

Introduction to digital electronics

by Elza Antonova Hadzhieva

Submission date: 18-May-2023 01:31PM (UTC+0100)

Submission ID: 206764746

File name: 46097_Elza_Antonova_Hadzhieva_Introduction_to_digital_electronics_500362_1304514029.docx
(395.62K)

Word count: 2255

Character count: 11487



Digital Circuit Design

Executive Summary

Digital logic design is the process of creating electronic circuits that operate on digital signals. Digital circuits are used to perform a wide range of functions, from simple arithmetic operations to complex control systems. This assignment on digital logic circuit design has covered a range of topics, including logic gates, logic equation simplification, designing logic circuits in software, and sequential circuits. By completing this assignment, we have gained a deep understanding of how to design and implement digital logic circuits using various tools and techniques. The skills and techniques we learned in this module can be applied in real-world scenarios, as digital logic circuits are vital in the design and operation of electronic devices and systems.

Contents

Executive Summary	2
Introduction	4
Task 1: Symbol and truth table for logic gates	4
Task 2: Logic circuit design with software	6
Task 3: Logic circuit simplification and implementation with software	11
Task 4: Designing a 4-bit Synchronous counter using software	13
Task 5: Types and applications of Sequential circuits	16
Conclusion	18
References	19

Introduction

Digital electronics is an essential subject for students studying electronics and computer engineering. This field is concerned with the design and analysis of digital circuits that use binary digits (bits) to represent and manipulate information.

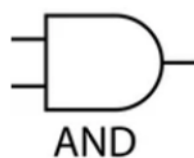
The process of designing electronic circuits that employ digital signals is known as digital logic design (Floyd, 2011). These signals comprise binary numbers, or bits, which can be either 0 or 1. Simple arithmetic operations to advanced control systems problems can be solved using digital circuits. The logical operations are performed using logic gates which are electronic devices and the fundamental blocks of digital electronics (Mano, 2017). AND, OR, and NOT are the three basic gates that can be combined into different configurations to perform different complex logic operations.

This assignment covers the fundamental concepts of digital logic circuit design, starting with logic gates and truth tables, then moving on to logic equation simplification and designing logic circuits using software tools. We will also delve into the topic of sequential circuits, which are digital circuits that rely on memory to perform their function. After completing this assignment, we will have a sound knowledge of how digital circuits are designed and analysed, both combinational and sequential, using a range of tools and techniques.

T5 1: Symbol and truth table for logic gates

(a) AND Gate

The logic circuit that performs logical AND operations is called an AND gate. It will produce a 1 if all of the inputs are in the logic 1 state; otherwise, it will produce a logic 0 (Widmer and Tocci, 2007). Figure 1 shows AND gate's logic symbol and truth table.



(a)

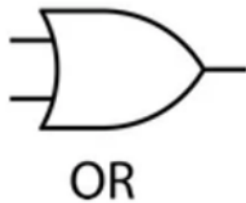
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

(b)

Figure 1: (a) logic symbol and (b) truth table of an AND gate

(b) OR Gate

The logic circuit that performs logical OR operations is called an OR gate. It will produce a 1 if one or both of the inputs are at the logic 1 state; it will only produce a logic 0 if both inputs are at the logic 0 state (Widmer and Tocci, 2007). Figure 1 shows OR gate's logic symbol and truth table.



(a)

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

(b)

Figure 2: (a) logic symbol and (b) truth table of an OR gate

(c) XOR Gate

An exclusive OR gate or an XOR gate is a complex logic circuit that performs mathematical exclusive OR operations. This logic gate will only produce a logic 1 or true output when only one of the inputs is 1. In all other cases, it will give a logic 0 or false output (Widmer and Tocci, 2007). The logic equation of an XOR gate is given below-

$$Y = A \oplus B = \bar{A}B + A\bar{B}$$

Figure 3 depicts the logic symbol and truth table of an XOR gate.



(a)

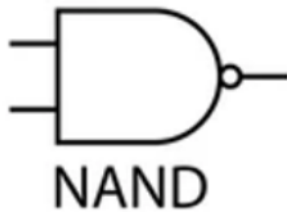
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

(b)

Figure 3: (a) logic symbol and (b) truth table of an XOR gate

(d) NAND Gate

A NAND gate is a logic circuit that performs the opposite operation of an AND gate. It will produce a 1 if one or all of its inputs are at the logic 0 state; otherwise, it will produce a logic 0 (Lee, 1976). Figure 4 depicts a NAND gate's logic symbol and truth table.



(a)

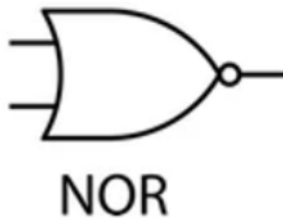
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

(b)

Figure 4: (a) logic symbol and (b) truth table of a NAND gate

(e) NOR Gate

A NOR gate is a logic circuit that performs the opposite operation of an OR gate. It will produce a 1 if all of its inputs are at the logic 0 state; in all other cases, it will produce a logic 0 (Widmer and Tocci, 2007). The logic symbol and the truth table of a NOR gate are given in Figure 5.



