PEC-2024

DATE / / PAGE

1a) Define microprocessor along with its applications. Differentiate Harvard and Von-Neumann architecture on the basis of storage.

3) A microprocessor is a central processing unit contained on a single integrated circuit or chip that are designed to execute instructions and manipulate data according to the instructions provided by the computer programs.

Applications of microprocessor are:

1) They are used to build personal computers.
2) They are used in embedded systems such as ~~wash~~ microcontrollers for controlling devices like washing machines, microwave ovens, and automotive systems.
3) They are used in consumer electronics such as: smartphone, digital computers, etc.

Harvard Architecture	Von-Neuman architecture
1) Both program instructions and data are ~~stored share the same~~ have seperate memory spaces.	1) Both program and data share the same memory.
2) Multiple buses are used by CPU to access instruction & data.	2) Instructions & data are accessed by using single bus.
3) It is faster in execution comparatively.	3) It is simple and flexible for ~~simple~~ general purpose computing.
4) Drawbacks: more complex and less flexible.	4) Drawbacks: slower performance.

DATE / / PAGE

1b) What are the types of instructions depending on the word size? Explain different types of addressing modes of 8085 microprocessor.

⇒ The types of instructions depending on the wordsize are:

~~1) 8-bit microproce~~

1) 1-byte instructions: These instructions ~~are~~ represent the opcode and the operand in one byte.

Eg: MOV B, A

2) 2-byte instructions:- Two byte instruction is the type of instruction in which the first 8 bit indicates the opcode and the next 8 bits indicate the operand.

Eg: MVI A, 32H

3) 3-byte instructions:- 3 byte instruction is the type of instruction in which the first 8 bits indicates the opcode and the next two bytes specify the 16-bit address. The lower address is represented in second byte and high-order address is represented in the third byte.

Eg: LDA 2050H

Types of addressing modes are:

1) Immediate Addressing: In this mode, the operand is specified directly in the instruction. For eg: MVI A, 10H.

DATE / / PAGE

2) Direct Addressing: In this mode, the operand is the address of the data in memory. Eg: LDA 2050H.

3) Register Addressing: The operand is contained in one of the registers of microprocessor. Eg: ADD B

4) Indirect Addressing: The address of the operand is stored in a register or memory location, and the actual operand is located at that address. For eg: MOV A, M.

5) Indexed Addressing: In this mode, an index register is added to a base address to generate the effective address of the operand. For eg: MOV A, [HL]

20) Write an ALP in 8085 to check whether the number stored in memory location 2060H is prime or not. If the number is prime store FFH in memory location C00FH else store 00H.

```
MVI B,02H
LXI H,2060H.
MOV A,M
CPI 01H.
JZ NOT_PRIME
CPI 02H
JZ IS_PRIME

CHECK_DIVISOR:
        MOV C,A
```

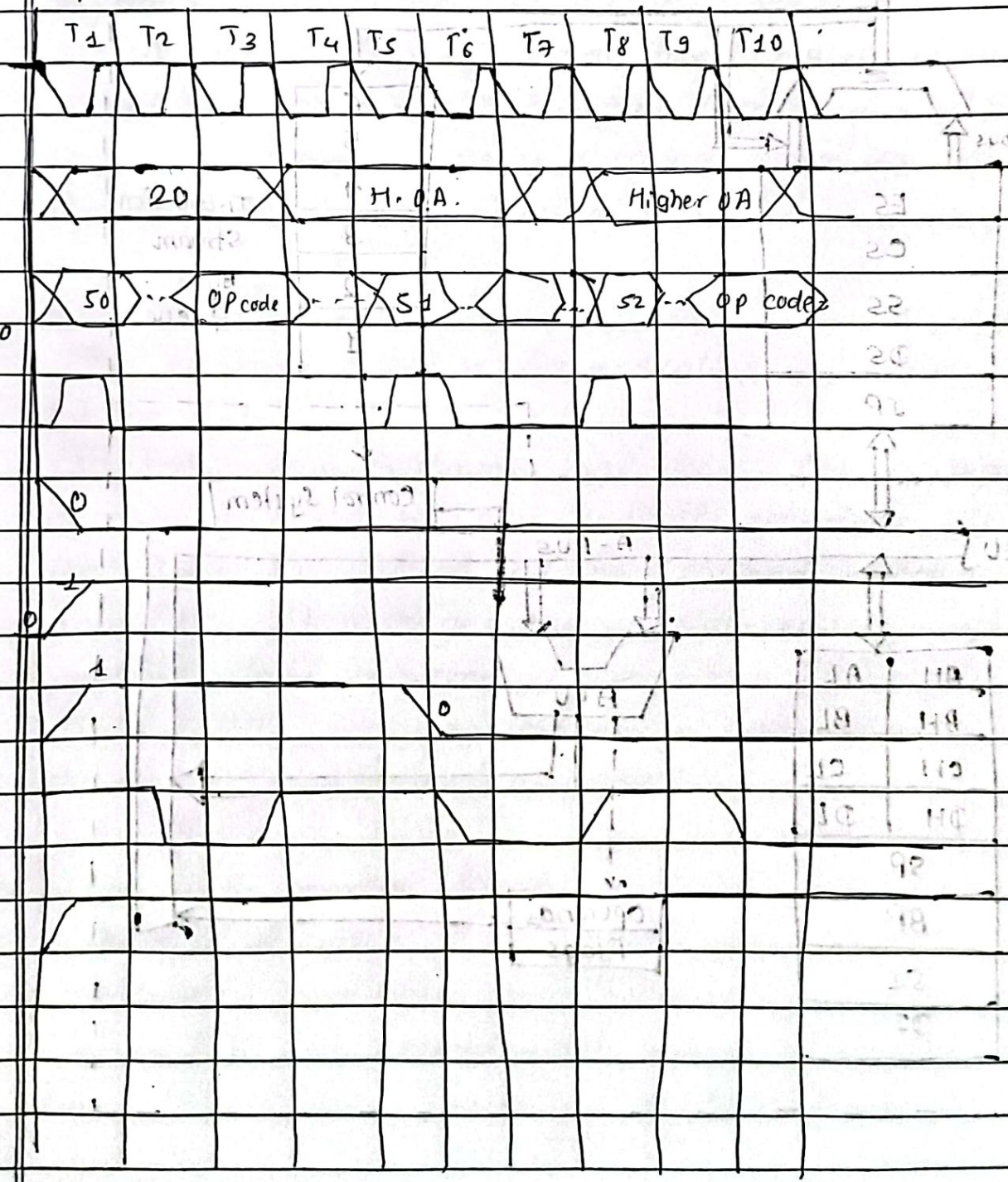
DATE

```
MOV A,B
CALL DIVIDE
JZ NOT_PRIME
INX B
MOV A,B
CPI 40H
JNC IS_PRIME
JMP CHECK_PRIME
IS_PRIME:
        MVI M,FFH
        JMP END
NOT_PRIME:
        MVI M,00H
END:
    HLT
DIVIDE:
      MOV D,A
       XRA A
DIV_LOOP:
        CMP D
        JNC DIV_DONE
        SUB D
        INR A
        JMP DIV_LOOP
DIV_DONE:
      RET
```

DATE / / PAGE

2b) Define T-state. Draw the labelled timing diagram of instruction LXI H, 2050H.

a) T-state is one time period of frequency of microprocessor.



DATE / / PAGE

3a) Draw a well labelled architecture of 8086 microprocessor and discuss about BIU and EU.

