UNIT - 4

Serial communication Interface : (8251)

To implement serial communication in microprocessor system we need basically two devices. They are

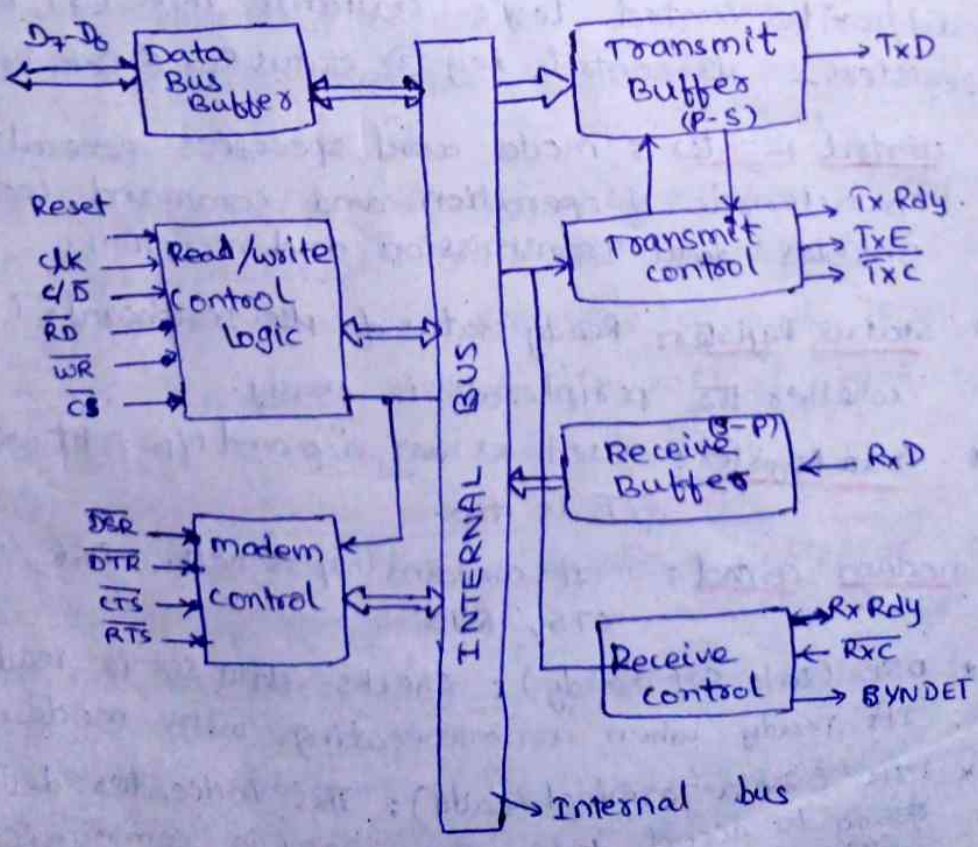
1. parallel to serial converter
2. serial to parallel converter

To transmit byte data, it's necessary to convert byte into eight serial bits. This can be done by using the parallel to serial converter.

similarly at the reception end these serial bits can be converted into parallel 8-bit data (byte). Here the serial to parallel converter is used.

The device which provides synchronous as well as asynchronous transmission and reception is called USART universal Synchronous - Asynchronous Receiver - Transmitter.

8251 (USART) - Block diagram :-



The block diagram includes Data Bus Buffer, Read/write control logic, modem control, Transmit buffer, Transmit control, Receive Buffer, Receive control.

Data Bus Buffer:- It's 8 bit Bi-directional data bus used to Read/write status word or command word or data from or to the 8251.

Read/write control logic:- It consists of 6 input signals

* $\overline{CS}$: 8251 is selected by microprocessor unit (MPU) for communication.
* C/$\overline{D}$: control/$\overline{Data}$. when it's high control or status register is addressed otherwise data buffer is addressed.
* $\overline{WR}$: a low on this I/P allows the MPU to write data or command word to the 8251.
* $\overline{RD}$: a low on this I/P allows the MPU to read data or command word from 8251.
* Reset: makes 8251 in idle state.
* clk: usually connected to system clock for communication with the µP.

