SECTION 1 DIGITAL LOGIC CIRCUITS

Structure **Page No.**

1.0 INTRODUCTION

The logic circuits are the basic building blocks of an electronic circuit. We have covered these concepts in Block 1 of MCS – 012. In this section, you must attempt to build the combinational circuits using tools created by "Alun Davies", Designer and Programmer of the software, called Logic. We hope that, you will find this useful and productive. We hope that it will generate interest in designing Logic Circuits on a small scale. In addition, you must do paper based design of some of the sequential circuit – related problems.

Our attempt in this section is to make you familiar with the package such that you are able to use the software as quickly as possible. We will discuss about how to build and test a successful logic circuit project. You must experiment with the package by yourself and attempt the problems; if doing this you will be gain valuable experience and your efficiency will increase. We hope that your experience with Logic Circuits design will be a productive experience. We have also presented an example for creation of logic circuit that you would like to test using Logic. We have also added practice problems that you can attempt. However, you are free to use any other tool freely available on Internet for this purpose.

1.1 OBJECTIVES

After completing this section, and doing all the practical problems given, you would be able to:

- design Combinational Circuit;
- design Sequential Circuits; and
- test your combinational circuits using Logic Software.

1.2 LOGIC GATES CIRCUIT SIMULATION PROGRAM

This program has been developed by "Alun Davies" as a simple window application primarily for Windows 3.1 but works mostly under Windows 95 and later. This was freely downloaded from the Website: www.pontybrenin.freeserve.co.uk/ logic/. If you have any comments or ideas for future enhancements of the software then please send them at the e-mail given on the Website.

You need to download the file logic zip and install it at your computer. On execution a simple screen appears having a two-menu option: File and Help. Select the "New Project" option of the File Menu to get the project window abeled "Logic Gates Construction" having a grid on the screen. It also includes a Gate and Connection Tool Bar. (Please refer to the Figure 1.1)

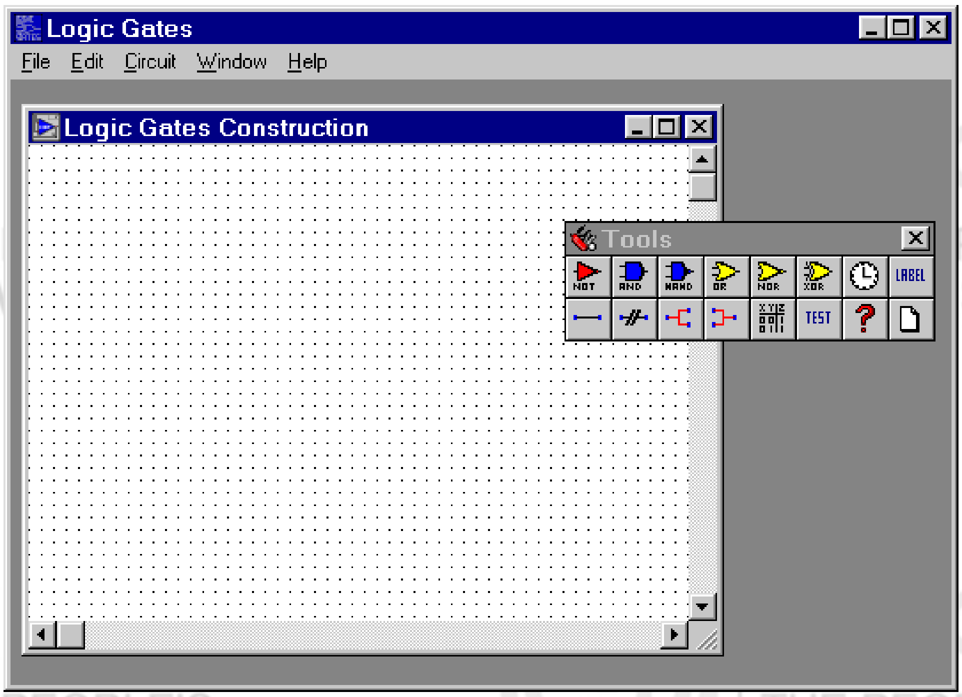
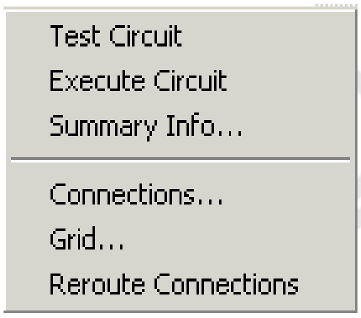


