*) Computer Components:

CPU → micro-processor (All components in a single chip)
memory ↔ I/O

• CPU = CU + ALU + MMU
All in Single Ship (ee)

• 4 bits = 1 nibble

→ Second micro Processor:
• Intel 8086 (June, 1978)
• Clock Speed 1K = 5MHz
⇒ (0.33 MIPS)
Million instructions per second
• Transistor = 29000
• Bus-width = 16 bits
• Addressing Memory = 1MB
↳ used in portable Computing

→ Intel made first micro-processor (Intel 4004) in Nov. 1971 and still making these processor as no other is making:
• having clock Speed 108 kHz
• No. of transistors: 2300
• Bus width 4-bit.
• Addressable memory 640 bytes

• $1MB = 2^{20} = 2^{10}_{1K} \cdot 2^{10}_{1K}$ ⇒ Address Bus 20 bits.

⇒ For assembly language we learn instruction set of 8086 micro-processor.

→ Intel 80286 (Feb 1982)
• 6MHz (0.9 MIPS)
• 134,000
• 16 bits
• 16 MB ⇒ $1MB = 2^4 \cdot 2^{20} = 2^{24}$ ⇒ 24-bits
• virtual memory 1GB.
↳ Hard disk have one GB of capacity to run a program (virtual memory).

→ Intel 80386 DX (Oct 1985)
• 16 MHz (5 mb mips)
• 275000

- 32 bits
- 4 GB $\Rightarrow 4(1K.1M)B = 2^2 . 2^{10} . 2^{20} = 2^{32} \Rightarrow$ 32-bits.
- 64 TB in virtual memory.

→ Intel 80486 (April 1989)
- 25 MHz (20 mips)
- 1.2 million ∴ 8K unified level 1 cache
- 32 bits.
- 4 GB.
- 64 TB.

*) Pentium Series:.

→ Pentium (March 1993)
- Clock Speed 60 MHz (100 Mips)
- Transistors 3.1 million (0.6 microns)
- Bus width 64 bit
- Addressable memory 4GB ⇒ 2^{32} (4.1K. 1M)
- virtual memory 64 TB
- L1 Split cache 16K (8K + 8K)
  - 8K instruction
  - 8K Data

↳ write addresses always in binary.

→ Pentium pro (Nov 1995)
- 166 MHz (7.11 SPEC int 95 ; 6.21 SPEC fp95).
- 64 bit.
- 75 million (0.35 micron)
- 64 GB
- 64 TB
- L1 Split cache

∴ 1 million = 10^6
∴ 1 mega = 2^{20}

∴ instead of using high clock processor use two processor of low clock speed to avoid heat.

⇒ write about microns.

*) Memory Banks:-

=> 8088

- Data bus 8 bit ($D_7 - D_0$) → width
- Address bus 20 bit → ($A_{19} - A_0$)

Addressable memory 1MB

$2^{20} - 1$ → FFFFFh

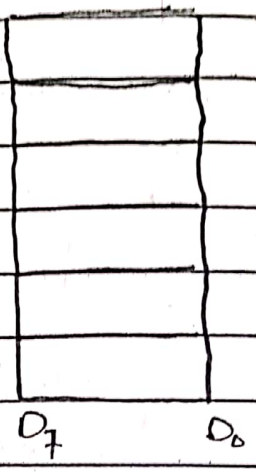
$1M - 1$ FFFFEh

00002h

00001h

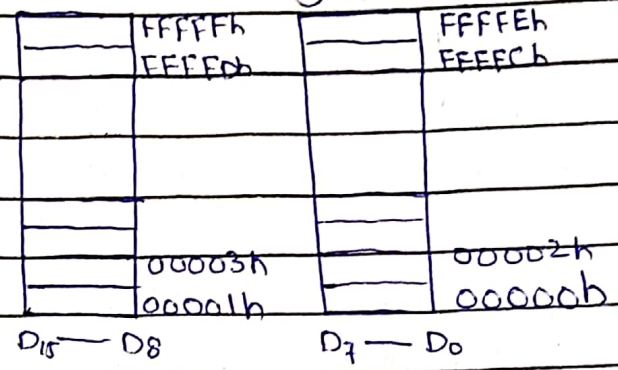
20/4 => 00000h

Hexa



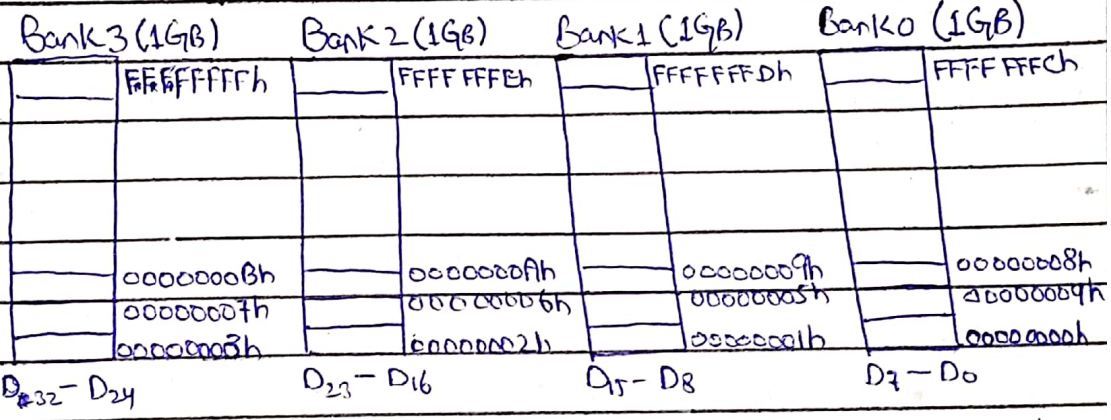
Memory banks

ii) Processor 8086 ⌜2-byies (At one time to handle)
Data bus 16-bits (D_{15} — D_0)
Address bus 20-bits (A_{19} — A_0)
(odd) (Even)



