chapter 01.

2 진수계

01. 10진수

❑ 10진수 표현법

- ❖ 기수가 10인 수
- ❖ 0,1,2,3,4,5,6,7,8,9 사용

$$9345.35 = 9\times1000 + 3\times100 + 4\times10 + 5\times1 + 3\times0.1 + 5\times0.01$$
$$= 9\times10^{3} + 3\times10^{2} + 4\times10^{1} + 5\times10^{0} + 3\times10^{-1} + 5\times10^{-2}$$

- ❖ 바빌로니아인 : 60진법을 사용(기원전 4000~3000년)
- ❖ 고대 로마의 기수법에는 5진법을 사용
- ❖ 10진법의 아라비아 숫자는 인도에서 기원전 2세기에 발명

진법을 나타내는 기본수를 기수(基數, radix)라 한다. 10이 기수인 수를 10진법, 2가 기수인 수를 2진법, 12가 기수인 수를 12진법이라 한다.

02. 2진수

❑ 2진수 표현법

- ❖ 기수가 2인 수
- ❖ 0, 1 사용

$$
\begin{aligned}
1010.1011_{(2)} &= 1\times1000_{(2)} + 0\times100_{(2)} + 1\times10_{(2)} + 0\times1_{(2)} \\
&\quad + 1\times0.1_{(2)} + 0\times0.01_{(2)} + 1\times0.001_{(2)} + 1\times0.0001_{(2)} \\
&= 1\times2^{3} + 0\times2^{2} + 1\times2^{1} + 0\times2^{0} + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3} + 1\times2^{-4}
\end{aligned}
$$

03. 8진수와 16진수

❑ 8진수 표현법

❖ 0에서 7까지 8개의 수로 표현

$$607.36_{(8)} = 6\times100_{(8)} + 0\times10_{(8)} + 7\times1_{(8)} + 3\times0.1_{(8)} + 6\times0.01_{(8)}$$

$$= 6\times8^{2} + 0\times8^{1} + 7\times8^{0} + 3\times8^{-1} + 6\times8^{-2}$$

❑ 16진수 표현법

❖ 0에서 9, A(a)에서 F(f)까지 16개의 기호로 표현

$$6C7.3A_{(16)} = 6\times100_{(16)} + C\times10_{(16)} + 7\times1_{(16)} + 3\times0.1_{(16)} + A\times0.01_{(16)}$$

$$= 6\times16^{2} + C\times16^{1} + 7\times16^{0} + 3\times16^{-1} + A\times16^{-2}$$

❑ 10진수에 해당하는 16진 기호

10진수	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16진수	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**

1. 10진수-2진수 변환

- ❖ 정수부분과 소수부분으로 나누어 변환
- ❖ 정수부분은 2로 나고, 소수부분은 2를 곱한다.
- ❖ 10진수 69.6875를 2진수로 변환하는 경우

		나머지	2진수
2	69		
2	34	… 1	1
2	17	… 0	01
2	8	… 1	101
2	4	… 0	0101
2	2	… 0	00101
2	1	… 0	000101
	0 (몫)	1	1000101

2진수	정수	소수
	0.	6875
	X	2
0.1	1.	3750
	X	2
0.10	0.	7500
	X	2
0.101	1.	5000
	X	2
0.1011	1.	0

곱셈결과 정수를 적는다.

소수 부분이 0이 될때까지 계산한다.

$$69.6875_{(10)} = 1000101.1011_{(2)}$$

❖ 10진수 69.6을 2진수로 변환하는 경우

❖ 10진수 소수부분은 대부분의 경우 정확한 2진수로 변환이 안 된다.

		나머지	2진수
2	69		
2	34	… 1	1
2	17	… 0	01
2	8	… 1	101
2	4	… 0	0101
2	2	… 0	00101
2	1	… 0	000101
	0 몫	1	1000101

정수	소수	2진수
0.	6	
X	2	
1.	2	0.1
X	2	
0.	4	0.10
X	2	
0.	8	0.100
X	2	
1.	6	0.1001
X	2	
1.	2	0.10011

$$69.6_{(10)} = 1000101.1001100110011001\ldots\ldots_{(2)}$$

2. 10진수-8진수 변환

❖ 10진수 69.6875를 8진수로 변환하는 경우

❖ 8로 나누고, 곱한다.

