Using *DESim* on DESL and ECF Computers

This tutorial shows you how to obtain access to the *DESim* software tool on ECF and DESL computers. We assume that you already know how to login to ECF and/or DESL machines by using your ECF login and Windows Remote Desktop. If you are not familiar with the remote-login process, please read the guide(s) available at:

`https://ssl.ecf.utoronto.ca/ecf/services/rd`

We do not give detailed instructions in this guide for using the *DESim* software tool, but only show how to obtain this software on the ECF and DESL machines. Instructions for using the *DESim* software are included with the design files for this exercise. These instructions are also available in the `Releases` section of the GitHub repository `https://github.com/fpgacademy/DESim`.

Getting Started

The discussion below assumes that you are using an ECF machine; any changes in the instructions that apply for DESL machines are also indicated. After you are connected to an ECF machine, open `File Explorer`. As indicated in Figure 1 navigate to the folder `S:\Courses\ECE243S\ECF`, which contains the folder named `DESim`. As indicated in the figure, right-click on `DESim` and select the `Copy` command. If you are logged into a DESL computer, then the same instructions should be followed, except that you should navigate to the folder `S:\Courses\ECE243S\DESL`.

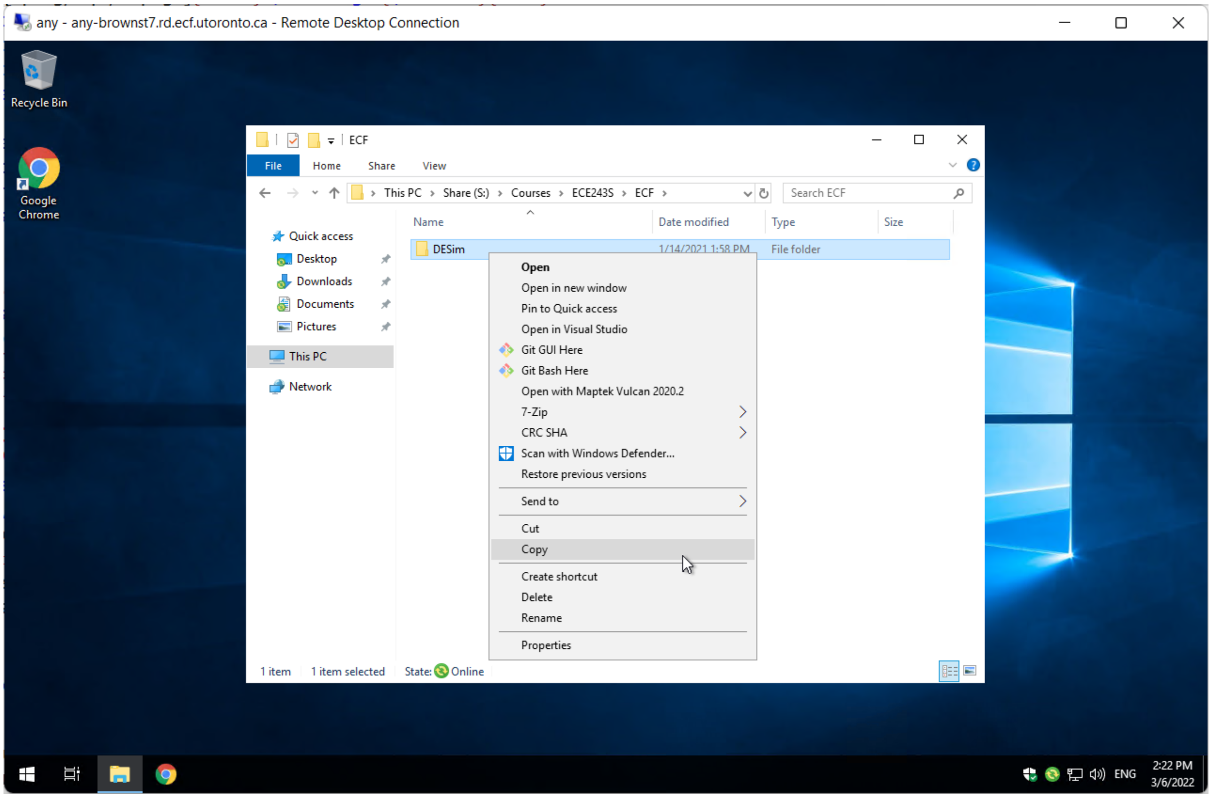


Figure 1: Copying the `DESim` folder from the `S:` drive.

