

RS 232C Standard:

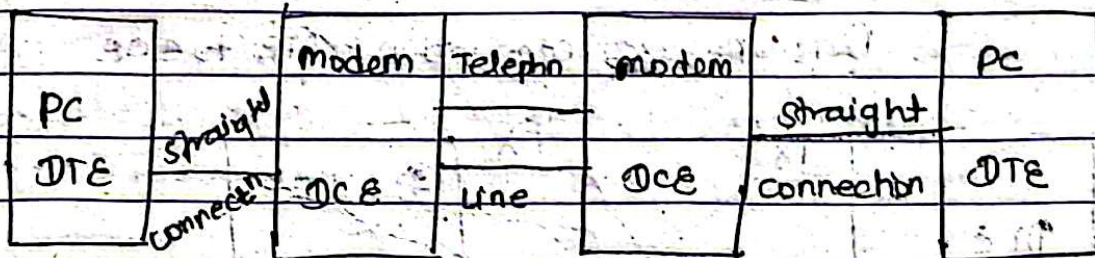
- Developed by Electronic Industry Alliance (EIA) for connection serial devices.
- Used to connect DTE and DCE.
- This defines DTE as a computer and DCE as modem.
- When data are transmitted as voltage, the commonly used standard is known as RS 232C.
- It uses 25 pins (DB-25P) or 9 pins (DE-9P) standard where 9 pins standard does not use all signals i.e. data, control, timing and ground.
- It specifies that DTE connector should be male and DCE connector should be female.

The voltage level for RS 232 are

logic level 0 $\Rightarrow$ +3V to +15V

logic level 1 $\Rightarrow$ -3V to -15V.

- The rate of data transmission is limited to 20K bits/sec and distance of 50ft.



The PC or microcomputer acts like DTE and Modem acts like a DCE. The RS232 standard is defined with respect to DTE and DCE devices. DTE transmits data through pin number 2 while DCE receives data via pin number 2 in RS 232 standard.

3. Direct addressing mode:

In this type of addressing mode, a data is memory address is directly specified in the instruction as a part of it. One of the operand is the direct memory and other operand is the register.

Instruction	Source	Destination
MOV AL, [1234H]	Assume DS = 1000H	Register AL

DS = 1000H

1000H + 1234H

= 11234H

Memory location = 11234H

4. Register indirect mode:

In this addressing mode, the operand field is the indirect address register pair.

Instruction	Source	Destination
MOV CL, [BX]	Assume DS = 1000H	Register CL

BX = 0300H

1000H + 0300H

= 10300H

Memory location = 10300H

10300H

Base plus Index addressing mode

Base plus Index addressing is similar to

indirect addressing mode because it indirectly addresses memory data.

This type of addressing uses one base register (BP or BX) and one index register (DI or SI) to indirectly address memory.

Instruction	Source	Destination
MOV [BX+SI], CL	Register CL	Assume DS = 1000H
		Assume BX = 0300H
		Assume SI = 0200H
		1000H + 0300H + 0200H
		= 10500H
		Memory location = 10500H

Register Relative addressing mode

In this case, the data in a segment of memory are addressed by adding the displacement to the content of base or an index register (BP, BX, DI, SI).

Instruction	Source	Destination
MOV [BX+4], CL	Register CL	Assume DS = 1000H
		Assume BX = 0300H
		1000H + 0300H + 4H
		= 10304H
		Memory location = 10304H

Base Relative plus Index addressing mode:-

The base relative plus index addressing mode is similar to the base plus index addressing mode but it add a displacement to form a memory address.

2. Title Directives:

- The title directive is used to define the title of a program & to print on line 2 of each page of the listing.
 - It is also optional directive.
- Format:

TITLE [TEXT]

TITLE "PROGRAM TO PRINT FACTORIAL NO."

3. Segment directives:

- These 3 directives are used together to define program segments.
- It gives the start of a segment for stack, data and code.

Format:

• stack [size]

• data

-- -- -- Initialize data variables

• code

- A stack segment defines stack storage, a data segment defines data items and a code segment provides executable code.

4. Memory Model Definition:

- The different model tell the assembler how to use segments to provide space and ensure optimum execution speed.

Format:

Model [model name]

- The memory model may be TINY, SMALL, MEDIUM, COMPACT, LARGE, AND HUGE.

5. THE PROC Directives:

- PROC marks the beginning beginning of a procedure or subroutine, while ENDP marks the end.

MY-PROC PROC

; procedure code goes here

MY-PROC ENDP

6. END Directives

- The directive marks the end of the assembly language program.

END: Marks the end of the program.

7. The EQU Directives:

- It can be used to assign a name to constants. It is used for replacing symbolic names.

Format:

MY-CONSTANT EQU 10

Example:

DATA DB 25

DATA EQU DATA

Full address decoding

Potential address decoding

1. All higher address lines are decoded to select the memory or I/O device. Few higher address lines are decoded to select the memory or I/O device.
2. More hardware is required to design decoding logic. Hardware required to design decoding logic is less. Sometimes it can be eliminated.
3. Higher cost for decoding ckt. less cost for decoding ckt.
4. No multiple addresses. It has a advantage of multiple addresses.
5. Used in large system. used in small system.

The 8086 up has several advantages over the 8085 Architecture:

The 8086 has a 16 bit architecture while the 8085 is an 8-bit processor. This means the 8086 can process data and address in 16-bit allowing it to handle larger amount of data more efficiently.

Addressing capability:

The 8086 can address upto 1MB of memory, where as the 8085 can only address 64KB of memory directly. This makes the 8086 suitable for more complex applications that requires large memory addressing.

Instruction set:

The 8086 has a more extensive instruction set compared to 8085. It includes instruction for performing 16 bit arithmetic operations directly which can lead to faster execution of certain tasks.

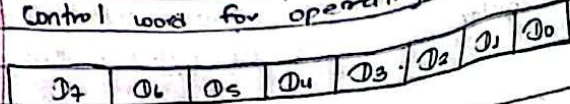
Speed:

The 8086 operates at a higher clock speed as compared to the 8085, which results in faster execution of instructions and better overall performance.

MODES OF OPERATIONS

DATE

Control word for operating mode of 8255



BSR mode

For port
No C effd
on I/O mode

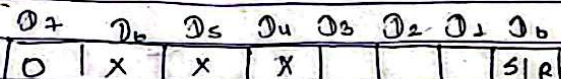
mode 0
Simple I/O
for ports
A, B and
C

mode 1
Handshake
for port A
and port B
Port C bits
are used
for handshake

mode 2
Bidirectional
data bus for
port A
Port B: Either
in mode 0 or
1
Port C bits are
used for hand
shake

1) BSR control word :-

If MSB of control word (D7) is 0, PPI works in BSR mode. In this mode only port C bits are used for set or reset



BSR mode
Not used
0 - Reset
0 - Reset
0 - Reset

PAGE

Port C Bit number

000	= Bit 0
001	= Bit 1
010	= Bit 2
011	= Bit 3
100	= Bit 4
101	= Bit 5
110	= Bit 6
111	= Bit 7

Fig: 8255 A control word format for BSR mode.

If MSB of control word (D7) is 1, PPI works in input-output mode. This is further divided into three modes:

(I/O control word): Hence this is further divided into three modes:

a) Mode 0:
In this mode all the three ports (port A, B, C) can work as simple input function or simple output function. In this mode there is no interrupt or handshaking capability.

b) Mode 1:
In this mode either port A or port B can work as simple output port, and port C bits are used for handshake signals before actual data transmission. It has interrupt handling capacity and input and output are latched.

PAGE

- CPU receives the HOLD request from DMA controller and relinquishes the bus and sends the HOLD Acknowledgement (HOLDA) to DMA controller.
- After receiving the HOLDA, DMA controller acknowledges I/O device (DACK) that the data transfer can be performed and DMA controller takes the charge of the system bus and transfer the data to or from memory.
- When the data transfer is accomplished the DMA raises an interrupt to let know the processor that task of data transfer is finished and the processor can take control over the bus again and start processing where it has left.

The DMA Controller transfers the data in three modes:-

a. Burst Mode:-

In this scheme, the I/O device withdraws the DMA request only after all the data bytes transferred.

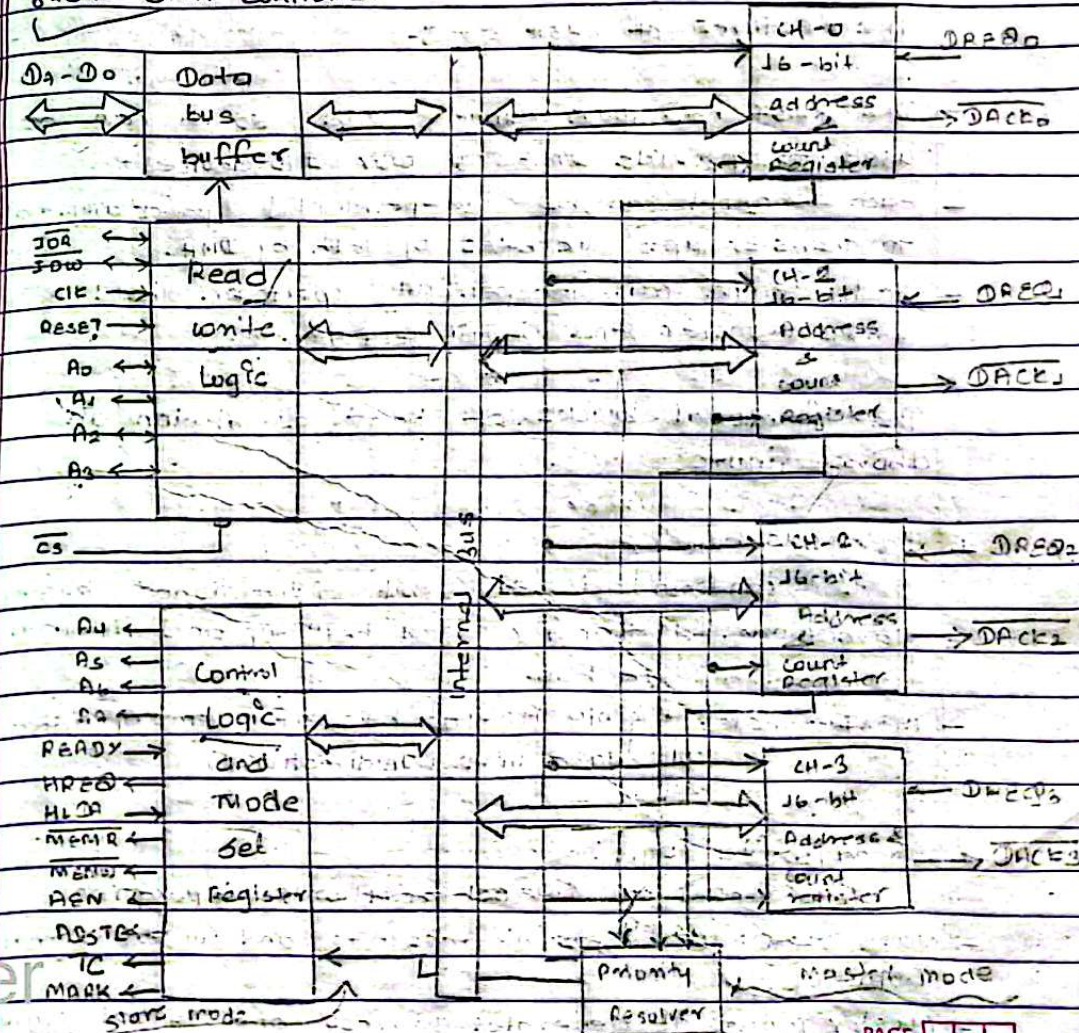
b. Cycle stealing technique (Mode)

In this scheme, the bytes are divided into several parts and after transferring every part, the control of bus is given back to CPU and later stolen back when CPU does not need it.

Transparent mode:

Here, the DMA controller takes the charge of system bus only if the processor does not require the system bus.

8237 - DMA Controller



functional block diagram of DMA controller

Design an interfacing ckt to interface one 4Kb EPROM and two 2Kb RAM memory for 8085 chip. Using 3x8 decoder.

Soln

Since no specific starting address is given.

Let be 8000H.

For EPROM

For EPROM

Base Address = 8000H

End Address = Base Address + (Number of locations in 4Kb EPROM - 1)

$$= 8000H + (4 \times 1024 - 1) \times 16$$

$$= 8000H + 40960$$

$$= 8000H + 0FFFFH = 8FFFFH$$

$$\text{Number of address line} = \frac{\log(4 \times 1024)}{\log(2)}$$

$$= 12$$

A11 ... A0 are connected to internal decoder of EPROM

For RAM 1.

$$\text{Base address} = \text{End address of EPROM} + 0001H$$

$$= 8FFF + 0001H = 9000H$$

$$\text{End address} = 9000H + (2 \times 1024 - 1) \times 16$$

$$= 9000H + 20480$$

$$= 9000H + 0FFFFH$$

$$\text{No. of address line} = \frac{\log(2 \times 1024)}{\log(2)}$$

$$= 11$$

i.e. A10 ... A0 are connected to internal decoder of RAM. For RAM 2

$$\text{Base address} = \text{End address of RAM 1} + 0001H$$

$$= 9FFFH + 0001H = 9800H$$

$$\text{End address} = 9800H + (2 \times 1024 - 1) \times 16$$

$$= 9800H + 20480$$

$$\text{address line} = 11$$

$$= 9FFFH$$

A10 ... A0 are connected to internal decoder

Memory mapping table (3x8 varax)

EPROM

$$\begin{array}{cccccccccccccccc} A_{15} & A_{14} & A_{13} & A_{12} & A_{11} & A_{10} & A_9 & A_8 & A_7 & A_6 & A_5 & A_4 & A_3 & A_2 & A_1 & A_0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array} = 8000H$$

$$= 8FFFFH$$

RAM 1

A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0

$$\begin{array}{cccccccccccccccc} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array} = 9000H$$

$$= 9FFFFH$$

RAM 2

A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0

$$\begin{array}{cccccccccccccccc} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array} = 9800H$$

$$= 9FFFFH$$

Data bus

- The data bus is a group of eight lines used for data flow.
- These lines are bi-directional - data flow in both directions betⁿ the MPU and a memory and peripheral devices.
- The MPU uses the data bus to perform the select function transferring binary information.
- The largest number that can appear on the data bus is 11111111.
- The eight data lines enable the MPU to manipulate 8-bit data ranging from 00H to FFH.

Control bus:-

- The control bus carries synchronization signals and provides timing signals.
- The MPU generates specific control signals for every operation it performs. These signals are used to identify a device type with which the MPU wants to communicate.
- Some control bus signals are
 - memory read
 - memory write
 - I/O read
 - I/O write

Microprocessor and Microcontroller

Microprocessor	Microcontroller
Application It is used where intensive processing is required. It is used in personal computers, laptops, mobiles, video games etc.	It is used where the task is fixed and predefined. It is used in the washing machine, alarm, microwave, cameras etc.
Structure: It has only the CPU in the chip. Other devices like I/O port, memory timer is connected externally.	CPU, memory, I/O port and all other devices are connected on the single chip.
The structure of the μp is flexible. Users can decide the amount of memory, the number of I/O ports and other peripheral devices.	The structure is fixed. Once it is designed the user cannot change the peripheral devices.
Clock speed The clock speed of μp is high. It is in terms of the MHz. It ranges between 1 MHz to 4 MHz.	The clock speed of microcontroller is less. It is in terms of the MHz. It ranges between 1 MHz to 300 MHz.

Highest number

2000

Back

2009

H = 200A

$$(A - 200) = 2$$

1

$$(A - 200) = 1$$

Explain with suitable circuit diagram how read and write operations are performed.

LXJ H1 2000H

MVJ A, 00AH

MVJ G1 0AH

MOV A, M

CMP B, M

JNC SKIP

MOV A, M

SKIP: INX H

DCR C

JNZ Back

MOV A, M

HLT

DATE



M A write
A M read

DATE

- The figure above shows a NMOS static memory cell which consists of six NMOS transistors. (N-channel metal-oxide semiconductor).
- The transistor Q_A is called the access transistor and is used to access the data during the read and write operation.
- The transistors Q_3 and Q_4 are pull up load transistors. Their cross connection with Q_2 and Q_1 respectively forms a flip-flop.
- The D & D' are data lines. There is external driving circuitry which forces D & D' to opposite values. When reading the bit from the cell the D and D' serve as output lines while writing a bit on the cell they serve as forced input.
- Before reading or writing on cell, the cell must be selected and the select line is used for the purpose.

Working on cell:-

- The access transistor Q_A 's are turned on via the select line (high).
- The data lines are forced into a state with D high and D' low.
- The MOSFET Q_1 will be turned on and Q_2 off.
- When forcing Q signals are removed Q_1 will continue hold D' low (i.e., Q_1 will be on) and will keep Q_2 off and D output high.
- The forced state the cell is thus self-sustaining and stable.

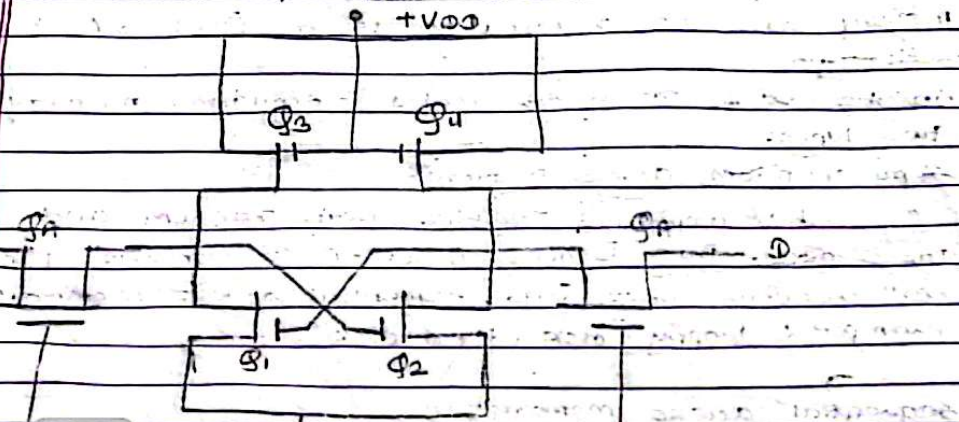


fig. static 1T1R1C1 cell

PAGE

PAGE

by Read only memory (ROM)

ROM is a non volatile memory and can be read only. It is used to store data and programs that are not be altered. ROM contains a permanent pattern of data that cannot be changed. ROM can only be read.

Masked ROM

It is programmed by a masking operation performed on the chip during the manufacturing process. The content of masked ROMs are permanent and cannot be changed by user.

Programmable ROM

It can be programmed by user by means of proper equipments. The content of this type of memory cannot be changed unless it is erased by special device.

Erasable PROM (EPROM)

It can be programmed by using PROM programmer. Its memory can be erased by staticing the chip to a ultraviolet light. EPROMS could be remove from the micro computer for programming.

IV. EEPROM (Electrically erasable PROM)

EEPROMs can be erased and programmed without removing the chip from the microcomputer system. The content of EEPROMs can be erased electrically.

V. Flash memory

It is variation of EEPROM. The difference is that EEPROM can be erased at a register level but flash memory must be erased either in its entirety or at the sector level.

Secondary Memory

The device that provide the backup storage are called secondary memory. These devices are used to hold large data files and large programs. The secondary memory are also referred to as auxiliary or backup storage.

On the basis of access modes, secondary memory are two types.

