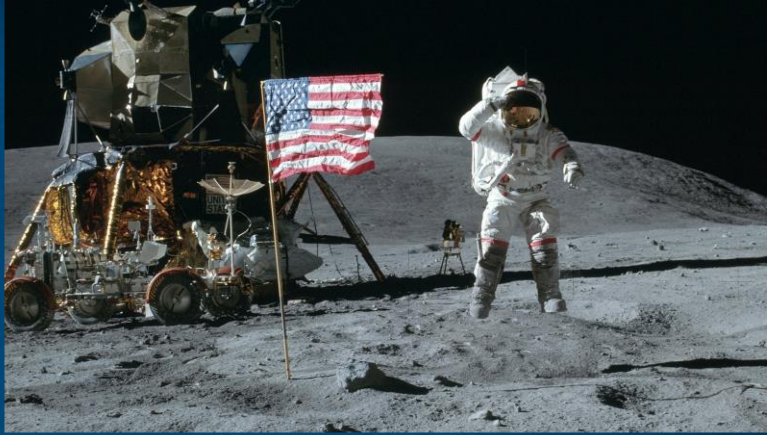
NASA
TMR RISC-V MCU for CWS

ECEN 499

Chris Porter, Kyle Tolliver, Spencer Cheney, Jonah Boe

Overview:



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

https://ecen499-nasa.github.io/

Purpose:

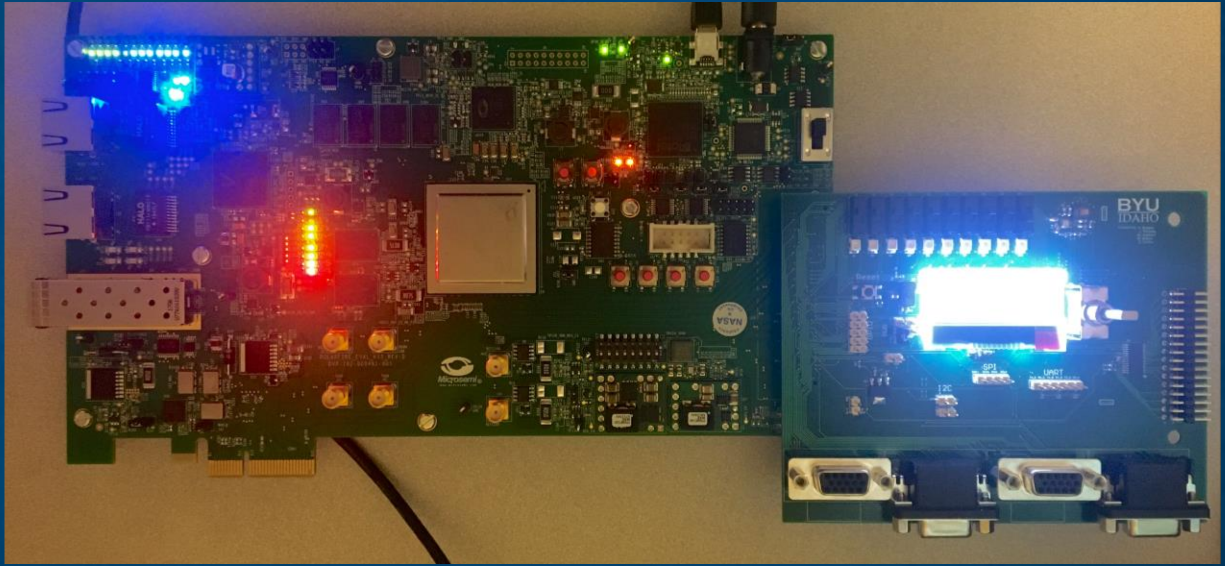
Using the PolarFire FPGA Eval kit and a custom PCB, benchmark various RISC-V core configurations in Triple Modular Redundancy (TMR) for use in NASA's Caution and Warning System (CWS) in the Portable Life Support System (PLSS) of the newest space suit (xEMU - Exploratory Extravehicular Mobility Unit).

Goals:

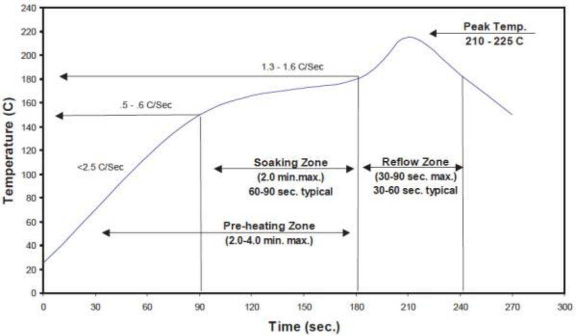
- PCB
  - Assemble Custom PCB
  - Test for and Resolve Conflicts
- Communication
  - Full-Duplex UART
  - SPI
  - I2C
- Create multiple configurations of RISC-V cores in TMR
- Benchmark configurations and create a detailed report of the outcome

