**DE0-V Quick Start Guide**

1. If using an existing QAR file (e.g. Assignment1\_frame.qar), you can skip steps 2­5.

2. Create a new project

● When you start Quartus, select File → New Project Wizard in the menu.

● Select a working directory (or just use the default Quartus directory).

● Give the project and its top­level entity a name (e.g. Assignment1), then click Next.

● Do not add any files (Next).

● Select Cyclone V Family (Device family ­> Family ­> Cyclone V (E/GX/GT/SX/SE/ST))

● Type 5CEBA4F23C7 on “Name Filter” to select our FPGA device (Cyclone V family, 5CEBA4F23C7N device)

● Click Finish.

3. Take care of the pin assignments

* Download the pin assignment file at T-squre resource (CS3220A-Resources/de0-cv) : DE0\_CV.qsf

● In Quartus, go to Assignments → Import Assignments and import the *DE0\_CV.qsf* file.

● Go to Assignments → Device, then click on the “Device and Pin Options” button. In the dialog that pops up, go to the “Unused Pins” category, change “Reserve all unused pins” to “As input tri­stated”.

4. Enable the design assistant

● Go to Assignments → Settings, then look at the “Category” list on the left side of the dialog.

● Select the “Design Assistant” category, check the “Run Design Assistant during compilation” box.

5. Create a Verilog file with your code. Go to File → New, select “Verilog HDL file” (from “Design Files”) in the dialog box.

Write something in there and when you try to save there will be a dialog box, choose the file name (e.g. SwitchToLedR.v), make sure the “Add file to current project” is checked, and find a directory where you want the file saved (e.g. the project directory) but in any case remember where the file is saved (you will need to find it so you can upload it on T­Square).

*[Test Code] SwitchToLedR.v*

module SwitchToLedR (SW, LEDR);

input[9:0] SW; output[9:0] LEDR; assign LEDR = SW;

endmodule

6. Write Verilog code for a circuit you want, save it, and compile it. To compile, click on the “Start Compilation” button on the toolbar (the button has what looks like a purple “Play” button from a music player), or select Processing → Start Compilation in the menu.

7. Connect the board (using the USB cable). When you connect the board, the system will ask for the driver if the driver was not successfully installed. Tell the system to look in “C:\altera\14.0\quartus\drivers\usb­blaster” for the driver (this path may be different if you installed Quartus in another directory).

8. Turn the board on and program it:

● To turn the board on, connect the power cord to the power jack on the board and push the red button. The board should start doing its thing – all the FPGA user HEXs and LEDs will be flashing.

● To program the board, click the “Programmer” button in Quartus (one of the last buttons on the toolbar, looks like some lines entering a blue chip), or select Tools → Programmer in the menu.

● When the programmer dialog opens, make sure the “Hardware Setup” says “USB­Blaster [USB­0]” (if it doesn’t, click the “Hardware Setup” button and select USB­Blaster [USB­0] there) and that the mode is set to “JTAG”. The file should be SwitchToLedR.sof (or whatever the design name is, with a .sof extension). You can add this file by clicking “Add File…” and selecting a sof file (e.g., SwitchToLedR.sof) under the output\_files directory.

● Make sure that the checkbox for “Program/Configure” is marked.

● Click the “Start” button in the programmer dialog. That should do it – the board’s behavior should change from its default (just flashing) to whatever you programmed it to do. In our SwitchToLedR example, the LED will be lit when the corresponding switch is ON.

9. Run ModelSim from Quartus

A. Implement a testbench file for your design.

*[Test Code] TestSwitchToLedR.v*

`timescale 1ns / 1ps

module TestSwitchToLedR (); reg [9:0] sw; wire[9:0] ledr;