

# Digital Logic Circuits Lab Manual

**Dr. Zafar Q. Taha**

Assistant Professor

Department of Electrical Engineering

College of Engineering

Al-Imam Muhammad Ibn Saud Islamic University.

February 3, 2013

## **Abstract**

# Contents

<b>1</b>	<b>Experiment 1: Digital Logic Gates</b>	<b>3</b>
1.1	Basic Logic Gates . . . . .	3
1.2	Alternate Construction of Basic Gates . . . . .	4
1.3	Simple Digital Circuit . . . . .	5
1.4	DeMorgan's Theorems . . . . .	5
1.5	Dual of a function . . . . .	5
<b>2</b>	<b>Experiment 2: Boolean Algebra and Functions</b>	<b>9</b>
2.1	Postulates and Theorem of Boolean Algebra . . . . .	9
2.2	Logic diagram and Boolean expression . . . . .	11
2.3	Logic diagram from sum of minterms . . . . .	12
2.4	Short Theory Questions . . . . .	13
<b>3</b>	<b>Experiment 3: Combinational Circuits</b>	<b>14</b>
3.1	Analysis of Combinational Circuit . . . . .	14
3.2	Design from Truth Table . . . . .	17

3.3	Majority Logic . . . . .	19
3.4	Parity Generation . . . . .	21
3.5	Short Theory Questions . . . . .	23
<b>4</b>	<b>Code Converters</b>	<b>24</b>
4.1	Gray Code to Binary Code . . . . .	24
4.2	9's Complementer . . . . .	26
4.3	7-Segment Display . . . . .	28
4.4	Short Theory Questions . . . . .	29
<b>5</b>	<b>Multiplexers and Decoders</b>	<b>30</b>
5.1	4 × 1 Multiplexer using Basic Gates . . . . .	30
5.2	Design Service Window Display . . . . .	32
5.3	Implementing Logic Circuits using Decoders . . . . .	33
5.4	Short Theory Questions . . . . .	35
<b>6</b>	<b>Adders and Subtractors</b>	<b>36</b>
6.1	Half Adder . . . . .	37
6.2	Full Adder . . . . .	38
6.3	Parallel Adder and Subtractor . . . . .	39
6.3.1	Parallel Adder . . . . .	39
6.3.2	Parallel Subtractor . . . . .	39
6.4	Magnitude Comparator . . . . .	40
6.5	Short Theory Questions . . . . .	41

<b>7</b>	<b>Experiment 7: Latches and Flip-Flops</b>	<b>42</b>
7.1	SR Latch . . . . .	43
7.2	D Latch . . . . .	44
7.3	Edge-Triggered D Flip-Flop . . . . .	45
7.4	Analyze a Sequential Circuit . . . . .	46
7.5	Short Theory Questions . . . . .	48
<b>8</b>	<b>Sequential Circuits</b>	<b>49</b>
8.1	Up-Down Counter with Enable . . . . .	49
8.2	Design using State Diagram . . . . .	51
8.3	Counter with specified sequence . . . . .	53
8.4	Short Theory Questions . . . . .	54
<b>9</b>	<b>Counters</b>	<b>55</b>
9.1	Ripple Counter . . . . .	56
9.2	Synchronous Counter . . . . .	57
9.3	Decimal Counter . . . . .	58
9.4	Short Theory Questions . . . . .	60
<b>10</b>	<b>Shift Registers</b>	<b>61</b>
10.1	Construct a Shift Register . . . . .	62
10.2	Ring Counter . . . . .	63

# Introduction

## Why digital systems?

The present technological period is referred to as the *digital age* and digital systems have an important role. Digital systems are used in communication, business transactions, traffic control, space guidance, medical treatment and many other applications. We have digital telephones, digital television, digital versatile discs, digital cameras, and, of course digital computers. One characteristic of digital systems is their ability to represent and manipulate discrete elements of information. Discrete elements of information are represented in a digital system by physical quantities called signals. The signals in most present-day digital systems use just two discrete values and are therefore said to be *binary*.

A digital system is an interconnection of digital modules. To understand the operation of each digital module, it is necessary to have a basic knowledge of digital circuits and their logical functions. The basic tools of digital circuits, such as logic gate structure, combinational and sequential circuits.