⇒

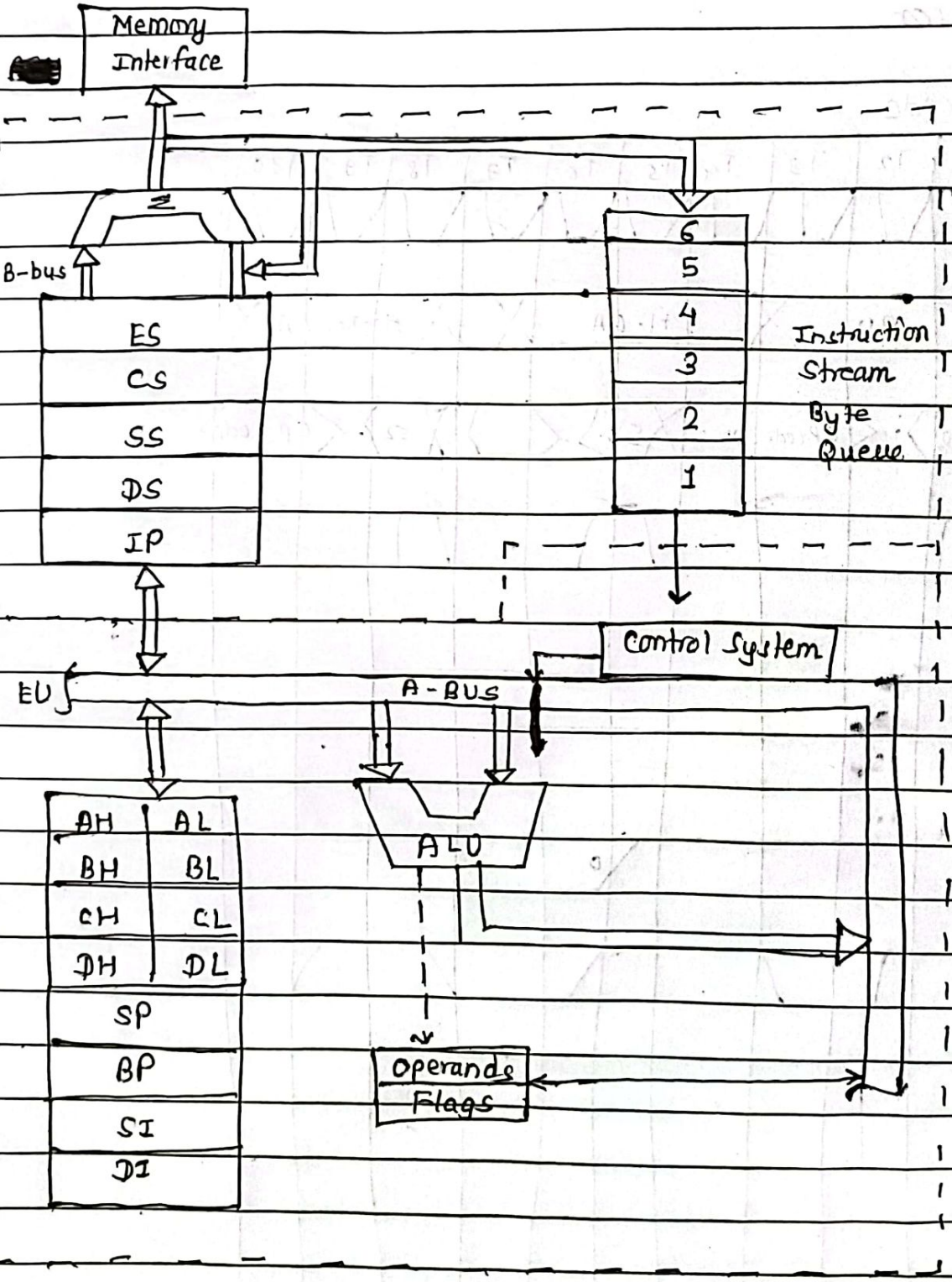


fig:- 8086 microprocessor architecture

DATE / / PAGE

- BIU

The BIU is responsible for interfacing with the system bus. It ~~handles~~ handles fetching instructions and data from memory or I/O devices. BIU manages the address bus for memory access and generates the necessary control signals. It also handles the segmentation mechanism, which allows accessing more than 1MB of memory. BIU prefetches the next instruction from memory while the current instruction is being executed, improving performance through instruction prefetching.

- EU

The EU is responsible for executing instructions fetched by the BIU.
It contains the Arithmetic Logic Unit (ALU) responsible for performing arithmetic and logical operations. EU decodes the instructions fetched by the BIU and executes them accordingly. It manages the general-purpose registers and performs data movement operations. EU controls the flow of instructions, including branching and looping, based on the results of executed operations.

3.b) What are assembler directives? Explain any six different assembler directives of 8086 microprocessor.

⇒ Assembler directives are instructions or commands that are used by an assembler to control the assembly process. Any six assembler directives of 8086 are:-

DATE / /

1) ORG : This directive specifies the starting address for the code or data segment. It tells the assembler where to place the following instructions or data in memory.

2) DB (Define Byte) :- Used to define byte-sized data in memory. It allocates memory space for one or more bytes and initializes them with specified data values.

3) DW (Define Word) :- It is similar to DB but for word-sized data (2 bytes on the 8086). It allocates memory space for one or more words & initializes them with specified data value

4) DS (Define Space) :- Reserves a block of uninitialized mema bytes or words. It allocates memory sp ~~for one or more words and initializes them with specified data values.~~ without initializing them with specif data values.

5) END : It marks the end of source file. It indicates to the assembler that it has reached the end of th source code and should stop assembling.

6) SEGMENT : It marks the start of a segment.

DATE / / PAGE

Q) Write an 8086 program to enter a string from the keyboard. Count the no. of repetitions of letter 'a' or 'A'. If the count is even, display "POKHARA" else "UNIVERSITY".

- MODEL SMALL
- DATA DB
  MSG1 "Enter a string: $"
  MSG2 DB 0DH, 0AH
  MSG3 DB 0DH, 0AH
  BUFFER DB 80 DUP('$').
  COUNT DB ?

- CODE
  MAIN PROC
  MOV AX, @DATA
  MOV DS, AX
  MOV AH, 09H
  LEA DX, MSG1.
  INT 21H.

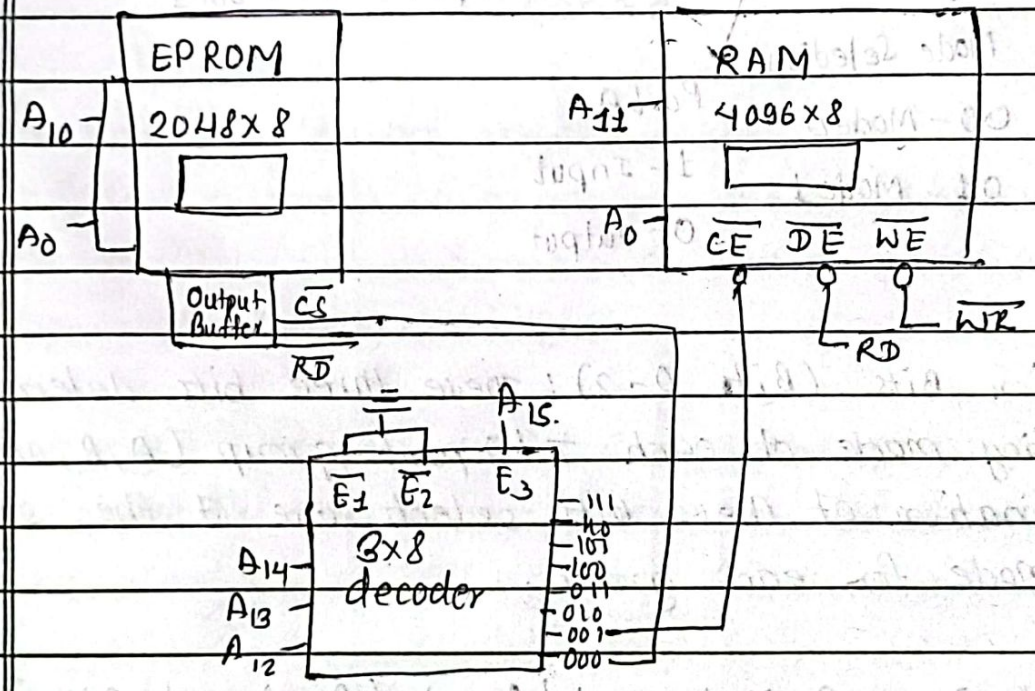
  MOV AH, 0AH
  MOV DX, OFFSET BUFFER.
  INT 21H.
  MOV SI, OFFSET BUFFER +1.
  MOV CX, 0.
  COUNT_LOOP:
  MOV AL, [SI]
  CMP AL, 0DH
  JE END_COUNT

```
CMP AL, 'A'
JE INCREAS INCREMENT_COUNT
INC SI
JMP COUNT_LOOP

INCREMENT_COUNT:
    INC CX
    INC SI
    JMP COUNT_LOOP
END_COUNT:
        MOV COUNT, CL
        MOV AL, COUNT
        AND AL, 01H
        JZ EVEN_COUNT
        JMP ODD_COUNT

EVEN_COUNT:
        MOV AH, 09H
        LEA DX, MSG2
        INT 21H
        JMP END_PROGRAM

ODD_COUNT:
        MOV AH, 09H
        LEA DX, MSG3
        INT 21H
        JMP END_PROGRAM
```

DATE / / PAGE

```
END_PROGRAM:
    MOV AH, 4CH
    INT 21H
MAIN ENDP
END MAIN
```

4b) Design an address decoding circuit to interface one ROM chip of 2 KB and one RAM chip of 4 KB at address 0000H and E000H respectively.



