

DESIGN OF AN INTEGRATED ACCELERATION ACQUISITION SUBSYSTEM TO
SATISFY HIGH-SPEED AND LOW-AREA REQUIREMENTS FOR CUBESATS

A Thesis
presented to
the Faculty of California Polytechnic State University,
San Luis Obispo

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Electrical Engineering

by
Ryan Rumsey
June 2016

© 2016

Ryan Rumsey

ALL RIGHTS RESERVED

COMMITTEE MEMBERSHIP

TITLE: Design of an Integrated Acceleration
Acquisition Subsystem to Satisfy High-Speed
and Low-Area Requirements for CubeSats

AUTHOR: Ryan Rumsey

DATE SUBMITTED: June 2016

COMMITTEE CHAIR: Bridget Benson, Ph.D.
Assistant Professor of Electrical Engineering

COMMITTEE MEMBER: Andrew Danowitz, Ph.D.
Assistant Professor of Electrical Engineering

COMMITTEE MEMBER: Jeffrey Gerfen
Lecturer of Electrical Engineering

COMMITTEE MEMBER: John Bellardo, Ph.D.
Associate Professor of Computer Science

ABSTRACT

Design of an Integrated Acceleration Acquisition Subsystem to Satisfy High-Speed and
Low-Area Requirements for CubeSats

Ryan Rumsey

Cal Poly San Luis Obispo's PolySat team is designing the Multipurpose Orbital Spring Ejection System (MOSES) in order to record acceleration data during the launch of CubeSats as well as to provide GPS coordinates to locate the position of CubeSats once they are injected into orbit. This work focuses on the design and development of the acceleration data acquisition (DAQ) subsystem of MOSES. This subsystem is designed around the need for a high-speed sampling system of at least 200 kHz across four channels of data, plus low-area limitations in the MOSES form factor which is roughly half the size of a standard CubeSat. To address these specifications, the design explores system implementation around a Xilinx Artix-7 FPGA with a built-in analog-to-digital converter and a custom hardware solution.

ACKNOWLEDGMENTS

Thanks to my parents, I was fortunate to attend Cal Poly. Thanks to my professors, I was privileged to learn and grow. Thanks to my friends, I enjoyed browsing dank memes.

TABLE OF CONTENTS

| | Page |
|---|------|
| LIST OF TABLES | viii |
| LIST OF FIGURES | ix |
| CHAPTER 1: INTRODUCTION | 1 |
| 1.1 CubeSat | 1 |
| 1.2 PolySat | 2 |
| 1.3 CubeSat Acceleration Challenges | 2 |
| 1.4 CubeSat Acceleration Solution | 4 |
| 1.5 Operating Environment | 6 |
| 1.6 Thesis Scope | 7 |
| CHAPTER 2: DEVELOPMENT ARCHITECTURE AND ENVIRONMENT | 9 |
| 2.1 DAQ System Architecture | 9 |
| 2.2 Analog Architecture | 11 |
| 2.3 Digital Design | 11 |
| 2.3.1 Vivado Tool Flow | 12 |
| 2.3.2 Testing Physical Hardware | 13 |
| CHAPTER 3: ANALOG INPUT CIRCUITRY DESIGN | 15 |
| 3.1 Overview | 15 |
| 3.2 Stages | 16 |
| CHAPTER 4: DIGITAL CIRCUITRY DESIGN | 19 |
| 4.1 Overview | 19 |
| 4.2 FPGA Interfacing | 20 |
| 4.3 Configuration Memory | 21 |
| 4.4 Crystal Oscillator | 22 |
| 4.5 XADC Reference Voltage | 23 |
| 4.6 Launch Detection Circuit | 23 |
| CHAPTER 5: VERILOG DESIGN | 26 |
| 5.1 Xilinx Analog to Digital Converter (XADC) | 26 |
| 5.1.1 XADC Sampling Specifications | 28 |
| 5.2 FPGA Design | 29 |
| 5.2.1 ADC Reader FSM | 30 |
| 5.2.2 SPI Master Module | 32 |
| 5.2.3 SPI Loader FSM | 36 |

| | |
|--|----|
| CHAPTER 6: TESTING RESULTS | 38 |
| 6.1 Analog Circuitry Simulation | 38 |
| 6.2 Digital Design Behavioral Simulation | 40 |
| 6.2.1 SPI Master Bus Verification..... | 41 |
| 6.2.2 Loader FSM Verification..... | 42 |
| 6.2.3 Top-Level Verification..... | 43 |
| 6.3 Digital Design Development Board Verification..... | 44 |
| 6.3.1 Initial Test Design | 44 |
| 6.3.2 Final Digital Design..... | 47 |
| CHAPTER 7: RELATED WORKS..... | 54 |
| 7.1 Processor-based Data Acquisition System..... | 54 |
| 7.2 Robust Wireless Data Acquisition | 56 |
| 7.3 Slow-Speed Wearable Data Acquisition..... | 59 |
| 7.4 Comparisons..... | 60 |
| CHAPTER 8: FUTURE WORK | 61 |
| 8.1 XADC Calibration..... | 61 |
| 8.2 Event Monitoring | 61 |
| CHAPTER 9: CONCLUSIONS | 63 |
| REFERENCES | 65 |

LIST OF TABLES

| Table | Page |
|---|------|
| Table 1: Launch detection circuitry power consumption | 25 |
| Table 2: SPIMasterBus module inputs/outputs..... | 32 |
| Table 3: LoaderFSM module inputs/outputs | 36 |
| Table 4: Transaction word partitioning (16 bits) | 48 |
| Table 5: FPGA Resource Utilization | 53 |
| Table 6: Design comparisons..... | 60 |

LIST OF FIGURES

| Figure | Page |
|---|------|
| Figure 1: Poly Picosatellite Orbital Deployer (P-POD) and internal view. Source: [1] | 1 |
| Figure 2: Top-Level architecture diagram | 10 |
| Figure 3: (Left) MOSES pointing towards the top-right, bottom view. (Right) MOSES pointing bottom-left, top view | 10 |
| Figure 4: Digilent Nexys 4 Development Board | 14 |
| Figure 5: Schematic diagram of a single analog input channel | 16 |
| Figure 6: XADC bipolar mode differential input voltage characteristics | 18 |
| Figure 7: Digital component diagram. Only inputs/outputs of interest are shown. | 20 |
| Figure 8: Xilinx 7-series FPGA Master-SPI configuration interface taken from Xilinx UG470..... | 22 |
| Figure 9: Illustration of a bending piezoelectric element | 24 |
| Figure 10: The launch detection circuitry schematic | 24 |
| Figure 11: Xilinx XADC block diagram taken from Xilinx UG480 | 26 |
| Figure 12: Verilog design block diagram | 30 |
| Figure 13: Reader FSM state diagram | 31 |
| Figure 14: SPIMasterBus transmission FSM state diagram | 34 |
| Figure 15: SPIMasterBus top-level diagram..... | 34 |
| Figure 16: Frequency clock division by 5 composite waveforms | 35 |
| Figure 17: LoaderFSM state diagram | 37 |
| Figure 18: Analog channel input simulation. (Yellow) 10 KHz input with 150 KHz component. (Blue) Attenuated input. (Red) Filtered & attenuated input. | 39 |
| Figure 19: Analog channel differential conversion simulation. (Yellow) single-ended input. (Red & Blue) Positive and negative differential output, respectively. | 39 |
| Figure 20: Analog filter frequency response simulation. (Solid Yellow) Magnitude response. (Dashed Yellow) Phase response..... | 40 |
| Figure 21: Sample analog stimulus text file | 41 |
| Figure 22: SPIMasterBus module behavioral simulation | 42 |
| Figure 23: LoaderFSM module behavioral simulation..... | 43 |
| Figure 24: Top-Level Verilog design behavioral simulation..... | 44 |

| | |
|--|----|
| Figure 25: Initial physical test bench block diagram | 44 |
| Figure 26: (Green) CS_B active low probed on channel 2. (Yellow) SCLK active high probed on channel 1. | 46 |
| Figure 27: (Yellow) SCLK active high probed on channel 1. (Green) MOSI active high probed on channel 2. | 46 |
| Figure 28: (Yellow) DAC output probed on channel 1. (Green) Sine wave input probed on channel 2. | 47 |
| Figure 29: (Yellow) Quantized single-ended DAC output in 2's Complement format. (Green) Positive signal of differential input pair. | 49 |
| Figure 30: (Yellow) Quantized DAC output for 1 KHz input differential pair. (Green) Positive signal of differential input pair. | 50 |
| Figure 31: (Yellow) Quantized DAC output for 50 KHz input differential pair. (Green) Positive signal of differential input pair. | 50 |
| Figure 32: (Yellow) Quantized DAC output for 100 KHz input differential pair. (Green) Positive signal of differential input pair. | 51 |
| Figure 33: (Yellow) MOSI 16-bit word. (Green) CS_B. Notice 4 CS_B pulses closer together in the center. | 52 |
| Figure 34: Processor-based data acquisition system..... | 55 |
| Figure 35: Detailed block diagram of AD7606 ADC..... | 55 |
| Figure 36: Data acquisition system block diagram..... | 58 |
| Figure 37: Interface circuit diagram | 59 |

CHAPTER 1: INTRODUCTION

1.1 CubeSat

The CubeSat Project is an international collaboration of over 100 universities, high schools and private firms developing picosatellites containing scientific, private and government payloads [1]. A CubeSat is a 10 cm cube with a mass of up to 1.33 Kg and can be made in a variety of configurations. The standard 10 cm cube dimension is referred to as 1U. In addition, 2U and 3U CubeSats are made to accommodate larger payloads. The California Polytechnic State University, San Luis Obispo (Cal Poly), CubeSat program has the primary responsibility of developing the Poly Picosatellite Orbital Deployer (P-POD), shown in **Figure 1**. This deployer is capable of holding three standard 1U CubeSats and is secured to the launch vehicle during ascent into orbit around Earth. When the P-POD receives a deployment signal from the launch vehicle, it opens its hinged door and releases its CubeSat payload. The P-POD contains a plate and spring at its base which pushes the payload out of the deployer.

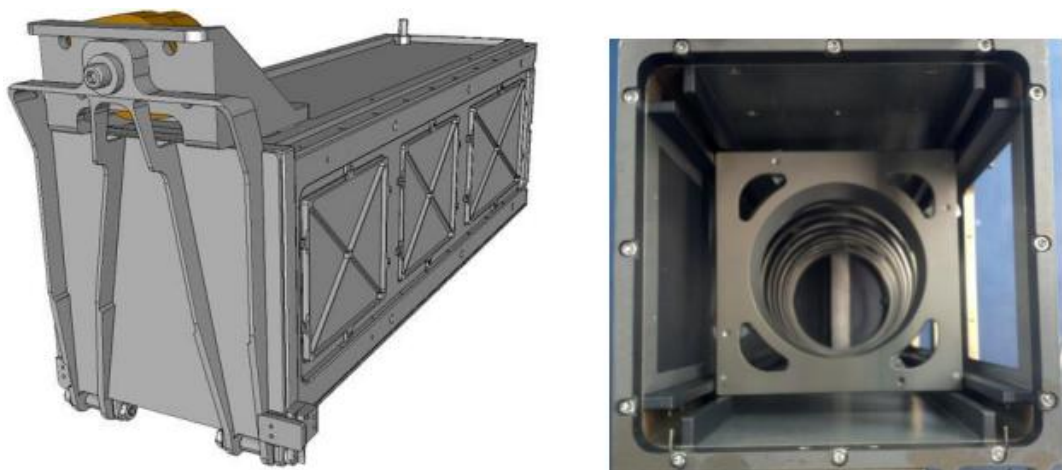


Figure 1: Poly Picosatellite Orbital Deployer (P-POD) and internal view. Source: [1]

The CubeSat program strives to provide practical, reliable, and cost-effective launch opportunities for small satellites and their payloads. To do this, CubeSat provides the community with [1]:

- A standard specification including physical layout and design guidelines.
- A standard, flight proven deployment system (P-POD), with the primary objective of protecting the launch vehicle and primary payload.
- Coordination of required documents and export licenses.
- Integration and testing facilities with formalized schedules.
- Shipment of flight hardware to the launch site and integration to launch vehicle.
- Confirmation of successful deployment and telemetry information.

1.2 PolySat

The Cal Poly PolySat Project began at the same time as CubeSat and includes a multidisciplinary team of undergraduate and graduate students working to design, construct, test, launch and operate CubeSat missions. PolySat is different than CubeSat, in that PolySat works to build CubeSats which contain a variety of scientific payloads as well as explore new technology in space. The CubeSat organization does not develop CubeSats, as its main focus is to develop and maintain the P-POD. PolySat includes teams of electrical, software and computer engineering majors from Cal Poly in the design flow of its CubeSat missions.

1.3 CubeSat Acceleration Challenges

CubeSats go through extreme acceleration and vibration during the launch of the P-POD on a launch vehicle. According to the CubeSat Design Specification, Section 4.1:

Random Vibration: Random vibration testing shall be performed as defined by the launch

provider [1]. In order to provide accurate vibrations and acceleration characteristics per launch vehicle, an active sensory acquisition system must be powered on to measure acceleration during the launch of the P-POD. By measuring the acceleration characteristics of the P-POD and CubeSats themselves during launch, PolySat and CubeSat are both better able to test future CubeSat missions and P-POD revisions for vibrations tests. This ensures the success of future missions more so than testing against random vibrations. By measuring the acceleration characteristics of the P-POD and CubeSats themselves during launch, PolySat and CubeSat developers will gain a better understanding of the stresses and vibrations their payloads will go through on its trip to orbit.

There are several challenges associated with gathering acceleration characteristics on P-PODs across multiple launch vehicles:

1. According to the CubeSat Design Specification, Section 3.3: Electrical Requirements, Subsection 3.3.1: “The CubeSat power system shall be at a power off state to prevent CubeSat from activating any powered functions while integrated in the P-POD from the time of delivery to the launch vehicle through on-orbit deployment.” This means that a standard CubeSat cannot have active sensory reading instruments during launch.
2. In order to observe spikes in acceleration up to 100 KHz, a digital sampling system would need to sample each analog input channel at no less than 200 KHz. Any slower sampling rate would violate Nyquist’s sampling theorem and result in signal aliasing.

3. An acceleration acquisition system would need to be compact enough to fit within a CubeSat's form factor, with additional area necessary for processing and communication capabilities. This is required to relay the sensory data back to Earth after it has been acquired during launch.

1.4 CubeSat Acceleration Solution

This paper focuses on a solution which addresses each of these challenges. The plate which rests on top of the spring in the P-POD will be converted into a stand-alone system with the purpose of measuring acceleration during launch, as well as providing GPS coordinates once the payload of the P-POD has been ejected into orbit. This plate will be called the Multipurpose Orbital Spring Ejection System (MOSES). To address each challenge:

1. MOSES is an integrated component of the P-POD and will not adhere to the CubeSat specifications, although the system is roughly half the size of a standard 1U CubeSat. Components of the system will be powered on during launch, which handle sensory reading, data transmission and storage. No RF functionality will be active during this process. A high-performance accelerometer will be mounted to the P-POD and interfaced to MOSES via a quick-disconnect connector. Upon payload ejection, MOSES will disconnect from the mounted accelerometer and stop gathering sensory data.
2. MOSES uses a Xilinx FPGA-based data acquisition system with a built-in Analog to Digital Converter (XADC). The XADC incorporates two 1 Msp/s ADC blocks with overlapping acquisition and conversion phases across 4 channels of analog

input. Designing a performance-optimized integrated system within the FPGA enables faster sampling rates for more channels giving faster development time.

3. With the volume of half of 1U, MOSES saves space by using the FPGA-based data acquisition system as well as a redesigned processing board. The data acquisition system utilizes an FPGA with built-in ADC, and the ability to create custom hardware drivers. This allows for future development of advanced sampling architectures.

The focus of this thesis is the subsystem which acquires and stores the acceleration sensory data during launch. This subsystem will be referred to as the Data Acquisition Subsystem (DAQ). The key requirements of the DAQ are as follows:

1. Minimize area taken up on the 4cm by 8cm side panel board.
2. Sample input sensory data between 1 Hz and 100 KHz.
3. Drive the accelerometer with 20 VDC and 1 mA constant current.
4. Filter the input analog channels with an 8th order Butterworth Low-Pass Filter for sharp cutoff at 110 KHz.
5. Convert the input analog channels from single-ended to differential paired signals.
6. Sample the input sensory data into digital words and transmit them over 3-wire SPI to a Linux Processing subsystem.

In order to store the sampled data from the accelerometer, the DAQ will transmit the 4 data channels (3 spatial axes and 1 temperature) to a Linux Processing subsystem, which will then store the data in an interfaced NAND Flash. The Atmel Microprocessor used in this subsystem includes a built-in NAND Controller interface, which handles the necessary control signals, addresses and commands needed to program and read to/from a

NAND Flash. This design choice was made in order to save time developing a NAND Controller solution for the FPGA. The choice also utilizes existing hardware on the processing subsystem and saves power by avoiding unnecessary FPGA fabric utilization. A NAND Flash was chosen because compared to NOR flash, it has a higher memory density and lower cost for high-performance applications [4]. NAND Flash devices offer faster programming and are well-suited for applications involving large sequential data sets [4]. An alternative solution to using a NAND flash for storing converted samples was to use the FPGA's Block RAM (BRAM) for data storage. However, this approach could not support the amount of time-domain data being sampled. Assuming a launch time of 10 minutes and a conversion rate of roughly 240 KHz, where each conversion yields 4 12-bit words (48 bits, 6 bytes), the data acquisition will produce almost 865 Megabytes of data. Even the largest Xilinx 7-series device, the XC7VH870T, only has 6.345 Megabytes of BRAM [8]. This option is quickly ruled out by the limiting number of BRAM blocks in 7-series devices.