			나머지	8진수
8	69			
8	8	…	5	5
8	1	…	0	05
	0	…	1	105
	몫			

8진수	정수	소수	
	0.	6875	
	X	8	곱셈결과 정수를 적는다.
0.5	5.	5000	
	X	8	소수 부분이 0이 될때까지 계산한다.
0.54	4.	0	

$$69.6875_{(10)} = 105.54_{(8)}$$

❖ 10진수 69.6을 8진수로 변환하는 경우

$$69.6 = 105.46314631\ldots_{(8)}$$

3. 10진수-16진수 변환

❖ 10진수 69.6875를 16진수로 변환하는 경우

16 ⌊ 69 나머지 16진수 16 ⌊ 4 … 5 ↑ 5 0 … 4 45 몫	16진수 정수 소수 0. 6875 X 16 ↓ 0.B 11. 0000 곱셈결과 정수를 적는다. 소수 부분이 0이 될때까지 계산한다.

$$69.6875_{(10)} = 45.B_{(16)}$$

❖ 10진수 69.6을 16진수로 변환하는 경우

$$69.6_{(10)} = 45.999\ldots_{(16)}$$

다른 진법의 경우도 같은 방법을 이용하여 변환할 수 있다

- Ex 1-2) 10진수 153을 8진수로 변환하라.

153	
19	1
2	3
0	2

$= (231)_8$

- Ex 1-3) 10진수 0.6875를 2진수로 변환하라.

	정수		소수	계수
0.6875*2 =	1	+	0.3750	$a_{-1} = 1$
0.3750*2 =	0	+	0.7500	$a_{-2} = 0$
0.7500*2 =	1	+	0.5000	$a_{-3} = 1$
0.5000*2 =	1	+	0.0000	$a_{-4} = 1$

답:$(0.6875)10 = (0.a_{-1}a_{-2}a_{-3}a_{-4})_2 = (0.1011)_2$

4. 2진수-8진수-16진수-10진수 상호변환

10진수	2진수	8진수	16진수
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

❑상호변환 예

$$69.6875 = 1000101.1011_{(2)}$$
$$= 001\ \ 000\ \ 101.101\ \ 100_{(2)}$$
$$= \ \ 1 \ \ \ \ 0 \ \ \ \ 5.\ \ 5 \ \ \ \ 4_{(8)}$$

10진->2진->8진
3자리씩 나눔

$$69.6 = 1000101.10011001100110011\ldots_{(2)}$$
$$= 001\ \ 000\ \ 101.100\ \ 110\ \ 011\ \ 001\ \ 100\ \ 110\ \ 011\ldots_{(2)}$$
$$= \ \ 1 \ \ \ \ 0 \ \ \ \ 5.\ \ 4 \ \ \ \ 6 \ \ \ \ 3 \ \ \ \ 1 \ \ \ \ 4 \ \ \ \ 6 \ \ \ \ 3\ldots_{(8)}$$

$$69.6875 = 1000101.1011_{(2)}$$
$$= 0100\ \ 0101.1011_{(2)}$$
$$= \ \ 4 \ \ \ \ 5.\ \ \ \ B_{(16)}$$

10진->2진->16진
4자리씩 나눔

$$69.6 = 1000101.10011001100110011001\ldots_{(2)}$$
$$= 0100\ \ 0101.1001\ \ 1001\ \ 1001\ \ 1001\ \ 1001\ldots_{(2)}$$
$$= \ \ 4 \ \ \ \ 5.\ \ 9 \ \ \ \ 9 \ \ \ \ 9 \ \ \ \ 9 \ \ \ \ 9\ldots_{(16)}$$

❑ 상호변환 예(Cont'd)

$$367.75_{(8)} = 011\ \ 110\ \ 111.111\ \ 101_{(2)}$$

8진수 1자리
=2진수 3자리

$$9A3.50F3_{(16)} = 1001\ \ 1010\ \ 0011.0101\ \ 0000\ \ 1111\ \ 0011_{(2)}$$

16진수 1자리
=2진수 4자리

$$101101.101_{(2)} = 1\times 2^5 + 0\times 2^4 + 1\times 2^3 + 1\times 2^2 + 0\times 2^1 + 1\times 2^0 + 1\times 2^{-1} + 0\times 2^{-2} + 1\times 2^{-3}$$
$$= 32+0+8+4+0+1+0.5+0+0.125 = 45.625_{(10)}$$

