Success Classes Bijnor

8085 Microprocessor

(Interfacing with 8085 - Overview)

(Memory Interfacing and I/O Interfacing)

By Vimal Sir

For PDF join our Telegram Channel from description

- **Microprocessor - I/O Interfacing with 8085 - Overview**

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

➢ Memory Interfacing

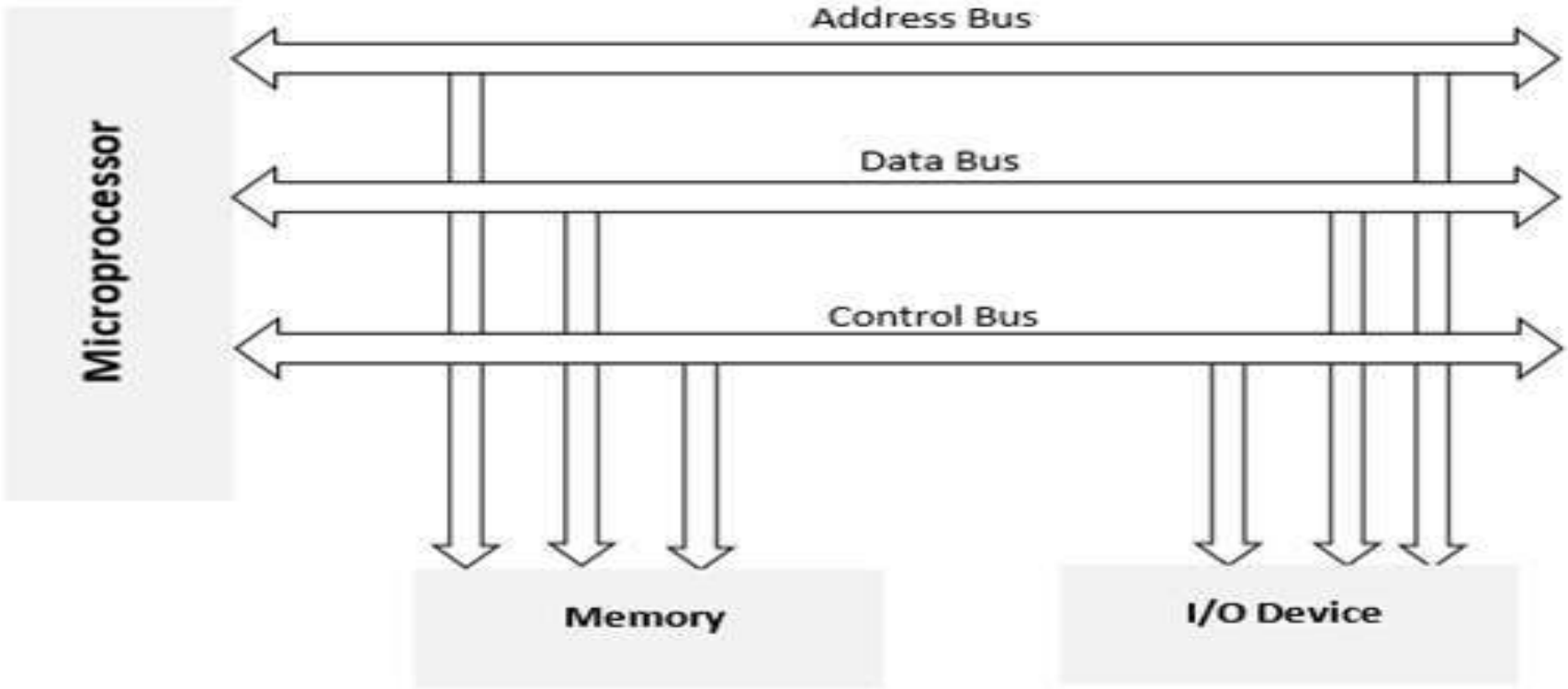
• When we are executing any instruction, we need the microprocessor to access the memory for reading instruction codes and the data stored in the memory. For this, both the memory and the microprocessor requires some signals to read from and write to registers.

• 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.

- **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.

- **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)
- ❖ 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.

THANK YOU For Watching

Success Classes Bijnor

DMA Controller

Today's Target

- ✓ 8237 DMA Controller
- ✓ How DMA Operations are Performed?
- ✓ Features of 8237 DMA Controller

By Vimal Sir

For PDF join our Telegram Channel from description

- **Microprocessor - 8237 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.
- 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 8237 DMA Controller:**

- It provides various modes of Direct memory access (DMA).
- It provides on-chip four independent DMA channels. The number of channels can be increased by cascading DMA controller chips.
- Each channel can be used in auto initialization mode.
- It can transfer data between two memory clocks in DMA mode i.e. memory to memory transfer.
- In memory-to-memory transfer, a single word can be written into all locations of memory blocks.

- The address of memory is either incremented or decremented after each DM cycle depending upon the mode.
- The clock frequency is 3 MHz (8237 DMA C0ntroller) or 5 MHz (8237-2 DMA Controller).
- The data transfer rate is very high i.e. 1.6 Mbytes/second for 8237-2 at 5 MHz.
- Directly expandable to any number of channels. It doesn't require any additional chip for cascading. There are no limitations in cascading.
- The DMA can be requested by setting an appropriate bit for the request register.
- It provides compressed timings to improve the throughput of the system. It can compress the transfer time to two cycles (2s).

