



3. 디지털 코드

논리회로

부경대 컴퓨터·인공지능공학부 최필주

목차

- 숫자 코드
 - BCD 코드와 3초과 코드
 - 다양한 2진 코드들
 - 그레이 코드
- 영숫자 코드
- 오류 검출 코드
- Summary



숫자 코드

BCD 코드와 3초과 코드

- BCD 코드(Binary Coded Decimal Code : 2진화 10진 코드, 8421코드)
 - 10진수 0(0000)~9(1001) → 2진화
 - 1010부터 1111까지 6개는 사용하지 않음

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

BCD 코드와 3초과 코드

- BCD 코드의 연산

10진 덧셈 (6+3=9)	10진 덧셈 (42+27=69)	(8+7=15)
$\begin{array}{r} 0110 \\ + 0011 \\ \hline 1001 \end{array}$	$\begin{array}{r} 0100\ 0010 \\ + 0010\ 0111 \\ \hline 0110\ 1001 \end{array}$	$\begin{array}{r} 1000 \\ + 0111 \\ \hline 1111 \\ + 0110 \\ \hline 0001\ 0101 \end{array}$ <div>6</div>

- 계산 결과가 BCD코드를 벗어나면(9(1001) 초과) + 6(0110)

BCD 코드와 3초과 코드

- BCD 코드의 연산 – 예시: $69 + 85$

10진 덧셈	BCD 코드의 덧셈
$\begin{array}{r} 69 \\ + 85 \\ \hline 154 \end{array}$	$\begin{array}{r} 0110\ 1001 \\ + 1000\ 0101 \\ \hline 1110\ 1110 \\ + 0110\ 0110 \\ \hline 0001\ 0101\ 0100 \end{array}$

BCD 코드와 3초과 코드

- BCD 코드의 특징

- 장점

- 10진수-BCD 변환 쉬움
 - 4 bits이 10진수 한 자리에 대응
 - 사용자에게 친숙, 입출력 하드웨어 구현 쉬움
 - 정확한 소수점 표현
 - 일부 계산기에서는 BCD 방식 사용

- 단점

- 2진수 표현에 비해 많은 bit 사용
 - 연산 복잡(각 자리에서 9를 넘어가면 +6 연산 필요)

BCD 코드와 3초과 코드

● 3초과 코드

- BCD코드(8421코드)로 표현된 값에 3을 더해 준 값으로 나타내는 코드
- 자기 보수의 성질

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

BCD 코드와 3초과 코드

● 3초과 코드의 특징

■ 장점

- bit를 반전하여 9의 보수를 얻을 수 있음
- 반올림이 쉬움(각 자리를 나타내는 4 bits 중 MSB에 따라 결정)
- 모든 비트가 0이 되는 경우가 없음
 - 단선 등의 신호 두절 구별 용이

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

다양한 2진 코드들 – 가중치 코드

- 8421 코드(BCD 코드)

10진수	8421코드	10진수 계산
0	0000	$8 \times 0 + 4 \times 0 + 2 \times 0 + 1 \times 0 = 0$
1	0001	$8 \times 0 + 4 \times 0 + 2 \times 0 + 1 \times 1 = 1$
2	0010	$8 \times 0 + 4 \times 0 + 2 \times 1 + 1 \times 0 = 2$
3	0011	$8 \times 0 + 4 \times 0 + 2 \times 1 + 1 \times 1 = 3$
4	0100	$8 \times 0 + 4 \times 1 + 2 \times 0 + 1 \times 0 = 4$
5	0101	$8 \times 0 + 4 \times 1 + 2 \times 0 + 1 \times 1 = 5$
6	0110	$8 \times 0 + 4 \times 1 + 2 \times 1 + 1 \times 0 = 6$
7	0111	$8 \times 0 + 4 \times 1 + 2 \times 1 + 1 \times 1 = 7$
8	1000	$8 \times 1 + 4 \times 0 + 2 \times 0 + 1 \times 0 = 8$
9	1001	$8 \times 1 + 4 \times 0 + 2 \times 0 + 1 \times 1 = 9$

