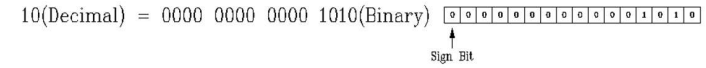
16. 음수의 표현 방법

음수를 표현하기 위해 먼저 알아두어야 할게 있는데 그것은 보수라는 용어이다.
보수란게 무엇인가?
보수란 주어진 값의 최대 자릿수(567 = 3자리수, 23 = 2자릿수,111(이진수) = 3자릿수)의 n승 3자릿수의 n승은 10^3 = 1000, 10^2 = 100, 2^3 = 1000 이 되기 위해 필요한 값이다.
10진수 567의 10의 보수가 무엇인가?
567(10진수 3자릿수) → 10^3 -567(Decimal) → 1000(Decimal) - 567(Decimal) = 9 9 10(Decimal) - 567(Decimal) = 433(10진수 567에 대한 10의 보수)
2진수 111의 2의 보수가 무엇인가?
111(2진수 3자릿수) → 2^3 - 111(Binary) → 1000(Binary) - 111(Binary) = 1 1 2(Binary) - 111(Binary) = 001(2진수 111에 대한 2의 보수)

실제 CPU에서는 80286(16Bit Processor), 80386(32Bit Processor), 80486(64Bit Processor), 80586(128Bit Processor) 등이 있기 때문에
4Bit만 처리하고 싶어도 CPU에서는 처리하는 Processor Bit 만큼 자동으로 채워진다.

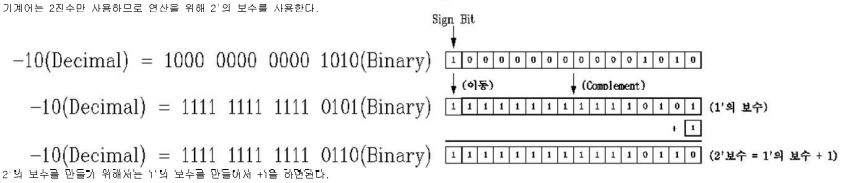
따라서 PLC CPU도 Processor Bit에 따라 쓰고 싶지 않아도 나머지 Bit는 자동으로 할당이 된다.

16Bit의 제일 마지막 Bit를 음수(1)와 양수(0)를 구분하는 Sign Bit로 사용한다.



그러면 -10 이면 Sign Bit만 1로 만들면 된다고 생각할지도 모르지만 틀렸다.

기계어는 2진수만 사용하므로 연산을 위해 2'의 보수를 사용한다.



2'의 보수를 만들기 위해서는 1'의 보수를 만들고서 +1을 하면된다.

그러면 왜 보수가 필요한가?

기계어는 16진수나 10진수, 8진수를 2진수 여러 개를 합치거나 변환하여 사용하는 것이지 직접 사용하지는 못한다.
기계가 인식하는 숫자는 1 아니면 0 이다.
내 컴퓨터는 다 인식하는데? 라고 말하는 사람이 있을까? 있겠지.
하지만 그건 기계를 너무 과대 평가한 것이다. 기계는 전기가 통하느냐 통하지 않느냐만 인식하기 때문에 전기가 통하는 1과 전기가 통하지 않는 0만 인식한다.
이걸 조합해서 사람이 알아볼 수 있게 만들었을 뿐이다.

하지만 0과 1만을 사용해서 연산을 처리할 경우 덧셈은 쉽게 처리가 되는데 뺄셈은 쉽지가 않았다.
예를들어 10 - 1 과 100 - 1, 1000 - 1, 10000 -1 등의 연산 처리에서 경우의 수가 다 달랐다.
이때 보수를 이용한 연산 처리 방식을 알게됐고, Gate에 이용하여 쉽게 뺄셈 연산을 하게됐다.
만일 보수를 사용하지 않고도 Gate만으로 뺄셈을 보수를 이용한 연산 방식보다 쉽게 처리할 수 있다면 컴퓨터의 처리 속도는 더욱 빨라질 것이다.