

# Applied High Resolution Digital Control for Universal Precision Systems

by

Aaron John Gawlik

B.S., Mechanical Engineering, University of Minnesota (2006)

Submitted to the Department of Mechanical Engineering  
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

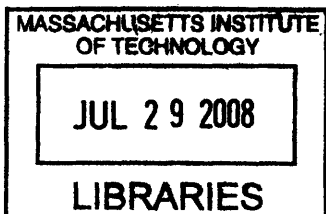
June 2008

© Massachusetts Institute of Technology 2008. All rights reserved.

Author .....  
Department of Mechanical Engineering  
May 13, 2008

Certified by.....  
David L. Trumper  
Professor of Mechanical Engineering  
Thesis Supervisor

Accepted by.....  
Lallit Anand  
Chairman, Department Committee on Graduate Students



**ARCHIVES**



# Applied High Resolution Digital Control for Universal Precision Systems

by

Aaron John Gawlik

Submitted to the Department of Mechanical Engineering  
on May 13, 2008, in partial fulfillment of the  
requirements for the degree of  
Master of Science in Mechanical Engineering

## Abstract

This thesis describes the design and characterization of a high-resolution analog interface for dSPACE digital control systems and a high-resolution, high-speed data acquisition and control system. These designs are intended to enable higher precision digital control than currently available. The dSPACE system was previously designed within the PMC Lab and includes higher resolution A/D and D/A interfaces than natively available. Characterization on the custom A/D channel demonstrates 20.1 effective bits, or a 121 dB dynamic range, and the custom D/A channel demonstrates 15.1 effective bits, or a 91 dB dynamic range. This compares to a 15.7 effective bits on the A/D dSPACE channel and 12.3 effective bits on the D/A dSPACE channel. The increased resolution is attained by higher performance hardware and oversampling and averaging the A/D channel. The sampling rate is limited to 8 kHz.

The high-resolution, high-speed data acquisition and control system can sample two A/D channels at 2.5 MHz and display/save an acquired one second burst. The A/D channel is characterized at 109 dB dynamic range with a grounded input and 96 dB dynamic range, or 0.74 nm RMS over a 50  $\mu\text{m}$  range, with a fixtured capacitive probe. Acquisition at 2.5 MHz and closed-loop control at 625 kHz sampling rate is implemented on a National Instruments FPGA. The A/D circuit was designed and built on a custom printed circuit board around the commercially available AD7760 sigma-delta converter from Analog Devices and includes fully differential  $\pm 10$  V inputs, a dedicated microcontroller to provide an initialization sequence, and digital galvanic isolation. LabVIEW FPGA code demonstrates arbitrary transfer function control implementation. The digital platform is applied to a 1-DOF positioner to demonstrate 0.10 nm RMS control over a 10  $\mu\text{m}$  mechanical range when filtered to the 1.5 kHz closed-loop bandwidth, which is limited by the A/D converter architecture propagation delay.

Thesis Supervisor: David L. Trumper  
Title: Professor of Mechanical Engineering



## Acknowledgments

I would first like to thank my advisor, Professor David Trumper. Professor Trumper has been an inspiration with his knowledge of all things mechanical, electrical, and interdisciplinary-related fields and provides insight that I would otherwise not have considered. His hands-off style has allowed me to learn on my own and yet have the guiding support to keep my perfectionist traits on track towards a final destination. Because of him I am a more confident than ever in my engineering skills. His review of this thesis was essential to my own understanding and hopefully clearly conveying my work.

I would also like to thank my family for their continual support, particularly my brother Noah who has personal experience in many applications that I have dealt with throughout this thesis work. His experience with LabVIEW FPGA implementations and general digital controls provided an excellent soundboard as well as an honest opinion.

David Otten was always willing to assist with the high-resolution dSPACE system that he designed, and his two notebooks of documentation were essential to understanding, characterizing, implementing, and improving the design.

The amazing educators between MIT and Harvard provided an understanding in multidiscipline fields that I would never have expected to achieve two years ago. Particularly I need to thank Professors Tom Hayes and Paul Horowitz for their fundamentally applied electronics course at Harvard, providing the skills to write firmware and design the AD7760 A/D PCB. Professor Horowitz was also a willing soundboard on high-resolution A/D characterization. I need to thank Professor James Roberge for pushing the students of 6.331 to constantly achieve more and innovate on traditional designs. Numerous early mornings finishing labs, piece sets, and design problems were worth the effort considering what I learned and my gained confidence in analog design. Dr Kent Lundberg was critical thanks to his comprehensive notes and recitation lectures.

The National Instruments FPGA implementation for high-speed data acquisition

and control provided a source of frustration at times but was eased by Lesley Yu - NI field applications engineer, Carla Uribe - NI applications engineer, and Erik Goethert - Boston Engineering program manager.

Lastly I need to thank my fellow graduate students and support staff throughout MIT. Ian MacKenzie provided a fundamental understanding of control and electromagnetics, as well as time to work together on the high-speed AFM project for which the FPGA-based system was designed for. We also found ourselves on the same course track and he was always willing to discuss how he understood problems from MEMS processes to the hybrid-pi model to switching power converters. Kevin Miu was also invaluable with all things controls, electrical, lab, and MIT related. His work effort has been an inspiration. Other students that influenced my work include Levi Wood, Eerik Hantsoo, Adam Wahab, Dan Burns, Dan Kluk, and Dean Ljubicic. Laura Zagonjori was invaluable with purchasing, administrative tasks, and tracking when Professor Trumper would be next available as well as providing a reprieve from a windowless basement lab. I need to thank Lenny Rigione of the Ceramics Processing Research Laboratory for use of a 6.5 Keithley digital multimeter. Finally Leslie Regan and the mechanical engineering graduate support staff have helped make the past two years possible and even enjoyable.

*For Mom & Dad*





# Contents

<b>1</b>	<b>Introduction</b>	<b>21</b>
1.1	Project Goal and Summary . . . . .	21
1.2	Motivation and Context . . . . .	32
1.2.1	Digital Control Systems . . . . .	33
1.2.2	Analog-to-Digital Conversion Methods . . . . .	36
1.2.3	Digital-to-Analog Converting Methods . . . . .	47
<b>2</b>	<b>dSPACE Interoperable High-Resolution Analog Interface</b>	<b>51</b>
2.1	System Description and Design . . . . .	52
2.2	System Characterization and Results . . . . .	57
<b>3</b>	<b>High-Speed, High-Resolution Digital Platform Requirements and Selection</b>	<b>71</b>
3.1	Application Description . . . . .	71
3.2	ADC Requirements and Selection . . . . .	73
3.3	Digital Platform Requirements and Selection . . . . .	80
3.4	DAC Requirements and Selection . . . . .	91
<b>4</b>	<b>24-bit A/D Circuit &amp; PCB Design</b>	<b>95</b>
4.1	Analog Interface . . . . .	98
4.2	Power Regulation and Decoupling . . . . .	108
4.3	Digital Interface and Microcontroller Design . . . . .	110
4.4	PCB Construction and Debugging . . . . .	117

<b>5</b>	<b>LabVIEW Control Software</b>	<b>121</b>
5.1	High Level Layout . . . . .	123
5.2	A/D Acquisition Interface . . . . .	127
5.3	D/A Output Interface . . . . .	132
5.4	Digital Control Implementation . . . . .	133
5.4.1	PID Control . . . . .	138
5.4.2	IIR Control from Arbitrary Discrete Transfer Function . . . . .	140
5.4.3	Additional Filter Implementations . . . . .	148
5.5	User Interface and Post-Processing . . . . .	151
<b>6</b>	<b>Experimental Results</b>	<b>155</b>
6.1	A/D Characterization Results . . . . .	155
6.2	Sub-Nanometer Position Control Results . . . . .	163
<b>7</b>	<b>Conclusions and Suggestions for Future Work</b>	<b>177</b>
7.1	Conclusions . . . . .	177
7.2	Suggestions for Future Work . . . . .	180
<b>A</b>	<b>Schematics</b>	<b>187</b>
A.1	dSPACE High-Resolution ADC PCB Schematic . . . . .	188
A.2	dSPACE High-Resolution DAC PCB Schematic . . . . .	190
A.3	AD7760 PCB Schematic, Rev 4 . . . . .	191
A.4	AD7760 PCB Bill of Materials, Rev 4 . . . . .	196
<b>B</b>	<b>AD7760 Microcontroller Firmware</b>	<b>197</b>
<b>C</b>	<b>LabVIEW FPGA Code</b>	<b>207</b>
C.1	A/D Acquisition State Machine . . . . .	207
C.2	D/A Output State Machine . . . . .	212
C.3	Filter/Controller FPGA Code . . . . .	215
C.3.1	Matlab Transfer Function Output m-file . . . . .	215
C.3.2	LabVIEW Generator Code . . . . .	216

C.3.3	LabVIEW Generated Filter Code . . . . .	220
C.3.4	Example of LabVIEW FPGA Control Implementation: Lag, Triple-Lead . . . . .	222



# List of Figures

1-1	dSPACE high-resolution DAC (Left) and ADC (Right) PCB. . . . .	23
1-2	A/D noise floor (left) and A/D response to 10 Hz sine wave (right). The standard dSPACE (DS) and out high-resolution (HR) A/D channel are compared. On the left is the zero-input case. The graph on the right shows the response to a 1 mV amplitude sine wave. . . . .	24
1-3	Small amplitude D/A output. The standard dSPACE D/A and and our high-resolution (HR) D/A are compared. . . . .	24
1-4	ADC PCB (quarter shown for scale). . . . .	27
1-5	IIR filter canonical control direct form II block diagram. . . . .	29
1-6	Simplified schematic of sigma-delta analog operation, adapted from [1].	40
1-7	Quantized sine wave with $n = 2$ bits. . . . .	42
1-8	Block diagram of sigma-delta modulator, adapted from [1]. . . . .	43
1-9	Second-order sigma-delta modulation, adapted from [1]. . . . .	44
1-10	Sigma-delta modulation noise distribution. . . . .	44
1-11	Sigma-delta FIR filter response [2]. . . . .	45
1-12	D/A converter hardware architectures. 1 - Weighted-resistor network. 2.1 - Voltage output R-2R resistor network. 2.2 - Current output R-2R resistor network. The output node can be terminated at a virtual ground of an op-amp to form a current-to-voltage converter. 3 - Segmented implementation . . . . .	48
2-1	dSPACE high-resolution DAC (Left) and ADC (Right) PCB. . . . .	52

2-2	Magnitude frequency response of averaging filter for $N = 100$ at 800 kSPS, before decimation. . . . .	54
2-3	dSPACE data flow with high-resolution platform, adapted from [3]. . . . .	56
2-4	dSPACE subsystem timing [4]. . . . .	58
2-5	dSPACE and high-resolution latching timeline at 8 kHz sampling. The dSPACE channels are represented by DS and the high-resolution are represented by HR. The time scale is microseconds. . . . .	58
2-6	Phase delay of 100 Hz sine wave for several connection options. The time axis is in milliseconds. Digital system sample rate is 8 kHz. . . . .	60
2-7	Frequency response comparison, up to the Nyquist frequency. The frequency response as measured with the dSPACE DSA for both magnitude (top) and phase (middle) and the magnitude frequency response as measured with the HP DSA (bottom). . . . .	61
2-8	Dynamic signal analyzer configurations for dSPACE software DSA (left) and hardware HP DSA (right). . . . .	62
2-9	A/D noise floor (Left) and A/D response to 10 Hz sine wave (Right). . . . .	63
2-10	High-resolution characterization schematic for A/D and D/A channels. . . . .	64
2-11	RMS noise on A/D channels across full input range. . . . .	65
2-12	Asymmetric RMS noise on high-resolution A/D channel across full input range. The original configuration has two capacitors references to common and the revised configuration has a single capacitor referenced between the differential signals. . . . .	66
2-13	High-resolution analog front-end configurations: asymmetric high-noise configuration (left) and symmetric low-noise configuration (right). . . . .	66
2-14	High-resolution D/A glitch. . . . .	67
2-15	D/A baseline noise with 1 MHz low-pass filter on differential amplifier. . . . .	68
2-16	RMS noise present on D/A channels across full range. A single dSPACE D/A channel (dach1) is provided against several high-resolution D/A channels (daxx). . . . .	69
2-17	Small amplitude D/A output. . . . .	70

3-1	X3-SDF hardware by Innovative Integration [5]. . . . .	78
3-2	Real-Time computer developed by Xiaodong Lu [6]. . . . .	81
3-3	VMETRO embedded computing FPGA block diagram [7]. . . . .	87
3-4	Data acquisition and control hardware overview. . . . .	88
3-5	National Instruments PXI chassis with embedded controller and FPGA board . . . . .	90
4-1	ADC PCB functional diagram. . . . .	96
4-2	ADC PCB with functional areas (quarter shown for scale). . . . .	97
4-3	Differential input shift and scaling; analog input to PCB (left) and analog input to AD7760 IC (right). . . . .	99
4-4	Fully differential amplifier symbol. . . . .	100
4-5	Simplified fully differential amplifier internal circuitry [8]. . . . .	101
4-6	Fully differential amplifier with scaling and anti-aliasing components. . . . .	102
4-7	Half-circuit analysis of symmetric fully differential amplifier with shunt capacitor. The single capacitor (left) between differential signals can be replaced by two capacitors (right) referenced to ground. . . . .	104
4-8	Analog input differential amplifier anti-alias configuration frequency response. . . . .	106
4-9	Analog input PCB layout. . . . .	107
4-10	PCB voltage regulation. . . . .	108
4-11	Example of AD7760 supply decoupling. . . . .	109
4-12	EMI suppression equivalent circuit. . . . .	110
4-13	AD7760 PCB digital interface and components. . . . .	111
4-14	Microcontroller code block diagram. . . . .	114
4-15	AD7760 PCB clock management block diagram. . . . .	115
4-16	Level translator simplified schematic [9]. . . . .	118
5-1	FPGA high level layout. . . . .	124
5-2	Digitized control platform block diagram. . . . .	125
5-3	FPGA functional elements. . . . .	125

5-4	Block diagram of discrete sampling rate compressor, adapted from [10].	126
5-5	A/D to FPGA signal propagation delays. . . . .	128
5-6	A/D acquisition state diagram. . . . .	129
5-7	Example of LabVIEW FPGA code for A/D acquisition - <i>RaiseRD</i> state.	130
5-8	A/D data sample output two's complement format. . . . .	131
5-9	Converting two's complement to signed 32-bit integer. . . . .	132
5-10	D/A output state diagram. . . . .	134
5-11	FPGA sequential (left) versus parallel/pipeline (right) processing. Pipeline processing can be implemented with shift registers (top) or feedback nodes (bottom). . . . .	135
5-12	Block diagram of discrete downsampler, adapted from [10]. . . . .	136
5-13	FIR $N = 2$ pole-zero map and frequency response at 2.5 MHz sampling rate. . . . .	136
5-14	Expected frequency response for the digital control system. . . . .	138
5-15	LabVIEW PID block. . . . .	139
5-16	IIR filter canonical control direct form II block diagram. . . . .	141
5-17	FPGA IIR filter code generation block diagram. . . . .	143
5-18	Quantizer error introduction in a fixed-point filter [11]. . . . .	143
5-19	Discrete-Time integrator with anti-windup example. . . . .	149
5-20	$N$ -point average window FPGA implementation. . . . .	150
5-21	Front panel user interface. . . . .	153
6-1	Grounded input A/D count histogram. . . . .	156
6-2	Grounded input time response in a 400 $\mu s$ interval(top) and corresponding unfiltered FFT (bottom). . . . .	157
6-3	A/D, FPGA, D/A system frequency response. The magnitude of the measured digital platform with delay and the expected with delay are coincident and thus the measured result is not independently visible.	159
6-4	Power density spectrum to 1 kHz sine wave with coarse (left) and fine (right) scale. . . . .	159



6-5	Grounding diagram for capacitive probe measurements with “star” earth ground configuration on experimental AFM scanner application. Stationary/fixtured probe tests use the same configuration, although the “isolation table” is replaced by the fixture. . . . .	164
6-6	FFT of the response of 50 $\mu\text{m}$ probe on stationary target with 298 $\mu\text{V}$ RMS unfiltered baseline noise. This was measured on with the custom A/D PCB hardware and LabVIEW acquisition software. . . . .	165
6-7	2-DOF high-scan rate positioner CAD model [12]. . . . .	166
6-8	2-DOF high-scan rate positioner hardware. . . . .	167
6-9	Spring-mass-damper mechanical model. . . . .	168
6-10	Z-axis measured open-loop and expected system loop transmission on dSPACE control platform [12]. . . . .	169
6-11	Controlled z-axis time response (top) and FFT (bottom). . . . .	170
6-12	Relative comparison of unfiltered measured RMS position with dSPACE and FPGA digital platform closed-loop control to a constant reference over 20 ms (left) and 200 $\mu\text{s}$ (right). . . . .	171
6-13	Z-axis measured open-loop and closed-loop loop transmission on FPGA control platform. . . . .	172
6-14	Z-axis measured step response and error. . . . .	173
6-15	Loop transmission measurement scheme. . . . .	174
6-16	Parametric amplitude control loop [12]. . . . .	175
7-1	High-resolution D/A voltage reference noise and effect of passive low-pass filtering: the voltage reference noise measured with a Tektronix AM502 differential amplifier and 1 MHz low-pass filtering (left), the voltage reference noise measured with the differential amplifier and 30 kHz low-pass filtering (middle), and the voltage reference after a 4 kHz passive low-pass filter measured with the differential amplifier and 1 MHz low-pass filtering (right). . . . .	181

7-2 Recommended high-resolution, high-speed data acquisition and control environment. Separate acquisition and control A/D channels are used for data acquisition and control. The D/A converter is replaced with one capable of a high output sample rate. . . . . 185

# List of Tables

2.1	dSPACE Subsystem Timing . . . . .	59
2.2	A/D Noise Floor Comparison . . . . .	63
3.1	Analog-to-Digital Converter IC Commercial Options . . . . .	75
3.2	Analog-to-Digital Converter Commercial Options . . . . .	76
3.3	Innovative Integration X3-SDF Implementation Estimated Costs . . . . .	80
3.4	Custom Real-Time Computer of Xiaodong Lu Estimated Costs . . . . .	84
3.5	Xilinx Development Platform Estimated Costs . . . . .	85
3.6	Third-Party Digital Platform Estimated Costs . . . . .	86
3.7	National Instruments Digital Platform Estimated Costs . . . . .	89
3.8	Digital Platform Estimated Comparison . . . . .	91
3.9	Digital-to-Analog Converter IC Commercial Options . . . . .	92
4.1	Differential Amplifier Component Values . . . . .	105
4.2	Analog Input Differential Amplifier Anti-alias Pole Locations . . . . .	106
6.1	Capacitive Probe Characterization and Baseline Measurement Results . . . . .	162



# Chapter 1

## Introduction

For any precision motion control application, it is critical to maintain precision among varying engineering fields and through the combination of actuators, mechanical systems, sensors, electronics and digital computations, which generally requires an advanced knowledge and application of structural mechanics, design, analog electronics, digital electronics, electromagnetics, signal processing, and control. This thesis focuses on enabling precision motion control hardware systems by improving or designing digital control environments that increase performance over what is currently available.

Precision control has a variety of definitions in a variety of applications. Precision is technically the degree to which a measurement (e.g., the mean estimate of a treatment effect) is derived from a set of observations having small variation (i.e., close in magnitude to each other) [13]. A narrow confidence interval indicates a more precise estimate of effect than a wide confidence interval. This is applicable to digital data, where a more precise numerical value contains a greater number of meaningful bits.

### 1.1 Project Goal and Summary

This thesis focuses on two separate high precision digital control systems for different applications. The first focuses on creating more precise analog-to-digital (A/D) and digital-to-analog (D/A) interfaces for the commonly used dSPACE [14] control

platform. The Precision Motion Control (PMC) Laboratory at the Massachusetts Institute of Technology, as well as the mechatronics and digital control graduate courses use varying products from dSPACE, Inc [14]. The platform is intended as an embedded control environment with built-in peripherals, providing a real-time environment that can be programmed with Matlab's Simulink Real-Time Workshop [15]. The DS1103 dSPACE platform provides 16-bit A/Ds and 14-bit D/As with double floating-point, or 64-bit sliding window, calculations. Loop rates up to 100 kHz are possible if the calculation load is small.

David Otten, a research scientist previously with the PMC Laboratory, designed and built these interfaces [4] as a universal high-resolution peripheral option opposed to the dSPACE native analog interfaces. The design was initially applied to control a sub-atomic measuring machine (SAMM) at the University of North Carolina-Charlotte (UNCC) [16]. The high-resolution system by Otten required the physical design of the A/D and D/A channels and their software interface to the real-time dSPACE environment and Simulink functions. Up to 8 inputs or 6 outputs could be interfaced through a modular breakout PCB and Simulink software. Each A/D and D/A channel is built on an individual PCB with dedicated power regulation. These are shown in Figure 1-1. A distinct feature are digital isolators for galvanic isolation which break ground loops between the analog hardware plant and the digital environment, a commonly significant source of disturbances or noise in precision systems.

High-speed sampling and averaging is used to increase the A/D resolution to 20.1 effective bits. A dedicated DSP on the A/D PCB sums the A/D samples at 800k samples per second (SPS) and counts the number of samples. The sum and count is then transferred to dSPACE hardware through the software interface where it is averaged. This resolution compares to 15.7 effective bits as measured with the dSPACE A/D channels. Figure 1-2 shows a comparison between the dSPACE (DS) and the high-resolution (HR) channel. Significant quantization relating to errors distributed over several LSB are apparent in the dSPACE baseline noise response. These quantization levels are 305 mV/LSB for a 16-bit converter on a 20 V range. The

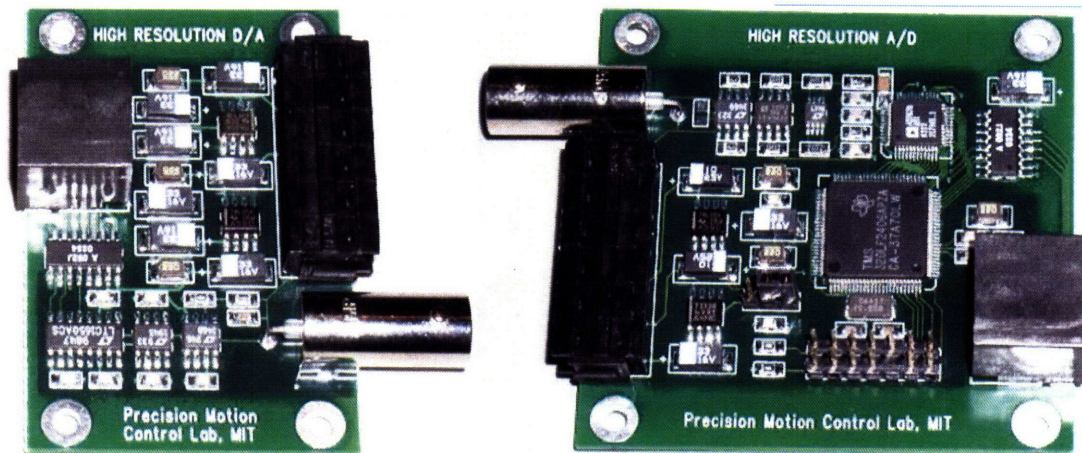


Figure 1-1: dSPACE high-resolution DAC (Left) and ADC (Right) PCB.

high-resolution measurement however displays no discernable quantization levels and has a noise level of  $15 \mu\text{V}$  RMS. Likewise, the 1 mV amplitude sine wave demonstrates the increased resolution performance as well.

There was little characterization data available on the performance of the two channels when I inherited the high-resolution design. Several anomalies appeared in initial tests that required design changes. The most significant was a change to the fully differential analog front-end input of the A/D converter. Initially the RMS noise of a digital sample would vary from 16 to over 20 effective bits based on the input voltage level. This issue was solved by altering the configuration of anti-aliasing capacitors.

A 16-bit D/A PCB was designed and built as a companion to the A/D channel which also utilized digital isolation a small footprint so it could be located near the analog plant. Figure 1-3 demonstrates the increased performance for the high-resolution channel against the dSPACE channel for a small amplitude output. The dSPACE D/A has a quanta size of 1.2 mV for its 14-bit converter and the high-resolution D/A has a quanta size of 305 mV for the 16-bit output.

This increased resolution comes at the cost of a maximum loop rate of 8 kHz. A slave DSP on the dSPACE hardware is used to interface with the custom PCBs. The slave DSP needs to interact with the main processor that runs the Simulink

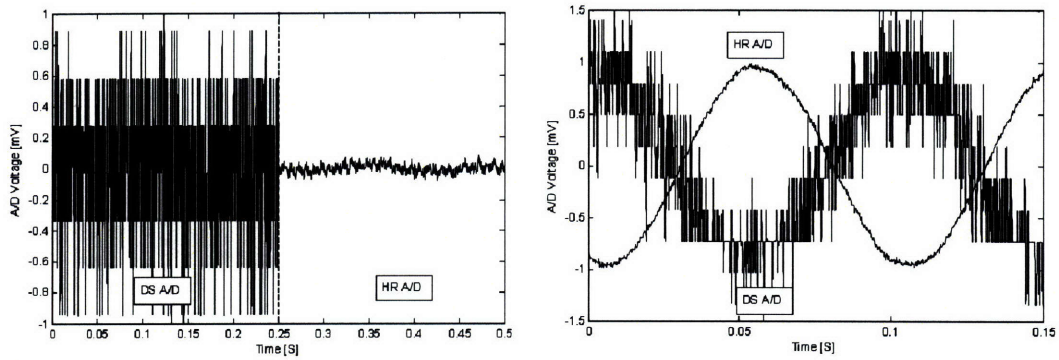


Figure 1-2: A/D noise floor (left) and A/D response to 10 Hz sine wave (right). The standard dSPACE (DS) and our high-resolution (HR) A/D channel are compared. On the left is the zero-input case. The graph on the right shows the response to a 1 mV amplitude sine wave.

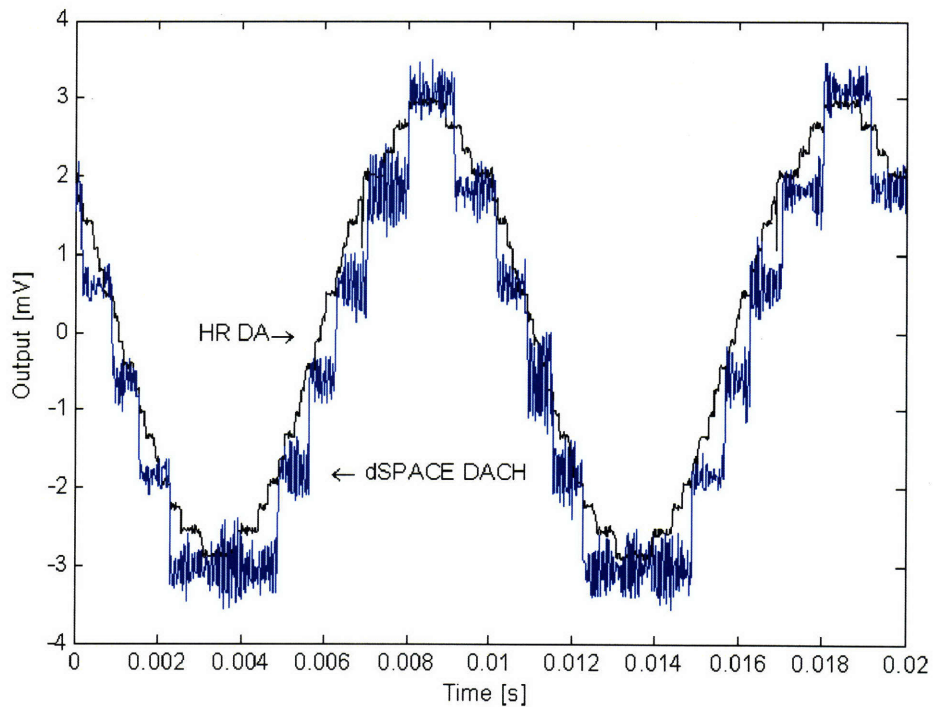


Figure 1-3: Small amplitude D/A output. The standard dSPACE D/A and our high-resolution (HR) D/A are compared.



application and does so through a communication buffer that requires communication time. The slave DSP then clocks data to D/A channels and from A/D channels, which requires 32 clock cycles. The total process time is 113  $\mu\text{s}$  and is the limiting factor for the maximum loop rate.

Additional improvements to the prior system designed by David Otten included debugging of why only 6 of 8 D/A channels were available at a time. I found that although software provided an 8-bit port in software, only 7 bits were available in the cabling pinout. Further, of these 7 bits, one was not correctly interfaced. The 7th D/A channel was implemented but in order to add the 8th channel additional steps would need to be added to the slave DSP and thus further decreasing the maximum loop rate.

Other than the maximum loop rate constraint, these high-resolution peripherals appear as any dSPACE peripheral does and are essentially invisible to the user to implement and operate. Applications for this first work are systems requiring high-resolution with relatively low bandwidth, i.e. on the order of bandwidths of Hz. Examples include vibration isolation with bandwidths on the order of 10 Hz or precision atomic force microscopy, with bandwidths on the order of 100 Hz.

The second half of this thesis focuses on a high-speed, high-resolution data acquisition and control digital platform. The design was driven by specifications for a high-speed atomic force microscope which required over 20-bit resolution data acquisition at greater than 1 MHz sampling rates as well as real-time control. Data acquisition as opposed to control implies that while data must be sampled and stored at a given rate, processing of said data can be done off-line and at slower rates. These specifications require both hardware and software to be operating at high rates and high resolutions, so it is attractive to use the same hardware and software to close the control loop.

Various commercial options were evaluated and it was determined that we should design our own A/D PCB built around a 24-bit, 2.5 million samples per second converter that is interfaced to field programmable gate array (FPGA) with a host for data offloading and supervisory control. A custom A/D PCB was designed and

tested based on the Analog Devices AD7760 sigma-delta IC. Most of the design concepts were adapted from an evaluation board design available from Analog Devices. I considered purchasing the evaluation board and merely designing the interface electronics, however the board was never available for purchase over the course of the project. Furthermore, errors were discovered in the evaluation board design throughout the testing and debugging phase that would have limited its functionality had it been available.

The A/D PCB incorporates the same digital galvanic isolators as the high-resolution dSPACE PCBs, as well as a dedicated microcontroller to provide an initialization sequence for the A/D converter IC and supervisory control during operation. Locating the initialization sequence on the PCB actually reduces the hardware complexity because fewer digital isolators are required, since only unidirectional digital isolators are available at the required data rates. Additionally, the inexpensive microcontroller reduces expensive resources that need to be allocated from the FPGA. The operational A/D PCB is shown in Figure 1-4, with a U.S. quarter for scale.

Although the A/D converter is listed as a 24-bit converter, the dynamic range is much less at high sampling speeds. The datasheet claims only 100 dB SNR at 2.5 MSPS. Tests on a grounded input demonstrated 66  $\mu\text{V}$  RMS noise over a  $\pm 10$  V range, which is equivalent to 109 dB RMS noise. Tests on a high performance capacitive probe fixtured to a stationary target matched the noise characterization of the probe itself. The capacitive probe had a characterization noise level of 309  $\mu\text{V}$  RMS and the A/D measured 298  $\mu\text{V}$  RMS without any additional filtering.

No viable commercial options were available at the time of initial design of the A/D PCB. Since that time a commercial option has become available from Innovative Integration [17] based on the same A/D hardware and similar processing hardware has become available. In retrospect, using that commercial board would have saved some repetitive design and debugging, however it would be less advantageous considering cost and future flexibility in implementation and expansion.

Several options existed as the digital processing platform. These ranged from a custom designed and programmed array of dedicated digital signal processors (DSP)

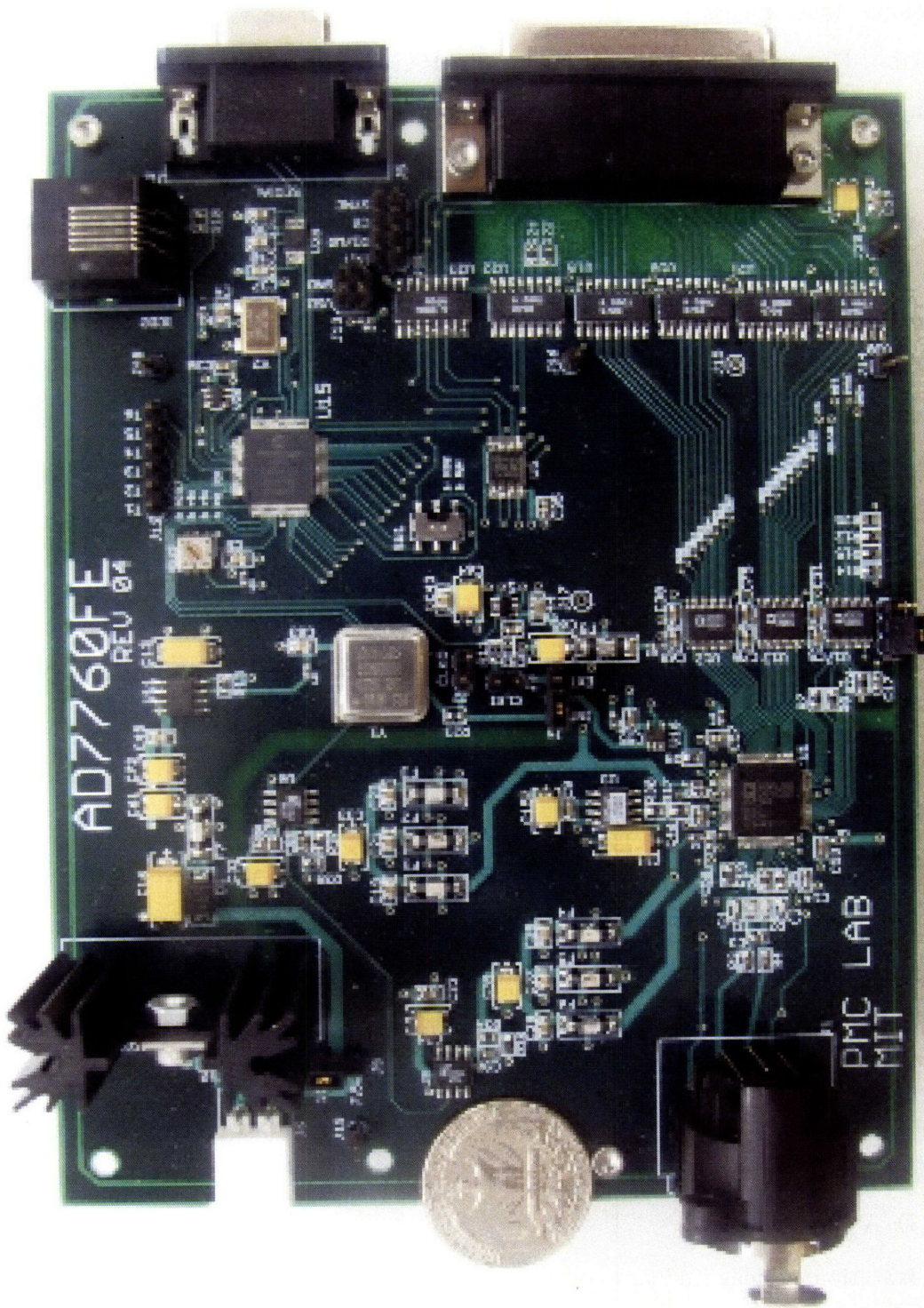


Figure 1-4: ADC PCB (quarter shown for scale).

to third-party hardware and software. Ultimately an FPGA-based product from National Instruments [18] was selected that could operate with existing hardware in lab. This existing hardware included a PXI chassis with a dedicated real-time computer. This computer runs a real-time operating system and has much higher real-time performance than a Windows or even Unix-based system.

National Instruments also supplied a high-level graphical programming language and environment (LabVIEW) for all hardware aspects. This included the development platform for the FPGA program, real-time application, and supervisory host. FPGAs are physically thousands of reprogrammable logic units that are connected by reprogrammable interconnects with logic suited to fixed bits of simple data manipulation operations. LabVIEW provides high-level, complex operations which reduces the learning curve for writing FPGA code and allows implementation of more complex applications more quickly.

FPGA code in the LabVIEW environment was written for 2.5 MHz data acquisition and closed-loop control rates up to 625 kHz. Data acquisition from the D/A is completed with a finite state machine clocked at 80 MHz. LabVIEW also provides other complex features that can be easily implemented. For example, direct memory access was used to transfer the acquired samples directly into the host computer's memory without being delayed by host processing. Although sustainability tests were not completed, the system was able to store at 20 MB over one second bursts. This enabled data storage and off-line post-processing.

A D/A channel is also required to enable closed-loop operation. Following an available product survey, I decided that the previously designed high-resolution D/A channel for the dSPACE environment was an appropriate option for the FPGA system as well. However, the closed-loop cycle rate is limited to 625 kHz by the rate at which the D/A can be clocked.

Traditional linear feedback control can be quantitatively designed and is typically implemented with lead-lag control. The NI LabVIEW FPGA module provides a PID implementation, however it is limited in functionality and flexibility. The PID coefficients cannot be easily adapted to a lead configuration, and the data path is limited to

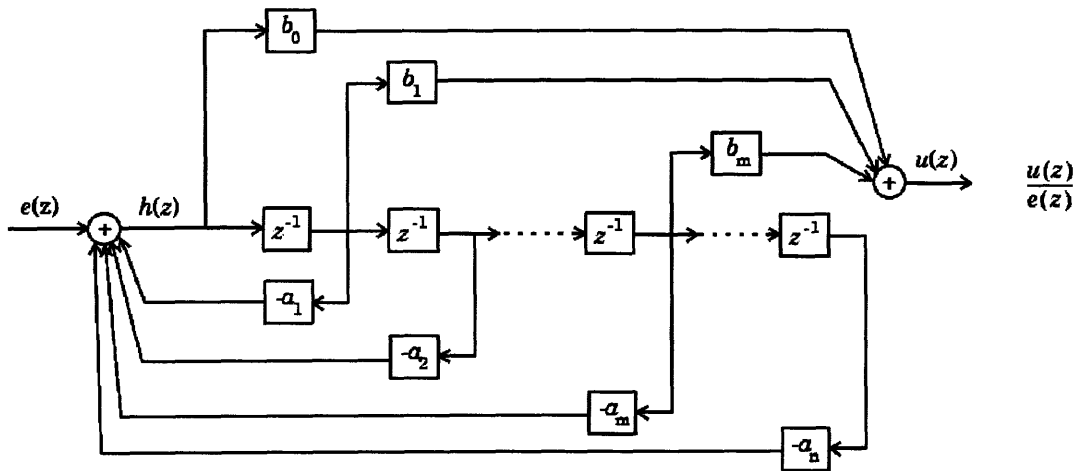


Figure 1-5: IIR filter canonical control direct form II block diagram.

16 bits whereas a 32-bit data path is preferred. The most efficient method to process a controller transfer function is in canonical direct form II, represented in Figure 1-5. The transfer function is implemented as zero coefficients  $b_i$  and pole coefficients  $a_i$ . The canonic form reduces the  $a_i$  and  $b_i$  coefficients to the minimal representation and direct form II reduces the number of delay/memory ( $z^{-1}$ ) elements required. The only operations required for processing a direct form II filter are addition, multiplication, and state delays. The filters are defined as discrete-time filters in the  $z$ -plane with sampling time  $T_S = 400$  ns. They are transformed from the  $s$ -plane with the Tustin transformation without warping. I found that time delays due to data transferring were minimized by implementing control before decimation.

A LabVIEW FPGA function is used to generate the controller filter code that is compiled to the FPGA. The filter is an infinite impulse response filter because it has coefficients in both the forward and reverse directions, indicating that there are non-zero poles and an impulse can persist infinitely. LabVIEW code was written to import an arbitrary floating-point control filter and convert it to a fixed-point filter. This introduces quantization errors as the overall word length and integer word lengths for the coefficients are individually specified. Coefficients that are widely spaced, such as 100, 0.1, and  $10^{-4}$ , provide issues in quantization because they require both a

wide range and decimal precision. For filters whose performance depends on closely spaced poles, particularly as they approach the unit circle of the z-plane, significant quantization errors can often reduce performance or introduce instability.

In our approach, control filters designed in the real-time domain are reduced to second-order stages. A cascaded transformation can be used in the LabVIEW filter generation, however quantization can only be specified for the entire system, whereas when separate control filters are manually cascaded then quantization can be specified for each filter individually. Quantization settings are also applied to other operations by the LabVIEW filter generation process, such as addition, multiplication, and delays. Delays are implemented as block RAM on the FPGA board and decrease the number of required registers.

Fixed-point filters and their operations are implemented as integer operations. This requires a pre- and post-scaling by a fixed decimal word length. Fixed-point multiplication also presents a challenging issue because for every operation of order  $n$ , the output order is  $2n$ . Two 16-bit integers multiplied together require a 32-bit output to avoid overflow saturation. Therefore a special multiplication block was created in LabVIEW that multiplies to the full precision and then translates back to the original order. This significantly reduces rounding errors. The block was implemented as parallel operations to use more readily available logic blocks on the FPGA and operate more efficiently. These multiplication blocks were also used to implement arbitrary gains that a user can vary. Along with traditional lead-lag transfer functions, other control oriented features are described and used. This includes saturation for integral anti-windup and reference signal generation.

This high-speed, high-resolution digital platform was implemented with AFM scanner hardware designed and built by Ian MacKenzie. The scanner has 2-axis position feedback from two high-performance capacitive probes from ADE [19], model 6501 with ranges of 40 and 50  $\mu\text{m}$ . These probes have a baseline noise of 181 and 309  $\mu\text{V}$  RMS, respectively, as characterized by ADE for  $\pm 10$  V outputs at a bandwidth of 100 kHz. When sampled with the custom A/D channel, an approximate 100 kHz disturbance was generated within the capacitive probe driver and measurement elec-

tronics. This is due to the A/D channel but the mechanism by which it is affecting the measurement electronics is not known, despite extensive tests. The disturbance can be measured even when the A/D channel is not connected but merely running in the vicinity. Varying grounding configurations were able to reduce this disturbance but it currently introduces the dominant noise content when the probe is measuring a stationary target.

The A/D channel is able to measure 252 and 298  $\mu\text{V}$  RMS unfiltered, respectively, on a stationary target at 2.5 MSPS. Control was implemented on one of the axes designed for 10  $\mu\text{m}$  range with the 40  $\mu\text{m}$  probe and achieved 430  $\mu\text{V}$  RMS unfiltered, or 0.86 nm RMS. When filtered to the 1.5 kHz closed-loop bandwidth, control achieved 0.10 nm RMS, equivalent to 111.7 dB dynamic range and 18.3 effective bits. The closed-loop sample rate was 625 kHz and the phase margin at 1.5 kHz crossover frequency was 37 degrees. The control scheme was a triple-lead, single-lag controller and was implemented as the series combination of a loop gain, three IIR lead filters, and lag with anti-windup. The loop required approximately 20 degrees of additional phase compensation due to the A/D and processing time delay. The A/D is a sigma-delta converter and thus introduces a propagation delay of 10.8  $\mu\text{s}$ . The remaining time delay leading to a total of 23.2  $\mu\text{s}$  is due to acquiring the data, passing the data between parallel loops, and passing the data through IIR control filters. The data processing is capable of a 400 ns sample rate with an arbitrary number of control filters because data pipelining is utilized. This maintains a high sample rate but also increases the propagation delay. Alternatively the sample rate can be decreased down to the output rate of 625 kHz but the time delays associated with data transfers would then be increased. The closed-loop bandwidth could be increased but the linear phase loss due to the time delays requires increasing lead control. This increasing lead compensation is limited by the magnitude roll-off.

The other axis consisted of a parametric amplitude control loop. Although the FPGA design is presented, it was not implemented in hardware. The single axis however demonstrates sub-nanometer control.

## 1.2 Motivation and Context

Analog electronics can resolve on the order of a part in a million in a carefully designed setup. This relates the absolute range to the resolution of the system. Precision control can thus be attained for meter ranges with sub-millimeter resolution, or micron ranges with sub-nanometer resolutions. Analog electronics are then able to control to this precision. The ability to actuate a given system and then sense that motion is another issue altogether. Precision motion control is driven by improving components with dominant noise contributions, which is a reason why piezoelectrics and electromagnetics are common actuators; their precision is commonly limited to the electronics driving them. This is also a reason why mechanical flexures are extremely popular for constrained motion as they allow linear motion without stiction and other discontinuous affects on a fine scale. Similarly, technology in capacitive sensors, encoders, and laser interferometry is providing higher precision position sensing.

Analog controls can be applied in contrast to digital controls. Analog systems are simple to implement for linear systems and high precision can easily be attained for high bandwidths with a low cost. However, analog controls are not very flexible. Digital systems on the other hand allow a multitude of control algorithms to be flexibly implemented, albeit at a greater cost, such as discontinuous, nonlinear, adaptive, or feedforward control. Digital systems and their implemented control are also not prone to environmental conditions, to the first order, as opposed to capacitors and resistors in analog systems. The maximum bandwidth for an analog system can easily be greater than 1 MHz with better than 10 ppm resolution. It is difficult to match these specifications with digital systems at this time.

This thesis works to improve the available resolution and bandwidth of digital control systems. Chapter 2 describes a design for a high-resolution analog interface that was previously designed and built by David Otten within the PMC lab as another option to native dSPACE A/D and D/A converters. The design is characterized and I describe improvements for a lower baseline noise. Chapter 3 presents requirements for a high-resolution and high-speed data acquisition and control digital platform as well



as viable options and the selected components. Subsequent chapters detail the design of the hardware, software, and control implementation as well as characterization and results when the digital system is applied to a 1-DOF positioner of 10  $\mu\text{m}$  range and sub-nanometer control at 1.5 kHz crossover frequency.

The remainder of this section describes various digital control, A/D, and D/A architectures. These three components determine the closed-loop precision and data rate. It is important to understand the background and available options for the various components and techniques within high-resolution digital systems and the associated interfaces.

### 1.2.1 Digital Control Systems

Digital control uses electronic logic to act on a system. The implemented hardware can range from an ASIC to a microcontroller to a full dedicated computer. The difference between a piezoelectric actuator and a stepper motor are analogous to the difference between an analog and a digital control system; the digital system is inherently finite precision whereas the analog system merely has a baseline noise floor. This introduces quantization in coefficients and operations. The analog-to-digital and digital-to-analog interfaces are also finite precision and introduce their own quantization. Another difference between analog and digital systems is propagation delays. Digital systems frequently have a non-negligible computation time. High data rates can be maintained by pipelined computations, however the time latency still introduces a phase lag at the bandwidth of interest, which is troublesome for closed-loop control bandwidth.

Digital sampling usually introduces a zero-order hold at its output due to the discontinuous nature of the input/output samples. The time delay from the input to the output of a digital system with ideal converters in this case is half the sampling time  $T_s$ . The time delay due to computations or latency in the digital system is  $T_d$ . The total delay time is then  $T_d + \frac{T_s}{2}$ . This demonstrates that it is necessary to minimize any system latency while also maintaining a high sampling rate for high bandwidth systems. Computations with increasing precision, such as floating-point

as opposed to fixed-point, require more time to complete.

The high-resolution, high-speed system described in Chapters 3 through 6 is implemented with a system that is expected to have a closed-loop bandwidth up to 5 kHz. Typically a digital system requires requires a sampling rate on the order of 10-20 times the closed-loop bandwidth [20], thus requiring a closed-loop rate of approximately 50-100 kHz for this bandwidth.

The digital system architecture determines  $T_d$  for a given controller. A digital system, or a real-time computer, needs to provide low latency real-time services as well as a user interface. Real-time services act on the signal and determine the controlled output at a fixed frequency. The user interface displays measured signals, allows user interaction with gains and controllers, and provides data logging. The earliest architecture to achieve these two services was the the Uni-Body architecture which operates with interrupts and a foreground-background architecture on a single computer. An interrupt is initiated at a fixed frequency. The interrupt then runs the real-time services in the background and the remaining time before the next interrupt constitutes the foreground where the user interface is processed. If the foreground processing consumes too many resources then loop jitter and latency is introduced. Running an operating system such as Windows requires a lot of resources and the fixed sampling rate needs to be decreased.

The next architecture is the Dual-Body which has a host and a target. The Uni-Body architecture is implemented on the target computer with a dedicated real-time operating system (RTOS). The host machine runs an operating system such as Windows and displays the semi-real-time data transmitted from the target machine. The target machine sampling rate is still constrained by processing an interrupt, the real-time services, and the foreground services. Examples of commercial Dual-Body architectures are Real-Time Windows Target by Mathworks, dSPACE, xPC Target by Mathworks, and the Real-Time Module from National Instruments. Most of these use a single dedicated processor, such as a PowerPC or computer chip from Intel or AMD.

Instead of interrupt-driven processing, polling operation can be implemented. This

removes the interrupt associated latency but also removes the host interface. This is generally not acceptable in real-time control applications, particularly in the controller development process or when data acquisition is required.

A multi-processor Dual-Body architecture is also becoming more popular, especially because dual and quadcore processors are decreasing in price. The processors communicate with each other over direct inter-processor data busses. These systems are capable of increased computing performance but are still limited by interrupt associated latency.

A Triple-Body architecture decouples the foreground and background threads on the target machine. This architecture dedicates one or more processors to the background real-time services tasks and a separate processor to the foreground threads and host machine interface. The foreground processor interfaces with the other processors on a shared data bus and multi-port RAM. This has been implemented in various applications [6, 21].

Thus far only traditional processor have been considered. Field programmable gate arrays (FPGA) are increasing in computational power while becoming less expensive. The logic gates are reconfigured for a specific application and the gates are essentially reconfigured to a customizable dedicated electric circuit with clocking rates up to 200 MHz or greater. The FPGA can have many separate logic circuits that are run in parallel. A single FPGA can then replace the multiple processors of a Dual- or Triple-Body architecture. The computations are limited however because the logic architecture is suited for fixed-point simple bit operations. Even division requires complex hardware to implement efficiently for both hardware utilization and operation rate. Newer FPGA models are beginning to incorporate dedicated processors directly into FPGA fabric, allowing for the high-speed data transfer and bit operations in the FPGA and complex data manipulation in the dedicated processor. Commercial vendors such as Innovative Integration [17] or VMETRO [22] provide an array of FPGA/DSP combinations intended for high loop rates.

Along with sampling speed and time latency, another consideration is the precision of the implemented computations. Data types from a single bit to signed integers to

double precision are available in processors and fixed-point types are available in FPGAs. Floating-point computations can be emulated in IP programs on FPGAs, however they are extremely resource intensive. The fixed-point data type introduces quantization that needs to be considered in an error budget for the digital platform. The series of computations also needs to be analyzed so there are not underflows or overflows due to rounding and saturation. Increasing data type resolutions require either increasing parallel hardware in an FPGA or a longer computation time in digital signal processors.

Chapter 2 focuses on the analog interfaces as they are implemented with a dSPACE system. Computation precision is considered briefly when the A/D oversample and average method is discussed. The limiting factor in the system design is system latency due to data transfer to/from and within the dSPACE multiple processors. The rest of this thesis considers a suitable architecture and then suitable hardware to implement for a high-resolution, high-speed digital acquisition and control platform.

## 1.2.2 Analog-to-Digital Conversion Methods

A primary focus of this research effort is the design of analog-to-digital interfaces. It is possible to design the converter from discrete components, however commercial options are available with different architectures to meet our requirements for various applications. This section details different analog-to-digital converter (ADC) technologies and the application each type is designed for.

State-of-the-art technology, in both research and commercial devices, demonstrates an inherent trade-off between resolution and sampling speed. Continuing development and advances are expected to follow this general trend [23]. High precision instrumentation as well the communications industry have continually pushed the limits of ADCs. Reviews of the state-of-the-art devices have been published every 3-6 years over the last several decades, and while earlier ones demonstrate the continuing trends, the limits are being pushed further due to technological advances [23, 24, 25].

Analog-to-digital conversion, as the name implies, is the interface between the

analog and digital environments. Many sensors, such as a capacitive probe or geophone, output an analog signal. This interface is then necessary in order to implement digital acquisition or control. Critical criteria include the ADC native resolution and signal distortion, or signal-to-noise ratio (SNR). Power consumption, the amount of hardware required, and the characteristics of erroneous readings are also important. Other factors include the sampling rate, generally measured in samples per second (SPS), and the throughput delay. In a pipeline architecture, data must pass through sequential stages. When a sample reaches one stage, a subsequent sample can enter the previous stage, thus numerous samples can sequentially be passing through the pipeline at a time. Therefore the differentiation must be made between the sampling rate period  $T_s$  and throughput delay  $T_d$ .

ADCs are available in several standard techniques. These include successive approximation, flash, integrating, sigma-delta, pipeline, and hybrid technique converters.

Consumer driven markets are currently pushing converters towards higher speeds at higher resolution while reducing power consumption. This is partly achieved by lower supply voltages. This however requires smaller signal voltages that are then more susceptible to noise from a variety of sources such as power supplies, references, digital signals, electromagnetic and radio frequency interference (EMI/RFI), and poor layout, grounding, and decoupling techniques. Moves away from bipolar devices also mean that ADC differential inputs are not generally referenced to ground and thus require differential amplifiers that can scale and shift the signal.

### **Flash Converter**

A flash, or parallel-encoded, converter uses  $2^n - 1$  comparators that are synchronously converted where  $n$  is the number of bits of resolution. The complementary comparator inputs are connected to a corresponding reference voltage on an equally spaced network of  $2^n - 1$  voltages. Digital logic then decodes the output to  $n$  bits. The digital value is found at the break between comparators being on and being off. This

method is by far the fastest method and can convert a sample on the order of several clock cycles. However, the hardware increases with the resolution. Not only does it add complexity to implement that many comparators, but tolerances also become extremely tight on the reference voltage network. Also, false comparator values in the thermometer code output can easily return a full scale error. Power consumption is a concern because each comparator has a minimum quiescent current. This current is increased in order to operate at high speeds. Assuming a 10-bit converter, 1k comparators would be required as well as logic devices. Assuming 1 mA quiescent current per comparator with a 5 V supply rail, the IC would need to dissipate over 5 W of power.

Typically flash converters do not require a sample and hold (S&H) because the sample is converted over a very small period and is not expected to be changing. Typically flash converters are available in 8 or 10-bit architectures with sampling rates up to 1 GSPS.

### **Successive Approximation Converter**

Successive approximation register (SAR) converters, as the name implies, estimate what the sample voltage is, compare the estimation to the sample, and then refine the estimation. The SAR converter uses DAC feedback and a comparator to compare the sample and estimation. A S&H circuit is required to maintain a constant voltage during the conversion period. The successive approximations are bitwise. All bits are initially zero and each bit under test is set to 1. The MSB comparison occurs first. If the sample is higher than the D/A voltage, the bit is left high and the next bit is tested. This requires  $n$  conversions for  $n$  bits of resolution.

The conversion time is limited by the settling time of the internal DAC. As the ADC resolution increases, the required DAC settling time decreases as well because it must settle to a finer resolution. Errors can be up to  $\frac{1}{2}$  full scale and be nonlinear due to jumped codes for a constant input. SAR converters can reach 20 bit resolution but require longer conversion periods due to the additional number of approximations

required and the DAC settling time. Typically the integrated DAC uses charge-redistribution or a switched capacitor configuration. The digital sample is generally ready after  $n$  comparisons as there is negligible latency in the conversion process.

### **Integration Converter**

Integration conversion is a very popular technique for precision instrumentation when using discrete components, however it has been used less with the advances of integrated circuits. Typically dual integration is used. A reference voltage is integrated, with an integrator op-amp configuration, for a given amount of time measured by a stable clock source and a counter. The integrator input is then switched to the sample voltage and integrated again. This essentially “deintegrates” the voltage back and when it reaches the initial voltage a comparator signals completion. The conversion is then proportional to the sample voltage and does not depend to first order on the capacitor or clock speed. Absolute accuracy is limited by the voltage reference and clock jitter. The resolution is limited by component errors, such as temperature coefficients and offsets, as well as the clock rate. A benefit is that changes in the sample voltage are averaged by the integration process, in particular at integer values of the integration frequency. This is useful for removing 60 Hz content on the sample. The trade-off between conversion rate and resolution is one of the main drawbacks for integration converters, however 18-bit converters are available at lower rates.

### **Sigma-Delta Converter**

Sigma-delta ( $\Sigma-\Delta$ ) converters are a form of an integrator converter and have become more popular with technology improvements in integrated circuits. Their analog electronics are simple compared to other techniques but are replaced by relatively complex operations of oversampling, noise shaping, digital filtering, and decimation on the digital side. The analog electronics are simplified as shown in Figure 1-6. The voltage input is subtracted, or summed depending on the configuration, with a binary

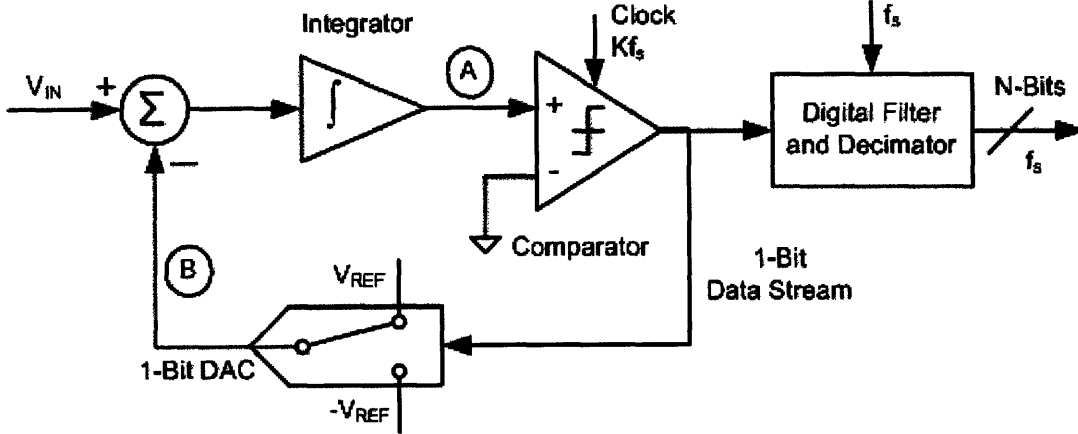


Figure 1-6: Simplified schematic of sigma-delta analog operation, adapted from [1].

feedback signal. This signal is then integrated and compared to a constant voltage. The comparator output is fed back to the input summing junction through a 1-bit DAC, creating the binary feedback. The comparator output is also fed to the digital circuitry as a bit stream, which is proportional to the sample voltage. Because the feedback is a single bit, the settling time is much faster than for SAR feedback.

For example, given a DC input at  $V_{IN}$ , the integrator is constantly ramping up or down at node A. Negative feedback through the single bit forces node B to be equal to  $V_{IN}$  on average. This intuitively means that the average bitwise stream is proportional to  $V_{IN}$ . A bitwise stream of all zeros relates to  $-V_{REF}$  and all ones relates to  $+V_{REF}$ .

