

UNIT-3

ssn

Objectives

To discuss about :

- 8255 – Programmable Peripheral Interface (PPI)
- 8254 - Programmable Timer Interface
- 8279 - Programmable Keyboard & Display Interface
- 8259 - Programmable Interrupt Controller
- 8237 - Programmable DMA Controller
- 8251 –Programmable Communication Interface
- Memory and IO interfacing



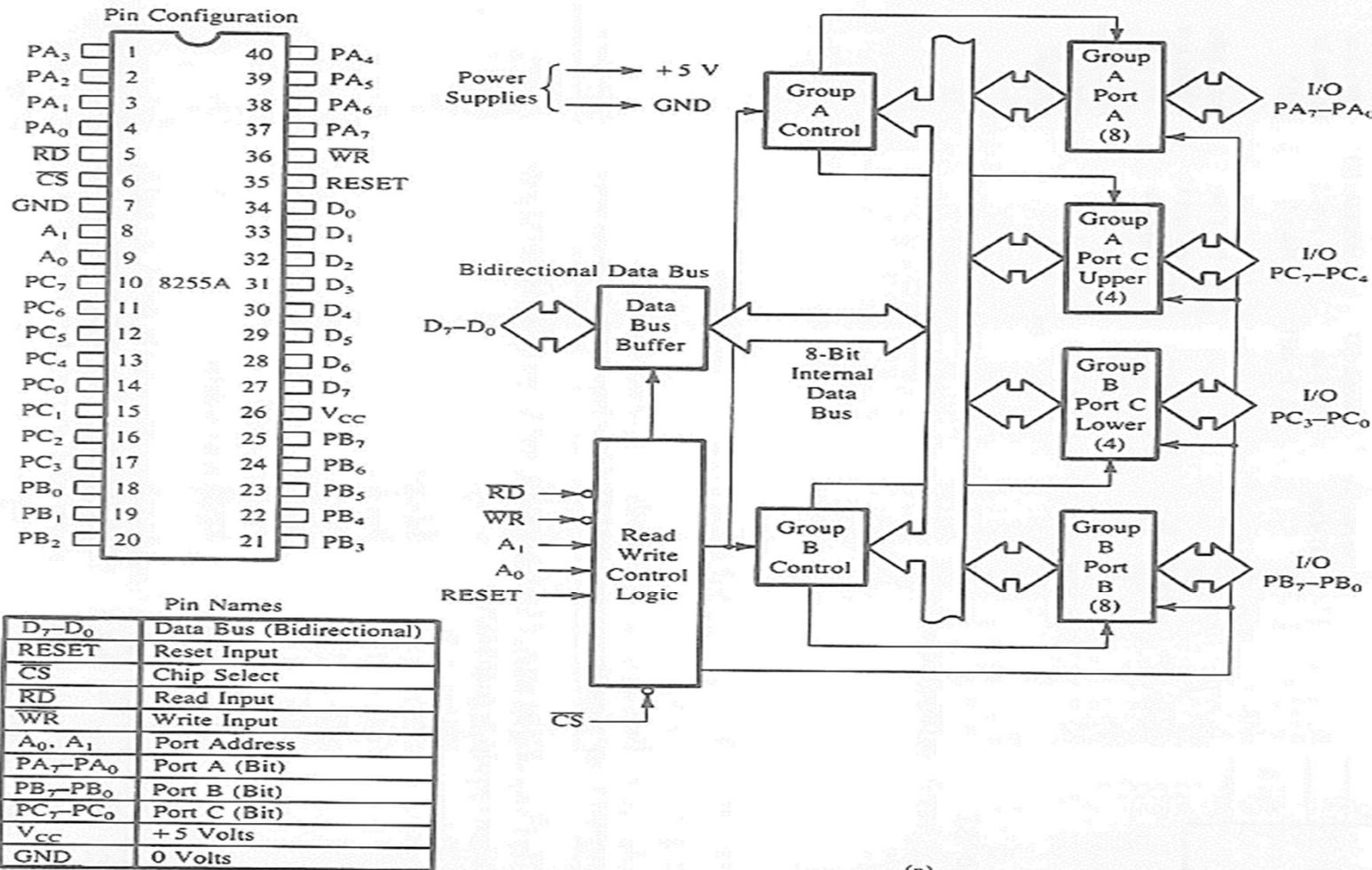
8255 – Programmable Peripheral Interface (PPI)



8255 - Programmable Peripheral Interface (PPI)

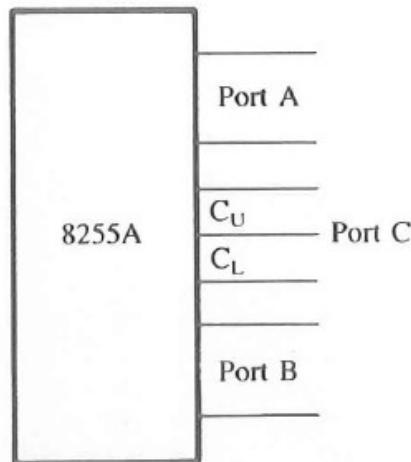
- 8255 is a general purpose programmable I/O device designed to interface the CPU with its outside world. We can program it according to the given condition. It can be used with almost any microprocessor.
- It consists of three 8-bit bidirectional I/O ports i.e. PORT A, PORT B and PORT C. We can assign different ports as input or output functions

8255 – Block diagram



(a)

8255 - Programmable Peripheral Interface (PPI)



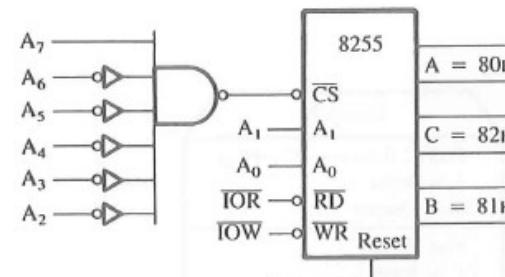
Group A

Port A + CU

Group B

Port B + CL

CS	A1	A0	Selects:
0	0	0	Port A
0	0	1	Port B
0	1	0	Port C
0	1	1	Control register
1	x	x	8255 is not selected



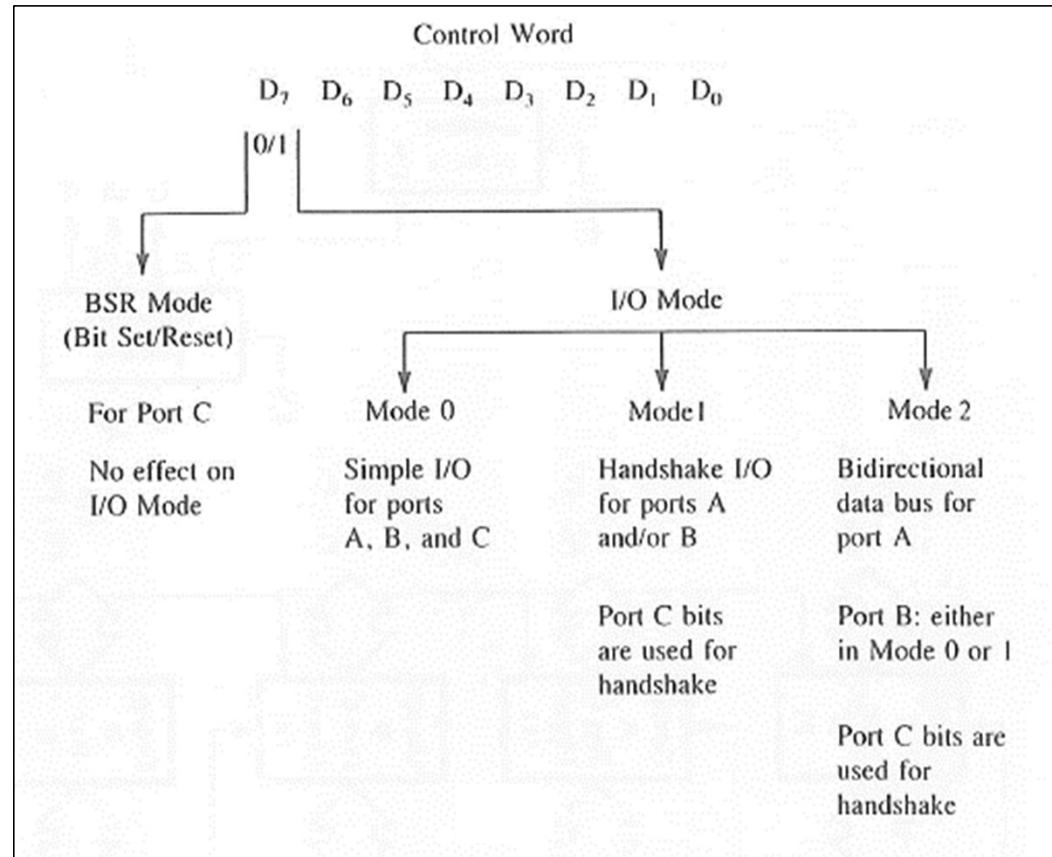
CS		Hex Address	Port
A ₇ A ₆ A ₅ A ₄ A ₃ A ₂	A ₁ A ₀	= 80H	A
1 0 0 0 0 0	0 0	= 81H	B
	0 1	= 82H	C
	1 0	= 83H	Control Register
	1 1		

8255 – Modes and control word

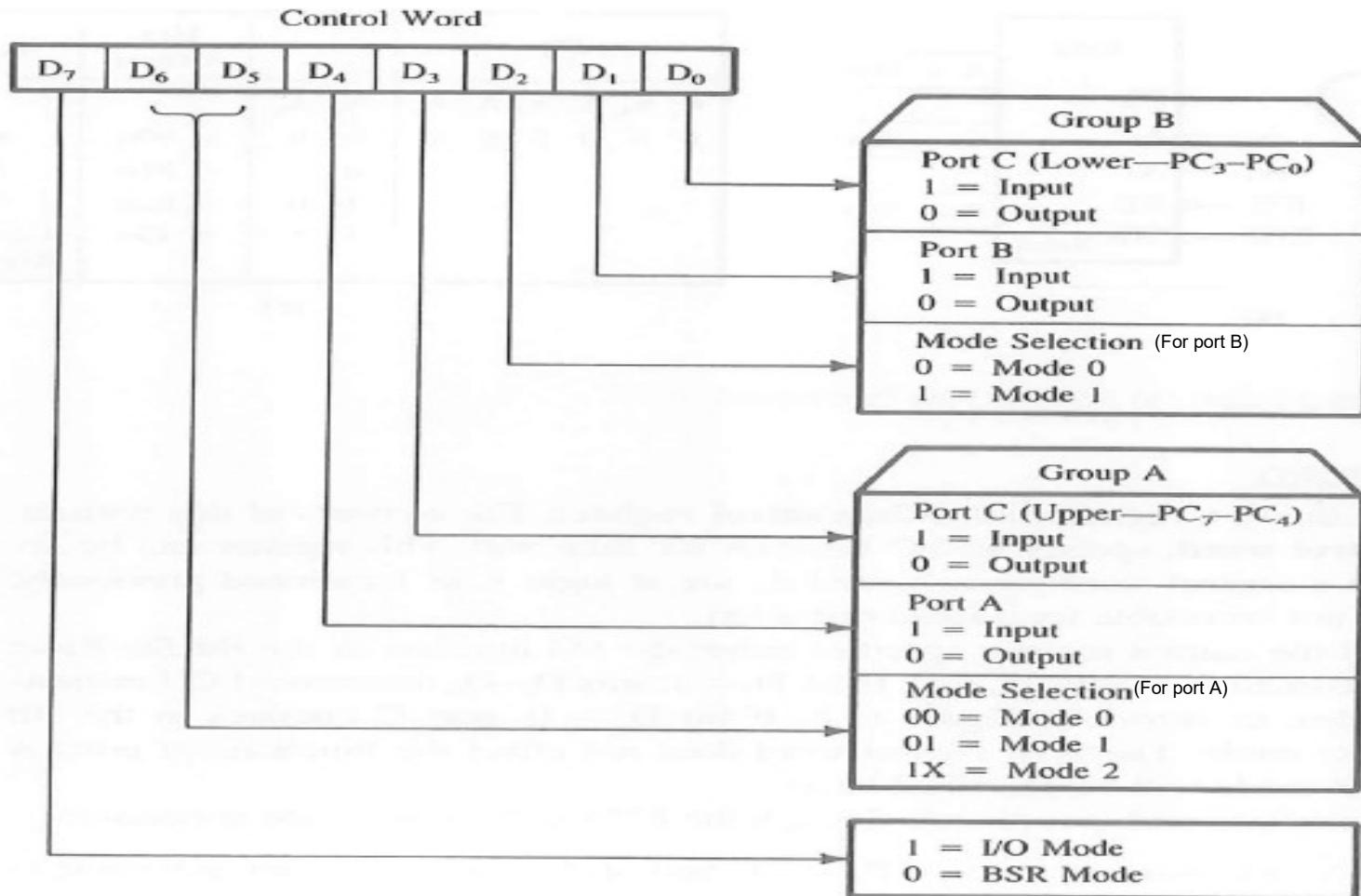
2 Modes

- **BSR mode** (Bit Set/Reset mode)
- **I/O mode**

- Control register controls the overall operation of 8255.
- Content of this register, called “**control word**”, specifies the operations



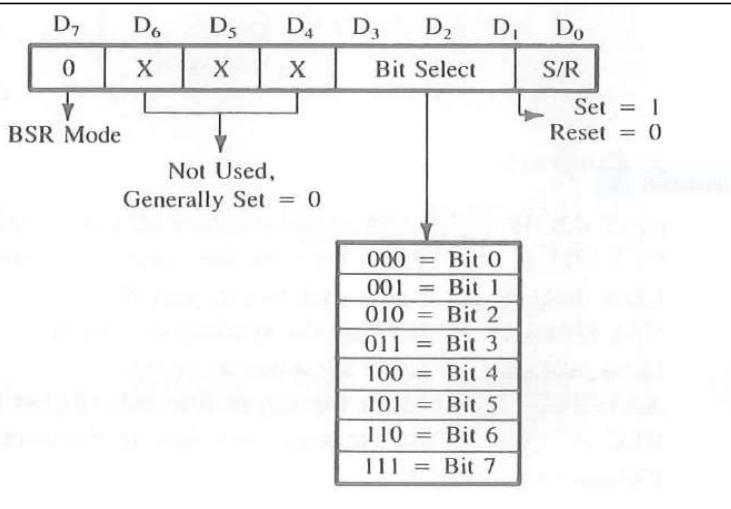
8255 – Control word format for I/O mode



8255 – BSR mode

The BSR mode is concerned only with the eight bits of port C, which can be set or reset by writing an appropriate control word in the control register. A control word with bit D₇ = 0 is recognized as a BSR control word, and it does not alter any previously transmitted control word with bit D₇ = 1; thus the I/O operations of ports A and B are not affected by a BSR control word. In the BSR mode, individual bits of port C can be used for applications such as an on/off switch.

BSR mode control word



BSR CONTROL WORDS

	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	
To set bit PC ₇	= 0	0	0	0	1	1	1	1	= 0FH
To reset bit PC ₇	= 0	0	0	0	1	1	1	0	= 0EH
To set bit PC ₃	= 0	0	0	0	0	1	1	1	= 07H
To reset bit PC ₃	= 0	0	0	0	0	1	1	0	= 06H

1. To set/reset bits in port C, a control word is written in the control register and not in port C.
2. A BSR control word affects only one bit in port C.
3. The BSR control word does not affect the I/O mode.

Instructions for
setting PC7
**MOV AL, 0F
OUT 83, AL**

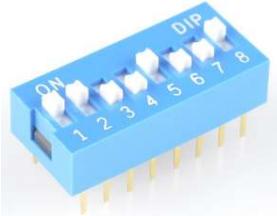


8255 – I/O Mode – Mode 0

- In this mode, ports A, B are used as **two simple 8-bit I/O ports** & port C as **two independent 4-bit ports**.
- **Each port** can be programmed to function as simply an input port or an output port.
- The **input/output features** in Mode 0 are as follows.
 1. Outputs are latched.
 2. Inputs are not latched.
 3. Ports don't have handshake or interrupt capability

8255 – I/O MODE – Mode 0 example

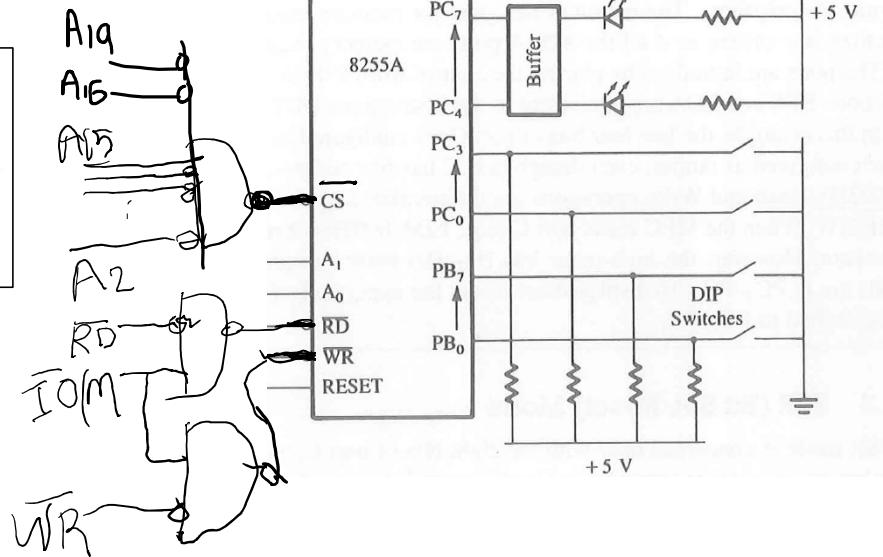
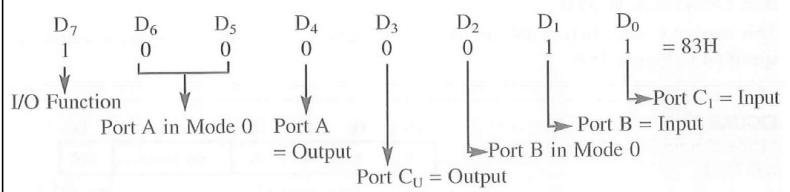
Write a program to read the DIP switches and display the reading from port B at port A and from port C_L at port C_U. (Use memory mapped I/O).



1. Port addresses:

Port A	= 08000H (A ₁ = 0, A ₀ = 0)
Port B	= 08001H (A ₁ = 0, A ₀ = 1)
Port C	= 08002H (A ₁ = 1, A ₀ = 0)
Control Register	= 08003H (A ₁ = 1, A ₀ = 1)

2. Control Word



```

MOV AL,83
MOV [8003],AL

MOV AL,[8001]
MOV [8000],AL

MOV AL,[8002]
AND AL,0F
MOV CL, 04
ROL AL,CL
MOV [8002],AL

```

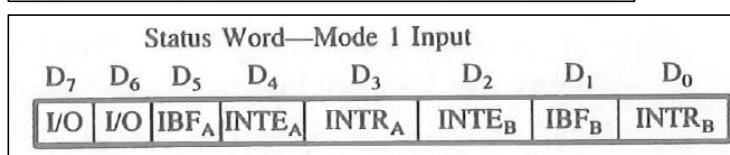
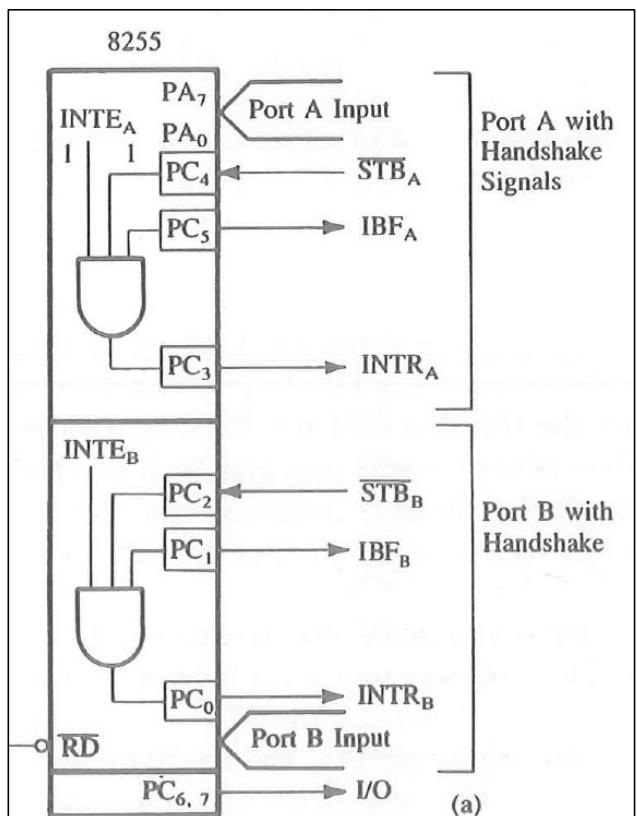
8255 – I/O Mode – Mode 1

In this mode, **handshake signals are exchanged** prior to data transfer.

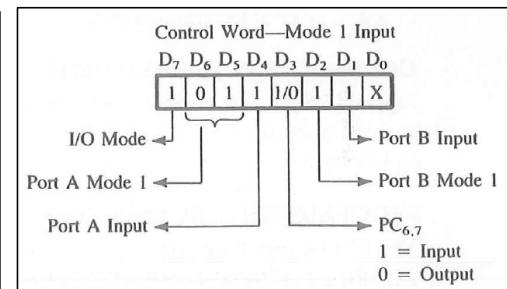
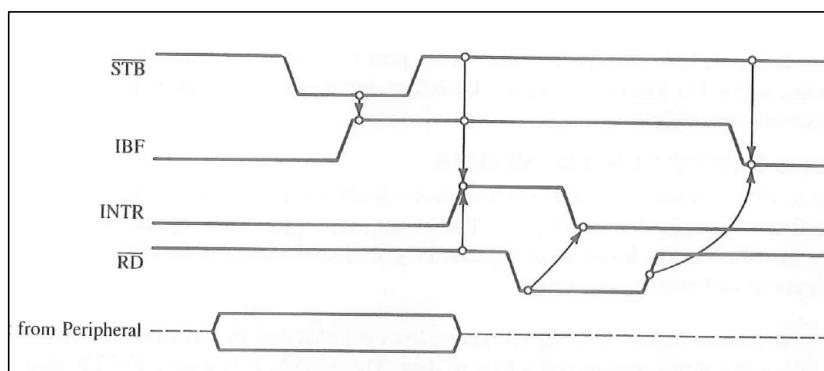
The **features** of the mode include the following:

1. Two ports (**A** and **B**) function as 8-bit I/O ports.
They can be configured as either as input or output ports.
2. Each port uses **three lines from port C as handshake signals**.
The remaining two lines of Port C can be used for simple I/O operations.
3. Input and Output data are latched.
4. Interrupt logic is supported.

