CSEE 4840
Embedded System Design Lab 1: Using the FPGA

Stephen A. Edwards, Columbia University

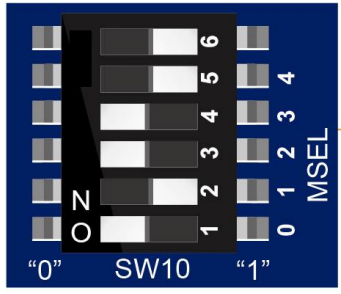
Spring 2020

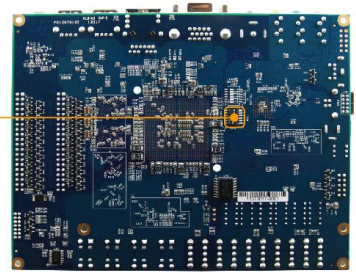
Learn how to compile and download an FPGA-only project to the DE1-SoC board. You will add functionality that allows a user to display and edit the contents of a 16 × 8 bit RAM.

For this lab, use Quartus 18.1 the FPGA design software produced by Intel for their chips. You can find it on the workstations in 1235 Mudd (on which you should have an account if you registered for the class), you can run it on your laptop using the VM image we will distribute, or you can download the free "Lite" version from the Intel website.

1 Set the FPGA Configuration Mode

A microscopic set of switches on the back of the board controls the source of the FPGA's configuration information. For this lab, set it to the "Active Serial" mode as shown on the right.





For this lab, we will be configuring the FPGA with a JTAG interface through USB; the Active Serial mode makes the board start up in a factory demonstration mode.

Mode	6	5	4	3	2	1
Active Serial (Default; use for this lab)	Off	Off	On	On	Off	On
FPPx16 (from SD card; later labs will use this)	Off	On	On	On	On	On
FPPx32 (from Linux)	Off	On	Off	On	Off	On

2 Download and Unpack the Lab 1 files

Download *lab1.tar.gz* from the class website and extract it by typing *tar zxf lab1.tar.gz*. This will create a *lab1* directory containing the files listed below.

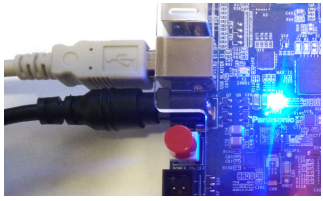
Name	Contents
lab1.sv	Skeleton lab 1 code: the memory, a seven-segment decoder, the controller, and a top-level module that connects these. **Your assignment: modify this file.**
de1-soc-project.tcl	A Tcl script that creates the lab1 project files. Includes pin assignments.
Makefile	Commands for creating the project files, compiling the project, building the *lab1.tar.gz* file, and cleaning up unneeded files.

3 Compile and Download the Project Via the Command-Line

Enter the *lab1* directory and run *make lab1.qpf* to create the project from the *de1-soc-project.tcl* script. This should report "Info (23030): Evaluation of Tcl script de1-soc-project.tcl was successful," but if it complains "quartus_sh: Command not found," make sure your PATH variable includes the directory for the Quartus binaries (on the 1235 Mudd machines, */etc/profile.d/quartus.sh* does so when you log in). Running this script creates *lab1.qpf* (the main project file), *lab1.qsf* (settings, including files and pins), and *lab1.sdc* (clock constraints).

Once the project is created, you can compile it from the command-line with *make output_files/lab1.sof*. This reads the lab1 files (including lab1.sv) and ultimately produces the *lab1.sof* (SRAM oobject) file, which is the configuration file that is downloaded to the FPGA to actually run your project. This takes a while, and will report and handful of warnings, but should eventually report "Info (293000): Quartus Prime Full Compilation was successful."

Connect the DE1-SoC board to your workstation. Connect the +12V power supply to the board near the red power button, connect a USB cable to the "USB Blaster" port on the board (next to the power button) and to your workstation, and power on the board with the red button.



Once your project is compiled, download the .sof file to the DE1-SoC board by running *make program*. A variety of things can go wrong. If you get "Error (213013): Programming hardware cable not detected," check your board's power and USB connection to the your workstation.

When powered and connected, the board should appear as a USB device. Under Linux, *lsusb* should report the board as `09fb:6810 Altera` or `09fb:6010 Altera`.

The Quartus software must also have permission to access the port. Run *jtagconfig*. With the board connected and powered on, it should report

```
$ jtagconfig
1) DE-SoC [1-1.5.2.2]
  4BA00477   SOCVHPS
  02D120DD   5CSE(BA5|MA5)/5CSTFD5D5/..
```

If *lsusb* "sees" the board but *jtagconfig* reports "No JTAG hardware available," there is a permission problem, which can be resolved by telling udev to make the board accessible to everybody. As root, create the file */etc/udev/rules.d/51-altera.rules* containing

```
ATTR{idVendor}=="09fb", ATTR{idProduct}=="6010", MODE="0666"
ATTR{idVendor}=="09fb", ATTR{idProduct}=="6810", MODE="0666"
```

4 Compile and Download the Project Via the GUI

The project can also be compiled and downloaded via the Quartus GUI. Start from a directory with a clean unpack of lab1.tar.gz (or run *make clean*), then start Quartus by typing *quartus*.

