EELE 367 - Introduction to Logic Circuits

Lab 8(a) - 7-Segment Display Decoder in VHDL using Processes

Objective

The objective of this lab is to introduce the lab hardware for this course, gain experience using the Quartus Prime development environment, and start using VHDL processes to model logic circuits. In this lab you will be designing a 7-segment decoder that will drive a character display on the DE0-CV FPGA board.

Outcomes

After completing this lab you should be able to:

- Use the Quartus development tools to implement a digital design on the DE0-CV FPGA board.
- Create a 4-input, 7-segment display decoder (0_{16} to F_{16}) on an FPGA using a VHDL process and conditional programming constructs.

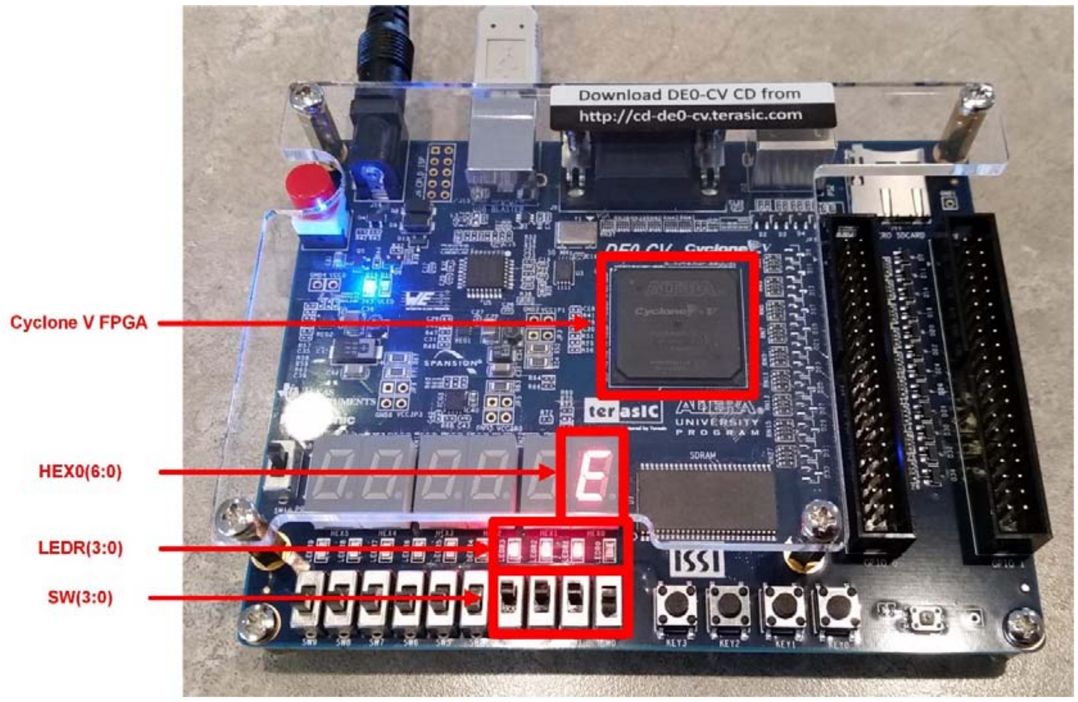
Deliverables

The deliverable(s) for this lab are as follows:

