Q1. Determine the minimum value in 6-bit sign-magnitude representation.

Sol. We know, to find the range of numbers of particular number of bits, we have:

$$\boxed{-(2^{n-1}-1) \text{ to } (2^{n-1}-1)}$$

here n = number of bits

∴ for 6-bit minimum number will be

$$-(2^{n-1}-1)$$

$$\Rightarrow -(2^{6-1}-1) = -(2^5-1) = -31$$

we know 31 in binary is 11111

so 6-bit sign-magnitude minimum value

$$\Rightarrow \boxed{1\ 11111} = \boxed{-31}$$

Q2. How does a computer programme in 32-bit and 64-bit processors calculate the precise value of the distance between a spacecraft and a planet, if the distance is approximated by 101.765 meters?

Sol. In 32-bit processor:

Sign bit	Exponent	Mantissa
↓	↓	↓
1 +	8 +	23 = 32

2	101	
2	50	1
2	25	0
2	12	1
2	6	0
2	3	0
2	1	1
	0	1

$0.765 \times 2 = 1.$
$0.53 \times 2 = 1.$
$0.06 \times 2 = 0$
$0.12 \times 2 = 0.$
$0.24 \times 2 = 0$
$0.48 \times 2 = 0.9$
$0.96 \times 2 = 1.9$
$0.92 \times 2 = 1.8$
$0.84 \times 2 = 1.6$
$0.68 \times 2 = 1.3$
0.36×2
⋮

Binary of 101.765

$\Rightarrow$ 1100101.110000111....

In normalised form:

$1.100101110000111... \times 2^6$

∵ number is positive
∴ Sign bit is positive (0)

For exponent

⇒ 127 + 6 ⇒ 133

Binary of 133
= 10000101

2	133	
2	66	1
2	33	0
2	16	1
2	8	0
2	4	0
2	2	0
2	1	0
	0	1

For mantissa we make fractional part upto 23 bits

∴ 10010111000011100000000

∴ in 32-bit precision

0	10000101	10010111000011100000000
Sign bit	Exponent	Mantissa

- In 64-bit precision

∵ we already calculated the binary of the number and converted to normalised form, but this time it is for 64 bit

∴

Sign bit	Exponent	Mantissa
1 +	11 +	52 = 64

∴ For exponent

1023 + 6 = 1029

It's binary

10000000101

2	1029	
2	514	1
2	257	0
2	128	1
2	64	0
2	32	0
2	16	0
2	8	0
2	4	0
2	2	0
2	1	0
	0	1

∴ in 64 bit precision

0	10000000101	100101110000111 0...... (36 0's)
Sign bit	Exponent	mantissa

Q3. Determine the decimal value of 1001101, if are in (i) sign - magnitude form, (ii) 2's complement form and (iii) 1's complement form.

Sol.

(i) ∵ MSB is 1 ∴ number is negative

001101 in decimal is $= 8+4+1$
$= 13$

∴ number is $\boxed{-13}$

(ii) If the number is in 2's complement, then we take the 2's complement.

→ 1001101
first 1's complement

0110010 than add 1

$$\begin{array}{r} 0110010 \\ + \quad 1 \\ \hline 0110011 \end{array}$$

→ $1+2+16+32 = 51$

∴ $\boxed{-51}$

(iii) 1's complement form

find 1's complement of the number.

⇒ 1001101 → 0110010
↳ $2+16+32 = 50$

⇒ $\boxed{-50}$

Q4 Perform the following arithmetic operations using Two's complement 8-bit representation. Also check of overflow in each operation.

(a) 20 − 19 (b) −67 − 34

Sol

(a)

$$\begin{array}{r} 20 \\ -19 \end{array}$$

$\Rightarrow 20 + (-19)$

Binary of 20 = 16 + 4 = 20

∴ 10100

in 8-bit = 00010100

Binary of 19 = 16 + 2 + 1 = 19

= 10011

in 8 bit = 00010011

∵ 19 is negative we take it's 2's complement

00010011 → 1's complement

⇒ 11101100

$$\begin{array}{r} 11101100 \\ +\quad 1 \\ \hline 11101101 \end{array}$$

Adding 2's complement of −19 to 20

$$\begin{array}{r} 00010100 \\ +\ 11101101 \\ \hline 00000001 \end{array} = \boxed{1 \text{ in decimal}}$$

Overflow condition

- Two positive numbers when added give a negative number.

