

# 2장 / 자료의 표현

정보처리산업기사

# SECTION 43

## 자료구성의 단위

# 자료 구성의 단위

- 컴퓨터에서 사용하는 자료는 컴퓨터 내부에서 사용하는 비트, 바이트, 워드와
- 사람이 인식하여 사용할 수 있는 필드, 레코드, 파일, 데이터베이스로 구성되어 있다.
- 비트(Bit, Binary Digit)
  - 자료(정보)표현의 최소 단위이다.
  - 두 가지 상태(0또는 1)를 표시하는 2진수 1자리이다.

비트	표현 가짓수	표현 범위
1Bit	$2^1 = 2$ 가지	0,1
2Bit	$2^2 = 4$ 가지	00,01,10,11
3Bit	$2^3 = 8$ 가지	000,001,010,011,100,101,110,111
4Bit	$2^4 = 16$ 가지	0000,0001,0010,0011,0100,0101,0110,0111,1000,1001,1010,1011,1100,1101,1110,1111
...	...	
nBit	$2^n$	

# 자료 구성의 단위

- 니블(Nibble)

- 네 개의 비트(Bit)가 모여 한 개의 니블을 구성한다.
- 4비트로 구성되며 16진수 1자리를 표현하기에 적합하다.

- 바이트(Byte)

- 문자를 표현하는 최소 단위로, 8개의 비트가 모여 1Byte를 구성한다.
- 1Byte는  $256(=2^8)$ 가지의 정보를 표현할 수 있다.
- 주소 지정의 단위로 사용된다.
- 일반적으로 영문자나 숫자는 1Byte로 1자를 표현하고 한글, 한자는 2Byte로 1자를 표현한다.

# 기억 용량의 단위

단위	Byte★	KB	MB	GB	TB	PB	EB★
읽기	바이트 (Byte)	킬로바이트 (Kilo Byte)	메가바이트 (Mega Byte)	기가바이트 (Giga Byte)	테라바이트 (Tera Byte)	페타바이트 (Peta Byte)	엑사바이트 (Exa Byte)
용량	8Bit	1,024Byte	1,024KB	1,204MB	1,204GB	1,204TB	1,204PB
	2진수 표기	$2^{10}$	$2^{20}$	$2^{30}$	$2^{40}$	$2^{50}$	$2^{60}$
	10진수 표기(약)	$10^3$	$10^6$	$10^9$	$10^{12}$	$10^{15}$	$10^{18}$

용량적용  
★



용량 크기  
☆

단 위	...	T (Tera)	G (Giga)	M (Mega)	K (Kilo)	m (Milli)	μ (Micro)	n (Nano)	p (Pico)	...
10진수	...	$10^{12}$	$10^9$	$10^6$	$10^3$	$10^{-3}$	$10^{-6}$	$10^{-9}$	$10^{-12}$	...
2진수	...	$2^{40}$	$2^{30}$	$2^{20}$	$2^{10}$	-	-	-	-	...
일반적 사용용도	← 저장 매체의 용량 표현 →					← 처리 속도 표현 →				

# 자료구성단위

- 워드 - 컴퓨터가 한 번에 처리할 수 있는 명령 단위. 반워드(Half-Word) : 2Byte, 전워드(Full-Word) : 4Byte, 더블워드(Double-Word) : 8Byte
- 필드 - 파일 구성의 최소 단위. 의미 있는 정보를 표현하는 최소 단위
- 레코드 - 하나 이상의 관련된 필드가 모여서 구성. 컴퓨터 내부의 자료 처리 단위로서, 일반적으로 레코더는 논리 레코드를 의미한다.
- 블록 - 하나 이상의 논리 레코드가 모여서 구성된다. 각종 저장 매체와의 입출력 단위를 의미하며, 일반적으로 물리레코드라고 한다.
- 파일 - 프로그램 구성의 기본 단위로, 같은 종류의 여러 레코드가 모여서 구성된다.
- 데이터베이스 - 여러 개의 관련된 파일의 집합이다. 관계형, 계층형, 망형 데이터 베이스가 있다.



# SECTION 44

수의 표현 및 진법 변환

# 진법

- 컴퓨터 내부에서는 2진법을 사용하여 모든 연산을 수행한다.
- 2진수 외에 컴퓨터를 연구할 때 자주 사용하는 진법은 8진수와 16진수이다.

2진법(Binary)	0과 1, 두 개의 숫자로 표현한다
8진법(Octal)	0~7까지의 숫자로 표현하며, 2진수 3자리를 묶어서 하나의 수로 표현한다
10진법 (Decimal)	0~9까지의 숫자로 표현한다.
16진법 (Hexadecimal)	0~9까지의 숫자와 10~15까지를 의미하는 A~F까지 의 문자로 표현한다.