## Laboratory course

The *digital logic circuits lab* course is designed to provide hands-on experience on digital circuits. This course is offered to students who have passed *Digital Logic Design - EE 231* a 3 credit hour theory course. A total of 10 experiments are designed which include building digital circuits, verify Boolean algebraic theorems and postulates, design combinational circuits, code converters such as Gray code to binary, use of multiplexers and demultiplexers, binary adders and subtractors, design of sequential circuits such as counters and register. These experiments can be performed in the laboratory using digital logic board. The digital logic board has in-built 4-input AND/NAND gates, 4-input OR/NOR gates, XOR gates, inverters, 8 inputs, D flip-flops, SR flip-flops and JK flip-flops. A set of wires are provided that allows construction of fairly large and complex digital circuit along with inputs and output (LEDs or seven segment). Digital logic board also has clock input useful for sequential circuit.

# Chapter 1

## Experiment 1: Digital Logic Gates

This experiment is designed to investigate the logic behavior of various fundamental logic gates such as AND gate, OR gate, NOT gate and XOR gate. The construction of simple digital logic circuits and alternate construction of AND, OR and NOT gate is given. The students will investigate various gates by changing inputs and recording the output using LEDs and fill truth tables to verify their behavior.

### 1.1 Basic Logic Gates

Verify binary logic for two variables  $(x,y)$  for AND  $(x.y)$ , OR  $(x + y)$ , XOR  $(x \oplus y)$ , NAND  $(x.y)'$ , NOR  $(x + y)'$  by filling the Table 2.1.



Table 1.1: Logic behavior of fundamental gates

x	y	$x \cdot y$	$x + y$	$x \oplus y$	$(x \cdot y)'$	$(x + y)'$

## 1.2 Alternate Construction of Basic Gates

a. Implement two variable AND gate and OR gate using NAND gates. Draw figure of your implementation and verify using a truth table.

b. Implement two variable AND gate and OR gate using NOR gates. Draw figure of your implementation and verify using a truth table.

### 1.3 Simple Digital Circuit

Connect the digital circuit  $F = AB + CD$  with four inputs given in Fig. 1.1 on the digital logic board and note the circuit output in the truth table of Table 3.1.

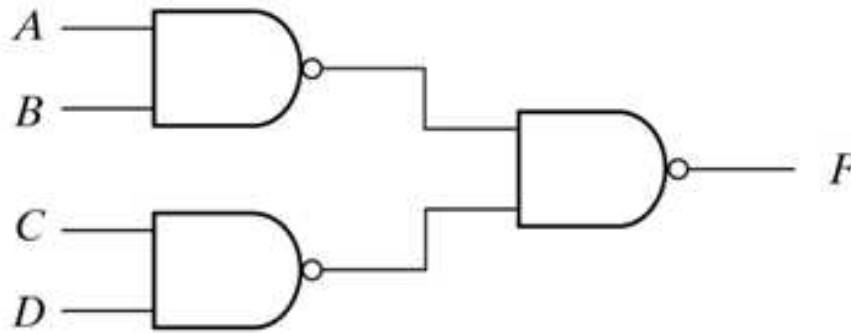


Figure 1.1: A Simple Digital Logic Circuit

### 1.4 DeMorgan's Theorems

Write DeMorgan's theorems for two variables,  $X$  and  $Y$ , and draw their logic diagrams.

Verify or proof DeMorgan's theorems by using the digital logic board. Use empty truth tables given below.

### 1.5 Dual of a function

Write the dual of  $F = xy + y(x + z)$ , draw the logic diagram and show that the output of dual function is same as  $F$ . Use the Table 5.1.



Table 1.3: Truth table for DeMorgan's theorem

X	Y	$X + Y$	$XY$	$(X + Y)'$	$(XY)'$

Table 1.4: Truth table for DeMorgan's theorem

X	Y	$XY$	$X'$	$Y'$	$(XY)'$	$X' + Y'$

Table 1.5: Truth table for function and its dual function

x	y	z	F	$F_{dual}$

## Chapter 2

# Experiment 2: Boolean Algebra and Functions

This experiment's **objectives** are to verify basic postulates and theorems of Boolean algebra. It also demonstrate the relationship between Boolean function and logic diagram. Simplification using Karnaugh map and design of logic circuits with normal and NAND gates is also shown.

### 2.1 Postulates and Theorem of Boolean Algebra

If  $x, y, z$  are the inputs verify the list of postulates by developing truth tables and using the digital logic board.

