COMPUTER SCIENCE

Computer Organization and Architecture

Machine Instruction and Addressing Modes

Lecture_03

Vijay Agarwal sir

01 Instruction Set Architecture

02 Expand Opcode Technique

Instruction

Instruction Format

Instruction Set

Questions & GATE PYQ's.

Instruction Format

← Instruction →

OPCODE	Address field.

OPCODE
(mnemonics)
operational
Code
⇓
Type of operation.

nbit ⇒ 2^n operation.

AF → Reg AF
AF → Mem AF

1MB $\left[\begin{matrix}2^{20}\\2^{B}\end{matrix}\right]$

Machine Instruction Characteristics

- The operation of the processor is determined by the instructions it executes, referred to as machine instructions or computer instructions
- The collection of different instruction that the processor can execute is referred to as the processor's instruction set **(ISA)**

Instruction Set = 20.

↳ 20 Distinct operation performed ⇒ opcode = 5 bit

$\lceil \log_2 20 \rceil$
⇓
5

Instruction Format

Instruction Format

OP : OPCODE
S_1 : Source 1
S_2 : Source 2
Destn : Destination

Gen-Reg.

3AI (or) 3AF

OPCODE	AF_1	AF_2	AF_3

: ADD | R_1 | R_2 | R_3 ; $R_1 \leftarrow R_2 + R_3$

(OP, Destn, S_1, S_2)

2AI (or) 2AF

OP	AF_1	AF_2

: ADD | R_1 | R_2 ; $R_1 \leftarrow R_1 + R_2$

(Destn, S_1, S_2)

AC-CPU

1AF (or) 1AI

OP	AF_1

ADD | R_1 ; $AC \leftarrow AC + R_1$

(OP, AF)

Stack-CPU

0AI (or) 0AF

OPCODE

ADD

ADD

Instruction Set Architecture Classification

1. Stack organization.
2. Single accumulator organization.
3. General register organization.

Instruction Set Architecture

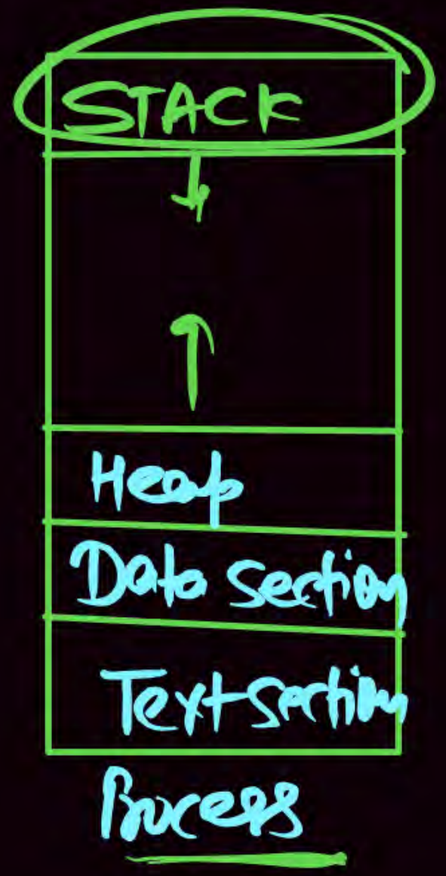
CPU ~~ALP~~ Organization is classified into 3 types based on the availability of ALU Operand (Data) (AF: Address field or AI: Address Instruction)

1. Stack-CPU **[0AF]**
2. Accumulator-CPU **[1AF]**
3. General Register organization
   i. Reg-Memory reference CPU **[2AF]**
   ii. Reg-Reg reference CPU **[3AF]**

① Stack Organization [STACK - CPU].

- In the Stack Based organization ALU operations are Performed only on Stack Data.
- So Both the operand (Data) must be Required in the Stack & After Processing Result is also Stored in the Stack.
- STACK is Part of Memory which is intialized by (STACK Pointer) SP Register.
- In stack Insert & Delete operation are performed at the [LIFO] Same end Called TOS [Top of the Stack] so its become default location.

