

$$\textcircled{5} \quad EA \rightarrow [200 + 100] = [300]$$

$$[IR] + \text{Address part} = 100 + 500 = 600$$

of instr

$$\textcircled{6} \quad EA = [R_1]$$

& constant \Rightarrow data of $R_1 \rightarrow 400$.

$$\textcircled{7} \quad EA = [R_1] = [400]$$

constant \rightarrow data at 400 = 700

write this
as EA

$$\textcircled{8} \quad [R_{auto}] + \text{Step-8x} = 400 + 1 = 401$$

$$[R_{auto}] + \text{Step-8x} \Rightarrow (400) + 1 = 401$$

$$\textcircled{9} \quad [R_{auto}] + \text{Step-8x} \Rightarrow 400 - 1 = 399$$

Q] Consider a 3 word machine m/s. Identifying # cycles needed during the exec of instr.

Add $A[R_o]$, @B. $\xrightarrow{\text{Indirect AM}}$

\hookrightarrow Indirect AM
when R_o is IR

Int part
 $A[R_o]$

$$EA = [IR] + \text{Address field of the result}$$

$$EA = [R_o] + \text{Address} \rightarrow 1 \text{ memory reference is required.}$$

$$[400] \rightarrow 19$$

$$1 \rightarrow 19$$

Indirect
 $(@B) = M[R_B]$
 2 memory references are required.

SRC1 / Dest
 Add $A[R_d], @B \rightarrow SRC_2$
 2 ref. \leftarrow 1 ref. \leftarrow 2 ref.
 opcode.

$$A[R_d] \leftarrow A[R_d] + @B$$

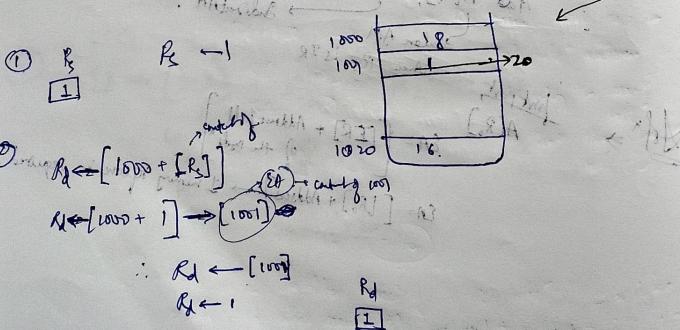
$$\therefore T_{Avg} = 1+1+2$$

0f = add to stack & then = 4 memory ref. are required.

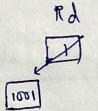
CATG PWD

① Memory loc's 1000, 1001, and 1002 have data values 18, 1, and 16.
 consider the following instr's:

I → Immediate
 1000 → 18
 @R_d → indirect
 R_d → index
 A[R_d] → base
 ADD I $R_d, 1000$
 Load $R_d, 1000(R_d)$
 ADD I $R_d, 1000$
 Store I $0(R_d), 20$
 What will be the value in loc' 1001 after exec?



② $R_d \leftarrow [R_d] + 1000$
 1 ref. \leftarrow 1 ref. \leftarrow 1000 is constant.



$$[0 + [R_d]] \leftarrow 20$$

$$[0 + 100] \leftarrow 20$$

③ Consider the following memory values & one address matching accumulator.
 Calc. the output of each instr.

20	40
30	50
40	60
50	70
60	80

Instr	Acc
Load I 20	20
Load D 20	60
Load @20.	60
Load I 30	30
Load D 30.	30
Load @30.	30

- ④ \Rightarrow Acc $\leftarrow 30$
 ⑤ \Rightarrow Acc $\leftarrow [20]$
 Acc $\leftarrow 50$.
 ⑥ \Rightarrow Acc $\leftarrow [20]$
 Acc $\leftarrow 50$.
 Acc $\leftarrow 70$.

