HDL Editor Lab

HDL Editor Lab

Introduction

This lab will enable the user to create the HEX2LED macro using the Foundation HDL Editor. This macro can then be added to the top-level schematic in Foundation. The HEX2LED macro is a 4-bit binary to seven-segment LED decoder (see Figure 1). Since HDLs lend themselves to generating large pieces of logic easily, VHDL is commonly used for similar applications such as muxes and state machines. If the decoder was built in the schematic editor, the appropriate logic would have to be designed and optimized on paper before being entered into the project. By using the HDL editor effectively, significant time and effort can be saved.

```
library IEEE;
use IEEE.std_logic_1164.all;

entity hex2led is
    port (
        hex: in STD_LOGIC_VECTOR (3 downto 0);
        led: buffer STD_LOGIC_VECTOR (6 downto 0)
        );
end hex2led;

architecture hex2led_arch of hex2led is
begin
  with hex select
        led <=  "1111001" when "0001",
                "0100100" when "0010",
                "0110000" when "0011",
                "0011001" when "0100",
                "0010010" when "0101",
                "0000010" when "0110",
                "1111000" when "0111",
                "0000000" when "1000",
                "0010000" when "1001",
                "0001000" when "1010",
                "0000011" when "1011",
                "1000110" when "1100",
                "0100001" when "1101",
                "0000110" when "1110",
                "0001110" when "1111",
                "1000000" when others;
end hex2led_arch;
```

Figure 1. The HEX2LED macrofunction.

Objectives

In this lab, it will be shown:

- How to create a symbol in the Foundation HDL Editor
- How to edit, check, and synthesize VHDL code in the Foundation HDL Editor
- How to add a VHDL macrofunction into a top-level schematic using the Foundation Schematic Editor

Procedure

Creating the HEX2LED symbol in the Foundation HDL Editor

1) Open the Foundation HDL Editor from within the Foundation Project Manager by clicking on the icon.

2) Once the HDL Editor has started, the "**HDL Editor**" dialog box will open (see Figure 2). Click on the "**Use HDL Design Wizard**" bubble and click on "**OK**".

3) The HDL Design Wizard enables the user to enter basic features and specify the ports for a new macrofunction. Click on "**Next**".

4) The HDL Editor can create designs using either ABEL or VHDL design languages. In the "**Languages**" dialog box, select "**VHDL**" and click on the "**Next**" pushbutton.

5) In the "**Name**" dialog box, enter "**HEX2LED**", and click on the "**Next**" pushbutton.

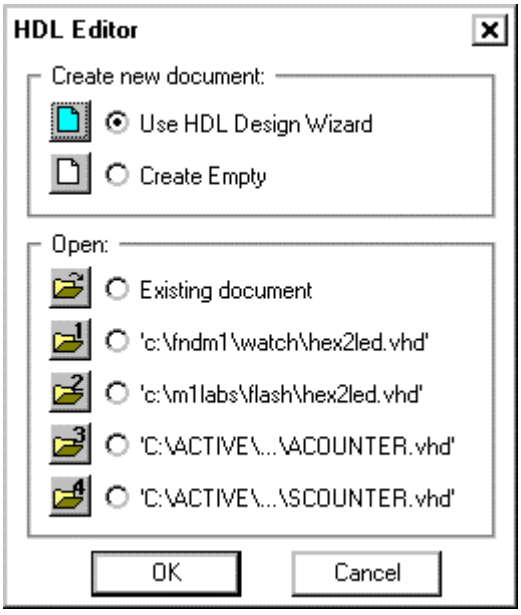

Figure 2. The HDL Editor dialog box.

6) The "**Port**" window allows the user to create input and output ports and to specify attributes for each port. Click on "**New**", and select a direction of "**Input**". Since the input bus is four bits wide, enter the name "**HEX**" and specify a bus width of "**3:0**" (see Figure 3). After entering the input bus, click on "**New**".

