1 Chapter Outline

1. Boolean Equations

   Boolean Equation is an equation which deals Boolean values, true or false. By using literals of two discrete values, it is possible to find the functionality of the gate or circuit, and able to design the output by controlling the inputs via logical gates. This is run by two forms of Boolean equations, Sum-of-Product and Product-of-Sum. Sum-of-Product takes the rows(sum) which are false in a product form, and Product-of-Sum takes the rows(product) which are true in a sum form. There are variations in indicating logic by Boolean equation because it is able to represent the logic differently by Sum-of-Product and Product-of-Sum.

2. Boolean Algebra

   The logic can be expressed in form of Boolean equation. However, it cannot be always expressed in a simple form of Boolean equation by using truth table. In order to simplify the complicated Boolean equation, Boolean algebra is necessary. Boolean algebra starts from the axioms including binary field, Not, And/or and etc. By adding more variables from the axioms, it is possible to simplify the complicated Boolean equation from the observation, and the design of the circuit.

3. Logic to Gates

   It is important to implement the gates from the obtained logic from the Boolean equation and algebra. To avoid the confusion in implementing the gates, there are rules in designing schematic because many circuits have multiple input and output, and each input is a Boolean function. For instance, inputs are on the left or top side of a schematic, and outputs are on the right or bottom side of a schematic. Furthermore, there are ways in drawing wires to distinguish the connection of the wire.

4. Multilevel Combinational Logic

   Many circuits are connected with more than one logic gate. In other words, an output of a logic gate can be an input of another logic gate. To find the Boolean function of the output of the circuit, interpretation of the multilevel combinational logic is necessary. To interpret the multilevel combinational logic, bubble pushing will simplify the complicated combination of logic gates so that it is easier to interpret the circuit.

5. X(Contention)'s and Z(Floating)'s

   In the previous sections, it mentioned that the circuit deals Boolean values. However, in this section, there are two more values which are not 0 or 1, but exist in the real-world circuit. There are X(contention) and Z(floating) in the real-world circuit.

   X value named contention is an actual value which is somewhere in between. Contention can be existing in 0, 1, or forbidden zone, and change with temperature, voltage, time, and noise. Furthermore, contention causes excessive power dissipation, and indicates a bug.

Z value named floating is a high impedance which exists in 0, 1, or somewhere between, but voltmeter will not indicate. Nodes which are floating are used in tristate busses.

6. Karnaugh Maps

When the number of the input of the circuit is less, it is able to find the logic of the output by using truth table. However, if the number of the input is not able to handle (i.e., the number of inputs is at least four) by using truth table, finding the logic of the output will require complicated Boolean equation or Boolean algebra. To minimize the process of finding the logic, Karnaugh Maps are required.

Karnaugh Map is in form of table, and marks down the result of Y with specified rows and columns. For instance, if the inputs are A, B, C, and D, then row can be AB, and the column can be CD. By mapping the circle by the rule of Karnaugh Map, the Boolean equation of Y is found by using Boolean algebra.

7. Combinational Building Blocks

Combinational logic is used to build much more complex system. In this process, the blocks are used to build the complex system. In the blocks, there are multiplexer and decoder.

Multiplexer is selects one of the inputs, and then connects to the output. Multiplexer is implemented by tristate and logic gates. Moreover, multiplexer's size can be reduced by using logic.

Decoder produces 2^n outputs when there are n inputs, and it only makes one output HIGH at once. Decoder is implemented by the combination of Not and And gates.

8. Timing

Timing is a delay between the change of input and the change of output which is caused by resistance and capacitance of the circuit and the speed limit of the light. There are propagation delay and contamination delay. Propagation delay is maximum delay from input to output, and contamination delay is minimum delay from input to output.

Glitch is a situation when the change of single input causes the change of multiple output. By handling critical path and using logic (K-map), glitch can be fixed, but it is impossible to fix all the glitches because simultaneous transitions on multiple inputs cause glitch.