Read/write control logic contains three (3) buffer registers- ① control Reg ② status Reg ③ Data Reg.

* control Register: mode word specifies general characteristic of operation and command word enables data Transmission and reception.
* Status Register: Ready status of the peripherals i.e whether the peripherals is ready.
* Data Register: used as an i/p and o/p port when C/$\overline{D}$ is low.

modem control: It contains 4 signals. DSR, DTR, CTS, RTS

* DSR (Data Set Ready): Checks data set is ready or not It's ready when communicating with modem.
* DTR (Data Terminal Ready): This indicates device is ready to accept data when 8251 is communicating with modem.

* CTS (clear to send Data): a low on this input enables the 8251 to transmit serial data if the TxE bit in the command byte is set to '1'.
* RTS (Request to send): Receiver is ready to receive the data.

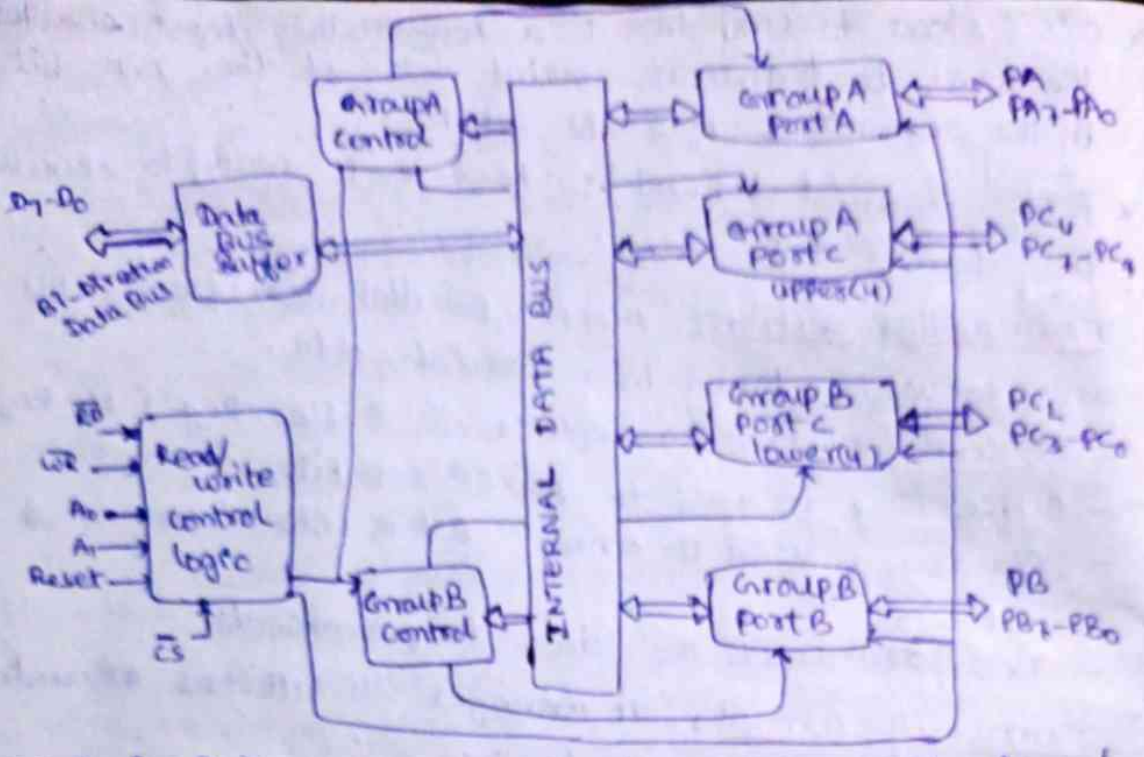
Transmitter section: Accepts parallel data from mpu and converts them into serial data.
- It contains two(2) registers - Buffer Reg & o/p Reg.
- Buffer Reg is used to process 8 bits.
- o/p Reg is used to convert 8 bits into stream of 8 serial bits.
- TxD transmits the data to peripherals.

