**LAB 2**

**Introduction to VHDL and Operation of Full Adder/Subtractor**

**Using Xilinx ISE Project Navigator and**

**Nexys2 FPGA Development Board**

**Acknowledgements:** Developed by Bassam Matar, Engineering Faculty at Chandler-Gilbert Community College, Chandler, Arizona. *Funded by NSF*

**NEED:** A widening gap exists between what being taught in schools and skills currently being utilized by the workforce. The jobs of the future are not labor intensive, they are brain intensive. Technicians and engineers no longer wire circuits or assemble a chip from scratch; they use software to run modern equipment. Traditionally, hardware digital circuits require at least five or six (Transistor Transistor Logic, TTL) ICs. The wiring is often very complex and messy. Digital test equipment is usually needed to make the system functional. TTL ICs technology is over thirty years old. An FPGA design minimizes the amount of electrical wiring and eliminated the use of complicated test equipment. This type of technology allows students to concentrate on digital principles and not on the electrical wiring. Larger and more complex projects can be undertaken now that the tedious manual procedures are automated. We will cover FPGAs in detail later in the course. However, in general, an FPGA is a chip that allows you to reconfigure it with any design. Thus, if you make a mistake in your design, you don’t need to replace wires or chips, as you might have done using TTL. Instead, you simply reconfigure the FPGA with each updated design.

1. **Lab Summary:** In previous labs, you became familiar with TTL technology and bread boarding. In this lab, you will get familiar with modern way of digital technology using one of the leading digital design software “Xilinx” and interfacing with Spartan **Nexys2** development board. Combinational Logic Circuits are used to make decisions based on a series of true statements that can be laid out in a truth table. In previous years the 74xx, family (TTL) circuits were used to design and build basic combinational logic circuits. In this lab activity, we will 1) use the software application Xilinx Project Navigator ISE to draw a schematic of full adder and then program it to the **Nexys2** FPGA device, 2) You will be able to implement the same design using hardware language programming (VHDL) and 3) you will apply your understanding of VHDL to implement a 4-bit Adder/Subtractor with Nexys2 FPGA development board.

**Lab Goal:** The goal of this lab is to lean the use of **Nexys2** FPGA development board from Xilinx; how to program in VHDL and create the hardware connections between the development board and your PC.

**Learning Objectives**

1. Getting familiar with Nexys2 board. The advantage of this board is that the USB cable will allow you to programmed and powered the board.
2. Design the full adder schematic and VHDL using Xilinx® ISE, compile and simulate for Xilinx Nexys2 FPGA
3. Develop a User Constraint File **“ucf”** that maps the input and output signals to the Nexys2FPGA
4. Learn how to program the full Adder into the FPGA Nexys2board
5. Test the Results
6. Repeat the same process for Full Adder with VHDL instead of Schematic
7. Test your understanding by applying what you learned in programming a 4-bit Adder/Subtractor

**Grading Criteria:** Your grade will be determined by your instructor.

**Time Required**: 4 - 5 hours

**Special Safety Requirements**

Static electricity can damage the FPGA devices used in this lab. Use appropriate ESD methods to protect the devices. Be sure to wear a grounded wrist-strap at all times while handling the electronic components in this circuit.

No serious hazards are involved in this laboratory experiment, but be careful to connect the components with the proper polarity to avoid damage.

**Lab Preparation**

* Review your PLD lecture from your class.
* Review “Introduction to Xilinx” previous lab.
* Read Task1 and work the full adder schematic from Task 2 as prelab assignment
* Using Xilinx software, draw the adder/subtractor circuit as shown in task 3 of this lab
* Create the adder/subtractor “UCF” file that maps the input and output signals to the Nexys2 FPGA
* Acquire required hardware components/equipment.
* Print out the laboratory experiment procedure that follows.

**Equipment and Materials**

Each team of students will need the test equipment, tools, and parts specified below. Students should work in teams of two or three.

|  |  |
| --- | --- |
| **Test Equipment and Power Supplies** | **Quantity** |
| The following items from the Xilinx:   * free software **ISE** WebPACK ([www.xilinx.com](http://www.xilinx.com)) that can be installed on your personal computer or full version of Xilinx in your classroom * Nexys2 FPGA kit including download and power cable * Free Digitlent Adept software : <http://www.digilentinc.com/Products/Detail.cfm?NavPath=2,66,828&Prod=ADEPT2> | 1 |

Additional References:

1. Xilinx Nexys2 FPGA Reference Manual and Schematic from:

<http://www.digilentinc.com/Products/Detail.cfm?Prod=NEXYS2>

**Task *1.***

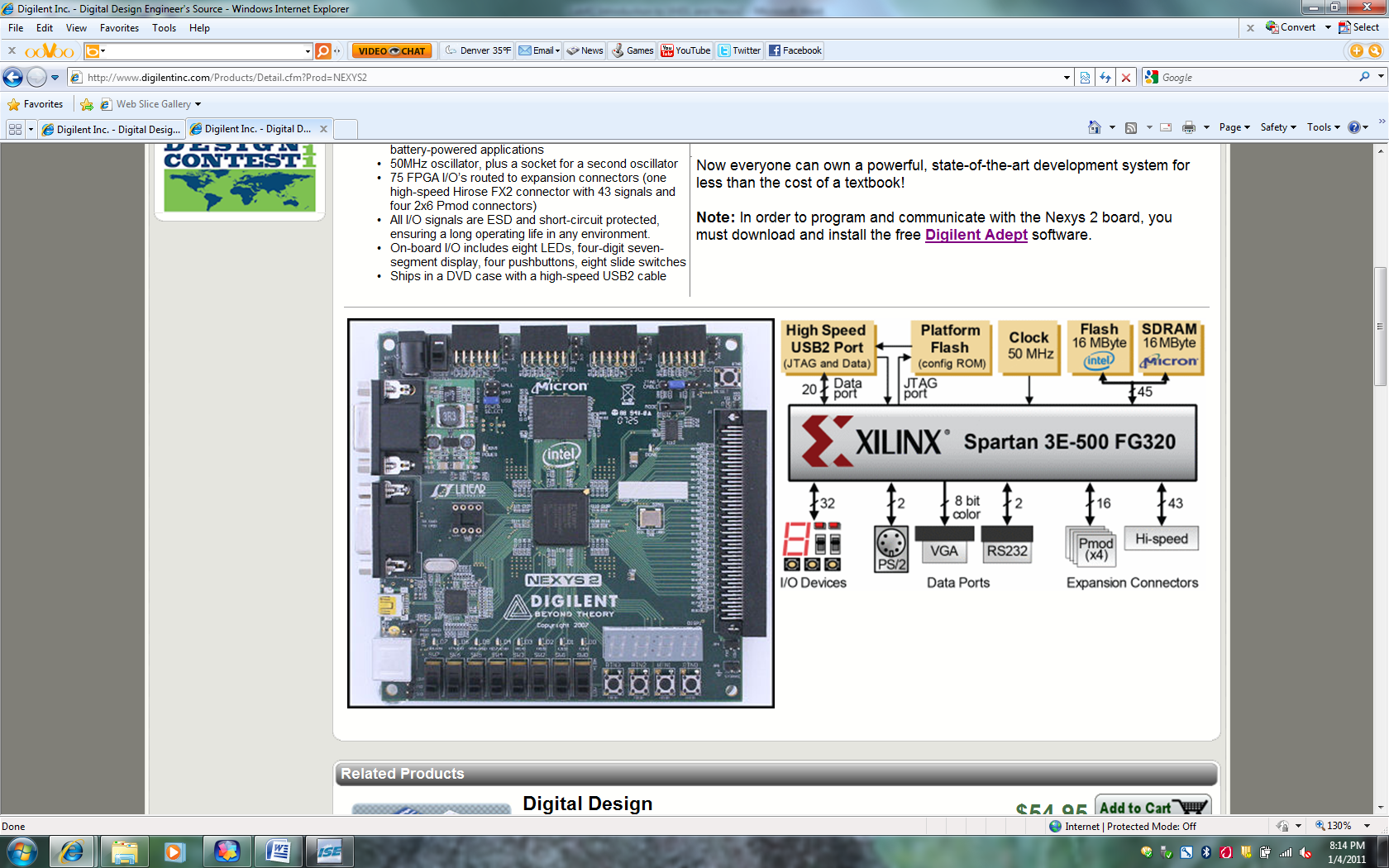
***Getting to know FPGA* Nexys2** ***board.***

The Nexys-2 is a powerful digital system design platform built around Xilinx Spartan 3E FPGA. The advantage of this board is that it is programmed and powered through a USB port. A list of the key features and their location on the board is listed below:

PS/2 mouse/keyboard port

On-Board 50 MHz Oscillator

CLK\_50MHz: (B8)

******

Seven Segment Display

FPGA IC

Done LED

Expansion connectors

Reset Button

USB Connector

Power In

Two RS-232 serial ports

Power Switch

Power On indicator LED

Figure 1: FPGA Nexys2 Board

Push-Button Switches

8 Toggle Switches [sw(7:0)]

8 LEDs [ld(7:0)]

**User I/O**

The Nexys2 board includes several input devices, output devices, and data ports, allowing many

designs to be implemented without the need for any other components.



