

2011

# Design and Testing of a Prototype High Speed Data Acquisition System for Nasa

Vishwas Tumkur Vijayendra  
*University of Massachusetts Amherst*

Follow this and additional works at: <https://scholarworks.umass.edu/theses>



Part of the [Electrical and Computer Engineering Commons](#)

---

Vijayendra, Vishwas Tumkur, "Design and Testing of a Prototype High Speed Data Acquisition System for Nasa" (2011). *Masters Theses 1911 - February 2014*. 732.

Retrieved from <https://scholarworks.umass.edu/theses/732>

This thesis is brought to you for free and open access by ScholarWorks@UMass Amherst. It has been accepted for inclusion in Masters Theses 1911 - February 2014 by an authorized administrator of ScholarWorks@UMass Amherst. For more information, please contact [scholarworks@library.umass.edu](mailto:scholarworks@library.umass.edu).

**DESIGN AND TESTING OF A PROTOTYPE HIGH SPEED DATA  
ACQUISITION SYSTEM FOR NASA**

A Thesis Presented

by

VISHWAS T. VIJAYENDRA

Submitted to the Graduate School of the  
University of Massachusetts Amherst in partial fulfillment  
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL AND COMPUTER ENGINEERING

September 2011

ELECTRICAL AND COMPUTER ENGINEERING

© Copyright by Vishwas T. Vijayendra 2011

All Rights Reserved

**DESIGN AND TESTING OF A PROTOTYPE HIGH SPEED DATA  
ACQUISITION SYSTEM FOR NASA**

A Thesis Presented

by

VISHWAS T. VIJAYENDRA

Approved as to style and content by:

---

Russell G. Tessier, Chair

---

Paul Siqueira, Member

---

Stephen Frasier, Member

---

C. V. Hollot, Department Head  
Electrical and Computer Engineering

## **ACKNOWLEDGMENTS**

I would first like to thank my parents and sister for their constant support and guidance throughout my life. This thesis is dedicated to my mother Bharathi who has been the single most influential person in my life. I would like to thank National Aeronautics and Space Administration (NASA) for funding this project and giving me an opportunity to work on it. I am very grateful to my advisor Professor Russell Tessier for his motivation, encouragement and guidance throughout this project. I would also like to acknowledge Professor Paul Siqueira for his invaluable inputs and suggestions. I would also like to thank my lab mates and seniors who helped me during the course of my thesis. Last but not the least, a special thank you to all my close friends at Amherst for being a part of my journey and making it memorable and unforgettable.

## **ABSTRACT**

### **DESIGN AND TESTING OF A PROTOTYPE HIGH SPEED DATA ACQUISITION SYSTEM FOR NASA**

SEPTEMBER 2011

VISHWAS T. VIJAYENDRA

B.E., VISVESVARAYA TECHNOLOGICAL UNIVERSITY, INDIA

M.S. E.C.E., UNIVERSITY OF MASSACHUSETTS, AMHERST

Directed by: Professor Russell G. Tessier

Modern radar and signal processing applications require data acquisition systems capable of high-speed analog data reception and processing. These systems need to support sophisticated signal processing algorithms and reliable high-speed interfaces. The objective of this project is to develop a prototype of a state of the art data acquisition system to aid NASA's Surface Water and Ocean Topography (SWOT) mission. The SWOT mission aims at monitoring water levels of various water bodies to predict and avoid any catastrophic events. The principal instrument is a Ka-band Radar Interferometer (KaRIN) that is used for the measurement of water levels. The collected data need to be digitized and processed using an FPGA based-data acquisition system housed in a satellite. The scope of this project involves the design, implementation and test of a high-speed printed circuit board (PCB) that serves as the prototype data acquisition system. A lot of emphasis is placed on layout design, as the PCB needs to support data rates up to three Giga samples per second. The goal of this research is to provide Jet Propulsion Laboratory (JPL), NASA with a prototype version of the high-speed acquisition system that can be integrated with the KaRIN system in future.

# TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS .....	iv
ABSTRACT .....	v
LIST OF TABLES .....	ix
LIST OF FIGURES .....	x
1. INTRODUCTION .....	1
2. SURFACE WATER AND OCEAN TOPOGRAPHY MISSION OF NASA .....	4
2.1 Mission Architecture.....	4
2.2 Background Work.....	7
3. DATA ACQUISITION SYSTEM SPECIFICATIONS .....	9
3.1 Radar Terminologies.....	9
3.2 DAS Specifications .....	9
3.3 DAS Features .....	10
4. SYSTEM ARCHITECTURE AND DESIGN DETAILS .....	12
4.1 Power Distribution .....	14
4.2 Clock Distribution.....	16
4.3 Analog to Digital Converter.....	17
4.3.1 Dual Edge Sampling .....	17
4.3.2 DDR Output Clock and 1:2 Output De-multiplexing .....	17
4.3.3 Alternate Output Clock .....	18
4.3.4 Extended Control Mode.....	18
4.3.5 Serial Interface .....	19
4.4 Data FPGA.....	20
4.4.1 ADC – Data FPGA Interface .....	21
4.4.2 Logic Circuits.....	23
4.4.3 Serial Controller.....	25
4.4.4 MicroBlaze.....	25
4.4.5 RocketIO MGT .....	25
4.5 PCI FPGA .....	25
4.5.1 Compact PCI Interface.....	26
4.6 SATA and SFP Interfaces.....	27

4.7 Thermal Sensor .....	27
4.8 FPGA Programming Interfaces.....	27
4.8.1 Slave SelectMAP8 .....	27
4.8.2 JTAG Interface.....	29
5. LAYOUT CONSIDERATIONS .....	30
5.1 Data FPGA Layout Guidelines .....	30
5.2 ADC Layout Guidelines .....	32
5.3 Compact PCI Layout Guidelines .....	34
6. PRELIMINARY VERIFICATION .....	36
6.1 Analog Simulation .....	36
6.2 Power budgeting of the board .....	39
6.3 Data FPGA Compilations .....	40
7. TESTING THE PROTOTYPE HARDWARE SYSTEM .....	41
7.1 Precautions and Safety .....	41
7.2 Powering the board .....	41
7.3 FPGA Test Designs without the Clock.....	42
7.4 Flash Programming.....	43
7.5 PCI FPGA Test Plan .....	43
7.6 FPGA Test Designs with the Clock .....	44
7.7 Establishing the Serial Communication.....	44
7.8 ADC Test Designs .....	45
7.8.1 IADC Test Mode Configuration .....	45
7.8.2 IADC Test Mode Configuration at higher input clock frequencies .....	45
7.8.3 IADC Test Mode Configuration beyond 1.3GHz.....	47
7.8.4 QADC Test Mode Configuration.....	49
7.9 Sampling the Input Signal.....	49
7.9.1 Time Skew and Amplitude offset Corrections.....	51
7.9.1.1 113MHz Input Cosine Wave .....	51
7.9.1.2 133MHz Input Cosine Wave .....	52
7.10 Error Spectra .....	55
7.11 Filter implementation, integration and testing.....	57
8. REAL TIME ESTIMATES OF DIFFERENTIAL SIGNAL PHASE.....	60
8.1 Introduction.....	60
8.2 Background Work.....	61
8.3 FPGA Implementation and Testing .....	62
8.4 Experimental approach .....	66
8.5 Results.....	67



9. CONCLUSIONS AND FUTURE WORK.....	73
9.1 Conclusions.....	73
9.2. Future Work.....	75
APPENDIX: SCHEMATICS .....	76
BIBLIOGRAPHY.....	104

## LIST OF TABLES

Table	Page
Table 1. On-board Power Regulators.....	14
Table 2. ADC Registers .....	19
Table 3. I/O pins on Data FPGA interfacing ADC data and clock pins .....	23
Table 4. PCB Specifications .....	37
Table 5. Analog Circuit - Trace Widths.....	37
Table 6. Analog Circuit - Trace lengths.....	37
Table 7. Power Consumption Estimate.....	39
Table 8. QADC -Time Skew & Amplitude Correction for 113 MHz input.....	51
Table 9. IADC -Time Skew & Amplitude Correction for 113 MHz input.....	51
Table 10. QADC -Time Skew & Amplitude Correction for 133 MHz input (Agilent waveform generator).....	53
Table 11. IADC -Time Skew & Amplitude Correction for 133 MHz input (Agilent waveform generator).....	53
Table 12. QADC -Time Skew & Amplitude Correction for 133 MHz input (Tektronix waveform generator).....	53
Table 13. IADC -Time Skew & Amplitude Correction for 133 MHz input (Tektronix waveform generator).....	53
Table 14. Corrected initial phase estimation ( <i>phic</i> ) for IADC versus actual phase .....	70
Table 15. Corrected initial phase estimation for QADC versus actual phase.....	71
Table 16. True (Matlab) versus FPGA measured phase differences .....	71

## LIST OF FIGURES

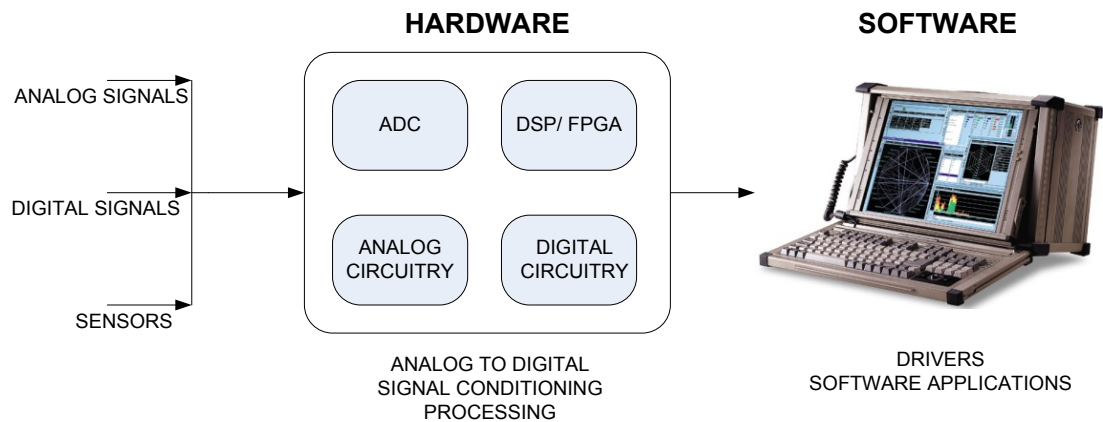
Figure	Page
Figure 1. Typical Data Acquisition System.....	1
Figure 2. KaRIN System.....	5
Figure 3. High Level Block Diagram of the SWOT Mission.....	5
Figure 4. Data Acquisition System.....	7
Figure 5. System Architecture .....	13
Figure 6. High frequency filtering circuit for MGTs.....	15
Figure 7. ADC Enable Circuit .....	16
Figure 8. Serial Interface Timing.....	20
Figure 9. ADC – Data FPGA interface.....	21
Figure 10. Clock regions in XC4VFX140 device.....	22
Figure 11. ISERDES and FIFO Implementation .....	24
Figure 12. Slave SelectMAP8 mode – Xilinx PROM and PCI FPGA.....	28
Figure 13. Slave SelectMAP8 mode – PCI FPGA and Data FPGA.....	28
Figure 14. JTAG Scan Chain.....	29
Figure 15. Recommended thermal land pattern for the ADC.....	33
Figure 16. Analog Circuit.....	36
Figure 17. S parameter plot for the analog section.....	38
Figure 18. 3U Extender card from Twin Industries.....	42
Figure 19. Circuit implementation for higher input clock frequencies.....	46
Figure 20. Data path and Clock path in Data FPGA.....	48
Figure 21. Data Plot - QADC .....	50

Figure 22. Data Plot - IADC .....	50
Figure 23. Data plot of QADC after correction .....	51
Figure 24. Data plot of IADC after correction.....	52
Figure 25. Error spectra - Agilent generator E4437B.....	55
Figure 26. Error spectra - Tektronix Waveform Generator AFG3252 .....	56
Figure 27. Spectrum plot of the ADC output of the up-converted input .....	58
Figure 28. Spectrum plot of the polyphase decimation filter.....	59
Figure 29. Artistic concept of the SWOT satellite mission. ....	61
Figure 30. Measured and theoretical phase accuracy versus the number of samples.....	62
Figure 31. Block diagram of the hardware for phase detection in Data FPGA .....	63
Figure 32. Block diagram of the experimental setup for the phase detection circuit .....	67
Figure 33. Hardware PCB board.....	74

# CHAPTER 1

## INTRODUCTION

Data Acquisition Systems (DAS) are ubiquitous in microwave, signal processing and radar applications [1] [2] [3]. The objective of data acquisition is to measure an electrical or physical quantity of interest such as voltage, current, pressure, temperature etc. Although each DAS is customized depending on the specific application, they share the common functionalities of acquiring, processing and analyzing the information. Most of the present-day systems use computer based data acquisition. A combination of hardware modules, software applications and a computer is used to measure and analyze the quantity of interest. A typical data acquisition system is shown in Figure 1.



**Figure 1. Typical Data Acquisition System**

A DAS used in satellite radar applications is distinctive from a typical acquisition system in many ways. Firstly, the components present on the DAS must be space-qualified radiation hardened to overcome soft errors. Secondly, these systems need to support real time data acquisition with hardware support for complex signal processing algorithms. Lastly, they need to incorporate several high-speed reliable interfaces [1] to

communicate with the on-board computer and other modules. With recent advancements in technology, many powerful devices and high-speed interfaces are available off the shelf. These enhancements facilitate the development of an advanced DAS.

The common devices used in a DAS are an analog-to-digital converter (ADC) for digitizing the analog signals and Field Programmable Gate Array (FPGA) or Digital Signal Processor (DSP) to process the digitized signals [4]. The prototype systems often use FPGAs since they offer system flexibility and reconfiguration. FPGAs incorporating high-speed interfaces, soft microprocessor, and custom hardware modules for digital filtering are now available in a single device [5] [6].

In this project, we have developed a prototype version of the DAS to meet the requirements of a radar application that monitors the water levels of various water bodies. This is part of the Surface Water and Ocean topography (SWOT) mission carried out by NASA [7]. The data acquisition system is implemented using off the shelf components such as analog-to-digital converters from National Semiconductor and Xilinx Virtex-4 FPGAs.

The initial prototype system is designed by including non-radiation hardened (Rad-Hard) components on the board. The final version of the PCB will include the Rad-Hard equivalent devices following the success of the prototype system.

The thesis report is organized as follows. Chapter 2 provides an overview of the SWOT project of NASA and its goals. The specifications of the data acquisition system, its primary objectives and the top-level design architecture are described in chapter 3. Chapter 4 discusses the system architecture and design details of the board architecture including specific circuit details. The set of layout guidelines for the successful operation

of the system is described in chapter 5. Chapter 6 outlines the preliminary verification procedures followed before the fabrication of the board. Chapter 7 discusses the details pertaining to the test and debug of the system including the experimental results. Chapter 8 describes a novel method of monitoring the phase of the signal and phase drifts between two associated signals using the Data FPGA resources. Finally, chapter 9 concludes the thesis with an overview of the future work.

## CHAPTER 2

### SURFACE WATER AND OCEAN TOPOGRAPHY MISSION OF NASA

The National Research Council decadal review “Earth Science and Applications from Space: National Imperatives for the Next Decade and Beyond” recommended the SWOT mission for implementation by NASA [7]. The SWOT mission combines goals of physical oceanography and hydrology communities. The mission targets ocean mesoscale and sub-mesoscale circulation characterizations at spatial resolution of 10 kilometers and greater. The goals include estimation of global storage change in fresh water bodies and global change in river discharge at sub-monthly, seasonal, and annual time scales.

#### **2.1 Mission Architecture**

The principal instrument of the SWOT mission is a Ka-band Radar Interferometer (KaRIN) [7] [8]. The system employs two swaths each of 50 km and produces heights and co-registered all-weather imagery. This system uses a conventional Jason-class altimeter for nadir coverage and radiometer for wet troposphere delay. The block diagram in Figure 2 outlines the system that will be employed.



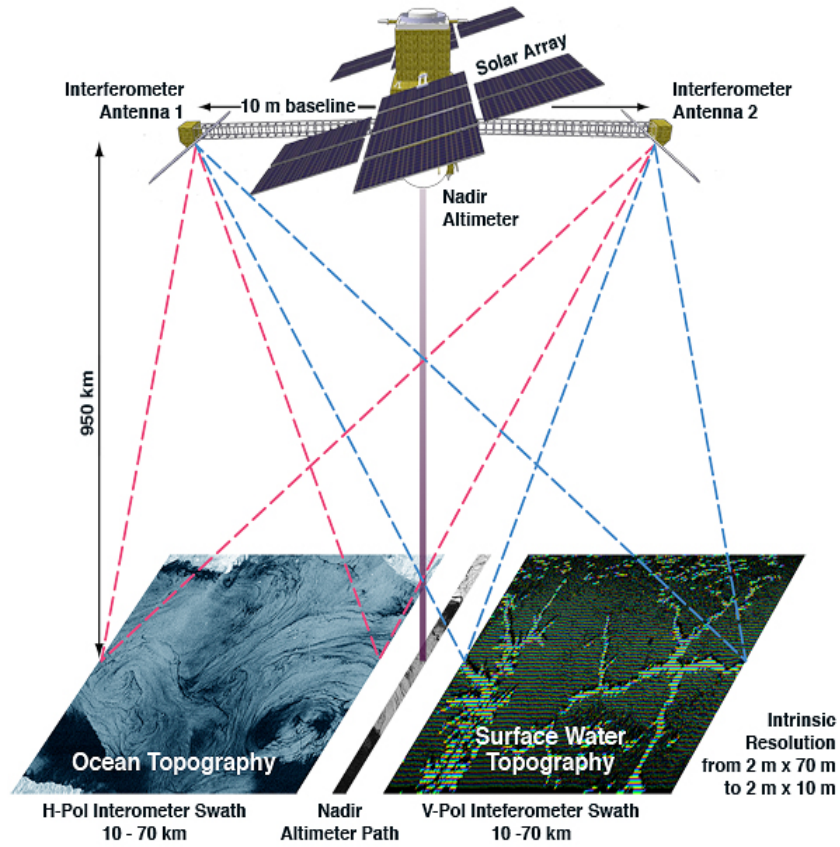


Figure 2. KaRIN System

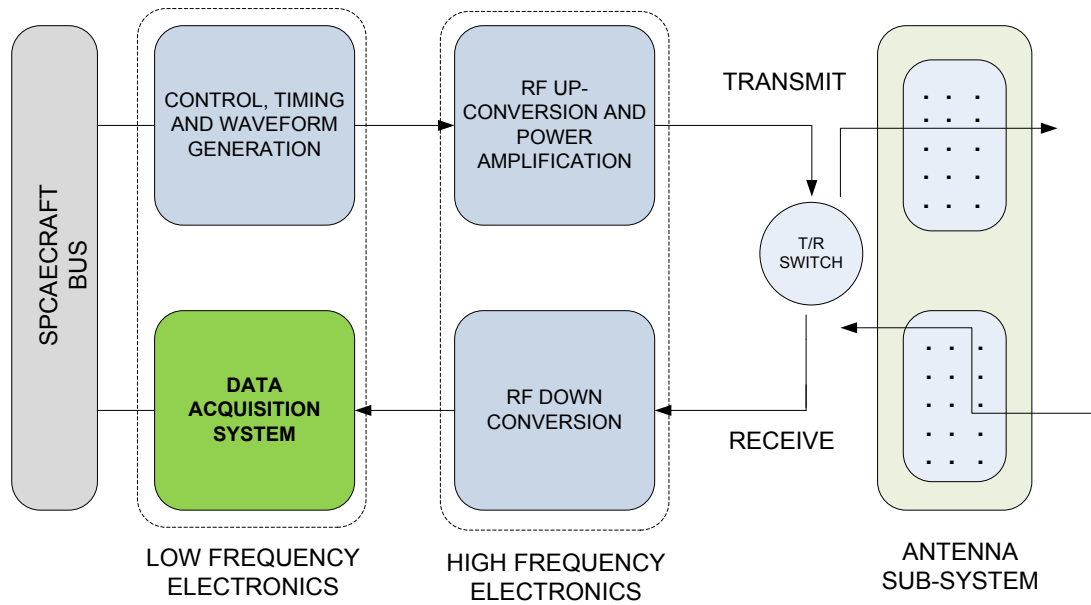
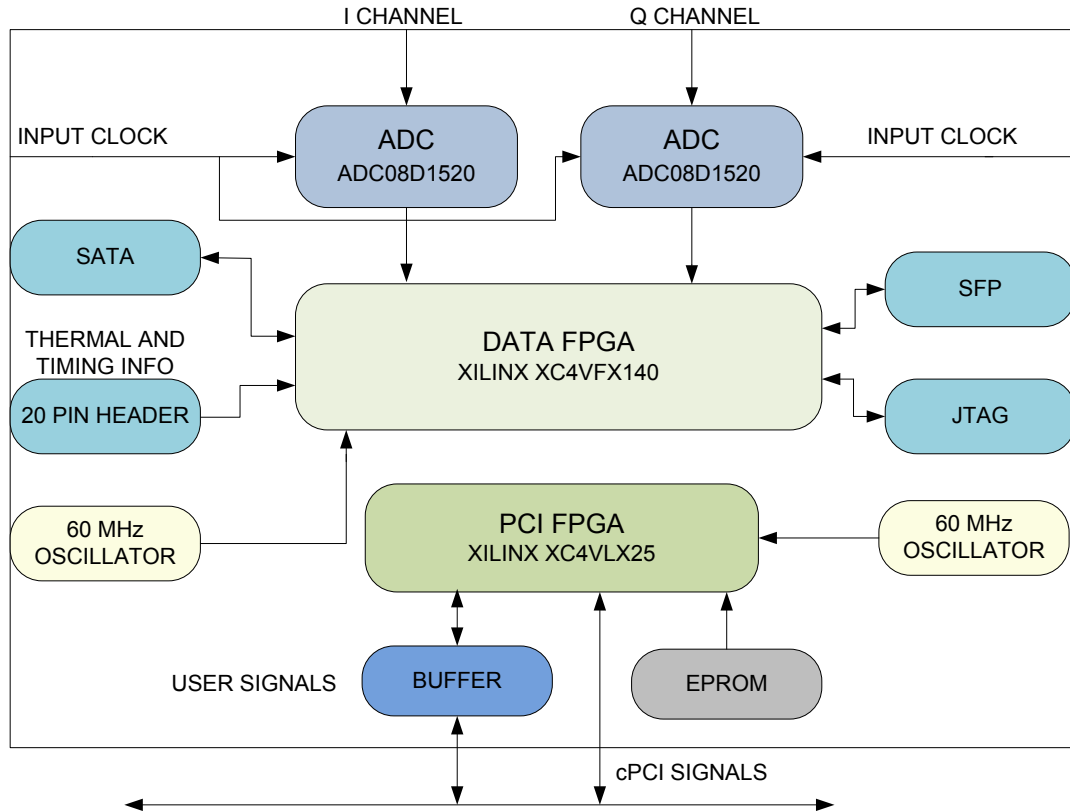


Figure 3. High Level Block Diagram of the SWOT Mission

The high-level block diagram of the system required for the SWOT mission is shown in Figure 3. On the transmitter side, the control, timing and waveform information received from the on board computer are translated to Ka-band signals by the RF up conversion circuitry and transmitted through the antenna sub-system. On the receiver side, the high frequency Ka-band signals are converted to L-band or baseband signals by the down conversion circuitry and fed to a high speed DAS. The DAS processes the analog signals and transfers the processed data to the on-board computer via the spacecraft bus for further analysis.

The proposed architecture includes two 3 Giga Samples Per Second (GSPS) analog-to-digital converters from National Semiconductor [9], a Xilinx Virtex-4 FPGA (XC4VFX140 - Data FPGA) [5] to process data from the ADCs and another Xilinx Virtex-4 FPGA (XC4VLX25 - PCI FPGA) [5] to interface with the cPCI bus. Several high speed interfaces are included in the system such as RocketIO Multi Gigabit Transceivers (MGT) [10], a small form factor pluggable (SFP) connector [11] and a serial advanced technology attachment (SATA) interface [12] to transfer data to other modules at high-speed. The system is equipped with a Joint Test Action Group (JTAG) port to program the FPGA devices and a 20-pin header to monitor the temperature and other timing information. The high-level block diagram of the proposed architecture is portrayed in Figure 4.



**Figure 4. Data Acquisition System**

## 2.2 Background Work

Many commercial solutions like [13] and [14] support high-speed data acquisition with high-speed interfaces. The Signatec PX1500-4 [13] is similar to the DAS system proposed for the SWOT mission. It is equipped with two 3GSPS ADCs [9] and two Xilinx Virtex-5 (XC5VLX50T) FPGAs [6]. There is support for 1400MB/sec continuous transfer over PCI Express bus. Virtex-5 FPGA supports complex signal processing algorithms. However, the system does not support high-speed interfaces like SATA and SFP.

Big3Gig reference design [14] from National Semiconductor employs a 3GSPS ADC (ADC083000/ADC08B3000) [15] [16] and a Virtex-4 FPGA (XC4VLX15) [5].

The reference design supports a single USB interface and lacks multiple high-speed interfaces on the board. Several key design implementation points were obtained from this design to implement the proposed DAS system.

## CHAPTER 3

### DATA ACQUISITION SYSTEM SPECIFICATIONS

The Ka-band radar interferometer in combination with the DAS constitutes the complete system employed in the SWOT project. This chapter discusses the requirements of the data acquisition system and the proposed feature set to meet the requirements.

#### 3.1 Radar Terminologies

This section lists the radar terminologies used in this document.

- (a) High Rate Sampling Mode (HRSM) - HRSM is the mode of the DAS when a single stage down conversion is used in the radar system (Ka-band to L-band).
- (b) Low Rate Sampling Mode (LRSM) – LRSM is the mode of the DAS when two stages of down conversion are used in the radar system (Ka-Band to L-band/S-band and L-band/S-band to baseband).
- (c) Receive Window (Rx Win) – Rx Win is the time window for which the Transmit/Receive Switch (T/R Switch) acts as the receive module.
- (d) Pulse Repetition Frequency (PRF) - PRF is the frequency at which radar pulses are transmitted.

#### 3.2 DAS Specifications

This section lists the specifications set by the radar unit for the DAS.

1. Support real time data acquisition, processing and storage.
2. Support reception of raw analog data on I and Q channels, each channel supporting a bandwidth of 200 MHz centered at 1.2GHz.
3. Support hardware for time-domain convolution filtering and decimation.

4. Support a processor that includes block RAM of up to 828Kbit in HRSM and 252Kbit in LRSM.
5. Support processing of up to 90K samples (HRSM) and 18K samples (LRSM) between PRF events.
6. Support up to 80 18 x 18 multipliers in HRSM and 42 18 x 18 multipliers in LRSM.
7. Support high speed interfaces capable of transferring data at 1GSPS on two channels.
8. Support a Compact PCI (cPCI) interface to enable communication with the on-board computer. This implies that the PCB should have a standard 6U cPCI form factor (160mm X 233mm) [17].

### **3.3 DAS Features**

The following section outlines the various features of the data acquisition system that meets the specifications set by the radar unit.

1. Two analog-to-digital converters corresponding to the two input data channels, I and Q.
2. A Virtex-4 FPGA device (XC4VFX140) implements complex time domain convolution filters.
3. Another Virtex-4 FPGA device (XC4VLX25) implements a 64-bit 66MHz cPCI core and supports a finite state machine (FSM) to load the configuration of XC4VFX140 device via cPCI bus.
4. High speed interfaces such as SATA, SFP and Rocket IO MGTs to transfer processed data to other modules.

5. JTAG port and Universal Asynchronous Receiver Transmitter (UART) channel to support debugging of the system.
6. Other components like power regulators, on-board oscillators, and EPROM to support the development of the data acquisition system.

The ADCs sample the input L-band/ baseband signals at 3GSPS. The Virtex-4 Data FPGA supports up to 192 18 x 18 multipliers and 9936 Kbit of memory [5] to perform time domain filtering operations. The Data FPGA supports processing of more than 90K samples in HRSM on a receive window of 30  $\mu$ sec. In addition to the SFP and SATA interfaces, the Data FPGA includes a fast Rocket IO MGT that supports data throughput up to 1.5GSPS [10]. This enables high-speed data transfer to other modules such as on-board computer and hard disks. The 64-bit 66MHz PCI core in PCI FPGA allows communication with the cPCI bus with ease.

Thus, the proposed data acquisition system satisfies all the requirements set by the radar unit and is ideally suited for integration with the Ka-band radar. The plan is to design a prototype version of the data acquisition system to support HRSM at the University of Massachusetts, Amherst with the non Rad-Hard components on the board. A final revision of the board shall be developed that integrates DAS with the Ka-band radar in the future.

