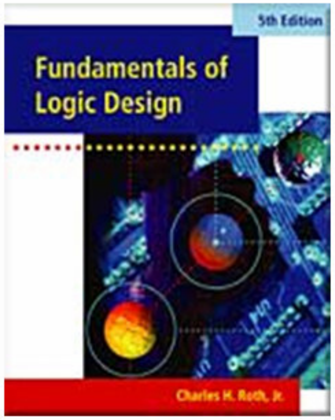
CHAPTER 1

INTRODUCTION
NUMBER SYSTEMS AND CONVERSION



This chapter in the book includes:

OIT EET 283 Steen Larsen

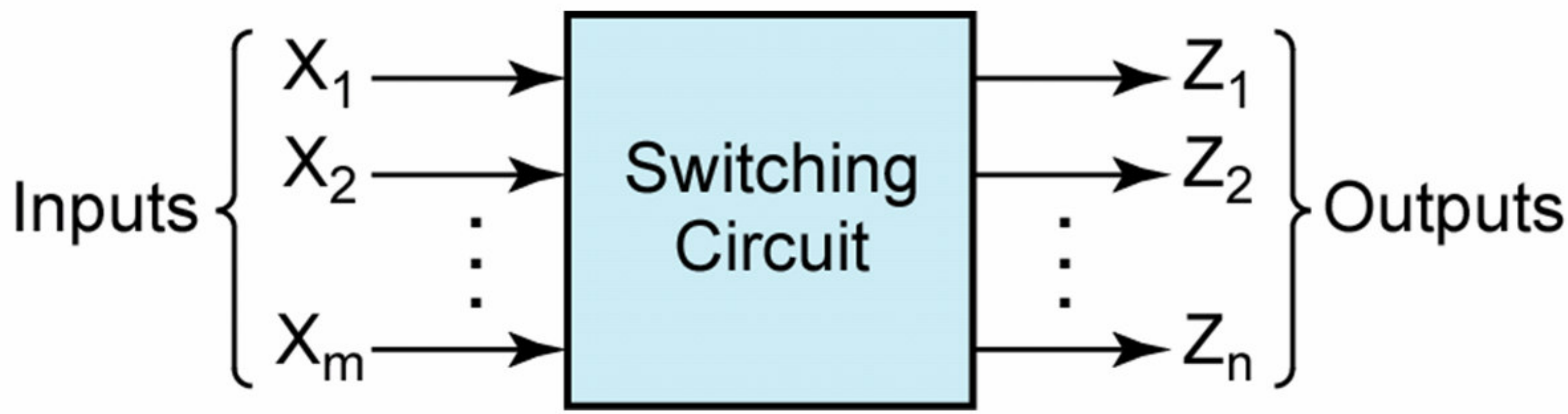


Figure 1-1: **Switching circuit**

EXAMPLE: Convert 53_{10} to binary.

2 / 53	
2 / 26	rem. $= 1 = a_0$
2 / 13	rem. $= 0 = a_1$
2 / 6	rem. $= 1 = a_2$
2 / 3	rem. $= 0 = a_3$
2 / 1	rem. $= 1 = a_4$
0	rem. $= 1 = a_5$

$$53_{10} = 110101_2$$

Conversion (a)

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

$$F = .625 \quad F_1 = .250 \quad F_2 = .500$$

$$\begin{array}{r} .625 \\ \times \quad 2 \\ \hline 1.250 \end{array} \qquad \begin{array}{r} .250 \\ \times \quad 2 \\ \hline 0.500 \end{array} \qquad \begin{array}{r} .500 \\ \times \quad 2 \\ \hline 1.000 \end{array}$$

$$(a_{-1} = 1) \qquad (a_{-2} = 0) \qquad (a_{-3} = 1)$$

$$.625_{10} = .101_2$$

Conversion (b)

EXAMPLE: Convert 0.7_{10} to binary.

```
    .7
     2
(1) .4
     2
(0) .8
     2
(1) .6
     2
(1) .2
     2
(0) .4  ←—— process starts repeating here since .4 was previously
     2        obtained above
(0) .8
```

$0.7_{10} = 0.1\,\underline{0110}\,\underline{0110}\,\underline{0110}\ldots_2$

Conversion (c)

EXAMPLE: Convert 231.3_4 to base 7.

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

$7 \,/\, 45$		$.75$	
$7 \,/\, 6$	rem. 3	$\times 7$	
0	rem. 6	$(5)\ .25$	$45.75_{10} = 63.5151\ldots_7$
		$\times 7$	
		$(1)\ .75$	
		$\times 7$	
		$(5)\ .25$	
		$\times 7$	
		$(1)\ .75$	

Conversion (d)

$$1001101.010111_2 = \underbrace{0100}_{4}\ \underbrace{1101}_{D}\ .\ \underbrace{0101}_{5}\ \underbrace{1100}_{C} = 4D.5C_{16}$$

Equation (1-1)

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

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

Addition

The subtraction table for binary numbers is

$0 - 0 = 0$
$0 - 1 = 1$ 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 ← (indicates
 11101    a borrow
−10011    from the
 -----
  1010    3rd column)
```

(b)

```
 1111 ← borrows
 10000
−    11
 -----
  1101
```

(c)

```
  111 ← borrows
 111001
−  1011
 ------
 101110
```

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:

$$
\begin{aligned}
205 - 18 &= [2 \times 10^2 + 0 \times 10^1 + 5 \times 10^0] \\
&\quad - [1 \times 10^1 + 8 \times 10^0]
\end{aligned}
$$

note borrow from column 1

$$
\begin{aligned}
&= [2 \times 10^2 + (0 - 1) \times 10^1 + (10 + 5) \times 10^0] \\
&\quad - [1 \times 10^1 + 8 \times 10^0]
\end{aligned}
$$

note borrow from column 2

$$
\begin{aligned}
&= [(2 - 1) \times 10^2 + (10 + 0 - 1) \times 10^1 + 15 \times 10^0] \\
&\quad - [1 \times 10^1 + 8 \times 10^0] \\
&= [1 \times 10^2 + 8 \times 10^1 + 7 \times 10^0] = 187
\end{aligned}
$$

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:

```
     1101
     1011
---------
     1101
    1101
   0000
  1101
---------
 10001111 = 143₁₀
```

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
     1101      multiplier
     ----
     1111      1st partial product
    0000       2nd partial product
    -----
   (01111)     sum of first two partial products
   1111        3rd partial product
   -------
 (1001011)     sum after adding 3rd partial product
  1111         4th partial product
  --------
 11000011      final product (sum after adding 4th
                            partial product)
```

Multiplication (c)

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

```
         1101
1011 ) 10010001
        1011
        ----
         1110
         1011
         -----
          1101
          1011
          ----
            10
```

The quotient is 1101 with a remainder of 10.

Binary Division

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

$+N$	Positive Integers (all systems)	$-N$	Negative Integers: Sign and Magnitude	Negative Integers: 2's Complement N^*	Negative Integers: 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 2 positive numbers, sum < 2^{n-1}.

+3	0011	
+4	0100	
+7	0111	(correct answer)

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

+5	0101	
+6	0110	
	1011	← 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).

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

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

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

2’s Complement Addition (b)

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

$$\begin{array}{rrl} -3 & & 1101 \\ -4 & & \underline{1100} \\ \hline -7 & (1) & 1001 \end{array}$$

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

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

$$\begin{array}{rrl} -5 & & 1011 \\ -6 & & \underline{1010} \\ \hline & (1) & 0101 \end{array}$$

← 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).

$+5$	0101	
-6	1001	
-1	1110	(correct answer)

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

-5	1010	
$+6$	0110	
	(1) 0000	
	└──→ 1	(end-around carry)
	0001	(correct answer, *no* overflow)

1's Complement Addition (b)

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

```
 -3        1100
 -4        1011
 ---      -----
      (1)  0111
       └──→   1   (end-around carry)
          -----
           1000   (correct answer, no overflow)
```

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

```
 -5        1010
 -6        1001
 ---      -----
      (1)  0011
       └──→   1   (end-around carry)
          -----
           0100   (wrong answer because of overflow)
```

1's Complement Addition (c)

1. Add –11 and –20 in 1's complement.

+11 = 00001011   +20 = 00010100

taking the bit-by-bit complement,

–11 is represented by 11110100 and – 20 by 11101011

$$
\begin{array}{rrl}
 & 11110100 & \quad (-11) \\
 & 11101011 & \quad +(-20) \\
\hline
(1) & 11011111 & \\
 & \hookrightarrow 1 & \quad \text{(end-around carry)} \\
\hline
 & 11100000 & = \; -31
\end{array}
$$

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

$$\begin{array}{rcr} 11111000 & & (-8) \\ 00010011 & & +19 \\ \hline (\not{1})00001011 & = & +11 \end{array}$$

(discard last carry)

2's Complement Addition (d)

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

Table 1-3 ASCII code (incomplete)

Character	ASCII Code A_6	A_5	A_4	A_3	A_2	A_1	A_0	Character	ASCII Code A_6	A_5	A_4	A_3	A_2	A_1	A_0
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