# CPE 2211 COMPUTER ENGINEERING LAB

# EXPERIMENT 3 LAB MANUAL

Revised: J. Tichenor FS19

# INTRODUCTION TO FIELD PROGRAMMABLE GATE ARRAYS AND LOGIC

## OBJECTIVES

In this experiment you will

- Become familiar with Field Programmable Gate Arrays (FPGA).
- Use the suite of EDA tools supplied by Altera: Quartus-II and ModelSim.
- Use the Altera FPGA DE2 Board verify in hardware with an FPGA.

## LAB REPORTS

The format of lab reports should be such that the information can be used to reproduce the lab, including what values were used in a circuit, why the values were used, how the values were determined, and any results and observations made. This lab manual will be used as a guide for what calculations need to be made, what values need to be recorded, and various other questions. The lab report does not need to repeat everything from the manual verbatim, but it does need to include enough information for a 3rd party to be able to use the report to obtain the same observations and answers. Throughout the lab manual, in the Preliminary (if there is one), and in the Procedure, there are areas designated by **Qxx followed by a question or statement**. These areas will be **bold**, and the lab TA will be looking for an answer or image for each. These answers or images are to be included in the lab report. The lab TA will let you know if the lab report will be paper form, or if you will be able to submit electronically.

## PURPOSE

Familiarization with Altera FPGAs. In this exercise you will use an Altera FPGA DE2 (Development and Education) board to download a simple circuit which will act as a signal source for experimenting with the digital capability of the Mixed Signal Oscilloscope (MSO). The digital inputs are a smaller and a less complex version of a device normally called a Logic Analyzer. The purpose of the board is to provide the ideal vehicle for learning about digital logic, computer organization, and FPGAs. Altera FPGA's are reprogrammable which means they must be loaded with data which configures the part for a specific design each time power is applied. The DE2 board features a state-of-the-art Cyclone® II 2C35 FPGA in a 672-pin package. All important components on the board are connected to pins of this chip, allowing the user to control all aspects of the board's operation. You will use the schematic of the circuit designed in Lab 2 and download it onto the FPGA and test the successful operation of the design on the board.

## REFERENCES

Altera Cyclone™ II Device Handbook: The Altera FPGA device handbook gives all the information required for the Cyclone II FPGAs from Altera (this is available in acrobat format at https://www.altera.com/products/fpga/cyclone-series/cyclone-ii/support.html). The acrobat format can be read with Adobe's acrobat reader (or other PDF file readers).

## BACKGROUND

### FPGA

Before attempting this exercise you should have completed the design and simulation of primitive gates (Lab 2). The **Altera data book** is a good reference with a wealth of material on their line of FPGAs. An FPGA (Field Programmable Gate Array) is a user-programmable logic device that can be configured to perform a variety of complex logic operations. It is similar to the PLAs studied in class, but is much more powerful. Figure 1 shows the basic layout of FPGAs. The FPGAs consist of an array of logic blocks which can be wired together in a user-defined manner using Programmable Interconnects. These internal logic blocks communicate with external hardware through I/O Blocks on the outer edge of the FPGA.