In OS



① Stack Organization

Compitable format

OPCODE

⇓

Type of operation.

ALU operation

Destination Source1 Source2

TOS ← TOS + TOS

eg A+B

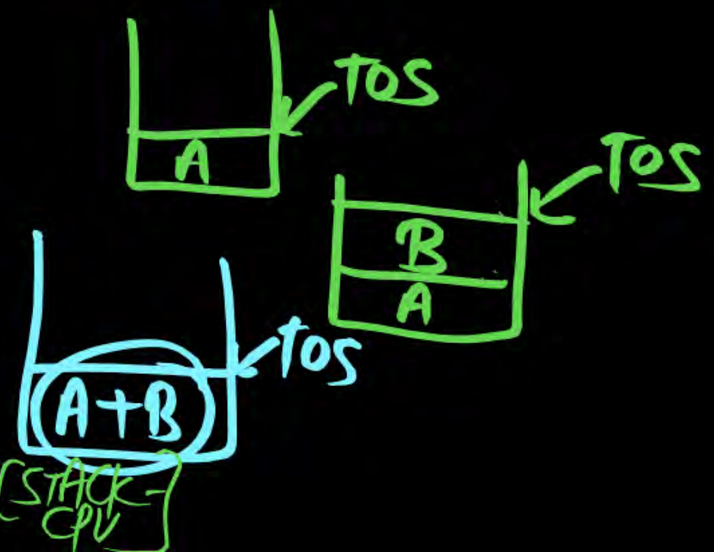
POP
POP
ADD (ALU operⁿ)
PUSH

I_1: PUSH A

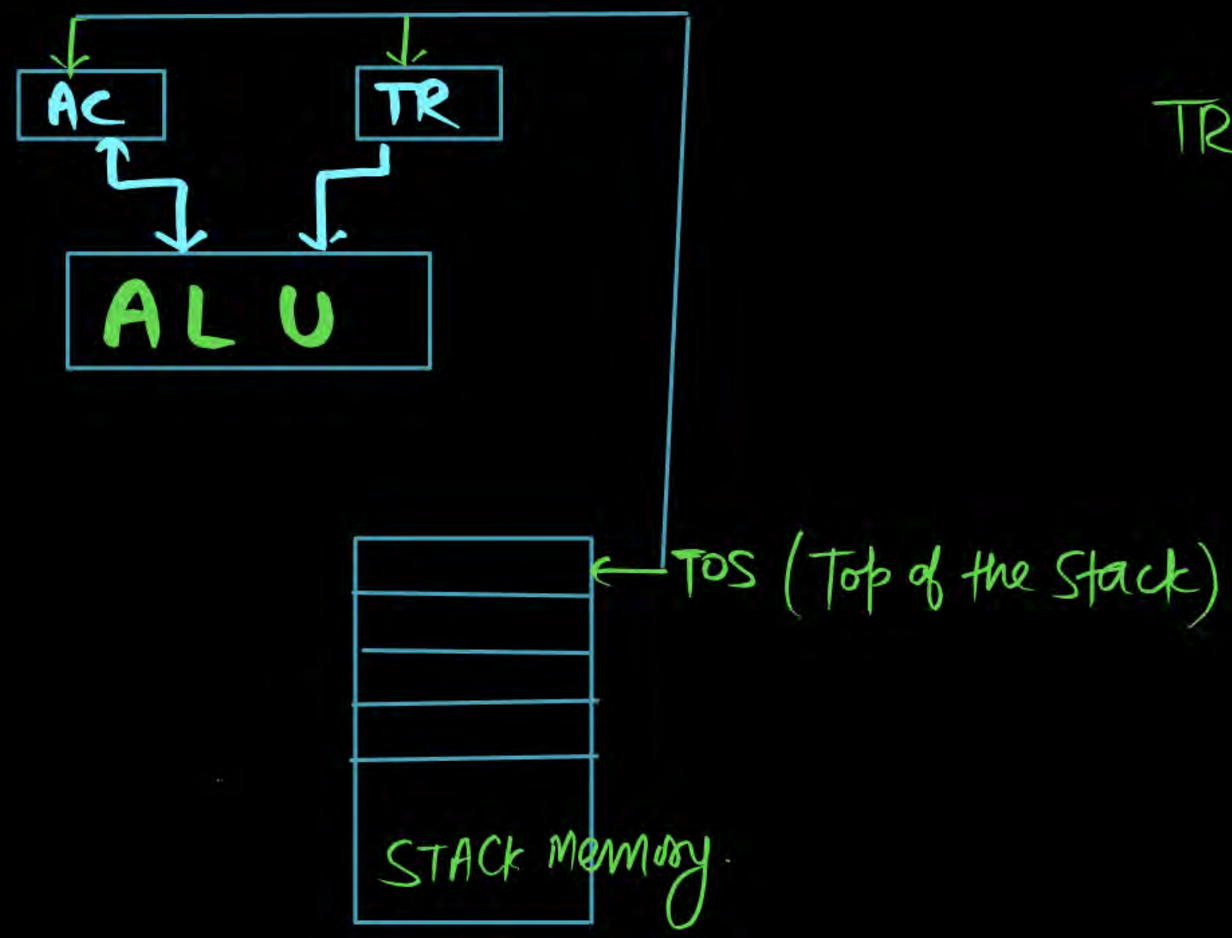
I_2: PUSH B

I_3: ADD

3 Machine Instⁿ (STACK-CPU)



① Stack Organization



TR: Temporary Register.

PUSH
POP
} Stack specific → Insert
→ Delete.

LOAD
STORE
} Memory Specific → Memory Read
→ Memory Write

IN
OUT
} IO Specific → IO Read
→ IO Write

'MOV' : General with all Combination.

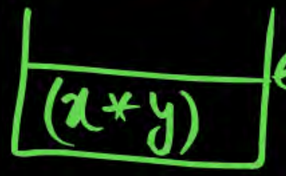
Stack Organization

eg $\boxed{(x*y)+z}$ How Many machine Instruction are Required to execute Using Stack Based org (STACK-CPU)?

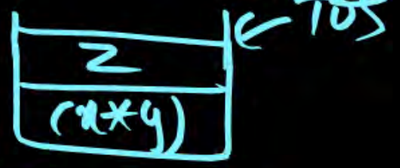
I_1: PUSH x

I_2: PUSH y

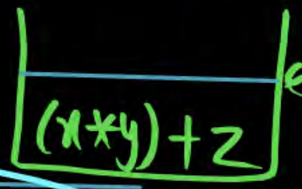
I_3: MUL



I_4: PUSH z



I_5: ADD



5 Machine Instn Using Stack CPU.

Stack Organization

Note

In the STACK-CPU, Only ALU Operation are OAI & Data transfer Instruction [PUSH & POP] are Not OAI. [Zero Address Instruction].

Stack Organization

Q.1, Q.2, Q.3.

Q.1 to 3 Consider a 32 bit Hypothetical Processor which use STACK-CPU. Which support 1 Word opcode and 24 bit address following statement is executed on a STACK-CPU (Stack is Initially Empty)

$$X = (A + B) \times (C + D)$$

Word Size = 32bit

OPCODE = 1Word
= 32bit ≈ 4Byte

Address = 24bit ≈ 3Byte.

Q.1 How many Machine Instruction are required using Stack-CPU?

