

## UNIT 3 INTERFACING

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

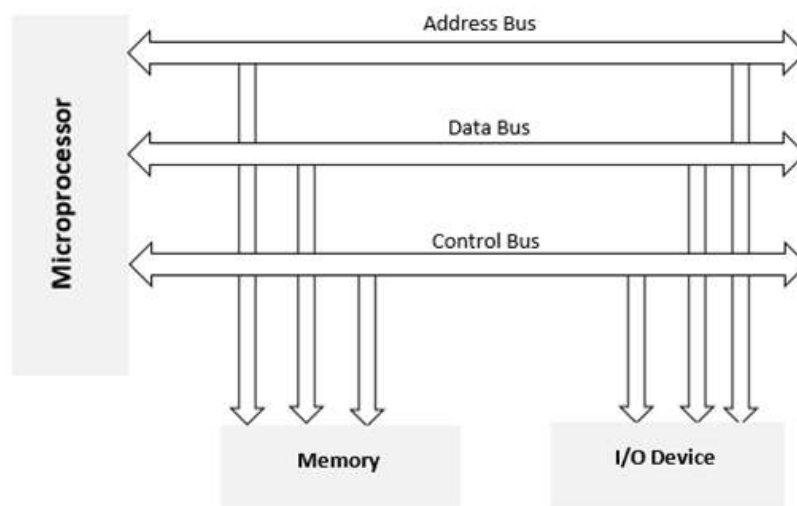
### Memory Interfacing

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 requires some signals to read from and write to registers. The interfacing process includes some key factors to match with the memory requirements and microprocessor signals. The interfacing circuit therefore should be designed in such a way that it matches the memory signal requirements with the signals of the microprocessor.

### IO 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.

Block Diagram of Memory and I/O Interfacing



### 8085 Interfacing Pins

Following is the list of 8085 pins used for interfacing with other devices –

- $A_{15} - A_8$  (Higher Address Bus)

## UNIT 3

- $AD_7 - AD_0$ (Lower Address/Data Bus)
- ALE
- RD
- WR
- READY

### Ways of Communication – Microprocessor with the Outside World?

There are two ways of communication in which the microprocessor can connect with the outside world.

- Serial Communication Interface
- Parallel Communication interface

**Serial Communication Interface** – In this type of communication, the interface gets a single byte of data from the microprocessor and sends it bit by bit to the other system serially and vice-a-versa.

**Parallel Communication Interface** – In this type of communication, the interface gets a byte of data from the microprocessor and sends it bit by bit to the other systems in simultaneous (or) parallel fashion and vice-a-versa.

### 8257 DMA Controller

DMA stands for Direct Memory Access. It is designed by Intel to transfer data at the fastest rate. It allows the device to transfer the data directly to/from memory without any interference of the CPU. Using a DMA controller, the device requests the CPU to hold its data, address and control bus, so the device is free to transfer data directly to/from the memory. The DMA data transfer is initiated only after receiving HLDA signal from the CPU.

### How DMA Operations are Performed?

Following is the sequence of operations performed by a DMA –

- Initially, when any device has to send data between the device and the memory, the device has to send DMA request (DRQ) to DMA controller.
- The DMA controller sends Hold request (HRQ) to the CPU and waits for the CPU to assert the HLDA.

## UNIT 3

- Then the microprocessor tri-states all the data bus, address bus, and control bus. The CPU leaves the control over bus and acknowledges the HOLD request through HLDA signal.
- Now the CPU is in HOLD state and the DMA controller has to manage the operations over buses between the CPU, memory, and I/O devices.

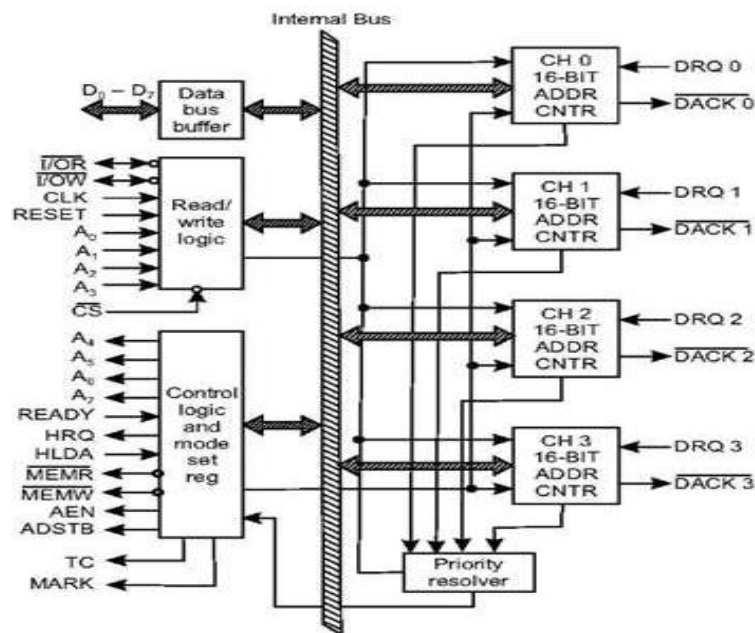
## Features of 8257

Here is a list of some of the prominent features of 8257 –

- It has four channels which can be used over four I/O devices.
- Each channel has 16-bit address and 14-bit counter.
- Each channel can transfer data up to 64kb.
- Each channel can be programmed independently.
- Each channel can perform read transfer, write transfer and verify transfer operations.
- It generates MARK signal to the peripheral device that 128 bytes have been transferred.
- It requires a single phase clock.
- Its frequency ranges from 250Hz to 3MHz.
- It operates in 2 modes, i.e., **Master mode** and **Slave mode**.

