

DATE
------

COURSE/YEAR/GROUP
-------------------

NAME
------

**UNIVERSITY OF PLYMOUTH**

**Department of Communication and Electronic Engineering**

**COMPUTER SIMULATION USING “ELECTRONICS WORKBENCH”**

**LOGIC GATES**

**Introduction**

This lab sheet introduces you to the “Digital Side” of Electronics Workbench - basic gate types, the instruments available, and how to use them.

Note that the software is running with the ANSI symbols loaded. Basically, these are the American style symbols for logic gates, where the logic function is indicated by the shape.

Complete the truth tables for the following logic gates, and, if necessary, refer to these tables constantly as you progress through this simulated experiment.

Each gate has inputs A and B and output Z. (The inverter has only one input A)



AND



OR



NOT



NAND



NOR

INPUTS		O/P
A	B	Z
0	0	
0	1	
1	0	
1	1	

INPUTS		O/P
A	B	Z
0	0	
0	1	
1	0	
1	1	

INPUT	O/P
A	Z
0	
1	

INPUTS		O/P
A	B	Z
0	0	
0	1	
1	0	
1	1	

INPUTS		O/P
A	B	Z
0	0	
0	1	
1	0	
1	1	

NB If you know the truth tables already, fill in the tables.

Otherwise use one of the two circuits on pages 2 and 3.

## Selecting Libraries of Components using the Window Menu

When EWB starts up, the library of Passive components is selected. The other libraries are

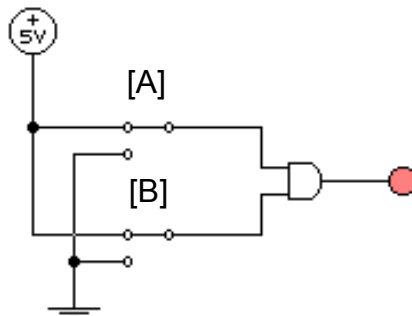
1. represented by the following row of icons.
- or 2. available from the Windows menu.

Click on each icon in turn to get an idea of the components available. There is also a useful Quick Reference Card for EWB version 4.



## Testing Truth Tables

Investigate each gate in turn using the following circuit. The logic gates are in the Gates library. The Control library – highlighted above, provides the single pole double throw switches in the circuit below. These are operated by a specified key on the keyboard. Highlight each switch in turn (click right), and then double click left. Assign key A to one switch and B to the other. We also need a small LED indicator from the Indicators library.

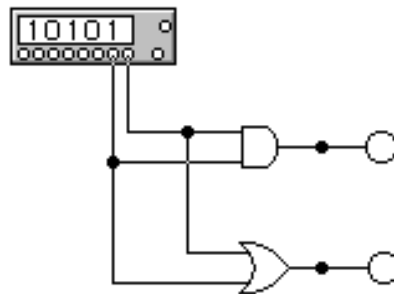


In the positions shown, inputs A and B are connected to 5V, ie Logic level 1. The alternative switch positions give Logic 0. Press the “Simulate” switch to activate the circuit, and then step through the truth table by using keys A and B (no need for capitals).

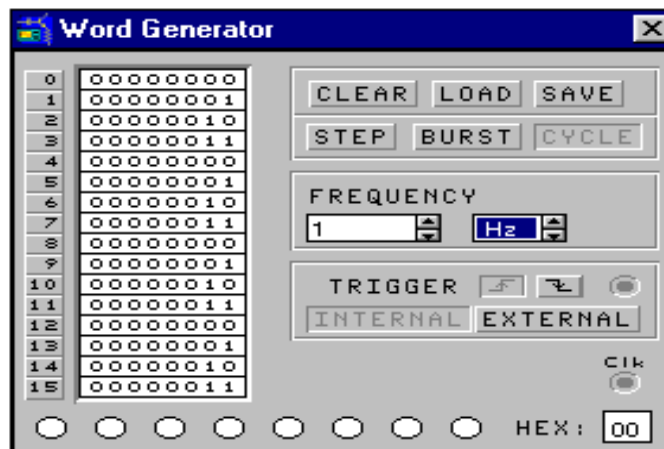
Replace the AND gate by another gate type to continue this test of the truth tables.