⑦ \Rightarrow Acc $\leftarrow [20]$
 Acc $\leftarrow 40$
 Acc $\leftarrow 60$.
 Acc $\leftarrow 80$.

Q) Consider a RISC machine where each instrⁿ is 4 bytes long (word). Here, conditional branch instrⁿ w/ PC relative A.M. is used. Offset is specified in bytes for the calcⁿ of target locⁿ of the branch instrⁿ. Consider the following sequence of instrⁿ.

Inst. no.	Inst ⁿ	PC
i	Add R ₂ , R ₃ , R ₄	100
i+1	Sub R ₅ , R ₆ , R ₇	
i+2	Jump R ₁ , R ₉ , R ₁₀	[i+3]
i+3	(beg) R ₁ , offset	

If the target branch of iⁿ is i, find the value of offset.

B.T.A.: iⁿ is asking to calcⁿ the offset
is asking to calcⁿ the offset
jump value

$$\Sigma A = [PC] + \text{Offset}$$

normal of []

so a less no constant is needed
so no control of SA is needed.

O. Direct A.M.

PC=100 (say)

* PC value will hold the next instrⁿ address, which

SA=[100].

SA=[PC]+offset

enquiry the curr
A.M.
to that there is no lag
in CPU.

SA=[PA]+offset

100=116+offst

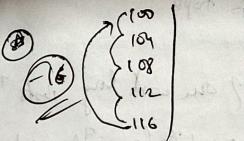
offst=100-116=-16

so it is loc after end of i+3 instrⁿ. (PC will be

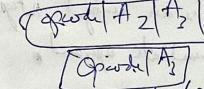
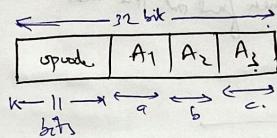
116 while executing it).

so offst=-16→16 needs to go back to i+3 instrⁿ (100)

so offst=16→16 needs to go back 16 bytes.



* Expanded Opcode Technique

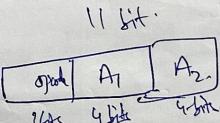


$$\# 3AI = k$$

$$\# 2AI = y$$

$$\# 1AI = z$$

In an 11-bit instrⁿ format, the size of address bit is 4-bits.
The computer uses expanded opcode technique to contains
5 (2 address instrⁿ & 32 1-address instrⁿ). Calc. the no. of
zero address instrⁿ it can support. It then typed PC,
no. of Address.instrⁿ = $\frac{1}{2} \times 11 = 5.5$ given



5-2 AI

3-1 AI.

2-0 AI.

1AI possible from given no. of

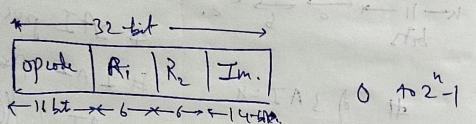
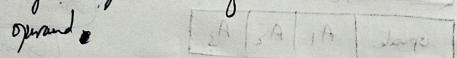
$$\text{Format} = (2^3 - 1) \cdot 2^4 = 48.$$

$$1AI \text{ a recurr } = (48 - 32) \cdot 2^4$$

$$= (6 \times 16) \times 256$$

* Take exam Q1 based on this topic.

- Q) A machine has 32-bit Architecture of one word long instrs?
 It has 64 registers, each of which 32-bit long. It needs to support 45 instrs. And immediate operand is also there along with 2 register operands. Assuming that immediate operand is an unsigned integer. Then find out the max. value of the immediate operand.

