



Computer Organization and Architecture

CPU Organization Part-2

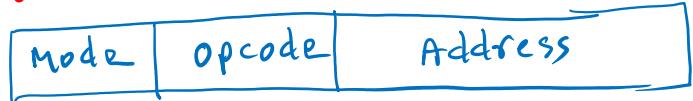
ABOUT ME : MURALIKRISHNA BUKKASAMUDRAM

- MTech with 20 years of Experience in Teaching GATE and Engineering colleges
- IIT NPTEL Course topper in Theory of computation with 96 %
- IGIP Certified (Certification on International Engineering educator)
- GATE Qualified
- Trained more than 50 Thousand students across the country
- Area of Expertise : TOC,OS,COA,CN,DLD



CPU Organization Part-2

Addressing Modes



- (1) Implied
- (2) Immediate
- (3) Register Direct
- (4) Register Indirect
 - Auto-increment
 - Auto-Decrement
- (5) Memory Direct
- (6) Memory Indirect
- (7) Relative Mode
- (8) Based Indexed Mode
- (9) Base register mode.

CPU Organization Part-2

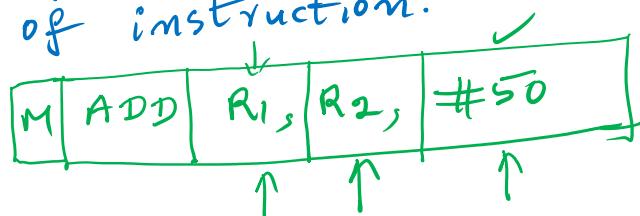
→ The way
in which we
are getting the
operands

Implied [Operands are available in some default location]

Ex:— CLA, CMA, CIA, CIL --- [Register Reference]
zero Address instruction. [Stack-organized]

Immediate Mode

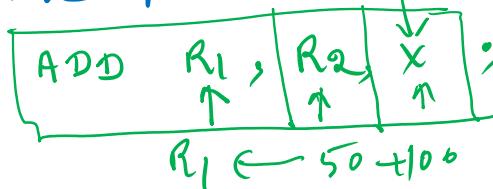
Operands are available at the address part
of instruction.



$$R_1 \leftarrow R_2 + 50$$

Register-Direct Mode

The Operands are available in the register



$$R_1 \leftarrow R_2 + M[X]$$

$R_2 = 50$

$X = 100$

CPU Organization Part-2

Register - Indirect

$$R_1 = 50$$

$$R_2 = 400$$

(50)

398	399
400	100
401	
402	
403	

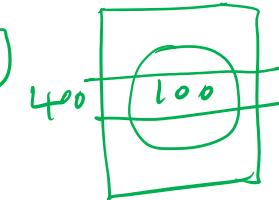
50 + 100

The Content of referred register is effective Address.

MUL $\downarrow R_1, \checkmark R_2, \frac{\checkmark}{\downarrow} (R_3)$

$R_1 \leftarrow R_2$

; $+ M[R_3]$



$$R_2 = 50$$

$$R_3 = 400$$

$$150 = 50 * 100$$

Auto-increment (or) Auto-Decrement

✓ ADD $\underline{R_1}, \underline{(R_2)} + ; R_1 \leftarrow R_1 + M[R_2]$

ADD $\underline{R_1} \underline{(R_2)} - ; R_1 \leftarrow R_1 - M[R_2]$

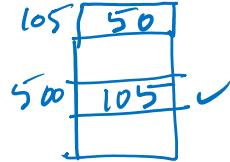
Program Loops

CPU Organization Part-2

$$\begin{array}{l} \text{ADD } 50 \\ \text{R1} \leftarrow \text{R1} + 50 \\ = 21 + (-10) \\ = 11 \end{array}$$

Memory - Direct

MOV R1, 500 ; $R_1 \leftarrow M[500]$
 ↑ ↑
 $R_1 \leftarrow 105$



Memory Indirect

MOV R1, (500) ; $R_1 \leftarrow M[M[500]]$
 ↑ ↑
 $R_1 \leftarrow M[105]$
 $R_1 \leftarrow 50$

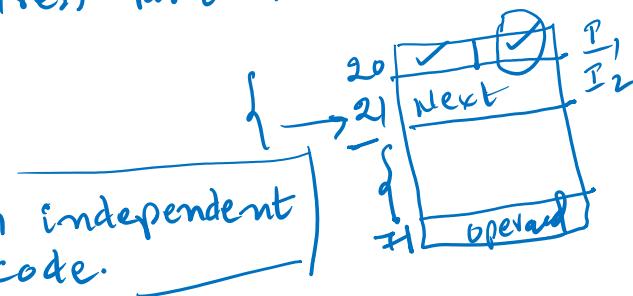
PC-Relative Mode

EA = PC + Address-part instruction.

$$EA = 21 + 50$$

$$= 71$$

Writing the Position independent code.

