



Computer Organization and Architecture

Programming the
basic computer
Part-1

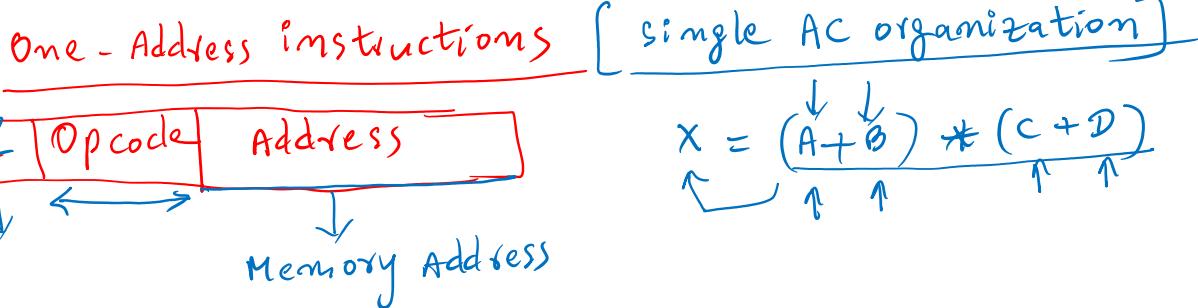
ABOUT ME : MURALIKRISHNA BUKKASAMUDRAM

- M.Tech 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



Programming the basic computer Part-1

AC
Accumulator
Register



$$X = (A + B) * (C + D)$$

A	✓
B	✓
C	✓
D	✓
X	
T	A+B ✓

LDA A ; $AC \leftarrow M[A]$
 ADD B ; $AC \leftarrow AC + M[B]$
 STA T ; $M[T] \leftarrow AC$
 LDA C ; $AC \leftarrow M[C]$
 ADD D ; $AC \leftarrow AC + M[D]$
 MUL T ; $AC \leftarrow AC * M[T]$
 STA X ; $M[X] \leftarrow AC$

Programming the basic computer Part-1

General Register's organization

GPR ✓

($R_0, R_1, R_2 \dots R_{63}$) ✓

64 ✓

$$\log_2 64 = 6\text{-bits}$$

$$R_0 = 000\ 000$$

$$R_1 = 000\ 001$$

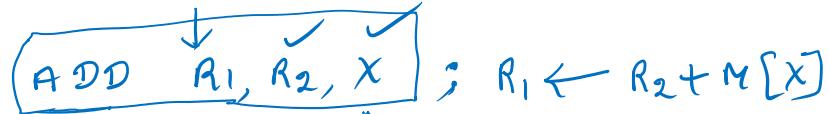
|

$$R_{63} = 111\ 111$$

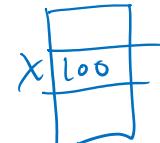
Three Address instructions ✓

Two-Address instructions . ✓

Opcode	Address 1	Address 2	Address 3
--------	-----------	-----------	-----------



ADD	R_1	R_2	#50
-----	-------	-------	-----



$$R_1 \leftarrow R_2 + 50$$

Programming the basic computer Part-1

Two-Address instructions

64

1M byte

6 bits

20 bits



ADD R₁, R₂ ; R₁ \leftarrow R₁ + R₂

SUB R₁, #50 ; R₁ \leftarrow R₁ - 50

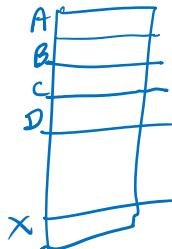


MUL R₁, Y ; R₁ \leftarrow R₁ * M[Y]



Evaluate the following Expression by using
Three Address and two Address instructions

$$X = (A + B) * (C + D)$$



Programming the basic computer Part-1

Three - Address instructions

\downarrow
 $x = (A + B) * (C + D)$
 \uparrow

```

ADD R1, A, B ; R1 ← M[A] + M[B]
ADD R2, C, D ; R2 ← M[C] + M[D]
MUL X, R1, R2 ; M[X] ← R1 * R2
  