$$364.35_{(8)} = 3\times 8^2 + 6\times 8^1 + 4\times 8^0 + 3\times 8^{-1} + 5\times 8^{-2}$$
$$= 3\times 64 + 6\times 8 + 4\times 1 + 3\times 0.125 + 5\times 0.015325$$
$$= 192+48+4+0.375+0.078125$$
$$= 244.453125_{(10)}$$

$$A3.D2_{(16)} = 10\times 16^1 + 3\times 16^0 + 13\times 16^{-1} + 2\times 16^{-2}$$
$$= 160+3+0.8125+0.0078125$$
$$= 163.8203125_{(10)}$$

❑ 상호변환 예(Cont'd)

$$364.35_{(8)} = 011\ 110\ 100.011\ 101_{(2)}$$
$$= 0\times2^8 + 1\times2^7 + 1\times2^6 + 1\times2^5 + 1\times2^4 + 0\times2^3 + 1\times2^2 + 0\times2^1 + 0\times2^0$$
$$+ 0\times2^{-1} + 1\times2^{-2} + 1\times2^{-3} + 1\times2^{-4} + 0\times2^{-5} + 1\times2^{6}$$
$$= 0+128+64+32+16+0+4+0+0+0+0.25+0.125+0.0625+0+0.15625$$
$$= 244.453125_{(10)}$$

8진->2진->10진

$$A3.D2_{(16)} = 1010\ 0011.1101\ 0010_{(2)}$$
$$= 1\times2^7 + 0\times2^6 + 1\times2^5 + 0\times2^4 + 0\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0$$
$$+ 1\times2^{-1} + 1\times2^{-2} + 0\times2^{-3} + 1\times2^{-4} + 0\times2^{-5} + 0\times2^{-6} + 1\times2^{-7} + 0\times2^{-8}$$
$$= 128+0+32+0+0+0+2+1+0.5+0.25+0+0.0625+0+0+0.0078125+0$$
$$= 163.8203125_{(10)}$$

16진->2진->10진

1. 2진수 양의 정수 덧셈

0+0=0, 0+1=1, 1+0=1, 1+1=10 (자리올림 발생)

10진수

```
Carry →  11
         49
+        58
---------------
        107
```

2진수

```
Carry →  0110000
         00110001
+        00111010
---------------
         01101011
```

8진수

```
Carry →  10
         61
+        72
---------------
        153
```

16진수

```
Carry →   0
         31
+        3A
---------------
         6B
```

2. 2진 음수의 표현과 보수(complement)

❖ 최상위비트(MSB)를 부호비트로 사용

양수(+) : 0 음수(-) : 1

❖ 2진 음수를 표시하는 방법

- 부호와 절대치(sign- magnitude)
- 1의 보수(1's complement)
- 2의 보수(2's complement)

567의 9의 보수

$$10^3 - 1 - 567 = 999 - 567 = 432$$

➢ r진법 n자릿수 x의 r의 보수 : $r^n - x$

➢ r진법 n자릿수 x의 r−1의 보수 : $r^n - 1 - x$

➢00000011의 1의 보수 = $2^8 - 1 - 00000011 = 11111111 - 00000011 = 11111100$

➢00000011의 2의 보수 = $2^8 - 00000011 = 100000000 - 00000011 = 11111101$

❖ 양수를 보수로 바꾸면 음수

❖ 음수를 보수로 바꾸면 양수

❖ 1의 보수로 변환하는 방법

- 0 → 1, 1 → 0으로 변환

  00000011 → 1의 보수 = 11111100

❖ 2의 보수로 변환하는 방법

1. 1의 보수 + 1 = 2의 보수

   00000011 → 2의 보수 = 1의 보수 + 1 = 11111100 + 1 = 11111101

2. 맨 뒤에서부터 최초의 1이 나타날 때까지(아래 빨간 부분)는 그대로 쓰고 나머지 앞부분은 0 → 1, 1→ 0으로 바꾼다.

   01101100 → 2의 보수 = 10010100

   = 1의 보수 + 1 = 10010011 + 1 = 10010100

- 예) 8비트의 2진수로 표현된 9.

