Lab Experiment No. 01:

Code Conversion in VHDL

By: Alexander Lukens (A20417036)
Instructor: Dr. Jafar Saniie
ECE 441-001
Lab Completion Date: 06/15/2020

Acknowledgement: I acknowledge all of the work (including figures and codes) belongs to me and/or persons who are referenced.

Signature:____Alexander Lukens____

I. Introduction

A. Purpose

The main purpose of this experiment is to introduce students to the Xilinx Vivado VHDL design environment. Students will become familiar with Xilinx Vivado UI sections and create their first VHDL code. This code will then be simulated in the Vivado software environment and optionally uploaded to a Xilinx Artix-7 based FPGA.

B. Background

Xilinx is a technology company based in California, USA that creates field programmable gate array (FPGA), system-on-chips (SoCs), and other semiconductor products. They compliment their hardware designs with a robust software environment for designing digital logic circuits using System Verilog and VHSIC-Hardware Description Language (VHDL). This allows silicon engineers to rapidly develop and test digital logic designs without physically designing them on a breadboard or waiting for dedicated, application specific silicon to be produced.

Binary Coded Decimal (BCD) is a code format used to represent decimal numbers in binary notation using 4-bits per decimal digit. Natural BCD represents decimal numbers by using their direct binary equivalent. An alternative format for representing BCD values is using Excess-3 BCD coding. In this coding scheme, “3” is added to the corresponding natural BCD value, resulting with the 10 digits representable by BCD (0-9 in base 10) having 3 values below and 3 values above the binary value range when using 4 digits for each BCD value. This is particularly useful for performing arithmetic operations on the BCD values, as the 1’s complement of the binary representation is the 9’s complement of the decimal representation of that BCD number.

II. Lab Procedure and Equipment List

A. Equipment

Equipment required for this lab includes the following:

- Xilinx Artix-7 based FPGA
- PC with Xilinx Vivado software

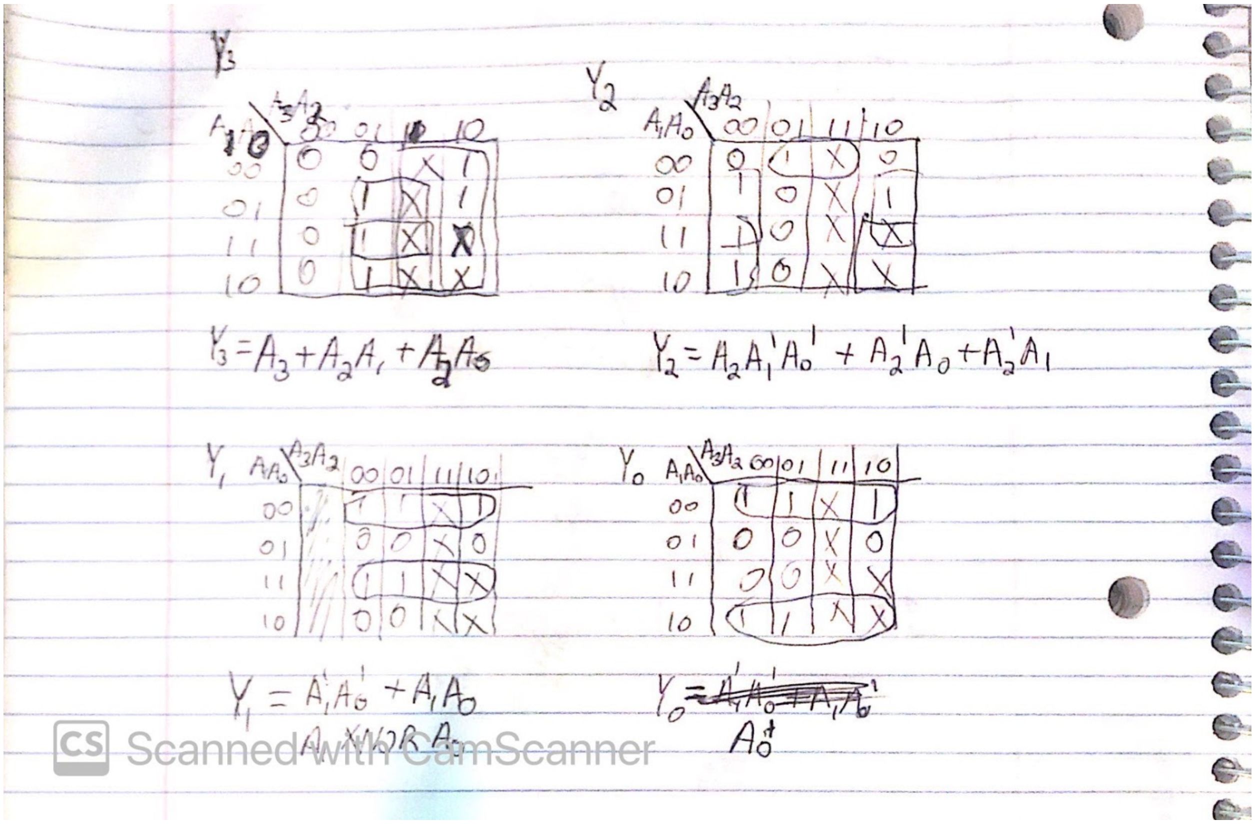
B. Procedure

Students will follow the instructions outlined in the ECE 742 - FPGA Lab Manual for experiment 1. This includes designing and implementing a code converter to translate natural BCD to Excess-3 BCD and Excess-3 to natural BCD.

III. Results and Analysis

A. Discussion

Initially, to create the code converter I utilized k-maps to find the simplified logic equations for each output value. A3-A0 are the input values and Y3-Y0 are the outputs. Utilizing the structural model of the code converter was tedious but did result with a perfectly functioning BCD to Excess-3 converter once the equations were implemented in VHDL.



Y_3

A_1A_0 \ A_3A_2	00	01	11	10
00	0	0	X	1
01	0	1	X	1
11	0	1	X	X
10	0	1	X	X

$Y_3 = A_3 + A_2A_1 + A_2A_0$

Y_2

A_1A_0 \ A_3A_2	00	01	11	10
00	0	1	X	0
01	1	0	X	1
11	1	0	X	X
10	1	0	X	X

$Y_2 = A_2A_1'A_0' + A_2'A_0 + A_2'A_1$

Y_1

A_1A_0 \ A_3A_2	00	01	11	10
00	1	1	X	1
01	0	0	X	0
11	1	1	X	X
10	0	0	X	X

$Y_1 = A_1'A_0' + A_1A_0$

A_1 XNOR A_0

Y_0

A_1A_0 \ A_3A_2	00	01	11	10
00	1	1	X	1
01	0	0	X	0
11	0	0	X	X
10	1	1	X	X

$Y_0 = \cancel{A_1'A_0' + A_1A_0'}$

A_0'

VHDL code for converter without switching mechanism

```
entity converter1_alex is
  Port    (sw : in STD_LOGIC_VECTOR(3 downto 0);
           led : out STD_LOGIC_VECTOR(3 downto 0));
end converter1_alex;

architecture Behavioral of converter1_alex is
signal A2_N,A1_N,A0_N : STD_LOGIC;
signal A,Y : STD_LOGIC_VECTOR(3 downto 0);
begin
    A2_N <= NOT A(2);
    A1_N <= NOT A(1);
    A0_N <= NOT A(0);
    A <=sw; --set input to be the BCD value given by the 4 switches on Arty A7-35t
```

```
        Y(3) <= A(3) or (A(2) and A(1)) or (A(2) and A(0));
        Y(2) <= (A(2) and A1_N and A0_N) or (A2_N and A(0)) or (A2_N and A(1));
        Y(1) <= A(1) xnor A(0);
        Y(0) <= A0_N;

        led <= Y; --set LED pins on Arty A7-35t to be the Excess-3 BCD value
end Behavioral;
```

When implementing the second code converter with the switch to toggle between natural to Excess-3 BCD translation and Excess-3 to natural BCD, I implemented a behavioral model in VHDL. This has the benefit of being much easier to follow and have relatively simple logic statements when compared to the structural model used in the first code converter.

VHDL code for converter with switching mechanism

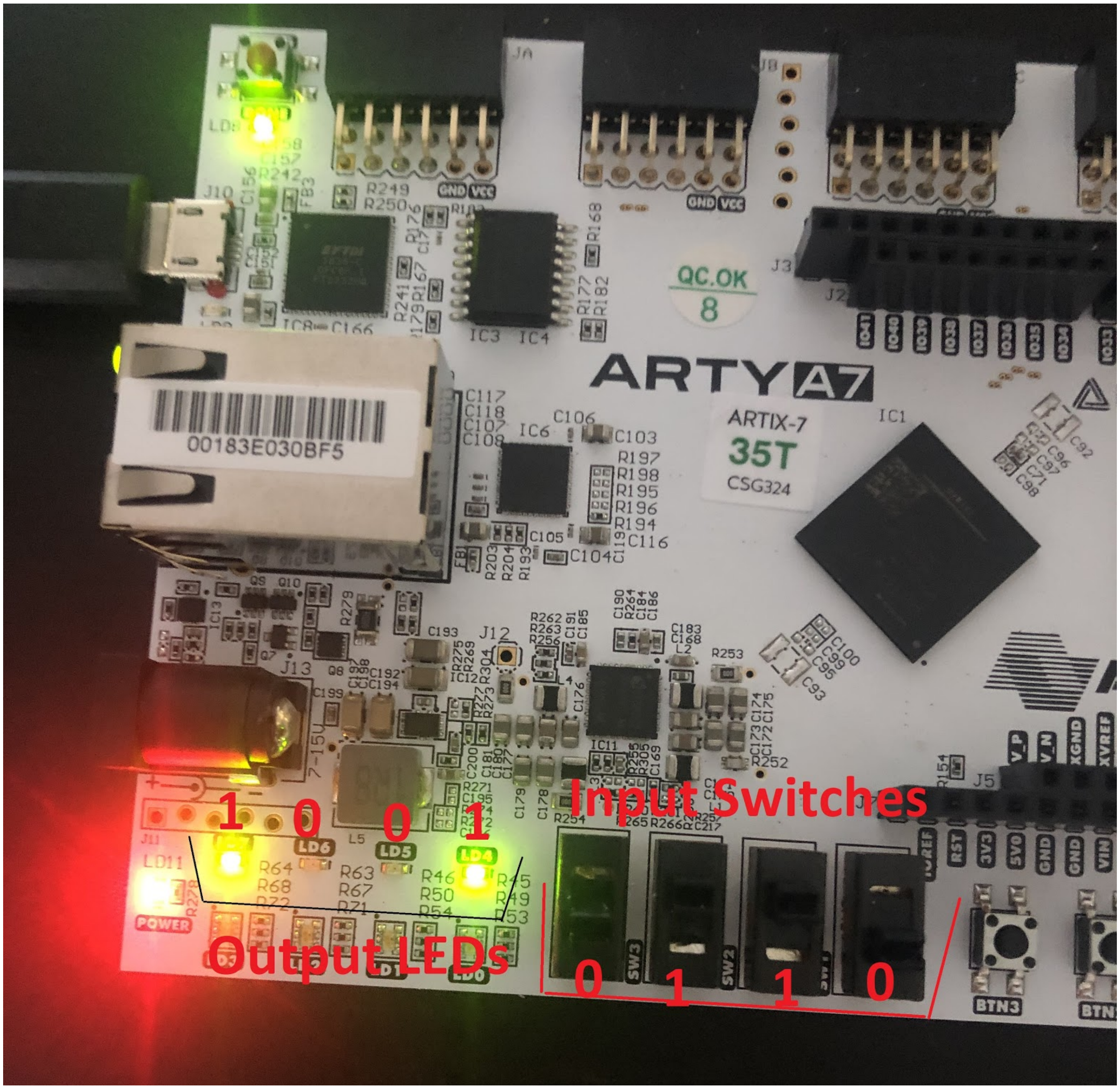
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity converter2_alex is
    Port ( sw : in STD_LOGIC_VECTOR (3 downto 0);
           led : out STD_LOGIC_VECTOR (3 downto 0);
           btn : in STD_LOGIC_VECTOR (3 downto 0)); -- btn(3) is used to switch between input BCD
format (natural or Excess-3)
end converter2_alex;

architecture Behavioral of converter2_alex is
signal A2_N,A1_N,A0_N : STD_LOGIC;
signal A,Y : STD_LOGIC_VECTOR(3 downto 0);
signal clk : std_logic :='1';
begin
        A <=sw; --set input to be the BCD value given by the 4 switches on Arty A7-35t
        process(btn) --run process every time btn is altered
        begin
        case btn(3) is
            when '0' => Y <= std_logic_vector(to_unsigned(to_integer(unsigned( A )) + 3, 4)); -- when
button not pressed, convert from natural to Excess-3 BCD
            when '1' => Y <= std_logic_vector(to_unsigned(to_integer(unsigned( A )) - 3, 4)); -- when btn
pressed, convert from Excess-3 to natural BCD
            when others => Y <= "0000";

        end case;
    end process;
    led <= Y; --set LED pins on Arty A7-35t to be the Excess-3 BCD value
end Behavioral;
```

In both of the above code converters, the design was realized using a Digilent Arty A7-35T FPGA utilizing the Xilinx Artix-7 architecture. This FPGA allowed me to build the design for each code converter in Vivado and then upload the design to the FPGA using a USB serial connection. For inputs, I used the four switches available on the FPGA to designate each individual bit, and for outputs I used the four embedded LEDs to represent each individual output bit. To set up the physical pins for the FPGA, I used the master template provided in the board files to design a constraints document. fThe below picture shows an implementation of the Natural BCD to Excess-3 BCD.



IV. Conclusion

To conclude, this laboratory should be considered a success. Students were successfully able to design code converters to translate natural BCD to Excess-3 BCD using VHDL in Xilinx Vivado. Furthermore, students were able to utilize a Xilinx FPGA to run their VHDL designs on physical hardware. These experiences in VHDL coding and FPGA implementation will be used in subsequent laboratories to design more complex digital logic circuits.

References

[1] Wikipedia contributors. (2020, June 14). Xilinx. In *Wikipedia, The Free Encyclopedia*. Retrieved 02:24, June 16, 2020, from https://en.wikipedia.org/w/index.php?title=Xilinx&oldid=962581047

[2] FPGA Lab Manuals - ECE 742