Experiment No. 1
Code Conversion
ECE 446

Peter Chinetti

September 9, 2014

Date Performed:	September 2, 2014
Instructor:	Professor Shanechi

1 Introduction

FPGAs have been taking over the Computer Engineering field as their ease of configuration and reconfiguration allows them to be an effective solution for many tasks. Although the 'field programmable' section of their name refers to their ability to be patched by customers, they are also widely used as a cheaper option than spinning a custom ASIC.

2 Background

2.1 BCD to Excess-3 Truth Table

Decimal	Natural BCD				Excess-3			
	A3	A2	A1	A0	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

2.2 Karnaugh Maps

Y3	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

Y2	00	01	11	10
00	0	1	1	1
01	1	0	0	0
11	X	X	X	X
10	0	1	X	X

Y1	00	01	11	10
00	1	0	1	0
01	1	0	1	0
11	X	X	X	X
10	1	0	X	X

Y0	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	X	X	X	X
10	1	0	X	X

A3	00	01	11	10
00	X	X	0	X
01	0	0	0	0
11	1	X	X	X
10	0	0	1	0

A2	00	01	11	10

00	X	X	0	X
01	0	0	1	0
11	0	X	X	X
10	1	1	0	1

A1	00	01	11	10
00	X	X	0	X
01	0	1	0	1
11	0	X	X	X
10	0	1	0	1

A0	00	01	11	10
00	X	X	0	X
01	1	0	0	1
11	1	X	X	X
10	1	0	0	1

2.3 Minimized Equations

$Y_3 = A_3 + A_2A_0 + A_1A_2$
$Y_2 = A_2\overline{A_1A_0} + \overline{A_2}A_0 + \overline{A_2}A_1$
$Y_1 = \overline{A_1A_0} + A_1A_0$
$Y_0 = \overline{A_0}$
$A_3 = Y_3Y_2 + Y_3Y_1Y_0$
$A_2 = \overline{Y_2Y_1} + Y_2Y_1Y_0 + Y_3\overline{Y_2}Y_1$
$A_1 = \overline{Y_1}Y_0 + Y_1\overline{Y_0}$
$A_0 = \overline{Y_0}$

3 Procedure

a. Generate minimized equations for converter.

b. Write VHDL to implement logic.

c. Assign pins to ports

d. Simulate

e. Program and Test

4 Equipment

- PC
- Spartan-3E development board

5 Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity BCD_to_Excess3 is
    Port ( input : in  STD_LOGIC_VECTOR (3 downto 0);
        SEL : in STD_LOGIC;
           output : out  STD_LOGIC_VECTOR (3 downto 0));
end BCD_to_Excess3;

architecture Universal_Converter of BCD_to_Excess3 is
begin

process(SEL)
begin
  if SEL='0' then
    output(0) <= not input(0);
    output(1) <= input(1) xnor input(0);
    output(2) <= (input(2) and (not input(1)) and (not input(0))) or ((not input(2)) and (input(1) or input(0)));
    output(3) <= input(3) or ( input(0) and input(2) ) or (input(1) and input(2) );
  else
    output(0) <= not input(0);
    output(1) <= (input(1) and (not input(0))) or (input(0) and (not input(1)));
    output(2) <= ((not input(2)) and (not input(1))) or (input(1) and ((input(2) and input(0)) or (input(3) and (not input(0)))));
    output(3) <= (input(3) and input(2)) or (input(3) and input(1) and input(0));
  end if;
end process;
end Universal_Converter;
```

BCD_Converter/short.vhd

6 Conclusions

The purpose of this lab was achieved. A code converter was built and tested. The basics of working with VHDL and FPGAs were taught.