

HOMEWORK – MODELSIM ANIMATION

Prof. Don Bouldin and Narayanan Raghuraman – 25 May 2004

Revised 28 September 2007

Purpose: Perform pre-synthesis simulation with animation using *ModelSim*.

Part A – Tutorial

To begin, make a subdirectory for this homework:

1. **mkdir hw4-07**

Then, copy the appropriate files to the new subdirectory:

2. **cp ~bouldin/webhome/protected/animate.tar hw4-07**

Now, move down to the subdirectory:

3. **cd hw4-07**

Expand the tar file by typing:

4. **tar -xvf animate.tar**

Now, move down to the subdirectory:

5. **cd animate**

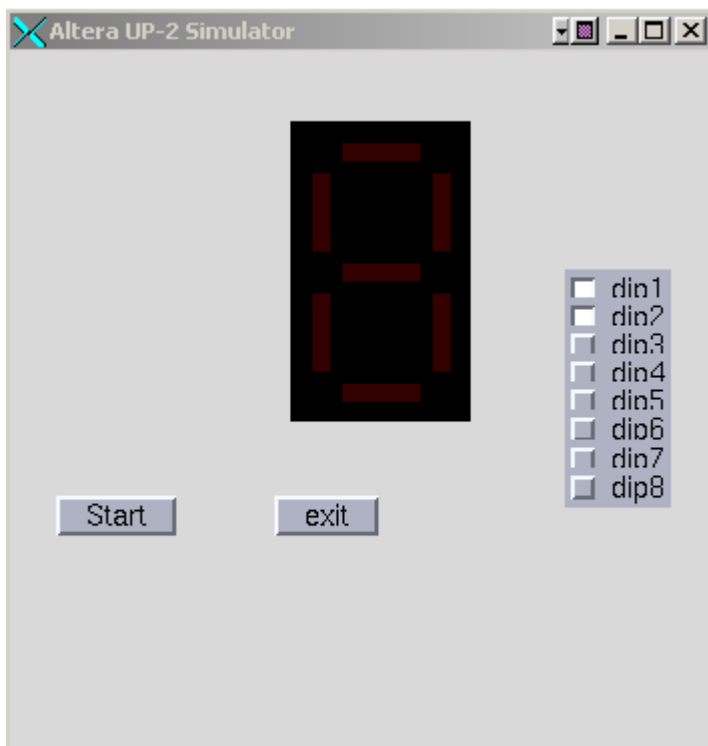
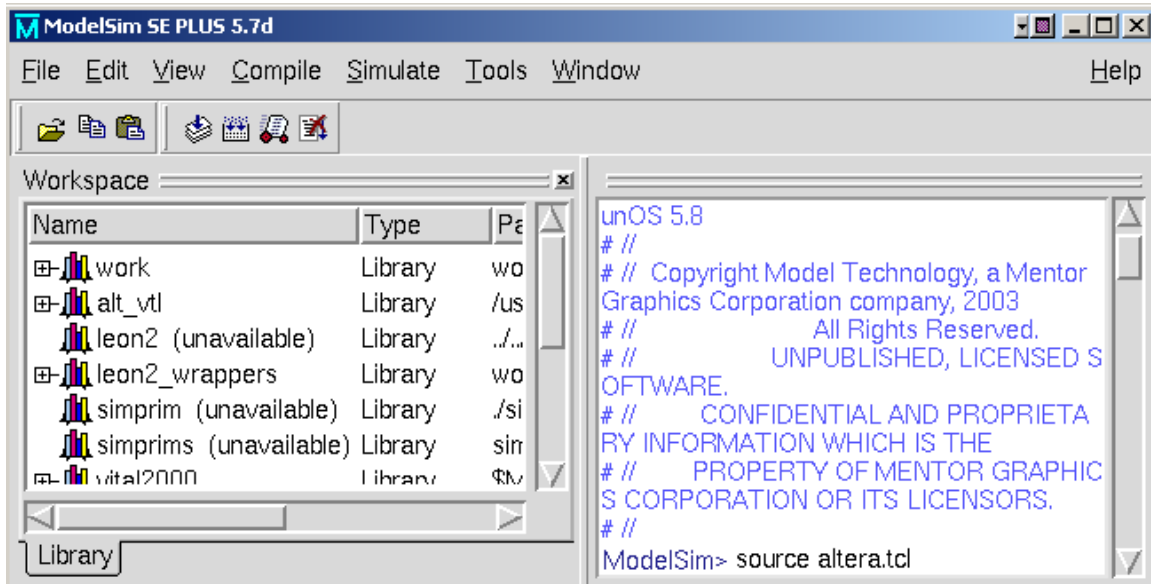
Now, perform pre-synthesis simulation by typing:

6. **vsim &**

In the command window, type:

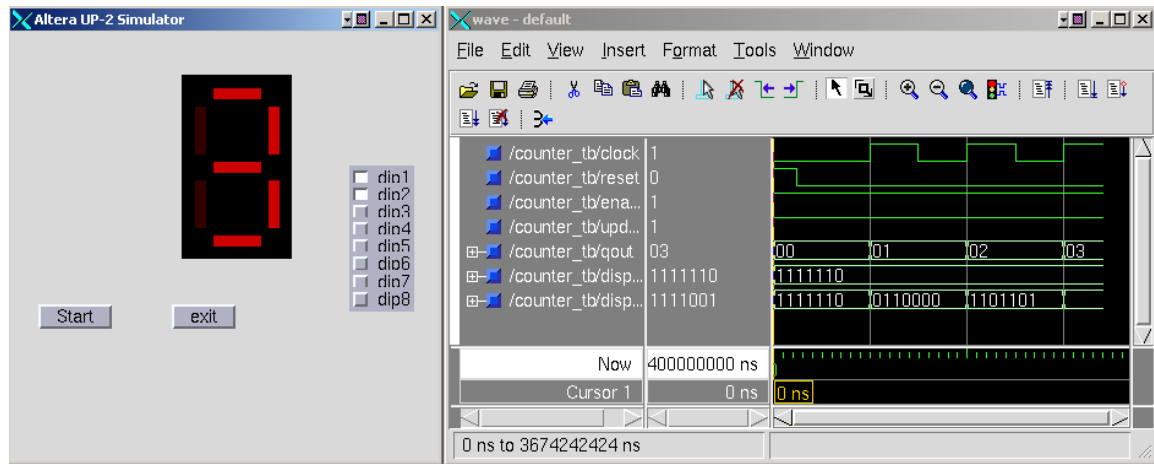
7. **source altera.tcl**

This will bring up the following windows:



8. **Press “Start”**

You should be able to see the counter running and the signals in the waveform window:



9. **Press “dip1”** (changes the direction of counting)

10. **Press “dip2”** (disables the counting)

11. **Press “dip1”** (changes the direction of counting)

12. **Press “dip2”** (disables the counting)

13. Capture the waveform and display windows to illustrate that you have exercised both dip switches.

14. **Press “exit”** (quits the simulation)

Copy the tcl file and edit it:

15. **cp altera.tcl altera_2.tcl**

16. Navigate to the part where it says “uncomment the following lines to add ‘Reset’ Button” and then uncomment them.

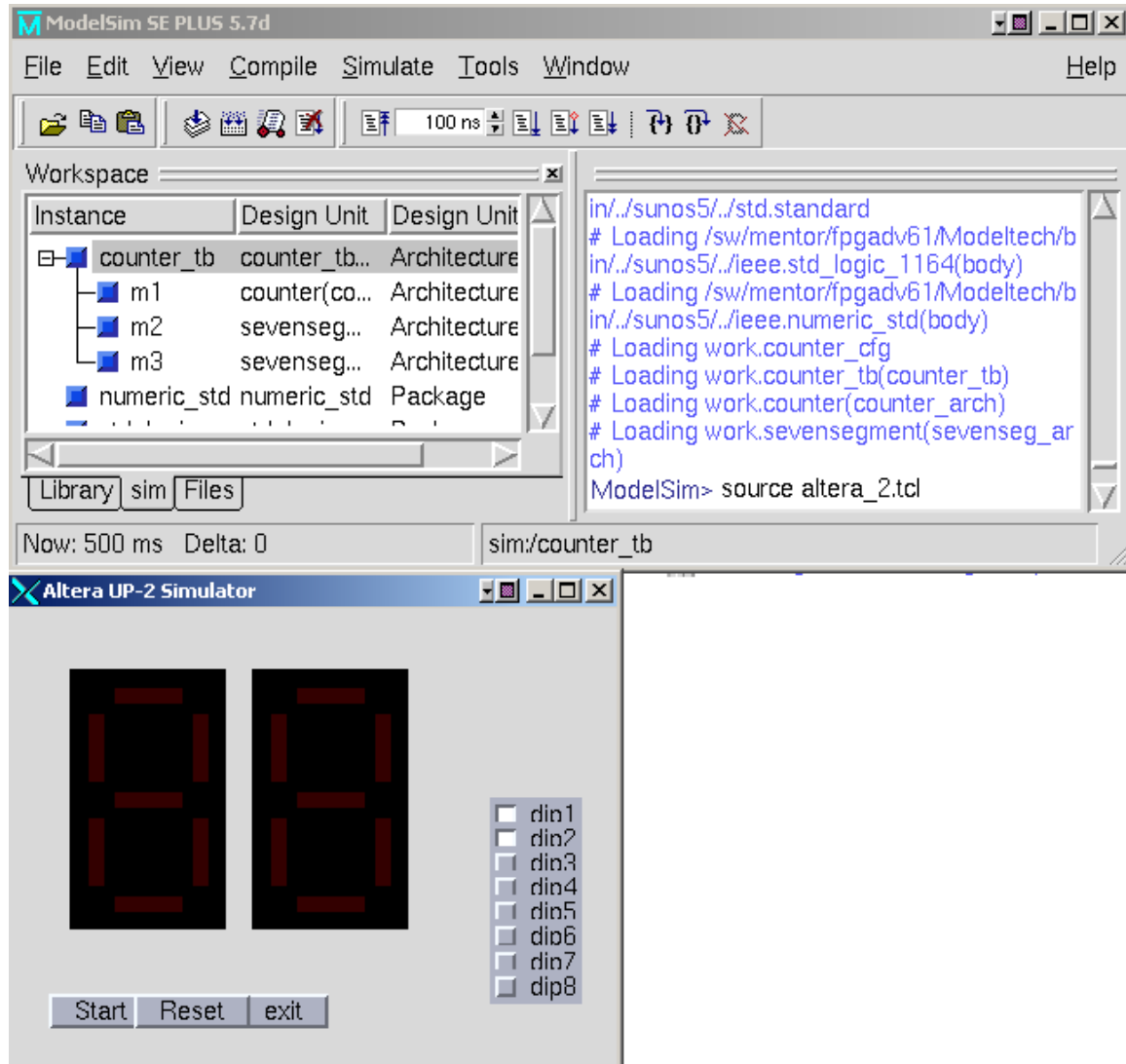
17. Next go to the part which says “uncomment the following lines for adding display 2” and uncomment them.