THANK YOU For Watching

Success Classes Bijnor

DMA Controller

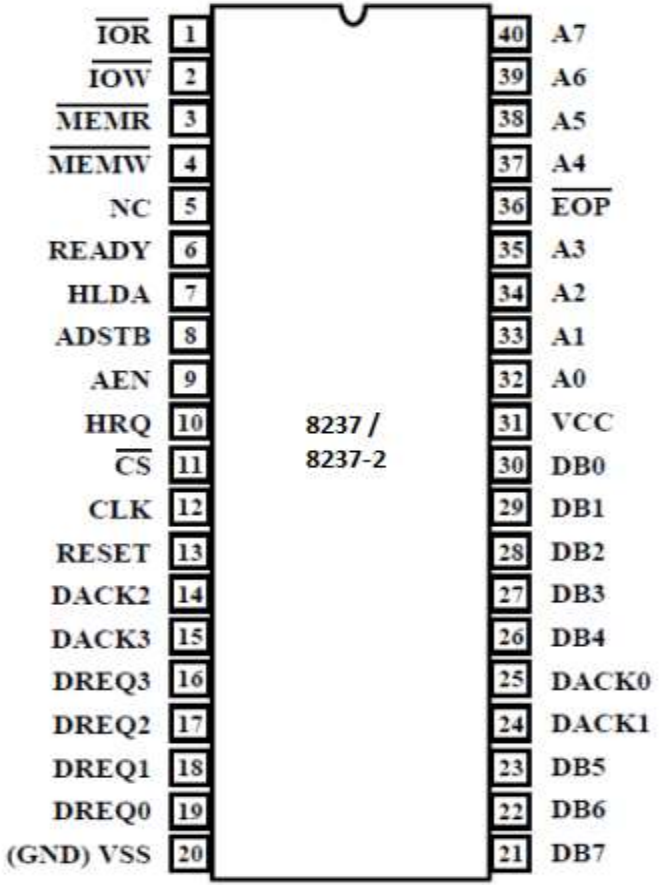
Today's Target

- ✓ **PIN Diagram of 8237 DMA Controller**

By Vimal Sir

For PDF join our Telegram Channel from description

- **PIN Configuration of 8237 DMA Controller**



Symbol	Description
CLK	This is the clock input line ignored in slave mode. In master mode, this signal control all internal and external DMA operations. The data transfer rate depends upon the frequency of this signal.
CS(bar)	In slave mode, this signal is generated by the address decoder to select the 8237 chip for communication between the CPU and 8237. In master mode, this signal is ignored.
Reset	It is an asynchronous input line. This signal clears the command, status, request, and temporary register and forces 8237 into slave mode.
READY	In master mode, this signal is used to add wait states into the DMA cycle.

HRQ	It is a hold request output line. It is connected to hold the input of the CPU. it is used to request control of the system bus.
HLDA	It is a hold acknowledge input line. This signal is generated by the CPU. In response to this signal, the 8237 gains control of the system bus and enters master mode.
IOR(bar)	It is an active low bi-directional tri-state line. In slave mode, it acts as an input line and is used to read the contents of the 8237 register. In master mode, it acts as an output line. This signal is generated during the DMA cycle to read data from the I/O device.
IOW(bar)	It is an active low bi-directional tri-state line. In slave mode, it acts as an input line and is used to write the contents to the 8237 register. In master mode, it acts as an output line. This signal is generated during the DMA read cycle to write data into the I/O device.

$A_0 - A_3$	These are bi-directional, address lines. In slave mode, these lines act as input lines, used to select one of the registers of 8237. In master mode, the 8237 provides lower bits of memory address on these lines.
$A_4 - A_7$	These are tristate address output lines. These lines are tri-stated in slave mode. In master mode, the 827 transfers bits of memory addressed on these lines.
MEMR(bar)	It is an active low tristate output line. it is tri-stated in slave mode. In master mode, this signal is generated during the DMA read cycle or during memory to memory transfer cycle to read the contents of source memory.
MEMW(bar)	It is an active low tristate output line. it is tri-stated in slave mode. In master mode, this signal is activated during the DMA write or during the memory-to-memory transfer cycle to write data into destination memory.

DB_0 – DB_7	These are bi-directional tristate buffered data lines. In slave mode, these lines are used to transfer data between the CPU and 8237 registers. In master mode, these lines act as address output lines. The 8237 places a higher byte of address on these lines during DMA cycles.
$DREQ_0$ – $DREQ_3$	These are asynchronous DMA channel request lines used by the peripheral. The polarity of each signal is programmable i.e. these lines can be used as either active high or active low input. DREQ must be maintained until the corresponding DACK is activated.
$DACK_0$ – $DACK_3$	These are DMA acknowledge output lines. The polarity of each line is programmable. The signal indicates that the requesting peripheral has been granted for the DMA cycle.
EOP(bar)	End of Process: It is an active low bi-directional signal This line is also used to terminate the DMA cycle. The DMA cycle can be terminated by pulling ¯**EOP** input low. The 8237 also generates an EOP pulse, when the terminal count for any channel is reached.