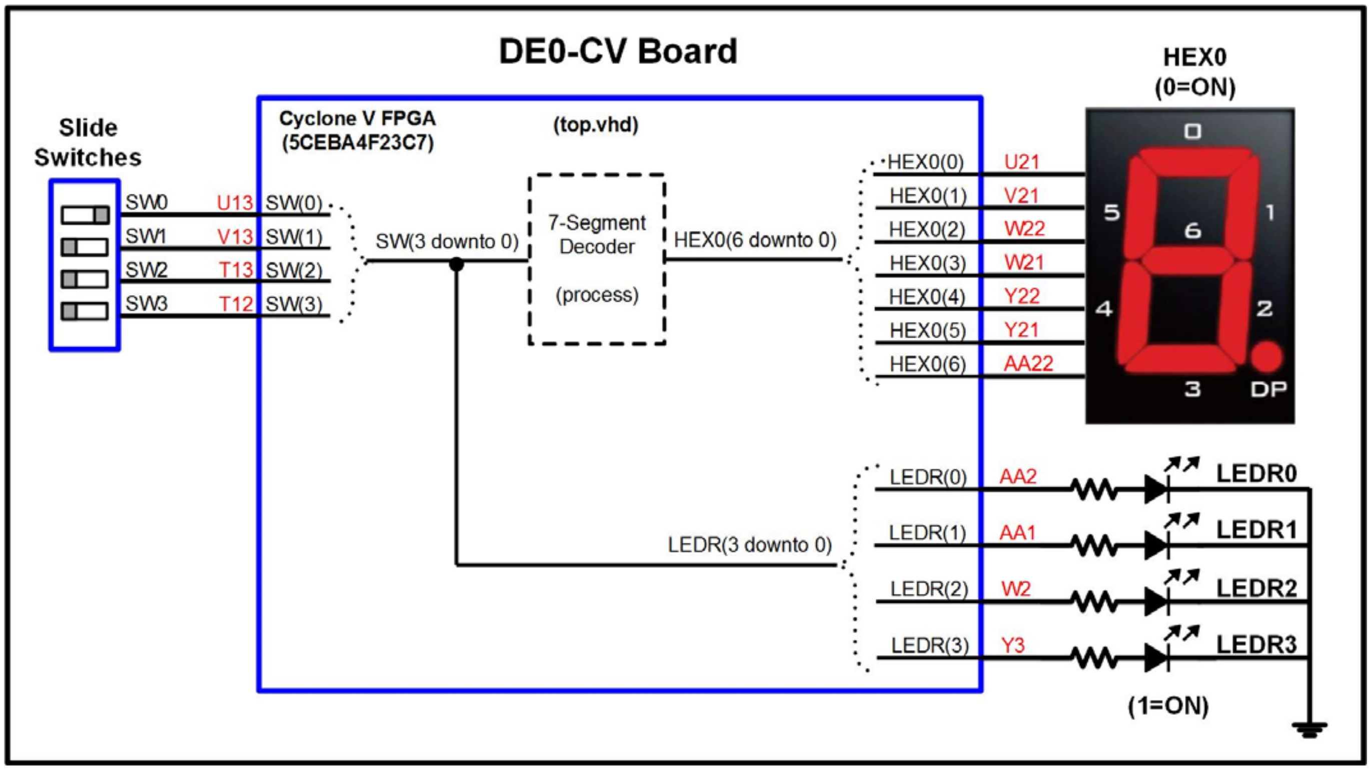
- Demonstrate a 4-input, 7-segment display decoder on the DE0-CV FPGA board implemented with a VHDL process (90%).
- Upload your top.vhd file to the lab DropBox (10%).

Lab Work & Demonstration

You are going to design a 7-segment display decoder in VHDL using a process and a conditional programming construct. The decoder will take in the values from the slider switches on the DE0-CV board (SW3, SW2, SW1, and SW0) and display the corresponding HEX characters (0_{16} to F_{16}) on one of the 7-segment displays (HEX0). You will also drive the values of the slider switches on the red LEDs on the DE0-CV (LEDR). The following figure shows the I/O on the DE0-CV that you will be using.



The following figure shows the block diagram for the system you will design. The pin numbers for each of the I/O used in this lab are shown in red. Note that you will be creating your VHDL ports as vectors in order to simplify your model.



Slider Switches - The slider switches on the DE0-CV board are labeled SW3, SW2, SW1, and SW0. You will create an input port called SW and give it a vector type of width 4 (i.e., std_logic_vector(3 downto 0)). When you assign the FPGA pins to this vector, you will address each switch using its index. For example, the SW0 switch on the FPGA board comes into the FPGA on pin U13. When assigning the pin in Quartus, this will be connected to the signal SW(0). These four switches will be used as the 4-bit input into the 7-segment decoder.

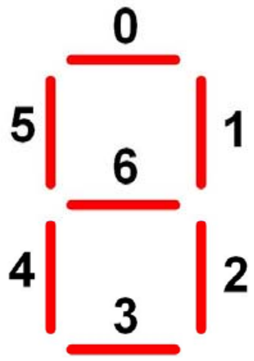
Red LEDs - The red LEDs on the FPGA board are turned on when driven with a logic 1. These LEDs on the DE0-CV board are labeled LEDR3, LEDR2, LEDR1, and LEDR0. You will create an output port called LEDR and give it a vector type of width 4. You will drive the LEDR vector with the SW input in your VHDL design. This will display the values of the slider switches on the LEDs.

7-Segment Character Display – The character displays on the DE0-CV board are active LOW. This means to turn on a segment, you need to drive a 0 to its pin. In this lab you will be displaying characters on the HEX0 display. You will create an output port called HEX0 and give it a vector type of width 7. Make note of the signal mapping in the above picture. You will be using bit position "0" to drive the top segment of the display. This represents the least significant position within the HEX0 vector. To turn it on, you need to drive a LOW to this position of the vector.

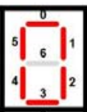
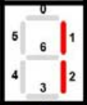
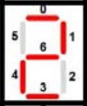
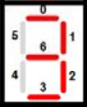
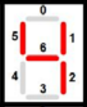
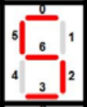
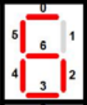
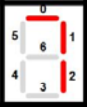
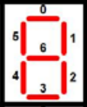
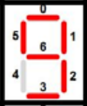
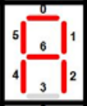
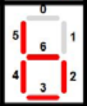
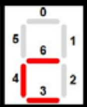
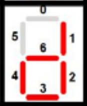
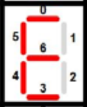
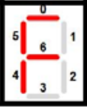
Part 1 – Design the Logic for the 4-Input 7-Segment Decoder

Before you can model the decoder in VHDL, you need to design the functionality you wish to model. You will need to come up with the values to drive to the character display in order to generate the appropriate characters for each input code. The following table can be used to tabulate the desired values. You will code these values into your process for the decoder. Keep in mind that to turn on a display, you need to drive a logic 0. You do not need to show this table to the lab instructor.

