Lab 0: Warmup

ECSE 324 - Computer Organization

Winter 2018

Getting ready for the labs

Find a lab partner and register as a group on myCourses. You must be signed up in a group by **5 PM Friday, January 19th**.

Once you have a lab partner, go to the Trottier 4th floor equipment desk (4140) to get a DE1-SoC board.

During the first week of labs, the lab rooms will be open in Trottier. You can go to the labs to meet potential lab partners. You can also find a lab partner using the myCourses discussion board.

Where to find the course software

We will use two primary software tools for this course: Quartus Prime 16.1 Lite Edition (only for Lab 0) and the Intel FPGA Monitor Program. Both tools are available on the lab computers in Trottier. There is a desktop icon for each tool.

You can also download the tools and use them on your own computer if you wish. (Note: the software uses a lot of disk space.)

- Quartus II version 16.1 can be found here: http://dl.altera.com/16.1/?edition=lite
- The Intel FPGA Monitor Program can be found here (choose version 16.1 and "University Program Installer"): https://www.altera.com/support/training/university/materials-software.html

Main tasks in this lab

This lab will help identify any software or hardware issues you might face before the actual labs begin. It is very important that you complete this lab!

There are two main tasks in this lab:

- Flashing the computer system onto the board.
- Running the demo program.

Flashing the Computer System to your board

You will be working with the DE1-SoC Computer System for the labs. Part of the Computer System is a hard ARM processor, and the other part is implemented as soft components on the FPGA. It will make your life slightly easier to save the configuration for the FPGA part of the system in the board's

nonvolatile flash memory so that you don't need to reload it every time you turn the board on.

Use the following instructions to flash the Computer System to your board. A TA will be present in the lab rooms during the first week of labs. If you have trouble getting the board set up, you can ask a TA for help. (The flashing process may not work for older boards.)

Before Programming Begins

The FPGA should be set to AS x1 mode i.e. MSEL[4..0] = "10010" to use the quad Flash as a FPGA configuration device.

Convert .SOF File to .JIC File

1. Open 'Quartus Prime 16.1 Lite Edition' from the desktop icon. Select File->Convert Programming Files, as shown in Figure 1. The Convert programming file window will now open.

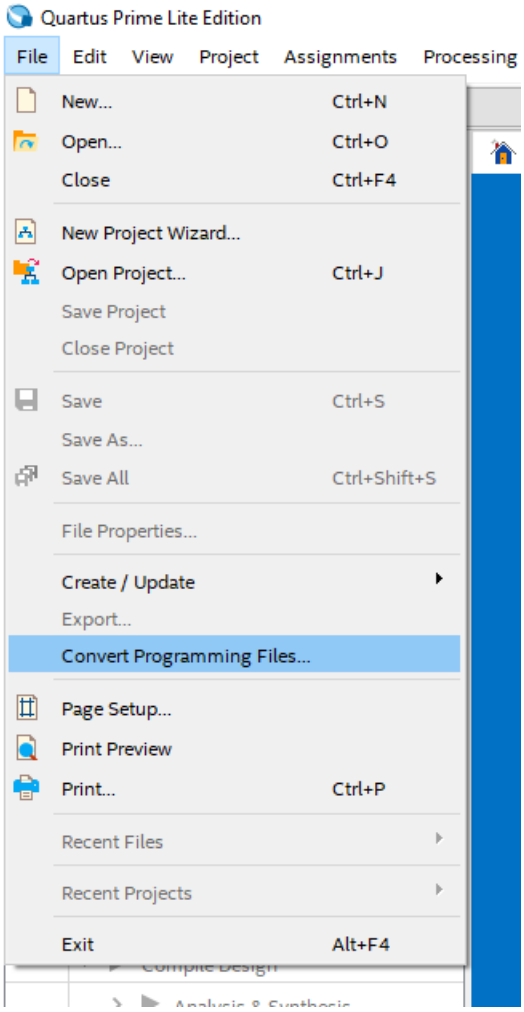

Figure 1: Convert SOF to JIC - Step 1

2. Set the 'Programming file type' to 'JTAG Indirect Configuration File (.jic)', the 'Configuration device' to 'EPCS128', and set the 'File name' field to:
C:\intelFPGA_lite\16.1\University_Program\Computer_Systems\DE1-SoC\DE1-SoC_Computer\verilog\DE1_SoC_Computer.jic

Verify that the settings so far match those shown Figure 2.

