الهيئة العامة للتعليم التطبيقي والتدريب
كلية التكنولوجيا

رقمية (١)

LAB

سعر التصوير.٥٠٠
سعر التغليف : ٥٠٠ فلس
سعر التصوير والتغليف : دينار

راشد الزعبي

الكترونيات دوائر رقميه ١

دوائر رقميه (١) + الكترونيات رقميه

ENT –120 + 142

Introduction to Digital Systems Lab Manual

م. نورة عبدالعزيز الصقر
قسم الهندسة الالكترونية

كلية الدراسات التكنولوجية
م. هيا عبدالرزاق الخضير

Prepared By

Eng. Ahmad H. Al-Mazyad Eng. Nasser Al-Rasheed

The Public Authority for Applied Educarion and Training
College Of Technological Studies
Electronics Department

TABLE OF CONTENTS

Exp. (1)
The Basic Logic Gates

Objective:

To become familiar with the logical gates AND, OR and NOT and their truth tables.

Introduction:

The Boolean 0 and 1 do not represent actual numbers but instead represent the state of a voltage variable, or what is called its logic level. Digital circuits are built based on the logic level of the voltage, either binary 1 (2-5 v) or binary 0 (0-0.8 v). The basic operations are called logic operations. Digital circuits, called logic gates, can be constructed using diodes, transistors, and resistors. When connected properly the output of the circuit is the result of a basic logic operation (OR, AND, NOT) performed on the inputs. A gate is simply an electronic circuit which operates on one or more input signals to produce an output signal. It is a decision- making logic element.

Part A: The Inverter Gate:

The inverter gate has one input line and one output line. If the input is high, the inverter output will be low, and vice-versa, as in fig (1). The Boolean expression for the inverter (Not operation) $x = \overline{A}$, this expression reads "x equals NOT A" or "x equals the inverse of A" or "x equals the complement of A". That is the value of x:

$\overline{1} = 0$ because NOT 1 is 0.

and $\overline{0} = 1$ because NOT 0 is 1

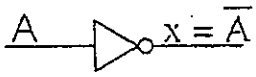


Figure 1

Procedure:

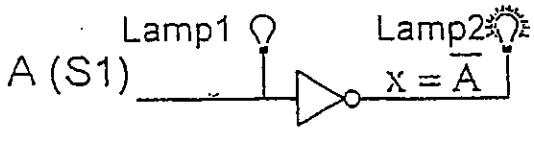


Figure 2

Table 1

A	$x = \overline{A}$
0	
1	

1- Connect switch S1 as in fig (2).

2- Set switch S1 up. Lamp 1 (input) is ON. Observe lamp 2 (output).

3- Set switch S1 down. Lamp 1 is OFF. Observe lamp 2.

4- Record the results in table (1).

5- Connect two inverters in series fig. (3). Check the output of the combination. Fill up the truth table.

The output is represented by $x = \overline{\overline{A}}$.

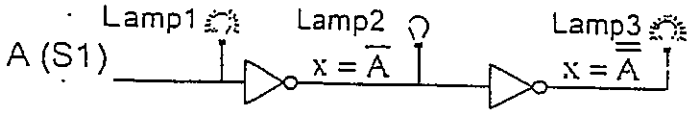


Figure 3

Table 2

A	$x = \overline{A}$	$x = \overline{\overline{A}}$
0		
1		

Part B: The AND Gate:

An AND gate provides an output of "logical 1" only if all the inputs are HIGH. Two logic variables A and B are combined using the AND operation, x = A . B which is read "x equals A AND B". The symbol "." stands for the Boolean AND operation. The result can be expressed as:

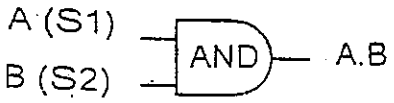


Figure 4

Table 3

A	B	AB
0	0	
0	1	
1	0	
1	1	

Procedure:

1- Hook up the switches to the AND gate as in fig (4).

2- Set switches S1 and S2 down, and observe the condition of the output lamp.

3- Try all possible combinations of the switches and record the results in table (3).

Observe that the only condition which can turn the output light on is when A and B are both HIGH.

Part C: The OR Gate:

One of the simplest and most frequently used gate is called the OR gate. The block diagram symbol for the OR gate is as shown in fig. (5). The OR gate operates in such a way that its output is high (logic 1) if either input A or B or both are at logic 1 level. The OR gate output will be low (logic 0) only if all its inputs are at logic 0. The Boolean expression x = A + B reads as " x equals A OR B".

The OR gate can be represented by the logic symbol shown in fig. (5).

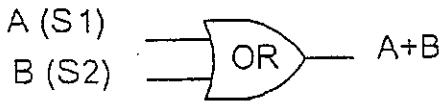


Figure 5

Table 4

A	B	A+B
0	0	
0	1	
1	0	
1	1	

Procedure:

1- Hook up the switches to the OR gate as shown in fig. (5).

2- Set switches S1 and S2 as shown in table (4) and record their output.

Assignment:

- Write a 3-input truth table for the AND and the OR gates.
- Is it possible to write a 3-input truth table for the inverter gate?
- What is the result of connecting two inverters in series?

Exp. (2)
NAND & NOR Gates

Objective:

To become familiar with the logical gates NAND and NOR and their truth tables also to show the associative law and its application on the basic gates.

Introduction:

In this experiment we will continue to investigate the basic gates. More gates can be built using these gates. The NAND & NOR gates can be built using the previous, introduced gates. NAND gate fig. (1.a) has a wide application in the computer circuits area, as the name implies the N stands for "NOT" while the AND is the normal AND gate. The Boolean expression: $x = \overline{A.B}$ is read "NOT (A AND B)" or "A NAND B". The truth table for a NAND gate is the complement of the AND gate truth table. The same thing applies to the NOR gate fig. (1.b), where the Boolean expression: $x = \overline{A+B}$ reads "NOT (A OR B)" or "A NOR B". The result of the NOR gate truth table is also the complement of the OR gate truth table.