10진수	2진수	8진수	16진수
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	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



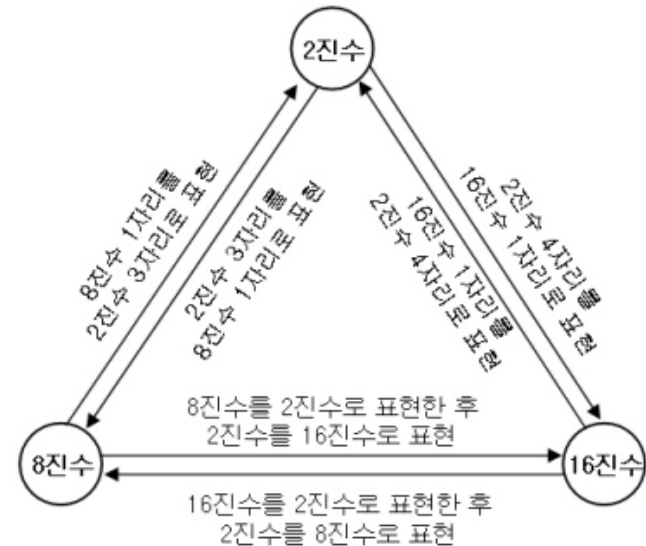
# 진법변환

- 10진수를 2진수, 8진수, 16진수로 변환

-정수 부분 : 10진수의 값을 변환할 진수로 나누어 더 이상 나뉘지지 않을 때까지 나누고, 나머지를 역순으로 표시한다.

-소수 부분 : 10진수의 값에 변환할 진수를 곱한 후 결과의 정수 부분만을 차례대로 표기하되, 소수 부분이 0 또는 반복되는 수가 나올 때 까지 곱하기를 반복한다.

- 정수 부분과 소수 부분의 각 자리를 분리하여 변환하려는 각 진수의 자리값과 자리의 지수승을 곱한 결과값을 모두 더하여 계산한다.



# 기출문제

# SECTION 45

보수

# 보수

- 보수는 컴퓨터가 기본적으로 수행하는 덧셈 회로를 이용하여 뺄셈을 수행하기 위해 사용한다.
- R진법에는  $r$ 의 보수와  $r-1$ 의 보수가 존재한다.

R의 보수	10진법에는 10의 보수가 있고, 2진법에는 2의 보수가 있다. 보수를 구할 숫자의 자릿수만큼 0을 채우고 가장 왼쪽에 1을 추가하여 기준을 만든다.
R-1의 보수	10진법에는 9의 보수가 있고, 2진법에는 1의 보수가 있다. 10진수 N에 대한 9의 보수는 주어진 숫자의 자릿수만큼 9를 채워 기준을 만든다.

- 보수를 이용한 뺄셈
  - $A-B$ 는  $A+(-B)$ 이므로 B에 대한 보수를 구하여 덧셈 연산으로 한다.
  - 1의 보수 이용 : 1의 보수를 이용할 때는 자리올림이 발생하면 결과에 자리올림수를 더한다.
  - 2의 보수 이용 : 2의 보수를 이용할 때는 자리올림이 발생하면 자리올림수를 버린다.
- 예제보기(p.260)

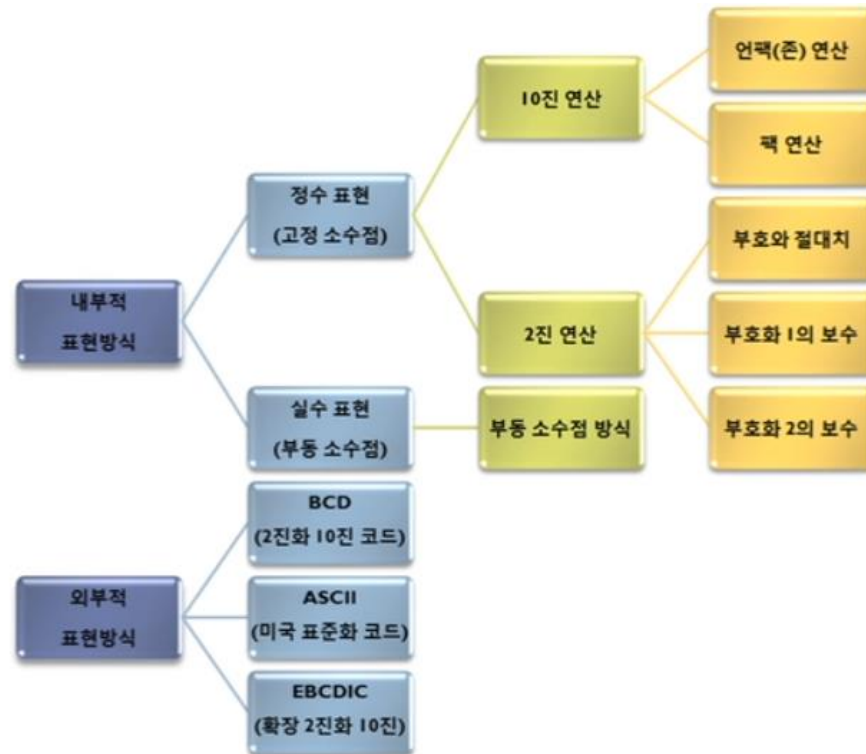
# 기출문제

# SECTION 46

자료의 내부적 표현  
- 고정 소수점 표현

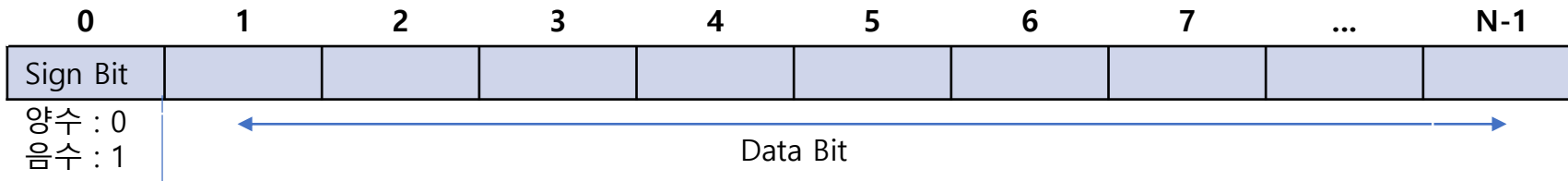
# 자료의 내부적 표현

- 고정 소수점 표현 방식은 정수 데이터의 표현 및 연산에 사용하는 방법으로, 2진 연산과 10진 연산이 있다.



# 2진 연산

- 2진 연산은 정수값을 2진수로 변환하여 표현하는 방식이다.
- $n$  Bit 크기의 워드가 있을 때 다음과 같이 맨 처음 1Bit는 부호비트로 사용되고 나머지  $n-1$  Bit에 2진수로 표현된 정수값이 저장된다.



- 양수 : 부호 비트에 0을 넣고, 변환된 2진수 값을 Data Bit의 오른쪽에서 왼쪽 순으로 차례로 채우고 남은 자리에 0을 채운다.
- 음수 : 음수를 표현할 때는 다음과 같은 세 가지 방법을 사용한다.

종류	표현 방법	비고
부호화 절대치법 (Signde Magnitude)	양수 표현에 대하여 부호 비트의 값만 0을 1로 바꾼다	두가지 형태의 0 존재 (+0, -0)
부호화 1의 보수법 (Signde 1's Complement)	양수 표현에 대하여 1의 보수를 취한다.	
부호화 2의 보수법 (Signed 2's Complement)	양수 표현에 대하여 2의 보수를 취한다.	한 가지 형태의 0만 존재 (+0)



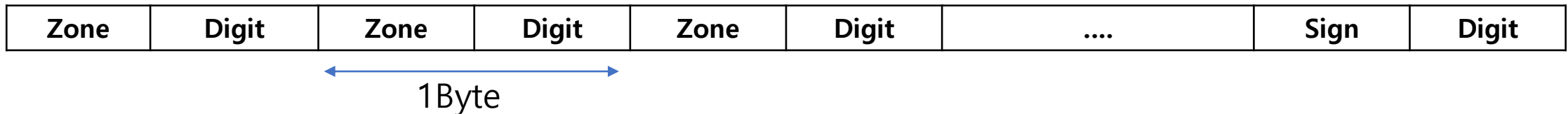
## 2진 연산 (3/3)

### 표현 범위

종류	범위	N = 8	N = 16	N = 32
부호화 절대치법	$-2^{n-1}+1 \sim +2^{n-1}-1$	-127 ~ +127	-32767 ~ +32767	$-2^{31-1}+1 \sim +2^{31-1}-1$
부호화 1의 보수법				
부호화 2의 보수법	$-2^{n-1} \sim +2^{n-1}-1$	-128 ~ +127	-32768 ~ +32767	$-2^{31} \sim +2^{31-1}-1$

# 10진 연산

- 10진 연산은 10진수 1자리를 2진수 4자리로 표현하는 방식으로, 언팩(Unpack)연산과 팩(Pack)연산이 있다.
- 언팩(Unpack) 연산
  - 존형 10진 연산이라고도 한다.
  - 연산이 불가능하고, 데이터의 입출력에 사용된다.
  - 1Byte로 10진수 1자리를 표현한다.
  - 4개의 존 비트와 4개의 숫자비트를 사용한다.
  - 최하위 바이트의 존 부분을 부호로 사용된다.



Zone 부분 : 무조건 1111을 넣는다.

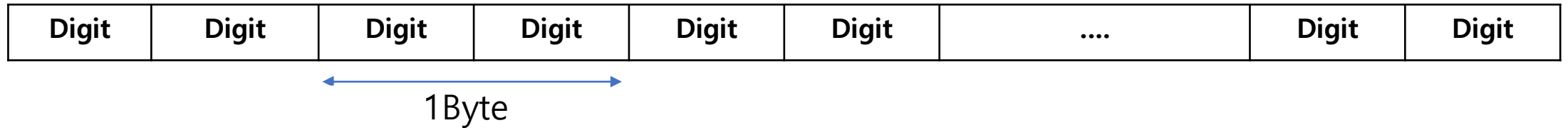
Digit 부분 : 10진수 1자리를 4Bit 2진수로 표현한다.

Sign 부분 : 양수는  $C(1100_2)$ , 음수는  $C(1101_2)$ , 부호 없는 양수는  $F(1111_2)$ 로 표현한다.

# 10진 연산

## ○ 팩(Pack)연산

- 연산이 가능하고, 데이터의 입출력이 불가능하다.
- 1Byte로 10진수 2자리를 표현한다.
- 최하위(가장 오른쪽) 바이트의 4Bit 부분을 부호로 사용한다.



Digit 부분 : 10진수 1자리를 4Bit 2진수로 표현한다.

Sign 부분 : 양수는  $C(1100_2)$ , 음수는  $C(1101_2)$ , 부호 없는 양수는  $F(1111_2)$ 로 표현한다.

# 기출문제

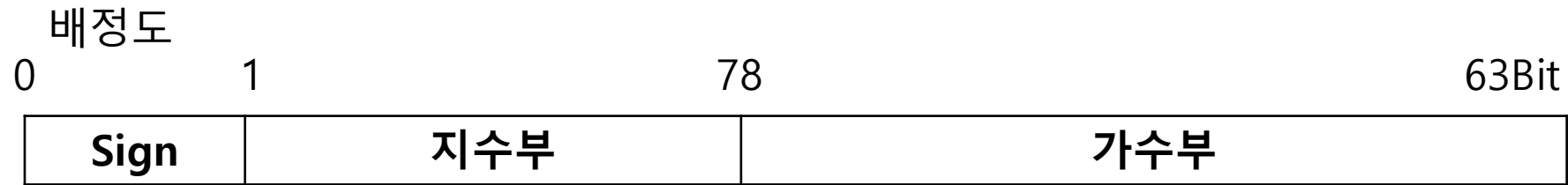
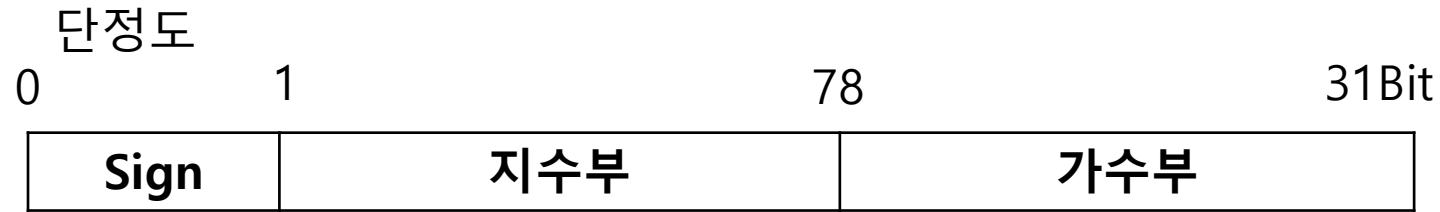
# SECTION 47

자료의 내부적 표현  
-부동 소수점 표현

# 부동 소수점 방식의 특징 (1/3)

- 고정 소수점 방식으로 표현하는 것보다 매우 큰 수나 작은 수, 매우 정밀한 수를 적은 비트로 표현할 수 있다.
- 과학이나 공학 또는 수학적인 응용에 주로 사용된다.
- 고정 소수점 방식에 비해 연산 시간이 많이 걸린다
- 부동 소수점 연산의 수행 횟수를 FLOPS로 표시하며, 이는 컴퓨터의 연산 속도를 나타내는 단위로 사용된다.
- 지수부와 가수부를 분리하는 정규화 과정이 필요하다.
- 4Byte를 사용하는 단정도와 가수부를 4Byte 추가하여 좀더 정밀하게 표현할 수 있는 8Byte 배정도 표현법이 있다.
- 표현 범위는 사용하는 기계와 Base를 무엇으로 하느냐에 따라 달라진다. 16진수를 Base로 하는 단정도 표현의 경우  $16^{-64} \sim 16^{+63}$ 까지의 수치를 표현할 수 있다.

## 부동 소수점 방식의 특징 (2/3)



- Sign : 양수 = 0, 음수 = 1
- 지수부 : 정규화시켜 분리한 지수값을 64Bias법으로 표현한다.
- 가수부 : 정규화시켜 분리한 소수 이하를 2진수로 변환하여 왼쪽에서부터 표현하고 빈 자리는 0으로 채운다

## 부동 소수점 방식의 특징 (3/3)

- 예제 (p.271)  
+36.125를 부동소수점 방식으로 표현하시오.
- IEEE 표준에 따른 단정도 부동 소수점 수 표현



# 부동소수점 수의 연산방법

- 덧셈, 뺄셈 : 0인지 여부를 조사한다. → 가수의 위치 조정(지수가 큰 쪽에) → 가수부끼리 더하거나 뺀다. → 결과를 정규화한다.
- 곱셈 : 0인지의 여부를 조사한다. → 지수를 더한다. → 가수를 곱한다. → 결과를 정규화한다.
- 나눗셈 : 0인지의 여부를 조사한다. → 부호를 결정한다. → 피제수가 제수보다 작게 피제수의 위치를 조정한다. → 지수의 뺄셈을 한다. → 가수의 나눗셈을 한다.

# 기출문제

# SECTION 48

자료의 외부적 표현

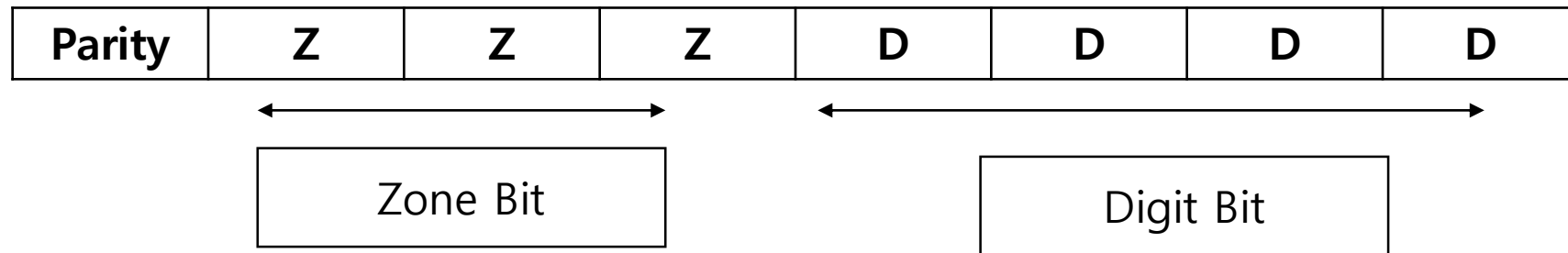
# 자료의 외부적 표현

- 자료의 외부적 표현은 처리된 결과를 사람이 확인할 수 있도록 출력할 때의 문자를 표현하는 방식이다.
- BCD (Binary Coded Decimal, 2진화 10진코드) : 6Bit 코드로 IBM에서 개발. 1개의 문자를 2개의 Zone 비트와 4개의 Digit 비트로 표현한다. 6Bit는  $2^6$ 개를 표현할 수 있으므로 64개의 문자를 표현할 수 있다. 영문 소문자를 표현하지 못한다.

패리티 비트	존 비트		숫자 비트			
6	5	4	3	2	1	0
하위 비트에 따라 달라짐	1	1	영문자 A~I(0001~1001)			
	1	0	영문자 J~R(0001~1001)			
	0	1	영문자 S~Z(0010~1001)			
	0	0	숫자 0~9(0001~1010)			
	혼용		특수 문자 및 기타 문자			

## ASCII 코드 (1/2)

- ASCII 코드 : 7Bit 코드로 미국 표준협회에서 개발. 1개의 문자를 3개의 Zone 비트와 4개의 Digit 비트로 표현.  $2^7 = 128$ 가지의 문자를 표현할 수 있다. 1Bit의 Parity Bit를 추가하여 8Bit로 사용한다. 통신 제어용 및 마이크로컴퓨터의 기본 코드로 사용한다.



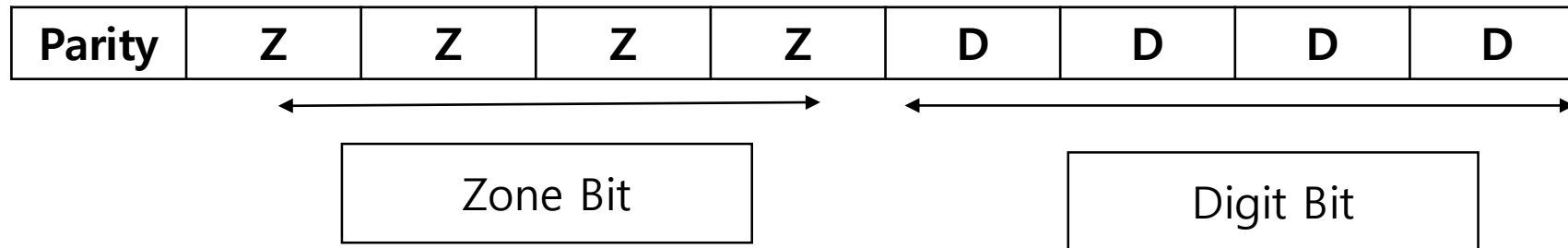
# ASCII 코드 (2/2)

Decimal - Binary - Octal - Hex – ASCII  
Conversion Chart

Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII
0	00000000	000	00	NUL	32	00100000	040	20	SP	64	01000000	100	40	@	96	01100000	140	60	`
1	00000001	001	01	SOH	33	00100001	041	21	!	65	01000001	101	41	A	97	01100001	141	61	a
2	00000010	002	02	STX	34	00100010	042	22	"	66	01000010	102	42	B	98	01100010	142	62	b
3	00000011	003	03	ETX	35	00100011	043	23	#	67	01000011	103	43	C	99	01100011	143	63	c
4	00000100	004	04	EOT	36	00100100	044	24	\$	68	01000100	104	44	D	100	01100100	144	64	d
5	00000101	005	05	ENQ	37	00100101	045	25	%	69	01000101	105	45	E	101	01100101	145	65	e
6	00000110	006	06	ACK	38	00100110	046	26	&	70	01000110	106	46	F	102	01100110	146	66	f
7	00000111	007	07	BEL	39	00100111	047	27	'	71	01000111	107	47	G	103	01100111	147	67	g
8	00001000	010	08	BS	40	00101000	050	28	(	72	01001000	110	48	H	104	01101000	150	68	h
9	00001001	011	09	HT	41	00101001	051	29	)	73	01001001	111	49	I	105	01101001	151	69	i
10	00001010	012	0A	LF	42	00101010	052	2A	*	74	01001010	112	4A	J	106	01101010	152	6A	j
11	00001011	013	0B	VT	43	00101011	053	2B	+	75	01001011	113	4B	K	107	01101011	153	6B	k
12	00001100	014	0C	FF	44	00101100	054	2C	,	76	01001100	114	4C	L	108	01101100	154	6C	l
13	00001101	015	0D	CR	45	00101101	055	2D	-	77	01001101	115	4D	M	109	01101101	155	6D	m
14	00001110	016	0E	SO	46	00101110	056	2E	.	78	01001110	116	4E	N	110	01101110	156	6E	n
15	00001111	017	0F	SI	47	00101111	057	2F	/	79	01001111	117	4F	O	111	01101111	157	6F	o
16	00010000	020	10	DLE	48	00110000	060	30	0	80	01010000	120	50	P	112	01110000	160	70	p
17	00010001	021	11	DC1	49	00110001	061	31	1	81	01010001	121	51	Q	113	01110001	161	71	q
18	00010010	022	12	DC2	50	00110010	062	32	2	82	01010010	122	52	R	114	01110010	162	72	r
19	00010011	023	13	DC3	51	00110011	063	33	3	83	01010011	123	53	S	115	01110011	163	73	s
20	00010100	024	14	DC4	52	00110100	064	34	4	84	01010100	124	54	T	116	01110100	164	74	t
21	00010101	025	15	NAK	53	00110101	065	35	5	85	01010101	125	55	U	117	01110101	165	75	u
22	00010110	026	16	SYN	54	00110110	066	36	6	86	01010110	126	56	V	118	01110110	166	76	v
23	00010111	027	17	ETB	55	00110111	067	37	7	87	01010111	127	57	W	119	01110111	167	77	w
24	00011000	030	18	CAN	56	00111000	070	38	8	88	01011000	130	58	X	120	01111000	170	78	x
25	00011001	031	19	EM	57	00111001	071	39	9	89	01011001	131	59	Y	121	01111001	171	79	y
26	00011010	032	1A	SUB	58	00111010	072	3A	:	90	01011010	132	5A	Z	122	01111010	172	7A	z
27	00011011	033	1B	ESC	59	00111011	073	3B	;	91	01011011	133	5B	[	123	01111011	173	7B	{
28	00011100	034	1C	FS	60	00111100	074	3C	<	92	01011100	134	5C	\	124	01111100	174	7C	
29	00011101	035	1D	GS	61	00111101	075	3D	=	93	01011101	135	5D	]	125	01111101	175	7D	}
30	00011110	036	1E	RS	62	00111110	076	3E	>	94	01011110	136	5E	^	126	01111110	176	7E	~
31	00011111	037	1F	US	63	00111111	077	3F	?	95	01011111	137	5F	_	127	01111111	177	7F	DEL

# EBCDIC. 확장 2진화 10진코드 (1/2)

- 8Bit 코드로 IBM에서 개발.
- 1개의 문자를 4개의 Zone비트와 4개의 Digit 비트로 표현.
- $2^8 = 256$ 가지의 문자를 표현할 수 있다.
- 1Bit의 Parity Bit를 추가하여 9Bit로 사용한다.



