UNIT II
HARDWARE INTERFACING WITH INTEL 8085

I/O INTERFACING WITH 8085
There are two types for interfacing I/O devices:

1. Memory mapped I/O device.

2. Standard I/O mapped I/O device or isolated I/O mapping.

Example 1:
A system requires 16kb EPROM and 16kb RAM. Also the system has 2 numbers of 8255, one number of 8279, one number of 8251 and one number of 8254. (8255 - Programmable peripheral interface; 8279-Keyboard/display controller, 8251 - USART and 8254 - Timer). Draw the Interface diagram. Allocate addresses to all the devices. The peripheral IC should be I/O mapped.

- The I/O devices in the system should be mapped by standard I/O mapping. Hence separate decoders can be used to generate chip select signals for memory IC and peripheral IC's.
- For 16kb EPROM, we can provide 2 numbers of 2764(8k x 8) EPROM.
- For 16kb RAM we can provide 2 numbers of 6264 (8k x 8) RAM.
- The 8kb memories require 13 address lines. Hence the address lines A0 - A12 are used for selecting the memory locations.
- The unused address lines A13, A14 and A15 are used as input to decoder 74LS138 (3-to-8-deeoder) of memory IC. The logic low enables of this decoder are tied to IO/ M(low) of 8085, so that this decoder is enabled for memory read/write operation. The other enable pins of decoder are tied to appropriate logic levels permanently. The 4-outputs of the decoder are used to select memory ICs and the remaining 4 are kept for future expansion.
- The EPROM is mapped in the beginning of memory space from 0000H to 3FFF.
- The RAM is mapped at the end of memory space from C000 to FFFFH.
There are five peripheral IC’s to be interfaced to the system. The chip-select signals for these IC’s are given through another 3-to-8 decoder 74LS138 (I/O decoder). The input to this decoder is A11, A12 and A13.

The address lines A13, A14 and A15 are logically ORed and applied to low enable of I/O decoder.

The logic high enable of I/O decoder is tied to IO / M(low) signal of 8085, so that this decoder is enabled for I/O read/write operation.

Fig - Internal address of 8255
I/O STRUCTURE OF A TYPICAL MICROCOMPUTER:
There are three major types of data transfer between the microcomputer and art I/O device. They are,

- Programmed I/O: In programmed I/O the data transfer is accomplished through an I/O port and controlled by software.
- Interrupt driven I/O: In interrupt driven I/O, the I/O device will interrupt the processor, and initiate data transfer.
- Direct memory access (DMA): In DMA, the data transfer between memory and I/O can be performed by bypassing the microprocessor.
INTERFACING I/O AND PERIPHERAL DEVICES:

1. For data transfer from input device to processor the following operations are performed.
   - The input device will load the data to the port.
   - When the port receives a data, it sends message to the processor to read the data.
   - The processor will read the data from the port.
   - After a data have been read by the processor the input device will load the next data into the port.

2. For data transfer from processor to output device the following operations are performed.
   - The processor will load the data to the port.
   - The port will send a message to the output device to read the data.
   - The output device will read the data from the port.
   - After the data have been read by the output device the processor can load the next data to the port.
   - The various INTEL 110 port devices are 8212, 8155/8156, 8255, 8355 and 8755.
     - 8212
     - The 8212 is a 24 pin IC.
     - It consists of eight number of D-type latches.
     - It has 8-input lines DI1 to DI8 and 8-output lines DO1 to DO8
     - The 8212 can be used as an input or output device
     - It has two selecting device DS1 (low) and DS2.
8155:
- It has two numbers of 8-bit parallel I/O port (port-A and B).
- One number of 6-bit parallel I/O port (port-C).
- It has 14 bit timer (operating in 4 modes).
- It has six internal addresses.
- It has one chip select pin CS (low).

8156:
- It has two numbers of 8-bit parallel I/O port (port-A and B).
- One number of 6-bit parallel 1 port (port-C).
- It has 14 bit timer (operating in 4 modes).
- It has six internal addresses.
- It has one chip select pin CS (low).
### Internal Device

<table>
<thead>
<tr>
<th>Internal Device</th>
<th>A₂</th>
<th>A₁</th>
<th>A₀</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Register/</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Status Register</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Port-A</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Port-B</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Port-C</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>LSB of Timer</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MSB of Timer</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Fig - Internal address of 8156

#### 8255:
- It has 3 numbers of 8-bit parallel I/O ports (port A, B and C).
- Port-A can be programmed in mode-0 mode-1 or mode-2 as input or output port.
- Port-B can be programmed in mode-1 and mode-2 as I/Oport.
- When ports A and B are in mode-0, the port-C can be used as I/O port.
- One logic low chip select (CS) pin.
- It requires four internal addresses.

#### 8355:
- It has 2KB ROM.
- It has two number of 8 bit port (A,B).
- It has one CS(low).
- It has four internal addresses.

