

Difference between MPU vs MC:

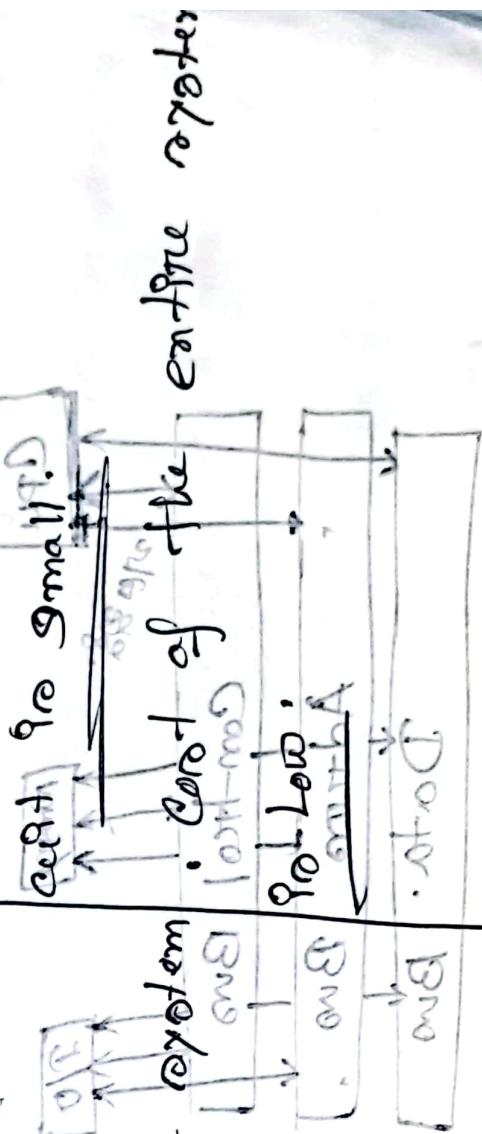
Microprocessor's part

- It is a heart of computer sys.
- Has memory and I/O component
- Has to be connected externally with memory and I/O.
- Memory and I/O has to be connected externally, So the circuit becomes large.

Microcontroller's part

- It is the heart of the embedded system.
- It has processor along with internal memory and I/O components.
- T/O Componants. (CPU, RAM)

- Memory and I/O already present and the internal org.



- Control of the entire system is high.
- Control of the entire system is low.

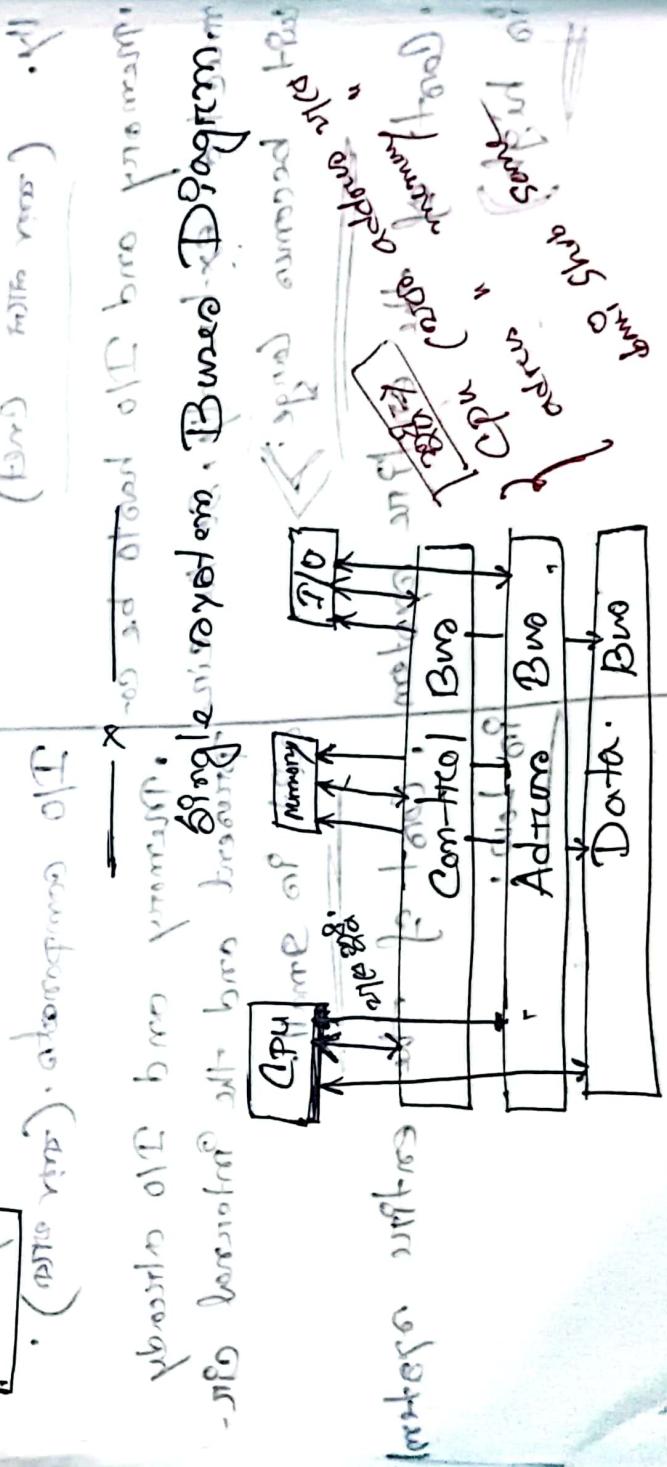
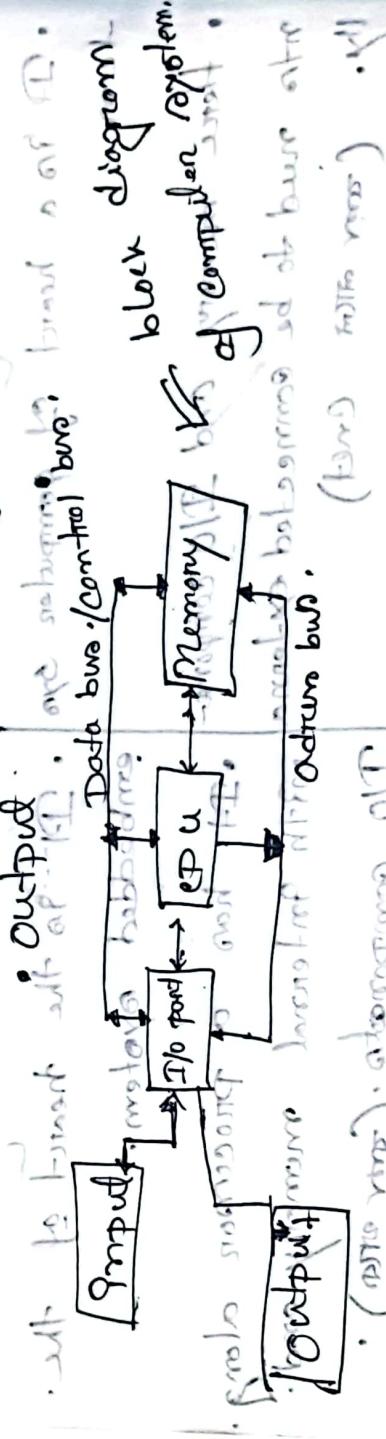
Computer system: consisting larger logic board computer system. architecture of computer system consists of Central Processing Unit (CPU), memory (Rom, Rom), input / output : (I/O). unit - by connected 8 kind of bus: (Data, Control, address) connection, interface.