(a)

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

(b)

Figure 5: (a) logic symbol and (b) truth table of a NOR gate

Task Logic circuit design with software

(a) The given logic equation is

$$Y = (\overline{AB})C + \overline{A}BC + \overline{A} + C$$

The logic circuit for this expression is drawn using Proteus 8.9 software. Figures 6 to 9 show the circuit with different input combinations.

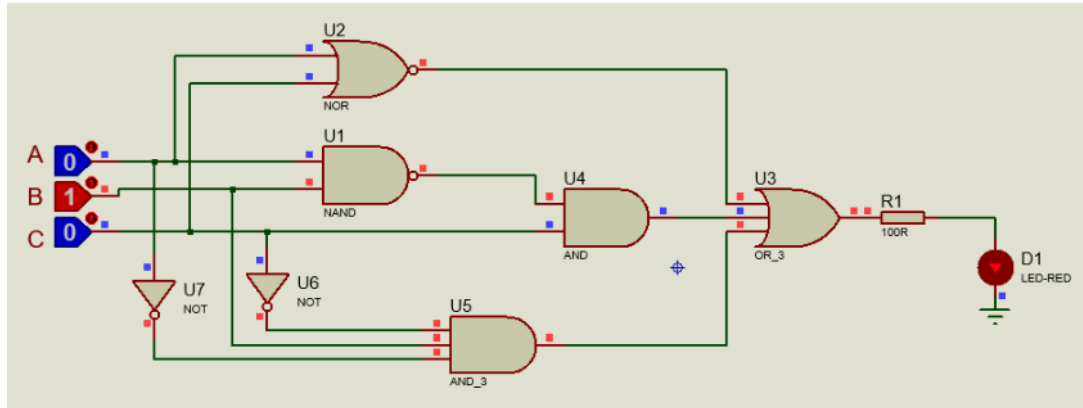


Figure 6: Output state for 010 input combination

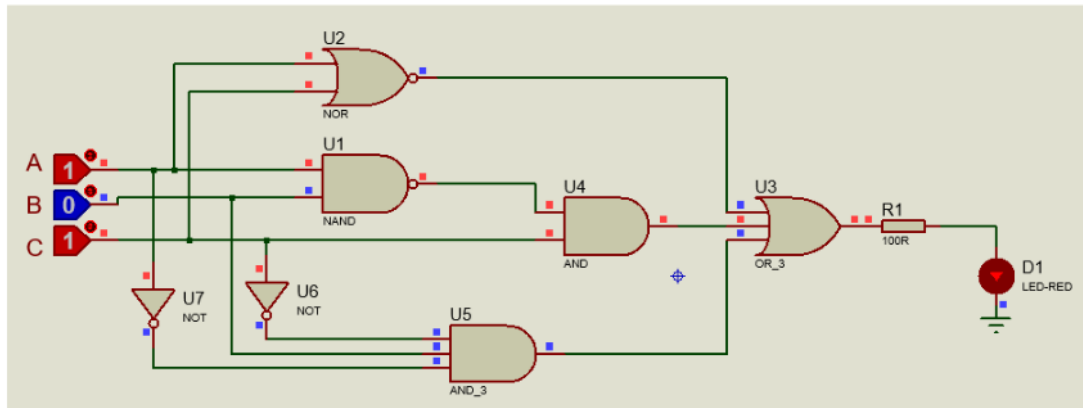


Figure 7: Output state for 101 input combination

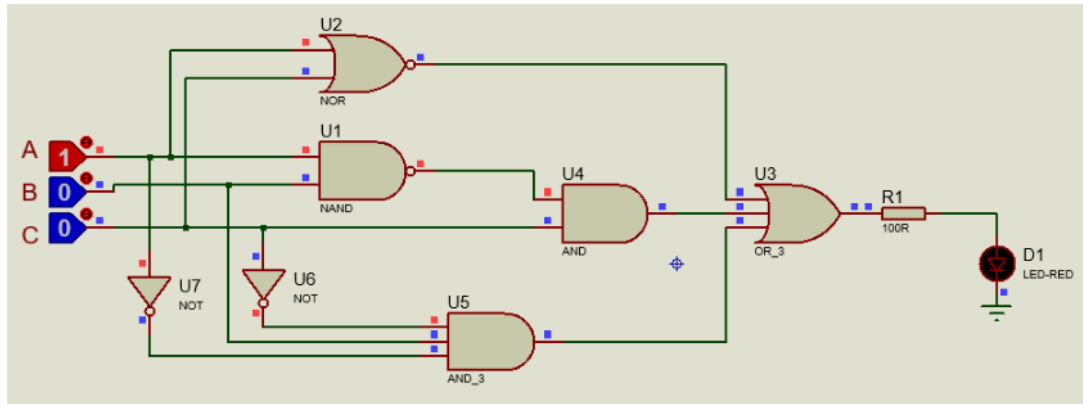


Figure 8: Output for 100 input combination

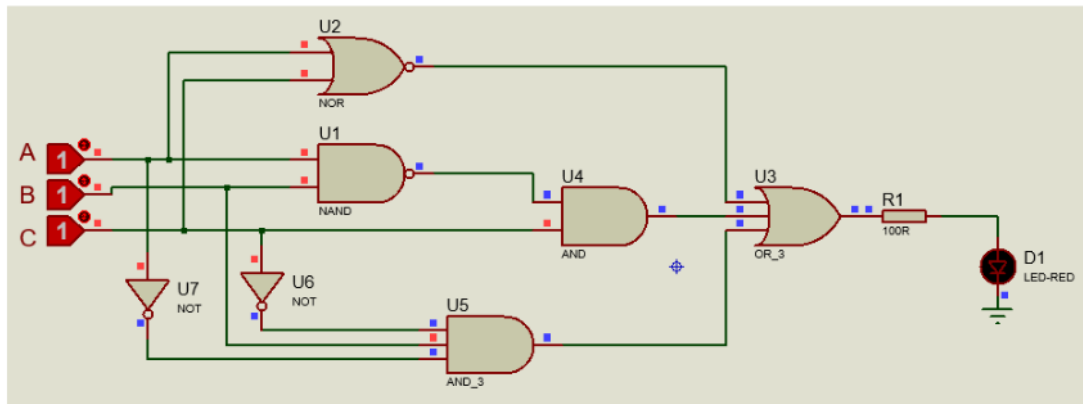


Figure 9: Output for 111 input combination

(b) The given logic equation is

$$Y(A, B, C, D) = \sum_m(0, 1, 5, 6, 9)$$

This equation can be simplified using K-map (Brown and Vranesic, 2005). After simplification, the expression reduces to

$$Y = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{C}D + \bar{B}\bar{C}\bar{D} + \bar{A}BC\bar{D}$$

The logic circuit for this expression is drawn using Proteus 8.9 software. Figures 10 to 13 show the circuit output with different input combinations.

