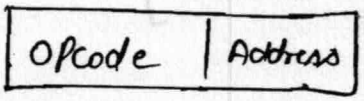
(b) Fixed Length Instⁿ supported CPU Design :-

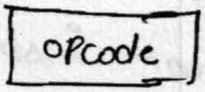
* Opcode size = 8 bit
* Address size = 8 bit

(1) 1-Address instⁿ Design :-



8 bit 8 bit = 16 bit

(2) 0-Address instⁿ Design :-



16 bit = 16 bit → { Expand opcode Technique }

Que: In a Hypothetical CPU, Instⁿ set size is 24. Instⁿ is designed with a 2 Register operands and 1 M/y operand. CPU supports 32 Registers and 2^{28} B main-memory space. How many bits are required to encode the instruction (Instruction size).

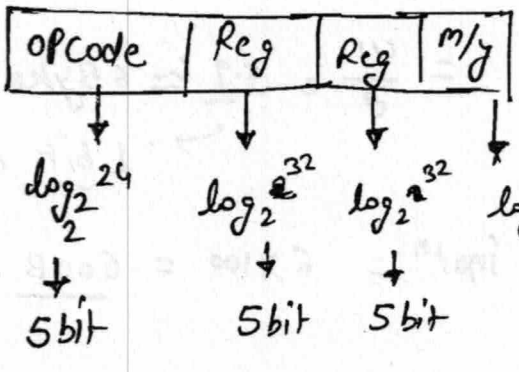
Sol^n:

** { # of instⁿ in CPU } = Instⁿ set = 24

# of Register = 32

M/y size = 2^{28} B.

Instⁿ design :

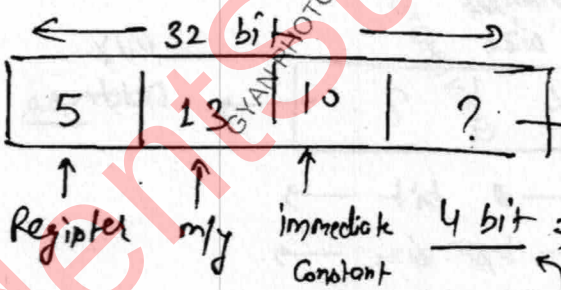


Que :- Consider a 16 bit Hypothetical processor which supports 2 word instructions with 1 Register operand, 1 m/y operand and 16 bit immediate Constant. Processor supports 20 Registers and 8 kB m/y. How many Number of instⁿ are possible in the CPU ?

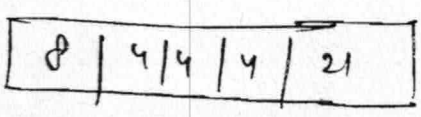
Solⁿ :

32 bit ← 2 word

Register = 20



* Que :- Consider a Hypothetical CPU which supports Instruction design with 8 bit opcode, 3 Register Address fields and 21 bit immediate constant. In the CPU Register file size is 16. Program contain 100 instⁿ. How much m/y space is required to store Program in Bytes.



5.1 → 6B

= 6 × 100 = 600B

In CPU,

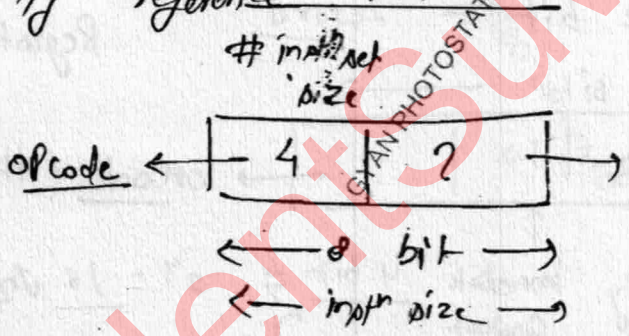
Instⁿ size $= \frac{41}{8} = 5.1 \approx 6$ Bytes

↳ 1 bit cell not available

1 Byte cell available so 6 cells

So for 100 instⁿ $= 6 \times 100 = 600B$ Ans.