1.5 Operating Environment

The operating environment of CubeSats is a harsh one, with internal component temperatures ranging from -30 to 20 C with no airflow to help cooling and hazardous radiation conditions [7]. The accelerometer provides three channels of spatial acceleration data (X Y and Z) plus one channel for measuring temperature. This fourth channel will help better characterize the temperatures CubeSats experience during launch. The farther away from Earth's surface an electronics payload gets, the more prone it becomes to stray particles of radiation colliding with a circuit path or element. These Single Event Upsets (SEU) can cause a digital circuit to latch, causing unexpected

conditions and may ultimately cause hardware or software to fail. Newer ASICs and FPGAs are even more prone to SEUs because newer chips are built using a smaller transistor process node. Current commercial FPGAs from Xilinx use transistors with channel lengths on the order of 16 nm, as seen in the Virtex and Kintex UltraSCALE+ families.

The DAQ uses a relatively new Xilinx Artix-7 FPGA with a process node of 28 nm. This scale of transistors may be prone to radiation hazards in the Low Earth Orbit environment. However, these concerns can be discarded when considering the operating time and conditions of the DAQ. The subsystem will only be active during the launch of the P-POD. During this time, the DAQ will be housed inside of the P-POD, which itself will be mounted to and housed inside of the launch vehicle's payload stage. Such shielding seems sufficient to protect the DAQ's digital logic from the radiation hazards of space. Furthermore, the recommended operating temperature for Xilinx expanded temperature devices ranges from -40 to 125 degrees Celsius, which can withstand the -30 to 20 degree range of expected temperature as mentioned previously. Once the launch vehicle gives the P-POD the command signal to unlatch and deploy its payload, the DAQ will cease data acquisition and power down, having completed its task. The relevant acceleration data will be stored in the processing subsystem's NAND Flash to be later transmitted to an Earth ground station.

1.6 Thesis Scope

The scope of this work includes the design and development of high-speed, low-area analog data acquisition hardware using signal conditioning/filtering, analog to digital conversion and digital logic/interfaces. The main focuses of this work include the design

choices, development methodologies and implementation of the DAQ. The motivation for designing the DAQ, and MOSES in general, is to enable versatile acceleration/vibration data acquisition as well as GPS coordinate tracking over a wide range of CubeSat missions. This project is able to address this challenge by integrating MOSES into the P-POD, enabling any CubeSat mission utilizing 3U or less of space to acquire the specific vibrations characteristics against the target launch vehicle and CubeSat payload. MOSES also allows the CubeSat mission to be tracked quickly from the ground by relaying the GPS coordinates of the P-POD ejection to a ground station. This method saves weeks of time compared to the conventional method of tracking CubeSats via ground-based antennas.

This work contains 9 chapters and is structured as follows:

Chapter 1 outlines the background of MOSES and the issues to address in the development of the DAQ.

Chapter 2 describes the development environment and system architecture of the DAQ.

Chapter 3 describes the analog input circuitry of the DAQ in detail.

Chapter 4 describes the digital circuitry and their interfaces in detail.

Chapter 5 describes the HDL digital design running on the FPGA in detail.

Chapter 6 describes the system tests and results after development of the DAQ.

Chapter 7 describes related works to this project and analyzes each.

Chapter 8 describes potential future work related to this project.

Chapter 9 is a summary of conclusions of the work.

CHAPTER 2: DEVELOPMENT ARCHITECTURE AND ENVIRONMENT

This chapter provides a high-level overview of the proposed DAQ system architecture.

This includes the components involved in the system design and physical hardware housing details. Then, the development environment is explored by outlining the tools used to develop both analog and digital circuitry. The digital components of this system are mostly focused on the design implemented in the FPGA, specified using a hardware description language called Verilog. Finally, because the digital components of this system are mostly focused on the design implemented in the FPGA, this chapter concludes with outlining the tool flow for generating a Verilog digital design.

2.1 DAQ System Architecture

The DAQ subsystem consists of two main components: the input conditioning analog circuitry and the sampling and acquisition digital components as shown in **Figure 2**. The analog components are responsible for low-pass filtering each of the four analog sensory inputs and converting them from single-ended to differential paired signals. The digital components are based around a Xilinx Artix-7 FPGA, which contains an internal instance of the Xilinx Analog to Digital Converter (XADC). In addition to the FPGA, there is an external clock source which provides a 100 MHz system clock, a configuration NOR Flash memory which stores the FPGA bitstream, and a microphone-based launch detector to control DAQ power-up.

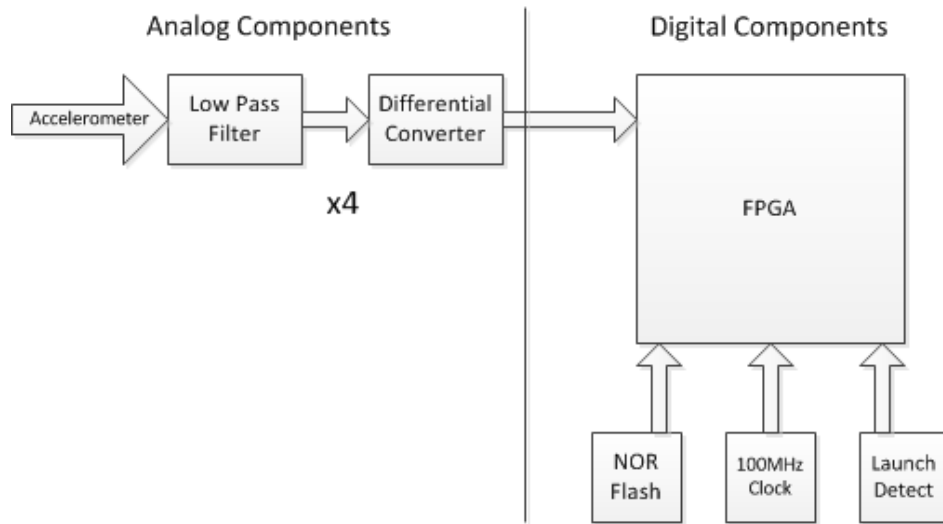


Figure 2: Top-Level architecture diagram

MOSES is shown below in **Figure 3**, viewing the underside on the left and top plate on the right. The cylinder in the center of the housing allows the P-POD ejection spring to push up against MOSES without shifting around inside the enclosure. The battery power and processing subsystems will be housed inside of the cylinder, while the DAQ and radio boards will be placed on the interior of the side panels. The DAQ will occupy a single side panel interior.

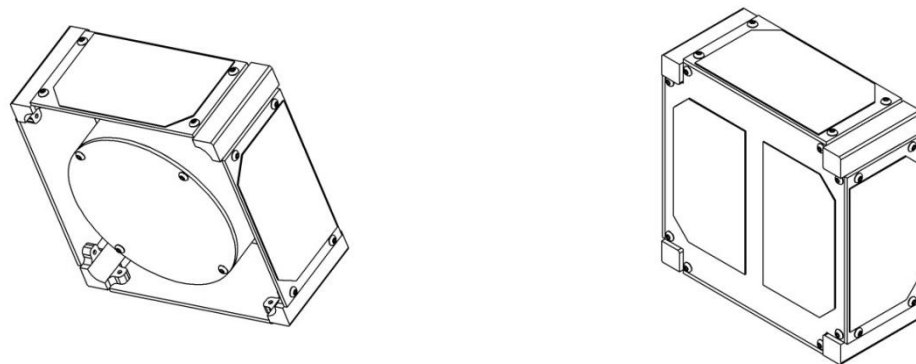


Figure 3: (Left) MOSES pointing towards the top-right, bottom view. (Right) MOSES pointing bottom-left, top view

2.2 Analog Architecture

The analog input circuitry was designed using two primary tools for the different stages.

The low-pass filter stage is implemented as an 8th order Sallen-Key Butterworth topology. The order of the filter impacts how sharp the gain drops after the cutoff frequency. The Sallen-Key topology was desirable because it allows for smaller component values while only using resistive and capacitive passive elements compared to other topologies that include inductive elements [9]. Inductors are undesirable because of their large sizes compared to resistors and capacitors. The Butterworth response was chosen because it offers a unity-gain pass band with a steep roll-off into the stop band.

In order to speed up development time, the filter was designed using Texas Instrument's free WEBENCH Filter Designer online tool [10]. Once the components were determined via the online tool, LTSpice was used to simulate the entire analog conditioner chain - bias circuit, filter and differential converter. Both transient and AC simulations supported that the analog components met the design specifications.

2.3 Digital Design

The FPGA digital design used the Xilinx Vivado Design Suite. The design is written in Verilog hardware description language (HDL). Verilog is a language designed to be simple, intuitive and effective at multiple levels of abstraction in a standard textual format for a variety of design tools, including verification simulation, timing analysis, test analysis and synthesis [2]. It is both machine and human-readable, making it a common tool for integrated circuit designers. Compared to an alternative HDL called VHDL, Verilog has syntax more in common with the C programming language and is fast to pick up for designers with prior VHDL experience.

2.3.1 Vivado Tool Flow

The Vivado design environment allows for hierarchical modeling of design sources as well as user-defined constraints targeting signal mapping to pin locations, pin I/O voltage standards and timing requirements. It also contains a powerful simulation tool capable of running a variety of simulators. This design flow used the default Vivado Simulator set with Verilog as its target language. Vivado allows for simple test bench designs under its simulation sources by instantiating the top level design source as the simulation's Design Under Test (DUT) and asserting different input stimuli. Because this design uses the 7-series XADC, Vivado can import a Text or CSV file containing analog input stimuli on multiple input channels across custom-defined time intervals.

Once behavioral simulation has been run, the next step in the Vivado design flow is to synthesize the design. After running synthesis, various reporting tools become available, such as timing, DRC and power estimations. Most importantly, timing constraints can be added at this stage of the tool flow. Timing constraints allow the Vivado tools to better analyze the design, as well as implement it onto the FPGA resources later on. By constraining the main system clock as well as and generated clocks from frequency dividers/multipliers, timing and power estimation reports become much more accurate. A design with unconstrained clocks will cause the tools to severely overestimate power and timing reports, because they are not capable of inferring the clock speed directly from the HDL.

After tweaking the design based on synthesis reports, the next stage of the tool flow is to implement the design using the Vivado Place & Route tool. This tool maps the “sea of gates” generated during synthesis on the FPGA fabric of Lookup Tables (LUTs),

flip-flops and block RAM. This process is dependent on the target device of the Vivado project. In this case, the design targets an Artix-7 XC7A15T die, the lowest-density device of the Artix-7 family. The package in this case is the CSG234, which is also the smallest package available. The small size is desirable because of the low space availability in MOSES. When the place and route tool finishes, post-implementation simulation becomes available in the simulation section of the Vivado tools.

Bitstream generation becomes available after the design has been implemented on the target device. The bitstream file is used during FPGA configuration to specify the placement and configuration of FPGA resources, including LUTs and BRAM, which implement the logic design specified by the HDL.

2.3.2 Testing Physical Hardware

In order to verify the FPGA design on physical hardware, a test version of the bitstream was ran on a Digilent Nexys-4 development platform shown in **Figure 4**. This board's target FPGA is an Artix XC7A100T device, which has about 6 times more logic cells than the XC7A15T device to be used in the final MOSES board. This development board is convenient because it includes many built-in peripherals including General Purpose Input/Output (GPIO) switches and buttons, Peripheral Module (PMOD) breakout ports and a NOR SPI Flash for storing configuration binary files. The test version of the FPGA design includes constraints for routing internal status signals to on-board LEDs and PMOD ports for easy probing and monitoring. The other advantage to the test design is that a custom 12-bit input, constrained to twelve on-board switches, could be placed to bypass the XADC and serve as a test signal input into the SPI bus driver circuitry. This makes it easy to verify that the SPI Master module is driving the correct output signals.

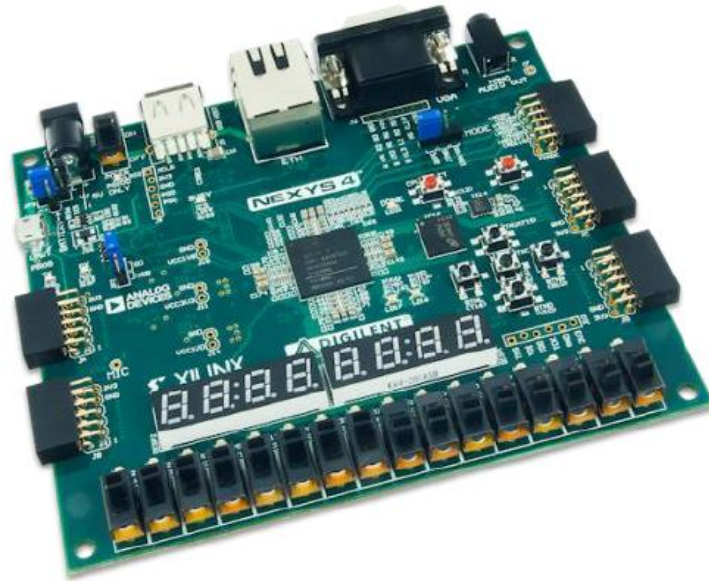


Figure 4: Digilent Nexys 4 Development Board

CHAPTER 3: ANALOG INPUT CIRCUITRY DESIGN

This chapter focuses on the design of the analog circuitry used to condition the input signals on each of the four sensory channels. Each chain of the circuit path is explained, plus some details on design choices in some stages.

3.1 Overview

The purpose of the analog circuitry component of the DAQ board is to condition and filter the four channels of analog voltage input. The reason for this conditioning is to prepare the sensory data to be input into the XADC. The accelerometer outputs a 10 Vpp waveform with 8 V DC offset, while the XADC requires differential input pairs measuring 0.5 Vpp with 0.5 V DC offset. There are three stages in each channel of analog circuitry as shown in **Figure 5**: the conditioning first stage, filter second stage, and differential converter third stage. The input to the first stage is the output of the accelerometer, while the output of the third stage is the input to the XADC.

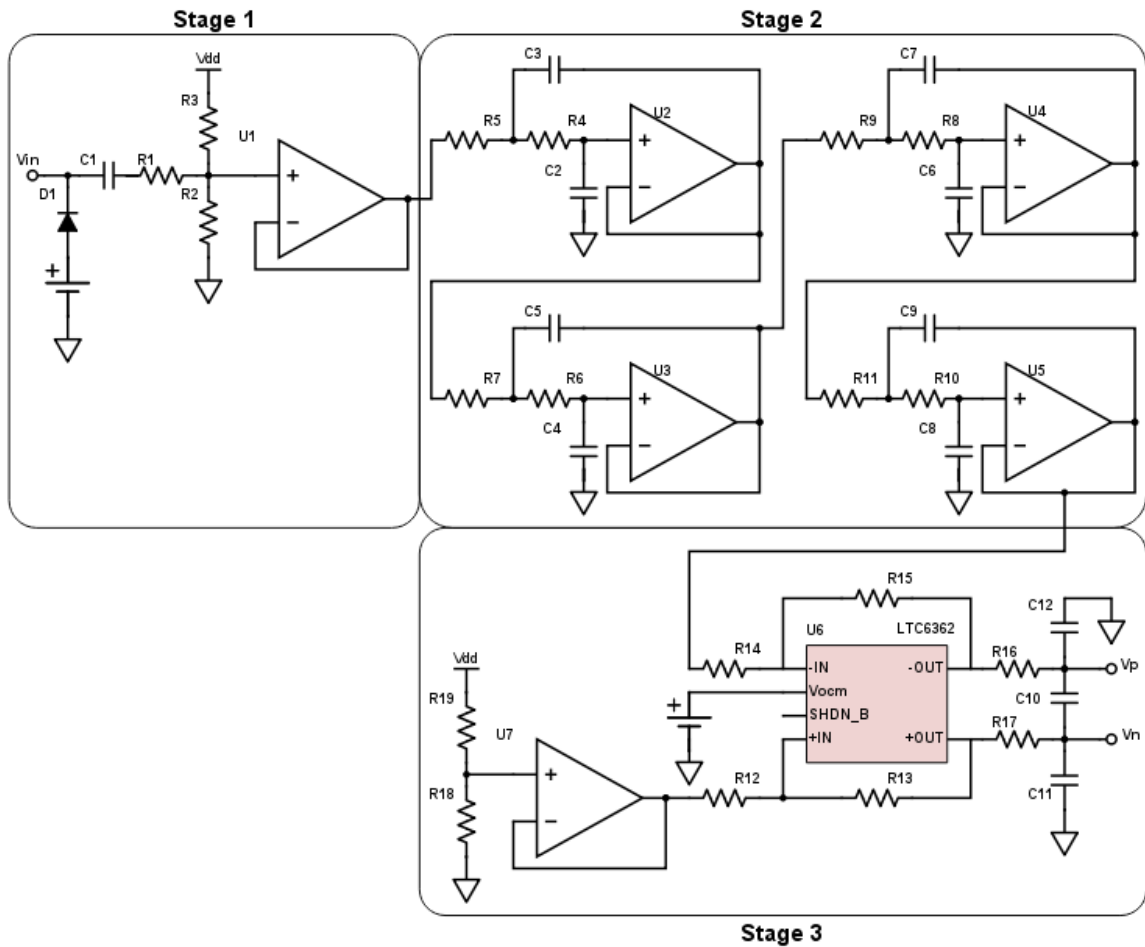


Figure 5: Schematic diagram of a single analog input channel

3.2 Stages

Stage one is connected to the input via a DC-coupled capacitor. This eliminates the 8 V DC offset, giving a 10 V DC signal centered at 0 V. A resistive divider following the capacitor conditions the signal by attenuating it by $\frac{1}{2}$ and giving it a 2.5 V DC offset. This produces a signal that has a peak-to-peak range of 5 V and is centered at 2.5 V. An op-amp configured as a voltage follower immediately after the resistive divider prevents the following stage from acting as a load to the first, thus preventing extra current from the resistive divider from being expended into the rest of the circuit.