Address Range.

FOR ROM

	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
0000H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
07FFH	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

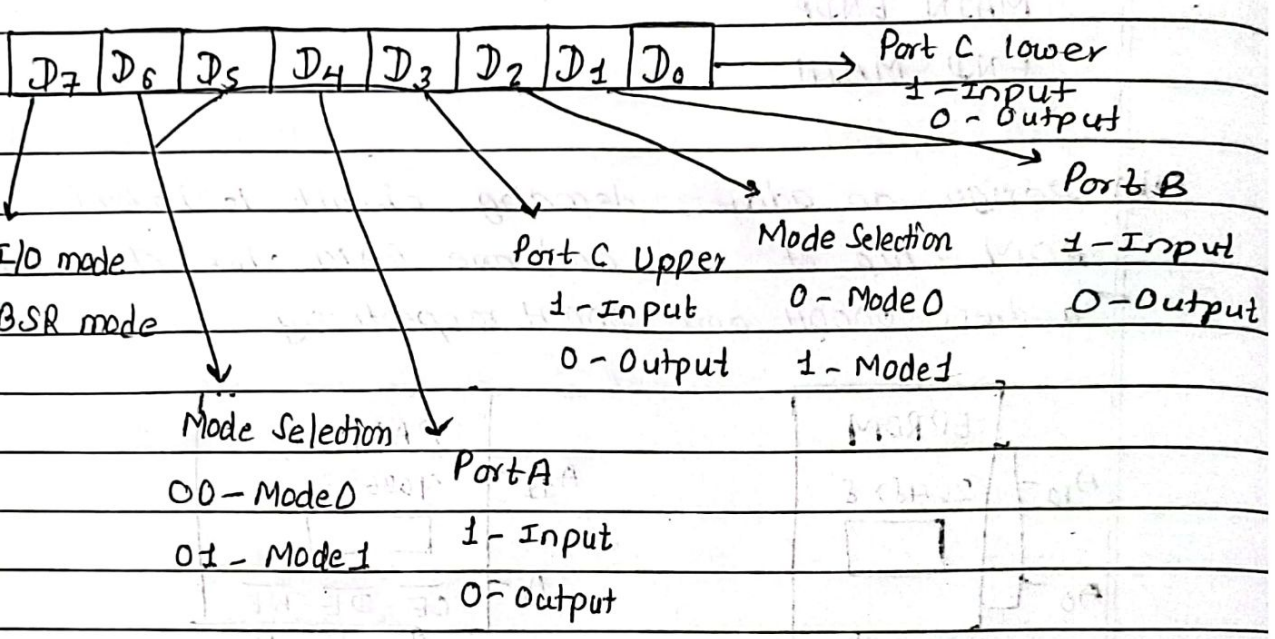
FOR RAM

	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
E000H	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
E7FFH	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1

DATE / / PAGE

5a) Draw and explain control word for 8255A PPI. Also, explain the different operating modes of 8255 PPI

⇒



- Mode Selection Bits (Bits 0-2): These three bits determine the operating mode of each I/O port group (A, B, and C). The combination of these bits selects one of the six possible mode for each group.

- Group A and Group B Mode set/Reset (Bit 3 and Bit 4): These bits allow you to individually set or reset the operating mode for Group A and Group B. Depending on the mode selected by the mode selection bits, these bits enable or disable certain features or functionalities associated with each groups.

- C0, C1 and C2 Bits (Bits 5-7): These bits, also known as control bits, are used for additional mode selection

DATE / / PAGE

or control. Their interpretation depends on the mode selected by the mode selection bits. In some modes, they may define additional parameters or functionalities.

Different operating modes.

1) Mode 0 - Simple Input/Output Mode (BIBO).
Both group A and Group B can be configured as input or output independently. No special control signals are needed for data transfer.

2) Mode 1 - Strobed Input/Output Mode:
Data transfer occurs only when a strobe signal is applied. Both Group A and Group B can be configured as input or output independently.

3) Mode 2 - Bi directional Bus Mode:
Both Group A and Group B act as bi-directional data bus lines.

4) Mode 3 - Combination of Simple and Strobed I/O:
One group operates as simple I/O, while the other group ~~act~~ operates with strobed input/output.

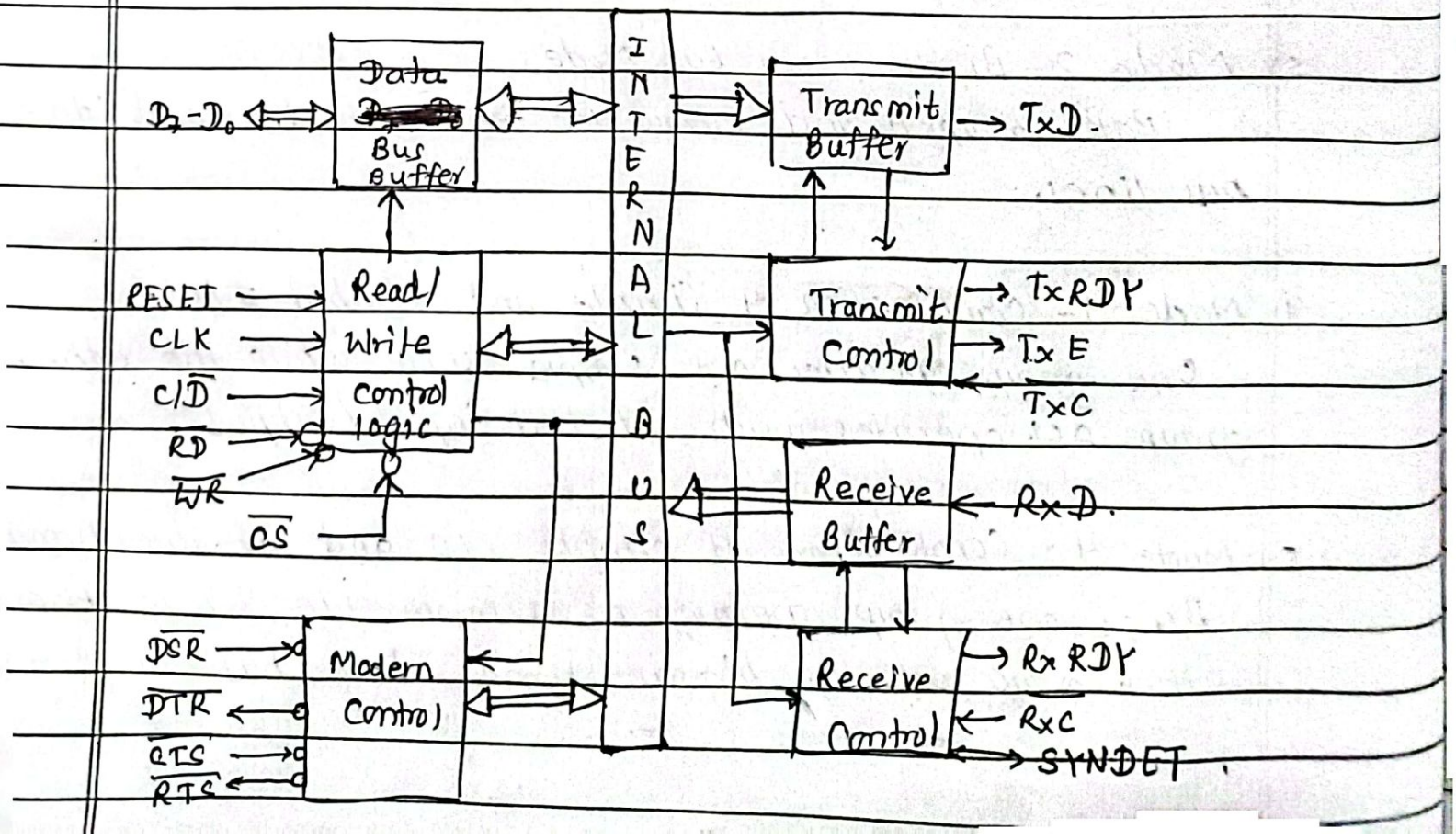
5) Mode 4 - Combination of simple I/O and Bi-directional Bus: one group operates as simple I/O, while the other group acts as bi-directional data bus.