In order to understand the digital techniques employed, it is important to understand some fundamental concepts. Quantization is the discretization of a continuous signal. In this case, a sample voltage is quantized to a digital word. The quanta size is synonymous with resolution, which is defined by

$$V_Q = \frac{V_{RANGE}}{2^n} \quad (1.1)$$

For example, a 1 V range with 8 bits has a resolution, or quanta size, of 3.9 mV. When a continuous signal is quantized, there is inherently an error of as large as  $\frac{1}{2}$  the resolution. This can be seen in Figure 1-7 where the quanta size is 0.5. Note that



the zero-order hold time delay is not shown. For a uniformly distributed signal across all quantization levels, the signal-to-noise ratio (SNR) is modeled as [26]

$$\text{SNR} = 6.0206n + 4.77 - 10\log_{10}\eta \text{ [dB]} \quad (1.2)$$

where  $\eta$  is the signal's peak-to-average-power ratio and  $n$  is the number of bits. For a sinusoidal signal  $\eta = 2$  [26],

$$\text{SNR} \approx 1.763 + 6.0206n \text{ [dB]} \quad (1.3)$$

This can be rearranged to solve for the effective number of bits (ENOB) when the SNR, in dB, is measured

$$\text{ENOB} = \frac{\text{SNR} - 1.763}{6.0206} \quad (1.4)$$

This is typically a better figure of merit compared to the specified number of bits for an ADC because it includes signal distortion. The effective number of bits is analogous to numerical precision. The fact that a number can be recorded with a certain precision does not guarantee that the precision is actually  $\frac{1}{2}$  LSB. The ENOB does guarantee precision to this standard.

For a signal that is sampled at a sampling rate of  $f_s$ , the quantization noise power is [27]

$$\sigma^2 = \frac{V_Q^2}{12} \quad (1.5)$$

The quantization noise in this model has a uniform spectrum is then spread from DC to the Nyquist frequency,  $\frac{f_s}{2}$ . When oversampling is implemented by a factor of  $K$ , the the Nyquist band is  $\frac{Kf_s}{2}$ . The simplest model assumes a uniformly distributed zero mean white noise [10]. The claim is also made that conversion values depend on past conversions, so the error is not entirely uniformly distributed [28]. Quantization is inherently a nonlinear process, but the linear models presented above give a simple model for its effects on a signal.

In a sigma-delta converter, the original signal is maintained while removing additional quantization noise by then digitally filtering the oversampled signal back to

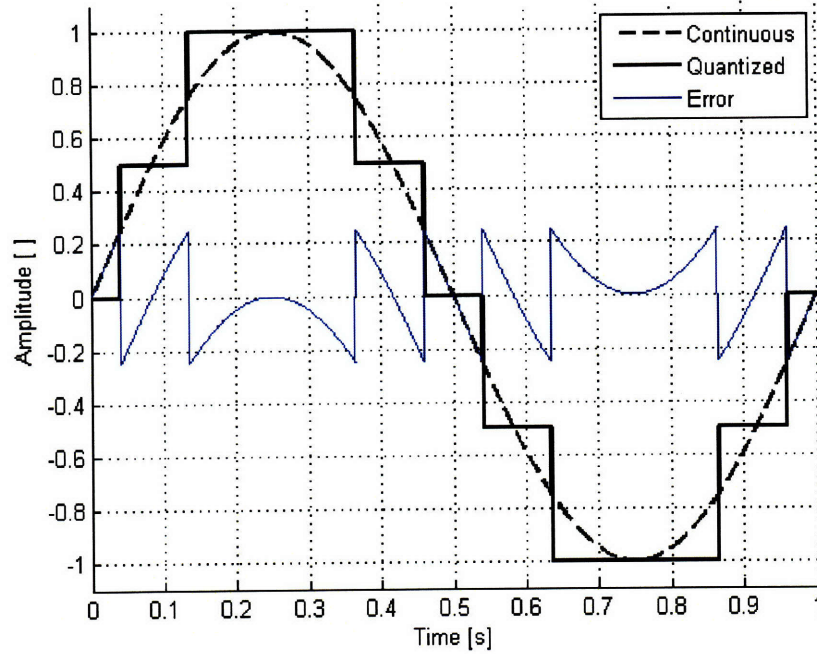


Figure 1-7: Quantized sine wave with  $n = 2$  bits.

$\frac{f_s}{2}$ . The relationship between the oversampling ratio  $K$  and the increased effective resolution  $n'$  due to removed noise spectra is [29]

$$K = 2^{2n'} \quad (1.6)$$

Another step that can be included before digitally filtering is noise shaping. This is the process of shaping the quantization noise so it lies above the passband of the digital output filter. This is completed in the analog electronics before the digital filtering. From the simple, first-order model shown in Figure 1-6, the bitwise stream is proportional to the signal. Figure 1-8 shows a simplified block diagram for the analog sigma-delta modulator adapted from [30]. The integrator is replaced by a theoretically ideal integrator with the Laplace variable  $s$ . In Figure 1-8,  $X$  is the input signal,  $Y$  is the single bit feedback, and  $Q$  is the quantization noise. By inspection, the block diagram is

$$Y = \frac{1}{s}(X - Y) + Q \quad (1.7)$$

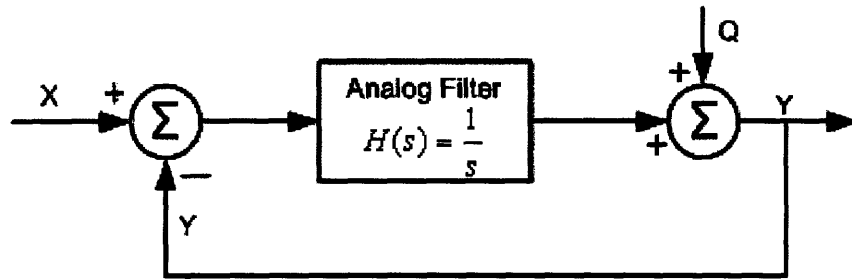


Figure 1-8: Block diagram of sigma-delta modulator, adapted from [1].

Solving for  $Y$  gives

$$Y = \frac{X}{s+1} + \frac{sQ}{s+1} \quad (1.8)$$

By taking the frequency  $f$  limits, where  $s = 2\pi fi$ , the distribution of signal and noise versus frequency is shown. As the frequency  $f \rightarrow 0$ , the output  $Y$  approaches the input signal  $X$  and the quantization noise is not present. As the frequency  $f \rightarrow \infty$ , the output  $Y$  approaches the quantization noise and the signal  $X$  is not present. Intuitively this is reasonable because the integrator acts as a low-pass average of the DC, non-zero input  $X$  and as a high-pass filter on the noise.

Just as higher orders of integration increase attenuation, they also increase their noise shaping effect. The additional orders are obtained by adding another integrator and summation block as shown in Figure 1-9, adapted from [1, 31]. The noise is then shaped as in Figure 1-10. With these results, one would then intuitively increase the integrator order until the noise is effectively eliminated in the band of interest. However, the system would then become unstable assuming an infinite gain comparator. When assuming finite gain comparators though, instability is not guaranteed. By properly monitoring the bitwise stream in the digital electronics, incipient instability can be detected and prevented [30, 32, 33]. These details in which the converter can exhibit self-excited oscillations are another level of complexity. Commercial ADCs are available with as high as fourth-order sigma-delta modulators.

The digital filtering is not as simple as a low-pass filter shown in simplified descriptions. A finite impulse response (FIR) filter is digitally implemented and, as the

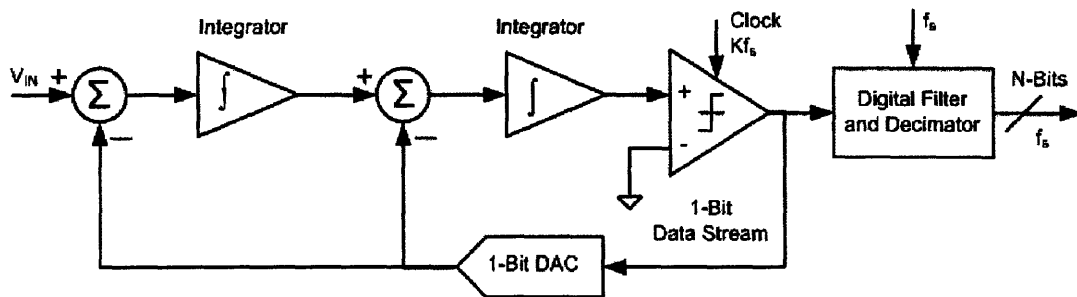


Figure 1-9: Second-order sigma-delta modulation, adapted from [1].

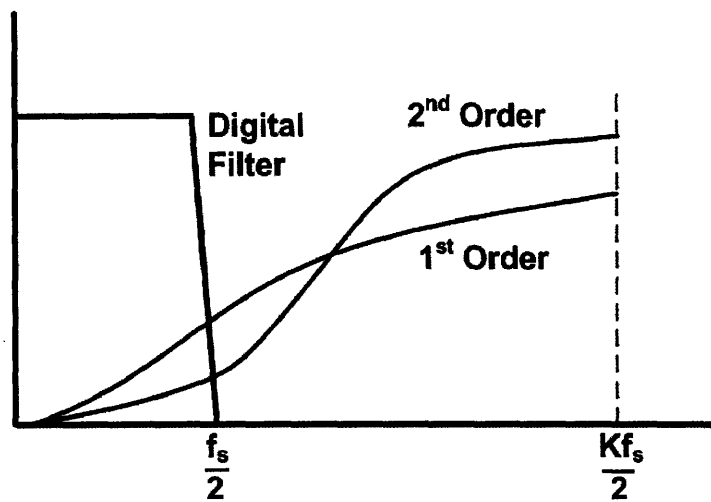


Figure 1-10: Sigma-delta modulation noise distribution.

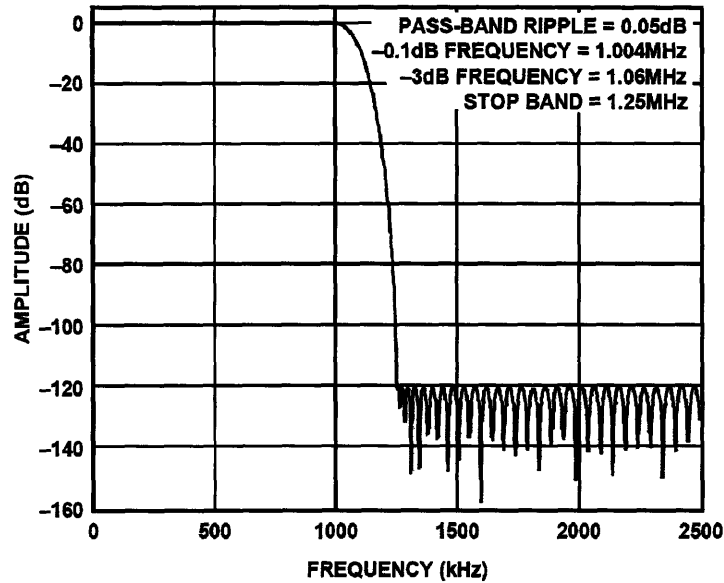


Figure 1-11: Sigma-delta FIR filter response [2].

name implies, has a finite response to an impulse. As opposed to an infinite impulse response (IIR) filter, which has internal feedback and can potentially respond indefinitely to a transient, an  $N$ th order FIR filter has a response that is  $N + 1$  samples long. This FIR filter can be described to have a flat passband with relatively sharp cutoff. A large portion of the group delay is due to the phase delay of the FIR filter [34]. For the ADC implemented in Chapter 4, the FIR filter response is shown in Figure 1-11. The group delay could be decreased by using a smaller order filter.

Finally the output signal is decimated (downsampled). High sampling rates are used to minimize noise content; however after digital filtering these additional samples provide no more information on the original signal. Therefore decimation by a factor of  $M$  could be accomplished by passing every  $M$ th result and dropping the rest. Conceivably these samples could be averaged together before passing on. Assuming the noise is a Gaussian distributed random signal in the  $M$  interval, the resolution would be increased by an additional factor of

$$\frac{1}{\sqrt{M}} \tag{1.9}$$

It is important that the output data rate after decimation is at least twice the signal bandwidth so that signal information is not lost.

Due to the single bit architecture of the ADC, the output is inherently very linear and errors are within a small window as opposed to the potential missing codes of other ADC techniques. Sigma-delta converters generally operate with the best resolutions at the highest rates, and exhibit a correspondingly high SNR within this performance area. These converters are capable of high data rates but are limited in real-time control due to the propagation delay.

### **Hybrid A/D Converter**

A multitude of possibilities exist for combining different A/D converting technologies in an effort to balance hardware, resolution, and conversion speed. One method is to have multiple flash converters for different ranges of bits. These are sometimes referred to as subranging or half-flash ADCs. For a 16-bit example, the 8 most significant bits (MSB) can be flash converted and the digital word can be fed back through a D/A converter and subtracted from the original analog value. The 8 least significant bits (LSB) are then flash converted and the bits are combined [35]. Another method would be to use successive approximation on the MSB and flash conversion on the LSB.

While requiring a number of comparators on the order with flash converter, a successive approximation converter can be produced from a chain of  $2^n - 1$  resistors and analog switches to track the sample. These resistors and switches replace the DAC of a typical SAR converter and does not have the discontinuous errors typically seen in SAR converters. However, the conversion rate for a new sample depends on the change in the sample voltage from the previous conversion.

Another SAR converter configuration uses several stages with multiple ADC modules. This creates a pipelined design. This increases the resolution with modest additional hardware, however it also introduces a pipeline latency delay.

### 1.2.3 Digital-to-Analog Converting Methods

A digital-to-analog converter generates an analog output based on a digital word input. Typically the voltage output is created by summing voltages representing a digital bit. The voltage can be created with a resistor or capacitor network and by switching either a reference voltage or current to the proper input terminals of the network as a function of the digital input.

The simplest method is a weighted-resistor network which uses a single, constant voltage across a resistor network. A 3-bit converter is shown in Figure 1-12. The resistor values increase to  $2^n$  for an  $n$ -bit converter. The tolerances required for the two extremes of resistor values limit the resolution. The voltage across these parallel resistors creates a current that is then dropped across a load resistor, thus generating an output voltage.

An improved design is a weighted R-2R resistor network. The currents associated with each digital bit are created across repeated stages of  $2R$  and  $R$  resistors. Two configurations of a 3-bit converter is shown in Figure 1-12. The first method creates a voltage output and the second produces a current output, which can be converted to a voltage across an output op-amp. The accuracy is dependent on the resistor matching as well as load resistance. The resistors in the first method act as current dividers and the current sum is dropped across a load resistor to create the output voltage. Without this output resistor, a current output is possible. The output buffer stage is typically more advanced than a single resistor, but rather an op-amp to drive the output which is generally the slowest part of the converter. Some high-speed D/A converters use a current output with a high-speed external op-amp to drive a voltage output.

Hybrid combinations are common for high-precision converters, particularly with separate MSB and LSB stages. For example, the AD768 converter from Analog Devices uses current sources for the MSB portion and an R-2R ladder for the LSB portion and the two outputs are summed together. Accuracy can be maintained by designing the circuit on a single monolithic IC and laser trimming components during

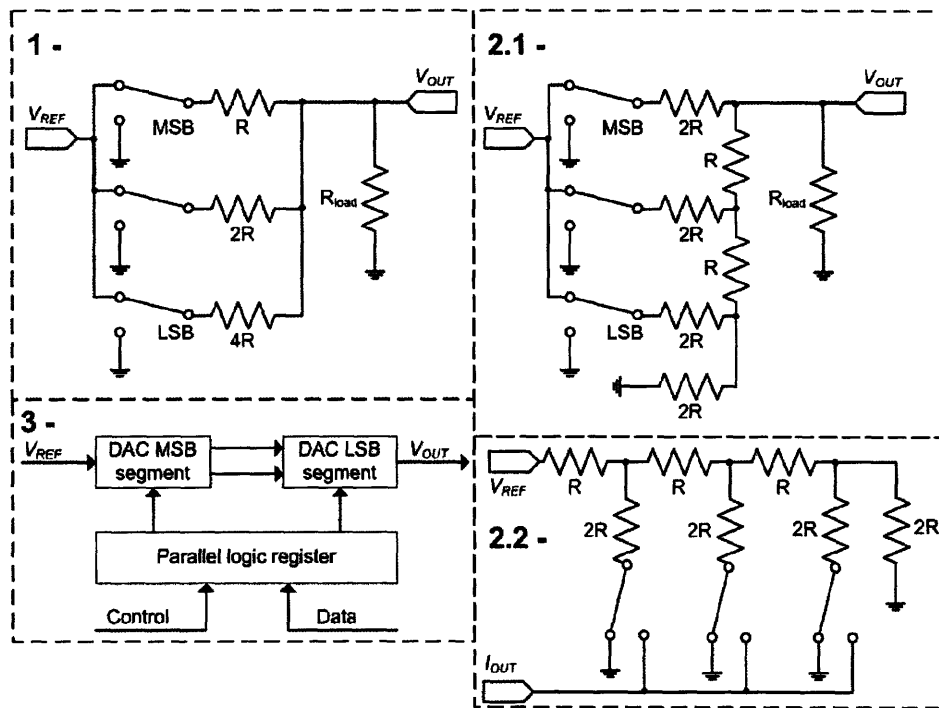


Figure 1-12: D/A converter hardware architectures. 1 - Weighted-resistor network. 2.1 - Voltage output R-2R resistor network. 2.2 - Current output R-2R resistor network. The output node can be terminated at a virtual ground of an op-amp to form a current-to-voltage converter. 3 - Segmented implementation



testing before final packaging. Essentially all D/A converters are a combination of either current or voltage sources with current/voltage steering or current/voltage dividers.

D/A converter resolution can range from 6 to 18 bits with a settling times from 15 ns to 100  $\mu$ s [27, 35]. The settling time is defined as the time it takes to settle to the specified accuracy. The update rate can be much higher however, and is limited by the rate at which digital logic can be clocked in the D/A and in the internal digital switching components. A fast settling time is essential to maintaining high-resolution when high update rates are necessary. Faster update rates are possible with parallel digital inputs. This removes a serial to parallel register within the D/A converter as well as the need to clock in up to 16 bits, generally with a maximum clocking frequency of 40 MHz. The parallel inputs also add hardware and necessary logic lines, increasing the cost, physical footprint, and required logic lines from the controlling circuit.

High-speed, high-resolution D/A converters are specified with their resolution, update rate, settling time, and input format. The specific internal D/A conversion implementation has little effect on additional specifications.

This chapter has presented an overview of the work completed in this thesis, as well as a background on high-resolution digital systems. These system require an understanding of the computation/data manipulation platform and the analog interfaces, both input and output, to the digital system. These architectures impact system performance. Chapter 2 discusses high-resolution analog interfaces designed, built, and tested for a dSPACE digital system. The remainder of the thesis describes a high-resolution, high-speed digital platform designed, built, and applied to an experimental plant.



# Chapter 2

## dSPACE Interoperable

## High-Resolution Analog Interface

This chapter describes high-resolution hardware and software designed to operate with a dSPACE control environment. The hardware consists of a custom A/D PCB, D/A PCB, and connector breakout PCB. The software consists of embedded firmware on the A/D PCB and interface software for the dSPACE system. The hardware and software was designed by David Otten as a research scientist with the Precision Motion Control Laboratory in cooperation with the SAMM stage design at the University of North Carolina in Charlotte [16]. The design is considered freely available to the public<sup>1</sup>. This chapter describes the overall design as well as characterization and improvements made to the design.

The high-resolution A/D and D/A system utilize a slave DSP on the dSPACE 1103 PPC controller board to transfer data. The custom PCBs are shown in Figure 2-1. The system provides galvanic isolation between dSPACE and the plant to disrupt ground loop effects. A small footprint permits the input/output to be located near the signal source/sink to increase signal fidelity. The A/D channel uses an on-board DSP and a high-speed A/D converter for oversampling and averaging to increase the effective number of bits from 18 to 20 bits compared to a maximum of 16 bits with dSPACE. The D/A channel uses a precision 16-bit converter as opposed to the

---

<sup>1</sup>Available at [dSPACE.mit.edu](http://dSPACE.mit.edu)

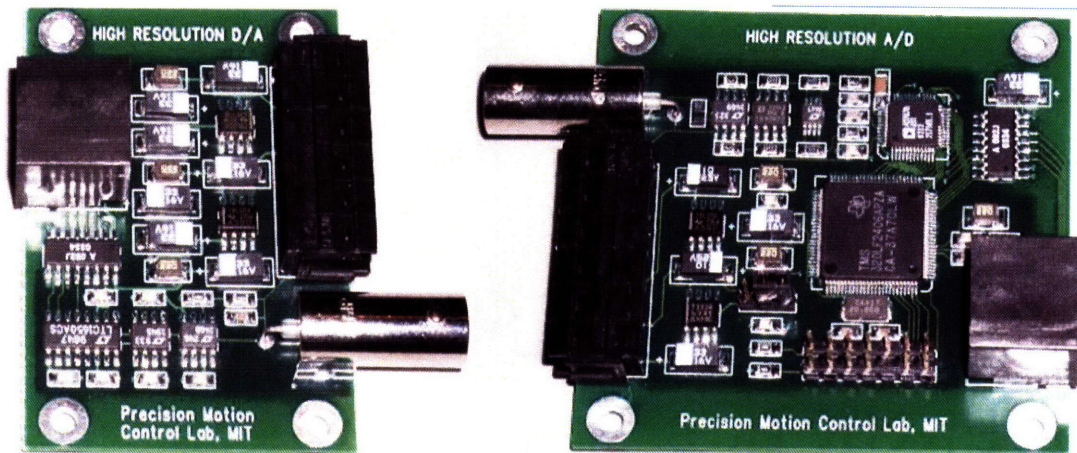


Figure 2-1: dSPACE high-resolution DAC (Left) and ADC (Right) PCB.

dSPACE 14-bit converter. Up to 8 A/D and 7 D/A channels are run to a dSPACE interface board through digital modular cabling. The increased resolution gained by oversampling and averaging introduces additional phase lag of 0.8 samples and a magnitude decrease as the signal frequency approaches the Nyquist frequency. The sampling frequency is limited to 8kHz due to the slave DSP communication processes.

## 2.1 System Description and Design

The dSPACE 1103 PPC controller board (DS1103) uses a PowerPC 604e running at 400 MHz with a slave Texas Instruments DSP running at 20 MHz. The DS1103 supplies 16-bit multiplexed ADCs with  $\pm 10$  V range and 80 dB signal-to-noise ratio (SNR), as well as eight 14-bit DACs with  $\pm 10$  V range. The high-resolution PCBs interface with the slave DSP to take advantage of its processing power, allowing time consuming arithmetic to be completed by the DS1103 as opposed to on-board the slower and less precise A/D DSP. A serial interface is used to facilitate galvanic isolation with high-speed digital isolators. This isolates the plant electronics from the dSPACE system in an effort to eliminate ground loops as well as allowing the input/output boards to be powered with the same power source as the plant. Signal fidelity can be maintained by locating the PCB boards near the signal source/sink

and then digitally interfacing the boards to a central dSPACE interface board.

Oversampling and averaging improves the resolution of a signal that is corrupted by white noise. In our implementation at most 100 samples can be averaged per cycle with the A/D converter operating at 800 kSPS and the slave DSP at 8 kHz. The noise is assumed to have a Gaussian distribution, and thus a sum of  $N$  equally weighted samples divided by  $N$  reduces the standard deviation by  $\frac{1}{\sqrt{N}}$ . Therefore the expected improvement is a factor of 10, or 3.3 effective bits. This also introduces an additional  $\frac{T_{AD \text{ SAMPLE}}}{2}$  time delay, where  $T_{AD \text{ SAMPLE}} = 1.25 \mu s$ , into the feedback loop. The frequency response of the averaging filter to an impulse over the sampling period is given by [36]

$$H(e^{j\omega}) = \frac{1}{2M+1} \sum_{k=-M}^M e^{-j\omega k} \quad (2.1)$$

where  $M = \frac{N}{2}$ . The magnitude of this frequency response is shown in Figure 2-2. The width of the first lobe is  $\frac{2\pi}{N}$ , or approximately 50.3kHz for 800 kSPS. The magnitude decrease at higher frequencies is intuitively understood by realizing the sample does not represent the precise signal at that instant; rather it represents an average of the signal since the last acquisition. At low frequencies there are many samples taken for each sine wave so the magnitude decrease at the peaks is negligible, but at higher frequencies where the signal is represented by only several samples, peaks become rounded off showing a noticeable magnitude decrease. The discrete z-domain transfer function is

$$H(z) = \frac{1 + z^{-1} + \dots + z^{-98} + z^{-99}}{100} = \frac{z^{99} + z^{98} + \dots + z^1 + 1}{100z^{99}} \quad (2.2)$$

where  $z$  is the z-transformation variable. This digital filter is a boxcar finite impulse response (FIR) filter. Equation 2.2 does not imply decimation however. Decimation can be a distinctly separate process of taking every  $N$ th point. By combining the boxcar filter with the decimation, the full sample data can be utilized for higher resolution at a slower, decimated data rate. The boxcar filter and decimation is discussed further in Section 5.1.

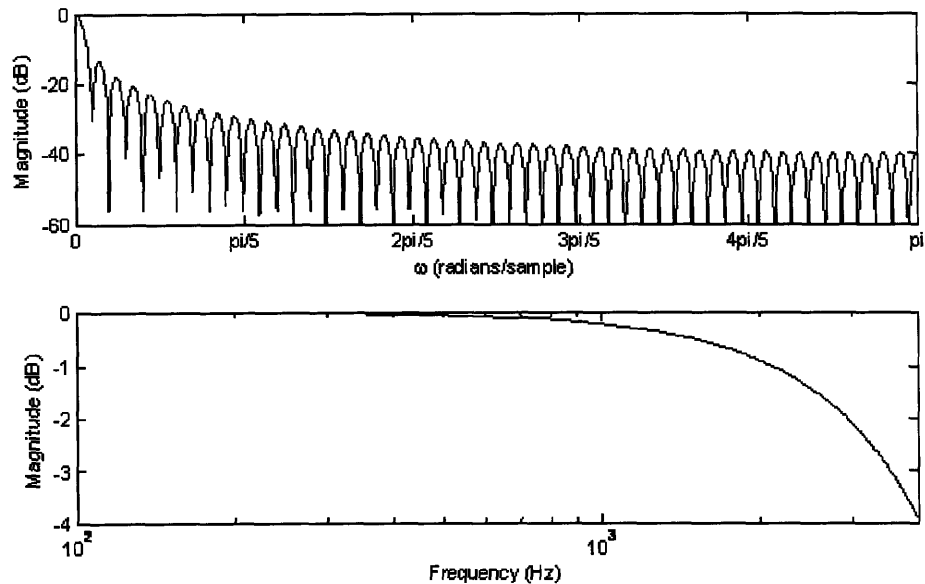


Figure 2-2: Magnitude frequency response of averaging filter for  $N = 100$  at 800 kSPS, before decimation.

The high-resolution A/D PCB designed by David Otten is built around a commercial, high-speed 18-bit A/D converter IC (AD7674) from Analog Devices which is coupled with a 16-bit fixed-point digital signal processor (DSP) IC (TMS320LF2604) commercially available from Texas Instruments. The A/D converter operates at 800k samples per second (SPS) while the DSP sums the samples and formats the data for serial transmission to the dSPACE slave DSP. The average is not calculated on the DSP because the fixed-point precision limits accuracy.

The A/D converter is provided with a fully differential input front-end that can accept up to  $\pm 10$  V signals. The circuit is built on a small printed circuit board (2.5" x 2.6") and located in an aluminum case that can be matched to any available electrical common. An 8-pin modular connector is used for the high-speed digital signals between the analog input board and dSPACE. The design uses readily available ethernet cables for cabling because their twisted pair construction and controlled impedance is favorable and they are available pre-terminated in a variety of lengths.

As a companion to the high-resolution A/D, a high-resolution D/A was also designed and built by David Otten. A 16-bit low glitch voltage output D/A from Linear

Technology (LTC1650) is used. The circuit is also built on a small printed circuit board that can be located close to the destination of the signal and utilizes an isolated interface to the DS1103 to eliminate ground loop problems. The data for the D/A is serially shifted out of the DS1103 slave DSP at the same time the A/D data is shifted in. This overlap efficiently minimizes serial clock signals for transmitting data to and from the DS1103.

The D/A IC itself only outputs a  $\pm 4.5$  V signal. The output of the PCB is driven with a non-inverting op-amp configuration to scale the voltage to  $\pm 10$  V. The resistor ratio is

$$k = 1 + \frac{R_2}{R_1} = 1 + \frac{110k}{90.9k} = 2.210 \quad (2.3)$$

where  $R_1$  is between the inverting terminal and ground and  $R_2$  is in the feedback path. Because the gain factor  $k$  is not exactly 2.222, the output range is limited to  $\pm 9.946$  V. Hence a scale factor is included in the slave DSP to account for this difference. The results incorporate this scale factor, however it is essential to check this factor for any new setup as resistor tolerance or reference accuracies can vary between PCBs.

The relevant internal components of the DS1103 relating to the high-resolution system are outlined in Figure 2-3. The digital I/O are used to clock and transmit data through the interface PCB to all channels. The DS1103 slave DSP reads from/to the 32-bit I/O bus and the master processing unit performs the complex data manipulation. Along with the high-resolution boards, 2 interfaces for Zygo ZMI 4004 interferometer boards are included in David Otten's design, allowing up to 8 measurement channels. This additional design implementation is not described herein but is documented in the available resources [4, 16].

Only 7 output D/A channels are available because the output data port only provides 7 bits. The software and DS1103 board itself uses the full 8 bits of the port but the DS1103 cabling and breakout box only brings out 7 bits. It is possible to use another port however the data transfer on the additional port limits the closed-loop sample rate to approximately 6 kHz as opposed to 8 kHz. Cabling is provided for this

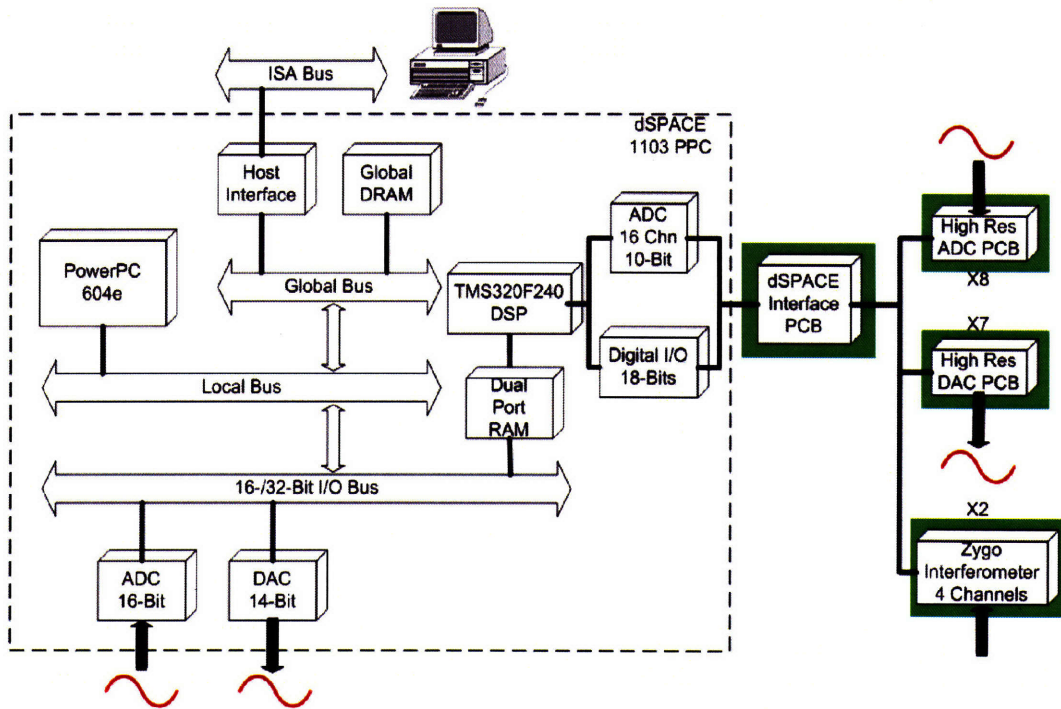


Figure 2-3: dSPACE data flow with high-resolution platform, adapted from [3].

extra channel on the PCB interface board, and the two sets of compiled software files are available for the instance when all 8 D/A channels are required, albeit at a slower sampling rate. Initially only 6 channels were operational but during characterization the breakout connector PCB was rerouted for the 7th channel.

The software interface is implemented with three programs: a custom written Simulink S-Function, a set of user-defined functions for the slave DSP on the DS1103 PPC controller, and a custom firmware program to control the A/D DSP. This DSP requires a one-time program write with specific program emulation hardware from TI. A user-defined Matlab Simulink model is generated for each specific application where the S-Function provides a user interface with 8 inputs and 7 outputs in the Simulink environment.

The S-Function generates a communication path between the DS1103 master and slave DSP. This includes data formatting, scaling, sorting, and interfacing to the 16-bit communication buffer. The slave DSP interfaces with the communication buffer



and then reformats the data from 16-bit words back into 32-bit words. One bit is sent to each D/A channel and one bit is received from each A/D channel on each clock pulse. The D/A channels require 16 bits for an output, and the A/D channels require 32 bits to be acquired. Of the 32 bits there are 25 bits of data and 7 bits representing the number of samples taken. For each instance where data is requested by the slave DSP from the A/D DSP, 32 bits are transferred to the A/D DSP serial output buffer and a new sum is started. This allows as many A/D samples  $N$  per period as possible.

The sampling rate of dSPACE is limited by the time it takes to transfer and process data between the components of Figure 2-3. Table 2.1 details the functions within one Simulink time step. The majority of time is spent by the slave DSP transferring and clocking data, as shown with Figure 2-4. This process is straight line coded in assembly language to minimize the delay time. The complete step time subsequently limits the dSPACE sampling rate to 8 kHz. The relative time at which each channel is latched to the input or output is shown over 2 time samples in Figure 2-5. The dSPACE channels are represented with DS and the custom high-resolution channels are represented by HR. The timescale is given in  $\mu\text{s}$ .

## 2.2 System Characterization and Results

All digital systems create a phase delay, or time lag, due to the processing and transport of digital signals. This requires at least one time step, as it takes time for the data to transfer from the Simulink model to the D/A converters and then time for the A/D converters to sample the data and send it back to the Simulink model. Figure 2-6 shows various time delays for different combinations of dSPACE (DS) and high-resolution (HR) systems. There is a single time sample delay from the DS D/A output to the DS A/D input relative to an internally generated sine wave. The HR D/A to the DS A/D is not shown here, since it has the same phase lag because both D/A channels are outputted before the subsequent DS A/D reading. This was also shown in Figure 2-5. The DS D/A to HR A/D has a larger phase lag due to the

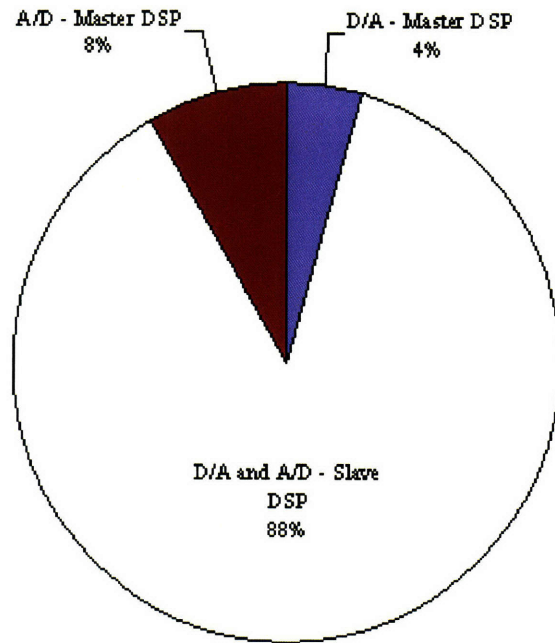


Figure 2-4: dSPACE subsystem timing [4].

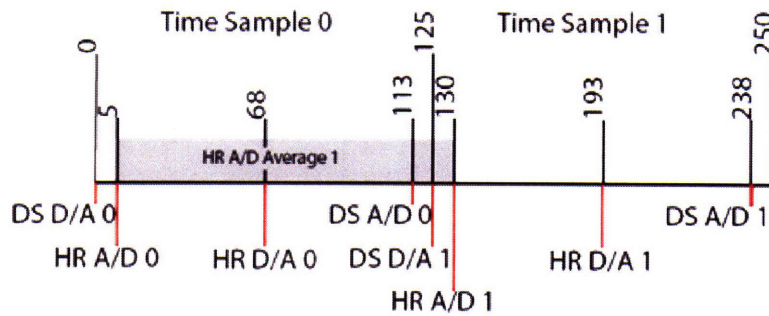


Figure 2-5: dSPACE and high-resolution latching timeline at 8 kHz sampling. The dSPACE channels are represented by DS and the high-resolution are represented by HR. The time scale is microseconds.

Table 2.1: dSPACE Subsystem Timing

<b>Master DSP - mdlUpdate Function</b>	<b>Time [<math>\mu</math>s]</b>	<b>Total [<math>\mu</math>s]</b>
Flush communication buffer	0.96	
Acquire D/A data from model and scale	0.68	
Sort channels	0.13	
Reformat data	1.31	
Output D/A data to comm. buffer	1.79	<b>4.87</b>
<b>Slave DSP</b>		
Acquire D/A data from comm. buffer	14.94	
Clock data to/from analog hardware	48.25	
Output A/D data to comm. buffer	35.35	<b>98.54</b>
<b>Master DSP - mdlOutput Function</b>		
Acquire A/D data from comm. buffer	4.07	
Reformat data	2.92	
Sort Channels	0.13	
Scale A/D data and output to model	2.42	<b>9.54</b>
<b>Complete Cycle Time:</b>	<b>112.95</b>	

averaging from the previous time sample. The HR D/A to HR A/D delay is over 2 time samples because for a given sample output, as shown in Figure 2-5, the A/D input is latched in before the output is latched out. This is in addition to the A/D averaging over the preceding time sample.

These delays can also be viewed in the frequency domain. Figure 2-7 shows the frequency responses of the four possible system combinations. The top magnitude and middle phase frequency responses were obtained with a digital dynamic signal analyzer designed for Simulink and the dSPACE family by Katherine Lilienkamp [37]. The bottom plot was measured with an external DSA. These two methods are further discussed below. For the DS D/A and HR D/A to DS A/D there is a 45 degree phase shift at 1 kHz, corresponding to a single sample delay, as demonstrated by Figure 2-6. The transfer function for this pure time delay is

$$G = e^{-T\omega j} \quad (2.4)$$

where  $T = 125 \mu$ s is the sample time. As previously described with Equations 2.1 and

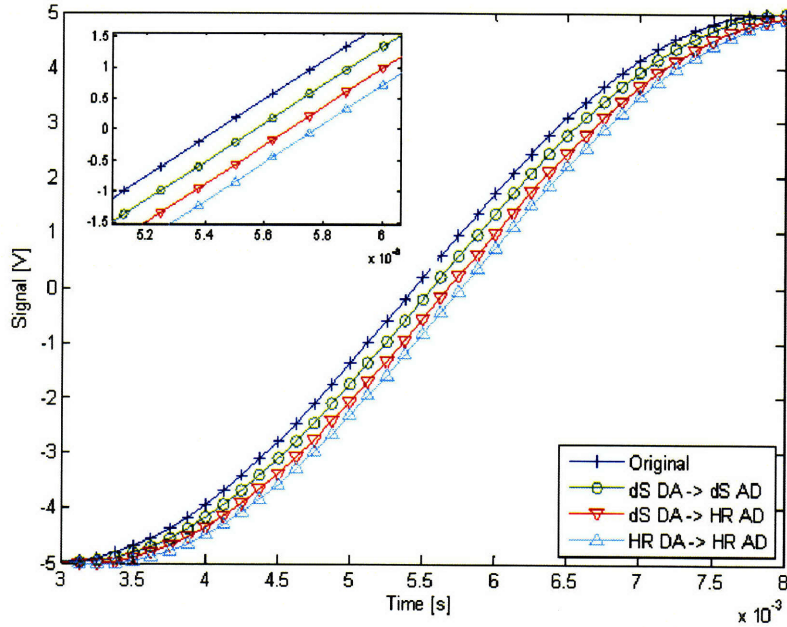


Figure 2-6: Phase delay of 100 Hz sine wave for several connection options. The time axis is in milliseconds. Digital system sample rate is 8 kHz.

2.2, the HR A/D introduces a magnitude and phase decrease. The magnitude decrease is expected to be 4 dB at the Nyquist frequency of 4 kHz and the phase delay is expected to be 1.5 samples, or  $188 \mu\text{s}$ . The experimental phase delay has a longer time constant of  $T = 225 \mu\text{s}$ , corresponding to a 1.8 sample delay and a magnitude decrease on the same order as that expected. This experimental difference is attributed to HR A/D latching occurring well into the time step and is thus averaging a changing D/A signal. The combination of both high-resolution systems introduces further dynamic effects due to the data clocking and latching order shown in Figure 2-5. The D/A data is latched out to the analog domain almost half a time sample after the A/D sample is latched in. Additionally, the A/D sample is averaged over the previous time sample. This introduces the larger phase lag of 2.5 samples, or  $T = 315 \mu\text{s}$ .

The increased phase lag when measuring with the high-resolution A/D from the high-resolution D/A also corresponds to a larger magnitude decrease due to the high-speed sample averaging, particularly as the sampling frequency approaches Nyquist.

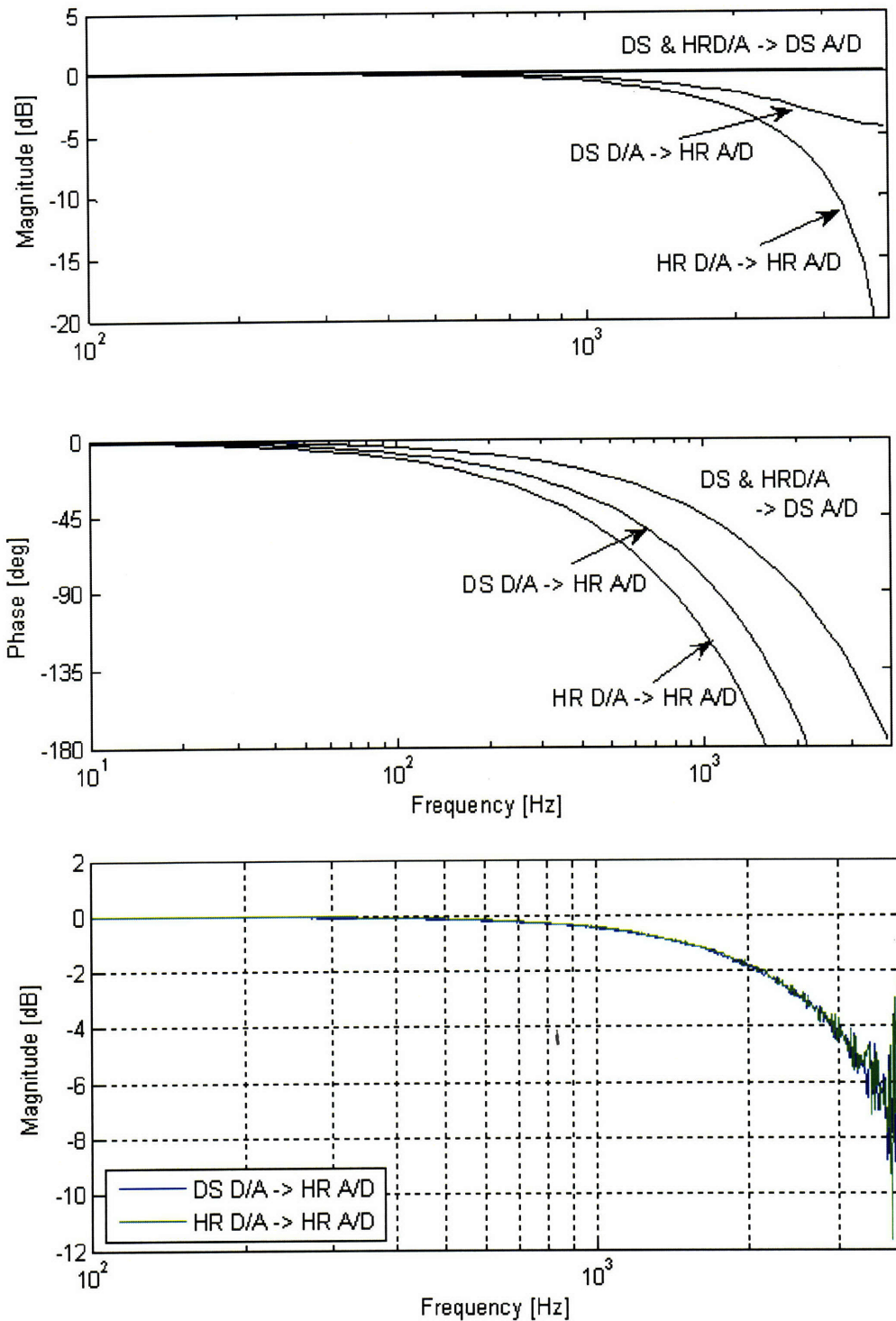


Figure 2-7: Frequency response comparison, up to the Nyquist frequency. The frequency response as measured with the dSPACE DSA for both magnitude (top) and phase (middle) and the magnitude frequency response as measured with the HP DSA (bottom).

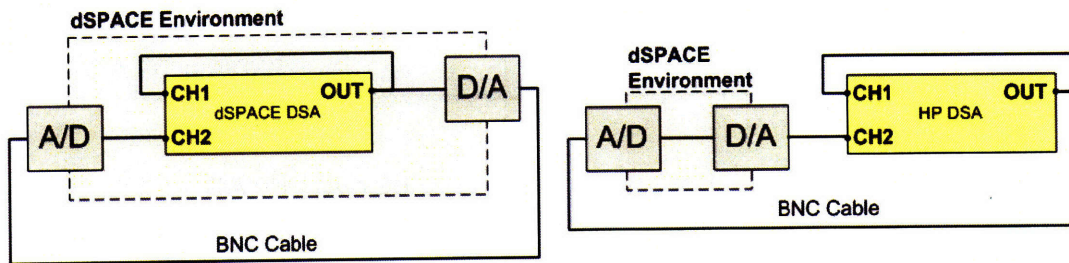


Figure 2-8: Dynamic signal analyzer configurations for dSPACE software DSA (left) and hardware HP DSA (right).

The clocking order was designed to optimize the overall DSP processing time, thus constraining the A/D and D/A latching sequences. The  $>20$  dB magnitude attenuation at Nyquist is merely an artifact of the frequency response measurement method however. An alternative method to the dSPACE dynamic signal analyzer (DSA) designed by Katherine Lilienkamp for measuring the frequency response is to use an external DSA (HP model 35665A). The two configurations are shown in Figure 2-8. The bottom plot in Figure 2-7 shows the magnitude frequency response when measuring with the high-resolution A/D. As expected, the magnitude decrease due to sample averaging is the same regardless of signal source. The external DSA better simulates the hardware as it would be used in a system.

While the high-resolution A/D design introduces magnitude and phase losses due to the digital averaging, the effective resolution of the analog input channels between the dSPACE channels and the high-resolution channels are increased from 15.7 to 20.1 bits. In analyzing analog-to-digital converter results, peak-peak and RMS amplitudes are used as well as the signal-to-noise ratio (SNR) and the effective number of bits (ENOB). Technically the noise includes all distortion in the signal up to the Nyquist frequency, excluding dc. The SNR is computed by

$$SNR = 20 \log \left( \frac{20 [V]}{RMS_{NOISE} [V]} \right) [dB] \quad (2.5)$$

where the  $RMS_{ERROR}$  is measured about the average of the total samples for a constant input and the range is 20 volts.

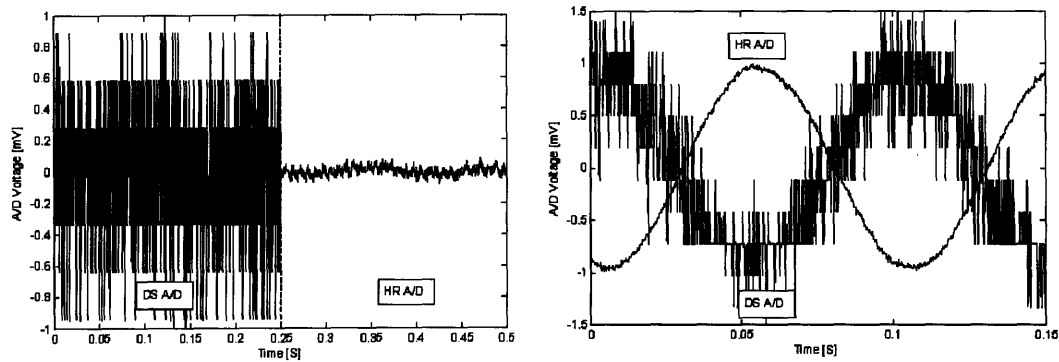


Figure 2-9: A/D noise floor (Left) and A/D response to 10 Hz sine wave (Right).

To quantitatively demonstrate the high-resolution improvement, the A/D channels were initially tested with 50  $\Omega$  BNC terminators. The terminator shorts the differential inputs to create a theoretically zero input. Figure 2-9 shows a sample data capture with grounded inputs. Results are also listed in Table 2.2. Most of the dSPACE A/D noise is limited to  $\pm 1$  LSB, or  $\pm 305 \mu\text{V}$ . These quanta divisions, and the lack thereof in the high-resolution A/D system, are further shown in response to a 1 mV, 10 Hz sine wave which was resistively attenuated by 60 dB to 1 mV amplitude.

Table 2.2: A/D Noise Floor Comparison

	dSPACE A/D	HR A/D
<b>Pk-Pk</b> [ $\mu\text{V}$ ]	2136.2	809.0
<b>RMS</b> [ $\mu\text{V}$ ]	302.3	15.0
<b>SNR</b> [dB]	96.4	122.5
<b>ENOB</b> [bits]	15.7	20.1

With an understanding of the A/D dynamics and basic noise floor, I implemented static tests to demonstrate linearity, offset, and RMS noise over the full output range. A schematic of the test equipment is shown in Figure 2-10. A high-resolution D/A board generates an expected voltage output that is low-passed at approximately 0.3 Hz with a high-performance polypropylene 0.47  $\mu\text{F}$  capacitor. This signal is then measured with an A/D board under test, by a Keithley 2700 6.5 digit DMM, and by a Tektronix AM502 differential amplifier. The A/D board under test samples

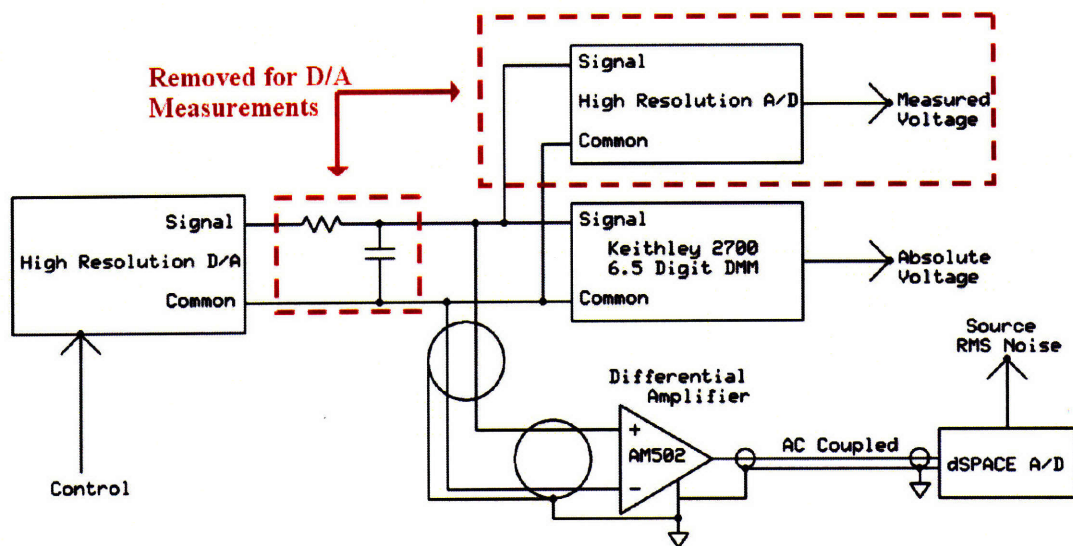


Figure 2-10: High-resolution characterization schematic for A/D and D/A channels.

over a given period whereupon the average is calculated to give the measured voltage and the RMS noise is calculated about this average. The DMM is operated with extended conversions to produce low variation results, giving an assumed absolute voltage. The differential amplifier is AC-coupled to a dSPACE A/D and is used to measure the RMS noise present in the signal. A large gain is used on the differential amplifier to avoid quantization due to the dSPACE A/D converters. The signal RMS noise is also measured to ensure it is significantly below the A/D measured RMS noise. This same setup is used to characterize the high-resolution D/A systems, with several items removed as shown.

The high-resolution A/D converter is a commercially available IC from Analog Devices (part number AD7678) and is an 18-bit switched capacitor successive approximation register (SAR) converter. The converter operates at its maximum conversion rate of 800 kHz and has fully differential inputs with a fixed gain front-end instrumentation amplifier configuration. Figure 2-11 shows the RMS noise levels for the source signal, the dSPACE A/D, and the high-resolution A/D. The noise present on the signal line is included to verify that the most significant noise source is the A/D itself. The dSPACE A/D achieves 15.4 effective bits over the full range. As



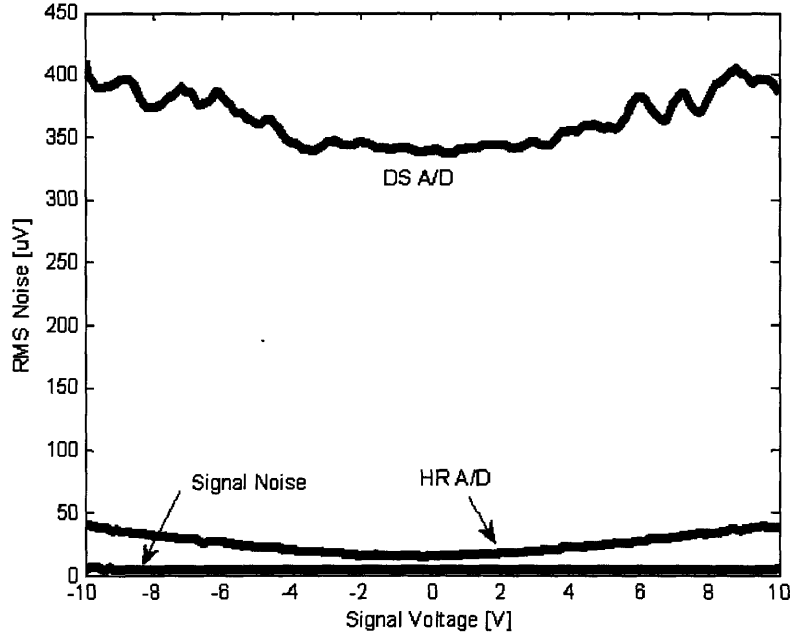


Figure 2-11: RMS noise on A/D channels across full input range.

previously shown, the HR A/D channel has an effective resolution of over 20 bits, but this resolution is only achieved for an input signal close to 0 V. The noise grows with the signal level. The effect is likely due to the SAR converter relying on an external voltage reference for the internal DAC. This DAC introduces a symmetrical noise source from the voltage reference which is further explained with the high-resolution D/A results.

When this design was first adopted and throughout characterization, the A/D design displayed an asymmetric RMS baseline noise distribution. This asymmetric response is shown in Figure 2-12. The debugging process focused on the analog portion of the circuit leading into the A/D converter. The converter datasheet recommends the configuration on the left of Figure 2-13 where shunt capacitors low-pass the differential signals to common. The poor noise was improved by changing to the configuration shown in the right of Figure 2-13 where a single capacitor is referenced to the differential signal. It is not clear why higher RMS noise levels are measured with positive versus negative input voltages. There is no commonly accepted method as the literature recommends both methods [2, 8]. The important frame of reference

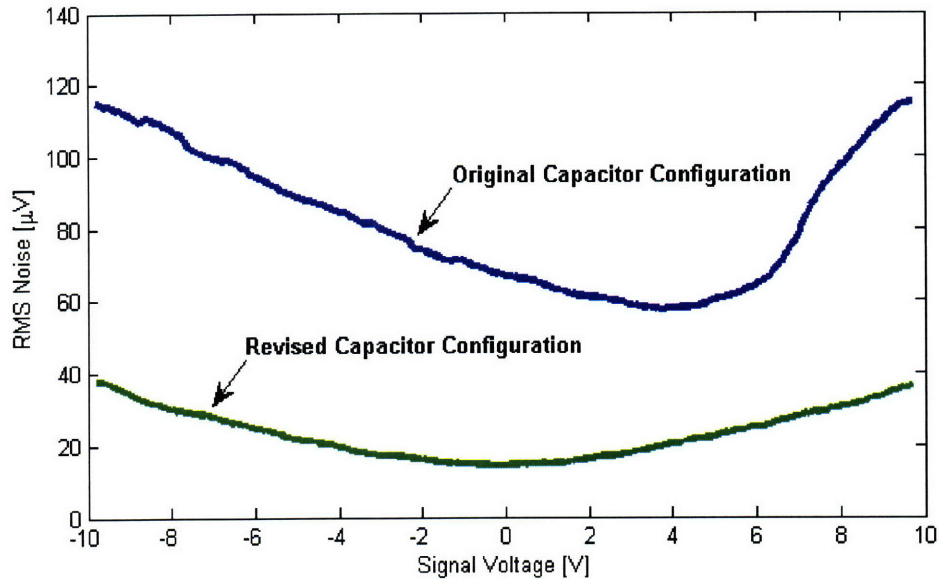


Figure 2-12: Asymmetric RMS noise on high-resolution A/D channel across full input range. The original configuration has two capacitors references to common and the revised configuration has a single capacitor referenced between the differential signals.

is the relative signal dynamics because that is the converted voltage as opposed to the absolute dynamics relative to the ground reference. This improved configuration is recommended by the differential amplifier datasheet.