7-Segment Display



Common Anode
0=ON, 1=OFF

SW(3 downto 0)					HEX0(6 downto 0)						
0	0	0	0		1	0	0	0	0	0	0
0	0	0	1								
0	0	1	0								
0	0	1	1								
0	1	0	0								
0	1	0	1								
0	1	1	0								
0	1	1	1								
1	0	0	0								
1	0	0	1								
1	0	1	0								
1	0	1	1								
1	1	0	0								
1	1	0	1								
1	1	1	0								
1	1	1	1								

Part 2 – Implement the Decoder Functionality on the FPGA Using Quartus

You are now going to create a Quartus project to implement the decoder. Keeping your files organized is VERY IMPORTANT. For each project you create, you will need to manually create the folder that it will go into. You will use the name of the folder as the primary way to identify your project. Your primary VHDL file within this folder will always be called "top.vhd". It is to your advantage to stay organized because for future labs, you can simply copy/paste this project in Quartus and avoid duplicating effort.

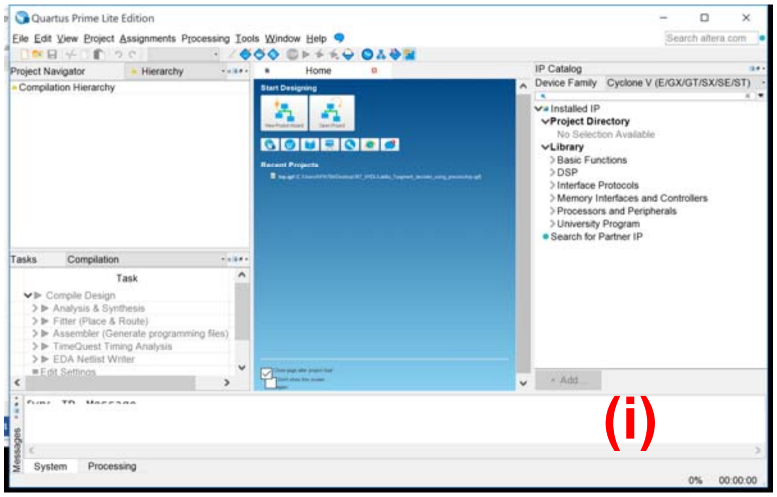
A) Log into a computer containing the Quartus software (if on campus, use your MSU domain login). You will want to create your folder directly on the C: drive of the computer as opposed to on a thumb drive or network drive. This will allow the synthesis operation to run much faster. After each lab, you should copy over your files to either your network drive or your thumb drive.

B) Create a folder for today's lab. Consider making a folder on the desktop of the computer you are working on called "EELE367". Then under it, make a folder called "Lab8a_7segment_decoder_using_process". This folder will be where Quartus will put all of your project files.

C) Launch the Altera Quartus design tool.

   - Start – All Programs – Altera xxx Lite Edition – Quartus Prime

   The following window (i) will appear (it might take a minute).

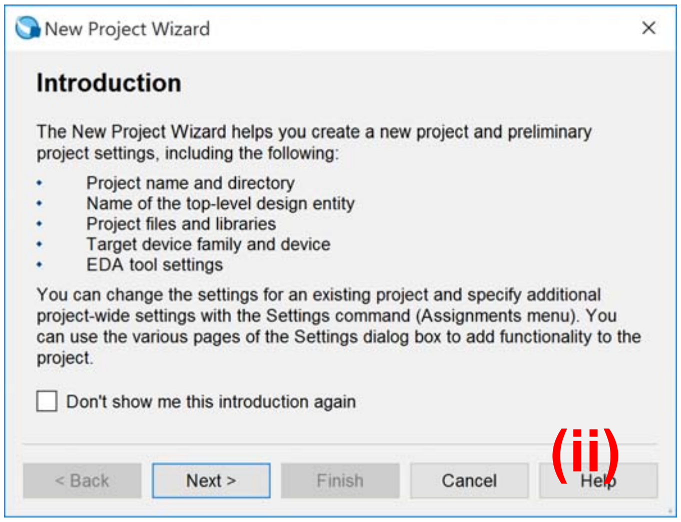


D) Create a new project using the "Project Wizard"

   File – New Project Wizard

   The following window (ii) will appear. Click "Next"

   In the next window (iii), browse to the folder you created and choose "Select Folder". Enter "top" as the name of the project and top-level design entity. (NOTE: it is important to name the project **top**). Click "Next"