**a.**  $x + 0 = x$

**b.**  $x + 1 = 1$

**c.**  $x \cdot 0 = 0$

**d.**  $x \cdot 1 = x$

**e.**  $x + x = x$

**f.**  $x + x' = 1$

**g.**  $x \cdot x = x$

**h.**  $x \cdot x' = 0$

**i.**  $(x')' = x$

**j.**  $x + xy = x$

**k.**  $x + x'y = x + y$

**l.**  $(x + y)(x + z) = x + yz$

**Procedure:**

## 2.2 Logic diagram and Boolean expression

For the logic diagram given below, obtain Boolean expression. Simplify the Boolean expression using Boolean algebra and draw the the simplified logic diagram.

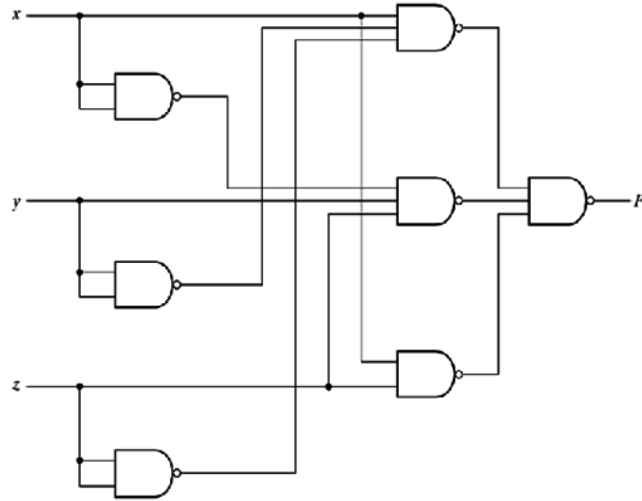


Figure 2.1: Digital Logic diagram

**Procedure:**



## 2.3 Logic diagram from sum of minterms

Two Boolean functions are given in terms of sum of minterms. Use K-map to get minimized Boolean expressions, draw the logic diagram and construct them on digital logic board.

$$F_1(A, B, C, D) = \sum(0, 1, 4, 5, 8, 9, 10, 12, 13)$$

$$F_2(A, B, C, D) = \sum(3, 5, 7, 8, 10, 11, 13, 15).$$

**Procedure:**

## 2.4 Short Theory Questions

1. What is the relation between minterms and k-map?
2. Prove  $x + x = x$  using Boolean algebra.
3. Prove  $x + x'y = x + y$  using Boolean algebra.
4. How to convert a logic circuit with normal gates to logic circuit with only NAND gates?

# Chapter 3

## Experiment 3: Combinational Circuits

This experiment's **objective** is design and analysis of combinational circuits. Combinational circuits are logic circuits with no feedback mechanism i.e., the output depends only on the current input values. Analysis of combinational circuit involves given a circuit you will need to identify inputs, outputs, truth table and the expression. Design of Combinational circuits involves creating truth table for a given design problem then derive the expressions for outputs using Karnaugh maps. Combinational circuits are discussed in section 3.9 and 4.9 in the textbook.

### 3.1 Analysis of Combinational Circuit

Analyze the combinational circuit given in Fig. 3.1. Derive the Boolean expressions for  $T_1$  through  $T_3$ . Evaluate the outputs  $F_1$  and  $F_2$  as functions 3 variables. List the truth table with 8 binary combinations of three input variables and show the binary values of

$T_1$  through  $T_3$  and the outputs  $F_1$  and  $F_2$ .