8255 – I/O Mode – Mode 1 – Input control signals

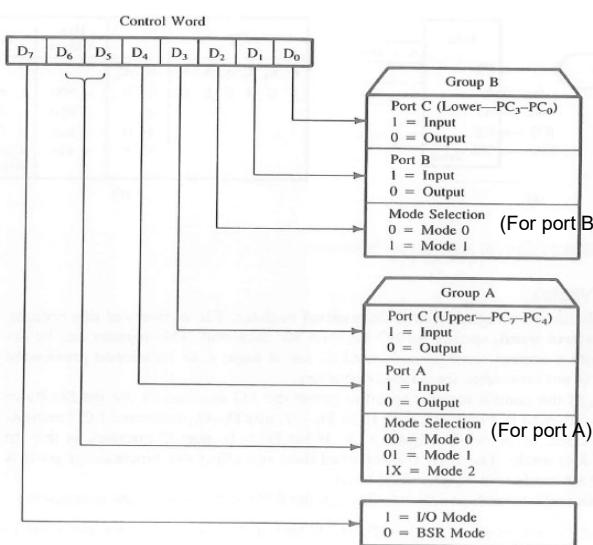
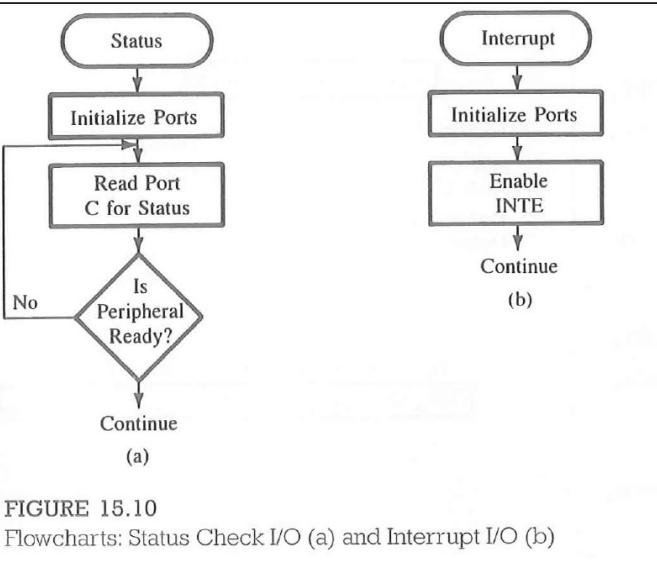


- **STB** (Strobe Input): This signal (active low) is generated by a peripheral device to indicate that it has transmitted a byte of data. The 8255A, in response to STB, generates IBF and INTR, as shown in Figure _____.
- **IBF (Input Buffer Full)**: This signal is an acknowledgment by the 8255A to indicate that the input latch has received the data byte. This is reset when the MPU reads the data
- **INTR (Interrupt Request)**: This is an output signal that may be used to interrupt the MPU. This signal is generated if STB, IBF, and INTE (Internal flip-flop) are all at logic 1. This is reset by the falling edge of the RD signal
- **INTE (Interrupt Enable)**: This is an internal flip-flop used to enable or disable the generation of the INTR signal. The two flip-flops INTE_A and INTE_B are set/reset using the BSR mode. The INTE_A is enabled or disabled through PC₄, and INTE_B is enabled or disabled through PC₂.



ssn

8255 – I/O Mode – Mode 1 – Programming



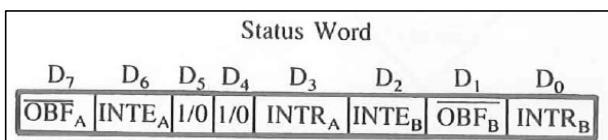
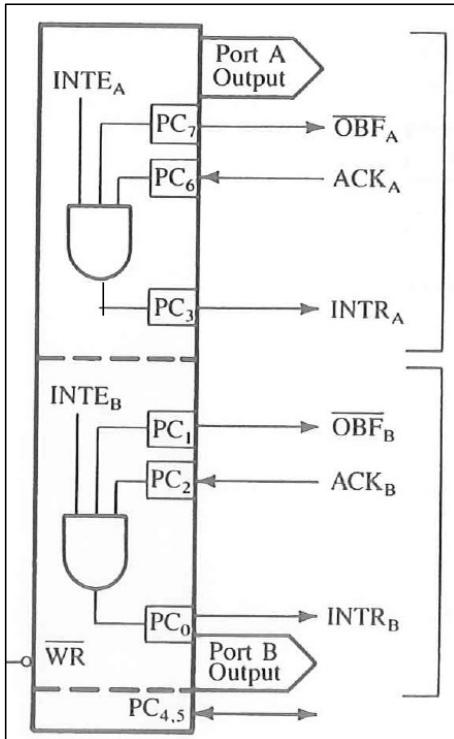
Eg: Read the input device connected in port A and display the data in port B. Configure port A in mode 1 and use status check I/O to communicate. Assume control register address=C6, port A address = C0, port B address = C2, port C address = C4

MOV AL, B4
OUT C6, AL

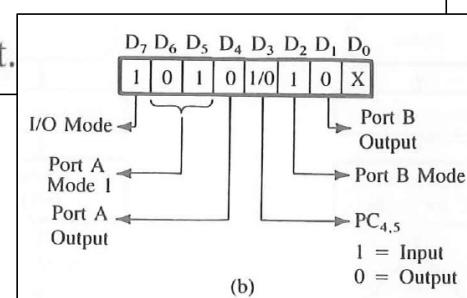
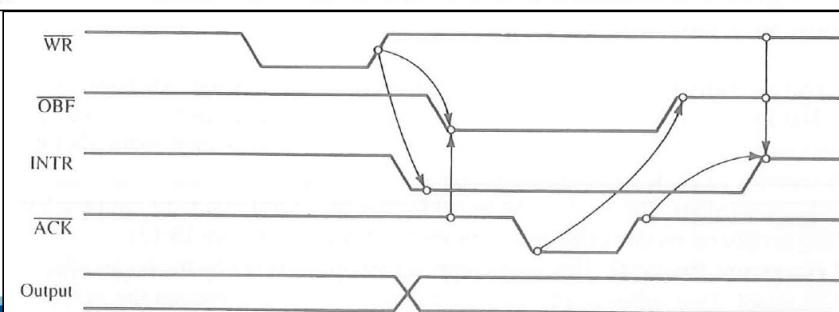
```
READ: IN AL, C4  
      AND AL, 20  
      JZ READ  
      IN AL,C0  
      OUT C2,AL  
      HLT
```



8255 – I/O Mode – Mode 1 – Output control signals

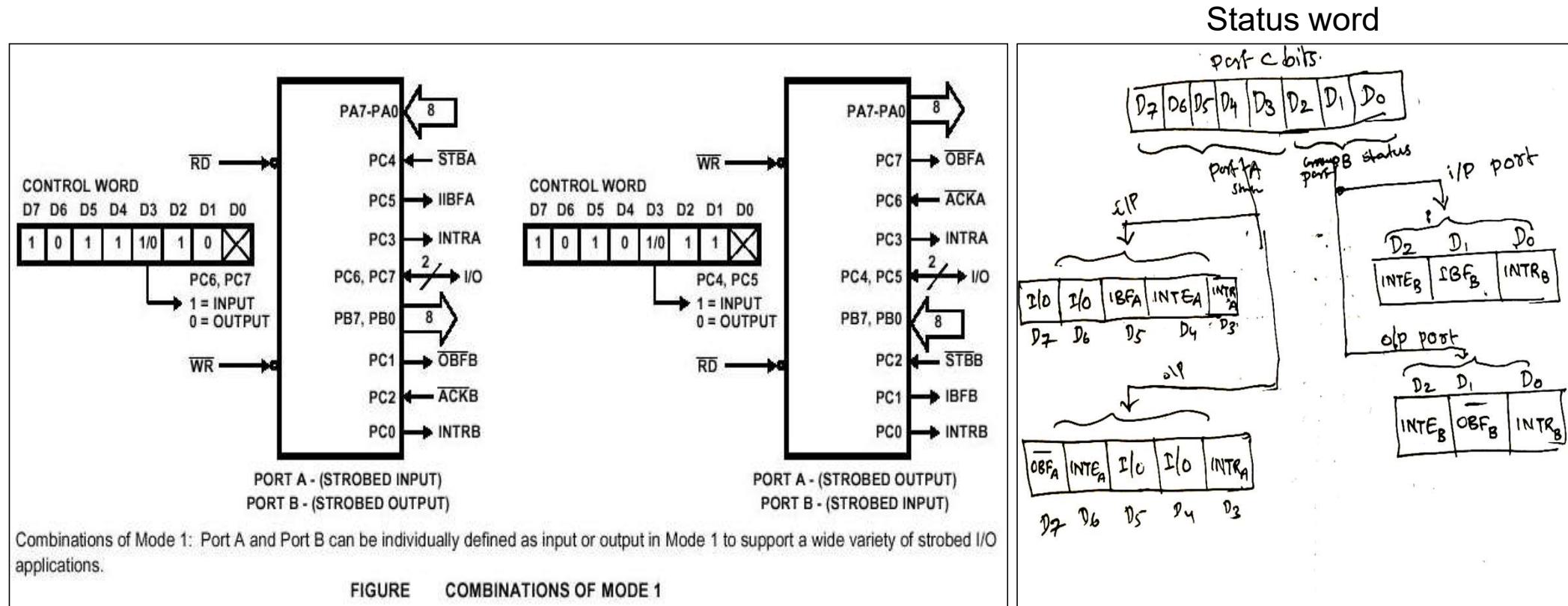


- OBF (Output Buffer Full):** This is an output signal that goes low when the MPU writes data into the output latch of the 8255A. This signal indicates to an output peripheral that new data are ready to be read (Figure). It goes high again after the 8255A receives an ACK from the peripheral.
- ACK (Acknowledge):** This is an input signal from a peripheral that must output a low when the peripheral receives the data from the 8255A ports (Figure).
- INTR (Interrupt Request):** This is an output signal, and it is set by the rising edge of the ACK signal. This signal can be used to interrupt the MPU to request the next data byte for output. The INTR is set when OBF, ACK, and INTE are all one (Figure) and reset by the falling edge of WR.
- INTE (Interrupt Enable):** This is an internal flip-flop to a port and needs to be set to generate the INTR signal. The two flip-flops INTE_A and INTE_B are controlled by bits PC₆ and PC₂, respectively, through the BSR mode.
- PC_{4,5}:** These two lines can be set up either as input or output.



ssn

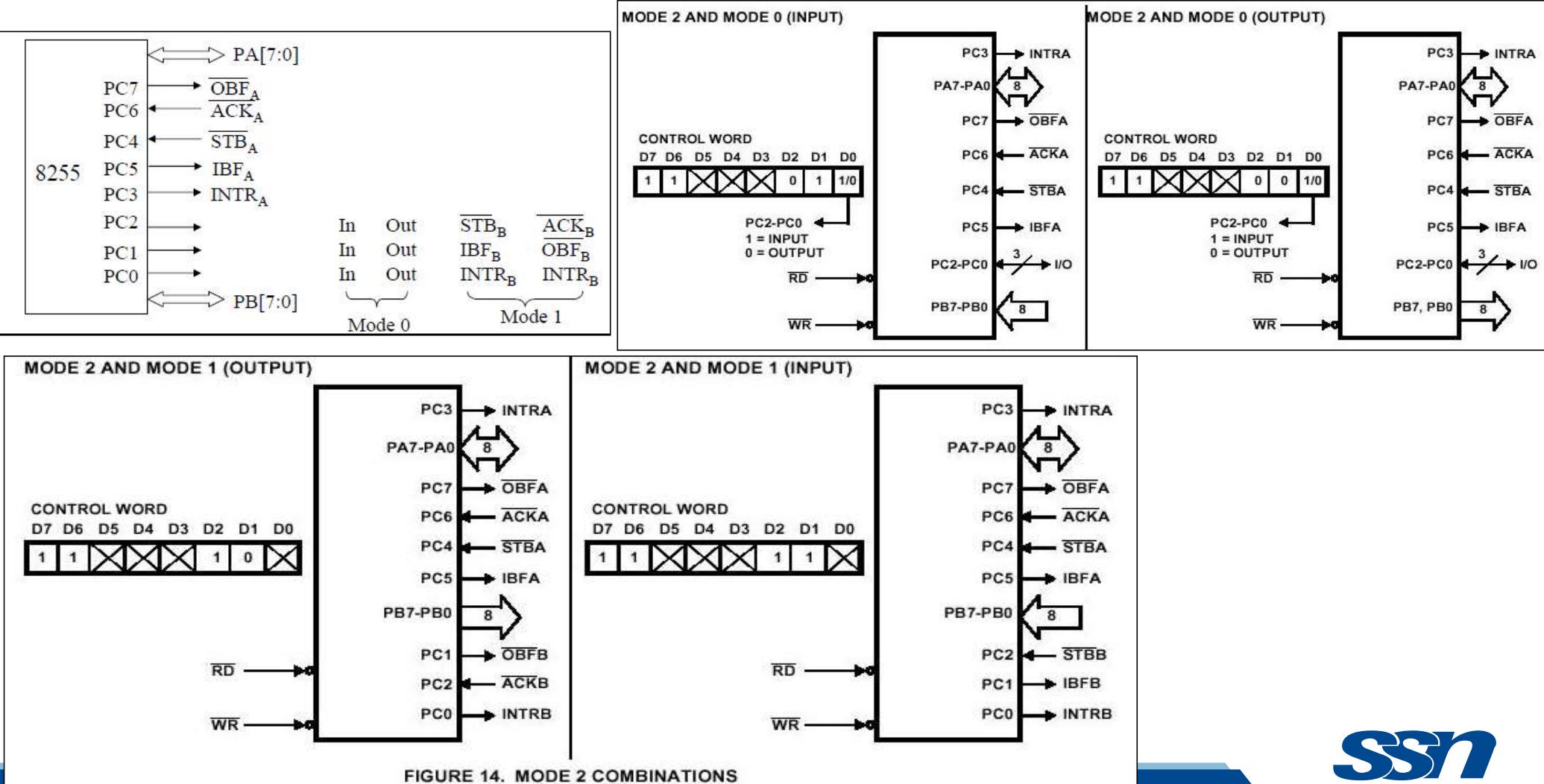
8255 – I/O Mode – Mode 1 – Input /Output combination



8255 – I/O Mode – Mode 2

- This mode is used primarily in applications such as **data transfer between two computers**.
- In this mode, **Port A** can be configured as the bidirectional port, **Port B** either in Mode 0 or Mode 1.
- **Port A** uses **five signals from Port C** as handshake signals for data transfer.
- The remaining three signals from **Port C** can be used either as simple I/O or as handshake for port B.

8255 - I/O Mode - Mode 2



ssn

Memory Interfacing

- Interface 2 chips of 16Kx8 EPROM and 2 chips of 32Kx8 RAM with 8086. Allocate higher address range to EPROM and lower address range to RAM.

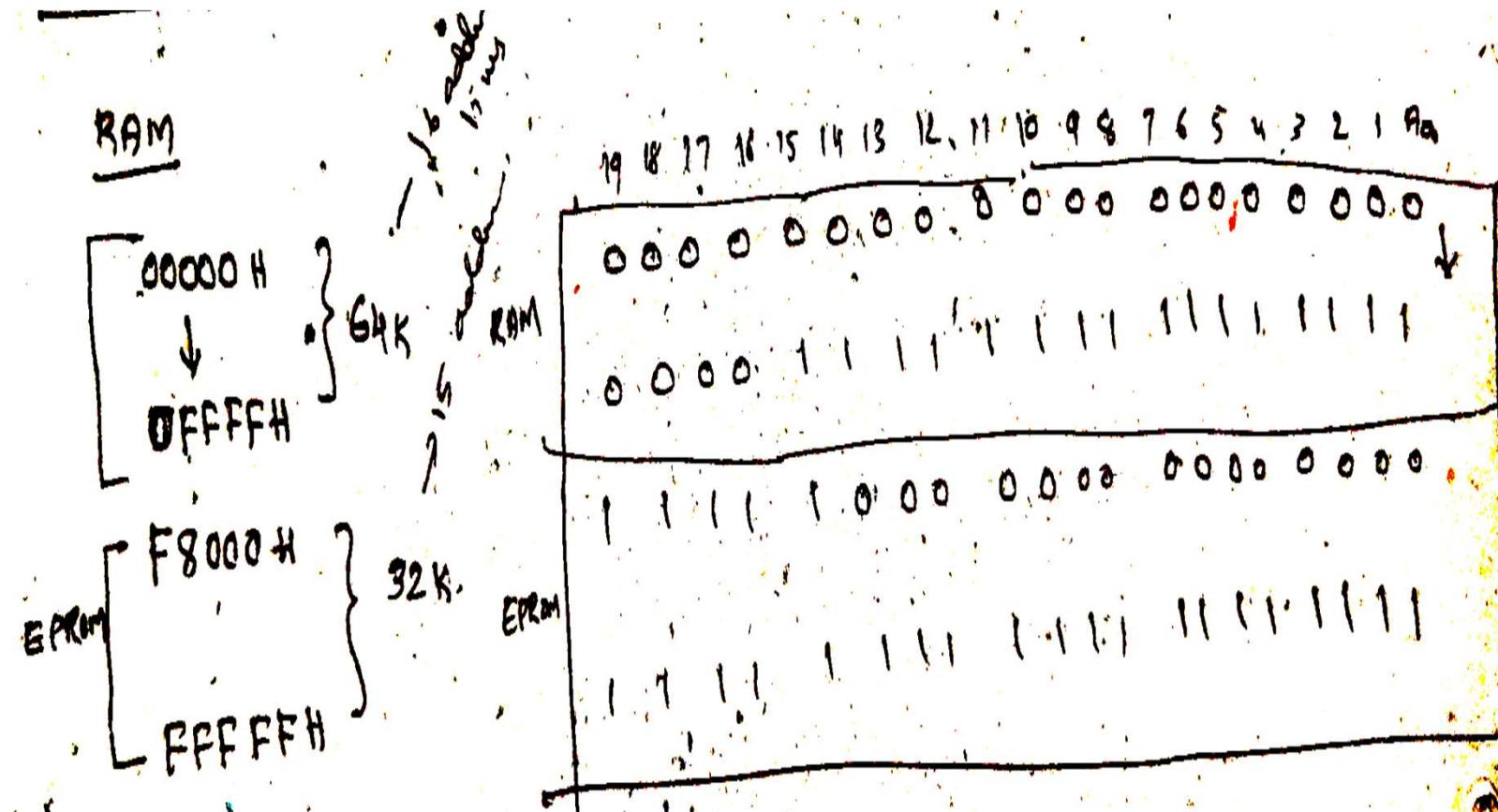
Memory Interfacing

- Interface 2 chips of 16Kx8 EPROM and 2 chips of 32Kx8 RAM with 8086. Allocate higher address range to EPROM and lower address range to RAM.
 - 2 chips of 16Kx8 EPROM
 - Total size = 32K
 - Asked to allocate 32K in the higher address range
 - So ???? To FFFF
 - Total 32K locations = $32 \times 1024 = 32768 = 8000H$
 - So to find ???? (starting address = HIGHEST LOCATION – SIZE + 1) ;
 - $FFFFF - 8000 + 1 = F8000$
 - **F8000 To FFFF**

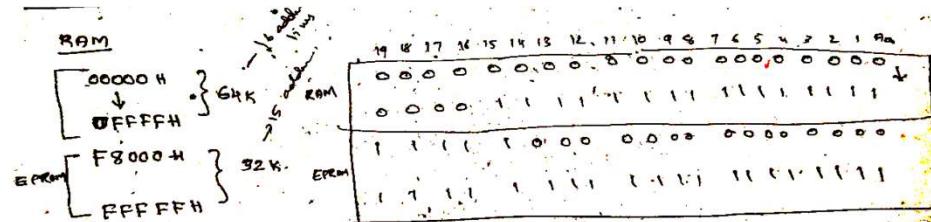
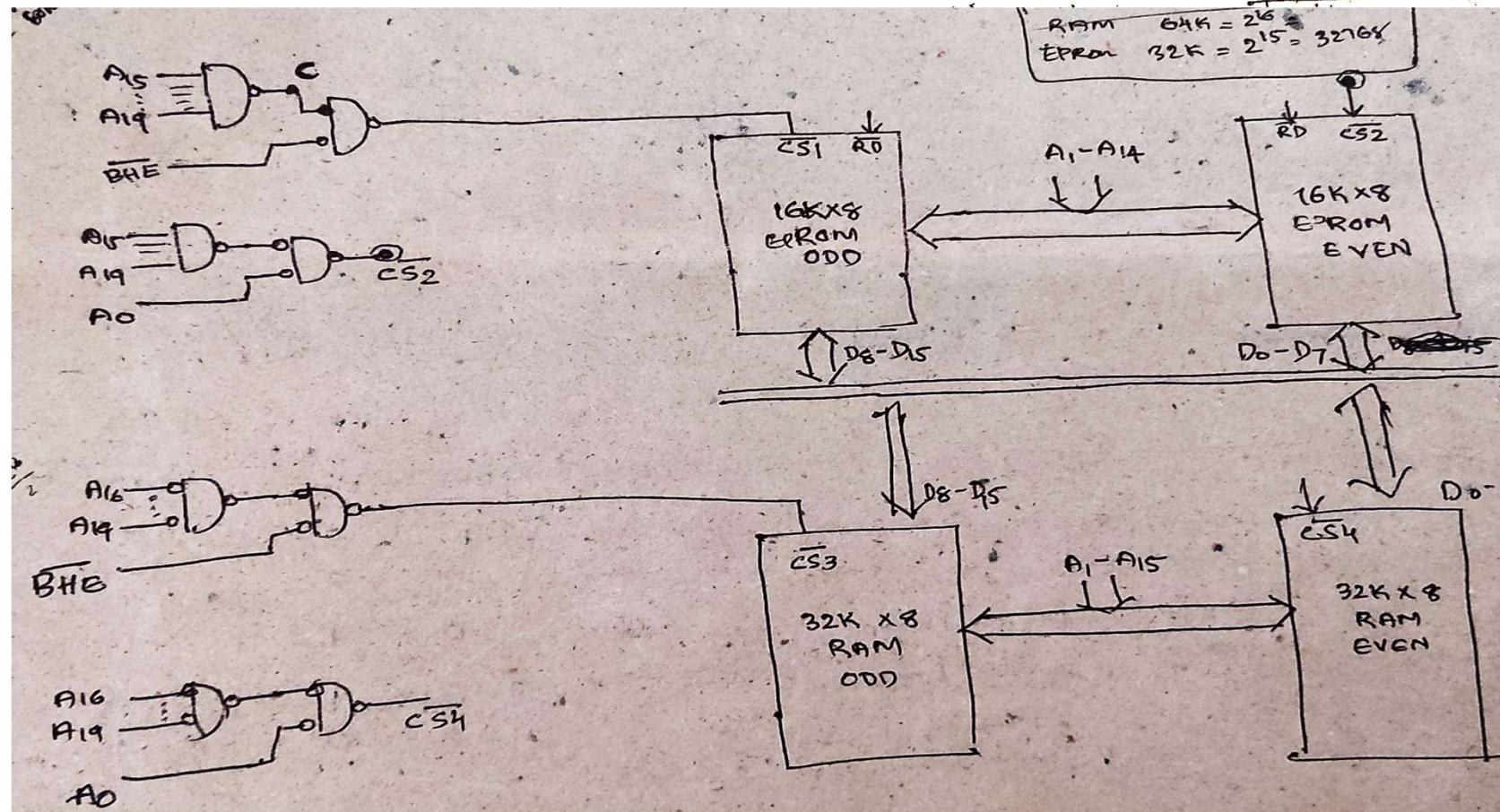
Memory Interfacing

- Interface 2 chips of 16Kx8 EPROM and 2 chips of 32Kx8 RAM with 8086. Allocate higher address range to EPROM and lower address range to RAM.
 - 2 chips of 32Kx8 RAM
 - Total size = 64K
 - Asked to allocate 64K in the lower address range
 - So 00000 To ?????
 - Total 64K locations = $64 \times 1024 = 65536 = 10000H$
 - So to find ????? (Ending address = Starting address + SIZE - 1) ;
 $00000 + 10000 - 1 = FFFF$
 - 00000 To 0FFFF

Memory Interfacing



Memory Interfacing



$$\text{RAM } 64K = 2^{16}$$

$$\text{EPROM } 32K = 2^{15} = 32768$$

$$16K = 16384$$

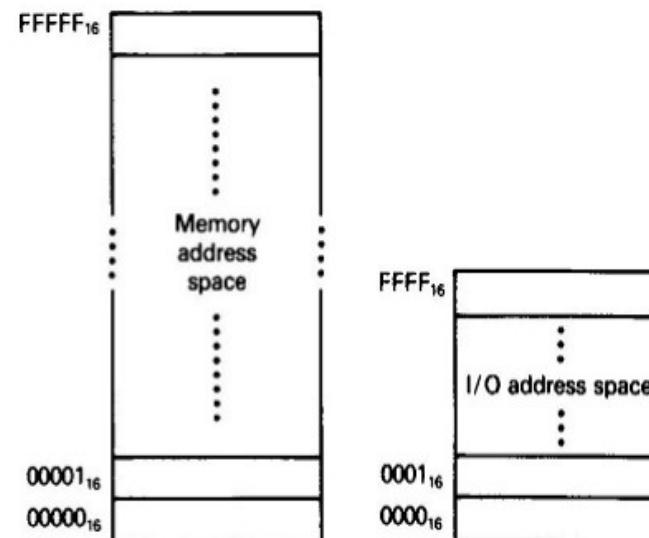
$$2^{14} = 16384$$

$$32K = 32768$$

$$2^{15} = 32768$$

I/O Interfacing

- 2 ways of interfacing
- Memory mapped I/O
- I/O mapped I/O (isolated I/O or peripheral mapped I/O)



I/O Interfacing

Memory mapped I/O

- Allocates memory addresses to Input-Output devices.
- I/O devices are treated as memory locations.
- Any instructions related to memory can be used for accessing the Input-Output device.
- The Input-Output device data are also given to the Arithmetic Logical Unit.

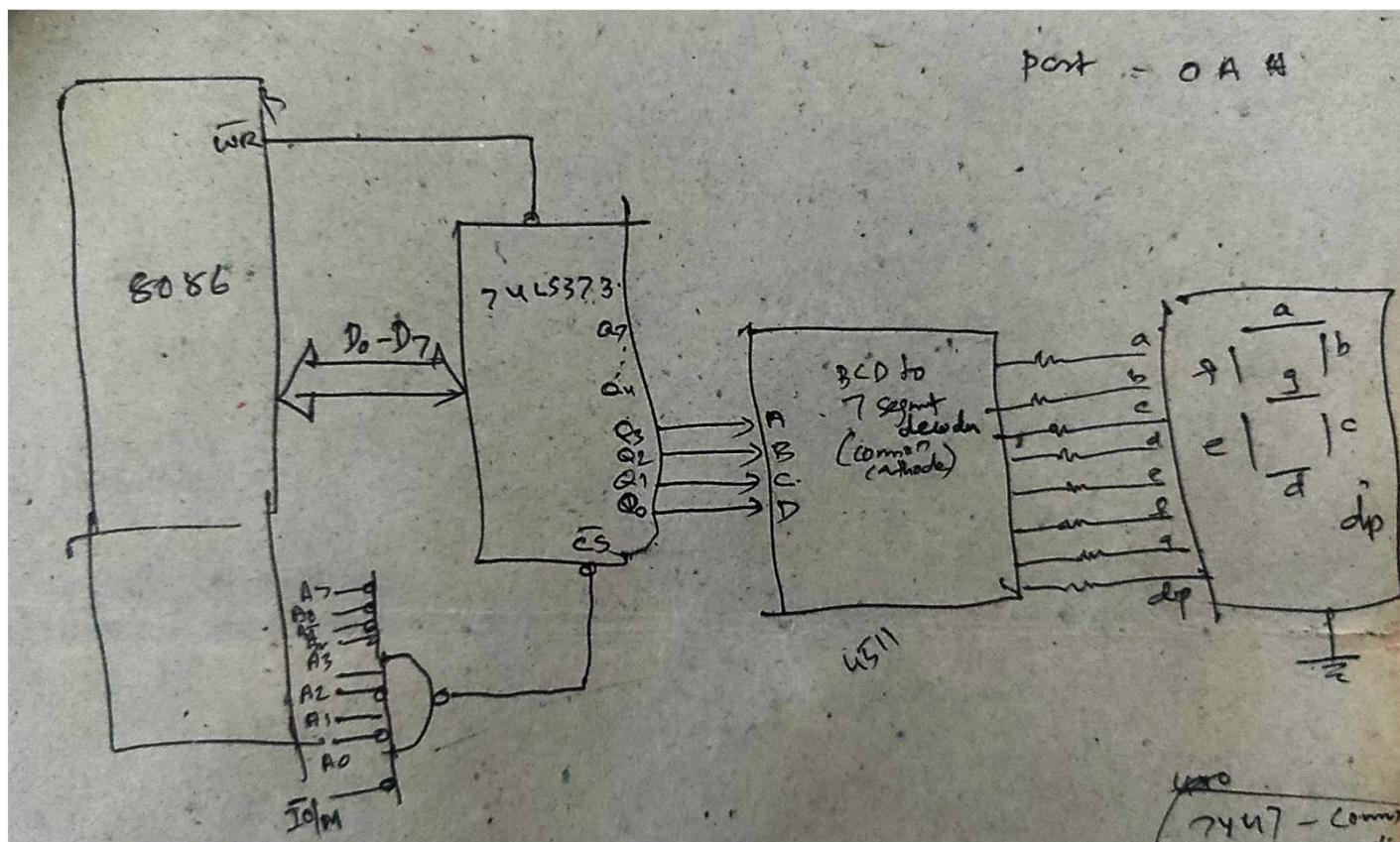
I/O mapped I/O

- Allocates separate I/O address ranges to Input-Output devices
- Only IN and OUT instructions can be used for accessing such devices.
- The ALU operations are not directly applicable to Input-Output data.