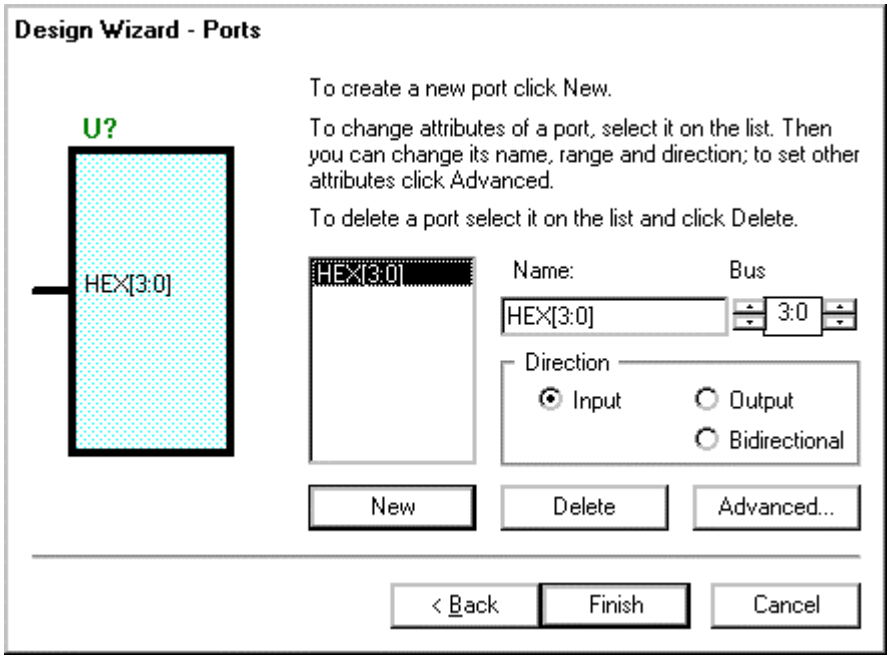

Figure 3. The "Ports" window.

7) For the LED port, select a direction of "**Output**", a name of "**LED**", and a bus width of "**6:0**". After entering these attributes, click on "**Finish**".

8) The Design Wizard creates the "**HEX2LED**" symbol and opens the "**HDL Editor**" with the necessary VHDL Design Template entered (see Figure 4).