The high-resolution D/A board utilizes a D/A converter with higher resolution than available in the DS1103. The dSPACE D/As implement a sample and hold that limits the glitch conversion effect common to D/A converters. There is a finite time where the output impulses when the D/A converter modifies its output. Figure

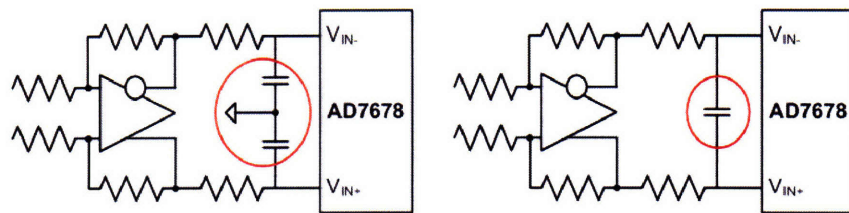


Figure 2-13: High-resolution analog front-end configurations: asymmetric high-noise configuration (left) and symmetric low-noise configuration (right).

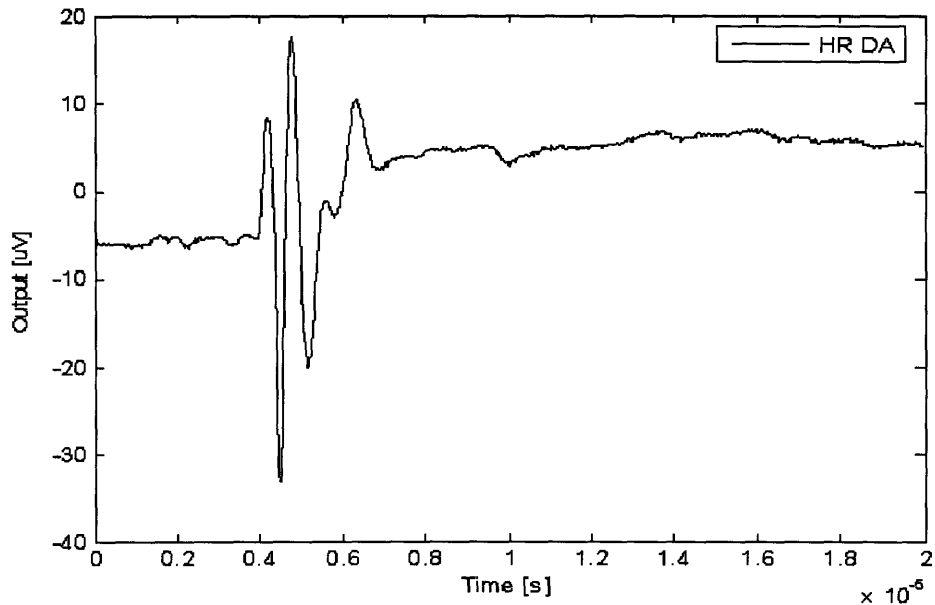


Figure 2-14: High-resolution D/A glitch.

2-14 demonstrates a glitch from the designed D/A channel after passing through the final gain. The peak-to-peak amplitude is  $50.8 \mu\text{V}$  over a  $2.5 \mu\text{s}$  span. The glitch energy shown is  $0.127 \text{ nV}\cdot\text{s}$ , compared to the  $1.8 \text{ nV}\cdot\text{s}$  specification in the datasheet. A sample and hold could be implemented as a possible means of mediating the glitch effect.

As with the A/D channels, the baseline noise was also taken for the D/A channels. Figure 2-15 shows a differential amplifier trace after adjusting for offset. The peak-peak and  $RMS_{ERROR}$  improvement provides over 2 effective bits of increased resolution. The dSPACE D/A is specified for 14-bit resolution, which corresponds to  $1.2 \text{ mV}$  per step. The high-resolution D/A uses a 16-bit converter for  $305 \mu\text{V}$  steps over the full range.

The noise present over the full range is further characterized in the plots in Figure 2-10. Static tests were also used to determine the total linearity, offset, and noise present. Figure 2-16 shows the RMS noise on the dSPACE D/A channel (dach1) versus the high-resolution D/A channels (da). There are several important features. The dSPACE channel has a peak effective resolution of approximately 12.3 bits while

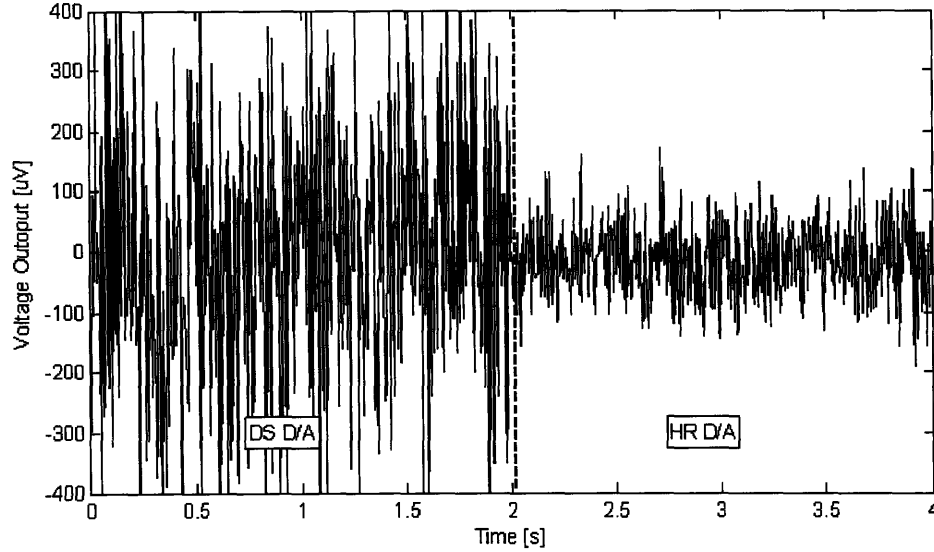


Figure 2-15: D/A baseline noise with 1 MHz low-pass filter on differential amplifier.

the high-resolution boards approach 15 bits. The RMS noise increases at the range extremes due to the architecture of a digital-to-analog converter, in that the reference voltage is multiplied by the digital code representing the desired output. The high-resolution design further amplifies this noise with an output gain of over 2. The noise present on the voltage reference is symmetrically scaled through the D/A converter over the output range. While the dSPACE D/A converter architecture has not been investigated, it is provided for relative comparisons and is assumed to operate similarly.

The offset and linearity for the system was also characterized. Differential nonlinearity is the error for a 1 LSB step and is rated at 0.15 LSB for the DAC. Integral nonlinearity is the worst case deviation between the ideal line and DAC output as adjusted for offset. This is rated at 8 LSB whereas I found an average of 6.2 LSB and an offset of  $-110 \mu V$ . I expected that offset and linearity adjustments or a lookup table within Simulink are implemented to maintain absolute voltage outputs. Additionally, the increased resolution is demonstrated by small amplitude signals. Figure 2-17 shows the output of a 3 mV amplitude sine wave of 100 Hz. The dSPACE D/A

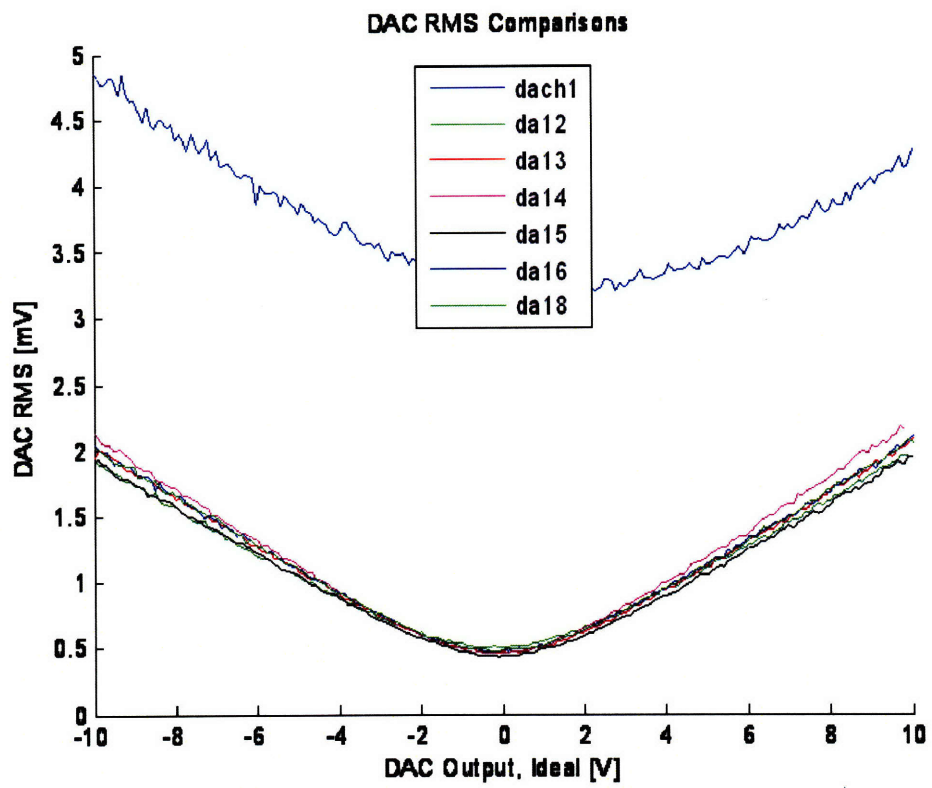


Figure 2-16: RMS noise present on D/A channels across full range. A single dSPACE D/A channel (dach1) is provided against several high-resolution D/A channels (da $xx$ ).

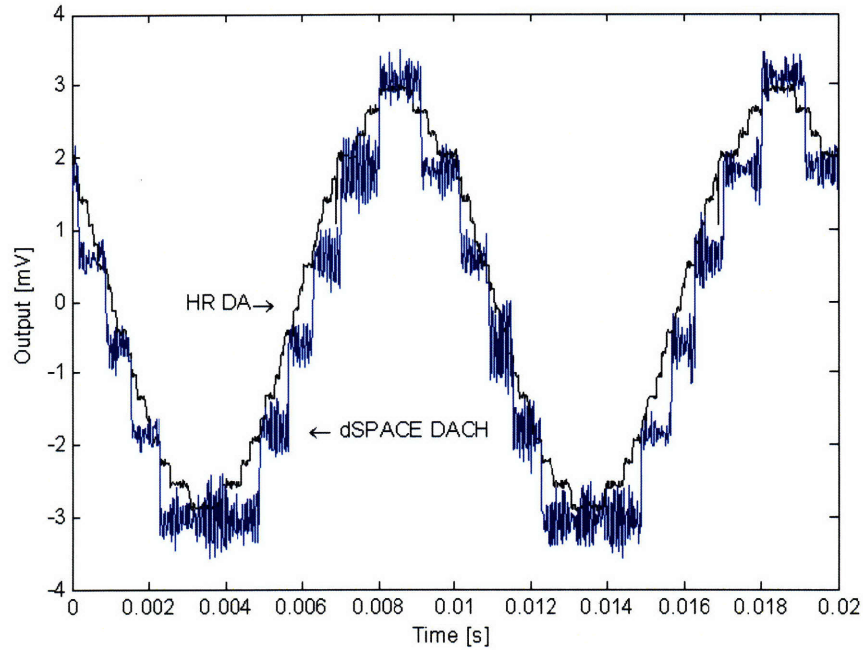


Figure 2-17: Small amplitude D/A output.

goes peak-peak in 5 steps, which is expected at

$$\frac{6mV}{5 \text{ quanta}} = 1.2mV \quad (2.6)$$

With a 30 kHz low-pass filter in place on the differential amplifier, the noise can be seen to be less than  $\pm 0.5$  bit, as expected from the specifications. The high-resolution D/A however provides 19 steps over 6mV, demonstrating the expected resolution of

$$\frac{6mV}{19 \text{ quanta}} = 316\mu V \quad (2.7)$$

The noise is also approaching 0.15 LSB.

This chapter described the high-resolution analog system previously design for the dSPACE digital platform as well as improvements and system characterization. The following chapters describe a new high-resolution, high-speed complete digital environment which I designed. The chapters describe project requirements, component selection, and system design.

## Chapter 3

# High-Speed, High-Resolution Digital Platform Requirements and Selection

This chapter introduces the second portion of my thesis and presents an application that requires both high-resolution and a high sampling rate analog-to-digital interface. This chapter details what is required for a digital platform, from the A/D to the digital processing component to the D/A, and compares and discusses why I selected the ultimately implemented design.

### 3.1 Application Description

In coordination with a project being led by Ian MacKenzie, the need for a high-speed, high-resolution digital acquisition and control system became necessary. The application was for an atomic force microscope (AFM) that could image a  $50 \times 50 \mu\text{m}$  surface with  $1024 \times 1024$  pixels at 1 Hz. This presents interesting mechanical challenges with respect to the actuation and sensing methods. In order to provide a proof of concept demonstration, a 2 degree-of-freedom positioner was designed by Ian MacKenzie [12]. This provides one scanning axis along the surface and a second height-following vertical axis. The remaining third axis required for imaging is driven

by a subsequent system either moving the sample or the 2-DOF scanner mechanism. Much lower accelerations and velocities are required on this third axis. The 50  $\mu\text{m}$  scan axis operates at resonance and the perpendicular axis provides random access positioning over a 10  $\mu\text{m}$  range. Stacked, 2-DOF electromagnetic actuators generate motion through constraining flexures. Position sensing is implemented with high-precision capacitive probes from ADE [19]. Subsequent design revisions will explore sensing from linear encoders and scales. A resolution of 0.1 nm RMS was desired on the scan axis and 0.02 nm RMS on the vertical axis. The hardware design is discussed further in Section 6.2.

To obtain the required specifications,  $2^{20}$  pixels must be acquired within 1 second. This is slightly over 1 million samples per second (MSPS). The capacitive probes were selected to minimize the RMS noise over the required range. The sample measurement needs to be good to 2 parts per million (ppm) on the scan axis to achieve 0.1 nm RMS resolution over a 50  $\mu\text{m}$  range. This is equivalent to 113.7 dB SNR or 18.6 equivalent number of bits. Care must be taken even in analog electronics to maintain 2 ppm resolutions, let alone analog and digital interfaces at the sampling speeds required by this application. The same requirements apply to the vertical axis because the RMS resolution and range are equivalently scaled. The selected high-performance capacitive probes have a 50 and 40  $\mu\text{m}$  range for the x- and z-axis, respectively. However, the probes were only specified at 0.77 nm and 0.36 nm RMS at 100 kHz bandwidth. This equates to a dynamic range of 96.2 dB and 100.9 dB for each axis, respectively. Even at these less stringent specifications, few options exist to provide lower RMS noise components.

The instrumentation needed to be capable of digitally acquiring pixels and storing them for review and post-processing at a subsequent time. In addition, real-time control must be implemented in order to scan at resonance and track the sample surface with the vertical axis. The vertical axis closed-loop bandwidth is set between 1 and 5 kHz. The closed-loop sampling rate needs to be at least 100 kHz for the vertical axis to accommodate up to a 5 kHz closed-loop bandwidth. Conceptually the real-time control could be completed with analog electronics and the pixel acquisition



could be completed with a high-resolution, high-speed analog-to-digital interface. An analog controller offers distinct advantages over a discrete control system because it is easily implemented with traditional op-amp circuits, has low cost, low latency, and high bandwidth. However, analog controllers are limited to gain, summing junctions, and traditional linear transfer functions. A digital control system, is much more flexible because it can implement complex feedforward, nonlinear, or discontinuous control that are not easily realized with analog circuits.

The following criteria were evaluated when selecting the digital control platform components:

- Required resources to develop/implement hardware and software to meet minimum baseline specifications
- Development/implementation 6-month timeline risk
- Flexibility for future expansion
- Single supplier dependence
- Technical support by supplier
- Cost

The three main sections that need to be selected and potentially designed if no commercial option is available are the analog-to-digital converter, the digital platform that acquires, stores and manipulates data, and the digital-to-analog converter to close the control loop. The rest of this chapter explores potential options for these three top-level components, their interdependence, and why each component was selected.

## **3.2 ADC Requirements and Selection**

The state-of-the-art technology in analog-to-digital converters exhibit a trade-off between high-resolution and high-speed. Different converting technologies exist that are advantageous for one aspect more than the other. Integration of an incoming signal requires a longer conversion time, but it also averages the signal to a higher preci-

sion over that period of time more than other methods. Alternatively, methods like flash conversion are extremely high-speed but require increasingly larger amounts of components for greater resolution. The number of integrated components in a flash converter increase by  $2^n$  for  $n$  bits. Different A/D topologies and their features were discussed in Section 1.2.

As shown in Chapter 2, it is possible to average samples to obtain a lower RMS noise. Conceptually it would then be possible to use a high-speed A/D converter of lower resolution and average the conversions to meet specifications. This however requires hardware operating at these higher speeds to complete the averaging. This technique can be implemented in hardware with a sigma-delta ADC or with a lower resolution ADC with a dedicated DSP or FPGA to sum and average the samples.

The required RMS resolution on the two separate axes is 18.6 effective bits, or a dynamic range of 113.7 dB. The specifications are given in dynamic range rather than an RMS voltage because the capacitive probe measurement can be scaled or shifted to match the ADC. The effective number of bits (ENOB) is different from the listed number of bits for an A/D converter. A true sample contains spectral noise that can vary by many LSB from sample to sample even though each individual sample is represented by a greater resolution of  $n$  bits. The effective number of bits accounts for this noise and describes the actual precision of the digital sample. A sample rated to 24-bit resolution offers 59.6 ppb native resolution. However, the 24-bit ADC selected for implementation only has a specified dynamic range of 100 dB. This equates to 16.3 ENOB or 12.2 ppm.

According to the described requirements, an ADC must have a sample rate of at least 1 MSPS. Operating at higher sampling speeds allows more flexibility in hardware operation. A minimum of 1 MSPS is obtained by assuming that a datapoint is taken uniformly over the sample area. For a resonant scanner, this implies that datapoints would be taken along the entire scan and in both directions of the scan axis. In many cases data is only collected in one direction along a scan axis to limit direction dependent hysteresis and errors. Therefore it is advisable to seek an ADC solution that has higher sampling rates available.

I initially evaluated potential commercial IC and then evaluated third-party vendors to implement the selected converter or a custom converter. Table 3.1 lists the highest performance IC options from the leading ADC suppliers.

Table 3.1: Analog-to-Digital Converter IC Commercial Options

ADC P/N	Supp.	Res. [bits]	SNR [dB]	Rate [MSPS]	Prop. Delay [ $\mu$ s]	Arch.
AD7760	AD	24	100	2.5	10.8	$\Sigma\Delta$
AD7641	AD	18	93.5	2	0	SAR
AD7982	AD	18	98	1	0	SAR
AD9446	AD	16	81.8	100	0.13	Pipeline
AD10678	AD	16	80.5	80	0.007	Subrange
ADS1271	TI	24	109	0.105	360	$\Sigma\Delta$
ADS8412	TI	16	93	2	0	SAR
ADS8482	TI	18	99	1	0	SAR
ADS1625	TI	18	93	1.25	20.8	$\Sigma\Delta$
LTC2440	LTC	24		0.0035		$\Sigma\Delta$
LTC2202	LTC	16	81.6	25	0.28	$\Sigma\Delta$
MAX11038	Maxim	16		2.25	0	SAR

From these findings of currently available ICs, there are three potential solutions. First is the AD7760, which I selected and developed. The AD7760 has an appropriate sampling rate and initially appears to satisfy the resolution requirement, however the dynamic range is 13 dB less than is required. Averaging by a factor of 2 will only increase the SNR to 103 dB. Another option is the AD9446. Clearly the native dynamic range is much too low but oversampling and averaging can be used. By averaging by a factor of 100, the dynamic range would then be 102 dB. The ADS1271 was not considered because of the very low sampling rate. Because the AD7760 and AD9446 have very similar signal-to-noise ratios after averaging, the AD7760 was selected because it presents a more flexible IC that does not require additional data processing. The SNR was found to be higher than 100 dB in actual testing. With post-processing and filtering, the measured SNR can surpass the requirement. The final option is the ADS8482 from Texas Instruments. Based strictly on dynamic range and sampling rates, this A/D matches the capacitive probe noise characterization and

the minimum sampling rate.

The AD7760 24-bit converter was also selected over 16- or 18-bit converters because the resolution is not constrained by the digital output precision. If the converter resolution were not limited by the dynamic range but rather by digital output precision, the 24-bit converter would have a precision of 144 db whereas the 18- and 16-bit converters would only have a precision of 108 and 96 dB, respectively. The datasheet dynamic range for these 16- and 18-bit converters are approaching their digital output limit. The AD7760 IC is able to measure up to a 109 dB dynamic range in experimental results.

The converter configuration affects how the A/D is implemented. Both the sigma-delta and pipeline converters introduce a significant propagation delay. This time delay causes a linear phase loss at increasing bandwidths. While the converter can provide temporally accurate samples for data acquisition, closed-loop control becomes more difficult at higher bandwidths. This delay places additional constraints on the rest of the control loop to minimize delays. For example, the selected AD7760 A/D has a propagation delay of 10.8  $\mu$ s whereas a sample from an SAR converter is available immediately following the conversion. The converter architecture indicates the order of group delay and the datasheet group delay is also listed in Table 3.1.

I also considered potential third-party solutions. This would include a constructed ADC interface with all supporting circuitry and an interface to subsequent hardware either through a computer data bus or digital outputs. Essentially the ADC would appear as a black-box peripheral to any higher-level control. These options are presented in Table 3.2 and are compared against developing our own ADC interface with the AD7760.

Table 3.2: Analog-to-Digital Converter Commercial Options

ADC	Supplier	Additional development
AD7760 converter	Analog Devices	PCB design & testing
AD7760 eval. board	Analog Devices	Interface & availability
X3-SFD PCIe <sup>1</sup>	Innovative Integration	Software Development
NI 5922	National Instruments	Resolution

The National Instruments platform does not provide a viable solution for a number of reasons. The instrument is designed as a digitizer/oscilloscope and is not intended for real-time applications. Interfacing to the data in a real-time environment would then be troublesome and introduce complexity. The converter has 24-bit resolution but only at low sampling rates. At greater than 1 MSPS, the resolution decreases to 16 bits. The system also costs on the order of \$15,000 to implement with a high-speed acquisition and control platform.

The AD7760 evaluation board could have provided the quickest development time as it would have only required interfacing to the digital connections, as well as being the lowest cost solution. From the time of product selection and the following ten months, Analog Devices was not able to deliver any evaluation boards due to lead-free certification delays. Even if the board were available, this route was not ideal for several reasons. The AD7760 requires a boot-up sequence upon startup to initialize settings and begin sampling. The evaluation boards would require a manual boot-up sequence through Analog Devices proprietary software. This is neither robust nor convenient. Additionally, there is no digital isolation between the PCB and the data acquisition/control hardware. This allows ground loops to form and digital noise to pass from the computer through sensitive analog electronics. Unbeknownst to me at the time of product selection, several errors also existed in the evaluation board design, which may account for delays in availability. These were found during the course of the custom PCB design and testing as the design was adapted from the evaluation board design. These errors are discussed in Chapter 4.

At the time of the product survey, building our own PCB based on the AD7760, the AD7760 evaluation, and the National Instruments system were the only available options. Thus I decided that a custom A/D PCB based on the AD7760 was necessary. This design is detailed in Chapter 4. After the implemented design was built and debugged, the Innovative Integration X3-SDF product was released. The third-party solution represents a viable option had it been available, and is listed for retrospective comparison. The product is shown in Figure 3-1.

---

<sup>1</sup>Product not available for initial product survey, [www.innovative-dsp.com](http://www.innovative-dsp.com)

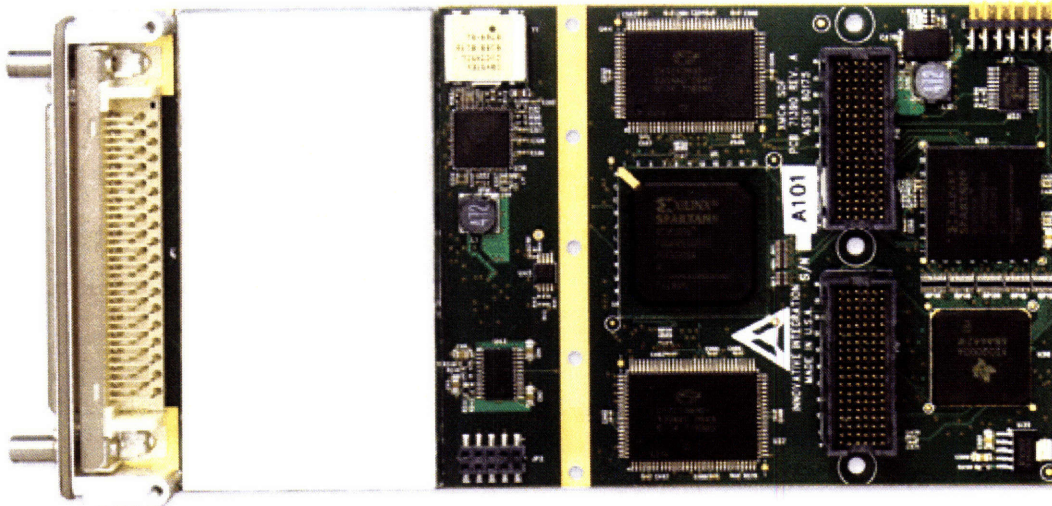


Figure 3-1: X3-SDF hardware by Innovative Integration [5].

The X3-SDF is based on four ADC channels with a 1-million gate Xilinx FPGA on a PCIe board. An external connector provides analog inputs, external triggering and synchronization, and 44 digital inputs/outputs. There is 4 MB of SRAM included on the board as well as direct memory access (DMA) hardware to allow DMA transfer for data acquisition.

The four A/D converters are the AD7760 ICs also implemented in the custom PCB design. They can be completely synchronized and the AD7760 registers are accessed through mapped libraries in software provided by Innovative Integration. These libraries are available for \$2,500 in addition to the hardware. FPGA functionality and the user interface are accessed through Matlab/Simulink blocks. The Innovative Integration blocks are similar to the dSPACE blocks; external peripherals are accessed by custom, proprietary S-Functions. This essentially hides their operation but allows access to their functionality. User defined code is then linked to these peripherals. The Simulink program is compiled with Xilinx's ISE CORE Generator, part of the Xilinx ISE Foundation pack and available for \$2,500.

The X3-SDF is intended for hardware-in-the-loop operation and can have very high loop rates when the PCIe bus and host computer are not included in the loop. Latency, or propagation delay, due to the AD7760 conversion is still a significant

factor. To utilize DMA transfers or include the host computer in the loop, a packet protocol on the PCI bus is utilized. The X3-SDF packages a data packet and transmits it over the PCI bus when a user-determined amount of data is collected on the DMA target. This generates an interrupt in the host computer to perform the necessary action, whether it be storing data or manipulating it and thus returning data to the X3-SDF. The loop determinism is greatly decreased due to the host computer operating system when the host computer is included in the loop. This product is not designed to operate in a real-time operating system (RTOS) platform such as the National Instruments RTOS and is intended primarily for data acquisitions. In this application the FPGA would strictly provide the data processing structure to implement controller and an external D/A would close the loop.

The costs associated with implementing the X3-SDF are shown in Table 3.3. Several disadvantages to this product are the lack of potential for future expansion, user operation, and long-term availability risk. The X3-SDF is designed as a stand-alone card in a host computer. While more cards could be added, their communication relies on the PCIe bus. The user operation is not as simple as merely designing a Simulink model and compiling it as with dSPACE products. A graphical user interface needs to be written by the user and the libraries for this are only available in C++, requiring user proficiency in this language. As opposed to the National Instruments LabVIEW platform which is designed for much more high-level drag-and-drop programming, the learning curve on the X3-SDF related software is much more difficult. Additionally, the Simulink blockset options are limited to those that can be implemented in FPGA logic, which removes many of the commonly used control and floating-point functions. Control would need to be custom designed. Even with LabVIEW features designed specifically for this task, control implementation proved to be difficult. Designing the control implementation from scratch adds more development risk to this option. Although the extent to which the X3-SDF and custom designed PCB differ in design, the analog signal input resistors would need to be changed to accommodate the fully differential  $\pm 10$  V signals that are measured from the capacitive probes. Lastly, the level of available user support for the hardware

and software from Innovative Integration is not known, and could potentially become an issue.

Table 3.3: Innovative Integration X3-SDF Implementation Estimated Costs

Description	Cost	Supplier
X3-SDF Hardware	\$2,000	Innovative Integration
FrameWork Logic Software	\$2,500	Innovative Integration
FPGA Code ISE Foundation	\$2,500	Xilinx
Real-Time/Development Computer	\$1,500	Dell
<b>Total Cost</b>	<b>\$8,500</b>	

I selected the AD7760 IC to be developed on a custom PCB for the analog-to-digital portion of the digital platform. Even if the X3-SDF product were available at the time of the initial product survey, the custom design has more flexibility than a fully packaged product. Likewise, the AD7760 evaluation board lacked both availability, features, and flexibility as opposed to the custom PCB design. The AD7760 was selected over other converter options because it provides the greatest resolution at the greatest sampling rate. However, the propagation delay negatively impacts the controllability of high bandwidth control loops.

### 3.3 Digital Platform Requirements and Selection

The digital platform needs to collect, process/manipulate, and output data at MHz rates, as well as acquire samples and store them for viewing and post-processing. The application requires a 1 second acquisition with a minimum of  $2^{20}$  data samples. The selected ADC was the AD7760 IC with a sample rate of 2.5 MSPS and thus data acquisition must operate at this minimum rate. Additionally, the system must be able to store these samples. Assuming a 24-bit value and 2.5 million samples collected in 1 second, the data rate over this single second is then 7.5 MB/second per channel. Unless custom data bus protocols are written that define the length of packet sizes, typical DMA transmissions occur with 32-bit word lengths, or 4 bytes, requiring a 10 MB/second per channel data rate. This data rate does not need to be sustained but



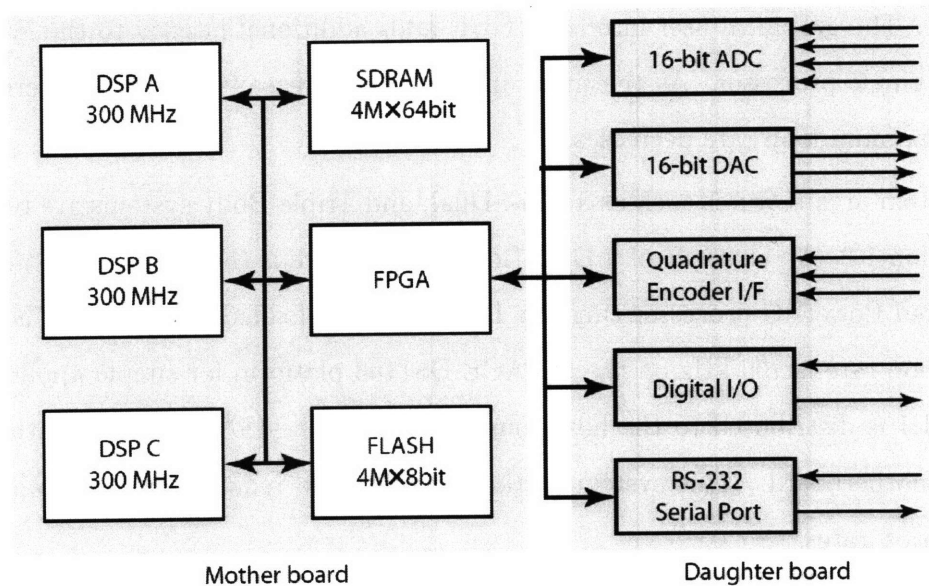


Figure 3-2: Real-Time computer developed by Xiaodong Lu [6].

cannot have dropped samples. The burst data rate becomes a factor for a minimum of two acquisition channels.

Other issues considered in evaluating potential commercial digital platforms versus what could be developed with laboratory resources included performance, timeline, future expansion, development support, and cost. The resources for a custom designed digital platform are evaluated on what has previously been done within the Precision Motion Control Laboratory by Xiaodong Lu [6]. The block diagram for the real-time computer system Xiaodong Lu developed is shown in Figure 3-2.

Xiaodong Lu compares the indeterminant temporal responses from existing real-time services (RTS) and the need for a wholly dedicated platform to operate with up to MHz loop rates [6]. The most flexible and universal RTS operate with interrupts. Peripherals initiate an interrupt when they require attention and the computer responds based on the interrupt priority and the current RTS operation. The processor saves the current operations to the stack, processes the interrupt function, reloads the stack, and continues with its previous operation. The RTS response to an interrupt has jitter on the  $\mu\text{s}$  level that is responsible for an indeterminant temporal response. This environment also requires that user interaction/monitoring use processor re-

sources. The graphical user interface (GUI) adds additional latency to the loop rate with a single processor. As the jitter and latency increase, the overall determinism and maximum loop rate decreases.

System architectures such as a Uni-, Dual, and Triple-Body systems are reviewed in Section 1.2. An example of a Dual-Body architecture is the dSPACE platform. A dedicated PowerPC processor runs the RTS with user-loaded operations. The maximum loop rate is 100 kHz on the dSPACE DS1103 platform for simple applications. The GUI is distributed to the host computer across the ISA data bus on the computer motherboard. This system neither provides the requisite A/D resolution or acquisition rates.

Xiaodong Lu's system, named the ThunderStorm, is a Triple-Body design. An FPGA front-end is used to interface with the high-speed peripheral inputs and outputs. The FPGA brings in analog inputs at 1 MSPS as well as tracking a digital encoder and moves the samples to the multiprocessor data bus into shared memory. Two dedicated DSPs poll for ready data, perform data manipulation computations, and return the output data to the shared memory where the FPGA closes the loop by outputting samples on the D/A channels. Polling operation is used instead of interrupts to eliminate the interrupt associated latency. This is possible because the DSPs are fully dedicated to a single processing thread. A third servant DSP utilizes the data bus in the interim and processes the GUI interface. Because a GUI requires relatively low update rates, the servant DSP is able to utilize resources as a secondary user while they are not being used by hardware that is maintaining critical loop rates. The servant DSP outputs data through parallel logic in the FPGA as serial output to a host computer GUI. The serial interface is a good example of the flexibility of an FPGA. Standardized RS-232 serial communication has a maximum baud rate of 115.2 kHz with 8-bit packets, whereas the custom designed serial interface by Xiaodong Lu has a maximum baud rate of 1 MHz with 64-bit packets.

This Triple-Body architecture is capable of 1 MHz loop rates with four ADCs, four DACs, quadrature encoder tracking, digital I/O, and a 64-bit serial port. The ADCs used by Lu are Linear Technology LTC8412 and are presented in Table 3.1. The

motherboard was constructed on a 12-layer central processing PCB and connected to a daughter peripheral board that interfaced through the FPGA. The ADCs are 16-bit, 2 MSPS SAR converters. If this digital design were adapted to this project, the A/D dynamic range would be a limiting factor and the A/D would need to be replaced. The system loop rate of 1 MHz would be entirely acceptable because while data acquisition occurs at 2.5 MSPS with the AD7760, the control loop can be decimated to lower loop rates. This decimation can be completed in the FPGA and not be limited by the DSPs. Assuming the real-time computer could be adapted within the available timeframe, we would own the entire design and have the resources to develop it due to the extensive documentation of Xiaodong Lu's original design. There would be a large risk associated with prototyping the design because any flaw in the layout would void that revision and all the prototyped hardware. Vendor support would be limited beyond the available datasheets for individual components. The system could also be adapted to future expansion as it would not be strictly dependent on any particular technology or vendor.

The primary reason why this design was not adapted to this application was the available timeline and development risk. Xiaodong Lu spent over a year on the development of the hardware design and software interface. At the time I was evaluating potential platforms, less than six months remained before a scheduled project demonstration. This was not enough time to become familiar with Xiaodong Lu's design, adapt, construct, debug, and test it. The DSP and FPGA coding was also implemented in C and VHDL, both low-level languages relative to LabVIEW. Being able to efficiently work in these languages on Lu's system would present its own learning curve. Of all potential options this design presented the most risk, particularly with the cost. An estimated cost of \$11,000 is presented in Table 3.4. The ADC and DAC hardware is not included in this cost summary.

The option exists to implement the data processing on DSPs or an FPGA. The DSPs allow more complex arithmetic as well as more precise computations, however computation time increases with precision and complexity. An FPGA would be still required as a digital front-end as was used in the ThunderStorm architecture. A single

Table 3.4: Custom Real-Time Computer of Xiaodong Lu Estimated Costs

Item	Cost	Supplier
Xilinx FPGA	\$3,000	Xilinx
FPGA Code ISE Foundation	\$2,500	Xilinx
Two DSPs	\$600	TI
DSP TI Development Software	\$3,600	TI
DSP Programmer JTAG Emulator	\$1,000	TI
PCB Prototype and Assemble	\$1,300	Advanced Circuits
<b>Total Cost</b>	<b>\$11,000</b>	

FPGA however is capable of both the digital interfacing and control implementation. As the technology has become more common in industry, prices have decreased for correspondingly increasing performance. A distinct benefit of an FPGA is the hardware parallelism. Logic elements are clocked in parallel whereas a DSP runs lines of instructions in sequence. For a complex operation to be completed on an FPGA, it is possible to break up the operation into simpler, more efficient parallel tasks. If the ThunderStorm architecture were to be adapted to this application, it would be necessary to reevaluate whether DSPs would be the proper processing hardware. Because of the necessity to have at least one FPGA in the system, products based on an FPGA were evaluated.

There are numerous companies that provide embedded high-performance digital environments. The simplest would begin with an evaluation or development board from Xilinx or Altera. These boards are available for all product lines and generally come with complete documentation and even demonstration software. Additionally, the hardware is completely tested. The compromise comes in the number of available I/O channels. These boards are generally designed to demonstrate a wide range of peripherals such as audio I/O, video I/O, data storage, or PCI interface. Compared to the custom real-time computer described above, this solution requires fewer resources on hardware development and testing but more resources on interfacing with a host system. Because the application requires data acquisition, a host interface is critical as opposed to entirely embedded control.

Most FPGA vendors offer embedded DSP solutions. Newer product lines have the DSP engine constructed directly on the FPGA fabric and are able to create a microcontroller style unit for high-speed throughput. Other models have an FPGA front-end and dedicated DSPs for computations. Although not determined to be essential for acquisition and linear control, DSPs allow complex computations generally not easily implemented in fixed-point FPGA HDL. Therefore it would be beneficial, although not required, to select an option with this flexibility.

The evaluation or development board solution is fairly flexible and allows a relatively inexpensive development platform to be adapted to future designs. Ideally the development kit would be tested and the features most useful would be developed into a custom PCB platform when deadlines are less stressed. An advantage of the development kits are that the design is completely open and basic design files such as bill of materials and PCB Gerber files are freely available. With the decision to design our own ADC PCB, being able to design the low-level FPGA code to interface with the A/Ds becomes a concern with respect to the 6-month project window. Other than the time to implement, there is little risk, as sufficient technical documentation is available and the platform selection would not constrain future development and expansion with different brands and products. Available development kits range widely in price and features available, however cost estimates for a higher-end model are shown in Table 3.5 at \$5,400.

Table 3.5: Xilinx Development Platform Estimated Costs

<b>Item</b>	<b>Cost</b>	<b>Supplier</b>
FPGA/DSP Development Kit	\$2,500	Xilinx
FPGA Code ISE Foundation	\$2,500	Xilinx
Interface Hardware	\$400	Misc.
<b>Total Cost</b>	<b>\$5,400</b>	

Third-party vendors also provide general products like the development kits available from Xilinx and Altera as well as more specialized products. A survey included the companies VMETRO, Hunt Engineering, HiTech Global Distribution LLC, Xelic

Inc., Cast Inc., Coreworks, and Nallatech. These products are generally intended for direct industrial implementation as opposed to a development platform. They also usually come with custom development software that can range from low-level HDL to high-level graphical code like LabVIEW. Because the products are specialized, it is easy to find one strictly for digital I/O, acquisition, and high-speed control.

Formal quotes and detailed product exploration was completed on VMETRO [22] options. Estimated costs are shown in Table 3.6. They offer a range of FPGA/DSP combined industry products. These products can be stand-alone or operate within a computer chassis on a PCI bus. Also included with the product are development tools for intellectual property (IP) development. Provided software includes libraries, examples, and extensive documentation for target as well as host development. A hardware block diagram for a potential solution is shown in Figure 3-3. VMETRO also offers products with analog interfaces but not to the resolution required for this application. The host interface interacts through the PCI bus and is implemented by C++ libraries for turnkey solutions. FPGA development can be completed within VHDL or Matlab through the Xilinx System Generator. These are both low-level programming environments with few complex operations included.

Table 3.6: Third-Party Digital Platform Estimated Costs

Item	Cost	Supplier
VMETRO FPGA/DSP Hardware	\$10,000	VMETRO
Programming Software	\$2,300	VMETRO
Support License	\$200	VMETRO
<b>Total Cost</b>	<b>\$12,500</b>	

Although this solution presents a viable option, it is less flexible for future development. In order to build on the software development required for this product, we would need to stay with products from the same company which could become limiting depending on what is needed in the future. Also, the amount of support required and provided are not entirely known without becoming deeply involved in the development. Lastly, there is a minimum lead-time of four weeks and potentially

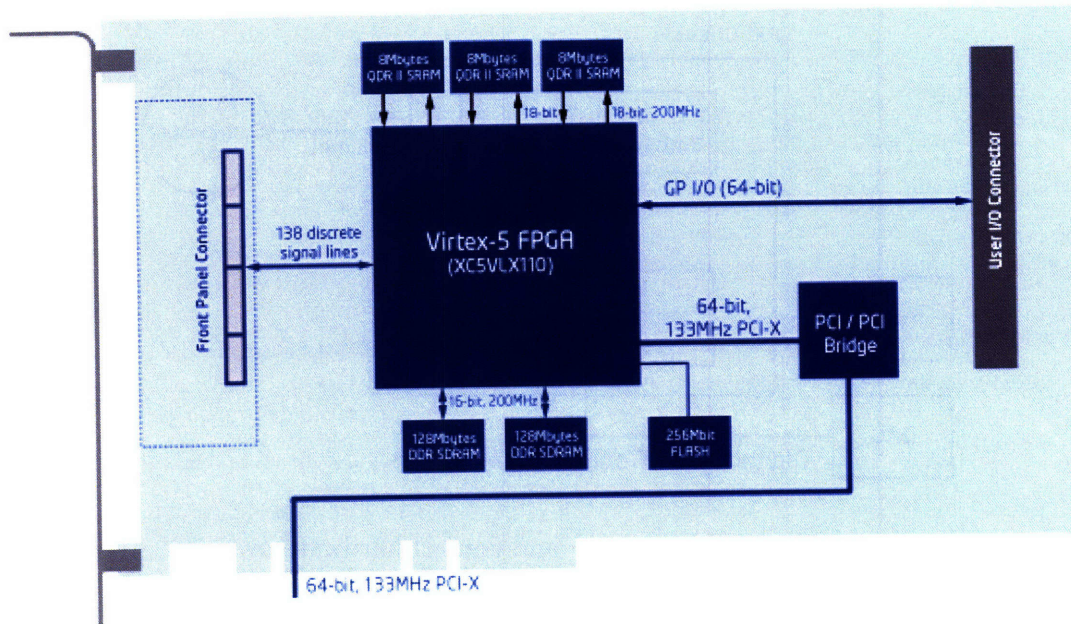


Figure 3-3: VMETRO embedded computing FPGA block diagram [7].

up to 10 weeks. For a product that would require significant software development, this lead-time becomes a risk in the project timeline.

Another third-party supplier not yet discussed is National Instruments (NI). They provide turnkey hardware and software to design, prototype, and deploy systems for measurement, automation, and embedded applications. They have a significant market share and presence in both academia and industry in a reported 25,000 different companies. Their software is based on NI LabVIEW which is a high-level graphical coding interface. Their hardware ranges from data acquisition to vision to real-time control. They provided the ultimately selected digital platform.

Both real-time hardware and software platforms are available. These embedded machines have a dedicated computer to run the real-time operating system (RTOS) while a host machine runs the user GUI. These systems have more determinism than a Windows-based system but can still have jitter on the order of  $\mu\text{s}$  which limits loop rates to less than 1 MHz. A recent product roll-out has been the FPGA Module. Select NI LabVIEW graphical coding elements are available to be compiled and implemented on the FPGA. Different commercial options for purchase include

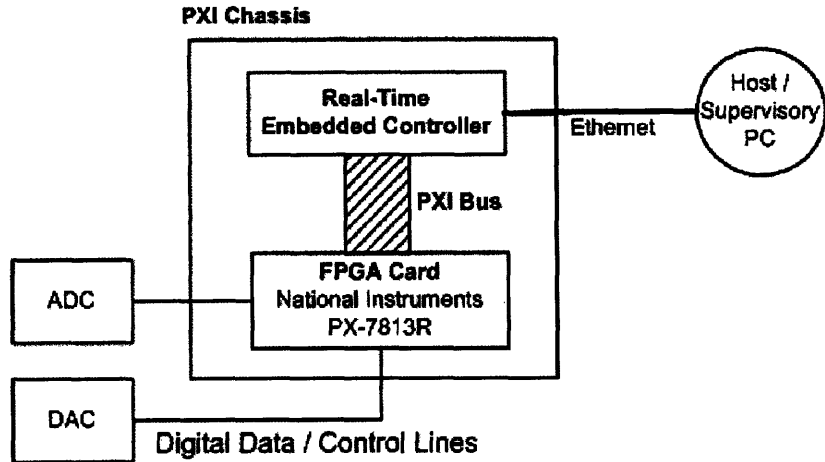


Figure 3-4: Data acquisition and control hardware overview.

analog converters and the number of logic elements on the FPGA. The hardware required to operate the FPGA is all local to the FPGA PCB and thus it can be deployed to a host or real-time dedicated computer. In the dedicated computer setup, the embedded computer and resources can be utilized by the FPGA. Figure 3-4 is a block diagram of the data acquisition system with only a single control channel.

A previous research project within the PMC Laboratory utilized a NI real-time application. The hardware was located in a PXI chassis. The PXI standards are built on the PCI computer backplane but intended for rugged, industrial applications. The embedded computer provides the DSP capabilities for slower loop rates and the off-line data acquisition and data storage required for this application. Another benefit is the high-level graphical coding interface which does not require an intimate knowledge of HDL but permits VHDL to be implemented for custom blocksets. The coding style easily allows both sequential and parallel loops with drag-and-drop interfaces to digital I/O and features on the embedded computer. These features include access to data transfer and storage. Included on the FPGA board is DMA hardware which allows a DMA buffer to control the PXI data bus and directly access the system memory without working through the processor.

The National Instruments FPGA platform provides the lowest risk of any options.



The technical support has been demonstrated in the past and the coding language is intended for quick implementations by beginner users. The hardware lead-time was approximately two weeks which was beneficial from a timeline aspect. The estimated costs are shown in Table 3.7.

Table 3.7: National Instruments Digital Platform Estimated Costs

Item	Cost New	Cost to PMC	Supplier
PXI-7813R FPGA Hardware	\$2,160	\$2,160	NI
Breakout Hardware	\$1,440	\$1,440	NI
PXI Chassis	\$800	\$0	NI
PXI Computer	\$3,145	\$0	NI
NI Software	\$2,000	\$2,000	NI
<b>Total Cost</b>	<b>\$9,545</b>	<b>\$5,600</b>	

National Instruments also offers such a wide range of options so that future expansion is not limited. Additional FPGA boards can fill seven additional PXI slots in the PXI chassis. Each board has a direct bus to each neighbor, allowing high-speed communication without relying on the backplane and its associated delays. Additional embedded controllers can also be added in parallel with the existing PXI chassis. Alternatively, FPGA and other real-time hardware can be implemented in a user-purchased computer chassis, thus allowing the user to determine the amount of real-time performance required. The latest NI Real-Time Module utilizes multiple computing cores as well. One can specify the threads running on each core within software and a quad-core processor is then the equivalent of four parallel DSPs.

There are advantages and disadvantages to each potential digital platform solution, but NI offers a fast implementation turnkey option and was subsequently selected for this design. The hardware setup for the embedded controller and FPGA board in a PXI chassis is shown in Figure 3-5. The embedded controller and PXI chassis were existing hardware and used for this setup.

The selected FPGA board (PXI-7813R) has 160 digital I/O with a Xilinx 3-million gate XC2V3000-4FG676I FPGA. Additional digital I/O were considered more important than low-resolution analog interfaces. The 3-million gate FPGA was selected over

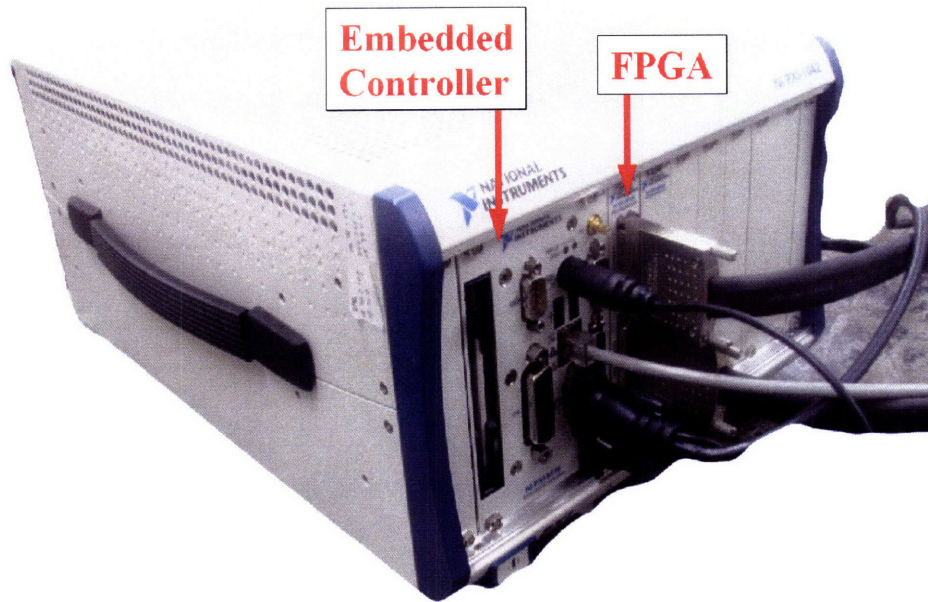


Figure 3-5: National Instruments PXI chassis with embedded controller and FPGA board

a 1-million gate option to allow more complex control algorithms to be implemented. The baseline clock is 40 MHz but additional clock rates can be derived at numerous speeds. The PXI chassis also has a real-time embedded controller and runs a real-time operating system that dedicates the hardware to specific tasks and does not need to deal with the overhead associated with an operating systems such as Windows. The embedded controller is a NI PXI-8176, which has a Pentium III 1.26 GHz processor with 384 MB of RAM. The host/supervisory PC interfaces with the embedded controller and allows user interaction and data monitoring.

Table 3.8 compares the various digital platform options. I selected the National Instruments option because it offered high-level programming with complex operations that could be implemented quickly. It also has the lowest risk and a relatively low cost. Because the coding language is very high-level but still accommodates low-level HDL, it has the flexibility to quickly implement processing algorithms not available in other options while still being able to build custom complex HDL operations from the ground up. Table 3.8 presents estimates and in retrospect these are not entirely

accurate. For example, the development time for the NI system was approximately 6 months due to the control implementation. Section 7.2 discusses whether the NI system was the ideal selection for the defined application.

Table 3.8: Digital Platform Estimated Comparison

Platform	Cost	Dev Time	Risk	Flx Exp	Support
X3-SDF	\$8,500	2 months	Med	Low	???
Custom RT Comp	\$11,000	6 months	Med	High	Med
Xilinx Dev Brd	\$5,400	2 months	Low	Med	Low
3rd Party Dev Brd	\$12,500	1.5 months	Med	Low	???
NI Real-Time	\$5,600	1 month	Low	High	High

### 3.4 DAC Requirements and Selection

An output D/A converter is required for closed-loop control to supply a control signal to power amplifiers and thus the electromagnetic actuators. The D/A selection is not as difficult as the A/D selection because there are generally fewer trade-offs in resolution versus speed and the requirements are not as demanding. The highest resolution converters available are 16-bit for high-speed operation. Faster settling times are available by using a current output because a voltage does not need to slew across any built-in or stray capacitance. However, a voltage is required to drive the power amplifier. An external op-amp can be added to convert the current to a voltage but this generally creates a slower settling rate than a monolithic IC that is designed to output a voltage.

There are few specifications related to selecting a D/A converter. The minimum closed-loop sampling rate is 100 kHz to accommodate potential z-axis bandwidths because the sampling rate should be at least 10-20 times greater than the expected closed-loop bandwidth [20]. Otherwise the control scheme needs to be adjusted. Other requirements are that the DAC must be able to be implemented within the timeline of the project and not be a dominant noise source in the control loop. Cost is a consideration as well.

Table 3.9 lists several commercial D/A IC converter options and is only a sampling of what is available. These are representative of the more important metrics. One metric not specified is the digital data input format. The update rate of the converters is limited by the rate at which data can be clocked into the data registers and latched. The LTC1650 requires a minimum of 80 ns clock pulses for all 16-bits while the LTC2641-16 requires only 19 ns per clock pulse. The AD768, DA712, and LTC1821 are parallel input DACs and have a 16-bit data bus along with three control lines. This allows data to be latched on the order of 100 ns. While it is initially tempting to select a converter that can be updated extremely fast, the additional parallel data bits require additional FPGA digital outputs. With 19 outputs per parallel converter, the number of converters that could be in a system becomes limited. The AD768 is a current output DAC which is why settling time is the fastest. This requires an output op-amp to generate the output voltage and thus the true settling time is limited by the selected op-amp.

Table 3.9: Digital-to-Analog Converter IC Commercial Options

DAC	Supp.	Res. [bits]	Settling Time [ $\mu$ s]	Rate [MSPS]	Cost
AD5542	AD	16	1	1.5	\$16.50
AD768	AD	16	0.025	30	\$42.41
DAC712	TI	16	4	10	\$28.50
LTC1650	LTC	16	4	0.694	\$32.50
LTC1821	LTC	16	2	5	\$87.13
LTC2641-16	LTC	16	1	3.09	\$12.83

I selected the LTC1650 DAC because it was already designed into a modular PCB with the work described in Chapter 2 and provides comparable performance to other high-performance DACs. The DAC design surpasses the minimum output rate requirement and has already demonstrated operation with a full noise characterization. The PCB and associated IC converter could potentially be upgraded at a later time with little or no affect on the rest of the system. The biggest advantage of this selection was that there were already several operational D/A PCBs available in lab

which meant there was no implementation risk or cost to this portion of the project. The D/A PCB design is not described further, but the software interface is described in Chapter 5. The complete schematic is included in Appendix A.

This chapter described the baseline specifications and component selection for a high-resolution, high-speed data acquisition and control digital platform including the A/D converter, digital processor, and D/A converter. The AD7760 was selected as the A/D converter because it offered the highest performance, particularly the highest dynamic range while meeting the 1 MSPS requirement. The converter architecture does introduce a propagation delay which will significantly affect a closed-loop phase response. Despite comparable products becoming available after completion of the AD7760 PCB, the custom design remains the best selection. A National Instruments FPGA system was also selected because it offers high-level functionality with the least risk to implement while maintaining flexibility for future expansion. The custom PCB design to implement the AD7760 is described in the next chapter.



# Chapter 4

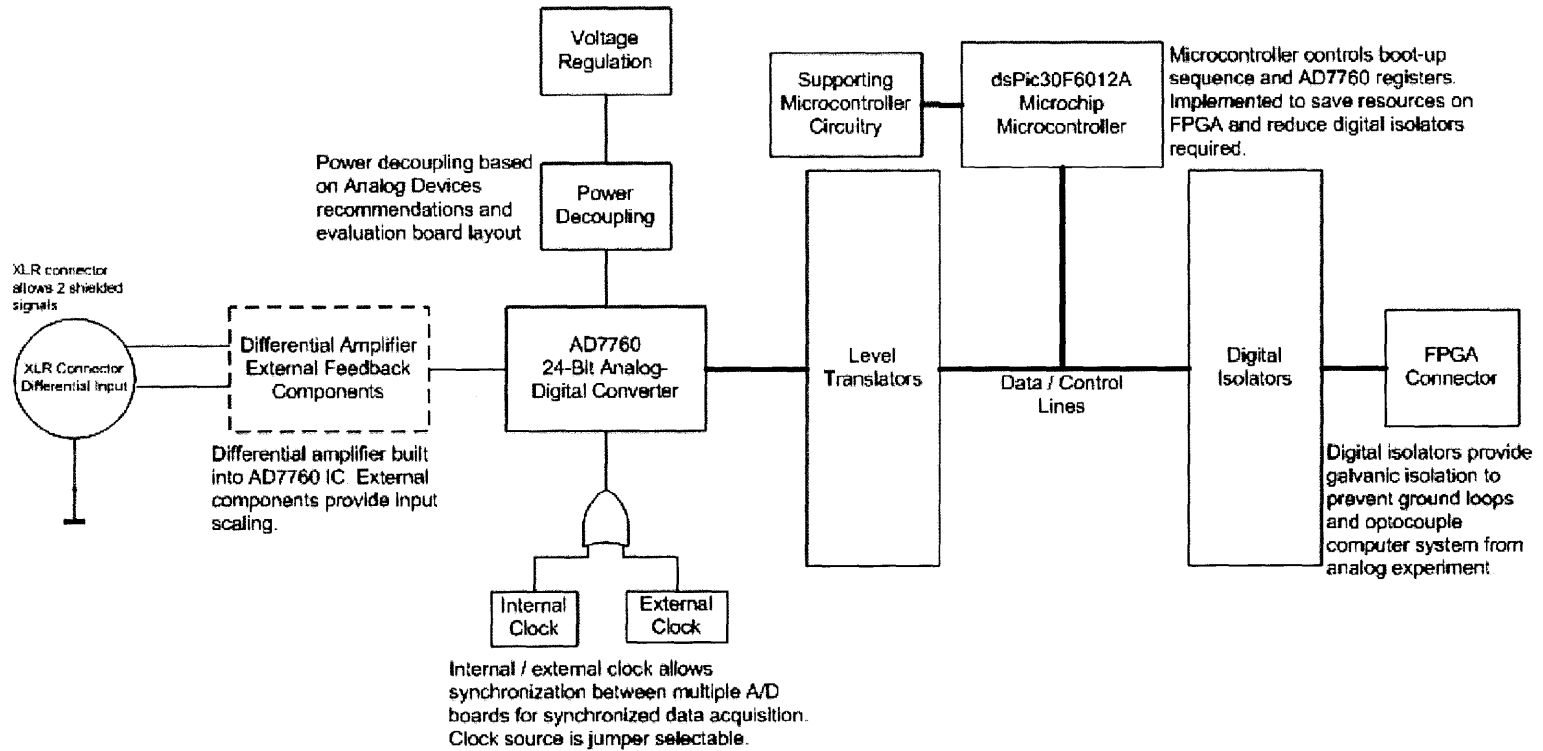
## 24-bit A/D Circuit & PCB Design

This chapter describes the detailed design, debugging, and development process of the A/D circuit and PCB. Results and characterization of the circuit are presented in Chapter 6.