THANK YOU For Watching

Success Classes Bijnor

DMA Controller

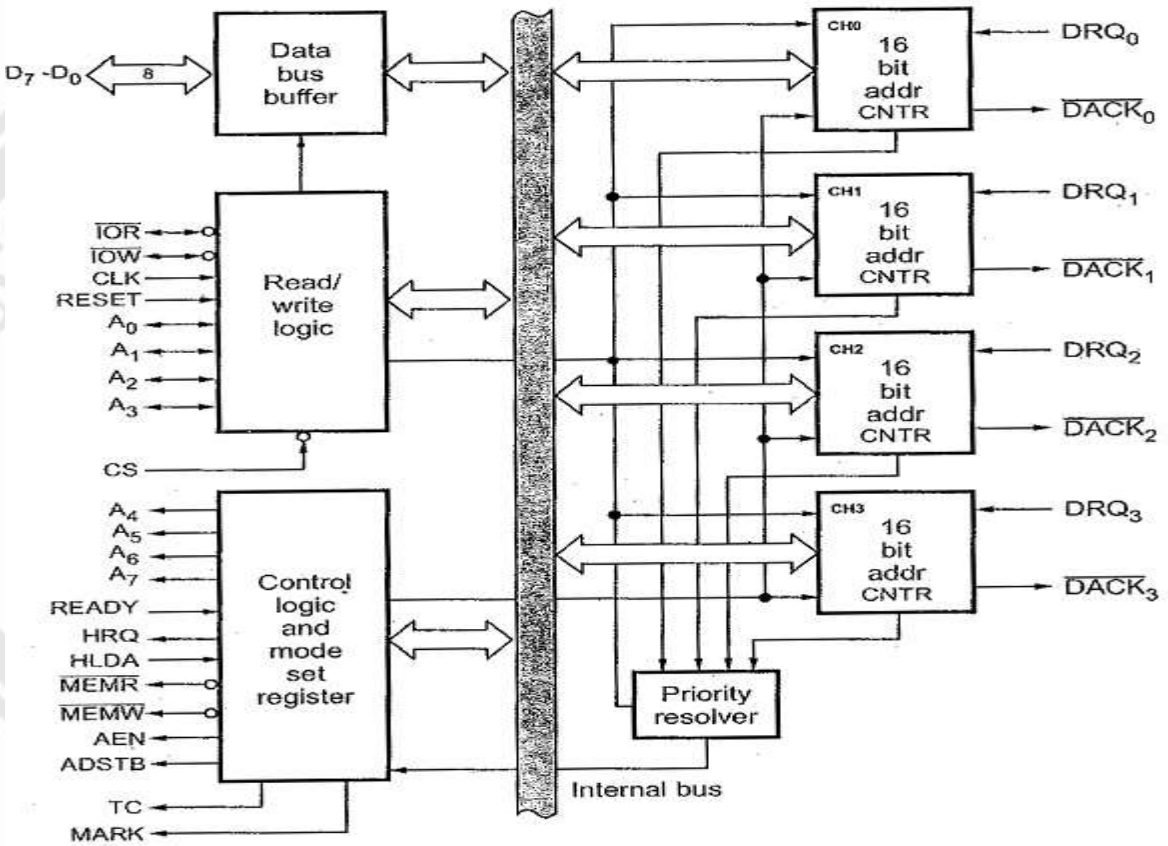
Today's Target

- ✓ **Block Diagram of DMA Controller**

By Vimal Sir

For PDF join our Telegram Channel from description

- **Block Diagram of DMA Controller:**



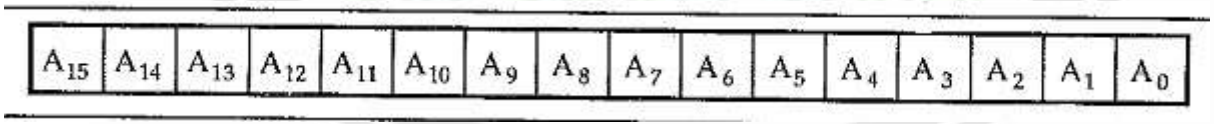
Data Bus Buffer: It is a tri-state, bi-directional, eight bit buffer which interfaces the 8257 to the system data bus. In the slave mode, it is used to transfer data between microprocessor and internal registers of 8257. In master mode, it is used to send higher byte address (A8-A15) on the data bus.

Read/Write logic: When the CPU is programming or reading one of the internal registers of Pin Diagram of 8257 (i.e, when the 8257 is in the slave mode), the Read/Write logic accepts the I/O Read (IOR) or I/O Write (IOW) signal, decodes the least significant four address bits (A0 – A3) and either writes the contents of the data bus into the addressed register (if IOW is low) or places the contents of the addressed register onto the data bus (if IOR is low). During DMA cycles (i.e. when the 8257 is in the master mode) the Read/Write logic generates the I/O read and memory write (DMA write cycle) or I/O write and memory read (DMA read cycle) signals which control the data transfer between peripheral and memory device.

DMA Channels: The Pin Diagram of 8257 provides four identical channels, labeled CH0 to CH3. Each channel has two sixteen bit registers:

1. DMA address register, and
2. Terminal count register.

DMA address register : It specifies the address of the first memory location to be accessed. It is necessary to load valid memory address in the DMA address register before channel is enabled.



Terminal Count Register :

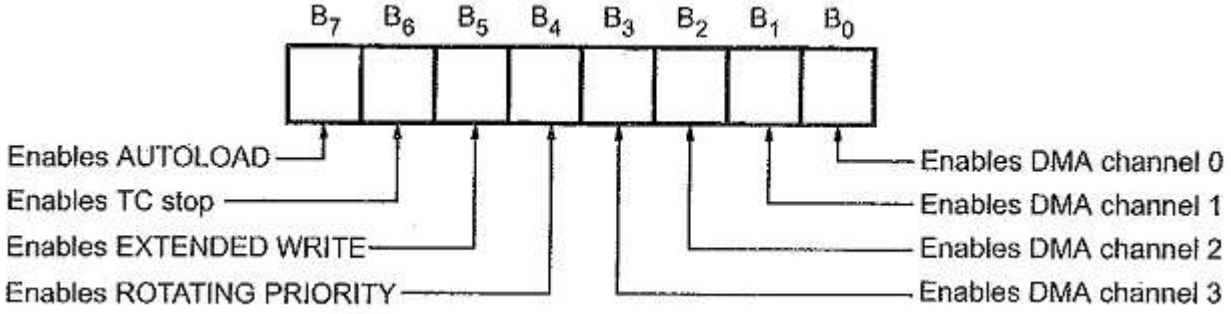
T_1	T_0	C_{13}	C_{12}	C_{11}	C_{10}	C_9	C_8	C_7	C_6	C_5	C_4	C_3	C_2	C_1	C_0

