Lab 1
Introduction to VHDL

Name_____ Class _____ Date _____

OBJECTIVES:

Upon completion of this laboratory exercise, you should be able to:

- Enter a simple combinational logic circuit in VHDL using the Quartus II Text Editor.
- Assign a target device and pin numbers and compile a VHDL design file.
- Write a test specification for a VHDL design entity and create a simulation to verify the correctness of the design.
- Download the file to an Altera CPLD on the Altera DE1 or DE2 board.

REFERENCE READING:

Dueck, Robert K., *Digital Design with CPLD Applications and VHDL*:
Chapter 5: Introduction to VHDL

EQUIPMENT REQUIRED:

Altera DE1 or DE2 Development and Education Board
AC Adapter, minimum output: 7 VDC, 250 mA DC
USB Blaster Cable
Quartus II Software
Anti-static wrist strap

EXPERIMENTAL NOTES:

Before working on this lab, you may wish to review the Digital Logic tutorial lab, *Introduction to Quartus II*, available on the SharePoint website for this course.

VHDL stands for VHSIC Hardware Description Language. (VHSIC = Very High Speed Integrated Circuit.) VHDL is an industry standard programming language for simulation and synthesis of digital circuits. In this lab, we will use VHDL to enter the designs for some simple combinational logic circuits, using VHDL constructs for Boolean equations and truth tables.

Every VHDL design requires an **entity declaration**, which describes the inputs and outputs of the design, and an **architecture body**, which describes the internal relationship between inputs and outputs. Within the architecture body, we can use many different language statements to describe our design. We will use two of the simplest: a **concurrent signal assignment statement**, which can be used to implement a Boolean expression and a **selected signal assignment statement**, which can be used, among other things, to implement a truth table.

Lab 1
Introduction to VHDL

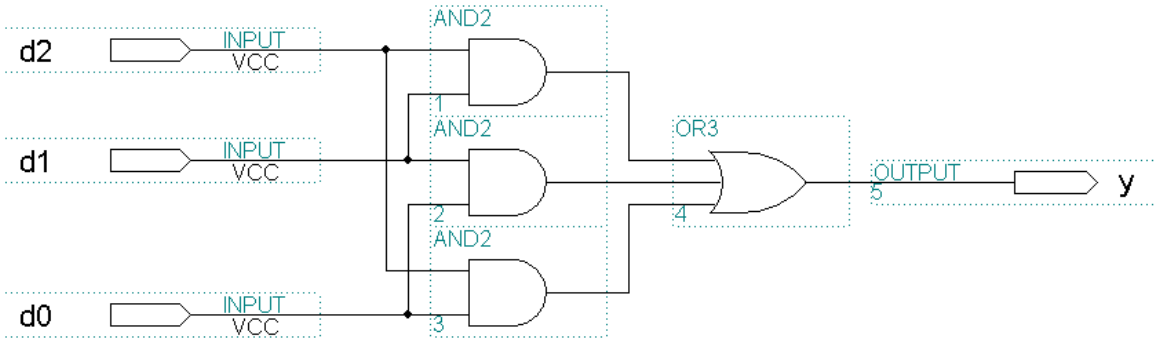

Figure 1 Majority Vote Circuit

Figure 1 shows a majority vote circuit. The circuit has the Boolean expression $Y = D_2D_1+D_1D_0+D_2D_0$. The output is HIGH if at least two out of three inputs are HIGH. A concurrent signal assignment statement would encode this equation as follows:

```
y <= (d2 and d1) or (d1 and d0) or (d2 and d0);
```

The complete VHDL file is shown below:

Example only: Do not enter this text.

```
-- majority_vote_vhdl.vhd
-- majority vote circuit using a concurrent signal assignment statement
library ieee;
use ieee.std_logic_1164.all;

entity majority_vote_vhdl is
port(
    d2, d1, d0 : in   std_logic;
    y          : out  std_logic);
end majority_vote_vhdl;

architecture a of majority_vote_vhdl is
begin
    y <= (d2 and d1) or (d1 and d0) or (d2 and d0);
end a;
```

Refer to Section 5.1 in *Digital Design with CPLD Applications and VHDL* for a more detailed description of the various parts of this design file.

Lab 1
Introduction to VHDL

Encoding a Truth Table in VHDL

We can encode a truth table in VHDL by using a selected signal assignment statement. Examine the truth table in Table 1.