Transmit control: It manages all activities associated with transmission of serial data.
TxRdy: (Transmit Ready) this o/p signal indicates that buffer register is empty and 8251 is ready to accept data
TxE: (Transmitter Empty) indicates the o/p buffer is empty
$\overline{TxC}$: (Transmitter clock): controls the rate at which bits are transmitted by 8251.

Receiver Buffer: Accepts serial data on the RxD line & converts the serial data to parallel data.

Receiver control: manages all receiver related activities
RxRdy: it goes high when 8251 has data in the buffer register and ready to transfer it to the mpu.
$\overline{RxC}$: The clock controls the rate at which the character is to be received by 8251.

PPI (programmable peripheral Interface) - 8255:
8255 is programmable peripheral Interface. Its para-
- llel I/O device (or) parallel communication interface.
- This acts as interfacing circuit b/n peripheral device and 8085 or 8086 up.
- The block diagram of 8255 consists of Data Bus Buffer, control logic and Group A, Group B controls.



* **Data Bus Buffer:** It consists of 3 state bidirectional Buffer and used to interface the 8255 to the system Data Bus. Based on the instruction data is transmitted or received.
  - Control word / status word is also transmitted.
* **Read/write control logic:** it manages all the internal or external transfer of both data + control or status
  - **A0 & A1:** these i/p signals along with $\overline{RD}$ & $\overline{WR}$ i/ps control selection of control/status word registers or one of the three ports.

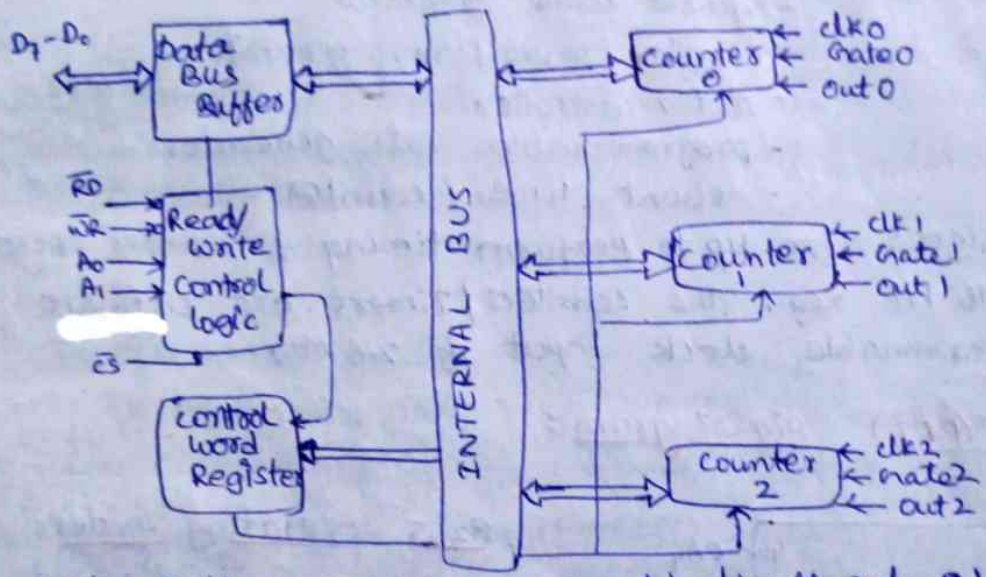
A_1	A_0	Port
0	0	port A
0	1	Port B
1	0	Port C
1	1	control Reg

* $\overline{CS}$: (chip select) a low on this i/p enables the com b/w 8255 and CPU.
* $\overline{RD}$, $\overline{WR}$: these are used to read the data from CPU and write data to the CPU.
* **Group A, Group B control:**
  Group A sends the control signals to port A and port C upper.

