A LOW POWER FPGA COMPUTE ARRAY FOR MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE APPLICATIONS

by

Todd Wilson

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Electrical Engineering

Approved:

Charles Swenson, Ph.D. Major Professor	Jonathan Phillips, Ph.D. Committee Member
Donald Cripps, Ph.D. Committee Member	David F. Feldon, Ph.D. Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2026

ABSTRACT

A Low Power FPGA Compute Array for Machine Learning and Artificial Intelligence Applications

by

Todd Wilson, Master of Science

Utah State University, 2026

Major Professor: Charles Swenson, Ph.D.
Department: Electrical and Computer Engineering

Edge computing is transforming the capabilities of satellite-based data systems by enabling real-time, in situ analysis of scientific and environmental data. This thesis proposes the design and implementation of a Low-power Array for Cubesat Edge Computing Architecture, Algorithms, and Applications (LACE-C3A) targeted for deployment in space-constrained environments such as small satellites. Unlike single-FPGA systems limited by computational and power constraints, LACE employs a modular cluster of FPGAs where a central controller routes data and manages workloads across multiple peripheral compute units. These compute units can be reconfigured in orbit to support a variety of AI and ML models. The system uses Microchip PolarFire™ FPGAs to ensure radiation resilience and integrates a robust interface architecture to support high-speed sensor data acquisition and real-time model execution. This thesis focuses on the hardware design, including schematic capture and simulation, to validate the feasibility of fabricating and operating such a system while in-situ. The LACE-C3A platform lays the groundwork for low-power, flexible, and scalable edge computing in aerospace and remote terrestrial applications.

(69 pages)

PUBLIC ABSTRACT

A Low Power FPGA Compute Array for Machine Learning and Artificial Intelligence Applications

Todd Wilson

Modern small satellites often lack the ability to analyze their data in real time, relying instead on downlinking raw measurements to Earth before any scientific interpretation can occur. This delay restricts missions that require rapid awareness of environmental or space-weather events. This thesis presents the Low-power Array for Cubesat Edge Computing Architecture, Algorithms, and Applications (LACE-C3A), a hardware platform designed to enable in-orbit data processing within the power and volume limits of CubeSat-class spacecraft.

LACE-C3A uses a modular cluster of flash-based FPGAs, which offer low power consumption, radiation resilience, and in-flight reconfigurability. The system consists of a Controller board responsible for ingesting and routing data from diverse sensors, and Peripheral compute boards that execute machine-learning algorithms using dedicated memory and multi-gigabit communication links. This work focuses on the hardware design of these components, including power regulation, high-speed serial interfaces, memory subsystems, and board-to-board interconnects.

The results demonstrate that a scalable, FPGA-based compute array can support real-time edge computing on small satellites, providing a viable path toward more autonomous and responsive spacecraft.

To my wife and children

ACKNOWLEDGMENTS

This thesis would not have been finished without the many ideas and contributions from my advisor, Dr. Charles Swenson, and my fellow students working on this project, Kade Howes and Shawn Jones. I thank them for their many hours of discussion and advice.

Todd Wilson

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

ACRONYMS

AI	Artificial Intelligence
COTS	Commercial Off-The-Shelf
CSE	Utah State University Center for Space Engineering
EPS	Electrical Power System
FFC	Flat-Flex Cable
FPGA	Field-Programmable Gate Array
LDO	Linear Drop-Out
LEO	Low-Earth Orbit
MT/s	Mega-Transfers per Second
ML	Machine Learning
NDA	Non-Disclosure Agreement
PCB	Printed Circuit Board
SerDes	Serializer/Deserializer
SEU	Single Event Upset
SOC	System On Chip
SOM	System On Module
STMD	Space Technology Mission Directorate
XGBoost	Extreme Gradient Boosting
YOLO	You Only Look Once

CHAPTER 1

Introduction

Downlinking and processing data from satellite observatories is a time-consuming and resource-intensive process. The increasing computational effort required to analyze scientific data in situ is at odds with the growing demand for real-time notification of environmental or space-weather events. In many missions, researchers want immediate awareness when conditions change — for example, when structures in the Van Allen radiation belts shift or when atmospheric conditions deviate from predicted models. Traditional data-collection workflows require acquiring raw data, buffering it, downlinking it over limited bandwidth, processing it on the ground, and then performing scientific analysis. Even under ideal conditions, this cycle takes hours. For many sensing applications, that latency is acceptable. For others, particularly those requiring event detection or rapid response, it is not.

Artificial Intelligence (AI) and Machine Learning (ML) models offer clear benefits for these scenarios. Lightweight models such as Extreme Gradient Boosting (XGBoost) can operate on low-power embedded hardware, while more advanced models, such as the You Only Look Once (YOLO) family used for real-time image detection, require substantially more computational capacity. These larger models are demanding even for ground-based hardware, making them difficult to deploy directly on resource-limited spacecraft platforms.

Small satellites exacerbate this challenge. Although they are cost-effective to build and launch, they offer limited power generation capability and far less payload volume compared to traditional satellites. Many missions rely on Field-Programmable Gate Arrays (FPGAs) because of their reconfigurability and proven reliability in orbit. FPGAs are capable of running ML models, but in practice, a single device typically supports only one significant model at a time due to logic, memory, and timing constraints. Missions requiring multiple simultaneously active algorithms often stack multiple FPGA or CPU/GPU based compute devices and associated electronics, driving up power consumption and system complexity.

A cluster-based approach offers a more scalable solution. In this architecture, a central Controller FPGA handles data ingestion and routing, while Peripheral FPGAs execute individual algorithms or AI/ML model pipelines. This structure reduces overall system power for a given workload and enables in-orbit reprogramming of compute units as mission priorities evolve.

This thesis investigates the feasibility of implementing such a system: the Low-power Array for Cubesat Edge Computing Architecture, Algorithms, and Applications (LACE-C3A). The work focuses exclusively on hardware design and fabrication. The central objective is to determine whether a modular, reconfigurable FPGA compute array can be realized within the power, size, and environmental constraints of small-satellite missions.

The hardware platform developed in this research includes a Controller board capable of powering, communicating with, and reprogramming multiple Peripheral FPGAs, along with several data-ingest interfaces to support sensors such as software-defined radios, cameras, and external spacecraft systems. Peripheral boards were designed to host compute-oriented FPGAs and their associated memory subsystems. Supporting test hardware, including a SerDes breakout board, was created to validate high-speed interconnect design decisions.

This work was performed in collaboration with the Utah State University Center for Space Engineering (CSE), which provided development tools, fabrication support, and laboratory test equipment. The scope of this thesis is limited to the hardware aspects of LACE-C3A; firmware, software, and model-execution pipelines are topics of ongoing research by other members of the project team.

The remainder of this thesis is organized as follows. Chapter 2 reviews current edge-computing solutions and identifies limitations in existing architectures. Chapter 3 details the methodology used to design the system, including requirements, design flow, simulation strategy, and schematic organization. Chapter 4 presents the hardware implementation of the Controller, Peripheral, and breakout boards. Chapter 5 discusses design outcomes, limitations, and areas for future improvement. Chapter 6 concludes with the findings of this work and the demonstrated feasibility of a scalable, low-power FPGA compute array

for in-situ data processing.

CHAPTER 2

Literature Review

2.1 Edge Computing In Situ

Moving data processing and computation nearer to the data source has many benefits, especially in the context of small satellite activities. Edge computing moves the data processing operations to the data source, rather than sending raw data to be processed to the ground [1]. The traditional approach of gathering data and downlinking to the ground has the major drawback of not being scalable [1]. Many data gathering missions do not require huge bandwidth to send their collected data to the ground. But the missions that collect high volumes of data to be processed, such as ground imaging missions, are severely limited by data downlink speed bottlenecks. State of the art communication systems operate in the X-band. This bandwidth allows for "just above 500 Mb/s" data transfer speeds [2]. There are current and future plans to use the 26 GHz band for higher bandwidth availability, but this band is much more susceptible to atmospheric interference [2].

Edge computing/data processing while in situ also opens avenues into real-time data analysis and now-casting services that have previously been unavailable to science mission operators. Processing sensor data at the edge allows for on-orbit data preprocessing, real-time remote sensing image analysis, and disaster monitoring [3]. For these reasons, moving high volume data processing to the edge is the direction many system designers are choosing to go.

There are a few current efforts to bring edge computing hardware to on-orbit systems. These systems typically use commercial off-the-shelf (COTS) components in whole, or combine these COTS components with custom hardware to accomplish edge computing goals.

Examples of these systems include:

- Unibap iX10-100A [4]
- Unibap iX5-106 [4]
- TelePIX TetraPLEX OBP [4]

These systems all use COTS parts and some custom hardware to accomplish data-gathering and edge computing operations. The Unibap iX10 system uses an AMD Ryzen™ Accelerated Processing Unit (APU) and an Intel Myriad-X™ Vision Processing Unit (VPU) to analyze sensor data [4]. The Unibap iX5 system combines an AMD APU and Intel VPU with a Microchip SmartFusion2™ FPGA [4]. These systems are able to process significant amounts of data, but at a large power cost (25-40 W and 15-25 W respectively) [4]. These two Unibap systems are also reliant on the radiation tolerance of the COTS components to ensure the hardware will be resistant to the space environment.

The TelePIX TetraPLEX OBP is another system combining many COTS components into one edge computing enabled package. The TetraPLEX uses an FPGA/RISC-V System On Chip (SOC) in combination with two NVIDIA Jetson™ System On Modules (SOM) to enhance data processing capabilities [4]. This system also comes at the cost of high power requirements. The TetraPLEX is advertised with a max power draw of 36 W [4].

These systems come at the cost of high power requirements. But since they both use advanced chips with sub-ten nanometer photolithography processes, the SOCs and SOMs are also susceptible to single particle events while in orbit [5]. Single particle upsets will not damage hardware, but can cause logical errors in software [5]. Edge computing systems immune to hardware-damaging radiation and software-corrupting single particle upsets are essential to make data processing in situ a viable option for future space-based science missions.

2.2 FPGA Clusters In Situ

As mentioned before, SEUs are a danger to the accuracy and longevity of space-based data operations. SEUs can cause problems ranging from data damage in memory up to mission failure [6]. Using FPGAs, especially FPGAs with Flash-based logic elements, immunizes the target system from SEUs [7]. This dramatically reduces the cost and complexity for missions that operate in Low-Earth Orbit (LEO).

Another challenge for missions operating in the space environment is power requirements. As discussed above, traditional methods of data processing, specifically processing using AI/ML methods, have power requirements from fifteen to forty watts. FPGAs offer a lower-power alternative while having the capability to run multiple algorithms simultaneously.

FPGAs, and more specifically FPGA clusters, have been shown to outperform traditional CPU/GPU solutions in both computing power and power efficiency when given the same task [8]. Some examples include the ARUZ 2018 cluster. It was used to simulate the fluid mechanics of 1.5 million molecules. When the FPGA cluster was compared to a 6-core CPU performing the same task, it was 1.6 more times efficient while performing the task 1600 times faster.

Combining the SEU immunity and strong compute abilities of Flash-based FPGAs will give this computing cluster the edge over traditional in situ edge computing solutions. Table 2.1 compares the features and capabilities of the aforementioned COTS processors and the proposed LACE-C3A computing platform.

Device	**Power (W)**	**Compute Platform**	**Radiation Tolerance**	**Configurability**
Unibap iX10-100A	25-40	AMD Ryzen V1000 CPU/GPU, Intel Myriad-X VPU	SEE mitigation	Linux platform running AI/ML algorithms
Unibap iX5-106	15-25	AMD Steppe Eagle CPU/GPU, Intel Myriad-X VPU, Microchip SmartFusion2 FPGA	SEE Mitigation	Linux platform running AI/ML algorithms
Telepix TetraPLEX OBP	36	Microchip PolarFire SOC, NVIDIA Jetson ORIN NX SOM	None reported	Runs AI models in typical formats, i.e. ONNx
LACE-C3A	10-20	Microchip PolarFire™ MPF300T and MPF500T	SEU Immunity	Custom operation blocks configured by ONNx models. Reconfigurable on-the-fly

Table 2.1: Comparison table of several edge-computing platforms and LACE-C3A

2.3 Architectural Gap and Motivation

Existing edge-computing payloads demonstrate that high-performance on-orbit processing is achievable, but they expose several architectural weaknesses that limit long-term feasibility for small-satellite missions. Current systems rely heavily on advanced CPUs, GPUs, and VPUs manufactured on submicron processes. These parts offer high compute

density but introduce three critical drawbacks: elevated power consumption, sensitivity to radiation-induced upsets, and poor scalability when multiple workloads must be executed concurrently.