  +9 : 00001001

  -9 : 10001001 (부호 크기 방식)

  11110110 (부호화된 1의 보수)

  11110111 (부호화된 2의 보수)

표 1-3 부호화된 2진수

10진수	부호화된 2의 보수	부호화된 1의 보수	부호-크기 방식
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000
−0	—	1111	1000
−1	1111	1110	1001
−2	1110	1101	1010
−3	1101	1100	1011
−4	1100	1011	1100
−5	1011	1010	1101
−6	1010	1001	1110
−7	1001	1000	1111
−8	1000	—	—

❖ 뺄셈 : 보수를 취하여 더하면 뺄셈을 수행(Carry가 있으면, 버림)

$$
\begin{aligned}
7928 - 879 &= 7928 + (-879) \\
&= 7928 + (-0879) \\
&\Rightarrow 7928 + (10^4 - 0879) \\
&= 7928 + 9121 \\
&= 17049 \\
&\Rightarrow 7049
\end{aligned}
$$

❖ 2의 보수를 사용한 2진 정수의 표현 범위

bit 수	2의 보수를 사용한 2진 정수의 표현 범위
n bit	$-2^{n-1} \sim +2^{n-1}-1$
4 bit	$-2^{4-1} \sim +2^{4-1}-1$ $(-8 \sim +7)$
8 bit	$-2^{8-1} \sim +2^{8-1}-1$ $(-128 \sim +127)$
16 bit	$-2^{16-1} \sim +2^{16-1}-1$ $(-32768 \sim +32767)$
32 bit	$-2^{32-1} \sim +2^{32-1}-1$ $(-2147483648 \sim +2147483647)$

3. 2의 보수로 표현된 음수를 10진수로 변환

(2의 보수 10101100을 10진수로 변환하는 경우)

❑ 첫 번째 방법.

MSB가 1이므로 음수이다. 실제크기는 −128이다.

$$10101100_{(2)} = -1\times 2^7 + 0\times 2^6 + 1\times 2^5 + 0\times 2^4 + 1\times 2^3 + 1\times 2^2 + 0\times 2^1 + 0\times 2^0$$
$$= -128 + 0 + 32 + 0 + 8 + 4 + 0 + 0 = -128 + 44$$
$$= -84$$

❑ 두 번째 방법.

2의 보수로 바꾸어 10진수로 바꾼 다음 −부호를 붙인다.

$$10101100_{(2)} \Rightarrow 2\text{의 보수 } 01010100_{(2)}$$
$$= 0\times 2^7 + 1\times 2^6 + 0\times 2^5 + 1\times 2^4 + 0\times 2^3 + 1\times 2^2 + 0\times 2^1 + 0\times 2^0$$
$$= 0 + 64 + 0 + 16 + 0 + 4 + 0 + 0$$
$$= 84$$
$$\Rightarrow -\text{부호를 붙이면} = -84$$

4. 2의 보수 연산 (8bit)

양수+양수 = 양수

(49+58=107)

Carry → 0110000

00110001

\+ 00111010

0 01101011

큰 수−작은 수=양수

(58−49=9)

Carry → 1111110

00111010

− 00110001

00111010

\+ 11001111

1 00001001

작은 수 − 큰 수= 음수

(49−58=−9)

Carry → 0000000

00110001

− 00111010

00110001

\+ 11000110

0 11110111

음수 + 음수= 음수

(−49−58=−107)

Carry → 1001110

− 00110001

− 00111010

11001111

\+ 11000110

1 10010101

큰 양수 + 큰 양수 = 음수

(98+74=−84)

Carry → 1000010

01100010

\+ 01001010

0 10101100

큰 음수 + 큰 음수 = 양수

(−98−74=+84)

Carry → 0111110

− 01100010

− 01001010

10011110

\+ 10110110

1 01010100

overflow

06. 2진 부동소수점의 표현

❖ 컴퓨터의 부동소수점수는 IEEE 754표준을 따른다.

❖ 부호(sign), 지수(exponent), 가수(mantissa)의 세 영역으로 표시.

❖ 단정도(single precision) 부동소수점수와 배정도(double precision) 부동소수점수의 두 가지 표현 방법이 있다.