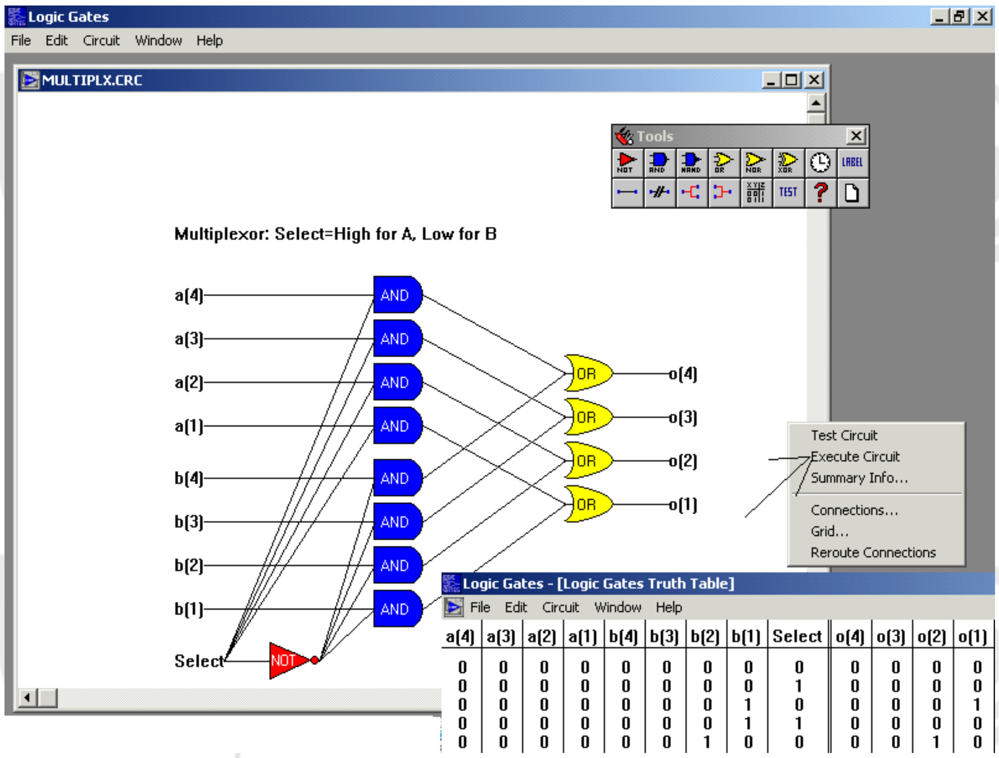
Figure 1.1 : New Project Window with Tool Bar

In this package, a lot of projects and circuits are given, so if you want to study the functionality of any given circuit, please open it from the file menu.

On right click of the circuit you will get options: Where Test Circuit is find that circuit having any error or not, if not you can proceed in execution of the circuit.



Like in the given in Figure 1.2 we have chosen MULTIPLEX.CRC from the projects. On clicking Execute circuit it displays the Truth table of circuit.