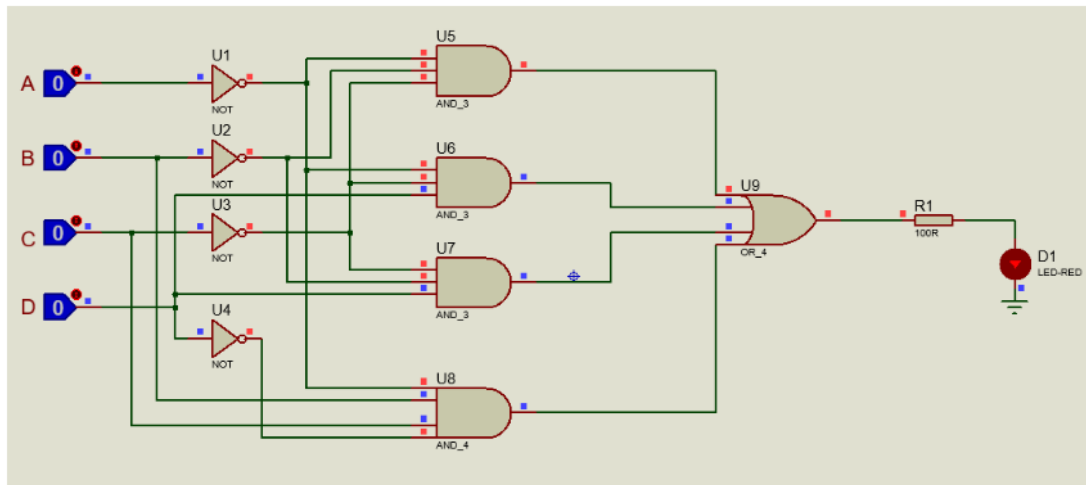


Figure 10: Output state at 1 (illuminated LED) for input combination 0000

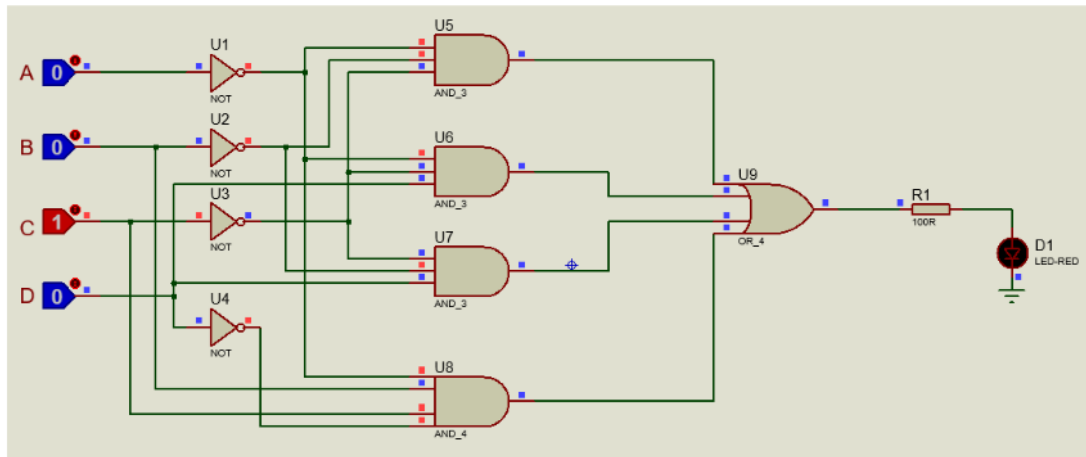


Figure 11: Output state at 0 for input combination 0010

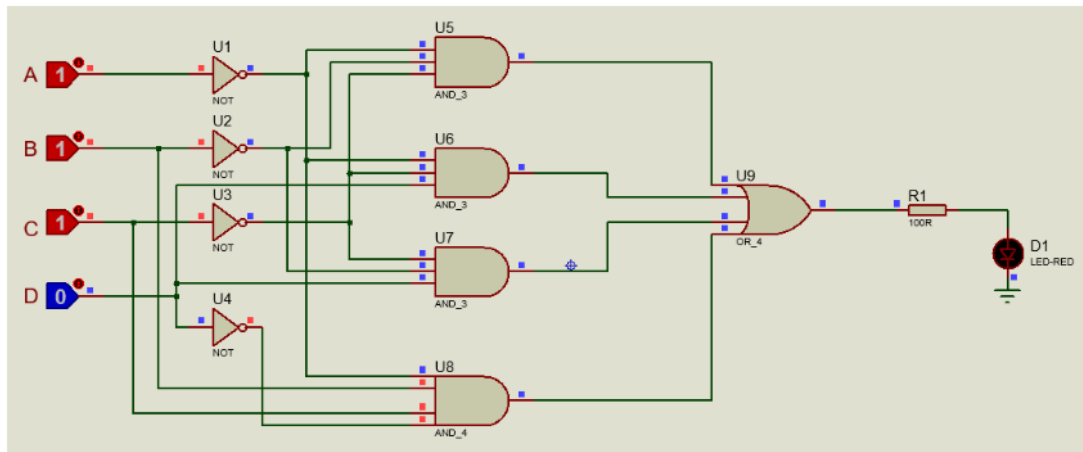


Figure 12: Output state at 0 for input combination 1110

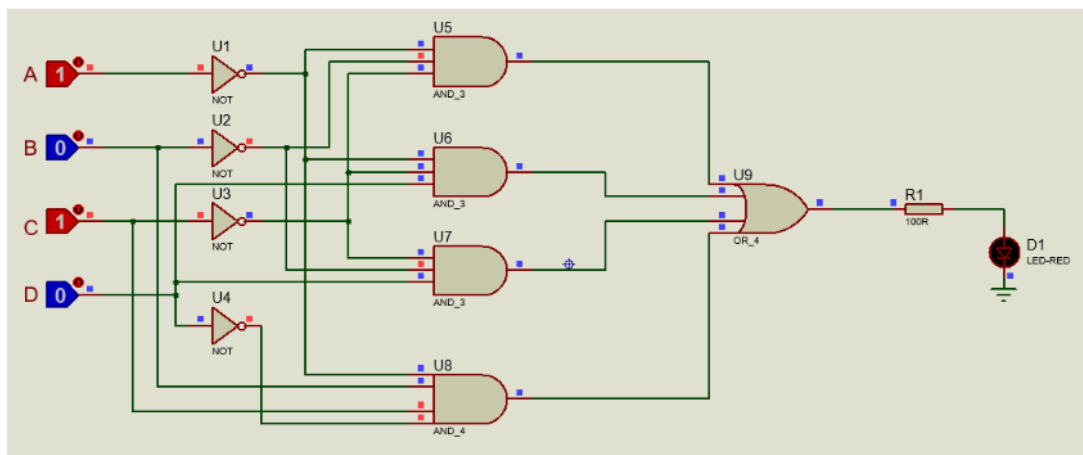


