

2주차 2차시

정보의 표현(2)

정보 체계_컴퓨터 내부의 정보 표현과 정보 처리

1 정보의 표현

2 문자 표현

3 정수 표현

4 실수 표현



컴퓨터의 개요(2)

◆ 학습목표

- 컴퓨터의 정보 표현 방법을 알아본다.
- 컴퓨터에서 문자, 정수, 실수의 표현 방법을 알아본다.

1. 정보의 표현

- ◆ 디지털 컴퓨터는 문자나 숫자 등의 정보를 0과 1의 2진 체계로 부호화한 디지털 데이터로 처리

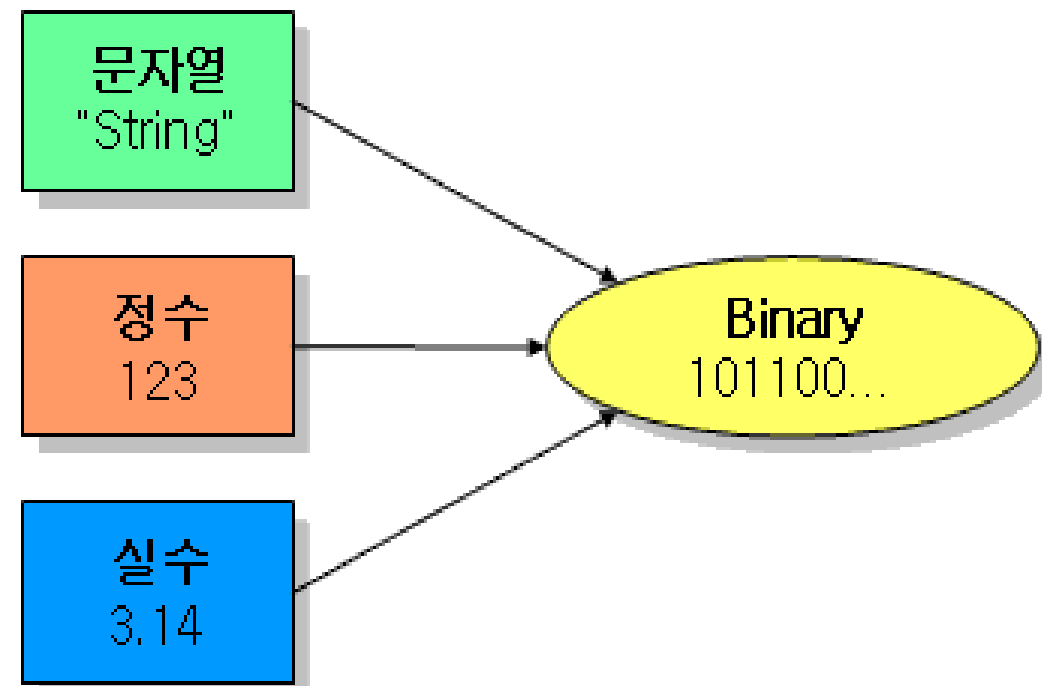


표 2-2 문자 A와 숫자 10의 부호화

정보	2진 체계 부호화
A	01000001
10	00001010

1. 정보의 표현

◆ 비트

- Binary digit는 컴퓨터에서 정보를 나타내는 최소 단위
- 2진수 0 또는 1을 의미
- N비트로 표현할 수 있는 정보는 2^N 개

◆ 바이트

- 문자를 나타내는 최소 단위로 영문자나 숫자
- 특수문자는 1바이트로 표현
- 한글이나 한자는 2바이트로 표현

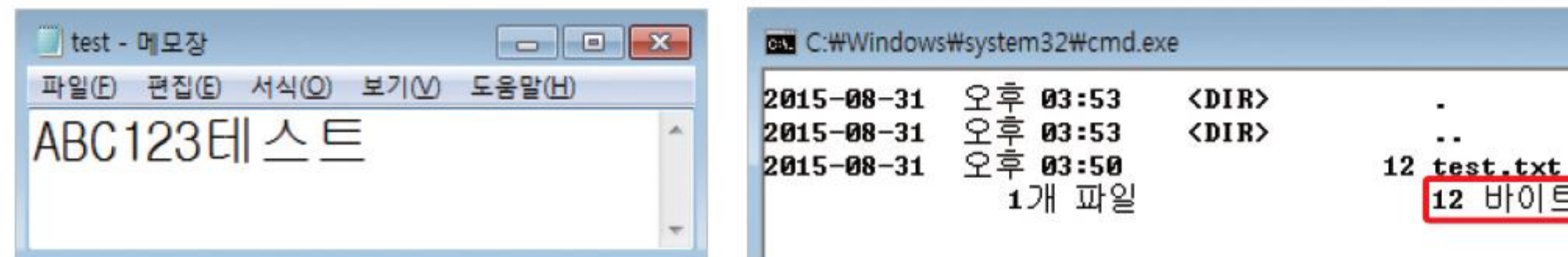


그림 2-6 test.txt 파일의 내용(왼쪽)과 크기(오른쪽)

1. 정보의 표현

◆ 워드

- 명령어나 연산을 처리하는 기본 단위
- 기억장치에 한 번 접근하여 얻을 수 있는 데이터의 양

◆ 기억 용량의 단위

기억 용량 단위	활용 예
KB(Kilo Byte)	20KB의 엑셀 파일
MB(Mega Byte)	4MB의 MP3 파일
GB(Giga Byte)	32GB의 USB 메모리
TB(Tera Byte)	2TB의 외장 하드디스크

2. 문자 표현

1) 아스키 코드

- ◆ 미국표준협회(ANSI)가 데이터를 처리하거나 통신 시스템 간의 정보를 교환할 때 쓸 표준 코드로 제안한 것
- ◆ 표현할 수 있는 문자는 $128(2^7)$ 개



그림 2-7 아스키 코드의 비트 구성

2. 문자 표현

1) 아스키 코드

◆ 아스키 코드표

- 0~31번과 127번 : 제어 문자
- 32~64번 : 특수문자와 숫자
- 65~96번 : 알파벳 대문자와 특수문자
- 97~126번 : 알파벳 소문자와 특수문자

표 2-4 아스키 코드표

10진수	2진수	ASCII	10진수	2진수	ASCII	10진수	2진수	ASCII	10진수	2진수	ASCII
0	0000000	NULL	32	0100000	SP	64	1000000	@	96	1100000	`
1	0000001	SOH	33	0100001	!	65	1000001	A	97	1100001	a
2	0000010	STX	34	0100010	"	66	1000010	B	98	1100010	b
3	0000011	ETX	35	0100011	#	67	1000011	C	99	1100011	c
4	0000100	EOT	36	0100100	\$	68	1000100	D	100	1100100	d
5	0000101	ENQ	37	0100101	%	69	1000101	E	101	1100101	e
