The Development of a Radar Digital Unit for the SASARII Project

Justin Mark Webster

A dissertation submitted to the Department of Electrical Engineering,
University of Cape Town, in fulfilment of the requirements
for the degree of Master of Science in Engineering.

Cape Town, May 2004

Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the degree of Master of Science in Engineering in the University of Cape Town. It has not been submitted before for any degree or examination in any other university.

Signature of Author . . . .

Cape Town

11 May 2004

Abstract

This dissertation describes the design, implementation and testing of the Radar Digital Unit (RDU), a subsystem for the South African Synthetic Aperture Radar II (SASARII). The SASARII is an airborne demonstrator SAR system for a spaceborne SAR and Unmanned Aerial Vehicle (UAV) imaging radar. The demonstrator system parameters, such as bandwidth, reflect the desired spaceborne SAR parameters.

The Radar Digital Unit contains the following three modules:

- Digital Pulse Generator (DPG) that outputs a chirp every Pulse Repetition Interval (PRI) for transmission
- Sampling Unit (SU), which samples the received IF signal every PRI, forms a packet of samples and flight information and sends the packet for storage
- Timing Unit (TU), which distributes triggers to the SASARII system every PRI.

Based upon the user requirements, Parsec, a company in Pretoria, South Africa, supplied generic hardware for the RDU. SASARII firmware was developed for each module at Parsec under their guidance and support. The testing of the three modules was conducted at Parsec and at the Radar Remote Sensing Group.

Each module was tested individually. The following was concluded from the test results:

- The DPG firmware and hardware operate to specification
- The SU firmware functions correctly
- The TU firmware simulates correctly. During testing a possible hardware bug was found. Parsec, the supplier of the module, was informed about the hardware bug. At time of writing this dissertation the problem had not been solved.

For my old man

Acknowledgements

I would like to thank the following institutions:

- SANDF for providing me with a bursary over the months I was at the RRSG
- Parsec for their patience and support.

Thanks goes out to the following people:

- My supervisor Prof. Mike Inggs for allowing me the opportunity to complete my MSc at the RRSG and for his continued support
- Research Officer Dr Richard Lord for his support and help. No matter how busy he was, he always provided help and guidance
- Andrew Martens at Openfuel for helping with the PM480 firmware when we went up to Parsec in Pretoria

Contents

List of Figures

List of Tables

Nomenclature

SAR—Synthetic Aperture Radar. The SAR technique uses aircraft or satellite-mounted radar to bounce microwave signals off the ground to produce image data. Both phase and amplitude of the signal are used, in contrast with conventional radar, which uses only amplitude. This allows a larger aperture to be synthesised, thus permitting high resolution images

SASARII—South African Synthetic Aperture Radar II

RCU—Radar Control Unit

RDU— Radar Digital Unit

RFU—Radio Frequency Unit

FDU—Frequency Distribution Unit

NU—Navigation Unit

PRF—Pulse Repetition Frequency

PRI—Pulse Repetition Interval

IF—Intermediate Frequency

DPG—Digital Pulse Generator

SU—Sampling Unit

TU—Timing Unit

RRSG—Radar and Remote Sensing Group

ADC—Analogue to Digital Converter

DAC—Digital to Analogue Converter

SFDR—Spurious Free Dynamic Range

SNR—Signal to Noise Ratio

ENOB—Effective Number Of Bits

LO—Local Oscillator

GPS—Global Positioning System

IMU—Inertial Measurement Unit

RAID—Redundant Arrays of Inexpensive Disk. RAID is the use of multiple hard disk drives in an array that behaves in most respects like a single fast large disk

I— In-phase

Q—Quadrature

FPGA—Field Programmable Gate Array. FPGA is a device providing a large array of configurable logic building blocks and configurable routing interconnects for implementing logic designs

ZBT—Zero Bus Turnaround. ZBT is a synchronous fast SRAM designed to eliminate dead bus cycles during back-to-back read/write and write/read cycles

DMA—Direct Memory Access

PCI—Peripheral Component Interface

FFT—Fast Fourier Transform. FFT is a mathematical algorithm for converting time domain data into frequency domain data

IO— Input Output

NCO—Numerically Controlled Oscillator. NCO is a lookup table is stored in memory containing a sinusoid, the memory pointer is incremented at the sample rate outputting the digital sinusoid which can be used to down convert or up convert a signal

SCSI—Small Computer System Interface. SCSI is a protocol which provides a standard means for transferring data between devices on a computer bus

IDE—Integrated Drive Electronics. IDE is a standard interface for connecting storage devices such as hard disks drives inside PCs

MSPS—Mega Samples Per Second

PMC—PCI Mezzanine Card. PMC is the standard which combines the electrical characteristics of the PCI Bus with the mechanical dimensions of the Common Mezzanine Card. This allows products to act as PCI bus cards but in a more compact and robust form.

PECL—Positive referenced Emitter Coupled Logic. Is a high speed differential signalling standard

RAM—Random Access Memory

PLD—Programmable Logic Device

FIFO—First In First Out

JTAG—Joint Test Action Group. JTAG is a IEEE 1149.1 compatible boundary scan access port for printed circuit board-level continuity and diagnostic testing of the device in which all digital input and output pins are tested

Chapter 1

Introduction

1.1 Project Background

This dissertation focuses on the digital system for the South African Synthetic Aperture Radar II (SASARII). The SASARII is an X-Band airborne SAR, which is scheduled to fly in September 2004. It is a technology demonstrator for a spaceborne SAR and Unmanned Aerial Vehicle (UAV) imaging radar. A consortium was formed with SunSpace, Kentron and the Radar Remote Sensing Group (RRSG) from the University of Cape Town (UCT) as the main members. The aim of the consortium was to show that South Africa has the ability to design and build a high resolution X-band imaging radar. Kentron, a local developer of defence systems, sponsored this project because of their interest in UAV SAR. SunSpace, a company in Stellenbosch, South Africa, also took an active role in the SASARII project. Quoting Mr. Martin Jacobs from SunSpace the Project Manager for the SASARII Project:

"SunSpace is active in the development of complex systems, including satellite technology. To date SunSpace has focused on high-resolution optical imagers as their main earth observation payloads. Yet SunSpace has recognised a growing utilisation awareness of and need for SAR imagers.

As a first step towards addressing these needs, SunSpace launched a SAR Technology Project. The aim being to consolidate and develop the South African SAR capability with SunSpace's ability to deploy these systems on satellites and UAV platforms. The technical capability and manpower development of the UCT's Radar Remote Sensing Group is considered core to the success of this initiative.

The 2003 Technology Project comprises several components, including:

- Simulations and evaluation of SAR processors
- Applications studies
- An X-band airborne demonstrator.

The prime objectives of the airborne demonstrator project are:

- To demonstrate a consolidated South African SAR ability to the local and international communities, by generating high quality images with a South African X-band demonstrator
- To identify and mitigate risks associated with a satellite payload, by first building an airborne SAR.

The initial choices of the airborne parameters, such as bandwidth and centre frequency, are aligned with technology that can be deployed on a micro-satellite SAR, which appears to follow a general trend toward high bandwidth X-band systems."

The SASARII system is comprised of smaller subsystems. Each subsystem has a specific task. The Radar Digital Unit (RDU) is such a subsystem, and it acts as the digital system or data acquisition system for the SASARII project. The design, implementation and testing of the RDU will be described in this dissertation. The RDU consists of the following modules: Digital Pulse Generator (DPG), Sampling Unit (SU) and the Timing Unit (TU). The DPG generates the chirp for the SASARII system. The SU samples the received signal and stores the data on a PC. The TU generates the necessary triggers for the SASARII system.

1.2 User Requirements

The following user requirements were specified for the DPG:

- The Pulse Repetition Frequency (PRF) was specified by the user as 1850 < PRF < 3100
- Bandwidth of the chirp must be 100 MHz
- Pulse Length T is specified as $3\mu s$ < T < $5\mu s$
- The DPG module must be capable of accepting an external clock and external trigger. The clock and trigger must be synchronous to the system clock to ensure coherence
- The clock and trigger jitter may not be greater than an eighth of the smallest period at the input of the Analogue to Digital Converter (ADC).

The following user requirements were specified for the SU:

- The PRF was specified by the user as 1850 < PRF < 3100
- The input bandwidth to the Sampling Unit will be 100 MHz

- 16K real samples or 8K quadrature samples are required per Pulse Repetition Interval (PRI)
- The Sampling Unit must be capable of accepting an external clock and external trigger. The clock and trigger must be synchronous to the system clock to ensure coherence
- IF Sampling will be used in the SASARII system as Baseband Sampling using the Direct Conversion method to acquire inphase and quadrature signals introduces non-linear amplitude and phase mismatch in alternate channels
- The minimum required resolution of the Analogue to Digital Converters is 8-bits
- The clock and trigger jitter may not be greater than an eighth of the smallest period at the input of the ADC.

The following user requirements were specified for the TU:

- All output clocks and triggers must be synchronous to the system clock to ensure coherence
- A maximum platform movement accuracy of eighth of a wavelength (4mm) is required for the platform moving at 150 m/s (540 km/h) in the Azimuth Direction. In other words, we can determine to an eighth of a wavelength where the aircraft is at a specific instant. It is essential for correct SAR operation to match sampled data to position data within a specific accuracy. Therefore data time-stamping in the Sampling Unit must have a resolution of less than 26μs
- The PRF of the system will change depending upon the speed of the aircraft. An increase in speed will mean an increase in PRF. The TU must accept a token and change system PRF depending upon the value sent.

1.3 Scope and Limitations

This dissertation describes the design, implementation and testing of the RDU. The hardware for the RDU was bought from a South African company as SunSpace required. The firmware implementation and testing was conducted at Parsec and at the RRSG, by the author of the dissertation. The design, implementation and testing of the firmware is also described herein.

The integration of the RDU in the SASARII system is beyond the scope of this dissertation. Details of the hardware bought from Parsec will not be mentioned, and it is left to the reader to contact Parsec and request datasheets etc. if desired (some datasheets are given in the Appendix D). Due to a lack of test equipment the TU could not be properly tested.

1.4 Project Overview

Chapter 2: The Background Theory chapter aims to link the user requirements to theory, and to introduce relevant theoretical concepts. The first section gives a basic description of SAR, and there-after it is shown how the system bandwidth is derived from range resolution, and how the number of samples captured every Pulse Repetition Interval is calculated. Next we introduce some sampling concepts. As stated in our user requirements, we are to use IF Sampling in the Sampling Unit. A comparison of IF Sampling and Baseband Sampling will thus be given. It will be noted that IF Sampling is superior to Baseband Sampling when clock jitter is kept to a minimum. Real Analogue to Digital Converters generally perform worse in the second Nyquist zone than in the first Nyquist zone. This reduces the dynamic range of the digitized signal, but we do not get the non-linear amplitude and phase effects caused by Direct Conversion.

The following sampling terms will be introduced to the reader:

- Spurious Free Dynamic Range (SFDR)
- Signal to Noise Ratio (SNR)
- Effective Number of Bits (ENOB)
- Aperture Jitter.

These terms indicate ADC performance and will be used during testing to verify correct operation.

Chapter 3: An overview of the SASARII system will be given, and the various subsystems will be discussed briefly. This prepares the reader for the concept study that defines the problem. Illustrated in Figure 1.1 is the SASARII system level diagram, reflecting the following subsystems:

- The Radio Frequency Unit (RFU) contains both the transceiver and the transceiver controller. The transmitter performs modulation, amplification and filtering of the signal to obtain the desired X-Band Chirp. The receiver demodulates, filters and controls the signal level of the received signal before it is sampled by the SU. The transceiver controller controls the various components in the transmitter and receiver
- The Frequency Distribution Unit (FDU) produces some of the Local Oscillators (LO) required for X-Band transmission and clocks used by the digital system. All LOs and clocks are generated from a 10 MHz stable oscillator
- The Platform is the aircraft, which will carry the SASARII equipment

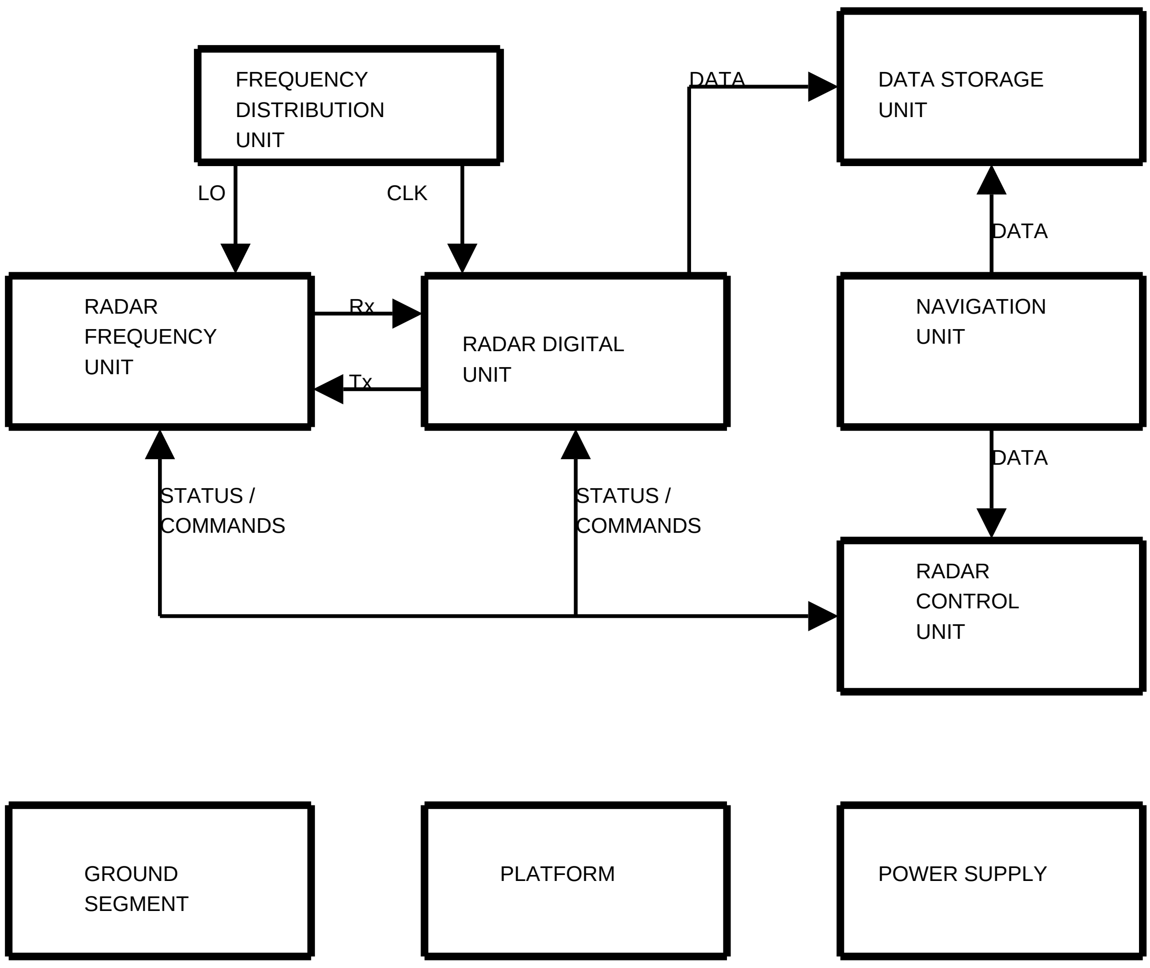

Figure 1.1: System Level Diagram

- The Ground Segment processes the data that is stored during flight. Digital down-conversion and filtering will produce the required quadrature samples before SAR processing

- The Radar Control Unit (RCU) is the system controller. It makes mode/state changes to the system, monitors the system status and changes the PRF based upon aircraft speed, to name but a few

- The Radar Digital Unit (RDU) contains the Sampling Unit, Timing Unit and Digital Pulse Generator. The RDU is the topic of this dissertation

- The Data Storage Unit (DSU) is a server, which will implement RAID 5 to store the space, time and sampled data

- The Navigation Unit (NU) contains an Inertial Measurement Unit (IMU) and Global Positioning System (GPS). The data from this unit is stored on a RAID 5 system for later processing.

