

Name: Masum Hossain

ID : 221902164

Sec : 221-D9

Course: Microprocessor & MicroController

Self (Microprocessor)

Subject L-1

Sat Sun Mon Tue Wed Thu Fri
0 0 0 0 0 0 0

Date : / - /

processor → which processor (number/text)
Manipulate data/perform operation

8085
Microprocessor → takes number & perform
↳ programmable Arithmetic & logical
device operations and

programmable device

which is programmable device
according to the instruction

The data that the
comes from:

i) Input device (Keyboard...)

program → sequence of
instruction

e.g. $x = y + z$ may translate,

MOV R1, 1004

MOV R2, 1000

ADD R1, R2

MOV R1, 1004

middle → assembly
language

language

language

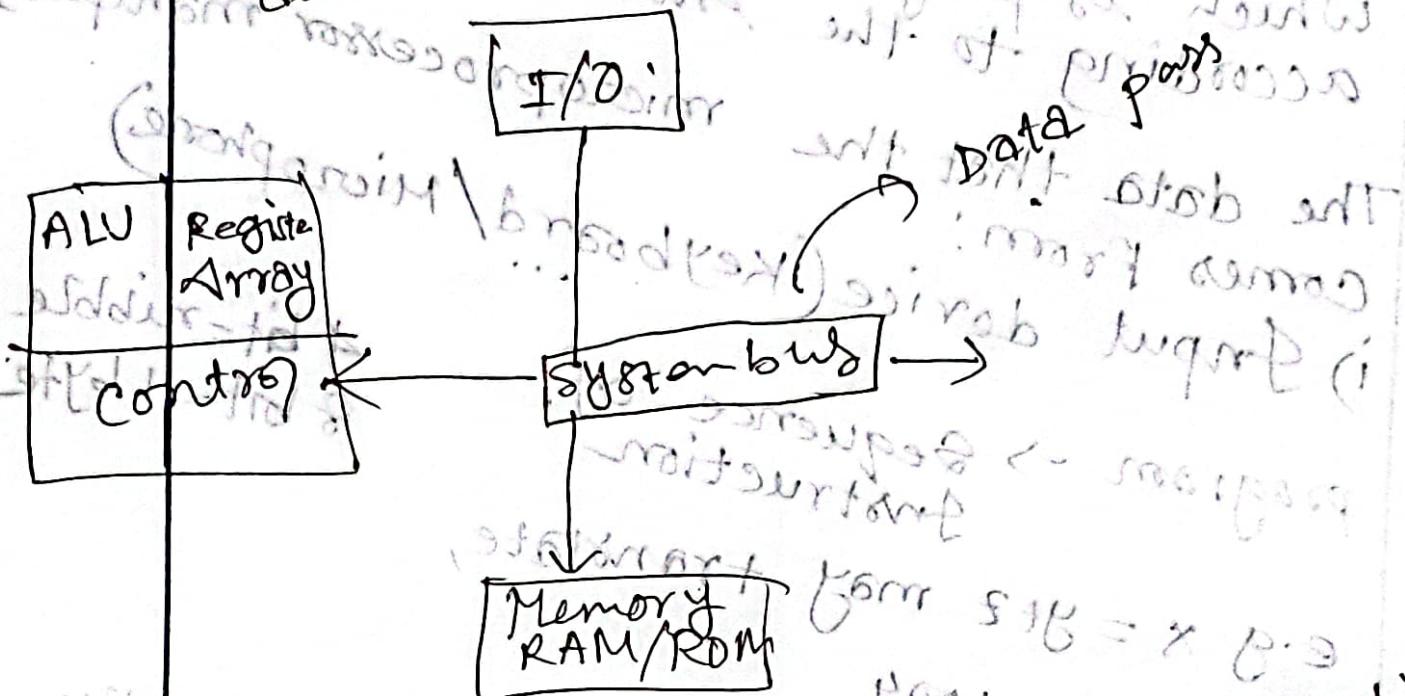
language

language

Memory → location where information is kept while not in current use

Inside Microprocessor Unit (ALU)

- 1) Arithmetic/Logic Unit (ALU)
- 2) Control Unit
- 3) An array of registers for holding manipulated data while it is being manipulated



System bus → Build up connection with CPU between peripheral devices

Subject _____

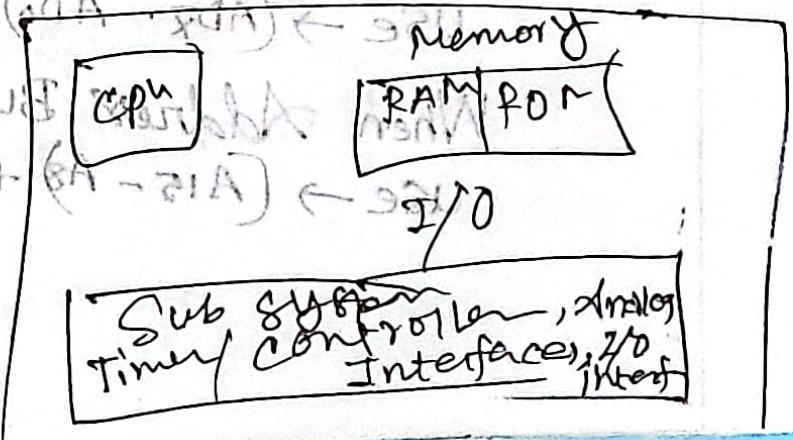
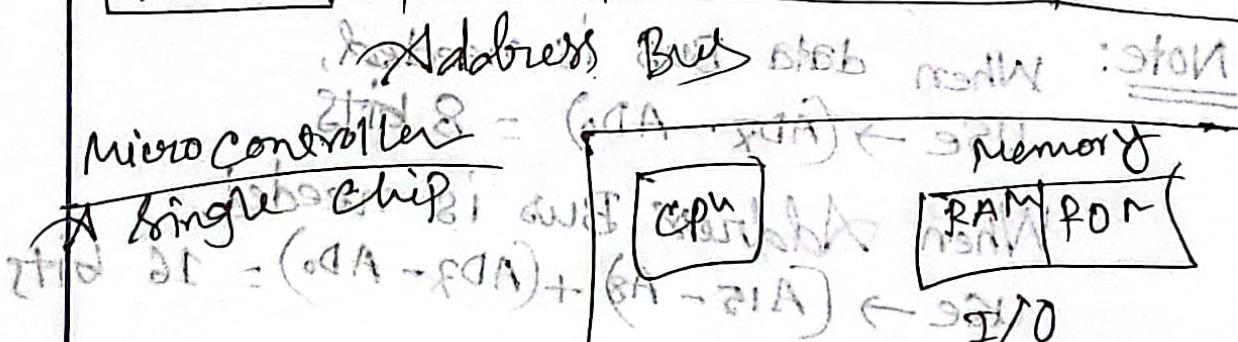
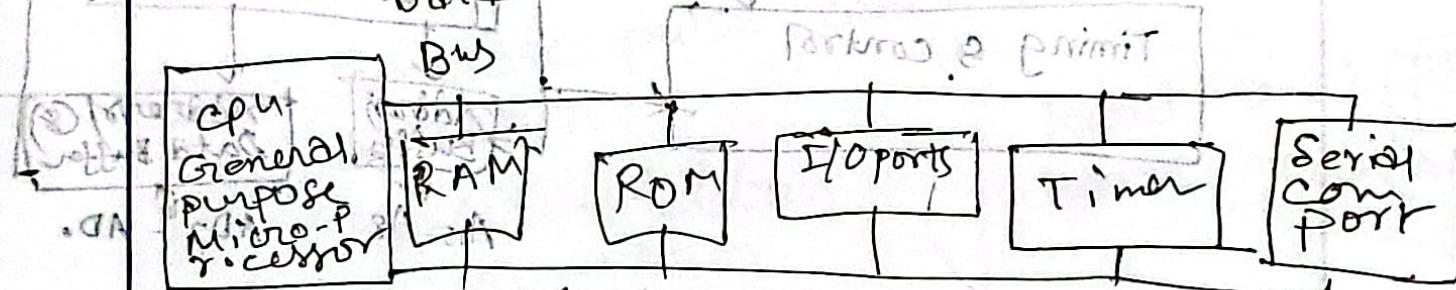
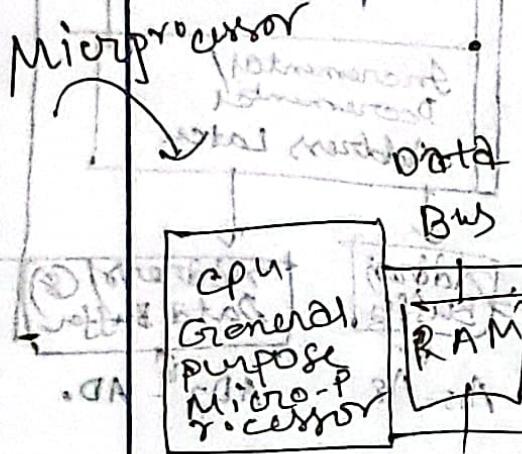
Sat Sun Mon Tue Wed Thu Fri
○ ○ ○ ○ ○ ○ ○