DATE / / PAGE

Mode 5 - Combination of Strobed I/O and Bi-Directional Bus: one group operates with strobed input/output, while the other group acts as a bidirectional data bus.

5b) What is synchronous serial data communication? With the help of block diagram explain the working of 8251 USART.

⇒ Synchronous serial data communication is a method of transmitting data where the sender and receiver use a shared clock signal to synchronize the timing of data transmission and reception. In this method, data is sent bit by bit over a single communication channel.



DATE / / PAGE

8251 USART works with following operations:

(1) Transmission:
Microprocessor writes data to the 8251's data Register.
8251 converts parallel data to serial and transmits it out of the TX pin.

(2) Reception:
8251 receives serial data at the RX pin.
It converts serial data to parallel and stores in its data register.

(3) Control and Status:
Microprocessor configures operation via control registers.
Status registers provides information like data availability.

(4) Handshaking:
Supports handshaking signals for flow control between devices.

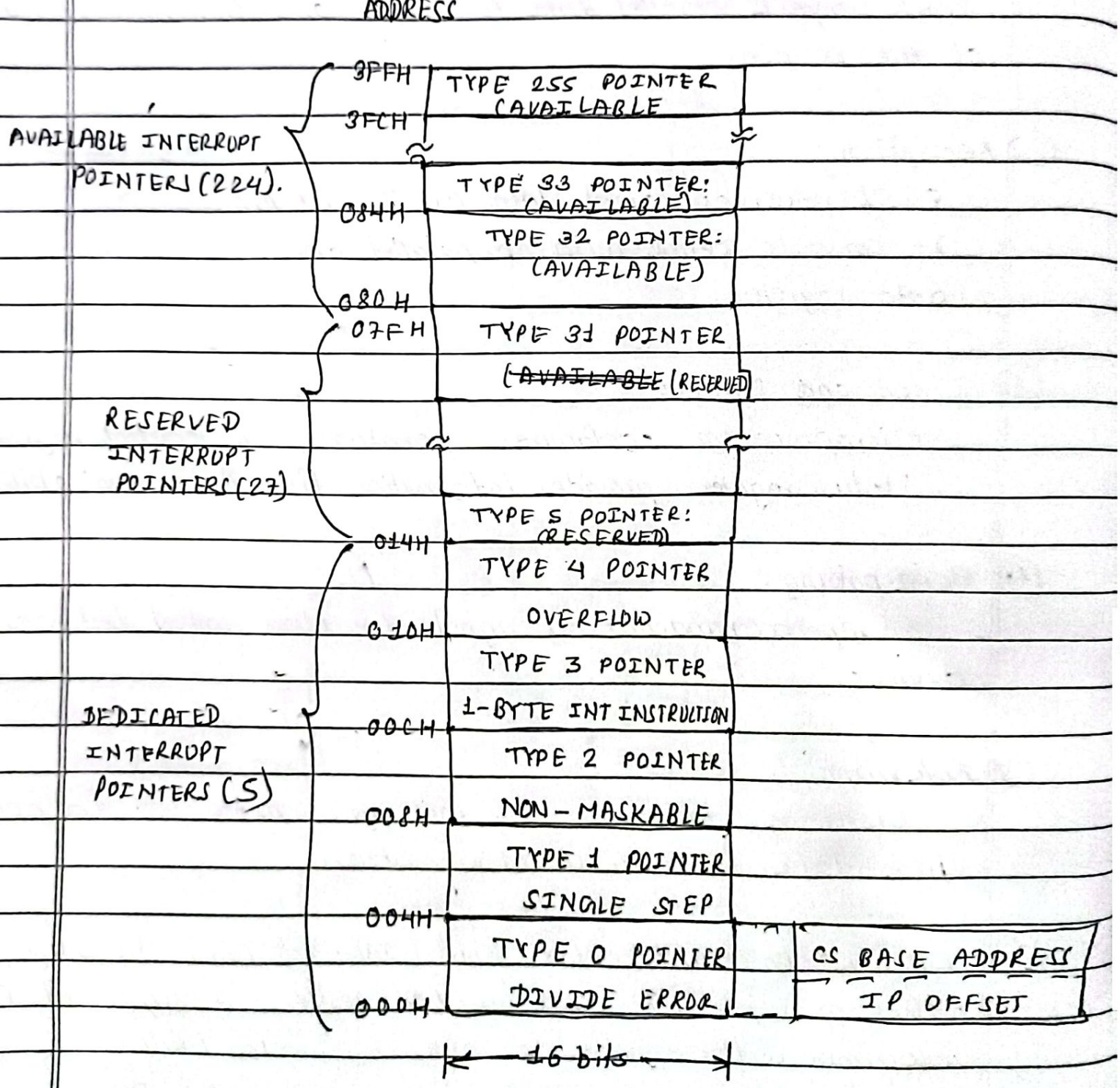
(5) Interrupts
Generates interrupts to the microprocessor for events like data transmission/reception.

6a) What is Interrupt Vector Table (IVT)? Draw the IVT for 8086 microprocessor and explain different types of 8086 interrupts with respect to interrupt vector table.

⇒ The Interrupt vector Table is a data structure used in microprocessor-based systems to store addresses of

DATE / / PAGE

interrupt service routines (ISRs), allowing the processor to quickly locate and execute the appropriate code when an interrupt occurs.



DATE / / PAGE

Hardware Interrupts (External Interrupts):

These interrupts are generated by external hardware devices connected to the microprocessor. Example include keyboard interrupts, timer interrupts, disk I/O interrupts and peripheral device interrupts.

Software Interrupts (Internal Interrupts):

