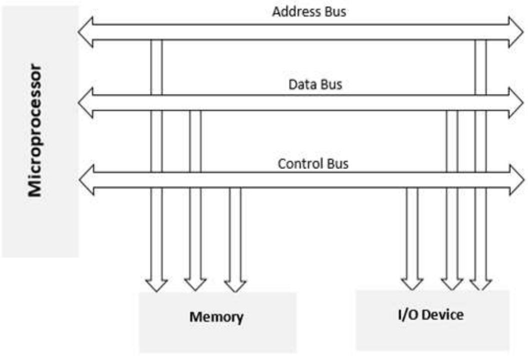
UNIT-2
8086 INTERFACING

1. Programmable Peripheral Interface 8255,
2. Programmable Interval Timer 8253,
3. Programmable Interrupt Controller 8259,
4. Programmable Communication Interface 8251 USART,
5. DMA Controller 8257.
6. Case Study:
   1. 8255 – PPI and its interfacing program– Stepper motor interfacing
   2. Interfacing of 7-Segment Display with 8086 microprocessor using 8255.

What is INTERFACE?

Interface is the path for communication between two components. Interfacing is of two types, **memory interfacing and I/O interfacing**.



Memory Interfacing:

When we are executing any instruction, we need the microprocessor to access the memory for reading instruction codes and the data stored in the memory. For this, both the memory and the microprocessor require some signals to read from and write to registers. If we are interfacing memory to a microprocessor, then it will be called as **Memory Interfacing.**

I/O Interfacing:

There are various communication devices like the keyboard, mouse, printer, etc. So, we need to interface the keyboard and other devices with the microprocessor by using latches and buffers. This type of interfacing is known as **I/O interfacing.**

8255A PPI-PROGRAMMABLE PERIPHERAL INTERFACING DEVICE

The Intel 82C55A is a high-performance, CHMOS version of the industry standard 8255A general purpose programmable I/O device which is designed for use with all Intel and most other microprocessors. It provides 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation.

Features of 8255

- It is Compatible with all Intel and Most Other Microprocessors
- It has High Speed, ``Zero Wait State'' Operation with 8 MHz 8086/88 and 80186/188
- Provides 24 Programmable I/O Pins
- Low Power CHMOS
- Completely TTL Compatible
- Control Word Read-Back Capability
- Direct Bit Set/Reset Capability
- 2.5 mA DC Drive Capability on all I/O Port Outputs
- It is available in 40-Pin DIP.

Internal Architecture of 8255A:

The parallel input-output (PIO) port chip 8255A is also called as **programmable**

peripheral input-output port. The Intel's 8255A is designed for use with Intel's 8-bit, 16-bit and higher capability microprocessors. It has 24 input/output lines which may be individually programmed in two groups of twelve lines each, or three groups of eight lines.

All of these ports can function independently either as input or as output ports. This can be achieved by programming the bits of an internal register of 8255 called as control word register (CWR)

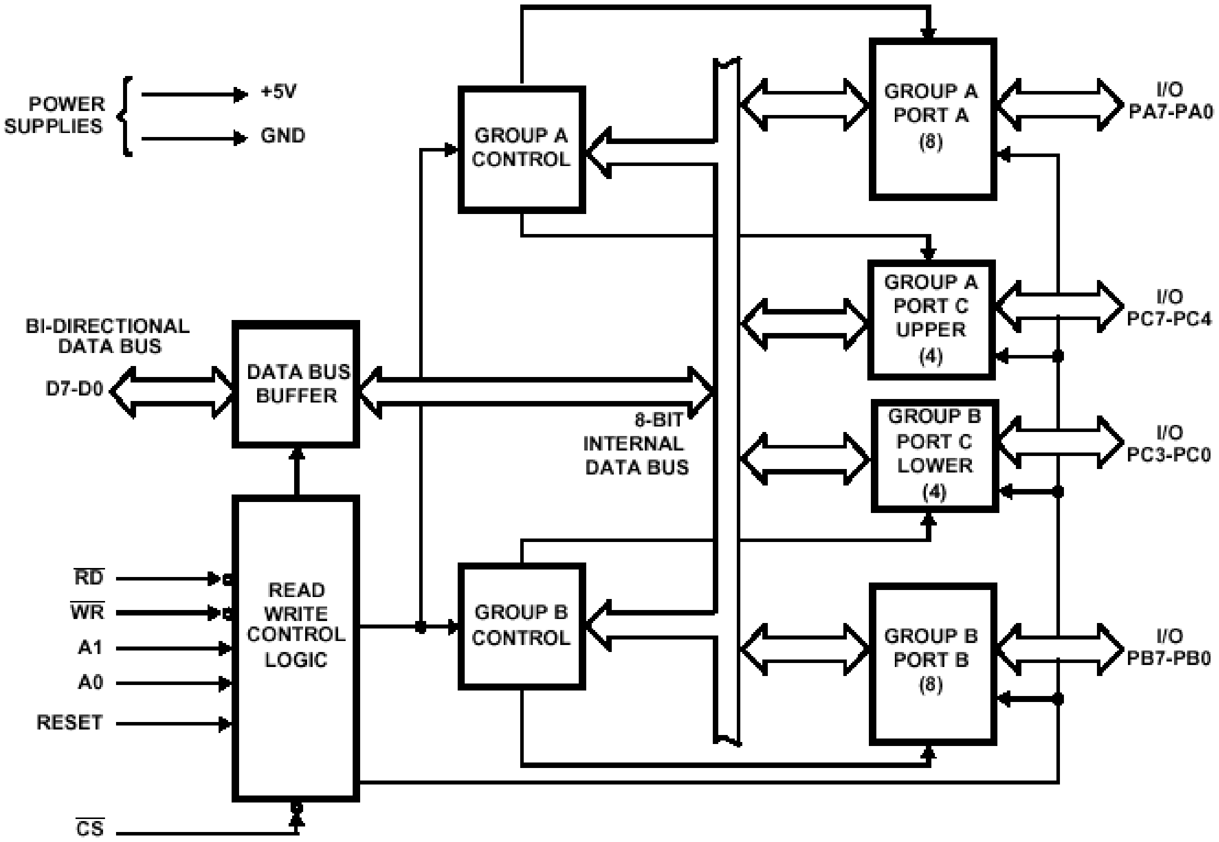


Fig. Internal Block diagram of 8255A PPI

Block Diagram Description:

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 8255A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the microprocessor. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the microprocessor Address and Control busses and in turn, issues commands to both of the Control Groups.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the microprocessor ``outputs'' a control word to the 8255A. The control word contains information such as ``mode'', ``bit set'', ``bit reset'', etc., that initializes the functional configuration of the 8255A. Each of the Control blocks (Group A and Group B) accepts ``commands'' from the Read/Write Control Logic, receives ``control words'' from the internal data bus and issues the proper commands to its associated ports.

Control Group A - Port A and Port C upper (C7-C4)

Control Group B - Port B and Port C lower (C3-C0)

Ports A, B, and C

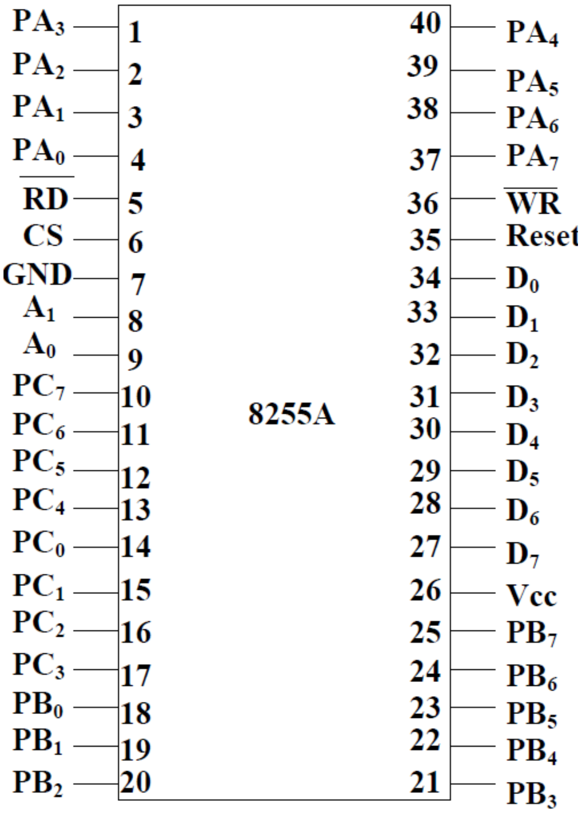
The 8255A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or ``personality'' to further enhance the power and flexibility of the 8255A.

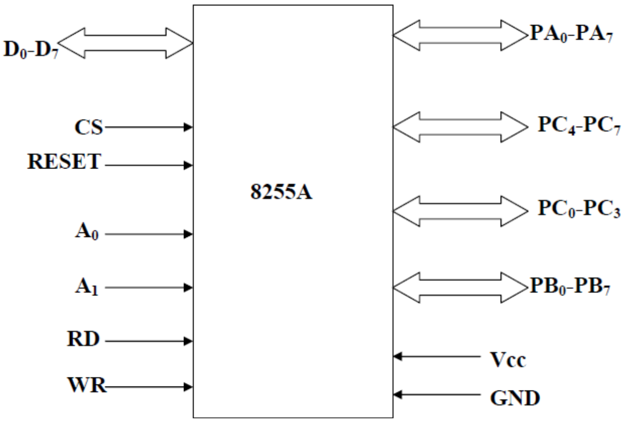
Port A- One 8-bit data output latch/buffer and one 8-bit input latch buffer.

Port B- One 8-bit data input/output latch/buffer.

Port C-One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B. Only

PIN DIAGRAM OF 8255A





8255A Pin Configuration

The signal descriptions of 8255 are briefly presented as follows:

PA7-PA0: These are eight port A lines that acts as either output or input lines depending upon the control word loaded into the control word register.

PC7-PC4: Upper nibble of port C lines. They may act as either output or input lines. This port also can be used for generation of handshake lines in mode 1 or mode 2.

PC3-PC0: These are the lower port C lines, other details are the same as PC7-PC4 lines.

PB0-PB7: These are the eight port B lines which are used as latched output lines or buffered input lines in the same way as port A.

RD': This is the input line driven by the microprocessor and should be low to indicate read operation to 8255.

WR': This is an input line driven by the microprocessor. A low on this line indicates write operation.

CS': This is a chip select line. If this line goes low, it enables the 8255 to respond to RD and WR signals, otherwise RD and WR signal are neglected.

- **A1-A0:** These are the address input lines and are driven by the microprocessor. These lines A1-A0 with RD, WR and CS form the following operations for 8255. These address lines are used for addressing any one of the four registers, i.e. three ports and a control word register as given in table below.

$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{CS}}$	A_1	A_0	Input (Read) cycle
0	1	0	0	0	Port A to Data bus
0	1	0	0	1	Port B to Data bus
0	1	0	1	0	Port C to Data bus
0	1	0	1	1	CWR to Data bus

$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{CS}}$	A_1	A_0	Output (Write) cycle
1	0	0	0	0	Data bus to Port A
1	0	0	0	1	Data bus to Port B
1	0	0	1	0	Data bus to Port C
1	0	0	1	1	Data bus to CWR

$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{CS}}$	A_1	A_0	Function
X	X	1	X	X	Data bus tristated
1	1	0	X	X	Data bus tristated

In case of 8086 systems, if the 8255 is to be interfaced with lower order data bus, the A0 and A1 pins of 8255 are connected with A1 and A2 respectively.

D0-D7: These are the data bus lines those carry data or control word to/from the microprocessor.

RESET: A logic high on this line clears the control word register of 8255. All ports are set as input ports by default after reset.

MODES OF OPERATION OF 8255A PPI:

These are two basic modes of operation of 8255.

1. Bit Set-Reset mode (BSR).

2. I/O mode

In BSR mode only port C (PC0-PC7) can be used to set or reset its individual port bits. In I/O mode, the 8255 ports work as programmable I/O ports. Under the I/O mode of operation, further there are three modes of operation of 8255, so as to support different types of applications.

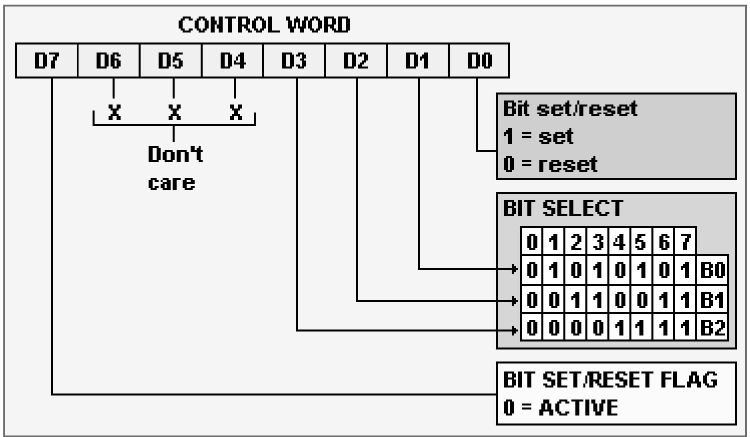
a. Mode 0 (Basic I/O mode)

b. Mode 1 (Strobed input/output mode)

c. Mode 2 (Strobed bidirectional I/O)

All these modes can be selected by programming a register internal to 8255 known as CWR (Control Word Register) which has two formats. One is for BSR mode of operation and second one is for I/O mode of operation.

BSR Mode: In this mode any of the 8-bits of port C can be set or reset depending on D0 of the control word.

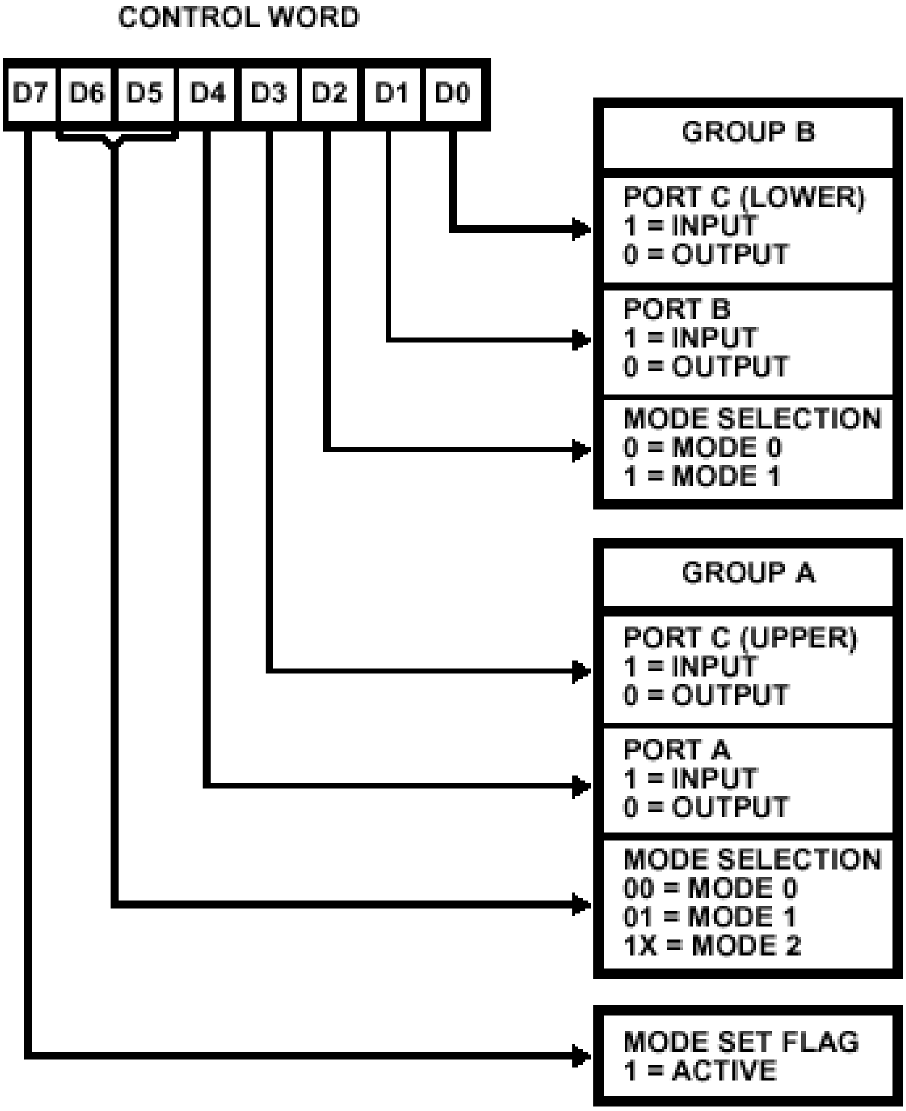


0	1	2	3	4	5	6	7	
0	1	0	1	0	1	0	1	B0
0	0	1	1	0	0	1	1	B1
0	0	0	0	1	1	1	1	B2

The bit to be set or reset is selected by bit select flags D3, D2 and D1 of the CWR as given in table.