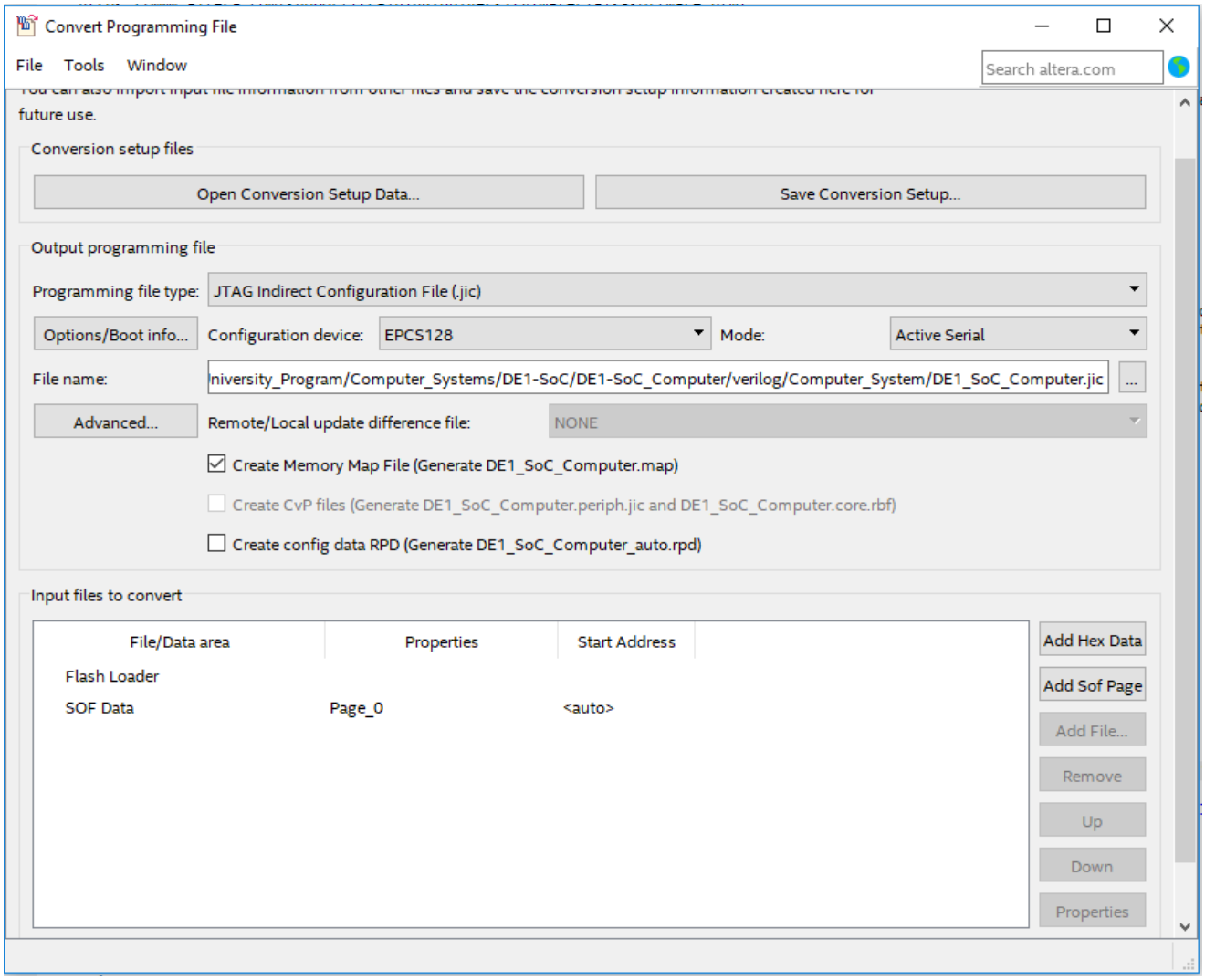

Figure 2: Convert SOF to JIC - Step 2

3. Next, under ‘Input files to convert’ click on ‘SOF data’, and then click ‘Add File’ as shown in Figure 3.

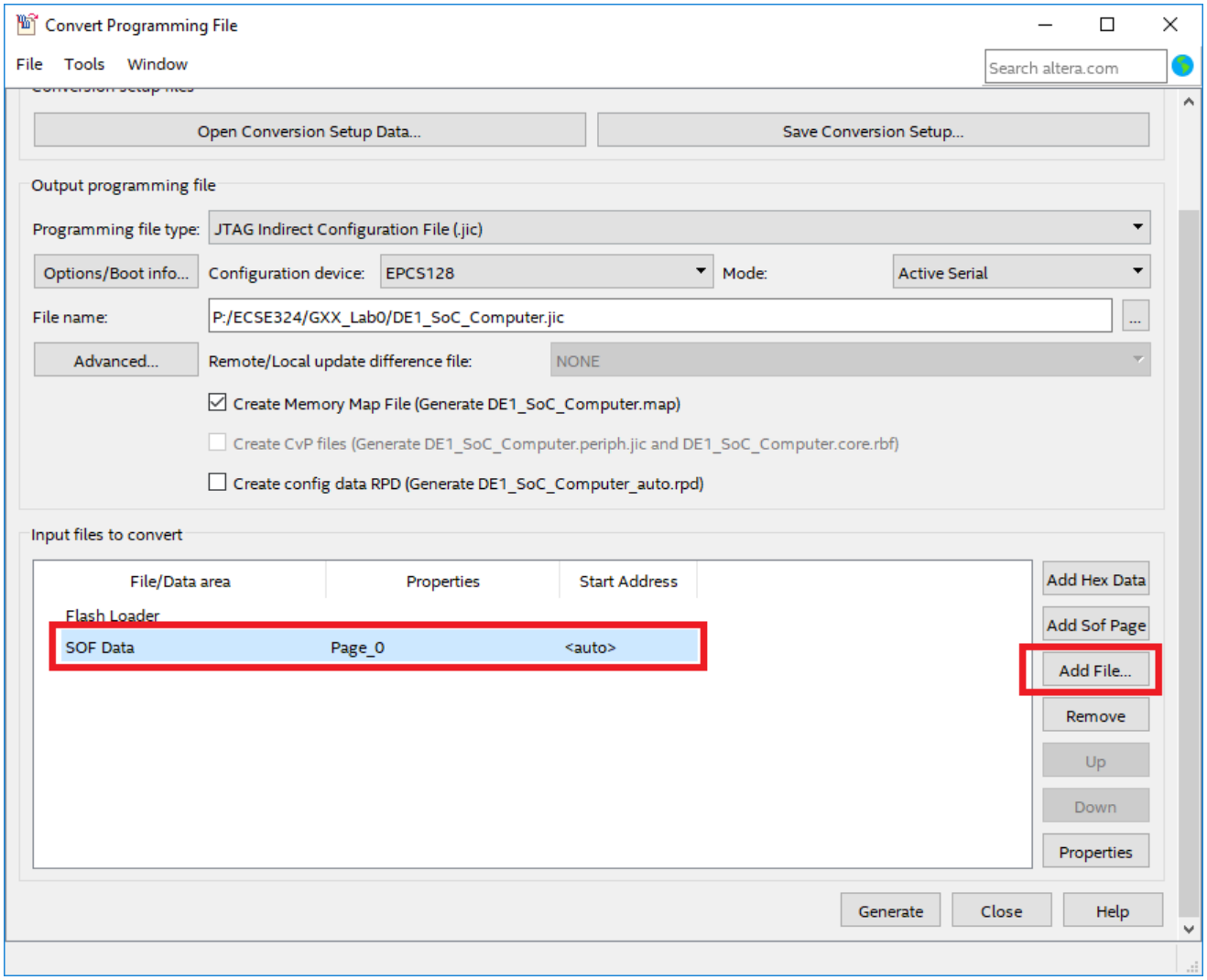

Figure 3: Convert SOF to JIC - Step 3

4. A window to select the file will open. As shown in Figure 4, the .sof file to be selected is:

C:\intelFPGA_lite\16.1\University_Program\Computer_Systems\DE1-SoC\DE1-SoC_Computer\verilog\DE1_SoC_Computer.sof

NOTE: If you cannot find the intelFPGA_lite folder in the C: drive, try enabling hidden folders in the view options.