❑ 단정도 및 배정도 부동소수점수의 비트 할당

구분	IEEE 754 표준 부동소수점수의 비트 할당	바이어스
단정도 부동소수점수	8 bit, 23 bit 31 | 30 29 … 24 23 | 22 21 … 1 0 S | Exponent | Mantissa	127
배정도 부동소수점수	11 bit, 52 bit 63 | 62 61 … 53 52 | 51 50 … 1 0 S | Exponent | Mantissa	1023

07. BCD코드와 3 초과코드

❑ BCD 코드

❖ BCD코드(Binary Coded Decimal Code : 2진화 10진 코드, 8421 코드)는 10진수 0부터 9까지를 2진화한 코드로서 실제 표기는 2진수로 하지만 10진수처럼 사용한다. 즉, 1010부터 1111까지의 6개는 사용되지 않는다.

10진수	BCD 코드	10진수	BCD 코드	10진수	BCD 코드
0	0000	10	0001 0000	20	0010 0000
1	0001	11	0001 0001	31	0011 0001
2	0010	12	0001 0010	42	0100 0010
3	0011	13	0001 0011	53	0101 0011
4	0100	14	0001 0100	64	0110 0100
5	0101	15	0001 0101	75	0111 0101
6	0110	16	0001 0110	86	1000 0110
7	0111	17	0001 0111	97	1001 0111
8	1000	18	0001 1000	196	0001 1001 0110
9	1001	19	0001 1001	237	0010 0011 0111

❑ BCD코드의 연산

❖ 10진 덧셈 (6+3=9) (42+27=69)

```
     0110              0100 0010
+    0011          +   0010 0111
---------          -------------
     1001              0110 1001
```

❖ 계산 결과가 BCD코드를 벗어나는 즉, 9를 초과하는 경우에는 계산 결과에 6(0110)을 더해준다.

(8+7=15)

```
         1000
    +    0111
-------------
         1111
    +    0110
-------------
    0001 0101
```

❑ 3초과 코드

- ❖ BCD코드(8421코드)로 표현된 값에 3을 더해 준 값으로 나타내는 코드
- ❖ 자기 보수의 성질이 있음.

10진수	BCD 코드	3-초과 코드
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

보수 관계

08. 다양한 2진 코드들

❑ 가중치코드(weighted code)

❖ 그 위치에 따라 정해진 값을 갖는 코드

10진수	8421코드 (BCD)	2421 코드	5421 코드	84-2-1 코드	51111 코드	바이퀴너리코드 (Biquinary Code) 5043210	링 카운터 (ring counter) 9876543210
0	0000	0000	0000	0000	00000	0100001	0000000001
1	0001	0001	0001	0111	00001	0100010	0000000010
2	0010	0010	0010	0110	00011	0100100	0000000100
3	0011	0011	0011	0101	00111	0101000	0000001000
4	0100	0100	0100	0100	01111	0110000	0000010000
5	0101	1011	1000	1011	10000	1000001	0000100000
6	0110	1100	1001	1010	11000	1000010	0001000000
7	0111	1101	1010	1001	11100	1000100	0010000000
8	1000	1110	1011	1000	11110	1001000	0100000000
9	1001	1111	1100	1111	11111	1010000	1000000000

❑ 비가중치코드(non-weighted code)

- ❖ 각각의 위치에 해당하는 값이 없는 코드
- ❖ 데이터 변환과 같은 특수한 용도로 사용되기 위한 코드 (2-out-of-5)

10진수	3-초과 코드	5중 2코드 (2-out-of-5)	shift counter	그레이코드
0	0011	11000	00000	0000
1	0100	00011	00001	0001
2	0101	00101	00011	0011
3	0110	00110	00111	0010
4	0111	01001	01111	0110
5	1000	01010	11111	0111
6	1001	01100	11110	0101
7	1010	10001	11100	0100
8	1011	10010	11000	1100
9	1100	10100	10000	1101

09. 그레이 코드

❑ 그레이 코드(Gray Code)

❖ 가중치가 없는 코드이기 때문에 연산에는 부적당하지만, 아날로그-디지털 변환기나 입출력 장치 코드로 주로 쓰인다.

❖ 연속되는 코드들 간에 하나의 비트만 변화하여 새로운 코드가 된다.

10진수	2진 코드	Gray 코드	10진수	2진 코드	Gray 코드
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

이웃하는 코드간에 한 비트만 다르다.