There are distinct functional areas required for the PCB to operate correctly. The analog signal must be cleanly brought onto the PCB and signal conditioning is used to properly shift and scale the input signal to that required by the ADC IC. Power conditioning must be implemented to reduce any affects from power supply variations or feedthrough of unwanted signals/noise. Coming out of the AD7760, the digital output must be interfaced from the IC to any external platform, such as the FPGA. Digital galvanic isolation is implemented to reduce ground loop affects between the analog plant and digital electronics. The AD7760 requires an external boot-up sequence to be inputted on the same bidirectional data bus that the digital conversion is outputted on. This boot-up sequence can be implemented by the FPGA or digital circuitry located on the PCB.

A high-level view of the PCB with these functional groups is shown in Figure 4-1. The complete schematic is provided in Appendix A along with the bill of materials. The PCB design is adapted from the Analog Devices AD7760 evaluation board [2, 38]. The PCB is shown in Figure 4-2 with the functional groups labeled.

Figure 4-1: ADC PCB functional diagram.





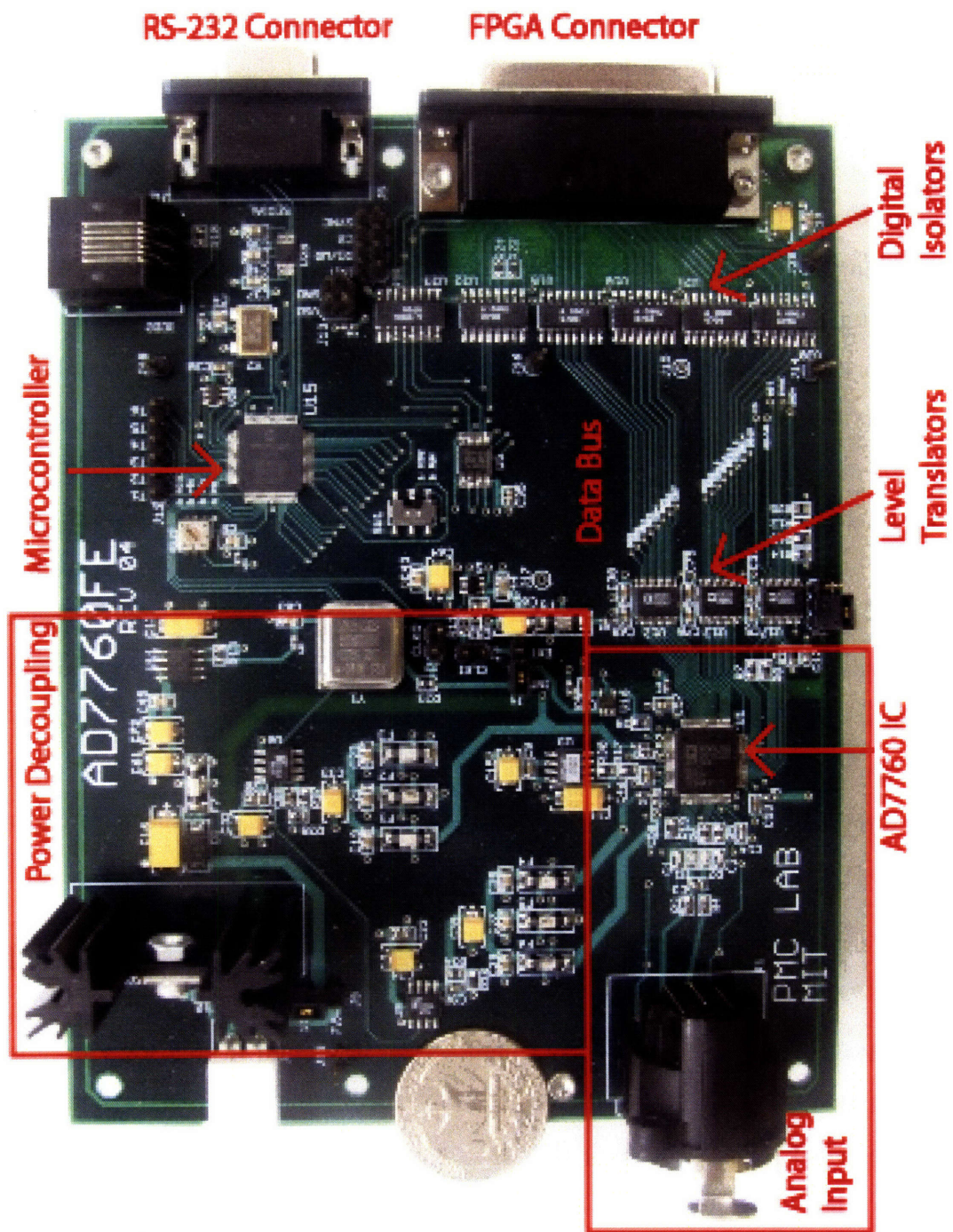


Figure 4-2: ADC PCB with functional areas (quarter shown for scale).

## 4.1 Analog Interface

An A/D converter is designed to generate a digital word for a corresponding differential input voltage. Some designs use a reference as one of the differential voltages, such as a shared common or earth ground. A ground reference refers to a single-ended input whereas pseudo-differential inputs have a user defined common as the common. The single-ended input is the least robust because the common can potentially form a ground loop and requires that the data acquisition hardware and device under test be at the same potential. This is difficult to achieve when instrumentation is remotely located or sensors are referenced to different commons. Also, this is impossible to achieve in the presence of any loops coupled by magnetic fields. In these cases, differential or instrumentation amplifiers are required to separate these commons, which adds an additional layer of complexity and source of error. A single-ended interface can be converted to a differential signal by pairing the signal with its inverted complement. A simple inverting op-amp circuit implements the inversion. However, the inverting circuit introduces additional issues of isolation and accuracy and the differential pair is not truly isolated.

A more robust approach is to use fully differential inputs. An example of differential inputs and their requisite shift and scaling for the AD7760 is shown in Figure 4-3. The scale factor required is

$$f_{scale} = \frac{3.685 - 2.048}{10} = 0.1637 \quad (4.1)$$

This scale factor converts the expected input voltage levels into an appropriate voltage range for the A/D. We also must offset the input to 2.048 V. If a pseudo-differential or single-ended input were used, the scale factor  $f_{scale}$  would be twice as big and only provide half as much common mode rejection. Common mode signals are most commonly seen as 60 Hz, or some multiple of 60 Hz, “noise”. Technically this is not noise because the source can be attributed to poor power supply rejection or a ground loop and is instead a disturbance. However these terms are commonly used interchangeably in literature. Other sources of error include electromagnetic

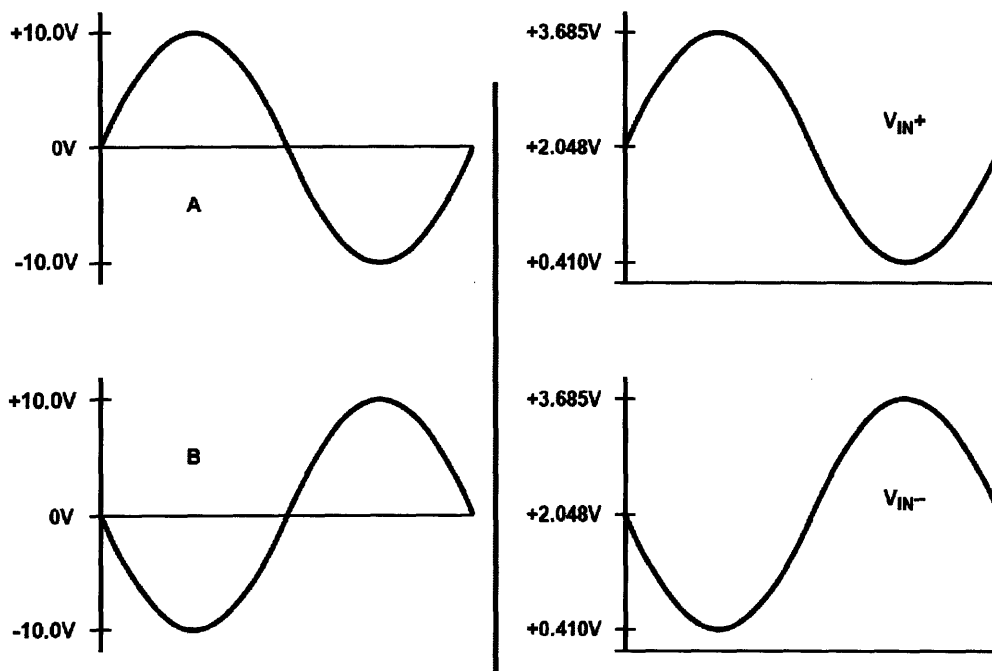


Figure 4-3: Differential input shift and scaling; analog input to PCB (left) and analog input to AD7760 IC (right).

interference such as from electronic ballasts in fluorescent lights or nearby high-power equipment and motors. The differential inputs do not ideally rely on any common voltage because if the common voltage increases relative to earth ground, each signal voltage also increases by that amount. Therefore the relative separation  $V_{IN+} - V_{IN-}$  remains the same. Typically the signals are carried on two lines that are kept in close proximity to one another. Noise invariably couples to any cabling. However, when it couples equally to both differential signals it appears as a common mode voltage disturbance. By rejecting this common mode voltage, the system is more robust to external noise.

The shift and scaling is performed with a fully differential amplifier as shown in Figure 4-5. As opposed to a standard operational amplifier which has a single output, a fully differential amplifier has two differential outputs. The symbol for such an amplifier is shown in Figure 4-4. The output common mode voltage,  $V_{OCM}$ , is independently set after the input common mode voltage is rejected. Figure 4-5 shows a simplified representation of the internal components of a fully differential amplifier

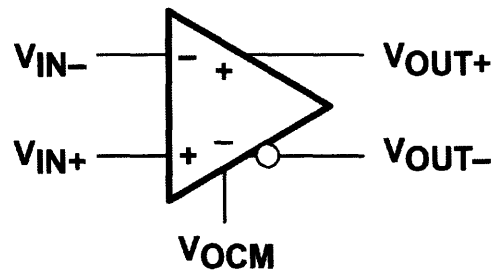


Figure 4-4: Fully differential amplifier symbol.

[8] and shows how common mode input voltage is rejected while independently setting the output common mode voltage. A differential front-end of  $Q_1$  and  $Q_2$  creates differential currents that are mirrored to the gain stage. The input common mode voltage is rejected in this first stage. The differential outputs are obtained by sampling both sides of the gain stage. A final output stage is shown with output buffers. The output common mode voltage is maintained by internal feedback as set by  $V_{OCM}$ .

Commercially available fully differential amplifiers are generally high speed, low cost, have a small footprint, and have low power consumption. Although not critical to this application, differential amplifiers also inherently introduce a 2x gain which is very beneficial in low voltage measurements. Some analog front ends, such as that designed for the dSPACE high-resolution system, use instrumentation amplifier configurations. These have several advantages. The analog inputs have a very high input impedance. The A/D differential amplifier configuration input impedance is  $R_{IN} = 4.02 \text{ k}\Omega$ . The instrumentation configuration also does not require perfectly symmetric matching of components. Resistor matching for the differential amplifier is tolerable with 0.1% resistors but becomes difficult for the anti-aliasing capacitors. Instrumentation amplifiers are available as integrated devices however the IC gain is generally restricted to being greater than or equal to one. Alternatively, three individual operational amplifiers can be used as was shown in the dSPACE high-resolution design in Chapter 2.

A distinct benefit for the used differential amplifier configuration is that only a single supply is required for bipolar operation. The AD7760 is designed to convert

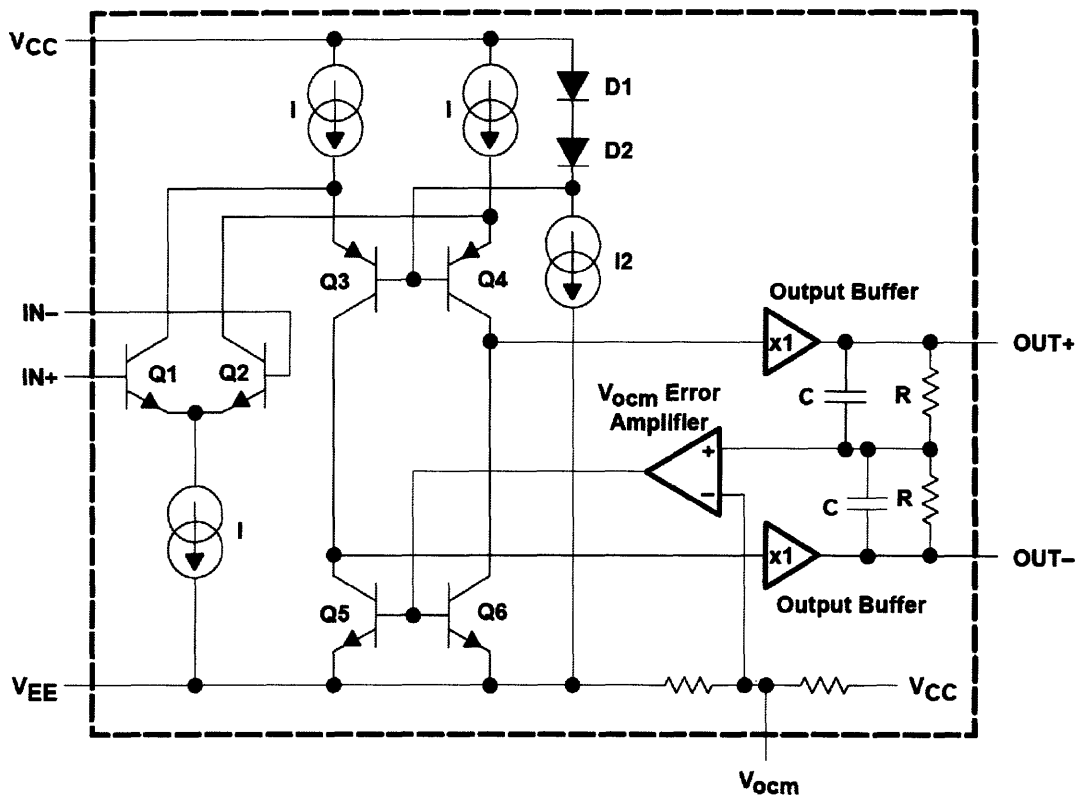


Figure 4-5: Simplified fully differential amplifier internal circuitry [8].

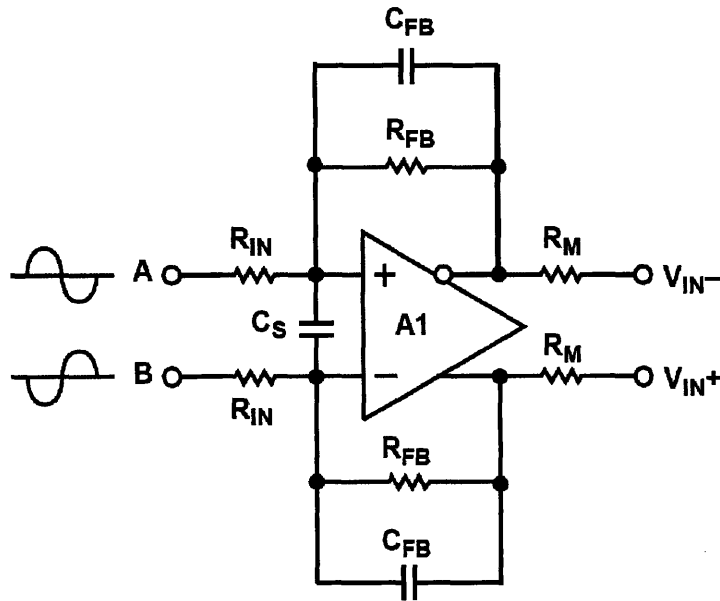


Figure 4-6: Fully differential amplifier with scaling and anti-aliasing components.

voltages between 0.410 V and 3.685 V which are centered around 2.048 V. The differential amplifier is able to convert a  $\pm 10$  V bipolar signal to this required range with only a single supply and appropriate feedback attenuation. An instrumentation amplifier would require dual supplies for the input buffer stage of the amplifier. This introduces additional power regulation components, cost, space, and complexity. However, a separate instrumentation amplifier probably would have given better noise rejection, as well as having allowed better board layout for stray impedances. I selected the differential amplifier configuration because an amplifier was already available built into the AD7760 IC and the evaluation board design used this method.

External feedback components used in this design form a third-order anti-aliasing filter in addition to scaling the signal. Figure 4-6 shows the configuration used in the analog front end design where A1 is built into the AD7760 IC. The voltage labels are referenced to the A/D converter; hence  $V_{IN}$  is the output of the differential amplifier. Voltages  $A$  and  $B$  are the amplifier differential input voltages.

The transfer function equations are generally straightforward without anti-aliasing poles, and get more complex with each additional order. The indicated symmetric

feedback do not drive the amplifier unstable because both feedback paths form negative feedback loops. This is so because the inverting input terminal is connected to the non-inverted output terminal through  $R_{FB}$ , and vice versa. It is important to understand the fundamental circuit equations before considering anti-aliasing configurations. First, assuming only  $R_{FB}$  and  $R_{IN}$  are used, the amplifier equation is

$$V_{IN+} - V_{IN-} = G(A1_+ - A1_-) \quad (4.2)$$

where  $G$  is the amplifier gain. By using superposition and considering the resistor network as voltage dividers, the input node equations are

$$A1_+ = A \left( \frac{R_{FB}}{R_{IN} + R_{FB}} \right) + V_{IN-} \left( \frac{R_{IN}}{R_{IN} + R_{FB}} \right) \quad (4.3)$$

$$A1_- = B \left( \frac{R_{FB}}{R_{IN} + R_{FB}} \right) + V_{IN+} \left( \frac{R_{IN}}{R_{IN} + R_{FB}} \right) \quad (4.4)$$

By substituting Equations 4.3 and 4.4 into Equation 4.2,

$$V_{IN+} - V_{IN-} = (A - B) \frac{R_{FB}}{\frac{R_{IN} + R_{FB}}{G} + R_{IN}} \quad (4.5)$$

Here,  $GR_{IN}$  is assumed to be much greater than  $(R_{IN} + R_{FB})$ . This gain expression then simplifies to

$$G_1 = V_{IN+} - V_{IN-} = (A - B) \frac{R_{FB}}{R_{IN}} \quad (4.6)$$

The differential circuit gain then becomes simply  $\frac{R_{FB}}{R_{IN}}$ . Matching external components here is extremely critical for good common mode rejection. The use of 0.1% resistors in the implemented design result in a 60 dB common mode rejection ratio (CMRR). The CMRR could be significantly improved with an instrumentation amplifier.

A first-order anti-aliasing filter is considered next. Referring to Figure 4-6, capacitors  $C_{FB}$  are now included in the analysis. By using the equivalent impedance for the feedback path, the ideal (high gain) transfer function becomes

$$G_2(s) = \frac{R_{FB}}{R_{IN}} \frac{1}{R_{FB}C_{FB}s + 1} \quad (4.7)$$

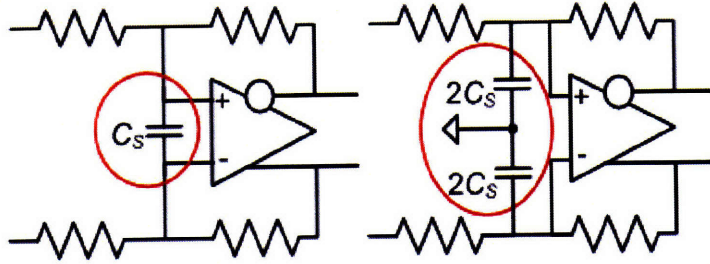


Figure 4-7: Half-circuit analysis of symmetric fully differential amplifier with shunt capacitor. The single capacitor (left) between differential signals can be replaced by two capacitors (right) referenced to ground.

where the DC gain is set by the resistor ratio as in a typical inverted amplifier configuration and a low-pass RC filter is created between the feedback components. A second pole is included by accounting for the shunt capacitor  $C_S$ . The second-order transfer function then becomes

$$G_3(s) = \frac{R_{FB}}{R_{IN}} \frac{1}{R_{FB}C_{FB}S + 1} \frac{1}{2R_{IN}C_S S + 1} \quad (4.8)$$

The second pole location is half the expected frequency because it is acting between differential signals. To maintain the same breakpoint frequency,  $C_S$  could be replaced by two capacitors of  $2 \times C_S$  each. This is shown in Figure 4-7. Each differential signal would have a single capacitor referenced to ground and the symmetric circuit can be analyzed with only a half-circuit. Some claims are made that this actually gives better common mode noise rejection [8]; however the opposite was demonstrated in characterization of the A/D designs for the high-resolution dSPACE system. Recalling the asymmetric noise distribution of the high-resolution A/D PCBs in Chapter 2, a single capacitor between the differential signals decreased the RMS noise compared to two capacitors referenced to ground.

Finally, a third pole is added at higher frequency with  $R_M$  in concert with the input capacitance of the AD7760 A/D converter, as well as stray capacitances. These resistors set the impedance between the differential amplifier and the A/D converter. The analog input capacitance for the AD7760 IC is listed in the datasheet as  $C_{AD} =$



55 pF. There will also be parasitic capacitance on any PCB layout. The parasitic capacitance between signal traces on the PCB is estimated as  $C_P = 0.5 \text{ pF}$ . These two capacitances paired with  $R_M$  creates the pole. This produces an overall transfer function of

$$G_4(s) = \frac{R_{FB}}{R_{IN}} \frac{1}{R_{FB}C_{FB}s + 1} \frac{1}{2R_{IN}C_Ss + 1} \frac{1}{2R_M(C_{AD} + C_P)s + 1} \quad (4.9)$$

The AD7760 IC has a built-in fully differential amplifier. The datasheet claims all specifications with the differential amplifier as part of the analog input signal path. This reduces external components and minimizes layout considerations. The common mode voltage,  $V_{OCM}$ , is not shown in Figure 4-6.  $V_{OCM}$  is set internally to the AD7760 as  $\frac{1}{2}$  the analog reference voltage, or  $V_{OCM} = 2.048 \text{ V}$ . This sets the midpoint of the A/D converter and allows for the greatest range to be achieved in the sigma-delta conversion.

The input voltages are scaled to utilize the range of the A/D converter about the shifted common mode voltage. The scaling is calculated as shown in Equation 4.1. Through discussions with an Analog Devices product applications engineer, recommended values for all components were given as in Table 4.1 where  $R_{IN}$  can be changed for different input ranges [39]. Fully differential inputs of  $\pm 10 \text{ V}$  are designed for, which sets  $R_{IN} = 4.02 \text{ k}\Omega$ .

Table 4.1: Differential Amplifier Component Values

$V_{OCM}$	$R_{IN}$	$R_{FB}$	$R_M$	$C_S$	$C_{FB}$
2.048 V	4.02 k $\Omega$	649 $\Omega$	17.8 $\Omega$	2.2 pF	33 pF

The dynamics of the differential amplifier itself are expected to be at higher frequencies than the anti-aliasing filter and are thus neglected. Specifications for the integrated differential amplifier are not given in the AD7760 IC datasheet. The low-pass pole locations are given in Table 4.2. The frequency response of this transfer function is shown in Figure 4-8. The recommended design by Analog Devices places the first two poles at 7.43 and 36.2 MHz. The presented design with breakpoints at

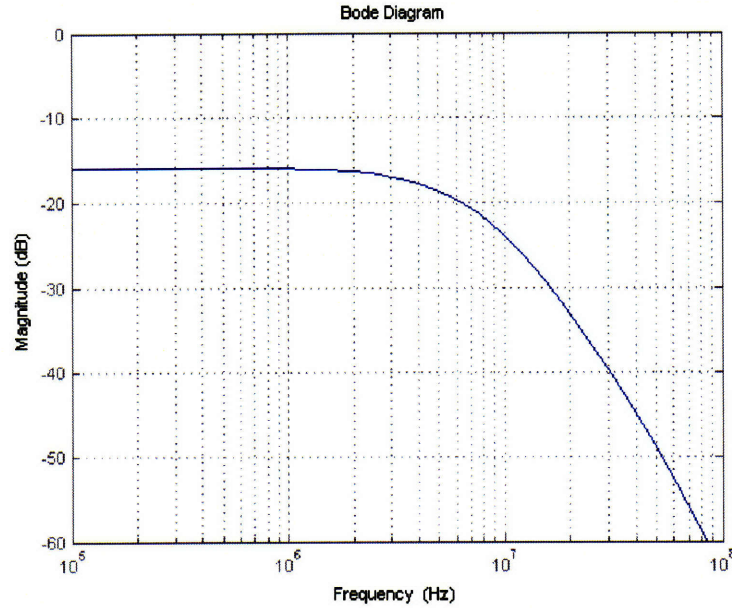


Figure 4-8: Analog input differential amplifier anti-alias configuration frequency response.

7.43 and 9.00 MHz will therefore have more roll-off than the recommended design. Because of the pipeline delay of sigma-delta converters, they are not expected to be used with extremely high control loop rates. Therefore, the additional attenuation gained by lower frequency poles is not expected to significantly affect the loop dynamics in any applications of this design. If this were considered to be an issue,  $C_S$  could be replaced with a 0.55 pF capacitor to maintain the same relative attenuation at the recommended breakpoint frequencies.

Table 4.2: Analog Input Differential Amplifier Anti-alias Pole Locations

Pole	Pole Location [ns]	Pole Location [MHz]
$R_{FB}C_{FB}$	21.42	7.43
$2R_{IN}C_S$	17.69	9.00
$2R_M(C_{AD} + C_P)$	1.98	80.55

Almost as important as analog design and component selection is the physical layout. The designed PCB is based directly on the evaluation board for the AD7760.

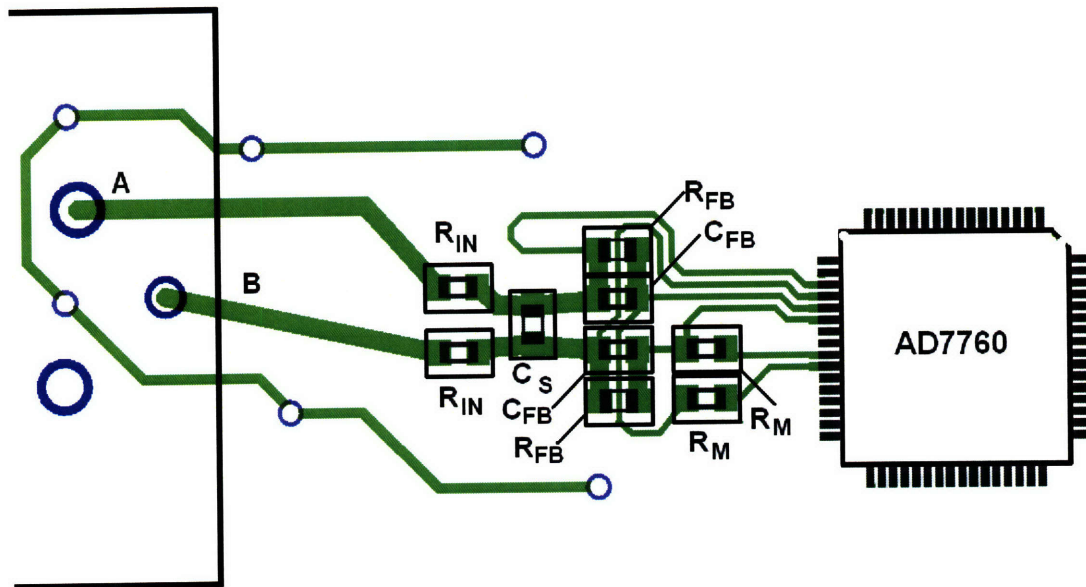


Figure 4-9: Analog input PCB layout.

This includes locating all components close to the AD7760 IC and using symmetric routing of traces on both sides of the differential amplifier. The layout is shown in Figure 4-9. A grounded shield path is included around the traces between the analog input connector and the differential components to reduce surface charge leakage currents. The 0405 and 0603 surface mount device (SMD) packages are used for all critical components located close to the AD7760 IC in order to save surface area. An XLR audio connector is used to bring the analog signals onto the PCB. This allows for low cost, pre-terminated twisted pair cables within a third shield conductor. The shield encapsulates the twisted pair through the connector and onto the PCB. The cable shield is tied back to the source and not tied to the PCB ground, in order to prevent ground loops. Additionally, charge coupled electrostatically to the shield conductor drives currents in the source common as opposed to the A/D PCB common where it could introduce signals on the sensitive analog input.

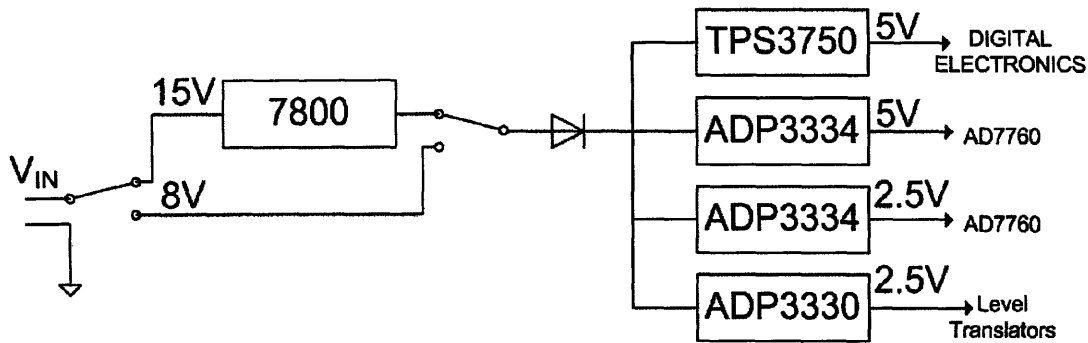


Figure 4-10: PCB voltage regulation.

## 4.2 Power Regulation and Decoupling

The AD7760 requires both 2.5 V and 5 V supplies while the rest of the PCB is designed to operate on 5 V. The digital outputs of the AD7760 IC are at 2.5 V and thus level translators are required to step the voltage up to 5 V to interface with the microcontroller and be interfaced to the external FPGA. In general, the recommended supporting circuitry design and devices given for the evaluation PCB were used, however there are differences due to additional requirements, as described below.

A simplified schematic of the voltage regulation for the PCB is shown in Figure 4-10. There are two options for the input voltage so different power supplies can be selected. Since 15 V supplies are commonly available, and is the same voltage supplies as used by the ADE capacitive probe drivers. Maintaining low power is not a design requirement for this PCB, so a linear regulator is acceptable for dropping the voltage. Linear regulators are selected versus switching converters because there is much less high frequency noise content. The input voltage is user selected by two jumpers. Alternatively an 8 V input can be used. Separate voltage regulators are dedicated to the AD7760 in order to follow the recommended design and limit the influence of digital electronics on the analog converter. This provides the best opportunity to match the datasheet performance specifications.

The ADP3334 ICs are adjustable regulators. The voltage is defined by a voltage

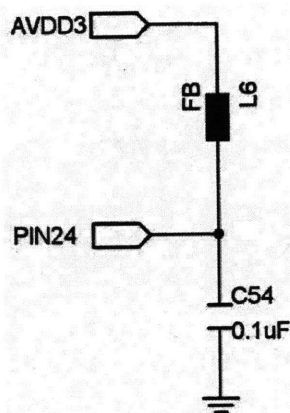


Figure 4-11: Example of AD7760 supply decoupling.

divider resistor network. The voltage absolute accuracy is only as good as the resistor tolerances and temperature coefficients but this is not critical. The AD7760 IC uses a precision voltage reference for the conversion process so a wide supply voltage window does not affect performance. The remaining two regulators are fixed voltage.

The voltage regulators also have extensive decoupling and each IC on the PCB is bypassed with at least a  $0.1 \mu\text{F}$  capacitor close to the supply pin with the other end near the IC ground pin. This ensures that return currents are flowing to the correct pin. The AD7760 datasheet recommends specific decoupling for each supply pin [2]. An example of this is shown in Figure 4-11. *FB* refers to a ferrite bead. These are used to suppress high frequency noise. These beads create inductance and have high impedance at high frequencies. These sub-circuits are located as close to the AD7760 IC as possible.

Also included on all AD7760 IC supply pins and other regulated lines are EMI suppression filters. An equivalent circuit is shown in Figure 4-12. These are 3-terminal capacitors with integrated ferrite beads to minimize resonance with surrounding circuits.

Typically the analog and digital ground planes are isolated from one another and only connect at the external ground connection. This is not done on the PCB because there is very limited space to route multiple ground planes. It is important to minimize the trace length of any ground pin to the ground plane. Analog and

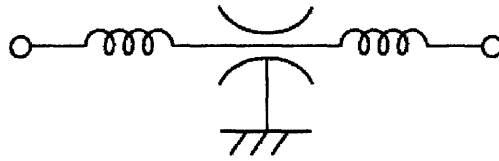


Figure 4-12: EMI suppression equivalent circuit.

digital ground pins from the AD7760 IC do not share return paths to the ground plane with any other pins. The PCB is manufactured as a 4-layer board where the second layer is a ground plane. This reduces ground loop affects and aids in heat dissipation. This is critical with surface mount devices as the ground plane has the lowest thermal resistance and can dissipate the most power. Roughly 50% of the AD7760 IC is an exposed paddle for dissipating power. If further PCB revisions were to be constructed, an interesting experiment would be to separate the analog and digital ground planes by placing the analog ground plane on the third layer and comparing performance measurements. The AD7760 evaluation board uses a single ground plane and thus that example was followed. A comprehensive discussion on grounding in mixed analog/digital systems is presented in [1, 40].

### 4.3 Digital Interface and Microcontroller Design

This section describes the selection and design of the digital interface to the AD7760 IC and the external sampling FPGA. This includes digital isolation, level shifting, microcontroller design, and multiple channel synchronization. The associated components are shown in Figure 4-13. The external interface connector is digitally isolated from the PCB signals, supplies, and ground through digital galvanic isolators. A microcontroller initializes the AD7760 IC and monitors the A/D operation. Level shifters convert between the 5 and 2.5 V logic necessary for the AD7760 IC.

The AD7760 IC has internal registers that control sampling rate and digital filtering. These registers are accessed through a bidirectional data bus. The AD7760 is interfaced with two control lines, the 16-bit data bus, and several other lines for

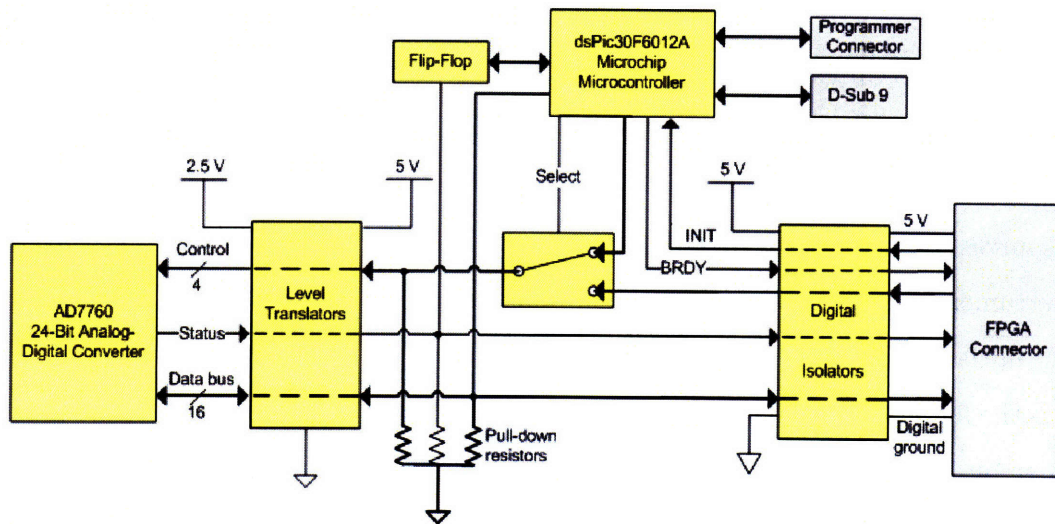


Figure 4-13: AD7760 PCB digital interface and components.

initialization. This is documented in the AD7760 datasheet [2]. The  $\overline{RD}/WR$  line controls the direction of the data bus. The chip select,  $CS$ , low pulse either sets or reads out the appropriate register on the rising edge depending on the  $\overline{RD}/WR$  status. Other control lines are unidirectional and include a clock source  $CLK$ , reset  $RST$ , and synchronization  $SYNC$ .

Chapter 2 discussed the importance of digital galvanic isolation. The external interface connections are all passed through high-speed isolators in order to ensure that potential ground loops are broken. A high density D-sub connector is used with twice the number of connections necessary so each signal is transmitted as a twisted pair with either a ground or source voltage. This source voltage drives the external side of the digital isolators. The internal side of the isolators is driven by the 5 V PCB digital logic voltage. Separate voltage sources are required for both sides of the digital isolation ICs or else there would be no isolation. Care is taken to reduce ground coupling. For example, the PCB ground plane is broken below the connector and the digital isolators. The isolators are 4-bit unidirectional units. Four ICs pass the data bus to the external connector, one IC passes two outgoing signals, and one IC passes incoming signals. Because the ICs are unidirectional, an

additional four ICs would be required to utilize a bidirectional data bus with the FPGA and retain galvanic isolation. At approximately \$9 per IC and the additional footprint associated with each SOIC-16 package, it was determined that AD7760 IC initialization should be completed locally on the PCB. Another benefit is that FPGA resources are not lost to a one-time operation. Upon initial PCB design, the National Instruments platform was not selected and options remained open, so incorporating the initialization onto the PCB also reduced the FPGA design requirements. This was the largest factor in deciding to use a local microcontroller. An additional benefit to an on-board microcontroller was the possibility to use it for data sampling during the testing phase.

The AD7760 outputs data at 2.5 MHz, however it is outputted as two 16-bit words. A microcontroller by Microchip [41] was selected to be implemented because our laboratory has programming hardware and personal experience with that architecture and programming language. This brand is known for quick implementation and is designed for hobbyists as well as industry. The device architecture, programming environment, and language has a relatively fast learning curve but also provides functionality appropriate for industrial applications. Competing devices are made by Atmel, Freescale, and Parallax. A full DSP from a company such as TI would be faster but more expensive, as well as require a more intimate design knowledge for this simple application. From the line of Microchip microcontrollers, the dsPIC6012a was selected because it has a 16-bit data path for all operations and operates at the highest rates available. The maximum cycle rate is 30 million instructions per second (MIPS). This allows 13 instruction cycles per 400 ns, which is the data update rate of the AD7760 IC. This is enough instructions to sample the AD7760 in real-time and fill the available RAM. The dsPIC6012a also has sufficient I/O for easy implementation and a full DSP library for any manipulation if that were deemed necessary. This line runs on 5 V like the NI FPGA board, as opposed to 3.3 V.

The complete microcontroller code is presented in Appendix B. The code is primarily written in C because it is more intuitive to a secondary user and quicker to implement high-level functionality such as UART communication. Initial microcon-



troller code development was completed without a fully functioning PCB. This was possible by using DIP components and developing the code structure. The final code structure is shown in Figure 4-14. There is a general setup of the I/O pins and peripheral devices when the program is initially run. This setup defines the sub-function macros and sets configuration registers for the microcontroller hardware. Three interrupts are used. A UART interrupt receives RS-232 commands from a host computer. The PCB has a serial RS-232 level translator and serial D-sub 9-pin connector. Within the interrupt macro, the received byte is read into a register and a case structure determines which action to respond with. This was typically used as a debugging process to ensure that the PCB was properly built and soldered, and that IC interacted as expected. This mode was also able to acquire a limited number of samples to output to the host. The host commands and interface can be accessed through Matlab.

Another interrupt is a timer, which is configured as a 32-bit timer and decimates the cycle clock so that an interrupt is generated approximately every 200 ms. An external flip-flop is triggered by a data ready pulse, *DRDY*, from the AD7760 IC. The status of this flip-flop is read on the timer interrupt and reset. Based on the status of the flip-flop, the board ready, *BRDY*, line is updated to the FPGA. This process ensures that the AD7760 IC is operating. If the A/D is not operating, this section is where the A/D IC would be reinitialized. This is not included in the current code but could be with several lines. Monitoring of the *DRDY* pulse could also be completed by the FPGA because it uses the *DRDY* pulse to acquire the sample data anyway. However, it was considered more flexible to locate this monitoring on the microcontroller and again save FPGA resources. A much more simple FPGA loop monitors the *BRDY* signal.

One of the signals coming in from the external FPGA is the initialization trigger *INIT*, which is an edge-triggered interrupt. Upon the interrupt, the microcontroller enters the initialization function. This can also be reached by a UART command. Outputs, which are timing-critical, are written in assembly language so as to be optimized for speed. The AD7760 initialization sequence follows that described in

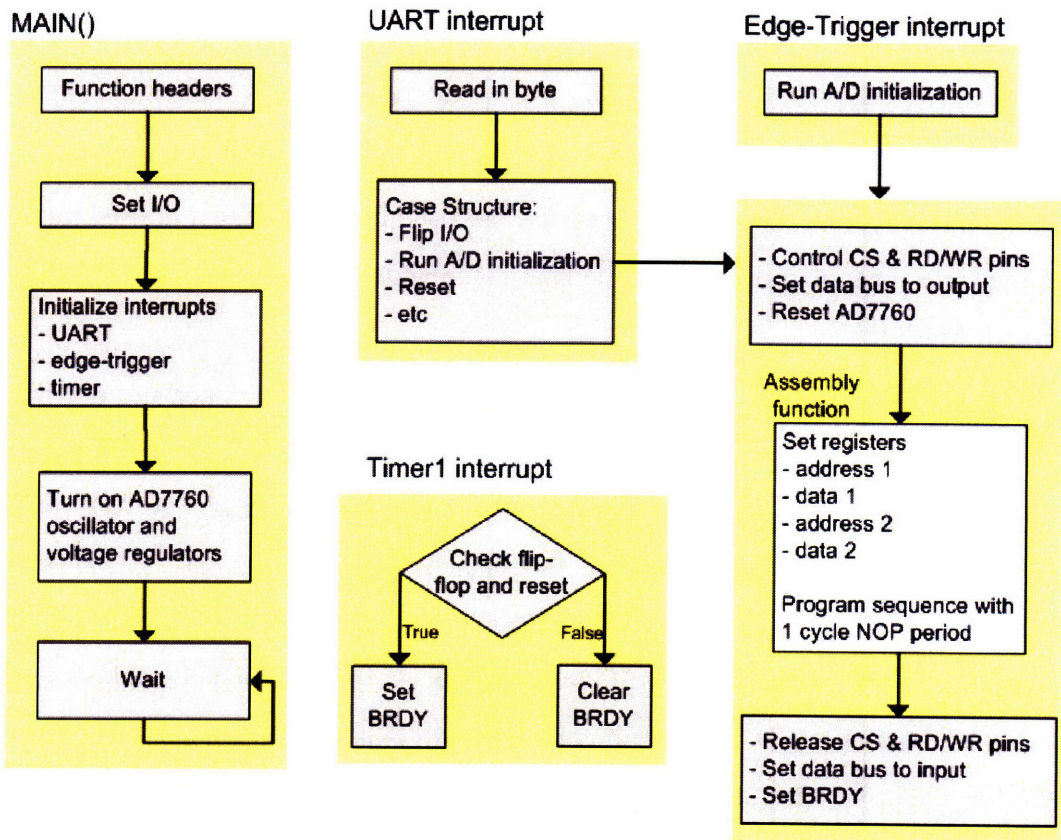


Figure 4-14: Microcontroller code block diagram.

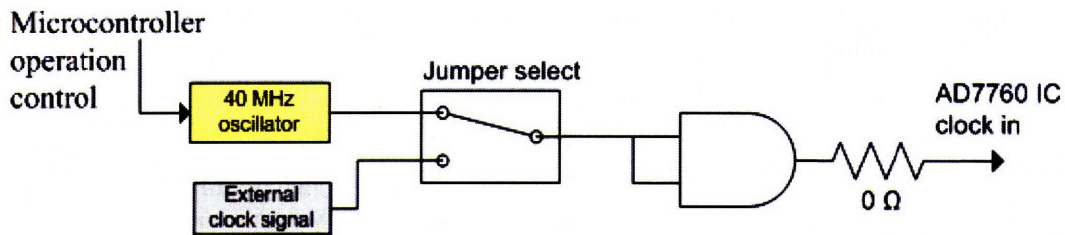


Figure 4-15: AD7760 PCB clock management block diagram.

the datasheet [2]. Two 16-bit registers are loaded into the AD7760 IC. One PCB feature is a built-in digital switch on the PCB that allows the microcontroller to command the  $\overline{RD}/WR$  and  $CS$  control lines as opposed to the FPGA. When the AD7760 is fully programmed, the digital switch is released back to the FPGA and the microcontroller sets the board ready  $BRDY$  line, and then returns to a wait mode in the main function. Early in the design phase, the microcontroller entered sleep mode to save power. This would be reasonable if all interrupts were turned off so it would not come out of sleep, but the PCB is intended to be able to be re-initialized or re-synchronized to allow more flexibility for how it is operated from the FPGA. I found that the microcontroller coming out of sleep mode would introduce very large, high frequency disturbances on the sensor channels. The sleep functionality was then removed. This is described further in Section 6.1.

The clock management is shown in Figure 4-15 with a simplified block diagram. There is a clock source built onto the PCB, which is a standard 40 MHz oscillator selected to meet the AD7760 clock jitter requirements. A jumper is also used to select either the on-board clock source or an external clock source. This allows the clock source to be shared between multiple PCBs so they may be synchronized together. From the jumper, the clock signal passes through an AND gate to clean up the edges. This is especially critical for external signals brought onto the board. The AND gate is located at this point in the signal path so that a clock source experiences the same time delays whether it is internal or external. From the AND gate the signal is passed into the AD7760 IC. A  $0\ \Omega$  resistor indicates that the signal path should be short and of very low impedance.

An important issue in synchronization between multiple channels are time delays experienced in parallel signal paths. The propagation delay of a signal along a wire can in principle be accurately modeled, however a conservative assumption is a delay of 1 ns/ft. For PCBs located in close proximity, a conservative estimation for the wire length is 1 ft, or a 1 ns delay. If the delay becomes greater than 12.5 ns, then the data measurements can be only synchronized to within 25 ns as opposed to  $\ll 1$  ns. A possible solution to long transmission lines is to bring the clock source off a PCB for all A/D channels being used, including that of the clock source. This means there would be a transmission line to all channels and each PCB would have the jumper set to external. With equal length lines this would ensure equal propagation delays. In this case it is important to use shielded cable to reduce interference. I took no considerations for transmission line termination in this current design. It would be possible to match termination impedance or add more clock management hardware in a future revision, however this would also increase the complexity.

Because the AD7760 operates on 2.5 V digital communication, a series of bidirectional level translators are used to convert the data bus and control signals to/from 5 V. They need to be bidirectional because the 16-bit data bus needs to be driven to boot-up the AD7760 initially while it is driven by the AD7760 when outputting data samples. A parallel data path, as opposed to a serial interface, is necessary due to the high sampling rate and large number of bits. While the selected level translators (Analog Devices ADG3308BRUZ) are the most applicable in terms of number of bits, speed, and selectable voltage levels, there is little documentation as to the required surrounding circuitry [9]. For example, the maximum current output is not listed. The datasheet only states that the output is intended for CMOS-compatible loads and buffers should be used for current-driving capabilities.

In this application, control of the data bus while sampling is only taken when the AD7760 is outputting data. The data bus is left floating with high source impedance for the other two-thirds of the sampling period. For a bidirectional level translator, simplified in Figure 4-16, a source on one side is required. Otherwise any charge build-up can cause the one-shot generator to drive the line. The one-shot generator is

a monostable multivibrator for creating fast switching characteristics. The one-shot generator output then in turn creates a charge at the other level. This process repeats itself and the line oscillates at the highest rate possible. This sinks a large amount of current which can either affect the voltage regulator by reducing the supplied voltage or it can disrupt the other level translator bits.

To avoid this problem, I used pull-down or pull-up resistors to provide a source when nothing else is controlling the lines. However, the level translators can only source a fixed amount of current, thus limiting their operation on additional pins when at this current limit. Datasheet specifications are not provided on sinking versus sourcing current capabilities. Several different resistor values were tested to find the suitable resistor value of 56 k $\Omega$ . A user selectable jumper determines whether the resistors are pull-up or pull-down. CMOS devices can commonly sink more current than they can source, however notable performance increases with either configuration were not documented. Currently the resistors are in the pull-down configuration. This issue presented difficult debugging challenges due to the lack of documentation. I found that as the analog input voltage level would change, certain data bits from the A/D would be unstable and result in erroneous data readings on the FPGA. This issue could be attributed to a number of components along the data bus, FPGA connector or breakout cabling, or even PCB construction. Only with a diligent debugging process was the issue tracked back to the pull-down resistor value. The debugging process was additionally slowed because all components on the PCB are surface mount devices.

## 4.4 PCB Construction and Debugging

The PCB is constructed on a 4.75  $\times$  6 inch 4-layer PCB from Advanced Circuits [42] for \$66 per board. There are 168 surface mount components located on the PCB. These were soldered by hand. This proved to be extremely tedious and required the use of soldering paste, a rework/desoldering station, and a 20x/40x microscope. It takes approximately 16 hours per board to hand-solder all components assuming there

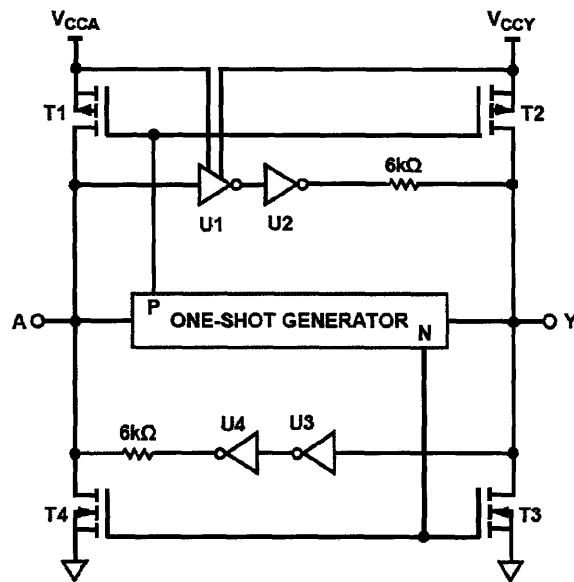


Figure 4-16: Level translator simplified schematic [9].

are no defaults or shorts/missed connections. Four complete PCBs were constructed in this fashion. This was beneficial to gain an intimate knowledge of the board layout and to debug issues throughout the development process. With a proven design however, Advanced Circuits is also able to do the component assembly for an additional \$180 per board for ICs that are difficult to solder by hand. This includes the AD7760, microcontroller, level translators, as well as several others. The remaining components could then easily be soldered by hand with little worry of error. The tightest IC pin spacing is found on the AD7760 and is 0.007 inches, or 179  $\mu\text{m}$ .

Several features were changed from the AD7760 evaluation board. The most troublesome issue was the level translators instability. It is possible that the evaluation board accounted for pull-up or pull-down resistors within another component that would control a data line when nothing else was controlling it, however this was not demonstrated in the available design documentation. Other features, such as a 2.5 V voltage monitoring IC, were removed because they were considered redundant with other features added to the design. Lastly, features such as the on-board microcontroller and digital isolation were added.

In Chapter 3, I discussed why it was necessary to design our own A/D PCB to

achieve high-resolution and high-speed acquisition and control. This chapter presented the A/D PCB design. The following chapters describe how the A/D PCB is interfaced with acquisition and control software in the overall digital platform, as well as experimental results. The A/D PCB design shown here is operational and achieves the specifications stated in the AD7760 datasheet.





# Chapter 5

## LabVIEW Control Software

The National Instruments (NI) LabVIEW FPGA Module and the FPGA hardware allows one to create custom I/O and control hardware without prior knowledge of traditional HDL languages or board-level hardware design. The NI LabVIEW FPGA Module uses the LabVIEW graphical development platform to directly compile applications onto FPGA hardware, thus allowing applications to be written for a number of platforms, such as an embedded real-time processor or FPGA, with little modification required between systems.

As with any real-time system, computing precision needs to be considered. Traditional DSP platforms allow double precision, or 64-bit floating-point precision. In order to decrease the number of clock cycles required to complete a full computation, shorter variable definitions are used. These include long integer, short integer, single floating-point precision, character, and logical that range from 1 bit to 32 bits. For efficient computations, it is critical to evaluate the required precision versus time to compute. Fixed-point precision is introduced by scaling decimal number to integers and allowing strictly integer computations. Rational numbers within a user-specified range and with a user-specified precision are represented as integers by scaling variables by the same number of bits  $n$ . The result is then post-scaled by  $-n$  bits to obtain the equivalent decimal value following the computations. A fixed-point number can be specified any size between 1 and 64 bits, inclusive, and as signed or unsigned within the NI language.

FPGA operations are physically based on reconfigurable logical gates along with more complex combinational functions. Floating-point precision is not possible except in advanced, proprietary intellectual property (IP) cores which compensate for the simplicity and precision with speed and utilized resources. The LabVIEW FPGA Module implements fixed-point numbers as the most complex definition. Boolean operations are preferred because these synthesize the most easily to FPGA logic. While the FPGA operations are scaled back from the general LabVIEW platform, additional hardware interface features are available. These include a host interface and first in, first out (FIFO) buffers. The host interface allows for triggering, generally used to initiate data transfer, processor computation, and data return for loops that require complex computations with the dedicated RTOS processor in the loop. Direct memory access (DMA) is also available which allows the FPGA hardware to directly access system memory without occupying processor resources. FIFO buffers are used for communication within the FPGA to store and transfer data from one control loop to another, generally at different loop rates.

Figure 5-1 shows the general relationship between generating the software applications and the implementation on hardware. The FPGA virtual instrument (VI) is created on the host/development computer and then compiled to a bitfile. Within the compilation process, the graphical LabVIEW FPGA code is first translated to text-based VHDL code. Additionally, timing constraints are applied to the circuit design. The Xilinx ISE compiler tools are then invoked. The VHDL code is optimized, the logic is reduced, and the gate array configuration is synthesized. This stage contains the detailed hardware information is implemented. Logic synthesis locates the logic blocks and routes interconnects. A final timing verification tests for expected errors. The output of this stage is the bitfile. The NI FPGA hardware has a base clock of 40 MHz and can be derived to other rates from 10 MHz to 200 MHz by a PLL. The loop rates defined in the VI are applied to the constraints in the synthesis stage. Compiling the FPGA code requires many computational resources to complete. A new host/development computer was purchased to satisfy the minimum hardware requirement for the FPGA Module software. This included 2 GB of RAM and a 2+ MHz multicore

processor. Even with a high-performance computer, synthesis can take 60 minutes for a bitfile that only utilizes 25% of the 3M gate FPGA. Once compiled, loading the actual bitfile to the FPGA board takes a matter of seconds.

The host VI loads and initiates the FPGA bitfile on the hardware and interacts with data I/O as necessary. During testing this included setting variables to optimize delay times and implement different filters without recompiling. The final implementation initiates the 1 second data acquisition, saves the data to a file, and displays data as well as characteristic measurements of it for the user. The final program is compiled to the real-time operating system of the embedded controller. From Figure 5-1, the host computer displays the GUI by communicating over an ethernet connection.

## 5.1 High Level Layout

Digital systems operate in a discrete domain as opposed to a continuous domain. Figure 5-2 shows the configuration used in the implemented system. This differs from a continuous system because it only acts on discrete samples  $kT$ , where  $k$  is any integer. This block diagram is more general than the actual implementation because the output sample rate is decimated from the input sample rate. The operation and PCB design of the A/D has been described in previous chapters. The reference signal is generated within the FPGA or real-time embedded computer.

Figure 5-3 shows the functional elements of the FPGA application. The primary advantage of FPGAs are the high-speed parallel processing capabilities. This allows parallel loops of varying complexity and varying loop rates. Figure 5-3 represents only a single control channel. The experimental hardware results operate with two simultaneous, parallel loops.

While the A/D outputs data at 2.5 MHz, the D/A can only output samples at 625 kHz. This means that the data stream must be downsampled to an appropriate frequency. Several approaches can be taken to downsampling the data. A discrete data sequence can be defined as integer samples of period  $T$  from a continuous signal

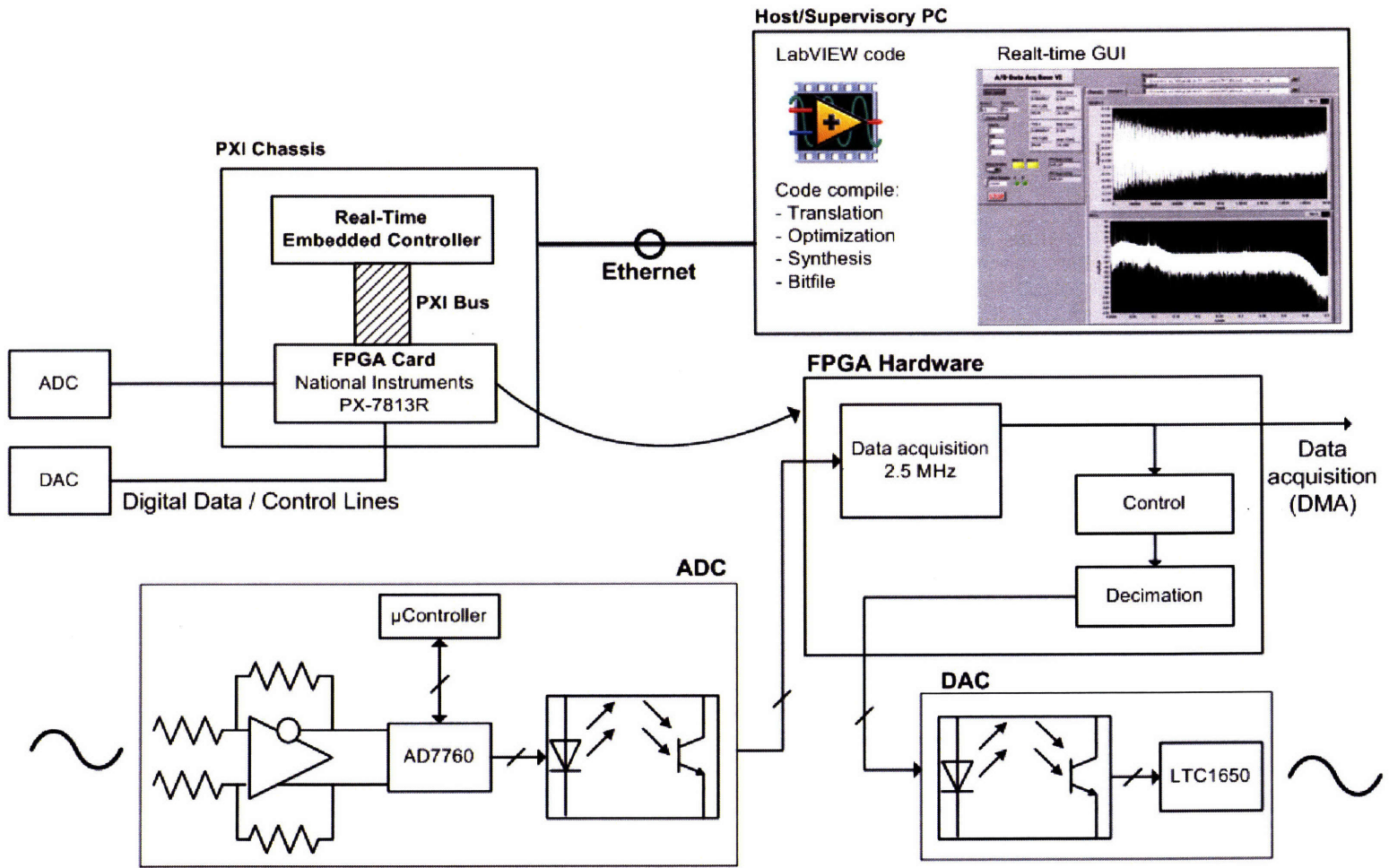


Figure 5-1: FPGA high level layout.