The primary limitation is architectural rigidity. Systems such as the Unibap iX5/iX10 and TelePIX TetraPLEX integrate heterogeneous COTS processors on monolithic boards. Each board is optimized around a fixed compute configuration. Adding a new model or increasing throughput requires adding an entire new module—often at tens of watts per module. This scaling model is incompatible with CubeSat-class power budgets. Moreover, the reliance on SRAM-based logic and modern CPU/GPU silicon leaves these systems vulnerable to single-event upsets. Mitigation is possible but incurs substantial overhead and does not eliminate the underlying sensitivity of the devices.

Single-FPGA approaches avoid some of these issues but introduce another constraint: resource saturation. Even mid-range FPGAs can typically host one significant ML model at a time once logic, DSP blocks, internal memory, and I/O are allocated. Attempting to pack multiple ML pipelines onto a single device leads to degraded timing closure margins, reduced maximum clock rates, and poor utilization efficiency. In practice, a single device becomes a bottleneck as soon as more than one algorithm is required.

The missing architecture in current systems is a modular, flash-based FPGA cluster that scales compute by adding devices rather than by increasing the size of a single device. Flash-based FPGAs eliminate configuration-bit SEUs, significantly reducing system-level fault susceptibility. Equally important, PolarFire-class devices include high-speed transceiver lanes capable of forming deterministic, low-latency interconnects between compute units. These links provide an order-of-magnitude more bandwidth than traditional parallel interfaces or microcontroller-mediated buses.

A Controller-Peripheral topology directly addresses these limitations. The controller FPGA concentrates external I/O, packetization, and workload routing, offloading all algorithm execution to peripheral devices. This keeps the controller's resource usage low and predictable. Peripheral boards can be scaled in count or capability depending on mission

needs, and different peripheral variants can be tailored for high-memory or high-throughput workloads. Because each compute unit is flash-based and independently reprogrammable in situ, the architecture supports model updates, reconfiguration, and algorithm rotation without modifying the hardware stack.

In short, no existing system combines radiation resilience, low power consumption, and scalable, reconfigurable compute in a form factor suitable for small spacecraft. The architecture proposed in LACE-C3A is designed to fill this gap: a modular cluster of flash-based FPGAs interconnected over high-speed SerDes, controlled by a dedicated routing FPGA, and capable of adapting to evolving mission requirements without additional hardware.

CHAPTER 3

Methodology

Designing a modular FPGA compute array required an engineering process that could translate architectural goals into a manufacturable hardware platform while controlling risk. The methodology used in this work followed a structured progression from establishing system requirements to validating hardware performance. This chapter outlines that process and the reasoning behind it. The specific design outcomes produced by this methodology are presented in Chapter 4.

3.1 Requirements and Constraints

The system architecture was shaped by system-level constraints identified below. These constraints come from the NASA Space Technology Mission Directorate (STMD) grant proposal submitted by the USU Center for Space Engineering (CSE).

1. Requirement 1 - LACE shall support up to three software defined radios (SDRs)
2. Requirement 2 - LACE shall support one Ethernet enabled camera
3. Requirement 3 - LACE shall support an external spacecraft interface including LVDS-UART and PPS
4. Requirement 4 - LACE shall provide an external interface to be enable use of SPI, UART, and USB
5. Requirement 5 - LACE shall ensure each Compute Unit (CU) has access to at least 1 Gb of external non-volatile memory
6. Requirement 6 - LACE shall support inter-FPGA data rates of at least 10 Gb/s
7. Requirement 7 - LACE shall provide at least one CU with access to at least 16 Gb of external volatile memory

These requirements informed the decisions made in both total system architecture and in the parts chosen in each subsystem. In order to support various applications, including Artificial Intelligence and Machine Learning (AI/ML) models, the system needed to include a method to increase the compute power without necessitating a redesign of the primary interface board. This additional requirement made necessary the requirement for 10 Gb/s inter-FPGA communication. Creating a scalable high-speed data interface from a data ingesting board to various computing boards enables expanding the system's data processing capabilities.

Several external data interfaces, including Ethernet, LVDS-UART, and USB-to-UART bridges were needed to ingest data from various data sources, including Software Defined Radios (SDR), cameras, and other sensors. The other main interface needed was a board-to-board interface to enable the high-speed data link between FPGAs. These interfaces, especially the high-speed ones, informed the physical constraints of the board, including board thickness, controlled impedance, and connector density. These constraints defined the design process and the choices made during implementation.

3.2 System-Level Design Workflow

The design process followed a top-down engineering workflow rather than an incremental or component-driven approach. The design flow is illustrated in Figure 3.1. Work began by establishing the Controller-Peripheral architecture and identifying the functional boundaries between those roles. With the overall framework defined, system requirements were decomposed into subsystem responsibilities, such as power distribution, memory, storage, high-speed transceivers, programming interfaces, and external I/O. Each subsystem was then evaluated to identify candidate devices capable of meeting its electrical and operational constraints.

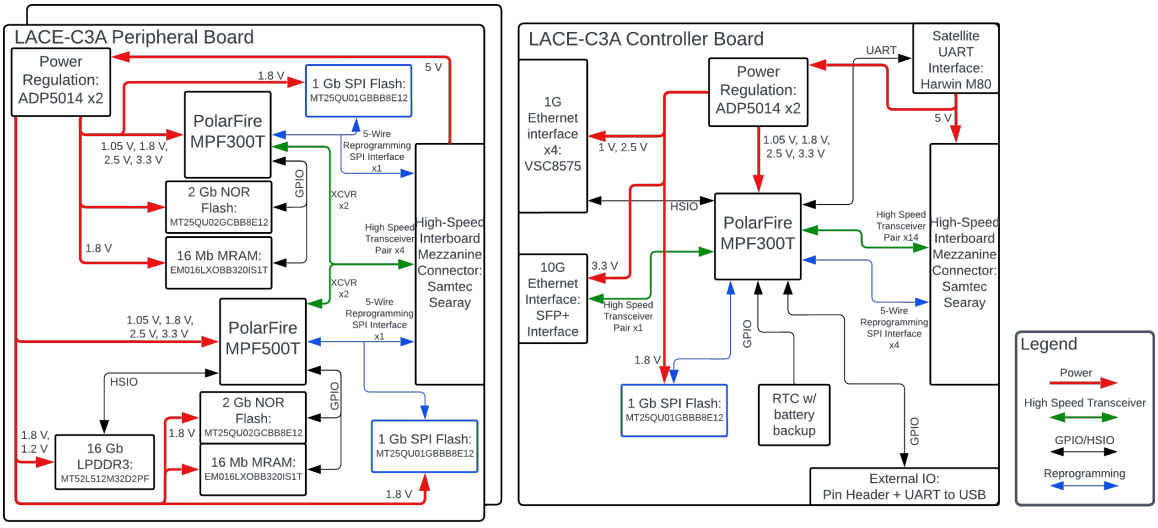

Fig. 3.1: Diagram of the basic interconnections between the controller and peripheral boards

Once subsystem expectations were understood, the boards were partitioned into a primary controller board and a family of peripheral boards. This partitioning was chosen because it allowed the compute capacity to scale independently of the core routing and I/O infrastructure. Schematic capture and preliminary electrical analysis proceeded in parallel, with power estimation, timing considerations, and physical footprint checks informing revisions as the design evolved. Analog subsystems — primarily the switching regulators and linear regulators — were simulated prior to layout to validate transient behavior, startup

sequencing, and stability. After all subsystems had been validated to the level possible in simulation, schematics were finalized and passed to layout for high-speed routing and mechanical integration.

3.3 Toolchain and Simulation Strategy

Several tools supported this process, each selected to evaluate a particular class of design behavior. KiCad 8.x was used for schematic capture because it provided full control over symbol creation, multiple hierarchical sheets, and the differential-pair routing constraints that would later be needed in layout. LTspice was used to analyze analog subsystems. The system's switching regulators, LDOs, and current-sense amplifiers each required transient and steady-state simulation to ensure that their behavior remained within device tolerances, particularly during startup where sequencing errors could damage the FPGAs.

The Microchip PolarFire Power Estimator complemented these simulations by providing realistic current estimates for each rail under worst-case load. These estimates were essential for sizing regulator channels and for verifying that layout decisions would not overstress the power network. MATLAB was incorporated into the workflow to automate component calculations, such as inductor and capacitor selection, and to evaluate sensitivity to component tolerances. MATLAB allowed quick iteration over ranges of values, reducing the risk of selecting marginal components based solely on nominal datasheet figures.

3.4 Schematic Capture Approach

The schematics were organized to mirror the architectural hierarchy of the system. Each board — controller, peripheral, and the SerDes breakout used for testing — was organized into its own KiCad project to keep a logical separation between the boards in a stack. Each KiCad project schematic file was divided into functional domains including FPGA cores, power regulation networks, memory interfaces, high-speed SerDes lanes, programming interfaces, housekeeping circuits, and connectors. This structure made it possible to trace requirements back to specific design elements and allowed each subsystem to be reviewed independently before layout. The hierarchical approach also reduced the likelihood of routing conflicts or overlooked constraints during later design stages.

3.5 Summary

This methodology established a structured process for transforming architectural requirements into validated hardware. The following chapter details the actual system implementation that resulted from this workflow, including the specific design decisions, simulations, and hardware outcomes that form the basis of the LACE-C3A platform.

CHAPTER 4

Implementation

This chapter presents the hardware implementation of the Controller and Peripheral boards, and a breakout board exposing several high-speed Serializer/Deserializer (SerDes) lanes for testing. Each decision was made to ensure conformance with the design objectives outlined in section 3.2. The chapter is divided into sections describing the major subsystems of each board and how they address the needs of the project.

4.1 The Controller Board

The Controller board provides the interface between external data sources and the compute fabric. Its role is to terminate incoming links, translate protocols, enforce power and timing constraints, and deliver packetized data to the Peripheral boards through the SerDes network. The design focuses on stable I/O handling rather than heavy computation, enabling the Controller to operate predictably under varying sensor loads.

4.1.1 FPGA

The Controller board has one main duty — to accept any incoming data and route it to the appropriate Peripheral FPGAs. To accomplish this task, the Controller FPGA needed to have sufficient compute resources to unwrap, parse, and then re-packetize incoming data and enough transceiver lanes to support as many Peripheral FPGAs as necessary for a given configuration.

Since the compute resources necessary to effectively route data are low, the main factor in selecting an appropriate FPGA was the number of available transceiver lanes. Microchip offers the PolarFire™ MPF100T, used in many low-power applications due to its compute resource density compared to its power draw. But the MPF100T only has eight SerDes lanes available. This would have been sufficient for this technology demonstration, but future

applications of this technology may require more options to connect Peripheral boards. With this in mind, the MPF300T was chosen due to its sixteen available transceiver lanes.

4.1.2 Power System

The Controller board's power architecture was designed to balance electrical efficiency, sequencing reliability, and minimal board footprint. Because the board serves as the system's primary interface to external power, its regulators needed to support the FPGA, memory devices, Ethernet subsystems, and GPIO interfaces.

The power network had to satisfy three major constraints:

1. Generate multiple voltage rails with precise sequencing as required by the PolarFire™ MPF300T.
2. Achieve high efficiency at low to moderate load currents to minimize overall power draw.
3. Occupy minimal PCB area to reserve space for the Controller's dense I/O routing.

The following sections describe the selected regulators, the simulations to verify their performance, and the sequencing needed to meet the design requirements of the PolarFire™ MPF300T FPGA and the LACE system at large.

Regulators

LACE-C3A was designed to be a low-power system when compared to other edge-computing systems. This means the power system had to efficiently create the correct rail voltages to power the FPGA and surrounding systems on the Controller board.

A multi-channel switching regulator was chosen as the ideal option for this design to satisfy the efficiency requirements. Choosing a multi-channel regulator reduced the amount of board space used by the power system. Care was taken to choose a regulator that had good efficiency over the several rails that were needed in the design. A switching regulator

was the natural choice for this due to its high efficiency across a range of regulated voltages and output currents.

The ADP5014 switching regulator was chosen for this design for its four-channel topology and capability of outputting up to 4 A on a single channel. The ADP5014 was also chosen because it is at least 85% efficient for 1 A loads across all the required voltages.

The ADP5014 datasheet [9] provides a detailed example circuit that provided a starting point for the final design. Figure 4.1 shows the core voltage and IO banks of an FPGA, a DDR3 memory module, and a Flash memory module all being powered by one chip.