## CHAPTER 4

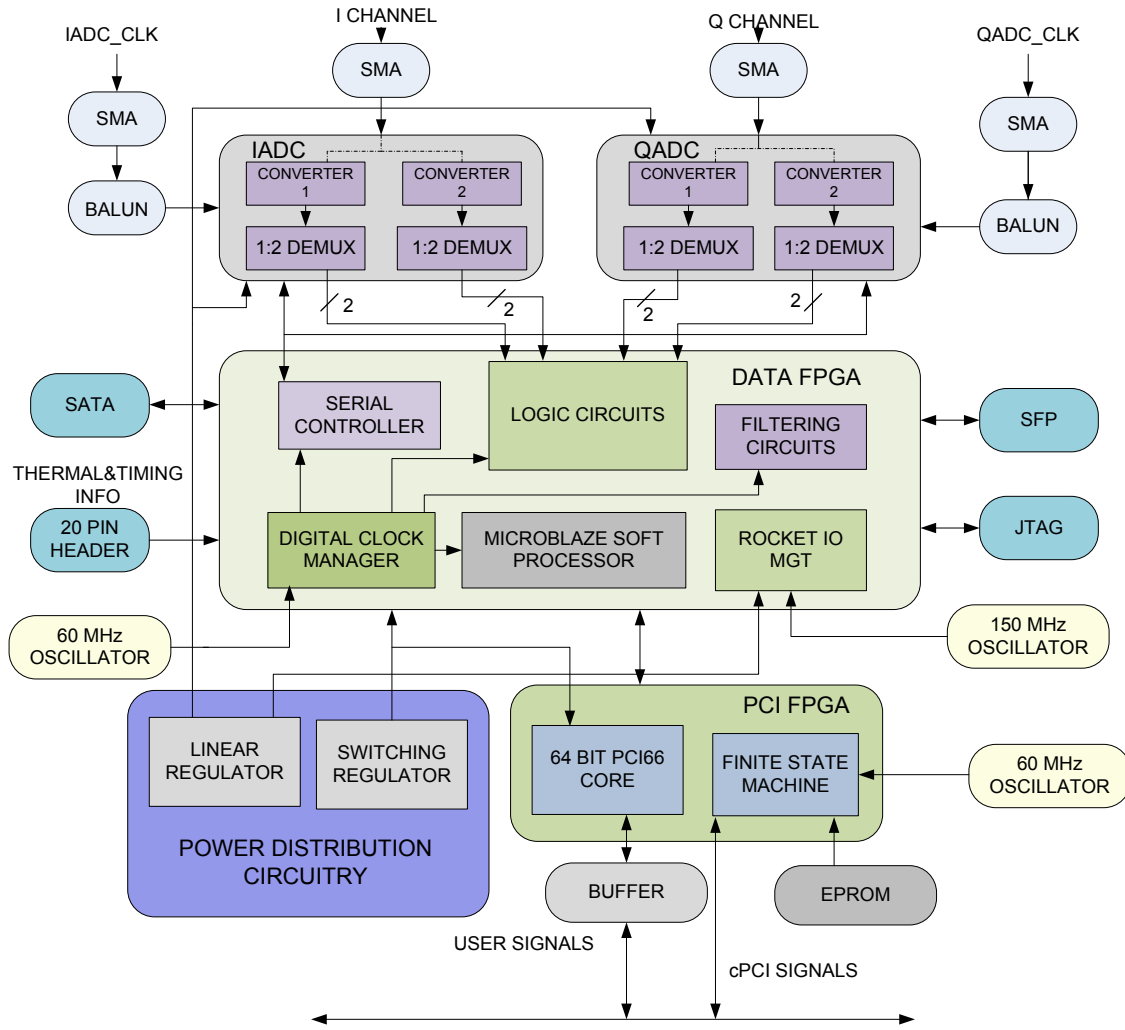
### SYSTEM ARCHITECTURE AND DESIGN DETAILS

This chapter discusses the system architecture and design details pertaining to the components used in the data acquisition system. The key components used in the system are listed below.

- (1) Analog to Digital Converters (ADCs)
- (2) Data FPGA (Virtex-4 XC4VFX140)
- (3) PCI FPGA (Virtex-4 XC4VLX25)
- (4) Linear Regulators and Switching Regulators
- (5) On Board Crystal Oscillators
- (6) High Speed Interfaces such as SFP and SATA
- (7) Compact PCI Connector

The detailed system architecture of DAS is shown in Figure 5. The system receives L-band signals from the down conversion circuitry on two channels, I and Q. The two ADCs, IADC and QADC sample these signals at 3GSPS and digitizes them to 8 bits. The output clock and output data signals of both the ADCs interface to the Virtex-4 FPGA called Data FPGA on its LVDS I/O pins. The system includes another Virtex-4 FPGA called PCI FPGA to establish communication with the cPCI bus. The power distribution circuitry on board comprises of both linear and switching regulators to power different components.





**Figure 5. System Architecture**

The Data FPGA and PCI FPGA form the core of the DAS. The Data FPGA includes the hardware logic for processing the digital data. A 32-bit soft microprocessor called MicroBlaze [18] is instantiated in the Data FPGA that acts as the control module. High-speed interfaces such as Rocket IO MGT, SATA, and SFP are supported by the system. JTAG port and UART interface constitute the debug channels on the board. The following sections describe the implementation details of power distribution circuitry, ADC, Data FPGA, PCI FPGA and the FPGA programming interfaces.

#### 4.1 Power Distribution

The design of power distribution circuitry constitutes an important design phase in any board design project. The selection of regulators is made based on several factors, the important ones being the input voltage range, desired output current, efficiency, and the drop out voltage.

The power distribution circuitry on our system receives two external unregulated voltages on the cPCI bus namely 5V and 3.3V. The various linear and switching regulators generate on-board regulated voltages. Table 1 below lists the regulators on the board, their rated voltage and the devices that the regulator powers.

SI. NO	Regulator	Manufacturer	Type of Regulator	Rated Output Voltage	Rated Output Current	Purpose
1	LT1764 – 1.5	Linear Technology	Linear	1.5V	3A	Powers the Multi Giga bit transceiver (MGT) on Data FPGA
2	LT1764 – 2.5	Linear Technology	Linear	2.5V	3A	
3	NCP5663	On Semiconductor	Linear	1.2V	3A	
4	5920RH	M S Kennedy corporation	Linear	1.9V	3A	Powers the two ADCs
5	PTH05010	Texas Instruments	Switching	1.2V	15A	Powers the internal logic of Data FPGA and PCI FPGA
6	PTH05060W	Texas Instruments	Switching	1.8V, 2.5V, 3.3V	10A	Powers the auxiliary circuits, IO banks of Data FPGA and PCI FPGA

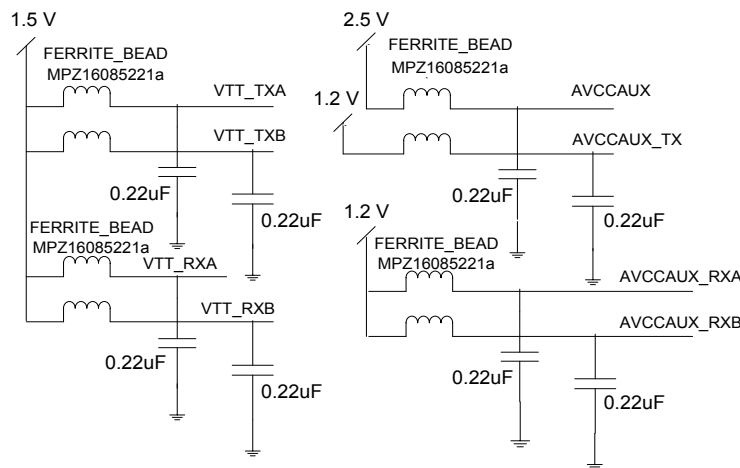
**Table 1. On-board Power Regulators**

Linear regulators [19] [20] [21] are used to derive voltages for the ADCs and the RocketIO MGTs on the Data FPGA. Switching power regulators [22] [23] are used to

generate core voltages, I/O voltages and the auxiliary voltages for the Data FPGA and PCI FPGA. The Data FPGA and the PCI FPGA require 1.2V for internal logic, 2.5V for the digital clock manager (DCM), JTAG and other auxiliary circuits, 3.3V, 2.5V and 1.8V for different I/O banks. Four switching power regulators from Texas Instruments generate these voltages. Certain PCI FPGA I/O banks are powered directly from the cPCI bus. This complies with the cPCI standard. A section is included in this chapter that explains the function of the PCI FPGA in detail.

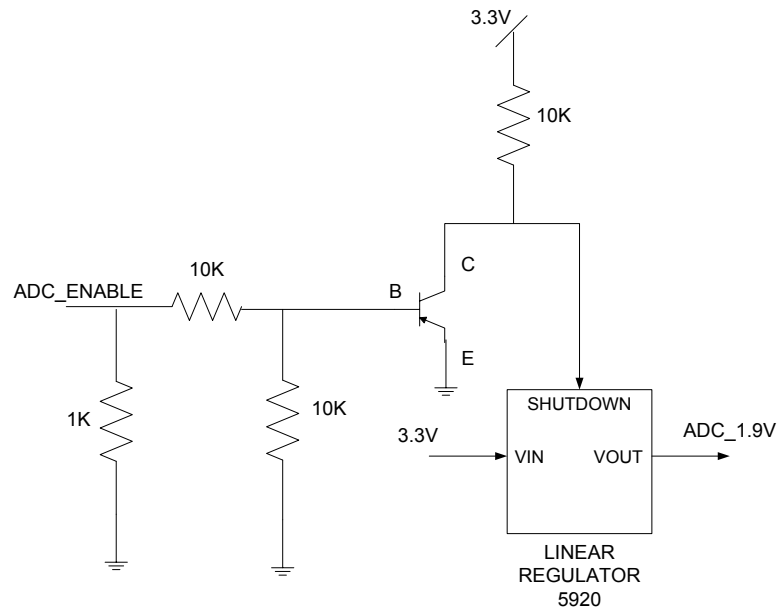
The switching regulators need to be replaced by linear regulators in the space qualified DAS due to the lack of availability of Rad-Hard switching regulators. This is a challenge since most linear regulators have a maximum output current rating of 5A and the FPGAs require more than 5A output current from the regulators.

Separate linear regulators [19] [20] power the MGTs to ensure sufficient noise isolation from the surrounding noise sources such as the switching regulators. The MGTs require a passive high frequency filtering circuit [10] in addition to the separate regulators to function properly. The deployed high frequency filtering circuit is shown in Figure 6.



**Figure 6. High frequency filtering circuit for MGTs**

The ADCs are powered only after the Data FPGA is configured. This design consideration eliminates the risk of active ADC outputs damaging the I/O buffers of Data FPGA during boot up. This is accomplished by controlling an active low shutdown pin of the linear regulator (5920RH) [21] through the Data FPGA. The specific circuit implementation is shown in the Figure 7. The Data FPGA I/O pins are tri-stated before configuration as the Data FPGA HSWAPEN pin [24] is pulled high through a 10K pull-up resistor.



**Figure 7. ADC Enable Circuit**

#### **4.2 Clock Distribution**

The PCB houses three crystal oscillators; one for generating 60/150MHz clock for Data FPGA, another one for generating 60MHz/200MHz for PCI FPGA and the other one for generating 150MHz clock for RocketIO MGT in the Data FPGA. The RocketIO MGT requires a reference clock that needs to have a high degree of accuracy [10]. The

EPSON 2121CA oscillator [25] is used to provide this reference (150MHz) which is in accordance with the part suggested in the RocketIO user guide [10].

The board has provisions for alternate clocks.

- (1) There is an option to provide an external clock to the Data FPGA through an SMA connector J14 (DATAFPGA\_CLK150\_EXT signal in the schematic).
- (2) The Data FPGA can drive the PCI FPGA clock (DATAFPGA\_ALTCLK signal and PCIFPGA\_ALTCLK signal in the schematic).

### **4.3 Analog to Digital Converter**

The PCB comprises of two Analog to Digital converters called as IADC and QADC. Each ADC is an 8 bit dual 1.5 GSPS or single 3.0 GSPS converter. Both IADC and QADC have two internal converters that receive input data on two channels, I and Q and produce outputs on four channels named as I, Id, Q, Qd ('d' stands for delayed output). Effectively, the four outputs represent four consecutive 3GSPS 8-bit data samples.

#### **4.3.1 Dual Edge Sampling**

Each ADC consists of two internal converters that can be interleaved to achieve a 3.0 GSPS sampling rate. This interleaved mode of the ADC is referred to as Dual Edge Sampling (DES) mode of the ADC. The input signal is sampled on both the positive and negative edges of the sample clock in DES mode. Each ADC has two input channels (I and Q), but in DES mode, the input signal is fed only through the I channel [9].

#### **4.3.2 DDR Output Clock and 1:2 Output De-multiplexing**

Each converter within the ADC provides an option of selectable output de-multiplexing that reduce the output data rate to half the input sample rate on each of the

output channels [9]. The ADCs support both Single Data Rate (SDR) and Double Data Rate (DDR) output clocking. In SDR mode, the output clock (DCLK) frequency on each converter is the same as the data rate on its two output channels. In DDR mode, the output clock (DCLK) frequency on each converter is half the input sample rate and the data appears on the output on both edges of the DCLK.

In our system, we use the DES mode with 1:2 de-multiplexing mode for each converter and DDR output clocking. The DCLK frequency is reduced by a factor of eight ( $3\text{GHz}/8 = 375\text{MHz}$ ) with respect to the sampling clock since we are using the 1:2 de-multiplexing mode on each converter and DDR output clock.

### **4.3.3 Alternate Output Clock**

DCLK is the primary output clock of the ADC. Alternatively, we can use DCLK2 as the second output clock to latch the outputs. To enable the alternate output clock, one of the ADC registers must be configured. The DCLK2 signal behaves as an overflow signal (OR) to indicate the out of range condition of an input signal when the alternate output clock is not used.

### **4.3.4 Extended Control Mode**

The configuration and control of the ADC can be controlled in two ways [9].

- a. Non-extended control mode: The user can control the configuration of the ADC by setting several control pins in this mode.
- b. Extended control mode: This mode provides additional configuration through a serial interface and a set of nine registers [9].

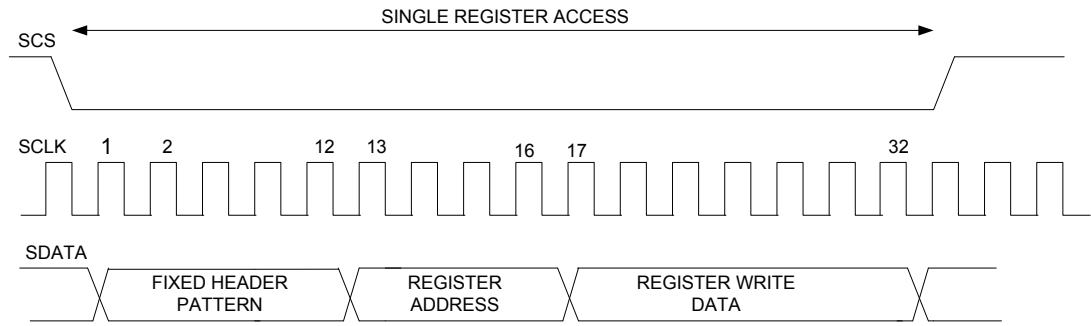
Both the ADCs in our system use extended control mode.

### 4.3.5 Serial Interface

The serial interface consists of three pins; SCLK (Serial clock), SDATA (Serial data) and SCS (Serial chip select). The serial interface provides an option to access nine write only registers. A specific 32-bit word on SDATA line accesses any one of these nine registers. The nine write only registers and their addresses are tabulated in Table 2. The timing diagram is depicted in Figure 8. The SCLK signal is terminated at the end of the chained layout using pull-up and pull-down 100 Ohm resistors.

Sl.NO	Register	Address (in hex)
1	Calibration	0
2	Configuration	1
3	I-channel Offset	2
4	I-channel Full-Scale Voltage Adjust	3
5	Extended Configuration	9
6	Q-channel Offset	A
7	Q-Channel Full-Scale Voltage Adjust	B
8	Sampling Clock Phase Fine Adjust	E
9	Sampling Clock Phase Intermediate and Course Adjust	F

**Table 2. ADC Registers**



**Figure 8. Serial Interface Timing**

#### 4.4 Data FPGA

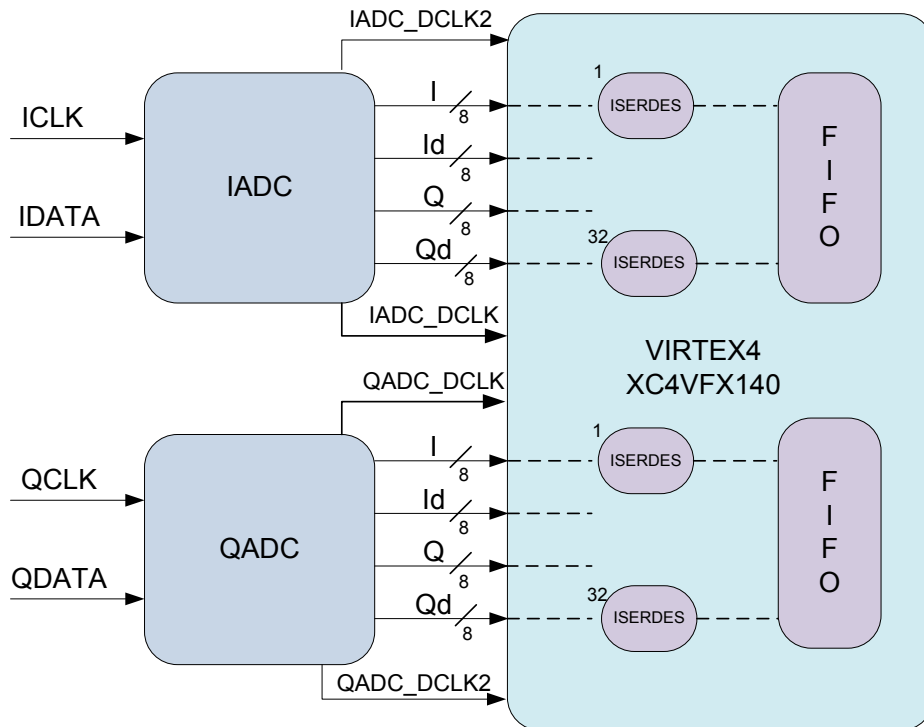
The Data FPGA is one of the key components of the DAS. The Xilinx Virtex-4 XC4VFX140 device is the largest device in the Virtex-4 family with a total of 142,128 logic cells, 768 user I/Os, 9936 Kbits of block RAM, 987 Kbits of distributed RAM, 24 Rocket I/O Multi Gigabit Transceivers, 20 Digital Clock Managers (DCM), and 17 I/O banks [5].



#### 4.4.1 ADC – Data FPGA Interface

The data channels (I, Id, Q and Qd) and clock channels (DCLK and DCLK2) of IADC and QADC interface to the Data FPGA on its LVDS I/O pins. Each output channel is an 8-bit word in LVDS format.

Figure 9 shows the interface between the ADC and the Data FPGA.



**Figure 9. ADC – Data FPGA interface**

It is imperative to choose the appropriate data and clock pins on the Data FPGA that connects the ADC output data and output clock lines. The assignment of output data lines is constrained by the fact that the clock signal on the Virtex-4 FX140 device reaches only three clock regions; its own and the two neighboring regions [17]. There are 24 clock regions in the FX140 device, as shown in Figure 10.

XC4VFX140 CLOCK REGIONS

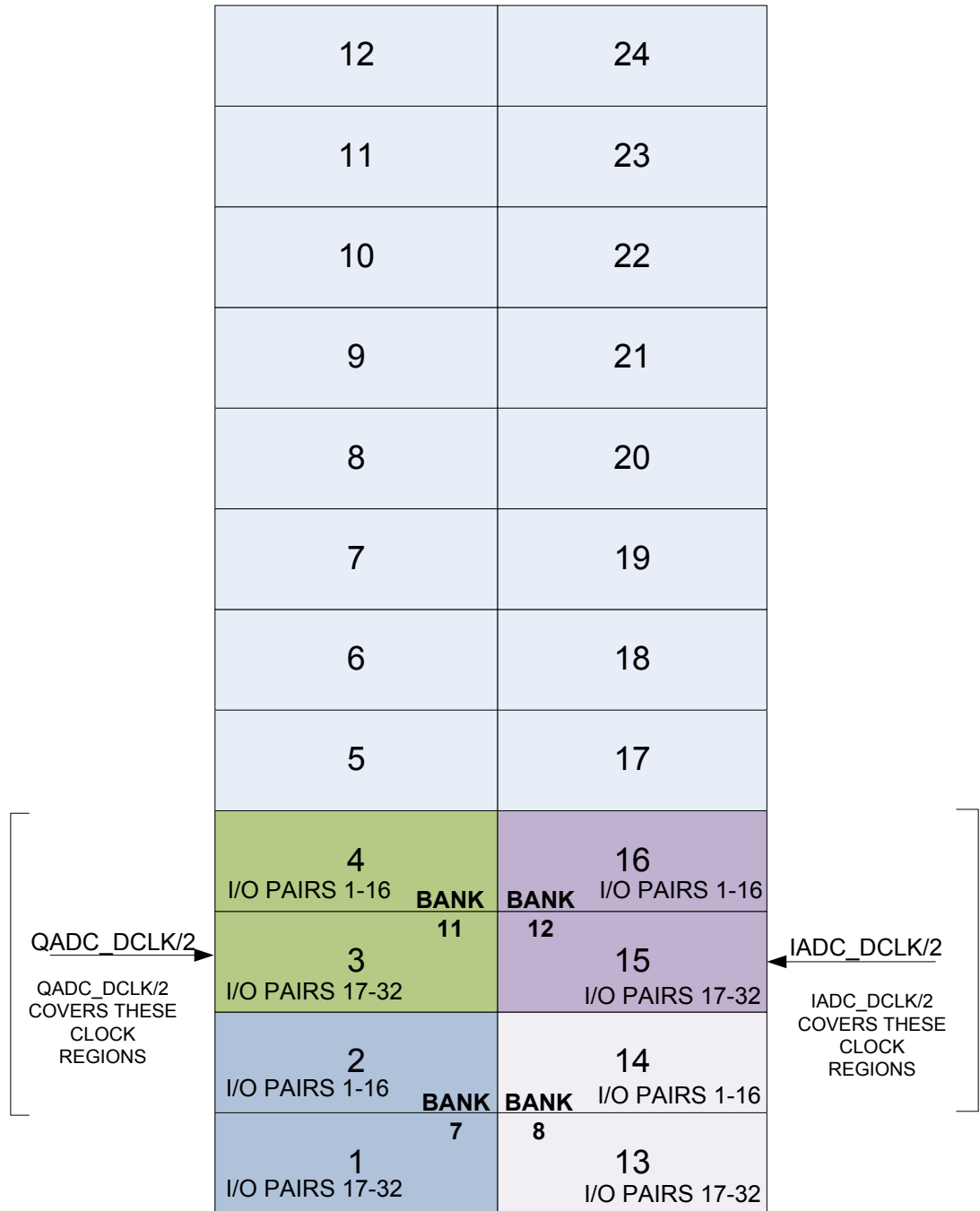


Figure 10. Clock regions in XC4VFX140 device

ADC	Clock/ Data	I/O bank	Clock Region	I/O pair number
IADC	DCLK	12	15	24
	DCLK2	12	15	25
	I	8	14	2, 6, 10, 4, 13, 11, 7, 1
	Id	8	14	8, 9, 14, 12, 16, 5, 3, 15
	Q	12	15	26, 32, 29, 30, 22, 20, 17, 21
	Qd	12	15	16, 15, 14, 27, 28, 31, 19, 18
QADC	DCLK	11	3	24
	DCLK2	11	3	25
	I	7	2	7, 14, 12, 10, 2, 4, 3, 13
	Id	7	2	9, 8, 15, 11, 6, 16, 5, 1
	Q	11	3	21, 17, 28, 18, 22, 30, 27, 26
	Qd	11	3	13, 11, 2, 32, 31, 20, 19, 29

**Table 3. I/O pins on Data FPGA interfacing ADC data and clock pins**

The Table 3 lists the I/O pairs, I/O bank and the associated clock region of the Data FPGA connecting the output data and clock lines of the ADC.

IADC: The DCLK and DCLK2 clock lines are assigned to clock capable pins in I/O bank 12. These clock lines are present in clock region 15. Hence, the data lines must be confined to clock regions 14, 15 and 16. Accordingly, the data pins are assigned to 1-16 I/O pairs in bank 8 and 16 I/O pairs in bank 12 [24].

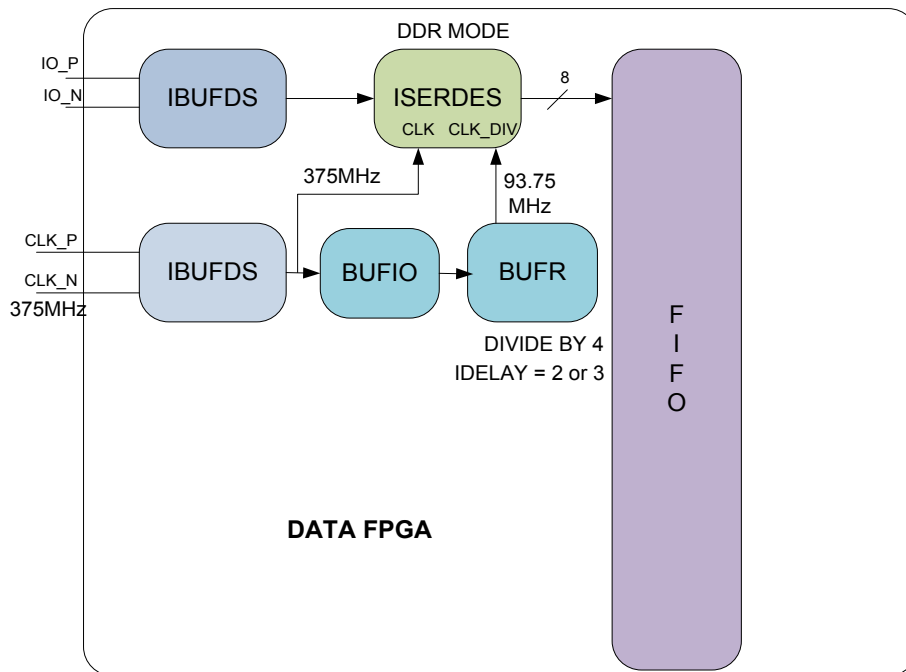
QADC: The DCLK and DCLK2 clock lines are assigned to clock capable pins in I/O bank 11. These clock lines are present in clock region 3. Hence, the data lines must be confined to clock regions 2, 3 and 4. Accordingly, the data pins are assigned to 1-16 I/O pairs in bank 7 and 16 I/O pairs in bank 11 [24].

#### **4.4.2 Logic Circuits**

The Data FPGA includes abundant hardware resources to perform implementation of complex time domain convolution filters. The Data FPGA supports up to 192 18 x18 multipliers, and 9936 Kbit of block RAM necessary for performing filtering operations [5]. At the maximum clock frequency of 150MHz, the Data FPGA

supports processing of up to 170Ksamples between the PRF events, which supersedes the requirement of processing 90Ksamples between PRF events.

The data rate of the digital samples at the input of the Data FPGA is 375 MHz. The Data FPGA fabric does not support such high speeds. However, the Data FPGA includes several dedicated serial to parallel converters or de-serializers called ISERDES that support fast I/O data rates. The ISERDES module supports DDR mode and allows high-speed data reception without the FPGA fabric matching the I/O data rate [17]. In our system, we use ISERDES to create an 8-bit parallel word. This implies the incoming clock rate is divided by a factor of four and the data rate is divided by 8. The specific circuit implemented in the FPGA is as shown in the Figure 11.



**Figure 11. ISERDES and FIFO Implementation**

The input LVDS pins of the Data FPGA are connected to the IBUFDS module that converts differential signals to single ended signals. These signals are then connected to the ISERDES modules. ISERDES receives two clocks [17], one is the undivided

375MHz clock and the other one is the divided 93.75 MHz clock. The BUFR module generates the 93.75 MHz clock, which is a local clock buffer for I/O and Configurable Logic Blocks (CLBs). BUFIO is a bi-directional buffer that needs to be used in conjunction with BUFR. The ISERDES outputs are interfaced to a FIFO queue in the Data FPGA.

#### **4.4.3 Serial Controller**

The IADC and QADC serial interface pins SCS, SCLK and SDATA are connected to the I/O pins of the Data FPGA. A finite state machine (FSM) [26] is implemented in the Data FPGA to enable the access of ADC registers through these pins.

#### **4.4.4 MicroBlaze**

A 32-bit soft microprocessor called ‘MicroBlaze’ [18] is instantiated in the Data FPGA that includes an RS232 interface. The RS-232 interface is used for de-bugging purposes, for reading values out of the FIFO queues in the Data FPGA. An application to access the serial controller in the Data FGPA is implemented in the MicroBlaze.

#### **4.4.5 RocketIO MGT**

The Data FPGA comprises of high-speed interface called RocketIO MGT. Two Rocket IO CoreGen modules [27] are instantiated in the Data FPGA to enable high-speed data transfer to other modules.