Date: / /

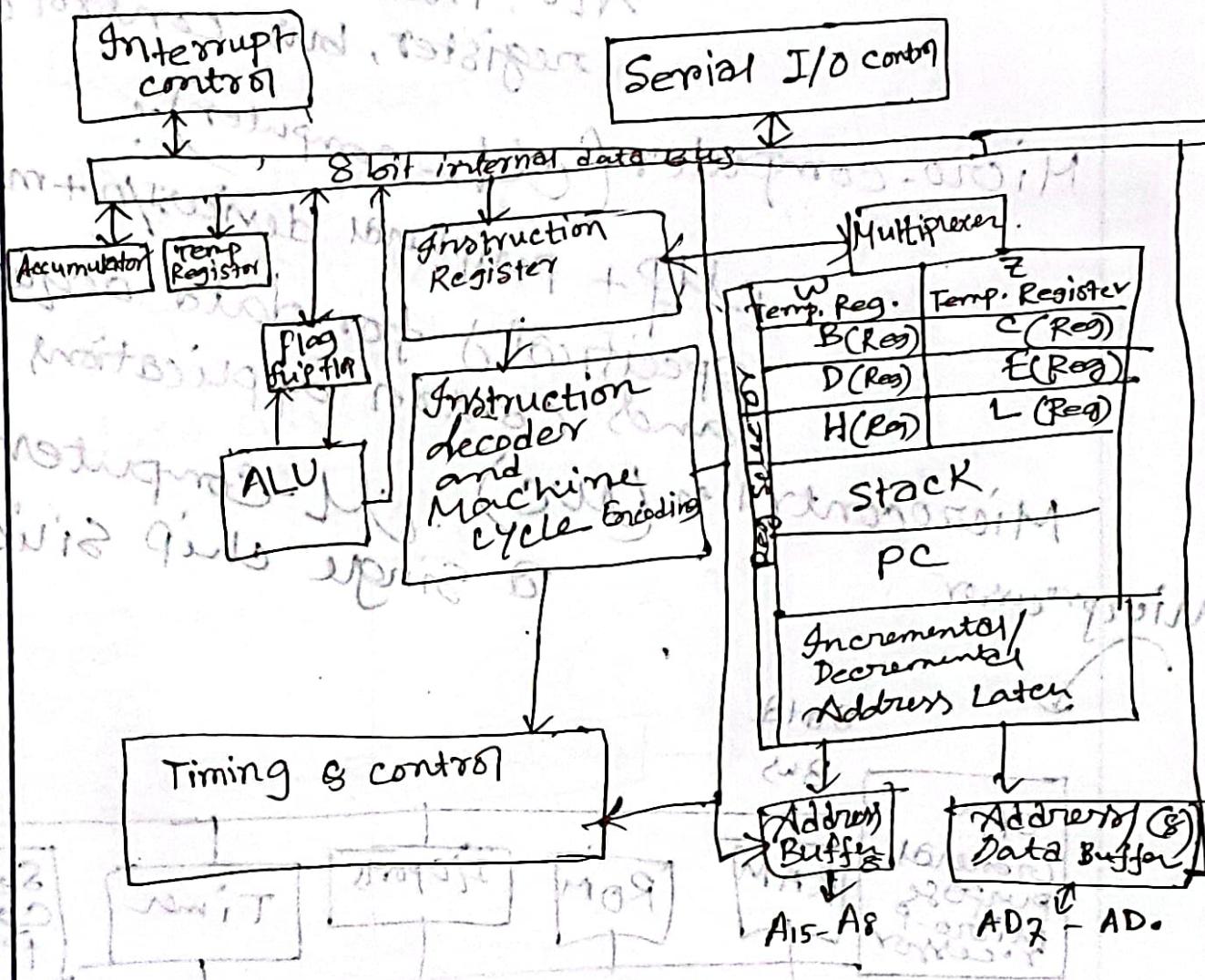
Microprocessor \rightarrow A ~~chip~~ is a single chip ALU, instruction decoder, register, bus control circuit

Micro-computer (CPU - computer)
CPU + peripheral devices I/O + memory
specifically for data acquisition
and control applications

Microcontroller (MC) : Micro-computer on
a single chip silicon



Internal details of 8085



Note: When data Bus is needed,
use $\rightarrow (AD_7 - AD_0) = 8 \text{ bits}$

When Address Bus is needed,
use $\rightarrow (A_{15} - A_8) + (AD_7 - AD_0) = 16 \text{ bits}$

Notes and for better understanding

For pin diagram of 8085
classified into seven groups

Address Bus \rightarrow A₁₅ - A₈, (Right pin 21 to 28)

V_{cc} = 10 V_{ss} = 20

Data Bus \rightarrow AD₇ - AD₀

Left pin (12 to 19)

control signals \rightarrow perform 3 operations

RD : performs Read operations on memory

WR : perform Write operation on a specific location according to the algorithm.

ALE : Generate pulse when perform new operation

high(1) \rightarrow Address
low(0) \rightarrow Data

RD : Read \rightarrow 31 (RD)

WR : Write \rightarrow 30 (WR)

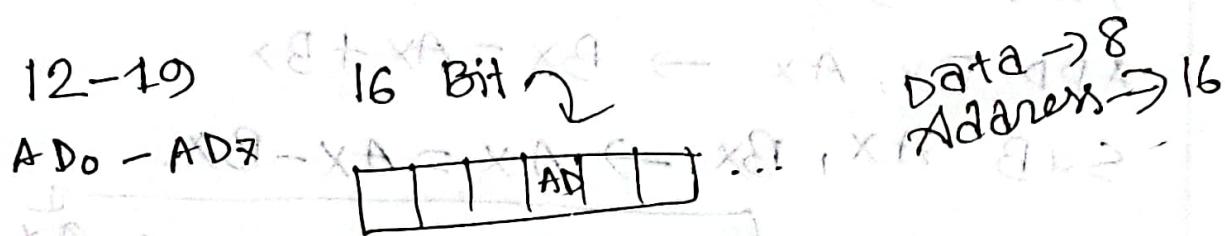
I/O \rightarrow Input, output
M \rightarrow Memory operation
IO/M \rightarrow 0 (Memory operation)

\hookrightarrow 1 (I/O) operation

Microprocessor vs Microcontroller vs Microcomputer

- 3 units
 - ALU (Arithmetic Logic unit)
 - Control unit
 - An Array of Registers and a small internal Memory

Memory of Microprocessor:

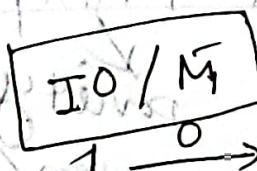


Interrupt:

6 → Trap → Non-maskable Interrupt

When (30) pin
ALE(1) → Address
ALE(0) → Data Bus

10 → INT



I/O G

Read/write sign

বাইনের দিকে

+ RD address ← bus write memory

8+8

Sat Sun Mon Tue Wed Thu Fri
 0 0 0 0 0 0 0

Subject _____

Date: / /

$Ax \rightarrow$ Accumulator Reg

$Bx \rightarrow$ Base Reg

$Cx \rightarrow$ Counter Reg

$Dx \rightarrow$ Data Reg

ADD SUB MUL DIV

int a = 5;

Mov Ax 5

Mov Bx, 6

ADD BX, AX \rightarrow BX = AX + BX

SUB AX, BX \rightarrow AX = AX - BX

MUL \rightarrow AX \rightarrow AX = AX * AX

\swarrow \searrow

Always Always
AX same

Flag Register (8) bits

S	Z	X	A_c	X	P	X	C	Y
---	---	---	---------------	---	---	---	---	---

Parity

even/odd

result = 0 20

flag = 1

program status word \rightarrow Accumulator +
Flag register
8 + 8

Sat Sun Mon Tue Wed Thu Fri
0 0 0 0 0 0 0

Subject _____

Date: / /

So G S₁ → Identify the type of the operation

CLK (out): 37A → 21A ← and enable

Interrupts:

INTA → { Interrupt Acknowledgement signal
→ (Occurred interrupt) + I

RESET IN: Program Counter Zero.

