



# 2 수의 체계

IT CookBook, 디지털 논리회로

# 학습목표

- 10진수, 2진수, 8진수, 16진수 등의 표현 방법을 알아본다.
- 10진수, 2진수, 8진수, 16진수 등의 상호 변환방법을 알아본다.
- 2진수의 연산과 2진수 음수의 표현방법을 알아본다.
- 2진수 부동소수점의 표준 표현 방식인 IEEE 754 표준 방식을 알아본다.

01. 10진수

02. 2진수

03. 8진수와 16진수

04. 진법 변환

05. 2진 정수 연산과 보수

06. 2진 부동소수점수의 표현

## □ 10진수 표현법

- ❖ 기수가 10인 수
- ❖ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 사용

$$\begin{aligned} 9345.35 &= 9 \times 1000 + 3 \times 100 + 4 \times 10 + 5 \times 1 + 3 \times 0.1 + 5 \times 0.01 \\ &= 9 \times 10^3 + 3 \times 10^2 + 4 \times 10^1 + 5 \times 10^0 + 3 \times 10^{-1} + 5 \times 10^{-2} \end{aligned}$$

- ❖ 바빌로니아인 : 60진법을 사용(기원전 4000~3000년)
- ❖ 고대 로마의 기수법에는 5진법을 사용
- ❖ 10진법의 아라비아 숫자는 인도에서 기원전 2세기에 발명

진법을 나타내는 기본수를 기수(基數, radix)라 한다. 10이 기수인 수를 10진법, 2가 기수인 수를 2진법, 12가 기수인 수를 12진법이라 한다.

## □ 2진수 표현법

- ❖ 기수가 2인 수
- ❖ 0, 1 사용

$$\begin{aligned} 1010.1011_{(2)} &= 1 \times 1000_{(2)} + 0 \times 100_{(2)} + 1 \times 10_{(2)} + 0 \times 1_{(2)} \\ &\quad + 1 \times 0.1_{(2)} + 0 \times 0.01_{(2)} + 1 \times 0.001_{(2)} + 1 \times 0.0001_{(2)} \\ &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} \end{aligned}$$

# 03 8진수와 16진수

## □ 8진수 표현법

❖ 0에서 7까지 8개의 수로 표현

$$\begin{aligned} 607.36_{(8)} &= 6 \times 100_{(8)} + 0 \times 10_{(8)} + 7 \times 1_{(8)} + 3 \times 0.1_{(8)} + 6 \times 0.01_{(8)} \\ &= 6 \times 8^2 + 0 \times 8^1 + 7 \times 8^0 + 3 \times 8^{-1} + 6 \times 8^{-2} \end{aligned}$$

## □ 16진수 표현법

❖ 0에서 9, A(a)에서 F(f)까지 16개의 기호로 표현

$$\begin{aligned} 6C7.3A_{(16)} &= 6 \times 100_{(16)} + C \times 10_{(16)} + 7 \times 1_{(16)} + 3 \times 0.1_{(16)} + A \times 0.01_{(16)} \\ &= 6 \times 16^2 + C \times 16^1 + 7 \times 16^0 + 3 \times 16^{-1} + A \times 16^{-2} \end{aligned}$$

## □ 10진수에 해당하는 16진 기호

10진수	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16진수	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

## 1. 10진수-2진수 변환

- ❖ 정수부분과 소수부분으로 나누어 변환
- ❖ 정수부분은 2로 나누고, 소수부분은 2를 곱한다.
- ❖ 10진수 69.6875를 2진수로 변환하는 경우

2	나머지	2진수	2진수	정수	소수	
2	69				0.	6875
2	34 ... 1		1		X	2
2	17 ... 0		01			
2	8 ... 1		101	0.1	1.	3750
2	4 ... 0		0101		X	2
2	2 ... 0		00101	0.10	0.	7500
2	1 ... 0		000101		X	2
	0		1000101	0.101	1.	5000
	몫			0.1011	1.	0

↑ 곱셈결과 정수를 적는다.  
 ↓ 소수 부분이 0이 될때까지 계산한다.

$$69.6875_{(10)} = 1000101.1011_{(2)}$$

# 04 진법 변환

- ❖ 10진수 69.6을 2진수로 변환하는 경우
- ❖ 10진수 소수부분은 대부분의 경우 정확한 2진수로 변환이 안 된다.

