

정수(Integer)의 표현

1. 일반적으로 1 바이트, 2 바이트, 4 바이트, 8 바이트에 저장
2. 부호 있는 정수(signed)와 부호 없는 정수(unsigned)로 나뉜다
3. 부호가 있는 경우 첫 bit가 부호를 나타냄(0 : 양수, 1 : 음수)

정수 표현 범위

Data type	Size	Range
unsigned char	1 byte	0 ~ 255
char	1 byte	-128 ~ 127
short	2 byte	-32,768 ~ 32,767
unsigned short	2 byte	0 ~ 65,535
int	4 byte	-2,147,483,648 ~ 2,147,483,647
unsigned int	4 byte	0 ~ 4,294,967,296

양의 정수 표현

1. 부호 비트는 0
2. 정수의 2진수로 표현 나머지는 0으로

예) 43을 1 바이트로 표현

$43 = 0010\ 1011_2$ 이므로

0010 1011

0x2b

| 음의 정수 표현

음의 정수 표현

1. 부호 비트는 1
2. 정수를 2의 보수(two's complement)로 저장

$43 \rightarrow 0010\ 1011_2 \rightarrow 1101\ 0100_2$ (1의 보수)
 $\rightarrow 1101\ 0101_2$ (2의 보수)

0010 1011₂



모든 비트를 반전

1101 0100₂

1의 보수



1을 더한다

1101 0101₂

2의 보수

| 음의 정수 표현

음의 정수를 2의 보수로 표현하는 이유

1. 2의 보수를 사용하지 않는다면

0000 0000과 1000 0000 모두 0을 표현

1) 1 비트 낭비

2) if 문으로 비교할 때 엉뚱한 결과가 나옴

| 음의 정수 표현

음의 정수를 2의 보수로 표현하는 이유

2. 정수의 뺄셈 과정

예) $43 - 25 = 18$

1. 25를 -25로 변환

2. $43 + (-25)$ 로 연산

0010 1011₂ 43

+ 1110 0111₂ -25

Carry는 버린다

~~1~~ 0001 0010₂ 18

4bit 연산 – 곱셈

$$\begin{array}{r}
 1010 \\
 \times 0101 \\
 \hline
 110010
 \end{array}$$

	0	0	0	0	0	1	0	1	
+ 1010	1	0	1	0					
	0	1	0	1	0	0	1	0	shift 1(> >)
	0	0	1	0	1	0	0	1	shift 2
+ 1010	1	1	0	0					
	0	1	1	0	0	1	0	0	shift 3
	0	0	1	1	0	0	1	0	shift 4

4bit 연산 - 나눗셈

0011 $\overline{)1011}$

subtract
2의 보수

0001
+1101

1110

음수
0011 > 0001

0010
+1101

1111

음수일 땀
비교만 하고
버린다.

0101
+1101

~~1~~0010

Replace!

0	0	0	0	1	0	1	1	shift 1(<<)
0	0	0	1	0	1	1	?	
0	0	0	1	0	1	1	0	shift 2
0	0	1	0	1	1	0	?	
0	0	1	0	1	1	0	0	shift 3
0	1	0	1	1	0	0	?	
0	0	1	0	1	0	0	1	shift 4
0	1	0	1	0	0	1	?	
0	0	1	0	0	0	1	1	
나머지				몫				