#### 8755:
- It has 2Kb EPROM.
- It has two number of 8 bit port (A,B).
- It has one CS(low).
- It has four internal addresses.
### Internal Address of 8255

<table>
<thead>
<tr>
<th>Internal Device</th>
<th>$A_1$</th>
<th>$A_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port-A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Port-B</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Port-C</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Control Register</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### Internal Address of 8355

<table>
<thead>
<tr>
<th>Internal Device</th>
<th>$A_1$</th>
<th>$A_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port-A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Port-B</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>DDR-A</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>DDR-B</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### Internal Address of 8755

<table>
<thead>
<tr>
<th>Internal Device</th>
<th>$A_1$</th>
<th>$A_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port-A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Port-B</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>DDR-A</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>DDR-B</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
### MEMORY INTERFACING WITH 8085

The memory is made up of semiconductor material used to store the programs and data. Three types of memory is,

- Process memory
- Primary or main memory
- Secondary memory

<table>
<thead>
<tr>
<th>Device</th>
<th>Function</th>
<th>Internal addresses</th>
</tr>
</thead>
</table>
| **INTEL 8279** | Keyboard/display controller. Used for keyboard scanning and display refreshing. | Two-internal addresses:  
|             |                                                                          | $A_0 = 0 \rightarrow$ Data register  
|             |                                                                          | $A_0 = 1 \rightarrow$ Control register |
| **INTEL 8257 or INTEL 8237** | DMA controller. Used for supporting DMA access to I/O device. It acts as a master during DMA mode. It is a slave device during programming mode. | Sixteen-internal addresses:  
|             |                                                                          | $A_3 \ A_2 \ A_1 \ A_0$  
|             |                                                                          | $0 \ 0 \ 0 \ 0$  
|             |                                                                          | $0 \ 0 \ 0 \ 1$  
|             |                                                                          | $\ldots \ldots$  
|             |                                                                          | $1 \ 1 \ 1 \ 1$  
| **INTEL 8259** | Interrupt controller. Used to expand the hardware interrupt INTR to eight interrupts in 8085 based system and 256 interrupts in 8086 based system. | Two-internal addresses:  
|             |                                                                          | $A_0 = 0$  
|             |                                                                          | $A_0 = 1$  
| **INTEL 8253/8254** | Programmable Timer. Used in the system to produce various timing signals. It has three independent counters and can be programmed in six operating modes. | Four-internal addresses:  
|             |                                                                          | $A_1 \ A_0$  
|             |                                                                          | Counter-0 $0 \ 0$  
|             |                                                                          | Counter-1 $0 \ 1$  
|             |                                                                          | Counter-2 $1 \ 0$  
|             |                                                                          | Control Register $1 \ 1$  
| **INTEL 8251 USART** | Universal Synchronous/Asynchronous Receiver Transmitter. Used for serial data communication. | Two-internal addresses:  
|             |                                                                          | $C/\overline{D} = 0 \rightarrow$ Data register  
|             |                                                                          | $C/\overline{D} = 1 \rightarrow$ Control register |
TYPICAL EPROM AND STATIC RAM:

- A typical semiconductor memory IC will have n address pins, m data pins (or output pins).

- Having two power supply pins (one for connecting required supply voltage (V and the other for connecting ground).

- The control signals needed for static RAM are chip select (chip enable), read control (output enable) and write control (write enable).

- The control signals needed for read operation in EPROM are chip select (chip enable) and read control (output enable).

DECODER:

It is used to select the memory chip of processor during the execution of a program. No of IC’s used for decoder is,

- 2-4 decoder (74LS139)
- 3-8 decoder (74LS138)
### Table - Number of Address Pins and Data Pins in Memory ICs

<table>
<thead>
<tr>
<th>Memory IC EPROM/RAM</th>
<th>Capacity</th>
<th>Number of address pins</th>
<th>Number of data pins</th>
</tr>
</thead>
<tbody>
<tr>
<td>2708/6208</td>
<td>1kb</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>2716/6216</td>
<td>2kb</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>2732/6232</td>
<td>4kb</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>2764/6264</td>
<td>8kb</td>
<td>13</td>
<td>8</td>
</tr>
<tr>
<td>27256/62256</td>
<td>32kb</td>
<td>15</td>
<td>8</td>
</tr>
<tr>
<td>27512/62512</td>
<td>64kb</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>27010/62128</td>
<td>128kb</td>
<td>17</td>
<td>8</td>
</tr>
<tr>
<td>27020/62138</td>
<td>256kb</td>
<td>18</td>
<td>8</td>
</tr>
<tr>
<td>27040/62148</td>
<td>512kb</td>
<td>19</td>
<td>8</td>
</tr>
</tbody>
</table>