6	0000110	ACK	38	0100110	&	70	1000110	F	102	1100110	f
7	0000111	BEL	39	0100111	'	71	1000111	G	103	1100111	g
8	0001000	BS	40	0101000	(72	1001000	H	104	1101000	h
9	0001001	HT	41	0101001)	73	1001001	I	105	1101001	i
10	0001010	LF	42	0101010	*	74	1001010	J	106	1101010	j
11	0001011	VT	43	0101011	+	75	1001011	K	107	1101011	k
12	0001100	FF	44	0101100	,	76	1001100	L	108	1101100	l
13	0001101	CR	45	0101101	-	77	1001101	M	109	1101101	m
14	0001110	SO	46	0101110	.	78	1001110	N	110	1101110	n
15	0001111	SI	47	0101111	/	79	1001111	O	111	1101111	o
16	0010000	DLE	48	0110000	0	80	1010000	P	112	1110000	p
17	0010001	DC1	49	0110001	1	81	1010001	Q	113	1110001	q
18	0010010	SC2	50	0110010	2	82	1010010	R	114	1110010	r
19	0010011	SC3	51	0110011	3	83	1010011	S	115	1110011	s
20	0010100	SC4	52	0110100	4	84	1010100	T	116	1110100	t
21	0010101	NAK	53	0110101	5	85	1010101	U	117	1110101	u
22	0010110	SYN	54	0110110	6	86	1010110	V	118	1110110	v
23	0010111	ETB	55	0110111	7	87	1010111	W	119	1110111	w
24	0011000	CAN	56	0111000	8	88	1011000	X	120	1111000	x
25	0011001	EM	57	0111001	9	89	1011001	Y	121	1111001	y
26	0011010	SUB	58	0111010	:	90	1011010	Z	122	1111010	z
27	0011011	ESC	59	0111011	;	91	1011011	[123	1111011	{
28	0011100	FS	60	0111100	<	92	1011100	\	124	1111100	
29	0011101	GS	61	0111101	=	93	1011101]	125	1111101	}
30	0011110	RS	62	0111110	>	94	1011110	^	126	1111110	~
31	0011111	US	63	0111111	?	95	1011111	_	127	1111111	DEL