Chapter 4: The Concept Study defines the problem based upon the user requirements and system overview, before moving onto the design phase. The design is influenced by the following:

- User Requirements
- The SASARII system architecture
- Hardware cost and availability
- Time until the scheduled flight.

The first section, being the Requirements Analysis, examines the User Requirements and draws certain conclusions. Data Rate calculations are made. The desired sampling rate is calculated using the Nyquist criterion plus a 10% extra oversampling consideration for anti-aliasing filtering.

The second section looks at what influence the system architecture has on the RDU. The timing or coherence issue is discussed further, with special emphasis on the TU and SU. It is mentioned that all clocks and triggers must be synchronous to the system clock. In addition, some sort of time-stamping method will be used to associate the space and time information with sampled data.

Digital downconversion and digital filtering are also discussed. Since SAR processing will be done at the ground segment by using the data from the Data Storage Unit, down-conversion and filtering will not be required in the Sampling Unit.

Limited funding, SunSpace's request for using "South African only products" and the small amount of time until flight, steered the RDU design in a certain direction. Two South African companies who could produce the hardware required were contacted. Several meetings where held between the companies and UCT. Parsec, a company in Pretoria, was chosen to supply the hardware. The device drivers and firmware for the RDU were to be developed by RRSG.

Chapter 5: The System Design chapter describes the RDU design using Parsec hardware.

The DPG is clocked by the FDU and triggered by the TU. Samples are stored in the Field Programmable Gate Array (FPGA) present on the module. The two DAC are clocked with on-board I and Q samples, which will be later mixed up to IF and combined to form the 100 MHz chirp.

The Sampling Unit is clocked by the FDU and triggered by the TU. A packet header will be generated upon trigger assertion, which contains critical information about the particular burst. The most important header information is the time-stamp, which is used to relate platform position to sampled data. The data packet is buffered in ZBT memory before a DMA transfer begins between the SU and the Data Storage Unit.

Each trigger sent to the SASARII system from the TU shall be discussed. The timing between triggers and the method to generate the triggers are described. The triggers will be generated by sending a token to the Timing Unit. The PRF of the system can be updated via this token. A detailed diagram will illustrate the Parsec system, with emphasis placed on the data-flow, clock and trigger paths.

Chapter 6: The Firmware Implementation chapter describes the firmware implemented in the DPG, SU and TU. The coding of the firmware modules was done at Parsec under their guidance and support. The SU and DPG each contain two ACEX Altera FPGAs, and the TU contains a single APEX FPGA. A total of seven FPGAs were programmed. Due to the use of multiple high speed clocks in many of the FPGAs, advanced firmware coding practices had to be implemented. A detailed description will be given of the implemented firmware components and examples of the techniques used to overcome meta-stability and data corruption shall be discussed.

Chapter 7: The Firmware Testing chapter describes the testing methods or procedures used to ensure the correct operation of the SU, DPG and TU. The testing of the modules was conducted at Parsec and the RRSG over a duration of several months. A lack of test equipment did not allow for complex testing. Only basic testing could be done because of a lack of cables, filters, Compact PCI rack and quality signal generators. The results gained from the testing procedures, indicate the SU and DPG operate to specification. A hardware bug was uncovered during testing of the TU. Parsec was notified of the hardware bug and will inform the RRSG when it has been solved.

Chapter 8: Conclusions are based upon the results gathered during testing. The user requirements have been met.

Chapter 2

Background Theory

The aim of this chapter is to link some of our user requirements to theoretical principles. Thus, a brief description of Synthetic Aperture Radar (SAR) is given fist. We also derive the system bandwidth and the number of samples captured every PRI from the range resolution and slant range. Further, we show that IF Sampling is preferred to Baseband Sampling, which is reflected in the user requirement: *IF Sampling will be used in the SASARII system, as Baseband Sampling using the Direct Conversion method to acquire in-phase and quadrature signals, introduces non-linear amplitude and phase mismatch in alternate channels.* Performance measurements and terms, such as Spurious Free Dynamic Range (SFDR) and Signal to Noise Ratio (SNR) will be introduced. These will be used in the testing Chapter 7 to qualify sampling performance.

2.1 Synthetic Aperture Radar (SAR) Basics

Imaging Radar is an active illumination system, unlike optical imaging systems which use the sun's illumination to acquire images. Its advantages over optical images are its ability to penetrate cloud cover and to acquire images at night.

The imaging radar system is usually mounted on an aircraft or spacecraft above the Earth's surface. It emits a radar pulse in a side-looking direction from an antenna. The echo from the reflected pulse off the Earth's surface is received by the radar system and digitized to form raw image data. The ideal side-looking airborne SAR geometry appears in figure 2.1. The swath runs parallel to the flight track.

Pulses with a bandwidth B_r and centred at f_c are transmitted at a Pulse Repetition Frequency f_{prf}. The received pulses are sampled at a sampling rate f_s using an Analogue to Digital Converter. The real sampling frequency $f_s > 2B_r$ satisfies the Nyquist criterion. When direct sampling is used to digitize the received waveform, In-phase (I) and Quadrature (Q) samples are extracted using signal processing methods, which are described in section 4.3.

Synthetic Aperture Radar is a coherent system, which means that the phase as well as

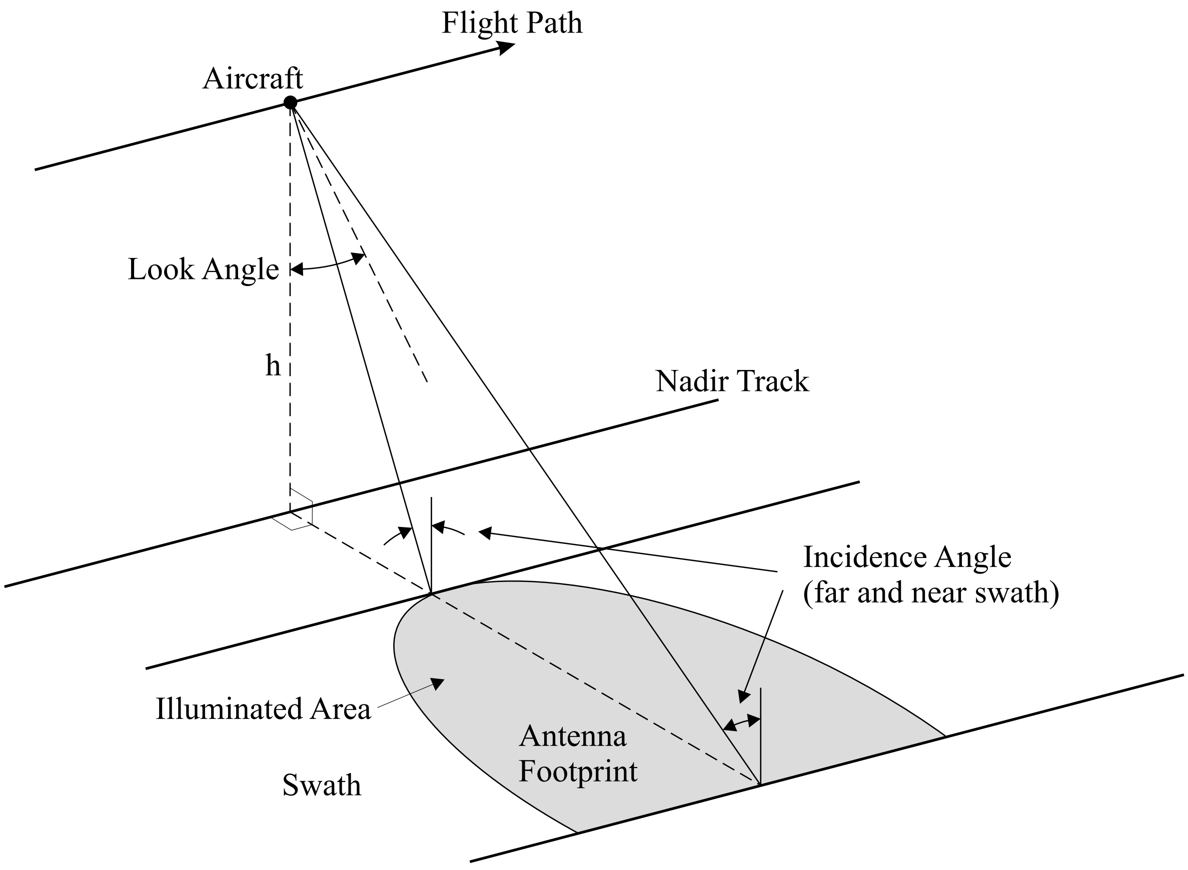

Figure 2.1: Synthetic Aperture Radar Geometry

the amplitude of the returned signal is sampled. Coherence is a term used to describe a system's ability to preserve the phase of a received signal [22]. A SAR system with poor coherence will produce images that are blurry and unclear.

2.2 Range Resolution and Sampling Bandwidth

"The resolution of the radar in (ground) range is defined as the minimum range separation of two points that can be distinguished as separate by the system" [23].

Ground range resolution is given as

$$\Delta R_g = \frac{c}{2B_r \sin\theta}$$

where θ is the incidence angle and B_r is the bandwidth of the transmitted pulse. Figure 2.2 displays a graph illustrating ground range resolution versus incidence angle for various system bandwidths.

A system bandwidth of 100 MHz was chosen, as a SASARII system requirement stipulated a ground resolution of 2m. An incidence angle of $80\,\mathrm{deg}$ will be used to achieve a 2m ground resolution. So we should directly sample at greater than 200 MHz or complex sample at greater than 100 MHz to satisfy the Nyquist criterion.

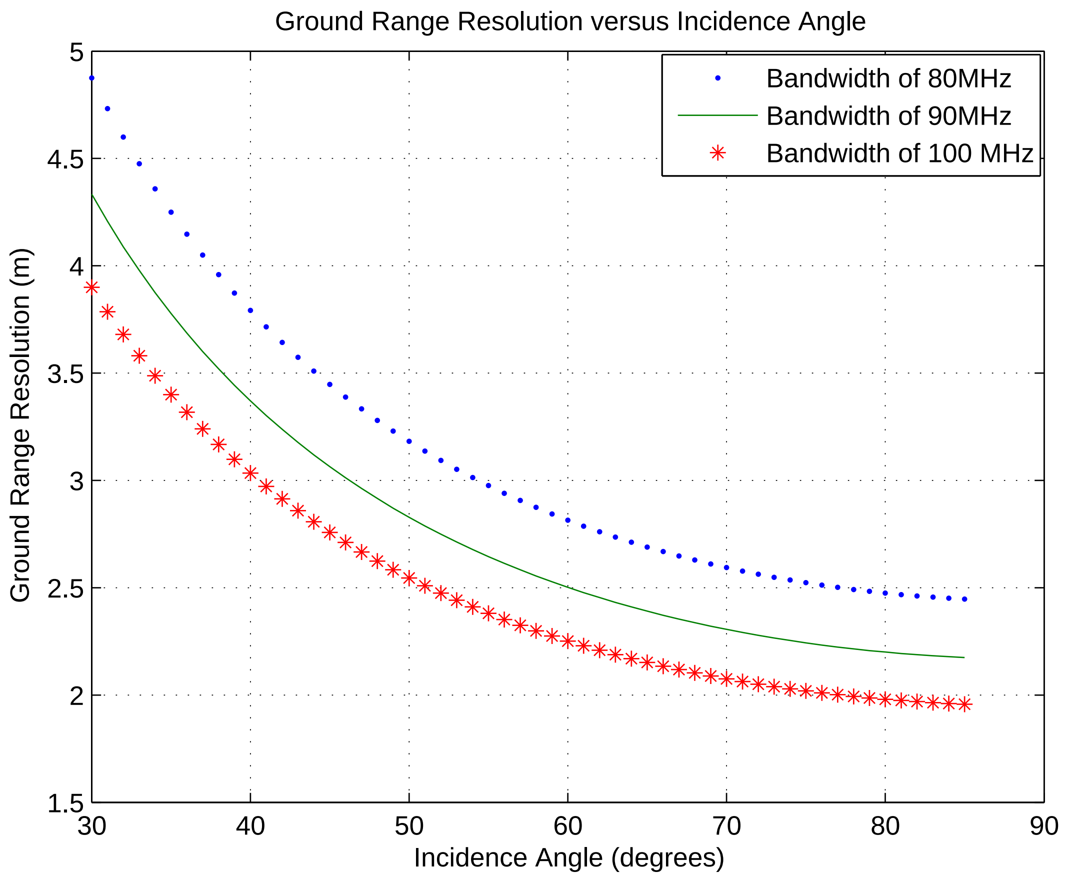

Figure 2.2: Ground Range Resolution vs. Incidence Angle

2.3 Slant Range and Sampling Duration

Slant range resolution is given as

$$\Delta R_s = \frac{c}{2B_r}$$

sampling the received signal at a complex frequency of f_{ad}, slant range bin width can be written as

$$\Delta r = \frac{c}{2f_{ad}}$$

Since the complex sampling rate is equal to or bigger than the system bandwidth, the slant range bin width is equal to or smaller than the slant range resolution. The total slant range width can then be calculated as follows

$$Rs = N_s \times \Delta R_s$$

where N_s is the number of samples captured every PRI. Assuming a sampling frequency of greater than 100 MHz, a graph of Slant Range vs. Number of Samples was plotted, the results of which are shown in Figure 2.3. A slant range within the region of 10 km is desired, and therefore the number of complex samples captured every PRI should be within the region of 8000 samples. To make it easier to process the data (FFTs), 8192 complex samples were specified for capture.