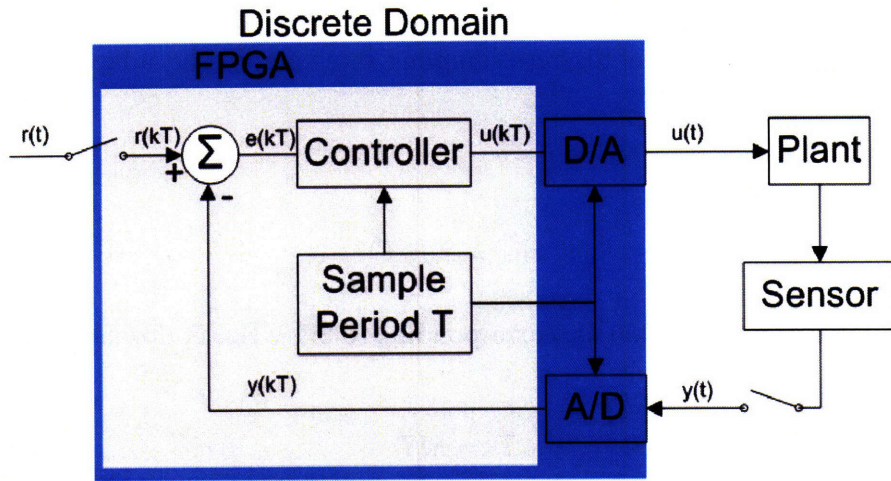


Figure 5-2: Digitized control platform block diagram.

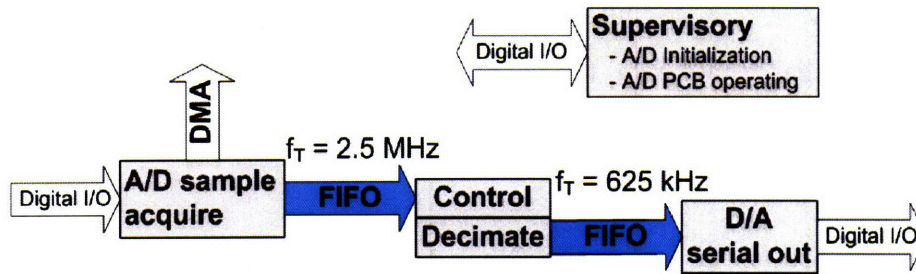


Figure 5-3: FPGA functional elements.

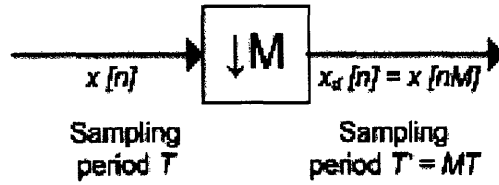


Figure 5-4: Block diagram of discrete sampling rate compressor, adapted from [10].

by

$$x[n] = x_c(nT) \quad (5.1)$$

In this case the period  $T$  is 400 ns, corresponding to 2.5 MHz. A downsampled period  $T'$  is then defined by

$$T' = MT \quad (5.2)$$

where  $M$  is an integer decimation factor. The new data sequence is

$$x_d[n] = x[nM] = x_c(nT') = x_c(nMT) \quad (5.3)$$

Integer  $M$  in Equation 5.3 is defined as a sampling rate compressor [10]. The block diagram of this system is shown in Figure 5-4.

For a bandlimited continuous signal, it is important to ensure no aliasing occurs. Aliasing occurs when the downsampled signal set drops below twice the continuous signal frequency. A low-pass filter must precede the compressor to ensure frequency content is not aliased into  $x_d[n]$  for an arbitrary continuous signal that is not aliased in  $x[n]$ . To avoid aliasing when downsampling by a factor of  $M$  in the frequency domain requires that

$$\omega_N < \frac{\pi}{M} \quad (5.4)$$

The AD7760 has a signal bandwidth of 1 MHz but the control bandwidth of interest is approximately 10 kHz due to the phase loss from the large group delay. In Figure 5-3, data acquisition through the DMA occurs before any digital filtering or decimation and thus acquired data is only bandwidth limited to 1 MHz. The control implementation is discussed further in Section 5.4.

## 5.2 A/D Acquisition Interface

The AD7760 datasheet specifies the timing requirements to acquire data. The 24-bit sample is acquired across a 16-bit data bus and is therefore acquired as two 16-bit words. The additional 8 bits provide A/D operation status and ensure data integrity. The timing requirements are not repeated here, although the overall A/D PCB design needs to be considered with respect to timing due to digital propagation delays. The worst case delays due to different components are shown in Figure 5-5. The delay time for signals along the cabling is estimated at 1 ns/ft. Specifications for the internal propagation delays on the FPGA board are not given and assumed to be negligible. This includes signal transmission between the I/O connector and the FPGA IC itself. The FPGA A/D acquisition loops run at 80 MHz, thus having a 12.5 ns period. For a signal generated at the AD7760 IC, such as the data ready *DRDY* signal, the estimated delay for the signal to reach the FPGA, be acted upon, and a response signal to reach the AD7760 is

$$\begin{aligned} T_D &= 2T_{TX} + 2T_{ISO} + 2T_{TRAN} + T_{FPGA} \\ &= 2 \cdot 8 + 2 \cdot 15 + 2 \cdot 5 + 12.5 = 68.5 \text{ ns} \end{aligned} \quad (5.5)$$

where  $T_{TX}$  is due to the level translators,  $T_{ISO}$  is due to the digital isolators,  $T_{TRAN}$  is due to the transmission length of external cabling, and  $T_{FPGA}$  is due to the FPGA 80 MHz loop response. The full window to acquire a valid sample at 2.5 MSPS is 400 ns and this simple sensing/responding that data is ready consumes 17% of the period. The FPGA responds by clocking chip select ( $\overline{CS}$ ) and read/write ( $\overline{RD}/WR$ ) pins. Sample data is then driven to the data bus by the AD7760 IC tristate. The propagation delay  $T_D$  is particularly important when reading in the sample data. The datasheet specifies that data access time is at most 41 ns, however when this is coupled with the propagation delay  $T_D$ , the total access time before data can be read into the FPGA is 109.5 ns. The sample reading is invalid if the sample is read into the FPGA too soon.

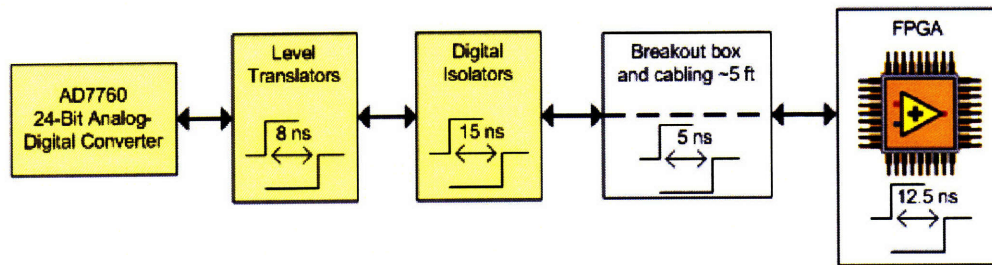


Figure 5-5: A/D to FPGA signal propagation delays.

The A/D data acquisition for each sample is completed by finite state control. A distinct advantage to the NI LabVIEW programming language is the ease of implementing sequential logic. FPGA code is most easily implemented with combinational logic, which is strictly a function of the current inputs. Sequential logic output is a function of the current input as well as previous inputs, implying that there is memory. This adds complexity not only for how to generate the memory but also the increasing logic required to sort out which state is being processed in the finite state machine.

The states of the finite state machine used here are:

- *SenseDRDY* - Detect *DRDY* signal from AD7760 IC that a data sample is ready. It is expected to arrive every 400 ns presuming the A/D is operating correctly. The A/D and FPGA use different clocks and cannot be assumed to be synchronized.
- *LowerCS* - Lower the A/D chip select pin  $\overline{CS}$ .
- *LowerRD* - Lower the read/write ( $\overline{RD}/WR$ ).
- *RaiseCS* - Raise the A/D chip select pin  $\overline{CS}$ .
- *RaiseRD* - Raise the read/write ( $\overline{RD}/WR$ ).
- *Wait* - Wait for a selected number of clock ticks. The wait period is defined by the previous state.
- *CounterSetup* - Increment or reset DMA counter. This is completed prior to DMA transfer to reduce the number of required computations in a single clock cycle.



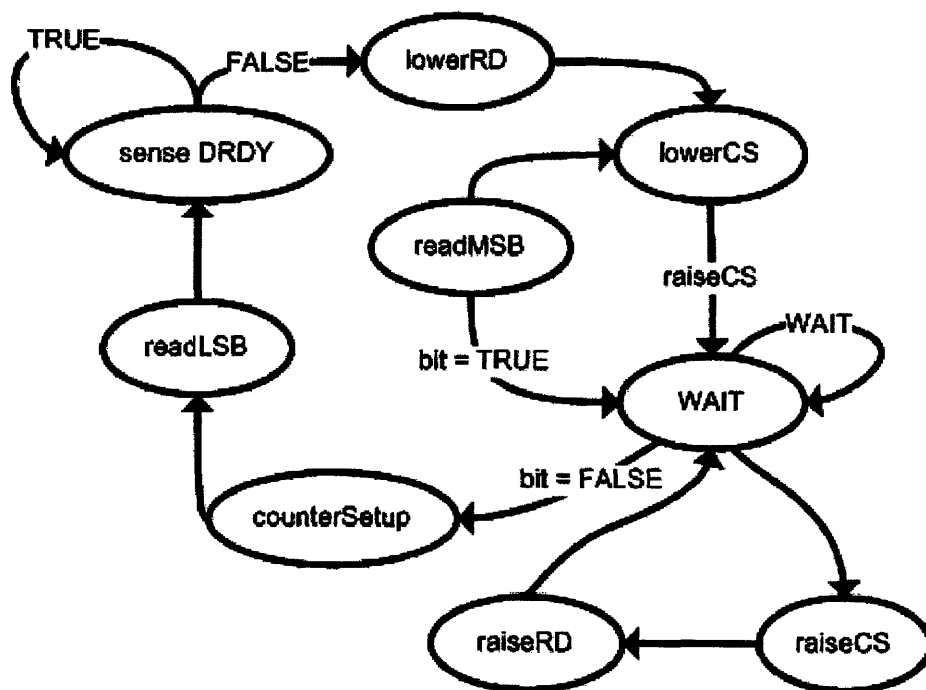


Figure 5-6: A/D acquisition state diagram.

- *ReadMSB* - Read in most significant bits (MSB). These are placed in a shift register to be combined with the least significant bits (LSB).
- *ReadLSB* - Read in least significant bits. These are combined with the MSB. Data integrity on the lower 8 bits is checked for A/D errors. The upper 24 bits are shifted from the unsigned two's complement format of the A/D output to unsigned and signed integers for transfer to the DMA and FIFO, respectively.

The finite state machine is shown in Figure 5-6. Each state completes its individual task as well as defines the next state. If the next state is *Wait*, then it also defines the wait period and post-wait state. The  $\overline{CS}$  and  $(\overline{RD}/WR)$  controls are each used twice for each A/D sample so a single bit, labeled as *bit* in Figure 5-6, is used to distinguish whether the MSB or LSB needs to be read in next. An example of the *RaiseRD* state is shown in Figure 5-7. Complete LabVIEW code for the acquisition finite state machine is included in Appendix C.

Understanding the scale factor of the A/D is essential to interpreting the data.

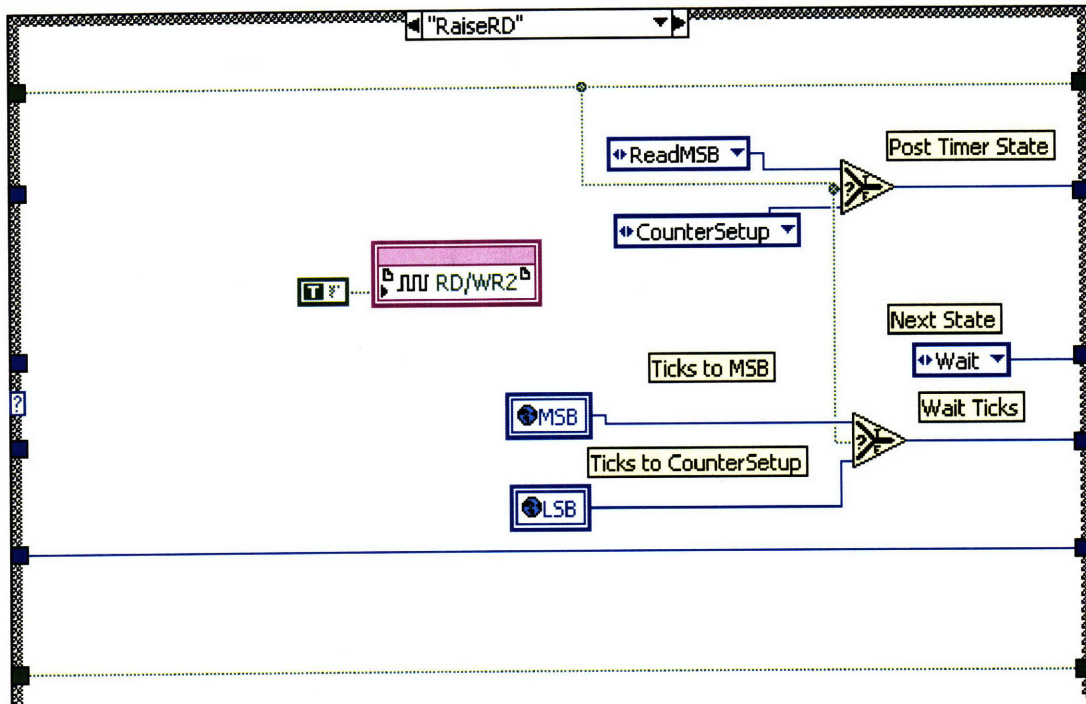


Figure 5-7: Example of LabVIEW FPGA code for A/D acquisition - *RaiseRD* state.

The A/D data sample is outputted in two's complement format. This is shown in Figure 5-8. Two's complement is a system in which the negative number is represented as the two's complement of the absolute value and is beneficial because it allows addition and subtraction by only adding two numbers as the sign is built into the number. For  $n$ -bit precision, numbers are wrapped around on overflow. It is also beneficial because zero is only represented once. However, for it to be used correctly, the word bit size must be maintained for all operations.

The LabVIEW FPGA module works with several different data types, including 8, 16, 32, and 64-bit signed and unsigned integers as well as a fixed-point data type. Double precision representation is only available in floating-point processors. The fixed-point data type allows the total and integer word length to be defined for each variable. This sets the maximum range and precision. However, few high-level LabVIEW functions used the fixed-point data type.

Fixed-point numbers can also be represented as integers by pre- and post-scaling the decimal number. As with A/D sampling where the sample is limited to the

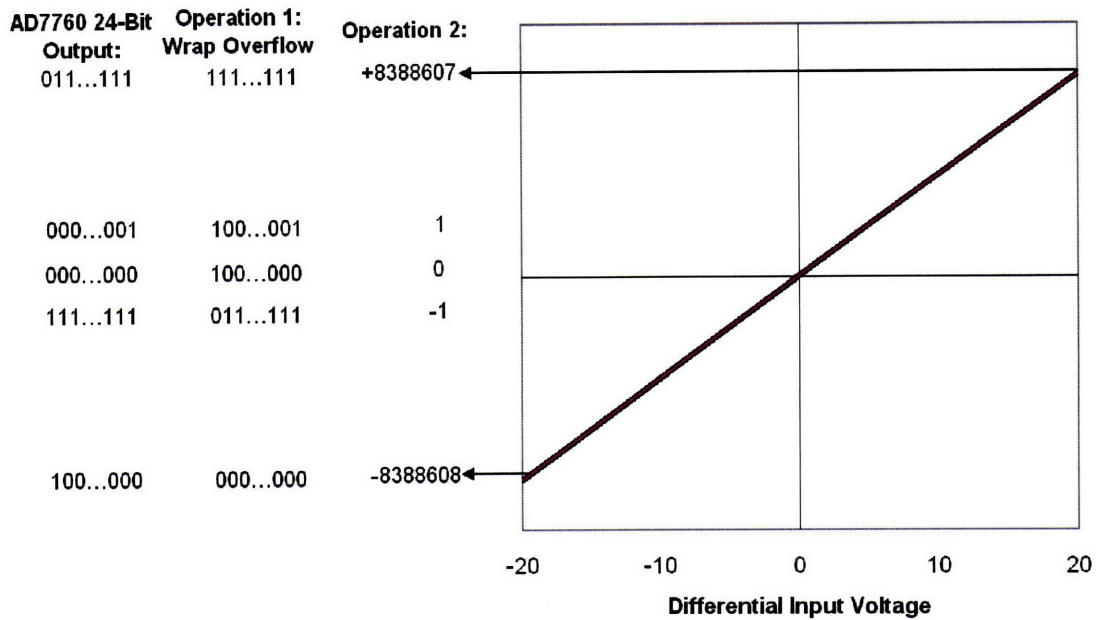


Figure 5-8: A/D data sample output two's complement format.

precision of the output number of bits and dynamic range, fixed-point numbers are quantized to a fixed-precision number of bits. To multiply the U32 (unsigned 32-bit integer) number 100 by Q16 (16 fractional bits) number 0.1, the coefficient 0.1 is pre- and post-scaled by  $2^{16}$ . The quantized value of the Q16 coefficient is subsequently 0.100006. The fixed-point multiplication of this operation is

$$\text{ROUND} (100 \cdot \text{ROUND} (0.1 \times 2^{16})) \times 2^{-16} = 10 \quad (5.6)$$

No quantization affects are seen in the output because values were chosen that utilized the ranges of the data types. For the number  $2^{23} = 8388608$  with the coefficient  $10^{-5}$ , the answer should be 83.9, however with quantization and rounding it is

$$\text{ROUND} (2^{23} \cdot \text{ROUND} (10^{-5} \times 2^{16})) \times 2^{-16} = 127 \quad (5.7)$$

The error is principally due to the coefficient quantization, which was actually  $1.526 \times 10^{-5}$ . The LSB size was  $1.526 \times 10^{-5}$  in this example.

Along with quantization, the data type word length is important for saturation

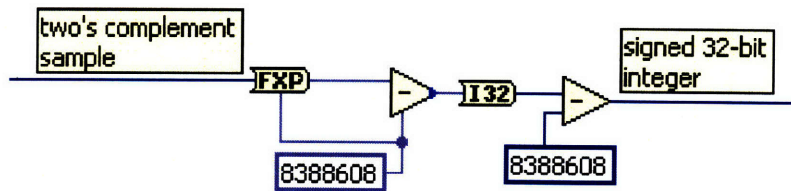


Figure 5-9: Converting two's complement to signed 32-bit integer.

effects. Two's complement is based on the wrap overflow method, but this is only for addition and subtraction while maintaining a constant word length. In multiplication operations, the word length needs to be doubled. For two 16-bit numbers multiplied together, the output data type needs to be 32 bits to avoid overflow. This can be compensated for by multiplying to 32-bit precision and then scaling the output down by 16 bits. This again introduces an error due to fixed-point computation but maintains 16-bit precision [10].

Because the A/D converter output is truly representative of signed values, the 24-bit two's complement value is converted to a signed 32-bit integer. This process is shown in Figure 5-9. The first subtraction block moves the zero point by  $2^{23}$  with the overflow wrap method. The second operation converts the value to the LabVIEW signed integer. The equivalent bit values for these two operations are shown in Figure 5-8.

### 5.3 D/A Output Interface

The D/A output module is implemented similarly to the A/D acquisition. The module is a finite state machine with 6 states and the block diagram is shown in Figure 5-10. The D/A has three input lines for data  $D_{IN}$ , clock  $CLK$ , and chip select latch  $\overline{CS}/LD$ . The datasheet specifies that the chip select latch is to be pulled low while loading data. The data is latched into the D/A register with serial communication, set on the rising edge of the clock  $CLK$ . As with the A/D acquisition module, wait commands are used so every state must specify the digital output condition, the next

state, and wait and post-wait state if necessary. The D/A converter is 16-bit and thus requires that the sample output be decimated to unsigned 16 bits. The 6 finite states are:

- *LowerCsn* - Lower the chip select latch  $\overline{CS/LD}$ .
- *SetSDI* - Set the data line. The data is latched to the D/A register from MSB to LSB. Therefore a left bit shift with carry operation is used.
- *Wait* - Wait for a selected number of clock ticks. The wait period is set at one 80 MHz tick, or 12.5 ns, for the fastest error-free operation.
- *SetSclk* - Set the clock to latch the data line into the D/A register.
- *ClearSclk* - Clear the clock line.
- *RaiseCsn* - Raise the chip select latch  $\overline{CS/LD}$  line and latch D/A register to D/A output. This creates the new conversion voltage on the D/A output.

The high-resolution dSPACE interoperable system described in Chapter 2 had a maximum loop rate of 8 kHz, however this was not limited by the D/A hardware. The datasheet claims a 16-bit settling time of 4  $\mu$ s, or a maximum sample rate of 250 kHz. However, timing constraints allow for 625 kHz loop rates. For each sample output a glitch is incurred. The 2 nV-s glitch impulse, documented in Figure 2-14, is generally negligible when the output is connected to electromechanical systems that have much lower bandwidths. Better accuracy on the output at these high sample rates could be obtained by using a different D/A converter that has a faster settling time to 16-bit accuracy.

## 5.4 Digital Control Implementation

Sigma-delta converters, while providing high-resolution at high sampling rates, introduce an inherent propagation delay because the necessary computations for filtering take time to complete. Samples between separate A/D converters remain synchronized throughout this pipeline delay so they also remain synchronized in the temporal domain. The main problems with this delay is a phase loss in the control loop. The

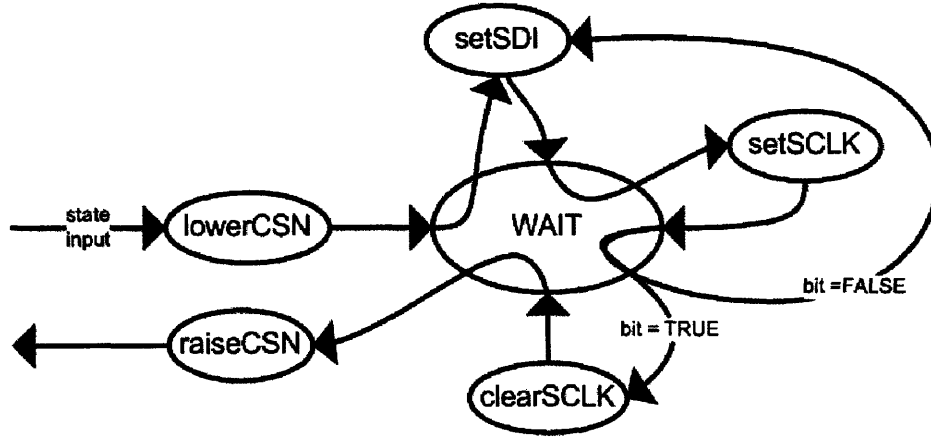


Figure 5-10: D/A output state diagram.

dominant time delay is due to the A/D converter itself. For 2.5 MHz sampling, the propagation delay is  $T_{AD} = 10.8 \mu s$ . The time to bring the digital sample into the FPGA to a FIFO is an additional 400 ns. The remaining stages are simplified in the Figure 5-3 block diagram.

As mentioned earlier, FPGAs are ideal for parallel processing. For a series of  $c_i$  computations that take  $t_i$  long to each compute, there are two methods for processing. The first is to be sequentially processed, where the loop period is

$$T_{SEQ} = \sum t_i \quad (5.8)$$

to complete all computations and repeat. The time for a sample to enter the computation stream and then exit is also  $T_{SEQ}$ . Only one stage is processed at a time which means that each stage is not efficiently used, however there are no delays between stages. The other method is to process the computations in parallel. In this case the loop period is

$$T_{PIPE} = \max(t_i) \quad (5.9)$$

and the time for a sample to pass through all the computations is  $n_c \times \max(t_i)$ . Each computation is processed in parallel and placed in a shift register, however the subsequent loop of computations cannot be processed until all shift registers

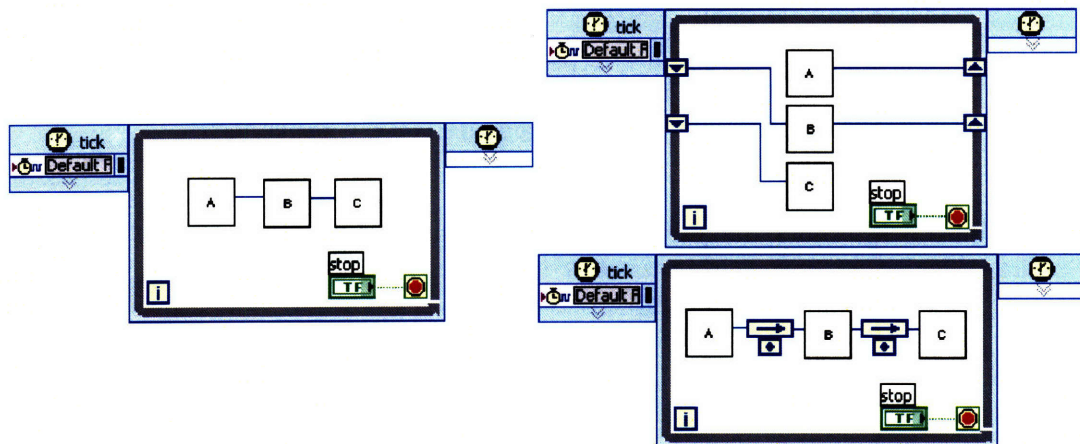


Figure 5-11: FPGA sequential (left) versus parallel/pipeline (right) processing. Pipeline processing can be implemented with shift registers (top) or feedback nodes (bottom).

clock in new data. The computation time for all stages then becomes the maximum computation time for any one stage. Parallel, or pipeline, processing can have higher loop rates but at the cost of the total computation time for a single sample. This is the same as the sigma-delta converter operation which uses the terms propagation or group delay. Alternatively, sequential processing has slower loop rates but a shorter group delay time. Data registers are used in pipelined processing to pass samples from one computation to another. The two methods are shown in Figure 5-11.

The computation time for the implemented controllers and logic depends on the controller complexity. For the lag, triple-lead, and high-frequency low-pass controller implemented in Section 6.2, there are six shift registers representing a controller computation time delay of  $T_{CTR} = 2.4 \mu s$ . From Section 5.1, downsampling is required to decimate from 2.5 MHz down to 625 kHz. The process is shown in Figure 5-12. A 4-point averager is used to decimate the data stream to 625 kHz by summing 4 points and on the last summation scaling by  $\frac{1}{4}$  to output to the D/A FIFO. The time delay due to this averaging is captured in the FIR representation first discussed in Chapter 2. This FIR filter representation is

$$G_{FIR}(z) = \frac{1}{4} (1 + z^{-1} + z^{-2} + z^{-3}) = \frac{z^3 + z^2 + z^1 + 1}{4z^3} \quad (5.10)$$

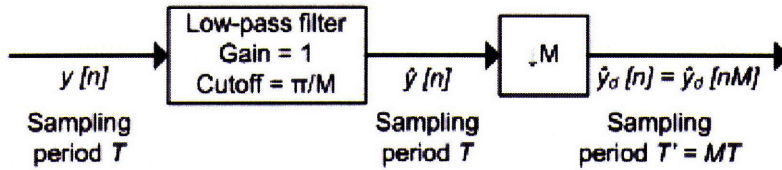


Figure 5-12: Block diagram of discrete downsampler, adapted from [10].

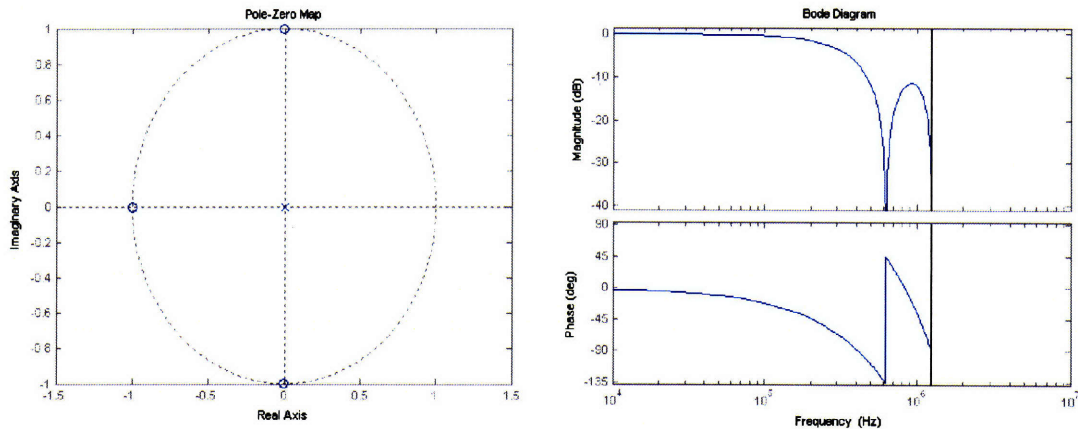


Figure 5-13: FIR  $N = 2$  pole-zero map and frequency response at 2.5 MHz sampling rate.

where  $z$  has a sampling time of 400 ns. This finite impulse response filter has memory limited to the last  $2^N$  points. All the system poles are located at zero and the zeros are equally spaced on the unit circle. The pole-zero map and frequency response are shown in Figure 5-13. The ideal filter cuts off at the first lobe of 625 kHz however a non-ideal lobe is also present. The non-ideal lobe allows aliasing however signal size is typically strongly attenuated at those frequencies. The first lobe is primarily responsible for reducing aliasing effects.

The time delay associated with the D/A FIFO and transmitting the data to the D/A is  $T_{DA} = 9.6 \mu s$ . A data register is required to break the path between the FIFO read and the D/A output. Without this register the path would require 85 clock ticks at 40 MHz which is equivalent to a maximum data rate of 470 kHz, however the register also adds an additional  $3.2 \mu s$  delay. This brings the total propagation/system



delay to

$$T_{DLY} = T_{AD} + T_{CTR} + T_{DA} = 11.2 + 2.4 + 9.6 = 23.2 \mu s \quad (5.11)$$

where  $T_{AD}$  is the time to acquire a sample through the A/D,  $T_{CTR}$  is the time to process the controller, and  $T_{DA}$  is the time to output a data sample. The transfer function due strictly to a time delay in the continuous-time domain is

$$G_{DLY}(s) = e^{-T_{DLY}s} = e^{-23.2\mu s \cdot s} \quad (5.12)$$

The time delay converted to the discrete-time domain requires that the delay time  $T_{DLY}$  be an integer number of time samples  $T = 400$  ns, which defines the discrete-time operator  $z$ . The equivalent discrete-time transfer function is

$$G_{DLY}(z) = z^{-58} \quad (5.13)$$

and the complete transfer function due to the digital platform is

$$G_{DIG}(z) = G_{DLY}(z) G_{FIR}(z) = z^{-58} \frac{z^3 + z^2 + z^1 + 1}{4z^3} \quad (5.14)$$

The expected phase loss is -42.8 degrees for a closed-loop bandwidth at 5 kHz and is the limiting factor on the closed-loop bandwidth of a controlled system because the phase loss increases linearly with frequency. The expected frequency response for the digital control system is shown in Figure 5-14. The 0.5 magnitude (-6.02 dB) scale factor is included to compensate for the A/D front-end configured for fully differential signals. Test hardware such as an HP dynamic signal analyzer are typically single-ended and are thus attenuated by a factor of 2, or 6.02 dB.

Closed-loop sample rates must be at least 10-20 times greater than the closed-loop bandwidth [20]. Sigma-delta converters paired with FPGA control systems shift the limiting factor from sampling rate as seen in traditional embedded processing systems to phase losses as described here.

The terms controller and filter are used interchangeably throughout this work.

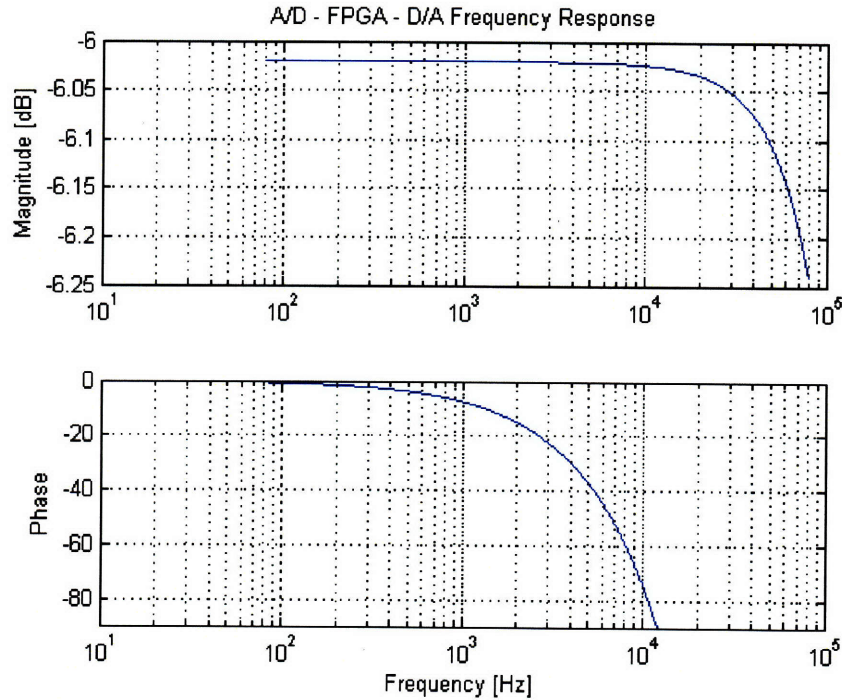


Figure 5-14: Expected frequency response for the digital control system.

Controllers typically refer to the hardware that implements digital or analog filters. These filters interact with dynamic signals and modify them to produce a desired output. It is common to refer to a discrete controller transfer function as  $H(z)$  but this is also considered a filter and used interchangeably throughout this work. LabVIEW refers to the implemented transfer function as a digital filter.

### 5.4.1 PID Control

LabVIEW provides basic PID control functionality with the discrete PID FPGA block shown in the Figure 5-15. This simplified block is limited in functionality and is intended for users with little experience in implementing linear controls. One limitation is that the data path is limited to 16 bits, with the gains having 8 bits to represent the decimal portion of the gain. Simple PID also does not have the universality that arbitrary controller transfer functions are capable of. For a typical

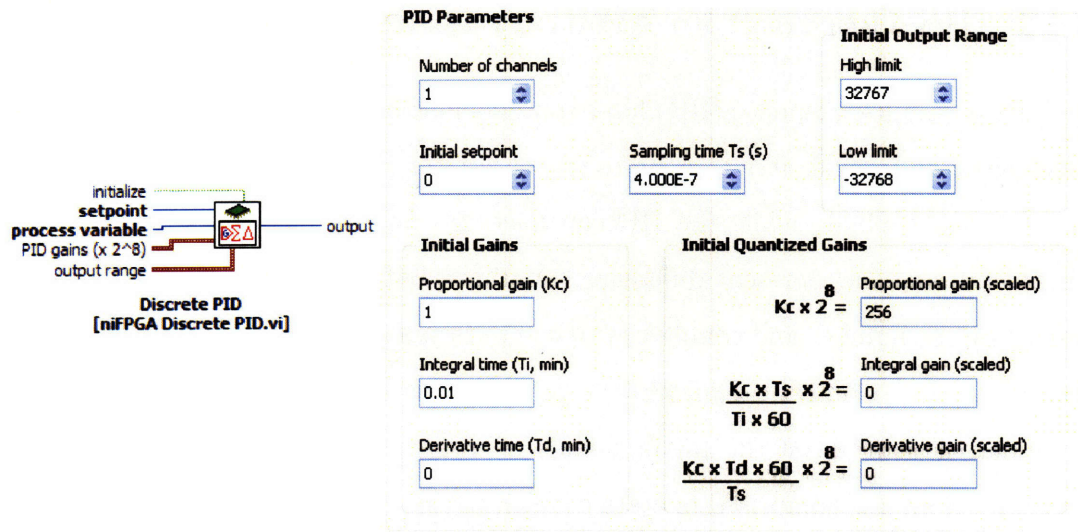


Figure 5-15: LabVIEW PID block.

PID representation of

$$H_{PID}(s) = K_p + \frac{K_i}{s} + K_D s \quad (5.15)$$

a single PID block is only capable of proportional and lag control. Lag control is achieved by a PI combination, giving

$$H_{PI}(s) = K_p + \frac{K_i}{s} = \frac{K_p s + K_i}{s} \quad (5.16)$$

Lead control would need to be implemented as a combination of several PID blocks and require more complex design than a lead controller in designed Matlab. LabVIEW functions such as the discrete PID block are also designed for ease of implementation and simple usability as opposed to efficiency or use of resources. Benchmark tests on FPGA utilization are not provided for the PID block but the maximum operating frequency is 3.33 MHz. The IIR filter method described below can run a single lead controller at 5.7 MHz on a 32-bit data path.

## 5.4.2 IIR Control from Arbitrary Discrete Transfer Function

An infinite impulse response (IIR) filter contains memory and has an infinite impulse response to an impulse input. A finite response filter (FIR) has all the poles at zero in the discrete plane, while the IIR can have the poles arbitrarily spaced. There are several ways to represent difference equations, but when system performance is limited by the number and complexity of computations, it is essential to implement a simplified form. Control canonical form provides this simplification and is described in numerous sources on the fundamentals of discrete systems [13, 10]. A control transfer function is comprised of state coefficients in

$$H(z) = \frac{b(z)}{a(z)} = \frac{b_0z^i + b_1z^{i-1} + \dots + b_i}{z^i + a_1z^{i-1} + \dots + a_i} \quad (5.17)$$

There are a number of ways to organize these coefficients such as direct form I, direct form I transposed, direct form II, or direct form II transposed [10, 11]. The minimum number of delay elements are required in direct form II. An example of this form is shown in Figure 5-16. The transposed form reverses the direction of all signal paths and is the same equivalent transfer function but alters whether the input flows through the  $b_i$  coefficients before the delay elements or after. Depending on the implemented filter, a transposed form may or may not be beneficial to minimize rounding and saturation effects in delay, addition, or multiplication operands. Both direct form II implementations were tested with negligible differences in the designed filters.

Another option is for cascaded systems. This takes a high-order filter and factors it into second-order cascaded systems. These can provide numerous combinations. The cascaded systems are equivalent in the case of infinite precision arithmetic, however varying fixed-point configurations can produce different results due to finite precision. Cascaded form generally requires more resources because each stage is constrained to second-order and thus a third-order filter will have a zero coefficient.

National Instruments provides a number of tools to work with discrete filters. Most of them are available in the add-on Discrete Filter Design Toolkit. The tools range from filter design, to simulation, to FPGA code generation. I completed all filter

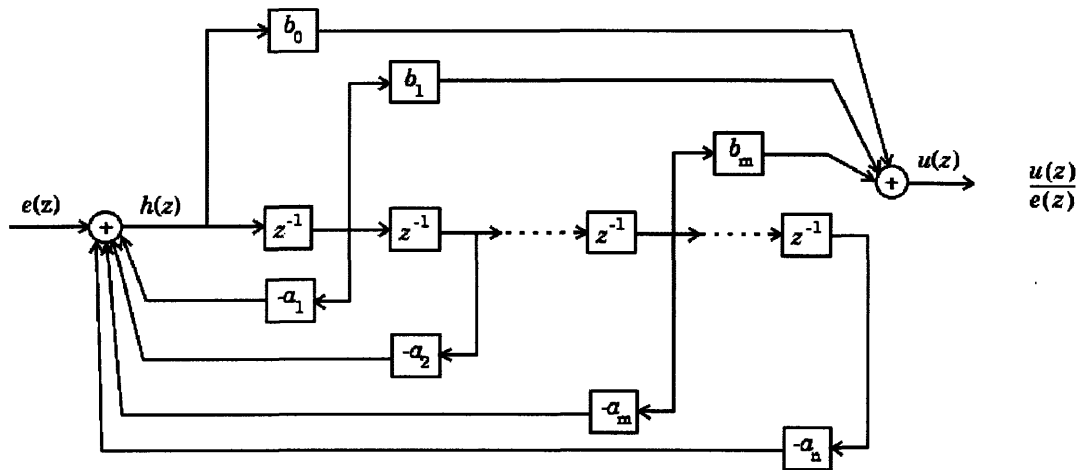


Figure 5-16: IIR filter canonical control direct form II block diagram.

design with Matlab in continuous-time and then converted to discrete form. The filter transfer function is then imported to a custom written LabVIEW virtual instrument that quantizes the discrete coefficients, simulates the filter response, and generates the FPGA code to be compiled in the end application. The code is based on an example VI provided with LabVIEW documentation but there are three major features. First, a custom VI was written to import a Matlab transfer function. Second, advanced instances were used as opposed to simple instances to allow more specific features to be set. Third, analysis and simulation features were added. Complete LabVIEW code for the *Filter\_FPGA\_Code\_Generator.vi* is included in Appendix C.

The transfer function designed in Matlab is converted to the discrete-time domain with the Tustin bilinear approximation, or trapezoid rule. Other options would be to use forward or backward Euler approximations. The Tustin method maps the entire left half of the s-plane into the unit circle of the z-plane. The s- to z-plane transformation is

$$s = \frac{2}{T_s} \frac{z - 1}{z + 1} \quad (5.18)$$

where  $T_s$  is the sampling period. The forward method can lead to instabilities because the left half of the s-plane can map outside of the unit circle whereas the backward method is conservative and maps to within a quarter portion of the unit circle. Lab-

VIEW uses the backward method for transforming to discrete-time filters, such as the discrete integrator. Although the stability boundaries are mapped coincidentally in the case of the Tustin approximation, there is distortion (or warping) within the unit circle. This warping is not compensated for in the discrete approximation.

Averaging and decimation can be implemented before or after control is implemented. This determines what discrete sampling time is defined in the discrete controller approximation. Using a discrete-time filter at sampling frequencies other than that designed for distorts the frequency response and can even lead to instability. An important factor in where to place the downsampler is the effect on time delays. When considering the time delays associated with sequential versus pipelined processing, it was shown that the overall time to process all the computations is longer for pipeline but the sample rate is higher. The sample rate is limited by the length of the longest single computation. Sampling at 2.5 MHz is possible as long as each string of computations can be processed in 400 ns or less. Generally a single lead filter can be have loop rates up to 5.7 MHz with a 40 MHz master clock. A sampling rate of 625 kHz for the controller would allow many more sequential computations but associated delays in parallel processes would be higher. I found that there was a smaller overall time delay by downsampling after the control implementation and directly before the output without a large increase in utilized FPGA resources.

Figure 5-17 presents a block diagram of the IIR filter generation code. The first step is to import the discrete-time filter with a custom written VI from a specified text file. The VI first reads the number of zeros in the filter and then the number of poles. Two separate FOR loops then read in the specified number of zeros and poles and indexes them to an output array. Sufficient precision is obtained with floating-point of %20.30f form from Matlab. To determine the expected coefficients for a given filter from ZPK form, the filter is converted to TF form and the coefficients are the numerator and denominator matrices.

The next stage in the IIR filter generation VI is coefficient quantization, or converting the floating-point filter to fixed-point. As previously discussed with fixed-point data types, quantization and overflow mode have a large affect on how a filter op-

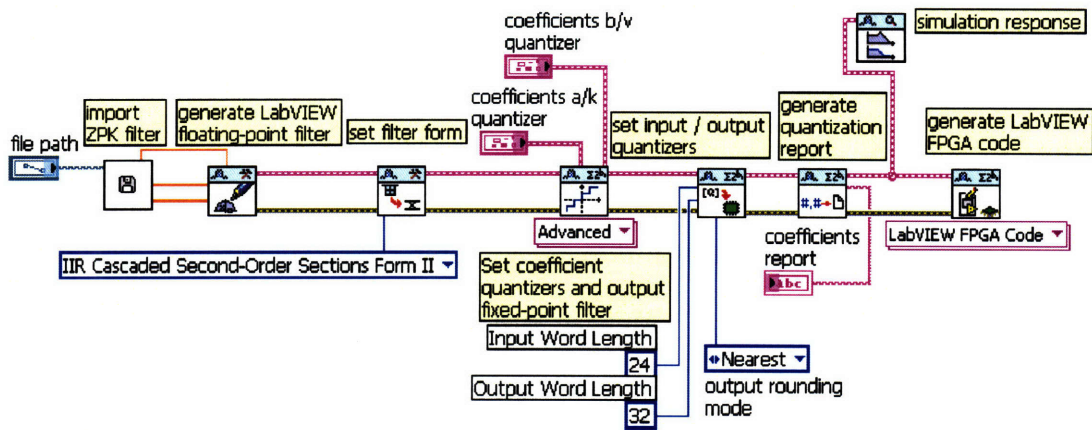


Figure 5-17: FPGA IIR filter code generation block diagram.

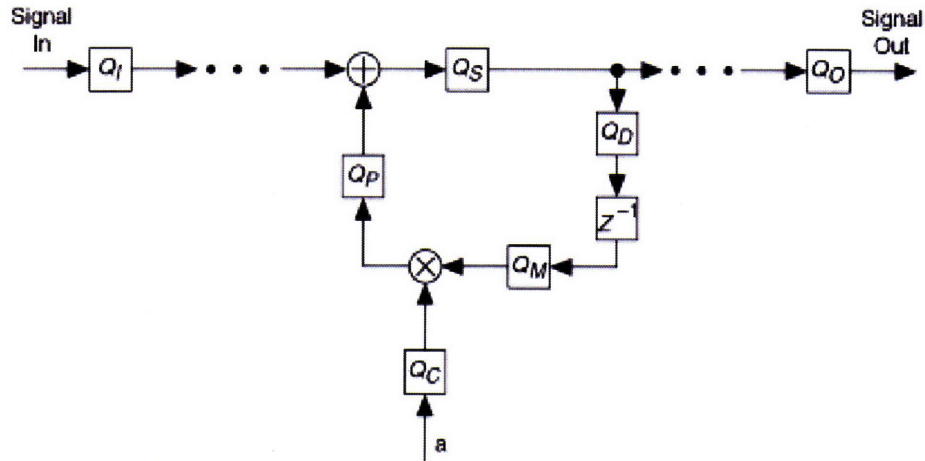


Figure 5-18: Quantizer error introduction in a fixed-point filter [11].

erates. Figure 5-18 shows the different sources of quantization error for a single  $a$  coefficient. The quantization settings are only set by the user for the  $b$  and  $a$  coefficient quantizers ( $Q_C$ ), but settings are also needed for input ( $Q_I$ ), output ( $Q_O$ ), multiplicand ( $Q_M$ ), product ( $Q_P$ ), sum ( $Q_S$ ), and delay ( $Q_D$ ). The input and output word lengths are manually set to accommodate the data stream. The remaining coefficients are automatically set by the LabVIEW VI to accommodate the user-defined quantizers and embedded design rules.

The coefficient quantizers need to be determined through trial and error with simulation. This requires using a simulated data stream similar to that found in

the end application. The overall word length and integer word length is set when defining the coefficient quantizer. The overall word length is maintained at 32 bits which is the maximum for the LabVIEW VI. The data stream word length could be decreased to save resources but this is only necessary when FPGA resources become limited. A narrowed data stream does allow faster throughput computing but the saved time is negligible relative to the decreased performance due to the decreased precision. The integer word length sets the integer range and the remaining bits set the decimal precision. The appropriate word length and integer word length are entirely filter dependent based on order, pole/zero locations, and overall gain. The frequency response and pole-zero map are both plotted for floating- and fixed-point filters to compare quantization effects.

The saturation mode is also a critical component in the quantizer. For example, an integral controller with a steady-state error will quickly rail and saturate. The saturation mode determines whether the internal operations and output will saturate at the limit or wrap around. Typically the output is set for saturation in order to avoid signal discontinuities or sudden changes in amplitude. However, the saturation mode is more complicated than the wrap mode and requires more FPGA resources. For internal quantizers, such as the sum quantizer, the wrap mode is required because this mode allows intermediate overflows and underflows within a certain range as long as the final output does not contain overflows or underflows.

A final quantizer setting is the rounding mode. The “nearest” mode rounds to the closest representable number. If the two closest representable numbers are an equal distance apart, this mode rounds to the closest representable number whose least significant bit is 0. The rounding error of this mode is zero-mean, but this mode has higher implementation complexity than the truncation mode due to the computation of choosing the closest representable number. The “truncation” mode rounds to the closest representable number less than the original value. This mode is the most common but has a nonzero mean.

The next subVI sets the input and output word length. The input is set to 24 bits, the same as the A/D output, and the output to 32 bits, the largest data path



possible for the LabVIEW provided Digital Filter Design Toolkit. Following that VI, a text report is generated about the floating-point to fixed-point quantization. The report lists both the reference value and the quantized value and indicates whether the quantized coefficients have overflows, underflows, or are zeroes, as well as providing the total number of overflows, underflows, and zeroes that the quantizing generates. No metric is provided on the amount of quantization. Provided there are no errors, significant quantization effects are shown on the comparative frequency responses.

The final stage is FPGA code generation, which creates a subVI that can be directly placed in the FPGA application code and compiled to hardware. It also returns the maximum sampling frequency of the filter based on the computation length. The number of required loop iterations increases as the order or number of cascaded systems increases, thus decreasing the maximum sampling frequency. An example IIR filter is included in Appendix C.3.

The precise implementation of the FPGA code generation subVI provided great difficulty in implementing the IIR FPGA control. The  $a$  and  $b$  coefficient quantizers are the only quantizers set in the process. The remaining quantizers are set by LabVIEW. Although the input and output word lengths are defined by the user, the full input and output quantizers are defined by LabVIEW, including the integer word length. Therefore it is necessary to modify the generated FPGA code to account for the integer word length scaling [43]. Each IIR FPGA filter has an input scale factor of  $2^{n_{in}}$ , however there is no output scale factor. The scale factor of  $2^{n_{out}}$ , where  $n_{in} = -n_{out}$ , is manually inserted into the output stream. Although no specific reason was provided by National Instruments through technical support on this issue, the lack of the output scale factor is most likely to keep the data stream from saturating. For example, a lead-filter has unity gain at DC and greater than unity gain at high frequencies. The filter may be compensating for the high frequency gain to avoid saturation. However, changing the set input and output word lengths had no effect on the input and output scale factors. The IIR FPGA code example shown in Appendix C.3 includes this scale factor, as does the experimental results in Chapter 6.

Within the generated IIR filter code, a complex control unit determines which

computations are completed on each loop iteration. The code is placed within a single cycle timed loop (SCTL) to create a known computation length and reduce utilized resources. The SCTL requires that all computations are processed within one clock cycle. The LabVIEW FPGA Module automatically inserts a shift register between every operation except when the operation is located in an SCTL structure. This proves to be an extremely conservative approach but reduces timing constraints in the code compilation process. The principal reason for this is that it reduces the required design ability of the user and allows a broader range of FPGA code users. Code located within an SCTL structure does not have registers located between operations and thus can fail more easily during compilation due to timing. The forward and backward coefficient paths are processed in parallel. The delay blocks are implemented as block RAM located on the NI FPGA board.

Addition and multiplication operations present unique challenges in fixed-point arithmetic. The output of two 16-bit values added together requires 17 bits to avoid overflow. The output of two 16-bit values multiplied together requires 32 bits to avoid overflow. The LabVIEW IIR filter implementation is only capable of 32-bit data paths, however sensor data is 24 bits and this precision is retained along the data path until decimation on the output. The multiplication subVI created by LabVIEW multiplies the signal and coefficient to 64 bits and then scales back the value to 32-bit precision. As discussed earlier, the  $a$  and  $b$  coefficients are converted to fixed-point integers. This post-scaling value is embedded in the multiplication block.

Another interesting approach is that the 32 by 32-bit multiplication is separated into parallel arithmetic paths and recombined at the end. Each 32-bit word is separated into two 16-bit words. For a signal  $x$  and coefficient  $a$ , the multiplication is

simplified to

$$\begin{array}{r}
 a_1 \ a_2 \\
 \times \ x_1 \ x_2 \\
 \hline
 a_1x_2 \ \ a_2x_2 \\
 + \ a_1x_1 \ \ \ \ a_2x_1 \ \ \ \ 0 \\
 \hline
 (a_1x_1) \ (a_1x_2 + a_2x_1) \ (a_2x_2)
 \end{array} \tag{5.19}$$

where the three outputs are combined to form a 64-bit word before it is post-scaled. This includes carry operations for the addition stage.

FPGA hardware is constructed with configurable logic blocks (CLB). The exact number of CLBs and features vary from device to device, but each CLB consists of a configurable switch matrix with 4 or 6 inputs, some selection circuitry (such as a MUX), and flip-flops. The switch matrix is highly flexible and can be configured to handle combinatorial logic, shift registers, or RAM. In some designs, these logic units are replaced with more application specific circuitry, such as dedicated multipliers. This is beneficial because the dedicated logic is more efficient for that specific application when utilized. While the CLB provides the logic capability, flexible interconnects route signals between CLBs and to and from the digital I/Os. With application specific logic units, signals must be routed to and from these blocks. This can cause timing issues and can increase the complexity of routing in the logic synthesis. The LabVIEW FPGA Module masks the interconnect routing task from the user to reduce design complexity.

The National Instruments FPGA device has 96 18 by 18-bit dedicated multiplier blocks. A single IIR filter uses 8 of these blocks, 4 for each 32-bit multiplication subVI. When these are consumed, CLBs need to be configured as multipliers which is not efficient for FPGA resources and can lead to timing errors. The IIR filter is designed to reuse these multiplication subVIs for each coefficient by utilizing the SCTL structure. The SCTL is clocked at 40 MHz, allowing for 16 loop iterations to process a 2.5 MHz data stream. The IIR filter master control unit determines which

signals, coefficients, and arithmetic operations are completed in a single cycle as well as the address of RAM for which it is stored to or read from.

### 5.4.3 Additional Filter Implementations

An important aspect of most control schemes is the accuracy of the DC response. This is obtained by integral control, generally with a lag term that provides infinite DC gain with minimal magnitude and phase at the crossover frequency. The integral windup term needs to be limited or else damaging outputs or long recovery times could be seen in the face of different error sources. An integral or lag controller can be designed and imported to the FPGA code generator VI previously described, however the internal quantizers use wrap for the overflow mode and this causes discontinuities in the filter output. The LabVIEW FPGA Module also provides a discrete-time integrator block. An example of this implementation is shown in Figure 5-19. An ideal integrator is transformed with the backward method so that

$$H_{INT} = \frac{1}{s} = \frac{1}{z - 1} \quad (5.20)$$

the integrator is implemented simply as an accumulator of the signal and previous sum by

$$y_i = x_i + y_{i-1} \quad (5.21)$$

The implementation assumes a sampling interval of  $dt = 1$  and the user is required to multiply the input or output by  $dt$  in a host VI. This external factor is implemented by an adaptation of the IIR filter multiplication block. The coefficient is converted to a fixed-point integer and a multiplication block with the correct post-scaler is used to maximize the range and precision. For the example given in Figure 5-19, the word length is 32 bits and the integer length is 4 bits. The integral gain  $K_{INT}$  is 50 and the sampling frequency is 2.5 MHz, which gives a coefficient value of

$$c_1 = \frac{K_{INT}}{f_s} = \text{ROUND} \left( \frac{50 \cdot 2^{28}}{2.5M} \right) = 5369 \quad (5.22)$$

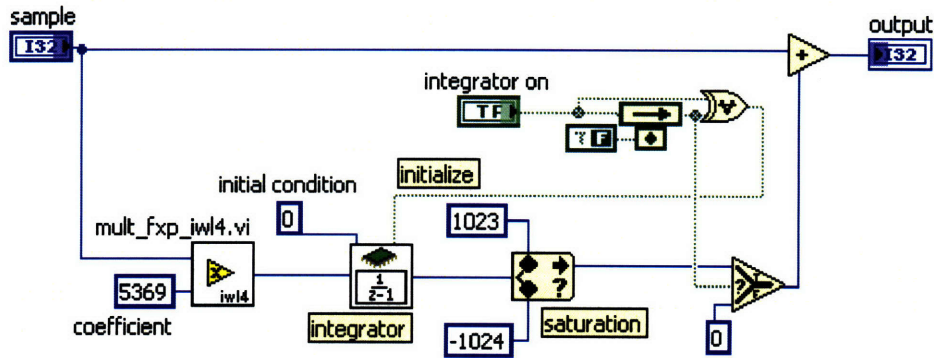


Figure 5-19: Discrete-Time integrator with anti-windup example.

The anti-windup term is achieved with saturation limits. This could easily be implemented with  $\leq$  or  $\geq$  and select logic, but it is conveniently provided as an “in range and coerce” LabVIEW block. The LabVIEW provided “saturate” block is merely the “in range and coerce” block with a different icon. The saturation limit in this example is set to  $\pm 4.76 \times 10^{-7}$  when the data path is normalized to  $\pm 1$ . The boolean logic shown in Figure 5-19 turns on/off the integrator term to the data stream and reinitializes the integrator when the integrator switch transitions true.

An FIR boxcar filter provides an efficient method of averaging a window of data samples. An  $N$ -point moving window is implemented as shown in Figure 5-20. The size of the window  $N$  is held constant. The first  $2^N$  points are then stored to RAM and summed together. The next point is then added to the sum and the first point is subtracted. The output is scaled by  $2^{-N}$  to produce the average of the  $N$  points. The RAM is implemented as a FIFO buffer, simplifying the tracking of the first point in the window. This moving window implementation is compiled within an SCTL so it is computed in 25 ns.