2	69	나머지	2진수
2	34	... 1	1
2	17	... 0	01
2	8	... 1	101
2	4	... 0	0101
2	2	... 0	00101
2	1	... 0	000101
	0	1	1000101
	몫		

정수	소수	2진수
0.	6	
X	2	
1.	2	0.1
X	2	
0.	4	0.10
X	2	
0.	8	0.100
X	2	
1.	6	0.1001
X	2	
1.	2	0.10011

$$69.6_{(10)} = 1000101.10\ 011001100110011001....._{(2)}$$

## 2. 10진수-8진수 변환

- ❖ 10진수 69.6875를 8진수로 변환하는 경우
- ❖ 8로 나누고, 곱한다.

$  \begin{array}{r l}  8 & 69 \text{ 나머지} \\  \hline  8 & 8 \cdots 5 \\  8 & 1 \cdots 0 \\  & 0 \cdots 1 \\  & \text{몫}  \end{array}  $	$\uparrow$	<p>8진수</p> <p>5</p> <p>05</p> <p>105</p>	<p>8진수</p> <p>정수</p> <p>소수</p> <p>0. 6875</p> <hr/> <p>X 8</p> <p>5. 5000</p> <hr/> <p>X 8</p> <p>4. 0</p>	$\downarrow$	<p>곱셈결과 정수를 적는다.</p> <p>소수 부분이 0이 될때까지 계산한다.</p>
---	------------	--	--	--------------	--

$$69.6875_{(10)} = 105.54_{(8)}$$

- ❖ 10진수 69.6을 8진수로 변환한 경우

$$69.6 = 105.46314631..._{(8)}$$



## 3. 10진수-16진수 변환

- ❖ 10진수 69.6875를 16진수로 변환하는 경우

16		69	나머지	16진수
16		4	...	5
		0	...	4
		몫		

↑

16진수	정수	소수
	0.	6875
	X	16
0.B	11.	0000

곱셈결과 정수를 적는다.  
 소수 부분이 0이  
 될때까지 계산한다.

$$69.6875_{(10)} = 45.B_{(16)}$$

- ❖ 10진수 69.6을 16진수로 변환하는 경우

$$69.6_{(10)} = 45.999..._{(16)}$$



다른 진법의 경우도 같은 방법을  
이용하여 변환할 수 있다.

## 4. 2진수-8진수-16진수-10진수 상호변환

10진수	2진수	8진수	16진수
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

# 04 진법 변환

## □ 상호변환 예

$$\begin{aligned} 69.6875 &= 1000101 .1011_{(2)} \\ &= 001\ 000\ 101 .101\ 100_{(2)} \\ &= 1\ 0\ 5. 5\ 4_{(8)} \end{aligned}$$

10진 → 2진 → 8진

3자리씩 나눔

$$\begin{aligned} 69.6 &= 1000101 .1001100110\ 0110011001\ 1..._{(2)} \\ &= 001\ 000\ 101 .100\ 110\ 011\ 001\ 100\ 110\ 011\ ..._{(2)} \\ &= 1\ 0\ 5. 4\ 6\ 3\ 1\ 4\ 6\ 3..._{(8)} \end{aligned}$$

$$\begin{aligned} 69.6875 &= 1000101 .1011_{(2)} \\ &= 0100\ 0101 .1011_{(2)} \\ &= 4\ 5. B_{(16)} \end{aligned}$$

10진 → 2진 → 16진

4자리씩 나눔

$$\begin{aligned} 69.6 &= 1000101 .1001100110\ 0110011001\ ..._{(2)} \\ &= 0100\ 0101 .1001\ 1001\ 1001\ 1001\ 1001\ ..._{(2)} \\ &= 4\ 5. 9\ 9\ 9\ 9\ 9..._{(16)} \end{aligned}$$

# 04 진법 변환

## □ 상호변환 예(Cont'd)

$$367.75_{(8)} = 011\ 110\ 111.111\ 101_{(2)}$$

8진수 1자리 = 2진수 3자리

$$9A3.50F3_{(16)} = 1001\ 1010\ 0011.0101\ 0000\ 1111\ 0011_{(2)}$$

16진수 1자리 = 2진수 4자리

$$\begin{aligned} 101101.101_{(2)} &= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= 32 + 0 + 8 + 4 + 0 + 1 + 0.5 + 0 + 0.125 = 45.625_{(10)} \end{aligned}$$

$$\begin{aligned} 364.35_{(8)} &= 3 \times 8^2 + 6 \times 8^1 + 4 \times 8^0 + 3 \times 8^{-1} + 5 \times 8^{-2} \\ &= 3 \times 64 + 6 \times 8 + 4 \times 1 + 3 \times 0.125 + 5 \times 0.015325 \\ &= 192 + 48 + 4 + 0.375 + 0.078125 \\ &= 244.453125_{(10)} \end{aligned}$$