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

I_1: PUSH A

I_2: PUSH B

I_3: ADD

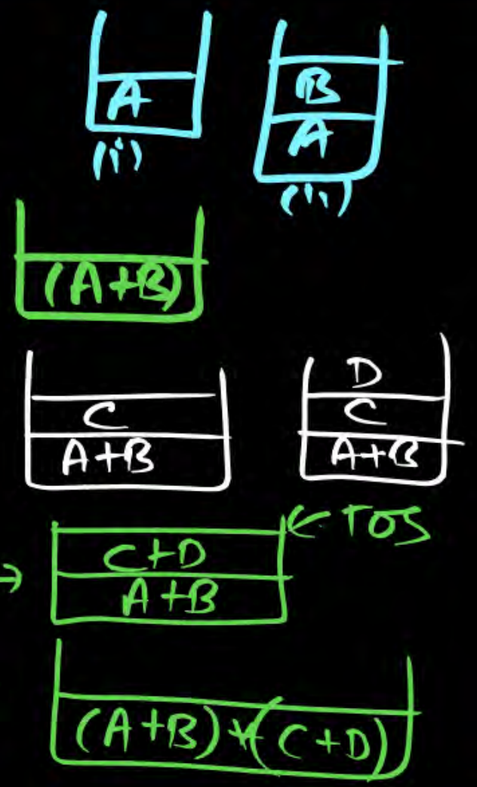
I_4: PUSH C

I_5: PUSH D

I_6: ADD

I_7: MUL

I_8 POP X



$M[X] \leftarrow (A+B) * (C+D)$

8 m/c Instn Using STACK-CPU.

Soln

8 m/c Instn Using STACK-CPU.

Q.1 How many Machine Instruction are required using Stack-CPU?

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

I_1: PUSH A

I_2: PUSH B

I_3: ADD

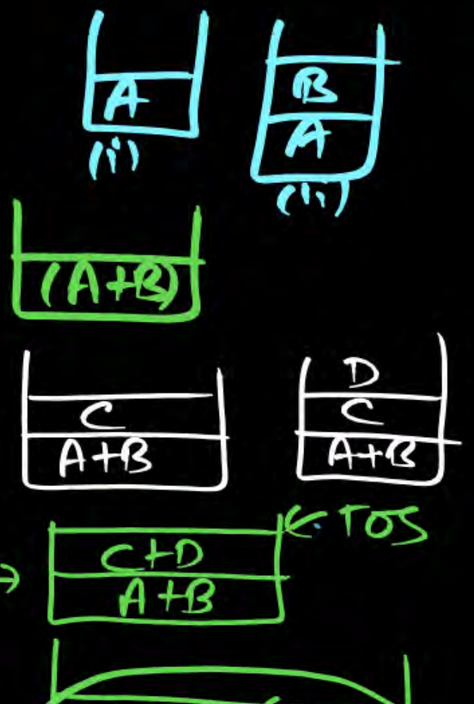
I_4: PUSH C

I_5: PUSH D

I_6: ADD $\longrightarrow$

I_7: MUL $\longrightarrow$

I_8 POP X



$M[X] \leftarrow (A+B) * (C+D)$

Soln 1

8 m/c Instn Using STACK - CPU.

Q.2 How much Memory space is required for the program in Byte?

$\boxed{X = (A+B) * (C+D)}$

OPCODE AF

OPCODE: 4B, A.F = 3Byte.

Soln 2 47 Byte.

OPCODE AF.

I_1: PUSH A

I_2: PUSH B

I_3: ADD

I_4: PUSH C

I_5: PUSH D

I_6: ADD

I_7: MUL

I_8: POP X

I_1: 4B + 3B = 7 Byte

I_2: 4B + 3B = 7 Byte

I_3: 4B = 4 Byte

I_4: 4B + 3B = 7 Byte

I_5: 4B + 3B = 7 Byte

I_6: 4B = 4 Byte

I_7: 4B = 4 Byte

I_8: 4B + 3B = 7 Byte

47 Byte Ans

Q.3 What is the status of the Stack at the end of program execution?

Sol^n 3

Empty. Bcz Result is Stored in M[X].