Fig. Control Word Register Format for BSR mode

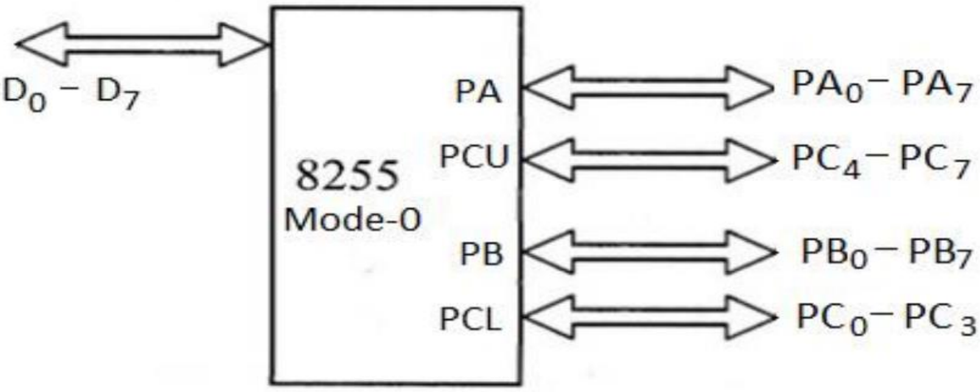
I/O Modes: The Control Word Register format for I/O mode of operation is shown below.



I/O modes :

Mode-0 : Simple I/O → PORT-A, PORT-B & PORT-C
Mode-1 : Handshaking I/O → PORT-A, PORT-B
Mode-2 : Bi-directional Handshaking I/O → only PORT-A can be operated

(i) Mode-0 : Simple I/O



- This mode provides simple input and output capability.
- All 3- ports can be operated in Mode-0

  PORT-A → 8-bit I/O port

  POTR-B → 8-bit I/O port

  PORT-C → Two 4-bit I/O ports → PORT-C upper & PORT-C lower
- Each port can be programmed to function as either input (or) output port
- This mode doesn't support handshaking and interrupt capability

Mode-1 : Handshaking I/O

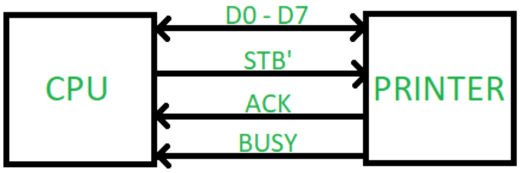
- In this mode, I/O data transfer is controlled by handshaking signals
- Only PORT-A & PORT-B can be operated in Mode-1
- Each port uses 3-lines from PORT-C as Handshaking signals.

  For PORT-B → PC_0 , PC_1 , PC_2 are used as Handshaking lines

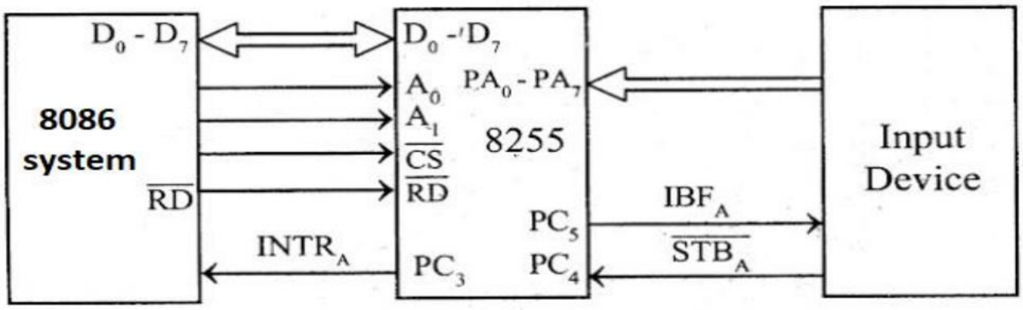
  For PORT-A → input port → PC_3 , PC_4 , PC_5

  output port → PC_3 , PC_6 , PC_7
- The remaining 2-line of PORT-C can be used as I/O lines
- This mode supports Interrupt logic.

Handshake signals are used for synchronization between CPU & peripherals.



Input operation:

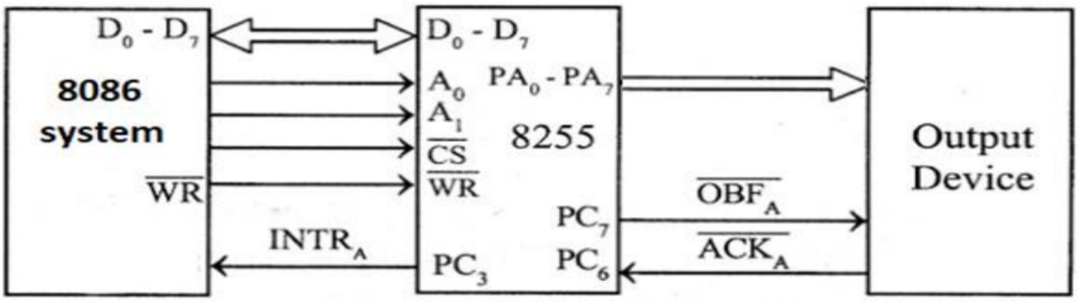


$\overline{\textbf{STB}}$ **(Strobe) :** It is active low input signal from input device. It indicates the 8255 that the data is placed on the port lines.

IBF (Input Buffer Full) : It is active high output signal from 8255 to input device. It indicates the input device that the input buffer is full and it is not ready to accept next byte from input device. This signal is deactivated when CPU reads data from input port of respective port, i.e., at the rising edge of RD signal

INTR (Interrupt Request) : It is active high output signal from 8255 to CPU. It indicated the CPU that the data from input device is available in input buffer. It is set when IBF is active and it is reset when RD signal is issued by CPU

Output operation:



$\overline{\textbf{OBF}}$ **(Output Buffer Full) :** It is active low output signal from 8255 to output device. It indicates the output device that the data is available on the port lines. Upon the activation of the OBF signal, the output device reads data from port lines

$\overline{\textbf{ACK}}$ **(Acknowledge) :** It is active low input signal from output device. It indicates the 8255 that the data from port lines has been received by output device

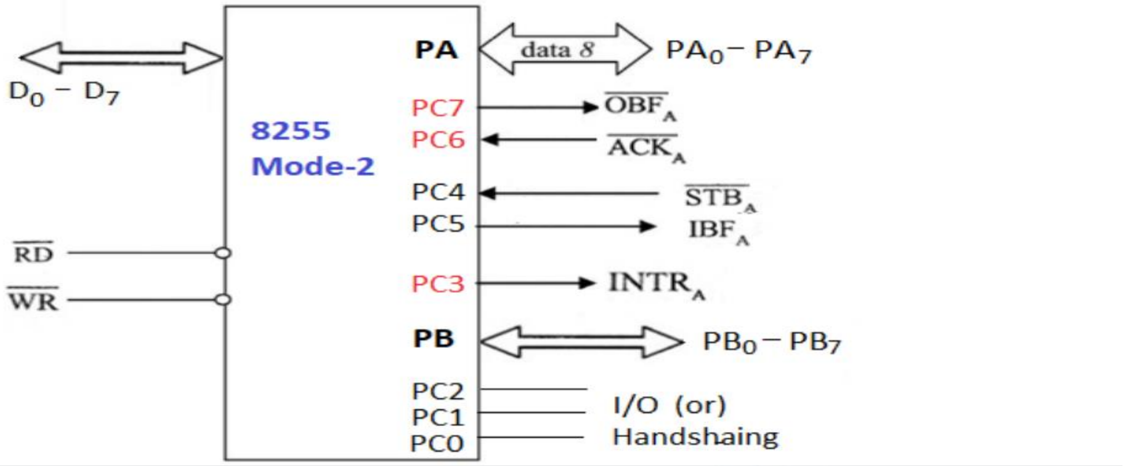
INTR (Interrupt Request) : It is active high output signal from 8255 to CPU. It indicated the CPU that the output device is ready to accept next data byte. It is set when ACK signal is active and it is reset when WR signal is issued by CPU

Mode-2 : Bi-directional Handshaking I/O

- Only PORT-A can be operated in Mode-2 as bi-directional handshaking I/O port
- The Five PORT-C lines PC_3 - PC_7 are used as Handshake signals

  PORT-A → input port → PC_3 , PC_4 , PC_5

  output port → PC_3 , PC_6 , PC_7
- The PORT-B can be operated in Mode-0 (or) Mode-1.
- When PORT-B is operated in Mode-1, the PORT-C lines PC_0 , PC_1 , PC_2 are used as Handshaking signals. Otherwise, these lines can be used as simple I/O lines



The common applications of 8255 are:

- Traffic light control
- Generating square wave (DAC)
- Interfacing Sensors (ADC)
- Interfacing with DC motors and stepper motors

8253 (PPI)-PROGRAMMABLE INTERVAL TIMER/ COUNTER:

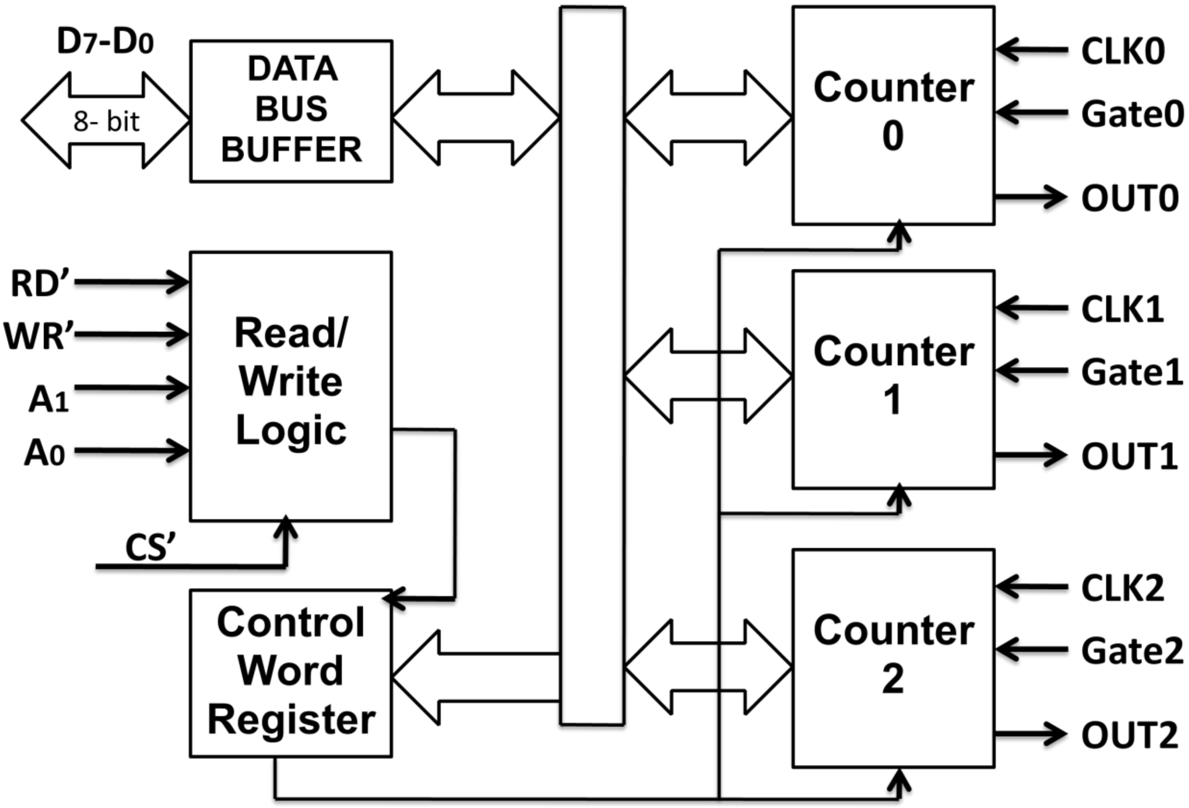
Generally it is not possible to generate an arbitrary time delay using delay routines. INTEL Programmable Timer/Counter 8253 facilitates the Generation of accurate time delay. When 8253 is used as a timing and delay generation peripheral, then the microprocessor becomes free from executing delay routines.

FEATURES OF 8253

- Compatible with all Intel and most other microprocessors.
- It has three independent 16-bit down counters.
- Each counter with a count rate up to 2.6 MHz
- Three counters are identical presettable and can be programmed for either binary or BCD count.
- Counter can be programmed in six different modes.
- It is available in 24 pin DIP
- It uses N-MOS technology with +5v single power supply.

INTERNAL BLOCK DIAGRAM OF 8253

The internal block diagram of 8253 are shown in below figure.



DATA BUS BUFFER:

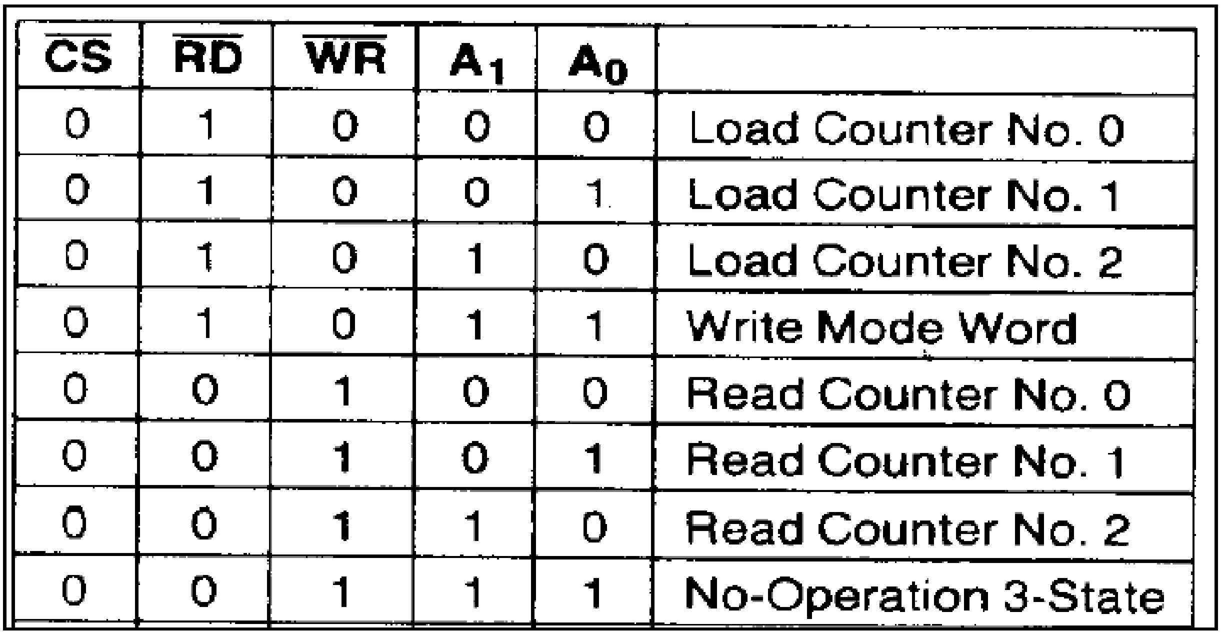
This tri-state, bi-directional, 8-bit data bus is used to interface the 8253 to the system data bus. The Data bus buffer has three basic functions.

1. Programming the modes of 8253.
2. Loading the count registers.
3. Reading the count values.

READ/WRITE LOGIC:

The Read/Write logic has five signals: RD, WR, CS and the address lines A0 and A1.

The control word register and counters are selected according to the signals on lines A0 and A1.



$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A_1	A_0	
0	1	0	0	0	Load Counter No. 0
0	1	0	0	1	Load Counter No. 1
0	1	0	1	0	Load Counter No. 2
0	1	0	1	1	Write Mode Word
0	0	1	0	0	Read Counter No. 0
0	0	1	0	1	Read Counter No. 1
0	0	1	1	0	Read Counter No. 2
0	0	1	1	1	No-Operation 3-State

CONTROL WORD REGISTER:

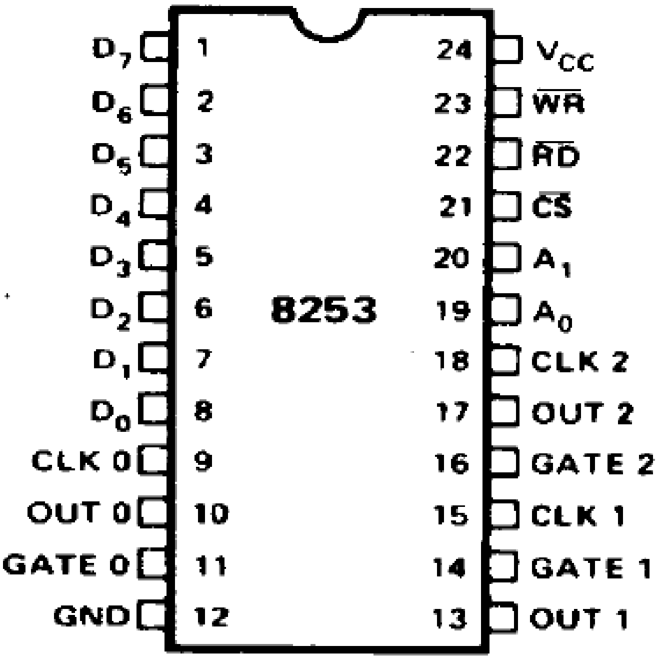
This register is accessed when lines A0 and A1 are at logic 1. It is used to write a command word which specifies the counter to be used (binary or BCD), its mode, and either a read or write operation.

COUNTERS:

These three functional blocks are identical in operation. Each counter consists of a single, 16 bit, pre-settable, down counter.

The counter can operate in either binary or BCD and its input, gate and output are configured by the selection of modes stored in the control word register. The counters are fully independent.

PIN DIAGRAM OF 8253:



D7-D0 (Data bus) :

This tri-state, bi-directional, 8-bit data bus is used to interface the 8253 to the system data bus. The Data bus has three basic functions.