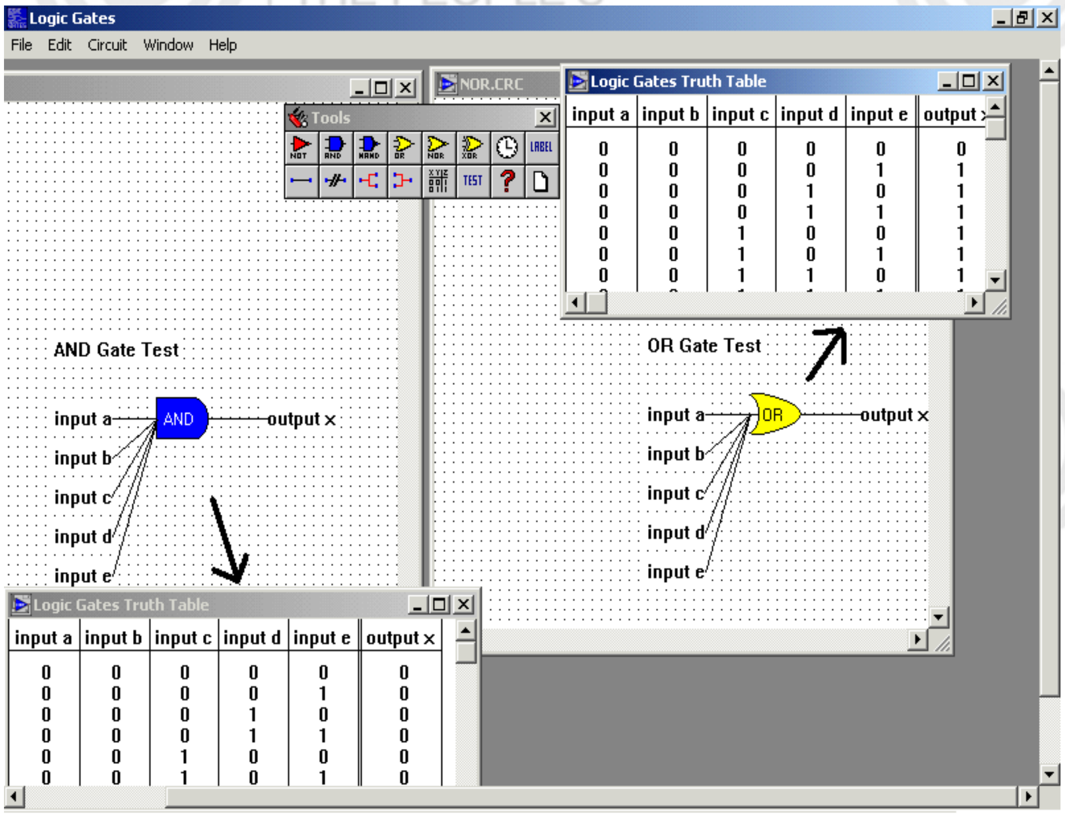
a[4]	a[3]	a[2]	a[1]	b[4]	b[3]	b[2]	b[1]	Select	o[4]	o[3]	o[2]	o[1]
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	1	0

Figure 1.2 : Circuit and its Truth Table

Please note that this Package cannot be used for making SEQUENTIAL Circuits, those can be made and tested on paper.

1.3 MAKING A LOGIC CIRCUIT USING LOGIC

Let us try to create the logic circuit given in the Figure 1.3 using LOGIC:



OR Gate Test truth table:

input a	input b	input c	input d	input e	output
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	1
0	0	0	1	1	1
0	0	1	0	0	1
0	0	1	0	1	1
0	0	1	1	0	1

AND Gate Test truth table:

input a	input b	input c	input d	input e	output x
0	0	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

Figure 1.3 : Window for Creating Logic Circuit

To create a logic circuit as above you need to perform the following steps:

Step 1: Create a new project using “File/ New Project” Menu options.

Step 2: Locate find AND gate symbol in the toolbox.

Step 3: Click on the AND gate button.

Step 4: Place the AND gate at desired locations by just clicking at that location.

Step 5: Similarly place OR gate on the desired location.

Note: (1) You can place as many gates as you like by just selecting and then clicking on screen.

(2) To delete any extra Gate Object, Right Click on that Object and select “Delete” from the menu that will get displayed.

(3) If you want to change the position of any gate then first deselect the Gate button on the tool bar by clicking on the selected button again. Now click on the desired object whose position you want to change and drag the pointer wherever you want to place it.

Step 6: Now add labels input ‘a’ ‘b’ ‘c’ ‘d’ and ‘e’ by first clicking on the label button of the tool bar. The names are given as you place the label on the project grid through the text dialog box.