Que:- Consider A Hypothetical CPU supports the instⁿ format with 2 fields that is opcode and address field refer as a 1 Address instruction. CPU supports 8 bit Instⁿ with a instruction set size of 16. How much main memory possible in the CPU =? when the instⁿ is designed as m/y reference instⁿ



Que:- A Hypothetical Processor supports 8 bit Instⁿ and also ~~8~~ support 5 bit Address If there exist 6 one Address Instⁿ then How many 0 Address instⁿ are formulated.

Steps:-

① Identify the High order Instⁿ format in the CPU

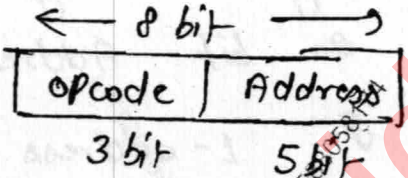
② Identify the total # operation (Instⁿ) possible.

③ Identify the # of free opcodes after allocate the existed Instⁿ.

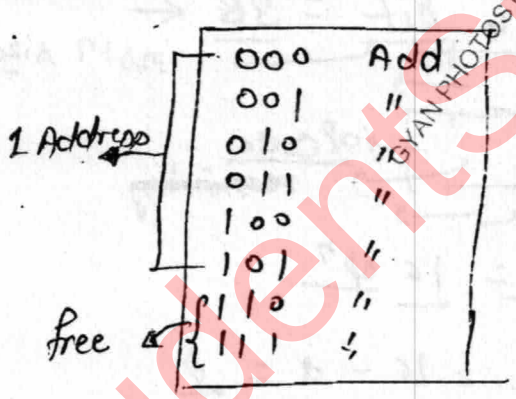
④ calculate the derived opcodes by multiply the free opcodes with a decoded form of a Address field.

:-

1)



2) # of opⁿ (n) = 2^opcode = 2^3 = 8 opcode

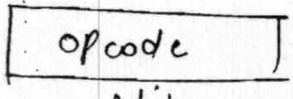


3) # of free opcodes = N − # of existed Instⁿ

= 8 − 6

= 2

4) # of zero Address Instⁿ Possible.



⇒ # free opcode * $2^{\text{Address size}}$ ⇒ $2^3 * 2^5$ = 64

110 00 0000 } 32
110 000 001
110 111 111

111 000 00 } 32
111 111 11
= 64

⇒ | # free opcodes * 2^address size |

Size of instⁿ set :-

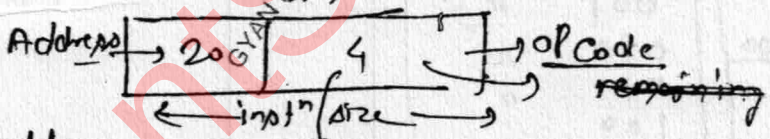
6 — 1 bit instⁿ

64 — 0 bit instⁿ

70 instⁿs

Que:- Consider a hypothetical Processor which support 1-word instructions. CPU interfaced with 3 cells of a m/y space. which contain the 20 bit Address space. If there exist 8 1-address instⁿ then How many 0-address instⁿ formulated.

word size = 24 bit = 3B ← instⁿ size



# of 1-address instⁿ = 8 ∴ 2^4 = 16 opⁿ

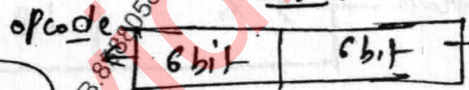
∴ free opcode = 16 − 8 = 8

# of 0-instⁿ formulation → 8 * 2^{20} = 2^23

(8 ↑ free opCode; 2^{20} Address)

* Que:- In a Hypothetical CPU, Instⁿ format contain 2 fields that is 6 bit opcode and 6 bit address field. In the opcode field 4 opcodes are reserved for special instⁿ. When these opcode patterns are present in the Instⁿ then Decode the special instⁿ by merging the address field to a opcode field. How many numbers of special instⁿ are present in the CPU and also report the size of a instⁿ set.