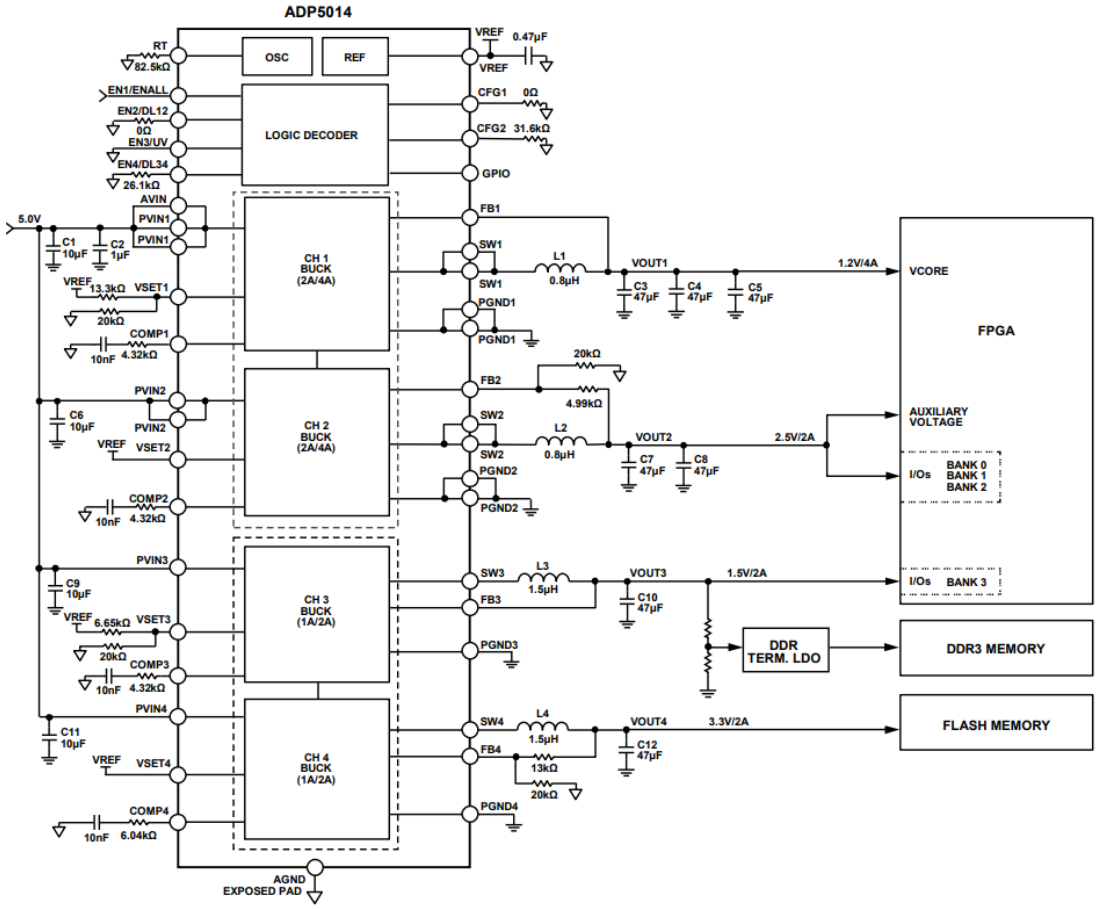

Fig. 4.1: Example circuit of the ADP5014

This design was replicated to provide the needed voltage levels to the Controller FPGA, memory modules, Ethernet circuitry, and other external interfaces.

The process of selecting the voltages necessary to ensure the needed operations was a balance between selecting as many chips that used similar voltages as possible and adding new rails as necessary.

The final power system design used two ADP5014s to supply six different rails to the components on the board.

Voltage Rail	**Current Source Capacity**	**Expected Power Usage**	**Devices Powered**
1.05	4 A	4.1 W	MPF300T Core Voltage

Table 4.1: Controller Board Voltage Rails

The core voltage required by the PolarFire™ MPF300T is 1.05 V. According to Table 4.2 in the PolarFire™ FPGA Datasheet [10], the core voltage of the FPGA must be 1.05 V to utilize the maximum transceiver lane data rate of 12.7 Gbps.

PolarFire™ devices use the 2.5 V rail as a reference voltage for the transceiver clocks and other banks. 2.5 V is also used by the Ethernet PHY and Ethernet clock synchronizer.

The 1.8 V rail was used as a bank voltage for the MPF300T and other logic devices. The same is applied to the 3.3 V rail.

The 1.0 V rail was used only by the Ethernet PHY.

The 3.0 V was used by the 1 Gb MRAM module.

Power Simulation

Simulations of the ADP5014's performance confirmed the regulator's output met the requirements of the sequencer input and the voltage-stability requirements of the various devices connected to it. Combining these results with MATLAB automated component selection. An example MATLAB script is included in Listing A.1.

The ADP5014 datasheet includes equations for all of the external components, as well

as recommendations for programmable settings, such as the switching frequency and GPIO functionality. These equations were used in the MATLAB script to generate values that were then used in the LTspice simulation. Figure 4.2 shows the LTspice setup for simulating the first two channels of one of the ADP5014s on the Controller board. Figure A.1 shows the results of simulating this circuit.

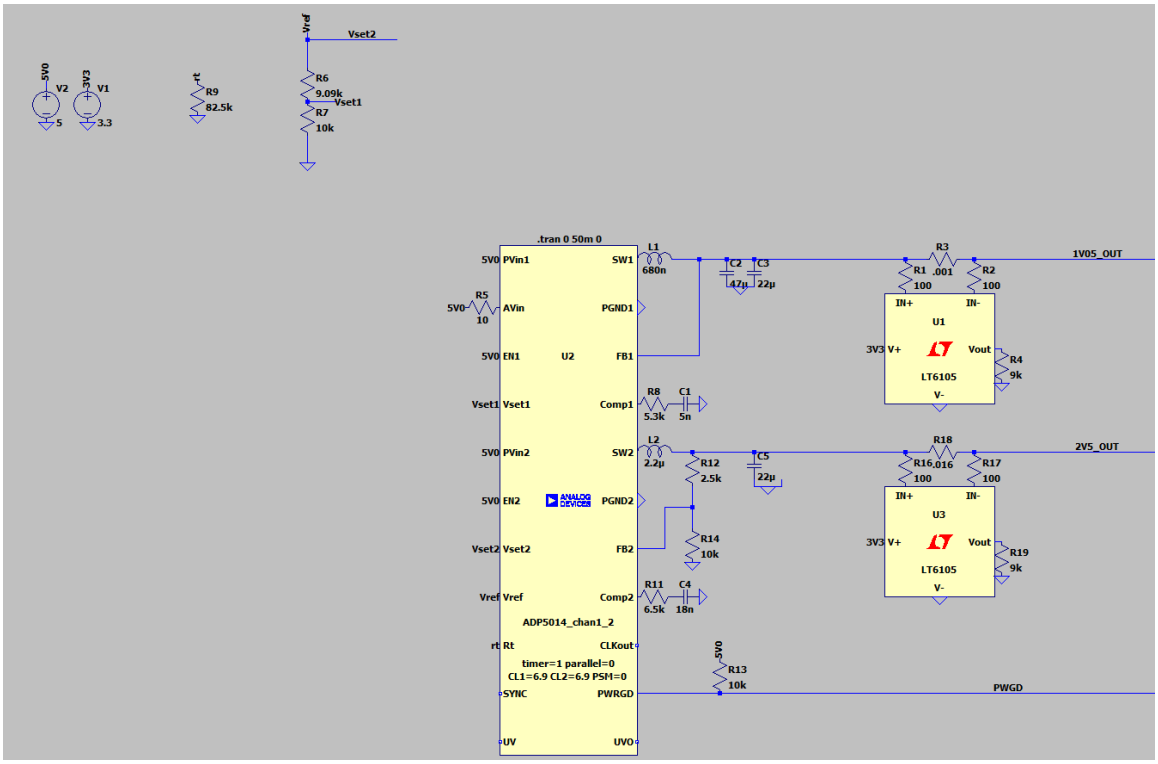

Fig. 4.2: LTspice schematic of one of the Controller board's ADP5014s

The simulation showed that the regulator is able to source the expected voltages within 9 ms of being powered. The produced voltages were steady within 1.2 mV of the expected value, as shown in Figure 4.3.

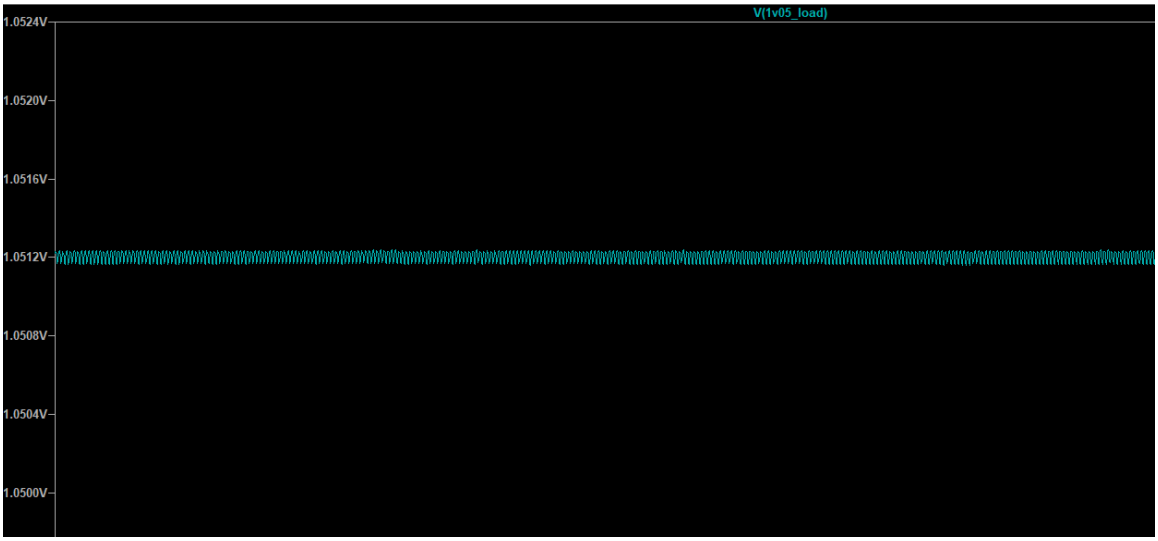

Fig. 4.3: Simulation results of steady regulated voltage

Sequencing

The PolarFire™ family of devices has specific power sequencing needs to ensure proper operation. These glitches present themselves in the IO banks during power up and power down. There are two possible scenarios for these glitches to occur on the current configuration for the Controller board. The first is on the 3.3 V IO bank. The second is on the 1.8 V IO bank.

All IO banks are powered primarily by the VDDIx bank (x being each bank). Each of the banks can also draw power from a VDDAUXx bank. For any IO bank that has its VDDI and VDDAUX at the same potential, the VDDI and VDDAUX should be powered by the same regulator. This is the case for the 3.3 V logic IOs on the Controller board, preventing IO glitch.

The PolarFire™ Board Design Guide specifies that IO banks with a logic level less than 2.5 V must have their VDDAUX powered by 2.5 V. This meant that more consideration had to be given to the sequencing for this bank. To prevent IO glitches, VDDAUX must come up first, followed be VDDI for the bank. This has been done to ensure no IO glitches occur.

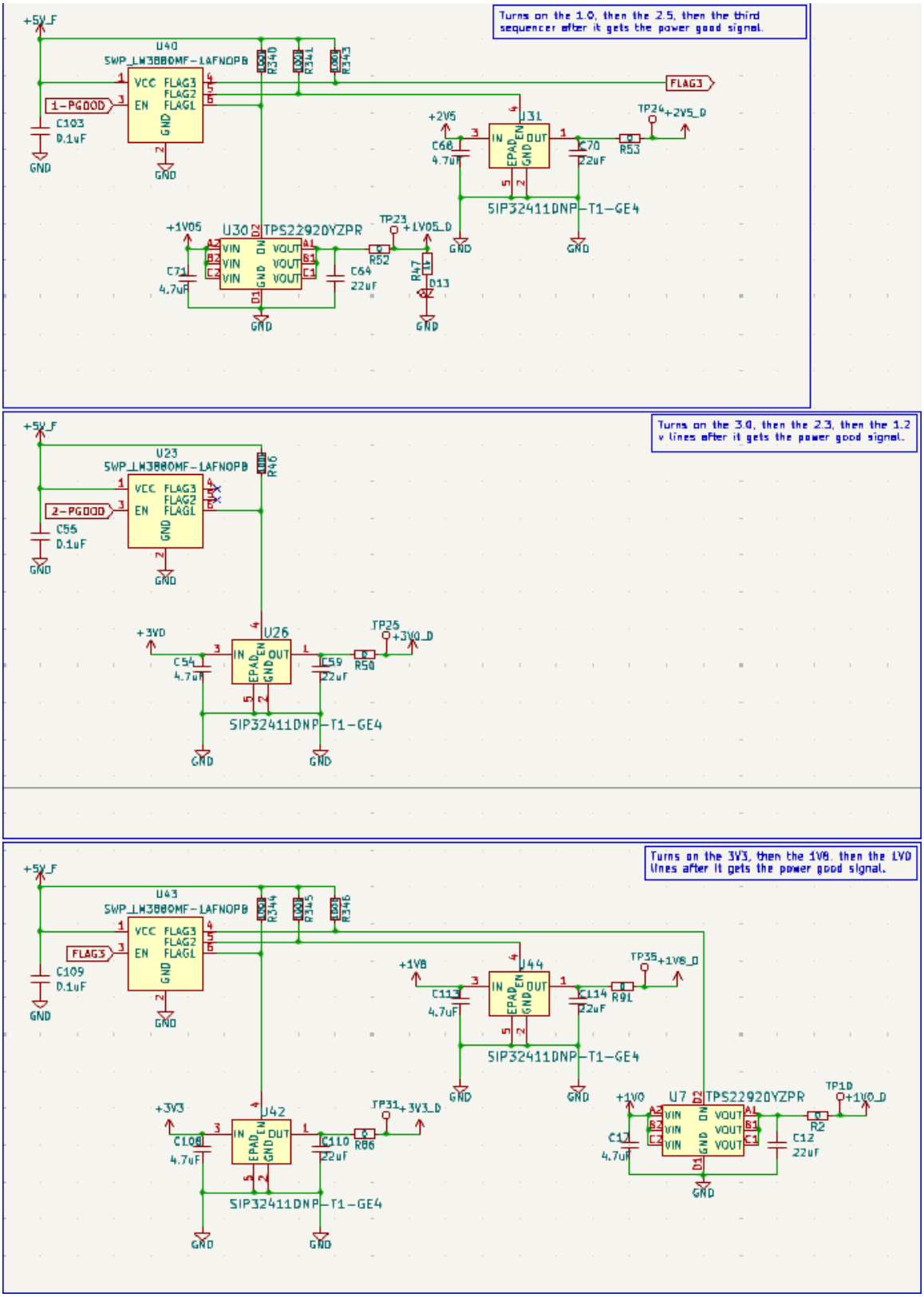

Fig. 4.4: Power sequencing for the Controller board

Two ADP5014 regulators, when paired with several LM3880 sequencers, were able to provide six sequenced rails. This ensured the FPGAs and other devices on the board would power up correctly.