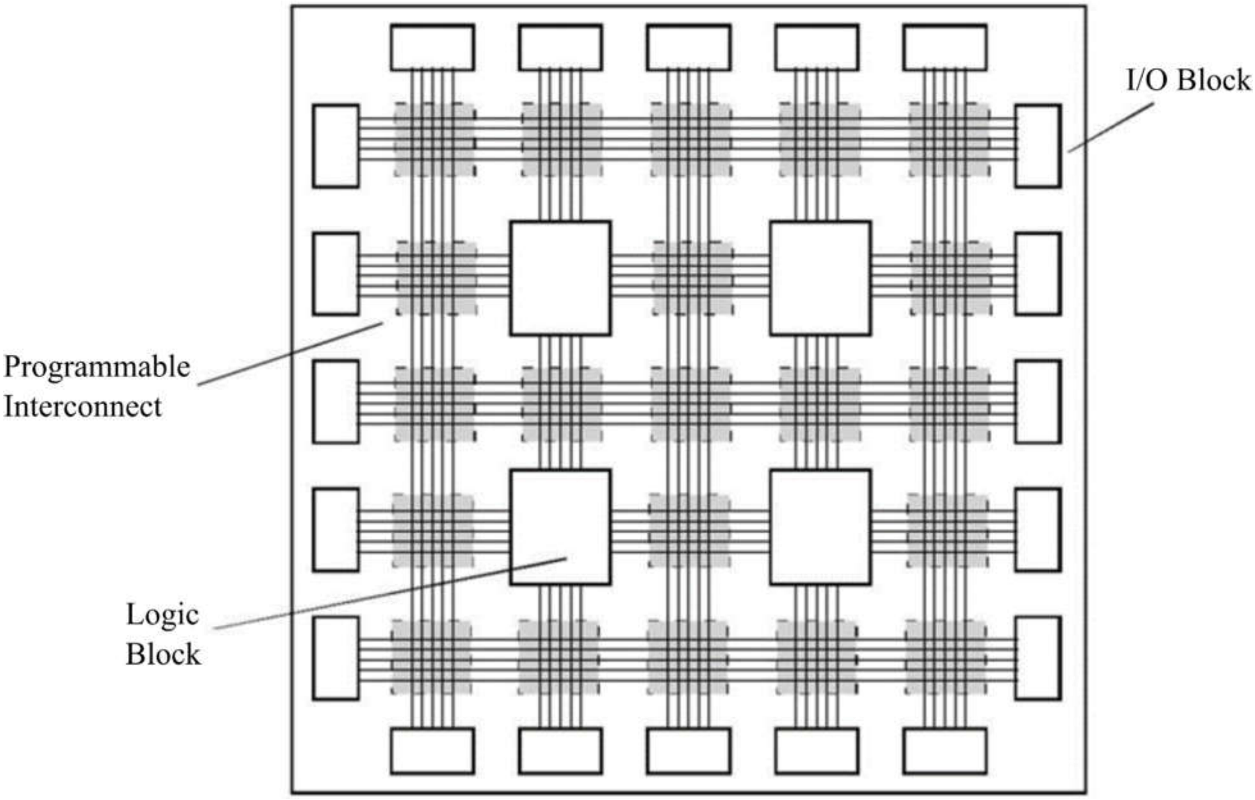


*Figure 1: FPGA Layout.*

The smallest unit of logic in the Cyclone II architecture, the Logic Block, is shown in Figure 2. The Logic Block is compact and provides advanced features with efficient logic utilization. Each Logic Block features: (a) a four-input look-up table (LUT), which is a function generator that can implement any function of four or fewer variables, (b) a programmable register, (c) a carry chain connection, (d) a register chain connection, (e) the ability to drive all types of interconnects: local, row, column, register chain, and direct link interconnects, (f) support for register packing, and (g) support for register feedback. The detailed information, although not needed for this lab, can be obtained from the reference link given earlier.

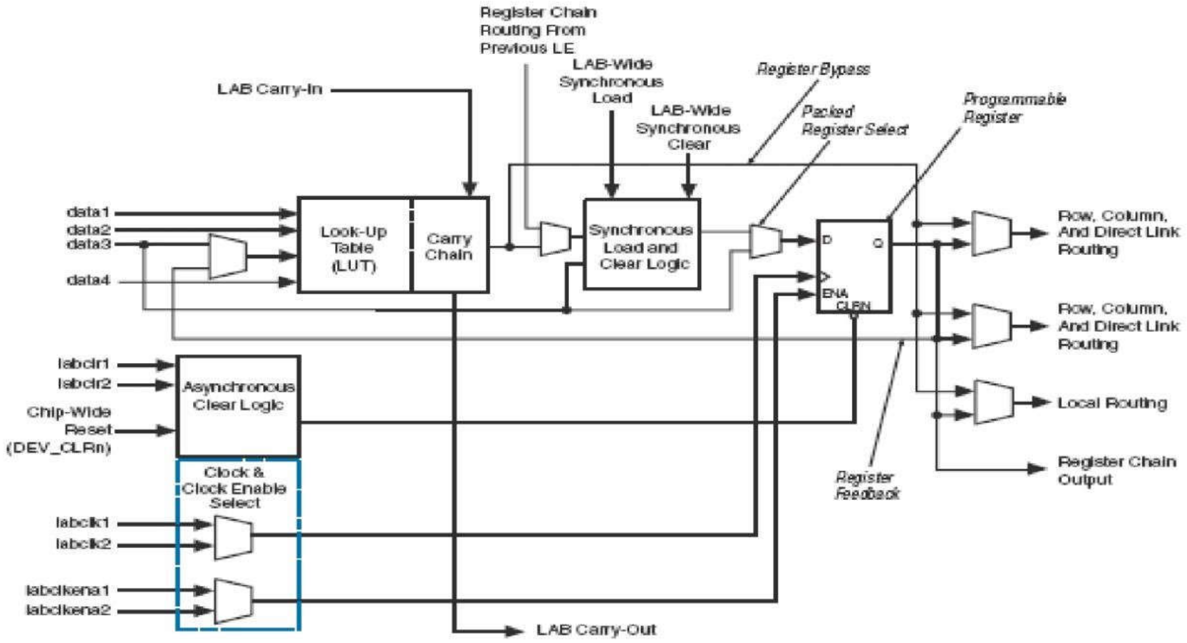


*Figure 2: Block diagram of Logic Block.*

In this exercise you will continue working on the circuit from Lab 2. Later you will design many circuits using Quartus and test them in Altera board using the same procedure.

