

**VALPARAISO UNIVERSITY**  
**ELECTRICAL AND COMPUTER ENGINEERING DEPARTMENT**

**ECE 221**

**Design Project #1 - Universal Gates**

**FALL 2003**

**Introduction:** In class we saw how any combinational logic function may be realized using AND, OR, and NOT gates. In this design project we verify that the NAND gate is a universal building block. That is, the operation is functionally complete, in that any combinational logic function may be realized solely with NAND gates. We also verify that the NOR gate, which is the dual of the NAND, is also a universal building block. Finally, we investigate the use of the NOR gates exclusively in the design of a simple digital circuit.

In the first laboratory exercise you learned how to create and verify a simple circuit using the Mentor Graphics tools. This design project will direct you step by step through the setup and simulation of a more complicated circuit to verify that both NAND and NOR gates can be used to create any other combinational circuit. With that background, you will complete this project by designing and simulating a switch counter built entirely of NOR gates.

**SECTION I. Universal NAND Gate**

The objective of this section is to create and simulate a number of small circuits to verify that the NAND gate is a universal gate. The functions below (which are also found in your book) illustrate that the NAND gate can be used to create equivalent INVERTER, AND and OR gates.

$$\begin{aligned}f_{NAND}(X, X) &= \overline{X \cdot X} = \bar{X} = f_{INV}(X) \\ \bar{f}_{NAND}(X, Y) &= \overline{\overline{X \cdot Y}} = X \cdot Y = f_{AND}(X, Y) \\ f_{NAND}(\bar{X}, \bar{Y}) &= \overline{\bar{X} \cdot \bar{Y}} = X + Y = f_{OR}(X, Y)\end{aligned}$$

The attached schematic sheet is partially completed showing the circuit for an INVERTER implemented using one NAND gate. There are two empty boxes for the NAND implementations of the OR and AND gates which you will need to design from the above equations. Using the attached sheet as a guide, create a schematic that has all three implementations on it. Once the schematic is completed, use simulation to verify your implementations are correct. The procedure for creating and simulating the design is described below. If you cannot remember the commands that must be done to complete a specific step, refer back to the tutorial you did in Lab #1. If a new command is needed, it will be described in detail.

1. Open Design Architect and open a new design (using the **Open Sheet** option). Go to the directory where you want to save your Mentor files and make the **Component Name**:

/<your mentor director>/uni\_nand

DO NOT change the sheet option from sheet1. If you do, it will cause you problems when you try to simulate the design.

2. Add the VU title block and make the appropriate changes to the attributes as shown below:

change DESIGN NAME to **Universal: NAND**  
change STUDENT NAME to <your name>  
change DATE to **9/22/03**  
change COURSE to **ECE 221**

3. Add the components in the design as shown in the schematic. Each of the components can be found in **gen\_lib** library.

**nand2** - two-input NAND gate  
**inv** - INVERTER gate  
**or2** - two-input OR gate  
**and2** - two-input AND gate  
**portin** and **portout** - input and output ports

You will need to add the appropriate 2-input NAND gates in each box to complete the design. Use the equations as a guide along with the figures on page 52 in your textbook.

4. Connect the components: Now that you've placed all the components they must be connected together by adding wires. Refer back to the tutorial if you can't remember how to draw wires.

5. Change the port names for the inputs and outputs to match those shown in the schematic.

6. Once you've completed the design, use the **Check Sheet** option to make sure there are no problems (you should have no warnings or errors) and then save your schematic.

7. Instead of applying stimulus in QuickSim (like the tutorial) you will create a *stimulus file* that can be used for all three parts of this design project. A stimulus file is simply a text file that contains the a set of values that are applied to the inputs at specific times. Open a text editor and place the following lines in it. Save the file as **dp1.forces** in your Mentor designs directory.

```
// Stimulus file for Design Project #1
//
FORCE X 0 0 -Fixed
FORCE Y 0 0 -Fixed

FORCE Y 1 100 -Fixed

FORCE X 1 200 -Fixed
FORCE Y 0 200 -Fixed

FORCE Y 1 300 -Fixed
```

Each line in the file describes one value placed on one input. After the keyword FORCE is the name of the input, followed by the value to be placed on the input, then the absolute time to put the value on the input and finally the strength of the input value.

8. In another command window, open the simulator (Quicksim II). Set up the trace window so that the inputs and outputs appear in the following order: **X, Y, INV, INV\_NAND, AND, AND\_NAND, OR, OR\_NAND**.

9. To apply the stimulus file, place your cursor in the trace window, click on it and then type:

```
dofile dp1.forces
```

A dialog box will appear as you are typing. Hit enter and the forces in the file will be applied to your inputs. Run the simulation for 400ns and verify that each pair of outputs match. If they don't, go back and check your design.