Figure 13: Output state at 0 for input combination 1010

Task 3: Logic circuit simplification and implementation with software

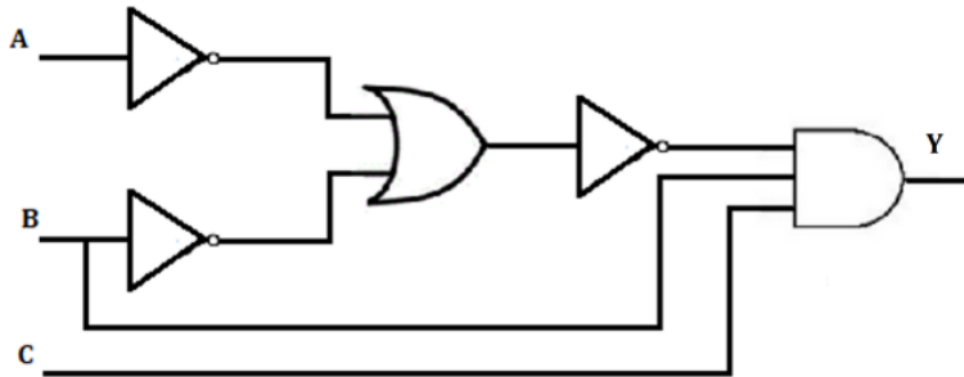


Figure 14: Given logic circuit

Figure 14 shows the given logic circuit. To simplify this, we first need to find out the logic expression for each node. The Boolean expression for this circuit is