I/O Interfacing

- Design an 8086 interface and write ALP for displaying the count from 0 to 9 continuously in a 7 segment LED display. Select the port address suitably. Use I/O mapped I/O for interfacing.

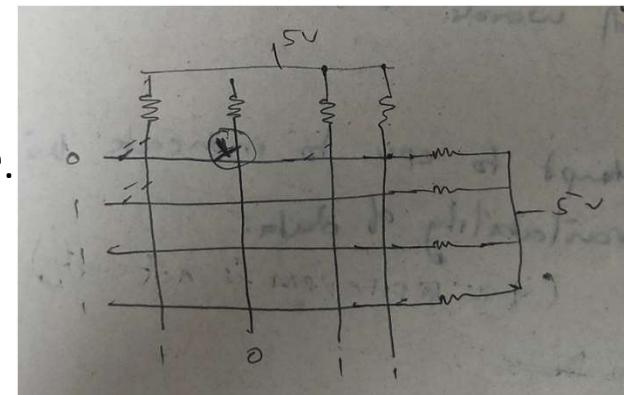
I/O Interfacing



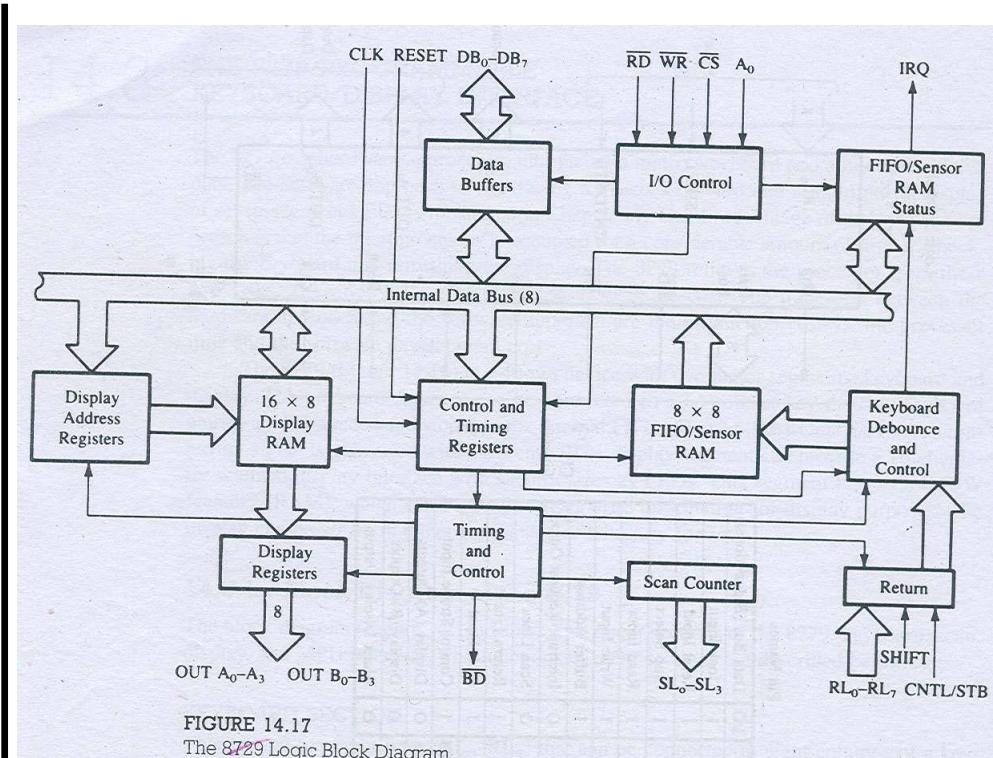
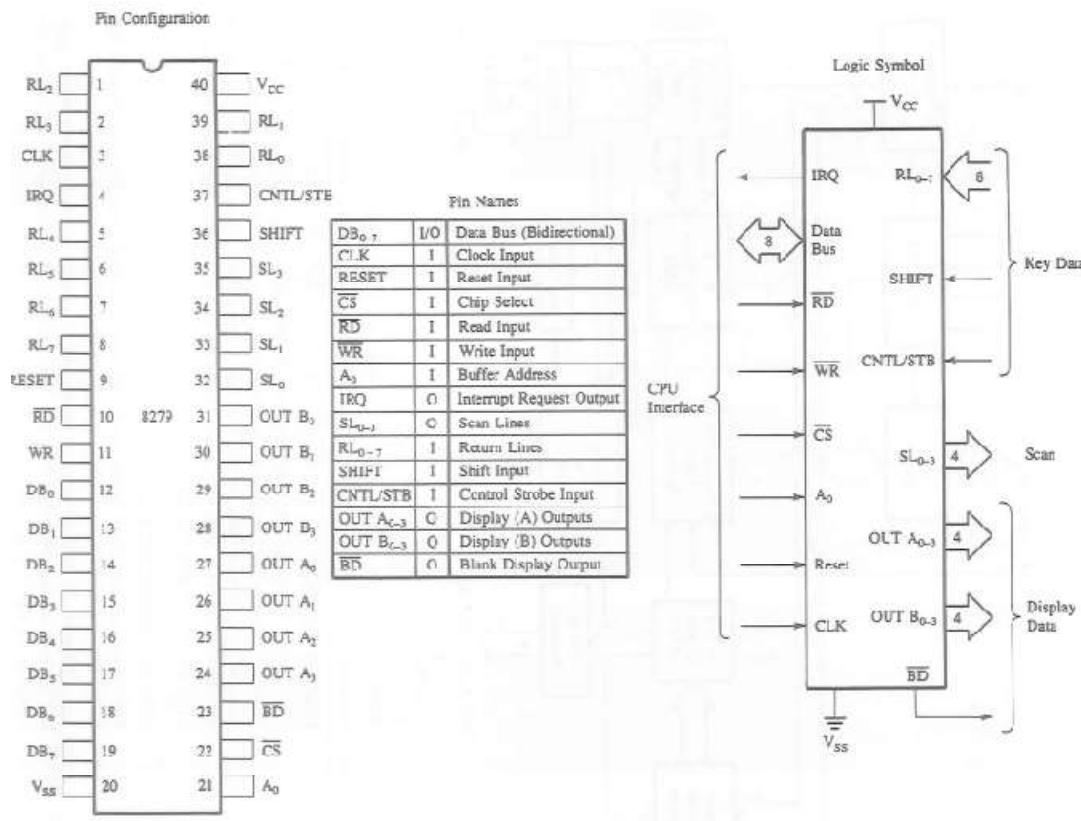
LABEL1: MOV AL,00
LABEL2:OUT 0AH,AL
CALL DELAY
INC AL
CMP AL, 0A
JZ LABEL1
JMP LABEL2

8279: Key Board and Display controller

- Scans and encodes up to a 64-key matrix keyboard.
- Controls up to a 16 seven segment LEDs .
- Keyboard has a built-in FIFO 8 character buffer to store the keyboard entries
- The display is controlled from an internal 16x8 RAM that stores the coded display information.
- Display can be setup in left entry or right entry mode.



Features of 8279



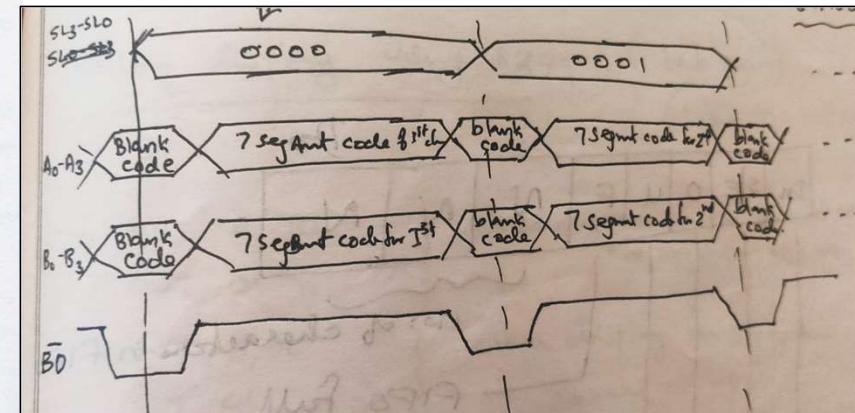
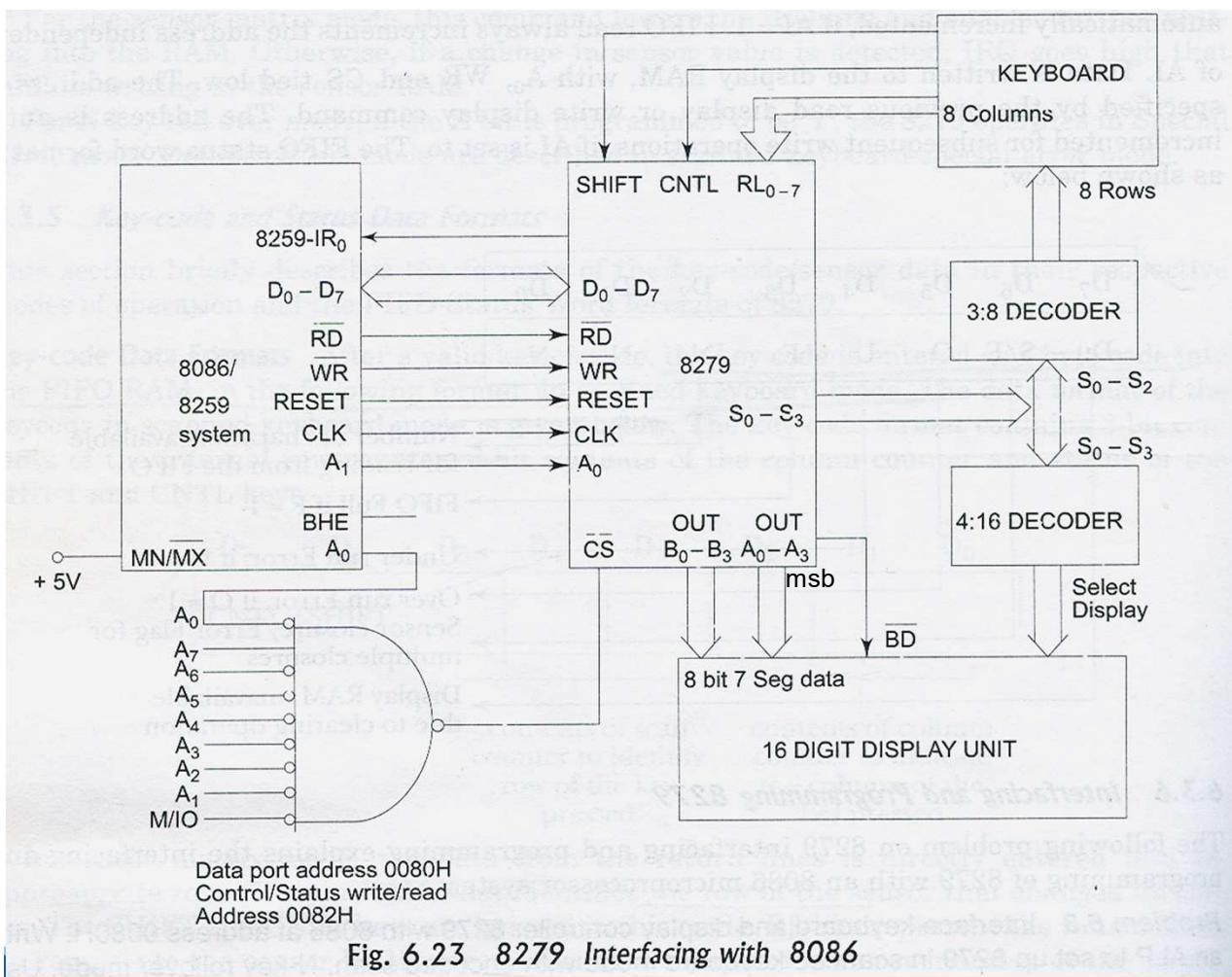
Pin Details

- A0 : Selects data (0) or control/status (1) for reads and writes between microprocessor and 8279.
- BD : Output that blanks the displays.
- CLK : Used internally for timing. Maximum is 3 MHz
- CNTL/STB : Control/strobe, connected to the control key on the keyboard.
- CS : Chip select that enables programming, reading the keyboard, etc.
- DB7-DB0 : Consists of bidirectional pins that connect to data bus on microprocessor.

Pin Details

- IRQ : Interrupt request to mp, to indicate the availability of data (if the FIFO RAM is not empty).
- OUT A3-A0/B3-B0 : Outputs that sends data to the most significant/least significant nibble of display.
- RD(WR) : Connects to micro's RD/WR signal
- RESET : Connects to system RESET.
- RL7-RL0 : Return lines are inputs used to sense key press in the keyboard matrix.
- Shift : Shift connects to Shift key on keyboard.
- SL3-SL0 : Scan line outputs scan both the keyboard and displays.

Interfacing with 8086



Input Modes

- Scanned Keyboard—with encoded (8 x 8 key keyboard) or decoded (4 x 8 key keyboard) scan lines. A key depression generates a 6-bit encoding of key position. Position and shift and control status are stored in the FIFO. Keys are automatically debounced with 2-key lockout or N-key roll-over.
- Scanned Sensor Matrix—with encoded (8 x 8 matrix switches) or decoded (4 x 8 matrix switches) scan lines. Key status (open or closed) stored in RAM addressable by CPU.
- Strobed Input—Data on return lines during control line strobe is transferred to FIFO.

Output Modes

- 8 or 16 character multiplexed displays that can be organized as dual 4-bit or single 8-bit ($B_0 = D_0, A_3 = D_7$).
- Right entry or left entry display formats.

Encoded mode: SL0-SL3 gives binary count. External decoder (7445) need to be used.

Decoded mode: 8279 outputs stepping 0s on 4 scan lines. No need of external decoder.

In sensor mode, if any switch changes, it sends IRQ.

In strobed input mode, RL0-RL7 can be acted as strobed input port as in 8255. Data is entered by the rising edge of CNTL/STB line

Data Format

In the Scanned Keyboard mode, the character entered into the FIFO corresponds to the position of the switch in the keyboard plus the status of the CNTL and SHIFT lines (non-inverted). CNTL is the MSB of the character and SHIFT is the next most significant bit. The next three bits are from the scan counter and indicate the row the key was found in. The last three bits are from the column counter and indicate to which return line the key was connected.

MSB	LSB
CNTL SHIFT	SCAN . RETURN

SCANNED KEYBOARD DATA FORMAT

Block Diagram

Mainly 4 Sections:

- 1. KeyBoard Section**
- 2. Scan Section**
- 3. Display Section**
- 4. µp Interface Section**

Scan section

- It has a Scan Counter and 4 Scan Lines
- Connected to 4 – to – 16 Decoder to generate 16 scan lines
- Scan lines can be connected to rows of Matrix Keyboard and Digit drivers of Display

Display section

- 8 Output lines are divided into 2 groups:
A3 – A0 and B3 – B0
- Can be used as 8 or 4/4
- BD line for display Blanking
- 16 X 8 Display RAM

Keyboard section

- RL0 – RL7 connected to 8 columns of keyboard
- 2 modes: **2-key lockout**
 - if 2 keys are pressed simultaneously, only first key is recognized. One key must be released before next keypress.
- N key Rollover-**
 - FIFO RAM with 8 registers to store 8 keyboard entries and each read in the order of their entries.
 - Sent IRQ signal when FIFO is not empty

µp Interface Section

- Data bus : DB0 – DB7
- One IRQ Line
- Six interface lines and A0
- A0 = 1 ; signals are control/status word

8279 COMMAND WORDS - (AO = 1)

- 1. Keyboard/Display Mode
- 2. Program Clock
- 3. Read FIFO/Sensor RAM
- 4. Read Display RAM
- 5. Write Display RAM
- 6. Display write inhibit /Blanking
- 7. Clear
- 8. End Interrupt/Error mode set

D7	D6	D5	Function	Purpose
0	0	0	Mode set	Selects the number of display positions, type of key scan...
0	0	1	Clock	Programs internal clk, sets scan and debounce times.
0	1	0	Read FIFO	Selects type of FIFO read and address of the read.
0	1	1	Read Display	Selects type of display read and address of the read.
1	0	0	Write Display	Selects type of write and the address of the write.
1	0	1	Display write inhibit	Allows half-bytes to be blanked.
1	1	0	Clear	Clears the display or FIFO
1	1	1	End interrupt	Clears the IRQ signal to the microprocessor.

Keyboard/Display Mode Set

	MSB		LSB
Code:	0	0	D D K K K
	0	0	D D K K K

Where DD is the Display Mode and KKK is the Keyboard Mode.

DD

- 0 0 8 8-bit character display—Left entry
- 0 1 16 8-bit character display—Left entry*
- 1 0 8 8-bit character display—Right entry
- 1 1 16 8-bit character display—Right entry

For description of right and left entry, see Interface Considerations. Note that when decoded scan is set in keyboard mode, the display is reduced to 4 characters independent of display mode set.

KKK

- 0 0 0 Encoded Scan Keyboard—2 Key Lockout*
- 0 0 1 Decoded Scan Keyboard—2-Key Lockout
- 0 1 0 Encoded Scan Keyboard—N-Key Roll-over
- 0 1 1 Decoded Scan Keyboard—N-Key Roll-over
- 1 0 0 Encoded Scan Sensor Matrix
- 1 0 1 Decoded Scan Sensor Matrix
- 1 1 0 Strobed Input, Encoded Display Scan
- 1 1 1 Strobed Input, Decoded Display Scan

Program Clock

Code:	0	0	1	P	P	P	P	P
-------	---	---	---	---	---	---	---	---

All timing and multiplexing signals for the 8270 are generated by an internal prescaler. This prescaler divides the external clock (pin 3) by a programmable integer. Bits PPPPP determine the value of this integer which ranges from 2 to 31.

Operating frequency of 8279 =
I/P clk frequency/Integer represented by PPPPP

I/P f= 2 MHz

To operate on 100KHz,
PPPPP??

$2M/100K = 20 = 10100$
PPPPP = 10100

Read FIFO/Sensor RAM

Code:

0	1	0	AI	X	A	A	A
---	---	---	----	---	---	---	---

 X = Don't Care

The CPU sets the 8279 for a read of the FIFO/Sensor RAM by first writing this command. In the Scan Keyboard Mode, the Auto-Increment flag (AI) and the RAM address bits (AAA) are irrelevant. The 8279 will automatically drive the data bus for each subsequent read ($A_0 = 0$) in the same sequence in which the data first entered the FIFO. All subsequent reads will be from the FIFO until another command is issued.

In the Sensor Matrix Mode, the RAM address bits AAA select one of the 8 rows of the Sensor RAM. If the AI flag is set ($AI = 1$), each successive read will be from the subsequent row of the sensor RAM.

Read Display RAM

Code:

0	1	1	AI	A	A	A	A
---	---	---	----	---	---	---	---

The CPU sets up the 8279 for a read of the Display RAM by first writing this command. The address bits AAAA select one of the 16 rows of the Display RAM. If the AI flag is set ($A1 = 1$), this row address will be incremented after each following read or write to the Display RAM. Since the same counter is used for both reading and writing, this command sets the next read or write address and the sense of the Auto-Increment mode for both operations.

Write Display RAM

Code:

1	0	0	AI	A	A	A	A
---	---	---	----	---	---	---	---

The CPU sets up the 8279 for a write to the Display RAM by first writing this command. After writing the command with $A_0 = 1$, all subsequent writes with $A_0 = 0$ will be to the Display RAM. The addressing and Auto-Increment functions are identical to those for the Read Display RAM.

Display Write Inhibit/Blanking

	A	B	A	B
Code:	1	0	1	X

The IW Bits can be used to mask nibble A and nibble B in applications requiring separate 4-bit display ports. By setting the IW flag (IW = 1) for one of the ports, the port becomes marked so that entries to the Display RAM from the CPU do not affect that port.

If the user wishes to blank the display, the BL flags are available for each nibble. The last Clear command issued determines the code to be used as a "blank." This code defaults to all zeros after a reset. Note that both BL flags must be set to blank a display formatted with a single 8-bit port.

Clear

Code:	1	1	0	C _D	C _D	C _D	C _F	C _A
-------	---	---	---	----------------	----------------	----------------	----------------	----------------

The C_D bits are available in this command to clear all rows of the Display RAM to a selectable blanking code as follows:

C _D	C _D	C _D		
0	X		All Zeros (X = Don't Care)	
1	0		Hex 20 (0010 0000)	
1	1		All Ones	

Enable clear display when = 1 (or by C_A = 1)

If the C_F bit is asserted (C_F = 1), the FIFO status is cleared and the interrupt output line is reset. Also, the Sensor RAM pointer is set to row 0.

C_A, the Clear All bit, has the combined effect of C_D and C_F; it uses the C_D clearing code on the Display RAM and also clears FIFO status. Furthermore, it resynchronizes the internal timing chain.

End Interrupt/Error Mode Set

Code:

1	1	1	E	X	X	X	X
---	---	---	---	---	---	---	---

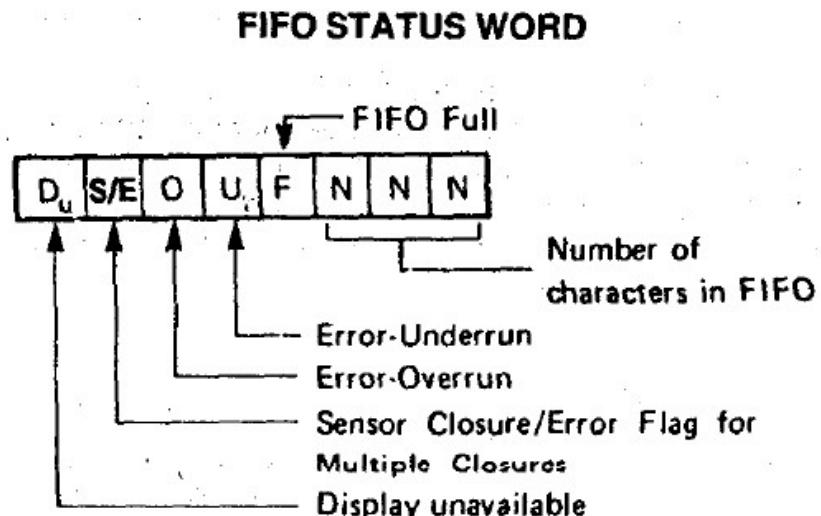
 X = Don't care

For the sensor matrix modes this command lowers the IRQ line and enables further writing into RAM.

For the N-key rollover mode—if the E bit is programmed to “1” the chip will operate in the special Error mode.

Scanned Keyboard—Special Error Modes

For N-key rollover mode the user can program a special error mode. This is done by the “End Interrupt/Error Mode Set” command. The debounce cycle and key-validity check are as in normal N-key mode. If during a *single debounce cycle*, two keys are found depressed, this is considered a simultaneous multiple depression, and sets an error flag. This flag will prevent any further writing into the FIFO and will set interrupt (if not yet set). The error flag could be read in this mode by reading the FIFO STATUS word. (See “FIFO STATUS” for further details.) The error flag is reset by sending the normal CLEAR command with CF = 1.



There are two types of errors possible: overrun and underrun. Overrun occurs when the entry of another character into a full FIFO is attempted. Underrun occurs when the CPU tries to read an empty FIFO.

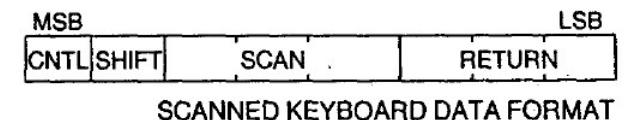
Write instructions to initialize 8279 in 2-key lock out , encoded mode, 8 bit, 8 character display, left entry mode. Write a subroutine which reads character data from keyboard (port 1900H) and stores it in the system memory. Assume the control register address is 1901H.

Main program

```
MOV AL,00  
MOV [1901], AL
```


Sub-routine

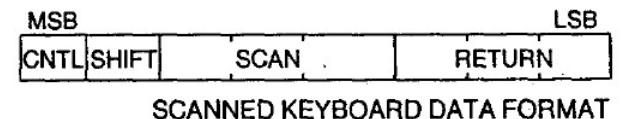
```
PUSH SI  
PUSH AX  
PUSHF  
  
MOV SI, 1901  
MOV [1901], 40 ; read FIFO RAM control word  
DEC SI  
MOV AL, [SI]  
AND AL, 3F  
MOV [2500], AL  
  
POPF  
POP AX  
POP SI
```



Write instructions to read character data from keyboard and stores it in the system memory. Use polling method to communicate with 8279. Assume control register address = C2, Data register address = C0.

```
MOV BX,1100
```

```
Loop1: IN AL,C2 ; Reading status register  
        TEST AL, 07  
        JZ LOOP1  
        MOV AL, 40 ; read FIFO RAM control word  
        OUT C2, AL  
        IN AL, C0  
        MOV [BX], AL  
        HLT
```



8254: Timer Features

Features of 8254

- The Intel 8253 and 8254 are Programmable Interval Timers designed for microprocessors to perform timing and counting functions using three 16-bit down counters.
- It can be used for applications such as real-time clock, event counter etc.
- Each counter has 2 input pins, i.e. Clock & Gate, and 1 pin for “OUT” output.
- To operate a counter, a 16-bit count is loaded in its register.
- On command, it begins to decrement the count until it reaches 0, then it generates a pulse that can be used to interrupt the CPU.
- These three counters can be programmed for either binary or BCD count.

Difference between 8254 and 8253

The 8254 is an upgraded version of the 8253, and they are pin-compatible. The features of these two devices are almost identical except that

- the 8254 can operate with higher clock frequency range (DC to 8 MHz and 10 MHz for 8254-2), and the 8253 can operate with clock frequency from DC to 2 MHz.
- the 8254 includes a Status Read-Back Command that can latch the count and the status of the counters.