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

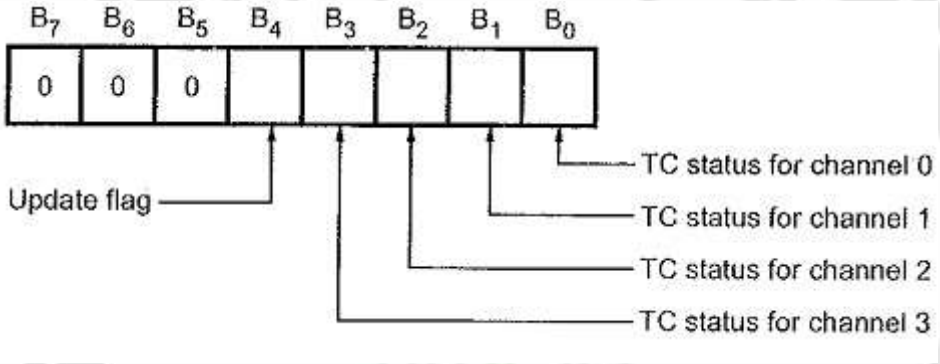
The value loaded into the low order 14 bits (C13 — C0) of the terminal count register specifies the number of DMA cycles minus one before the terminal count (TC) output is activated. Therefore, for N number of desired DMA cycles it is necessary to load the value N-1 into the low order 14-bits of the terminal count register. The most significant 2 bits of the terminal count register specifies the type of DMA operation to be performed. It is necessary to load count for DMA cycles and operational code for valid DMA cycle in the terminal count register before channel is enabled.

Control logic: It controls the sequence of operations during all DMA cycles (DMA read, DMA write, DMA verify) by generating the appropriate control signals and the 16-bit address that specifies the memory location to be accessed. It consists of mode set register and status register. Mode set register is programmed by the CPU to configure 8257 whereas the status register is read by CPU to check which channels have reached a terminal count condition and status of update flag.

Mode Set Register: Least significant four bits of mode set register, when set, enable each of the four DMA channels. Most significant four bits allow four different options for the Pin Diagram of 8257. It is normally programmed by the CPU after initializing the DMA address registers and terminal count registers. It is cleared by the RESET input, thus disabling all options, inhibiting all channels, and preventing bus conflicts on power-up.



Status Register: Fig shows the status register format. As said earlier, it indicates which channels have reached a terminal count condition and includes the update flag described previously.



Priority Resolver: It resolves the peripherals requests. It can be programmed to work in two modes, either in fixed mode or rotating priority mode.

THANK YOU For Watching

Success Classes Bijnor

PPI

Today's Target

- ✓ Programmable Peripheral Interface 8255
- ✓Features of 8255 PPI
- ✓PIN Configuration of 8255 PPI
- ✓Block Diagram of 8255 PPI

By Vimal Sir

For PDF join our Telegram Channel from description

8255 - Programmable Peripheral Interface

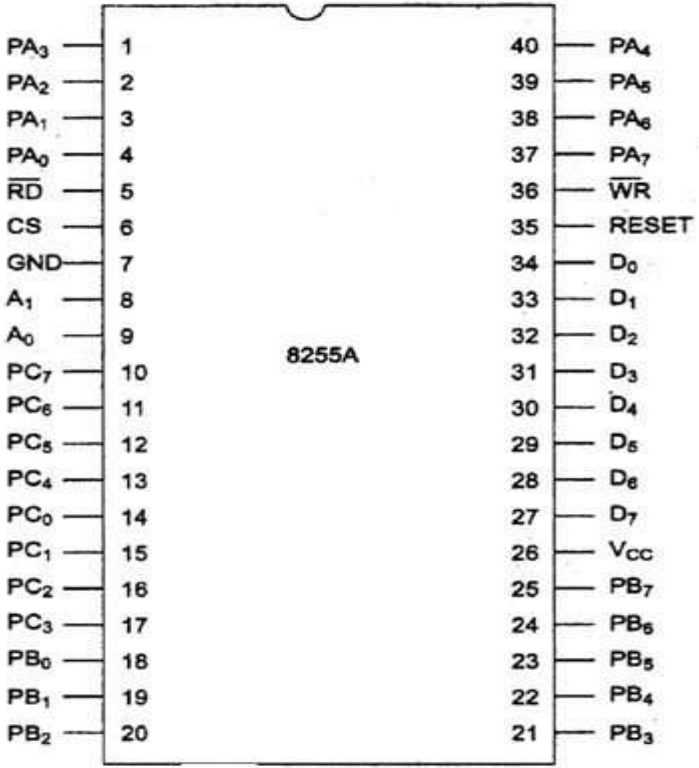
- The 8255 PPI is a programmable peripheral interface device.
- It is a general-purpose programmable parallel I/O device.
- It contains 3 I/O ports which can be programmed in different modes.
- To program the function to all three I/O ports contains a register called control registers. The control register defines the function of each I/O port and in which mode they should operate.
- 8055 PPI is general purpose in nature and provides many facilities for connecting different devices. So it is used frequently in different applications.