RESET OUT: Reset all the connected devices : RW

READY: CPU have to wait if READY is zero. When (1) Start op.

HOLD: In case of DMA, Send hold request to perform any operation. Make hold the CPU and Data transfer as well as Received.

(notif. Gramm) 0 ← M0I

mitrag (0I) 1 ←

Subject _____

Sat Sun Mon Tue Wed Thu Fri
0 0 0 0 0 0 0

Date: / /

Addressing Mode:

Immediate addressing mode:

'8/16 bit data is specified' TMS
The \rightarrow value

viz: MVI K, 20F

Register \rightarrow value is copied

It indicates that, 20F

is copied into register K.

MV = Move
I = Immediate

Register Addressing Mode: copy one Reg to another

Register \rightarrow value is copied to K Reg.

MOV K, B Here, B is copied

Direct addressing mode:

LDB 500K

data at Address 500K

is copied to Register

LD = Load

B \rightarrow Register Name

500K

9. M

Indirect Addressing mode: pointed by Reg

Indirect Addressing mode: from memory Address

MOV K, B, transferring from memory Address pointed by the Register to the Register K

8.98 IXJ

Subject _____

Sat Sun Mon Tue Wed Thu Fri
● ○ ○ ○ ○ ○ ○

Date: 11/09/2024

Implied addressing mode:

Doesn't require any operand.

CMP

Instruction Set

Group of instructions that microprocessor supports

246 instruction for 8085

8 bit Binary value (Op-code/Instruction Byte)

Data transfer instruction: Memory to Reg
Reg to Memory

Arithmetic instruction: Add, Sub, Mul

Logical instruction: AND, OR, NOT.

Branching instructions: Jump, Call, Return.

OR case

I/O and machine control instructions:

I/O related contents I/O

MOV R1, R2

MOV M, R

MVI R, FFH

MVI R, 8-bit data

LXI SP, 8-bit data

R1 ← R2 to Reg

M ← R Copy of

R ← FFH

R ← 8-bit data

SP ← 8-bit data
Stack pointer

Subject _____

Sat Sun Mon Tue Wed Thu Fri
○ ○ ○ ○ ○ ○ ●

Arithmetic Instruction

Date: 20/09/24

Instruction

ADD R

ADD M

ADI 8-bit data

ADC R/M (Anything variable)

SBB R

INR R

Decrement by one DCR R

INX RP

DCX RP

INR R, B

ANA M

ORA R

XRA R

CMP M

RLC

CMA

complement of accumulator

Operation

$A \leftarrow A + R$

$A \leftarrow A + M$

$A \leftarrow A + 8\text{-bit data}$

$A \leftarrow A + M + CY$

$A \leftarrow A - R - CY$

$R \leftarrow R + 1$

$R \leftarrow R - 1$

$RP \leftarrow RP + 1$

$RP \leftarrow RP - 1$

$R \leftarrow R + 3$

$A \leftarrow A \wedge M$

$A \leftarrow AVR$

$A \leftarrow A \oplus R$

$A \leftarrow A - M$

RP = register pair

rotate left without carry

Subject _____

Sat Sun Mon Tue Wed Thu Fri
○ ○ ○ ○ ○ ○ ●

Date : 20/09/24

conditional
&
non
conditional

Three types of branching instructions

Jump $A \rightarrow A$

Call $M+A \rightarrow A$

Return $A \rightarrow M$

Jump 16 bit address $\rightarrow A$
call " (push) $\rightarrow M$

Stack RET $\rightarrow A$ (POP)

Stack, I/O and machine control
instruction:

Stack, $A \rightarrow A$

push → Push two bytes of data into stack

pop → pop two bytes of data from stack

HLHL → Exchange top of stack with HL

HLT

Condition depends on flag Register.

MAN
 $M+A \rightarrow A$

JZ
CZ
RZ

JZ
CZ
RZ

microprocessor for

HLT → pause the
while.

ring rotator

Memory Interfacing 8085:

Interfacing : Connecting outside peripherals
 Connecting Microprocessor with memory.

Why need?
 To match the memory requirements and microprocessor signals.

We have address Lines ($A_0 - A_{15}$) = 16
 $2^{16} = 65,536 \approx 64 \text{ KB Memory}$

$$0000H \rightarrow FFFFH$$

↓
 Starting blending

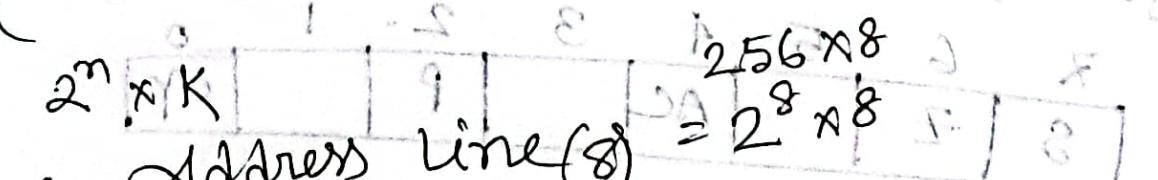
There are three control signals for interfacing

I/O/M'

RD'

WR'

→ Input/Output operation to



$n = \text{address line (8)}$

$K = \text{data line (8)}$

Ans

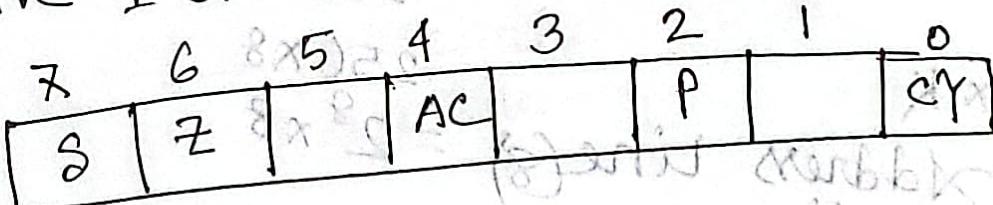
Accumulator: 8 bit register used to perform arithmetic, logical, I/O and LOAD/STORE operations. It is connected with internal data bus & ALU.

Program Counter: 16 bit register used to store the memory Address Location [the next instruction execute].

Stack Pointer: 16 bit register perform push & pop operations.

Temporary Register: Hold temporary data of arithmetic & logical operations.

Flag Register: 8 bit register having five 1-bit flip flop.



Subject _____

Sat Sun Mon Tue Wed Thu Fri
○ ○ ○ ○ ○ ○ ○

Date: / /

Interrupts in 8085:

Vector interrupt: interrupt address is known to the processor. viz: RST 7.5, RST 6.5, RST 5.5 Trap.

Non-vector interrupt: NOT known by the processor

Maskable interrupt: can be disabled or masked.

Non-Maskable interrupt: The request must be executed. TRAP.

Software interrupt: has to add the instructions into the program.