Figure 2: FPGA Nexys2 inputs and outputs

***Inputs: Slide Switches and Pushbuttons***

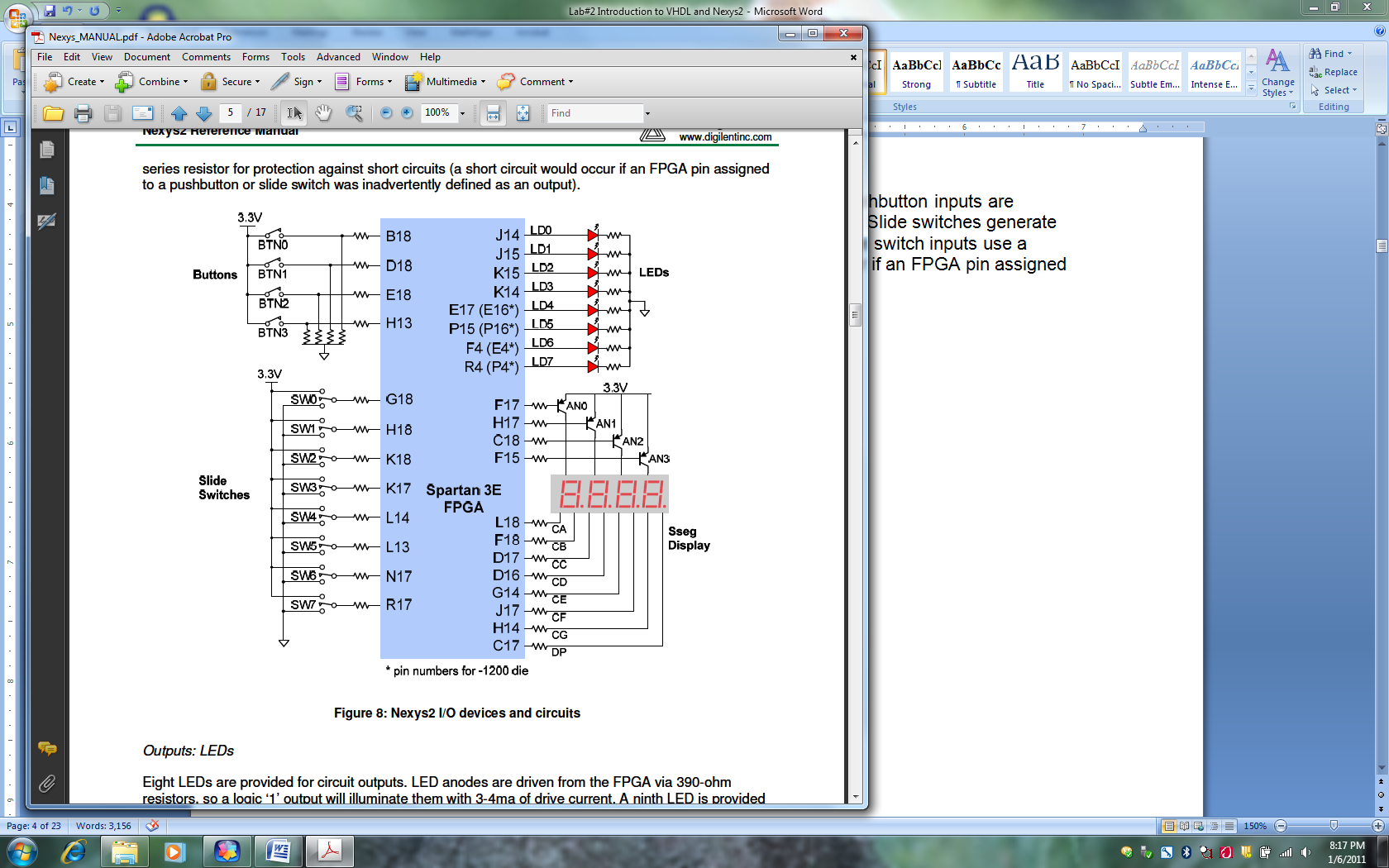
Four pushbuttons and eight slide switches are provided for circuit inputs. Pushbutton inputs are

normally low, and they are driven high only when the pushbutton is pressed. Slide switches generate

constant high or low inputs depending on their position. Pushbutton and slide switch inputs use a

series resistor for protection against short circuits (a short circuit would occur if an FPGA pin assigned

to a pushbutton or slide switch was inadvertently defined as an output).



*Please note that the pin locations that are in parenthesis are designated to Nexys2, 1200 version.*

Figure 3: Nexys2 I/O devices and circuits

Please refer the reference manual for any additional information:

<http://www.digilentinc.com/Data/Products/NEXYS2/Nexys2_rm.pdf>

T**ask 2:**

**Implement Full Adder schematic using Xilinx ISE 12.1 tools for Nexys2 FPGA board:**

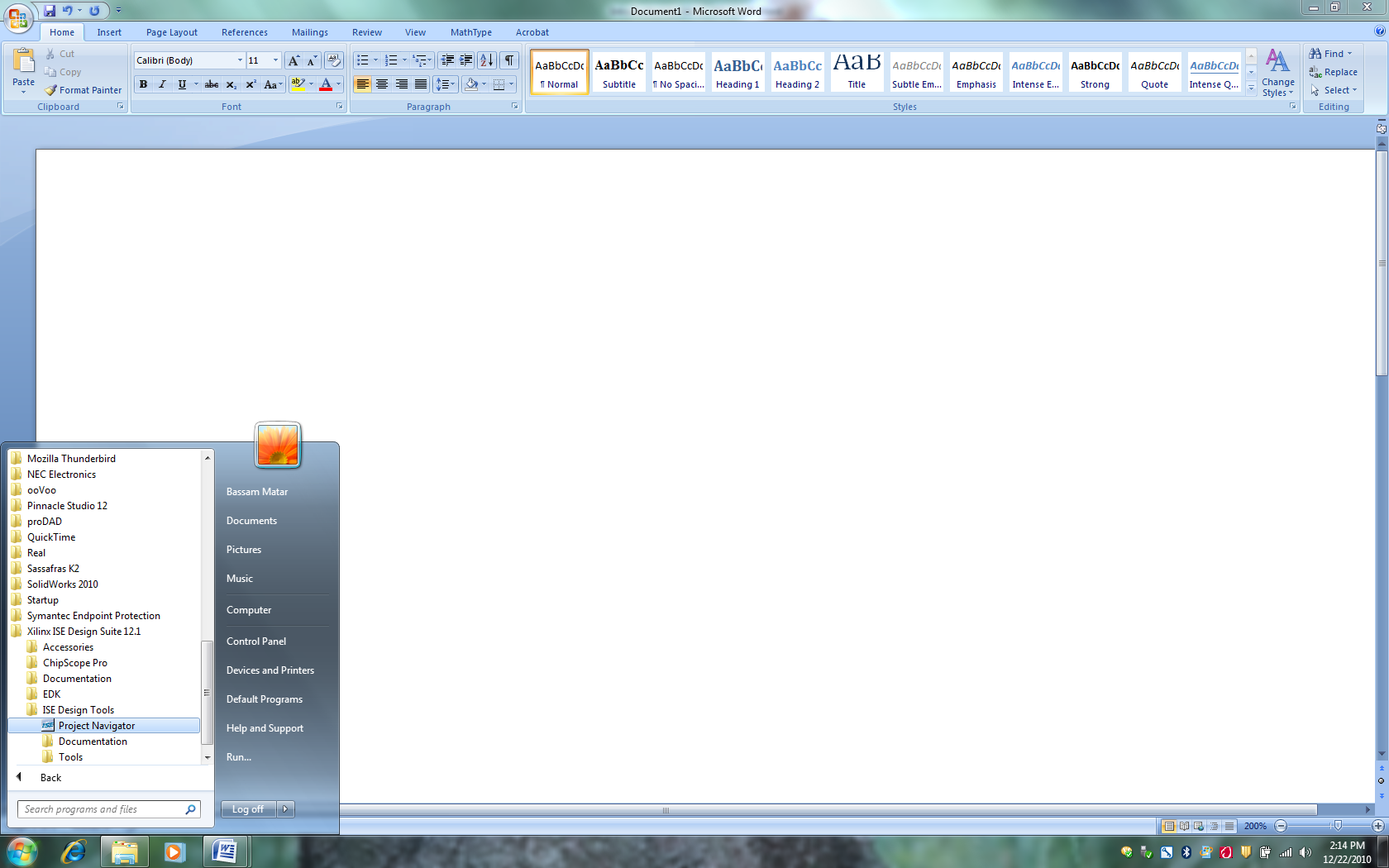
**Part 1:**

Please refer to previous lab on how to use Xilinx ISE 12.1 software.

* 1. We need to set up our project correctly to reflect **Nexys2** FPGA board.

**Design Entry Instructions**

1. Open Xilinx ISE 12.1 edition software



a. Select Start

b. All Programs

c. Xilinx ISE Design Suite 12.1

d. ISE Design Tools

d. Project Navigator

|  |  |
| --- | --- |
| The starting windows should look like this |  |

Figure 4: Xilinx Starting Window

|  |  |
| --- | --- |
| 2. Create a new project by selecting:  **File** from the main menu  **New Project**   1. In the New Project window, **name** your project *Full\_Adder* in the project name text box. 2. In the **Project Location** selection box, enter the folder or directory where your project will be saved. Use your name as for *Student\_Name* and locate the place where you want to save all your files (i.e C:\) 3. Under **Top-Level Source Type**, select **HDL** and click Next as shown in figure 5 | New Project |

Figure 5: New Project Window

|  |  |
| --- | --- |
| 3. We will design our full adder for a particular device “Nexys2 FPGA”.  **Product Category:** All  **Family:** Choose **Spartan 3E**, the device we will be using.  **Device**: **XC3S500E**, the specific Spartan 3E device we use. This is actually printed (very small) on the FPGA package which you will see in the hardware lab.  **Package:** FG320, this is the package type of our device (Ball Grid Array, 320 pins)  **Speed Grade**: The speed grade for this device is -5.  **Preferred Language**: VHDL |  |

Figure 6: Device Properties

Hit the Next button to go the next menu

|  |  |
| --- | --- |
| 4. Just review the information listed in below figure and make sure it matches the information in the window.  Then click **Finish** to complete the process and verify the file name and type. Click **Next** once again to proceed and finish. |  |

