Chapter 2. 데이터의 표현

01 : 진법과 진법 변환

02 : 정수 표현

03 : 실수 표현

04 : 디지털 코드

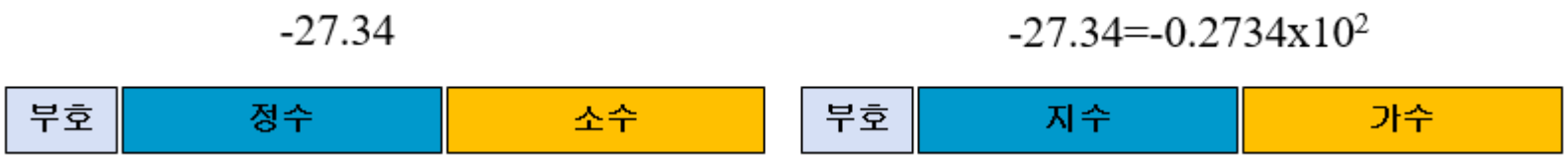
05 : 에러 검출 코드

고정 소수점 수, 부동 소수점 수
(fixed point number, floating point number)

고정 소수점 수 : 소수점의 위치가 고정되어 표현
부동 소수점 수 : 소수점의 위치가 수 크기에 따라 이동

- 컴퓨터에서는 모든 데이터를 0과 1로 표현 따라서 부호도 양수(+)는 0, 음수(-)는 1로 표현

- 부호 비트는 최상위 비트(Most Significant Bit)에 표현



2진수의 음수와 양수의 표현

음수와 양수를 표현하는 3가지 방식

1. 부호와 절대값 표현
2. 1의 보수 사용
3. 2의 보수 사용(실제로 컴퓨터에서 사용하는 방법)

□ 부호와 절대값

+27 표현

0	0	0	1	1	0	1	1
부호 (+)	27						

−27 표현

1	0	0	1	1	0	1	1
부호 (−)	27						

보수(complement)

- r-1의 보수 : $r^n - 1 - N$ (r은 진수, n은 숫자의 자릿수, N은 원래의 수)
- r의 보수 :

$$N \neq 0, \quad r^n - N$$

$$N = 0, \quad 0$$

- 10진수 : 10의 보수, 9의 보수
- 2진수 : 2의 보수, 1의 보수

345_{10}의 10의 보수 : $r^n - N$

$10^3 - 345 = 655$

345_{10}의 9의 보수 : $r^n - N - 1$

$10^3 - 345 - 1 = 654$

0110_2의 2의 보수

$2^4 - 0110_2 = 1010_2$

0110_2의 1의 보수

$2^4 - 0110_2 - 1_2 = 1001_2$

1의 보수를 쉽게 찾는 법
주어진 2진수의 1의 보수는 1은 0으로 0은 1로 바꾸면 됨
(예제)
0110의 1의 보수는 1001

보수(complement)

- 부호없는 숫자의 뺄셈(M-N)
  - 피감수 M에 감수 N에 대한 r의 보수를 더한다.
  - M>=N 이라면, r^n 을 무시하고 결과로 M-N을 얻는다.(양의 결과)
  - M<N이면, 자리올림수가 발생하지 않으므로 구한 값의 r의 보수를 취하고 마이너스 부호를 붙인다.(음의 결과)

□ 1의 보수 방식

27 표현

0	0	0	1	1	0	1	1
부호 (+)	27						

−27 표현

1	1	1	0	0	1	0	0
부호 (−)	1의 보수 27						

□ 2의 보수 방식

27 표현

0	0	0	1	1	0	1	1
부호 (+)	27						

−27 표현

1	1	1	0	0	1	0	1
부호 (−)	2의 보수 27						

수의 표현 방법에 따른 10진수 대응 값

4비트 2진수	부호 없는 수	부호와 절대값	1의 보수	2의 보수
0000	0	0	0	0
0001	1	1	1	1
0010	2	2	2	2
0011	3	3	3	3
0100	4	4	4	4
0101	5	5	5	5
0110	6	6	6	6
0111	7	7	7	7
1000	8	−0	−7	−8
1001	9	−1	−6	−7
1010	10	−2	−5	−6
1011	11	−3	−4	−5
1100	12	−4	−3	−4
1101	13	−5	−2	−3
1110	14	−6	−1	−2
1111	15	−7	−0	−1