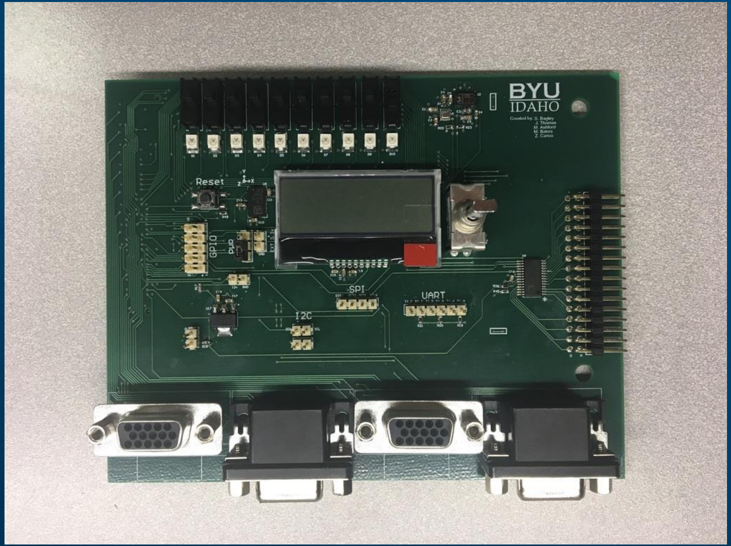
Hardware & Software:

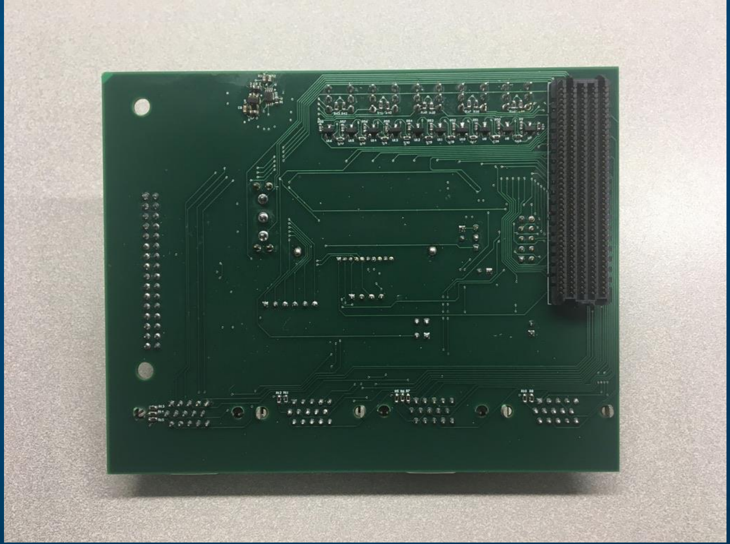
- PolarFire FPGA
- Libero
- SoftConsole
- Custom PCB



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:

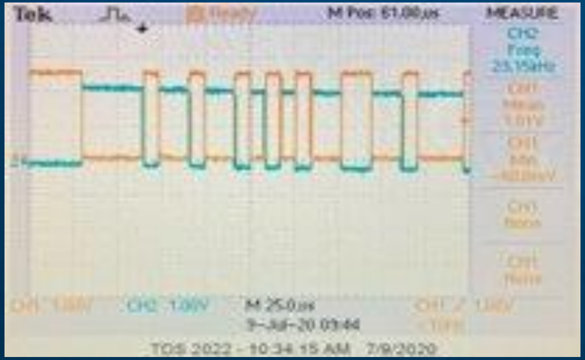
- **All known issues have been resolved**

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:

- Tested and debugged modules to ensure they're 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:

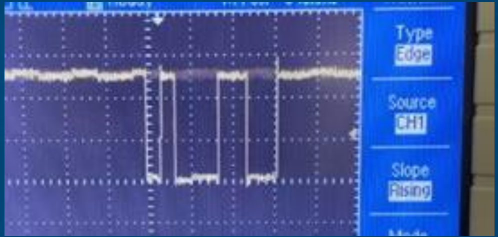
integrates industry standard Synopsys Synplify Pro® synthesis and Mentor Graphics ModelSim® simulation with best-in-class constraints management, Programming & Debug Tools capabilities, and secure production programming support.

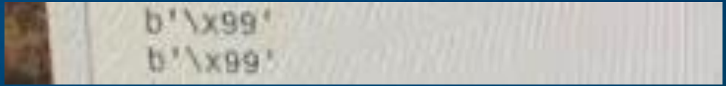
Problems Overcome:

- 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





```
b'\x99'
b'\x99'
```

- We are transmitting 00110001 and receiving 10011001
- The second or third bit is getting skipped and the parity bit is getting read in as data instead of being use to check signal accuracy

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)
- Created functions for LVDS and Full Duplex UART

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
- Setting GPIO outputs high or low

Problems Not Overcome:

- LCD Does not display anything when communicating with it. Using an oscilloscope we can confirm that the messages are being sent, and the chip select is being triggered
- Controlling both GPIO Inputs and Outputs using the same GPIO Core.

Future Areas of Improvement:

Libero

- More architectures
- Benchmark CPUs

SoftConsole

- Write driver code for remaining sensors
- Write application for displaying information on the LCD Screen.

Set up devices to communicate with FPGA (Arduino, Raspberry pi, etc).

Questions?