Lab 08
Code Converters

ECE 380-002
University of Alabama

Yichen Huang
Thomas Dillman

2019/10/28

Introduction

One of the critical factors concerning engineers designing embedded systems is the design of the efficient interfaces to provide information to users. One of the most simple and useful methods for displaying numerical information is the 7-segement LED display. Many embedded systems incorporating clocks or timers display hours, minute, and/or seconds in this way. However, microprocessors normally request numbers internally in a binary format and 7-segemant displays are typically used to display decimal or hexadecimal digits externally, digital circuit must be developed to translate from the internal representations. In this lab, we use code converts to drive LED display.

Procedure

A. Prelab

a. Design A

In the design A, we write VHDL code for a BCD-to-seven segment LED display converter with 4 input which are representing a single decimal digit and a seven-bit output which is suitable for driving a seven segment LED display on the Altera DE1 board. Finally, we compile the file.

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
ENTITY designA IS
PORT ( bcd: IN STD_LOGIC_VECTOR(3 DOWNTO 0) ;
leds: OUT STD_LOGIC_VECTOR(0 TO 6) ) ;
END designA ;
ARCHITECTURE Behavior OF designA IS
BEGIN
PROCESS ( bcd )
BEGIN
CASE bcd IS -- abcdefg
WHEN "0000" => leds <= "0000001";
WHEN "0001" => leds <= "1001111";
WHEN "0010" => leds <= "0010010";
WHEN "0011" => leds <= "0000110";
WHEN "0100" => leds <= "1001100";
WHEN "0101" => leds <= "0100100";
WHEN "0110" => leds <= "0100000";
WHEN "0111" => leds <= "0001111";
WHEN "1000" => leds <= "0000000";
WHEN "1001" => leds <= "0001100";
WHEN "1010" => leds <= "0000001";
WHEN "1011" => leds <= "0000001";
WHEN "1100" => leds <= "0000001";
WHEN "1101" => leds <= "0000001";
WHEN "1110" => leds <= "0000001";
WHEN "1111" => leds <= "0000001";
WHEN OTHERS => leds <= "-------" ;
END CASE ;
END PROCESS ;
END Behavior;
```

b. Design B

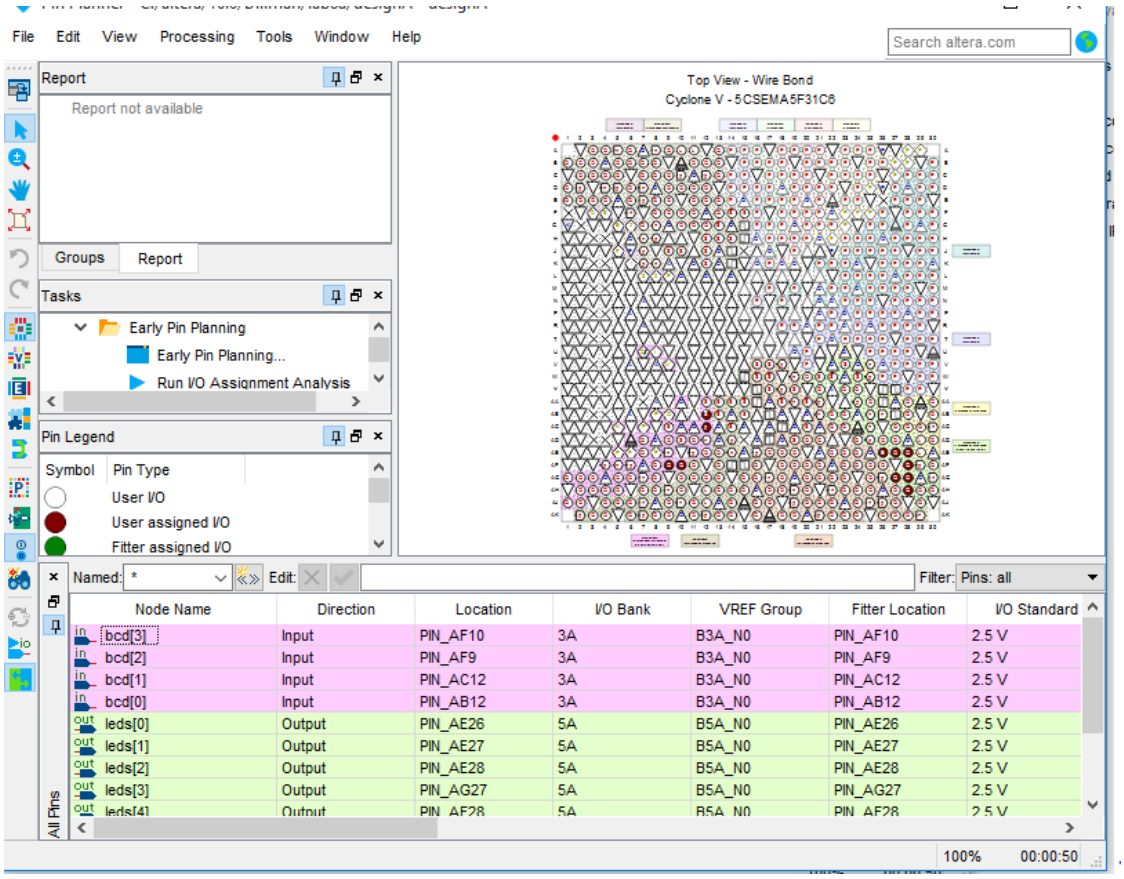
In the design B, we use two 7-segment LEDs to display a 4-bit signed number in 2's complement representation. We achieve this feature mainly through modify the previous design. We add one more digit in the output for the sign.