Stage two is an 8th order Sallen-Key topology [9] butterworth low pass filter with a cutoff frequency of 110 KHz and rolloff of 160 dB per decade. This sharp rolloff is desired in order to prevent vibrations data higher than 100 KHz from entering the FPGA. The extra 10 KHz in the designed cutoff frequency allows for slight passive component variation due to temperature changes while the system is running. The Sallen-Key topology is a common and compact design which allows for the same natural frequency while using smaller-valued components compared with other topologies. Each of the four repeated structures is a second-order low-pass filter. When cascaded, the filters produce an 8-th order system with 160 dB/decade rolloff into the stop-band.

Stage three is a differential converter, which takes a single-ended signal as input and produces a pair of differentially paired signals. Differential signals are desired as inputs to the FPGA's XADC because the inputs to the ADC use a differential sampling scheme to reduce the effects of common-mode noise signals [3]. Common ground impedance couples noise voltages into other parts of the system. This noise voltage on the input can result in significant conversion errors in the binary conversion result. To avoid this, the differential analog input pairs are effectively subtracted because the Track-and-Hold amplifier of the ADC captures the difference between the positive and negative analog inputs [3]. The differential converter is an LTC6362 op-amp which takes a single-ended input and amplifies (or in this case, attenuates) it into a differential pair centered on a common mode DC offset voltage. The XADC expects a differential input signal to have a common mode range from 0.25V to 0.75V, with the difference of the positive and negative inputs to be 0.5V. These voltage characteristics are shown in detail in **Figure 6**. The differential converter receives the filtered 0V-5V waveform and attenuates then

converts it into a differential pair centered at 0.5V DC with each signal in the pair ranging from 0.25V to 0.75V, differentially.

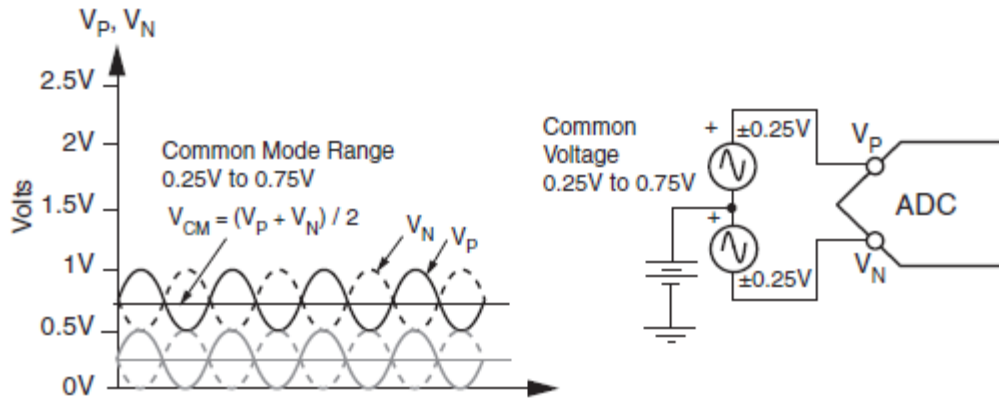


Figure 6: XADC bipolar mode differential input voltage characteristics

CHAPTER 4: DIGITAL CIRCUITRY DESIGN

This chapter focuses on the digital hardware components of the DAQ. A top-level description is given of the digital components and their functions. Then, each component is analyzed in depth in order to explain its role in the overall system.

4.1 Overview

The digital components of the DAQ board are centered on an Artix-7 FPGA with an internal ADC (XADC). The FPGA receives 4 pairs of differentially paired analog inputs, corresponding to the four channels of sensory data from the P-POD-mounted accelerometer. The other main input to the FPGA is a status signal from the launch vehicle, which enables the operation of the XADC, thus sampling acceleration data. The primary outputs of the FPGA include the three signals associated with the SPI bus: Chip Select (CS), SPI Clock (SCLK) and Master Out Slave In (MOSI). These signals compose the SPI bus which interfaces across boards to the Atmel processor. This processor includes a NAND Flash Controller which will take the received time-domain measurements from the SPI bus and store them onto a NAND flash.

There are several peripheral circuits which interface with the FPGA. A NOR flash which contains the configuration bitstream is connected to the FPGA's configuration interface. A 100 MHz crystal oscillator is routed into the FPGA as an external system clock. A 1.25V reference IC provides a reference voltage for the XADC. Finally, launch detection circuitry built around a microphone provides the startup signal for the entire DAQ system. **Figure 7** below shows the digital component interfaces.

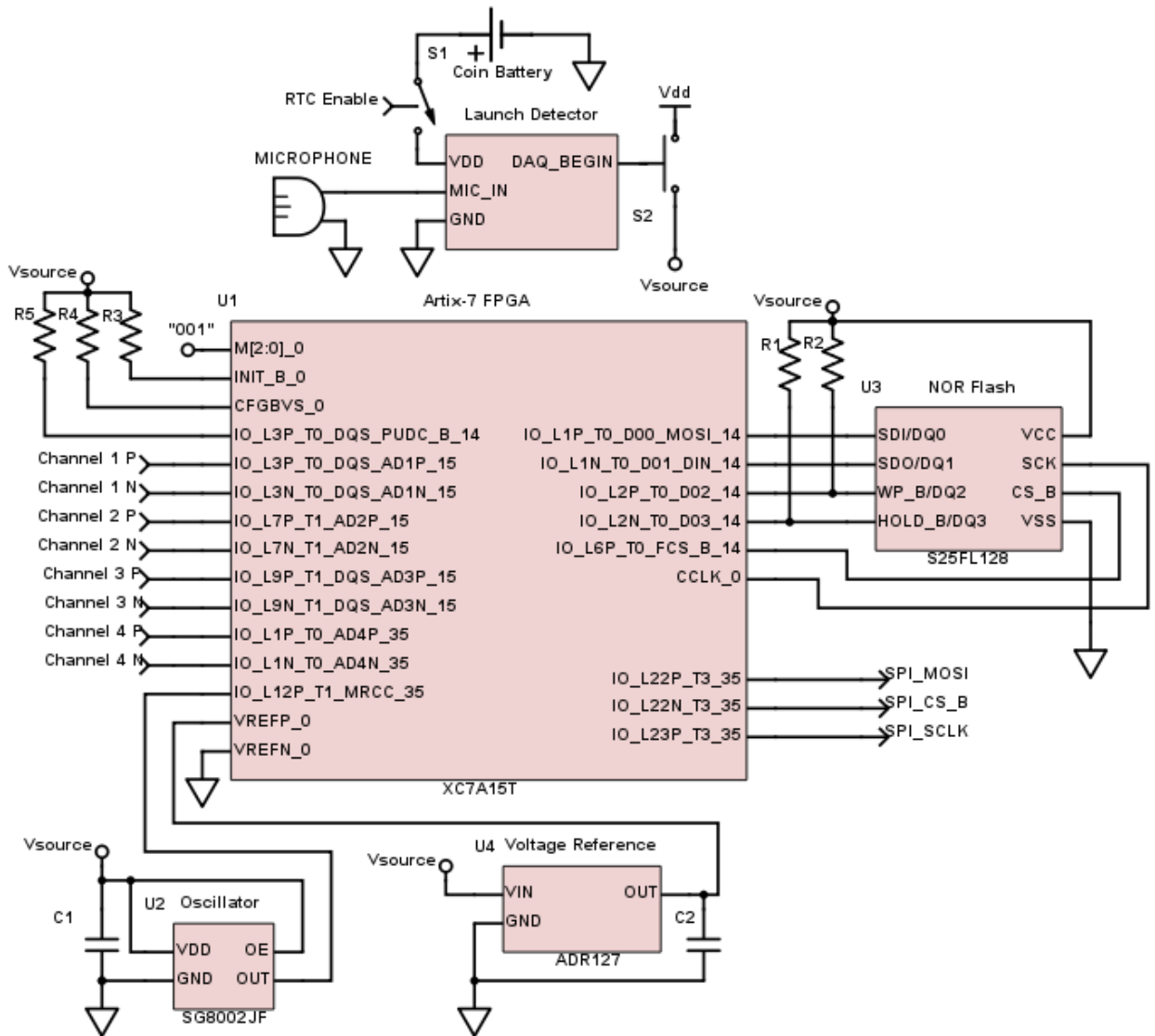


Figure 7: Digital component diagram. Only inputs/outputs of interest are shown.

4.2 FPGA Interfacing

The FPGA interfaces with the other digital components via its multi-purpose configuration pins as well as its bidirectional digital I/O ports. Xilinx FPGAs have multiple logic banks that contain different logic signals, often grouped together by functionality. Xilinx 7-Series FPGAs have dedicated configuration pins located in Bank 0 which are always part of every configuration interface. Multiple interfaces can be used, such as a Master-Slave mode, where the FPGA reads from a flash peripheral which

contains the configuration bitstream file. Depending on the specified configuration settings embedded in the bitstream file, the interface can vary from a 1-bit wide Serial Peripheral Interface (SPI) bus to a 16-bit wide Bus Peripheral Interface (BPI) bus. Bank 14 also contains some multi-purpose ports which are enabled for use during configuration based on which mode the FPGA is using. The rest of the digital I/O ports used in the Verilog design are bidirectional and are constrained to specific module inputs and outputs based on the HDL design.

4.3 Configuration Memory

The configuration memory is set up as a quad SPI (x4) NOR Flash memory. The theory of operation behind a quad SPI flash device is that four of its IO pins switch to data/address IOs during dual or quad read/write commands. During normal operation, the flash device has a Serial Data In (SDI), Serial Data Out (SDO), Write Protect (WP) and Hold signal. When a quad read command is issued to the device by the FPGA, these four inputs switch into a secondary IO mode, where the configuration bitstream is read from the flash device on this 4-bit output bus. This allows for faster configuration times while maintaining small package area.

The FPGA is set up to operate in the Master SPI x4 configuration mode by setting the Configuration Mode pins M2-0 as “001”. In this mode of operation, the FPGA generates a configuration clock, CCLK, using an internal oscillator. The FPGA then sends a read command to the interfaced quad SPI flash device to enable a x4 data path bus width for reading the configuration data. When configuration has completed, the FPGA asserts its DONE output. **Figure 8** below shows a reference interface between a 7-series FPGA and a SPI flash in the Master SPI x1 mode.

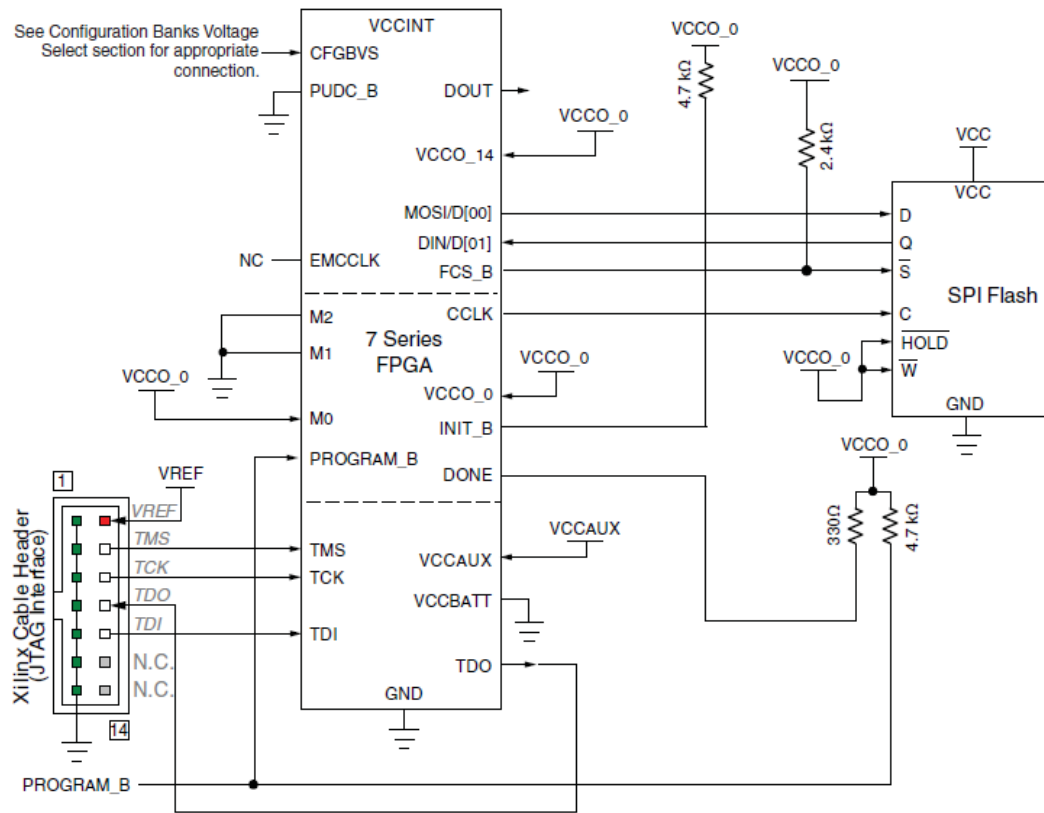


Figure 8: Xilinx 7-series FPGA Master-SPI configuration interface taken from Xilinx UG470

4.4 Crystal Oscillator

The FPGA requires a 100 MHz system clock for its internal design. This clock is generated using a pre-programmed crystal oscillator IC, SG8002JF. The IC utilizes a Phase-Locked Loop (PLL) design in order to provide a low-noise oscillating voltage on the output. The chip can operate at 5V and in temperatures ranging from -40 to +85 C. Upon ordering the IC, the manufacturer programs the desired output frequency into the PLL.

4.5 XADC Reference Voltage

The XADC inside of the Artix-7 FPGA requires an accurate reference voltage as the basis for the digital conversion. Deviation from the reference voltage can result in error against the ideal ADC transfer function. Noise on the reference voltage can also add noise to the ADC conversion and poor Signal to Noise Ratio (SNR) [3]. The XADC reference voltage inputs are V_{REFP} and V_{REFN} . These pins should be maintained at $1.25\text{V} \pm 0.2\%$. An external reference IC is used to provide 1.25V to V_{REFP} while V_{REFN} is tied to the GND pin of the reference IC. As per the recommended application guidelines for the XADC, this reference IC is placed as close as possible to the power pins in order to decrease interference on the board.

4.6 Launch Detection Circuit

In order to prevent wasteful power drain on the MOSES batteries, the DAQ must only be powered on during launch in order to conserve the most power. The launch detection circuitry on-board is crucial for these energy savings. The issue then is to design a low-power system which can detect the launch of a rocket within a time frame of several months. Once the P-POD, and thus MOSES, is packaged and ready for flight, there will be a large time window that passes until launch. The launch detection circuit must be able to detect when the launch begins without stimulus from external logic drivers.

When the P-POD is being prepared for flight, the estimated launch date will be known. This date may of course be delayed due to weather conditions. The processing subsystem on MOSES will run its Real Time Clock (RTC) powered by its own coin battery. A launch date window will be pre-programmed on the processing subsystem.

When the launch date is several days away, the processing subsystem will assert a Ready to Detect (RD) signal which will power-on the DAQ's launch detection circuitry.

The primary sensory component of the DAQ's launch detection circuitry is a piezoceramic bender element, shown in **Figure 9**. This device operates as a microphone. When a mechanical force causes the polarized 2-layer piezoelectric element to bend, one layer is compressed and the other is stretched. Charge develops across each piezo layer in an effort to counteract the imposed strain [5]. Such a piezoelectric element can be used to detect substantial pressure waves in the air, i.e. the sound from a rocket engine. This piezoelectric sensor does not require external power. When powered on by RD, the launch detection circuitry shown in **Figure 10** amplifies the signal from the piezoelectric sensor (MIC_IN) and compares it against a certain voltage threshold of $V_{dd}/2$. The 1-bit comparison value is stored in an S-R latch. This latched signal will trigger the power-up of the entire DAQ system as well as the beginning of the FPGA data acquisition.

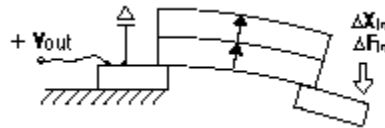


Figure 9: Illustration of a bending piezoelectric element

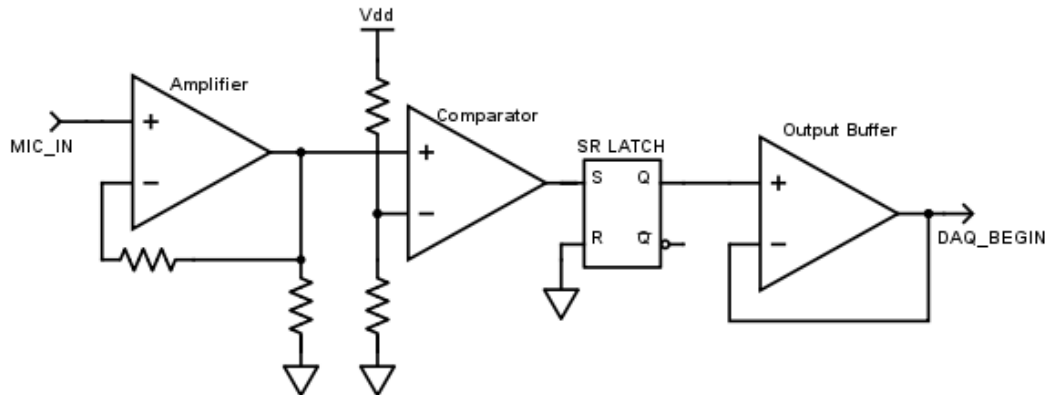


Figure 10: The launch detection circuitry schematic

The power consumption of the launch detection circuitry is crucial because it must remain operational until the rocket launch occurs. Due to possible launch delays of months past the intended launch day, the detection circuitry must remain operation for just as long. **Table 1** lists the estimated power consumption for each component, as well as a possible battery power supply. Given these estimates, the circuit can operate continuously for 538 days.

