CMPS 375
Computer Architecture
Lab1: Adder1 (1-bit Adder)
Introduction to the Nexys4 DDR Package (VIVADO 2016.4)

Objectives

The purpose of this lab is to learn the basics of the VIVADO software and the Xilinx Nexys4 DDR FPGA board as shown in Figure 1. After completing this exercise, you will be able to:

- Create a Vivado project sourcing VHDL models and targeting a specific FPGA device located on the Nexys4 DDR board
- Understand the design circuit writing in VHDL hardware description language and its schematic
- Use the provided user constraint file (XDC) to constrain pin locations
- Simulate the design using the XSIM simulator
- Synthesize and implement the design
- Generate the bitstream
- Program and download the design to the FPGA device on the Nexys4 DDR board
- Test and verify the functionality of the designed circuit on the Nexys4 DDR board

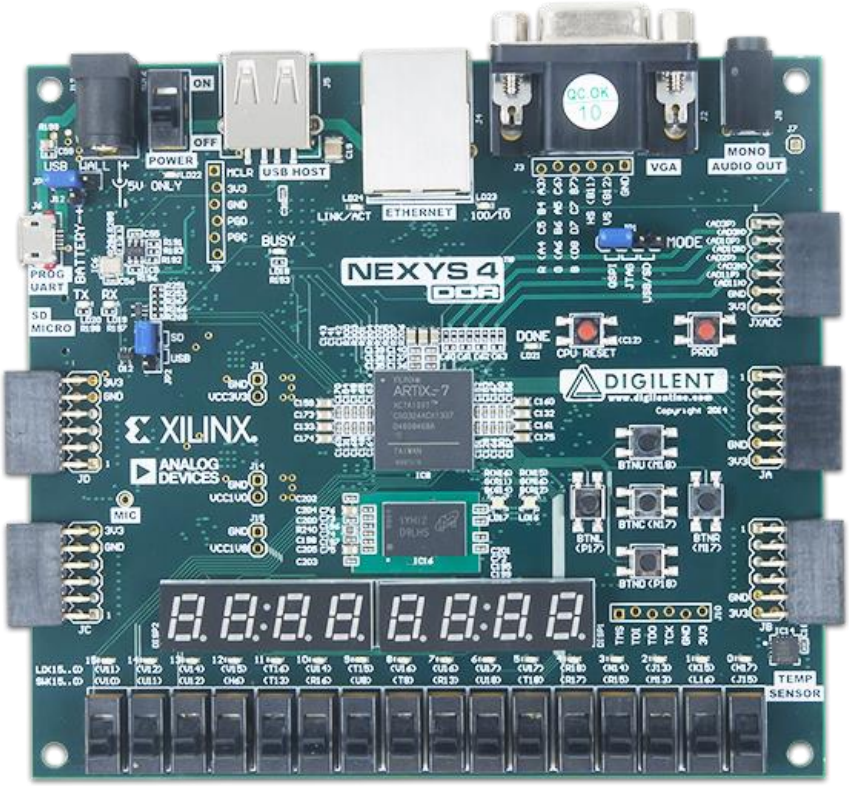

Figure 1. Xilinx Nexys4 DDR Board: Artix-7 FPGA (XC7A100TCSG324-1)

Problem Description

You are required to create a 1-bit adder in in VHDL language. Like Figure 3.12 in textbook, this 1-bit adder can perform a binary addition using switches as inputs and LEDs as outputs on the Nexys4 DDR board.

Inputs			Outputs	
X	Y	Carry In	Sum	Carry Out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

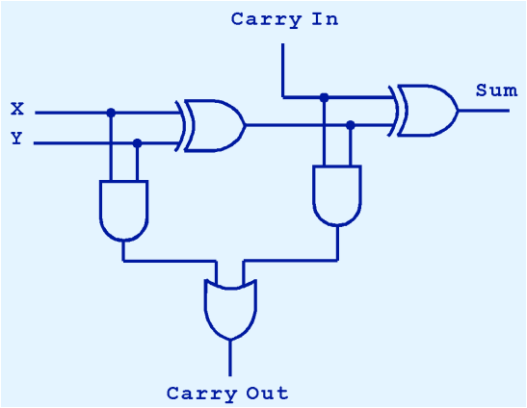

FIGURE 3.12 The Truth Table and The Logic Diagram for a Full-Adder

Note that this full-adder is composed of two half-adder;