8254 Block Diagram

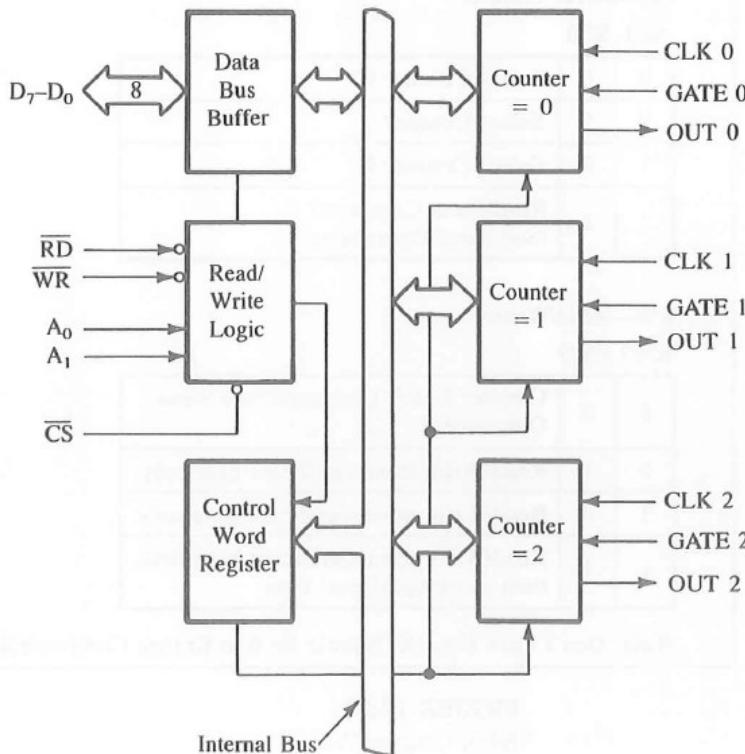
Pin Configuration

D ₇	1	24	V _{CC}
D ₆	2	23	WR
D ₅	3	22	RD
D ₄	4	21	CS
D ₃	5	20	A ₁
D ₂	6	19	A ₀
8254	18	19	CLK 2
D ₁	7	18	OUT 2
D ₀	8	17	GATE 2
CLK 0	9	16	CLK 1
OUT 0	10	15	GATE 1
GATE 0	11	14	GATE 1
GND	12	13	OUT 1

Pin Names

D ₇ -D ₀	Data Bus (8 Bit)
CLK N	Counter Clock Inputs
GATE N	Counter Gate Inputs
OUT N	Counter Outputs
RD	Read Counter
WR	Write Command or Data
CS	Chip Select
A ₀ -A ₁	Counter Select
V _{CC}	+5 Volts
GND	Ground

Block Diagram



- **Clock:** This is the clock input for the counter.
- **Out:** This single output line is the signal that is the final programmed output of the device. Actual operation of the out line depends on how the device has been programmed.
- **Gate:** This input can act as a gate for the clock input line, or it can act as a start pulse, depending on the programmed mode of the counter.

Counter Selection

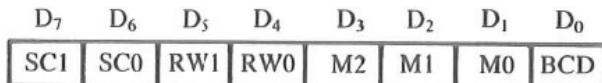
The control section has five signals: \overline{RD} (Read), \overline{WR} (Write), \overline{CS} (Chip Select), and the address lines A_0 and A_1 . In the peripheral I/O mode, the \overline{RD} and \overline{WR} signals are connected to IOR and IOW , respectively. In memory-mapped I/O, these are connected to $MEMR$ (Memory Read) and $MEMW$ (Memory Write). Address lines A_0 and A_1 of the MPU are usually connected to lines A_0 and A_1 of the 8254, and CS is tied to a decoded address.

The control word register and counters are selected according to the signals on lines A_0 and A_1 , as shown below:

A_1	A_0	Selection
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2
1	1	Control Register

Control Word Register and Format

The data in the register **controls the operation mode** and the selection of either binary or BCD counting format.



SC—Select Counter:

SC1 SC0

0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command (only in 8254, can latch more (See Read Operations) counters at the same time)

RW—Read/Write:

RW1 RW0

0	0	Counter Latch Command (see Read Operations)
0	1	Read/Write least significant byte only.
1	0	Read/Write most significant byte only.
1	1	Read/Write least significant byte first, then most significant byte.

M—MODE:

M2 M1 M0

0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

BCD:

0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

D3	D2	D1	
M2	M1	M0	
0	0	0	mode 0: interrupt on terminal count
0	0	1	mode 1: programmable one-shot
x	1	0	mode 2: rate generator
x	1	1	mode 3: square wave generator
1	0	0	mode 4: software triggered strobe
1	0	1	mode 5: hardware triggered strobe

Programming the 8254

WRITE OPERATIONS

To initialize a counter, the following steps are necessary.

1. Write a control word into the control register.
2. Load the low-order byte of a count in the counter register.
3. Load the high-order byte of a count in the counter register.

With a clock and an appropriate gate signal to one of the counters, the above steps should start the counter and provide appropriate output according to the control word.

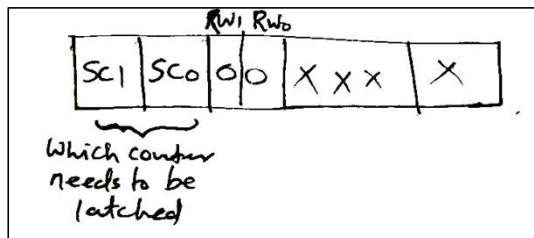
Programming the 8254

READ OPERATIONS

In some applications, especially in event counters, it is necessary to read the value of the count in progress. This can be done by either of two methods. One method involves reading a count after inhibiting (stopping) the counter to be read. The second method involves reading a count while the count is in progress (known as reading on the fly).

In the first method, counting is stopped (or inhibited) by controlling the gate input or the clock input of the selected counter, and two I/O read operations are performed by the MPU. The first I/O operation reads the low-order byte, and the second I/O operation reads the high-order byte.

In the second method, an appropriate control word is written into the control register to latch a count in the output latch, and two I/O Read operations are performed by the MPU.



Counter latch command

**When the counter latch command is sent,
the latched count is held until it is read.
After reading, it follows the current counting**

Programming the 8254

READ-BACK COMMAND

The Read-Back Command in the 8254 allows the user to read the count and the status of the counter; this command is not available in the 8253. The format of the command is shown in Figure

(a) Read-Back Command Format

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	1	COUNT	STATUS	CNT 2	CNT 1	CNT 0	0

D₅: 0 = Latch Count of Selected Counter(s)

D₄: 0 = Latch Status of Selected Counter(s)

D₃: 1 = Select Counter 2

D₂: 1 = Select Counter 1

D₁: 1 = Select Counter 0

D₀: Reserved for Future Expansion; Must Be 0

A₀, A₁ = 11

\overline{CS} = 0

\overline{RD} = 1

\overline{WR} = 0

The command is written in the control register, and the count of the specified counter(s) can be latched if COUNT (bit D₅) is 0. A counter or a combination of counters is specified by keeping the respective CNT bits (D₁, D₂, and D₃) high. For example, the control word 1 1 0 1 0 1 1 0 (D6H) written in the control register will latch the counts of Counter 0 and Counter 1, and these counts can be obtained by reading respective counter address.

(b) Status Byte

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
OUTPUT	NULL COUNT	RW1	RW0	M2	M1	M0	BCD

D₇: 1 = Out Pin is 1

: 0 = Out Pin is 0

D₆: 1 = Null Count

: 0 = Count Available
for Reading

D₅-D₀: Counter Programmed Mode

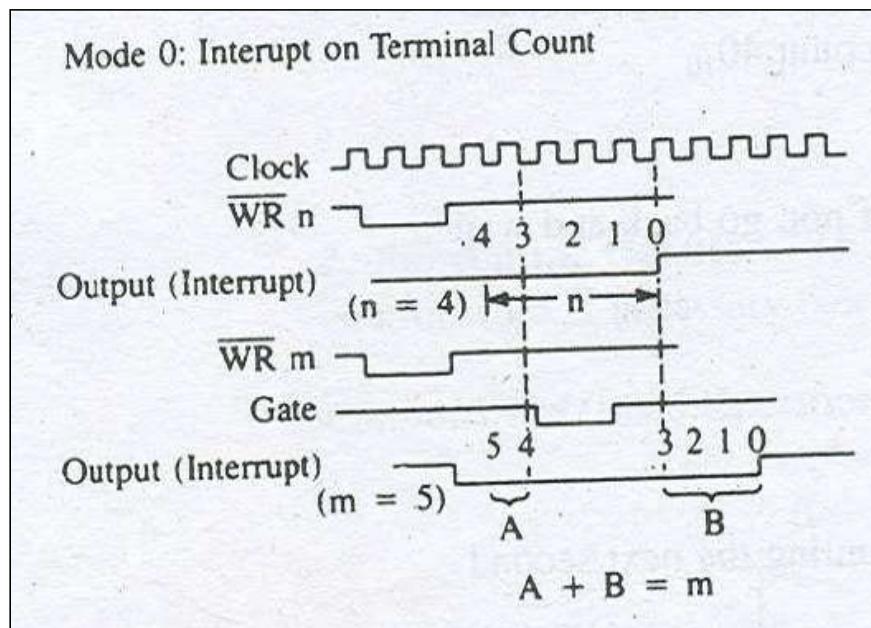
The status of the counter(s) can be read if STATUS bit (D₄) of the Read-Back Command is low. Figure (b) shows the format of the status byte.

After writing the read-back command with D4=0, read the corresponding counter for getting its status.

Modes of 8253 – Mode 0 - Interrupt on Terminal Count

MODE 0: INTERRUPT ON TERMINAL COUNT

In this mode, initially the OUT is low. Once a count is loaded in the register, the counter is decremented every cycle, and when the count reaches zero, the OUT goes high. This can be used as an interrupt. The OUT remains high until a new count or a command word is loaded. Figure also shows that the counting ($m = 5$) is temporarily stopped when the Gate is disabled ($G = 0$), and continued again when the Gate is at logic 1.

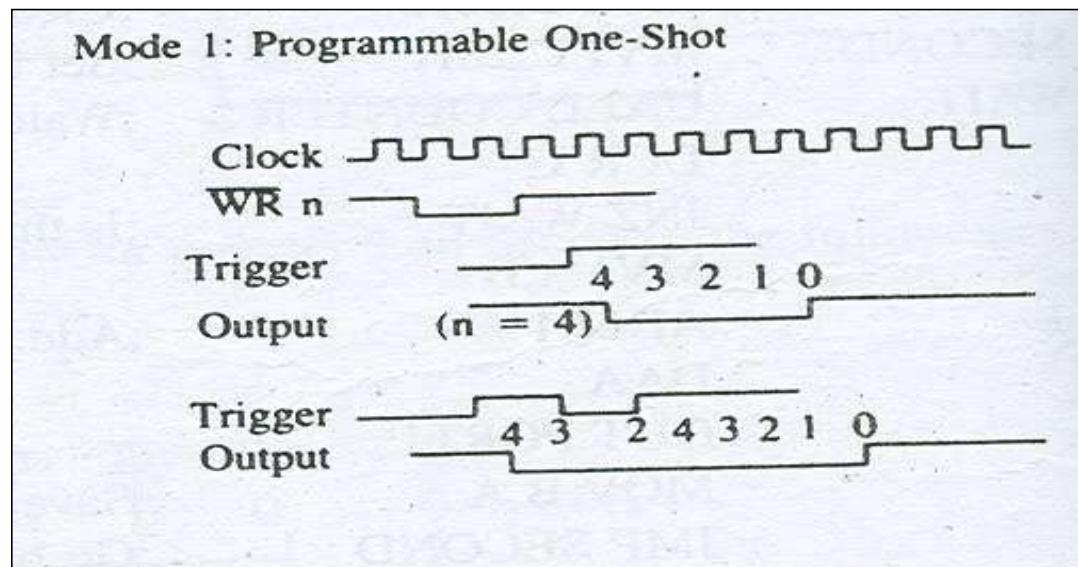


Modes of 8253 – Mode 1 - Programmable One-Shot / Hardware re-triggerable one shot

MODE 1: HARDWARE-RETRIGGERABLE ONE-SHOT

In this mode, the OUT is initially high. When the Gate is triggered, the OUT goes low, and at the end of the count, the OUT goes high again, thus generating a one-shot pulse

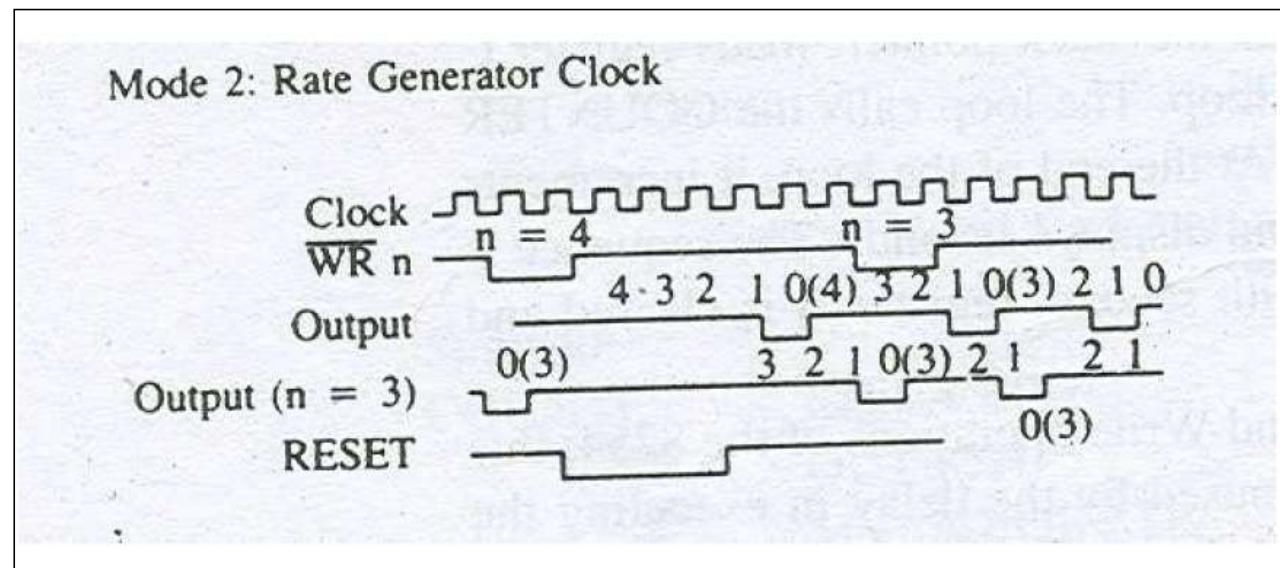
If the trigger occurs during the pulse output, the 8254 will be **retriggered** again.



Modes of 8253 – Mode 2 - Rate Generator Clock

MODE 2: RATE GENERATOR

This mode is used to generate a pulse equal to the clock period at a given interval. When a count is loaded, the OUT stays high until the count reaches 1, and then the OUT goes low for one clock period. The count is reloaded automatically, and the pulse is generated continuously. The count = 1 is illegal in this mode.

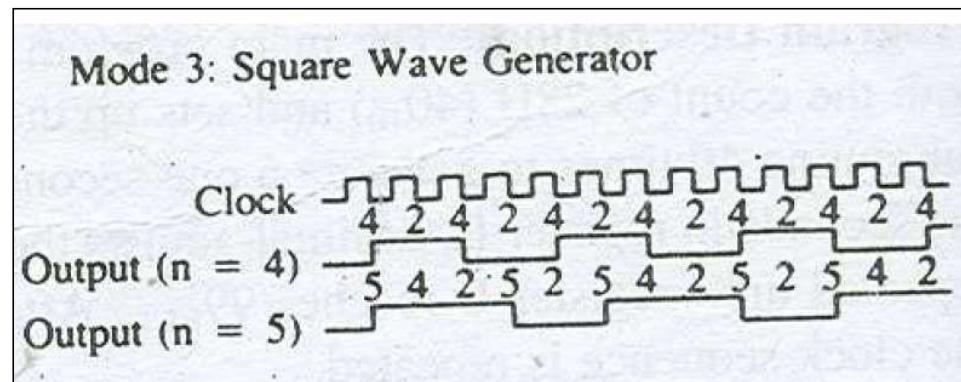


Modes of 8253 – Mode 3 - Square Wave Generator

MODE 3: SQUARE-WAVE GENERATOR

In this mode, when a count is loaded, the OUT is high. The count is decremented by two at every clock cycle, and when it reaches zero, the OUT goes low, and the count is reloaded again. This is repeated continuously; thus, a continuous square wave with period equal to the period of the count is generated. In other words, the frequency of the square wave is equal to the frequency of the clock divided by the count. If the count (N) is odd, the pulse stays high for $(N + 1)/2$ clock cycles and stays low for $(N - 1)/2$ clock cycles.

If N is even, output stays high for $N/2$ clock cycles and stays low for $N/2$ clock cycles.



Time period of output = clk period x count
Frequency of output = $1 / (\text{clk period} \times \text{count})$
= clk frequency / count
Count = clk frequency of 8253 / frequency of output

Modes of 8253 – Mode 3 - Square Wave Generator

Write instructions to generate 1KHz square wave from counter 1 of 8254. Assume the input clock frequency of 8254 as 2MHz.

$$\begin{aligned} \text{Count} &= \frac{\text{clock frequency of 8254}}{\text{frequency of output wave}} \\ &= \frac{2 \text{ MHz}}{1 \text{ KHz}} \\ &= 2000 = 07D0H \end{aligned}$$

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
SC1	SC0	RW1	RW0	M2	M1	M0	BCD
SC—Select Counter:							
SC1 SC0							
0	0	Select Counter 0					
0	1	Select Counter 1					
1	0	Select Counter 2					
1	1	Read-Back Command (See Read Operations)					
RW—Read/Write:							
RW1 RW0							
0	0	Counter Latch Command (see Read Operations)					
0	1	Read/Write least significant byte only.					
1	0	Read/Write most significant byte only.					
1	1	Read/Write least significant byte first, then most significant byte.					
M—MODE:							
M2 M1 M0							
0	0	0	Mode 0				
0	0	1	Mode 1				
X	1	0	Mode 2				
X	1	1	Mode 3				
1	0	0	Mode 4				
1	0	1	Mode 5				
BCD:							
0 Binary Counter 16-bits							
1 Binary Coded Decimal (BCD) Counter (4 Decades)							

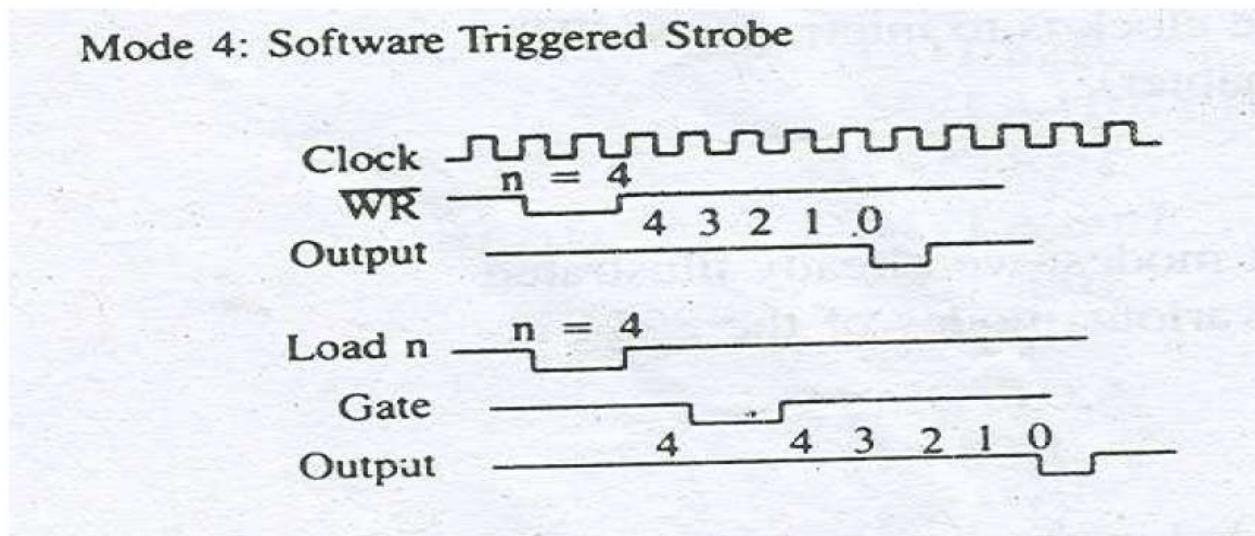
Control word = 01 11 011 0 =76H

MOV [8003], 76
MOV [8001], D0H
MOV [8001], 07H
HLT

Modes of 8253 – Mode 4 - Software Triggered Strobe

MODE 4: SOFTWARE-TRIGGERED STROBE

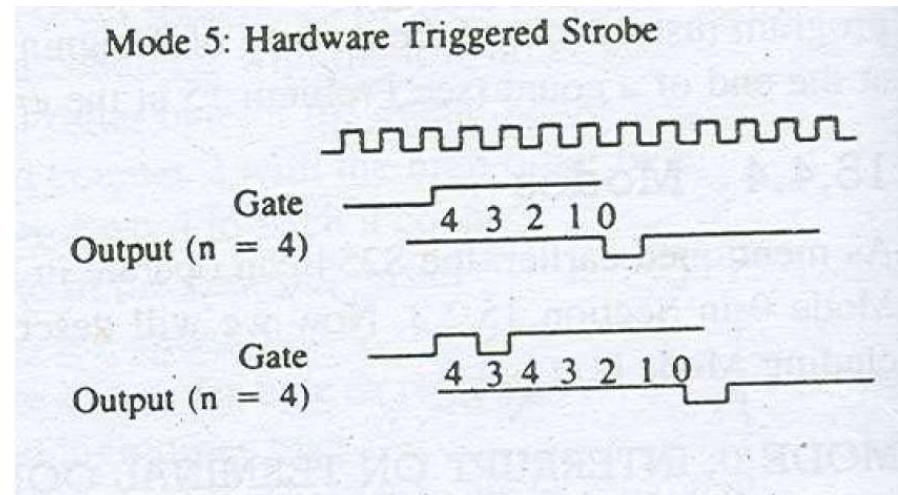
In this mode, the OUT is initially high; it goes low for one clock period at the end of the count. The count must be reloaded for subsequent outputs.



Modes of 8253 – Mode 5 - Hardware Triggered Strobe

MODE 5: HARDWARE-TRIGGERED STROBE

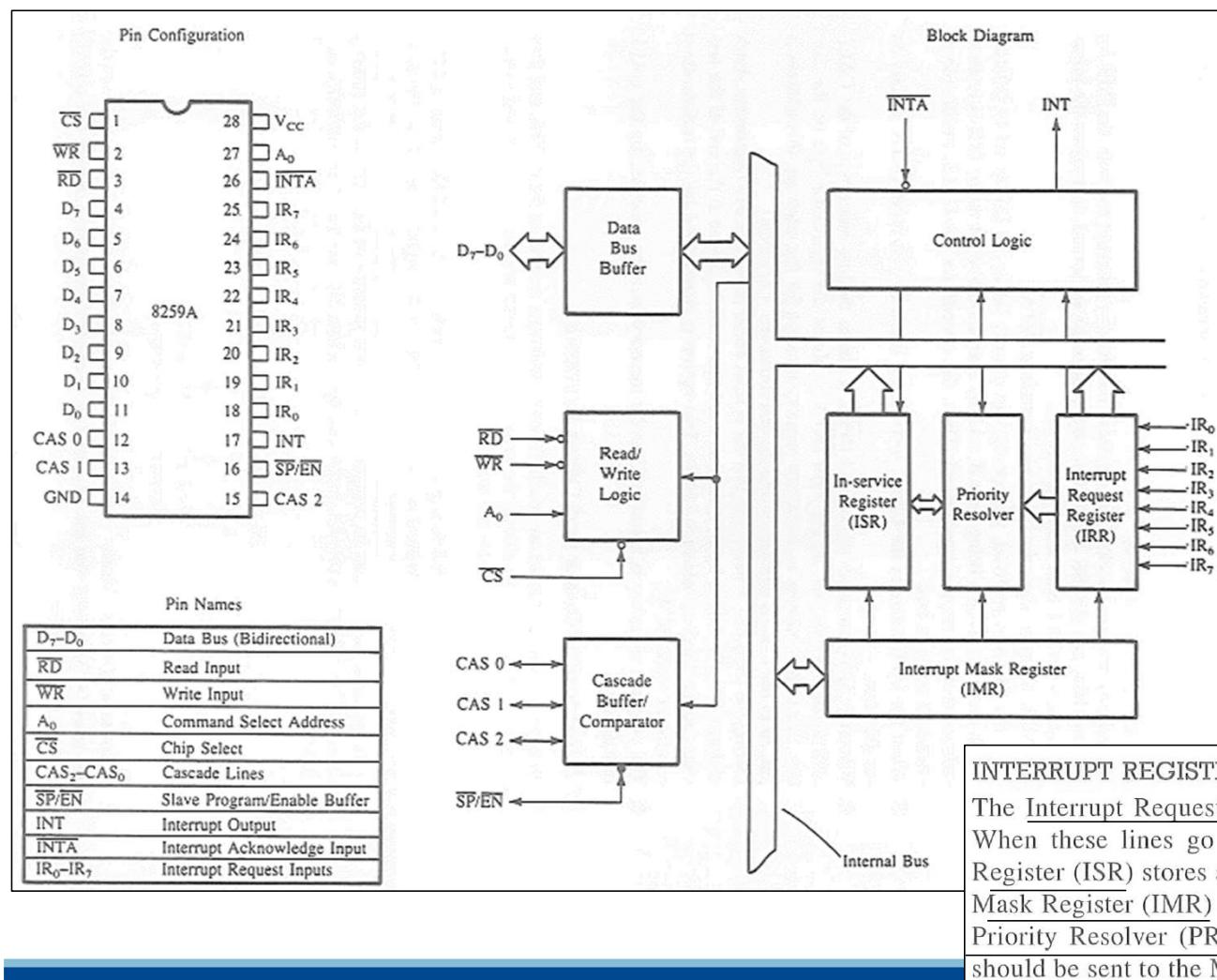
This mode is similar to Mode 4, except that it is triggered by the rising pulse at the gate.
when the Gate pulse is triggered from low to high, the count begins. At the end of the count, the OUT goes low for one clock period.



8259: Interrupt Controller

- The 8259 is known as the Programmable Interrupt Controller (PIC) microprocessor.
- In 8085 and 8086 there are five hardware interrupts and two hardware interrupts respectively.
- By adding 8259, we can increase the interrupt handling capability.
- This chip combines the multi-interrupt input source to single interrupt output.
- This provides 8-interrupts inputs (IR0 to IR7).
- This chip is designed for 8085 and 8086.
- It can be programmed either in edge triggered, or in level triggered mode
- We can mask individual interrupts.
- By cascading 8259 chips, we can increase interrupts up to 64 interrupt lines

8259 Block diagram

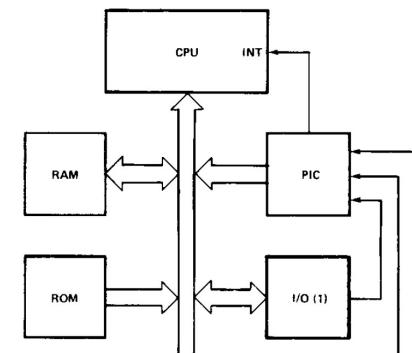


For master – CAS line act as o/p line
For slave – CAS line act as i/p line

In non-buffered mode,
It act as input pin

$\overline{SP}/\overline{EN}$ = 1 for master and 0 for slaves

In buffered mode,
It act as output line,
used to enable the buffers.



