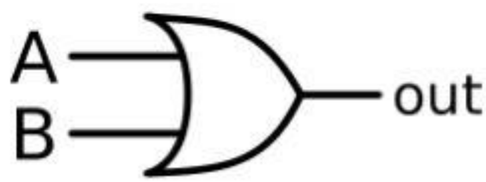
Lab Manual:
Digital Electronics

Using the Digilent Digital Electronics Board for NI ELVIS III

A	B	O
0	0	0
0	1	1
1	0	1
1	1	1



Lab 2: Truth Tables and Basic Logic Gates

NATIONAL INSTRUMENTS

© 2018 National Instruments
All rights reserved. Neither this resource, nor any portion of it, may be copied or reproduced in any form or by any means without written permission of the publisher.

National Instruments respects the intellectual property of others, and we ask our readers to do the same. This resource is protected by copyright and other intellectual property laws. Where the software referred to in this resource may be used to reproduce software or other materials belonging to others, you should use such software only to reproduce materials that you may reproduce in accordance with the terms of any applicable license or other legal restriction.

LabVIEW and National Instruments are trademarks of National Instruments.

All other trademarks or product names are the property of their respective owners.

Additional Disclaimers: The reader assumes all risk of use of this resource and of all information, theories, and programs contained or described in it. This resource may contain technical inaccuracies, typographical errors, other errors and omissions, and out-of-date information. Neither the author nor the publisher assumes any responsibility or liability for any errors or omissions of any kind, to update any information, or for any infringement of any patent or other intellectual property right.

Neither the author nor the publisher makes any warranties of any kind, including without limitation any warranty as to the sufficiency of the resource or of any information, theories, or programs contained or described in it, and any warranty that use of any information, theories, or programs contained or described in the resource will not infringe any patent or other intellectual property right. THIS RESOURCE IS PROVIDED “AS IS.” ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY AND ALL IMPLIEDWARRANTIES OFMERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS, ARE DISCLAIMED.

No right or license is granted by publisher or author under any patent or other intellectual property right, expressly, or by implication or estoppel.

IN NO EVENT SHALL THE PUBLISHER OR THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, COVER, ECONOMIC, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS RESOURCE OR ANY INFORMATION, THEORIES, OR PROGRAMS CONTAINED OR DESCRIBED IN IT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, AND EVEN IF CAUSED OR CONTRIBUTED TO BY THE NEGLIGENCE OF THE PUBLISHER, THE AUTHOR, OR OTHERS. Applicable law may not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Lab 2: Truth Tables and Basic Logic Gates

Nowadays, digital hardware is encountered in almost every aspect of our everyday life, being part of personal computers, household appliances, robots, television networks, etc.

Logic circuits are the building blocks of digital hardware. The logic circuits perform operations on digital signals and are usually implemented as electronic circuits where the signal values are restricted to a few discrete values. The most common are the binary logic circuits, where the only values are 0 and 1.

The three basic logic operations are:

- Logical **AND**
- Logical **OR**
- **NOT** operation (inversion)

The logic operations are implemented with logic gates. A logic gate is an electronic circuit made up of transistors. The information related to the logic gates and logic functions and be described by a *truth table*.

Learning Objectives

In this lab, students will:

1. Explore the behavior of different configurations of logic gates.
2. Configure and build circuits and PLD designs in Multisim.

Required Tools and Technology

Platform: NI ELVIS III	✓ View User Manual: http://www.ni.com/en-us/support/model.ni-elvis-iii.html ✓ View Tutorials: https://www.youtube.com/playlist?list=PLvcPIuVaUMIWm8ziaSxv0gwtshBA2dh_M
Hardware: Digilent Digital Electronics Board for NI ELVIS III	✓ View NI Digital Electronics Board Manual: http://www.ni.com/pdf/manuals/376627b.pdf
Software: NI Multisim 14.0.1 Education Version or newer	✓ Install Multisim: http://www.ni.com/gate/gb/GB_ACADEMICEVALMULTISIM/US ✓ View Help: http://www.ni.com/multisim/technical-resources/
Software: NI LabVIEW FPGA Vivado 2014.4	✓ Install: http://www.ni.com/download/labview-fpga-module-2015-sp1/5920/en/ **Note:** Digilent Driver (The installer above automatically downloads the installer below onto your computer) ✓ Navigate to: C:\NIFPGA\programs\Vivado2014_4\data\xicom\cable_drivers\nt64\digilent ✓ Install: install_digilent.exe

Expected Deliverables

In this lab, you will collect the following deliverables:

- Probe results
- Analysis of gate behavior
- Analysis of circuit behavior
- Circuit calculations
- Conclusion questions

Your instructor may expect you to complete a lab report. Refer to your instructor for specific requirements or templates.