4.1.3 Memory

The Controller board was designed with modularity and flexibility in mind. This meant that, although there were no plans from a firmware perspective to use external memory when the conceptual design of the board was finished, the design allowed for future functionality. With this in mind, a memory module with high tolerance for the space environment and more usable capacity than necessary was chosen. This module was the 1 Gb Micross MYXSMS01GP32 [11]. This module was chosen because its spin-torque magnetoresitive MRAM technology is radiation tolerant by nature of its design.

This MRAM module is a novel product from Micross that is unique apart from one other product from Everspin. It is unique because of the high memory density in a natively radiation-tolerant package.

This MRAM operates at a single core voltage for memory and communication operations. As mentioned earlier, this core voltage is 3.0 V and is only used by the MRAM module.

Unfortunately, this MRAM module was cost prohibitive to populate on the board. The footprint was left in the design to implement later.

4.1.4 Programming

The LACE Controller board has a single programming interface. This interface allows an external programmer to program the FPGA. The programming interface uses a simple 1.27 mm pitch twelve pin connector paired with a programmer breakout board designed by students in the Center for Space Engineering. The programming interface connects to two interfaces on the MPF300T and its associated programming devices.

The first option for connection is the JTAG programming interface. This is an interface built-in to the PolarFire™ family that uses the JTAG standard to program new images onto

the FPGA fabric. The programming is done by a Microchip FlashPro™ device. It is able to quickly load a bitstream file from a computer for programming.

The second option for programming is the built-in SPI interface. This SPI interface from the programmer breakout board serves two purposes. It can communicate with the PolarFire™ fabric SPI interface for direct programming. This SPI interface is also connected to a 1 Gb NOR Flash. This NOR Flash is the MT25QU01GBBB8E12-0AUT from Micron. It is used by the FPGA to store reprogramming images. The NOR Flash can be programmed through the programmer breakout board, typically by a single-board computer or by the FPGA to which it is connected. The NOR Flash can store a golden image, in case the FPGA needs to be reprogrammed to a known good state, or some number of other images the FPGA can choose from to program itself.

4.1.5 Interfaces

The LACE Controller board was designed to take in data from a variable number of sensors, ranging from ionospheric sensor suites to traditional optical cameras. To accomplish this task, the Controller board was designed with several interfaces to ingest data. These included Ethernet interfaces, dual USB type A headers, a GPIO breakout interface, and an LVDS UART interface for connecting to an external computer and for power.

The Controller board also needed to be able to transmit high-speed data, low-speed data, and power to the Peripheral boards. This was accomplished with a mezzanine-style board-to-board connector.

Ethernet

1G Ethernet on the LACE Controller board was implemented in the traditional manner with RJ45 connectors, Ethernet timing synchronization, and an Ethernet PHY. This topology allowed the board to plugged into a traditional Ethernet switch and communicate with other devices on the network. The Ethernet subsystem of LACE also included an SFP+ connector and cage to allow for higher speed interfaces, such as 10G copper or fiber-optic Ethernet.

The 1G Ethernet subsystem was designed around the Microchip VSC8575 Ethernet PHY. An Ethernet PHY is the physical translation layer interface that takes in differential transmit and receive pairs and splits them into the eight differential pairs of the Ethernet protocol. The VSC8575 supports up to four 1G Ethernet interfaces. To satisfy the requirements of the LACE project and ensure enough external connectivity, all four interfaces offered by the VSC8575 were implemented. Each of the four ports on a four-port RJ45 connector was connected to the outputs of the PHY through Ethernet transformers. The transformers protected the sensitive electronics on the Controller board from induced currents on a connected Ethernet cable.

Timing synchronization for the 1G Ethernet subsystem was handled by the Microchip ZL30364. This chip takes in several reference clocks and produces the necessary clocks for Ethernet timing. Those clocks were fed to the Controller FPGA and to the VSC8575.

This Ethernet subsystem implementation satisfies the requirement that the LACE system will be able to support data from up to three Software Defined Radios (SDR) and one camera (Requirements 1 and 2 in section 3.1).

LVDS UART Interface

The LVDS UART interface used a Harwin M80-5T11205M1 connector to enable UART communication and a power interface in one package. This connector has twelve data pins that were setup to enable a differential UART Rx and Tx line via LVDS. After adding ground pins for return paths for the differential lines, there were enough data pins to add a dedicated GPS PPS signal line and a spare GPIO pin to the connector. This connector is unique in that it has two large pins specifically for power and ground connections to a power supply, like a spacecraft's Electrical Power System (EPS). The pinout that was implemented is shown in Figure 4.5. This connector satisfies Requirement 3 in section 3.1 to provide an external spacecraft interface with a UART LVDS connection and a dedicated GPS PPS input.

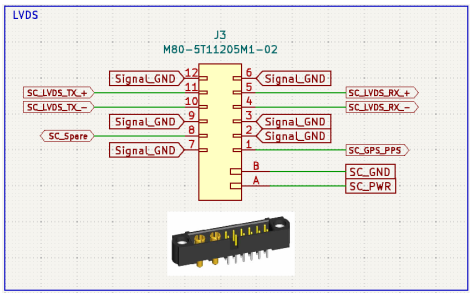

Fig. 4.5: Schematic capture of the LVDS UART connector

USB Interface

The dual USB type A headers utilize a stacked connector with two ports. Each port was connected to its own USB to UART bridge from FTDI™. The FT230XQ is a converter that takes in a differential USB signal and translates it to a simple UART signal. This translator enabled USB 2.0 Full Speed communication (12 Mbps). This interface can be used with native USB data sources like cameras and software-defined radios. This USB interface partially satisfies the requirement to provide an external interface for incoming data (Requirement 4 in section 3.1).

GPIO Interface

The GPIO breakout interface exposed sixteen GPIO from the Controller FPGA to a Hirose™ DF11 connector. Future engineers can design cables and breakout boards that will convert these broken out pins to any numbers of external sensors. This GPIO breakout connector satisfies the rest of Requirement 4 in section 3.1 to provide several low-speed external data interfaces.

Board-to-Board Connector

The Controller board exposes its high-speed SerDes lanes, programming interfaces, and power rails through a high-density Samtec SEARAY™ mezzanine connector. This connector forms the electrical and mechanical interface to the Peripheral boards and provides the controlled-impedance signaling environment required for 12.7-Gbps transceiver links. The

detailed pin allocation and mechanical selection are discussed in section 4.2.5, where the Peripheral implementation is described.

4.1.6 Manufacturing

The LACE Controller board was laid out by Orbis Engineering Inc. Orbis Engineering has provided layout services for several previous boards designed by the USU Center for Space Engineering. They were able to complete the layout quickly while routing the high-speed sections of the board correctly.

The placement of the components on the top layer of the board is seen in Figure 4.6. This placement allowed for all connectors to be easily accessible and noisy components to be isolated from high-speed traces.

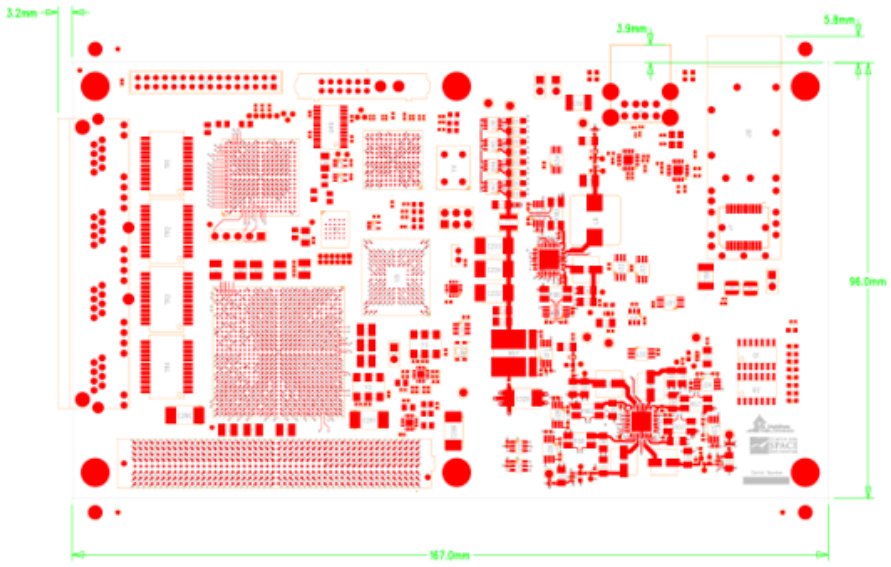

Fig. 4.6: Controller board layout

Board fabrication was completed by Excello Circuits.

The board was assembled by Xanfab Inc. They have also previously done work for the Center for Space Engineering and have delivered consistent, high-quality results.

4.1.7 Summary

The Controller board was implemented in a way to satisfy the data receiving and transmitting requirements defined in section 3.1. The implementation focused on interfaces to ingest data, an FPGA powerful to receive, parse, packetize, and transmit data with very low latency, and an interface to power and quickly transmit data to a set of Peripheral data processing devices.

4.2 The Peripheral Boards

The initial idea of the LACE-C3A system was to create a modular edge computing system that could be customized to a range of data processing applications. The plan to accomplish this included designing several variants of the Peripheral board with differing compute resources like FPGAs, microcontrollers, and high-speed memory.

Due to time constraints on this project, one variant of the Peripheral board was designed. This Peripheral board design had one PolarFire™ MPF500T and one PolarFire™ MPF300T. These two FPGAs provided flexibility for running various algorithms and data processing operations. Each FPGA can be programmed to use only the resources necessary to run the algorithm, enabling multiple algorithms to run on a single compute unit.

4.2.1 FPGAs

The heart of the LACE-C3A Peripheral board is the processing devices that will run various AI/ML algorithms and models. As explored in chapter 2, there are many options to accomplish this task. FPGAs were the clear choice when attempting to balance power efficiency and raw compute power. Since this was a prototype of a device that will fly on small-satellites, radiation tolerance was also a factor to consider.

For these reasons, the PolarFire™ MPF300T and MPF500T were chosen as the main compute units on the Peripheral board. For smaller algorithms and applications, the MPF300T provides ample compute resources including

- 300K logic elements

- 924 Math blocks
- 952 20 Kb LSRAM blocks
- 2772 64x12 uSRAM blocks
- 459 Kb uPROM

The larger MPF500T had even more resources to run the most demanding models that can fit on a single PolarFire™ FPGA, including

- 481K logic elements
- 1480 Math blocks
- 1520 20 Kb LSRAM blocks
- 4440 64x12 uSRAM blocks
- 513 Kb uPROM

The PolarFire™ line of FPGAs uses Flash-based fabric, giving them natural radiation tolerance and SEU immunity. The PolarFire™ line is also power efficient when compared to traditional compute devices, like CPUs and GPUs, and even other competing FPGAs.

