Fundamentals of Computer Engineering Laboratory
ECE340

GAL Programming with VHDL
Report for Week # 3

By: Andrew Miller

Instructor's Name: Filip Cuckov

Section: Tuesday 9:30 am to 12:10 pm
Date: 9/30/2008

Honor Pledge:

"I pledge to support the Honor System of Old Dominion University. I will refrain from any form of academic dishonesty or deception, such as cheating or plagiarism. I am aware that as a member of the academic community, it is my responsibility to turn in all suspected violators of the Honor System. I will report to an Honor Council hearing if summoned."

Signature: _____

Introduction:

In this lab, techniques to implement programmable logic with VHDL to specify functionality will be used. A complex combinational circuit that includes VHDL will be used to design a 4-bit binary code converter to drive a seven-segment display.

Preliminary Work:

To begin the design for this lab, a truth table is necessary to determine all of the output functions for the seven-segment display. The truth table will show hex numbers 0-9 and A-F. Four inputs on the GAL chip will be needed to represent all of the possible outputs for the display. In order to not burn out of LEDs for the seven-segment display, the truth table of the functions will be inverted so that 0s are outputted. Those 0s will then be fed through an inverter and then on to the display.

Regular Output Table

Hex	Inputs				Outputs						
	A	B	C	D	Ya	Yb	Yc	Yd	Ye	Yf	Yg
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	0	0	0	1	1	0
2	0	0	1	0	1	0	1	1	0	1	1
3	0	0	1	1	1	0	0	1	1	1	1
4	0	1	0	0	0	1	0	0	1	1	1
5	0	1	0	1	1	1	0	1	1	0	1
6	0	1	1	0	1	1	1	1	1	0	1
7	0	1	1	1	1	0	0	0	1	1	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	0	0	1	1	1
A	1	0	1	0	1	1	1	0	1	1	1
B	1	0	1	1	0	1	1	1	1	0	1
C	1	1	0	0	1	0	0	1	1	1	0
D	1	1	0	1	0	0	1	1	1	1	1
E	1	1	1	0	1	1	1	1	0	0	1
F	1	1	1	1	1	1	1	0	0	0	1

Inverted Output Table

Hex	Inputs				Outputs						
	A	B	C	D	Ya	Yb	Yc	Yd	Ye	Yf	Yg
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	1	1	1	0	0	1
2	0	0	1	0	0	1	0	0	1	0	0
3	0	0	1	1	0	1	1	0	0	0	0
4	0	1	0	0	1	0	1	1	0	0	0
5	0	1	0	1	0	0	1	0	0	1	0
6	0	1	1	0	0	0	0	0	0	1	0
7	0	1	1	1	0	1	1	1	0	0	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	1	1	0	0	0
A	1	0	1	0	0	0	0	1	0	0	0
B	1	0	1	1	1	0	0	0	0	1	0
C	1	1	0	0	0	1	1	0	0	0	1
D	1	1	0	1	1	1	0	0	0	0	0
E	1	1	1	0	0	0	0	0	1	1	0
F	1	1	1	1	0	0	0	1	1	1	0

Now that the truth table is filled out, the output functions for eagle of the LEDs on the seven-segment display can be determined.

Ya = A’B’C’D + A’BC’D’ + AB’CD + ABC’D
Yb = A’B’C’D + A’B’CD’ + A’B’CD + A’BCD + ABC’D’ + ABC’D
Yc = A’B’C’D + A’B’CD + A’BC’D’ + A’BC’D + A’BCD + AB’C’D + ABC’D’
Yd = A’B’C’D + A’BC’D’ + A’BCD + AB’C’D + AB’CD’ + ABCD
Ye = A’B’CD’ + ABCD’ + ABCD
Yf = A’BC’D + A’BCD’ + AB’CD + ABCD’ + ABCD
Yg = A’B’C’D’ + A’B’C’D + A’BCD + ABC’D’

The seven output functions can now be minimized with K-maps.

Ya

CD \ AB	00	01	11	10
00	0	1	0	0
01	1	0	1	0
11	0	0	0	1
10	0	0	0	0

A’B’C’D + A’BC’D’ + AB’CD + ABC’D

Yb

CD \ AB	00	01	11	10
00	0	0	1	0
01	1	0	1	0
11	1	1	0	0
10	1	0	0	0

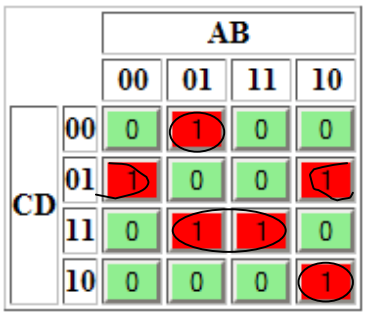
A’B’D + A’B’C + ABC’ + A’CD

Yc

CD \ AB	00	01	11	10
00	0	1	1	0
01	1	1	0	1
11	1	1	0	0
10	0	0	0	0

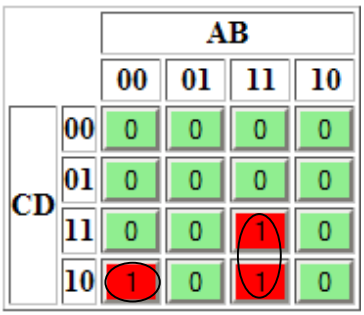
A’D + A’BC’ + BC’D’ + B’C’D

Yd



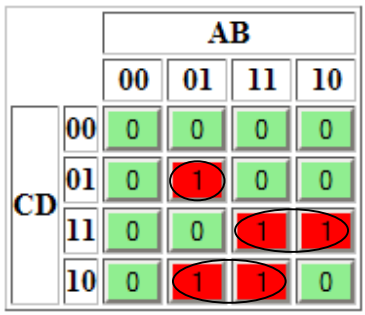
BCD + B'C'D + A'BC'D' + AB'CD'

Ye



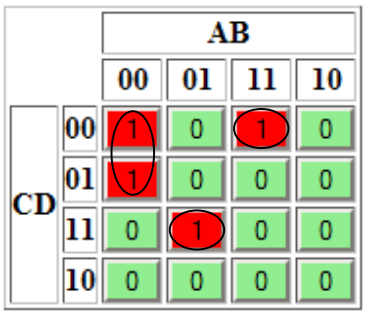
A'B'CD' + ABC

Yf



BCD' + ACD + A'BC'D

Yg



A'B'C' + A'BCD + ABC'D'

Now that the simplified expressions for the output functions are made, they can be put into VHDL code and burned onto the GAL chip. The VHDL expressions are below.

Ya = ((not A) and (not B) and (not C) and D) or ((not A) and B and (not C) and (not D)) or (A and B and (not C) and D) or (A and (not B) and C and D)

Yb = ((not A) and (not B) and D) or ((not A) and (not B) and C) or (A and B and (not C)) or ((not A) and C and D)

Yc = ((not a) and D) or ((not A) and B and (not C)) or (B and (not C) and (not D)) or ((not B) and (not C) and D)

Yd = (B and C and D) or ((not B) and (not C) and D) or ((not A) and B and (not C) and (not D)) or (A and (not B) and C and (not D))

Ye = (A and B and C) or ((not A) and (not B) and C and (not D))

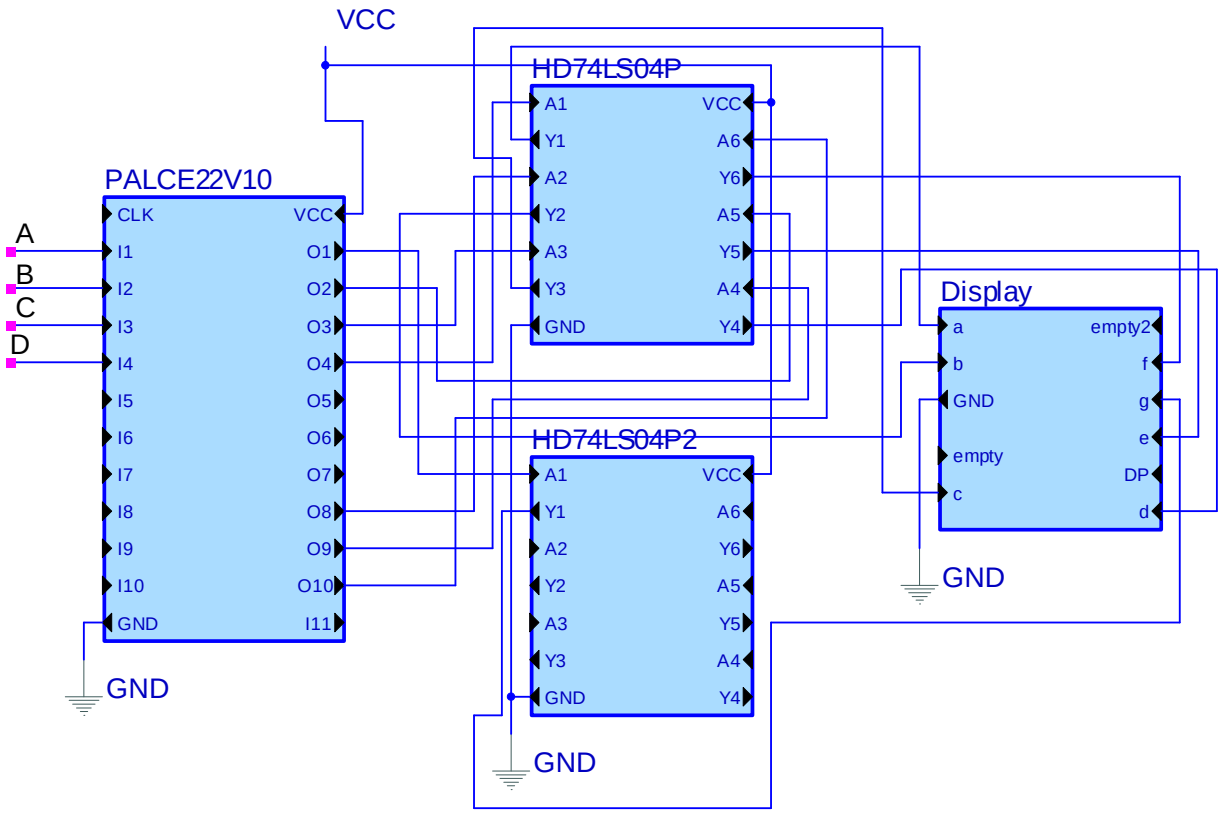
Yf = (B and C and (not D)) or (A and C and D) or ((not A) and B and (not C) and D)

Yg = ((not A) and (not B) and (not C)) or ((not A) and B and C and D) or (A and B and (not C) and (not D))

After burning the functions onto the GAL chip, a wire list can now be created for all of the functions.

U1 – PALCE22V10	SW2 -> B -> U1-2	U2-4 -> U4-2
U2 – HD74LS04P	SW3 -> C -> U1-3	U2-6 -> U4-5
U3 – HD74LS04P	U3-2 -> U4-9	U2-8 -> U4-6
U4 – Display	U1-20 -> Ya -> U2-1	U2-10 -> U4-8
U1-24 – VCC	U1-16 -> Yb -> U2-3	U2-12 -> U4-10
U1-12 – GND	U1-21 -> Yc -> U2-5	
U2-3 – GND	U1-15 -> Yd - >U2-9	
U4-14 – VCC	U1-22 -> Ye -> U2-11	
U4-7 – GND	U1-14 -> Yf -> U2-13	
SW1 -> A -> U1-1	U1-23 -> Yg -> U3-1	
SW4 -> D -> U1-4	U2-2 -> U4-1	

Now that the wire list is made the circuit can be wired up. Following is the chip diagram for the circuit.



Lab Results:

After wiring up the above circuit and turning on the power the seven-segment display worked as expected. Hex values 0-9 display as 0-9 on the display. Hex values A-F display as lowercase letters on the display.

Observations and Conclusions:

The seven-segment display worked as expected. The circuit was designed so that resistors did not need to be used in order to not burn up the LEDs in the display unit.