Name → Vaishik Vishwakarma
Reg No. → 21BIT0003

A1) Decimal values → 55000
→ 30000

Decimal to binary :- $(55000)_{10}$

2	55000	0
2	27500	0
2	13750	0
2	6875	1
2	3437	1
2	1718	0
2	859	1
2	429	1
2	214	0
2	107	1
2	53	1
2	26	0
2	13	1
2	6	0
2	3	1
	1	

$(55000)_{10} = (1101011011011000)_2$

$(30000)_{10} \rightarrow$

2	30000	0
2	15000	0
2	7500	0
2	3750	0
2	1875	1
2	937	1
2	468	0
2	234	0
2	117	1
2	58	0
2	29	1
2	14	0
2	7	1
2	3	1
	1	

$(30000)_{10} = (111010100110000)_2$

Binary Addition ⇒ 1 1 0 1 0 1 1 0 1 1 0 1 1 0 0 0 } addition
1 1 1 0 1 0 1 0 0 1 1 0 0 0 0

$$
\begin{array}{r}
1101011011011000 \\
+\,0111010100110000 \\
\hline
010100110000001000 \\
\hline
\end{array}
$$

Here $1+1=0$, 1 as carry

$0+0=0$

$0+1=1$

$1+0=1$

No over flow occured

Binary Subtraction →

⇒

$$\begin{array}{r} 1101011011011000 \\ -\ 0111010100110000 \\ \hline 0110000110101000 \end{array}$$

Here $0-1=1$ (Borrow 1)

$1-0=1$

$1-1=0$

$0-0=0$ no. overflow occured

If there is sum of two numbers (Positive) & the result is negative then a overflow condition occurs or two negative number gives a positive outcome.

Overflow occur when two binary no.'s is added and the result is too large to fit in the group of bit.

In this case it does not result in overflow condition.

~~Max^m positive bits = $2^{15}-1$ = 32767~~

A2) 5.125 to binary.

5 → 101

0.125 → 0.125 × 2 = 0.25

0.25 × 2 = 0.5

0.5 × 2 = 1

∴ 0.125 → 0.001

Now 5.125 = 101.001

shift 1.01001 × 2^2

For a single precision (32 bits), the exponent bias is 127

Biased exponent = 2+127 = 129

129 ⇒ 10000001

Mantisa = 01001

IEEE 754 representation of 5.125 is single format is

0 10000001 01001000000000000000000 —(1)

Repeating same process for 4.75 we get

4.75 ⇒ 100.11

shifting point → 1.0011 × 2^2

The number is positive sol bit is 0

exponent bias is still 127

biased exponent = 2+127 = 129

129 ⇒ 10000001

mantisa is 0011

IEE 754 representation of 4.75 is 0 10000001 0011000000000000000
00000 —(11)

Align (i) & (ii)

Add the mantisa we get,

$\Rightarrow$ 0011000000000000000000000

Normalising

$\Rightarrow$ 01111000000000000000000

$= 1.1110000000000000000000000 \times 2^{2}$

Biased exponen = 129 − 127 = 2

Mantissa = 1.1110000000000000000000

The decimal representation is

= 7.75 Ans

A3) 28 = 11100 → dividend

13 = 01101 → divisor

Dividend register (8 bits) = 00011100

Divisor register (8 bits) = 00001101

quotient = 0, counter - 8

Step	Dividend register	Divisor register	Quotient	Counter
0	00011100	00001101	00000000	0000
1	00111000	"	00000001	1111
2	01110000	"	00000011	1110
3	11100000	"	00000110	1101
4	11000001	"	00001100	1100
5	10000011	"	00011001	1011
6	00000111	"	00110011	1010
7	00001110	"	01100110	1001
8	00011100	"	11001101	1000

Converting binary quotient & remainder to decimal

Quotient (binary) = 11001101

Quotient (decimal) = 205

Remainder (binary) = 00001100

Remainder (decimal) = 12

∴ Quotient = 205

Remainder = 12 Ans

A4) DMA improves performance of computer in many ways →

1) DMA reduces CPU involvement in data transfer, freeing up the CPU for the other tasks, thereby improving overall performance.

2) DMA allows for faster data ~~s~~ transfer between devices and memory, enhancing system efficiency.

3) By enabling bulk data transfers, DMA reduces the overhead associated with individual data transfer, resulting in improved performance.

4) DMA enables simultaneous data transfer and processing, allowing for concurrent operation and maximizing system throughput.

5) With DMA, devices can directly access memory without involving the CPU, reducing latency and enhancing system performance & responsiveness.

6) DMA ~~s~~ minimizes data bottlenecks by efficiently managing data movement, leading to faster overall system performance.

7) DMA is particularly beneficial for high-bandwidth application, like multimedia streaming disk I/O, and network communi^m enhancing thes performance by offloading data transfer tasks from the CPU.

A5) a) 23 × 12

↓ ↓

Multiplicant Multiplier

23 → 10111

12 → 01100

Product register (5 bits) : 00000

Accumulator register (6 bits): 000000

000000 + 23 = 23

After shift, Product register : 00001

~~Product~~ Accumulator register : 000011

Bit 3 of multiplier is 1 :

000011 + 23 = 46

Product register : 000 11 (shifted)

Accumulator register : 000011 (shifted)

The content of accumulator (000011)

The final result is (000011) = 11 in decimal.

∴ 23 × 12 = 11 using sequential circuit multiplier algorithm.

(iv). 23 × −12 using Booth algorithm & bit pair recoding algorithm

23 → 010111

−12 → 111100

Bit pair recoding of multiplier (111100): 1111001

Product register (6 bits): 000000

Accumulator register (7 bits): 0000000

Applying booth algorithm & bit recoding algorithm

Bit 0 (multiplier) = 1, Bit pair recoding = 1

Subtract the multiplicand (23) from accumulator →

0000000 − 010111 = 1010010

Product register → 000000

Accumulator register → 1101001

Similarly for,

Bit 1 : → PR = 000000

AR = 0110100

Bit 2 : → PR = 000000

AR = 0011010

Bit 3 → PR → 000000

AR → 0001101

Bit 4 → PR → 000000

AR → 1011011

Bit 5 → PR → 000000

AR → 1100100

The final result in binary is 1100100 which is −276 in decimal.

∴ 23 × −12 = −276 Ans