2. 문자 표현

2) 2진화 10진 코드(BCD 코드)

- ◆ 문자 하나를 표현하기 위해 6비트를 사용, $64(2^6)$ 개의 문자를 표현
- ◆ 디지털 비트가 0~9까지 가중치 코드로 자릿값을 갖기 때문에 8421 코드라고도 함
- ◆ 2진수 네 자리가 10진수 한 자리에 대응되기 때문에 10진수로 변환하기 쉬움



그림 2-8 BCD 코드의 비트 구성

2. 문자 표현

3) 확장 2진화 10진 코드(EBCDIC 코드)

- ◆ IBM사가 문자 코드에 대한 필요성으로 제정
- ◆ IBM의 메인프레임 컴퓨터에서 사용
- ◆ BCD 코드를 8비트로 확장하여 사용, $256(2^8)$ 개의 문자를 표현

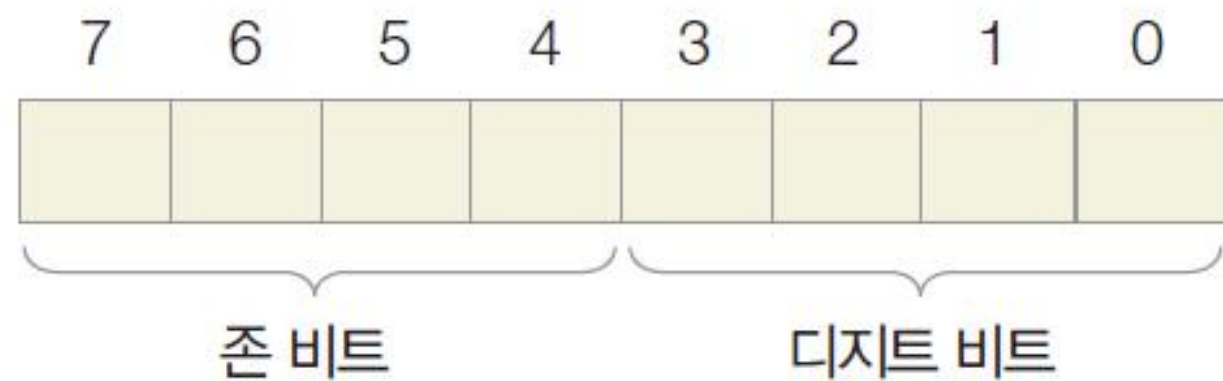


그림 2-9 EBCDIC 코드의 비트 구성

2. 문자 표현

3) 확장 2진화 10진 코드(EBCDIC 코드)

표 2-5 알파벳 대문자와 숫자의 코드표

문자	아스키 코드	BCD 코드	EBCDIC 코드
A	1000001	110001	11000001
B	1000010	110010	11000010
C	1000011	110011	11000011
D	1000100	110100	11000100
E	1000101	110101	11000101
F	1000110	110110	11000110
G	1000111	110111	11000111
H	1001000	111000	11001000
I	1001001	111001	11001001
J	1001010	100001	11010001
K	1001011	100010	11010010
L	1001100	100011	11010011
M	1001101	100100	11010100
N	1001110	100101	11010101
O	1001111	100110	11010110