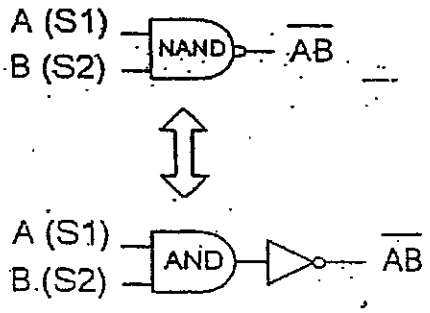


Figure 1.a

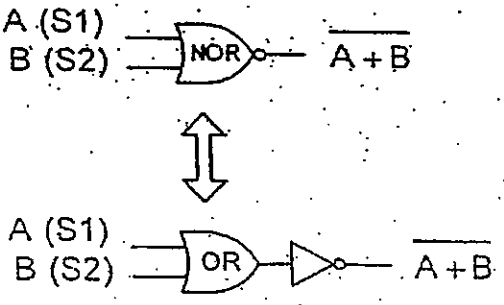


Figure 1.b

Part A: NAND Gate:

Procedure:

1- Hook up the switches for the NAND gate as shown in fig. (2).

2- Using the switches try all possible combination for truth table (1) inputs. Record the output according to the output lamp results.

Figure 2

Table 1: NAND gate

A	B	X
0	0	
0	1	
1	0	
1	1	

3- Add to your circuit an inverter as in fig. (3).

4- Fill up truth table (2) following all the combinations for the inputs. Observe the difference between table (1) & table (2).

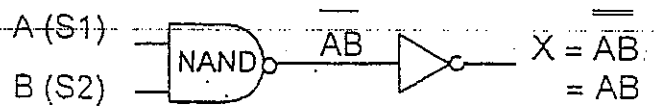


Figure 3

Table 2: AND gate

A	B	X
0	0	
0	1	
1	0	
1	1	

Part B: NOR Gate:

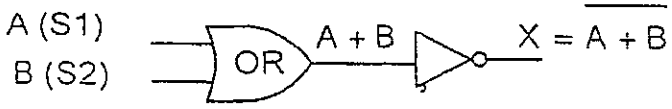


Figure 4

Table 3: NOR gate

A	B	X
0	0	
0	1	
1	0	
1	1	

Procedure:

1- Connect the given circuit in fig. (4).

2- Try all conditions for the inputs in the truth table. Record your output according to the output lamp response.

Associative Laws:

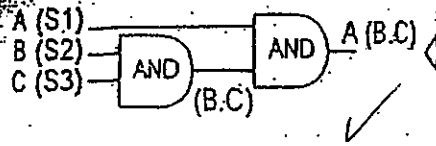


Figure 5

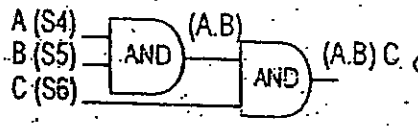


Figure 6

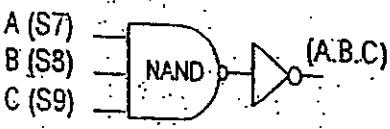


Figure 7

1- Construct the circuits in fig. (5-7).

2- Using switches try all possible input combinations for table (3).

3- Compare both results.

Table 4: 3-input AND gate

A	B	C	A(B.C)	(A.B)C	A.B.C
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	1	1	1

This concept is called the associative law. The law applies to both the OR and the AND gate. This can be stated for both gates as follows:

$$A\,(BC) = (AB)\,C = ABC \rightarrow \text{yes}$$
$$A + (B + C) = (A + B) + C = A + B + C$$

Assignment:

$$A.B = B.A$$
$$A + B = B + A$$

yes ↓ associative

- For the two theorems above:
  - What do we call this property?
  - Is the result of AB + CD is the same as DC + BA ? Why?

yes

yes

- Write a 3-input truth table for the NAND and NOR gates.

$2^3 = 8$

Exp. (3)
Universal NAND Gate

Objective:

To implement the basic gates (AND, OR, and Inverter) using either NAND or NOR gates.

Introduction:

In Exp. (1) The basic operation gates were introduced, while in Exp. (2) we learned how to build NAND & NOR gates. The NAND and the NOR gates are said to be universal because any digital system can be implemented using them. To show that any Boolean function can be constructed with NAND & NOR gates, we need only to show that the logical operations of AND, OR and complement can be obtained using NAND and NOR gates. In this experiment, we will implement the three basic gates using only NAND gates. The same thing can be accomplished using NOR gate (which will be given as an assignment).

Part A: Constructing An Inverter Gate Using A NAND Gate:

Starting with the NAND gate, the expression $X = \overline{A.A}$ represents a two input NAND gate with the same input value for both inputs. From our basic Boolean theorem $A.A = A$, then $X = \overline{A.A} = \overline{A}$ which stands for the inverter gate fig 1.

Procedure:

1- Connect the NAND gate as shown in fig (1).

2- Fill up the given truth table according to the input condition.

Table1:

Inverter

A	$Z=\overline{A}$
0	
1	

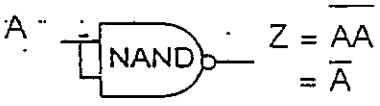


Figure 1: NAND gate as an inverter

Part B: Constructing An AND Gate Using NAND Gates:

If the NAND gate is followed by another NAND shorted as in fig. (2), the result is an AND gate where the Boolean expression is $X = \overline{A.B}$ & $Z = \overline{X} = \overline{\overline{A.B}} = A.B$ as in fig (2). The two bars on the top of AB can be canceled as stated in Exp. (1).

Procedure:

1-Connect the NAND gate as in fig. (2).

2- Check if the truth table of your circuit stands for the AND gate.