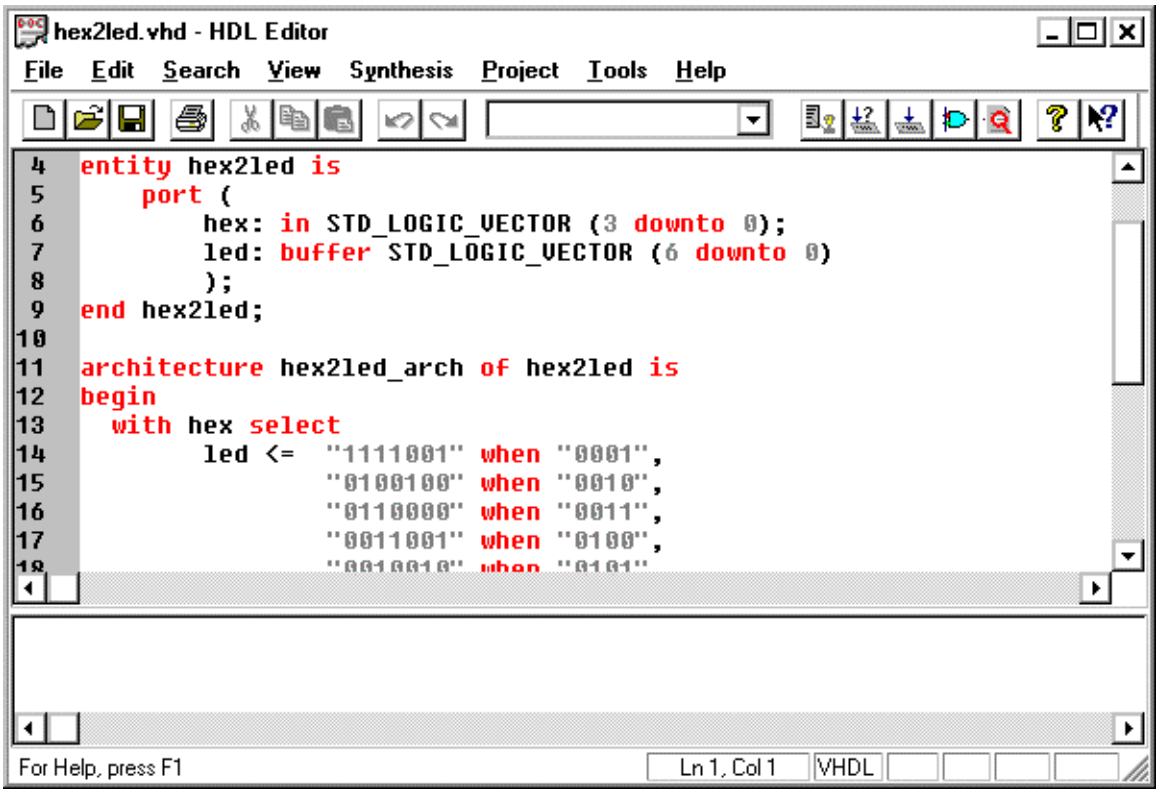

Figure 4. The HDL Editor with the VHDL Design Template entered.

Entering VHDL code into the HDL Editor for the HEX2LED macro

1) The Standard Template entered into the design includes all the input and output ports entered into the Design Wizard. Since it is color coded, it is easy to see where the additional VHDL code must be entered. Delete the line with **"--<<enter your statements here>>**", and replace it with the code in Figure 5. Note that in the HDL Editor, any characters after a "--" is considered a comment, and is ignored by the synthesizer.

```
with HEX select
        LED<=       "1111001" when "0001",      --1
                    "0100100" when "0010",      --2
                    "0110000" when "0011",      --3
                    "0011001" when "0100",      --4
                    "0010010" when "0101",      --5
                    "0000010" when "0110",      --6
                    "1111000" when "0111",      --7
                    "0000000" when "1000",      --8
                    "0010000" when "1001",      --9
                    "0001000" when "1010",      --A
                    "0000011" when "1011",      --B
                    "1000110" when "1100",      --C
                    "0100001" when "1101",      --D
                    "0000110" when "1110",      --E
                    "0001110" when "1111",      --F
                    "1000000" when others;      --0
```

Figure 5. The necessary code for the HEX2LED macrofunction.

Checking the syntax of the HEX2LED macro's VHDL code

1) After entering the VHDL code into the editor, click on the menu command: ***Synthesis>Chech Syntax***. This will enable the Metamor synthesizer to verify the syntax of each line of code. If any warnings or errors are reported, compare the design file with those in Figures 4 and 5. If no warnings occur, click on "**OK**" when the check is complete.

Synthesizing the HEX2LED's VHDL code

1) Next, generate the synthesized code by clicking on the menu command: ***Synthesis>Synthesize***. This converts the VHDL code into an EDIF netlist that can be combined with the rest of a project. If no warnings or errors occur, click on "**OK**" when the synthesis is complete.

2) Click on the command: ***Project>Create Macro***.

Adding the HEX2LED macrofunction into the WATCH Project

1) After the macro has been created, it is associated with the current project and can be entered into the schematic. After exiting the HDL Editor, click on the "**Schematic Editor**" icon in the Project Manager.

2) Once the program has started, click on the "**Symbols Toolbox**" icon on the vertical toolbar. Find the HEX2LED macro by entering its name at the bottom of the "**SC Symbols**" box, and place it in the schematic as seen in Figure 6.