### **4.5 PCI FPGA**

Control information will be transferred back and forth between the host computer and the DAS using the cPCI bus. The DAS will plug into the backplane of a cPCI slot using the five connectors on the periphery of the board. Configuration information for the

Data FPGA will be transferred to the board using this interface, although currently it is configured via the JTAG interface for testing purposes.

#### **4.5.1 Compact PCI Interface**

The system implements a 6U version of the Compact PCI interface. The dimensions of a 6U Compact PCI board are 160 mm x 233 mm [17]. The Compact PCI bus is a 64-bit bus that runs at 33 or 66MHz. The board can support either clock speed via a manual on-board U63 jumper setting. There are five connectors J1, J2, J3, J4, J5 on the periphery of the board that interface to the cPCI backplane. J1 and J2 consist of 110 pins each. J1 carries the lower 32-bit cPCI signals while J2 carries the upper 32-bit cPCI signals. J3, J4 and J5 are additional connectors that are intentionally undefined at this point of time. They can be used as a bridge to other buses like VersaModular Eurocard (VME) or Industry Standard Architecture (ISA) in the future [17].

A Xilinx Virtex-4 XC4VLX25 device supports 64-bit 66MHz cPCI core [28]. The system utilizes this core to interact with the cPCI interface. A -10 speedgrade version of the LX25 device will only support a 33 MHz cPCI bus. A -11 speedgrade device will support either a 33 or a 66 MHz cPCI interface. In the final version of the board, this PCI FPGA device will be replaced by Actel Rad-Hard equivalent device RTAX, RT ProASIC3 (flash), or RTSX-SU device.

The cPCI signals interface to two I/O banks on the PCI FPGA. Hence, the I/O voltage pins of these banks are directly connected to 3.3V on the cPCI bus in compliance with the cPCI standard [17]. The UCF (User Constraint File) generated by the 64 bit PCI core [28] is used as reference for assigning PCI FPGA pins to the cPCI connectors. A

series of 10 Ohm stub resistors are placed close to the cPCI connectors to prevent signal reflections.

#### **4.6 SATA and SFP Interfaces**

A Serial Advanced Technology Attachment (SATA) connector and a Small Form Factor Pluggable (SFP) connector are included in our system. This is primarily to transfer data to other modules at up to 1 Gbps. The SFP optical module is a low profile 20-pin connector [11] while the SATA connector is a 9-pin connector [12]. SATA connectors are used for disk interfacing and inter-board transfer of data. SATA allows data transfer rates up to 1.5 Gbps.

The SATA interface includes two transmit and two receive lines connected to the RocketIO MGT pins. Similarly, the SFP interface consists of two transmit and two receive lines connected to the MGT pins. The clock for the RocketIO MGT is derived from 150MHz EPSON crystal oscillator present on board.

#### **4.7 Thermal Sensor**

The thermal diodes on the IADC and the Data FPGA are connected to a temperature sensor (LM95221) to monitor the temperature level of these components. The SCL and SDA outputs of the sensor interface to the Data FPGA I/O pins.

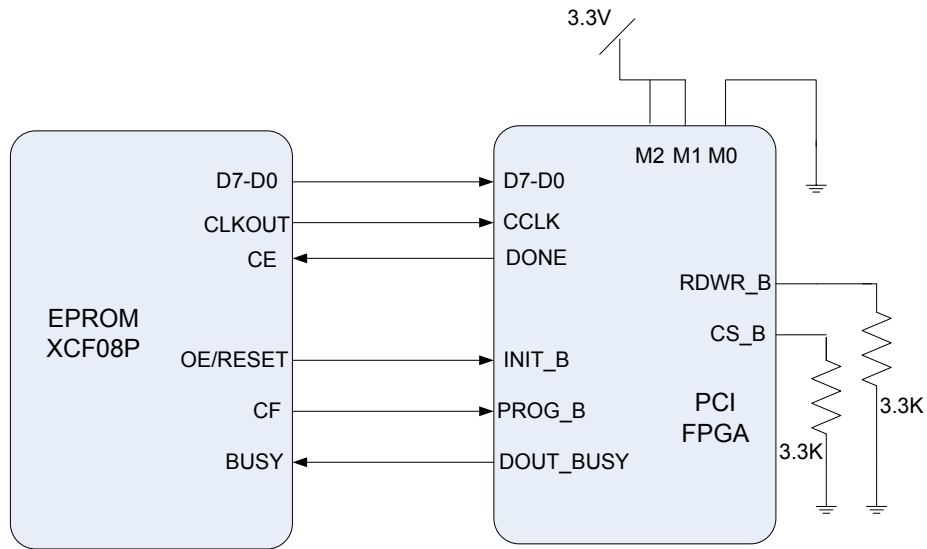
#### **4.8 FPGA Programming Interfaces**

The Data FPGA and PCI FPGA can be programmed through two configuration interfaces; Slave SelectMAP8 and JTAG.

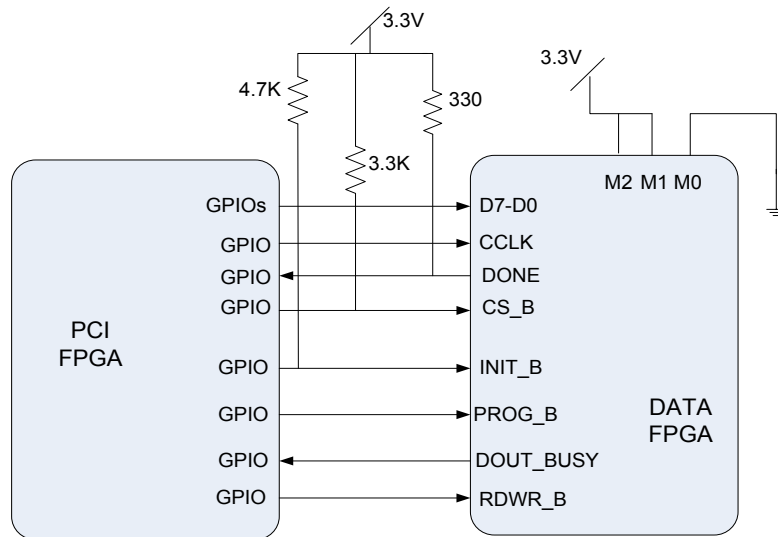
##### **4.8.1 Slave SelectMAP8**

The mode pins M2, M1 and M0 of both Data and PCI FPGA are in logic state  $(M2\ M1\ M0) = (1\ 1\ 0)$  to enable the Slave SelectMAP8 mode [29]. In this mode, the

configuration clock CCLK is an input signal to the FPGA. An 8-bit bidirectional data bus interfaces to the Virtex-4 configuration logic. The Slave SelectMAP8 mode between the Xilinx PROM and PCI FPGA is shown in Figure 12 where as the SelectMAP mode between the PCI and Data FPGA is shown in Figure 13.



**Figure 12. Slave SelectMAP8 mode – Xilinx PROM and PCI FPGA**

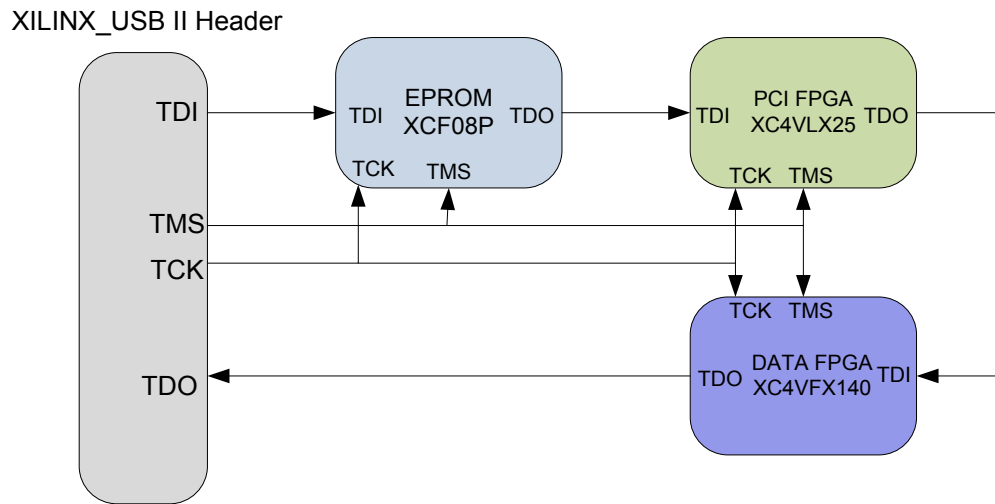


**Figure 13. Slave SelectMAP8 mode – PCI FPGA and Data FPGA**



### 4.8.2 JTAG Interface

JTAG is the simpler of the two programming interfaces available in the Data FPGA. JTAG programming mode overrides any other mode and can always be used irrespective of the mode pin configuration [29]. The JTAG chain on the board includes Xilinx PROM, PCI FPGA and Data FPGA. The scan chain is shown in Figure 14.



**Figure 14. JTAG Scan Chain**

Once the FPGA devices are deployed in space, JTAG programming interface cannot be used. The PCI FPGA reads the configuration file from the non-volatile EPROM (XCF08P) on power up through Slave SelectMAP8 mode. The PCI FPGA then receives the configuration file of Data FPGA through the cPCI interface and configures the Data FPGA through Slave SelectMAP8 mode [30].

## CHAPTER 5

### LAYOUT CONSIDERATIONS

Layout is one of the important steps in any board design project. A lot of emphasis is placed on placement and routing of components, since the data acquisition system needs to support up to a 3GSPS data rate. For successful operation of the system, the layout guidelines listed in the datasheets of all the components present on board needs to be incorporated. This chapter discusses the layout guidelines pertaining to the critical components on system namely ADC, Data FPGA and PCI FPGA.

#### 5.1 Data FPGA Layout Guidelines

The following section lists the guidelines from the Virtex-4 FPGA PCB designer's guide that need to be incorporated into the layout.

1. The capacitor mounting and the power and ground planes of the PCB are the primary sources of parasitic inductances in the PCB.
  - a. Shorter connecting trace lengths minimize the parasitic inductance since it reduces the area the current traverses. Via lengths need to be minimized to reduce inductance.
  - b. The vias should not be shared among multiple capacitors. The capacitor mounting (lands, traces, and vias) contributes about the same amount (or more) inductance than the capacitor's own parasitic inductance. It is recommended to reduce the total number of capacitors than to connect a second capacitor into the vias of an existing capacitor [31].

- c. The thickness of the dielectric plays an important role in minimizing the spreading inductance. The lesser the dielectric thickness between the power plane and its associated ground plane, the lesser is the spreading inductance. Vcc power planes are often directly adjacent to Gnd planes in the stack up. The Vcc Gnd sandwiches offer high frequency decoupling capacitance [31].
2. Layer order is extremely important; high-priority supplies need to be in the top half of the stack-up and low-priority supplies need to be in the bottom half of the stack-up [31]. The Vcc planes that connect to the power supplies with high transient current need to be in the top half of the PCB stack-up. The Vcc-Gnd planes carrying high frequency energy are given higher priority over the Vcc-Gnd planes carrying lower frequency energy. Accordingly, they are either placed on the top half or the bottom half.
3. Decoupling capacitors, as the name suggests are used to perform decoupling function and need to be present closer to the device. Shorter traces lengths are used to connect capacitor lands to vias. In order to reduce the intrinsic parasitic inductance, vias are inserted in between the solder lands for large package capacitors [31].

Not all the high-frequency capacitors need to be placed closer to the FPGA. It is acceptable to place the capacitors around the periphery of the device, provided the separation between the Vcc and the Gnd plane adjacent to it is less than 4mils in thickness [31]. In cases where Vcc-Gnd plane pairs are in the top half of the stack-up, the capacitors are placed on the top surface of the board, around the periphery of the device.

In order to have the power supply noise within half the maximum allowed power supply noise ( $VRIPPLE/2$ ), the Power Distribution System (PDS) needs to have approximately one capacitor per Vcc pin on a per-supply basis. Generally, the rule of thumb is to have one 0.01  $\mu$ F capacitor for every power pin. However, Xilinx recommends using a broad range of capacitor values to cover a broad range of frequencies [31]. The proportion of high frequency capacitors to low-frequency capacitors plays a significant role.

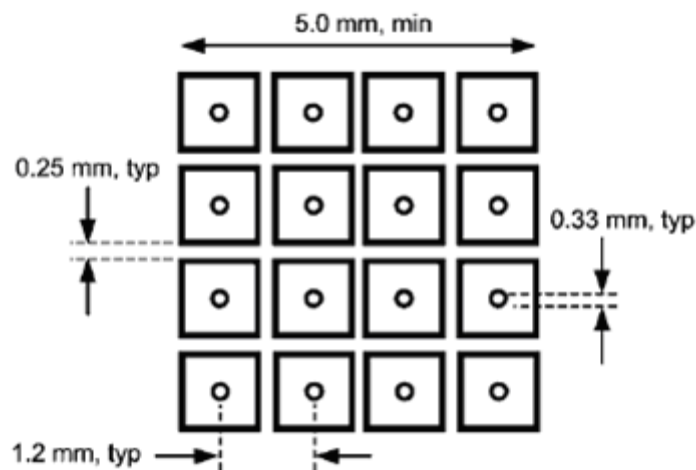
The idea behind using different values of capacitors is to maintain low and flat power supply impedance from 500 kHz to 500 MHz frequency range. Both large value (low frequency) and small value (high frequency) capacitors are needed. It is recommended to include a capacitor in every decade of the capacitor value range to keep the impedance profile smooth and free of anti-resonance spikes. The quantity of capacitors needs to be roughly doubled for every decade of decrease in size in order to maintain a relatively flat impedance profile.

4. In cases where large numbers of external termination resistors are used at the destination, placement of the termination resistors takes priority over the decoupling capacitors. Termination resistors should be closest to the device, followed by the smallest-value decoupling capacitors, then followed by larger-value decoupling capacitors [31].

## **5.2 ADC Layout Guidelines**

The following are the guidelines from National Semiconductor for inclusion of ADC08D1520 devices into the layout.

1. Every power pin needs to have a  $0.0.1\mu\text{F}$  de-coupling capacitor. They need to be placed closer to each power supply pin ( $< 0.5\text{cm}$ ).
2. The 128LQFP package of the ADC08D1520 has an exposed pad on its back to provide a heat removal path as well as electrical grounding to the PCB. This exposed pad needs to be attached to the PCB to remove a maximum amount of heat from the package [9].
3. A thermal land pattern needs to be incorporated on the PCB within the footprint of the package to aid heat dissipation. The recommended thermal land pattern is shown in Figure 15.



**Figure 15. Recommended thermal land pattern for the ADC**

4. The ADC inputs should have shield pads. The DGND on the SMA connector should connect to the DGND on the balun using an etch trace on the top layer for all four SMAs.
5. A heat sink needs to be included in order to reduce the junction temperature. A solder coated copper area of about 2 sq. inches needs to be present on the opposite side of the PCB [9].

6. Each ADC includes a special shield called a dog house to shield them from high frequency noise.
7. A single ground plane needs to be used, instead of splitting the ground plane into analog and digital grounds [9].
8. Coupling between the noisy digital circuitry and the sensitive analog circuitry can lead to poor performance. High power digital components should not be located on or near any linear component or power supply trace or plane that services analog or mixed signal components.
9. The input clock lines should be isolated from all other lines, analog and digital. The generally accepted 90° crossing are avoided. Best performance at high frequencies is obtained with a straight signal path.

### **5.3 Compact PCI Layout Guidelines**

The following are the guidelines with respect to cPCI interface that needs to be incorporated into the layout.

1. cPCI interconnect needs to be customized for standard FR4 board design
2. cPCI 10 Ohm stub resistors should be placed near the cPCI connectors.
3. The cPCI pins should terminate with a 100 Ohm differential impedance [17].
4. The clock lines have to be straight and clock signal loops must be avoided.
5. The signals should not be routed close to the edge of the PCB board.
6. Connectors on clock traces should be avoided.
7. Clock signals need to be routed on the top layer and vias should be avoided. The reason for this is vias change impedance and introduce more skew and reflections.

These layout guidelines were strictly followed during the layout design. In order to comply with the above guidelines, placements of several components were changed and several schematic design changes were incorporated.

- (a) The placement of the ADCs was made close to the Data FPGA. A conscious effort was made to have symmetrical trace lengths for both IADC and QADC signals connecting the Data FPGA.
- (b) The switching regulators were placed far away from the Data FPGA and the ADCs.
- (c) The PCI FPGA was placed closer to the PCI connectors so that the trace lengths connecting them were shortened.
- (d) In order to place and route all the components, 14 layers were incorporated in the PCB.
- (e) Two 9-pin DIN connectors were included instead of the bulky DB-9 connectors to save area.
- (f) 0805 package termination resistors on the LVDS input lines of the Data FPGA were replaced with 0402 package resistors. This allowed shorter stub lengths and the resistors could be placed directly under the pin of the Data FPGA.

## CHAPTER 6

### PRELIMINARY VERIFICATION

The pre-fabrication activities played a significant role in the overall success of the system. It helped us to simulate and verify key components of the system before the PCB was fabricated. It included:

- a) Simulation of analog components on the board.
- b) Analyzing the power requirements of the Data FPGA, PCI FPGA and the two ADCs.
- c) Compilations of Data and PCI FPGA designs

#### 6.1 Analog Simulation

The SMA connectors (J1, J17), the RF transformer/balun (U6, U44), the capacitors (C30, C248) and (C35, C255) and the two ADC (U9, U47) inputs VIN+ and VIN- constitute the analog circuit on the data acquisition system. Signals of frequencies up to 1.5GHz flow through the traces (transmission lines) connecting these components. A simulation environment is created in Ansoft Designer to obtain/verify the S parameters of this analog system. The analog system created in Ansoft Designer software is shown in Figure 16.

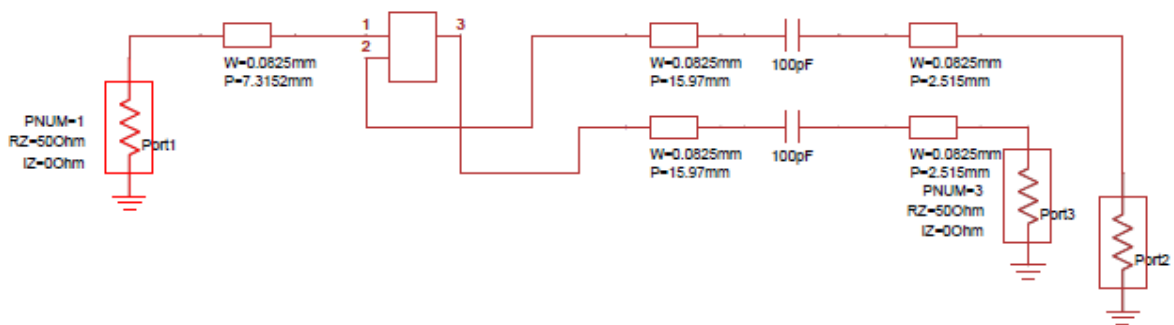


Figure 16. Analog Circuit



The RF transformer ADTL2-18 is simulated as a 3 port device. A \*.s3p file is created using the S parameters provided by the manufacturer of the RF transformer, Minicircuits. Port1 (input port), Port2 and Port3 (output ports) are terminated using 50 ohm resistances. The traces are simulated using the specifications given below. All are Microstrip transmission lines.

### Specifications

Substrate	FR4
Dielectric constant of the substrate ( $\epsilon_r$ )	4.4
Thickness of the Dielectric (H)	4 mils
Metallization	Copper (resistivity - 1.7241 uohm*cm, thickness - 0.675 mils)

**Table 4. PCB Specifications**

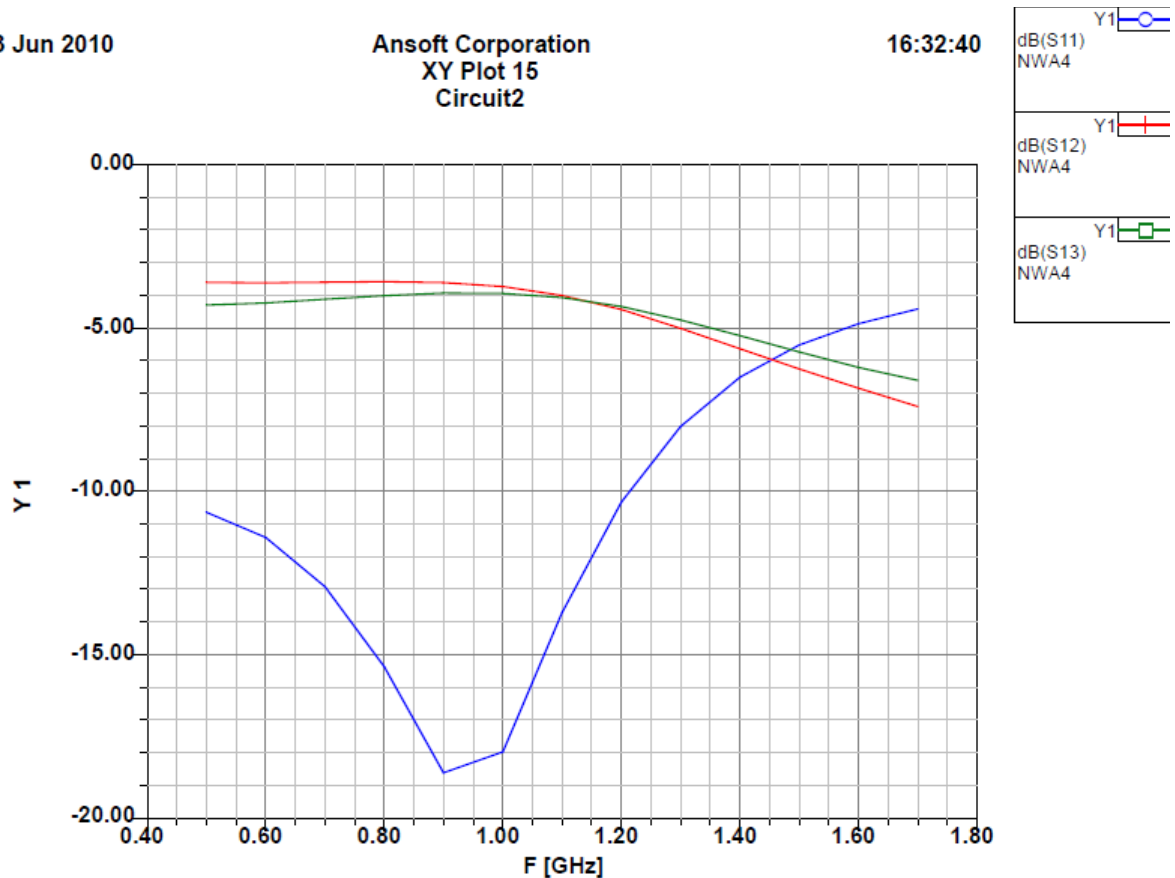
Trace width	(in mils)	(in mm)
SMA -> Balun	3.25	0.08255
Balun -> 100pf cap	3.25	0.08255
100pf cap -> ADC input	3.25	0.08255

**Table 5. Analog Circuit - Trace Widths**

Trace length	(in mils)	(in mm)
SMA -> Balun	288	7.3152
Balun -> 100pf cap	629	15.97
100pf cap -> ADC input	99	2.515

**Table 6. Analog Circuit - Trace lengths**

The circuit in Figure 16 is analyzed over a frequency range of 500MHz to 1.7GHz. The S parameters S11, S12 and S13 are obtained and shown for the entire system in the Figure 17.



**Figure 17. S parameter plot for the analog section**

S11 represents the input return loss, S12 and S13 represents the insertion losses through the paths 1-2 and 1-3 respectively.

The frequency of interest is 1.5GHz. From the plots,

$$S11 @1.5\text{GHz} = -5.53 \text{ dB}$$

$$S12 @1.5\text{GHz} = -6.18 \text{ dB}$$

$$S13 @1.5\text{GHz} = -5.66 \text{ dB}$$

S11 should be ideally low indicating less reflection (order of -10dB). A -5.53dB corresponds to 27.98% of the power being reflected back to Port 1. S12 and S13 should be ideally high indicating that most of the power is propagated to port2 and port3 from port1. A -6.18 and -5.66 dB corresponds to 25% and 27% of power being transmitted

from port 1 to ports 2 and 3 respectively. Although S11, S12 and S13 parameters were deviating from the expected values, the results were acceptable since these calculations were performed using a theoretical based simulator (with approximations for effective dielectric constant, thickness of etches between layers, and thickness of the traces).

## 6.2 Power budgeting of the board

An estimate of the total power consumption of the board was made before the board fabrication. The estimate was helpful to choose the appropriate linear regulators and switching regulators with appropriate output current ratings. To estimate the total static and dynamic power consumption of the Data FPGA, XPOWER analyzer [32] of the Xilinx ISE feature was used. Two MGTs, all the I/O pins, input and output buffers were instantiated in the Data FPGA. Dummy logic was created utilizing all the inputs to the system. The tool estimated the total power consumption of the Data FPGA assuming random inputs and the power averaged over the time period.

A similar experiment was performed using the PCI FPGA. A 64-bit 66MHz PCI core was instantiated in the PCI FPGA. The maximum power consumption on each ADC in active mode is about 2W [9] in active mode. The total power consumption estimate is tabulated in Table 7.

Component	Power Consumption
Data FPGA	6 W
PCI FPGA	2 W
ADCs	4 W
Others	1 W
Total	13 W

**Table 7. Power Consumption Estimate**

### **6.3 Data FPGA Compilations**

The ADC – Data FPGA interface was critical to the overall functioning of the entire system. The output data channels and the clock channels from the two ADCs had to be connected to the appropriate I/O pins on the Data FPGA. The pin assignments were constrained by the fact that the clock signal on the Virtex-4 XC4VFX140 device reached only three clock regions; its own region and the two neighboring regions [33]. In other words, we had to ensure that we assign the output data of ADCs to only those I/O pins of the Data FPGA that were within three clock regions of the clock pin.

Hence, several test designs were created in Xilinx ISE that included the components IBUFDS, ISERDES, BUFR, BUFIO and FIFO. The user constraint file (UCF) contained pin assignments that were made on the schematic. Synthesis, translation, mapping, place and route were run in the ISE software. The compilations were successful with no errors, which meant that the pin assignments made were correct.

## CHAPTER 7

### TESTING THE PROTOTYPE HARDWARE SYSTEM

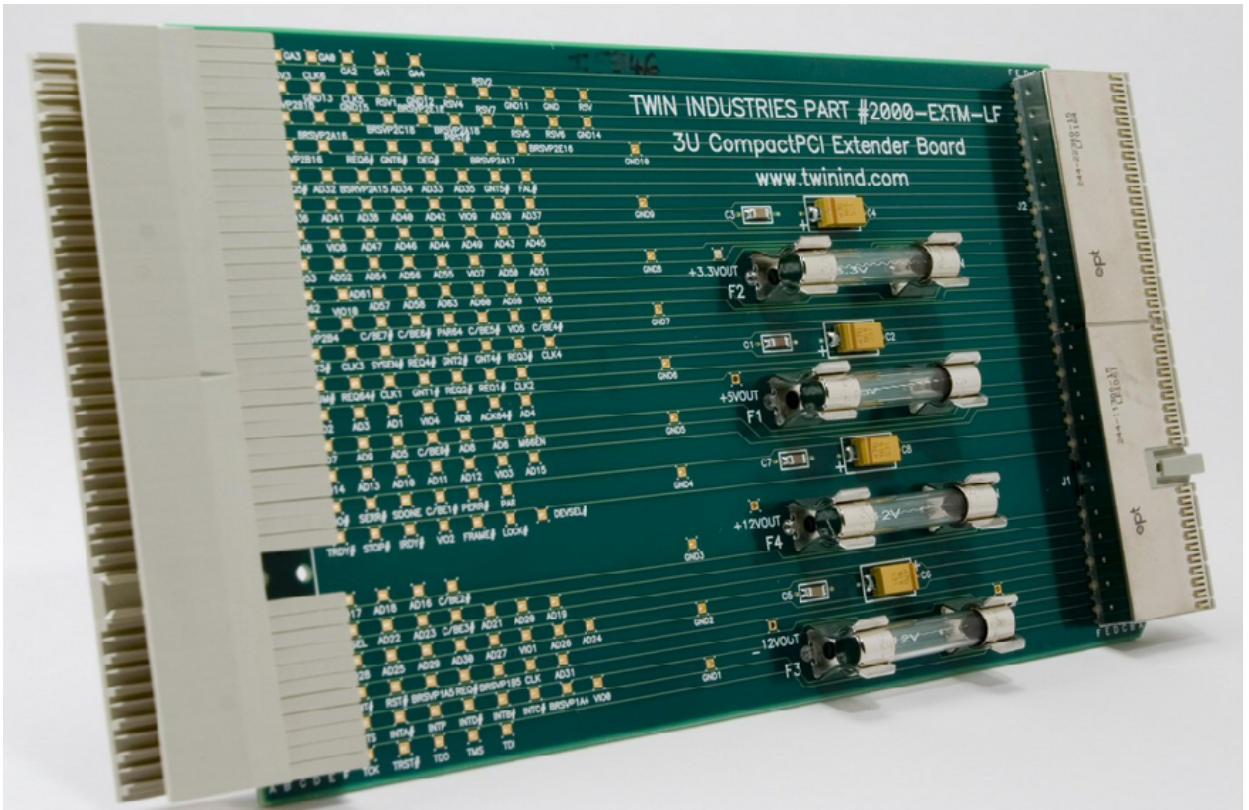
A systematic test procedure was followed to test the individual components on the board. The test methodology developed was independent of the integration of DAS with the down conversion circuitry. The experiments were performed by connecting the board to a customized test set up in the laboratory. The results of these experiments served as a proof of concept of the design of the system.