각 자릿수에 각  
진법의 누승을  
곱하여  
10진수로 변환

$$\begin{aligned} A3.D2_{(16)} &= 10 \times 16^1 + 3 \times 16^0 + 13 \times 16^{-1} + 2 \times 16^{-2} \\ &= 10 \times 16 + 3 \times 1 + 13 \times 0.0625 + 2 \times 0.00390625 \\ &= 160 + 3 + 0.8125 + 0.0078125 \\ &= 163.8203125_{(10)} \end{aligned}$$

## □ 상호변환 예(Cont'd)

$$\begin{aligned} 364.35_{(8)} &= 011\ 110\ 100.011\ 101_{(2)} \\ &= 0 \times 2^8 + 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 \\ &\quad + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} + 0 \times 2^{-5} + 1 \times 2^{-6} \\ &= 0 + 128 + 64 + 32 + 16 + 0 + 4 + 0 + 0 + 0 + 0.25 + 0.125 + 0.0625 + 0 + 0.15625 \\ &= 244.453125_{(10)} \end{aligned}$$

8진 → 2진 → 10진

$$\begin{aligned} A3.D2_{(16)} &= 1010\ 0011.1101\ 0010_{(2)} \\ &= 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &\quad + 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} + 0 \times 2^{-5} + 0 \times 2^{-6} + 1 \times 2^{-7} + 0 \times 2^{-8} \\ &= 128 + 0 + 32 + 0 + 0 + 0 + 2 + 1 + 0.5 + 0.25 + 0 + 0.0625 + 0 + 0 + 0.0078125 + 0 \\ &= 163.8203125_{(10)} \end{aligned}$$

16진 → 2진 → 10진

## 1. 2진수 양의 정수 덧셈

$0+0=0$ ,  $0+1=1$ ,  $1+0=1$ ,  $1+1=10$  (자리올림 발생)

10진수

$$\begin{array}{r} \text{Carry} \rightarrow 11 \\ 49 \\ + 58 \\ \hline 107 \end{array}$$

2진수

$$\begin{array}{r} \text{Carry} \rightarrow 0110000 \\ 00110001 \\ + 00111010 \\ \hline 01101011 \end{array}$$

8진수

$$\begin{array}{r} \text{Carry} \rightarrow 10 \\ 61 \\ + 72 \\ \hline 153 \end{array}$$

16진수

$$\begin{array}{r} \text{Carry} \rightarrow 0 \\ 31 \\ + 3A \\ \hline 6B \end{array}$$

## 2. 2진 음의 정수 표현과 보수(complement)

❖ 최상위비트(MSB)를 부호비트로 사용

양수(+) : 0      음수(-) : 1

❖ 2진 음수를 표시하는 방법

- 부호와 절대치(sign- magnitude)
- 1의 보수(1's complement)
- 2의 보수(2' s complement)

❖ 1의 보수로 변환하는 방법

• 0 → 1, 1 → 0으로 변환

00000011 → 1의 보수 = 11111100

❖ 2의 보수로 변환하는 방법

• 1의 보수 + 1 = 2의 보수

00000011 → 2의 보수 = 1의 보수 + 1 = 11111100 + 1 = 11111101

01101100 → 2의 보수 = 1의 보수 + 1 = 10010011 + 1 = 10010100

## 05 2진수 정수 연산과 보수

❖  $r$ 진법  $n$ 자릿수  $x$ 의  $r$ 의 보수  $r^n - x$

❖  $r$ 진법  $n$ 자릿수  $x$ 의  $r-1$ 의 보수  $:r^n - 1 - x$

➤ 00000011의 1의 보수 :  $2^8 - 1 - 00000011 = 11111111 - 00000011 = 11111100$

➤ 00000011의 2의 보수 :  $2^8 - 00000011 = 100000000 - 00000011 = 11111101$

❖ 양수를 보수로 바꾸면 음수

❖ 음수를 보수로 바꾸면 양수



## 2진수의 표현 방법 3가지

$b_7b_6b_5b_4b_3b_2b_1b_0$	8bit 크기이며, MSB가 부호비트 임.		
	부호와 절대치	1의 보수	2의 보수
01111111	+127	+127	+127
01111110	+126	+126	+126
01111101	+125	+125	+125
01111100	+124	+124	+124
...	...	...	...
00000011	+3	+3	+3
00000010	+2	+2	+2
00000001	+1	+1	+1
00000000	+0	+0	+0
10000000	-0	-127	-128
10000001	-1	-126	-127
10000010	-2	-125	-126
10000011	-3	-124	-125
...	...	...	...
11111100	-124	-3	-4
11111101	-125	-2	-3
11111110	-126	-1	-2
11111111	-127	-0	-1