18. Save the file and quit the editor.

In the command window, type:

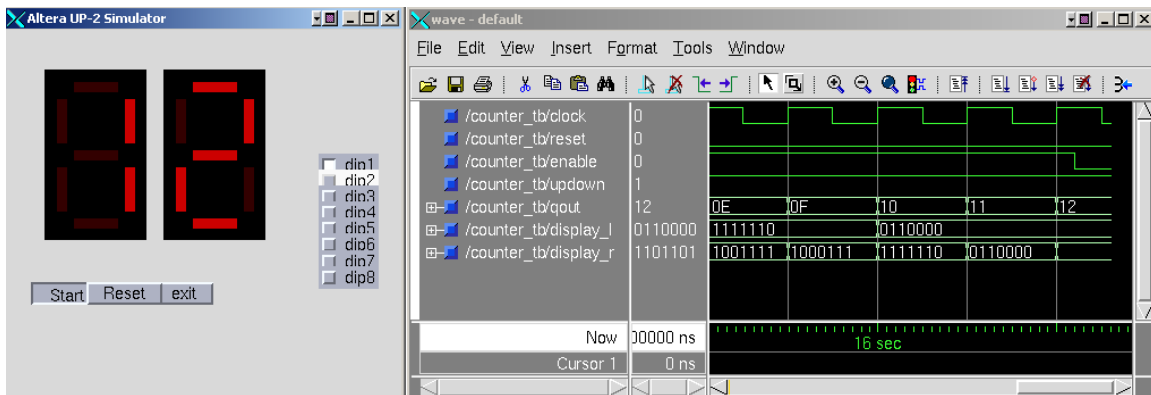
19. **source altera_2.tcl**

This will bring up the following windows:



The animation window should now show the new additions of a second display digit and the “reset” button.

20. Press “start” and let the counter go up to “12”.



21. Press “dip1” (changes the direction of counting)
22. Press “dip2” (disables the counting)
23. Press “reset” (forces the counter to “00”)
22. Press “dip2” (disables the counting)
23. Capture the waveform and display windows to illustrate that you have exercised the switches.
24. Press “exit” (quits the simulation)

To learn more about animation, see the Appendix to this document.

Appendix

Here is a brief description of the functions in the altera_2.tcl file:

start_running - initializes the simulation. This is linked to the start button.

run_clock - This is initialized from start_running.
It runs recursively every 100 ms.

paint_displays - called from run_clock and also initially.
This sets the value of the displays after examining the state of the signals from the testbench.

CreateWindow - This function creates the complete interface including buttons, checkbuttons and the displays.

close_window - terminates the simulation. called by the 'exit' button.

create_display - creates the seven segment display.
It creates an individual canvas so that the background color can be set as black. The different segments are created as rectangles and set to have a very light red color initially.

update_display - updates the color of the 7-segment display.
It uses the command itemconfigure to change the color
The command 'index' selects the color from the array \$colours according to the variable \$pattern (which is nothing but the output signal from the VHDL design).

run_command - This procedure is used to group together all the commands to be executed. since all commands are subsets of this procedure, it reduces the risk associated with accidentally calling any of modelsim's built-in commands.

initialize - this procedure compiles the vhdl files and loads the design.
It also initializes the wave window and adds signals to it.
It also sets the initial values of the inputs and runs the simulation for 500 ms.

CHANGES MADE

Before describing the steps, the three steps in the addition of a widget are described.

These three steps constitute the first change to be made in part II.

The first step is the creation of the widget. Some of the common attributes like the dimensions, the orientation, the background and foreground colors, the text to be displayed on or beside the widget, are set in this step.

The next step is packing or placing the widgets. The difference between the 'pack' and 'place' command is that the 'place' command can specify the exact location in the canvas. The 'pack' command specifies relative location and also bind the widget to a function. Without this step, the widgets are invisible.

The third optional step in this process is the binding of the widgets if necessary. The button widget needs explicit binding if different commands need to be executed at the pressed and released states of the button. For most other widgets the command to be executed is specified in the 'creation' step.

The other change that needs explanation is the addition of the display. This is done with a function named 'create_display'. A tag is passed as the argument to this function. This tag when passed to the 'update_display' function will set the states of the various line widgets to the required state. The two widgets are named as 'display_l' and 'display_r' to signify the left and right display widgets. A variable named 'twodisplays' is used so that the updating of the display proceeds smoothly when the second widget is added. This is used in the function 'paint_displays' which gets called each time the output changes state.