

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 |
| 2.5 | Minterm and Maxterm Expansions | 7 |
| Chapter 3 | Karnaugh Maps | Page 9 |

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

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

1.3.3 Multiplication

$$0 \times 0 = 0$$

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 1 = 1$$

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

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.

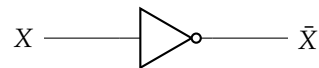
Chapter 2

Boolean Algebra

2.1 Logic Gates

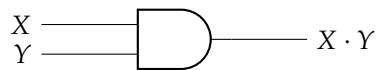
NOT Gate The output is the inverse of the input.

| X | \bar{X} |
|-----|-----------|
| 0 | 1 |
| 1 | 0 |



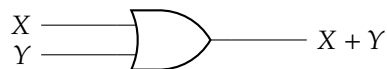
AND Gate The output is 1 if both inputs are 1.

| X | Y | $X \cdot Y$ |
|-----|-----|-------------|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



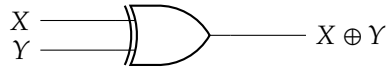
OR Gate The output is 1 if either input is 1.

| X | Y | $X + Y$ |
|-----|-----|---------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |



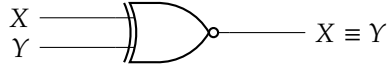
XOR Gate The output is 1 if either input is 1 but not both.

| X | Y | $X \oplus Y$ |
|-----|-----|--------------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



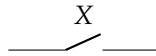
Equivalence Gate The output is 1 if both inputs are the same.

| X | Y | $X \equiv Y$ |
|---|---|--------------|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



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

$$\begin{aligned} X + 0 &= X \\ X + 1 &= 1 \\ X \cdot 1 &= X \\ X \cdot 0 &= 0 \end{aligned}$$

- Idempotent Laws

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

- Involution Law

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

- Laws of complementation

$$\begin{aligned} X + \bar{X} &= 1 \\ X \cdot \bar{X} &= 0 \end{aligned}$$

- Commutative Laws

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

- Associative Laws

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

- Distributive Laws

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

- Simplification Laws

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

$$\begin{aligned}
X \oplus 0 &= X \\
X \oplus 1 &= \bar{X} \\
X \oplus X &= 0 \\
X \oplus \bar{X} &= 1 \\
X \oplus Y &= Y \oplus X \\
X \oplus (Y \oplus Z) &= (X \oplus Y) \oplus Z = X \oplus Y \oplus Z \\
X \cdot (Y \oplus Z) &= X \cdot Y \oplus X \cdot Z \\
\overline{X \oplus Y} &= \bar{X} \oplus Y = X \oplus \bar{Y} = X \cdot Y + \bar{X} \cdot \bar{Y} = X \equiv Y
\end{aligned}$$

Consensus Theorem

$$X \cdot Y + \bar{X} \cdot Z + Y \cdot Z = X \cdot Y + \bar{X} \cdot Z.$$

in dual form

$$(X + Y)(\bar{X} + Z)(Y + Z) = (X + Y)(\bar{X} + Z).$$

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

2.5 Minterm and Maxterm Expansions

Minterm A product term containing all the variables of the function in either true or complemented form.

Maxterm A sum term containing all the variables of the function in either true or complemented form.

Any Boolean function can be expressed as a sum of minterms or a product of maxterms.

Example 2.5.1 (Minterm)

$$f(A, B, C) = \underbrace{\bar{A}\bar{B}\bar{C}}_{000} + \underbrace{\bar{A}\bar{B}C}_{001} + \underbrace{\bar{A}BC}_{011} + \underbrace{A\bar{B}\bar{C}}_{010} + \underbrace{ABC}_{111} = \Sigma m(0, 1, 3, 2, 7).$$

Example 2.5.2 (Maxterm)

$$f(A, B, C) = \underbrace{(\bar{A} + \bar{B} + \bar{C})}_{111} \cdot \underbrace{(\bar{A} + \bar{B} + C)_{110}} \cdot \underbrace{(\bar{A} + B + C)_{100}} \cdot \underbrace{(\bar{A} + B + \bar{C})_{101}} \cdot \underbrace{(A + B + C)_{000}} = \Pi M(7, 6, 4, 5, 0).$$

Chapter 3

Karnaugh Maps

A Karnaugh map is a graphical representation of a truth table.

It is useful for simplifying Boolean expressions, by grouping together adjacent cells containing 1s.

We first draw a grid with the number of rows and columns equal to the number of variables.

We then label the rows and columns with the binary values of the variables using a Gray code. (000, 001, 011, 010, 110, 111, 101, 100) No 2 adjacent numbers differ by more than 1 bit

We then fill in the cells with the output of the function.

We then group together adjacent cells containing 1s in powers of 2 (1, 2, 4, 8, 16, 32, 64, 128, 256, etc.)

We then go over the groups and write down the variables that remain the same throughout the group.

Example 3.0.1 (Two Variable K-map)

Take the following K-map

| $\begin{matrix} A \\ B \end{matrix}$ | 0 | 1 |
|--------------------------------------|---|---|
| 0 | 1 | |
| 1 | 1 | 1 |

We look at the first group is 0 and 2(vertical), in this group A is not changing and it's value is 0. So the first group corresponds to \bar{A} .

The second group is 2 and 3(horizontal), in this group B is not changing and it's value is 1. So the second group corresponds to B .

Therefore the simplified expression is

$$f(A, B) = \bar{A} + B.$$

Example 3.0.2 (Three Variable K-map)

Take the following function in min-term form

$$f(A, B, C) = \Sigma m(0, 1, 3, 4, 5, 7).$$

We can represent this function in a K-map as follows

| $A \backslash BC$ | 0 | 1 |
|-------------------|---|---|
| 00 | 1 | 1 |
| 01 | 1 | |
| 11 | 1 | 1 |
| 10 | | 1 |

First group(red) the only changing variable is A . So it's excluded in the term of the group which is $\bar{B}\bar{C}$.
 Second group(yellow) the only changing variable is B . So it's excluded in the term of the group which is $\bar{A}C$.

Third group(green) the only changing variable is C . So it's excluded in the term of the group which is AB .

Therefore the simplified expression is

$$f(A, B, C) = \bar{B}\bar{C} + \bar{A}C + AB.$$

Example 3.0.3 (Four Variable K-map)

Take the following function in min-term form

$$f(A, B, C, D) = \Sigma m(0, 1, 2, 3, 5, 7, 8, 10).$$

We can represent this function in a K-map as follows

| | AB | | | |
|----|------|----|----|----|
| | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 | 1 | 1 |
| 01 | | 1 | 1 | |
| 11 | | | | |
| 10 | 1 | | | 1 |

| | AB | | | |
|----|------|----|----|----|
| | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 | 1 | 1 |
| 01 | | 1 | 1 | |
| 11 | | | | |
| 10 | 1 | | | 1 |

First K-map gives us the simplified form as

$$f(A, B, C, D) = B\bar{C} + \bar{B}\bar{D}.$$

And the second K-map gives us

$$f(A, B, C, D) = \bar{C}\bar{D} + B\bar{C}D + \bar{B}C\bar{D}.$$

To utilize maxterms in a K-map we first fill the maxterm cells with 0s and the rest with 1s, and then we follow the same procedure as before. This will give us \bar{f} then we simply compute the inverse to get f .

Note:-

Sometimes we don't care about some of the outputs of a function (Don't care conditions). We can represent this in a K-map by filling the cells with don't care conditions with an X. An example is

$$f(A, B, C) = \Sigma m(0, 1, 3, 4, 5, 7) + d(2, 6).$$