4.2.2 Power System

The power subsystem of the Peripheral board was designed to replicate the efficiency and sequencing stability of the Controller board while delivering higher current to the compute FPGAs. Because both boards operate from a common 5 V bus, maintaining consistent regulation topology simplified design reuse and reduced component diversity. Each PolarFire™ FPGA requires multiple tightly regulated rails with defined sequencing, so the Peripheral board uses the same Analog Devices ADP5014 quad-channel switching regulators that power the Controller board. This device provides four independently programmable outputs capable of up to 4 A per channel and an efficiency exceeding 85 % for

1-3 V rails—adequate for the increased load of the MPF500T device. Two small Linear Drop-Out (LDO) regulators were added to the power system to provide a low-noise rail for each FPGA's VDDA input.

Switching Regulators

The ADP5014 was chosen due to its 4 adjustable outputs and high current capacities. The Peripheral board required six different potentials for the various components. The rail potentials designed were:

- 1.05 V (4A maximum current draw)
- 1.05 V (8A maximum current draw)
- 2.5 V
- 1.8 V
- 3.3 V
- 1.2 V

These rails supplied power to both FPGAs, GPIO banks, and external memory.

The primary power rails—particularly the 1.05 V core rails—dominate the total current demand of both FPGAs. According to the Microchip PolarFire™ Power Estimator Tool [12], the MPF500T draws up to 5 W at full utilization, approximately 4 W of which is consumed by the core supply. Applying Ohm's Law,

$$I = P/V = 3.967/1.05 = 3.778 \text{ A}$$

This value approaches the single-channel current limit of the ADP5014. To ensure headroom under transient conditions and to minimize thermal stress, channels 1 and 2 of one ADP5014 were configured in parallel, doubling the available current to 8 A. The manufacturer's datasheet [9] provides a validated parallel-channel configuration that maintains

current sharing and loop stability. This design decision balanced board area, efficiency, and reliability.

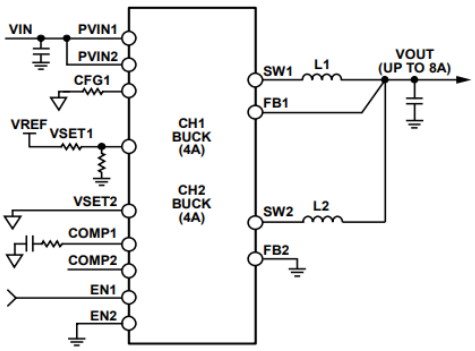

Fig. 4.7: ADP5014 parallel channel example circuit

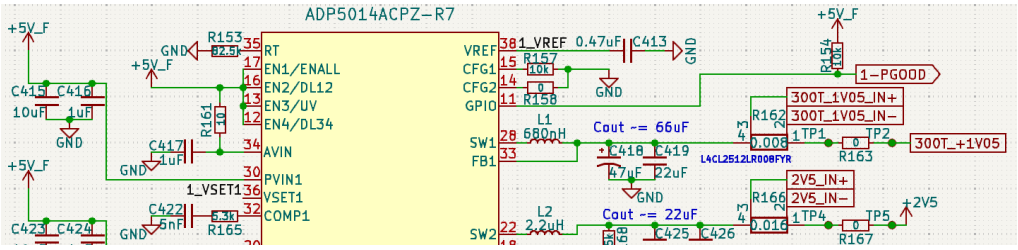

Fig. 4.8: ADP5014 1V05 channel schematic capture

The Microchip power estimator tool calculates the PolarFire™ MPF300T power draw at nearly 100% utilization to be 4.1 W. As with the MPF500T, most of this draw is on the core voltage rail, 1.05 V. Applying Ohm's law again, the current on the core voltage rail is 2.61 A. This value falls squarely within the acceptable margin of a single 4 A channel of the ADP5014. The design for this is found in Figure 4.8.

Linear Regulators

The Microchip board design user guide recommends VDDA receives a quiet rail [13]. VDDA supplies the power to the SerDes transceiver subsystem. It needs quiet power to ensure the high speed signals are not interrupted by power supply noise.

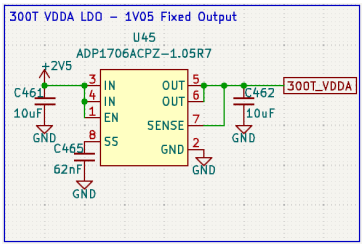

Fig. 4.9: ADP1706 schematic capture

The chip chosen to supply this power was the ADP1706. This is a small LDO that can source up to 1 A of current. The ADP1706 is a very small LDO with several fixed voltages in addition to an adjustable version. VDDA is required to be the same potential as the core voltage — 1.05 V in this case. The ADP1706 comes in a fixed 1.05 V version, and it was used in this design. It has low external component requirements. Only 3 passive components were required to make the chip fully functional. The design is shown in Figure 4.9.

Housekeeping

Since this system was an engineering unit, a housekeeping subsystem was implemented to monitor the current and voltage on the main voltage rails. This subsystem was built using the LT6105 precision current sense amplifier from Linear Technology paired with the ADS1118 16-bit ADC from Texas Instruments. This pairing allowed the housekeeping subsystem to accurately measure the current on all the voltage rails. The ADS1118 is also able to measure the voltage rails using resistor dividers to bring the voltages into its effective sense range.

The current sense circuitry functioned by using a current sense resistor placed in series with the outputs of each voltage rail. The current flowing through each rail created a small voltage. This voltage was then measured by the inputs of the LT6105, producing a voltage that was amplified by the gain resistor on the output. The gain resistors have been tuned on all rails to output a voltage within the effective input range of the ADS1118.

The housekeeping ADCs reported their measured voltages via a SPI interface. Since the Controller acted as the interface to outside the system, it needed to ingest the data

from the housekeeping subsystem and send it out the main data interface. There were two options for getting this data to the Controller. The first was to route the data lines from the ADCs to one of the FPGAs on the Peripheral board. The Peripheral board would have then packetized the data and sent it to the Controller FPGA via the transceiver interface. The second was to route the SPI data lines directly to the Controller FPGA via the mezzanine connector.

The second option to route the SPI lines from each Peripheral board to the Controller board was chosen to avoid consuming fabric space on the Peripheral FPGAs or bandwidth on the high-speed data link. A MUX was implemented to select between the Chip Select (CS) lines and the data lines were shared between the ADS1118 devices.

The ADS1118 ADCs are able to share the Clock (SCLK), Data In (DIN) and Data Out (DOUT) lines. The ADS1118 datasheet [14] specifies that multiple devices can be connected as long as they have unique CS lines. To reduce the pin count going to each Peripheral board, a 2-to-4 MUX was implemented to select between the four ADC's CS inputs, as shown in Figure 4.10.

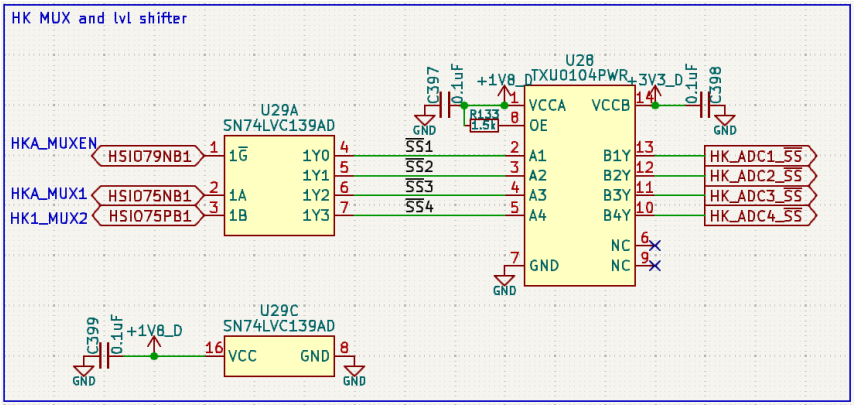

Fig. 4.10: Housekeeping Chip Select MUX

Summary

In summary, the Peripheral board's power system combines high-efficiency switching regulation with localized low-noise linear regulation to meet the stringent current, sequenc-

ing, and noise requirements of two PolarFire™ FPGAs. Parallelized ADP5014 channels ensure sufficient headroom for the MPF500T core, while identical regulator topologies across both board classes simplify layout and system integration. The housekeeping subsystem gives accurate reporting of the power conditions of the various rails in realtime. This configuration maintains stability across all operating modes.

4.2.3 Memory

The memory architecture on the Peripheral board was designed with to balance performance, reliability, and cost. The memory architecture also needed to support the variable workloads from both the MPF300T and MPF500T computes devices. Each FPGA requires both high-speed volatile memory for algorithm execution and non-volatile storage for persistent weights and configuration data. The architecture therefore combines three memory technologies — MRAM, NOR Flash, and LPDDR3 — to satisfy these requirements.

Initially, the two smaller memory devices, the MRAM and NOR Flash, were going to be a single device that seemed to be tailor-made for this application. This device was the Micross MYXSMS01GP32 1 Gb MRAM. As mentioned in subsection 4.1.3, this device is prohibitively expensive. The alternative memory topology used to satisfy the requirements in section 3.1 is described in the following sections.

Non-Volatile High-Speed MRAM

The first device to replace the 1 Gb MRAM from micross was the Everspin EM016LXOBB320IS1T. It was chosen because it uses the same spin-torque MRAM technology, retaining intrinsic SEU immunity and low-latency random access while reducing cost by an order of magnitude. Although its capacity is lower (32 Mb), its 400 MB/s read and write speeds and sub-400 mW operating power make it well suited for high-use intermediate data storage such as partial algorithm outputs or temporary feature maps. This trade-off maintains the LACE system's resilience to radiation-induced bit errors.

Non-Volatile Bulk Storage

To store persistent data such as model weights and configuration instructions, a 2 Gb Micron MT25QU02GCBB8E12-0AAT NOR Flash device was integrated. NOR Flash was selected over NAND alternatives because of its superior radiation tolerance and random-read performance, both important for in-orbit reliability and dynamic model loading. With read/write throughput up to 90 MB/s, this device enables the FPGAs to fetch model parameters without introducing a bandwidth bottleneck.

This memory device satisfies the requirement that each compute unit shall have at least 1 Gb of external non-volatile memory (Requirement 5 in section 3.1).

Volatile High-Density LPDDR3

A dedicated 16 Gb LPDDR3 interface was chosen and implemented for the MPF500T since it will run the most data-intensive AI/ML models. The MT52L512M32D2PF from Micron Technology was chosen for its low power compared to traditional DDR3 memory and high throughput at 1866 Mega-Transfers per Second (MT/s). The high throughput allows this device to receive, store, and transmit high volumes of data for real-time image classification, bulk sensor-data processing, or other high-performance algorithms. This LPDDR3 module is only connected the the MPF500T since its primary purpose is to run the most resource-intensive AI/ML workloads.

In addition to the high-throughput of LPDDR3, it also has a lower power draw when compared to traditional DDR3 memory. Typical DDR3 memory typically consumes 1.5-2.5 W while under heavy load. In comparison, the MT52L512M32D2PF LPDDR3 consumes only 350 mW while performing heavy read/write operations. This lower power draw at the same high-speed allows the memory to support longer sustained computation algorithms in low-power environments.

This memory device satisfies the requirement that at least one compute unit shall have at least 16 Gb of external volatile memory (Requirement 7 in section 3.1).

Summary

These three devices form a memory architecture that allows the two FPGAs on the Peripheral board to handle fast intermediate data, provide persistent instruction and weight storage, and handle high-speed volatile data for large computational workloads.

The layered approach of using a combination of MRAM, NOR Flash, and LPDDR3 optimizes cost and performance while still offering radiation resilience. These specific parts also offer a path for upgradeability in the future with pin-compatible parts with higher memory density and the same layout requirements.

In combination, the memory subsystems enable each Peripheral FPGA to operate autonomously and reconfigure dynamically while remaining within the system's power and reliability constraints.

4.2.4 Programming

Each FPGA on the Peripheral board had its own programming interface. Each interface allowed its FPGA to be programmed from one of two sources. They each had a small 12-pin connector from Harwin. The M55-7101242R is a keyed 1.27 mm pitch connector with cabling that allows it to be plugged directly into the external programmer device. The keying protects the programmer and the FPGA programming interface from damage.

The programming interface also consisted of a large storage device to hold bitstream images. This device is the MT25QU01GBBB8E12-0AUT NOR Flash from Micron Technologies. This is the recommended design from Microchip when a user wishes to use any of the automatic reprogramming features of the FPGA.

Each 12-pin connector was connected to two different programming interfaces on each FPGA: the SPI interface and the JTAG interface, shown in Figure 4.11. While JTAG is typically used more often by external programming devices, the SPI interface is connected to both the external connector and the large inter-board mezzanine connector. The external connector allows an external device, like a Raspberry Pi, to program either the FPGA or the NOR Flash device with a new image. The mezzanine connector allows the Controller FPGA to program either the FPGA or the NOR Flash as well. The user can switch between

the two options using a jumper on a 3x5 pin header on the board. This flexibility allows for several options to reprogram each FPGA.