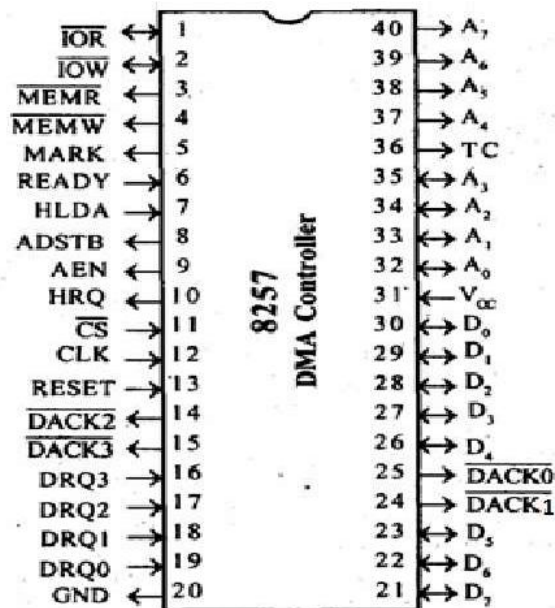
## 8257 Architecture

The following image shows the architecture of 8257 –



### 8257 Pin Description

The following image shows the pin diagram of a 8257 DMA controller –



**$DRQ_0 - DRQ_3$**  - These are the four-individual channel DMA request inputs, which are used by the peripheral devices for using DMA services. When the fixed priority mode is selected, then  $DRQ_0$  has the highest priority and  $DRQ_3$  has the lowest priority among them.

**$\overline{DACK_0} - \overline{DACK_3}$**  - These are the active-low DMA acknowledge lines, which updates the requesting peripheral about the status of their request by the CPU. These lines can also act as strobe lines for the requesting devices.

## UNIT 3

**D<sub>0</sub> – D<sub>7</sub>** - These are bidirectional, data lines which are used to interface the system bus with the internal data bus of DMA controller. In the Slave mode, it carries command words to 8257 and status word from 8257. In the master mode, these lines are used to send higher byte of the generated address to the latch. This address is further latched using ADSTB signal.

**IOR** - It is an active-low bidirectional tri-state input line, which is used by the CPU to read internal registers of 8257 in the Slave mode. In the master mode, it is used to read data from the peripheral devices during a memory write cycle.

**IOW** - It is an active low bi-direction tri-state line, which is used to load the contents of the data bus to the 8-bit mode register or upper/lower byte of a 16-bit DMA address register or terminal count register. In the master mode, it is used to load the data to the peripheral devices during DMA memory read cycle.

**CLK** - It is a clock frequency signal which is required for the internal operation of 8257.

**RESET** - This signal is used to RESET the DMA controller by disabling all the DMA channels.

**A<sub>0</sub> - A<sub>3</sub>** - These are the four least significant address lines. In the slave mode, they act as an input, which selects one of the registers to be read or written. In the master mode, they are the four least significant memory address output lines generated by 8257.

**CS** - It is an active-low chip select line. In the Slave mode, it enables the read/write operations to/from 8257. In the master mode, it disables the read/write operations to/from 8257.

**A<sub>4</sub> - A<sub>7</sub>** - These are the higher nibble of the lower byte address generated by DMA in the master mode.

**READY** - It is an active-high asynchronous input signal, which makes DMA ready by inserting wait states.

**HRQ** - This signal is used to receive the hold request signal from the output device. In the slave mode, it is connected with a DRQ input line 8257. In Master mode, it is connected with HOLD input of the CPU.

**HLDA** - It is the hold acknowledgement signal which indicates the DMA controller that the bus has been granted to the requesting peripheral by the CPU when it is set to 1.

**MEMR** - It is the low memory read signal, which is used to read the data from the addressed memory locations during DMA read cycles.

**MEMW** - It is the active-low three state signal which is used to write the data to the addressed memory location during DMA write operation.

**ADST** - This signal is used to convert the higher byte of the memory address generated by the DMA controller into the latches.