- 자기보수 성질 없음

다양한 2진 코드들 – 가중치 코드

- 2421 코드

10진수	2421 코드	10진수 계산	2421 코드	10진수 계산
0	0000	$2 \times 0 + 4 \times 0 + 2 \times 0 + 1 \times 0 = 0$	0000	$2 \times 0 + 4 \times 0 + 2 \times 0 + 1 \times 0 = 0$
1	0001	$2 \times 0 + 4 \times 0 + 2 \times 0 + 1 \times 1 = 1$	0001	$2 \times 0 + 4 \times 0 + 2 \times 0 + 1 \times 1 = 1$
2	0010	$2 \times 0 + 4 \times 0 + 2 \times 1 + 1 \times 0 = 2$	1000	$2 \times 1 + 4 \times 0 + 2 \times 0 + 1 \times 0 = 2$
3	0011	$2 \times 0 + 4 \times 0 + 2 \times 1 + 1 \times 1 = 3$	1001	$2 \times 1 + 4 \times 0 + 2 \times 0 + 1 \times 1 = 3$
4	0100	$2 \times 0 + 4 \times 1 + 2 \times 0 + 1 \times 0 = 4$	1010	$2 \times 1 + 4 \times 0 + 2 \times 1 + 1 \times 0 = 4$
5	1011	$2 \times 1 + 4 \times 0 + 2 \times 1 + 1 \times 1 = 5$	0101	$2 \times 0 + 4 \times 1 + 2 \times 0 + 1 \times 1 = 5$
6	1100	$2 \times 1 + 4 \times 1 + 2 \times 0 + 1 \times 0 = 6$	0110	$2 \times 0 + 4 \times 1 + 2 \times 1 + 1 \times 0 = 6$
7	1101	$2 \times 1 + 4 \times 1 + 2 \times 0 + 1 \times 1 = 7$	0111	$2 \times 0 + 4 \times 1 + 2 \times 1 + 1 \times 1 = 7$
8	1110	$2 \times 1 + 4 \times 1 + 2 \times 1 + 1 \times 0 = 8$	1110	$2 \times 1 + 4 \times 1 + 2 \times 1 + 1 \times 0 = 8$
9	1111	$2 \times 1 + 4 \times 1 + 2 \times 1 + 1 \times 1 = 9$	1111	$2 \times 1 + 4 \times 1 + 2 \times 1 + 1 \times 1 = 9$

- 자기보수 성질을 가짐

다양한 2진 코드들 – 가중치 코드

- 5421 코드

10진수	5421 코드	10진수 계산	5421 코드	10진수 계산
0	0000	$5 \times 0 + 4 \times 0 + 2 \times 0 + 1 \times 0 = 0$	0000	$5 \times 0 + 4 \times 0 + 2 \times 0 + 1 \times 0 = 0$
1	0001	$5 \times 0 + 4 \times 0 + 2 \times 0 + 1 \times 1 = 1$	0001	$5 \times 0 + 4 \times 0 + 2 \times 0 + 1 \times 1 = 1$
2	0010	$5 \times 0 + 4 \times 0 + 2 \times 1 + 1 \times 0 = 2$	0010	$5 \times 0 + 4 \times 0 + 2 \times 1 + 1 \times 0 = 2$
3	0011	$5 \times 0 + 4 \times 0 + 2 \times 1 + 1 \times 1 = 3$	0011	$5 \times 0 + 4 \times 0 + 2 \times 1 + 1 \times 1 = 3$
4	0100	$5 \times 0 + 4 \times 1 + 2 \times 0 + 1 \times 0 = 4$	0100	$5 \times 0 + 4 \times 1 + 2 \times 0 + 1 \times 0 = 4$
5	1000	$5 \times 1 + 4 \times 0 + 2 \times 0 + 1 \times 0 = 5$	0101	$5 \times 0 + 4 \times 1 + 2 \times 0 + 1 \times 1 = 5$
6	1001	$5 \times 1 + 4 \times 0 + 2 \times 0 + 1 \times 1 = 6$	0110	$5 \times 0 + 4 \times 1 + 2 \times 1 + 1 \times 0 = 6$
7	1010	$5 \times 1 + 4 \times 0 + 2 \times 1 + 1 \times 0 = 7$	0111	$5 \times 0 + 4 \times 1 + 2 \times 1 + 1 \times 1 = 7$
8	1011	$5 \times 1 + 4 \times 0 + 2 \times 1 + 1 \times 1 = 8$	1011	$5 \times 1 + 4 \times 0 + 2 \times 1 + 1 \times 1 = 8$
9	1100	$5 \times 1 + 4 \times 1 + 2 \times 0 + 1 \times 0 = 9$	1100	$5 \times 1 + 4 \times 1 + 2 \times 0 + 1 \times 0 = 9$

