

\* How define processor?  $\rightarrow$  8 bit | 16 bit | 32 bit  
In terms of bits

② How many bits does the seg. of a processor have?

### 16-bit pro. (8086)

$\rightarrow$  has 16 data lines

$\rightarrow$  all "internal" registers are 16 bit long

① CS  $\rightarrow$  holds inst codes of a program  
② DS  $\rightarrow$  hold data, variable and constants  
③ SS: addresses and

### 32-bit processor

$\rightarrow$  All int reg. are 32 bit long

$\rightarrow$  Has 32 bit data bus

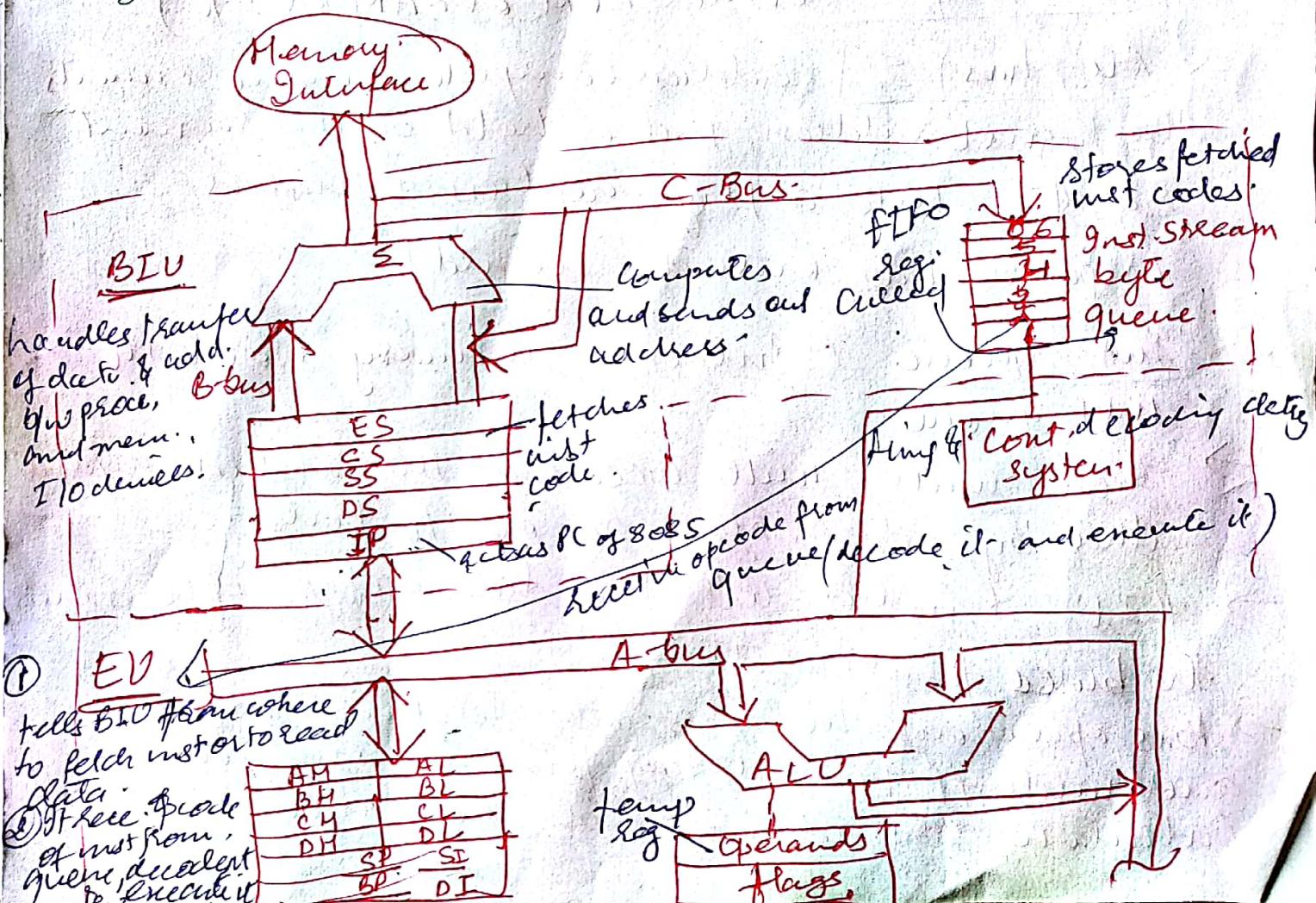
also hold contents of Reg or M[ $\cdot$ ]  
given in push inst.