### Truth Table of 2-4 Decoder

<table>
<thead>
<tr>
<th>C</th>
<th>B</th>
<th>A</th>
<th>Y₁</th>
<th>Y₂</th>
<th>Y₃</th>
<th>Y₄</th>
<th>Y₅</th>
<th>Y₆</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### Block Diagram and Truth Table of 2-4 Decoder

**Fig:** Block diagram and Truth table of 2-4 decoder
Features of 8259:

1. It is programmed to work with either 8085 or 8086 processor.
2. It manages 8 interrupts according to the instructions written into its control registers.
3. In 8086 processor, it supplies the type number of the interrupt and the type number is programmable. In 8085 processor, the interrupt vector address is programmable. The priorities of the interrupts are programmable.
4. The interrupts can be masked or unmasked individually.
5. The 8259s can be cascaded to accept a maximum of 64 interrupts.
FUNCTIONAL BLOCK DIAGRAM OF 8259:

It has eight functional blocks. They are,

1. Control logic
2. Read Write logic
3. Data bus buffer
4. Interrupt Request Register (IRR)
5. In-Service Register (ISR)
6. Interrupt Mask Register (IMR)
7. Priority Resolver (PR)
8. Cascade buffer.

The data bus and its buffer are used for the following activities.

1. The processor sends control word to data bus buffer through D0-D7.
2. The processor read status word from data bus buffer through D0-D7
3. From the data bus buffer the 8259 send type number (in case of 8086) or the call opcode and address (in case of 8085) through D0-D7 to the processor.
   - The processor uses the RD (low), WR (low) and A0 to read or write 8259.
   - The 8259 is selected by CS (low).
   - The IRR has eight input lines (IR0-IR7) for interrupts. When these lines go high, the request is stored in IRR. It registers a request only if the interrupt is unmasked.
   - Normally IR0 has highest priority and IR7 has the lowest priority. The priorities of the interrupt request input are also programmable.
First the 8259 should be programmed by sending Initialization Command Word (ICW) and Operational Command Word (OCW). These command words will inform 8259 about the following:

Type of interrupt signal (Level triggered / Edge triggered).

1. Type of processor (8085/8086).
2. Call address and its interval (4 or 8)
3. Masking of interrupts.
4. Priority of interrupts.
5. Type of end of interrupts.

- The interrupt mask register (IMR) stores the masking bits of the interrupt lines to be masked. The relevant information is send by the processor through OCW.

- The in-service register keeps track of which interrupt is currently being serviced.

- The priority resolver examines the interrupt request, mask and in-service registers and determines whether INT signal should be sent to the processor or not.

- The cascade buffer/comparator is used to expand the interrupts of 8259.

- In cascade connection one 8259 will be directly interrupting 8086 and it is called master 8259.

- To each interrupt request input of master 8259 (IR0-IR7), one slave 8259 can be connected. The 8259s interrupting the master 8259 are called slave 8259s.

- Each 8259 has its own addresses so that each 8259 can be programmed independently by sending command words and independently the status bytes can be read from it.
PERIPHERAL INTERFACING - PPI INTERFACING (8255)

Peripheral Interfacing is considered to be a main part of Microprocessor, as it is the only way to interact with the external world. The interfacing happens with the ports of the Microprocessor.

- The main IC's which are to be interfaced with 8085 are:
  1. 8255 PPI
  2. 8259 PIC
  3. 8251 USART
  4. 8279 Key board display controller
  5. 8253 Timer/ Counter
  6. A/D and D/A converter interfacing.

PROGRAMMABLE PERIPHERAL INTERFACE - INTEL 8255

Pins, Signals and internal block diagram of 8255:

- It has 40 pins and requires a single +5V supply.
- The INTEL 8255 is a device used to parallel data transfer between processor and slow peripheral devices like ADC, DAC, keyboard, 7-segment display, LCD, etc.
- The 8255 has three ports: Port-A, Port-B and Port-C.
- Port-A can be programmed to work in any one of the three operating modes mode-0, mode-1 and mode-2 as input or output port.
- Port-B can be programmed to work either in mode-0 or mode-1 as input or output port.
- Port-C (8-pins) has different assignments depending on the mode of port-A and port-B.
- If port-A and B are programmed in mode-0, then the port-C can perform any one of the following functions.
• As 8-bit parallel port in mode-0 for input or output.

• As two numbers of 4-bit parallel ports in mode-0 for input or output.

• The individual pins of port-C can be set or reset for various control applications.

• If port-A is programmed in mode-1/mode-2 and port-B is programmed in mode-1 then some of the pins of port-C are used for handshake signals and the remaining pins can be used as input/ output lines or individually set/reset for control applications.

• The read/write control logic requires six control signals. These signals are given below.

1. RD (low): This control signal enables the read operation. When this signal is low, the microprocessor reads data from a selected I/O port of the 8255A.