D_2	D_1	D_0	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Table 1 Sample Truth Table

In the entity declaration of a VHDL file, we can define the combined input values of D_2, D_1, and D_0 as a 3-bit vector of type STD_LOGIC_VECTOR:

```
d: in std_logic_vector(2 downto 0);
```

This truth table can be encoded by the following selected signal assignment statement:

```
with d select
   y   <=  '1' when "001",
           '1' when "010",
           '1' when "100",
           '0' when others;
```

The output value is shown on the left side of each line of the statement and the corresponding input combinations on the right side. The **others** clause specifies the default value of the output, which in this case is '0'. This method is much simpler for cases that would otherwise require encoding a long and difficult-to-read Boolean expression.

The complete VHDL file is shown next.

Example only: Do not enter this text.

```
-- table1.vhd
-- truth table, encoded using a selected signal assignment statement
library ieee;
use ieee.std_logic_1164.all;

entity table1 is
port(
   d: in  std_logic_vector (2 downto 0);
   y: out  std_logic);
end table1;

architecture arch_name of table1 is
begin
   with d select
       y   <=  '1' when "001",
               '1' when "010",
               '1' when "100",
               '0' when others;
end arch_name;
```

Simulating a VHDL Design Entity

In order to verify the correctness of our VHDL design, we should create a simulation using the Quartus II Waveform Editor. To do so, first we should examine our design, determine the way it should operate, and create a test specification for the design. We can then use this test spec to create our simulation waveforms.

Example: *Write a test specification for the majority vote circuit example above and use it to make the simulation waveforms for the design entity.*

For many combinational circuits, a simple test spec involves examining the output under all input conditions. That is, taking its truth table.

Test Specification
• Group the inputs together as a single unit: **d[2..0]**.
• Apply an increasing binary count, from 000 to 111, to the **d[2..0]** group.
• The output, **y**, should go HIGH for input values 011, 101, 110, and 111, the states where at least two of the three inputs are HIGH.

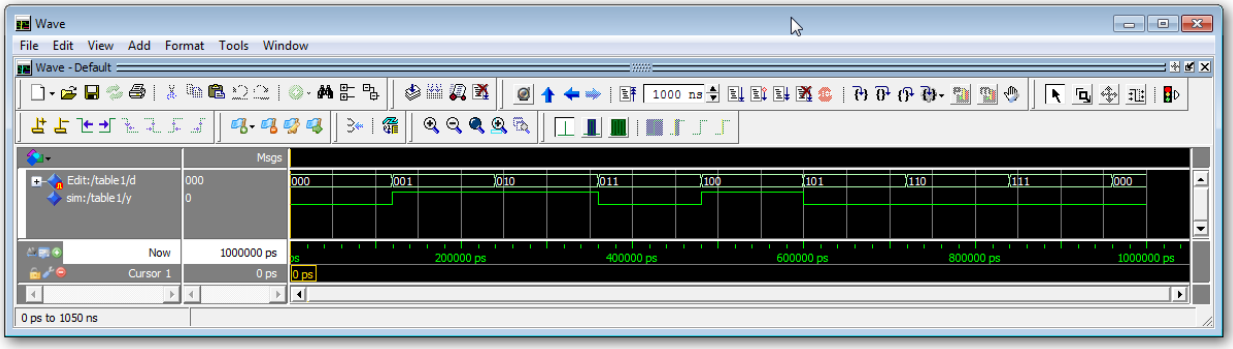

Figure 2 Simulation Report

Figure 2 shows the simulation waveforms that fulfill these conditions. The mechanics of creating this simulation are described in the tutorial lab, *Introduction to Quartus II*, available on the SharePoint website for this page. The two design entities in this lab can be simulated using similar criteria.

PROCEDURE:

Encoding a Circuit with a Concurrent Signal Assignment Statement

Refer to the half-adder circuit shown in Figure 3.