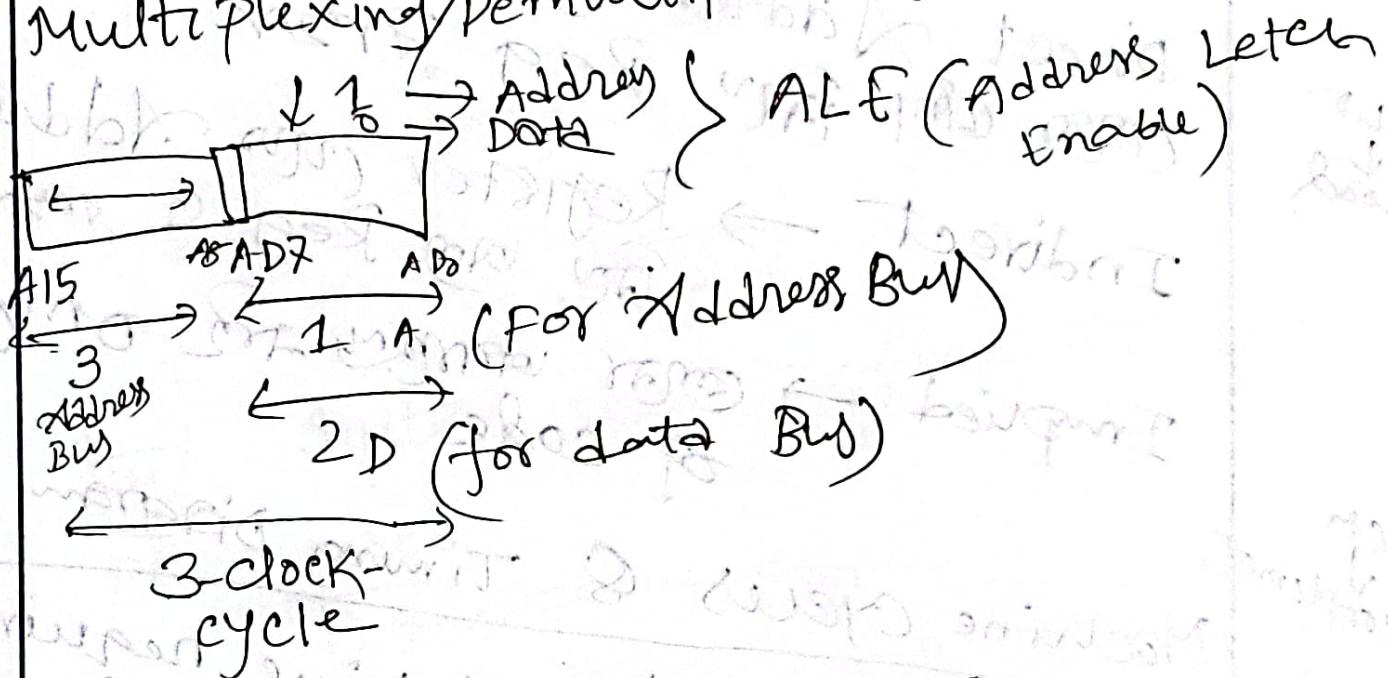
viz: RST 0 to RST 7.

Hardware interrupt: External device interrupt occurs. TRAP, RST 7.5, RST 6.5, RST 5.5, INTA.

Instruction fetch

Address Bus \rightarrow Accum - Data bus
 \rightarrow Data Decoder (with fig and proper explanation)

Multiplexing Demultiplexing ADD₀-ADD₇



Instruction, WORD size 8085

one byte instruction \rightarrow 1 byte

Two bytes \rightarrow 1 \rightarrow 1 \rightarrow Data

Three bytes \rightarrow 1 \rightarrow 2 \rightarrow Data

Subject _____

Sat Sun Mon Tue Wed Thu Fri
0 0 0 0 0 0 0

Date : 1-8-2019

Addressing Mode:

Immediate Addressing mode: parameter \rightarrow Register

Register Addressing Mode: Address

Direct Addressing mode: Address

Indirect \rightarrow Register via Address

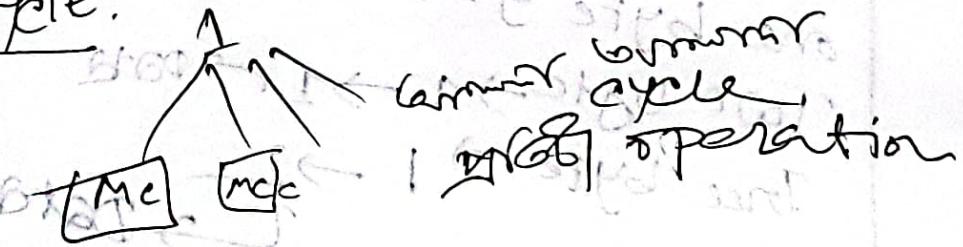
Implied \rightarrow same memory only

Machine Cycles & Timing Diagram!

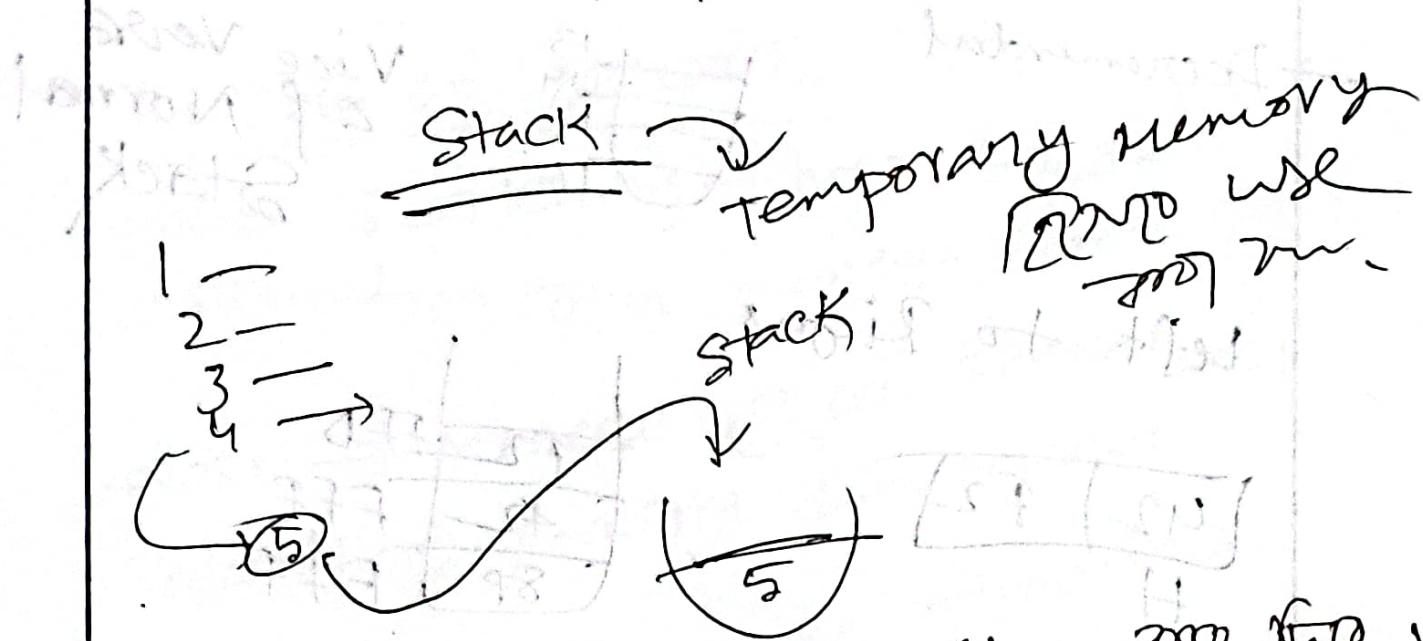
Instruction cycle: total time required to execute an instruction

Machine cycle:

T-State:



Timing diagram: control signal & Graphical representation.



initial user define Z.M.F.C.

used

- 1) Interrupt Z.M.
- 2) Subroutine (one kind of function)

↓
Z.M., Z.M. same Z.M.

call int

call function and
Return function

exit Z.C. stack → Z.M. ↓

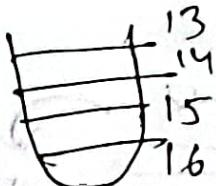
Subject _____

Sat Sun Mon Tue Wed Thu Fri
0 0 0 0 0 0 0

Date: / /

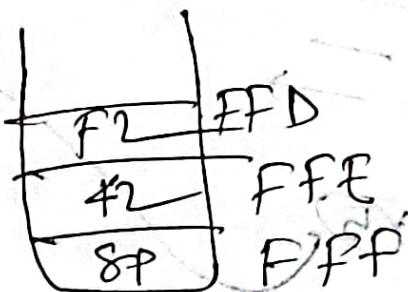
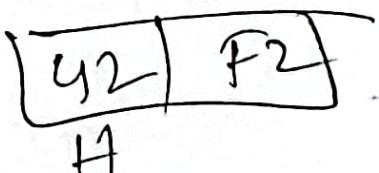
Stack principle

→ Decremental



Vice Versa
of Normal
Stack

Left to Right



push → Decrement
push

pop.

