

E155 Final Project Status Report: μ Mudd Mark V Debugging and Lab 6 Revision

Christopher Ferrarin and Kaveh Pezeshki
31 November 2018

1 Completed Deliverables Status

To summarize the status of our final project, below is a summary of project deliverables and deliverable status.

Deliverable Category	Deliverable Name	Deliverable Status
Identifying blocking μ Mudd Bugs	Identifying MCU program- ming failure	Complete
Revising μ Mudd to allow MCU functionality	Hardware modification of pre-existing PCBs	Complete
	New JTAG cable	Complete
	Modified schematic and lay- out	In progress
	Completed and assembled μ Mudd respin	Not started
Reworking Lab 6	Rewrite EasyPIO.h with- SAM4S support	In progress
	Integrate MCP3002, photo- diode, and BlueSMiRF	In progress
Testing other labs (pending further discussion)	Lab 4	Not started
	Lab 5	Not started
	Lab 7	Not started

2 Deliverable Status: Revised μ Mudd

A major component of this final project is identifying errors in the PCB design that lead to a non-programmable MCU. We have identified two errors which when solved allowed MCU programming

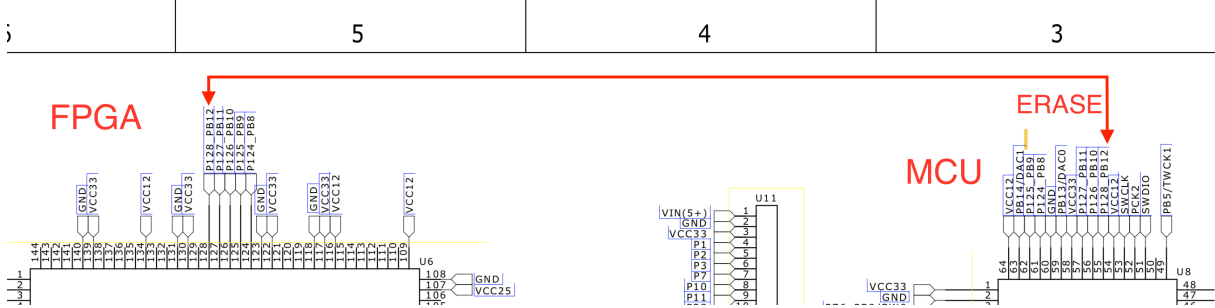
2.1 Schematic Errors

2.1.1 MCU ERASE Pin

The largest problem with the current μ Mudd design lies in the MCU ERASE pin, which reinitializes the onboard flash as well as resetting the processor. The ERASE pin can also serve as general-purpose I/O after configuration.¹

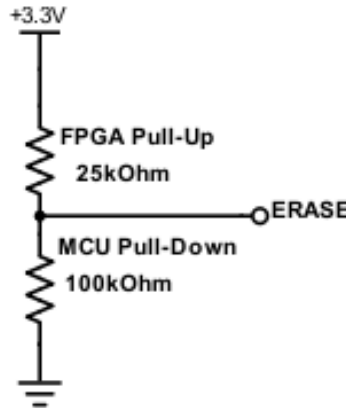
¹SAM4S Series Datasheet p37

On boot, ERASE must be held low to prevent flash erase and reinitialization of the processor. On the current μ Mudd, ERASE was tied to a general I/O pin on the Cyclone IV FPGA. The connection can be seen in the following schematic:



The marked connection ties ERASE on the MCU to pin 128 on the FPGA

The ERASE pin contains a 100k Ω pull-down resistor². An unconfigured Cyclone IV I/O pin contains a 25k Ω pull-up resistor³. This creates a resistor divider circuit as shown below:



This provides a predicted voltage of 2.64V on the MCU ERASE pin, close to the 2.86V we observed. This is a high logic level which prevented FPGA programming.

2.1.2 MCU Power Supply

The MCU requires a 3.3V and 1.2V power supply. It can be powered via one 3.3V supply, and use an internal regulator to generate 1.2V, or it can be powered with an external 3.3V and a 1.2V supply. The dual-regulator design of the current board can introduce startup issues if timing is not correct.

We believe that these potential timing errors can cause system instability, as we observed an unresponsive MCU after startup that could only be solved with a full erase and reset.

