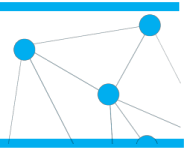
CHAPTER

02 정보의 표현과 디지털 논리

2.1 디지털 정보
2.2 정보의 표현
2.3 부울 대수와 디지털 논리
2.4 정보의 압축과 오류

2.1 디지털 정보



- 아날로그 정보와 디지털 정보
- 정보의 디지털화

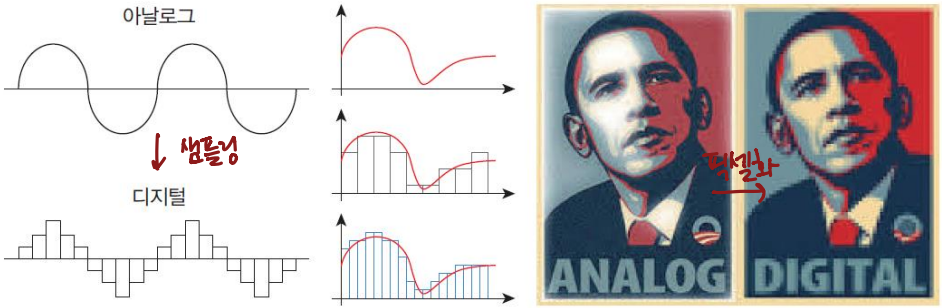
2.1.1 아날로그 정보와 디지털 정보

- 아날로그와 디지털의 예
  - 아날로그 데이터: 연속적(Continuous)인 데이터
  - 디지털 데이터: 비연속적(Discrete)인 데이터

| 그림 2-1 시계와 체중계의 아날로그 방식과 디지털 방식

  - 숫자뿐만 아니라 사운드, 이미지, 동영상과 같은 다양한 데이터도 아날로그 방식의 표현과 디지털 방식의 표현 존재

- 아날로그에서 디지털로 변화 디지털화
  - 실세계 데이터는 원래 아날로그 데이터
    - 예) 숫자 "3", 문자 "사과"는 쓰는 사람에 따라 다양함
    - 디지털 세계에서는 글자는 모두 동일한 방식으로 표현, 저장



(a) 사운드의 디지털화 (b) 이미지의 디지털화

  - 아날로그 데이터나 정보는 모두 적절한 과정을 거쳐 디지털 데이터로 변화
    - 원래 아날로그 데이터에 얼마든지 가깝게 디지털화 가능

- 디지털 데이터의 이점
  - 한가지 방식으로 표현 ⇒ 정보의 호환성이 높아짐
  - 컴퓨터의 회로는 디지털 정보를 처리, 저장하기 쉽게 설계
  - 데이터 전송 오류나 손실의 문제점 최소화
  - 문자, 숫자, 이미지, 사운드, 동영상 등이 아날로그 형태로 존재할 때는 그 특성이 상이하여 저장, 처리, 전송 방식이 모두 다름 ⇒ 디지털 데이터는 동일하게 다룸

퀴즈

2.1.2 정보의 디지털화

- 이진법(Binary system)
  - "0"과 "1"의 두 가지 상태
  - 이진법의 예
    - "On", "Off"
    - "True", "False"
    - "컵에 물이 차 있음", "컵에 물이 비어 있음"
    - "5 볼트", "0 볼트"

(a) "1" 상태 (b) "0" 상태 (c) 존재하지 않는 상태

| 그림 2-5 컵에 물이 차 있는 정도를 "1"과 "0"로 표현

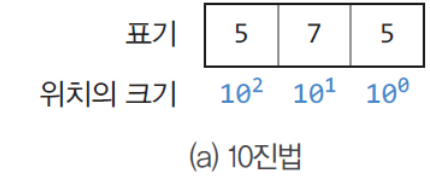
- 비트(Bit), 바이트(Byte), 워드(Word)
  - 비트: 0 또는 1과 같은 가장 작은 정보를 나타내는 단위
  - 바이트: 비트 8개의 모임, 영어 한 글자를 표현
    - 예) 비트 패턴 "10011100"
  - 워드: 컴퓨터에 따라 상이함
    - 예) 16비트, 32비트 등