Features of 8255 PPI

- It is a programmable parallel I/O device.
- It contains 24 programmable I/O pins arranged as 2-8 bit ports and 2-4 bit ports.
- It has 3, 8-bit ports: Port A, Port B, and Port C, which are arranged in two groups of 12 pins.
- Fully compatible with Intel microprocessor families.
- TTL is compatible.
- Direct bit set/reset capability is available for port C.
- Improved DC driving capability.
- It can operate in 3 Modes:

Mode 0: Simple I/O, Mode 1: Strobed I/O, Mode 2: Strobed bi-directional I/O

PIN Configuration of 8255 PPI



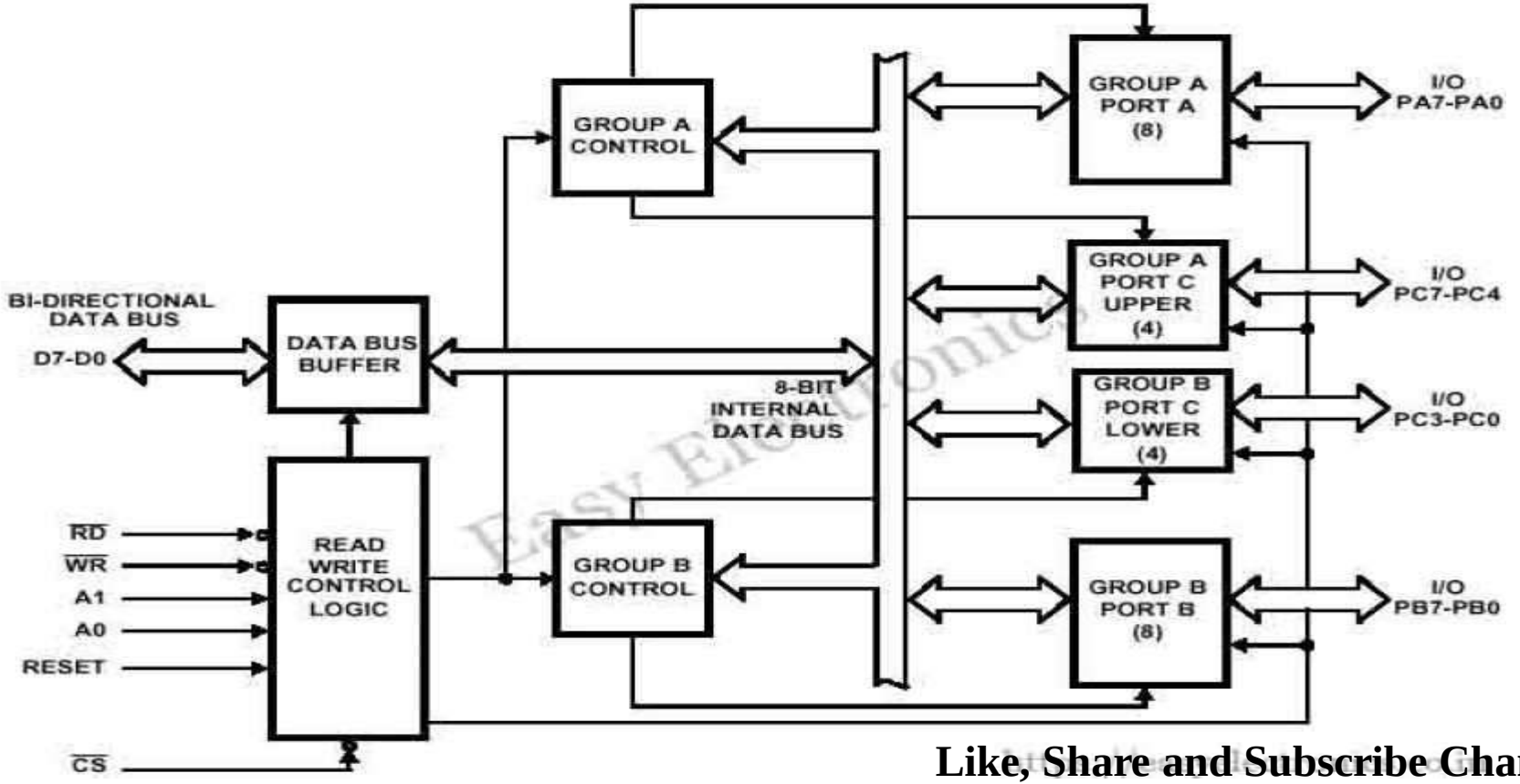
PA_0-PA_7	I/O	Port A pins
PB_0-PB_7	I/O	Port B pins
PC_0-PC_7	I/O	Post C pins
D_0-D_7	I/O	Data pins
RESET	I	Reset input
$\overline{RD}$	I	Read input
$\overline{WR}$	I	Write input
A_0-A_1	I	Address pins
$\overline{CS}$	I	Chip select
VCC-GND	I	+5 V supply ground

Symbol	Name	Function
D_0-D_7	Data Bus	These are 8-bit bi-directional data bus lines, connected to the system data bus for data transfer between CPU and 8255.
$\overline{CS}$	Chip select	These are input, active HIGH address lines used to distinguish different ports of 8255 such as Port A, Port B, Port C, and Control register.
$\overline{RD}$	Read	This is an active low input signal used in coordination with another signal to send data to the CPU through data lines.
$\overline{WR}$	Wite	This is an active low input signal used in coordination with another signal to send data to 8255.
A_0-A_1	Address lines	These are input, active HIGH address lines used to distinguish different ports of 8255 such as Port A, Port B, Port C, Control register.
RESET	Reset	This is an active HIGH input signal used to reset 8255. When 8255 is reset, it clears the control word register and all ports are set to input mode.
PA_0-PA_7	Port A pins 0 to 7	These are 8-bit bidirectional I/O pins used to send data to the peripheral or to read data from the peripheral. The contents are transferred to/from Port A.
PB_0-PB_7	Port B pins 0 to 7	These are 8-bit bidirectional I/O pins used the same as PA_0-PA_7
PC_0-PC_7	Port C pins 0 to 7	These are 8-bit bidirectional I/O pins. These lines are divided into 2 sections i.e. PC_0-PC_3 and PC_4-PC_7. These two sections can be individually used to transfer 4 bits of data from two separate port C sections.