Stack Organization

$X = (A + B) \times (C + D)$

I_1:	PUSH	A	TOS ← A
I_2:	PUSH	B	TOS ← B
I_3:	ADD		TOS ← (A + B)
I_4:	PUSH	C	TOS ← C
I_5:	PUSH	D	TOS ← D
I_6:	ADD		TOS ← (C + D)
I_7:	MUL		TOS ← (C + D) × (A + B)
I_8:	POP	X	M[X] ← TOS

8 Machine Instruction Required (Stack-CPU)

② Single Accumulator Organization

- In Accumulator Based org. first ALU operand are Required in Accumulator, Second ALU operand (DATA) either Present in Register (or) in Memory.
- After the Processing Result is also Stored in the Accumulator (OR) Accumulator is Used as a Destination.
- In the Processor. Only 1 [One] Accumulator is Present, So Not Need to explict Mention the address.

② Single Accumulator Organization

Compitable format:

OPCODE	AF1

Type of operation

① ALU operation

Destination	Source1	Source2
AC	AC	Reg/Mem

(ii) Data transfer operation

Destination	Source	
AC	Mem	[memory Read] [LOAD]
Mem	AC	[memory write] [STORE]

ALU operation.

eg1 ADD R_1

$AC \leftarrow AC + R_1$

[Destn] AC | [S1] AC | [S2] Reg

eg2 ADD [6000]

$AC \leftarrow AC + M[6000]$

[Destn] [AC] | [S1] AC | [S2] mem

Data transfer operation

eg1 LOAD [7000]

$AC \leftarrow M[7000]$

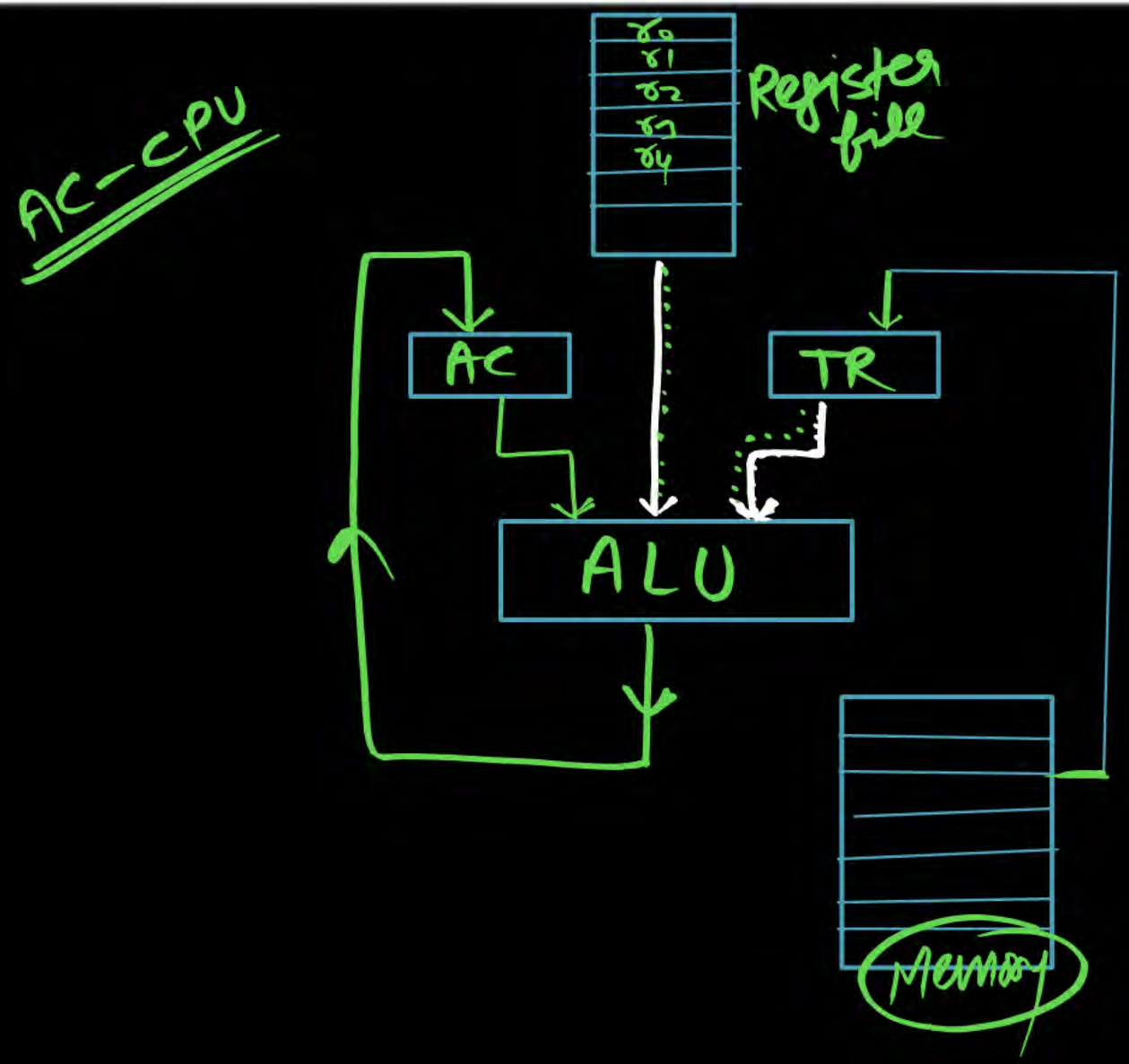
Destn [AC] | Source (mem)