10. Once you have verified the outputs match, print out your design schematic and the waveforms.

**SECTION II. Universal NOR Gate**

The objective of this section is to create and simulate a number of small circuits to verify that the NOR gate is an universal gate. Complete the functions below to illustrate that the NOR gate can be used to create equivalent INVERTER, OR and AND gates.

$$f_{NOR}(X, X) = \underline{\hspace{4cm}} = \bar{X} = f_{INV}(X)$$

$$f_{NOR}(\bar{X}, \bar{Y}) = \underline{\hspace{4cm}} = X \cdot Y = f_{AND}(X, Y)$$

$$f_{NOR}(X, Y) = \underline{\hspace{4cm}} = X + Y = f_{OR}(X, Y)$$

- Using the functions above and the schematic you created in the previous section as a guide, create another schematic (**uni\_nor**) to show that the NOR gate is indeed a universal gate. In the title block change the project name to **Universal: NOR**. Follow the same procedure outlined in Section I with inputs to the schematic as **X** and **Y** and outputs as **INV**, **INV\_NOR**, **AND**, **AND\_NOR**, **OR**, **OR\_NOR**. The 2 input NOR gate in the **gen\_lib** is **nor2**.
- Simulate your circuit using the same stimulus file and execution time (400 nsecs) as used in previous section. Arrange the waveforms in the Trace window in the following order, from top to bottom: **X**, **Y**, **INV**, **INV\_NOR**, **AND**, **AND\_NOR**, **OR**, **OR\_NOR**. Verify that your design is working properly and then print out the waveforms along with the design schematic.

**SECTION III. Design of a Half Adder**

In this section you are to create a digital circuit that acts as a half adder using only 2-input NOR gates. The system should add two one bit numbers, **X** and **Y**. The result of the addition is given by two output signals, **SUM** and **COUT**. The output **SUM** is asserted if the addition produces an asserted value in the sum bit, the output **COUT** is asserted if the addition produces a carry out value.

- Complete the truth table below describing the circuit operation.

<b>X</b>	<b>Y</b>	<b>SUM</b>	<b>COUT</b>
0	0		
0	1		
1	0		
1	1		

From the truth table write the two boolean expressions in POS form.

SUM = \_\_\_\_\_  
 COUT = \_\_\_\_\_

- Create a schematic (call it **half\_add**) that implements the circuit (both outputs). You are to only use 2-input NOR gates for your implementation and minimize the number of gates used.
  - In the title block change the project name to **Half Adder**.
  - For this design we will be using components from the the Board Process Libraries (BPL) not the general library (GEN\_LIB). To access the BPL select the **Library** button on the palette. When the different libraries appear, choose **Glue Logic**.

2c. A box will appear with all of the different library families. Move the cursor down and select the **74LS** family. Then hit **List Parts**.

2d. Another box will appear with all of the parts in that library. Choose the **74LS02** part and hit **OK**. (this is a 2-input NOR gate) The part should now appear in the **By Component Name** field.

2e. If you just hit OK right now, the symbol for the NOR gate would be the IEEE standard symbol (a box). To change the symbol to look like a traditional NOR gate choose the **Refine Component** option near the bottom of the library window. When the Refine window appears select the **ANSI\_ALT** option in the **Symbol** area and hit **OK**.