CarryOut = XY + (X xor Y) CarryIn; Sum = X xor Y xor CarryIn;

Lab Attachment

- CMPS375Lab1.zip consists of 4 files:
  1. Current document (CMPS375Lab1.pdf)
  2. VHDL design file (Adder1.vhd)
  3. VHDL Test Bench simulation file (Adder1_tb.vhd)
  4. I/O Constraints (Nexys4DDR_Master.xdc)

Lab Submission

1. Check the following before your submission
   a. **Adder1.vhd** under Adder1/Adder1.srcs/sources_1/imports/CMPS375Lab1 folder
   b. **Adder1_tb.vhd** under Adder1/Adder1.srcs/sim_1/imports/CMPS375Lab1 folder
   c. **Nexys4DDR_Master.xdc** under Adder1/Adder1.srcs/sim_1/imports/CMPS375Lab1 folder
2. Zip the entire Adder1 folder into a single file name Adder1.zip
3. Submit **Adder1.zip** to Moodle as your lab1.

Note: Perform binary addition using toggle switches as the inputs and LEDs as the outputs on the Nexys4 DDR board.

Scalar Ports	*Direction*	*Package Pin*	*Name*
a	IN	M13	SW[2]
b	IN	L16	SW[1]
cIn	IN	J15	SW[0]
cOut	OUT	K15	LED[1]
sum	OUT	H17	LED[0]

A 1-bit Adder in VHDL Language

1. Create a Vivado project: Adder1
   a. Click on *File* menu and then *New Project* item to start the wizard.
   b. In the *New Project* form as shown in Figure 3, enter **Adder1** in the *Project name* field and select **C:/Cmps375** in the *Project location*.
   c. Make sure the *Create Project Subdirectory* box is checked. Click *Next*.

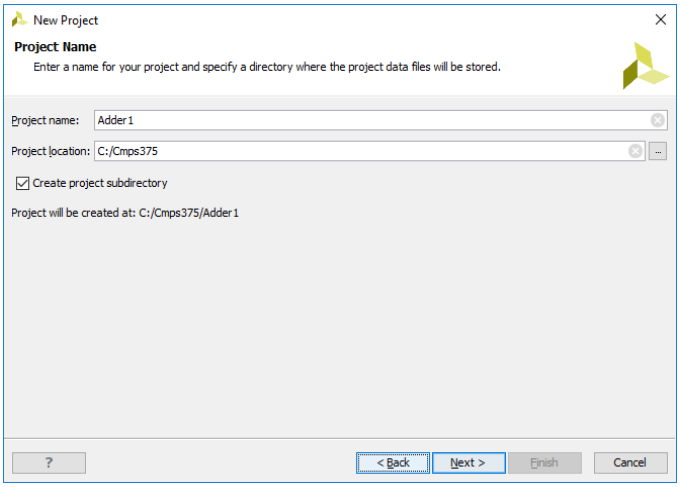

Figure 3. Project Name and Location

   d. In the *Project Type* form, select the *RTL Project* option, and click *Next*.
   e. In the *Add Sources* form as shown in Figure 4, click on the *Add File* button, then browse directory, select **Adder1.vhd**.
   f. Verify the ***Copy Constraints files into projects*** box is checked. Select ***VHDL*** as the *Target Language* and as the *Simulator language*. Then, click *Next*.