• Semi random access memory

These memories combine both random and sequential access mode. This is done in order to achieve a compromise between random access and sequential access memories.

Example: Floppy disk, Hard disk

• Sequential access memory

In this mode, the access method is strictly sequential. The memory access time depends upon the location of data. It is also called serial access memories.

Eg: Magnetic tapes

Primary / main memory:

- It is the storage or memory used by μp to execute all the programs.
- The μp can directly access only those instructions or data that are stored in primary memory.
- Hence, All data and program must be within primary prior or its execution.
- Primary memory is much larger than processor memory that is included in the μp chip.

The primary memory can be divided into two parts:

A. Read / write Memory (Random Access Memory)

- It is a volatile memory and is used to store program and data for immediate use of the processor.
- μp can read and write into this memory.
- This memory is used for information that are likely to be altered such as writing program or receiving data.

RAM are basically two types.

- Static RAM (SRAM)
- Dynamic RAM (DRAM)

SRAM

- This memory is made up of flip-flops and it stores bit as voltage.
- These memory are commonly used in cache memory.
- It is more expensive & power consuming than DRAM.
- It is available in smaller storage capacity of few MB.
- SRAM are low density devices.
- It does not have a refreshing unit.
- SRAM is used in high performance application.

DRAM

- This memory is made up of MOS transistors, gates and stores the bit as charge.
- These memory used in main memory.
- It is cheaper and low power consumption than SRAM.
- It is usually available in large storage capacity of few GB.
- DRAM are high density devices.
- It has refreshing unit.
- DRAM is used in memory purpose application.

It is strictly sequential on the location of data.

RS 422 STANDARD

DATE

- Developed by Electronic Alliance (EIA) for connection serial devices.
- RS 422 is able to provide data rates upto 10Mbps at distance upto 5000 feet.
- It is a newer standard for serial data transfer.
- uses differential signalling.
- Full duplex communication.
- used as an improvement over the RS 232.
- No connector defined.
- Signal level -6V to +6V
- using Reduced data rates, RS 422 is able to transmit data over distances of 4000 feet
- Physical media - Twisted pair of wire.
- RS 422 is a balanced four wire system.
- One device is known as data terminal equipment (DTE) & another device known as DCE.

	Rx+	Tx+	
DTE	Rx-	Tx-	DCE
Computer	Tx+	Rx+	device
	Tx-	Rx-	

Fig: Typical RS 422 wiring.

Advantage:

DATE

Higher data rates

longer cable length

Noise rejection.

Disadvantage

only one transmitter for most application

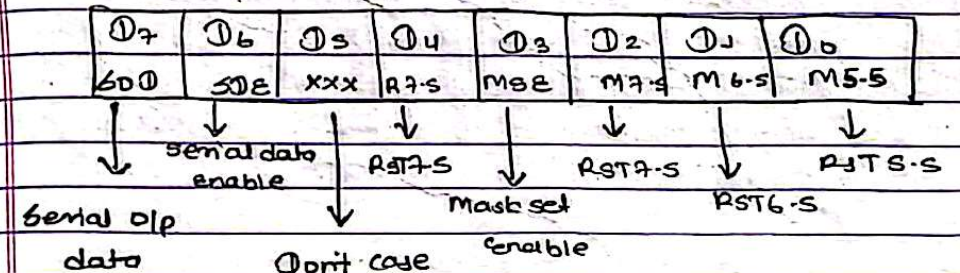
Unidirectional

Not as common

SIM and RIM INSTRUCTION

SIM (Set Interrupt Mask)

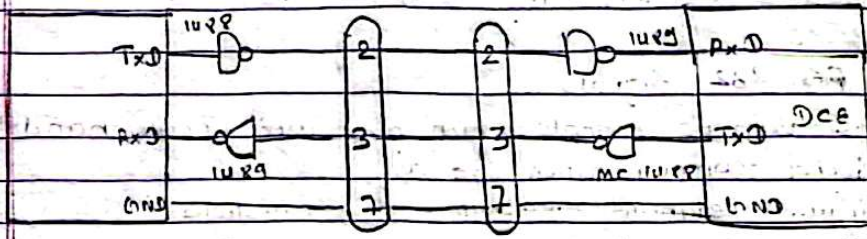
This instruction is used to set interrupt mask and send serial output. It transfer the content of accumulator to interrupt control logic and serial I/O ports. Thus it is necessary to load appropriate content in the accumulator before execution of SIM instruction. It is of 1 byte instruction and no flags are affected. The format of SIM instruction is shown in figure below.



- 8 Data carrier detect DCD: Generally used to disable data reception
- 9 Data terminal ready DTR: Output: Generally used to indicate that DTE is ready.

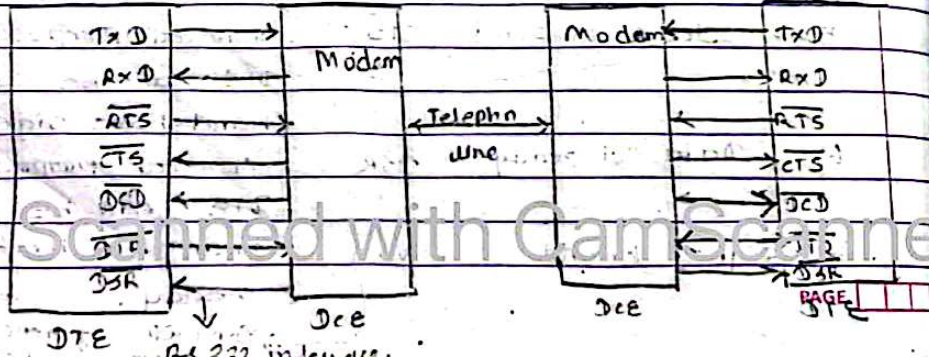
How can you transfer data

Connection of DTE and DCE through RS-232C interface.



How can you transfer the data betⁿ two computer without using modem? Draw and explain the interface with handshake signal.

Fig: Digital data transmission using modem



Null modem without using modem:-

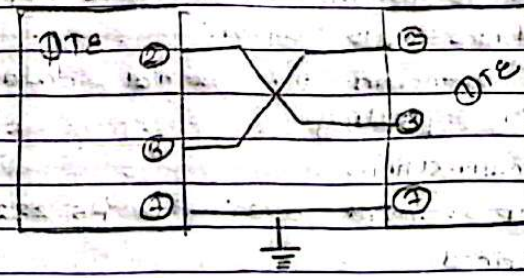


Fig: RS-232C connection DTE to DTE

while the direct connection betⁿ a PC and PC without modem is crossover cable. The pin number 2 is connected to a pin number 3 because DTE sends data through pin 2 and DTE receives data through pin number 3. This connection is known as Null modem connection.

Hence the connection betⁿ a PC (DTE) and modem (DCE) is a straight through cable.

While the direct connection between a PC and PC without modem is crossover cable. The pin number 2 is connected to a pin number 3 because DTE sends data through pin 2 and DTE receives data through pin number 3. This connection is also known as Null modem connection. These two connections between DTE and DCE and DTE and DTE is shown below.

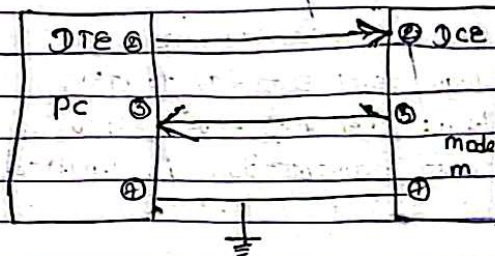
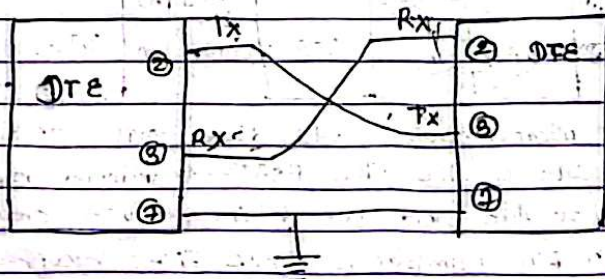


Fig: RS 232 connection: DTE to DCE



RS 232 connection: DTE to DTE

Advantage of RS 232

- It is widely used for point to point connection between DCE and DTE devices.
- low cost
- Due to simplicity, the interface RS 232 is supported in many devices.
- It is free from noise.

Disadvantage

- Bandwidth is low
- cable length is limited
- power consumption increases

RS 232 Signals

The various signals which are used with handshake data communication are as

Pin no.	signals	Acronym	Functions
2	Transmitted data	TxD	Output transmits data from DTE to DCE
3	Received data	RxD	Input DTE receives data from DCE
4	Request to send	RTS	General purpose output from DTE
5	Clear to send	CTS	General purpose input to DTE; can be used as a handshake signal
6	Data set ready	DSR	General purpose input to DTE; can be used to indicate that DCE is ready.
7	Signal ground	GND	Common reference bet ⁿ

→ **AX:**

It is used as 16 bit accumulator in 16-bit operations and AL is used as an accumulator in 8 bit operation.

→ **BX:**

It is also called the base register can be used as memory pointer in data segment.

→ **CX:**

It is also called counter register, can be used as a counter in string manipulation and shift rotate instructions.

→ **DX:** It is a data register can be used to contain the I/O port address for I/O instructions.→ **Stack pointer Register:**

The stack pointer (SP) register contains the 16-bit offset from the start of the segment to the memory location where word uses most recently stored on stack.

→ **Other pointer and segment register:**

The 8086 also contains a 16-bit source index (SI) register, base pointer (BP) register and instruction pointer (IP) registers. These three registers can be mainly used for the temporary instructions.

16 bit data just like a general purpose register.

→ **Flag Register:**

The Flag Register also known as a status register is a 16 bit register that contains nine flags out of which five are like 8085 flags and remaining four are new flags. These flags are divided into two categories.

• **Conditional flag**• **Control flags**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OF	DF	IF	OF	DF	IF	OF	DF	IF	OF	DF	IF	OF	DF	IF	CF

Direction flag: used in string operation. If set string bytes accessed from higher mem address to lower mem address. when it is reset, the string bytes are accessed from lower mem addresses to higher mem address.

Interrupt flag: Set of maskable interrupt is enabled and if maskable interrupt is disabled it is reset.

Trap Flag: - is used to single step control. If set the program can be run in single step mode.

Control Flag - Overflow flag: - 80 to 7F

i) code segment:-

The CS register is used for the addressing a memory location in the code segment of the memory, where executable program is stored.

ii) Data segment (DS):-

The DS contains most data used by program. data are accessed in the data segment by an offset address or the content of other register that holds the offset address.

iii) Stack segment

Stack segment defines the area of memory used for the stack. SS also defines a section of memory to store address and data while a sub-program executes.

iv) Extra segment:-

ES is an additional data segment that is used by some of the string to hold the extra destination data.

- Instruction pointer:-

IP is a 16-bit register pointer to next instruction to be executed.

2. Execution Unit (EU)

The EU tells the BIU where to fetch instructions or data from, decode instructions and execute instructions.

The EU contains the ALU, flags and general purpose registers and is responsible for carrying out all arithmetic and logical operations.

b) Control circuitry: Instruction decoder and ALU

The EU contains control circuitry which directs internal operations and controls the flow of data inside the CPU.

The instruction decoder translates or decodes the instructions which are fetched from the memory.

After decoding it places the instruction in a series to perform the required tasks by the EU.

The ALU is a 16-bit unit which performs all the arithmetic operations also does logical operations.

c) General purpose registers:-

The EU has 8 general purpose registers labeled as AH, AL, BH, BL, CH, CL and DH, DL.

These registers can be used individually for temporary storage of 8-bit data.

The certain pairs of general purpose registers can be used to store 16-bit data words.

The possible pairs and corresponding names are as follows:
AH, AL = AX (accumulator) CH, CL = CX (counter)
BH, BL = BX (base register) DH, DL = DX (data)

19 20 21 22 23 24 25 26 27 28 29 30

4 5 6 7

18/06/21

DATE

Instruction	source	destination
MOV [BX+SI+05]	Register CX	Assume DS = 1000H
1 CL		Assume BX = 0300H
		Assume SI = 0200H
		1000H + 0300H + 0200H
		+ 05H
		10505H
		memory location
		10505H

String addressing mode

This mode uses index registers, where DI is used to point the first byte or word of the source string and SI is used to point to first byte or word of destination string when string instruction is executed.

eg: MOV SI, MOV SB, MOV SW.

8086 Block diagram description

a) Bus interface unit (BIU)

The BIU send out address, fetches instruction from memory, reads data from ports and memory and write data from port of memory. In other word BIU handle all transfers of data and addresses on the buses for execution unit. It has mainly two parts: instruction queue and segment register.

a) Instruction queue

The BIU can store upto 6 bytes of instructions with FIFO manner in a register called queue. When EU is ready for next instruction, it simply reads the instruction from the queue in the BIU.

b) Segment register

The BIU contains a dedicated address, which is used to produce 20 bit address.

→ The 16-bit four segment registers are used to hold upper 16 bits of the starting address of four memory segments.

→ The four 16-bit segment registers in BIU are code segment, data segment, stack segment, extra segment (es).

PAGE

PAGE

RESET OUT	1	40	VCC
Y2	2	39	HOLD
		38	HOLD
		37	CLKOW
		36	RESET IN
		35	READY
		34	IO/M
		33	S1
		32	S2
		31	S3
		30	S4
		29	S5
		28	S6
		27	S7
		26	S8
		25	S9
		24	S10
		23	S11
		22	S12
		21	S13
		20	S14
		19	S15
		18	S16
		17	S17
		16	S18
		15	S19
		14	S20
		13	S21
		12	S22
		11	S23
		10	S24
		9	S25
		8	S26
		7	S27
		6	S28
		5	S29
		4	S30
		3	S31
		2	S32
		1	S33
		0	S34

8085
A

HOLD: This indicates the acknowledgement for the HOLD request.

HOLD: This indicates peripheral controller requesting the bus.

READY: It is used to delay the up read and write cycles until a slow responding peripheral is ready to send or accept data. We up enter into

WAIT: WAIT state while the READY pin is disabled.

INTA: It is a maskable interrupt. It has the lowest priority. It is also level triggered.

SIO: It takes 1 bit i/p from serial port of 8085.

AIM: AIM instruction is used to transfer the bit

* Addressing Modes of 8086 *

The way of specifying the operand in an instruction is called an Addressing mode.

Register addressing mode:-

It is the most common form of data addressing. For this mode, a register may contain source operand, destination operand or both.

Instruction	Source	destination
MOV AX, BX	Register BX	Register AX

Immediate Addressing mode:

In this mode, the data is directly defined in the operand field. This mode contains a constant value or an expression.

Instruction	Source	destination
MOV CH, 34H	Data 34H	Register AX

Uses / Importance

- ↳ DAA is commonly used in application where arithmetic operation involving BCD numbers are required.
- ↳ It simplifies the process of performing arithmetic operations on BCD numbers by automatically adjusting the result into BCD format.
- ↳ DAA helps in reducing the complexity of software routines for BCD arithmetic by handling the conversion of binary results to BCD automatically.

Example:

WAP to add two 4-digit BCD number.

LXI H, 2431H

LXI B, 3040H

MOV A, L

ADD C

DAA

MOV C, A

MOV A, H

ADC B

MOV B, A

HLT

PIN Configuration of 8085

Explain the function of following pins in 8085 uP: $\overline{RD}$, $\overline{WR}$, RESET, READY and INTR.

The uP is capable of performing various computing functions and making decisions to change the sequence of program execution. 8085 is a 40-pin IC, DIP package.

1. Power supply and clock signals
2. Address bus
3. Multiplexed address/data bus
4. Control and status signals.

8. The STARTUP AND EXIT Directive:-

STARTUP generates the instruction to initialize the segment registers.

• EXIT generates the INT21H junction which is instruction for exiting the program.

9. ASSUME directives

a) The directive is used to specify the default segment registers that point to different segments.

ASSUME CS: CODE, DS: DATA

10. PROCESSOR directive

Processor directive is used to notify the assembler that the instructions or features introduced by the other processors are used in the program.

* Decimal Adjustment Accumulator *

The DAA instruction in the Intel 8085 μ p is used to adjust the result of an arithmetic operation in the accumulator register to form a correct binary coded decimal (BCD) representation in the accumulator.

Operation:

After an arithmetic operation involving addition or subtraction, the DAA instruction is used to adjust the content of the accumulator to hold a valid BCD representation of the result.

It adjusts the accumulator by adding or subtracting a correction factor, if necessary, to ensure that each nibble (4 bits) in the accumulator holds a valid BCD digit (0-9).

The DAA instruction operates on the content of the accumulator directly.

If the lower nibble (bit 0-3) of the accumulator contains a value that is greater than 9 or if the auxiliary carry flag is set, the instruction adds 6 to the lower nibble.

After the adjustment, the accumulator holds the correct BCD representation of the result.

Eg: MOV A, 78H ; A = 78

MOV B, 42H ; B = 42

ADD B ; A = A + B = BA

DAA ; A = 20, CY = 1

Instructions written in assembly language to computer readable machine code.

DATE

One pass and Two pass Assemblers

One pass Assembler	Two pass Assemblers
1. It perform translation in one pass only.	It perform translation in two pass.
2. Intermediate code not generated.	Generation of intermediate code.
3. Forward referencing is handled by back patching	After pass one, all symbols and literals are getting address.
4. Back patching is handled by TII (Table of incomplete instruction)	No need of back patching.
5. More memory required as compare to two pass assembler	less memory required compare to single pass assembler.
6. It is faster than two pass assembler.	It is slower than single pass assembler.
7. Data structures used: Symbol table, literal table, local Table & TII	Data structures used! Symbol table, literal table, local Table.

PAGE

Assembler Directives

DATE

What are assembler directives? Explain all the assembler's directives of 8086 up.

→ Assembler directives (also known as pseudo-ops) are commands or instructions used in assembly language programming to provide instruction to the assembler itself rather than generating machine code. They serve various purposes such as defining constants, allocating memory, controlling the assembly process and specifying program structure.

- Assembler directives are +
1. Page directives: direction to the assembler which indicate how an op. or section of the program is to be processed.
 - Title directive
 - Segment directive
 - Memory model definition
 - The PROC directive
 - End directive
 - The EQU directive
 - The STARTUP And EXIT directive
 - ASSUME directive
 - Processor directives

1. PAGE directives:
 - The page directive helps to control the format of a listing of an assembled program.
 - It is optional directives.
 - format.

PAGE [length], [width]

Page [50], [80]

PAGE

Differentiate betⁿ macros and procedure.

Macros	Procedure
A macros is an instruction sequence that appears repeatedly in a program assigned with a specific name.	Procedure is a key concept for modular programming the essential way to reduce complexity.
When a macro is invoked the corresponding code is inserted into a source. Thus multiple copies of the same code exist in the memory.	Only one copy exist in memory. Thus, memory consumed is less.
There is no execution overhead because there are no additional call & return instruction.	Execution time overhead is present because of the call and return instruction.
It expanded during pre-processing.	It executed at runtime.
It dependent on pre-processor.	It dependent on runtime environment.

If macro is used for the small set of instructions. A procedure is used for large set of instructions.

We require more memory in the case of macro. We require less memory in the case of procedure.

In the case of macro, the machine code is constructed several lines. In case of a procedure, the machine code is created only once.