ES: holds the destination addresses of some data

of certain string inst.

8086 has 16 bit data bus and a 20 bit add. bus

It can address  $2^{20} = 1\text{MB}$  unique memory.



If has two units: EU & BIU.

EU → executes inst. for the processor.

→ It consists of ALU, which performs the ALU operations required of any program.

→ Consists of GPR'S, that can be addressed either as 8-bit or 16-bit reg. → BX use in operand and destination inst.

→ 8-bit → AL      Acc. low } for all  
                  AH      Acc. high } other  
                  AX      same.

→ BX - use as offset storage to form phys.add.  
→ CX - CX use as default counter in string and loop inst.

BIU → If has segment registers used to address memory space (either ROM or RAM or I/O).

→ Also has [Σ] address compute engine converts the logical address that is held by the segment registers into a phys. address (that is exposed into outside world).

→ Inst. queue { Internal memory holds the inst. queue.

→ These units will access to the outside world either by add or data bus or I/O ports.

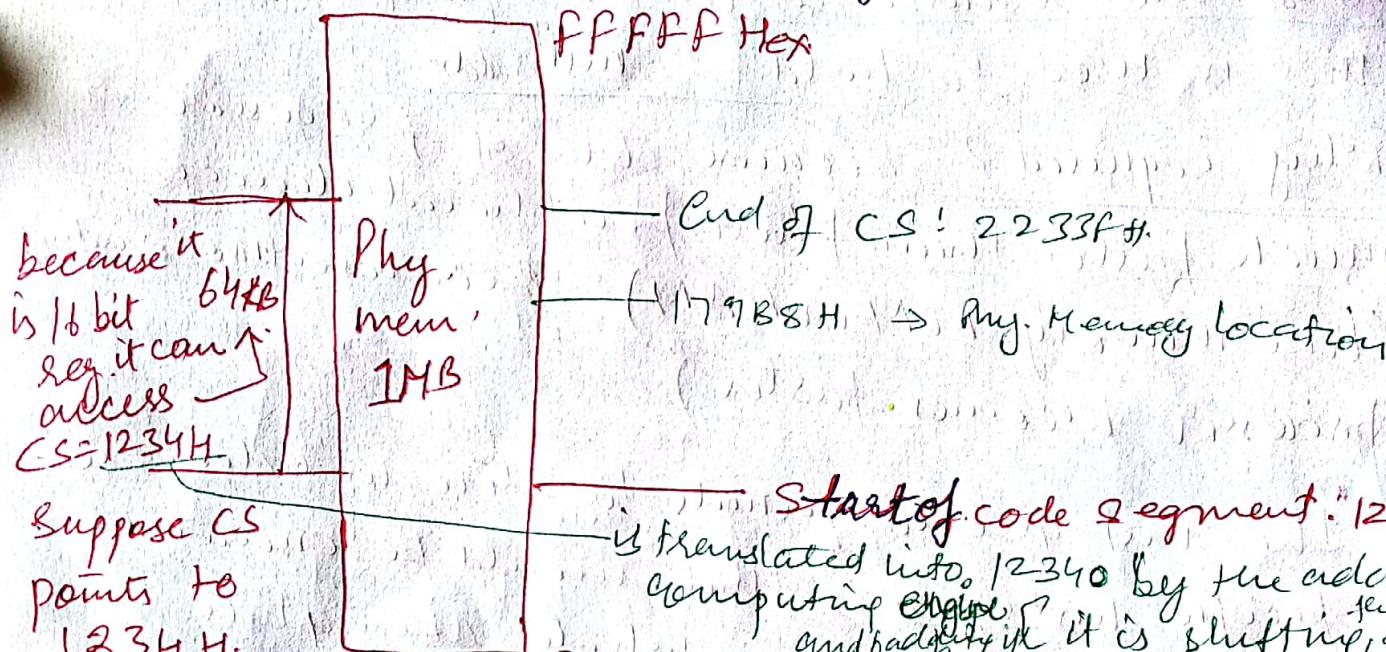
→ BIU handles all the data and address on the busses for the EU.