## Altera DE2 Board

A snapshot of the DE2 board is shown in Figure 3. It depicts the layout of the board and indicates the location of the connectors and key components. The DE2 board has many features that allow the user to implement a wide range of circuits, from simple ones to various multimedia projects. The following hardware is provided on the DE2 board:

1. Altera Cyclone® II 2C35 FPGA device
2. Altera Serial Configuration device - EPCS16
3. USB Blaster (on board) for programming and user API control; both JTAG and Active Serial(AS) programming modes are supported
4. 512-Kbyte SRAM
5. 8-Mbyte SDRAM
6. 4-Mbyte Flash memory (1 Mbyte on some boards)
7. SD Card socket
8. 4 pushbutton switches
9. 18 toggle switches
10. 18 red user LEDs

11. 9 green user LEDs
12. 50-MHz oscillator and 27-MHz oscillator for clock sources
13. 24-bit CD-quality audio CODEC with line-in, line-out, and microphone-in jacks
14. VGA DAC (10-bit high-speed triple DACs) with VGA-out connector
15. TV Decoder (NTSC/PAL) and TV-in connector
16. 10/100 Ethernet Controller with a connector
17. USB Host/Slave Controller with USB type A and type B connectors
18. RS-232 transceiver and 9-pin connector
19. PS/2 mouse/keyboard connector
20. IrDA transceiver
21. Two 40-pin Expansion Headers with diode protection
22. 8 7-segments displays

In addition to these hardware features, the DE2 board has software support for standard I/O interfaces and a control panel facility for accessing various components. In order to use the DE2 board, the user has to be familiar with the Quartus II software. The necessary knowledge can be acquired by reading the tutorials *Getting Started with Altera's DE2 Board* and *Quartus II* tutorials which can be found on Altera's DE2 web pages.