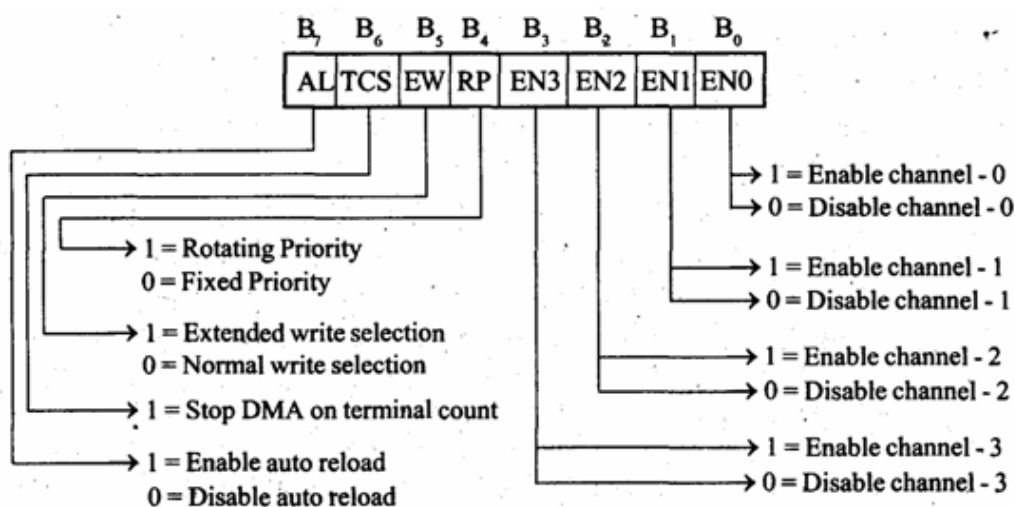
**AEN** - This signal is used to disable the address bus/data bus.

**TC** - It stands for 'Terminal Count', which indicates the present DMA cycle to the present peripheral devices.

**MARK** - The mark will be activated after each 128 cycles or integral multiples of it from the beginning. It indicates the current DMA cycle is the 128th cycle since the previous MARK output to the selected peripheral device.

**V<sub>cc</sub>** - It is the power signal which is required for the operation of the circuit.

### Control word format



### 8255A - Programmable Peripheral Interface

The 8255A is a general purpose programmable I/O device designed to transfer the data from I/O to interrupt I/O under certain conditions as required. It can be used with almost any microprocessor.

It consists of three 8-bit bidirectional I/O ports (24I/O lines) which can be configured as per the requirement.

#### Ports of 8255A

8255A has three ports, i.e., PORT A, PORT B, and PORT C.

- **Port A** contains one 8-bit output latch/buffer and one 8-bit input buffer.
- **Port B** is similar to PORT A.
- **Port C** can be split into two parts, i.e. PORT C lower (PC0-PC3) and PORT C upper (PC7-PC4) by the control word.

These three ports are further divided into two groups, i.e. Group A includes PORT A and upper PORT C. Group B includes PORT B and lower PORT C. These two groups can be programmed in three different modes, i.e. the first mode is named as mode 0, the second mode is named as Mode 1 and the third mode is named as Mode 2.

#### Operating Modes

8255A has three different operating modes –

- **Mode 0** – In this mode, Port A and B is used as two 8-bit ports and Port C as two 4-bit ports. Each port can be programmed in either input mode or output mode where outputs are latched and inputs are not latched. Ports do not have interrupt capability.
- **Mode 1** – In this mode, Port A and B is used as 8-bit I/O ports. They can be configured as either input or output ports. Each port uses three lines from port C as handshake signals. Inputs and outputs are latched.
- **Mode 2** – In this mode, Port A can be configured as the bidirectional port and 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.

