

02. Binary Expression

yongzzai

Summary of Chap.02 of book [1]

1 컴퓨터가 이해하는 정보 단위

- * 컴퓨터가 이해하는 가장 작은 정보 단위를 **비트(bit)**라고 함.
- * 1 bit는 2가지 정보 (0또는 1)을 나타낼 수 있음.
- * 2 bit는 4가지 정보 (00, 01, 10, 11)을 표현할 수 있음.
- * 결과적으로 n bit로는 2^n 가지의 정보를 표현할 수 있음.

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)

c.f.) 이전 단위를 1024개씩 묶은 단위는 kiB, MiB, GiB, ...

- * 일반적으로 비트 단위보단 kB, MB, GB, TB 단위를 사용함.

1.1 워드(word)

- * 워드란 CPU가 한번에 처리할 수 있는 정보의 크기 단위를 의미함.
- * 예를 들어, 어떤 CPU가 한번에 32bit씩 처리할 수 있다면 해당 CPU의 1word = 32bit를 의미함.
- * 파생된 워드 단위도 있음.
 - 1) 하프 워드 (half word): 워드의 절반크기
 - 2) 풀 워드 (full word): 워드의 크기
 - 3) 더블 워드 (double word): 워드의 두배 크기

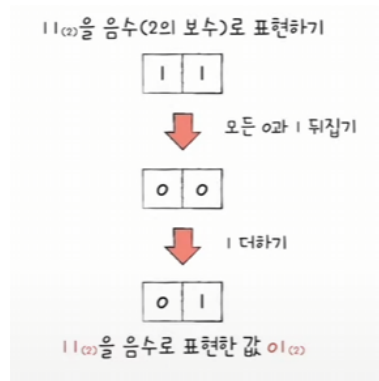
2 숫자표현방법

2.1 이진법

- * 이진법은 0과 1로 수를 표현하는 방법
- * 일반적으로 사용된 십진법은 숫자가 9를 넘어가면 자리올림을 하는 방법이고, 이진법은 숫자가 1을 넘어가면 자리올림함.
- * 이진수를 표기할 때, 예를 들어, $10 + 1000$ 이 이진수로 표현한건지, 십진수로 표현한 $10 + 1000$ 인지를 구분하기 어려움. 그렇기때문에 보통 이진수를 표기할 땐, 아랫첨자 $1000_{(2)}$ 를 표기하거나, $0b1000$ 으로 표기함.

2.2 음수 표기법

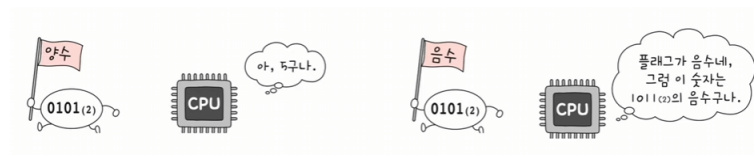
- * 0과 1로 음수를 표기할 땐, **2의 보수법**을 사용함. **2의 보수법**이란, 어떤 수를 그보다 큰 2^n 에서 뺀 값을 의미함.



- * $11_{(2)}$ 를 음수로 표현하고 싶을 때, 2^n 보다 위와 같이 간단하게 표현할 수 있음.

2.3 플래그 (Flag)

- * $-1011_{(2)}$ 를 표현하기 위한 $0101_{(2)}$ 와 십진수 5를 표현하기 위한 $0101_{(2)}$ 은 동일하게 생겨서 구분을 위해 **플래그 (Flag) 레지스터**가 CPU 내부에 존재함.



2.4 십육진법

* 이진법을 사용하면 숫자의 길이가 너무 길어진다는 단점이 있음. 예를 들어, 숫자 32는 이진수로 100000이 됨. 따라서 컴퓨터의 데이터를 표현할 때, 십육진법도 자주 활용이됨 (수가 15를 넘어가는 시점에 자리올림).

십진수	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	...
십육진수	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	...

자리올림

* 십육진수에선 10=A, 11=B, 12=C, ..., 15=F로 표기하고 수가 16이 되면 10으로 표현함.

* 이진수와 마찬가지로, 십육진수에서 15를 표기할 땐, 15_{16} 또는 $0x15$ 로 표기함.

3 문자 표현 방법

3.1 문자 집합과 인코딩

1. 문자 집합 (Character set)

문자 집합이란 컴퓨터가 이해할 수 있는 문자의 모음임. 문자 집합에 포함되지 않는 문자는 컴퓨터가 이해할 수 없음

2. 인코딩 (Encoding)

문자집합에 속한 문자를 0과 1로 이루어진 문자 코드로 변환하는 과정을 **인코딩**이라고 함.

3. 디코딩 (Decoding)

0과 1로 표현된 문자코드를 사람이 이해할 수 있는 문자로 변환하는 과정을 **디코딩**이라고 함.

3.2 아스키 코드

* 가장 대표적인 문자집합, 인코딩 방식임.

* 기본적으로 알파벳과 아라비아 숫자, 일부 특수문자 및 제어 문자(ex. 스페이스바)를 포함함.

* 8bit로 하나의 문자를 표현하는데 그중 7bit는 문자를 표현하기 위해 사용하고 나머지 1bit는 **패리티 비트 (parity bit)**라고 불리우며 오류 검출을 위해 사용됨.

* 하지만 아스키코드는 7비트로 문자를 표현하기 때문에 128개보다 많은 문자를 표현할 수 없음. → 영어를 제외한 다른 언어는 인코딩이 안되서 언어별 인코딩이 등장하였음. (참고로 한글 인코딩은 완성형 인코딩 방식과 조합형 인코딩 방식이 있음)

3.3 유니코드 문자집합과 UTF 인코딩 방식

* **유니코드**는 통일된 문자집합으로 한글, 영어, 특수문자, 이모티콘까지 표기가 가능함.

* **utf**는 유니코드를 인코딩하기 위해 사용되는 인코딩 방법 중 하나로, utf-8, utf-16, utf-32 등이 있음.

References

[1] Minchul Kang. 혼자 공부하는 컴퓨터 구조+운영체제. Hanbit Media, 2022.