Format:

MACRO [Parameter list];

END [Instructions]

END

IVT (Interrupt Vector Table)

DATE [] [] [] [] [] []

Where the

- An interrupt vector is a pointer to where the ISR is stored in memory.
- All the interrupts are mapped onto a memory area and are called the IVT.
- IVT is 1024 bytes sized table that contains address of interrupts.
- The IVT is usually located in memory page 00. (0000H - 00FFFH)
- The purpose of the IVT is to hold the vector that redirect the CPU to the right place when an interrupt arrives.
- Structured: IVT is typically located at a fixed memory address in the system. Each entry in the table corresponds to a specific interrupt service.

3FA

DATE

--	--

--	--

--	--	--	--

[illegible]

Fig: JVT structure.

DATE

Construction: Not easy to construct.	Simpler in construction
stored data do not change with time.	Stored data changes with time.
Material: It is made up of with transistors and flip flop.	It is made up with capacitor and few transistor.
capacity: It is available in smaller storage capacity of few mp.	It is usually available in large storage capacity of few gb.
It stores data in the form of voltage.	It stores data in the form of charges.
Refreshing circuit is not implemented.	Refreshing circuit is implemented.
Example: cache memory	DDR1, DDR2, DDR3 etc.

PAGE

DATE

6.1

Serial and parallel data communication

Serial data communication

Parallel data communication

The data bits are transmitted serially over a common communication link one after the other.

The various data bits are simultaneous transmitted using multiple communication links between sender and receiver.



Fig: serial communication

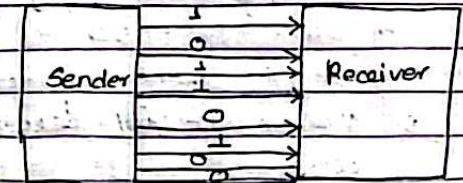


Fig: parallel communication

Its data transmission speed is slow.

It is comparatively fast.

No. of communication link used signal is single.

multiple.

Cost: Low

High

Crosstalk: Not present

present

Mode of transmission: full duplex

Half duplex

Suitable for: long distance

short distance

Circuit used: simple

relatively complex

System up-gradation: easy

Quite difficult

High Frequency: More efficient operation

less efficient

PAGE

ii) Data Bus Buffer

- This block helps in interfacing the internal data bus of 8051 to the system data bus.
- The data transmission is possible between 8051 and CPU by the data bus buffer block.

iii) Read/Write Control Logic:-

- It is a control block for overall device. It controls overall working by selecting the operation to be done.
- The operation selection depends upon input signal as:-

$\overline{CS}$	$\overline{C/D}$	$\overline{RD}$	$\overline{WR}$	Operation
0	0	0	1	CPU $\xleftarrow{\text{data}}$ 8051
0	0	1	0	CPU $\xrightarrow{\text{data}}$ 8051
0	1	0	1	CPU $\xleftarrow{\text{status word}}$ 8051
0	1	1	0	CPU $\xrightarrow{\text{control word}}$ 8051
1	X	X	X	Invalid

When the RESET pin is high, it forces 8051 into the Idle mode.

iii) Modem Control (Modulator/Demodulator):-

- A device converts analog signal to digital signals and vice-versa and helps the computers to communicate over telephone lines or cable wires.

iv) Transmit Buffer:-

This block is used for parallel to serial converter that receives a parallel byte for conversion into a serial signal and further transmission onto the common channel.

v) Receiver Section

- Receives serial data on RXD.
- Converts serial into parallel.
- Transfer data to microprocessor unit.
- These data transmission can be done for synchronous and asynchronous data transfer.

SRAM (Static RAM) and DRAM (Dynamic RAM)

SRAM (Static RAM)	DRAM (Dynamic RAM)
Usage: Cache Memory	Main memory
Speed: Very fast	Fast
Cost: Costly	Cheaper than SRAM
Density: less packing density	Higher packing density
Power: Consume more power	Consume less power than SRAM
Maintenance: No maintenance	Maintenance needed

Mode 2

- Bidirectional I/O port with handshaking.
- Only port A can operate in mode 2.
- The 8-bit data port (port A) can be either input or output. Both I/P and O/P are latched.
- The 5 bits from port C are used to provide control and status for 8-bit data port.
- It also has interrupt handling capacity.

8051 - USART (Universal Synchronous Asynchronous Receiver Transmitted)

- The 8051 is a programmable serial communication interface chip designed for synchronous and asynchronous serial data communication.
- As a peripheral device of a microcontroller computer system, it receives parallel data from the CPU and transmits serial data after conversion.
- This device also receives serial data from the outside and transmits parallel data to the CPU after conversion.

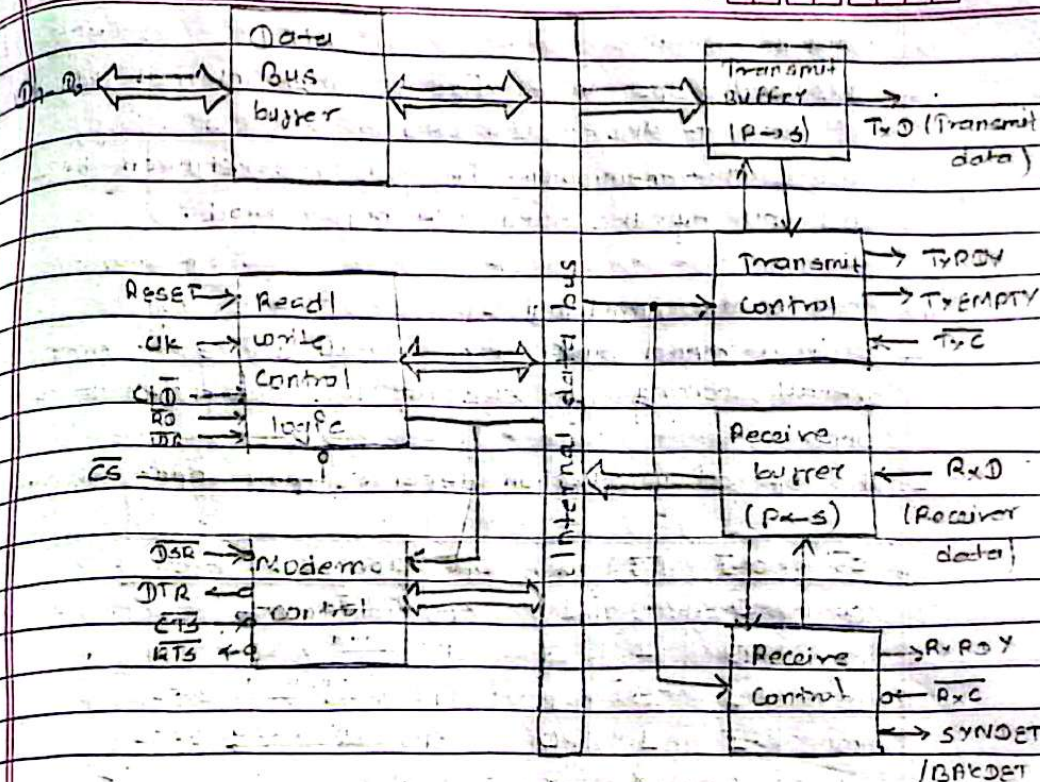


Fig. functional block diagram of 8051A - USART

The functional block diagram of 8051A consists of five sections. They are:

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

The 8055A has the following main blocks

Data bus buffer:

- The 3-state bidirectional 8-bit buffer is used to interface the 8055A to the system data bus.
- Data is transmitted or received by the buffer upon execution of input or output instruction by the CPU.
- Control words and status information are also transferred through the data bus buffer.

Read/Write 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.
- Control signals of 8055A are
 - a) Read ($\overline{RD}$): This control signal enables the read operation. When the signal is low, the μP reads the data from the selected I/O port of the 8055A.

b) Write ($\overline{WR}$): This control signal enables the write operation. When this signal goes low, the μP writes data or control word into a selected I/O port or control register of 8055A.

c) Chip select ($\overline{CS}$): This is active low $\overline{P/P}$ signal.

This now enables the chip and enables the communication between the 8055A and CPU.

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

es) A_0 and A_1 : This input signals work with $\overline{RD}$, $\overline{WR}$, and one of the control signal.

CS	A_1	A_0	{(Result) selection}
0	0	0	Port A
0	0	1	Port B
0	1	0	Port C
0	1	1	Control Register
1	X	X	No Selection

iii) Group A and Group B controls:

These two groups accept 'commands' from the read/write control logic, receives control word from the internal data bus, and issues the proper commands to its associated ports.

- Control Group A $\rightarrow$ Port A and Port C upper (C_7-C_4)
- Control Group B $\rightarrow$ Port B and Port C lower (C_3-C_0)

- It works in two modes i.e. master mode & slave mode.
- It is an I/O to the μp (slave mode) and it is a data transfer processor to peripherals such as floppy disk (master mode).

⑥ 8255 - PPI (Programmable Peripherals Interface) :-

- 8255 is a general purpose programmable I/O device designed to interface the μp with its outside world such as ADC, DAC, keyboard etc.
- It is designed to increase I/O interfacing capability of μp .
- We can program it according to the given config.
- It can be used with almost any μp .
- It has 24 pins that can be grouped in two 8-bits parallel ports: A and B, with the remaining 8-bits into port C.
- Port C is further divided into 4-bits ports i.e. port C lower and port C upper.
- 8255 functions in following modes:
 - 1) I/O modes [mode 0, mode 1, mode 2]
 - 2) BSR mode

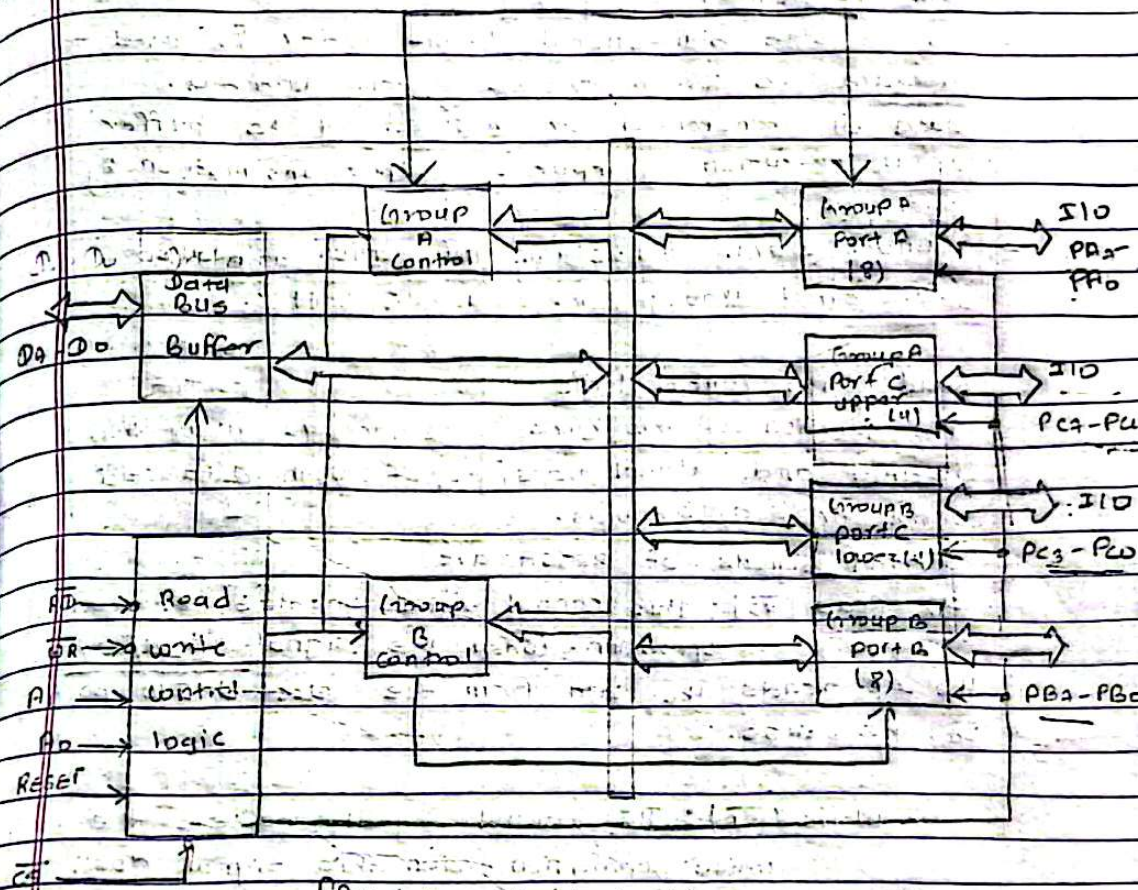


Fig: Internal block diagram of 8255A

It is a device to transfer the data directly between I/O device and memory without through the CPU. so it performs a high speed data transfer between memory and I/O device.

The features of 8237 are:-

- The 8237 has four channels and so it can be used to provide DMA to four I/O devices.
- Each channel can be independently programmable to transfer upto 65536 bytes of data by DMA.
- Each channel can independently perform read transfer, write transfer and verify register.

The functional blocks of 8237 as shown in above figure:

1. Data bus buffer:-

It contains tristate, 8-bit bidirectional buffer.

- Slave mode: It transfers data between internal data bus.
- Master mode: the 16 bits of memory address on data lines (unidirectional)

2. Read/Write logic:

It controls all internal read/write operation.

- Slave mode: It accepts address bits and control signals from CPU.

- Master mode: It generates address and control signals.

III. Control logic & Mode set registers:-

- It generates address and control signals.
- Master mode: It controls the sequence of DMA operation during all DMA cycles.
- Slave mode: It is disabled.

IV. Priority Resolves:

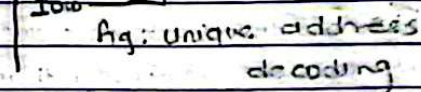
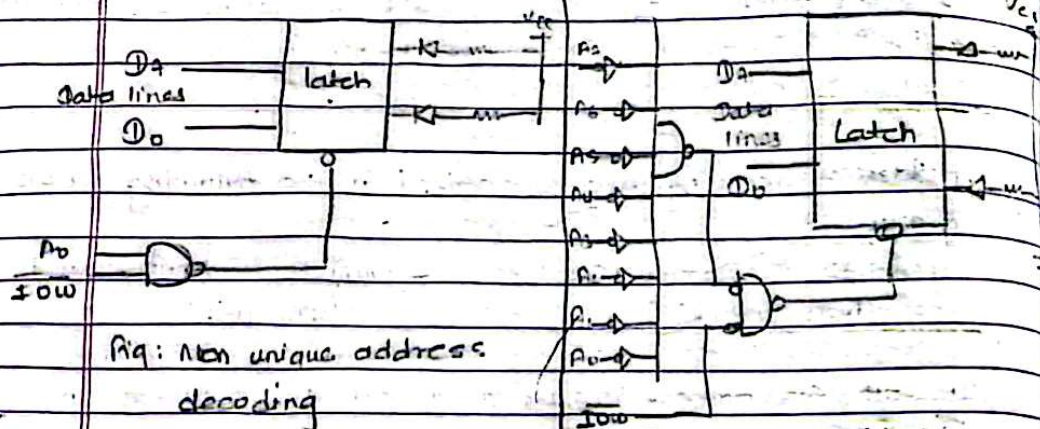
- Priority of each of the 4 channels can be set in fixed priority and rotating priority.

Operation of DMA Controller:-

- Each channel of 8237 has two programmable 16-bit registers named as address register and count register.
- Address register is used to store the starting address of memory location for DMA data transfer.
- The address in the address register is automatically incremented after every read/write/verify transfer.
- The count register counts the number of byte or word transferred by DMA.
- In read transfer, the data is transferred from memory to I/O device.
- In write transfer, the data is transferred from I/O device to memory.
- Verification operations generate the DMA addresses without generating the DMA memory and I/O control signals.
- The 8237 has two eight-bit registers called mode set register and status register.

Q. Non-unique decoding technique is used for in small memory systems and it decreases the cost of decoding circuit simply eliminates the unnecessary hardware by using only the required number of address lines.	Unique decoding technique is used in large memory systems. However it increases the cost of decoding circuit.
---	---

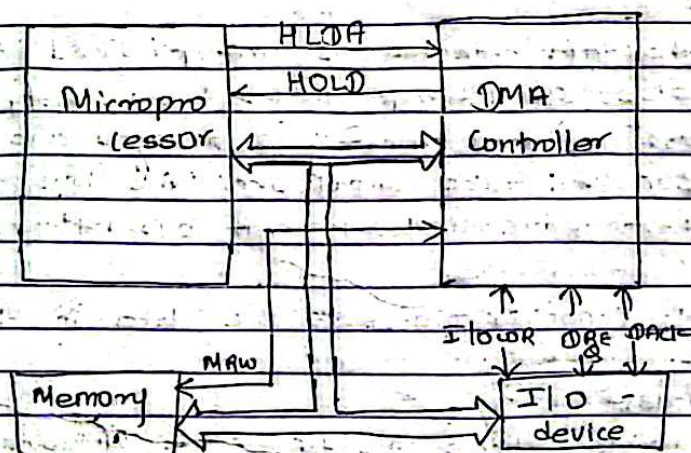
4- Difficult to future expansion	Easy for future expansion
----------------------------------	---------------------------



I/O Interfacing

DMA (Direct Memory Access)

BASIC DMA operation :-



1. I/O mapped I/O
2. Memory mapped input.

I/O mapped I/O and Memory mapped input

Memory mapped I/O	I/O mapped I/O
1. I/O devices are identified and addressed with 16-bit address i.e. A ₀ to A ₁₅ lines are used to generate device address.	I/O devices are identified and addressed with 8-bit address. So, A ₀ to A ₇ or A ₈ to A ₁₅ lines are used to generate device address.

2. The devices are enabled by memory related control signals.

Data bytes are transferred using the instruction related to memory access like STA address, LDA address, LDAX, RP, CMP M, MOV R₁ M etc.

3. The total addressing capability of the 8085 µP in this method is $2^{16} = 65536$ bytes = 64Kb.

Decoding 16-bit address may require more hardware.

4. Decoding 16-bit address may require more hardware.
5. Memory like RAM, ROM, EPROM, programmable peripheral device etc use memory mapped I/O like 8255A, 8251, 8254 etc used I/O mapped I/O.
6. Data transfer betⁿ any register and I/O devices.
7. Data transferred betⁿ I/O devices and accumulator only.
8. MEMR and MEMW control signals are used to control read & write I/O operation.
9. IOR and IOW controls signals are used to control read and write I/O operation.

Address decoding are categorized in the following two groups:-

Partial Address Decoding
(Non unique)

Complete address decoding
(Unique)

1. If all the address lines available on that mode are not used in address decoding, then that decoding is called partial address decoding.
2. This technique of address decoding is simple but there may be a chance of address conflict.
3. If all the address lines are available on that mapping mode are used for address decoding then that decoding is called unique address decoding.
4. This address decoding technique is complicated but it is the best for all cases.