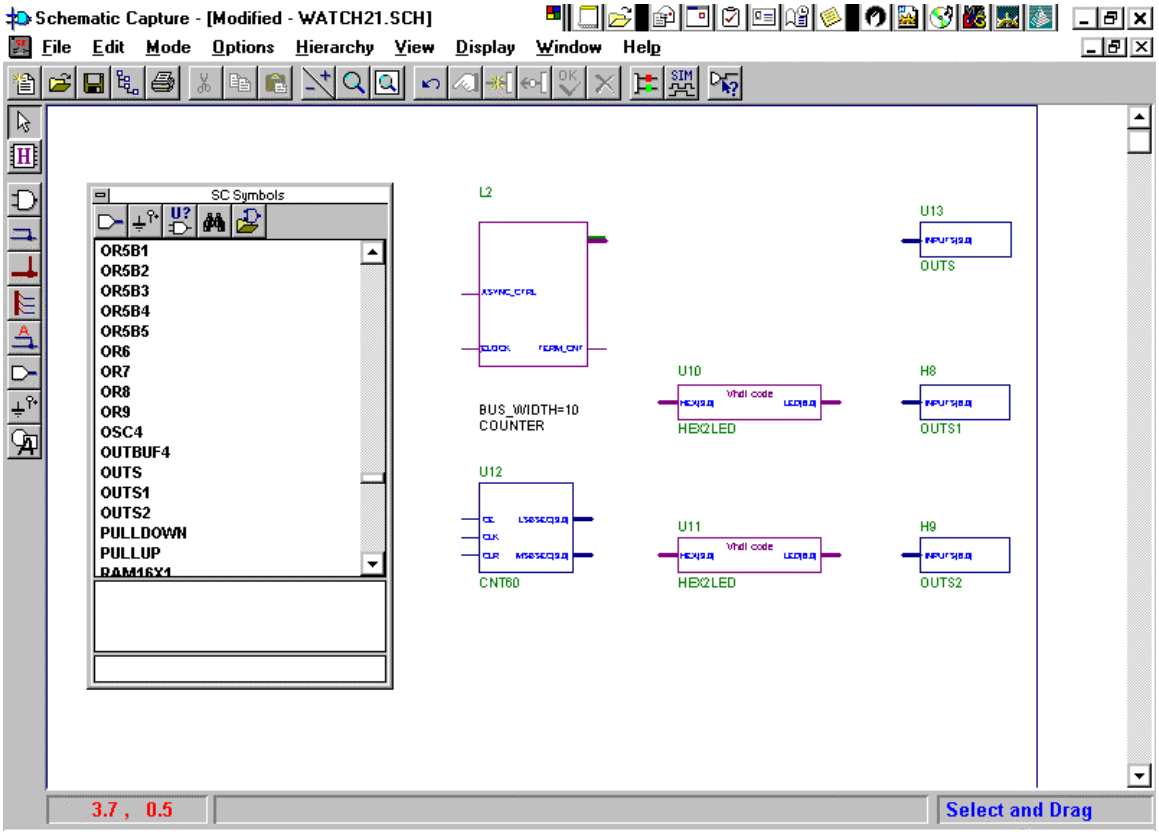

Figure 6. The WATCH Project with the HEX2LED macrofunction.

3) Save the WATCH Project within the Schematic Editor with the menu command: ***File>Save***.

4) Exit the Schematic Editor by clicking on ***File>Exit***.

Conclusion

In this laboratory, it was shown:

- The HDL Design Wizard is used to create a symbol that represents the VHDL code entered for the HEX2LED macro.
- The Standard Template is entered automatically for the user after creating a symbol using the HDL Design Wizard.
- After entering the VHDL code for the HEX2LED macro, the syntax had to be checked and synthesized.
- After compilation, the HEX2LED can be entered into the top-level schematic using the Schematic Editor and the Symbols Toolbox.

Questions

1) What HDL languages does Foundation support?

2) Is it possible to enter the HDL code without first creating the necessary symbol?