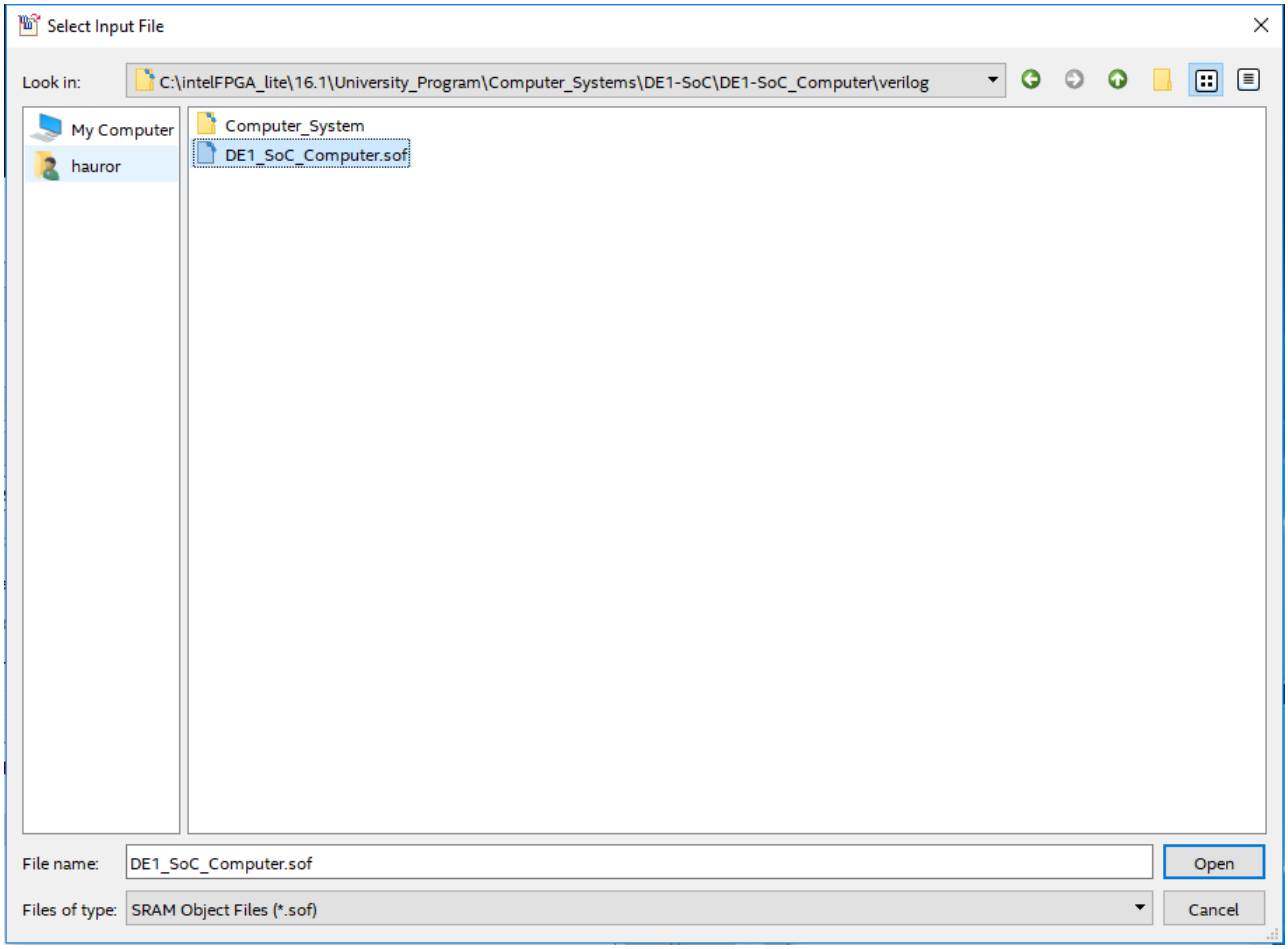

Figure 4: Convert SOF to JIC - Step 4

5. Next, under 'Input files to convert' click on 'Flash Loader', and then click 'Add Device' as shown in Figure 5.

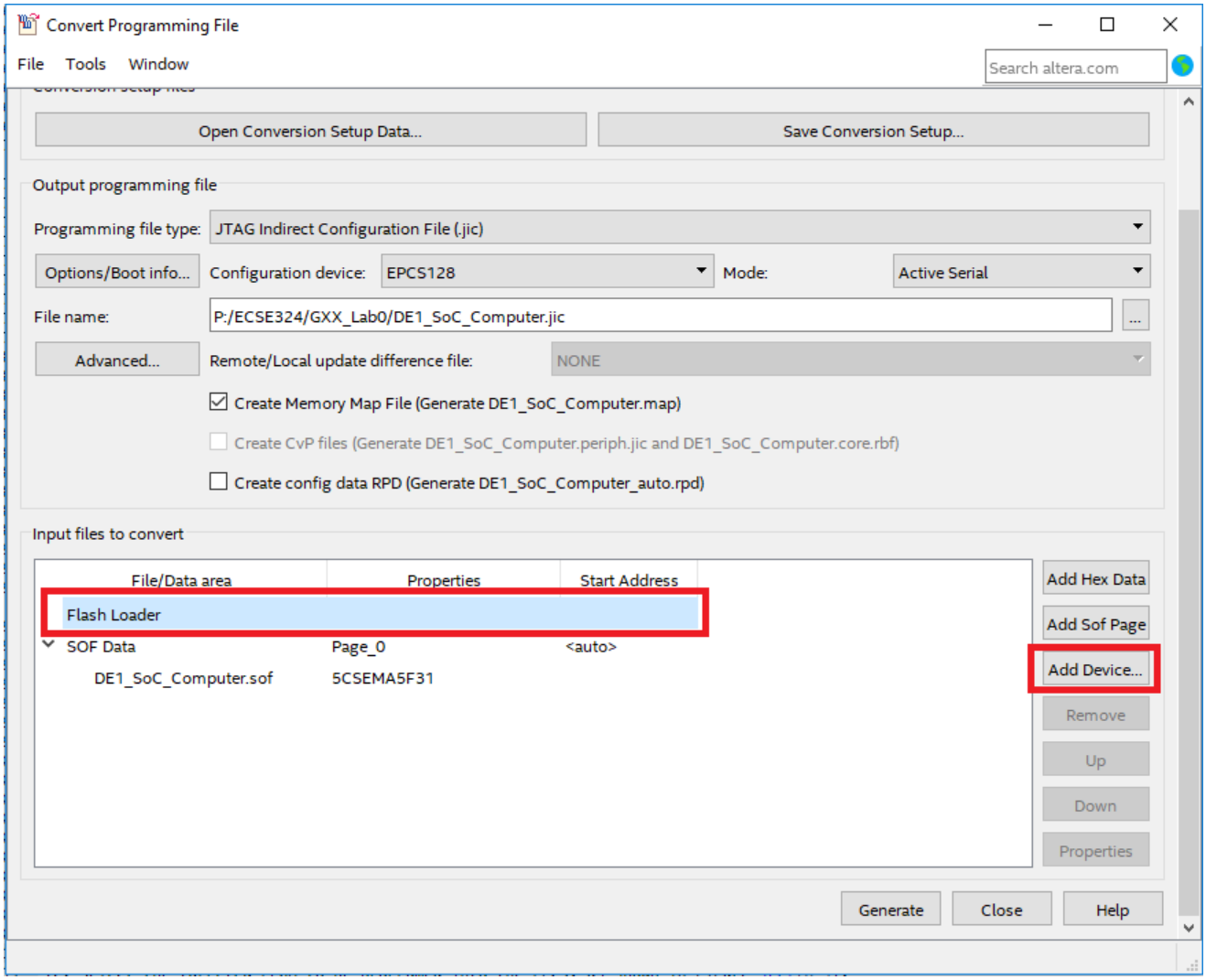

Figure 5: Convert SOF to JIC - Step 5

6. A window to select the device will open. Under 'Device family' select 'Cyclone V', and under 'Device name' select '5CSEMA5', as shown in Figure 6.

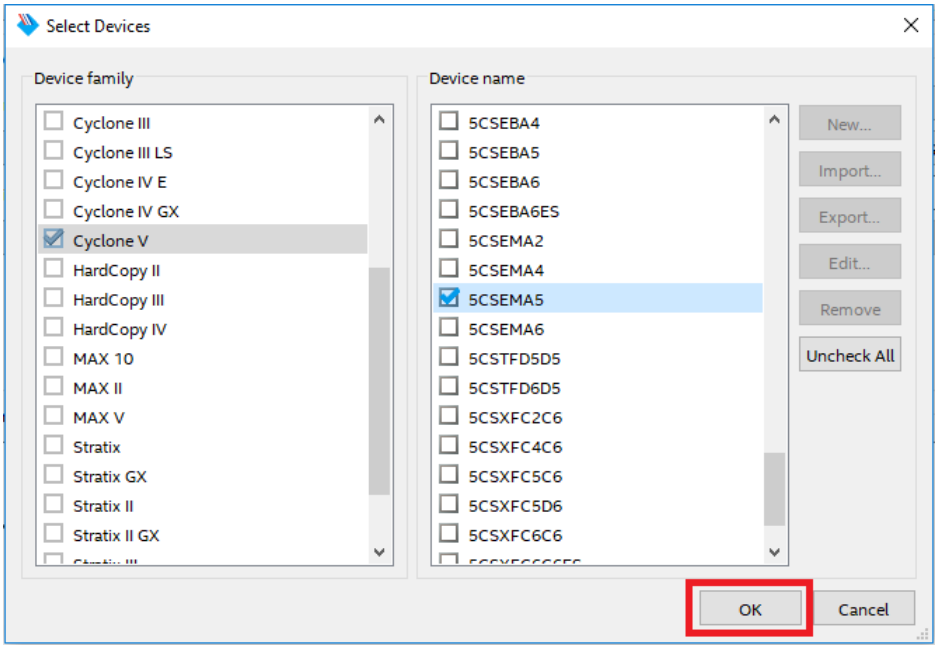

Figure 6: Convert SOF to JIC - Step 6

7. Verify that the settings match those shown in Figure 7, and click ‘Generate’. After the .jic file is successfully generated, close the ‘Convert Programming File’ window.

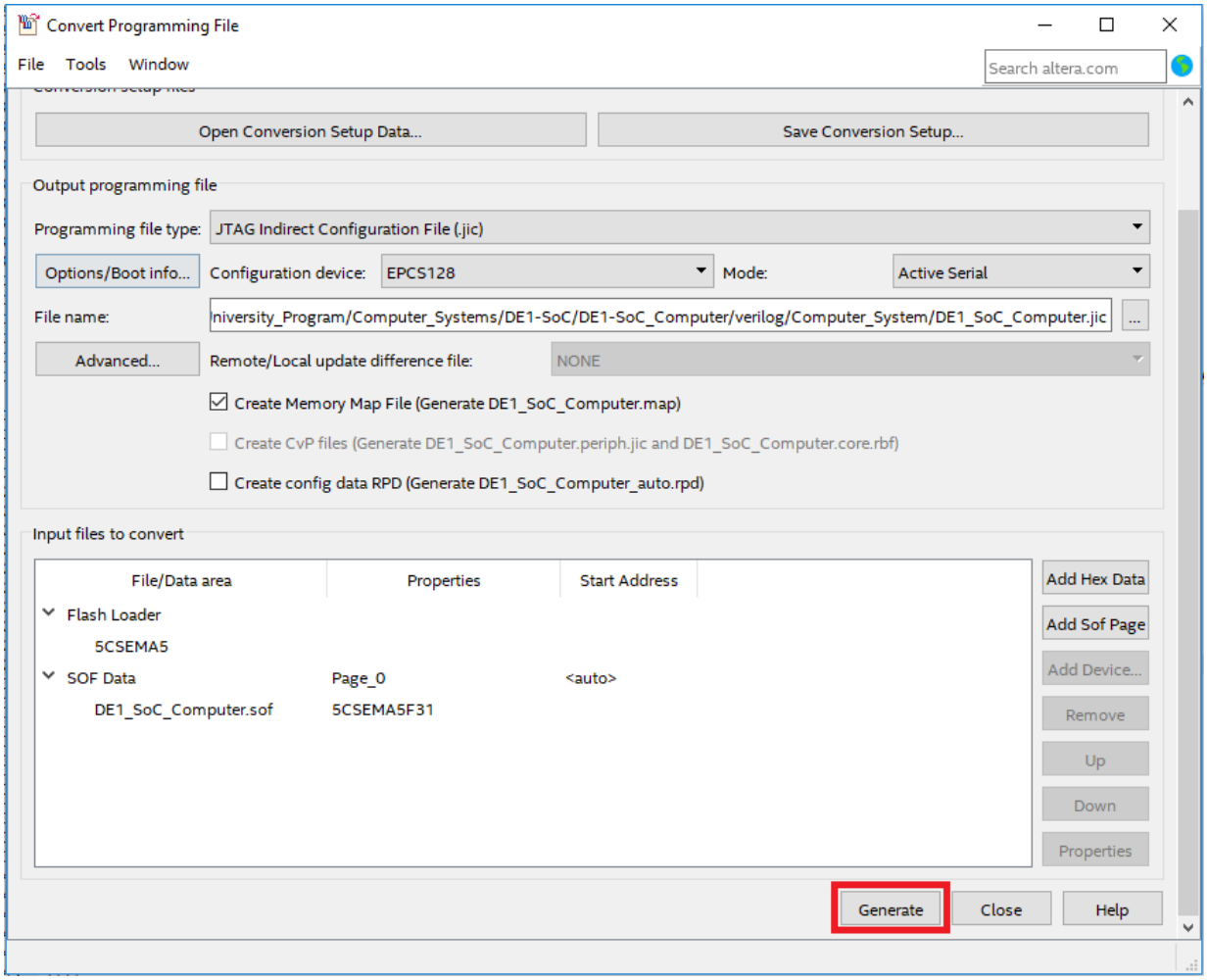

Figure 7: Convert SOF to JIC - Step 7

Write JIC File into the EPCS Device

When the conversion of SOF-to-JIC file is complete, please follow the steps below to program the EPCS device with the .jic file created.

1. Verify that MSEL[4..0] = “10010”. The MSEL switches are found on the underside of the DE1-SoC board. Please power-off and unplug the board before changing the state of any switches. After the MSEL switches are set appropriately, connect the DE1-SoC board to the computer via the USB Blaster cable and power-on the board.