#### Features of 8255A

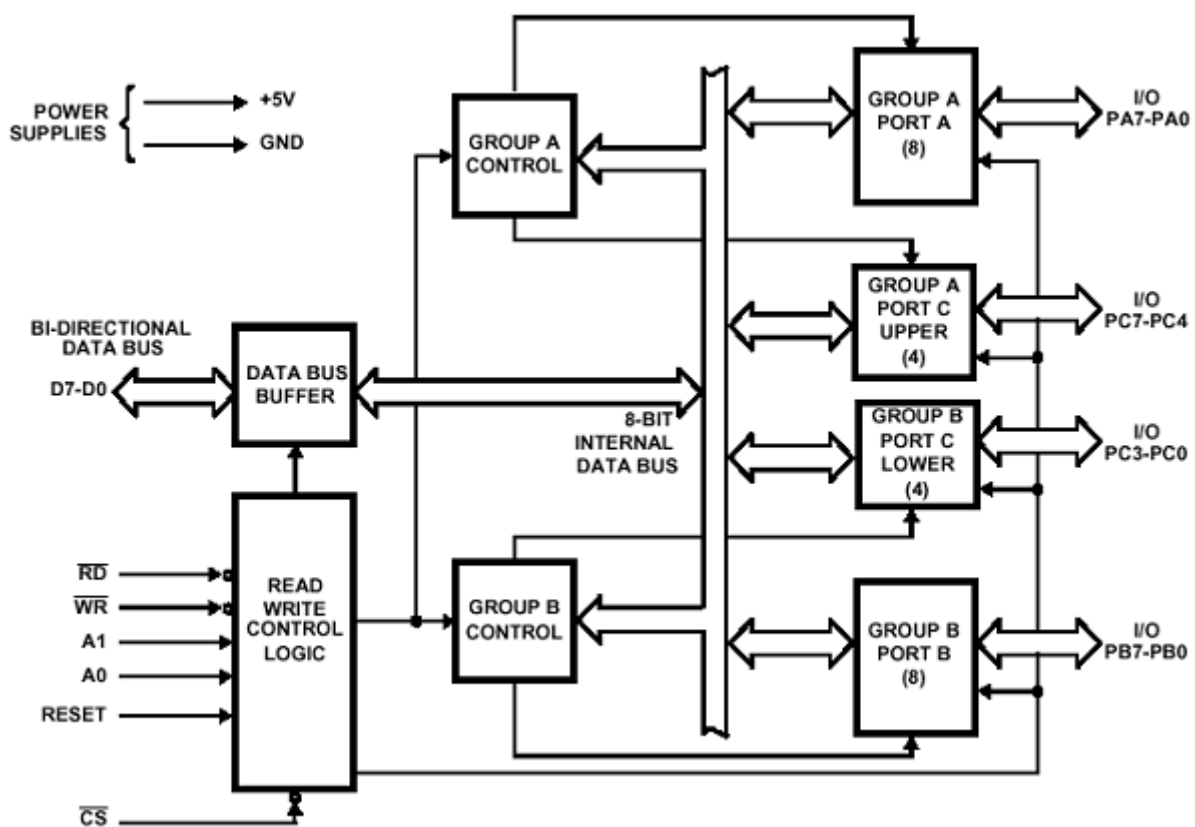
## UNIT 3

The prominent features of 8255A are as follows –

- It consists of 3 8-bit IO ports i.e. PA, PB, and PC.
- Address/data bus must be externally demux'd.
- It is TTL compatible.
- It has improved DC driving capability.

## 8255 Architecture

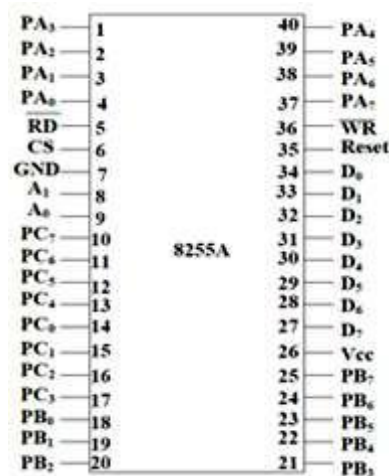
The following figure shows the architecture of 8255A –



Let us first look at the pin diagram of Intel 8255A –



## UNIT 3



Now let us discuss the functional description of the pins in 8255A.

### Data Bus Buffer

It is a tri-state 8-bit buffer, which is used to interface the microprocessor to the system data bus. Data is transmitted or received by the buffer as per the instructions by the CPU. Control words and status information is also transferred using this bus.

### Read/Write Control Logic

This block is responsible for controlling the internal/external transfer of data/control/status word. It accepts the input from the CPU address and control buses, and in turn issues command to both the control groups.

### CS

It stands for Chip Select. A LOW on this input selects the chip and enables the communication between the 8255A and the CPU. It is connected to the decoded address, and  $A_0$  &  $A_1$  are connected to the microprocessor address lines.

Their result depends on the following conditions –

CS	$A_1$	$A_0$	Result
0	0	0	PORT A
0	0	1	PORT B

## UNIT 3

0	1	0	PORT C
0	1	1	Control Register
1	X	X	No Selection

**WR**

It stands for write. This control signal enables the write operation. When this signal goes low, the microprocessor writes into a selected I/O port or control register.

**RESET**

This is an active high signal. It clears the control register and sets all ports in the input mode.

**RD**

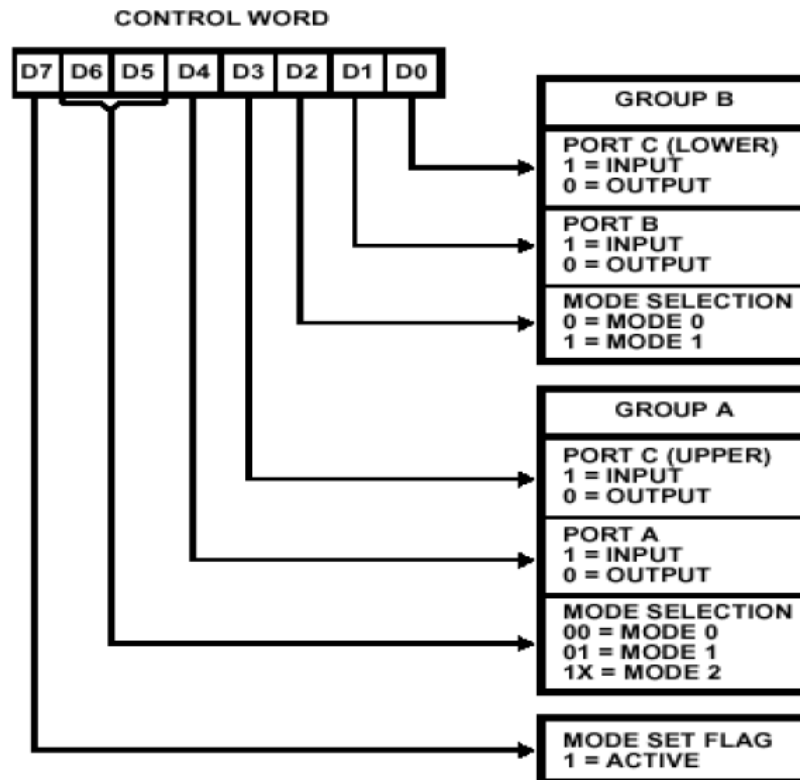
It stands for Read. This control signal enables the Read operation. When the signal is low, the microprocessor reads the data from the selected I/O port of the 8255.