The DESim folder copied above contains the *DESim* software program and some example projects (called *demos*). You need to make a personal copy of this folder for your own use. Navigate to your W: drive and paste the DESim folder, as depicted in Figure 2.

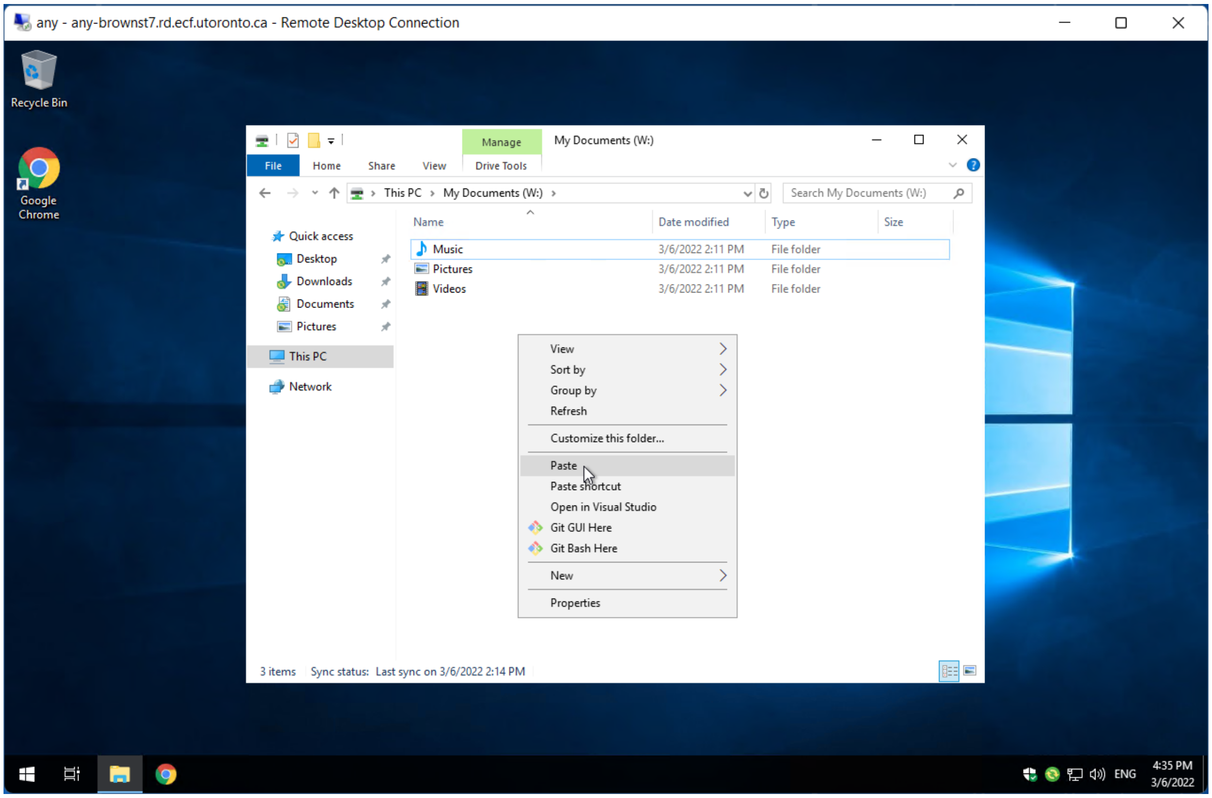
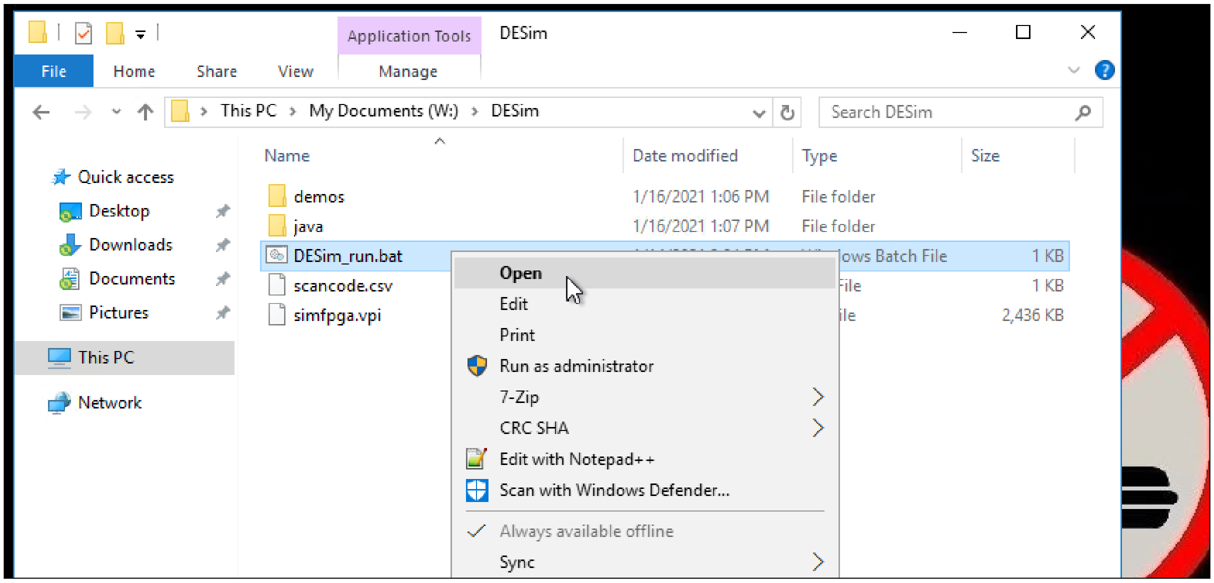