2. WR (low): This control signal enables the write operation. When this signal goes low, the microprocessor writes into a selected I/O port or the control register.

3. RESET: This is an active high signal. It clears the control register and set all ports in the input mode.

4. CS (low), A0 and A1: These are device select signals. They are,

Interfacing of 8255 with 8085 processor:

• A simple schematic for interfacing the 8255 with 8085 processor is shown in fig.
Block diagram of 8255:

- The internal block diagram of 8255 is shown in fig:

<table>
<thead>
<tr>
<th>Internal Devices</th>
<th>A1</th>
<th>A0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Port B</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Port C</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Control Register</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
- The 8255 can be either memory mapped or I/O mapped in the system. In the schematic shown in above is I/O mapped in the system.

- Using a 3-to-8 decoder generates the chip select signals for I/O mapped devices.

- The address lines A4, A5 and A6 are decoded to generate eight chip select signals (IOCS-0 to IOCS-7) and in this, the chip select IOCS-1 is used to select 8255.

- The address line A7 and the control signal IO/M (low) are used as enable for the decoder.

- The address line A0 of 8085 is connected to A0 of 8255 and A1 of 8085 is connected to A1 of 8255 to provide the internal addresses.

- The data lines D0-D7 are connected to D0-D7 of the processor to achieve parallel data transfer.

- The I/O addresses allotted to the internal devices of 8255 are listed in table.
The 8251A is a programmable serial communication interface chip designed for synchronous and asynchronous serial data communication.

- It supports the serial transmission of data.
- It is packed in a 28 pin DIP.
- It is packed in a 28 pin DIP.

### INTERFACING WITH INTEL 8251A (USART)

<table>
<thead>
<tr>
<th>Internal Device</th>
<th>Decoder input and enable</th>
<th>Input to address pins of 8255</th>
<th>Hexa Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port-A</td>
<td>0 0 0 1</td>
<td>x x 0 0</td>
<td>10</td>
</tr>
<tr>
<td>Port-B</td>
<td>0 0 1</td>
<td>x x 0 1</td>
<td>11</td>
</tr>
<tr>
<td>Port-C</td>
<td>0 1</td>
<td>x x 1 0</td>
<td>12</td>
</tr>
<tr>
<td>Control Register</td>
<td>0 1</td>
<td>x x 1 1</td>
<td>13</td>
</tr>
</tbody>
</table>

*Note: Don’t care “x” is considered as zero.*
Read/Write control logic:

- The Read/Write Control logic interfaces the 8251A with CPU, determines the functions of the 8251A according to the control word written into its control register.

- It monitors the data flow.

- This section has three registers and they are control register, status register and data buffer.
The active low signals RD, WR, CS and C/D(Low) are used for read/write operations with these three registers.

When C/D(low) is high, the control register is selected for writing control word or reading status word.

When C/D(low) is low, the data buffer is selected for read/write operation.

When the reset is high, it forces 8251A into the idle mode.

The clock input is necessary for 8251A for communication with CPU and this clock does not control either the serial transmission or the reception rate.

Transmitter section:

- The transmitter section accepts parallel data from CPU and converts them into serial data.

- The transmitter section is double buffered, i.e., it has a buffer register to hold an 8-bit parallel data and another register called output register to convert the parallel data into serial bits.

- When output register is empty, the data is transferred from buffer to output register. Now the processor can again load another data in buffer register.

- If buffer register is empty, then TxRDY is goes to high.

- If output register is empty then TxEMPTY goes to high.

- The clock signal, TxC (low) controls the rate at which the bits are transmitted by the USART.

- The clock frequency can be 1,16 or 64 times the baud rate.
Receiver Section:

- The receiver section accepts serial data and convert them into parallel data

- The receiver section is double buffered, i.e., it has an input register to receive serial data and convert to parallel, and a buffer register to hold the parallel data.

- When the RxD line goes low, the control logic assumes it as a START bit, waits for half a bit time and samples the line again.

- If the line is still low, then the input register accepts the following bits, forms a character and loads it into the buffer register.

- The CPU reads the parallel data from the buffer register.

- When the input register loads a parallel data to buffer register, the RxRDY line goes high.

- The clock signal RxC (low) controls the rate at which bits are received by the USART.

- During asynchronous mode, the signal SYNDET/BRKDET will indicate the break in the data transmission.

- During synchronous mode, the signal SYNDET/BRKDET will indicate the reception of synchronous character.

MODEM Control:

- The MODEM control unit allows to interface a MODEM to 8251A and to establish data communication through MODEM over telephone lines.

- This unit takes care of handshake signals for MODEM interface.

- The 825 1A can be either memory mapped or I/O mapped in the system.
• 8251A in I/O mapped in the system is shown in the figure.

• Using a 3-to-8 decoder generates the chip select signals for I/O mapped devices.