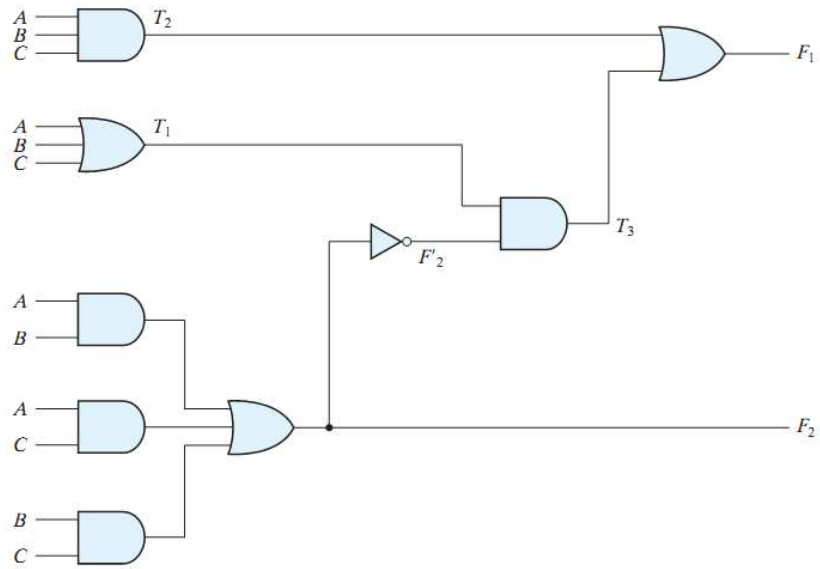


Figure 3.1: Combinational Circuit

**Procedure:**

Table 3.1: Truth table for analysis of combinational circuit

A	B	C	$T_1$	$T_2$	$T_3$	$F_1$	$F_2$

## 3.2 Design from Truth Table

Design a combinational circuit with 4-bit inputs using Table 3.2. Obtain the expression for the output function and draw the logic diagram with basic gates. Verify the design by filling the last column of Table 3.2.

### Procedure:

- Express the  $F$  in terms sum of minterms

.

$$F = \sum( \quad )$$

- Use Karnaugh map to obtain minimized expression for  $F$

.


- Draw the circuit for expression of  $F$ .

Table 3.2: Truth table for designing combinational circuit

A	B	C	D	$F$	Circ. O/P
0	0	0	0	0	
0	0	0	1	0	
0	0	1	0	0	
0	0	1	1	0	
0	1	0	0	0	
0	1	0	1	1	
0	1	1	0	1	
0	1	1	1	0	
1	0	0	0	1	
1	0	0	1	1	
1	0	1	0	1	
1	0	1	1	1	
1	1	0	0	0	
1	1	0	1	1	
1	1	1	0	1	
1	1	1	1	0	

### 3.3 Majority Logic

Design a combinational circuit whose output is equal to "1" if majority of inputs are 1's else 0, let the 3 inputs be  $x, y, z$  and output  $F$ . Simplify the output function, write the procedure for solving the majority logic problem and draw the logic diagram using basic and NAND gates.

**Procedure:**



Table 3.3: Truth table for majority logic

x	y	z	F	$F_{normal}$	$F_{NAND}$

### **3.4 Parity Generation**

Design a logic circuit that generates an even parity bit from 4-bit message using XOR and extend the even parity circuit to generate odd parity circuit. Use the Table 3.4. Write procedure, derive expressions using Karnaugh maps, draw circuit of even parity circuit and odd parity circuit and verify the circuit's behavior.

**Procedure:**

Table 3.4: Truth table for designing even and odd parity bit generation

A	B	C	D	$P_{even}$	$P_{odd}$	Exp. $P_{even}$	Exp. $P_{odd}$
0	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	0	0	0				
1	0	0	1				
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

## **3.5 Short Theory Questions**

1. What are combinational circuits?
2. Can a JK flip-flop be part of combinational circuit?
3. What is the main feature of Karnaugh map?
4. What is a universal gate?

# Chapter 4

## Code Converters

This experiment emphasizes on design of code converters. To minimize the obtained output expression by the use of Karnaugh map. To design the corresponding circuit on hardware. Code conversion is discussed in section 4.4 in the textbook.

### 4.1 Gray Code to Binary Code

Design a combinational circuit that converts a four-bit Gray code input to four-bit binary number; draw the logic diagram with XOR gates.

**Procedure:**

Table 4.1: Truth table for Gray Code to Binary Code

Decimal	Gray Code				Binary Code			
	A	B	C	D	w	x	y	z
0								
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								

## 4.2 9's Complementer

Design a combinational circuit that inputs 4-bit BCD code and outputs its 9's complement. The decimal digit is represented in BCD therefore for decimal number after 9 the BCD code is not useful, if there is a number after decimal 9 it should give a flag bit which is equal to one. Write the procedure, use Table 4.2, obtain expressions using Karnaugh maps, draw the logic diagrams.

**Procedure:**

Table 4.2: Truth table for 9's complementer

Decimal	BCD Code				9's complement				flag bit
	A	B	C	D	w	x	y	z	g
0									
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									



## 4.3 7-Segment Display

Demonstrate on digital logic board the working of 7-segment display. Display numbers from 0 to 31 on it.