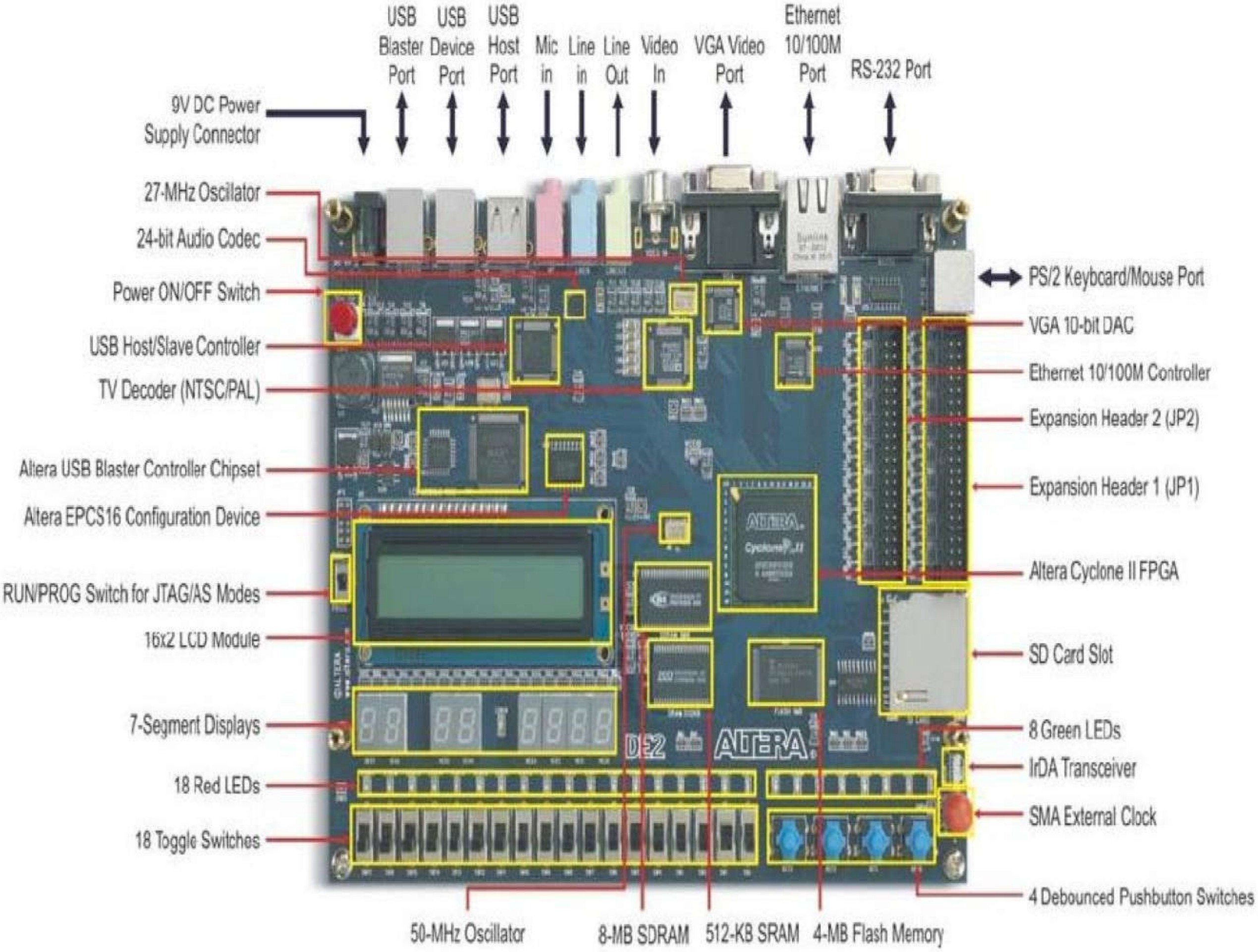


*Figure 3: Altera FPGA DE2 Board.*

Figure 4 gives the block diagram of the DE2 board. To provide maximum flexibility for the user, all connections are made through the Cyclone II FPGA device. Thus, the user can configure the FPGA to implement any system design.

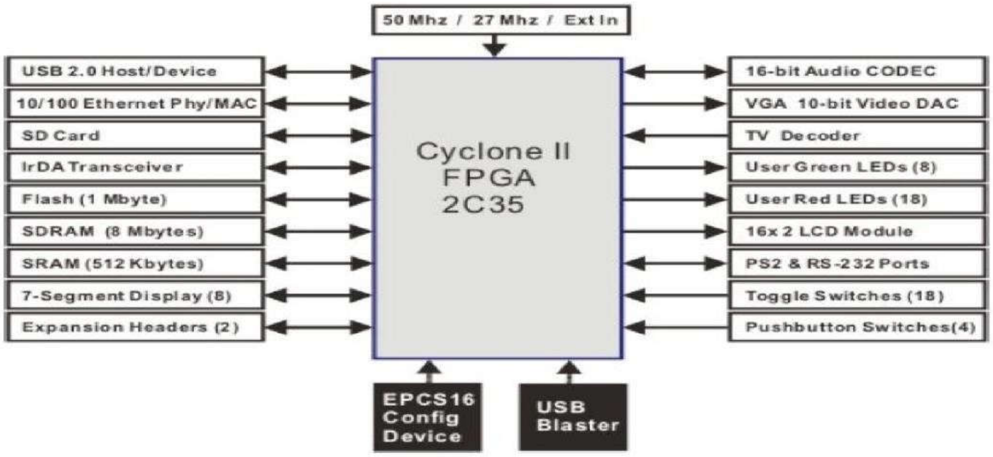


*Figure 4: Block diagram of DE2 board.*

**Configuring the Cyclone II FPGA**

The DE2 board contains a serial EEPROM chip that stores configuration data for the Cyclone II FPGA. This configuration data is automatically loaded from the EEPROM chip into the FPGA each time power is applied to the board. Using the Quartus II software, it is possible to reprogram the FPGA at any time, and it is also possible to change the nonvolatile data that is stored in the serial EEPROM chip. The DE2 board provides two programming modes JTAG and AS. Although you are only supposed to use the JTAG mode, both the types of programming methods are described below:

**JTAG programming**: In this method of programming, named after the IEEE standards Joint Test Action Group, the configuration bit stream is downloaded directly into the Cyclone II FPGA. The FPGA will retain this configuration as long as power is applied to the board; the configuration is lost when the power is turned off.

**AS programming**: In this method, called Active Serial programming, the configuration bit stream is downloaded into the Altera EPCS16 serial EEPROM chip. It provides nonvolatile storage of the bit stream, so that the information is retained even when the power supply to the DE2 board is turned off. When the board's power is turned on, the configuration data in the EPCS16 device is automatically loaded into the Cyclone II FPGA.

Figure 5 illustrates the JTAG configuration setup. To download a configuration bit stream into the Cyclone II FPGA, perform the following steps:

1. Ensure that power is applied to the DE2 board
2. Connect the supplied USB cable to the USB Blaster port on the DE2 board (see Figure 3)
3. Configure the JTAG programming circuit by setting the RUN/PROG switch (on the left

side of the board) to the RUN position.