Solⁿ:-



2^6 = 64 opcodes

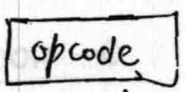
Reserved 4 opcodes are reserved for special instⁿ

60 opcodes used for old Instⁿ (Primitive Instⁿ)

$\Rightarrow (4) * 2^6$

$\Rightarrow 256$

Special instⁿ



$\Rightarrow 4 \times 2^6$

$\Rightarrow$ 256 instⁿ Special instⁿ

# of instⁿ

= 256 + 60

= 316 instⁿ

Size of instⁿ set.

Que:- Consider a Hypothetical CPU which supports both 1-address and 0-address instⁿ.

Instⁿ size is 2w long. with a word length of 12 bit. Processor supports 256 KB m/y.

a) what is the Range of 1-address and 0-address instⁿ possible.

Solⁿ:-

Instⁿ size = 2w = 2 × 12 = 24 bit

← 24 bit →

opcode → 6	18 → Address

~~1-address :- 1 * 2^18 2^6 = 64 instⁿ~~

~~0-address : 2^6 * 2^18 = 2^24~~

# of opⁿ (Instⁿ) = $2^6 = 64$

share the opcodes :-

	1-Address		0-address
1)	1	—	63
2)	63	—	1

Range of a 1-Address instⁿ : [opcode (6 bit) | Add. (18 bit)] {1 – 63}

Range of a 0-Address instⁿ : [opcode (24 bit)] : $\{(1 * 2^{18})$ to $(63 * 2^{18})\}$

Date 11/08/2017

"General Register CPU" :-

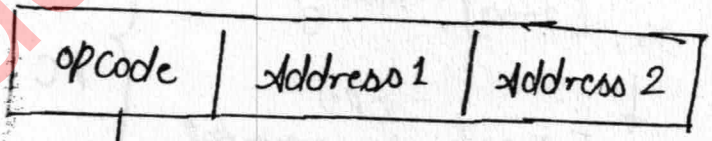
Baped on the Number of Registers present on the CPU, Architecture is divided into two types :-

i) Register - memory Reference Architecture (CPU with less # Register)
ii) Register - Register Reference Architecture (CPU with more # Register)

Register - memory Reference CPU :-

★ In this organization, 1st operand is always required in the Register and the 2nd operand is present either in the Register or in the memory. After the Data processing 'Result is placed into a Source 1 location'