**Procedure:**

## 4.4 Short Theory Questions

1. Represent decimals 0 to 9 in four-bit 6-3-1-1 code.
2. If the word length is  $n = 4$  bits (including sign), what decimal number does  $1000_2$  represent in sign and magnitude? In 2's complement? In 1's complement?
3. What is meant by overflow? How can you tell that an overflow has occurred when performing 1's or 2's complement addition?
4. Is it possible to construct a 5-4-1-1 weighted code? Justify your answer.

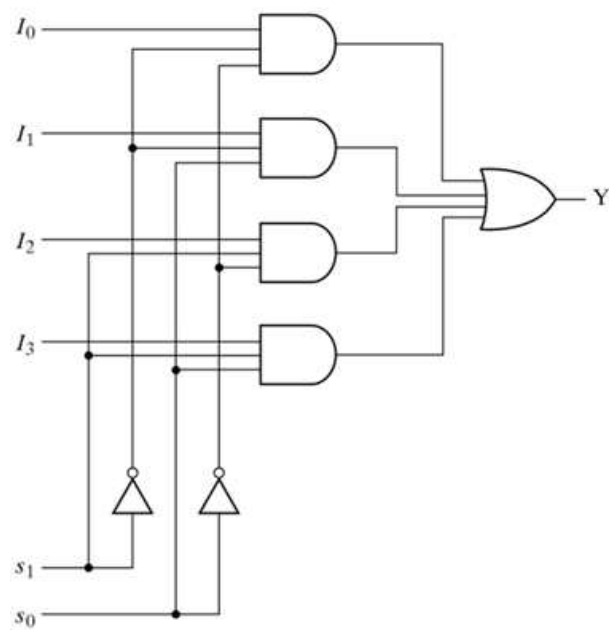
# Chapter 5

## Multiplexers and Decoders

The **objective** of this experiment is to learn to construct multiplexers using basic gates and design of combinational circuits that involves multiplexers and demultiplexers. It deals with decoders and their implementation. The digital logic board has  $4 \times 1$  multiplexer and  $1 \times 4$  demultiplexer built into it. There is a enable input which should be set at high in order for the multiplexer and demultiplexer to work. Multiplexers are discussed in section 4.11 of the textbook.

### 5.1 $4 \times 1$ Multiplexer using Basic Gates

Multiplexer is combinational circuit that selects binary information from one of many input lines and directs it to a single output line. An example of  $4 \times 1$  multiplexer and its function table are given in Fig. 5.1.



$s_1$	$s_0$	$Y$
0	0	$I_0$
0	1	$I_1$
1	0	$I_2$
1	1	$I_3$

(b) Function table

(a) Logic diagram

4-to-1-Line Multiplexer

Figure 5.1: 4-to-1 Line multiplexer

Demonstrate the connections and working of  $4 \times 1$  multiplexer given on the digital logic board.

## **5.2 Design Service Window Display**

There are four windows in a bank, if any bank officer presses the button the LED should glow and the number of his window should be displayed on seven segment display.

Write the steps for the procedure, draw the circuit diagram and tabular forms wherever necessary.

### **Procedure:**

## 5.3 Implementing Logic Circuits using Decoders

Using a demultiplexer and external gates, design the combinational circuit defined by following three Boolean functions:

$$F_1 = x'y' + xy$$

$$F_2 = xy' + x'y$$

$$F_3 = x'y' + y$$

**Procedure:**

Table 5.1: Truth table for decoder implementation

$x$	$y$	$F_1$	$F_2$	$F_3$

## 5.4 Short Theory Questions

1. How many selection lines are needed for  $8 \times 1$  multiplexer?
2. Draw the block diagram of  $4 \times 1$  multiplexer with selection inputs.
3. What is a decoder?
4. If the sum  $S = x \oplus y$  and carry  $C = x.y$  then draw the block diagram to implement a full adder using multiplexers?



# Chapter 6

## Adders and Subtractors

The **objectives** of this experiment are to construct and test various adder and subtractor circuits. Adders and subtractors are one of the most commonly used combinational circuits in logic design. Subtractor is used to compare relative magnitude of two numbers. Adders are discussed in Section 4.3. Subtraction with 2's complement is discussed in Section 1.6. A four-bit parallel adder-subtractor and comparison of two numbers are discussed in Section 4.4 and Section 4.8 respectively.

## 6.1 Half Adder

Design a half adder logic circuit. Use inputs  $x$  and  $y$ , and the outputs  $S$  (sum) and  $C$  (carry). Build the truth table 6.1, get the expressions for  $S$  and  $C$ , construct the logic diagram. Build the circuit on the logic board to verify its operation.