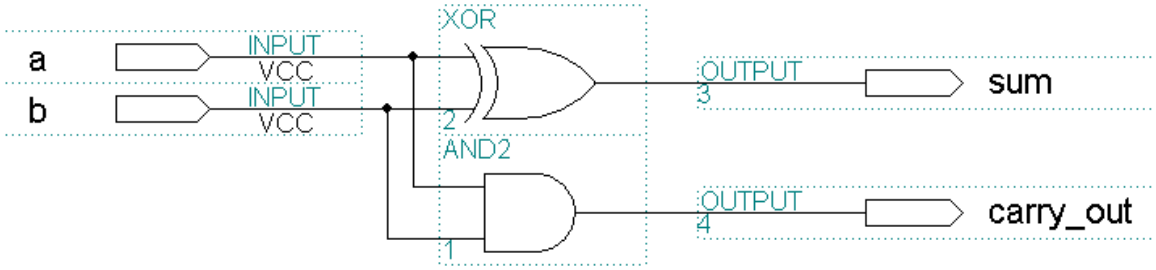

Figure 3 Half Adder

Design Entry

- Create a new folder called **g:\qdesigns\lab01\half_add**.
- Use the Quartus II Text Editor to enter the VHDL file for the half adder circuit, using a concurrent signal assignment statement, as described in the Experimental Notes for this lab exercise.
- Save the file as **g:\qdesigns\lab01\half_add\half_add.vhd** and use the file to create a project in Quartus II. (Make sure that the box labeled **Create new project based on this file** is checked.)
- In the New Project Wizard, assign the device family and target device, as appropriate for your board. (See Table 2.)
- From the **Assignments** menu, select **Pins** or **Assignment Editor.** Assign the pin numbers in Table 2 to the circuit inputs and outputs, choosing the column

that pertains to your CPLD/FPGA board. When you have finished assigning the pin numbers, compile the project.

Board:	**DE-1**	**DE-2**
Family:	**Cyclone II**	**Cyclone II**
Device:	**EP2C20F484C7**	**EP2C356F72C6**
Pin Name	**Pin Number and I/O Device Name**	
a	L22 (SW[1])	N26 (SW[1])
b	L21 (SW[0])	N25 (SW[0])
sum	R20 (LEDR[0])	AE23 (LEDR[0])
carry_out	R19 (LEDR[1])	AF23 (LEDR[1])

Table 2 Pin Assignments for Half Adder

Simulation

- Write a test spec for the half adder function.

Test Spec:

- Use the test spec to create a Quartus II simulation for the half adder

Instructor's Initials_____

Programming the CPLD/FPGA board

- Download the project to your CPLD/FPGA trainer board.
- Take the truth table of the circuit and show it to your instructor.

Truth Table:

A	B	Carry_out	Sum

Table 3 Half Adder Truth Table

Instructor's Initials_____

- Close the project for the half adder by selecting **Close Project** from the **File** menu.

Lab 1
Introduction to VHDL

Encoding a Circuit with a Selected Signal Assignment Statement

Examine the SOP logic gate network shown in Figure 4. Inputs are labeled d3, d2, d1, d0.

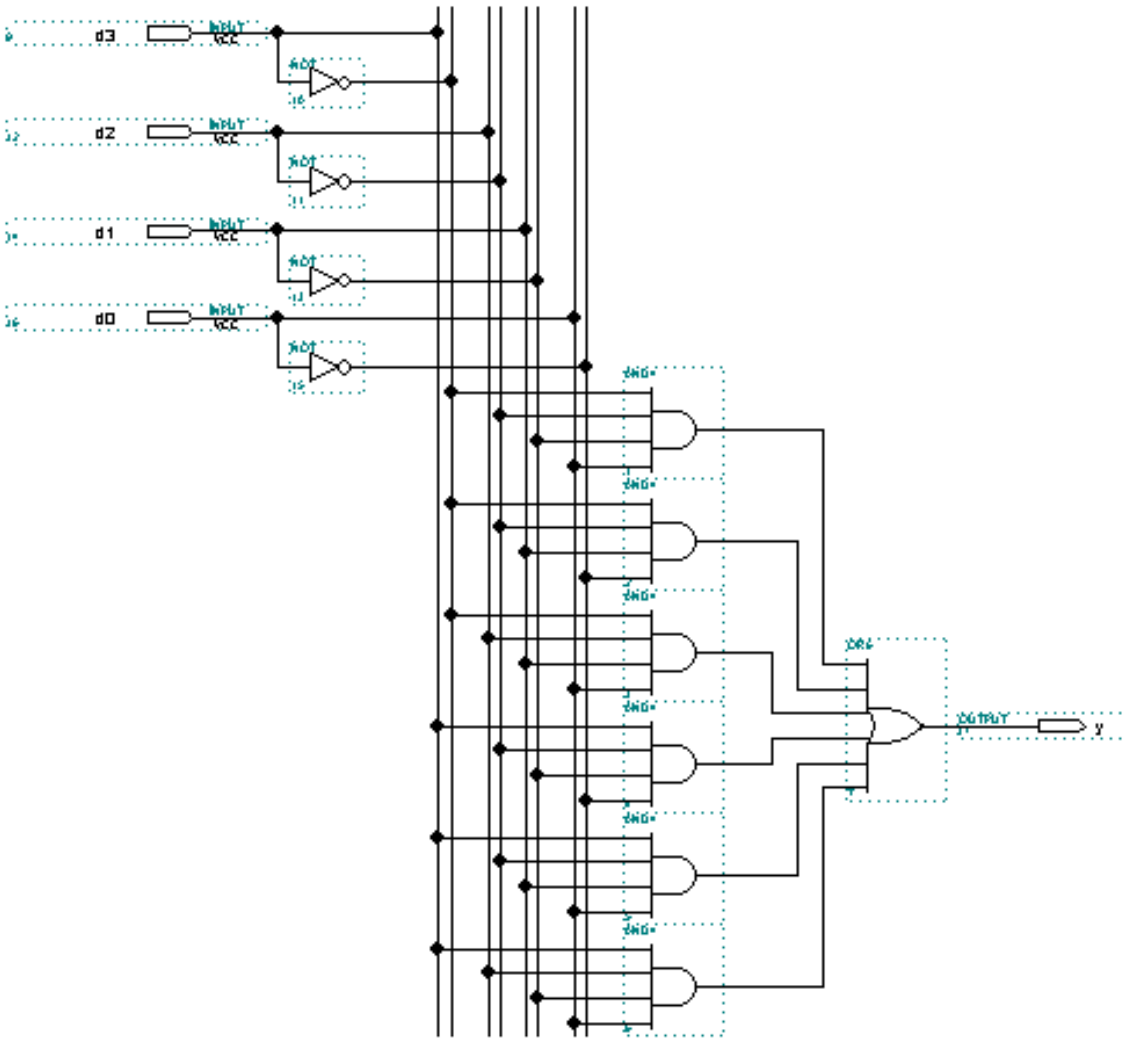

Figure 4 Sum-of-Products Logic Gate Network

Write the Boolean expression of the logic gate network shown in Figure 4.

Boolean expression:

Note that the Boolean expression is already in its simplest SOP form. Since this equation would be difficult to read (and code) in VHDL, we will use a selected signal assignment statement to encode the circuit's truth table.

Design Entry

- Create a new folder called **g:\qdesigns\lab01\sop1**.
- Write a VHDL file for the logic gate network in Figure 8.4, using a selected signal assignment statement to encode its truth table. Write the **d** inputs as a single variable of type STD_LOGIC_VECTOR.
- Save the file as **g:\qdesigns\lab01\sop1\sop1.vhd** and use this file to create a new project in Quartus II.
- In the New Project Wizard, assign the device family and target device, as appropriate for your board. (See Table 4.)
- Assign pin numbers to the VHDL file as shown in Table 4.

Board:	**DE-1**	**DE-2**
Family:	**Cyclone II**	**Cyclone II**
Device:	**EP2C20F484C7**	**EP2C356F72C6**
Pin Name	**Pin Number and I/O Device Name**	
d[3]	V12 (SW[3])	AE14 (SW[3])
d[2]	M22 (SW[2])	P25 (SW[2])
d[1]	L21 (SW[1])	N26 (SW[1])
d[0]	L22 (SW[0])	N25 (SW[0])
y	R20 (LEDR[0])	AE23 (LEDR[0])

Table 4 Pin Assignments for Figure 8.4

Simulation

- Write a test spec for the SOP function in you VHDL file.

Test Spec:

- Use the test spec to create a Quartus II simulation for the VHDL design entity.

Instructor's Initials_____

Lab 1
Introduction to VHDL

Programming the CPLD/FPGA board

- Download the project to the DE1 or DE2 board and test all combinations of D inputs.
- Write the truth table for the circuit in Figure 4.

D_3	D_2	D_1	D_0	Y

Table 5 Truth Table for Figure 4

Instructor's Initials_____

- Close the project for the SOP circuit by selecting **Close Project** from the **File** menu.