P	1010000	100111	11010111
Q	1010001	101000	11011000
R	1010010	101001	11011001
S	1010011	010010	11100010
T	1010100	010011	11100011
U	1010101	010100	11100100
V	1010110	010101	11100101
W	1010111	010110	11100110
X	1011000	010111	11100111
Y	1011001	011000	11101000
Z	1011010	011001	11101001
0	0110000	001010	11110000
1	0110001	000001	11110001
2	0110010	000010	11110010
3	0110011	000011	11110011
4	0110100	000100	11110100
5	0110101	000101	11110101
6	0110110	000110	11110110
7	0110111	000111	11110111
8	0111000	001000	11111000
9	0111001	001001	11111001

2. 문자 표현

4) 유니코드

- ◆ 전 세계의 언어를 일관된 방법으로 표현하고 다룰 수 있는 국제적인 문자 코드 규약
- ◆ 문자 하나를 16비트로 표현, 65,536(2^{16})개의 문자와 기호를 나타냄
- ◆ 인코딩 방식은 UTF-8, UTF-16, UTF-32
(UTF 뒤의 숫자는 문자 인코딩에 사용되는 비트 수)
- ◆ 언어별 유니코드 차트 자료 : <http://www.unicode.org/charts>

2. 문자 표현

4) 유니코드

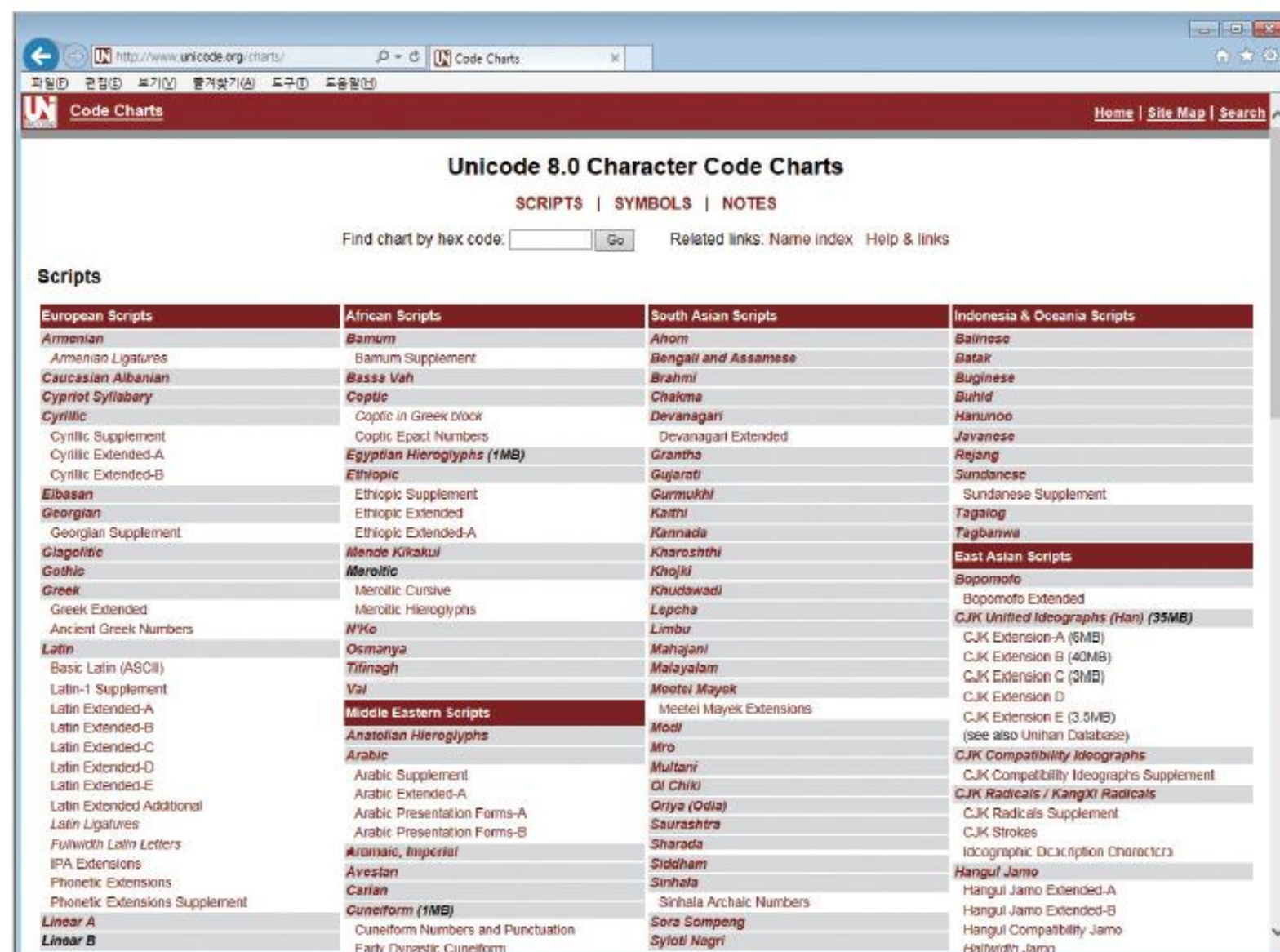


그림 2-10 유니코드 차트 웹사이트

2. 문자 표현

4) 유니코드

	AC0	AC1	AC2	AC3	AC4	AC5	AC6	AC7	AC8	AC9	ACA	ACB	ACC	ACD	ACE	ACF