$$Y = (\overline{A} + \overline{B}).B.C$$

Figure 15 shows the same circuit with logic expressions assigned to each node. Here, we applied de Morgan's law for simplification as follows-

$$(A' + B')' = A''. B'' = AB \text{ (Mano, 2017)}$$

At the AND gate of the last stage, we have

$$Y = AB.B.C = ABC$$

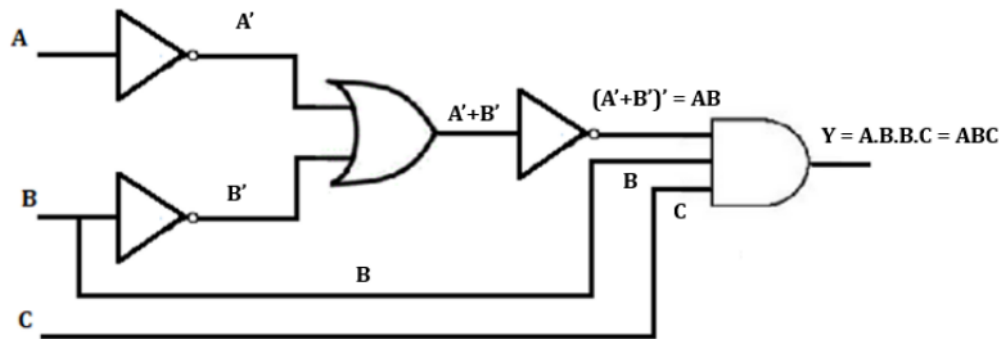


Figure 15: Circuit with logic expression for each node

The simplification of the given circuit gives the output of a simple three-input AND gate as shown in Figure 16.



Figure 16: Simplified logic circuit

Now, we will design this simplified circuit in Proteus software.

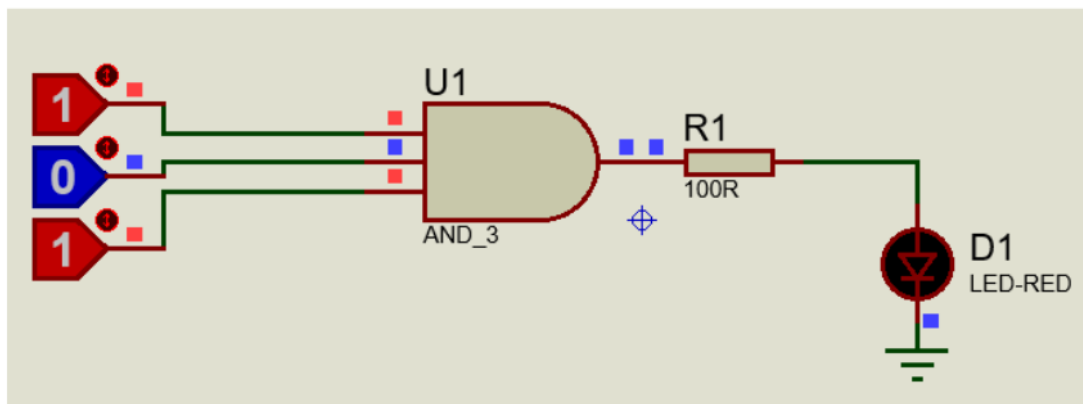


Figure 17: Circuit output at logic 0 state since one input is 0

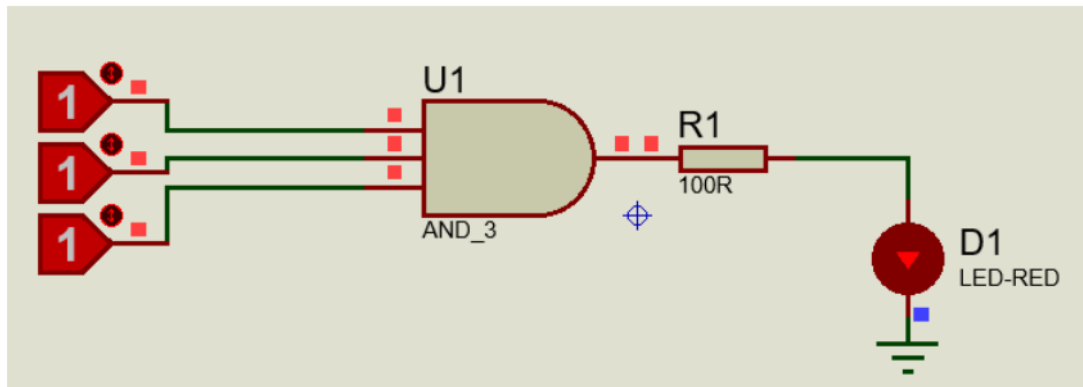


Figure 18: Circuit output at 1 state (illuminated LED) since all three inputs are at logic 1

The truth table for this circuit is given below-

Table 1: Truth Table for given logic circuit

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

Task 8 Designing a 4-bit Synchronous counter using software

A synchronous counter is a digital timing circuit that can count from 0 to a certain level and then resets back to 0, repeating the cycle. The maximum value a counter can count up to depends on how many bits are being used. For example, a 4-bit counter can count 0 to $(2^4 - 1) = 15$. Counters are normally designed using flip-flops and some additional elements. The term synchronous, means that all of the flip-flops that make up the counter change their states simultaneously after receiving a common clock signal.