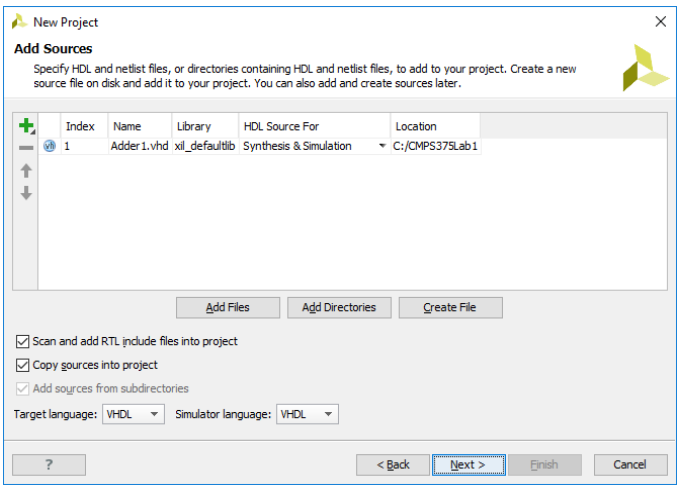

Figure 4. Add Source Form

   g. In the *Add Existing IP* form, click *Next*.
   h. In the *Add Constraints* form as shown in Figure 5, click on the *Add File* button, then browse directory, select **Nexys4DDR_Master.xdc**.

i. Verify the ***Copy Constraints files into projects*** box is checked. Then, click *Next*.

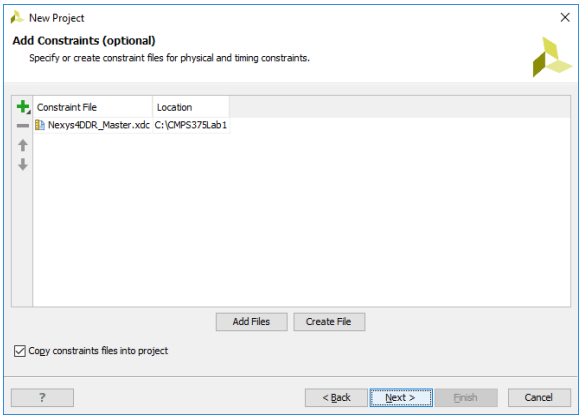

Figure 5. Add Constraints Form

j. In the *Default Part* form as shown in Figure 6, select device Artix-7 FPGA **XC7A100TCSG324-1**. Then click *Next*.

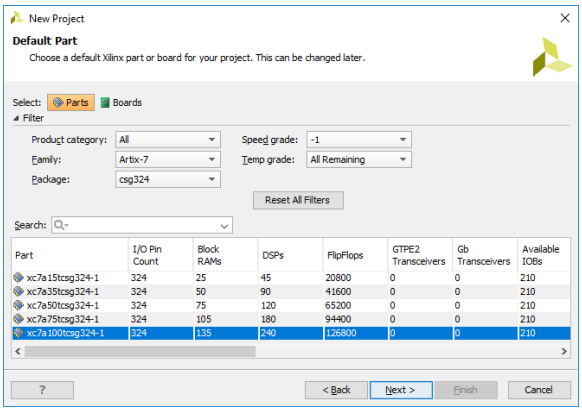

Figure 6. Default Part Form

k. In the *New Project Summary* as shown in Figure 7, click *Finish* to create the project.

Figure 7. New Project Summary Form

2. Write VHDL code for the design circuit and view its schematic entry
   a. In the *Sources* pane as shown in Figure 8, double-click the Adder1.vhd entry to open the file in the text mode. The file contents of **Adder1.vhd** are shown in Figure 9.

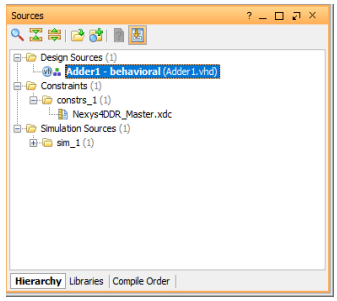

Figure 8. Sources Pane

   b. **Add** your team's and members' names as Authors.
   c. **Insert** two lines (one for cOut and the other for sum) of VHDL code.

```
-------------------------------------------------------------------------------
-- Description:   1-bit Adder
-- Project:       Adder1
-- Program-ID:    Adder1.vhd
-- Authors:            -- Add your team's and members' names as Authors Here
-- Package:       Xilinx Nexys4 DDR Board
-- Device:        Artix-7 FPGA (XC7A100TCSG324-1)
-- Software:      Vivado Design Suite
-------------------------------------------------------------------------------

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY Adder1 IS
   PORT (
      a, b, cIn : IN std_logic;
      cOut, sum : OUT std_logic);
END Adder1;

ARCHITECTURE behavioral OF Adder1 IS
BEGIN

    -- Insert two lines (one for cOut and the other for sum) of VHDL code Here

END behavioral;
```