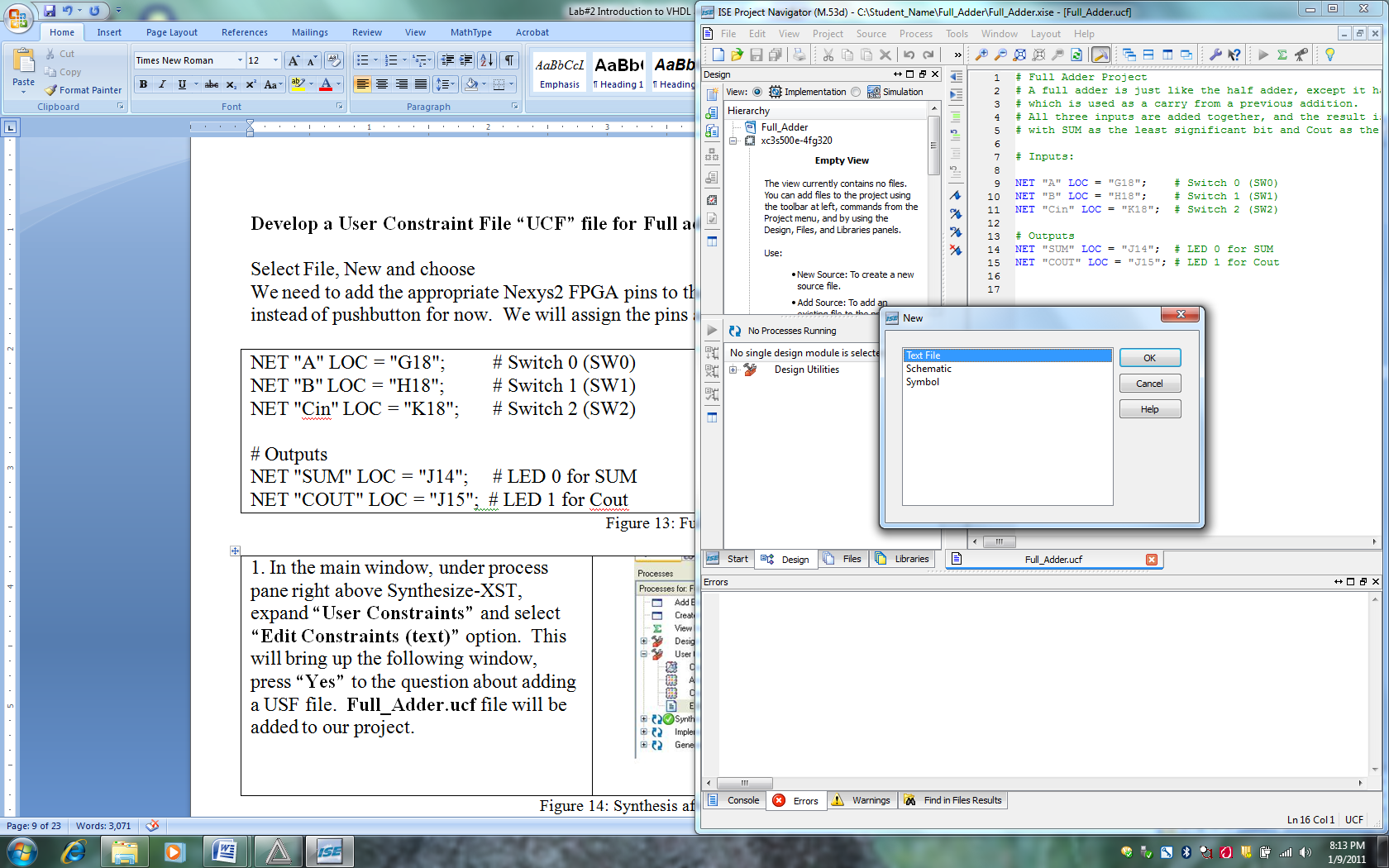
Figure 7: Project Summary

|  |  |
| --- | --- |
| 5. Create the Full Adder circuit. Please refer to previous lab for more details on how to create the schematic.    Attach input and output pins to the circuit and label them as shown. |  |

Figure 8: Full Adder Circuits

**6. Develop a User Constraint File “UCF” file for Full adder I**/O

Select **File**, **New** and choose ***Text File*** as shown:





We need to add the appropriate Nexys2 FPGA pins to the I/O makers of our design. Let us choose the switches instead of pushbutton for now.

Enter the following pin assignment. Save the file as ***Full\_Adder.ucf***

We will assign the pins as follows:

NET "A" LOC = "G18"; # Switch 0 (SW0)

NET "B" LOC = "H18"; # Switch 1 (SW1)

NET "Cin" LOC = "K18"; # Switch 2 (SW2)

# Outputs

NET "SUM" LOC = "J14"; # LED 0 for SUM

NET "COUT" LOC = "J15"; # LED 1 for Cout

Figure 9: Full Adder UCF

|  |  |
| --- | --- |
| 7. Once both files are created (Schematic and UCF), select **Project🡪Add Source** and select both files to be included as part of your project.  Once you select the files and hit OK, you should get the following screen🡪  If you select both files (the schematic is already part of the project) you will get a warning. This is not a problem. |  |

Figure 10: Adding Full Adder Schematic and UCF to the project

|  |  |
| --- | --- |
| 8.Double click on **“Generate Programming File”** under the implement design on the left hand side. If all goes well, you should get green check marks on **“Programming File Generation Report”** as shown. | generate programming file |

Figure 11: Generate Programming File

|  |  |
| --- | --- |
| 9. The software that we will use for configuring the FPGA is named ***Adept***, from digilentinc.com. To Access Adept software, Select Start 🡪 Programs🡪 Digilent🡪Adept🡪Adept |  |
| 10.Choose Nexys 2 board from the Connect Product option.  Check you USB connection if you are not able to locate the name Nexys 2. |  |
| Click on browse and select **full\_adder.bit** from your project directory and click on open.  You will prompt with a warning screen, ignore it selecting yes.  Now click on **program** button next to Browse.  ***You will prompt with the same warning screen as before, ignore it selecting yes.*** |  |

Figure 12: Programming Nexys 2 FPGA board.