1. Programming the modes of 8253.
2. Loading the count registers.
3. Reading the count values.

CLK:

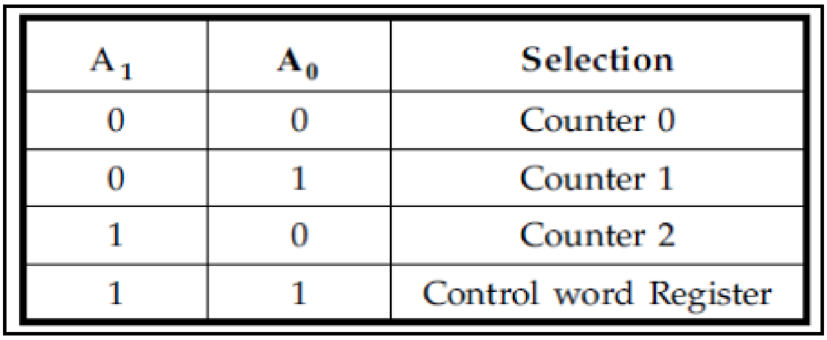
CLK is the input clock frequency, which can range between 0 and 2 MHz for the 8253.

A1 & A0:

These address lines selects counters or CWR and normally connected to the address bus.

CS':

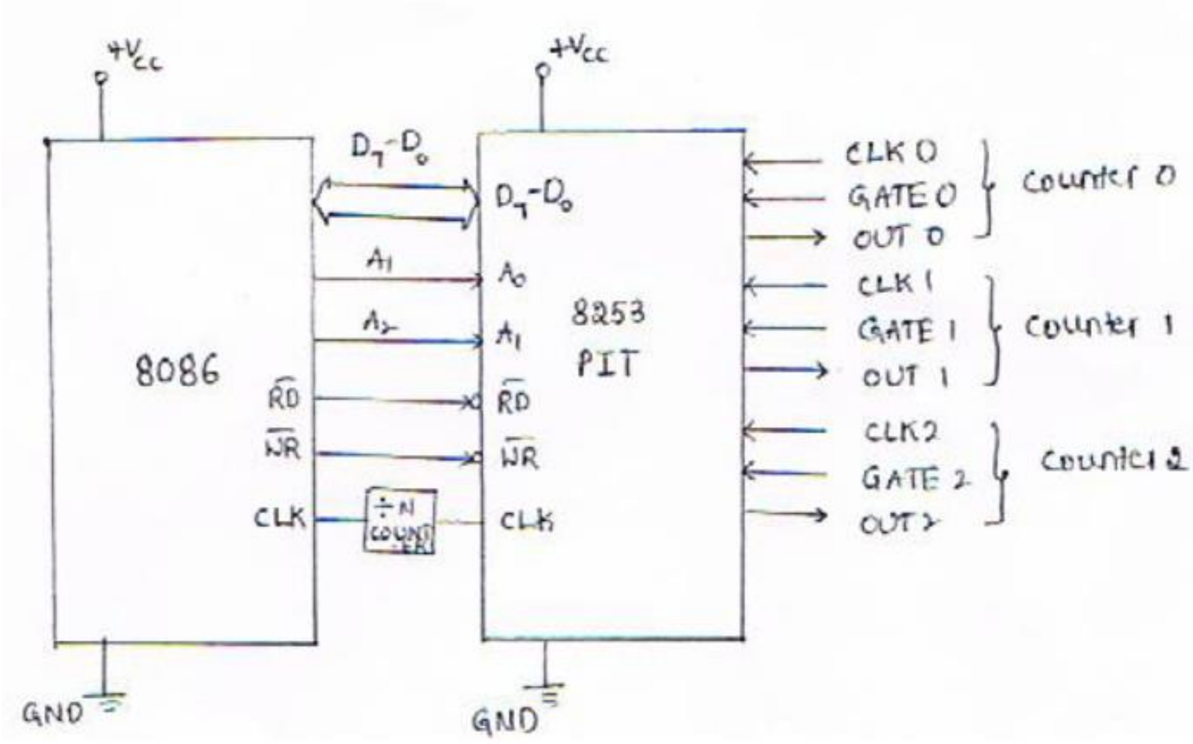
A LOW on this input enables the 8253 for programming, reading and writing counters. No reading or writing of the chip will occur unless the device is selected.



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

RD' causes data to be read from 8253 and **WR'** causes data to be written to 8253.
RD' and WR' are connected to IOR and IOW control signals of the system bus.
OUT: Can have square-wave, one-shot, and other square-shape waves for various duty cycles but no sine-wave or saw-tooth shapes.
GATE: This pin is used to enable or disable the counter.
Vcc: Power supply pin. +5v.
GND: Ground connected to system ground.

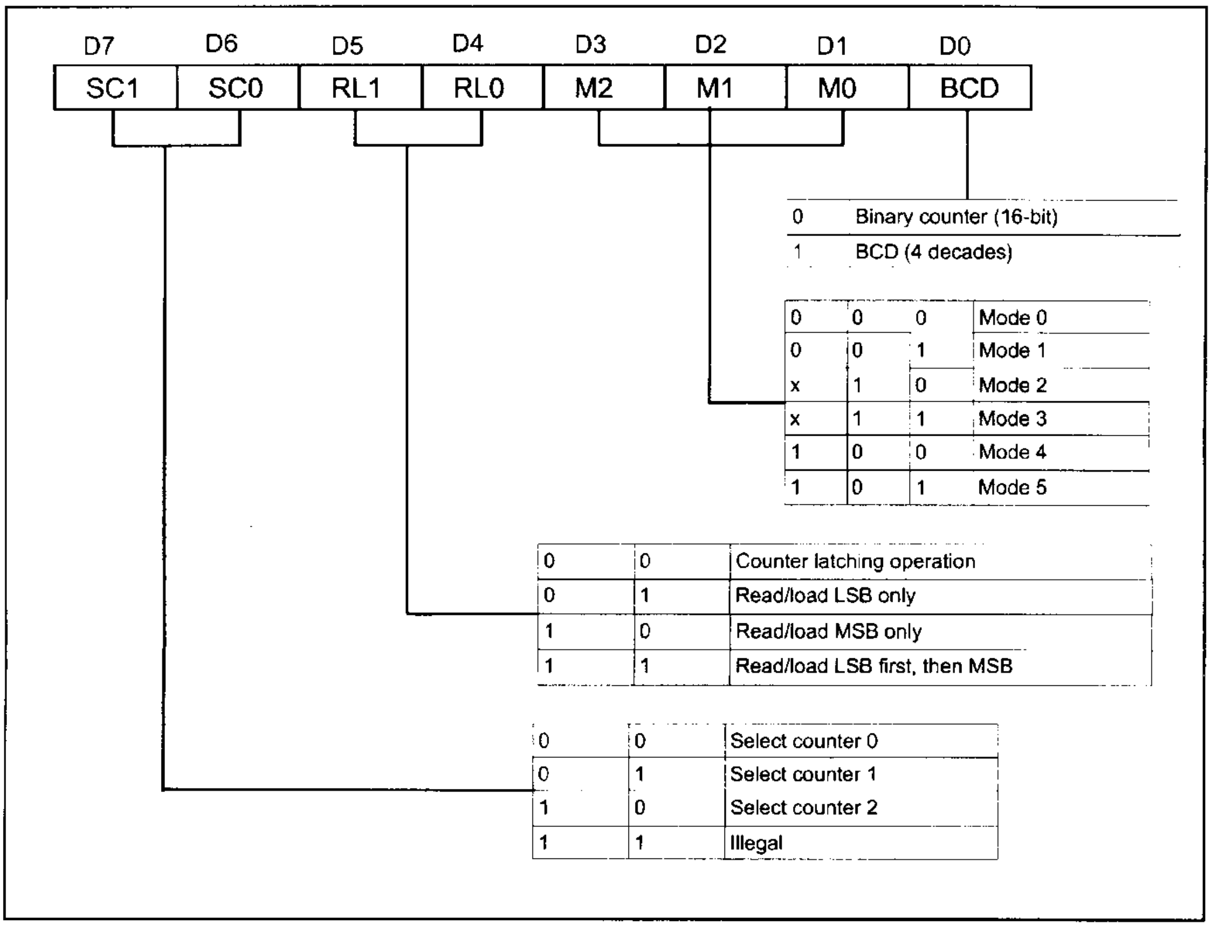
8253 INTERFACING WITH 8086



Initialization of the 8253/54 (Programming of 8253)

- Each of the three counters of the 8253/54 must be programmed separately programmed by writing a control word into the **Control Word Register**.
- The 8253/54 must be initialized before it is used.

The Control Word Register format of 8253 is shown below.



0	Binary counter (16-bit)
1	BCD (4 decades)

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

0	0	Counter latching operation
0	1	Read/load LSB only
1	0	Read/load MSB only
1	1	Read/load LSB first, then MSB

0	0	Select counter 0
0	1	Select counter 1
1	0	Select counter 2
1	1	Illegal

Figure 5-2. 8253/54 Control Word Format

OPERATING MODES OF 8253-PIT

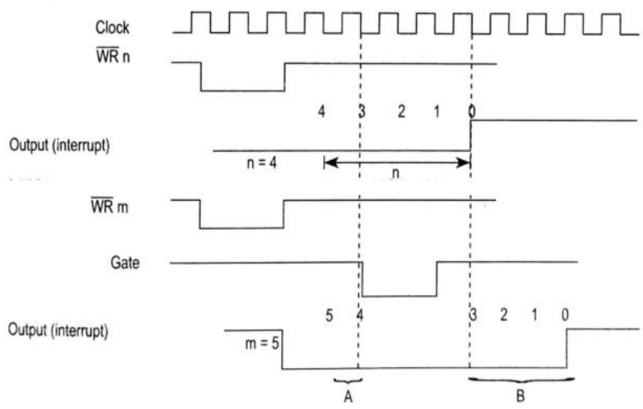
Each of these three counters of 8253 can be operated in one of the following six modes of operation.

1. Mode 0 (Interrupt on terminal count)
2. Mode 1 (Programmable monoshot)
3. Mode 2 (Rate generator)
4. Mode 3 (Square wave generator)
5. Mode 4 (Software triggered strobe)
6. Mode 5 (Hardware triggered strobe)

Mode 0 (Interrupt on terminal count):

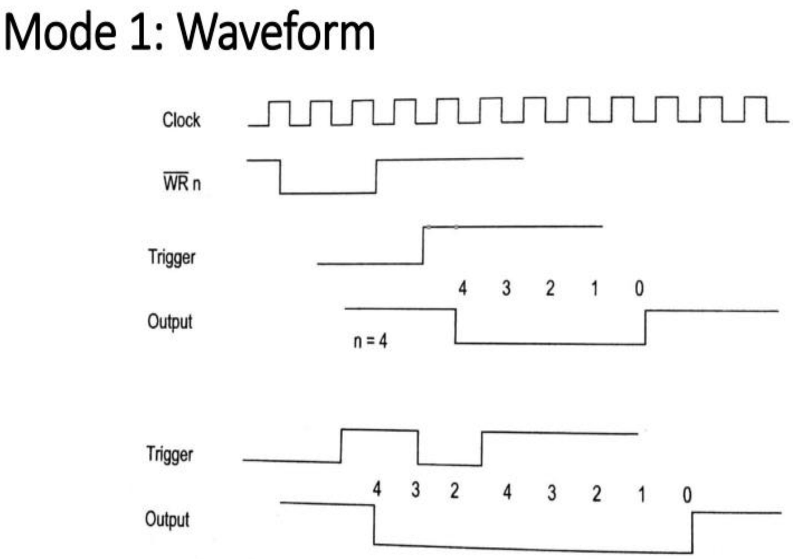
- Initially the OUT is LOW
- Once the Count is Loaded, the Counter is decremented every Cycle.
- When count reaches zero he OUT is high.
- This can be used as Interrupt.

Mode 0: Waveform



Mode 1: programmable one-shot:

- Initially the OUT is HIGH
- When GATE is Triggered, OUT goes LOW and at the end of the count OUT goes HIGH.

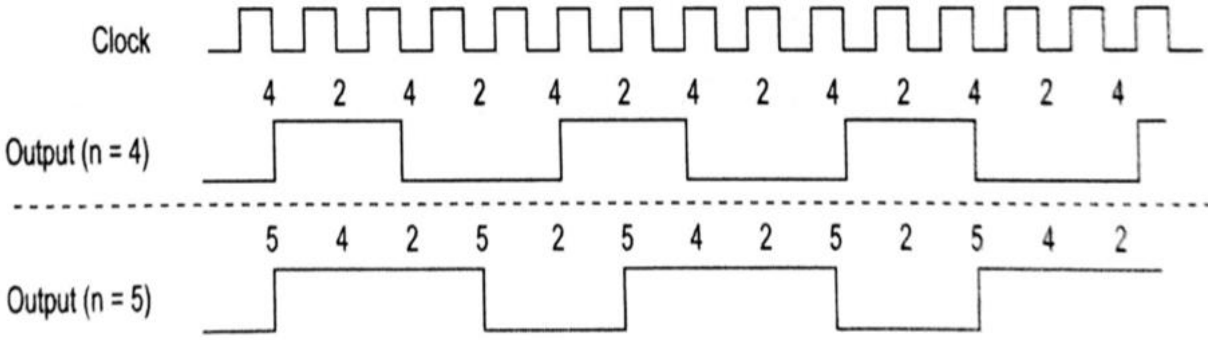


Mode 2: Rate Generator:

- Used to generate a pulse equal to the clock period at a Given Interval.
- OUT is high until the count reaches 1and then OUT goes low for 1 clock pulse.

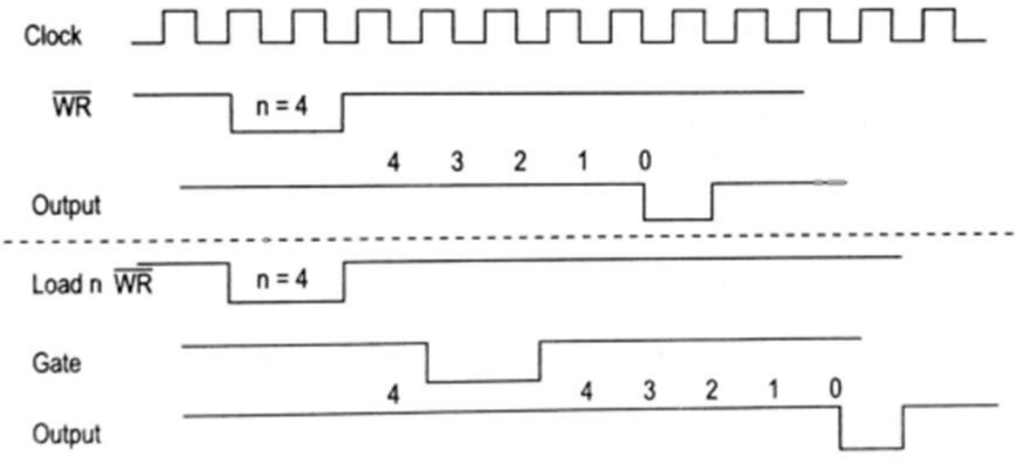
Mode 3: square wave rate generator:

- Initially OUT is LOW, Loading the Count OUT goes HIGH.
- If count n is even number, high= low= n/2.
- If count n is odd number, then high pulse = (n+1)/2,
- low pulse = (n-1)/2



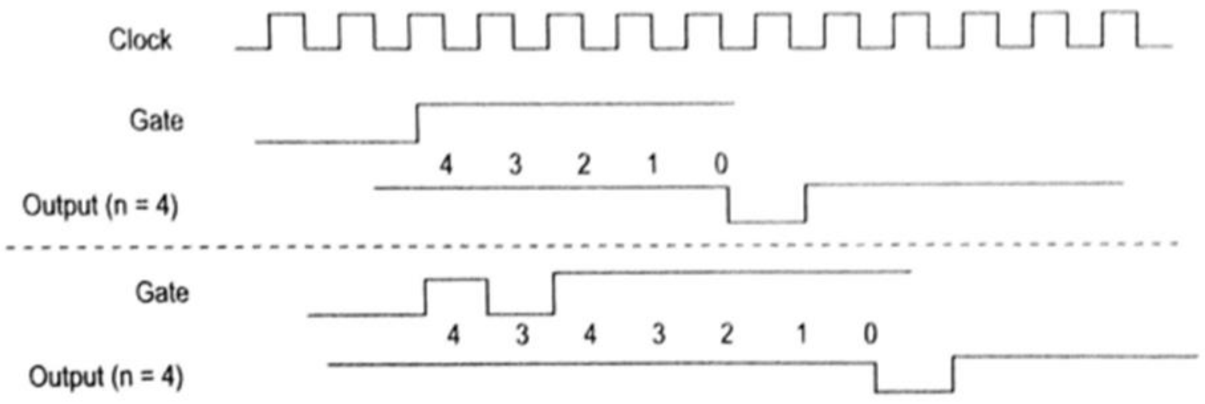
Mode 4: software trigger strobe:

- After Control Word and COUNT is loaded, the output will remain high until the counter reaches zero.
- The counter will then generate a low pulse for 1 clock cycle (a strobe) – after that the output will become high again.



Mode 5: Hardware triggered strobe:

- Similar as mode 4
- Except if trigger by the rising pulse at the Gate.
- Initially out is LOW, GATE triggered from low to high count begins.