Figure 9. VHDL code for Adder1 Design Circuit: Adder1.vhd

   d. In the *Flow Navigator* pane, click *Open Elaborated Design* entry under the *RTL Analysis* to perform RTL (Register-Transfer Level) analysis on the source file. Click the ***Schematic*** to see a logic view of the design as shown in Figure 10.

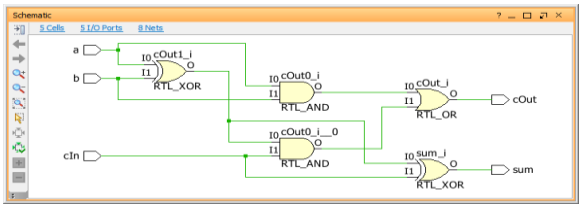

Figure 10. Schematic of Adder1: 1-bit Adder

3. Use the provided user constraint file (XDC) to constrain pin locations
   a. In the *Sources* pane, double-click Nexys4DDR_Master.xdc entry under *Constraints* to open the file in the text mode. The file contents of **Nexys4DDR_Master.xdc** shown in Figure 11.
   b. **Edited** the port names for the designed model, Adder1.vhd.

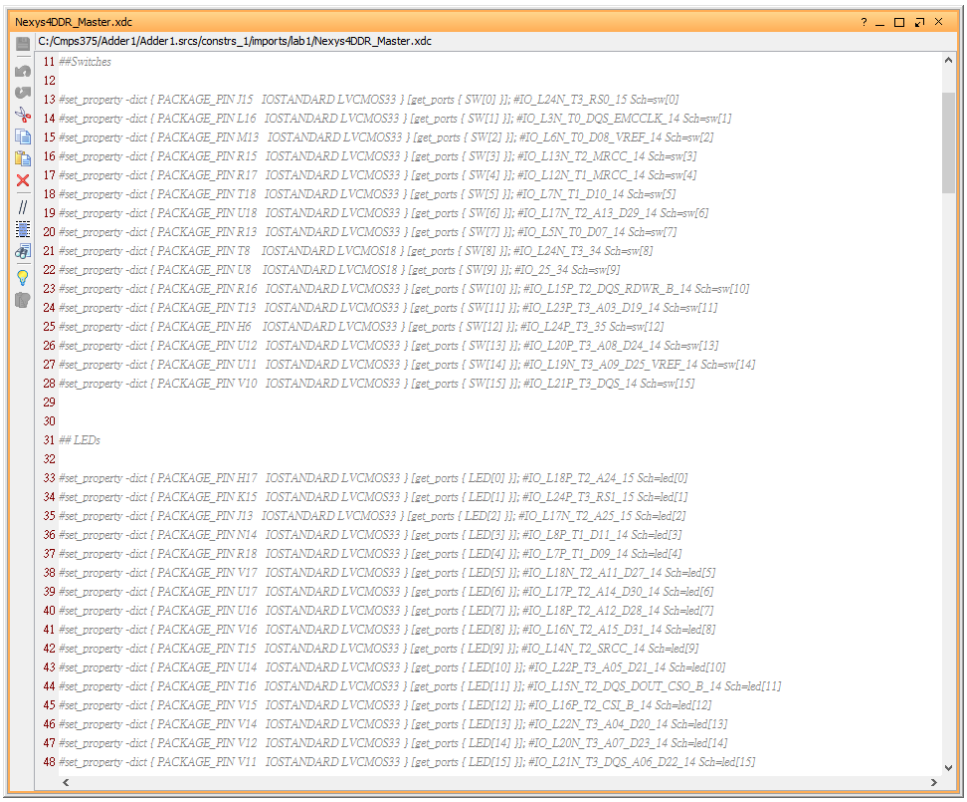

Figure 11. Editing I/O Constraints: Nexys4DDR_Master.xdc

   c. Click on the *I/O Panning* under the *Layout* menu. Notice that once RTL analysis is performed, I/O Planning layout is available. In Figure 12, the design ports (a, b, cIn, cOut, and sum) are listed in the *I/O Ports* tab of the *Console View* area.