The program gets downloaded to the FPGA board.

At this point the FPGA should be programmed and ready for test. Test it by sliding the programmed switches up and down for ON and OFF. See if the appropriate LEDs for the two outputs turn on. Here is a truth table that should guide you with your test.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SW(0) (A)**  **G18** | **SW(1) (B)**  **H18** | **SW(2) (Cin)**  **K18** | **LD0 (SUM)**  **J14** | **LD1 (Cout)**  **J15** |
| 0 | 0 | 0 | OFF | OFF |
| 0 | 0 | 1 | ON | OFF |
| 0 | 1 | 0 | ON | OFF |
| 0 | 1 | 1 | OFF | ON |
| 1 | 0 | 0 | ON | OFF |
| 1 | 0 | 1 | OFF | ON |
| 1 | 1 | 0 | OFF | ON |
| 1 | 1 | 1 | ON | ON |

Figure 13: Hardware Truth Table Results



*When all three Switches are ON, both outputs are High (SUM and Cout).*

Figure 14: Hardware Display of Full Adder

As you can see with this approach, you do not need to replace wires and ICs if you make mistake as you might have done in Labs 1 and 2 with TTL. Instead, you will have to reconfigure the FPGA with the updated design.

This type of technology allows you to concentrate on digital principles and not on the electrical wiring. Larger and more complex projects can be undertaken now that the tedious manual procedures are automated. Task 3 of this lab demonstrates the design of a large design.

**Task 3:**

**Implement Full Adder with (VHDL) using Xilinx ISE tools for Nexys2 FPGA board:**

VHDL is an acronym inside of an acronym. The ‘V’ stands for Very High Speed Integrated Circuit (VHSIC) and ‘HDL’ stands for Hardware Descriptive Language. VHDL is a powerful language with numerous language constructs that are capable of describing very complex behavior needed for today’s programmable devices.

**Part 1:**

**Close the previous project, and create a new one**

|  |  |  |
| --- | --- | --- |
| 1. Create a new project by selecting:  **File** from the main menu  **New Project**   1. In the New Project window, name your project *Full\_Adder\_VHDL* in the project name text box. 2. In the **Project Location** selection box, enter the folder or directory where your project will be saved. Use your name for *Student\_Name* and locate the place where you want to save all your files (i.e C:\) 3. Under **Top-Level Source Type**, select **HDL** instead of schematic and click **Next**. | Figure 15: New Project with VHDL | |
| 3. We will design our full adder for a particular device “Nexys2 FPGA”.  **Product Category:** All  **Family:** Choose **Spartan 3E**, the device we will be using.  **Device**: **XC3S500E**, the specific Spartan 3E device we use. This is actually printed (very small) on the FPGA package which you will see in the hardware lab.  **Package:** FG320, this is the package type of our device (Ball Grid Array, 320 pins)  **Speed Grade**: The speed grade for this device is -5.  **Preferred Language**: VHDL | | Figure 16: New Project Properties |
| 1. Select ***Project🡪New Source*** as shown | | Figure 17: New Source as VHDL | |
| 3. Select V***HDL Module*** and type Full\_Adder\_VHDL for ***File Name***.  Click Next | programming6 | |

Figure 18: VHDL Module setup

|  |  |
| --- | --- |
| 4. Fill in the inputs (in) and Outputs (out) as shown. Click Next and Finish | programming7 |

Figure 19: Full Adder Input / Output set up

Click several next and finish until, you get the following screen:

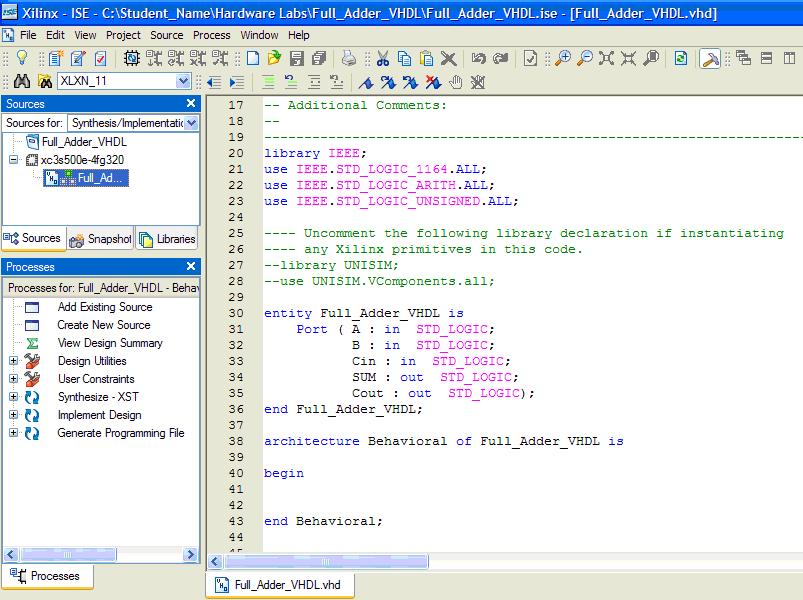


Figure 20: VHDL Layout of input and output

|  |  |
| --- | --- |
| . Between ***begin*** and ***end*** ***Behavior (line 40-43)***, we need to enter the following expression for Full Adder.  Based on our previous schematic design, the logic expression for the output Sum and Cout are: | vhdl2 |

Figure 21: Complete Full Adder VHDL

Follow the same steps as before from part 1 to include UCF file as part of your project. Synthesize, and generate the programming to file.

***Demo your working hardware to your teammates.***

**TASK 4: Test your Knowledge**

*Implement the following 4-bit adder/subtractor schematic using Xilinx ISE tools for Nexys2 FPGA board. Also, implement the circuit with VHDL.*

Create below schematic by choosing ***ADD4*** symbol from Xilinx library of devices

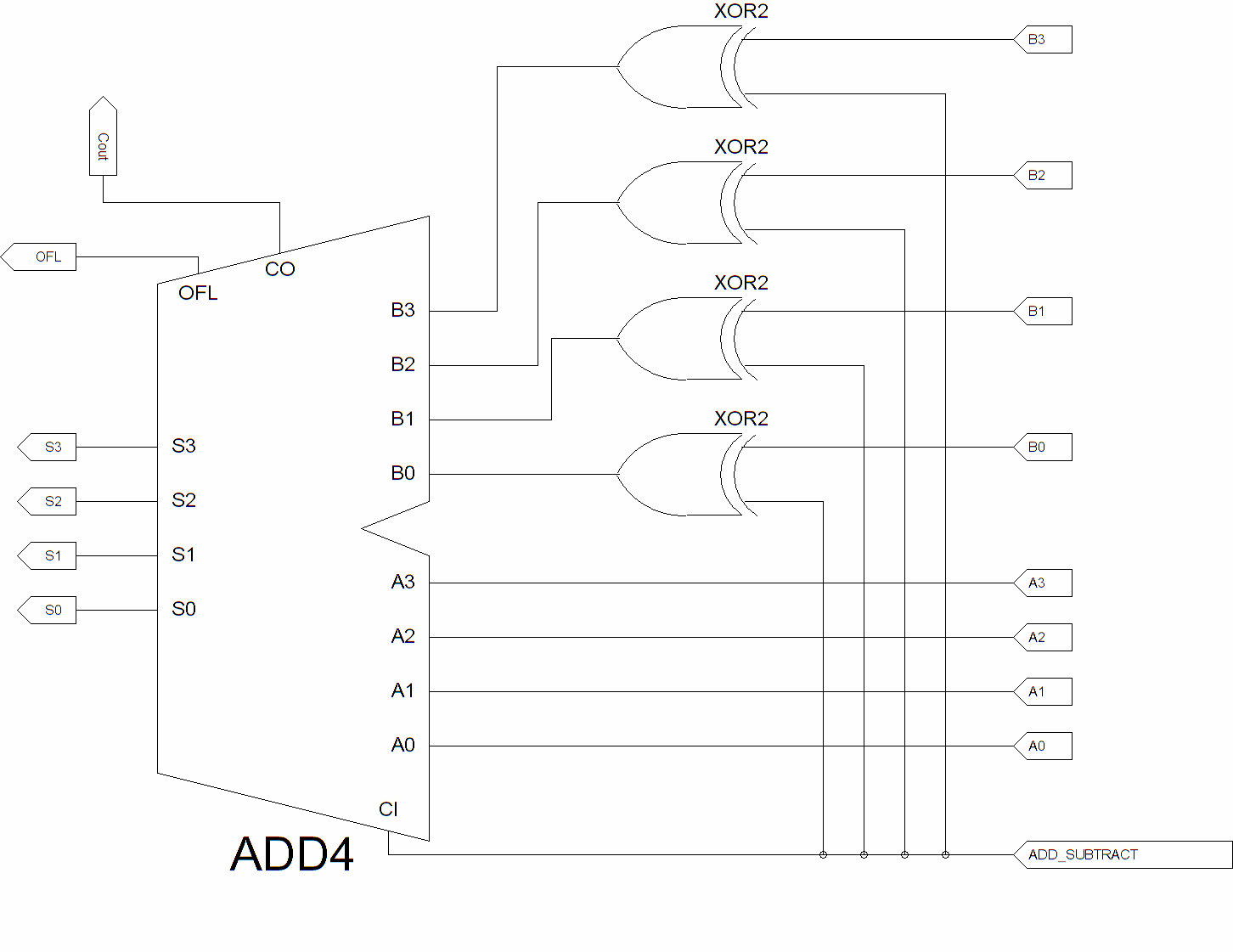


Figure 22: Adder/Subtractor Schematic

Add inputs and output ports

Simulate your design.