Table 2: AND gate

A	B	Z
0	0	
0	1	
1	0	
1	1	

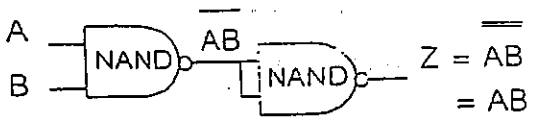


Figure 2: NAND gates constructing an AND gate.

Part C: Constructing An OR Gate Using NAND Gates:

An OR gate can be produced by using a NAND gate in which each input to that NAND gate is inverted using a shorted NAND, as in fig (3). The Boolean expression will be $Z = \overline{\overline{A}.\overline{B}} = \overline{\overline{A}} + \overline{\overline{B}} = A + B$, which produces the needed OR gate.

Procedure:

1- Using NAND gates only, implement fig. (3) circuit.

2- Fill up the truth table and observe the output Z if it match the OR gate output.

Table 3: OR gate

A	B	Z
0	0	
0	1	
1	0	
1	1	

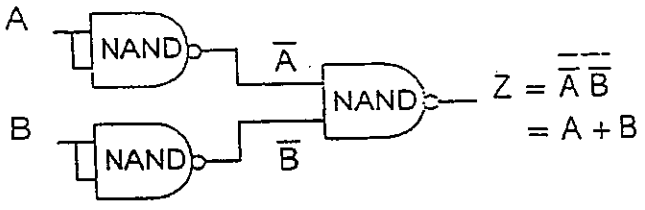


Figure 3: Three NAND gates constructing a two-input OR gate.

Assignment:

- Implement the three basic gates (Inverter, AND and OR gate) using NOR gates only.

Exp. (4)
Boolean Theorems

Objective:

To become familiar with the Boolean theorem and to prove the theorems practically.

Introduction:

In this experiment we will introduce some of the basic relationships in Boolean algebra. Such relations can be used to analyze and design digital circuits. Also, the Boolean algebra relations can be used to simplify large circuits to the smallest possible circuits. Boolean algebra deals with binary variables and logic operations. The three basic logic operations used are AND, OR and complement. For a given value of the binary variables, the Boolean function can be equal to either 1 or 0.

The experiment is divided into two parts. The first part demonstrates single variable theorems. The second part implements multivariable theorems.

Notice: This experiment will be covered in two weeks in which the first week will be devoted for the first part (single variable theorems) and the second part (multi variable theorems) will be covered in the second week.

Part A: Single Variable Boolean Theorem:

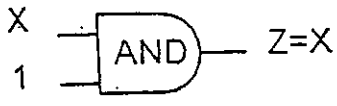
For the following Boolean expressions connect its equivalent gate to verify the answer for each theorem:

Case 1: $X \cdot 0 = 0$

X	0	Z
0	0	
1	0	

Case 2: $X \cdot 1 = X$

X	1	Z
0	1	
1	1	

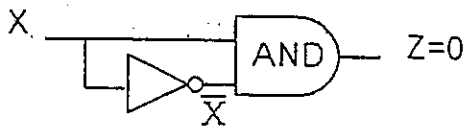


Case 3: $X \cdot X = X$

X	X	Z
0	0	
1	1	

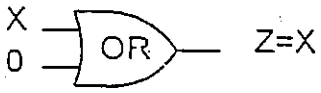
Case 4: $X \cdot \overline{X} = 0$

X	$\overline{X}$	Z
0	1	
1	0	



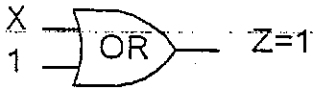
Case 5: $X + 0 = X$

X	0	Z
0	0	
1	0	



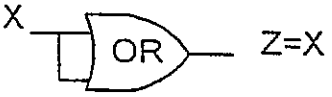
Case 6: $X + 1 = 1$

X	1	Z
0	1	
1	1	



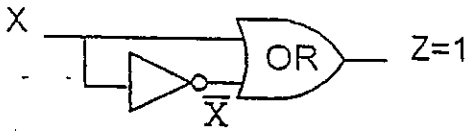
Case 7: $X + X = X$

X	X	Z
0	0	
1	1	



Case 8: $X + \overline{X} = 1$

X	$\overline{X}$	Z
0	1	
1	0	



Part B: Multivariable Boolean Theorem:

For each of the following expressions:

1. Draw both equal logic circuits.
2. Connect and fill the tables with the output.
3. Compare the results of both circuits.

Expressions:

1) The Common Factor:

$$A + AB = A\,(1 + B) = A$$

Table 9: Expression 1

A	B	A + AB	A
0	0		
0	1		
1	0		
1	1		

2) Distributive Law:

$x\,(y + z) = xy + xz$
$(w + x)\,(y + z) = wy + wz + xy + xz$

$$(A + B)\,(A + C) = AA + AC + AB + BC$$
$$= A + AC + AB + BC$$
$$= A\,(1 + C + B) + BC$$
$$= A + BC$$

Table 10: Expression 2

A	B	C	(A + B).(A + C)	A + BC
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

3) DeMorgan's Theorem:

$$\overline{x \cdot y} = \overline{x} + \overline{y}$$
$$\overline{x + y} = \overline{x} \cdot \overline{y}$$

$$\overline{(\overline{A} + \overline{B})C} = \overline{(\overline{A} + \overline{B})} + \overline{C}$$
$$= (\overline{\overline{A}}\,\overline{\overline{B}}) + \overline{C}$$
$$= AB + \overline{C}$$

Table 11: Expression 3

A	B	C	$\overline{(\overline{A}+\overline{B})C}$	$AB+\overline{C}$
0	0	0	1	1
0	0	1	0	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	1	1
1	1	1	1	1

Assignment:

- Using Boolean algebra theorems, simplify the following expressions:

$\overline{x}+xy = x+y$

1- $F = A\overline{C} + AB\overline{C}$

2- $F = \overline{A}\overline{B}C\overline{D} + A\overline{B}CD$

3- $F = \overline{A}\overline{B}C\overline{D} + \overline{A}B\overline{C}\overline{D}$

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

(2) $F = \overline{A}\overline{B}C\overline{D} + A\overline{B}CD$

$= \overline{A}\overline{B}\cdot C\overline{D} + \overline{A}+B+\overline{C}+\overline{D}$

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

$= \overline{A}C\overline{D} + \overline{B}C\overline{D} + A+\overline{B}+\overline{C}+D$

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

$= \overline{A} + \overline{B} + \overline{C} + D$

(3) $F = A\overline{B}C\overline{D} + \overline{A}\overline{B}C\overline{D}$

$= A\overline{B}C\overline{D} + \overline{A}\overline{B}C\overline{D} + \overline{A}\overline{B}C\overline{D}$

$= A\overline{C}\overline{D} + \overline{B}C\overline{D} + \overline{A}\overline{B}C\overline{D}$

$= A\overline{C}\overline{D} + C\overline{D}(\overline{B}+\overline{A}\overline{B})$ $\quad y+\overline{y}x = y+x$

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

$= \overline{A}C\overline{D} + \overline{B}C\overline{D}$

$\overline{A}\overline{D}(C+\overline{B}C) + \overline{B}C\overline{D}$

$\overline{A}\overline{B}(C+\overline{B}) + \overline{B}C\overline{D}$

$\overline{A}\overline{B}C + \overline{A}\overline{B}\overline{B} + \overline{B}C\overline{D}$

Exp. (5)
Combinational Logic Circuits

Objective:

To validate algebraic simplifications through demonstrating the function equivalence of both the simplified and the unsimplified logic circuits.

Introduction:

The Boolean algebra theorems that we went through in the last experiment can be used to simplify logic circuit expressions. The major problem with Boolean algebra simplification is that there is no obvious way to conclude that the simplest form of an expression is reached.

This experiment will start with implementing a reasonably large logic circuit. Then the circuit will be simplified and both circuits will be tested for equivalence.

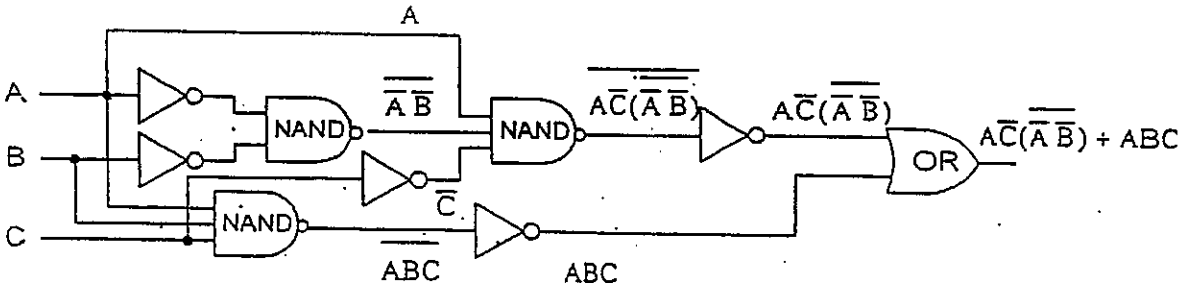


Figure 1

Procedure:

To simplify the logic circuit shown in fig. (1):

First we have to determine the output expression:

$$Z = A\overline{C}(\overline{A\,\overline{B}}) + ABC \quad \text{--------(1)}$$

Using DeMorgans Theorems to break down all large inverter signs

$$Z = ABC + A\overline{C}(A + B)$$
$$= ABC + A\overline{C}A + A\overline{C}B$$
$$= ABC + A\overline{C} + A\overline{C}B$$

Taking AB as a common factor:

$$= AB(C + \overline{C}) + A\overline{C}$$
$$= AB + A\overline{C}$$
$$= A(B + \overline{C}) \quad \text{--------(2)}$$

1- Complete table (1) below to produce a truth table for expression (1) and (2).

2- Draw the logic circuit for the simplified expression, expression (2).

3- Connect both circuits.

4- Compare the truth table you have completed to the output of the two circuits.

Table 1

A	B	C	$\overline{A}$	$\overline{B}$	$\overline{C}$	$\overline{A}\ \overline{B}$	$\overline{\overline{A}\ \overline{B}}$	$A\overline{C}$	ABC	$A\overline{C}.\overline{(\overline{A}\ \overline{B})} + ABC$	$B+\overline{C}$	$A(B+\overline{C})$
0	0	0										
0	0	1										
0	1	0										
0	1	1										
1	0	0										
1	0	1										
1	1	0										
1	1	1										

Assignment:

Simplify:

$$W = \overline{A}\,C\,\overline{(\overline{A}\,B\,D)} + \overline{A}\,B\,C + A\,\overline{B}\,C$$

Exp. (6)
Sum of Product Form

Objective:

To introduce the concept of sum of product & product of sum.

Introduction:

The Boolean algebra expression form and the resulting form of the gating network are based on two main concepts: Product term & Sum term. Where the product term either a single term (A) or the logical product of several term (A.B.C), while the sum term is a single (A) or the sum of several variables (A + B + C). For any given truth table the product term of each line is found by replacing each "zero" in the input of the truth table with the variable complement symbol ($\overline{A}$), while "one" takes the variable as it is (A) as shown in truth table (1). To find the sum term of the truth table take the complement of your product term as in table (1):

Table 1

A	B	Product term	Sum terms
0	0	$\overline{A}.\overline{B}$	$\overline{\overline{A}.\overline{B}} = A + B$
0	1	$\overline{A}.B$	$\overline{\overline{A}.B} = A + \overline{B}$
1	0	$A.\overline{B}$	$\overline{A.\overline{B}} = \overline{A} + B$
1	1	$A.B$	$\overline{A.B} = \overline{A} + \overline{B}$

To find the expression of your logic circuit, you can follow either the sum of product method, where we select the product term for any output equal to "one" and sum those terms. Or to take the product of sum terms, and in this case select the expression related to the output equal to "zero" in the truth table and multiply those terms. Now lets find the expression for truth table (2):

A- Sum of the product method:

X is equal to the sum of the product whenever the output equals one.