• The address lines A4, A5 and A6 are decoded to generate eight chip select signals (IOCS-0 to IOCS-7) and in this, the chip select signal IOCS-2 is used to select 8251A.

• The address line A7 and the control signal IO / M(low) are used as enable for decoder.

• The address line A0 of 8085 is connected to C/D(low) of 8251A to provide the internal addresses.

• The data lines D0 - D7 are connected to D0 - D7 of the processor to achieve parallel data transfer.

• The RESET and clock signals are supplied by the processor. Here the processor clock is directly connected to 8251A. This clock controls the parallel data transfer between the processor and 8251A.

• The output clock signal of 8085 is divided by suitable clock dividers like programmable timer 8254 and then used as clock for serial transmission and reception.

• The TTL logic levels of the serial data lines and the control signals necessary for serial transmission and reception are converted to RS232 logic levels using MAX232 and then terminated on a standard 9-pin D-type connector.

• In 8251A the transmission and reception baud rates can be different or same.

• The device which requires serial communication with processor can be connected to this 9-pin D-type connector using 9-core cable.
- The signals TxEMPTY, TxRdy and RxRdy can be used as interrupt signals to initiate interrupt driven data transfer scheme between processor and 8251 A.

- I/O addresses of 8251A interfaced to 8085 is,
CASCADING OF 8259 & INTERFACING OF 8259 WITH 8085

CASCADING OF 8259

- The cascade pins (CAS0, CAS1 and CAS2) from the master are connected to the corresponding pins of the slave.

- For the slave 8259, the SP (low) / EN (low) pin is tied low to let the device know that it is a slave.

- The SP (low) / EN (low) pin can be used as input or output signal.

- In non-buffered mode it is used as input signal and tied to logic-1 in master 8259 and logic-0 in slave 8259.

- In buffered mode it is used as output signal to disable the data buffers while data is transferred from 8259A to the CPU.

It requires two internal address and they are A = 0 or A = 1.
- It can be either memory mapped or I/O mapped in the system. The interfacing of 8259 to 8085 is shown in figure is I/O mapped in the system.

- The low order data bus lines D0-D7 are connected to D0-D7 of 8259.

- The address line A0 of the 8085 processor is connected to A0 of 8259 to provide the internal address.

- The 8259 require one chip select signal. Using 3-to-8 decoder generates the chip select signal for 8259.

- The address lines A4, A5 and A6 are used as input to decoder.

- The control signal IO/M (low) is used as logic high enables for decoder and the address line A7 is used as logic low enable for decoder.

- The I/O addresses of 8259 are shown in table.

<table>
<thead>
<tr>
<th>Binary Address</th>
<th>Hexa address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A2, A6, A5</td>
</tr>
<tr>
<td>For A0 of 8259 to be zero</td>
<td>0 0 0</td>
</tr>
<tr>
<td>For A0 of 8259 to be one</td>
<td>0 0 0</td>
</tr>
</tbody>
</table>

*Note: Don't care "x" is considered as zero.*
INTERFACING 8259 WITH 8085 MICROPROCESSOR

Working of 8259 with 8085 processor:

- First the 8259 should be programmed by sending Initialization Command Word (ICW) and Operational Command Word (OCW). These command words will inform 8259 about the following,

1. Type of interrupt signal (Level triggered / Edge triggered).

2. Type of processor (8085/8086).

3. Call address and its interval (4 or 8)

4. Masking of interrupts.

5. Priority of interrupts.

6. Type of end of interrupts.
• Once 8259 is programmed it is ready for accepting interrupt signal. When it receives an interrupt through any one of the interrupt lines IR0-IR7 it checks for its priority and also checks whether it is masked or not.

• If the previous interrupt is completed and if the current request has highest priority and unmasked, then it is serviced.

• For servicing this interrupt the 8259 will send INT signal to INTR pin of 8085.

• In response it expects an acknowledge INTA (low) from the processor.

• When the processor accepts the interrupt, it sends three INTA (low) one by one.

• In response to first, second and third INTA (low) signals, the 8259 will supply CALL opcode, low byte of call address and high byte of call address respectively. Once the processor receives the call opcode and its address, it saves the content of program counter (PC) in stack and load the CALL address in PC and start executing the interrupt service routine stored in this call address.

➢ INTERFACING OF 8279 WITH 8085
The INTEL 8279 is specially developed for interfacing keyboard and display devices to 8085/8086/8088 microprocessor based system. The important features of 8279 are,

- Simultaneous keyboard and display operations.
- Scanned keyboard mode.
- Scanned sensor mode.
- 8-character keyboard FIFO.
- 1 6-character display.
- Right or left entry 1 6-byte display RAM.
- Programmable scan timing.