Block Diagram of 8255 PPI

It contains the following blocks

- Data bus buffer
- Read/Write control logic
- Group A and Group B control
- Port A and Port B
- Port C



1. Data Bus Buffer

- The 8-bit bidirectional tristate data bus buffer is used to interface the 8255 internal data bus with the system data bus.
- The direction of the data buffer is decided by read and write control signals.
- When the read is activated, it transmits data to the system data bus.
- When a write is activated, it receives data from the system data bus.

2. Read/Write Control Logic

- This block accepts inputs from the system control bus and addresses bus and performs operations.
- The control signals are $\overline{RD}$ and $\overline{WR}$ and address signals used are A_0 and A_1 and $\overline{CS}$.
- The signls $\overline{RD}$ and $\overline{WR}$ are connected to $\overline{IOR}$, $\overline{IOW}$ or $\overline{MEMR}$, $\overline{MEMW}$.
- A_0 and A_1 of 8085 are directly connected to address lines A_0 and A_1 of 8255.
- $\overline{CS}$ is connected to address chip select decoder.
- The 8255 operation/selection is enabled/disabled by $\overline{CS}$ signal.

3. Group A and Group B Control

- The 8255 I/O ports are divided into 2 sections. Group A(GA) and Group B(GB).
- Group A consist of port A and port C upper.
- Group B consists of port B and port C lower.
- Each group is programmed through software.
- The GA and GB control block receives commands from R/W control logic to accept bit pattern from the CPU.
- GA control will control GA ports and GB control will control GB ports.
- The bit pattern given by the CPU consists of the following information:
- To control the operation of GA and GB
- The mode in which they should be operated.

4. Port A and Port B

- Port A and port B consist of an 8-bit bidirectional data output latch/buffer and an 8-bit data input buffer.
- The function of ports A and B is decided by the control bit pattern available in GA and GB control.
- The function of ports A and B are also independent of the mode of operation.

5. Port C

- Port C consists of an 8-bit bidirectional data output latch/buffer and an 8-bit data input buffer.
- It is divided into 2 sections, port C upper PC_U and port C lower PC_L. These two sections can be programmed and used separately as a 4-bit I/O port.
- It can be used as (i)Simple I/O (ii) handshake signals (iii) status signal inputs.
- For handshake signals and status signals, it is used in coordination with port A and port B.
- The direct but set/reset capability is provided by port C only.

THANK YOU For Watching

Success Classes Bijnor

USART

Today's Target

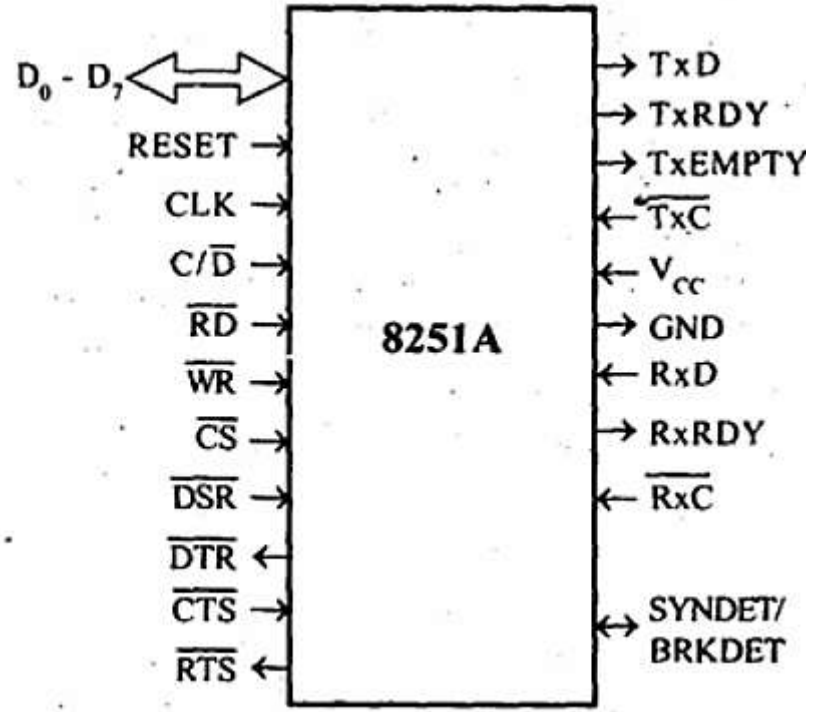
- ✓ USART 8251
- ✓PIN Configuration of 8251 USART
- ✓Block Diagram of 8251 USART