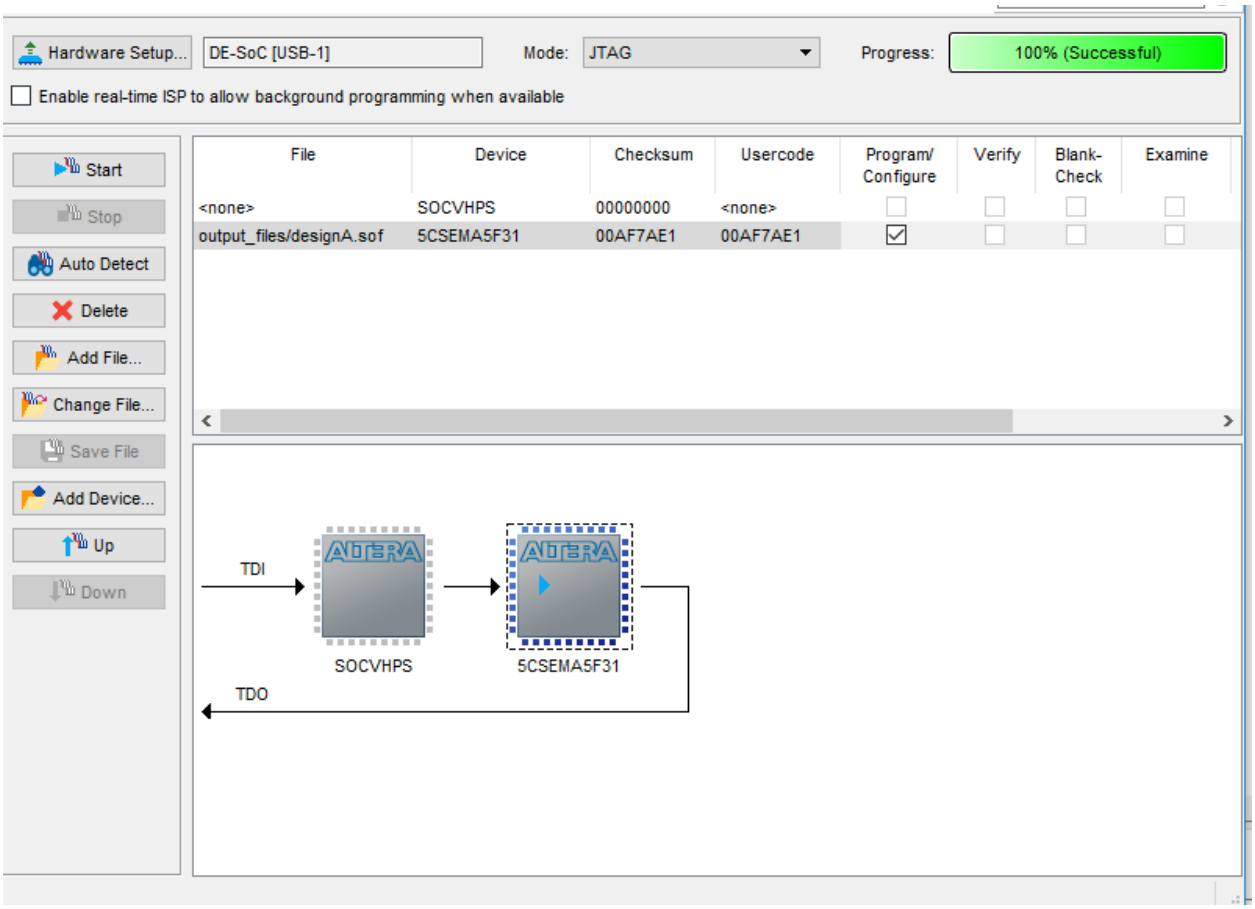
```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
ENTITY designA IS
PORT ( bcd: IN STD_LOGIC_VECTOR(3 DOWNTO 0) ;
leds: OUT STD_LOGIC_VECTOR(0 TO 7) ) ;
END designA ;
ARCHITECTURE Behavior OF designA IS
BEGIN
PROCESS ( bcd )
BEGIN
CASE bcd IS -- abcdefg
WHEN "0000" => leds <= "00000011";
WHEN "0001" => leds <= "10011111";
WHEN "0010" => leds <= "00100101";
WHEN "0011" => leds <= "00001101";
WHEN "0100" => leds <= "10011001";
WHEN "0101" => leds <= "01001001";
WHEN "0110" => leds <= "01000001";
WHEN "0111" => leds <= "00011111";
WHEN "1000" => leds <= "00000000";
WHEN "1001" => leds <= "00011110";
WHEN "1010" => leds <= "01000000";
WHEN "1011" => leds <= "01001000";
WHEN "1100" => leds <= "10011000";
WHEN "1101" => leds <= "00001100";
WHEN "1110" => leds <= "00100100";
WHEN "1111" => leds <= "10011110";
WHEN OTHERS => leds <= "--------";
END CASE ;
END PROCESS ;
END Behavior;
```

B. During the lab

a. Design A

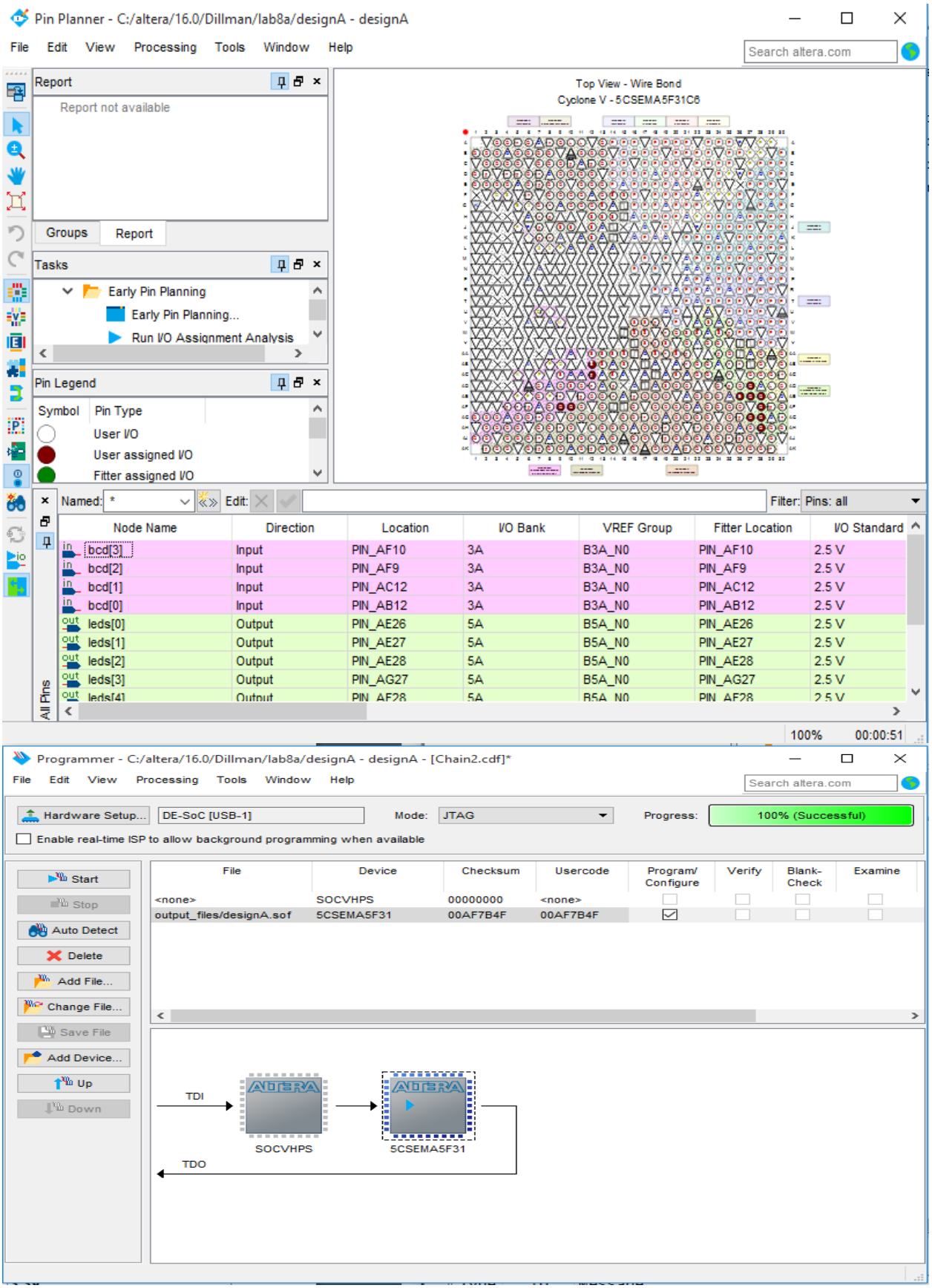
During the lab, we assign the pin of the input and output. Then we compiled the file again. Finally, we set the board. When we test the board the board change binary to decimal successfully.