2.4 Sampling Methods

This section describes two sampling methods, Direct Conversion (Zero-IF) and Direct IF Sampling.

2.4.1 Direct Conversion

The Direct Conversion method is shown in Figure 2.4 [2].

The IF signal is mixed down to baseband, using two mixers with signals at the same frequency but with 90 degrees phase difference. Anti-aliasing lowpass filters are used before sampling. Two ADCs are used for Direct Conversion or Zero-IF.

As a result of a parallel downconversion stage, Direct Conversion introduces device mismatch in the mixers, filters and ADCs, all of which cause phase and gain imbalance. Ad-

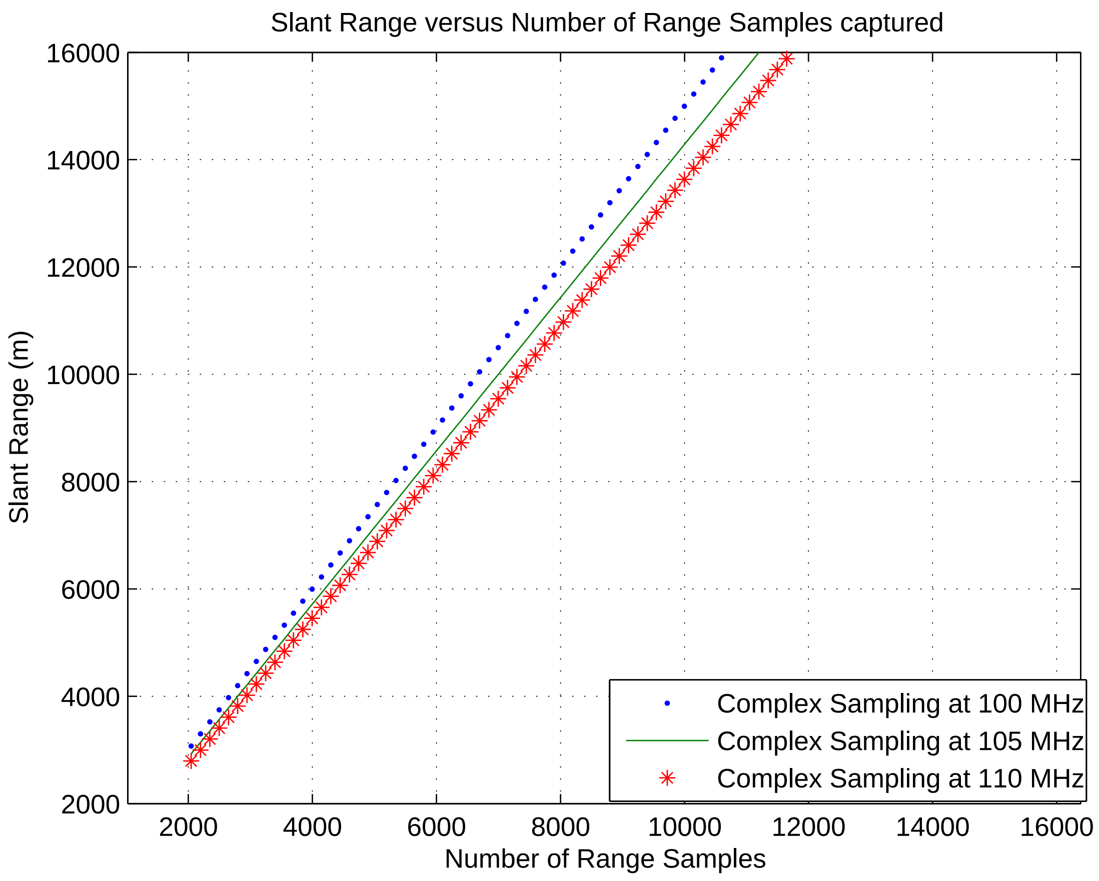

Figure 2.3: Slant Range vs. Number of Samples

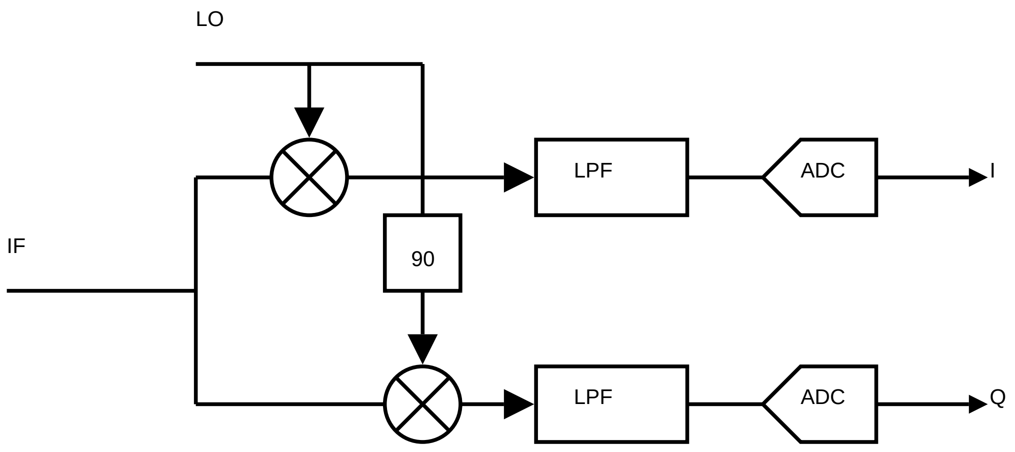

Figure 2.4: Direct Conversion

ditionally, component parameters change with environmental effects, such as temperature and humidity, which cause drifts in performance over time[1].

2.4.2 Direct IF Sampling

When IF Sampling or Direct Sampling is used in receiver applications, it eliminates one or several IF downconversion stages. The ADC replicates the signal in all Nyquist zones. Sampling must be greater than twice the bandwidth of the signal. If the signal is not already at baseband, then a Numerically Controlled Oscillator (NCO) can be used to digitally downconvert the signal in one of the Nyquist zones (discussed in Chapter 4). Figure 2.5 illustrates the effect of IF sampling.

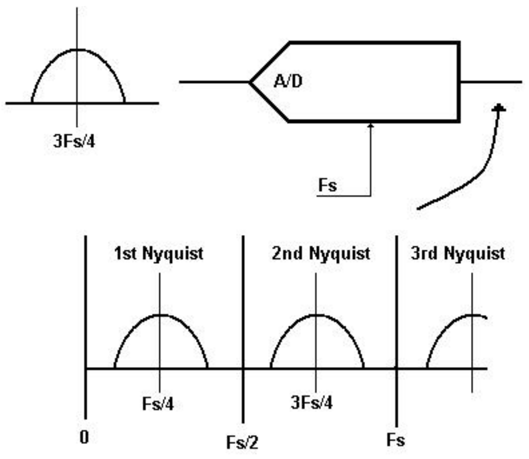

Figure 2.5: Direct IF Sampling

F_s is the sampling frequency. The original signal is at an IF of $\frac{3F_s}{4}$. Sampling at F_s replicates the signal in the Nyquist zones $\frac{NF_s}{2}$. The signal now centred at $\frac{F_s}{4}$ can be digitally downconverted using an NCO.

Direct sampling eliminates many of the undesirable effects caused by parallel downconversion in Zero-IF. Dual Mixers, dual lowpass filters and dual ADCs associated with Zero-IF are replaced with a single high performance ADC, thereby eliminating phase and gain imbalance caused by parallel channels.

2.5 Sampling Performance Terms

The performance terms introduced in this section will be used to qualify the results obtained during testing.

2.5.1 Spurious Free Dynamic Range (SFDR)

This is the ratio of the signal amplitude to the spurious noise component or the highest harmonic amplitude [3]. When measuring SFDR the input signal must be at its allowable maximum. SFDR is illustrated in Figure 2.6.

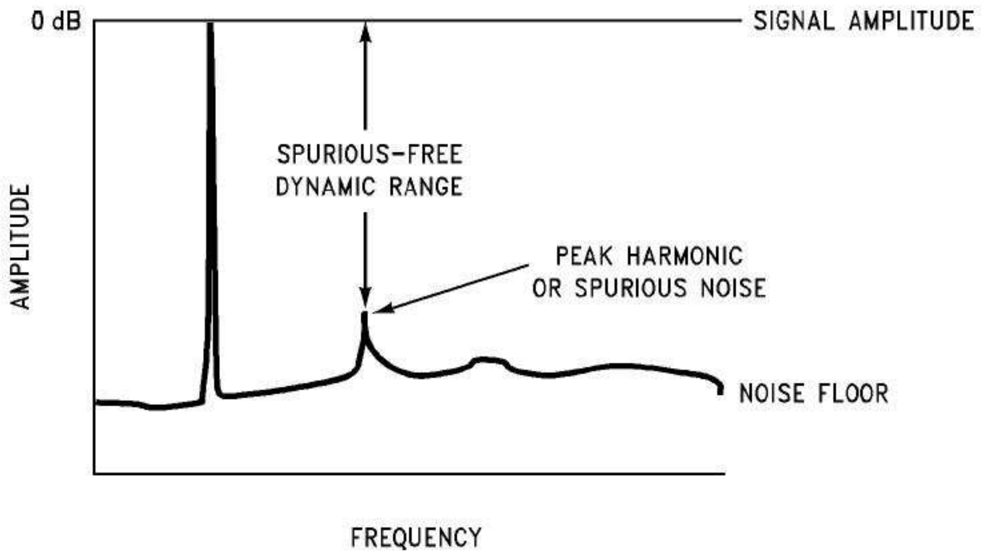

Figure 2.6: Spurious Free Dynamic Range [3]

2.5.2 Signal to Noise Ratio (SNR)

This is the ratio of the *rms* signal amplitude to the sum of the rms noise components [4].

$$SNR = \frac{S}{\sum N}$$

The noise level calculation excludes the spurious content / harmonics of the signal.

2.5.3 Effective Number Of Bits (ENOB)

ENOB is a measurement derived from the Signal to Noise Ratio [5].

$$ENOB = \frac{SNR - 1.76}{6.02}$$

ENOB is a measure of performance given in sampling resolution (bits). Generally, as the input frequency increases so the ENOB decreases[3]. When the signal to be sampled is contained in the 2nd Nyquist zone as in IF sampling, the ENOB will generally be lower than sampling at baseband.

2.5.4 Aperture Jitter or Aperture Uncertainty

“Aperture Uncertainty is the sample to sample variation in the encode process. Aperture uncertainty has three residual effects: the first an increase in system noise, the second an uncertainty in the actual phase of the sampled signal itself and the third is inter-symbol interference."[6]

The theoretical SNR for an ADC is given below, where only Aperture jitter is taken into consideration (thermal noise etc. is ignored).

$$SNR = -20log(2\pi ft)$$

where f is the analogue input frequency and t is the *rm*s jitter.

As input frequency increases, SNR decreases. This is the major limiting factor of a direct IF sampling system. The clock jitter has to be extremely small to give the desired SNR (generally several pico seconds). Costly and precise design practices are needed to ensure pico second jitter specs. The Table 2.1 shows SNR vs. input frequency.

Table 2.1: SNR vs. Input Frequency

SNR (dB)	**Freq(MHz)**	**Jitter (pico)**	**SNR(dB)**	**Freq (MHz)**	**Jitter (pico)**
74.5	30	1	54.5	30	10
68.5	60	1	48.75	60	10
65	90	1	45	90	10
62.5	120	1	42.45	120	10
60.5	150	1	40.5	150	10
58.9	180	1	38.93	180	10
57.6	210	1	37.59	210	10
56.4	240	1	36.43	240	10

Consider the case where the jitter is 10 pico seconds. The calculated ENOB for an input frequency of 150 MHz is 6 bits. So even if you have an ADC with 14-bit resolution, the effective resolution is only 6-bit, when clock jitter is 10 pico seconds.

2.6 Conclusions

Coherence is a system's ability to preserve both magnitude and phase. To preserve phase, we must satisfy the user requirement, *all output clocks and triggers must be synchronous to the system clock.* The system bandwidth and number of samples captured every PRI was derived from range resolution and slant range, reinforcing some of our user requirements.

Sampling methods and sampling performance parameters were introduced. The sampling method direct IF sampling has the following advantages over Direct Conversion:

- Eliminates parallel downconversion, filtering and sampling stages. Phase and amplitude mismatch therefore do not occur between in-phase and quadrature channels

- Performance is not downgraded due to environmental effects, such as temperature and humidity differences.

Direct IF sampling has the following disadvantages:

- Special design practices must be made to keep clock jitter to a minimum, which is usually costly
- Sampling a signal in the second Nyquist zone generally reduces the ENOB.

Chapter 3

SASARII System Overview

The SASARII system comprises a number of smaller subsystems. In this chapter a brief description will be given of each subsystem. This will provide the reader with a general understanding of system operation. A system level diagram is given in Figure 3.1 to illustrate connectivity.