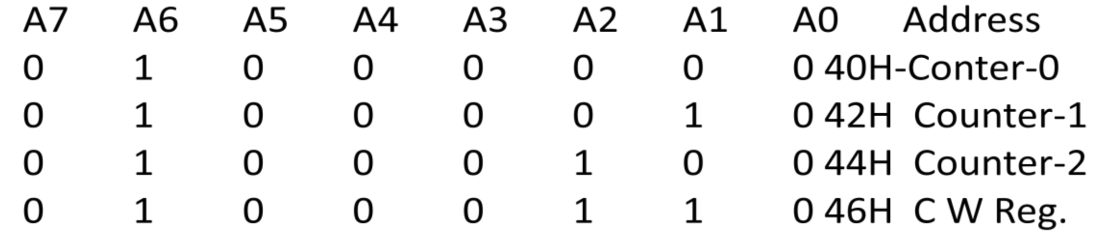


Example: Design a programmable interval timer using 8253 and 8086. Interface 8253 at an address 0040H for counter-0. The 8086 and 8253 run at 6 MHz and 1.5 MHz respectively. Write an ALP to generate a square wave of period 1ms.

Solution:

Design: Neglecting the higher order address lines (A15- A8), the address of the counter-0 is 40H. Address lines A0 and A1 in 8253 are connected to the A1 and A2 of the 8086. These address lines are the selection lines for counters and control word register.

The addresses of the counters are as follows.



A7	A6	A5	A4	A3	A2	A1	A0	Address
0	1	0	0	0	0	0	0	40H-Conter-0
0	1	0	0	0	0	1	0	42H Counter-1
0	1	0	0	0	1	0	0	44H Counter-2
0	1	0	0	0	1	1	0	46H C W Reg.

INTERFACING DIAGRAM:

Control Word

0 0 1 1 0 1 1 1

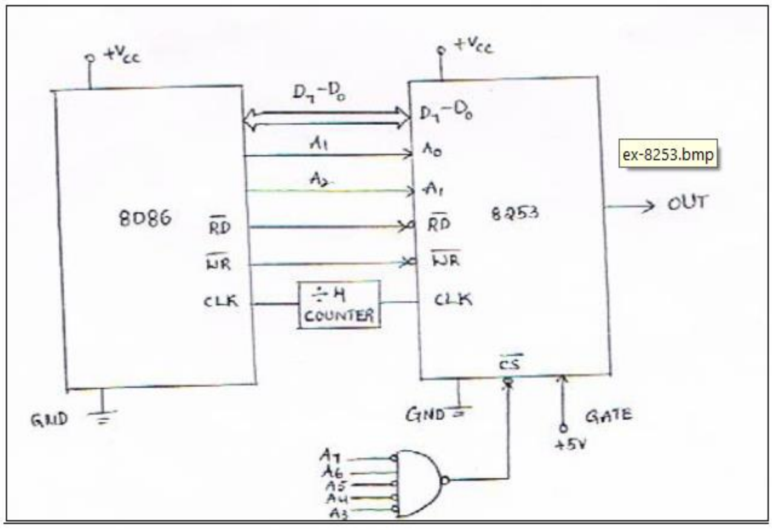
Control word= 37H

The running frequency of 8253 is f= 1.5 MHz

T=1/f=1/1.5×10^-3=0.66 μs

N=Required Time period T

N=1×10^-3* .66×10^−6 = 1500 states



Program

```
ASSUME CS: CODE
CODE SEGMENT
ORG 3000H
MOV AL, 37H
OUT 46H, AL
MOV AL, 00
OUT 40H, AL
 MOV AL, 15
OUT 40H, AL
INT 03H
CODE ENDS
 END
```

INTERRUPTS IN MICROCOMPUTER SYSTEMS

Microcomputer system design requires that, I/O devices such as keyboards, displays, sensors and other components receive service in a an efficient manner so that large amounts of the total system tasks can be assumed by the microcomputer with little or no effect on throughput.

There are two methods for servicing such devices:

1. Polled Method 2. Interrupt Method

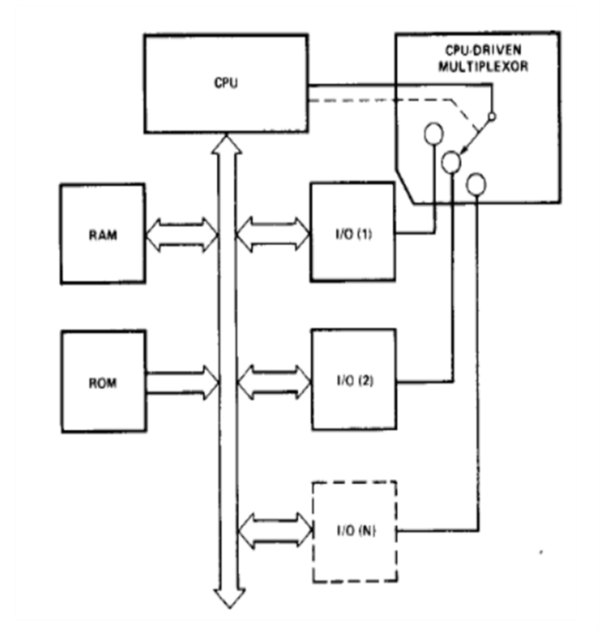


Fig. Polled method

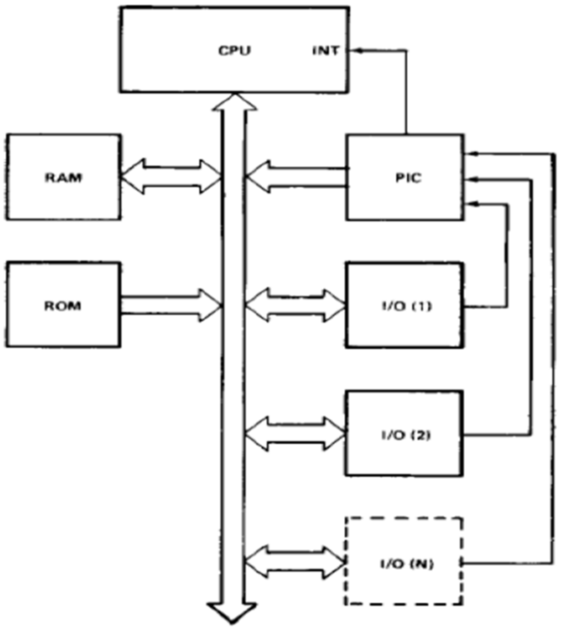


Fig. Interrupt method

NEED OF 8259-PIC

The 8086 has only two interrupt inputs

1. NMI (Non Maskable Interrupt)
2. INTR (Can be masked (ignored) thru instructions CLI and STI)

If more than one Interrupts occurs at a time, we can use an external device called a ***programmable interrupt controller (PIC)*** *to convert the interrupt signals into a single interrupt input on the processor.*

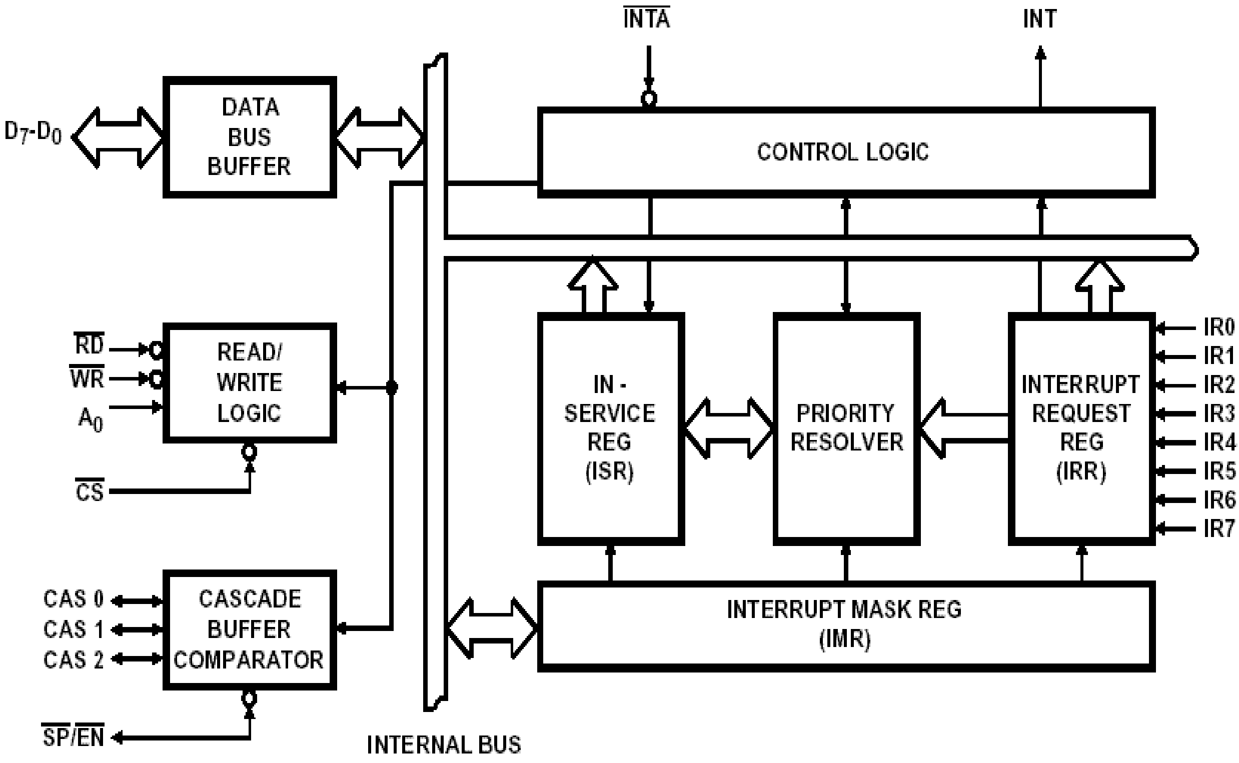
FEATURES OF 8259A:

- It is a tool for managing the interrupt requests.
- Compatible with 8-bit as well as 16-bit microprocessors.
- It provides 8 priority interrupt request levels.
- Be expanded to 64 priority levels by cascading additional 8259A's.
- The interrupts can be masked or unmasked individually.
- Programmable interrupt modes.
- Available in 28-pin DIP.
- It requires a single +5v supply.

ARCHITECTURE OF 8259A-PIC:

- The 8259A is a device specifically designed for use in real time, interrupt driven microcomputer systems.
- It manages eight levels or requests and has built-in features for expandability to other 8259A's (up to 64 levels).

The internal block diagram of 8259A is shown below



DATA BUS BUFFER:

- This bidirectional 8-bit buffer is used to interface the 8259A to the system Data Bus.
- Control words and status information are transferred through the Data Bus Buffer.

READ/WRITE CONTROL LOGIC

- The function of this block is to accept output commands from the microprocessor.
- It contains the Initialization Command Word (ICW) registers and Operation Command Word (OCW) registers which store the various control formats for device operation.
- This function block also allows the status of the 8259A to be transferred onto the Data Bus.

THE CASCADE BUFFER/COMPARATOR

- This function block stores and compares the IDs of all 8259A's used in the system.
- The associated three I/O pins (CAS0,CAS1 & CAS2) are outputs when the 8259A is used as a master and are inputs when the 8259A is used as a slave.
- As a master, the 8259A sends the ID of the interrupting slave device onto the

CAS0,CAS1 & CAS2 lines.

CONTROL LOGIC:

Control logic has two signals.

INT (Interrupt): This output goes directly to the microprocessor interrupt input.

INTA **(Interrupt Acknowledge):** INTA pulses will cause the 8259A to release vectoring information onto the data bus.

INTERRUPT REQUEST REGISTER (IRR) AND IN-SERVICE REGISTER (ISR):

- The IRR is used to store all the interrupt levels which are requesting service.
- Normally IR0 has highest priority and IR7 has the lowest priority
- The ISR is used to store all the interrupt levels which are being serviced.

PRIORITY RESOLVER

- This logic block determines the priorities of the bits set in the IRR.
- The highest priority is selected and strobed into the corresponding bit of the ISR during INTA pulse.

INTERRUPT MASK REGISTER (IMR):

- The IMR stores the bits which mask the interrupt lines to be masked.
- The IMR operates on the IRR

INTERRUPT SEQUENCE IN 8086 SYSTEM:

- The events occur as follows in an 8086-8259A system:

1. One or more of the INTERRUPT REQUEST lines (IR7 – IR0) 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 � � � � pulse.

4. Upon receiving an � � � � from the CPU, 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 - D0 pins.

5. This CALL instruction will initiate second INTA pulses to be sent to the 8259A from the CPU group.

6. This � � � � pulse allows the 8259A to release an 8-bit preprogrammed subroutine address onto the Data Bus.

7. ISR bit is reset at the end of the 2nd � � � � pulse if automatic EOI mode is programmed.

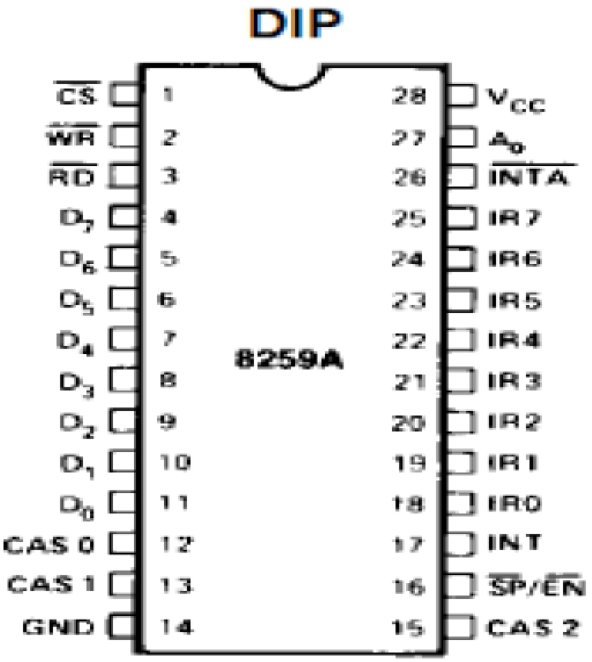
PIN DIAGRAM OF 8259A:

Vcc: +5V Supply.

GND: Ground.

CS' : A LOW on this input enables the 8259A. No reading or writing of the chip will occur unless the device is selected.

WR': A LOW on this input enables the microprocessor



to write control words (ICWs and OCWs) to the 8259A.

RD’: A LOW on this input enables the 8259A to send the status of the Interrupt Request Register (IRR), In Service Register (ISR), the Interrupt Mask Register (IMR), or the Interrupt level onto the Data Bus.

D7-D0: Control, status and interrupt-vector information is transferred via this bus.

A0: This input signal is used in conjunction with WR and RD signals to write commands into the various command registers, as well as reading the various status registers of the chip. This line can be tied directly to one of the address lines.

CAS0-CAS2: 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.

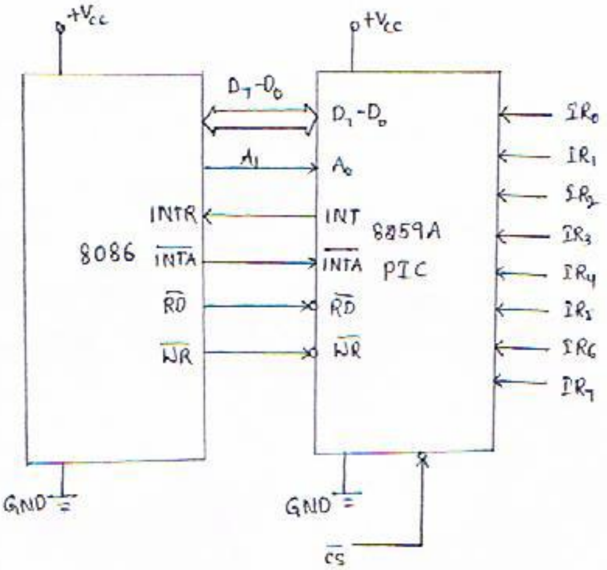
IR7-IR0: Asynchronous inputs. An interrupt request is executed by raising an IR input (low to high), and holding it high until it is acknowledged (Edge Triggered Mode), or just by a high level on an IR input (Level Triggered Mode).

SP/EN: 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: This pin goes high whenever a valid interrupt request is asserted. It is used to interrupt the microprocessor, thus it is connected to the microprocessor's interrupt pin.

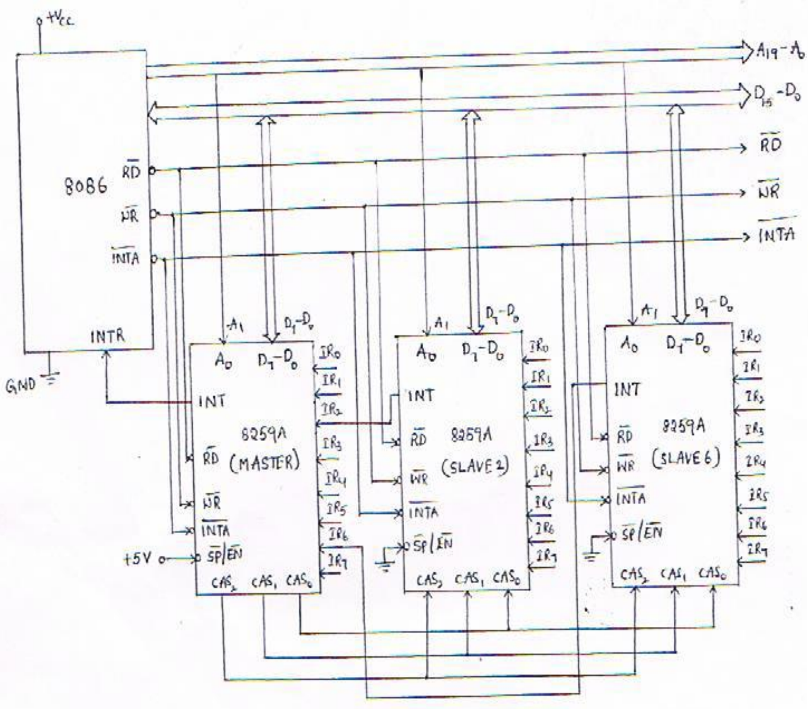
⍰ ⍰ ⍰ ⍰: This pin is used to enable 8259A interrupt-vector data onto the data bus by a sequence of interrupt acknowledge pulses issued by the microprocessor.

