COEN 313 Lab 1 (UR-X)

by

Mamadou Diao Kaba

27070179

Performed on

January 31th, 2023

"I certify that this submission is my original work and meets the Faculty's Expectations of Originality"

Objectives

The objectives of the first laboratory of COEN 313 are:

- To familiarized with the VHDL simulation software tool by using Modelsim and its different tools like the DO files.
- To get a basic introduction to using the Xilinx Vivado software tools to program the Digilent Nexys-A7100T FPGA board.
- To demonstrate understanding of a Full-adder circuit by coding and implementing it to the Digilent Nexys-A7100T FPGA board.

Results

To demonstrate understanding of a full-adder circuit on the Nexys board, it required to code the circuit using VHDL. First, a half adder was coded. A half-adder adds two binary digits inputs and generates a carry and a sum. To perform the sum operation, a XOR-gate is applied to both the inputs, and AND-gate is applied to both inputs to produce the carry. Using two half-adders, a full-adder can be constructed. A full-adder circuit, adds 3 one-bit numbers, where two of the three bits are operands and the other bit is a carried in. The produced output is 2-bit output: carry and sum. Here is a truth table of what to be expected for the full-adder demonstration on the Nexys board using the LED outputs:

LED outputs Nexys-A7100T	L16	M13	J15	H17	K15
Inputs/outputs	A	B	C-in	SUM	C-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

This truth table matches perfectly the demonstration performed on the Board for this lab.

The waveform shown in the previous page represents the simulation of all the inputs and outputs of the full-adder VHDL code. The waveform shows a perfect simulation a full-adder.

Questions

1. What is the advantage of using the -r option in a force command within a DO file?

Within a DO file, the advantage of using the -r option in force command is that if we input a number N after the -r, it will force the recursion of the force command N times.

2. Briefly explain two methods of creating a repeating periodic signal using DO files.

It can be done using the command run plus the number of times the repeated periodic signal should be used. It can also be done by using the -r command.

3. What is a .xdc file used for?

A xdc.file is used in the Vivado IDE to configurated the inputs and outputs of the code to the outputs (LED signals) of the Nexys board.

Conclusion

To conclude, this lab offered a good introduction of the VHDL simulation software tool by using Modelsim and its different tools like the DO files. The implementation of the full-adder circuit was successful on the Digilent Nexys-A7100T FPGA board as well as its simulation on the Modelsim software.

Appendix

A) Code and xdc files

FULL-ADDER VHDL

```
library ieee;
use ieee.std_logic_1164.all;

entity full_adder is
        port(carry_in, input1, input2 : in std_logic;
                sum_out, carry_out : out std_logic);
end full_adder;

architecture structural of full_adder is

-- declare a half-adder component

component half_adder
        port ( in1, in2 :   in std_logic;
                carry, sum : out std_logic);
end component;

-- declare internal signals used to â€œhook upâ€  components
-- and to communicate to the display decoder process

signal carry1, carry2     : std_logic;
signal sum_int            : std_logic;

-- declare configuration specification
-- NOTE: we want to use the half adder with true outputs
-- not the inverted ones we synthesized earlier!!

for ha1, ha2 : half_adder use entity
WORK.half_adder(true_outputs);

begin

-- component instantiation

ha1: half_adder port map(in1 => input1, in2 => input2,
                         carry => carry1, sum => sum_int);

ha2: half_adder port map(in1 => sum_int, in2 => carry_in,
                         carry => carry2, sum => sum_out);

carry_out <= carry1 or carry2;

-- Nexys boards has active high LED outputs
-- so no need to negate top level output ports

end structural;
```

HALF-ADDER VHDL

```
library ieee;
use ieee.std_logic_1164.all;

entity half_adder is
        port ( in1, in2 :   in std_logic;
                carry, sum : out std_logic);
end half_adder;

architecture true_outputs of half_adder is
begin
        carry <=  (in1 and in2);
        sum   <=  (in1 xor in2);
end true_outputs;
```

FULLADDER.XDC

```
# Vivado does not support old UCF syntax
# must use XDC syntax

set_property -dict { PACKAGE_PIN J15 IOSTANDARD LVCMOS33 } [ get_ports { carry_in}];
set_property -dict { PACKAGE_PIN L16 IOSTANDARD LVCMOS33 } [ get_ports { input1 }   ] ;
set_property -dict { PACKAGE_PIN M13 IOSTANDARD LVCMOS33 } [ get_ports { input2 }  ] ;
set_property -dict { PACKAGE_PIN K15 IOSTANDARD LVCMOS33 } [ get_ports { carry_out } ] ;
set_property -dict { PACKAGE_PIN H17 IOSTANDARD LVCMOS33 } [ get_ports { sum_out }  ] ;
```