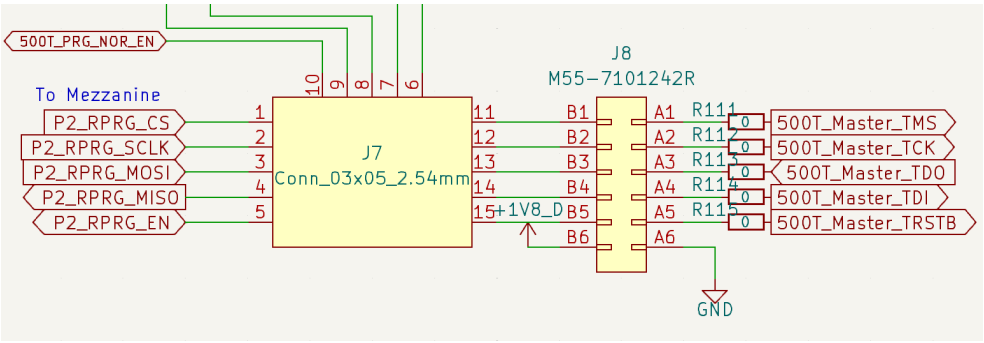

Fig. 4.11: Programming interface schematic capture

Since the SPI reprogramming lines pass from the Controller board through all the Peripheral boards on the mezzanine connector, an external device, like a microcontroller, could be plugged in and given access to reprogram the Peripheral FPGAs as well.

4.2.5 Interfaces

While the Controller FPGA needed to have many interfaces for ingesting data from outside the LACE system, the Peripheral FPGAs only needed to ingest data from one source: the Controller FPGA. With this in mind, the Peripheral board design had far fewer interfaces.

Selection Considerations

An important consideration that was made while selecting the interface for board-to-board communications was the ability to pass all the necessary signals between all the boards in a LACE stack. Two options to accomplish this task became clear as the needs of the system were discussed.

The initial conception of the LACE project included the ideas that the Peripheral FPGAs would be able to be reprogrammed on the fly by the Controller FPGA, high-speed communication would be done with the native transceiver lanes, and there would be a

low-speed communication interface between the Controller and Peripheral FPGAs. The Controller board would also provide the power interface for the Peripheral boards via some board-to-board connector. This board-to-board connector would have to have enough pins to support all of these functionalities.

Using the native PolarFire™ programming interface, five pins per FPGA are required to be passed between the boards. Each FPGA was given two TxRx pairs for redundancy, meaning each FPGA has eight pins for the transceiver interface. After using the power estimator tool and figures from data sheets, it was determined the connector would have to support current levels of up to 10 A, further increasing the number of pins depending on the capabilities of the chosen connector. The connector also needed to be able to support sufficient return paths for all of the data signals and provide a solid reference plane between the Controller and Peripheral boards. This meant that there should be a ground pin for each transceiver pin, each GPIO pin, and each reprogramming interface pin. There should also be banks of ground for the reference plane. The low speed communication interface was left without a definite pin count; it would use any remaining pins.

These pin counts have been condensed in Table 4.2, assuming the typical LACE system as described in this thesis: one Controller board and two Peripheral boards with two FPGAs each.

Signal Type	**Base Pin Count**	**Full System Pin Count**
Reprogramming	5	20
High-Speed Transceiver	8	32
Low-Speed GPIO	12	48
Power	-	24
Ground	-	150
Total	25	274

Table 4.2: Table of board-to-board connector pin counts

Two methods of accomplishing the board-to-board communication emerged once the necessary communication types were determined. The first option was to use a flat-flex cable (FFC) that would flex down to the next board in the stack. This concept is shown in Figure 4.12. This design had the benefit of being very flexible and relatively high density for its size.

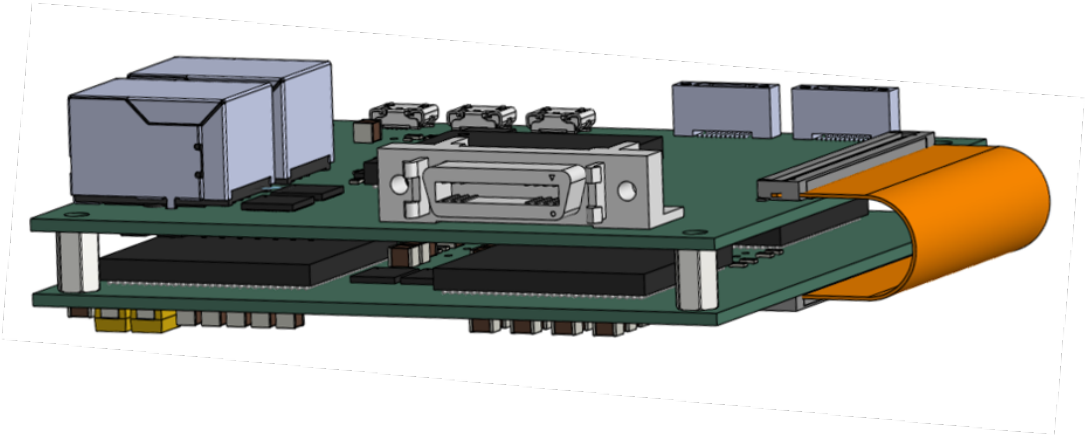

Fig. 4.12: LACE System concept with a Flat-Flex Cable

The second option was to use a high-density board-to-board connector to assemble the LACE system stack. This option comes with the benefit of extremely high pin count and strong mechanical stability when assembling a stack. Figure 4.13 shows the mechanical strength of the high-density mezzanine connector concept.

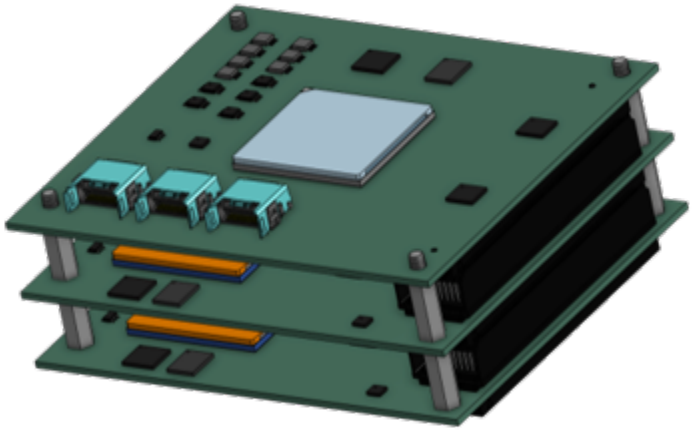

Fig. 4.13: LACE system concept with a mezzanine connector

Board-to-Board Interface

The main interface for transmitting data to and receiving data from the Controller FPGA was chosen to be a high-density mezzanine connector from Samtec. The Samtec SEARAY™ connector line offers up to 56 Gbps data rates — much higher than the 12.7 Gbps max speed of the PolarFire™ transceivers. The SEARAY™ connectors have several options for high-density pin counts. The variant chosen for this design has eight rows of fifty pins each, totaling to 400 available pins to support the data and power needs of the system. These connectors also have several different height options to ensure components on top of a board won't make contact with the components on the bottom of the next board in the stack. Taller height options will also make testing easier once a stack has been fully assembled.

Other Interfaces

Several interfaces were implemented on the Peripheral board to aid in testing and development. These interfaces are two PMOD-style connectors. Each FPGA on the Peripheral board has access to one of these interfaces. These allowed external devices, such as sensors, microcontrollers, or even simple LEDs to interface with either FPGA to extend the functionality or enable another avenue for communication.

The final interface added for testing and development was a small bank of LEDs. There

were eight LEDs a user can use to output data on the state of an algorithm running on each FPGA. These can also be used during the bring-up and testing process to establish signs of life on each FPGA.

4.2.6 Manufacturing

Due to the complex nature of the high-speed elements on the Peripheral board design, including LPDDR3 and the transceiver lanes, the layout portion of the Peripheral board was outsourced to a company called Viridian Labs. They will take special care to route the sensitive signals in a way that protects their data integrity.

Manufacturing will not begin until this layout services are finished.

4.2.7 Summary

The Peripheral board was implemented to satisfy the data processing and compute resource requirements in section 3.1. The design focused on powerful FPGAs, a high-speed interface to transmit and receive data to and from the Controller FPGA, and sufficient resources, like external memory devices, to accomplish the various data processing and algorithm needs of the LACE system.

4.3 SerDes Breakout

The USU Center for Space Engineering provided two Microchip MPF300T Dev Kits for testing in the various stages of the project. These boards each have a transceiver lane pair available for high-speed communications. The Controller board, as designed, only has exposed transceiver lanes on the high-density mezzanine connector. A breakout board with two transceiver lane pairs was designed to interface with the high-density connector and the SMA connectors on the MPF300T Dev Kits for testing before the Peripheral boards became available.

4.3.1 Design Considerations

High-speed design requires several considerations to be successful. The most important when designing this breakout board were trace width, trace spacing, and impedance matching [15].

JLCPCB was the most cost-effective quick-turn fabricator that offers controlled impedance manufacturing. JLCPCB provides a trace width calculator to calculate the proper constraints for various dielectric materials they have available. The transceivers on the PolarFire™ devices require 100 ohm differential impedance. To accomplish this on this board, a trace width of 0.1326 mm and a trace spacing of 0.25 mm. This ensured that, when paired with the 3313 prepreg dielectric, the traces would have the required differential impedance.

Another consideration made to the design was in the routing. This was a relatively simple routing job in terms of placement of components and connectors. The complexity arose when routing the SerDes lanes. Since the lanes are rated to up to 12.7 Gbps, care had to be taken to eliminate reflections in the signal. Impedance control solved a portion of that problem, but there are other design considerations as well. A significant aid to reducing noise in high-speed signal lines is to use curved traces rather than the more traditional 45° corners when turns are necessary [16]. Eliminating sharp angles reduces an area of impedance change, thereby reducing reflections in the line. The curved traces are shown in Figure 4.14.

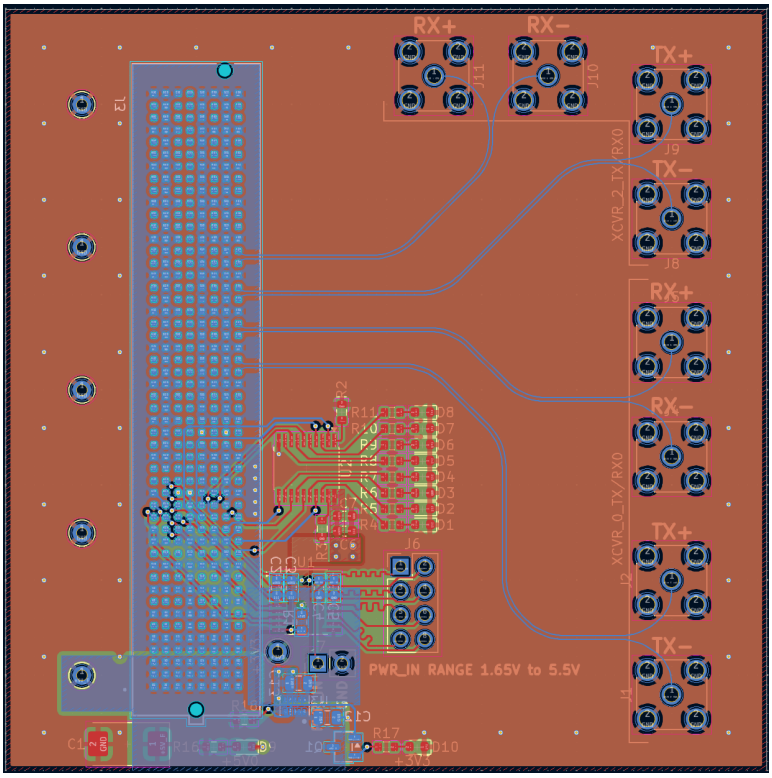

Fig. 4.14: SerDes Breakout Board Layout

4.3.2 Parts

This breakout board was relatively simple in terms of parts. The MPF300T Dev kits used SMA connectors to expose a transmit and receive pair. The breakout board used eight RF2-04A-T-00-50-G SMA connectors from Adam Tech. These connectors provided the required 50 ohm single-ended impedance necessary to maintain a clean signal.

The breakout board used the male mate to the female Samtec mezzanine connector on the Controller board. The mezzanine connector has 14 SerDes transceiver pairs exposed. To enable testing with the MPF300T dev kits, two transmit and two receive pairs were routed from the mezzanine connector to the eight SMA connectors.

The breakout board had several quality-of-life functions to improve testing. The first was eight LEDs that were controlled through an LED buffer, the Texas Instruments SN74HC244PWR. These are able to be driven by the Controller board, offering an interface

for simple data output. The other component in this category was the Texas Instruments TXS0108EPW. This is an IO buffer that connected eight Controller board IO to a standard 2.54 mm pitch pin header. This can be used for any IO needs.