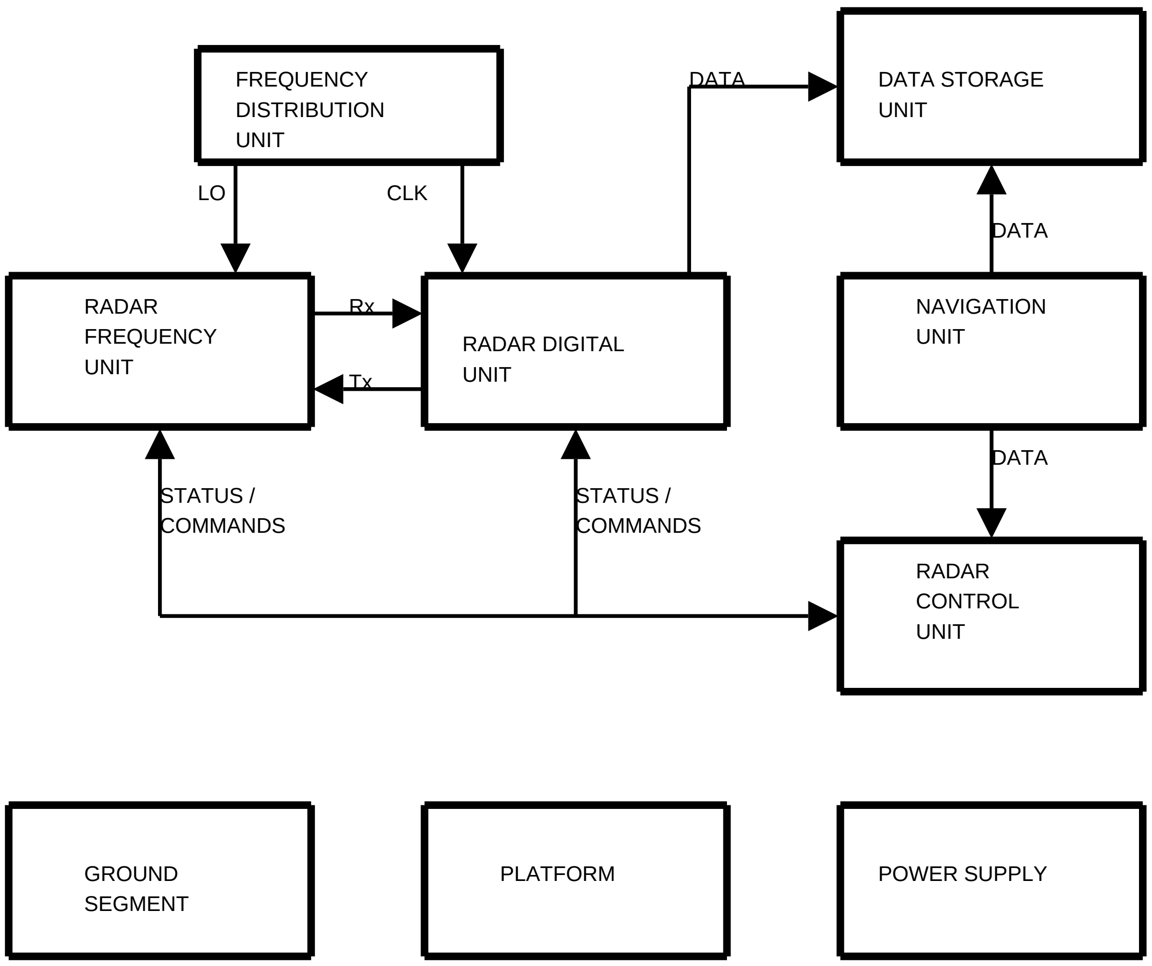

Figure 3.1: SASARII System Level diagram

The Data Storage Unit (DSU) receives data from the Navigational Unit (NU) and the Radar Digital Unit. The NU provides position and time measurements, whereas the RDU captures samples from the received pulse. To form images it must be possible to relate space and time data to the captured data.

The Frequency Distribution Unit (FDU) supplies clocks to the RDU and Local Oscillators (LO) to the Radar Frequency Unit (RFU).

In order to control the system, the Radar Control Unit (RCU) must receive space and time measurements and user commands. Space and time measurements set the PRF of the system. An increase in speed will mean an increase in PRF. During operation the system will move through various states and modes. The RCU supplies these commands to the various subsystems.

3.1 Radio Frequency Unit (RFU)

The Radio Frequency Unit is comprised of the transceiver controller, transmitter and receiver.

3.1.1 Transceiver controller

This provides an interface between the transceiver and the RCU by using an Ethernet connection. The module contains a Mac and an IP address. A web page is stored in memory on the module. An Atmel microcontroller runs software which updates the web page and translates commands which are supplied by the RCU. The web page is used to monitor the transceiver status and modes.

3.1.2 Transmitter

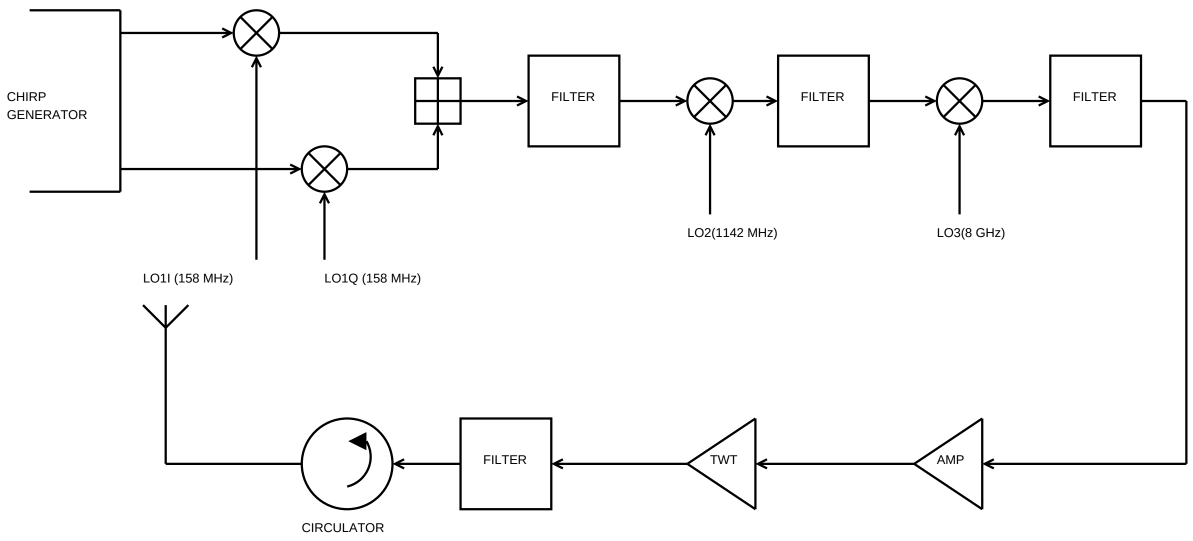

Figure 3.2: Transmitter Architecture [7]

Figure 3.2 shows the architecture of the transmitter. The chirp generator outputs In-phase and Quadrature waveforms which are mixed with an IF In-phase and Quadrature modulator that forms the IF chirp of 100 MHz. The signal moves through two stages of up conversion. Filtering removes unwanted noise. The signal is amplified by the Travelling Wave Tube (TWT) to produce the high powered X-band chirp.

3.1.3 Receiver

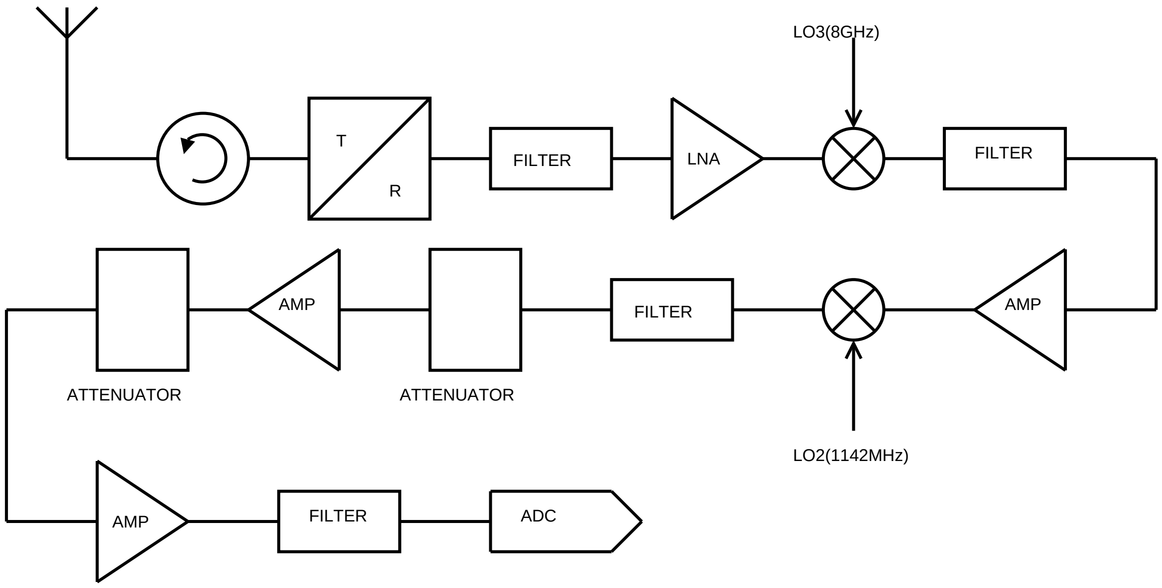

Figure 3.3: Receiver Architecture [8]

Figure 3.3 illustrates the architecture of the receiver. A pulse is received every PRI. The first stage contains the circulator, transmit-receive component, filter and Low Noise Amplifier. The transmit-receive component is used to protect the receiver when a chirp is transmitted. The signal passes through two downconversion stages. Amplifiers and attenuators are used to set the desired signal level. Before direct IF sampling, the signal is passed through a bandpass filter centred at the IF frequency with a bandwidth of 100 MHz.

3.2 Frequency Distribution Unit (FDU)

The FDU is illustrated in Figure 3.4. A 10 MHz stable oscillator is used to synthesize all oscillators and clock signals. The clocks are used for clocking digital circuitry, and the oscillators are used to modulate and demodulate. All clocks and oscillators are synchronous to the 10 MHz stable oscillator.

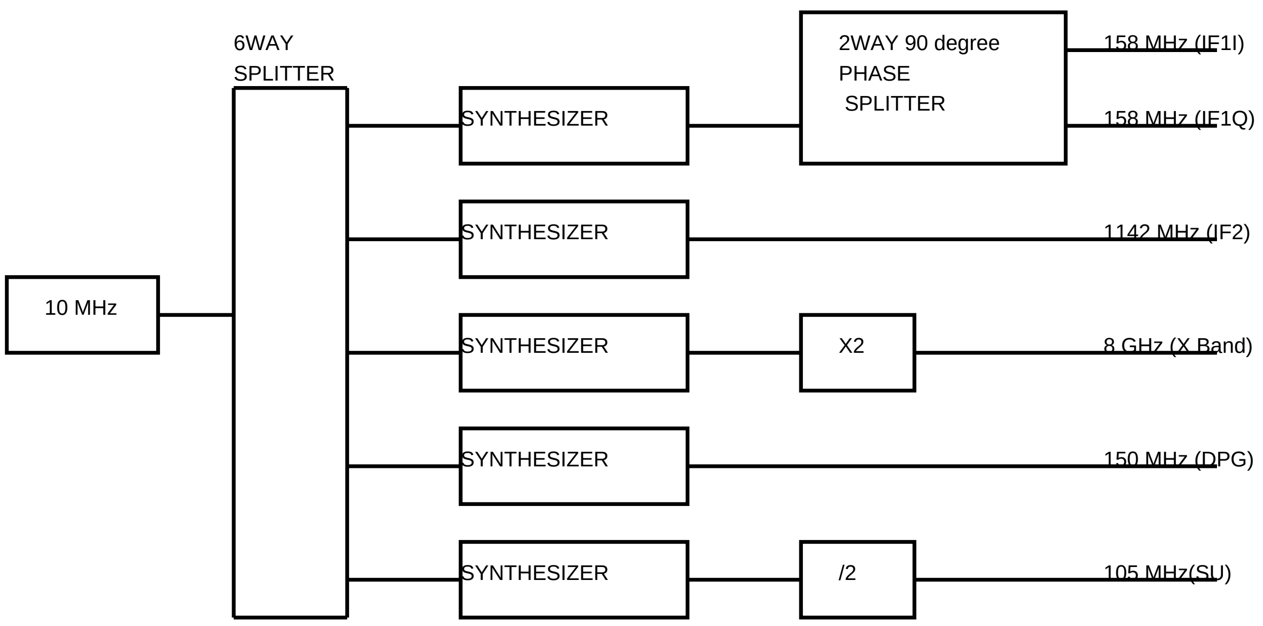

Figure 3.4: Frequency Distribution Unit [9]

3.3 Platform

The platform is the vehicle on which the SASARII is mounted. A small aircraft used for weather monitoring shall be the platform (Aero-commander 690). It has an adequate amount of space, and provision has been made to mount racks. No alterations to the platform are required.

3.4 Ground Segment

After the flight, raw data stored in the Data Storage Unit is processed to form the images. Downconversion and digital filtering are performed before processing to obtain baseband In-phase and Quadrature samples.

3.5 Radar Control Unit (RCU)

The RCU receives space and time data from the NU. The RCU processes the information and makes PRF changes based upon the calculated speed. The RCU monitors the RFU via the transceiver controller (Ethernet). The RCU alters modes and states based upon certain conditions, the received power will be monitored, and gain changes will be made to obtain the desired level. The RCU controls the Radar Digital Unit via a standard PC interface, and it issues commands to change the modes/states of the RDU (sample mode, test mode, etc.).

3.6 Radar Digital Unit (RDU)

The RDU is the topic of this dissertation. It contains three modules: Sampling Unit (SU), Digital Pulse Generator (DPG) and Timing Unit (TU). The Sampling Unit samples the received pulse. The input to the SU is an IF signal of 158 MHz with a bandwidth of 100 MHz. The DPG generates the chirp waveform that is stored in the module. At every rising edge of the DPG external trigger, the DPG outputs a chirp pulse with a bandwidth of 100 MHz. The trigger occurs every PRI. The Timing Unit receives a token from the RCU and alters the PRF of the triggers based upon the value sent.

3.7 Data Storage Unit (DSU)

The Data Storage Unit stores data received during the flight. The data contained on the SCSI disks is implemented in a RAID 5 configuration. SCSI disks are used as storage devices because they are generally more robust and have higher transfer rates than IDE. A server type PC is used for increased processing power and higher memory transfer rates.

3.8 Navigational Unit (NU)

The Navigational Unit consists of the IMU and GPS modules. The IMU with built in GPS was loaned from Kentron, it outputs:

- position data in the form of x, y, z coordinates
- orientation data in the form of roll, pitch and yaw
- time data which is produced from a built-in GPS system.

