CpE 272

Section 002

Lab 6: Behavioral Modeling Of Combinational Logic Circuits Using VHDL-1

Prepared By:

Olumuyiwa Oyeleye

Prepared For:

Cletis Nicklow

October 6, 2010

**Introduction**

The main goal of this lab was to teach students an alternative to the “structural modeling” VHDL coding style called “behavioral modeling”, in which the behavior of a circuit is described based on the inputs and outputs.

**Experiments**

**Part I**

The objective of this part of the lab was to design a 4-bit multiplexer and verify its operations

**Methodology**

The first thing the group did was to create a new VHDL project named Lab6, next in line was to atate the four inputs A,B,C and D of the multiplexer, the two selects X1 and X2, and the single output of the multiplexer Y. After all this was done, the only thing left to do was to code and test the multiplexer. The group coded the multiplexer using if statements, in which all the possible combinations for the selects X1 and X2 were stated and assigned to either inputs A,B,C and D. For example, the first “if” statement stated that “if X1 and X2 were both 0, then input A would be selected to generate the output Y.”

Below is the VHDL code for the multiplexer:

**LIBRARY ieee;  
USE ieee.std\_logic\_1164.all;  
  
ENTITY Lab6 IS  
  
PORT (  
    A : IN STD\_LOGIC;  
    B : IN STD\_LOGIC;  
    C : IN STD\_LOGIC;  
    D : IN STD\_LOGIC;  
    X1 : IN STD\_LOGIC;  
    X2 : IN STD\_LOGIC;  
    Y : OUT STD\_LOGIC  
);  
END Lab6;  
      
ARCHITECTURE Behavior of Lab6 IS  
    begin  
    process(A,B,C,D,X1,X2)  
        begin  
            if(X1 = '0' and X2 = '0') then  
                Y<=A;  
            elsif(X1 = '0' and X2 = '1') then  
                Y<=B;  
            elsif(X1 = '1' and X2 = '0') then  
                Y<=C;  
            elsif(X1 = '1' and X2 = '1') then  
                Y<=D;  
            end if;  
    END process;  
END Behavior;**

The code was compiled and the inputs X1, X2, A, B, C and D were assigned to pins N25, N26. P25, AE14, AF14 and AD13 respectively, while output Y was assigned to pin AE23. Finally, the code was compiled again and programmed unto the FPGA.

**Results**

As demonstrated to the lab T.A, the multiplexer was a success as the different select combinations tried yielded the desired output.

**Part II**

The objective of this part of the lab was to design a 3 to 8 decoder using behavioral modeling, and also to program it on an FPGA and verify its operations.

**Methodology**

Just as in the previous experiment, the group started by creating a new VHDL project. Next the group came up with the truth table for the 3-input decoder where the input combination determined which one of the 8 outputs would have a high value, and after this the decoder was programmed. The first step taken to program the decoder was to state the inputs A, B and C (with A being the most significant bit and C the least), and the outputs Y0 through Y7. Next the team defined the outputs using the values from the truth table as a guide, so for example the truth table had the value of output Y2 to be 1 when the input combination read 2 in binary, that is all inputs except B had a value of 0.

Below is the VHDL code used to program the FPGA chip:

**LIBRARY ieee;  
USE ieee.std\_logic\_1164.all;  
  
ENTITY Lab6\_2 IS  
  
PORT (  
    A : IN STD\_LOGIC;  
    B : IN STD\_LOGIC;  
    C : IN STD\_LOGIC;  
    Y1 : OUT STD\_LOGIC;  
    Y2 : OUT STD\_LOGIC;  
    Y3 : OUT STD\_LOGIC;  
    Y4 : OUT STD\_LOGIC;  
    Y5 : OUT STD\_LOGIC;  
    Y6 : OUT STD\_LOGIC;  
    Y7 : OUT STD\_LOGIC;  
    Y8 : OUT STD\_LOGIC  
);  
END Lab6\_2;  
      
ARCHITECTURE Behavior of Lab6\_2 IS  
    begin  
        Y1 <= ((not A and not B) and not C);  
        Y2 <= ((not A and not B) and C);  
        Y3 <= ((not A and B ) and not C);  
        Y4 <= ((not A and B ) and C);  
        Y5 <= ((A  and not B) and not C);  
        Y6 <= ((A  and not B) and C);  
        Y7 <= ((A  and B ) and not C);  
        Y8 <= ((A  and B ) and C);  
END Behavior;**

The code was compiled and the inputs were assigned to pins N25, N26 and p25, while the outputs were assigned to pins AE23, AF23, AB21, AC22, AD22, AD23, AD21 and AC21. The code was re-compiled and programmed unto a FPGA.

**Result**

The design was a success as the decoder worked well yielding the expected output from a unique input combination. Below is the result of the experiment in the form of a truth table



**Part III**

The task in this part of the lab is to design a 3-input, 2-output priority encoder using behavioral modeling and to program the FPGA and verify its operation.

**Methodology**

The first thing the group did was to come up with a truth table which would serve as a guide when programming the encoder. Once this was accomplished, all that was left was to translate the truth table into a VHDL code. As usual a new project was created, the inputs A, B, C and outputs Y and Z were stated, and if statement were used to set the values of the outputs based on the input values.

Shown below is the VHDL code for the priority encoder

**LIBRARY ieee;  
USE ieee.std\_logic\_1164.all;  
  
ENTITY Lab6\_3 IS  
  
PORT (  
    A : IN STD\_LOGIC;  
    B : IN STD\_LOGIC;  
    C : IN STD\_LOGIC;  
    Y : OUT STD\_LOGIC;  
    Z : OUT STD\_LOGIC  
);  
END Lab6\_3;  
      
ARCHITECTURE Behavior of Lab6\_3 IS  
    begin  
    process(A,B,C)  
        begin  
            if(C = '1') then  
                Y <= '1';  
                Z <= '1';  
            elsif(B = '1' and C = '0') then  
                Y <= '1';  
                Z <= '0';  
            elsif(A = '1' and B = '0' and C = '0') then  
                Y <= '0';  
                Z <= '1';  
            else  
                Y <= '0';  
                Z <= '0';  
            end if;  
    END process;  
END Behavior;**

The inputs A, B and C were assigned to pins N25, N26 and P25 and the outputs Y and Z were assigned to pins AE23 and AF23 respectively. In conclusion the code was compiled and programmed unto the FPGA.

**Results**

The expected results were obtained as all the input combinations were tested with the corresponding LED lighting up. Shown below is the truth table for ht operation



**Post lab questions for Lab 6**

1. A decoder converts coded data back into its original form, it can also be used to convert a binary number into a decimal number and vice-versa.
2. Behavior modeling is VHDL coding style in which the behavior of a given circuit is described relative to its inputs and outputs. It is useful because it makes the designing of circuits that have “conditions” easy. By circuits that have conditions I mean circuits where selections have to be made from the available inputs provided.

**Pre lab questions for Lab 7**

1. Modular design is simply a process of designing a system by dividing it into smaller bits, and adding those bits up.
2. Comparators are used to compare the magnitude of two *n* bit numbers represented by pairs of *n* bit lines.