★ Instn Design :-



2 Address Instn

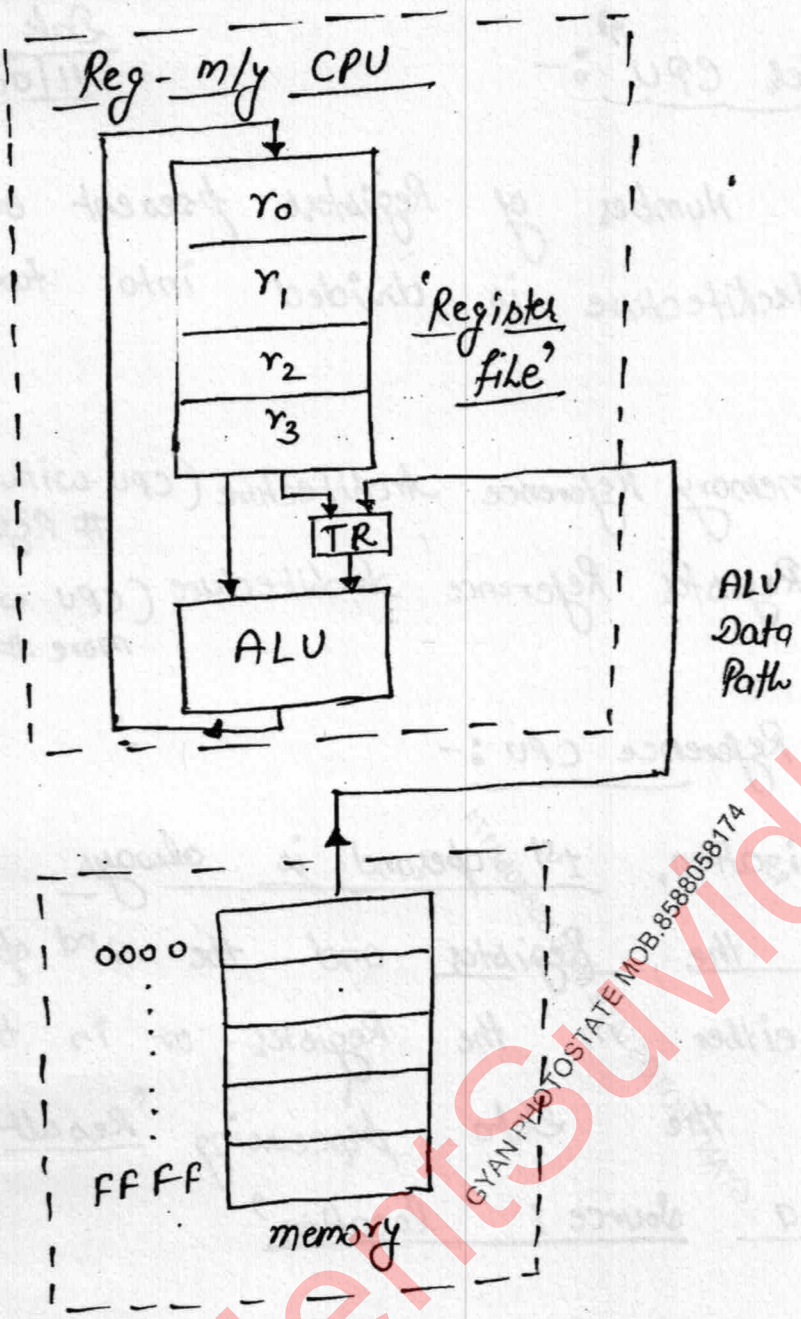
Type of opn

Data Transfer opn ;

Source	Destn
Reg.	Reg
Reg.	mem
m/y	Reg

} "mov" (Data Transfer Instn)

ALU opn ; S1 S2 D



Program :-

~~Address~~

Add r0, [2000] ; memory

Add r0, [2000] ; CPU PC: sequential address

FC

| Add | r0 | [2000] | ; Inst^{n} format

ALU op^{n} — SL & Dest^{n} — S_2

EC

eg:- $(A * B) + C$

variables are in the memory.

I_1: MOV r_0, A ;

$r_0 \leftarrow m[A]$

I_2 : MUL r_0, B ;

$r_0 \leftarrow r_0 * m[B]$

I_3 : Add r_0, C ;

$r_0 \leftarrow r_0 + m[C]$

Over

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

variables are in the m/y.

I_1 : MOV r_0, A ; $r_0 \leftarrow m[A]$

I_2 : ADD r_0, B ; $r_0 \leftarrow r_0 + m[A]$

I_3 : MOV r_1, C ; $r_1 \leftarrow m[C]$

I_4 ; ADD r_1, D ; $r_2 \leftarrow r_1 + m[D]$

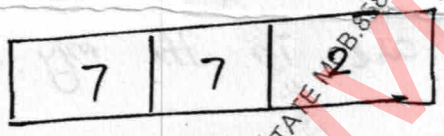
I_5 ; MUL r_0, r_1 ; $r_0 \leftarrow r_0 * r_1$

I_6 ; MOV x, r_0 ; $x \leftarrow r_0$

Que:- Consider a Hypothetical processor which supports 2 word instruction. word length of a CPU is 8 bit. Insth is placed in a 128 word memory. If there exist 2, 2-Address insth, 200, 1-Address instructions then How many 0-address Instructions are formulated ?