Table 2: OR gate

A	B	X
0	0	0
0	1	0
1	0	1
1	1	1

$$\therefore X = A.\overline{B} + A.B$$
$$= A(\overline{B} + B)$$
$$= A$$

B- Product of sum terms:

X is equal to the product of the sum terms where the output equals zero.

Directions: Change the statements to questions using the question word in parenthesis. Check verb tense.

Example: The train arrived at ten o'clock. (what time) ***What time did the train arrive?***

1. They do their homework at night. (when)
2. Mr. Hamad came to the meeting alone. (who)
3. The car is across the street from the house. (where)
4. I like the red blouse, not the blue one. (which)
5. He felt better after he took a nap. (how)
6. That is an English book. (what)
7. Tariq talked to him for an hour. (how long)
8. He studies biology at the university. (what)
9. The party lasted all night. (how long)
10. The check was for K.D. 5.50. (how much)
11. He was eating a sandwich. (what)
12. He is working hard. (how)
13. His parents have two cars. (how many)
14. They are coming to visit us tomorrow. (when)
15. He is going to work right now. (where)
16. This man is my brother. (who)
17. I didn't get up early because I was tired. (why)
18. We have English every day. (how often)
19. They like to rest on weekends. (what)
20. The club is not far from their house. (where)
21. Their favorite kind of food is pizza. (what)
22. I gave him my dictionary. (what)

$$X = (\overline{\overline{A}\,\overline{B}}) \cdot (\overline{\overline{A}\,B})$$
$$= \overline{\overline{A}\,\overline{B} + \overline{A}\,B}$$
$$= \overline{\overline{A}\,(\overline{B} + B)}$$
$$= \overline{\overline{A}}$$
$$= A \Rightarrow \text{SAME AS THE SUM OF PRODUCT METHOD RESULT.}$$

Design Problem:

Design a single output, 3-input combinational circuit whose output is 1 if the sum of the inputs is greater than or equal to 2, otherwise it is 0.

Procedure:

1- From the design problem above fill up the output column in truth table (3).

2- Find the product term for each line having an output of "1" in truth table (3).

Table 3

A	B	C	Output	Product term
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

3- Find the expression of your logic circuit using the sum of product method, simplify it with the Boolean algebra theorem:

F =

4- Draw the simplified circuit for your expression.

5- Connect the circuit and validate the output by following truth table (3).

Assignment:

- Find the expression for your logic circuit using the product of sum terms method. Prove that the two methods are equal.

Exp. (7)
The Karnaugh Map

Objective:

To introduce the K-map as a quick method for solving and minimizing Boolean expressions.

Introduction:

In the previous experiments you were introduced to the Boolean theorems as a tool for simplifying Boolean expressions. K-map, which stands for Karnaugh maps, is a fast graphical method for minimizing Boolean expressions in a single step. K-map can be used as a mean for showing the relationship between logic inputs and the desired output. Each cell in the graphical representation of the K-map can contain one unsimplified product (containing all the variables and allowing single variable complements) from the sum of the product of the output and is called a minterm. The expression for the output Z can be simplified by properly combining those cells in the K-map which contains 1s. This process of combining these 1s is called looping. Looping can be performed by a single cell loop, or 2, 4, 8, 16.. cells loop (Refer to your text for further details). As an example see table (1) and (2) below:

Table 1: 3-variables K-map

	$\overline{C}$	C
$\overline{A}\,\overline{B}$	1	0
$\overline{A}\,B$	1	0
$A\,B$	0	1
$A\,\overline{B}$	0	0

$\overline{A}\,\overline{C}$; ABC

$F = \overline{A}\,\overline{C} + ABC$

Table 2: 4-variables K-map

	$\overline{C}\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$	0	1	0	1
$\overline{A}\,B$	0	0	0	1
$A\,B$	0	1	1	1
$A\,\overline{B}$	0	0	0	1

$\overline{A}\,\overline{B}\overline{C}D$; $A\,B\,D$; C

$F = \overline{A}\,\overline{B}\overline{C}D + C\overline{D} + ABD$

Procedure:

For the given Boolean expression:

$$Z = (A\,\overline{B}\,\overline{C} + A\,C) \div (\overline{B} + B\,\overline{C})$$

1- Draw the expression.

2- From the expression derive the circuit output and record it in truth table (3).

3- Using Boolean algebra, simplify the given expression.

4- Use the K-map method to simplify your expression. Compare the results with step 3 results.

Table 3: The expression output.

A	B	C	Z
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Table 4: K-map

	$\overline{C}$	C
$\overline{A}\,\overline{B}$		
$\overline{A}\,B$		
$A\,B$		
$A\,\overline{B}$		

5- Draw the new simplified expression.

6- Connect both drawn circuits (step 1 & step 5), and verify their equivalence using truth table (3).

Assignment:

- For the K-map in table (5):
  - Find the unsimplified expression (product terms containing all variables, minterms) and simplify it using Boolean algebra.
  - Simplify using the K-map.

Table 5: K-map

	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$	1	0	0	1
$\overline{A}\,B$	1	1	1	1
$A\,B$	0	0	0	0
$A\,\overline{B}$	1	0	0	1

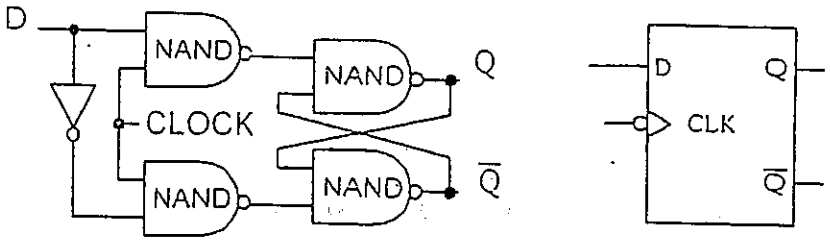


Figure 3: D Flip-Flop

Table 5: D Flip-Flop.