New Project Wizard

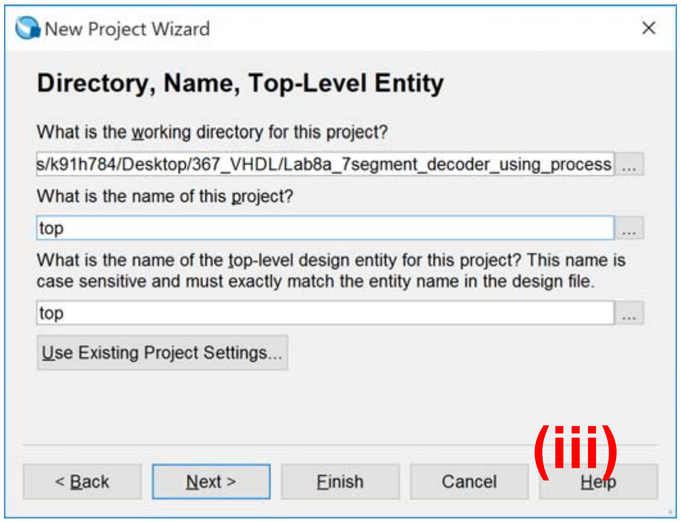
Introduction

The New Project Wizard helps you create a new project and preliminary project settings, including the following:

- Project name and directory
- Name of the top-level design entity
- Project files and libraries
- Target device family and device
- EDA tool settings

You can change the settings for an existing project and specify additional project-wide settings with the Settings command (Assignments menu). You can use the various pages of the Settings dialog box to add functionality to the project.

☐ Don't show me this introduction again


New Project Wizard

Directory, Name, Top-Level Entity

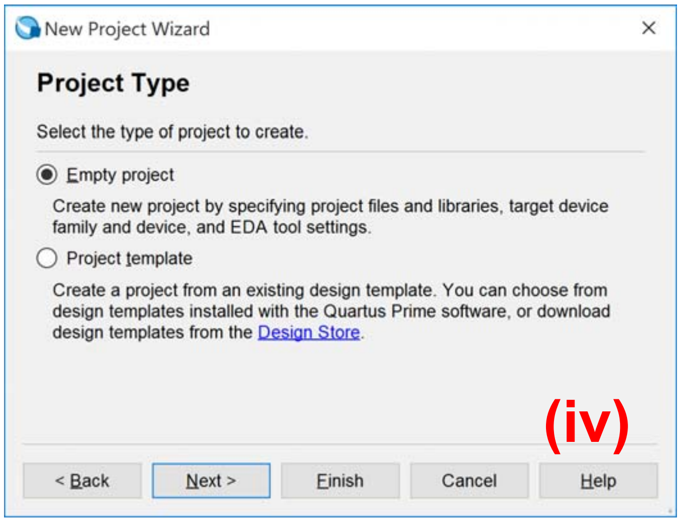
What is the working directory for this project?
s/k91h784/Desktop/367_VHDL/Lab8a_7segment_decoder_using_process

What is the name of this project?
top

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.
top

The next screen (iv) allows you to use a template. We will not be using a template so select “Empty Project”. Click “Next”

The next screen (v) allows you to add existing design files. We will be creating our own files so leave this screen blank. Click “Next”

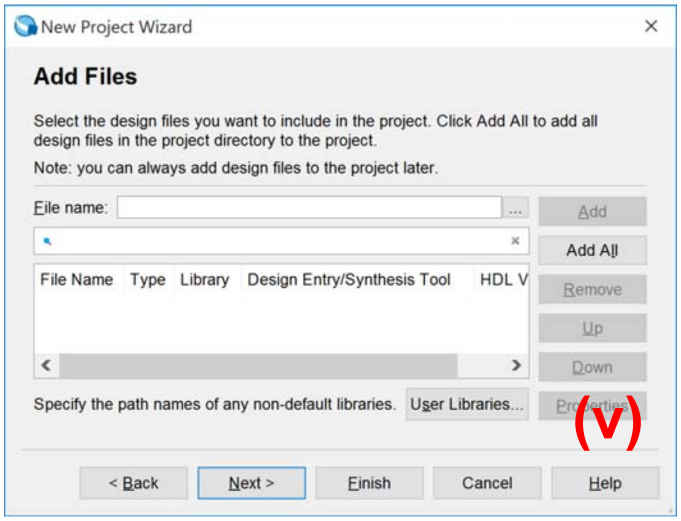

New Project Wizard

Project Type

Select the type of project to create.

● Empty project
Create new project by specifying project files and libraries, target device family and device, and EDA tool settings.

○ Project template
Create a project from an existing design template. You can choose from design templates installed with the Quartus Prime software, or download design templates from the Design Store.


New Project Wizard

Add Files

Select the design files you want to include in the project. Click Add All to add all design files in the project directory to the project.
Note: you can always add design files to the project later.