• The four major sections of 8279 are keyboard, scan, display and CPU interface.
Keyboard section:

- The keyboard section consists of eight return lines RL0 - RL7 that can be used to form the columns of a keyboard matrix.
- It has two additional input: shift and control/strobe. The keys are automatically debounced.
- The two operating modes of keyboard section are 2-key lockout and N-key rollover.
- In the 2-key lockout mode, if two keys are pressed simultaneously, only the first key is recognized.
- In the N-key rollover mode simultaneous keys are recognized and their codes are stored in FIFO.
- The keyboard section also have an 8 x 8 FIFO (First In First Out) RAM.
- The FIFO can store eight key codes in the scan keyboard mode. The status of the shift key and control key are also stored along with key code. The 8279 generate an interrupt signal when there is an entry in FIFO. The format of key code entry in FIFO for scan keyboard mode is,
### Functional block diagram

#### KEYBOARD AND DISPLAY INTERFACE USING 8279

**Block diagram of 8279:**

- The functional block diagram of 8279 is shown.

- In sensor matrix mode the condition (i.e., open/close status) of 64 switches is stored in FIFO RAM. If the condition of any of the switches changes then the 8279 asserts IRQ as high to interrupt the processor.
Display section:

- The display section has eight output lines divided into two groups A0-A3 and B0-B3.
- The output lines can be used either as a single group of eight lines or as two groups of four lines, in conjunction with the scan lines for a multiplexed display.
- The output lines are connected to the anodes through driver transistor in case of common cathode 7-segment LEDs.
- The cathodes are connected to scan lines through driver transistors.
- The display can be blanked by BD (low) line.
- The display section consists of 16 x 8 display RAM. The CPU can read from or write into any location of the display RAM.

Scan section:

- The scan section has a scan counter and four scan lines, SL0 to SL3.
- In decoded scan mode, the output of scan lines will be similar to a 2-to-4 decoder.
- In encoded scan mode, the output of scan lines will be binary count, and so an external decoder should be used to convert the binary count to decoded output.
- The scan lines are common for keyboard and display.
- The scan lines are used to form the rows of a matrix keyboard and also connected to digit drivers of a multiplexed display, to turn ON/OFF.

CPU interface section:

- The CPU interface section takes care of data transfer between 8279 and the processor.
- This section has eight bidirectional data lines DB0 to DB7 for data transfer between 8279 and CPU.
- It requires two internal address A =0 for selecting data buffer and A = 1 for selecting control register of 8279.
- The control signals WR (low), RD (low), CS (low) and A0 are used for read/write to 8279.
• It has an interrupt request line IRQ, for interrupt driven data transfer with processor.
• The 8279 require an internal clock frequency of 100 kHz. This can be obtained by dividing the input clock by an internal prescaler.
• The RESET signal sets the 8279 in 16-character display with two-key lockout keyboard modes.

**Programming the 8279:**

• The 8279 can be programmed to perform various functions through eight command words.

In a microprocessor system, when keyboard and 7-segment LED display is interfaced using ports or latches then the processor has to carry the following task.

• Keyboard scanning
• Key debouncing
• Key code generation
• Sending display code to LED
• Display refreshing

**Interfacing 8279 with 8085 processor:**

• A typical Hexa keyboard and 7-segment LED display interfacing circuit using 8279 is shown.

• The circuit can be used in 8085 microprocessor system and consist of 16 numbers of hexa-keys and 6 numbers of 7-segment LEDs.

• The 7-segment LEDs can be used to display six digit alphanumeric character.
- The 8279 can be either memory mapped or I/O mapped in the system. In the circuit shown is the 8279 is I/O mapped.

- The address line A0 of the system is used as A0 of 8279.

- The clock signal for 8279 is obtained by dividing the output clock signal of 8085 by a clock divider circuit.

- The chip select signal is obtained from the I/O address decoder of the 8085 system. The chip select signals for I/O mapped devices are generated by using a 3-to-8 decoder.

- The address lines A4, A5 and A6 are used as input to decoder.

- The address line A7 and the control signal IO/M (low) are used as enable for decoder.

- The chip select signal IOCS-3 is used to select 8279.

- The I/O address of the internal devices of 8279 are shown in table.

<table>
<thead>
<tr>
<th>Internal Device</th>
<th>Binary Address</th>
<th>Hexa Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Decoder input and enable</td>
<td>Input to address line of 8279</td>
</tr>
<tr>
<td></td>
<td>$A_0$ $A_1$ $A_2$ $A_3$ $A_4$ $A_5$ $A_6$</td>
<td></td>
</tr>
<tr>
<td>Data register</td>
<td>0 0 1 1 x x x 0</td>
<td>30</td>
</tr>
<tr>
<td>Control register</td>
<td>0 0 1 1 x x x 1</td>
<td>31</td>
</tr>
</tbody>
</table>
The circuit has 6 numbers of 7-segment LEDs and so the 8279 has to be programmed in encoded scan. (Because in decoded scan, only 4 numbers of 7-segment LEDs can be interfaced)

In encoded scan the output of scan lines will be binary count. Therefore an external, 3-to-8 decoder is used to decode the scan lines SL0, SL1 and SL2 of 8279 to produce eight scan lines S0 to S7.

