

Logic Circuits

Semester 5

Ahmad Abu Zainab

Contents

Chapter 1	Number Systems and Conversion	Page 2
1.1	Decimal and Binary Decimal to Binary — 2 • Binary to Decimal — 3	2
1.2	Binary and Hexidecimal	3
1.3	Operations on Binary Numbers Addition — 3 • Subtraction — 3 • Multiplication — 4 • Division — 4	3
1.4	Negative Numbers in Binary	4
Chapter 2	Boolean Algebra	Page 5
2.1	Logic Gates Switches — 6	5
2.2	Laws and Theorems	6
2.3	De Morgan's Theorems	7
2.4	Duality	7

Chapter 1

Number Systems and Conversion

- **Decimal** (Base 10): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- **Binary** (Base 2): 0, 1
- **Octal** (Base 8): 0, 1, 2, 3, 4, 5, 6, 7
- **Hexadecimal** (Base 16): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

1.1 Decimal and Binary

1.1.1 Decimal to Binary

1. Divide the decimal number by 2.
2. Keep the remainder.
3. Divide the quotient by 2.
4. Keep the remainder.
5. Repeat until the quotient is 0.
6. The binary number is the remainders in reverse order.

If the number has a decimal point then we multiply the decimal part by 2 the integer part of the result is the binary digit and the decimal part is multiplied by 2 again and so on until we obtain only a 1.

Example 1.1.1 (Convert 53_{10} to binary.)

$$\begin{aligned}53/2 &= 26 \text{ remainder } 1 \\26/2 &= 13 \text{ remainder } 0 \\13/2 &= 6 \text{ remainder } 1 \\6/2 &= 3 \text{ remainder } 0 \\3/2 &= 1 \text{ remainder } 1 \\1/2 &= 0 \text{ remainder } 1\end{aligned}$$

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

Example 1.1.2 (Convert $.625_{10}$ to binary)

$$.625 \times 2 = 1.25$$

$$.25 \times 2 = .5$$

$$.5 \times 2 = 1$$

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

1.1.2 Binary to Decimal

1. Write the binary number.
2. Multiply each digit by 2^n where n is the position of the digit.
3. Add the results.

Example 1.1.3 (Convert 110101_2 to decimal.)

$$\begin{aligned} 110101_2 &= 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 32 + 16 + 0 + 4 + 0 + 1 \\ &= 53_{10} \end{aligned}$$

1.2 Binary and Hexidecimal

To convert from binary to hexadecimal we group the binary digits into groups of 4 starting from the right and convert each group to a hexadecimal digit.

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

1.3 Operations on Binary Numbers

1.3.1 Addition

$$\begin{array}{r} \text{carry} \quad 1 \\ \quad 0 \quad 1 \quad 1 \\ + \quad 1 \quad 1 \quad 0 \quad . \\ \hline \quad 1 \quad 0 \quad 1 \end{array}$$

1.3.2 Subtraction

$$\begin{array}{r} \text{borrow} \quad 1 \quad 1 \quad 1 \\ \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \\ - \quad \quad \quad 1 \quad 0 \quad 1 \quad 1 \quad . \\ \hline \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \end{array}$$

1.3.3 Multiplication

$$0 \times 0 = 0$$

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 1 = 1$$

$$\begin{array}{rrrrrrrr} & & & & 1 & 1 & 0 & 1 \\ \times & & & & 1 & 0 & 1 & 1 \\ \hline & & & & 1 & 1 & 0 & 1 \\ & & & 1 & 1 & 0 & 1 & \\ & & 0 & 0 & 0 & 0 & & \\ & 1 & 1 & 0 & 1 & & & \\ \hline 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}.$$

1.3.4 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.

1.4 Negative Numbers in Binary

Sign and Magnitude The leftmost bit is the sign bit and the rest of the bits are the magnitude.

$$-5_{10} = 1101_2.$$

One's Complement To negate a number we flip all the bits. $\bar{N} = (2^n - 1) - N$ where n is the number of bits.

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

Two's Complement To negate a number we flip all the bits and add 1. $N^* = 2^n - N$ where n is the number of bits.

$$-5_{10} = 2^4 - 5 = 11_{10} = 1011_2.$$

Adding a negative number in two's complement is the same as subtracting the positive number. In one's complement we add the numbers and add the carry to the result.

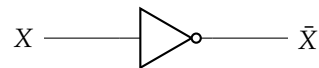
Chapter 2

Boolean Algebra

2.1 Logic Gates

NOT Gate The output is the inverse of the input.

X	\bar{X}
0	1
1	0



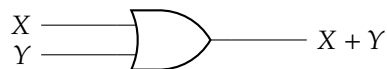
AND Gate The output is 1 if both inputs are 1.

X	Y	$X \cdot Y$
0	0	0
0	1	0
1	0	0
1	1	1



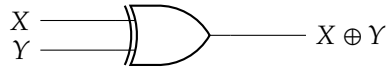
OR Gate The output is 1 if either input is 1.

X	Y	$X + Y$
0	0	0
0	1	1
1	0	1
1	1	1



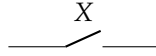
XOR Gate The output is 1 if either input is 1 but not both.

X	Y	$X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0



2.1.1 Switches

If switch X is open, then we will define the value of X to be 0; if switch X is closed, then we will define the value of X to be 1.



2.2 Laws and Theorems

- Operations involving 0 and 1

$$X + 0 = X$$

$$X + 1 = 1$$

$$X \cdot 1 = X$$

$$X \cdot 0 = 0$$

- Idempotent Laws

$$X + X = X$$

$$X \cdot X = X$$

- Involution Law

$$\overline{\overline{X}} = X.$$

- Laws of complementation

$$X + \bar{X} = 1$$

$$X \cdot \bar{X} = 0$$

- Commutative Laws

$$X + Y = Y + X$$

$$X \cdot Y = Y \cdot X$$

- Associative Laws

$$X + (Y + Z) = (X + Y) + Z$$

$$X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$$

- Distributive Laws

$$X \cdot (Y + Z) = X \cdot Y + X \cdot Z$$

$$X + Y \cdot Z = (X + Y) \cdot (X + Z)$$

- Simplification Laws

$$\begin{aligned}
X \cdot Y + X \cdot \bar{Y} &= X \\
X + X \cdot Y &= X \\
(X + \bar{Y}) \cdot Y &= X \cdot Y \\
(X + Y) \cdot (X + \bar{Y}) &= X \\
X \cdot (X + Y) &= X \\
X \cdot \bar{Y} + Y &= X + Y
\end{aligned}$$

An expression is in sum of products form if it is a sum of products of literals.

$$A\bar{B} + C\bar{D}E + A\bar{C}\bar{E}.$$

Likewise, an expression is in product of sums form if it is a product of sums of literals.

$$(A + B)(C + D)E.$$

2.3 De Morgan's Theorems

$$\begin{aligned}
\overline{X + Y} &= \bar{X} \cdot \bar{Y} \\
\overline{X \cdot Y} &= \bar{X} + \bar{Y}
\end{aligned}$$

2.4 Duality

Given an expression, we can obtain its dual by replacing all ANDs with ORs and all ORs with ANDs and all 0s with 1s and all 1s with 0s.

$$\begin{aligned}
(XYZ \cdots)^D &= X + Y + Z + \cdots \\
(X + Y + Z + \cdots)^D &= XYZ \cdots
\end{aligned}$$

The dual can also be found by taking the complement of the whole expression then complementing each variable individually