Basic computer system consists of Central Processing Unit (CPU), memory (Rom, Rom), input / output : (I/O). unit - by connected 8 kind of bus: (Data, Control, address) connection, interface.

Computer system: • output
 • storage manipulation.

Memory : DRAM Processor controller

smaller products of Ram, Ram



#System Bind: 981119100000000000

- The QPS used for connection between the various Point and Handover of data between the various Point.

• Procedure of communication with memory and P/O procedures.

• And also using original, direct & changing route numbers.

• Along a set of offices called Buses, / branch of routes and numbers

B kind of Buses / signal point or bus stop.

• Data Bases/Signal

- D-drum bundle signal and are stored in board. Control knobs / signal.

卷之三

- Data Bus : used for exchange of Data between the processor, memory and peripherals. (RAM, I/O)
 - I/O Bus : Input/Output bus - dedicated bus method of allowing data exchange in both directions such as between CPU and I/O devices.
 - The maybe, 8, 16, 32 and 64 bit buses

Two 50-gram rolls. Quicksilver & silvered connected
to data base bus.

Adrin Bus: • Conditions of 16, 20, 24 or 32; 16, 20, 24 or 32

- Information transfer takes place from the microprocessor to the memory or I/O elements.
- Input addressable by memory bus or I/O bus for 16 bit address bus, I/O can generate 2¹⁶ different possible addresses. Each one of these address represents a definite memory location or I/O element.
- The unidirectional bus connection is shown below.

Control Bus: • The control lines that select the memory or I/O and cause them to perform read or write operation.

Processor: • The LMD or MMIC Processor, it does all the processing, controlling of external devices and peripherals like keyboard, monitor, printer, etc. It also performs arithmetic and logical operations. It has program counter, stack pointer, instruction register, data register, program memory, etc. It also has data bus and address bus.

External Data Cache

and most of the information is lost (P/M). Then what about

- # Function of Bus.
- Data sharing. — all types of buses transfer data in microprocessor.
- Direct sharing → data to be read from specific memory beth.
 - Between memory & processor → memory order of execution is supplied by processor.
 - Between (A \rightarrow C \rightarrow D) → global clock after Dic_{sd} .
- # Basic operation of microprocessor on machine cycle.
- ① Fetch → main memory takes instruction word from memory.
- ② Decode → instruction (or op) → control microprocessor (or microcontroller)
- ③ Execute → op to be executed.
- # Expansion Bus type:
- PCP - Peripheral Component Interconnect
 - PCI - Personal Computer Memory Card industry association
- # Gap Access Extended Graphics port - not part of standard microprocessor bus → graphics card.
- SCSI - Small Computer System Interface - PC is standard bus.
- This is the circuit boards there can be hundred's of them.
- a computer system to give extra facilities.

what's the operand and operator?

- Operand is the input part of an instruction that tells the computer what function to perform. (also called operation code.)

• Operand is the instruction that describes the op.

- Operand is the instruction to be performed. It indicates what operation 2nd part of instruction, sign indicated.
 - Operand Computer system where to find the data or where to store data.
- More, add GET op codes/ operation code]
- More, add [1+2, 2, 2 " operand]
- [4 or 2nd operation] ,
longest op codes + TOT
op code : some destination source,
can number by memory

Cpu is a brain of computer. Cpu (cpu) have 30 computing and decision making ability. So, it is a computer. Computer is a brain of computer. So known as brain of computer.

Fetch - Decode - Execute

Fetch

CPU collects the instructions from the main memory and put them in CPU register. This unit is also called Instruction Register.

Decode

When instruction reaches from Processor register, CPU decodes the instruction and sends memory signals and address to ALU.

execute

ALU performs the data with arithmetic and logic operation and gives a result according to the instruction. Fetch \Rightarrow memory \Rightarrow Processor register \Rightarrow ALU \Rightarrow Memory \Rightarrow output. Decode \Rightarrow memory \Rightarrow ALU \Rightarrow output.

Execute \Rightarrow Program flow of modules under each stage. i.e. Fetch, Decode, Execute and Write back. In each stage there is a control line.

Memory \rightarrow Read and Write.

Intel 8086 Microarchitecture - Architecture

Features

- Has 16 bit microprocessor.
- Can process data and memory addresses that are represented by 16 bits at a time.
- Having 20 address lines, 16 data lines provides up to 1 MB of storage.

• 40 pin of integrated circuit. / 16 bit flag.

• Clock speed 5MHz.

• Has 2 independent functional units to carry out instructions.

• Bus interface unit.

• Execution unit.

- Fetch to bring instruction from memory.
- Read data from port and memory.
- Write to data to port and memory.

→ Tells the BRU where to bring. (Fetch)

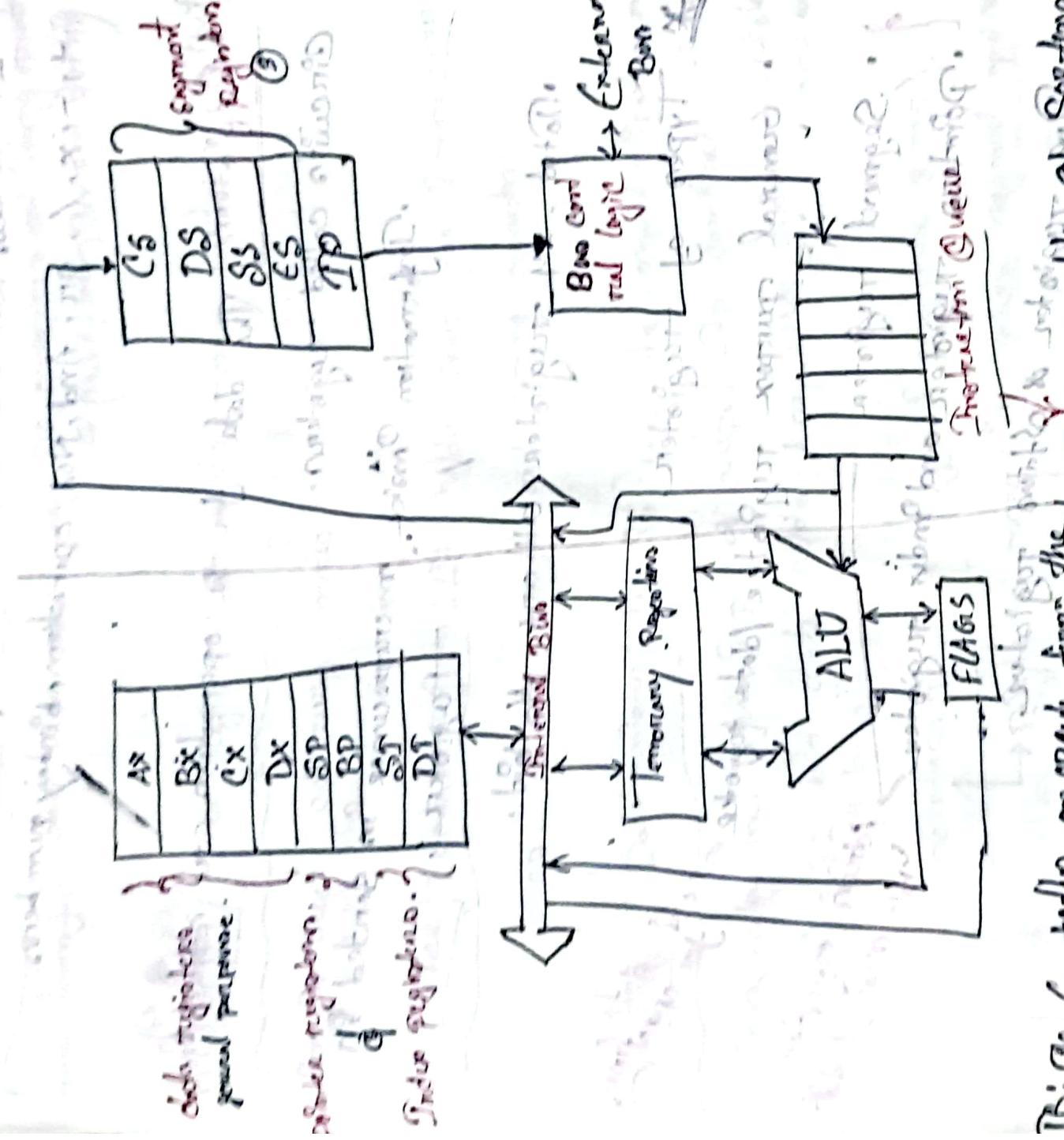
- Decodes instruction.

