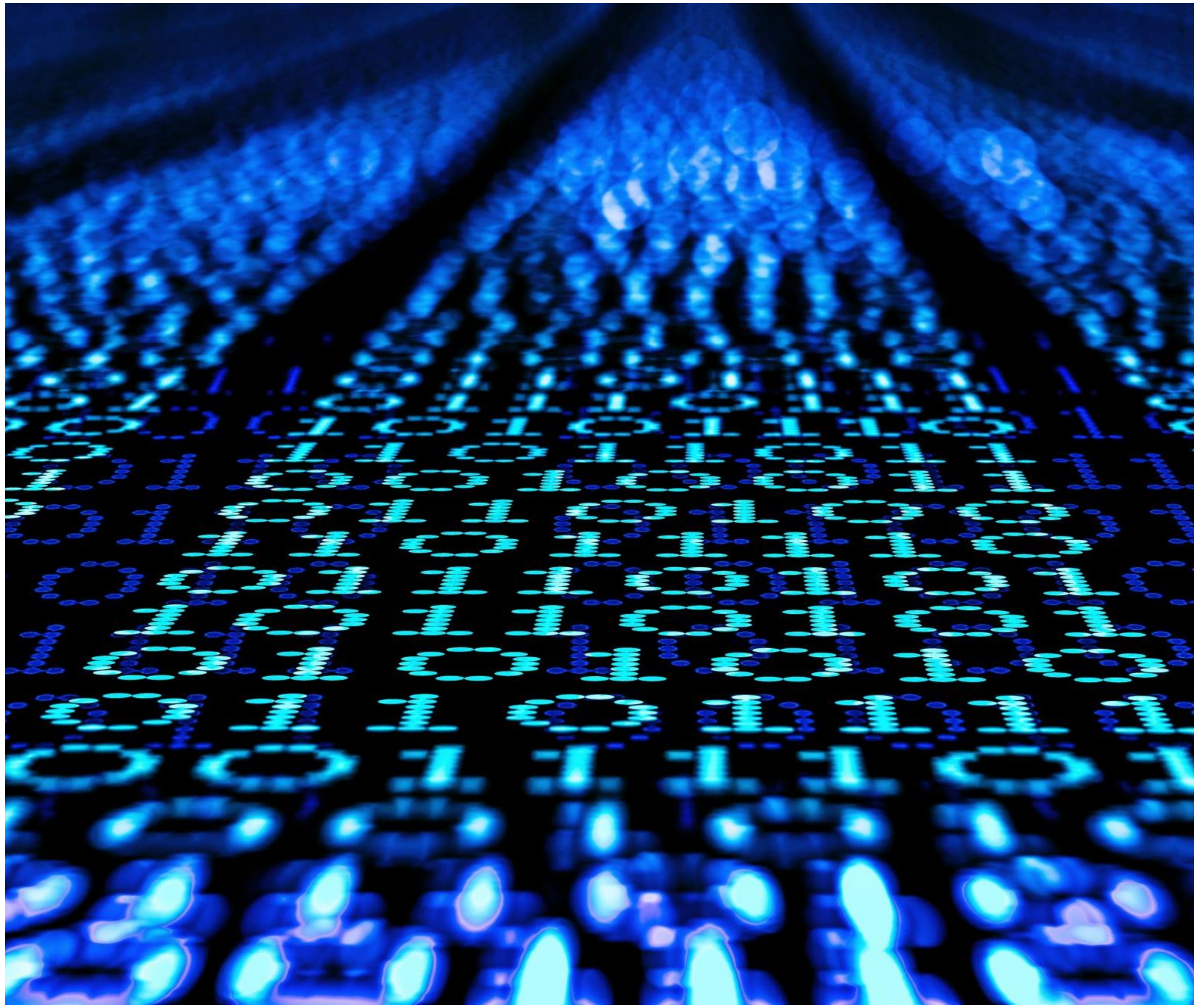
Chapter 1
Binary Systems



Binary Systems

1.1 Digital Computers and Digital Systems

1.2 Binary Numbers

1.3 Number Base Conversions

1.4 Octal and Hexadecimal Numbers

1.5 Complements

1.6 Binary Codes

1.7 Binary Storage and Registers

1.8 Binary Logics

1.9 Integrated Circuits

1.1 Digital Computers and Digital Systems

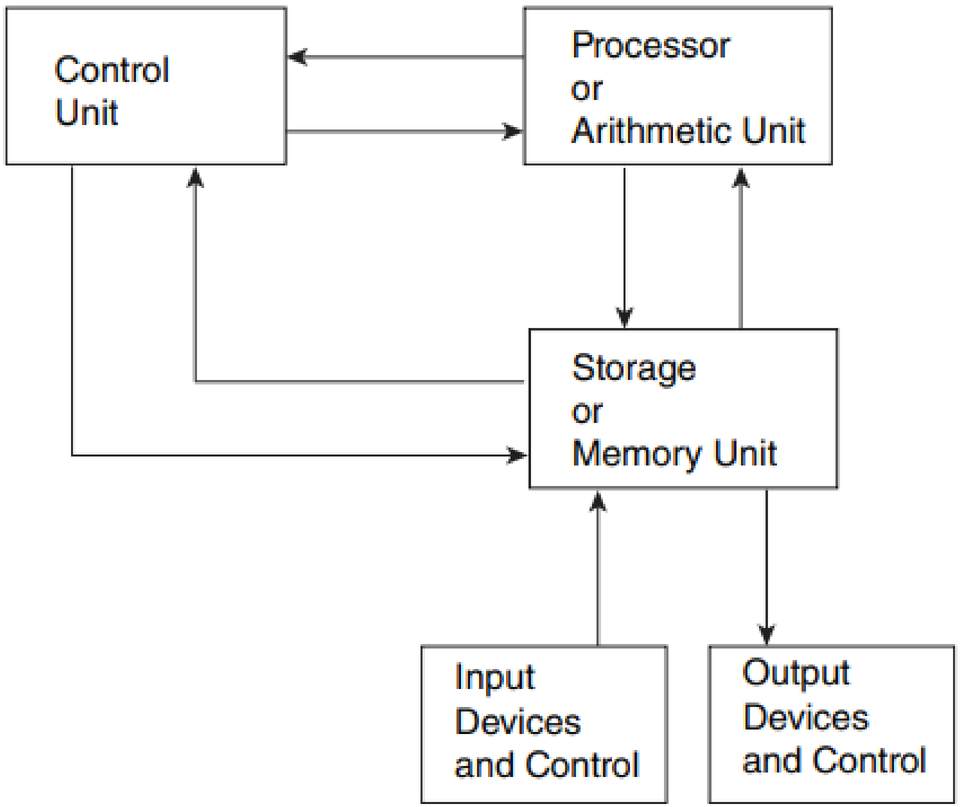


Figure 1.1 Block diagram of a digital computer

1.2 Binary Numbers

Table 1-1 Numbers with different bases

Decimal (base 10)	Binary (base 2)	Octal (base 8)	Hexadecimal (base 16)
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

1.3 Number Base Conversions

EXAMPLE 1-1: Convert decimal 41 to binary. First, 41 is divided by 2 to give an integer quotient of 20 and a remainder of $\frac{1}{2}$. The quotient is again divided by 2 to give a new quotient and remainder. This process is continued until the integer quotient becomes 0. The *coefficients* of the desired binary number are obtained from the *remainders* as follows:

integer quotient		*remainder*	*coefficient*
$\frac{41}{2} = 20$	+	$\frac{1}{2}$	$a_0 = 1$
$\frac{20}{2} = 10$	+	0	$a_1 = 0$
$\frac{10}{2} = 5$	+	0	$a_2 = 0$
$\frac{5}{2} = 2$	+	$\frac{1}{2}$	$a_3 = 1$
$\frac{2}{2} = 1$	+	0	$a_4 = 0$
$\frac{1}{2} = 0$	+	$\frac{1}{2}$	$a_5 = 1$

answer: $(41)_{10} = (a_5a_4a_3a_2a_1a_0)_2 = (101001)_2$

1.3 Number Base Conversions

The arithmetic process can be manipulated more conveniently as follows:

integer	*remainder*
41	
20	1
10	0
5	0
2	1
1	0
0	1

101001 = answer

1.3 Number Base Conversions

EXAMPLE 1-4: Convert $(0.513)_{10}$ to octal.