The final major component on this breakout board was the power regulation. The Controller board provided only a 5 V rail on the mezzanine connector. The LED buffer and IO buffer required 3.3 V for proper operation with the IO banks they were connected to on the Controller board. Because this was a small board, the regulator needed to use as little space as possible. To accomplish this, the TPSM828214SILR from Texas Instruments was implemented in the design. This is a small switching regulator module with an integrated inductor. It is capable of sourcing up to 1 A of current while maintaining a fixed 3.3 V output.

4.3.3 Result

The SerDes breakout board was designed, fabricated, and assembled. It was not used for testing since the Controller board had not been received from the fabricators. It was able to connect to the Controller via a SamTec SEARAY™ mezzanine connector and breakout the SerDes lanes to connect to the MPF300T dev kits at the CSE. This enables testing in the future with different compute devices without altering the design of the Controller board.

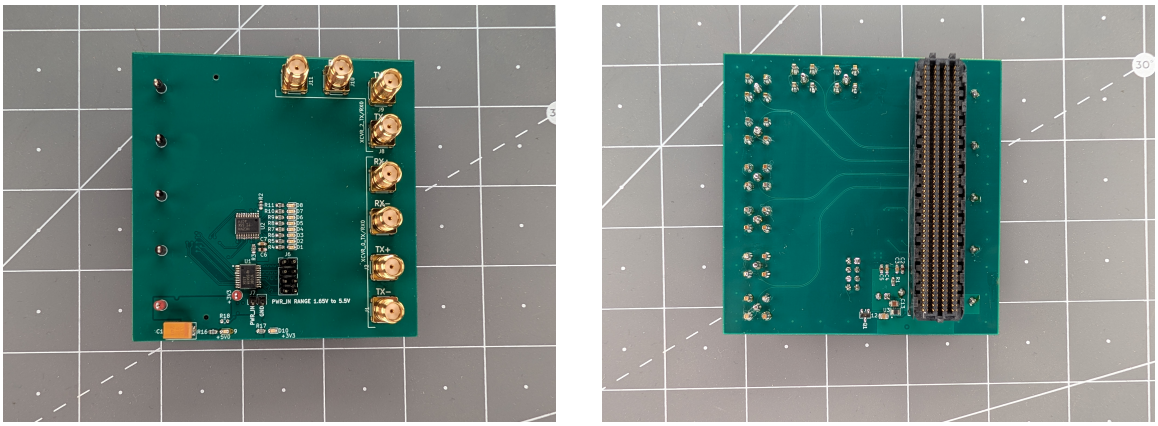

Fig. 4.15: Fully assembled SerDes Breakout Board

4.4 Summary

The work done in the implementation phase of this thesis furthered the development of the LACE-C3A project by designing a Controller board and Peripheral board to perform as a compute platform for various algorithms and applications. The design allows external sensors to send data to the system via one of several interface options, including Ethernet, USB, or other interface-independent protocols. The data can then be ingested by the Controller FPGA and routed to the appropriate Peripheral FPGA for processing. The Peripheral FPGAs have external memory provided to them for intermediate data storage and non-volatile storage devices for important immutable data, like weights and instructions. This allows the Peripheral FPGAs to compute the necessary data deliver desired data products. Finally, the Controller FPGA has the ability to reprogram the Peripheral FPGAs to allow for on the fly algorithm or application changes.

The Controller board was fabricated and, at time of writing, is being assembled. The Peripheral board completed schematic review and is currently in layout at Viridian Labs.

This work ensures the designed LACE-C3A system will perform as a valuable testing environment for firmware engineers and algorithm designers.

CHAPTER 5

Discussion and Conclusion

The hardware implementation of the LACE-C3A system demonstrates the feasibility of constructing a modular, flash-based FPGA compute array suitable for low-power edge-computing applications. The results of Chapter 4 confirm that the Controller-Peripheral architecture can be realized within the power, interface, and mechanical constraints defined in this thesis. This chapter interprets those results, evaluates the design against the system requirements, and identifies the limitations and future development paths for the LACE-C3A platform.

5.1 Overview of Design Outcomes

The Controller board was successfully designed, fabricated, and assembled, providing a stable platform for data ingress, protocol handling, and workload routing. The Peripheral board completed schematic capture and entered layout, with the electrical design validated through simulation and pin-budget analysis. Supporting hardware, including the SerDes breakout board, enabled early verification of high-speed signaling prior to the availability of the full system. Across the design process, key architectural decisions—such as the use of PolarFire™ FPGAs, a mezzanine-connector stack, and multi-rail power sequencing—proved both technically viable and scalable.

Several insights emerged during implementation. First, transceiver-based interconnects imposed practical constraints on connector selection, routing density, and board stacking, making the mezzanine connector a design-defining element. Second, the cost and availability of radiation-tolerant memory proved to be a major driver in subsystem design, requiring tradeoffs between ideal and achievable configurations. Finally, the division of responsibilities between the Controller and Peripheral boards created a natural separation between I/O management and compute resources, validating the architectural rationale developed in chapter 2.

5.2 Evaluation Against System Requirements

The design was measured against the seven system requirements defined in Section 3.1. The evaluation below summarizes how each requirement was addressed.

External data ingest (Requirements 1-4)

The Controller board provides four independent 1G Ethernet ports, an SFP+ interface, dual USB 2.0 ports, an LVDS-UART interface with PPS, and a 16-pin GPIO breakout. These collectively satisfy the requirement to support three software-defined radios, one Ethernet-enabled camera, and a flexible external spacecraft interface. The PHY, timing synchronizer, and FTDI USB bridges were validated electrically and provide sufficient bandwidth for all expected mission scenarios.

Non-volatile memory capacity (Requirement 5)

Each compute unit includes NOR Flash meeting or exceeding the required 1 Gb of persistent storage. While the initial MRAM solution proved impractical due to cost, the NOR-plus-MRAM pairing offers both reliability and performance, with a path for future upgrades.

High-speed inter-FPGA data rate (Requirement 6)

The SerDes interface was designed for 12.7-Gbps operation using four differential pairs per Peripheral FPGA, exceeding the 10-Gbps requirement. Breakout-board testing validated impedance control and link integrity over the chosen connector and routing topology.

Volatile memory capacity (Requirement 7)

The MPF500T compute unit integrates 16 Gb of LPDDR3, fulfilling the requirement for at least one high-memory compute node. The decision to place LPDDR3 exclusively on the MPF500T aligns memory capacity with expected workload demands and minimizes route congestion.

Overall, the system meets or exceeds all defined requirements, with the primary deviation being the substitution of NOR Flash for the initially targeted high-density MRAM—a change driven by cost rather than technical limitations.

5.3 Architectural Feasibility and Performance Implications

The Controller-Peripheral architecture proved effective in separating high-bandwidth I/O management from compute-intensive workloads. The MPF300T Controller FPGA provides sufficient logic resources and transceiver lanes to support future Peripheral expansions, and its stable power draw enables predictable thermal and electrical behavior.

From a scalability perspective, the mezzanine-connector stack allows additional compute boards to be added without redesigning the main I/O infrastructure. Transceiver allocation remains the practical limit on system size; each Peripheral FPGA consumed two SerDes lanes (two TxRx pairs with redundancy), suggesting that a single Controller can support multiple peripherals before exhausting its sixteen-lane budget.

The decision to use PolarFire devices continues to offer advantages in radiation tolerance and power efficiency. Flash-based configuration eliminates the need for scrubbing or SEU mitigation, reducing firmware complexity and improving long-term system reliability. The measured and estimated power consumption of the MPF300T and MPF500T aligns with mission-level constraints, making the cluster suitable for small-satellite electrical power systems.

The breakout-board testing also confirmed that the high-speed interconnect can be validated independently of full board availability, effectively de-risking the transceiver portion of the design.

5.4 Limitations Identified During Implementation

Several practical limitations emerged during implementation.

The most notable was the cost of radiation-tolerant MRAM. While MRAM offers ideal characteristics for persistent accelerator data, its price at prototype quantities far exceeded available resources. As a result, the MRAM footprint was left unpopulated for future

evaluation. This tradeoff affects the available resources to the Controller FPGA but system functionality is not reduced.

Memory routing complexity for LPDDR3 required external layout specialists. The interface demands tight length-matching, impedance control, and careful escape routing, raising the barrier for rapid design iteration. Although the design is electrically sound, it adds schedule and cost considerations for future revisions.

Connector density and pin budgeting on the mezzanine connector proved to be a central challenge. Supporting power, reprogramming interfaces, GPIO, and multiple SerDes lanes pushed the connector near its pin-count limits. While the selected SEARAY™ connector family remains feasible, any future expansion in interface count or compute density may necessitate either a second connector or a transition to a new architecture utilizing a smaller mezzanine connector for power and low-speed signals and high-speed signal capable cabling for transceiver lanes.

Moving to a cable-based architecture for the transceiver lanes solves another limitation in this design. In order to add additional Peripheral boards to a LACE stack, a hardware change must be made at the schematic and layout level to pick off the correct transceiver lanes from the mezzanine connector. Transceiver lanes can't be shared between multiple FPGAs, so the routing dictates that each FPGA has its own connection to a unique transceiver lane from the Controller. Moving to cabling-carried transceiver lanes enables them to be moved to any of the peripherals with no hardware changes.

Thermal and power headroom for the MPF500T are also areas of concern. The core rail draws several amps at high utilization, approaching the limit of a single ADP5014 channel and necessitating a parallel-channel configuration. Although simulations demonstrate stable operation, real-world thermal load must be evaluated during full bring-up.

Finally, full-system testing was constrained by staggered board availability. The breakout-board approach mitigated this issue but cannot replace full integration testing. Complete validation of multi-FPGA coordination, reprogramming flows, and workload routing requires firmware components still under development.

5.5 Risks and Reliability Considerations

Several risks remain for a flight-ready implementation. Supporting components such as USB-UART bridges, Ethernet PHYs, and NOR Flash devices have varying levels of radiation characterization. While the PolarFire FPGA fabric is inherently SEU-immune, peripheral components may require shielding, redundancy, or radiation testing before deployment.

Small satellite systems commonly use the spacecraft bus battery voltage to power payloads with high power requirements. While it may be possible to use a satellite's EPS to power the LACE system, this increases the load significantly for a typical 5 V rail. A revised power system that can accept the battery voltage (14-16 V) would give the LACE system more flexibility for future missions.

Mechanical reliability of the mezzanine connector is another consideration, especially in environments involving vibration or thermal cycling. Although the SEARAY™ connector family is well-established, future work should characterize its long-term performance in a stacked configuration.

Finally, the system's scalability introduces risks of aggregate power draw, connector heating, and increased EMI coupling. These issues must be evaluated as compute capacity grows.

5.6 Future Work

Several development paths follow naturally from this work.

The highest-priority next step is the bring up of the Controller board and fabrication of the Peripheral board. The Controller board has not been received from the fabricators, and layout services are still being done on the Peripheral board at the time of writing. Full system bring-up, including SerDes link initialization, workload routing, and in-situ reprogramming will immediately follow reception of both boards. Firmware development for high-speed packetization, buffer management, and model loading remains an open area and will determine the system's practical compute throughput

The memory subsystems on both the Controller and Peripheral board design can be upgraded to better support the needs of the various AI/ML applications that will be implemented. A new memory device to replace the expensive Micross MRAM module will need to be found for the Controller board. The Peripheral board's LPDDR3 can be upgraded to LPDDR4 to reduce power draw and increase the speed of accessing data to be processed.

Additional mechanical research is needed to optimize board stacking, thermal dissipation, and connector strain relief. Including environmental testing such as thermal-vacuum cycling, vibe testing, and radiation exposure would be essential for flight qualification.

A follow-on revision could migrate high-speed SerDes to high-bandwidth flexible cables (e.g., Samtec AcceleRate) while retaining the SEARAY for power/low-speed signals, enabling arbitrary stack sizes without schematic changes.

Finally, evaluating larger PolarFire devices or mixed-device clusters may yield higher compute density for a marginal increase in power draw, enabling more advanced model pipelines within the same architectural framework.

5.7 Conclusion

This thesis presented the design and implementation of the Low-power Array for Cubesat Edge Computing Architecture, Algorithms, and Applications (LACE-C3A), a modular FPGA-based compute platform intended to support in-situ data processing on small satellites. The work demonstrated that a Controller-Peripheral architecture built around flash-based FPGAs can satisfy the power, performance, and interface requirements necessary for running machine-learning workloads in resource-constrained environments.

The research defined a set of system requirements drawn from mission concepts and stakeholder constraints, then developed a hardware design methodology to satisfy those requirements through disciplined subsystem decomposition, simulation, and verification. The resulting implementation includes a fully assembled Controller board capable of ingesting heterogeneous sensor data, managing protocol translation, and distributing workloads across high-speed interconnects. Peripheral compute boards were designed with scalable memory and transceiver resources, and a SerDes breakout board enabled early testing of the inter-FPGA communication fabric.

