CS & IT ENGINEERING

Computer Organization & Architecture

1500 Series

Lecture No.- 02

By- Vijay Agarwal Sir

Recap of Previous Lecture

Topic	Floating Point Representation
Topic	Memory Concept
Topic	Little Endian & Big Endian(Byte Ordering)
Topic	Clock Cycle Concept.

Topics to be Covered

- Topic: Clock Cycle Concept.
- Topic: Machine Instruction
- Topic: Expand Opcode Techniue
- Topic: Addressing Modes

[NAT]

#Q. Initial values of R_1, R_2 and index registers are 30, 20 and 10 respectively. The memory locations 10, 20, 30 and 40 have data values 10, 11, 12 and 13 respectively. Consider the following instruction. Add R_1, (R_2). Value of R_1 after executing above instruction is _____.

$R_1 = 30$
$R_2 = 20$
Index Reg. = 10

10	10
20	11
30	12
40	13

Reg Direct Reg Indirect

$R_1 \leftarrow R_1 + (R_2)$

$R_1 \leftarrow R_1 + M[R_2]$

$R_1 \leftarrow 30 + M[20]$

$\leftarrow 30 + 11$

$R_1 = 41$ Ans

[MCQ]

#Q. A computer has 24 bit instructions and 8 bit addresses. Which of the following combinations of two addresses, one address and zero address instructions respectively cannot be implemented in this machine?

A 250, 1500, 9216

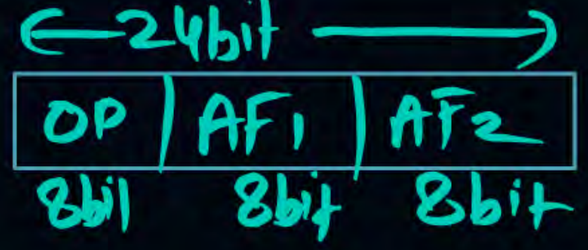
B 254, 256, 65000

C 250, 1530, 1540

D 255, 255, 256

1536

Ans [C]



250, 1500, 9216 ⇒ Implement

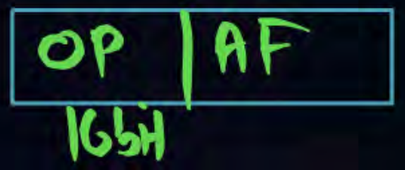
Total # operation 2AF = $2^8 = 256$.

Given = 250

Free = 256 − 250 = (6)

16 − 8 = 8

1AF ⇒ 6×2^8



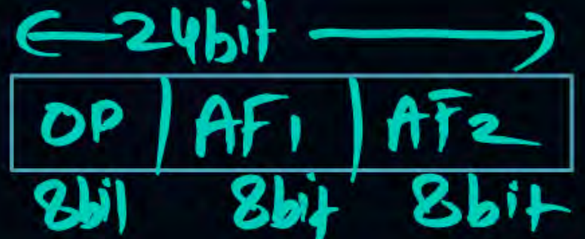
= 1536

Used = 1500

Free = (36)

0AF = 36×2^8

= 9216



(b) 254, 256 . 65000 ⇒ Implemented

Total # operation 2AF = 2^8 = 256.

Given = 254

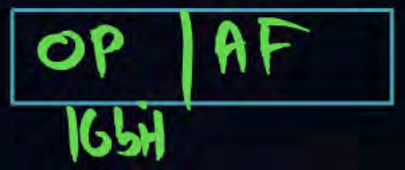
16-8 Free = 256 − 254 = (2) Free

1AF ⇒ 2×2^8

= (512)

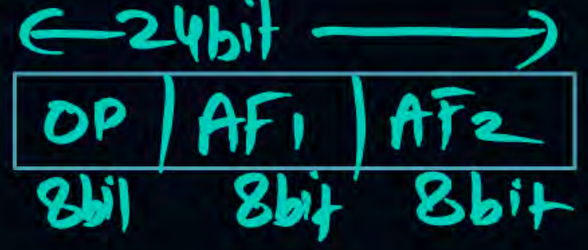
Used = 256

Free = 256



OAF = 256×2^8

= 65,536



© 250, 1530, 1540.

Total # operation 2AF = $2^8 = 256$.

Given = 250

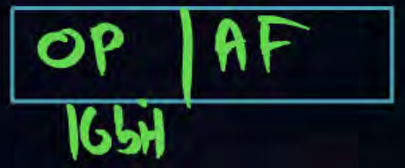
Free = 256 − 250 = (6) Free

Not Implemented

$16-8$

8

1AF ⇒ 6×2^8



= 1536

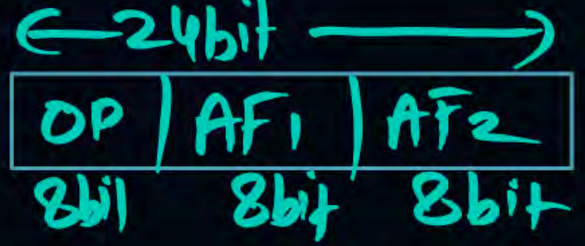
Used = 1530

Free = 1536 − 1530

= (6)

OAF = 6×2^8

= (1536) But OAF given (1540)



(d) 255, 255, 256. → YES Implemented

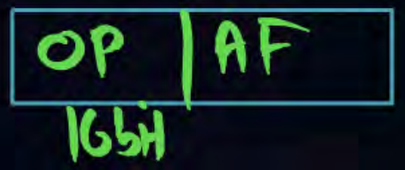
Total # operation 2AF = $2^8 = 256$.

Given = 255

Free = 256 − 255 = (1) Free

16−8 = 8

IAF ⇒ 1×2^8 = 256



Used = 255

256 − 255 = Free = (1)

OAF = 1×2^8 = (256) ✓

[MCQ]

#Q. In a certain processor, a 2 byte Jump instruction is encountered at memory address 3010 H, the Jump instruction is in PC relative mode. The instruction is JMP-7 where -7 is signed byte. Determine the Branch Target Address

A 3005 H

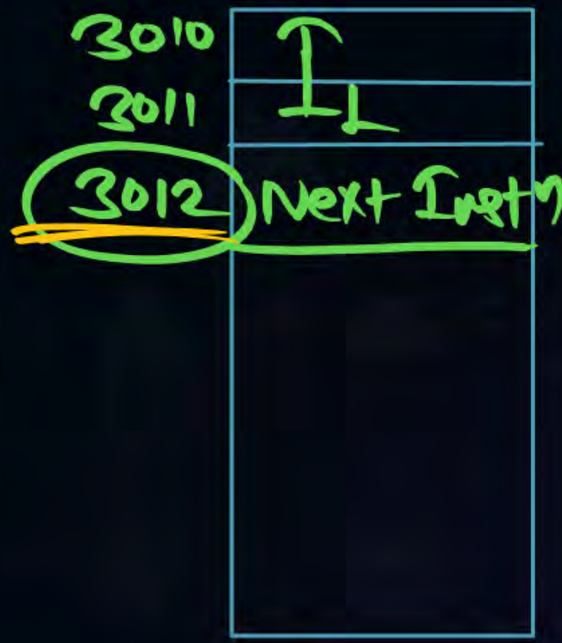
B 3009 H

C 3003 H

D 3007 H

Relative Value

JMP	-7



Target Address = Current PC Value + Relative Value (OFFSET) AF

⇒ 3012 − 7

= 3005 Ans

Ans (A)

[NAT]

#Q. Consider the following program segment for a CPU having 3 registers:

instruction	Operation	Size(in words)
MOV R_0, 2000	$R_0 \leftarrow M[2000]$	2
MOV R_1, $[R_0]$	$R_1 \leftarrow M[R_0]$	2
SUB R_1, R_2	$R_1 \leftarrow R_1 - R_2$	1
MOV 4000, R_1	$M[4000] \leftarrow R_1$	2
HALT	Stop	1

2x1 + 3 = 5
2x1 + 3 = 5
1x1 + 1 = 2
2x1 + 3 = 5
1x1 + = 1

18 Cycle

let the clock cycles required for operations are

Register to/from m/ r transfer = 3 CC

SUB with both registers operands = 1CC

Ans (18).

Instruction fetch and decode = 1 CC per word

Total number of CC required to execute the program is 18 Ans.

[NAT]

1 word = 4 Byte ⇒ 1 word

Consider the following program segment for a hypothetical CPU. Here r_0, r_1 and r_2 represents general purpose register.

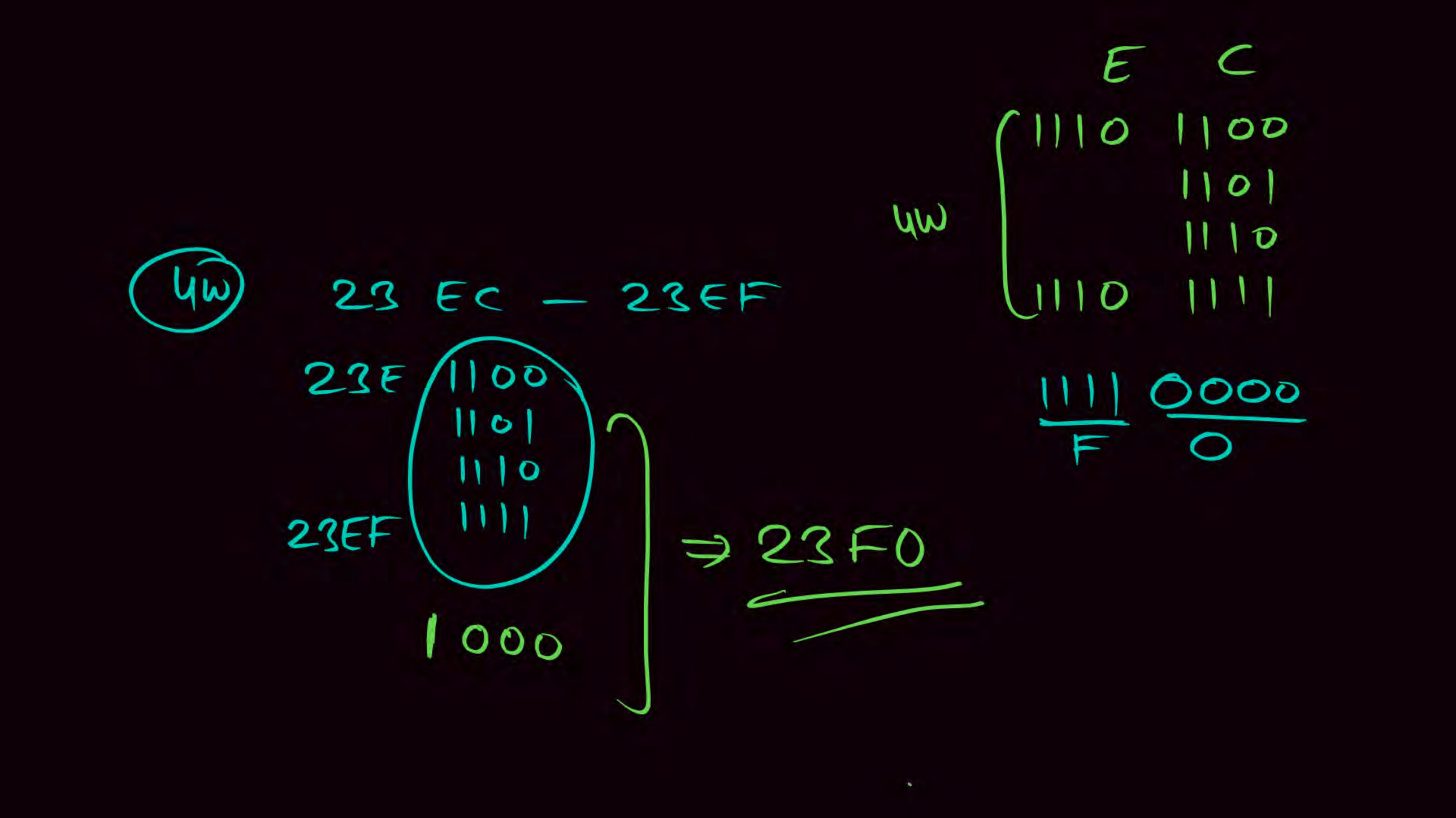
Instruction	Operation	Instruction size (in byte)		
Mov r_0, @ 7000	$r_0 \leftarrow M[(7000)]$	4W	16	23EC - 23EF
Mov r_1, @ r_2	$r_1 \leftarrow M[[r_2]]$	3W	12	23F0 - 23F2
ADD r_0, r_1	$r_0 \leftarrow r_0 + r_1$	2W	8	23F3 - 23F4
MUL r_0, r_1	$r_0 \leftarrow r_0 \times r_1$	2W	8	23F5 - 23F6
MOV @7000, r_0	$M[[7000]] \leftarrow r_0$	4W	16	23F7 - 23FA
Halt	Machine halt	2W	8	23FB - 23FC

I_1 I_2 I_3 I_4 I_5 I_6

23FD

Assume that memory is word addressable with word size is 32 bits & program has been loaded starting from memory location $(23EC)_{16}$ [Hexadecimal] onwards. If an interrupt occurs while the CPU has been halted after executing halt instruction. What will be the return address (in Hexadecimal) pushed onto the stack is:

Ans (23FB)


E C
4W
1110 1100
1101
1110
1110 1111
4W
23 EC — 23EF
23E 1100
1101
1110
23EF 1111
1000
⇒ 23F0
1111 0000
F 0

[NAT]

$\frac{1}{500} \times 10^{-6} \Rightarrow$ Cycle time = 2 nsec

A 500 MHz processor was used to execute the program with the following instruction & their clock cycle count.

Instruction type	Instruction Count	CPI
Integer Arithmetic	40000	1
Floating Point	33000	2
Data transfer	9000	1
Control transfer	11000	3
Shift operation	7000	2

$$\text{Avg CPI} = \frac{40000 \times 1 + 33000 \times 2 + 9000 \times 1 + 11000 \times 3 + 7000 \times 2}{1,00,000}$$

Avg CPI = 1.62 cycle

$\Rightarrow 1.62 \times 2$

$\Rightarrow$ 3.24 nsec

The MIPS (millions of Instructions per second) rate, for this program is .

(Note: Ignore the fractional part)

Ans(308)

1 Instⁿ ——— 3.24×10^{-9} sec.

In 1 sec ——— $\frac{1}{3.24} \times 10^{9}$ Instⁿ/sec

$\Rightarrow \frac{1000 \times 10^{6}}{3.24}$

$\Rightarrow$ 308 MIPS.

[NAT]

The memory location 100 to 110 have data value from 1 to 11 in sequence respectively. Initial value of registers R_1 and R_2 are 0 to 1000 respectively.

Consider the following two address instructions.

Add R_1, #1; $R_1 \leftarrow R_1 + 1$

Add R_2, 100(R_1); $R_2 \leftarrow R_2 + M[100 + R_1]$

The value of R_2 after executing above sequence of instructions seven times is 1035. Ans

I_1: Add R_1 #1 $R_1 \leftarrow R_1 + 1$

I_2: $R_2 \leftarrow R_2 + M[100 + R_1]$

$R_1 = 0$

$R_2 = 1000$

1st Iteration

I_1: $R_1 = 0 + 1 \Rightarrow (R_1 = 1)$

I_2: $R_2 \leftarrow 1000 + M[100 + 1]$

$1000 + M[101]$

$R_2 = 1000 + 2$

$(R_2 = 1002)$

2nd Iteration

I_1 $R_1 \leftarrow 1 + 1 \Rightarrow (R_1 = 2)$

I_2: $R_2 \leftarrow 1002 + M[100 + 2]$

$1002 + 3$

$R_2 = 1005$

111th Iteration

$R_1 = 2 + 1 \Rightarrow (R_1 = 3)$

$R_2 \leftarrow 1005 + M[100 + 3]$

$R_2 = 1005 + 4$

$(R_2 = 1009)$

① ② ③ ④ ⑤ ⑥ ⑦

$1002 + 3 + 4 + 5 + 6 + 7 + 8$

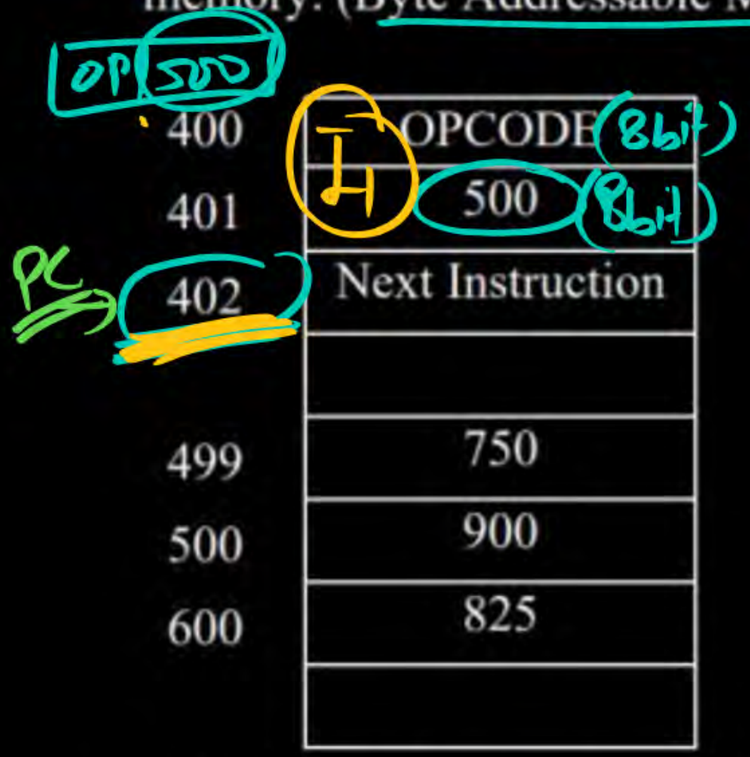
(1035) Ans

100	1
101	2
102	3
103	4
104	5
105	6
106	7
107	8
108	9
109	10
110	11

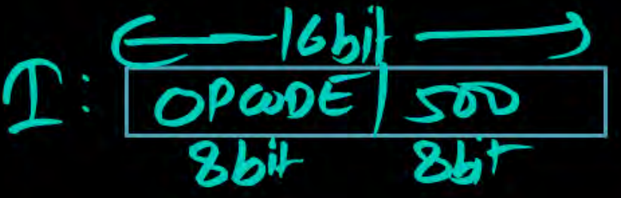
[NAT]

2 Byte

Consider a 16-bit Instruction in which 8bit opcode and one address instruction is loaded in main memory. (Byte Addressable Memory)



The effective address using PC-Relative addressing mode when processor is executing an instruction at location 400 is 902 Ans



PC Relative AM

TA(EA) = Current PC Value + AF (OFFSET) Relative Value.

$\Rightarrow 402 + \overset{AF}{500}$

$= 902$ Ans

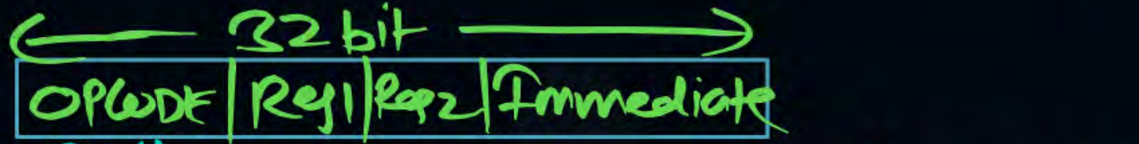
[NAT]

#Q. A machine has 32 bit Architecture with 1 word long instructions. It has 28 Register each of which is 32 bit long. Machine support 300 instruction, which have an immediate operand in addition to two register operand. Assuming that the immediate operand is an unsigned integer. Then the maximum value of the immediate operand is __8191__ Ans

Instⁿ Size = 32 bit

28 Reg ⇒ Reg AF = 5 bit

300 operation ⇒ Opcode = 9 bit



= 32 − (9 + 5 + 5)

Immediate field = 13 bit

Unsigned Range = $2^{13} - 1$

⇒ 8192 − 1 = 8191 Ans

Ans (8191)

[NAT]

#Q. Consider the Hypothetical CPU using PC-relative addressing mode instruction of RISC instruction set architecture. Instruction is stored in the memory location with starting address of 2000 decimal onwords.

2000	I_1:	ADD	R_0	R_1	R_2
2004	I_2:	BEQ	R_3	R_4	Lable (OFFSET)
2008	I_3:	MUL	R_5	R_5	R_6
2012	I_4:	ADD	R_7	R_5	R_2
2016	I_5:	MUL	R_8	R_1	R_6

Ans (2040)

if $R_3 == R_4$ OFFSET: 32

(BEQ: Brach if equal)

Here label is used as an offset. R_0, R_1, R_2, R_3 .... R_8 are general purpose registers. The BEQ instruction branches the PC if the register R_3 and register R_4 content are equal then label =32. The value of register $R_0 = 10$, $R_1 = 20$, $R_2 = 30$, $R_3 = 40$, $R_4 = 40$, $R_5 = 50$, $R_6 = 60$ respectively. What is the memory address (in decimal) of the next instruction to be executed ____. 2040 Ans

Label = OFFSET + 32

PC Relative AM.

EA(TA) = Current PC Value + AF(OFFSET) Relative Value

When I_2 Fetch then PC value = 2008 (I_3 starting address)

& $R_3 == R_4$ (40 = 40) So PC Relative OFFSET = 32

2008 + 32

= 2040 Ans

[MSQ]

The most appropriate matching for the following pairs is/are possible

List-I		List-II	
A.	Loops	1.	Auto Increment
B.	A[I] = B[J];	2.	Auto decrement
C.	Array implementation	3.	Base register addressing
D.	While [*A++];	4.	Indirect addressing
E.	Writing re-located code	5.	Indexed addressing

Ans (A) & (B)

(A)

A	B	C	D	E
2	5	5	1	3

(B)

A	B	C	D	E
1	5	5	1	3

(C)

A	B	C	D	E
4	3	2	5	1

(D)

A	B	C	D	E
1	5	3	2	4

[NAT]

The initial value of register R_1 and R_2 are 2000 and 4000 respectively. The memory location 2000 and 4000 have data values 10 and 20 respectively. Consider the following instructions.

I_1 Add R_2, R_2, (2000) R_1; displacement addressing

I_2 Add R_2, R_2, (R_1) ; Register Indirect

The value of R_2 after executing above instructions is 4030 Ans

$R_1 = 2000$, $R_2 = 4000$

I_1: $R_2 \in R_2 + M[2000 + R_1]$ $\Rightarrow$ $R_2 \in 4000 + M[2000 + 2000]$

$4000 + M(4000) \Rightarrow 4000 + 20$

$R_2 = 4020$

I_2: $R_2 \in R_2 + M[R_1]$

$= 4020 + M[2000] \Rightarrow 4020 + 10$

$R_2 = 4030$ Ans

[MSQ]

Consider the following statements about the various addressing modes (AM)

S_1: Immediate AM are used to access the variable → Constant

S_2: In the Indirect AM address field of the instruction contain the address where the operand is stored.

S_3: Index AM are used to implement the Array.

S_4: Absolute AM are used to implement the pointer.

Which of above statements is/are incorrect?

Immediate AM – Constant
Direct / Absolute – Variable
Indirect AM – Pointer
Index AM – Array.

A S_1

Ans (A) (B) & (D)

B S_2

C S_3

D S_4

Home Work

[MCQ]

#Q. Consider the following assembly level program for a hypothetical processor.
R_1, R_2 and R_3
are 32-bit registers.

```
MOV R1, #0          ; R1 = 0
MOV R2, #1          ; R2 = 1
CMP R3,#0           ; Compare R3 with 0
BEQ DONE            ; Branch to DONE if zero flag is set
X:
ADD R2, R1, R2      ; R2 ← R1 + R2
SUB R1, R2, R1      ; R1 ← R2 – R1
SUB R3, R3, #1      ; R3 ← R3 – 1
BNE X               ; Jump to X if zero flag is not set.
DONE:
```

If the initial value of R_3 is 10, what will be the value of R_2 (in decimal)?

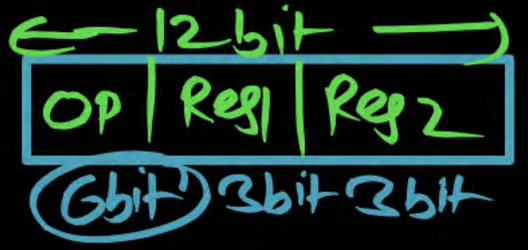
A 55

B 89

C 144

D None of these

[MCQ]



Consider a design will expand opcode technique with 12bit instructions, where a register operand requires 3 bits. There are 24 – 2 address instruction consisting of two register operands, and 5-one address instructions consisting of one memory operand of 8 bits. Then find the number of 0-address instruction in the system possible.

(a) 512

(b) 1024

(c) 2048

(d) ~~Not possible with the given instruction size.~~
None of these.

2 mins Summary

- Topic: Clock Cycle Concept.
- Topic: Machine Instruction
- Topic: Expand Opcode Techniue
- Topic: Addressing Modes

THANK - YOU