- 자기보수 성질 없음

다양한 2진 코드들 – 가중치 코드

- 84-2-1 ($84\bar{2}\bar{1}$) 코드

10진수	84-2-1코드	10진수 계산
0	0000	$8 \times 0 + 4 \times 0 - 2 \times 0 - 1 \times 0 = 0$
1	0111	$8 \times 0 + 4 \times 1 - 2 \times 1 - 1 \times 1 = 1$
2	0110	$8 \times 0 + 4 \times 1 - 2 \times 1 - 1 \times 0 = 2$
3	0101	$8 \times 0 + 4 \times 1 - 2 \times 0 - 1 \times 1 = 3$
4	0100	$8 \times 0 + 4 \times 1 - 2 \times 0 - 1 \times 0 = 4$
5	1011	$8 \times 1 + 4 \times 0 - 2 \times 1 - 1 \times 1 = 5$
6	1010	$8 \times 1 + 4 \times 0 - 2 \times 1 - 1 \times 0 = 6$
7	1001	$8 \times 1 + 4 \times 0 - 2 \times 0 - 1 \times 1 = 7$
8	1000	$8 \times 1 + 4 \times 0 - 2 \times 0 - 1 \times 0 = 8$
9	1111	$8 \times 1 + 4 \times 1 - 2 \times 1 - 1 \times 1 = 9$

- 자기보수 성질을 가짐

다양한 2진 코드들 – 비가중치 코드

- 비가중치코드(non-weighted code)

- 각각의 위치에 해당하는 값이 없는 코드
- 데이터 변환과 같은 특수한 용도로 사용

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

다양한 2진 코드들 – 비가중치 코드

- 그레이 코드(Gray Code)

- 연산에는 부적당
- 연속되는 코드들 간에 하나의 비트만 차이남
- 아날로그-디지털 변환기나 입출력 장치 코드 등에 사용됨

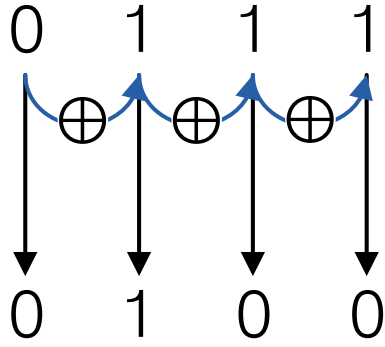
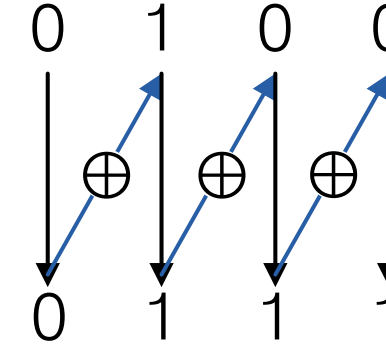
10진수	2진 코드	그레이 코드
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100

10진수	2진 코드	그레이 코드
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

이웃하는 코드간에
한 비트만 다르다.

다양한 2진 코드들 – 비가중치 코드

- 2진 코드 \leftrightarrow 그레이 코드

2진 코드 \rightarrow 그레이 코드		그레이 코드 \rightarrow 2진 코드	
2진 코드	0 1 1 1	그레이 코드	0 1 0 0
			
그레이 코드	0 1 0 0	2진 코드	0 1 1 1

<XOR 진리표>

입력		출력
A	B	$F=A\oplus B$
0	0	0
0	1	1
1	0	1
1	1	0



영숫자코드

ASCII 코드

- ASCII(American Standard Code for Information Interchange)
 - 미국 국립 표준 연구소(ANSI)가 제정한 정보 교환용 미국 표준 코드
 - 128가지의 문자를 표현 가능
 - ASCII 코드의 구성

zone bit			digit bit			
6	5	4	3	2	1	0
0	1	1	숫자 0~9(0000~1001)			
1	0	0	영문자 A~O(0001~1111)			
1	0	1	영문자 P~Z(0000~1010)			
1	1	0	영문자 a~o(0001~1111)			
1	1	1	영문자 p~z(0000~1010)			