$$0.513 \times 8 = 4.104$$
$$0.104 \times 8 = 0.832$$
$$0.832 \times 8 = 6.656$$
$$0.656 \times 8 = 5{,}248$$
$$0.248 \times 8 = 1.984$$
$$0.984 \times 8 = 7.872$$

The answer, to seven significant figures, is obtained from the integer part of the products:

$$(0.513)_{10} = (0.406517 \ldots)_8$$

1.4 Octal and Hexadecimal Numbers

$$(\underbrace{10}_{2}\ \underbrace{110}_{6}\ \underbrace{001}_{1}\ \underbrace{101}_{5}\ \underbrace{011}_{3} \cdot \underbrace{111}_{7}\ \underbrace{100}_{4}\ \underbrace{000}_{0}\ \underbrace{110}_{6})_2 = (26153.7406)_8$$

Conversion from binary to hexadecimal is similar, except that the binary number is divided into groups of four digits:

$$(\underbrace{10}_{2}\ \underbrace{1100}_{C}\ \underbrace{0110}_{6}\ \underbrace{1011}_{B} \cdot \underbrace{1111}_{F}\ \underbrace{0010}_{2})_2 = (2C6B.F2)_{16}$$

1.4 Octal and Hexadecimal Numbers

Conversion from octal or hexadecimal to binary is done by a procedure reverse to the above. Each octal digit is converted to its three-digit binary equivalent. Similarly, each hexadecimal digit is converted to its four-digit binary equivalent. This is illustrated in the following examples:

$$(673.124)_8 = (\underbrace{110}_{6}\ \underbrace{111}_{7}\ \underbrace{011}_{3} \cdot \underbrace{001}_{1}\ \underbrace{010}_{2}\ \underbrace{100}_{4})_2$$

$$(306.\ D)_{16} = (\underbrace{0011}_{3}\ \underbrace{0000}_{0}\ \underbrace{0110}_{6} \cdot \underbrace{1101}_{D})_2$$

1.5 Complements

1.5.1 The r's Complement

1.5.2 The (r – 1)'s Complement

1.5.3 Subtraction with r's Complements

1.5.4 Subtraction with (r – 1)'s Complement

1.5.5 Comparison between 1's and 2's Complements

1.5.1 The r's Complement

Given a positive number N in base r with an integer part of n digits, the r's complement of N is defined as $r^n - N$ for $N \neq 0$ and 0 for $N = 0$. The following numerical example will help clarify the definition.

The 10's complement of $(52520)_{10}$ is $10^5 - 52520 = 47480$.

The number of digits in the number is $n = 5$.

The 10's complement or $(0.3267)_{10}$ is $1 - 0.3267 = 0.6733$.

1.5.2 The (r – 1)'s Complement

The 9's complement of $(52520)_{10}$ is $(10^5 - 1 - 52520) = 99999 - 52520 = 47479$.

No fraction part, so $10^{-m} = 10^0 = 1$.

The 9's complement of $(0.3267)_{10}$ is $(1 - 10^{-4}\ 0.3267) = 0.9999 - 0.3267 = 0.6732$.

No integer part, so $10^n = 10^0 - 1$.

The 9's complement of $(25.639)_{10}$ is $(10^2 - 10^{-3} - 25.639) = 99.999 - 25.639 = 74.360$.

The l's complement of $(101100)_2$ is $(2^6 - 1) - (101100) = (111111 - 101100)_2 = 010011$.