A design for parametric amplitude control is presented in Section 6.2. While the concept is somewhat advanced, it is implemented with a series of parallel loops comprised of traditional controllers. The only advanced feature is an amplitude measuring function. LabVIEW provides an RMS measuring block where the measurement period is set. The parametric amplitude control ensures resonant control of a system even as the resonant frequency shifts. Therefore the RMS measurement period needs

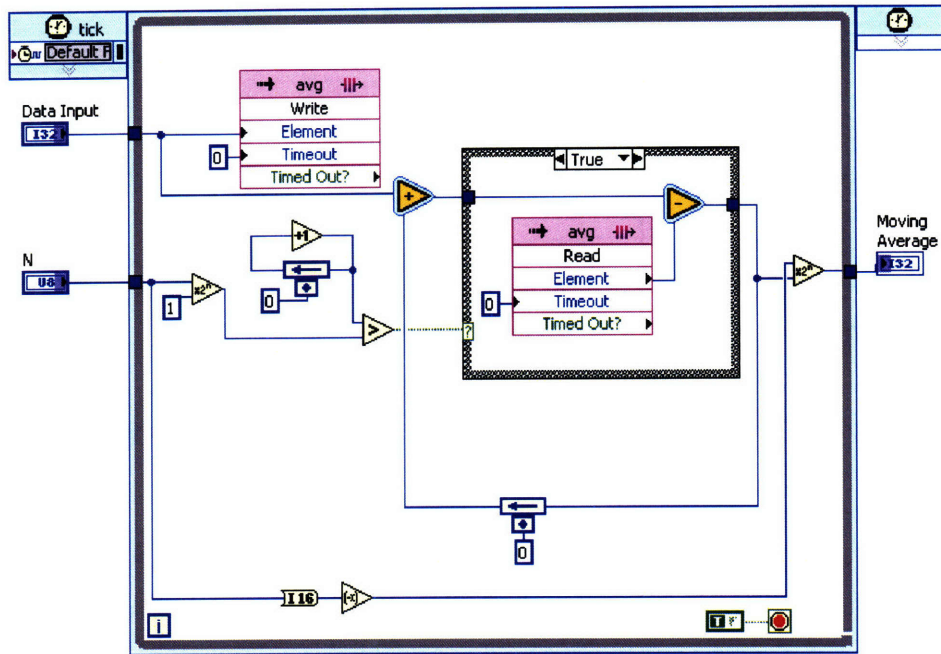


Figure 5-20:  $N$ -point average window FPGA implementation.

to be larger than any resonant period. A custom peak-peak detector was also written, which finds the maximum and minimum values over a given period and then outputs the amplitude of them. The custom peak-peak detector has a larger RMS of returned amplitude values because signal noise on any peaks are considered to be the peak itself. Another option is to square the signal and then low-pass it.

Waveform generation is another set of useful LabVIEW FPGA subVIs. They allow for sine or square waves to be easily embedded within control VIs while still interfacing amplitude, frequency, etc settings. These are useful for generating known references without requiring an additional A/D to bring in analog references.

With the ability to convert any transfer function into an FPGA IIR filter and the additional control features that can be implemented within the LabVIEW environment, it is clear that most traditional control systems can be run with the LabVIEW setup described here. The largest disadvantage is to the FPGA control implementation is in flexibility, development, and debugging. The compile process takes approximately one hour for any change in the VI and thus the debugging process must be designed in during the development and design process. The addition of

lookup tables also allow for complex equations to be embedded and processed quickly.

## 5.5 User Interface and Post-Processing

While a control system runs on embedded hardware, it is equally critical that a user be able to view the system and interact with settings, variables, gains, and filters. A benefit of the LabVIEW setup is the ease with which user interfaces are created and linked to embedded systems. This was critical throughout the testing and debugging phase to locate errors and evaluate performance. Although data can be transferred as a single word, it is more efficiently transferred in large quantities with direct memory access (DMA). The user interface presented below collects a one second burst of two channels sampling at 2.5 MHz. For 32 bits per sample, 8 bits for data integrity and 24 for the sample itself, a 20 MB data burst is seen. This is well within the DMA capabilities.

Moving the data to a computer also allows for floating-point computations. The user interface does not require nearly as high update rates as the control loop itself so the additional computation time is generally tolerable. The data can also be postprocessed. National Instruments and LabVIEW, as well as users and third-party vendors, provide numerous features to use, evaluate, and store data.

An example of a user interface is shown in Figure 5-21. The embedded FPGA VI does not have any control in this example but if it did, it would include inputs for references, loop gains, and integrator/filter control. The data samples are read in and statistics such as the mean and RMS noise in several units are calculated. The raw data as well as an FFT is displayed for each channel through tabbed control.

The data samples are also stored to a Matlab file for further viewing. This data would be postprocessed and mapped onto a topographical plot for the intended application as the data acquisition and control system for an AFM scanner. At this point of experimental results, each channel measures an individual axis and cross-coupling of those axes are not analyzed so pairing of the data is not yet an issue. However, the synchronization of the two channels is guaranteed by the initialization

and operation of the DMA transfer process. Postprocessing will most certainly include high-order filtering. This is useful in removing high-frequency noise that often dominates baseline noise measurements.

This chapter described the LabVIEW software design to interface with analog peripherals and implement real-time control, as well as a host GUI. The next chapter presents experimental results for the A/D PCB design and the full digital platform as it is applied to a mechanical positioner. The application demonstrates 1.5 kHz control bandwidth sub-nanometer control for micron-scale ranges at 625 kHz closed-loop sampling rates.



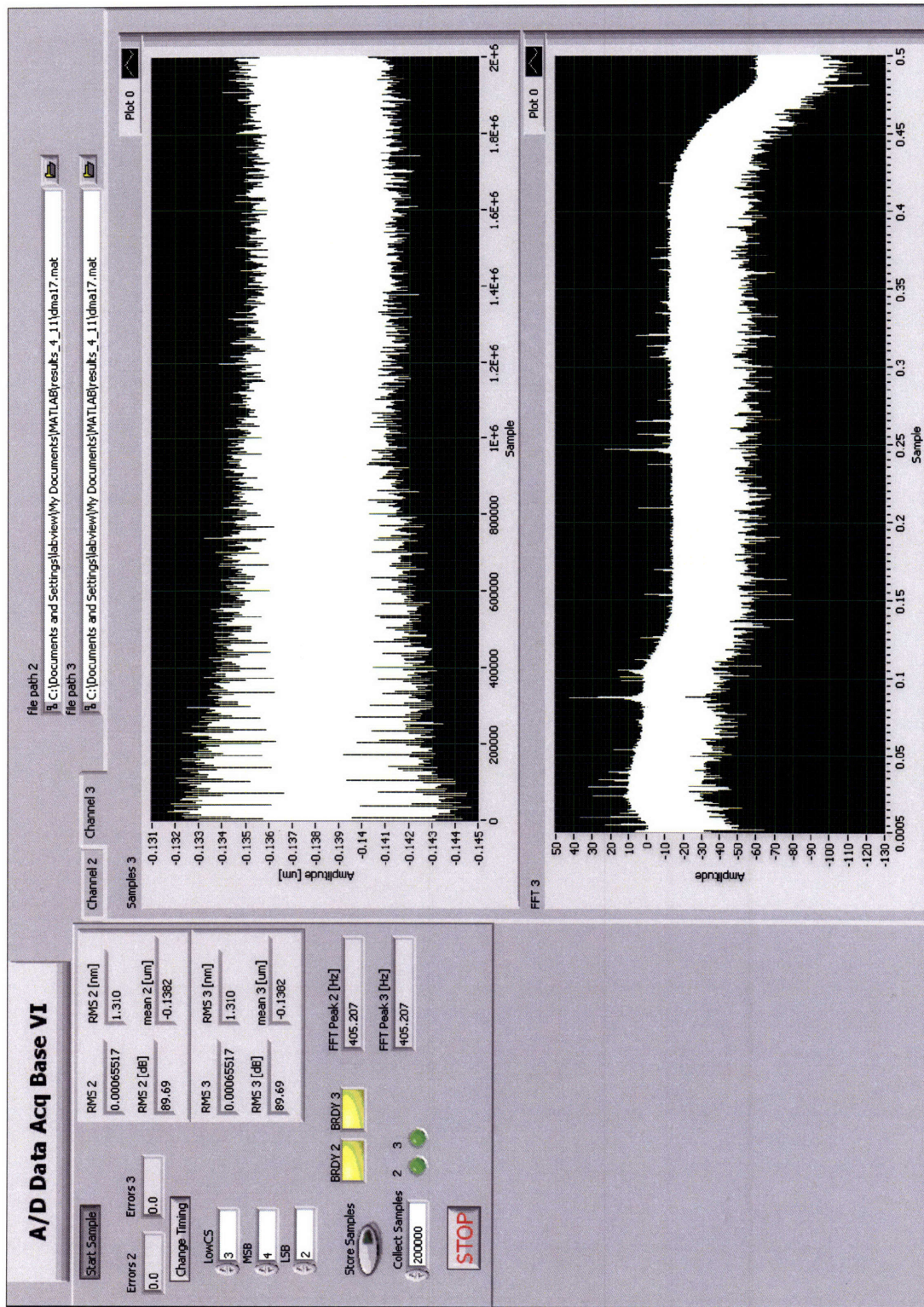


Figure 5-21: Front panel user interface.



# Chapter 6

## Experimental Results

This chapter discusses the characterization of the custom 2.5 MHz A/D PCB and the application of the digital platform to an experimental hardware setup. The hardware is a flexure-based, electromagnetic 2-DOF scanner designed by Ian MacKenzie for high-speed atomic force microscopy. The AFM scanner provided the initial specifications for the digital platform. The digital platform can be applied to a variety of hardware systems and the AFM scanner results are presented as an example of the digital platforms application in a precision system.

### 6.1 A/D Characterization Results

The custom A/D PCB was tested with a variety of inputs. One caveat of characterizing a high-performance A/D converter is that the analog source needs to be extremely low-noise, accurate, and provide its own known characterization. Most of the characterization tests presented here are completed with a grounded input. The precision of the analog source or electronics is discussed as well as the measured results for dynamic inputs. Three operational A/D PCBs have been built and tested. There are several revision changes between them but only for debugging purposes. The measured results are representative of all three PCBs.

By providing a grounded input between the differential inputs of the A/D, the measured unfiltered signal-to-noise ratio (SNR) is  $66 \mu\text{V}$  RMS relative to a 20 V range

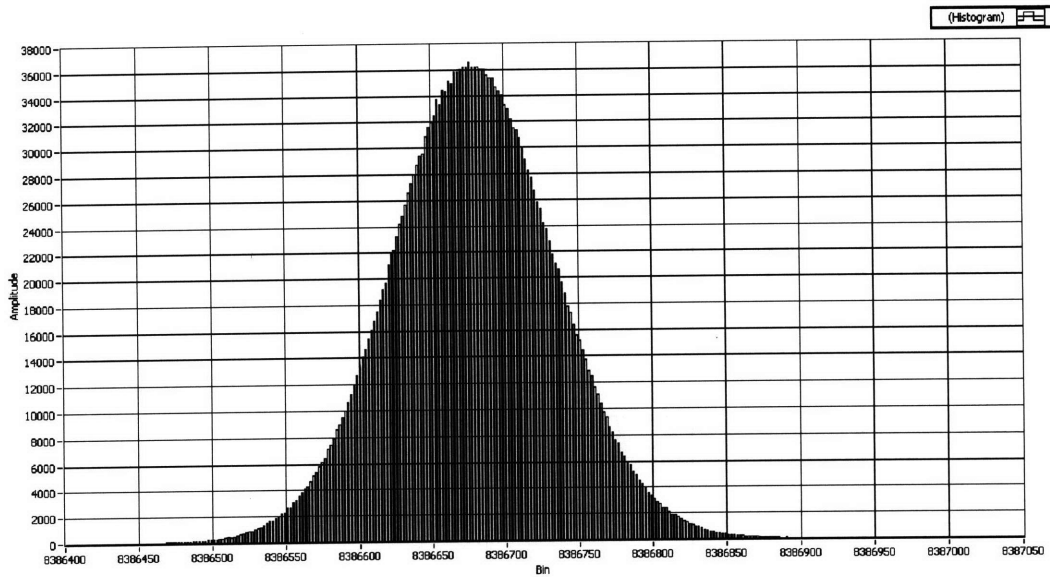


Figure 6-1: Grounded input A/D count histogram.

at 2.5 MSPS. This is equivalent to 109.6 dB or 17.9 effective bits. The histogram for a one-second set of  $2.5 \times 10^6$  data samples is shown in Figure 6-1. The x-axis is presented in integer counts where  $\frac{2^{24}}{2} = 8388608$  is the midpoint corresponding to 0 V input. The results show that there is an offset within the converter of 2 mV which is corrected for in software. The AD7760 datasheet claims a SNR of at least 100 dB, which demonstrates that the analog front-end, power-conditioning, and PCB layout matches or exceeds the requirements for the A/D converter IC. This is a factor of 2.8 better than the expected RMS noise.

A time-scale response over a  $400 \mu\text{s}$  interval as well as the FFT over a one-second interval for the grounded input is shown in Figure 6-2. The maximum frequency content occurs at approximately 670 Hz and can arbitrarily shift by several hundred Hz. Although an effort could be made to locate the source this content, the noise floor is better than expected from the datasheet and larger disturbances are introduced when measuring actual signals. Also shown is the signal response when filtered to 100 kHz with a 25-point digital boxcar filter. The signal baseline noise then improves to  $14.8 \mu\text{V}$  RMS, or 122.6 dB SNR.

A dynamic frequency response was taken on the A/D, FPGA, and D/A system.

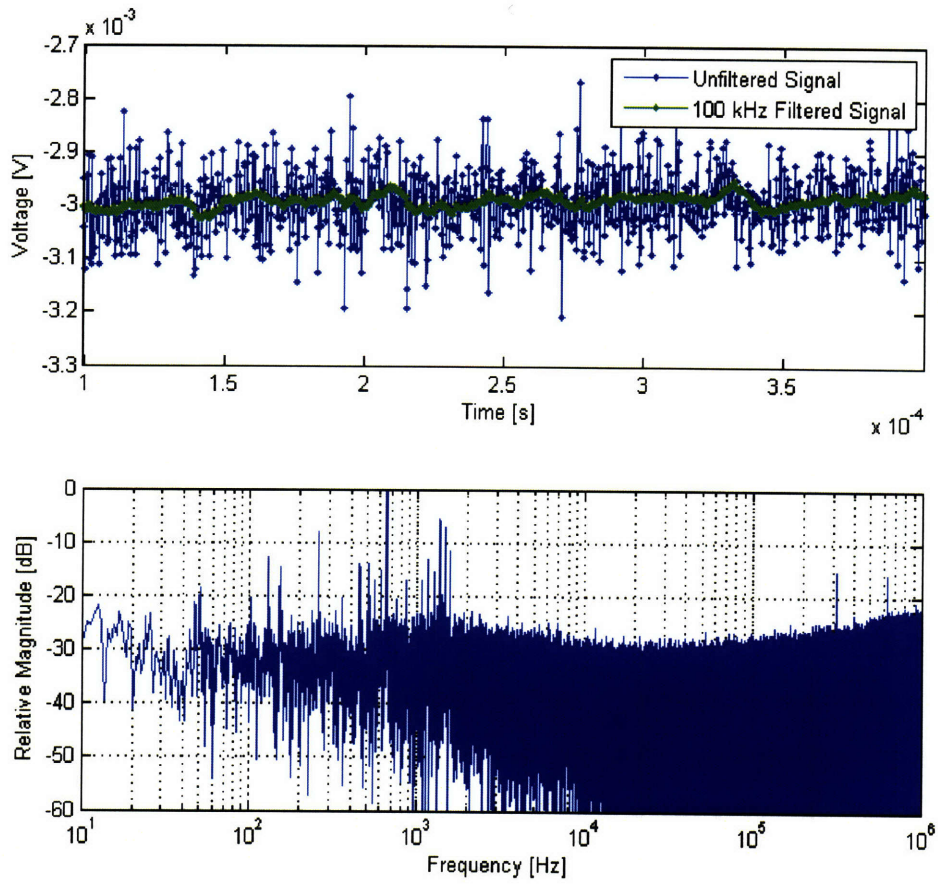


Figure 6-2: Grounded input time response in a  $400 \mu\text{s}$  interval(top) and corresponding unfiltered FFT (bottom).

The system input is the analog signal to the A/D and sampled at 2.5 MHz. The 24-bit digital sample is then transferred to the FPGA where a 4-sample average is computed to decimate the output sample rate down to 625 kHz. This is then transferred to the D/A and converted to an analog output. Figure 6-3 shows this frequency response. The predicted model given by

$$G_{DIG}(z) = G_{DLY}(z) G_{FIR}(z) = z^{-58} \frac{z^3 + z^2 + z^1 + 1}{4z^3} \quad (6.1)$$

The  $z^{-58}$  time delay is equivalent to  $T_{DLY} = 23.2 \mu\text{s}$  where  $T = 400 \text{ ns}$  defines  $z$ . The expected controller computing time  $T_{CTR} = 2.4 \mu\text{s}$  is simulated with shift registers where control would otherwise be implemented. The expected frequency response matches the measured response with the computation delay included. The constant scale factor discussed in Chapter 2 relating to the D/A output amplifier gain is also included in the digital control system to remove the magnitude offset. The linear phase loss due to the time delay becomes non-negligible above 1 kHz and limits the ability to close a control loop beyond 10 kHz. This is discussed further below when the system is applied to a position system with 1.5 kHz closed-loop bandwidth.

A signal generator was used to generate a constant 1 kHz sine wave input in order to demonstrate the acquisition of an external signal. Although the signal generator<sup>1</sup> has a frequency resolution of  $\pm 0.05 \text{ Hz}$ , the distortion is only rated to 70 dB. Figure 6-4 shows the power density of the acquired signal and demonstrates a distortion floor limited by the function generator and not the ADC. Any power line pickup or feedthrough is also eliminated by the common mode rejection of the ADC differential amplifier as no 60 Hz multiple is shown in the response.

Several issues arose when the data acquisition system was coupled with an analog input from an ADE capacitive probe (gauge 6810, probe model 6501) [19]. The first was a ground loop. This was clear by the dominant signal content at a multiple of 60 Hz, generally 120 Hz or 180 Hz. This was measured with the capacitive probe fixtured to a stationary target in order to measure the baseline noise of the probe and

---

<sup>1</sup>Hewlett Packard HP33120A

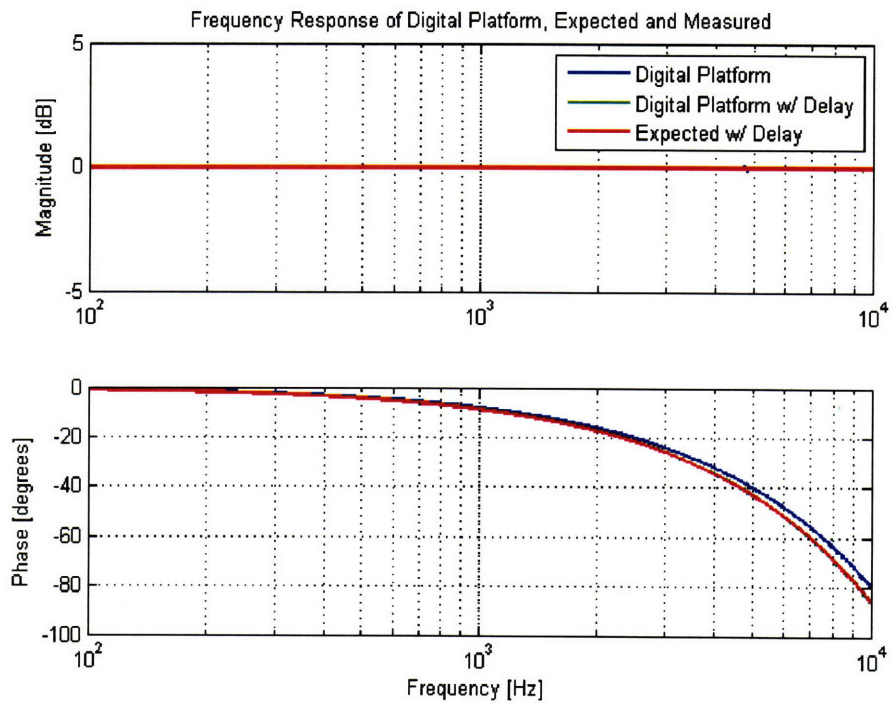


Figure 6-3: A/D, FPGA, D/A system frequency response. The magnitude of the measured digital platform with delay and the expected with delay are coincident and thus the measured result is not independently visible.

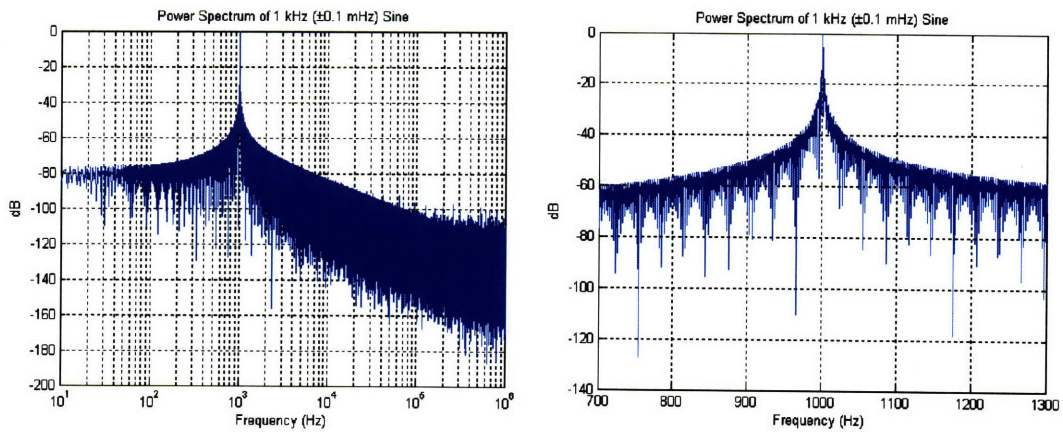


Figure 6-4: Power density spectrum to 1 kHz sine wave with coarse (left) and fine (right) scale.

analog interface itself. This effect was attributed to ground loops forming through the separate instruments. Signal isolation is critical between analog and digital systems, which is why the PCB designs include galvanic digital isolators. However, analog components also need to be isolated, particularly when different power supplies are used.

The ground loop was solved by using fully differential outputs from the probe driver. The A/D analog input was originally designed as a single-ended input with the second input referenced to ground. This meant that the capacitive probe driver and the A/D PCB shared a common ground, which is where the ground loop formed. The change required different A/D analog front-end passive components to accommodate twice the attenuation, which meant doubling the resistance of the feedback resistors and varying the other components to retain the appropriate anti-aliasing poles. The final design for fully differential inputs has been described throughout this work. The output of any sensor needs to be known because if a single-ended output were now connected to the A/D PCB input, the digital sample would be attenuated by half. The single-ended input can be compensated with a simple, commonly described op-amp input circuit for converting single-ended channels to fully differential [35].

In addition to power line pickup, operation of the A/D PCB injects an approximate 100 kHz disturbance into the capacitive probe measurements. This is present even when the driver output is only connected to a Tektronix AM502 differential amplifier and oscilloscope with the A/D operating several feet away and measuring ground input. The disturbance has been attributed to the A/D PCB running but the disturbance is generated within the capacitive probe electronics. The disturbance can be measured with different different analog sinks individually without the A/D PCB connected but merely running. The disturbance is also eliminated when the custom A/D PCB is turned off. Significant efforts were made to identify and eliminate the medium on which this disturbance is injected, however it could only be minimized, and the exact source was never determined. Efforts included:

- using an isolated power transformer for the National Instruments PXI chassis and A/D PCB power supplies.



- earth grounding the isolation table test surface with a “star” configuration. This method creates a single point where all voltages are referenced. This can also introduce additional loops because power supplies can add unwanted noise or that supply currents, flowing in existing ground paths, are sufficiently large, or noisy, or both. This is minimized with separate power supplies for each component. Separate analog and digital supplies, and separate analog and digital grounds, joined at the star point, can potentially assist in minimizing ground loop issues.
- connecting the common of the FPGA breakout chassis to the earth ground, A/D PCB chassis, or PXI chassis.
- floating or earth grounding the A/D PCB power supply.
- floating or earth grounding the capacitive probe driver chassis and system ground.
- connecting the analog signal shields to the A/D PCB or capacitive probe driver chassis.
- connecting the A/D PCB chassis to the A/D PCB, earth ground, or analog or digital cable shields.

Variations of these combinations were tested to achieve the lowest RMS baseline noise measurement. Some variations had very little effect while others could introduce mV level disturbances. The isolated transformer proved to have the worst results because then the ground reference needed to be provided through the test surface or another instrument ground, creating much worse ground loops.

Another issue that introduced additional disturbances was the microcontroller operation on the A/D PCB. I measured approximately 800 kHz disturbance bursts which were generated each time the microcontroller came out of sleep mode. The initial microcontroller design used an internal timer as an interrupt. The microcontroller would enter sleep mode and be brought out by the interrupt, check that the A/D was still operating properly, reset the timer counter, and then enter sleep mode. This issue was solved by eliminating the microcontroller sleep mode and leaving it

constantly running. The described design in Chapter 4 and Appendix B reflects this final design implementation.

Ultimately, a baseline noise measurement of approximately  $300 \mu\text{V}$  was measured with the capacitive probe on a stationary target. The grounding configuration is shown in Figure 6-5. This figure presents the full configuration for the experimental setup discussed in Section 6.2, however for this test the “isolation table” is replaced with the fixture clamped to an electrically isolated surface. The most significant factor was providing an earth ground to the test structure. The capacitive probe technically grounds the target surface and probe, but the a direct connection to earth ground provided better results. Two separate capacitive probes were tested. Their characterization results from the manufacturer and the measurements are provided in Table 6.1.

Table 6.1: Capacitive Probe Characterization and Baseline Measurement Results

	<b>Probe 1</b>	<b>Probe 2</b>
Bandwidth	100 kHz	100 kHz
Range	40 $\mu\text{m}$	50 $\mu\text{m}$
Specified Noise (RMS)	181 $\mu\text{V}$	309 $\mu\text{V}$
A/D Measured Noise (RMS)	252 $\mu\text{V}$	298 $\mu\text{V}$
Diff Amp Measured Noise (RMS) <sup>2</sup>	165 $\mu\text{V}$	260 $\mu\text{V}$

The measurements on the differential amplifier were taken with the A/D PCB turned off and not connected. The measurements were also taken with a 100 kHz, 3rd-order low-pass filter in place on the differential amplifier. An example of the FFT of the approximately 100 kHz disturbance is shown in Figure 6-6. This is an FFT response of the 50  $\mu\text{m}$  probe on the stationary target. The A/D is the dominant noise source for Probe 1, however the A/D is not the dominant noise source for Probe 2, even with the approximately 100 kHz disturbance, because the measured noise is lower than the characterized noise. One reason the A/D measured noise is lower than the characterized noise is because the characterized noise is measured out to 4.6 MHz,

---

<sup>2</sup>Low-pass filtered to 100 kHz on Tektronix AM502 differential amplifier

whereas the A/D noise is only measured to 1 MHz as set by the anti-aliasing filters and digital filters in the sigma-delta converter.

## 6.2 Sub-Nanometer Position Control Results

This section describes results for applying the designed digital platform to a single axis of the AFM scanner designed by Ian MacKenzie in the Precision Motion Control Laboratory. We intended to control both axes with the digital platform, however the project was terminated and the following results are merely an example of the application of the digital platform. A design for parametric amplitude control is presented but was not implemented.

The hardware and 2-axis control designed and tested by Ian MacKenzie was the second of two prototypes [12]. The intended application of the atomic force microscope (AFM) was for high-speed, high-resolution in-line measurement processes for the semiconductor industry. The design is a 2 degree of freedom scanner for a high-speed scan axis and an orthogonal vertical axis. The ultimate goal was to scan a  $50 \mu\text{m}$  by  $50 \mu\text{m}$  scan area of  $10^6$  pixels with a vertical range of  $10 \mu\text{m}$  in one second. The third axis is not part of this design because the specifications are not as demanding.

The high-speed scan axis operates in resonance at approximately 1030 Hz along the x-axis as shown in the Figure 6-7. The point of interest is  $m_2$ , since it carries the probe. The vertical axis then acts along the z-axis with conventional random access control. Motion is constrained by the pairs of flexures labeled in Figure 6-7. Flexures are the primary structure element in part because they can be designed to allow single DOF movement without friction, to the first order, and can be readily scaled to small ranges of motion. The hardware is machined as a monolithic structure to allow each DOF to essentially operate independently. The actuator used is a 2-DOF Lorentz motor with a moving-magnet, labeled in Figure 6-7, and stationary-coil design. The shear-mode actuators are in a stacked configuration which allows for the same type of coil to be used for both the x- and z-axis with decoupled forces, to the first order. The coils are stacked in complimentary pairs about both sides of the magnet. The

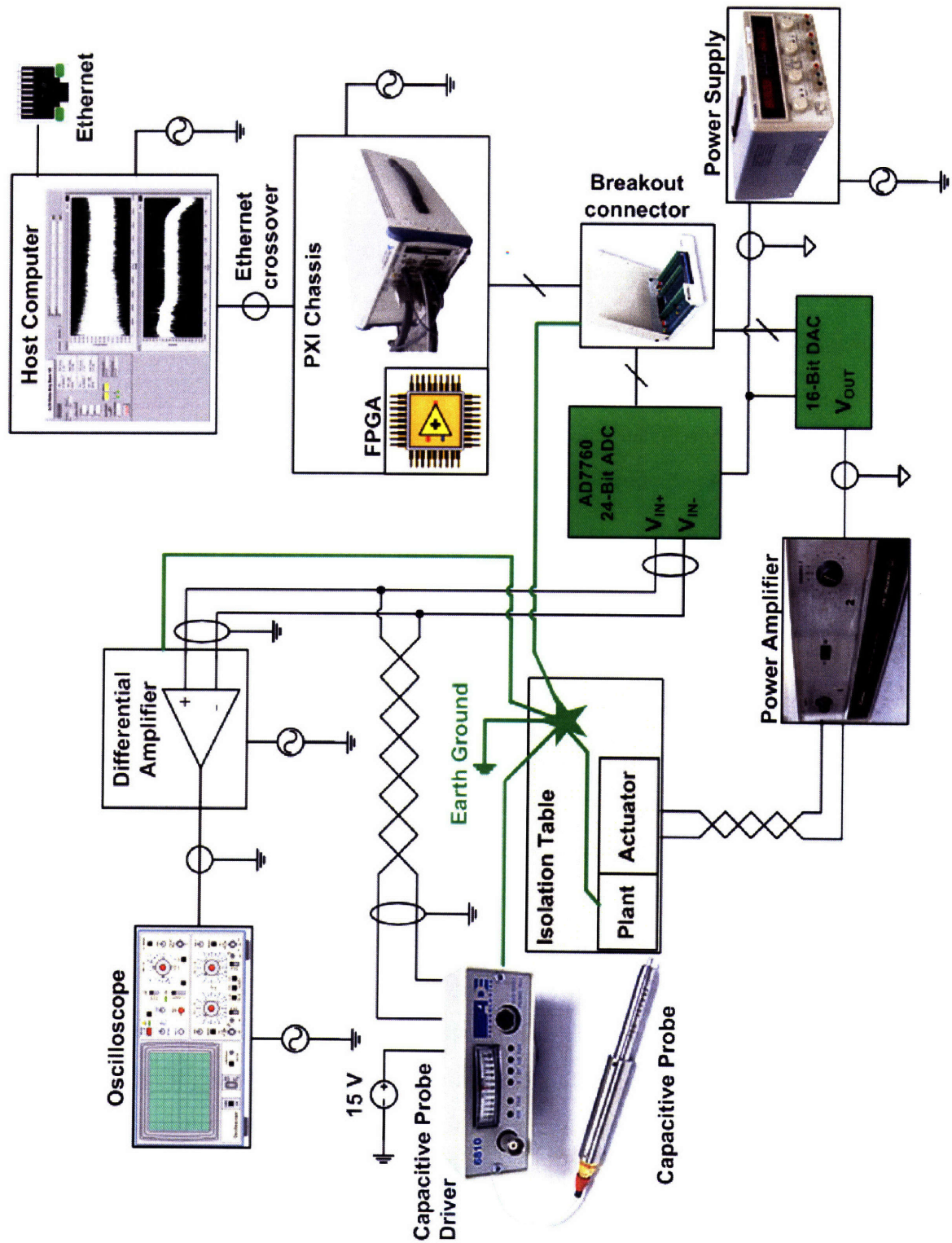


Figure 6-5: Grounding diagram for capacitive probe measurements with “star” earth ground configuration on experimental AFM scanner application. Stationary/fixtured probe tests use the same configuration, although the “isolation table” is replaced by the fixture.

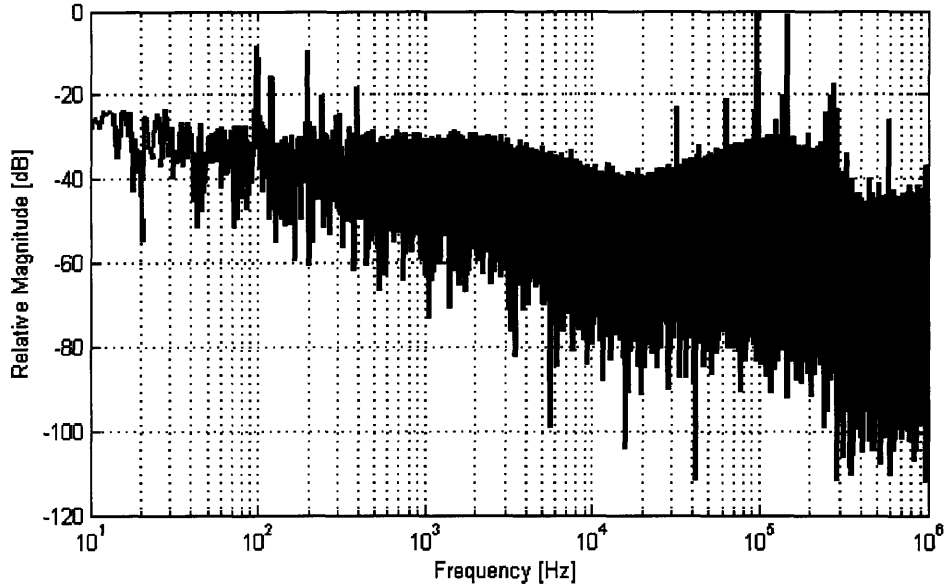


Figure 6-6: FFT of the response of 50  $\mu\text{m}$  probe on stationary target with 298  $\mu\text{V}$  RMS unfiltered baseline noise. This was measured on with the custom A/D PCB hardware and LabVIEW acquisition software.

bottom of the moving mass  $m_2$  is where the AFM tip would be located in future designs, but this design is only intended to demonstrate the scanner design.

The physical hardware with the actuator and capacitive probes in place is shown in Figure 6-8. The capacitive probes measure the moving target. The flexures are designed as part of a lightly-damped spring-mass-damper system. The x-axis is designed for the first mode shape of  $[1 \ 2]^T$  where the moving mass  $m_2$  translates 2 units for every unit the magnet moves. The x-axis flexure pairs are the same length so there is theoretically no cross-coupling with the vertical axis.

The spring-mass-damper system for the model shown in Figure 6-9 is a simple 2nd-order system given by

$$G'_Z(s) = \frac{Z(s)}{F(s)} = \frac{1}{m_z s^2 + b_z s + k_z} \quad (6.2)$$

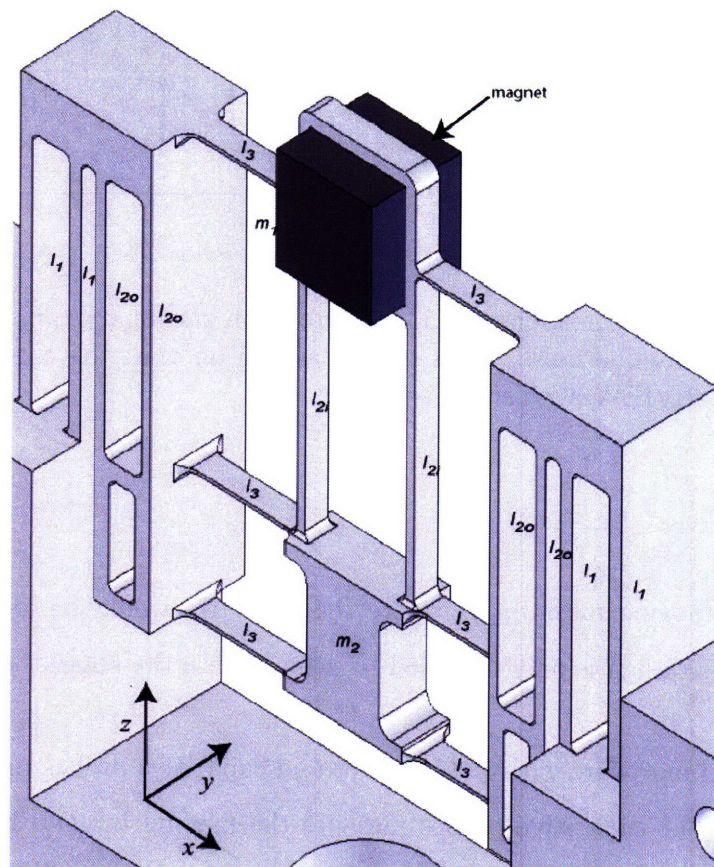


Figure 6-7: 2-DOF high-scan rate positioner CAD model [12].

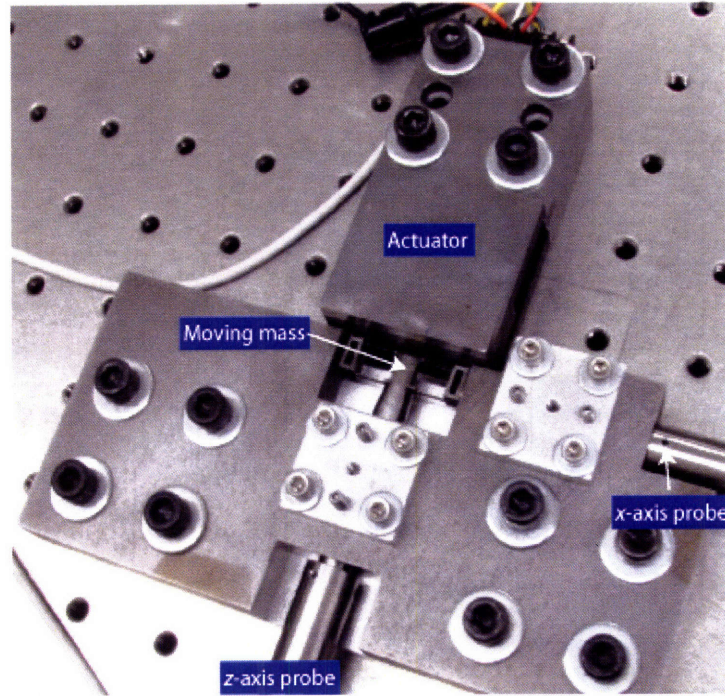


Figure 6-8: 2-DOF high-scan rate positioner hardware.

By accounting for the Lorentz force actuators, the plant is modeled by

$$G_Z(s) = \frac{Z(s)}{V_{in}(s)} = \frac{K}{m_z L s^3 + (m_z R + b_z L) s^2 + (b_z R + k_z L + K^2) s + k_z R} \quad (6.3)$$

where  $R$  is the series resistance,  $L$  is the inductance, and  $K$  is the motor constant. The three poles occur very near each other at 400 Hz. The  $z$ -axis controller implemented by Ian MacKenzie was a lag, triple-lead controller of the form

$$H'_Z(s) = 2.3478 \frac{0.002344s + 1}{0.002344s} \left( \frac{0.0002344s + 1}{4.803 \times 10^{-5}s + 1} \right)^3 \quad (6.4)$$

This design for a 35 degree phase margin at a closed-loop bandwidth of 1.5 kHz. The measured plant frequency response and the applied controller to the plant is shown in Figure 6-10. This controller was implemented on a dSPACE platform with a 50 kHz closed-loop sampling rate. The sensor was the 40  $\mu\text{m}$  range capacitive probe and associated driver. With the native 16-bit ADCs and 14-bit DACs provided by dSPACE, this system was capable of 50 kHz data sampling and 5.0 nm RMS unfiltered

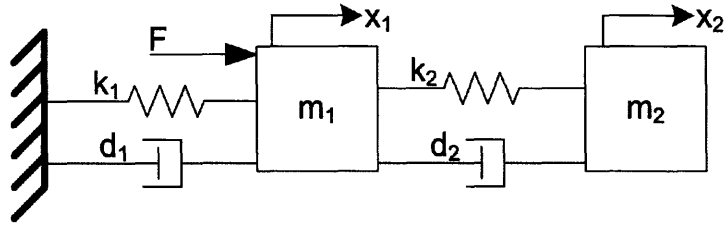


Figure 6-9: Spring-mass-damper mechanical model.

control positional response.

In order to implement the controller on the high-speed, high-resolution platform, the controller was redesigned to accommodate the additional phase loss due to the time delay. At 1.5 kHz, an additional 23 degrees of positive phase is added to the design. The controller used was

$$\begin{aligned}
 H_Z(s) = & \left( \frac{0.002287s + 1}{0.002287s} \right) \left( \frac{0.0001714s + 0.7495}{2.188 \times 10^{-5}s + 1} \right) \\
 & \times \left( \frac{0.0002287s + 1}{2.188 \times 10^{-5}s + 1} \right) \left( \frac{0.0002437s + 1}{2.053 \times 10^{-5}s + 1} \right) \quad (6.5)
 \end{aligned}$$

The gain was combined with the first lead controller to reduce the number of operations required in the data path. The third lead controller has an additional 2 degrees added to the desired phase so that the zeros and poles of all the filters do not lie directly on top of each other. I found in testing that when this was the case, high frequency blips would be generated after several filter stages and be continually amplified until the high frequency blip dominated the signal.

A Butterworth filter was also included in the feedback path to reduce high frequency content. This reduces the possibility of saturation in the data stream due to high frequency gain of the lead controllers. Figure 6-11 shows the time and frequency response for a constant reference with closed-loop control. The unfiltered RMS noise is 430  $\mu\text{V}$  or 0.86 nm and 51.9  $\mu\text{V}$  or 0.10 nm RMS when filtered to 1.5 kHz. The filtered measurements correspond to 111.7 dB dynamic range and 18.3 effective bits. The unfiltered FFT response shows that the dominant signal in the data acquisition stream is the 100 to 200 kHz disturbance. The 60 Hz, or multiple of 60 Hz, is a strong



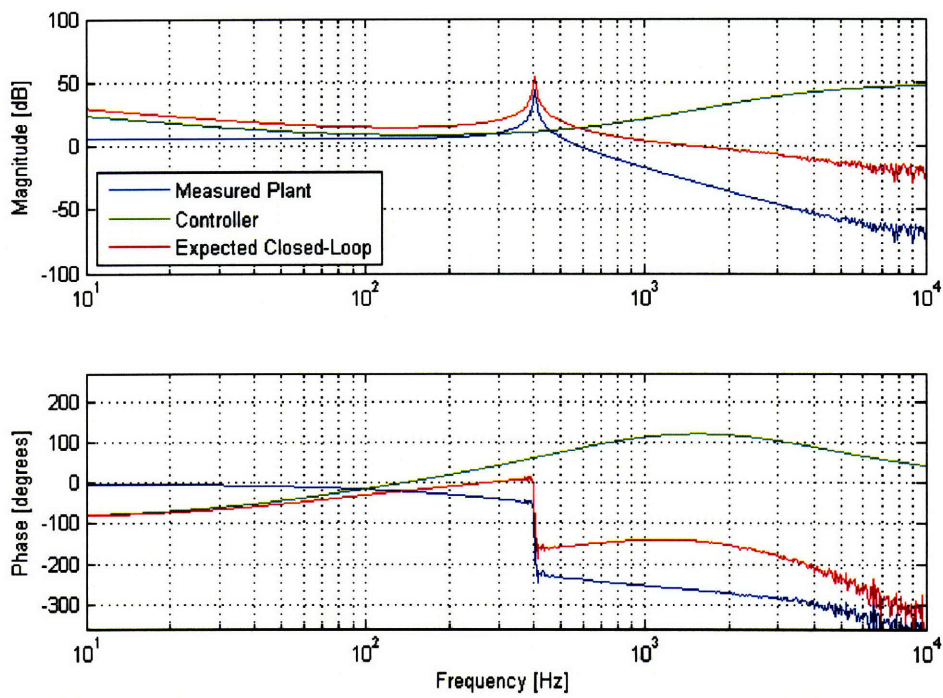


Figure 6-10: Z-axis measured open-loop and expected system loop transmission on dSPACE control platform [12].

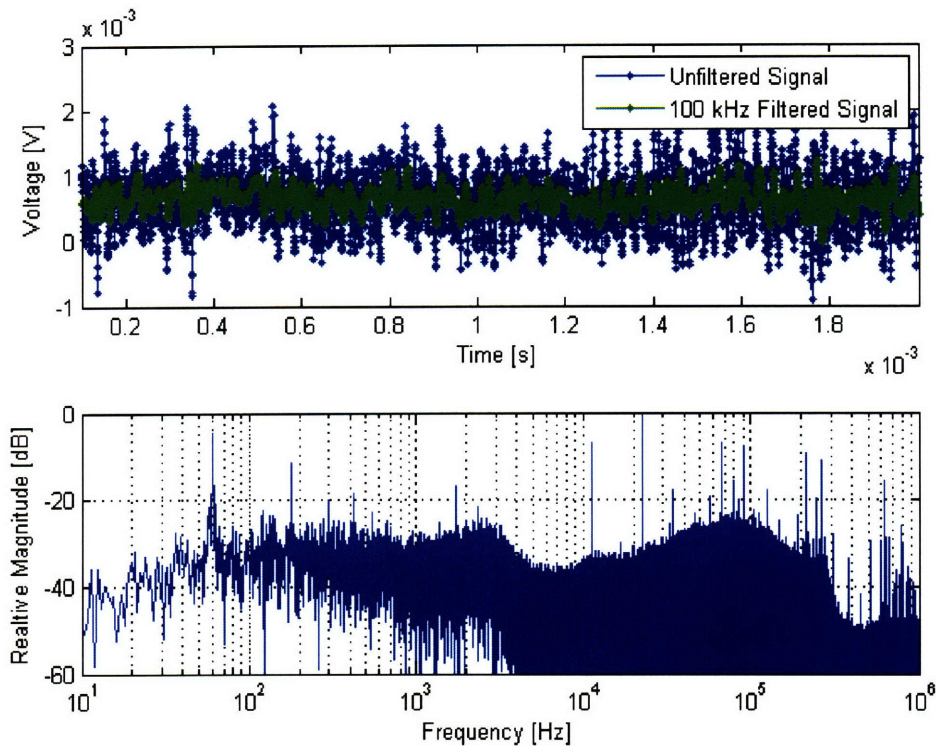


Figure 6-11: Controlled z-axis time response (top) and FFT (bottom).

contributor to noise content even with fully differential signals.

These results compare to the dSPACE control implementation on the z-axis for a constant reference. The dSPACE closed-loop system was measured at 5.0 nm RMS unfiltered and 3.2 nm RMS low-pass filtered at 30 kHz. The unfiltered relative measurements between the two control systems are shown in Figure 6-12. The plot on the right demonstrates both the high-resolution and high-sample rate of the FPGA platform versus the dSPACE platform.

Figure 6-6 and Table 6.1 demonstrates that low-noise acquisition of the capacitive probe is possible. In the stationary target instance, which exhibited less 60 Hz content, an aluminum mount was machined and clamped to a nonconductive table. The capacitive surface of this target is much less than the capacitive surface of an isolation table. Different variations of grounding were tested to find that with the least amount of 60 Hz content. However, the baseline noise on the isolation table

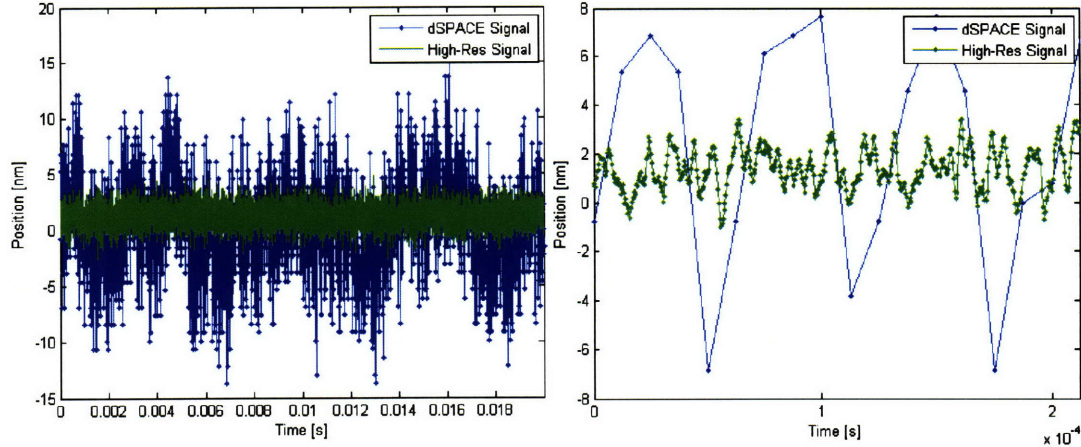


Figure 6-12: Relative comparison of unfiltered measured RMS position with dSPACE and FPGA digital platform closed-loop control to a constant reference over 20 ms (left) and 200  $\mu$ s (right).

could not be matched to that of the stationary target. An ultimate lesson from this application is that for each high-resolution measurement setup, ground paths need to be critically analyzed and tested with different configurations to find the lowest noise setup possible. The best configuration tested is shown in Figure 6-5.

The loop transmission for the closed-loop system is shown in Figure 6-13. The additional phase peak was shifted to a higher frequency to account for the linear phase loss. The additional phase loss due to the digital platform is apparent versus the phase loss for the plant in Figure 6-10, as well as the additional phase required in the controller. The measured step response and error is shown in Figure 6-14. With a phase margin of 35 degrees, the expected overshoot is approximately 35% whereas the measured overshoot was only 8%. The controller could be redesigned with a smaller phase increase which would improve the gain margin as well.

The loop transmission was measured by the scheme shown in Figure 6-15. An active summer was built to allow for an error signal to be injected into the plant. The loop from channel 1 to channel 2 then allows for the loop transmission to be measured via analog signals while maintaining closed-loop control on the system. Alternatively an error signal could be injected into the error data path within the FPGA. This would require an additional A/D channel to bring the DSA signal into

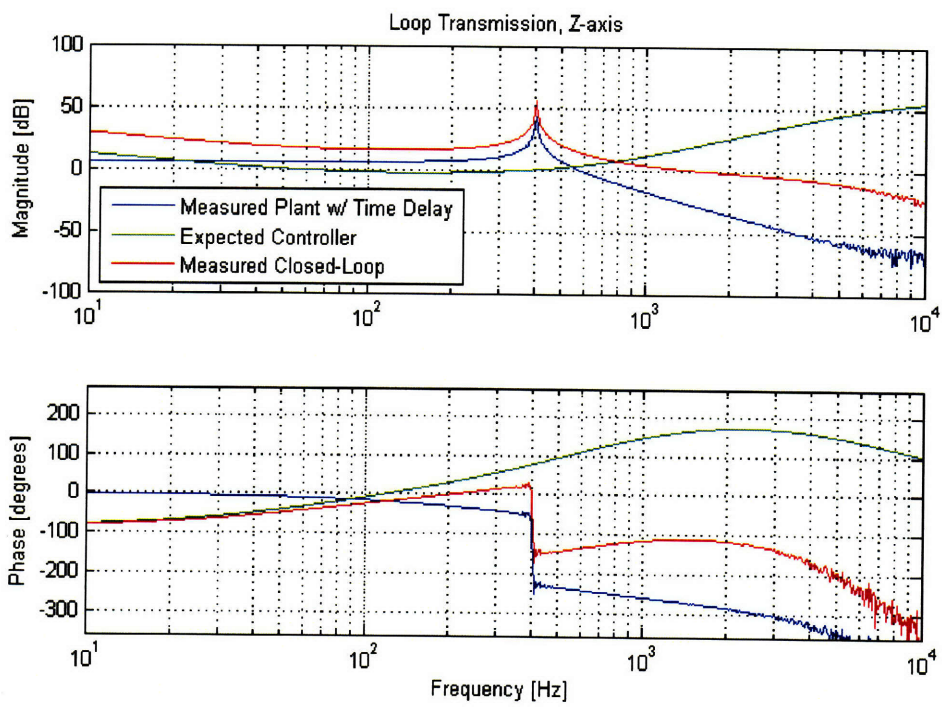


Figure 6-13: Z-axis measured open-loop and closed-loop loop transmission on FPGA control platform.

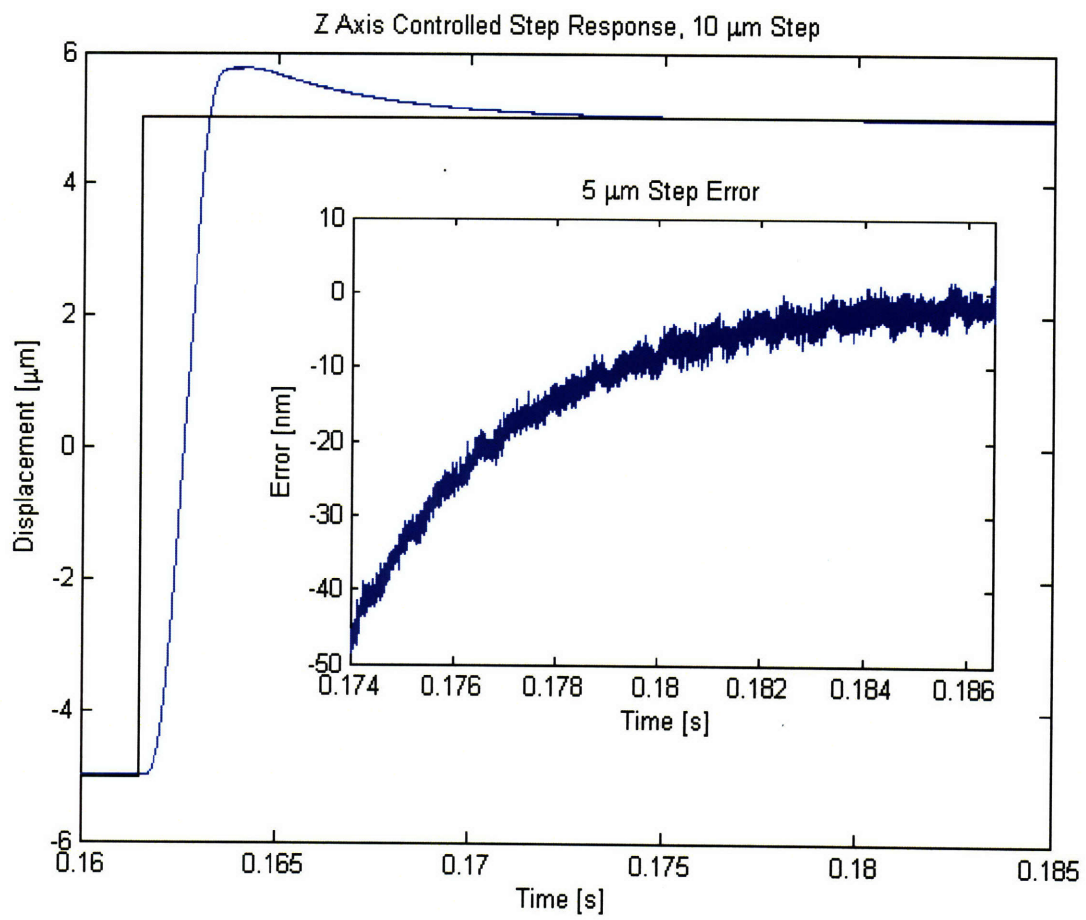


Figure 6-14: Z-axis measured step response and error.



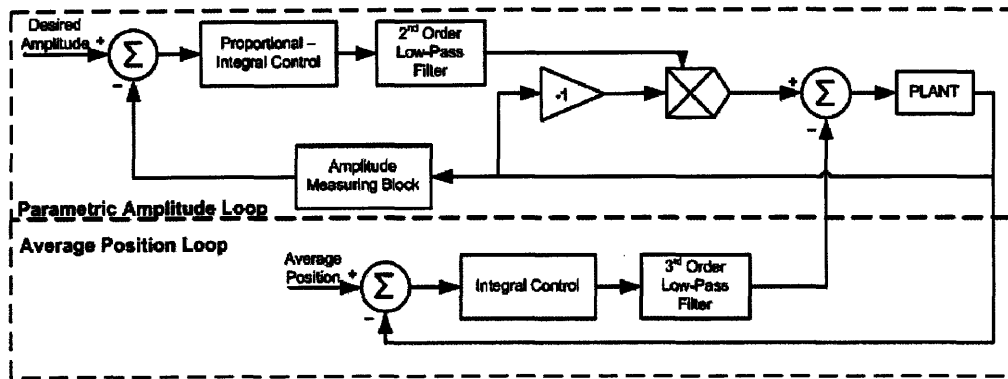


Figure 6-16: Parametric amplitude control loop [12].

terms of control filters and the data path computation complexity.

This chapter presented the characterization of the A/D PCB and the application of the full digital platform to a motion control application. Full 2.5 MHz sampling was demonstrated with sub-nanometer and 111.7 dB filtered dynamic range control. The next chapter discusses overall conclusions and potential future work to improve the previously discussed designs.





# Chapter 7

## Conclusions and Suggestions for Future Work

This thesis provides a step in enabling high-resolution digital control of precision motion systems. This included a complimentary pair of dSPACE interoperable analog interfaces that were revised and characterized so they could be implemented in applications. The work has also presented a design for a high-resolution, high-speed data acquisition and real-time control, FPGA-based digital environment. The design includes a custom A/D PCB design and characterization, as well as high-level software design with arbitrary controller transfer function implementation.

This chapter summarizes the advantages and disadvantages of each digital platform, as well as the ideal applications. In addition, recommendations for future work to improve the designs are included.

### 7.1 Conclusions

#### **dSPACE Interoperable High-Resolution System:**

A pair of high-resolution analog-to-digital and digital-to-analog channels were previously designed in the Precision Motion Control Laboratory by David Otten. These designs were built and tested as an improvement over dSPACE-provided peripher-

als. Each channel is located on a modular PCB with individual power decoupling and digital galvanic isolation. The A/D channel is based on a 800 kSPS converter and dedicated DSP that oversamples and sums. The DSP transmits the sum to the dSPACE DS1103 slave DSP and the average is computed. This effectively increases the A/D resolution from the native 16 bits to 20.1 effective bits, or 15.0  $\mu\text{V}$  RMS on a 20 V range. The D/A channel implements a 16-bit converter as opposed to the native 14-bit dSPACE DS1103 converter. These peripherals are interfaced through a Simulink S-Function and are utilized by the user the same as the native channels. The trade-off for the increased resolution is a decreased closed-loop sampling rate. The native dSPACE system can run up to 100 kHz, however the slave DSP utilization limits the system to an 8 kHz sampling rate.