D	Q_t	Q_{t+1}
0	0	
0	1	
1	0	
1	1	

Table 6: D Flip-Flop summarized.

D	Q_{t+1}
0	
1	

Procedure:

1- Connect the circuit shown in fig. (3).

2- Fill table (5) with the circuit's results.

3- From table (5) derive the result for each state in table (6).

Notice:

- Tables (1), (3) and (5) contain the columns Q_t and Q_{t+1}. The first (Q_t) indicates the state of the lamp before applying the tested state to the circuit. While Q_{t+1} is the result (in the Q output) of applying the state.
- To derive the tables (2), (4), and (6) you need to follow the following table:

The result of Q_{t+1}	Summarized as
$Q_{t+1} = Q_t$	No change (Q_t)
Q_{t+1} the complement of Q_t	Toggle ($\overline{Q_t}$)
Always 1.	1
Always 0.	0

Assignment:

- Compare the results and the circuits drawing in this experiment with the book circuits.

Exp. (9)
J-K Flip Flop

Objective:

To study the function of the J-K flip-flop and to investigate its advantages over the S-C Flip Flop.

Introduction:

In Exp. 8 You were introduced to a basic memory element the S-C flip-flop, and the function of the Set and Reset input on the output Q. The main disadvantage of that flip-flop was the invalid state when Set = Reset = 0. The solution to such problem was solved by the feedback link between the outputs and inputs. The resulting circuit was called J-K flip-flop in which the invalid state was converted to a toggle condition. Additional inputs are usually added to allow the flip-flop to operate independently of the function of the clock (asynchronous inputs). These inputs are called preset and clear. They are used to set the FF to 1 (preset) or 0 (clear) at any time regardless of the input condition. The asynchronous inputs can be either active low or active high. Fig. (1) a J-K flip-flop with active low asynchronous inputs, ($\overline{\text{preset}}$) & ($\overline{\text{clear}}$).

Truth Table 1: J-K flip-flop

J	K	Q_t	Q_{t+1}
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

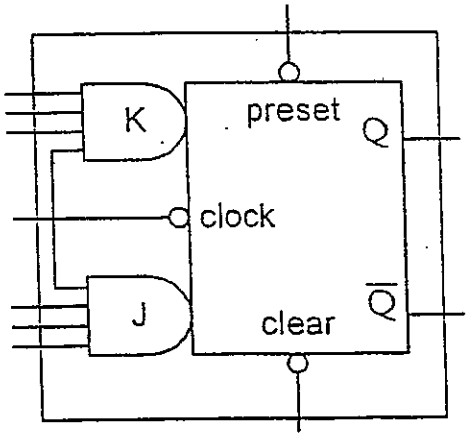


Figure 1: J-K Flip-Flop (module 7472).

Part A: J-K Flip-Flop As A Given Module.

Procedure:

1- Connect the J-K flip-flop (module 7472) as shown in fig (1). Set the asynchronous inputs = 1 ($\overline{\text{preset}}$ & $\overline{\text{clear}}$ = 1)

2- Following truth table (1) investigate the effect of the J & K inputs on the output Q.

3- Minimize table (1) into truth table (2).

Truth Table 2: J-K Flip-Flop

J	K	Q_{t+1}
0	0	
0	1	
1	0	
1	1	

4- Set ($\overline{preset}$) = 0. Now apply the inputs of truth table (1) again and notice the effect of the $\overline{preset}$ input on the function of the FF. Fill the results in table (4).

5- Repeat step 3 only this time set ($\overline{clear}$) = 0 and return ($\overline{preset}$) = 1.

Truth Table 3: Asynchronous inputs

$\overline{preset}$	$\overline{clear}$	Q_{t+1}
0	0	
0	1	
1	0	
1	1	

Part B: Constructing The J-K Flip-flop:

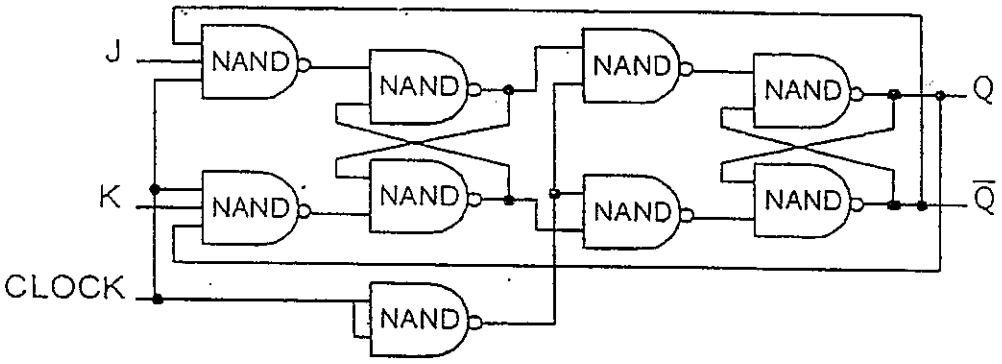


Figure 2: J-K Flip-Flop

Procedure:

1- Build the J-K flip-flop (master-slave) circuit as shown in fig. (2).

2- Confirm that the result of table (1) & (2) applies to fig. (2) circuit.

Assignment:

- If you have the option to choose either J-K flip-flops or S-C flip-flops for a circuit you want to build, which flip-flops would you use? Why?

Exp. (8)
Latches and Flip-Flops

Objective:

To introduce the basic flip-flops (FF) gates, and to study the characteristic of the different types of FF.