10. 영숫자 코드

1. ASCII(American Standard Code for Information Interchange) 코드

- ❖ 미국 국립 표준 연구소(ANSI)가 제정한 정보 교환용 미국 표준 코드
- ❖ 128가지의 문자를 표현할 수 있다.

패리티	존(zone)			디지트(digit)			
b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1

❑ ASCII 코드의 구성

parity	Zone bit			Digit Bit			
7	6	5	4	3	2	1	0
C	1	0	0	영문자 A~O(0001~1111)			
	1	0	1	영문자 P~Z(0000~1010)			
	0	1	1	숫자 0~9(0000~1001)			

❑ 표준 ASCII 코드 표

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	–	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	L	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	₩	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	I	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	|	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	C	}	~	

표 1-7 ASCII 표

$b_4b_3b_2b_1$ \ $b_7b_6b_5$	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	“	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	‘	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	−	=	M	]	m	}
1110	SO	RS	.	>	N	∧	n	~
1111	SI	US	/	?	O	—	o	DEL

2. 표준 BCD 코드

❖ 6비트로 하나의 문자를 표현

❖ 최대 64문자까지 표현 가능한 코드

<table>
<tr><th>parity</th><th colspan="2">Zone bit</th><th colspan="4">Digit Bit</th></tr>
<tr><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td rowspan="5">C</td><td>1</td><td>1</td><td colspan="4">영문자 A~I(0001~1001)</td></tr>
<tr><td>1</td><td>0</td><td colspan="4">영문자 J~R(0001~1001)</td></tr>
<tr><td>0</td><td>1</td><td colspan="4">영문자 S~Z(0010~1001)</td></tr>
<tr><td>0</td><td>0</td><td colspan="4">숫자 0~9(0001~1010)</td></tr>
<tr><td colspan="2">혼용</td><td colspan="4">특수문자 및 기타문자</td></tr>
</table>

❑ 표준 BCD 코드 표

문자	C ZZ8421	문자	C ZZ8421	문자	C ZZ8421	문자	C ZZ8421	문자	C ZZ8421
A	0 110001	J	1 100001	S	1 010010	1	0 000001	=	0 001011
B	0 110010	K	1 100010	T	0 010011	2	0 000010	>	1 001100
C	1 110011	L	0 100011	U	1 010100	3	1 000011	+	0 010000
D	0 110100	M	1 100100	V	0 010101	4	0 000100	,	1 011011
E	1 110101	N	0 100101	W	0 010110	5	1 000101	)	0 011100
F	1 110110	O	0 100110	X	1 010111	6	1 000110	%	1 011101
G	0 110111	P	1 100111	Y	1 011000	7	0 000111	?	0 011111
H	0 111000	Q	1 101000	Z	0 011001	8	0 001000	−	1 100001
I	1 111001	R	0 101001			9	1 001001	@	1 111010
						0	1 001010	$	1 111111

3. 유니코드(Unicode)

- ❖ ASCII 코드의 한계성을 극복하기 위하여 개발된 인터넷 시대의 표준
- ❖ 유니코드 컨소시엄(IBM, Novell, Microsoft, DEC, Apple 등)에 의해서 32(UTF-32), 16(UTF-16), 8bit(UTF-8)의 세 가지 기본 코드로 현재 4.0까지 개발
- ❖ 미국, 유럽, 동아시아, 아프리카, 아시아 태평양 지역 등의 주요 언어들에 적용될 수 있다.
- ❖ 유니코드는 유럽, 중동, 아시아 등 거의 대부분의 문자를 포함하고 있으며, 96,382개의 문자로 구성되어 있다.
- ❖ 특히 아시아의 중국, 일본, 한국, 타이완, 베트남, 싱가포르에서 사용하는 표의 문자(한자) 70,207개를 나타낼 수 있다.
- ❖ 구두표시, 수학기호, 전문기호, 기하학적 모양, 딩벳 기호 등을 포함
- ❖ 앞으로도 계속해서 산업계의 요구나 새로운 문자들을 추가하여 나갈 것이다.