**A<sub>0</sub> and A<sub>1</sub>**

These input signals work with RD, WR, and one of the control signal. Following is the table showing their various signals with their result.

A <sub>1</sub>	A <sub>0</sub>	RD	WR	CS	Result
0	0	0	1	0	<b><u>Input Operation</u></b> PORT A → Data Bus
0	1	0	1	0	PORT B → Data Bus
1	0	0	1	0	PORT C → Data Bus
0	0	1	0	0	<b><u>Output Operation</u></b> Data Bus → PORT A
0	1	1	0	0	Data Bus → PORT A
1	0	1	0	0	Data Bus → PORT B

1	1	1	0	0	Data Bus → PORT D
---	---	---	---	---	-------------------



### Intel 8253/54 - Programmable Interval Timer

The Intel 8253 and 8254 are Programmable Interval Timers (PTIs) designed for microprocessors to perform timing and counting functions using three 16-bit registers. 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.

Difference between 8253 and 8254

The following table differentiates the features of 8253 and 8254 –

8253	8254
Its operating frequency is 0 - 2.6 MHz	Its operating frequency is 0 - 10 MHz
It uses N-MOS technology	It uses H-MOS technology

Read-Back command is not available	Read-Back command is available
Reads and writes of the same counter cannot be interleaved.	Reads and writes of the same counter can be interleaved.

**Features of 8253/54**

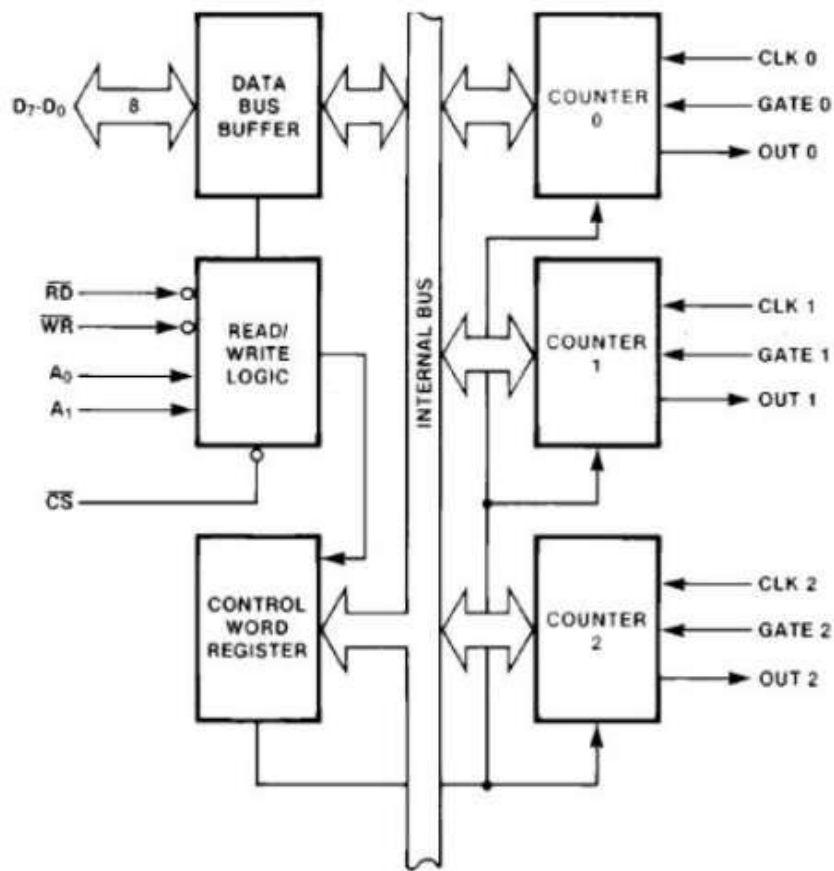
The most prominent features of 8253/54 are as follows –

- It has three independent 16-bit down counters.
- It can handle inputs from DC to 10MHz.
- These three counters can be programmed for either binary or BCD count.
- It is compatible with almost all microprocessors.
- 8254 has a powerful command called READ BACK command, which allows the user to check the count value, the programmed mode, the current mode, and the status of the counter.