INTERRUPT REGISTERS 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 the **Interrupt Mask Register (IMR)** stores the masking bits of the interrupt lines to be masked. The **Priority Resolver (PR)** examines these three registers and determines whether INT should be sent to the MPU.

8259 Pin Details

Symbol	Pin No.	Type	Name and Function
V _{CC}	28	I	SUPPLY: +5V Supply.
GND	14	I	GROUND
CS	1	I	CHIP SELECT: A low on this pin enables RD and WR communication between the CPU and the 8259A. INTA functions are independent of CS.
WR	2	I	WRITE: A low on this pin when CS is low enables the 8259A to accept command words from the CPU.
RD	3	I	READ: A low on this pin when CS is low enables the 8259A to release status onto the data bus for the CPU.
D ₇ -D ₀	4-11	I/O	BIDIRECTIONAL DATA BUS: Control, status and interrupt-vector information is transferred via this bus.
CAS ₀ -CAS ₂	12, 13, 15	I/O	CASCADE LINES: The CAS lines form a private 8259A bus to control a multiple 8259A structure. These pins are outputs for a master 8259A and inputs for a slave 8259A.

Pin Details

$\overline{SP/EN}$	16	I/O	SLAVE PROGRAM/ENABLE BUFFER: This is a dual function pin. When in the Buffered Mode it can be used as an output to control buffer transceivers (EN). When not in the buffered mode it is used as an input to designate a master ($SP = 1$) or slave ($SP = 0$).
INT	17	O	INTERRUPT: This pin goes high whenever a valid interrupt request is asserted. It is used to interrupt the CPU, thus it is connected to the CPU's interrupt pin.
IR_0-IR_7	18–25	I	INTERRUPT REQUESTS: Asynchronous inputs. An interrupt request is executed by raising an IR input (low to high), and holding it high until it is acknowledged (Edge Triggered Mode), or just by a high level on an IR input (Level Triggered Mode).
INTA	26	I	INTERRUPT ACKNOWLEDGE: This pin is used to enable 8259A interrupt-vector data onto the data bus by a sequence of interrupt acknowledge pulses issued by the CPU.
A_0	27	I	AO ADDRESS LINE: This pin acts in conjunction with the \overline{CS} , \overline{WR} , and \overline{RD} pins. It is used by the 8259A to decipher various Command Words the CPU writes and status the CPU wishes to read. It is typically connected to the CPU A0 address line (A1 for 8086, 8088).

Operation of 8259

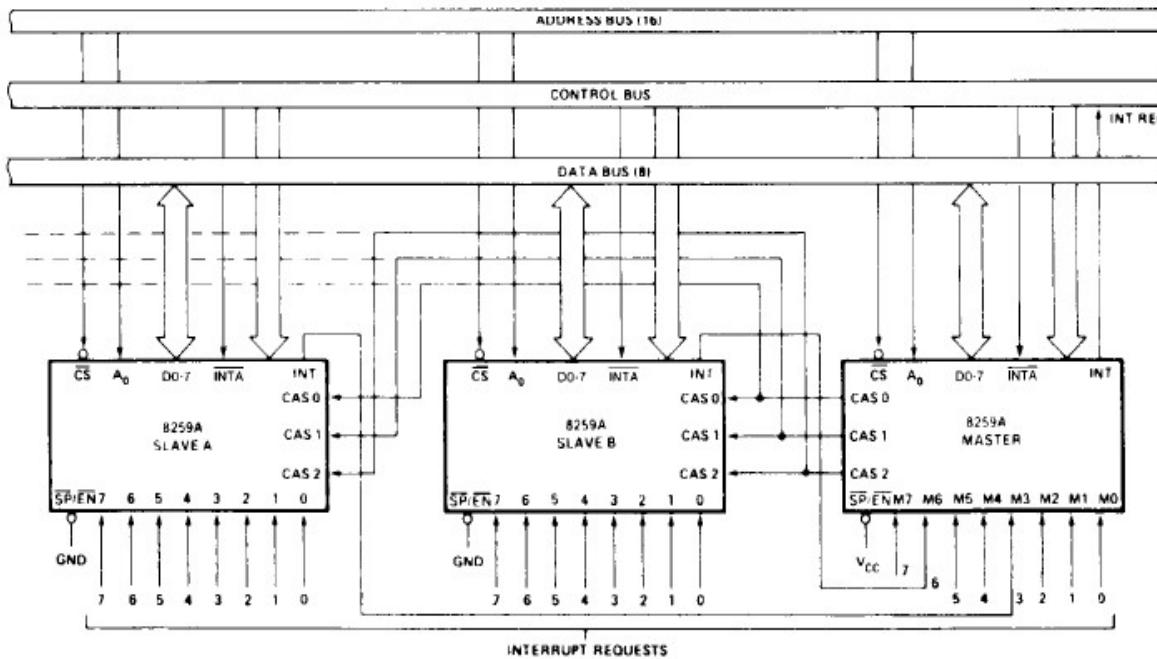
With 8085

1. One or more of the INTERRUPT REQUEST lines (IR7–0) are raised high, setting the corresponding IRR bit(s).
2. The 8259A evaluates these requests, and sends an INT to the CPU, if appropriate.
3. The CPU acknowledges the INT and responds with an INTA pulse.
4. Upon receiving an INTA from the CPU group, the highest priority ISR bit is set, and the corresponding IRR bit is reset. The 8259A will also release a CALL instruction code (11001101) onto the 8-bit Data Bus through its D7–0 pins.
5. This CALL instruction will initiate two more INTA pulses to be sent to the 8259A from the CPU group.
6. These two INTA pulses allow the 8259A to release its preprogrammed subroutine address onto the Data Bus. The lower 8-bit address is released at the first INTA pulse and the higher 8-bit address is released at the second INTA pulse.
7. This completes the 3-byte CALL instruction released by the 8259A. In the AEOI mode the ISR bit is reset at the end of the third INTA pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt sequence.

With 8086

1. One or more of the INTERRUPT REQUEST lines (IR7–0) are raised high, setting the corresponding IRR bit(s).
2. The 8259A evaluates these requests, and sends an INT to the CPU, if appropriate.
3. The CPU acknowledges the INT and responds with an INTA pulse.
4. Upon receiving an INTA from the CPU group, the highest priority ISR bit is set and the corresponding IRR bit is reset. The 8259A does not drive the Data Bus during this cycle.
5. The 8086 will initiate a second INTA pulse. During this pulse, the 8259A releases an 8-bit pointer onto the Data Bus where it is read by the CPU.
6. This completes the interrupt cycle. In the AEOI mode the ISR bit is reset at the end of the second INTA pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

Cascading operation of 8259



Interrupt to slave

If it is not masked and no high priority interrupt, INT will be sent to master

In master, If it is not masked and no high priority interrupt, INT will be sent to 8086

If Interrupt flag is enabled, 8086 will send !INTA to all 8259s

Slave ignores first !INTA

Master receives first !INTA

It outputs a 3 bit slave identification on CAS0 – CAS2 (slave id is given to each slave as part of its initialization)

This ID enables the slave

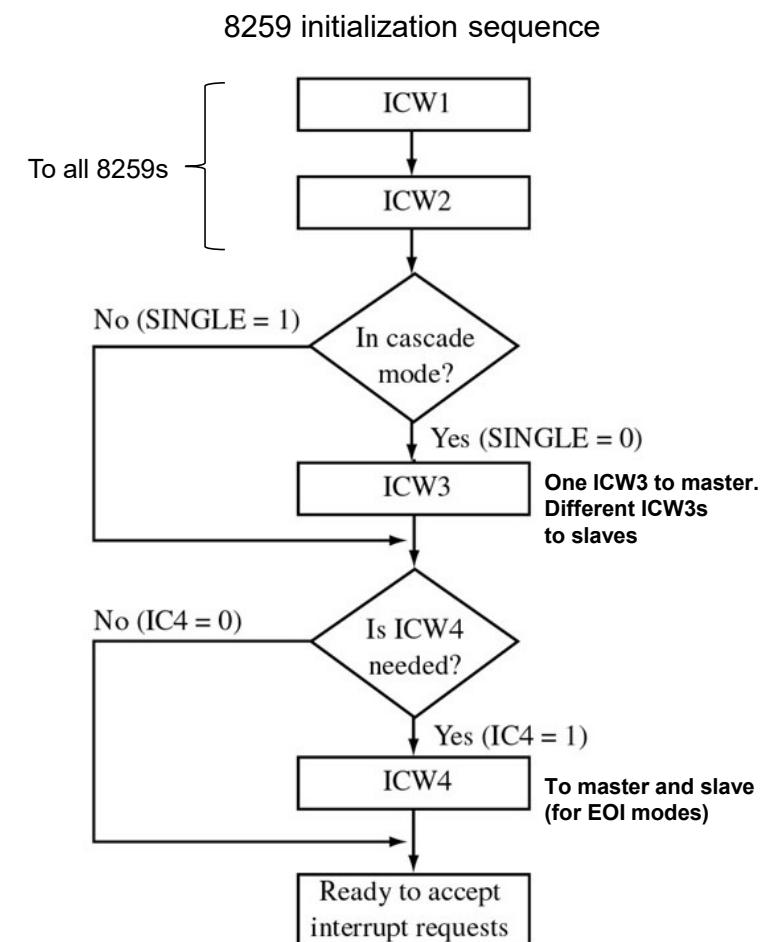
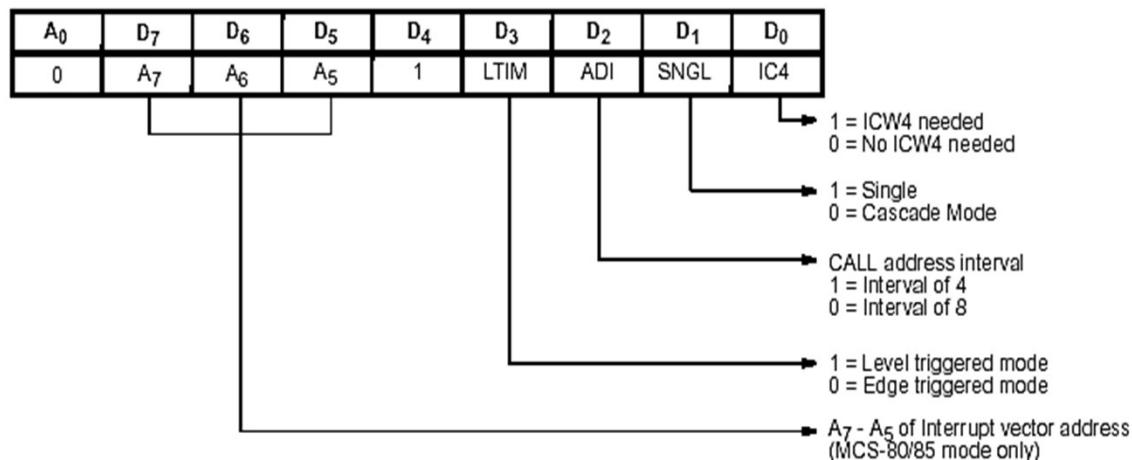
When slave receives second !INTA from 8086, it will send the interrupt type number to data bus.

Programming 8259

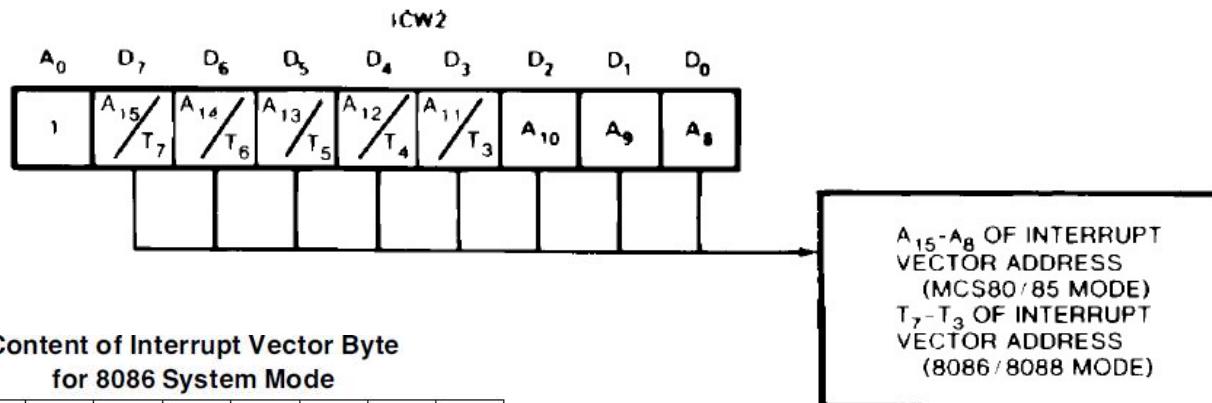
- There are 2 Command Words in 8259.
- 1. Initialization Command Words (ICWs): Before normal operation can begin, each 8259 in the system must be brought to a starting point using these command words.
 - There are 4 ICWs in 8259.
 - ICW1 – to give basic settings
 - ICW2 – to setup call vector address or interrupt number
 - ICW 3 – to give ID for slaves
 - ICW 4 – 85/86 selection, buffered mode selection, priority modes etc.
- 2. Operation Command Words (OCWs): These are the command words which command the 8259 to operate in various interrupt modes.
 - There are 3 OCWs in 8259
 - OCW1 – to setup masking
 - OCW2 – EOI (used to reset ISR bits)
 - OCW3 – To setup special mask mode , poll mode etc

Programming 8259

ICW1 format



Programming 8259



**Content of Interrupt Vector Byte
for 8086 System Mode**

	D7	D6	D5	D4	D3	D2	D1	D0
IR7	T7	T6	T5	T4	T3	1	1	1
IR6	T7	T6	T5	T4	T3	1	1	0
IR5	T7	T6	T5	T4	T3	1	0	1
IR4	T7	T6	T5	T4	T3	1	0	0
IR3	T7	T6	T5	T4	T3	0	1	1
IR2	T7	T6	T5	T4	T3	0	1	0
IR1	T7	T6	T5	T4	T3	0	0	1
IR0	T7	T6	T5	T4	T3	0	0	0

Assume 0010 0000 (32₁₀)
If IR1=1, then 0010 0001 will be given to data bus
i.e. INT 33 will be executed.

Content of Second Interrupt Vector Byte

IR	Interval = 4							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	A5	1	1	1	0	0
6	A7	A6	A5	1	1	0	0	0
5	A7	A6	A5	1	0	1	0	0
4	A7	A6	A5	1	0	0	0	0
3	A7	A6	A5	0	1	1	0	0
2	A7	A6	A5	0	1	0	0	0
1	A7	A6	A5	0	0	1	0	0
0	A7	A6	A5	0	0	0	0	0

IR	Interval = 8							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	1	1	1	0	0	0
6	A7	A6	1	1	0	0	0	0
5	A7	A6	1	0	1	0	0	0
4	A7	A6	1	0	0	0	0	0
3	A7	A6	0	1	1	0	0	0
2	A7	A6	0	1	0	0	0	0
1	A7	A6	0	0	1	0	0	0
0	A7	A6	0	0	0	0	0	0

Programming 8259

ICW3 (Master device)

A ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	S ₇	S ₆	S ₅	S ₄	S ₃	S ₂	S ₁	S ₀

0 = IR Input has a slave
1 = IR Input does not have a slave

ICW3 (SLAVE DEVICE)

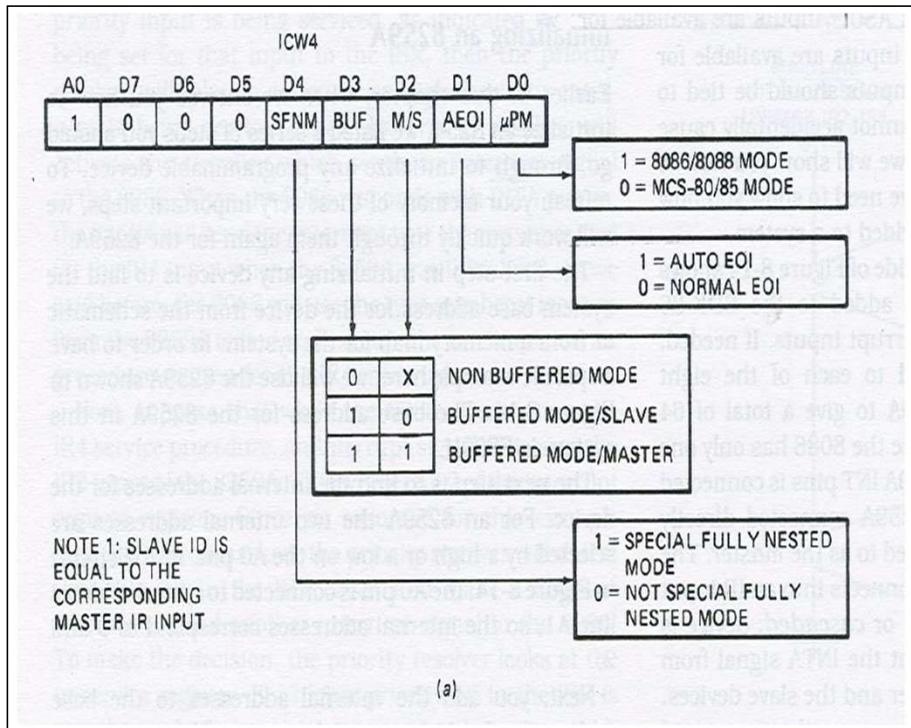
A ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	0	0	ID ₂	ID ₁	ID ₀

SLAVE ID

0	1	2	3	4	5	6	7
0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	0	0	0	1	1	1	1

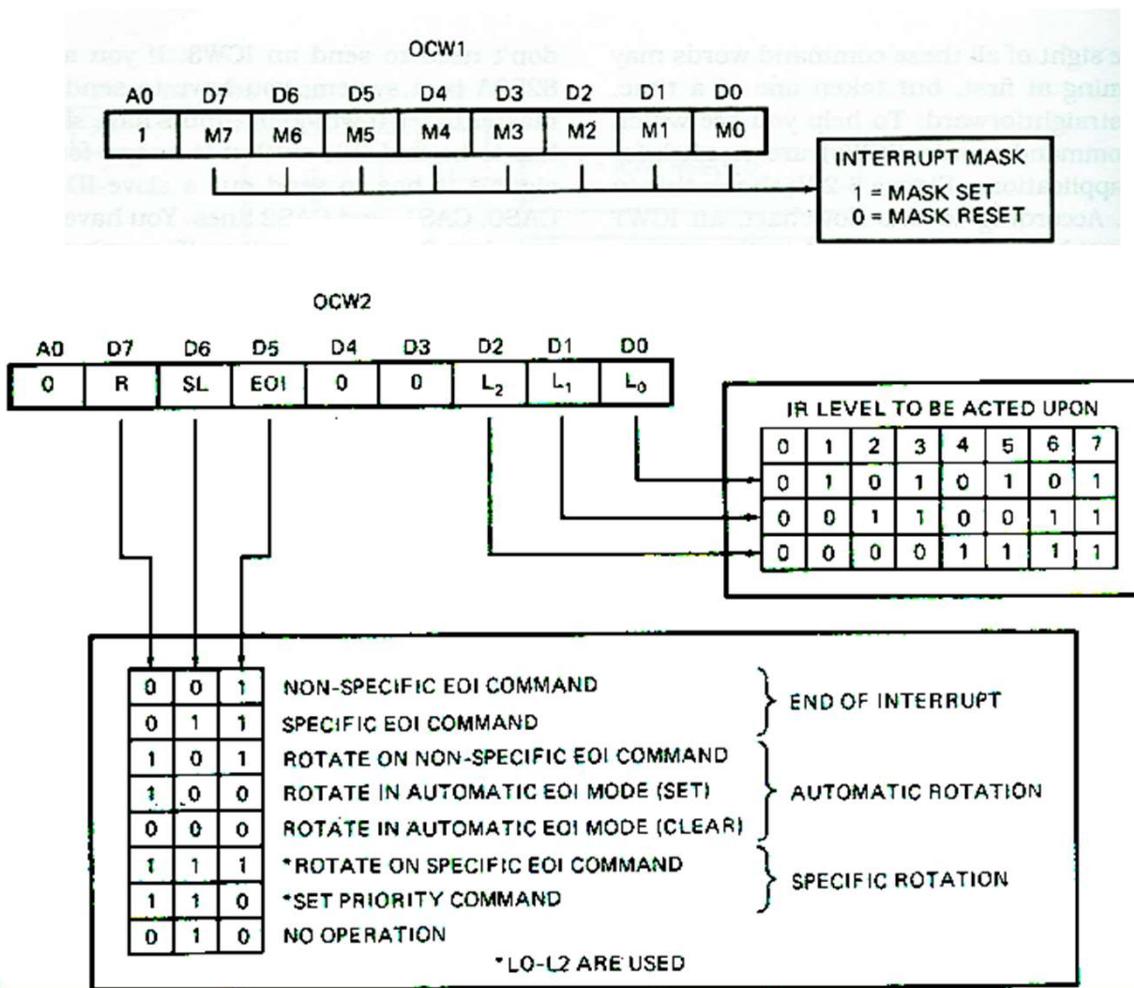
Programming 8259

ICW4



- SFNM: If SFNM = 1, the special fully nested mode is programmed.
- BUF: If BUF = 1, the buffered mode is programmed. In buffered mode, $\overline{SP}/\overline{EN}$ becomes an enable output and the master/slave determination is by M/S.
- M/S: If buffered mode is selected: M/S = 1 means the 82C59A is programmed to be a master, M/S = 0 means the 82C59A is programmed to be a slave. If BUF = 0, M/S has no function.
- AEOI: If AEOI = 1, the automatic end of interrupt mode is programmed.
- μ PM: Microprocessor mode: μ PM = 0 sets the 82C59A for 8080/85 system operation, μ PM = 1 sets the 82C59A for 80C86/88/286 system operation.

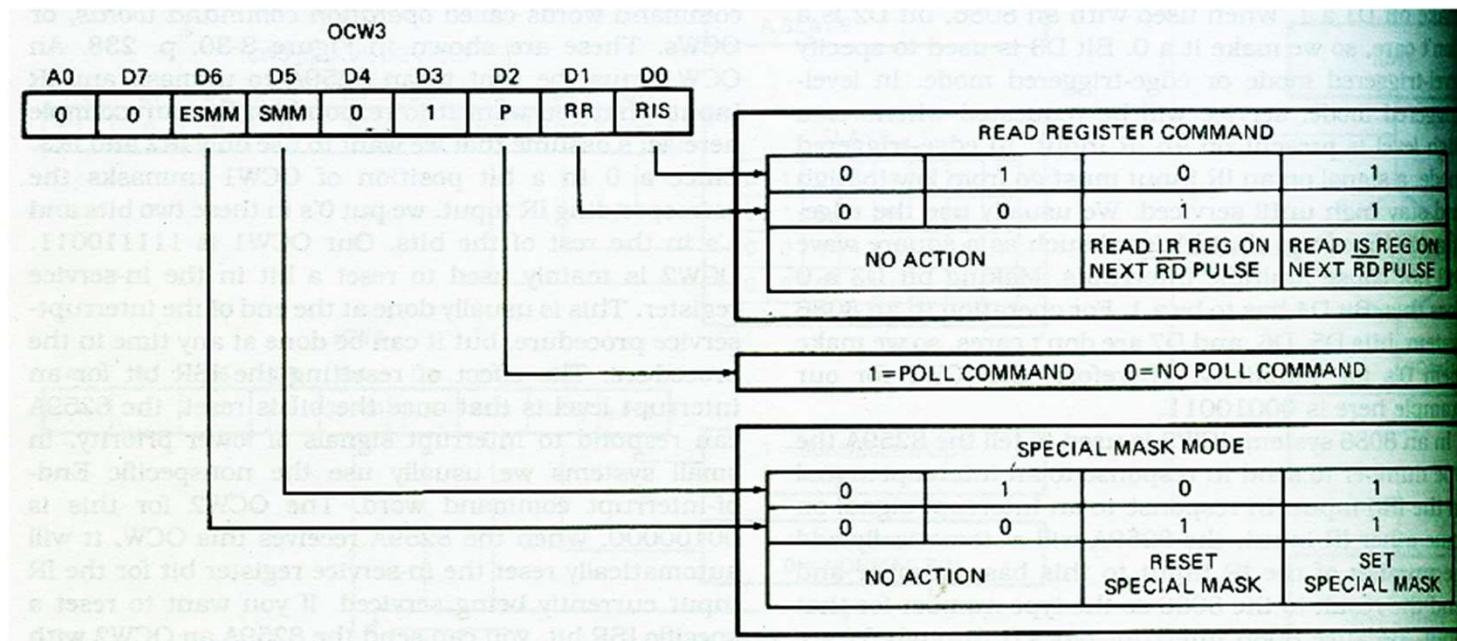
Programming 8259



R, SL, EOI: These three bits control the Rotate and End of Interrupt modes and combinations of the two.

L₂, L₁, L₀: These bits determine the interrupt level acted upon when the SL bit is active.

Programming 8259



ESMM - Enable Special Mask Mode. When this bit is set to 1 it enables the SMM bit to set or reset the Special Mask Mode. When ESMM = 0, the SMM bit becomes a “don’t care”.

SMM - Special Mask Mode. If ESMM = 1 and SMM = 1, the 82C59A will enter Special Mask Mode. If ESMM = 1 and SMM = 0, the 82C59A will revert to normal mask mode. When ESMM = 0, SMM has no effect.

8259 Working Modes

- There are 4 different modes for 8259.
 1. Fully nested mode.
 2. Rotating priority mode.
 3. Special mask mode.
 4. Polled mode.

Fully nested mode

- This mode is entered after initialization unless another mode is programmed.
- The interrupt requests are ordered in priority from 0 through 7 (IR0 – highest, IR7 - lowest).
- When an interrupt is acknowledged the highest priority request is determined and its vector placed on the bus.
- Additionally, a bit of the Interrupt Service register (IS0-7) is set.
- If AEOI (Automatic End of Interrupt) bit is set, this bit remains set until the trailing edge of the last INTA.
- This bit remains set until the microprocessor issues an End of Interrupt (EOI) command immediately before returning from the service routine
- 2 types of EOI : specific EOI -> can specify which ISR bit to reset
 - non-specific -> highest priority bit in ISR will be reset

Rotating priority mode

- In some applications there are a number of interrupting devices of equal priority.
- In this mode, a device after being serviced, receives the lowest priority.
- So a device requesting an interrupt will have to wait, in the worst case until each of 7 other devices are serviced at most once .