eg2 STORE [9000]

$M[9000] \leftarrow AC$

(Destn) | Source

Single Accumulator Organization



Memory ⇒ Main Memory.

P.A.S (or) Physical memory

L.A.S (or) Logical Memory

Register file : The Number of Registers Supported by the Processor is denoted [called] Register file.

(eg) Register file = 16
Contain 16 Registers.

Single Accumulator Organization

Q.1 (A+B) A & B are the variables in the Memory. How Many M/c Instn Required to execute Using AC - CPU ?

I_1: LOAD A; $AC \leftarrow M[A]$

I_2: ADD B; $AC \leftarrow AC + M[B]$.

---

2 m/c Instn Using AC-CPU.

Single Accumulator Organization

Q.2 $(x*y)+z$. How Many M/c Instn Required Using AC-CPU?

I_1 : LOAD x; $AC \leftarrow M[x]$

I_2 : MUL y; $AC \leftarrow AC * M[y]$

I_3 : ADD z; $AC \leftarrow AC + M[z]$

$\boxed{AC \leftarrow x * y}$

$AC \leftarrow [x*y]+z$.

3 machine Instn Using AC-CPU.

$\boxed{X = (A+B) * (C+D)}$ A, B, C, D & X are variable in the memory.

Q.1 How many M/c Instn Required to execute Using AC-CPU?

Q.2 How Many Spills (memory Spills) are Required?

Soln 1

I_1 :	LOAD A;	$AC \leftarrow M[A]$	
I_2 :	ADD B;	$AC \leftarrow AC + M[B]$	$AC \leftarrow A+B$
I_3 :	STORE T;	$M[T] \leftarrow AC$	$M[T] = (A+B)$
I_4 :	LOAD C;	$AC \leftarrow M[C]$	
I_5 :	ADD D;	$AC \leftarrow AC + M[D]$	
I_6 :	MUL T;	$AC \leftarrow AC * M[T]$	
I_7 :	STORE X;	$M[X] \leftarrow AC$;	$M[X] = (A+B) * (C+D)$

Ans 1 7

Ans 2 1

↳ Spills (To Store Intermediate Result in Temporary Memory is called Spills.)