2. In Quartus, select Tools->Programmer, and the Chain.cdf window will appear.

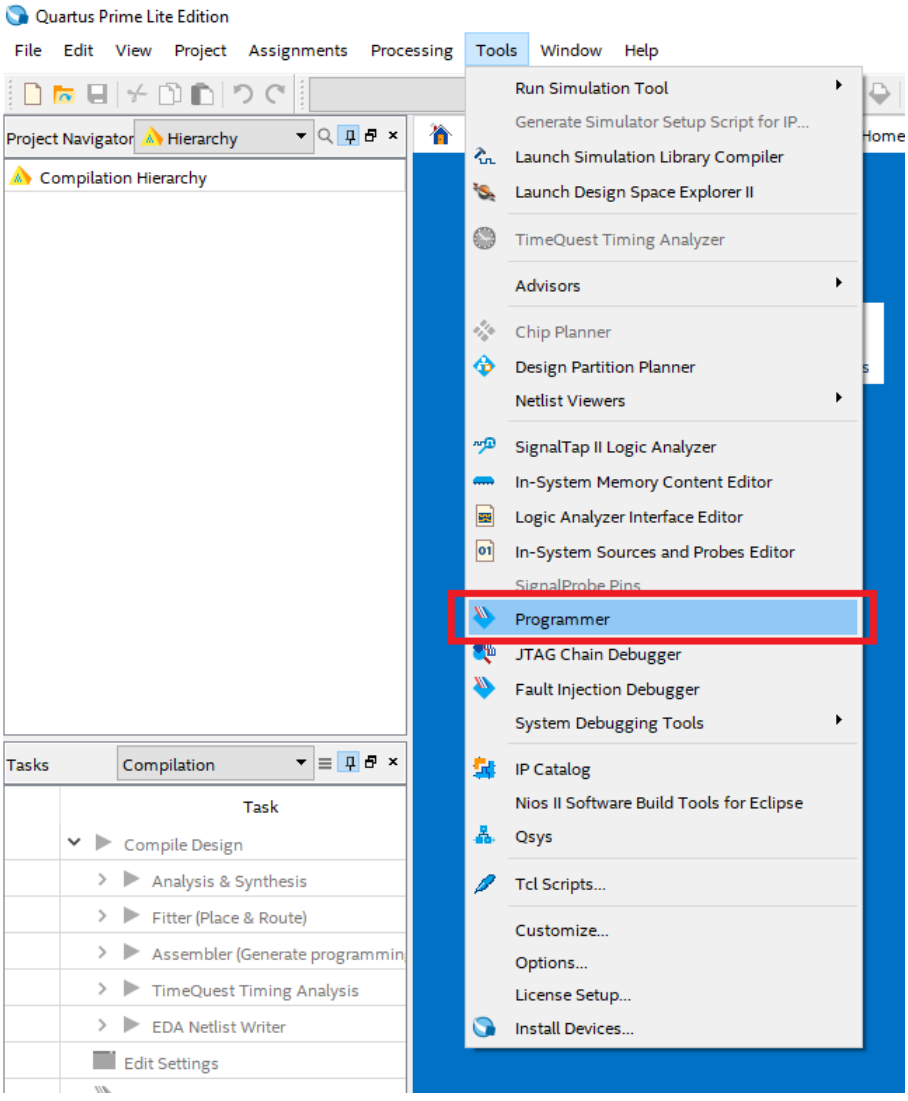

Figure 8: Program the JIC file - Step 2

3. Click Auto Detect and then select the correct device (5CSEMA5), as shown in Figure 9. Both FPGA device and HPS should be detected.

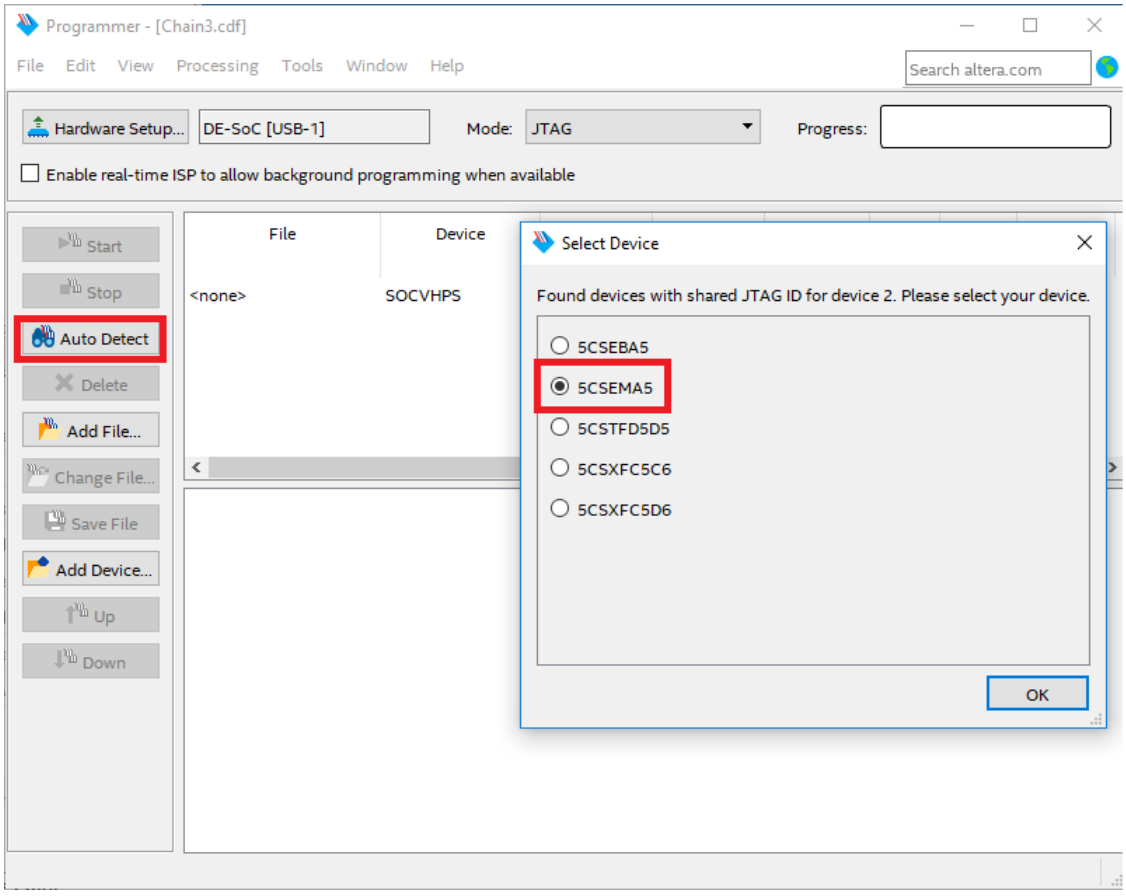

Figure 9: Program the JIC file - Step 3

4. Next, double-click the FPGA device (5CSEMA5), and from the window that opens add the .jic file created in the previous section. Refer to Figure 10.

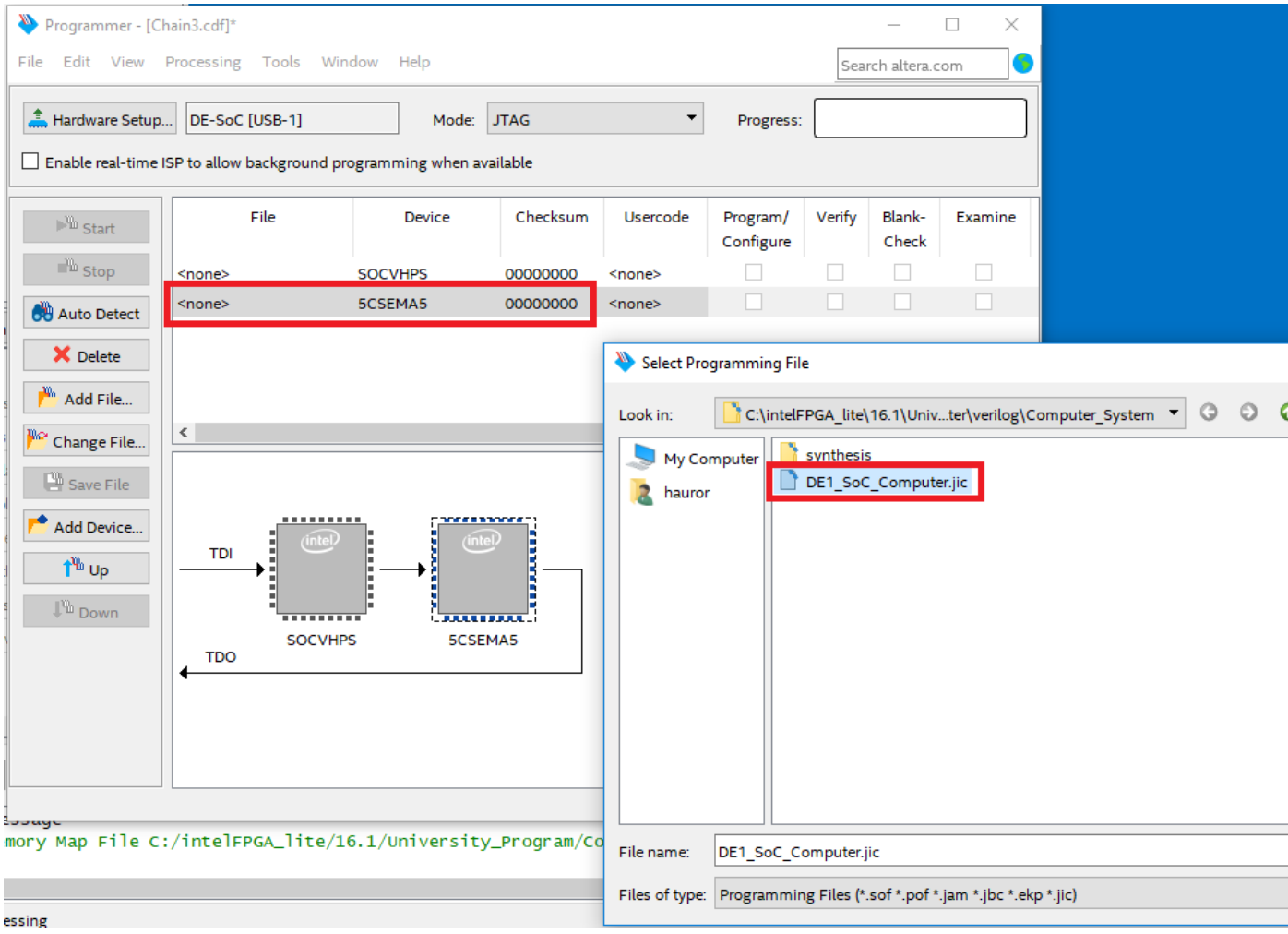

Figure 10: Program the JIC file - Step 4

5. Finally, check the 'Program/configure' box beside the 'EPCS128' device, and then click 'Start'. Refer to Figure 11.

After the .jic file has ben programmed, you may close the programmer window and exit Quartus (There is no need to save changes to the existing chain in case a dialogue box appears asking to do so). Lastly, switch the DE1-SoC board off and on. The Altera demo program from before should not be running on the board anymore.