Create the project files by running a Tcl script. Open the Tcl console window with View→Utility Windows…→Tcl Console. Type *source de1-soc-project.tcl* in the Quartus Prime Tcl Console window. This will create the project files lab1.qpf, lab1.qsf, and lab1.sdc.

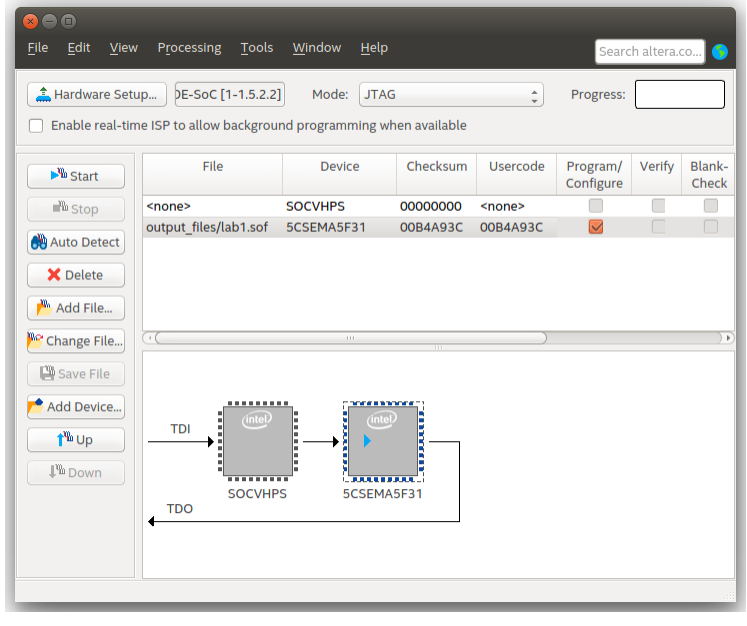
Open the *lab1.qpf* project with File→Open Project….

Compile the project with Processing→Start Compilation. This will take a while and should eventually report *Quartus Prime Full Compilation was successful*. There should be no errors, but there may be warnings.

Download the configuration to the FPGA. Select Tools→Programmer. If "No Hardware" appears, connect the board to your workstations via USB and power it on (see the previous section), then click on "Hardware Setup…" You should see "DE-SoC" under "Available hardware items." Select "De-SoC[⋯]" under "Currently selected hardware" and click "Close."

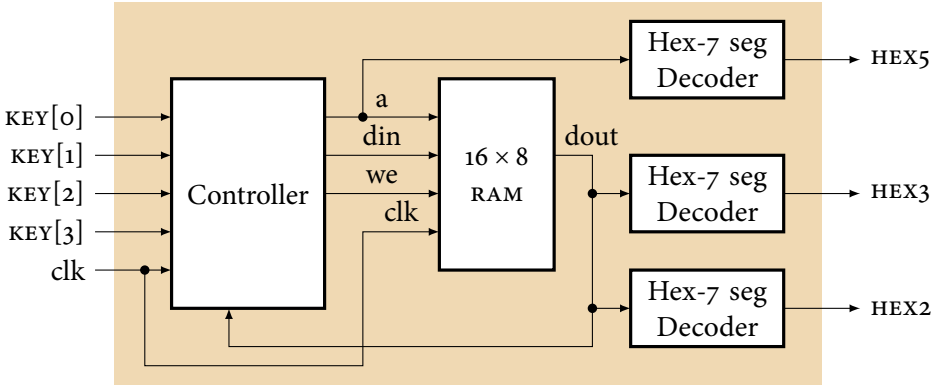
Set up the JTAG chain by clicking on "Auto Detect" and select "5CSEMA5." Answer "yes" if it asks to update the programmer's device list.

Tell it to configure the FPGA with the *lab1.sof* file by clicking on the "5CSEMA5" device then "Change File" and choose the *lab1.sof* file in the *output_files* directory. Mark the "Program/-Configure" checkbox on the 5CSEMA5F31 line. It should look like the image on the right.

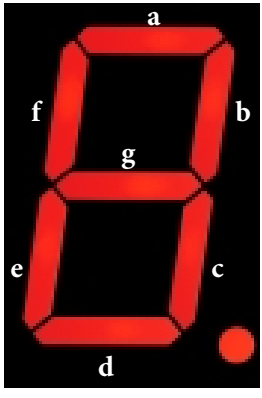


Finally, click on "Start" to program the FPGA. This should quickly report "100% (Successful)" on the programmer.

5 The Lab 1 Design



Implement a memory display and modification circuit according to the block diagram above. The circuit should always display the address and contents of one of 16 byte-wide memory locations.



Have KEY[3] and KEY[2] increment and decrement the address and KEY[1] and KEY[0] modify its contents. The KEY inputs are active-low signals from the four pushbuttons on the right side of the DE1-SoC board. See the *DE1-SoC User Manual* for details.

The six 7-segment LED displays are controlled by the six HEX signals. The segments are active-low: a "0" turns them on. HEX0[0] controls the "a" segment of the rightmost digit, HEX0[1] is the "b" segment of the rightmost digit, …, and HEX5[6] is the "g" segment of the leftmost digit.

Modify the code in *lab1.sv* to implement your lab. Put your names and unis in the comments.

Demonstrate your working lab1 to a TA during his/er office hours and submit your modified lab1.sv file on Courseworks.