1KB(Kilo Byte) = 2^{10} Byte = 1,024 Byte 10^3

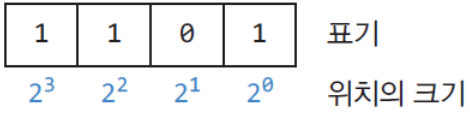
1MB(Mega Byte) = 2^{20} Byte = 1,048,576 Byte(약 백만 바이트) 10^6

1GB(Giga Byte) = 2^{30} Byte = 1,073,741,824 Byte(약 10억 바이트) 10^9

- 진법의 변환 – 10진법, 16진법
  - 10진법
    - 숫자는 4세기 고대 이집트 문명과 인도에서 시작
  - 위치기수법(Positional number system)



(a) 10진법



(b) 2진법

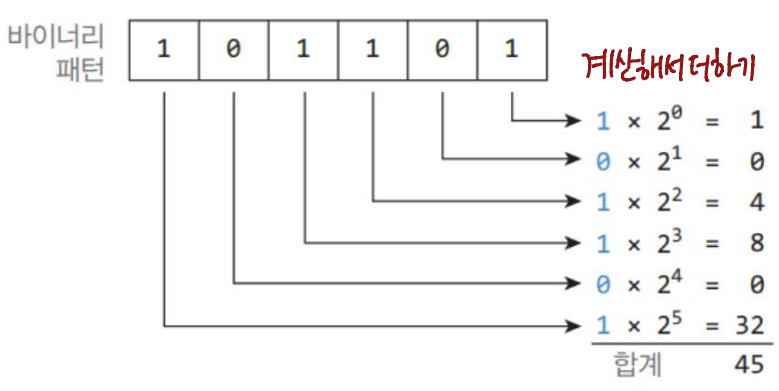
16진법, 8진법도 마찬가지

비트 패턴	10진법	16진법
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

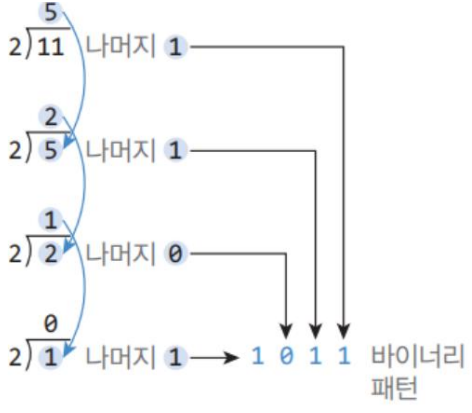
2진법, 10진법 및 16진법의 관계

8진법 → 0~7
16진법 → 0~9, A~F

- 2진법과 10진법 간의 변환



(a) 2진법에서 10진법으로



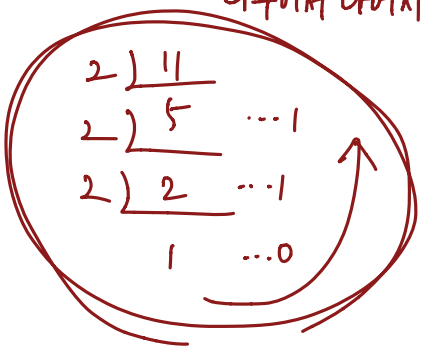
(b) 10진법에서 2진법으로

8, 16진수로 바꾸기

8) 11
1 ... 3

⇒ 11 = 13(8)
= B(16)

