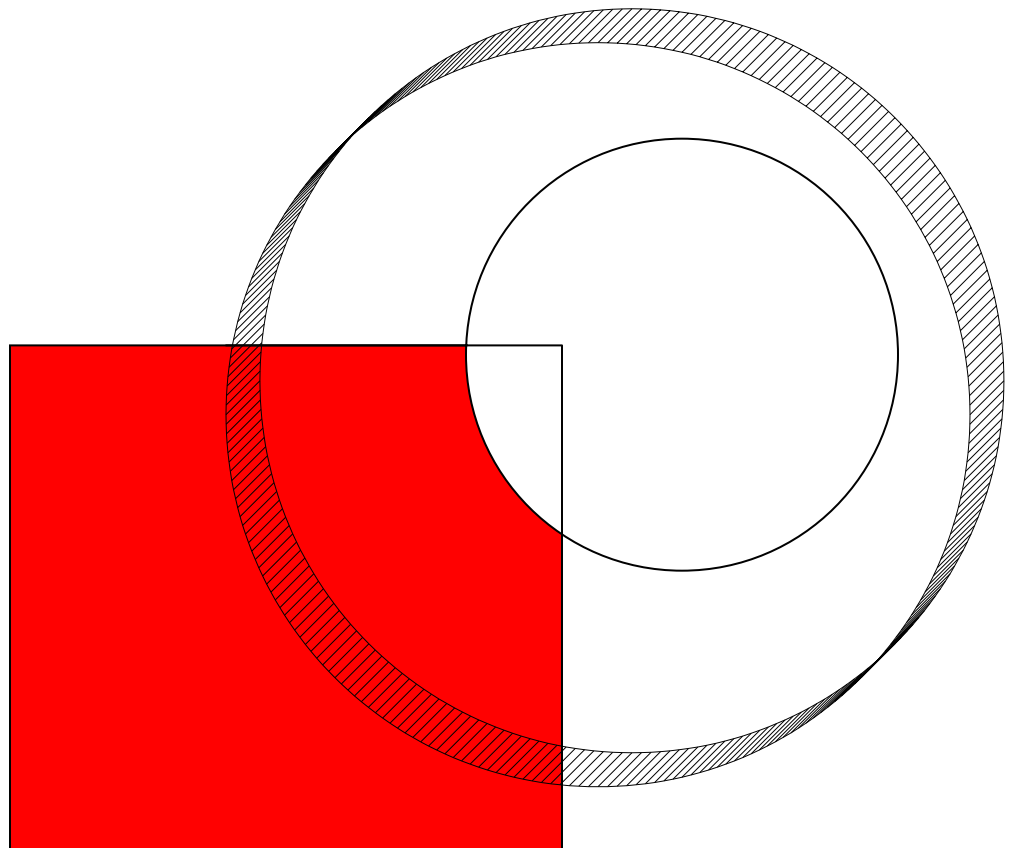


Seven Segment Display Module

Supplemental Tutorial



Overview

This tutorial discusses how to use the built-in seven segment displays on the Pegasus FPGA. In order to use the seven segment displays you will need to add a source file to your XILINX project from the internet and create a schematic symbol from it. The tutorial is broken up into three sections.

- Getting the Verilog file
- Creating the Schematic Symbol
- Using the Sevenseg Schematic Block

“Getting the Verilog” discusses where to find the file and how to add it properly to your project. “Creating the Schematic Symbol” discusses how to create the symbol and where to access it. “Using the Sevenseg Schematic Block” discusses the functions and options built into the schematic symbol.

The internet file **MUST** be saved in the folder containing the XILINX project you are trying to use the module in. It must then be converted from Verilog code into a useable block of logic. From here you can use it inside schematics in XILINX.

When you are completed, you will be able to use all of the above. Also, the FPGA has special pins for the outputs of the block. These pins **MUST** be given specific pin assignments in the UCF file.

Basic Functionality

The block has four input **busses**. Each bus is a four bit number.