→ Bus operations includes inst. fetching, reading memory and writing operands for calculating the addresses of memory operands.

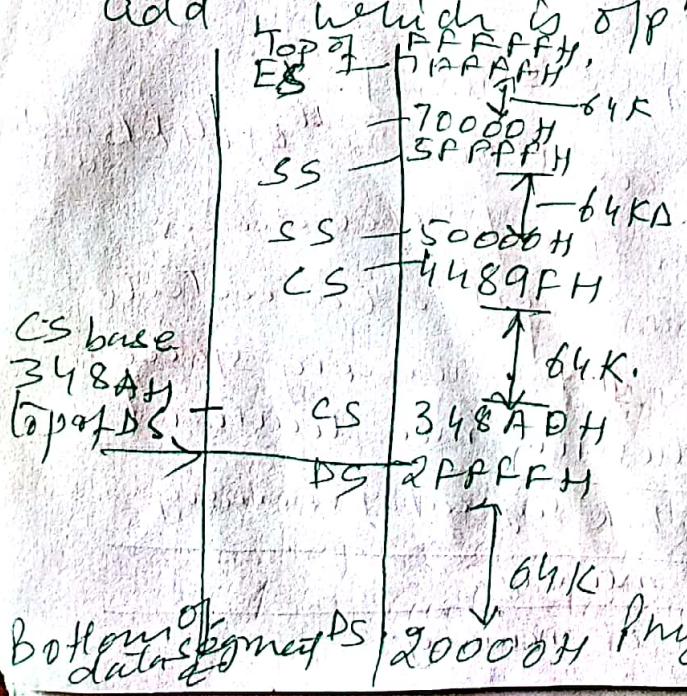
- Just are transferred to the inst queue.  
this is used to hold 6 consecutive insts.
  - Why required? Because EU of processor is much faster than other peripheral devices (like memory ROM or RAM). Because nowadays processor becomes faster).
  - The inst queue permits to store in advance specific instructions that would be executed in the future. This whole concept is called "pipelining". Pipelining is a technique wherein the bus interface unit pre-fetches instruction for the EU to use. In case of 8086 upto 3 inst. are pre-fetched.
  - It helps EU to perform at optimum speed. If only when jump inst has to execute and pipe needs to be refreshed other it will be filled in normal sequence of inst.
- Now segment registers: [ES, CS, SS, DS, IP]

- Each SR is 16-bit long and has  $2^{16}$  or 64 KB add. res.
- They are used to perform specific operations
  - different kinds of SR's are used to calculate the addresses within the 1MB memory space.
- |    |  |
|----|--|
| CS | Code Seg. Reg. use for addressing mem. location        |
| SS | Stack Seg. Reg. use addressing stack segment of memory |
| DS | Data Seg. Reg. use data Seg. of memory                 |
| ES | Extra Seg. Reg. also use data of memory                |

During addressing in a memory bank location add. calculated in two parts : Seg add + offset add.