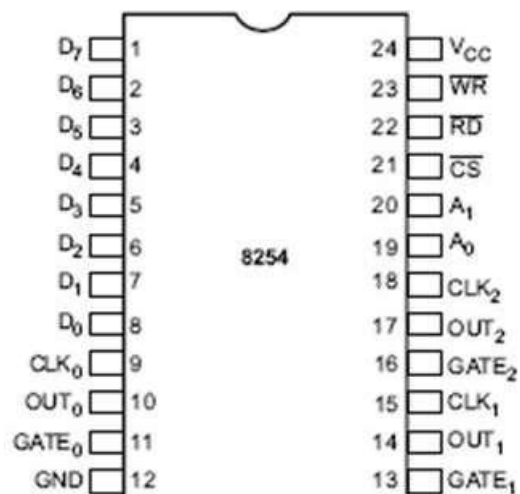
**8254 Architecture**

The architecture of 8254 looks as follows –

## UNIT 3

**8254 Pin Description**

Here is the pin diagram of 8254 –



In the above figure, there are three counters, a data bus buffer, Read/Write control logic, and a control register. Each counter has two input signals - CLOCK & GATE, and one output signal - OUT.

**Data Bus Buffer**

It is a tristate, bi-directional, 8-bit buffer, which is used to interface the 8253/54 to the system data bus. It has three basic functions –

- Programming the modes of 8253/54.
- Loading the count registers.
- Reading the count values.

**Read/Write Logic**

It includes 5 signals, i.e. RD, WR, CS, and the address lines  $A_0$  &  $A_1$ . In the peripheral I/O mode, the RD and WR signals are connected to IOR and IOW, respectively. In the memory mapped I/O mode, these are connected to MEMR and MEMW.

Address lines  $A_0$  &  $A_1$  of the CPU are connected to lines  $A_0$  and  $A_1$  of the 8253/54, and CS is tied to a decoded address. The control word register and counters are selected according to the signals on lines  $A_0$  &  $A_1$ .

$A_1$	$A_0$	Result
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2
1	1	Control Word Register
X	X	No Selection

**Control Word Register**

This register is accessed when lines  $A_0$  &  $A_1$  are at logic 1. It is used to write a command word, which specifies the counter to be used, its mode, and either a read or write operation. Following table shows the result for various control inputs.

## UNIT 3

A <sub>1</sub>	A <sub>0</sub>	RD	WR	CS	Result
0	0	1	0	0	Write Counter 0
0	1	1	0	0	Write Counter 1
1	0	1	0	0	Write Counter 2
1	1	1	0	0	Write Control Word
0	0	0	1	0	Read Counter 0
0	1	0	1	0	Read Counter 1
1	0	0	1	0	Read Counter 2
1	1	0	1	0	No operation
X	X	1	1	0	No operation
X	X	X	X	1	No operation

**Counters**

Each counter consists of a single, 16 bit-down counter, which can be operated in either binary or BCD. Its input and output is configured by the selection of modes stored in the control word register. The programmer can read the contents of any of the three counters without disturbing the actual count in process.

**Control Word Format**

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
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)

**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

**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

**BCD**

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

**Intel 8253/54 - Operational Modes**

8253/54 can be operated in 6 different modes. In this chapter, we will discuss these operational modes.

**Mode 0 — Interrupt on Terminal Count**

- It is used to generate an interrupt to the microprocessor after a certain interval.
- Initially the output is low after the mode is set. The output remains LOW after the count value is loaded into the counter.
- The process of decrementing the counter continues till the terminal count is reached, i.e., the count become zero and the output goes HIGH and will remain high until it reloads a new count.
- The GATE signal is high for normal counting. When GATE goes low, counting is terminated and the current count is latched till the GATE goes high again.

**Mode 1 – Programmable One Shot**

- It can be used as a mono stable multi-vibrator.
- The gate input is used as a trigger input in this mode.



## UNIT 3

- The output remains high until the count is loaded and a trigger is applied.

## Mode 2 – Rate Generator

- The output is normally high after initialization.
- Whenever the count becomes zero, another low pulse is generated at the output and the counter will be reloaded.

## Mode 3 – Square Wave Generator

- This mode is similar to Mode 2 except the output remains low for half of the timer period and high for the other half of the period.

## Mode 4 – Software Triggered Mode

- In this mode, the output will remain high until the timer has counted to zero, at which point the output will pulse low and then go high again.
- The count is latched when the GATE signal goes LOW.
- On the terminal count, the output goes low for one clock cycle then goes HIGH. This low pulse can be used as a strobe.

## Mode 5 – Hardware Triggered Mode

- This mode generates a strobe in response to an externally generated signal.
- This mode is similar to mode 4 except that the counting is initiated by a signal at the gate input, which means it is hardware triggered instead of software triggered.
- After it is initialized, the output goes high.
- When the terminal count is reached, the output goes low for one clock cycle.