Step 7: **Linking Gates:** To link a label and a gate (or gate to gate) select "Link" button from the toolbar. Press and hold mouse on the source gate/label and move the mouse to the destination connection object and release the mouse button.

A line should appear. You can remove a connection by selecting the Break connection button on the toolbar.

Step 8: **Testing a Circuit :** Once you have made all the connections test your circuit for correctness by pressing the Test button on the toolbar.

Step 9: **Executing a Circuit :** If testing is successful then execute the circuit to get the truth table. You can do it by selecting Execute button, which is labelled as a truth table in the tool bar.

Step 10: **Saving the Project :** Select File/ Save Project option to save with a suitable name with CRC for future use.

1.4 A REVISIT OF STEPS OF LOGIC CIRCUIT DESIGN

The MCS 012 course has discussed in detail about the procedure of making logic circuits. However, let us revisit the process. This design procedure results in a reliable and economical combinational logic circuit, if correctly applied. Whenever, you design a digital circuit you should follow the systematic steps given below for efficient design and write all the steps in your file.

1. Write precise circuit specification you understand.
2. Develop truth table on paper for circuit.
3. Identify the minterms corresponding to each row in the table.

4. Draw Karnaugh maps & forming groups of 1's on the Karnaugh map.
5. Write the reduced expression.
6. Convert the reduced expression into a realizable expression.
7. Draw the circuit diagram using software.
8. Test the Digital Circuits using given software.

A Complete Example

One of our objectives is that you should be able to design your very own combinational logic circuit. Here we are explaining you one example of Full Adder circuit.

Step 1: Write precise circuit specification you understand

Full-adder is a combinational circuit that forms the arithmetic sum of 3 input bits. Let us assume the 3 inputs are a_1, a_2, a_3 and 2 outputs are S, C.

Inputs:	3 input bits to be added (a_1, a_2, a_3)
Outputs:	Two output bits (S (sum bit) , C (carry bit))

Step 2: Development of a truth table on paper for circuit

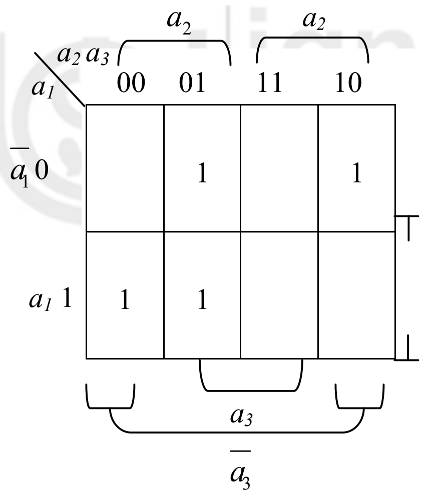
a_1	a_2	a_3
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

S	C	
0	0	0.
1	0	1.
1	0	2.
0	1	3.
1	0	4.
0	1	5.
0	1	6.
1	1	7.

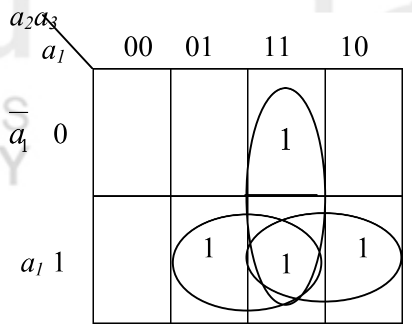
Step 3: Identifying the minterms corresponding to each row in the table

S = F1 (1,2,4,7)
C = F2 (3,5,6,7)

Step 4: Draw Karnaugh maps & forming groups of 1's on the Karnaugh map



(*a*) Sum bit (s)



Step 5: Write the reduced expression

$$S = \bar{a}_1\ \bar{a}_2\ a_3 + \bar{a}_1\ a_2\ \bar{a}_3 + a_1\ \bar{a}_2\ \bar{a}_3 + a_1\ a_2\ a_3$$

$$C = a_1\ a_2 + a_1\ a_3 + a_2\ a_3$$

Step 6: Convert the reduced expression into a realizable expression

$$S = \bar{a}_1\ \bar{a}_2\ a_3 + \bar{a}_1\ a_2\ \bar{a}_3 + a_1\ \bar{a}_2\ \bar{a}_3 + a_1\ a_2\ a_3$$