An asymmetric RMS noise distribution was initially measured across the full input voltage range. The resolution decreased to approximately 16 bits with a -10 V input. The disturbance was due to incorrectly referencing the anti-alias capacitors between the differential input signals into the converter.

Originally the system was designed for 8 A/D channels and 8 D/A channels, however 2 of the D/A channels were not functional. The digital port used for serially loading data to the D/A PCBs only has 7 physical pins although there are 8 logic bits. The breakout PCB was modified to access one additional bit, allowing 7 D/A channels. To use a full 8 D/A channels, an additional port needs to be used which requires additional time to clock data onto, thus decreasing the maximum sampling rate.

The high-resolution system designs of this thesis are utilized in two separate research projects within the Precision Motion Control Laboratory which demonstrate appropriate applications. The first is a vibration isolation system that utilizes three A/D and three D/A channels. The closed-loop bandwidth is 30 Hz. The second system is a 1-DOF positioner for atomic microscopy, which utilizes three A/D channels and one D/A channel with a 200 Hz closed-loop bandwidth.

### **High-Resolution, High-Speed FPGA System:**

An FPGA-based digital platform was designed for high-resolution, high-speed data acquisition and control. A sigma-delta A/D converter was selected because it provides the greatest resolution and dynamic range with a sampling rate in excess of 1 MSPS. No commercial PCB options were available at the time of initial design and thus a circuit was designed to support the selected AD7760 IC A/D converter. The design was based on an evaluation circuit provided by Analog Devices but significant changes were made, such as a dedicated microcontroller with custom firmware to operate the A/D converter, digital galvanic isolation, as well other modifications to features that were not functional as described in the evaluation circuit. The custom PCB achieved 109 dB unfiltered SNR at 2.5 MSPS, or 71  $\mu\text{V}$  RMS on a 20 V range.

I selected a National Instruments FPGA-based digital platform for high-speed acquisition and control computations. The NI LabVIEW language provides high-level functionality that can be easily and quickly implemented, such as direct memory access to quickly transfer large amounts of data. The FPGA board was used within a PXI chassis with a dedicated real-time computer on an NI RTOS. Data acquisition was demonstrated for two 24-bit A/D channels at 2.5 MSPS over 1 second.

Real-time control was implemented on the FPGA, with the control loop closed by the high-resolution D/A PCB designed for the dSPACE system. LabVIEW provides 16-bit PID control, however this control scheme is limiting for more advanced linear control techniques. I designed a LabVIEW virtual instrument to convert an arbitrary transfer function into embedded FPGA logic representing an IIR filter/controller, which included several LabVIEW provided subVIs. This design required a detailed understanding of the LabVIEW implemented quantizers. The analog input and outputs are controlled with finite state machines. The D/A hardware is limited to 625 kHz sample rate due to serial latching constraints, and thus the closed-loop sampling rate is 625 kHz. The control path is processed at 2.5 MHz and downsampled to 625 kHz.

The full digital platform was used to control a flexure-based, electromagnetic

scanner. The 2-DOF scanner, designed by Ian MacKenzie in the Precision Motion Control Laboratory, has the x-axis operated at resonance and the z-axis operated in random access. Experimental results were only collected on the z-axis, although an initial implementation of the parametric amplitude control for the x-axis is presented. The z-axis has a range of 10  $\mu\text{m}$  and the sensing capacitive probe has a range of 40  $\mu\text{m}$ . A z-axis lag, triple-lead controller with 1.5 kHz control closed-loop bandwidth was designed on the native dSPACE DS1103 system and achieved 5.0 nm RMS control unfiltered and 3.2 nm RMS control filtered to 30 kHz. A similar controller was implemented on the FPGA-based system, accommodating for an increased phase loss due to the sigma-delta propagation delay. The system achieved 0.10 nm RMS control filtered to the 1.5 kHz closed-loop bandwidth with 2.5 MHz data acquisition and a 625 kHz closed-loop sampling rate, thus demonstrating the increased resolution and sample rate available with the A/D and FPGA digital environment.

The measured total time delay in the loop is  $T_d + \frac{T_s}{2} = 23.2 \mu\text{s}$ . This introduces a real-time phase loss of 42 degrees at 5 kHz, and is the limiting factor in the real-time control implementation. This system is ideally suited for system bandwidths of 1 kHz or below and requiring high-speed, high-resolution data acquisition.

## 7.2 Suggestions for Future Work

### **dSPACE Interoperable High-Resolution System:**

The high-resolution dSPACE system is successfully being used in several experiments which demonstrates its functionality. The changes discussed here are expected to provide only incrementally increased performance. The first recommended change is to produce a new breakout connector PCB that has wiring included for all 7 channels. The current temporary solution is a soldered jumper that connects the 7th bit. A system demanding 7 operational channels has not been required yet so this has not been a priority.

The next recommended change is to implement a passive low-pass filter on the

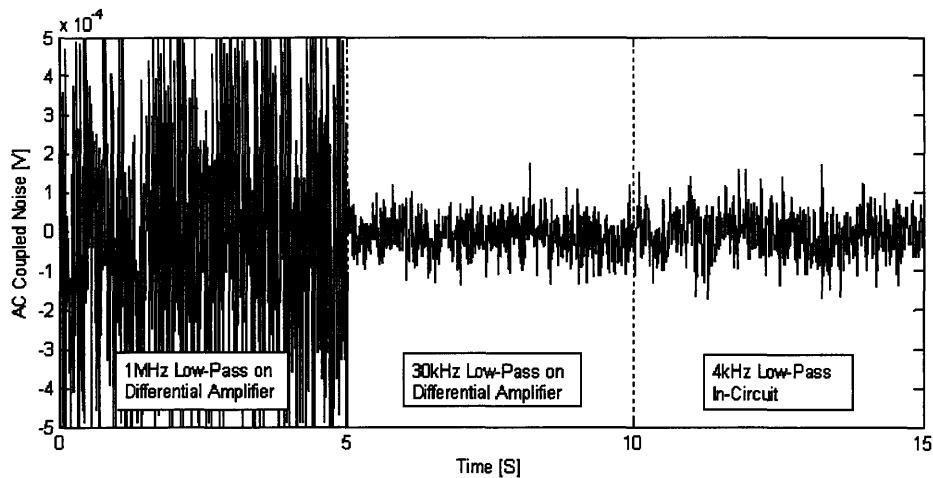


Figure 7-1: High-resolution D/A voltage reference noise and effect of passive low-pass filtering: the voltage reference noise measured with a Tektronix AM502 differential amplifier and 1 MHz low-pass filtering (left), the voltage reference noise measured with the differential amplifier and 30 kHz low-pass filtering (middle), and the voltage reference after a 4 kHz passive low-pass filter measured with the differential amplifier and 1 MHz low-pass filtering (right).

D/A voltage reference. This would require a new PCB revision. Currently there is no filtering between the precision voltage reference and the D/A converter. The characterized D/A PCB only demonstrates 15.1 ENOB and 16 bits can be approached with this filtering. Figure 7-1 shows the current noise present on the LT1019 precision reference used in the design and how it can be improved. The inserted filter decreases RMS noise by over 16 dB and would assist the high-resolution D/A system to approach 16 effective bits.

The last recommendation is to further investigate the theoretical difference in referencing the anti-alias capacitor on the A/D analog input. With capacitors located between the differential signal and ground, an asymmetric noise distribution across the full input range was measured. The noise was significantly improved by replacing these two capacitors with a single capacitor between the two differential signal lines. Literature recommends both methods, however the underlying reasons for the measured difference are not understood.

## High-Resolution, High-Speed FPGA System:

An initial recommendation for this system is to use separate A/D converters for the data acquisition and control. The sigma-delta converter was selected because of its superior dynamic range at the 2.5 MHz sampling rate, however the 10.8  $\mu$ s propagation delay of the converter severely limits the applicability to controlling high bandwidth systems.

If individual A/D PCBs are continued into a new revision, signal termination for the synchronized clocks between channels should be implemented. This would include an output driver. A single IC can be used to clean up the signal and ensure synchronization.

Building and debugging the AD7760 PCB circuit proved to be more challenging and time-consuming than initially expected, however now that an operational PCB has been demonstrated I recommend continuing to build off the AD7760 circuit as the data acquisition converter. Instead of single PCBs for each channel though, the design could be consolidated to incorporate several converters on a single PCB. This would only require a single microcontroller and decrease the total power regulators required, as well as simplify peer channel synchronization. Currently, wires need to be run between separate PCBs to provide a synchronized clocking signal. Multiple channels on a single PCB also simplifies the digital connector to the FPGA. A proprietary National Instruments connector could be designed onto the board to avoid transmitting signals through the NI breakout box. These unguarded lengths introduce cross-coupling between signals, particularly at high switching frequencies.

Changing the analog input to the AD7760 is another consideration. The current design uses a single differential amplifier, built into the AD7760 IC, to accommodate single-supply attenuation, shifting, and anti-aliasing. The input impedance and common mode rejection can be improved by using an instrumentation amplifier configuration, however this would require dual supplies and the related voltage regulation.

If continuing with the AD7760 A/D design, the microcontroller may also be exchanged for a newer product line. Microchip now offers microcontrollers that operate

at 80 MHz with a 32-bit data bus compared to the 30 MHz, 16-bit data bus currently used. This does not affect overall performance though.

An alternative possibility would be to purchase the X3-SDF product from Innovative Integration which has four AD7760 A/D channels with a dedicated FPGA. The cost is \$8,500 and interfacing with the acquired data would require additional resources to develop, whereas the custom PCB has already been implemented with National Instruments hardware and software. The primary benefit for the X3-SDF is multiple channel proven hardware.

If the National Instruments FPGA hardware is used in the further revisions, I recommend that the next design step be the hardware shown in Figure 7-2. The specifications defined by the high-speed AFM scanner discussed in Chapter 3 are no longer essential to meet because the project specifications have been redefined. Instead, a more universal system could be developed. The primary difference is using separate A/D converters for data acquisition and control. The AD7760 sigma-delta converter is best suited for acquisition in light of the inherent propagation delay, and provides the highest dynamic range available at MHz sampling rates. The control A/D would be provided with the 18-bit ADS8482 from Texas Instruments which has a dynamic range of 99 dB at 1 MSPS, compared to 100 dB at 2.5 MSPS for the AD7760. The true benefit is that the ADS8482 has a synchronous approximation register (SAR) converter architecture which means that the associated propagation delay is only the transmission delay. This decreases the time delay due to the digital control environment by at least 10.8  $\mu$ s. If 1 MHz data acquisition is adequate for an application, then just the SAR A/D could be implemented.

Another change to a revised system would be a faster D/A converter. The current design was used because it was previously available. I would alternatively select the AD768 from Analog Devices which is parallel input 16-bit converter with a 25 ns settling time to 16-bit accuracy that can be clocked up to 30 MSPS. The converter has a current output and would require a high-speed external op-amp so the update rate is not limited by the op-amp slew rate. The present D/A converter can only be clocked to output data up to 0.625 MSPS. The true settling time is then determined

by the external op-amp.

While parallel interfaces with the A/D and D/A converters require more FPGA digital I/O, the data bus lines can be multiplexed. Up to three closed-loop channels can share the data bus and maintain a 1 MHz closed-loop control rate. The samples would not be simultaneous, but would offset from one another by 330 ns.

This described system, with dual A/D converters, would have a data acquisition rate of 2.5 MHz, closed-loop sampling rate of 1 MHz and an estimated time delay of 1.2  $\mu\text{s}$  for 5th-order controller computations compared to 23.2  $\mu\text{s}$  measured in the current design. Although the dynamic range of the A/D converter is slightly worse and the output resolution is the same as the implemented design, the system resolution is expected to improve with this described system because the output has a faster settling time to 16-bit accuracy. The present D/A converter has a 4  $\mu\text{s}$  settling time to 16-bit accuracy whereas the recommended D/A converter has a 0.35  $\mu\text{s}$  settling time to 16-bit accuracy. The hardware would be located on a single PCB and include dedicated voltage regulation, digital galvanic isolation, and a National Instruments connector to connect directly into the FPGA.

If the National Instruments hardware is not used in a further revision, I recommend adapting the Thunderstorm architecture designed by Xiaodong Lu [6]. Lu's system has four simultaneous A/D converters, a 100 MHz 64-bit data bus, three 300 MHz DSPs, and four simultaneous D/A converters. This system was not adapted for this initial design because the required timeline was too short and the National Instruments setup provided a quicker, simpler, and more user-friendly implementation. Lu's system has a 1 MHz closed-loop sample rate and a time latency of 1.4  $\mu\text{s}$  for an overall time delay of 1.9  $\mu\text{s}$ . This time allows for four simultaneous A/D conversions, a 20th-order filter, a cosine computation, a sine computation, a square root computation, and finally four simultaneous D/A conversions. The 20th-order filter implemented in LabVIEW FPGA code as described in Section 6.2 would require a latency of at least 3.5  $\mu\text{s}$  to compute. It is possible to write more efficient FPGA code that can compute faster, however the code would not be as universal as the code generator described earlier in Section 5.4.2. The DSP architecture allows



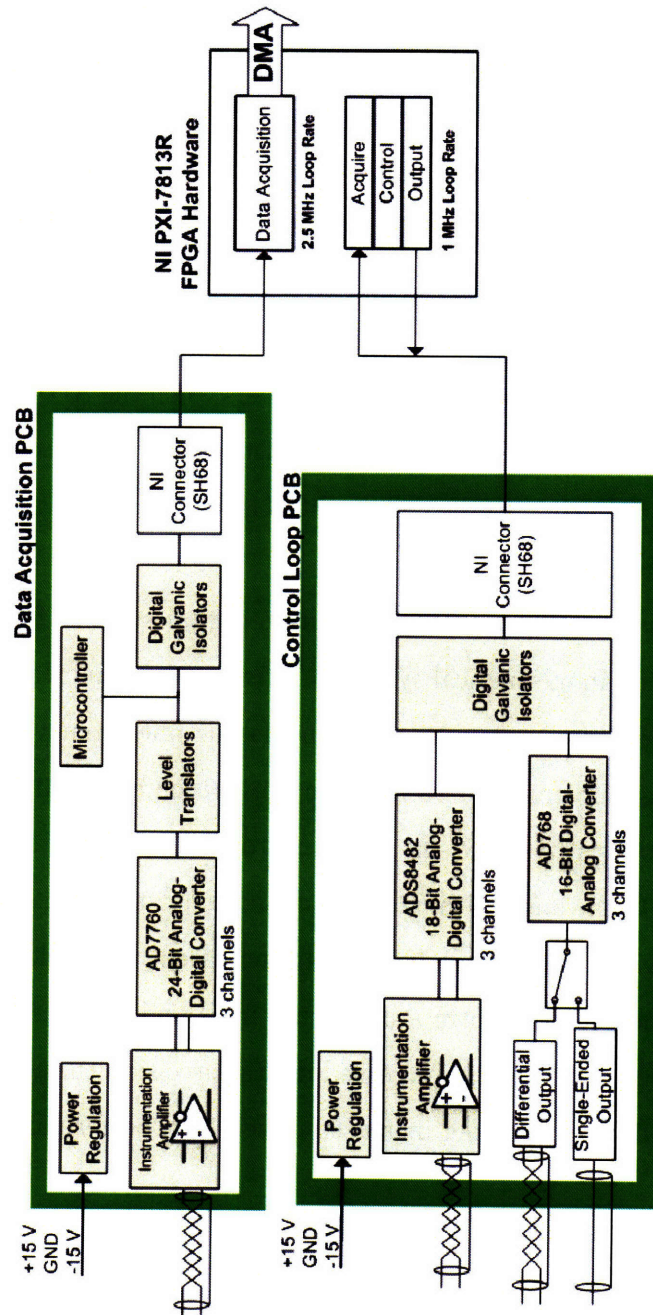


Figure 7-2: Recommended high-resolution, high-speed data acquisition and control environment. Separate acquisition and control A/D channels are used for data acquisition and control. The D/A converter is replaced with one capable of a high output sample rate.

more flexibility with more complex computations, particularly at increasing speeds, and simpler coding. The controller design iteration time is also much lower for DSPs because the FPGA compile time is eliminated, allowing more flexible and efficient development and debugging. The Thunderstorm DSPs could also be updated with 600 MHz TigerSHARC processors from Analog Devices which would decrease the loop latency to approximately 1  $\mu$ s, providing much higher performance than the strictly FPGA based system.

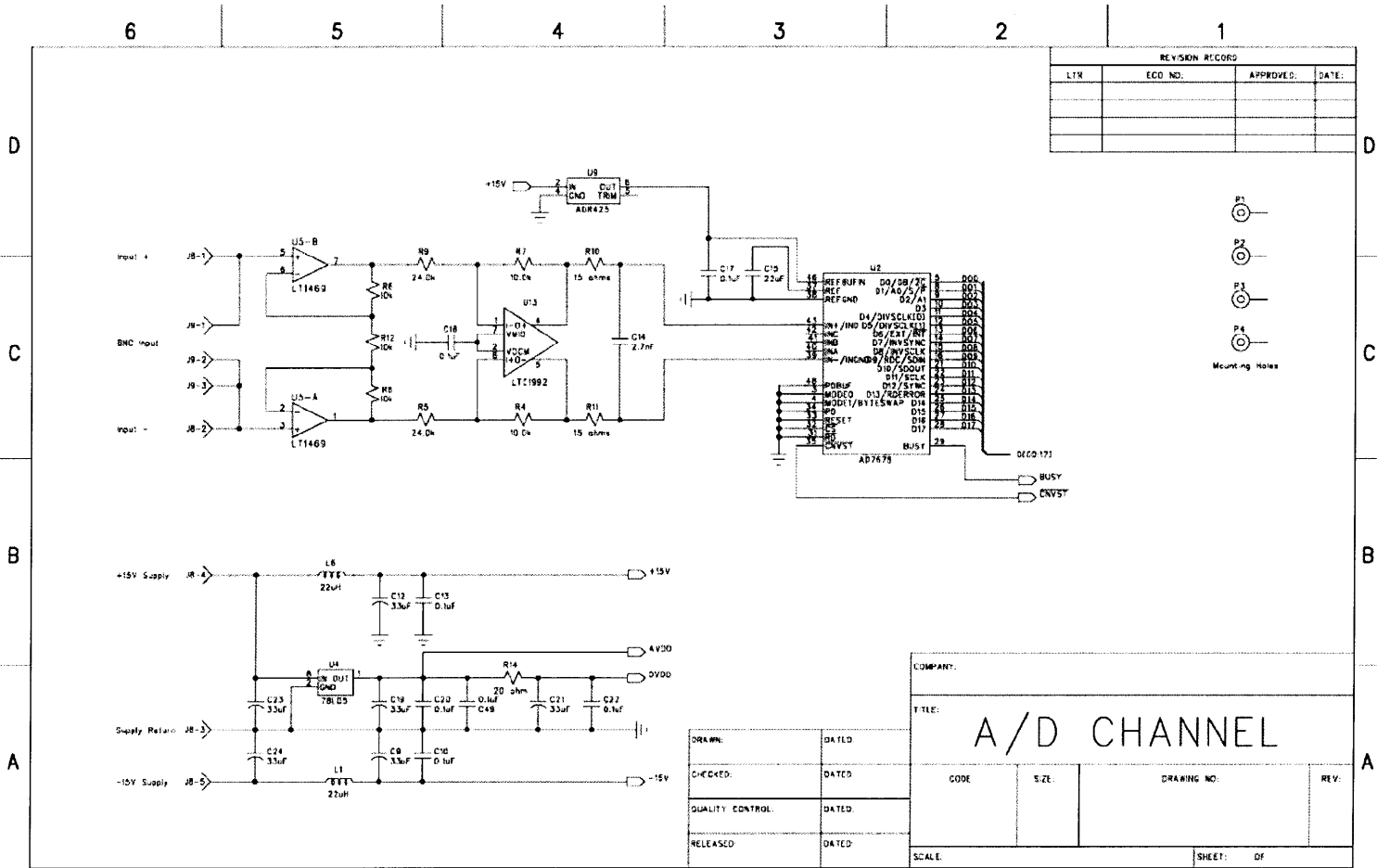
The final recommendation is to further investigate the approximately 100 kHz disturbance measured in the capacitive probe drivers when the AD7760 A/D converter is running. As discussed in Section 6.2, the disturbance is present even when the analog signal from the driver to the custom PCB is not connected so even the transfer mechanism is not completely understood.

This thesis has presented two high-resolution digital systems for increasing available control precision in mechanical systems. This resolution increase generally comes with either a decreased sample rate or an increased time latency. An FPGA-based control system was implemented to improve both resolution and speed and experimental results demonstrated 0.10 nm RMS control filtered to the 1.5 kHz closed-loop bandwidth over 10  $\mu$ m range with 625 kHz closed-loop sampling rate. However, the closed-loop bandwidth could be substantially increased by developing a DSP-based system, such as the Thunderstorm architecture by Xiaodong Lu. In summary, these designs demonstrate significantly increased digital precision that can be quickly and easily implemented, thus allowing increased closed-loop performance.

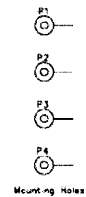
# Appendix A

## Schematics

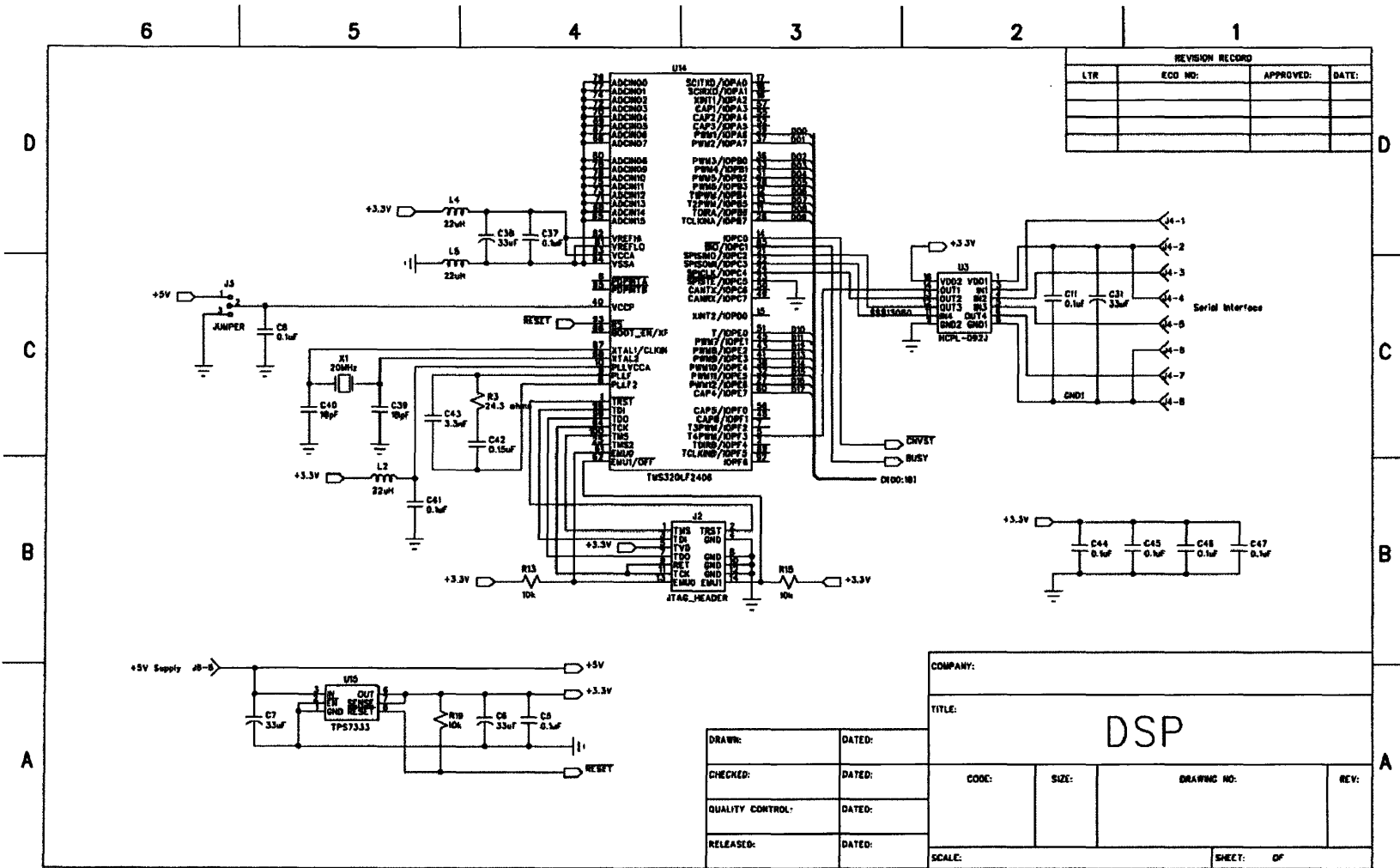
# A.1 dSPACE High-Resolution ADC PCB Schematic



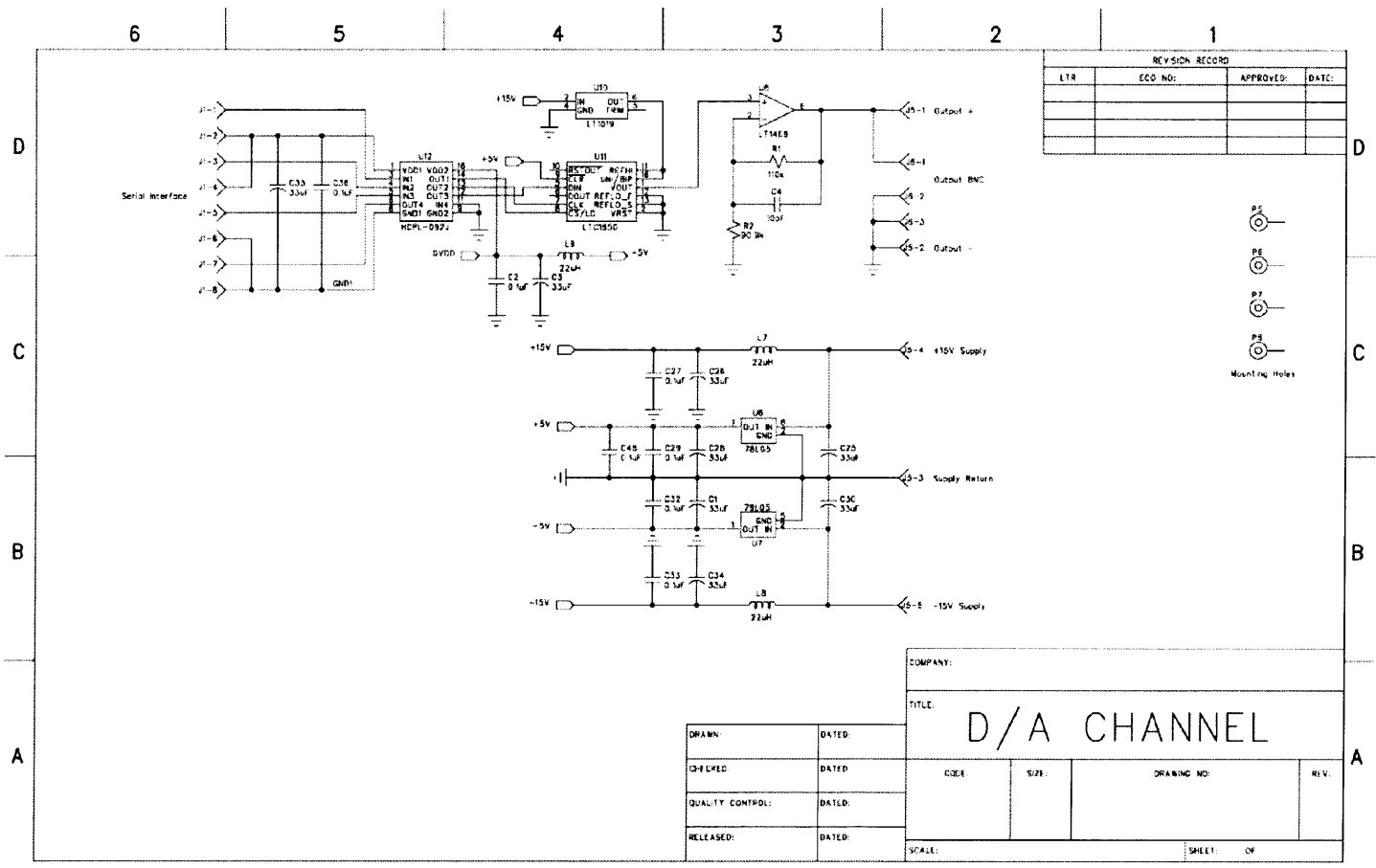
REVISION RECORD			
LTN	ECD NO.	APPROVED:	DATE:



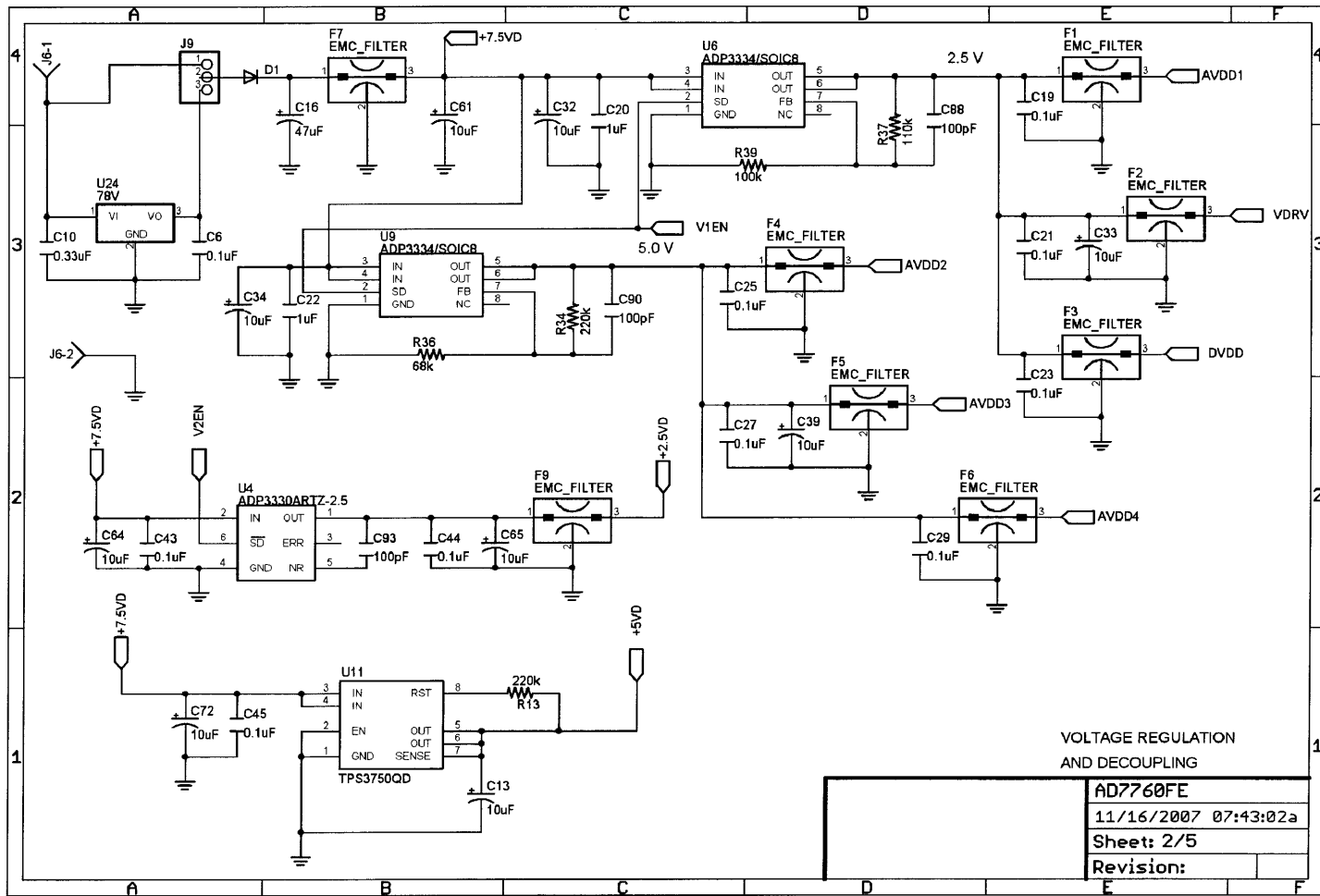
COMPANY:			
TITLE: <b>A/D CHANNEL</b>			
DRAWN:	DATED:	CODE:	SIZE:
CHECKED:	DATED:	DRAWING NO.:	
QUALITY CONTROL:	DATED:	REV.:	
RELEASED:	DATED:	SCALE:	SHEET: DF



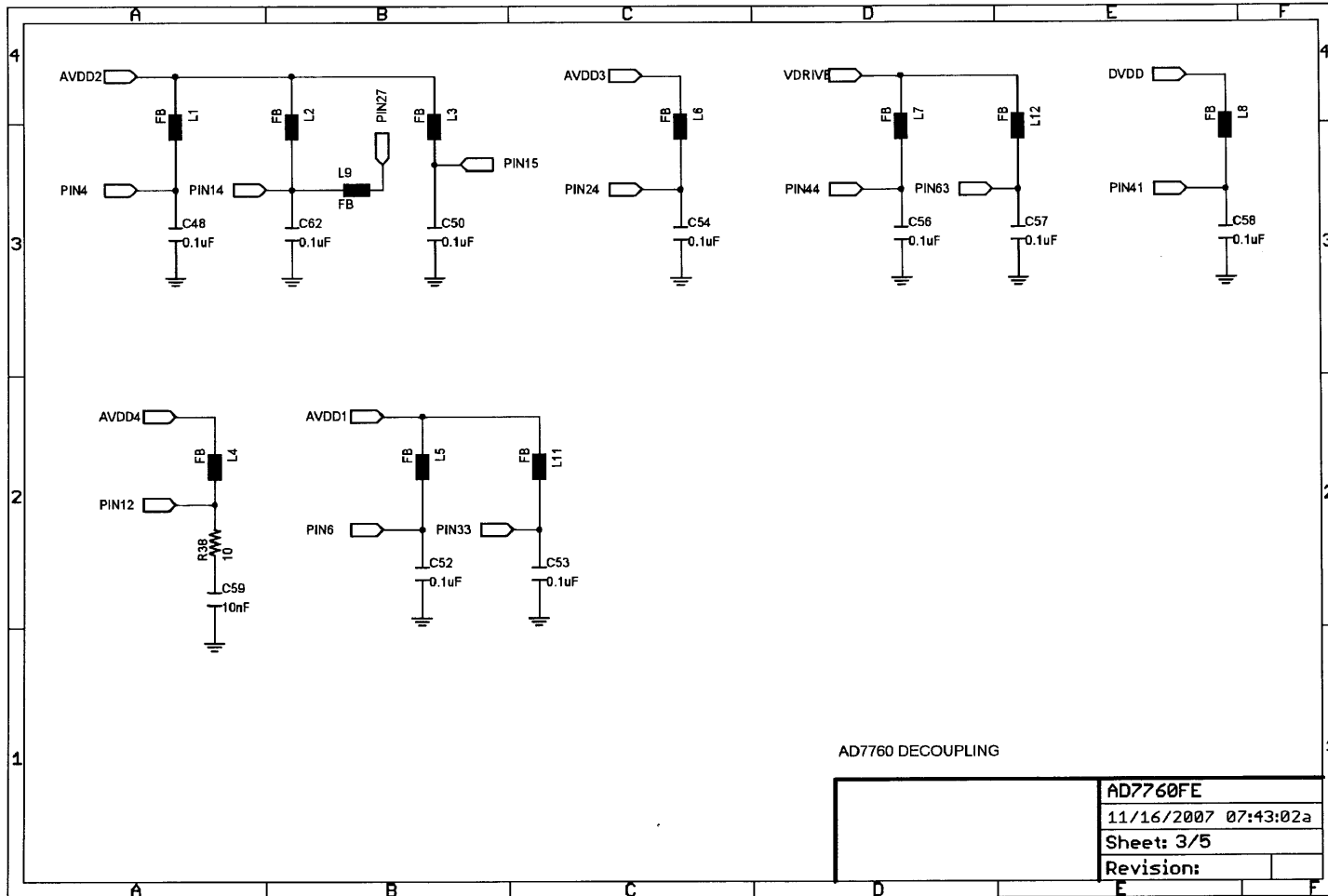
# A.2 dSPACE High-Resolution DAC PCB Schematic





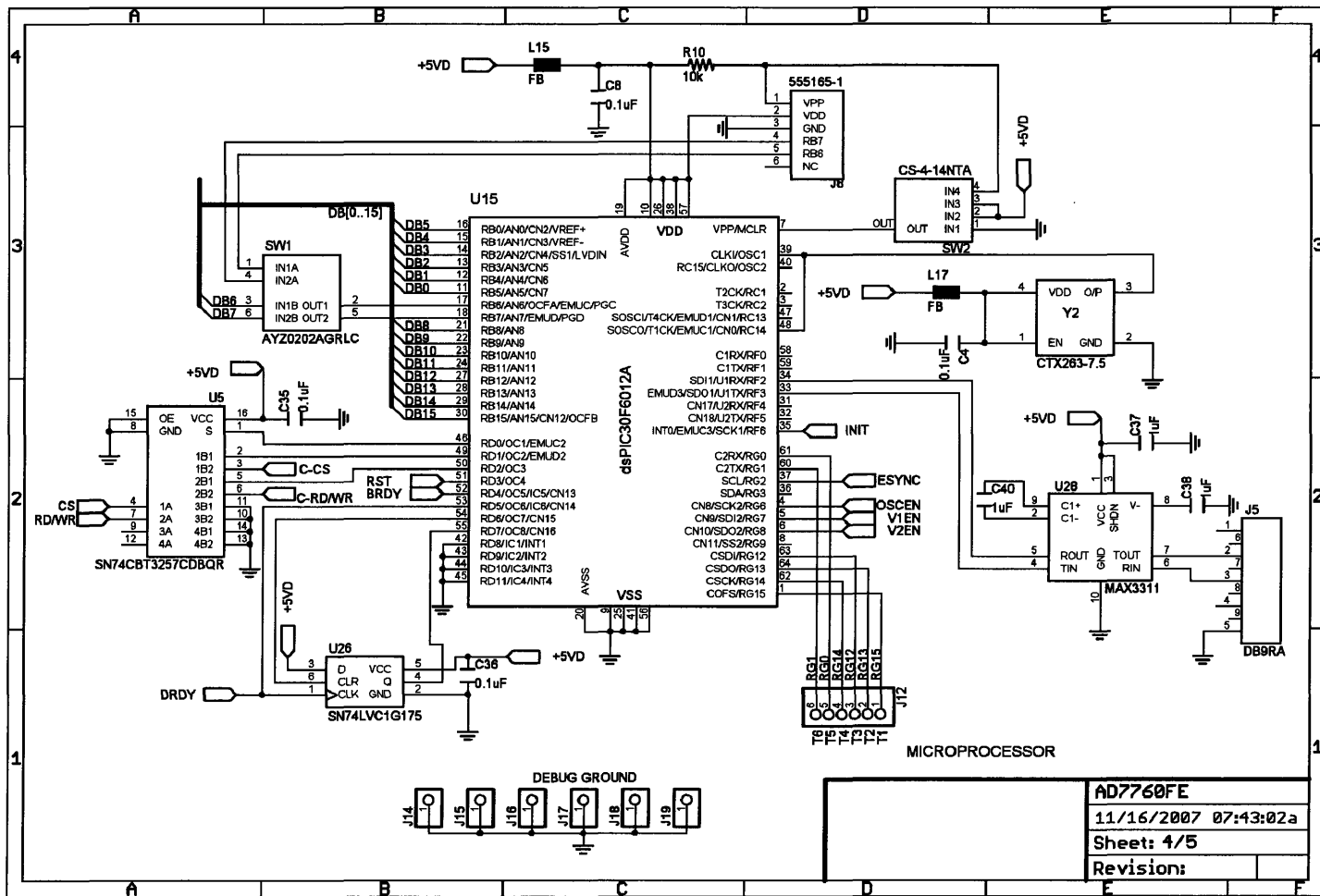


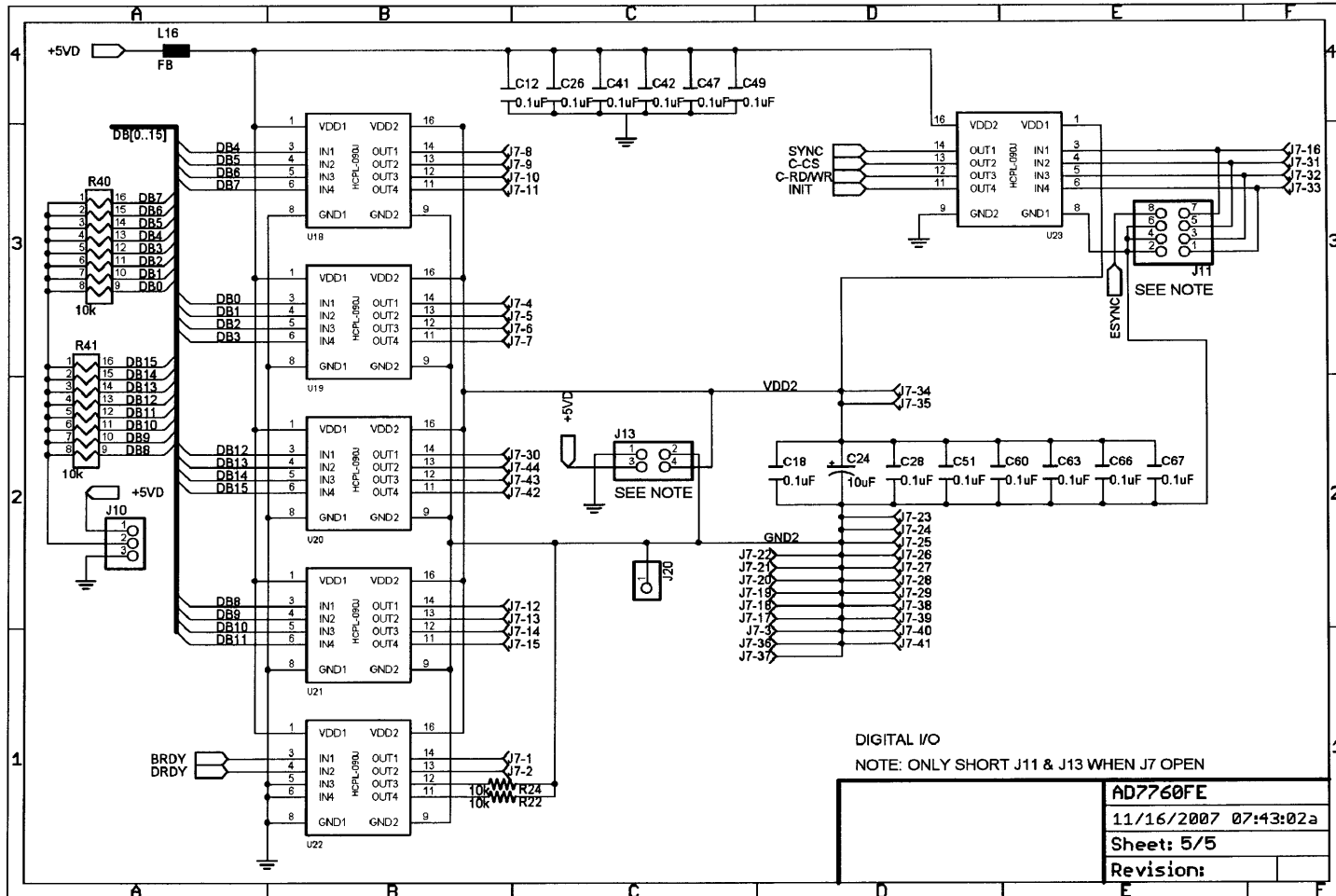




AD7760 DECOUPLING

AD7760FE	
11/16/2007 07:43:02a	
Sheet: 3/5	
Revision:	







# Appendix B

## AD7760 Microcontroller Firmware

### C Source Code:

```
/*
// Source code for AD7760 PCB microcontroller
// Written by Aaron Gawlik 11-13-07
// This includes debugging and testing interfaces
*/

#include <p30f6012A.h> //dsPIC30F6012A microcontroller
#include <uart.h> //Microchip UART RS-232 interface

//received UART global variable data register
unsigned char RX_data;

extern void asmInitFunction(void);
extern void testMIPS(void);

void _ISR _U1RXInterrupt(void); //UART1 receive ISR
void _ISR _INT0Interrupt(void); //interrupt0 ISR
void _ISR _T1Interrupt(void); //timer1 inerrupt ISR
void run_init(void); //initialization macro
void UARTConfig(void); //UART configuration macro
void debugFcn(void); //macro for debug purposes

int main() {
    SRbits.IPL = 4; //set CPU interrupt priority to level 4

    //define inputs/outputs
    ADCON1bits.ADON = 0; //Allow PORTB to be digital I/O
    ADPCFG = 0xFFFF;

    TRISB = 0xFFFF; //data bus (RB) is all inputs
    TRISD = 0xFFA0; //PIC input(1)/output(0) pins, see design notes
```

```

TRISF = 0xFFFF; //Interrupt input pins
TRISG = 0x0000; //used for debugging

LATG = 0x0000;
LATGbits.LATG6 = 1; //oscillator on, enable high
LATGbits.LATG7 = 0; //AD7760 votlage regulators on, enable low
LATGbits.LATG8 = 1; //level translators voltage regulator on, enable high
LATGbits.LATG2 = 1; //SYNC sim, high
LATGbits.LATG15 = 1; //SYNC sim, high

LATDbits.LATD1 = 1; //AD7760 CS (0)
LATDbits.LATD2 = 1; //AD7760 RD/WR (0)
LATDbits.LATD0 = 0; //analog switch select to FPGA (1)
LATDbits.LATD3 = 1; //AD7760 RESET command (1)
LATDbits.LATD4 = 0; //BRDY (AD7760FE board not initialized) (0)
LATDbits.LATD6 = 1; //CLR (flip-flop, active-low) (1)
//LATD = 0x0049;

UARTConfig();

//setup interrupt on INT0
INTCON1bits.NSTDIS = 1; //disable nested loops (1)
IPC0bits.INT0IP = 7; //set INT0 priority to level 7 (7)
INTCON2bits.INT0EP = 1; //edge detect on rising (positive) edge (0)
IFS0bits.INT0IF = 0; //clear interrupt flag status bit (0)
IEC0bits.INT0IE = 1; //INT0 enabled (1)

//Sleep();

//sleep/idle mode and wait to re-initialize on interrupt
while (1) {
    ;
}

//-----
/* INT0 ISR */
void _ISR _INT0Interrupt(void) {
    ;;
    if(PORTFbits.RF6 == 0) {
        LATGbits.LATG1 = ~LATGbits.LATG1;
        LATDbits.LATD4 = 0; //BRDY
        run_init();

        //setup interrupt on T1 (check DRDY and set BRDY)
        if (0) {
            IEC0bits.T1IE = 1; //enable timer1 interrupt
            IFS0bits.T1IF = 0; //clear timer1 flag
            IPC0bits.T1IP = 5; //interrupt priority
            T1CONbits.TCS = 1;
            T1CONbits.TSYNC = 0;
            TMR1 = 0; //reset timer1
            PR1 = 0xFFFF; //timer1 period, appx 3000/29*256 = 26 ms
            T1CONbits.TCKPS = 0x2; //timer1 prescaler set to 1:256,

```

```

        T1CONbits.TON = 1;
    }
}
IFS0bits.INT0IF=0; //clear interrupt flag
//Sleep();
}

void _ISR_T1Interrupt(void) {
    LATGbits.LATG0 = ~LATGbits.LATG0;

    if (PORTDbits.RD7 == 1) {
        LATDbits.LATD4=1; } //BRDY
    else {
        LATDbits.LATD4=0; } //BRDY

    LATDbits.LATD6 = 0; //CLR (flip-flop, active-low) (1)
    ;;;;
    LATDbits.LATD6 = 1; //CLR (flip-flop, active-low) (1)

    IFS0bits.T1IF=0;
    Sleep();
}

/* This is UART1 receive ISR */
void _ISR_U1RXInterrupt(void)
{
    // Read the receive buffer until empty
    while( DataRdyUART1() ) {
        RX_data = ReadUART1();

        if (RX_data == 97) { //'a' received, turn on oscillator
            LATGbits.LATG6 = ~LATGbits.LATG6; }
        else if (RX_data == 98) { //'b' received, run_init
            LATGbits.LATG7 = ~LATGbits.LATG7; }
        else if (RX_data == 99) { //'c' received, enter sleep mode
            IFS0bits.U1RXIF = 0; //clear interrupt flag
            Sleep();
        }
        else if (RX_data == 100) { //'d' received, run_init
            run_init();
        }
        else if (RX_data == 101) { //'e' received, testMIPS
            testMIPS();
        }
        else if (RX_data == 104) { //'h' received, flip RST
            LATDbits.LATD3 = ~LATDbits.LATD3;
        }
        else if (RX_data == 106) { //'j' received, flip CS
            LATDbits.LATD1 = ~LATDbits.LATD1;
        }
        else if (RX_data == 108) { //'l' received, twiddle Port G
            LATGbits.LATG1 = ~LATGbits.LATG1;
        }
        else if (RX_data == 109) { //'m' received, test FPGA port

```

```

        PORTBbits.RB0 = ~PORTBbits.RB0;
    }
    else if (RX_data == 110) { //'n' received, test FPGA port
        PORTBbits.RB1 = ~PORTBbits.RB1;
    }
    else if (RX_data == 111) { //'o' received, test FPGA port
        PORTBbits.RB2 = ~PORTBbits.RB2;
    }
    else if (RX_data == 112) { //'p' received, test FPGA port
        PORTBbits.RB3 = ~PORTBbits.RB3;
    }
    else if (RX_data == 113) { //'q' received, test FPGA port
        PORTBbits.RB4 = ~PORTBbits.RB4;
    }
    else if (RX_data == 114) { //'r' received, test FPGA port
        PORTBbits.RB5 = ~PORTBbits.RB5;
    }
    else if (RX_data == 115) { //'s' received, test FPGA port
        PORTBbits.RB6 = ~PORTBbits.RB6;
    }
    else if (RX_data == 116) { //'t' received, test FPGA port
        PORTBbits.RB7 = ~PORTBbits.RB7;
    }
    else if (RX_data == 117) { //'u' received, twiddle Port G
        LATB = ~LATB;
    }
    else if (RX_data == 118) { //'v' received, test FPGA port
        PORTBbits.RB8 = ~PORTBbits.RB8;
    }
    else if (RX_data == 119) { //'w' received, test FPGA port
        PORTBbits.RB9 = ~PORTBbits.RB9;
    }
    else if (RX_data == 120) { //'x' received, test FPGA port
        PORTBbits.RB10 = ~PORTBbits.RB10;
    }
    else if (RX_data == 121) { //'y' received, test FPGA port
        PORTBbits.RB12 = ~PORTBbits.RB12;
    }
    else if (RX_data == 122) { //'z' received, test FPGA port
        PORTBbits.RB11 = ~PORTBbits.RB11;
    }
    else if (RX_data == 65) { //'A' received, test FPGA port
        PORTBbits.RB13 = ~PORTBbits.RB13;
    }
    else if (RX_data == 66) { //'B' received, test FPGA port
        PORTBbits.RB14 = ~PORTBbits.RB14;
    }
    else if (RX_data == 67) { //'C' received, test FPGA port
        PORTBbits.RB15 = ~PORTBbits.RB15;
    }
    else if (RX_data == 68) { //'D' received, twiddle G port
        LATG = ~LATG;
    }
    else if (RX_data == 69) { //'E' received, twiddle G port

```



```

        LATGbits.LATG8 = ~LATGbits.LATG8;
    }
}
IFS0bits.U1RXIF = 0;
}

/* AD7760 setup procedure */
void run_init(void) {

LATDbits.LATD1 = 1; //AD7760 CS
LATDbits.LATD2 = 1; //AD7760 RD/WR
LATDbits.LATD0 = 0; //analog switch select (1 = FPGA)

LATB = 0x0000; //set Port B
TRISB = 0x0000; //data bus (RB) is all outputs

//apply power
//start osc, applying MCLK
//take RESET low for minimum of 1 MCLK
LATDbits.LATD3 = 0; //AD7760 RESET command (0 = reset)
;
;
LATDbits.LATD3 = 1; //AD7760 RESET command (0 = reset)
//wait for minimum of 2 MCLK
;
;
;
;

asmInitFunction();

LATB = 0x0000;
TRISB = 0xFFFF; //data bus (RB) is all inputs

//Removed for debugging, put back in when connected to FPGA
//LATD = 0x0019; //return analog switch select to FPGA and enable BRDY
//LATD=0x005E; //RD/WR & CS high, select low (PIC maintains control)

    LATDbits.LATD6 = 0; //CLR (flip-flop, active-low) (1)
    LATDbits.LATD6 = 1; //CLR (flip-flop, active-low) (1)

int n = 0;
for (n=0;n<100;n++) {
    ; }
if (PORTDbits.RD7 == 1) {
    LATDbits.LATD4=1; //BRDY
    LATDbits.LATD0=1;
}
}

void UARTConfig(void) {
    //Initialize configuration variables
    unsigned int baudvalue;

```

```

unsigned int U1MODEvalue; //config1
unsigned int U1STAvalue; //config2

//Specify baud rate
baudvalue = 194; //defines 9.6kbs at 30Mhz clock
//USE 194 for 9.6kbs at 30MIPS

ConfigIntUART1(UART_RX_INT_EN & UART_RX_INT_PR7 & UART_TX_INT_DIS);
//(recieve interrupt enable, priority 7, transmit interrupt disable)

U1MODEvalue = UART_EN &
    UART_IDLE_STOP & UART_ALTRX_ALTTX & UART_EN_WAKE
    & UART_DIS_LOOPBACK & UART_DIS_ABAUD
    & UART_NO_PAR_8BIT & UART_1_STOPBIT;
//(enable,
// stop in idle mode, communication through alternate pins,enable on start,
// disable loopback, disable autobaud,
// no parity 8-bit, 1 stopbit)

U1STAvalue = UART_TX_PIN_NORMAL & UART_TX_ENABLE
    & UART_INT_RX_CHAR & UART_ADR_DETECT_DIS
    & UART_RX_OVERRUN_CLEAR;
//(Interrupt on transfer of every character to TSR, UART TX pin operates normally,
// Transmit enable, Interrupt on every char received, address detect disable,
// Rx buffer Over run status bit clear)

//Open the UART
OpenUART1(U1MODEvalue, U1STAvalue, baudvalue);
}

```

### Assembly Source Code:

```

;
; file: asmfun.s
;
.global _asmInitFunction ;setup AD7760 function
.global _testMIPS ;sample AD7760 function

_asmInitFunction:
mov #0x0020, w2 ;register 1 address (0x0001)
mov #0x0006, w3 ;register 1 data (0x0006 -> 0x0018)
mov #0x0010, w4 ;register 2 address (0x0002)
mov #0x0010, w5 ;register 2 data (0x0002)

;modified values for wrong xlator pinout
;mov #0x0080, w2 ;register 1 address
;mov #0x4018, w3 ;register 1 data, with 4x decimation
;mov #0x0040, w4 ;register 2 address
;mov #0x0040, w5 ;register 2 data

;write sequence 1

```

```

mov #0x0006, w1 ;take CS low for 4 ICLK
mov w1, 0x02D6
nop
mov #0x000E, w1 ;take CS low for 4 ICLK
mov w1, 0x02D6
nop
nop
mov w4, 0x02C8 ;write reg 2 address to port B
mov #0x000C, w1 ;take CS low for 4 ICLK
mov w1, 0x02D6
nop ;1
nop ;2
nop ;3
nop ;4
nop ;5
nop ;6
mov #0x000E, w1 ;take CS high for 4 ICLK
mov w1, 0x02D6
nop ;1
nop ;2
nop ;3
nop ;4
nop ;5
nop ;6
mov w5, 0x02C8 ;write reg 2 data to port B
mov #0x000C, w1 ;take CS low for 4 ICLK
mov w1, 0x02D6
nop ;1
nop ;2
nop ;3
nop ;4
nop ;5
nop ;6
mov #0x000E, w1 ;take CS high for 4 ICLK
mov w1, 0x02D6
nop ;1
nop ;2
nop ;3
nop ;4
nop ;5
nop ;6

;write sequence 2
mov w2, 0x02C8 ;write reg 1 address to port B
mov #0x000C, w1 ;take CS low for 4 ICLK

```

```

mov w1, 0x02D6
nop ;1
nop ;2
nop ;3
nop ;4
nop ;5
nop ;6
mov #0x000E, w1 ;take CS high for 4 ICLK
mov w1, 0x02D6
nop ;1
nop ;2
nop ;3
nop ;4
nop ;5
nop ;6
mov w3, 0x02C8 ;write reg 1 data to port B
mov #0x000C, w1 ;take CS low for 4 ICLK
mov w1, 0x02D6
nop ;1
nop ;2
nop ;3
nop ;4
nop ;5
nop ;6
mov #0x000E, w1 ;take CS high for 4 ICLK
mov w1, 0x02D6
nop ;1
nop ;2
nop ;3
nop ;4
nop ;5
nop ;6

clr w1
clr w2
clr w3
clr w4
clr w5

return

```

```

_testMIPS: ;Function to test MIPS - twiddles cs bit, demonstrates 29.6 MIPS
mov #0x0018, w5 ;cs low and rd/wr low
mov #0x001E, w6 ;cs high and rd/wr high