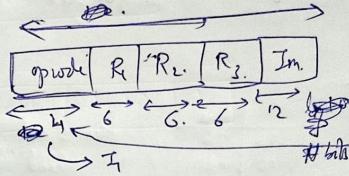


$$\begin{aligned}
 0\ 00 &\rightarrow + \\
 0\ 01 &\rightarrow x \\
 0\ 10 &\rightarrow - \\
 1\ 11 &\rightarrow \text{immediate}
 \end{aligned}$$

$= 32 - (6+6+6)$
 $\log_2 45 = 5$
 round off.
 $\# \text{bits } R_1 = R_2 = \log_2 64 = 6$
 $\# \text{bits } R_3 = \log_2 45 = 5$
 $\# \text{bits } \text{Im.} = 14$



- Q) Consider a processor of 64 registers & memory set of size = 12.
 Each instr has OPCODE, 2 src. registers, 1 destination register.
 & 12 bit immediate value. Each instr is stored by aligned fashion. If a program has 100 instrs, the amount of memory covered by the program in bytes are _____.
 (Ans) \Rightarrow



$$= 4 + 6 + 6 + 12$$

$$= 34$$

101	8
102	8
103	8
104	8
105	8
106	8
107	8
108	8
109	8
110	8
111	8

$$8+8+8+8+8 = 40 \text{ bits.}$$

$\therefore 5 \text{ blocks} = 5 \text{ bytes.}$

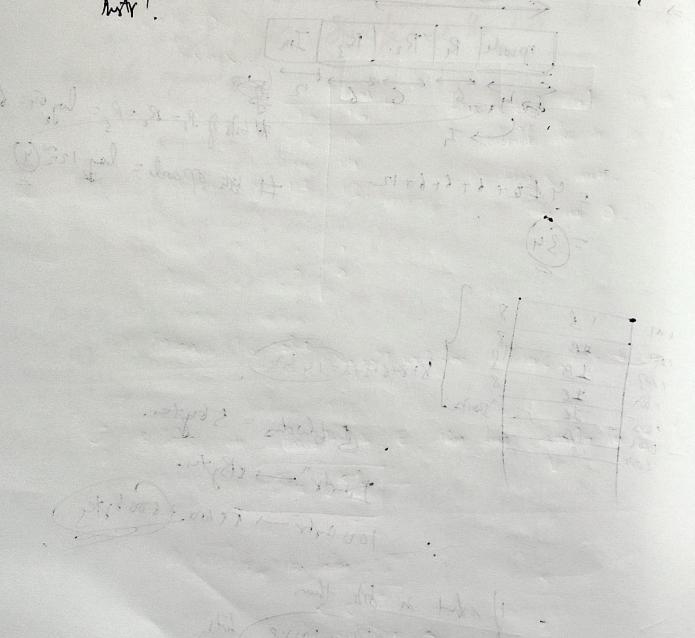
$1 \text{ instr} \rightarrow 1 \text{ byte.}$

$100 \text{ instrs} \rightarrow 100 \times 1 = 100 \text{ bytes}$

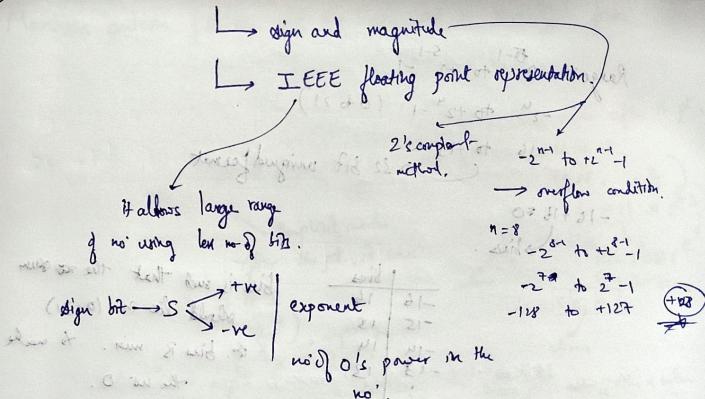
If word in 8 bits then

$$234 \times 100 \times 8 \text{ bits.}$$

- Q) A CPU is designed to have 58 bits. The CPU is able to address a max. of 64 memory locations. The length of machine code is same for all the instructions. Determine the no. of bits allocated for each address field.
- (B) Determine the min. no. of bits allocated for the opcode field of 1 address instruction.
- (C) Determine the min. no. of bits allocated for the fields of 2-address instruction.