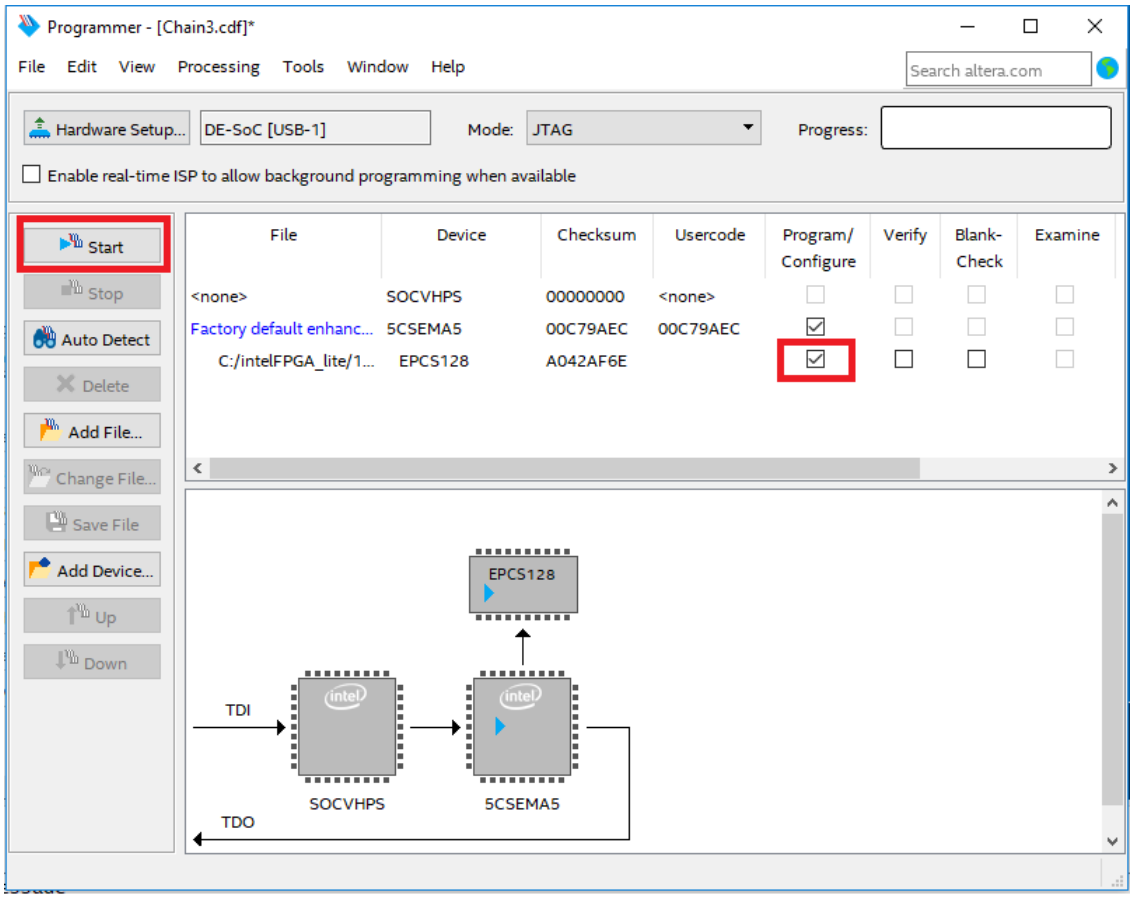

Figure 11: Program the JIC file - Step 5

Running the demo program

Now that the SOF file has been flashed onto the board, use the Altera Monitor program to create and run a sample program.

1. Open the 'Intel FPGA Monitor Program 16.1' from the desktop icon (if asked to enable a C debugging beta feature, click 'Yes'), and select File->New Project.

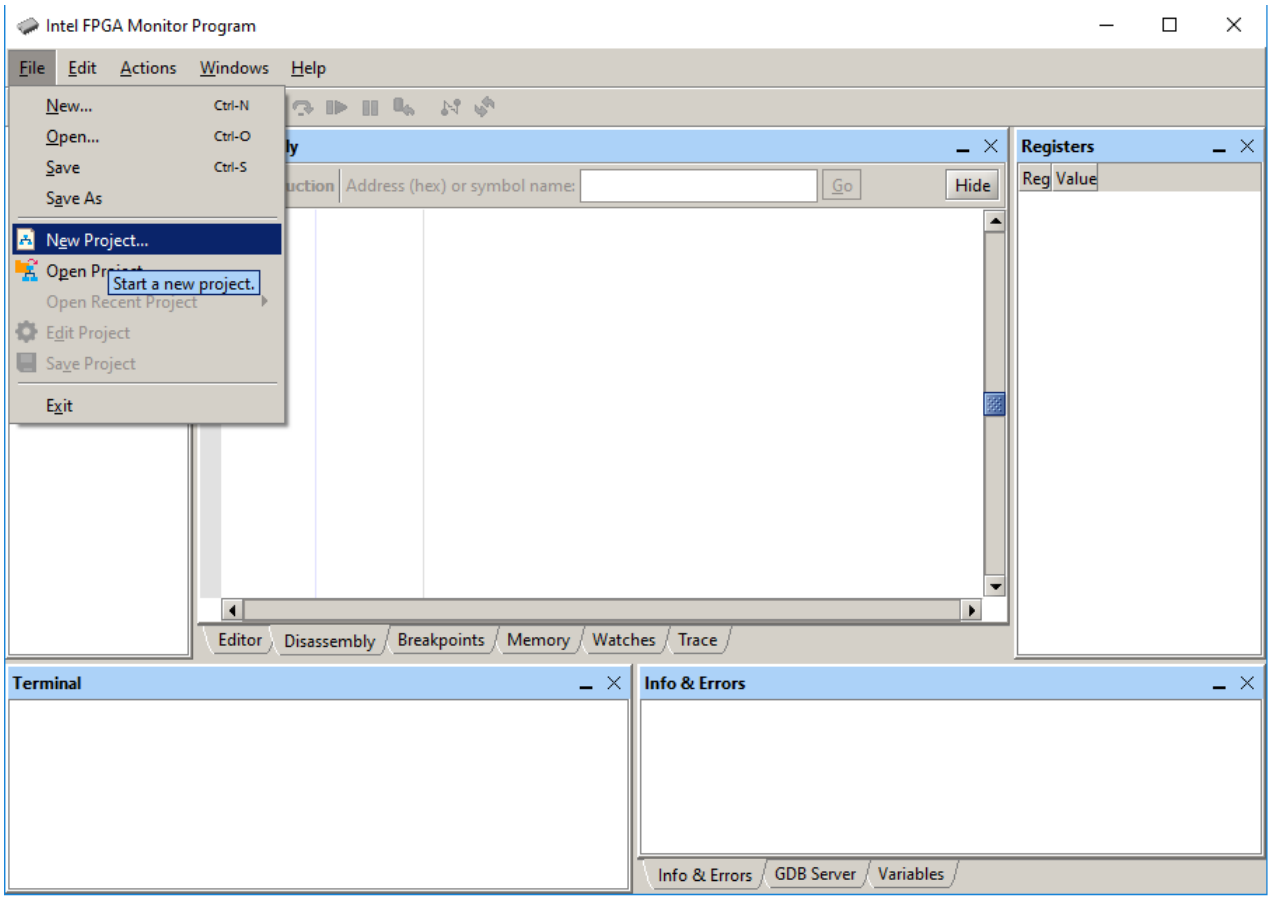

Figure 12: Running the demo program - Step 1

2. Set the project directory as desired, but only within your network drive, and set the project name to GXX_Lab0. Select the 'ARM Cortex-A9' processor architecture, and click 'Next'.