By Vimal Sir

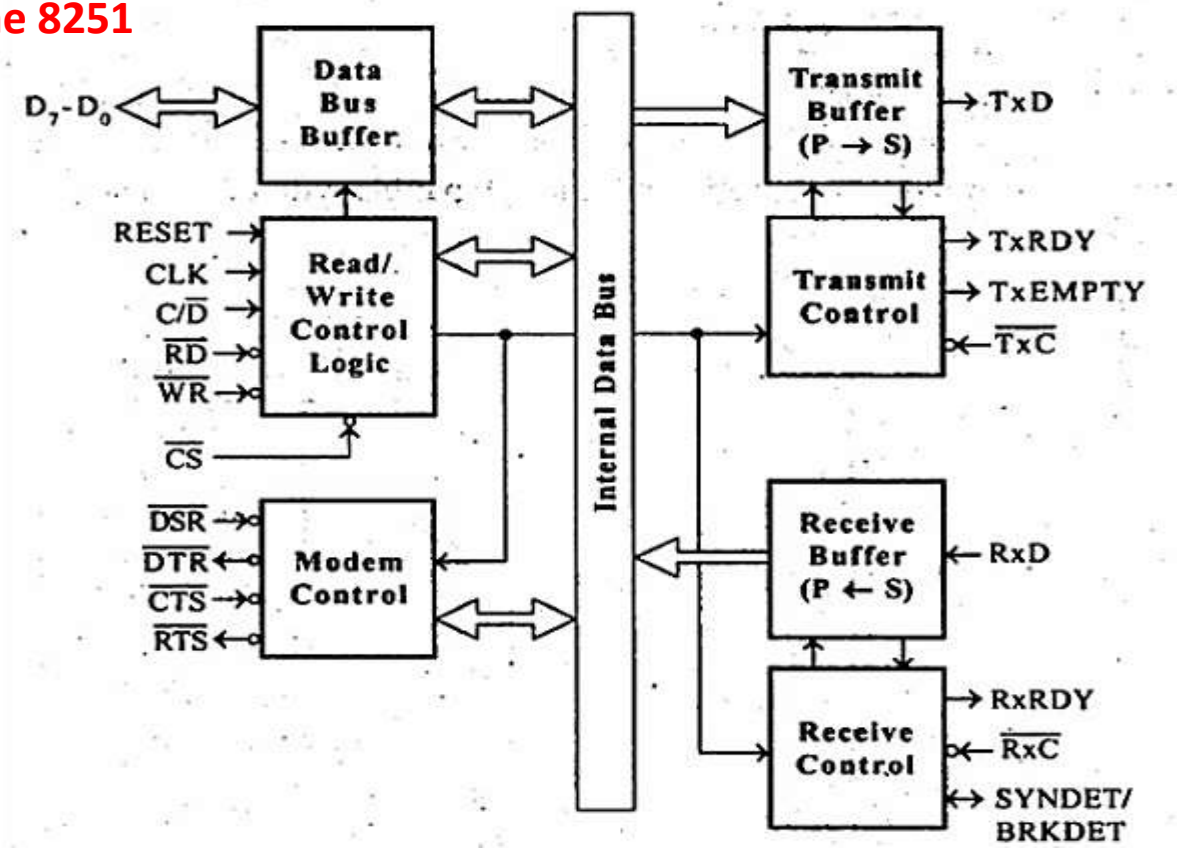
For PDF join our Telegram Channel from description

- **8251 USART**

The 8251 chip is Universal Synchronous Asynchronous Receiver Transmitter (USART). It acts as a mediator between the microprocessor and peripheral devices. It converts serial data to parallel form and vice versa. This chip is 28 pin DIP.



- **Block diagram of the 8251**



Pin	Description
D_0 - D_7	parallel data
C/D	Control register or Data buffer select
RD	Read Control
WR	Write control
CS	Chip Select
CLK	clock pulse
RESET	Reset

TxC	Transmitter Clock
TxD	transmitted data
RxC	Receiver Clock
RxD	Receiver Data
RxRDY	Receiver Ready
TxRDY	Transmitter Ready
DSR	Data Set Ready
DTR	Data Terminal Ready
SYNDET/	Synchronous Detect/
BRKDET	DetectBreak
RTS	Request to send Data
CTS	Clear to send Data
TxEMPTY	Transmitter Empty
V_{cc}	V_{cc} (5V)
GND	Ground(0V)

THANK YOU For Watching

Success Classes Bijnor

PIC

Today's Target

- ✓ PIC 8259
- ✓Features of 8259 PIC
- ✓PIN Configuration of 8259 PIC
- ✓Block Diagram of 8259 PIC

By Vimal Sir

For PDF join our Telegram Channel from description

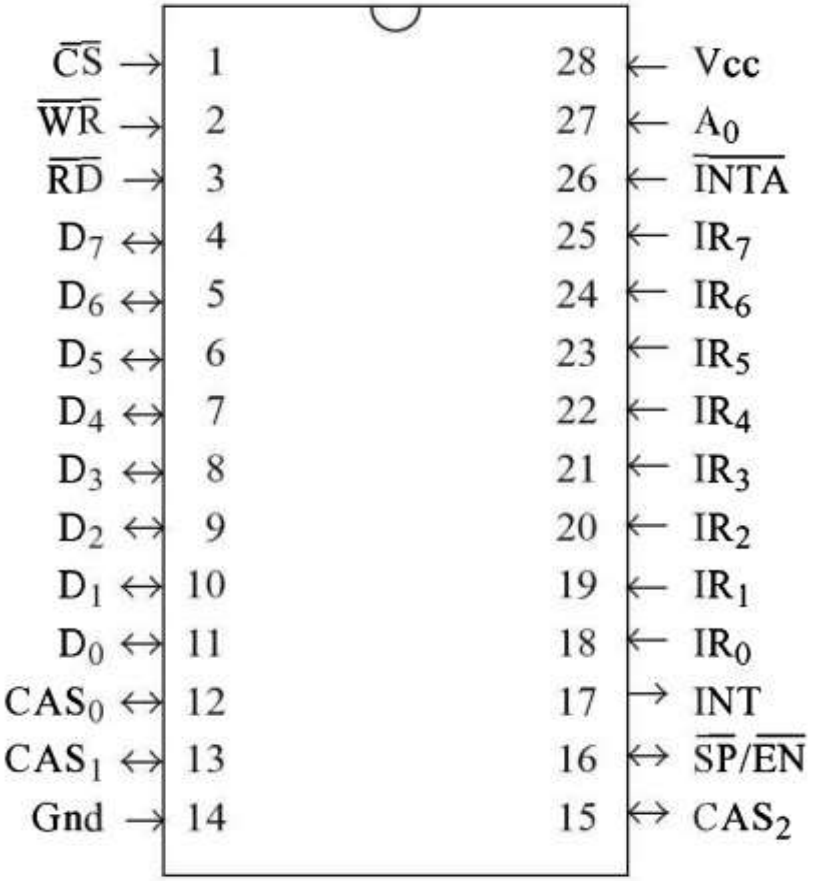
- **PIC 8259**

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. Bu 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 from IR0 to IR7.