2 Figures

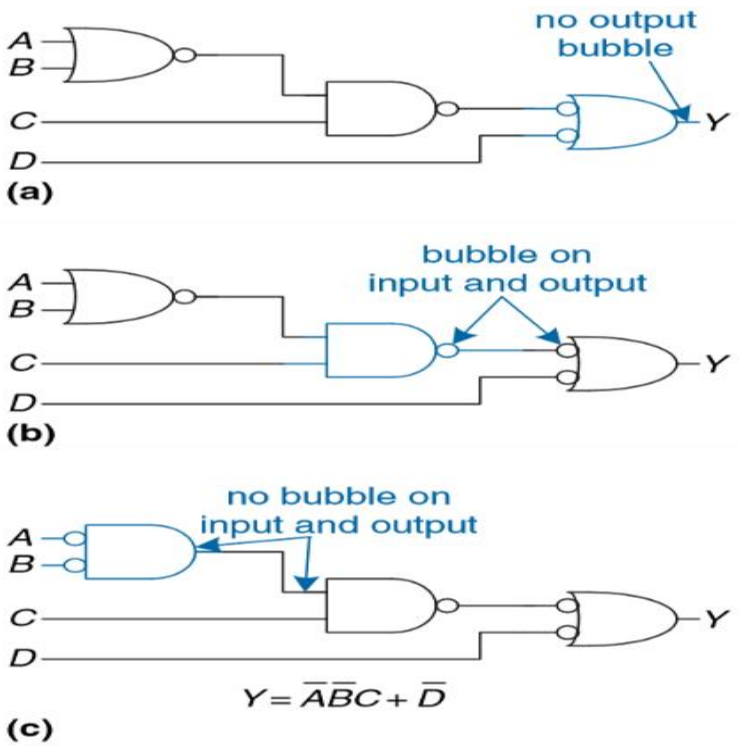


Figure 2.34 is about bubble pushing. To interpret the Boolean function of the circuit it is important to do bubble pushing. This figure shows the process of the bubble pushing thoroughly so that this figure is selected.

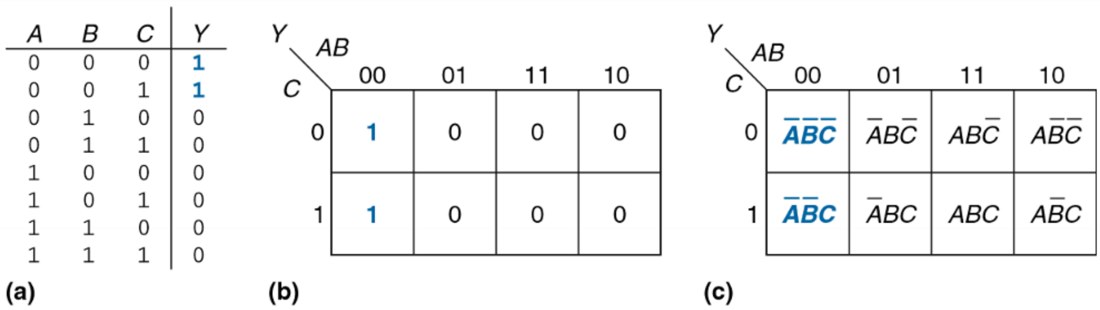


Figure 2.43 **Three-input function: (a) truth table, (b) K-map, (c) K-map showing minterms**

Figure 2.43 is about handling K-map with 3 inputs. This figure is dealing about the application of K-map. K-map is important to deal with combinational logic in much more complex system such as fixing glitch in the circuit. This is the reason why figure 2.43 is selected.

3 Example Problems

Ex1)

A	B	Y	Minterm	Maxterm
0	0	0	$\bar{A}\ \bar{B}$	$A+B$
0	1	1	$\bar{A}\ B$	$A+\bar{B}$
1	0	1	$A\ \bar{B}$	$\bar{A}+B$
1	1	0	$A\ B$	$\bar{A}+\bar{B}$

$$\therefore Y = \bar{A}B + A\bar{B}$$
$$= (A+B)(\bar{A}+\bar{B})$$
$$= \sum(1,2) = \pi(0,3)$$

Ex2)

$$\text{(a) } Y = \bar{A}BC + \bar{A}B\bar{C} = \bar{A}B(C+\bar{C}) = \bar{A}B$$

$$\text{(b) } Y = \overline{ABC} + \bar{A}B = A+B+C + A\bar{B} = A(1+\bar{B})+B+C$$
$$= A+B+C$$

$$\text{(c) } Y = AB\bar{C}D + A\overline{BCD} + \overline{(A+B+C+D)}$$

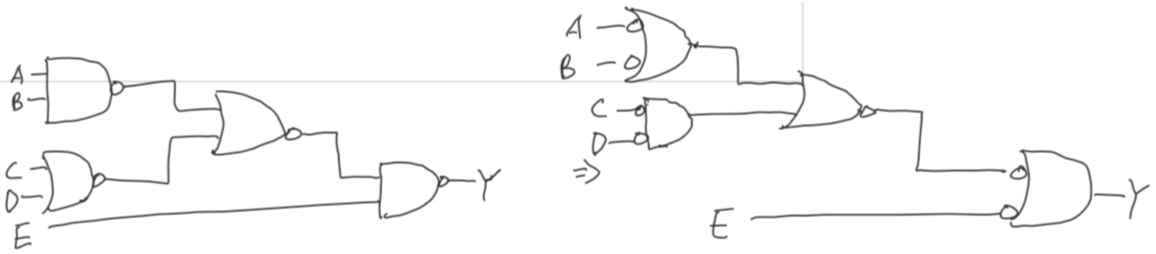
$$= AB\bar{C}D + A(\bar{B}+\bar{C}+\bar{D}) + \bar{A}\bar{B}\bar{C}\bar{D}$$

$$= AB\bar{C}D + A\bar{B} + A\bar{C} + A\bar{D} + \bar{A}\bar{B}\bar{C}\bar{D}$$

$$= AB\bar{C}D + A\overline{BCD} + \bar{A}\cdot\overline{BCD}$$

$$= \overline{BCD} + AB\bar{C}D$$

Ex3)



$$Y = \left(\overline{(\bar{A}+\bar{B}) + \bar{C}\bar{D}}\right) + \bar{E} = AB(C+D)+\bar{E}$$