Table 6.1: Truth table for Half Adder

x	y	$S$	$C$

**Procedure:**

## 6.2 Full Adder

Design a full adder logic circuit. Use inputs  $x$ ,  $y$  and  $z$ , and the outputs  $S$  (sum) and  $C$  (carry). Build the truth table 6.2, get the expressions for  $S$  and  $C$ , construct the logic diagram. Build the circuit on the logic board to verify its operation.

Table 6.2: Truth table for Full Adder

$x$	$y$	$z$	$S$	$C$

**Procedure:**

## 6.3 Parallel Adder and Subtractor

### 6.3.1 Parallel Adder

Construct a parallel adder using the adder provided on the logic board which adds two 4-bit binary numbers and gives 4-bit sum and 1-bit carry. Let  $a_3a_2a_1a_0$  and  $b_3b_2b_1b_0$  are added to give  $s_3s_2s_1s_0$  (4-bit sum) and  $C$  (1-bit carry). Test your circuit by doing at least 3 test additions.

### 6.3.2 Parallel Subtractor

Modify the parallel adder circuit used in part a to get a parallel subtractor which subtracts two 4-bit binary numbers and gives 4-bit result. Let  $a_3a_2a_1a_0$  and  $b_3b_2b_1b_0$  are subtracted to give  $s_3s_2s_1s_0$  (4-bit result). Test your circuit by doing at least 3 test subtractions.

**Procedure:**

## 6.4 Magnitude Comparator

Design magnitude comparator using the subtractor circuit such that if  $a_3a_2a_1a_0 = b_3b_2b_1b_0$  then  $s_3s_2s_1s_0 = 0000$ . If  $a_3a_2a_1a_0 > b_3b_2b_1b_0$  then carry bit is 1. If  $a_3a_2a_1a_0 < b_3b_2b_1b_0$  then carry bit is 0.

**Procedure:**

## 6.5 Short Theory Questions

1. How can you modify a parallel adder to a parallel subtractor circuit?
2. How can you connect two half adders to get a full adder circuit?
3. What is overflow bit in the context of binary addition?
4. Is there any other way of constructing a magnitude comparator than the one discussed in the experiment? Discuss.

# Chapter 7

## Experiment 7: Latches and Flip-Flops

The **objectives** of this experiment are to construct and verify operation of various latches and flip-flops. Latches and flip-flops are storage elements of sequential circuits which involve feedback. The sequential circuit receives binary information from external inputs that, together with the present state of the storage elements, determine the binary value of the outputs. Different storage elements discussed in SR latch, D latch, D flip-flop, JK flip-flop and T flip-flop. Sequential circuits are discussed in Chapter 5 of the text book.

## 7.1 SR Latch

Construct a SR latch using NAND gates according to the truth table given in Fig. 7.1.

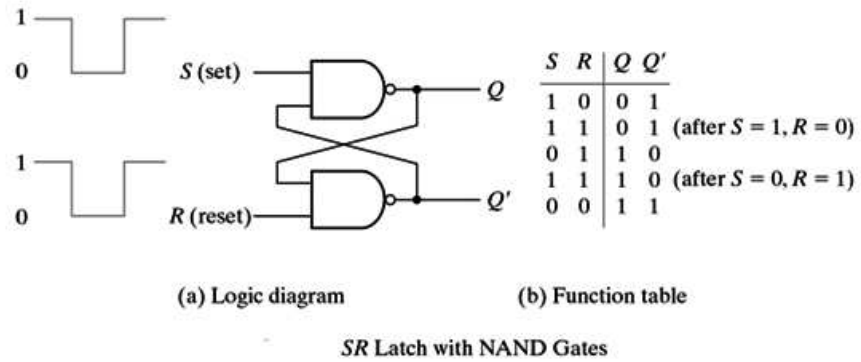


Figure 7.1: SR latch diagram and truth table

**Procedure:**



## 7.2 D Latch

Construct a D latch using NAND gates according to the truth table given in Fig. 7.2.