The l's complement of $(0.0110)_2$ is $(1 - 2^{-4})_{10} - (0.0110)_2 = (0.1111 - 0.0110)_2 = 0.1001$.

The 7's complement of $(76)_8$ is $(8^2 - 8^0)_{10} - 76_8 = (63)_{10} - 76_8 = (77 - 76)_8 = 1_8$

1.5.3 Subtraction with r's Complements

EXAMPLE 1-5: Using 10's complement, subtract 72532 – 3250.

$$
\begin{array}{rr}
M = 72532 & 72532 \\
N = 03250 & \\
 & + \\
\text{10's complement of } N = 96750 & 96750 \\
\hline
\text{end carry} \rightarrow 1 & 69282
\end{array}
$$

answer: 69282

1.5.3 Subtraction with r's Complements

EXAMPLE 1-6: Subtract: $(3250 - 72532)_{10}$.

$$M = 03250 \qquad\qquad 03250$$
$$N = 72532$$
$$+$$
$$\text{10's complement of } N = 27468 \qquad\qquad 27468$$
$$\text{no carry} \qquad\qquad 30718$$

answer: – 69282 = – (10**'s** complement of 30718)

1.5.3 Subtraction with r's Complements

EXAMPLE 1-7: Use 2's complement to perform $M - N$ with the given binary numbers.

(a) $M = 1010100$
$N = 1000100$

2's complement of $N = 0111100$

$$\begin{array}{r} 1010100 \\ +\ 0111100 \\ \hline \text{end carry} \rightarrow 1\ \ 0010000 \end{array}$$

answer: 10000

(b) $M = 1000100$
$N = 1010100$

2's complement of $N = 0101100$

$$\begin{array}{r} 1000100 \\ +\ 0101100 \\ \hline \text{no carry}\ \ 1110000 \end{array}$$

answer: – 10000 = – (2's complement of 1110000)

1.5.4 Subtraction with (r – 1)'s Complement

EXAMPLE 1-8: Repeat Examples 1-5 and 1-6 using 9's complements.

(a)

$$M = 72532$$
$$N = 03250$$
$$\text{9's complement of } N = 96749$$

$$\begin{array}{r} 72532 \\ +\ 96749 \\ \hline 1\ 69281 \\ +\ 1 \\ \hline 69282 \end{array}$$

end-around carry

answer: 69282

(b)

$$M = 03250$$
$$N = 72532$$
$$\text{9's complement of } N = 27467$$

$$\begin{array}{r} 03250 \\ +\ 27467 \\ \hline 30717 \end{array}$$

no carry

answer: – 69282 = – (9's complement of 30717)

1.5.4 Subtraction with (r – 1)'s Complement

EXAMPLE 1-9: Repeat Example 1 = 7 using l's complement.

(a)	$M = 1010100$	1010100
	$N = 1000100$	
	l's complement of $N = 0111011$	+ 0111011
	end-around carry	1 0001111
		+ 1
		0010000

answer: 10000

(b)	$M = 1000100$	1000100
	$N = 1010100$	+
	l's complement of $N = 0101011$	0101011
	no carry	1101111

answer: – 10000 = – (l's complement of 1101111)

1.5.5 Comparison between 1's and 2's Complements

Using l's complement:

$$\begin{array}{r} 1100 \\ + \quad 0011 \\ \hline + \quad 1111 \end{array}$$

Complement again to obtain −0000.

Using 2's complement:

$$\begin{array}{r} 1100 \\ + \quad 0100 \\ \hline + \quad 0000 \end{array}$$

While the 2's complement has only one arithmetic zero, the l's complement zero can be positive or negative, which may complicate matters.