Figure 2: Pasting the DESim folder into your W: drive.

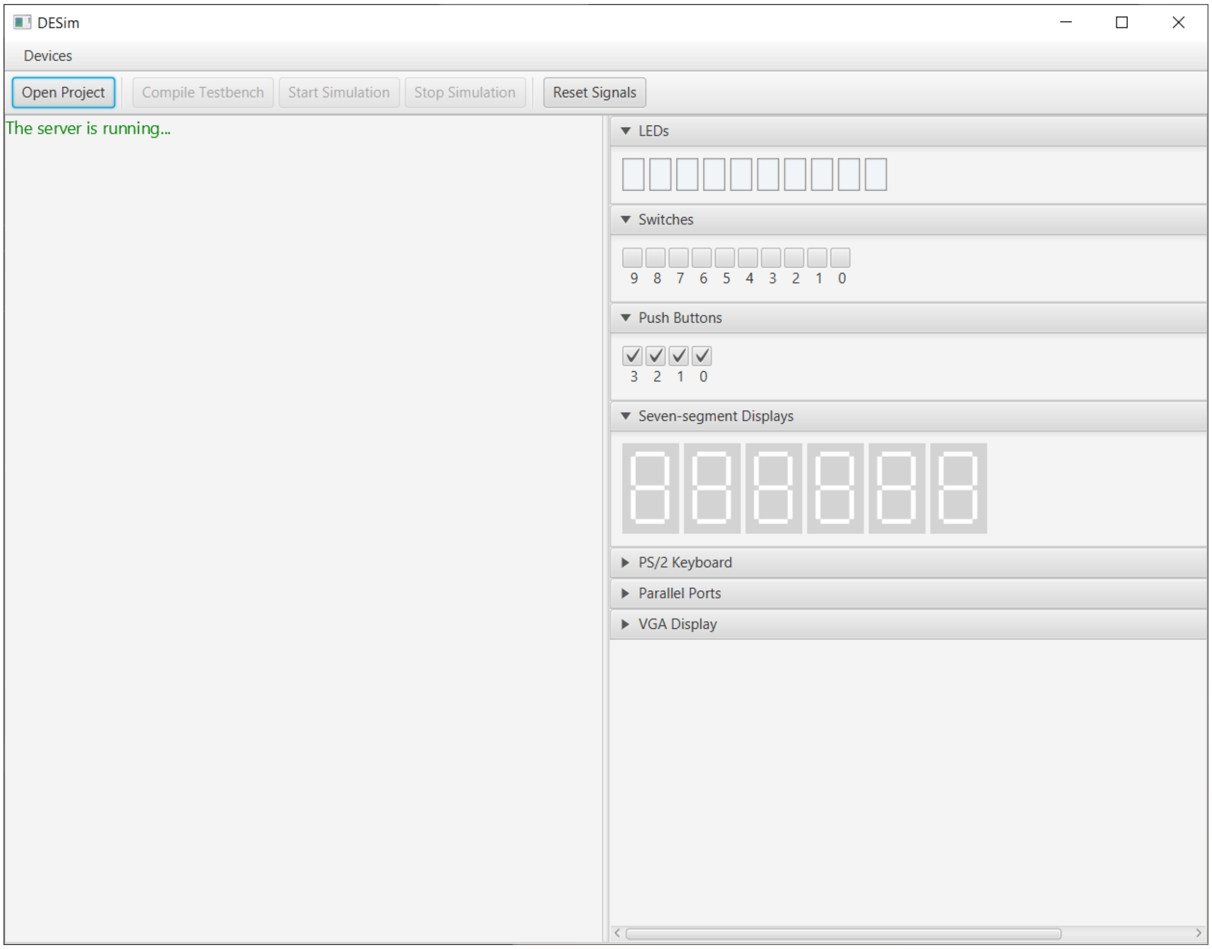
Now, navigate into the W:\DESim folder that you created. To run the *DESim* program, in File Explorer right-click on the *batch* file DESim_run.bat, as illustrated in Figure 3, and then select the Open command (alternatively, you can double-click on the batch file to open it). This action opens the *DESim* graphical user interface (GUI), displayed in Figure 4. You should see the message "The server is running..." near the top of the *message pane* in the *DESim* GUI. If you do not see this message, but instead you see a message saying that the Server setup failed, then this could be an issue with your W: drive. If this happens, then please copy the DESim folder to the C: drive instead of W: (this is not ideal, because the C: drive is unique to each machine and is cleared each night, but if the W: drive doesn't work for you, then C: may be the only option).

To ensure that your *DEsim* program can communicate with the *ModelSim* simulator, you may wish to try out one (or more) of the *demo* projects that come with *DESim*. As an example, click the Open Project command in the *DESim* GUI and then navigate into the demos folder. As illustrated in Figure 5, click to select the folder named LED_HEX and then click on the Select Folder button.



