation mode, and specific rotation mode.

4. Mask each interrupt request individually.

5. Read the status of pending interrupt, in service interrupts and masked interrupts.

6. Be set up to accept either the level triggered or edge triggered interrupt request.

7. Be expanded to 64 priority levels by cascading additional 8259As.

8. be set up to work with either the 8085 microprocessor mode or 8086/8088 microprocessor mode.

6.8. Block Diagram of 8259A

The block diagram of 8259A is shown in figure 6-8. It consists of eight parts. They are:

1. Data bus buffer
2. Read/Write logic
3. Control logic
4. Interrupt request register (IRR)
5. In Service register (ISR)
6. Interrupt mask register (IMR)
7. Priority resolver
8. The cascade buffer and comparator

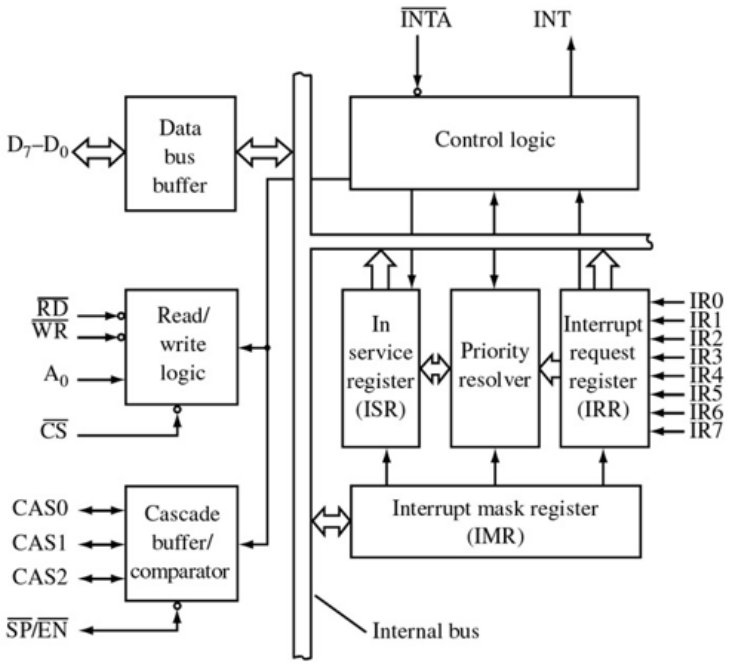

Fig. 6-8: Block diagram of 8259A

1. Data bus buffer: This is tri-stated, bidirectional 8-bit buffer is used to interface the internal data bus of the 8259A with the microprocessor data bus. Control words and status information are transferred through the data bus buffer.

2. Read/write control logic: The function of this block is to control all internal read/ write operations. When the address line A0 is at logic 0, the controller is selected to write a command or read a status. The chip select logic and A0 determine the port address of the accumulator.

3. Control logic: This block has two inputs: INT (Interrupt) as an output, and INTA¯(Interrupt Acknowledge) as an input. The INT is connected to the interrupt pin of the MPU. Whenever a valid interrupt is asserted, this signal goes high. The INTA is the interrupt acknowledge signal from MPU

4. Interrupt request register (IRR) and Priority Resolver: The Interrupt Request Register (IRR) has eight input lines (IR0-IR7) for interrupts. When these lines go high, the requests are stored in the register. The In-Service register (ISR) stores all the levels that are currently being serviced, and Interrupt Mask Register (IMR) stores the masking bits of the interrupt lines to be masked. The Priority resolvers (PR) examines these three registers and determines whether INT should be sent to the MPU.

5. Cascade Buffer/ Comparator: This block is used to expand the number of interrupt levels by cascading two or more 8259A.

6.9. Interrupt Operation

To implement interrupts, 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 word in the control register. The 8259A requires two types of control words: Initialization Command Words (ICWs) and Operational Command Words (OCWs). The 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. After the 8259A is initialized, the following sequence of events occurs when one or more interrupts request lines go high.

1. The IRR stores the requests.

2. The priority resolver checks three registers: the IRR for interrupt requests, the IMR for masking bits, and the ISR for the interrupt request is being serviced. It resolves the priority and sets the INT high when appropriate

3. The MPU acknowledges the interrupt by sending $\overline{INTA}$.

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 is accepted. Then, the opcode for the CALL instruction is placed on the data bus.

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

6. When the 8259A receives the second $\overline{INTA}$, it places the low order byte of CALL address on the data bus. At the third $\overline{INTA}$, 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 command word (ICW).

7. During the third $\overline{INTA}$ pulse, the ISR bit is reset either automatically or by a command word that must be issued at the end of the service routine. This option is determined by the initialization command word (ICW)

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

6.10. Priority Modes and Other Features:

Many Types of priority modes are available under software control in 8259A, and they can be changed dynamically during the program by writing appropriate command words. Commonly used priority modes are discussed as:

1. Fully Nested Mode:

This is a general-purpose mode in which all IRs (Interrupt Requests) are arranged from highest to lowest, with IR0 as the highest and IR7 as the lowest.

In addition, any IR can be assigned the highest priority the highest priority in this mode; the priority sequence will then begin at that IR. In the example below, IR4 has the highest priority, and IR3 has the lowest priority.

IR0	IR1	IR2	IR3	IR4	IR5	IR6	IR7
4	5	6	7	0	1	2	3

Lowest Priority

Highest Priority

2. Automatic Rotation Mode:

In this mode, a device, after being serviced, receives the lowest priority. Assume that the IR2 has just been serviced, it will receive the seventh priority as shown below:

IR0	IR1	IR2	IR3	IR4	IR5	IR6	IR7
1	6	7	0	1	5	3	4

3. Specific Rotation mode:

This mode is similar to the automatic rotation mode, except that the user can select any IR for the lowest priority, thus fixing all other priorities.

Additional Features of the 8259A

The 8259A is a complex device with various modes of operation. These modes are listed below for reference.

- Interrupt Triggering: The 8259A can accept an interrupt request with either the edge trigged mode or the level-triggered mode. The mode is determined by the initialization instructions.
- Interrupt Status: The status of the three interrupt register (IRR, ISR, and IMR) can be read, and this status information can be used to make interrupt process versatile.
- Polled Method: The 8259A can be set up to function in a polled environment. The MPU polls the 8259A rather than each peripheral.