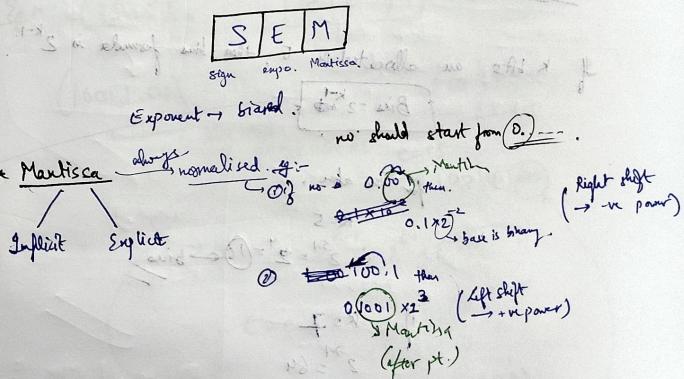


Number Representation



Mantissa \rightarrow actual no. in binary.

Standard format



also no. is 1001

Range is -2^{5-1} to $+2^{5-1}$
 -2^4 to $+2^4 - 1$ (0 to 21).

16 bit of \rightarrow 32 bit unsigned format
 difference between \rightarrow

$$-16 + 16 = 0$$

biases.

$$\begin{array}{c} \text{bias} \\ -16 \\ -15 \\ -14 \\ -13 \\ \vdots \\ 0 \end{array}$$

bias	16	15	14	13	...	0
min no.	16	15	14	13	...	0
max no.	16	15	14	13	...	0
difference	1	1	1	1	...	1
value	1	1	1	1	...	1

for sign & scale it

bit before pt. is power of 2.
 bias is such that the result
 should lie M(0.10...1)
 i.e. bias is min. to make
 the no. 0.

positive 0 \rightarrow 0.2FFFFM

Excess 16 code \rightarrow will work for all nos. ($16+0$)

If k bits are allocated to M & E , then bias formula is 2^{k-1} .

$$\therefore \text{Bias} = 2^{k-1}$$

so many bias bits

for the above, 3 bits will be bias bits

so bias bits = 3 \rightarrow 2³⁻¹ = 2² = 4

$$2^{5-1} = 2^4 = 16$$

$$2^{7-1} = 2^6 = 64$$

\Rightarrow Excess 64 code

Mantissa

Mantissa contains fraction part in binary but in normalized form.

only IEEE floating

101.11 \rightarrow Explicit mode (by default) In the explicit mode, after the pt., the 1st bit should be 1.
 $\Rightarrow 1.1\ldots$
 $\Rightarrow 0.1011 \times 2^{+3}$

Implicit mode.

After the pt., it can

be either 0 or 1.
 \therefore anything

The 1st bit before pt. should be 1.

$$\Rightarrow 1.0111 \times 2^{+2} \rightarrow M = 1.0111 \times 2^{+2}$$

10111

16 bits

S E M

1 0 1 1 1

($\Rightarrow k$ given)

then exponent bit

Eg: $(13.25)_{10}$

old no.

$(1001.1.01)_2$

new no.

$M: 0.100101 \times 2^{+5}$

$e = +5$

$M = 1001101.00$

\therefore bias $\rightarrow 2^{k-1} = 2^{6-1} = 2^5 = 32$

$E = e + bias = 5 + 32 = (37)_{10}$

$(100101)_2$

\therefore Mantissa has 9 bits given,

thus we append 2 zeros at last

\rightarrow we can do it, 'cause Mantissa is after pt.,

so it won't make any difference.