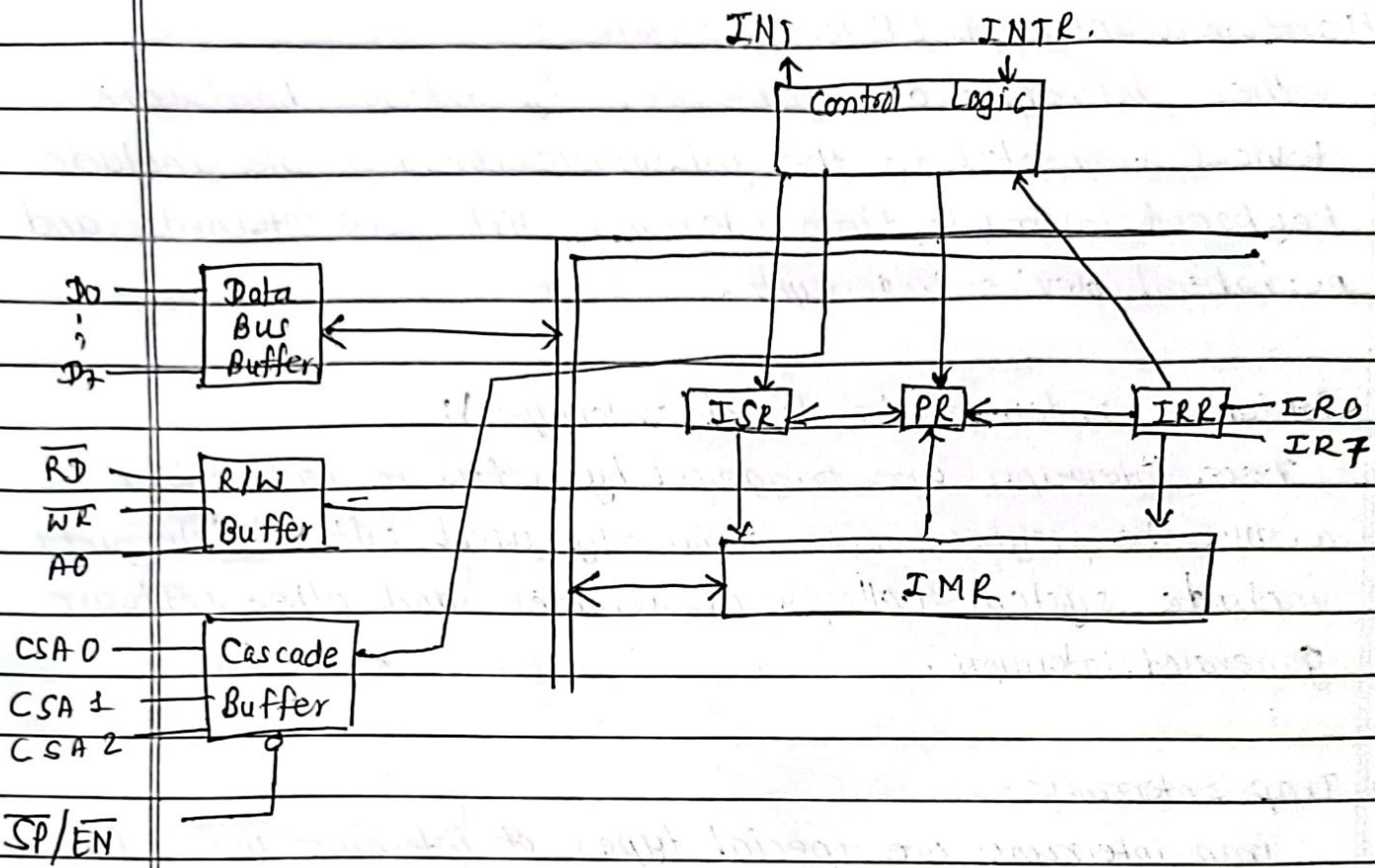
These interrupts are triggered by software instructions within the program code. Commonly used software interrupts include system calls, BIOS services, and other software generated interrupts.

Trap interrupts:-

Trap interrupts are special types of interrupts used for error handling or debugging purposes. They are similar to software interrupts but are typically used for non-maskable situations, such as illegal instructions or division by zero errors. They are located at specific addresses within the IVT to handle these exceptional conditions.

What is interrupt? Draw well labelled architecture of Programmable Interrupt Controller (PIC) Intel 8259 and explain its working.

An interrupt is a signal sent by a hardware device or generated by software that temporarily halts the normal execution of a program by a microprocessor or microcontroller. Interrupts are used to handle events that require immediate attention, such as input/output operations, timer events, or errors.



- Data Bus Buffer : This tri-state, bidirectional 8-bit buffer is used to interface 8259A with the system data bus. Control words and status information are transfered through data bus buffer.

- Read/Write Control Logic : This function of this block is to accept commands from CPU and control the device. It contains the initialization command word register (ICW) and operational command register (OCW) to store various control formats for device operation.

- Status Registers :- Interrupt Request Register, In-Service Register, Interrupt Mask Registers are used to indicate the active interrupt request inputs, contain the level of

DATE / / PAGE

interrupt being serviced and store the mask bits & non-masked bits respectively.

- Priority Resolver: This block determines the priority of bits set in IRR.

- Cascade Buffer.
  This block is used for cascading multiple 8259s to handle more than 8 interrupts. $CAS_2 - CAS_0$ cascade lines are used as output from master to slave 8259s. SP/$\overline{EN}$ pin is used to identify master or slave 8259A. For master SP/$\overline{EN}$ = 1, for slave it's 0.

Q7) Write short notes on:

b) Polled and Vectored interrupt

Polled interrupts and vector interrupts are two different approaches used in microprocessor-based systems to handle interrupt requests efficiently.

Polled interrupts rely on the microprocessor continuously checking the status of a specific hardware device or condition to determine if an interrupt has occurred. This checking is typically performed in a loop within the main program. While simple to implement, polled interrupts can be less efficient as they require the processor to devote processing time to constantly monitoring the interrupt status. This can result in wasted processing cycles, especially if the device being polled doesn't require immediate attention.

DATE / / PAGE

In contrast, vector interrupts use a more structured approach. When an interrupt occurs, the hardware sends a signal to the microprocessor along with an interrupt vector number, which identifies the source of the interrupt. The microprocessor then uses this vector number to index into an IVT, which contains the addresses of interrupt service routines corresponding to the different interrupt sources. By consulting the IVT, the processor can quickly locate and execute the appropriate ISR without the need for constant polling. This approach is more efficient as it minimizes processing overhead and allows the processor to respond promptly to interrupt requests.

a) Memory mapped I/O vs I/O mapped I/O.

Memory Mapped I/O	I/O mapped I/O.
1) Address is 16 bit.	1) Address is 8 bit.
2) MEMR MEMW control signals are used.	2) IOR and IOW control signals are used.
3) Data transfer is bet^n any register and I/O device.	3) Data transfer is between accumulator and I/O device.
4) Decoding 16 bit address may require more hardware.	4) Decoding 8 bit address will require less hardware.
5) Instruction available are LDA, STA, LDAX, etc.	5) Instruction available are IN and OUT.

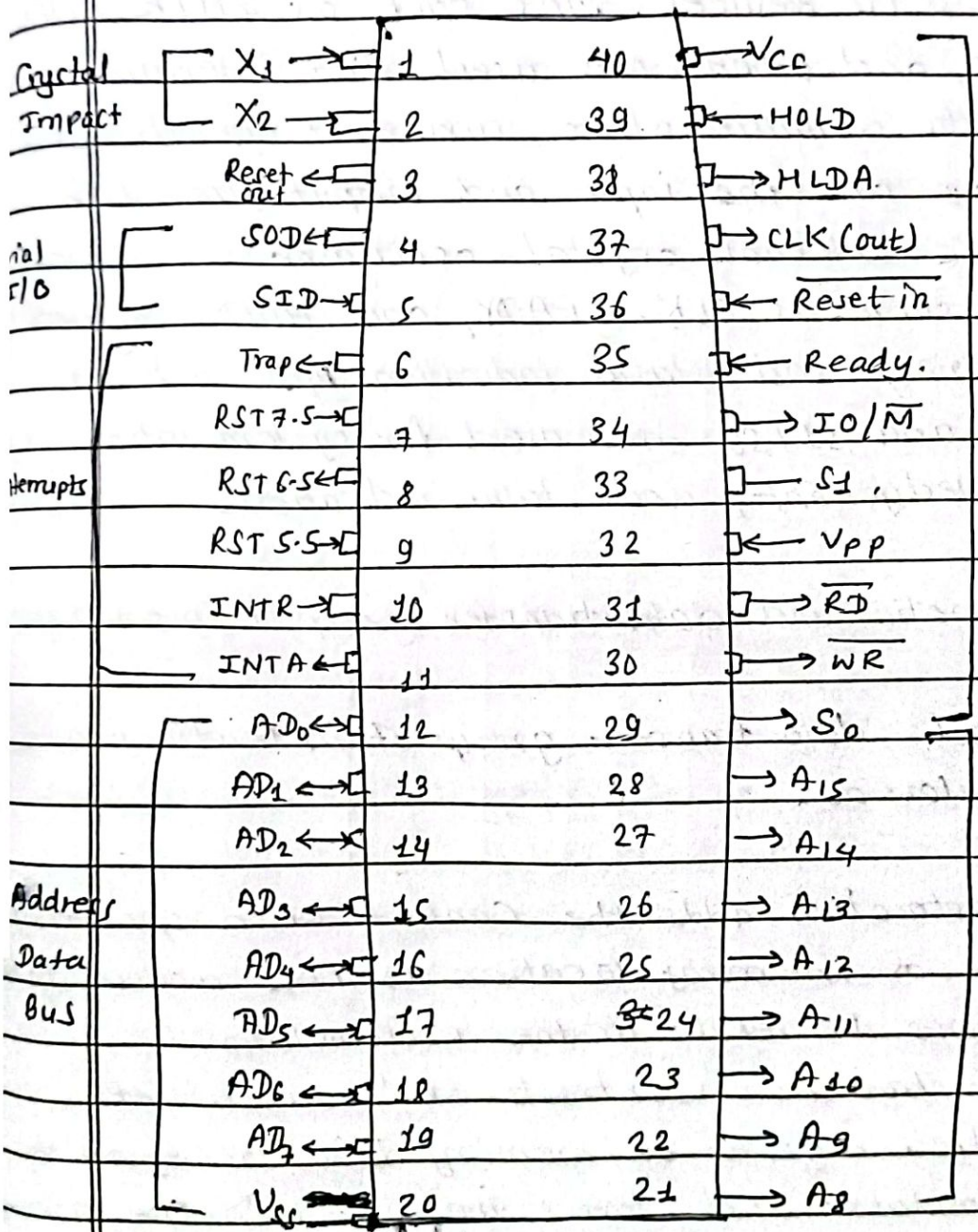
SOE - 2024

DATE / / PAGE

1) a) Differentiate Von Neumann and Harvard Architecture. (Done)