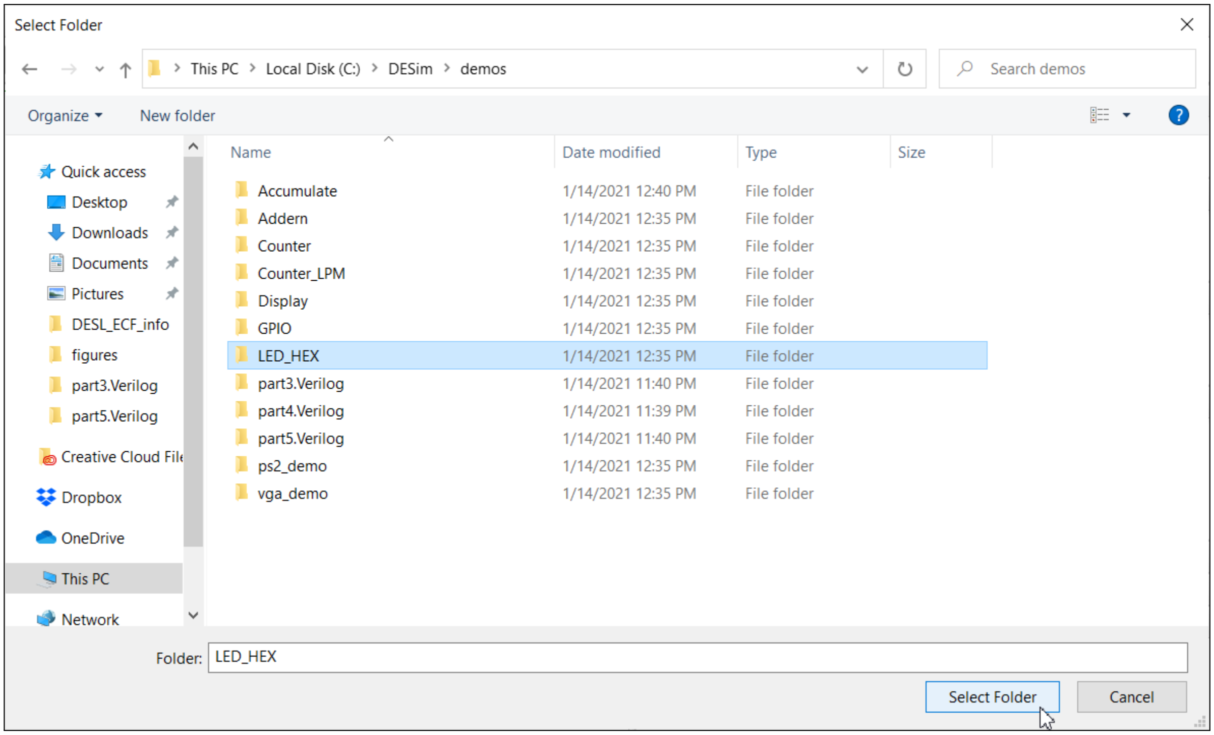
Name	Date modified	Type	Size
demos	1/16/2021 1:06 PM	File folder	
java	1/16/2021 1:07 PM	File folder	
DESim_run.bat		ows Batch File	1 KB
scancode.csv		File	1 KB
simfpga.vpi		ile	2,436 KB

Figure 3: Starting the *DESim* software.



The server is running...

Figure 4: The *DESim* window.



Name	Date modified	Type	Size
Accumulate	1/14/2021 12:40 PM	File folder	
Addern	1/14/2021 12:35 PM	File folder	
Counter	1/14/2021 12:35 PM	File folder	
Counter_LPM	1/14/2021 12:35 PM	File folder	
Display	1/14/2021 12:35 PM	File folder	
GPIO	1/14/2021 12:35 PM	File folder	
LED_HEX	1/14/2021 12:35 PM	File folder	
part3.Verilog	1/14/2021 11:40 PM	File folder	
part4.Verilog	1/14/2021 11:39 PM	File folder	
part5.Verilog	1/14/2021 11:40 PM	File folder	
ps2_demo	1/14/2021 12:35 PM	File folder	
vga_demo	1/14/2021 12:35 PM	File folder	