ADDRESS DECODING

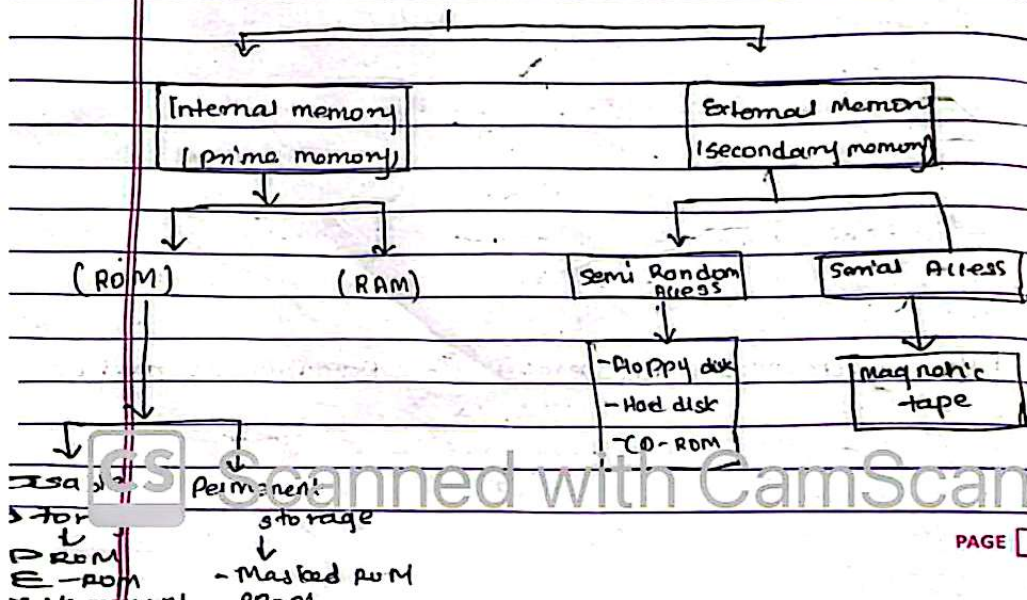
- The processor communicates with all the parts interconnected in the system through a common address and data bus. As a result of this, only one device can transmit data at a time and others can only receive that data.
- If more than one device attempt to send data through the bus at the same time, the proper communication among the device does not become possible because the data sent by them gets ~~errot~~ garbled. To avoid this situation, insuring that the proper device gets addressed at proper time, the technique called address decoding is used.
- In the address decoding method, all device like memory blocks, I/O units, etc. are assigned with a specific address. The address of this device is determine from the way in which the address lines are used to derive a special device selection signal known as chip select. If the μp has to write to or read from a device, the cs signal to that block should be enabled and the address decoding circuit must ensure that cs signal to other devices are not activated.

Depending upon the no of address lines used to generate chip select signal for the device, there are two methods of address decoding, they are.

Memory Interfacing

- Interface: It is a path of communication between two components.
- Interfacing is a technique of connecting up with memory.
- When up is executing an instruction, it needs to access the memory for reading instruction codes and the data.
- For Interfacing, both the memory and the up require some signals to read from and write to registers.
- Interfacing process needs to match the memory requirements and up signals.

TYPES OF MEMORY



Design the address decoding circuit for two RAM blocks of 256 bytes each consequently at address 5600H.

Sol:

Address decoding ckt.

For RAM 1

Base address = 5600H

End address = Base address + (Number of locations in 256 RAM-1)

$$= 5600H + (256-1)D$$

$$= 5600H + 255D$$

$$= 5600H + 00FFH$$

$$\text{Number of address line (N)} = \log_2(256) = 8$$

i.e. A7 - A0 are connected to internal decoder of RAM

For RAM 2

Base address = End address of RAM 1 + 0001H

$$= 56FFH + 0001H$$

$$= 5700H$$

End address = 5700H + 0 (256-1)D

$$= 5700H + 255D$$

$$= 57FFH$$

No. of address line = 8

i.e. A7 - A0 are connected to internal decoder of RAM 2.

- It has three control signals $\overline{ALE}$ (active low), $\overline{RD}$ (active low) and $\overline{WR}$ (active low) and three status signals $\overline{IO/\overline{M}}$ (active low), $\overline{S_0}$ and $\overline{S_1}$.
- $\overline{ALE}$ is used for providing control signal to synchronize the components of microprocessor and timing for instruction to perform the operation.

Significance of timing diagram

- Timing diagram gives us a perspective and help us understand the process of execution of a particular instruction in detail.
- We get to see what's going on inside the μp in every clock cycle.
- It is an extremely minute level of examining the working of the μp .
- It helps us visualize the whole process with respect to time.

Synchronous and Asynchronous bus

Synchronous bus	Asynchronous bus
- In synchronous buses, the steps of data transfer take place at fixed clock cycles.	- There is no fixed clock cycles in asynchronous buses.
Synchronous buses are typically faster than asynchronous buses.	Asynchronous buses are slower than synchronous buses.

Used in high speed transmission. Used in low speed transmission.

Advantages
 Involves very little logic and can run very fast. It is appropriate for different speed devices.

Disadvantage
 It is not reliable for long transmission.
 Every device on the bus run at the same clock rate.

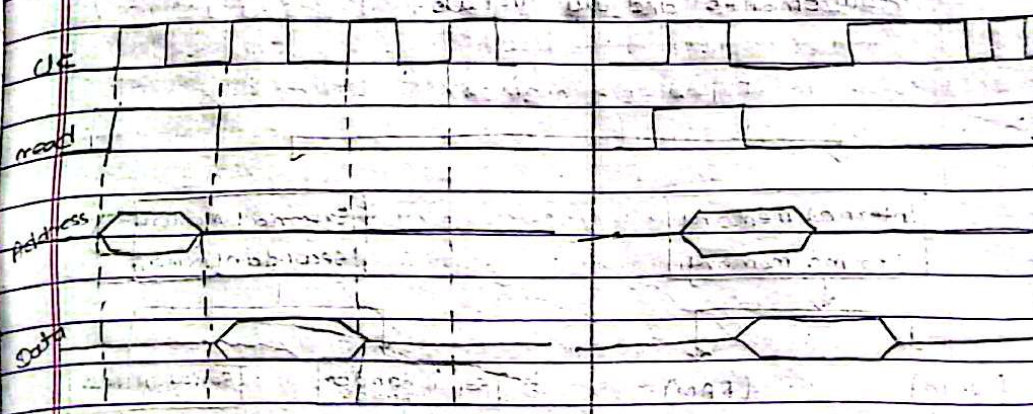


Fig: Synchronous Read Operation Fig: Asynchronous read operation

Register of 8085

- The 8085 has six general-purpose register to store 8-bit data during program execution.
- These register are identified as B, C, D, E, H, & L.
- They can be combined as 16 bit register pair as B, C, D, E and H, L to perform some 16-bit operation.

Accumulator (A)

- The accumulator is an 8 bit register that is part of the arithmetic/logic unit (ALU).
- This register is used to store 8-bit data and to perform arithmetic and logical operations.
- The result of an operation is stored in the accumulator.
- All the in 8085 up is performed on the accumulator or with the help of accumulator, therefore 8085 microprocessor is commonly called 8085A (Accumulator-based up).

Flags:- $D_7, D_6, D_5, D_4, D_3, D_2, D_1, D_0$

S	Z	AC	P	CY
---	---	----	---	----

Flag register of 8085

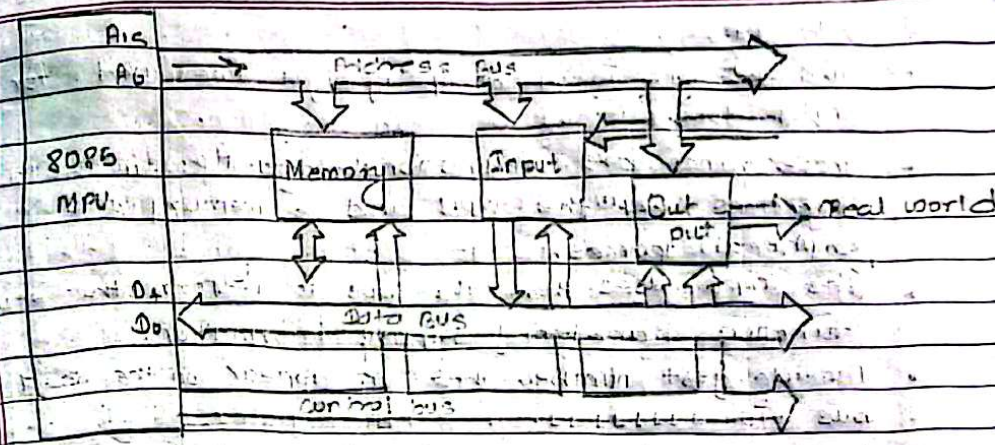
- The ALU includes five flip flop that are set or reset according to the result of an ALU operation.

Introduction to 8085 microprocessor:

The salient features of 8085 microprocessor

- 8085 is an 8-bit microprocessor manufactured with N-MOS technology.
- It has 16 bit address bus and hence can address up to $2^{16} = 65536$ bytes (64KB) memory locations through A0-A15.
- It supports external interrupt request 8085 consists of 16-bit program counter (PC) and stack pointer (SP).
- The first 8 lines of address bus and 8 lines of data bus are multiplexed A00-A07. Data bus is a group of 8 lines D0-D7.
- It is enclosed with 40 pins DIP (Dual In Line package).
- It requires a signal +5V power supply and can operate at 3MHz, 5MHz and 6MHz serial in | serial out port.
- Six 8-bit general purpose register arranged in pairs: BC, DE, HL.

8085 Bus Structure:



8085 Bus Structure

Address Bus

- The address bus is a group of 16 lines generally identified as A0 to A15.
- The address bus is unidirectional: bits flow in one direction from the MPU to peripheral devices.
- The MPU uses the address bus to perform the A15 function: identifying a peripheral or a memory location.
- 8085 has $2^{16} = 65536$ -byte memory locations ranging from 0000H to FFFFH. The memory address bits are represented as

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
= 0000H															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
= FFFFH															

Do not have power saving feature

Have power saving mode like idle mode, and power saving mode.

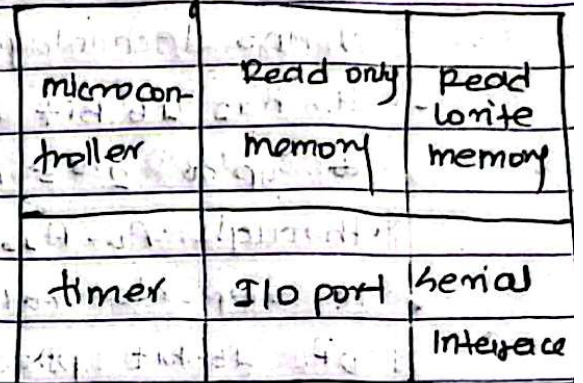
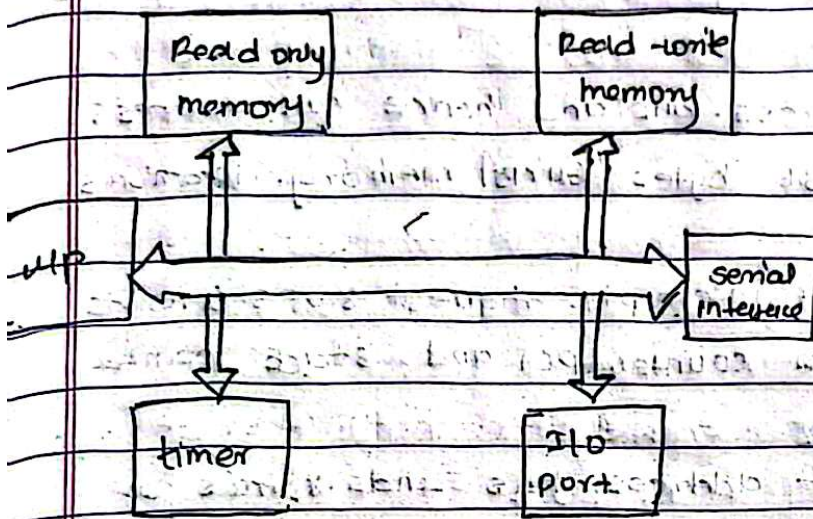


Fig. 1. Block diagram of

Block diagram of upc and 2 to 10 upc

SAM.	The volatile memory (RAM) for the μp is in the range of the 512MB to 32MB.	The volatile memory (RAM) for the μc is the range of 2KB to 256KB.
ROM.	The hard disk (ROM) for the μp is in the range of the 128KB to 2TB.	The hard drive or flash memory (ROM) is in the range of the 32KB to 2MB.
	Programming:	
	The program for the μp can be changed for different applications. The programming of the μp is difficult compared to the μc .	The program for the μc is fixed once it is designed.
Size	It is available in 32-bit & 64-bit.	It is available in 8-bit, 16-bit & 32-bit.
C.	The cost of μp is high compared to the μc .	It is cheaper.
E.	The overall size of the system is large.	Small.
Architecture	Based on Von Neumann architecture	Harvard architecture

Use RISC processor
(Reduced Instruction set
computer)

Use CISC processor
(Complex Instruction
set computer)

More space is required
in Harvard Architecture.

Less space is required
in Von Neumann architec-
-ture.

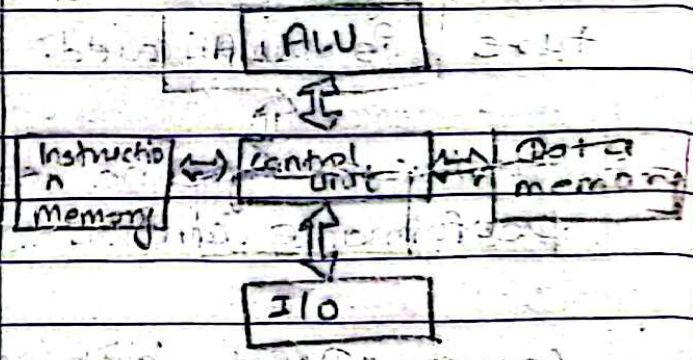
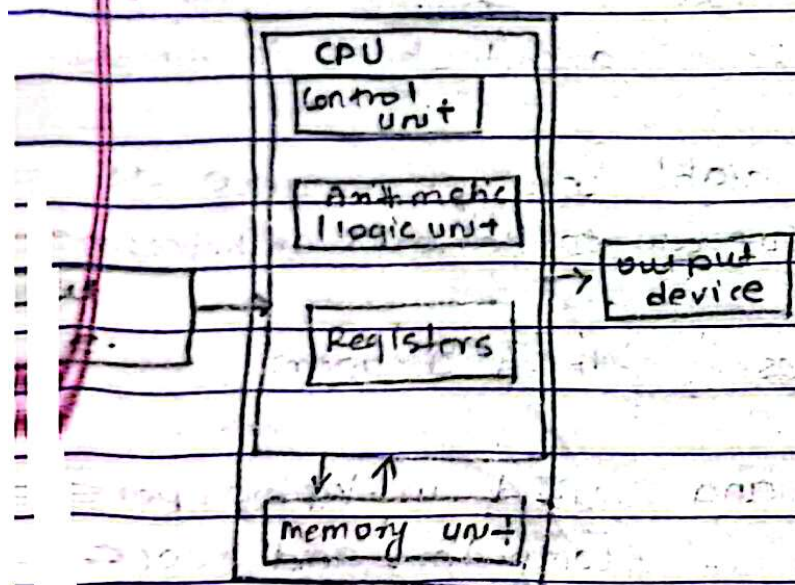
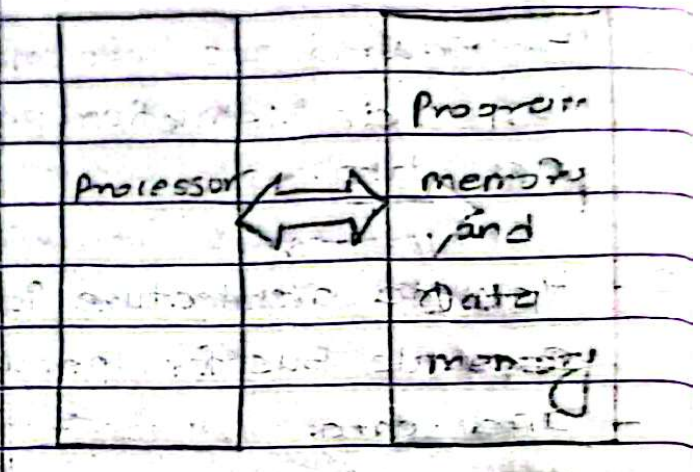
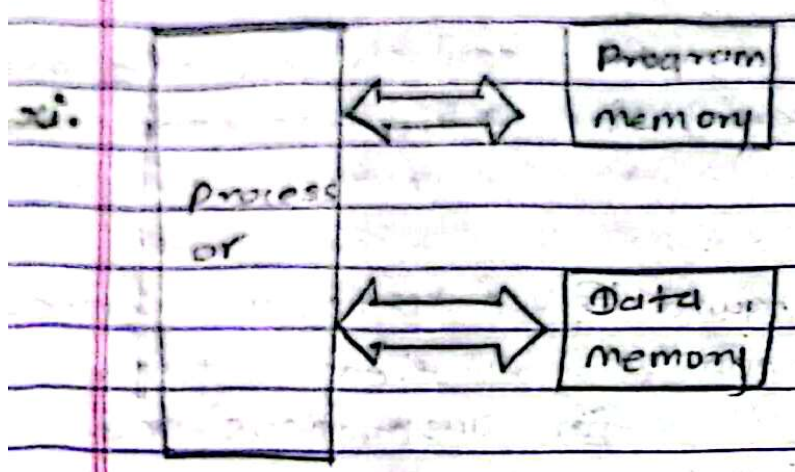


Fig: Harvard model

Fig:

Difference between Harvard and Von-Neumann Architecture.

Harvard Architecture	Von-Neumann Architecture
1. This name is originated from Harvard Mark II, a relay based old computer.	It is named after the mathematician and early computer scientist John von Neumann.
2. It required two memories for their instruction (program) and data.	It required only one memory for their instruction and data.
3. Harvard architecture is required separate bus for instruction and data.	Von-Neumann architecture is required only one bus for instructions and data.
4. Design of Harvard architecture is complicated.	Design of the von-Neumann architecture is simple.
5. Easier to pipeline, so high performance can be achieved.	Low performance as compare to Harvard architecture.
Comparatively high cost.	It is cheaper.
Used in microcontroller and signal processing.	Used in laptop, personal computer and work station.

VON NEUMANN AND HARVARD ARCHITECTURE

- Von Neumann architecture was first published by John W. Van Neumann in 1946.
- His computer architecture design consists of a control unit, arithmetic and logic (ALU), memory unit, Registers and I/P, O/P.
- Von Neumann architecture is based on the stored program, computer concept, where an instruction data and program data are stored in the same memory.
- This design is still used in most computers produced today.
- Von Neumann architecture is required only one bus for instruction and data.
- The processor needs two clock cycles to complete an instruction. So pipelining is not possible with this architecture.
- In the first clock cycle, processor gets the instruction from memory and decodes it. In next clock cycle, the required data is taken from memory. For each instruction, this cycle repeats and hence needs two cycles to complete an instruction.
- Von Neumann architecture is relatively older and was replaced by Harvard architecture.

Fourth Generation (32-bit)

- Several different companies introduced the 32-bit, but the most popular one is the Intel 80386 in 1985
- clock speed = 16MHz
- No. of transistors = 295K
- Address bus = 32 bit
- Data Bus = 32-bit
- IPS = more than 5 million
- Memory addressing capacity = 4GB real, 64TB virtual
- Pins = 132
- contains = MMU on chip
- Pentium in 1993, CS = 66MHz
- Pins = 234 P(n)
- Address bus = 32 bit
- Data Bus = 32 / 64-bit

Fifth Generation (64-bit)

From 1995 till now we are in the fifth generation i.e. 64-bit processors like Pentium 4, Celeron, dual, quad and octa core processors came into existence.