0	가 AC00	감 AC10	감 AC20	갓 AC30	갈 AC40	각 AC50	갸 AC60	거 AC70	검 AC80	겐 AC90	갹 ACA0	결 ACB0	격 ACC0	겻 ACD0	고 ACE0	곰 ACF0
1	각 AC01	갑 AC11	갸 AC21	갯 AC31	갼 AC41	갽 AC51	갺 AC61	걱 AC71	겁 AC81	겻 AC91	경 ACA1	겪 ACB1	곁 ACC1	겼 ACD1	곡 ACE1	곱 ACF1
2	갻 AC02	갼 AC12	갽 AC22	갿 AC32	갾 AC42	갿 AC52	갿 AC62	긔 AC72	긔 AC82	경 AC92	겻 ACA2	곁 ACB2	곁 ACC2	곁 ACD2	곡 ACE2	곁 ACF2

그림 2-11 한글 유니코드

표 2-6 훈민정음의 유니코드 표현

1101 0110 1100 1000	1011 1011 1111 1100	1100 1000 0001 0101	1100 0111 0100 1100
하	민	정	음

3. 정수 표현

1) 보수

◆ 두 수의 합이 진법의 밑수(N)가 되게 하는 수

- 예를 들어 10진수 4의 10의 보수는 6이고, 10진수 2의 10의 보수는 8이다.

◆ 음의 정수를 표현하기 위해 고안한 개념

◆ 컴퓨터 내부에서는 사칙연산을 할 때 덧셈을 담당하는 가산기를 이용하기 때문에 뺄셈은 덧셈 형식으로 변환하여 계산해야 함

- $A-B$ 는 B 의 보수($-B$)를 구한 후 $A+(-B)$ 로 계산

3. 정수 표현

1) 보수

◆ 1의 보수

- 2진수 1010의 1의 보수는 $0101_{(2)}$ ➡

$$\begin{array}{r} 1111 \\ - 1010 \\ \hline 0101 \end{array}$$

그림 2-12 1의 보수

◆ 2의 보수

- 2진수 1010의 2의 보수는 $0110_{(2)}$ ➡

$$\begin{array}{r} 1111 \\ - 1010 \\ \hline 0101 \\ + \quad 1 \\ \hline 0110 \end{array}$$