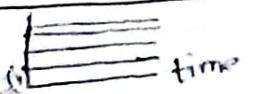
→ pop

→ increment

→ MS, DS, SS

→ DS, SS, CS, ES

Circuit switching.

FDM 
TDM 

96RT

Interrupts: STOP the current instruction stored in Microprocessor

Vectored \rightarrow Address stored in Microprocessor.

Non-Vectored \rightarrow Not in Microprocessor.

Masked \rightarrow can be ignored

Non-maskable interrupt \rightarrow Must be handled first

Hardware interrupt \rightarrow occurred by hardware

Software \rightarrow opcode interrupt which is written on IDE (Code)

Vector vs Nonvector interrupt

(4)

(1)

starts from

1.5 and 32M 0
Hexa (decimal)
value + 8

Non-maskable

Trap

no 2600000000000000
stop 2600000000000000
trap 2600000000000000

priority of interrupts:

Trap

RST 7.5

RST 6.5

Total : 24 priority

Emergency → \downarrow \leftarrow Emergency

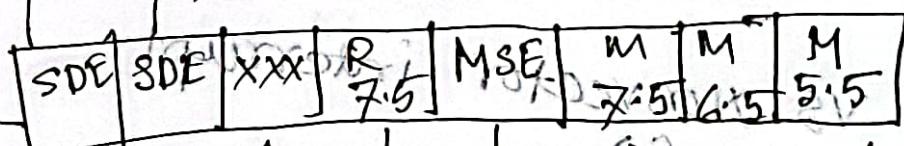
When interrupt enable comes request

for interrupt

When disable \rightarrow No interrupt comesWhen enable \rightarrow Vector table is assignedInterrupt \rightarrow Set Interrupt MaskDisable \rightarrow No Data pass

Signal Data Enable

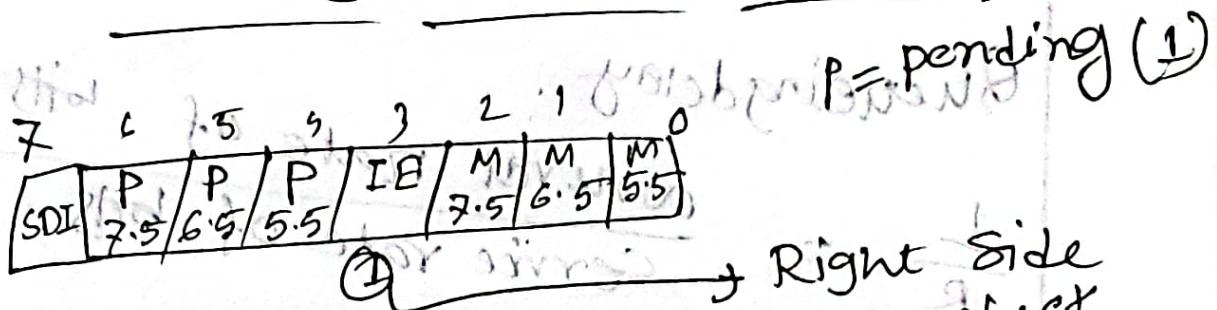
M = MASK

0 \rightarrow Active
1 \rightarrow Mask

Serial Data Output

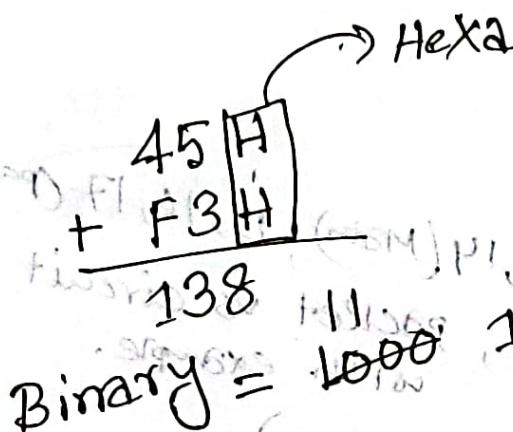
Side Data Pass depends on this pin

0 (Mask Set Disable)
1 (perform the right 3 pins operation)

RIM (Read Interrupt Mask)

RIM vs SIM

Memory Mapped vs IO Mapped



DATA < 9M

Sign bit \oplus zero = 0 if $0 \oplus 0 = 0$ zero = 1 if $1 \oplus 1 = 0$

AC = 0

P = 0

CY = 1

if $0 \oplus 0 = 0$ if $1 \oplus 1 = 0$

event = 1 ps

(6 + process) CF

Flag Register = 8 bits

$$\begin{array}{r}
 45 \text{ H} \\
 + F3 \text{ H} \\
 \hline
 45 \\
 F3 \\
 \hline
 38 \\
 = 00111000 \\
 S=0 \quad \text{Zero} \quad \text{Bit } = 0 \\
 \text{sign bit}
 \end{array}$$

Carry bit =

Auxiliary carry = 000

1 bit যদি পূর্ণ হয়ে থাকে = 1
যদি নয় হয়ে থাকে = 0

Parity (3) = 0

even(1) = 1

odd(1) = 0

$\rightarrow CY = 1$

$$\begin{array}{r}
 85 \\
 + 1E \\
 \hline
 A3 \\
 = 0011 \\
 10101
 \end{array}$$

$S = 1$ + 1 এবং 1 এবং 1
zero = 0

Auxiliary carry = 1

Parity = 1

$CY = 0$

Microprocessor & Mi

Subject

Sat Sun Mon Tue Wed Thu Fri
○ ○ ○ ○ ○ ○ ○

Date: 14/09/24

PUSH → Decrement then store.

POP → pop & then Increment

PSW → program status word

Accumulator → Higher Bit

Stack → (2 Decrement) works that way decrementing by two.

Call → 16 bits address included always (Subroutine address)

Follow the instructions:

RC 0 = (0) 111111

RNC 1 = (1) 000000

RZ 0 = (1) bbb

RNZ

Advantages of Subroutine: Repeated use of the same code

→ 1 byte ↓ 2 bytes

ICALL [8032] → 3 bytes

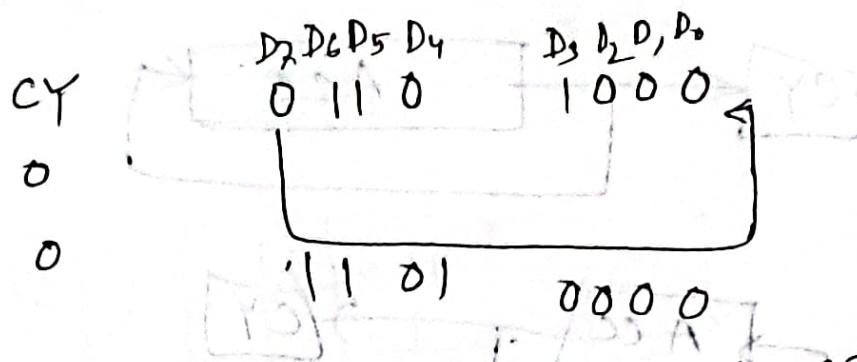
RET SIM → opcode (3D) send
RIM → opcode (2D) Received

1 byte *** SIM VS RIM

Number of Instruction:

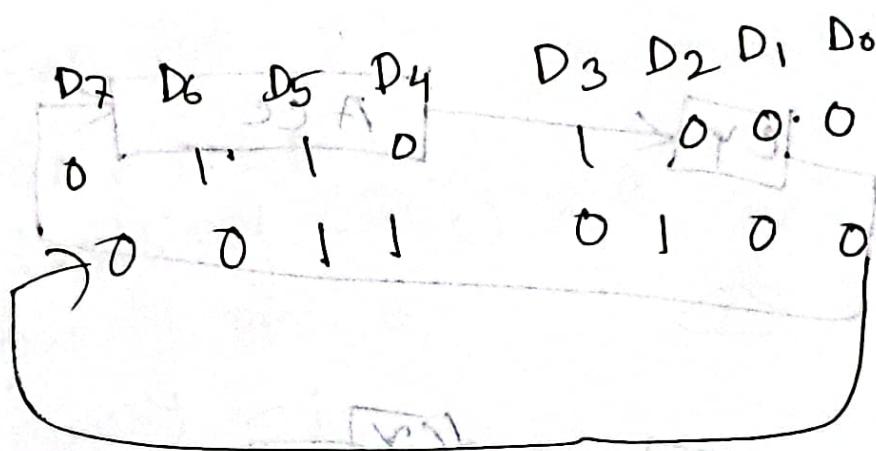
Rotate Instructions

RLC

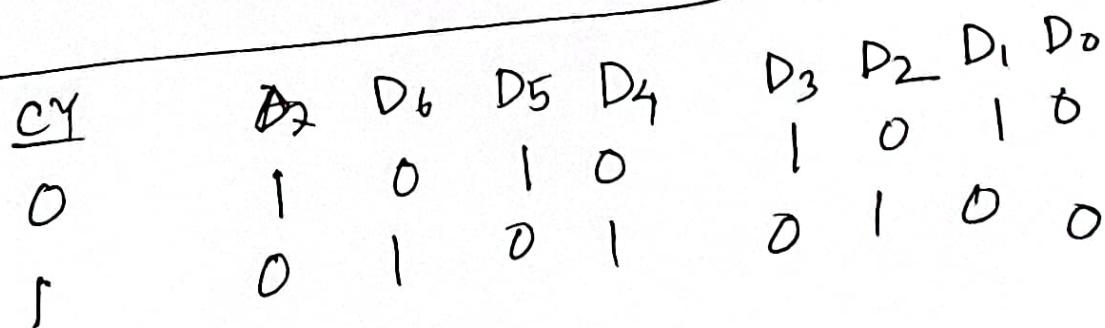
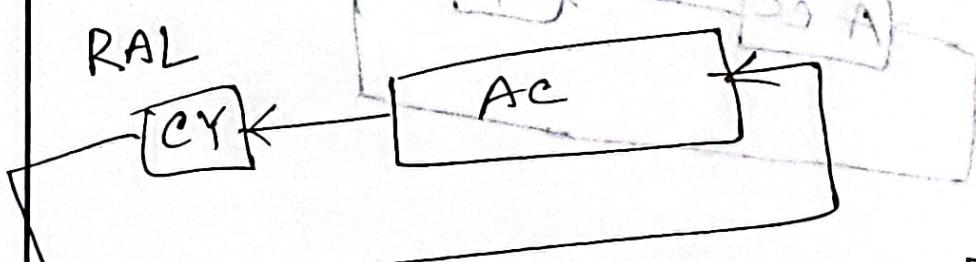


RRC (Rotate Right without carry)