**INTEL 8259A Programmable Interrupt Controller**

The 8259A is a programmable interrupt controller designed to work with Intel microprocessor 8080 A, 8085, 8086, 8088. The 8259 A interrupt controller can

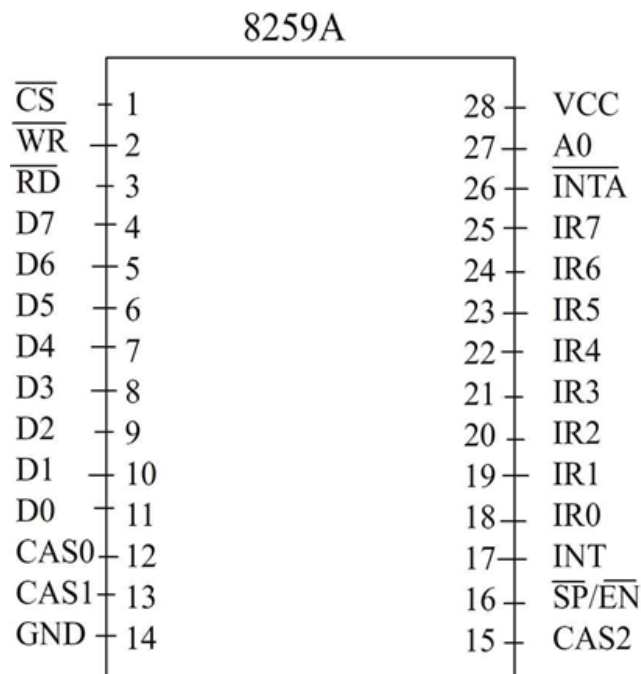
- 1) Handle eight interrupt inputs. This is equivalent to providing eight interrupt pins on the processor in place of one INTR/INT pin.
- 2) Vector an interrupt request anywhere in the memory map. However, all the eight interrupt are spaced at the interval of either four or eight location. This

## UNIT 3

eliminates the major drawback, 8085 interrupt, in which all interrupts are vectored to memory location on page 00H.

- 3) Resolve eight levels of interrupt priorities in a variety of modes.
- 4) Mask each interrupt request individually.
- 5) Read the status of pending interrupts, in service interrupts, and masked interrupts.
- 6) Be set up to accept either the level triggered or edge triggered interrupt request.
- 7) Mine 8259 as can be cascade in a master slave configuration to handle 64 interrupt inputs.

The 8259 A is contained in a 28-element in line package that requires only a compatible with 8259. The main difference between the two is that the 8259 A can be used with Intel 8086/8088 processor. It also induces additional features such as level triggered mode, buffered mode and automatic end of interrupt mode. The pin diagram and interval block diagram is shown below:



The pins are defined as follows:

**$\overline{CS}$**  Chip select - To access this chip,  $\overline{CS}$  is made low. A LOW on this pin enables  **$\overline{RD}$**  &  **$\overline{WR}$**  communication between the CPU and the 8259A. This pin is connected to address bus through the decoder logic circuits.

**$\overline{WR}$**  - A low on this pin. When  **$\overline{CS}$**  is low enables the 8259 A to accept command words from CPU.

**$\overline{RD}$**  - A low on this pin when  $\overline{CS}$  is low enables these 8259 A to release status on to the data bus for the CPU. The status includes the contents of IMR, ISR or TRR register or a priority level.

D7-D0 - Bidirectional data bus control status and interrupt in a this bus. This bus is connected to BDB of 8085.

CAS0-CAS2 - Cascade lines: The CAS lines form a private 8259A bus to control a multiple 8259A structure ie to identify a particular slave device. These pins are outputs of a master 8259A and inputs for a slave 8259A.

**$\overline{SP}$  /  $\overline{EN}$** : Slave program/enable buffer: This is a dual function pin. It is used as an input to determine whether the 8259A is to a master or as a slave .It is also used as an output to disable the data bus transceivers when data are being transferred from the 8259A to the CPU. When in buffered mode, it can be used as an output and when not in the buffered mode it is used as an input.

