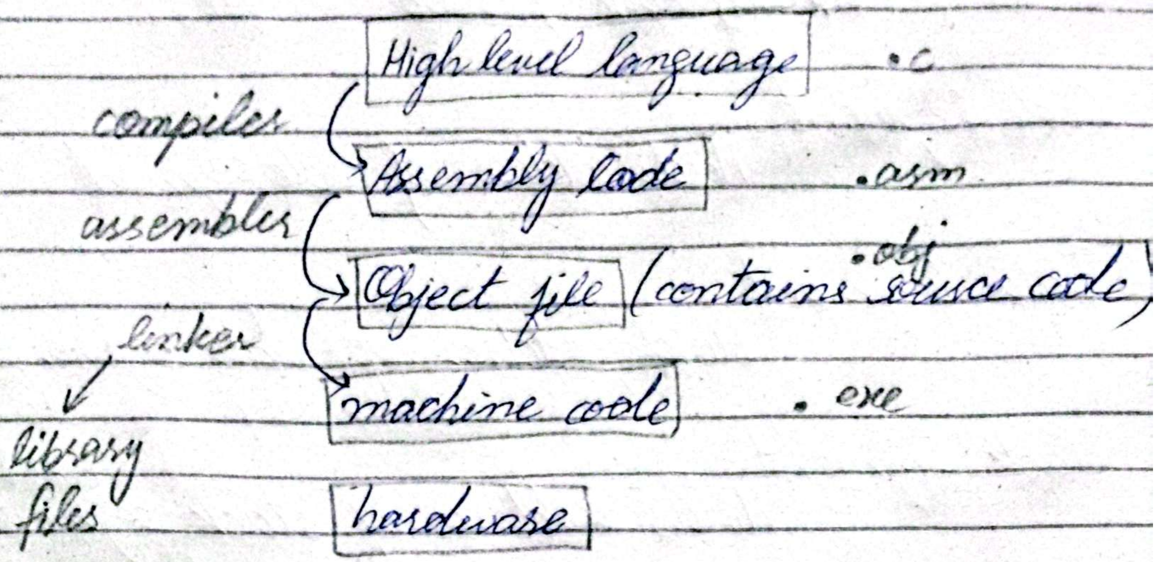
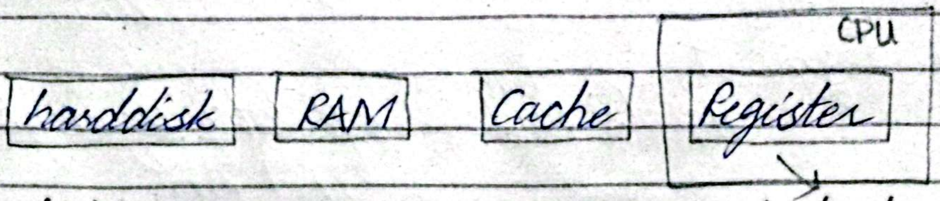
→ Assembly language

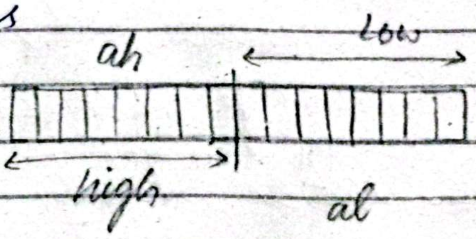




Quickly access as they are present in CPU

fastest memory location

Types of Registers



Accumulator for input/output operations

a, ax, eax, rax

X = extended to 16 bits

E = extended to 32 bits

R = Rich Register to 64 bits

Base for holding address of data

b, bx, ebx, rbx

Counter for counts used for loop

c, cx, ecx, rcx

Data for holds data for output

dl , dx , edx , rdx

Code Segment

Holds address of code segment

Data Segment

Holds address of data segment

Stack Segment

Holds address of stack segment

Extra segment

Holds address of data segment

Source index

points to source operand.