1b) Explain ~~block diagram~~ pin layout of 8085 microprocessor in detail.

(OR)



DATE / / PAGE

- Address Bus lines are pins D_0 to D_7, used for transmitting data betⁿ the microprocessor and memory or I/O devices. Pins such as RD, WR, ALE, IO/M, and other control signals used to control data transfer and communication with memory and I/O devices. Pins such as INTR, INTA, SID, SOD, RST, and others are used for interrupt handling, serial data communication, and reset operations. Pins X_1 and X_2 are the input and output pins for connecting an external crystal oscillator or clock source. Pins such as CLK, READY, and HOLD are used for clock timing and status indication. Pins such as RESET, HLDA, and S1/S0 are used for system initialization, hold acknowledgement, and status indication.

2a) Describe arithmetic and data transfer group of instructions in 8085.

⇒ Arithmetic and data transfer group of instructions are described below as:

1) ADD : This instruction adds the content of a specified register or memory location to the accumulator and stores the result in the accumulator.
2) SUB : This instruction subtracts the content of a specified register or memory location from the accumulator and stores the result in the accumulator.
3) Increment (INR) : This instruction increments the content of a specified register or memory location by one.

DATE / / PAGE

4) Decrement (DCR): This instruction decrements the content of a specified register or memory location by one.

5) MOV: the instruction copies the content of a specified register or memory location into another register or memory location.

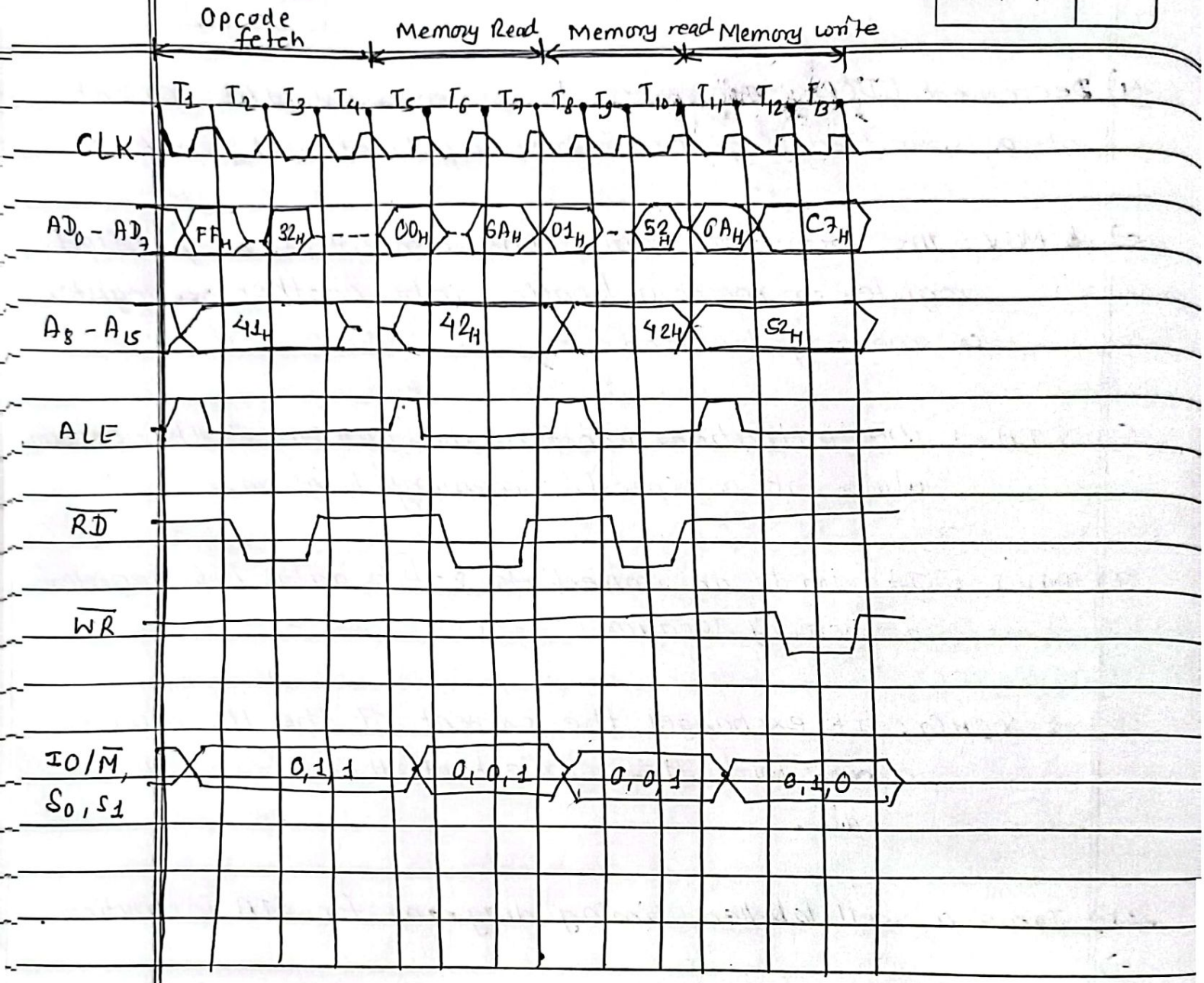
6) STA: This instruction transfers the content of the accumulator into a specific memory location.

7) MVI: It loads an immediate 8-bit data into register or memory location.

8) XCHG: It exchanges the content of the HL register pair with the content of the DE register pair.

26) Draw a well labelled timing diagram for STA instruction.

→



3a) Write an assembly program for 8085 to find the square of the given numbers from memory location 6100H and store the result in memory location 7000H.

```
ORG 0000H
MVI C, 00H
MVI H, 61H
MVI L, 00H
```

DATE / / PAGE

```
LOOP:
    MOV A,M
    ADD A
    STA 7000H
    INX H
    INR C
    CPI 10H
    JNZ LOOP
    HLT
    END
```

3b) Write or Explain programmable interrupt controller (8259) in detail (done).

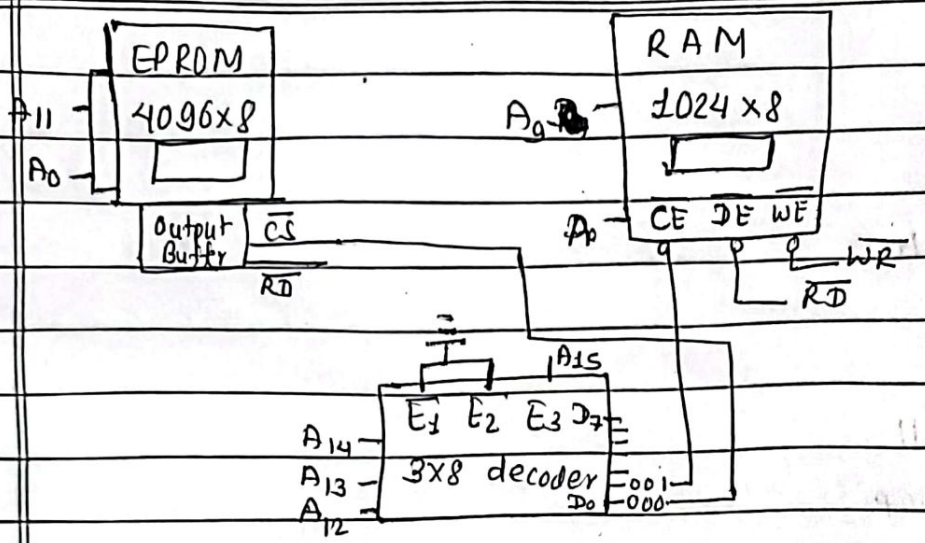
4a) Draw an address decoding circuit to interface 4K x 8 ROM and 1K x 8 RAM with starting address 8000H.