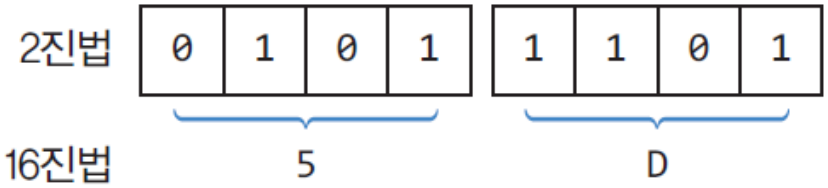
나누어서 나머지를 역순으로



4자리씩 끊어서 (오른쪽 기준) ex. 01/1101

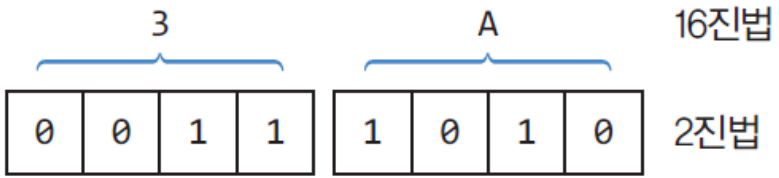
2진법 ↔ 8진법 : 4자리씩 끊어서

- 2진법과 16진법 간의 변환
  - $0101\,1101_2 = 5D_{16}$



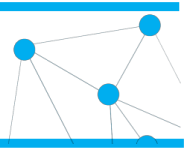
(a) 2진법에서 16진법으로

  - $3A_{16} = 00111010_2$



(b) 16진법에서 2진법으로

2.2 정보의 표현



- 숫자의 표현
- 문자의 표현
- 멀티미디어 정보의 표현

0, 양수, 음수

2.2.1 숫자의 표현

- 정수의 표현 : 부호-크기방식, 2의 보수 표현

1. 부호-크기 방식 : 부호비트를 따로둠

   - 제일 왼쪽 비트(MSB: Most Significant Bit): 정수의 부호
   - 나머지 비트: 정수의 절대값

0 → 양수
1 → 음수
+ 절댓값

비트 패턴	10진법 수
011	+3
010	+2
001	+1
000	+0
100	-0
101	-1
110	-2
111	-3

문제점.

(a) 3 비트를 이용한 부호–크기 방식

비트 패턴	10진법 수
0111	+7
0110	+6
0101	+5
0100	+4
0011	+3
0010	+2
0001	+1
0000	+0
1000	-0
1001	-1
1010	-2
1011	-3
1100	-4
1101	-5
1110	-6
1111	-7

(b) 4 비트를 이용한 부호–크기 방식

컴퓨터가 연산하기 더 쉬움

2. 2의 보수(2's Complement)

- 보수의 개념 : 어떤 수 a에 대한 n의 보수 x는 a + x = n가 되는 수
- 예) 1에 대한 10의 보수 : 1 + x = 10 ∴ x = 9
- 2의 보수 : 어떤 수 a와의 합이 $2(10_2)$가 되는 수
  = 1의 보수 + 1

2의 보수 구하는 방법 : 음수 구할 때

1. -2의 절대값 2를 2진법으로 표현 ⇒ 010
2. 010의 1의 보수를 구함 ⇒ 101 : 0과 1을 바꾸기
3. 1의 보수에 1을 더함 ⇒ 110
4. -2의 2의 보수 값은 110_2

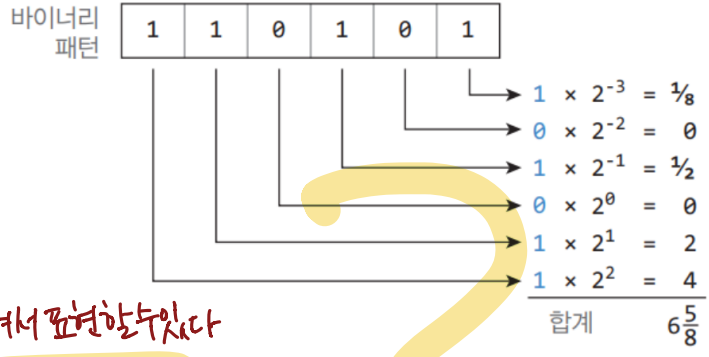
(-3) 0011 → 1100 → 1101

양수는 X

비트 패턴	10진법 수
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	-1
1110	-2
1101	-3
1100	-4
1011	-5
1010	-6
1001	-7
1000	-8

- 정수의 덧셈과 2의 보수 방식
  1. 부호-크기 방식
     - a + b의 경우: a, b의 부호와 절대값에 따라 복잡한 과정
  2. **2의 보수 방식**
     - a + b의 경우: 정수 a, b의 부호와 크기에 관계없음
     - 0의 표현이 한 가지
     - ➢ 훨씬 효율적인 계산 ⇒ 디지털 회로가 훨씬 단순해 짐

- 실수의 표현
  - 예) 110.101의 각 자리는 2^2, 2^1, 2^0, 2^{-1}, 2^{-2}, 2^{-3}을 의미
    - 아주 큰 수나 아주 작은 수를 표현하는 데 어려움



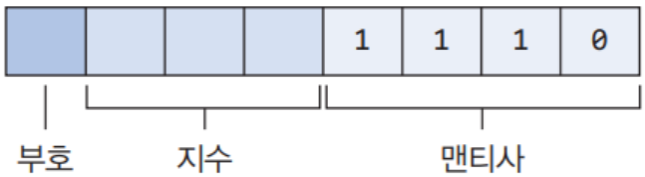
↳ 점을 옮겨서 표현할수있다

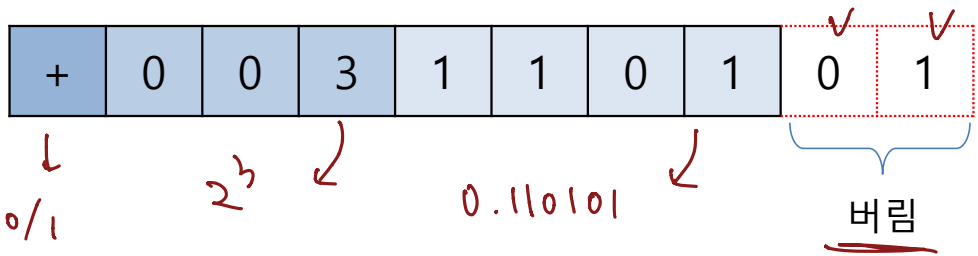
  - 부동소수점실수(Floating Point Number)
    - 예) 110.101의 표현

$$0.10110 \times 2^3,\ 1.0110 \times 2^2,\ 10.110 \times 2^1,$$
$$101.10 \times 2^0,\ 1011.0 \times 2^{-1}$$ 등

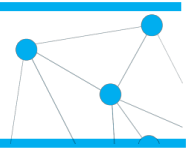
- 정규화 부동소수점실수
  - 110.101을 0.110101×2^3과 같이 소수점 다음에 항상 1이 나오도록 표현하는 방식

➢ 실수를 정규화 부동소수로 표현하여 부호(sign), 지수(exponent), 가수(mantissa)로 나누어 저장





2.2.2 문자의 표현



- ASCII 코드 아스키 (American Standard Code)
  - 미국 표준국(ANSI)에서 영어 문자를 위해 제정한 코드
    - 알파벳, 숫자, 문법 기호, 제어 부호로 원래 7 비트
    - ISO에서 1 비트 추가하여 8 비트 → 확장 (국제표준기구)
  - 예) “Happy!”

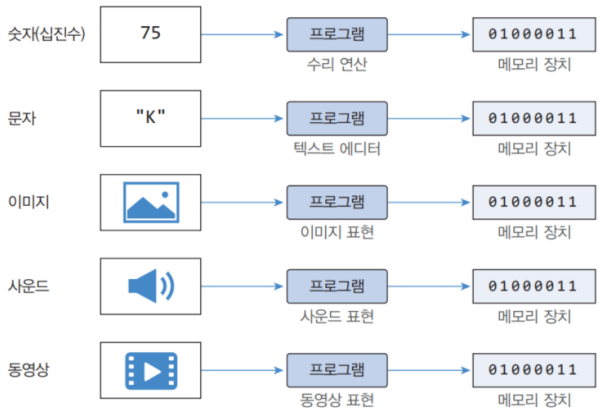
H	a	p	p	y	!
01001000	01100001	01110000	01110000	01111001	00100001

  - 한글: 초성, 중성, 종성
    - ASCII와 같이 8 비트로는 불가능 → 2바이트사용.
    - 16비트 조합형 및 완성형 코드

- Unicode
  - 영어를 제외한 언어에 새로운 코드
    - ASCII 코드 포함
    - 한글, 중국어, 일본어, 히브리어 등 다양한 언어를 위해 코드 영역 지정
    - 16 비트 또는 32 비트
  - 텍스트의 호환성 보장