Across the system, the hardware achieved or exceeded all defined requirements. The Controller board supports the necessary SDR, camera, UART, and GPIO interfaces; provides reliable power sequencing; and exposes a high-bandwidth routing fabric to downstream compute units. The Peripheral design meets the specified memory capacities for both volatile and non-volatile storage and provides sufficient transceiver performance for multi-gigabit communication. The architectural decision to concentrate I/O handling on the MPF300T and allocate compute workloads to MPF500T devices proved effective in balancing power, complexity, and scalability.

Several limitations were also identified. Cost constraints influenced memory-device selection, and the complexity of high-speed routing, particularly for LPDDR3, introduced schedule and tooling challenges. The mezzanine connector, while effective, operates near its pin and density limits, and future expansions may require alternate interconnect strategies. Full system verification awaits the completion of firmware responsible for SerDes routing,

workload packetization, and model loading.

Despite these limitations, the results of this work demonstrate that a reconfigurable, multi-FPGA compute array is technically feasible within the constraints of small-satellite missions. The LACE-C3A system provides a viable path toward real-time on-orbit data reduction and machine-learning inference, reducing reliance on ground-based processing and enabling more responsive science operations.

Future work will focus on full integration testing, firmware development, environmental qualification, and exploration of higher-density memory and compute resources. These efforts will extend the foundation established in this thesis and move LACE-C3A toward a flight-capable implementation.

In summary, this thesis establishes a practical and scalable architecture for low-power edge computing on small satellites. The implemented hardware demonstrates that modern flash-based FPGAs, when organized into a modular compute cluster, can meet the throughput and flexibility demands of on-orbit machine-learning workloads. The LACE-C3A platform provides a strong basis for continued innovation in spacecraft autonomy and real-time scientific data processing.

REFERENCES

[1] B. Denby and B. Lucia, "Orbital Edge Computing: Machine Inference in Space," *IEEE Computer Architecture Letters*, vol. 18, no. 1, pp. 59–62, Jan. 2019, conference Name: IEEE Computer Architecture Letters. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8674608/authors#authors

[2] J. Roselló, A. Martellucci, R. Acosta, J. Nessel, L. E. Bråten, and C. Riva, "26-GHz data downlink for LEO satellites," in *2012 6th European Conference on Antennas and Propagation (EUCAP)*, Mar. 2012, pp. 111–115, iSSN: 2164-3342. [Online]. Available: https://ieeexplore.ieee.org/document/6206717

[3] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi, and J. Kepner, "Survey and Benchmarking of Machine Learning Accelerators," in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, Sep. 2019, pp. 1–9, iSSN: 2643-1971. [Online]. Available: https://ieeexplore.ieee.org/document/8916327/?arnumber=8916327

[4] S. V. A.-P. E. . I. C. . Weston, "Small Spacecraft Technology State of the Art 2024 report," 2025.

[5] C. Wu, Y. Li, M. Xu, C. Guo, Z. Yin, W. Gao, and C. Chi, "A Comprehensive Survey on Orbital Edge Computing: Systems, Applications, and Algorithms," Jun. 2023, arXiv:2306.00275 [cs]. [Online]. Available: http://arxiv.org/abs/2306.00275

[6] L. D. Akers, "MICROPROCESSOR TECHNOLOGY AND SINGLE EVENT UPSET SUSCEPTIBILITY."

[7] T. Marena, "How flash-based FPGAs simplify functional safety requirements," Jun. 2018. [Online]. Available: https://www.embedded.com/how-flash-based-fpgas-simplify-functional-safety-requirements/

[8] W. F. Samayoa, M. L. Crespo, A. Cicuttin, and S. Carrato, "A Survey on FPGA-Based Heterogeneous Clusters Architectures," *IEEE Access*, vol. 11, pp. 67 679–67 706, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10158531

[9] *ADP5014 Quad-Channel Switching Regulator Datasheet*, Analog Devices Inc., accessed: 2025-08-04. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ADP5014.pdf

[10] *PolarFire FPGA Datasheet (DS00003831 F)*, Microchip Technology Inc., 2025, electrical AC/DC specifications, revision F. [Online]. Available: https://ww1.microchip.com/downloads/aemDocuments/documents/FPGA/ProductDocuments/DataSheets/PolarFireFPGADatasheetDS00003831.pdf

[11] Micross Components, *1Gb - 32M x 32 QED Spin-Torque Persistent MRAM Datasheet, Revision 2.1*, Sep. 2023, form # CSI-D-686, Document #105. [Online]. Available: https://www.micross.com/

[12] Microchip Technology Inc., "Polarfire fpga power and resource estimator," [Software], 2025, version 2021.2. [Online]. Available: https://ww1.microchip.com/downloads/secure/aemDocuments/documents/FPGA/ProductDocuments/power-estimator/3-06-00024++PolarFire+Power+Estimator.xlsm

[13] *PolarFire FPGA Board Design User Guide (UG0726 V11)*, Microchip Technology Inc., 2024, user Guide DS50003612B, version 11. [Online]. Available: https://ww1.microchip.com/downloads/aemDocuments/documents/FPGA/ProductDocuments/UserGuides/PolarFire_FPGA_Board_Design_UG0726_V11.pdf

[14] Texas Instruments Inc., *ADS1118 Ultra-Small, Low-Power, SPI-Compatible, 16-Bit Analog-to-Digital Converter With Internal Reference and Temperature Sensor (Rev. F)*, Sep. 2019, document No. SBAS457F. [Online]. Available: https://www.ti.com/lit/ds/symlink/ads1118.pdf

[15] ——, "High-speed interface layout guidelines (rev. j)," Texas Instruments, Tech. Rep. SPRAAR7J, 2020. [Online]. Available: https://www.ti.com/lit/an/spraar7j/spraar7j.pdf

[16] Cadence System Analysis, "Routing high-frequency pcb traces for signal integrity," https://resources.system-analysis.cadence.com/blog/msa2021-routing-high-frequency-pcb-traces-for-signal-integrity, Aug. 2021, blog post.

APPENDICES

APPENDIX A

Scripts

A.1 MATLAB Scripts

Listing A.1: ADP5014 MATLAB Script

```
Vout1 = 1.05;
Vout2 = 2.5;
Vout3 = 1.8;
Vout4 = 3.3;

Vref = 2;
Vin = 5;

Fsw = 1.2e6;

R2 = 10000;
Rbot = R2;

R1_1 = Vref*R2/Vout1 - R2
Rtop_2 = (Vout2/Vref - 1)*Rbot
R1_3 = Vref*R2/Vout3 - R2
Rtop_4 = (Vout4/Vref - 1)*Rbot

Vout_1 = Vref * (R2/(9.09e3+R2))
Vout_2 = Vref * (1+(2.5e3/Rbot))
Vout_3 = Vref * (R2/(1.11e3+R2))
```

```
Vout_4 = Vref * (1+(6.49e3/Rbot))

D1 = Vout1/Vin
D2 = Vout2/Vin
D3 = Vout3/Vin
D4 = Vout4/Vin

dIl1 = 1;
dIl2 = 0.5;
dIl3 = 0.5;
dIl4 = 0.5;

L1 = ((Vin-Vout1)*D1)/(dIl1*Fsw)
L2 = ((Vin-Vout2)*D2)/(dIl2*Fsw)
L3 = ((Vin-Vout3)*D3)/(dIl3*Fsw)
L4 = ((Vin-Vout4)*D4)/(dIl4*Fsw)

Kuv = 2;
Kov = Kuv;
dIstep1 = 1;
dIstep2 = 0.5;
dIstep3 = 0.5;
dIstep4 = 0.5;
dVout_uv = .01;
dVout_ov = .01;
dVout_rip = .01;
Resr = 2;
```

```
Cout1_uv = (Kuv * dIstep1^2 * L1) / (2 * (Vin-Vout1) * dVout_uv)
Cout1_ov = (Kov * dIstep1^2 * L1) / ((Vout1 + dVout_ov)^2 - Vout1^2)
Cout1_rip = (dIl1) / (8 * Fsw * dVout_rip)

Cout1 = max([Cout1_uv, Cout1_ov, Cout1_rip])

Cout2_uv = (Kuv * dIstep2^2 * L2) / (2 * (Vin-Vout2) * dVout_uv)
Cout2_ov = (Kov * dIstep2^2 * L2) / ((Vout2 + dVout_ov)^2 - Vout2^2)
Cout2_rip = (dIl2) / (8 * Fsw * dVout_rip)

Cout2 = max([Cout2_uv, Cout2_ov, Cout2_rip])

Cout3_uv = (Kuv * dIstep3^2 * L3) / (2 * (Vin-Vout3) * dVout_uv)
Cout3_ov = (Kov * dIstep3^2 * L3) / ((Vout3 + dVout_ov)^2 - Vout3^2)
Cout3_rip = (dIl3) / (8 * Fsw * dVout_rip)

Cout3 = max([Cout3_uv, Cout3_ov, Cout3_rip])

Cout4_uv = (Kuv * dIstep4^2 * L4) / (2 * (Vin-Vout4) * dVout_uv)
Cout4_ov = (Kov * dIstep4^2 * L4) / ((Vout4 + dVout_ov)^2 - Vout4^2)
Cout4_rip = (dIl4) / (8 * Fsw * dVout_rip)

Cout4 = max([Cout4_uv, Cout4_ov, Cout4_rip])
```

A.2 Additional Figures

LTspice Simluation Results

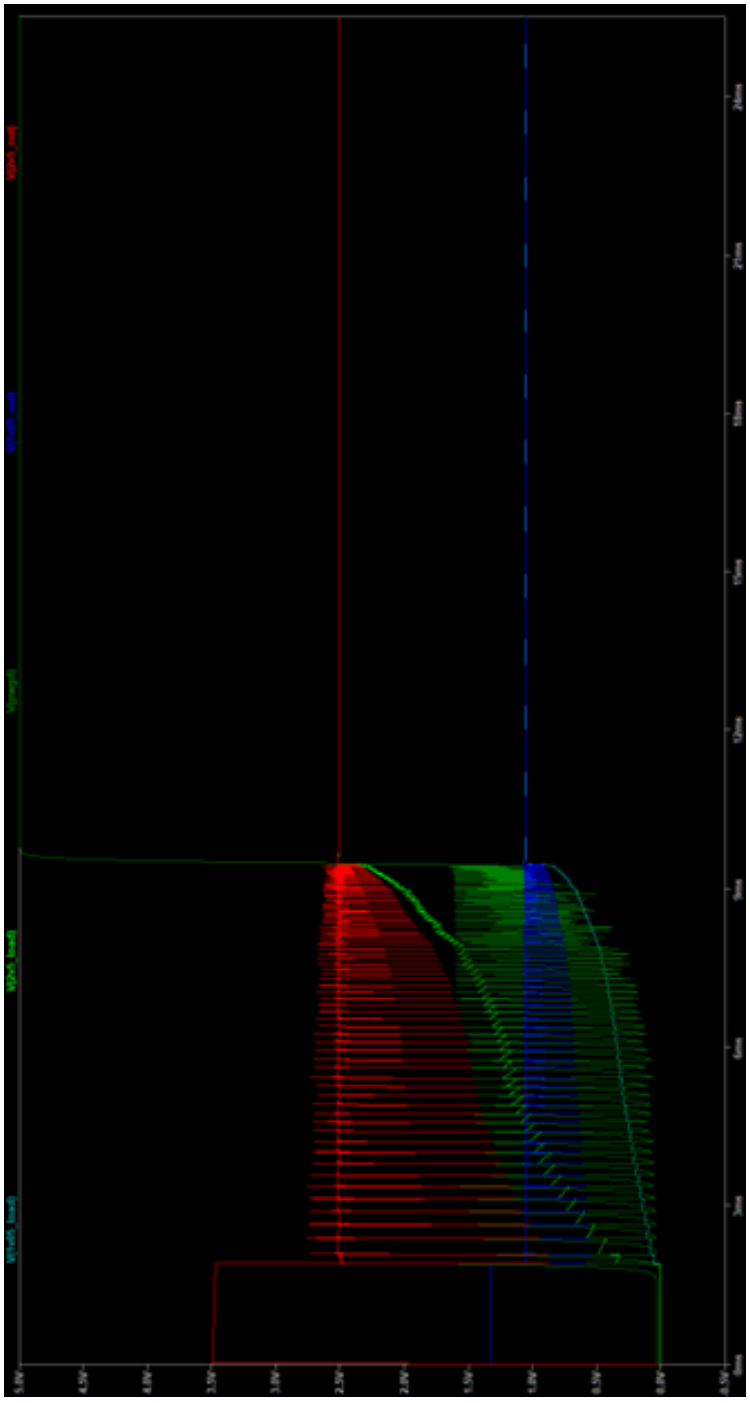

Fig. A.1: LTspice simulation results of one of the Controller board's ADP5014s