201902708 신동훈

1.

(a) $(365)_r = (194)_{10}$

$3r^2 + 6r = 189$

$3r(r+2) = 189$

$r^2 + 2r - 63 = 0$

$(r+9)(r-7) = 0$

$\therefore r = 7$

(b) $(BEE)_r = (2699)_{10}$

$11 \cdot r^2 + 14r + 14 = 2699$

$11r^2 + 14r - 2685 = 0$

$(11r+179)(r-15) = 0$

$\therefore r = 15$

2. (a) $10.75_{10} = 10 + 0.75$ 이다.

10을 2진수로 변환 ⇒ $\begin{array}{r l} 2\,)\,10 & \cdots 0 \\ 2\,)\,5 & \cdots 1 \\ 2\,)\,2 & \cdots 0 \\ 1 & \end{array}$ ⇒ 1010_2

0.75를 2진수로 변환 ⇒ $\begin{array}{r} 0.75 \\ \times \quad 2 \\ \hline (1).50 \\ \times \quad 2 \\ \hline (1).0 \end{array}$ ⇒ 0.11_2

⇒ 따라서 1010.11_2

(b) $1.875_{10} = 1 + 0.875$ 이다.

1을 2진수로 변환 ⇒ 1_2

0.875를 2진수로 변환 ⇒ $\begin{array}{r} 0.875 \\ \times \quad 2 \\ \hline (1).750 \\ \times \quad 2 \\ \hline (1).50 \\ \times \quad 2 \\ \hline (1)\ 0 \end{array}$ ⇒ 0.111_2

⇒ 따라서 1.111

3. (a) $57_8 + 75_8 = 154_8$,

⇒ $\begin{array}{r} 57 \\ 75 \\ \hline 154 \end{array}$

(b) $0.524_6 + 1.333_6 = 2.301_6$

⇒ $\begin{array}{r} 0.524 \\ 1.333 \\ \hline 2.301 \end{array}$

(c) $AB7_{16} + 999_{16} = 1450_{16}$

⇓ $\begin{array}{r} AB7 \\ 999 \\ \hline 1450 \end{array}$

4. (a) 9는 3의 제곱이므로 3진수의 2비트에 해당하는 수를 9진수에서는 1비트의 수로 표현할 수 있다. 따라서 3진수를 2비트씩 끊어서 그에 해당하는 9진수로 바꾸어주면 된다.

(b) 1 20 12 01 . 20 12 01 2 = 1651.6516_9

1 6 5 1 . 6 5 1 6

5. (a) $12.3_4 = 12_4 + 0.3_4$ 이고

12_4를 2진수로 변환 ⇒ 2) 12_4 … 0, 2) 3_4 … 1, 1 ⇒ 110_2 이고

0.3_4를 2진수로 변환 ⇒ $0.3_4 \times 2 = (1).2_4$, $\times 2 = (1).0$ ⇒ 0.11_2 이다.

⇒ 110.11_2

(b) $123.4_5 = 123_5 + 0.4_5$ 이고

123_5를 2진수로 변환 ⇒ 2) 123_5 … 0, 2) 34 … 1, 2) 14 … 1, 2) 4 … 0, 2) 2 … 0, 1 ⇒ 100110_2

0.4_5를 2진수로 변환 ⇒ $0.4_5 \times 2 = (1).3_5$, $\times 2 = (1).1$, $\times 2 = (0).2$, $\times 2 = (0).4$, $\times 2$ ⋮ ⇒ $1100\,1100\cdots_2$

⇒ 100110.11001_2

6.

(a) 1은 이진수로 00000001이다.

2의 보수: 11111111

1의 보수: 11111110

부호-크기: 10000001

(b) 2의 보수가 8이기 위해서는 $2^8-(-8)$ 이어야 한다.

2^8은 8비트에서는 00000000으로 표현된다. 즉 $2^8+8 = 00001000$ 이다.

따라서 2의 보수: 00001000

1의 보수: $2^8-1-(-8) = 2^8+7 = 00000111$

부호-크기: 00001000

(c) 15는 이진수로 00001111이다.

2의 보수: 11110001

1의 보수: 11110000

부호-크기: 10001111

7. (a) $4 = 0100_2$

$-6 = 1010_2$

$$\begin{array}{r} 0100 \\ +\ 1010 \\ \hline 1110 \end{array} \Rightarrow 1110 = -2$$

overflow는 발생하지 않았다.