4. 한글코드

❖ 한글은 ASCII코드를 기반으로 16비트를 사용하여 하나의 문자를 표현

❖ 조합형과 완성형으로 분류

❑ 조합형

- 조합형으로 표현된 한글은 때에 따라서 다른 응용프로그램에서는
- 사용할 수 없는 문자들이 많다.
- 조합형은 자음과 모음으로 조합 가능한 모든 한글을 사용할 수 있으며,
- 심지어 우리나라 고어(古語)까지 취급할 수 있는 장점이 있으나,
- 출력 시 다시 모아 써야 하는 불편이 있다는 것이 단점이다.

두번째 바이트								첫번째 바이트							
1															
	초성					중성					종성				

❑ 완성형

- 완성형 한글코드는 1987년 정부가 한국표준으로 정한 것으로 가장 많이 사용되는 한글 음절을 2 바이트의 2 진수와 1 대 1로 대응하여 표현하는 방법이다.

11. 오류검출 코드

1. 패리티 비트

❖ 짝수패리티(even parity) : 데이터에서 1의 개수를 짝수 개로 맞춤

❖ 홀수패리티(odd parity) : 1의 개수를 홀수 개로 맞춤

❖ 패리티 비트는 데이터 전송과정에서 오류 검사를 위한 추가비트. 패리티는 단지 오류 검출만 가능하며, 여러 비트에 오류가 발생할 경우에는 검출이 안 될 수도 있음.

❑ 7비트 ASCII 코드에 패리티 비트를 추가한 코드

데이터	짝수패리티	홀수패리티
…	…	…
A	0 1000001	1 1000001
B	0 1000010	1 1000010
C	1 1000011	0 1000011
D	0 1000100	1 1000100
…	…	…