A 4-bit counter consists of four flip-flops, each representing one bit of the counter value. The clock input of each flip-flop is supplied from a common clock signal. The output of the first flip-flop is supplied as the input of the second flip-flop, forming a ripple carry chain. The most significant bit (MSB) of the counter is the output of the last flip-flop while the output of the first flip-flop is the least significant bit (LSB).

To implement the counting sequence, the input of the first flip-flop is connected to a logic high (1) value, and the inputs of the other flip-flops are connected to a combination of the previous flip-flop's output and the clock signal, such that the counter increments by 1 on each clock cycle.

To reset the counter back to 0, all of the flip-flops are supplied with a reset signal at the clear pin. The clear input forces all the flip-flop outputs to be 0, resetting the counter to its initial state.

A synchronous counters are one of the basic building blocks of digital circuits, and it has many applications, such as frequency division, time-division multiplexing, and address generation in memory systems.

For this task, we will design a 4-bit synchronous up counter using Proteus 8.9 software.

- The counter is designed using 4 J-K flip-flops and 2 AND gates.
- A pulse generator is added to supply clock signals to the flip-flops.
- One toggle pin along with an SPST switch is added for resetting the flipflops. Closing the switch resets all the flipflops and counter resets to 0.
- Another toggle pin was added to supply inputs to the first J-K flip-flop.
- A seven-segment display was added for displaying the count values.

The following figures show the circuit diagrams from Proteus. Figure 19 shows the circuit in the OFF state. Figure 20 shows the circuit in operation and Figure 21 shows the circuit when it is reset to 0. In the current configuration, the counter can count from 0 to 9. If we add a two-digit seven-segment display, it will be able to show count values 0 to 15.