The data is output in the form of a packet, via the RS-422 standard. The IMU outputs a 1 Hz clock, which is synchronous to the capture of data. This clock can be used to synchronize the captured space and time data with another system's captured data (received sampled data).

3.9 Conclusions

The RDU interfaces with several subsystems within the SASARII system. Generic hardware or off-the-shelf hardware must be purchased and made SASARII specific by programming. Another option is to provide user requirements to a engineering company or organisation and to sub-contract the work. Purchasing the hardware or contracting out the work is discussed further in the next chapter, the Concept Study.

Chapter 4

Concept Study

The concept study dealt with in this chapter defines the problem examined in this dissertation. The design of the RDU is based upon the User Requirements, SASARII system architecture, time until scheduled flight, local hardware suppliers and cost. We will analyse the requirements and the SASARII architecture, and draw relevant conclusions. Two local hardware suppliers were contacted, and details of the findings will be given.

4.1 Requirements Analysis

Based upon the requirements for the RDU, a number of conclusions are drawn.

4.1.1 DPG Requirements Analysis

- *The Pulse Repetition Frequency was specified by the user as 1850 < PRF < 3100*: No special considerations need to be examined
- *Bandwidth of the chirp must be 100 MHz*: To satisfy the Nyquist criterion we must sample greater than 200 MSPS. Due to expected losses we must oversample by at least 10%. So our sampling rate should be > 220 MSPS
- *Pulse Length T is specified as 3μs < T < 5μs*: The number of samples clocked out the DPG is dependent upon the sampling rate and the pulse length. $N = Fs \times T$ where N is the number of samples, Fs is the sampling frequency and T is the pulse length
- *The DPG module must be capable of accepting an external clock and external trigger. The clock and trigger must be synchronous to the system clock to ensure coherence*: This poses no special problems
- *The clock and trigger jitter may not be greater than an eighth of the smallest period at the input of the ADC*: The FDU will be supplying the clock to the DPG and should have low jitter. The trigger will come from the TU, and should have a jitter

value of less than a DPG clock cycle. The trigger level will be detected on the rising edge of the DPG clock, so it does not matter where the rising edge of the trigger value occurs, as long as it occurs within a DPG clock cycle. This argument is illustrated as an example in Figure 4.1.

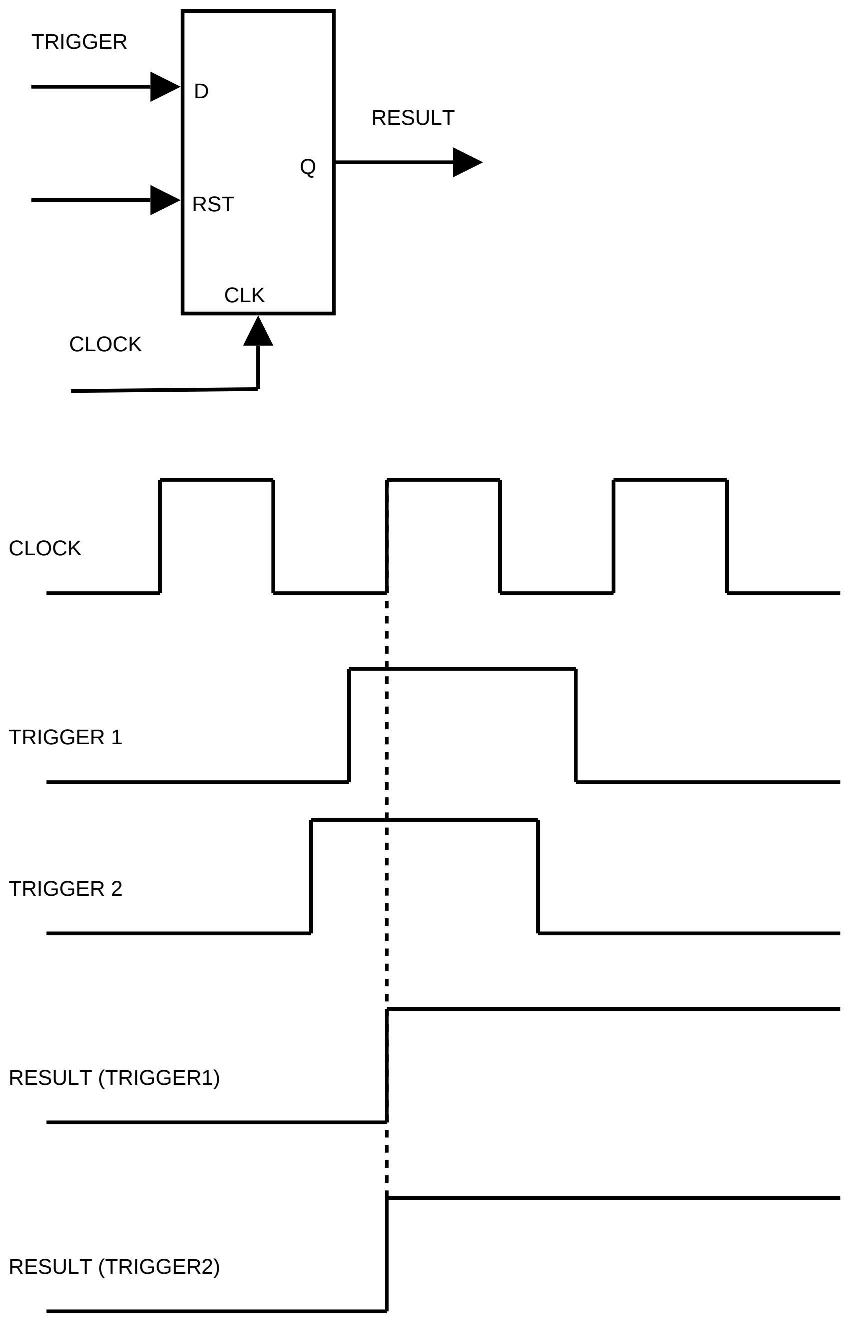

Figure 4.1: Trigger Detection

4.1.2 SU Requirements Analysis

- *The Pulse Repetition Frequency was specified by the user as 1850 < PRF < 3100*: As PRF increases so does the data rate. The data rate is calculated as follows: $D = PRF \times R \times S$, where D is the Data Rate, R is ADC resolution and S is the

Number of Samples captured per PRI. Figure 4.2 illustrates the possible data rates to be expected, with Number of Samples per PRI equal to 16K

- *The input bandwidth to the Sampling Unit will be 100 MHz*: To satisfy the Nyquist criterion we must sample greater than 200 MSPS. We should oversample by 10% to allow for anti-aliasing filtering. Our sample rate should thus be greater than 220 MSPS

- *16K real samples or 8K quadrature samples are required per Pulse Repetition Interval (PRI)*: This requirement may be dropped down to a smaller number of samples per PRI if the data rate is too large for the Data Storage Unit

- *The Sampling Unit must be capable of accepting an external clock and external trigger. The clock and trigger must be synchronous to the system clock to ensure coherence*: This poses no special problems

- *IF Sampling will be used in the SASARII system, as Baseband Sampling using the Direct Conversion method to acquire inphase and quadrature signals, introduces non-linear amplitude and phase mismatch in alternate channels*: To IF sample successfully and attain a high dynamic range, the sampling clock has to have a small jitter spec. (several pico seconds)

- *The minimum required resolution of the Analogue-to-Digital Converter is 8-bits*: In the data rate calculation, the data rate increases with ADC resolution. A decrease in resolution however means a decrease in dynamic range. Ideally we would like a system with a higher resolution than 8-bits if the Data Storage Unit can accept the data rate and the ENOB is greater than 8-bits

- *The clock and trigger jitter may not be greater than an eighth of the smallest period at the input of the ADC*: The FDU will supple the clock to the SU and should have low jitter. The trigger will come from the TU, and it has to be detected by the SU with a jitter of less than a clock cycle (SU clock). The SU clock controls the trigger detection circuitry; that is why we can say the clock and trigger jitter must be within an eighth of the smallest period expected at the ADC input.

4.1.3 TU Requirements Analysis

- *All output clocks and triggers must be synchronous to the system clock to ensure coherence*: This poses no special problems

- *A maximum platform movement accuracy of an eighth of a wavelength (4mm) is required for the platform moving at 150 m/s (540 km/h) in the Azimuth Direction. In other words we can determine to an eighth of a wavelength where the aircraft is at a specific instant. It is essential for correct SAR operation to match sampled*

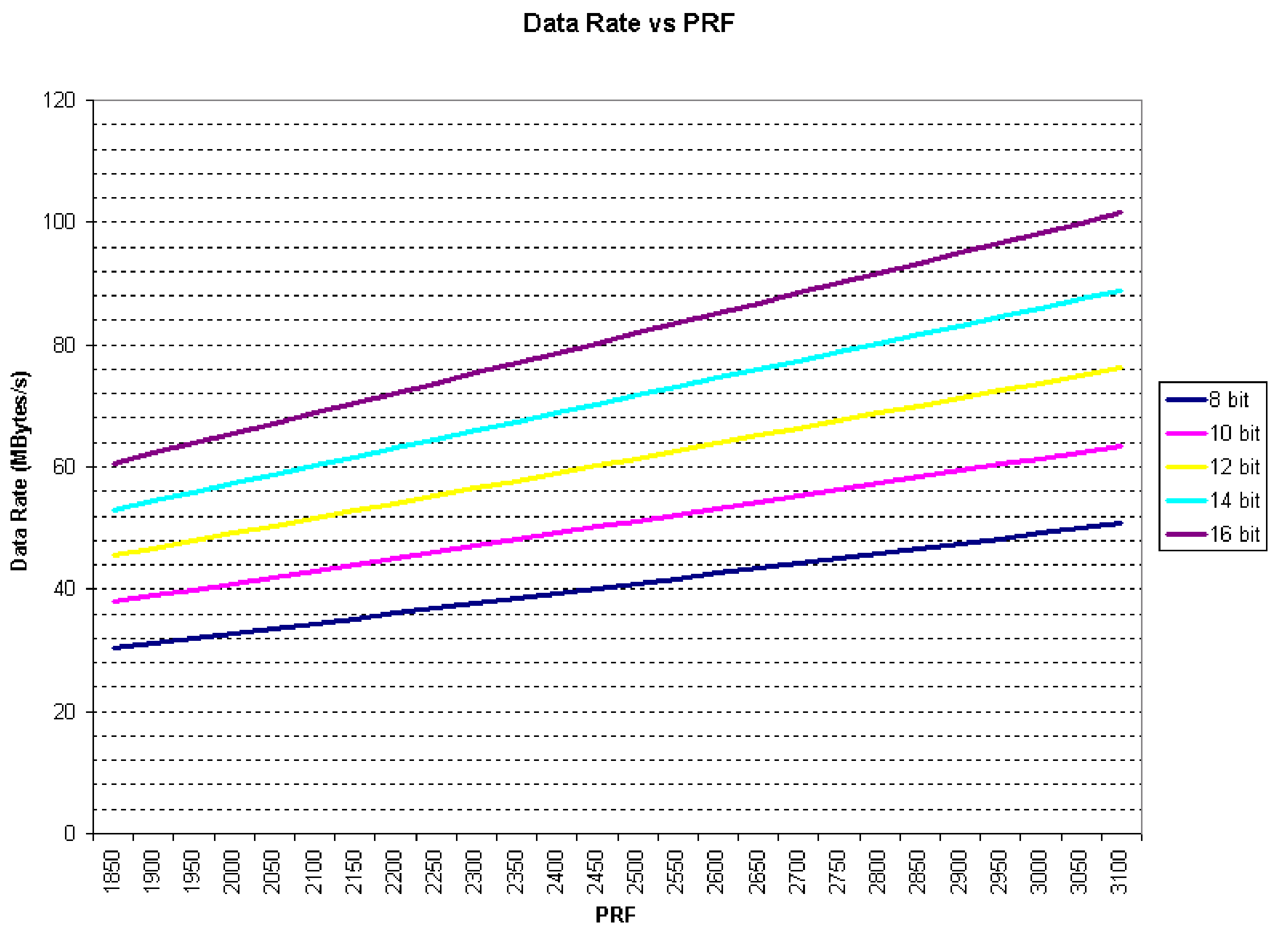

Figure 4.2: Data Rates vs. PRF

data to position data within a specific accuracy. Therefore data time-stamping in the Sampling Unit must have a resolution of less than 26μs: The Navigational Unit supplies the space and time measurements. We want to know to an accuracy of an eighth of a wavelength where the aircraft is positioned and how it is orientated. The space measurements are time-stamped in the Navigational Unit with GPS time. We also want to time-stamp the data in the Sampling Unit, so the space and time data can be accurately matched to the sampled data. This will be done by using a counter in the SU. The counter clock will have to be greater than 40 kHz ($\frac{150m/s}{4mm}$)

- *The PRF of the system will change depending upon the speed of the aircraft. An increase in speed will mean an increase in PRF. The TU must accept a token and change system PRF depending upon the value sent*: This token will be sent by the Radar Control Unit. The RCU will be a PC therefore it will be desirable if the interface is standard to most PCs (PCI, USB, RS-422 etc.).

4.2 SASARII System Architecture concept study

4.2.1 Digital Pulse Generator

The DPG will be clocked externally with a low jitter clock from the FDU. It will be triggered by the TU. Samples prior to flight will be written to the DPG by the Radar Control Unit. During flight the DPG will be triggered and the samples will be clocked out to form the chirp. The pulse length T of the chirp must be within the range, 3μs < T < 5μs. The number of samples stored in on-board memory is directly proportional to the pulse length. Sampling at 220 MSPS will thus require N samples to be stored in memory within the range, 660 < N < 1100, which does not require a large amount of memory to store the samples. A microcontroller or FPGA would be able to store this number of samples in on-board memory. The DPG must however be able to sample greater than 220 MSPS; this requires a single high performance DAC or dual DACs, sampling at greater than 110 MSPS each.