Table 1: Launch detection circuitry power consumption

| Component | Part Number | Power Consumption |
|---------------------------|--------------------|--------------------------|
| Amplifier & Output Buffer | LT6004 | 3 μ W (x2) |
| Comparator | LTC1540 | 0.9 μ W |
| SR Latch | CD4044B | 60 nW |
| Total | | 6.96 μ W |
| Battery (3V, 30 mAh) | | 90 mWh |

CHAPTER 5: VERILOG DESIGN

This chapter describes the functional modules of the digital design configured in the FPGA. The requirements to be met are first outlined in order to give context to the design choices of the hardware modules. Then, each module of the FPGA, including the built-in ADC, is described in detail.

5.1 Xilinx Analog to Digital Converter (XADC)

The XADC is a general-purpose, high-precision analog to digital converter internal to Xilinx 7-series FPGAs [3]. It includes a dual 12-bit, 1 Mega sample per second ADC and on-chip sensors. The ADCs are controlled via a set of control registers, while conversion results are stored in a set of status registers. Both of these register sets can be written/read via the Dynamic Reconfiguration Port (DRP) in the FPGA interconnect. **Figure 11** shows the block diagram for the XADC.

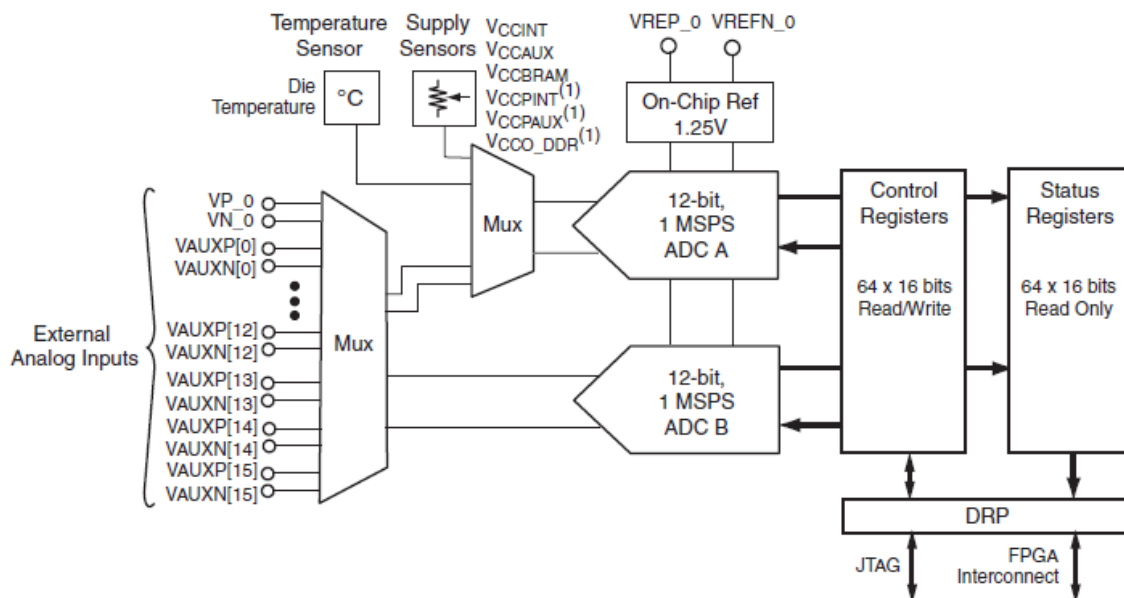


Figure 11: Xilinx XADC block diagram taken from Xilinx UG480

The XADC is instantiated into the top-level design and has its primary and auxiliary inputs connected to the top-level analog inputs. The control registers are also initialized to meet the requirements outlined above. Only auxiliary inputs 0, 1, 2 and 3 are enabled to receiving input from the four channels of the accelerometer. All built-in alarm signals are disabled. Lastly, only the four auxiliary channel conversions are enabled in the XADC sequencer auxiliary channel selection register (49h) for each conversion sequence.

The XADC has an input clock (DCLK) of 100 MHz. This source clock is divided by 4 by configuring Configuration Register #2 (42h) appropriately to achieve a 25 MHz ADCCLK, which clocks the conversion sequence. A conversion sequence for a single channel takes 26 ADCCLK cycles. This design performs four channel conversions, hence the conversion rate for all four channels is computed in equation 1.

$$f_c = \frac{25MHz}{26 \frac{cycles}{channel} * 4 channels} = 240.38 KHz \quad (\text{Equation 1})$$

There are several key outputs from the XADC that enable the high-speed transfer of converted words from its internal status register to the SPD Master bus and thus off-chip. When the XADC has finished converting a single channel in sequence, the EOC (End Of Channel) output is asserted. When the last channel in a sequence has finished, the EOS (End Of Sequence) output is asserted. EOS is used by a reader FSM to start a read sequence from the conversion result status registers. Other important XADC status

outputs include BUSY, which indicates when the XADC is busy doing a conversion, and DRDY, which indicates when the DRP is ready for another read/write.

5.1.1 XADC Sampling Specifications

In order to perform analog sampling, the system must instantiate an instance of the XADC. The core attributes of the XADC are listed below, followed by justifications for each attribute.

1. Operate in continuous sequence mode
2. Divide DCLK by 4 to achieve 25 MHz sampling clock ADCCLK
3. Operate the XADC in continuous sampling mode
4. Enable auxiliary input channels 0 through 3
5. Disable averaging
6. Enable bipolar signal inputs

In continuous sequence mode, the XADC operates for one pass through the selected sequencer channels (Auxiliary 0 through 3 in this case) and then automatically restarts as long as the mode is enabled [3]. The XADC uses an internally generated clock ADCCLK which drives the analog-to-digital conversion process. ADCCLK is derived via a frequency division of the main XADC input clock, DCLK. In continuous sampling mode, the ADC automatically starts a new conversion at the end of a current conversion cycle [3]. The auxiliary analog inputs of an Artix-7 FPGA are analog inputs that are shared with regular digital I/O pins and are automatically enabled when the XADC is instantiated in a design. When these inputs are designed to be used as analog inputs, they become unavailable for use as digital I/Os [3]. Each analog input channel may have averaging enabled. The result of a measurement on an averaged channel is generated by

using 16, 64 or 256 samples [3]. In this design, averaging is disabled on all channels in order to speed up the sampling process and avoid gathering extra samples per conversion.

In order to support differential pair analog inputs, the XADC must be configured to operate in bipolar mode. In this case, the differential inputs V_P and V_N can swing positive and negative relative to a common mode or reference voltage [3]. By using this differential signal scheme, the ADC samples both the signal and any common mode noise voltages at both analog inputs. The common mode signal is effectively subtracted, removing most of any noise present.

Furthermore, the design must read the converted 12-bit words from each of the four channel conversion result registers of the XADC as soon as each conversion sequence has completed. All four words must be output on the SPI bus serially before the next four words have been converted. Each channel is sampled at 240 KHz in order to avoid aliasing of the maximum desired input signal of 100 KHz. The desired bandwidth of the sensory data ranges from 1 Hz to 100 KHz. The Nyquist Theorem requires that the sampling frequency be at least twice that of the analog signal bandwidth. The 240 KHz sampling frequency is the minimum achievable frequency that satisfies this theorem.

5.2 FPGA Design

The target FPGA for this design is an Artix-7 XC7A15T device, CPG236 package. This is the smallest available Xilinx FPGA, as well as the least power-consuming. The XC7A15T has 16,640 logic cells and 250 I/O pins. The top level module has inputs for four channels of differentially paired analog input as well as a clock and status signals. The external clock is 100 MHz. The outputs of the top level are the 3 signals associated with the SPI bus: Chip Select, Master Out Slave In, and SPI Clock. The top level module,

shown in **Figure 12**, instantiates the XADC and includes an interface FSM which reads converted values from registers internal to the XADC. Sub-modules responsible for controlling the SPI Master bus and loading words onto the SPI bus are also instantiated in the top level design. The following sections provide more detail on each hierarchical block of the total design.

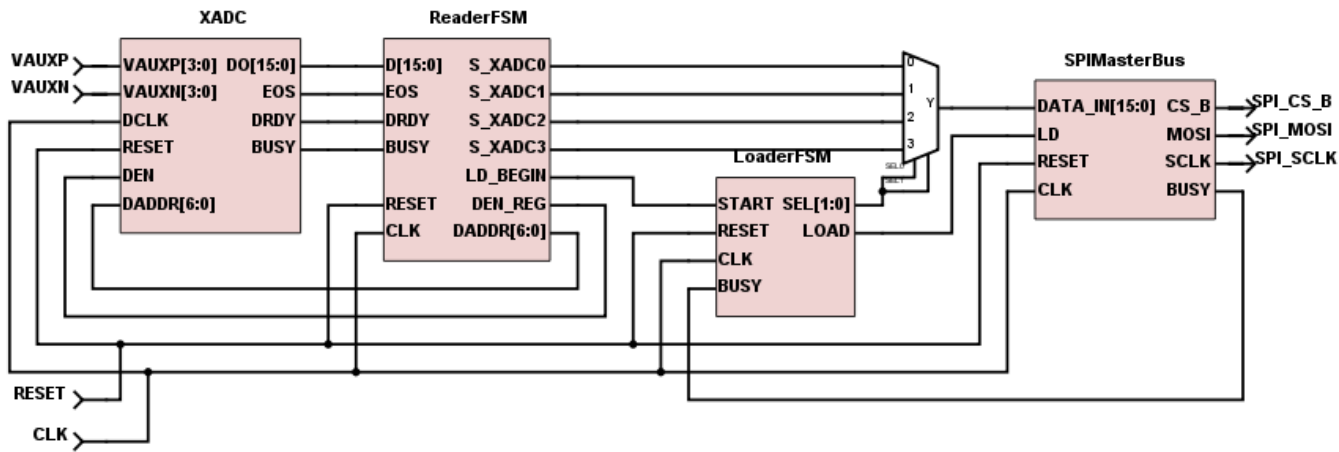


Figure 12: Verilog design block diagram

5.2.1 ADC Reader FSM

Defined in the top-level module along-side the XADC instantiation, the reader Finite State Machine (FSM) is responsible for reading the conversion results from the XADC status registers when the EOS signal is asserted. For each status register to be read from, the FSM sends that register's address on the DADDR line and waits for the DRDY signal to assert, indicating that the data is ready to be read. Once this occurs, the status register is latched onto an intermediate register to be used later by the SPI loader (s_xadc0-3). The FSM then moves onto the next status register in the sequence until all four

conversion results are read from the XADC and stored in intermediate registers. The cycle is then repeated. **Figure 13** shows the complete FSM state diagram for this module.

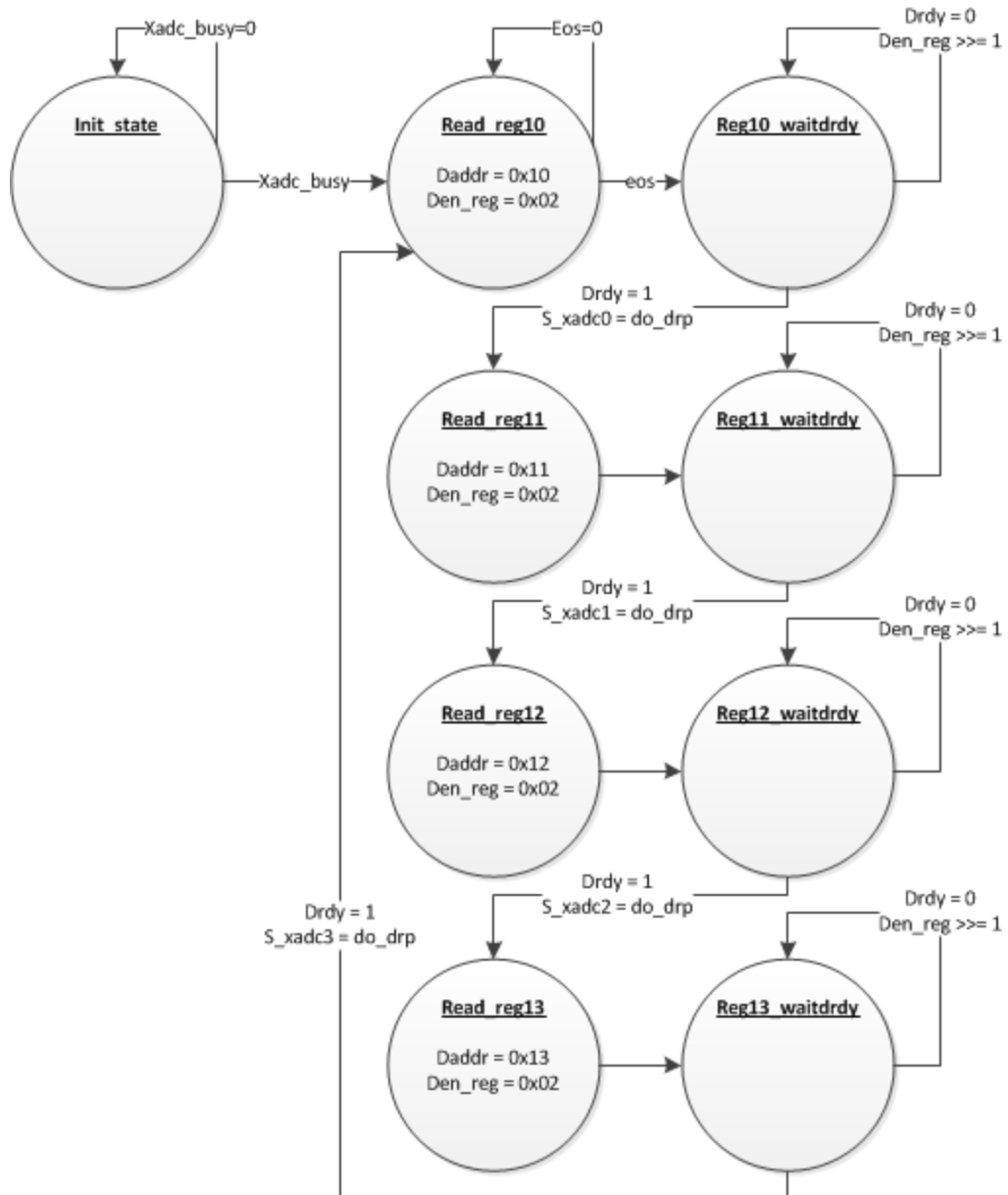


Figure 13: Reader FSM state diagram

The DEN input to the XADC initiates a read or write by capturing the values present on the DRP address (DADDR) and write enable (DWE) inputs when DEN is pulsed for one DCLK period. In the FSM, each RegX_waitdrdy state pulses DEN for one

clock period by shifting the previously stored binary value of “10” logically right while the drdy signal is low, indicating the requested data is not available on the output port DO yet. By mapping the DEN input to the LSB of this den_reg value, the input observes a single pulse lasting one clock period.

5.2.2 SPI Master Module

The SPI Master pinout is shown in **Table 2**:

Table 2: SPIMasterBus module inputs/outputs

| Signal Name | Direction | Active | Description |
|-------------------|-----------|--------|--|
| RESET | In | High | Resets the states and signals of the SPI bus. |
| LD | In | High | Load. When asserted, the data on DATA_IN is latched onto the SPI shift register. |
| DATA_IN [15:0] | In | High | 16-bit data word to be shifted out on MOSI during a SPI transaction. |
| CLK | In | High | Clock. Expect 100MHz |
| CS_B | Out | Low | Chip Select. Indicates when a SPI transaction is occurring. |
| MOSI | Out | High | Master Out Slave In. Serial data shifted out during a SPI transaction. |
| SCLK | Out | High | SPI Clock. Only active during a SPI transaction. |

| | | | |
|------|-----|------|--|
| | | | Expect 20 MHz. |
| BUSY | Out | High | Indicates when a SPI transaction is occurring. |

The SPI Master Bus module contains two sub-modules: a clock divider (by 5) and a shift register. **Figure 15** shows the module block diagram. The rest of the SPI Master Bus module is a behaviorally-modeled FSM to control the states of the SPI transaction. **Figure 14** shows the complete state diagram. These states include Idle, Wait and Transmit. Note that this SPI module only supports shifting out data and cannot receive any data input data. In order to track the number of bits serially shifted out of the MOSI line, a register holds the count of clock cycles while in the Transmit state. By default, the bus state is in Idle, where CS_B is not active. Then, when the LD signal is asserted, the shift register is loaded with the value on DATA_IN and the bus state changes to Wait. This waiting state allows the shift register to spend a single clock cycle to load the new data value. After spending one clock cycle in the Wait state, it moves to Transmit, where CS_B is made active, MOSI gets the MSB of the shift register value, and the transaction count is incremented. Once the counter reaches 16, the state returns to Idle and the module waits for another load signal.