Folder: LED_HEX

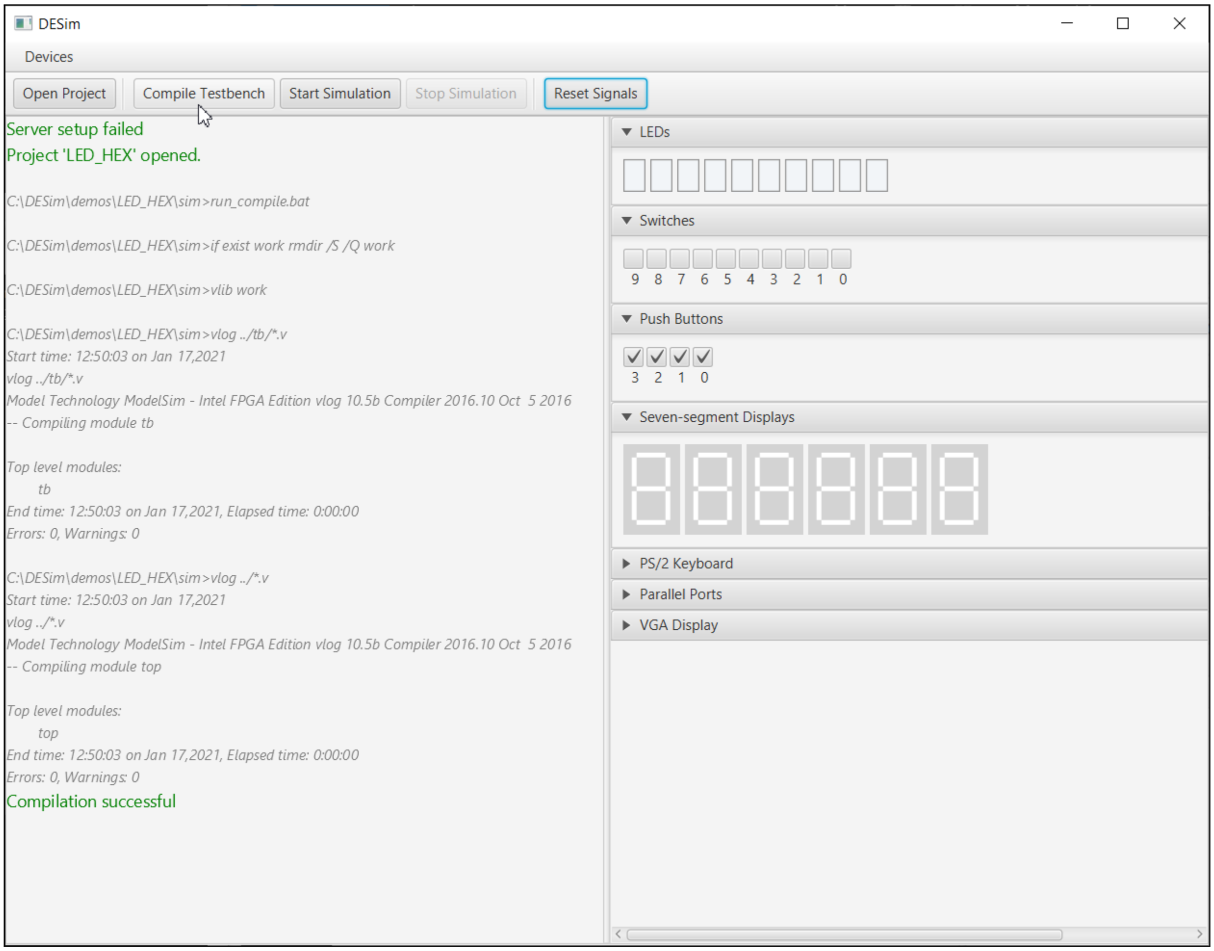
Figure 5: Opening a sample project in the demos folder.

Click the Compile Testbench command in *DESim*. As shown in Figure 6, the ModelSim simulator is executed to compile the Verilog code for the sample project, and the compilation messages that are produced by *ModelSim* are displayed in the *DESim* message pane.

If you use File Explorer to navigate to the folder for the LED_HEX project, you will be see a Verilog source-code file *top.v* and a documentation file called *Readme.txt*. These files are displayed in Figures 7 and 8.

In the *DESim* window select the Start Simulation command, which runs the ModelSim Verilog simulator. As illustrated in Figure 9 any messages produced by *ModelSim* are displayed in the message pane of the *DESim* window. To make your display look like the one in the figure, in the *DESim* GUI click on the Switch with index number 6, which causes the corresponding LED to turn red. To activate the Seven-segment Display output you have to reset the LED_HEX circuit. To do this, click on Push Button 0 to press it, and then click again to release this button. To experiment with additional features of this *demo* project follow the instructions in the *Readme.txt* file in Figure 8 and/or read through the Verilog source code in Figure 7.

