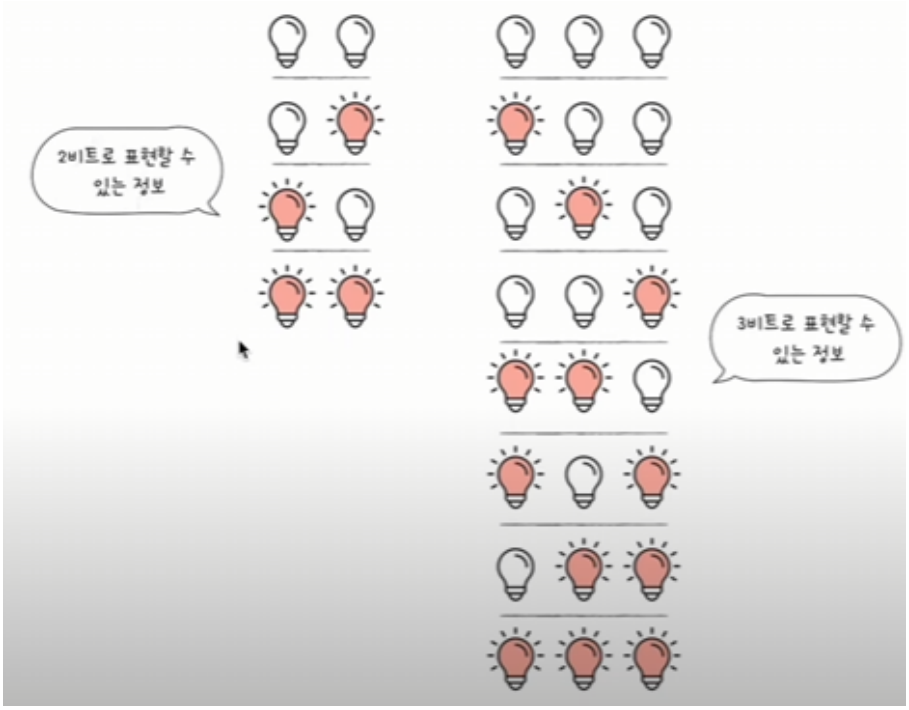
4강. 0과 1로 숫자를 표현하는 방법

용어 정리

정보 단위

비트 (bit)는 0과 1을 표현하는 가장 작은 정보 단위입니다.

다음 예시는 2비트와 3비트로 표현할 수 있는 정보입니다.



해당 예시를 일반화해보면, 다음과 같습니다.

- n비트로 2^n가지의 정보를 표현할 수 있습니다.
- 프로그램은 수많은 비트로 이루어져 있습니다.
- 평소에는 비트 단위보다는 byte, kilobyte, megabyte, gigabyte, terabyte 단위로 표현합니다.

1바이트(1byte)	8비트(8bit)
1킬로바이트(1kB)	1,000바이트(1,000byte)
1메가바이트(1MB)	1,000킬로바이트(1,000kB)
1기가바이트(1GB)	1,000메가바이트(1,000MB)
1테라바이트(1TB)	1,000기가바이트(1,000GB)

워드 (word)

워드는 CPU가 한 번에 처리할 수 있는 정보의 크기 단위입니다.

단순한 예시로, n bit 프로세서 (64bit / 32bit) 라고 이야기 할 때, 해당 n이 바로 컴퓨터의 WORD인 셈입니다.

위 예시와 다르게, 일반적으로는 WORD는 32bit를 나타내는 말입니다.

기술의 발전으로 위 예시처럼 다양한 단위의 WORD가 등장하게 되었습니다.

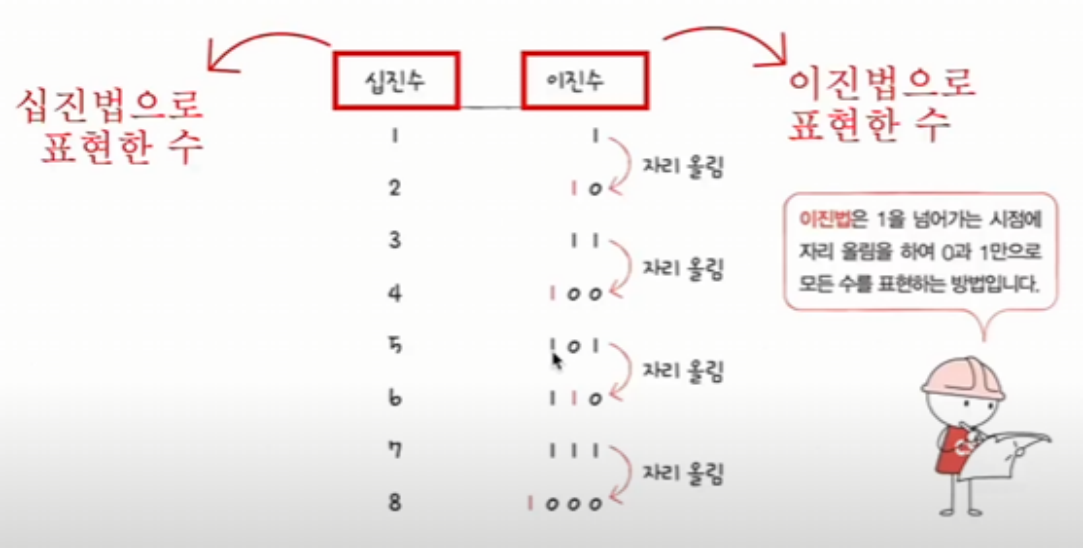
- word : 32bit
- Half-word : 16bit
- Double-word : 64bit