n 비트 2의 보수에 대한 10진수의 표현 범위

비트 수	2의 보수를 사용한 2진 정수의 표현 범위
n 비트	$-2^{n-1} \sim +2^{n-1}+1$
4 비트	$-2^{4-1}(-8) \sim +2^{4-1}+1(+7)$
8 비트	$-2^{8-1}(-128) \sim +2^{8-1}+1(+127)$
16 비트	$-2^{16-1}(-32,768) \sim +2^{16-1}+1(+32,767)$
32 비트	$-2^{32-1}(-2,147,483,648) \sim +2^{32-1}+1(+2,147,483,647)$
64 비트	$-2^{64-1}(-9,223,372,036,854,775,808) \sim +2^{64-1}+1(+9,223,372,036,854,775,807)$

char : -128 ~ +127
unsigned char : 0 ~ 255
int : -2,147,483,648 ~ +2,147,483,647

부호확장

8비트 수를 16비트의 수로 비트 수를 늘리는 방법

2진수 표현 방식	부호 확장 방법	구분	8비트	16비트 확장
부호와 절대값	부호만 MSB에 복사하고 나머지는 0으로 채운다	양수	00101010	00000000 00101010
		음수	10010111	10000000 00010111
1의 보수	늘어난 길이만큼 부호와 같은 값으로 모두 채운다	양수	00101010	00000000 00101010
		음수	10010111	11111111 10010111
2의 보수	늘어난 길이만큼 부호와 같은 값으로 모두 채운다	양수	00101010	00000000 001010100
		음수	10010111	11111111 10010111

2진 정수 연산(8비트 연산의 예)

양수	49			0	0	1	1	0	0	0	1
+양수	+58	=	+	0	0	1	1	1	0	1	0
양수	107	=	0	0	1	1	0	1	0	1	1

양수	58			0	0	1	1	1	0	1	0
−양수	−49	=	−	0	0	1	1	0	0	0	1
양수				0	0	1	1	1	0	1	0
+음수		=	+	1	1	0	0	1	1	1	1
양수	9		1	0	0	0	0	1	0	0	1

양수	49		0 0 1 1 0 0 0 1
−양수	−58	=	− 0 0 1 1 1 0 1 0
양수			0 0 1 1 0 0 0 1
+음수		=	+ 1 1 0 0 0 1 1 0
양수	9		0 1 1 1 1 0 1 1 1

−양수	−49		− 0 0 1 1 0 0 0 1
−양수	−58	=	− 0 0 1 1 1 0 1 0
음수			1 1 0 0 1 1 1 1
+음수		=	+ 1 1 0 0 0 1 1 0
음수	−107		1 1 0 0 1 0 1 0 1

overflow(연산의 자릿수 넘침)

양수	98		0 1 1 0 0 0 1 0
+양수	+74	=	+ 0 1 0 0 1 0 1 0
양수	−84	=	0 1 0 1 0 1 1 0 0

결과가 비정상으로 나옴(8비트의 표현범위 넘침)

−양수	−98		− 0 1 1 0 0 0 1 0
−양수	−74	=	− 0 1 0 0 1 0 1 0
음수			1 0 0 1 1 1 1 0
+음수		=	+ 1 0 1 1 0 1 1 0
음수	+84		1 0 1 0 1 0 1 0 0

결과가 비정상으로 나옴(8비트의 표현범위 넘침)

10진수 23+47을 이진수 덧셈으로 표현하시오

10진수 23-47을 이진수로 표현하시오