Extra Credit: Flashing the Spartan 7

Summary

Normally, and thus far in the semester, we have stored the Spartan 7 programs in volatile memory. This memory allows the board to run the uploaded program for only as long as the board has power. Sometimes it may be necessary to have your FPGA boot up and begin running the desired program without a PC. In order to do this we can store the program in non-volatile memory called *flash memory*. Once a program is stored in flash memory when the FPGA board receives power, it will automatically transfer the stored program into it's volatile memory and begin running the code.

1 Code Limitations

Pretty much any verilog code programmed in Vivado can be uploaded to the Spartan 7 flash memory. The only limitation is if your project/code is too large to fit onto the boards flash memory chip. To ensure that your code will be able to fit on the flash memory chip it is recommended that you upload and test your code normally. Running the code from the board's volatile memory, ensuring its full functionality first, you may then flash the system with your code.

2 Flashing the Spartan 7 Board

2.1. First, as described above, ensure that your code will run as intended on the Spartan 7 board in volatile memory. Once the code has been tested, use the following steps to store your program in flash memory:

2.2. As hinted at above, make sure that you have an up-to-date bitsteam file for use to create the flash image.

2.3. To use the flash chip on the CMOD Spartan 7, we must tell Vivado which chip the board is using. Start by connecting to your Spartan 7 board in Vivado, and opening the **"Hardware Manager"**.

2.4. Under the **"Hardware Manager"** select **"add Configuration Memory Device"**, then **"xc7s25_0"**. When prompted to select a memory device, search **"3233"**, and select the **"mx25l3233f"** chip.

2.5. Once the chip has been selected, Vivado will prompt you to select the `.mcs` file to upload.

2.6. The first thing that we want to do is to convert the current bitstream file into a file readable by the flash chip. To do this we need to create a `.mcs` file. Once in the **"Hardware Manager"**, select **"Tools"** from the top toolbar, and then **"Generate Memory Configuration File..."** This will open a prompt to create the `.mcs` file.

2.7. From the top, select the following options:

- Format: **MCS**
- Memory Part: **mx25l3233f-spi-x1_x2_x4**
- Filename: **"<Insert File Name Here>"**
- Interface: **SPIx1**
- Select **"Load bitstream files"**, with the start address of **000000000** and direction **UP**.
- After bitfile, click on the three small dots, and select the bitstream file that you want the board to run. (the bitstream file should be under something like **Project.runs/impl_1/Project.bit**)
- Lastly, check **Overwrite**

2.8. Now, to upload the `.mcs` file to the Spartan 7 board, under the **Hardware** window, right click the **mx25l3233f-spi-x1_x2_x4** (Under localhost, xilinx_tcf/Digilent..., xc7s25_0) and select **"Program Memory Configuration Device"** This should open a prompt to flash the board.

2.9. From the top, select the following options:

- Select the `.mcs` file to upload
- Address Range: **Entire Configuration Memory Device**
- Check **Erase** , **Program** , and **Verify**

After a short wait, the flash memory should be programmed. Although the board is now flashed, the board will not be running the flashed code because Vivado is still controlling the Spartan 7. To force the Spartan 7 to boot from the Flash memory you have two options:

2.10. 1) Unplug the board, close Vivado, and re-plug the board into the computer or another USB power source.

2.11. 2) Under the **Hardware** window, right click the **xc7s25_0** (Under localhost, xilinx_tcf/Digilent...) and select **"Boot from Program Memory Configuration Device"** This should have the Spartan 7 place the flash stored bitstream into volatile running memory, programming the FPGA.

Congratulations! If successful, your code should be running on the FPGA! The uploaded program will now be persistent if the FPGA has a power loss, and should start running shortly after boot-up.