Instruction size = 2word = 16 bit

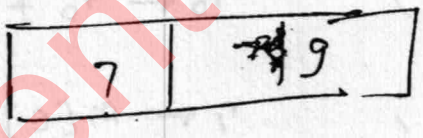
M/y size = 2^7 word



- 2 address

$2 * 2^{14}$

4-2 = 2



- 1-address

$2^9 = 512$

-200

$312 \Rightarrow 312 * 2^7$

$2^{15} + 312 * 2^7$

$2^7 (256 + 312)$

568×2^7

Soln :

16 bit

opcode	Address1	Address2
2 bit	7	7

↳ # opⁿ = 2^2 = 4

⇒ # free opcodes
after allocate = (4-2)
the 2 addr Instⁿ = 2

# of 1-Addr Instⁿ
possible = $2 * 2^7$
= 256

opcode	Addr
9 bit	7 bit

# of free opcodes
after allocating = 256 - 200
1-Address = 56

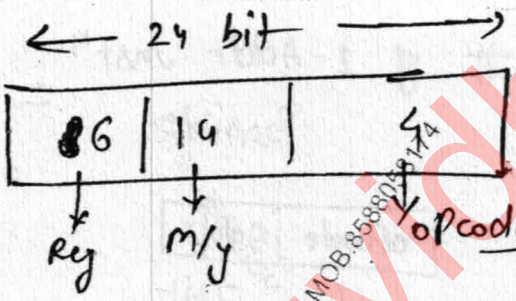
zero address
Instⁿ possible = $56 * 2^7$

opcode
16 bit

Instⁿ set :-
2-Address Instⁿ : 2
1-Address Instⁿ : 200
0-Address Instⁿ : 56×2^7
total : $2 + 200 + 56 * 2^7$

Que:- Consider a Hypothetical CPU which support 3 Byte instruction with 3 fields of information i.e. opcode, ~~Addr~~ Register operand & m/y operand. & System supports 60 Processor registers, and 16 KB m/y If there exist 2, 2-Address instⁿ then How many 1-Address m/y operand instⁿ are formulated.

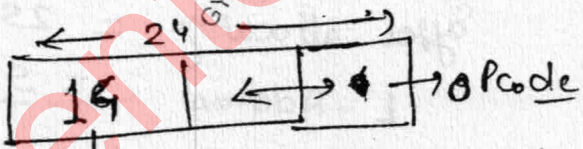
$60 < 2^6$ ← Reg bits



$\Rightarrow 2^4 = 16$ opⁿ

free opcodes $\Rightarrow$ $16 - 2 = 14$

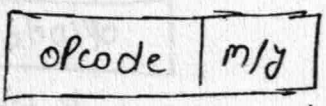
1-Address m/y operand Instⁿ :



$\Rightarrow 14 * 2^6$ ← Register operand address bits

(or)

m/y operand Instⁿ =



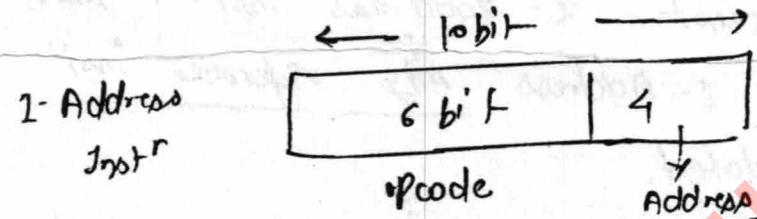
6-bit register operand

4 bit opcode

free opcode = 14

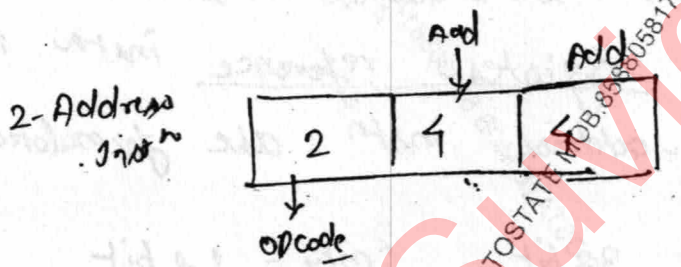
$\Rightarrow 2^6 \times 14$

Que:- Consider a Hypothetical CPU which supports 10 bit instruction with 4 bit Address. If there exist 2, 2-Address instⁿ and x, 1-Address instⁿ then what is the value of x when the system supports 16, 0-Address instruction.



