

# CS & IT ENGINEERING

## Computer Organization & Architecture

Lecture No.- 01

By- Vijay Agarwal Sir



# Recap of Previous Lecture





# Topics to be Covered



Topic

Floating Point Representation

Topic

Memory Concept

Topic

Little Endian & Big Endian(Byte Ordering)

Topic

Clock Cycle Concept.

#Q. 44FC6000H represents a floating point number in IEEE-754 single precision format. The value in decimal form is

$$bias = 127$$

A	2017	$s \quad E(8\text{bit}) \quad m(23\text{bit})$	$E = 10001001 = 137$			
B	2018	<table border="1"> <tr> <td>0</td> <td>100 01001</td> <td>1111100 01100000 0000 0000</td> </tr> </table>	0	100 01001	1111100 01100000 0000 0000	
0	100 01001	1111100 01100000 0000 0000				
C	2019		$(-1)^s \cdot 1.m \times 2^e \Rightarrow (-1)^s \cdot 1.M \times 2^{E - bias}$			
D	2020		$\Rightarrow (-1)^0 \cdot 1.11100011000000000 \times 2^{137 - 127}$			
			$+ 1.11100011000000000 \times 2^{+10}$			
			$+ 1.11100011000000000$			
			$+ (2019)_{\text{Ans}}$			

Ans [C].

#Q. A number -1/8 is represented in IEEE 754 format as:

- |      |                            |            |
|------|----------------------------|------------|
| i)   | <u>1</u>   <u>01111100</u> | 00....0    |
| ii)  | 1   01111100               | 10.....0   |
| iii) | 1   01111111               | 0010.....0 |
| iv)  | 1   01111100               | 011....1   |

Which of the following is TRUE?

A

i and ii

Ans (B).

C

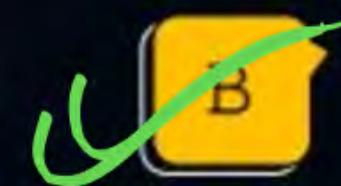
i, ii, iii



only i

D

i, iii, iv



$$-\frac{1}{8} \Rightarrow -0.\underbrace{001}_\text{3} \times 2^0 \\ \Rightarrow -1.0 \times 2^{-3}$$

bias = 127

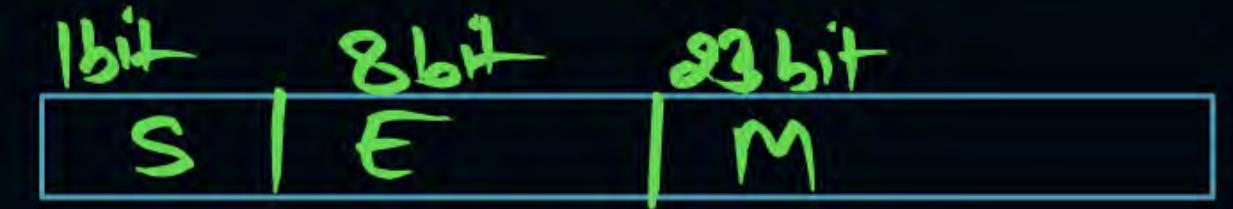
e = -3

$$E = e + bias \\ = -3 + 127$$

$E = 124$
-----------

$$S = -ve \\ E = 124$$

$$E = 01111100$$



1	01111100	0000000000000000000
---	----------	---------------------

#Q. **0x3F800000** represents a floating-point number in IEEE-754 single precision format. The value in decimal format is \_\_\_\_.

**3F8 00000**

**Ans(1)**

		$E = 8 \text{ bit}$						
S		E = 127, bias = 127, M = 00000000000000000						
0	0111111	00000000000000000000000000000000						

$$\text{Sign} = 0 \quad E = 127, \text{bias} = 127, \quad M = 00000000000000000$$

$$(-1)^S \times M \times 2^{E - \text{bias}}$$

$$(-1)^0 \times 00000000000000000 \times 2^{127 - 127}$$

$$+ 1.000000000 \times 2^0$$

$$\Rightarrow +1.000000000$$

#Q. Which one of the following represents a denormal number in IEEE 754 single precision format?

A 0xff800000

C 0x00000001

B 0x7f7fffff

D 0x80000000

①   
S=L E=255 , M=0  $\Rightarrow -\infty$

②   
S=0 E=254 M: Any thing : Implicit Normalized.

#Q. Which one of the following represents a denormal number in IEEE 754 single precision format?

A 0xff800000

C 0x00000001

E   
 $S=0$     $E=0$     $M \neq 0$ .   De Normalized Number.

d   
 $S=1$     $E=0$     $M=0$     $\Rightarrow -0$ .



# Topic : IEEE 754 Floating Point Representation



## IEEE 754 Floating Point Representation

Single Precision  
(32 bit)  
Excess 127

S	E	M
1bit	8bit	23bit

32 bit

Double Precision  
(64 bit)  
Excess 1023

S	E	M
1bit	11bit	52bit

64 bit

$$\begin{aligned}\text{bias} &= 2^{K-1} - 1 \\ &= 2^8 - 1\end{aligned}$$

$$\text{Bias} = 127$$

$$\begin{aligned}\text{bias} &= 2^{11-1} - 1 \\ \text{Bias} &= 1023\end{aligned}$$

# Single Precision (32 bit)



S	E	M
1 bit	8 bit	23 bit

Sign(1 bit)	E(1 bit)	M(23 bit)	Value
0 or 1	00000000 <u>E = 0</u>	0000000000000000 0000000 M = 0	$\pm 0$
<u>0 or 1</u>	11111111 <u>E = 255</u>	0000000000000000 0000000 M = 0	$\pm \infty$
0 or 1	$1 \leq E \leq 254$	M = .....	Implicit Normalized form $(-1)^S \times 1.M \times 2^E$ $(-1)^S \times 1.M \times 2^{E-127 \text{ bias}}$
0 or 1	<u>E = 0</u>	<u>M <math>\neq 0</math></u>	Denormalized number/Fractional form $(-1)^S \times 0.M \times 2^{E-127 \text{ bias}}$
0 or 1	<u>E = 255</u>	M $\neq 0$	Not a Number (NAN)



## Topic : Double Precision

1-bit    11-bit    52-bit

S	E	M
---	---	---

Excess - 1023

Sign (1 bit)	E(11 bit)	M(52 bit)	Value
0 or 1	0000 0000 000 E = 0	000000000000.. M = 0	$\pm 0$
0 or 1	1111 1111 111 E = 2047	0000000000.. M = 0	$\pm \infty$
0 or 1	$1 \leq E \leq 2046$	M = -----	Implicit Normalization $(-1)^S \cdot M \times 2^E$ $(-1)^S \times 1 \cdot M \times 2^{E-1023}$
0 or 1	E = 0	M $\neq 0$	Denormalized number/ Fractional Form $(-1)^S 0 \cdot M \times 2^{E-1023}$
	E = 2047	M $\neq 0$	Not a number

**NOTE:** When  $E = 0$  then Value 0      or      fractional form

$$\begin{cases} \text{when} \\ M = 0 \end{cases} \quad \begin{cases} \text{when} \\ M \neq 0 \end{cases}$$

**NOTE:** When  $E = 2047$   
then Value  $\infty$       or      Not a Number(NAN)

$$\begin{cases} \text{when} \\ M = 0 \end{cases} \quad \begin{cases} \text{when} \\ M \neq 0 \end{cases}$$

#Q. Which of the given number has its IEEE-754 32-bit floating point representation as

(010000000|110 0000 0000 0000 0000 0000)

A

2.5

$E=128$

B

3.0

$\text{Arg}(C)$

C

3.5

D

4.5

$$(-1)^S \cdot 1.M \times 2^{E-biag}$$

$$(-1)^0 \cdot 1.1100000000 \times 2^{128-127}$$

$$+ 1.110000 \times 2^{+1}$$

$$\Rightarrow 11.100 \Rightarrow +3.5 \text{ Arg}$$

#Q. The range of representable normalized numbers in the floating point binary fractional representation in a 32-bit word with 1-bit sign, 8-bit excess 128 biased exponent and 23-bit mantissa is

A

$2^{-128}$  to  $(2 - 2^{-23}) \times 2^{127}$

B

$(2 - 2^{-23}) \times 2^{-127}$  to  $2^{128}$

C

$(2 - 2^{-23}) \times 2^{-127}$  to  $2^{23}$

D

$2^{-129}$  to  $(2 - 2^{-23}) \times 2^{127}$

0 | 0000 0000 | 0000 0000 0000 0000 0000 0000 0000

0 | 1111 1111 | 1111 1111 1111 1111 1111 1111 1111 . 1

Ans(A).

$m(23bit)$ 

0	000000000	00000000000000000000
---	-----------	----------------------

 $biag = 128$  $E - biag$ 

$$(-1)^S \ 1 \cdot M \times 2$$

 $0 - 128$ 

$$(-1)^P \ 1 \cdot 000000000000 \times 2$$

$$\Rightarrow +1 \cdot 0000000 \times 2^{-128}$$

$$\Rightarrow +1 \cdot 0 \times 2^{-128}$$

$$\Rightarrow \boxed{2^{-128}}$$



$$S=0 \quad E=255, \text{ bias}=128$$

$$(-1)^S \cdot M \times 2^{E-\text{bias}}$$

$$\Rightarrow (-1)^0 \cdot 1 \cdot 111111111111111111111111 \times 2^{255-128}$$

$$\Rightarrow + \underbrace{\overleftarrow{1} \cdot \overrightarrow{111111111111111111111111} \times 2^{+127}$$

23 times

Right Shift 1 time

$$\text{O} \cdot \underbrace{11111111}_{24 \text{ times}} \times 2 \times 2^{127}$$

$$\left[ 1 - 2^{-24} \right] \times 2 \times 2^{127}$$

$$\left[ 2 - 2^{-23} \right] \times 2^{127} \text{ Ans}$$

Left Shift 23 time

$$\Rightarrow \left( 2 - 2^{-23} \right) \times 2^{127} \left( 2^{24} - 1 \right) \times 2^{-23} \times 2^{127}$$

⋮

$$0111 = 1 - \frac{1}{2^3}$$
$$\left[ -\frac{1}{2} \right]$$

#Q. Consider IEEE 754 single precision floating point format

Sign	Exponent	Fraction
	8bits	23 bits

31            30            23 22            0

What is the maximum positive normal value represented by this format?

A

$$2^{127}$$

Ans (D)

B

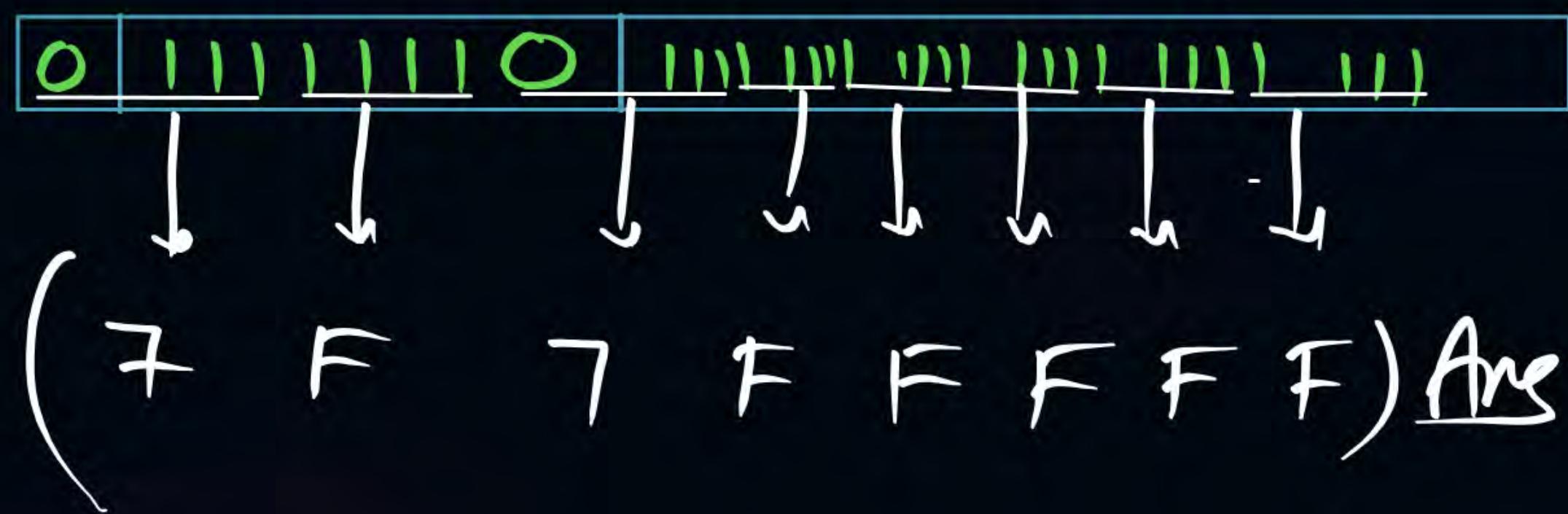
$$2^{128}$$

C

$$(1 - 2^{24}) \cdot 2^{127}$$

D

$$(1 - 2^{24}) \cdot 2^{128}$$



0	11111110	111111111111111111
---	----------	--------------------

$$(-1)^S \cdot M \times 2^{E-\text{bias}}$$

$$E = 254$$

$$\begin{aligned} &\text{bias} = 127 \\ &254 - 127 \end{aligned}$$

$$(-1)^0 \cdot 1 \cdot \underbrace{111111111111111111}_{23 \text{ times}} \times 2$$

$$+ L \cdot \underbrace{1111111}_{23 \text{ Times}} \times 2^{127}$$

Right Shift 1 bit

$$0 \cdot \underbrace{111111111111111111}_{24 \text{ Times}} \times 2^{\cancel{127}} \times 2^{128}$$

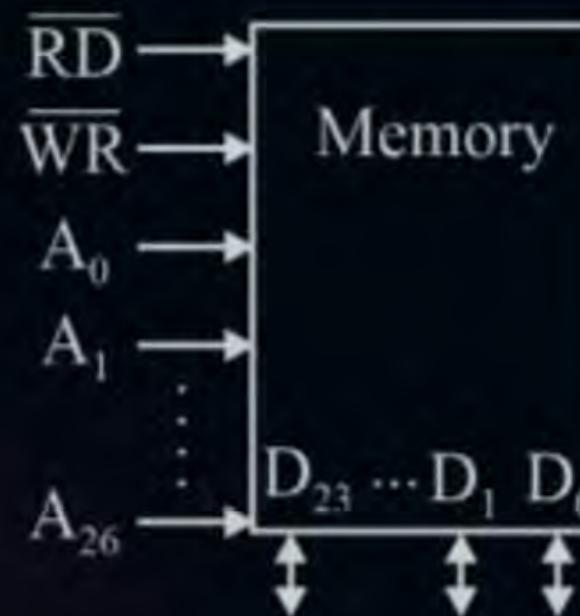
$$(1 - 2^{-24}) \times 2^{128}$$

left shift 23 bits.

$$\underbrace{111111111111111111}_{24 \text{ Times}} \times 2^{-23} \times 2^{127}$$

$$\begin{aligned} &\left[ 2^{24} - 1 \right] \times 2^{-23} \times 2^{127} \\ &\left( 1 - 2^{-23} \right) \times 2^{127} \Rightarrow \left( 1 - 2^{-24} \right) \times 2^{128} \end{aligned}$$

Consider the following memory design:



In above design  $A_0$  to  $A_{26}$  are address pins and  $D_0$  to  $D_{23}$  are data pins. The memory size in terms of byte is 384 MB.

$$\text{Address} = 27$$

$$\text{Data} = 24 \text{ bits}$$

$$2^{\underline{27}} \times 24 \text{ bits} \Rightarrow \frac{2^{\underline{27}} \times 24}{8} \text{ Byte}$$

$$\Rightarrow 2^{\underline{27}} \times 3 \text{ Byte}$$

$$\Rightarrow 128 \text{ M} \times 3 \text{ Byte}$$

$$\Rightarrow 384 \text{ MByte}$$

Ans (384)

## [MCQ]



The memory size for P address lines and q data lines is given as:

A  $2^P \times q$

B  $2^q \times p$

C  $2^{(P+q)}$

D  $2^{(P-q)}$

$$2^P \times q$$

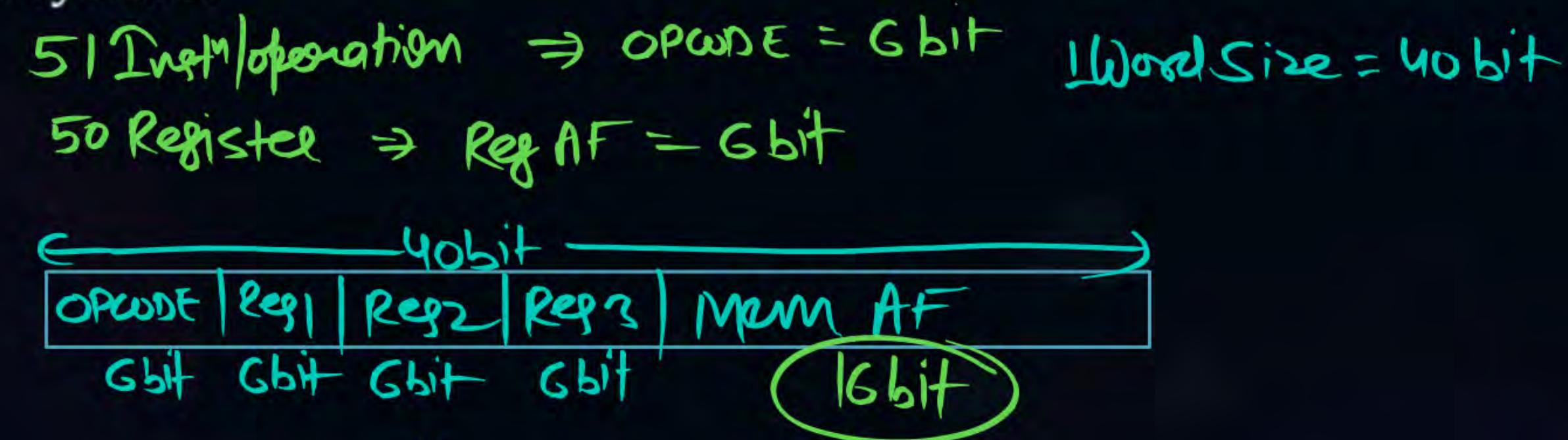
Ans (A)

# [MCQ]

A processor has 51 distinct instructions and 50 general purpose register. System supports word addressable memory. Word size is 40 bits. One word instruction has an opcode, 3 register operands and memory operand. What is the size of main memory possible in the system?

- A 64 kB
- B 320 kW
- C 320 kB
- D None of these

Ans(C)



$$\begin{aligned}
 \text{Memory} &= 2^{16} \text{ Word} \\
 &\Rightarrow 2^{16} \times 40 \text{ bits} \Rightarrow 2^{16} \times 5 \text{ Byte} \rightarrow 64K \times 5 \text{ Byte} \\
 &\Rightarrow 320K \text{ Byte}
 \end{aligned}$$

## [MCQ]



The capacity of a memory unit is defined by the number of word multiplied by the number of bits/word. How many separate address and data lines are needed for a memory of  $32K \times 16$ ?

A 8 address, 8 data line

C 15 address, 16 data line

B 14 address, 8 data line

D 16 address, 16 data line

$32K \times 16$

Ans (C).

$2^5 \times 2^{10} \times 16$

$2^{15} \times 16$

15 bit Address

16 bit Dataline

Consider a computer system having 2 category of code module x & y with the following details.

Instruction type	CPI for this Instruction type
Type -1	1
Type -2	2
Type -3	3
Type -4	4

For the particular programming language statements, compiler writer is considering 2 category of code module x & y with the following instruction count.

Code Module	Instructions Count. (IC)				Instruction type
	Type-1	Type-2	Type-3	Type-4	
X	1	2	2	1	
Y	4	2	1	1	

$$\text{Avg}(4 \cdot 375)$$

The CPI for code module X is A & CPI for code module Y is B then the value of A + B is \_\_\_\_\_.

$$\text{CODE MODULE } X = 1 + 2 + 2 + 1 = 6 \text{ Inst^n Total}$$

$$\text{CODE MODULE } Y = 4 + 2 + 1 + 1 = 8 \text{ Inst^n}$$

$$\text{CPU Clock Cycle} = \sum DC_i \times CPI_i$$

$$\text{MODULE } X = 1 \times 1 + 2 \times 2 + 2 \times 3 + 1 \times 4 = 1 + 4 + 6 + 4 = 15 \text{ Cycle}^{\text{Total}}$$

$$\text{MODULE } Y = 4 \times 1 + 2 \times 2 + 1 \times 3 + 1 \times 4 = 4 + 4 + 3 + 4 = 15 \text{ Cycle}$$

$$\text{Avg CPI}_X = \frac{15}{6} = 2.5$$

$$\text{Avg CPI}_Y = \frac{15}{8} = 1.875$$

$$A + B = 2.5 + 1.875 = \boxed{4.375} \text{ Avg}$$

Machine Instruction  
& ADDRESSING MODE.

**[MCQ]**

Consider the following signed data and perform the addition operation

11010101

11010010

What is the status of carry, overflow, zero and sign flags after processing respectively?

A 1101

B 1011

C 1001

D 1100

$$\begin{array}{r}
 \text{Carry} \quad 0 \\
 + 11010101 \\
 + 11010010 \\
 \hline
 10100111
 \end{array}$$

Carry = 1

Overflow =  $1 \oplus 1 = 0$

Zero = 0

Sign = 1

**Ans (C).**

out of  
MSB  $\oplus$  MSB

0 1 ]  $\Rightarrow$  overflow = Set = 1

0 0 } Not overflow = Reset [0]  
| 1 }

**[MCQ]**

The unsigned integer can be written in 32bit- binary as

11110100 10011000 10110111 00001111

**Ans(D)**

Putting it into four byte of memory beginning at address 100100 in big endian byte ordering scheme given in which picture? *Little*

A

100100	100101	100102	100103
00001111	10110111	10011000	11110100

B

100100	100101	100102	100103
10110111	00001111	11110100	10011000

C

100100	100101	100102	100103
10011000	11110100	00001111	10110111

D

100100	100101	100102	100103
11110100	10011000	10110111	00001111

*Little*

100103	100102	100101	100100
11110100	10011000	10110111	00001111

Big Endian

100103	100102	100101	100100
00001111	10110111	10011000	11110100

If each address space represents one byte of storage space. The number of address lines needed to access the RAM CHIPS arranged in  $4 \times 8$  array. Where size of each RAM CHIP  $16K \times 4$  bits is \_\_\_\_\_.

$$16K \times 4 \text{ bit} \Rightarrow 2^{14} \times 2^2 \text{ bits} = 2^K \text{ bits}$$

Ans(18)

Byte  $\Rightarrow \frac{2^{16}}{8} \text{ Byte} \Rightarrow 2^{13} \text{ Byte}$ .

To Represent  $2^{13}$  Byte Memory

RAM AF = 13 bits

Total RAM CHIP =  $4 \times 8 = 32$  RAM CHIP  $\Rightarrow$  5 bit Required to Represent these RAM Chip.

$$\begin{aligned} \text{Total} &= 13 + 5 \\ &= 18 \text{ bits Ans} \end{aligned}$$

[MCQ]



Consider a 32 bit register which stores floating numbers in IEEE single precision format. What is the value of the number, if 32 bit are given below?

Sign(1bit)	Exponent (8bit)	Mantissa (23 bit)
------------	-----------------	-------------------

0	10000011	1101 0000 0000 0000 0000 000
---	----------	------------------------------

$$E=13$$

$$13 - 127$$

- A 48  $(-1)^0 \cdot 1.11010000000 \times 2^{13}$
- B 50
- C 56  $+1 \cdot 11010000000 \times 2^{+4}$
- D None of these

$$1.1101 \cdot 000000$$

**+29 Ans**

**Ans(D)**



## 2 mins Summary



Topic

Floating Point Representation

Topic

Memory Concept

Topic

Little Endian & Big Endian(Byte Ordering)

Topic

Clock Cycle Concept.



THANK - YOU

# 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 Technique

Topic

Addressing Modes

#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

#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?

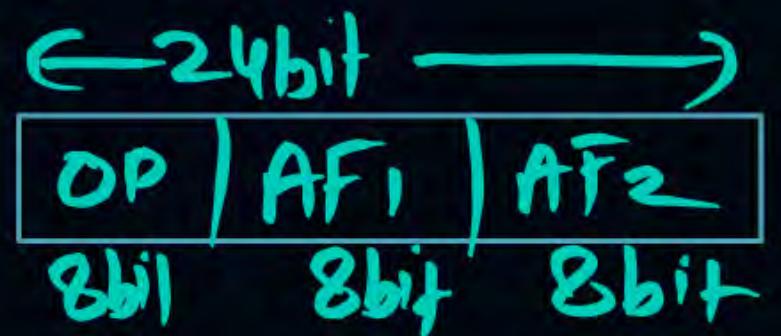
A 250, 1500, 9216

C 250, 1530, 1540

B 254, 256, 65000

D 255, 255, 256

Ans (C)



↙ 250, 1500, 9216 ⇒ Implement

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

Given = 250

$$\text{Free} = 256 - 250 = \textcircled{6}$$

1AF ⇒  $6 \times 2^8$

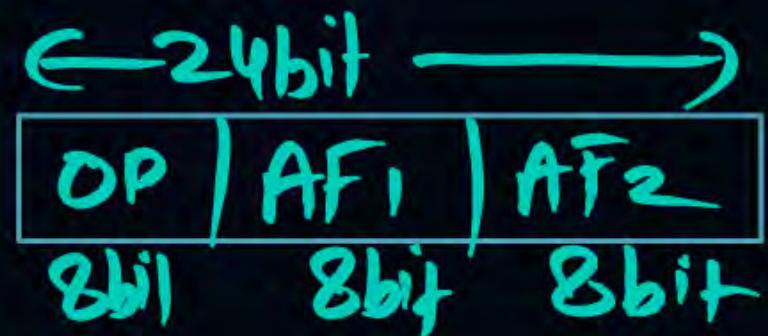
OP	AF
16bit	

$$= 1536$$

$$\text{Used} = 1500$$

$$\text{Free} = \textcircled{36}$$

$$\begin{aligned} \text{OAF} &= 36 \times 2^8 \\ &= 9216 \end{aligned}$$



⑥ 254, 256 → Implemented

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

Given = 254

$$\text{Free} = 256 - 254 = \textcircled{2} \text{ Free}$$

1AF  $\Rightarrow 2 \times 2^8$

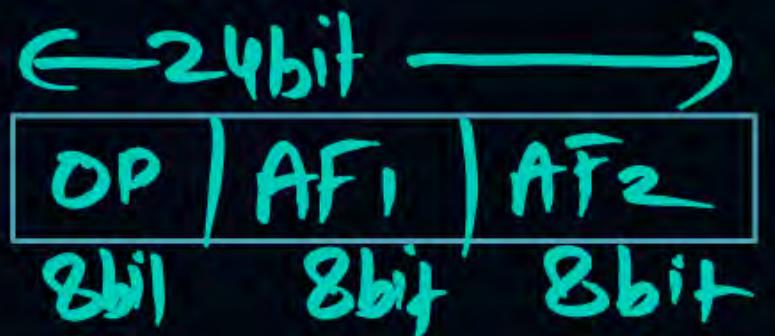
OP	AF
16bit	

= 52

$$\text{Used} = 256$$

$$\text{Free} = 256$$

$$\begin{aligned} \text{OAF} &= 256 \times 2^8 \\ &= \underline{\underline{65,536}} \end{aligned}$$



① 250, 1530, 1540.

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

Given = 250

$$\text{Free} = 256 - 250 = \textcircled{6} \text{ Free}$$

1AF  $\Rightarrow 6 \times 2$

OP	AF
16bit	

= 1536

Used = 1530

Free = 1536 - 1530

= 6

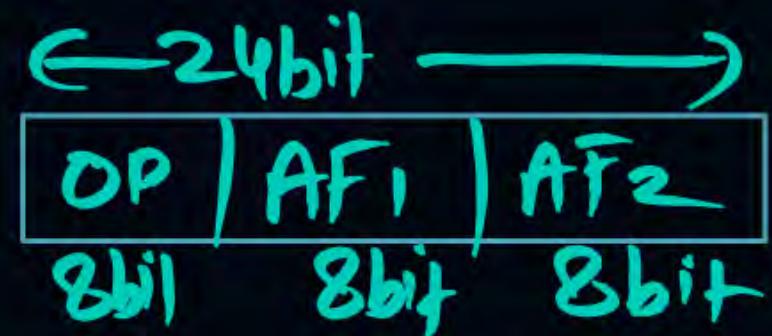
$$\text{OAF} = 6 \times 2^8$$

= 1536

But OAF given 1540

Not Implemented

1540



d) 255, 255, 256.

YES Implemented

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

Given = 255

$^{16-8}_{\text{Free}} = 256 - 255 = \textcircled{1} \text{ Free}$

$$\underline{\text{IAF}} \Rightarrow 1 \times 2^8 = 256$$

OP	AF
16bit	

Used = 255

$$256 - 255 = \textcircled{1}$$

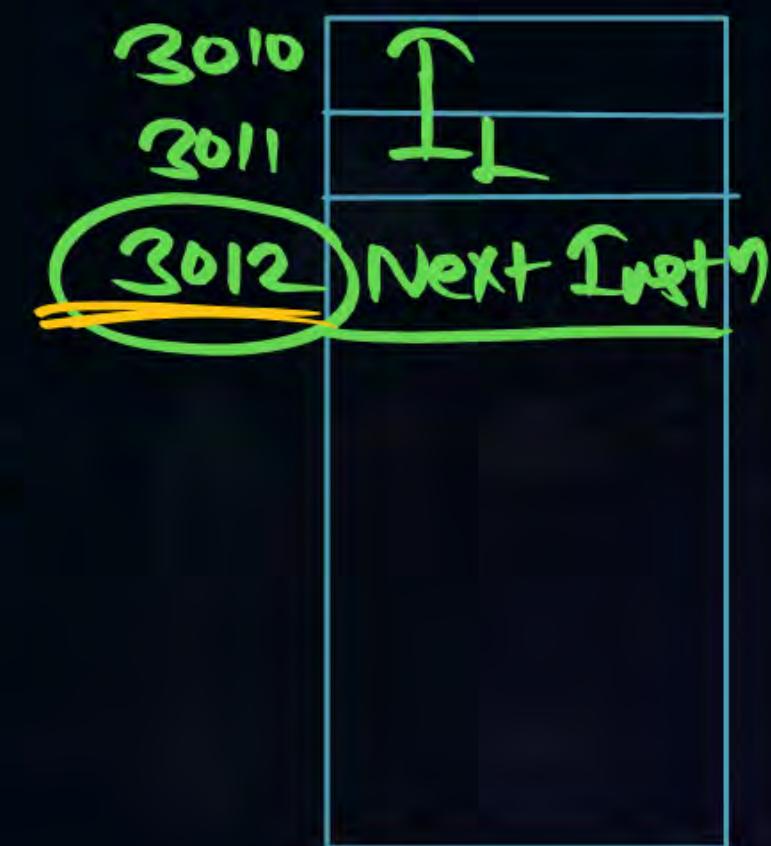
$$\text{OAF} = 1 \times 2^8$$

$$= \textcircled{256}$$

#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
- C 3003 H

- Relative Value*
- |           |          |
|-----------|----------|
| JMP   -7. | B 3009 H |
| D 3007 H  |          |



$$\begin{aligned}
 \text{Target Address} &= \frac{\text{Current PC Value}}{\text{PC Value}} + \frac{\text{Relative Value (OFFSET)}}{\text{AF}} \\
 &\Rightarrow 3012 - 7 \\
 &= 3005 \text{ Ans}
 \end{aligned}$$

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

instruction	Operation	Size(in words)
MOV R <sub>0</sub> , 2000	R <sub>0</sub> $\leftarrow$ M[2000]	2 $2 \times 1 + 3 = 5$
MOV R <sub>1</sub> , [R <sub>0</sub> ]	R <sub>1</sub> $\leftarrow$ M[R <sub>0</sub> ]	2 $2 \times 1 + 3 = 5$
SUB R <sub>1</sub> , R <sub>2</sub>	R <sub>1</sub> $\leftarrow$ R <sub>1</sub> - R <sub>2</sub>	1 $1 \times 1 + 1 = 2$
MOV 4000, R <sub>1</sub>	M[4000] $\leftarrow$ R <sub>1</sub>	2 $2 \times 1 + 3 = 5$
HALT	Stop	1 $1 \times 1 + 0 = 1$

let the clock cycles required for operations are

Register to/from m/ r transfer = 3 CC

SUB with both registers operands = 1CC

Instruction fetch and decode = 1 CC per word

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

18 cycle

Ans(1B)

18 Ans

[NAT]

$$1\text{ word} = 4\text{ Byte} \Rightarrow 1\text{ word}$$

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

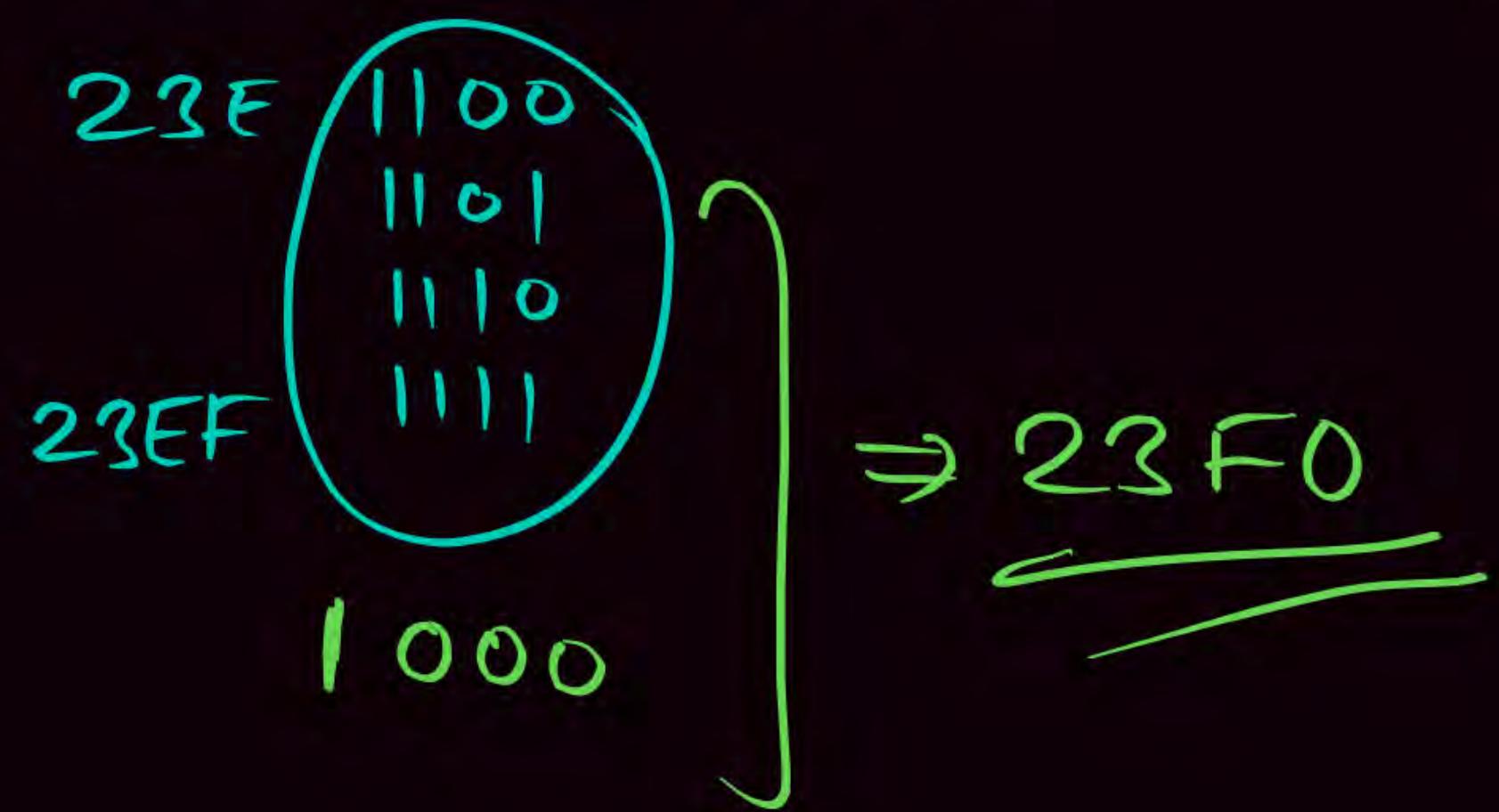
Instruction	Operation	Instruction size (in byte)
Mov $r_0, @7000$	$r_0 \leftarrow M[(7000)]$	4W 16 23EC - 23EF
Mov $r_1, @r_2$	$r_1 \leftarrow M[[r_2]]$	3W 12 23F0 - 23F2
ADD $r_0, r_1$	$r_0 \leftarrow r_0 + r_1$	2W 8 23F3 - 23F4
MUL $r_0, r_1$	$r_0 \leftarrow r_0 \times r_1$	2W 8 23F5 - 23F6
MOV $@7000, r_0$	$M[[7000]] \leftarrow r_0$	4W 16 23E7 - 23FA
Halt	Machine halt	2W 8 23FB - 23FC

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)

4w

23 EC - 23 EF



4w

E C

1110	1100
1101	
1110	
1110	1111
<hr/>	
F	O

[NAT]

P  
W

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

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

$$\text{Avg CPI} = \frac{40000 + 33000 \times 2 + 9000 \times 1 + 11000 \times 3 + 7000 \times 7}{1,00,000}$$

$$\begin{aligned}\text{Avg CPI} &= 1.62 \text{ cycle} \\ &\Rightarrow 1.62 \times 2 \\ &\Rightarrow 3.24 \text{ nsec}\end{aligned}$$

The MIPS (millions of Instructions per second) rate, for this program is  
 (Note: Ignore the fractional part)

Avg(308)

$$1 \text{ Dust}^n \longrightarrow 3.24 \times 10^{-9} \text{ sec}$$

$$D_n L \text{ sec} \longrightarrow \frac{1}{3.24} \times 10^9 \text{ Dust}^n / \text{sec}$$

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

$$\Rightarrow 308 \text{ MEPS.}$$

**[NAT]**

P  
W

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  $R_1 \#1 \cdot R_1 \leftarrow R_1 + 1$

$I_2$ :  $R_2 \leftarrow R_2 + m(100 + R_1)$

### 1<sup>st</sup> Iteration

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

$$I_2: R_2 \leftarrow 1000 + m(100 + 1)$$

$$1000 + m(101)$$

$$R_2 = 1000 + 2$$

$$R_2 = 1002$$

### 2<sup>nd</sup> Iteration

$$I_1: R_1 \in [1, 1] \Rightarrow R_1 = 2$$

$$I_2: R_2 \leftarrow 1002 + m(100 + 2)$$

$$1002 + 3$$

$$R_2 = 1005$$

$$R_1 = 0$$

$$R_2 = 1000$$

### 3<sup>rd</sup> Iteration

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

$$R_2 \leftarrow 1005 + m(100 + 3)$$

$$R_2 = 1005 + 4$$

$$R_2 = 1009$$

$$\underbrace{1002}_{\textcircled{1}} + \underbrace{3}_{\textcircled{2}} + \underbrace{4}_{\textcircled{3}} + \underbrace{5}_{\textcircled{4}} + \underbrace{6}_{\textcircled{5}} + \underbrace{7}_{\textcircled{6}} + \underbrace{8}_{\textcircled{7}}$$

$$\underbrace{1035}_{\text{Avg}}$$

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)

<u>OP</u>	<u>500</u>
400	OPCODE (8bit)
401	500 (8bit)
PC	402
Next Instruction	
499	750
500	900
600	825



PC Relative AM

$$TA(EA) = \frac{\text{Current PC Value}}{\text{Relative Value}} + AF(OFFSET)$$

$$\Rightarrow \underline{402} + \underline{500}$$

$$= \underline{902} \text{ Ans}$$

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

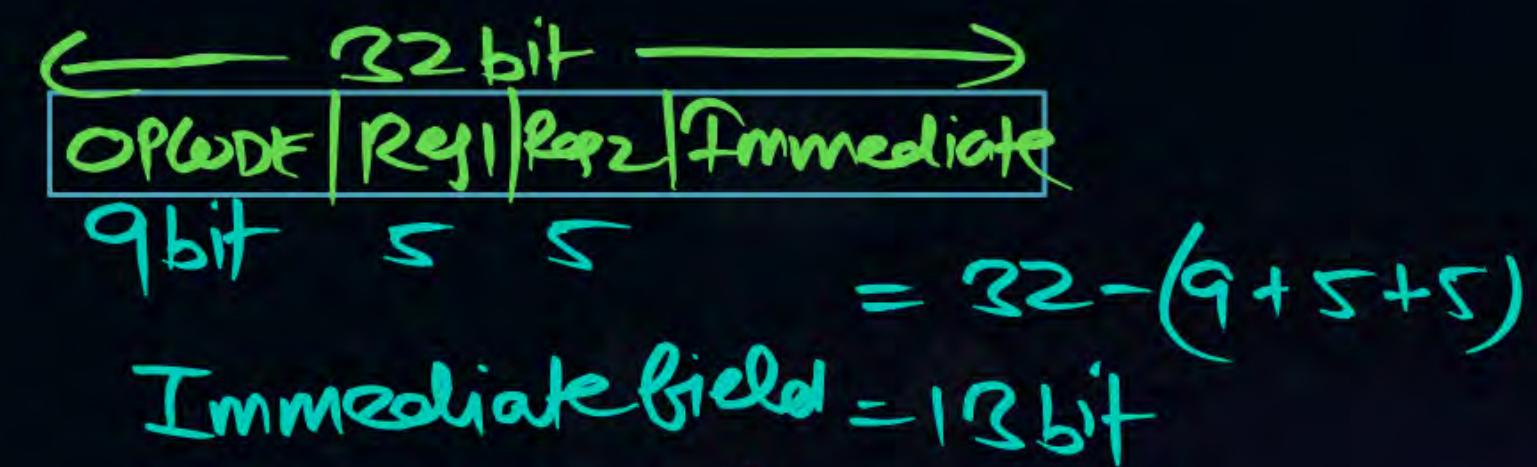
#Q. A machine has 32 bit Architecture with 1 word long instructions. It has 28 Register each of which is 32 bit long. Machine support 300 instruction, which have an immediate operand in addition to two register operand. Assuming that the immediate operand is an unsigned integer. Then the maximum value of the immediate operand is 8191 Ans

Instn Size = 32 bit

28 Reg  $\Rightarrow$  Reg AF = 5 bit

300 operation  $\Rightarrow$  Opcode = 9 bit

Ans (8191)



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

$$\Rightarrow 8192 - 1 = 8191$$

Ans

- #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 ~~on~~words.

Ans (2040)

2000	I <sub>1</sub> :	ADD	R <sub>0</sub>	R <sub>1</sub>	R <sub>2</sub>
2004	I <sub>2</sub> :	BEQ	R <sub>3</sub>	R <sub>4</sub>	Lable (OFFSET)
2008	I <sub>3</sub> :	MUL	R <sub>5</sub>	R <sub>5</sub>	R <sub>6</sub>
2012	I <sub>4</sub> :	ADD	R <sub>7</sub>	R <sub>5</sub>	R <sub>2</sub>
2016	I <sub>5</sub> :	MUL	R <sub>8</sub>	R <sub>1</sub>	R <sub>6</sub>

(BEQ: Branch if equal)

If R<sub>3</sub> == R<sub>4</sub>  
OFFSET: 32

Here label is used as an offset. R<sub>0</sub>, R<sub>1</sub>, R<sub>2</sub>, R<sub>3</sub> ... R<sub>8</sub> are general purpose registers. The BEQ instruction branches the PC if the register R<sub>3</sub> and register R<sub>4</sub> content are equal then label = 32. The value of register R<sub>0</sub> = 10, R<sub>1</sub> = 20, R<sub>2</sub> = 30, R<sub>3</sub> = 40, R<sub>4</sub> = 40, R<sub>5</sub> = 50, R<sub>6</sub> = 60 respectively. What is the memory address (in decimal) of the next instruction to be executed 2040 Ans

~~Label = OFFSET + 32~~

## PC Relative AM.

$$EA(TA) = \frac{\text{Current PC Value}}{\text{PC Value}} + AF(\text{OFFSET})$$

Relative Value

When I<sub>2</sub> Fetch then

PC value = 2008

(I<sub>3</sub> starting address)

$$\begin{aligned} & R_3 = R_4 \\ & 2008 + 32 \\ & = 2040 \text{ Ans} \end{aligned}$$

So PC Relative OFFSET = 32

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

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

	A	B	C	D	E	$\text{Ans}(A) \& (B)$		A	B	C	D	E
A	2	5	5	1	3			1	5	5	1	3
B												
C	4	3	2	5	1							
D								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.

$$R_1 = 2000, R_2 = 4000$$

2000	10
4000	20

I<sub>1</sub>: Add  $R_2, R_2, (2000) R_1$ ; displacement addressing

I<sub>2</sub>: Add  $R_2, R_2, (R_1)$ ; Register Indirect

The value of  $R_2$  after executing above instructions

is 4030 Ans

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

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

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

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

$$R_2 = 4030$$

Ans

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

S<sub>1</sub>: ✗ Immediate AM are used to access the variable

S<sub>2</sub>: ✗ In the Indirect AM address field of the instruction contain the address where the operand is stored.

S<sub>3</sub>: Index AM are used to implement the Array.

S<sub>4</sub>: ✗ Absolute AM are used to implement the pointer.

Which of above statements is/are incorrect?

A S<sub>1</sub>

Ans (A)(B)&(D)

B S<sub>2</sub>

C S<sub>3</sub>

D S<sub>4</sub>

→ Constant

Immediate AM - Constant

Direct / Absolute - Variable

Indirect AM - Pointer

Index AM - Array



# Home Work

#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 ;  $R_1 = 0$ 
MOV R2, #1 ;  $R_2 = 1$ 
CMP R3, #0 ; Compare  $R_3$  with 0
BEQ DONE ; Branch to DONE if zero flag is set
```

X:

```
ADD R2, R1, R2 ;  $R_2 \leftarrow R_1 + R_2$ 
SUB R1, R2, R1 ;  $R_1 \leftarrow R_2 - R_1$ 
SUB R3, R3, #1 ;  $R_3 \leftarrow R_3 - 1$ 
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

C

144

B

89

D

None of these

## [MCQ]

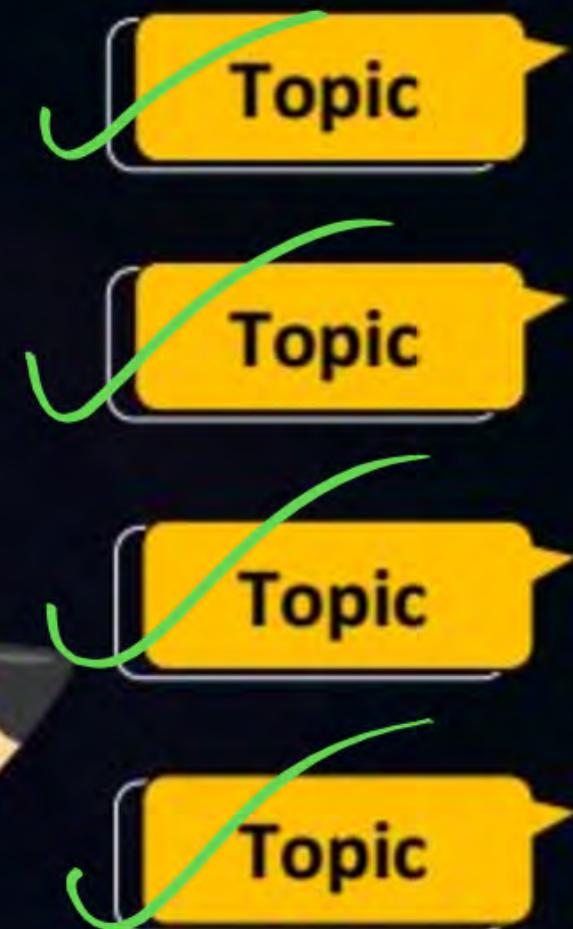
Consider a design will expand opcode technique with 12bit instructions, where a register operand requires 3 bits. There are 24 – 2 address instruction consisting of two register operands, and 5-one address instructions consisting of one memory operand of 8 bits. Then find the number of 0-address instruction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

- (a) 512
- (b) 1024
- (c) 2048
- (d) Not possible with the given instruction size.  
None of these.





## 2 mins Summary



Clock Cycle Concept.

Machine Instruction

Expand Opcode Technique

Addressing Modes



THANK - YOU

# CS & IT ENGINEERING

## Computer Organization & Architecture

1500 Series

Lecture No.- 03



By- Vijay Agarwal Sir



# Recap of Previous Lecture



Topic

Clock Cycle Concept.

Topic

Machine Instruction

Topic

Expand Opcode Technique

Topic

Addressing Modes



# Topics to be Covered



Topic

Expand Opcode Technique

Topic

ALU Data Path

Topic

Micro Operation & Micro Program

Topic

Control Unit

#Q. Consider the following assembly level program for a hypothetical processor.

$R_1$ ,  $R_2$  and  $R_3$   
are 32-bit registers.

MOV  $R_1$ , #0  
MOV  $R_2$ , #1  
CMP  $R_3$ , #0  
BEQ DONE

X:

ADD  $R_2$ ,  $R_1$ ,  $R_2$  ;  $R_2 \leftarrow R_1 + R_2$   
SUB  $R_1$ ,  $R_2$ ,  $R_1$  ;  $R_1 \leftarrow R_2 - R_1$   
SUB  $R_3$ ,  $R_3$ , #1 ;  $R_3 \leftarrow R_3 - 1$

BNE X

DONE:

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

$R_1 = 0$   
;  $R_2 = 1$   
; Compare  $R_3$  with 0  
; Branch to DONE if zero flag is set

; Jump to X if zero flag is not set.

A

55

Ans (B)

C

144

B

89

D

None of these

$I_1: ADD \quad R_2 \leftarrow R_1 + R_2$

$I_2: \quad R_1 \leftarrow R_2 - R_1$

$I_3: SUB \quad R_3 \leftarrow R_3 - 1$

$$\begin{aligned}R_1 &= 0 \\R_2 &= 1 \\R_3 &= 10\end{aligned}$$

$$\begin{aligned}I_1: R_2 &= 0 + 1 = \textcircled{R_2=1} & I_1: R_2 &= 1 + 1 = \textcircled{R_2=2} \\I_2: R_1 &= 1 - 0 = \textcircled{R_1=1} & I_2: R_1 &= 2 - 1 = \textcircled{1} \\I_3: R_3 &= 10 - 1 = \textcircled{R_3=9} & R_3 &= 9 - 1 = 8\end{aligned}$$

$$\begin{aligned}I_1: 3+5 &= \textcircled{R_2=8} \\I_2: 8-3 &= 5 \\I_3: R_3 &= 6-1 = 5\end{aligned}$$

$$\begin{aligned}I_1: R_2 &= 1+2 = \textcircled{3} & I_1: R_2 &= 2+3 = \textcircled{5} \\I_2: R_1 &= 3-1 = 2 & I_2: 5 - 2 = \textcircled{3} \\R_3 &= 8-1 = 7 & R_3 &= 7-1 = 6\end{aligned}$$

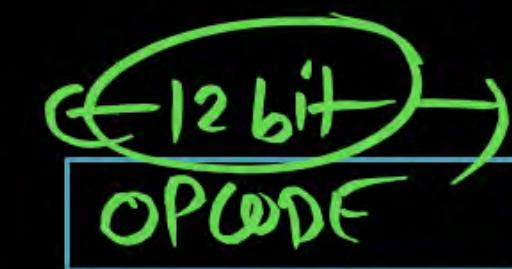
$R_2$
1
2
3
5
8
13
21
34
55
89

## [MCQ]

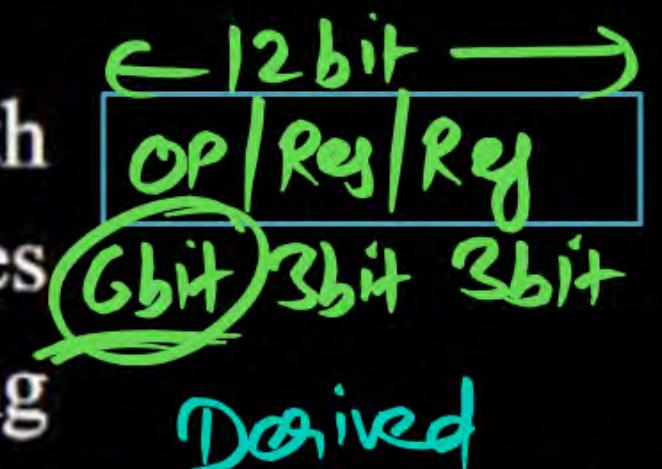
P  
W

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.



$$20 \times 2^{12-6} \Rightarrow 20 \times 2^6$$



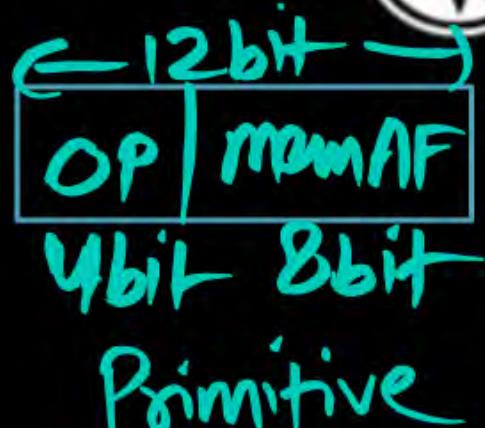
Derived

$$\begin{aligned} \text{Total} &= 11 \times 2^4 \\ \text{Operation} &= 11 \times 4 \end{aligned}$$

$$\begin{aligned} \text{Given} &= 2^4 \\ \text{Free} &= 2^4 - 2^4 \end{aligned}$$

$$20 \times 2^6 \Rightarrow 20 \times 64 = 1280$$

Ans



$$\begin{aligned} \text{Total #} &= 2^4 \\ \text{Operation} &= 16 \end{aligned}$$

Given = 5

$$\begin{aligned} \text{Free} &= 16 - 5 \\ &= 11 \end{aligned}$$

#Q. in  $X = \frac{(M+N \times O)}{(P \times Q)}$ , how many one-address instruction are required to evaluate it?

A

4

 $\text{Ans}(C)$ 

C

8

B

6

D

10

 $I_1: \text{LOAD } P : AC \leftarrow M[P]$  $I_2: \text{MUL } Q : AC \leftarrow AC * Q$  $I_3: \text{STORE } T; [T] \leftarrow AC$  $I_4: \text{LOAD } N : AC \leftarrow M[N]$  $I_5: \text{MUL } O : AC \leftarrow N * O$  $I_6: \text{ADD } M : AC \leftarrow M + (N * O)$  $I_7: \text{DIV } T : AC \leftarrow AC / M[T]$  $I_8: \text{STORE } X : M[X] \leftarrow \frac{M + N * O}{P * Q}$ 

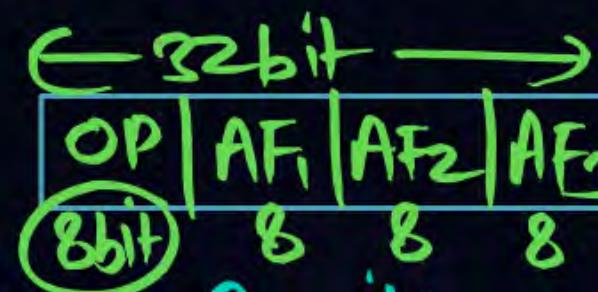
8 Instruction

#Q. A computer supports 32-bit wide instructions and 8-bit wide addresses. If it supports  $x$  3 -address instructions,  $y$  2-address instructions, how many 1-address instructions can be supported?  $x, y \neq 0$

A  $2^{24}$

Ans(D)

C  $((2^8 - x) \cdot y) \cdot 2^8$



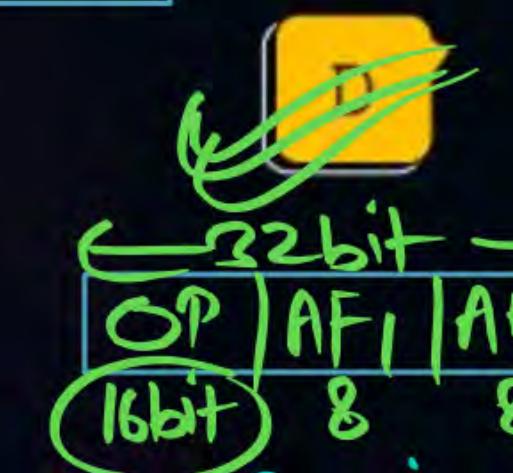
Primitive  
Total operation =  $2^8$

Given =  $x$

Free =  $(2^8 - x)$

B  $((2^{16} - x) - y) \cdot 2^8$

~~D~~



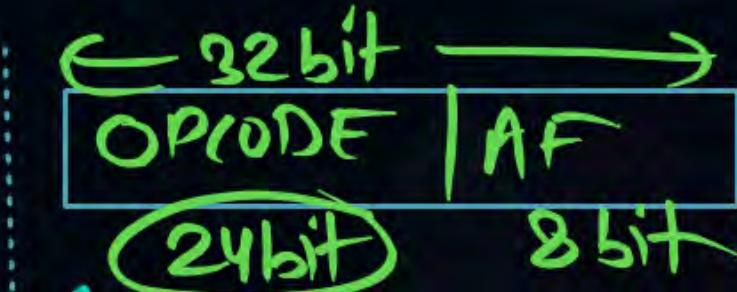
Derived

Total operation =  $(2^{16} - x) \cdot 2^8$

Given =  $y$

Free =  $(2^{16} - x \cdot 2^8) - y$

D  $((2^{16} - x \cdot 2^8) - y) \cdot 2^8$



Further Derived

$(2^{24} - 2^{16})$

Total =  $\left[ ((2^{24} - 2^{16}) - y) \cdot 2^8 \right] \cdot \underline{\text{Ans}}$

# T.D

#Q. Consider a hypothetical control unit which supports:

4 control signals  $\{S_0, S_1, S_2, S_3\}$

3 instruction  $\{I_1, I_2, I_3\}$

Each instruction takes 4  $\mu$ -instructor  $\{T_1, T_2, T_3, T_4\}$  to complete the execution.

The following table shows the control signals request for each  $\mu$ -operator for each instructor the control operator for  $S_0$  and  $S_3$ ?

$\mu$ - Op $\downarrow$ inst $\rightarrow$	$I_1$	$I_2$	$I_3$
$T_1$	$S_0, S_2$	$S_1, S_3, S_2$	$S_0, S_3, S_1$
$T_2$	$S_0, S_3, S_2$	$S_1, S_0, S_3$	$S_0, S_2, S_1$
$T_3$	$S_1, S_0, S_2$	$S_1, S_2, S_3$	$S_1, S_2$
$T_4$	$S_1, S_2, S_3$	$S_0, S_2$	$S_1, S_2$

$$S_0 = T_1(I_1 + I_3) + T_2(I_1 + I_2 + I_3)$$

$$+ T_3 I_1 + T_4 I_2$$

$$S_0 = T_1(I_1 + I_3) + T_2 + T_3 I_1 + T_4 I_2$$

Ans

$$S_3 = T_1(I_2 + I_3) + T_2(I_1 + I_2) + T_3 I_2$$

$$+ T_4(I_1 + I_2)$$

Continues ...

A

$$\begin{aligned}S_0 &= T_1(I_1 + I_3) + T_2 + T_3 I_1 + T_4 I_2 \\S_3 &= T_1 \cancel{(I_2 + I_3)} + T_2(I_1 + I_2) + T_3 \cancel{I_2} \\&\quad T_4 \cancel{(I_1 + I_2)}\end{aligned}$$

Ans (a)

B

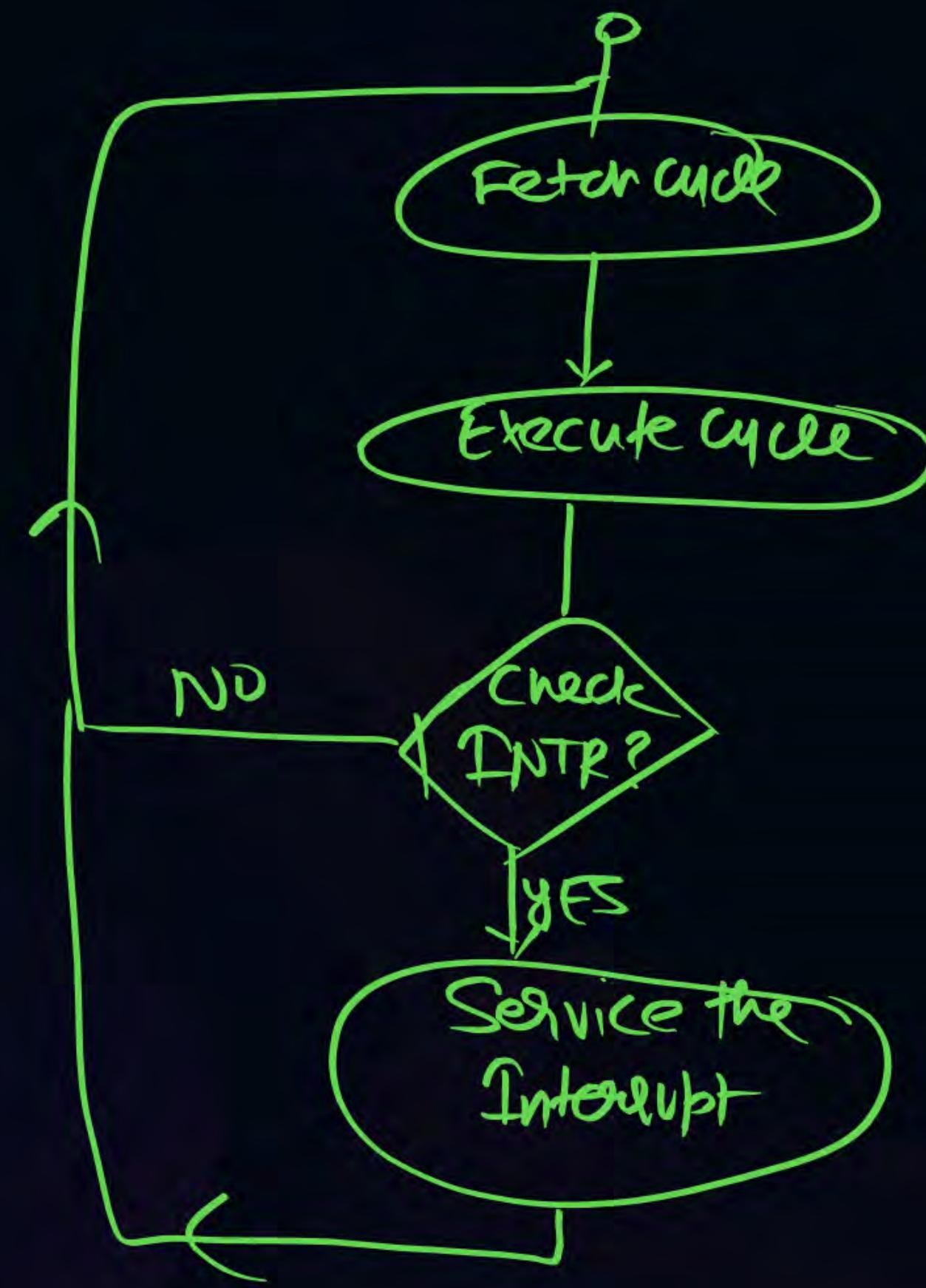
$$\begin{aligned}S_0 &= T_1(I_1 + I_3) + T_2(I_1 + I_3) + T_3 I_1 + T_4 I_2 \\S_3 &= T_1(I_2 + I_3) + T_2(I_1 + I_2) + T_3 I_2 + T_4(I_1 + I_2)\end{aligned}$$

C

$$\begin{aligned}S_0 &= T_1(I_1 + I_3) + T_2 + T_3 I_1 + T_4 I_2 \\S_3 &= T_1(I_1 + I_3) + T_2(I_1 + I_2) + T_3 I_3 + T_4(I_1 + I_2)\end{aligned}$$

D

None



## [MCQ]



Consider the following micro-operations about instruction fetch (IF):

- (i) The content of the MDR are loaded into the IR.
- (ii) The result of a memory read operation, the instruction is loaded in to the MDR.
- (iii) The content of the program counter is loaded into the MAR.

Which of the following is correct sequence of instructions fetch in micro program?

(iii)  $PC \rightarrow MAR$

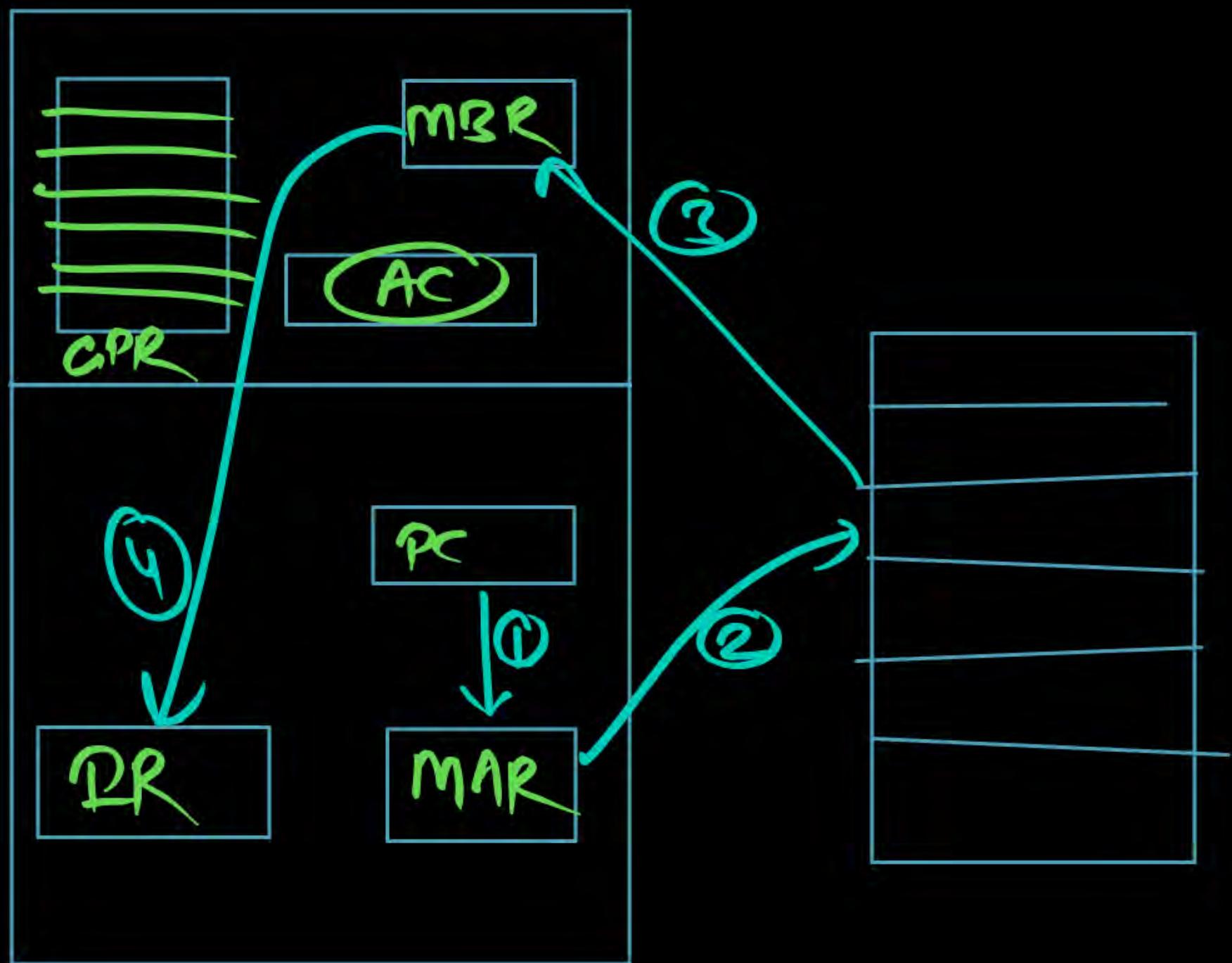
A (iii), (ii), (i) (ii)  $Mem \rightarrow MBR/MDR$

B (i), (iii), (ii) (i)  $MBR \rightarrow DR$ .

C (i), (ii), (iii)

Ans (A)

D (ii), (iii), (i)



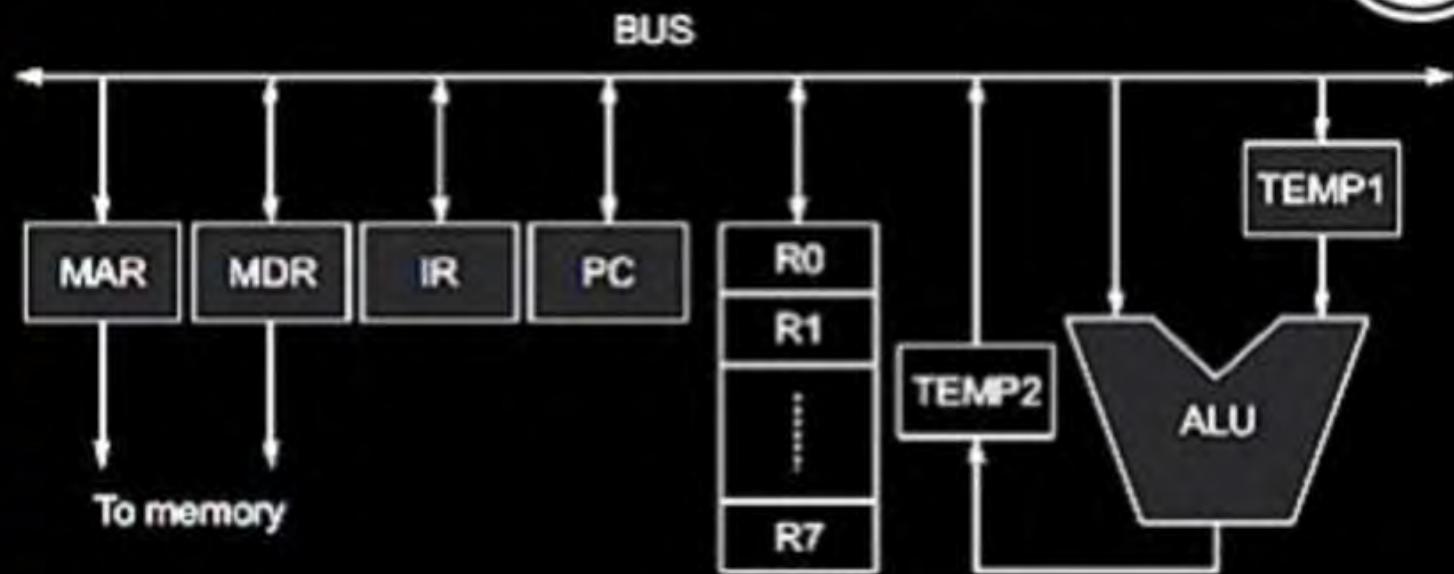
**MCQ**

Consider the following data path diagram

Consider an instruction:  $R0 \leftarrow R1 + R2$ . The following steps are used to execute it over the given data path. Assume that PC is incremented appropriately. The subscripts r and w indicate read and write operations, respectively.

1.  $R2_r$ ,  $TEMP1_R$ ,  $ALU_{add}$ ,  $TEMP2_w$
2.  $R1_r$ ,  $TEMP1_w$ ,
3.  $PC_r$ ,  $MAR_w$ ,  $MEM_r$
4.  $TEMP2_R$ ,  $R0_w$
5.  $MDR_r$ ,  $IR_w$

Which one of the following is the correct order of execution of the above steps?



- A 3, 5, 1, 2, 4
- B 2, 1, 4, 5, 3
- C 3, 5, 2, 1, 4
- D 1, 2, 4, 3, 5

Which of the following is/are INCORRECT about micro instruction?

- A Individual bits in horizontal micro instructions correspond to individual control lines. → **CORRECT**
- B Vertical micro instructions are much shorter than horizontal ones. **Correct**
- C Vertical micro instructions are allowed maximum parallelism.   
*low Degree of Parallelism*
- D Decoding is necessary in both micro instructions, horizontal as well as vertical.

**Ans(C) & (D)**

**No**

Consider a CPU where all the instructions require 9 clock cycles to complete execution. There are 200 instructions in the instruction set. It is found that 130 control signals are needed to be generated by the control unit. While designing the horizontal micro-programmed control unit, single address field format is used for branch control logic. What is the minimum size of the control word and control address register?

$$\text{Total } \# \text{Inst}^n = 200$$

$$\# \text{Cycle} | \text{Inst}^n = 9$$

$$\text{Total } \# \text{Microop}^n = 200 \times 9 = 1800 \text{ Microop}^n | \text{Cw}$$

$$\text{Control Memory} = 1800 \text{ Cw}$$

$$\text{CAR} | \text{AF} = 11 \text{ bit}$$

~~Horizontal~~  
~~CS = 130~~  $\Rightarrow CF = 130 \text{ bit}$

CF	IAF
----	-----

130  
LL

14L, LL  
Aug

# MCQ



Consider the following sequence of micro -operations.

$\text{MBR} \leftarrow \text{PC}$

$\text{MAR} \leftarrow \text{X}(\text{SP(Tos)})$

$\text{PC} \leftarrow \text{Y}(\text{ISR})$

~~$\text{Tos(SP)}$~~   
 $\text{Memory} \leftarrow \text{MBR}$

Which one of the following is a possible operation performed by this sequence

A  Instruction fetch

~~Avg(D)~~

C  Conditional branch

B  operand fetch

D  Initiation of interrupt service

@ Instr Fetch

PC → MAR → Mem → MBR → DR

b) Operand Fetch

DR(AF) → MAR → MEM → MBR → AC/ALU.

Consider a CPU where all instruction takes 11 cycles to complete execution. There are total 290 instructions in a instruction set. In this 3190 control signals are needed to be generated by control unit while single address field format is used for designing the vertical micro-programmed control unit. Then the size of control memory (in byte) is 9570 Ans

$$\frac{2^y}{8} = 384t$$

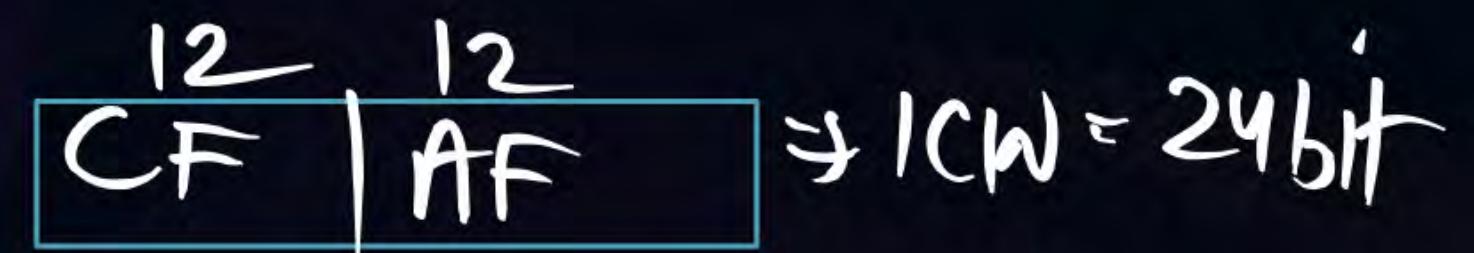
$$\text{Total } n \text{ operation} = 290 \times 11 = 3190 \text{ n'Opern/CW}$$

$$\text{Control Memory} = 3190 \text{ CW} \Rightarrow 2^{12}$$

Vertical

$$AF | CAR | NIA = 12 \text{ bit}$$

$$CS = 3190 \Rightarrow CF = 12 \text{ bit}$$



$$\begin{aligned}
 CM &= 3190 \text{ CW} \\
 &\Rightarrow 3190 \times 24 \text{ bit} \\
 &\Rightarrow 3190 \times 284 \text{ byte} \\
 &= 9570 \text{ Ans}
 \end{aligned}$$

**[MCQ]**

Consider the following microprogram to fetch the data from the memory to CPU register ( $r_1$ ) using the indirect addressing mode:

$$T_1: P \rightarrow \text{MAR} \longrightarrow \text{TR} \rightarrow \text{MAR}$$

$$T_2: \text{Memory} \rightarrow \text{MBR}$$

$$T_3: \underline{\text{MBR}} \rightarrow Q \longrightarrow \text{MBR} \rightarrow \text{MAR}$$

$$T_4: \text{Memory} \rightarrow \text{MBR}$$

$$T_5: \text{Memory} \rightarrow R \longrightarrow \text{Memory} \rightarrow r_1$$

What are the registers used in the place of P, Q and R variables respectively?

A  $r_1, \text{PC}, \text{MAR}$

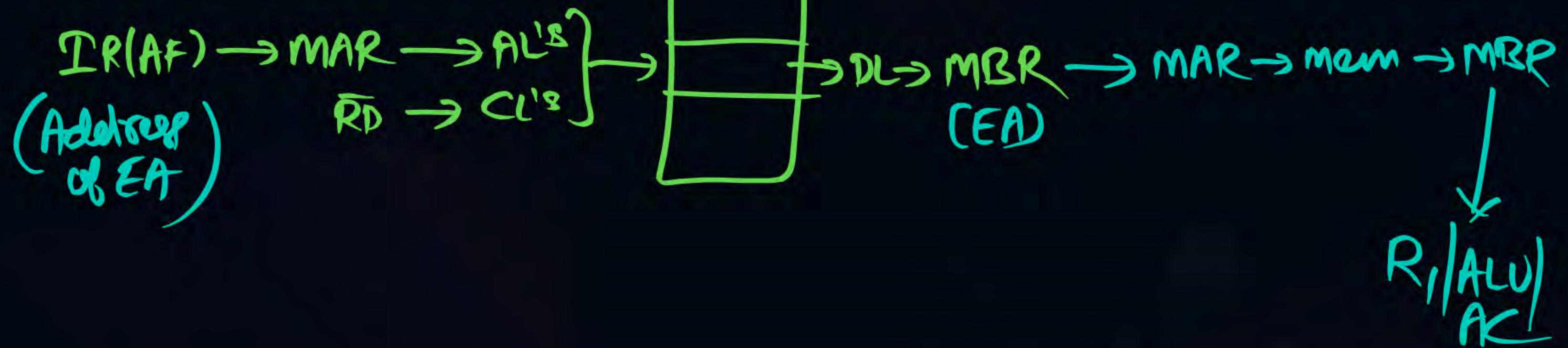
~~Ang(B)~~

~~B~~ IR, MAR,  $r_1$

C  $r_1, \text{MAR}, \text{IR}$

D None of these

## Indirect AIM



Mumbai

2014  
2015

IIT

Which of the following characteristics are correct about the design of RISC processor?

- A It support more number of registers
- B It support less number of addressing modes
- C It uses micro programmed control unit  
*RISC  $\Rightarrow$  Hardwired CU.*
- D It support smaller instruction set

Ans (A)(B) & (D)

#Q. Consider a hypothetical control unit that support 9 groups of mutually exclusive control signals. Also assume that group-1, group-2, group-3 and group-4 are using horizontal micro-programming where as group-5, group-6, group-7, group-8, and group-9 are using vertical microprogramming. Identify which of the following is/are in correct?

Groups	G <sub>1</sub>	G <sub>2</sub>	G <sub>3</sub>	G <sub>4</sub>	G <sub>5</sub>	G <sub>6</sub>	G <sub>7</sub>	G <sub>8</sub>	G <sub>9</sub>
Control Signal	8	9	10	11	19	31	40	51	74

A

Total bits for groups (G<sub>5</sub> G<sub>6</sub> G<sub>7</sub> G<sub>8</sub> G<sub>9</sub>) is 29 bits and Total bits for control word is 42.

B

Total bits for Group (G<sub>1</sub> G<sub>2</sub> G<sub>3</sub> G<sub>4</sub>) is 38 bits and total bits for control word is 43.

C

Total bits for control word is 67 bits. → wrong

Ans (A)(B) ↘(D)

D

Total bits for control word is 46 bits

Horizatlon $G_1 \ G_2 \ G_3 \ G_4$  $8 \ 9 \ 10 \ 11$ 

$$\text{bit} = 8 + 9 + 10 + 11$$

$\Rightarrow$  38 bit

Vertical $G_5 \ G_6 \ G_7 \ G_8 \ G_9$  $19 \ 31 \ 40 \ 51 \ 74$   
 $\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$ 

( $\log_{10} 2^n$ )

$$\text{bits} = 5 + 5 + 6 + 6 + 7 = 29 \text{ bit}$$

$$\begin{aligned} \text{CH} &= 38 + 29 \\ &= 67 \text{ bits} \end{aligned}$$



## 2 mins Summary



Topic

Expand Opcode Technique

Topic

ALU Data Path

Topic

Micro Operation & Micro Program

Topic

Control Unit



THANK - YOU

# CS & IT ENGINEERING

## Computer Organization & Architecture

1500 Series

Lecture No.- 04

By- Vijay Agarwal Sir





# Recap of Previous Lecture



Topic

Expand Opcode Technique

Topic

ALU Data Path

Topic

Micro Operation & Micro Program

Topic

Control Unit



# Topics to be Covered



Topic

Pipeline

Topic

Pipeline Hazards

## PIPELINE

Min. 2 Marks

( 3-4 Marks )

- PIPELINE CONCEPT [ET, Speed up factors, efficiency, Throughput]
- Timing Diagram.
- PIPELINE HAZARDS.

## Pipeline Hazards

- ↳ ① Structural Hazard
- ② Data Hazards
- ↳ ③ Control Hazards.

1. A five-stage pipeline has stage delays of 100, 120, 125, 95 and 90 nanoseconds. The register that are used between the pipelined stages have a delay of 5 nano seconds. The total time to execute 500 independent instruction on this pipeline, assuming no pipeline stalls, is \_\_\_(nanoseconds, upto 1 decimal)  $k = 5$

$$t_p = \max (\text{Stage Delay} + \text{Buffer Delay})$$

$$t_p = 130 \text{ nsec}$$

$$n = 500$$

$$\boxed{\text{Ans} (65,520)}$$

$$\begin{aligned}
 ET_{PIPE} &= [k + (n-1)] t_p \\
 &\Rightarrow [5 + (500-1)] \times 125 \\
 &= 504 \times 136 \\
 &= 65,520 \text{ nsec} \quad \text{Ans}
 \end{aligned}$$

2 The performance of a pipelined processor has no effect if:

- A The pipeline stages share hardware resources.  $\Rightarrow$  Structural Deb.
- B The pipelined stages have different delays.  $\Rightarrow$  Control Deb.
- C The consecutive instruction are dependent on one another.  $\Rightarrow$  Data Dependency
- D None of these.

Ans (D)

## The Performance of Pipeline effect Due to

- ① Structural Hazard | Dependency  $\Rightarrow$  Resource Conflict  
Share the H/w Resource at same time.
- ② Data Dependency  $\Rightarrow$  Consecutive Inst<sup>n</sup> depend the result of previous
- ③ Control Dependency  $\Rightarrow$  Branch operation @ Different Delay.

Q. A 5-stage pipeline runs at 1 GHz. The pipeline can overlap all the instructions except branch instructions. Branch instructions incur 1 stall. If 25% instructions are branches, throughput of the system in MIPS (Million Instruction Per Second) is 8000.

$$\text{Cycle time} = \frac{1}{f} \text{ sec}$$

$$1 \text{ sec} = 10^9 \text{ cycles}$$

Ans (800)

$$\text{Cycle time} = L \text{ nsec}$$

$$RTF = 25 \cdot 1$$

$$\text{Stall} = 1$$

$$\# \text{ Stall} / \text{Inst} = 25 \times 1 = 25$$

Avg Instn ET =  $(1 + \# Stalls/Inst^n) \times \text{Cycle time}$

$$\Rightarrow (1 + .25) \times 1\text{nsec}$$

$$ET = 1.25 \text{nsec}$$

$$\text{Throughput} = \frac{1}{ET} \Rightarrow \frac{1}{1.25} \times 10^9$$

$$= \frac{1000 \times 10^6}{1.25} = 800 \times 10^6 \text{ Instn/sec}$$

= 800 MIPS Avg

#Q. **y** The stage delay in a 5-stage pipeline is 400, 600, 500, 400, and 200 respectively in picoseconds. The second stage with delays 600 picoseconds is replaced with functionality equivalent design involving two stages 400 and 200 picoseconds. What is the percentage increase in throughput?

A

16.67

C

25

B

20

D

30

$\text{Ans(B)}$

OLD Design

$$tp = \max(400, 600, 500, 400, 200)$$

$$tp_{\text{old}} = 600$$

$$\text{Throughput}_{\text{old}} = \frac{1}{600}$$

New Design

$$tp = \max(400, 400, 200, 500, 400, 200)$$

$$tp_{\text{new}} = 500 \text{ sec}$$

$$\text{Throughput}_{\text{new}} = \frac{1}{500}$$

$$\therefore \text{Increment in Throughput} = \frac{\text{New} - \text{OLD}}{\text{OLD}} \Rightarrow \frac{\frac{1}{500} - \frac{1}{600}}{\frac{1}{600}} = \frac{1}{6}$$

$$\Rightarrow \frac{6-5}{30} \times 6$$

$$\Rightarrow \frac{1}{30} \times \frac{1}{5} = \frac{1}{150} = 20\% \text{ Avg}$$

#Q. A pipeline P operating at 400 MHz has a speed up factor of 6 and operating at 70% efficiency. How many stages are there in pipeline?

A

5

C

8

B

6

D

9

Ans(D).

$$\eta = \frac{S}{K}$$
$$70\% = \frac{S}{K} \Rightarrow K = \frac{S}{70\%} = \frac{60}{70\%} = 8.57 \approx 9$$

#Q.6. An 8-stages perfectly balanced instruction pipeline has ~~one~~-cycle-time overhead. If 35% of the instruction in 4 pipeline stall cycles, the speedup achieved with respect to non-piped execution when an application is executing on this 8-stage pipeline is 3.33 Ans

Ans (3.33).

K = 8 Stage (Perfectly Balanced)

$$BIF = 35.1$$

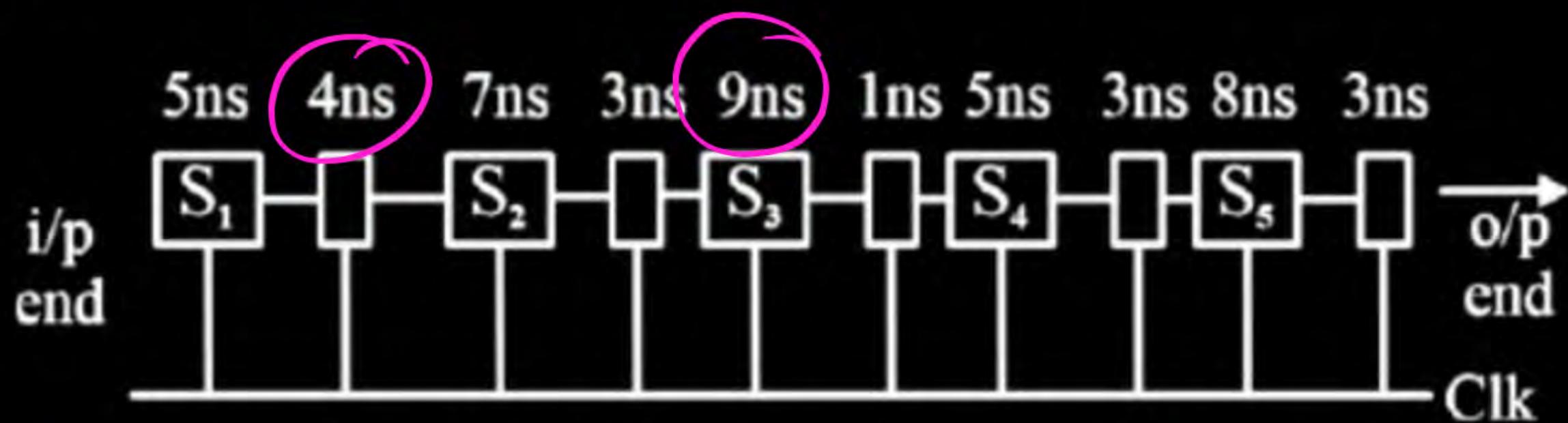
Stall = 4 Stalls

$$\# \text{Stalls} / \text{Inst^n} = 35 \times 4 = 1.4$$

$$\begin{aligned}\text{Avg Inst^n ET} &= (1 + \# \text{Stalls} / \text{Inst^n}) \\ &= (1 + 1.4) = 2.4\end{aligned}$$

$$\begin{aligned}\text{Speed Up Factor} &= \frac{\# \text{Stages}(K)}{(1 + \# \text{Stalls} / \text{Inst^n})} \Rightarrow \frac{8}{(1 + 1.4)} = \frac{8}{2.4} = 3.33 \text{ Avg}\end{aligned}$$

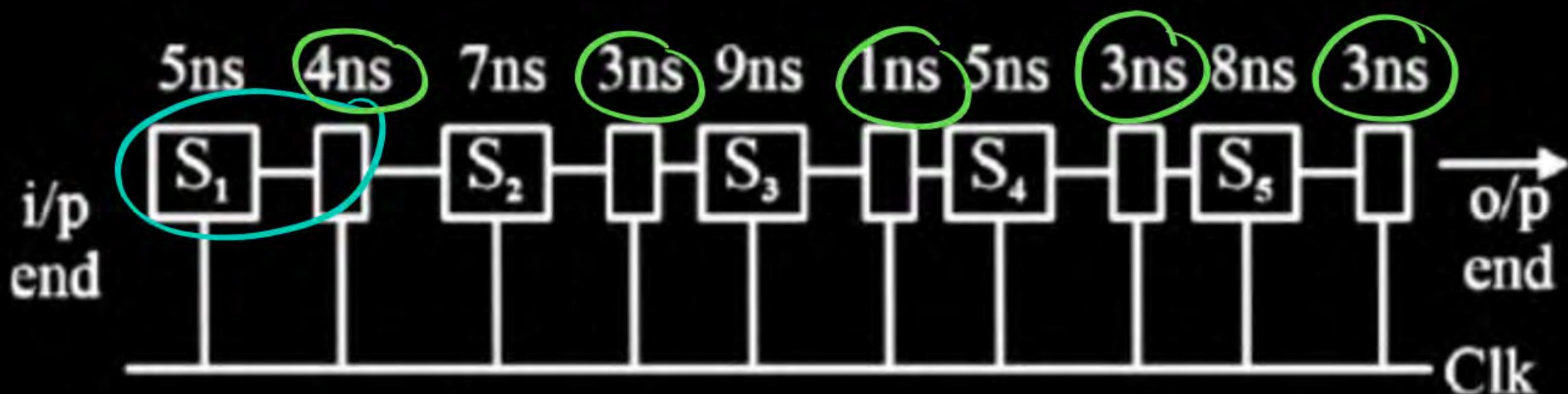
7 Consider the following Pipeline used to execute the program which contain 700 Instructions.



The execution time (in nano second) of the program is \_\_\_\_\_.

$$\begin{aligned}tp &= 9 + 4 \\&= 13 \text{ nsec}\end{aligned}$$

~~(x)~~ Consider the following Pipeline used to execute the program which contain 700 Instructions.



The execution time (in nano second) of the program

is 7744 Ans

$$t_p = \max(\text{Stage Delay} + \text{Buffer Delay})$$

$$= \max(5+4, 7+3, 9+1, 5+3, 8+3)$$

$$\max(9, 10, 10, 8, 11)$$

Ans (7744)

$$t_p = 1 \text{ msec}$$

$$k = 5$$

$$n = 700$$

$$ET = [k + (n-1)] t_p$$

$$\Rightarrow [5 + (700-1)] \times 1$$

$$= 704 \times 1$$

$$\Rightarrow 7744 \text{ msec } \underline{\text{Avg}}$$

#Q. 8 A pipelined processor uses a 4 stage instruction pipeline with the following stages: instruction fetch (IF) instruction decode (ID), execute (EX) and write back (WB). The arithmetic operation as well as the local and store operation are carried out in the EX-stage.

Consider the following sequence of instruction:

			Meaning
I <sub>1</sub> :	ADD	R <sub>4</sub> , R <sub>0</sub> , R <sub>1</sub> :	R <sub>4</sub> ← R <sub>0</sub> + R <sub>1</sub>
I <sub>2</sub> :	MUL	R <sub>5</sub> , R <sub>4</sub> , R <sub>2</sub> :	R <sub>5</sub> ← R <sub>4</sub> * R <sub>2</sub>
I <sub>3</sub> :	ADD	R <sub>7</sub> , R <sub>4</sub> , R <sub>5</sub> :	R <sub>7</sub> ← R <sub>4</sub> + R <sub>5</sub>
I <sub>4</sub> :	LOAD	R <sub>6</sub> , M(R <sub>5</sub> ):	R <sub>6</sub> ← M(R <sub>5</sub> )
I <sub>5</sub> :	MUL	R <sub>4</sub> , R <sub>7</sub> , R <sub>7</sub> :	R <sub>4</sub> ← R <sub>7</sub> * R <sub>7</sub>
I <sub>6</sub> :	STORE	M(R <sub>6</sub> ) R <sub>4</sub> :	M[R <sub>6</sub> ] ← R <sub>4</sub>
I <sub>7</sub> :	BNE	R <sub>4</sub> R <sub>6</sub> I <sub>2</sub> :	If (R <sub>4</sub> ≠ R <sub>6</sub> ) then I <sub>2</sub>

The number of read-after-write (RAW) dependencies in the sequence of instructions are 9 Ans

- #Q⑧ A pipelined processor uses a 4 stage instruction pipeline with the following stages: instruction fetch (IF) instruction decode (ID), execute (EX) and write back (WB). The arithmetic operation as well as the local and store operation are carried out in the EX-stage.

Consider the following sequence of instruction:

I <sub>1</sub> :	ADD	R <sub>4</sub> , R <sub>0</sub> , R <sub>1</sub> : R <sub>4</sub> ← R <sub>0</sub> + R <sub>1</sub>	Meaning:
I <sub>2</sub> :	MUL	R <sub>5</sub> , R <sub>4</sub> , R <sub>2</sub> : R <sub>5</sub> ← R <sub>4</sub> * R <sub>2</sub>	
I <sub>3</sub> :	ADD	R <sub>7</sub> , R <sub>4</sub> , R <sub>5</sub> : R <sub>7</sub> ← R <sub>4</sub> + R <sub>5</sub>	
I <sub>4</sub> :	LOAD	R <sub>6</sub> , M(R <sub>5</sub> ): R <sub>6</sub> ← M(R <sub>5</sub> )	
I <sub>5</sub> :	MUL	R <sub>4</sub> , R <sub>7</sub> , R <sub>2</sub> : R <sub>4</sub> ← R <sub>7</sub> * R <sub>2</sub>	
I <sub>6</sub> :	STORE	M(R <sub>6</sub> ) R <sub>4</sub> : M[R <sub>6</sub> ] ← R <sub>4</sub>	
I <sub>7</sub> :	BNE	R <sub>4</sub> R <sub>6</sub> I <sub>2</sub> : If (R <sub>4</sub> ≠ R <sub>6</sub> ) then I <sub>2</sub>	

- # RAW Dependency
- ① I<sub>2</sub> - I<sub>1</sub>
  - ② I<sub>3</sub> - I<sub>2</sub>
  - ③ I<sub>3</sub> - I<sub>1</sub>
  - ④ I<sub>4</sub> - I<sub>2</sub>
  - ⑤ I<sub>5</sub> - I<sub>3</sub>
  - ⑥ I<sub>6</sub> - I<sub>5</sub>
  - ⑦ I<sub>6</sub> - I<sub>4</sub>
  - ⑧ I<sub>7</sub> - I<sub>6</sub>
  - ⑨ I<sub>7</sub> - I<sub>5</sub>
- ⑨ Ans

The number of read-after-write (RAW) dependencies in the sequence of instructions are ⑨ Ans

Consider a pipeline with IF, ID and WB stages taking 1 clock cycle each. The EX stage takes 2 clock cycle for any arithmetic operation and 3 clock cycle for store operation. Operand forwarding from the Ex to ID stage is used for the below set of instruction sequence.

ADD,  $R_2 \leftarrow R_0, R_1$ ,       $R_2 \leftarrow R_0 + R_1$ ,

MUL,  $R_4 \leftarrow R_2, R_3$ ,       $R_4 \leftarrow R_2 \times R_3$

SUB,  $R_5 \leftarrow R_2, R_4$ ,       $R_5 \leftarrow R_2 - R_4$

STORE  $R_5, x$ , store the content of  $M[X]$  to register  $R_5$

The number of clock cycles required to complete the sequence of instruction is?

Q

[NAT]

⑩

H.W

P  
W

Consider a pipeline which is operating with a 1.8 GHz frequency contain 8 stages. Pipeline allow overlapping of all the instruction except branch instruction. Processor stop fetching of a sequential instruction after the branch instruction until the outcome is known. All the instruction output is available at the end of execution [Last Stage]. Program contain 40% branch instruction, among them 60% are conditional branch in which 40% instruction does not satisfy the condition (when condition is false) then the following instruction are overlapped then what is the average instruction execution time (in nano seconds) \_\_\_\_?  
(upto 2 decimal point)

②

①

The instruction pipeline of RISC processor has the following stages:

Instruction fetch (IF), Instruction Decode(ID), Operand Fetch (OF), Perform Operation (PO), and Write Back (WB). The IF, ID, OF and WB stage takes 1 clock cycle each for every instruction. Consider a sequence of 100 instruction. In the PO stage 50 instruction takes 6 clock cycle each, 30 instruction takes 5 clock cycle and 20 instruction takes 3 clock cycle. Assume that there is no Data Hazard and no control Hazard. Then number of clock cycle required for completion of execution of the sequence of instruction is 514 Avg

Avg(514)

Stage : CPI = 1

IF, ID, OF, PO, WB

$k=5, n=100$

$$ET_{PIPE} = [k + (n-1)] \text{ cycle}$$

$$\Rightarrow (\Sigma + 100 - 1)$$

$$= 104 \text{ cycle}$$

$ET_{Total} = ET_{Normal} + Stalls$
-------------------------------------

$$\Rightarrow 104 + 410$$

$$= 514 \text{ Ans}$$

PO

50 6 cycle

30 5

20 3

Extra Cycle

5 cycle

4 cycle

2 cycle

$$Stalls = 50 \times 5 + 30 \times 4 + 20 \times 2$$

$$\Rightarrow 250 + 120 + 40$$

$$\Rightarrow 410$$



Stage.

TF, TD, OF, PO, WB

PO

k=5, n=100.

50 6

30 5

20 3

TF TD OF WB + PO

$$\Rightarrow 1+1+1+1 + (50 \times 6 + 30 \times 5 + 20 \times 3)$$

$$\Rightarrow 4 + (300 + 150 + 60)$$

$$\Rightarrow 4 + 510$$

$$= 514 \text{ msec } \underline{\text{Avg}}$$

Consider a hypothetical processor with five pipeline stages (IF, ID, EX, MEM, WB) perfectly balanced & clock cycle time 11 nsec. With the following branch frequencies:

Jump and calls-25%

Conditional branches -30%

Taken conditional branch - 60%

Un-conditional & conditional branches are resolved at the end of fifth and sixth stage respectively processor always executes the branch successor regardless of target and flushes the pipeline if branch is taken. What is the throughput (in million instructions per second) of the system?

③

A

C

B

D

We have two design  $D_1$  and  $D_2$  for a synchronous pipeline processor.  $D_1$  has 6 pipeline stages with execution times of 5ns, 3ns, 6ns, 4ns, 5ns, and 2ns. While the design  $D_2$  has 8 pipeline stage each with 3ns execution time. How much time can be saved using design  $D_2$  over design  $D_1$  for executing 200 instructions (609) Ans

$$\underline{D_1 \Rightarrow k=6}$$

$$t_p = 6 \text{ ns} \quad n = 200$$

$$ET_{D_1} = [k + (n-1)] t_p$$

$$= [6 + (200-1)] \times 6$$

$$= 205 \times 6$$

$$= 1230 \text{ ns}$$

Design  $D_2$

$$k=8, \quad t_p=3 \text{ nsec}, \quad n=200$$

$$ET_{D_2} = [k + (n-1)] * t_p$$

$$= [8 + (200-1)] \times 3$$

$$= 207 \times 3$$

$$= 621 \text{ ns}$$

1230  
621

(609) Ans

Q Consider a 5-stage pipeline processor. The number of cycles needed by four instructions  $I_1, I_2, I_3, I_4$  in stage  $S_1, S_2, S_3, S_4$  and  $S_5$  is shown below:

	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$
$I_1$	2	2	1	2	3
$I_2$	1	1	2	1	2
$I_3$	1	3	1	1	1
$I_4$	1	1	1	1	1

What is the number of cycles needed to execute instruction  $i=1$  completely for first iteration, for the below loop?

for( $i = 1$  to 2)

{

$I_1;$

$I_2;$

$I_3;$

$I_4;$

};

[NAT]

P  
W

Consider a hypothetical 6 stage pipeline processor.  
 Let P is the probability of an instruction being a branch. What must be value of p such that speed up factor is at least 5. Also assume each stage takes 1 cycle to perform its task and branch predicted on fifth stage of the pipeline is 0.05 (upto 2 decimal places)

Ans (0.05)

$$B_P = 5 - 1 = 4 \quad B \cdot P = P$$

$$\Rightarrow \frac{6}{1+4P} \geq 5$$

$$S = \frac{\text{PIPELINE Depth} (\# \text{Stages})}{(1 + \# \text{Stalls} / T_{inst})}$$

$$\frac{\# \text{Stages}}{(1 + (BDF \times B_P))} = \frac{6}{1 + P \times 4} \geq 5$$

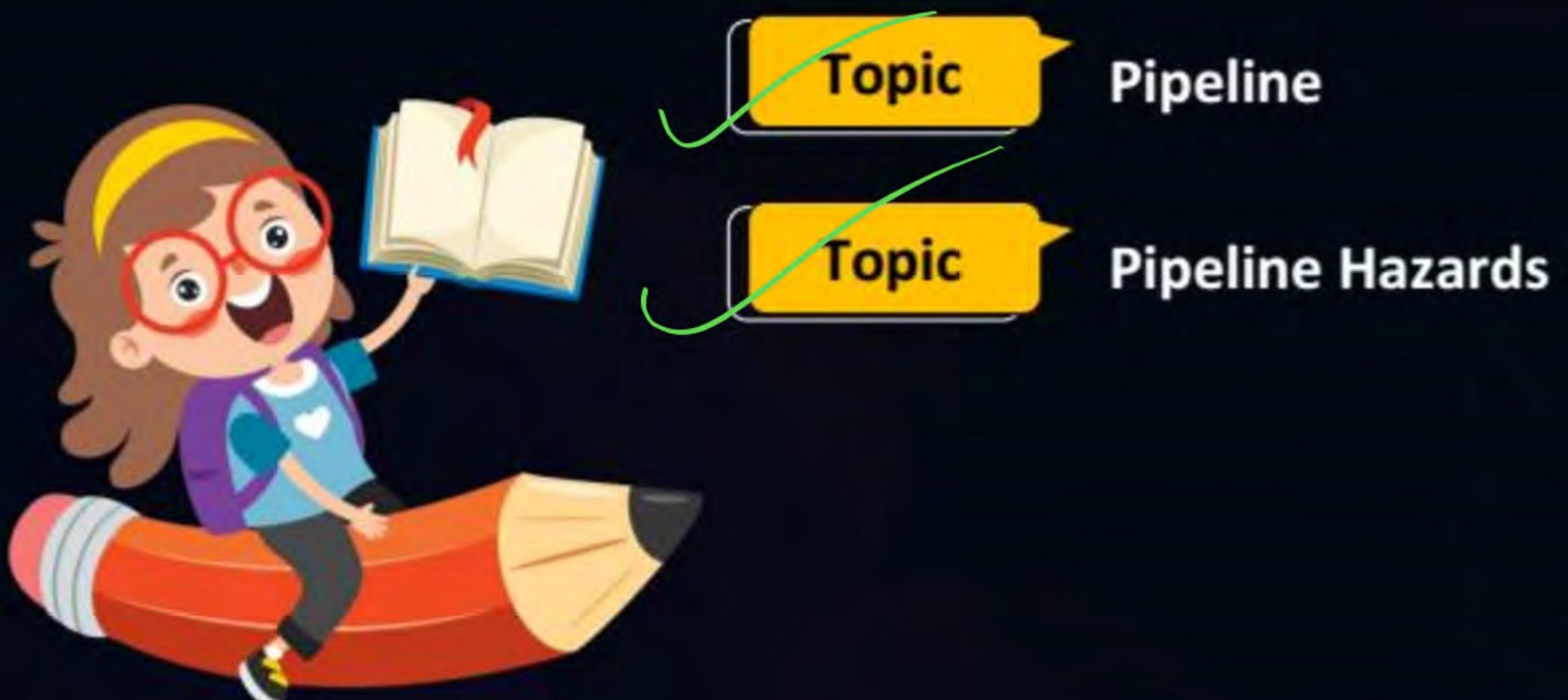
$$\Rightarrow 6 > 5 + 20P$$

$$1 = 20P$$

$$P = \frac{1}{20} = 0.05 \quad \underline{\text{Ans}}$$



## 2 mins Summary





THANK - YOU

# CS & IT ENGINEERING

Computer Organization  
& Architecture

1500 Series

Lecture No.- 05

By- Vijay Agarwal Sir





# Recap of Previous Lecture



Topic

Expand Opcode Technique

Topic

ALU Data Path

Topic

Micro Operation & Micro Program

Topic

Control Unit



# Topics to be Covered



Topic

Pipeline

Topic

Pipeline Hazards

#Q. An 7-stages perfectly balanced instruction pipeline has 2-cycle-time overhead. If 40% of the instruction in 4 pipeline stall cycles, the speedup achieved with respect to non-piped execution when an application is executing on this 7-stage pipeline is \_\_\_\_\_

$$BTF = 40\%$$

$$\# \text{stalls} = 4$$

$$\begin{aligned}\# \text{stalls}/\text{Inst} &= .40 \times 4 \\ &= 1.6\end{aligned}$$

$$\begin{aligned}S &= \frac{7}{(1+1.6)} \\ &= \frac{7}{2.6} = 2.692 \text{ Avg}\end{aligned}$$

Consider the pipeline processor have instruction fetch, instruction decode, execute, memory and write back have the latencies  $300\text{ps}$ ,  $180\text{ps}$ ,  $170\text{ps}$ ,  $200\text{ps}$  and  $140\text{ps}$  respectively & Each pipeline have registers between the stages having delay of  $20\text{ps}$ . For the enhancement purpose the largest pipeline stages are split into 2 equal delays. The new cycle time value is  $x$  (in pico sec.) & the latency of a instruction in a new pipeline is  $y$  (in pico sec). The value of  $x + y$  is 1540 (in pico sec.).

Ans (1540)

OLD Design  
5 Stage

300 180 170 200 140

New Design

6 Stages

150 150 180 170 200 140

Buffer Delay = 20

$$t_{\text{new}} = (200 + 20) = 220 \text{ nsec}$$

Latency of One Instn =  $6 \times 220$

$$y = 1320 \text{ ns}$$

$$x = 220 \text{ nsec}$$

$$\begin{aligned} x + y &= 220 + 1320 \\ &= 1540 \text{ nsec} \end{aligned}$$

Consider a 4 stage pipeline with IF, ID and WB stages taking 1 clock cycle each. The EX stage takes 2 clock cycle for any arithmetic operation and 4 clock cycle for store operation. Operand forwarding from the Ex to ID stage is used for the below set of instruction sequence.

1	ADD, $R_2 \leftarrow R_0, R_1$ , $R_2 \leftarrow R_0 + R_1$ ,
2	MUL, $R_4 \leftarrow R_2, R_3$ , $R_4 \leftarrow R_2 \times R_3$
3	SUB, $R_5 \leftarrow R_2, R_4$ , $R_5 \leftarrow R_2 - R_4$
4	STORE $R_5, x$ , store the content of M [X] to register $R_5$

Q3 Ans

The number of clock cycles required to complete the sequence of instruction is?

Ans(13)



Consider a 5-stage pipeline processor. The number of cycles needed by four instructions  $I_1, I_2, I_3, I_4$  in stage  $S_1, S_2, S_3, S_4$  and  $S_5$  is shown below:

	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$
$I_1$	2	2	1	2	3
$I_2$	1	1	2	1	2
$I_3$	1	3	1	1	1
$I_4$	1	1	1	1	1

What is the number of cycles needed to execute instruction  $i=1$  completely for first iteration, for the below loop?

```
for(i = 1 to 2)
```

```
{
```

```
    I1;
```

```
    I2;
```

```
    I3;
```

```
    I4;
```

```
};
```

P  
W

	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$
$T_1$	2	2	1	2	3
$T_2$	1	1	2	1	2
$T_3$	1	3	1	1	1
$T_4$	1	1	1	1	1

$s_5$		$\Theta$	$I_1$	$I_1$	$I_1$	$I_2$	$I_2$	$I_2$	$I_3$	$I_3$	$I_4$	
$s_4$			$I_1$	$I_1$	$I_2$	$\Theta$	$\Theta$	$\Theta$	$I_3$	$\Theta$	$I_4$	
$s_3$				$I_1$	$I_2$	$I_2$		$I_3$	$\Theta$	$I_4$		
$s_2$					$I_1$	$I_2$	$I_3$	$I_3$	$I_3$	$I_3$	$I_4$	
$s_1$	1	2	3	4	$I_1$	$I_1$	$I_2$	$\Theta$	$I_3$	$I_4$	$\Theta$	(14)

	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$
$T_1$	2	2	1	2	3
$T_2$	1	1	2	1	2
$T_3$	1	3	1	1	1
$T_4$	1	1	1	1	1

Assume Data is stored in Buffer.



## [MCQ]

Consider a hypothetical processor with seven pipeline stages (IF, ID, EX, MEM, WB) perfectly balanced & clock cycle time 11 nsec. With the following branch frequencies:

Jump and calls - 25% - 4  
Conditional branches - 30% - 5  
Taken conditional branch - 60% - 5

$$\begin{aligned}\# \text{Stalls/Dinst} &= .25 \times 4 + .30 \times .60 \times 5 \\ &= 1 + .9 \\ &= 1.9\end{aligned}$$

Un-conditional & conditional branches are resolved at the end of fifth and sixth stage respectively processor always executes the branch successor regardless of target and flushes the pipeline if branch is taken. What is the throughput (in million instructions per second) of the system?

- A 34.4
- C 29.2

- B 31.3
- D None of these

$$\text{Unconditional} = B.P = 5 - 1 = 4$$

$$\text{Conditional} = G - 1 = 5$$

$$\text{Avg Instn} = (1 + \# \text{Stalled Instn}) \times \text{cycle time}$$

$$\Rightarrow (1 + 1.9) \times 11 \Rightarrow 2.9 \times 11 \text{ nsec}$$
$$= 31.9 \text{ nsec}$$

$$\text{Throughput} = \frac{1}{31.9 \times 10^{-9}} \Rightarrow \frac{1}{31.9} \times 10^9 \Rightarrow \frac{1000 \times 10^6}{31.9}$$

$$= 31.3 \times 10^6$$

$$= 31.3 \text{ MIPS.}$$

Consider two processors  $P_1$  and  $P_2$  executing the same instructions set. Assume that under identical conditions, for the same input, a program running on  $P_2$  takes 25% less time but incurs 20% more CPI (clock cycles per instruction) as compared to the program running on  $P_1$ . If the clock frequency of  $P_1$  is 1GHz, then the clock frequency of  $P_2$  (in GHz) is \_\_\_\_\_.

**NAT**

Consider a non-pipelined processor with a clock rate of 4 gigahertz and average cycles per instruction of six. The same processor is upgraded to a pipelined processor with five stages; but due to the internal pipeline delay, the clock speed is reduced to 2.5 gigahertz. Assume that there are no stalls in the pipeline. The speed up achieved in this pipelined processor is ~~3.75~~ Ans

### Non Pipeline Processor

$$\text{Cycle time} = \frac{1}{4\text{GHz}} = 0.25\text{nsec}$$

1 Instruction takes = 6 cycle.

### In Nonpipeline

$$\begin{aligned} ET &= 0.25 \times 6 \\ &= 1.5\text{nsec} \end{aligned}$$

### Pipeline Processor

$$CPI = 1$$

$$\begin{aligned} \text{Cycle time} &= \frac{1}{2.5\text{GHz}} = 0.4\text{nsec.} \\ ET_{PIPE} &= 0.4\text{nsec} \end{aligned}$$

$$S = \frac{ET_{NP}}{ET_p}$$

$$= \frac{1.5}{0.4} = 3.75 \text{ Avg}$$

**MCQ**

Delayed branching can help in the handling of control hazards.

The following code is to run on a pipelined processor with one branch delay slot:

I1: ADD R2  $\leftarrow$  R7 + R8  
I2: SUB R4  $\leftarrow$  R5 - R6x  
I3: ADD R1  $\leftarrow$  R2 + R3  
I4: STORE Memory [R4]  $\leftarrow$  R1

BRANCH to Label if /R1 == 0

Which of the instruction I1, I2, I3 or I4 can legitimately occupy the delay slot without any other program modification?

A I1

B I2

C I3

D I4

$\text{D: ADD } R_2 \leftarrow R_1 + R_8.$

$I_2: \text{SUB } R_4 \leftarrow R_5 - R_6.$

$I_3: \text{ADD } R_1 \leftarrow R_2 + R_3$

$I_4: \text{STORE } M[R_4] \leftarrow R_1$

Branch to label if  $R_1 = 0$ .

⑥  $I_2$

$R_u \leftarrow R_5 - R_6.$

↳ Used in  $I_4$  for  $M[R_4]$  storing.

⑦  $I_3: R_1 \leftarrow R_2 + R_3$

↳ Used in  $I_4$  Not working  
if we used Dern-Pooleby

$R_2 \leftarrow R_1 + R_8.$

↳ Used in  $I_3$ . (Result of  
 $I_1$  as operand  
using in  $I_3$ )

⑧  $I_4$

$M[R_4] \leftarrow R_1$

$M[R_4] \leftarrow R_1$

⑨ ~~X~~

Assume a processor that has 5-segment in order-pipelining

Instruction	Meaning of instruction
I <sub>0</sub> : X: LOAD R <sub>2</sub> , 8(R <sub>3</sub> )	//R <sub>2</sub> ← MEM [8 + R <sub>3</sub> ]
I <sub>1</sub> : ADD R <sub>4</sub> , R <sub>2</sub> , R <sub>9</sub>	// R <sub>4</sub> ← R <sub>2</sub> + R <sub>9</sub>
I <sub>2</sub> : SUB R <sub>3</sub> , R <sub>2</sub> , R <sub>5</sub>	// R <sub>3</sub> ← R <sub>2</sub> - R <sub>5</sub>
I <sub>3</sub> : STORE R <sub>8</sub> , 12(R <sub>3</sub> )	//MEM [12 + R <sub>3</sub> ] = R <sub>8</sub>
I <sub>4</sub> : ADD R <sub>7</sub> , R <sub>3</sub> , R <sub>6</sub>	// R <sub>7</sub> ← R <sub>3</sub> + R <sub>6</sub>
I <sub>5</sub> : BEQ R <sub>7</sub> , R <sub>8</sub> , X	//If R <sub>7</sub> == R <sub>8</sub> then jump to X

Which instruction(s) in the above assembly sequence would you place in the delay slot? Assume that the number of available delay slots is 2?

T<sub>1</sub>, T<sub>3</sub>.

#Q. A pipelined processor uses a 5 stage instruction pipeline with the following stages: **instruction fetch (IF)** **instruction decode & Operand Fetch (ID), Execute(EX), Memory Access(MA) and write back (WB)**. Consider the following sequence of instruction:

Instruction	Meaning of instruction
I <sub>1</sub> : ADD R <sub>0</sub> , R <sub>1</sub> , R <sub>2</sub>	//R <sub>0</sub> ← R <sub>1</sub> + R <sub>2</sub>
I <sub>2</sub> : LOAD R <sub>3</sub> 100(R <sub>0</sub> )	//R <sub>3</sub> ← MEM [100 + R <sub>0</sub> ]
I <sub>3</sub> : ADDI R <sub>4</sub> , R <sub>3</sub> , 20	//R <sub>4</sub> ← R <sub>3</sub> + 20
I <sub>4</sub> : ADD R <sub>3</sub> , R <sub>2</sub> , R <sub>1</sub>	//R <sub>3</sub> ← R <sub>2</sub> + R <sub>1</sub>
I <sub>5</sub> : STORE R <sub>3</sub> 100(R <sub>0</sub> )	//MEM [100 + R <sub>0</sub> ] ← R <sub>3</sub>

The number of Clock cycle Required in the sequence of instructions  
 Operand forwarding 10 Cycles

Using

$I_2$ : LOAD  $\Rightarrow$  M1 Stage.



WB

MA

EX

DD

DF

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

Now Change in Question  
& Change in Stage working  
without Operand forwarding.

#Q. A pipelined processor uses a 5 stage instruction pipeline with the following stages: **instruction fetch (IF)** **instruction decode (ID)**, **Operand Fetch (OF)** **Perform operation (PU)** & **Memory Access and write back (WB)**.

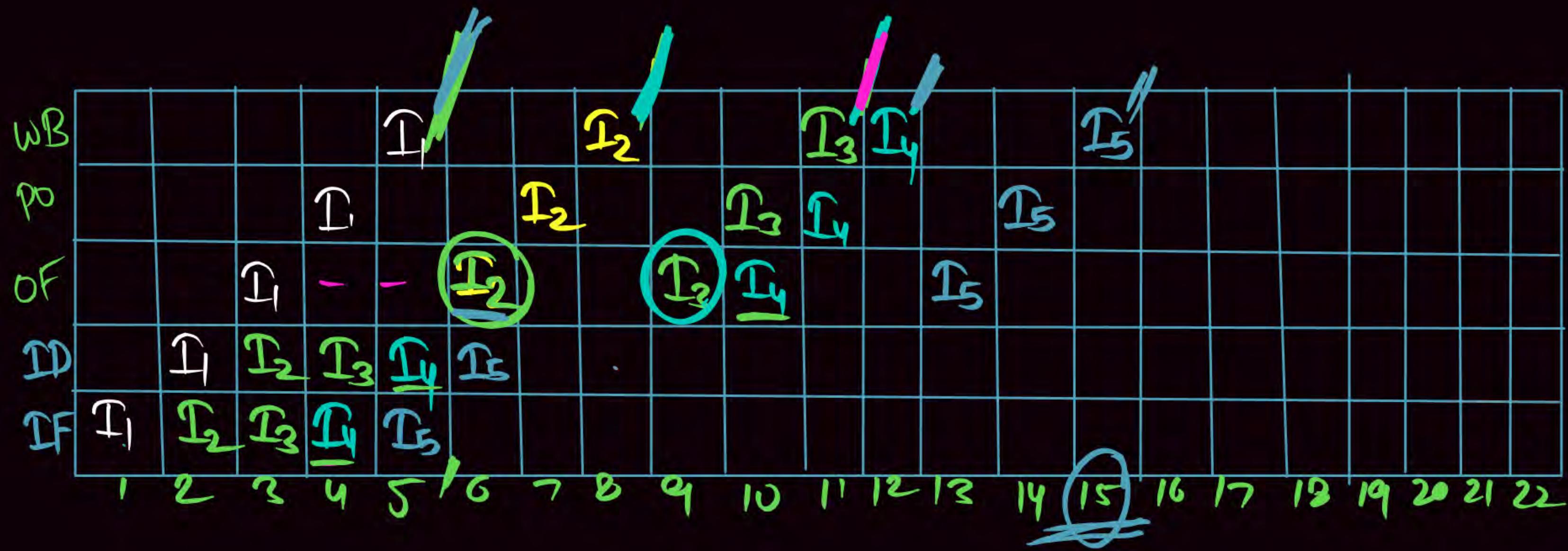
Consider the following sequence of instruction:

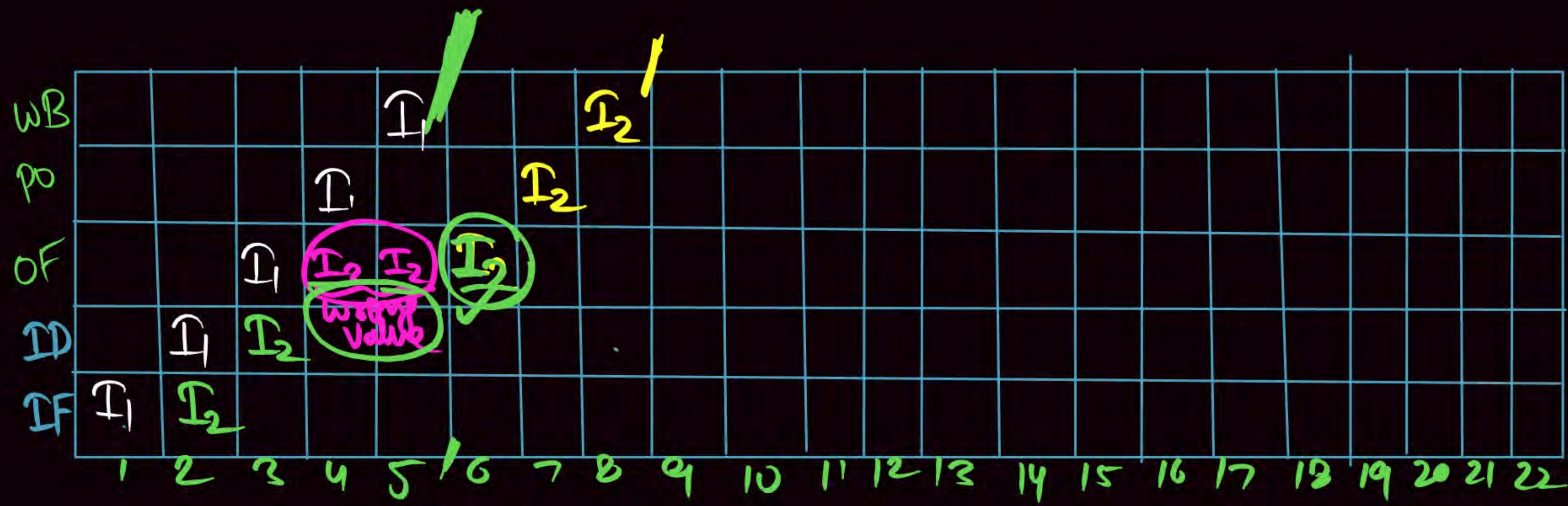
Instruction	Meaning of instruction
I <sub>1</sub> : ADD R <sub>0</sub> , R <sub>1</sub> , R <sub>2</sub>	// R <sub>0</sub> ← R <sub>1</sub> + R <sub>2</sub>
I <sub>2</sub> : LOAD R <sub>3</sub> 100(R <sub>0</sub> )	// R <sub>3</sub> ← MEM [100 + R <sub>0</sub> ]
I <sub>3</sub> : ADDI R <sub>4</sub> , R <sub>3</sub> , 20	// R <sub>4</sub> ← R <sub>3</sub> + 20
I <sub>4</sub> : ADD R <sub>3</sub> , R <sub>2</sub> , R <sub>1</sub>	// R <sub>3</sub> ← R <sub>2</sub> + R <sub>1</sub>
I <sub>5</sub> : STORE R <sub>3</sub> 100(R <sub>0</sub> )	// MEM [100 + R <sub>0</sub> ] ← R <sub>3</sub>

Not Depend

Without The number of Clock cycle Required in the sequence of instructions  
Operand forwarding \_\_\_\_.

Using





Which of the following are NOT true in a pipelined processor?

1. Bypassing can handle all RAW hazards. → False (Load & Store (MA Stage))
2. Register renaming can eliminate all register carried WAR hazards.

A 1 only

C 1 and 2 Both

- B 2 only
- D None of these

Anti  
dep } ⇒ Register  
Renaming

An instruction pipeline has five stages, namely, instruction fetch (IF), instruction decode and register fetch (ID/RF), instruction execution (EX), memory access (MEM), and register write back (WB) with stage latencies 1 ns, 2.2 ns, 2ns ,1 ns, and 0.75 ns, respectively (ns stands for nanoseconds). To gain in terms of frequency, the designers have decided to split the ID/RF stage into three stages (ID, RF1, RF2) each of latency 2.2/3 ns. Also, the EX stage is split into two stages (EX1, EX2) each of latency 1 ns. The new design has a total of eight pipeline stages. A program has 20% branch instructions which execute in the EX stage and produce the next instruction pointer at the end of the EX stage in the old design and at the end of the EX2 stage in the new design. The IF stage stalls after fetching a branch instruction until the next instruction pointer is computed. All instructions other than the branch instruction have an average CPI of one in both the designs. The execution times of this program on the old and the new design are P and Q nanoseconds, respectively. The value of P/Q is \_\_\_\_.

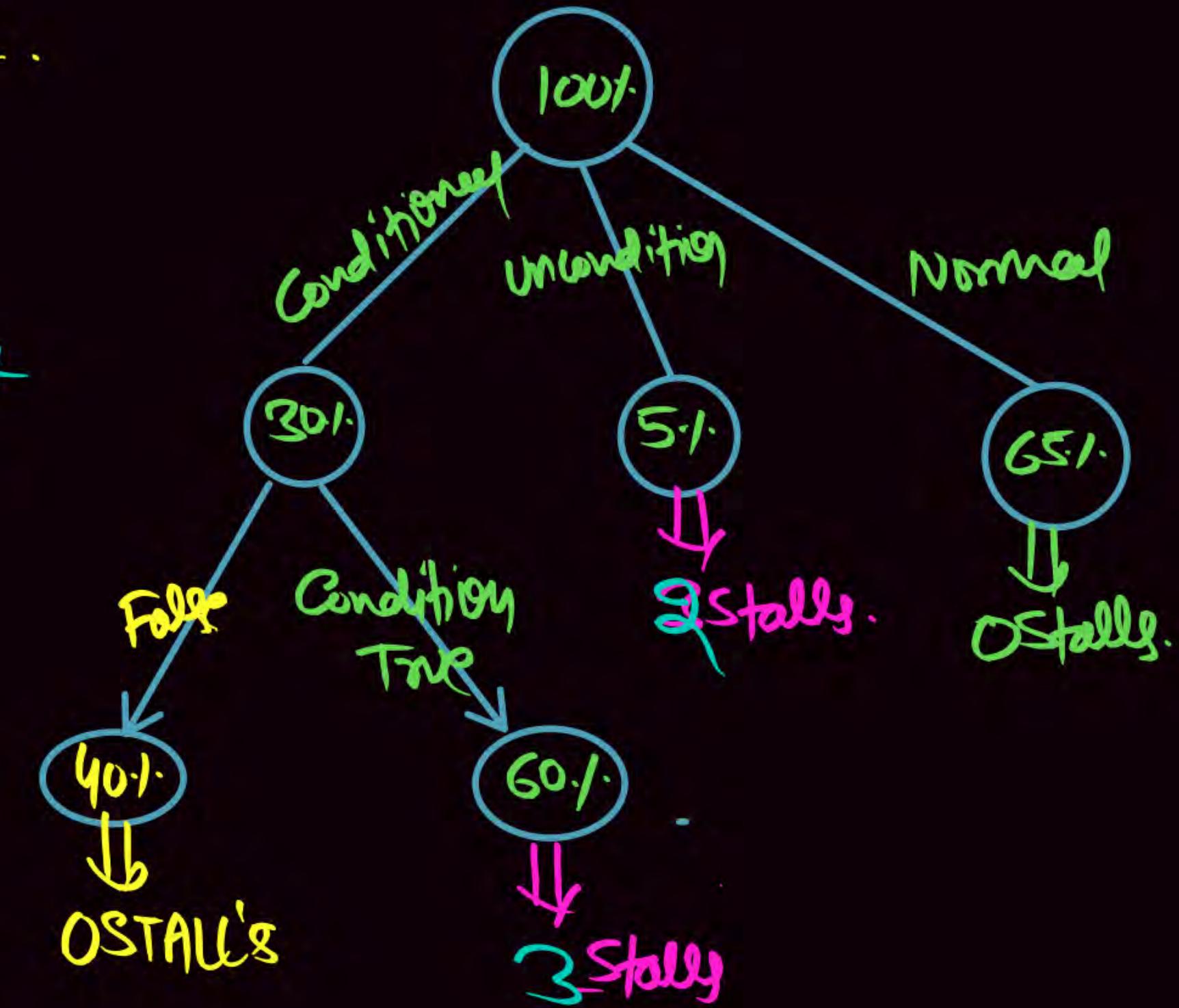
Consider a Pipelined processor Operating at 2.5GHZ. Pipeline have 6-segment pipeline, IF, ID, EX1, EX2, MA, and WB. The branch is resolved at the end of 3<sup>rd</sup> cycle for unconditional branches and at the end of 4<sup>th</sup> <sup>stage</sup> cycle for conditional branches. Assume 30% of all instructions are conditional branches out of which 60% are taken and 5% are unconditional branches or procedure calls. Assume that only the branches instructions result in stall. then what is Average instruction Execution time of this processor? (in ~~Micro~~ <sup>Nano</sup> Second)

$$\text{Uncondition} \Rightarrow \text{Stall} = 3 - 1 = 2.$$

$$\text{Conditional} = 4 - 1 = 3$$

$$\begin{aligned}\#\text{Stalls/TInstruction} &= .30 \times .40 \times 0 + \\ &.30 \times .60 \times 3 + \\ &.05 \times 2 + \\ &.65 \times 0 \\ &= 0.64.\end{aligned}$$

$$\text{Cycle time} = \frac{1}{2.54} \text{ sec} = \underline{\underline{0.4 \text{ msec}}}$$



$$\text{Avg Dustn ET} = (1 + \# \text{Stalls}/\text{inst}) \times \text{Cycle time.}$$

$$\Rightarrow (1 + 0.64) \times 0.4 \text{ nsec}$$

$$\Rightarrow 1.64 \times 0.4 \text{ nsec}$$

$$\Rightarrow 0.656 \text{ nsec}$$

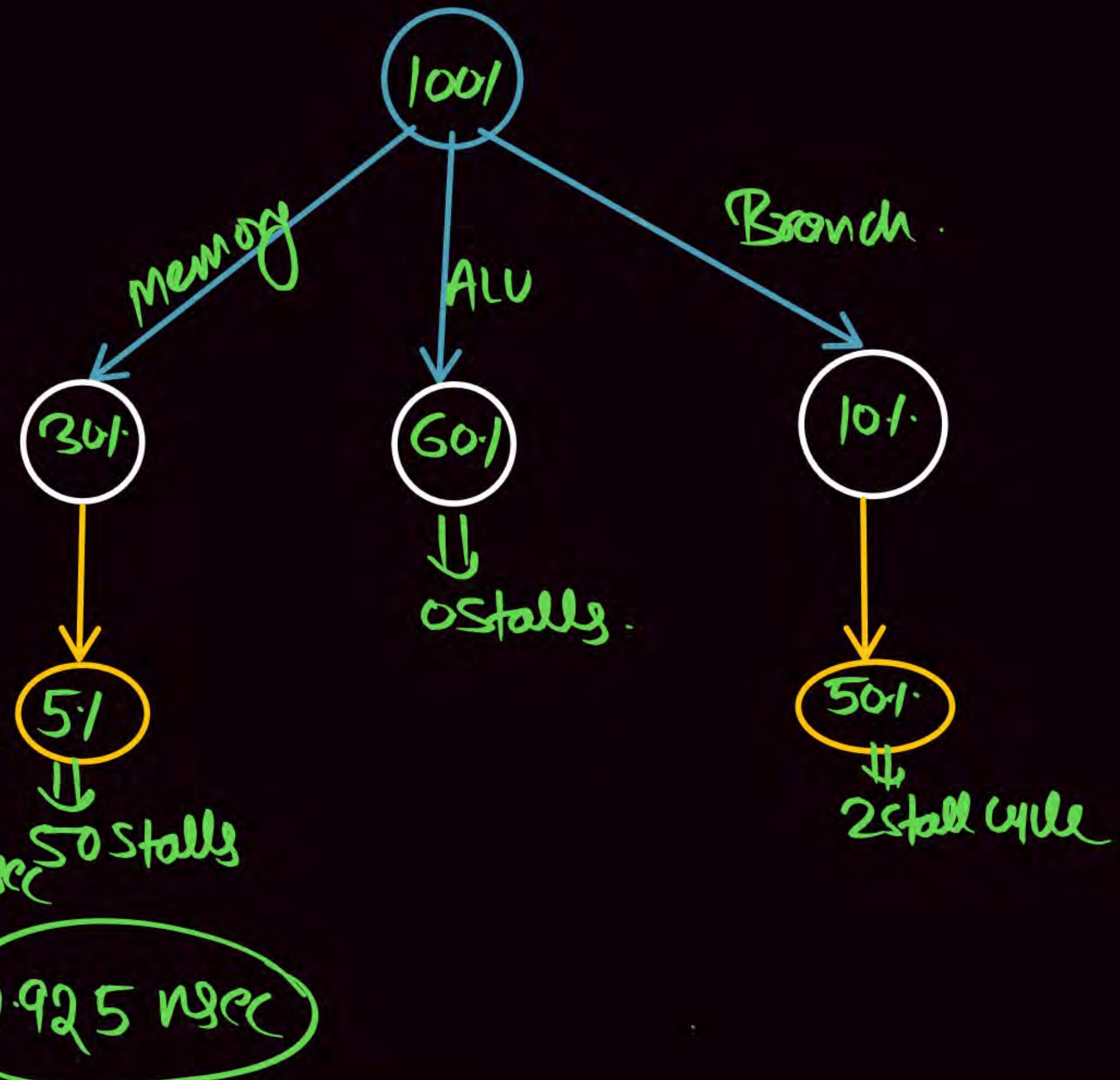
Consider a non-pipelined processor operating at 4 GHz. It takes 7 clock cycles to complete an instruction. You are going to make a 6-stage pipeline out of this processor. Overheads associated with pipelining force you to operate the pipelined processor at 2 GHz. In a given program, assume that 30% are memory instructions, 60% are ALU instructions and the rest are branch instructions. 5% of the memory instructions cause stalls of 50 clock cycles each due to cache misses and 50% of the branch instructions cause stalls of 2 cycles each. Assume that there are no stalls associated with the execution of ALU instructions. For this program, the speedup achieved by the pipelined processor over the non-pipelined processor (round off to 2 decimal places) is \_\_\_\_\_

$$\# \text{Stalls} / T_{\text{instr}} =$$

$$\begin{aligned} & \cdot 30 \times 0.05 \times 50 + \\ & \cdot 60 \times 0 + \\ & \cdot 10 \times 50 \times 2 \end{aligned}$$

$$= 0.85$$

$$\begin{aligned} \text{Avg Cycles} &= \left(1 + \frac{\# \text{Stall}}{T_{\text{instr}}}\right) \text{cycles} \\ &\Rightarrow \left(1 + 0.85\right) \times 1 \text{ cycle} \quad \text{50 stalls} \\ &= \frac{1.85}{2} \text{ cycles} = 0.925 \text{ cycles} \end{aligned}$$



$$\text{Non Pipeline} = 4 \text{ GHz}$$

$$\text{Cycle time} = \frac{1}{4} = 0.25 \text{ nsec}$$

7 cycle

$$CPL \times \text{cycle time}$$

$$\begin{aligned}\text{Non Pipeline} &= 7 \times 0.25 \\ &= 1.75 \text{ nsec}\end{aligned}$$

$$S = \frac{ET_{NP}}{ET_{PIPE}}$$

$$= \frac{1.75}{0.925}$$

$$S = 1.89 \text{ Ans}$$

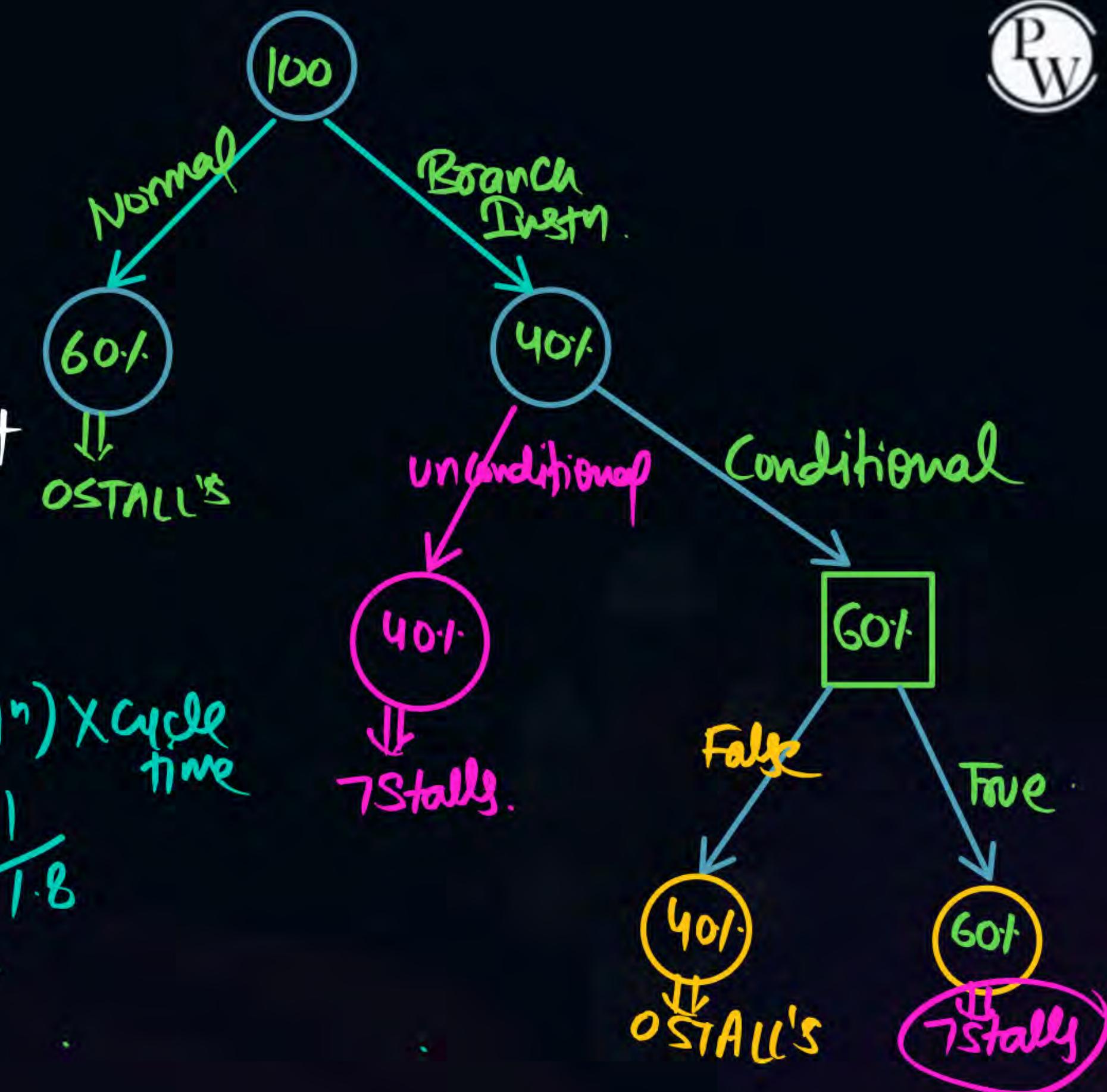
Consider a pipeline which is operating with a 1.8 GHz frequency contain 8 stages. Pipeline allow overlapping of all the instruction except branch instruction. Processor stop fetching of a sequential instruction after the branch instruction until the outcome is known. All the instruction output is available at the end of execution [Last Stage]. Program contain 40% branch instruction, among them 60% are conditional branch in which 40% instruction does not satisfy the condition (when condition is false) then the following instruction are overlapped then what is the average instruction execution time (in nano seconds) \_\_\_\_?  
(upto 2 decimal point)

$$B.P = 8 - 1 = 7 \text{ stalls}$$

$$\begin{aligned} \# \text{Stalls/Instn} &= 0.60 \times 0 + \\ &+ 0.40 \times 0.40 \times 7 + \\ &+ 0.40 \times 0.60 \times 0.40 \times 0 + \\ &+ 0.40 \times 0.60 \times 0.60 \times 7 \end{aligned}$$

$$\boxed{\# \text{Stalls/Instn} \approx 2.128}$$

$$\begin{aligned} \text{Avg Instn ET} &= (1 + \# \text{Stalls/Instn}) \times \text{Cycle time} \\ &\Rightarrow (1 + 2.128) \times \frac{1}{1.8} \\ &= 1.73 \text{ nsec.} \end{aligned}$$



The width of the physical address on a machine is 32 bits, 4 way set associative cache each block hold 2k byte (2 kilo byte) & having word of size 4 bytes. There are 512 blocks in the cache. Initially cache is empty then how many data words are brought into the cache is \_\_\_\_\_.

[NAT] *H.W.*

P  
W

If a 4-way set associative cache is made up of 64 bit words. 8 words per line and 8192 sets. What is the size of cache memory (in MB(Mega Bytes) byte) \_\_\_\_\_



## 2 mins Summary



Topic

Pipeline

Topic

Pipeline Hazards



THANK - YOU

# CS & IT ENGINEERING

## Computer Organization & Architecture

1500 Series

Lecture No.- 06

By- Vijay Agarwal Sir





# Recap of Previous Lecture



Topic

Pipeline

Topic

Pipeline Hazards



# Topics to be Covered



Topic

Cache Memory

Topic

Cache mapping Technique

Topic

Cache Replacement Algorithm

Topic

Cache Updating Technique

#Q. Consider the main memory of a computer system consisting of " $2^m$ " blocks, and cache " $2^m$ " blocks, and cache has " $2^m$ " blocks. If the two-way set associative mapping scheme is used, the  $k^{\text{th}}$  block of main memory to be placed in \_\_\_\_ set?

A

$$2^m \bmod 2^m$$

C

$$2^m \bmod m$$

Ans (D)

B

$$k \bmod 2^m$$

D

$$k \bmod m$$

$$K \bmod S = i$$

k: MM Block No.

S: #SET

i: set Number

$$K \bmod M = i$$

$$\# \text{MemBlock} (\# \text{CM Lines}) = 2^M$$

2 way set Associ.

$$\# \text{SET (S)} = \frac{2^M}{2} = M$$

#Q. Number of chips ( $128 \times 8$  RAM) needed to provide a memory capacity of 2048 bytes is \_\_\_\_\_

$$\text{Memory Size} = 2048 \text{ Byte} \Rightarrow 2048 \times 8 \text{ bit}$$

$$\text{RAM CHIP Size} = 128 \times 8$$

$$\# \text{RAM CHIP} = \frac{(2^4) 2048 \times 8}{(2^3) 128 \times 8} \Rightarrow 2^4 = 16 \text{ RAM CHIP Ans}$$

Ans (16)

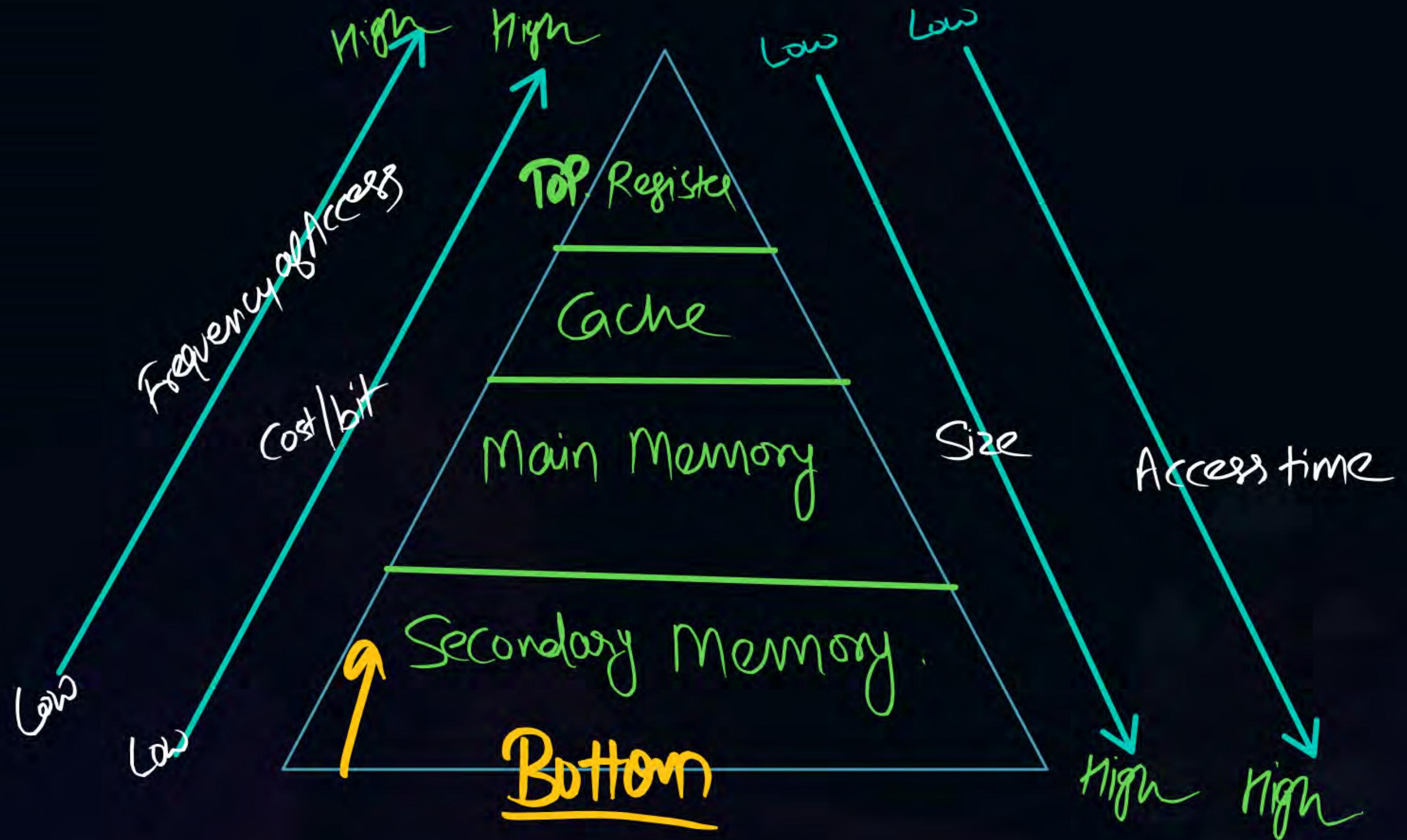
#Q. Match List-I and List-II using memory hierarchy from bottom to top and find the best suitable matching for List-I.

List-I		List-II	
a.	size — 2	1.	Increase
b.	Cost/bit — 1	2.	Decrease
c.	Access time — 2	3.	No change
d.	Frequency of access — 1		

Arg(D)

	a	b	c	d
A	1	1	2	2
	a	b	c	d
C	2	2	1	1

	a	b	c	d
B	1	2	1	2
D	2	1	2	1



#Q. How many  $128 \times 8$  bit RAMs are required to design  $32K \times 32$  bit RAM

- |   |     |   |      |
|---|-----|---|------|
| A | 512 | E | 1024 |
| C | 128 | D | 32   |
- (Ans(B))**

$$\frac{2^8 \times 32K \times 32}{(2^3) 128 \times 8} \rightarrow 2^8 \times 4 = 2^{10}$$

1024

H.W

#Q. Find the correct order of True (T) and False (F) for the following statements.

1. DRAM consists of internal flip-flops.
2. SRAM consists of capacitors.
3. SRAM is easier to use than DRAM.
4. SRAM has shorter read and write cycles than that of DRAM.

A

F F T F

B

T T T T

C

F F T T

D

T T F F

#Q. Construct  $128K \times 16$  bit RAM from  $128 \times 16$  bit RAM chip. What is the size of decoder is required ?

A  $9 \times 512$

Ans(c)

C  $10 \times 1024$

n bit

$n \times 2^n$

$10 \times 2^{10}$

$10 \times 1024$

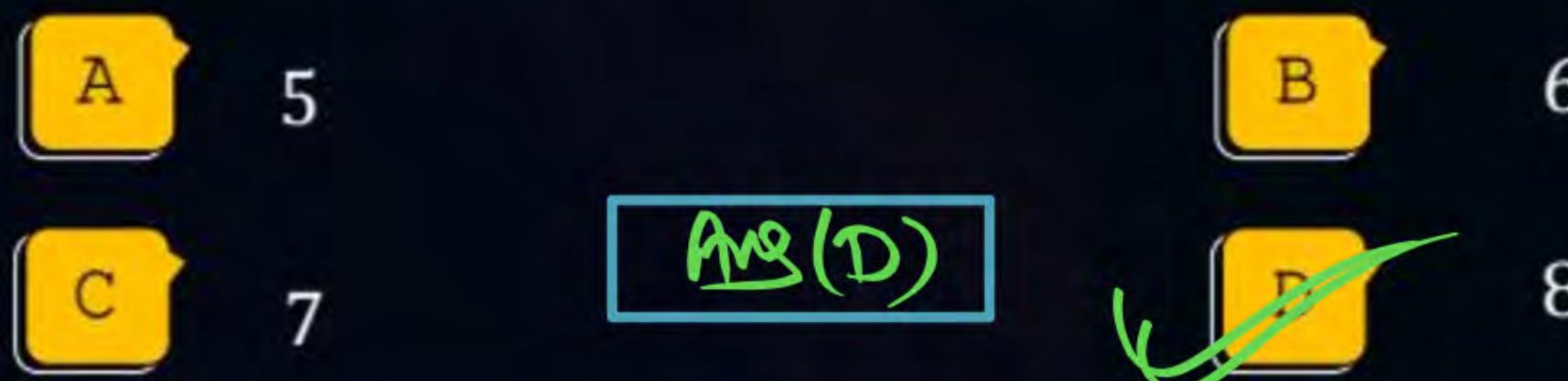
B  $11 \times 2048$

D  $7 \times 128$

$$\frac{128K \times 16}{128 \times 16} = 2^{10} (1K)$$

#Q. Consider a 2 way set associative cache, consisting of 4 blocks. Assume that LRU block replacement policy is used. The number of cache misses for the following sequence of block address is

10 3 5 7 3 7 12 24 10 3



$$\begin{aligned}\# \text{SET} &= \frac{\# \text{LINES}}{N\text{-way}} \\ &= \frac{4}{2} = 2\end{aligned}$$



SET 0	10 24 12 10
SET 1	3 7 5 3

#Q. If a Cache Memory is designed as a direct mapped cache, the length of the TAG field is 10 bits. If the cache is now designed as 8-way set associative cache, the length of the TAG is 13 bits. Ans

$$\text{Tag bit in Set Associative} = \text{Tag bits in Direct Mapping} + \lceil \log_2 N \rceil$$

$$\begin{aligned}\text{Tag bit in 8Way Set Associative} &= 10 + \lceil \log_2 8 \rceil \\ &\Rightarrow 10 + 3 \\ &\Rightarrow 13 \text{ bits } \underline{\text{Ans}}\end{aligned}$$

$$\text{Direct mapping} \Rightarrow \text{Tag} = \frac{\text{MM Size}}{\text{CM Size}}$$



$$\text{Tag bit} = \lceil \log_2 \text{tags} \rceil$$

#Q. In memory hierarchy of cache and main memory, the main memory takes 100 nsec to return the first word (32 bit) of a line and 10 nsec to return each subsequent word. The cache memory has a hit rate of 95%, 128 bytes lines and cache hit latency of 5 nsec. The  $T_{miss}$  for this cache will be \_\_\_\_\_

(Assume that the cache waits until the line has been fetched into the cache and then re-executes the memory operation, resulting in a cache hit. Neglect the time required to write the line into the cache once it has been fetched from the main memory. Also assume that the cache takes the same amount of time to detect that a miss has occurred as to handle a cache hit.)

A

410 ns

Ans(B).

C

240 ns

D

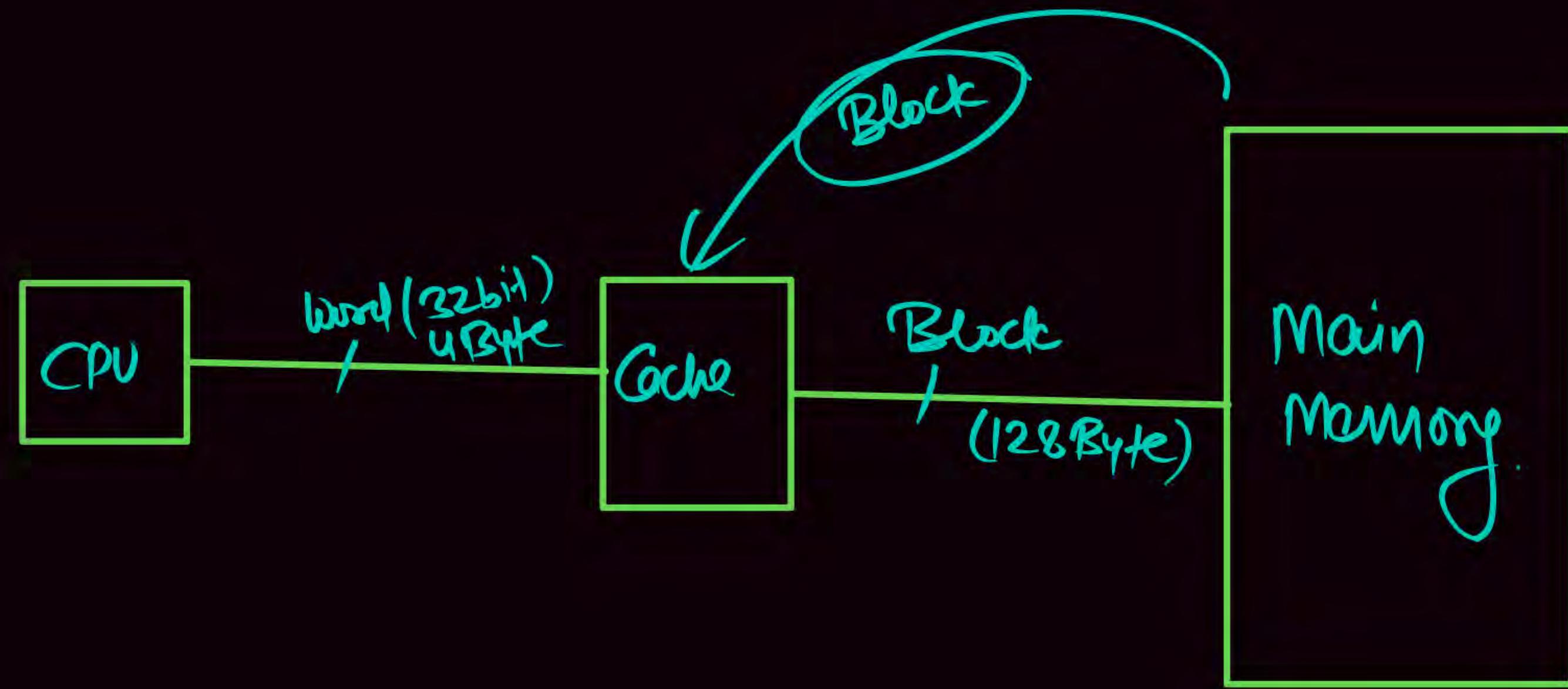
540 ns

B

420 ns

$$T_{avg} = h \cdot t_c + (1-h) (t_c + t_m)$$

Twiss



$T_{avg} = 10\text{ns/sec}$  for 1 word (32bit)

& longer for subsequent word

Cache Access = 5ns/sec.

Hit Ratio = 95%.

Line (Block Size) = 128 Byte

Time ?

$$\# \text{Words} = \frac{128 \text{ Byte}}{4 \text{ Byte}(32 \text{ bit})} = 32 \text{ Words}$$

$$\begin{aligned} T_{miss} &= 5 + [L(100) + 3L(10)] + 5 \\ &= 5 + 4L_0 + 5 \\ &= 42 \text{ ns/sec. Avg} \end{aligned}$$

#Q. Which of the following is not true regarding set associative cache when cache size is kept fixed?

A

Decreasing associativity increases set index size. *True*.

B

Decreasing associativity decreases tag metadata *True*

C

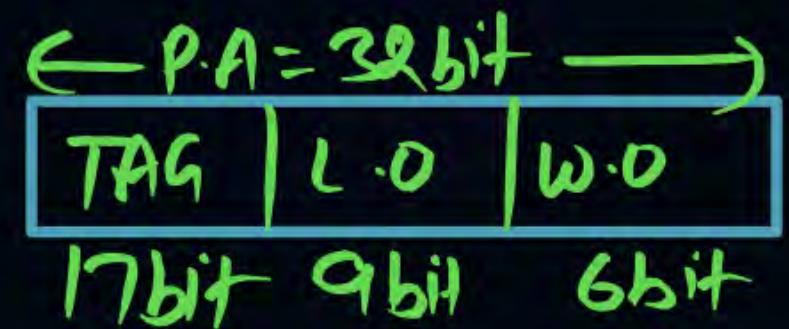
Increasing associativity increases tag metadata. *True*.

D

Increasing associativity increases number of sets *False*.

Ans(D).

Direct Map

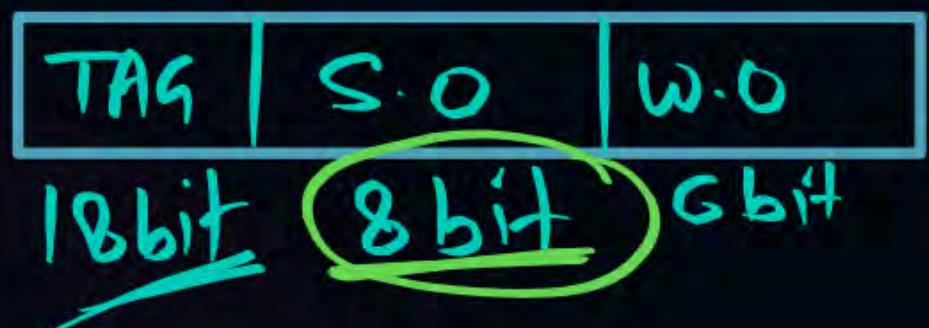


@ 4way  $\rightarrow$  2way

2 way Set  
Associative

$$\#SET = \frac{2^9}{2^1} = \cancel{2^8}$$

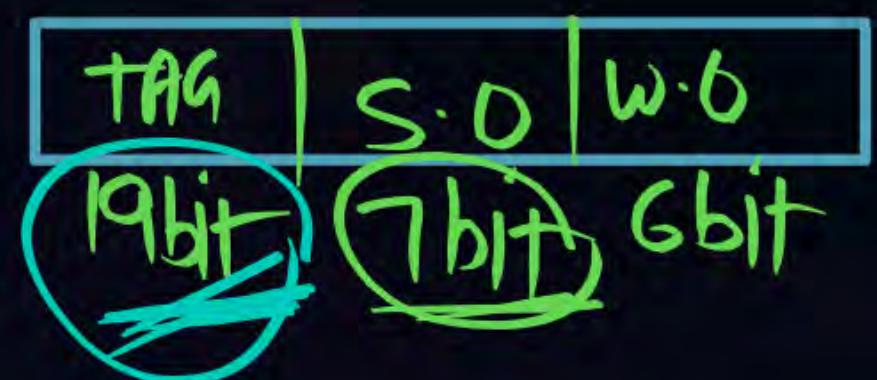
↓



Tag width = # Lines  $\times$  Tag bit

4 way Set  
Associative

$$\#SET = \frac{2^9}{2^2} = \cancel{2^7}$$



#Q. A system uses 2-level cache. For every 100 memory accesses generated by CPU, 90 hits in L<sub>1</sub> cache and 8 hits in L<sub>2</sub> cache. Access times of L<sub>1</sub>, L<sub>2</sub> and main memory are 10, 20 and 100 nanoseconds respectively. What is the average memory access time (in nanoseconds)?

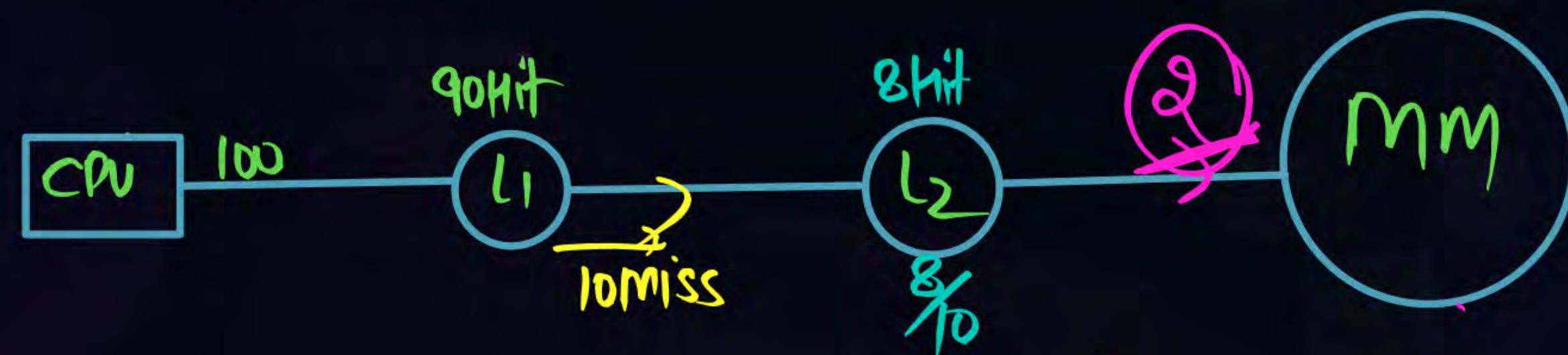
A 14 ns

C 21.2 ns

B 16 ns

D None of these

Ans (A)



$$1-h_1 = 0.1 \quad h_1 = \frac{90}{100} = 0.9$$

$$1-h_2 = 0.2 \quad h_2 = \frac{8}{10} = 0.8$$

$$L_1 = t_1 = 10 \text{ ms}$$

$$L_2 = t_2 = 20 \text{ ms}$$

$$Ma = 100 \text{ ms}$$

$$T_{avg} = h_1 t_1 + (1-h_1) h_2 (t_2 + t_1) + (1-h_1)(1-h_2) (t_m + t_2 + t_1)$$

OR

$$T_{avg} = t_1 + (1-h_1) [t_2 + (1-h_2)t_m]$$

$$\Rightarrow 10 + 0.1 [20 + 0.2(100)] \Rightarrow 10 + 0.1 [20 + 20] \\ = 10 + 0.1 [40]$$

$$T_{avg} = 14 \text{ ms}$$

$$T_{avg} = h \cdot tc + (1-h) (tm + tc)$$
$$\cancel{htc} + tm + tc - htm - \cancel{htc}$$

**OR**

$$tc + (1-h)tm$$

$$T_{avg} = tc + (1-h)tm$$

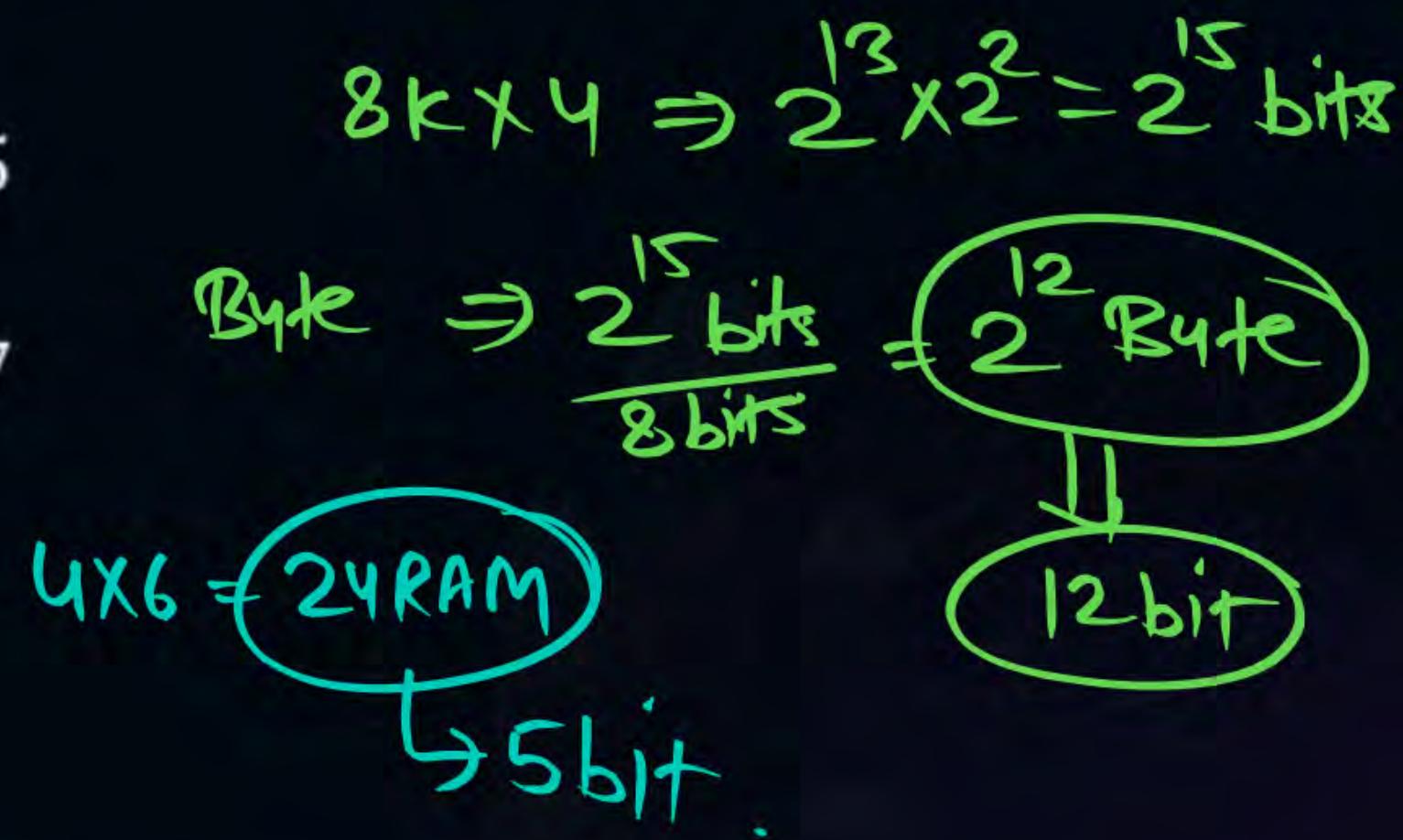
#Q. If each address space represents one byte of storage space, how many address lines are needed to access RAM chips arranged in a  $4 \times 6$  array, where each chip is  $8\text{K} \times 4$  bits?

- A 13
- C 16

**Ans(D)**

- B 15
- D 17

$$5 + 12 = 17 \text{ bit}$$



Consider a 64 kB ( $1 \text{ kB} = 2^{10} \text{ Byte}$ ), N-way set associative cache with cache block size of 32 byte. Assume that the cache is Byte addressable and the address sent by the processor is of 32 bit. If the tag field size is 19 bit then the value of N is 8 Way

$$\text{Cache Size} = 64 \text{ kB} (2^{16} \text{ Byte})$$

$$\text{Block Size} = 32 \text{ B} (2^5 \text{ Byte}) \Rightarrow w.o = 5 \text{ bit}$$

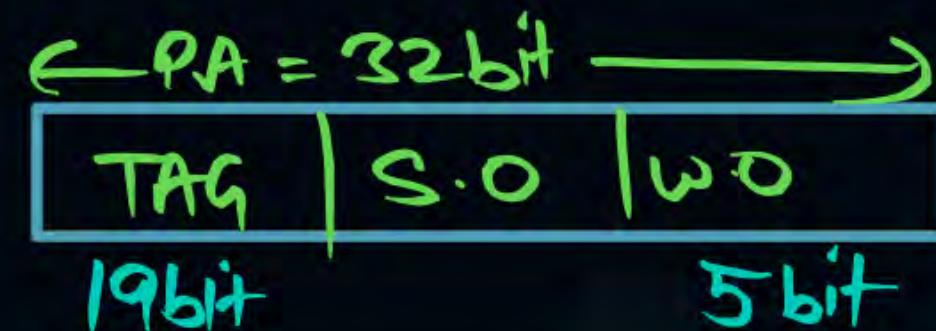
$$P.A = 32 \text{ bit}, \text{ TAG} = 19 \text{ bit}$$

$$\# \text{LINES} = \frac{\text{Cache Size}}{\text{Block Size}} = \frac{2^{16} \text{ B}}{2^5 \text{ B}} = 2^{11} \text{ lines}$$

$$\# \text{SET} = \frac{\# \text{LINES}}{N \text{-ways}}$$

$$2^8 = \frac{2^{11}}{N \text{-ways}}$$

$$N \text{-way} = \frac{2^{11}}{2^8} = 2^3 \Rightarrow 8 \text{ way set associative}$$



$$S.O = 32 - (19 + 5)$$

$$S.O = 8 \text{ bit}$$

$$\# \text{SET} = 2^3 \text{ sets}$$

[MCQ]

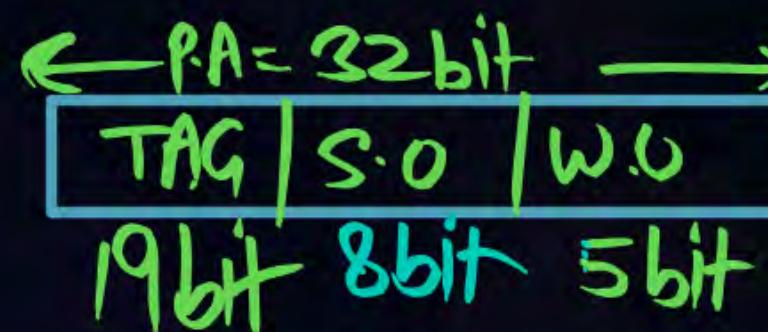
P  
W

#Q. Consider an 8-way set associative cache of size 64 kB. Block size is 32 bytes  
CPU generates 32-bit addresses. What is the size of tag comparators?

- A 17  
C 19

Ans (C)

- B 18  
D 20



$$\# \text{LINF} = \frac{64 \text{ kB}}{32 \text{ B}} = 2^10 \text{ lines}$$

$$\begin{aligned} &\text{8way} \\ &\# \text{SET} = \frac{2^{10}}{2^3} = 2^5 \end{aligned}$$

$$S.O = 8bit$$

# [MCQ]

P  
W

If a 8 way set associative cache is made up of 64 bit words. 8 words per line and 8192 sets. What is the size of cache memory?

- A 1 Mbyte
- B 2 Mbyte
- C 4 Mbyte
- D 8 Mbyte

8 Way Set Associative

$8192 [2^{13}]$  Set

Block Size = 8 Words

Block Size =  $8 \times 8B = \cancel{64\text{Byte}} = 2^6\text{Byte}$

#SET =  $\frac{\#UNES}{N\text{-ways}}$   $\Rightarrow$

$(2^{13})$   
 $8192 = \frac{\#UNF}{8(2^3)}$

#Lines =  $2^3 \times 2^3 = \cancel{2^6 \text{Lines}}$

Ans(C).

$$\# \text{LINES} = \frac{\text{CM Size}}{\text{Block Size}}$$

$$\text{CM Size} = \# \text{Lines} \times \text{Block Size}$$

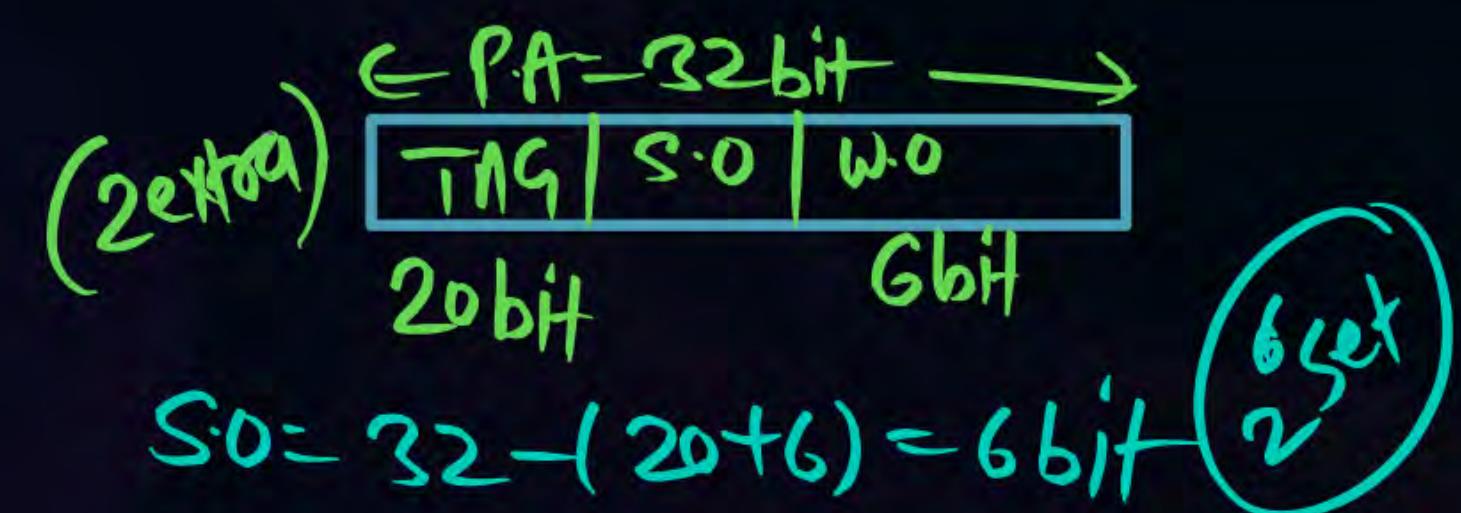
$$\Rightarrow 2^6 \times 2^6 \text{ Byte}$$

$$\Rightarrow 2^{12} \text{ Byte} \Rightarrow 4 \text{ MByte Ang}$$

#Q. Consider a 4-way set associative cache with block size = 64 bytes. CPU generates 32-bit addresses. Tag comparator requires 22 bits including a valid and a modified bit. What is the size of the cache?

- A 8 KB
- C 32 KB

- B 16 KB
- D 64 KB



$$\# \text{SET} = \frac{\# \text{LINES}}{N\text{-ways}}$$

$$2^6 = \frac{\# \text{LINES}}{4}$$

$\# \text{LINES} = 2^6 \times 4 = 2^8 \text{ Lines}$

$$\Rightarrow 2^8 \times 2^6 \times 64 = 2^{14} B$$

$= 16 kB$

#Q. Consider two level-memory, contains cache and main memory. Cache access time is 10 ns and main memory access time is 120 ns/word. The size of block is 4 words. Main memory is referred 20% of time. What is the average access time 106 (in Nano seconds)

$$t_c = 10 \text{ ns}$$

$$t_m = 120 \text{ ns/word}$$

4 word

$$TB = 4 \times 120 = 480 \text{ ns}$$

Ans (106)

$$\begin{aligned}T_{avg} &= .80 \times 10 + .20(10 + \frac{480}{4 \times 120}) \\&\Rightarrow 8 + .20(490) \\&= 8 + 98 \\&= 106\end{aligned}$$

Consider a system with two level cache hierarchies with  $L_1$  and  $L_2$  cache. Program refers memory 3000 times, out of which 30 misses are in  $L_1$  cache and 21 misses are in  $L_2$  cache. If miss penalty of  $L_2$  is 500 clock cycles, hit time of  $L_1$  is 2 clock cycle and hit time of  $L_2$  is 16 clock cycle, the average memory access time (in clock cycle) \_\_\_\_\_(upto 2 decimal places.)

$$L_1 \text{ Miss Rate} = \frac{30}{3000} = 0.010$$

$$L_2 \text{ Miss Rate} = \frac{21}{30} = 0.7$$

P  
W

$$T_{avg} = \text{Hit time}_{L_1} + \text{Miss Rate}_{\text{in } L_1} \left[ \text{Hit time}_{\text{in } L_2} + \text{Miss Rate}_{\text{in } L_2} (\text{Miss Penalty}) \right]$$

$$2 + 0.010 [16 + 0.7(500)]$$

$$2 + 0.010 [16 + 350]$$

$$2 + 0.010 [366]$$

$$2 + 3.66$$

$$5.66 \underline{\text{Avg}}$$

[MCQ]

P  
W

1 Word = 32 bit  
Word Addressable

#Q. Suppose you want to build a memory with 4 byte words and a capacity of  $2^{21}$  bits. What is type of decoder required if the memory is built using  $2K \times 8$  RAM chips?

- A 5 to 32
- C 4 to 16

**Ans (D).**

- B 6 to 64
- D 7 to 128

$2^7$  bits

$$2^{16} \times 2^5 \Rightarrow 2^{16} \times 32$$

$$\frac{2^{16} \times 32}{2K \times 8} \Rightarrow \frac{2^{16} \times 2^5}{2^11 \times 2^3}$$

RAM  
chip =

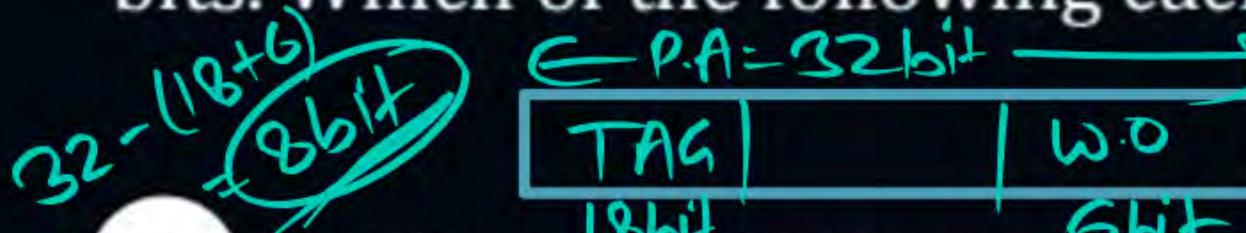
$7 \times 2^7$   
7 to 128.  
**Ans**

$7 \text{ bit req.}$  =  $2^7 \text{ RAM chip}$

#Q. Consider a write through cache memory which is 10 times faster than the main memory. The main memory access time is 800 ns. When there is a miss in the cache memory, an 8 word block is transferred from main memory to the cache memory. Hit ratio for read and write operations are 70% and 90% respectively. System generates 70% of read request. What is the throughput of the system (in K words/sec) is \_\_\_\_\_

# [MCQ]

Consider a computer system with a byte addressable main memory of size  $2^{32}$  byte and 64K bytes data cache memory with block of size 64 byte. Tag field length is 18 bits. Which of the following cache mapping used in this system?



Fully associative mapping

2-way set associative mapping

Direct mapping

4-way set associative mapping

**Ans (D)**

$$\begin{aligned} PA &= 32 \text{ bits} \\ \text{Block Size} &= 64 \text{ Byte} \\ \text{Cm Size} &= 64 \text{ KB} \\ \text{Tag} &= 18 \text{ bit} \end{aligned}$$

$$\# \text{Lines} = \frac{\text{Cm Size}}{\text{Block Size}} = \frac{64 \text{ KB}}{64 \text{ B}}$$

$$\# \text{Lines} = 2^{10} \text{ Lines}$$

$$S.O = 8 \text{ bit} \Rightarrow \# \text{SET} = 2^8$$

$$\# \text{SET} = \frac{\# \text{Lines}}{N \cdot \text{Way}}$$

$$2^8 = 2^{10} \xrightarrow{N \cdot \text{Way}} N \cdot \text{Way} = 2^8 / 2^2 = 4 \text{ Way}$$



# Home Work

#Q. Consider three cache organizations each of size 16 KB, with associativity as  $C_1$  - 2 WSA,  $C_2$ -4 WSA and  $C_3$ -8 WSA, and in all such organizations the block size is of 32 bytes and the size of physical address is 30 bits. A  $(4 \times 1)$  multiplexer having latency of "0.4" nsec along with "T" bits tag comparator latency of  $(T/10)$  nsec. If the hit latencies of cache organizations  $C_1$ ,  $C_2$  and  $C_3$  are  $H_1$ ,  $H_2$  and  $H_3$  then the relationship that can be established between hit latencies is \_\_\_\_\_

A

$H_1 > H_2 > H_3$

B

$H_1 > H_3 > H_2$

C

$H_1 < H_3 < H_2$

D

$H_1 < H_2 < H_3$

- #Q. Consider a direct mapped cache of size 32 KB and block size of 64 bytes.
- ② CPU generates 32-bit addresses. The difference in number of bits in tag meta(Memory) data in this organization with respect to 4-way set associative implementation is \_\_\_\_\_

#Q. The main memory address is divided into three field. The least significant 'w' bits can identify a unique word or byte within a block of main memory. Main memory has  $2^s$  blocks to represent there blocks we need 's' bits. The cache logic interprets there 's' bit as a tag of ' $s-r$ ' bits and a line field of 'r' bits.

The number of lines in cache will be

A  $2^r$

B  $2^{r+w}$

C  $2^{s+w}$

D undetermined

#Q. Consider a computer system with a byte addressable main memory of size  $2^{32}$  byte and 64 k Byte write back direct mapped cache with block of size 64 byte. Cache controller maintains the tag information for each cache block comprising of the following 1 bit for valid/invalid and 1 modified bit.

A

Tag memory size is 16 K bits.

B

Tag memory size is 18 k bits.

C

A total of  $2^{16}$  main memory block map to each cache line

D

A total of  $2^{18}$  main memory block map to each cache line.

#Q. Consider a p-way set associative cache with  $(8*p)$  blocks. Assume that main memory has  $(16*p)$  blocks. What is the tag size in this organization?

A

$$\log_2 P + 4$$

B

$$\log_2 P - 1$$

C

$$\log_2 P + 1$$

D

None of these



Ans [C]

Cache Size = 8P Block

MM Size = 16P Blocks

$$\text{Tag} = \frac{\text{MM Size}}{\text{CM Size}} = \frac{16P}{8P} = 2$$

TAG = 2

Tag bit = 1bit

Pway Set  
Associative

$$\text{Tag bits} = 1 + \lceil \log_2 P \rceil \cong$$

5 Consider a computer system which has 4GB, byte addressable main memory and cache size 8MB, 4way set associative cache memory with block size 4096 byte. Consider the following six physical addresses represented in a hexadecimal notation.

$$A_1 = 0 \times 47\text{ CA4 ABC}$$

$$A_2 = 0 \times 56\text{ ECF 38D}$$

$$A_3 = 0 \times 29\text{ FDB 4CF}$$

$$A_4 = 0 \times 38\text{ 8A4 DAC}$$

$$A_5 = 0 \times C3\text{ EFC A47}$$

$$A_6 = 0 \times AB\text{ 9DB 128}$$

Which of the following is correct ?

- A  $A_1$  and  $A_4$  are mapped to the same cache set.
- B  $A_2$  and  $A_5$  are mapped to the same cache set.
- C  $A_2$  and  $A_5$  are mapped to the different cache set.
- D  $A_3$  and  $A_6$  are mapped to the same cache set.

#Q. 6

Consider a 4 way set associative cache of size 16 KB and having block size 1024 byte. Assume cache is initially empty. Consider least recently used (LRU) policy for replacement. What is the miss ratio (in %) for the below memory access request?

OX A25BC8DF, OX FBCDFBAB, OX ABCDE9DF  
OX FDADC8DF, OXFBCDFA3D, OX44A11C25  
OX A25BCBAE, OXBCADE19F, OXA25BC92D  
OXFBCDF849, OX 44A11FFF, OX FDADCA13

A 40%

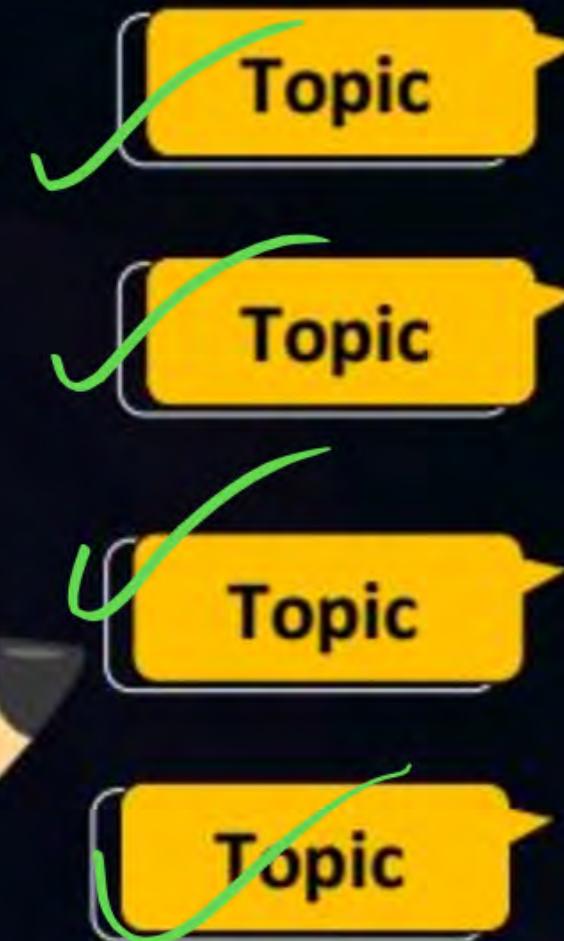
C 60%

B 50%

D 100%



## 2 mins Summary



Cache Memory

Cache mapping Technique

Cache Replacement Algo

Cache Updating Technique



THANK - YOU

# CS & IT ENGINEERING

## Computer Organization & Architecture

1500 Series

Lecture No.- 07

By- Vijay Agarwal Sir





# Recap of Previous Lecture



Topic

Cache Memory

Topic

Cache mapping Technique

Topic

Cache Replacement Algorithm

Topic

Cache Updating Technique



# Topics to be Covered



Topic

Cache Memory

Topic

Disk Access Time

Topic

Disk Addressing

Topic

IO Interface & DMA

#Q. Consider three cache organizations each of size 16 KB, with associativity as  $C_1$  - 2 WSA,  $C_2$ -4 WSA and  $C_3$ -8 WSA, and in all such organizations the block size is of 32 bytes and the size of physical address is 30 bits. A  $(4 \times 1)$  multiplexer having latency of "0.4" nsec along with "T" bits tag comparator latency of  $(T/10)$  nsec. If the hit latencies of cache organizations  $C_1$ ,  $C_2$  and  $C_3$  are  $H_1$ ,  $H_2$  and  $H_3$  then the relationship that can be established between hit latencies is \_\_\_\_\_

A  $H_1 > H_2 > H_3$

C  $H_1 < H_3 < H_2$

B  $H_1 > H_3 > H_2$

D  $H_1 < H_2 < H_3$

Ans(D)

$$H_1 = 2 \cdot 1$$

$$H_2 = 2 \cdot 2$$

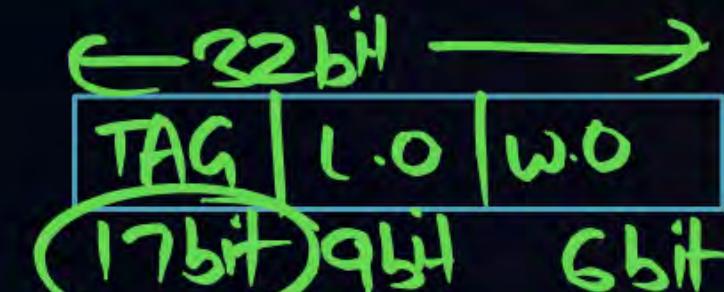
$$H_3 = 2 \cdot 3$$

Tag   S.O   W.O	$= \frac{17}{10} + 0.4 = 2.1$
17bit 8bit 5bit	

Tag   S.O   W.O	$= \frac{18}{10} + 0.4 = 2.2$
18bit 7bit 5bit	

Tag   S.O   W.O	$= \frac{19}{10} + 0.4 = 2.3$
19bit 6bit 5bit	

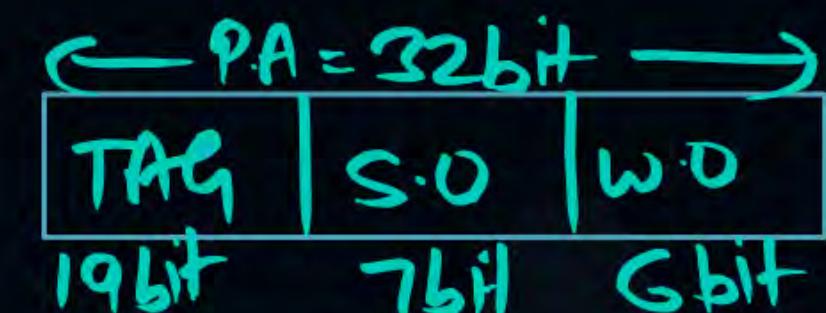
#Q. Consider a direct mapped cache of size 32 KB and block size of 64 bytes. CPU generates 32-bit addresses. The difference in number of bits in tag meta(Memory) data in this organization with respect to 4-way set associative implementation is 1024 Ans



Ans (1024)

$$\# \text{Lines} = \frac{32 \text{ kB}}{64 \text{ Byte}} = \frac{2^{15}}{2^6} = 2^9$$

$$\text{Tag Memory Size} = 17 \times 2^9 = 8704$$



$$\# \text{SET} = \frac{2^9}{2^2} = 2^7$$

$$\begin{aligned} \text{Tag Memory} &= 19 \times 2^9 \\ &= 9728 \end{aligned}$$

$$9728 - 8704$$

$$= \boxed{1024} \text{ Ans}$$

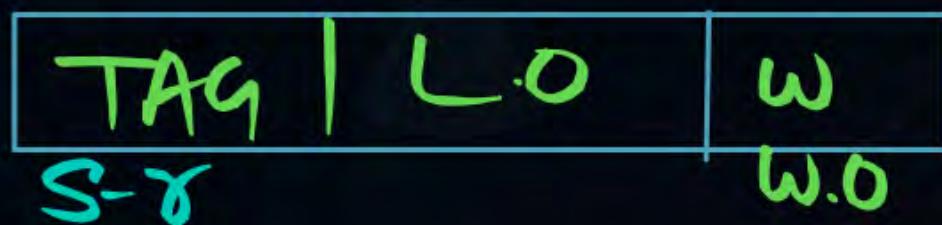
#Q. The main memory address is divided into three field. The least significant 'w' bits can identify a unique word or byte within a block of main memory. Main memory has  $2^s$  blocks to represent there blocks we need 's' bits. The cache logic interprets there 's' bit as a tag of 's-r' bits and a line field of 'r' bits.

After Mapping

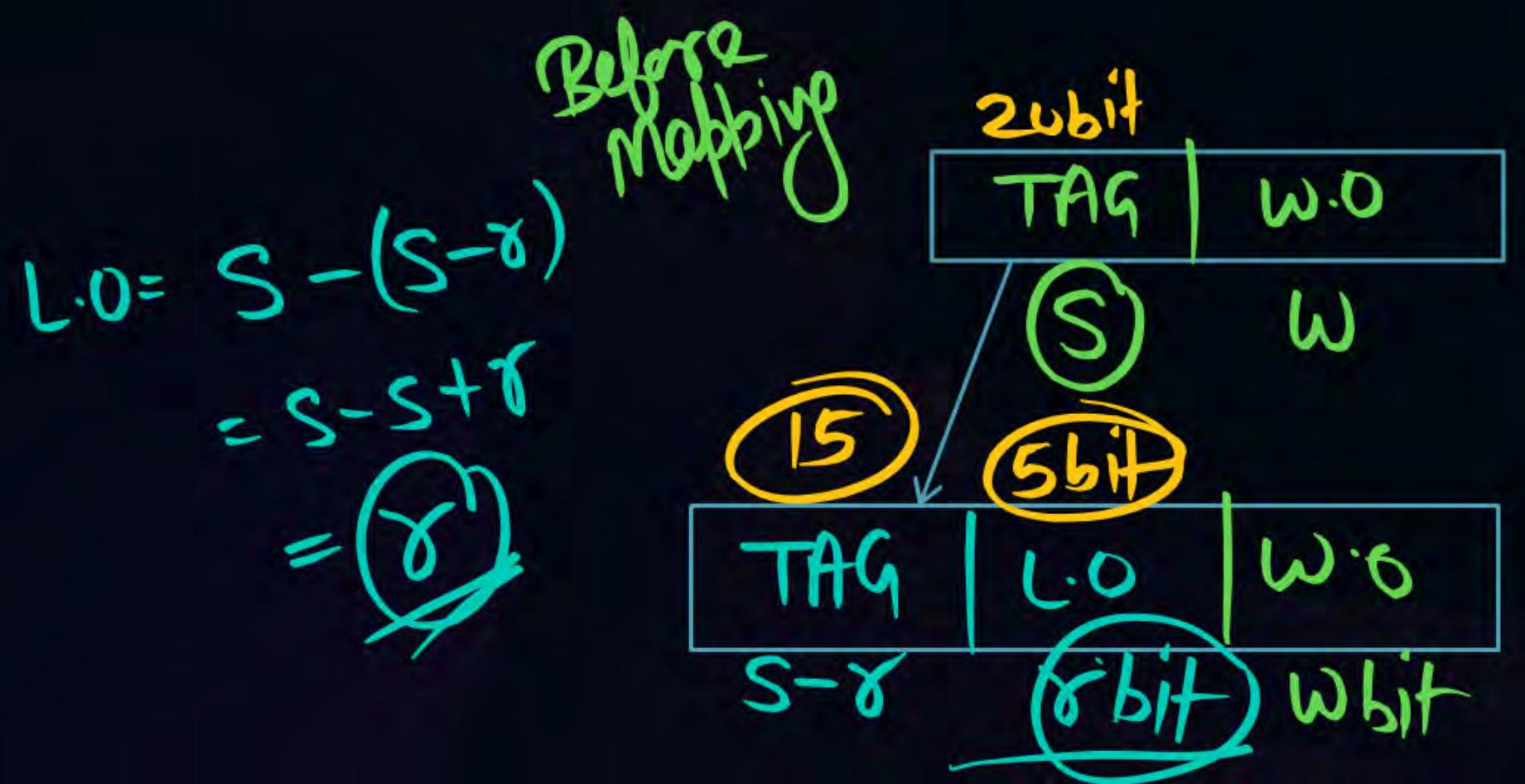
The number of lines in cache will be

- A  $2^r$
- C  $2^{s+w}$

- B  $2^{r+w}$
- D undetermined



$$\# \text{MM Blocks} = 2^S$$



#Q. Consider a computer system with a byte addressable main memory of size  $2^{32}$  byte and 64 k Byte write back direct mapped cache with block of size 64 byte. Cache controller maintains the tag information for each cache block comprising of the following 1 bit for valid/invalid and 1 modified bit. Which of the following is correct?

A

Tag memory size is 16 K bits.

B

Tag memory size is 18 k bits.

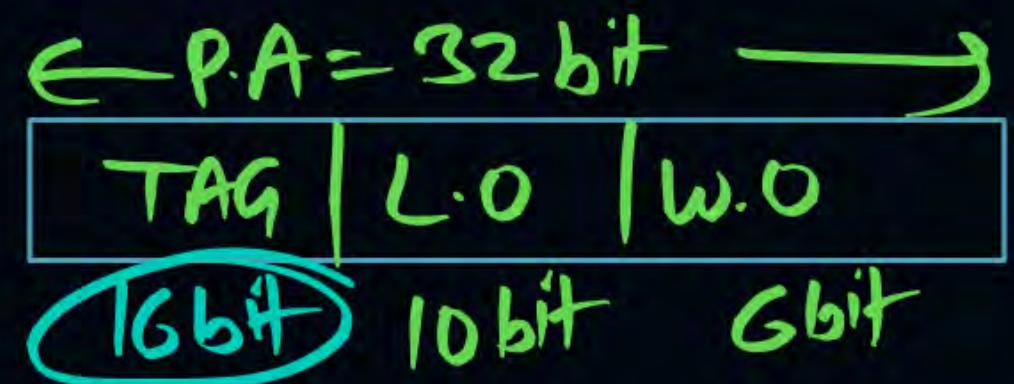
C

A total of  $2^{16}$  main memory block map to each cache line

D

A total of  $2^{18}$  main memory block map to each cache line.

Ans (B) & (C)



$$\# \text{Lines} = \frac{\text{Mem Size}}{\text{Block Size}} = \frac{64\text{ kB}}{64\text{ B}} = 2^{10}$$

$\text{TAG} = 16 \text{ bit} \Rightarrow 2^6 \text{ Main Memory are fighting for each Cache line.}$

$$\text{Tag Entry Size} = 16 + 1 + 1 = 18 \text{ bits}$$

$$\text{Tag Memory Size} = \# \text{Lines} \times \text{Tag entry size}$$

$$\Rightarrow 2^{10} \times 18 \\ = 18 \text{ k bits Avg}$$

#Q. Consider a p-way set associative cache with  $(8*p)$  blocks. Assume that main memory has  $(16*p)$  blocks. What is the tag size in this organization?

A       $\log_2 P + 4$

C       $\log_2 P + 1$

B       $\log_2 P - 1$

D      None of these

*Done*

[MSQ]

P  
W

Consider a computer system which has 4GB, byte addressable main memory and cache size 8MB, 4way set associative cache memory with block size 4096 byte. ( $2^{12}$ )

Consider the following six physical addresses represented in a hexadecimal notation.

$$A_1 = 0 \times 47\text{ CA4 ABC}$$

$$A_2 = 0 \times 56\text{ ECF 38D}$$

$$A_3 = 0 \times 29\text{ FDB 4CF}$$

$$A_4 = 0 \times 38\text{ 8A4 DAC}$$

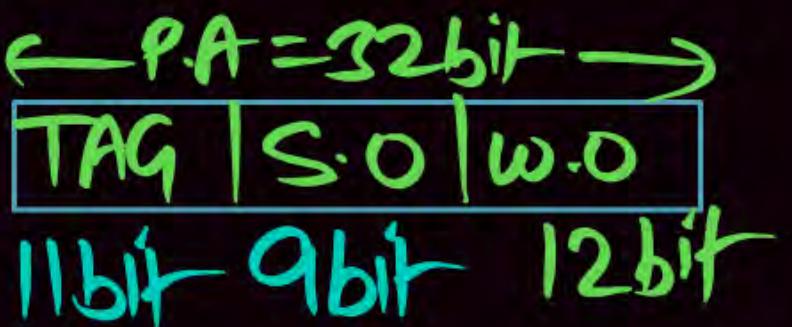
$$A_5 = 0 \times C3\text{ EFC A47}$$

$$A_6 = 0 \times AB\text{ 9DB 128}$$

Which of the following is correct ?

- A  $A_1$  and  $A_4$  are mapped to the same cache set.
- B  $A_2$  and  $A_5$  are mapped to the same cache set.
- C  $A_2$  and  $A_5$  are mapped to the different cache set.
- D  $A_3$  and  $A_6$  are mapped to the same cache set.

**Ans(A)(C) & (D)**

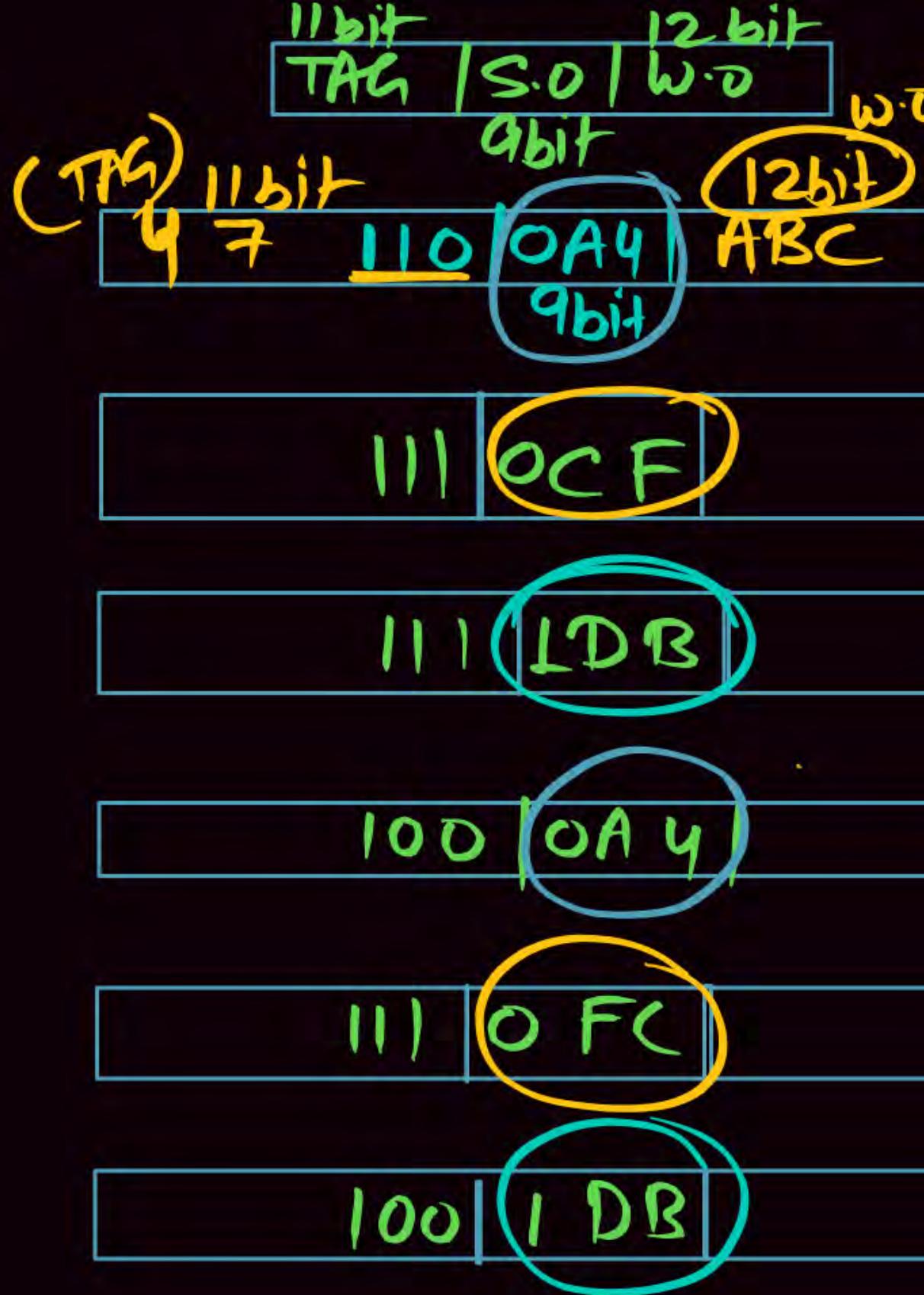


$$\# \text{Lines} = \frac{\text{CM Size}}{\text{Block Size}} \Rightarrow \frac{8\text{MB}}{4096\text{B}} \Rightarrow \frac{2^{23}\text{B}}{2^{12}\text{B}} = 2^11 \text{ Lines}$$

$$\# \text{SET} = \frac{\# \text{Lines}}{\text{N-way}} = \frac{2^{11}}{2^2} = 2^9 \text{ set}$$

S.O = 9bit

A1 : 0x 47 CA4ABC  
 A2: 0x 56 ECF B8D  
 A3: 0x 29 FDRCCF  
 A4: 0x 38 8A4DAC  
 A5: 0x C3 EFC A47  
 A6: 0x AB9DB 28



A1 & A4 Same Cache Set.

A3 & A6 Same Cache Set

A2 & A5 Different Cache Set.

		Set No.	W.O	DF	→ Miss
①	0x A25BC8DF	①	8:	10 00	→ Miss
②	0x FBCDF BAB	②	B:	10 11	→ Miss
③	0x ABCDE 9DF	③	9:	10 0L	→ Miss
④	0x FDADC 8DF	④	8:	10 00	→ miss
⑤	0x FBCDF A3D	⑤	A:	10 10	⇒ HIT
⑥	0x 4VA11C25	⑥	C:	11 00	→ Miss
⑦	0x A25BC BAE	⑦	B:	10 11	⇒ HIT
⑧	0x BC ADE 1AF	⑧	L:	00 0L	→ miss
⑨	0x A25BC 92D	⑨	9:	10 0L	⇒ HIT
⑩	0x FBCDF 849	⑩	8:	10 00	⇒ HIT
⑪	0x 4VA11FFF	⑪	F:	11 11	⇒ HIT
⑫	0x FDADC A13	⑫	A:	10 10	⇒ HIT

TAG	S.O	W.O	P W
20bit	2bit	10bit	
0x BC ADE 19F			
SET 0	00		
SET 1	01		
SET 2	10		
SET 3	11		

OX A25BC8DF  
OX FBCDFBAB  
OX ABCDE9DF  
OX FDADC8DF  
OX 4VA11C25

## Disk

- ① Disk Structure
- ② Disk Access time
- ③ Disk Addressing

## Do Interface

Consider a disk which has 16 platter each platter has two surfaces. Every surface has 1k tracks and every track is further divided into 512 sector and every sector can store the 2kB data. Then which of the following is correct?

$$(i) \text{ Capacity} = 16 \times 2 \times 1\text{k} \times 512 \times 2\text{kB}$$
$$= 2^4 \times 2^1 \times 2^{10} \times 2^9 \times 2^{11} = 2^{35} \text{ Byte} = 32\text{GB}$$

A The capacity of Disk is 16 GB.

B The capacity of Disk is 32 GB

C 24 bits are required to identify any particular sector of the disk.

D 35 bits are required to identify any particular sector of the Disk.

$$(ii) \#bits = 2^4 \times 2^1 \times 2^{10} \times 2^9 = 2^{24} = 24\text{bits}$$

Ans(B) & (C).

Consider a typical disk that rotates at 12000 RPM (Rotation per minutes) and has a transfer rate of 30 MBps [ $30 \times 10^6$  bytes/sec]. If the average seek time of the disk is thrice the average rotational delay and the controllers transfer time is 40 times the disk transfer time. The average time (in milliseconds) to read or write 256 byte sector of the disk is 10.35 msec Ans

$$\text{Disk Access time} = \text{Avg S.T} + \text{Avg R.L} + \text{D.T.T} + \text{Controller overhead}$$

(T<sub>avg</sub>)

$$\text{Ans}(10.35)$$

12000 Rotation in 60sec

$$\text{In } \perp \text{ rotation} = \frac{G\theta}{12000} \Rightarrow \frac{1}{200} \text{ sec} \times \frac{1000}{1000} \\ \Rightarrow 5 \text{ msec}$$

$$R.L = \frac{1}{2} \times 5 \text{ msec} \Rightarrow R.L = 2.5 \text{ msec}$$

$$\text{Ang S.T} = 3 \times R.T \Rightarrow 3 \times 2.5 = 7.5 \text{ msec}$$

$$\text{Ang S.T} = 7.5 \text{ msec}$$

30MBps

30  $\times$  10<sup>6</sup> Byte ————— 1 Sec.

$$256 \text{ Byte} \rightarrow \frac{256 \text{ B}}{30 \times 10^6 \text{ B}} \text{ SEC} \Rightarrow \frac{256}{30} \times 10^{-6} \text{ sec}$$

$$D.T.T = 8.33 \times 10^{-6} \text{ SEC} \Rightarrow 0.00833 \times 10^3 \text{ SEC} \Rightarrow 8.33 \times 10^{-6} \text{ SEC}$$

$$D.T.T = 0.00853 \text{ msec}$$

$$\text{Controlled overhead} = 40 \times (0.00853) \Rightarrow 0.341 \text{ msec}$$

$$\begin{aligned} D.A.T &= 7.5 + 2.5 + 0.00853 + 0.341 \\ &= 10.35 \text{ msec} \quad \underline{\text{Ans}} \end{aligned}$$

A hard disk has 63 sector per track, 20 platters each with 2 recording surface and 1600 cylinders.

The address of a sector is given as a triple  $\langle C, H, S \rangle$  where C is the cylinder number, h is the surface number and S is sector number. Thus the 0th sector is addressed as  $\langle 0,0,0 \rangle$  and the 1st sector is  $\langle 0,0,1 \rangle$  and So on.

The sector number of the corresponding address  $\langle 600, 38, 47 \rangle$  is \_\_\_\_\_.

$$\begin{array}{l} (\#track | \text{Cylinder}) \\ \text{Total } \# \text{ Surface} = 2 \times 20 = 40 \text{ Surface} \\ \# \text{Sectors Per track} = 63 \text{ Sector} \end{array}$$



<C h S>  
(600, 38, 47)

To Cross (Traversed) 600 Cylinder  $\Rightarrow$   $600 \times 40 \times 63 = 1512000$   
(0-599)

To Cross (Traversed) 38 Surface =  $38 \times 63 = 2394$   
(0-37)

To Cross 47 Sector  
(0-46) =  $\frac{47}{1514441}$

$(C, h, S)$        $C \text{ h } S$   
 $(600, 38, 47)$

$$\begin{array}{l} \text{Sector} \\ \text{Number} \end{array} = S + ST * h + \underbrace{ST * TC * C}_{SC}$$

$$\begin{aligned} & 47 + 63 * 38 + 63 * 40 * 600 \\ & = 1514441 \quad \underline{\text{Ans}} \end{aligned}$$

ST: # Sector Per track

TC: # Track Per cylinder

SC: # Sector Per cylinder.

# [MCQ]



Suppose, in a disk addressing cylinder number is C, surface number is P and sector number Q. Suppose, number of sectors per cylinder are R and number of sectors per track are S then, the sector sequence number will be:

$$\langle C, h, S \rangle \Rightarrow \langle C, P, Q \rangle$$

- A CR + PS + Q
- B PRQS + Q
- C (CR + PS)Q
- D None of the above

$$\text{Sector Number} = \underbrace{Q}_{\# \text{Sector/cylinder}} + \underbrace{\frac{ST * R}{S}}_{SC(R)} + \underbrace{ST * TC * C}_{SC(R)}$$

$$\text{Sector Number} = Q + SP + RC$$

Ans(A)

Number of Records per Sector =  $\frac{512B}{256B} = 2$  Record per Sector

# Sectors per track = 63

# Tracks per cylinder = 10

# Sectors per cylinder =  $63 \times 10 = 630$  sectors

# Records per cylinder =  $630 \times 2 = 1260$  Records per cylinder

Total # Records = 92,900

# Cylinders Needed =  $\lceil \frac{92,900}{1260} \rceil = 74$  Cylinders

#Q. A device With data transfer rate 10 KB/sec is connected to a CPU. Data is transferred byte-wise. Let the interrupt overhead be 4  $\mu$ sec. The byte transfer time between the device interface register and CPU or memory' is negligible. The minimum performance gain of operating the device under interrupt mode over operating it under program controlled mode is

25 Ans

Prog 10 :  $10KB \rightarrow 1sec$   
 $1Byte \rightarrow \frac{1}{10K} sec \Rightarrow 100 msec.$

Ans(25)

Interrupt = 4  $\mu$ sec

$$\text{Speedup Factor} = \frac{100}{4} = \boxed{25} \text{ Ans}$$

$$S = \frac{\text{Performance of Interrupt}}{\text{Performance of Programmed I/O}}$$

$$\Rightarrow \frac{T_{ET\text{ Interrupt}}}{T_{ET\text{ Prog}}} \Rightarrow \frac{ET_{\text{Prog}}}{ET_{\text{Inter}}} \Rightarrow 100 : 25$$

#Q. On a non-pipelined sequential processor, a program segment, which is a part of the interrupt service routine, is given to transfer 500 bytes from an I/O device to memory.

Initialize the address register

Initialize the count to 500

LOOP: Load a byte from device

Store in memory at address given by address register

Increment the address register

Decrement the count

If count! = 0 go to LOOP

Assume that each statement in this program is equivalent to a machine instruction which takes one clock cycle to execute if it is a non-load/store instruction. The load-store instructions take two clock cycles to execute.

The designer of the system also has an alternate approach of using the DMA controller to implement the same transfer. The DM controller requires 20 clock cycles for initialization and other overhead. Each DMA transfer cycle takes two clock cycles to transfer one byte of data from interrupt driven program-based input-output?

A

3.4

B

4.4

C

5.1

D

6.7

[NAT]

P  
W

The size of the data count register of DMA controller is 16 bits. The processor needs to transfer a file of 28,824 kilo bytes from disk to main memory. The memory is byte addressable. The minimum number of times the DMA controller need to get the control of the system bus from the processor to transfer the file from the disk to main memory is \_\_\_\_.

$$\text{Count} = 16 \text{ bit} \Rightarrow 2^{16} \text{ Byte} \Rightarrow 65,535 \text{ B}$$

$$\text{Total File Size} = 28,824 \text{ KByte}$$

$$\begin{aligned}\# \text{Times DMA Controller Need the Control} &= \frac{28824 \times 1024 \text{ B}}{65535 \text{ B}} \\ &= 451 \text{ Ans}\end{aligned}$$

- #Q. A hypothetical DMA is designed to transfer the data from i/o device to main memory under burst transfer mode. The count register size is 36 bits and gets the control of the system buses 8 times then the maximum size of the data transferred by controller is (in Giga Byte) 512 GB

Count Register : 36 bits  $\Rightarrow 2^{36}$  Byte . in One time

In 8time  $\Rightarrow 2^{36} \times 8(2^3)$

$$\Rightarrow 2^{36} \times 8 \times 2^{30} = 2^9 \text{ GB.}$$
$$= 512 \text{ GB}$$

Common Data for next Four questions:

A hard disk has 63 sectors per track, 10 platters each with 2 recording surfaces and 1000 cylinders. The address of a sector is given as a triple  $\langle c, h, s \rangle$ , where  $c$  is the cylinder number,  $h$  is the surface number and  $s$  is the sector number. Thus, the 0<sup>th</sup> sector is addressed as  $\langle 0, 0, 0 \rangle$ , the 1<sup>st</sup> sector as  $\langle 0, 0, 1 \rangle$ , and so on.

$$\# \text{Surface} = 10 \times 2 = 20 \text{ Surface}$$

$$\# \text{sector per track} = 63$$

$$\# \text{sector per Cylinder} = 20 \times 63$$

$$= 1260 \text{ sector per Cylinder}$$

#Q. The address  $\langle 400, 16, 29 \rangle$  corresponds to sector number:

$$\begin{aligned} \text{(i) To cross } 400 \text{ Cylinder (0-399)} &= 400 \times \underline{20 \times 63} = 504000 \\ \text{(ii) To cross 16 Surface (0-15)} \Rightarrow & 16 \times 63 = 1008 \\ \text{(iii) To cross 29 Sector (0-28)} &= 29 \end{aligned}$$

- |   |        |   |        |   |        |   |        |
|---|--------|---|--------|---|--------|---|--------|
| A | 505035 | B | 505036 | C | 505037 | D | 505038 |
|---|--------|---|--------|---|--------|---|--------|

#Q. The address of 1039<sup>th</sup> sector is

$$\# \text{Sectors Per Cylinder} = 1260$$

- A  $\langle 0, 15, 31 \rangle$
- Ans (C)
- C  $\langle 0, 16, 31 \rangle$
- B  $\langle 0, 16, 30 \rangle$
- D  $\langle 0, 17, 31 \rangle$

@  $15 \times 63 + 31 = 976$

⑤  $16 \times 63 + 30 = 1038$

⑥  $16 \times 63 + 31 = 1039$

⑦  $17 \times 63 + 31 = 1102$

#Q. In the Previous Question what is the sector address of 4734th sector is ?

$\langle C, h, s \rangle$

$$C = \left\lfloor \frac{\text{Given Sector}}{\# \text{Sector Per Cylinder}} \right\rfloor$$

Cylinder No.

$$h = \left\lfloor \frac{\text{Given Sector} / \# \text{Sector Per Cylinder}}{\# \text{Sector Per track}} \right\rfloor$$

Surface No.

P  
W

⑧ 4734

# Sectors per cylinder = 1260

$$C = \left\lfloor \frac{4734}{1260} \right\rfloor = 3$$

$$4734 - 3 \times 1260$$

$$\Rightarrow 4734 - 3780$$

h  
Surface

$$\left\lfloor \frac{(4734 \% 1260) / 63}{63} \right\rfloor = \frac{954}{63} = \left\lfloor 15 \cdot 14 \right\rfloor = 15 = 954$$

Sector

$$S = 954 - 15 * 63 = 9$$

$\langle 3, 15, 9 \rangle$  Ang

#Q. In the Previous Question what is the sector address of 5433 th sector is ?

$$C = \left\lfloor \frac{5433}{1260} \right\rfloor = \lfloor 4.311 \rfloor = 4$$

$$\begin{aligned} S_{433} - 4 \times 1260 \\ = 393 \end{aligned}$$

$$H = \left\lfloor \frac{393}{63} \right\rfloor = \lfloor 6.2 \rfloor = 6$$

$4, 6, 15 > Avg$

$$S = 393 - \underline{6 \times 63} - 15$$



## 2 mins Summary





THANK - YOU