```

Two - Address instructions

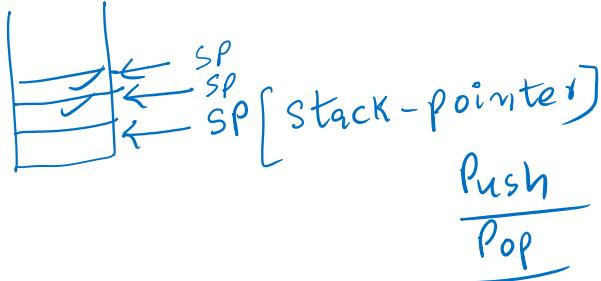
```

MOV R1, A ; R1 ← M[A]
ADD R1, B ; R1 ← R1 + M[B]
MOV R2, C ; R2 ← M[C]
ADD R2, D ; R2 ← R2 + M[D]
MUL R1, R2 ; R1 ← R1 * R2
MOV X, R1 ; M[X] ← R1
  
```

Programming the basic computer Part-1

zero - Address instructions

(Stack-organization)



APN { Reverse Polish Notation }

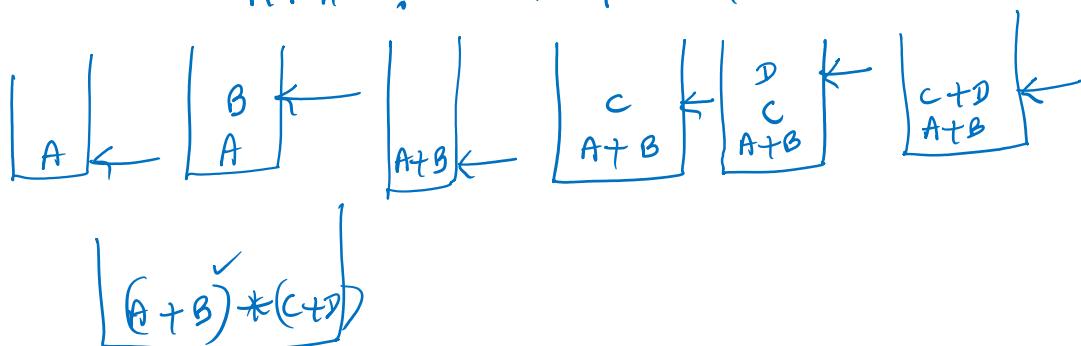
$$A+B, \quad \frac{AB+}{\uparrow}$$

RPN :

$$\downarrow x = \underline{(A+B)} * \underline{(C+D)}$$

$$\swarrow \nearrow \swarrow \nearrow \swarrow \nearrow \swarrow \nearrow$$

$$\overbrace{AB+} \quad \overbrace{CD+} \quad \overbrace{*}$$

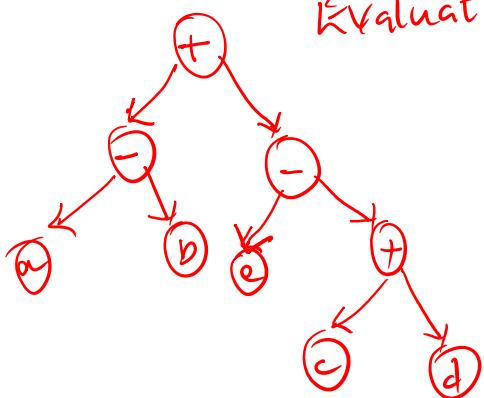


Programming the basic computer Part-1

Push A ; TOS $\leftarrow M[A]$
Push B ; TOS $\leftarrow M[B]$
ADD ; TOS $\leftarrow A + B$
Push C ; TOS $\leftarrow M[C]$
Push D ; TOS $\leftarrow M[D]$
ADD ; TOS $\leftarrow C + D$
MUL ; TOS $\leftarrow (A+B) * (C+D)$
Pop X ; M[X] $\leftarrow TOS$

$$\begin{array}{r} AB + CD + * \\ \hline \uparrow \uparrow \uparrow \uparrow \uparrow \end{array}$$

Programming the basic computer Part-1



- Evaluate → Two Address instructions
 → Perform Operation only when both Operands
 are in registers
 → No intermediate result should be stored in
 memory
 → what is Min. no. of Registers required
 to Evaluate the Expression ?

$$[(a - b) + (c - (c + d))]$$

$ab - cd + - +$

✓

Programming the basic computer Part-1

$$(a - b) + (e - (c + d))$$

a, b, c, d, e are initially available in memory.