4.2.2 Sampling Unit

With regard to the SU, the main concern was data rate handling. In terms of Figure 4.2 the smallest possible data rate will be 50 Mbytes/s for 8-bit resolution, increases to 100 Mbytes/s for an ADC resolution of 16-bits. These are large data rates and standard PC architectures may not perform at such high speeds. Suppliers selling modules that sample above 220 MSPS tend to use the PCI standard to transfer the data to a PC[10, 12, 11]. Theoretically a 33 MHz/32-bit PCI PC can burst 132 Mbytes/s but in the tests conducted at the RRSG the average data rates were found to be 76 MBytes/s in one direction only on the PCI bus. The experiment had a PCI device writing data to the CPU, but in the

SASARII case we need to store data on hard disk, therefore two iterations along the PCI bus is required, as a storage device controller is likely to use the same bus. So the expected data rate is 38 MBytes/s, which is to low. It is possible to have a number of these PCI modules placed in several PCs and each module triggered on a different PRI to share out the data rate. As this will require several sampling cards and several PCs, this method is undesirable, primarily because funding was limited for this study. It was concluded that we woul need to purchase a sampling module with a sampling rate greater than 220 MSPS, with 66 MHz/64-bit PCI capabilities.

The Navigational Unit, which contains the IMU and GPS modules, will produce data locating the aircraft in space and time, which will be stored in a RAID system. A method was required to relate the space and time data to the received sampled data. The received sampled data will have to be time-stamped. This particular task caused much debate in the RRSG, but it was finally concluded that:

- A time-stamp counter in the SU will count at a frequency > 40 kHz, thus satisfying one of our user requirements. This counter is to be synchronous to our system clock. The system clock is the input clock to the FDU. The input clock is a 10 MHz stable oscillator

- The IMU outputs a 1 Hz trigger, which will reset the time-stamp counter in the SU every GPS second. The 1 Hz trigger has a rising edge exactly when a second rolls over

- The Navigation Unit outputs a 32-bit word giving the GPS time at that specific instant within a resolution of 1s. The SU forms a packet containing the sampled data and information about that specific PRI. Both the 32-bit word and the time-stamp will be latched into every sampling packet header

- It was also decided that a PRI counter should be placed in the sampling packet header. Every time the SU is triggered, the PRI counter is incremented.

To perform the above operations, the SU must have some sort of connector, where we can receive the 32-bit GPS time and the 1 Hz IMU trigger. The time-stamp counter can be generated from the external clock received by the SU, since it is synchronous to the system clock.

4.2.3 Timing Unit

The FDU produces the clocks for the system. The FDU has a number of high quality clock synthesizers, which ensure low jitter. Thus the DPG and SU receive its clocks from the FDU. The main purpose of the TU is to distribute triggers to the SASARII system. As mentioned in the requirements analysis, the DPG and SU must receive their respective triggers from the TU, with a jitter spec of less than one clock cycle with respect to their input clocks. This requires the TU, DPG and SU to use high speed signalling standards like LVDS and ECL for trigger transmission and detection.

There are three main triggers that should be user adjustable by sending a packet or token to the TU, namely:

- Pre-Pulse Trigger, which signals to the Transmitter that the DPG is about to emit a chirp
- DPG Trigger, which signals to the DPG that it must burst a chirp
- SU Trigger, which signals that the SU must begin sampling.

The PRF of these triggers must be adjustable, as well as their positions with respect to one another.

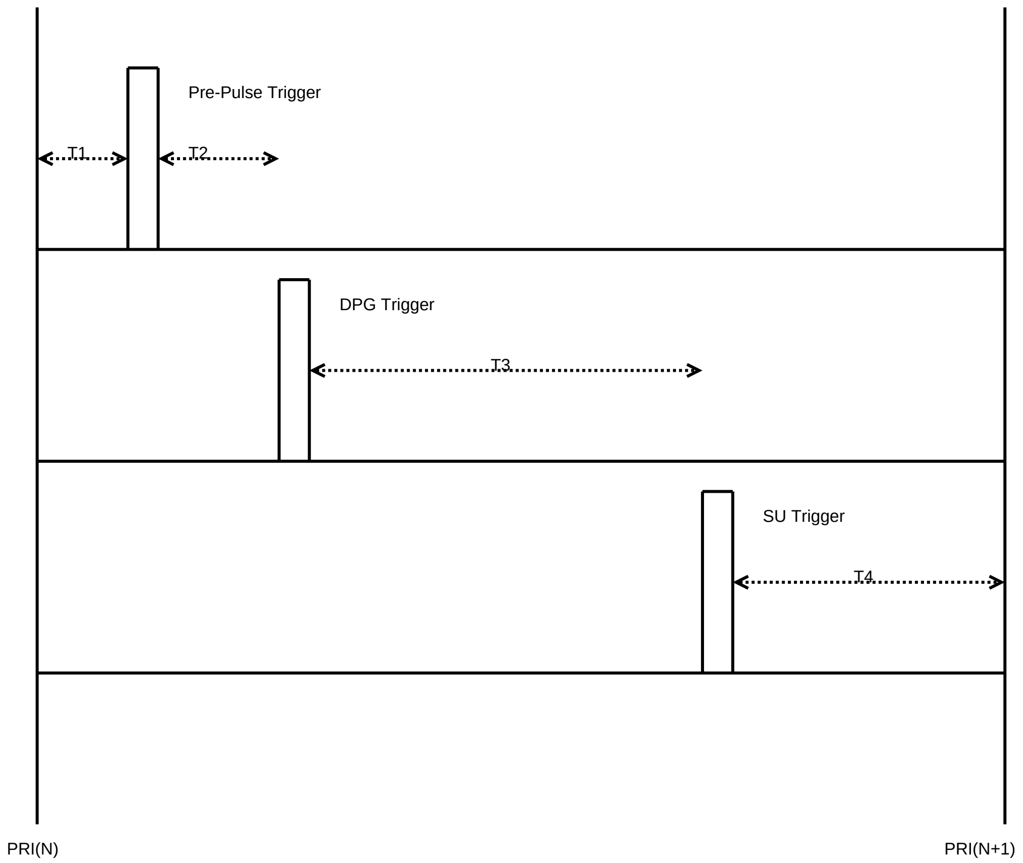

Figure 4.3: Trigger Timing

Figure 4.3 shows a single PRI and the timing of the triggers:

- T1 is the time interval from the start of a PRI until the Pre-Pulse trigger is asserted.
- T2 is the time interval between the assertion of the Pre-Pulse trigger and the DPG trigger. T2 should be long enough to allow the amplifiers in the transmitter to stabilise.
- T3 is the time interval that it takes for the chirp to be transmitted and received.

- T4 is the interval of time it takes to sample 16k samples.

To ensure coherence, the rising edge of the triggers distributed to the system every PRI, must occur at a common rising edge of the digital clock sources and Local Oscillator sources. If the system had to have the following digital clocks and Local Oscillators:

- Digital Clocks: SU clock 105 MHz, DPG clock 150 MHz, TU clock 150 MHz
- Local Oscillators: First IF LO 158 MHz, Second IF LO 1142 MHz.

Then the triggers should have their rising edges synchronous to a 1 MHz clock. The 1 MHz clock is derived by taking the highest common multiple of the clocks and oscillators present in the system. Likewise a 500 kHz and 250 kHz clock would have a rising edge that is common to the digital clocks and Local Oscillators. To have the smallest possible PRF range resolution we would want the highest common clock to drive the trigger pulses.

4.3 Digital Downconversion

Since we are performing Direct IF sampling on the received signal, the downconversion process has to be done digitally. In a real-time system the digital downconversion and filtering would be performed in the SU. But as this is not a real-time system, and as the data is stored in the Data Storage Unit, the downconversion and filtering can be done at the Ground Segment when the data is retrieved. If we are doing IF sampling, with a sampling rate of Fs**,** the IF signal should ideally be placed at $3Fs/4$. This is in the middle of the second Nyquist zone. Sampling at Fs replicates the signal in all other Nyquist zones. To get the desired quadrature signals to baseband, the replicated signal in the first Nyquist zone is multiplied by a NCO, and then lowpassed filtered to remove unwanted signals. Figure 4.4 illustrates this concept.

4.4 Hardware Supplier Concept Study

SunSpace requested the SASARII system to be made up of South African products, if possible. Not only would this allow closer communication and support with local companies, but it would also mean that, when the time came for the spaceborne system to be developed the initial contacts with suitable local companies would have been established already. Our own schedule had to fit in with SunSpaces tight schedule for developing a spaceborne system. We were to have the first flight within a year from when SunSpace approached the RRSG at UCT. Since this is only a demonstrator system and would become useless once images had been acquired, funding was kept to a minimum. All of the above led us to contact two South African companies that could produce the hardware required, namely; Peralex in Cape Town and Parsec in Pretoria[14, 13].

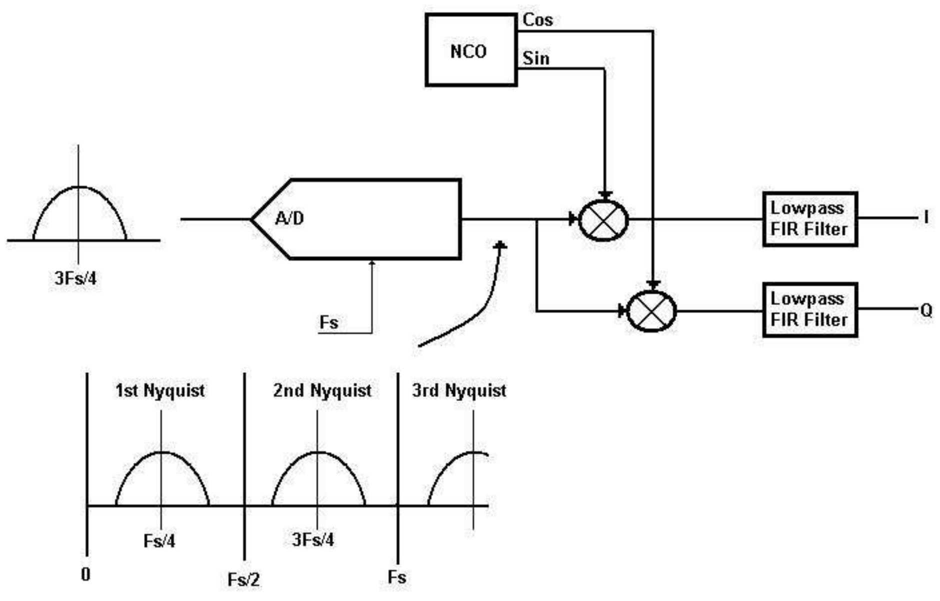

Figure 4.4: Digital Downconversion

At the time of approaching Peralex, our main focus was the SU. We gave them our requirements, where-upon they proposed a SU with a single 250 MSPS ADC and agreed to complete the hardware and software required. This was an ideal unit for our needs. We did not discuss the DPG or TU with Peralex at this stage.

A member of the RRSG contacted Parsec about their PM488, which is a compact PCI module with dual 160 MSPS DAC on board. During the meeting with Parsec, an entire unit which would contain the DPG, TU and SU was proposed.

The PM488 (DPG) had the following features:

- Two 160 MSPS DAC
- 66 MHz/64-bit compact PCI PMC interface
- External clock and external trigger inputs (PECL).

The PM480 (SU) had the following features:

- Two 105 MSPS ADC
- 66 MHz/64-bit compact PCI PMC interface
- External clock and external trigger inputs (PECL).

The PM440 (SU) had the following features:

- 3 PECL output triggers
- 3 PECL output clocks

- 1 PECL input clock
- 1 PECL input trigger
- 66 MHz/64-bit compact PCI PMC interface.

The modules are all in the PMC form, which allows one to make backplane connections between the different modules and the rest of the SASARII system. This allows us to easily send the 1Hz IMU Trigger and 32-bit GPS time to the SU. Parsec specifically used the PECL signalling standard for clocks and triggers to keep jitter to a minimum. This should make it safe to use IF sampling.

As the products being sold were generic parts, the cost was low. At the time of the first meeting, the hardware was already being developed and did not require the initial design phase. The firmware for the modules had not been developed though, and thus it was decided that members of UCT would develop the firmware and device drivers.

Although all these factors made the system look ideal, the SU did however, have two problems that had to be considered:

- The module was designed for Zero-IF, as both ADC are clocked by the same clock
- The two ADC sampling rates combine to 210 MSPS, only allowing for 5 % over-sampling.

The above observations where discussed extensively between Parsec and the RRSG. **Parsec declared that a hardware modification could be made so that the two ADCs would sample 180 degrees out of phase and appear as a single ADC sampling at 210 MSPS**. Furthermore, the 5 % oversampling rate was discussed: it was concluded that, although this was not the ideal situation, aliasing may not occur, as the receiver does extensive filtering prior to sampling.

Between Parsec and Peralex, Parsec was finally chosen to supply the hardware for the RDU. The next chapter looks more closely at the Parsec hardware, and presents the RDU system level design is given.

4.5 Conclusions

The user requirements and system architecture were analyzed to obtain specifications. The specifications were used to choose the data acquisition hardware for the RDU. Two South African hardware suppliers were contacted, and Parsec was chosen to supply the digital hardware. The following three Parsec modules were thus purchased:

- the PM488, a generic dual DAC card, which was used for the DPG
- the PM480, a generic dual ADC card, which was used for the SU

- the PM440, a generic timing card which, was used for the TU.

Writing SASARII specific firmware and programming the Parsec PM4XX modules, our user requirements will be satisfied.

Chapter 5

Radar Digital Unit Design

A description of the PM480, PM488 and PM440 modules with diagrams is taken from Parsec's website[13] and given in Appendix D. The details of the modules are presented at a system level and the reader should browse through the datasheets to gain a general idea of the module architecture and operation.

Described in this chapter is the RDU design. As the Parsec modules had been designed to be generic and thus to allow for a number of applications, some of the hardware components present on the modules were not used. Features were kept to a minimum, as added functionality would require more development and debugging. A simple system tends to be more robust. Using the PM480, PM488 and PM440 modules, we designed a system that contained our SU, TU and DPG.

5.1 Digital Pulse Generator

In the previous chapter we covered the requirements analysis and the effect that the SASARII system architecture had on the DPG design. A brief summary of the main points is given below:

- sampling must be greater than 220 MSPS
- the DPG should be clocked externally by the FDU
- the DPG should be triggered externally by the TU
- the stored samples to be clocked out of the DPG require a small amount of on-board memory.

