Combinational Circuits Using VHDL

**Due:** By 6:00pm on Wednesday April 16.

In this lab we introduce the use of a design language that can simplify the design process.

## 1 VHDL

The use of a Hardware Description Language (HDL) can simplify the design process by allowing the user to program the behavior of a circuit and let the synthesis tools create the logic-circuit structure. The text book *Digital Design: Principles and Practices, 3rd Edition* by Wakerly has a good overview of HDL’s in section 4.7.

The program structure of VHDL consists of two basic parts. The *entity* is simply a declaration of the module’s inputs and outputs. The *architecture* contains a detailed description of the module’s behavior.

```vhdl
library IEEE;
use IEEE.std_logic_1164.all;

entity sam is
  port (  
    A: in STD_LOGIC;
    B: in STD_LOGIC;
    C: in STD_LOGIC;
    D: in STD_LOGIC;
    X: out STD_LOGIC;
    Y: out STD_LOGIC
  );
end sam;

architecture sam_arch of sam is
begin
  X <= (A and B and C) or (C and D);
  Y <= (A and C) or (B and D);
end sam_arch;
```

## 2 Programming VHDL with Xilinx Tools

Xilinx ISE Design Manager allows you to work in either schematic based, or HDL based design entry environments, but you must declare the type of design entry when you start a new project. When working
in the schematic based “flow” you can not add VHDL files to the project, even though you can access the HDL editor. Similarly you can not add schematics to a project declared as HDL based flow.

When starting a new project choose the Flow type to be XST VHDL. In the Project Manager, click on Project, select New Source, select VHDL Module, and give it a filename. Click Next twice and then Finish. At this point, the software will generate a VHDL entity/architecture template and open it in the HDL editor.

Next, the port declarations must be made in the port( ); statement in the entity body. Now, enter your statements in the architecture body. This is where you implement the VHDL functionality. After you have completed your design entry, save the document. Next, under “Processes for Current Source” window, run the synthesis by double-clicking “Synthesis” in order to analyze the hierarchy and perform the syntax checking.

After synthesis, you can perform the simulation steps as you did in the previous labs. Once you have completed simulation, the remaining steps for downloading the design onto the board are the same as you performed before.

### 3 Switch Matrix to Seven Segment Converter

In this lab we will be designing logic circuits that convert a pressed switch in a 2 by 2 switch matrix to a seven segment code. This in turn can drive a seven segment display in order to produce a visual indication of which switch was pressed.

Figure 1 shows the organization of the system. We will first focus on the operation of the switch matrix. If no switch is pressed, all the inputs, \( r_1 \), \( r_2 \), \( c_1 \), and \( c_2 \) will be high since they are connected to Vcc through pull-up resistors. If any one switch is pressed, then either \( r_1 \) or \( r_2 \) and either \( c_1 \) or \( c_2 \) will go low. The \( r \) input that goes low gives the row of the pressed switch. The \( c \) input that goes low gives the column of the Pressed switch. Rather than implement the switch matrix itself, we will use the Digilab buttons directly as the \( r \) and \( c \) inputs.
4 Preliminary Questions

1. Calculate the minimal functions needed to implement the 2x2 switch matrix to seven segment code converter circuit for the following specifications:

- Single button presses should cause the display of the corresponding switch number on a seven segment display. Don’t forget that the seven segment display is active low.
- Use the switch numbering shown in Figure 1. That is, when SW1 is pressed, 1 should appear on the display. Remember that when the circuit is implemented on the Digilab board, pressing SW1 will require that the switches corresponding to \( r_1 \) and \( c_1 \) will have to be taken low, while the other inputs remain high.
- Display the switch (button) number on each separate seven segment display. For example, if you push SW1 (button one), the first seven segment display should show the number one. If you were to press the second switch SW2 (button two), the number two should show on the second seven segment display, etc.
- When no switches are pressed, the display should simply remain blank.
- Do the design assuming that multiple switch presses, and illegal input combinations cannot occur (such as one or more of \( r_1 \) through \( r_2 \) low without at least one of \( c_1 \) through \( c_2 \) also low). If more than one switch (button) is pressed at the same time, nothing should be displayed.

Since both the input and output signals are active low it may be helpful to design the circuit using active low (type 2) logic. Look at what low signals it takes to drive signals low rather than looking at the equations needed to drive signals high.

2. Using simple boolean equations write the VHDL statements needed to implement the 2x2 switch matrix to seven segment code converter circuit. Turn in the equations with your lab report and any other items such as truth tables, etc. that you use during the design.

3. Obtain printouts of the .vhd design file, and the simulation waveforms. These will be useful during debugging, and also should be turned in with the Lab report.

5 The Lab

1. Implement, test, and download your design to the FPGA board. After verifying correct functionality, have the entire design passed off by the TA including the VHDL design file.