# 05 2진수 정수 연산과 보수

## 부호와 절대치의 표현

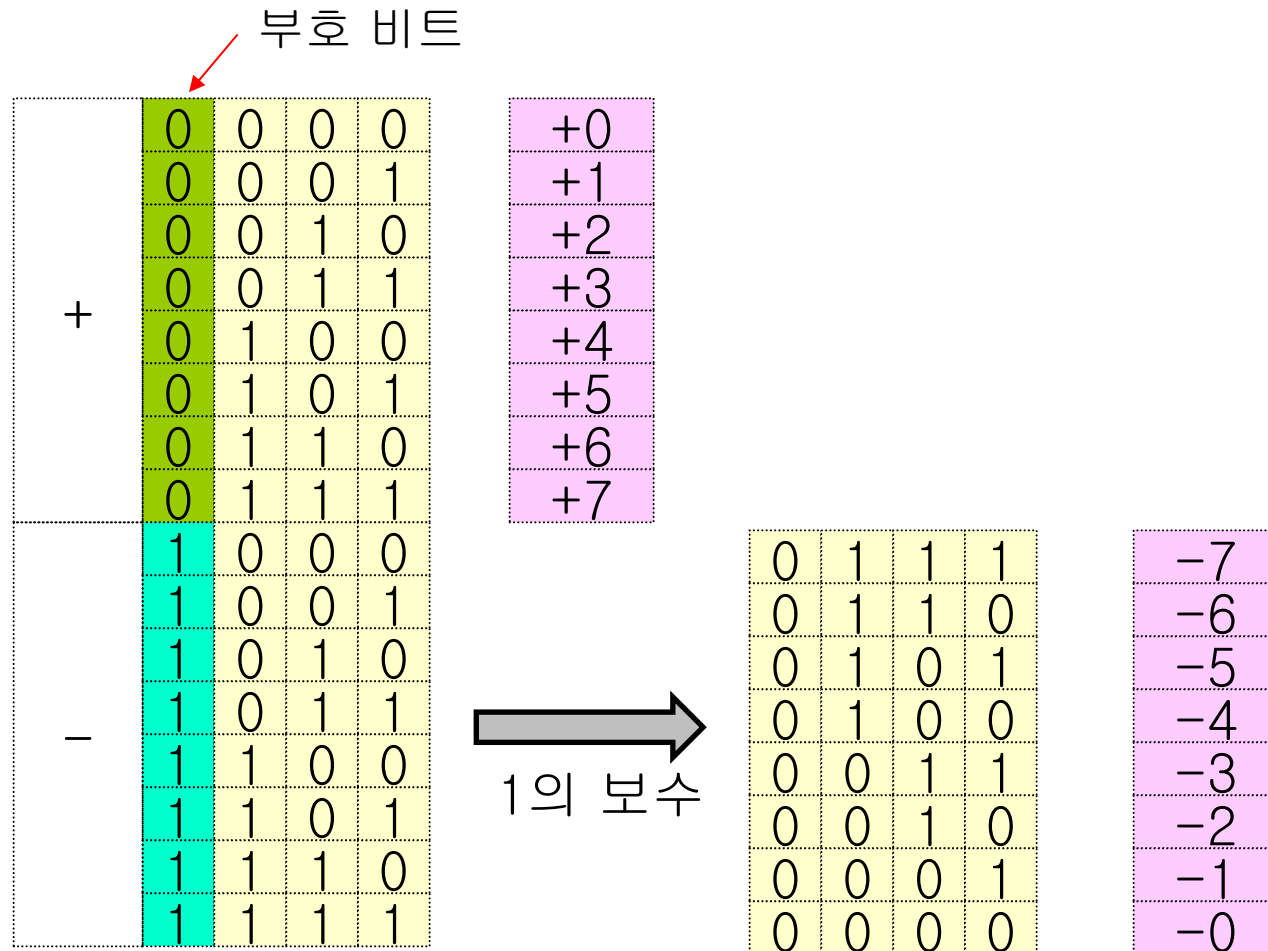
부호 비트

+	0	0	0	0	+0
	0	0	0	1	+1
	0	0	1	0	+2
	0	0	1	1	+3
	0	1	0	0	+4
	0	1	0	1	+5
	0	1	1	0	+6
	0	1	1	1	+7
-	1	0	0	0	-0
	1	0	0	1	-1
	1	0	1	0	-2
	1	0	1	1	-3
	1	1	0	0	-4
	1	1	0	1	-5
	1	1	1	0	-6
	1	1	1	1	-7