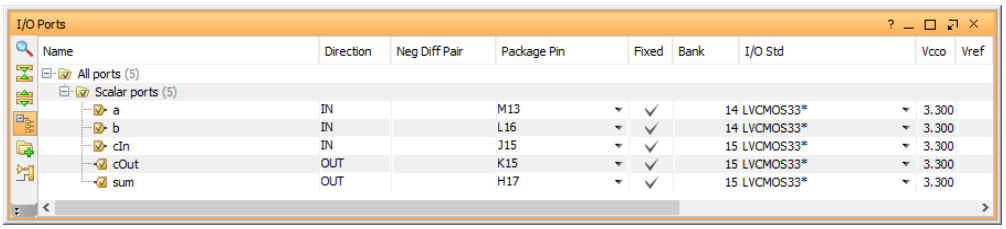

Figure 12. I/O Ports in Console View: a, b, cIn, cOut, and sum

4. Simulate the design using the XSim Simulator
   a. In the *Project Manager* tasks of the *Flow Navigator* pane, click the *Add Sources*, select the *Add or Create Simulation Sources* option as shown in Figure 13, and click *Next*.

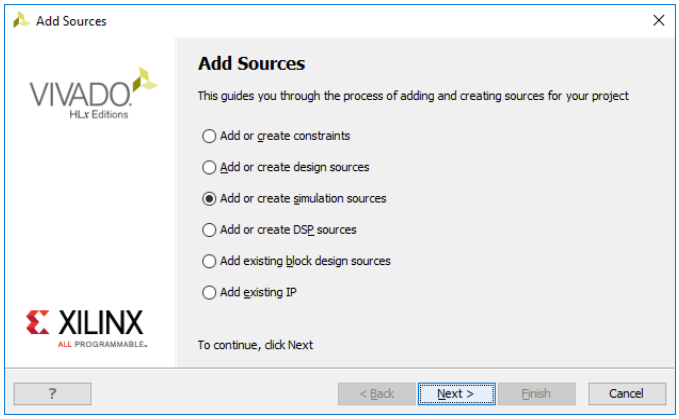

Figure 13. Selecting Simulation Sources option

   b. In the *Add or Create Simulation Sources* form as shown in Figure 14, click on the *Add File* button, then browse directory, select **Adder1_tb.vhd**. Then, click *Finish*.