## Testing the Word Generator

Build the following circuit, dragging parts from the two appropriate libraries. The indicators will need to be rotated through 180 degrees by highlighting each on the work area and pressing CTRL R twice. The Word Generator is dragged off its shelf.



We will apply the same signals from the Word Generator to the AND and OR gates. It can store up to 16 8-bit words. As we only need a cycle of four as inputs to the two-input gates, repeat the pattern four times over. This is done by expanding the Word Generator (double click), and then overtyping the zeros where necessary, as shown below.



Try clicking on the STEP button. Each click should move to the next line of stored bit patterns. Watch the LED indicators carefully. You should be able to follow the AND and OR logic functions.

Try the BURST and CYCLE buttons on the Word Generator. What is the function of

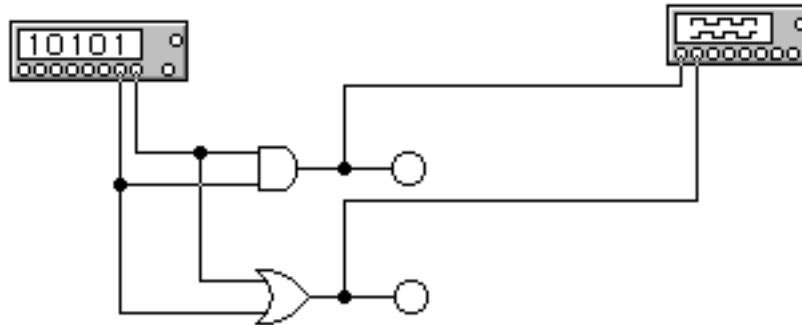
BURST .....

CYCLE .....

Reduce the frequency to 1 Hz, and observe the effect.

## Testing the Logic Analyser

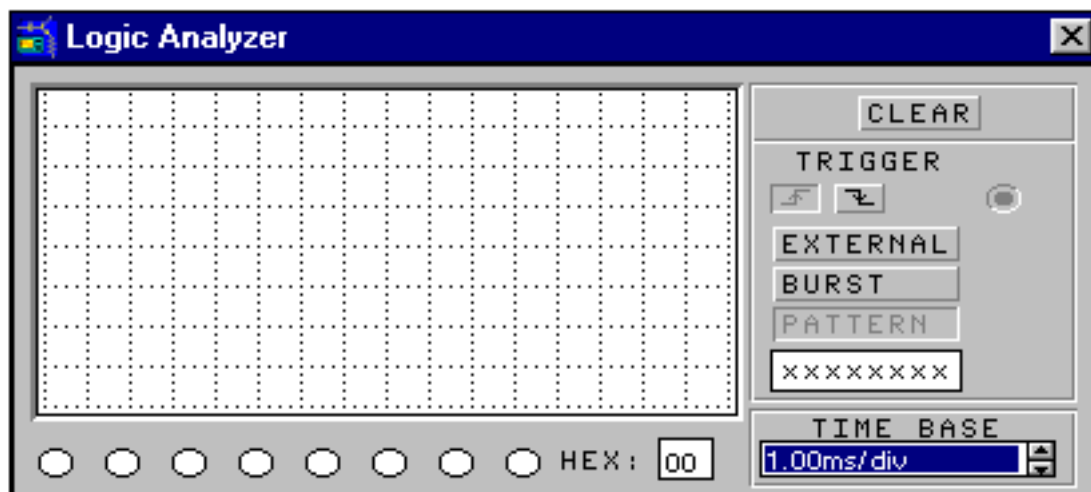
Drag the Logic Analyser off the shelf and add it to the circuit. 8 traces can be displayed. Use the two left hand inputs.



Expand the Logic Analyser by double clicking on it. We wish to display the full cycle of 16 inputs from the Word Generator.

What is the best timebase (seconds/division) to use?

.....



Step the Word Generator until the last line is highlighted, clear the logic analyser, and then give it 16 steps from the Word Generator. Hopefully, the first step displayed on the Logic Analyser will be the first line of the table.

Record the wave patterns on the instrument above, identifying which is which between AND and OR.