• Processor generate the process which is output.

iii) Processor 80386 ⌜4-byies (8byies X 4 Banks) = 32
Data Bus 32-bit (D_{31} — D_0)
Address Bus 32-bit (A_{31} — A_0)



Imp
⇒ why these two cycle if we start from (B_1 / B_2 / B_3) and only one cycle if start from (B_0)

*) Assignment : 1 :-
Draw memory map of following 4 processors.
a) 80286
b) Pentium pro

*) Programming Model of X86 micro-processors:

→ General Purpose Registers
→ Special Purpose Registers.

→ These registers use by programmer according to his wish that he may put data or any thing on it

→ These registers use accordingly by which purpose they made not as general

For 16 bits:

15 ... 8	7 ... 0	
AH	AL	Accumulator (AX)
BH	BL	Base (BX)
CH	CL	Count (CX)
DH	DL	Data (DX

: Here H Shows high L Shows Low and X Shows Total.

→ The above box Shows name of registers and their representation.

The above Box Continue as:-

-	-	Base pointer (BP)
-	-	Stack Pointer (SP)
-	-	Source index (SI)
-	-	Destination index (DI)

These all registers are present in all

Types of CPU like (8086/80286/80386)

Now for 32-bits :- (8086/80286/80386)

31	16	8	7	0	
EAX	AH		AL		Accumulator (AX
EBX	BH		BL		Base (BX)
ECX	CH		CL		Count (CX)
EDX	DH		DL		Data (DX)
EBP	-		-		Base pointer (BP
ESP	-		-		Stack Pointer (SP
EST	-		-		Source index (SI
EDI	-		-		Destination index (D

386DX/486DX/Pentium

→ The above both diagrams are general purpose Registers.

⇒ Now if we have 64-bit then we go upto 63-bit and Above Column E will simply replace by R ex.
EAX ⟶ RAX.

*) Accumulator :-
Usually holds one operand of of a binary operation.

*) Base :-
Usually it is used to hold the address of any data item.

*) Count :-
Holds the number of iteration for a loop.

*) Data :-
The other operand of a binary operation may be kept in data register.

*) Base pointer
usually points to the start of any data item → (Holds address of)

*) Stack pointer :-
Holds the address of top element of Stack

*) Source Index :-
Holds the address of Source string.

*) Destination index :-
Holds the address of destination string.

x) Special Purpose Registers :-

∴ They only the 16-bits registers.

→ Program = instr + Data

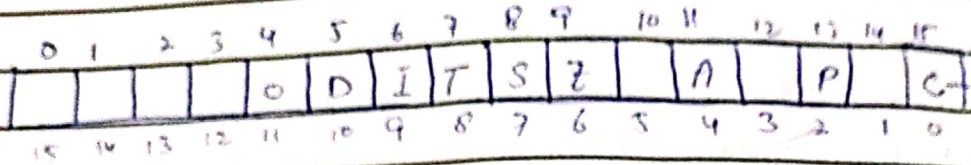
∴ when a program run Special purpose register can be change but general purpose can't be change

8086 & 80286	Code Segment	CS
	Data Segment	DS
	Extra Segment	ES
	Stack Segment	SS
		FS
		GS

→ CS points to the start of Code Segment

→ DS points to the start of data Segment

*) Flags :- ∴ SI is also a register.



↳ Describes the status of micro-processor after an operation.

mov AL :- 0110 0011 b
mov bL :- 1000 1111 b
add al, bl
al = al + bl

- (Carry Flag) CF = 0
- (Parity Flag) PF = 0
- Auxiliary carry (AC) = 1
- zero flag (ZF) = 0
- Sign Flag (SF) = 1

ob-code des, Source

↳ By Adding above :- 1111 0010 ↳ This will store in al

→ A byte will be Low order nibble or high order nibble. e.g. 1111 0010

High order Low order nibble

→ MSB represents Sign for Signed data.
eg. (0 → +), (1 → −)

= Flags =

- IF (TF) = 1 then debugging is ON
- Interrupt Flag (IF) = 1 Normally interrupts are enabled by default.
- Direction Flag (DF) = 0 Low index To high index ~~reverse~~ ← Normal
- Overflow Flag (OF) = 0 clear

→ There are Some instructions in Assembly language that check flags.

JC Jump if (CF) = 1

⇒ (Bx) will default with (DS) → Data Segment. Bx will work inside DS not outside. eg. (0 → 59) not 60 Seconds

⇒ (ST) have default Segment with (ES).

(*) Segment : offset → method to write ← write in binary

For (Bx) → bit 16 : 16

→ offset not can't be change but Segment not can be change.

• windows mode → Protected mode

• Dose mode → Real mode

• Instruction pointer flag (IP) : it points to the next instruction to be executed.

⇒ The default Segment of (IP) is CS.

offset Code Segment