

# 1장. 디지털 컴퓨터와 정보

오늘의 교훈

극기하는자 실패도 약이요, 극기하지 못하는자 성공도 병이다

## 디지털 컴퓨터

---

- 범용성 (generality)
- program (특정 목적의 동작)
- 단지 2개의 신호 사용 → 높은 신뢰도 (잡음에 민감하지 않음)
- 2진 신호의 전압범위의 예 (그림 1-1)

전압 (입력)	전압 (출력)	해석	논리		2진 값	
3.0 ~ 5.5	4.0 ~ 5.5	High	True	False	1	0
-0.5 ~ 2.0	-0.5 ~ 1.0	Low	False	True	0	1

- active low

## 정보표현

---

- bit (binary digit) : 2진수 하나
- 이산정보  $\rightarrow n$  bit 로 표현
- analog 정보  $\rightarrow$  이산정보로 변환  $\rightarrow n$  bit 로 표현
- computer 는 이산정보만 처리함
- analog 정보를 처리할때는 AD 변환기 및 DA 변환기 필요

## 컴퓨터 구조

---

- 디지털 컴퓨터의 block diagram (\*그림 1-2)
- 메모리 : 입력, 출력, 중간 데이터 및 프로그램 저장
- 데이터 처리장치 : 프로그램에 의해 데이터 처리
- 제어 장치 (control unit) : 정보흐름 관할
- 중앙처리장치 (CPU (Central Processing Unit)) : 데이터 처리장치 + 제어장치
- 입력 장치 : 키보드
- 출력 장치 : 모니터

## 일반적인 컴퓨터 구성

---

- 표준 컴퓨터 (그림 p. 2)
- 프로세서 : CPU + FPU + MMU + Cache
- 버스 : 정보교류를 위한 공통의 회선 (프로세서 버스, 입출력 버스)
- 메모리
  - 전기성 메모리 : RAM, ROM
  - 자기성 메모리 : 하드디스크, 플로피 디스크

## 숫자 시스템

---

- 일반적인 수체계

$$a_n r^n + a_{n-1} r^{n-1} + \cdots + a_0 r^0 + a_{-1} r^{-1} + \cdots + a_{-m} r^{-m}$$

수체계	$a$	$r$
2 진수	0, 1	2
8 진수	0, 1, 2, ..., 7	8
10 진수	0, 1, 2, ..., 9	10
16 진수	0, 1, 2, ..., 9, A, B, C, D, E, F	16

- 수의 표시 :  $(a_n a_{n-1} \cdots a_0 . a_{-1} a_{-2} \cdots a_{-m})_r$

## 숫자 시스템 (Cont'd)

---

- 2진수 : 0, 1 로 수를 나타낸다 (2의 거듭제곱수 (\*표 1-1))
- 수의 변환 (다른 기수를 갖는 숫자 (\*표 1-2))
  - 10진수  $\rightarrow$  2진수 : 소숫점 앞은 2로 나누어서 소숫점 뒤는 2로 곱해서
  - ex)  $(37.6875)_{10} \rightarrow ?$

---

2   37	0.6875 * 2 = 1.3750 ... 1
2   18 ... 1	0.3750 * 2 = 0.7500 ... 0
2   9 ... 0	0.7500 * 2 = 1.5000 ... 1
2   4 ... 1	0.5000 * 2 = 1.0000 ... 1
2   2 ... 0	
2   1 ... 0	

---

result)  $(37.6875)_{10} \rightarrow (100101.1011)_2$

- 2, 8, 16 진수  $\rightarrow$  10진수 : 전개식 이용
- \* ex)  $(573.2)_8 = 5 \times 8^2 + 7 \times 8^1 + 3 \times 8^0 + 2 \times 8^{-1} = (379.25)_{10}$