b. Design B

During the lab of design B, we repeat the same step in the design A. After we set the board, we test the number through -8 to 7. The board display the decimal number successfully.

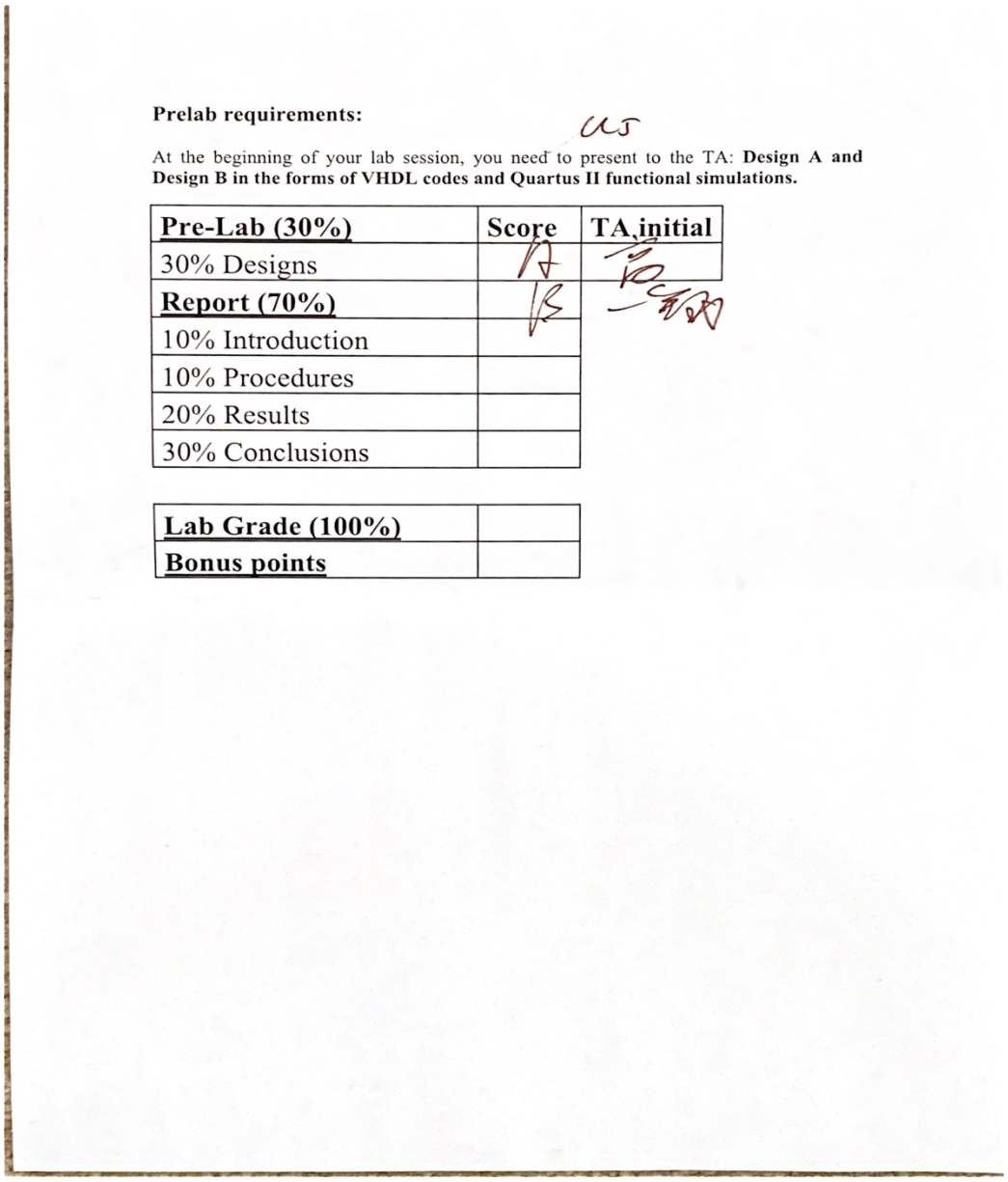

Pin Planner - C:/altera/16.0/Dillman/lab8a/designA - designA
File Edit View Processing Tools Window Help
Search altera.com
Report
Report not available
Groups Report
Tasks
Early Pin Planning
Early Pin Planning...
Run I/O Assignment Analysis
Pin Legend
Symbol Pin Type
User I/O
User assigned I/O
Fitter assigned I/O
Top View - Wire Bond
Cyclone V - 5CSEMA5F31C6
Named: * Edit: Filter: Pins: all
Node Name Direction Location I/O Bank VREF Group Fitter Location I/O Standard
bcd[3] Input PIN_AF10 3A B3A_N0 PIN_AF10 2.5 V
bcd[2] Input PIN_AF9 3A B3A_N0 PIN_AF9 2.5 V
bcd[1] Input PIN_AC12 3A B3A_N0 PIN_AC12 2.5 V
bcd[0] Input PIN_AB12 3A B3A_N0 PIN_AB12 2.5 V
leds[0] Output PIN_AE26 5A B5A_N0 PIN_AE26 2.5 V
leds[1] Output PIN_AE27 5A B5A_N0 PIN_AE27 2.5 V
leds[2] Output PIN_AE28 5A B5A_N0 PIN_AE28 2.5 V
leds[3] Output PIN_AG27 5A B5A_N0 PIN_AG27 2.5 V
All Pins
100% 00:00:51
Programmer - C:/altera/16.0/Dillman/lab8a/designA - designA - [Chain2.cdf]*
File Edit View Processing Tools Window Help
Search altera.com
Hardware Setup... DE-SoC [USB-1] Mode: JTAG Progress: 100% (Successful)
Enable real-time ISP to allow background programming when available
Start
Stop
Auto Detect
Delete
Add File...
Change File...
Save File
Add Device...
Up
Down
File Device Checksum Usercode Program/ Configure Verify Blank-Check Examine
<none> SOCVHPS 00000000 <none>
output_files/designA.sof 5CSEMA5F31 00AF7B4F 00AF7B4F
TDI
SOCVHPS
5CSEMA5F31
TDO

Result

The altera DE1 board display design A and B successfully.

Conclusion

In this lab, we understand how to display decimal number and use 2's complement to display negative number.



Prelab requirements:

At the beginning of your lab session, you need to present to the TA: **Design A and Design B in the forms of VHDL codes and Quartus II functional simulations.**

Pre-Lab (30%)	**Score**	**TA initial**
30% Designs		
Report (70%)		
10% Introduction		
10% Procedures		
20% Results		
30% Conclusions		

Lab Grade (100%)	
Bonus points	