#### **7.1 Precautions and Safety**

Extreme care was taken while handling the board, anti-static straps were used at all times to ensure that the static charges discharge to ground and do not affect the components present on the board. The input voltages to the board were provided only after double-checking the voltage levels from the power supply to avoid any mishaps due to voltage fluctuations.

#### **7.2 Powering the board**

One of the challenges we faced during the testing process was figuring out a technique to provide input voltages to the board. Normally, the board receives input voltages via the cPCI pins. Since we did not have the set up to connect our board to the cPCI interface at the university laboratory, we had to use an external 3U CompactPCI extender board from Twin Industries. The extender card was plugged into P1 and P2 slots on our board and input voltages 3.3V and 5V were given to the metal holders on the 3.3V and 5V fuses respectively. A 3U extender card was sufficient since all the active signals and voltage pins were connected to P1 and P2 interfaces of the cPCI slot. The screen shot of the extender card is shown in Figure 18.



**Figure 18. 3U Extender card from Twin Industries**

The first test that we performed post fabrication was to check for any manufacturing defects such as trace discontinuity, accidental opposite connections of polarized capacitor leads while assembly and other soldering related manual checks. We verified that there were no manufacturing related defects.

### **7.3 FPGA Test Designs without the Clock**

The next step was to verify whether the programming of both the FPGAs was possible through the JTAG interface. In order to perform this experiment, two test designs were created in Xilinx ISE, one for Data FPGA and the other for PCI FPGA. The Data FPGA was loaded with a configuration file that included a simple program to turn on two out of the four LEDs on the board. For this experiment, the 60MHz oscillator was not mounted in the socket U11 provided on the board. Both the ADCs were shut off.

Similarly, the PCI FPGA was loaded with a configuration file that turned on two LEDs on the board.

To program the FPGAs, iMPACT software was used. iMPACT provides an option of automatically detecting the JTAG scan chain on the board. This was used to verify whether the scan chain was properly set up on the board. On power up, iMPACT detected three devices in the scan chain; a Xilinx PROM XCF08P connected to the Xilinx Virtex-4 XC4VLX25 PCI FPGA, which in turn connected to the Xilinx Virtex-4 XC4VFX140 Data FPGA. This is exactly the way we had set up the scan chain in the hardware. The scan chain detection implied that the JTAG connections were properly made on these three devices.

Next, the FPGAs were programmed one at a time with their respective configuration files through the JTAG port. iMPACT successfully programmed both the FPGAs. This was a significant milestone in the testing process since this experiment verified that the FPGA voltage pins were properly assigned on the board for both FPGAs.

#### **7.4 Flash Programming**

The configuration file of the PCI FPGA was next stored in the Xilinx PROM XCF08P device. On subsequent power ups, PCI FPGA read the configuration file from the PROM and was configured.

#### **7.5 PCI FPGA Test Plan**

There were no tests conducted for the PCI FPGA at the UMass lab except for the FPGA configuration test. Long-term plans include instantiation of the 66 MHz 64-bit PCI core and the design of a finite state machine that accepts data from the cPCI bus and configures the Data FPGA.

The rest of the experiments focused on various test designs involving the Data FPGA and the two ADCs.

### **7.6 FPGA Test Designs with the Clock**

A test design was created that included a clock in the design. This was used to blink four LEDs on the board each at a different frequency. For this experiment, the 60MHz oscillator was mounted in the socket U11. The ADCs were disabled.

After the Data FPGA was programmed, the LEDs started blinking at different frequencies, which confirmed the operation of the FPGA with the clock

### **7.7 Establishing the Serial Communication**

The next step was to test the ADCs on the board. The idea was to configure the ADC in test mode and observe the fixed pattern output on the four ADC output channels. Due to the space constraint, the board houses a 9-pin DIN connector instead of DB9 connector for serial communication. An external connector is used to connect DB9 serial cable and DIN connector.

The serial communication on Data FPGA was established by instantiating a soft microprocessor called 'MicroBlaze' that includes a UART as peripheral. Serial data was transmitted and received using the standard Application Programming Interfaces (APIs) available in MicroBlaze. A test application was written in C language with UART APIs in MicroBlaze. When the application was loaded, it was able to send and receive data through the serial interface at a baud rate of 9600 bytes per second. We used the serial interface as a debug interface for various purposes during the course of validation.



## **7.8 ADC Test Designs**

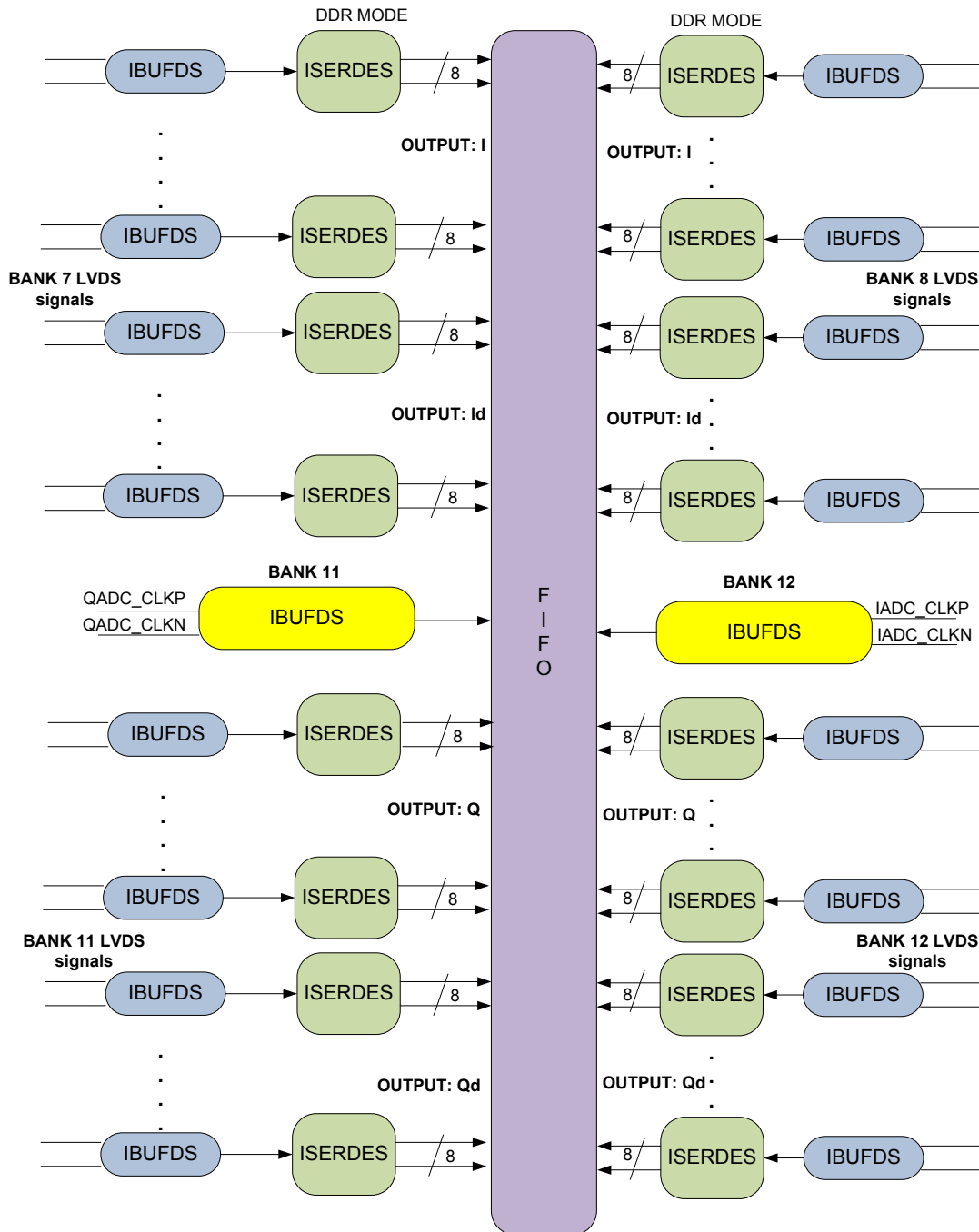
### **7.8.1 IADC Test Mode Configuration**

A test design was created in Xilinx ISE to enable the IADC, configure it (to enter into test mode), collect the outputs and store them in FIFO queues in the Data FPGA. A MicroBlaze processor was created with a UART interface using the base system builder wizard in EDK software and instantiated in the ISE software. Signals from the MicroBlaze processor were given to initiate the serial controller in the Data FPGA.

Specifically, the following registers in IADC were configured; a) Configuration Register and b) Extended Configuration Register. An input clock of 200MHz was given to the IADC clock pin (minimum clock frequency) [9]. In 1:2 DeMux and DDR mode, the output clock DCLK runs at 50MHz and can be given to the write clock of the FIFO queue without using the SERDES interface. The extended configuration mode setting ensured the ADCs enter the test mode. When the system was powered up, a specific test pattern [9] appeared on the four output channels of IADC. This was collected in the FIFO queue of Data FPGA and printed on the hyper-terminal through the UART interface.

### **7.8.2 IADC Test Mode Configuration at higher input clock frequencies**

The input clock speeds on the IADC clock were cranked up steadily from 200MHz until 1.5GHz in multiples of 100MHz. Since the FIFO queues in Data FPGA could not handle such high speeds, customized input de-serializers (ISERDES) were used in the FPGA that allowed FPGA fabric to run slower than the I/O [17]. The implementation block diagram is shown in Figure 19.



**Figure 19. Circuit implementation for higher input clock frequencies**

The ISERDES modules accepted a high-speed clock and divided clock as inputs, and generated output data that transitioned on every edge of the high-speed clock (DDR mode). The BUFR module, a local clock buffer with respect to the input clock, provided

the divided clock. BUFR block provides a divided by four clock to the ISERDES module. The ISERDES modules in Data FPGA were configured as 1:8 de-serializers in DDR mode. A 1:8 de-serializer was created by instantiating two ISERDES blocks, master ISERDES and slave ISERDES. The ISERDES module produced eight samples per input pair on every edge transition of the divided output clock. Since there were 32 LVDS input pairs connected to the Data FPGA from the IADC, 64 ISERDES blocks were instantiated in the Data FPGA and together these 64 blocks produced a 256 bit (32 samples \* 8 bits/sample) output data. These data values were then interfaced to a 256 wide, 2048 deep FIFO module.

The FIFOs in Data FPGA were created using block RAM. The write and read widths were different since 256 bits were written to the FIFO at once and only 8 bits were read out by the MicroBlaze. The write clock of the FIFO was synchronized with the divided clock of ISERDES module and the read clock was synchronized with the MicroBlaze clock. The full and empty signals were used by the MicroBlaze as FIFO indicators. The test pattern [9] appeared on all four output (I, Id, Q, Qd) channels until the clock frequency was 1.3GHz. Beyond 1.3GHz, few samples were lost. At 1.5GHz, a large number of samples were lost.

### **7.8.3 IADC Test Mode Configuration beyond 1.3GHz**

At input clock frequencies greater than 1.3GHz, few samples were lost from the test pattern. This was because of the hold time violation resulting in race condition (early data, late clock scenario) [34]. The data and clock path in the Data FPGA are shown in Figure 20.

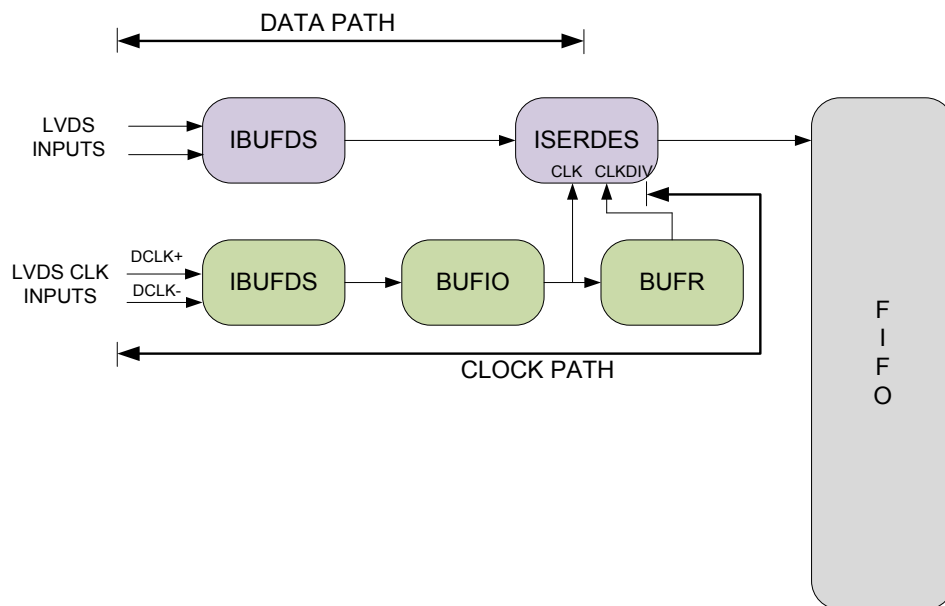
The clock path consisted of modules IBUFDS (input buffer to convert differential signals to single ended signals), BUFIO (input output buffer used in conjunction with BUFR) and BUFR (local clock buffer capable of generating divided clocks) where as the data path consisted of modules IBUFDS and ISERDES. Our analysis showed that the data path had less delay than the clock path and hence few samples were violating hold time on the ISERDES module. The analysis is shown below; the delay on the IBUFDS module is neglected since it is present on both the paths.

a. Clock path

$$\begin{aligned} \text{Total delay} &= \text{BUFR delay} + \text{BUFIO delay} + \text{ISERDES hold time} + \text{routing delay} \\ &= 500\text{ps (BUFR+BUFIO) [35]} + 340\text{ps [35]} + 150\text{ps} = \sim 1000\text{ps} \end{aligned}$$

b. Data path

$$\begin{aligned} \text{Total delay} &= \text{Routing delay from IBUFDS to ISERDES} + \text{hold time of the data} = \\ &360 \text{ ps [34]} + 560 \text{ ps [9]} = \sim 920 \text{ ps.} \end{aligned}$$



**Figure 20. Data path and Clock path in Data FPGA**

In order to overcome this problem, a combinational logic delay had to be inserted on the data path. The ISERDES module provided this option of inserting programmable delay element called IDELAY on the data path [33]. IDELAY, a 64-tap wrap around delay element was inserted which delayed the incoming signals to the ISERDES block by an amount specified by the attribute IOB\_DELAY\_VALUE. The possible values are 0 to 63. Each IDELAY element provides approximately 78ps of delay ( $78 \times 0\text{ps} = 0$  to  $78 \times 63\text{ps} = 4.914\text{ns}$  of delay is possible).

Approximately 70-120ps delay was necessary on the data path. IOB\_DELAY\_VALUE of two or three addressed the problem. Beyond three IDELAYs, the set up time on the ISERDES modules were violated (late data, early clock scenario).

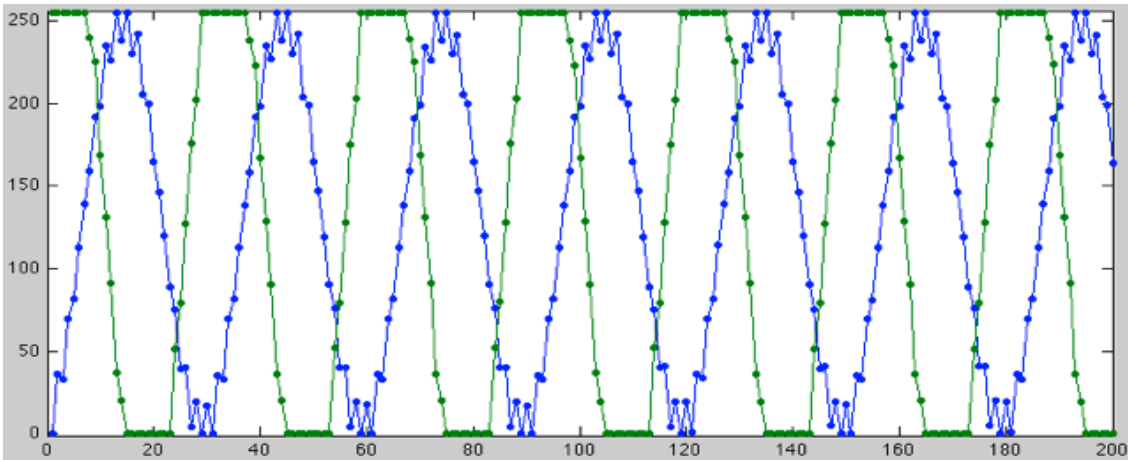
#### **7.8.4 QADC Test Mode Configuration**

After the verification of IADC, the same procedure was repeated for the QADC. The results of the experiment were very similar to the one we obtained with the IADC. The test pattern were observed accurately on the four output channels of the QADC until the input clock frequency of 1.3GHz. Beyond 1.3GHz, there was hold time violations and as before, IDELAY modules were inserted in the data path to solve the problem (IOB\_DELAY\_VALUE of 2 or 3)

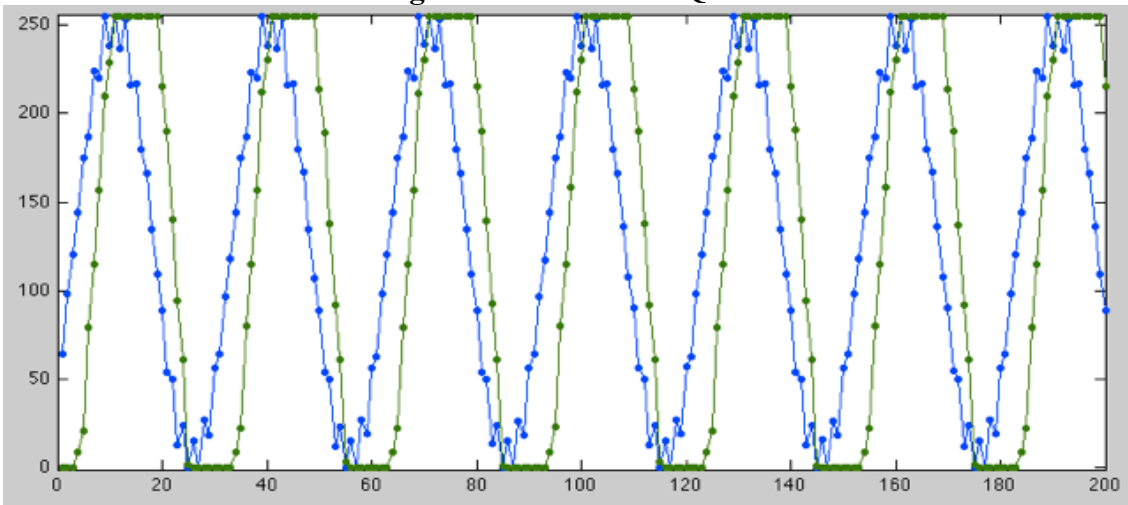
#### **7.9 Sampling the Input Signal**

After the verification of test mode pattern, the next step was verifying the working of IADC and QADC for real input signals. Two cosine waves of 100MHz, one 500mV (p-p) and the other 850mV (p-p) were given to the IADC and QADC. The input clock was set at 1.5GHz.

The outputs from IADC and QADC were collected and processed. The plots are shown in Figure 21 and Figure 22 with the X-axis representing the sample number and Y-axis representing the sample value. The blue lines and dots represent the signal with 500mV (p-p) and the green lines and dots represent the signal with 850mV (p-p).



**Figure 21. Data Plot - QADC**



**Figure 22. Data Plot - IADC**

There was a good correlation between the input signal given to the two ADCs and the plots with respect to both frequency and amplitude. However, these signals had to be corrected for time skew and amplitude offsets.

## 7.9.1 Time Skew and Amplitude offset Corrections

### 7.9.1.1 113MHz Input Cosine Wave

Time skew and amplitude offset were calculated for different channels by taking the Qd channel as a reference. The signal generator used for this purpose was a Tektronix Arbitrary Waveform Generator (AFG3252). Table 8 and Table 9 shows the values of time skews and amplitude offsets for QADC and IADC.

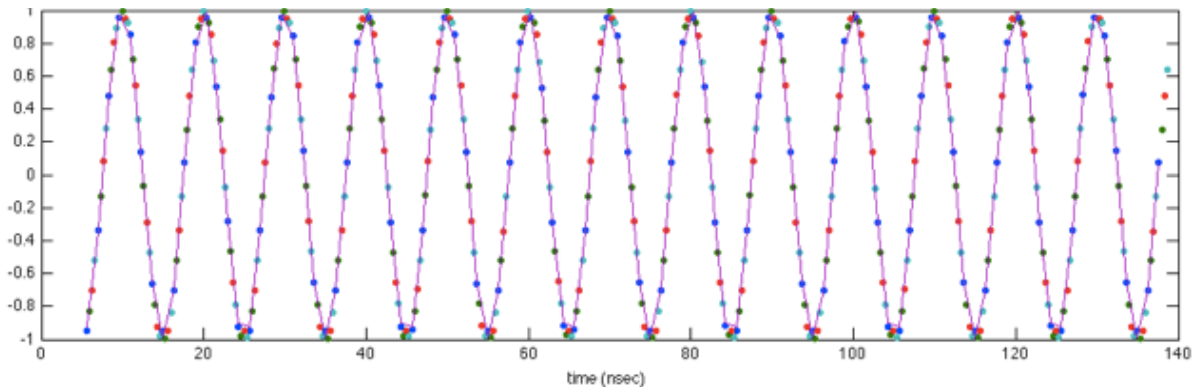
Channel	Time skew (in ps)	Gain Correction
Q	4.41	1.21
Id	0.73	1.00
I	4.20	1.21

**Table 8. QADC -Time Skew & Amplitude Correction for 113 MHz input**

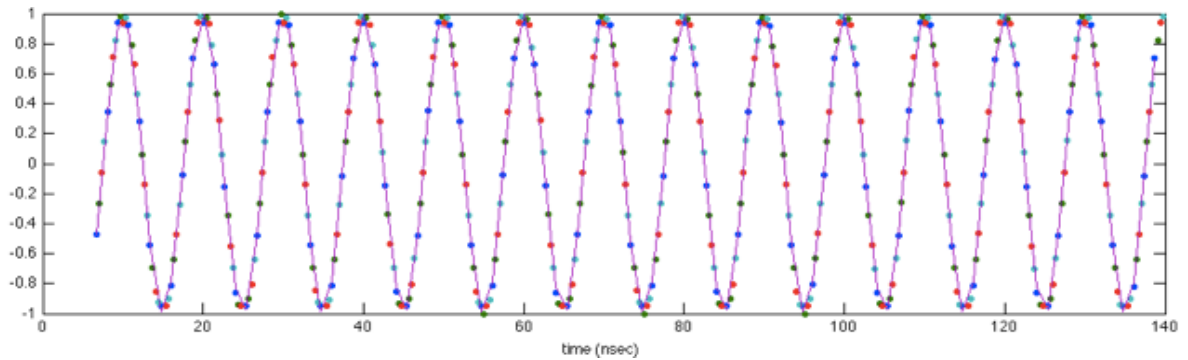
Channel	Time skew (in ps)	Gain Correction
Q	1.39	1.21
Id	-0.42	1.00
I	3.98	1.21

**Table 9. IADC -Time Skew & Amplitude Correction for 113 MHz input**

To correct for the time offset, the skew mentioned in Tables 7 and 8 must be subtracted from the associated channel. To correct for the gain, the number above must be multiplied by the data from the associated channel. Figure 23 and Figure 24 below show the IADC and QADC data points for the first 400 points (or 140ns) after they have been corrected for time skews and amplitude offsets.



**Figure 23. Data plot of QADC after correction**



**Figure 24. Data plot of IADC after correction**

The plots show components in five colors. The dots relate to the four channels of each ADC. They follow one another in color sequence, going from blue, to green, to red, to cyan, and then all over again. The magenta line is a 100 MHz cosine function that is fit to the data. We observed the difference between the fit and the samples collected by the ADCs, and the fit was excellent.

The time skew and amplitude-offset numbers could be fine tuned and more accurate by choosing an input signal whose frequency is not a multiple of the sampling frequency. In other words, choosing 100 MHz for the signal, and 1.5 GHz for the clock meant that every cycle, the waveform was sampled in the same place (currently there are 30 points per signal cycle). This made the quantization errors correlated, which caused inaccuracies in the estimation routines. Hence, if we were to use a signal frequency of 133 MHz (for instance), the number of points per waveform would be 22.5564, which would take many cycles of the sampled waveform before the quantization errors became correlated.

### **7.9.1.2 133MHz Input Cosine Wave**

An accurate measurement was obtained for time skew and amplitude offset by providing a 133MHz cosine input signal. Two signal generators were used for this



experiment; a) Agilent E4437B and b) Tektronix Arbitrary Waveform generator AFG3252. This was done to verify whether the superior quality generator (Agilent E4437B) had any effect on the performance of the ADC.

The results are tabulated in the Table 10, Table 11, Table 12 and Table 13.

**Agilent Signal Generator**

Channel	Phase (deg)	Skew (ps)	Sinad (db)	Gain error
Qd	50.25	0.00	43.99	1.27
Id	66.26	0.99	43.59	1.56
Q	82.13	-0.79	43.96	1.27
I	98.24	2.21	43.14	1.56

**Table 10. QADC -Time Skew & Amplitude Correction for 133 MHz input (Agilent waveform generator)**

Channel	Phase (deg)	Skew (ps)	Sinad (db)	Gain error
Qd	61.09	0.00	43.51	1.29
Id	77.16	2.24	43.25	1.60
Q	93.03	0.30	42.60	1.28
I	109.12	3.18	43.24	1.60

**Table 11. IADC -Time Skew & Amplitude Correction for 133 MHz input (Agilent waveform generator)**

**Tektronix ARB**

Channel	Phase (deg)	Skew (ps)	Sinad (db)	Gain error
Qd	82.09	0.00	42.11	1.32
Id	98.11	1.28	42.03	1.62
Q	113.97	-0.77	41.91	1.32
I	130.08	2.33	42.04	1.62

**Table 12. QADC -Time Skew & Amplitude Correction for 133 MHz input (Tektronix waveform generator)**

Channel	Phase (deg)	Skew (ps)	Sinad (db)	Gain error
Qd	132.30	0.00	40.36	1.33
Id	148.38	2.40	42.50	1.66
Q	164.24	0.29	41.57	1.34
I	180.35	3.38	40.74	1.65

**Table 13. IADC -Time Skew & Amplitude Correction for 133 MHz input (Tektronix waveform generator)**

The first column is the ADC output channel, the second column represents the phase measured for the incoming signal (the phase of the Qd channel is arbitrary, as it depends on when exactly we begin collecting data relative to the input signal).

The third column represents the clock skew that is relative to the Qd channel (the reference). The measured phase in the second column is converted into time (based on the period of the 133 MHz measured signal). This time is compared with the "correct" sample time for a 3GSPS clock. Any difference between the "correct" time and the measured "phase" time is termed skew. Ideally, it should be zero. If it is positive, then the measurement is taking place a little after the time that it really should be and if it is negative, the opposite is happening. For example, the skew for the I-channel is about 2 ps different than the Q-channel for the Agilent generator. This is 0.3% (or 3/1000th) of a 1.5 GHz clock cycle, or equivalently, about 1mm of differential path length, that is internal to the ADCs.