## 숫자 시스템 (Cont'd)

---

- 2 진수  $\iff$  8 진수, 16진수
  - \*  $2^3 = 8$  : 그러므로 2진수의 세개의수는 8진수의 한개의수
  - \*  $2^4 = 16$  : 그러므로 2진수의 네개의수는 16진수의 한 개의수
  - \* ex)  $(10110001101011.111100000110)_2$ 
    - $\rightarrow (10\ 110\ 001\ 101\ 011.111\ 100\ 000\ 110)_2$
    - $\rightarrow (26153.7406)_8$
    - $\rightarrow (10\ 1100\ 0110\ 1011.1111\ 0000\ 0110)_2$
    - $\rightarrow (2C6B.F06)_{16}$



## 산술 연산

---

- 2진 가, 감산

00000	101100	00000	00110
01100	10110	10110	11110
+10001	+10111	-10010	-10011
-----	-----	-----	-----
11101	101101	00100	01011

- 승산

1011 (픽승수)
x 101 (승수)
-----
1011
0000
1011
-----
110111 (곱)

- 8진, 16진 연산

## 산술 연산 (Cont'd)

---

16진수 덧셈

```

  59F
+ E46
----
13E5

```

8진수 곱셈

```

  762
x  45
-----
 4672
 3710
-----
43772

```

- 보수 (complements)
  - 감산작용을 간단히 하고, 논리적 처리를 간단히함
  - 밑수  $r$  의 보수 :  $r$  의보수와  $r - 1$  의 보수가 있음

## 산술 연산 (Cont'd)

---

### \* $r$ 의 보수

$n$  자릿수 이고 밑수가  $r$ 인 양수  $N$  에서

$$N \neq 0 \text{ 일때 } r^n - N$$

$$N = 0 \text{ 일때 } 0$$

$$\text{ex1) } (25.639)_{10} \Rightarrow 10^2 - 25.639 = 74.361$$

$$\text{ex2) } (101100)_2 \Rightarrow (2^6)_2 - (101100)_2 = 010100$$

### \* $r - 1$ 의 보수

정수부분이  $n$  자릿수 이고 소수부분이  $m$  자릿수 양수  $N$  에서

$$r^n - r^{-m} - N$$

$$\text{ex1) } (25.639)_{10} \Rightarrow 10^2 - 10^{-3} - 25.639 = 99.999 - 25.639 = 74.360$$

