NASA
TMR RISC-V MCU for CWS

ECEN 499

Chris Porter, Kyle Toliver, Spencer Cheney, Jonah Boe

Overview:

- **Purpose**
- **Goals**
- **Hardware & Software**
- **Libero**
- **SoftConsole**
- **Future Areas of Improvement**

Purpose:

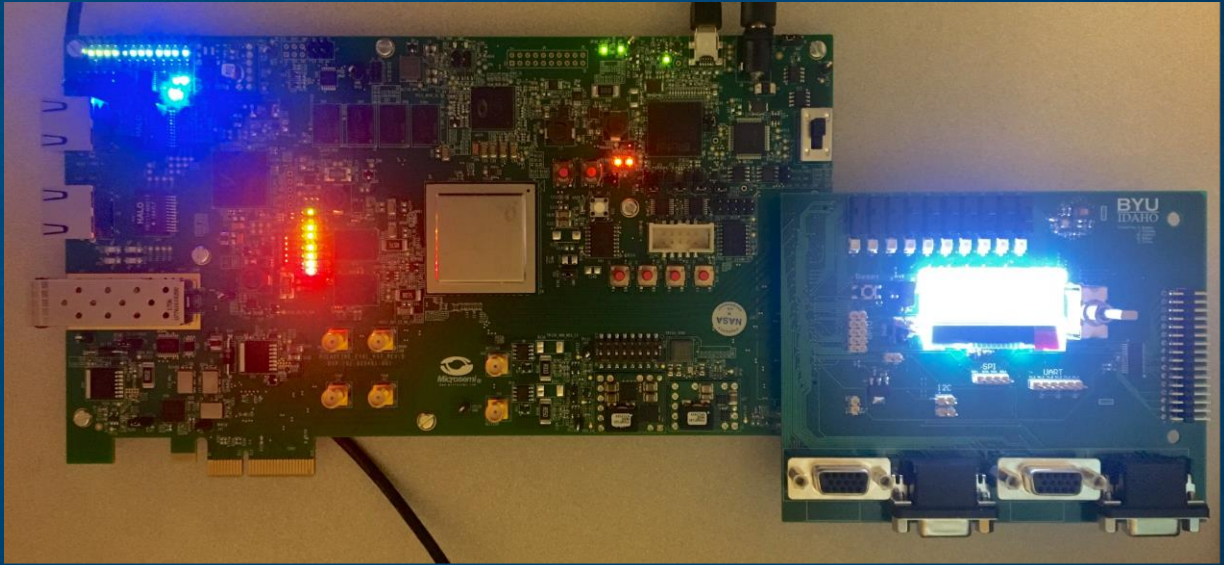
Using the PolarFire FPGA Eval kit and a custom PCB, benchmark various RISC-V core configurations in Triple Modular Redundancy (TMR) for use in NASA's Caution and Warning System (CWS) in the Portable Life Support System (PLSS) of the newest space suit (xEMU).

Goals:

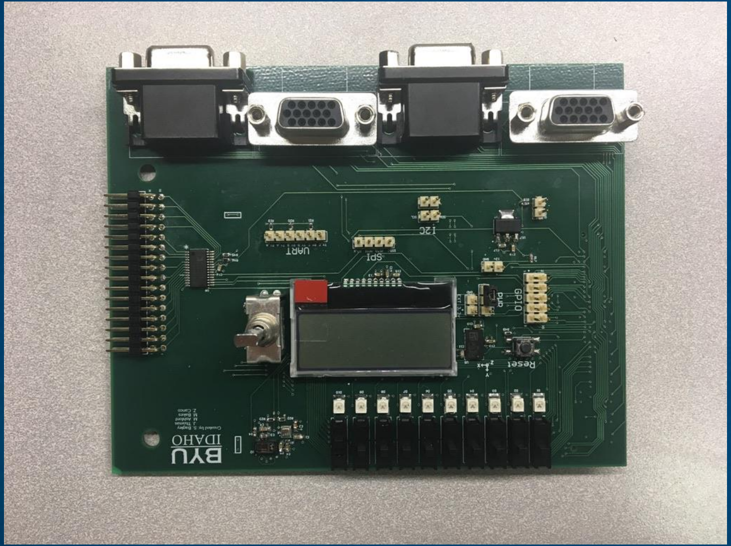
- **PCB**
  - **Assemble Custom PCB**
  - **Test for and Resolve Conflicts**
- **Communication**
  - **Full-Duplex UART**
  - **SPI**
  - **I2C**
- **Create multiple configurations of RISC-V cores in TMR**
- **Benchmark configurations and create a detailed report of the outcome**