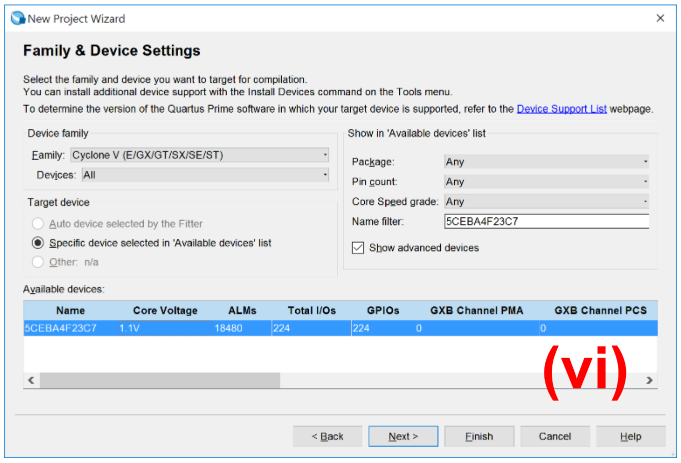
File name:

File Name	Type	Library	Design Entry/Synthesis Tool	HDL V

Specify the path names of any non-default libraries.

In the next window (vi), you will choose the FPGA device that will be targeted in this project. The FPGA that is on the DE0-CV board is a Cyclone V 5CEBA4F23C7. One of the easiest ways to select the correct device is to start typing in the “Name Filter” field on the right. This will start removing devices. As you type, you will eventually get down to only one FPGA device. Highlight the 5CEBA4F23C7. Click “Next”.

The next window (vii) allows you to specify other design tools you will be using. We will only be developing within the Quartus environment, so leave everything in this screen as is, and click “Next”.



Family & Device Settings

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.
To determine the version of the Quartus Prime software in which your target device is supported, refer to the Device Support List webpage.

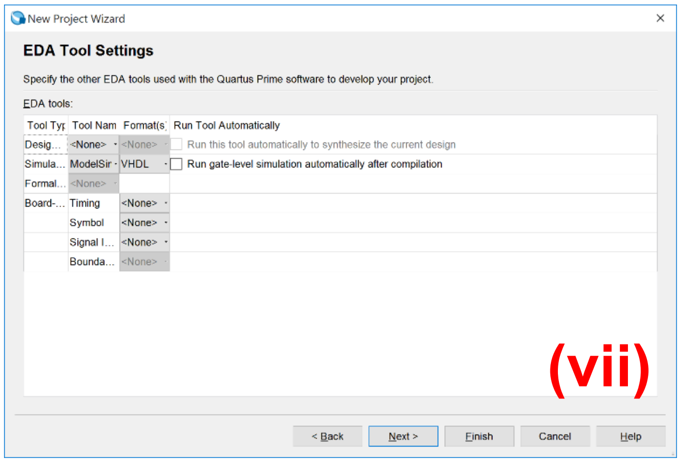
Device family
Family: Cyclone V (E/GX/GT/SX/SE/ST)
Devices: All

Target device
Auto device selected by the Fitter
[✓ Specific device selected in 'Available devices' list]
Other: n/a

Show in 'Available devices' list
Package: Any
Pin count: Any
Core Speed grade: Any
Name filter: 5CEBA4F23C7
[✓] Show advanced devices

Available devices:

Name	Core Voltage	ALMs	Total I/Os	GPIOs	GXB Channel PMA	GXB Channel PCS
5CEBA4F23C7	1.1V	18480	224	224	0	0



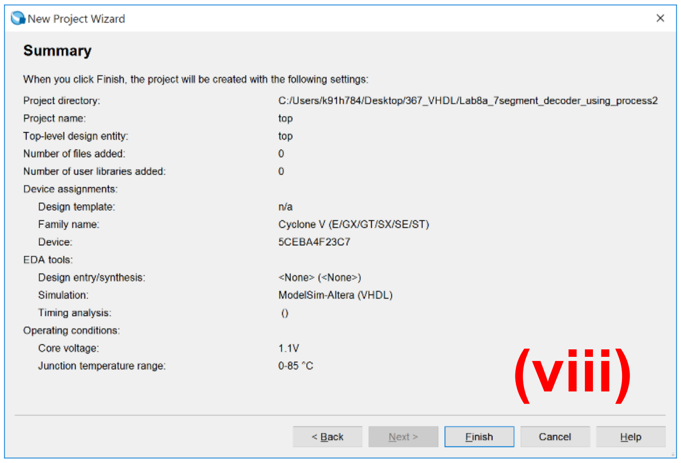
EDA Tool Settings

Specify the other EDA tools used with the Quartus Prime software to develop your project.

EDA tools:

Tool Typ	Tool Nam	Format(s	Run Tool Automatically
Desig...	<None>	<None>	Run this tool automatically to synthesize the current design
Simula...	ModelSir	VHDL	Run gate-level simulation automatically after compilation
Formal...	<None>		
Board-...	Timing	<None>	
	Symbol	<None>	
	Signal I...	<None>	
	Bounda...	<None>	

The final screen is a summary of the settings you have selected. Click “Finish”.



Summary

When you click Finish, the project will be created with the following settings:

Project directory:	C:/Users/k91h784/Desktop/367_VHDL/Lab8a_7segment_decoder_using_process2
Project name:	top
Top-level design entity:	top
Number of files added:	0
Number of user libraries added:	0
Device assignments:	
Design template:	n/a
Family name:	Cyclone V (E/GX/GT/SX/SE/ST)
Device:	5CEBA4F23C7
EDA tools:	
Design entry/synthesis:	<None> (<None>)
Simulation:	ModelSim-Altera (VHDL)
Timing analysis:	()
Operating conditions:	
Core voltage:	1.1V
Junction temperature range:	0-85 °C

E) Create a new VHDL design file called “top.vhd”.