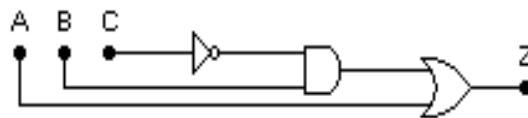
## The Logic Converter

This is useful when designing and building circuits with several logic gates.

### Starting a New Circuit

To begin building a new circuit, choose **New** from the **File** menu. A new workspace will open. (If you have any unsaved changes in your current circuit, you will be invited to save them).

Construct the following circuit by dragging components from the Gates library.



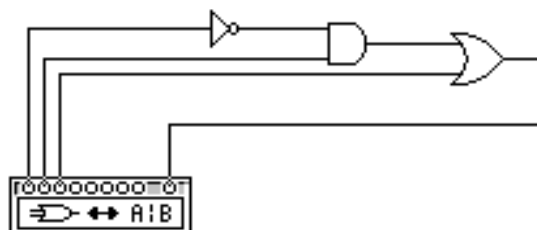
NB You can change the number of inputs on a gate by simply double clicking to open a dialogue box. Click on the arrows to increase or decrease the number of inputs. Then click OK to accept the number or CANCEL to ignore the change. Here we only require the default number of two inputs for the AND and OR gates.

### Trying out the Logic Converter

This natty instrument can convert a digital circuit between the three different representations:-

**Truth Table,**  
**Boolean Expression,**  
and **Circuit Schematic.**

Use it to test out the above circuit. Drag the Logic Converter from the instrument shelf, and place it below the circuit. Then attach it to the circuits inputs and outputs as shown.

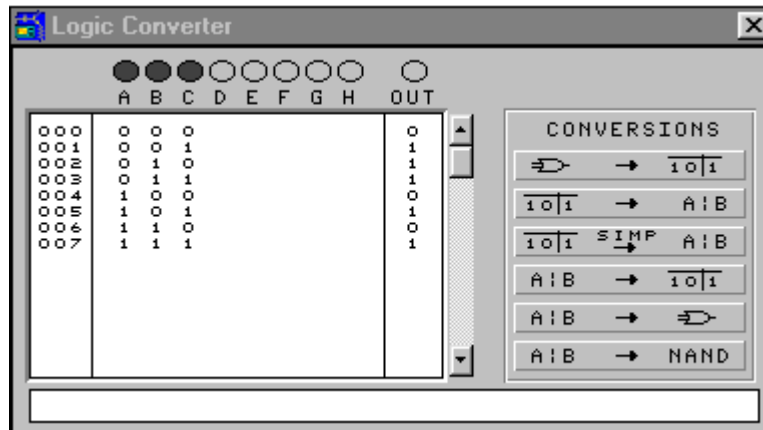


## Deriving a Truth Table for the Circuit

Open the Logic Converter by double clicking on it. Click the 'Circuit to Truth Table' button.



The Logic Converter will calculate and display the circuit's truth table.



## Converting a Truth Table to a Boolean Expression

Once you have a truth table, the Logic Converter can transform it into a Boolean function in the form of an algebraic expression.

Click the 'Truth Table to Boolean Expression' button.



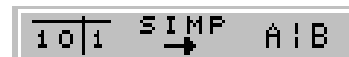
The Boolean expression will be displayed at the bottom of the Logic Converter. In this case:-

$$A'B'C + A'BC' + A'BC + AB'C + ABC$$

Note the primes, representing inversion.  $A'$  means NOT A, or  $\bar{A}$ .

Some expressions can be recalculated in a simpler form.

To try to simplify the expression, click the 'Simplify' button.



In this case, the expression can be simplified to  $A'B + C$

## Converting a Boolean Expression to a Circuit

To do this, click the 'Boolean to Circuit' button



This should redraw the circuit you started with, perhaps in a somewhat different layout.

Sketch the circuit in the space below, or get a printout and stick that in to these notes.

To get a printout, click **File** → **Print** → **Schematic**

You can also request other circuit components, eg the expanded Logic Converter.