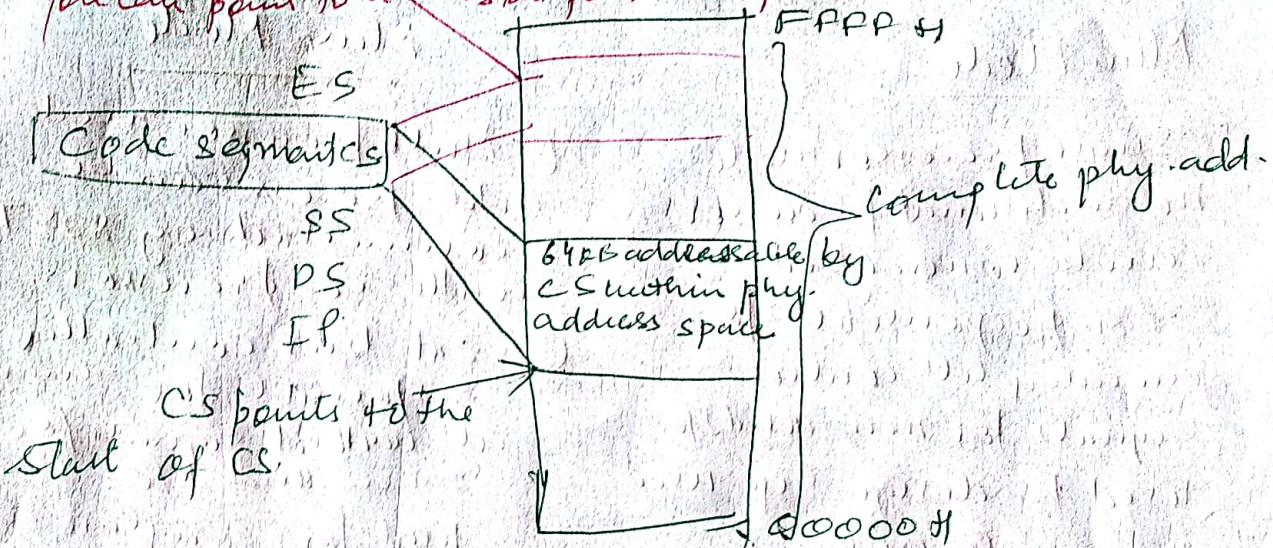


Thus, two 16 bit reg. are used to calculate one 20 bit physical add which is opt to A<sub>19</sub> to A<sub>0</sub> (20 bit long).



(3)

By changing CS value anywhere within the memory  
you can point to a new start point, anywhere within the memory space limit.



- \* EU consists of GP Registers
  - { All arithmetic and logical computation
  - { Memory pointers in DS and
  - { Other addressing modes
  - { Counter reg used in loop inst
  - { Used in divide and multiplication operations

→ Index reg: SI and DI are used primarily in string operations.

→ Pointer reg: BP → points to data within <sup>the location</sup> where the data is located in mem. location with in the SS.

→ Flag reg: Shows cond. or changes due to execution of instructions.

P	A	X	X	O	D	I	T	T	S	Z	X	A	D	P	A	C
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

## Modes of operation

### Mini Mode

- ① Single processor mode, no additional processor can be connected
- ② 8086 responsible for generating all control signals for memory and I/O.
- ③ used to design sys. units in simple apps.
- ④ pin 30 = 1 to enable min mode

### Max Mode

- ① Multi-processor mode. add-process can be connected eg 8087 auth-co-processor.
- ② ext bus controller responsible for generating all control signals for memory and I/O.
- ③ used for complex and large applications.
- ④ Pin 30 = 0 enables max mode.