Various discussions were held within the RRSG regarding the use of the PM488. It was decided that the DPG would do Baseband Sampling. In-phase (I) and Quadrature (Q) samples would be stored in memory prior to flight. A 32-bit word would hold both I and Q samples. Writing it as 32-bits, and not 28-bits (each DAC has 14-bit resolution) would reduce complexity in the device driver and firmware that still had to be written. Upon trigger assertion, the DPG would thus start reading from on-board memory. At the sampling frequency, a 32-bit word would be read from memory and split into two 16-bit words; the lower 14-bits of the 16-bit word would be clocked out of the DACs. The two channels represent the In-phase and Quadrature parts of a baseband chirp. In the transmitter the two signals would be mixed to an IF and combined to form the Chirp. Figure 5.1 illustrates the above description.

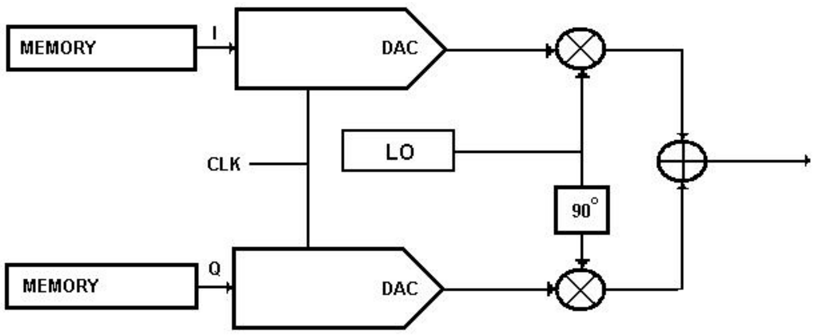

Figure 5.1: Digital Pulse Generator and IQ Modulation

The PM488 has two Altera ACEX EP1K100 FPGA on-board the PMC module[15]. The FPGA closest to the PMC connectors would contain an Altera PCI Core, which is capable of running at 66 MHz / 64-bit[16]. The Core written by Altera and the Wrapper functions written by Parsec would fill the first FPGA, leaving little space for more firmware. But the FPGA closest to the DAC are empty and are large enough to hold our design. The ACEX EP1K100 FPGA has 49152 RAM bits available. This RAM present inside the FPGA would be able to hold 1536x32-bit words. That means that we can hold 1536 14-bit samples for each DAC. If we clock each DAC at 160 MHz and have a chirp of $5\mu s$, we will only require 800 samples per channel.

5.2 Sampling Unit

In the previous chapter we covered the requirements analysis and the effect of the SASARII system architecture on the SU design. A brief summary of the main points is given below:

- sampling rate will be 210 MSPS with two 105 MSPS ADC sampling out of phase by $180 \deg$
- the SU is to be clocked externally by the FDU
- the SU is to be triggered externally by the TU
- the SU output data rate may be between 50 and100 Mbytes/s
- a SU packet containing data and PRI information must have GPS time, PRI number and generated time-stamp placed in its header.

Since we require 16k real samples with 14-bit resolution, which will amount to 8192x32-bit words, there is clearly not enough space in the two FPGAs to buffer this data. Placed on the PM480 board is a Micron ZBT SRAM with enough space for 128kx32-bit words[17]. Thus the 8192x32-bit words are stored in this memory, prior to a PCI transfer.

When triggered, the SU creates the sampling packet header and writes it to ZBT memory. Once finished with the header the 16k real samples are written to ZBT memory. The data stored in memory forms a packet which will be read out via PCI. The packet header has the following structure:

1. Synchronisation Word (0xFFFFFFFF)
2. Packet ID
3. Packet Length
4. Packet Number, PRI Number
5. Time Stamp
6. GPS time
7. Over range counter.

The over range counter indicates how many samples are the result of the input signal being too large. An over range value greater than zero indicates clipping.

The time-stamp is a 32-bit counter that is clocked at 105 MHz. The external clock is connected to the front panel of the module and is fed from the FDU. The counter will take approximately 40.9s to roll over. This counter is rest every 1s using the IMU 1 Hz trigger. The IMU 1 Hz trigger and 32-bit GPS time will be accessible via the Pn4 connector (refer to the PM480 datasheet). By resetting the time-stamp counter every 1s, we can re-synchronize the time-stamp counter to the GPS time. By placing the 32-bit GPS time in the sample packet header and resetting the time-stamp counter once every second, we are able to align the space and time data to sampled data.

Every PRI a sampled data packet is created. The sample packet header is generated by latching the PRI number, time-stamp counter, GPS time and over range counter. The sample header is transferred to the ZBT RAM. After the sample header transfer, the sampled data is transferred to ZBT RAM. Once the last sample has been stored, the packet from ZBT RAM is transferred to the PCI Core, where it acts as a master and does a DMA between itself and PC memory. This is done via the 66 MHz/ 64-bit Compact PCI standard. The expected average data rate will be 100 Mbyte/s. To transfer at these rates, the PCI Core placed in one of the ACEX FPGA must have PCI master capabilities. This allows the PM480 to act as a master on the PCI Bus and to do DMA transfers directly to memory, in bursts. Parsec was queried about the PM480 having PCI master capabilities and

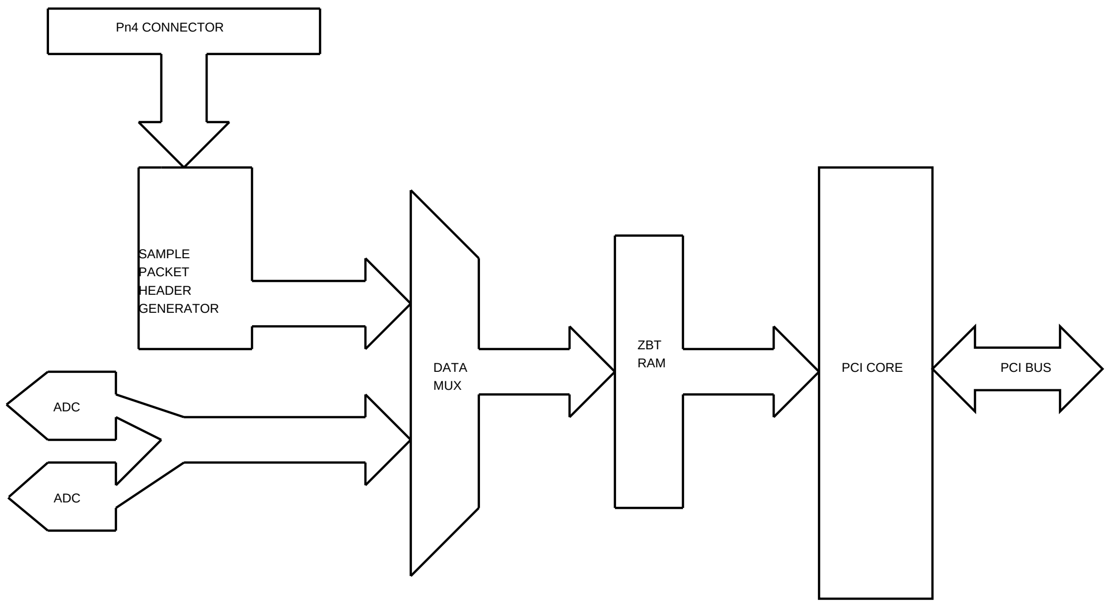

Figure 5.2: PM480 Data Flow

the RRSG were informed that the module did indeed have master capabilities. Figure 5.2 illustrates the data flow.

The IF centre frequency for the SASARII system is determined by the sampling rate. The IF frequency is calculated as follows: the sampling rate is 210 MSPS, and we want the IF signal in the centre of the second Nyquist zone, which is halfway between 210 MHz and 105 MHz, i.e. 157.5 MHz ($3Fs/4$). For practical reasons, the IF frequency was moved up to 158 MHz, as the generation of the 157.5 MHz in FDU would be difficult.

5.3 Timing Unit

In the previous chapter we covered the requirements analysis and the effect of the SASARII system architecture has on the TU design. A brief summary of the main points is given below:

- a token will be sent to the TU to change the PRF
- three external triggers are to be controllable: Pre-Pulse Trigger, DPG Trigger and ADC Trigger
- the three external triggers must occur on the rising edge of the highest common multiple of clocks and LOs in the system.

The token is sent to the TU via PCI. The PRF of the triggers and their relative position with respect to each other must be adjustable. The PRF of the triggers may only be changed once the current PRI is finished. The token appears as shown in Figure 5.3.

The PRF of the system is dependent upon the PRF Frequency Counter Value. This 32-bit word is loaded into a 32-bit counter. The counter counts down. When it reaches a trigger

PRF FREQUENCY COUNTER VALUE	PREPULSE TRIGGER VALUE	DPG TRIGGER VALUE	ADC TRIGGER VALUE

Figure 5.3: Timing Token

value (32-bit word) set by the user Pre-pulse Trigger Value, DPG Trigger Value or ADC Trigger Value, a pulse is output by the TU. This simple method makes sure that all the triggers have the same PRF.

The input clock to the TU must be synchronous to the system clock. To output the triggers on the rising edge of the highest common clock in the system, one must make all values in the token divisible by the factor, to get the input clock to a highest common multiple. For example:

- IF LO = 158 MHz
- DPG clock = 150 MHz
- ADC clock = 105 MHz
- Highest common multiple clock = 1 MHz.

If our input clock to the TU is the DPG clock, which is 150 MHz, the values in the token must all be divisible by 150.

To visualise the entire RDU design a system diagram has been drawn. Figure 5.4 displays data flow, clock paths, trigger paths and connections to the rest of the SASARII system.

5.4 Conclusions

In this chapter we discussed the system level design of the RDU. The design was influenced by the requirements analysis and SASARII system architecture. We can conclude the following:

- the DPG will do baseband sampling with I and Q samples stored in an ACEX FPGA.
- the SU using PCI master capabilities will successfully manage the high data rate expected. The first IF stage frequency (158 MHz) has been set, which was dependent upon the sampling rate (210 MSPS). The structure of the sample packet has been designed and will be implemented in firmware. A system level view of data-flow has been described.
- the TU will accept a token of size 4x32-bit. This token will control the trigger outputs.

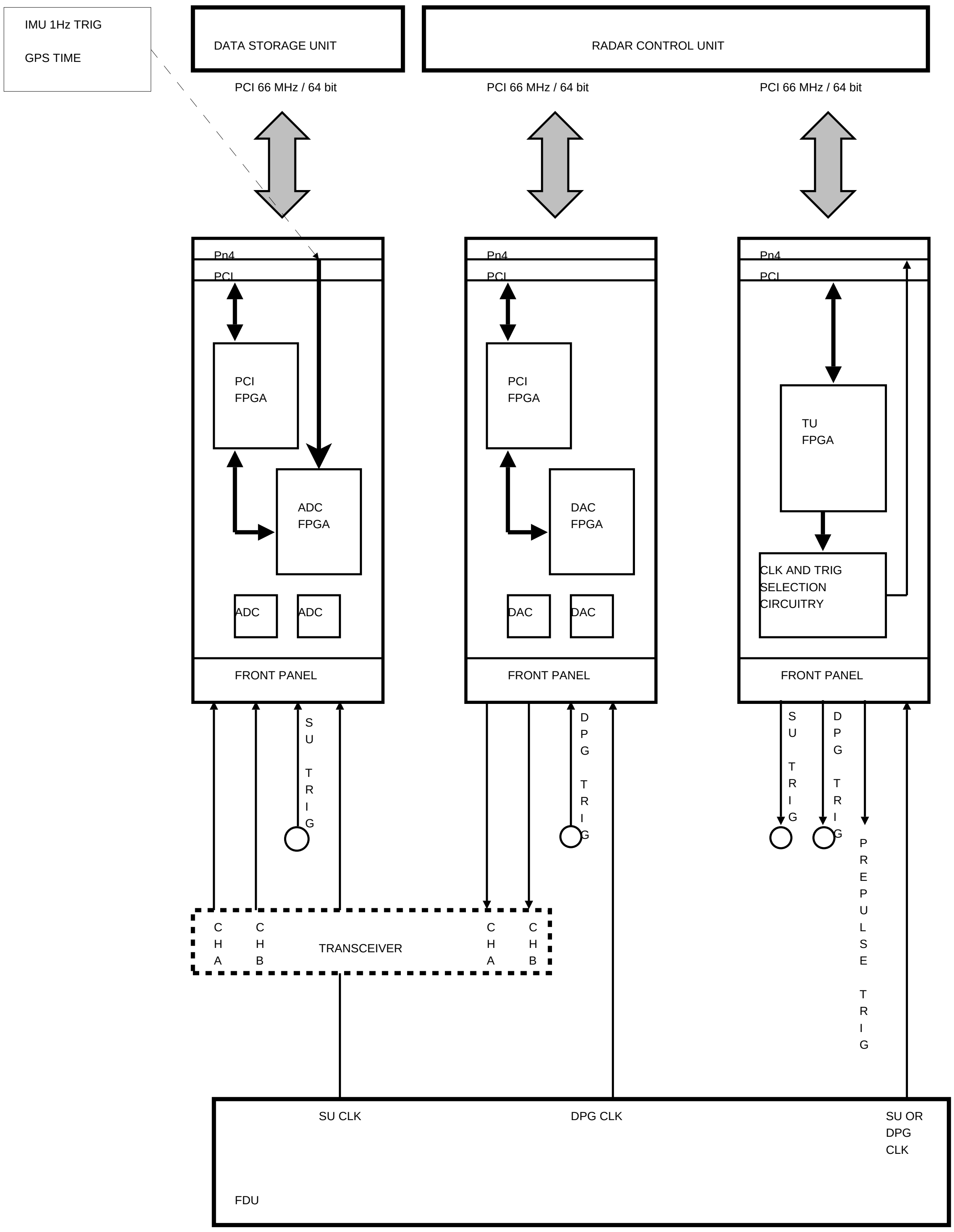

Figure 5.4: RDU System

Chapter 6

Firmware Implementation

This chapter describes the firmware that was implemented in the FPGAs present on the modules. The firmware was designed and written using Mentor Graphics HDL Designer, a tool specifically used for firmware development. Once written, the firmware was simulated with ModelSim. Once the design was simulated correctly, it was synthesized with Synplify. The output file, a type of netlist, was handed over to Quartus. Quartus is an Altera specific tool used for design, synthesis and fitting for Altera PLDs. Quartus was only used to fit the firmware.

The firmware for the modules was developed at Parsec, who provided the tools, licenses and support. Support was mostly required in the debugging stages. Several hardware problems were encountered as the modules had not been tested by Parsec. The sections to follow in this chapter will describe the firmware implemented in each module.