The decoded scan lines S0 and S1 are common for keyboard and display.

The decoded scan lines S2 to S5 are used only for display and the decoded scan lines S6 and S7 are not used in the system.

Anode and Cathode drivers are provided to take care of the current requirement of LEDs.
• The pnp transistors, BC 158 are used as driver transistors.

• The anode drivers are called segment drivers and cathode drivers are called digit drivers.

• The 8279 output the display code for one digit through its output lines (OUT A0 to OUT A3 and OUT B0 to OUT B3) and send a scan code through, SL0- SL3.

• The display code is inverted by segment drivers and sent to segment bus.

• The scan code is decoded by the decoder and turns ON the corresponding digit driver. Now one digit of the display character is displayed. After a small interval (10 milli-second, typical), the display is turned OFF (i.e., display is blanked) and the above process is repeated for next digit. Thus multiplexed display is performed by 8279.

• The keyboard matrix is formed using the return lines, RL0 to RL3 of 8279 as columns and decoded scan lines S0 and S1 as rows.

• A hexa key is placed at the crossing point of each row and column. A key press will short the row and column. Normally the column and row line will be high.

• During scanning the 8279 will output binary count on SL0 to SL3, which is decoded by decoder to make a row as zero. When a row is zero the 8279 reads the columns. If there is a key press then the corresponding column will be zero.

• If 8279 detects a key press then it wait for debounce time and again read the columns to generate key code.

• In encoded scan keyboard mode, the 8279 stores an 8-bit code for each valid key press. The keycode consist of the binary value of the column and row in which the key is found and the status of shift and control key.
- After a scan time, the next row is made zero and the above process is repeated and so on. Thus 8279 continuously scan the keyboard.

**ADC 0809**

- The ADC0809 is an 8-bit successive approximation type ADC with inbuilt 8-channel multiplexer.

- The ADC0809 is suitable for interface with 8086 microprocessor.

- The ADC0809 is available as a 28 pin IC in DIP (Dual Inline Package).

- The ADC0809 has a total unadjusted error of ±1 LSD (Least Significant Digit).

- The ADC0808 is also same as ADC0809 except the error. The total unadjusted error in ADC0808 is ±1/2 LSD.

- The pin configuration of ADC0809/ADC0808 is,
### Internal Block Diagram & Working of ADC0809/ADC0808

The various functional blocks of ADC are 8-channel multiplexer, comparator, 256R resistor ladder, switch tree, successive approximation register, output buffer, address latch and decoder.

- The 8-channel multiplexer can accept eight analog inputs in the range of 0 to 5V and allow one by one for conversion depending on the 3-bit address input. The channel selection logic is,
- The successive approximation register (SAR) performs eight iterations to determine the digital code for input value. The SAR is reset on the positive edge of START pulse and start the conversion process on the falling edge of START pulse.

A conversion process will be interrupted on receipt of new START pulse.
The End-Of-Conversion (EOC) will go low between 0 and 8 clock pulses after the positive edge of START pulse.

The ADC can be used in continuous conversion mode by tying the EOC output to START input. In this mode an external START pulse should be applied whenever power is switched ON.

- The 256R resistor network and the switch tree is shown in fig.

- The 256R ladder network has been provided instead of conventional R/2R ladder because of its inherent monotonic, which guarantees no missing digital codes.

- Also the 256R resistor network does not cause load variations on the reference voltage.

- The comparator in ADC0809/ADC0808 is a chopper- stabilized comparator. It converts the DC input signal into an AC signal, and amplifies the AC signal using high gain AC amplifier. Then it converts AC signal to DC signal. This technique limits the drift component of the amplifier, because the drift is a DC component.
and it is not amplified/passed by the AC amplifier. This makes the ADC extremely insensitive to temperature, long term drift and input offset errors.

- In ADC conversion process the input analog value is quantized and each quantized analog value will have a unique binary equivalent.

- The quantization step in ADC0809/ADC0808 is given by,

\[
Q_{\text{step}} = \frac{V_{\text{REF}}}{2^8} = \frac{V_{\text{REF}(+) - V_{\text{REF}(-)}}}{256_{10}}
\]

The digital data corresponding to an analog input \((V_{\text{in}})\) is given by,

\[
\text{Digital data} = \left( \frac{V_{\text{in}}}{Q_{\text{step}}} - 1 \right)_{10}
\]

**EXAMPLE 1**

Let, \(V_{\text{REF}(+)} = 3.84\text{V},\ V_{\text{REF}(-)} = 0\text{V}\)

\[
\therefore Q_{\text{step}} = \frac{V_{\text{REF}(+) - V_{\text{REF}(-)}}}{256_{10}} = \frac{3.84}{256} = 0.015\text{V} = 15\text{mV}
\]

Let the input analog voltage be 2.56V. Now the digital data corresponding to 2.56V is given by.
DAC 0800

- To convert the digital signal to analog signal a Digital-to-Analog Converter (DAC) has to be employed.