GroupB sends control signals to port C lower and port B.

- Port A: its 8 bit buffer I/O latch. It can be program... mode 0, mode 1 and mode 2.
- Port B: its 8 bit buffer I/O latch. can be programmed by mode 0, mode 1.
- port C: its 8 bit unlatched buffer. its splitted into upper and lower(4).

programmable Interval Timer (8253/8254):



* Data Bus Buffer: its a tristate bi-directional 8 bit buffer which is used to interface the 8253/8254 to the system bus. It has 3 basic functions.
  - → (1) programming the modes of 8253/8254.
  - → (2) loading the count Reg.
  - → (3) reading the count values.

Read/write control logic: it includes $\overline{RD}$, $\overline{WR}$, A0, A1 and $\overline{CS}$ counters are selected by A1, A0.

A0	A1	counter
0	0	counter 0
0	1	counter 1
1	0	counter 2
1	1	control word

control word Register: this reg is selected when A1 A0 are both '1'. Its used to write a command word.

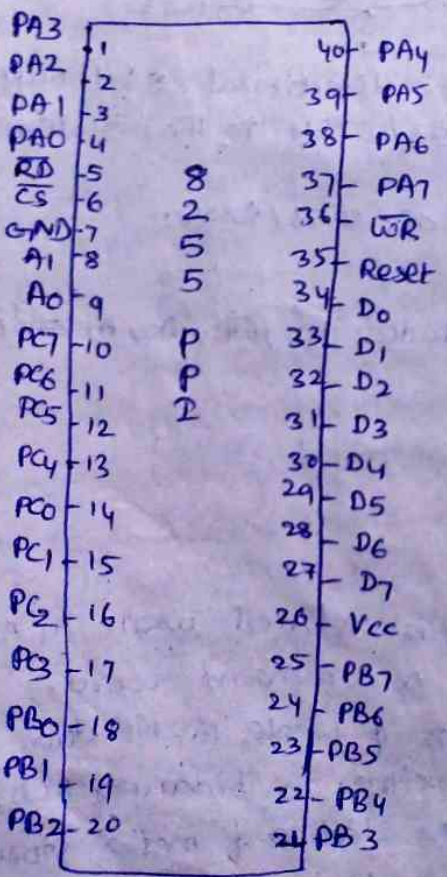
* counter: Each counter consists of single 16-bit down-counter. which can be operated either in binary or BCD.
- Its I/p & O/p is configured by the selection of modes stored in the control word register.

- The programmer can read the contents of any 3 counters without disturbing the actual count in process.
- Each counter has 2 i/p pins Gate, clock and one o/p pin 'out'
- To operate a counter a 16-bit counter is loaded in Reg
- Gate is used to starting interuption and restart the counter.
- OUT pins are o/p of the counter.

Applications:
- Real time clock
- square wave generator
- Complex wave form generator
- motor controller
- programmable rate generator
- Event timer / counter.

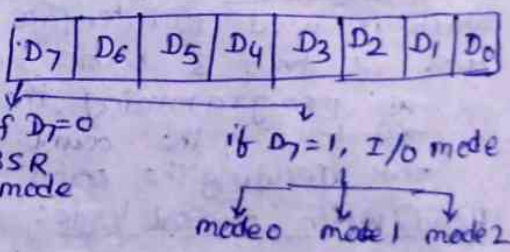
- Designed for up to perform timing & counter operation using 16-bit Reg. The counters / Timers are software programmable clock input of 2.6 MHz

8255 (PPI) Pin Diagram:



8255 operating modes:

Based on control word



I/O modes:

Three are 3 i/o modes in 8255 they are mode 0, mode 1, mode

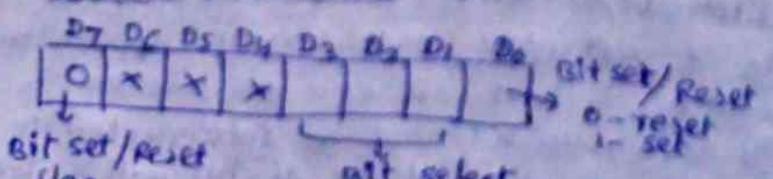
mode 0: (simple I/O)
- Port A, Port B is used as 2 8 bit ports.
- Port C used as 2 - 4 bit ports
- each port can be programmed in either i/p or o/p mode.
- o/p's are latched and i/p are not latched.