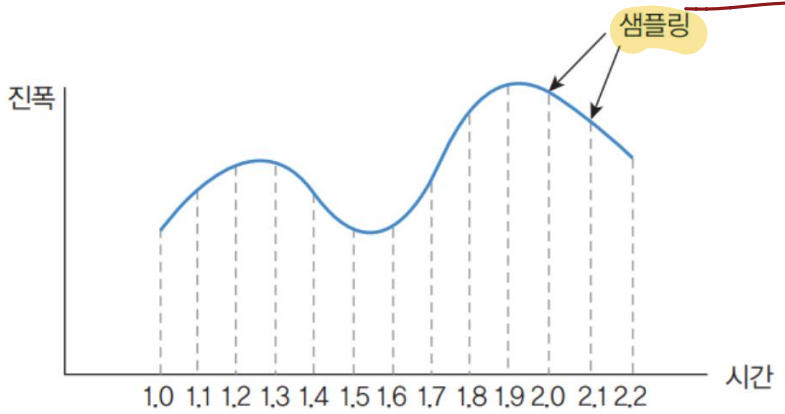
2.2.3 멀티미디어 정보의 표현

- 멀티미디어 정보의 특성
  - 멀티미디어 데이터: 숫자, 문자, 사운드, 이미지, 그래픽, 동영상 등, 동일한 바이너리 데이터를 다르게 해석 (프로그램에 따라)



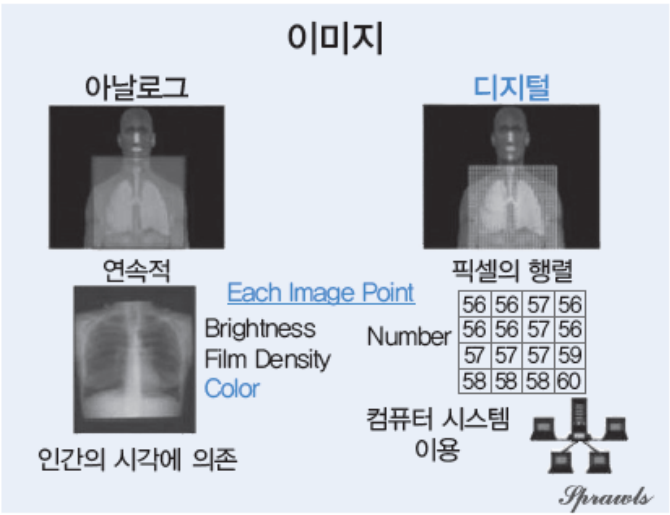
  - 특성: 1. 데이터 용량이 매우 크다 → 데이터의 압축 ↑
    2. 용도와 환경에 따라 다양한 형태로 표현
    (↳확장자)

- 사운드(Sound)의 표현
  - 사운드의 요소: 주파수,진폭, 음색
  - 가청 주파수: 20Hz~20KHz
  - 아날로그 파형인 사운드 데이터는 디지털화를 위해 샘플링 → 데이터를 일부추출
  - 나이키스트(Nyquist) 정리 : 최대 주파수의 2배로 샘플링 샘플링을 많이 할수록.
    - (20,000 x 2) = 40,000 sampling/sec → 1바이트?
    - 샘플링 값을 저장하기 위해 2 바이트 = 초당 80,000 바이트



- 이미지(Image)의 표현
  - 실 세계 장면은 아날로그 형태로 존재
  - 디지털 이미지는 픽셀(Pixel)로 구성
    - 수직, 수평 방향의 픽셀
    - 예) 수직, 수평 방향으로 각각 1000개의 픽셀

      1000 x 1000 = 1,000,000 픽셀



- RGB 컬러 모델
  - Red, Green, Blue 각 컬러가 1 바이트로 표현
  - 한 픽셀을 표현하기 위해서는 3바이트가 필요함
  - 예) 1,000,000 픽셀 x 3 바이트 = 3백만 바이트

- 대표적 이미지 압축: JPEG
  - .jpg