Given



RAL



Subject _____

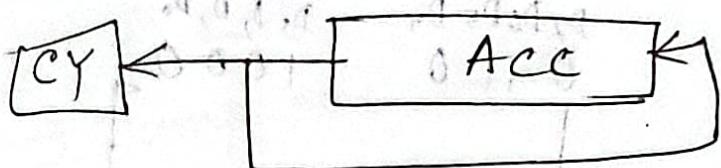
PC-00-00

Sat Sun Mon Tue Wed Thu Fri
0 0 0 0 0 0 0

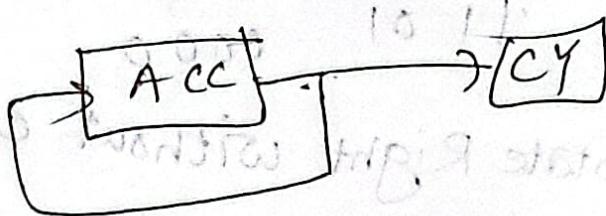
Date: / /

RR → carry bit will be pointed
on D2

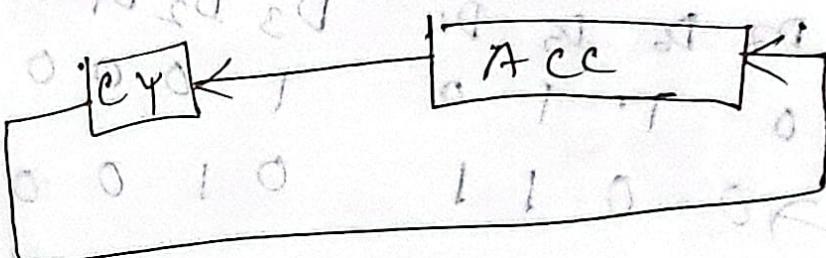
RLC



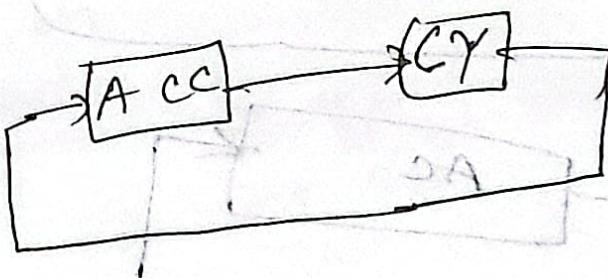
RRC



RAL



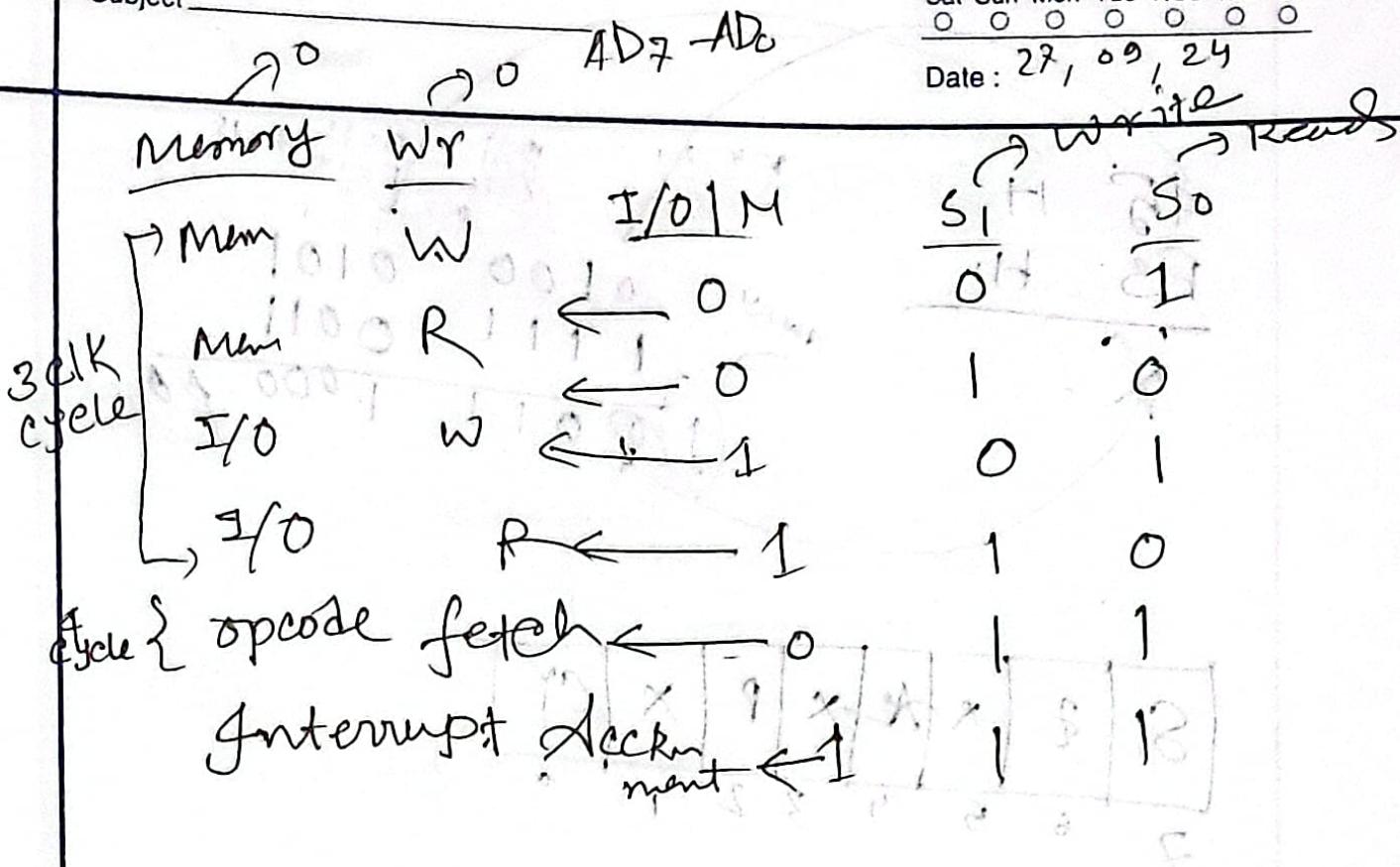
RAR



Subject

Sat Sun Mon Tue Wed Thu Fri

Date : 27/03/20



ALE → Address
Data → D_{ata}

RD

WR

$\frac{WR}{RD}$

1 → Address 0 → file after

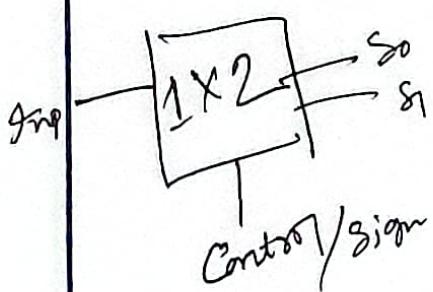
Data 0 = Bymo 0-18 036

For Demultiplexing

$\begin{matrix} 1 \\ 0 \end{matrix}$ → Latches → Address
 $\begin{matrix} 1 \\ 1 \end{matrix}$ / → Data Bus

Temporary storage (

Address ৰ
Store বংশা
(খান)



Opcode fetch

- 4 T
- 3 clock-cycle
- 1. A Bus
- 2 D Bus
- ALE → 1
- RD
- I/O/M, S₁ S₀

ALE → Address
latch enable

NOA

NOB

Memory (only S₁ → 1) → Memorize

→ 3T

→ 3 CLK cycle

I/O/M, S₁, S₀ M RD 1 writing memory real
0 1 0

Memory write:

As usual Memory
only change WR

I/O/M, S₁ S₀
0 1

Subject _____

Sat Sun Mon Tue Wed Thu Fri
○ ○ ○ ○ ○ ○ ○

Date: / /

Calculate the number of IC required for Memory Interfacing.

$$\text{Num of IC's} = \frac{\text{Memory size (32 KB)}}{\text{capacity } 256 \times 4}$$

$$= 256 = \frac{32}{2^8 \times 2^2} = \frac{2^5 \times 2^{10}}{2^8 \times 2^2}$$

$$32 \text{ KB} \rightarrow 32 \times 2^10 \times 8 \rightarrow \text{Data line} = 2^8$$

$256 \times 4 \leftarrow 4$ 8 bit data line.
 So, two IC's will be placed in a row to form 8 bit data line

$$\text{So, total of } = \frac{256}{2} = 128 \text{ rows}$$

$$\text{IC need} = 2 \text{ Ans}$$

EP ROM

RAM

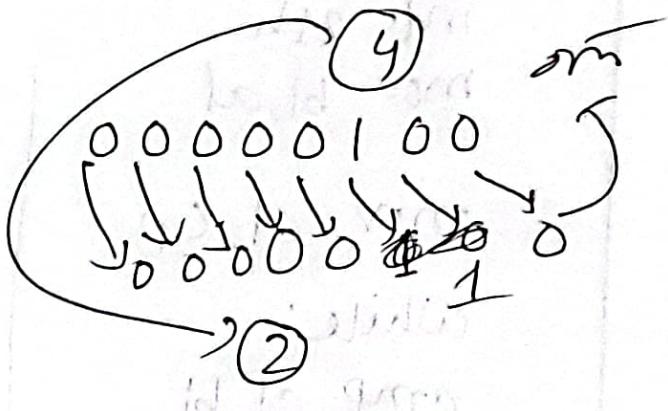
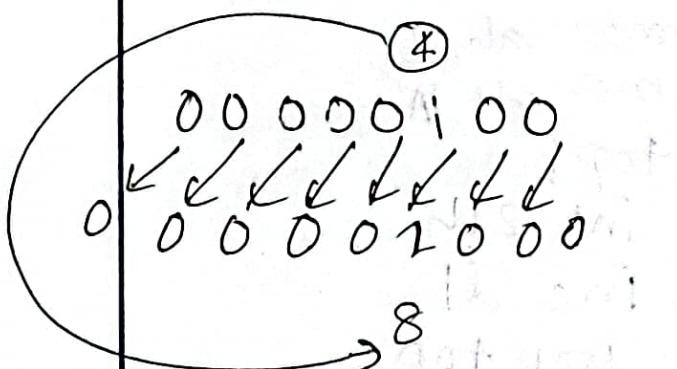
Subject _____

Sat Sun Mon Tue Wed Thu Fri
0 0 0 0 0 0 0

Date: / /

Left shift → multiply by two

Right shift → divide by two



mov ax, 4 ; value 4 taken

shl ax, 2

Shift
left

→ কানেক্স শিফট করব

বয়েচ্যুল কিম্বা করব

mov ah, 2

mov dl, ax → Add int dx, 48

int 21h

back or

Blank

Screen

Show করো

নি দিব

Internal Architecture of 8086

→ Bus Interface Unit.

- BIU → fetch

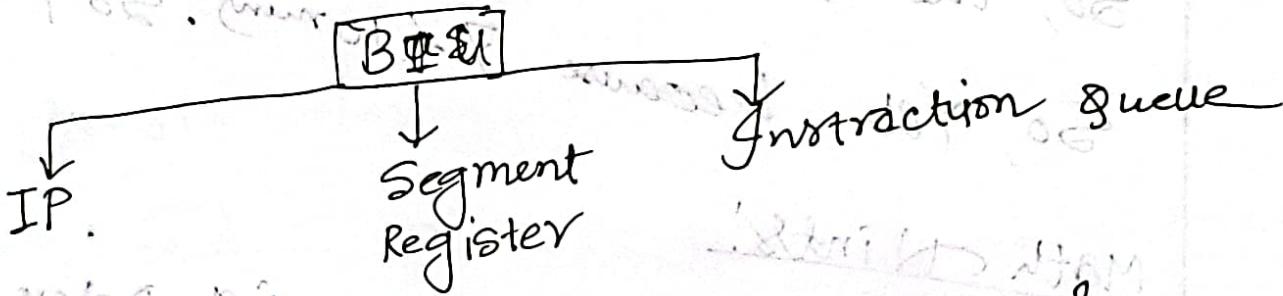
→ EU → DECODE + Execute

8086 → 20 bits Address Bus

16-bit D/A

4 bit → A

Data Bus → 16 bit



Data Segment (DS) → Data stored

Code Segment (CS) → code & data R/W

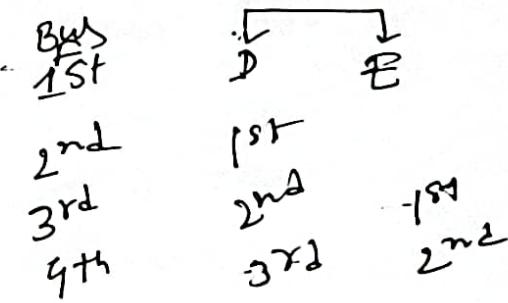
Stack Segment (SS) → store the correct

Extra Segment (ES) → like Data segment

Pipelining

1st CLK cycle

1st
2nd
3rd
4th



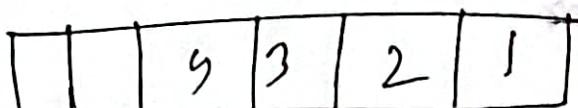
Subject _____

Sat Sun Mon Tue Wed Thu Fri
○ ○ ○ ○ ○ ○ ○

Date: / /

Queue → containing 6 instruction at a time

Queue have to empty two portions
to execute the instruction.