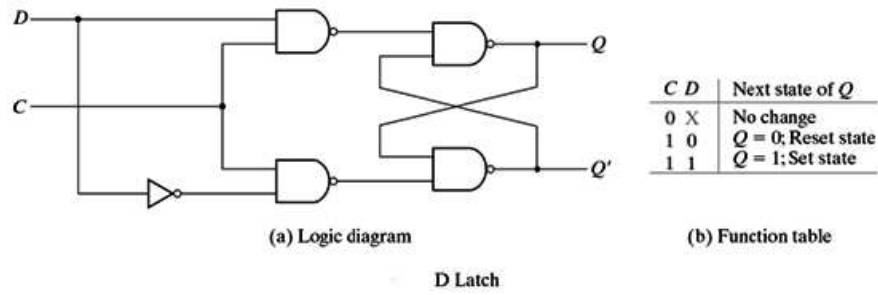


Figure 7.2: D latch diagram and truth table

**Procedure:**

## 7.3 Edge-Triggered D Flip-Flop

Construct a D flip-flop with two D latches and an inverter as shown in Fig. 7.3. The first latch is called the master and second the slave. The circuit samples the D input and changes its output Q only at the negative edge of the controlling clock. Draw a timing diagram by changing input and clk at different times and observing the output.

What do you notice?

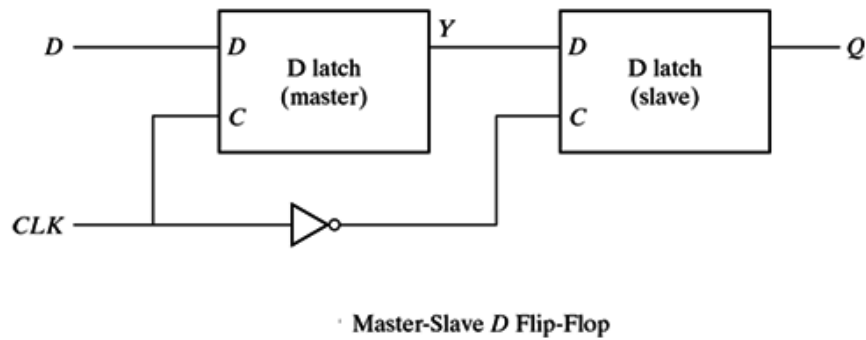


Figure 7.3: Master Slave D flip-flop

**Procedure:**

## 7.4 Analyze a Sequential Circuit

Analyze the clocked sequential circuit given in the Fig. 7.4. Get expressions for  $A(t+1)$ ,  $B(t+1)$  and  $y(t)$ . Develop state table and state diagram for the circuit.