8259A PIC INTERFACING WITH 8086:



CASCADED INTERFACING OF 8259A

Example: Interface 3 IC’s of 8259A PIC with 8086 in such way that one is master and rest two are slaves connected at IR2 and IR6 interrupt request level of the master.



OPERATING MODES OF 8259A PIC:

The different modes of operation of 8259A can be programmed by setting or resting the appropriate bits of the ICW or OCW.

The different modes of operation of 8259A are explained in the following.

FULLY NESTED MODE: This is the default mode of operation of 8259A. IR0 has the highest priority and IR7 has the lowest one. When interrupt request are noticed, the highest priority request amongst them is determined and the vector is placed on the data bus. The corresponding bit of ISR is set and remains set till the microprocessor issues an EOI command just before returning from the service routine or the AEOI bit is set.

END OF INTERRUPT (EOI):

The ISR bit can be reset either with AEOI bit of ICW1 or by EOI command, issued before returning from the interrupt service routine. There are two types of EOI commands specific and non-specific.

AUTOMATIC ROTATION:

This is used in the applications where all the interrupting devices are of equal priority. In this mode, an interrupt request IR level receives priority after it is served while the next device to be served gets the highest priority in sequence. Once all the device are served like this, the first device again receives highest priority.

AUTOMATIC EOI MODE:

Till AEOI=1 in ICW4, the 8259A operates in AEOI mode. In this mode, the 8259A performs a non-specific EOI operation at the trailing edge of the last INTA pulse automatically. This mode should be used only when a nested multilevel interrupt structure is not required with a single 8259A.

SPECIFIC ROTATION:

In this mode a bottom priority level can be selected, using L2, L1 and L0 in OCW2 and R=1, SL=1, EOI=0. The selected bottom priority fixes other priorities. If IR5 is selected as a bottom priority, then IR5 will have least priority and IR4 will have a next higher priority. Thus IR6 will have the highest priority.

SPECIFIC MASK MODE:

In specific mask mode, when a mask bit is set in OCW1, it inhibits further interrupts at that level and enables interrupt from other levels, which are not masked.

BUFFERED MODE:

When the 83259A is used in the systems where bus driving buffers are used on data buses. The problem of enabling the buffers exists. The 8259A sends buffer enable signal on SP / EN pin, whenever data is placed on the bus.

CASCADE MODE:

The 8259A can be connected in a system containing one master and eight slaves (maximum) to handle upto 64 priority levels. The master controls the slaves using CAS0-CAS2 which act as chip select inputs (encoded) for slaves.

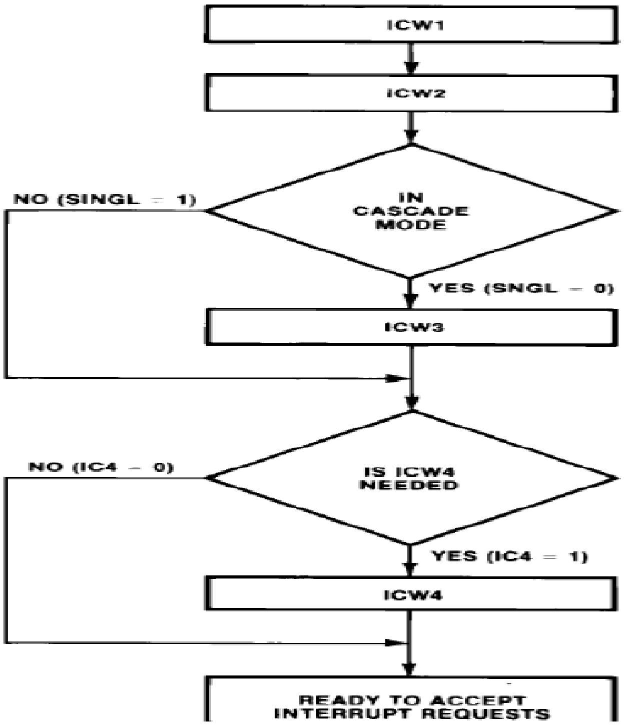
PROGRAMMING THE 8259A PIC:

The 8259A accepts two types of command words generated by the microprocessor.

1. Initialization Command Words (ICWs)
2. Operation Command Words (OCWs)

ICWs: Before it starts functioning, the 8259A must be initialized by writing two to four command words into the respective command word registers. These are called as initialized command words.

The initialization sequence of 8259A is described in form of a flow chart in fig below.

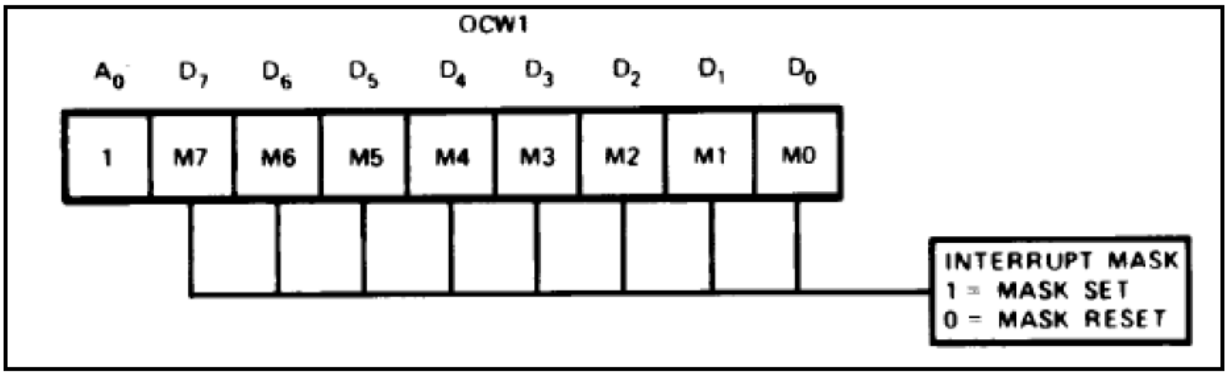


OPERATION COMMAND WORDS (OCWS):

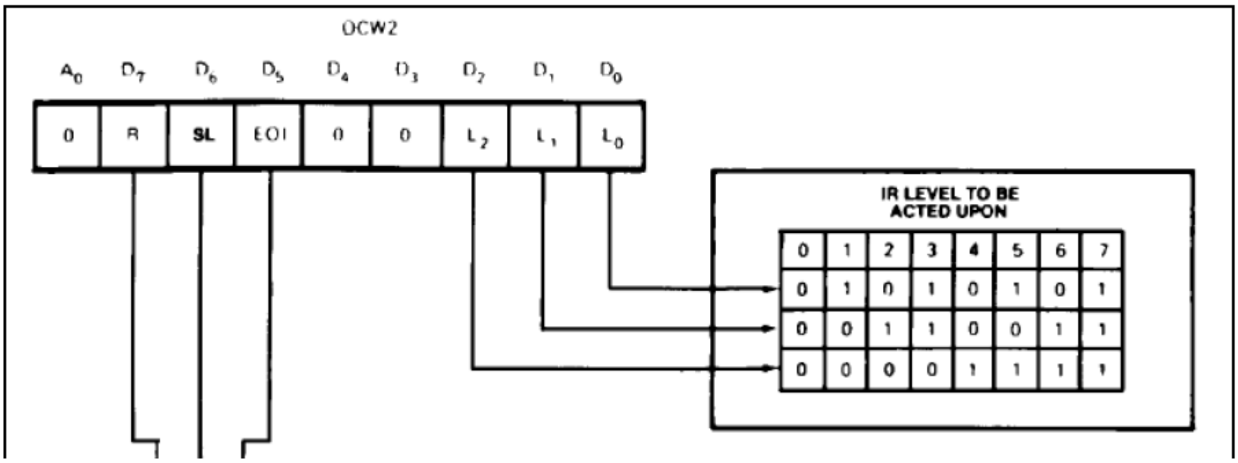
Once 8259A is initialized using ICW'S then modes of operations can be selected by programming, i.e. writing three

Internal registers called as operation command words.

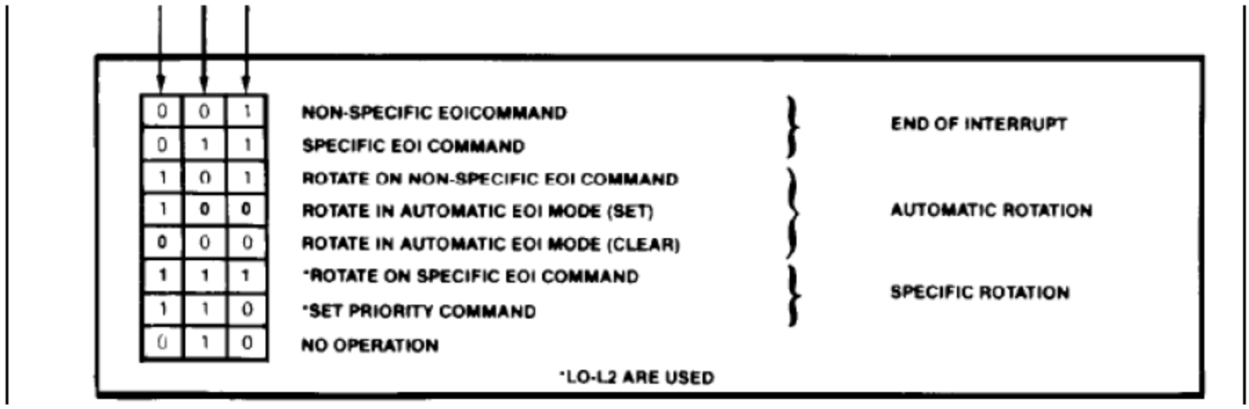
The three operation command words are shown in fig with the bit selection details. OCW1 is used to mask the masked and if it is 0 the request is enabled.



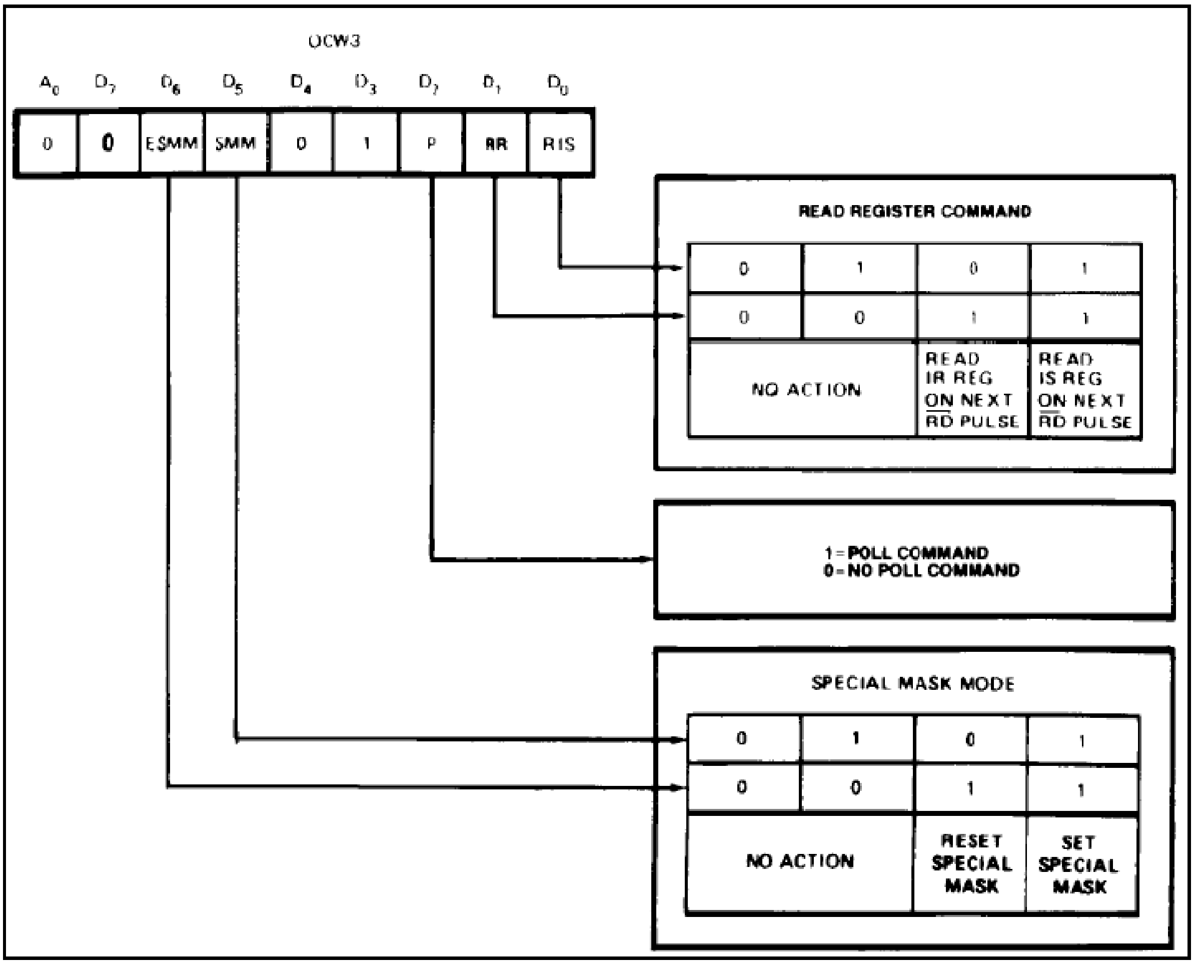
OPERATION COMMAND WORD 2 (OCW 2):



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


0 0 1 NON-SPECIFIC EOICOMMAND
0 1 1 SPECIFIC EOI COMMAND
END OF INTERRUPT
1 0 1 ROTATE ON NON-SPECIFIC EOI COMMAND
1 0 0 ROTATE IN AUTOMATIC EOI MODE (SET)
0 0 0 ROTATE IN AUTOMATIC EOI MODE (CLEAR)
AUTOMATIC ROTATION
1 1 1 *ROTATE ON SPECIFIC EOI COMMAND
1 1 0 *SET PRIORITY COMMAND
SPECIFIC ROTATION
0 1 0 NO OPERATION
*LO-L2 ARE USED

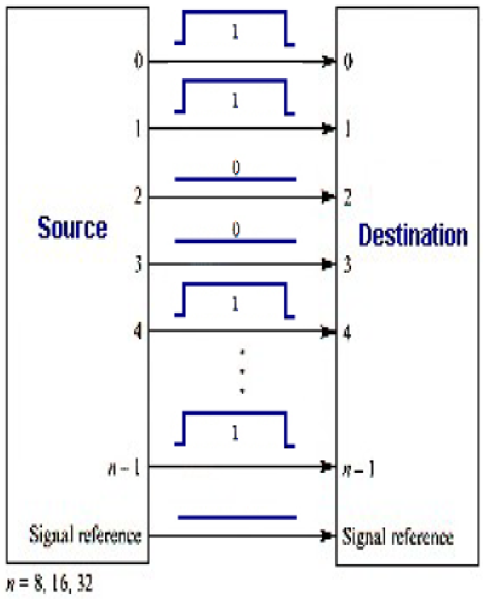
OPERATION COMMAND WORD 3 (OCW 3):



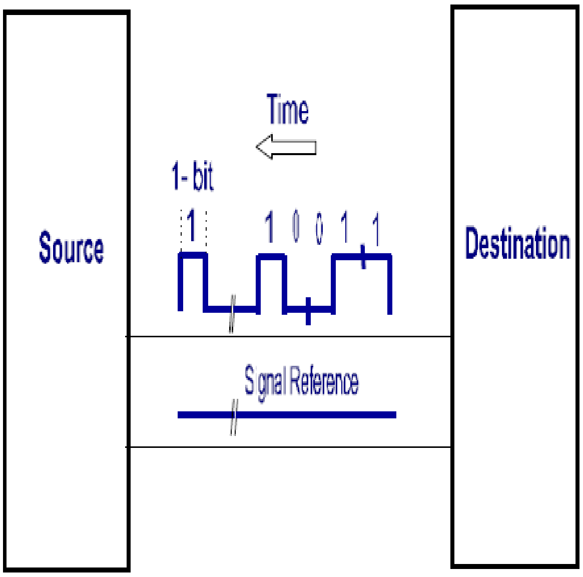
SERIAL COMMUNICATION: Serial communication transmits data one bit at a time, sequentially, over a single communication line to a receiver.

This method is used when data transfer rates are very low or the data must be transferred over long distances.

SERIAL AND PARALLEL Data TRANSMISSION:



Parallel Transmission



Serial Transmission

Simplex: In Simplex connection, data is transmitted in one direction only. For example, T.V Broad Casting.