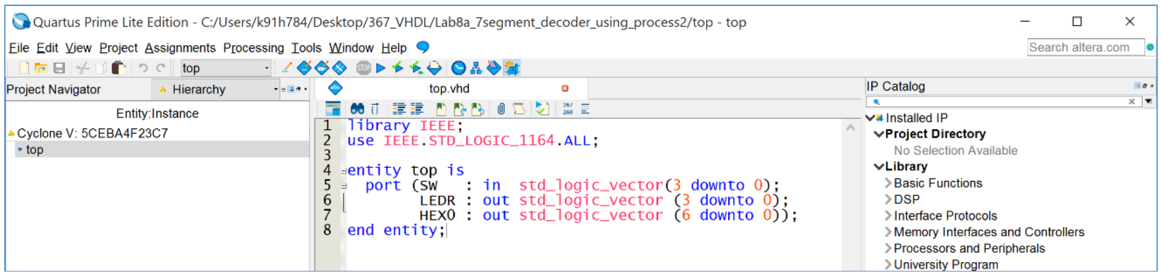
File – New – VHDL File – “OK”. A blank text file will appear. We need to first save this as top.vhd

File – Save As. The file name should default to “top.vhd”, verify it is named correctly and click “Save”.

You will enter all of the VHDL for today’s design in this top.vhd file.

F) Enter the VHDL entity for the system.

You should use the std_logic_vector data types for this lab. This data type requires including the STD_LOGIC_1164 package. You should always call your entity **top**. The way you should define your entity for this lab is given in the next figure.

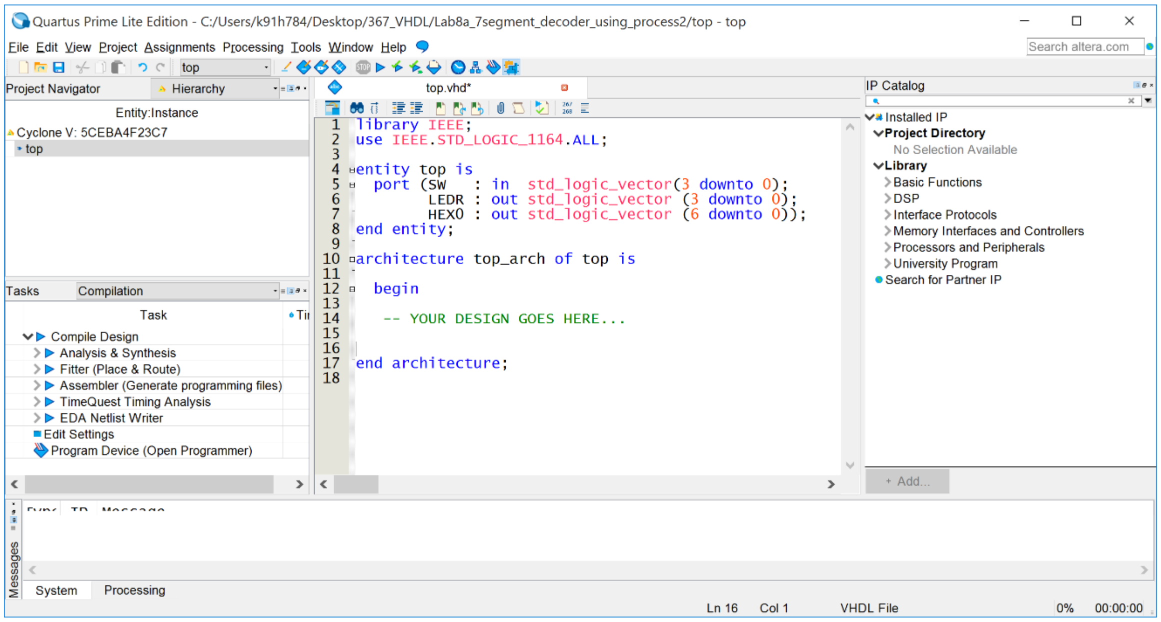

Quartus Prime Lite Edition - C:/Users/k91h784/Desktop/367_VHDL/Lab8a_7segment_decoder_using_process2/top - top
File Edit View Project Assignments Processing Tools Window Help
Project Navigator
Hierarchy
Entity:Instance
Cyclone V: 5CEBA4F23C7
top
top.vhd
IP Catalog
Installed IP
Project Directory
No Selection Available
Library
Basic Functions
DSP
Interface Protocols
Memory Interfaces and Controllers
Processors and Peripherals
University Program

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

entity top is
  port (SW    : in  std_logic_vector(3 downto 0);
        LEDR  : out std_logic_vector (3 downto 0);
        HEX0  : out std_logic_vector (6 downto 0));
end entity;
```

G) Create the VHDL architecture for the system.

Enter the architecture for the design. The architecture is where you describe the functionality of the system. You should name your architecture **top_arch**.



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

entity top is
  port (SW    : in  std_logic_vector(3 downto 0);
        LEDR  : out std_logic_vector (3 downto 0);
        HEX0  : out std_logic_vector (6 downto 0));
end entity;

architecture top_arch of top is

  begin

    -- YOUR DESIGN GOES HERE...


end architecture;
```

The first part of your design should drive the SW inputs to the LEDR outputs. This can be done using a simple signal assignment. This does NOT need a process.

