ECE271, Chapter 2 Reading Report

WeiHao Kuang

April 16, 2018

1 Chapter Outline

1. Introduction

   In the world of digital circuits, there two categories that digital circuits can fall under: combinational or sequential. A combinational circuit is a circuit that takes the current inputs of a to compute an output; one of the key features that combinational circuits have is that they are memoryless and act independently of time. An example of a combinational circuit are logic gates where the output is determined by the inputs. Sequential circuit are the circuits that have RAM (random access memory) and these circuits also have their outputs dependent on time and sequence, hence the name sequential. Examples of a sequential circuit would be a shift register or a sequential logic counters.

2. Boolean Equations

   Boolean equations deal with only true and false, since we are working with logic of 1's and 0's the binary aspect of booleans are very appealing to use. Boolean equations still have to follow order of operations, but the the order of precedence is a little bit different compared to the traditional PEMDAS (parenthesis, exponents, multiple, divide, add, and subtract), in boolean order of operation NOT has the highest precedence, followed by AND and then OR.

   $$\overline{A}B + B\overline{C} = ((\overline{A})B) + (B(\overline{C})) \tag{1}$$

   When a boolean equation is written in the form that is displayed above where it is a sum of two products it is know as a *sum-of-products canonical form.* With the sum-of-products form a boolean equation can be formed from any truth table regardless of the number of variables.

3. Boolean Algebra

   Like algebraic equations boolean equations can be simplified using theorems and axioms. The algebraic rules that are used for boolean equation are very similar to the rules that are used in traditional algebra. The reason for simplifying down equations is to reduce the number of gates used to physically implement a function, thus making it smaller, cheaper, and possibly faster. Reducing the amount of gates will help the designer save resources for other parts of the project that they need gates (transistors) for. As mentioned in section 2 the orders or operation are very similar to regular algebra, but there are some rules that define the difference between the algebra and boolean algebra.

4. From Logic to Gates

   When drawing schematics of gates there are specific standards that the designer should taking into account, such as having inputs are on the left (or top) side of a schematic and outputs to the right (or bottom) side. Also whenever possible, gates should flow from left to right. There a more conventions to follow but the main thing is when it comes to design as designer should keep them clean and consistent for better readability and the debugging.

5. Multilevel Combination Logic

   In digital circuits often times are many levels of logic that are connected to one another by connecting the output of logic gate to the input input/s of another, by doing this a designer can create multi-layered logic circuits. A tool that is very useful when it comes to designing multi-layered circuits is some called bubble pushing. Bubble pushing when done right can solve for the boolean equation from a circuit schematic. Even when equipped with the right tools sometimes designers still have to face the pros and cons of each thing that they decide to put into their circuits, since benefits to once areas will most likely result in a hindrance for another constraint of design.

6. X's and Z's, Oh My

   The X's and Z's that are used in the digital circuit design have special meanings to them, in the circuit world X and Z values represent the illegal and floating values (or high impedance) respectively.

7. Karnaugh Maps

   Karnaugh maps is a method of logic simplification. Ultimately, the goal is to find a low-cost method of implementing a particular logic function. The way like the the boolean algebra this method finds a simple equivalence to a boolean function, but instead of theorems they method to simplify used graphical method of simplifying down the boolean equations.

8. Combinational Building Blocks

   When it comes to making a more complex logic system there is a level of abstraction that is involved to the making the more complex; since often times the more complex system is built from smaller combinational logic blocks. Among the types of combinational circuits, a few stick out: multiplexers and decoders.

9. Timing

   Speed and efficiency is another one of the main concerns when designing a circuit. The time it takes to have an output from a changing input is very important when it comes to managing the speed of a circuit. The logic of any given combinational circuit is characterized by their propagation delay and contamination delay. Where propagation delay is the max time it takes for an output to final its values after the input has been made. The contamination delay is the min time it take for an output to start changing after input. There is a another factor that can affect the timing of combinational logic as well and the thing would be glitch where an single input can have multiple output, which in the long run will elongate the critical path slowing down the overall circuit.