4. The FPGA can now be programmed by using the Quartus II Programmer module.

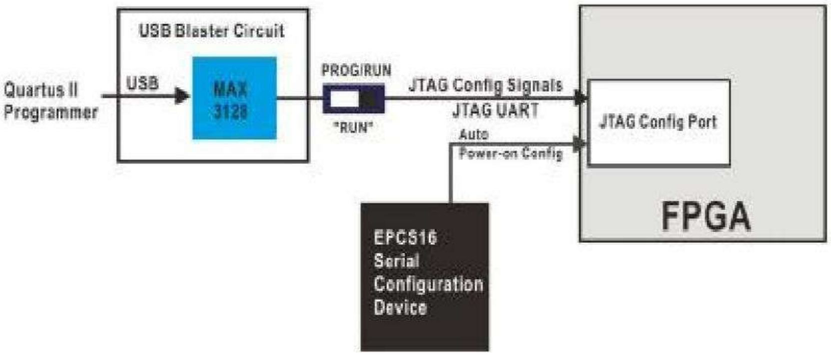


*Figure 5: The JTAG configuration scheme.*

In summary, the design flow for Altera Cyclone II FPGA on DE2 board is as follows:

1. Schematic Design Entry (da)
2. Design Verification (ModelSim)
3. Netlist Conversion (EDIF to EDDM)
4. Download Bit File (Using Quartus II)
5. Test Hardware

Altera® Cyclone™ II devices are designed on an all-layer-copper, low-k, 1.2-V SRAM process and are optimized for the smallest possible die size. Built on TSMC‟s highly successful 90-nm process technology using 300-mm wafers, Cyclone II devices offer higher densities, more features, exceptional performance, and the benefits of programmable logic at ASIC prices. Cyclone II devices offer from 4,608 to 68,416 logic blocks and are designed with an optimal set of features, including embedded 18x18 multipliers, dedicated external memory interface circuitry, 4-kbit embedded memory blocks, phase-locked loops (PLLs), and high-speed differential I/O capabilities.

More information can be obtained for Altera Cyclone II family of FPGAs from https://www.altera.com/products/fpga/cyclone-series/cyclone-ii/overview.html. FPGA part numbering scheme information can be obtained from https://www.altera.com/products/general/devices/dev-format.html.

## PRELIMINARY

Find a copy of the Altera databook and read at least the overview section. In the online documents this is file:

https://www.altera.com/content/dam/alterawww/global/en_US/pdfs/literature/hb/cyc2/cyc2_cii5v1.pdf

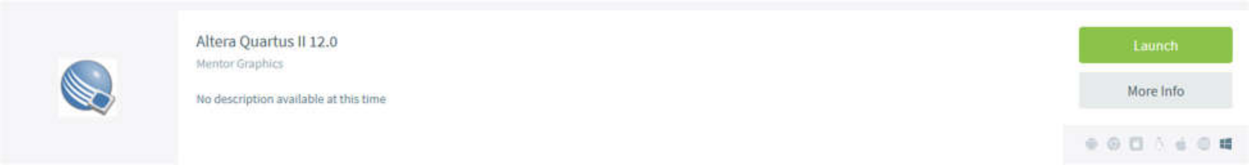
This page can also be found at:

https://www.altera.com/products/fpga/cyclone-series/cyclone-ii/support.html

## PROCEDURE

### Using Quartus II Software for configuring Altera FPGA

1. You can find Quartus II in AppsAnywhere as shown in Figure 6.



Altera Quartus II 12.0

Mentor Graphics

No description available at this time

*Figure 6: Quartus II Launch in AppsAnywhere.*

2. Click on File | Open Project (or Open Existing Project from startup screen). Go to and select the Lab 2 project created in Lab 2.
3. Add 2 outputs nodes and connect them to the 2 input nodes (Figure 7), name these 'in1' and 'in2'.