S E M
0 100101 100110100

$$\frac{V_1/V_2 \cdot 3^{16}}{2}$$

1. Now 6 digits in bit field of sign, neither mantissa nor bias.
 01001011 00110100 to bin. $V_2 =$

$$\begin{aligned}
 & 0.25 \rightarrow 1/4 \text{ bias?} \\
 & 0.28125 \rightarrow 3. \text{ Mantissa part} \\
 & 0.6875 \rightarrow 11/16 \\
 & 1101 \\
 & 4 \text{ new flt. format} \\
 & (43B4)_H
 \end{aligned}$$

* Addition / Subtraction of floating numbers

- i) Check for 0's
- ii) Align w/ Mantissa
- iii) Perform operation
- iv) Normalize the result.

Example. Perform Add.

$$0.583123 \times 10^3$$

$$0.123000 \times 10^{-4}$$

$$(0.583123 \times 10^3) \times 10^{-4}$$

$$\Rightarrow 0.230000 \times 10^{-1}$$

$$(0.123000 \times 10^{-4})$$

$$0.352000 \times 10^{-1}$$

0.12 less than 0.10

0.12 has 2 bits

0.12

0.12×10^4

0.00012×10^2

0.583123×10^3

0.583135×10^2

we will prefer +ve point.

and to move -ve sign we have to move 1 bit left

so we have to move 1 bit left

so we have to move 1 bit left

so we have to move 1 bit left

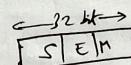
i] In Addition, if there is overflow bit.

$$\begin{aligned}
 & \rightarrow \text{Point should be left shift} \\
 & + 0.534 \times 10^3 \\
 & + 0.712 \times 10^2 \\
 & \hline
 & 1.246 \times 10^3
 \end{aligned}$$

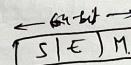
ii] In subtraction, if there is underflow bit.

$$\begin{aligned}
 & \rightarrow \text{Point should be right shift} \\
 & - 0.534 \times 10^3 \\
 & - 0.521 \times 10^2 \\
 & \hline
 & 0.013 \times 10^2 \\
 & 0.130 \times 10^2
 \end{aligned}$$

* IEEE 754 Standard



single precision
bias = 127



double precision
bias = 1023

	S	E	M	Number
Special cases	0	000-000	000-000	+0
	1	000-000	000-000	-0
	0	111-111	000-000	$+\infty$
	1	111-111	000-000	$-\infty$
	1/0	111...111	$M \neq 0$	not a number
	0/1	000-000	$M \neq 0$	denormalized no.
	0/1	$E \neq 0 - \infty$	$M = xxx$	number (Impl. limit mode)
	0/1	11-11		

$$E = 111 - 111 \quad \text{must be achieved as per IEEE std}$$

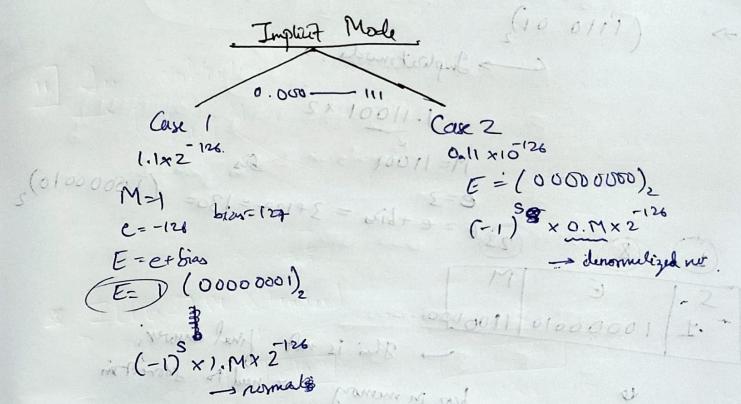
Range 0 to 255 → 254

$$\begin{aligned}
 E &= e + \text{bias} & \text{bias} &= 127 \\
 E &= e + 127 & e &= 128 \\
 &\Rightarrow \underline{128 \ 127} \\
 E &= 255 & e \text{ range} & -2^{17} \rightarrow 2^{17}-1 \\
 &= 127 + 127 & -2^7 \rightarrow 2^7-1 \\
 E &= 254 & -127 \rightarrow +127 \quad \text{if } E < 0
 \end{aligned}$$

* Denormalized

`Denseonly.no` is very very small and can't be normalized.

We can't store -ve int, bcoz only unsigned ints are allowed.
∴ IEEE 754 std. states, for a very small no. normalize it till
 ± 126



Report into IEEE Std.
~~IEEE 754-2008~~ 00000000000000000000000000000000 M It is not mentioned in question, then it is single precision.

According to table, it's a denouement

$$(-0.1)^s \times 0.1 \times 10^{-126}$$

May 1000

E=0

$$(-1)^5 * 1.110$$

(五)

1.110

$$+0 \\ \text{bias} = ND$$

- Page 1

$$e = E - b$$

— 5 — 122

$$= -1$$

Q) Given the number IEEE 754 std. format for a high precision, the number is $(-1425)_{10}$. The answer is 11111000 10000000 01010101 01010101.

$$\rightarrow (1110.01)_2 \quad \begin{matrix} \text{Invert mode} \\ \longleftarrow \end{matrix}$$

$$\begin{array}{r} \text{S} \times 10^3 \\ 0.1 \times 10^3 \\ (0.0000000) = \end{array} \xrightarrow{\text{M} = 11001} \begin{array}{r} 1.11001 \times 10^3 \\ \downarrow \\ M = 11001 \end{array} \quad \begin{array}{l} \text{Ans} \\ 5 \times 1.1 \\ 5.5 \end{array} \quad \begin{array}{r} 1.11001 \times 10^3 \\ (1.10000010)_2 \end{array}$$

1	8	23	$E = e + b$
5 +1	E	M	100000010

→ This is not final answer,
bcz in memory we need to store it in
hex form

d) From the given IEEE 754 repr., find the w. in decimal.

$$\begin{array}{r} \text{decimal.} & 8(1) & 6(0) & M(\text{remain}) \\ \text{given } \rightarrow & (00\ 0001)(11\ 00\ \dots\ 00) \\ \hline (1) & (8) & (2) & \} \text{ by default style} \\ \hline M & & & \end{array}$$

$$\boxed{S} \quad E \quad \boxed{M}$$

for reactant front

$$= (+) 28)_d \quad \text{sign bit} = 0$$

$\begin{array}{r} 01000000011000 \cdots \\ \hline S(1) \quad S(2) \quad M(255) \end{array}$

$$M = 1100 \quad \text{---} \quad E \in (1000, 0000) \\ = (128)_{10}$$

$$M=1100 \text{ -- -- }$$

$$\Rightarrow 1.1100\ldots x^2$$

$$\Rightarrow 1.11\text{dB} = -x^2$$

(11-100-1)

$\Rightarrow (1 \dots n) \uparrow$

$$\Rightarrow (+3.5)_{1.0} \cancel{+}$$

$$\therefore c = E - bias \\ = 128 - 124 = 4$$

Report in IEEE format

$$MS = 0 \rightarrow$$

$$E = e^{j\omega t} = 0 + j2\pi f (VA) = 01111111$$

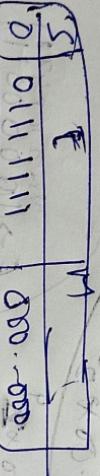
Implicit form

$$M = 000...000$$

$$e = 0$$

$$E = e^{j\omega t}$$

depends on



0001111100000000

3 F 8 0 0000000000000000

$$\rightarrow (24800000)_{10}$$

behavior

- Q) What can be the max. value which can be reported in IEEE 754?

Ans: 01111111 01111111 11111111 11111111

- Q) What can be the min. value which can be reported in IEEE 754?

Ans: 00000000

Word → 10011111 00011111 00011111 00011111

Sign → 00000000

Exponent → 00111111

Fraction → 00000000

Ans: 00111111 00011111 00011111 00011111