Specific rotation:
Bottom priority can be
Given to a specific line

Before Rotate (IR4 the highest priority requiring service)

IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
0	1	0	1	0	0	0	0

"IS" Status

Priority Status							
Lowest Priority				Highest Priority			
7	6	5	4	3	2	1	0

After Rotate (IR4 was serviced, all other priorities rotated correspondingly)

IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
0	1	0	0	0	0	0	0

"IS" Status

Priority Status							
Highest Priority				Lowest Priority			
2	1	0	7	6	5	4	3

Special mask mode

- Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control.
- For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion.
- That is where the Special Mask Mode comes in.
- In the special Mask Mode, when a mask bit is set in OCW1, it inhibits further interrupts at that level and enables interrupts from all other levels (lower as well as higher) that are not masked.
- Thus, any interrupts may be selectively enabled by loading the mask register.

Polled mode

- In Polled mode the INT output functions as it normally does.
 - The microprocessor should ignore this output.
 - This can be accomplished either by not connecting the INT output or by masking interrupts within the microprocessor, thereby disabling its interrupt input.
 - Service to devices is achieved by software using a Poll command.
 - The Poll command is issued by setting P = 1 in OCW3.

The Poll command is issued by setting P = '1' in OCW3. The 8259A treats the next RD pulse to the 8259A (i.e., RD = 0, CS = 0) as an interrupt acknowledgement, sets the appropriate IS bit if there is a request, and reads the | el. Interrupt is fro-

The word enabled onto the data bus during RD is:

D7	D6	D5	D4	D3	D2	D1	D0
I	—	—	—	—	W2	W1	W0

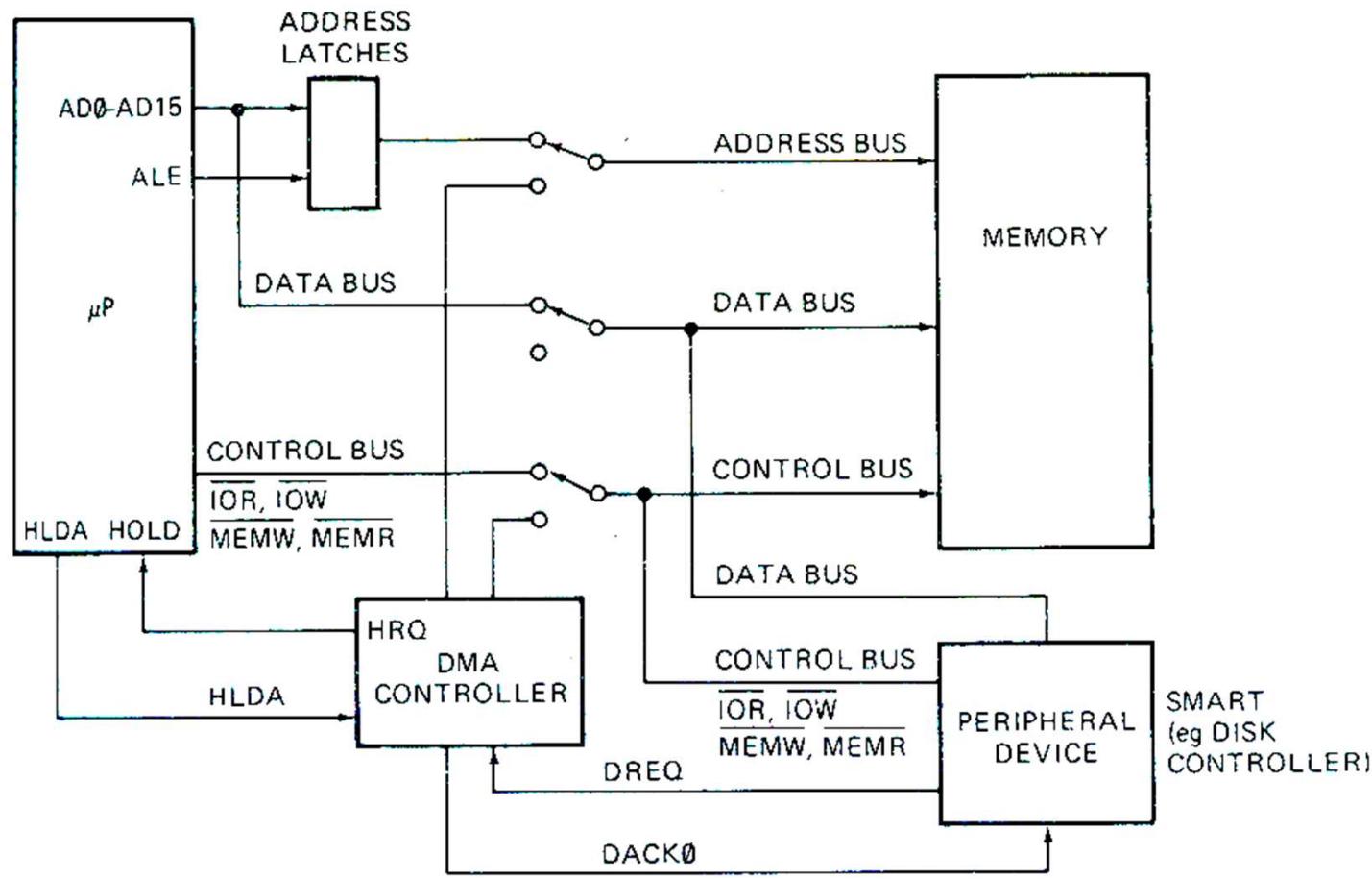
W0–W2: Binary code of the highest priority level requesting service.

I: Equal to “1” if there is an interrupt.

8237: Direct memory access

- DMA is for high-speed data transfer from/to mass storage peripherals, e.g. hard disk drive, magnetic tape, CD-ROM, and sometimes video controllers.
- Direct memory access (DMA) is a process in which an external device takes over the control of system bus from the CPU.
- The basic idea of DMA is to transfer blocks of data directly between memory and peripherals.
- The data don't go through the microprocessor but the data bus is occupied.
- The HOLD and HLDA pins are used to receive and acknowledge the hold request respectively (In minimum mode).
- Normally the CPU has full control of the system bus.
- In a DMA operation, the peripheral takes over bus control temporarily.

Basic process of DMA



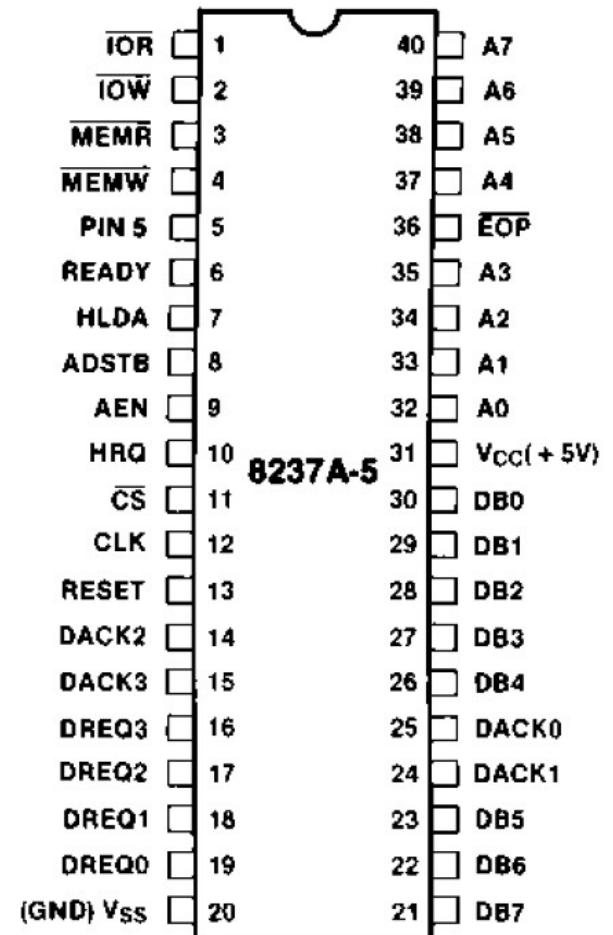
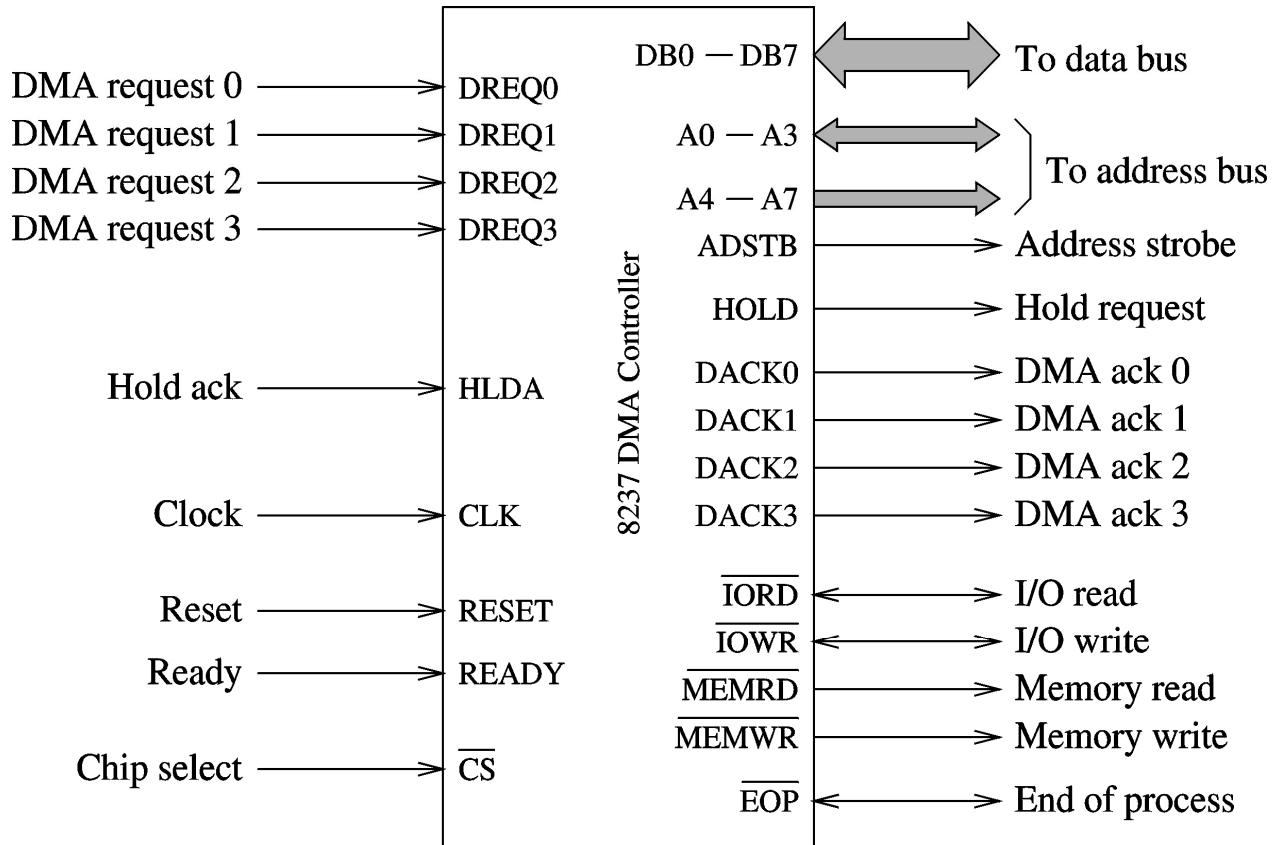
Basic process of DMA

- The RQ/GT1 and RQ/GT0 pins are used to issue DMA request and receive acknowledge signals (In maximum mode).
- Sequence of events of a typical DMA process:
 1. Peripheral asserts one of the request pins, e.g. RQ/GT1 or RQ/GT0 (RQ/GT0 has higher priority)
 2. 8086 completes its current bus cycle and enters into a HOLD state.
 3. 8086 grants the right of bus control by asserting a grant signal via the same pin as the request signal.
 4. DMA operation starts.
 5. Upon completion of the DMA operation, the peripheral asserts the request/grant pin again to relinquish bus control.

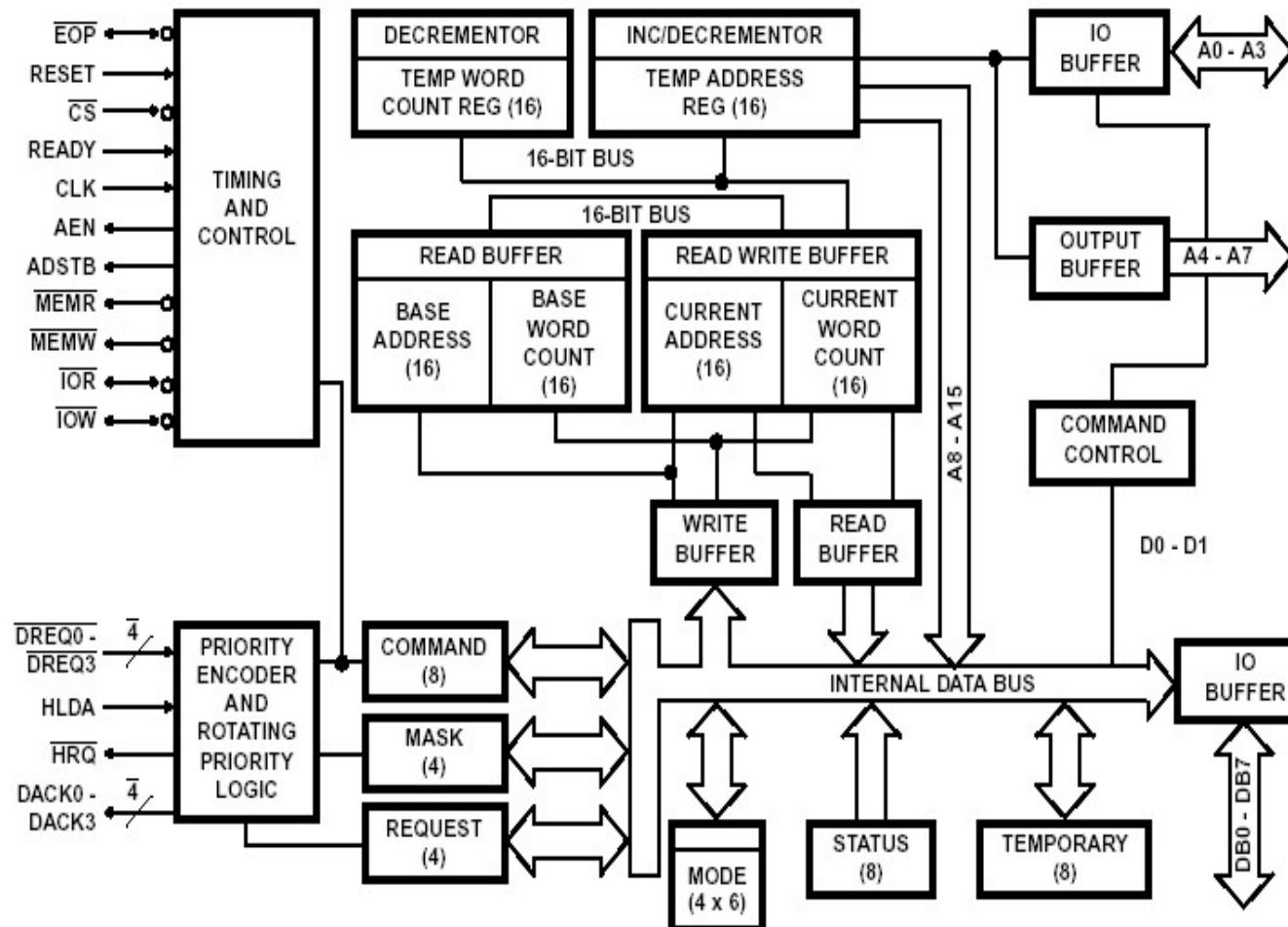
DMA controller

- A DMA controller interfaces with several peripherals that may request DMA.
- The controller decides the priority of simultaneous DMA requests communicates with the peripheral and the CPU, and provides memory addresses for data transfer.
- DMA controller commonly used with 8086 is the 8257/8237 programmable device.
- The 8257/8237 is a 4-channel device.
- 8237 is the advanced version of 8257
- Each channel is dedicated to a specific peripheral device and capable of addressing 64 K bytes section of memory.

8237



Block Diagram

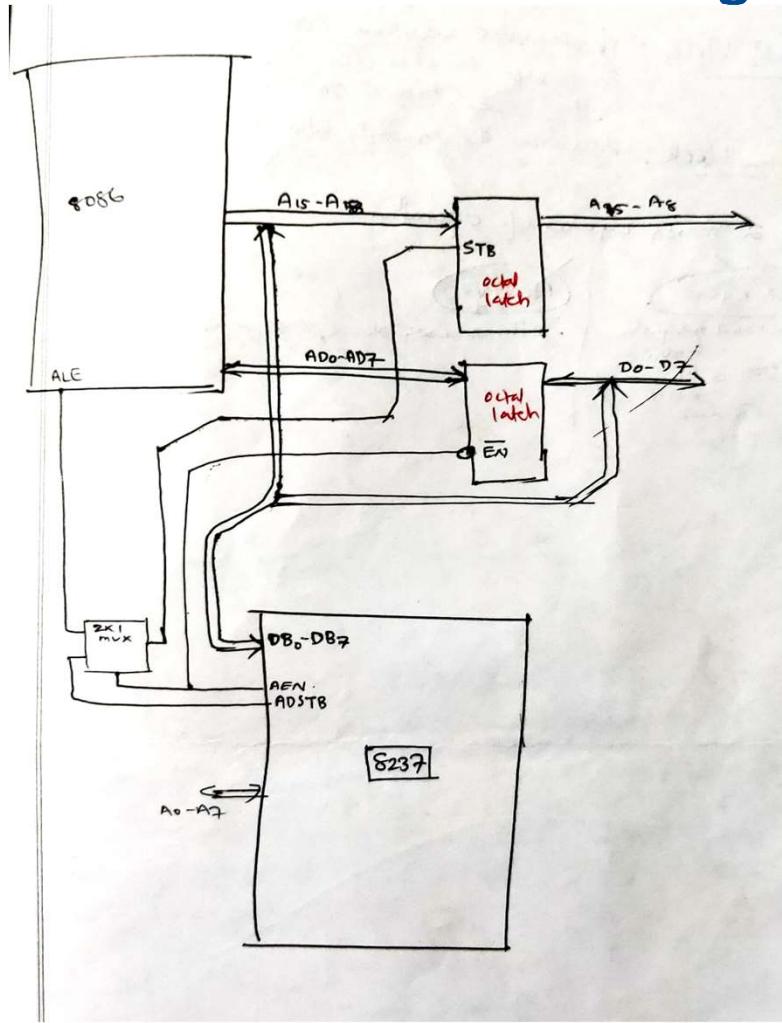


- 4 channels
- Two 16 bit registers associated with each channel (for storing base address and count)
- 8 more registers (to write commands or read status)
- Totally 16 registers
- Registers can be selected using A3-A0 and chip select line

Memory to memory transfer

- Reads the source using channel 0
- And stores in internal buffer
- Then from buffer to destination, by using channel 1,
- Channel 2 count register is used as count register

16 bit address generation



AEN = Address enable.
AEN = 1 when 8237 wants to access the bus
ADSTB = Address strobe.
High if DB₇-DB₀ contains the higher 8 bit address bus.

8237 Registers

- 1. Current word register**
- 2. Command register**
- 3. Mode register**
- 4. Request register**
- 5. Mask register**
- 6. Status register**
- 7. Temporary register**
- 8. Current address register**

Reg. Addr	Internal Registers
00	CH0 Memory Address Reg.
01	CH0 Count Reg.
02	CH1 Memory Address Reg.
03	CH1 Count Reg.
04	CH2 Memory Address Reg.
05	CH2 Count Reg.
06	CH3 Memory Address Reg.
07	CH3 Count Reg.
08	R/W Status/Command Reg.
09	WR Request Reg.
0A	WR Single Mask Reg.
0B	WR Mode Reg.
0C	WR Clear Byte Pointer F/F
0D	R/W Master Clear/Temp. Reg.
0E	WR Clear Mask Reg.
0F	WR All Mask Reg. Bits

8237 Registers

1. Current address register

- One 16-bit register for each channel
- Holds address for the current DMA transfer

2. Current word register

- Keeps the byte count
- Generates terminal count (TC) signal when the count goes from zero to FFFFH

3. Command register

- Used to program 8257

4. Mode register

- Each channel can be programmed to
 - Read or write
 - Autoincrement or autodecrement the address
 - Auto initialize the channel

5. Request register

- For software-initiated DMA

6. Mask register

- Used to disable a specific channel

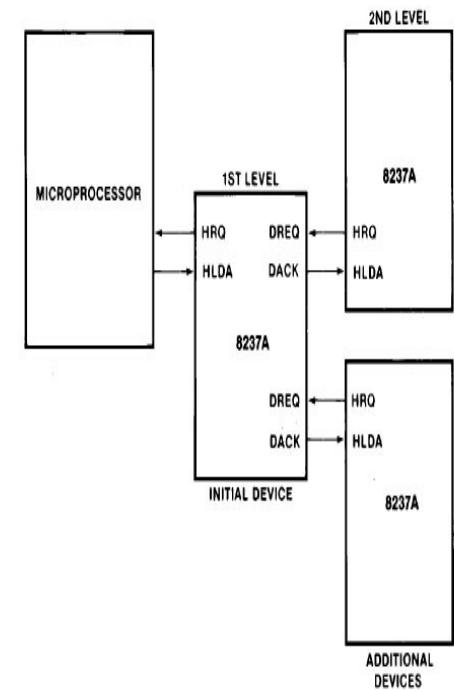
7. Status register

8. Temporary register

- Used for memory-to-memory transfers

Types of data transfer

- 8237 supports **four** types of data transfer
 1. **Single cycle transfer (Cycle stealing mode)**
 - Only single transfer takes place
 - Useful for slow devices
 2. **Block transfer mode**
 - Transfers data until TC is generated or external EOP signal is received
 3. **Demand transfer mode**
 - Similar to the block transfer mode
 - In addition to TC and EOP, transfer can be terminated by deactivating DREQ signal
 - I/O device can re-initiate the transfer by giving DREQ, so the address and count of last transfer are kept in current address and current word count reg
 4. **Cascade mode**
 - Useful to expand the number channels beyond four



.Cascaded 8237As

231466-3

Command Register

Describes the complete operation of 8237

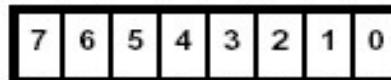
Extended write:

Time period of mem/IO write signal is extended

Late write: to sync with slow speed devices

Address hold : set if same word need To be written into block of locations

Command Register



BIT NUMBER

- { 0 Memory-to-memory disable
1 Memory-to-memory enable
- { 0 Channel 0 address hold disable
1 Channel 0 address hold enable
X If bit 0 = 0
- { 0 Controller enable
1 Controller disable
- { 0 Normal timing
1 Compressed timing
X If bit 0 = 1
- { 0 Fixed priority
1 Rotating priority
- { 0 Late write selection
1 Extended write selection
X If bit 3 = 1
- { 0 DREQ sense active high
1 DREQ sense active low
- { 0 DACK sense active low
1 DACK sense active high