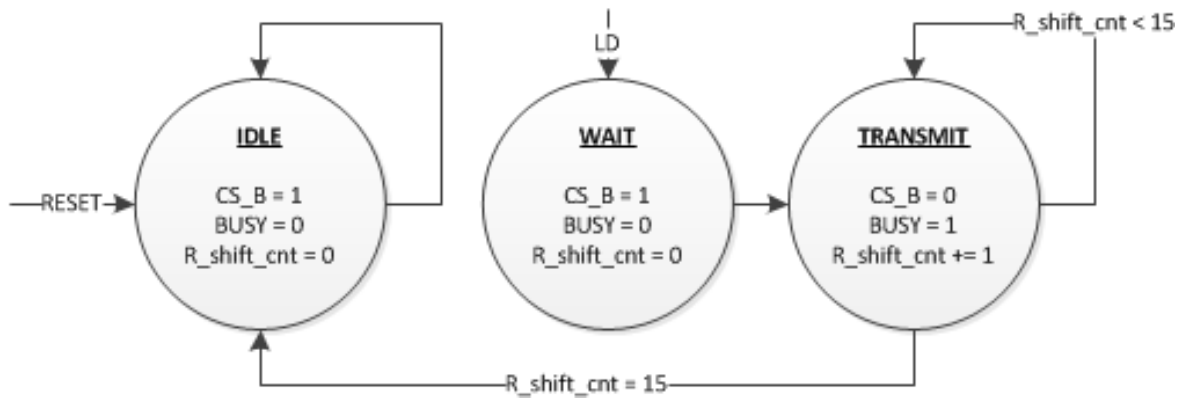


Figure 14: SPIMasterBus transmission FSM state diagram

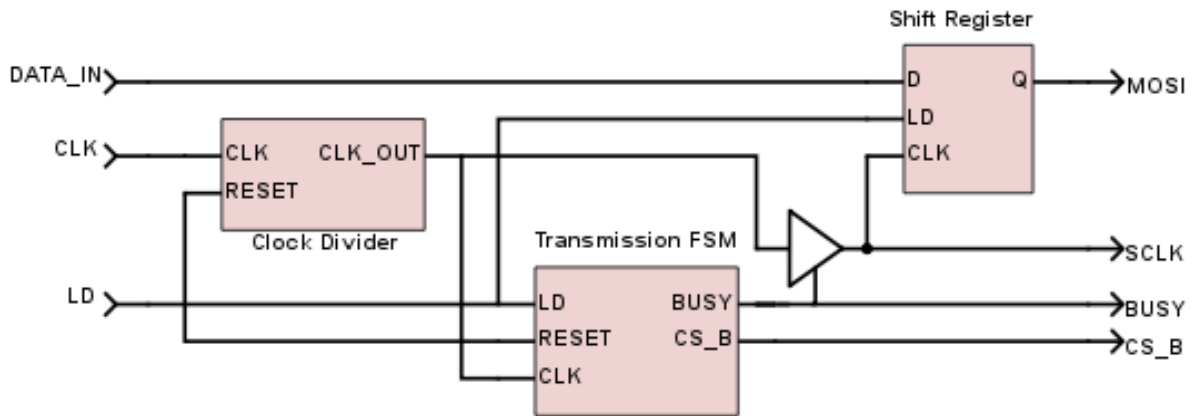


Figure 15: SPIMasterBus top-level diagram

The shift register is modeled as a shift-left asynchronous load serial output, parallel input register. It stores a 16-bit value since the SPI transmission outputs 16 bits serially. It is only active when its input clock is being driven, which itself is only enabled when the transmission FSM is driving its BUSY output. This eliminates the need for an enable input and for additional complexity.

The clock divider must divide the module input CLK by 5 to provide a 20MHz clock to be used by the rest of the module. Division-by-5 is an odd division, which cannot be achieved by simply counting the number of source clock edges for 50% of the duty cycle. Instead, the division is implemented as the addition of two counters, where one is

triggered on the rising edge of the source and the other is triggered on the falling edge. Each counter outputs high for the first two edges of the source clock, and low for the next three. By overlapping the two output waveforms together, a waveform that is active for 2.5 clock cycles of the source clock is achieved. **Figure 16** provides a visual composite reference for this process.

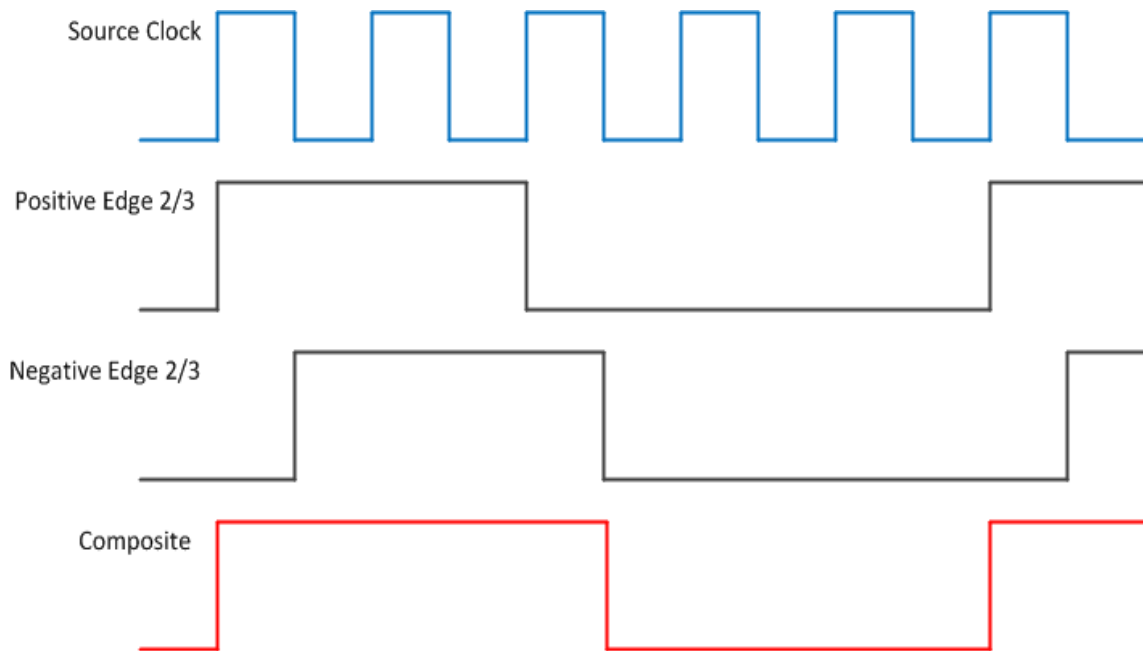


Figure 16: Frequency clock division by 5 composite waveforms

5.2.3 SPI Loader FSM

The Loader FSM pinout is shown in **Table 3**:

Table 3: LoaderFSM module inputs/outputs

| Signal Name | Direction | Active | Description |
|-------------|-----------|--------|--|
| RESET | In | High | Resets the states and signals of the FSM. |
| CLK | In | High | Clock. Expect 100MHz |
| START | In | High | Load start. Begins the process of loading in four conversion results into SPI bus. |
| BUSY | In | High | SPI Busy. Indicates when the SPI bus is busy transmitting. |
| SEL | Out | High | Conversion result select. Specifies which conversion result stored in an intermediate register is to be loaded into the SPI bus. |
| LOAD | Out | High | SPI Load. Signals the SPI Master bus to load a value into its shift register. |

The Loader FSM sequentially loads each of the four conversion results into the SPI master to be transmitted. Its full state diagram is shown in **Figure 17**. There is an external 4-to-1 multiplexer which routes each conversion result to the DATA_IN input of the SPI Master bus. The Loader FSM controls the select signal of this MUX, as well as the LD input of the SPI Master bus. The FSM's states sequentially select which

conversion result to be input into the SPI bus and then assert the LD signal. It then waits for the SPI Master module's BUSY signal to be driven low, and initiates the loading of the next conversion result. Once all four conversion results have been loaded into the SPI Master module, the FSM goes into an idle state, waiting for the next pulse of EOS from the XADC. This pulse means there are four fresh conversions to be loaded.

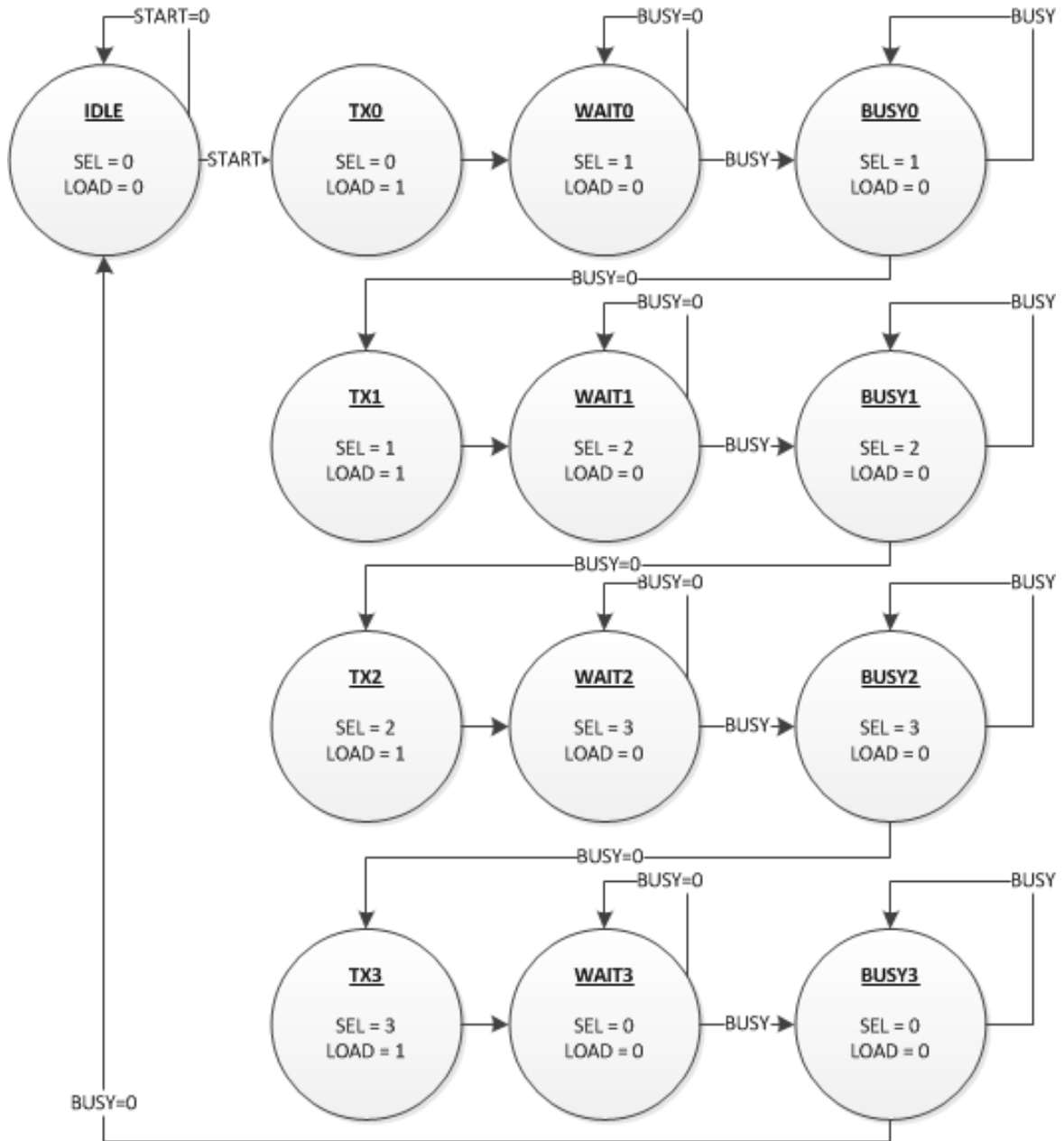


Figure 17: LoaderFSM state diagram

CHAPTER 6: TESTING RESULTS

This chapter outlines the testing that was performed on both the analog and digital components of the DAQ. The analog circuitry is tested using software simulation, while the digital design is tested both using behavioral simulation and on a physical development board. The digital design is then analyzed based on its estimated power performance and FPGA device utilization.

Throughout development of the DAQ, numerous analog and digital behavioral simulations were performed to verify design functionality and requirements. The analog components for the input channel conditioning of the DAQ were simulated in LTSpice. The digital Verilog design was simulated behaviorally in the Vivado Simulator tool, as well as tested on the Digilent Nexys4 development board.

6.1 Analog Circuitry Simulation

The analog input channels were designed and simulated in LTSpice using a voltage stimulus based on the expected output of the accelerometer, plus a high-frequency noise voltage on top of the signal. A transient simulation was performed with a 10 KHz, 10 V_{pp}, 8 V DC offset input sine wave with an added 2 V_{pp} 150 KHz noise signal. Given this input, stage 1 is expected to provide a conditioned output signal with 2.5 V amplitude and 2.5 V DC offset. Stage 2 is then expected to filter out the 150 KHz component as it is past the pass band of the low-pass filter. Simulation of this test case matches expectations, as the output of stage 2 is a clean 10 KHz sine wave without the 150 KHz noise component and is demonstrated in **Figure 18**.

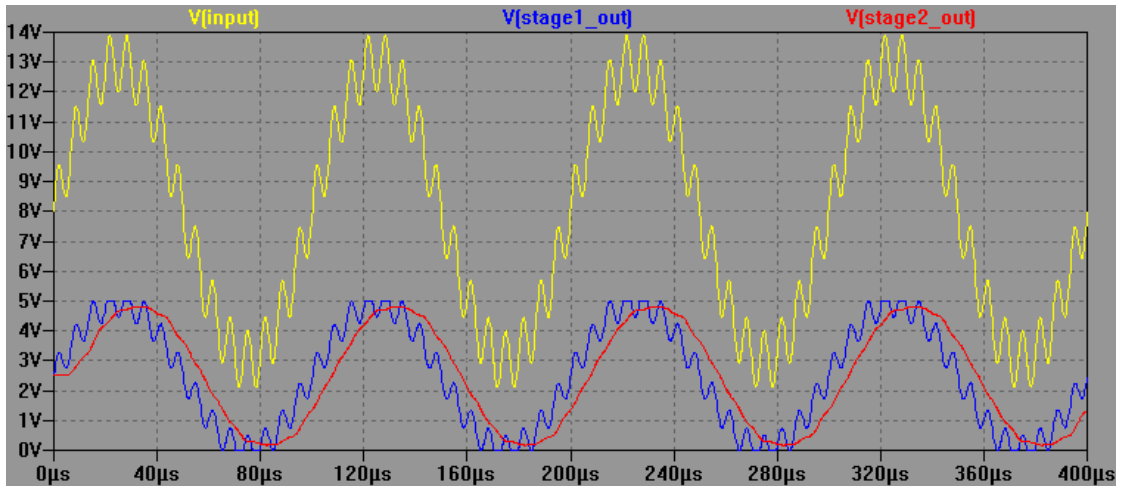


Figure 18: Analog channel input simulation. (Yellow) 10 KHz input with 150 KHz component. (Blue) Attenuated input. (Red) Filtered & attenuated input.

Stage 3 consists of the single-ended to differential signal converter and expects the 2.5 Vp 2.5 VDC offset waveform as input. The converter attenuates the input signal by 1/10 and sets the output common-mode voltage at 0.5 V. This provides a differential pair on the positive and negative output nodes which is consistent with the expected XADC bipolar input specifications. **Figure 19** shows the desired simulation result of the single-ended to differential signal conversion.

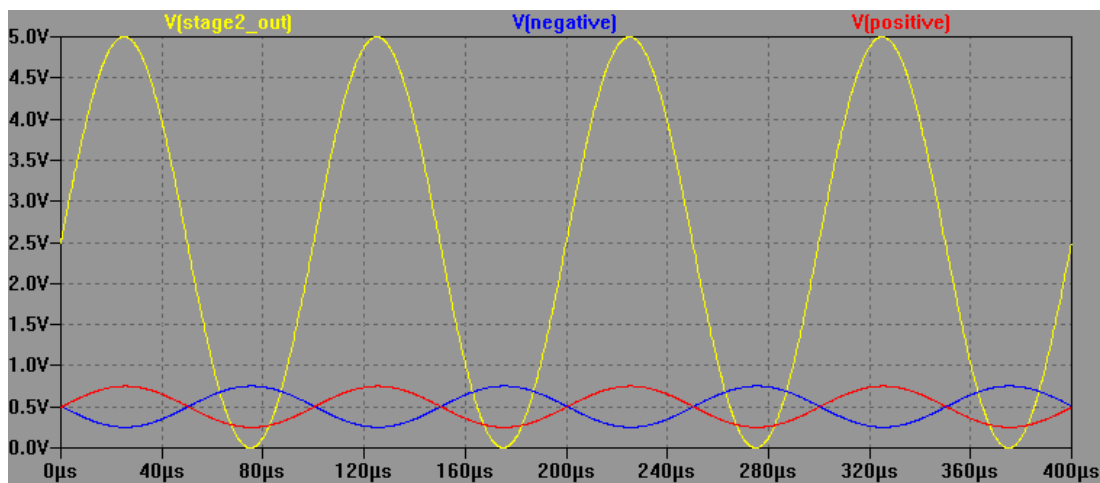


Figure 19: Analog channel differential conversion simulation. (Yellow) single-ended input. (Red & Blue) Positive and negative differential output, respectively.