$4K = 4 \times 1024 = 4096$

No. of address $= \log_2(4096) = 12$.

$1K = 1024$

No. of address $= \log_2(1024) = 10$



Address Range

For ROM

Address	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
8000H	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
8FFFH	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

For RAM

Address	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
9000H	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!
97FFH	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1

DATE / / PAGE

4b) OR Write an ALP to change lowercase string into uppercase.

```
DATA_SEG SEGMENT.
    STR DB 'hello world $'
    LEN equ $-STR1
    DATA_SEG ENDS.

CODE_SEG SEGMENT
    ASSUME CS : CODE_SEG, DS: DATA_SEG.

START:
    MOV AX, DATA_SEG.
    MOV DS, AX
    MOV SI, OFFSET STR1.

CONVERT_LOOP:
    MOV AL, [SI].
    CMP AL, 'a'.
    JB NOT_LOWER
    CMP AL, 'z'
    JA NOT_LOWER.
    SUB AL, 20H.
    MOV [SI], AL
NOT_LOWER:
    INC SI
    LOOP CONVERT_LOOP.
    MOV AH, 09H.
    LEA DX, STR1
    INT 21H
```

DATE / / PAGE

```
MOV AH, 4CH
INT 21H

CODE_SEG ENDS
END START.
```

5(a) What are different pre-defined interrupts in 8086 mp? Explain with the use of Interrupt Vector Table.

⇒ Different pre-defined interrupts in 8086 mp are:

- Hardware Interrupts (External Interrupts).

These interrupts are generated by external hardware devices to request attention from the microprocessor.

Example:

① INT 8h: Timer Interrupt (IRQ0) - generated by the programmable interval timer at regular intervals.

② INT 9h: generated by the keyboard controller when a key is pressed or released.

- Software Interrupts (Internal Interrupts)

These interrupts are triggered by software instructions within the program.

Example:

① INT 21h: DOS Interrupt - provides various services through the DOS operating system, such as file operations, input/output, and system functions.

DATE / / PAGE

② INT 16h: Keyboard Services Interrupt - Used for keyboard input services, such as reading keystrokes.

• Other Predefined Interrupts:

① INT 20h: Terminate program - used to terminate a program and return control to the OS.

② INT 21h: set Interrupt Vector - used to set the address of an interrupt service routine (ISR) for a specific interrupt vector.

Use of IVT

1) Provides a centralized mechanism for managing & handling interrupt requests from external hardware devices.
2) It maps each interrupt request to its corresponding Interrupt Service Routine (ISR), allowing the processor to quickly locate and execute the appropriate code when an error occurs.
3) It allows developers to customize interrupt handling by replacing default ISR addresses with custom ones.
4) By providing a lookup table of ISR addresses, it enables efficient & rapid response to interrupt requests without the need for extensive searching or processing overheads.

DATE / / PAGE

5b) Explain 8255-PPI in detail.

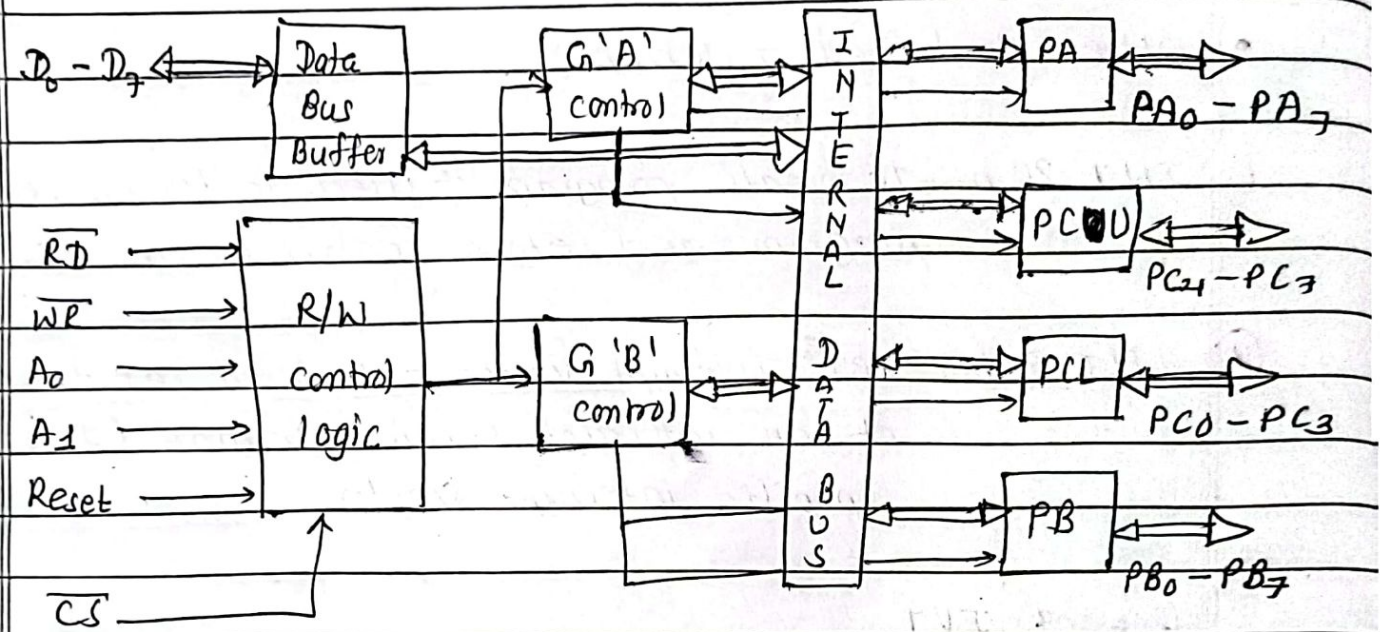


fig: 8255 Architecture.

- Functionality: The 8255 PPI provides parallel I:O capabilities, allowing a microprocessor to interface with peripheral devices such as keyboards, displays, sensors and actuators.
- Architecture: It consists of three 8-bit bidirectional I/O ports (Port A, Port B and Port C), each with its control and status registers. Port C can be further divided into two 4-bit nibbles. Port C (lower) and Port C (upper).
- Modes of Operation: The 8255 PPI supports several modes of operations:

  Mode 0: Simple I/O mode, where each port can be individually configured as input or output.

DATE / / PAGE

Mode 1: Strobed I/O mode, where data transfer occurs only when a strobe signal is applied.

Mode 2: Bidirectional bus mode, where port A acts as input and Port B as output, forming a bidirectional data bus.

Mode 3: Basic Input/Output mode, a combination of Mode 0 and Mode 1.

Mode 4: Combination of Mode 0 and bidirectional bus mode.

Mode 5: Combination of Mode 1 and bidirectional bus mode.

- Control Registers: It has several control registers:
  Group A for configuring port A.
  Group B for configuring Port B.
  Group C for configuring Port C and defining operational modes.