Intel core 2 in 2006, CS = 1.2-3GHz, NoT = 291 million

P₃, i₅ in 2009, CS = 2.2GHz

i₇ in 2010, CS = 3.3GHz

i₉ in 2014 (core i₉ x-series) 6.1GHz

Second generation (8-bit)

- The second generation microprocessor was introduced in 1972 again by Intel.
- It was a first 8-bit up which could perform arithmetic and logic operations on 8-bit words.
- It was Intel 8008, and another ~~ver~~ improved version was Intel 8088 / 1979 8085 in 1976.
- Some other 8-bit processors are Zilog 80 and Motorola M-6800.
- Clock speed = 500 kHz for 8008
2 MHz for 8080
3-5 MHz for 8085
- Pins = 40
- No. of transistors = 6.5K for 8085
- Address bus = 16
- Data bus = 8 bit
- Memory address capacity = 64Kb for 8085

THIRD GENERATION (16-bit)

- The third generation μ P, introduced in 1978, were represented by Intel's 8086, Zilog Z800 and 80286, which were 16-bit processors with performance like mini computer.
- Clock speed = 5-8 MHz
- No. of transistors = 29K
- IPS = 2.5 million
- Memory addressing capacity = 1MP
- Pins = 40, widely used in PC / XT
- Address bus = 20 bit
- Data bus = 16 bit

Introduction to microprocessor

1.1 Evolution of microprocessor

DATE

Transistor was invented in 1948 (03 dec, 1948 in Bell lab)
IC was invented in 1958 (Fair child semiconductors) by
Texas instruments J Kilby.

We can categorize the μp according to the generation
or according to the size of the μp .

First generation (μ -bit)
 μ bit μp (4004)

- The first microprocessor was introduced in 1971 by Intel corp
- It was named intel 4004 since it was a μ bit processor.
- It was a processor on a single chip.
- It could perform simple arithmetic and logical operations.
- No of transistors = 2310
- Instruction per second (IP) = 60K
- ⊙ It could perform simple arithmetic and logical operation.
- It had a control unit capable of performing control functions like fetching an instruction from storage memory, decoding it and then generating control pulses to execute it.
- Memory addressing capacity = 1KB
- Pins = 16
- Address bus = 10 bit, Data bus = μ bit
- Clock speed = 740KHz

PAGE

	attached to it which indicate the start of the new byte.
3. External clock is used.	Internal clock is used.
4. Synchronous transmission is fast.	Asynchronous transmission is slow.
5. Transmitter and receiver having common clock.	Separate clock.
6. costly	economical
7. Used for high speed of transmission.	Used for low speed of transmission.
8. In synchronous transmission, there is no gap between data.	In asynchronous transmission there is present gap bet ⁿ data.
9. Time interval of transmission is constant.	not constant it is random.

DMA (Direct Memory Access): [8054/8237]

Microprocessor:

It is multipurpose, programmable, clock-driven, register-based electronic device that reads binary instruction from a storage device (memory), accepts binary as input and processes data according to those instructions and provides result as output.

- Each MP communicates and operates in the binary number 0 and 1, called bits.
- Each MP has fixed sets of instructions in the form of binary pattern called machine language.

#> Synchronous data transmission and asynchronous data transmission.

Synchronous data transmission	Asynchronous data transmission
→ In synchronous transmission data is sent in form of blocks or frames.	In asynchronous transmission data is sent in form of byte character.
- Synchronous transmission requires a clock signal bet ⁿ the sender and receiver so as to inform the receiver about new byte.	Asynchronous transmission sender and receiver do not require a clock signal as the data sent here has a parity bit.

INTERRUPT VECTOR TABLE: (IVT) → Pdf

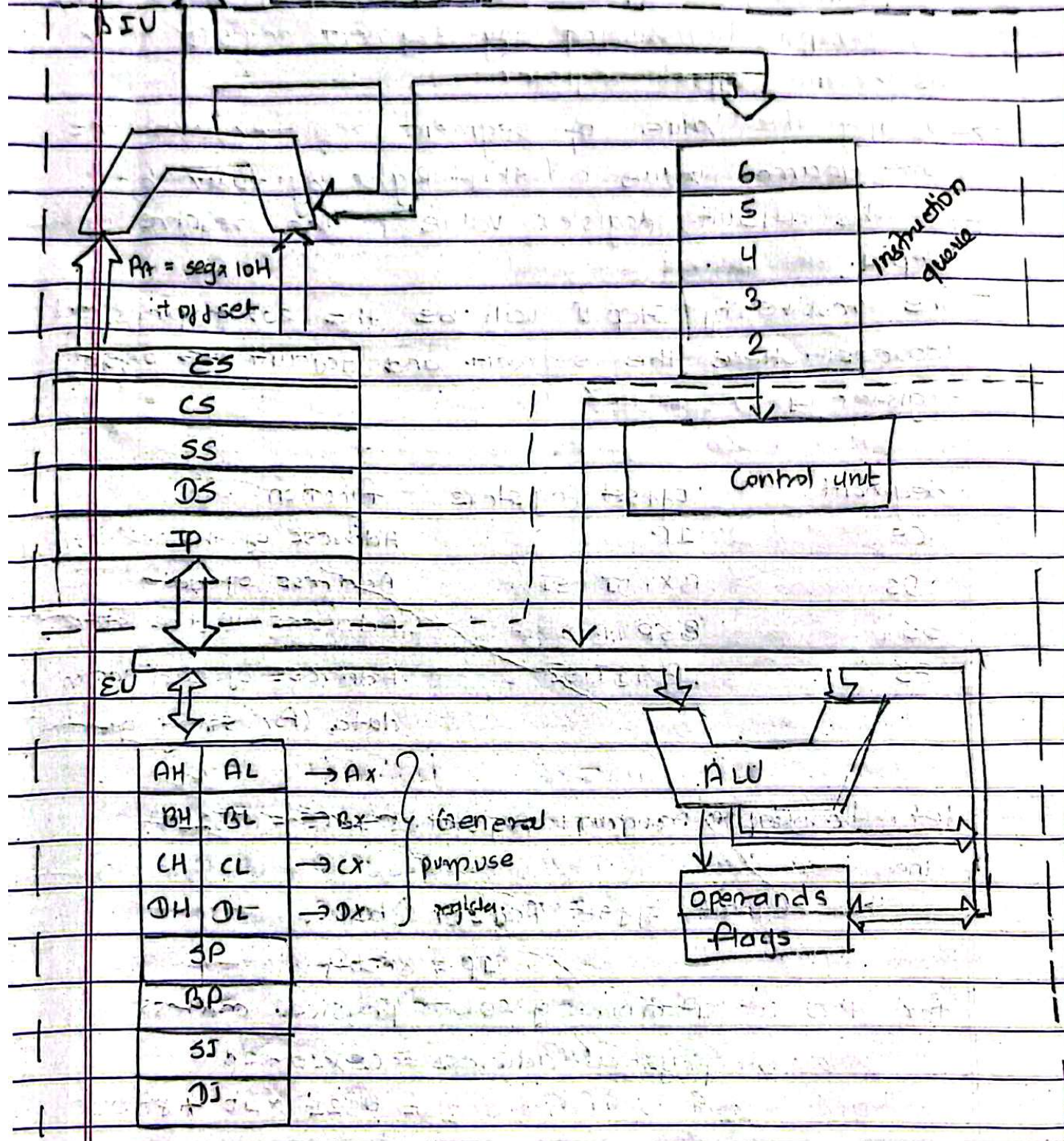
Synchronous and Asynchronous Communication:-

Serial and Parallel communication

Serial communication	Parallel communication
- In this mode of transmission single cable is used to transmits number of bits serially. - Only one bit is transmitted at a time. - This mode is one by n time ($1/n$) slower as compare to parallel connection. - It is employed for long distance communication. - cost low - simple and easy ckt used - High frequency operation is more efficient - full duplex	- In this mode of transmiss. n number of cable are connected to transmits n-number of bits. - n-number of bits are transmitted at once from n number of cable. - Its transmits rate is n times faster than serial transmission. - It is limited with normally with in a system (short distance). - cost high - relative complex ckt used & system up generation is quite difficult. - High frequency operation is less efficient - half duplex

To memory and
input/output

DATE



Block diagram of 8086 uip

8085 Differentiate betⁿ 8085 and 8086.

8085	8086
- It is an 8-bit microprocessor. (28)	- It is a 16 bit. micro-proc. (286)
- 8085 has 16-bit address. i.e. its min. size will be $2^{16} = 65536$.	- Its address is in 20 bit, therefore the min size is, $2^{20} = 1\text{mb}$
- 8085 does not support instruction pipe line.	- It support instruction pipe line.
- 8085 has five flip flop (flags).	- It has 9 flags.
- 8085 does not support maximum mode (multiprocessor).	- It support minimum as well as maximum mode of operation.
- 8085 has limited instruction.	- 8086 has complex instruction set.
- 8085 does not support string.	- 8086 has support string.

is called master 8259A. With the master 8259A eight slave 8259A can be connected, generating a maximum of 64 interrupts per cascade connection:-

- The INT pin of all the slave 8259A are connected with the one of the IR line (IR₀ to IR₇) of the master 8259A.
- The cascade line (CAS₀ to CAS₂) of all the slave 8259A get connected with the CAS₀ to CAS₂ of master 8259A.
- The slave 8259A are enable use $\overline{SP}/\overline{EM}$ (slave program enable buffer bar) which are connected between master 8259A and all the slave 8259A. Following block diagram shows the generation of 64 interrupts using 8259A.

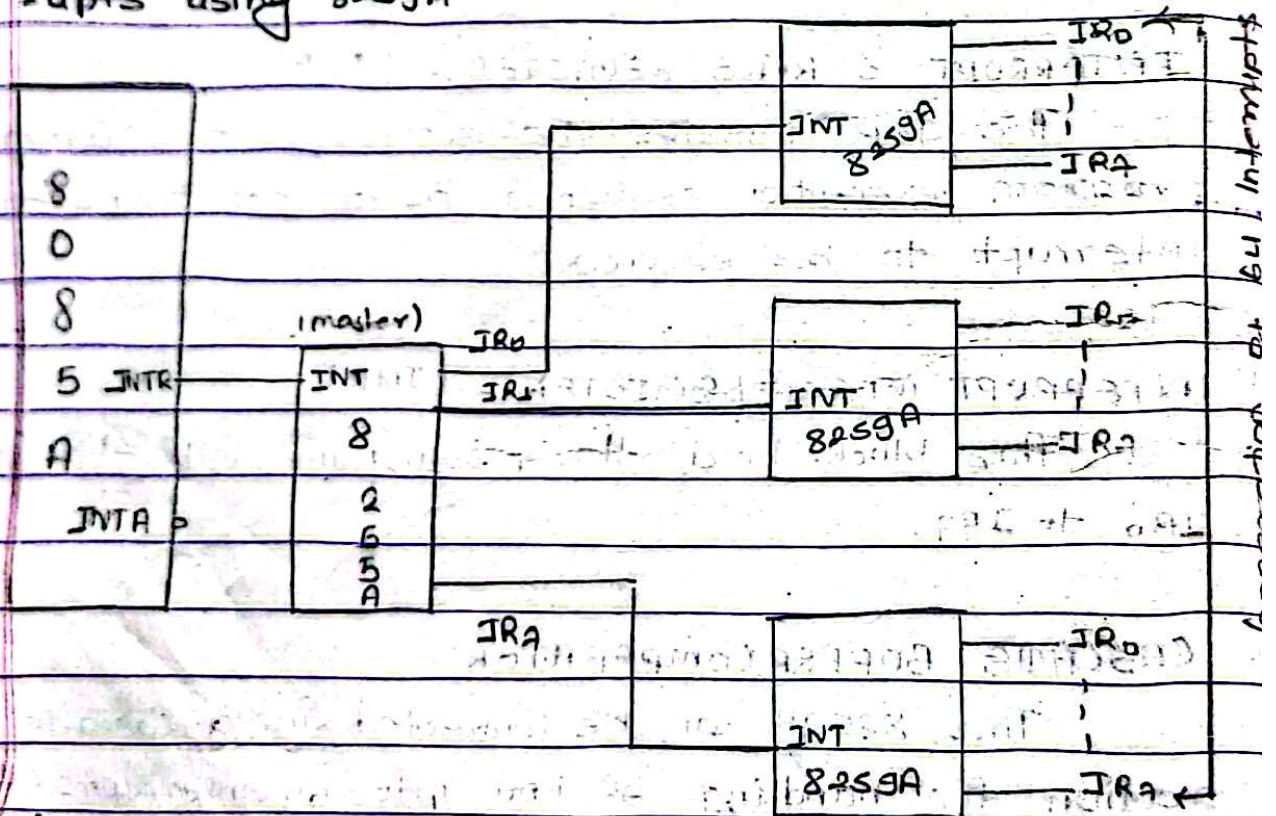


Fig. Generation of 64 Interrupts using 8259A

- INTA:-

This pin of all master slave 8259A get connected with the INTA of µp.

- iv. IRR (Interrupt request register):-

This block has eight interrupt request line IR₀ to IR₇. When an external device need to be service by the 8085 µp the device is connected through one of the IR₀ to IR₇ pin. A 8259A can be handle 64 interrupts in cascade mode.

- v. PRIORITY RESOLVE:-

This block assigned the priority level for the incoming interrupts from IR₀ to IR₇.

- vi. INTERRUPT SERVICE REGISTER:- (ISR)

The ISR provides the service routine where the program execution sequence has to be transfer for the interrupt to be service.

- vii. INTERRUPT MASK REGISTER:- (IMR)

This block hold the pending interrupts that occur from IR₀ to IR₇.

- CASCADE BUFFER COMPARTOR:-

The 8259A, can be connected in a cascade configuration for handling of two upto 64 interrupts. For these the 8259A which is directly connected with the 8085 µp

This pin goes active.

- **WRITE ($\overline{WR}$):-**

When an eight bit data from μp has to be transfer to the 8259A from the data bus buffer. The write ($\overline{WR}$) is active.

- **$\overline{CS}$:-**

The 8085 μp has 16 address line among these address line A_0 is directly connected betn 8085 μp & 8259A. Therefore the remaining address line A_1 to A_{15} are unused address line through which the $\overline{CS}$ signal is generated.

- **A_0 :-**

The 8259A is directly connected with the address line A_0 of the μp .

III. INTERRUPT CONTROL LOGIC.

This block generate the interrupt signal to the μp . It has two pin INT (Interrupt-request) and $\overline{INTA}$ (Interrupt acknowledgement).

- **INT:**

This pin is directly connected with the INTR of 8085A. The 8259A whose INT pin is connected with the 8085's INTR pin that 8259A is called master 8259A.

Non vectored

Software

✓

✓
✓

8259A PIC

The 8259A is used for handling multiple interrupt. As the 8085 μp has only five interrupts pin. Therefore the μp itself can not handle more than five interrupt. Therefore the programmable interrupts controller PIC is introduced.

Fig. above shows the block diagram of 8259A. It has 8 blocks and it is of 28 pin. The different blocks are:-

- **Data BUS BUFFER**:- It is an eight bit bidirectional bus in which the data to be communicated with μp is placed.

- **READ / WRITE CONTROL LOGIC**:- In this block the control signals are generated which are:-

READ ($\overline{RD}$):- When an eight bit data has to be placed into the data bus buffer from 8259A to 8085 μp

Overflow

Divide by 0

PIC (Programmable Interrupt Controller)

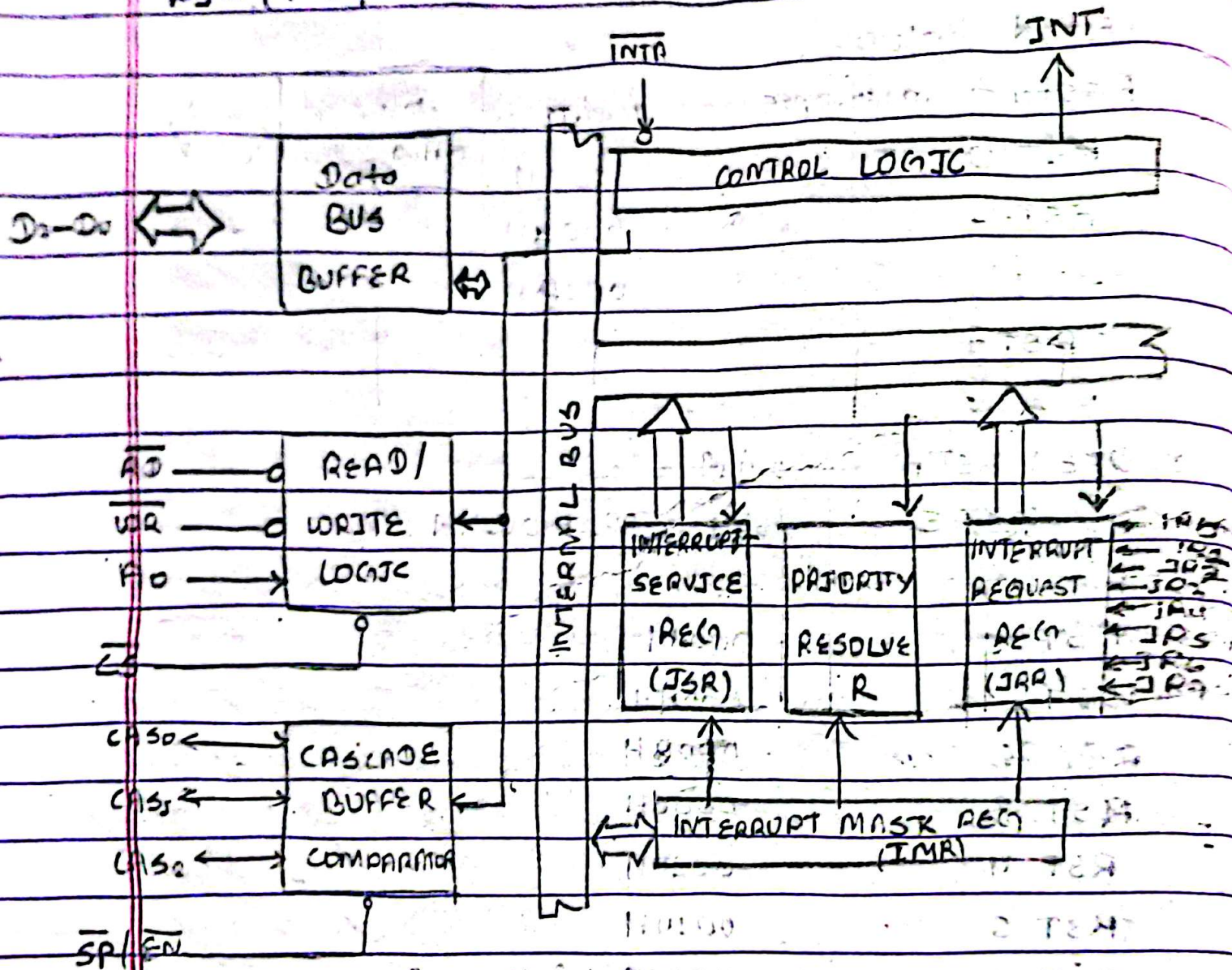


Fig: Block diagram of 8259A

External Interrupt:-

All the external interrupts are initiated by the peripheral devices that seek the attendance of the microprocessor.

Internal Interrupt:

All the internal interrupts are initiated within the μp which may be divide by zero, execution of illegitimate opcode and a register overflow.

However, in μp there is software interrupts called restart interrupts which are listed below:

Restart interrupts Vector location

RST 0 0000H

RST 1 0004H

!