The low-pass filter was verified using an AC Analysis simulation and probing the output of the filter. This simulation verified the 110 KHz cutoff frequency with 160 dB/decade rolloff into the stop band. By inserting the AC stimulus at the input of stage 1 and probing the output of stage 2, the attenuation by 1/2 is seen in the pass band gain of -6 dB in **Figure 20**.

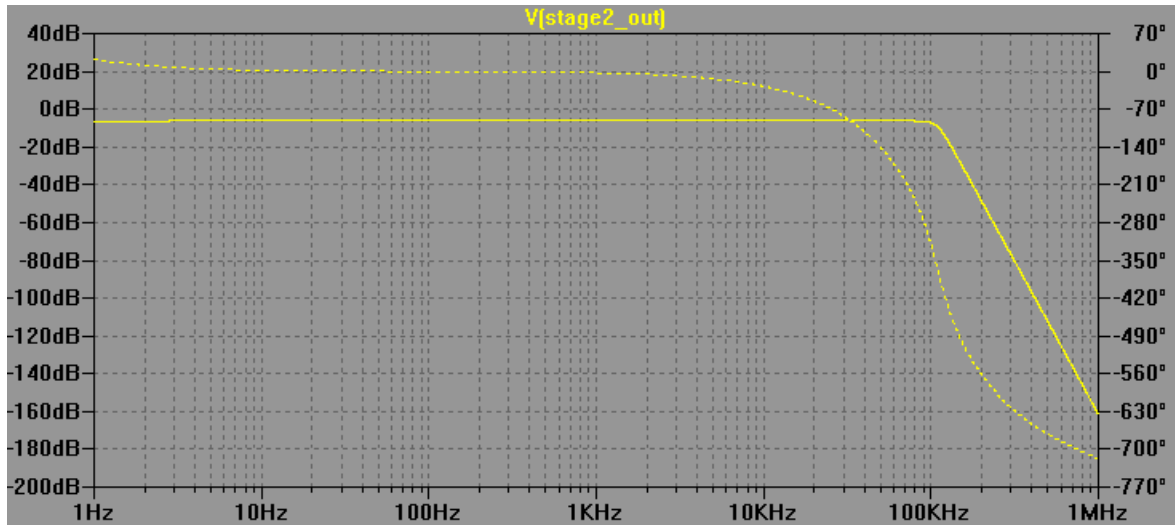


Figure 20: Analog filter frequency response simulation. (Solid Yellow) Magnitude response. (Dashed Yellow) Phase response.

6.2 Digital Design Behavioral Simulation

The Verilog HDL design was simulated using the Vivado Simulator tool and a sample accelerometer stimulus file. Each major component of the top-level module was simulated on its own test-bench and verified to work in a pre-synthesis behavioral simulation. In order to perform a meaningful behavioral simulation, the Vivado Simulator made use of a simulation stimulus file. The file name was specified as an attribute of the XADC instance of the top-level module. This file is formatted to include tab-separated columns for time step and the appropriate auxiliary inputs. The values in these columns represent analog voltage levels ranging from 0 to 1 V. During top-level simulation, the

tool will simulate the conversion of the values stored in the stimulus file. The sample file shown below in **Figure 21** uses previously measured analog acceleration data from a 3-axis accelerometer sampled at 4 KHz. Note that the Time column is measured in nanoseconds.

| TIME | VAUXP[0] | VAUXN[0] | VAUXP[1] | VAUXN[1] | VAUXP[2] | VAUXN[2] |
|--------|----------|----------|----------|----------|----------|----------|
| 0 | 0.5973 | 0.0 | 0.6325 | 0.0 | 0.6090 | 0.0 |
| 250000 | 0.6070 | 0.0 | 0.5992 | 0.0 | 0.5982 | 0.0 |
| 500000 | 0.6246 | 0.0 | 0.6080 | 0.0 | 0.6139 | 0.0 |

Figure 21: Sample analog stimulus text file

6.2.1 SPI Master Bus Verification

The SPI Master Bus operates as a standard SPI protocol 16-bit transaction master device. The slave device receiving the SPI transmission latches data on the MOSI line on rising edges of the SPI Clock (SCLK). The Master device outputs data on its MOSI line on falling edges of the SCLK in order to ensure that the data is valid during the following rising edge. The Chip Select (CS_B) line is held low at ½ period before the first SCLK edge to begin the SPI transaction. After 16 SCLK rising edges, CS_B is driven high to mark the end of the transaction. In the simulation waveform capture below, the value on the s_xadc0 register (0x98e) is to be output on the MOSI line. This value will always be a 12-bit value because the XADC has an output resolution of 12 bits. In order to support physical hardware testing by driving the MOSI line into a slave DAC, the XADC conversion value (12 bits) is appended with a DAC configuration header of 4 bits. This header has the value 0x7. This value configures the DAC to write the input into its internal register, buffer the input reference voltage, select an output gain of 1, and finally operate in active mode.

In **Figure 22**, the transaction is triggered by the eos signal, marking the end of a sequence conversion by the XADC. CS_B is driven low on the next falling edge of SCLK, and the 16-bit transaction begins. The value 0x798e is shifted serially, MSB first, on the MOSI line. At the end of the transaction, CS_B is driven high and SCLK is disabled. The SPI Master Bus module has an additional BUSY signal which is used by the Loader FSM. BUSY is simply the inverse of CS_B.

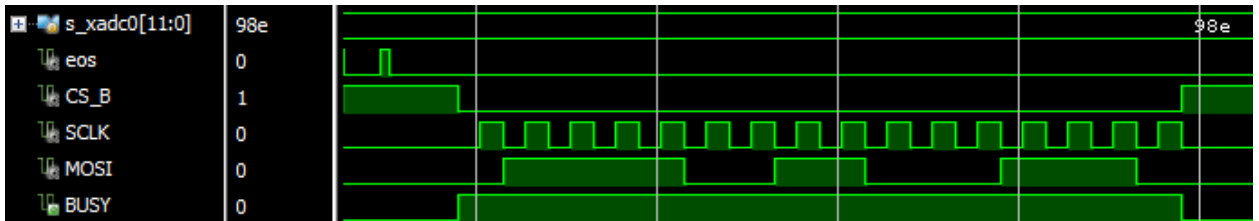


Figure 22: SPI Master Bus module behavioral simulation

6.2.2 Loader FSM Verification

The Loader FSM interfaces with the SPI Master Bus by loading each of the four XADC conversion results via a multiplexer into the SPI Master. Once the appropriate conversion result is selected through the multiplexer, the Loader FSM asserts its LOAD signal, which initiates an SPI transaction in the SPI Master Bus module shown in **Figure 23**. The START signal is driven by an intermediate signal ld_begin, which is driven high when all four of the most recent conversion results are read from the Dynamic Reconfiguration Port (DRP) of the XADC and stored in registers s_xadc0-3. Once START is asserted, the FSM increments the mux select signal SEL and waits for the SPI Master Bus output BUSY to be driven low. Once this happens, the LOAD output is driven and SEL is incremented again.

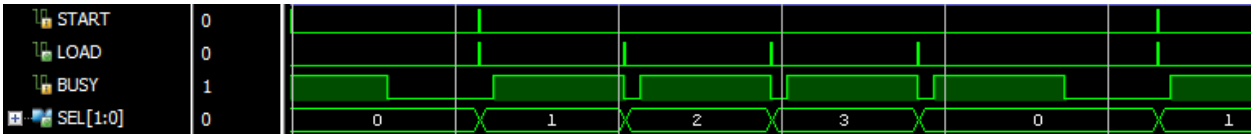


Figure 23: LoaderFSM module behavioral simulation

6.2.3 Top-Level Verification

The Top-Level DAQ design must send all four channel conversions out on the MOSI line in four sequential SPI transactions, before the next conversion results are completed. The XADC conversion clock (ADCCLK) runs at 25 MHz. With 26 ADCCLK cycles per channel conversion and four channels to convert, this gives a 4-channel conversion period of 4.16 microseconds. A 16-bit SPI transaction running on a SCLK frequency of 20 MHz takes 800 nanoseconds. Four such transactions takes 3.2 microseconds. There is an additional 100 nanosecond delay between each SPI transaction, giving a total 4-transaction time of 3.6 microseconds. Behavioral simulation in **Figure 24** verifies that the total transaction time is 3.6 microseconds and fits within the 4.16 microsecond conversion window. The simulation waveform below shows the converted values on s_xadc0 – 2 shifted out serially on MOSI, plus the appended value of 0x7 for DAC configuration. The fourth transaction is zero because there is no analog stimulus for the fourth XADC auxiliary input. This is because the analog stimulus file sample data comes from an ADC sampled on 3 channels. It also verifies that a conversion result of zero is obtained when there is no signal on an input channel.

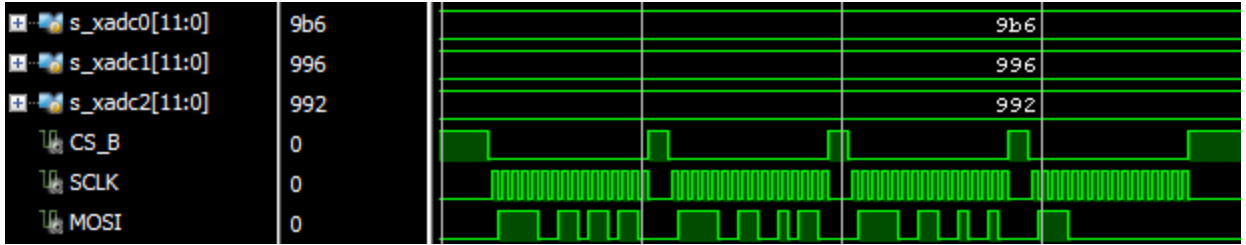


Figure 24: Top-Level Verilog design behavioral simulation

6.3 Digital Design Development Board Verification

With the components working in simulation, a testing version of the top-level design was made to verify that the physical design could sample an analog input and output the converted data via SPI to an external Digital to Analog Converter (DAC) which was expected to recreate a quantized version of the input signal. This test design simply takes a single analog input conversion and performs a single SPI transaction to an interfaced DAC, as shown in **Figure 25**.

6.3.1 Initial Test Design

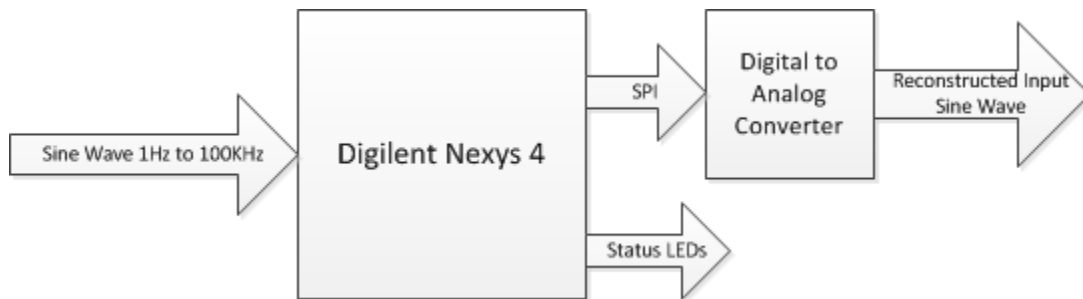


Figure 25: Initial physical test bench block diagram

The input to the test bench is a single-ended sinusoidal waveform ranging from 1 Hz to 100 KHz, which is sampled and converted by the XADC. The Reader FSM reads the conversion results from the XADC's Dynamic Reconfiguration Port and loads it into the

SPI Master bus for a transaction. This initial test bench verifies several functional requirements of the DAQ:

1. The XADC must sample analog input signals in unipolar mode and convert them into 12-bit digital words.
2. The Reader FSM must read all four conversions within a 240 KHz sample cycle.
3. The SPI Master bus must output a 16-bit serial transaction at 20 MHz.

The analog input to the test system was driven by a function generator providing a 10 KHz sine wave with 1 V_{pp} and 0.5 VDC offset. The SPI outputs were first verified by probing the CS_B, SCLK and MOSI outputs using a digital oscilloscope as shown in **Figure 26**. CS_B is driven low to indicate a 16-bit transaction, during which SCLK outputs 16 rising edges. Also during the transaction, MOSI shifts out the 16-bit value MSB first on each falling edge of SCLK so that the slave device may capture each bit value on the rising edge of SCLK. In **Figure 27** below, the binary value 0111_1110_1001_0111 is seen on the MOSI output, which corresponds to a hex value of 0x7E97. The 12-bit value (masking the 4 bit DAC header) is 0xE97. Dividing by 0xFFFF full-scale conversion value gives 0.912 V normalized.

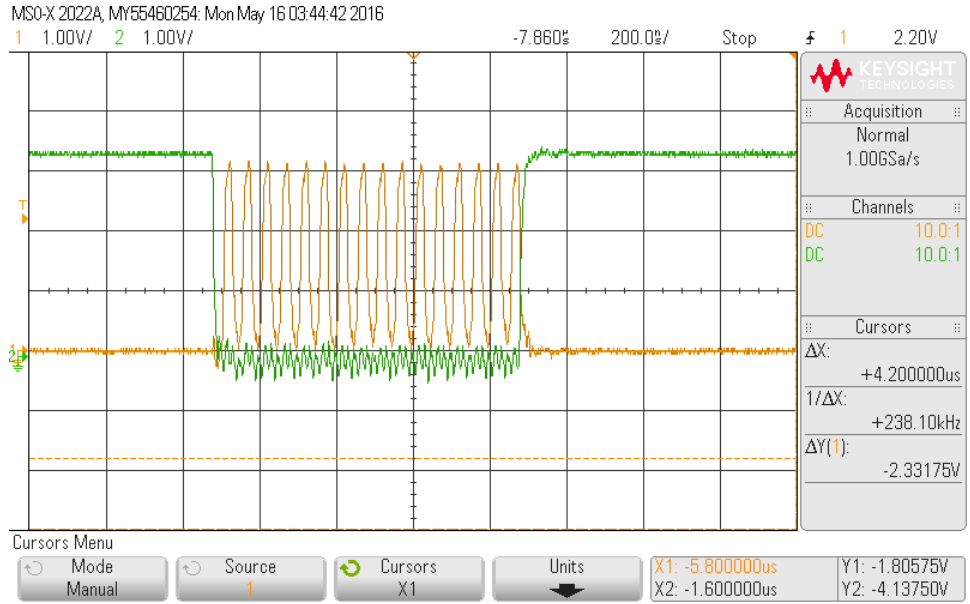


Figure 26: (Green) CS_B active low probed on channel 2. (Yellow) SCLK active high probed on channel 1.

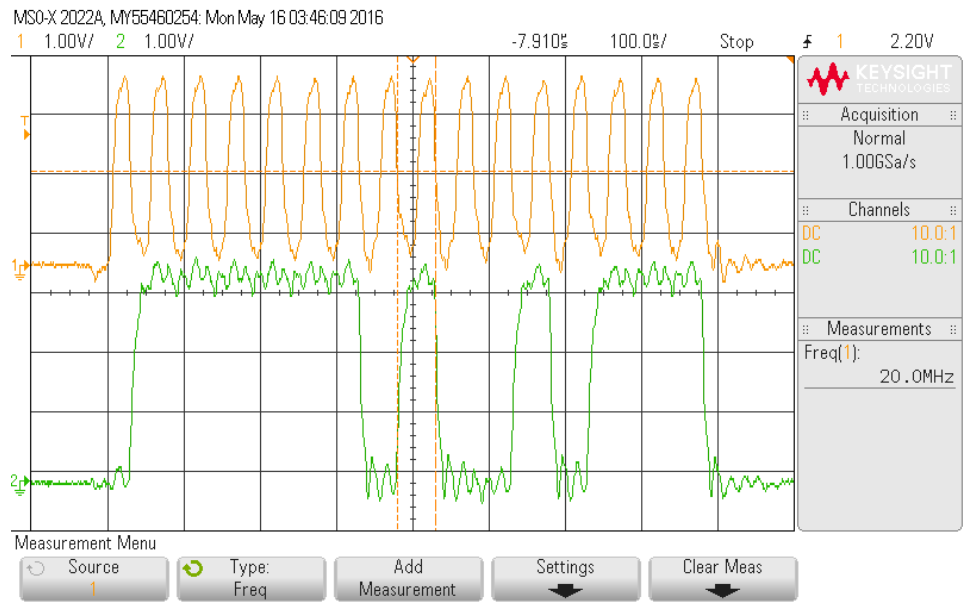


Figure 27: (Yellow) SCLK active high probed on channel 1. (Green) MOSI active high probed on channel 2.

Probing the output of the DAC shows the resulting quantized version of the input sine wave in **Figure 28**. Note that the supply voltage of the DAC is 3.3 V, so the output signal ranges from 0 to 3.3 V even though the input to the XADC ranges from 0 to 1 V. The scope capture shows that each quantized level of the output sine wave lasts for 238.1 kHz, which is due to the sampling frequency of 240 kHz. By Nyquist's sampling theorem, the theoretical maximum frequency component which can be sampled by this DAQ is less than 120 kHz, which will have at least two samples per period.