Load R₁, a ; R₁ $\leftarrow M[a]$
Load R₂, b ; R₂ $\leftarrow M[b]$
Sub R₁, R₂ ; R₁ $\leftarrow R_1 - R_2$
↑ ↑
Load R₂, c ; R₂ $\leftarrow M[c]$
Load R₃, d ; R₃ $\leftarrow M[d]$
Add R₂, R₃ ; R₂ $\leftarrow R_2 + R_3$
Load R₃, e ; R₃ $\leftarrow M[e]$
Sub R₃, R₂ ; R₃ $\leftarrow R_3 - R_2$
↑
Add R₁, R₃ ; R₁ $\leftarrow R_1 + R_3$
↑

3 Registers Only

Programming the basic computer Part-1

Evaluate the Expression : $\frac{A + (B - C)}{D * E}$

$A \mid B$
 $AB \mid$
 $B \backslash A$

ONE-Address

LDA	D	;	AC $\leftarrow M[D]$
MUL	E	;	AC $\leftarrow AC * M[E]$
STA	T	;	$M[T] \leftarrow AC$
LDA	B	;	AC $\leftarrow M[B]$
SUB	C	;	AC $\leftarrow AC - M[C]$
ADD	A	;	AC $\leftarrow AC + M[A]$
DIV	F	;	AC $\leftarrow AC \mid M[F]$

Programming the basic computer Part-1

Two - Address instructions

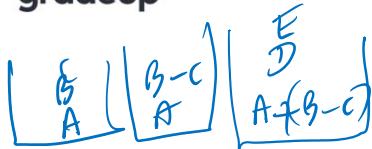
$$\left[\frac{A + (B - C)}{(D * E)} \right]$$

{
 MOV R1, B ; R1 $\leftarrow M[B]$
 SUB R1, C ; R1 $\leftarrow R1 - M[C]$
 ADD R1, A ; R1 $\leftarrow R1 + M[A]$
 MOV R2, D ; R2 $\leftarrow M[D]$
 MUL R2, E ; R2 $\leftarrow R2 * M[E]$
 DIV R1, R2 ; R1 $\leftarrow R1 / R2$

Three Address instructions.

{
 SUB R1, B, C ; R1 $\leftarrow M[B] - M[C]$
 ADD R2, R1, A ; R2 $\leftarrow R1 + M[A]$
 MUL R1, D, E ; R1 $\leftarrow M[D] * M[E]$
 DIV R3, R2, R1 ; R3 $\leftarrow R2 / R1$

Programming the basic computer Part-1



Zero - Address instructions

$$\frac{A + (B - C)}{(D * E)}$$

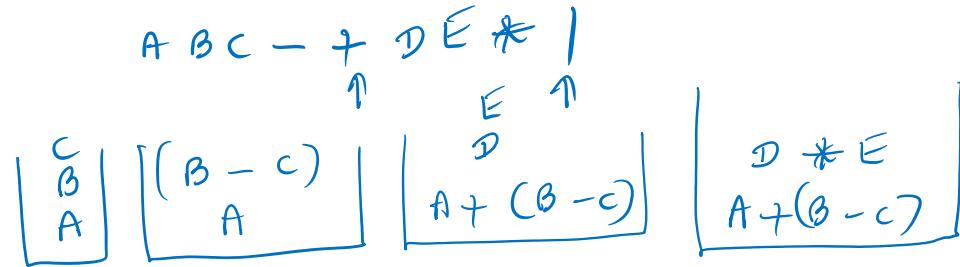
$A B C - + D E *$

$$\begin{array}{|c|} \hline (D * E) \\ \hline A + (B - C) \\ \hline \end{array}$$