```

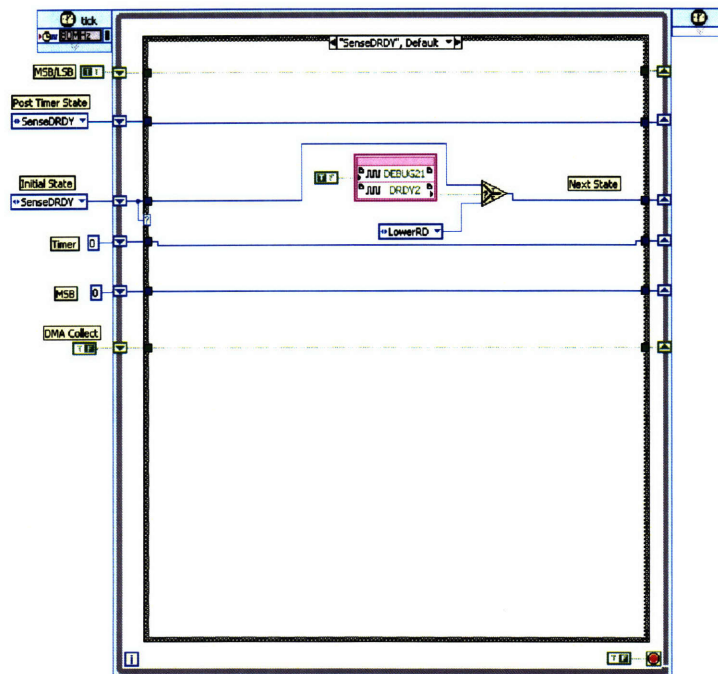
```
mov w6, 0x02D6 ;cs high and rd/wr high
mov w5, 0x02D6 ;cs low and rd/wr low
mov w6, 0x02D6 ;cs high and rd/wr high
mov w5, 0x02D6 ;cs low and rd/wr low
mov w6, 0x02D6 ;cs high and rd/wr high
mov w5, 0x02D6 ;cs low and rd/wr low
mov w6, 0x02D6 ;cs high and rd/wr high
mov w5, 0x02D6 ;cs low and rd/wr low
mov w6, 0x02D6 ;cs high and rd/wr high
mov w5, 0x02D6 ;cs low and rd/wr low
mov w6, 0x02D6 ;cs high and rd/wr high
mov w5, 0x02D6 ;cs low and rd/wr low
mov w6, 0x02D6 ;cs high and rd/wr high
mov w5, 0x02D6 ;cs low and rd/wr low
mov w6, 0x02D6 ;cs high and rd/wr high
mov w5, 0x02D6 ;cs low and rd/wr low
mov w6, 0x02D6 ;cs high and rd/wr high
mov w5, 0x02D6 ;cs low and rd/wr low
return
```

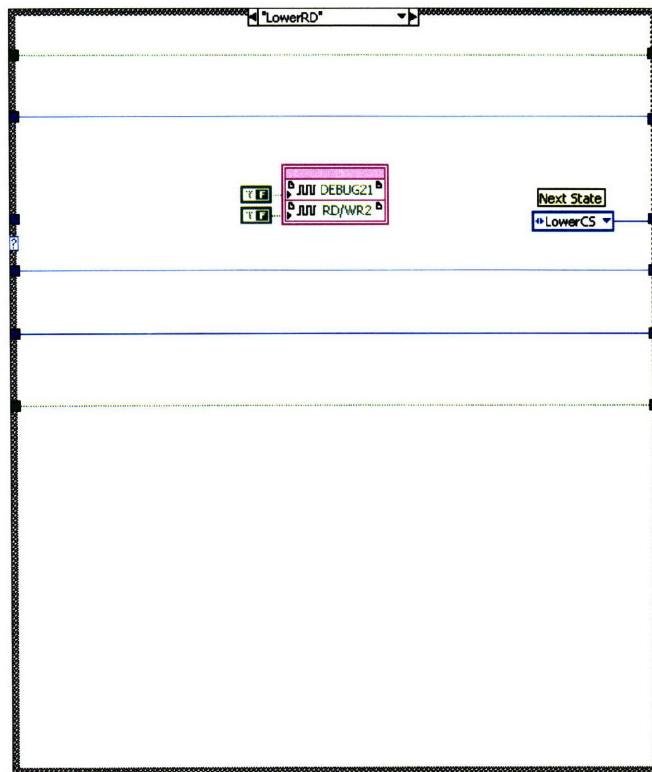
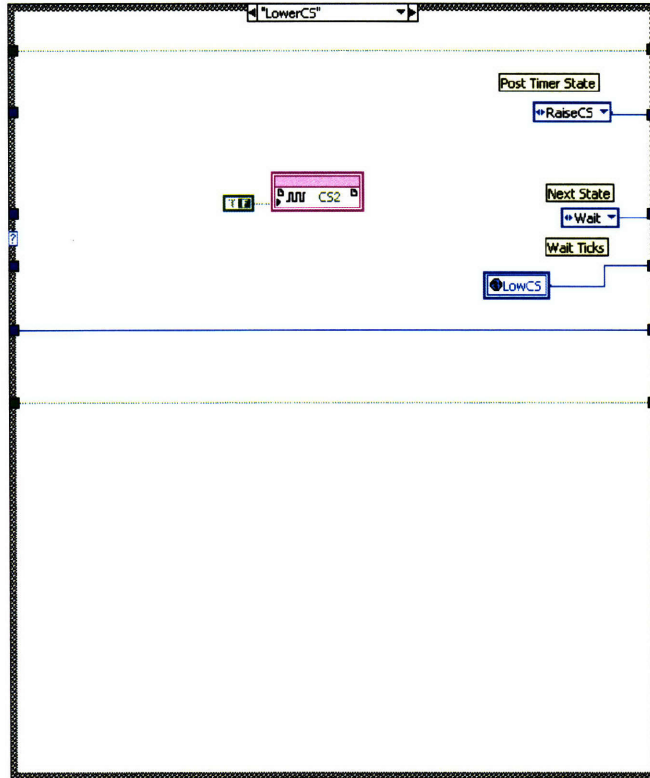


# Appendix C

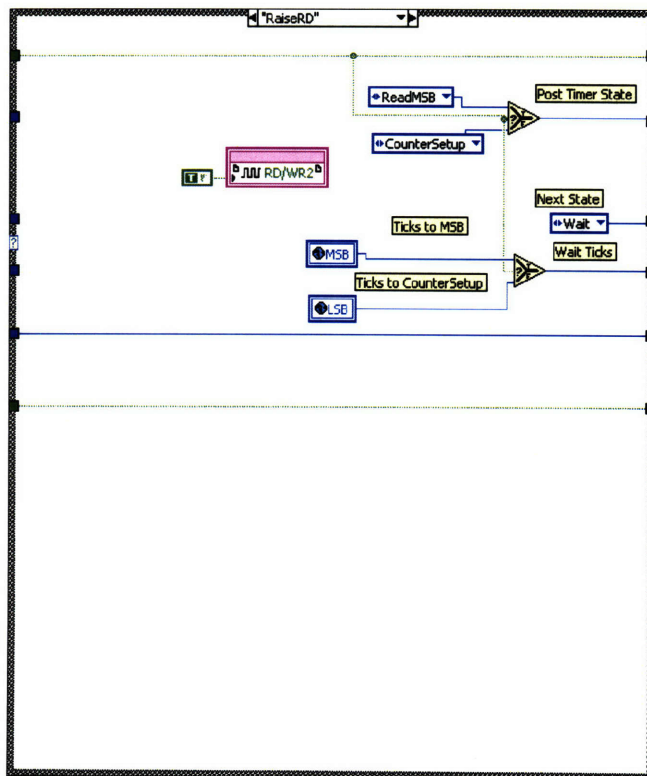
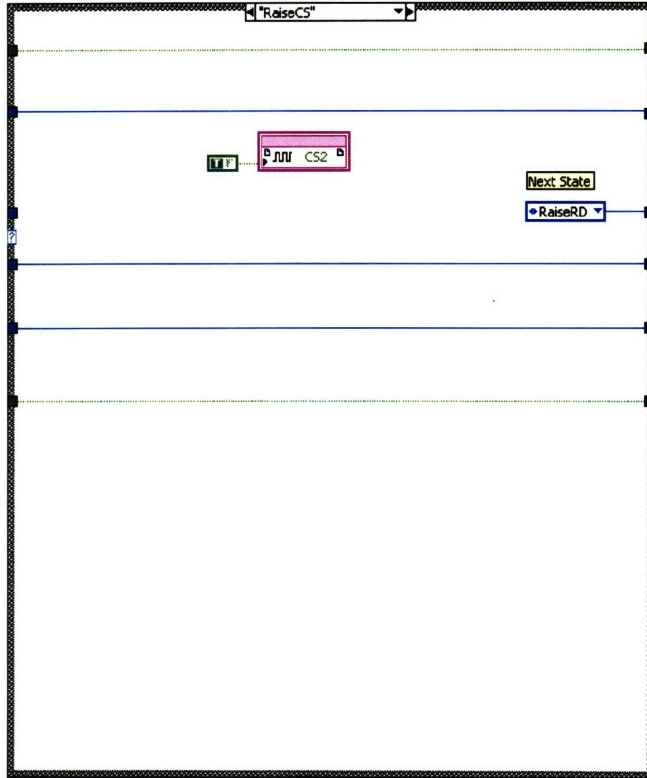
## LabVIEW FPGA Code

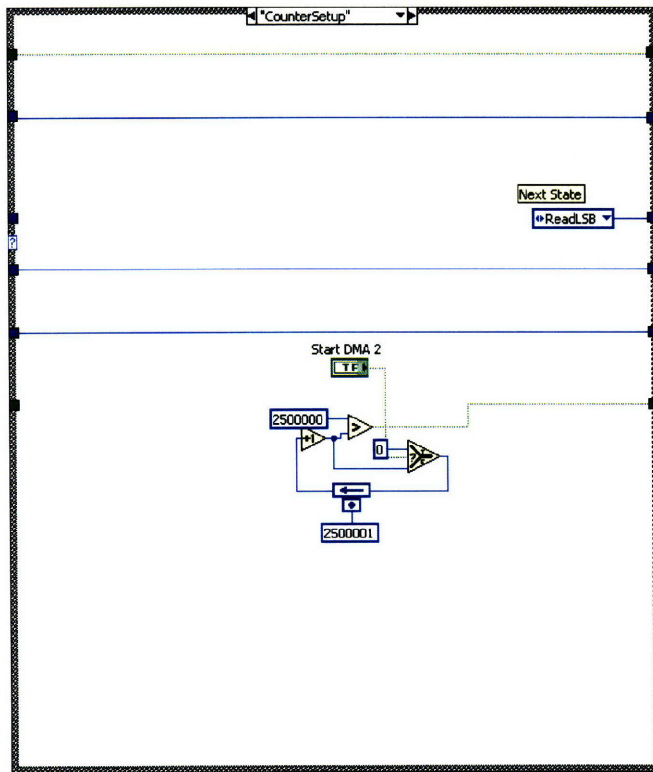
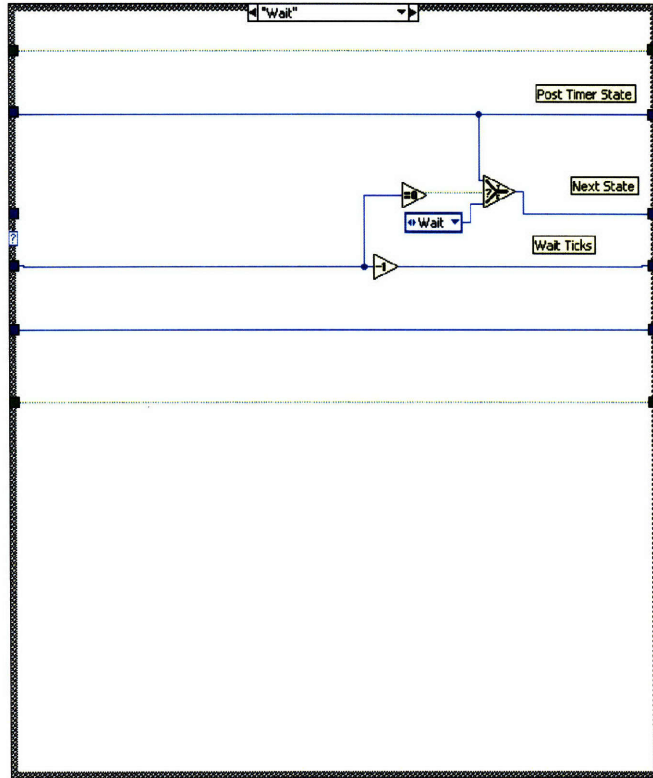
### C.1 A/D Acquisition State Machine

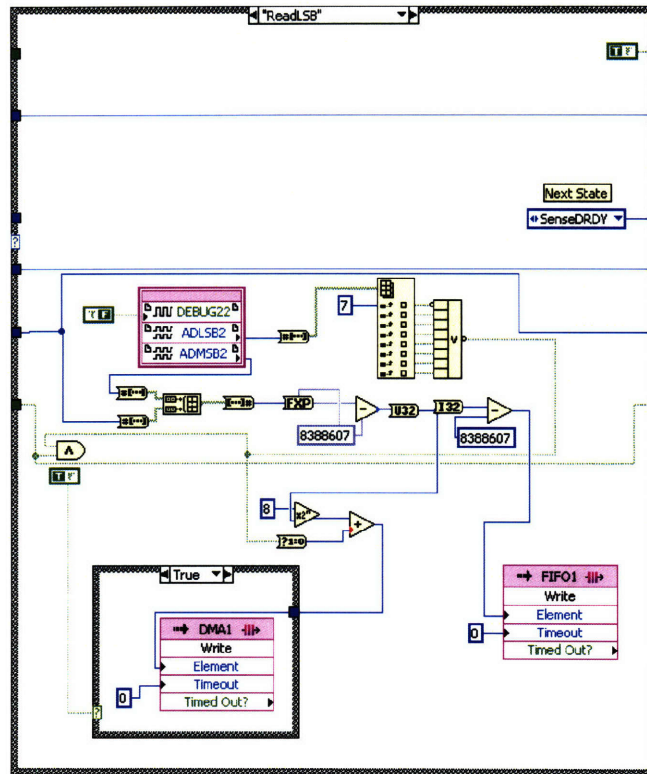
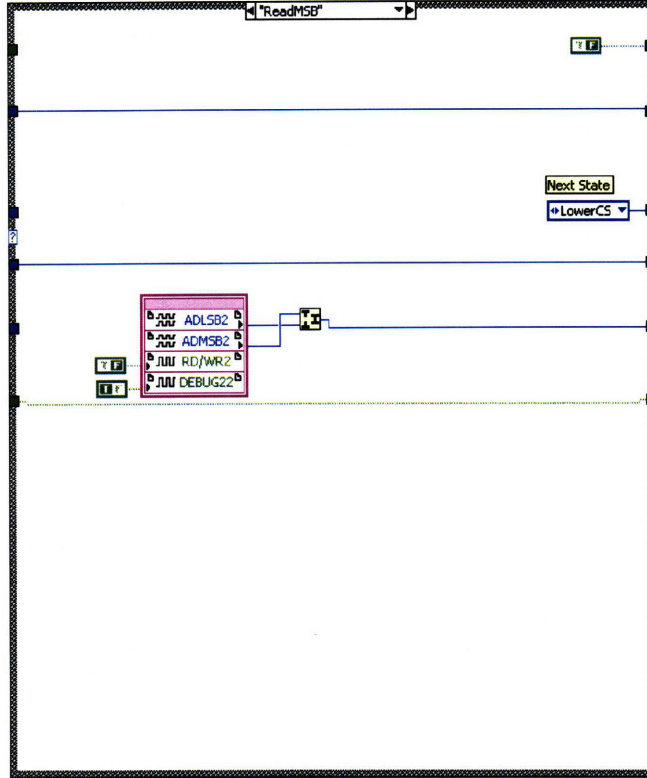




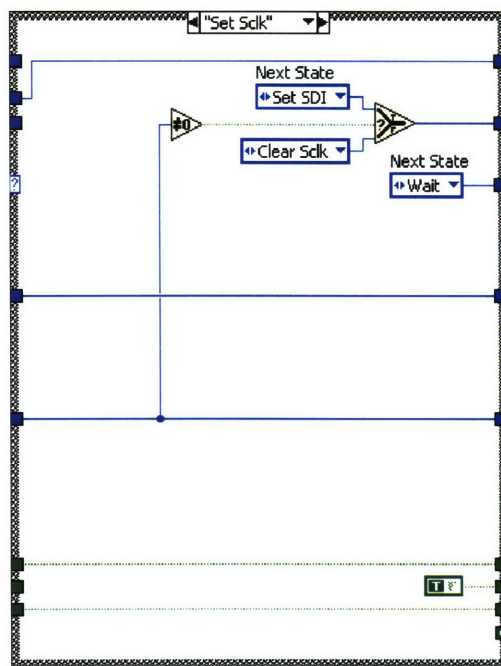
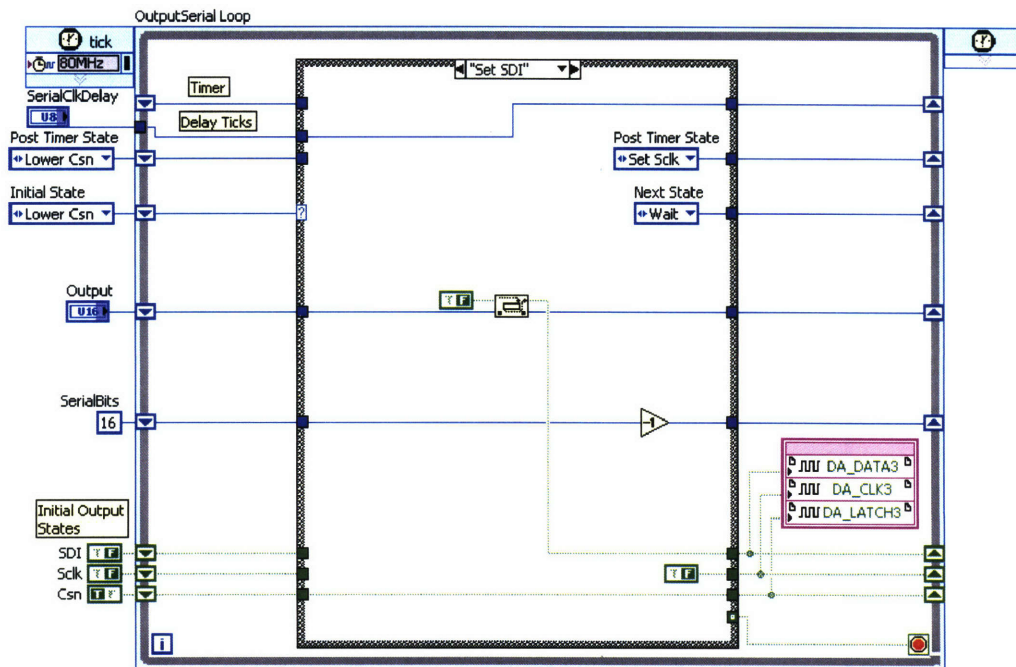


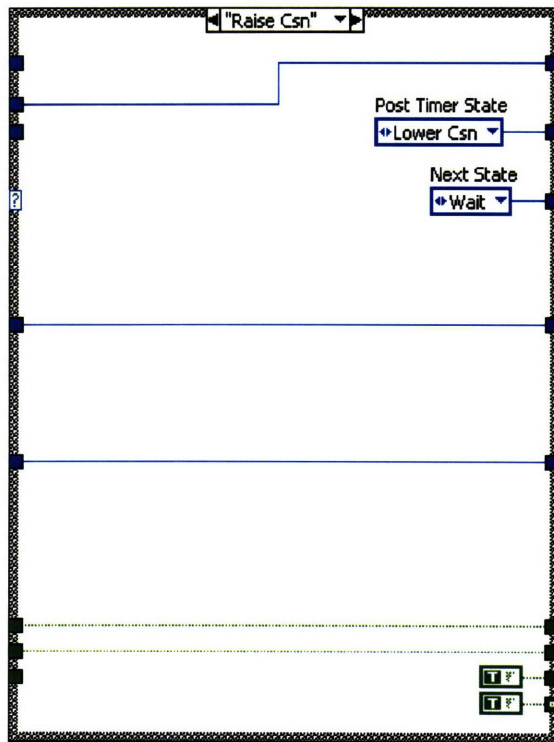
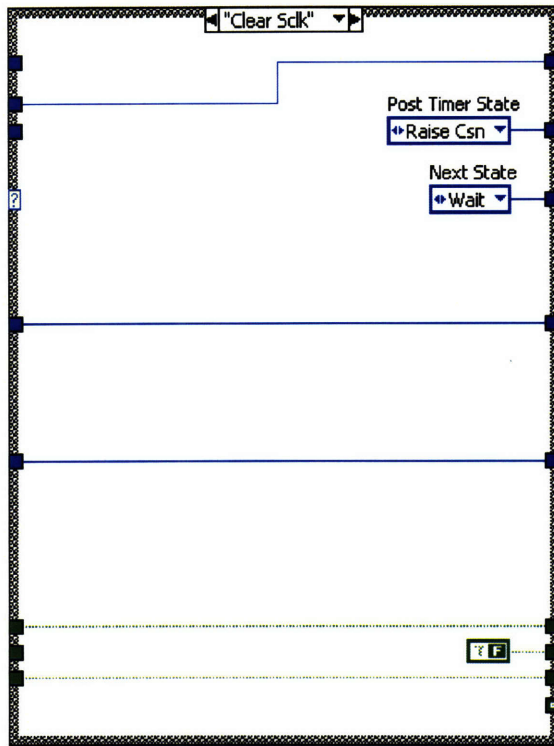


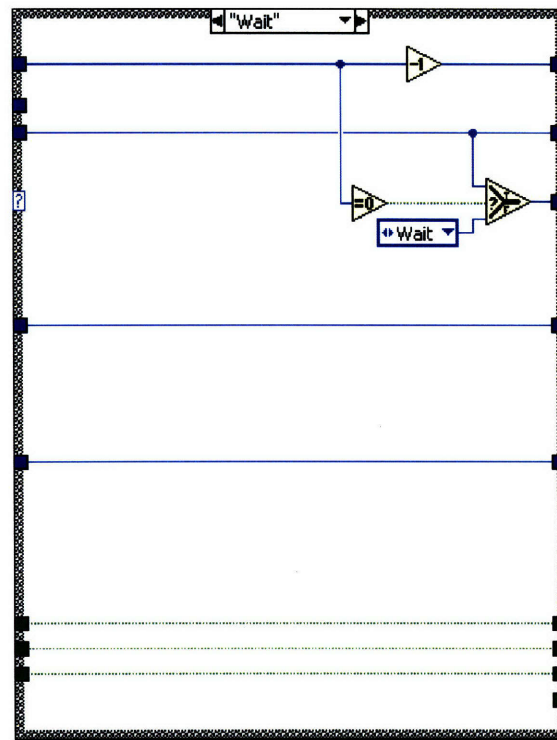
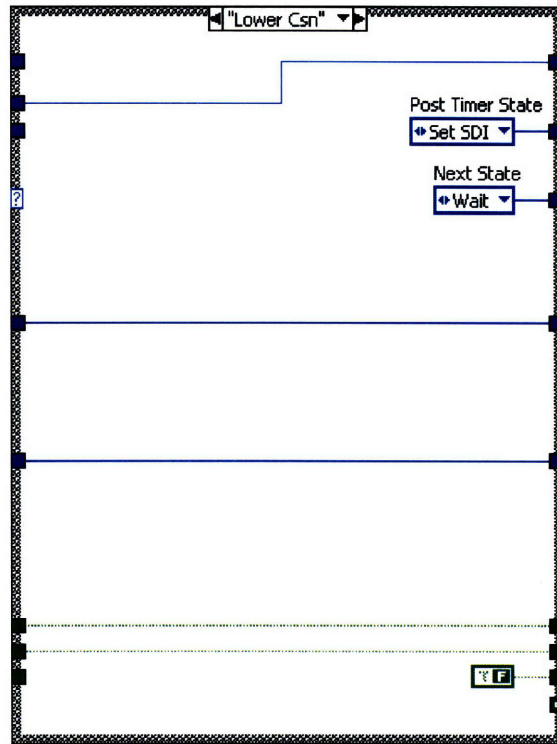




## C.2 D/A Output State Machine







## C.3 Filter/Controller FPGA Code

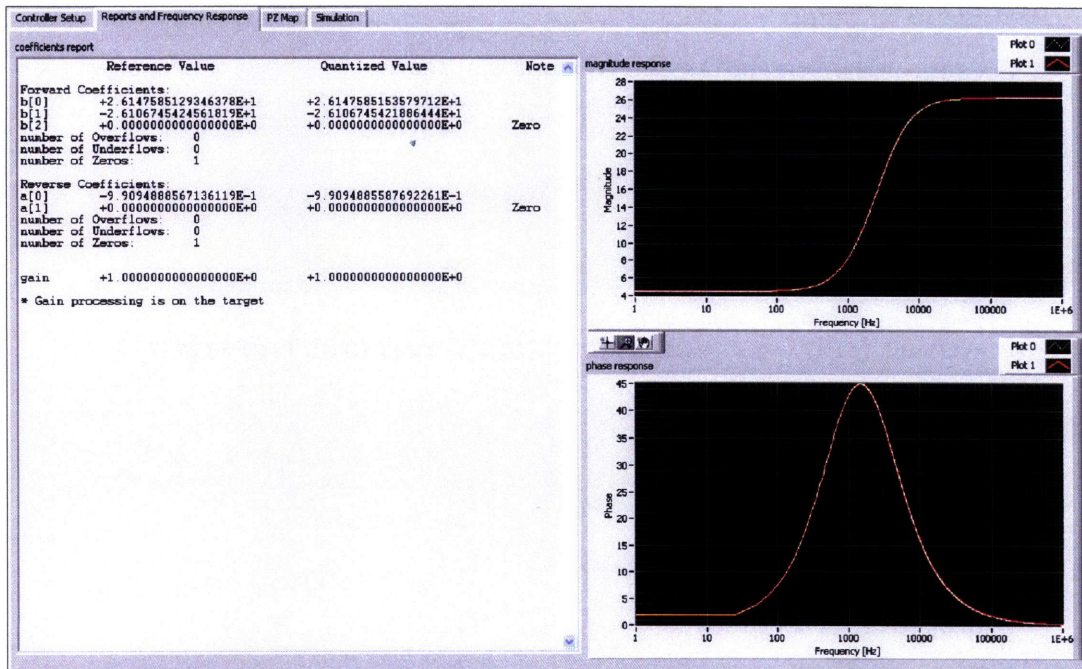
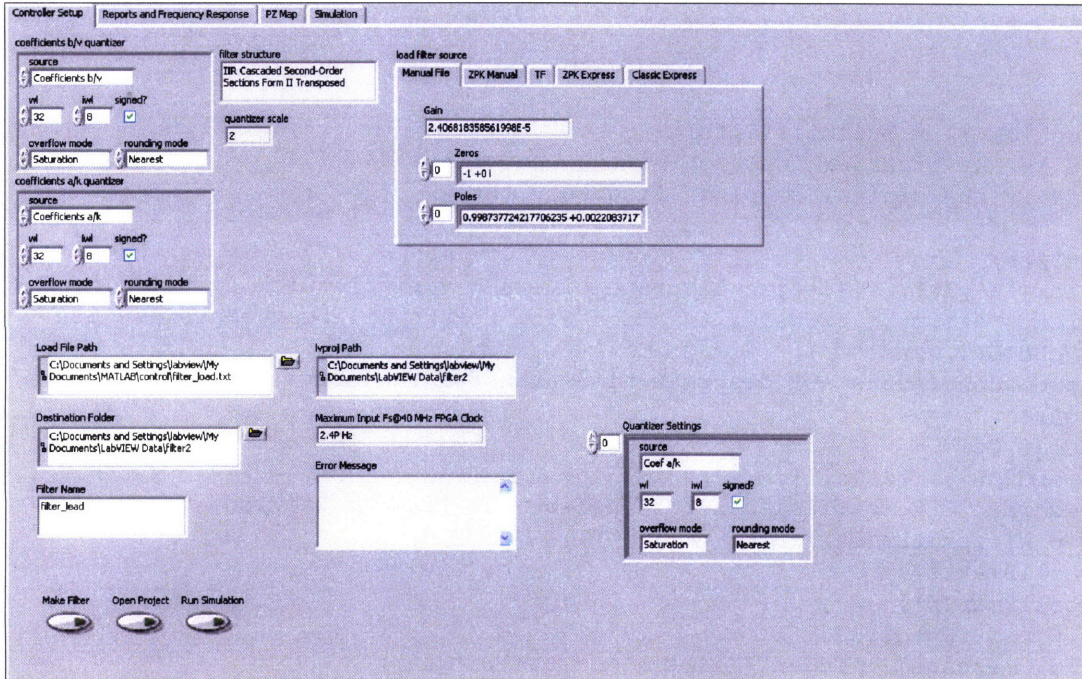
### C.3.1 Matlab Transfer Function Output m-file

```
function [] = buildiir(G,fs)
% G is the transfer function of the controller
% fs is the digital sampling frequency [Hz]

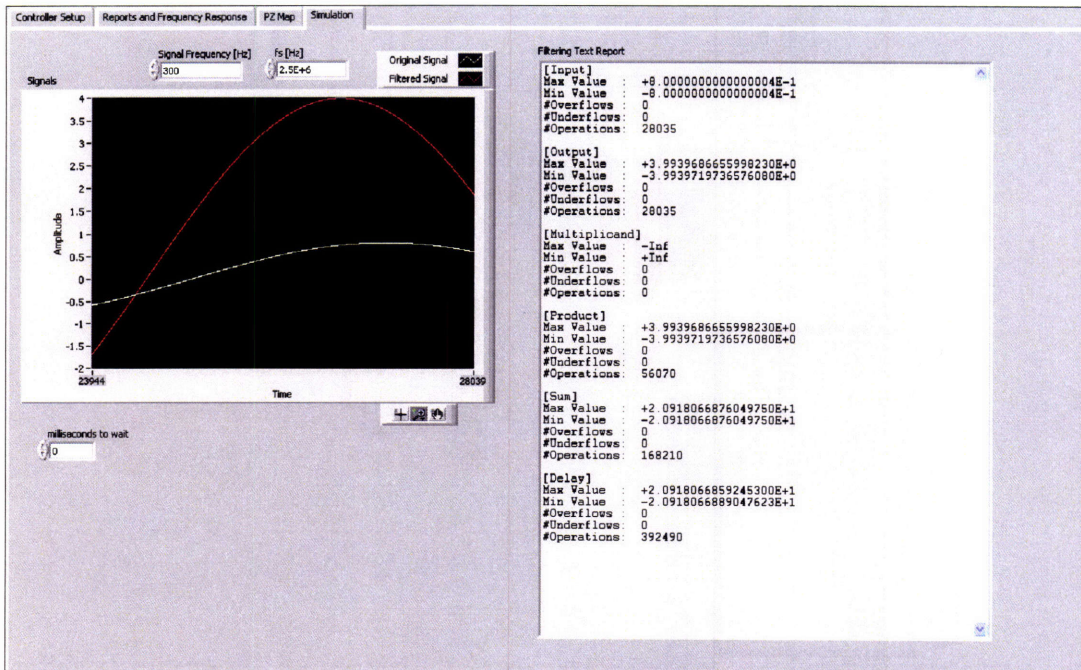
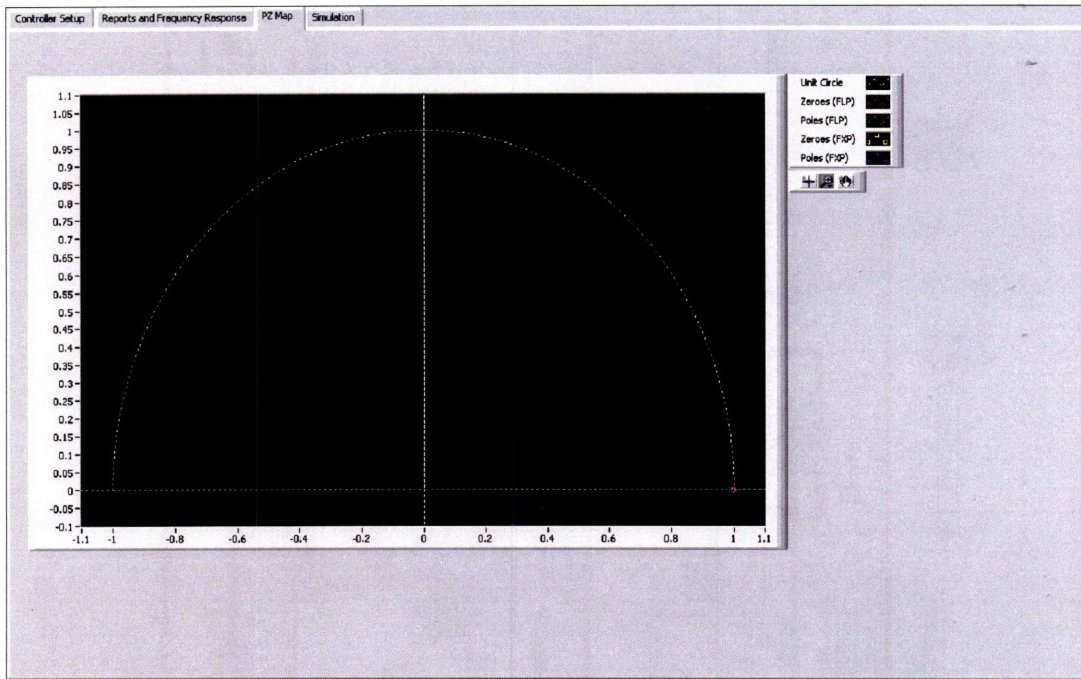
Ts=1/fs;
z=zpk('z',Ts); %define discrete-time ZPK variable with sample time T
%create discrete transfer function with tustin bilinear approximation
Gz=c2d(G,T,'tustin');
% plot continuous and discrete-time transfer functions
figure
bode(G,Gz)
legend('G(s)', 'G(z)');
% extract ZPK coefficients for text output file to LabVIEW
[z p k]=zpkdata(Gz);
z=cell2mat(z);
p=cell2mat(p);
% open output file
[fid, message] = fopen('filter.load.txt', 'wt');
% check to make sure output file opened correctly
if (fid == -1)
    display('error opening text file');
    display(message);
else
    % output number of poles and number fo zeros
    fprintf(fid, '%3d %3d\n', length(p), length(z));
    % output list of zeros with real and imaginary components
    for n=1:length(z)
        fprintf(fid, '%20.30f + %20.30f\n', real(z(n)), imag(z(n)));
    end
    fprintf(fid, '\n');
    % output list of poles with real and imaginary components
    for n=1:length(p)
        fprintf(fid, '%20.30f + %20.30f\n', real(p(n)), imag(p(n)));
    end
    fprintf(fid, '\n');
    % output gain K
    fprintf(fid, '%20.30f\n', k);
    status=fclose(fid); % close output file
    if(status == 0)
        display('File built successfully');
    else
        display('File close unsuccessful');
    end
end
end
```

## C.3.2 LabVIEW Generator Code

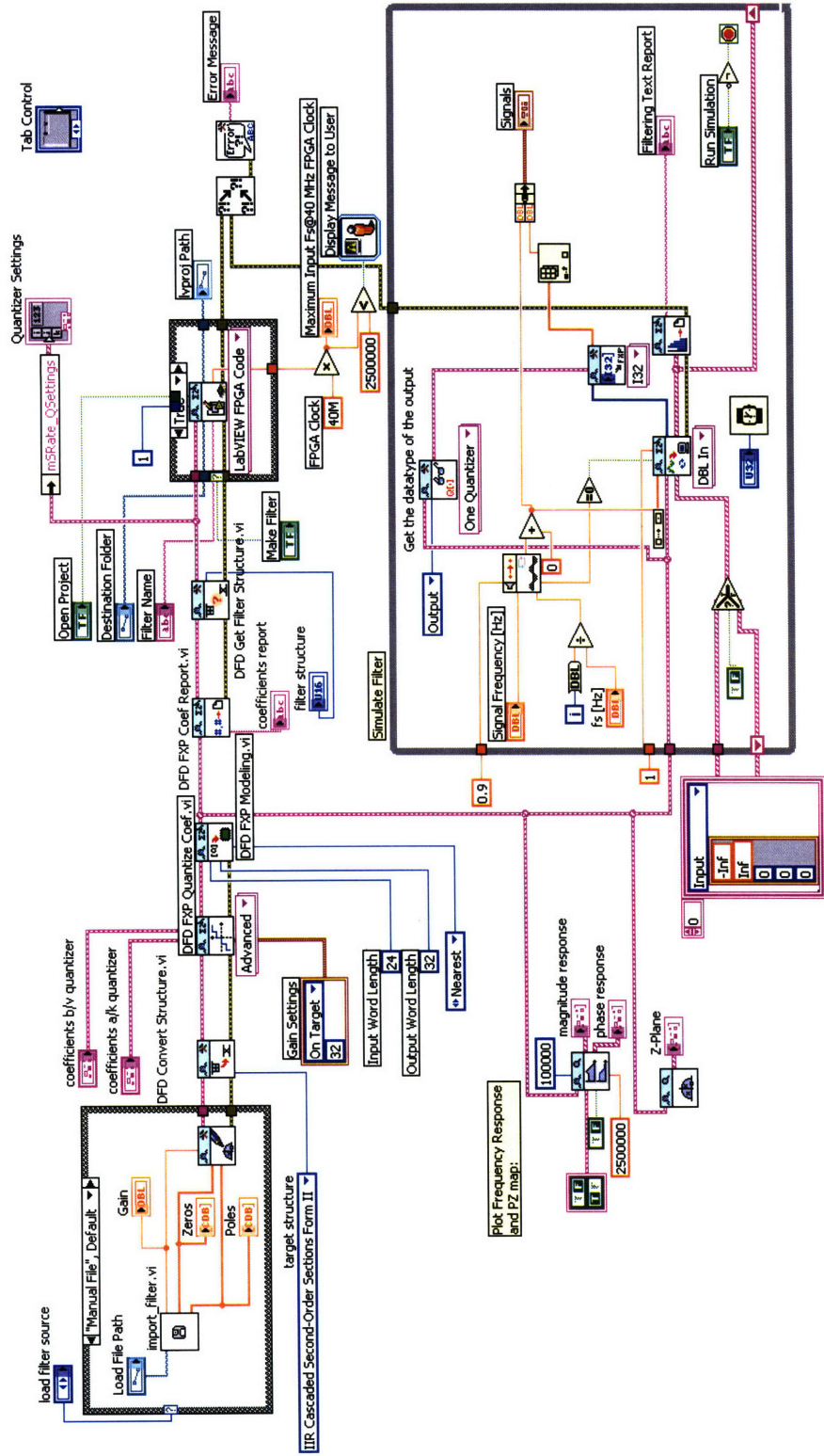
### Front Panel Tabs



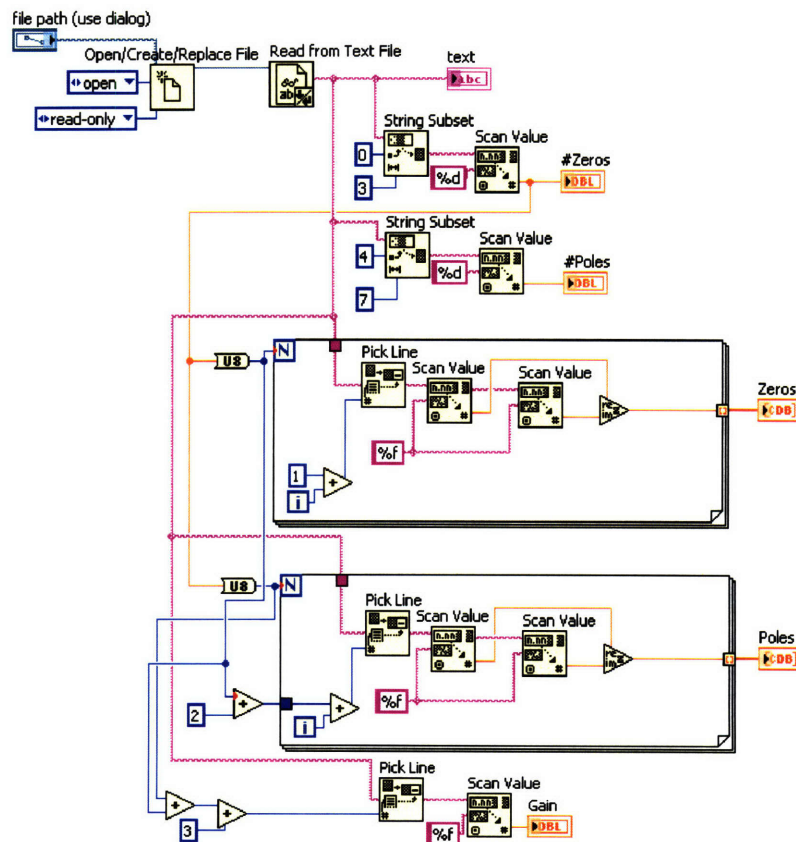
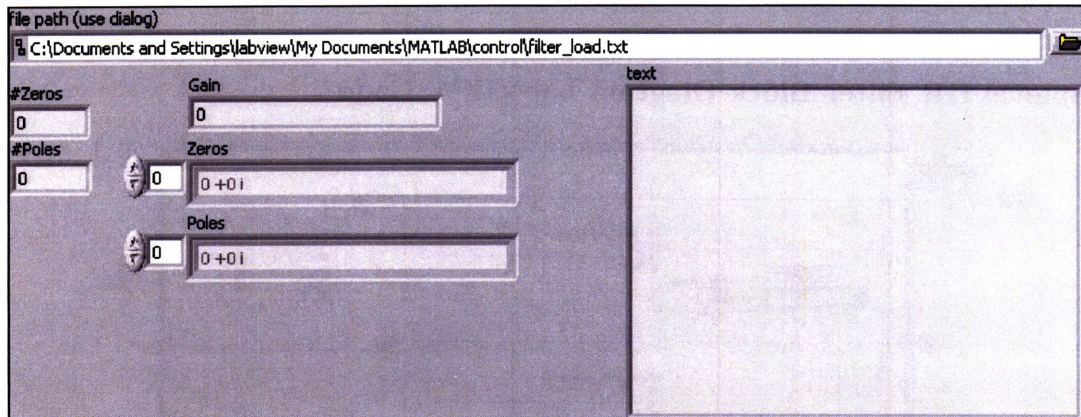




# Source Code VI

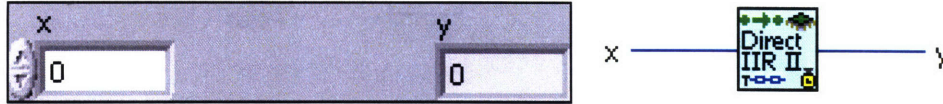


# LabVIEW Transfer Function Input VI

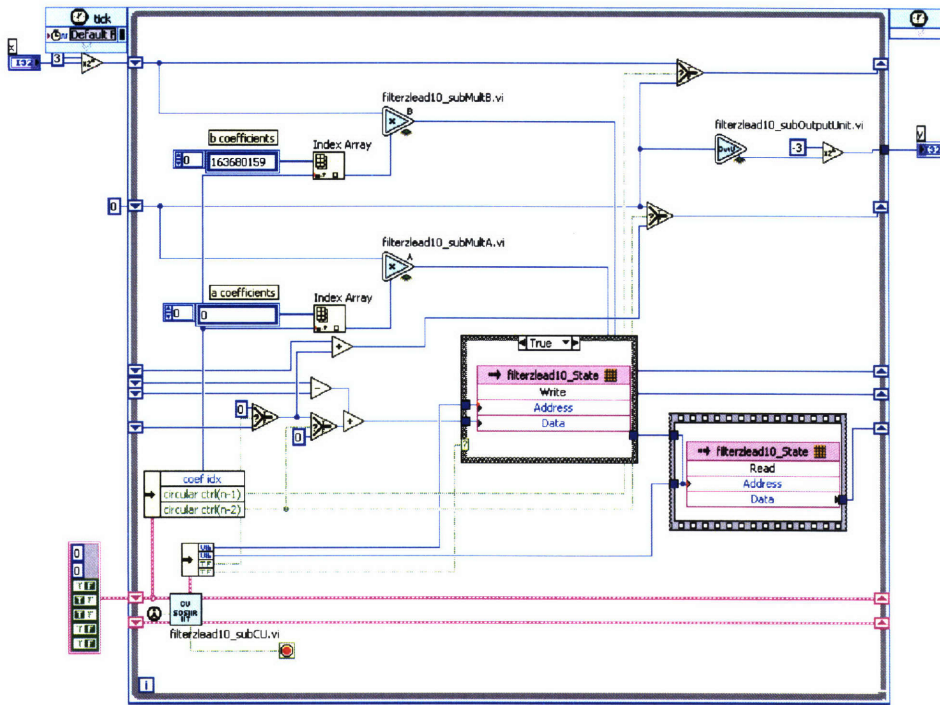


### C.3.3 LabVIEW Generated Filter Code

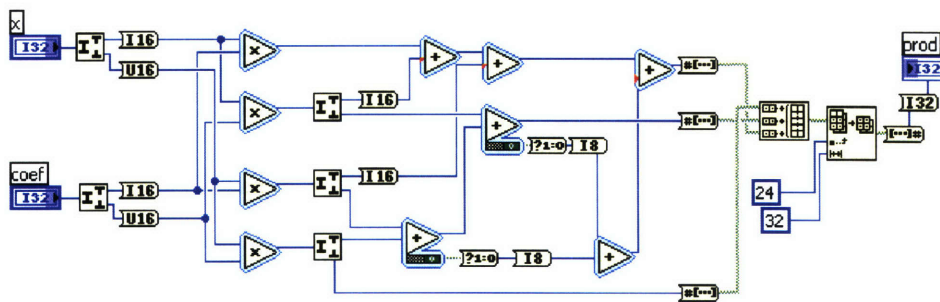
#### IIR Filter Front Panel and Icon



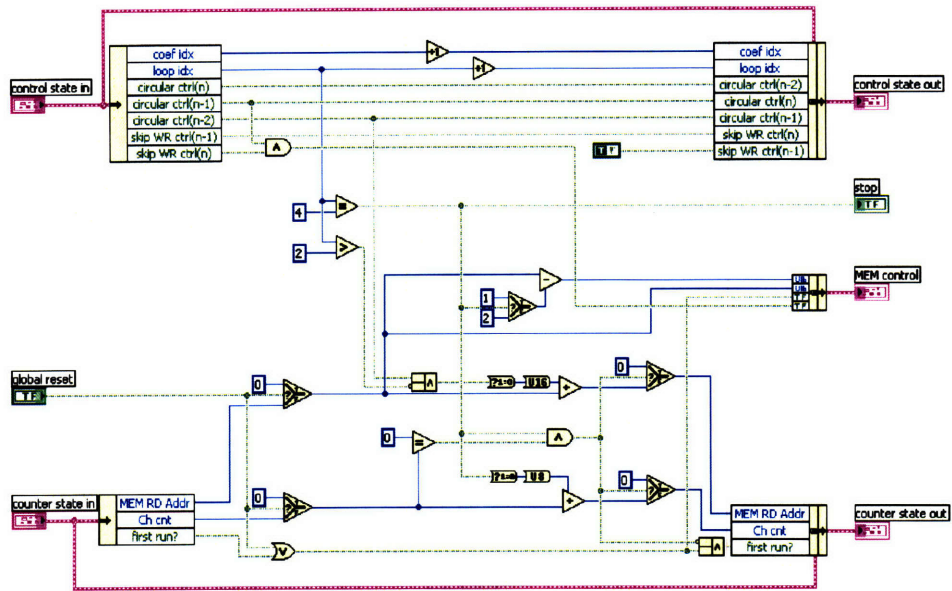
#### Toplevel IIR Filter Block Diagram LabVIEW Code



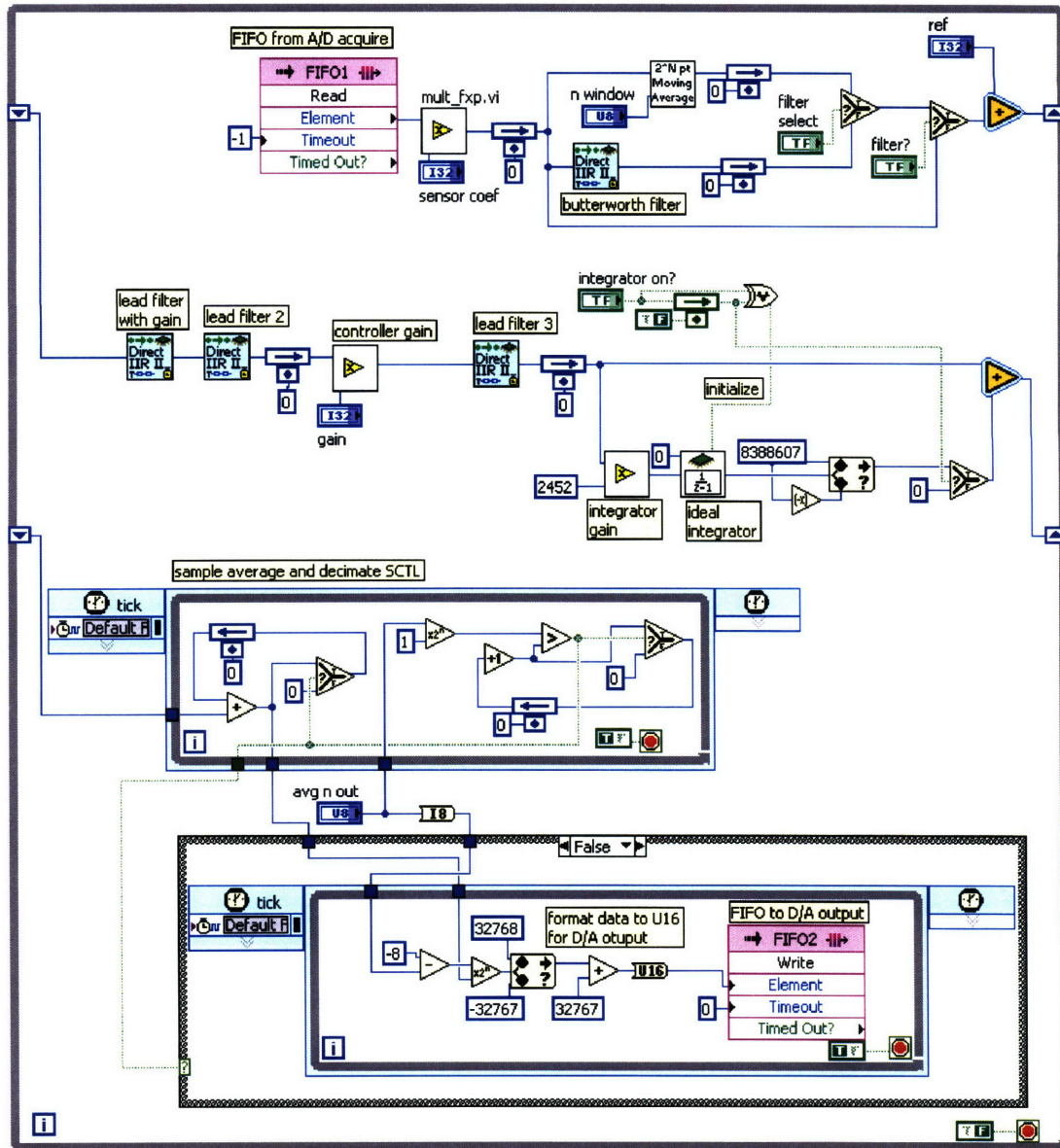
#### IIR Filter Multiplier Block LabVIEW Code



# IIR Filter Control Block LabVIEW Code



### C.3.4 Example of LabVIEW FPGA Control Implementation: Lag, Triple-Lead



# Bibliography

- [1] Walt Kester, editor. *Mixed-Signal and DSP Design Techniques*. Newnes, 2003.
- [2] Analog Devices, Inc., Norwood, MA. *AD7760: 2.5 MSPS, 24-Bit, 100 dB, Sigma-Delta ADC with On-Chip Buffer*, 2006. [http://www.analog.com/UploadedFiles/Data\\_Sheets/AD7760.pdf](http://www.analog.com/UploadedFiles/Data_Sheets/AD7760.pdf).
- [3] dSPACE, Inc., Paderborn, Germany. *DS1103 PPC Controller Board Hardware Reference*, 2001. dSPACE Hardware Manual, available with DS1103 documentation.
- [4] David Otten. *Instruction Manual: High Resolution Analog Input and Output for the dSPACE DS1103 PPC Controller*. Precision Motion Control Laboratory, MIT, Cambridge, MA, 2005. Available at [dSPACE.mit.edu](http://dSPACE.mit.edu).
- [5] Innovative Integration, Simi Valley, CA. *X3-SDF*, 2007. <http://www.innovative-dsp.com/products/xmce-sdf.htm>.
- [6] Xiaodong Lu. *Electromagnetically-driven ultra-fast tool servos for diamond turning*. Ph.D. thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, 2005.
- [7] VMERTO, Inc. Accessed March 1, 2008. <http://www.vmetro.com/category3723.html#>.
- [8] Texas Instruments, Inc., Dallas, TX. *Fully-Differential Amplifiers: Application Report SLOA054D*, 2002. <http://focus.ti.com/lit/an/sloa054d/sloa054d.pdf>.
- [9] Analog Devices, Inc., Norwood, MA. *ADG3308: Low Voltage, 1.15 V to 5.5 V, 8-channel Bidirectional Logic Level Translators*, 2007. [http://www.analog.com/UploadedFiles/Data\\_Sheets/ADG3308\\_3308-1.pdf](http://www.analog.com/UploadedFiles/Data_Sheets/ADG3308_3308-1.pdf).
- [10] Alan V. Oppenheim and Ronald W. Shafer. *Discrete Time Signal Processing*. Prentice Hall, Inc., Second edition, 1998.
- [11] National Instruments. *LabVIEW 8.5 FPGA Module Help*. National Instruments Corporation, Austin, TX. Available at [www.ni.com](http://www.ni.com).
- [12] Ross I. MacKenzie, Aaron J. Gawlik, and David L. Trumper. High-scan rate positioner for scanned probe microscopy. *Submitted to ASPE Annual Meeting, October 2008*.

- [13] Gene F. Franklin, J. David Powell, and Michael L. Workman. *Digital Control of Dynamic Systems*. Addison Wesley Longman, Pearson Education, Third edition, 1998.
- [14] dSPACE, Inc. 50131 Pontiac Trail, Wixom, MI 48393.  
Tel: 248-295-4700. URL: [www.dspaceinc.com](http://www.dspaceinc.com).
- [15] The Mathworks, Inc. 3 Apple Hill Drive, Natick, MA 01760.  
Tel: 508-647-7000. URL: [www.mathworks.com](http://www.mathworks.com).
- [16] Stuart T. Smith, David L. Trumper, David Otten, Robert J. Hocken, Hua Yang, and Richard M. Seugling. Motion control platform for accurate measurement and manufacture of nanostructures. *NSF DMII Annual Conference*, 2005.
- [17] Innovative Integration. 2390-A Ward Avenue, Simi Valley, CA 93065.  
Tel: 805-578-4260. URL: [www.innovative-dsp.com](http://www.innovative-dsp.com).
- [18] National Instruments Corporation. 11500 N Mopac Expwy, Austin, TX 78759.  
Tel: 800-531-5066. URL: [www.ni.com](http://www.ni.com).
- [19] ADE Technology. 80 Wilson Way, Westwood, MA 02090. Tel: 781-467-3500.  
URL: [www.adetech.com](http://www.adetech.com).
- [20] Gene F. Franklin, J. David Powell, and Abbas Emani-Naeini. *Feedback Control of Dynamic Systems*. Prentice Hall, Inc., Fourth edition, 2002.
- [21] Xiao Wang and Boon-Teck Ooi. Real-time multi-dsp control of three phase current-source unity power factor pwm rectifier. *Power Electronics Specialists Conference, 1992. PESC '92 Record., 23rd Annual IEEE*, page 1376, 1992.
- [22] VMETRO, Inc. 1880 S. Dairy Ashford, Suite 400, Houtson, TX 77077.  
Tel: 281-584-0728. URL: [www.vmetro.com](http://www.vmetro.com).
- [23] Bin Le, Thomas W. Rondeau, Jeffrey H. Reed, and Charles W. Bostian. Analog-to-digital converters. *IEEE Signal Processing Magazine*, 22(6):69, November 2005.
- [24] R.H. Walden. Analog-to-digital converter technology comparison. In *Gallium Arsenide Integrated Circuit (GaAs IC) Symposium*, page 217, March 1994.
- [25] R.H. Walden. Analog-to-digital converter converter survey and analysis. *IEEE Journal on Selected Areas in Communications*, 17(4):539, April 1999.
- [26] Jeffrey H. Reed. *Software Radio: a modern approach to radio engineering*. Prentice Hall, Inc., 2002.
- [27] David F Hoeschele. *Analog-to-Digital and Digital-to-Analog Conversion Techniques*. John Wiley & Sons, 1994.



- [28] Robert M. Gray. Quantization noise spectra. *IEEE Transactions on Information Theory*, 36(6):1220, 1990.
- [29] R.W. Stewart and E. Pfann. Oversampling and sigma-delta strategies for data conversion. *Electronics & Communication Engineering Journal*, 10(1):37, 1998.
- [30] Yasuyuki Matsuya, Kuniharu Uchimura, Atsushi Iwata, Tsutomu Kobayashi, Masayuki Ishikawa, and Takeshi Yoshitome. A 16-bit oversampling a-to-d conversion technology using triple-integration noise shaping. *IEEE Journal of Solid-state Circuits*, 22(6):921, 1987.
- [31] Max W. Hauser. Principles of oversampling A/D conversion. *Journal Audio Engineering Society*, 39(1):3, January 1991.
- [32] Walt Jung. *Op Amp Applications*. Analog Devices, Inc., 2002.
- [33] Bernhard E. Boser and Bruce A. Wooley. The design of sigma-delta modulation analog-to-digital converters. *IEEE Journal of Solid-State Circuits*, 23(6):1298, December 1988.
- [34] Rudy van de Plassche. *CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters*. Kluwer Academic Publishers, Second edition, 2003.
- [35] Paul Horowitz and Winfield Hill. *The Art of Electronics*. Cambridge University Press, Second edition, 1989.
- [36] Alan V. Oppenheim, Alan S. Willsky, and S. Hamid Nawab. *Signals & Systems*. Prentice Hall, Inc., Second edition, 1997.
- [37] Katherine Lilienkamp. A simulink-driven dynamic signal analyzer. Undergraduate thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, 1999.
- [38] Analog Devices, Inc., Norwood, MA. *EVAL-AD7760: Evaluation Board for AD7760 using Blackfin ADSP-BF537 EZ-KIT Lite*, 2006. [http://www.analog.com/UploadedFiles/Evaluation\\_Boards\\_Tools/252014382EVAL\\_AD7760\\_7762\\_7763EB.pdf](http://www.analog.com/UploadedFiles/Evaluation_Boards_Tools/252014382EVAL_AD7760_7762_7763EB.pdf).
- [39] Michael Coln. Analog Devices, Inc., 2007. personal communication with product application engineer.
- [40] Walt Kester, editor. *The Data Conversion Handbook*. Newnes, 2005.
- [41] Microchip Technology, Inc. 2355 West Chandler Blvd., Chandler, AZ. Tel: 480-792-7200. URL: [www.microchip.com](http://www.microchip.com).
- [42] Advanced Circuits. 21101 E. 32nd Parkway, Aurora, CO 80011. Tel: 800-979-4722. URL: [www.4pcb.com](http://www.4pcb.com).
- [43] Carla Uribe. National Instruments Corporation, 2008. personal communication with product application engineer.