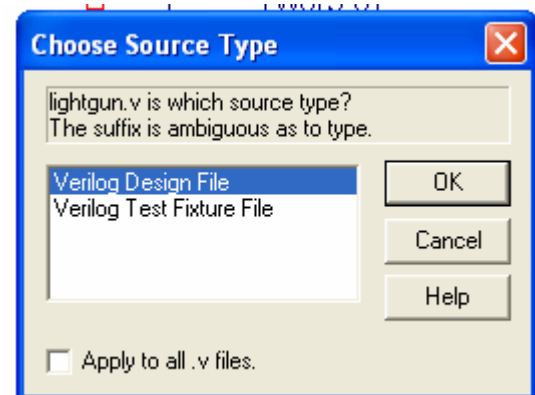
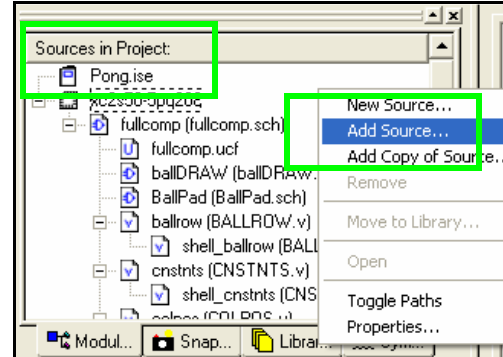
1. The Block converts the number into logic to drive the seven segments.
2. The seven segments display the four bit input in hexadecimal.
3. The outputs of the block must be tied to specific pins on the FPGA (this is done in the UCF file).

The block also has a provision for you to control each individual segment of the display. This is discussed in the last section of this tutorial.

This tutorial assumes you know how to find pin assignments on the FPGA datasheet and how to make a UCF file in your project. If this is not the case, you will need to find out how.

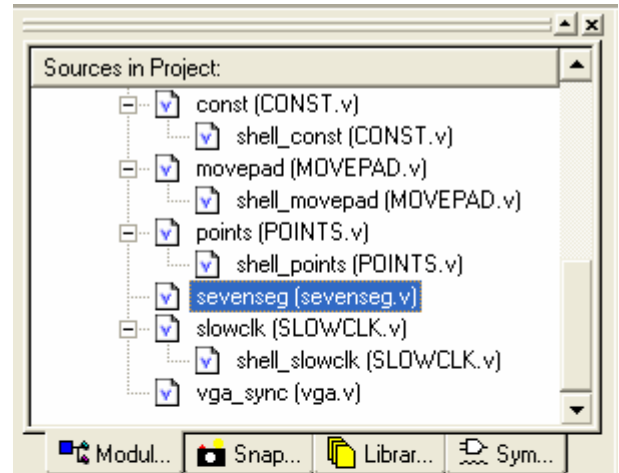
Getting the Verilog File

1. Access the **tutorials** section of the ECEN3233 webpage.
2. **Right Click** on **Sevensseg.v** and select **save target as**.
3. **Save the file** into the **current file** you are using to save your XILINX sources.
4. **Verify** the target was saved by opening the folder and checking the filename. It should say **Sevensseg.v**. *The folder **MUST** be the same folder that contains your project or XILINX will not create a schematic symbol. XILINX has difficulty with spaces in filenames so the easiest way to avoid this is to save the file into the project folder.*
5. **Open** your current XILINX project.
6. **Right Click** inside the **Sources in Project** window and select **add source**.
7. **Double Click** on **Sevensseg.v** and select **Verilog Design File** when prompted about source type.

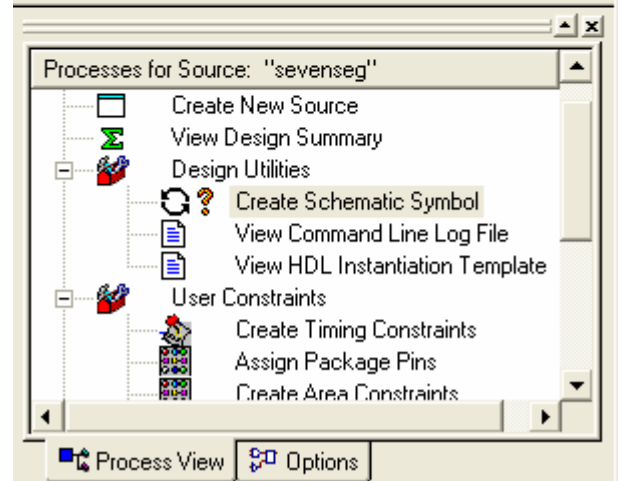


Creating the Schematic Symbol

1. **Highlight Sevensseg** in the **Sources in Project** window by clicking once on the module name.