$$\text{ex2) } (101100)_2 \Rightarrow (2^6)_2 - 1 - (101100)_2 = 010011$$

### \* 결론

$$\underline{r - 1 \text{ 의 보수 : } r - 1 - \text{각자릿수}}$$

$$\underline{r \text{ 의 보수 : } r - 1 \text{ 의 보수} + r^{-m}}$$

## 산술 연산 (Cont'd)

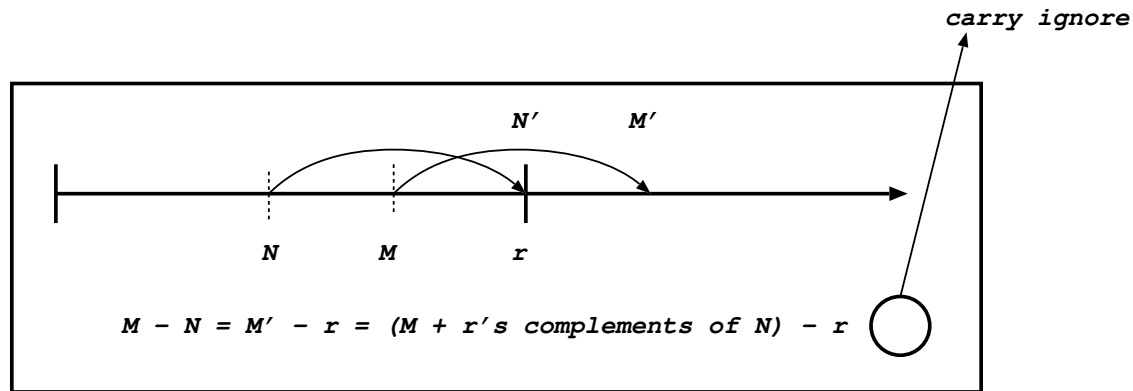
---

- $r$  의 보수에 의한 감산
  - \* 밑수가  $r$  인 양수  $M - N$ :  $M + N$  의  $r$  의 보수 후
    - carry 가 있으면 무시
    - carry 가 없으면 다시  $r$  의 보수 후 -
    - ex1)  $2397 - 1785 = 2397 + 8215 = (1) 0612 \rightarrow 0612$
    - ex2)  $3846 - 5213 = 3846 + 4787 = (0) 8633 \rightarrow -1367$
- $r - 1$  의 보수에 의한 감산
  - \* 밑수가  $r$  인 양수  $M - N$ :  $M + N$  의  $r - 1$  의 보수 후
    - carry 가 있으면 최하위 숫자에 1을 더함
    - carry 가 없으면 다시  $r - 1$  의 보수 후 -
    - ex1)  $2397 - 1785 = 2397 + 8214 = (1) 0611 \rightarrow 0612$
    - ex2)  $3846 - 5213 = 3846 + 4786 = (0) 8632 \rightarrow -1367$

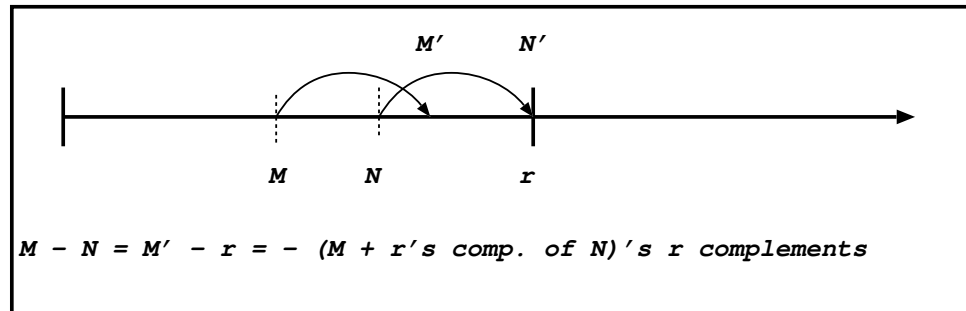
## 산술 연산 (Cont'd)

### - 보수이용에대한 개념

Subtract by  $r$ 's complements



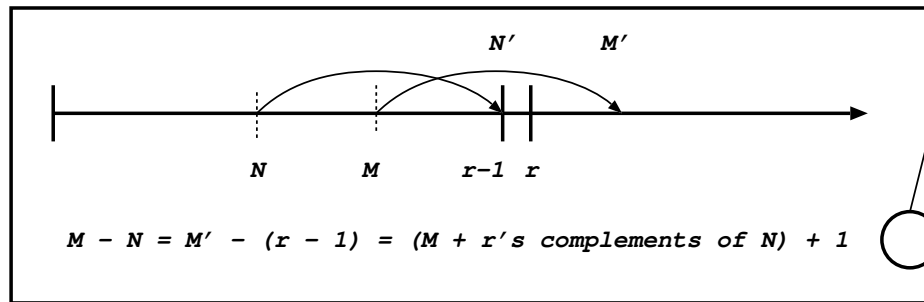
(a)  $M > N$  case



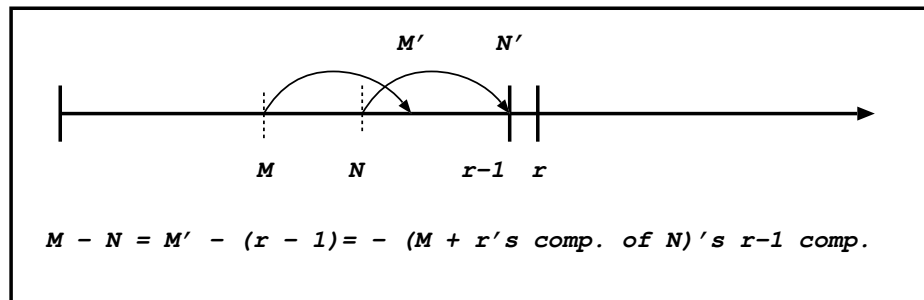
(b)  $M < N$  case

# 산술 연산 (Cont'd)

Subtract by  $r-1$ 's complements



(a)  $M > N$  case



(b)  $M < N$  case

## 10진 코드

---

- 사람이 자주사용하는 10진수를 컴퓨터가 사용하는 2진수로 나타냄
- $2^n \geq 10$  이어야함으로 4 bit 로 10진수 표현 (나머지 6개는 사용안됨)
- 일반적인 코드 (BCD (Binary Coded Decimal) 코드 표 1-3)
- BCD 가산