Mode Register

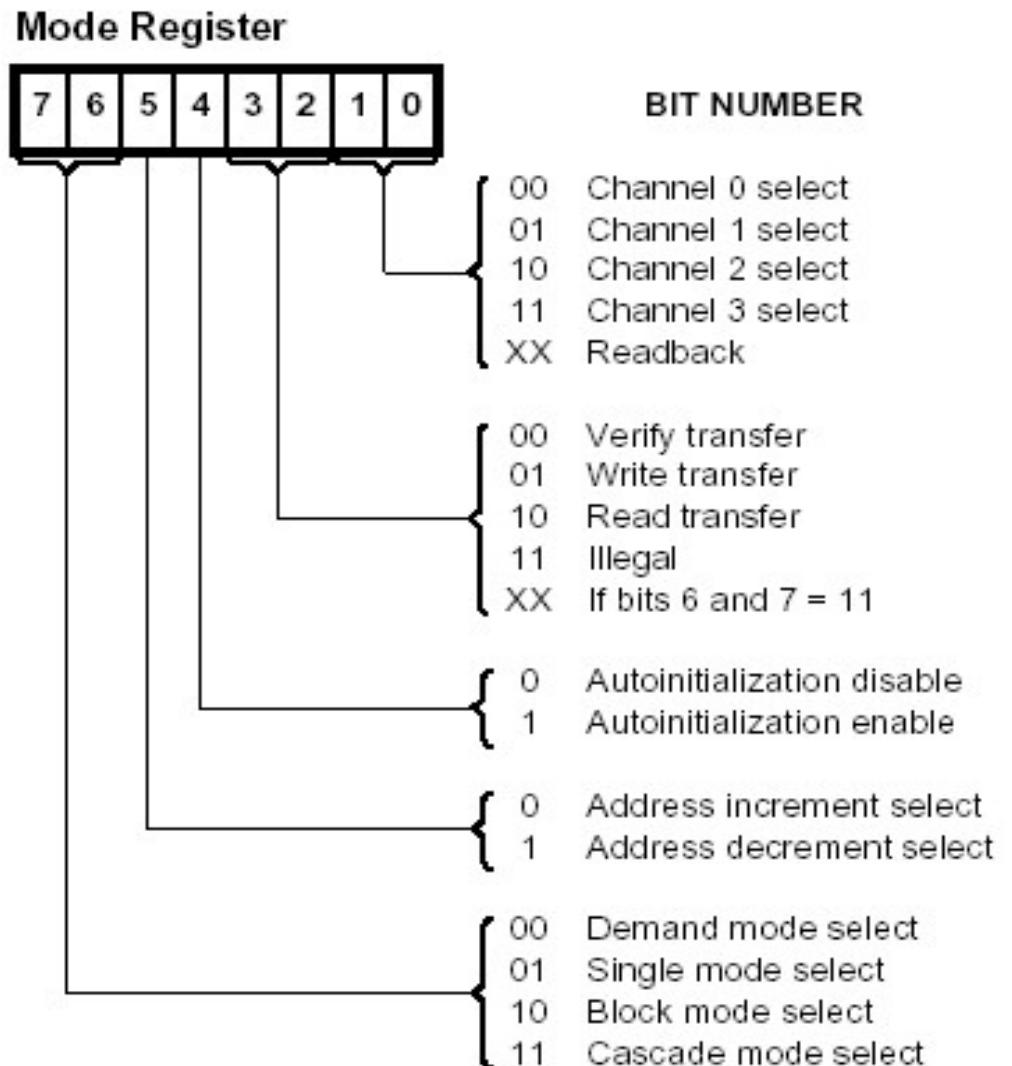
Each channel has 6 bit mode register

Read transfer : Memory to I/O

Write transfer: I/O to Memory

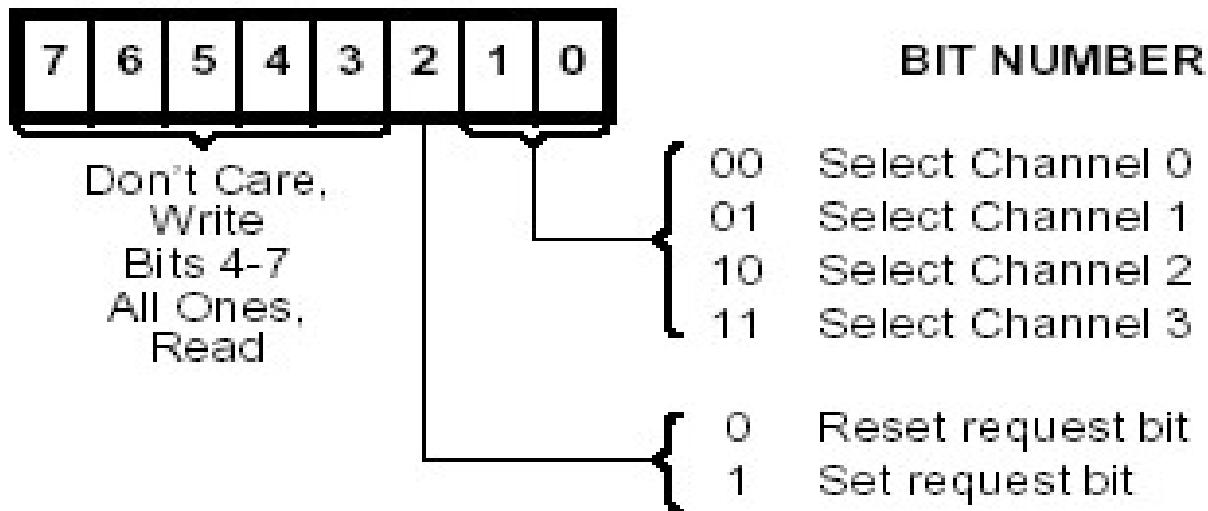
Verify transfer: Pseudo transfer mode

- Address and count register will be modified
- Responds to EOP
- But Mem and I/O control lines are inactive



Request Register

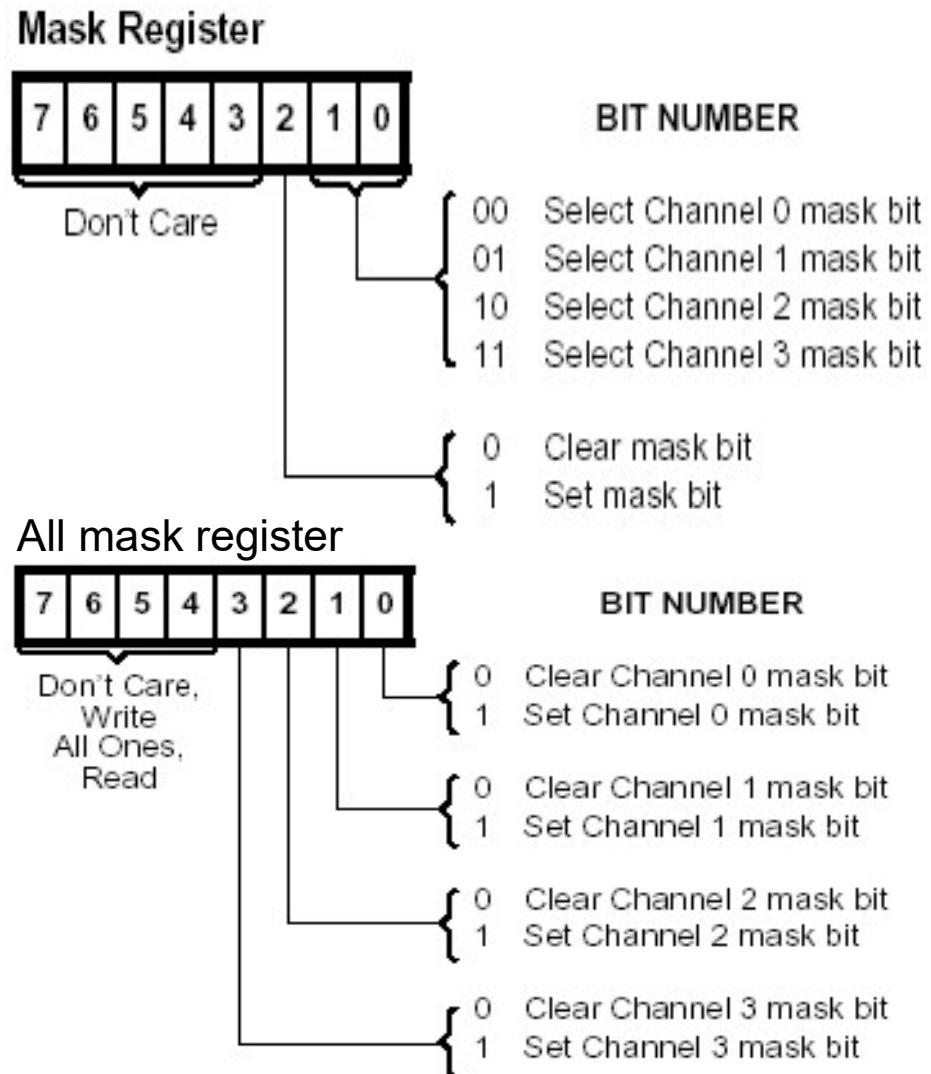
Request Register



Each channel has a request bit
Used for DMA request through S/I

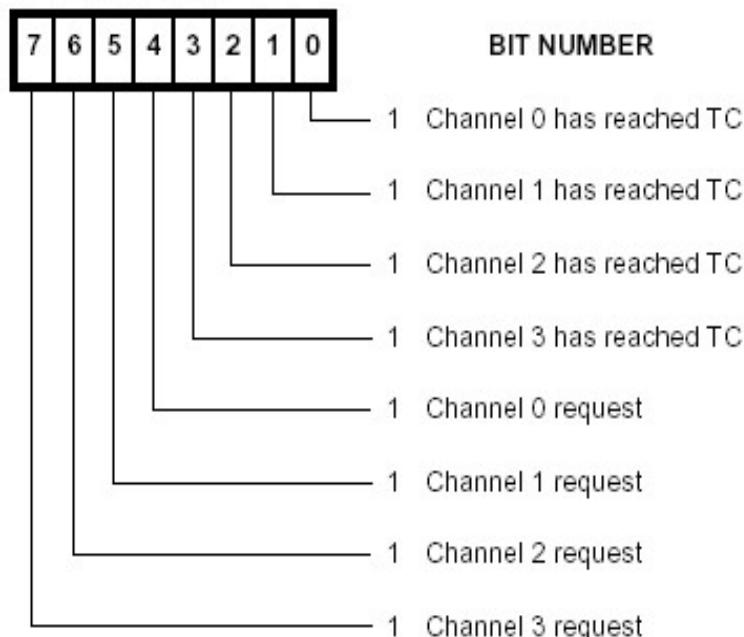
Mask Register

Each channel has a mask bit



Status Register

Status Register



Software commands

0C	WR Clear Byte Pointer F/F
0D	R/W Master Clear/Temp. Reg.
0E	WR Clear Mask Reg.

Clear First/Last Flip-Flop: This command must be executed prior to writing or reading new address or word count information to the 8237A.

Master Clear: This software instruction has the same effect as the hardware Reset. The Command, Status, Request, Temporary, and Internal First/Last Flip-Flop registers are cleared and the Mask register is set. The 8237A will enter the Idle cycle.

Clear Mask Register: This command clears the mask bits of all four channels, enabling them to accept DMA requests.

Initialization of 8237

- Write control word in mode register
- Write control word in command register
- Write starting address in channel's MAR
- Write count in count register

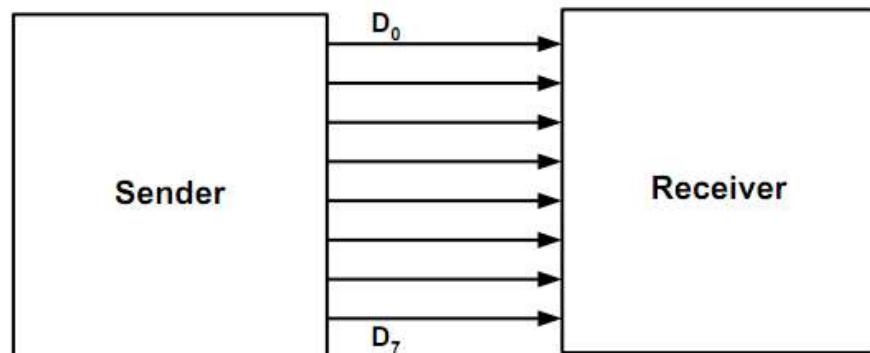
Serial Vs Parallel Communication

Serial Transfer



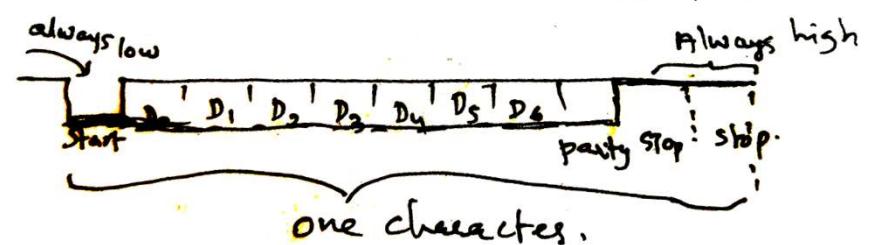
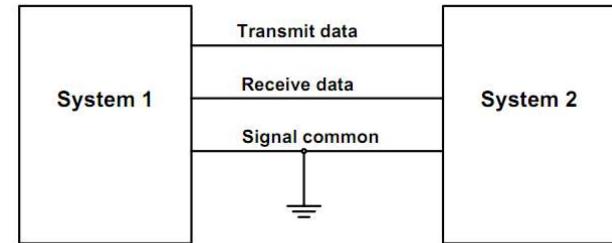
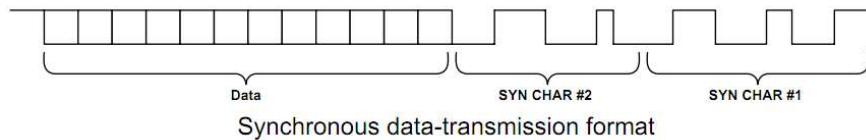
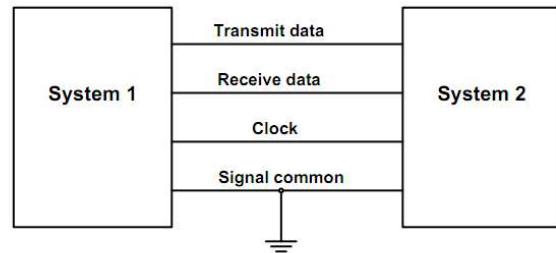
Serial communication uses a single line data.

Parallel Transfer

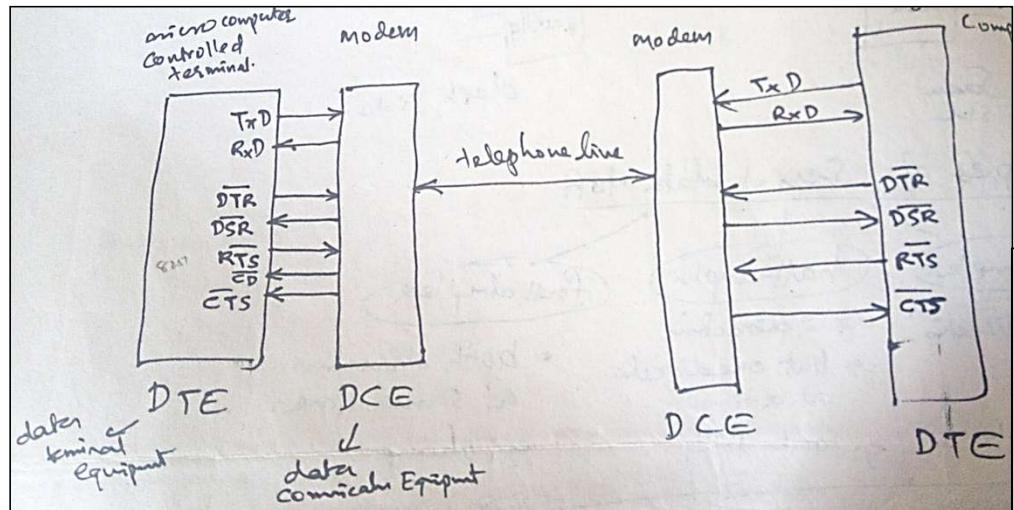


Parallel communication uses n-bit data line.

Synchronous Vs Asynchronous communication



DTE – DCE Handshaking

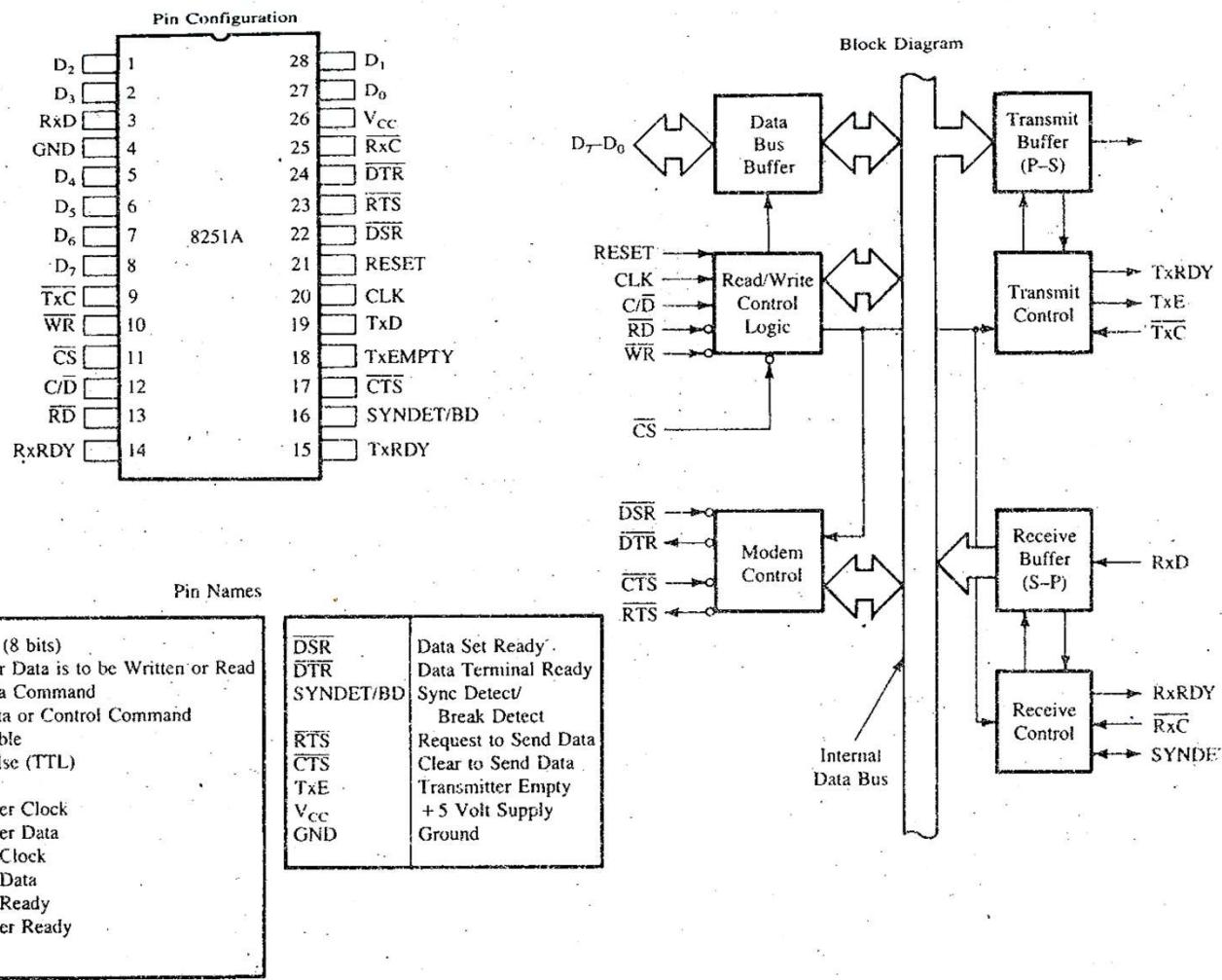


DTR – Data Terminal Ready
DSR – Data Set Ready
RTS – Request To Send
CD – Carrier Detect
CTS – Clear To Send

- **DTR** is to tell the modem that DTE is ready
- **DSR** is given when modem is ready to transmit / receive data, it gives **DSR** to **DTE**
- If **DTE** has a character to send then **RTS** is issued
- The modem will give **CD** signal to **DTE** to indicate that the connection with destination is established.
- **CTS** is issued, if the modem is fully ready to transmit the data.
- The terminal sends the serial data to modem through **TxD**
- After sending all data, **DTE** makes **RTS** high and modem makes **CTS** high.

8251 – Programmable Communication Interface - USART

- The 8251 USART (Universal Synchronous Asynchronous Receiver Transmitter) is capable of implementing either an asynchronous or synchronous serial data communication.
 - As a peripheral device of a microcomputer system, the 8251 receives parallel data from the CPU and transmits serial data after conversion.
 - This device also receives serial data from the outside and transmits parallel data to the CPU after conversion.



8251 – Pin Details

- **D0 - D7** - This is bidirectional data bus which receive control words and transmits data from the CPU and sends status words and received data to CPU.
- **RESET** - A "High" on this input forces the 8251 into "reset status". The min. reset width is six clock inputs during the operating status of CLK.
- **CLK** - CLK signal is used to generate internal device timing. CLK signal is independent of RXC or TXC.
- **WR** - This is the "active low" input terminal which receives a signal for writing transmit data and control words from the CPU into the 8251.
- **RD** - This is the "active low" input terminal which receives a signal for reading receive data and status words from the 8251.
- **C/D** - This is an input terminal which receives a signal for selecting data or command words and status words when the 8251 is accessed by the CPU.
- **CS** - This is the "active low" input terminal which selects the 8251 at low level when the CPU accesses.



8251 – Programmable Communication Interface - USART

- TXD - This is an output terminal for transmitting data from which serial-converted data is sent out.
- TXRDY - This is an output terminal which indicates that the 8251 is ready to accept a data character.
- TXEMPTY - This is an output terminal which indicates that the 8251 has transmitted all the characters and had no data character.
- TXC - This is a clock input signal (Active Low) which determines the transfer speed of transmitted data.
 - In "synchronous mode," the baud rate will be the same as the frequency of TXC. In "asynchronous mode", it is possible to select the baud rate factor by mode instruction. It can be 1, 1/16 or 1/64 the TXC.
- RXD - This is a terminal which receives serial data.
- RXRDY - This is a terminal which indicates that the 8251 contains a character that is ready to be READ.
 - If the CPU reads a data character, RXRDY will be reset by the leading edge of RD signal. Unless the CPU reads a data character before the next one is received completely, the preceding data will be lost. In such a case, an overrun error flag status word will be set.
- RXC - This is a clock input signal which determines the transfer speed of received data.
 - In "synchronous mode," the baud rate is the same as the frequency of RXC. In "asynchronous mode," it is possible to select the baud rate factor by mode instruction. It can be 1, 1/16, 1/64 the RXC.

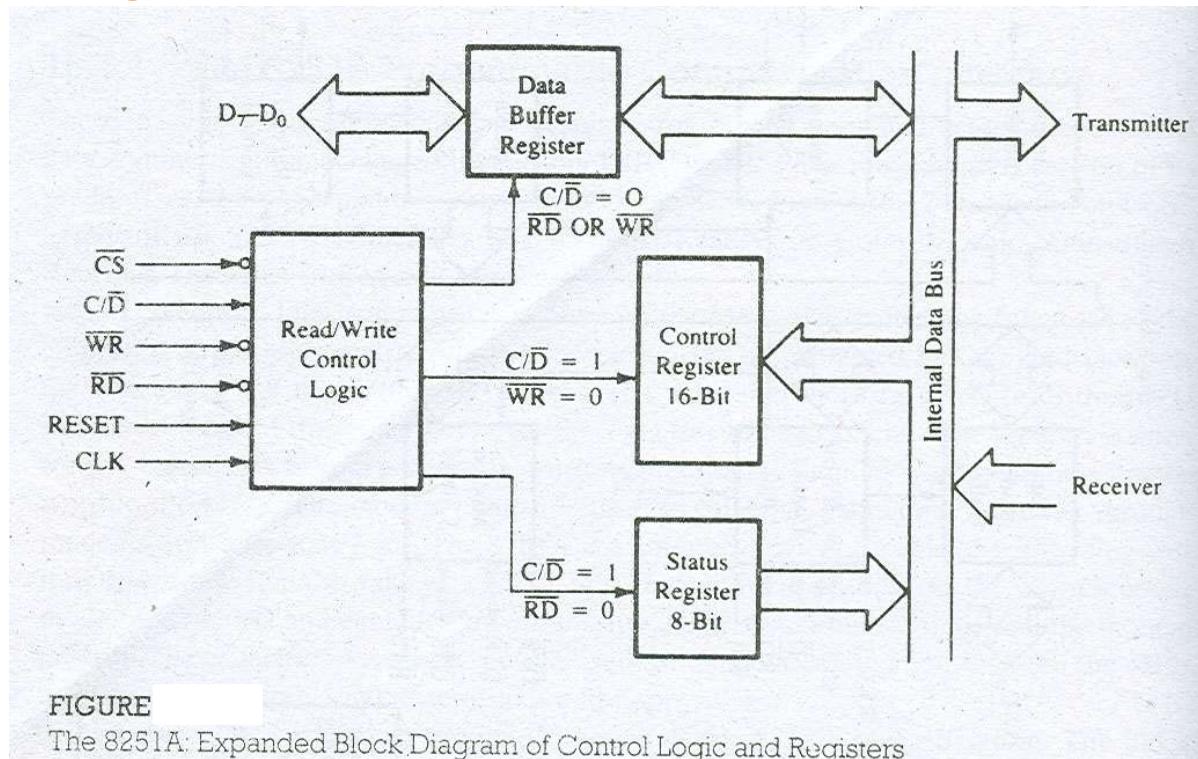


8251 – Programmable Communication Interface - USART

- SYNDET/BD - This is a terminal whose function changes according to mode.
 - In “internal synchronous mode”, this terminal is output line. High, if sync characters are received.
 - In “external synchronous mode”, this is an input terminal. A "High" on this input forces the 8251 to start receiving data characters on the next rising edge of RxC.
 - In “asynchronous mode”, this is an output terminal which generates "high level" output upon the detection of a "break" character. i.e. if receiver data contains a "low-level" for two continuous character times. One character time includes start bit, data bits and parity bit.
 - The terminal will be reset, if RXD is at high level. After Reset is active, the terminal will be output at low level.
- DSR - This is an input port for MODEM interface. The input status of the terminal can be recognized by the CPU reading status words.
- DTR - This is an output port for MODEM interface. It is possible to set the status of DTR by a command.
- CTS - This is an input terminal for MODEM interface which is used for controlling a transmit circuit.
 - The terminal controls data transmission if the device is set in "TX Enable" status by a command. Data is transmittable if the terminal is at low level.
- RTS - This is an output port for MODEM interface. It is possible to set the status RTS by a command.