* mode1 : (I/O with Handshake) :
- Port A / Port B are used as 8-bit ports I/O ports.
- in this mode, i/p or o/p data transfer is controlled by Handshake signal.
- Handshaking signals are used to transfer data b/w devices whose data transfer speeds are not same.
- each port uses '3' lines from port 'c' as hand shake the remaining two lines can be used for simple I/O functions.
- Input and o/p data are latched.
- Interrupt logic is supported.

* mode2 : this mode allows bi-directional data transfer over a single 8-bit data bus using handshaking signals. This feature is available only in groups with port A as 8-bit bi-directional data bus and PC_3-PC_7 are used for hand shaking purpose.
- In this both i/ps and o/ps are latched.
- Due to use of single 8-bit data bus for bi-direction data transfer, the data sent out by the CPU through Port A appears on the bus connecting it to the peripheral only when the peripheral requests it.
- The remaing lines of port c i.e PC0-PC2 can be used for simple I/O functions.
- port B can be programmed in mode 0 or mode 1. when port B programed in mode 1, PC0-PC2 lines of port C are used as hand shake signals.

Control word formats: The ports of 8255 can be programmed for any other mode by writing a single control word into control Register, when required.

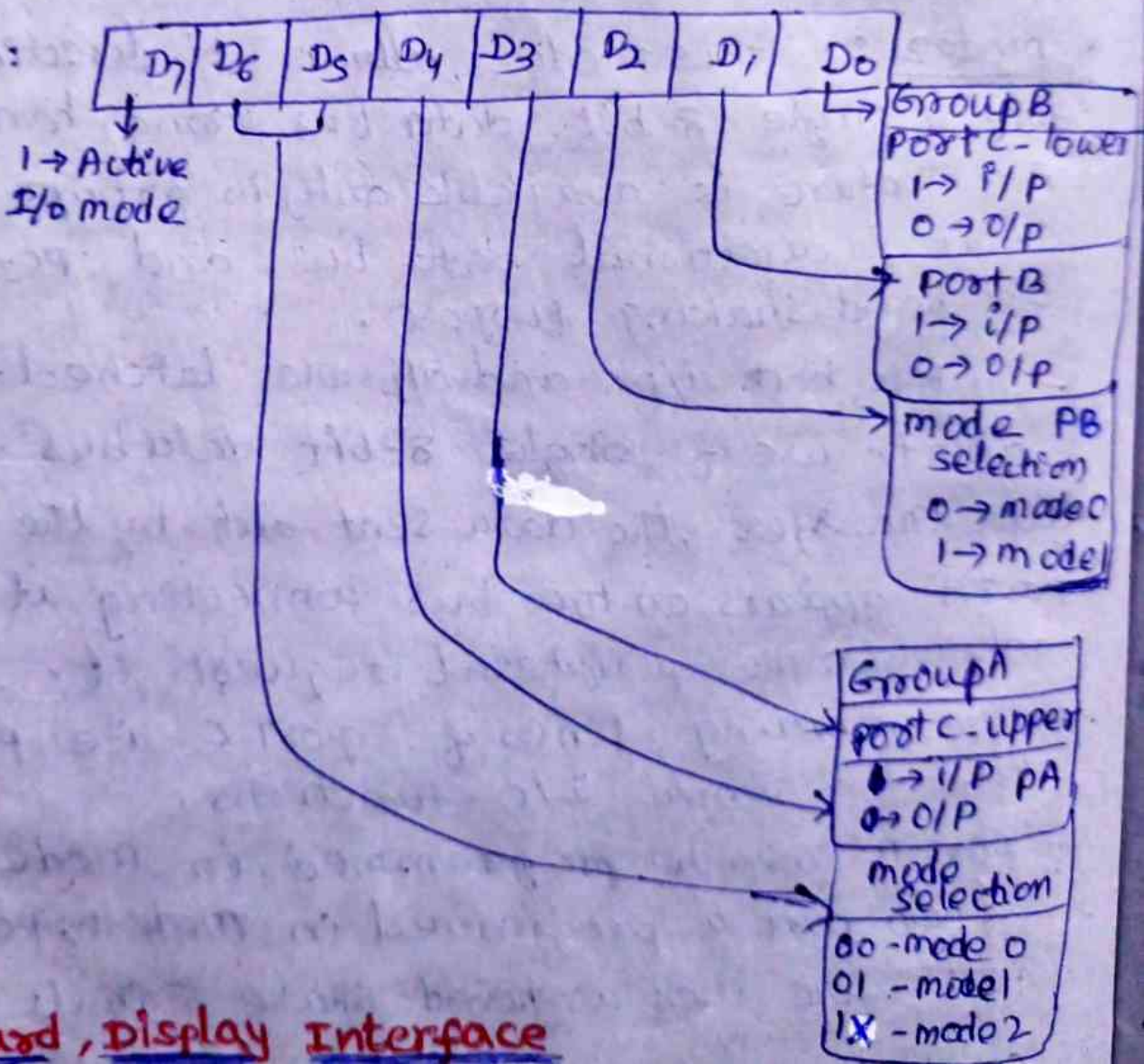
Bit set / Reset mode :