Introduction:

In our study of the combinational circuits we noticed that the output of such circuits at any time is dependent only on the levels present at the inputs at that time. On the other hand, a flip-flop is a sequential circuit or memory element in which its output has the ability to remain at one state even after the input condition that caused the state was changed. Also a flip-flop can be affected by the previous output while combinational circuits responds only to input changes. Flip-flops can be divided into clocked and unclocked flip-flops. If the FF output does not recognize input changes unless the circuit was clocked, this circuit is called a clocked flip-flop.

The flip-flop outputs are labeled Q and $\overline{Q}$. Under normal conditions these outputs will be always inverse of each other. Whenever the two outputs are equal, this state will be called invalid.

In this experiment we will start by building the NAND latch (S-C latch), then in the next stage we will use the NAND latch to build the S-C flip-flop. Finally, by adding an inverter to the S-C flip-flop we will end up with the D flip-flop.

Part A: NAND Latch:

The latch, shown in fig. (1), has two inputs (SET & CLEAR) and two outputs.

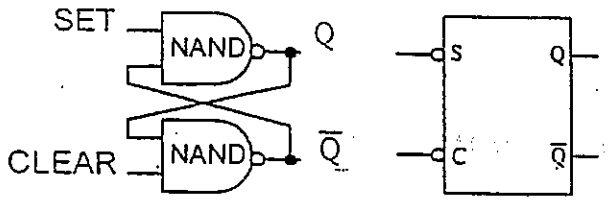


Figure 1: NAND Latch

Table 1: NAND Latch.

SET	CLEAR	Q_t	Q_{t+1}
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Table 2: NAND Latch summarized.

SET	CLEAR	Q_{t+1}
0	0	
0	1	
1	0	
1	1	

Procedure:

1- Connect the circuit shown in fig. (1).

2- Fill table (1) with the circuit's results. Q_t represents the Q lamp state before applying the specified inputs in the truth table while Q_{t+1} represents the same lamp after applying the inputs.

3- From table (1) derive the result for each state in table (2).

Part B: S-C Flip-Flop:

Is a clocked flip-flop that behaves like a NOR latch except that it does not permit the input to propagate through the circuit without a clock pulse.

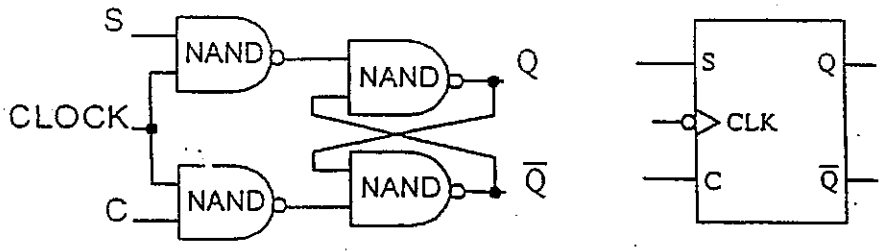


Figure 2: S-C Clocked Flip-Flop

Table 3: S-C Flip-Flop.

S	C	Q_t	Q_{t+1}
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Table 4: S-C Flip-Flop summarized.

S	C	Q_{t+1}
0	0	
0	1	
1	0	
1	1	

Procedure:

1- Connect the circuit shown in fig (2).

2- Fill table (3) with the circuit's results.

3- From table (3) derive the result for each state in table (4).

Part C: D Flip-Flop:

The DFF is also a clocked flip-flop (D is short for data). It has only one input. This flip-flop succeeded in eliminating the invalid state the previous flip-flops contained.

Exp. (10)
Application Digital Circuits
(Instructor Demo)

Objective:

To give the student a wide view of the applications of digital circuits.

Introduction:

It is important for any field studier to know what could the basic knowledge acquired accomplish to be able to grasp the wide scope of the field, its importance, and the reason to study it. Heading in this direction we would like to present some of the common applications in digital systems.

In the first part asynchronous up counter will be presented. Then the second part will cover adder/subtractor circuit. Finally using shift registers binary division will be studied.

Part A: Asynchronous Up-Counter:

The asynchronous counter shown in fig. (1) counts from 0-7. The same circuit can be modified to count down from 7-0 by feeding the clock from $\overline{Q}$ instead of Q. Notice that the lamp has to be read from Q in both cases.

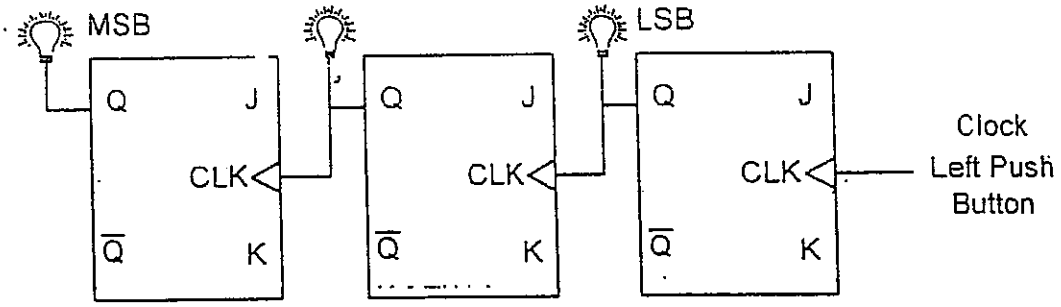


Figure 1: Asynchronous Up-Counter

Procedure:

1- Connect the circuit shown in fig. (1).

2- Using the left bush button (or the low speed clock), change the states of the counter from one number to the other.

