Course Title:
Digital Logic Design

Shahriar Rahman Khan
Lecturer
Dept. of CSE, MIST

Course Code: CSE 103
Credit Hr: 3.00
Contact Hr: 3.00

Overview

- **Signed Binary Numbers**
- **Binary Codes**
- **Decimal Codes**
- **Gray Codes**
- **Parity Bit**
- **Alphanumeric Codes**

Signed Binary Numbers

Digital systems (computers) should be able to handle both positive and negative numbers. A signed binary number consists of both sign and magnitude information. Sign represents (+) ve or (-) ve and magnitude represents value. 1 means (-) ve and 0 means (+) ve.

For example, the positive number 58 is written using 8-bits as 00111010 (true form).

Sign bit

Magnitude bits

Signed integer can be represented in 3 ways:

i. Signed Magnitude
ii. 1's Complement
iii. 2's Complement

Signed Binary Numbers

Negative numbers are written as the 2's complement of the corresponding positive number.

The negative number −58 is written as:

−58 = 11000110 (complement form)

Sign bit   Magnitude bits

An easy way to read a signed number that uses this notation is to assign the sign bit a column weight of −128 (for an 8-bit number). Then add the column weights for the 1's.

Assuming that the sign bit = −128, show that 11000110 = −58 as a 2's complement signed number:

Column weights:	−128	64	32	16	8	4	2	1.	
	1	1	0	0	0	1	1	0	
	−128	+64				+4	+2		= −58

Signed Numbers

- Signed Magnitude form

  The +25 is written as: +25 = 00011001

  The -25 is written as: −25 = 10011001

- 1's Complement form

  The +25 is written as: +25 = 00011001

  The -25 is written as: −25 = 11100110

- 2's Complement form

  The +25 is written as: +25 = 00011001

  The -25 is written as: −25 = 11100111

+25 is same in 3 cases.
For 2's complement signed numbers, range of values are from
-128 to +127. n bit : range= -(2^(n-1)) to +(2^(n-1) – 1)

Arithmetic Operations With Signed Numbers

We saw, signed numbers can be represented in 3 ways, but we will 2's complement arithmetic operations as this is widely used.

Rules for **addition**: Add the two signed numbers. Discard any final carries. The result is in signed form.

Examples:

00011110 = +30	00001110 = +14	11111111 = −1
00001111 = +15	11101111 = −17	11111000 = −8
00101101 = +45	11111101 = −3	~~1~~11110111 = −9

Result is in 2's comp, so need to do 2's comp again to get actual value. -(00000011)=-3

Discard carry

Signed Binary Numbers

Note that if the number of bits required for the answer is exceeded, overflow will occur. This occurs only if both numbers have the same sign. The overflow will be indicated by an incorrect sign bit.

Two examples are:

01000000 = +128	10000001 = −127
01000001 = +129	10000001 = −127
10000001 = ~~−126~~	Discard carry → ~~1~~00000010 = ~~+2~~

Wrong! The answer is incorrect and the sign bit has changed.

Signed Binary Numbers

Rules for **subtraction**: 2's complement the subtrahend and add the numbers. Discard any final carries. The result is in signed form.

Repeat the examples done previously, but subtract:

$$\begin{array}{rl} 00011110 & (+30) \\ -\ 00001111 & -(+15) \\ \hline \end{array} \qquad \begin{array}{rl} 00001110 & (+14) \\ -\ 11101111 & -(-17) \\ \hline \end{array} \qquad \begin{array}{rl} 11111111 & (-1) \\ -\ 11111000 & -(-8) \\ \hline \end{array}$$

2's complement subtrahend and add:

$$\begin{array}{rl} 00011110 & = +30 \\ 11110001 & = -15 \\ \hline \cancel{1}00001111 & = +15 \end{array} \qquad \begin{array}{rl} 00001110 & = +14 \\ 00010001 & = +17 \\ \hline 00011111 & = +31 \end{array} \qquad \begin{array}{rl} 11111111 & = -1 \\ 00001000 & = +8 \\ \hline \cancel{1}00000111 & = +7 \end{array}$$

Discard carry

Discard carry

Self Study

Binary Numbers and Binary Coding

- Flexibility of representation
  - Everything represented in 0 or 1 in a digital system.
  - Digital Systems represent and manipulate not only binary numbers, but also many other discrete elements of information.
  - Most of the bits in a computer are coded information rather than binary numbers.

- Information Types
  - Numeric
  - Non-numeric

Non-numeric Binary Codes

- Given *n* binary digits (called <u>bits</u>), a <u>binary code</u> is a mapping from a set of <u>represented elements</u> to a subset of the 2^n binary numbers.
- Example: A binary code for the seven colors of the rainbow
- Code 100 is not used

Color	Binary Number
Red	**000**
Orange	**001**
Yellow	**010**
Green	**011**
Blue	**101**
Indigo	**110**
Violet	**111**

DECIMAL CODES - Binary Codes for Decimal Digits

Binary codes for decimal digits require a minimum of four bits. Numerous different codes can be obtained by arranging four or more bits in ten 10 distinct possible combinations.

A few possibilities are:

Decimal	8,4,2,1	Excess-3	8,4,-2,-1	Gray
0	0000	0011	0000	0000
1	0001	0100	0111	0001
2	0010	0101	0110	0011
3	0011	0110	0101	0010
4	0100	0111	0100	0110
5	0101	1000	1011	0111
6	0110	1001	1010	0101
7	0111	1010	1001	0100
8	1000	1011	1000	1100
9	1001	1100	1111	1101

 # Conversion and Coding are completely different?

Binary Coded Decimal (BCD)

- The BCD code is the 8,4,2,1 code.
- 8, 4, 2, and 1 are weights
  - BCD is a *weighted* code
- This code is the simplest, most intuitive binary code for decimal digits and uses the same powers of 2 as a binary number, but only encodes the first ten values from 0 to 9.
- Example: 58 = 0101 1000
- How many “invalid” code words are there?
- What are the “invalid” code words?

GRAY CODE – Decimal

- Gray code is an unweighted code that has a single bit change between one code word and the next in a sequence. Gray code is used to avoid problems in systems where an error can occur if more than one bit changes at a time.

GRAY CODE – Decimal

Decimal	8,4,2,1	Gray
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

- **What special property does the Gray code have in relation to adjacent decimal digits?**

Binary to Gray Code Conversion

- Keep the MSB unchanged.
- XOR MSB with next significant bit and record this.
- Repeat the process.

Truth Table XOR Gate

X	Y	Output
0	0	0
0	1	1
1	0	1
1	1	0

- Let 0= 0 0 0 0 – BCD
- XOR 0 0 0

  0 0 0 0 – Gray
- Let 1= 0 0 0 1 – BCD
- XOR 0 0 0

  0 0 0 1 - Gray

Four-bit Reflected Code

Decimal Equivalent	Reflected Code
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100
9	1101
10	1111
11	1110
12	1010
13	1011
14	1001
15	1000

Error-Detection Codes

- External noise introduce into a physical communication medium changes bit values from 0 to 1 or vice versa.
  - EDC can be used to detect errors during transmission.
  - Extra bit included with a message[called parity bit] to make up the total number of 1's either odd or even.
- A message of four bits and a parity bit, P, are shown in the table:
- 01111 -> 00101

Message	P (odd)	Message	P (even)
0000	1	0000	0
0001	0	0001	1
0010	0	0010	1
0011	1	0011	0
0100	0	0100	1
0101	1	0101	0
0110	1	0110	0
0111	0	0111	1
1000	0	1000	1
1001	1	1001	0
1010	1	1010	0
1011	0	1011	1
1100	1	1100	0
1101	0	1101	1
1110	0	1110	1
1111	1	1111	0

4-Bit Parity Code Example

- Fill in the even and odd parity bits:

Even Parity Message - Parity	Odd Parity Message - Parity
000 _	000 _
001 _	001 _
010 _	010 _
011 _	011 _
100 _	100 _
101 _	101 _
110 _	110 _
111 _	111 _

- The codeword "1111" has <u>even parity</u> and the codeword "1110" has <u>odd parity</u>. Both can be used to represent 3-bit data.

Warning: Conversion or Coding?

- Do NOT mix up conversion of a decimal number to a binary number with coding a decimal number with a BINARY CODE.
- $13_{10} = 1101_2$ (This is conversion)
- 13 $\Leftrightarrow$ 0001|0011 (This is coding)

Alphanumeric Codes

- Alphanumeric code is a binary code of a group of elements consisting of 10 decimal digits, 26 letters of the alphabet, and a certain number of special symbols such as $.
- Total number of element in an alphanumeric group is greater than 36. Therefore, it must be coded with a minimum of six bits ($2^6 = 64$, but $2^5 = 32$ is in sufficient) [6-Bit internal code]
- One such code is known as ASCII (American Standard code for Information Interchange); another is known as EBCDIC (Extended BCD Interchange Code)

ASCII Code

	$B_7B_6B_5$							
$B_4B_3B_2B_1$	000	001	010	011	100	101	110	111
0000	NULL	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


GOOD NEWS!
THE CLASS IS OVER…
THANK YOU!
AFSALCMK