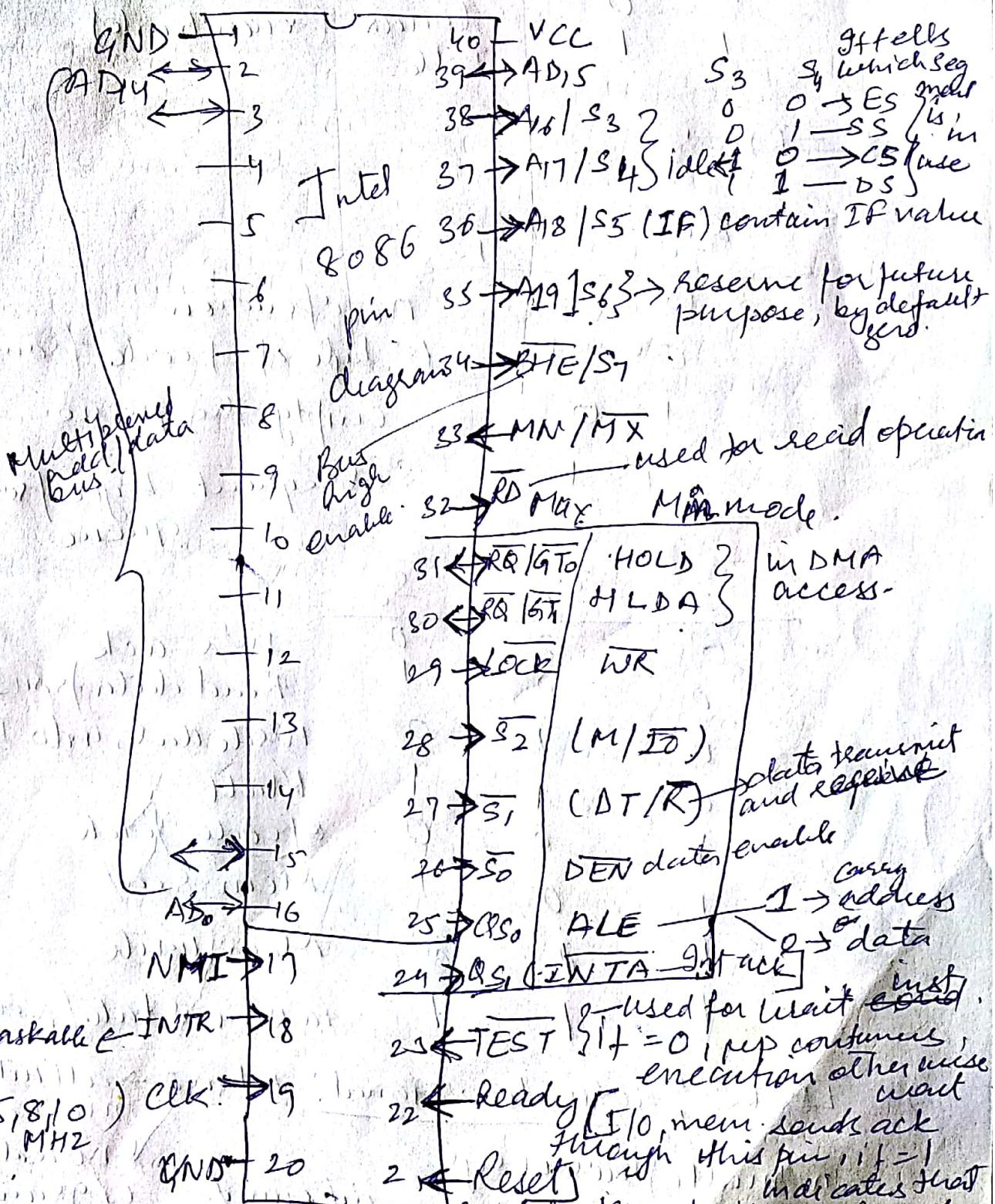
8086 flag reg: 0 → If result is > than destination reg.

- S. if result -ve, flag is set, denoted by MSB.
- Z. if prev inst is zero this flag is set.
- P. if lower byte of result contains even no. of 1's this flag is set.
- C. if carry produce in MSB this flag is set.
- T. if process in single step execution mode this flag is set.
- I. if maskable inter detect by CPU, this flag is set.
- D. if maskable inter detect by CPU, this flag is set, 0 string is process, in auto increment mode.
- AC and 1 in auto decrement mode.

①

# PIN DIAGRAM (1978) first 16-bit chip.

Add bus  $\rightarrow$  20 bit  
Data bus  $\rightarrow$  16 bit



- if 1. pps connected with E<sub>10</sub>, memory connected with Sys. bus then min. mode + 1
- 3. MN/MX  $\rightarrow$  GND  $\rightarrow$  MAX  
MN/MX  $\rightarrow$  VCC  $\rightarrow$  Min

BHE

S7

(It is used to encode data onto MSB  
half of data bus D<sub>8</sub> - D<sub>15</sub>)

0

0

whole

0

1

upper (odd bank)

1

0

lower (even bank)

1

1

idle

\* ~~RD~~

R<sub>S0</sub>

Q<sub>S1</sub>

0

no operation

0

1st byte of opcode from queue

1

empty the queue

1

Subsequent byte from queue.

S<sub>2</sub>

S<sub>1</sub>

S<sub>0</sub>

0

intr ack

0

Read data from I/O port

0

Write data into I/O port

0

halt

1

opcode fetch

1

Mem. read

1

Mem. write

1

Passive state / None

\* Lock : activation signal when all intps are masked

In multi processor sys.

All other processors are informed no hold request is granted.

by this signal that they should not ask the CPU for relinquishing the bus control.

- \*  $\overline{RQ/GT_1}$ ,  $\overline{RQ/GT_0}$  ( bidirectional)  
Local bus Priority control signal.
- Other processors ask the CPU through these lines to release the local bus.
- $\overline{RQ/GT_0}$  has higher priority than  $\overline{RQ/GT_1}$ .
- In max. mode of operation signals  $\overline{WR}$ ,  $\overline{ALE}$ ,  $\overline{DEN}$ ,  $\overline{DT/R}$  are not available directly from processor.
- These signals are available from the controller 8288 (bus controller).