- Execute instruction.

Execution unit (EU)

Bus interface Unit (BIU)

Used for memory access, I/O and system control



Tr: EU gets its opcodes from the Instruction Register.
It is capable of getting information from the Memory and I/O units.
It sends the information to the Control unit for execution.

All = An arithmetic "logical unit".

TJ handles all arithmetic and logical operations
+,-,x,/ , or, And, Not operations, binary numbers
live

- Regions: The data for the operation is grouped called regions.
 - Transformation of microprocessor on board in a aircraft

- Total by registrant all one 16 bit.

- Types of registers
 - General purpose register / data register, hold your operand
 - Segment register and index register, hold address
 - Processor register and control register, hold control

cont'd. (1) - separation & desorption

Open and - seen no one
for exercises - exposed
the ~~posterior~~ ^{posterior} part of the body
and the ~~anterior~~ ^{anterior} part of the body
was covered by a thin skin.
The body was about 10 mm long.
The head was very small and
the body was very long.

14

General Departmental Budget : (1980-81) — low budget

Ax[14][15]: accumulation → orthodoxy, logic, and

data - trans - fer. and also used for 50% operations and shifting operation
to (hand) jacking/jetting system

$B \times [BH | BL]$; Bone regeneration; It holds memory address (offload)

on and used during modes.

Ch 14: Consider regulation: It holds council for construction

line loops, totale, half and other
parts opened.

→ changing the
number of nodes
→ changing the
location of nodes

... & produces more and more of what it wants

~~data held here when division multiple
done were~~

• Tigray and Tigrayan together: (official language of Tigray).

→ Points to memory in check segment and code

Mr. Savince under rugger! Storeroom the offal adreas' of

The ~~Soviet~~ Chinese and Cambodian
Nations have
been fighting for
the last 10 years.

General purpose / data register : (A, B, C, D) - low
16bit 8bit 8bit
Ax [AH] AL : Accumulator to arithmetic, logic, and
data - register.

(and also used for I/O operations and string operation
multiplication/division)

bx [BH BL] ; Base register : It holds memory address (offail)
in quadrupl addressing modes.
cx [CH CL] ; Counter register : It holds count for contraction
live loops, rotate, right and left shifting operation.

dx [DH DL] Data Register : It is used for multiplication and
division . also used for input & output . It is combined with
 $dx (16+16) = 32$ bit data held here (when division multiplication
and division not here). Example 2 register 32bit.

④ Index and pointer register : (offail register 32bit).
→ Points to memory in stack segment and code
→ Points to memory in data segment.
→ Points to memory in extra segment.

⑤ Structure pointer register : Stores the offset address of
the structure. After alignment pointer stores the
size of structure.

AVT - Request. Code of Request Code of Response
Additional memory log will.

On request during execution of program : Memory Log : Temporary Address : Temporary Value : Temporary Address : Temporary Value : Temporary Address : Temporary Value

for obtaining address of the bone address of the

Chromosome segment (Cs) : Stone of the bone address of the
chromosome segment (Cs) : Stone of the bone address of the

Code Segment (Cs) : Stone of the bone address of the

Stack Frame Segment (Cs) : Stone of the bone address of the

Code Segment (Cs) : Stone of the bone address of the

Stack Frame Segment (Cs) : Stone of the bone address of the

Code Segment (Cs) : Stone of the bone address of the

Segment Register : Each Segment made with Register

Segment Register : Each Segment made with Register

Segment Register : The Pasting of program after work

Base Register (BP) : Stone of the offal address of the

Base Register (BP) : Stone of the offal address of the

Base Register (BP) : Stone of the offal address of the

Base Register (BP) : Stone of the offal address of the

Base Register (BP) : Stone of the offal address of the

Flags in Status Register

Setting individual bits. Called flag.

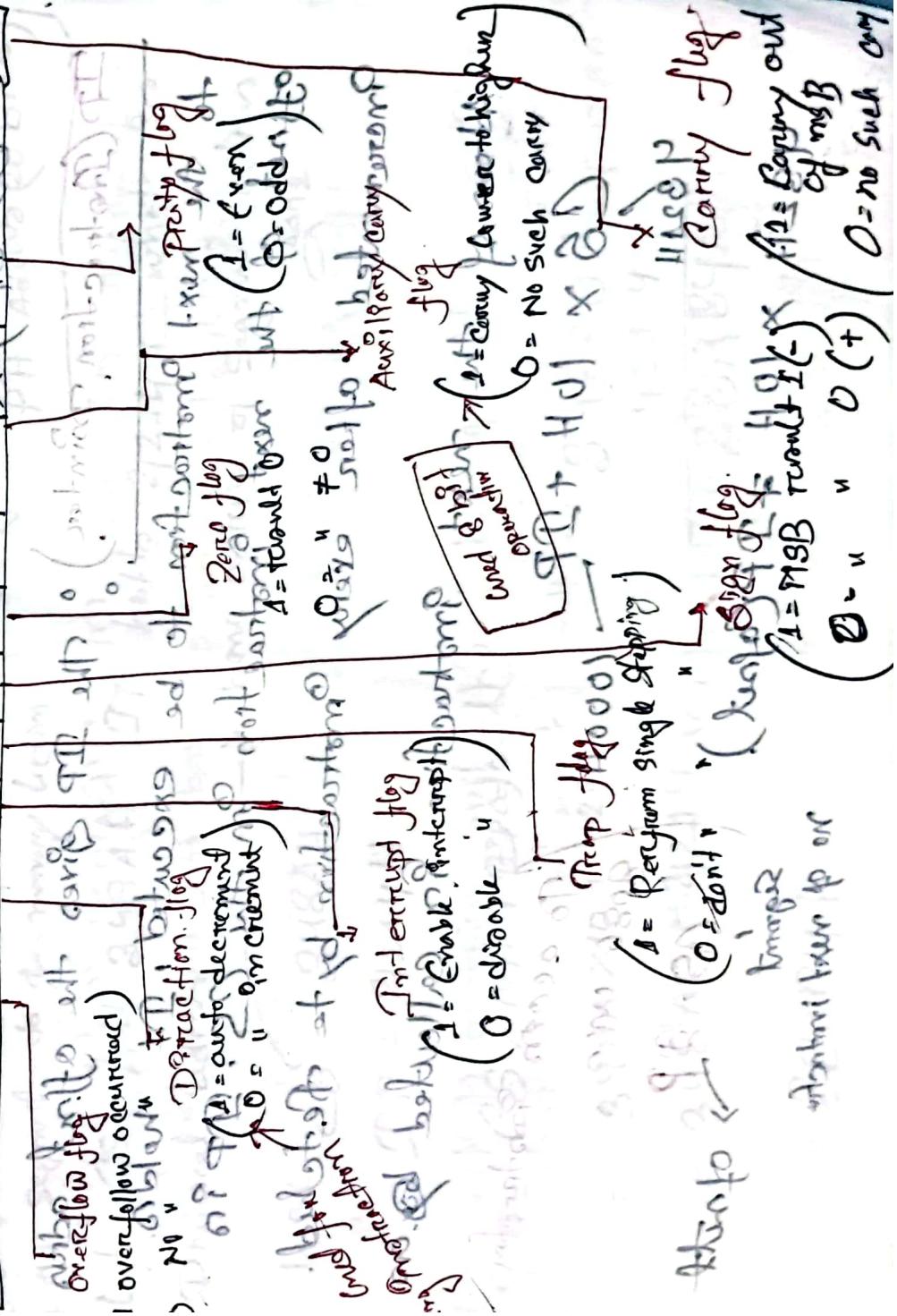
2 kinds of flag
1) Software - operation controlled by CPU