0008H

RST 7

NOTE: $RST_n \rightarrow n \times 4 =$ _____

$RST 5 \rightarrow 5 \times 4 = 20 = 0014H$

RST 0 0000H

RST 1 0004H

RST 2 0008H

RST 3 000CH

RST 4 0010H

RST 5 0014H

RST 6 0018H

RST 7 001CH

* Non-maskable Interrupt

The non-maskable interrupt cannot be enable or disable by using an instruction in a program. They have the highest priority level. In 8085 μp there is only one non-maskable interrupt called TRAP.

* Vector & Non-vector:-

All the interrupts which are having pre-defined MIM address within the μp are called vector interrupt. The program that execute from the vect address is called Interrupt service routine (ISR). When vector interrupts are active the program execution sequence will be transfer to the MIM location given by the vector address. In 8085 μ the vector interrupts are TRAP, RST7.5, RST6.5 & RST5.5 & RST.

* Non-vector Interrupt:

The interrupts which do not have predefined MIM address within the μp is called non-vector interrupt. The non-vector interrupt will have vector address corresponding to the external device initiating the interrupt. In 8085 μp there is one non vector interrupt called INTR.

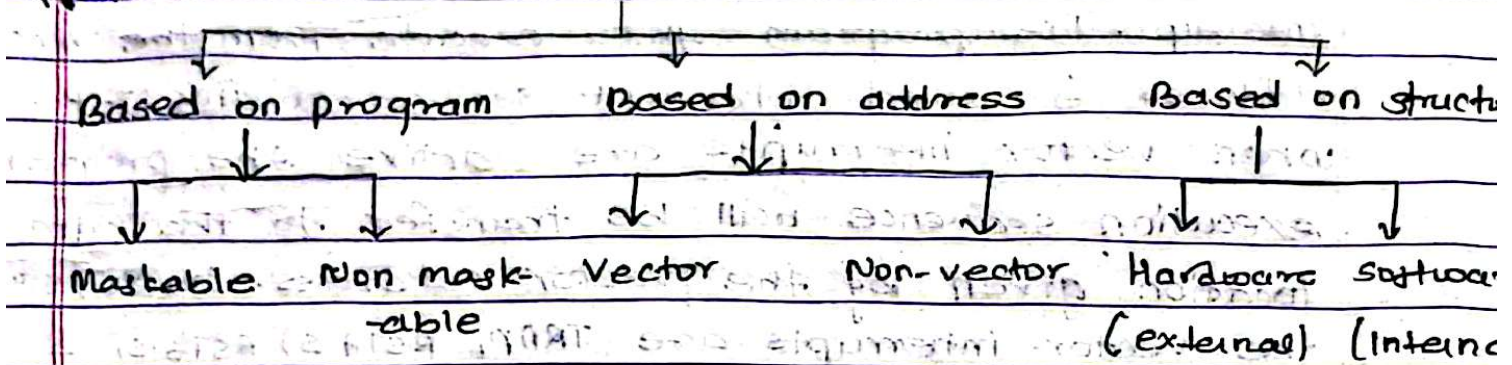
* External Interrupt and Internal interrupts

- When external device need to be service the $\overline{INTA}$ pin goes high.

$\overline{INTA}$:-

- It is an acknowledgement given by the μp in response to the occurrence of an interrupt.
- It is an active low outgoing signal through which the μp responds to the occurrence of the interrupt.

Types OF INTERRUPT:



Maskable and Nonmaskable

Maskable Interrupt :-

Those interrupt that can be enable or disable by using instruction in a program are called maskable interrupt. The maskable interrupt has lowest priority than the non-maskable interrupt. The maskable interrupt can be enable by using instruction EI (enable interrupt) and disable by using instruction DI . In 8085 μp , the maskable interrupts are $\text{RST}7.5$, $\text{RST}6.5$, $\text{RST}5.5$ & $\text{RST}4.5$.

RST 7.5 :-

→ It is a maskable interrupt when this interrupt is active the program execution sequence will transfer to memory location is given by

$$7.5 \times 8 = 60 = 003CH$$

→ It has the second highest priority and it is positive edge trigger.

RST 6.5

→ It is a maskable interrupt when this interrupt is active the program execution sequence will be transfer to memory address is given by

$$6.5 \times 8 = 52 = 0034H$$

→ It has the third highest priority.

→ It is level trigger.

RST 5.5 :-

→ It is a maskable interrupt when this interrupt is active the program execution sequence will be transfer to memory location is given by

$$5.5 \times 8 = 44 = 002CH$$

→ It has the fourth highest priority.

→ It is level trigger.

INTR (Interrupt Request)

→ It is a maskable interrupt without any predefined memory address.

→ It has the lowest priority.

Interrupt:

TRAP

RST 7.5

RST 6.5

RST 5.5

Restart

INTR (Interrupt Request)

INTA (Interrupt Acknowledge)

Priority level.

Trap > RST 7.5 > RST 6.5 > RST 5.5 > INTR

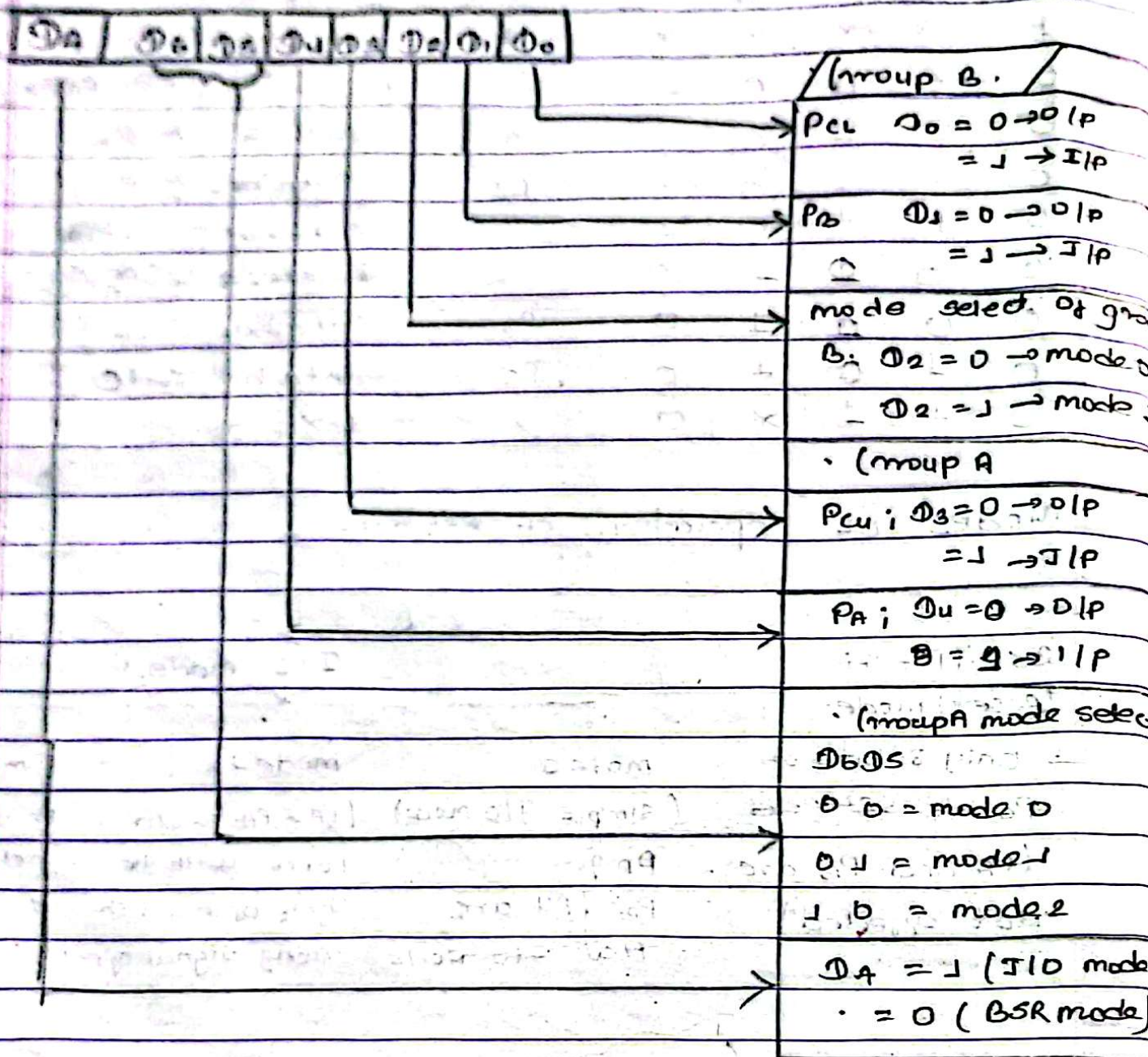
The process by which the microprocessor is suspended by what it is currently doing is called an interrupt. In 8085 ^{TRAP} up there are five interrupt pin they are, RST 7.5, RST 6.5, RST 5.5, INTR. In response to these interrupt the ^{TRAP} up acknowledge then by sending a active low signal called interrupt acknowledge bar.

TRAP:-

- It is a non maskable interrupt.
- It has the highest priority.
- It can be edge trigger or level trigger.
- When TRAP is encounter the program execution sequence will be transfer to the interrupt service within whose address is given by, $4.5 \times 8 = 36 = 0024H$.
- When TRAP is active the up doesnot response to any other interrupt.

Example:- power failer, emergency shut down.

Control word format :



The control word is the 8-bit binary pattern based on the value of which the 8255A can be configured in different mode. For this purpose the value of control word must be placed into the control register.

$\overline{CS} = 0$; 8255 selected
 $\overline{CS} = 1$; 8255 not selected

DATE

Operation of 8085 A with 8255A

$\overline{CS}$	A_1	A_0	$\overline{R_0}$	$\overline{WR}$	Ports	operation
1	x	x	x	x	x	8255A is not selected
0	0	0	0	1	PA	Data Bus $\leftarrow$ PA (data is I/P of 8085)
0	0	1	0	1	PB	Data Bus $\leftarrow$ PB
0	1	0	0	1	PC	Data bus $\leftarrow$ PC
0	1	1	0	1	CR	Data bus $\leftarrow$ Control register
0	0	0	1	0	PA	Data bus $\Rightarrow$ PA [data is O/P from 8085 into PA]
0	0	1	1	0	PB	data bus $\Rightarrow$ PB
0	1	0	1	0	PC	data bus $\Rightarrow$ PC
0	1	1	x	0	x	x

Modes of operation of 8255A.

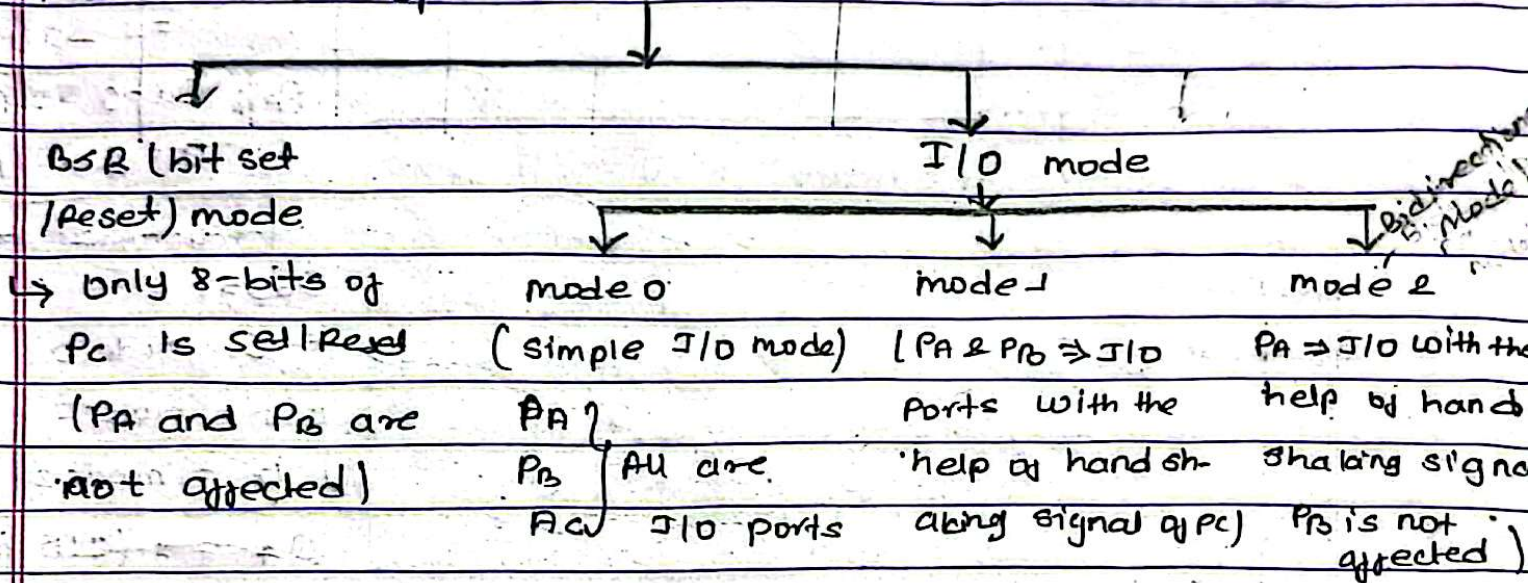


Fig: (Classification of modes of operation of 8255A)

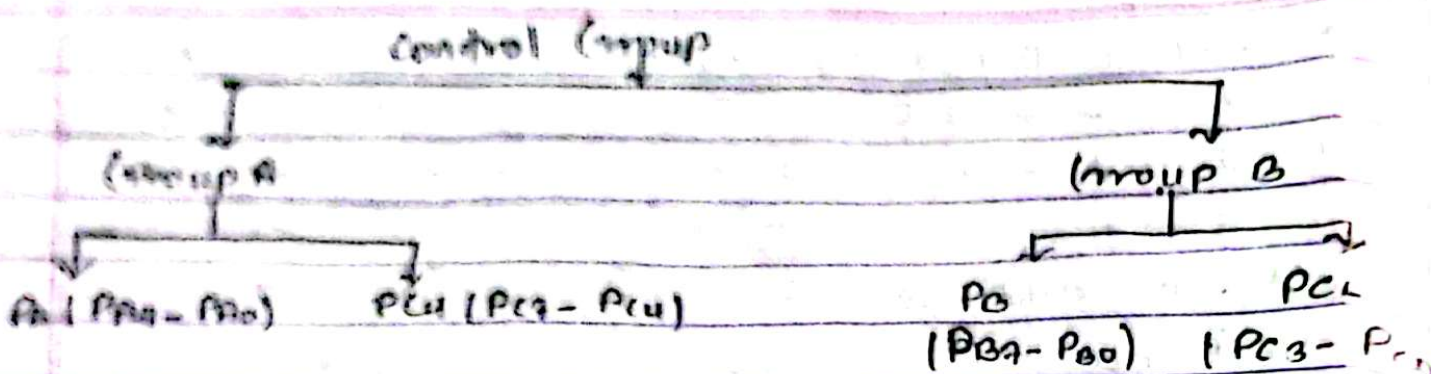
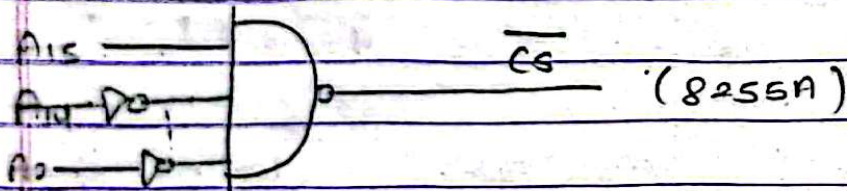


Fig: Classification of PPI/O ports in 8255

A ₁	A ₀	Port selected
0	0	PA
0	1	PB
1	0	PC
1	1	Control Register (CR)

Selection of ports

Q) Determine the address of Port A, Port B, Port C & Port CR



Soln

A₁₅ A₁₄ A₁₃ A₁₂ A₁₁ A₁₀ A₉ A₈ A₇ A₆ A₅ A₄ A₃ A₂ A₁ A₀

A₁₅ A₁₄ A₁₃ A₁₂ A₁₁ A₁₀ A₉ A₈ A₇ A₆ A₅ A₄ A₃ A₂ A₁ A₀

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 → 8000H ⇒ PA

0 1 = 8001H ⇒ PB

1 0 = 8002H ⇒ PC

1 1 = 8003H ⇒ CR

We know

$$12 = \log_2 (\text{m/m size})$$

$$\text{m/m size} = 2^{12} \\ = 4096$$

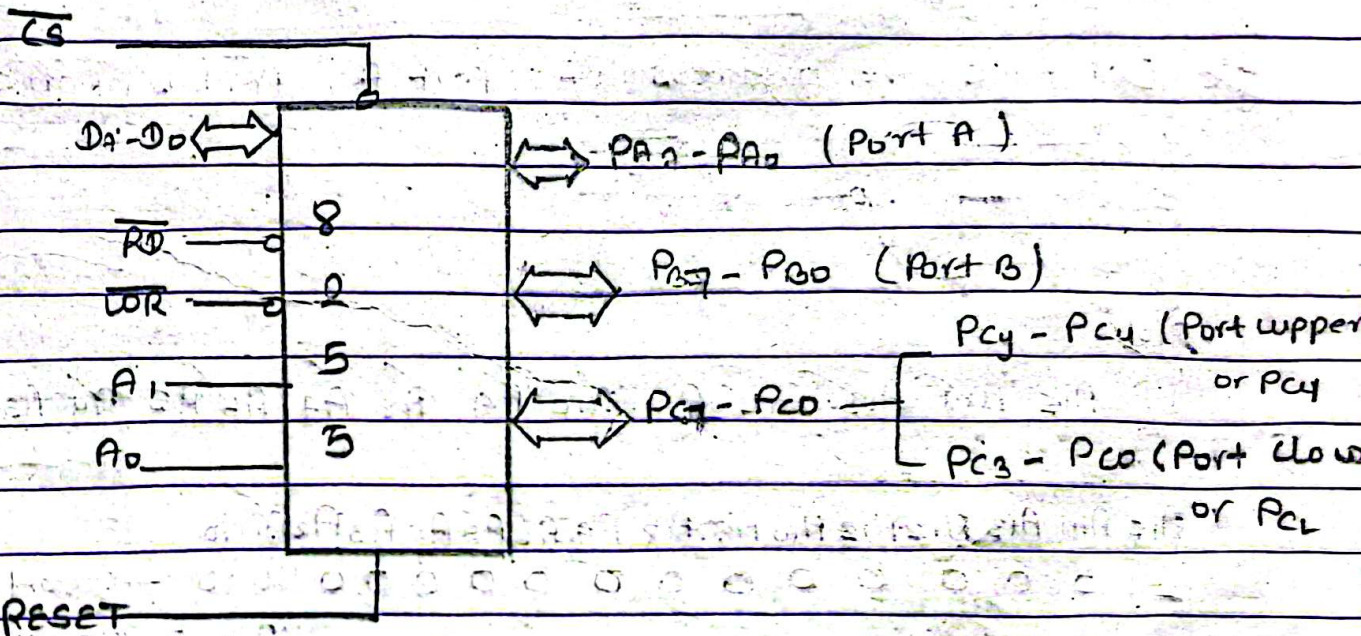
$$\therefore \text{Range} = 0000H - 0FFFH$$