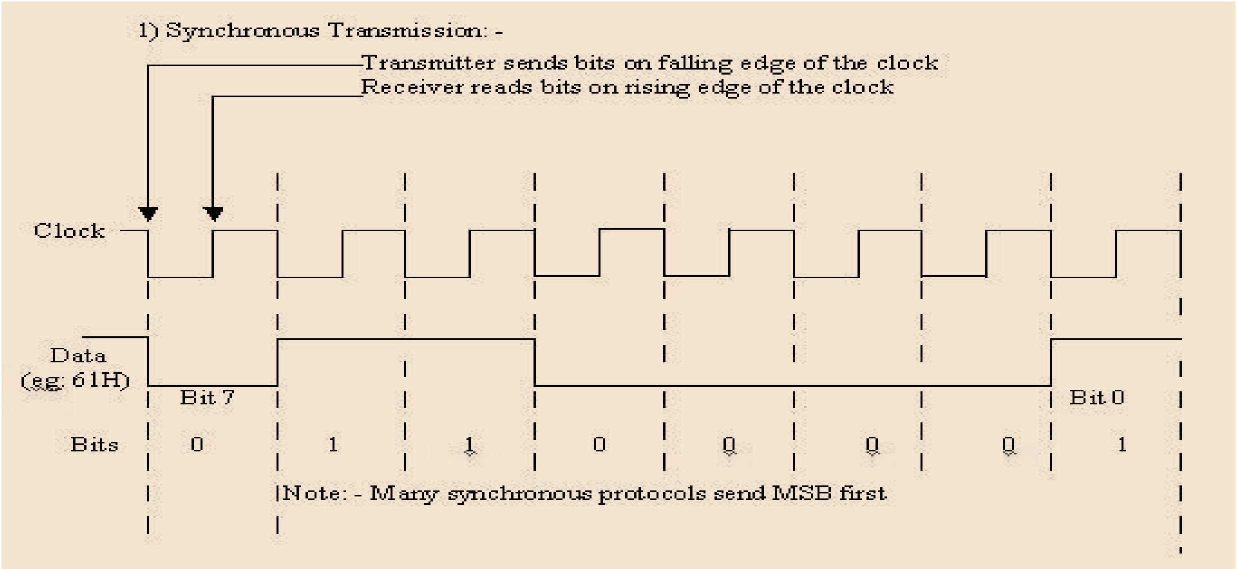
Half-duplex: In a half-duplex connection, two-way transfer of data is possible, but only in one direction at a time. For example, POLICE RADAR.

Full duplex: In a full-duplex configuration, both ends can send and receive data simultaneously, which technique is common in our PCs. For example, Telephone systems.

SERIAL DATA TRANSFER MODES: There are two modes. They are 1. Synchronous data transmission 2. Asynchronous data transmission.

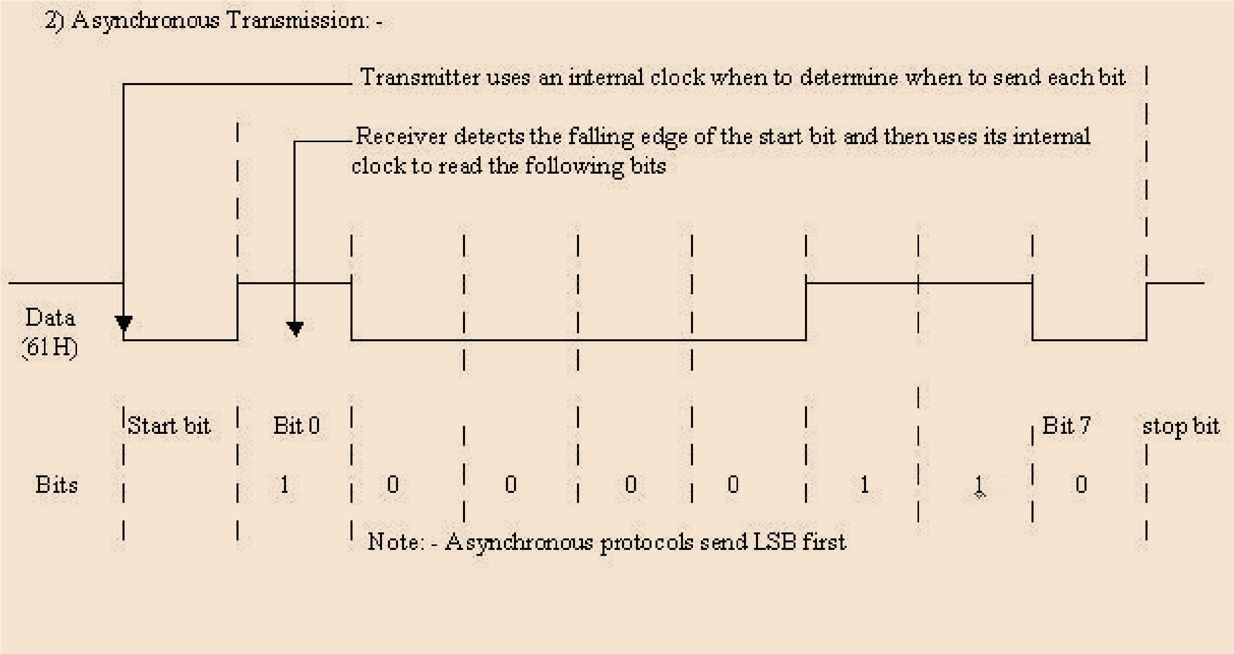
SYNCHRONOUS DATA TRANSMISSION MODE:

In synchronous Mode, the stream of data to be transferred is encoded and sent on one line, and a clock is put on another line, that tells the receiver about the beginning and the ending of each bit.

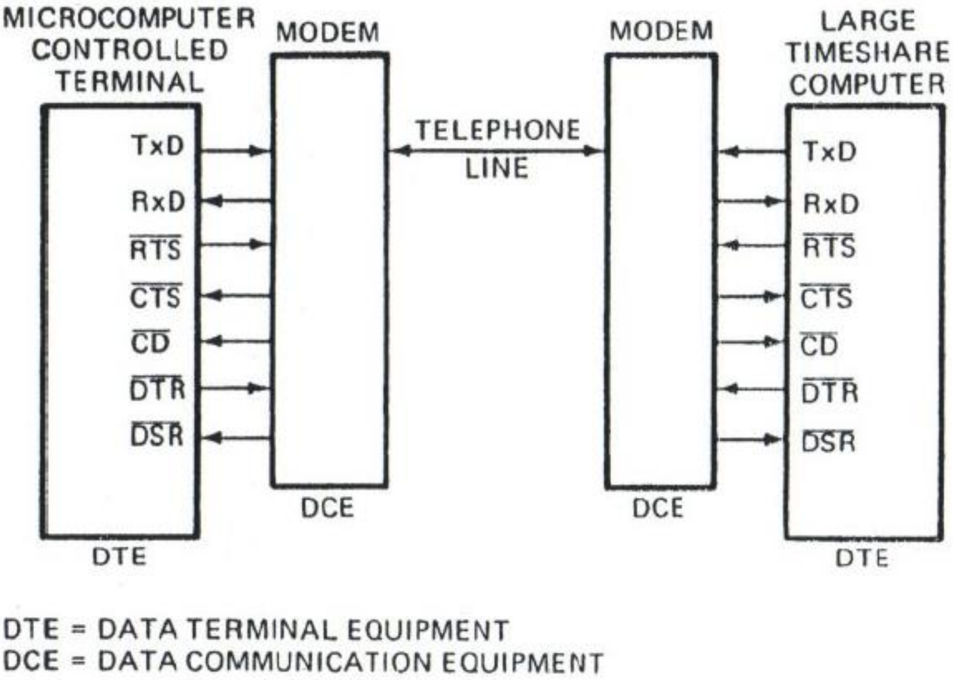

1) Synchronous Transmission: -
Transmitter sends bits on falling edge of the clock
Receiver reads bits on rising edge of the clock
Clock
Data
(eg: 61H)
Bit 7
Bit 0
Bits
0
1
1
0
0
0
0
1
Note: - Many synchronous protocols send MSB first

ASYNCHRONOUS DATA TRANSMISSION MODE:

In Asynchronous Mode, the stream of data to be transferred is encoded and sent on one line, the beginning and the ending of each Character is identified by Start(Always '0') and Stop Bits (Always '1').


2) Asynchronous Transmission: -
Transmitter uses an internal clock when to determine when to send each bit
Receiver detects the falling edge of the start bit and then uses its internal clock to read the following bits
Data
(61H)
Start bit
Bit 0
Bit 7
stop bit
Bits
1
0
0
0
0
1
1
0
Note: - Asynchronous protocols send LSB first

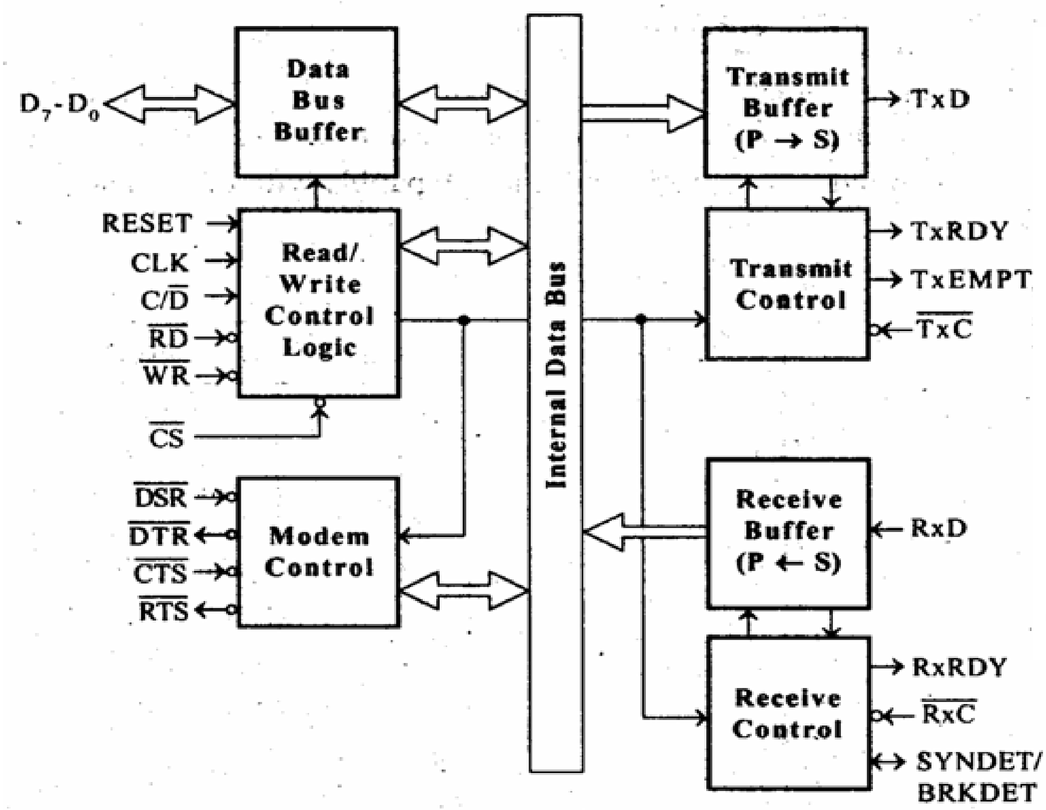
DIGITAL DATA TRANSMISSION USING MODEMS AND TELEPHONE LINES:

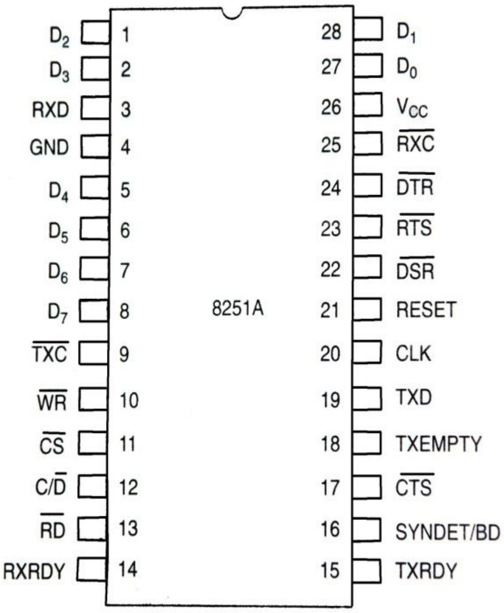


8251A programmable Communication Interface:

8251A is a USART (Universal Synchronous Asynchronous Receiver Transmitter) for serial data communication. Programmable peripheral designed for synchronous /asynchronous serial data communication, packaged in a 28-pin DIP. Receives parallel data from the CPU & transmits serial data after conversion. Also receives serial data from the outside & transmits parallel data to the CPU after conversion.

Block diagram of the 8251 USART and Pin diagram:





Sections of 8251A:

- Data Bus buffer
- Read/Write Control Logic
- Modem Control
- Transmitter Receiver

Data Bus Buffer: D0-D7 : 8-bit data bus used to read or write status, command word or data from or to the 8251A.

Read/Write Control logic: Includes a control logic, six input signals & three buffer registers: Data register, control register & status register.

Control logic : Interfaces the chip with MPU, determines the functions of the chip according to the control word in the control register & monitors the data flow.

Input signals:

- CS – Chip Select : When signal goes low, the 8251A is selected by the MPU for communication.
- C/D – Control/Data : When signal is high, the control or status register is addressed; when it is low, data buffer is addressed. (Control register & status register are differentiated by WR and RD signals)
- WR : When signal is low, the MPU either writes in the control register or sends output to the data buffer.
- RD : When signal goes low, the MPU either reads a status from the status register or accepts data from data buffer.
- RESET : A high on this signal reset 8252A & forces it into the idle mode.
- CLK : Clock input, usually connected to the system clock for communication with

the microprocessor.

Control Register:

- **16-bit register** for a control word consist of two independent bytes namely **mode word & command word.**
- **Mode word :** Specifies the general characteristics of operation such as baud, parity, number of bits etc.
- **Command word :** Enables the data transmission and reception.
- Register can be accessed as an output port when the **Control/Data pin is high**.

Status register:

- Checks the ready status of the peripheral.
- Status word in the status register provides the information concerning register status and transmission errors.

Data register: Used as an input and output port when the C/D is low

CS	C/D	WR	RD	Operation
0	0	1	0	MPU reads data from data buffer
0	0	0	1	MPU writes data from data buffer
0	1	0	1	MPU writes a word to control register
0	1	1	0	MPU reads a word from status register
	×	×	×	Chip is not selected for any operation

Modem Control:

- DSR' - Data Set Ready : Checks if the Data Set is ready when communicating with a modem.
- DTR' - Data Terminal Ready : Indicates that the device is ready to accept data when the 8251 is communicating with a modem.
- CTS' - Clear to Send : If its low, the 8251A is enabled to transmit the serial data provided the enable bit in the command byte is set to '1'.
- RTS' - Request to Send Data : Low signal indicates the modem that the receiver is ready to receive a data byte from the modem.

Transmitter section: Accepts parallel data from MPU & converts them into serial data.

- Has two registers:

Buffer register : To hold eight bits

Output register : To convert eight bits into a stream of serial bits.

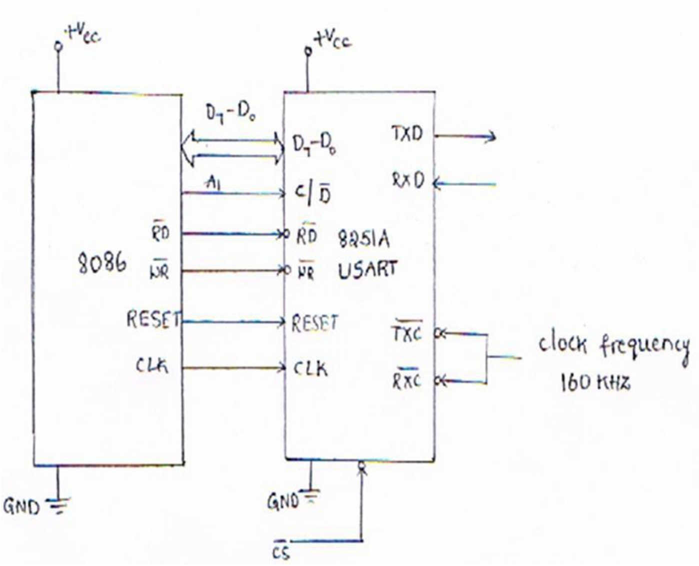
- The MPU writes a byte in the buffer register.
- Whenever the output register is empty; the contents of buffer register are transferred to output register.
- Transmitter section consists of three output & one input signals
  - TxD - Transmitted Data Output : Output signal to transmit the data to peripherals
  - TxC ' - Transmitter Clock Input : Input signal, controls the rate of transmission.
  - TxRDY - Transmitter Ready : Output signal, indicates the buffer register is empty and the USART is ready to accept the next data byte
  - TxE - Transmitter Empty : Output signal to indicate the output register is

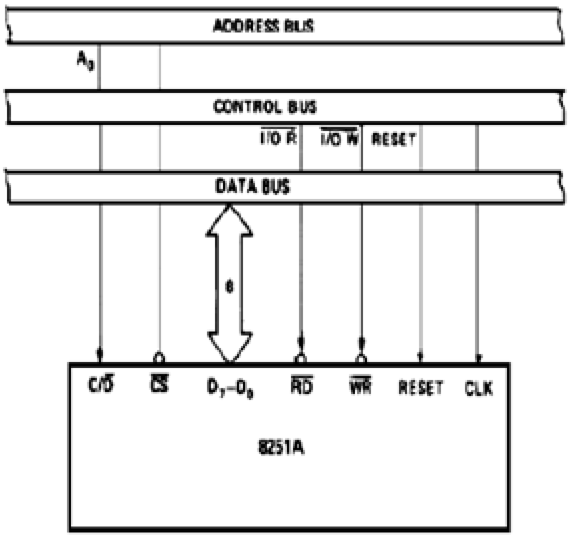
empty and the USART is ready to accept the next data byte.