or

- Two negative numbers when added give an positive number

In this case both numbers are different

∴ $\boxed{\text{No overflow.}}$

(b) -67
-34

Binary of 67 = $64+2+1=67$
= 1000011

in 8-bit = 01000011

since it is negative
$\therefore$ MSB = 1
$\Rightarrow 11000011$

Binary of 34 = $32+2=34$
= 100010

in 8-bit = 00100010

since it is negative
$\therefore$ MSB = 1
$\Rightarrow 10100010$

Taking 2's complement of both

-67

11000011

1's complement

$\Rightarrow 10111100$

$+ \quad 1$

10111101

-34

10100010

1's complement

$\Rightarrow 11011101$

$+ \quad 1$

11011110

$$\Rightarrow \begin{array}{r} 10111101 \\ +\ 11011110 \\ \hline 1\,10011011 \end{array}$$

↳ neglecting it, we get = 10011011

Taking it's 2's complement again
1's complement : 01100100
$+ \quad 1$

$\underline{01100101} \rightarrow \boxed{-101}$

So, $(-64)+(-34) = -101$

$\because$ These are two negative numbers but their answer is not positive

$\therefore$ $\boxed{\text{No overflow}}$

Q5. Express 65.25 as a floating point number using IEEE double precision.

Sol. In double precision

Sign bit	Exponent	Mantissa

$1 + 11 + 52 = 64$

Binary of 65 = $64+1=65$
= 1000001

Binary 0.25 = $0.25 \times 2 = 0.5$
$0.5 \times 2 = 1.0$

$\Rightarrow$ 01

In normalised form

1000001.01

$\Rightarrow 1.00000101 \times 2^6$

Sign bit is 0 as number is positive

For exponent

$1023 + 6 = 1029$

Binary of 1029 = $1024 + 4 + 1 = 1029$
= 10000000101

$\therefore$ In double precision

0	1 000 0000 101	0000 01010. . . . . . . (43 0's)
Sign bit	Exponent	Mantissa

Q6. Find out the floating-point number from the following representation.

1 10000001 01100000000000000000000

Sol

We know,

$$N = (-1)^{sign} \times \text{mantissa} \times 2^{\text{exponent}}$$

Sign = 1

mantissa $= \cancel{1}(0 + \frac{1}{4} + \frac{1}{8}) = 1.375$

Exponent = Binary of exponent − 127   [10000001 → 129]

= 129 − 127

= 2

$\Rightarrow N = (-1)^1 \times (1.375) \times 2^2$

$\boxed{N = -5.5}$

Q7. In an arithmetic logic unit, two half adders are connected to form a full adder. Show the connections between half adders. Also, determine the required Boolean logic for the outputs.

Sol We have to use two half adders to form a full adder. To do so, first we should know the truth table and boolean expression.

A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Sum expression $= \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