2) Hardware flags - controlled by external department in MP

9 definite flags lib. controlled by hardware
Hardware T/F, AF, ZF, OF, PF, SF, CF, OF, AF, ZF, PF, SF, CF

6 status area : OF, SF, ZF, AF, PF, CF.
3 control area : DF, TF, IF (CD), decide mode of operation

X	X	X	OF	DF	TF	SF	ZF	X	AF	DF	CF	X	NP	PF	CF
---	---	---	----	----	----	----	----	---	----	----	----	---	----	----	----

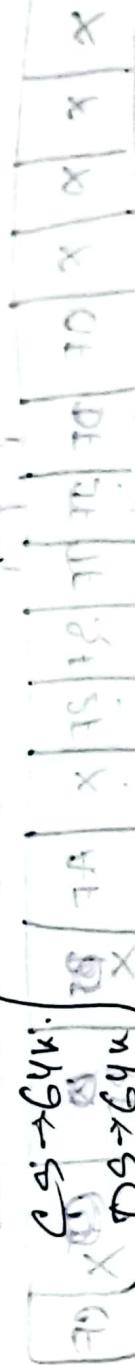


Segmentation:

Program 2¹²⁸ = 1MB
by send

The 8086 CPU has 2¹⁶ words physical address to access 1MB memory location. But register of 8086 CPU hold logical address 16 bit. So main memory of the computer is divided into different segment that may be up to $2^{16} = 64\text{ KB}$ long. In 1 segment.

ES → 64K
SS → 64K
CS → 64K
DS → 64K



IP (Instruction pointer): The IP gives the offset address to the next instruction to be executed. It holds offset of the next instruction on the CS. IP is incremented after every instruction by 1 to fetch next.

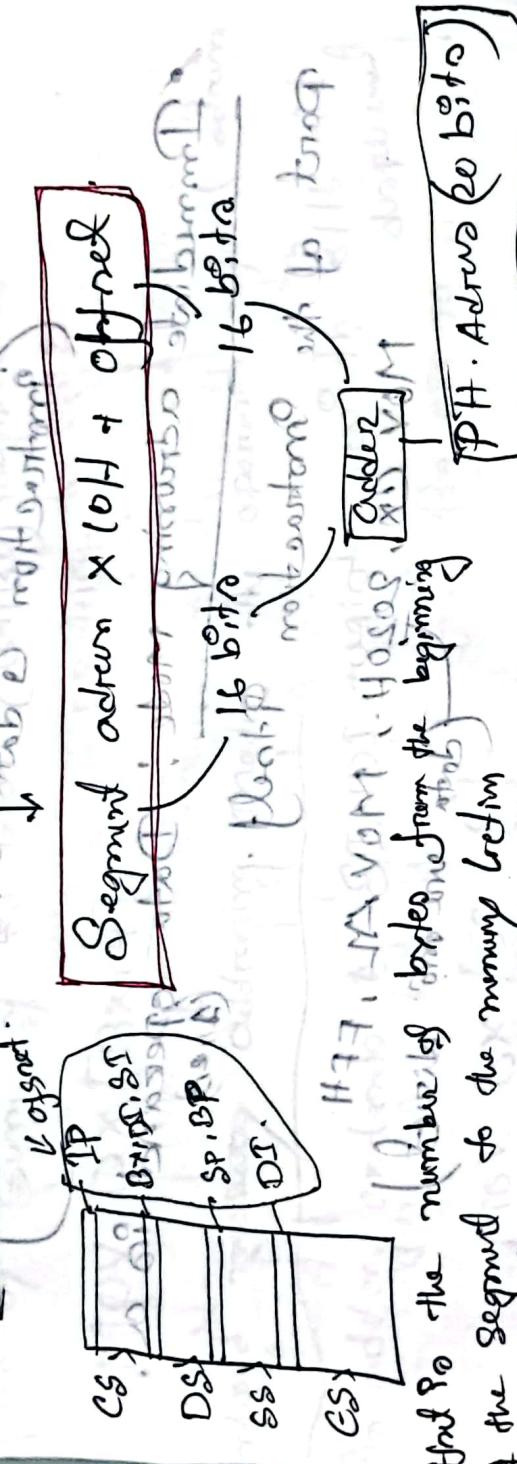
Address of the next instruction is called base address of the segment.

CS × 16H + IP → 1000H
43211
To access program
CS, DS, SS, IP
no of next instruction

offered address: It relieved any location within 64 km by the man.
only segment.

and segment address on the beginning address of 64k but
with division of 16 bit word.

Actual address of Physical Address (1020 10th)



Find 90 the number of boxes from the beginning
+ the segment to the morning before

Let CS 348A115 T-1 hold 2/24/4 whole year

lure cost = 348411

$$TP = 21244 H$$

abam gianwibbo Lantict.

$$= 348A \times 10 + 4214$$

multisets 348A001 4214

= 348 [A08+2] 214. vom c. H. C. G. B. G. A. B. C. M. T.

$$= 38ABCH.$$

~~Microprocessor~~ ~~Microcontroller~~ ~~Microcomputer~~ ~~Microsystem~~ ~~Microelectronic System~~

Addressing modes

Indirect addressing mode
In which the processor
works in the memory
in which the program
is stored.

Memory addressing mode
in which the processor
accesses memory to store
and retrieve data.

Instruction & data register.

Direct addressing mode: Data operand
is directly present in
the instruction field.

Part of the instruction field.

MOV CX, 2020H, MOV AL, FFH
(direct address)
Instruction field contains
data of memory and
memory address of
register CX.

Register addressing mode: Register CX is the source
and destination of an operand for an instruction
and doesn't require memory and
register.