10. Summary

    In this chapter we learned about what combinational circuits are and the many elements that are involved with with analyzing them, we learned how to used boolean algebra and k-maps to help better understand circuits and simplify equations. With that we learned about the detailed workings such as the timing for combinational logic and how to

2 Grey Box Exploration

1. The first blurb is on page 63, as stated in the blurb *Augustus De Morgan was an English mathematician that was born in India and had a blind eye. At age 22 De Morgan was appointed Professor of Mathematics in London University. De Morgan Wrote widely on many mathematical subjects, including logic, algebra, and paradoxes. De Morgan's crater on the moon is named for him. He proposed a riddle for the year of his birth: "I was x years of age in the year* x^2*."*

   His best know works in algebra, he was also the discoverer of relation algebra, here are some of his best known works, *Trigonometry and Double Algebra* publish in 1849, *Budget of Paradoxes*, *Syllabus of a Proposed System of Logic*, De Morgan also framed the *De Morgan Laws* and was the creator of the term *mathematical induction.* [1]

   In one life time one can only do so much De Morgan, retires from the field and later dies at an age of 65 in London and leaves a great mathematical legacy. [1]

2. The second blurb is on page 90, timing of a circuit is concerned, and the variable the affects the timing in this case is wiring it a circuit, *Although we are ignoring wire delay in this analysis, digital circuits are now so fast that the delay of long wires can be as important as the delay of the gates. The speed of light delay in wires is covered in.* One of the issues when

account for wiring delay in the real world is there is not a specific number that goes with how much wire will be used, so there are often estimations that need to be made in the design process of a circuit so that the timing of the output desired is thrown off by the delay. The cause for wire delays in circuits is the transmission process of the electrons through the wires, in wire there is a set amount of resistance due to wire not being perfect.

$$R = \rho l/A \tag{2}$$

From the equation we can see that a wire with a fixed resistivity ρ and area A as the wire gets longer the resistance grows which slows down signals in the wires, leading to delays in circuits. [2]

3 Figures

Two figure were choosing for because of their effectiveness at explaining a concept to an audience. Figure 2.34 was selected because it highlights how bubble pushing a circuit is a very effective and intuitive way to redraw a circuit to show the layers or logic that in the circuit.

Figure 2.34 Bubble-pushed circuit

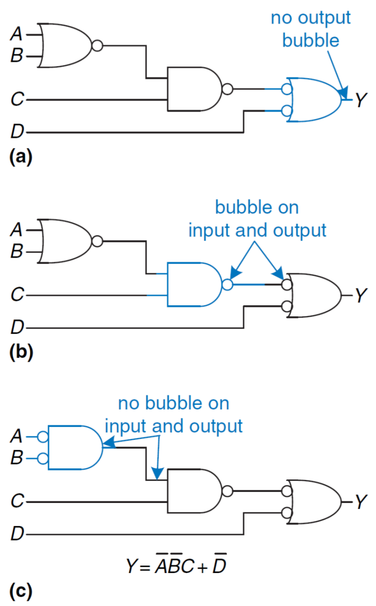


Figure 1: Bubble Pushed Circuit

Figure 2.43 was selected because shows an way to simplify a boolean equation without having to remember certain theorems and laws that are associated with boolean algebra. Using k-maps to simplify boolean equations is also great way to visualizing the minimizing process.

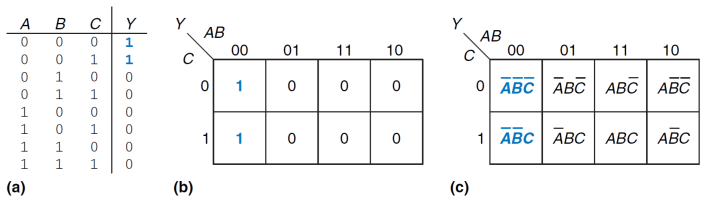


Figure 2.43 **Three-input function: (a) truth table, (b) K-map, (c) K-map showing minterms**

Figure 2: Karnaugh Maps with minterms, and Truth Table

4 Example Problems

There will be 16 example problems, and they will be attached towards the end of the this report.

5 Glossary

These terms where all obtained from Google, by using there definition searcher.

1. canonical:

   adjective:

   1. according to or ordered by canon law.
   "the canonical rites of the Roman Church"

   2. included in the list of sacred books officially accepted as genuine.
   "the canonical Gospels of the New Testament"

   3. relating to a cathedral chapter or a member of it.
   "Cardinal Bea in full canonicals"

   noun:

   1. the prescribed official dress of the clergy.
   "Cardinal Bea in full canonicals"

2. Axiom:

   noun:

   a statement or proposition that is regarded as being established, accepted, or self-evidently true.
   "the axiom that supply equals demand"
   synonyms: accepted truth, general truth, dictum, truism, principle;
   (mathematics) a statement or proposition on which an abstractly defined structure is based.