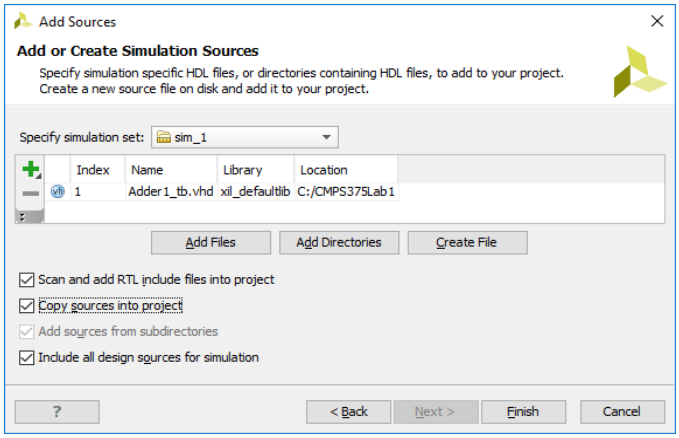

Figure 14. Add Simulation Sources: Adder1_tb.vhd

c. In the *Sources* pane, double-click the Adder1_tb.vhd entry to open the file in the text mode. The file contents of **Adder1_tb.vhd** are shown in Figure 15.
d. Do **not** change this simulation file!

```
-------------------------------------------------------------------------------
-- Description:   1-bit Adder (Test Bench)
-- Project:       Adder1
-- Program-ID:    Adder1_tb.vhd
-- Author:        Kuo-pao Yang
-- Package:       Xilinx Nexys4 DDR Board
-- Device:        Artix-7 FPGA (XC7A100TCSG324-1)
-- Software:      Vivado Design Suite
-- Notes:   1. ENTITY: No code (empty)
--          2. ARCHITECTURE: Put simulation code under PROCESS
-------------------------------------------------------------------------------

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY Adder1_tb IS
END Adder1_tb;

ARCHITECTURE simulate OF Adder1_tb IS
   COMPONENT Adder1
      PORT (
         a, b, cIn : IN std_logic;
         cOut, sum : OUT std_logic);
   END COMPONENT;
   SIGNAL a, b, cIn: STD_LOGIC;
   SIGNAL cOut, sum: STD_LOGIC;
BEGIN
   uut: Adder1 PORT MAP (a, b, cIn, cOut, sum);

   stimulus: PROCESS
   BEGIN
         -- test bench stimulus code
         a     <= '0';     b     <= '0';     cIn   <= '0';
         WAIT FOR 100 ns;
         a     <= '0';     b     <= '0';     cIn   <= '1';
         WAIT FOR 100 ns;
         a     <= '0';     b     <= '1';     cIn   <= '0';
         WAIT FOR 100 ns;
         a     <= '0';     b     <= '1';     cIn   <= '1';
         WAIT FOR 100 ns;
         a     <= '1';     b     <= '0';     cIn   <= '0';
         WAIT FOR 100 ns;
         a     <= '1';     b     <= '0';     cIn   <= '1';
         WAIT FOR 100 ns;
         a     <= '1';     b     <= '1';     cIn   <= '0';
         WAIT FOR 100 ns;
         a     <= '1';     b     <= '1';     cIn   <= '1';
         WAIT FOR 100 ns;
         a     <= '0';     b     <= '0';     cIn   <= '0';

         WAIT;
   END PROCESS;
END simulate;
```

Figure 15. VHDL code for Adder1 Test Bench Simulation File: Adder1_tb.vhd

e. Simulate the design for 1000 ns using the XSIM simulator. Click on *Simulation settings* of *Flow Navigator* pane, select the *Simulation* tab, and change the simulation time to **1000 ns** as shown in Figure 16.

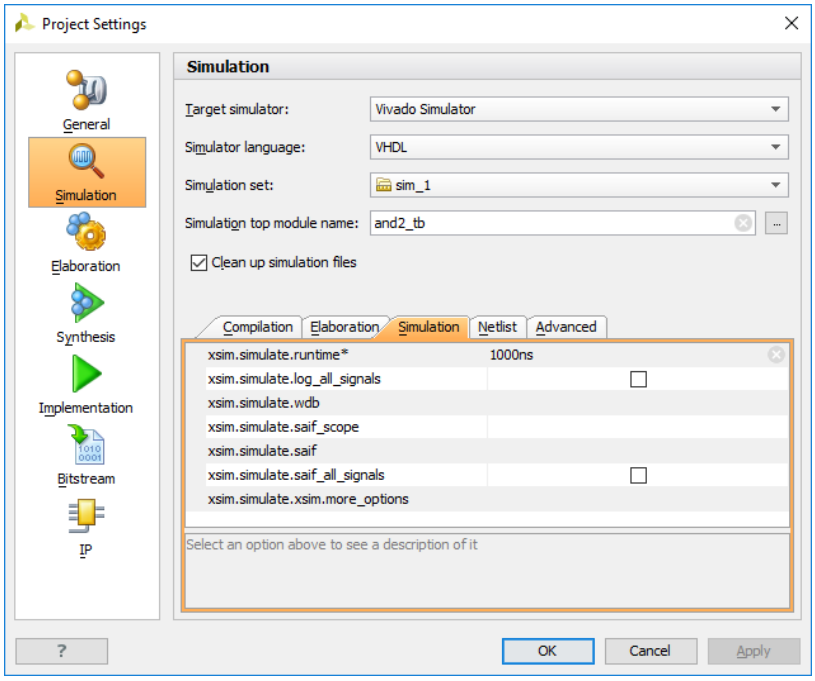

Figure 16. Simulate the Design for 1000 ns

f. Click on the *Run Simulation* under the *Flow Navigator* pane. Select the ***Run Behavioral Simulation*** option. The simulation results are shown in Figure 17.