Based on your knowledge of tasks I and II, develop a User Constraint File **“ucf”** that maps the input (A3..0, B3..0 and ADD/SUB) and output signals (S3..0, Cout and OFL) to the Nexys2 FPGA.

***Please note you can rotate a symbol by selecting it and hit CTRL “R” for rotate.***

***OR***

***Select the symbol, click on the right key mouse🡪select symbol🡪rotate.***

Program your design in Nexys2 FPGA

Demo your design to your instructor and complete the following truth table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ADD/SUBTRACT** | B3 B2 B1 B0 | **A3 A2 A1 A0** | **S3 S2 S1 S0** | **OFL (yes/no)** | **Cout** |
| 1 | 0 1 0 1 | 1 0 0 1 |  |  |  |
| 0 | 0 1 0 1 | 1 0 0 1 |  |  |  |

Here is one version of the VHDL and UCF files:

**VHDL:**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

USE ieee.std\_logic\_arith.ALL; --You will need to add these two libraries

USE ieee.std\_logic\_unsigned.ALL; --You will need to add these two libraries

ENTITY addsub IS

PORT

(

A: IN std\_logic\_vector(3 downto 0);

B: IN std\_logic\_vector(3 downto 0);

ADD\_SUBTRACT: IN STD\_LOGIC;

S: OUT std\_logic\_vector(3 downto 0);

COUT: OUT std\_logic

);

END addsub;

ARCHITECTURE behavioral OF addsub IS

signal result: std\_logic\_vector(4 downto 0);

BEGIN

PROCESS (A, B, ADD\_SUBTRACT)

BEGIN

IF (ADD\_SUBTRACT = '1') THEN

result <= (“0” & A) - B;

ELSE

result <= (“0” & A) + B;

END IF;

END PROCESS;

S<= result(3 downto 0);

COUT <= result(4);

END behavioral;

**UCF:**

NET "B(0)" LOC = "G18"; # Switch 0 (SW0)

NET "B(1)" LOC = "H18"; # Switch 1 (SW1)

NET "B(2)" LOC = "K18"; # Switch 2 (SW2)

NET "B(3)" LOC = "K17"; # Switch 3 (SW3)

NET "A(0)" LOC = "L14"; # Switch 4 (SW4)

NET "B(1)" LOC = "L13"; # Switch 5 (SW5)

NET "B(2)" LOC = "N17"; # Switch 6 (SW6)

NET "B(3)" LOC = "R17"; # Switch 7 (SW7)

NET "ADD\_SUBTRACT" LOC = "H13"; # Push Button 3

**# Outputs**

NET "S(0)" LOC = "J14"; # LED 0 for Sum

NET "S(1)" LOC = "J15"; # LED 1 for Sum

NET "S(2)" LOC = "K15"; # LED 2 for Sum

NET "S(3)" LOC = "K14"; # LED 3 for Sum

NET "COUT" LOC = "P16"; # LED 5 for Cout PINS FOR 1200 only

NET “OFL” LOC = “P4”; #not defined in VHDL so not used for VHDL portion of the project

**LAB REPORT GRADE SHEET**

**HARDWARE LAB 2**

**Operation of Full Adder and Subtractor**

**Using Xilinx ISE Project Navigator and**

**Nexys2 FPGA Development Board with VHDL**

Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Instructor Assessment

|  |  |  |
| --- | --- | --- |
| Grading Criteria | Max.  Points | Points |
| **Report Writing** | **15** |  |
| Complete Title Page | 1 |  |
| Organization, Neatness, Clarity and Concision | 10 |  |
| Statement of Learning Objectives and Outcomes | 4 |  |
| **Description of Assigned Tasks, Work Performed & Outcome Met** | **75** |  |
| **Task 1:** Build, debug and demonstrate the operation of Full adder schematic in Xilinx and Nexys2 FPGA board. Demo your design | 20 |  |
| **Task 2:** Build, debug and demonstrate the operation of Full adder **VHDL** in Xilinx and Nexys2 FPGA board. Demo your design | 20 |  |
| **Task 3:** Build, debug and demonstrate the operation of a 4 bit adder/subtractor schematic in Xilinx and Nexys2 FPGA board. Simulate your circuit and exhaustively test all possible input combinations. Demo your design. Include a copy of UCF file and timing diagram in your lab report. | 35 |  |
| **Conclusion** Include technical comments and observations which answer the objectives of the lab experiment. | **10** |  |
| **Lab Score** | **100** |  |

**Report Writing Comments:**

1. Title Page: Include **Your name, Course Number, Laboratory Experiment Number and Title**, **Date Performed, and Date Submitted** in all reports.
2. Include **Truth Tables** or **Function Tables** as required to explain how circuits work and as proof of circuit tests.
3. Label all figures or circuits. (e.g. Figure 1. Schematic of….., Figure 2……. , etc.)
4. Refer to all circuits or figures that you include in the text of your report.