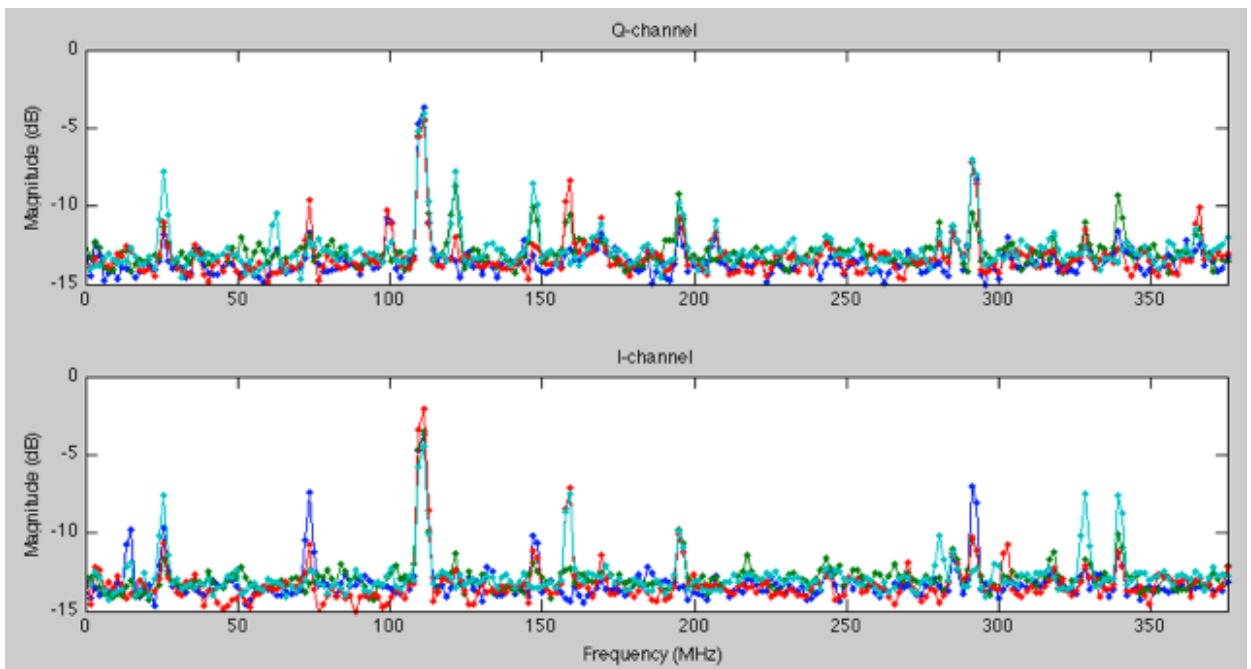
The fourth column (sinad), is the signal power to (noise and distortion) ratio. This value can be converted to the effective number of bits by using the formula:  $ENOB = (SINAD - 1.76)/6.02$ . A SINAD of 44 dB gives an ENOB of 7 bits (43 dB -> 6.85 bits; 42 dB -> 6.7 bits; 41 dB -> 6.5 bits; etc.). The Agilent signal generator has a better SINAD than the Tektronix ARB, which is probably reflective of the quality of the signal sources.

The fifth and last column shows the gain error. This is related to the peak-to-peak voltage of our input waveform compared to the peak-to-peak setting on the ADC. The difference between the measured full-scale range and the ADC set full-scale range is

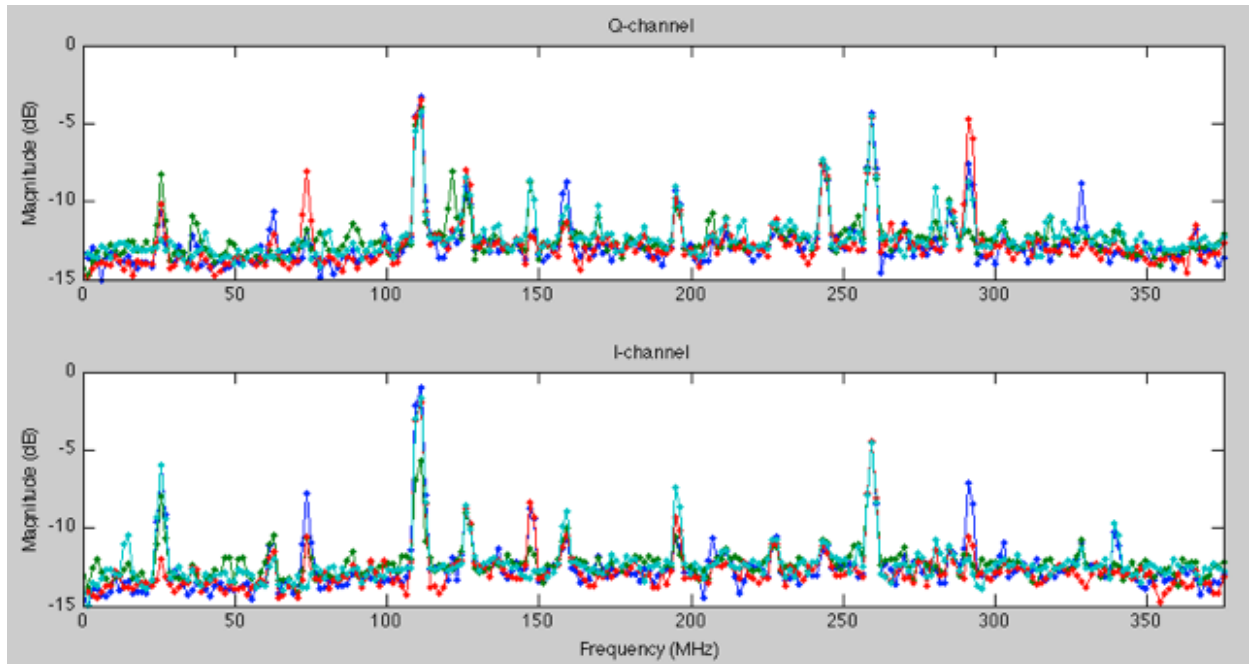
called the gain error (e.g.  $1.56 = 700/445$ ). This type of error is not problematic as long as it is consistent.

### 7.10 Error Spectra

Error spectra are determined by subtracting the measured time series of the ADC from a theoretical 133 MHz waveform that has been corrected for phase and amplitude errors. This type of plot will show if there are external signals leaking into our signal path as well as highlight the presence of higher order frequency products caused by distortion (e.g. 266 MHz, etc.).



**Figure 25. Error spectra - Agilent generator E4437B**



**Figure 26. Error spectra - Tektronix Waveform Generator AFG3252**

The error spectra for both waveform generators are plotted and shown in Figure 25 and Figure 26. In these plots, four colors are shown, one for each channel of the IADC and QADC.

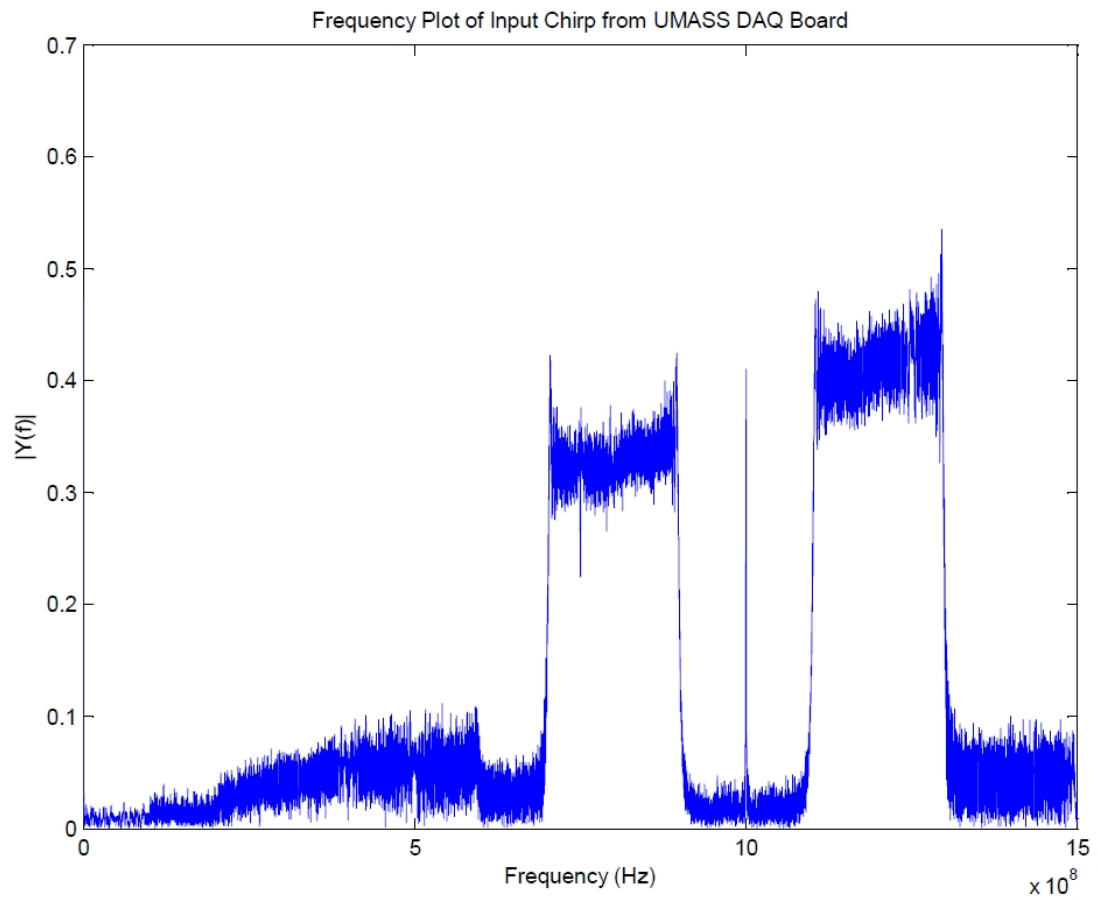
The following conclusions were drawn from the above plots.

1. The Tektronix ARB showed signs of higher order harmonics at 266 MHz, whereas the Agilent E4437B did not.
2. The true noise floor was about -14 dB.
3. The true effect of the dominant signal at 110MHz is less strong than it appears in the spectrum because the noise contributing to the SINAD and ENOB is essentially integrated over the spectrum. Hence, many of these point sources of signals will average out.

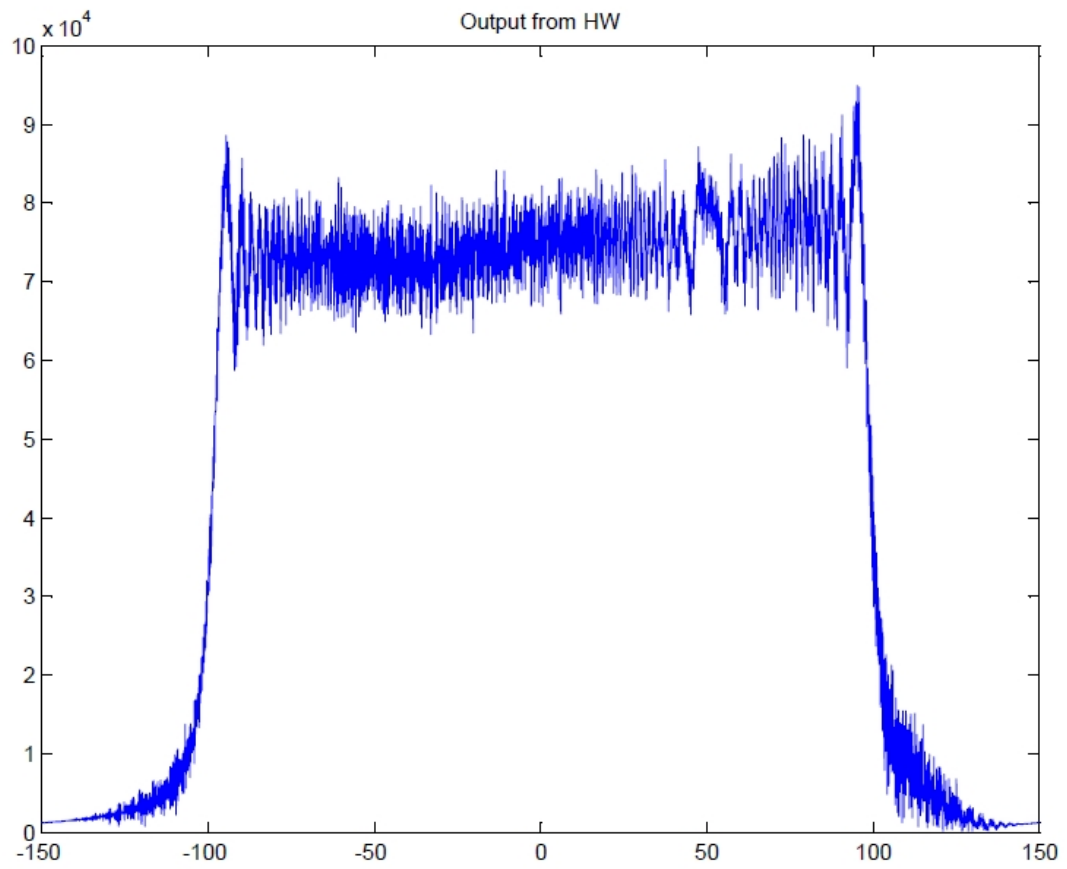
## 7.11 Filter implementation, integration and testing

A polyphase decimation filter is implemented in the Data FPGA to process the L-band signals coming into the DAS and converts them to baseband signals. Vu Duong and Mike Nakashima, engineers at JPL, NASA designed and implemented the filter. The integration of the filter with the SERDES and the FIFO is done at the MIRS Lab, UMass. The filter module is integrated with the rest of the modules in Xilinx ISE environment, and synthesized. In order to perform the initial round of system testing with the filter, a test chirp input with frequency range of 100MHz – 300MHz is mixed with a 1GHz frequency component with the help of a frequency mixer. This produces two frequency components, one centered at 800 MHz (700MHz – 900MHz) and the other centered at 1.2GHz (1.1GHz – 1.3GHz). This set up mimics the real time system where the DAS receives L-band signals centered around 1.2GHz (1.1GHz – 1.3GHz).

Figure 27 shows the spectrum plot of the up-converted input via the mixer (ADC output). X axis represents the frequency and Y axis represents the signal power. Figure 28 shows the demodulated output of the filter. The plot shows good SNR (Signal to Noise Ratio) and most importantly, there are no unwanted harmonics present. Thus, we are able to verify the working of the hardware and FPGA design (with the filter). To accurately verify filter performance, we need to input a chirp that contains actual radar pulse data and fine tune filter parameters. Detailed filter performance and validation will take place at JPL, NASA in the future.



**Figure 27. Spectrum plot of the ADC output of the up-converted input**



**Figure 28. Spectrum plot of the polyphase decimation filter**

## CHAPTER 8

### REAL TIME ESTIMATES OF DIFFERENTIAL SIGNAL PHASE

This section describes a novel approach to monitor the phase of a single signal over time and phase drifts between two associated signals using Data FPGA resources in the DAS [36].

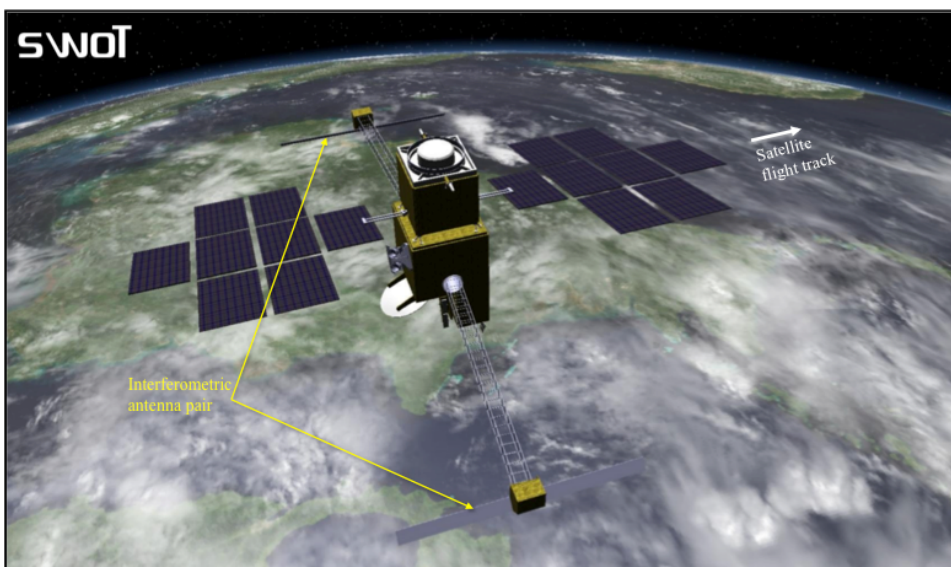
#### 8.1 Introduction

A fundamental characteristic of the microwave signals is the signal's phase, which is associated with the round-trip distance between the signal's source and destination. Signal phase measurements play an important role in fundamental satellite system operation. As a result, it is important to measure and monitor a signal's phase from a measurement point of view and in terms of its use as a characterization of the overall system's performance [38].

For the SWOT mission, a common signal is reflected from the Earth's surface and received by two antenna panels (Figure 29) co-mounted on a satellite that are separated by a fixed distance called the baseline (nominally 10m). The differential phase of the received signal by the two ends of the baseline is used to infer the angle of arrival of the reflected signal. Since the viewing geometry is known, it is possible to determine the topography of the ocean and inland waters to a high degree of accuracy [37]. The precision to which this phase difference can be measured directly affects the intended science product of height. Hence, an improved phase measurement capability can be used to improve overall system performance, or can be traded against other system parameters such as the baseline length, to reduce the size of the spacecraft structure and dramatically impact the overall system cost.



By measuring and monitoring the differential phase of a known signal fed into the two analog-to-digital converters on the DAS, a low-level check can be performed on the science data integrity on a pulse-to-pulse basis. The results are subsequently passed down and incorporated into a standard data telemetry data stream. On-board FPGA monitoring of these results can be used as a "watch dog" to flag problematic data and to record intermediate products that can be saved and sent to the ground for further analysis.



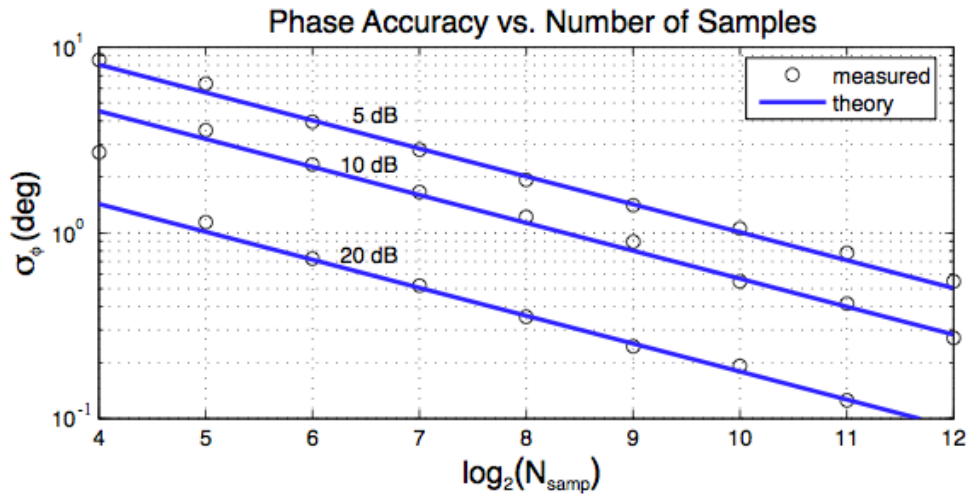
**Figure 29. Artistic concept of the SWOT satellite mission.**

## **8.2 Background Work**

An algorithm for monitoring system phase has previously been implemented and tested in software [38]. The most basic form of the algorithm is based on the monitoring of a single-tone signal fed into an analog-to-digital converter. The resulting waveform is compared to a reference waveform of the expected frequency and phase. Differences between the observed waveform and the reference waveform are used to determine the gain and phase characteristics of the observed waveform's signal path. The accuracy of the gain and phase characteristics is dictated by the number of data points and the signal

to noise ratio (SNR) of the observed signal. Increasing either the SNR or the number of points increases the accuracy of these estimates in a mathematically predictable way, and the estimation of these numbers can be optimized to achieve a desired level of measurement accuracy.

Figure 30 illustrates measured and theoretic phase accuracy versus sample count. The theoretical phase measurement accuracy (standard deviation in degrees) versus the number of samples is shown on a log scale (on this scale, the number 10 represents  $2^{10}$  samples, or 1024 samples). Simulations using Matlab were performed for varying signal to noise ratios (5 dB, 10 dB, and 20 dB) and for varying numbers of samples. An excellent agreement between the theory and simulations can be seen. Similar results have also been demonstrated with measured data in the laboratory.

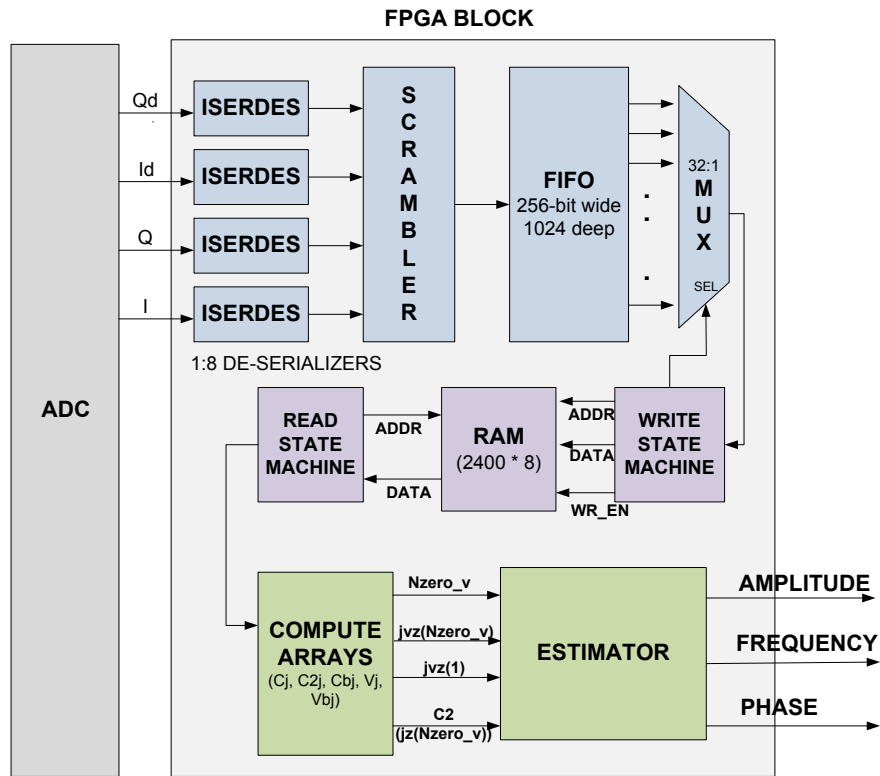


**Figure 30. Measured and theoretical phase accuracy versus the number of samples.**

### 8.3 FPGA Implementation and Testing

The implementation of phase testing in the FPGA takes place as a sequence of steps performed on stored data. Operations take place in the following sequence:

a. Storing M consecutive samples in a FIFO - As shown in Figure 31, a series of consecutive data samples are stored in a FIFO after collection from an ADC. A subset of M samples are taken from the FIFO and stored in the RAM bank shown in Figure 31. As values are collected from the FIFO they are converted from 8-bit signed magnitude representation to two's complement representation by the subtraction of 127. The array values stored in RAM (called  $v[]$ ) are located in adjacent memory locations. The size of the array (M) should be greater than 1000 and should cover multiple cycles of the input sinusoidal waveform.



**Figure 31. Block diagram of the hardware for phase detection in Data FPGA**

b. Determine and count signal zero value transitions – A total of M data values ( $v[]$ ) are fetched one at a time from consecutive RAM locations starting with the first value in an

array of M values. Each sequence of three consecutive  $v[]$  values are then averaged using a three-point boxcar filter to create a new sequence of values.

$$vb[j] = \frac{vb[j-1] + vb[j] + vb[j+1]}{3}$$

The generated  $vb[]$  values are used to determine positive-to-negative and negative-to-positive transitions in the sinusoidal data stream. The number of these zero transitions ( $Nzero\_v$ ) in the  $vb[]$  array is determined along with the index in  $vb[]$  where the first ( $javz[1]$ ) and last ( $javz[Nzero\_v]$ ) zero transition in the data stream occurs.

c. Determine a running sum of the square of the signal magnitudes - Data values ( $v[]$ ) are also used to calculate a running sum of the square of the magnitude of all points such that

$$c2[j] = \sum_{i=0}^j (v[i])^2$$

The intermediate values  $vb[]$ ,  $Nzero\_v$ ,  $javz[Nzero\_v]$ , and  $c2[]$  are calculated in parallel in the FPGA. These intermediate values are used to calculate estimated amplitude, frequency, and the phase of the first sample in the  $v[]$  array.

d. Estimated amplitude calculation - In terms of the values noted above, the amplitude of a signal can be estimated by the following equation.

$$A = \sqrt{\frac{2}{Nzero\_v}} \times \sqrt{c2[javz[Nzero\_v]]} \quad (1)$$

The numerator in the expression includes the sum of squared magnitudes of all points up to the index of the last data zero transition. The denominator includes the number of zero transitions of the sinusoidal signal. A square root is taken of the resulting value to determine the RMS amplitude value.

e. Estimated frequency calculation - The estimated frequency of the signal is determined using the following formula:

$$freq = \frac{(Nzero\_v - 1)}{jvz[Nzero\_v - 1] - jvz[1]} \times (F_{SAMP}) \quad (2)$$

where  $F_{SAMP}$  is the system sampling frequency of the analog-to-digital converters.

f. Calculation of corrected initial phase - The goal of this calculation is to estimate the phase of the initial sample in the array sequence. The calculation of this initial phase takes place in a series of steps. The calculation of initial phase is represented by the following equation:

$$phii = (2\pi \times (Nzero\_v)) + \left(\frac{\pi}{2}\right) - \left(2\pi \times \frac{freq}{F_{SAMP}} \times jvz[Nzero\_v - 1]\right) \quad (3)$$

This initial phase is then used in a series of iterations to minimize the error involved in the calculations. The equations associated with this analysis are:

$$phic = phii - \frac{\sum_{m=1}^M (\sin(\theta_m) \times (v[m] - A \cos(\theta_m)))}{A \sum_{m=1}^M (\sin(\theta_m))^2} \quad (4)$$

such that

$$w = 2\pi \times freq \quad (5)$$

$$dt = \frac{1}{F_{SAMP}} \quad (6)$$

$$\theta_m = w \times dt \times (m - 1) + phii \quad (7)$$

In the first iteration, the  $phii$  value determined in Equation (3) is used as the input to Equations (4) and (7). In subsequent iterations, the  $phic$  value found in the immediately previous iteration is used as the equation input in the  $phii$  locations. In Equation (4), the corrected initial phase is effectively refined by evaluating estimated phase values across all  $M$  data points in  $v[]$ . A total of 3 iterations were used to estimate phase values in our experimentation.

To evaluate phase drift, a series of corrected initial phase values ( $phic$ ) can be determined for a series of  $M$  point data blocks sampled from a single input sinusoidal

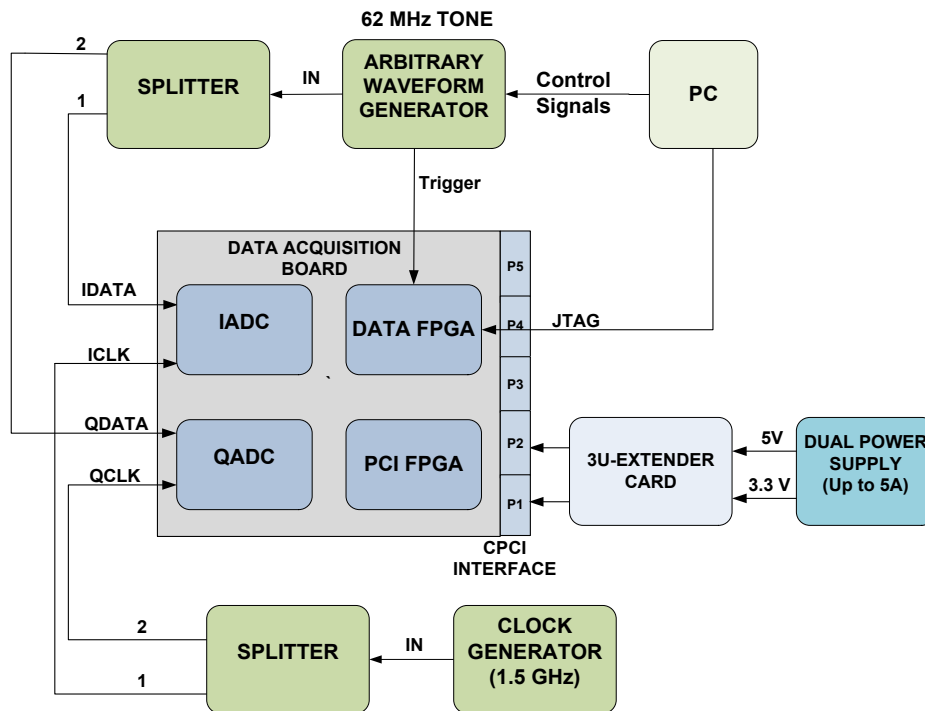
signal. These values can then be compared to expected phase values to determine if a measurable phase drift can be determined for the sequence.

#### **8.4 Experimental approach**

The phase detection circuit has been implemented in the lab using the experimental setup shown in Figure 32. An arbitrary waveform generator is used to create a 62 MHz signal which is used as a reference signal. The signal is sent through a splitter to generate two equivalent signals. These signals are then input into a state-of-the-art FPGA-based data acquisition system.