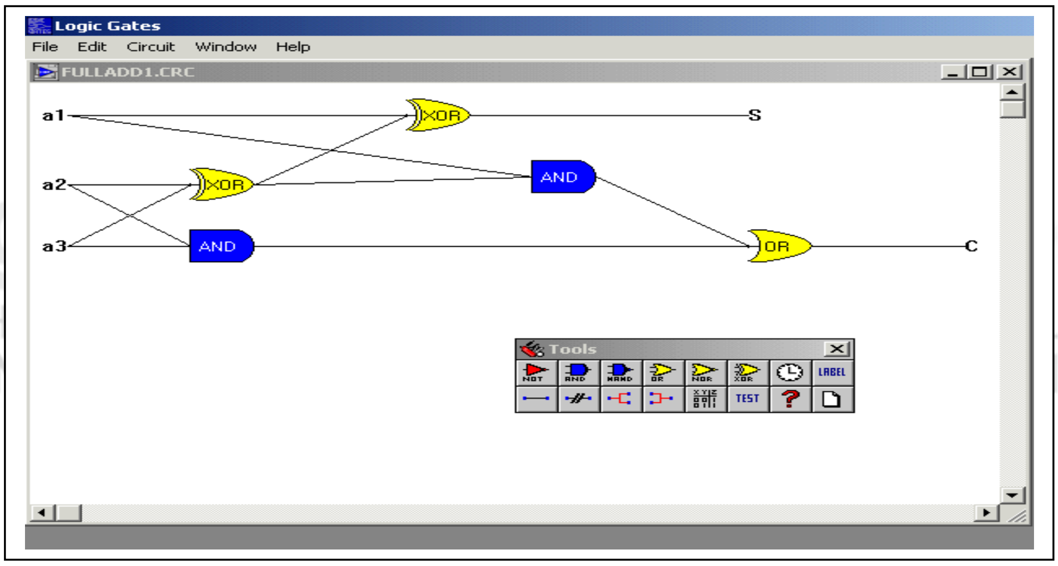
$$S = \bar{a}_1\left(\bar{a}_2\ a_3 + a_2\ \bar{a}_3\right) + a_1\left(\bar{a}_2\ \bar{a}_3 + a_2\ a_3\right)$$

$S = a_1$ XOR a_2 XOR a_3 (Refer to de Morgan's theorem)

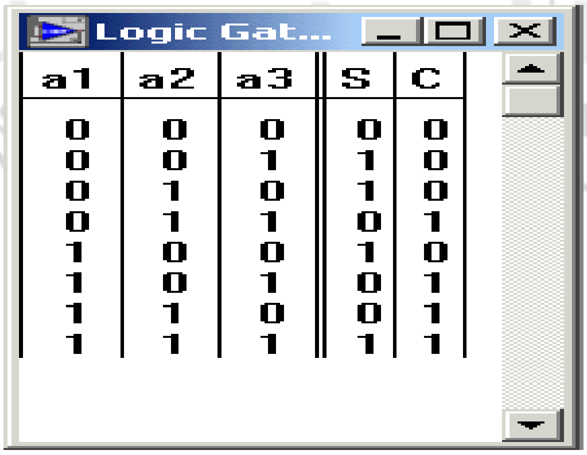
$$C = a_1\ a_2 + a_1\ a_3 + a_2\ a_3$$

$$C = a_2\ a_3 + a_1\left(a_2 \text{ XOR } a_3\right)$$

Step 7: Draw the circuit diagram using software



Step 8: Test the Digital Circuits using given software



a1	a2	a3	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Some Additional Tips

- **Highlighting Inputs and Outputs:** In a complex circuit, you would like to highlight the connections. You can do it by selecting show output () or show input () buttons. The output is shown in red colour and input is shown in blue colour.
- **Editing Circuits:** You can use options for copy, paste, or reposition gates etc.
- **Printing Circuits and Truth Tables:** To print a logic circuit or a truth table simply select the window containing the data and then select "File / Print...", option.
- Clocks are not used for any purpose in the circuit at present.
- Sample Circuits are available, use them for better understanding.