- | | |
|----------|--|
| Push A ; | Tos $\leftarrow M[A]$ |
| Push B ; | Tos $\leftarrow M[B]$ |
| Push C ; | Tos $\leftarrow M[C]$ |
| SUB ; | Tos $\leftarrow (B - C)$ |
| ADD ; | Tos $\leftarrow A + (B - C)$ |
| Push D ; | Tos $\leftarrow M[D]$ |
| Push E ; | Tos $\leftarrow M[E]$ |
| MUL ; | Tos $\leftarrow (D * E)$ |
| DIV ; | Tos $\leftarrow (A + (B - C)) / (D * E)$ |

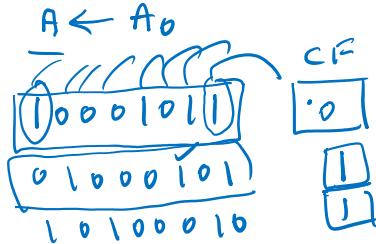
Programming the basic computer Part-1

$$\frac{A + (B - C)}{D * E}$$



Programming the basic computer Part-1

A, B, C are the 8 bit Registers.



MOV B, #0 ; $B \leftarrow 0$
MOV C, #8 ; $C \leftarrow 8$

Z : CMP C, #0 ; Compare C with 0.

JZ X ✓ ; Jump to X if zero flag is set

SUB C, #1 ; $C \leftarrow C - 1$

RRD A, #1 ; Right Rotate A through carry flag by 1-bit

JC Y ; Jump to Y if CF=1

JMP Z ; jump to Z

Y : ADD B, #1 ; $B \leftarrow B + 1$

JMP Z ; jump to Z

X : q. ✓

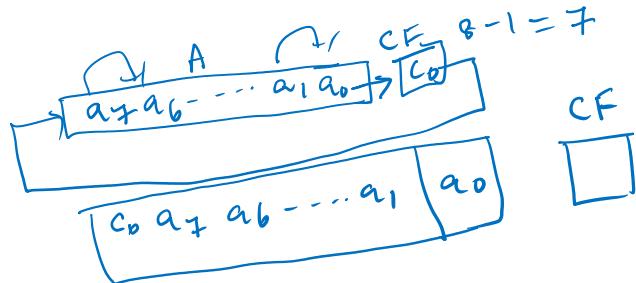
$$8-0 = 8$$

$$7-0 = 7$$

$$0-0 = 0$$

$$B = 1$$

(4)

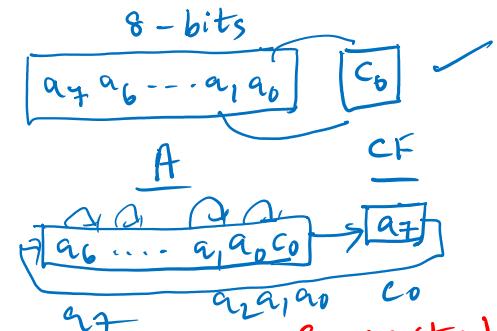


Programming the basic computer Part-1

Q1). If Initial Value of A is A₀, The Value of Register-B after Program execution will be ?

- (A) Number of 0 bits in A₀.
- (B) Number of 1 bits in A₀
- (C) A₀
- (D) 8

V	C	S	Z
BV, BNV, BP, BM, BZ, BNZ			

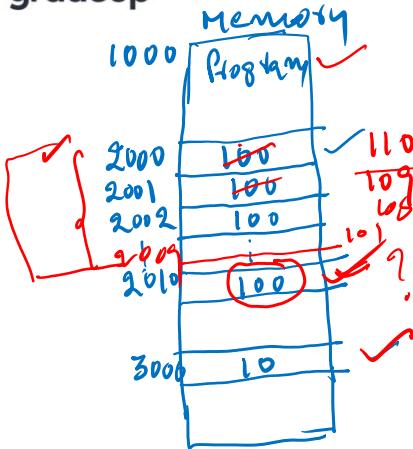


Q2). which of the following instruction when inserted at label X, will ensure that value of register-A after program execution is same as the initial value ?

- (A) RRC A, #1
- (B) Nop
- (C) LRC A, #1
- (D) ADD A, #1

X : RRC A, #1

Programming the basic computer Part-1



Consider the following Assembly Language Code. [R₁, R₂, R₃ GPA]

<u>MOV R₁ (3000)</u>	;	R ₁ \leftarrow M[3000]	2 words	I ₁
<u>LOOP: MOV R₂, (R₃)</u>	;	R ₂ \leftarrow M[R ₃]	1 "	I ₂
<u>ADD R₂, R₁</u>	;	R ₂ \leftarrow R ₁ + R ₂	1 "	I ₃
<u>MOV (R₃), R₂</u>	;	M[R ₃] \leftarrow R ₂	1 "	I ₄
<u>INC R₃</u>	;	R ₃ \leftarrow R ₃ + 1	1 "	I ₅
<u>DEC R₁</u>	;	R ₁ \leftarrow R ₁ - 1	1 "	I ₆
<u>BNZ LOOP</u>	;	Branch on Non-zero	2 "	I ₇
<u>HALT</u>	;	Stop	1 "	I ₈

Assume that $R_3 = \overline{2000}$

$$R_1 = 10$$

$$\text{at } R_3 = \frac{2000}{\boxed{2000}}$$

2001

$$R_2 = 100 + 9 = 109$$

$$R_1 = \textcircled{10} \rightarrow$$

$$R_2 \leftarrow M[\underline{2000}]$$

$$R_2 = 100 \text{ } \square$$

$$R_2 = 100 + 10 = \underline{110}$$

Programming the basic computer Part-1

$10 \times 2 + 1$

(21)

Q1). the no. of memory references for accessing data in executing the program Completely is ?

- (A) 10 (B) 11 (C) 20 ~~(D) 21~~

Q2). After executing this Program the content of mem. Location 2010 is ?

- ~~(A) 100 (B) 101 (C) 102 (D) 110.~~

2000	110
2001	109
2010	108

Q3). Memory is byte Addressable, words size is 32 bits. If interrupt occurs during execution of INC, R3, the Return Address pushed onto the stack will be ?

- (A) 1005 (B) 1020 ~~(C) 1024 (D) 1040~~

Programming the basic computer Part-1

Sol. of Q3)

2 words (I_1)

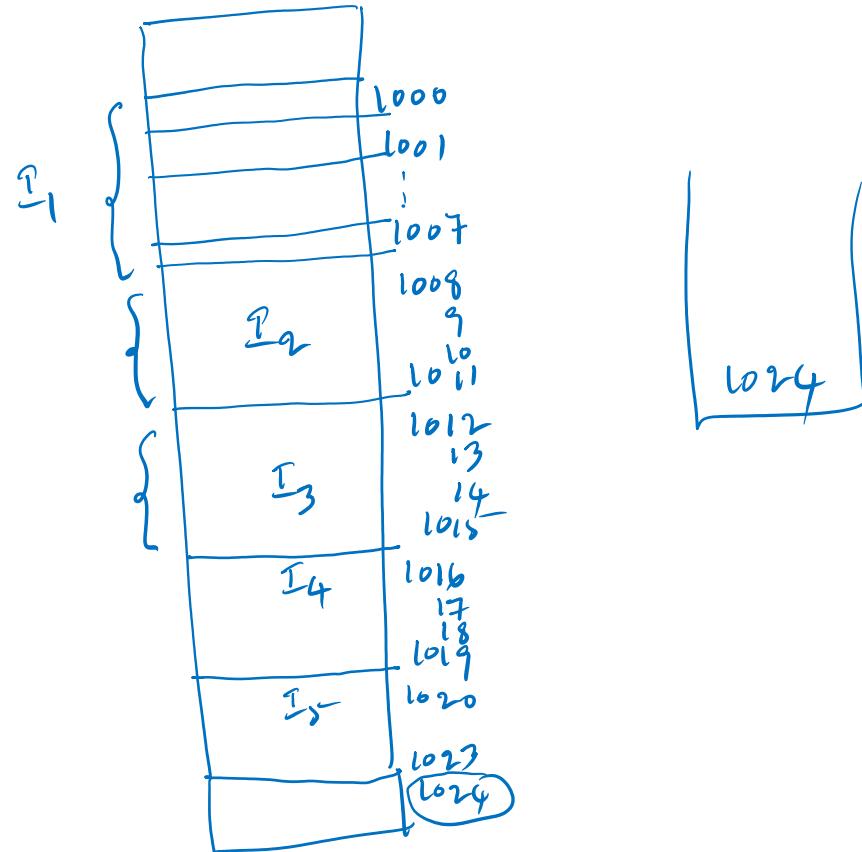
I_1
 I_2

I_1
 I_2

32 bits

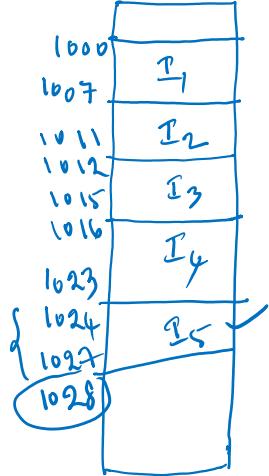
4 Bytes

2 words =
8 Bytes



Programming the basic computer Part-1

Consider the following Program.



SB I ₁ : MOV R ₁ , 5000 ✓ ; R ₁ ← M[5000]	2 words
4B I ₂ : MOV R ₂ , (R ₁) ✓ ; R ₂ ← M[R ₁]	1 "
4B I ₃ : ADD R ₂ , R ₃ ; R ₂ ← R ₂ + R ₃	1 "
8B I ₄ : MOV 6000, R ₂ ✓ ; M[6000] ← R ₂	2 "
4B I ₅ : HALT ✓ ; Machine halts	1 "

Assume mem. is byte addressable with word size 32 bits. Program has been loaded starting from mem. location 1000. If interrupt occurs while the CPU has been halted after executing the halt, the return address saved in the stack will be?

- (A) 1007 (B) 1020 (C) 1024 (D) 1028.

Programming the basic computer Part-1

Q2). Let the clock cycles required for various operations be as follows

Register to memory transfer : 3 cycles ✓

ADD with both operands in Registers : 1 " ✓

Instruction fetch/Decode : 2 cycles } per word

The total no. of cycles required for executing the program is

	(A) 29	(B) 24	(C) 23	(D) 20	Total
			<u>Fetch and Decode</u>	<u>Execute</u>	
I ₁		4 ✓		3	7
I ₂		2		3	5
I ₃		2		3	3
I ₄		4 ✓		—	7
I ₅		2		—	2
					24

Programming the basic computer Part-1

→ Consider the Code given below.

$$R_1 = 1000$$

$$R_2 = 5000$$

$$R_2 = 5000 + 500 = \underline{\underline{5500}}$$

Load $R_1, \#1000$ Load $R_2, \#5000$ ADD $R_2, \#500$ <u>STORE</u> $\underline{500(R_1)}, R_2$ <u>STORE</u> $\underline{1001}, R_2$ <u>HALT</u>	$; R_1 \leftarrow 1000$ $; R_2 \leftarrow 5000$ $; R_2 \leftarrow R_2 + 500$ $; M[R_1 + 500] \leftarrow R_2$ $; M[1001] \leftarrow R_2$ $; Stop.$
---	--

Find the Value of memory Location 1500, after the execution of above program.

$$500 + (R_1)$$

$$500 + 1000 = \underline{\underline{1500}}$$

1500

5500	1001
5500	1500