You can stop the *ModelSim* simulation by selecting the Stop Simulation command in the *DESim* GUI. To close the *DESim* program click the X in the upper-right corner of the window.



```
Server setup failed
Project 'LED_HEX' opened.

C:\DESim\demos\LED_HEX\sim>run_compile.bat

C:\DESim\demos\LED_HEX\sim>if exist work rmdir /S /Q work

C:\DESim\demos\LED_HEX\sim>vlib work

C:\DESim\demos\LED_HEX\sim>vlog ../tb/*.v
Start time: 12:50:03 on Jan 17,2021
vlog ../tb/*.v
Model Technology ModelSim - Intel FPGA Edition vlog 10.5b Compiler 2016.10 Oct  5 2016
-- Compiling module tb

Top level modules:
	tb
End time: 12:50:03 on Jan 17,2021, Elapsed time: 0:00:00
Errors: 0, Warnings: 0

C:\DESim\demos\LED_HEX\sim>vlog ../*.v
Start time: 12:50:03 on Jan 17,2021
vlog ../*.v
Model Technology ModelSim - Intel FPGA Edition vlog 10.5b Compiler 2016.10 Oct  5 2016
-- Compiling module top

Top level modules:
	top
End time: 12:50:03 on Jan 17,2021, Elapsed time: 0:00:00
Errors: 0, Warnings: 0
Compilation successful
```

Figure 6: Compiling the sample project.

```
module top (CLOCK_50, SW, KEY, HEX0, HEX1, HEX2, HEX3, HEX4,
    HEX5, LEDR);

    input wire CLOCK_50;            // DE-series 50 MHz clock signal
    input wire [9:0] SW;            // DE-series switches
    input wire [3:0] KEY;           // DE-series pushbuttons

    output reg [6:0] HEX0;          // DE-series HEX displays
    output reg [6:0] HEX1;
    output reg [6:0] HEX2;
    output reg [6:0] HEX3;
    output reg [6:0] HEX4;
    output reg [6:0] HEX5;

    output wire [9:0] LEDR;         // DE-series LEDs

    assign LEDR = SW;

    reg [2:0] Addr;                 // used to select a HEX display

    always @ (posedge CLOCK_50 or negedge KEY[0])
        if (KEY[0] == 0)
            Addr <= 3'b0;
        else if (KEY[1] == 0)
            Addr <= SW[9:7];

    always @ (posedge CLOCK_50)
        case (Addr)
            3'b000:  HEX0 <= SW[6:0];
            3'b001:  HEX1 <= SW[6:0];
            3'b010:  HEX2 <= SW[6:0];
            3'b011:  HEX3 <= SW[6:0];
            3'b100:  HEX4 <= SW[6:0];
            3'b101:  HEX5 <= SW[6:0];
            default: ;
        endcase

endmodule
```