3. Combination:

   noun:

   1. a joining or merging of different parts or qualities in which the component elements are individually distinct.
   "a combination of blackberries, raspberries, and rhubarb"
   synonyms: amalgamation, amalgam, merger, merging, blend, mixture, mix, fusion, marriage, coalition, integration, incorporation, synthesis, composite;

   2. a sequence of numbers or letters used to open a combination lock. "a combination briefcase"

   3. (mathematics) a selection of a given number of elements from a larger number without regard to their arrangement.

4. Contamination:

   noun:
   the action or state of making or being made impure by polluting or poisoning.
   "the risk of contamination by dangerous bacteria"

5. Propagate

   noun:

   1. breed specimens of (a plant, animal, etc.) by natural processes from the parent stock.
   "try propagating your own houseplants from cuttings"
   synonyms: breed, grow, cultivate

   2. spread and promote (an idea, theory, etc.) widely.
   "the French propagated the idea that the English were violent and gluttonous drunkards"
   synonyms: spread, disseminate, communicate, make known, promulgate, circulate, broadcast, publicize, proclaim, preach, promote;

3. (with reference to motion, light, sound, etc.) transmit or be transmitted in a particular direction or through a medium.
"electromagnetic effects can be propagated at a finite velocity only through material substances"

6. Glitch

   verb:

   1. suffer a sudden malfunction or irregularity.

   noun:

   1. a sudden, usually temporary malfunction or irregularity of equipment.
   "a draft version was lost in a computer glitch"

   2. an unexpected setback in a plan.
   "this has been the first real glitch they've encountered in a three months' tour"

6 Interview Question

Question 2.3 What is a tristate buffer? How and why is it used?

Figure 3: A tristate buffer?

First off I would like to describe what a buffer is a what uses it has, so a buffer it the part in a circuit that can produce an output which matches the input. This process is especially useful when the designer wants certain parts of the circuit isolated from another part. Another use that buffers have are to drive up the power of circuits since their output drive capability is generally much higher than their input signal requirements. [3]

Now that I have defined and explain what buffers are and what they are used for; I would like to talk about the tri-state buffer this buffer like the a regular buffer can isolate circuits from each other, but in the case of a tri-state buffer there are 2 inputs instead of just one. The two inputs being the regular input, and the switch input that controls that switch inside the buffer, this way the output of the buffer can be turn off or on. Tri-state buffers can have up to three outputs: 0, 1, High Impedance (High-Z). The way the switch of a tri-state works is by sending a 1 signal to the switch which enable the switch producing an output either inverted or non inverted, or the switch signal can be sent a 0 which would turn off the output and produce a HIGH-Z. A tri-state buffer can be used to invert inputs and outputs and the also amplify outputs as well. Tri-state buffers are used because they allow multiple logic devices to be connected to the same wire or bus without damage or loss of data, due to contention.[3]

7 Reflection

In chapter 2, a lot of things were discussed, The most beneficial things that I took away from the chapter was a good introduction to the elements that in are in combinational logic gates and devices. The boolean equations section of the chapter gave me a good review of the discrete math that is involved with digital logic, when I read through it, I remembered lots of things and I also had lots of questions that involved the more complex equations and how to turn them into a truth table and find logical equivalences, so I took some time to look some things up in my old discrete textbook and find out logic behind some equations. I had trouble converting a truth table into a product and sums equation, and lo behold a little reading into my old textbook had a problem done a discussed that was very similar to it. One of the things that I found pretty interesting was the tools that are used to analysis the combination logical circuits such as the k-maps section of the chapter. Once I understood the concept of the k-maps I compared it to the boolean algebra, and let me tell you one thing the satisfaction of simplifying an equation using k-maps in incomparable to that of the using traditional boolean algebra. One of the things that I had trouble understanding was the timing, not very much so the concept of time delay, but the calculations and how the book derived some of the answers. I also loved reading about the multiplexers and the section on the segment displays, it got me thinking about the lab that coming up in 272, I was thinking that I should probably read up and the segment displays since we are going to have a 2 week lab on it, this is one of the goal that I have set for myself before I turn in reading report 2. Another thing that I had trouble with visualizing glitches, I know they slow down the circuit by elongating the critical path the a circuit takes, but I fail to come up with a real life example of it. Other than the issues I talk about I thought the chapter was a good insight on combinational logic, I really appreciate the book splitting up the combinational and sequential circuits and giving them a chapter each.