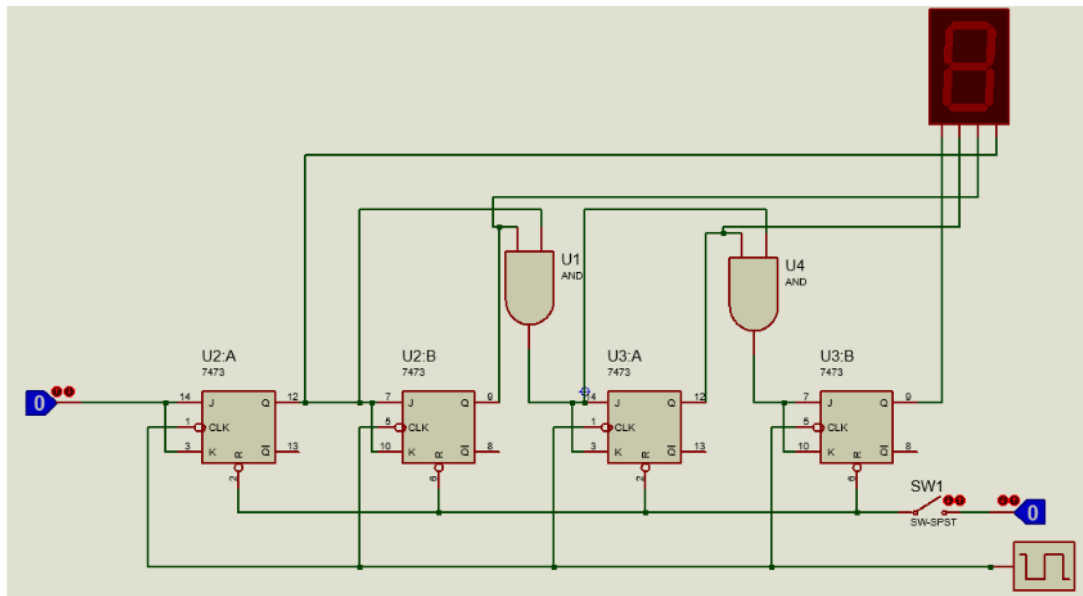


Figure 19: 4-bit Synchronous Up counter in OFF state

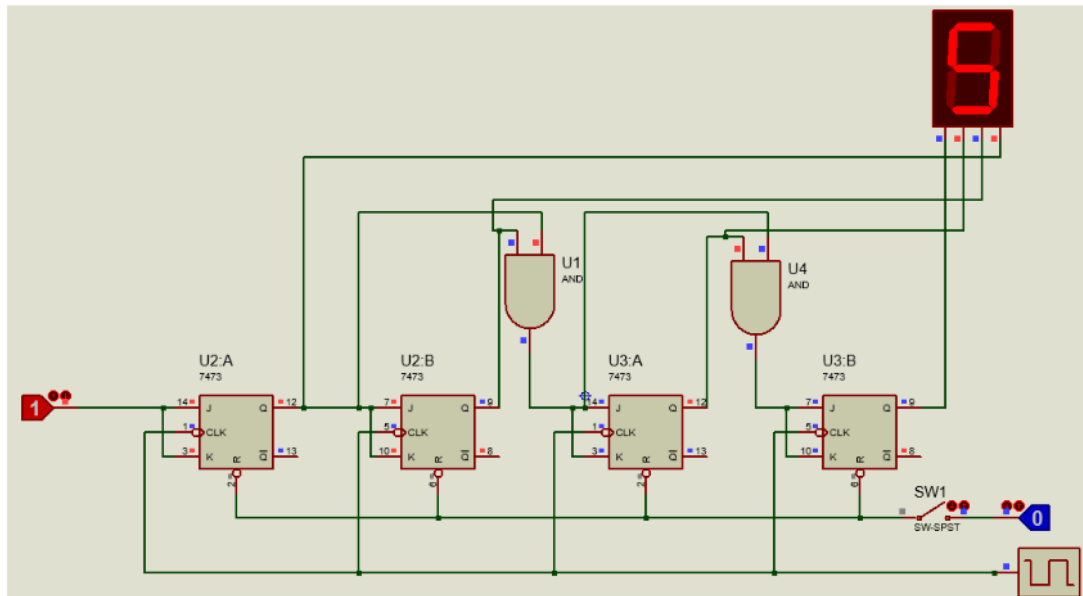


Figure 20: Counter in operation: the display shows 5

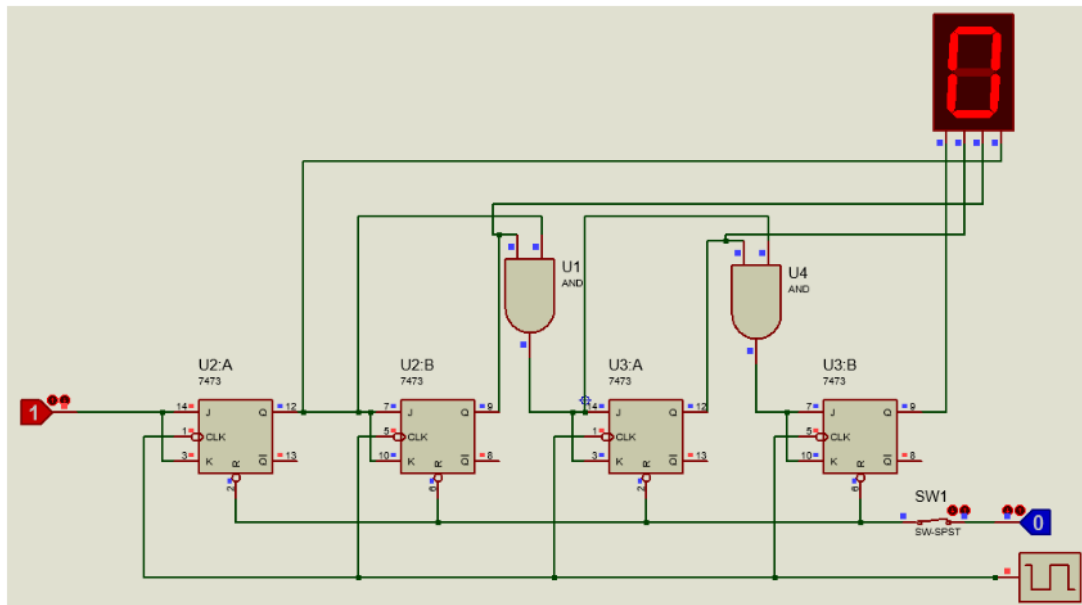


Figure 21: Counter reset to 0

Task 5: Types and applications of Sequential circuits

Sequential circuits are a type of digital circuit that incorporate memory elements to store data and generate output based on the current input and the previous state (Lee, 1976). Sequential circuits can be of two types:

1. Synchronous circuits
2. Asynchronous circuits

Synchronous circuits

Synchronous sequential circuits use a common clock signal for synchronising the operation of their memory elements. The clock signal acts as a timing reference and its job is to make sure that all the memory elements change state simultaneously, allowing for precise control of the circuit's behaviour, and making synchronous sequential circuits suitable for applications that require high-speed and complex data processing (Brown and Vranesic, 2005). Synchronous sequential circuits are commonly used in microprocessors, digital signal processors, and communication systems.

Asynchronous circuits

Asynchronous sequential circuits, on the other hand, do not rely on a clock signal and instead, use the input signals to trigger state changes in their memory elements. Asynchronous sequential circuits can operate at lower power and are simpler to design than synchronous circuits, but they are less reliable due to the lack of timing control (Harris and Harris, 2010). Asynchronous sequential circuits are commonly used in applications such as data transfer, control systems, and signal processing.