1-Address Instr

free opcode = $(1) \times 2^{16} = 16$



2-Address Instⁿ

free opcode

free opcode = 4 - 2 = 2

1-address Instⁿ = $2 * 2^4 = 32 - 1 = 31$

free opcode

$x = 31$

Or

$(2^5 - x)2^4 = 16$

So $x = 31$

Que:- Consider a Hypothetical Processor which supports instruction format with 1 Register operand and 1-m/y operand. System contain 32 Registers and supports 256 kB memory.

Processor word size is 32 bit and supports 1-word long instructions:

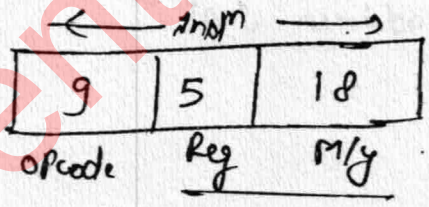
a) If there exist x-2address instⁿ then How many 1-Address m/y reference instⁿ are formulated.

b) If there exist x-2-address instⁿ and y, 1-Address registers reference instⁿ then How many 0-address instⁿ are formulated.

Solⁿ:

Instⁿ = 32 bit, m/y = 18 bit

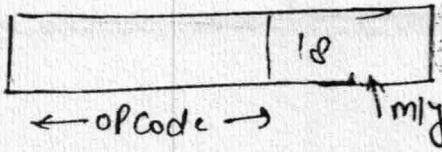
Regⁿ



# of 2-address free opcodes ⇓ given

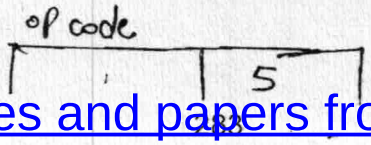
a) 2-Address instⁿ = 2^9 = 512-x

then 1 address instⁿ ⇒ $(2^9 - x) * 2^5$



$((2^9 - x) * 2^{18} - y)$

b) $(2^9 - x * 2^{18} - y) * 2^5$



$\Rightarrow (2^9 - x * 2^{18} - y)$

(b) # opⁿ = 2^9

* # free opcodes = $(2^9 - x)$

* # 1-Address reg Reference instⁿ possible

opcode	Reg.
27 bit	5 bit

$(2^9 - x * 2^{18})$

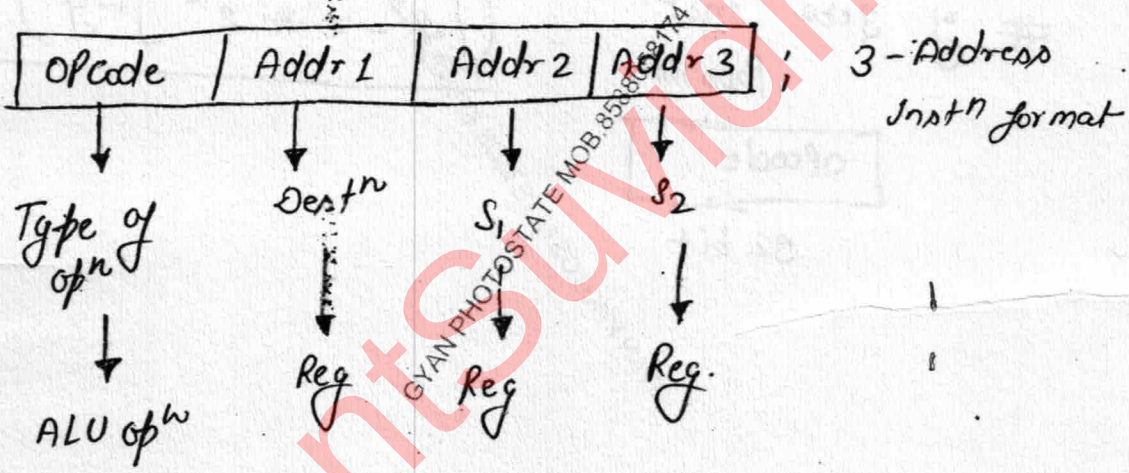
y free opcodes = $[2^9 - x * 2^{18}] - y$