그림 2-13 2의 보수

3. 정수 표현

2) 덧셈

◆ 두 수의 합이 2가 되면 자리올림이 발생

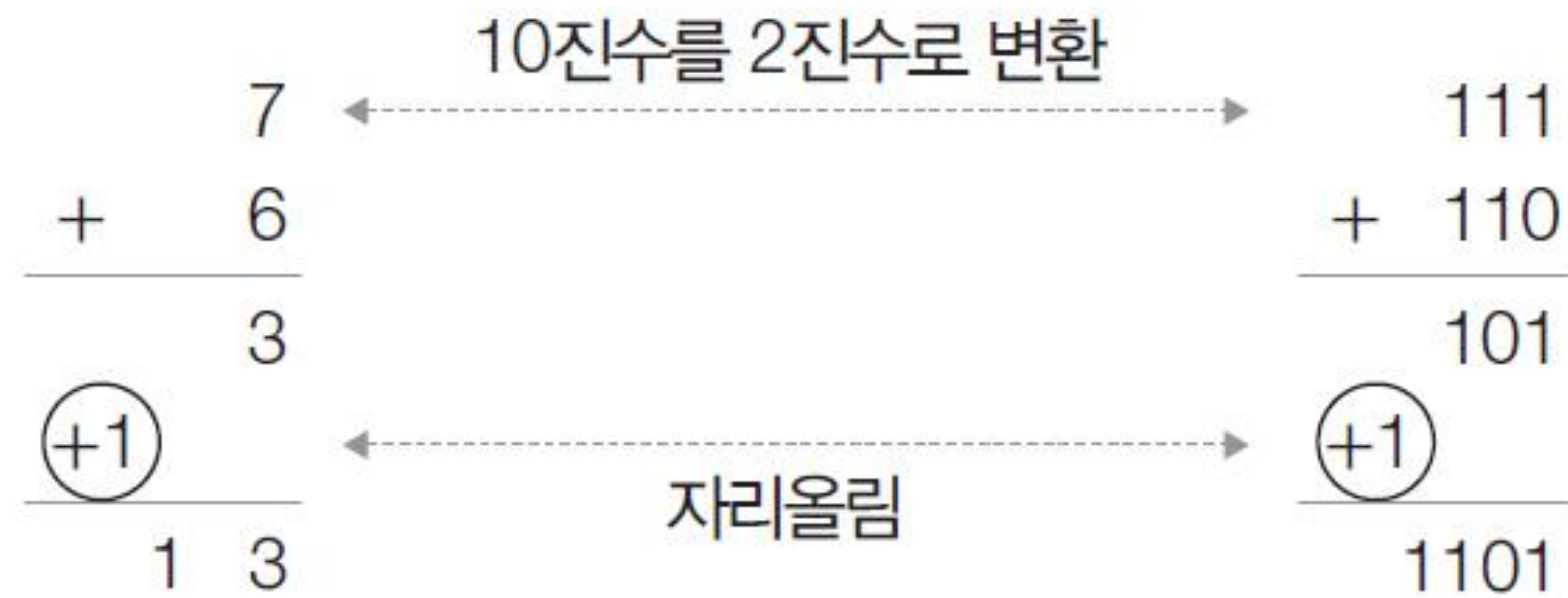


그림 2-14 자리올림이 발생한 덧셈

3. 정수 표현

3) 뺄셈

◆ 1의 보수 뺄셈

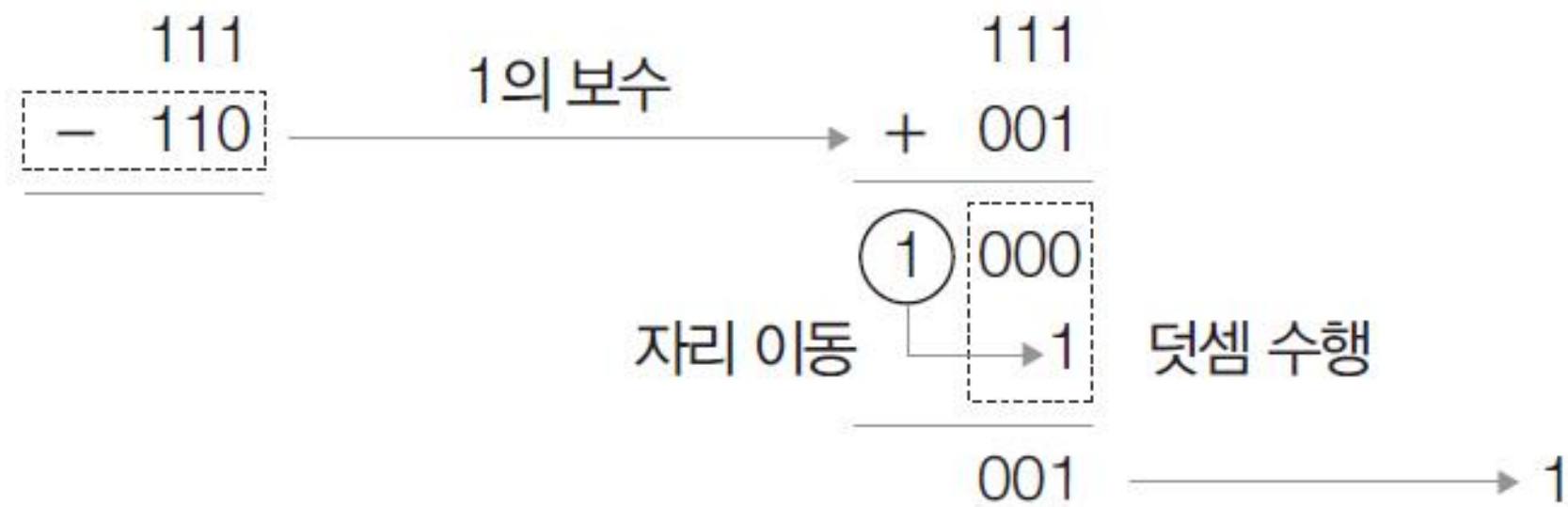


그림 2-15 자리올림이 생긴 1의 보수 뺄셈 : 7 - 6

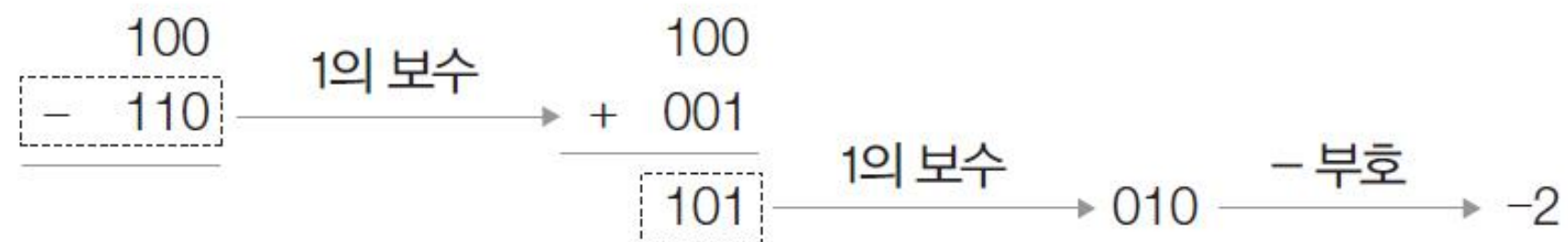


그림 2-16 자리올림이 생기지 않는 1의 보수 뺄셈 : 4 - 6

3. 정수 표현

3) 뺄셈

◆ 2의 보수 뺄셈

$$\begin{array}{r}
 111 \\
 - 110 \\
 \hline
 \end{array}
 \xrightarrow{\text{2의 보수}}
 \begin{array}{r}
 111 \\
 + 010 \\
 \hline
 \end{array}$$

자리올림 제외 (1) 001 \longrightarrow 1

그림 2-17 자리올림이 생긴 2의 보수 뺄셈 : 7 - 6

$$\begin{array}{r}
 100 \\
 - 110 \\
 \hline
 \end{array}
 \xrightarrow{\text{2의 보수}}
 \begin{array}{r}
 100 \\
 + 010 \\
 \hline
 \end{array}$$

2의 보수 $\xrightarrow{\text{2의 보수}}$ 110 $\xrightarrow{\text{부호}}$ -2

그림 2-18 자리올림이 생기지 않은 2의 보수 뺄셈 : 4 - 6

3. 정수 표현

4) 곱셈

- ◆ 피승수에 승수의 각 수를 곱하여 부분 곱을 구함
- ◆ 각 부분 곱은 직전 단계의 부분 곱보다 왼쪽으로 한 비트만큼 시프트한 후 더함

	110011	(피승수)
×	110	(승수)
<hr/>		
	000000	(부분 곱)
	110011	(부분 곱)
+	110011	(부분 곱)
<hr/>		
	100110010	(결과)