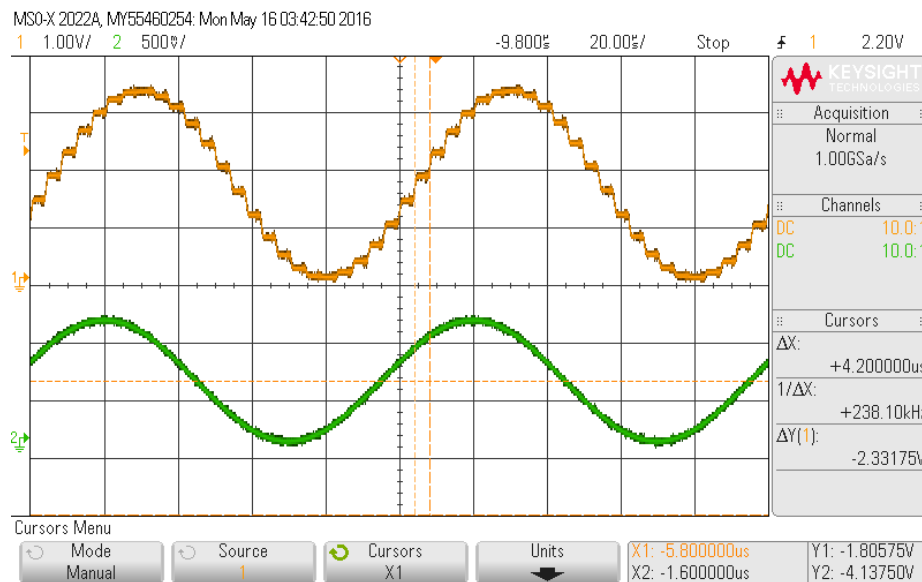


Figure 28: (Yellow) DAC output probed on channel 1. (Green) Sine wave input probed on channel 2.

6.3.2 Final Digital Design

The final digital design transmits four sequential 16-bit SPI transactions within one conversion cycle given the four analog differential inputs. Each conversion result per channel is 12 bits, which means that there is 48 bits of information to transmit. Instead of prefixing the DAC configuration header to the conversion result like before, 4 bits of zero

are added to the top-most bits of the conversion result to make a complete 16-bit transaction word. **Table 4** shows the partition of the 16-bit transaction word.

Table 4: Transaction word partitioning (16 bits)

| | |
|---------------|--------------------------|
| Zeros (15:12) | Conversion Result (11:0) |
|---------------|--------------------------|

This padding results in 4 transactions of 16-bit words at a rate of roughly 240 KHz, which means a data throughput rate of 15.38 Mbits (1.92 MB) per second. During typical rocket launch duration of 10 minutes, approximately 1.15 GB of data is produced.

The first step to verifying the final design was to ensure the XADC could operate in bipolar mode and receive differentially paired inputs. The test setup in this instance is similar to the initial test setup using the DAC with the only difference being that the input to the system is a differential sine wave pair. This setup outputs a single 16-bit SPI transmission to the interfaced DAC. **Figure 29** below shows the first attempt at reconstructing a single-ended quantized version of the inputs. This test design did not account for the conversion results from the XADC were formatted as 2's Complement, where the first bit of the 12-bit result represents the sign of the 11-bit binary number. This sign bit provides information on the polarity of the positive input relative to the negative one.

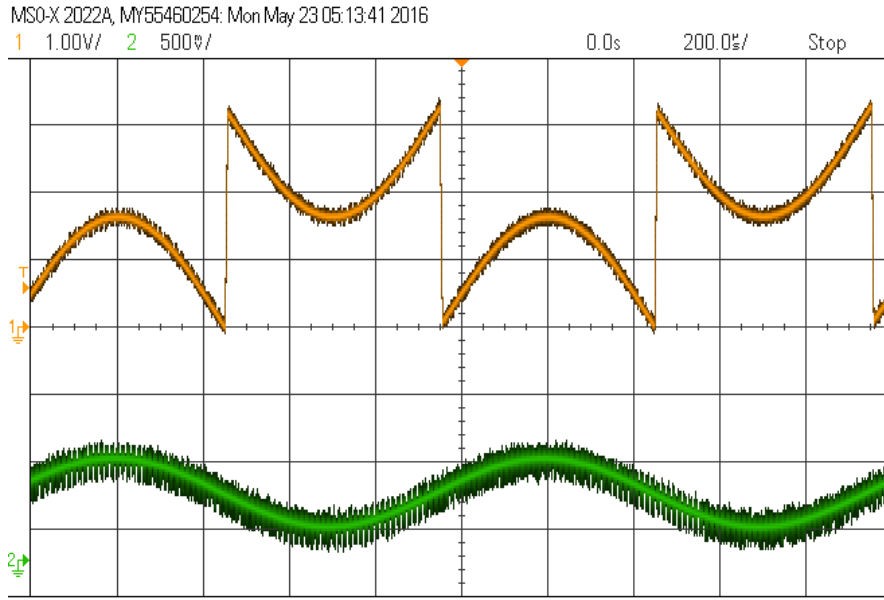


Figure 29: (Yellow) Quantized single-ended DAC output in 2's Complement format. (Green) Positive signal of differential input pair.

In order to reconstruct a continuous quantized signal, the final design inverts the 12th bit of the conversion result. This operation effectively shifts the peaks up and the troughs down, resulting in the output waveform of **Figure 30**. The following figures show differential input waveforms of varying frequencies and the resulting quantized DAC output. It's important to note that the maximum quantized output frequency of the DAC is 48.17 KHz, due to its input sample rate of 240 KHz, operating voltage of 3.3V and the device's slew rate of 0.55 V/us [11]. Once the input waveforms go past 48 KHz, the DAC output begins to show sharper transitions and exhibit increased distortions in the waveform. This is especially apparent in **Figure 32** with an input frequency of 100 KHz. Nevertheless, this testing shows that the XADC is able to sample and convert the inputs at varying frequencies.

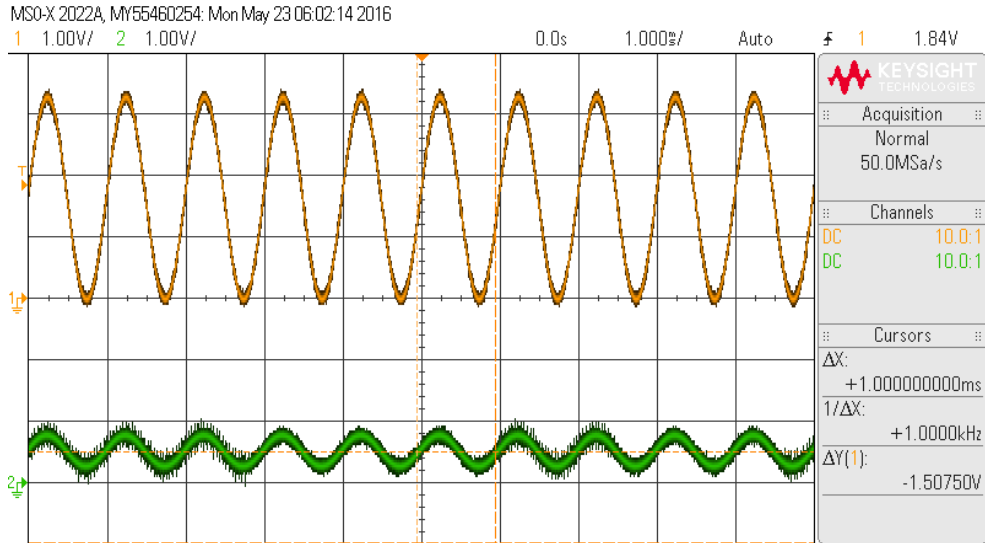


Figure 30: (Yellow) Quantized DAC output for 1 KHz input differential pair. (Green) Positive signal of differential input pair.

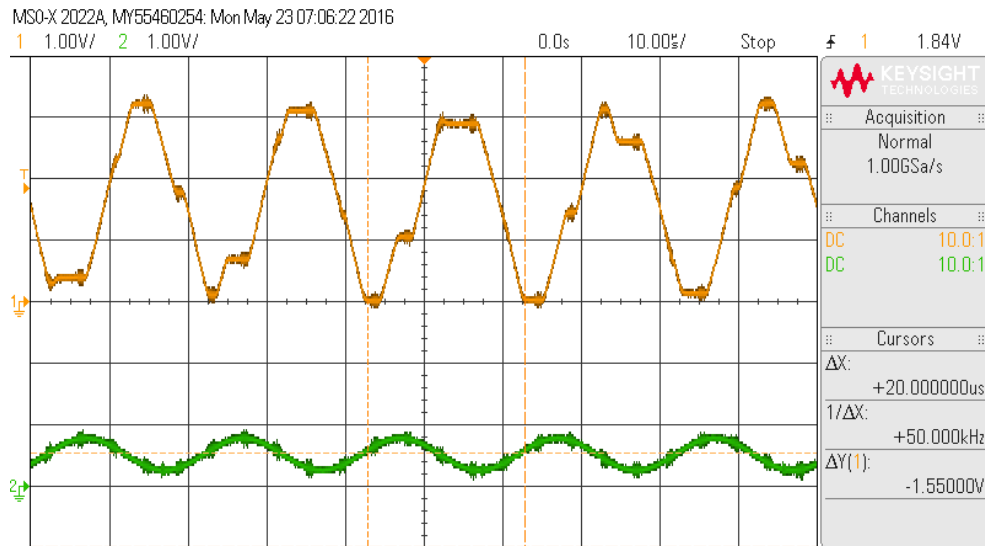


Figure 31: (Yellow) Quantized DAC output for 50 KHz input differential pair. (Green) Positive signal of differential input pair.

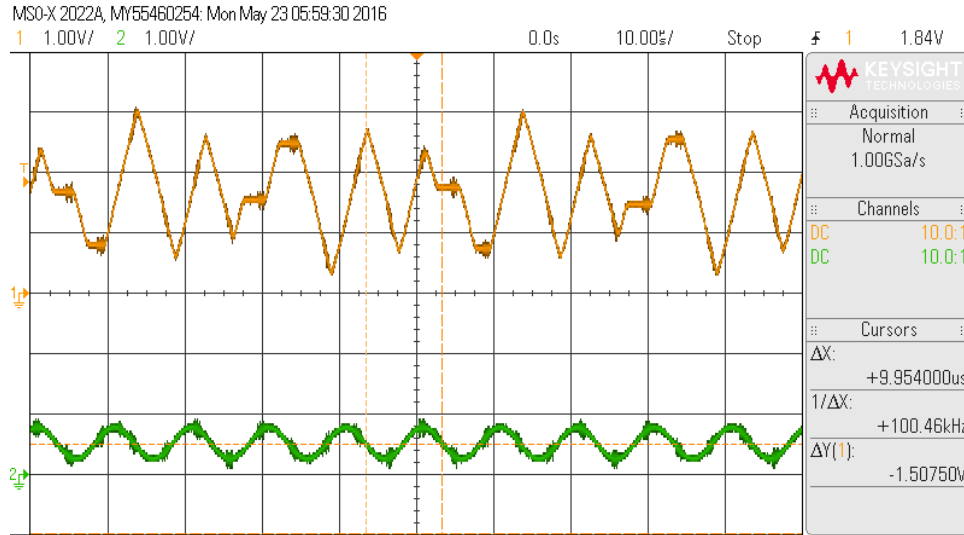


Figure 32: (Yellow) Quantized DAC output for 100 KHz input differential pair. (Green) Positive signal of differential input pair.

The second step towards verifying the final design was to test the 4 sequential 16-bit SPI transmissions. The FPGA was set up to sample from the auxiliary input channel VAUX[2] and load the conversion result into the third SPI transmission. The other three transmissions were driven with zeros to verify that the SPI Master Bus driver hardware would not introduce any glitches into the transmission. **Figure 33** below shows clusters of four transmissions, with the third containing valid conversion result data. Note that the noise present on the level regions of the scope traces is due to parasitic capacitances in the breadboard setup used to interface the DAC with the FPGA.

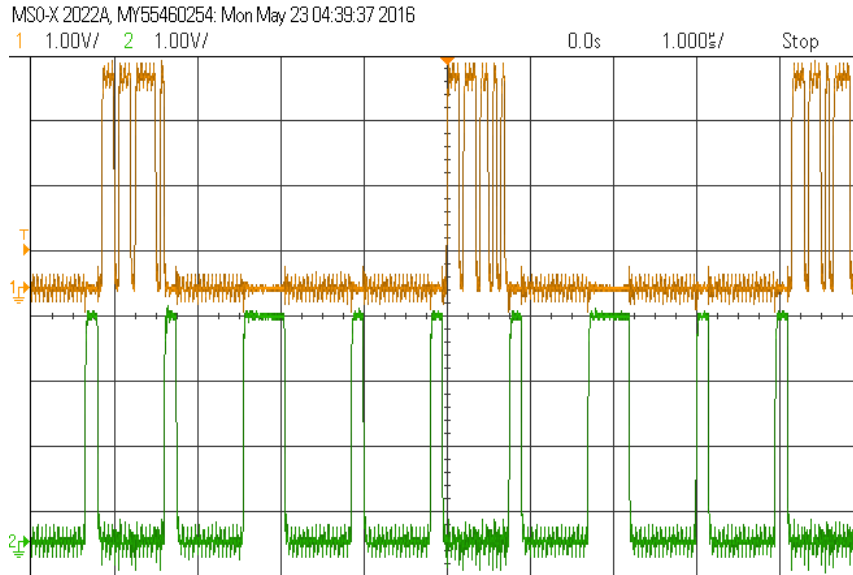


Figure 33: (Yellow) MOSI 16-bit word. (Green) CS_B. Notice 4 CS_B pulses closer together in the center.

The power estimation and device utilization metrics for the XC7A100T device are shown in **Table 5**. This design utilizes a small fraction of available FPGA hardware, which is desirable for low-power operation. This is also beneficial because the design will be running on the smallest available Artix-7 device, the XC7A15T. The power estimation value of 101 mW is derived by the Vivado tools and uses device-specific modeling of the FPGA, as well as user-specified environment parameters. These include things like airflow, load capacitance, board size and heat sink size. For context, a 1U CubeSat's power budget is typically around 20 W, using photovoltaic cells as a power source.

Table 5: FPGA Resource Utilization

| | |
|--------------------------------|-------------|
| LUT Utilization | 85 (0.13%) |
| Flip-Flop Utilization | 107 (0.08%) |
| IO Utilization | 13 (6.19%) |
| Global Buffer Utilization | 2 (6.25%) |
| Total On-Chip Power Estimation | 101 mW |

CHAPTER 7: RELATED WORKS

In this chapter, other approaches to analog data acquisition are analyzed with respect to sample throughput/speed, board area and power.

7.1 Processor-based Data Acquisition System

An alternative method to acquiring analog sensory data for use in digital systems is to use a processor and other peripheral circuitry. An example of this approach is described by Huangcheng Guo et al. in a paper from the Institute of Acoustics at the Chinese Academy of Sciences [6]. Guo describes a hardware solution for acquiring real-time vibration data using a 32-bit ARM11 CPU and a 16-bit resolution ADC, shown in **Figure 34**. The ADC used in this design is the AD7606, which supports 8 channel simultaneous sampling (**Figure 35**). It can provide up to 200,000 samples per second (200 kSPS) on all channels. In order to support the high throughput of the ADC, there is an intermediate First-In First-Out (FIFO) buffer between the ADC and the microprocessor. This processor is a Samsung mobile process that runs Linux and supports several peripheral devices such as a NAND flash and SD interface. In addition to the hardware components, there are several layers of application software and drivers programmed into the Linux system.

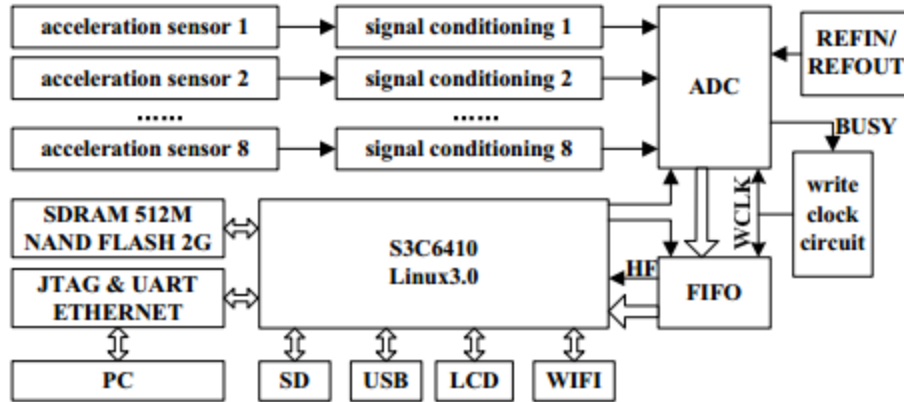


Figure 34: Processor-based data acquisition system

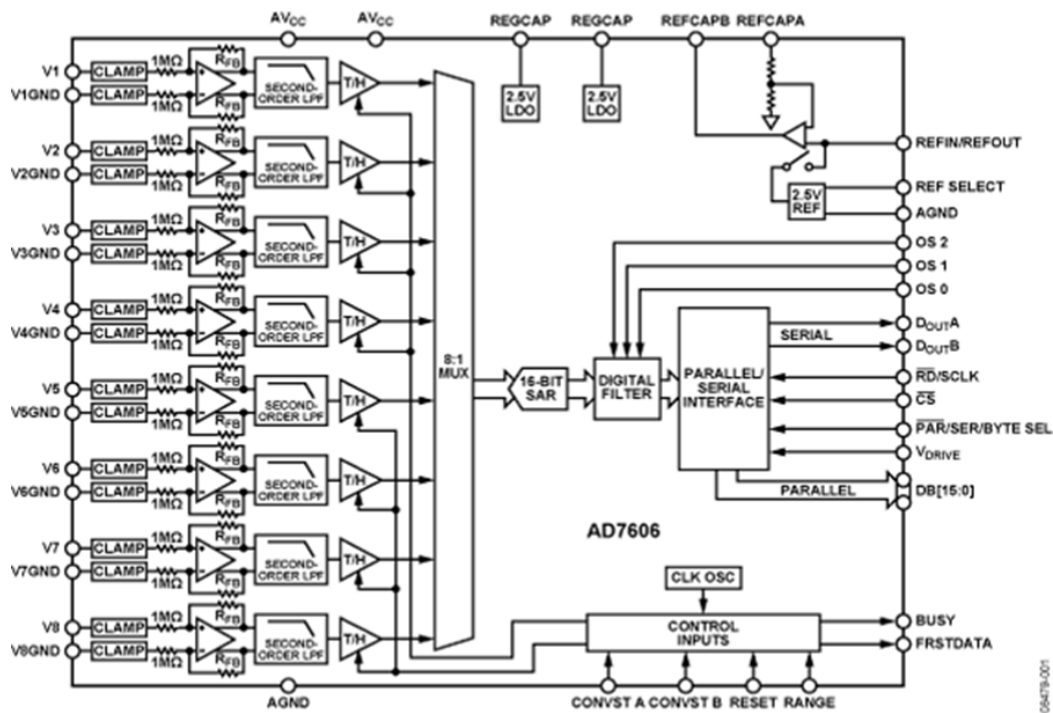


Figure 35: Detailed block diagram of AD7606 ADC

The greatest contrast between this design and the MOSES DAQ system is area usage. The two designs share the input signal conditioning components, but differ in the number of other ICs. Guo et al. utilize seven ICs including the Linux processor and the ADC. The MOSES DAQ uses four ICs, which include the FPGA with internal ADC,

NOR Flash configuration memory, crystal oscillator and launch detection circuitry. The Samsung mobile processor operates on 3.3 V of supply voltage while the separate ADC operates on a 5 V supply. The MOSES DAQ uses the smallest Xilinx Artix-7 FPGA device, which draws an estimated 101 mW. Lastly, the throughput of each system is comparable. Guo's design offers 8 channels with up to 200 kSPS each, while the MOSES DAQ uses 4 channels at 240 kSPS each. Above 4 channels, Guo's design offers better throughput per channel.