12. 2진식 기억장치와 레지스터

- 레지스터 – n개의 셀이 있는 레지스터는 n비트의 정보를 이산적인 양의 형태로 저장할 수 있다.
- 레지스터 전이

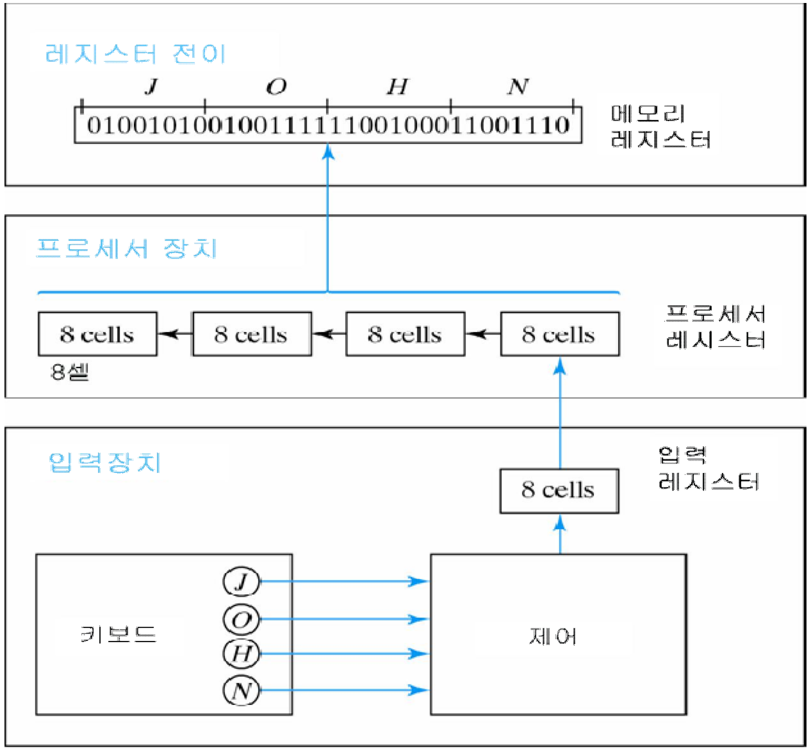


그림 1-1 레지스터에 의한 정보전이

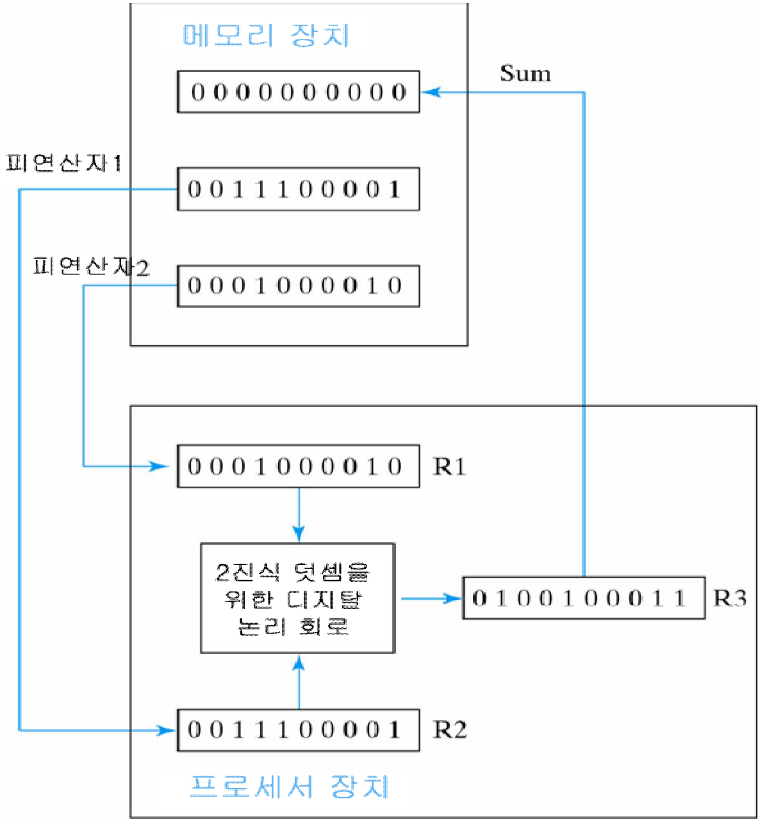


그림 1-2 2진 정보처리의 예

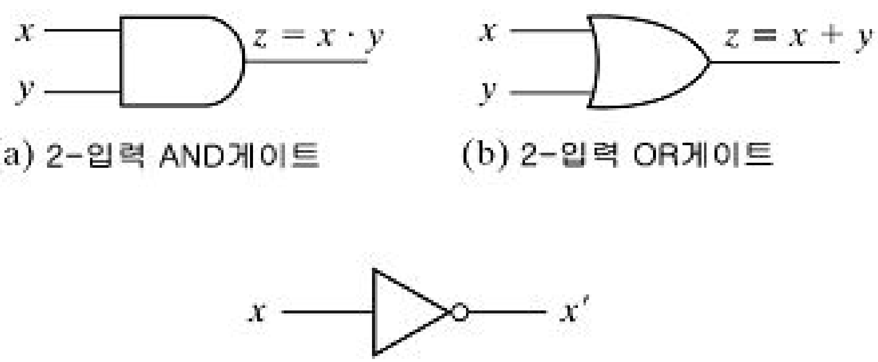
13. Binary Logic

- 2진식 논리의 정의

표 1-8 논리 연산의 진리표

AND			OR			NOT	
x	y	$x \cdot y$	x	y	$x + y$	x	x'
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

- 논리 게이트



(a) 2-입력 AND게이트 (b) 2-입력 OR게이트

(c) NOT게이트 또는 인버터

그림1-4 디지털 논리 회로의 기호

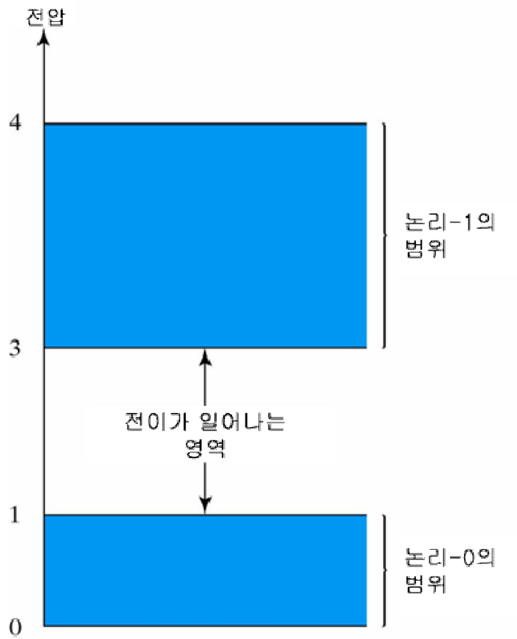


그림 1-3 2진식 신호의 예

Homework

- 1장: 10,11,14,16,18,20,21,22,24,29,32
- 제출 기한: