

# 3 디지털 코드

# 학습목표

- 디지털 코드의 종류를 알아본다.
- 문자와 숫자를 나타내는 코드에 대해 알아본다.
- 숫자를 나타내는 코드인 가중치 코드와 비가중치 코드에 대해 알아본다.
- 에러 검출 코드의 종류에 대해 알아본다.

01. BCD 코드와 3초과 코드

02. 다양한 2진 코드들

03. 그레이 코드

04. 에러검출 코드

05. 영숫자 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 에러 검사를 위한 추가비트. 패리티는 단지 에러 검출만 가능하며, 여러 비트에 에러가 발생할 경우에는 검출이 안될 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
...	...	...

# 04 에러 검출 코드

## □ 병렬 패리티(parallel parity)

- ❖ 패리티를 블록 데이터에 적용해서 가로와 세로 데이터들에 대해서 패리티를 적용하면 에러를 검출하여 그 위치를 찾아 정정할 수 있다.

1	0	1	0	1	1	1	1	0
1	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	1
1	1	1	1	0	0	0	0	0
1	0	1	1	1	0	0	1	1
0	0	0	0	0	1	1	1	1
1	1	1	1	1	1	1	1	0
0	1	1	1	1	0	0	0	0
1	0	1	0	0	1	0	1	0

원래 데이터 블록

가로세로 모두 1의  
개수가 짝수임

1	0	1	0	1	1	1	1	0
1	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	1
1	1	1	1	0	0	0	0	0
1	0	1	1	0	0	0	1	1
0	0	0	0	0	1	1	1	1
1	1	1	1	1	1	1	1	0
0	1	1	1	1	0	0	0	0
1	0	1	0	0	1	0	1	0

에러가 발생한 블록

가로세로 회색 부분에  
1의 개수가 홀수임 : 겹  
치는 부분 에러

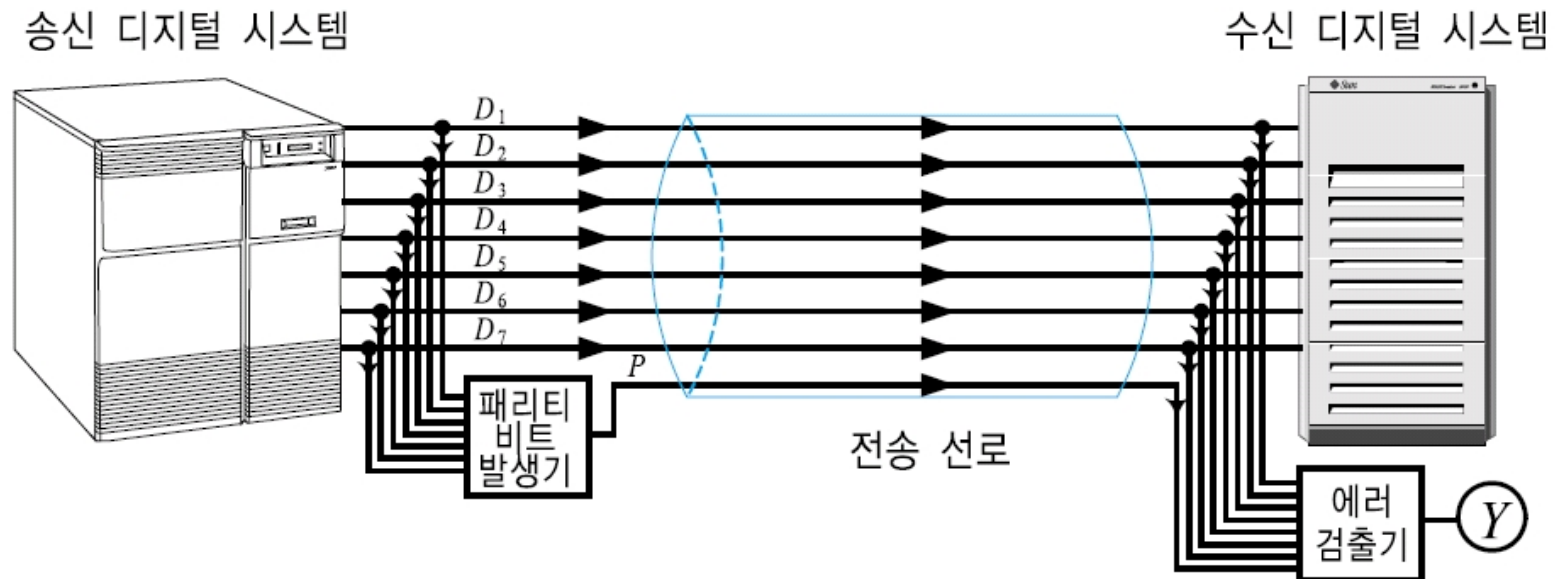
# 04 에러 검출 코드

## □ 데이터 전송 시스템에서 패리티 비트를 사용한 에러 검출

- ❖ 에러를 검출하기 위하여 송신측에 패리티 발생기를 구성하고 수신측에는 패리티 검출기를 구성하여 그 출력을 보고 에러 발생 여부를 판단

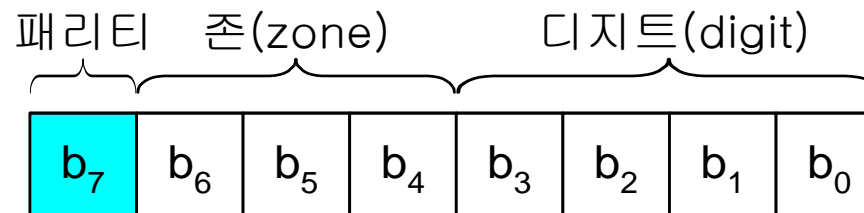
짝수 패리티  $Y=0$ (에러 없음),  $Y=1$ (에러 발생)

홀수 패리티  $Y=1$ (에러 없음),  $Y=0$ (에러 발생)



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

- ❖ 미국 국립 표준 연구소(ANSI)가 제정한 정보 교환용 미국 표준 코드
- ❖ 128가지의 문자를 표현 가능



### □ 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)			

# 05 영숫자 코드

## □ 표준 ASCII 코드표

	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	:	;		=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

# 05 영숫자 코드

## □ 확장 ASCII 코드표

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
9	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ç	£	¥	Pt	f
A	á	í	ó	ú	ñ	Ñ	ª	º	¿	«	»	½	¼	¿	«	»
B	⌘	⌘	⌘		└	┌	┐	┌	┐	┌	┐	┌	┐	┌	┐	┌
C	⌘	⌘	⌘		└	┌	┐	┌	┐	┌	┐	┌	┐	┌	┐	┌
D	⌘	⌘	⌘		└	┌	┐	┌	┐	┌	┐	┌	┐	┌	┐	┌
E	α	β	Γ	π	Σ	σ	μ	τ	Φ	Θ	Ω	δ	∞	∅	ε	∩
F	≡	±	≥	≤	∫	∫	÷	≈	◦	•	·	√	n	2	■	



## 4. 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 **버전 5.1.0.0(2008. 4. 4)까지 개발**
- ❖ 미국, 유럽, 동아시아, 아프리카, 아시아 태평양 지역 등의 주요 언어들에 적용될 수 있다.
- ❖ 유니코드는 유럽, 중동, 아시아 등 거의 대부분의 문자를 포함하고 있으며, 10만개 이상의 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, 덩벳 기호 등을 포함
- ❖ 앞으로도 계속해서 산업계의 요구나 새로운 문자들을 추가하여 나갈 것이다.

## 5. 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 방법