- The DAC will accept a digital (binary) input and convert to analog voltage or current.

- Every DAC will have "n" input lines and an analog output.

- The DAC require a reference analog voltage (Vref) or current (Iref) source.

- The smallest possible analog value that can be represented by the n-bit binary code is called resolution.

- The resolution of DAC with n-bit binary input is $1/2^n$ of reference analog value.

- Every analog output will be a multiple of the resolution.

**EXAMPLE 2**

Let $V_{REF}(+) = 5V$, $V_{REF}(-) = 0V$

\[
Q_{step} = \frac{V_{REF}(+) - V_{REF}(-)}{256_{10}} = \frac{5}{256} = 0.01953125
\]

Let the input analog voltage be 1.25V. Now the digital data corresponding to 1.25V given by,

\[
\text{Digital data} \times Q_{step} = \frac{V_{in}}{256_{10}} - 1 = \frac{1.25}{0.01953125} - 1 = 63_{10} = 3F_{16} = 0011'1111_{2}
\]
For example, consider an 8-bit DAC with reference analog voltage of 5 volts. The analog values for all possible digital input are as shown.

**PIN DIAGRAM & BLOCK DIAGRAM OF DAC0800**

- The DAC0800 is an 8-bit, high speed, current output DAC with a typical settling time (conversion time) of 100 ns.

- It produces complementary current output, which can be converted to voltage by using simple resistor load.

- The DAC0800 require a positive and a negative supply voltage in the range of ±5V to ±18V.

- It can be directly interfaced with TTL, CMOS, PMOS and other logic families.

- For TTL input, the threshold pin should be tied to ground (VLC = 0V).

- The reference voltage and the digital input will decide the analog output current, which can be converted to a voltage by simply connecting a resistor to output terminal or by using an op-amp I to V converter.

- The DAC0800 is available as a 16-pin IC in DIP.

- The pin configuration of DAC0800 is,

**Interfacing DAC0-800 with 8085**

<table>
<thead>
<tr>
<th>Digital Input</th>
<th>Analog Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000 0000</td>
<td>$0 \times 5$ Volts</td>
</tr>
<tr>
<td>0000 0001</td>
<td>$1 \times 5$ Volts</td>
</tr>
<tr>
<td>0000 0010'</td>
<td>$2 \times 5$ Volts</td>
</tr>
<tr>
<td>0000 0011'</td>
<td>$3 \times 5$ Volts</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>1111 1111</td>
<td>$255 \times 5$ Volts</td>
</tr>
</tbody>
</table>
The internal block diagram of DACO800 is,
INTERFACING OF DAC0-800 WITH 8085

The DAC0800 can be interfaced to 8085 system bus by using an 8-bit latch and the latch can be enabled by using one of the chip select signal generated for I/O devices. A simple schematic for interfacing DAC0800 with 8085 is,

- In this schematic the DAC0800 is interfaced using an 8-bit latch 74LS273 to the system bus.

- The 3-to-8 decoder 74LS 138 is used to generate chip select signals for I/O devices.

- The address lines A4, A5 and A6 are used as input to decoder.

- The address line A7 and the control signal IO/M (low) are used as enable for decoder.

- The decoder will generate eight chip select signals and in this the signal IOCS-7 is used as enable for latch of DAC.

- The I/O address of the DAC is shown in table.

<table>
<thead>
<tr>
<th>Device</th>
<th>Binary Address</th>
<th>Hexa Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAC Latch 74LS273</td>
<td>0 1 1 1 x x x x</td>
<td>70</td>
</tr>
</tbody>
</table>
- In order to convert a digital data to analog value, the processor has to load the data to latch.

- The latch will hold the previous data until next data is loaded.

- The DAC will take definite time to convert the data. The software should take care of loading successive data only after the conversion time.

- The DAC 0800 produces a current output, which is converted to voltage output using Ito V converter.

<table>
<thead>
<tr>
<th>Device</th>
<th>Binary Address</th>
<th>Hexa Address</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAC Latch 74LS273</td>
<td>A_13 A_12 A_11</td>
<td>6000</td>
<td>External data memory address space</td>
</tr>
</tbody>
</table>

---

**Table:**

<table>
<thead>
<tr>
<th>Device</th>
<th>Decoder Input</th>
<th>Unused Address Lines</th>
<th>Hexa Address</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAC Latch 74LS273</td>
<td>A_13 A_12 A_11</td>
<td>x x x x x x x x x</td>
<td>6000</td>
<td>External data memory address space</td>
</tr>
</tbody>
</table>