# 05 2진수 정수 연산과 보수

## 1의 보수 표현

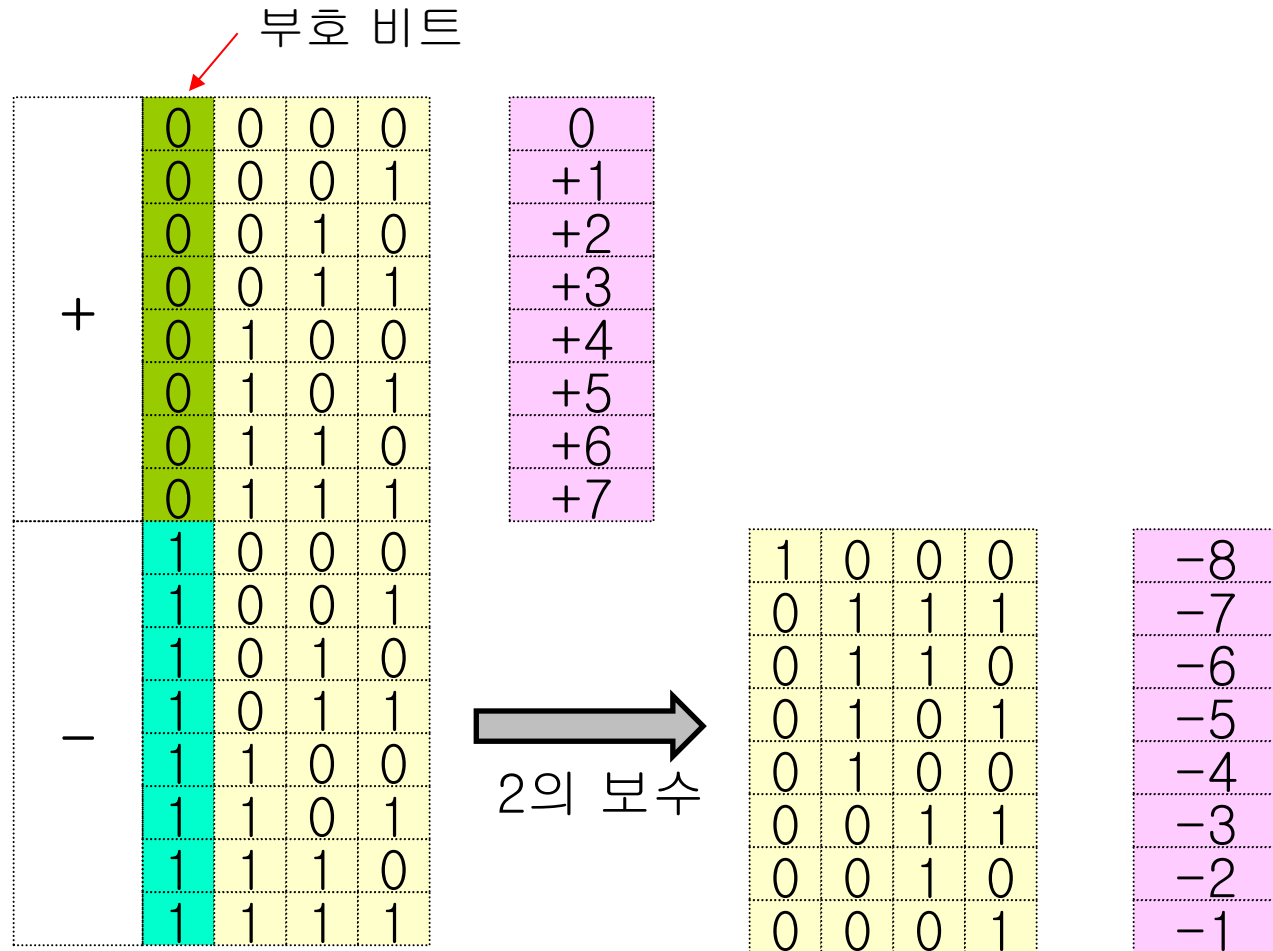
- 음수의 실제 크기는 1의 보수를 취한 후, 그 수에 음의 부호를 붙인다.