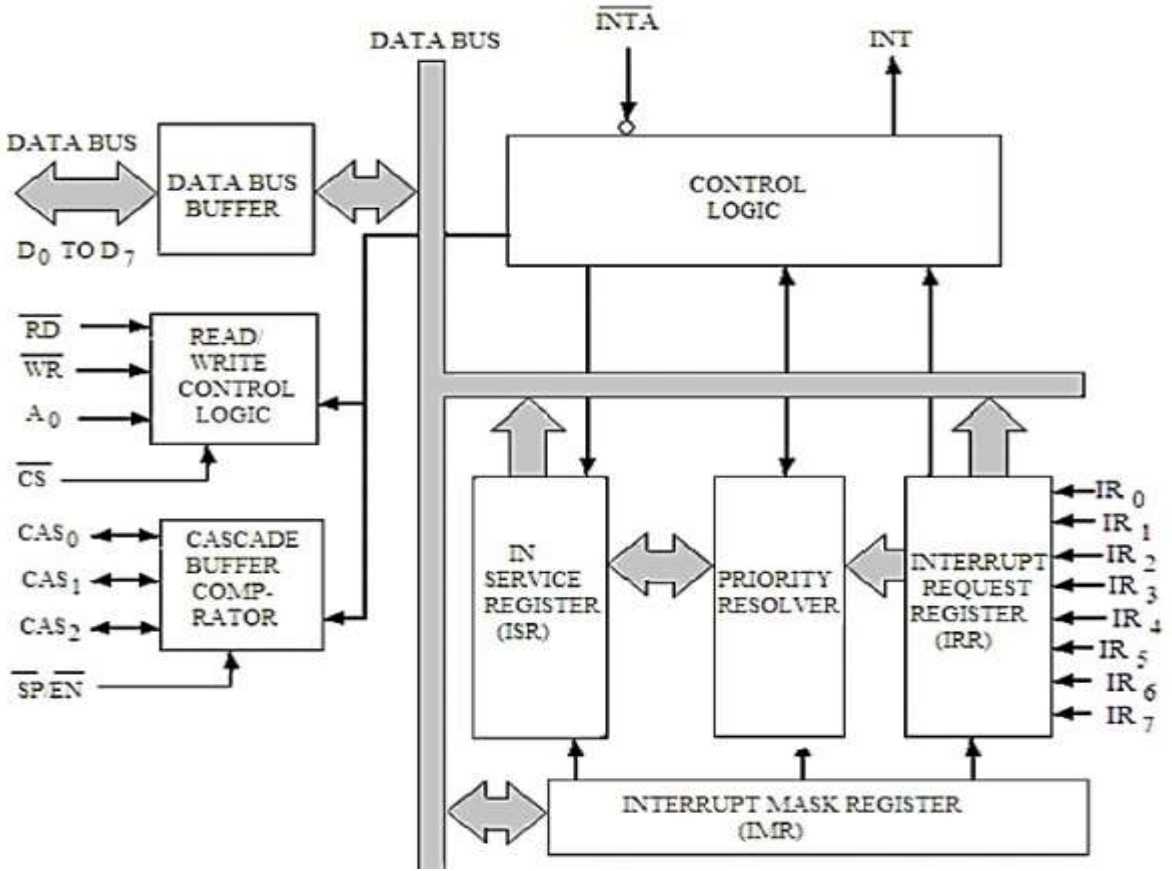
Features of 8259

- 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 bits of Interrupt Request Register.
- By cascading 8259 chips, we can increase interrupts up to 64 interrupt lines
- Clock cycle is not needed

PIN Diagram of 8259



Block Diagram of 8259



Block	Description
Data Bus Buffer	This block is used to communicate between 8259 and 8085/8086 by acting as buffer. It takes the control word from 8085/8086 and send it to the 8259. It transfers the opcode of the selected interrupts and address of ISR to the other connected microprocessor. It can send maximum 8-bit at a time.
R/W Control Logic	This block works when the value of pin CS is 0. This block is used to flow the data depending upon the inputs of RD and WR. These are active low pins for read and write.
Control Logic	It controls the functionality of each block. It has pin called INTR. This is connected to other microprocessors for taking the interrupt request. The INT pin is used to give the output. If 8259 is enabled, and also the interrupt flags of other microprocessors are high then this causes the value of the output INT pin high, and in this way this chip can responds requests made by other microprocessors.

Interrupt Request Register	It stores all interrupt level that are requesting for interrupt service.
Interrupt Service Register	It stores interrupt level that are currently being execute.
Interrupt Mask Register	It stores interrupt level that will be masked, by storing the masking bits of interrupt level.
Priority Resolver	It checks all three registers, and set the priority of the interrupts. Interrupt with the highest priority is set in the ISR register. It also reset the interrupt level which is already been serviced in the IRR.
Cascade Buffer	To increase number of interrupt pin, we can cascade more number of pins, by using cascade buffer. When we are going to increase the interrupt capability, CSA lines are used to control multiple interrupts.

THANK YOU For Watching