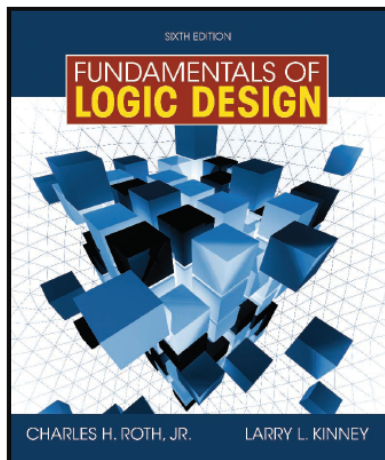


# SLIDES FOR CHAPTER 1

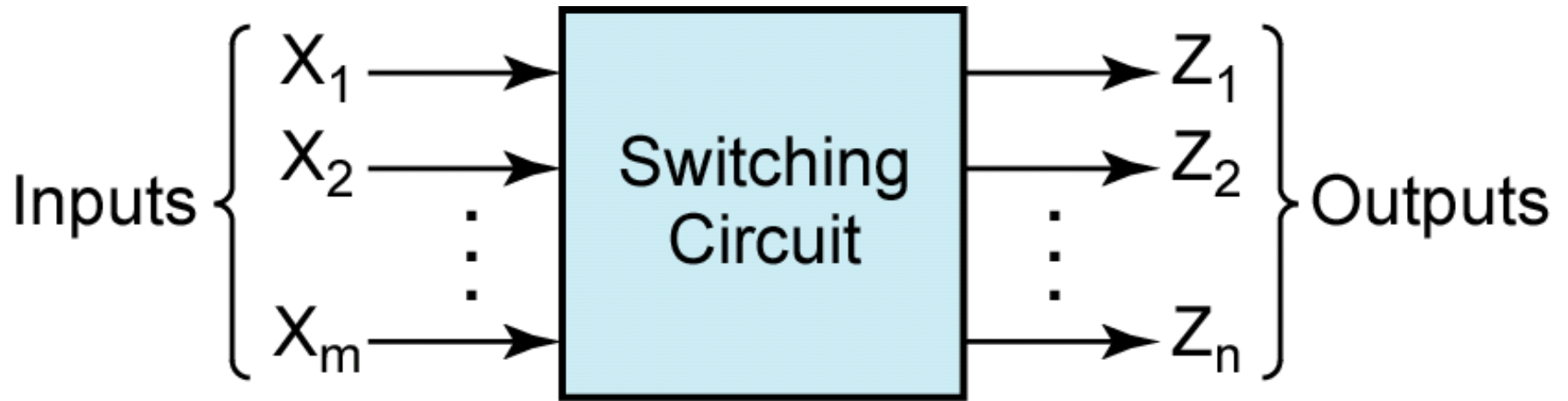
## INTRODUCTION NUMBER SYSTEMS AND CONVERSION



*This chapter in the book includes:*

- Objectives
- Study Guide
- 1.1 Digital Systems and Switching Circuits
- 1.2 Number Systems and Conversion
- 1.3 Binary Arithmetic
- 1.4 Representation of Negative Numbers
- 1.5 Binary Codes
- Problems

Click the mouse to move to the next page.  
Use the ESC key to exit this chapter.



**Figure 1-1: Switching circuit**

## Decimal Notation

$$953.78_{10} = 9 \times 10^2 + 5 \times 10^1 + 3 \times 10^0 + 7 \times 10^{-1} + 8 \times 10^{-2}$$

## Binary

$$\begin{aligned} 1011.11_2 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} \\ &= 8 + 0 + 2 + 1 + \frac{1}{2} + \frac{1}{4} \\ &= 11.75_{10} \end{aligned}$$