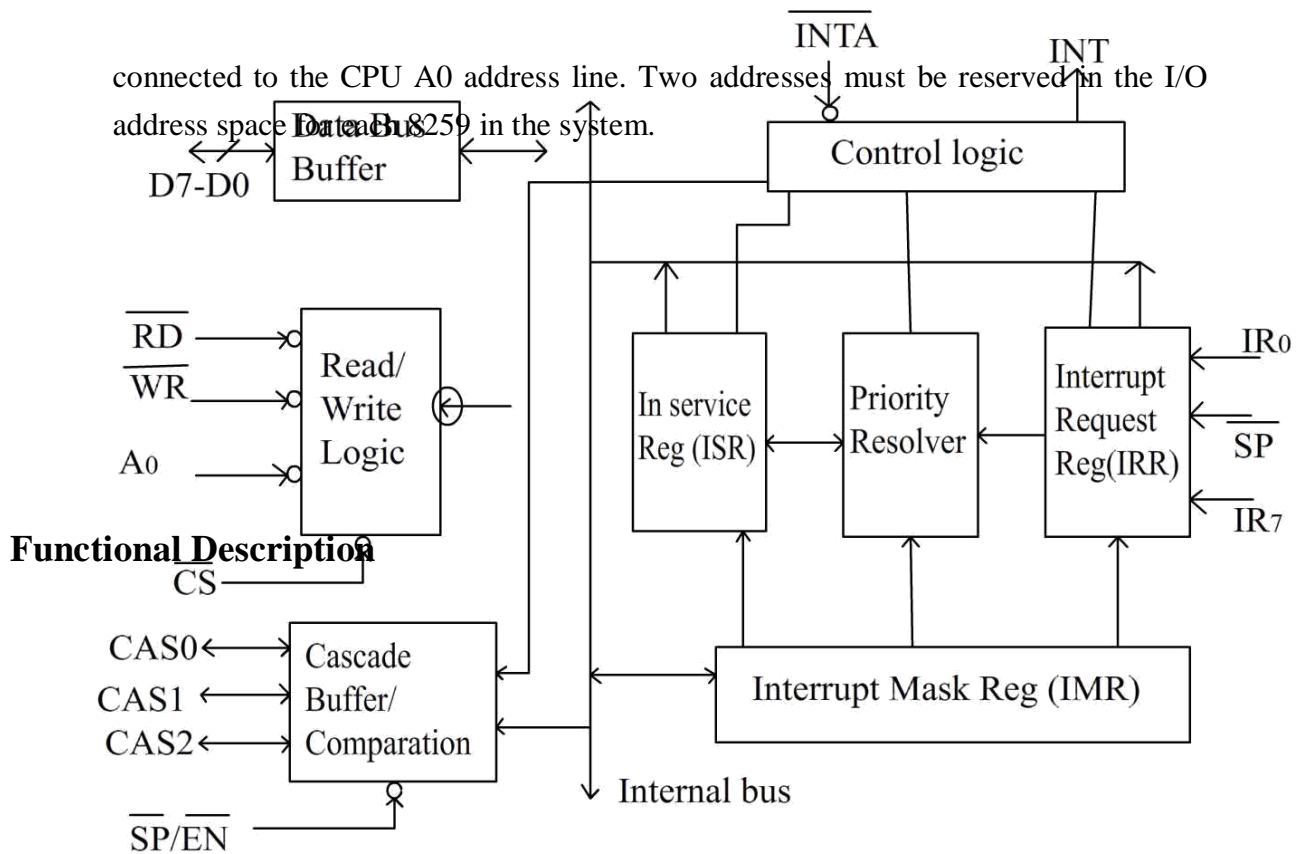
INT - This pin goes high whenever a valid interrupt request is asserted. It is used to interrupt the CPU; thus, it is connected to the CPU's interrupt pin (INTR).

**$\overline{INTA}$**  - Interrupt: Acknowledge. This pin is used to enable 8259A interrupt vector data on the data bus by a sequence of interrupt request pulses issued by the CPU.

IR0-IR7 - Interrupt Requests: Asynchronous interrupt 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 (levels triggered mode).

A0 - A0 address line: This pin acts in conjunction with the  $\overline{CS}$  &  $\overline{RD}$  pins. It is used by the 8259A to send various command words from the CPU and to read the status. If is

## UNIT 3



The 8259 A has eight interrupt request inputs, TR2 IR0. The 8259 A uses its INT output to interrupt the 8085A via INTR pin. The 8259A receives interrupt acknowledge pulses from the at its input. Vector address used by the 8085 A to transfer control to the service subroutine of the interrupting device, is provided by the 8259 A on the data bus. The 8259A is a programmable device that must be initialized by command words sent by the. After

initialization, the 8259 A mode of operation can be changed by operation command words from the. The descriptions of various blocks are,

**Data bus buffer** - This 3- state, 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 CPU. It contains the initialization command word (ICW) register and operation command word (OCW) register which store the various control formats for device operation. This function block also allows the status of 8159A to be transferred to the data bus.

**Interrupt request register (IRR)** - IRR stores all the interrupt inputs that are requesting service. Basically, it keeps track of which interrupt inputs are asking for service. If an interrupt input is unmasked, and has an interrupt signal on it, then the corresponding bit in the IRR will be set.

**Interrupt mask register (IMR)** -The IMR is used to disable (Mask) or enable (Unmask) individual interrupt inputs. Each bit in this register corresponds to the interrupt input with the same number. The IMR operation on the IRR. Masking of higher priority input will not affect the interrupt request lines of lower priority. To unmask any interrupt the corresponding bit is set '0'.

**In service register (ISR)** - The in-service registers keeps tracks of which interrupt inputs are currently being serviced. For each input that is currently being serviced the corresponding bit will be set in the in-service register. Each of these 3-reg can be read as status reg.

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

**Cascade buffer/comparator** - This function blocks stores and compare the IDS of all 8259A's in the reg. The associated 3-I/O pins (CAS0-CAS2) are outputs when 8259A is used a master. Master and are inputs when 8259A is used as a slave. As a master, the 8259A sends the ID of the interrupting slave device onto the cas2-cas0. The slave thus selected will send its pre-programmed subroutine address on to the data bus during the next one or two successive **INTA** pulses.