²SAM4S Series Datasheet p37

³Cyclone IV Device Handbook p6-3

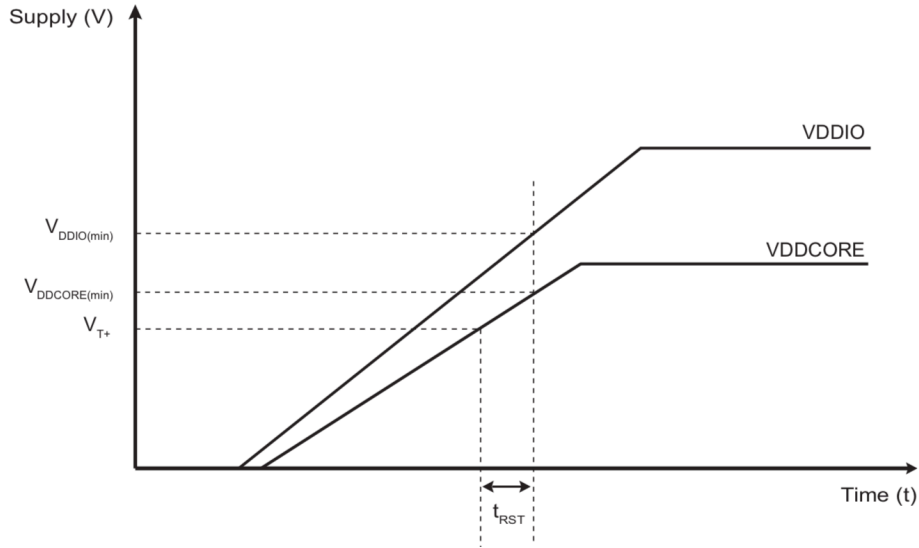


Figure 1: Timing requirements for the 1.2V (VDDCORE) and 3.3V (VDDIO) supplies, taken from the SAM4S Series Datasheet p27

2.1.3 JTAG connector pinout

The MCU JTAG connector was incorrectly wired on the current μ Mudd.

2.2 Schematic and Layout Changes

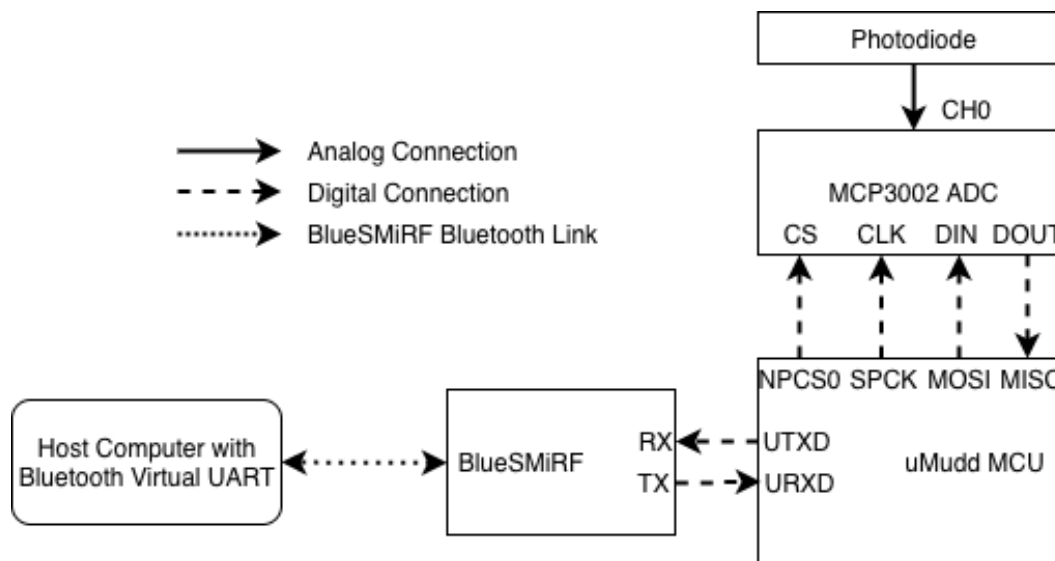
We are currently implementing a set of changes to solve the problems noted above and to improve the PCB. These include:

1. Moving ERASE control to the MCU RESET pushbutton. RESET will be accessible through JTAG
2. Powering the 1.2V MCU VDDCORE with the onboard regulator
3. Correcting JTAG wiring erros
4. Replacing 0.1" pitch JTAG connectors with 0.05" pitch SWD connectors. This adds compatibility with J-Link EDU Mini programmers
5. Adding a separate 40MHz clock to the FPGA. The clock is currently supplied by a MCU I/O pin

3 Deliverable Status: Reworking Lab 6 and EasyPIO.h

3.1 Reworking Lab 6

The proposed Lab 6 architecture is shown in detail below:



To maintain a low lab cost we exchanged one BlueSMiRF and the serial display for a computer with an integrated bluetooth module. We added a MCP3002 ADC to retain the datasheet interpretation component of the lab.

We have successfully demonstrated bluetooth communication between two computers, using a Bus Pirate as a USB to UART converter. We have also successfully demonstrated voltage measurement with the MCP3002 through a SAM3S/SAM4S SPI peripheral.

The remaining tasks in the rework of Lab 6 are testing our UART peripheral implementation on the SAM3S/SAM4S, and writing an updated version of the lab manual.

3.2 Targeting EasyPIO.h for the SAM4S

We aim to create an I/O header file, `EasySamIO.h`, which provides Arduino-style access to the GPIO, timer, UART, and SPI peripherals on the SAM4S. In the style of `EasyPIO.h`, we provide only the configuration and functionality necessary to complete labs. We aim to provide adequate inline documentation for students to add functionality as necessary. This documentation includes functional descriptions of memory access, references to the lab manual, and brief descriptions of other peripheral features.

We have completed implementation of the GPIO, timer, and SPI peripherals, and are currently testing our UART implementation.

We believe a thorough and well-documented `EasySamIO.h` will be more valuable than tests of pre-existing labs as discussed in our project proposal.

4 Appendix 1: Schematics and Layout

5 Appendix 2: C Code

5.1 EasySamIO.h

5.2 MCP3002 Voltage Measurement

5.3 BlueSMiRF UART Communication