add dl, bl   add 3, bl

destination   source

Destination Index

Points the destination operand

Instruction Pointer

Holds the next instruction

Stack pointer

points current top of stack

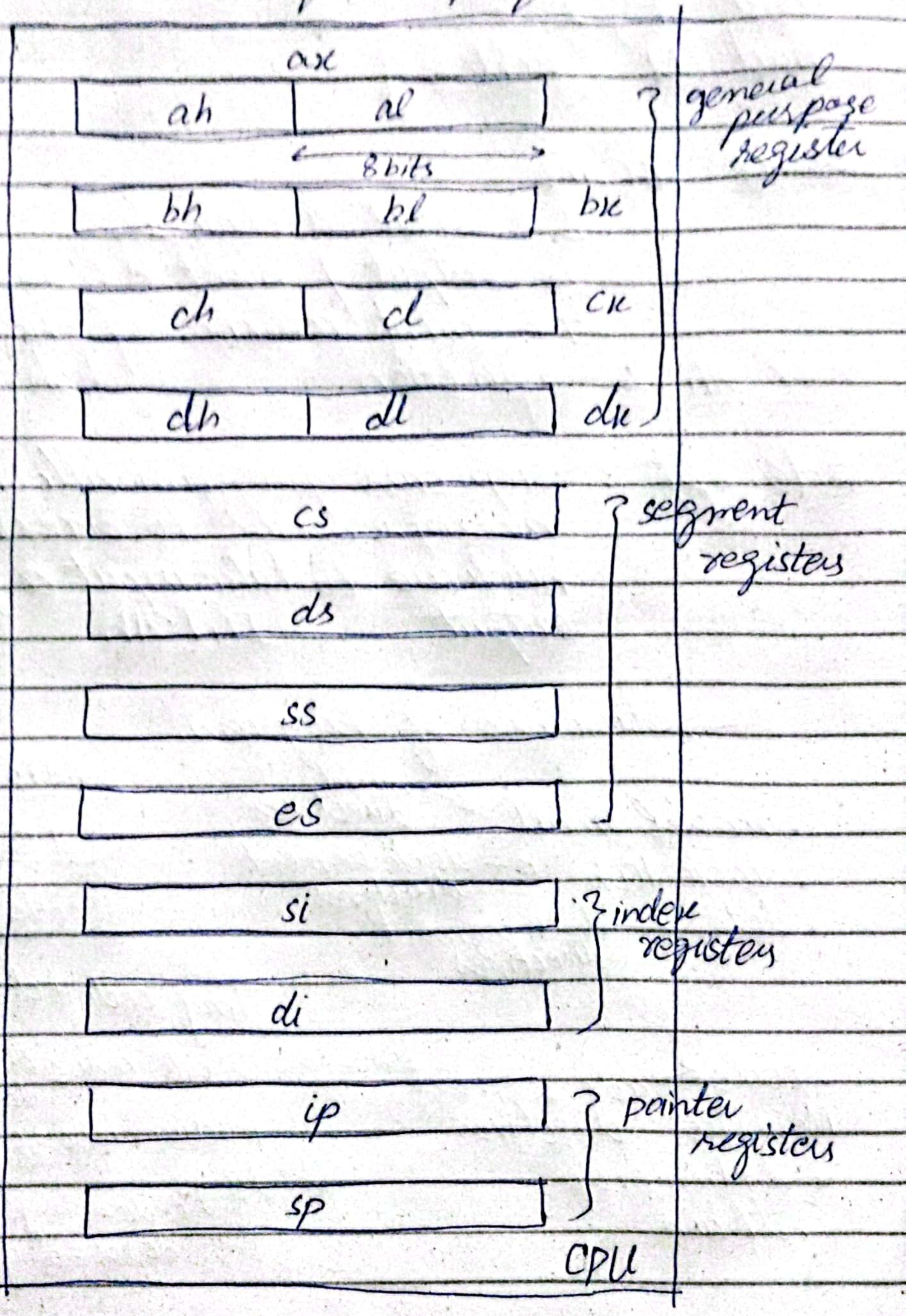
Flag registers

Hold current status of program

$$\begin{array}{r} \textcircled{1}\\ 1\\ +\,1\\ \hline 0 \end{array}$$

Base Pointer

Base of the top of stack



Addressing mods: Data Transfer Inst

Add DL, Al   mov DL, 2

Add DL, 2

Add DL, [address]

Service Routine

1 = input a character

2 = output / print a character

9 = print collection of characters

mov Ah, 2 → function   4ch = EXIT

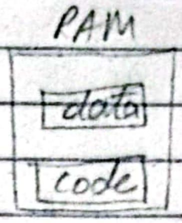
Interrupt : stop curr program allow microprocessor to access hardware to take input or give output   INT 21h

→ Structure of Program

.model small ← model directives

.stack 100h ← stack segment directives

.data ← data segment directives



.code ← code segment directives

Main Proc

~~End~~ Main endp

End Main

Tiny code + data ≤ 64 kb

Small code ≤ 64kb && data ≤ 64kb

Medium code = anysize data ≤ 64kb

Large code = anysize, data = anysize