	1	1	
448	0100	0100	1000
+489	+0100	+1000	+1001
---	----	----	----
937	1001	1101	10001
		+0110	+0110
		----	-----
		10011	10111
	1001	0011	0111

## 영문 숫자 코드

---

- 영문 알파벳 문자에 대한 2진 코드 → 7bit 로 표현
- ASCII (American Standard Code for Information Interchange) 문자코드 (\*표 1-4)
- 일반적인 에러 검출 방법
  - 전송 data : data + parity bit
    1. odd parity : 전송 code 의 1의 bit 수가 홀수가 되게 parity bit 를 첨가함
    2. even parity : 전송 code 의 1의 bit 수가 짝수가 되게 parity bit 를 첨가함



## 영문 숫자 코드 (Cont'd)

---

–  $\oplus$  : exclusive OR, ring sum

\* 2진수에서는 1의 갯수가 홀수일 때 : 1, 짝수일 때 : 0

2진수	8진수
$0 \oplus 0 = 0$	$3 \oplus 6 = 1$
$0 \oplus 1 = 1$	$1 \oplus 3 \oplus 7 = 3$
$1 \oplus 0 = 1$	$5 \oplus 5 = 2$
$1 \oplus 1 = 0$	
$1 \oplus 0 \oplus 1 = 0$	

\* 짝수 parity bit 만드는 방법:

$$p = bit_n \oplus bit_{n-1} \oplus \cdots \oplus bit_0$$

## 영문 숫자 코드 (Cont'd)

---

- reflected code (gray code) : 연속되는 code 사이에는 오로지 1 bit 만 틀린 code
  - \* analog signal → digital data
  - \* BCD → Gray
$$G_n = B_n,$$
$$G_i = B_{i+1} \oplus B_i$$
  - \* Gray → BCD
$$B_n = G_n,$$
$$B_i = G_n \oplus G_{n-1} \cdots \oplus G_i$$
- 유니코드
  - \* 16bit 영문숫자 코드에 대한 새로운 표준 → 65,536 개의 문자 지정가능
  - \* 전세계 언어와 기호 문자 표기가능 (\*표 1-5)