8251 – Programmable Communication Interface



FIGURE

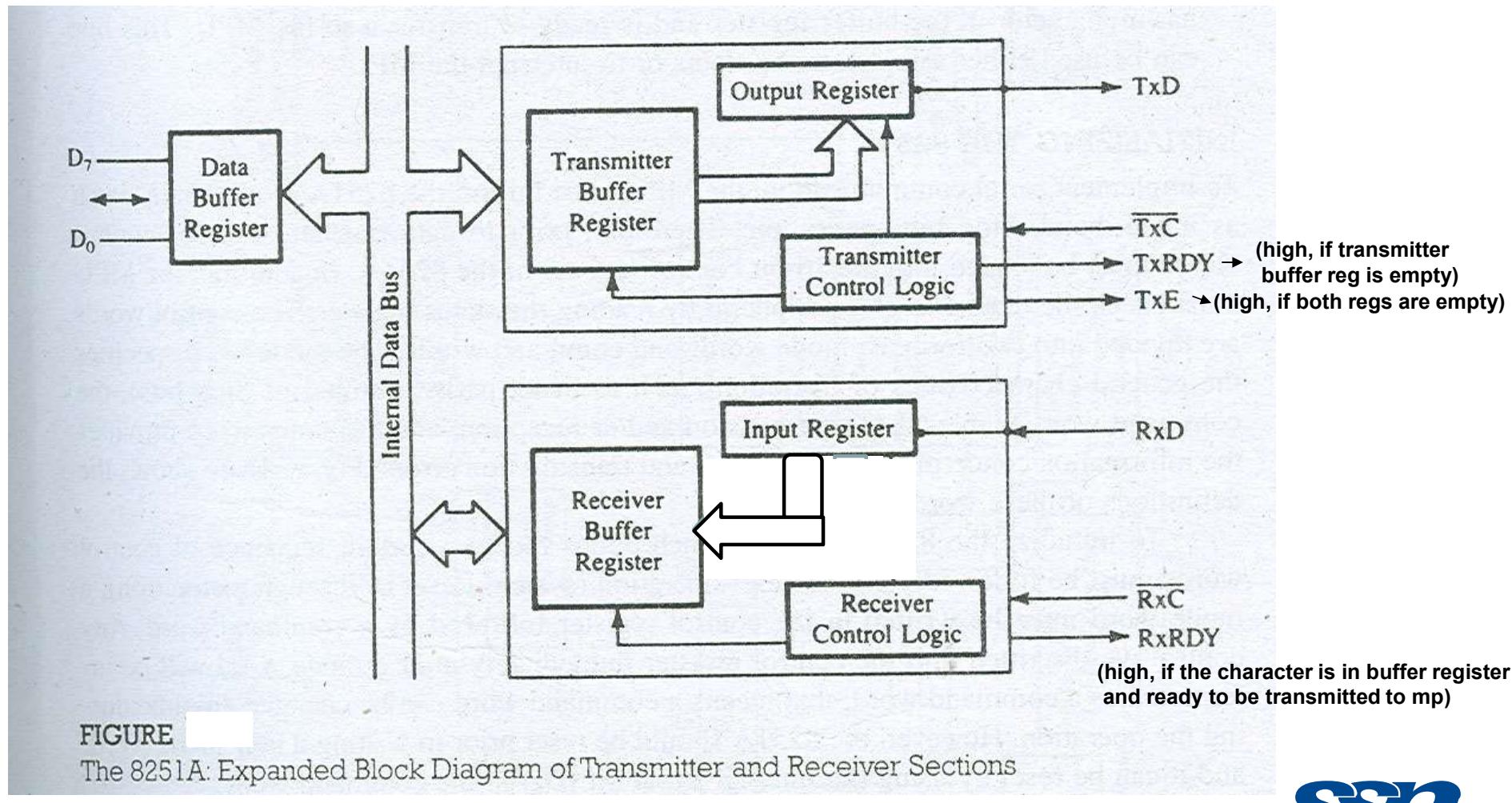
The 8251A: Expanded Block Diagram of Control Logic and Registers

TABLE

Summary of Control Signals for the 8251A

CS	C/D	RD	WR	Function
0	1	1	0	MPU writes instructions in the control register
0	1	0	1	MPU reads status from the status register
0	0	1	0	MPU outputs data to the Data Buffer
0	0	0	1	MPU accepts data from the Data Buffer
1	X	X	X	USART is not selected

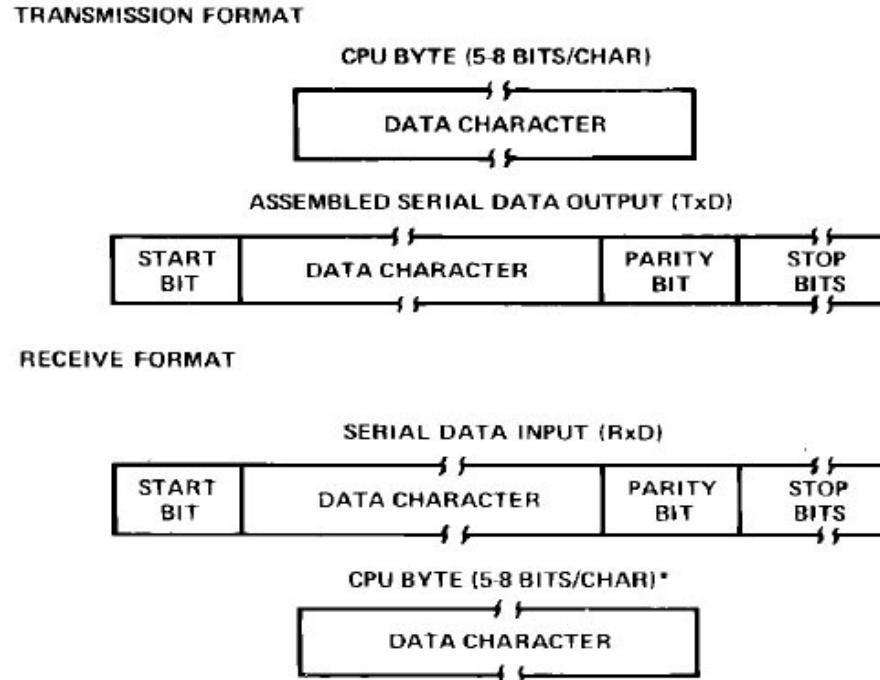
8251 – Programmable Communication Interface



FIGURE

The 8251A: Expanded Block Diagram of Transmitter and Receiver Sections

8251 – Programmable Communication Interface



205222-9

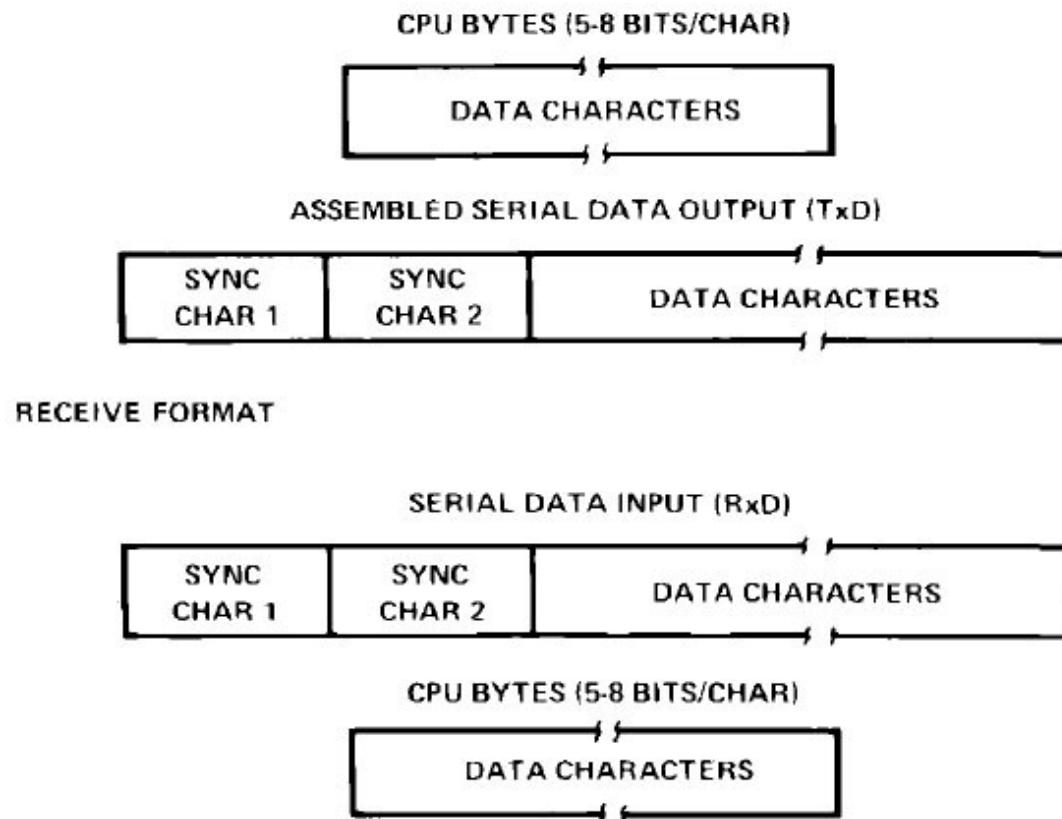
***NOTE:**

If character length is defined as 5, 6, or 7 bits the unused bits are set to "zero".

Figure 9. Asynchronous Mode



8251 – Programmable Communication Interface



205222-12

Figure 11. Data Format, Synchronous Mode



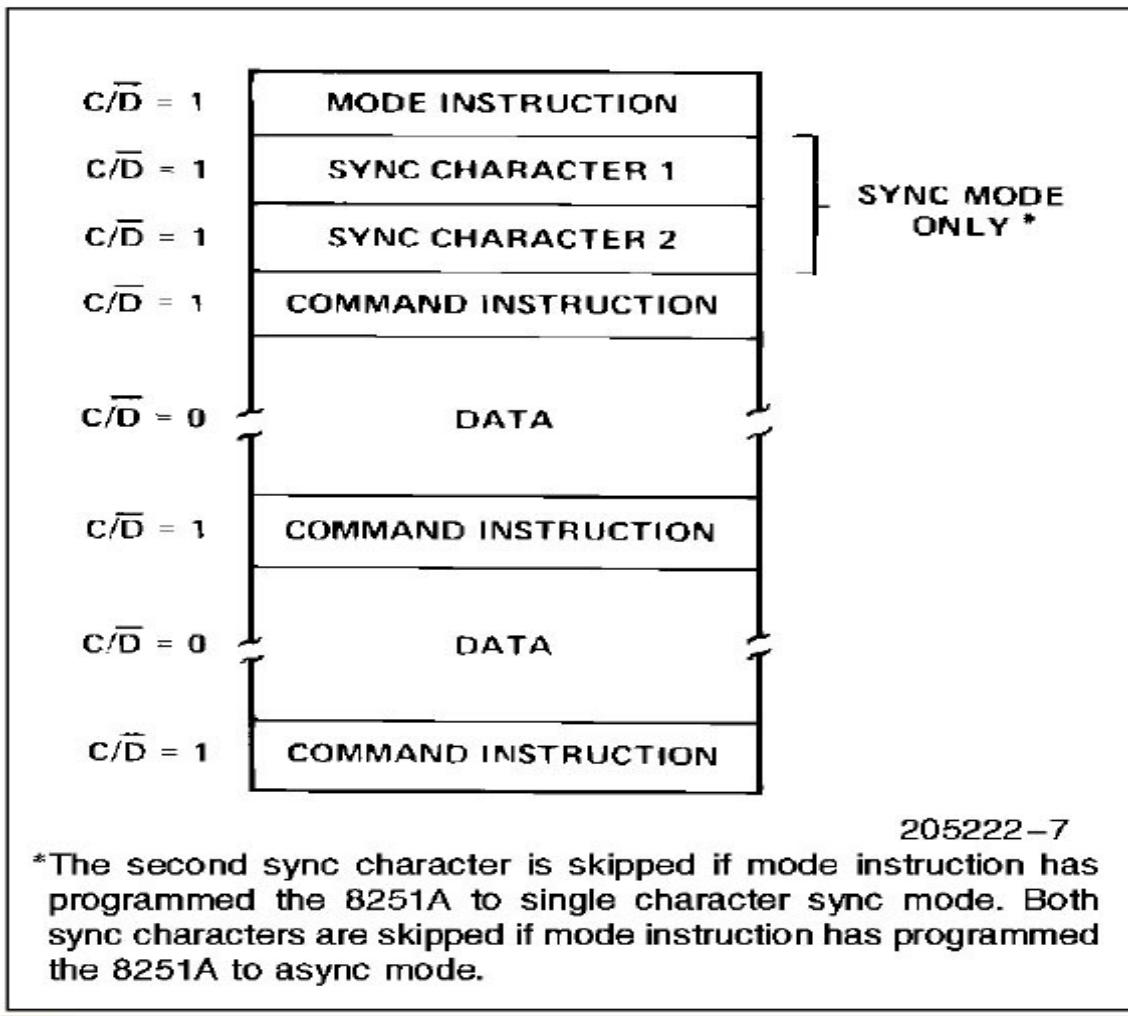
Programming the 8251A

Prior to starting data transmission or reception, the 8251A must be loaded with a set of control words generated by the CPU. These control signals define the complete functional definition of the 8251A and must immediately follow a Reset operation (internal or external).

The control words are split into two formats:

1. Mode Instruction
2. Command Instruction

8251 – Programmable Communication Interface



8251 – Programmable Communication Interface

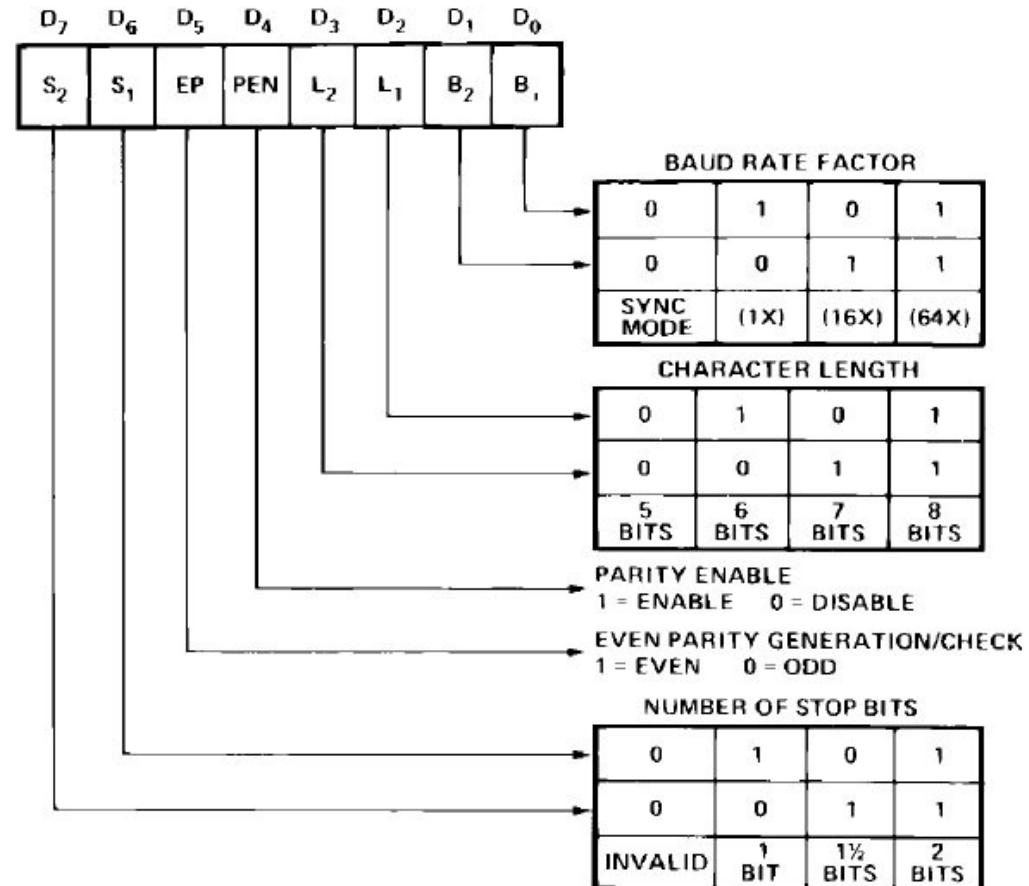


Figure 8. Mode Instruction Format, Asynchronous Mode

8251 – Programmable Communication Interface

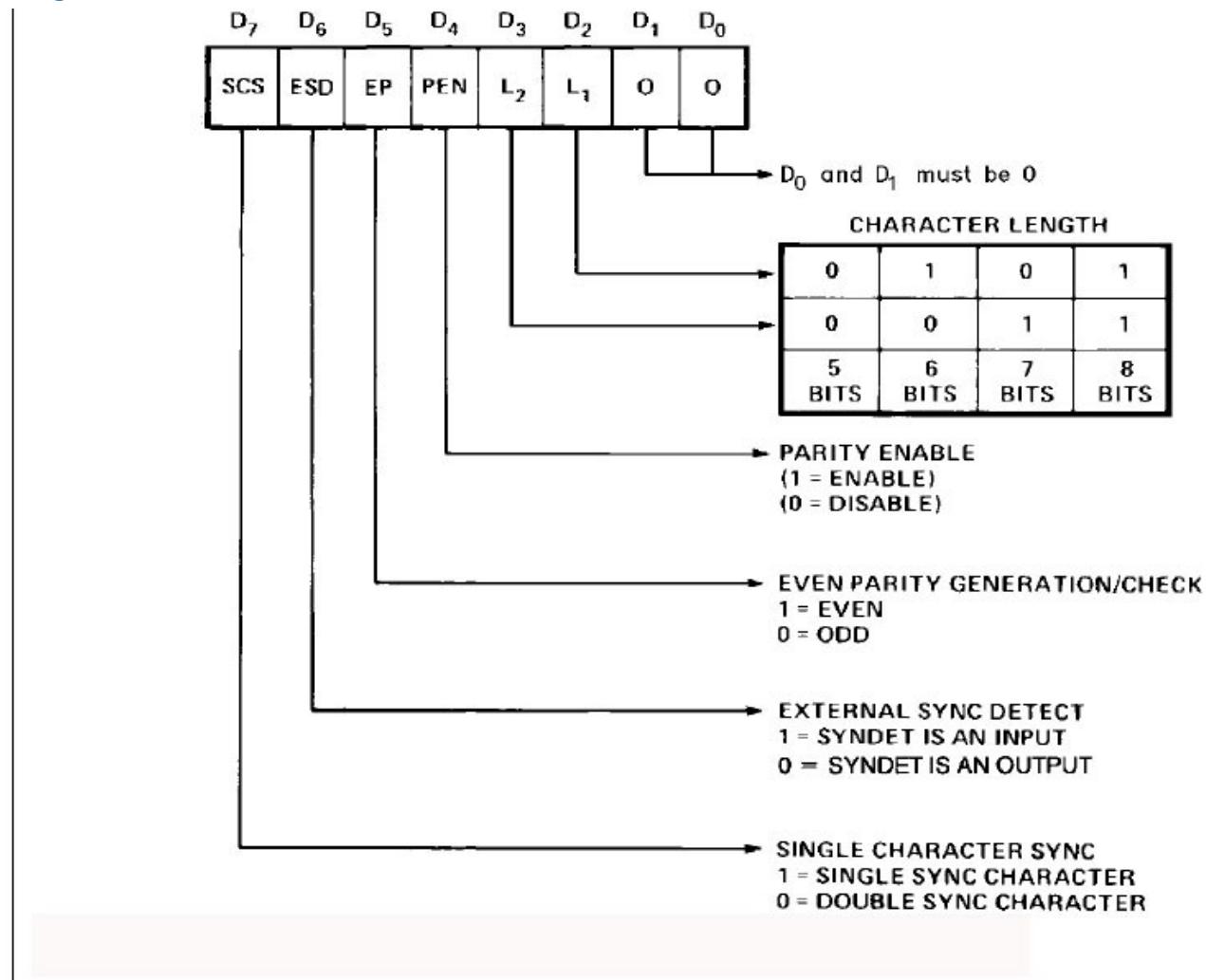
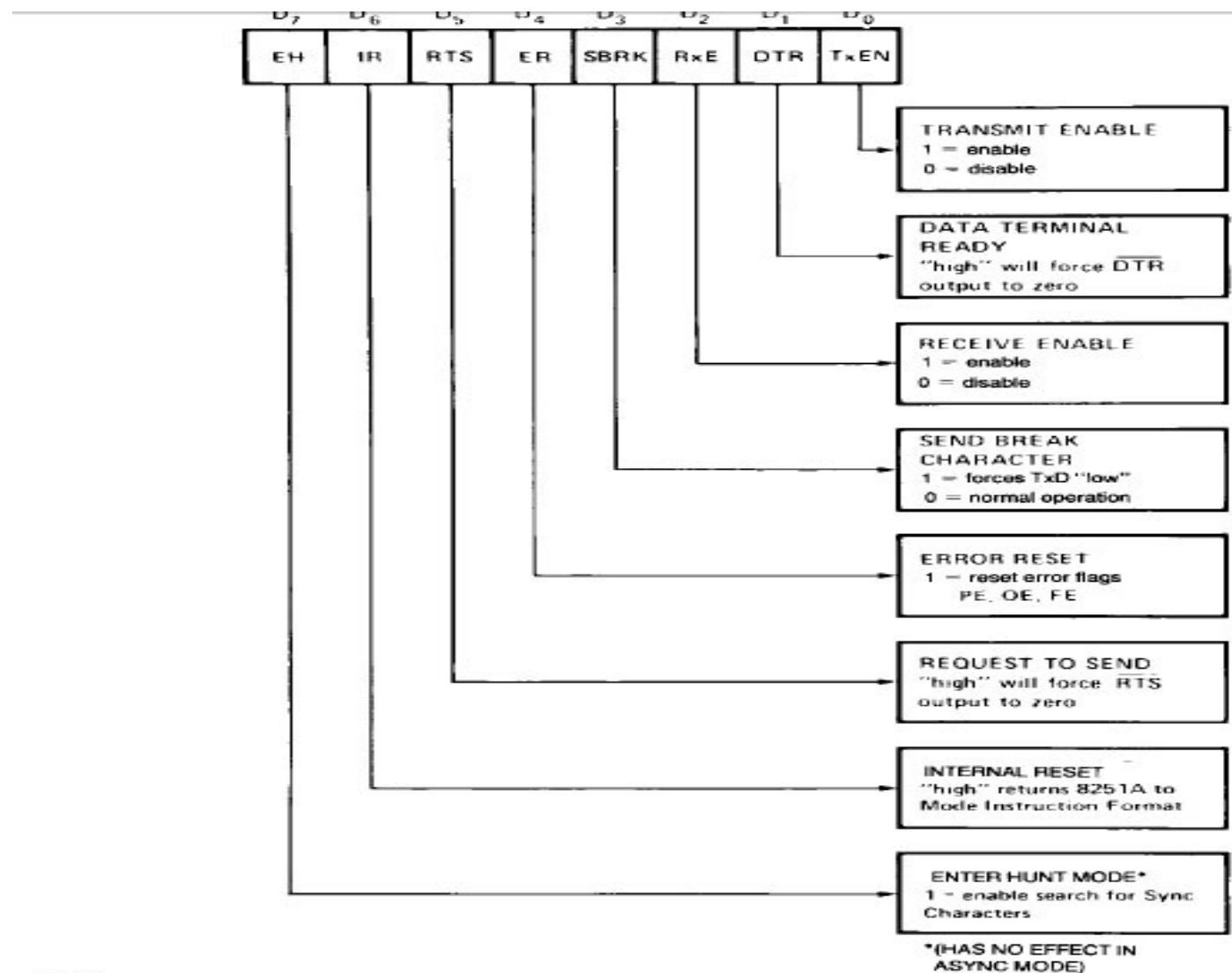


Figure 10. Mode Instruction Format, Synchronous Mode



8251



NOTE:

Error Reset must be performed whenever RxEnable and Enter Hunt are programmed.

Figure 12. Command Instruction Format

8251

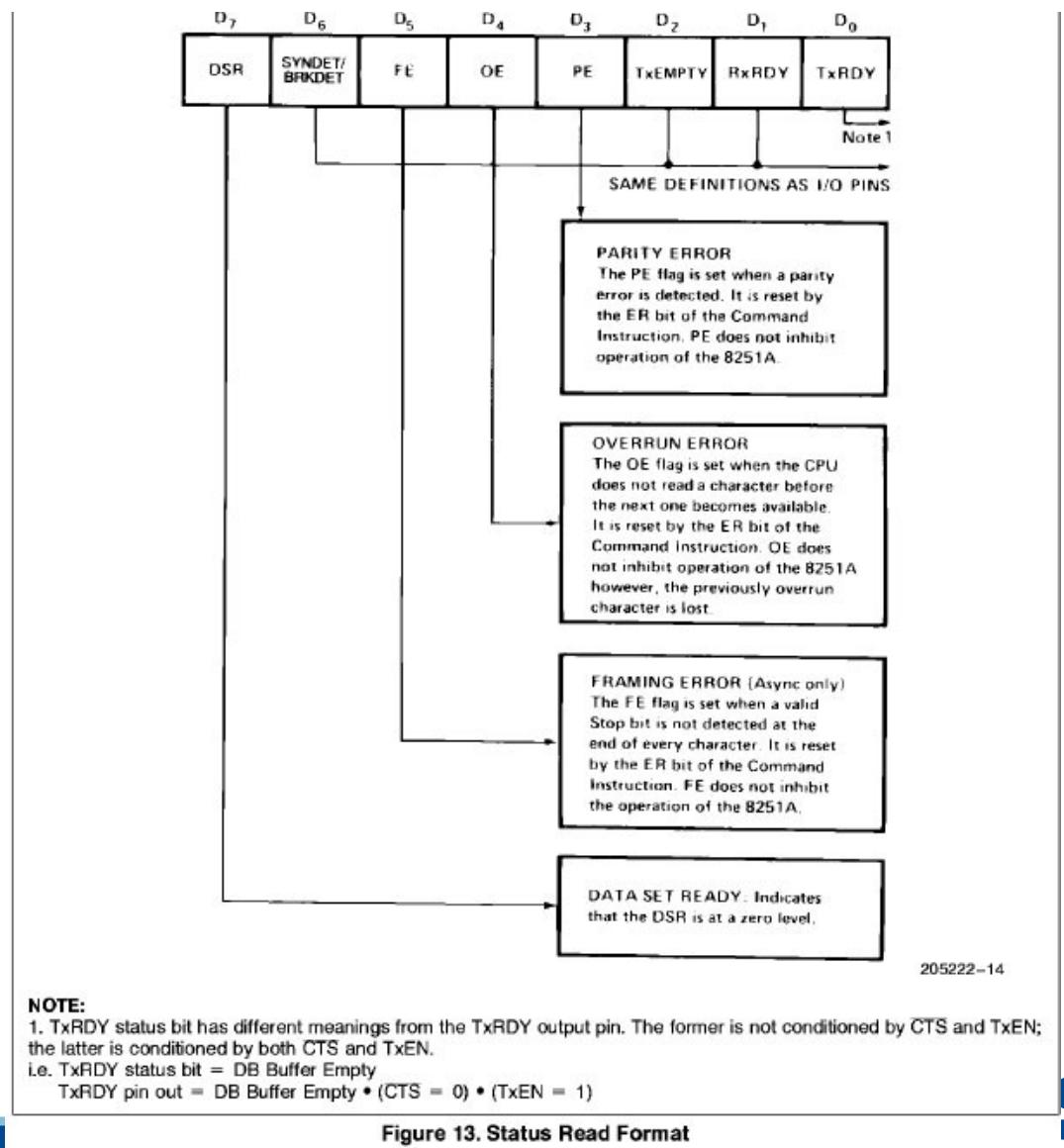


Figure 13. Status Read Format

ssn

TEST YOUR UNDERSTANDING

- What is PPI? What are its applications?
- Draw the control word format of 8255.
- What are the modes of operation of Timer?

Summary

discussed about :

- 8255 – Programmable Peripheral Interface (PPI)
- 8254 - Programmable Timer Interface
- 8279 - Programmable Keyboard & Display Interface
- 8259 - Programmable Interrupt Controller
- 8237 - Programmable DMA Controller
- 8251 –Programmable Communication Interface
- Memory and IO interfacing

References

- Douglas V. Hall, "Microprocessors and Interfacing, Programming and Hardware", Second Edition, TMH.
- 8255 Intel data sheet
- Microprocessor Architecture, Programming, and Applications with the 8085, 5th edition, Ramesh Gaonkar.

Thank you