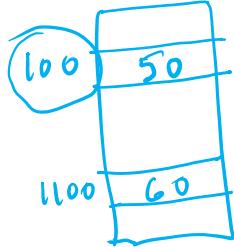


CPU Organization Part-2

Indexed Addressing Mode

$a[i]^{\text{th}}$

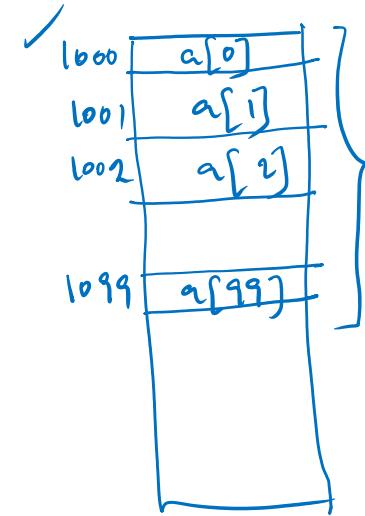
$$R_2 = 100 \\ R_3 = 1000$$



int a[100];

$a[0] \ a[1] \ \dots \ a[99]$

$$1000 + i = \begin{matrix} 1000 \\ 3 \\ 1003 \end{matrix}$$



Example :-

ADD R1, (R2), $100(R_3)$

$R_1 \leftarrow$

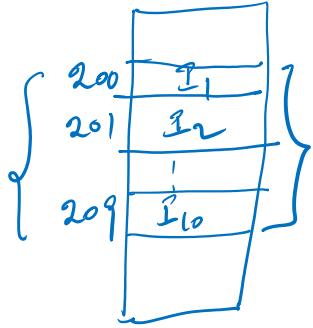
$\boxed{EA = XR + API}$

$$110 \leftarrow 50 + 60$$

$100 + 1000$

1100

CPU Organization Part-2



Base - Register Addressing Mode

this is used for writing the relocatable code.

Base Register ;— smallest phy. Address

$$EA = BR + \underbrace{\text{Address Part of instruction}}_{\uparrow} \quad (\text{Displacement})$$

$$\text{i}^{\text{th}} \quad (200 + i)$$

CPU Organization Part-2

Negative
Positive ✓

→ A Relative mode branch type of instruction is stored in memory at an address 750 (decimal). Branch is made to an address 500 (decimal). what is the relative address field of the instruction ?

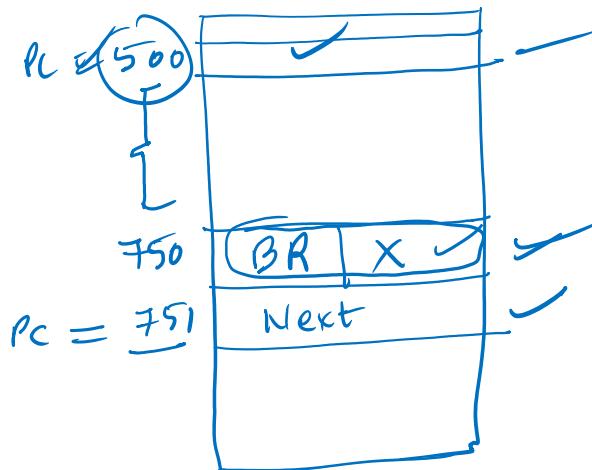
Sol

$$EA = PC + API$$

$$500 = 751 + X$$

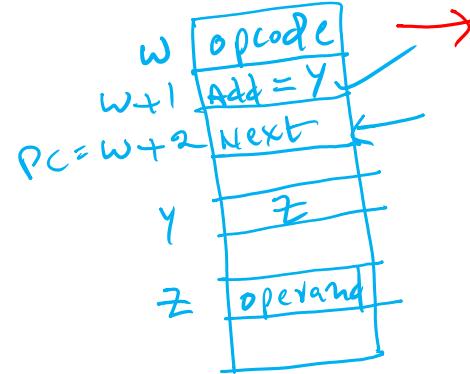
$$X = 500 - 751$$

$$= \underline{-251}$$



$$PC \leftarrow PC + 1$$

CPU Organization Part-2



A two word instruction is stored in memory at an address " w " and the address of part of instruction stored at " $w+1$ " is symbolized by " y ". the Operand used during the execution of the instruction is stored at address symbolized by z . How " z " is calculated from the other addresses if Addressing modes used are,

(1) Direct

$$z = y = M[w+1]$$

opcode y

(2) Indirect

$$z = M[y] = M[M[w+1]]$$

(3) Relative

$$EA = PC + API$$

$$z = w+2+y$$

(4) Indexed. (Assume that index register contains Value x)

$$EA = xR + API$$

$$z = x + y$$