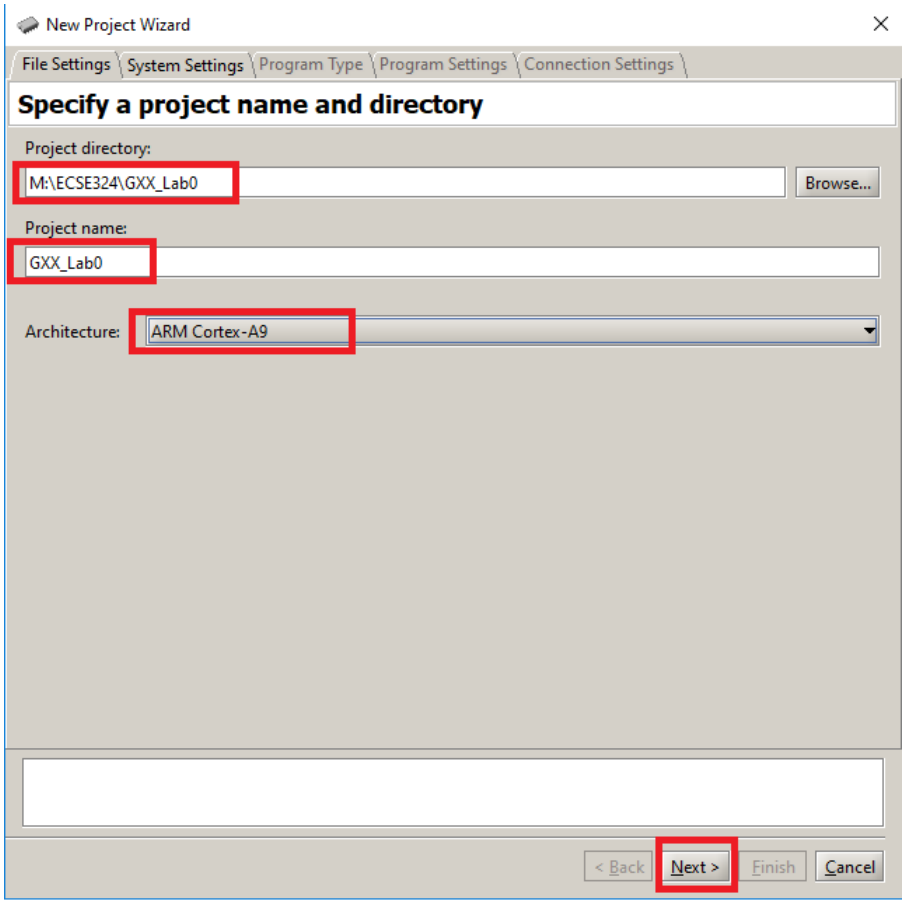

Figure 13: Running the demo program - Step 2

3. In the next window, under ‘Select a system’ select the ‘DE1-SoC Computer’ and click ‘Next’.

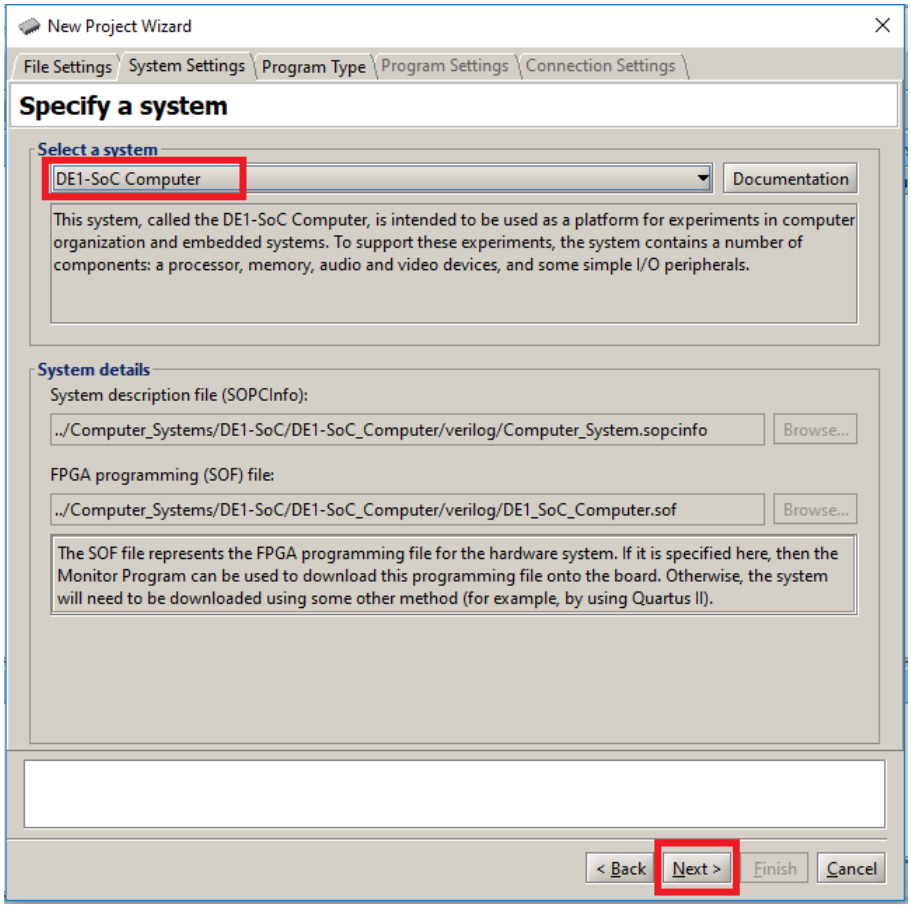

Figure 14: Running the demo program - Step 3

4. In the next window, under 'Program type' select the 'Assembly Program' and check the box marked 'Include a sample program with this project'. Then in the sample program menu select 'Getting Started' and click 'Next'.