**EXAMPLE:** Convert  $53_{10}$  to binary.

$$2 \overline{)53}$$

$$2 \overline{)26} \quad \text{rem.} = 1 = a_0$$

$$2 \overline{)13} \quad \text{rem.} = 0 = a_1$$

$$2 \overline{)6} \quad \text{rem.} = 1 = a_2 \quad 53_{10} = 110101_2$$

$$2 \overline{)3} \quad \text{rem.} = 0 = a_3$$

$$2 \overline{)1} \quad \text{rem.} = 1 = a_4$$

$$0 \quad \text{rem.} = 1 = a_5$$

**Conversion (a)**

# EXAMPLE: Convert $.625_{10}$ to binary.

$$\begin{array}{r} F = .625 \\ \times 2 \\ \hline 1.250 \\ (a_{-1} = 1) \end{array} \qquad \begin{array}{r} F_1 = .250 \\ \times 2 \\ \hline 0.500 \\ (a_{-2} = 0) \end{array}$$

$$\begin{array}{r} F_2 = .500 \\ \times 2 \\ \hline 1.000 \\ (a_{-3} = 1) \end{array} \qquad .625_{10} = .101_2$$

**Conversion (b)**

**EXAMPLE:** Convert  $0.7_{10}$  to binary.

$$\begin{array}{r}
 .7 \\
 \underline{\phantom{.}2} \\
 (1).4 \\
 \underline{\phantom{.}2} \\
 (0).8 \\
 \underline{\phantom{.}2} \\
 (1).6 \\
 \underline{\phantom{.}2} \\
 (1).2 \\
 \underline{\phantom{.}2} \\
 (0).4 \\
 \underline{\phantom{.}2} \\
 (0).8
 \end{array}$$

← process starts repeating here because 0.4 was previously obtained

$$0.7_{10} = 0.1 \underline{0110} \underline{0110} \underline{0110} \dots_2$$

**Conversion (c)**

**EXAMPLE:** Convert  $231.3_4$  to base 7.

$$231.3_4 = 2 \times 16 + 3 \times 4 + 1 + \frac{3}{4} = 45.75_{10}$$

$$7 \overline{)45}$$

$$7 \overline{)6}$$

0

rem. 3

rem. 6

.75

7

(5) .25

7

(1) .75

7

(5) .25

7

(1) .75

$$45.75_{10} = 63.5151 \dots_7$$

**Conversion (d)**

# Binary ⇔ Hexadecimal Conversion

$$1001101.010111_2 = \underbrace{0100}_4 \underbrace{1101}_D . \underbrace{0101}_5 \underbrace{1100}_C = 4D.5C_{16}$$

## Equation (1-1)

Conversion from binary to hexadecimal (and conversely) can be done by inspection because each hexadecimal digit corresponds to exactly four binary digits (bits).

Add  $13_{10}$  and  $11_{10}$  in binary.

$$\begin{array}{r} 1111 \leftarrow \text{carries} \\ 13_{10} = 1101 \\ 11_{10} = \underline{1011} \\ \hline 11000 = 24_{10} \end{array}$$

**Addition**

The subtraction table for binary numbers is

$$0 - 0 = 0$$

$$0 - 1 = 1 \quad \text{and borrow 1 from the next column}$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

Borrowing 1 from a column  
is equivalent to subtracting 1 from that column.

## **Subtraction (a)**

## EXAMPLES OF BINARY SUBTRACTION:

(a)  $1 \leftarrow$  (indicates a borrow from the 3rd column)

$$\begin{array}{r} 11101 \\ -10011 \\ \hline 1010 \end{array}$$

(b)  $1111 \leftarrow$  borrows

$$\begin{array}{r} 11111 \\ -10000 \\ \hline 1101 \end{array}$$

(c)  $111 \leftarrow$  borrows

$$\begin{array}{r} 111001 \\ -1011 \\ \hline 101110 \end{array}$$

### Subtraction (b)

A detailed analysis of the borrowing process for this example, indicating first a borrow of 1 from column 1 and then a borrow of 1 from column 2, is as follows:

$$\begin{aligned}
 205 - 18 &= [2 \times 10^2 + 0 \times 10^1 + 5 \times 10^0] \\
 &- [ \qquad \qquad \qquad 1 \times 10^1 + 8 \times 10^0] \\
 &= [2 \times 10^2 + (0 - 1) \times 10^1 + (10 + 5) \times 10^0] \quad \text{note borrow from column 1} \\
 &- [ \qquad \qquad \qquad 1 \times 10^1 + \qquad \qquad \qquad 8 \times 10^0] \\
 &= [(2 - 1) \times 10^2 + (10 + 0 - 1) \times 10^1 + 15 \times 10^0] \quad \text{note borrow from column 2} \\
 &- [ \qquad \qquad \qquad 1 \times 10^1 + 8 \times 10^0] \\
 &= [1 \times 10^2 \qquad \qquad + 8 \times 10^1 \qquad \qquad + 7 \times 10^0] = 187
 \end{aligned}$$

## Subtraction (c)

The multiplication table for binary numbers is

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

**Multiplication (a)**

The following example illustrates multiplication of  $13_{10}$  by  $11_{10}$  in binary:

$$\begin{array}{r} 1101 \\ 1011 \\ \hline 1101 \\ 1101 \\ 0000 \\ 1101 \\ \hline 10001111 = 143_{10} \end{array}$$

**Multiplication (b)**

When doing binary multiplication, a common way to avoid carries greater than 1 is to add in the partial products one at a time as illustrated by the following example:

1111	multiplicand
<u>1101</u>	multiplier
1111	1st partial product
<u>0000</u>	2nd partial product
(01111)	sum of first two partial products
<u>1111</u>	3rd partial product
(1001011)	sum after adding 3rd partial product
<u>1111</u>	4th partial product
11000011	final product (sum after adding 4th partial product)

## Multiplication (c)

# Binary Division

Binary division is similar to decimal division, except it is much easier because the only two possible quotient digits are 0 and 1.

We start division by comparing the divisor with the upper bits of the dividend.

If we cannot subtract without getting a negative result, we move one place to the right and try again.

If we can subtract, we place a 1 for the quotient above the number we subtracted from and append the next dividend bit to the end of the difference and repeat this process with this modified difference until we run out of bits in the dividend.

The following example illustrates division of  $145_{10}$  by  $11_{10}$  in binary:

$$\begin{array}{r}
 \phantom{1011} \overline{1101} \\
 1011 \overline{)10010001} \\
 \underline{1011} \phantom{0000} \\
 1110 \phantom{000} \\
 \underline{1011} \phantom{00} \\
 1101 \phantom{0} \\
 \underline{1011} \\
 10
 \end{array}$$

The quotient is 1101 with a remainder of 10.

## Binary Division

### 3 Systems for representing negative numbers in binary

**Sign & Magnitude:** Most significant bit is the sign

Ex:  $-5_{10} = 1101_2$

**1's Complement:**  $\bar{N} = (2^n - 1) - N$

Ex:  $-5_{10} = (2^4 - 1) - 5 = 16 - 1 - 5 = 10_{10} = 1010_2$

**2's Complement:**  $N^* = 2^n - N$

Ex:  $-5_{10} = 2^4 - 5 = 16 - 5 = 11_{10} = 1011_2$

**Section 1.4 (p. 16)**

**Table 1-1: Signed Binary Integers (word length  $n = 4$ )**

$+N$	Positive Integers (all systems)	$-N$	Negative Integers		
			Sign and Magnitude	2's Complement $N^*$	1's Complement $\bar{N}$
+0	0000	-0	1000	—	1111
+1	0001	-1	1001	1111	1110
+2	0010	-2	1010	1110	1101
+3	0011	-3	1011	1101	1100
+4	0100	-4	1100	1100	1011
+5	0101	-5	1101	1011	1010
+6	0110	-6	1110	1010	1001
+7	0111	-7	1111	1001	1000
		-8	—	1000	—

1. Addition of two positive numbers,  $\text{sum} < 2^{n-1}$

$$\begin{array}{r} +3 \quad 0011 \\ +4 \quad 0100 \\ \hline +7 \quad 0111 \end{array} \quad (\text{correct answer})$$

2. Addition of two positive numbers,  $\text{sum} \geq 2^{n-1}$

$$\begin{array}{r} +5 \quad 0101 \\ +6 \quad 0110 \\ \hline 1011 \end{array} \quad \leftarrow \text{wrong answer because of overflow (+11 requires 5 bits including sign)}$$

## 2's Complement Addition (a)

3. Addition of positive and negative numbers (negative number has greater magnitude)

$$\begin{array}{r} +5 \quad 0101 \\ -6 \quad 1010 \\ \hline -1 \quad 1111 \end{array} \quad (\text{correct answer})$$

4. Same as case 3 except positive number has greater magnitude

$$\begin{array}{r} -5 \quad 1011 \\ +6 \quad 0110 \\ \hline +1 \quad (1)0001 \end{array} \quad \leftarrow \text{correct answer when the carry from the sign bit is ignored (this is *not* an overflow)}$$

## 2's Complement Addition (b)

5. Addition of two negative numbers,  $|\text{sum}| \leq 2^{n-1}$

$$\begin{array}{r} -3 \quad 1101 \\ -4 \quad 1100 \\ \hline -7 \quad (1)1001 \end{array}$$

← correct answer when the last carry is ignored  
(this is *not* an overflow)

6. Addition of two negative numbers,  $|\text{sum}| > 2^{n-1}$

$$\begin{array}{r} -5 \quad 1011 \\ -6 \quad 1010 \\ \hline (1)0101 \end{array}$$

← wrong answer because of overflow  
(-11 requires 5 bits including sign)

## 2's Complement Addition (c)

3. Addition of positive and negative numbers (negative number with greater magnitude)

$$\begin{array}{r}
 +5 \quad 0101 \\
 -6 \quad 1001 \\
 \hline
 -1 \quad 1110
 \end{array}
 \quad \text{(correct answer)}$$

4. Same as case 3 except positive number has greater magnitude

$$\begin{array}{r}
 -5 \quad 1010 \\
 +6 \quad 0110 \\
 \hline
 (1) \ 0000 \\
 \quad \longleftarrow 1 \\
 \quad \quad \hline
 \quad \quad 0001
 \end{array}
 \quad \begin{array}{l}
 \text{(end-around carry)} \\
 \text{(correct answer, no overflow)}
 \end{array}$$

## 1's Complement Addition (b)

5. Addition of two negative numbers,  $|\text{sum}| < 2^{n-1}$

$$\begin{array}{r}
 -3 \quad 1100 \\
 -4 \quad 1011 \\
 \hline
 (1) \quad 0111 \\
 \quad \quad \longleftarrow 1 \quad \text{(end-around carry)} \\
 \quad \quad \hline
 \quad \quad 1000 \quad \text{(correct answer, no overflow)}
 \end{array}$$

6. Addition of two negative numbers,  $|\text{sum}| \geq 2^{n-1}$

$$\begin{array}{r}
 -5 \quad 1010 \\
 -6 \quad 1001 \\
 \hline
 (1) \quad 0011 \\
 \quad \quad \longleftarrow 1 \quad \text{(end-around carry)} \\
 \quad \quad \hline
 \quad \quad 0100 \quad \text{(wrong answer because of overflow)}
 \end{array}$$

## 1's Complement Addition (c)

1. Add  $-11$  and  $-20$  in 1's complement.

$$+11 = 00001011 \qquad +20 = 00010100$$

taking the bit-by-bit complement,

$-11$  is represented by  $11110100$  and  $-20$  by  $11101011$

$$\begin{array}{r} 11110100 \quad (-11) \\ \underline{11101011} \quad +(-20) \\ (1) 11011111 \\ \quad \longleftarrow 1 \quad (\text{end-around carry}) \\ \underline{\hspace{1.5em}} \\ 11100000 = -31 \end{array}$$

## 1's Complement Addition (d)

2. Add  $-8$  and  $+19$  in 2's complement

$$+ 8 = 00001000$$

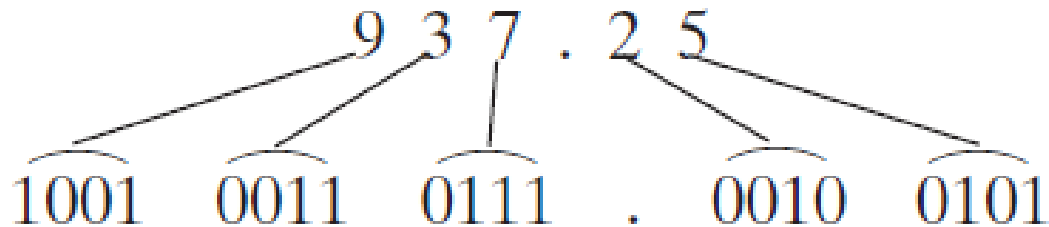
complementing all bits to the left of the first 1,  $-8$ , is represented by 11111000

$$\begin{array}{r} 11111000 \quad (-8) \\ \underline{00010011 \quad +19} \\ \text{~~1~~00001011 = +11 \\ \uparrow \text{ (discard last carry)} \end{array}$$

## 2's Complement Addition (d)

# Binary Codes

Although most large computers work internally with binary numbers, the input-output equipment generally uses decimal numbers. Because most logic circuits only accept two-valued signals, the decimal numbers must be coded in terms of binary signals. In the simplest form of binary code, each decimal digit is replaced by its binary equivalent. For example, 937.25 is represented by:



**Section 1.5 (p. 21)**

## Table 1–2. Binary Codes for Decimal Digits

Decimal Digit	8-4-2-1 Code (BCD)	6-3-1-1 Code	Excess-3 Code	2-out-of-5 Code	Gray Code
0	0000	0000	0011	00011	0000
1	0001	0001	0100	00101	0001
2	0010	0011	0101	00110	0011
3	0011	0100	0110	01001	0010
4	0100	0101	0111	01010	0110
5	0101	0111	1000	01100	1110
6	0110	1000	1001	10001	1010
7	0111	1001	1010	10010	1011
8	1000	1011	1011	10100	1001
9	1001	1100	1100	11000	1000

Character	ASCII Code							Character	ASCII Code						
	A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>		A <sub>6</sub>	A <sub>5</sub>	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>
space	0	1	0	0	0	0	0	@	1	0	0	0	0	0	0
!	0	1	0	0	0	0	1	A	1	0	0	0	0	0	1
"	0	1	0	0	0	1	0	B	1	0	0	0	0	1	0
#	0	1	0	0	0	1	1	C	1	0	0	0	0	1	1
\$	0	1	0	0	1	0	0	D	1	0	0	0	1	0	0
%	0	1	0	0	1	0	1	E	1	0	0	0	1	0	1
&	0	1	0	0	1	1	0	F	1	0	0	0	1	1	0
'	0	1	0	0	1	1	1	G	1	0	0	0	1	1	1
(	0	1	0	1	0	0	0	H	1	0	0	1	0	0	0
)	0	1	0	1	0	0	1	I	1	0	0	1	0	0	1
*	0	1	0	1	0	1	0	J	1	0	0	1	0	1	0
+	0	1	0	1	0	1	1	K	1	0	0	1	0	1	1
,	0	1	0	1	1	0	0	L	1	0	0	1	1	0	0
-	0	1	0	1	1	0	1	M	1	0	0	1	1	0	1
.	0	1	0	1	1	1	0	N	1	0	0	1	1	1	0
/	0	1	0	1	1	1	1	O	1	0	0	1	1	1	1
0	0	1	1	0	0	0	0	P	1	0	1	0	0	0	0
1	0	1	1	0	0	0	1	Q	1	0	1	0	0	0	1
2	0	1	1	0	0	1	0	R	1	0	1	0	0	1	0
3	0	1	1	0	0	1	1	S	1	0	1	0	0	1	1
4	0	1	1	0	1	0	0	T	1	0	1	0	1	0	0

**Table 1-3  
ASCII code  
(incomplete)**