Lecture 2

Codes

Revised by WJ Han

Youpyo Hong, Dongguk University

BCD Code

- **A binary number is the most natural system for a computer, but people feel decimal system more convenient.**

- **Binary Coded Decimal (BCD) is to use 4-bit binary code for one decimal number as follows.**

Decimal symbol	BCD digit
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

$(5)_{10} = (0101)_{BCD}$

$(31)_{10} = (0011\ \ 0001)_{BCD}$

- **BCD code is called a weighted code.**

BCD Addition

- **If we add two BCD numbers like binary numbers, the result may be incorrect.**

$$\begin{array}{r} 4 \\ +\,5 \\ \hline 9 \end{array} \qquad \begin{array}{r} 0100 \\ +\,0101 \\ \hline 1001 \end{array}$$ **correct**

$$\begin{array}{r} 4 \\ +\,8 \\ \hline 12 \end{array} \qquad \begin{array}{r} 0100 \\ +\,1000 \\ \hline 1100 \end{array}$$ **incorrect**

- **1100 is not a BCD code. There was an overflow.**

- **Let's suppose we have another BCD digit. If you add 0110 to the incorrect result, you get the correct result with a carry. Why?**

$$\begin{array}{r} 4 \\ +\,8 \\ \hline 12 \end{array} \qquad \begin{array}{r} 0100 \\ +\,1000 \\ \hline \square\square\square\square\ 1100 \end{array}$$

$$\begin{array}{r} 0100 \\ +\,1000 \\ \hline 1100 \\ +\,0110 \\ \hline \mathbf{0001\ 0010} \end{array}$$ **correct**

Notes on BCD Addition

- Since each digit does not exceed 9, the sum cannot be greater than 9+9+1 = 19, with the 1 being a previous carry.
- When the binary sum is greater than 1001, the result is an invalid BCD digit. The addition of 6 = $(1001)_2$ to the binary sum converts it to the correct digit and also produces a carry as required.
- This is because a carry in the MSB position of the binary sum and a decimal carry deffer by 16 - 10 = 6.
- Consider the BCD addition 8 + 9 = 17 :
- Consider the addition of 184 + 576 = 760 in BCD :

Other Decimal Codes

Table 1.5
Four Different Binary Codes for the Decimal Digits

Decimal Digit	BCD 8421	2421	Excess-3	8, 4, −2, −1
0	0000	0000	0011	0000
1	0001	0001	0100	0111
2	0010	0010	0101	0110
3	0011	0011	0110	0101
4	0100	0100	0111	0100
5	0101	1011	1000	1011
6	0110	1100	1001	1010
7	0111	1101	1010	1001
8	1000	1110	1011	1000
9	1001	1111	1100	1111
Unused bit combi-nations	1010	0101	0000	0001
	1011	0110	0001	0010
	1100	0111	0010	0011
	1101	1000	1101	1100
	1110	1001	1110	1101
	1111	1010	1111	1110

Excess-3 Code

- **For excess-3 code (XS-3), add 3 to the BCD code.**

  Ex) XS-3 code for 24 ?

- **XS-3 code is called self-complementing code.**

Decimal Digit	BCD	Excess-3
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

Decimal Digit
9's complement

0	---	9
1	---	8

XS-3 code
1's complement

0011	1100
0100	1011

- **XS-3 code is the only unweighted code which is self-complementing.**

Self-complementing codes

- How to check for a code to be self-complementing in the weighted codes?

  w3 w2 w1 w0 (4 weights)

  w3 + w2 + w1 + w0 = 9 : self-complementing

  w3 + w2 + w1 + w0 != 9 : non self-complementing

  Ex) 2421 code -> self-comp. because of 2+4+2+1 = 9

  BCD code -> non self-comp. because of 8+4+2+1 = 15

- How about 5211 code ?

Gray Code

- **We call it Gray code after Frank Gray.**
- **Gray code is a code for which only one bit changes between each pair of successive codes.**

Decimal symbol	Gray Code
0	000
1	001
2	011
3	010
4	110
5	111
6	101
7	100

- **Gray code is called unit distance code or cyclic.**

cf. Hamilton path

Binary to Gray Code Conversion

- **How to derive a gray code?**

  Step1. Record the MSB as it is.

  Step2. Add the MSB to the next bit,
  record the sum and neglect the carry.

  Step3. Repeat the step2 for the next bit.

- **For Example,**

Binary	**1011**	**1 (MSB)**	**+ 0**	**+ 1**	**+ 1**
Gray code	**1110**	**1**	**= 1**	**= 1**	**= 0 (neglect the carry)**

Alphanumeric Codes

- **Many computers need to handle letters in addition to numbers.**
- **But, computers can recognize binary numbers only. What can we do?**
- **We can assign a code to each letter.**

  Ex) Use BCD for numbers and interpret 1010 as A, 1011 as B, etc.

- **There are alphanumeric codes like ASCII codes and EBCDIC codes.**

ASCII Codes

- ASCII stands for American Standard Code for Information Interchange.
- The standard binary code for the alphanumeric characters.
- It uses seven bits to code 128 characters including numbers.

$B_6B_5B_4B_3B_2B_1B_0$	ASCII
0 0 0 0 0 0 0	NULL
⋮	
0 1 1 0 0 0 0	0
0 1 1 0 0 0 1	1
0 1 1 0 0 1 0	2
⋮	

$B_6B_5B_4B_3B_2B_1B_0$	ASCII
1 0 0 0 0 0 1	A
1 0 0 0 0 1 0	B
⋮	
1 0 1 1 1 1 1	-
⋮	

ASCII Codes

$b_4b_3b_2b_1$	$b_7b_6b_5$ 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

Extra Bit in ASCII Codes

- **Most computers use 8-bits as a basic data unit called a byte.**

- **When we use ASCII code, the extra 1-bit is used for many other purposes, e.g. parity bit.**

Parity Bit for Error Detection

- **There can be an error during transmission or storage.**
- **There are many ways to detect if data is modified from its original value.**
- **Parity bit is a widely used error detection technique.**

	Even Parity Bit Added (make # of 1s even)	Odd Parity Bit Added
ASCII A = 1000001	01000001	11000001