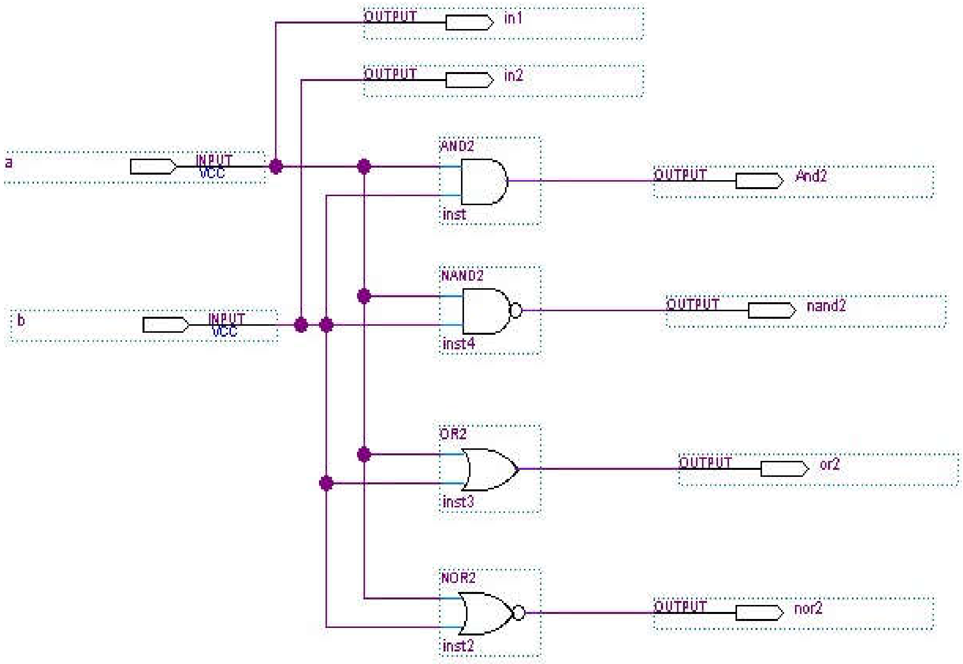
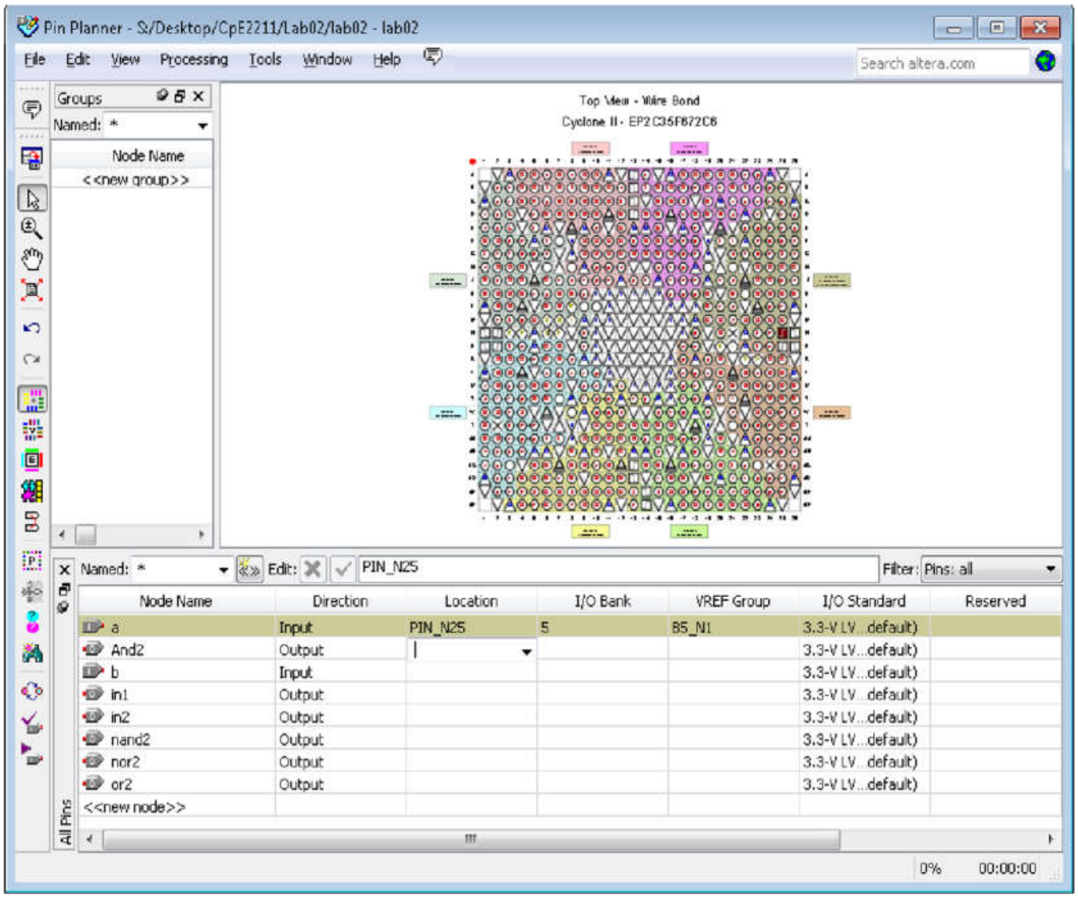


*Figure 7: Lab 2 circuit with two Output nodes added.*

4. Click on **Processing | Start | Start Analysis & Synthesis** to get the newly added nodes recognized by the pin planner.
5. Click on **Assignments | Pin Planner** to get the following menu (Figure 8). Select the input/output variable and a corresponding pin address by referring to Tables A-1 and A-3 in Appendix A. The Switches on the board are typically used as inputs and the LEDs as outputs.