1.6 Binary Codes

Table 1-2 Binary codes for the decimal digits

Decimal digit	(BCD) 8421	Excess-3	84-2-1	2421	(Biquinary) 5043210
0	0000	0011	0000	0000	0100001
1	0001	0100	0111	0001	0100010
2	0010	0101	0110	0010	0100100
3	0011	0110	0101	0011	0101000
4	0100	0111	0100	0100	0110000
5	0101	1000	1011	1011	1000001
6	0110	1001	1010	1100	1000010
7	0111	1010	1001	1101	1000100
8	1000	1011	1000	1110	1001000
9	1001	1100	1111	1111	1010000

1.6.1 Decimal Codes

- Numbers are represented in digital computers either in binary or in decimal through a binary code.
- For example, the decimal number 395, when converted to binary, is equal to 110001011 and consists of nine binary digits.
- The same number, when represented internally in the BCD code, occupies four bits for each decimal digit, for a total of 12 bits: 001110010101. The first four bits represent a 3, the next four a 9, and the last four a 5

1.6.2 Error-Detection Codes

Table 1-3 Parity-bit generation

(a) Message	P(odd)	(b) 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

1.6.3 The Reflected Code

Table 1-4 Four-bit reflected code

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

1.6.4 Alphanumeric Codes

Table 1-5 Alphanumeric character codes

Character	6-Bit internal code		7-Bit ASCII code		8-Bit EBCDIC code		12-Bit card code
A	010	001	100	0001	1100	0001	12,1
B	010	010	100	0010	1100	0010	12,2
C	010	011	100	0011	1100	0011	12,3
D	010	100	100	0100	1100	0100	12,4
E	010	101	100	0101	1100	0101	12,5
F	010	110	100	0110	1100	0110	12,6
G	010	111	100	0111	1100	0111	12,7
H	011	000	100	1000	1100	1000	12,8
I	011	001	100	1001	1100	1001	12,9
J	100	001	100	1010	1101	0001	11,1
K	100	010	100	1011	1101	0010	11,2
L	100	011	100	1100	1101	0011	11.3
M	100	100	100	1101	1101	0100	11,4

1.7 Binary Storage and Registers

1.7.1 Registers

1.7.2 Register Transfer

1.7.1 Registers

A *register* is a group of binary cells. Since a cell stores one bit of information, it follows that a register with n cells can store any discrete quantity of information that contains n bits. The *state* of a register is an n-tuple number of 1's and 0's, with each bit designating the state of one cell in the register. The *content* of a register is a function of the interpretation given to the information stored in it. Consider, for example, the following 16-cell register:

1	1	0	0	0	0	1	1	1	1	0	0	1	0	0	1
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

1.7.2 Register Transfer(a)

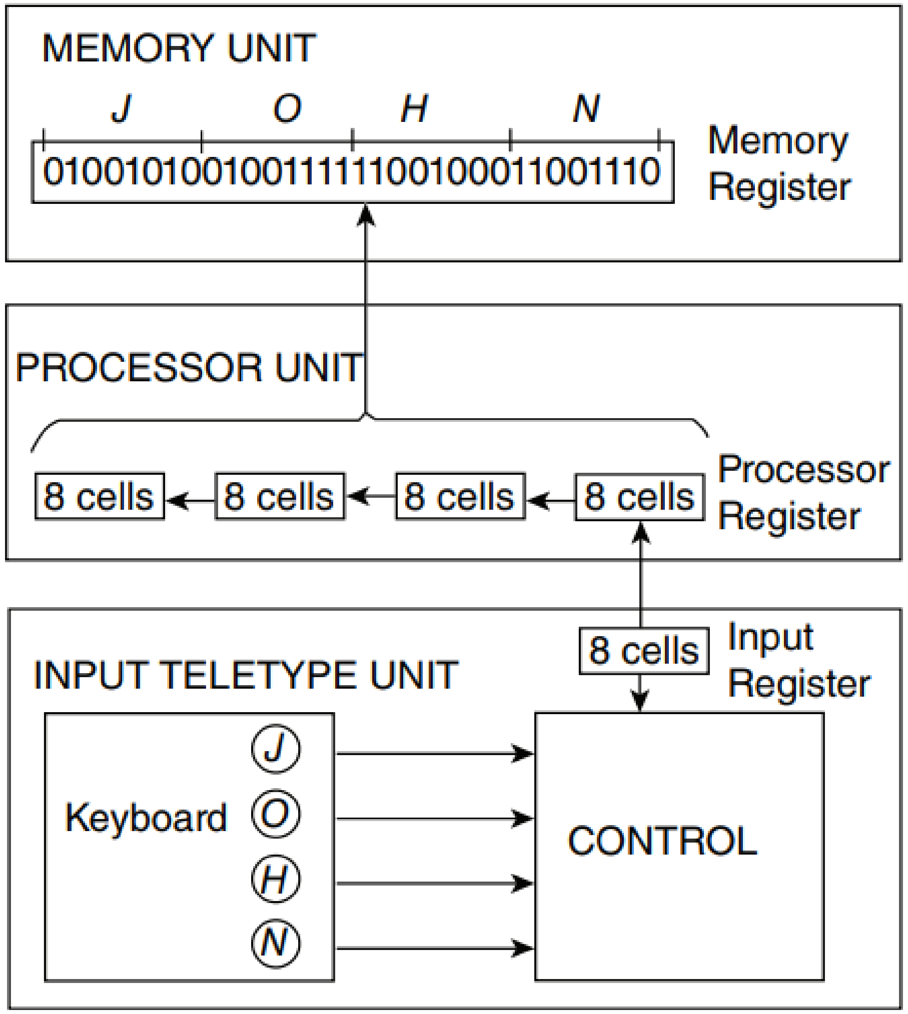


Figure 1.2 Transfer of information with registers

1.7.2 Register Transfer(b)

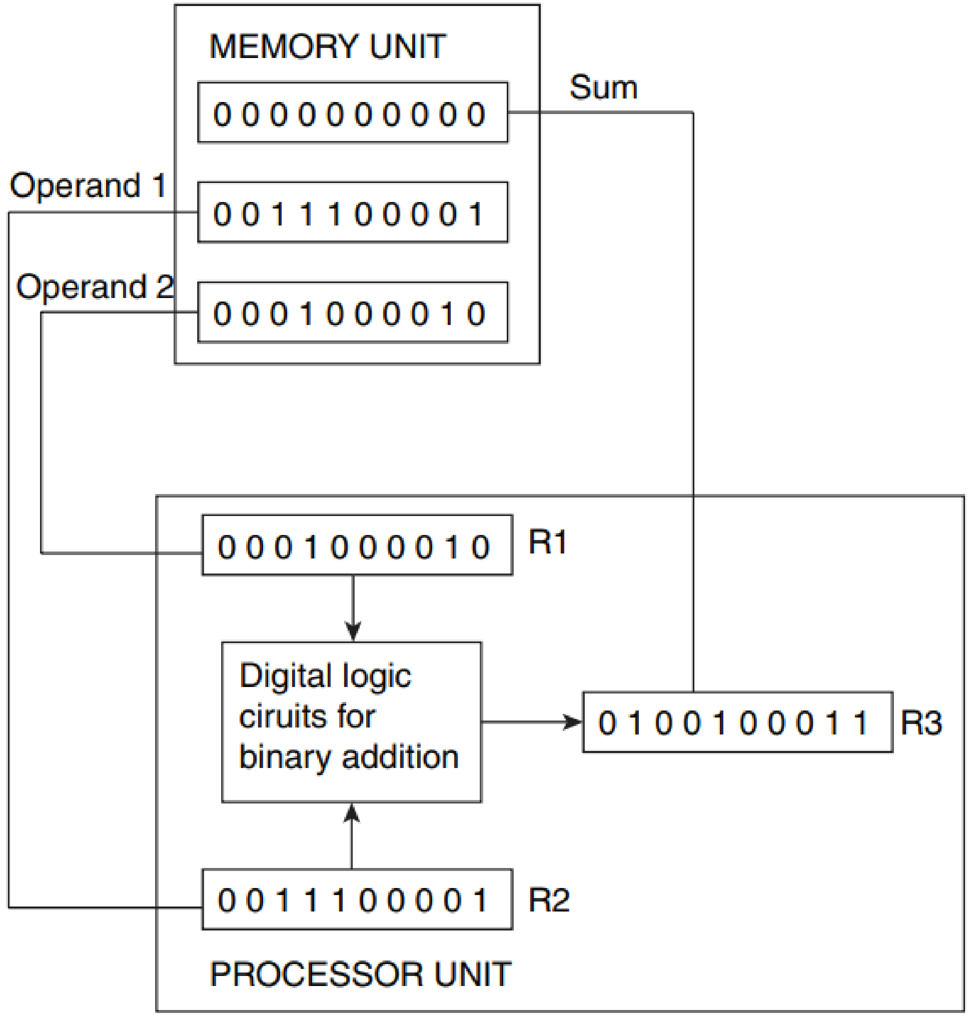


Figure 1.3 Example of binary information processing

1.8 Binary Logics

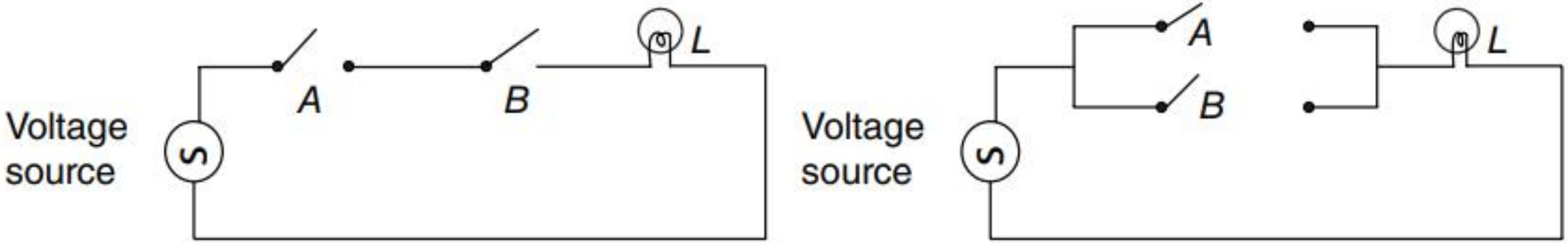
- 1.8.1 Definition of Binary Logic
- 1.8.2 Switching Circuits and Binary Signals
- 1.8.3 Logic Gates

1.8.1 Definition of Binary Logic

Table 1-6 Truth tables of logical operations

AND			OR			NOT	
x	y	$x \cdot y$	x	y	$x + y$	x	x'
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

1.8.2 Switching Circuits and Binary Signals



(a) Switches in series – logic AND (b) Switches in parallel – Logic OR

Figure 1.4 Switching circuits that demonstrate binary logic

1.8.3 Logic Gates

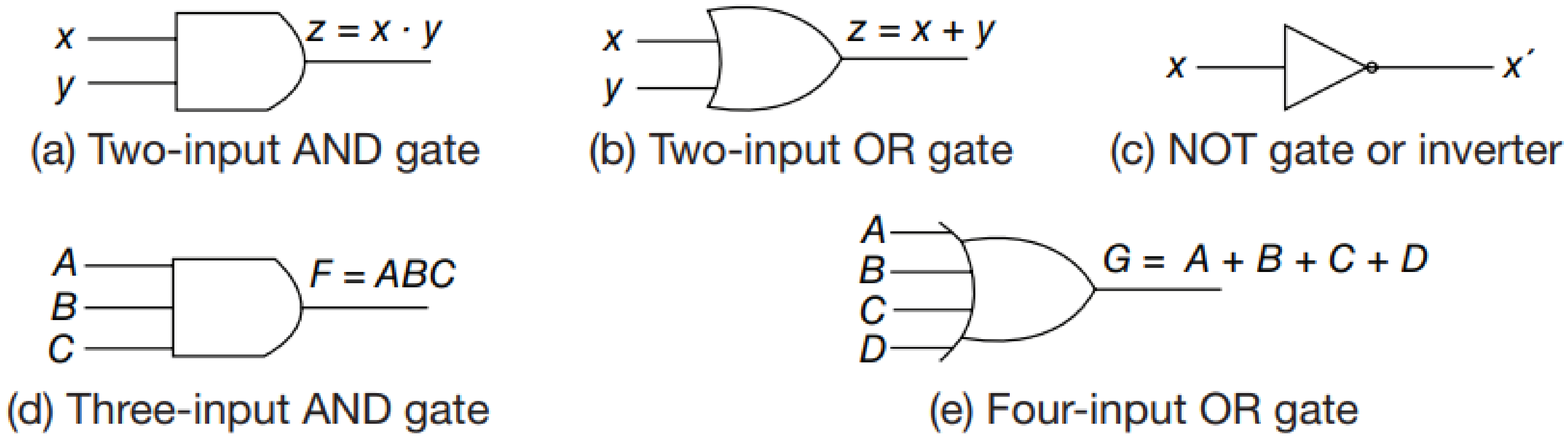


Figure 1.6 Symbols for digital logic circuits

1.9 Integrated Circuits(a)

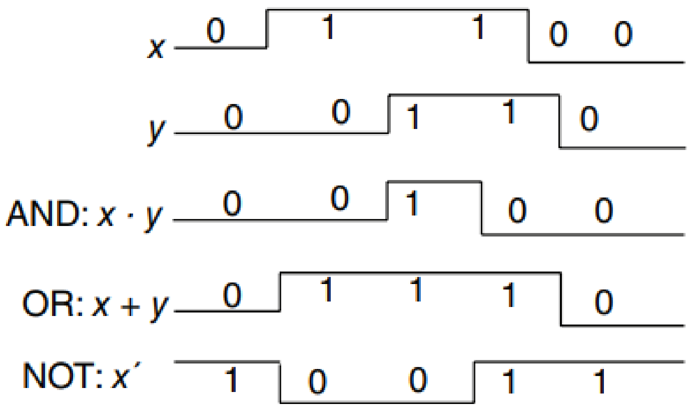


Figure 1.7 Input-output signals for gates (a), (b), and (c) of Fig. 1.6

1.9 Integrated Circuits(b)

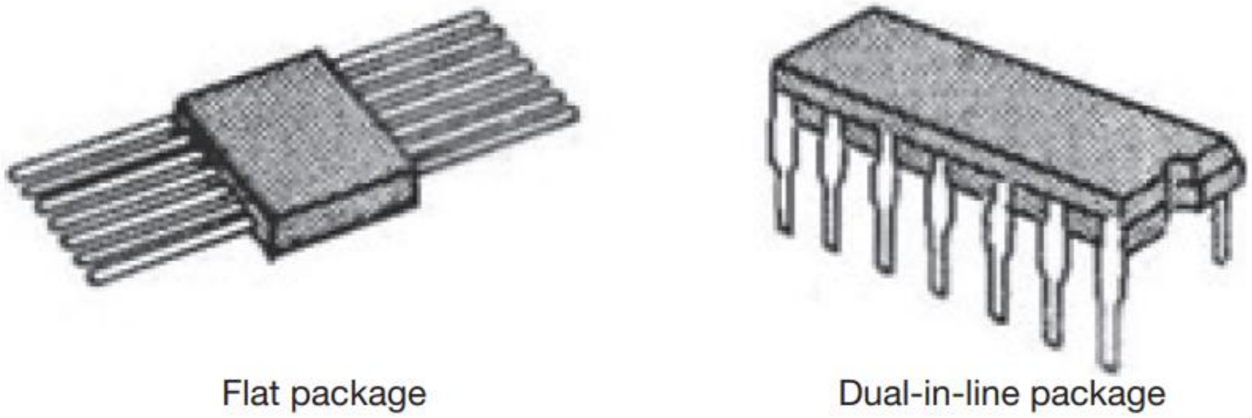


Figure 1.8 Integrated-circuit packages

The End