# EBCDIC. 확장 2진화 10진코드 (2/2)

16진		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	2진	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	0000	NUL	SOH	STX	ETX		HT		DEL				VT	FF	CR	SO	SI
1	0001	DLE						BS		CAN	EM			IFS	IGS	IRS	IUS
2	0010						LF	ETB	ESC						ENQ	ACK	BEL
3	0011			SYN					EOT						NAK		SUB
4	0100	space										[	.		(	+	
5	0101	&										!	\$	*	)		^
6	0110	-	/										,	%	_	>	?
7	0111										`	:	#	@	'	=	"
8	1000		a	b	c	d	e	f	g	h	i						
9	1001		j	k	l	m	n	o	p	q	r						
A	1010		~	s	t	u	v	w	x	y	z						
B	1011																
C	1100	{	A	B	C	D	E	F	G	H	I						
D	1101	}	J	K	L	M	N	O	P	Q	R						
E	1110	\		S	T	U	V	W	X	Y	Z						
F	1111	0	1	2	3	4	5	6	7	8	9						



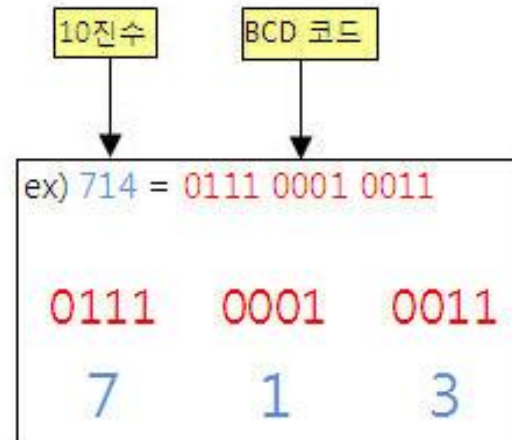
# SECTION 49

기타 자료의 표현 방식

# BCD 코드

- BCD 코드 : 10진수 1자리의 수를 2진수 4Bit로 표현한다. 4Bit의 2진수 각 Bit가  $8(2^3)$ ,  $4(2^2)$ ,  $2(2^1)$ ,  $1(2^0)$  의 자리값을 가지므로 8421 코드라고도 한다. 대표적인 가중치 코드이다. 문자 코드인 BCD에서 Zone 부분을 생략한 형태이다. 10진수 입출력이 간편하다.

10진수	2진수
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001



## Excess-3 코드 (3초과 코드)

- BCD + 3, 즉 BCD코드에  $3_{10}(0011_2)$ 을 더하여 만든 코드이기 때문에 모든 비트가 동시에 0이 되는 경우는 없다.
- 대표적인 자기 보수(self complementing) 코드이며, 비가중치 코드이다.

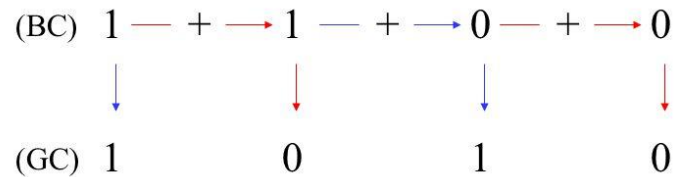
10진수	8421	3초과 코드	9의 보수
0	0000	0011	1100
1	0001	0100	1011
2	0010	0101	1010
3	0011	0110	1001
4	0100	0111	1000
5	0101	1000	0111
6	0110	1001	0110
7	0111	1010	0101
8	1000	1011	0100
9	1001	1100	0011

# Gray 코드 (1/2)

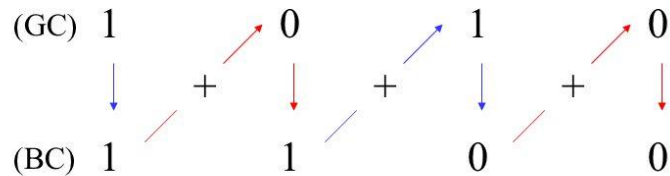
- BCD 코드의 인접하는 비트를 X-OR 연산하여 만든 코드. 이웃하는 코드가 한 비트만 다르기 때문에 코드 변환이 용이하다. 입출력장치, A/D변환기, 주변장치 등에서 숫자를 표현할 때 사용한다. 1Bit만 변화시켜 다음 수치로 증가시키기 때문에 하드웨어적인 오류가 적다.
- 2진수를 Gray Code로 변환하는 방법
  1. 첫 번째 그레이 비트는 2진수 비트를 그대로 내려쓴다.
  2. 두 번째 Gray Bit부터는 변경할 2진수의 해당 번째 비트와 그 왼쪽의 비트를 XOR 연산하여 쓴다.
- Gray Code를 2진수로 변환하는 방법
  1. 첫 번째 2진수 비트는 그레이 비트를 그대로 내려쓴다.
  2. 두 번째 2진수 비트부터는 왼쪽에 구해 놓은 2진수 비트와 변경할 그레이 코드의 해당 번째 비트를 XOR 연산하여 쓴다.