(b) $-4 = 1100_2$

$-6 = 1010_2$

$$\begin{array}{r} 1100 \\ +\ 1010 \\ \hline 1\,0110 \end{array} = 6$$

→ 음수와 음수를 더했는데 양수 ⇒ overflow가 발생

(c) $-4 = 1100_2$

$6 = 0110_2$

$$\begin{array}{r} 1100 \\ -\ 0110 \\ \hline 0110 \end{array} \Rightarrow 0110_2 = 6$$

→ 음수에서 양수를 뺐는데 양수 = overflow 발생

(d) $-4 = 1100_2$

$-6 = 1010_2$

$$\begin{array}{r} 1100 \\ -\ 1010 \\ \hline 0010 \end{array} \Rightarrow 0010_2 = 2.$$

overflow 발생 X.

8. (a) 2 = 0010

5 = 0101

6 = 0110

⇒ 001001010110 , 12비트 필요.

(b)
$$\begin{array}{l} 2\,)\,256_{10} \\ 2\,)\,128 \quad \cdots 0 \\ 2\,)\,64 \quad \cdots 0 \\ 2\,)\,32 \quad \cdots 0 \\ 2\,)\,16 \quad \cdots 0 \\ 2\,)\,8 \quad \cdots 0 \\ 2\,)\,4 \quad \cdots 0 \\ 2\,)\,2 \quad \cdots 0 \\ \quad 1 \quad \cdots 0. \end{array}$$

⇒ 0100000000 , 부호비트가 필요하므로 10비트가 필요하다.

10000000

(C) 세 자리 십진수 중 최댓값은 999 이고 최솟값은 100 이다.

$2^9 < 999 < 2^{10}$ 이므로

세 자리 십진수를 이진수로 표현하기 위해서는 11비트가 필요하고.

BCD는 4×3 =12 이므로 12비트가 필요하다.

9.

(a) 짝수 parity bit는 parity bit를 더하였을 때 1의 개수가 짝수개여야 하므로 1의 개수가 홀수인 경우 parity bit는 1이고, 1의 개수가 짝수인 경우 parity bit는 0이다.

이때 1의 개수가 홀수일 때 1이 나오며, 1의 개수가 짝수일 때 0이 나오는 것은 Exclusive - OR 이다.

따라서 $\underset{\uparrow}{P} = a_6 \oplus a_5 \oplus a_4 \oplus a_3 \oplus a_2 \oplus a_1 \oplus a_0$

짝수 parity bit

(b) 오류가 발생하는 경우는 1의 개수가 홀수인 경우이며, 이때 E가 1이 나오는 것은 Exclusive - OR의 결과와 같다.

따라서 $E = a_6 \oplus a_5 \oplus a_4 \oplus a_3 \oplus a_2 \oplus a_1 \oplus a_0 \oplus P$

1010
1011 1100

10

(a) $6_{10} = 0110_2$
$5_{10} = 0101_2$

$$
\begin{array}{r}
01 \\
0110 \\
+\,0101 \\
\hline
1011
\end{array}
$$

⇒ 최상위 비트의 캐리는 0, 그 직전의 캐리는 1 이며 overflow가 발생했다.

(b) $-7_{10} = 1001_2$
$-4_{10} = 1100_2$

$$
\begin{array}{r}
10 \\
1001 \\
+\,1100 \\
\hline
(1)0101
\end{array}
$$

⇒ 최상위 비트의 캐리는 1, 그 직전의 캐리는 0 이며 overflow가 발생했다.

(c) 양수와 양수를 더하는 경우, 양수의 부호비트는 0이므로 항상 최상위 캐리는 0이 된다.
또한 음수와 음수를 더하는 경우, 음수의 부호비트는 1이므로, 항상 최상위 캐리는 1이 된다.
모든 경우에서 최상위 비트의 캐리는 생략되며, 그 직전 캐리된 비트가 부호비트가 된다.
따라서 양수끼리의 덧셈에서 최상위 캐리는 0, 그 직전 캐리가 1인 경우, 이는 결과가 음수가 나왔단 것이며 overflow가 발생한 것이다.
마찬가지로 음수끼리의 덧셈에서도 최상위 캐리는 1, 그 직전 캐리가 0인 경우, 이는 결과가 양수라는 것을 의미하므로 overflow가 발생한다.