Pin Planner - S:/Desktop/CpE2211/Lab02/lab02 - lab02
File Edit View Processing Tools Window Help
Search altera.com
Groups
Named: *
Node Name
<<new group>>
Top View - Wire Bond
Cyclone II - EP2C35F672C6
All Pins
0%
00:00:00

Named: * Edit: PIN_N25 Filter: Pins: all

| Node Name | Direction | Location | I/O Bank | VREF Group | I/O Standard | Reserved |
|---|---|---|---|---|---|---|
| a | Input | PIN_N25 | 5 | B5_N1 | 3.3-V LV...default) | |
| And2 | Output | | | | 3.3-V LV...default) | |
| b | Input | | | | 3.3-V LV...default) | |
| in1 | Output | | | | 3.3-V LV...default) | |
| in2 | Output | | | | 3.3-V LV...default) | |
| nand2 | Output | | | | 3.3-V LV...default) | |
| nor2 | Output | | | | 3.3-V LV...default) | |
| or2 | Output | | | | 3.3-V LV...default) | |
| <<new node>> | | | | | | |

*Figure 8: Pin Planner.*

6. For this example lab2.bdf file, the pin assignments are as shown in the following table.

| ***Design I/O*** | ***FPGA Pins*** |
|---|---|
| *A* | *Pin_N25* |
| *B* | *Pin_N2* |
| *in1* | *Pin_D25* |
| *in2* | *Pin_J22* |
| *And2* | *Pin_E26* |
| *Nand2* | *Pin_E25* |
| *Or2* | *Pin_F24* |
| *Nor2* | *Pin_F23* |

Some points to note:

- A and B are inputs and in1 and in2 are the same inputs propagated to the output through a buffer. This makes it easier to view the inputs on the oscilloscope.
- Pin_N25 corresponds to an on-board switch. Find out the exact switch number using Table A-1 in Appendix A and locate this switch on the board as this will serve as input A for your design.
- Pin_N2 is a 50 MHz on-board clock pin as shown in Table A-6 in Appendix A, which will serve as input B.
- The remaining pins are all I/O pins which map to the expansion header pins on the board. See Table A-5 in Appendix A for more details.

7. After the Pin Assignment is complete, compile the project again by following Step 6. If a dialog box appears asking the user to save the pin assignments or not, click 'Yes'.
8. The above compilation actually assigns the input and output variables to the pins of the FPGA, to be written to it in the next step. Now Click on **Tools | Programmer** to configure the FPGA for the intended logic operation. Make sure that the FPGA board is switched on, and the RUN/PROG switch is in "RUN" position. Also make sure that the Programmer is set to "JTAG" mode. Now check the "**Program/Configure**" box, and hit the "**Start**" button.
9. The above process writes a code to the FPGA so that it now works as the intended combinational circuit. Use the digital inputs of the MSO to display the inputs and outputs of the design on the FPGA.
10. Connect the digital probes of the MSO to the circuit as follows:
    - Black probe to GND (ground)
    - Channel 0 to 'in1'
    - Channel 1 to 'in2'
    - Channel 2 to AND2
    - Channel 3 to NAND2
    - Channel 4 to OR2
    - Channel 5 to NOR2

    All these pins (A0 to A5) are available on the Expansion Header JP1 as seen in Table A-5 in Appendix A.
11. Press Autoscale on the scope. Notice that channels D0 through D7 are detected as active and are displayed. Some of the channels should be 'jittering' somewhat. This is normal since the scope will be triggering on a rising edge on channel D1 which happens more than once during a period. Trigger on some event which is unique during a period in order to give a stable display of the entire period.
12. Use the leftmost softkey to select the medium level for the amplitude of the signals. Press D0-D15. From the Softkey menu, turn channels D0-D7 on, and channels D8-D15 on.
13. Find the SELECT softkey. Press it up and down, and notice how the highlighting moves among the channels. The highlighted channel designates the active channel which will be affected by the actions you take.
14. Using the SELECT softkey, select channel 0.
15. Find the POSITION softkey. Press it up and down and notice how channel 0 is repositioned

on the  display. Position channel 0 at the top of the display.