MOV AX, BX [16 bit]. 16 bit
instruction = 32 bit
format

Hence = 32

Direcet addressing mode: Effective address of the
memory location is written
directly on the instruction
field + 01X A81F8 =
1154H

MOV AX, [1620H], MOV AX, [0300H] 8 bit =
10 bit
16 bit memory location and
16 bit memory location (8 bit)

Register Indirect addressing mode: addressed at any memory location through offset held in following reg ($\text{BP}, \text{BX}, \text{DT}, \text{SI}$)

MOV Ax [BX] → 0011 1100 0010 0110
Base addressing mode: the offset addressed in part of base and given by sum of contents ($\text{BX}[\text{BP}]$) register

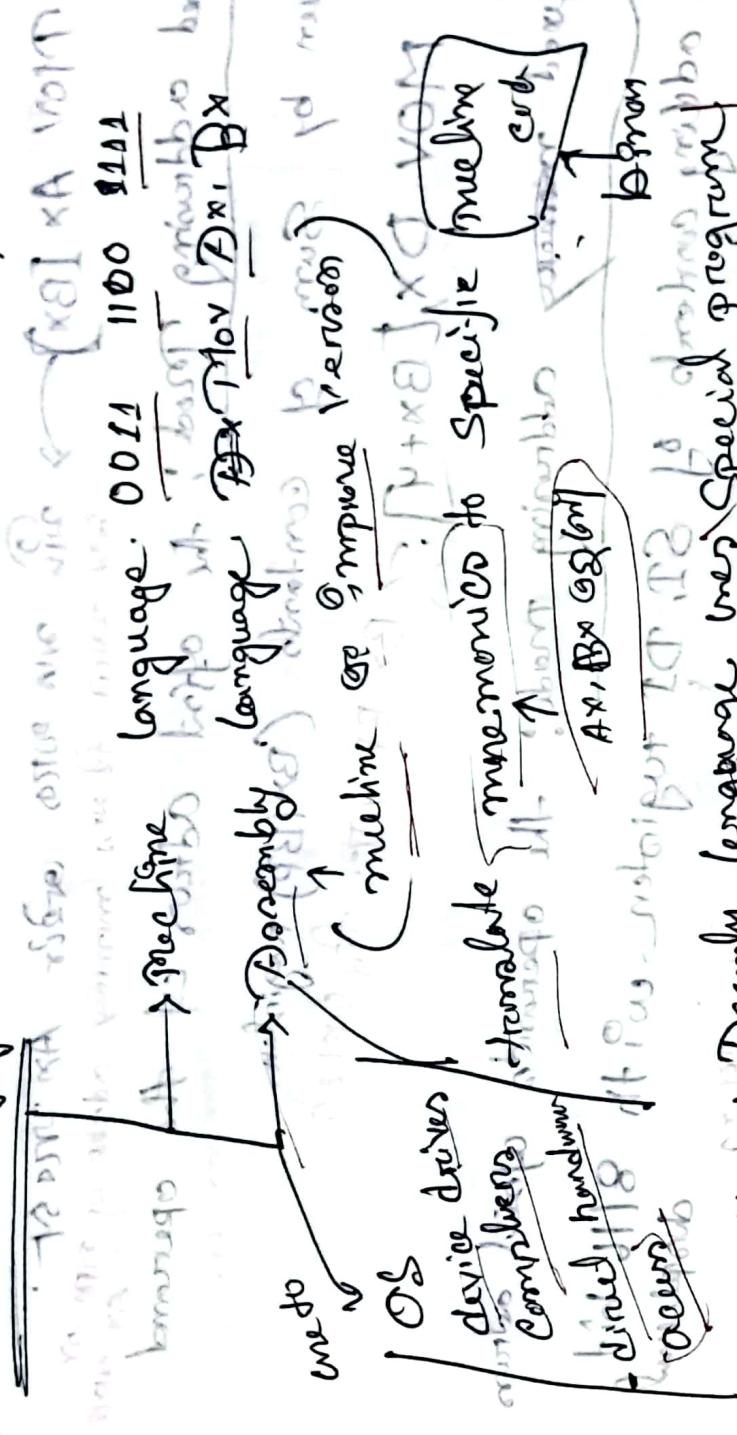
MOV Dx,[BX+4]. → BX BP PTR for QFL 20
Indexed addressing mode: The operand is obtained by adding contents of ST, DT register with the displacement

MOV V, BX [ST + 6]
Combined index memory addressing mode: Summing the base register to the contents of and indexed register with index register plus displacement.

MOV Ax, CX [AX+SI]
Based index with displacement: adding base and index register with displacement.

Index register with displacement 8/16 bit.
ADD Ax, [BX + SI + 16], MOV Array [$\text{BX} + \text{SI}$], DS

low level bombing
~~(DIAID)~~
targeting of infrastructure
to burthen, banal, predictable / DIAID
assessable



Low level language:

Assembler: ~~ESSANG~~ 01-~~3483~~

Program, Global and Local
Program: A set of instructions which
tells the computer what to do.
Local variable: A variable which
exists only within the function or
block where it is defined.
Global variable: A variable which
exists outside all functions and
blocks and can be used anywhere
in the program.

FRAN VOLLMER
12 + X = 15
Opinion of ground
communist

Now when $\sqrt{Bx + d}$ is divided by A , we get $\frac{\sqrt{Bx + d}}{A} = \frac{1}{A} \sqrt{Bx + d}$.

Assembly language

Rules \Rightarrow ① Only one horizontal line per line.

- ② each column contains of a sequence of

Assembly ending

Each statement is a combination of ~~multiple~~ words

(so obviously) ① an instruction or label $ax, bx, cx = a - b$

- ② An assembler directive

↳ motivates the assembler to perform some

specific work

Allocating memory space for a variable

- Creating a procedure

for main global procedures

name field : Assembler translates names into memory address

→ can be a - separator, letter, digit, -, ., \$, @,

obligatory blanks & not begin

blanks are not allowed

- Period must be the last char

- different to all

Illegal names:

- Two words
- Two words with a hyphen
- Period or space or both
- Period or space or both of them
- Period and space
- Period and space and a symbol
- Period and space and a symbol and a number

Syntax

- Don't mention word mnemonic (MOV, SUM) ↑ operation order (opcode), if it is not assembly directive.
- "Pseudo operation" → no machine code (Pseudocode)
- Assembly language →
 - No main procedure or procedure.
 - No goto, loop, etc.
 - Assembly language is not assembly language.
- Simple operation
 - Nop - no operand statement.
 - Pseudo operation, one operand - A and may be two operand fields.
 - AD Ax2 → 2 operand fields
 - Information from source (usually not modified)
 - Information to destination (modified)
 - And has with add form of binary.

Opand field: For an assembler directive

(usually contains more information about the assembly language).

more descriptive, with sub-fields

Command field: Stand with Semicolon.

- low level language used for programming purpose
- containing each line

Program Path (Number):

1010 \rightarrow Default Decimal mode

1010B \rightarrow Binary

1010H \rightarrow hex

1,2,3 \rightarrow Octal

1 A B C D E F \rightarrow Hex

Character: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Char Segment
char

• Assembler translate char to ASCII value

Byte Variable

Assembly directive BYTE stores initial value in memory.

Assembler directive

Name DB Initial value, converts if any
Ex: ALPHA DB Initial value stores initial value in memory.