Trouble Shooting Tips

Trouble	Probable Solution
Circuit cannot be executed.	The circuit may be incorrect. Please check connections are made in right direction. This can be checked through right clicking on a gate for selecting the statistics button.
Circuit or Truth Table does not print.	Ensure proper printer is set up and check printer to be online. On some high-resolution printers circuits and truth tables will be printed smaller.
Clock does not work when circuit is executed.	The clock is not there for execution purposes. It is not supported by this version of "Logic Gates."
File does not load.	Please check that the file you are trying to load is a valid ".CRC" file and it exists.
File cannot be saved.	Check for the storage device, whether full?
Tool Box does not appear.	It has probably been closed. To open it again select "Window / Show" Menu option.

1.5 SESSION WISE PROBLEMS

We have allotted two practical sessions for you to draw the circuits using Logic Software. You must only draw the combinational circuits using this package, as it is not capable of drawing the sequential circuits. Design the following digital circuits and make a document in the file including all the steps defined in sub-section 1.3. You must come prepared with your design of the following problems on paper in order to take maximum advantage of the Lab session. We hope that with a little design preparation, you will be gaining a lot in these two sessions.

Session 1:

1. Design and implement the Exclusive-OR gate using AND, OR & NOT gates.

2. Design an "Alarm circuit" using only OR gate in which, if 'doors' OR 'windows' OR 'fire alarm' is activated, and then alarm sound should start.
   (Hint: Alarm is sounded if the output of the above circuit is 1. The Output will be 1 only if any of the OR conditions given above is true.)

3. We know NAND gate is universal gate but we need proof, so Design other gates like NOT, OR, AND & NOR using only NAND gates which will prove that NAND is universal gate.

4. Design a digital circuit whose output is equal to 1 if the majority of inputs are 1's. The output is 0 otherwise.

5. Design the following digital circuit.
   i) Half Adder
   ii) Half Subtractor
   iii) Full Subtractor

6. Design a logical circuit that will calculate the following function:

Inputs			Output
A	B	C	D
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

   Explain why your circuit is correct

7. Design a combinational circuit that takes a 3-bit number and the output of that circuit should be the square of the input Number.

8. Design a combinational circuit where input is a 4 bit number and whose output is the 2's complement of the input number.

9. Design the Encoder Circuit, which will convert decimal number to binary number.

Session 2:

10. Design Sequential Circuit, of clocked RS flip flop with 4 NAND gates.

11. Design Sequential Circuit of clocked D flip flop with AND and NOR gates.

12. Design a 8-bit counter using two 4-bit counters.

13. Design Linear Feed-Back Shift Register.

14. Design a logical circuit that will calculate the less-than (<) function for two 2-bit inputs. That is, if the inputs are A and B, each of whose values can be in the range 0-3 (i.e., 00-11 in binary), then the output should be 1 whenever A < B, and 0 otherwise. This circuit requires four inputs, referred to as a1, a2, b1, and b2. a1 and a2 represent a 2-bit number, as do b1 and b2. The output will be true if the decimal number represented by the pair a1a2 is less than the decimal number represented by b1b2. Design this circuit with an optimal number of gates.

15. A multiplexer circuit accepts N inputs and outputs the value of one of those inputs. The selection of which input goes out on the output is determined by a set of M control inputs. A multiplexer with M control inputs can steer up to 2^M inputs to a single output. Design 2-to-1 multiplexer.

16. A decoder has M inputs and up to 2^M outputs. If the logic values on the M inputs are interpreted as a binary number of value P, then the Pth output will be at logic 1 while all the others are at logic 0. Design 2-to-4 decoder.

1.6 SUMMARY

This section, although it involves only two practice sessions, requires a lot of groundwork done by you on the paper. These problems will provide you a foundation on the logic circuit design. You must attempt all these problems. This will also help your basics with MCS 012 subject and will provide confidence to you for any challenge in circuit design.