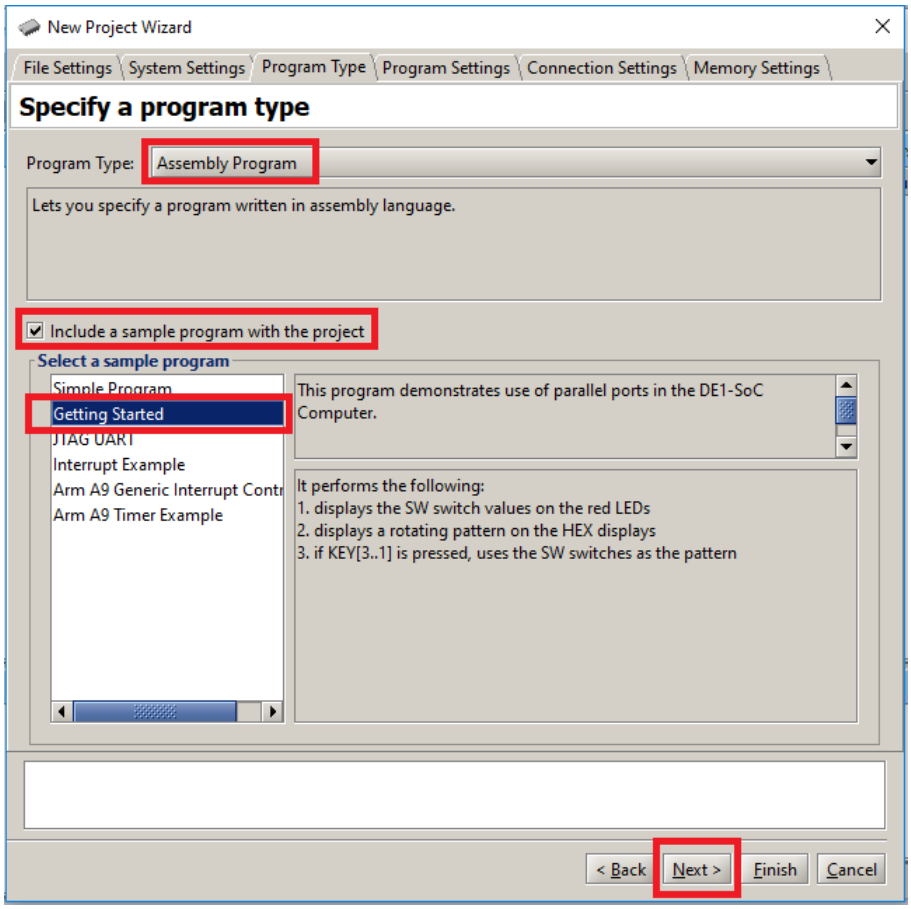

Figure 15: Running the demo program - Step 4

5. In the next window ‘Specify program details’, simply click ‘Next’.

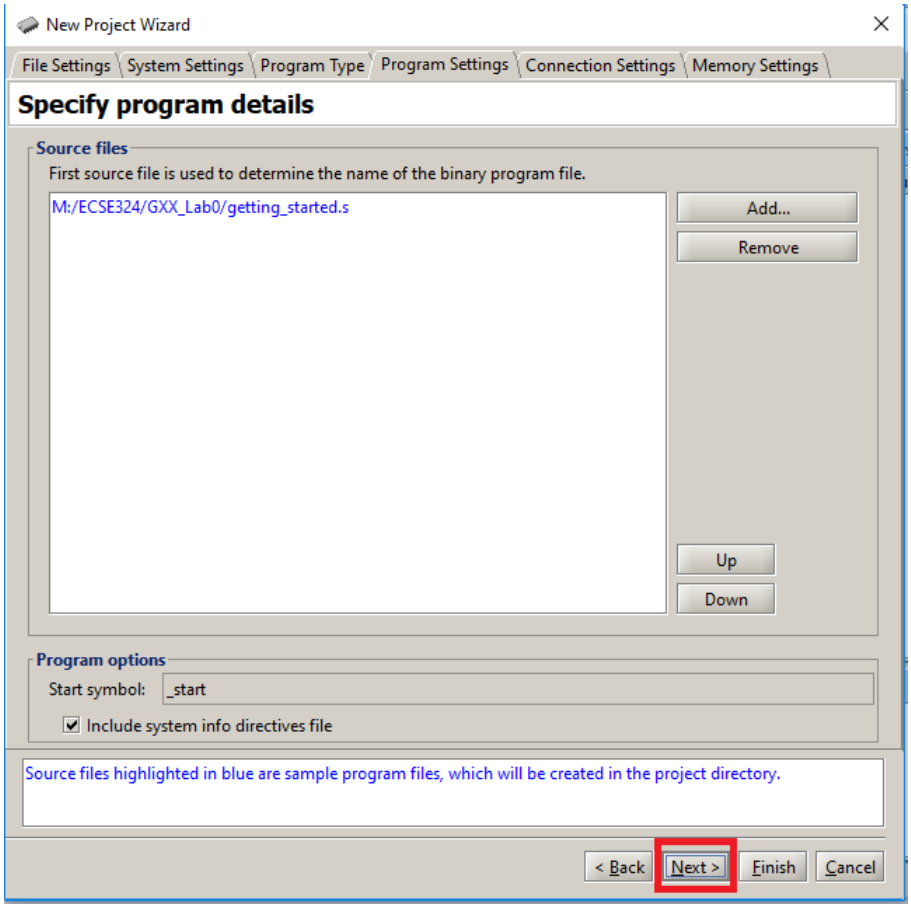

Figure 16: Running the demo program - Step 5

6. In the next window ‘Specify system parameters’, ensure that the board is detected in the ‘Host connection’ box, and click ‘Next’. Note that the board has to be plugged in via USB and powered on to be detected.

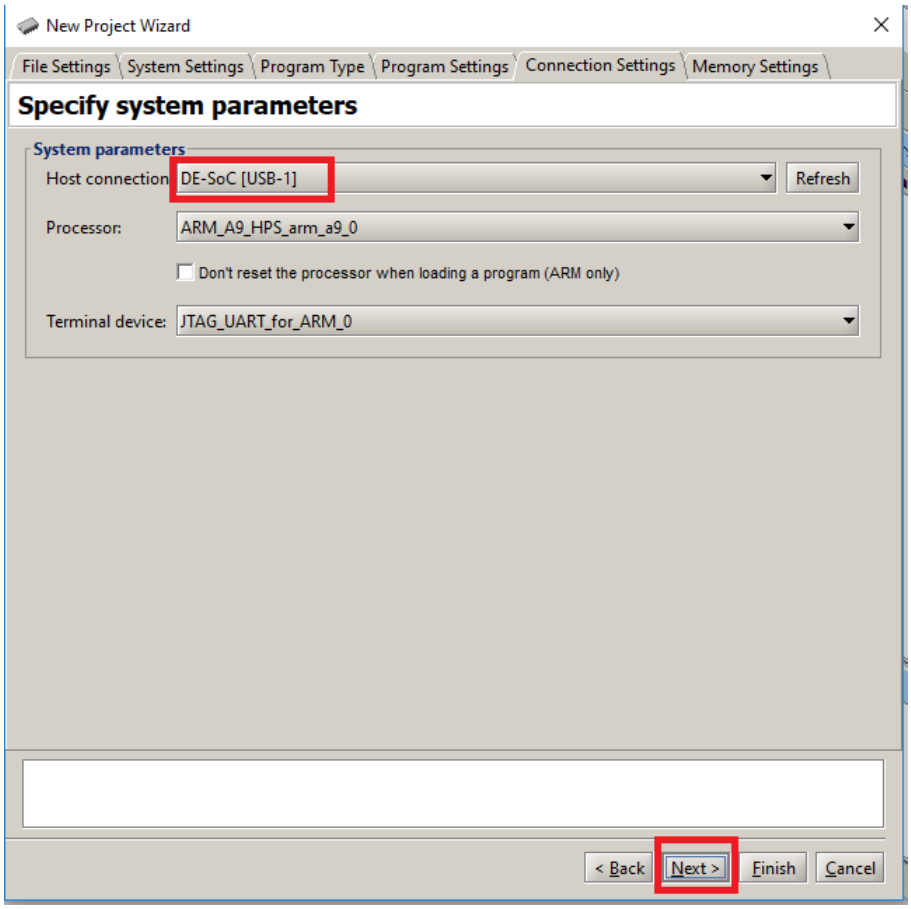

Figure 17: Running the demo program - Step 6

7. In the next window ‘Specify program memory settings’, simply click ‘Finish’.

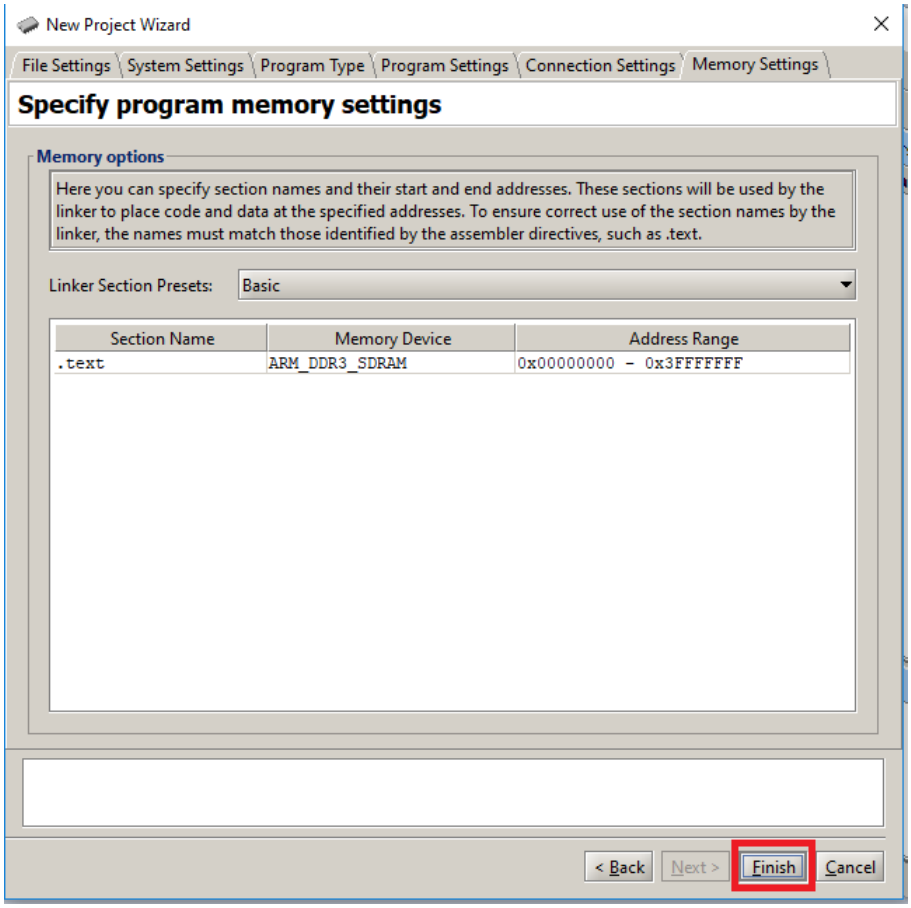

Figure 18: Running the demo program - Step 7

8. A dialogue box should now pop up, asking whether you would like to download the system onto the board. Since we have already flashed the SOF file onto the board, this is not necessary, so click 'No'.