이진법 (binary)

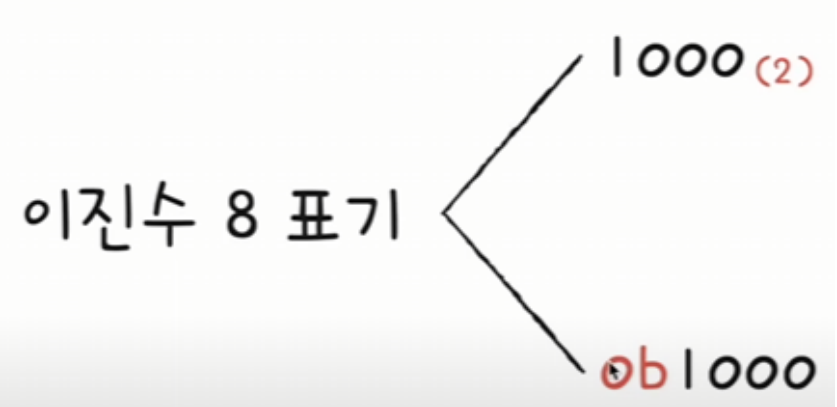
이진법은 0과 1로 수를 표현하는 방법입니다.

숫자가 1을 넘어가는 시점에 자리를 올립니다.

일반적으로 십진법을 사용한다고 했을 때, 숫자가 9를 넘어갈 때 자리올림을 합니다.

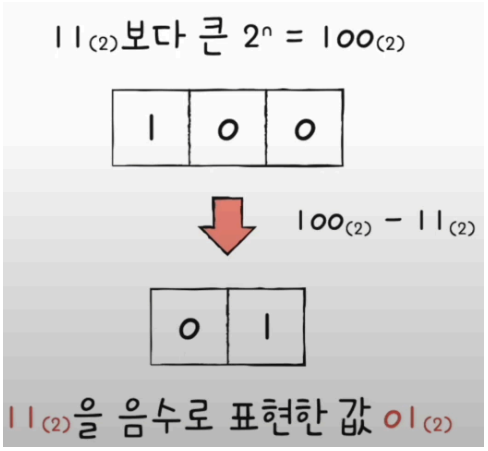


이진수의 표기법은 다음과 같습니다.



0과 1로 음수 표현하기

- 2의 보수
- 어떤 수를 그보다 큰 2^n에서 뺀 값



2의 보수를 사용하는 이유

- 단일 0 표현

  +0 과 -0 처럼 두 개로 표현되는 1의 보수 방식과 달리, 2의 보수는 0이 하나만 존재합니다.

- 산술 연산 간편화

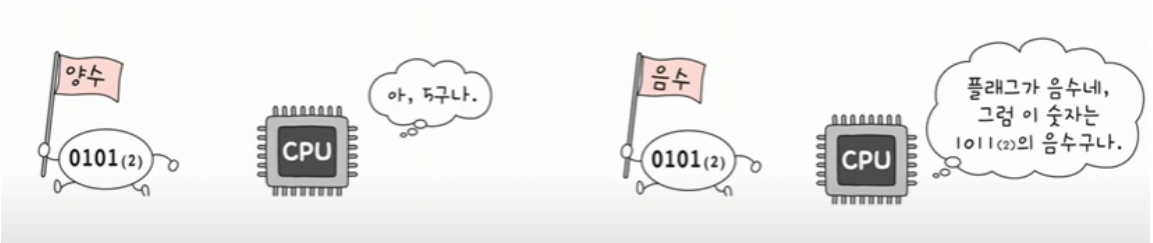
  컴퓨터에서 덧셈과 뺄셈 연산을 하나의 연산(덧셈)으로 처리할 수 있어 하드웨어를 단순하게 설계할 수 있습니다.

- 음수 표현

  2의 보수에서 가장 왼쪽의 비트(MSB)가 1이면 음수, 0이면 양수를 나타내어 부호 비트 역할을 합니다.

이진법을 음수로 표현하는 데 어떻게 구분할까?