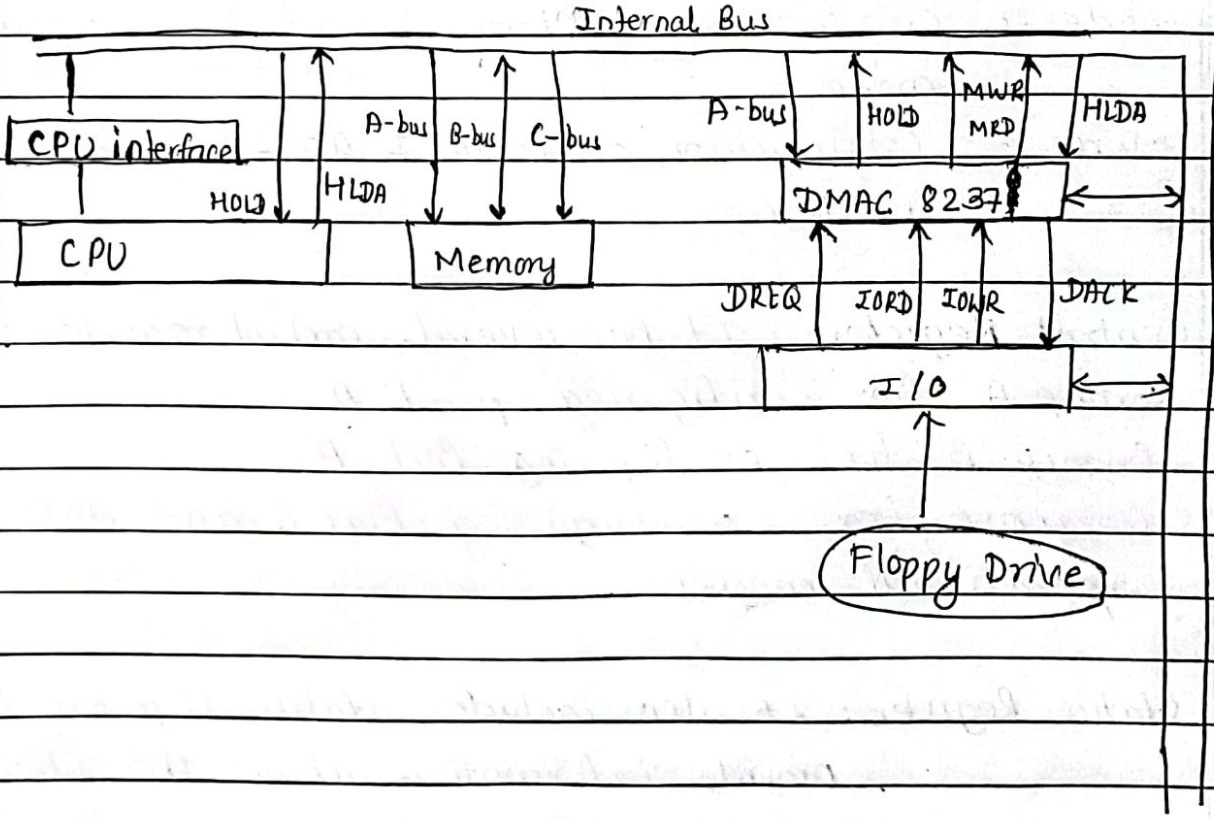
- Status Registers: It also includes status registers that provide information about the status of each I/O port, such as data ready, input/output status, and interrupt status.

- Interrupt Support: The 8255 PPI can generate interrupts to the microprocessor based on certain conditions, such as data transfer completion or external events, allowing for efficient handling of I/O operations.

DATE / / PAGE

6a) What is the importance of DMAC? Explain 8237A.

⇒ Direct Memory Access Controller (DMAC) is important because it helps the computer systems transfer data betⁿ devices and memory without needing the CPU to manage every step. This speeds up data transfer, frees up the CPU for other tasks, and improves overall system performance.



8237 is a ~~the~~ chip commonly used in computer systems to handle data transfer betⁿ peripherals and memory without involving the CPU for every operation. Here's a simple explanation:

1) Data Transfer: to move data between devices (like disc drives, keyboards, etc.) and memory without

DATE / / PAGE

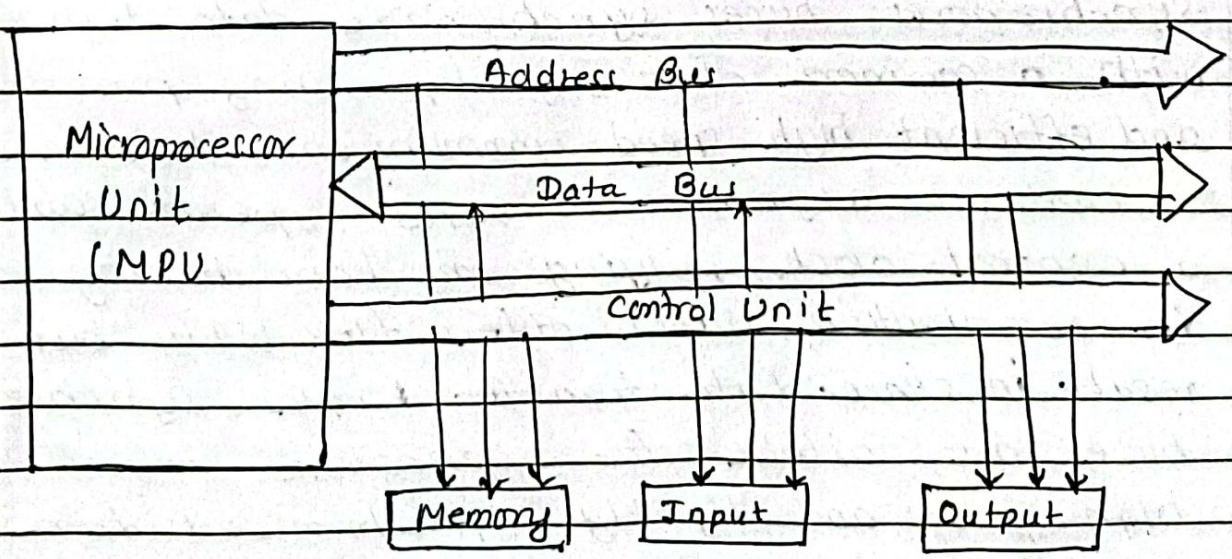
needing the CPU to oversee every transfer.

2) CPU Offloading: by taking over data transfer tasks, it frees up the CPU to focus on other important tasks, improving overall system performance.

3) Efficient Handling: The DMAC operates independently, enabling fast and efficient data transfer, which is essential for tasks like file copying, disk I/O operations, and multimedia processing.

4) Multitasking Support: It allows the CPU to handle multiple tasks simultaneously by managing data transfers in the background, enabling smoother operation of the system.

6b) Explain bus structure of microprocessor. What do you mean by synchronous and asynchronous bus.



DATE / / PAGE

1) Address Bus: It is used by microprocessor to specify the memory location or I/O port for data transfer. It carries the binary address information from the microprocessor to memory or I/O devices.

2) Data Bus: The data bus is used to transfer data between the microprocessor and memory or I/O devices. It carries binary data in parallel form between the microprocessor and external components.

3) Control Bus: The control bus consists of various control signal used to manage and co-ordinate the operation of microprocessor and other system components. These signals include read/write signals, bus request / acknowledge signals, clock signals, and interupt signals.

- Synchronous buses synchronize data transfers with a common clock signal, ensuring precise timing and efficient high speed communication between devices. In contrast, asynchronous buses operate without a central clock, relying on handshaking signals for coordination, which offers flexibility but may result in slower data transfer rates. Synchronous buses are suitable for applications requiring high-speed and tightly synchronized data transfer while asynchronous buses are simpler to implement

DATE / / PAGE

and more flexibility in timing control. Ultimately, the choice between synchronous and asynchronous buses depends on the specific requirements and constraints of the system design.

7) Write short notes on:

1) Memory mapped I.O and IO mapped IO (done)

~~2) RTL Instructions:~~

2) Handshaking.

Handshaking refers to the process by which two or more devices establish and co-ordinate communication with each other. It involves a series of predefined signals or protocols exchanged between the devices to ensure that data transfer occurs smoothly and without error.

Initiation: Handshaking typically begins with one device sending a request or initiating communication with another device.

Acknowledgement: Upon receiving the request the receiving device sends back an acknowledgement signal to confirm that it is ready to proceed with the communication.

Data Transfer: Once the acknowledgement is received, the devices exchange data according to the agreed upon protocol.

DATE / / PAGE

Completion: After the data transfer is complete, a final signal ~~may be~~ may be sent to indicate the end of communication or to signal readiness for the next operation.