## Creating a Circuit from a Truth Table

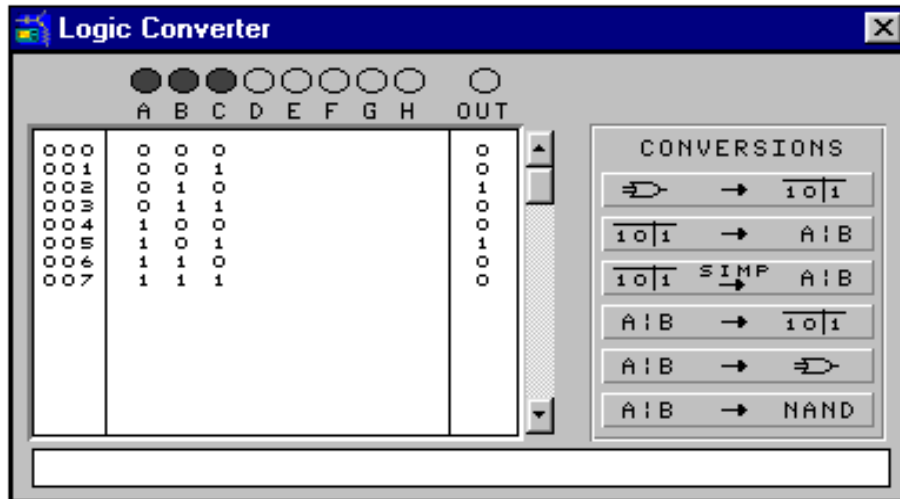
This is the most useful conversion for a circuit designer. Normally, you will have translated the client's specification into a truth table, and have then to produce a logic gate circuit to do the job.

This requires two conversions using the Logic Converter. We will practice using a different problem.

Start a new circuit using **File** → **New** (Save the previous circuit first if you wish). Drag a logic converter to the workspace and open it.

To create a truth table:-

1. Click the number of inputs you want, from A to H, at the top of the Logic Converter.  
The inputs are preset in standard binary count format. In this case, select A, B, and C.
2. The values in the output column are initially set to zero. Click the output values you want to change, and type a 1. In this example, type the pattern shown. (PTO).



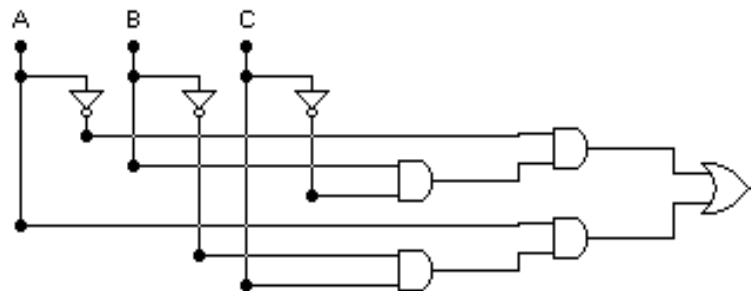
- Click the 'Simplify' button to convert the truth table to the simplest possible Boolean expression ( $A'BC' + AB'C$ )



- Click the 'Boolean to Circuit' button.



The resulting circuit will appear, selected, on the workspace. If you want to move it, point to one component and drag the circuit. Then click an empty spot to deselect it.



### Conclusion to Logic Converter

Look up your lecture notes and problem sheets on:-

Converting circuits between the Three Representations  
Minimisation Techniques for Circuit Design

Relate your notes to what you have learned from this lab sheet.

### Problem 1

Design a circuit using AND, OR, and NOT gates to produce the outputs  $Z_0$  and  $Z_1$  according to the table below. Draw the circuit in the space beside the table.

A	B	$Z_1$	$Z_2$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

To test your circuit, connect up the Word Generator, using the same pattern of inputs as on page 2. Connect up the Logic Analyser, and apply the inputs.

Sketch the two waveforms below.

### Problem 2

Devise and test a circuit to drive a set of traffic lights according to this sequence.

Use three inputs A, B, and C with a binary count of 000 to 111 in the table to generate the three outputs in the table

Red	Amber	Green
1		
1		
1		
1	1	
		1
		1
		1
	1	

A binary counter would be used in this situation to generate the inputs A, B, and C