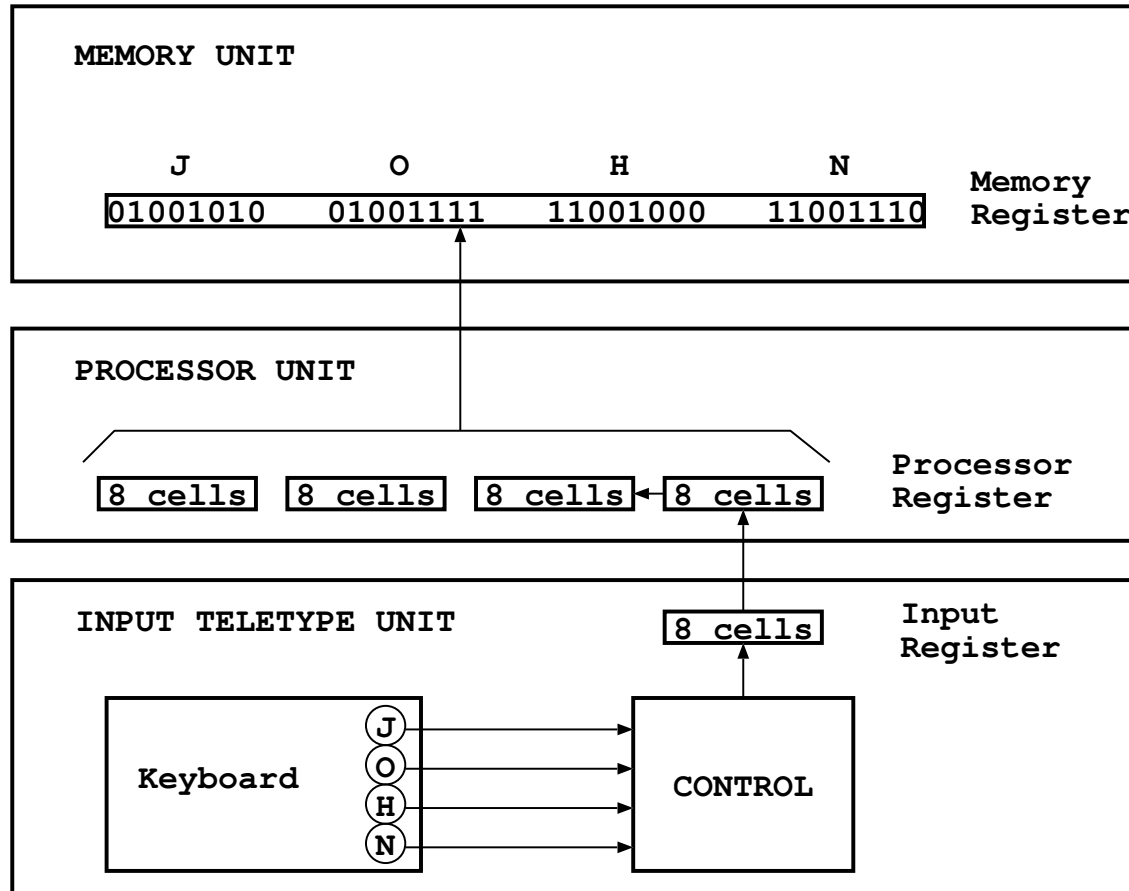
## binary storage and registers

---

- 2진 데이터를 저장하기 위한 장치
- 데이터 저장장치 : 1bit (Logic 0, 1)을 단위로 저장
- Flip Flop → Register → Memory (ROM, RAM, FD, HD, ...)

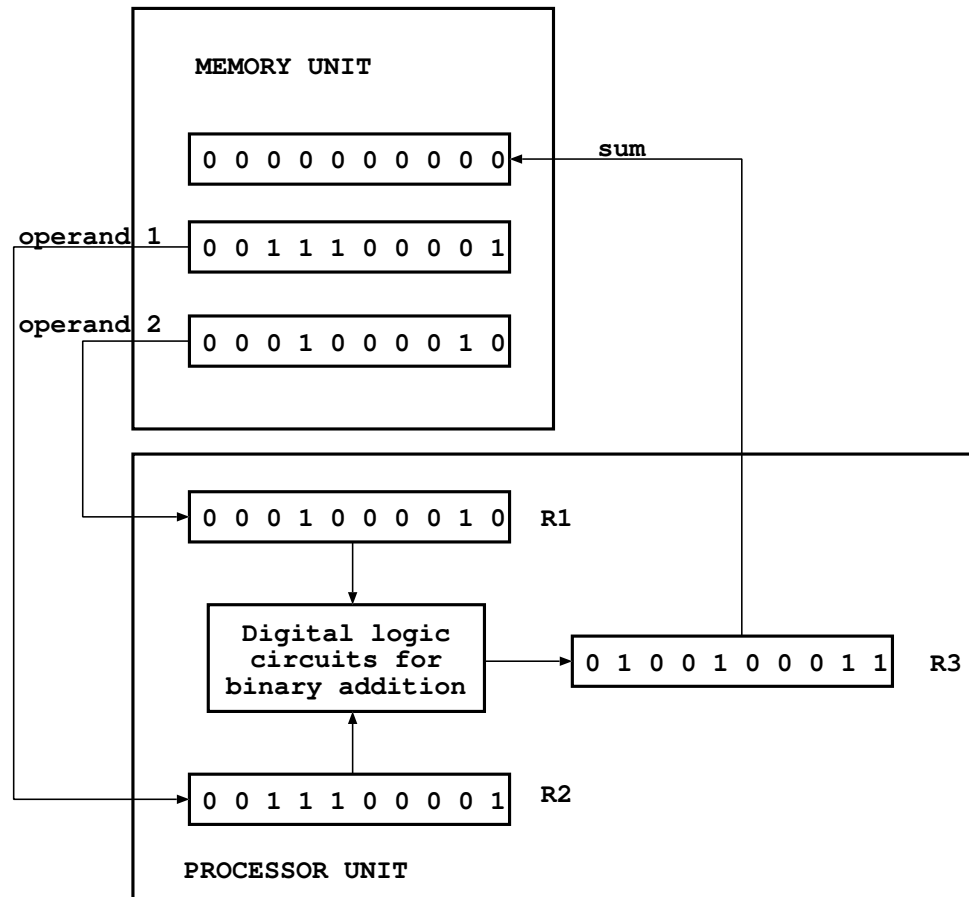
## binary storage and registers (Cont'd)