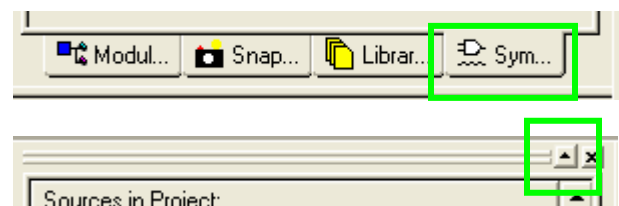


2. In the **Processes for Source** window, select **Create Schematic Symbol**. *The question mark next to the line will change to a check if done properly.*

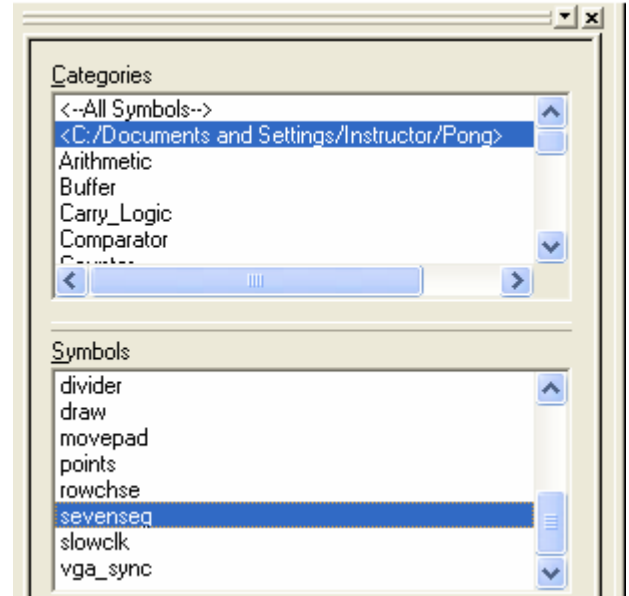


3. Open the schematic you wish to place the symbol in.

4. **Select the symbols** tab under the **sources in project** window and **expand** the window using the **arrow** in the upper right corner.



5. Under the **categories** list select the **folder and directory** you placed the Verilog file into.
6. Under the symbols list, **select Sevenseg**. You can now place the symbol inside your schematic.