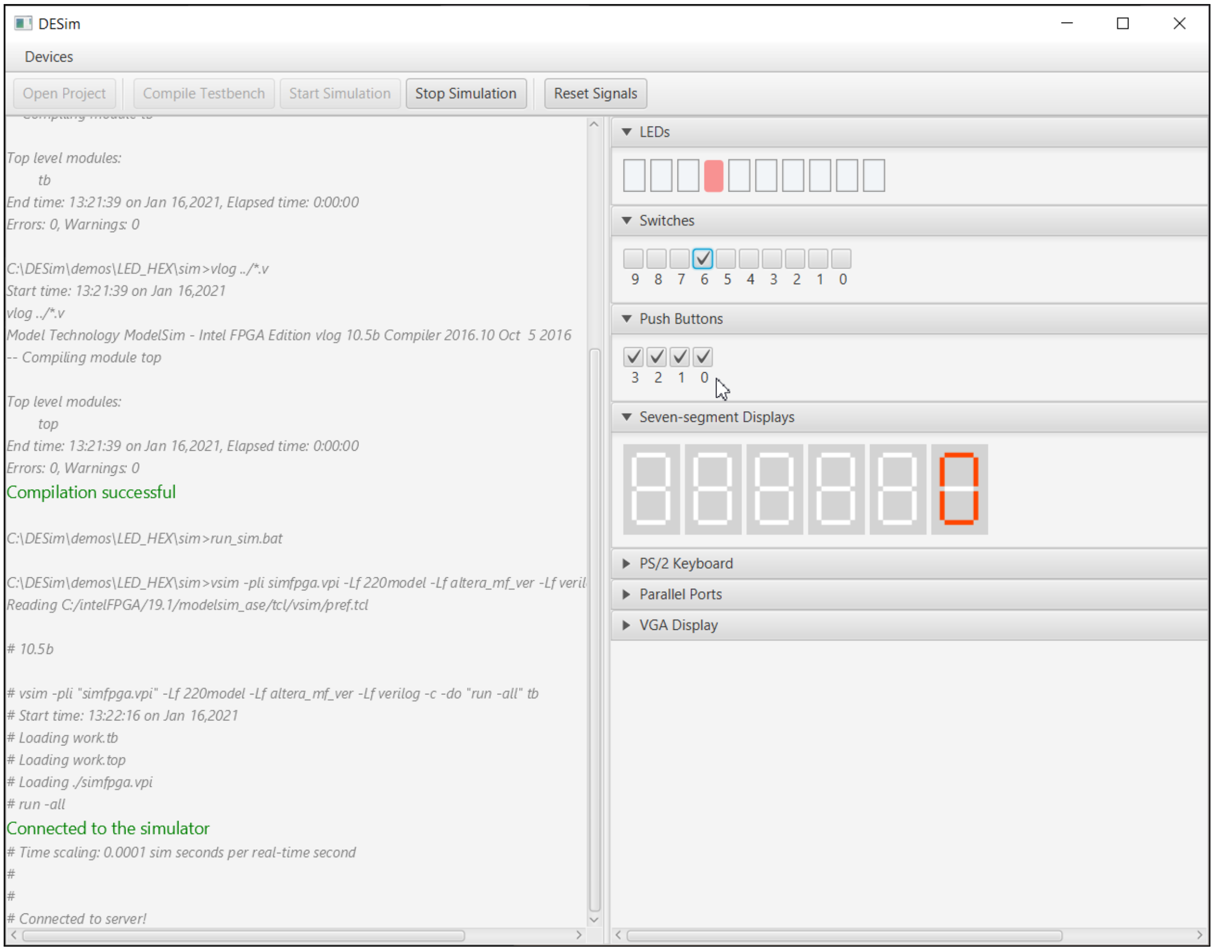
Figure 7: The Verilog source-code file *top.v* for the LED_HEX project.

```
To use this demo:

-- SW are displayed on LEDR
-- KEY[0] is the synchronous reset. It sets the HEX-display
    selector to 0.
-- KEY[1] provides the active-low enable for the HEX-display
    selector

To use:
1. First press/release KEY[0] to reset the circuit; HEX0 is
    selected
   -- HEX0 can be changed using SW[6:0]
2. Set SW[9:7] to select a different HEX display (from 0 to 5)
   -- press/release KEY[1] to store the selected HEX address
   -- the selected HEX display can now be changed using SW[6:0]
3. Set SW[9:7] to select another display, etc.
```

Figure 8: The *Readme.txt* file for the `LED_HEX` project.



```
Top level modules:
	tb
End time: 13:21:39 on Jan 16,2021, Elapsed time: 0:00:00
Errors: 0, Warnings: 0

C:\DESim\demos\LED_HEX\sim>vlog ../*.v
Start time: 13:21:39 on Jan 16,2021
vlog ../*.v
Model Technology ModelSim - Intel FPGA Edition vlog 10.5b Compiler 2016.10 Oct  5 2016
-- Compiling module top

Top level modules:
	top
End time: 13:21:39 on Jan 16,2021, Elapsed time: 0:00:00
Errors: 0, Warnings: 0
Compilation successful

C:\DESim\demos\LED_HEX\sim>run_sim.bat

C:\DESim\demos\LED_HEX\sim>vsim -pli simfpga.vpi -Lf 220model -Lf altera_mf_ver -Lf veril
Reading C:/intelFPGA/19.1/modelsim_ase/tcl/vsim/pref.tcl

# 10.5b

# vsim -pli "simfpga.vpi" -Lf 220model -Lf altera_mf_ver -Lf verilog -c -do "run -all" tb
# Start time: 13:22:16 on Jan 16,2021
# Loading work.tb
# Loading work.top
# Loading ./simfpga.vpi
# run -all
Connected to the simulator
# Time scaling: 0.0001 sim seconds per real-time second
#
#
# Connected to server!
```

Figure 9: Simulating the sample project.