Instructions: OpenCL Compilation and Setup

Instruction Overview: These instructions detail how to prepare the DE1-SoC to run OpenCL projects and how to create the executables for the boards.

Hardware Setup:

1. On the bottom of the DE1-SoC, there is a set of switches which dictate how the FPGA is programmed. Figure 1 shows the location of these switches. For the default JTAG configuration, these switches should be set "**001101**" from highest to lowest order. For OpenCL, these switches should be set to "**110101**"

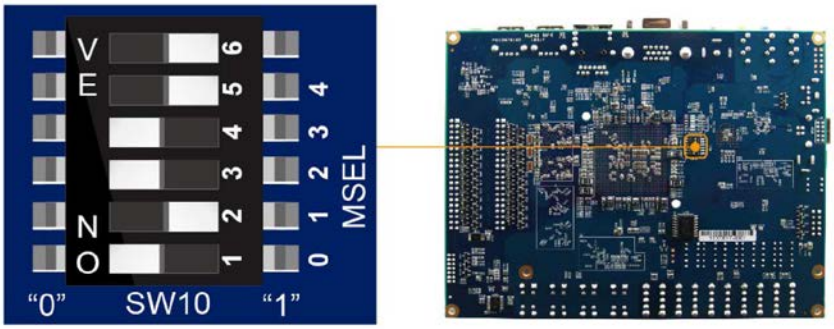

Figure 1: SW10 Location with Default Configuration for JTAG Programming

2. From here, you must have a microSD card with Altera's Linux Console image built on it. Upon powering on the board, you should see the red LEDs begin a binary counting sequence. All of the SD cards in lab should have working copies of this image already installed. If this does not seem to work for you at a particular board, contact a TA.

3. Verify the UART cable is connected to the host computer (mini-USB to USB Cable located near the SD Card slot).

Software Setup:

1. Open a terminal and type "minicom". You should see an "Initializing Modem" message followed by a few messages from minicom. If this does not connect, or states that permission is denied or "ttyUSB0 is unavailable", then disconnect and reconnect the UART cable on the FPGA board.

2. Power Cycle the FPGA. You should see boot messages appear from Terasic, don't press anything until prompted for login.

3. When prompted for "socfpga login:", type **root** and you should be logged in.

4. Next, you **MUST** type "source ./init_opencl.sh" or you will not be able to run your OpenCL code.

5. You can now browse to the location of your executable and test.

Programming and Compiling Your Kernel:

1. You will receive a folder structure containing a bunch of files, including a skeleton for your OpenCL Kernel. The kernel will be located at "**OpenCL/device/PROJECT.cl**" inside the root folder. **DO NOT MODIFY THE FOLDER STRUCTURE.**

2. Edit the *.cl file with your favorite text editor.

3. Browse to the emulation folder if you would like to test your project before spending time compiling. Here, modify the "run.sh" file, if necessary, to change running parameters if you prefer a different test case. Simply type "./run.sh" to **make** and **execute** your program for emulation. **DO NOT EXECUTE MAKE OUTSIDE OF THE RUN SCRIPT. IT WILL FAIL.**

4. Once you are ready to compile, you need to open a terminal window and browse to this location and type "make".

    a. It may be useful to note that the Makefile includes a "resource" option! You can use this first to verify your design will fit on the FPGA before wasting time compiling.

Programming and Compiling the Host Code:

1. The host code will be located at "**OpenCL/host/src/main.cpp**" inside the root folder. **DO NOT MODIFY THE FOLDER STRUCTURE.**

2. Edit the *.cpp file with your favorite text editor.

3. Once you are ready to compile, you need to open a terminal window and browse to **OpenCL/** and type "make". This will automatically run the host code with the ARM cross compiler.

Copying Files to the SD Card:

1. Once you have compiled your kernel, you need to copy the **PROJECT.aocx** and **PROJECT** binaries (both located in "OpenCL /bin" to the SD Card. To do this, you will need a micro-SD card reader. There are two versions in lab, a USB Reader (only 1!) and many SD card converters with micro-SD card slots. The USB reader will work on the Linux machines in lab, but may not always be available. The SD card converter will require you use your laptop (if equipped with an SD card slot), and you will need to copy the binaries to your computer to use this method. The readers are located in the second room of lab on a shelf. *Please keep in mind the format of these SD Cards is ext4. Neither Windows nor Apple computers recognize this format without Third-Party software, which is easily found online.*

2. Once you have the micro-SD card mounted, you can copy files into the "**home/root/PROJECT**" folder. If the **PROJECT** binary exists, you do not need to copy this as well (this is the host code which you should not have modified).

3. Anytime you are copying files to the SD card, **MAKE SURE TO EJECT SECURELY BEFORE REMOVING.** SD cards have a high read/write time, and some file transfers may appear complete, but the buffer is still writing data to the card. Removing early may corrupt files

on the card and require the card be wiped. If using Linux commands in terminal, type "sync" and wait for this to finish. If using the GUI to copy, be patient and wait until the OS tells you it is safe to eject.

4. You can now re-insert the card into the FPGA and test the program execution by browsing to the **PROJECT** folder in the file structure and running **"./PROJECT PARAMETERS"**.

Obtaining an SD Card:

You will be held responsible for obtaining your own **Micro SD Card**. The minimum required size is 4GB, but 8GB may be more suitable for larger data purposes. Make sure any purchased card is the ***MICRO*** size! This is the only size that is compatible with the FPGAs in lab. If you cannot obtain a Micro SD Card, then you can contact Dr. Smith and ask to borrow one. If you choose to borrow an SD card, you will be held responsible for returning this by the end of the semester. Failure to do so will result in an F for the course.

Please contact a TA if any of these instructions are incorrect, confusing, or if any issues arise when attempting to follow them.

Brad Green: wbgreen@g.clemson.edu Nilim Sarma: nsarma@g.clemson.edu

Sufeng Niu: sniu@g.clemson.edu