Using the Sevenseg Schematic Block

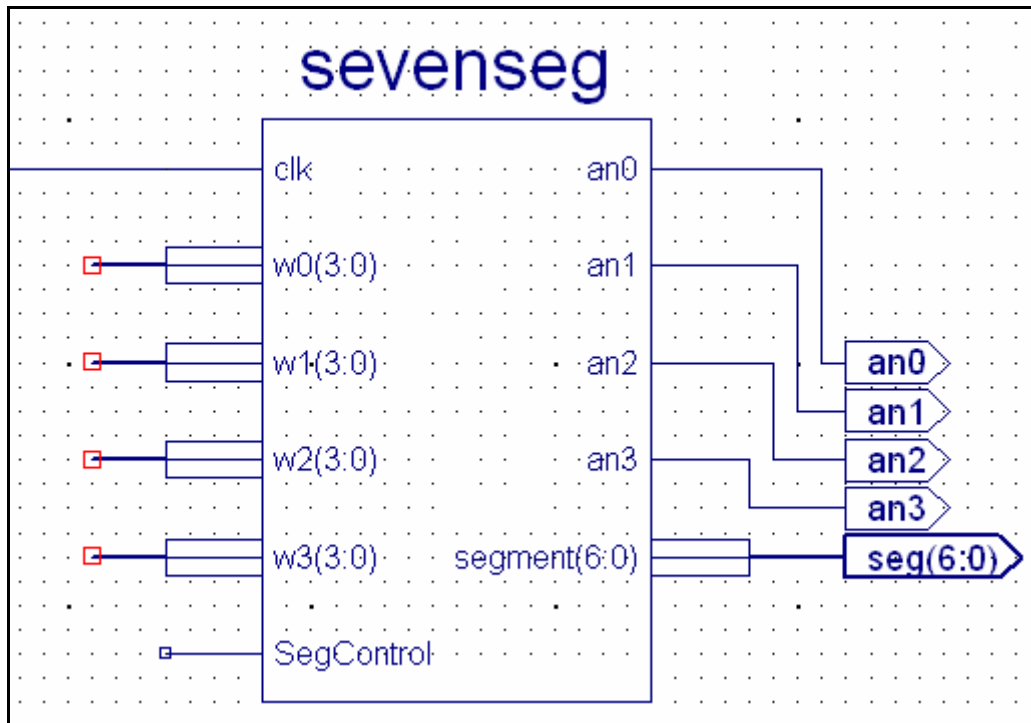


Figure 1

Normal Operation

Normal operation will take four separate inputs of four bits each and display them in hexadecimal on separate seven segment displays. Note that your block may not look like the above block. XILINX places the inputs and outputs in different places on the block sometimes.

Inputs:

Clk – the clock input must be the 50MHz signal generated internally by the FPGA. It will need the pin assignment for the global clock (usually pin77) on your Pegasus 4. This information can be found in the data sheet for the FPGA.

w0-w3 – each input is a four wire bus. Each bus controls one seven segment. The output will be hexadecimal.

For example, if $w0(0:3)$ is all logic zeros, then the first seven segment will be the number zero. If $w0(0:3)$ is '1000' in binary, then the seven segment will show an 8. The hexadecimal displays are 0-9, A, b, C, d, E, F for their corresponding four bit input.

SegControl – for normal operation the SegControl signal must be held at logic 0.

Outputs:

The output wires must each be given a distinct output pin on the FPGA in the UCF file. All of these are on the Data sheet for the Pegasus FPGA. The names for each pin are given below.

an0 – **an3** have the same names on the datasheet.

Block Output	FPGA Pin name
Segment[0]	= CA
Segment[1]	= CB
Segment[2]	= CC
Segment[3]	= CD
Segment[4]	= CE
Segment[5]	= CF
Segment[6]	= CG

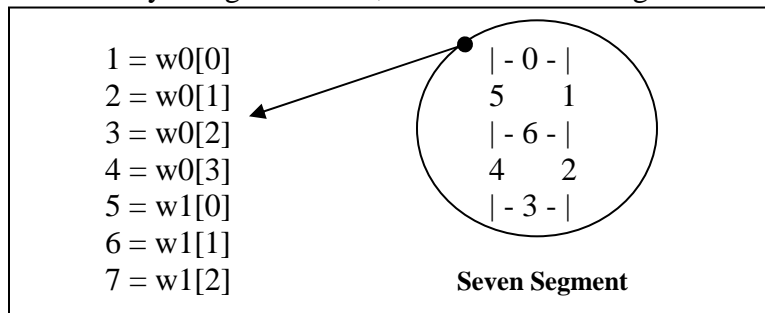
Segment controlled operation

Segment controlled operation is used if you wish to display something other than hexadecimal on the displays. With this, you can only use one seven segment display but are given control of each segment. The segments are active low, meaning they will light up if given a logic zero.

Inputs:

Clk – same as the normal operation clocked input.

w0–w1 – the input to each segment of the seven segment display is controlled individually using two buses, w0 and w1. The segments are controlled as follows:



w2-w3 - Leave the other input busses unconnected.

SegControl – this signal must be tied high for segment controlled operation.

Outputs:

The outputs must each be given a distinct output pin on the FPGA. All of these are on the Data sheet for the Pegasus FPGA. The names for each pin are given below.

an0 – connected same as for normal operation.

an1 – **an3** – each of these inputs must be tied high. For this, leave the outputs on the block unconnected and create three outputs, tying them all to power.

Block Output		FPGA Pin name
Segment[0]	=	CA
Segment[1]	=	CB
Segment[2]	=	CC
Segment[3]	=	CD
Segment[4]	=	CE
Segment[5]	=	CF
Segment[6]	=	CG