# Gray 코드 (2/2)

## Binary to Gray Code Conversion



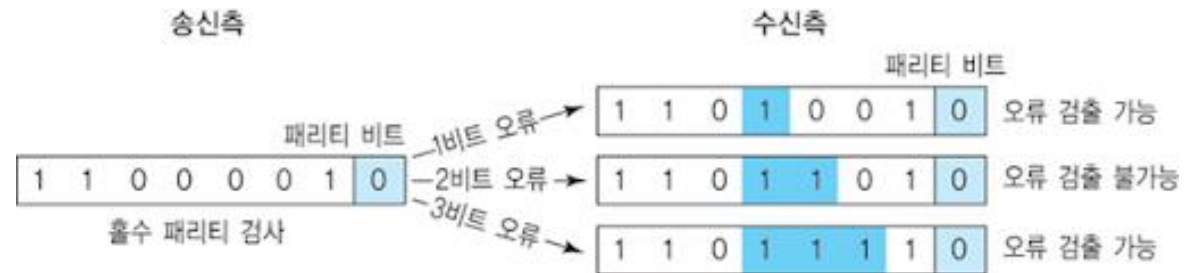
## Gray to Binary Code Conversion



Decimal Number	4 bit Binary Number <u>ABCD</u>	4 bit Gray Code <u>G<sub>1</sub>G<sub>2</sub>G<sub>3</sub>G<sub>4</sub></u>
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

# 패리티 검사 코드

- 전송된 코드의 오류를 검사하기 위해서 데이터 비트 외에 1Bit의 패리티 체크 비트를 추가하는 것으로 1Bit의 오류만 검출할 수 있다.
- 1의 개수에 따라 짝수(Even, 우수)패리티와 홀수(Odd,기수) 패리티 방법이 있다. Odd 패리티는 코드에서 1인 비트의 수가 홀수가 되도록, Even 패리티는 코드에서 1인 비트의 수가 짝수가 되도록, 0이나 1을 추가한다.



# 해밍 코드

- 해밍 코드는 오류를 스스로 검출하여 교정이 가능하다.
- 해밍 코드는 2Bit의 오류를 검출할 수 있고, 1Bit의 오류를 교정할 수 있다.
- 데이터 비트 외에 오류 검출 및 교정을 위한 잉여 비트가 많이 필요하다.
- 해밍 코드 중 1,2,4,8,16 ....  $2^n$  번째는 오류 검출을 위한 패리티 비트이다.

비트 번호	1	2	3	4	5	6	7	8	9	10	11
정보 비트			0		0	1	1		0	0	1
패리티 비트	P1	P2		P3				P4			
	0	1		0				1			
P1	P1		0		0		1		0		1
P2		P2	0			1	1			0	1
P3				P3	0	1	1				
P4								P4	0	0	1
해밍코드	0	1	0	0	0	1	1	1	0	0	1

# 코드의 분류

분류	코드 종류
가중치 코드 (Weight Code)	BCD(8421), 2421, 84-2-1, Biquinary(5043210), 51111, Ring-Counter(9876543210)
비가중치 코드 (Non-Weight Code)	3초과(Excess-3), Gray, Jonson, 2-out-of-5, 3-out-of-5
자기 보수 코드 (Self-Complement Code)	3초과(Excess-3), 2421, 51111, 84-2-1
오류 검출용 코드	해밍 코드, 패리티 검사 코드, Biquinary, Ring-Counter, 2-out-of-5, 3-out-of-5



# 코드의 분류

10 진수	8421 코드 (BCD)	2421 코드	5421 코드	84-2-1 코드	51111 코드	바이퀴너리 코드 (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	<u>1001</u>	1010	11000	1000010	0001000000
7	0111	1101	1010	1001	11100	1000100	0010000000
8	1000	1110	1011	1000	11110	1001000	0100000000
9	1001	1111	<u>1100</u>	1111	11111	1010000	1000000000

digit	8421 code	5421 code	2421 code	Excess 3 code	2 of 5 code
0	0000	0000	0000	0011	11000
1	0001	0001	0001	0100	10100
2	0010	0010	0010	0101	10010
3	0011	0011	0011	0110	10001
4	0100	0100	0100	0111	01100
5	0101	1000	1011	1000	01010
6	0110	1001	1100	1001	01001
7	0111	1010	1101	1010	00110
8	1000	1011	1110	1011	00101
9	1001	1100	1111	1100	00011

# 기출문제