# 05 2진수 정수 연산과 보수

## 2의 보수 표현

- 음수의 실제 크기는 2의 보수를 취한 후, 그 수에 음의 부호를 붙인다.



## 05 2진수 정수 연산과 보수

❖ 뺄셈 : 보수를 취하여 더하면 뺄셈을 수행(Carry가 있으면 버림)

$$\begin{aligned} 7928 - 879 &= 7928 + (-879) \\ &= 7928 + (-0879) \\ &\Rightarrow 7928 + (10^4 - 0879) \\ &= 7928 + 9121 \\ &= 17049 \\ &\Rightarrow 7049 \end{aligned}$$

자릿수  
맞춤

### $n$ 비트 2의 보수에 대한 10진수의 표현 범위

bit 수	2의 보수를 사용한 2진 정수의 표현 범위
$n$ bit	$-2^{n-1} \sim +2^{n-1} - 1$
4 bit	$-2^{4-1} \sim +2^{4-1} - 1$ (-8 ~ +7)
8 bit	$-2^{8-1} \sim +2^{8-1} - 1$ (-128 ~ +127)
16 bit	$-2^{16-1} \sim +2^{16-1} - 1$ (-32768 ~ +32767)
32 bit	$-2^{32-1} \sim +2^{32-1} - 1$ (- 2147483648 ~ + 2147483647)

## 3. 부호 확장

❖ 부호 확장이란 늘어난 비트 수 만큼 부호를 늘려주는 방법

### 2진수 표현 방법에 따른 부호 확장

2진수 표현 방법	부호 확장 방법	예		
			8bit	16bit
부호와 크기	부호만 MSB에 복사하고, 나머지는 0으로 채움	양수	00101010	00000000 00101010
		음수	10010111	10000000 00010111
1의 보수	늘어난 길이만큼 부호와 같은 값으로 모두 채움	양수	00101010	00000000 00101010
		음수	10010111	11111111 10010111
2의 보수	늘어난 길이만큼 부호와 같은 값으로 모두 채움	양수	00101010	00000000 00101010
		음수	10010111	11111111 10010111

## 4. 2의 보수로 표현된 음수를 10진수로 변환하기

(2의 보수 10101100을 10진수로 변환하는 경우)

첫 번째 방법

MSB가 1이므로 음수이다. 실제크기는 -128이다.

$$\begin{aligned} 10101100_{(2)} &= -1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 \\ &= -128 + 0 + 32 + 0 + 8 + 4 + 0 + 0 = -128 + 44 \\ &= -84 \end{aligned}$$

두 번째 방법

2의 보수로 바꾸어 10진수로 바꾼 다음 -부호를 붙인다.

$$\begin{aligned} 10101100_{(2)} &\Rightarrow 2\text{의 보수 } 01010100_{(2)} \\ &= 0 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 \\ &= 0 + 64 + 0 + 16 + 0 + 4 + 0 + 0 \\ &= 84 \end{aligned}$$

➡ - 부호를 붙이면 -84

# 05 2진수 정수 연산과 보수

## 5. 2의 보수 연산 (8bit)

양수+양수=양수 (49+58=107)	큰 수-작은 수=양수 (58-49=9)	작은 수 - 큰 수= 음수 (49-58=-9)
Carry → <u>0</u> 110000 00110001 + 00111010 <hr/> <u>0</u> 01101011	Carry → <u>1</u> 111110 00111010 - 00110001 <hr/> 00111010 + 11001111 <hr/> <u>1</u> 00001001 서로 같은	Carry → <u>0</u> 000000 00110001 - 00111010 <hr/> 00110001 + 11000110 <hr/> <u>0</u> 11110111
음수 + 음수= 음수 (-49-58=-107)	큰 양수 + 큰 양수 = 음수 (98+74=-84) 비정상	큰 음수 + 큰 음수 = 양수 (-98-74=+84) 비정상
Carry → <u>1</u> 001110 - 00110001 - 00111010 <hr/> 11001111 + 11000110 <hr/> <u>1</u> 10010101	Carry → <u>1</u> 000010 01100010 + 01001010 <hr/> <u>0</u> 10101100 서로 다름	Carry → <u>0</u> 111110 - 01100010 - 01001010 <hr/> 10011110 + 10110110 <hr/> <u>1</u> 01010100

overflow