1.1 Theory and Background

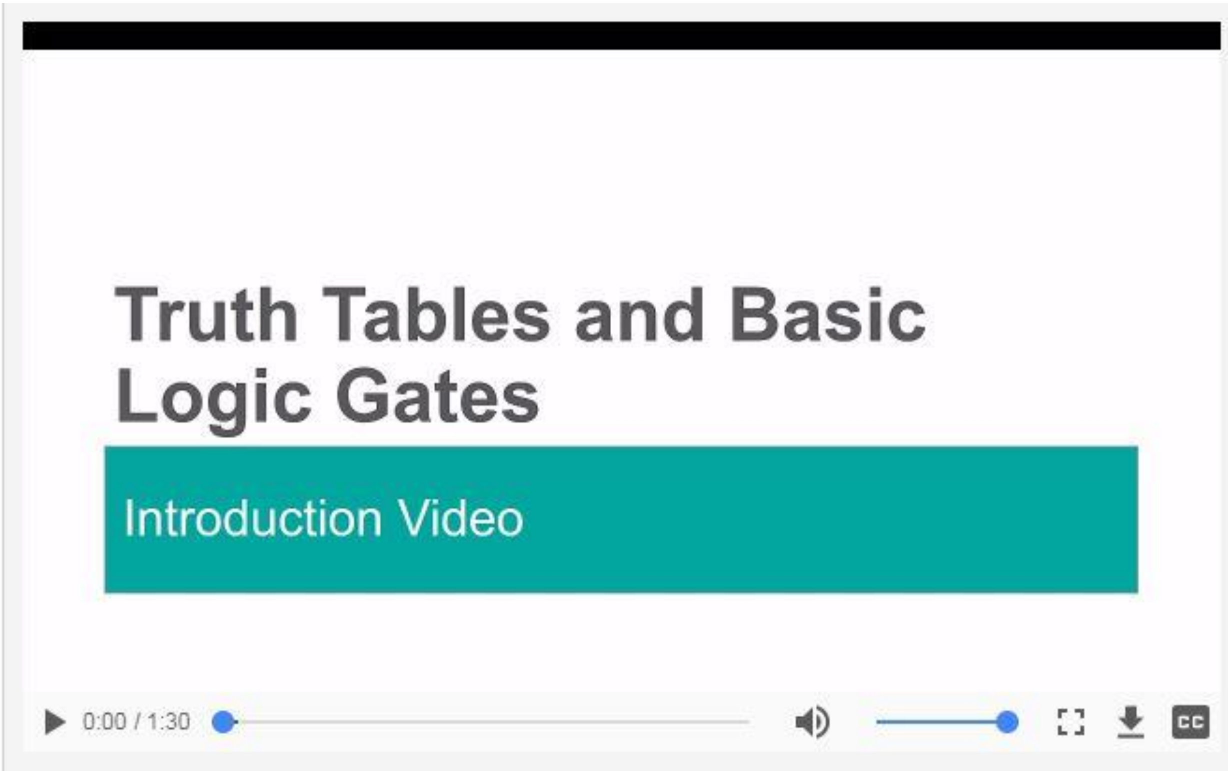


Figure 1-1 Video. View the video here: https://youtu.be/PhlGDrqgmj8

Video Summary

- Logic gates are the building blocks of all digital electronics
- AND and OR gates have at least two inputs and only one output
- NOT gates have one input and one output
- Inputs and outputs are expressed solely in binary (0’s or 1’s)

Truth Tables

One common way to express the particular function of a logic circuit is called a *truth table*. Truth tables show all permutations of the inputs with their corresponding output values in terms of logic level states. Logic level states are typically expressed as:

- 1 and 0
- HIGH and LOW
- True and False

This is an example of a truth table for two inputs:

A	B	O
0	0	0
0	1	1
1	0	1
1	1	1

Figure 1-2 Truth table for two inputs

A gate or logic circuit's truth table must have as many rows as there are possibilities of unique input combinations. For a single-input gate, like the inverter, there are only two input possibilities, namely 0 and 1. For a two-input gate there are four possibilities (00, 01, 10, and 11), and thus four rows for the corresponding truth table. For a three-input logic device, there are eight possibilities and so forth. The input columns are typically written in binary order as shown here:

A	B	C	O
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Figure 1-3 Truth table for three inputs written in binary

Logic Gates

Logic gates are physical devices that implement the Boolean functions of truth tables. The two most basic logic gates are the "AND" and the "OR".