$= C(\bar{A}\bar{B} + AB) + \bar{C}(\bar{A}B + A\bar{B})$

$= C(\overline{A \oplus B}) + \bar{C}(A \oplus B)$

$= \boxed{C \oplus (A \oplus B)}$

Carry expression $= \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$

$= BC(\bar{A} + A) + A(\bar{B}C + B\bar{C})$

$= BC + A\bar{B}C + AB\bar{C}$

$= C(B + A\bar{B}) + AB\bar{C}$

$= CB + CA + AB\bar{C}$

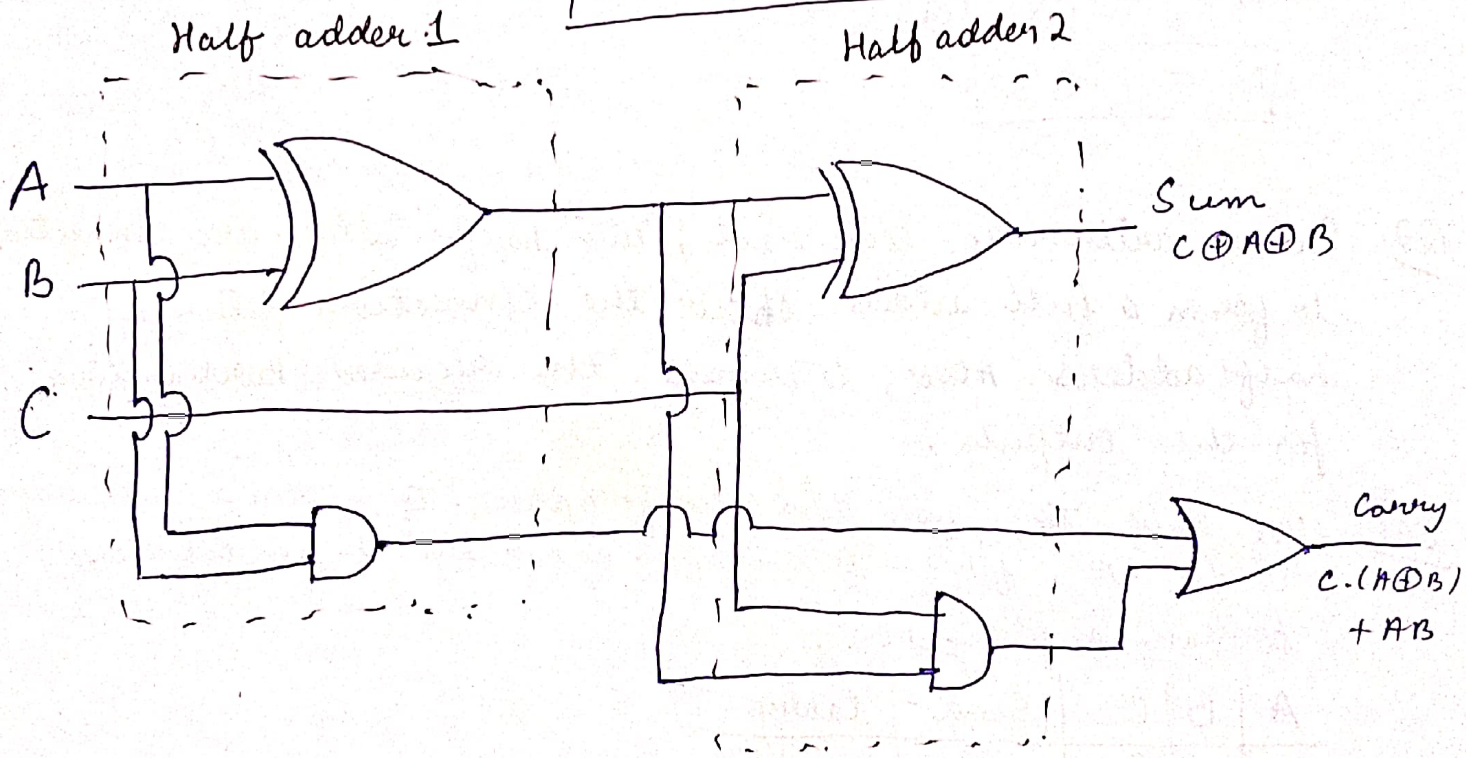
$= B(C + A\bar{C}) + CA$

$= BC + BA + CA$

$= CA + CB + AB$

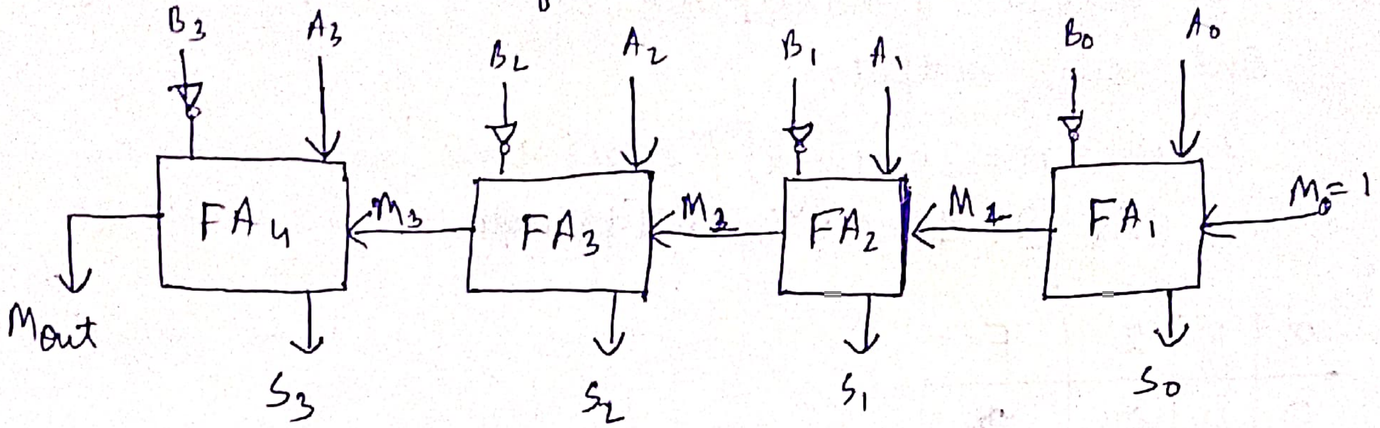
$= C(\bar{A}B + A\bar{B}) + AB$

$= \boxed{C.(A \oplus B) + AB}$



Q8. Explain how control input M results a 4-bit binary subtractor with suitable logic diagram.

Sol. For subtraction we do addition, so for this we will be using 4 full adders and an input $M = 1$



Let us take an example, subtract 7 from 10.
So for this we provide the complement of each bit of 7 in each full adder, and the rest two inputs as it is. The value of M is get by the carry of the full adder (M_i = $Carry_i$)

Binary of 10 in 4-bit = 1010
Binary of 7 in 4-bit = 0111

For FA1, $A_0 = 0$, $B_0 = \bar{1} = 0$, $M_0 = 1$
$S_0 = 1$

For FA2, $A_1 = 1$, $B_1 = (\bar{1}) = 0$, $M_1 = 0$
$S_1 = 1$

For FA3, $A_2 = 0$, $B_2 = \bar{1} = 0$, $M_2 = 0$
$S_2 = 0$

For FA4, $A_3 = 1$, $B_3 = \bar{0} = 1$, $M_3 = 0$
$S_3 = 0$, with carry = 1 (Mout)

So the result we get is 0011 which is 3, this is how subtraction is done.

Q9. Implement the following Boolean function with a suitable multiplexer.

$$F(x, y, z) = \Sigma(0, 3, 6, 7)$$

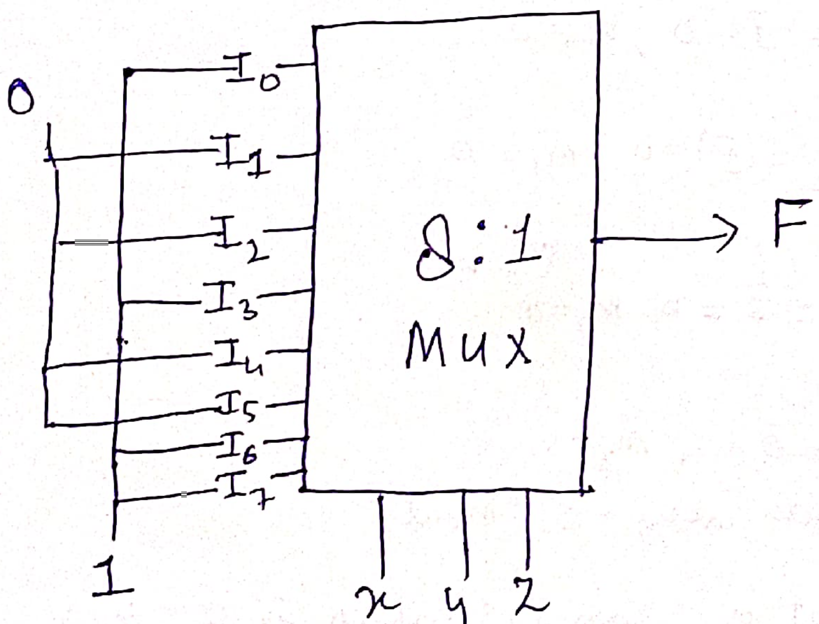
Sol. We know, a multiplexer takes 2^n inputs and gives only one output.

here $n = 3$ (x, y, z) [select lines]

$\therefore$ 8 : 1 MUX will be used.

x	y	z	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

So only 0, 3, 6 and 7 will be high and 1, 2, 4, 5 will be low



Q 10. Determine the minimum number of bits needed to represent -32 in 2's compliment representation.

Sol For n-bit two's compliment number, range is given as:-

$$-(2^{n-1}) \text{ to } (2^{n-1}-1)$$

we need to find smallest 'n' such that -32 lies within the range.

for $n = 5$

-2^4 to $(2^4 - 1)$

$\Rightarrow$ -16 to 15 ✗

for $n = 6$

-2^5 to $(2^5 - 1)$

-32 to 31 ✓

$\therefore$ The minimum number of bits to represent -32

$\Rightarrow$ $\boxed{6 \text{ bits}}$