B_3	B_2	B_1
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

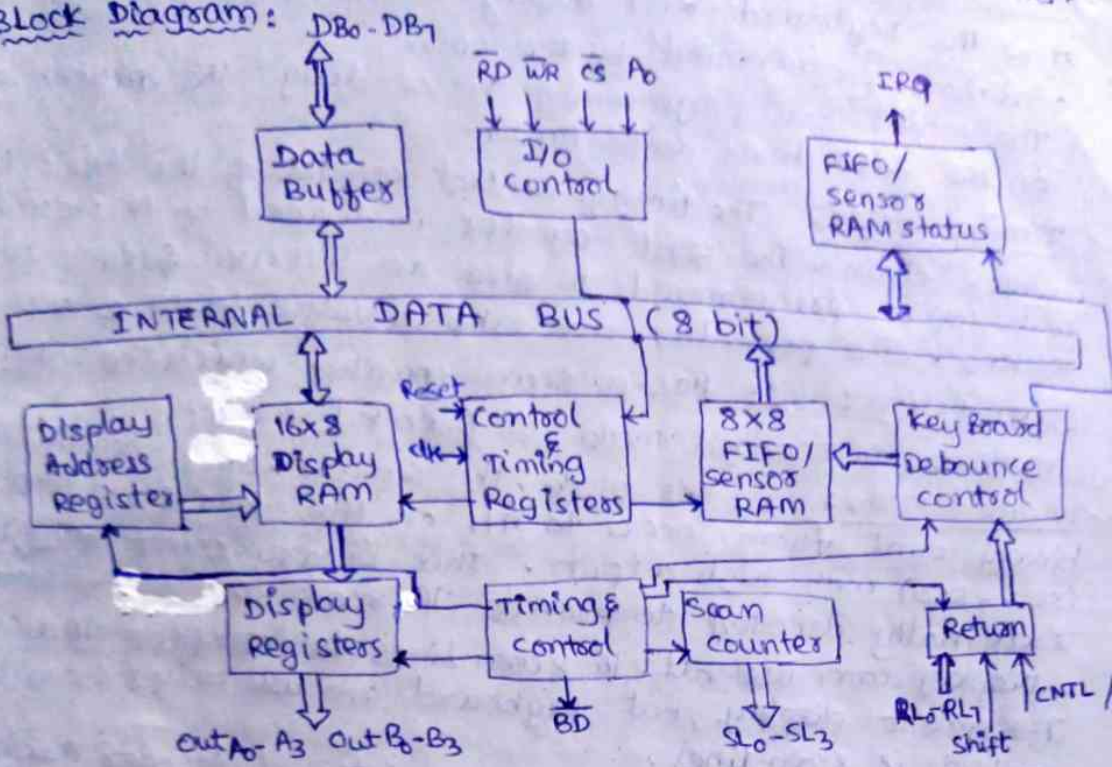
- The eight possible combinations of the states of bits $D_3 - D_1$ (B_3, B_2, B_1) in the BSR mode determine particular bit in $PC_0 - PC_7$ being set or Reset as per status of bit D_0.
- A BSR word is to be written for each bit that is to be set or reset.
- The BSR word can also be used for enabling or disabling interrupt signals generated by port c when 8255 is programmed for mode 1 and mode 2 operations

