Princekumar Pijora 656570553

Task0.pdf

E.A.1. a.

1 0 0 1 0 1 1 0 0 0

binary to decimal

base 2 base 10

$$\begin{array}{cccccccccc} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 2^9 & 2^8 & 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{array}$$

$= 2^9 \times 1 + 2^8 \times 0 + 2^7 \times 0 + 2^6 \times 1 + 2^5 \times 0 + 2^4 \times 1 + 2^3 \times 1 + 2^2 \times 0 + 2^1 \times 0 + 2^0 \times 0$

$= 2^9 + 0 + 0 + 2^6 + 0 + 2^4 + 2^3 + 0 + 0 + 0$

$= 2^9 + 2^6 + 2^4 + 2^3$

$= 512 + 64 + 16 + 8$

$= 600$

$$\boxed{\underset{\text{binary}}{(1001011000)_2} \longrightarrow \underset{\text{decimal}}{(600)_{10}}}$$

E.A.2 . a.

2335

↓

octal to decimal

↓ ↓

base 8 base 10

2 3 3 5

8^3 8^2 8^1 8^0

$= 2 \times 8^3 + 3 \times 8^2 + 3 \times 8^1 + 5 \times 8^0$

$= 2 \times 512 + 3 \times 64 + 3 \times 8 + 5 \times 1$

$= 1024 + 192 + 24 + 5$

$\boxed{= 1245}$

$$\boxed{\underset{\text{octal}}{(2335)_8} \longrightarrow \underset{\text{decimal}}{(1245)_{10}}}$$

0 1 2 3 4 5 6 7 8 9 A B C D E F

A → 10, B → 11, C → 12, D → 13, E → 14, F → 15

E.A.3 d.

fffe. 428

↓

hexadecimal to decimal

↓ ↓

Base 16 Base 10

f = 15

e = 14

$$\begin{array}{ccccccc} f & f & f & e. & 4 & 2 & 8 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 16^3 & 16^2 & 16^1 & 16^0 & 16^{-1} & 16^{-2} & 16^{-3} \end{array}$$

because it is the right side of the Precision

$$= 15\times16^3 + 15\times16^2 + 15\times16^1 + 14\times16^0 + \frac{4}{16} + \frac{2}{16^2} + \frac{8}{16^3}$$

$$= 61440 + 3840 + 240 + 14 + 0.25 + 0.0078125 + 0.001953125$$

$$= 65534.25977$$

$$\boxed{= 65534.260}$$

$(fffe.428)_{16} \longrightarrow (65534.260)_{10}$

hexadecimal decimal

E.A.7 9.

38

↓

hexadecimal to binary

↓ base 16 ↓ base 2

3 8

2^3	2^2	2^1	2^0
8	4	2	1
0	0	1	1

→ sum of this two equals to 3.

↓ binary representation of 3

2^3	2^2	2^1	2^0
8	4	2	1
1	0	0	0

→ binary representation of 8

↓ because we need addition 8 so if take 8 then we get the sum.
and we will put 0 in other places because we are not using those digits.

$(38)_{16}$ hexadecimal $\rightarrow$ $(0011\ 1000)_2$ binary

EA. 10.

① 8 bit 2's Complement

0100 1011 − $\underbrace{0001\ 0111}$

1's Complement of the given 8 bit
→ just basically flipping the digits

1110 1000
+1 } and then we are adding one to get 2's complement

1110 1001 → now we have to add this to given 8 bit

so, we will not consider this one

```
 1  1   1  11
   0100 1011
+  1110 1001
   0011 0100
```

so, $\boxed{0011\ 0100}$ → is 2's complement for given 8 bit binary digits.

(2)

0100 0001 − $\underbrace{0001\ 0001}$

flipping the digits

1110 1110 → 1's complement for given 8 bit

\+ 1 (adding one)

1110 1111 → 2's complement for given 8 bit

add this together

will not consider this carry (1)

(1) 1 1 1 1 1

0100 0001

\+ 1110 1111

0011 0000

so, $\boxed{0011\ 0000}$ → is 2's complement for given 8 bit number

(3)

1000 0001 − $\underbrace{0010\ 0000}$

flipping the digits

add this two together

1101 1111 → 1's complement

\+ 1 (∵ adding 1)

11100000

will not consider this

(1)

1000 0001

\+ 1110 0000

so, $\boxed{0110\ 0001}$ → is 2's complement for given 8 bit number

Q.1 coverting into IEEE 754 single Precision floating point representation

(a) 2.5 (base 10)

dividing 2 by 2.

$$\begin{array}{r|l l} 2 & 2 & \\ 2 & 1 & 0 \\ & 0 & 1 \end{array}$$

2 in binary : 10

multiplying only fractional part by 2

0.5×2	1.0	1
0.0×2	0.0	0

0.5 in binary : 1

$(2.5)_{10} \longrightarrow (10.1)_2 \; \boxed{= 1.01 \times 2^1}$

→ s(sign) = 0 because positive number

→ Exponent (biased 127) = 127 + 1 = 128 (∵ 1 from 2^1)

$$\begin{array}{r|l l} 2 & 128 & \\ 2 & 64 & 0 \\ 2 & 32 & 0 \\ 2 & 16 & 0 \\ 2 & 8 & 0 \\ 2 & 4 & 0 \\ 2 & 2 & 0 \\ 2 & 1 & 0 \\ & 0 & 1 \end{array}$$

= (8 bits) 1000 0000

→ Fractional part = 01

= (23 bits) 0100 0000 0000 0000 0000 000

Sign	Exponent	fractional part
0	1000 0000	010 0000 0000 0000 0000 0000

(b) -0.001 (base 10)

→ we are just starting with positive version

$$|-0.001| = 0.001$$

$2 \underline{|0}$ 0 ↑ 0 in binary : 0

0

0.001 × 2	0.002	0
0.002 × 2	0.004	0
0.004 × 2	0.008	0
0.008 × 2	0.016	0
0.016 × 2	0.032	0
0.032 × 2	0.064	0
0.064 × 2	0.128	0
0.128 × 2	0.256	0
0.256 × 2	0.512	0
0.512 × 2	1.024	1
0.024 × 2	0.048	0

→ I think, I didn't get any fractional part that was equal to 0.

$= (0.001)_{10} = 0.00000000010$

$= -(0.001)_{10} = -0.00000000010$

→ To maintain the same degree of precision

$$2^K * 10^{-m} \geq 1$$ -m - fractional part

$$2^K * 10^{-3} \geq 1$$

$$2^K \geq 1$$

$$K \geq 10$$

$\boxed{-(0.001)_{10} = -1.0 \times 2^{-10}}$

→ s (sign) = 1 because it is negative

→ Exponent (biased, 127) = 127−10 (∵ −10 from 2^{-10})
= 117

2	117	
2	58	1
2	29	0
2	14	1
2	7	0
2	3	1
2	1	1
	0	1

= (8 bits) 0111 0101

→ fractional part :- 1.0

= (23 bits) 0000 0000 0000 0000 0000 000

1	0111 0101	0000 0000 0000 0000 0000 000
sign	exponent	fractional part