- In the "AND" logic gate, the output is 1 if both the inputs for A and B are also 1. If one or all of the inputs for A and B are 0, then the resulting output is 0. This is summarized in the truth table below.

- Generally, the “AND” logic gate outputs the minimum value between the two input digits.
- The “AND” symbol is represented on the right. In this case, we can see two inputs (A and B) and one output.

A	B	O
0	0	0
0	1	0
1	0	0
1	1	1

Fig 1-4 AND Truth Table

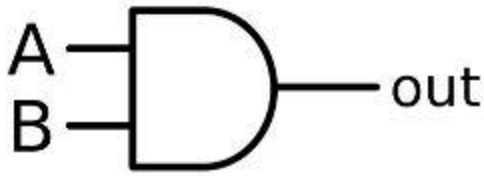


Fig 1-5 AND Logic Gate

- In the “OR” logic gate, the output is 0 if both the inputs for A and B are also 0. If one or all of the inputs for A and B are 1, then the resulting output is also 1. This is summarized in the truth table below.
- The “OR” logic gate outputs the maximum value between the two input digits.
- The “OR” symbol is represented below. As above, there are two inputs and one output.

A	B	O
0	0	0
0	1	1
1	0	1
1	1	1

Figure 1-6 OR Truth Table

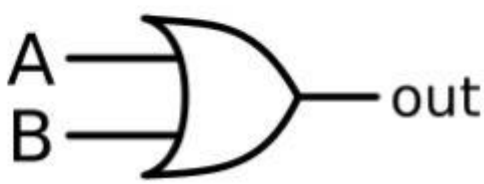


Figure 1-7 OR Logic Gate

1.2 Simulate: Building a Circuit with Multiple Gates

Circuit Example 1

Build the following circuit using multiple AND/OR Gates in Multisim:

- Place an **OR** gate and **two AND** gates from the **Misc Digital** group.
- Place **three INTERACTIVE_DIGITAL_CONSTANT**s from the **Sources** group.
- Place one **PROBE_DIG_RED** from the **Indicators** group.
- Wire them as shown.

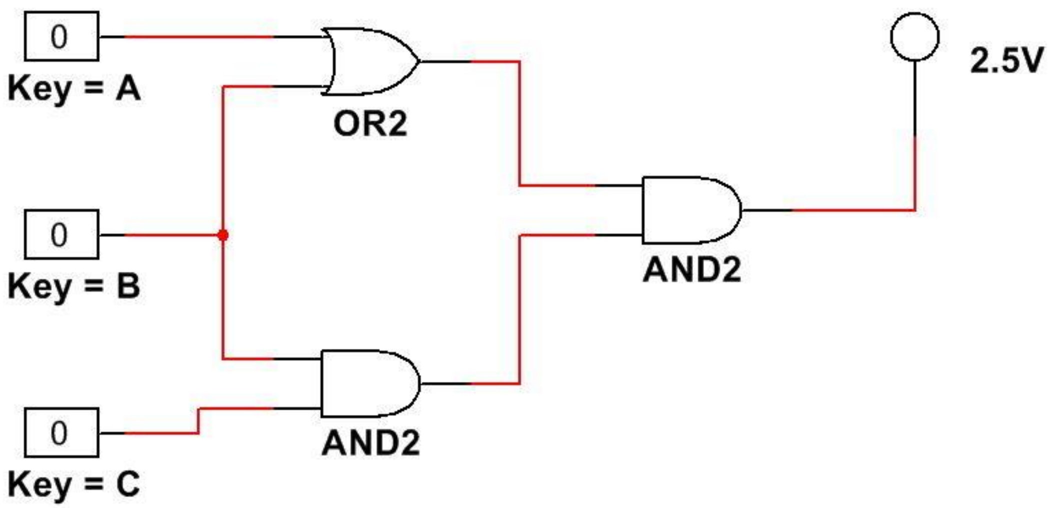


Figure 1-8 Circuit with AND OR Logic gates

- Click the **Run** button to begin simulating the circuit.

Figure 1-9 Run button

- Using the **A**, **B**, and **C** keys, vary the inputs into the circuit.