* I/O mode:



✓ IC 8279 independently keeps display refresh and scans matrix keyboard.

Block Diagram:



It contains 4 main sections:

- ✓ CPU Interface & control section
- ✓ keyboard section
- ✓ scan section
- ✓ Display section

CPU Interface & control section: This section consists of data buffers, I/O control, control & Timing Registers and Timing & control logic.

✓ Data Buffer: The data buffers are 8 bit bi-directional buffe that connects the internal data bus to the external data bus.

✓ I/O control: The I/O control section uses the A0, $\overline{CS}$, $\overline{RD}$, $\overline{WR}$ signals to control data flow to and from the various internal registers and buffers.

- The data flow toard from 8279 is enabled when $\overline{CS}=0$
- when A0 is '0' ($A_0=0$) data is transferred and when
- $A_0=1$, command word or status word is transferred.
- $\overline{RD}$, $\overline{WR}$ determine the direction of data flow through the data buffers.

A0	$\overline{RD}$	$\overline{WR}$	Interpretation
0	1	0	Data from CPU to 8279
0	0	1	Data to CPU from 8279
1	1	0	command word from CPU to 8279
1	0	1	status word to CPU from 8279

Control and Timing Registers: The control and timing Reg. store the keyboard and display modes and other operating conditions programmed by the CPU.
- The modes are programmed by sending the proper com... on the data lines with A0 = 1.

Timing control: The timing control consists of the basic timing counter chain. The first counter is divided by N prescaler that can be programmed to give an internal frequency of 100 kHz. The prescalar can take a value from 2 to 31.

Scan Section: It has a scan counter which has two modes: ① Encoded mode ② Decoded mode.

Encoded mode: In this mode, the scan counter provides a binary count from 0000 to 1111 on the four scan lines (SC3 - SC0) with high outputs. This binary count must be externally decoded to provide 16 scan lines.
- Display can use all 16 scan lines to interface 16 digit 7-segment display, but keyboard can use only 8 scan lines out of 16 scan lines.

Decoded mode: In this mode, the internal decoder decodes the least significant 2 bits of binary count and provides four possible combinations on the 4 scan lines SC3 - SC0: 1110, 1101, 1011, 0111. Thus the output of decoded scan is active low.
- These four active low output lines can be used directly to interface 4 digit 7-segment display, 8×4 matrix keyboard eliminating the external decoder.

Keyboard Section: consists of Return Buffers, Keyboard debounce and control, FIFO/sensor RAM and FIFO/sensor RAM status.

Return Buffers: The 8 (RL7 - RL0) return lines, input from keyboard to 8279.

- whenever key pressed, the information transferred to FIFO sensor RAM.