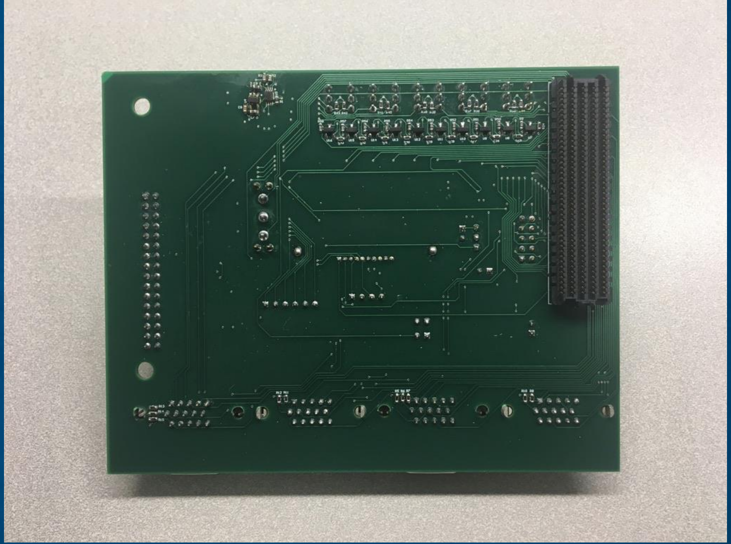
Hardware & Software:

- PolarFire FPGA
- Libero
- SoftConsole
- Custom PCB



PCB Assembly:





PCB Assembly:

Problems Overcome:

- **Got all components connected and tested successfully**
- **Due to world health condition parts were delayed, but made it in time to accomplish some tasks**

Problems Not Overcome:

- **All known issues have been resolved**

Libero:

integrates industry standard Synopsys Synplify Pro® synthesis and Mentor Graphics ModelSim® simulation with best-in-class constraints management, Programming & Debug Tools capabilities, and secure production programming support.

Accomplished:

- **Debugged several module and set them up to work correctly**
- **Configured UART module to be set up for Full-Duplex**
- **Fixed the LVDS UART Module so that it can transmit a differential signal**

Yet to be done:

- **Set up the other three processor design to work with various other memory configurations**
- **Turn LVDS UARTs differential signal into a LVDS signal**
- **Add a LVDS module for receiving data**
- **Fully test SPI and I2C**
- **Fix GPIO I/O configuration**

Libero:

integrates industry standard Synopsys Synplify Pro® synthesis and Mentor Graphics ModelSim® simulation with best-in-class constraints management, Programming & Debug Tools capabilities, and secure production programming support.

Problems Overcome:

- In order to work on Libero remotely we had to set up a zoom meeting share the Libero screen and request remote access from our computers at home
- Change UART configuration to work with a baud rate that is usable with LVDS

Problems Not Overcome:

- LVDS UART baud rate/timing isn’t accurate

SoftConsole: Free software development environment facilitating the rapid development of bare-metal and RTOS based C/C++ software for Microsemi CPU and SoC based FPGAs.

Accomplished:

- **Further developed a SoftConsole project that can run on the first processor design**
- **Mostly finished driver code for the LCD screen that can possibly run on other core configurations (depending on if the APIs change between processors)**
- **Further developed code for GPIO. Can possibly run on other core configurations (depending on if the APIs change between processors)**

Yet to be done:

- **Write driver code for remaining sensors**
- **Debug test program for LCD screen**

SoftConsole: **Free software development environment facilitating the rapid development of bare-metal and RTOS based C/C++ software for Microsemi CPU and SoC based FPGAs.**

Problems Overcome:

- **Properly transmitting data over SPI to communicate with the LCD**
- Correctly addressing each of the GPIO inputs in order to display current states of peripherals

Problems Not Overcome:

- **LCD Does not display anything when communicated with. Using an oscilloscope we can confirm that the messages are being sent, and the chip select is being triggered**

Future Areas of Improvement:

Libero

- **Fix issues with full duplex LVDS UART**
- **More architectures**
- **Benchmark CPUs**

SoftConsole

- **Write driver code for remaining sensors**
- **Debug issues with LCD test**

Questions?