The second part of your design is the decoder logic. You should create a process to perform the decoding. The process is implementing combinational logic so the sensitivity list should contain all inputs (i.e., SW). You can use either if/then or case statements within the process to implement the decoder. Implement the logic from your table in part 1. Always remember that the process is driving the outputs HEX0 depending on the inputs SW.

Once done, save your top.vhd.

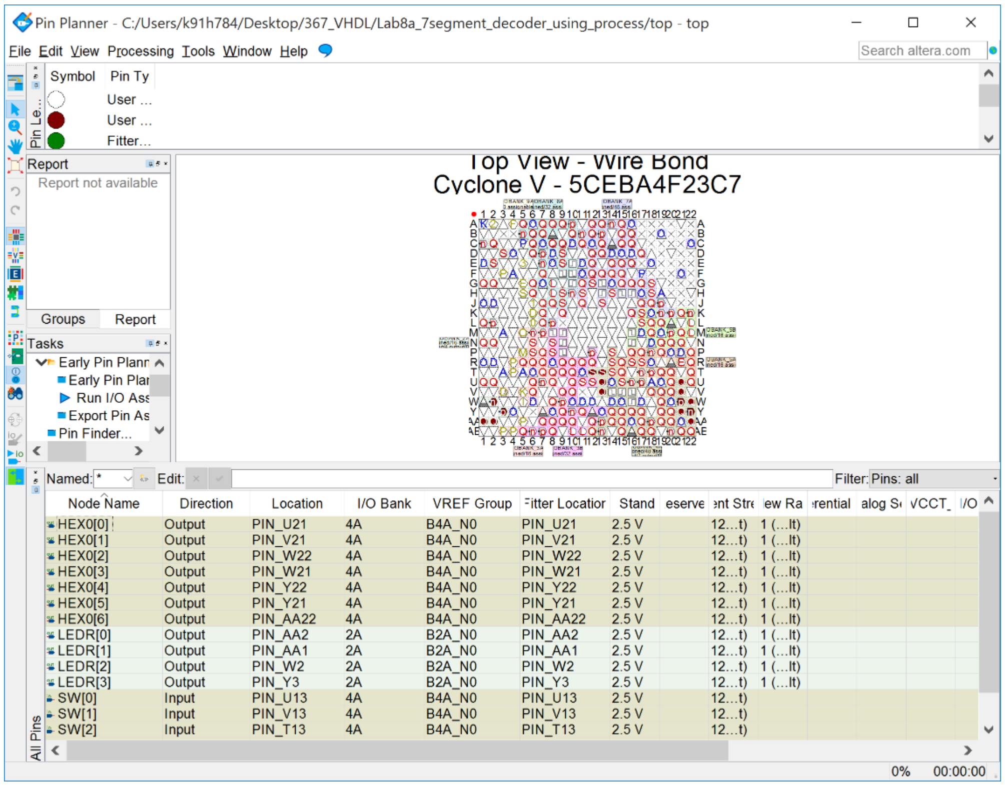
H) Compile your design.

You are now going to compile your design in order to find any typos you may have made. All of the synthesis operations within Quartus can be launched by double clicking on the *flow* pane. Double click on “Compile Design”. This will launch all of the steps beneath it.

Note that in this step, Quartus will implement the design as much as it can. At this point, you haven’t entered the pins assignments yet. So once the compile is done, you will do that. The compile step is necessary in order to read in the port names so Quartus knows the names to assign pins to.

I) Assign the pins of the FPGA.

After successfully compiling your design, you need to tell Quartus where the ports of our entity should be connected to the pins of the FPGA. Launch the *Pin Planner* tool using the pull-down menus *Assignments – Pin Planner*. You will see a graphical depiction of the FPGA pins. At the bottom, you will see all of the ports that you defined in your entity. For each port, double click in the “Location” box and enter the pin numbers given in the block diagram in this document. Once done, close the Pin Planner window (it will save automatically). Now recompile your design.



Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Locatior	Stand	eserve	ent Stre	ew Ra	erential	alog S	VCCT_	I/O
HEX0[0]	Output	PIN_U21	4A	B4A_N0	PIN_U21	2.5 V		12...t)	1 (...lt)				
HEX0[1]	Output	PIN_V21	4A	B4A_N0	PIN_V21	2.5 V		12...t)	1 (...lt)				
HEX0[2]	Output	PIN_W22	4A	B4A_N0	PIN_W22	2.5 V		12...t)	1 (...lt)				
HEX0[3]	Output	PIN_W21	4A	B4A_N0	PIN_W21	2.5 V		12...t)	1 (...lt)				
HEX0[4]	Output	PIN_Y22	4A	B4A_N0	PIN_Y22	2.5 V		12...t)	1 (...lt)				
HEX0[5]	Output	PIN_Y21	4A	B4A_N0	PIN_Y21	2.5 V		12...t)	1 (...lt)				
HEX0[6]	Output	PIN_AA22	4A	B4A_N0	PIN_AA22	2.5 V		12...t)	1 (...lt)				
LEDR[0]	Output	PIN_AA2	2A	B2A_N0	PIN_AA2	2.5 V		12...t)	1 (...lt)				
LEDR[1]	Output	PIN_AA1	2A	B2A_N0	PIN_AA1	2.5 V		12...t)	1 (...lt)				
LEDR[2]	Output	PIN_W2	2A	B2A_N0	PIN_W2	2.5 V		12...t)	1 (...lt)				
LEDR[3]	Output	PIN_Y3	2A	B2A_N0	PIN_Y3	2.5 V		12...t)	1 (...lt)				
SW[0]	Input	PIN_U13	4A	B4A_N0	PIN_U13	2.5 V		12...t)					
SW[1]	Input	PIN_V13	4A	B4A_N0	PIN_V13	2.5 V		12...t)					
SW[2]	Input	PIN_T13	4A	B4A_N0	PIN_T13	2.5 V		12...t)					

J) Program the FPGA

You are now going to download your design to the FPGA. Connect the DE0-CV board to the computer using a USB cable. Turn the board on using the big red button.

NOTE: You have to have the slider switch to the left of the HEX displays in the RUN position, NOT the PROG position. The PROG position is used for something else. If this switch is in the PROG position the download will fail.

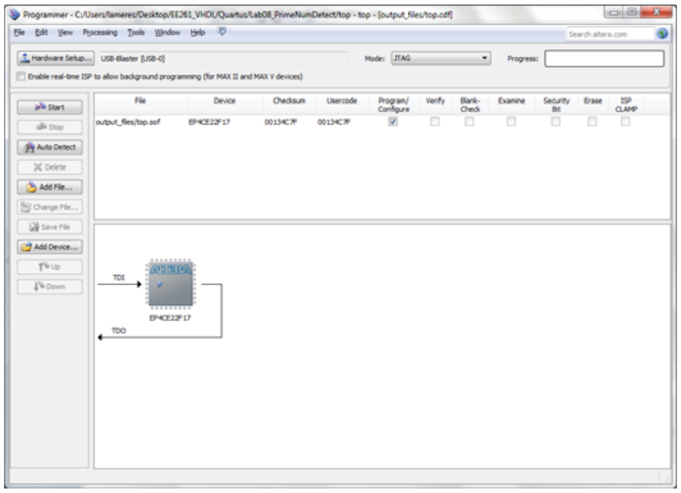
In the flow pane of Quartus, double click on “Program Device (Open Programmer)”.

Click on “Auto Detect” and the programmer will go out and find the FPGA. It will return with all programmable devices on the programming chain. Choose the 5CEBA.. FPGA.

The FPGA now shows up graphically. You need to assign the programming file to be downloaded. Right click on the device, and select “Add File”. You will be brought to your Quartus project directory. Double click on the “output_files” subfolder. You will see a file called “top.sof”. Select this file and click “OK”.

NOTE: Sometimes the programmer will add a second version of the FPGA after you select this file. If it does this, highlight the FPGA that shows up that does NOT have an output associated with it and delete it.

You are now ready to download your design to the FPGA. Click on the “Start” button.



Programmer - C:/Users/lameres/Desktop/EE261_VHDL/Quartus/Lab08_PrimeNumDetect/top - top - [output_files/top.cdf]

Hardware Setup... USB-Blaster [USB-0]   Mode: JTAG   Progress:

Enable real-time ISP to allow background programming (for MAX II and MAX V devices)

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
output_files/top.sof	EP4CE22F17	00134C7F	00134C7F	✓						

K) Test your design

You should now see the values of the slider switches being displayed on the LEDS and the HEX0 display showing the corresponding character. Verify that your design works for each of the 16 possible input codes. Fix any errors you discover. Note that if you change *anything* in your VHDL design file, you will need to fully compile/synthesize your design and then reprogram. If you don’t close the programmer window, you won’t have to reinitialize the chain and assign the output file. You can just click “Start” on the programmer to download the new design.

L) Demonstrate your design.

DEMO – Show the lab instructor the proper operation of your design. Once checked off, upload your top.vhd file to the lab DropBox. This file is found in the main project folder for your design.

Lab Grading

Demo – Proper Operation of your Circuit	____________ **/ 90**
Review of your top.vhd file	____________ **/ 10** (uploaded to DropBox)
Total	____________ **/ 100**

Post Lab Survey (Just for your own knowledge. This is not graded)

1) Can you use the Quartus development tools to design and implement a VHDL design on the Cyclone V FPGA?
2) Can you use a VHDL process and conditional programming constructs (i.e., if/then or case) to create the logic for a 7-segment decoder?