Ex: BRANCH DB Initial value stores initial value in memory.

Word Variable

Name DW Initial value, stores initial value in memory.

Ex: WRD DW -2 stores -2 in memory.

Assembly directive DW stores lowest address of the memory area.

Ex: WD DW 1052H stores 1052H in memory.

High byte, low byte stored.

array B current

D - Array DB 10, 25, 20

Symbol Practical Examples Definitions

B-array 02.00 H. 10
central cell

B-ary tree 1 B-ary tree 2

B-ary tree 3 B-ary tree 4 B-ary tree 5

~~(Das) 6. 2. 2000~~

Some variables cannot be set

From

Inventory DWS 10000000

Inventory Items about DWS
at
entity 10000000

DB Only

entity color white and green

EGS Constant

Name EGS Constant
HARDWARE
Nominal Constant

LF EGS OAH

↳ new DL LF
More DL Only
Name

Constant

XCHG = exchange

Two register
XCHG AL BL

Thomas D. Winkler

Memory to memory move \rightarrow at $\boxed{\text{last}}$ \rightarrow last

Mon word 1 word 2 { pop^o, Alligat^o st fruit wind night
exchange

Q910

Mov Ax, 000H
Xchq Ax 000H

Address: 1041 20th Street business # 1
Name: John G. Fugler
Phone: 970-242-1111
Business: Business Services
Category: Business Services

Nov 04	PL mett with sub	AD	AL 20.00	crossed sub	bottoms	for sub sub
Time 02	2	1	odd			

W&G J B '41. ↗
N Englands art 2s complement
D W G 1941. ↗ under
Mead D W G 1941. ↗
• Dager.

100101 N P3 N neg
tubing

Program Structure

Program Structure

Machine language program consists of binary digits
language of program consists of binary digits
very large number of binary digits

Chloro-
Dodecyl
Sulfate
Lipid.
H₂O + HCl
Water
NaOH

- # Determined and specifying a memory model
- modif.
- directive

- Data words Down ~~twelve~~ 16 bits among MSG DB, HI → mark equal 10100' Contd.

Stack program

and to set aside storage for the stack.

Stack address \Rightarrow are offset onto his segment table up to the order

Line \cdot Stack \rightarrow indicate size of

Stack 100H (100 bytes for stack area)

Libm memory table • Home allocation,
global conformaton, Program code.

Global Conformaton \Rightarrow Program code.

variables with global name object.

Code Name \rightarrow options of environment variable

Small program Name function environment

→ made of the construction are organized by program

Procedure

Example: ~~Program~~ \rightarrow new procedure

PROC. { Name of procedure
FENDP }

Code segment

all code
Main Proc

beginning and end
Main EndP

beginning and end
Main EndP

Program structure Example

model small ; Select a memory model

stack 100h ; Define stack size 100h

data ; declare the variables

code
main Proc

beginning write here Program

Main EndP
; other procedure. & organize them.
; on procedure

end main ; To mark end of program

beginning
; some fill

EDB
DOS T
END

Short note

$$Ax = 16 \text{ b}_i$$

Application

$$\underline{RAX} = 64 \text{ bit register}$$

$\text{B}_\text{EJ} \rightarrow \text{molecul}\text{e}$

~~1500 words~~ ~~32 bits~~ = ~~Ax~~ = C

longer → longer

④ Real mode operation allowed the CPU to access the 1st 1MB of memory space. (Length 64KB)

→ called real memory

DOS " System

Real mode Address \Rightarrow Segment address + offset address
any beginning address. only location
64 kb memory

Find memory size address known are 20 bits, 34 bits
 $(8.0 \Rightarrow MB)$ 248 2^{10} bytes long.

for 2008-10

$$= 220 \div 210 \text{ KJ}$$

=

$$BC = 2^{34} \times \frac{2^{10}}{2^{10}} \times 10^6 \text{ kB} = 2^{44} \text{ kB}$$

$$= 2^{14} \times 2^{10} \text{ kB} = 2^{24} \text{ kB}$$

$$= 2^{14} \times 2^{10} \text{ kB} = 2^{24} \text{ kB}$$

$$= 2^{18} \times 2^{10} \text{ kB} = 2^{28} \text{ kB}$$

② PA (Ans)

$$2900H : 3A00H$$

$$\text{Lure } BA = 2900H : 3A00H$$

$$\text{offset} = 3A00H$$

$$PA = \frac{BA \times 10H + offset}{2^{10}}$$

$$= 2900 \times 10 + 3A00$$

$$= 29000 + 3A00$$

$$\# ① 2000H : 0000A000H$$

$$PA = BA \times 10H + offset$$

$$= 2000 \times 10H + 0000A000H$$

$$= 2A000H$$

$$\# ① 2000H : 0000A000H$$

$$PA = BA \times 10H + offset$$

$$= 2000 \times 10H + 0000A000H$$

Calculate PA of CS, DS and ES

$$\underline{SP = 0001H} \quad CS = \underline{0003H} \quad CS = 0002H \quad ST = 0001H$$

$$DS = 0002H \quad SP = 0000H$$

Ans

$$CS = 0003 \times 10H + SP$$

$$= 00030 + 0000$$

$$= 00030H$$

Code segment register
use one destination

$$DS: DS \times 10H + ST$$

$$= 0100H \times 10 + 0000H$$

$$= 01000 + 0000H$$

$$= 01000H$$

$$ES: ES \times 10H + ST$$

$$= 0002H \times 10 + 0000H$$

$$= 00020 + 0000H$$

$$= 00020H$$

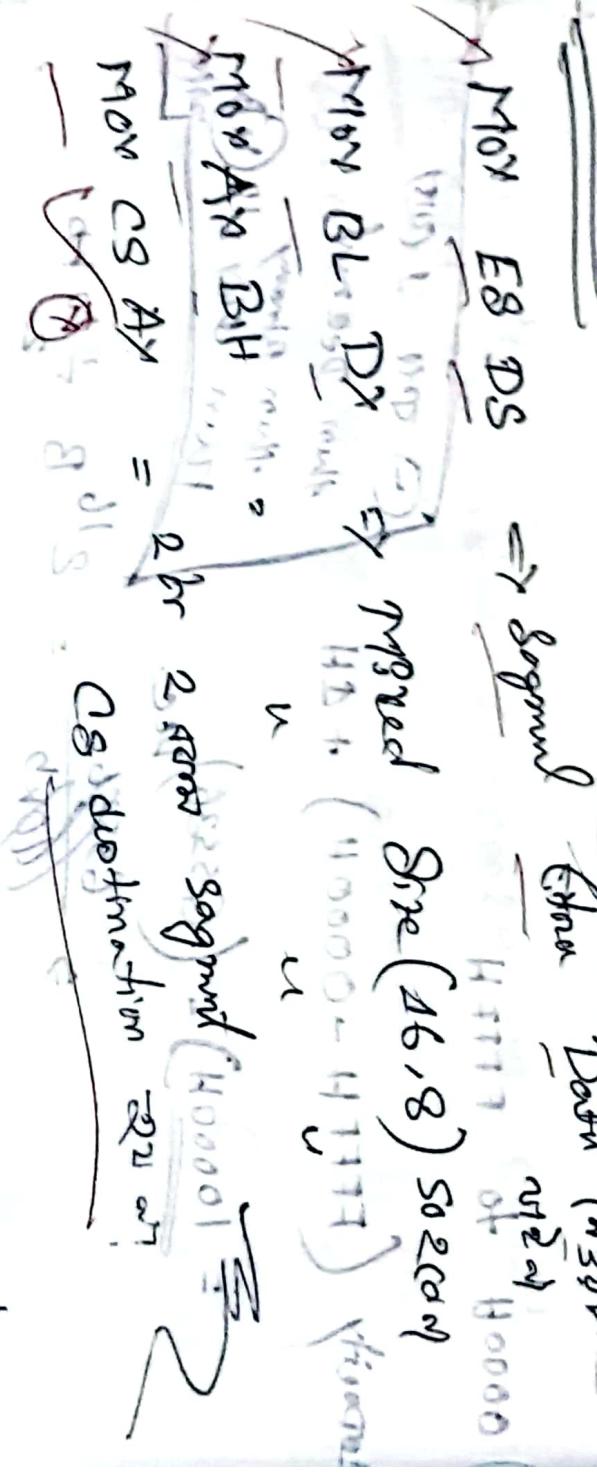
$$DS = DS \times 10H + ST$$

$$CS = CS \times 10H + SP$$

$$ES = DS \times 10H + ST$$

$$BP = BP \times 10H + SP$$

Copy word



MOV ES DS \Rightarrow Segment Base Data (1000H) of 1024H words

#

MOV BL DX \Rightarrow Merged Size (16,8) second

MOV AX BH \Rightarrow 14H. (10000H - 11111H) (from

MOV CS AX \Rightarrow 2 or 2 segment (10000H)

— OR —
BS DS . CS destination 221H

Copy the word contents of the data with numbers from memory location and destroyed by first segment memory location addressed by first

DX plus 100H onto AX

1000H.

MOV DX [BX+DI] \Rightarrow Base and Index

+ 1111H (Work pointer)

Copied the word element 0000H

1000H

of the data segment memory of the H word addressed by (BX+DI) ⑤

1000H + 100H

[BX+DI]

in DX.

+ Max DH [Bx + DS + 201] ⑧ Bone and tendon with dislodging

diaper
wet

→ Copy the byte contents of the data memory location addressed by the sum of 40000 + 8000 = 48000.

$B_n, DT + 204$) into Dried in
baking oven

④ #
D 11-2014
Date: 08/08/2014
Time: 11:00 AM
Subject: Bonding
Teacher: Mr. Sathish
Page No.: 1

$$100000 + 5000 + 3000 + 5532 = 115364$$

#MOV AL [BX] [SI]. valid

Mov ~~ES~~, Data
Code seg and use varid for destination

L	B	G	L
A	B	B	A
B	B	B	B

MOV AX 001 ← combine and ungroup
C2 in assembly

	51
50	51
49	50
48	51
47	50
46	51
45	50
44	51
43	50
42	51
41	50
40	51
39	50
38	51
37	50
36	51
35	50
34	51
33	50
32	51
31	50
30	51
29	50
28	51
27	50
26	51
25	50
24	51
23	50
22	51
21	50
20	51
19	50
18	51
17	50
16	51
15	50
14	51
13	50
12	51
11	50
10	51
9	50
8	51
7	50
6	51
5	50
4	51
3	50
2	51
1	50
0	51

Hospital address

1. All RAM data, softwares occupies during operation.
It is called global memory.

Program [B2] [ST] + 2234H

$$AS = 0200H, DP_1 = 0000H, DP_2 = 2000H, SI = 3000H$$

Memory control by 4 bytes formation 81A034H
which quantification we can diagram.
with displacement

H Based and orderly addition

$$PA = DS \times H + BX + SI + 2234H$$

$$= 1000 \times 10 + 2000 + 3000 + 2234 = 17234H$$

Value [AS] JA VOMT

Value [BX]

Value [SI]

Value [H]

Value [DS]

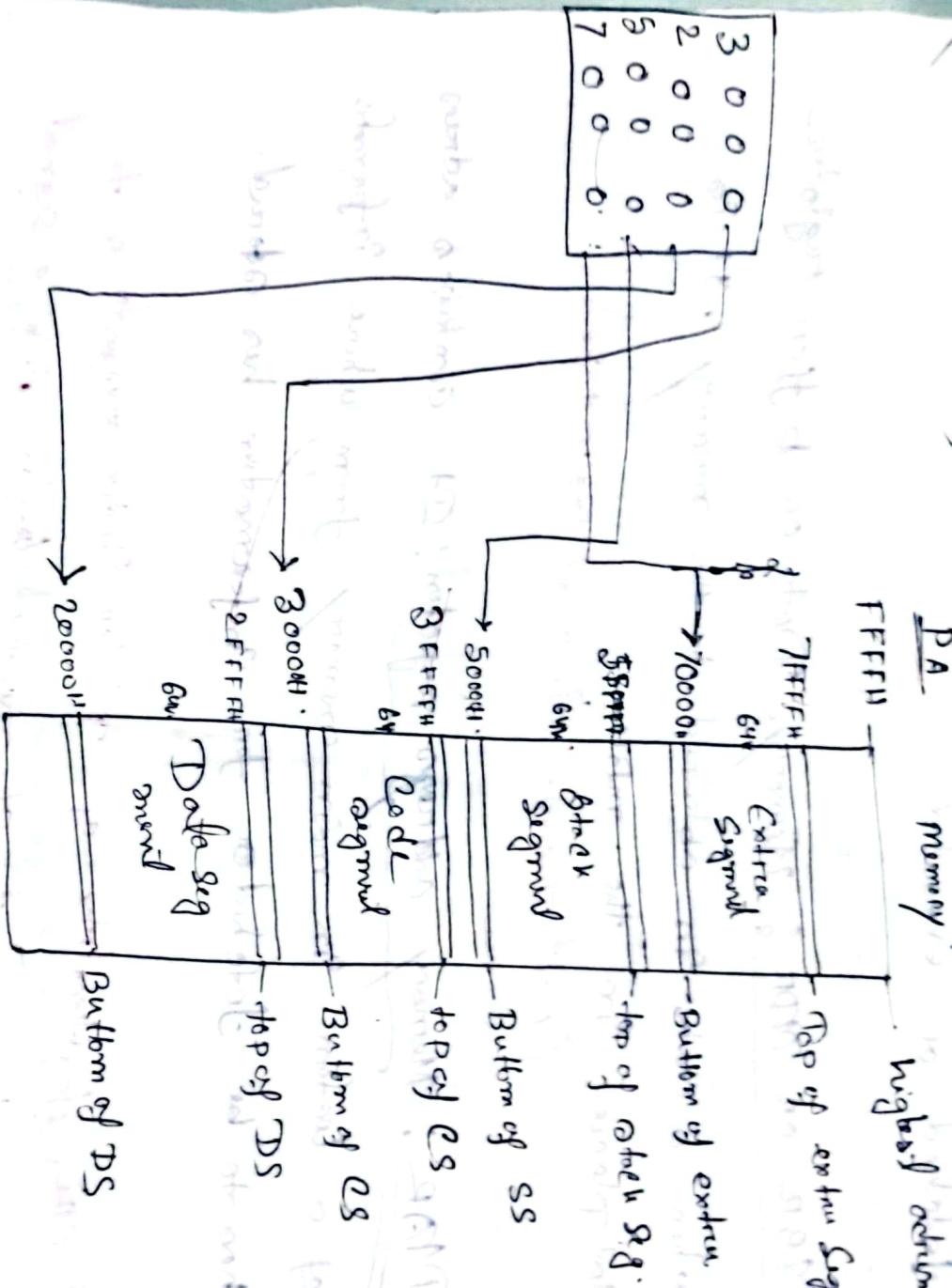
Value [BX]

Value [SI]

Segment register with diagram

base address of instruction

and data in memory



Essential registers for protection (exception)

Program Counter: PC is updated by the CPU after each instruction. So that it always finds the next instruction to be executed.