- 플래그 (flag)

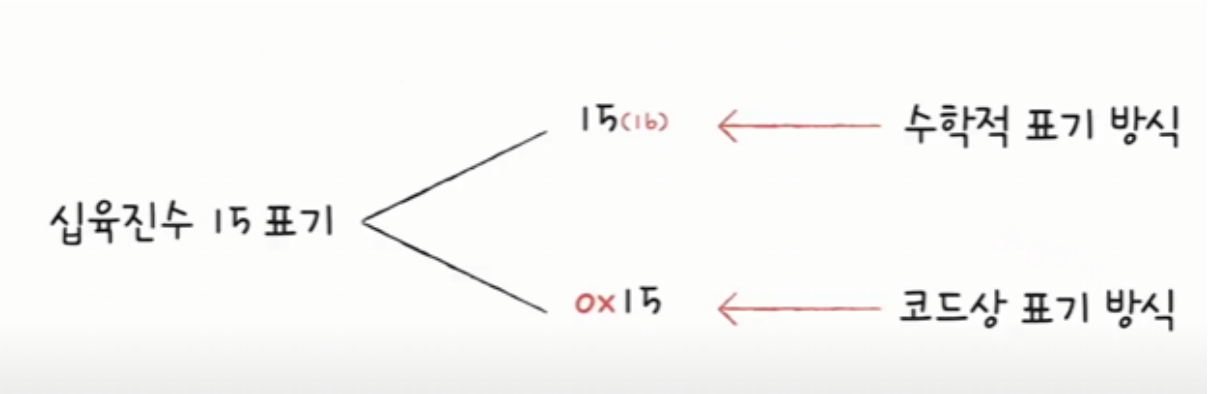


16진법

이진법으로 표현하기에는 숫자가 너무 길어지는 단점이 있습니다.

그래서 컴퓨터의 데이터를 표현할 때 16진법도 많이 사용합니다.

십육진법의 표기법은 다음과 같습니다.

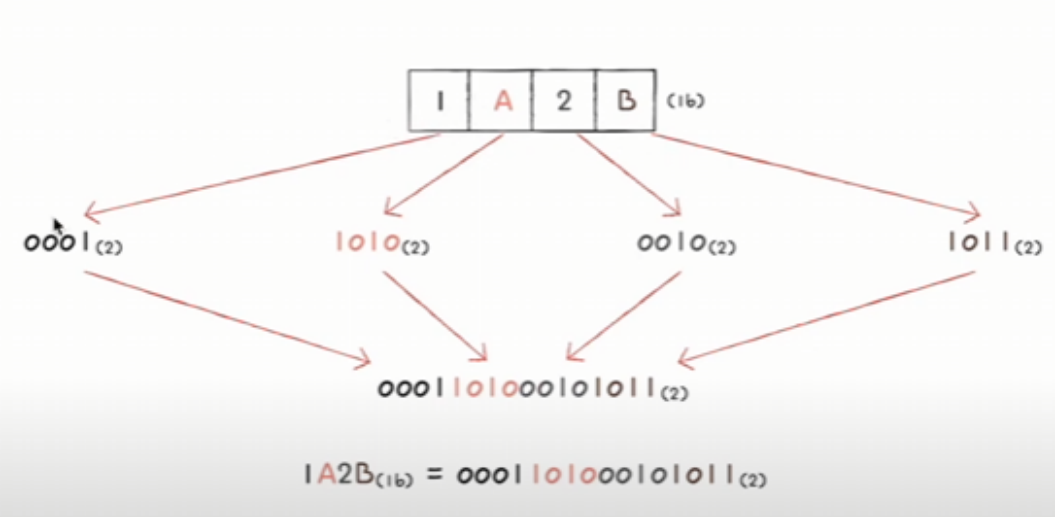


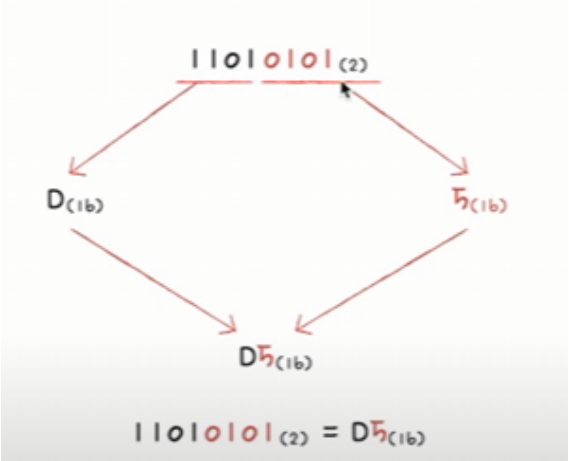
직접 코드에 쓰는 사례도 있습니다.

```
offset = __mem_to_opcode_arm(*(u32 *)loc);
offset = (offset & 0x00ffffff) << 2;
if (offset & 0x02000000)
offset -= 0x04000000;
offset += sym->st_value - loc;
```

그렇다면 왜 하필 십진법이 아닌 십육진법을 사용하는 걸까?

저자 생각으로는 이진수와 십육진수의 변환이 매우 쉽기 때문입니다.





그리고 데이터 크기 표현의 효율성이 있습니다.

- 32비트 데이터의 경우 → 2진수로는 32자리가 되지만, 이를 16진수로 바꾸면 단 8자리의 문자로 표현할 수 있습니다.

그리고 가독성과 디버깅의 편의성이 있습니다.