8 Questions for Lecture

1. What is a real world example of a glitch in circuits, and how will the issue be worked around in the circuit; what measures are taken to prevent glitches in circuits?
2. 3 input k-maps are understandable me, but what of k-maps that have 4 or more inputs, do we do those in the same way?
3. In terms of timing of circuits what are the ways that we can increase the speed of something without having to sacrifice functionality (complexity) and power (if possible).

References

[1] F. Authors, "Augustus de morgan." https://www.famous-mathematicians.com/augustus-de-morgan/, 2012.

[2] Hades, "1gate vs wire delay." https://www.youtube.com/watch?v=Cs-TGLxQfBM, 2005.

[3] E. Tutorials, "Digital buffer tutorial." https://www.electronics-tutorials.ws/logic/logic_9.html.

Example 2.3

Write an equation in product-of-sums form for

A	B	y	maxterm	maxterm Name
0	0	0	$A + B$	M_0
0	1	1	$A + \bar{B}$	M_1
1	0	0	$\bar{A} + B$	M_2
1	1	1	$\bar{A} + \bar{B}$	M_3

Since the circled rows are output false we can write

$$\boxed{y = (A+B)(\bar{A}+B)}$$

AND with "0" guaranteed "0"

Example 2.4

using De Morgans Theorm derive products-sums form of y from product of sum form of $\bar{y}$.

A	B	y	$\bar{y}$	minterm
0	0	0	1	$\bar{A}\,\bar{B}$
1	0	0	1	$\bar{A}\,B$
0	1	1	0	$A\,\bar{B}$
1	1	1	0	$A\,B$

$\bar{y} = \bar{A}\,\bar{B} + \bar{A}\,B$

$\bar{\bar{y}} = \overline{\bar{A}\bar{B} + \bar{A}B} = (\overline{\bar{A}\bar{B}})(\overline{\bar{A}B})$

$$\boxed{y = (A+B)(A\bar{B})}$$

p2

Example 2.8 Bubble pushing for cmos logic

use NAND and Nor to convert:

Figure 1

invert the first two NAND after replacing AND gates like so..

Figure 2.

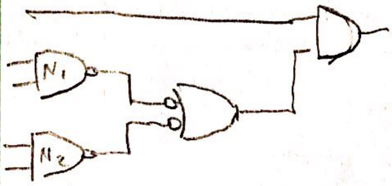


Figure 1 and Figure 2 are logically equivalent.

Example 2.13.

Truth table — 8:1 truth table

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

only y output of true.
can a 4:1 replace this?

ANswer is yes,

4:1 truth table

A	B	y
0	0	$\bar{C}$
0	1	C
1	0	1
1	1	0

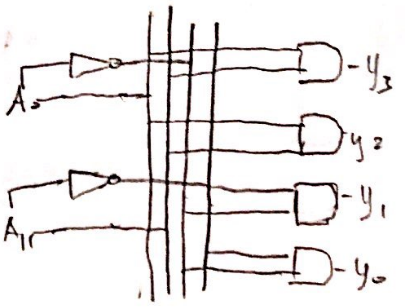
by shrinking down first 2 rows: and have $\bar{C}$ be y and shrinking the next 2 rows and having y be C. and then shrinking the next 2 rows and having y be 1. and then shrinking the last 2 rows and have them be 0.

Example 2.14

Implement a 2:4 decoder using AND, Not for gates

A_1	A_2	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

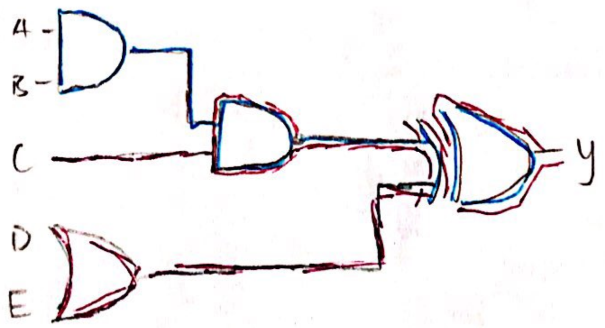
2:4 using Not and AND gates



Example
2.15

Find delays in This circuit

propogation of 100 picosecond
contamint of 60 nanoseconds



– blue . critical path,
≡ red . shortest path

The critical path is from A or be to y. Where there are 3 gates involved, since they are first there will be 300 pico senconds of delay because there are 3 gates.
For the shortest path. C, D or E only 2 gates passed (so logically I would think 200 ps). but why is the text book saying 120ps?