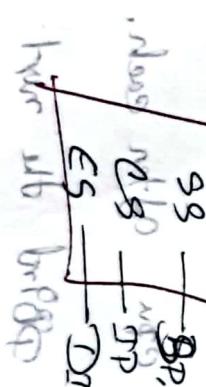
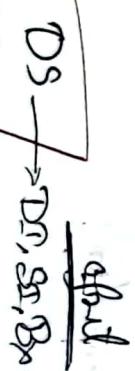
Base address of next instruction. It also has to be updated.

Structure triggered It contain the memory recently fetched or executed.

MBR on MDR Memory Data on buffer register which is data - is obtained to the memory. It is placed on the MBR unit (Memory Buffer) (most recently used (MRB))

PAR Memory address register: DA contains address of a location in main memory from where information has to be fetched & information has to be fetched for instruction.

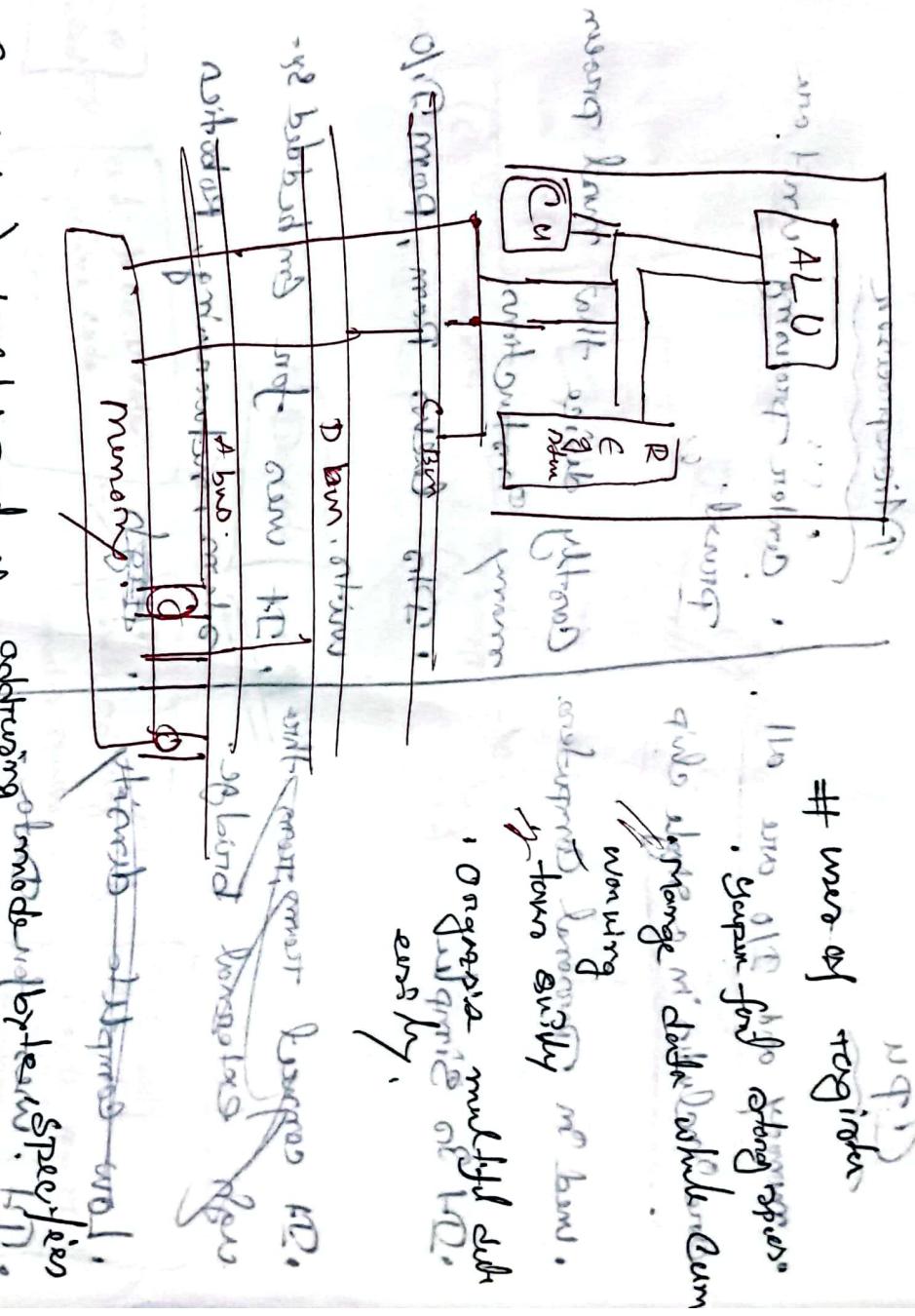
The functionality of pointer Order trigger in the other the offset of memory



CPU fetches data from memory containing address of memory.

Fetch - decode - execute mode.

וְלֹא יָבֹא כִּי־בַּעֲדֵךְ תְּמִימָה



user of tag index

Geopon found strong species.

July 20th 1909
M. D. Johnson

• pure joy & excitement in him.

1. *Argania multiflora*

20

Mr. F. W. M. L. - 12.

~~Authorised for issue~~ A bnd. on 16/10/2012

~~Memory~~ ~~Brain~~ ~~Memory~~ ~~Brain~~

MOD (modulo) field: part of memory controller

The way the memory operates is

(word) binding wire (size) separating the mesh one end the other

when set 1914
in 1915

3
4
5
6
7

- 10 -

CPU vs Microprocessor

CPU

Microprocessor

- Memory and I/O are all included in a single chip

Processor

- Used on Personal Computers.

Microprocessor

- Costly device that need process many instructions.

CPU

- CPU has Ram, ROM, I/O ports

- CPU connects to memory through external bridge.

Low complexity

- It is used for embedded systems, networking, Robotics

High complexity

- Low complexity

- It is used for Data processing, web browsing, IOT,

High complexity

- CPU gets RAM, keyboard, monitor, monitor screen

Low complexity

- CPU gets RAM, keyboard, monitor, monitor screen

Low complexity

Costlier

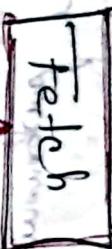
Low cost

B

A

Fetch

Decode execution



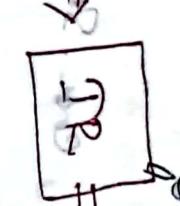
program counter to instruction register.

current instruction to be executed, hold until next.

instruction

memory

current content



instruction

next

operator

current

number of

numbers

available.

number of

numbers

available.

number of

numbers

available.



one & sorted

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)

which the opcode

=>

which addressing

mode used

in CPU

decode into

two

instruction

memory

current content

(A+B)