- 데이터의 처리 (그림 1-2)



## binary storage and registers (Cont'd)

- 정보처리의 예 (그림 1-3)



## 2진 논리 (binary logic)

---

- 부울대수 (boolean algebra)
  - AND :  $z = x \cdot y$ 
    - \* 기호 :  $\cdot, \wedge, \cap$
    - \*  $z = 1$ ,  $x$  와  $y$  가 모두 1일때만, 다른 경우는 0
    - \* multiple valued logic에서는 최소치 (minimum)
    - \* 스위칭 회로에서 직렬 연결 (그림 1-4 (a))
    - \* 진리표

AND		
x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

## 2진 논리 (binary logic) (Cont'd)

---

- OR :  $z = x + y$ 
  - \* 기호 : +, V, U
  - \*  $z = 0$ ,  $x$  와  $y$  가 모두 0일때만, 다른 경우는 1
  - \* multiple valued logic에서는 최대치 (maximum)
  - \* 스위칭 회로에서 병렬 연결 (그림 1-4 (b))
  - \* 진리표

OR		
x	y	x + y
0	0	0
0	1	1
1	0	1
1	1	1

## 2진 논리 (binary logic) (Cont'd)

---

- NOT :  $z = x'$  or  $\bar{x}$ 
  - \* 기호 :  $\sim, -, '$
  - \*  $z = 0, x$  가 1일때,  $z = 1, x$  가 0일때
  - \* 진리표

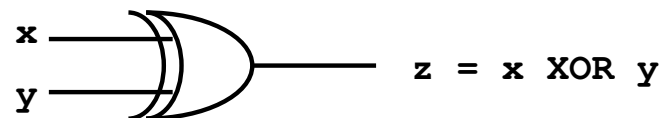
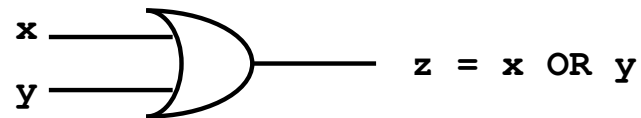
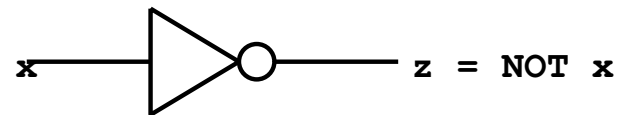
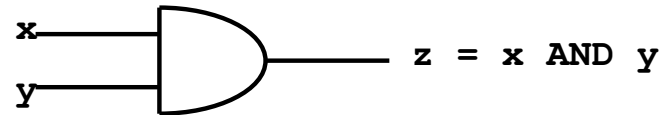
NOT	
x	x'
0	1
1	0



## 2진 논리 (binary logic) (Cont'd)

---

- 논리 게이트 (logic gate): 전자회로적으로 binary logic 을 구현한 것 (logic circuit 라고도함)
  - AND, OR, NOT, XOR gate (그림 1-6)



## 2진 논리 (binary logic) (Cont'd)

---

- IC (Integrated Circuit) : 논리 게이트 집적하여 하나의 칩에 구현
  - SSI (Small Scale IC) : 대여섯개의 논리게이트
  - MSI (Medium Scale IC) : 10 ~ 100 개의 논리게이트
  - LSI (Large Scale IC) : 100 개이상의 논리게이트
  - VLSI (Very Large Scale IC) : 수천개 이상의 논리게이트