ASCII 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	{		}	~	DEL

ASCII 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	■	

표준 BCD 코드

- 표준 BCD 코드

- 6비트로 하나의 문자를 표현
- 최대 64문자까지 표현 가능한 코드

zone bit		digit bit			
5	4	3	2	1	0
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		특수문자 및 기타문자			

표준 BCD 코드

● 표준 BCD 코드표

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

- EBCDIC(Extended Binary Coded Decimal Interchange Code)
 - IBM에서 개발
 - 대형 컴퓨터와 IBM 계열 컴퓨터에서 많이 사용
 - 256종류의 문자 코드 표현

zone bit				digit bit			
7	6	5	4	3	2	1	0
0	0			통신 제어 문자			
0	1			특수 문자			
1	0			소문자			
1	1	00 ~ 10		대문자			
		11		숫자			

EBCDIC

● 코드표

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

유니코드(Unicode)

- ASCII 코드의 한계 극복을 위해 개발된 인터넷 시대의 표준
 - 다양한 언어의 문자 포함
 - 10만개 이상의 문자
 - 유럽, 중동, 아시아 등 거의 대부분의 문자 포함
 - 구두표시, 수학기호, 전문기호, 기하학적 모양, 덩벙 기호 등을 포함
- 종류
 - 32(UTF-32), 16(UTF-16), 8bit(UTF-8)
 - UTF-16: 한 문자에 2~4바이트 사용, 영문/한글은 각 2바이트
 - UTF-8: 한 문자에 1~4바이트 사용, 영문/한글은 각 1/3바이트

한글코드

● 조합형

- MSB = 1: 한글을 의미
- 초성, 중성, 종성으로 나누어 표현
- 모든 한글 사용 가능
 - 고어까지 취감 가능
 - 다른 응용 프로그램에서는 표현 불가능할 수 있음

두 번째 바이트								첫 번째 바이트							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1	초성						중성				종성				

● 완성형

- 1987년 정부가 한국표준으로 정함
- 가장 많이 사용되는 한글 음절을 2 바이트의 2진수와 1 대 1로 대응
- 각 바이트의 MSB는 1: ASCII 코드와의 중복 방지

Summary

숫자코드

- 장점: 변환 쉬움, 실수의 정확한 표현 가능
- 단점: 더 많은 bit 필요, 연산 복잡

연산보다는 카운팅, 데이터 변환, 입출력 등에 주로 활용

10진수	2진코드	가중치코드				비가중치코드			
		8421	2421	5421	842 $\bar{1}$	3초과	5중 2	Shift counter	Gray
0	0000	0000	0000	0000	0000	0011	11000	00000	0000
1	0001	0001	0001	0001	0111	0100	00011	00001	0001
2	0010	0010	0010, 1000	0010	0110	0101	00101	00011	0011
3	0011	0011	0011, 1001	0011	0101	0110	00110	00111	0010
4	0100	0100	0100, 1010	0100	0100	0111	01001	01111	0110
5	0101	0101	0101, 1011	0101, 1000	1011	1000	01010	11111	0111
6	0110	0110	0110, 1100	0110, 1001	1010	1001	01100	11110	0101
7	0111	0111	0111, 1101	0111, 1010	1001	1010	10001	11100	0100
8	1000	1000	1110	1011	1000	1011	10010	11000	1100
9	1001	1001	1111	1100	1111	1100	10100	10000	1101
10	1010	0001 0000	0001 0000	0001 0000	0111 0000	0100 0011	<div data-bbox="1778 978 2242 1306"> <p>2진 코드</p> <p>그레이 코드</p> <p>2진 코드</p> </div>		1111
11	1011	0001 0001	0001 0001	0001 0001	0111 0111	0100 0100			1110
12	1100	0001 0010	0001 0010	0001 0010	0111 0110	0100 0101			1010
13	1101	0001 0011	0001 0011	0001 0011	0111 0101	0100 0110			1011
14	1110	0001 0100	0001 0100	0001 0100	0111 0100	0100 0111			1001
15	1111	0001 0101	0001 0101	0001 0101	0000 1011	0100 1000			1000
자기보수성질		X	O	X	O	O			

Summary

- 영숫자코드
 - 영문 + 숫자

	크기	최대 표현 가능 문자수	나타낼 수 있는 표현
ASCII	7~8 bits	128(표준), 256(확장)	대문자, 소문자, 숫자, 특수문자, 기타문자
표준 BCD	6 bits	64	대문자, 숫자, 특수문자, 기타문자
EBCDIC	8 bits	256	대문자, 소문자, 숫자, 특수문자, 기타문자, 통신제어문자

- 유니코드

- 다양한 나라의 문자와 구두/수학/전문 기호 등 포함
- 종류: UTF32, UTF16, UTF8

- 한글코드

	설명	특징	ASCII 문자와의 구분
조합형	초성, 중성, 종성으로 나누어 표현	고어까지 표현 가능	2바이트의 MSB가 1
완성형	많이 사용하는 한글을 숫자와 1대1 대응	1987 한글 표준	각 바이트의 MSB가 1



오류 검출 코드

패리티 비트

- 패리티 비트

- 데이터 전송과정에서 오류 검사를 위한 추가 비트
- 오류 검출만 가능
- 여러 비트에 오류 발생 시 검출이 불가능할 수 있음

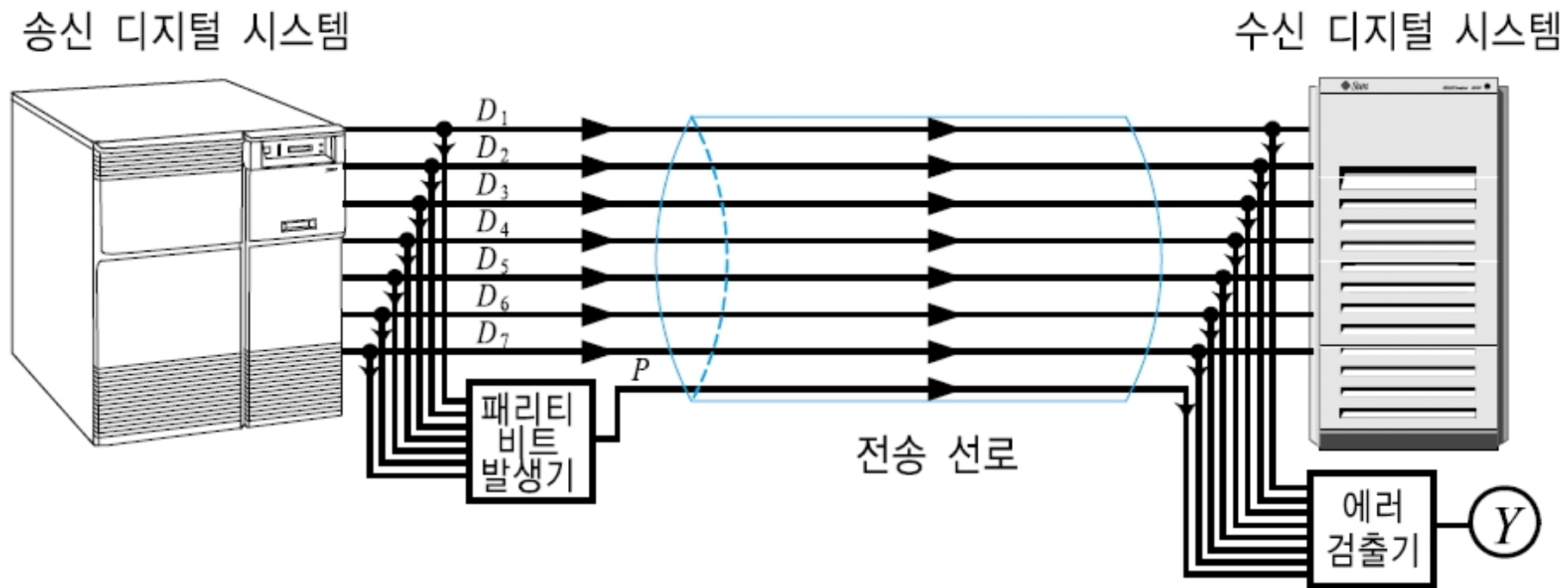
- 종류

- 짝수패리티(even parity) :
parity bit 포함하여 1의 개수가 짝수
- 홀수패리티(odd parity) :
parity bit 포함하여 1의 개수가 홀수

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

패리티 비트

- 패리티 비트의 활용 – 통신 오류 검출
 - 송신측: 패리티 발생기 구성
 - 수신측: 패리티 검출기 구성 → 오류 발생 여부 판단



패리티 비트

- 병렬 패리티(parallel parity)

- 블록 데이터의 가로/세로에 패리티 적용
- 오류를 검출하여 정정 가능

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

오류 발생

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

패리티 비트

- 병렬 패리티(parallel parity)

- ASCII 문자 N 개와 병렬 패리티를 전송할 때의 효율은?
- 패리티 비트를 포함한 총 비트 수: $8(N + 1)$
- 패리티 비트 제외한 총 비트 수: $7N$

- 효율: $\frac{7N}{8(N+1)}$

해밍코드(Hamming Code)

- 해밍코드 개요

- 오류를 정정할 수 있는 코드
- 추가적으로 많은 비트 필요 → 많은 양의 데이터 전달 필요
- 짝수 패리티 사용

- 데이터 비트와 패리티 비트와의 관계

- $2^{p-1} - p + 1 \leq d \leq 2^p - p - 1$
 - $p(\geq 2)$: 패리티 비트 수
 - d : 데이터 비트의 수
- 예: $p = 4$ 일 때
 - $2^{4-1} - 4 + 1 \leq d \leq 2^4 - 4 - 1 \rightarrow 5 \leq d \leq 11$
 - 데이터 비트가 5~11일 때 패리티 비트는 4개 필요

해밍코드(Hamming Code)

- 패리티 비트와 데이터 비트의 위치
 - 패리티 비트의 위치: $2^0, 2^1, 2^2, 2^3, 2^4, \dots$ 번째
 - 데이터 비트의 위치: 나머지 위치에 순서대로

비트 위치	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	10 1010	11 1011	12 1100	P_i 의 계산
기호	P_1	P_2	D_3	P_4	D_5	D_6	D_7	P_8	D_9	D_{10}	D_{11}	D_{12}	
P_1 영역	✓		✓		✓		✓		✓		✓		$D_3 \oplus D_5 \oplus D_7 \oplus D_9 \oplus D_{11}$
P_2 영역		✓	✓			✓	✓			✓	✓		$D_3 \oplus D_6 \oplus D_7 \oplus D_{10} \oplus D_{11}$
P_4 영역				✓	✓	✓	✓					✓	$D_5 \oplus D_6 \oplus D_7 \oplus D_{12}$
P_8 영역								✓	✓	✓	✓	✓	$D_9 \oplus D_{10} \oplus D_{11} \oplus D_{12}$

해밍코드(Hamming Code)

● 해밍코드의 생성

- 예: 데이터=00101110

P_i 의 계산

$$P_1 = D_3 \oplus D_5 \oplus D_7 \oplus D_9 \oplus D_{11}$$

$$P_2 = D_3 \oplus D_6 \oplus D_7 \oplus D_{10} \oplus D_{11}$$

$$P_4 = D_5 \oplus D_6 \oplus D_7 \oplus D_{12}$$

$$P_8 = D_9 \oplus D_{10} \oplus D_{11} \oplus D_{12}$$

비트 위치	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	10 1010	11 1011	12 1100
기호	P_1	P_2	D_3	P_4	D_5	D_6	D_7	P_8	D_9	D_{10}	D_{11}	D_{12}
P_1 영역	✓		✓		✓		✓		✓		✓	
P_2 영역		✓	✓			✓	✓			✓	✓	
P_4 영역				✓	✓	✓	✓					✓
P_8 영역								✓	✓	✓	✓	✓
원본 데이터			0		0	1	0		1	1	1	0
생성된 코드	0	1	0	1	0	1	0	1	1	1	1	0

해밍코드(Hamming Code)

오류의 정정

- 예: 해밍코드=010111011110

P'_i 의 계산

$$P'_1 = P_1 \oplus D_3 \oplus D_5 \oplus D_7 \oplus D_9 \oplus D_{11}$$

$$P'_2 = P_2 \oplus D_3 \oplus D_6 \oplus D_7 \oplus D_{10} \oplus D_{11}$$

$$P'_4 = P_4 \oplus D_5 \oplus D_6 \oplus D_7 \oplus D_{12}$$

$$P'_8 = P_8 \oplus D_9 \oplus D_{10} \oplus D_{11} \oplus D_{12}$$

비트 위치	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	10 1010	11 1011	12 1100
기호	P_1	P_2	D_3	P_4	D_5	D_6	D_7	P_8	D_9	D_{10}	D_{11}	D_{12}
P_1 영역	✓		✓		✓		✓		✓		✓	
P_2 영역		✓	✓			✓	✓			✓	✓	
P_4 영역				✓	✓	✓	✓					✓
P_8 영역								✓	✓	✓	✓	✓
해밍 코드	0	1	0	1	1	1	0	1	1	1	1	0
패리티 검사	1	0		1				0				

- 검출 시엔 패리티 비트를 포함하여 패리티 생성
- 수신된 해밍코드에 대해 P'_i 계산하면 $P'_8 P'_4 P'_2 P'_1$ 이 오류의 위치

Summary

- 오류 검출/정정 코드

- 패리티 비트 (오류 검출 코드)

- 홀수 개의 비트에서 오류가 발생하였는지 검출 가능
 - 짝수/홀수 패리티 비트: 패리티 비트를 포함하여 1의 개수가 짝수/홀수
 - 짝수 패리티 비트의 생성: 데이터 비트를 모두 XOR
 - 짝수 패리티 비트의 검출: 데이터 비트+패리티 비트를 모두 XOR
 - 1: 홀수개의 비트에서 오류 발생
 - 0: 짝수개의 비트에서 오류 발생 또는 오류 X

Summary

- 오류 검출/정정 코드

- 병렬 패리티 비트 (오류 정정 코드)

- 가로/세로 두 방향으로 패리티 비트 추가
 - 가로: 각 데이터마다 패리티 비트 추가
 - 세로: 패리티 비트로만 이루어진 값 추가
- 한 비트 오류에 대한 정정 가능
- 전송효율: k-bit 데이터를 N개씩 보낼 때 $\rightarrow \frac{kN}{(k+1)(N+1)}$

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

Summary

● 오류 검출/정정 코드

■ 해밍코드 (오류 정정 코드)

- 한 비트의 오류 정정이 가능한 코드
- 데이터 크기(d)와 패리티 비트 개수(p)의 관계: $2^{p-1} - p + 1 \leq d \leq 2^p - p - 1$
- 패리티 비트와 데이터 비트의 배치

비트 위치	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	10 1010	11 1011	12 1100	P_i 의 계산
기호	P_1	P_2	D_3	P_4	D_5	D_6	D_7	P_8	D_9	D_{10}	D_{11}	D_{12}	
P_1 영역	✓		✓		✓		✓		✓		✓		$D_3 \oplus D_5 \oplus D_7 \oplus D_9 \oplus D_{11}$
P_2 영역		✓	✓			✓	✓			✓	✓		$D_3 \oplus D_6 \oplus D_7 \oplus D_{10} \oplus D_{11}$
P_4 영역				✓	✓	✓	✓					✓	$D_5 \oplus D_6 \oplus D_7 \oplus D_{12}$
P_8 영역								✓	✓	✓	✓	✓	$D_9 \oplus D_{10} \oplus D_{11} \oplus D_{12}$

- P_i 의 계산: 비트 위치를 i 로 나눈 몫이 홀수인 부분의 데이터 bits를 XOR
- 오류 검출: 수신된 해밍코드에 대해 P_i 계산하면 $P_8P_4P_2P_1$ 이 오류의 위치