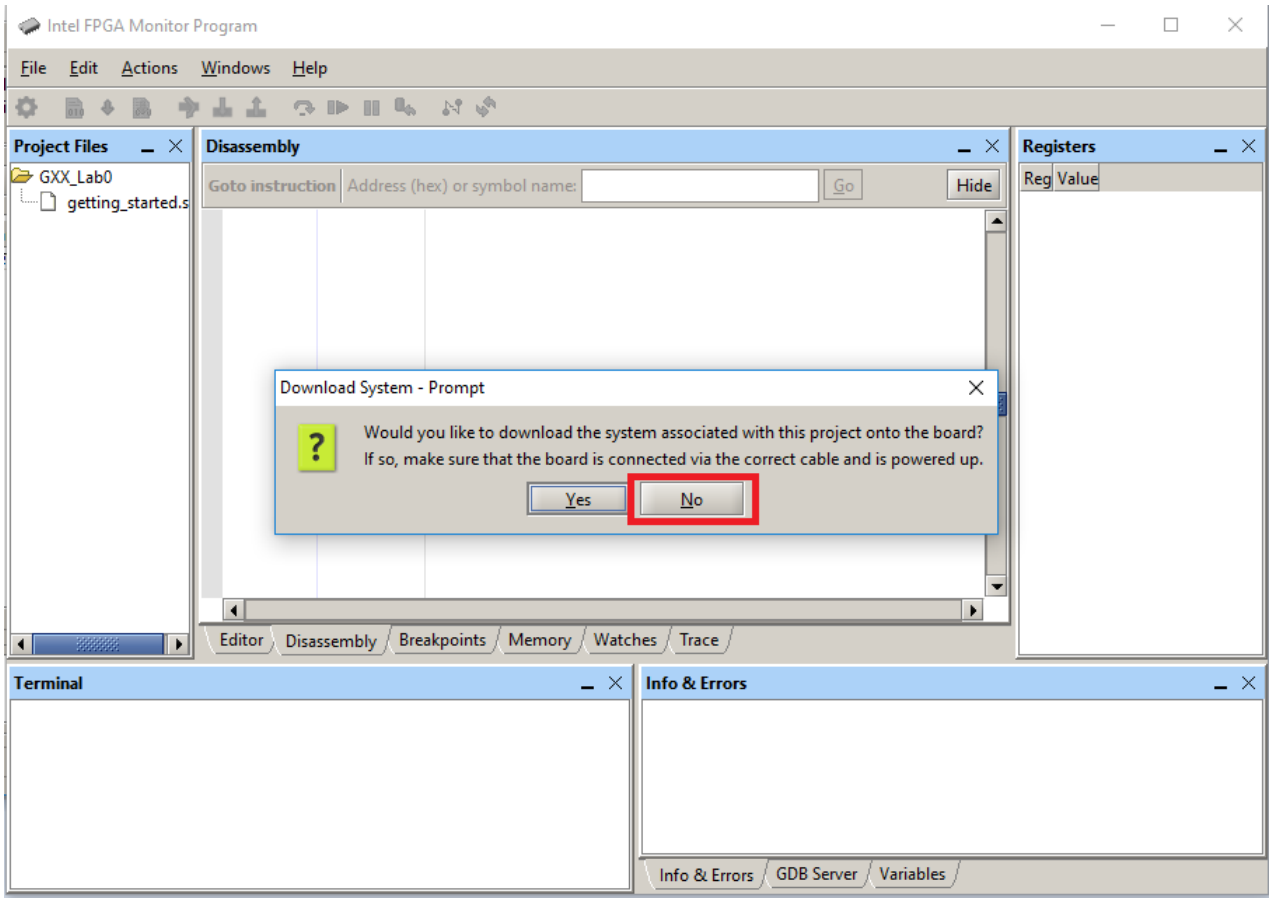

Figure 19: Running the demo program - Step 8

9. Finally, select Actions->Compile & Load, and when the project has finished loading select Actions->Continue. The demo project should now be running successfully on the board.

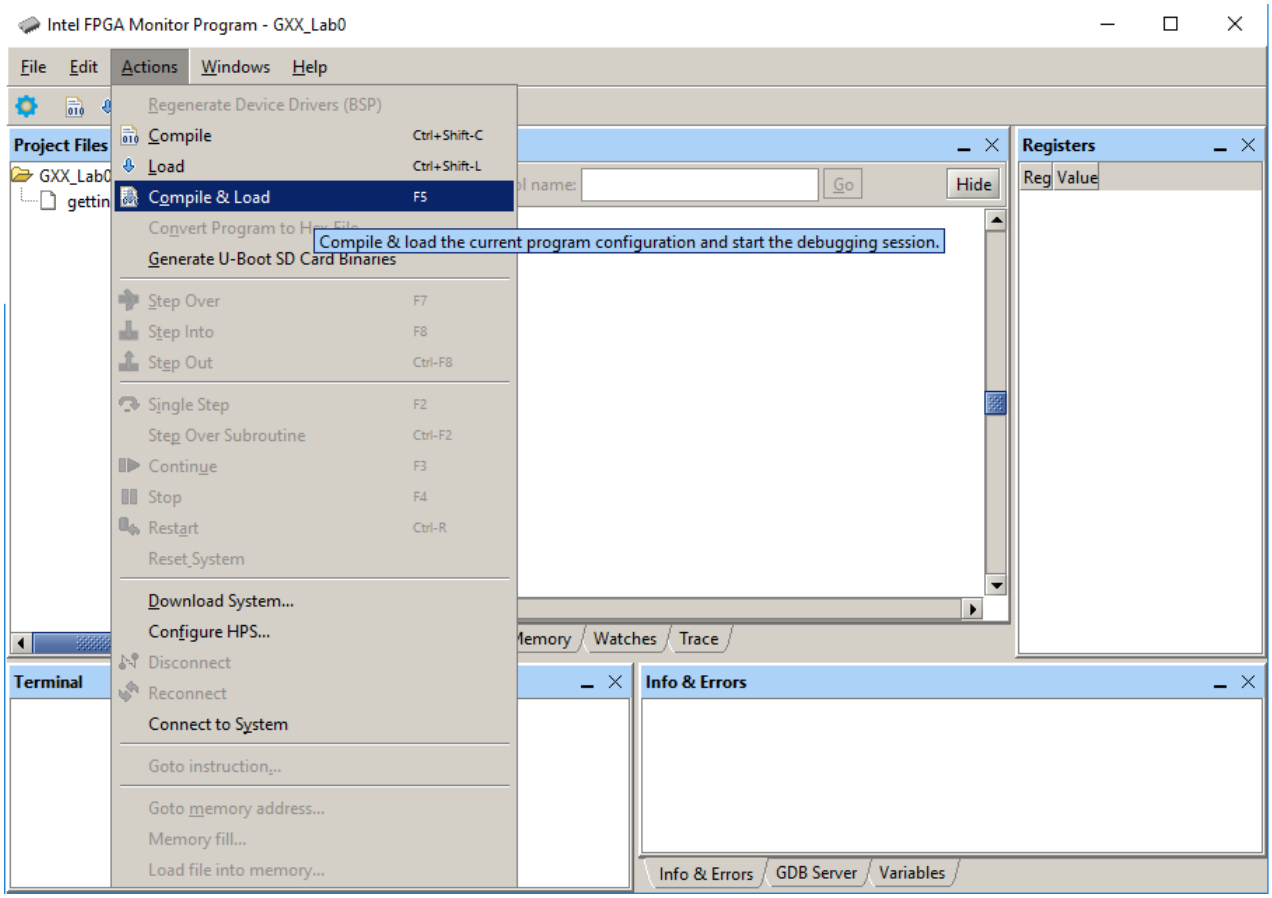

Figure 20: Running the demo program - Step 9

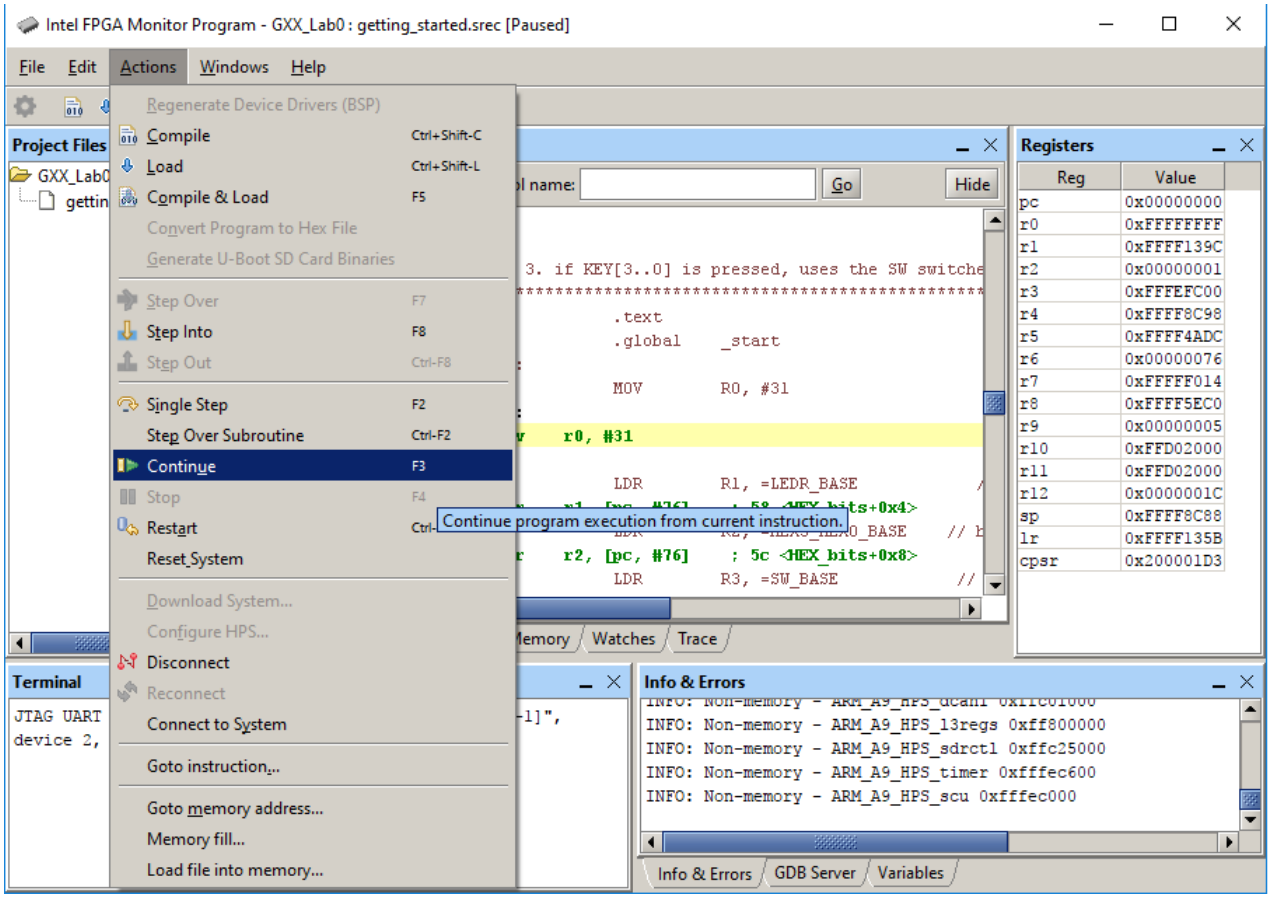

Figure 21: Running the demo program - Step 9