7.2 Robust Wireless Data Acquisition

Another approach to high-speed data acquisition on multiple channels is covered in a paper by C. Gaudin et al. from the Center for Offshore Foundation Systems at the University of Western Australia, Australia [12]. This work focuses on a data acquisition solution for sensory data onboard a geotechnical centrifuge and experiences accelerations up to 300 times Earth's gravity. This wireless data acquisition system (WDAS) powers and monitors eight instrument channels, each at a sampling rate up to 1 MHz simultaneously at a resolution of 16 bits. The analog sensory data undergoes amplification, conditioning and quantization all on-board. The system also stores sampled data and interface wirelessly to a control computer. **Figure 36** depicts a top-level block diagram of the control/centrifuge system. The area encased in the dashed box labeled "Channels 1-8" represents a single WDAS unit.

The core drivers behind the speed of the WDAS unit are its set of parallel microprocessors. The system utilizes four C8051F060 Flash Micro-Controller Units (MCU), each containing two 16-bit Successive Approximation (SAR) ADCs. This parallelized design is what allows the simultaneous 8-channel sampling at 1 MHz per

channel. Despite this massively parallel design, the WDAS unit is packed into a surprisingly compressed package. The entire unit is integrated into a metal box measuring 150 x 60 x 40 mm, which is equivalent to 36% of the volume of a 1U CubeSat. If high-speed analog sampling was the desired payload of a CubeSat mission, then a downscaled version of the WDAS architecture could feasibly be integrated into a CubeSat. However, in the context of the MOSES project, the volume left to work with is roughly half that of a 1U CubeSat, which also must accommodate hardware for power, processing and radio subsystems. Finally, the WDAS unit is mentioned to operate from a 12 V power supply, with no further mention of current usage or power specs. With the amount of hardware present in the WDAS unit, one can assume that power usage is in the range of tens of Watts.

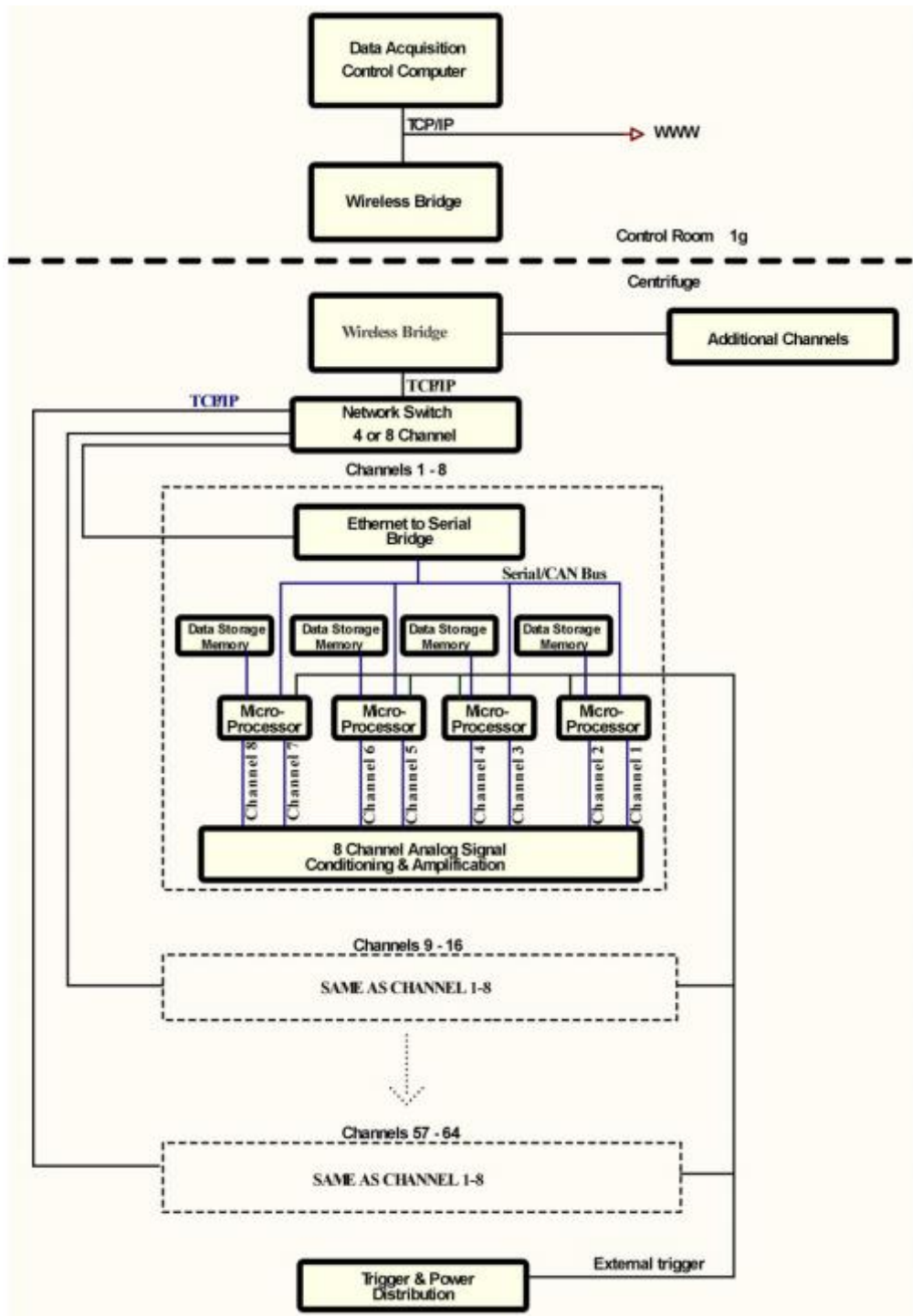


Figure 36: Data acquisition system block diagram

7.3 Slow-Speed Wearable Data Acquisition

A more simple approach to an acceleration acquisition system is outlined in the paper “A Wearable Acceleration Sensor System for Gait Recognition” by Lui Rong et al. from the Digital Engineering and Simulation Center at Huazhong University of Science and Technology, Hubei, China. This work focuses on tracking the acceleration characteristics of walking humans in order to analyze gait. Rong et al. propose a simple microprocessor-based data acquisition device to be worn on the wrist of test subjects. The device measures 3 channels of special acceleration data and samples it using an ADUC841 microcontroller with an internal 420 kSPS 12-bit ADC. The device however only samples each channel at 250 Hz. **Figure 37** shows the interface between the device’s primary components: tri-axial accelerometer and the microcontroller. The device implements simple filters to reduce noise on the analog channels. Once the samples have been converted, they are stored on a near-by 32 MB RAM. Once the data has been collected from a test subject’s gait, the data is transferred to a PC for analysis.

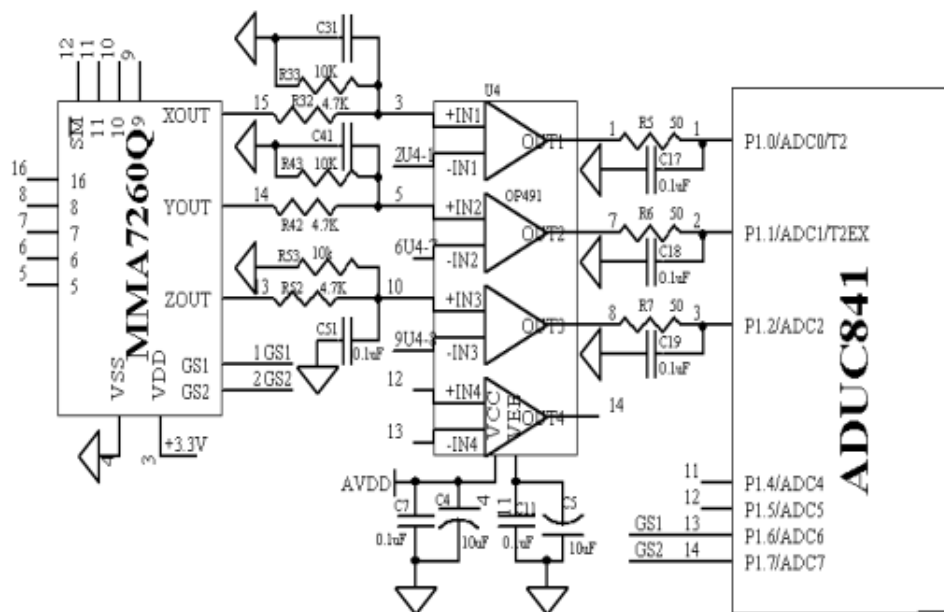


Figure 37: Interface circuit diagram

This device is a simple solution to the gait sampling application and utilizes low sampling rates. In order to make the testing easier, the device was designed to fit in a small package as to be wearable on a person's wrist. Although nothing was mentioned about power usage, it can be assumed that the device operated on a battery in the range of 3V.

7.4 Comparisons

Table 6 below compares the above related works with the MOSES DAQ with respect to area, power and speed. While the related works have a primary focus on speed, the advantage of the MOSES DAQ is its low area design.

Table 6: Design comparisons

| Design | Area | Power | Throughput |
|---------------|---|---|---|
| Guo et al. | Increased IC usage and board routing | Unknown power usage, assume higher because larger number of devices | 200 kSPS on 8 channels of simultaneous sampling |
| Gaudin et al. | Many electronic components over 2 boards. 360 cm ³ | Unknown, assume >10 W | 1 MSPS on 8 channels of simultaneous sampling |
| Rong et al. | Single board, fits on a person's wrist. | Unknown, assume <1 W | 250 SPS on 3 channels of sequential sampling |
| MOSES DAQ | Decreased IC usage on a 4 x 8 cm board. | 101 mW | 240 kSPS on 4 channels of sequential sampling |

CHAPTER 8: FUTURE WORK

This chapter touches on potential areas of improvement and topics which could not be explored in more detail due to time restrictions.

8.1 XADC Calibration

The XADC has the capability to self-configure its own calibration coefficients to correct gain and offset errors that may occur during conversion. In order to sample four channels at 240 KHz, this calibration was omitted from the design in focus. This is because the ADCCLK frequency is 25 MHz, and it takes 26 ADCCLK cycles to complete a single conversion cycle. Sampling 4 channels takes 4 conversion cycles, which can be accomplished in about 240 KHz. Adding the calibration channel takes an additional 4 conversion cycles, making the sequence last 8 conversion cycles in total. Having 4 channels and the calibration would only have a rate of 120 KHz. If future work requires more accurate conversions, then the sampling rate will have to be decreased in order to allow for calibration and four channels of input conversion.

8.2 Event Monitoring

An idea proposed at the beginning of development was to incorporate event monitoring logic into the FPGA, which analyzed conversion results for “areas of interest”. The logic would then decide whether or not to store this data in the off-chip NAND memory. This would be beneficial for saving storage space but would not help reduce any power costs of operating the FPGA. The context behind “areas of interest” is that during the rocket launch, certain periods of time will have more “interesting” accelerations than others. For instance, during stage separation the vibrations characteristics might be particularly

interesting to the CubeSat team. One drawback however is that such digital logic may require large resource overhead and increase power costs significantly.

CHAPTER 9: CONCLUSIONS

This work outlined the design and testing of an acceleration analog-to-digital acquisition system to meet high-speed and low-area requirements. This project is designed for use in a CubeSat P-POD, where area for electronic components is extremely limited. The design specification required sampling an accelerometer output for signals up to 100 KHz on four channels. To address these challenges, a Xilinx 7-series FPGA was chosen to be the basis of the acquisition system. 7-series FPGAs contain an instance of the Xilinx ADC (XADC) and only require an external configuration memory and oscillator to operate effectively. By containing both the ADC and custom hardware accelerators on a single IC as opposed to a processor-based system, this acquisition system was able to save board space compared to other systems. Furthermore, the XADC is able to support fast data sampling rates due to its efficient architecture. It utilizes a dual 1 MSPS ADC scheme with overlapping acquisition and conversion phases for multiple channels. That is, when one ADC is converting the sampled value on one channel, the second ADC can simultaneously be acquiring the value of another channel on its Track-and-Hold capacitor. This scheme greatly increases sample throughput and allows for the final rate of 240 KSPS on 4 channels each.

The use of an FPGA also allows for the implementation of custom hardware modules. This data acquisition subsystem must transfer the four sampled conversion results from each sensor channel to the processing subsystem on MOSES which stores the values in a NAND flash. This design is able to utilize the FPGA to implement custom Finite State Machines (FSMs) which controls the transmissions of a single SPI Master bus with only a single data line. This means the inter-board bus only uses three signal

lines: data, clock and chip select. This saves routing space between boards, vastly simplifying the housing structure of MOSES.

REFERENCES

- [1] A. Mehrparvar. (2014, February 20). *CubeSat Design Specification (Rev 13)*. [Online]. Available:
http://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/56e9b62337013b6c063a655a/1458157095454/cds_rev13_final2.pdf
- [2] IEEE. (2006, April 7). *IEEE Standard for Verilog Hardware Description Language (IEEE Std 1364-2005)*. [Online]. Available:
<http://ieeexplore.ieee.org/servlet/opac?punumber=10779>
- [3] Xilinx. (2016, May 24). *7 Series FPGAs and Zynq-7000 All Programmable SoC XADC Dual 12-bit 1 MSPS Analog-to-Digital Converter (v1.8)*. [Online]. Available:
http://www.xilinx.com/support/documentation/user_guides/ug480_7Series_XADC.pdf
- [4] Micron. (2006). *NAND Flash 101 (Rev B)*. [Online]. Available:
<http://www.ece.umd.edu/~blj/CS-590.26/micron-tn2919.pdf>
- [5] Piezo Systems. *Piezoelectric Bending Sensors*. [Online]. Available:
<http://www.piezo.com/prodbg8dqm.html>
- [6] H. Guo, et. al. "Continuous and Real-Time Vibration Data Acquisition and Analysis System Based on S3C6310 and Linux," in 2013 5th Int. Conf. on Measuring Technology and Mechatronics Automation, 2013 © IEEE. DOI: 10.1109/ICMTMA.2013.98
- [7] J. Friedel, S. McKibbin, "Thermal Analysis of the CubeSat CP3 Satellite," B.S. Senior Project, Dept. Aero. Eng., Cal Poly State Univ., San Luis Obispo, CA, 2011. [Online]. Available:
<http://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1054&context=aerospace>
- [8] Xilinx. (2015, May 27). *7 Series FPGAs Overview (v1.17)*. [Online]. Available:
http://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf
- [9] J. Karki. (2002, September). *Active Low-Pass Filter Design*. [Online]. Available:
<http://www.ti.com/lit/an/sloa049b/sloa049b.pdf>
- [10] Texas Instruments. *WEBENCH Design Center*. [Online]. Available:
http://www.ti.com/ltds/ti/analog/webench/overview.page?DCMP=analog_power_mr&HQS=webench-pr

- [11] Microchip Technology. (2004). *12-Bit DAC with SPI Interface MCP4921*. [Online]. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/21897a.pdf>
- [12] C. Gaudin, et. al. "A wireless high-speed data acquisition system for geotechnical centrifuge model testing," in *Meas. Sci. Technology*, Vol. 20, No. 9. 2009 © IOP. DOI: <http://dx.doi.org/10.1088/0957-0233/20/9/095709>
- [13] L. Rong, et. al. "A Wearable Acceleration Sensor System for Gait Recognition," in *2007 2nd IEEE Conf. on Industrial Electronics and Applications*, 2007 © IEEE. DOI: <http://dx.doi.org/10.1109/ICIEA.2007.4318894>