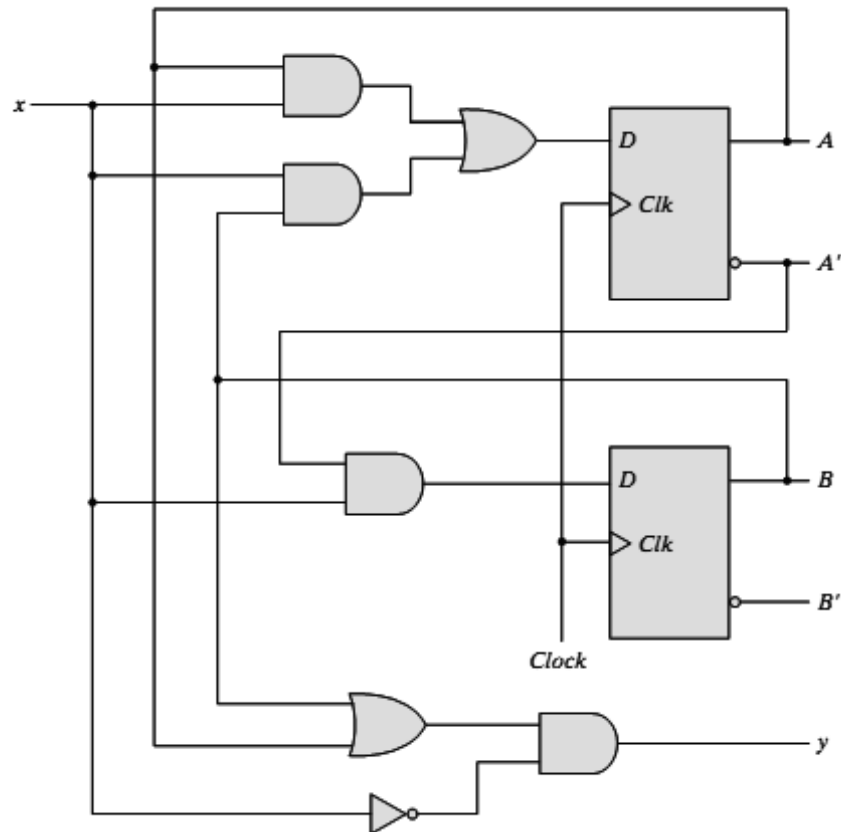


Figure 7.4: Sequential Circuit with two D flip-flops

**Procedure:**



## 7.5 Short Theory Questions

1. What is the main difference between a latch and a flip-flop?
2. Construct a dummy state table with two flip-flops A and B, input x and an output y?
3. What is the main disadvantage of using a latch?
4. What type of logic circuits you can construct or design with flip-flops?

# Chapter 8

## Sequential Circuits

The **objectives** of this experiment are to construct a class of sequential of circuits known as counters. A register that goes through a prescribed sequence of states upon the application of input pulses is called a counter. The input pulses may be clock pulses, or they may originate from some external source and may occur at a fixed interval of time or at random. The sequence of states may follow the binary number sequence or any other sequence of states. Counters are discussed in Section 6.3-6.6 in the book.

### 8.1 Up-Down Counter with Enable

Design a counter using two T flip-flops which counts up from 00 to 11 (binary sequence) when input  $x = 1$ , the counter counts down from 11 to 00 (binary sequence) when input  $x = 0$ .

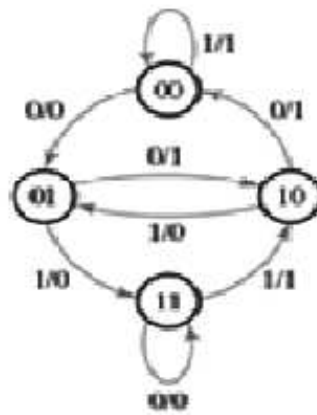
**Procedure:**

Table 8.1: State transition table for up counter

Present State		input	Next State		D Flip-flops	
A	B	x	A	B	$D_A$	$D_B$

## 8.2 Design using State Diagram

Design, construct and test a sequential circuit using D flip-flops whose state diagram is given in 8.1.



State Diagram:

Figure 8.1: State diagram

**Procedure:**



Table 8.2: State transition table for state diagram

Present State		input	Next State		JK Flip-flops				output
A	B	x	A	B	$J_A$	$K_A$	$J_B$	$K_B$	y



## 8.4 Short Theory Questions

1. What is a Ring Counter?
2. What is the state table for T flip-flop?
3. How many flip-flops do you need to design a BCD counter?
4. Which flip-flop you found was the easiest to be used in design? And why?

# Chapter 9

## Counters

The **objectives** of this experiment are design, construct and verify ripple and synchronous counters. A ripple counter consists of a series connection of complementing flip-flops, with the output of each flip-flop connected to the clock input of the next higher order flip-flop. The flip-flop hold the least significant bit receives the incoming count pulses. In synchronous counters, the clock input of all flip-flops receive the common clock. The output of synchronous counter changes according to the current states of the flip-flops. Ripple and synchronous counters are discussed in Section 6.4 and 6.5.

## 9.1 Ripple Counter

Design a four-bit binary ripple counter that is a counter which goes from 0000 to 1111 and then back to 0000. Design the counter using JK flip-flops. Draw the circuit, construct on logic board and verify the counter. How can you modify the circuit to count back from 1111 to 0000?

**Procedure:**



## 9.3 Decimal Counter

Design a decimal counter that counts from 0000 to 1001 and then goes back to 0000.

Design a synchronous counter using JK flip-flops and AND gates. Draw the circuit, construct on logic board and verify.

### **Procedure:**





## 9.4 Short Theory Questions

1. What is the difference between ripple and synchronous counter?
2. Why counters are easier to design? And why?
3. Can similar procedure be adopted to design synchronous counters with T and D flip-flops?

# Chapter 10

## Shift Registers

The **objective** of this experiment is to investigate the working of shift registers using digital logic board. Parallel and serial loading of shift registers will be performed. Shift register module which is in-built the logic board will be used. Shift registers are discussed in Section 6.2 of the textbook.

## 10.1 Construct a Shift Register

Construct a four-bit shift register on the digital logic board. Demonstrate the serial and parallel load operations. Draw a block diagram and note steps.

### **Procedure:**

## 10.2 Ring Counter

Construct a four-bit ring counter on the digital logic board and verify.

**Procedure:**