7 M/c Instn Using AC-CPU.

Single Accumulator Organization

$$X = (A + B) \times (C + D)$$

I_1:	LOAD	A	$AC \leftarrow M[A]$
I_2:	ADD	B	$AC \leftarrow AC + M[B]$
I_3:	STORE	T	$M[T] \leftarrow AC$
I_4:	LOAD	C	$AC \leftarrow M[C]$
I_5:	ADD	D	$AC \leftarrow AC + M[D]$
I_6:	MUL	T	$AC \leftarrow AC \times M[T]$
I_7:	STORE	X	$M[x] \leftarrow AC$

7 Machine Instruction Required (AC-CPU)

General Register Organization

Based on the Number of Register (Register file size) Supported by the Processor this Arch. is divided into 2 Types.

1. Register – memory Reference [Reg-mem CPU]
2. Register – Register Reference [Reg-Reg CPU].

HOME - WORK.

General Register Organization

① Register - Memory Reference [2AF]

(A+B)

'MOV'

(x*y) + z

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

Destn	S1	S2
Source 1 (Reg)	Reg	Reg/mem

① ADD R_1 R_2

$R_1 \leftarrow R_1 + R_2$

② ADD R_1 [6000]

$R_1 \leftarrow R_1 + M[6000]$

General Register Organization

(RISC)

① Register - Register Reference. [2AF]

Destn	S1	S2
Reg	Reg	Reg.

(A+B)

(x*y) + z

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

① ADD R_1 R_2 R_3

$R_1 \leftarrow R_2 + R_3$.

General Register Organization

$X = (A + B) \times (C + D)$

I_1:	MOV	R1, A	$R1 \leftarrow M[A]$
I_2:	ADD	R1, B	$R1 \leftarrow R1 + M[B]$
I_3:	MOV	R2, C	$R2 \leftarrow M[C]$
I_4:	ADD	R2, D	$R2 \leftarrow R2 + M[D]$
I_5:	MUL	R1, R2	$R1 \leftarrow R1 \times R2$
I_6:	MOV	X, R1	$M[X] \leftarrow R1$

6 Machine Instruction Required (Reg-CPU)

RISC Instructions

$X = (A + B) \times (C + D)$

LOAD	R1, A	$R1 \leftarrow M[A]$
LOAD	R2, B	$R2 \leftarrow M[B]$
LOAD	R3, C	$R3 \leftarrow M[C]$
LOAD	R4, D	$R4 \leftarrow M[D]$
ADD	R1, R1, R2	$R1 \leftarrow R1 + R2$
ADD	R3, R3, R2	$R3 \leftarrow R3 + R4$
MUL	R1, R1, R3	$R1 \leftarrow R1 \times R3$
STORE	X, R1	$M[X] \leftarrow R1$

Note:

Immediate field is n bit

Unsigned Range = $(0 \text{ to } 2^n - 1)$

Signed Range = $-(2^{n-1})$ to $+ (2^{n-1} - 1)$

Example

If immediate field is 4 bit

Then unsigned range = $(0 \text{ to } 2^4 - 1) \Rightarrow 0 \text{ to } 15$

Signed Range = $-(2^{4-1})$ to $+ (2^{4-1} - 1)$

Q.

A machine has a 32-bit architecture, with 1-word long instructions. It has 64 registers, each of which is 32 bits long. It needs to support 45 instruction, which have an immediate operand in addition to two register operands. Assuming that the immediate operand is an unsigned integer the maximum value of the immediate operand is _____. **[GATE-2014 (Set-1)]**

Q.

A processor has 40 distinct instructions and 24 general purpose registers. A 32-bit instruction word has an opcode, two register operands and an immediate operand. The number of bits available for the immediate operand field is _____.

[GATE-2016 (Set-2)]

Q.