그림 2-19 2진수 110011과 110의 곱셈

3. 정수 표현

5) 나눗셈

- ◆ 피제수에서 제수를 뺄 수 없을 때까지 뺄셈을 계속해서 횡수는 몫이 되고 남은 것은 나머지가 됨

$$\begin{array}{r}
 \text{(제수)} \quad 110 \overline{) 100110} \quad \begin{array}{l} 110 \text{ (몫)} \\ \text{(피제수)} \end{array} \\
 \underline{-110} \\
 111 \quad \text{(부분 나머지)} \\
 \underline{-110} \\
 10 \quad \text{(나머지)}
 \end{array}$$

그림 2-20 2진수 100110과 110의 나눗셈

3. 정수 표현

6) 고정 소수점 표현

◆ 소수점이 고정된 위치에 있다는 뜻



그림 2-21 고정 소수점 표현

3. 정수 표현

6) 고정 소수점 표현

◆ 정수 표현에 사용

- 부호화 절댓값 방식: 부호비트 1과 2진수의 절댓값으로 나타냄
- 1의 보수 방식: 부호비트와 2진수의 절댓값에 대한 1의 보수로 나타냄
- 2의 보수 방식: 부호비트와 2진수의 절댓값에 대한 2의 보수로 나타냄

7	6	5	4	3	2	1	0
0	절댓값						

(a) 양의 정수 표현

7	6	5	4	3	2	1	0
1	절댓값						

7	6	5	4	3	2	1	0
1	1의 보수						

7	6	5	4	3	2	1	0
1	2의 보수						

(b) 음의 정수 표현

그림 2-22 양수와 음수의 고정 소수점 표현

3. 정수 표현

6) 고정 소수점 표현

표 2-7 정수 +13과 -13에 대한 8비트 표현

표현 방식	+13	-13
부호화 절댓값	0 0001101	1 0001101
1의 보수	0 0001101	1 1110010
2의 보수	0 0001101	1 1110011

표 2-8 3비트로 표현할 수 있는 수의 범위

10진수	부호화 절댓값	1의 보수	2의 보수
-4	-	-	100
-3	111	100	101
-2	110	101	110
-1	101	110	111
-0	100	111	-
+0	000	000	000
+1	001	001	001
+2	010	010	010
+3	011	011	011

4. 실수 표현

◆ 고정 소수점 방식

- 소수점이 항상 고정된 위치에 있다는 의미로, 정수 표현에 주로 사용

◆ 부동 소수점 방식

- 소수점의 위치가 변하기 때문에 실수 표현에 주로 사용
- 고정 소수점 방식보다 넓은 범위의 수를 표현

$$m \times r^e \quad (m : \text{가수} \quad r : \text{밑수}, e : \text{지수})$$

- 표현범위에 따라서 4바이트의 단일 정밀도 형식과 8바이트의 이중 정밀도 형식으로 나뉨



(a) 단일 정밀도 형식



(b) 이중 정밀도 형식

그림 2-23 부동 소수점 표현 형식

4. 실수 표현

◆ 정규화 : 2진 소수를 1.XXX로 표현하는 과정

2진수 10110.11011을 정규화하면 1.011011011×2^4



31	30	23	22	0
0	10000011	011011011000000000000000		

그림 2-24 2진수 10110.11011의 부동 소수점 표현



요약

◆ 정보의 표현단위

- 비트: 정보를 표현하는 최소 단위로, N개의 비트로 표현 할 수 있는 정보의 수는 2^N 개이다.
- 바이트: 문자를 표현하는 최소 단위로, 8개의 비트로 구성된다. 1바이트는 $256(2^8)$ 개의 서로 다른 데이터를 표현할 수 있다.
- 워드: 명령이나 연산을 처리하는 기본 단위로, 기억장치에 한번 접근하여 얻을 수 있는 데이터의 양이다.



요약

◆ 코드의 종류

- 아스키 코드: 미국 표준협회가 제정한 데이터 처리 및 통신 시스템 상호간에 정보 교환용 표준코드이다.
- 2진화 10진 코드: 문자 하나를 표현하기 위해 6비트 사용하므로 $64(2^6)$ 개의 문자를 표현할 수 있다. 자릿값을 갖는 가중치 코드로 8421 코드라고도 한다.
- 확장 2진화 10진코드: 기존의 BCD코드를 8비트로 확장한 코드로 $256(2^8)$ 개의 문자를 표현 할수 있다.