Ex4)

A	B	C	D	Y
~~0~~	~~0~~	~~0~~	~~0~~	~~X~~
0	0	0	1	0
~~0~~	~~0~~	~~1~~	~~0~~	~~X~~
~~0~~	~~0~~	~~1~~	~~1~~	~~X~~
0	1	0	0	0
0	1	0	1	0
~~0~~	~~1~~	~~1~~	~~0~~	~~X~~
~~0~~	~~1~~	~~1~~	~~1~~	~~X~~
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
~~1~~	~~0~~	~~1~~	~~1~~	~~X~~
~~1~~	~~1~~	~~0~~	~~0~~	~~X~~
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

CD \ AB	00	01	10	11
00	X	0	0	X
01	0	X	1	1
10	X	0	0	1
11	X	X	X	1

i) $A\bar{B}\bar{C}D + AB\bar{C}D$

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

ii) $AB\bar{C}D + ABC\bar{D} + ABCD$

$= AB(\bar{C}D + C\bar{D} + CD) = AB(\bar{C}D + C)$

$= ABC + AB\bar{C}D$

iii) $ABC + A\bar{C}D + AB\bar{C}D = ABC + A\bar{C}D(1+B)$

$= ABC + A\bar{C}D = A(BC+\bar{C}D)$

$\therefore Y = A(BC+\bar{C}D)$

4 Glossary

1. Axiom:

   Noun: 1 A self-evident truth that requires no proof.

   2 a universally accepted principle or rule.

   3 a proposition that is assumed without proof for the sake of studying the consequences that follow from it.

2. Idempotent

   Adjective: unchanged when multiplied by itself.

   Noun: an idempotent element.

3. Associative

   Adjective: (Mathematics, logic)

   (Of an operation on a set of elements) giving an equivalent expression when elements are grouped without change of order.

   Having reference to this property.

4. Consensus

   Noun: majority of opinion, general agreement or concord; harmony

5. Involution

   Noun:

   An act or instance of involving or entangling

   The state of being involved

   Something complicated

6. Covering

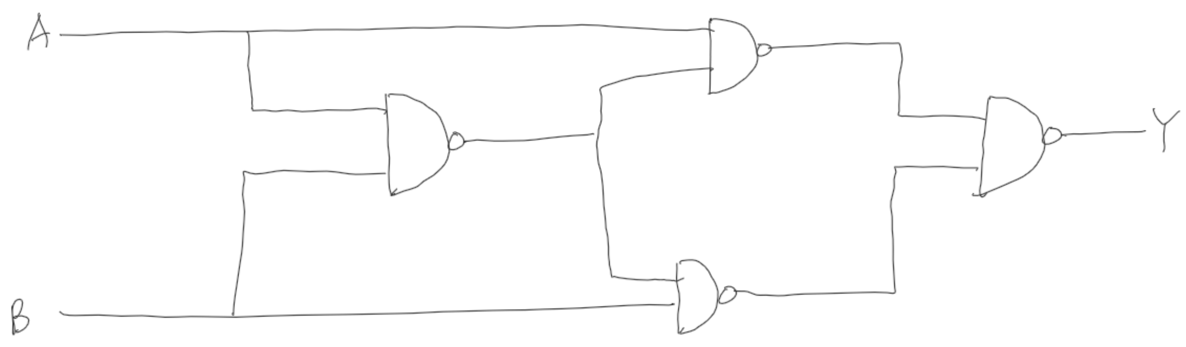
   Noun:

   Something laid over or wrapped around a thing, especially for concealment, protection warmth.

   (Math.) a collection of sets having the property that a given set is contained in the union of the sets in the collection.

5 Interview Question

Question 2.1 Sketch a schematic for the two-input XOR function using only NAND gates. How few can you use?



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Used 5 NAND Gates.

6 Reflection

The easiest section was Boolean equation and Boolean algebra because these topics were covered in my past math course. However, I felt difficulties in tristate, combinational building blocks and timing. I did not have prior experience with these topics, but I will try my best to obtain knowledge about these topics. In the textbook, there are logic gates. As I know these gates are made by semiconductors. I can relate how the logic gate works with physic course knowledge. Furthermore, I learned that how the electrical engineers solved the delay in our everyday electrical devices such as computer and television.

7 Questions

1. How do we implement tristate by using basic logic gates?
2. How was glitch first discovered?
3. Can we cover more about tristate?