y zero instⁿ possible = $([2^9 - x * 2^{18}] - y) * 2^5$

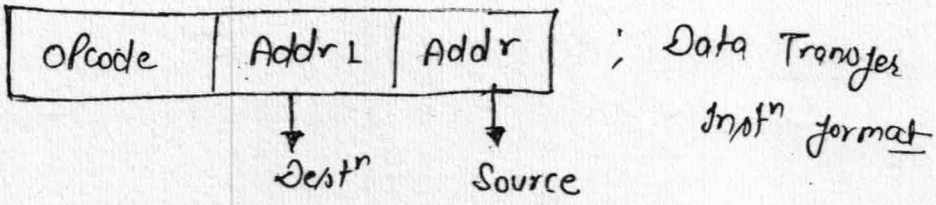
opcode
32 bit

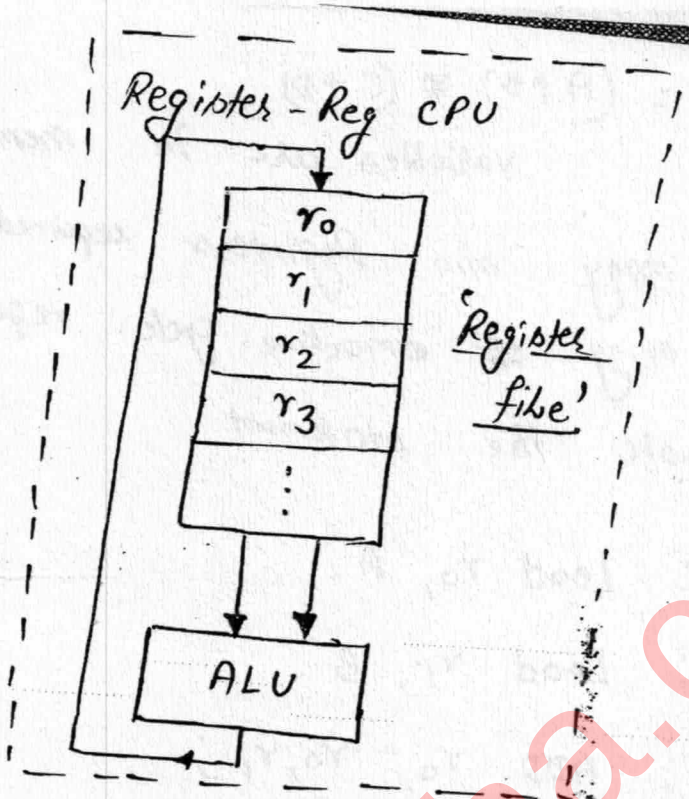
"Register - Register Reference CPU" :-

* In this organization ALU operations are performed only on a Register data.
* In this organization both of the operands are always required in the Register.
* After the Data Processing, Result is also placed in the Register.
* Instruction Design :-



* In this organization, 'LOAD'; and 'STORE' instn are used to transfer the data between memory and Register respectively. these are designed as a '2-Address instn'.