16. Find the HORIZONTAL Time/Div knob. Turn it clockwise several clicks. Then turn it CCW several clicks. Observe the status line on top of the screen. Notice the time per division indicator changes as you turn the knob.
17. Press D0-D15. Then press the D8-D15 (soft key) off.  Note how channels 8 through 15 are turned off, and the remaining channels are resized to fit the display.
18. Use the leftmost softkey to select the highest level for the amplitude of the signals. Turn the Entry knob to select D3. Use the second softkey from the left to turn it on/off.
19. Press Autoscale. Note how the channels with activity are turned on, and a seconds/division setting is chosen so that several cycles are displayed on the screen. All inactive channels are turned off. The MSO automatically sets up the proper threshold voltage for the signals found. The digital signals are displayed as a high level whenever the voltage at the probe is above a certain threshold and as a low level when it is below. This threshold can be adjusted if required.
20. Make sure that channels 0-5 are displayed on the screen. If not all six channels are displayed, turn on the inactive channels manually. Press Label.
21. Press the Labels off/on softkey several times and note how the labels on the left side of the waveforms appear and the waveform area is reduced accordingly. The channel numbers remain on screen when the labels are removed.
22. The area called New Label allows you to enter a predefined label or define a label with letters and characters. The Library softkey shows the predefined list of labels. The Spell softkey provides an alphanumeric list, from which you can select (using the Entry knob) individual letters, numbers, and characters to define new labels. The Spell, Enter, and Delete Character softkeys are used to type the new label. Then pressing the Apply New Label adds the new label to the selected channel.
23. Use the Select softkey until D0 is highlighted.
24. Press the Spell softkey and use the Entry knob to select the letter “I”. Press the Enter softkey.
25. Do step 24 for “N”, and “1” to create the new label “IN1”.
26. Press the Apply New Label softkey. Note how the label “IN1” has been assigned to D0.
27. Press the leftmost softkey and use the Entry knob to select D1.
28. Press the Apply New Label and note how D1 has been labeled “IN2”.
29. Similar to Step 28, select D2. Use the Spell, Enter and Delete Character softkeys to name it as "AND2". Then press Apply New Label.

30. Repeat the process for D3, D4 and D5 to label them as "NAND2", "OR2" and "NOR2", respectively.

31. Make sure the assigned switch to input “A” on the board is ON (“IN1” on the oscilloscope will be HIGH).  Press the Run/Stop or Single and observe the delay between the “IN2” bit (D1) and the next event (rising or falling edge) on the output D2 (labeled as “OUT1”).

**Q1. Can you see the delay?**

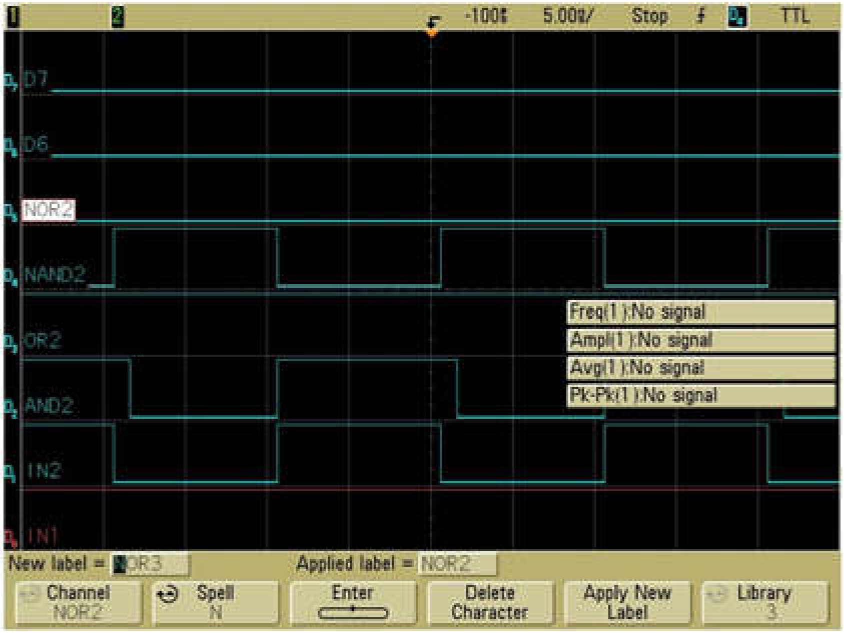
**Q2. Why or why not?**

**Q3. What does the delay signify?**

32. While in “RUN” mode, view the value of “IN1” on the scope.  Change it using the switch connected to input “A” and observe how the outputs change.

**Q4.** **By changing the input switch to a "1" or a "0" on the board, record the corresponding outputs for all the possible input combinations of "IN1" and "IN2" in a truth table.**

**Q5.** **Take a snapshot of the scope output similar to Figure 9 (make sure you have proper names of the signals); take one snapshot when the input is 0 and another snapshot when the input is 1. You can do that by pressing the save softkey, you will need to insert a flash drive into the MOS to save it.**



Freq(1 ):No signal
Ampl(1 ):No signal
Avg(1 ):No signal
Pk-Pk(1 ):No signal

New label = NOR3
Applied label = NOR2

*Figure 9: Oscilloscope capture.*

**Q6.** **What is the period of IN2?**

**Q7.** **How about the period of IN1?**

**Q8.** **Outputs AND2, NAND2, OR2 and NOR2 are logic functions of IN1 and IN2. Write down the logic functions implemented for the outputs in terms of IN1 and IN2.**