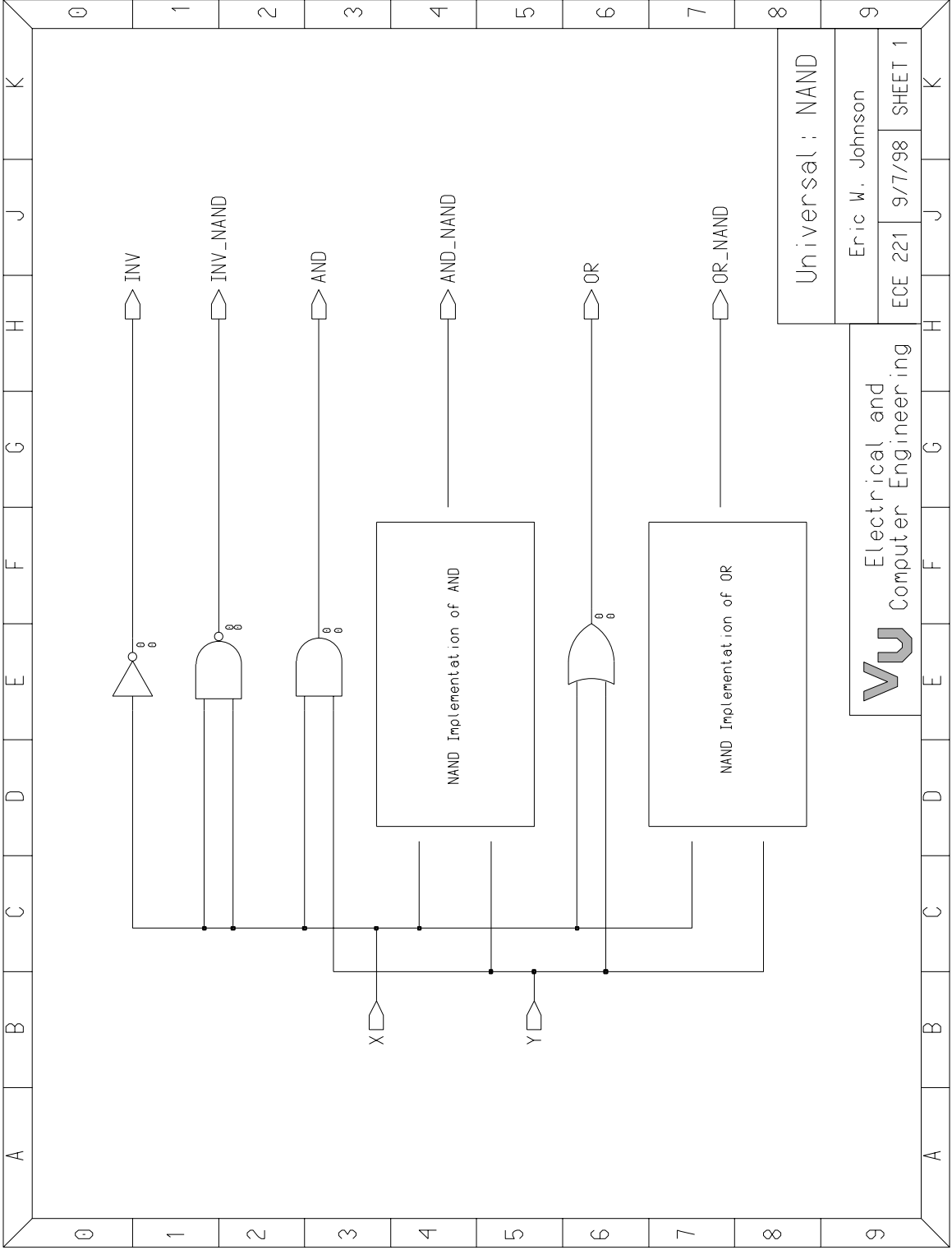
2f. Finally hit **OK** in the library window. The gate should appear and can be placed in your schematic. The parts show up as gates even though they are a part of a specific integrated circuit (74LS02). To add additional parts you can use the copy command on the palette.

2g. The input and output ports can be found in the BPL under the Connectivity Symbols library. Once you have added the ports change the inputs to **X** and **Y** and the outputs, **SUM**, and **COUT**.

3. Simulate your circuit using the same stimulus file as used in Sections I and II. Arrange the waveforms in the Trace window in the following order, from top to bottom: **X**, **Y**, **SUM**, and **COUT**. Run the simulation for 400ns. When your design is correct (matches the specification), print out your schematic and the waveforms.

#### **SECTION IV. WHAT TO TURN IN:**

1. Design project header page containing copies of the conceptual portions of the project and the signed honor code.
2. A copy of design schematic showing the universal NAND gate and the output waveforms verifying the NAND gate as a universal building block.
3. A copy of design schematic showing the universal NOR gate and the output waveforms verifying the NOR gate as a universal building block.
4. A copy of design schematic for the half adder and the output waveforms verifying the adder's operation.



# ECE 221 Digital Logic Design Design Project #1: Universal Gates

September 22, 2003

Name: \_\_\_\_\_

1. NOR functions:

$$f_{NOR}(X, X) = \underline{\hspace{10em}} = \bar{X} = f_{INV}(X)$$

$$f_{NOR}(\bar{X}, \bar{Y}) = \underline{\hspace{10em}} = X \cdot Y = f_{AND}(X, Y)$$

$$\bar{f}_{NOR}(X, Y) = \underline{\hspace{10em}} = X + Y = f_{OR}(X, Y)$$

2. Completed truth table for half adder.

X	Y	SUM	COUT
0	0		
0	1		
1	0		
1	1		

3. Minimized boolean expressions in POS form for the half adder

SUM = \_\_\_\_\_  
COUT = \_\_\_\_\_

Honor Code Pledge:

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Signature: \_\_\_\_\_

Please staple this sheet to the front of your assignment