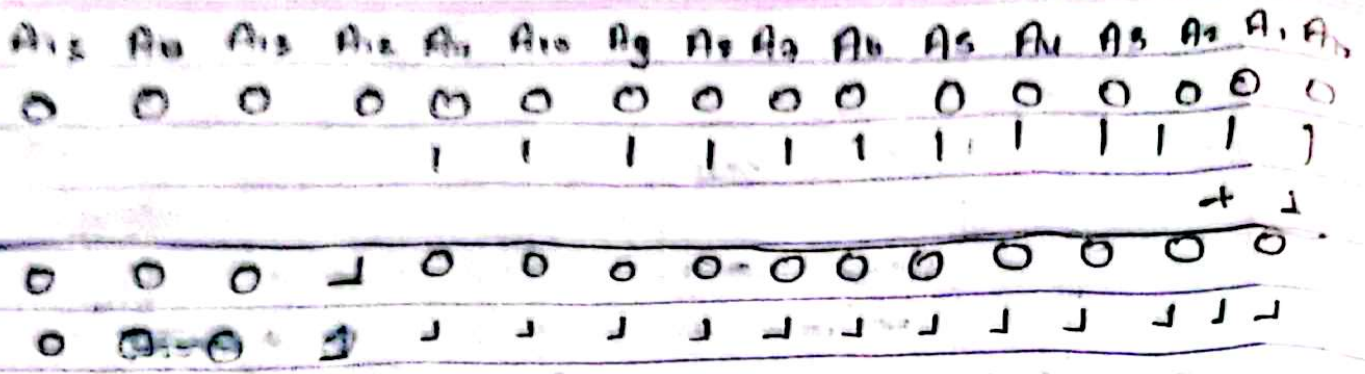
memory size = 4KB #

1	0	0	→ Y ₄
1	0	1	→ Y ₅
1	1	0	→ Y ₆
1	1	1	→ Y ₇

PPT (PROGRAMMABLE PERIPHERAL DEVICE)

(P255 or 8255A)

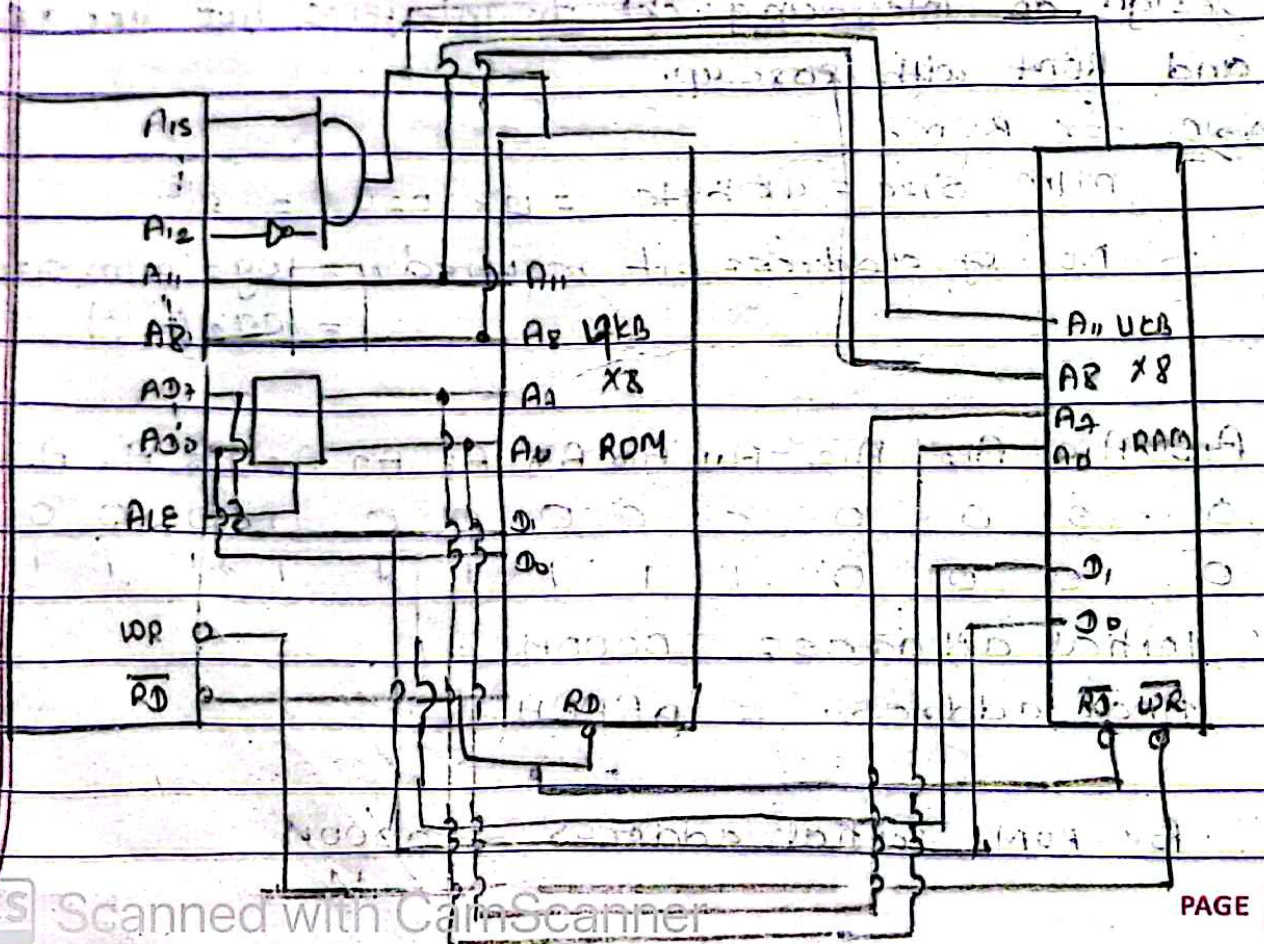
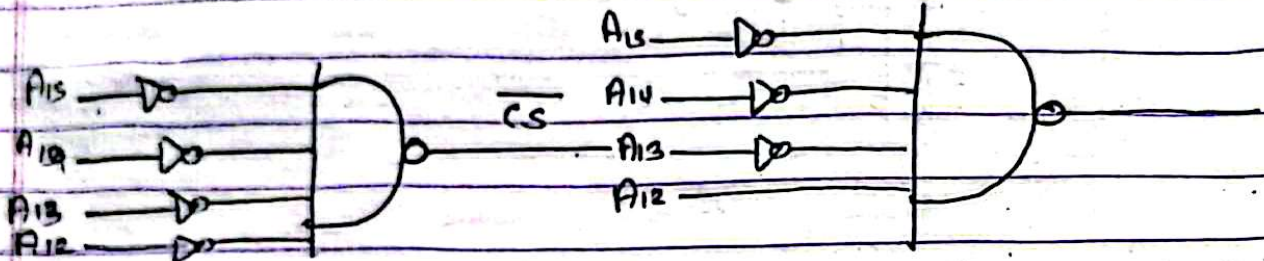




Initial min address of ROM = 1000H.
 Final address = 1FFFFH

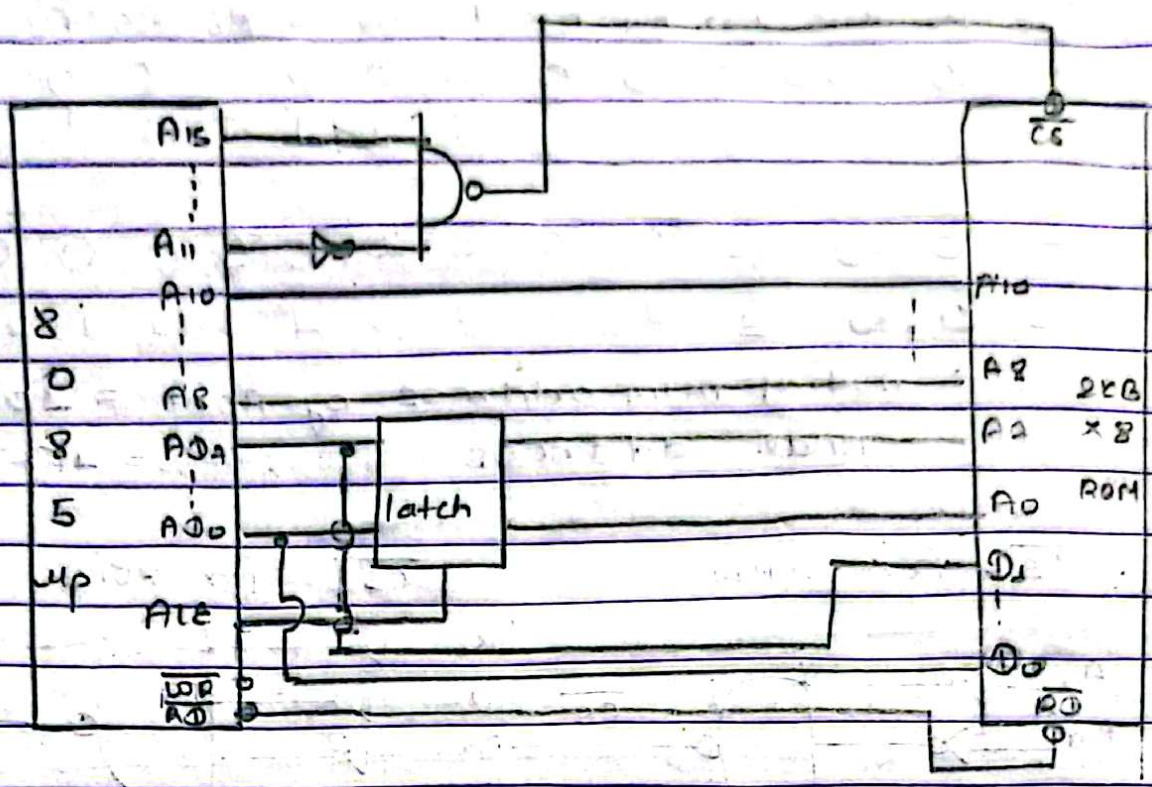
$\overline{CS}$ for RAM

$\overline{CS}$ for ROM



POM = 1000

DATE



Design an interfacing ckt to interface two 4K x 8 RAM and ROM with 8085up.

Soln for RAM.

MIM Size = 4K byte = $4 \times 1024 = 2^{12}$

No. of address bit required, $= \log_2(\text{MIM size})$
 $= \log_2(2^{12}) = 12 \text{ bytes,}$

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Initial address = 0000H

Final address = 0FFFFH

for ROM, initial address = 0000H

Design an interfacing circuit to interface two 2Kbyte X
8 ROM with 8085 up, having initial m/m addr
ess 8000H

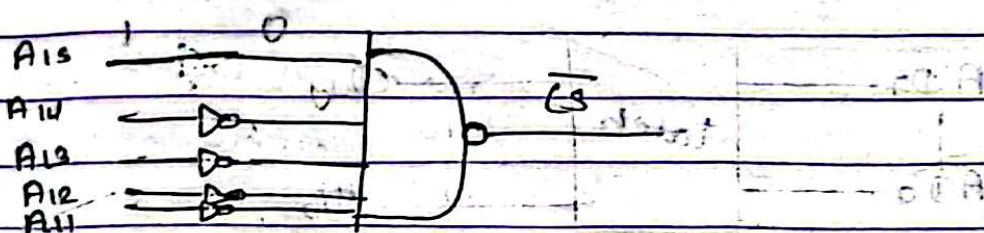
Soln

$$\begin{aligned} \text{m/m size} &= 2\text{k byte} \\ &= 2 \times 1024 \\ &= 2^{11} \end{aligned}$$

$$\begin{aligned} \text{No. of address bit required,} \\ &= \log_2 (\text{m/m size}) \\ &= \log_2 (2^{11}) \\ &= 11 \text{ bytes} \end{aligned}$$

A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

The $\overline{\text{CS}}$ generated as



Initial address = 8000H

Final address = 87FFH

Design the interfacing ext to interface 4K byte
RAM with 8085 μ p.

(Given)

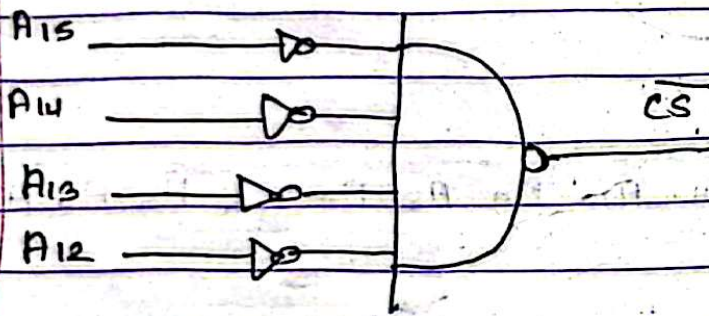
$$\begin{aligned} \text{m/m size} &= 4\text{K byte} \\ &= 4 \times 1024 \\ &= 2^{12} \text{ bytes} \end{aligned}$$

$$\begin{aligned} \text{No. of address bit required} &= \log_2(\text{m/m size}) \\ &= \log_2 2^{12} \\ &= 12 \text{ bits} \end{aligned}$$

Now

A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

The $\overline{\text{CS}}$ is generated as,



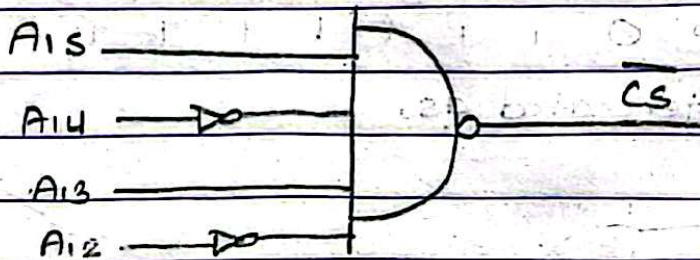
Full address decoding (Absolute..)

The address decoding process in which all the unused bit are used for generating the chip select signal it is called full address decoding.

Partial addressing decoding :- in which

The address decoding process some of the unused address bit are unused during the generation of $\overline{CS}$ signal. it is called partial address decoding.

Determine the address range for the device that is selected with the following $\overline{CS}$.



	unused				used address bits											
	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
initial	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
final	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1

$A000H$ = Initial address
 $AFFFFH$ = final address

address range.

No. of address bit used = 12 = $\log_2 (m/m \text{ size})$

$$\therefore m/m \text{ size} = 2^{12}$$

$$= 2^2 \times 2^{10} \text{ Bytes}$$

$$= 4 \times 1024 = 4096 \text{ byte}$$

$$= 4K \text{ byte.}$$

* Interfacing:-

The process of connecting peripheral devices to the microprocessor is called an interfacing. The interfacing is done so as to establish the communication between the mp and peripheral device.

Parameters required for interfacing:-

- i. $A_{15} - A_8$
- ii. $A_{07} - A_{00}$
- iii. ALE
- iv. $\overline{RD}$ (Read)
- v. Write ($\overline{WR}$)
- vi. Chip select ($\overline{CS}$)

$\overline{CS}$ (chip select) (Chip Enable):-

- It is an active low control signal which is generated by using unused address bit of the mp.

Address Decoding:-

The process of generating chip select signal from the unused address bit is called address decoding. The address decoding can be performed in two ways.

- i. Absolute address decoding
- ii. Full address decoding
- iii. Partial address decoding

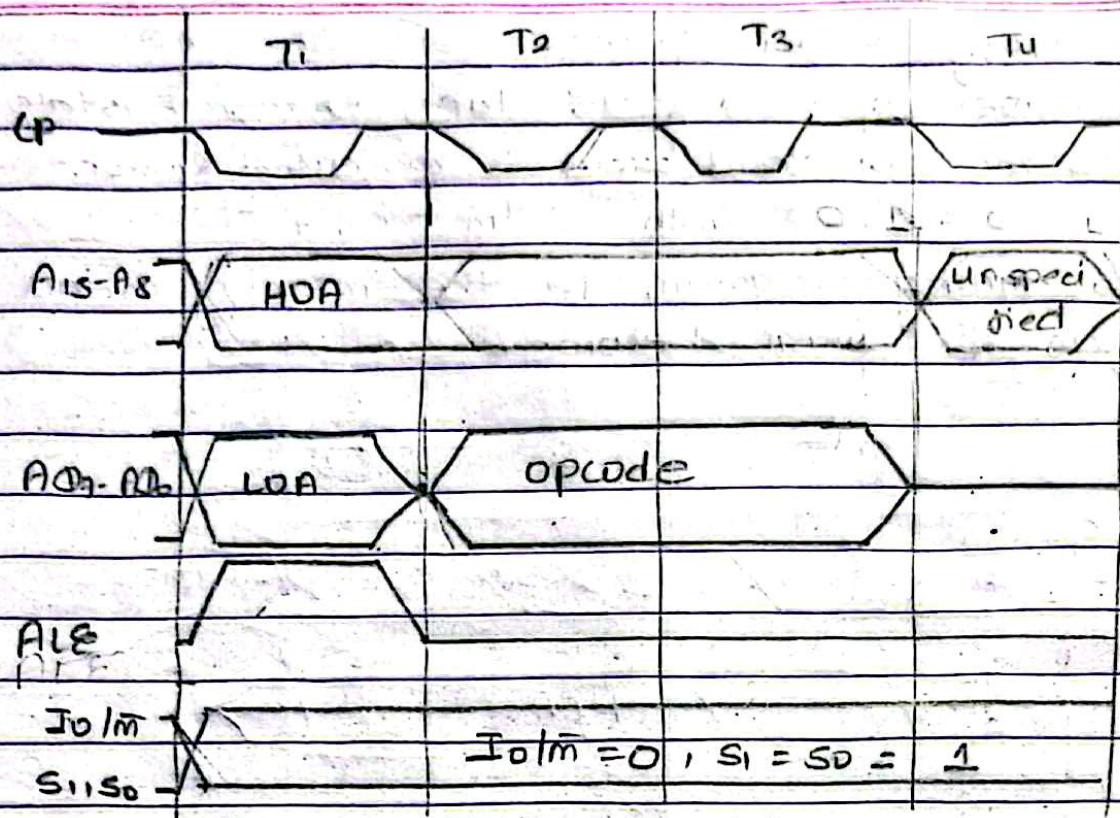


Fig: Timing diagram of opcode fetch diagram

How many clock cycle is required to complete the operation ADD B. If the mp is operating at 5MHz how what is the time required to complete the operation?

→ $f = 5\text{MHz}$

Now,

$$\text{Time period (T)} = \frac{1}{f} = \frac{1}{5 \times 10^6}$$

$$= 0.2 \mu\text{s}$$

4 clock cycle is required to complete the operation ADD B.

Hence, the time required to complete operation

$$4T = 4 \times 0.2 = 0.8 \mu\text{s}$$

Q.1.1. T-state:

The T-state defined as the sub division of the machine cycle i.e. synchronise with the system clock. Given by,

$$T = \frac{1}{f}$$

Where, f = Frequency

T = Time period.

Q.1.2

Opcode fetch cycle:-

→ The ~~first~~ operation in any instruction is the opcode fetch in which the opcode of the instruction i.e. pre-defined in the instruction register, it is read from the memory.

→ It takes 4 to 6 T state depending on whether the operation is performed on the register or register pair.

→ Except those instructions whose operation is performed on the register pair. They all take four T state and those instructions whose operation is based on register pair takes 6 T state.

→ The T-state four, five, six are undefined.

The timing diagram of opcode fetch is shown below:-

Status and Control signals :-

Status signal			Control signal	Operation
S_1	S_0	$I/O/\overline{M}$	$\overline{A\overline{R}}$	
1	1	0	$\overline{R\overline{O}}$	opcode fetch
1	0	0	$\overline{R\overline{O}}$	Memory Read
0	1	0	$\overline{W\overline{R}}$	Memory write
1	0	1	$\overline{R\overline{O}}$	I/O Read
0	1	1	write, $\overline{W\overline{R}}$	I/O write

Machine Cycle :-

It is defined as the sub-operation perform in μp . The different machine cycle are:

opcode fetch

memory read

memory write

I/O read

I/O write

Instruction cycle :-

$IC = \text{fetch cycle} + \text{execute cycle}$

$IC = FC + EC$

The time taken by the instruction to complete its operation is called an instruction cycle. It comp is composed a opcode fetch cycle and execute cycle.

