RISC-V Configurations

There are six main processor designs you should worry about. There is a .zip folder in the Final USB Folder on OneDrive, but these are also still in the Public folder of computer one in the Senior Project Lab. Opening Libero will show all these projects to be opened.

"Solution" is the example processor provided by Microsemi, accompanying the PDF about building a RISC-V processor using the included IP in Libero.

"Solution - Copy" is a copy of the "Solution" processor. We resaved the original solution architecture into a new project and made all the necessary modifications to add I2C, SPI, UART, extra GPIO pins, and the rest necessary to interface with everything on the PCB. Save this just as a point of reference. Don't change anything, as this will make a good reset point if things go bad in the other 4 projects and you need a place from which to restore basic functionality. All the current SoftConsole code is included with this project. It runs as-is and will also serve as a point of reference.

"Solution - MIV_AXI" is a copy of the finished "Solution - Copy" design. This is where you should continue modifying and changing the design to ensure everything is working. Run your benchmark code on this processor and compare to the other three.

"Solution - CoreRISCV" is very similar to "Solution - MIV_AXI" because they both use the AXI memory controller standards. We unplugged the MIV processor and plugged in a CoreRISCV processor instead. This processor is based on SiFive's E31 RISC-V design, unlike the MIV designs that were built by Microsemi. This is the most lightweight of all the processor designs. No SoftConsole code is currently included. Make sure that everything works alright, that the design compiles correctly, and then copy the code over from the "Solution - MIV_AXI" project. Make sure to put this design into TMR. Benchmark this design and compare it to the other three.

"Solution - MIV_AHB" is also similar to the "Solution - MIV_AXI" processor. This one uses the AHB memory controller standard instead of AXI. We ran out of time and were unable to finish the conversion between AHB to AXI for the memory controller. The IO controller is correctly connected. Finish converting the memory controller and make sure the rest of the design is working properly, then compile and run. Port over the SoftConsole code. Make sure to put this design into TMR. Benchmark this design and compare it to the other three.

"Solution - MIV_FP" also uses AHB. We took the "Solution - MIV_AHB" design, unplugged the processor, and plugged in this one with a Floating-Point unit. This is specially designed to deal with doubles as well as integers. With the AHB controller, figure out the conversion to AXI for the memory controller, then verify the rest of the design. Port over the SoftConsole code. Make sure to put this design into TMR. Benchmark this design and compare it to the other three.