Following steps were followed to collect data for our experiments

- The 1.5 GHz clock is enabled to start ADC sampling
- The 62 MHz input signal is enabled
- A start trigger is sent from the signal generator to the FPGA.
- This trigger enables the FIFOs to store state.
- Processing starts once the input FIFO is full.



**Figure 32. Block diagram of the experimental setup for the phase detection circuit**

Inside the Data FPGA, a 256x1024 FIFO is used to hold a total of 32,768 data samples. There are two state machines employed in the FPGA. The write state machine reads the values from the FIFO, splits it into 32 8-bit samples and writes to successive locations of a RAM. The read state machine reads data at appropriate addresses of RAM and passes the values to a compute stage.

## 8.5 Results

Two sets of experiments were performed using the experimental setup described in Section 8.4. First, the corrected initial phase (*phic*) of each individual channel was determined for a series of  $M$  point blocks. The actual initial phase (the phase of the first sample of the first sequence) was then determined using Matlab for comparison. In a second set of experiments, the corrected initial phases for two channels were compared to

each other for a series of data point blocks. These results were also compared to Matlab calculated results.

The hardware implementation of Equations (1-7) includes a mix of fixed and floating point formats. Values  $N_{zero\_v}$ ,  $v[]$ ,  $vb[]$ , and  $juvz[]$  are represented using 8-bit 2's complement values. Values  $c2[]$  are represented using 26-bit 2's complement values. The amplitude calculation in Equation (1) requires the conversion of  $c2[]$  and  $N_{zero\_v}$  values to floating point. This conversion is performed in the FPGA hardware. Xilinx floating point multiply, divide, and square root operations are performed on 32-bit floating point values. An amplitude result (Equation (1)) is generated as a 32-bit integer. The frequency calculation (Equation (2)) and corrected initial phase calculation (Equations (3-7)) similarly require the conversion of input values to floating point and the use of a floating point divider. These calculations also result in 32-bit integer results. The corrected initial phase ( $phic$ ) calculations additionally require the use of 32-bit Xilinx Cordic blocks to perform sin and cos operations [40]. In the Virtex 4 architecture, each RAM block contains 18 Kbits of data and multipliers perform 18x18 2's complement operations.

For our initial single channel experiments, a 600 mV peak-to-peak signal of 62 MHz was fed to the I channel ADC (IADC) and a total of 32,768 8-bit consecutive samples were stored in the FIFO shown in Figure 31. These samples were then considered as a series of consecutive  $M = 2400$  sample blocks. The phase of the first sample of the first sequence was calculated from the samples in post processing (using algorithms written in Matlab) for reference.



For the first 2400 value sample block (block 1), the comparison of the calculated  $phic_1$  and the post-processed value is straightforward. After calculation using Equations (3-7), the comparison can be made directly. For subsequent sample blocks, the calculated values using Equation (4) determine the phase of that block's (i.e. block m) sequence,  $phic_m$ . To adjust this value to allow comparison to the phase of the first sequence, a phase equivalent to the time delay of 2400 samples at a sample rate of 3 GSamp/sec was subtracted from the current sample block.

$$phic_1 = phic_m - 360 \times \Delta T \times (62MHz) \quad (8)$$

This approach can be extended to monitor/estimate the initial phase of the signal in real time for several seconds/minutes. The incoming data stream is collected in the FIFO continuously for the entire receive window duration and the algorithm is run to estimate the corrected initial phase of the signal at a variety of time points, indicating the possible presence of phase drift

Table 14 shows the FPGA algorithm's estimated phase for consecutive blocks of 2400 samples each measured at a sample rate of 3 GSamp/sec (Equations (3-7)) for the I-channel ADC being read into the FPGA. Three iterations of  $phic$  calculation using Equation (4) were necessary to converge to the result shown in Table 14. The signal's true starting phase (I channel - 55.03 deg, Q channel - 45.43 deg) has been determined in post-processing, and is accurate to within 0.01 degrees. The starting phase itself, while somewhat arbitrary, should stay constant for each subsequent block. The FPGA estimated phases,  $phic$ , are within 3.1 degrees of the true value and better than 1% accurate overall (out of 360 degrees of phase), thus validating our approach. The small remaining error is expected to be due to round off error in the data processing and we are

currently investigating how to make the measurement accurate to within its theoretical bounds of 0.01 degrees.

A similar experiment was performed for a 62 MHz input signal for the Q channel ADC (QADC). Table 15 shows the phase of the signal monitored for 7 successive sample blocks in the FIFO. For both the results in Table 14 and 15, the clock speed of the FPGA circuitry was 60 MHz. A total of 0.37 ms was needed to determine  $phii$  and 10.3 ms was required to determine  $phic$ . These execution times was determined starting from the initial sample read from RAM.

The second experiment used to validate the corrected initial phase estimator involved a comparison of initial phase for signals arriving simultaneously on both I and Q channels. In this experiment,  $phic$  is determined for both channels and subtracted to determine the phase difference. As shown in Figure 32, a single signal from the arbitrary waveform generator is split into I and Q channels. Since equal length 2m cables were used for both channels, minor phase differences were expected. While there are various reasons that the phase might be different for the two channels (e.g. timing differences between the two ADCs) the phase difference between the two channels as a function of time, if performed accurately, shows path-integrated temperature differences between the two channels that feed into the ADCs.

Sample block	Estimated phase, $phic$	Estimated error	
1	58.1	-3.0 deg	-0.8 %
2	53.1	+2.0 deg	+0.5 %
3	53.0	+2.0 deg	+0.6 %
4	55.7	-0.7 deg	-0.2 %
5	55.5	-0.4 deg	-0.1 %
6	58.1	-3.1 deg	-0.9 %
7	53.1	+1.9 deg	+0.5 %

**Table 14. Corrected initial phase estimation ( $phic$ ) for IADC versus actual phase**

Sample block	Estimated phase, phic	Estimated error	
1	47.73	+2.3 deg	+0.7 %
2	47.83	+2.4 deg	+0.7 %
3	45.23	-0.2 deg	-0.0 %
4	42.63	-2.8 deg	-0.8 %
5	42.83	-2.6 deg	-0.7 %
6	47.73	+2.3 deg	+0.7 %
7	47.83	+2.4 deg	+0.7 %

**Table 15. Corrected initial phase estimation for QADC versus actual phase**

Actual (determined from post-processing) and FPGA estimated phases for a split 62MHz signal input into both I and Q ADC channels are shown in Table 16. A slight error is seen in the phase estimation due to round off errors and sampling accuracy, which at times manifests itself as a seven degree error when an entire sample is dropped (see blocks 4 and 5). With the exception of the errors induced from a dropped sample, it can be seen that the phase difference measures between the post-processed and FPGA processed data is consistent to within  $0.03^\circ$ , an indication that the FPGA phase measurement errors are systematic (and not random), and therefore cancel one another when calculating phase differences.

Sample block	True (phic <sub>i</sub> -phic <sub>o</sub> )	FPGA (phic <sub>i</sub> -phic <sub>o</sub> )	Difference Error	Difference Error
1	29.22°	29.78°	0.56°	0.2%
2	29.22°	29.77°	0.55°	0.2%
3	29.20°	29.74°	0.54°	0.2%
4	29.23°	22.18°	-7.05°	1.9%
5	29.20°	22.28°	-6.92°	1.9%
6	29.18°	29.78°	0.60°	0.2%
7	29.19°	29.77°	0.58°	0.2%
8	29.18°	29.74°	0.55°	0.2%

**Table 16. True (Matlab) versus FPGA measured phase differences**

A final experiment was performed to evaluate an estimate of the signal frequency using Equation (2). Using a 2400 sample block, an estimate of 61.9964 MHz was determined for the 62.0000 MHz signal.

## CHAPTER 9

### CONCLUSIONS AND FUTURE WORK

#### 9.1 Conclusions

A prototype high speed DAS has been designed and implemented to support NASA's SWOT mission. The PCB design phase concentrated on every minute design aspect with careful attention to schematics, and layout of the board. The board has been fabricated after several rigorous rounds of design review. The prototype hardware system has been implemented with a form factor same as that of a standard 6U cPCI form factor (160mm X 233mm) which makes the board pluggable to the backplane of the cPCI slot. The hardware system is shown in the Figure 33. The system has been verified for both FPGA functionalities and the ADC functionalities using the customized test set up in laboratory conditions. Specialized waveform generators have been used to provide high-speed clocks and inputs to the system and the results show that the data acquisition system has been performing up to the mark in terms of performance. The digital polyphase filter designed by Mike has been successfully integrated with the rest of the modules and tested. In addition to developing the DAS, an elegant method is described to monitor the phase of the signal and phase difference between two associated signals using Data FPGA resources[36]. Results indicate that phase calculations with an error of less than 2% are achieved using the adaptive approach[36].

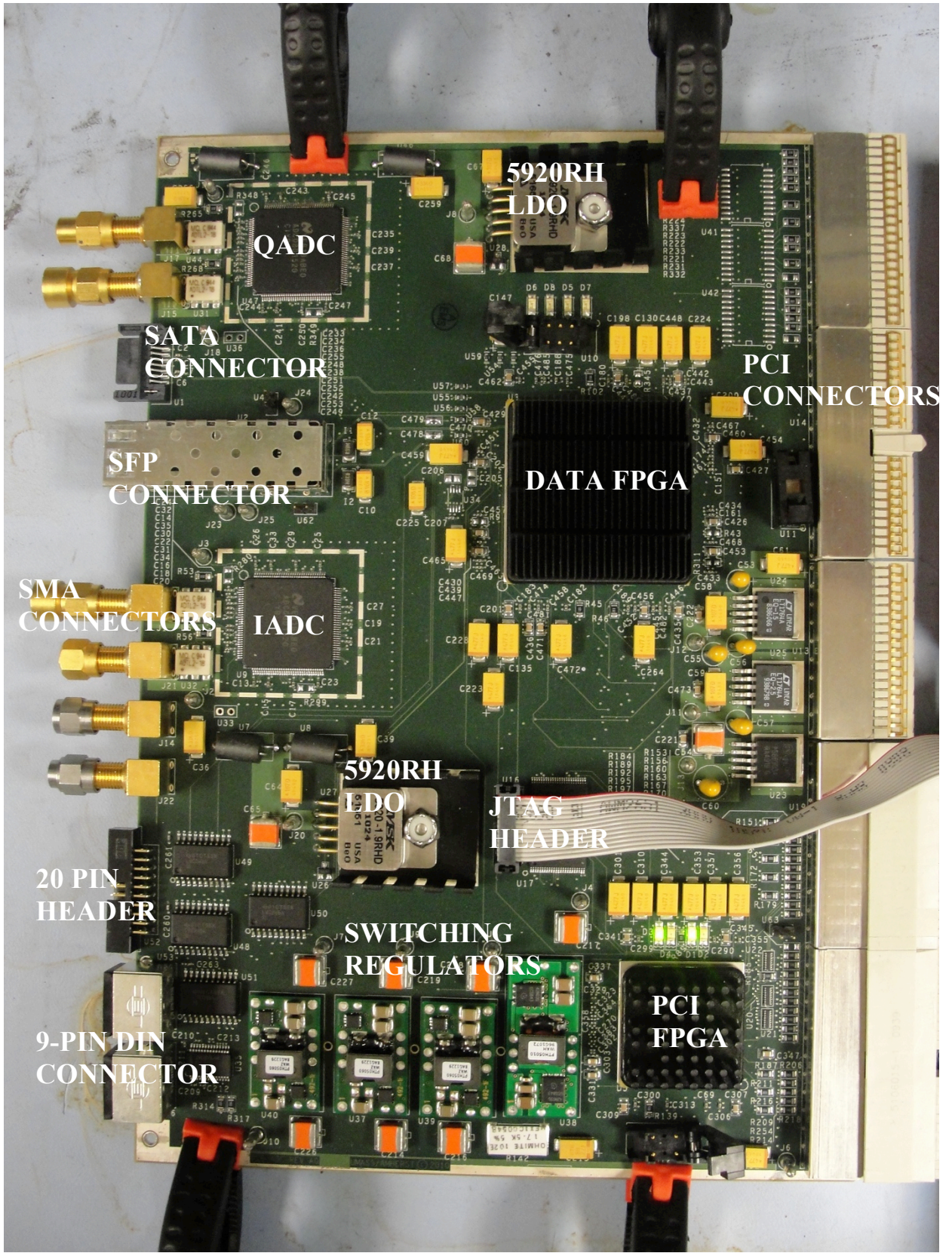


Figure 33. Hardware PCB board

## 9.2. Future Work

This section provides an overview of the future work before the integration of the system with the down conversion circuitry and field-testing.

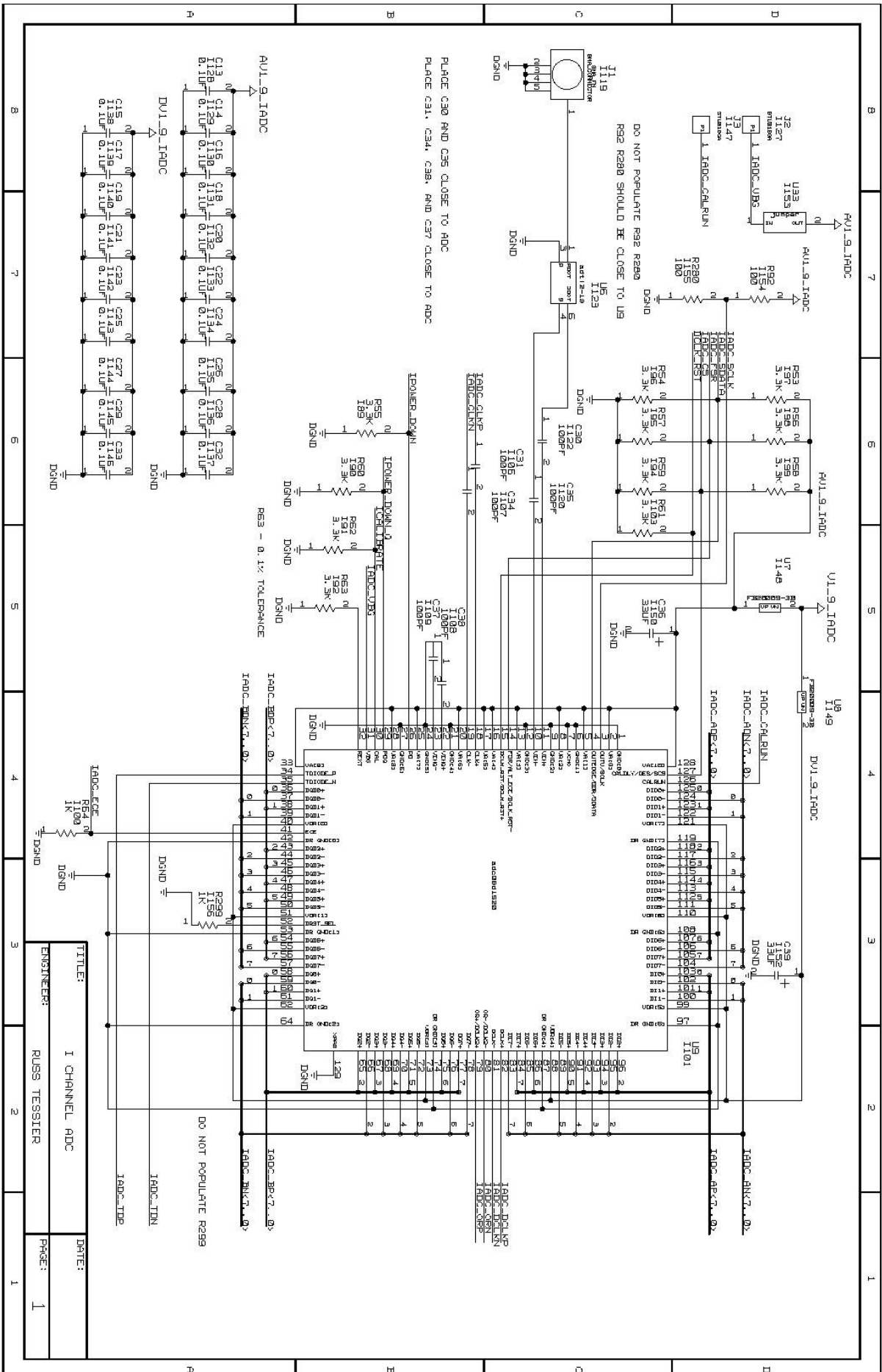
- a. The RocketIO MGTs in the Data PGA need to be tested and verified. A test design needs to be created in Xilinx ISE platform involving the two MGTs. A test methodology must be developed to perform testing at high speeds.
- b. Similarly, the SATA and SFP interfaces must be tested and verified. Necessary drivers needed for SATA and SFP interfaces need to be installed in the Data FPGA. Test design and a test platform must be created facilitating the testing of these interfaces.
- c. A state machine is implemented in PCI FPGA that configures the Data FPGA on start up. This is necessary since the configuration file of the Data FPGA is too large to be stored in any external non-volatile memory. An initial design is developed and simulated at UMass and sent to JPL, NASA for further integration and testing with the real hardware.
- d. Finally after the completion of the above tasks, the non Rad-Hard components on board must be replaced with Rad-Hard equivalents to allow the system to be housed in a satellite.

## **APPENDIX A**

### **SCHEMATICS**

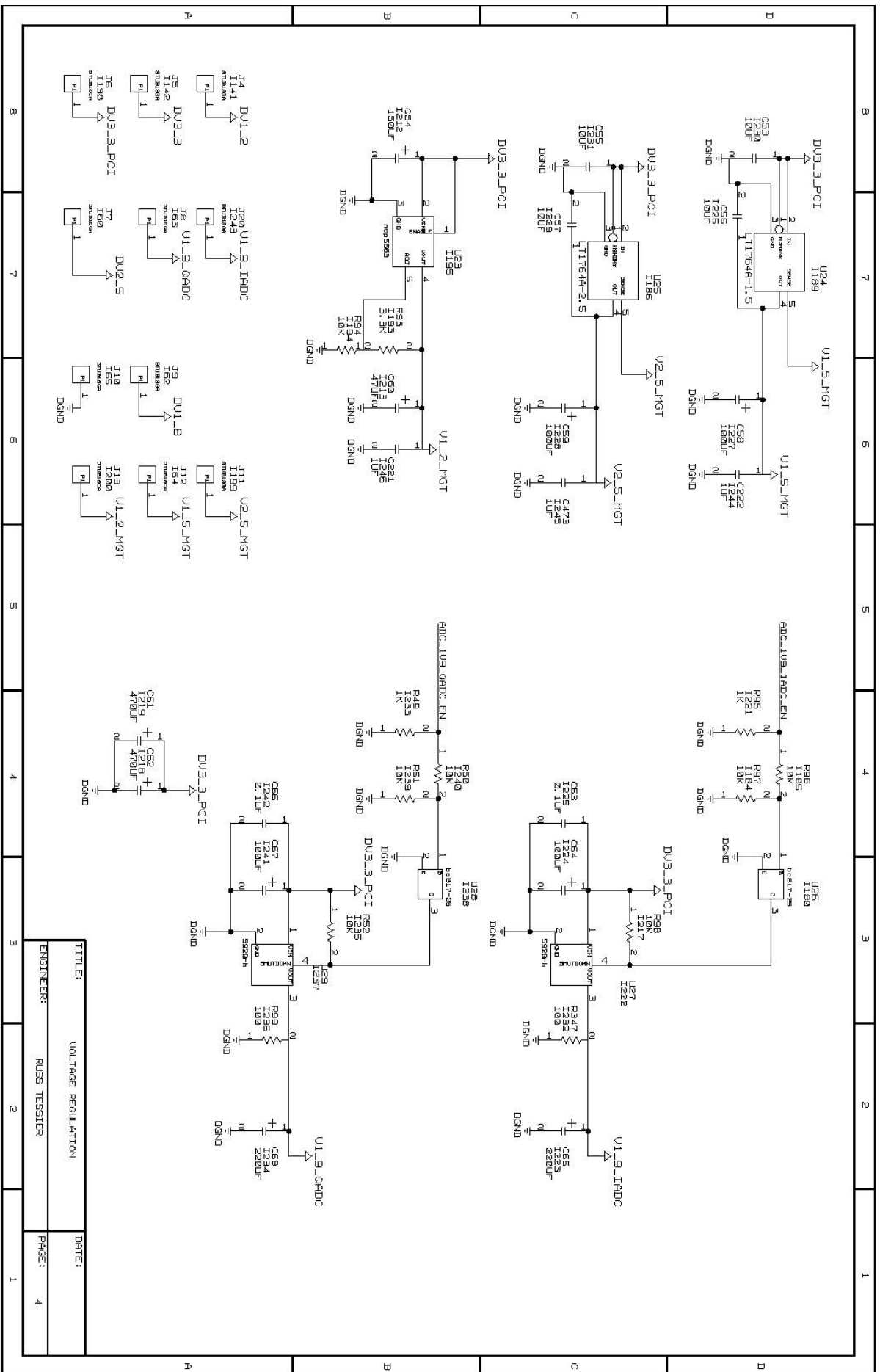
The Schematics for the Data Acquisition Board is given in Appendix A. It spans from page 77 to page 103.











TITLE: VOLTAGE REGULATION  
 ENGINEER: RUSS TESTER  
 DATE: \_\_\_\_\_  
 PAGE: 4

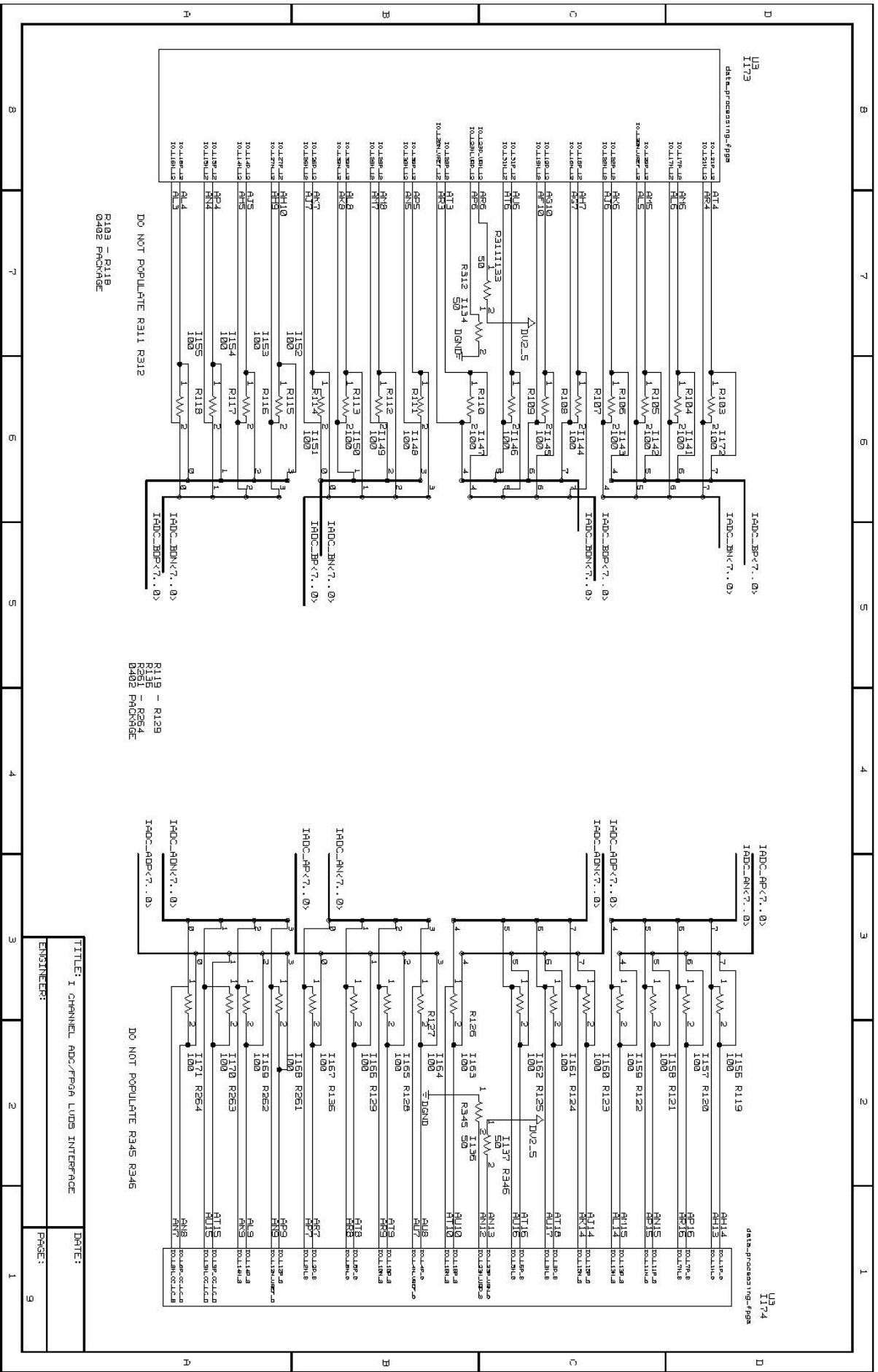


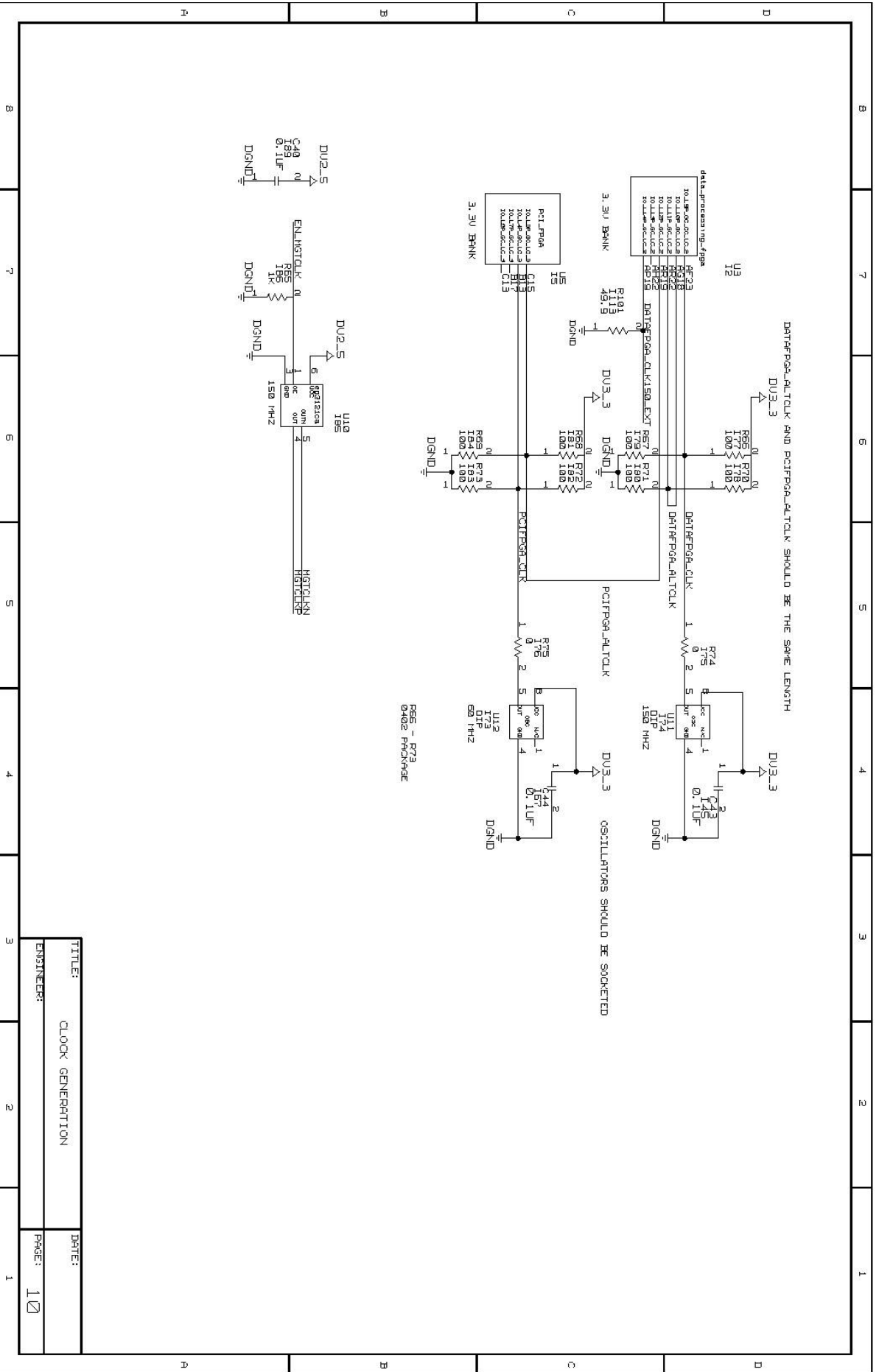




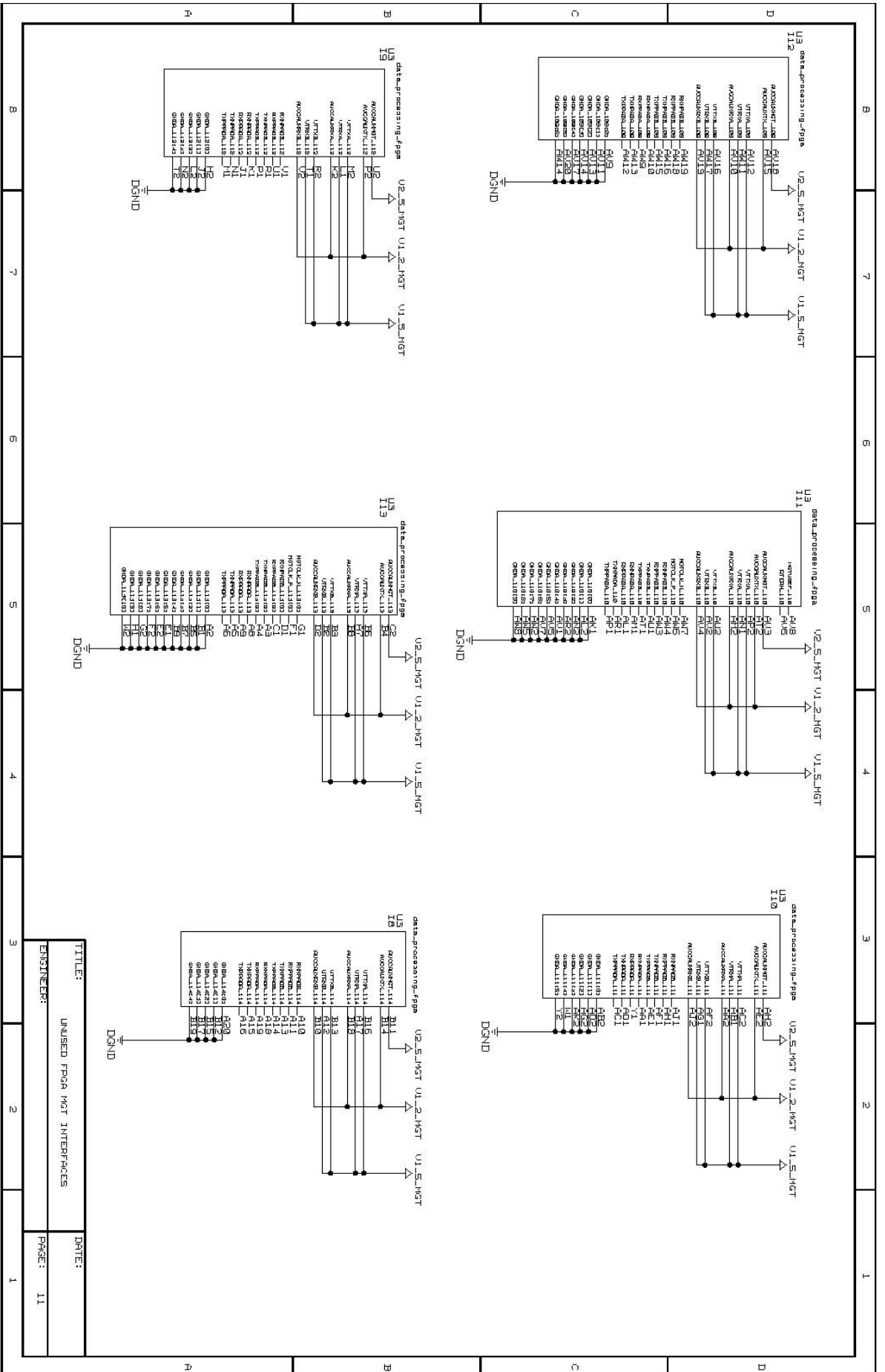






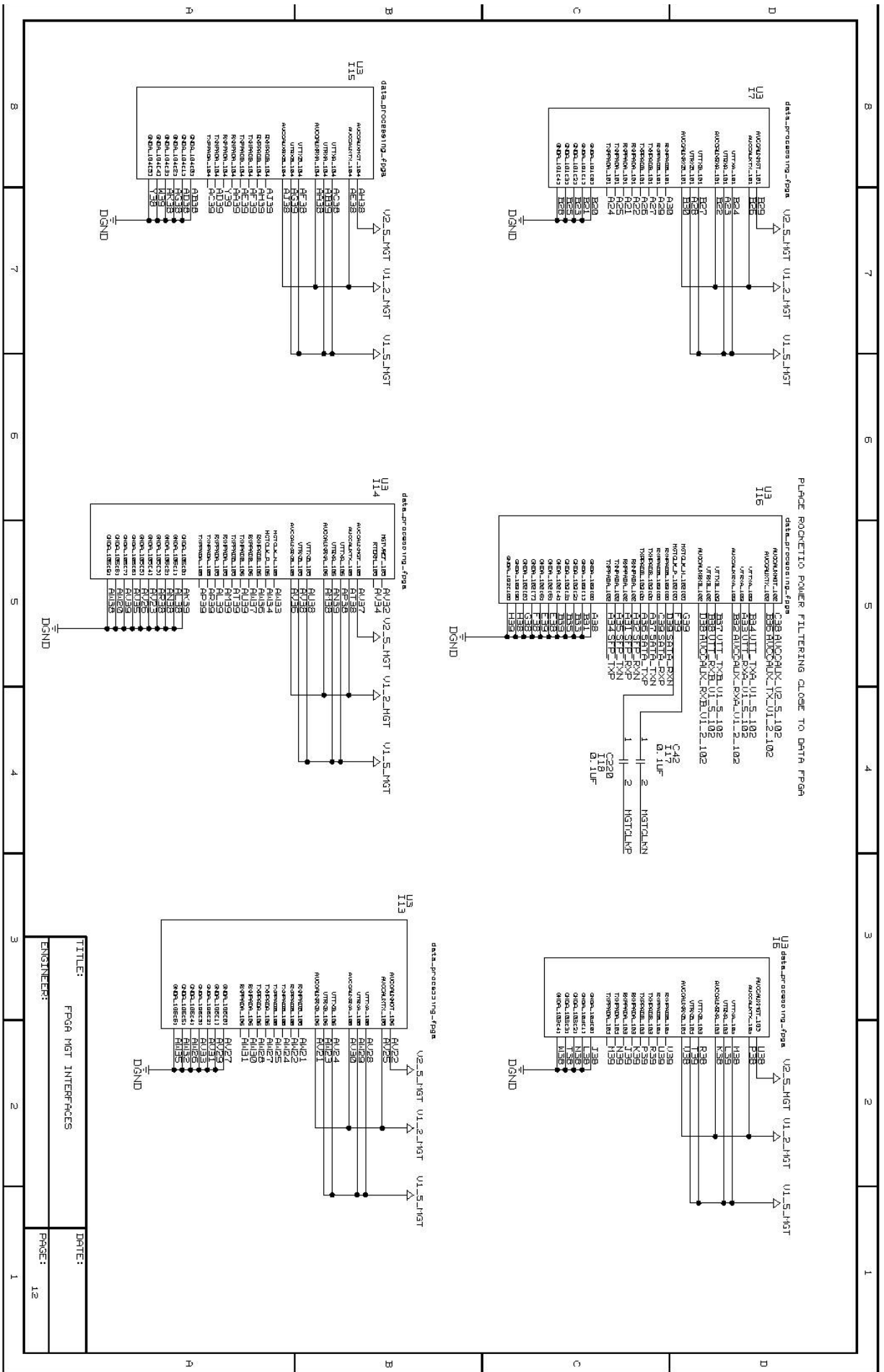


TITLE:	CLOCK GENERATION	DATE:	
ENGINEER:		PAGE:	1/1

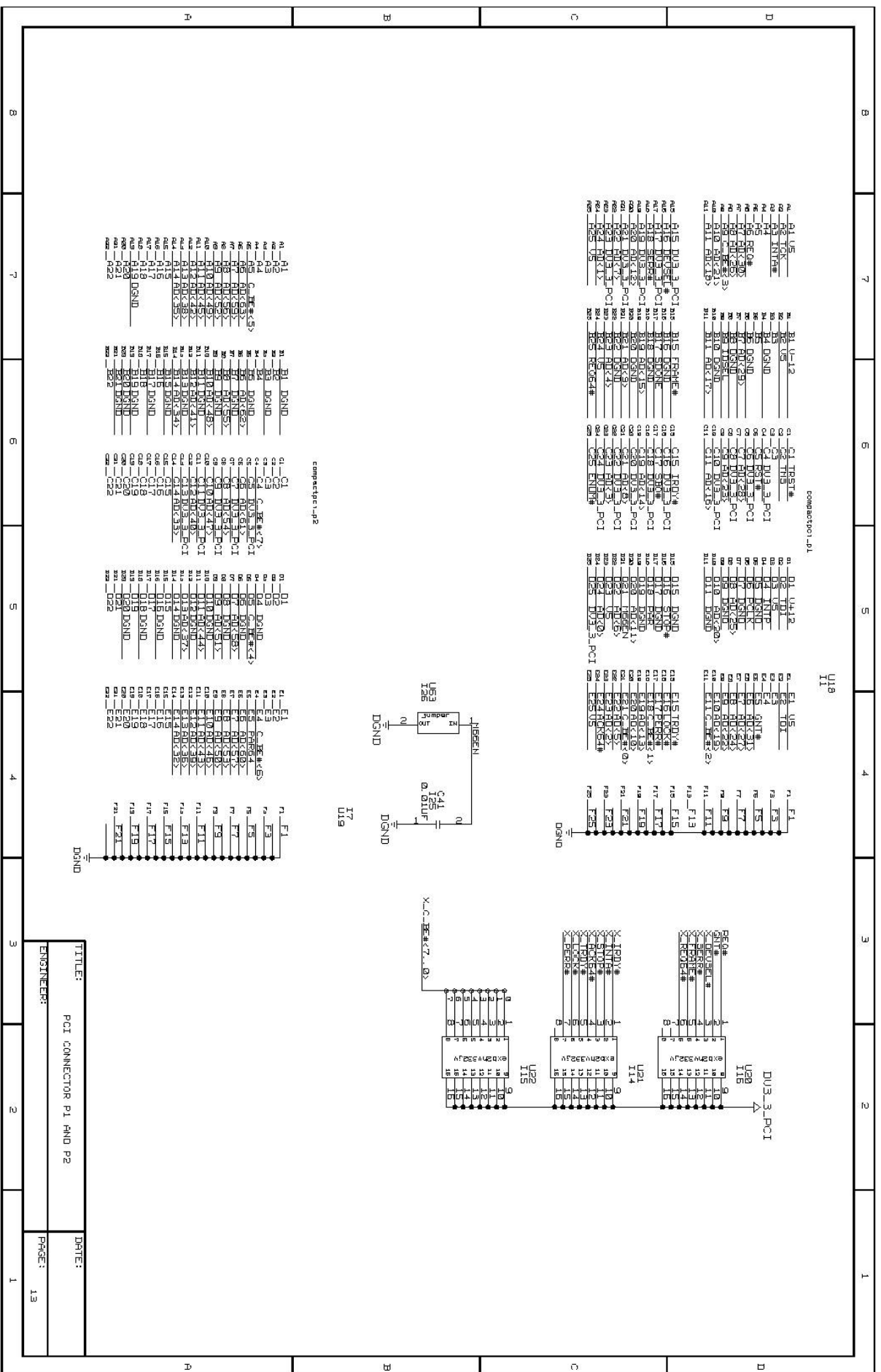


TITLE: UNUSED FPGA MGT INTERFACES

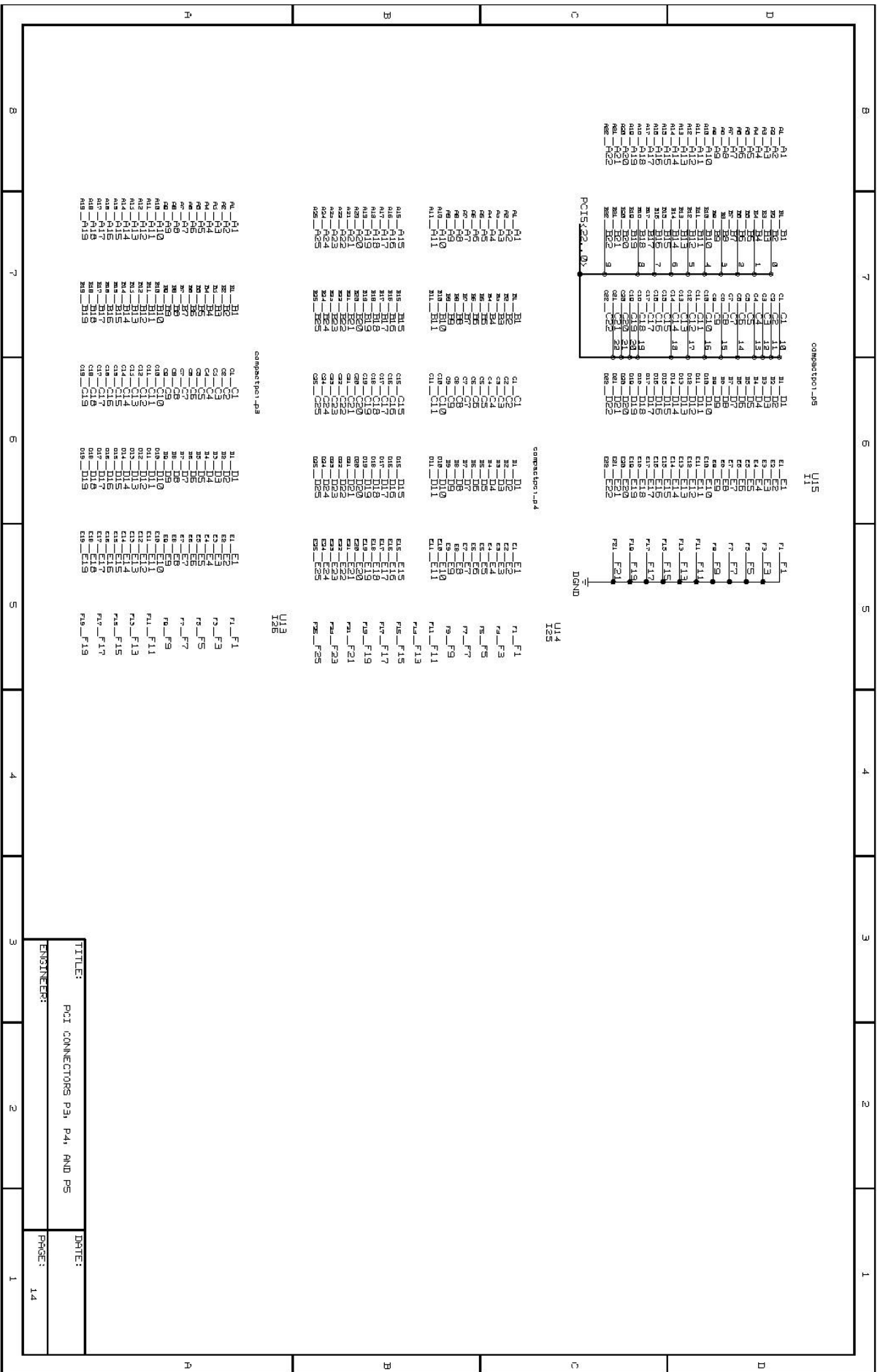
DATE:   
 ENGINEER:   
 PAGE: 11



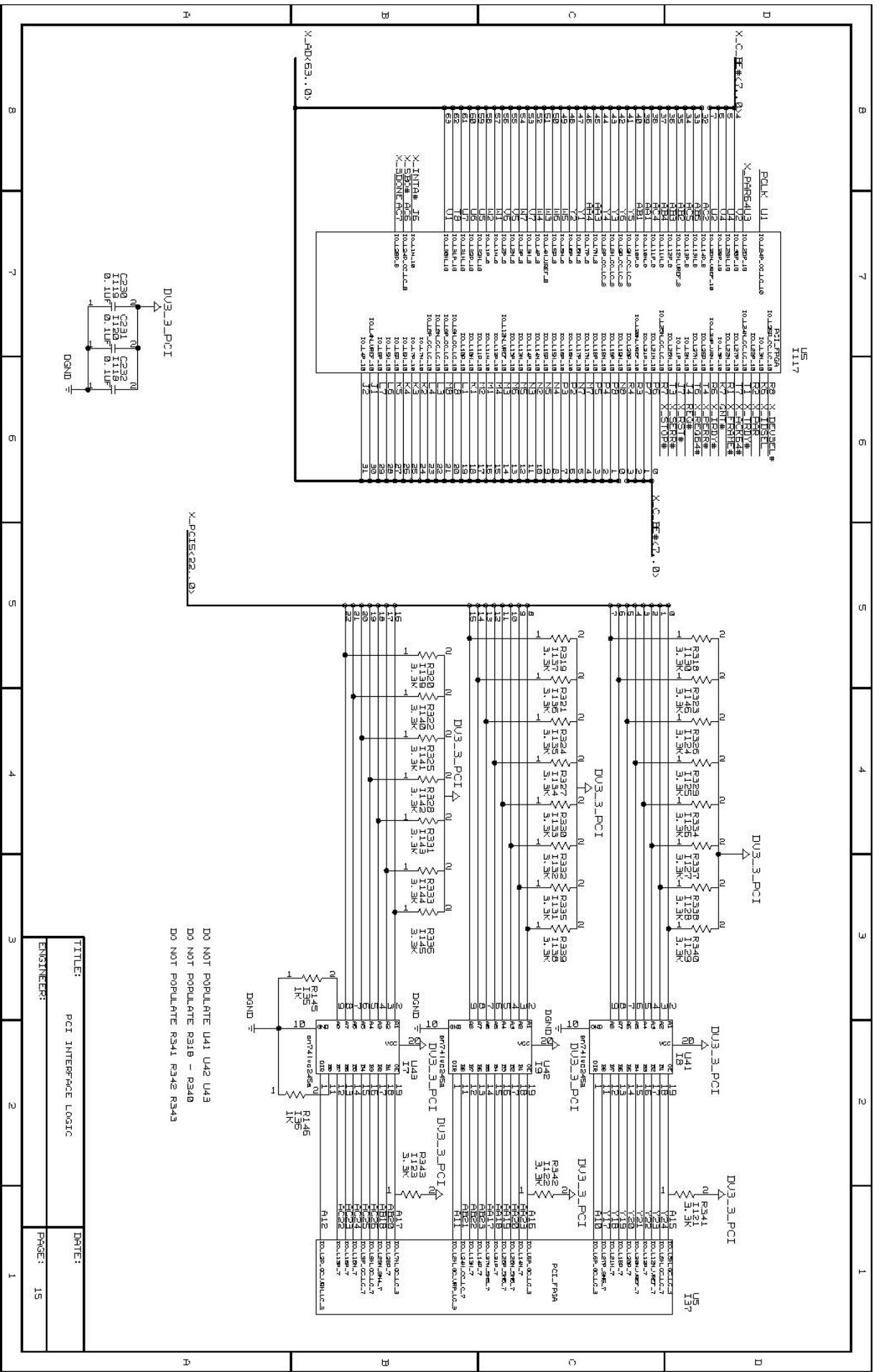
TITLE: FPGA MGT INTERFACES  
 ENGINEER: DATE:  
 PAGE: 12



TITLE: PCI CONNECTOR P1 AND P2  
 ENGINEER: [Blank]  
 DATE: [Blank]  
 PHASE: 13



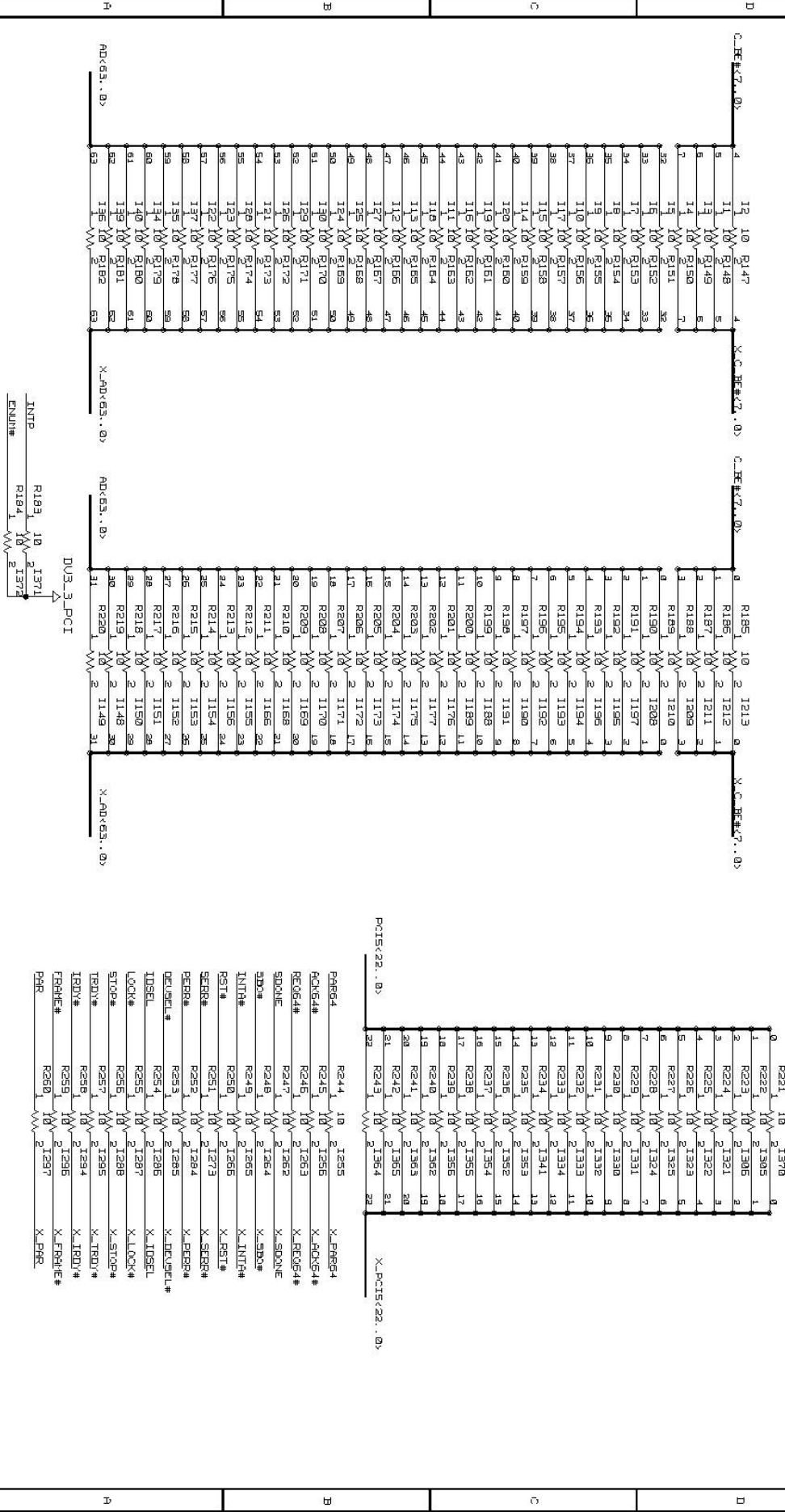
TITLE: PCI CONNECTORS P3, P4, AND P5  
 ENGINEER: DATE:  
 PRGE: 14



DO NOT POPULATE U41 U42 U43  
 DO NOT POPULATE R319 - R340  
 DO NOT POPULATE R341 R342 R343

TITLE:	PCI INTERFACE LOGIC
ENGINEER:	
DATE:	
PAGE:	15

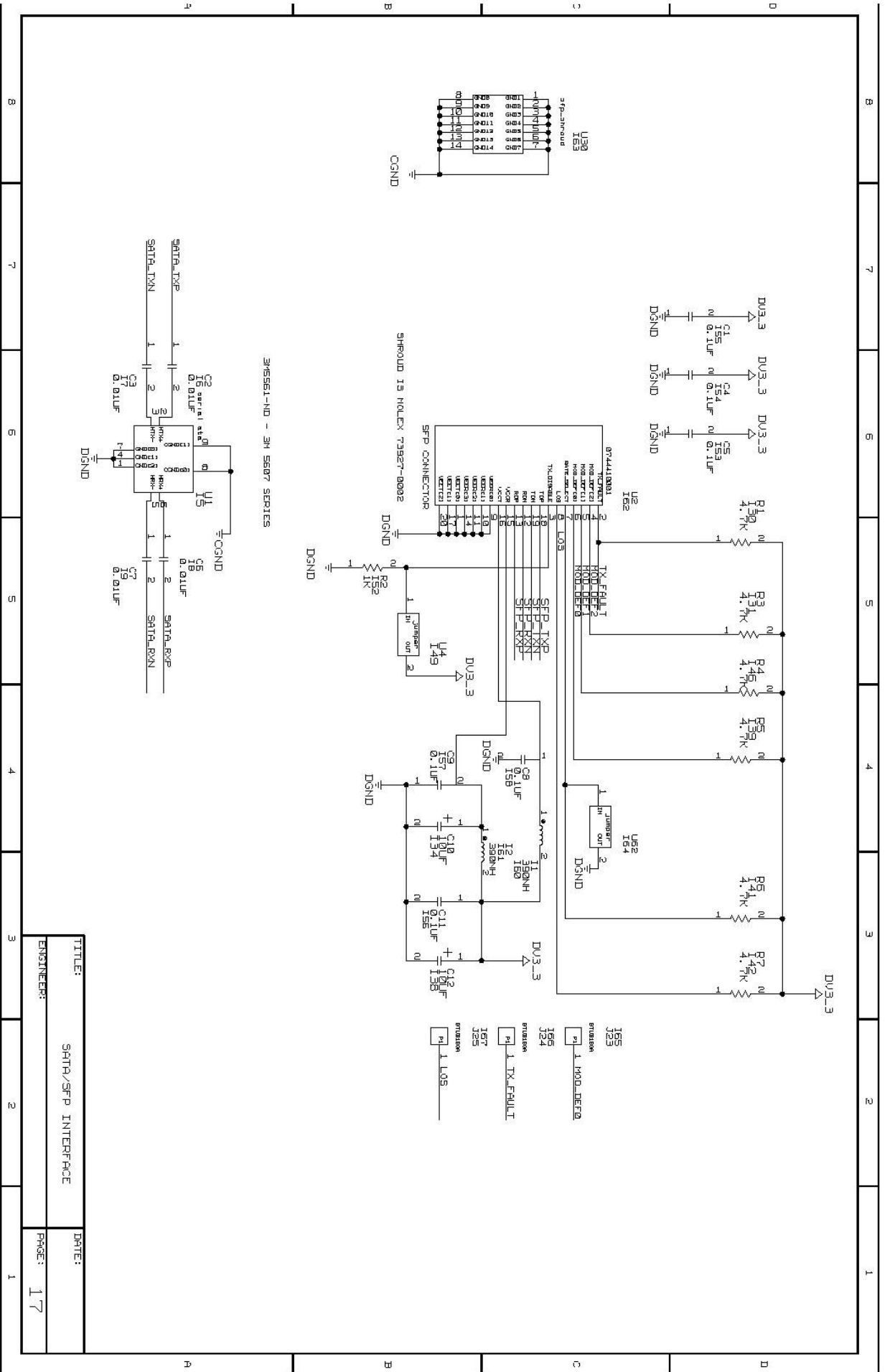
PLACE STUB RESISTORS NEAR PCI CONNECTORS



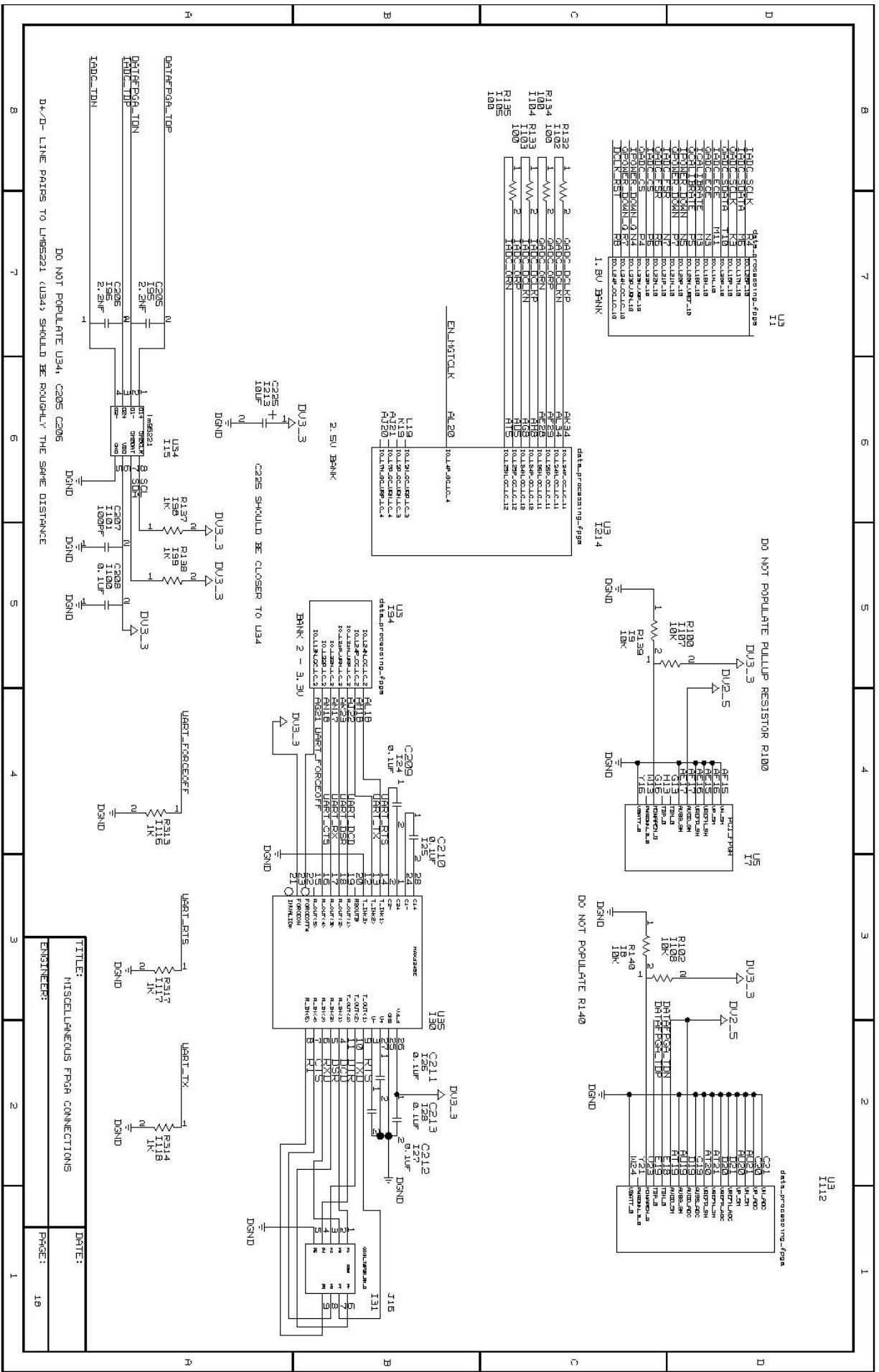
DO NOT POPULATE R183 R184

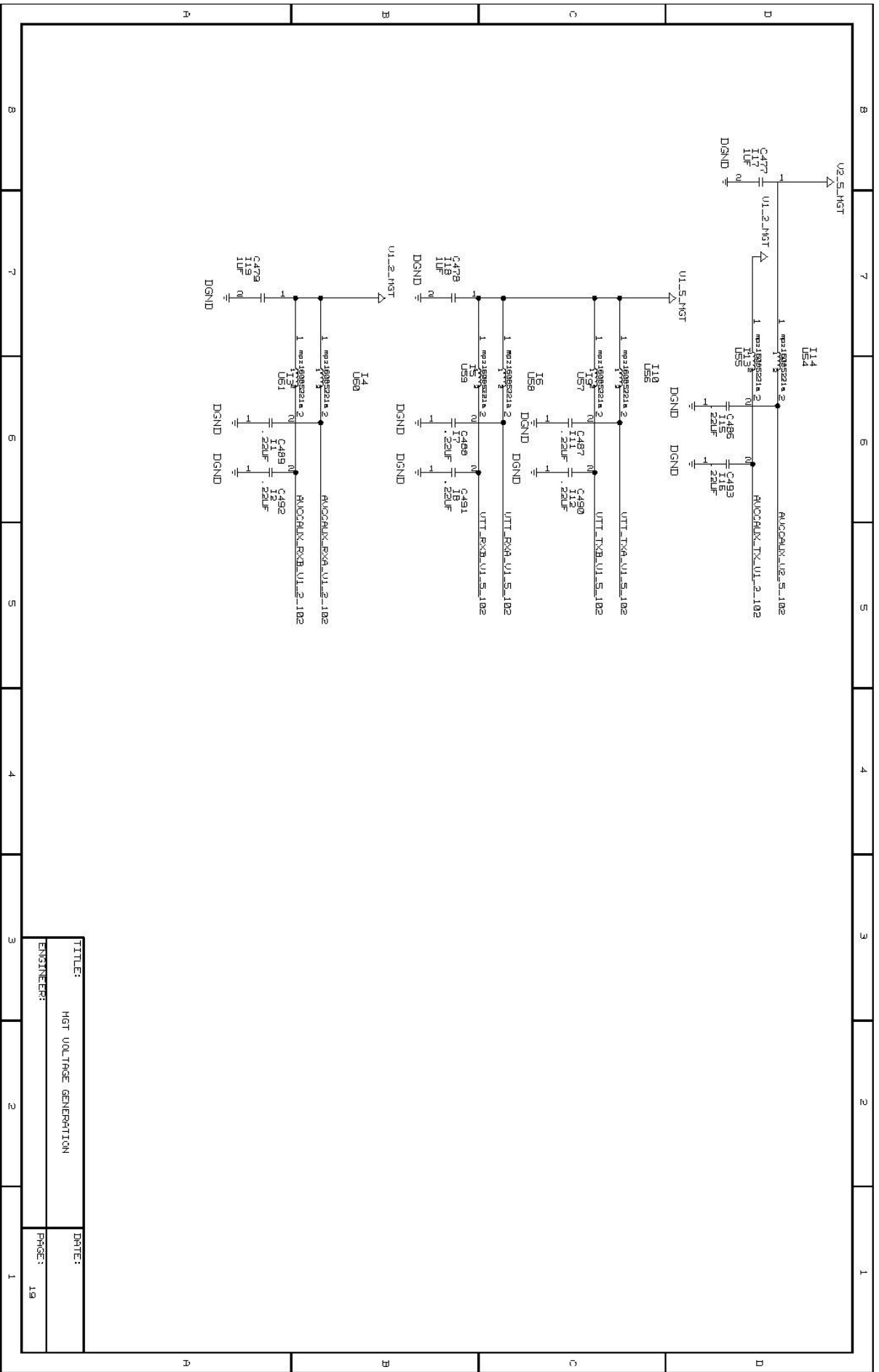
TITLE: PCI STUB RESISTORS DATE: 16  
 ENGINEER: PAGE: 16



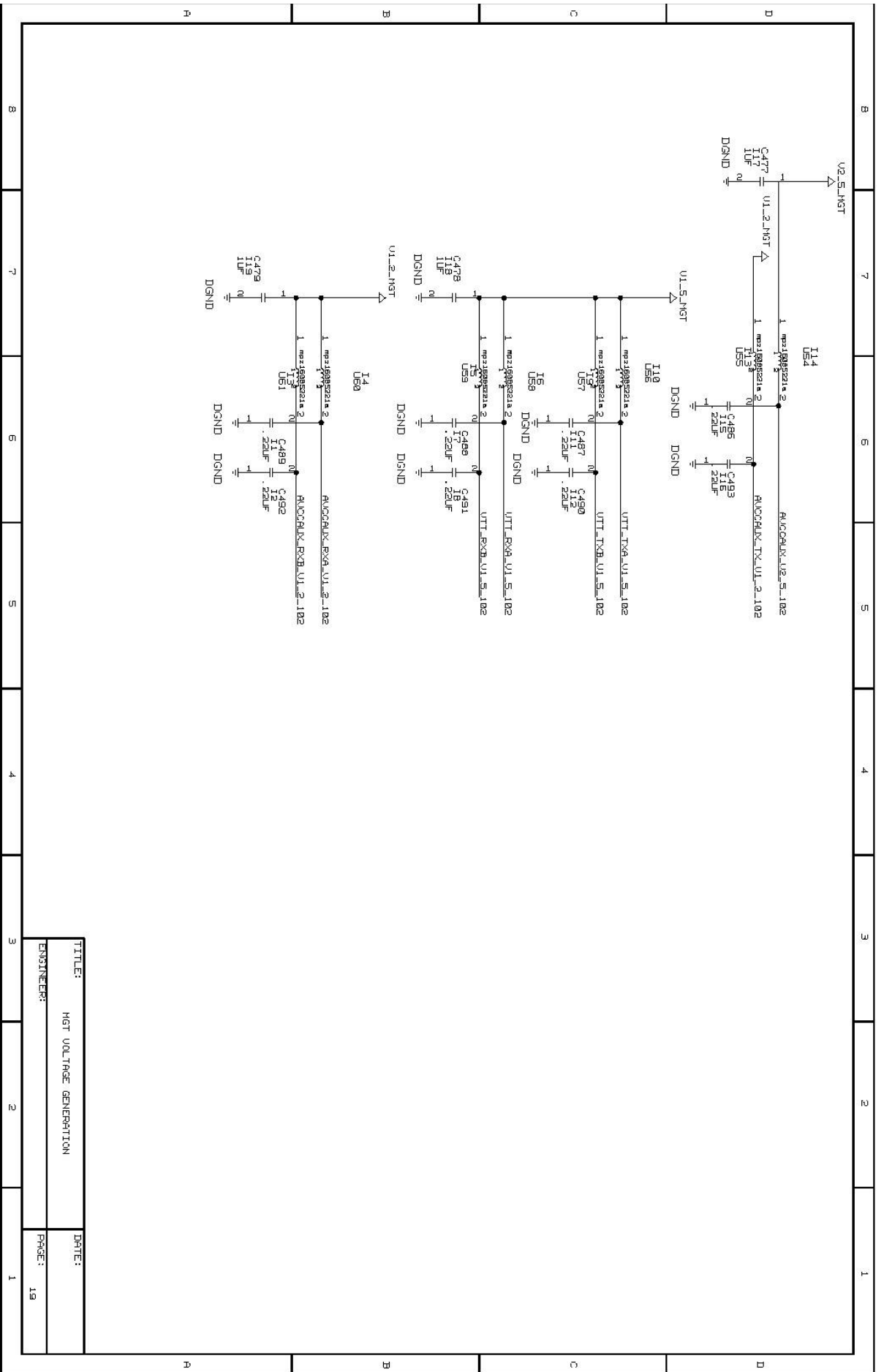


TITLE:	SATA/SFP INTERFACE	DATE:	
ENGINEER:		PAGE:	17



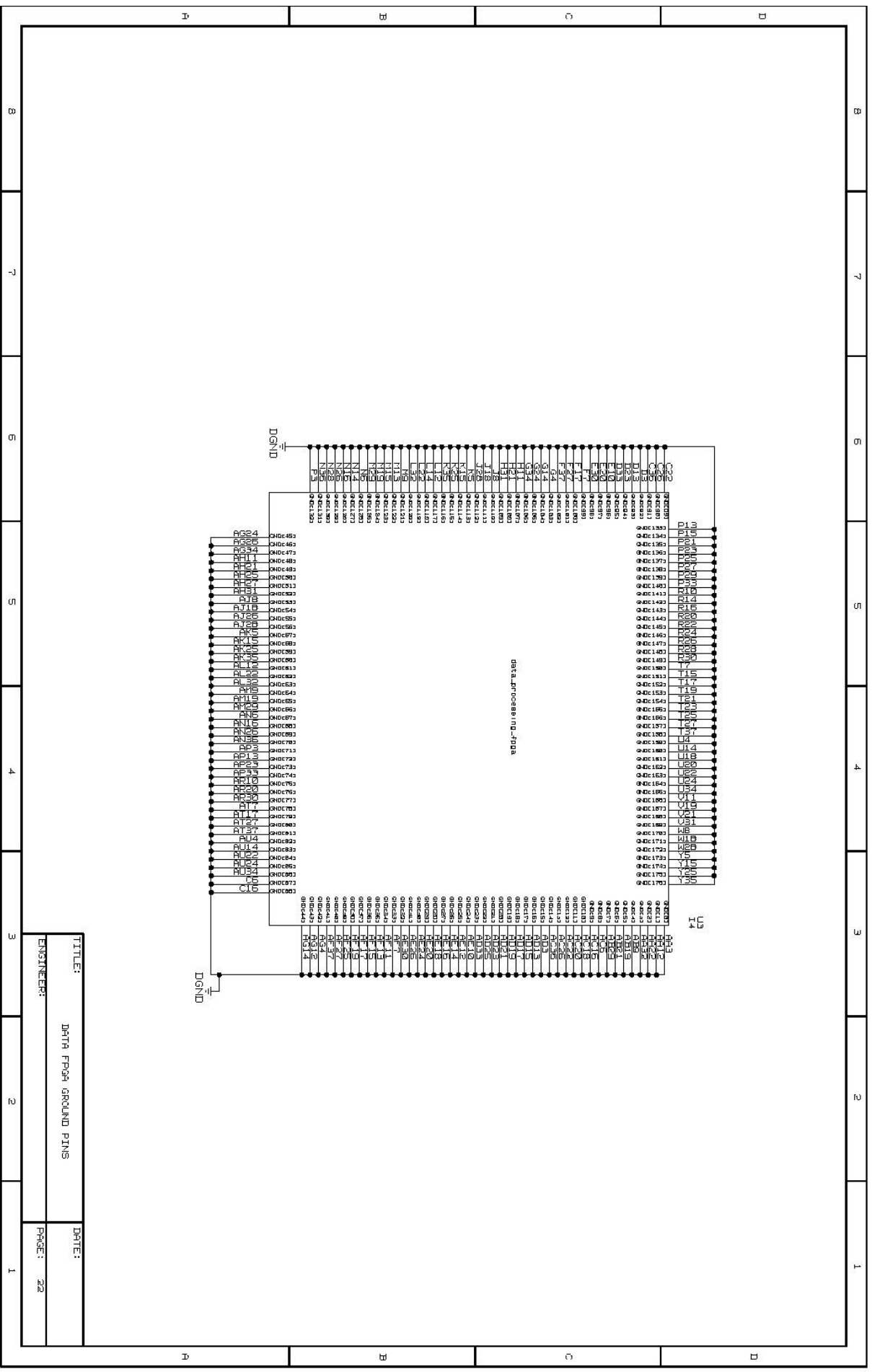


TITLE:	HGT VOLTAGE GENERATION	DATE:
ENGINEER:		PAGE: 19



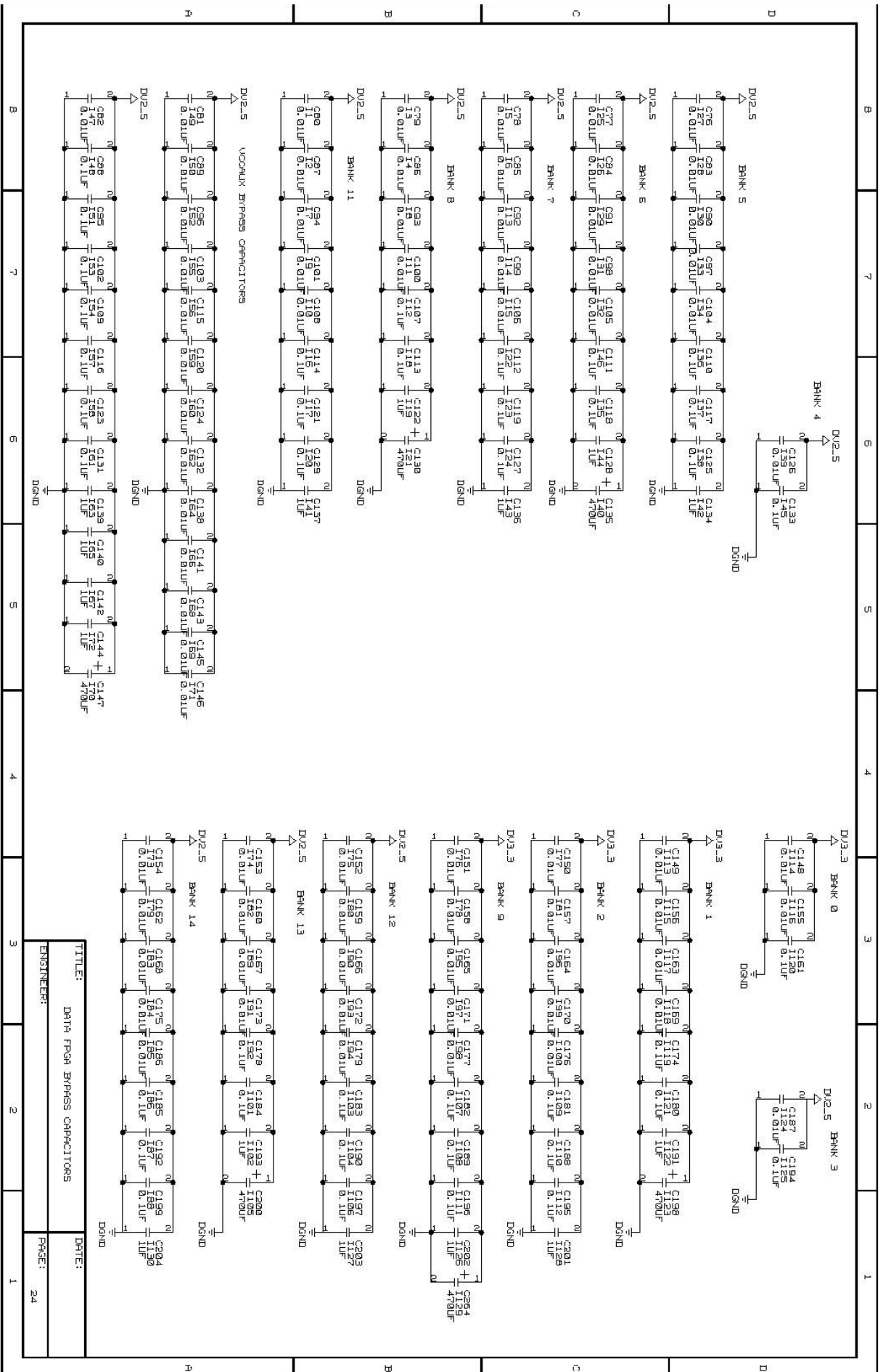
TITLE:	HGT VOLTAGE GENERATION	DATE:	
ENGINEER:		PAGE:	19



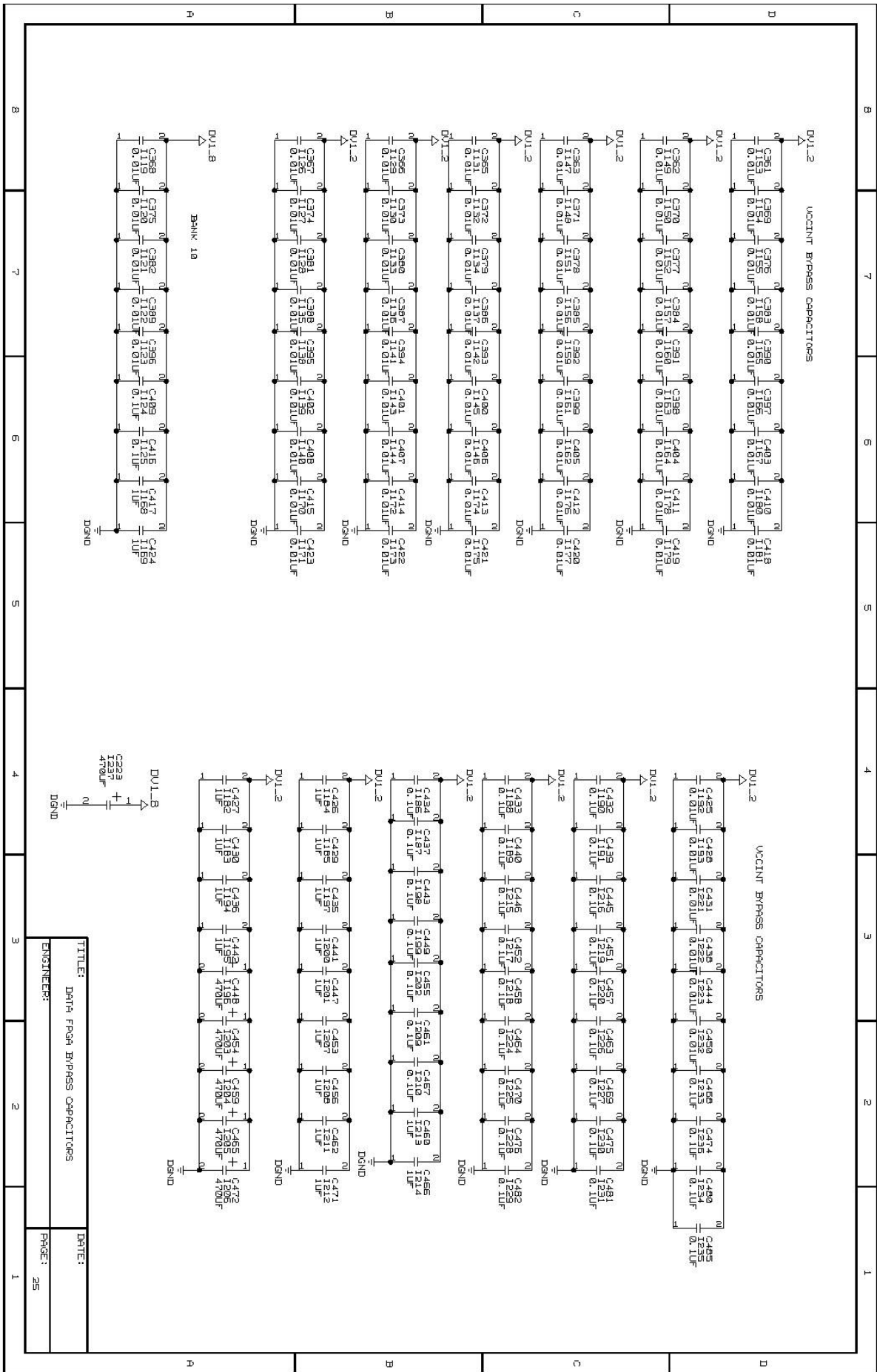


TITLE: DATA FPGA GROUND PINS  
 ENGINEER: DATE:  
 PAGE: 22









TITLE: DATA FROM BYPASS CAPACITORS  
 ENGINEER:  
 DATE:  
 PAGE: 25





## BIBLIOGRAPHY

- [1] A. Ramanathan, R.Tessier, D.McLaughlin, J.Carswell and S. Frasier. Acquisition of Sensing Data on a Reconfigurable Platform. In *Proc. Of International Geo-Science and Remote Sensing Symposium, Sydney, Australia* (July 2001).
- [2] Capedevila, Juan. Design and Implementation of a Dual Beam Interferometer for Ocean Surface Current Vector Mapping. Master's thesis, University of Massachusetts, Amherst, May 2001.
- [3] M. Petronino, R.Bambha, J. Carswell and W.Burleson. An FPGA-based Data Acquisition System for a 95 GHz W-band Radar. In *Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP), Munich, Germany* (April 1997)
- [4] R.Tessier, and W.Burleson. Reconfigurable Computing and Digital Signal Processing. *Journal of VLSI Signal Processing* (September 2000).
- [5] Xilinx Corporation, <http://www.xilinx.com>. *Virtex-4 Family Overview*, August 2010.
- [6] Xilinx Corporation, <http://www.xilinx.com>. *Virtex-5 Family Overview*, February 2009.
- [7] NASA Jet propulsion Laboratory, <http://jpl.nasa.gov>. *Surface Water and Ocean topography Mission (SWOT) Science Requirements Document*, March 2009.
- [8] L. Fu, D. Alsdorf, E. Rodriguez et. all. The SWOT (Surface Water and Ocean Topography Mission: Spaceborne Radar Interferometry for Oceanographic and Hydrological Applications. In *Conference of Ocean Information for Society (OCEANOBS), Venice, Italy* (September 2009).
- [9] National Semiconductor Corporation, <http://www.national.com>. *Low Power, 8-Bit, Dual 1.5GSPS or Single 3.0GSPS A/D Converter Datasheet*, November 2009.
- [10] Xilinx Corporation, <http://www.xilinx.com>. *Virtex-4 RocketIO Multi-Gigabit Transceiver User Guide*, November 2008.
- [11] OptoIC Technology Corporation, <http://www.optoICtech.com>. *Fiber Optics Transceivers Design Reference Guide: SFP Dual LC Optical Transceivers*
- [12] Design GateWay Corporation, <http://www.dgway.com>. *SATA-IP Host Reference design manual*, December 2008.

- [13] Signatec Corporation, <http://www.signatec.com>. Signatec PX1500-4, 4Channel 1.5GHz/2 channel 3.0GHz Digitizer, September 2010.
- [14] National Semiconductor Corporation, <http://www.national.com>. *Big Gig Reference Platform Manufacturing Kit / Reference Design (A1)*, February 2008.
- [15] National Semiconductor Corporation, <http://www.national.com>. *8-Bit, 3GSPS, High Performance, Low Power A/D Converter with 4K Buffer Datasheet*, July 2009.
- [16] National Semiconductor Corporation, <http://www.national.com>. *8-Bit, 3GSPS, High Performance, Low Power A/D Converter Datasheet*, July 2009.
- [17] PCI Industrial Computer Manufacturers Group (PICMG), <http://www.picmg.org>. *Compact PCI Short Form Specification*, November 1995.
- [18] Xilinx Corporation, <http://www.xilinx.com>. *MicroBlaze Processor Reference Guide*, August 2004.
- [19] On Semiconductor Corporation, <http://www.onsemi.com>. *A Fast Response Low Dropout Linear Voltage Regulator with Enable Datasheet*, September 2009.
- [20] Linear Technology Corporation, <http://www.linear.com>. *3A, Fast Transient Response, Low Noise, LDO Regulators Datasheet*, January 2008.
- [21] M.S.Kennedy Corporation, <http://www.mskennedy.com>. *Rad Hard Ultra Low Dropout Positive Linear Regulator Datasheet*, December 2009.
- [22] Texas Instruments, <http://www.ti.com>. *15-A, 5-V, Input Non-Isolated Wide-Output Adjust Power Module Datasheet*, December 2003
- [23] Texas Instruments, <http://www.ti.com>. *10-A, 5-V, Input Non-Isolated Wide-Output Adjust Power Module Datasheet*, December 2003.
- [24] Xilinx Corporation, <http://www.xilinx.com>. *Virtex-4 FPGA Packaging and Pinout Specification*, September 2008.
- [25] Epson Toyocom Corporation, <http://www.eea.epson.com>. *2.5V Opearation LVDS SAW Osciallator*, Rev1.4
- [26] Palnitkar, Samir. *Verilog HDL: A Guide to Digital Design and Synthesis*, prentice Hall, 1996.
- [27] Xilinx Corporation, <http://www.xilinx.com>. *CORE Generator User Guide*, April 2000.

- [28] Xilinx Corporation, <http://www.xilinx.com>. *LogiCORE IP Initiator/Target v3.167 for PCI User Guide*, June 2009.
- [29] Xilinx Corporation <http://www.xilinx.com>. *Virtex-4 FPGA Configuration User Guide*, June 2009.
- [30] Xilinx Corporation, <http://www.xilinx.com>. *Using a Microprocessor to Configure Xilinx FPGAs via Slave Serial or Select MAP mode*, December 2007.
- [31] Xilinx Corporation, <http://www.xilinx.com>. *Virtex-4 FPGA PCB Designer's Guide*, June 2008.
- [32] Xilinx Corporation <http://www.xilinx.com>. *Xpower Estimator User Guide*, October 2010
- [33] Xilinx Corporation, <http://www.xilinx.com>. *Virtex-4 User Guide*, December 2008.
- [34] M. Rabaey, Jan. *Digital Integrated Circuits, A Design Perspective*. Prentice Hall, 1996.
- [35] Xilinx Corporation, <http://www.xilinx.com>. *Virtex-4 Data Sheet: DC and Switching Characteristics*, September 2004.
- [36] V. Vijayendra, P. Siqueira, H. Chandrikakutty, A. Krishnamurthy, and R. Tessier, "Real-Time Estimates of Differential Signal Phase for Spaceborne Systems Using FPGAs", in the Proceedings of the NASA/ESA Conference on Adaptive Hardware and Systems, San Diego, CA, June 2011.
- [37] P. Rosen, S. Hensley, I.R. Joughin, F.K. Li, S.N. Madsen, E. Rodriguez, and R.M. Goldstein, "Synthetic Aperture Radar Interferometry," IEEE Proc., 88(3), 333-382, 2000.
- [38] P. Siqueira, R. R. Ahmed, J. Wirth, A. Bachmann, "Variable Precision Two-Channel Phase, Amplitude, and Timing Measurements for Radar Interferometry and Polarimetry," IEEE Trans. Microw. Theor. Tech., 55(10), 2248-2256, 2007.
- [39] P. Siqueira, R. Tessier, A. Swochak, K. Fu, D. Esteban-Fernandez, and B. Heavey, "A single-stage, two- channel Ka-band to digital, thermal compensating receiver for SWOT," NASA Earth Sci. Tech. Forum, 4 pp., Arlington, VA, 2010.
- [40] Xilinx Corporation, <http://www.xilinx.com>. *Cordic v4.0, DS 249 data sheet*, April 2009.