Physical Address → Address of Next Instruction

16 bits segment address + 16 bits offset
address → Physical Address

ALU → Generate Physical Address

- ↳ Segment Address
- ↳ Offset Address

8085
Z, S, AC, P, C

Diagram of 8086 Mp

8086

- ↳ Status Flag
overflow → S, Z, AC, P, C
- ↳ Control Flag
 - ↳ TF, DF, IF

Subject

Microprocessor & Micro-Controller

Sat Sun Mon Tue Wed Thu Fri
0 0 0 0 0 0 0

Date: 19/10/2024

Given,

The memory size:

2KBytes

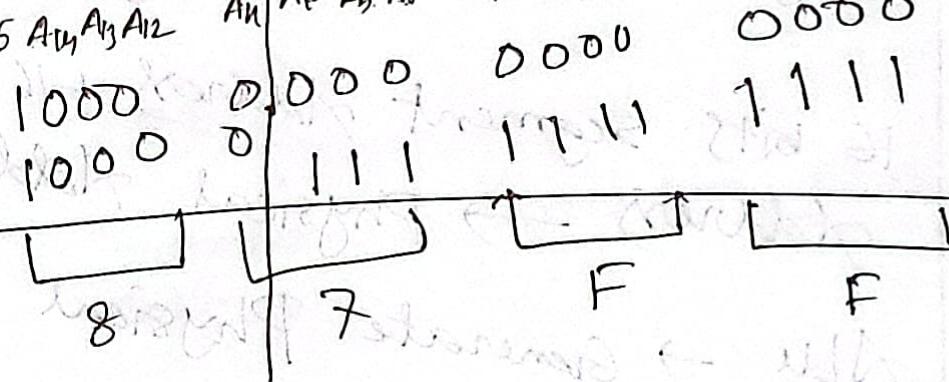
$$= 2^1 \times 2^{10} \times 8 \text{ bytes}$$

$$= 2^{11} \times 8$$

So, the address line is = 11

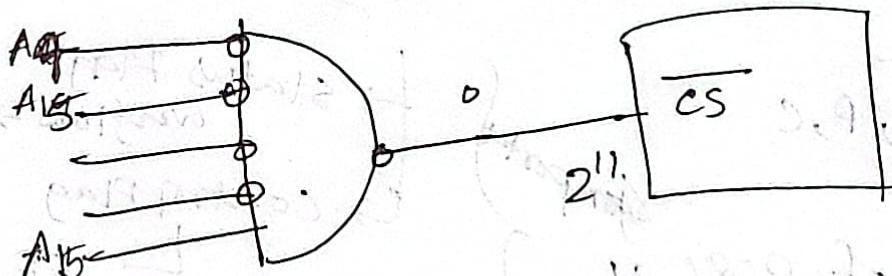
The starting Address is = 8000H

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



Starting = 8000H

Ending = 87FH



The address and data Bus:

ID/M	S ₁	S ₀	Operation
0	0	1	Memory Write
0	1	0	Memory Read
1	0	1	I/O Write
1	1	0	I/O Read
0	1	1	opcode Fetch
1	1	1	Interrupt acknowledgement

opcode → It is an instruction so it will be fetched from Memory.

Interrupt acknowledgement → Sent by I/O.

Machine cycle: Multiple operation like taken from user or directly assigned and operand. Add B, 13H; (3 to 6) T-state.

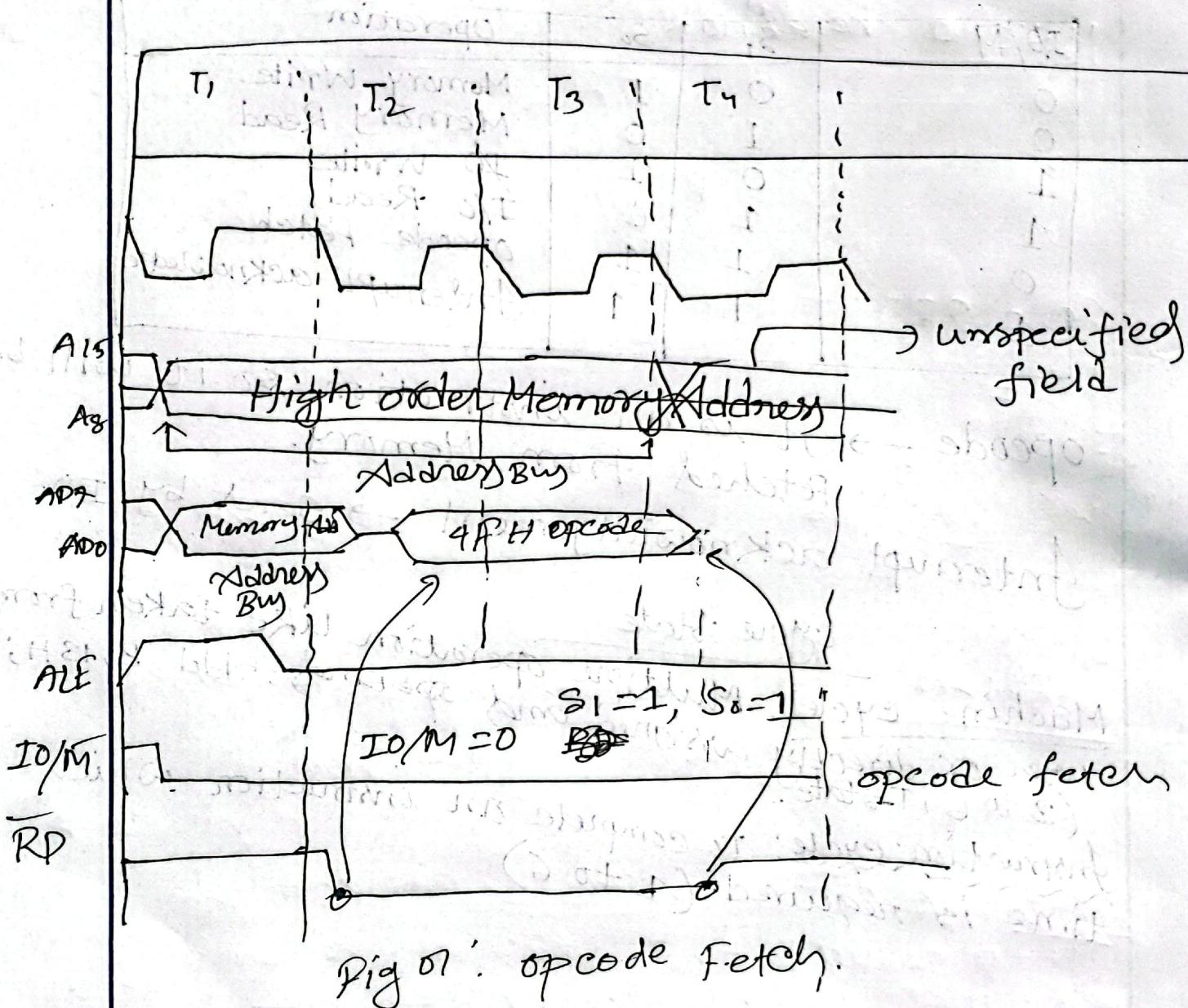
Instruction cycle: To complete an instruction which time is required. (1 to 6)

Subject _____

Sat	Sun	Mon	Tue	Wed	Thu	Fri
○	○	○	○	○	○	○

IO/M S₁ S₀

Date: / /

opcode Fetch:

Note: others have only Three T state.

Writing Microprocessor

Memory Interfacing (8085)

Interface 2Kbytes of memory to 8085 with starting address 8000H.

2K bytes

$$= 2^1 \times 2^{10} \times 8 = 2^{11} \times 8$$

11 → Address line and 8 = Data line.

A₁₅ - A₀

Address																A ₀
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8000H
1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	87FFH

Starting = 8000H

Ending address = 87FFH

From, A₁₅ - A₁₁ will act as chip Selector.