Keyboard Debounce and control: is enabled only when scanned keyboard mode is selected.
- In the scanned keyboard mode, return lines are scanned looking for key closures in that row.
- If the debounce circuit detects a close switch, it waits about 10 msec to check if the switch remains closed. If it does, the address of the switch in the matrix plus status of shift and control keys are transferred to FIFO RAM.

FIFO/sensor RAM: Its a dual function 8x8 RAM. Each new entry is written into successive RAM positions and then read in the order of entry.
- Each row of sensor RAM is loaded with the status of the corresponding row of sensor in the sensor matrix.

FIFO/sensor RAM status: it keeps track of the number of characters in the FIFO and whether it is full or empty.
- The status logic also makes IRQ signal high when the FIFO is not empty, which can be used to interrupt CPU telling that key press is detected and key code is available in FIFO RAM.

Display RAM: It is 16x8 RAM which stores the display codes for 16 digits. It can be accessed directly by CPU.
- In decoded mode, 8279 uses only first four locations of display RAM.
- In encoded mode, 8279 uses first locations for 8 digit display and 16 locations for 16 digit display.

Display Address Registers: hold the address of the byte currently being written or read by the CPU and scan count value.

Display Registers: these are two 4-bit registers A and B. They hold the bit pattern of character to be displayed.

DMA controller (8257)
- Direct memory Access controller designed by Intel to transfer data at a faster rate.
- It allows the device to transfer data directly to/from memory without CPU interference.
- using it, the device requests the CPU to hold its data, address and control bus so the device is free to transfer data directly to/from the memory. The DMA data transfer is initiated only after receiving HLDA signal from the CPU.
- Initially, when any device has to send data b/n the device & memory the device has to send DMA Request (DRQ) to DMA controller.
- The DMA controller sends Hold request to the CPU and waits for the CPU to ascertain the HLDA, then the µP tristates all the data bus, address bus and control bus. The CPU leaves the control over bus and acknowledges the hold request through HLDA signal.
- Now the CPU will be in hold state and DMA controller has to manage the operations over buses b/n CPU, memory & I/O devices.

8257 Architecture:
- DRQ0 - DRQ3: These are 4 individual channels DMA request i/p's which are used by the peripherals for DMA services. when a fixed priority mode is selected, then DRQ0 has highest priority and DRQ3 has lowest priority.
- DACK0 - DACK3: These are active low DMA acknowledge lines

which updates the requesting the peripheral about the status of their request by the CPU. These lines can also act as strobe lines for the requesting devices.

✓ Data Bus Buffer: (D_7-D_0) These are bi-directional data lines which are used to interface the system bus with the internal bus of DMA controller.

✓ Read/write control logic:

- $\overline{IOR}$: It's used by the CPU to read internal registers of 8257, in the slave mode. In the master mode it is used to read data from the peripheral devices during a memory write cycle.
- $\overline{IOW}$: It's used to load the contents of the data bus to the 8-bit mode register or upper/lower byte of a 16-bit DMA address register or terminal count register.
- A_0-A_3: These are 4-least significant address lines. In the slave mode, they act as an i/p, which selects one of the register to be read or written.
  In master-mode, they are the 4 least significant memory address o/p lines generated by 8257.

✓ mode set Register:

- A_4-A_7: These are the higher nibble of the lower byte address generated by DMA in the master-mode.
- Ready: Is an active-high i/p signal, which makes DMA ready by inserting wait states.
- HRQ: it's used to receive the hold request signal from the o/p device.
- HLDA: It's the hold acknowledgement signal which indicates the DMA controller that the bus has been granted to the requesting peripheral by the CPU when it is set to 1.
- ADSTB: This signal is used to convert the higher byte of the memory address generated by the DMA controller into the latches.
- AEN: This signal is used to disable the address bus/data bus.
- TC: It stands for Terminal count, indicates the present DMA cycle to the present peripheral devices.
- Mark: The mark will be activated after each 128 cycles or integral multiples of it from the begining. It indicates the current DMA cycle is the 128th cycle since the previous mark output to the

8257 Block Diagram: