CS & IT ENGINEERING

COMPUTER ORGANIZATION AND ARCHITECTURE

Instruction & Addressing Modes

Lecture No.- 01

By- Vishvadeep Gothi sir

Recap of Previous Lecture

Topic Micro operations

Topics to be Covered

Topic Instructions

Topic Opcode

$$\underline{x * 2}$$

$\Downarrow$

one time left shift

$(11)_2 = 3$

$(110)_2 = 6$

$(1100)_2 = 12$

$(11000)_2 = 24$

$* 8$

3 times left shift

[MCQ] **[2023 : 2M]**

#Q. Consider the given C-code and its corresponding assembly code, with a few operands U1–U4 being unknown. Some useful information as well as the semantics of each unique assembly instruction is annotated as inline comments in the code. The memory is byte-addressable.

$r_1 = \cancel{0}\ 1$

$r_3 = a + 4$

$r_4 = b + 4$

$r_5 = b[0] * 8$

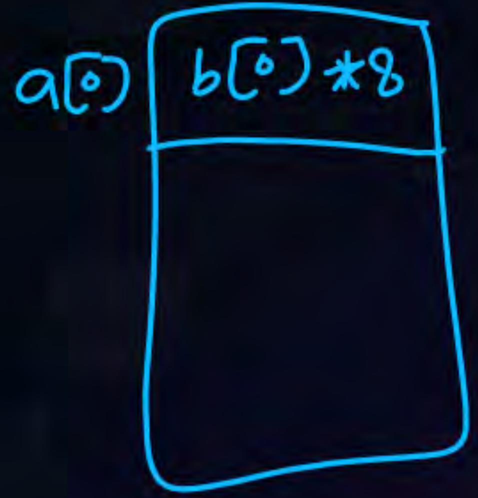
```
//C-code




int a[10], b[10], i;
// int is 32-bit → 4B
for (i=0; i<10;i++)
a[i] = b[i] * 8;
```

a[0] = b[0] *8

a[1] = b[1] *8

⋮

```
;assembly-code (; indicates comments)
;r1-r5 are 32-bit integer registers
;initialize r1=0, r2=10
;initialize r3, r4 with base address of a, b
L01: jeq r1, r2, end       ;if(r1==r2) goto end
L02: lw r5, 0(r4)          ;r5 <- Memory[r4+0]
L03: shl r5, r5, U1        ;r5 <- r5 <<U1 3
L04: sw r5, 0(r3)          ;Memory[r3+0] <- r5
L05: add r3, r3, U2        ;r3 <- r3+U2 4
L06: add r4, r4, U3 4
L07: add r1, r1, 1
L08: jmp U4                ;goto U4
L09: end
```



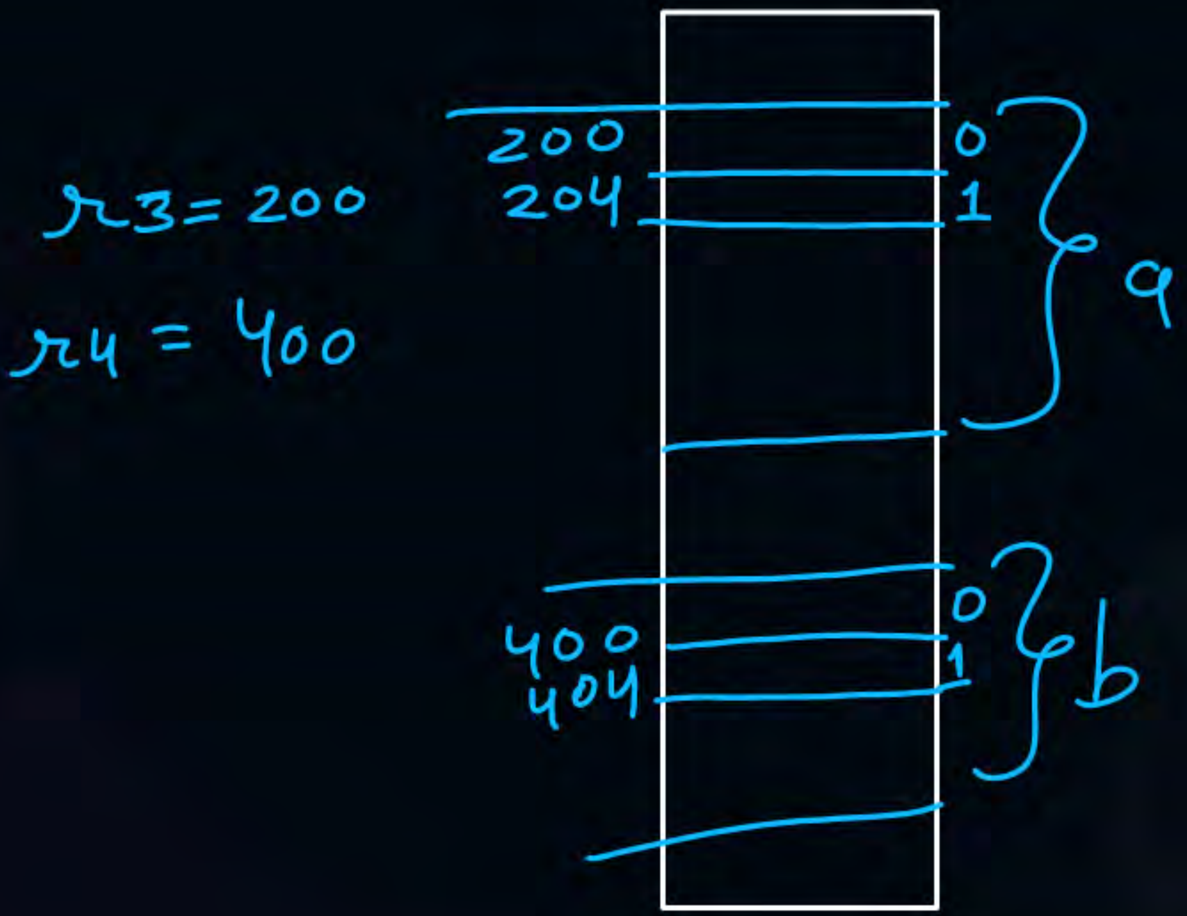
Which of the following options is a correct replacement for operands in the position (U1, U2, U3, U4) in the above assembly code?

A (8, 4, 1, L02)

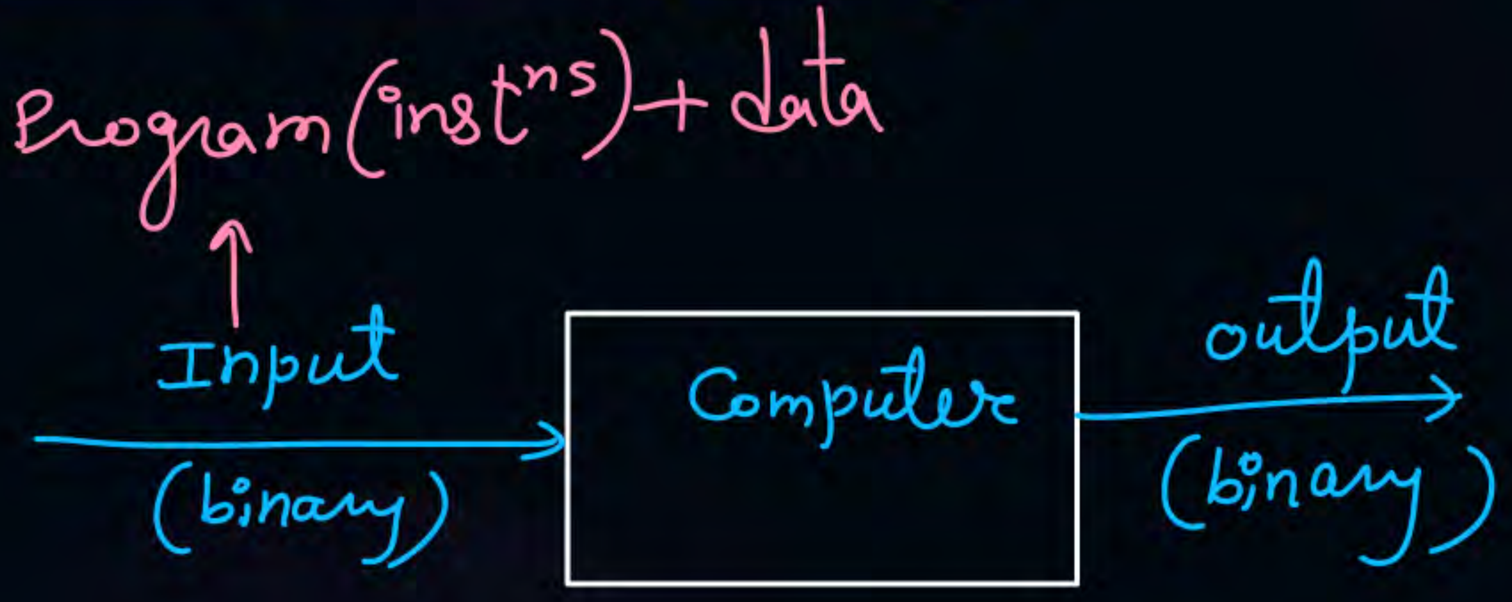
[✓ B] (3, 4, 4, L01)

C (8, 1, 1, L02)

D (3, 1, 1, L01)



Topic : Digital Computer



Topic : Instruction

```
#include<stdio.h>
void main()
{
int a, b, c;
printf(“Enter 2 values: ”);
scanf(“%d %d”, &a, &b);
c = a + b;
printf(“Sum = %d”, c);
}
```

Topic : Instruction

High level lang.

```
#include<stdio.h>
void main()
{
int a, b, c;
printf("Enter 2 values: ");
scanf("%d %d", &a, &b);
c = a + b;
printf("Sum = %d", c);
}
```

Programming stmts

(Compiler)

Language Translation

Low level lang. prog.

(machine code or binary code or byte code or object code)

Instructions

```
1 0 1 1 1 0 0 0
1 0 0 0 0 0 0 1
1 1 1 1 0 0 1 0
0 1 0 1 0 1 0 1
1 1 1 1 0 1 1 0
0 1 0 1 0 1 0 1
1 0 0 0 1 1 1 1
1 0 1 0 0 0 1 1
0 0 1 1 1 1 0 1
```

Topic : Instruction

A group of bits which instructs computer to perform some operation

operation code	operands info^{n}

or op'code

000 ⇒ ADD

001 ⇒ SUB

010 ⇒ complement

⋮

111 ⇒ ____

if no. of distinct operations CPU can perform = 8

⇓

inst^{n} opcode = 3 bits

→ no. of distinct inst^{ns} supported by CPU.

Topic : Instruction

no. of distinct inst^{ns} supported by CPU = n

opcode = $\lceil \log_2 n \rceil$ bits

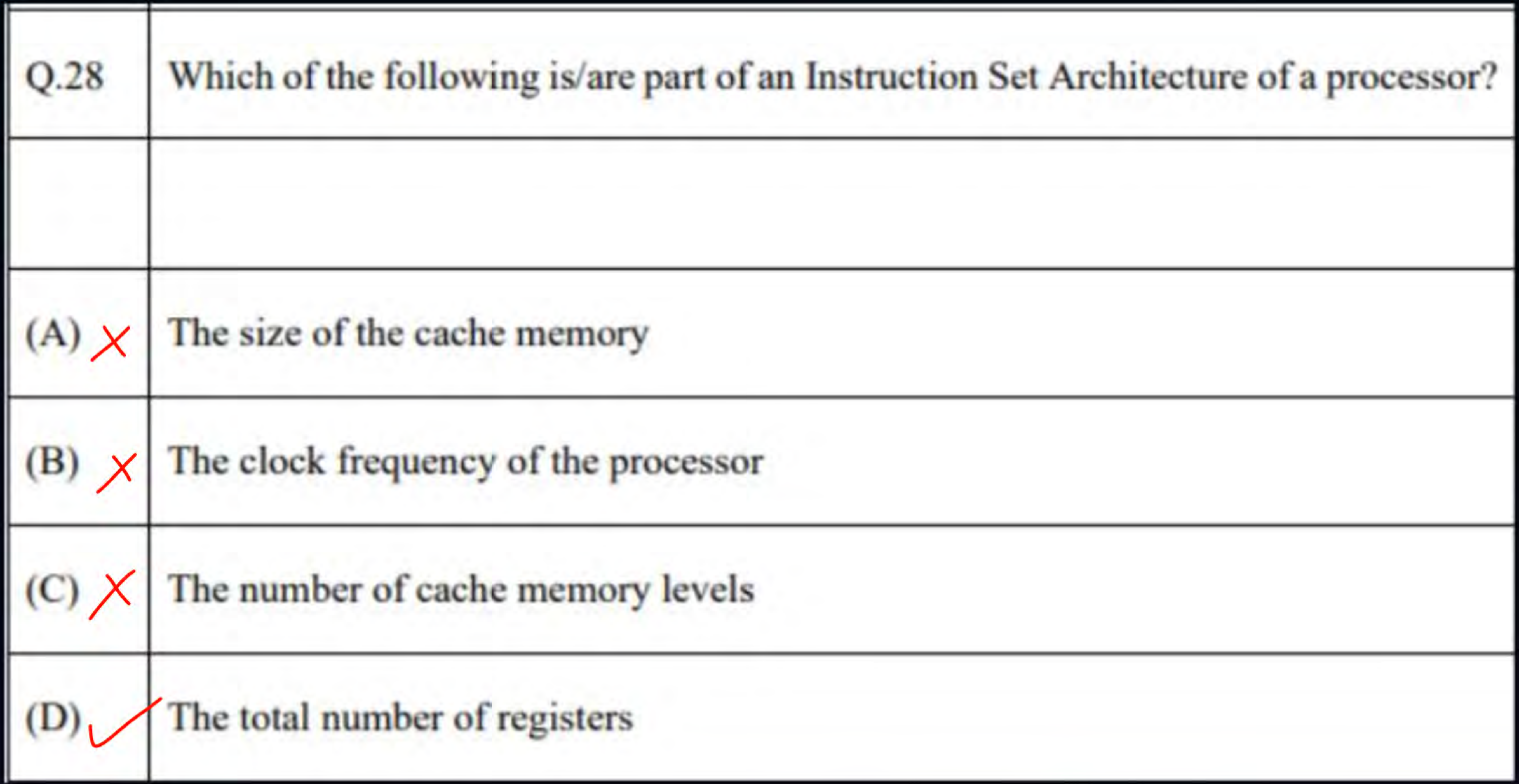
Topic : ISA → inst^{n} set architecture

→ Collection of all inst^{ns} supported by a CPU.

$$\left.\begin{array}{c}\text{size of ISA}\\ \text{or}\\ \text{size of inst}^{n}\text{ set}\end{array}\right\} \Rightarrow \text{no. of distinct inst}^{ns}\text{ supported by a CPU.}$$

MSQ GATE-2025

#Q.



Q.28	Which of the following is/are part of an Instruction Set Architecture of a processor?
(A) ✗	The size of the cache memory
(B) ✗	The clock frequency of the processor
(C) ✗	The number of cache memory levels
(D) ✓	The total number of registers

Topic : Types of Instruction

- 3-Address Instruction:
- 2-Address Instruction:
- 1-Address Instruction:
- 0-Address Instruction:

Topic : 3-Address Instruction

Max 3 addresses can be specified within an instruction for operands

	Opcode	Add. 1	Add. 2	Add. 3
ex:-	10101	11	10	01
	ADD	R3,	R2,	R1

$\underbrace{R3 \longleftarrow R2 + R1}_{\text{default}}$ or $R3 + R2 \longrightarrow R1$

Topic : 2-Address Instruction

Max 2 addresses can be specified within an instruction

opcode	add. 1	add. 2
10101	11	10
ADD	R3,	R2

$\underbrace{R3 \longleftarrow R3 + R2}_{\text{default}}$ or $R3 + R2 \longrightarrow R2$

Topic : 1-Address Instruction

Max 1 address can be specified within an instruction

opcode	add.1

10101 ↓ ADD

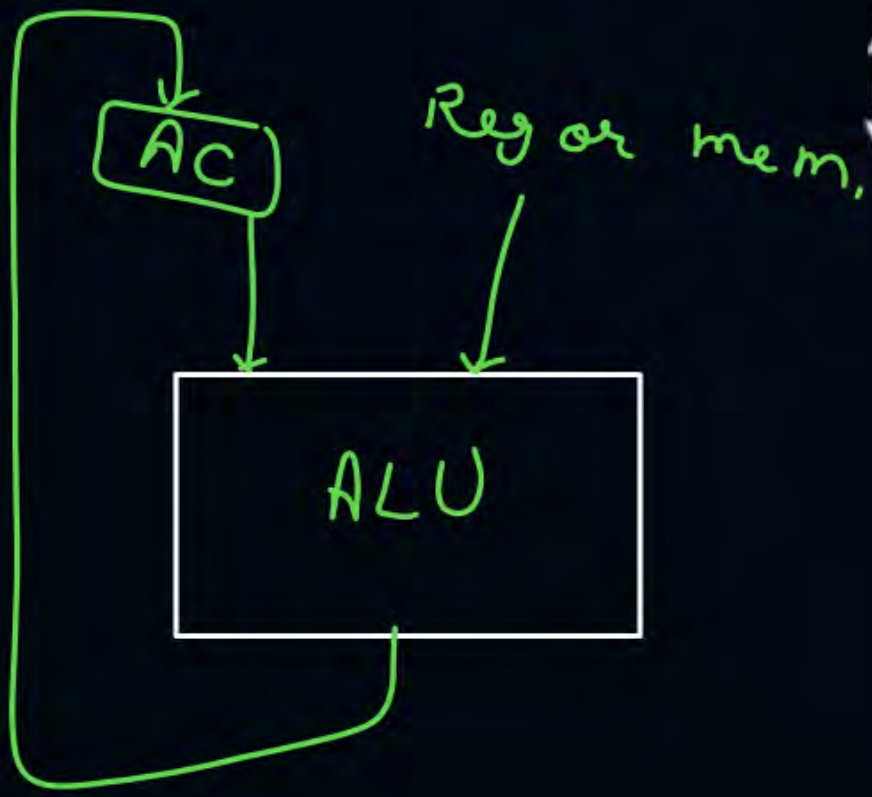
11 ↓ R3

ADD R3

⇓

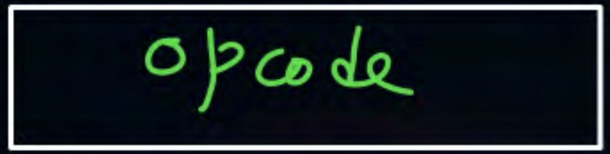
AC - based arch.

$$AC \leftarrow AC + R3$$



Topic : 0-Address Instruction

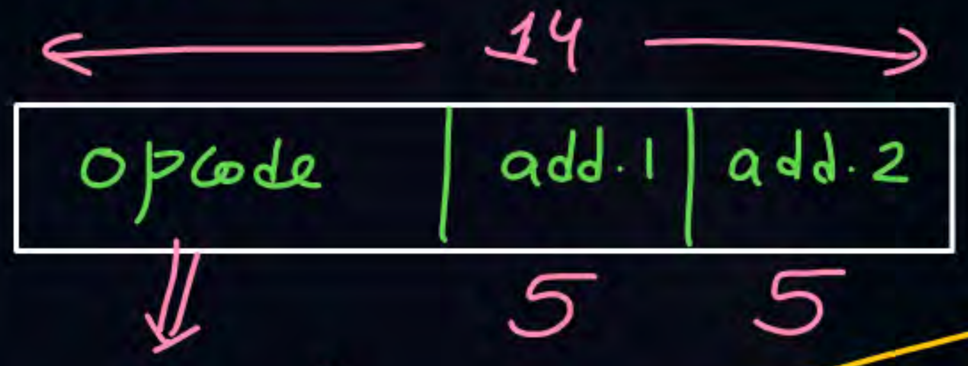
No any address can be specified within an instruction


opcode

[NAT]

#Q. Consider a digital computer which supports only 2-address instructions each with 14-bits. If address length is 5-bits then maximum and minimum how many instructions the system can support?

2-add. Inst^{ns}



$= 14-(5+5)$

$= 4$ bits

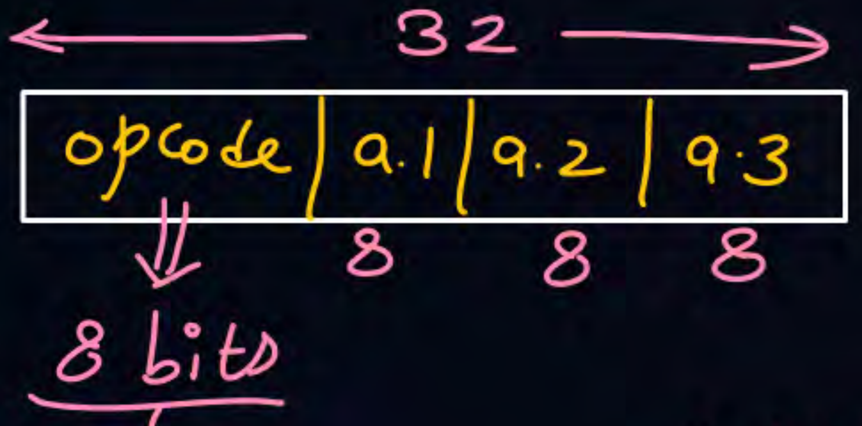
→ max no. of inst^{ns} supported $= 2^4 = 16$

min —— " —— $= 1$

[NAT]

#Q. Consider a digital computer which supports only 3-address instructions each with 32-bits. If address length is 8-bits then maximum and minimum how many instructions the system can support?

3- add. insts



$max = 2^8 = 256$

$min = 1$

[NAT]

#Q. Consider a digital computer which supports 64 2-address instructions. If address length is 9-bits then the length of the instruction is <u>24</u> bits?

no. of instⁿˢ supported = 64 ⇒ opcode = 6 bits

2-add. instⁿˢ

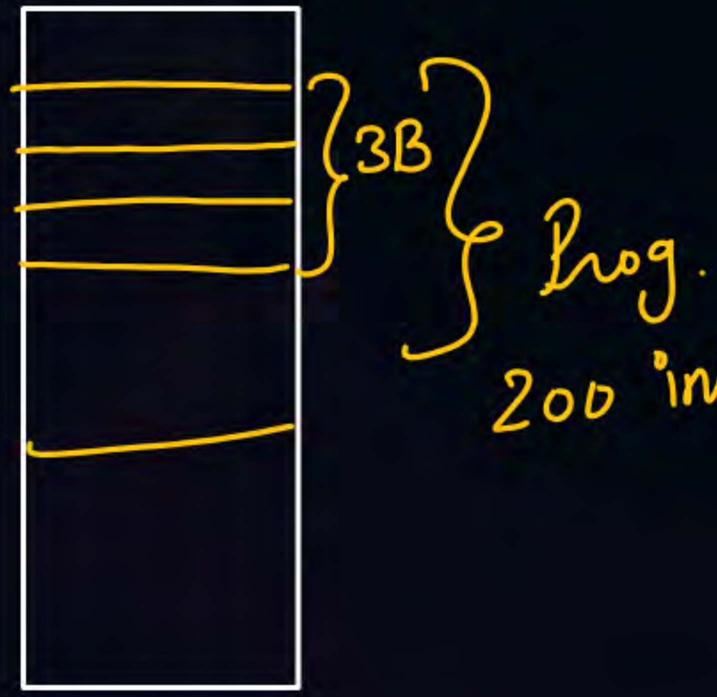
opcode	a.1	a.2
6	9	9

$\underbrace{6 \quad 9 \quad 9}_{24 \text{ bits}}$ = 3 bytes

[NAT]

#Q. Consider a digital computer which supports 64 2-address instructions. If address length is 9-bits then the length of the instruction is _24_ bits?

In above question: Each instruction must be stored in memory in a byte-aligned fashion. If a program has 200 instructions, then amount of memory required to store the program text is ____ bytes?

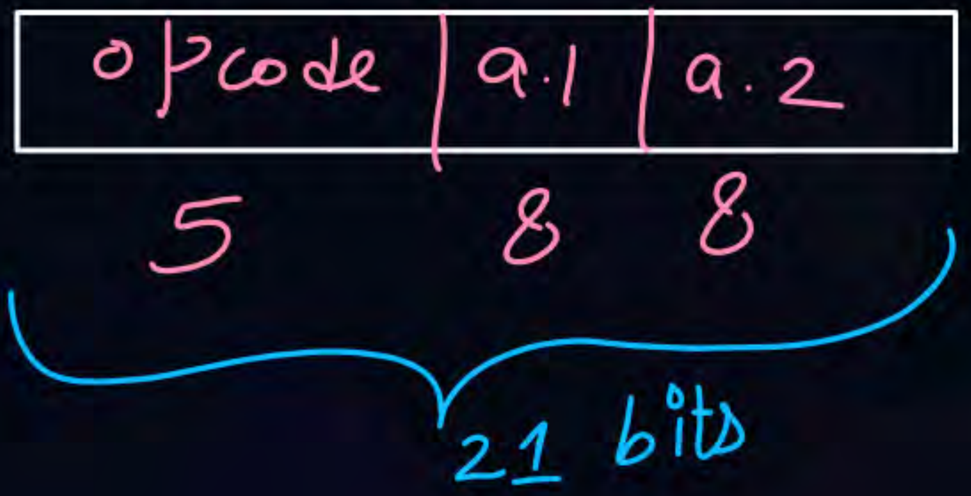


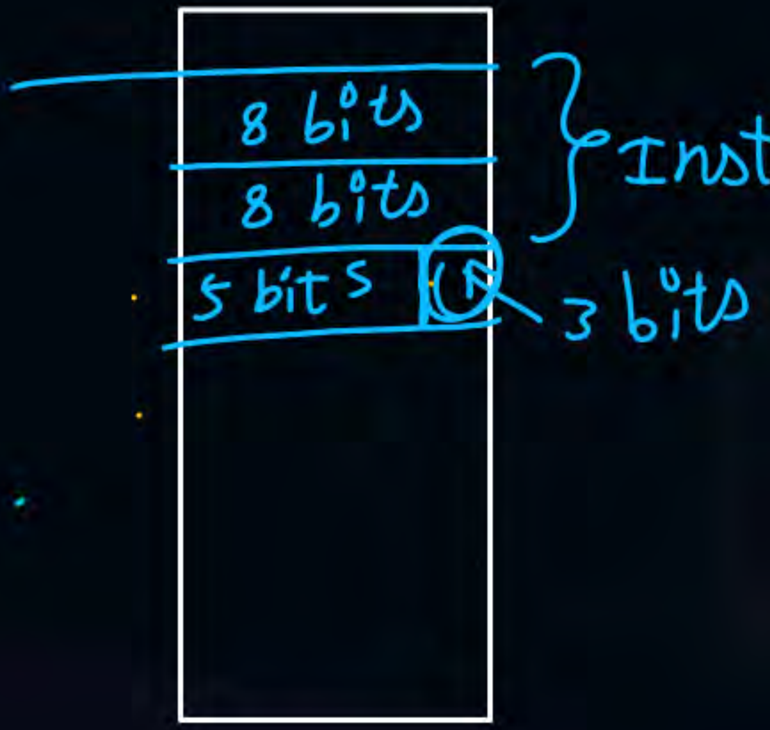
200 inst^{ns} = 200 * 3 = 600 B

[NAT]

#Q. Consider a digital computer which supports 32 2-address instructions. Consider the address length is 8-bits. Each instruction must be stored in memory in a byte-aligned fashion. If a program has 300 instructions, then amount of memory required to store the program text is 900 bytes?

21 bit but for storing instruction it will require 3 byte as new instruction can't be started from previous byte it requires new one.





$\text{Inst}^{ns} = 3$ bytes

$300\ \text{inst}^{ns} = 900\,B$

[NAT]

opcode = 6 bits

Reg. number = 4 bits

4 bit for nomenclature

#Q. A processor has 50 distinct instructions and 16 general purpose registers. Each instruction in system has one opcode field, 2 register operand field and a 10 bits memory address field. The length of the instruction is 24 bits?

opcode	Reg1	Reg2	mem. address
6	4	4	10

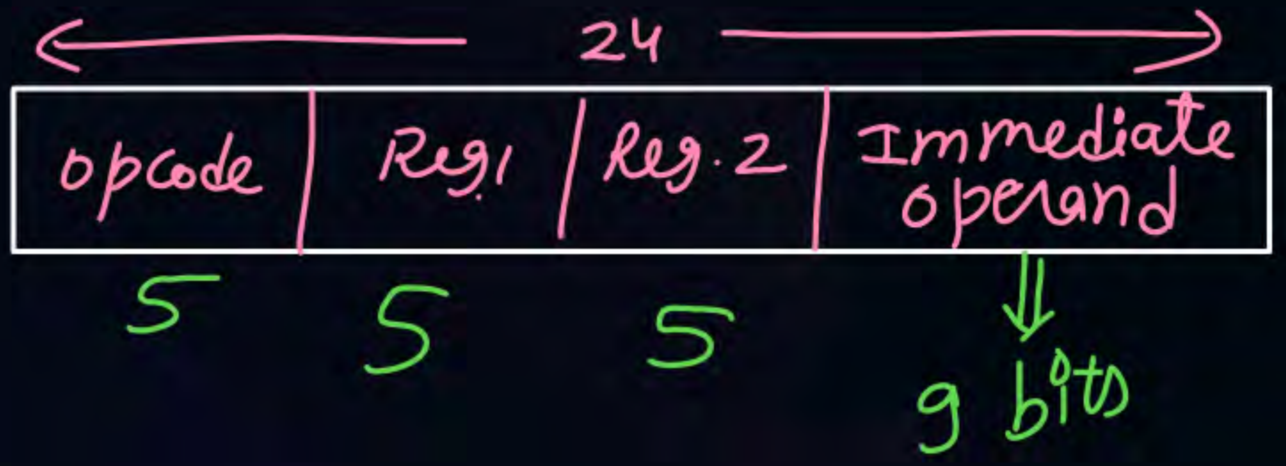
24 bits

[NAT]

opcode = 5 bits

Reg = 5 bits

#Q. A processor has 20 distinct instructions and 32 general purpose registers. A 24 bits instruction word has an opcode, two register operands and an immediate operand. The number of bits available for the immediate operand field is _9_?



opcode	Reg1	Reg.2	Immediate operand
5	5	5	⇓ 9 bits

[NAT]

#Q. A processor has 20 distinct instructions and 32 general purpose registers. A 24- bit instruction word has an opcode, two register operands and an immediate operand. The number of bits available for the immediate operand field is _9_?

In above question: Assume that immediate operand field is an unsigned number, What is its maximum and minimum value possible?

$\Rightarrow$ max $= 2^9 - 1 = 511$

min $= 0$

	min	max
Unsigned	0	$(2^n - 1)$
sign mag. 1^s comp	$-(2^{n-1} - 1)$	$+(2^{n-1} - 1)$
2^s comp.	-2^{n-1}	$+(2^{n-1} - 1)$

[NAT]

byte addressable

#Q. Consider a system which support only 3 address instructions only, and supports 256B memory. If the instruction size is 40-bits then maximum & minimum number of instruction supported by the system are?

← 40 →

opcode	a.1	a.2	a.3
16 bits	8	8	8

max = 2^{16}

min = 1

no. of cells = $\frac{256B}{1B} = 256 = 2^{8}$

add. = 8 bits

True/False

$R1 \leftarrow R2 + M[2000]$

$R3 \leftarrow R4 + R5$

addition ⇒ same opcode

opcode defines inst^n

Above two instructions treated as same type when counted into Instruction set architecture? True

True/False

False

32-bit architecture CPU has 1 word instruction, it means instruction size = 5B?

↓

32 bits = 4B

True/False

False

X-bit architecture computer means the address bus width is x-bits?

no relation

2 mins Summary

Topic Instructions

Topic Opcode

Happy Learning

THANK - YOU