1-1 Record the results, as indicated by the probe, in the following truth table.

A	B	C	O
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

- When you're done, stop the simulation by clicking the **Stop** button.

Figure 1-10 Stop button

1.3 Simulate: Building a Circuit with Multiple Gates

Circuit Example 2

Now build a new PLD circuit using the same gates as before, but in a different configuration:

- Place an **OR** gate and **two AND** gates from the **Misc Digital** group.
- Place the input connectors **SW0, SW1, SW2.**
- Place the output connector **LED0.**
- Wire them as shown.

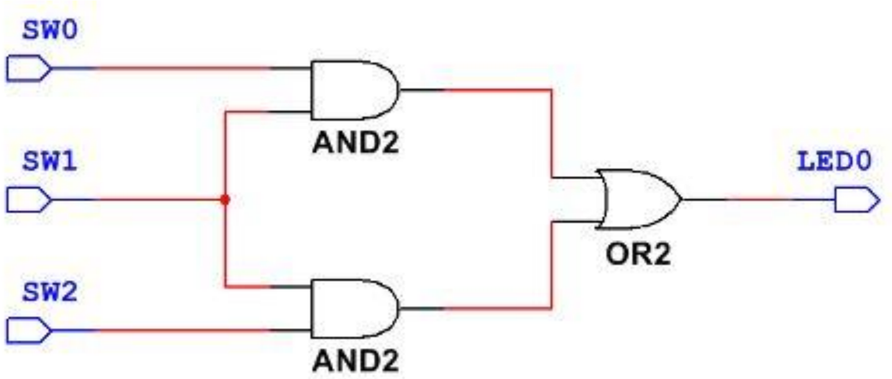


Figure 1-11 PLD circuit

- Export the circuit to the Digital Electronics Board. Using the switches on the board, vary the inputs into the circuit.

1-2` Given that the same gates are being used, predict if this circuit will demonstrate the same behavior of the previous one.

____

____

____

____

1-3 Record the results, as indicated by the probe, in the following truth table.

SW0	SW1	SW2	O
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

1-4 Explain the behaviour of this circuit. Did it match your prediction?

____

____

____

____

1.4 Exercise: Determining a Circuit from a Truth Table

Truth Tables

We have now observed how truth tables define the behavior of a given circuit. However, we often know what type of behavior we need to implement, and we need to figure out what configuration of gates would give us this behavior. This can be a lot more complex.

In this part, you'll look at a few simple truth tables and determine their circuits.

Given the following truth table, determine which configuration of gates would give you these results. You can:

- Use trial and error simulation in Multisim.
- Calculate different circuits on paper.
- Use whatever other methods you find useful.

A	B	C	O
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

Figure 1-12 Truth table

Note: Provide a screenshot, take a picture, or create a sketch of your design.

- When you've picked a circuit design that you think will result in this behavior, simulate it in Multisim.

1-5 Did your circuit design work as expected? If not, why?

____

____

1-6 Is there more than one configuration of gates that would give you this result?

____

____

Given the following truth table, determine which configuration of gates would give you these results.

You can:

- Use trial and error simulating in Multisim.
- Calculate different circuits on paper.
- Use whatever other methods you find useful.

A	B	C	O
0	0	0	0
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	1

Figure 1-13 Truth table

Note: Provide a screenshot, take a picture, or create a sketch of your design.

- When you've picked a circuit design that you think will result in this behavior, simulate it in Multisim.

1-7 Did your circuit design work as expected? If not, why?

____

____

1-8 Is there more than one configuration of gates that would give you this result?

____

____

1.5 Conclusion

1-9 Why is it useful to represent the behavior of a gate or series of gates in a table?

____

____

____

____

1-10 What is the difference between “AND” and “OR” logic gates?

____

____

____

____

1-11 The basic logic operations are:

A. AND
B. OR
C. NOT
D. All of the above

1-12 Truth tables:

A. Express the particular function of a logic circuit
B. Show all possible permutations of inputs and corresponding output values
C. Are quantified as logic level states
D. All of the above

1-13 A truth table with 3 inputs can have how many possible outputs?

A. 8
B. 4
C. 6
D. 12

1-14 AND logic gates have:

A. One input and two outputs
B. Two inputs and one output
C. A current inverter
D. None of the above

1-15 OR logic gates output:

A. The minimum value between the two inputs
B. The maximum value between the two inputs
C. The average of the two inputs
D. The output does not have any direct relationship to the inputs