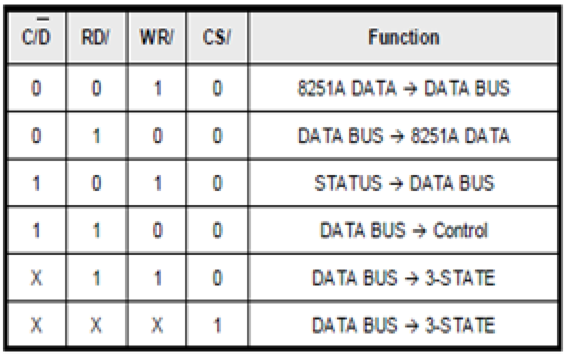
Receiver Section: Accepts serial data on the RxD pin and converts them to parallel data.

- Has two registers :
  - Receiver input register
  - Buffer register
- When RxD goes low, the control logic assumes it is a start bit, waits for half bit time, and samples the line again. If the line is still low, the input register accepts the following data, and loads it into buffer register at the rate determined by the receiver clock.
- RxRDY - Receiver Ready Output: Output signal, goes high when the USART has a character in the buffer register & is ready to transfer it to the MPU.
- RxD - Receive Data Input : Bits are received serially on this line & converted into a parallel byte in the receiver input register.
- RxC’ - Receiver Clock Input : Clock signal that controls the rate at which bits are received by the USART.

8251A USART INTERFACING WITH 8086:







C/D	RD/	WR/	CS/	Function
0	0	1	0	8251A DATA → DATA BUS
0	1	0	0	DATA BUS → 8251A DATA
1	0	1	0	STATUS → DATA BUS
1	1	0	0	DATA BUS → Control
X	1	1	0	DATA BUS → 3-STATE
X	X	X	1	DATA BUS → 3-STATE

PROGRAMMING THE 8251A: Prior to starting a data transmission or reception, the 8251A must be loaded with a set of control words generated by the microprocessor.

The control words are split into two formats.

1. Mode instruction
2. Command instruction

MODE INSTRUCTION:

Mode instruction is used for setting the function of the 8251A

Items set by mode instruction are as follows:

- Synchronous/asynchronous mode
- Stop bit length (asynchronous mode)
- Character length
- Parity bit
- Baud rate factor (asynchronous mode)
- Internal/external synchronization (synchronous mode)
- Number of synchronous characters (Synchronous mode)

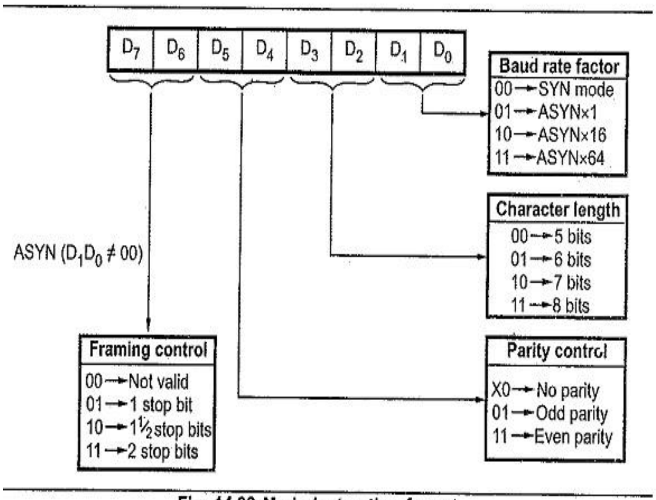


Fig. 14.38 Mode instruction format

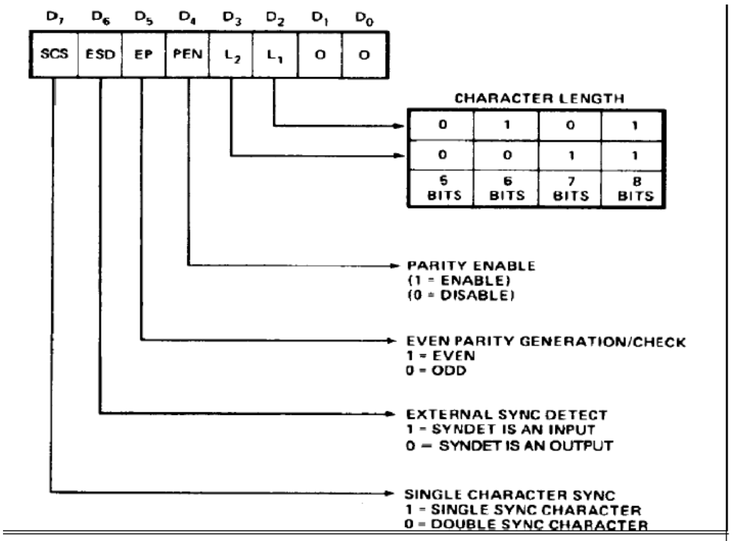


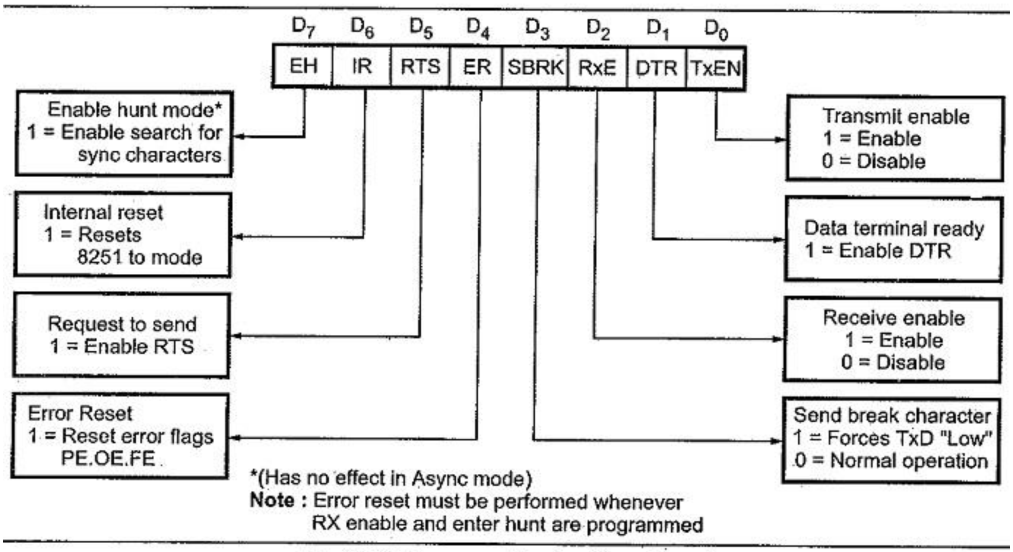
Fig. Mode instruction format, Synchronous mode

COMMAND INSTRUCTION:

Command is used for setting the operation of the 8251.

Items to be set by command are as follows:

- Transmit Enable/Disable
- Receive Enable/Disable
- DTR, RTS Output of data.
- Resetting of error flag.
- Sending to break characters
- Internal resetting
- Hunt mode (synchronous mode)



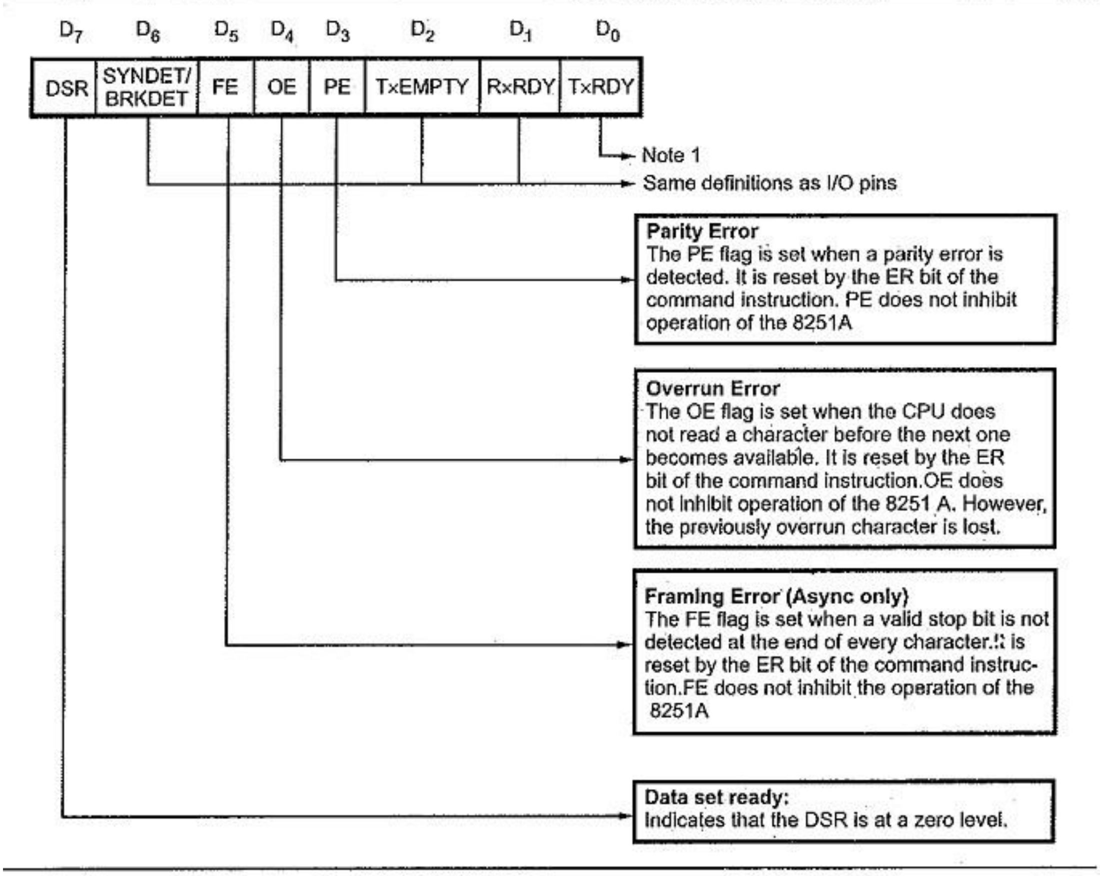
*(Has no effect in Async mode)

Note : Error reset must be performed whenever RX enable and enter hunt are programmed

Fig. 14.39 Command instruction format

STATUS WORD:

It is possible to see the internal status of the 8251 by reading a status word. The format of status word is shown below.



Parity Error
The PE flag is set when a parity error is detected. It is reset by the ER bit of the command instruction. PE does not inhibit operation of the 8251A

Overrun Error
The OE flag is set when the CPU does not read a character before the next one becomes available. It is reset by the ER bit of the command instruction.OE does not inhibit operation of the 8251 A. However, the previously overrun character is lost.

Framing Error (Async only)
The FE flag is set when a valid stop bit is not detected at the end of every character.It is reset by the ER bit of the command instruction.FE does not inhibit the operation of the 8251A

Data set ready:
Indicates that the DSR is at a zero level.

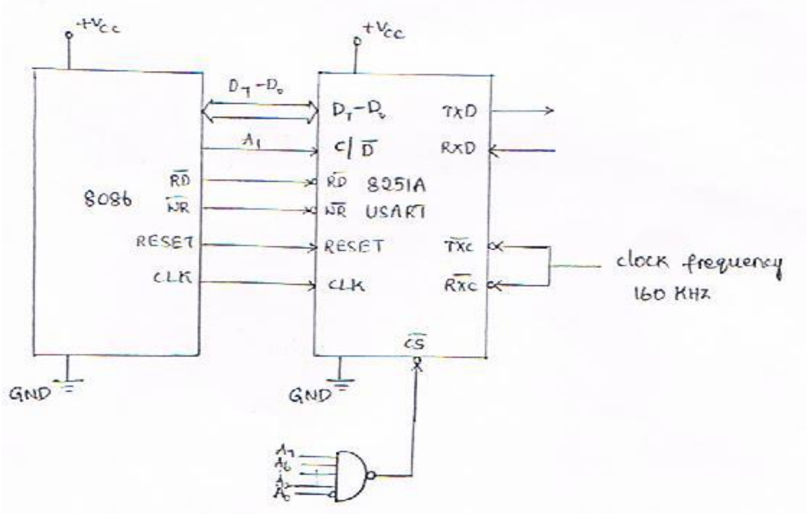
Fig. 14.40 Status register format

Programming Example:

Design the hardware interface circuit for interfacing 8251 with 8086. Set the 8251A in asynchronous mode as a transmitter and receiver with even parity enabled, 2 stop bits, 8-bit character length, frequency 160 KHz and baud rate 10K.

a). Write an ALP to transmit 100 bytes of data string starting at location 2000:5000H.

b). Write an ALP to receive 100 bytes of data string and store it at 3000:4000H



PORT ADDRESSES

A7	A6	A5	A4	A3	A2	A1=C/$\overline{D}$	A0	Address
1	1	1	1	1	1	1	0	FEH→Control word register
1	1	1	1	1	1	0	0	FCH→ Data

CONTROL WORD

Mode control word

1	1	1	1	1	1	1	0

Mode control word=FEH

Command word

0	0	0	1	0	0	0	1

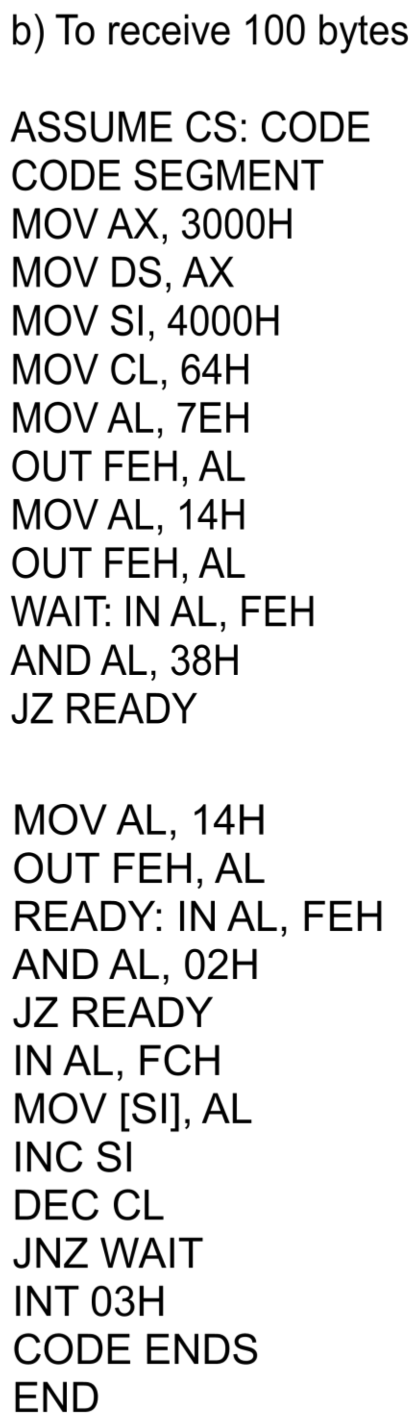
Command word= 11H

Program:

a). To transmit 100 bytes

```
ASSUME CS: CODE
CODE SEGMENT
ORG 3000H
MOV AX, 2000H
MOV DS, AX
MOV SI, 5000H
MOV CL, 64H
MOV AL, FEH
OUT FEH, AL
MOV AL, 11H
OUT FEH, AL
WAIT: IN AL, FEH
AND AL, 01H
JZ WAIT

MOV AL, [SI]
OUT FCH, AL
INC SI
DEC CL
JNZ WAIT
INT 03H
CODE ENDS
END
```



b) To receive 100 bytes of data

```
ASSUME CS: CODE
CODE SEGMENT
MOV AX, 3000H
MOV DS, AX
MOV SI, 4000H
MOV CL, 64H
MOV AL, 7EH
OUT FEH, AL
MOV AL, 14H
OUT FEH, AL
WAIT: IN AL, FEH
AND AL, 38H
JZ READY

MOV AL, 14H
OUT FEH, AL
READY: IN AL, FEH
AND AL, 02H
JZ READY
IN AL, FCH
MOV [SI], AL
INC SI
DEC CL
JNZ WAIT
INT 03H
CODE ENDS
END
```

DMA CONTROLLER 8257:

It is a 4-channel DMA.