As given by,

$$IC = FC + EC$$

→ For every instruction and every machine cycle, the first T state is for the IOA and for 2nd & 3rd T state there is data (IOA-DO).

ALE (Address Latch Enable):-

$AD_7 - AD_0 = \text{IOA} \mid \text{ALE} = 1$

$= \text{Data} \mid \text{ALE} = 0$

→ This is status pin based on the values of which the mp distributes whether AD_7 to AD_0 is IOA or data.

→ If ALE is high the value $AD_7 - AD_0$ carries IOA. If ALE is 0, the value AD_0 to AD_7 carries data.

→

$\overline{RD}$ (Read):-

→ This is an active low control signal. When this signal is active and I/O operation is performed i.e. the data is entered from peripheral device to mp.

→ The $\overline{RD}$ is active low at T₂ and T₃ of every instruction and every machine cycle.

→

$\overline{WR}$ (Write):-

→ This is an active low control signal when this signal is active and O/P operation is performed.

→ This edge is active for the second and third T state for every machine cycle for every instruction.

Information of Timing diagram:-

⇒ Higher order (Byte) Address (HOA):-

16-bit Address of 8085 μp : $A_{15} A_{14} A_{13} A_{12} A_{11}$

$A_{10} A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0$

where,

$A_{15} - A_8 \Rightarrow$ HOA (Higher order Address)

$A_7 - A_0 \Rightarrow$ lower order Address (LOA)

$A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0$

$D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0$

$A_0 A_1 A_2 A_3 A_4 A_5 A_6 A_7$

(Multiplexed Address / Data bus bit line)

Higher order address:-

8085 microprocessor has 16-bit address represented as A_{15} to A_0 . Among these A_8 to A_{15} is the higher order address and A_7 to A_0 is the lower order address.

→ The higher order address has three T state.

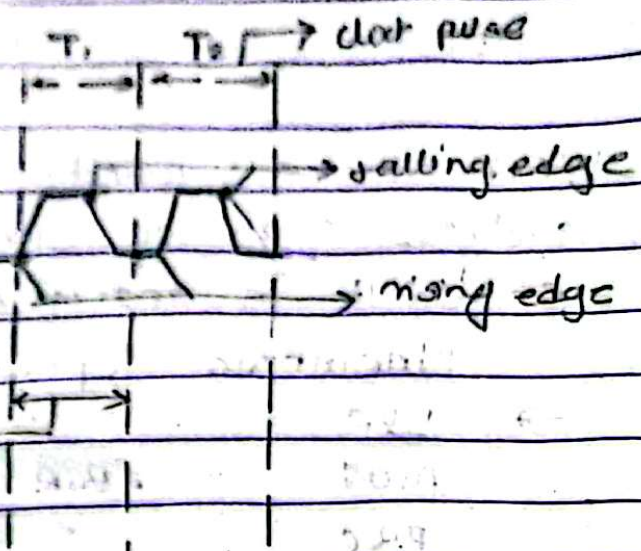
→ For every instruction for every machine cycle

Multiplexed address / Data bus :- (A_0 to A_7)

→ These are 8-bit bidirectional bus. It means that the data can flow in two ways.

→ The multiplexed address / data bus carries 8-bit data (D_7 to D_0) or lower address (A_7 to A_0).

Timing diagram:



$$f_1 = 1 \text{ MHz}$$

$$f_2 = 1 \text{ MHz}$$

$$T_1 = \frac{1}{f_1}$$

$$T_2 = \frac{1}{f_2}$$

$$= \frac{1}{10^6}$$

$$= \frac{1}{10^6}$$

The graphical representation of the operation of an instruction is called timing diagram -

It includes different machine cycles such as

- i. Opcode phase fetch (fetch cycle)
- ii. Memory Read
- iii. Memory Write
- iv. I/O Read
- v. I/O Write
- vi. Halt

Mnemonic	operand
MOV	D, R
ADD	C
INC	D
JNR	A
DEC	B
etc.	

4. Register Indirect in addressing mode:-

In this addressing mode, the operand field is the indirect address register whose pair whose contain content # will be the effective address. All instruction in this addressing mode are one byte instruction. some of the example are listed below:-

Mnemonic	operand
STAX	B
STAX	D
LDA	B/D
MOV	A, M
ADD	M
SUB	M

Inherent addressing mode:-

In this addressing mode, the operand is implicitly defined (inherent). All the instruction

Mnemonic	operand
MVI	Rg, 8-bit data
ADI	8-bit data
SUI	8-bit data
ANI	8-bit data
CPJ	8-bit data
LXI	Rg. pair, 16-bit data

e) Direct Addressing:-

In this addressing mode direct address is specified in the operand field. The address may be either in 8-bit or 16-bit, some of the examples of direct addressing are listed below:-

Mnemonic	operand
IN	8-bit port address
OUT	8-bit port address
LDA	16-bit address
STA	16-bit address
JC	16-bit address
JNC	16-bit address
JNZ	16-bit address
etc.	

Register direct addressing mode:-

In this addressing mode the operand field is only in register / registers. Therefore all the instructions in this addressing mode are a one byte instruction.

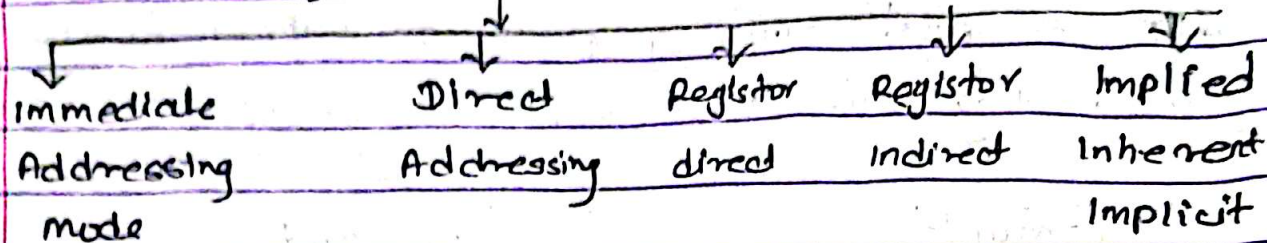
STA

HLT

2023H

V.V.M.P.

Addressing Modes of 8085 μ p



Addressing Mode.

The way of specifying the operand in an instruction is called an addressing mode. In 8085 μ p there are different five types of addressing mode.

1. Immediate Addressing mode

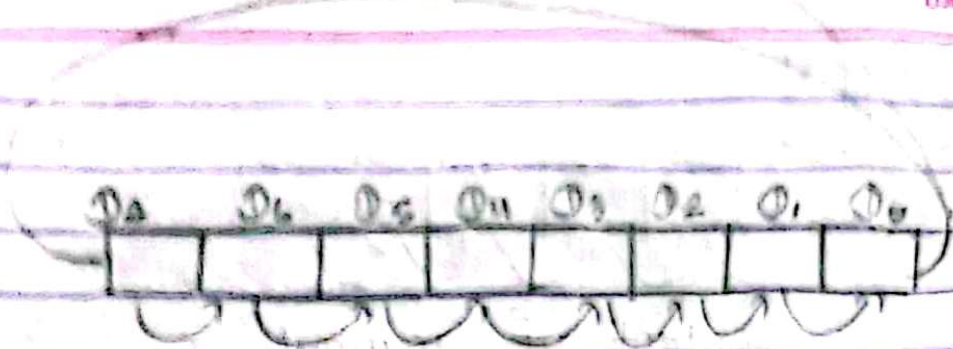
In this addressing mode directly data is defined in the operand field. The data may be 8-bit or 16-bit. Some of the examples are given below:-



Scanned with CamScanner

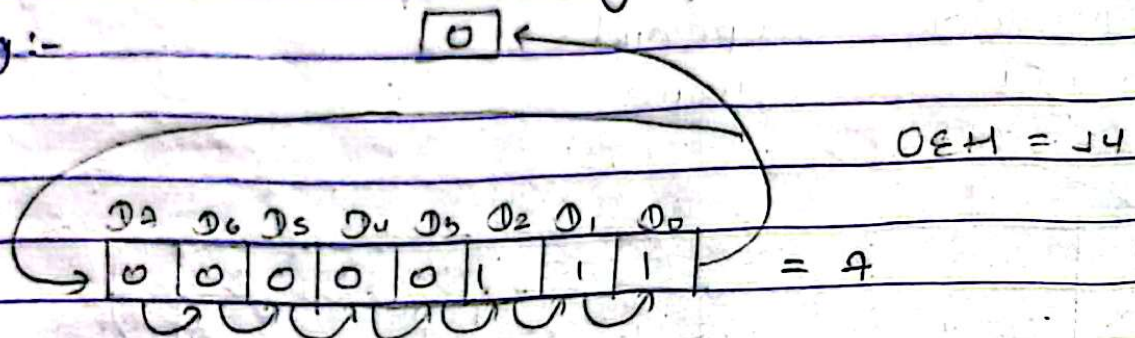
PAGE

--	--	--



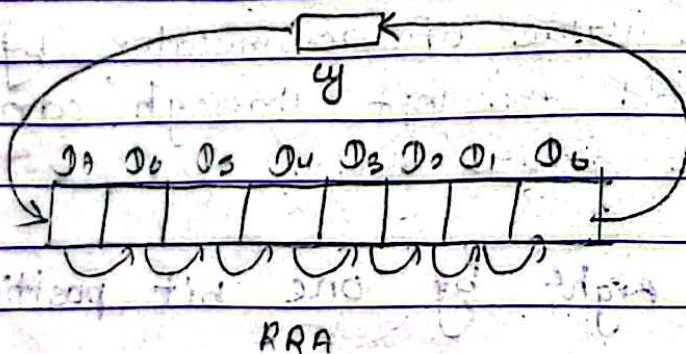
It rotate value of accumulator towards the right by one bit position as shown in the figure above. The result of rotation corresponds to div. divide by two.

eg:-

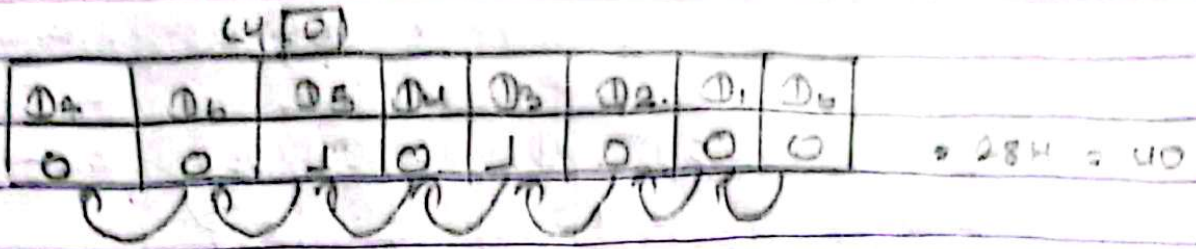


Rotate acc. ^{RRC} right by one bit position through carry.

Mnemonic operand
RRC RAR none.



It rotate the content of accumulator towards the right by one bit position through carry as shown in the figure above.

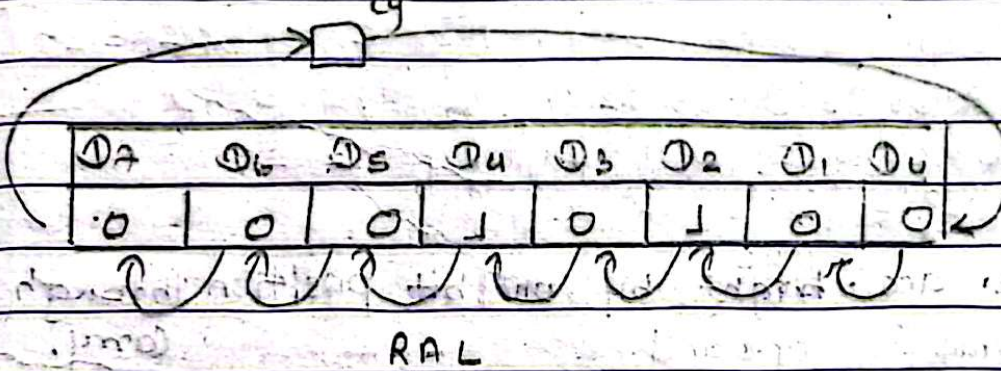


It rotate the content of accumulator towards the left by one bit position as shown in the figure above. The result of RLC is correspond to multiply by 2.

Rotate Acc. left by one bit position through carry

Mnemonic operand

RAL none



It rotate the value of accumulator by one bit position toward the left through carry flag. as shown in the figure above.

Rotate Acc. right by one bit position.

Mnemonic operand

RAC None

ORI	8-bit data	$A \leftarrow A \vee \text{8-bit data}$
XRA	Rg1M	$A \leftarrow A \oplus (Rg1M)$
XRI	8-bit data	$A \leftarrow A \oplus \text{8-bit data}$
CMA	None	$A \leftarrow \bar{A}$

Additional logical Instruction:-

Rotating Instruction

Comparing Instruction

Rotating Instruction

All the rotating instructions rotate the value of accumulator either towards the left or towards the right with or without carry by one bit position. There are four types of Rotating instruction. All the rotating instruction are one byte instruction.

1. Rotate ACC. left by one bit position:

Mnemonic Operand
RLC None

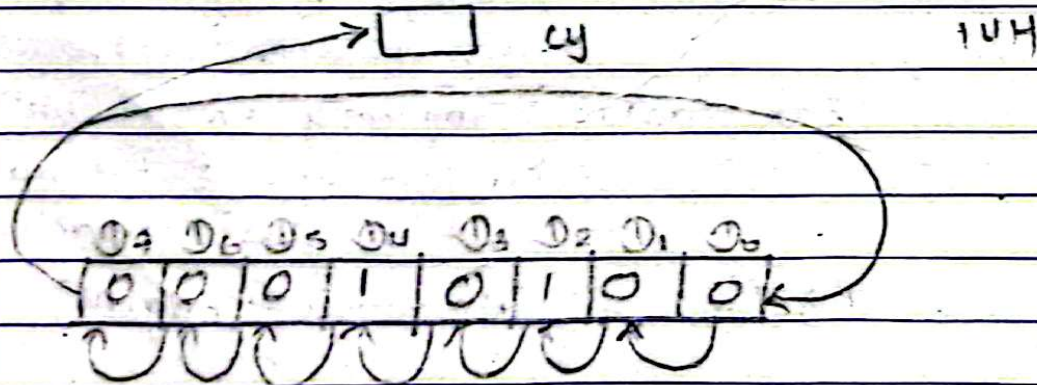


Fig: RLC

ORI	8-bit data	$A \leftarrow A \vee \text{8-bit data}$
XRA	Rg1M	$A \leftarrow A \oplus (Rg1M)$
XRI	8-bit data	$A \leftarrow A \oplus \text{8-bit data}$
CMA	None	$A \leftarrow \bar{A}$

Additional logical Instruction:-

Rotating Instruction

Comparing Instruction

Rotating Instruction

All the rotating instructions rotate the value of accumulator either towards the left or towards the right with or without carry by one bit position. There are four types of Rotating instruction. All the rotating instruction are one byte instruction.

1. Rotate ACC. left by one bit position:

Mnemonic Operand
RLC None

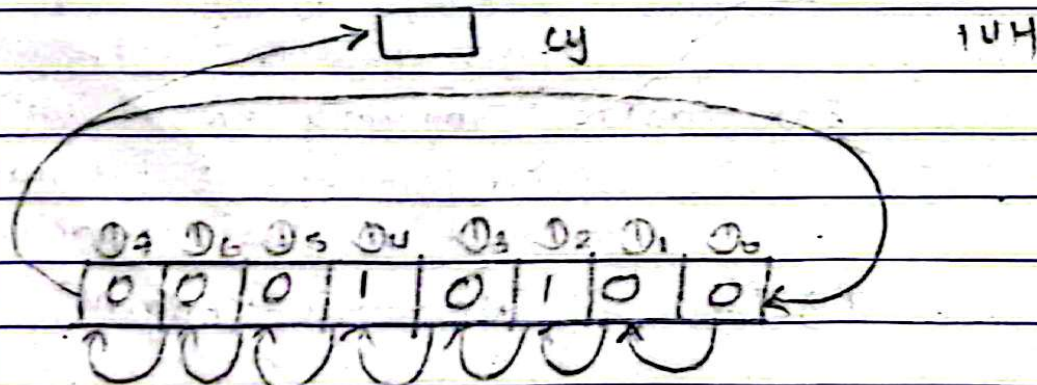


Fig 1: RLC

Mnemonic	Operand	Description
ADD	Rg1M	$A \leftarrow A + Rg1M$
ADJ	8-bit data	$A \leftarrow A + 8 \text{ bit data}$
SUB	Rg1M	$A \leftarrow A - Rg1M$
SUBJ	8-bit data	$A \leftarrow A - 8 \text{ bit data}$
INR	Rg	$Rg = Rg + 1$
DEC	Rg	$Rg = Rg - 1$
ADC	Rg	$A = A + Rg + Cy$

Logical instruction:-

All the logical operation are perform on the accumulator for the logical operation to be perform at least one data must be available in accumulator and the final result of the logical operation is by default store in A. Based on the result of logical operation perform the flag status get modify. Some of the logical instruction are given below.

PC-12/11

Mnemonic	Operand	Description
ANA	Rg1M	logically ANDed bet ⁿ A and content of 'Rg1M' specified in the operand field.
AND	8-bit data	logically ANDed bet ⁿ A and content of 8 bit data specified in the operand.
ORA	Rg1M	$A \leftarrow A \vee (Rg1M)$

LD	16-bit data	It loads register value by the 16-bit data specified in the operand field.
IN	8-bit port Address	It inputs 8-bit data from input port whose address is specified in the operand field.
OUT	8-bit port address	It outputs the value of A into I/O port whose address is specified in the operand field.

Arithmetic Instruction:-

All the arithmetic instructions perform arithmetic operation such as addition, subtraction, increment, decrement. For the arithmetic operation to be executed at least one data must be available in the accumulator and the final result of arithmetic operation is by default stored in 'A'. Based on the result of arithmetic operation, the flag status gets modified. Some of the examples of arithmetic operations are listed below:-

B C $\rightarrow$ B
 D E $\rightarrow$ D
 H L $\rightarrow$ H

DATE

Instruction Based on operation

1. Data Transfer Instruction:-

In this group of instruction the data is transfer from source to destination without modifying the content. The transfer may be betⁿ the registers themselves or betⁿ accumulator and memory or betⁿ accumulator and IO devices.

The instruction in this group either 1 byte, or two byte or three byte. The status of flag is unaffected. some of the example of data transfer instruction are given below:-

Mnemonic	operand	Description
MOV	Rd, Rs	$R_d \leftarrow R_s$
MVI	Rg/m, 8-bit data	$R_g/m \leftarrow 8\text{-bit data}$
STA	16-bit address	$16\text{-bit address} \leftarrow A$
LDA	" " " "	$A \leftarrow 16\text{-bit address}$
STAX	Rg pair	Store the content of A in the memory address which is the content of register pair.
LDAx	Rg pair	Load the content of memory address which is hold by register pair specified in the operand into accumulator.

PAGE