eg :- $(A * B) + C$

variables are in memory

I_1: Load r_0, A ; $r_0 \leftarrow m[A]$

I_2: Load r_1, B ; $r_1 \leftarrow m[B]$

I_3: MUL r_2, r_0, r_1 ; $r_2 \leftarrow r_0 * r_1$

I_4 ; Load r_3, C ; $r_3 \leftarrow m[C]$

I_5 ; Add r_4, r_2, r_3 ; $r_4 \leftarrow r_2 + r_3$

How many minimum Registers required :-

I_1: Load r_0, A ;

I_2: Load r_1, B ;

I_3: MUL r_0, r_0, r_1 ;

I_4: Load r_1, C ;

I_5: Add r_0, r_0, r_1 ;

2 Registers

5 m/c instn

2 { 3 addr } 3 { 2 addr }

Que:- $X = (A+B) * (C+D)$

variables are in memory.

How many min Registers required and also how many # machine cycle required to evaluate the statement.

I_1: Load r_0, A

I_2: Load r_1, B

I_3: ADD r_0, r_0, r_1;

I_4: Load r_1, C

I_5: Load r_2, D

I_6: ADD r_1, r_1, r_2;

~~I_7: Store X, I~~

I_7: MUL r_0, r_0, r_1;

I_8: Store A, r_0;

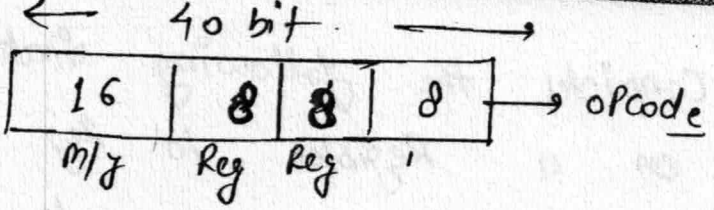
3- Registers

8 - m/c cycles

5 { 2 Addr inst^n } 3 { 3 Addr inst^n }

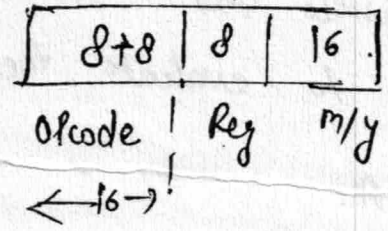
Que:- Consider a Hypothetical Processor which supports instruction format with 8 bit opcode, 2 Register operands and 1 m/y operand. Register file size is 256. m/y Bank size is 64kB, If there exist 128 3-address inst^n then How many 2-address inst^n are formulated which refers both register and m/y operand.

3 address Instⁿ :-



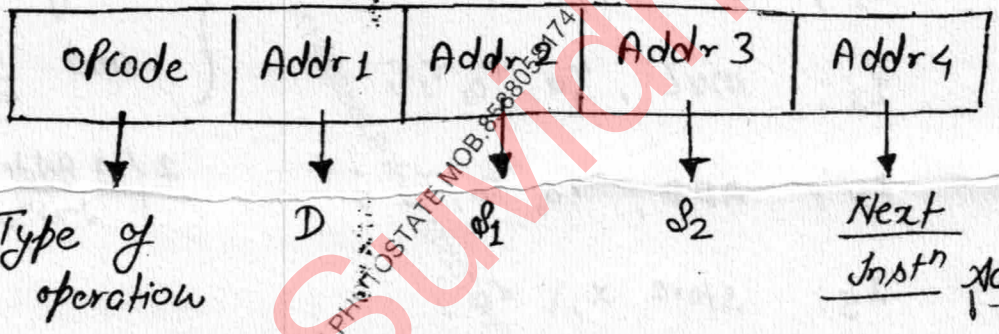
$2^8 = 256 - 128 = 128$ free opcode

2 address Instⁿ



$128 \times 2^8 = 2^{15}$ 2 address instⁿ with Reg & m/y Reference

4- address Instⁿ format :-



* Pc is a comulsory register in the CPU design, used to hold the Address of a Next Instruction during the Execution of a current Instruction therefore this Instruction format is not in the use.