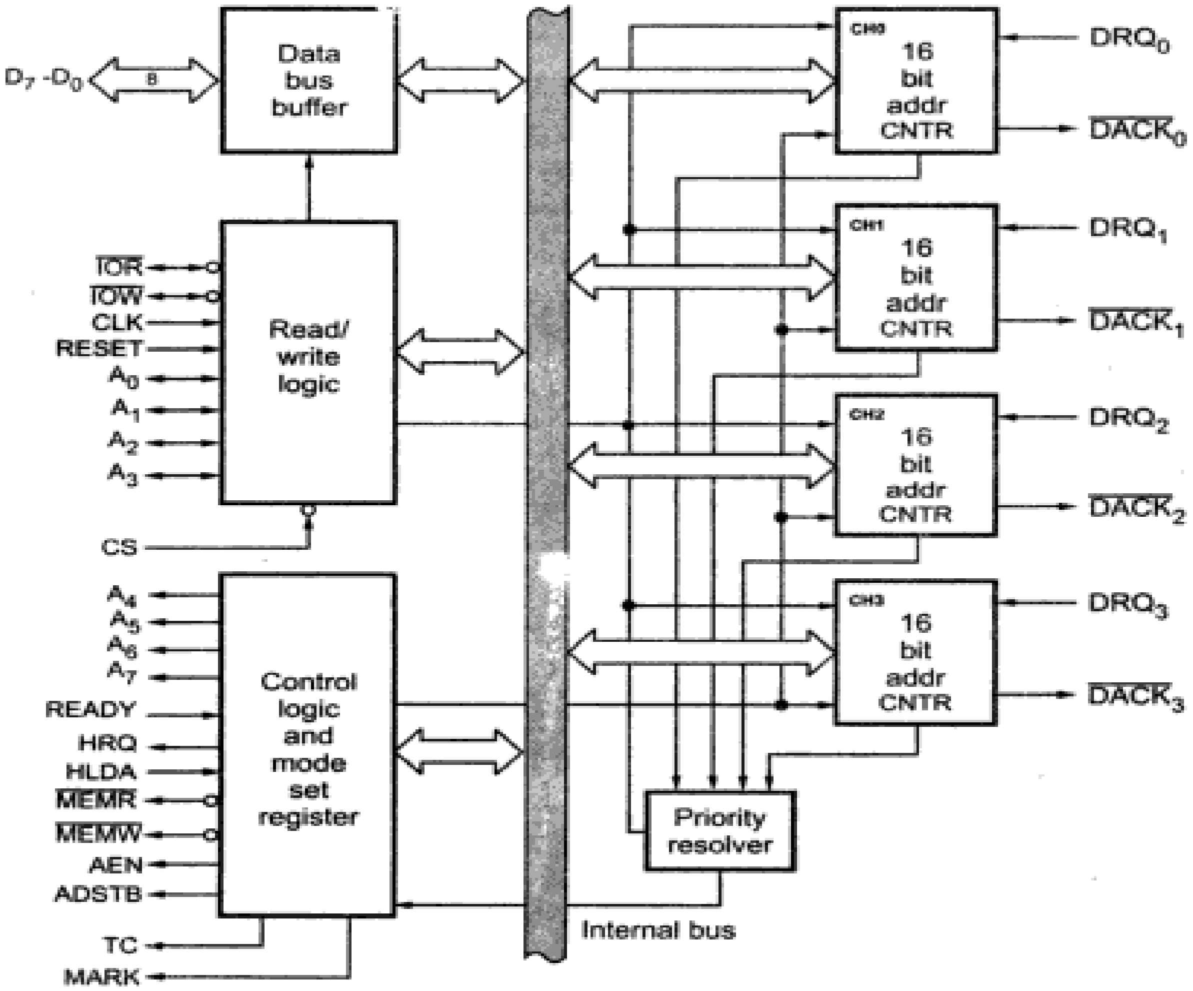
- So 4 I/O devices can be interfaced to DMA
- It is designed by Intel
- Each channel have 16-bit address and 14 bit counter
- It provides chip priority resolver that resolves priority of channels in fixed or rotating mode.
- It provide on chip channel inhibit logic.

It generates a TC signal to indicate the peripheral that the programmed number of data bytes have been transferred. It generates MARK signal to indicate the peripheral that 128 bytes have been transferred. It requires single phase clock.
The maximum frequency is 3Mhz and minimum frequency is 250 Hz.

- It execute 3 DMA cycles
  1.DMA read
  2.DMA write
  3.DMA verify.

It provide AEN signal that can be used to isolate CPU and other devices from the system bus. It is operate in two modes.
1.Master Mode
2.Slave Mode

8257 BLOCK DIAGRAM:



Description: It containing Five main Blocks.

1. Data bus buffer
2. Read/Control logic
3. Control logic block
4. Priority resolver
5. DMA channels.

DATA BUS BUFFER:

- It contain tri-state , 8 bit bi-directional buffer.
- Slave mode ,it transfer data between microprocessor and internal data bus.
- Master mode ,the outputs A8-A15 bits of memory address on data lines (Unidirectional).

READ/CONTROL LOGIC:

- It control all internal Read/Write operation.
- Slave mode ,it accepts address bits and control signal from microprocessor.
- Master mode ,it generate address bits and control signal.

Control logic block:

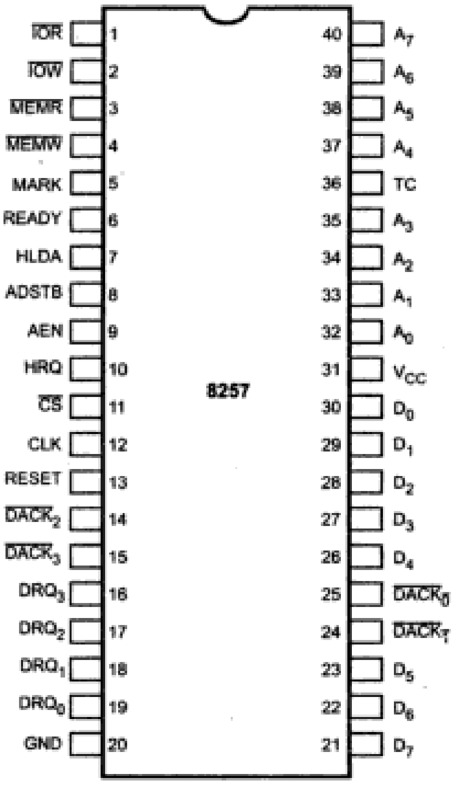
It contains ,

1. Control logic
2. Mode set register and
3. Status Register.

CONTROL LOGIC:

- Master mode ,It control the sequence of DMA operation during all DMA cycles.
- It generates address and control signals.
- It increments 16 bit address and decrement 14 bit counter registers.
- It activate a HRQ signal on DMA channel Request.
- Slave ,mode it is disabled.

Pin Diagram of DMA controller:



D0-D7:

- it is a bidirectional ,tri state ,Buffered ,Multiplexed data (D0-D7).
- In the slave mode it is a bidirectional (Data is moving).
- In the Master mode it is a unidirectional (Address is moving).

IOR & IOW:

- It is active low ,tristate ,buffered ,Bidirectional lines.
- In idle cycle, these are an input control signals used by CPU to read/write the control registers.
- In the active cycle, IOR signal is used to access data from peripheral and IOW signal is used to send data to the peripheral.

CLK:

- It is the input line ,connected with TTL clock generator.
- This signal is ignored in slave mode.

RESET:

- Used to clear mode set registers and status registers

A0-A3:

- These are the tristate, buffer, bidirectional address lines.
- In slave mode ,these lines are used as address inputs lines and internally decoded to access the internal registers.

- In master mode, these lines are used as address outputs lines,A0-A3 bits of memory address on the lines.

CS:

- It is active low, Chip select input line.
- In the slave mode, it is used to select the chip.
- In the master mode, it is ignored.

A4-A7:

- These are the tristate, buffer, output address lines.
- In slave mode ,these lines are used as address outputs lines.
- In master mode, these lines are used as address outputs lines,A0-A3 bits of memory address on the lines.

READY:

- It is a asynchronous input line.

In master mode,

- When ready is high it is received the signal.
- When ready is low, it adds wait state between S1 and S3
- In slave mode ,this signal is ignored.

HRQ:

It is used to receiving the hold request signal from the output device.

HLDA:

- It is acknowledgment signal from microprocessor.

MEMR:

- It is active low ,tristate ,Buffered control output line.
- In slave mode, it is tristated.
- In master mode ,it activated during DMA read cycle.

MEMW:

- It is active low ,tristate ,Buffered control input line.
- In slave mode, it is tristated.
- In master mode ,it activated during DMA write cycle.

AEN (Address enable):

- It is a control output line.
- In master mode ,it is high
- In slave mode ,it is low
- Used it isolate the system address ,data ,and control lines.

ADSTB: (Address Strobe)

- It is a control output line.
- Used to split data and address line.
- It is working in master mode only.
- In slave mode it is ignore.

TC (Terminal Count):

- It is a status of output line.
- It is activated in master mode only.
- It is high ,it selected the peripheral.
- It is low ,it free and looking for a new peripheral.

MARK:

- It is a modulo 128 MARK output line.
- It is activated in master mode only.
- It goes high ,after transferring every 128 bytes of data block.

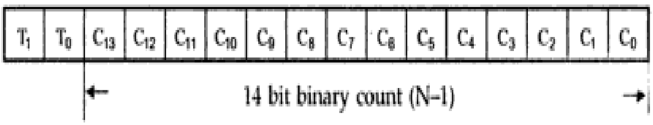
DRQ0-DRQ3(DMA Request):

- These are the asynchronous peripheral request input signal.
- The request signals is generated by external peripheral device.

DACK0-DACK3:

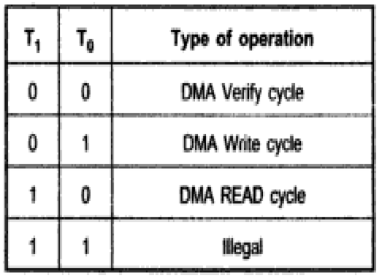
- These are the active low DMA acknowledge output lines.
- Low level indicate that ,peripheral is selected for giving the information (DMA cycle).
- In master mode it is used for chip select.

Terminal Count Register : Fig. 6.6 shows the format of terminal count register.



T_1	T_0	C_{13}	C_{12}	C_{11}	C_{10}	C_9	C_8	C_7	C_6	C_5	C_4	C_3	C_2	C_1	C_0

Fig. 6.6

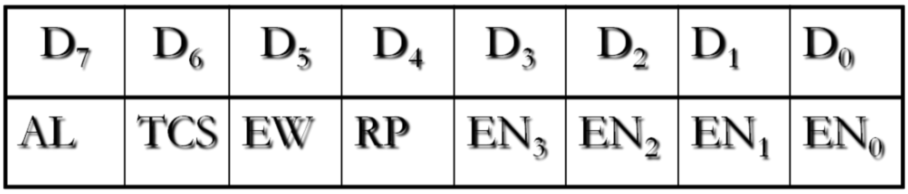


T_1	T_0	Type of operation
0	0	DMA Verify cycle
0	1	DMA Write cycle
1	0	DMA READ cycle
1	1	Illegal

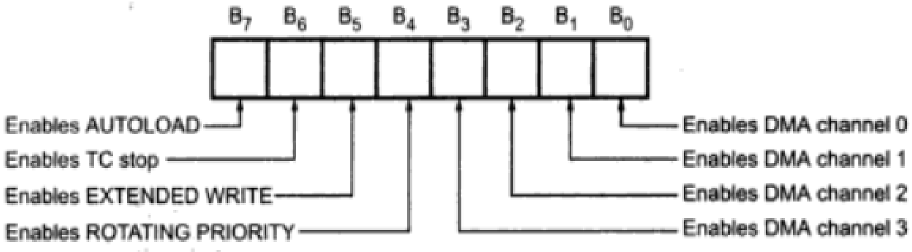
Note : N is number of bytes to be transferred.

MODE SET REGISTERS:

- The function of the mode set register is to enable the DMA channels individually and also to set the various modes of operation.
- 4 LSB's of mode set register , when set, enable each of the four DMA channels.
- 4 MSB's is used to set the operating modes.



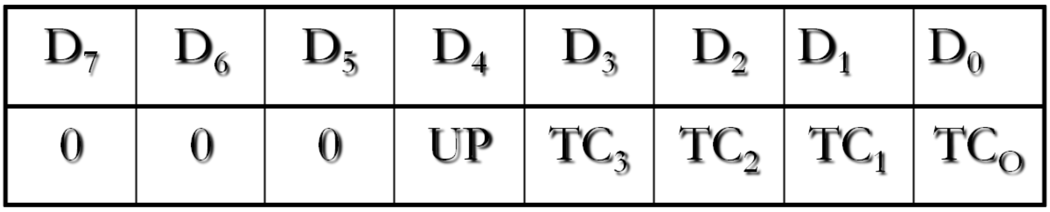
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
AL	TCS	EW	RP	EN_3	EN_2	EN_1	EN_0



- AL=1=Auto load mode
- AL=0=Rotating mode
- TCS=1=Stop after TC (Disable Channel)
- TCS=0=Start after TC (Enable Channel)
- EW=1=Extended write mode
- EW=0=normal mode.
- RP=1=Rotating priority
- RP=0=Fixed priority.
- EN_3=1=Enable DMA CH-3
- EN_3=0=Disable DMA CH-3
- EN_2=1=Enable DMA CH-2
- EN_2=0=Disable DMA CH-2
- EN_1=1=Enable DMA CH-1
- EN_1=0=Disable DMA CH-1
- EN_0=1=Enable DMA CH-0
- EN_0=0=Disable DMA CH-0

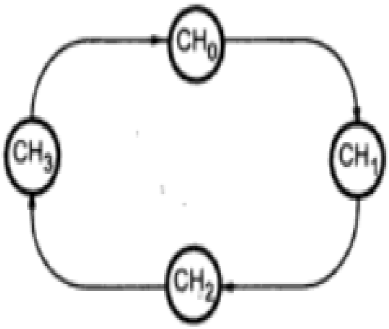
STATUS REGISTERS:

- It is read only registers.
- It is tell the status of DMA channels
- TC status bits are set when TC signal is activated for that channel.
- Update flag is not affected during read operation.
- The UP bit is set during update cycle . It is cleared after completion of update cycle.



D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	UP	TC_3	TC_2	TC_1	TC_O

- UP=Update flag
- UP=1=8257 executing update cycle
- UP=0=8257 executing DMA cycle
- TC_3=1= Terminal count has been reached for CH-3
- TC_3=0= Terminal count has not been reached for CH-3
- TC_2=1= Terminal count has been reached for CH-2
- TC_2=0= Terminal count has not been reached for CH-2
- TC_1=1= Terminal count has been reached for CH-1
- TC_1=0= Terminal count has not been reached for CH-1
- TC_0=1= Terminal count has been reached for CH-0
- TC_0=0= Terminal count has not been reached for CH-0



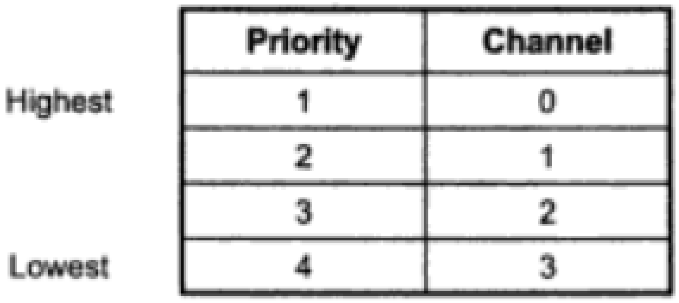
Modes of Operation:

- Rotating priority Mode
- Fixed Priority Rotating Mode
- TC Stop Mode
- Auto Load mode

- Extended Write mode

Rotating priority Mode: The priority of the channels has a circular sequence. In this channel being serviced gets the lowest priority and the channel next to it gets the highest priority.

Fixed priority mode: In the fixed priority, channel 0 has the highest priority and channel 3 has the lowest priority.



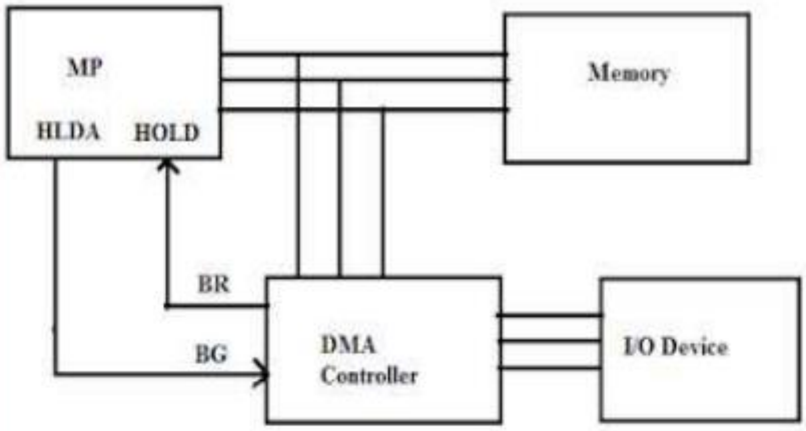
	Priority	Channel
Highest	1	0
	2	1
	3	2
Lowest	4	3

DMA CYCLES:

- **DMA READ:** In this cycle, data is transferred from memory to I/O Device
- **DMA WRITE:** In this cycle, data is transferred from I/O Device to memory
- **DMA VERIFY:** In this cycle, data is not transferred from memory to I/O Device. It is used to verify the data that has been recently transferred. To avoid overwriting registers.

Need for DMA:

Direct memory access (DMA) is the process of transferring data without the involvement of the processor itself. It is often used for transferring data to/from input/output devices. A separate DMA controller is required to handle the transfer. The controller notifies the DSP processor that it is ready for a transfer. Then the processor relinquishes control of its external memory bus and grants the control of the bus to the DMA controller. The DMA controller then transfers the specified amount of data and signals the processor upon completion of the transfer.



Block diagram of Direct Memory Access