Consider a processor with 64 registers and an instruction set of size twelve. Each instruction has five distinct fields, namely, opcode, two source register identifiers, one destination register identifier, and a twelve-bit immediate value. Each instruction must be stored in memory in a byte-aligned fashion. If a program has 100 instructions, the amount of memory (in bytes) consumed by the program text is _____.

[GATE-2016 (Set-2): 2Marks]

Expand Opcode Technique

Expand Opcode Technique

Expand opcode length is required in the fixed length instruction supported CPU Design to implement the various instruction with different formats.

Variable Length Instruction Supported CPU Design

OPCODE = 8 bit

Address field = 8 bit

(i) 1 Address Instruction Design:

← Instruction →

OPCODE	Address
8 bit	8 bit

= 16 bit

(ii) 0 Address Instruction Design:

← 8 bit →

OPCODE

= 8 bit [No need of expand opcode technique]

← Instruction →

Fixed Length Instruction Supported CPU Design

OPCODE = 8 bit

A.F = 8 bit

(i) 1 Address Instruction Design:

← Instruction →

OPCODE	Address
8 bit	8 bit

= 16 bit

(ii) 0 Address Instruction Design:

← Instruction →

OPCODE

← 16 bit →

= 16 bit [Here expand opcode technique required]

Expand Opcode Technique

- Primitive instruction means smallest opcode instruction.

Step 1: Identify the primitive instruction in the CPU.

Step 2: Calculate the total number of possible operation.

Step 3: Identify the free opcode after allocating the existed instruction

Step 4: Calculate the number of Derived instruction possible by multiply

$$\text{Free opcode} \times 2^{\text{Increment bit in opcode}}$$

Q.

Consider a processor which support 6 bit instruction and 4 bit address field. If there exist 2 one address instruction then how many 0 address instruction can be formulated?

Q.

Consider a processor which contain 8 bit word and 256 word memory. It support 3 word instruction. If these exist 254 2-address instruction and 256 1-address instruction then how many 0 address instruction can be formulated?

Q. Consider a processor with 16 bit instruction. Processor has 15 registers and support 2 address instruction and 1 address instruction. If processor support 256 1-address instruction then number of 2-address instruction are

(A) 128

(B) 192

(C) 240

(D) 248

Solution(c): 240

15 register $= 2^4 \Rightarrow$ Register A.F = 4 bit

16 bit

OPCODE	Register A.F	Register A.F
8 bit	4 bit	4 bit

OPCDE field $= 16 - (4 + 4) = 8$ bit

So total number of 2 address instruction $= 2^8 = 256$

Let 'x' 2 address instruction used

Number of free opcode $= (2^8 - x)$

1 Address field

OPCODE	A.F
12 bit	4 bit

Total no of 1 address instruction = $(2^8 - x) \times 2^{12-8}$

$[2^8]256 \Rightarrow (2^8 - x) \times 2^4$

$2^4 = 2^8 - x$

$x = 2^8 - 2^4 \Rightarrow 256 - 16$

$= 240$

Q.

A processor has 16 register (R0, R1, ..., R15) and 64 floating point registers (F0, F1, ... F63). It uses a 2-byte instruction format. There are four categories of instructions: Type-1 Tpye-2 Type-3 and Type-4. Type-1 category consists of four instructions, each with 3 integer register operands (3Rs) Type-2 category consists of eight instructions, each with 2 floating point register operands (2fs). Type-3 category consist of fourteen instructions, each with one integer register operand and one floating point register operand (1R + IF). Type-4 category consists of N instructions, each with a floating point register operand (FR). The maximum value of N is _____.

[GATE-2018 : 2 Marks]

Q.

A processor has 64 registers and uses 16-bit instruction format. It has two types of instructions: I-type and R-type. Each I-type instruction contains an opcode, a register name, and a 4-bit immediate value. Each R-type instruction contains an opcode and two register names. If there are 8 distinct I-type opcodes, then the maximum number of distinct R-type opcodes is _____.

[GATE-2020 : 2 Marks]

THANK YOU!