Sequential circuits are used for a range of applications requiring precise timing control (Floyd, 2011). Some of the applications of sequential circuits are as follows-

- **Counters:** Sequential circuits can be used to build counters that keep track of the number of events that occur. Counters can be implemented as either synchronous or asynchronous circuits depending on the application requirements.
- **Shift Registers:** Shift registers are sequential circuits that are used for serial data transfer, storage, and conversion. They can be implemented using either synchronous or asynchronous circuits.
- **State Machines:** Sequential circuits can be used to implement state machines that respond to a sequence of input signals and produce a corresponding sequence of output signals. State machines can be used in a variety of applications such as control systems, vending machines, and traffic lights.
- **Communication Systems:** Sequential circuits are used in communication systems such as wireless networks and satellite communication systems to process and transmit data.
- **Data Transfer:** Asynchronous circuits are used for data transfer applications such as serial data transmission and asynchronous communication protocols.
- **Control Systems:** Asynchronous circuits are used in control systems such as robotics and automation systems, where precise timing and control are required.

Conclusion

In conclusion, this assignment on digital logic circuit design has covered a range of topics, including logic gates, logic equation simplification, designing logic circuits in software, and sequential circuits. By completing this assignment, we have gained a deep understanding of how to design and implement digital logic circuits using various tools and techniques.

Through this assignment, we have learned how to simplify complex logic equations using Boolean algebra and Karnaugh maps, design digital logic circuits using software tools such as Proteus, and implement sequential circuits using flip-flops.

The skills and techniques we learned in this module can be applied in real-world scenarios, as digital logic circuits are vital for the design and operation of electronic devices and systems. Overall, this assignment has provided us with a solid foundation in digital logic circuit design and equipped us with the necessary skills to pursue more advanced topics in this field.

References

- Brown, S. and Vranesic, Z., 2005. Digital Logic with VHDL Design. *International Edition on*, pp.292-293.
- Floyd, T.L., 2011. *Digital fundamentals, 10/e*. Pearson Education India.
- Harris, D. and Harris, S.L., 2010. *Digital design and computer architecture*. Morgan Kaufmann.
- Lee, S.C., 1976. *Digital circuits and logic design*. Prentice Hall PTR.
- Mano, M.M., 2017. *Digital logic and computer design*. Pearson Education India.
- Widmer, N.S. and Tocci, R.J., 2007. *Digital systems: Principles and applications*. Prentice Hall. Pearson Education International.

Introduction to digital electronics

GRADEMARK REPORT

FINAL GRADE

83 / 100

GENERAL COMMENTS

Instructor

Engagement with Literature Skills: Marks: 0/0

Knowledge and Understanding Skills: Marks: 22/25

It is evident through task 1 to 5, there is excellent knowledge and understanding of the basic underlying concepts and principles of the Digital Electronics Module. The output functions for each combinations of input has been provided in the report.

Cognitive and Intellectual Skills: Marks: 20/25

Good evaluation of the underlaying concepts and principles is evident through the truth tables provided and a brief description has been provided for each concepts of the tasks.

Practical Application Skills: Marks : 38/45

An advance application of a range of taught, basic methods, materials, tool and techniques are used. A very good understanding and appropriate knowledge to apply the suitable tools in Digital Electronics module is evident. The screenshot of the outputs for various input combinations are provided for task 1 to 4

Transferable Skills for Life and Professional Practice: Marks : 3/5

Work is accurate, coherent, fluent, well-structured and organised. Work is properly presented with a continuation in flow of information. The title pages, table of content, figure name and numbers were provided. The page number is missing in this report.



Comment 1

Task 1: 17/20

Task 2: 25/30

Task 3: 17/20

Task 4: 17/20

Task 5: 7/10

Total: 83/100



Comment 2

Good Attempt, the first page or the title page of the report should have the name, student ID, Module Name, Module code, Title of the Coursework, programme of study. Most of these are in the title page, but however few are missing.



Comment 3

Good



Comment 4

It is better to have page number

PAGE 4



Comment 5

Good Work, The logic gates and the respective truth tables are correct. The figures and tables are named and number. It would have been better if outputs were shown for various combinations of inputs in the logic gates.

Task 1: 17/20

PAGE 5

PAGE 6



Comment 6

Good Attempt, the design part is good. It is evident, that you can design circuits using software,. the outputs for various combinations of inputs are shown in the circuit.

Task 2: 25/30

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11



Comment 7

The simplified Boolean expression is correct. The truth table for the simplified circuit is correct.

Task 3: 17/20

PAGE 12

PAGE 13



Comment 8

The design of circuit diagram is good. A brief description has been provided The truth table is missing

Task 4: 17/20

PAGE 14

PAGE 15

PAGE 16



Comment 9

Good Description, The different types of sequential circuits, and the applications of sequential circuits are mentioned. However, it is better to describe with necessary diagrams and references.

Task 5: 7/10

PAGE 17

PAGE 18

PAGE 19