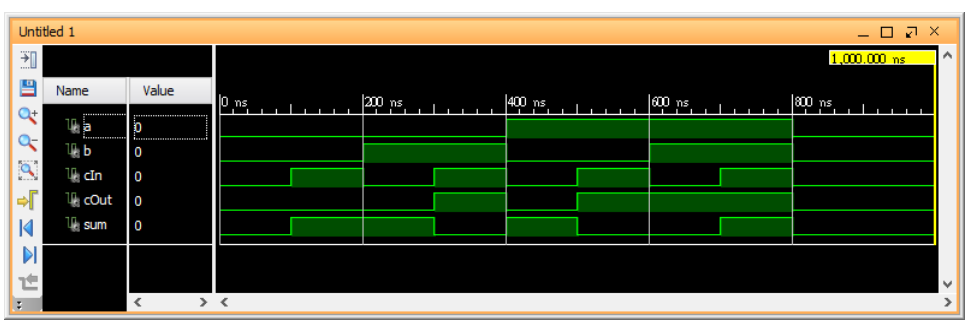

Figure 17. Simulation Results: Adder1_tb.vhd

5. Synthesize and implement the design
   a. Click on the *Run Synthesis* under the *Synthesis* tasks of the *Flow Navigator* pane. Note that: Vivado synthesis is the process of transforming an RTL-specified design into a gate-level representation. This synthesis is **timing-driven** and optimized for memory usage and performance.

   b. Click on the *Run Implementation* under the *Implementation* tasks of the *Flow Navigator* pane. Note that: Vivado implementation includes all steps necessary to place and route the netlist onto **device** resources, within the logical, physical, and timing constraints of the design.

6. Generate the bitstream
   a. Click on the *Generate Bitstream* entry under the *Program and Debug* tasks of the *Flow Navigator* pane.

7. Program and download the design to the FPGA device
   a. Make sure the Micro-USB cable is connected between the board and the PC. Power *ON* the switch on the board. Note that it does **not** need to connect the power jack and the board.
   b. Click on the *Open Target* of the *Hardware Manager* entry under the *Program and Debug* tasks of the *Flow Navigator* pane, and then select the *Auto Connect*.
   c. Click on the *Program Device* of the *Hardware Manager* entry under the *Program and Debug* tasks of the *Flow Navigator* pane, and then select **xc7a100t_1**.
   d. Make sure the *Bitstream file* of *Program Device* form is **Adder1.bit** as shown in Figure 18. Then, click ***Program*** to program the FPGA. Note that: the *DONE* light on the board will light when the device is programmed.

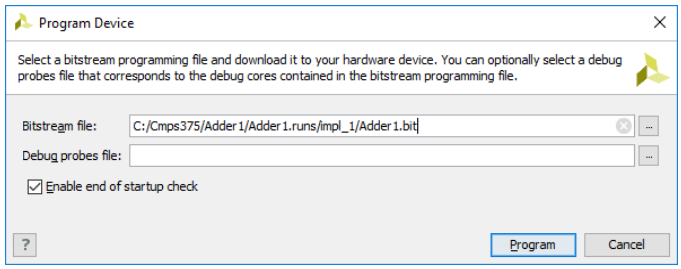

Figure 18. Bitstream for Nexys4 DDR: Adder1.bit

8. Test and verify functionality of the designed circuit
   a. Verify the functionality by flipping switches and observing the output on the LEDs.

Scalar Ports	***Direction***	***Package Pin***	***Name***
a	IN	M13	SW[2]
b	IN	L16	SW[1]
cIn	IN	J15	SW[0]
cOut	OUT	K15	LED[1]
sum	OUT	H17	LED[0]