Part B: Serial Shift-Registers:

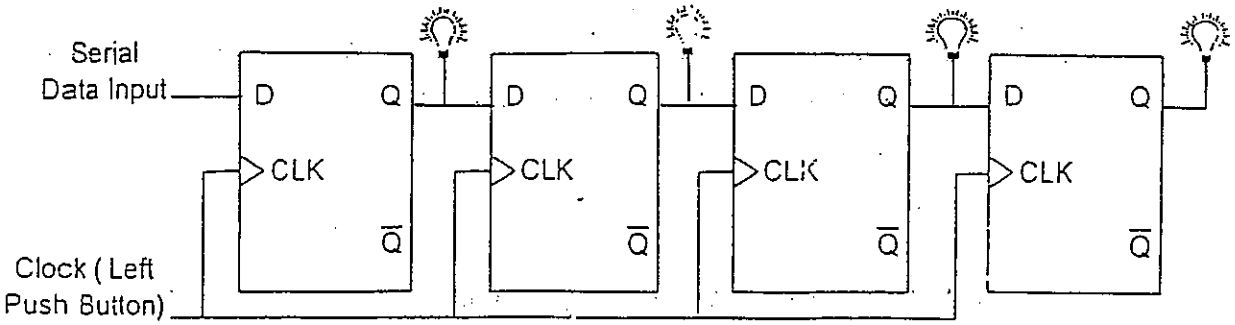


Figure 2: Right Shift Register

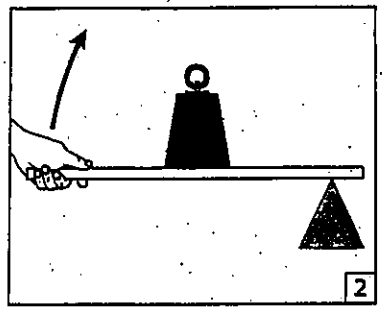
Procedure:

A serial shift register will load the inputs and transfer the data bit by bit. The shift register can either shift the bits to the right or the left. The circuit in fig. (2) represents a right shift register.

1- Connect the circuit shown in fig. (2).

2- Using the left bush button, move the bits from left to right. Use the Serial Data Input to load the new inputs to the first D FF.

3 Communication



2

3

Technology quiz

Student A

Complete the questions with the correct form of the verbs in brackets. Then ask Student B the questions and check his / her answers.

1 How tall _____ the world's tallest building? (be)
a > 500m b < 500 but > 450m c < 450m

2 What kind of lever _____ this? (be)
a first class b second class c third class

3 What's this _____? (call)
a bar code b PIN c zip code

4 Who _____ the first chemical battery in 1800? (produce)
a André-Marie Ampère b Alessandro Volta c James Watt

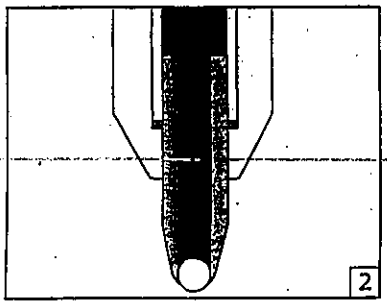
5 Which of these _____ Leonardo da Vinci (1452–1519) not design? (do)
a helicopter b speedboat c submarine

6 Who _____ the World Wide Web? (invent)
a Bill Gates b President Clinton c Tim Berners-Lee

7 What _____ the letters *RA* stand for in *RADAR*? (do)
a RAdio b Reflected Alternate c Red Alert

8 Who _____ the first yo-yo? (make)
a ancient Japanese b ancient Koreans c ancient Romans

Answers: 1c, 2b, 3a, 4b, 5b, 6c, 7a, 8c



2

Technology quiz

Student B

Complete the questions with the correct form of the verbs in brackets. Then ask Student A the questions and check his / her answers.

1 Which gas _____ a light bulb contain? (do)
a oxygen b nitrogen c argon

2 What's this device _____? (call)
a printer head b ballpoint pen c inkjet

3 Who _____ the first internal combustion engine in 1885? (produce)
a Henry Ford b Kiichiro Toyoda c Gottlieb Daimler

4 What _____ the letters *LA* stand for in *LASER*? (do)
a Los Angeles b Light Amplification c LAmp

5 When _____ the World Wide Web start? (do)
a 1990 b 1985 c 1995

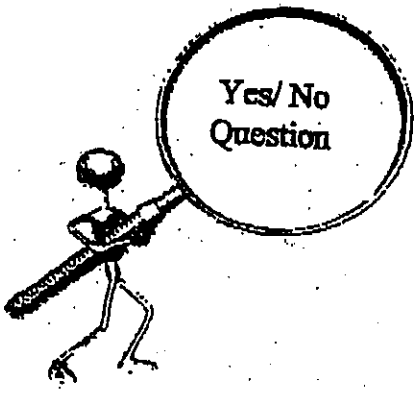
6 What _____ an MP3 player do? (do)
a transports people b plays music c plays games

7 Who _____ the mercury thermometer? (invent)
a Anders Celsius b Gabriel Fahrenheit c Baron Kelvin

8 Who _____ the first scissors? (make)
a ancient Egyptians b ancient Greeks c ancient Chinese

Answers: 1c, 2b, 3c, 4b, 5a, 6b, 7b, 8a

Question Types



Examples:

1- ARE you a student?
2- DO you like England?
3- HAVE you been here long?
4- CAN you understand people when they speak to you?
5- DO you understand people when they speak to you?
6- DID you go to the theatre last night?
7- COULD you help me with this exercise?
8- CAN you help me with this exercise?
9- WERE you at the football-match yesterday?
10-DID you GO to the football-match yesterday?

Exercises: Fill in the spaces with the correct auxiliary chosen from the list below:

(Do - Are - Is - Was - Can - Has - Did - Does - Have - Were)

1. _____ you carry these boxes with me?
2. _____ you see Maher yesterday?
3. _____ they finished their work yet?
4. _____ this your pen?
5. _____ you sure we have an exam tomorrow?
6. _____ Saad speak French?
7. _____ the students study biology this semester?
8. _____ Khalid written his homework?
9. _____ you alone at home last week end?