6.1 Digital Pulse Generator

The DPG consists of two FPGAs, two DACs, clock and trigger circuitry. The FPGA closest to the PMC connectors contains the PCI Core and wrapper functions. The DAC FPGA contains the DPG specific firmware.

6.1.1 PCI FPGA

Figure 6.1 illustrates the top level firmware for the PCI FPGA. Parsec purchased an Altera PCI Core and added wrapper functions to the Core. The Core provides an interface between the PCI Bus and the Local Bus. The following functionality was added to the PCI Core as wrapper functions:

- FIFO for master burst writes to PC memory
- Registers for master DMA setup
- Interface to the User Space

- Status and Command Registers.

The FIFO and registers to setup a DMA were not used in the DPG, as we do not require high speed data transfers from the DPG to PC memory. These registers are located in Base Address 0 (BAR0). The SU will however use these features. User Space is memory mapped and contains registers or memory that can be read or written. The registers or memory will be contained in the DAC FPGA.

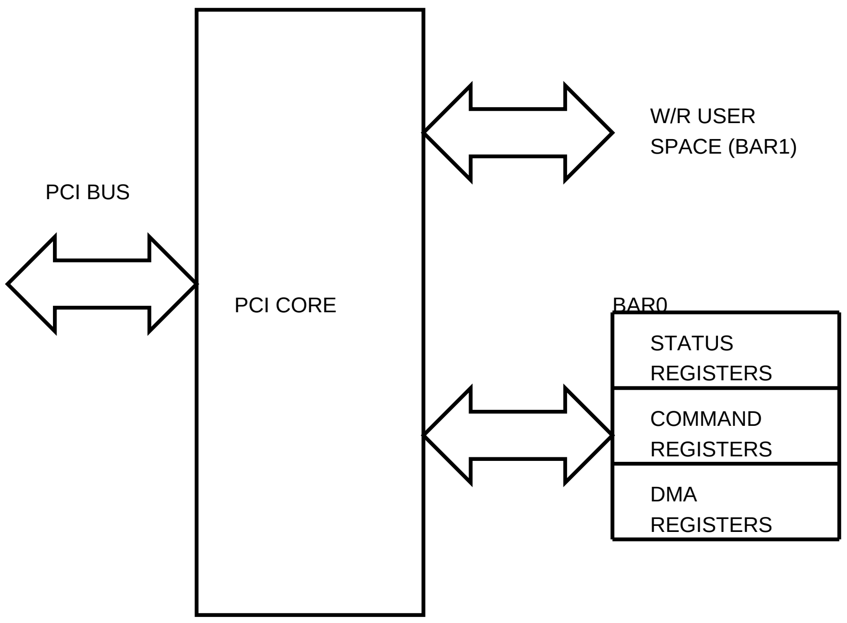

Figure 6.1: PM488 PCI FPGA firmware

6.1.2 DAC FPGA

The DAC FPGA firmware is illustrated in Figure 6.2 (note that it is only a top level illustration, and that not all signals and functionality have been shown, but it gives the reader a general idea of operation and data flow).

W/R User Space

W/R User Space is a local bus that comes from the PCI FPGA. It has the following signals:

- Read Target, which signals a PCI Read to user space
- Write Target, which signals a PCI Write to user space
- Address Target, which signals which user space location is to be accessed
- Data Bus, a 32-bit bus that carries data to or from Dual Port Memory or Control Registers.

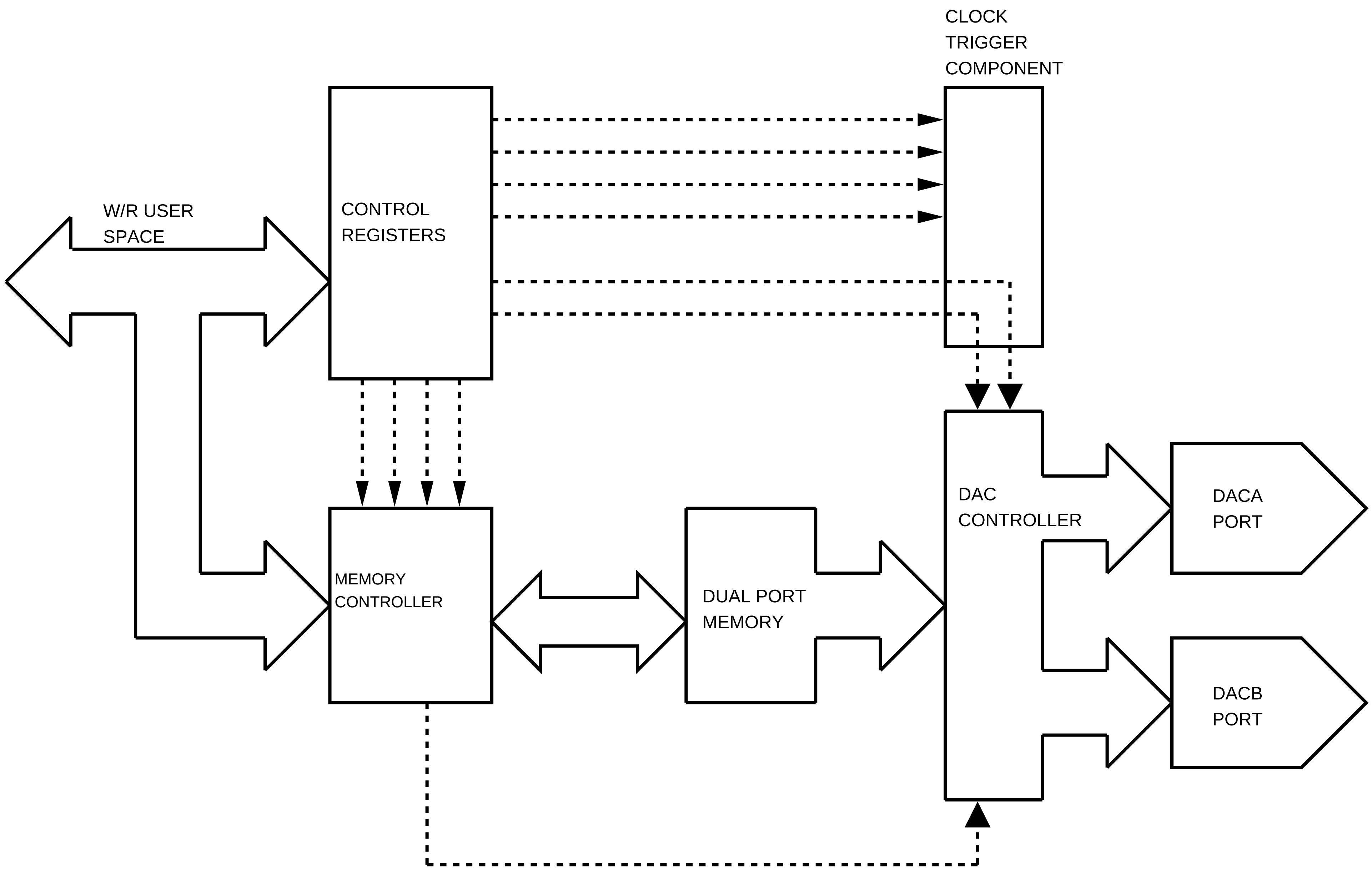

Figure 6.2: PM488 DAC FPGA Firmware

Control Registers

The Control Registers are a block or firmware component that contains registers which dictate the operation of the DAC FPGA. There are two registers present in the component: SamplingCntrlReg and TriggerReg. SamplingCntrlReg allows the user to select:

- DPG modes (Read/Write Mode, Test Mode and Sample Mode)
- Clock and triggers
- Channel enables
- DAC modes (clock division, zero bit stuffing and filtering).

TriggerReg is a register that can provide a software trigger for test purposes when written. Both of the registers are detailed in Appendix A.

Memory Controller

The Memory Controller controls the Dual Port Memory. A Dual Port Memory was implemented as two different clocks are used for reading and writing to memory. When bursting, the Dual Port Memory is read at the rate of the external clock. When samples are written or read via PCI, it occurs at the PCI clock rate.

DAC Controller

The DAC Controller decodes the 32-bit word that is clocked out from the Dual Port Memory to two 14-bit samples. Depending upon the values contained in the Control Register, a channel may be enabled or disabled. The Memory Controller also lets the DAC Controller know if data at the output of the Dual Port Memory is valid.

Clock and Trigger Selection

The clock and trigger selection firmware component is controlled by the Control Register block. Depending on the values of the SamplingCntrlReg at the next rising edge of a trigger, a new clock or trigger source may be selected.

Quartus place and route report

After synthesis, the firmware was placed and routed using Quartus. Quartus outputs text files, which describe the compilation specifics, the FPGA resources used and the expected behaviour of the logic. The following output parameters were taken from the reports:

- Total logic elements 818/4992 (16%)

- Total Pins used 320/333 (96 %)
- Total Memory bits 36352/49152 (73%)
- DACClk fmax 163.93 MHz

From the reports the DAC FPGA firmware used 16% of the total logic elements. The firmware was designed to be small and robust with limited features to cut down on development time. Limited funding did not allow a longer period of development time at Parsec. In addition, the more features added the greater the possibility of not meeting the firmware timing requirements. The Dual Port Memory took up 73% of the memory bits used in the FPGA, which allows a user to store up to 1024 samples per channel. A large amount of effort went into increasing the speed of the logic. The firmware can be clocked at a maximum of 163.93 MHz, which meets our firmware timing requirement of 150 MHz (DPG input clock of 150 MHz).

Logic Pipelining

Logic Pipelining was used in the DPG, SU and TU to increase clock speeds. The firmware was described using VHDL. The synthesis and routing processes converted the VHDL code into logic elements. The logic elements consist mostly of Look Up Tables (LUT) , registers and asynchronous logic. Along some of the logic paths between registers, large time delays would occur, from a chain of connected asynchronous logic. This decreases the overall clock speed. To maximize clock speed the delay between registers must be minimized, which can be done by breaking up long asynchronous logic paths using additional registers. This method of dividing up an asynchronous circuit using registers is called Pipelining, and it was used in several instances in the DAC FPGA firmware.

6.2 Sampling Unit

The Sampling Unit hardware is very similar to the DPG hardware. The only real difference is that two ADCs were used, instead of two DACs. The PCI FPGA contains the PCI Core and Parsec wrapper functions, wereas the ADC FPGA contains the SU specific firmware.

6.2.1 PCI FPGA

Figure 6.3 illustrates the firmware in the Altera ACEX PCI FPGA.

LBus

LBus is a high speed wide data bus. Data travels in only one direction, from the ADC FPGA to the PCI FPGA. LBus is the input to a 64-bit wide 512 deep FIFO. Not shown

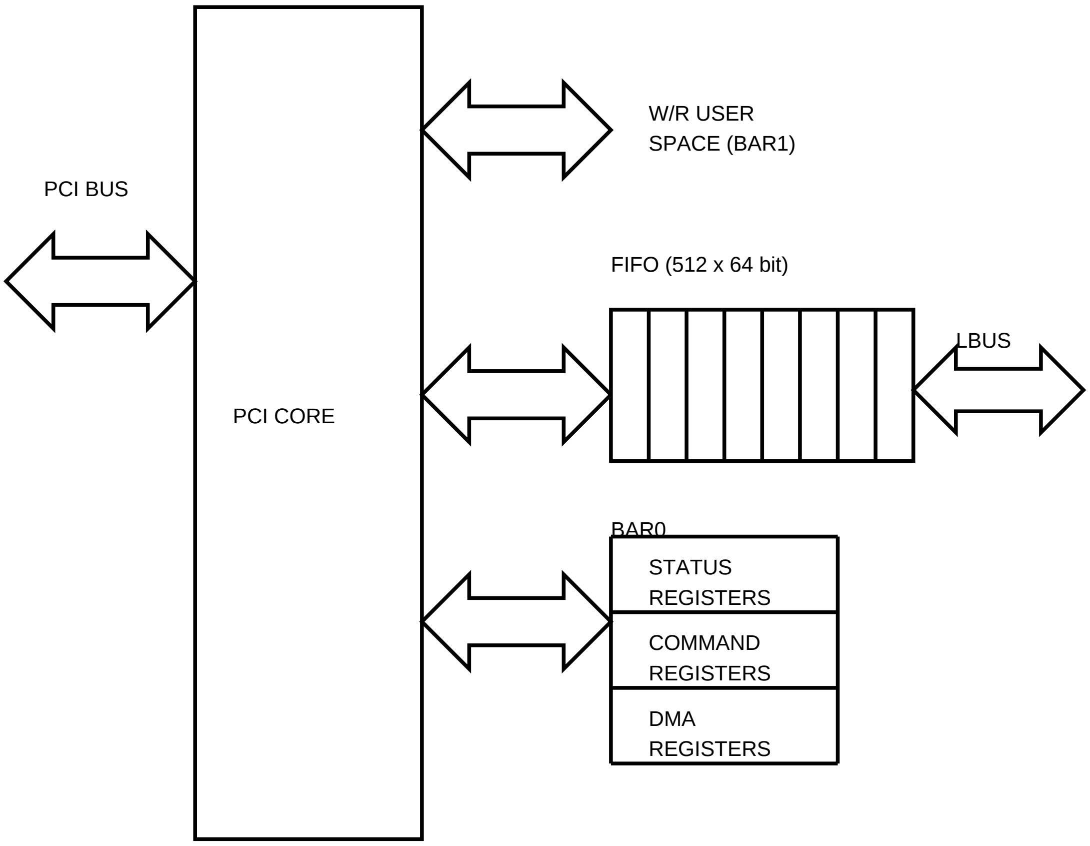

Figure 6.3: PM480 PCI FPGA firmware

in the diagram is the FIFO status and write signals. These signals are used for data flow control.

Data Flow

On each trigger, 16k samples are taken and a sampling packet header formed. The packet is stored in ZBT memory. Once all the samples have been stored, the packet is written to the PCI FPGA via LBus. The user sets up a DMA transfer by using the registers in BAR0. When the FIFO reaches a certain level, the PCI Core changes from a target device to a master device. This allows the PCI Core to do DMA transfers from the FIFO to PC memory. The size and number of DMA transfers is set by the user.

6.2.2 ADC FPGA

The SU ADC FPGA firmware is illustrated in Figure 6.4.

Control Registers

There are three registers present in the Control Register firmware component, which control the operation of the ADC FPGA. These registers are memory mapped to Base Address 1 (BAR1), and are read writable: SamplingCntrlReg, GeneralCntrlReg and TriggerReg.

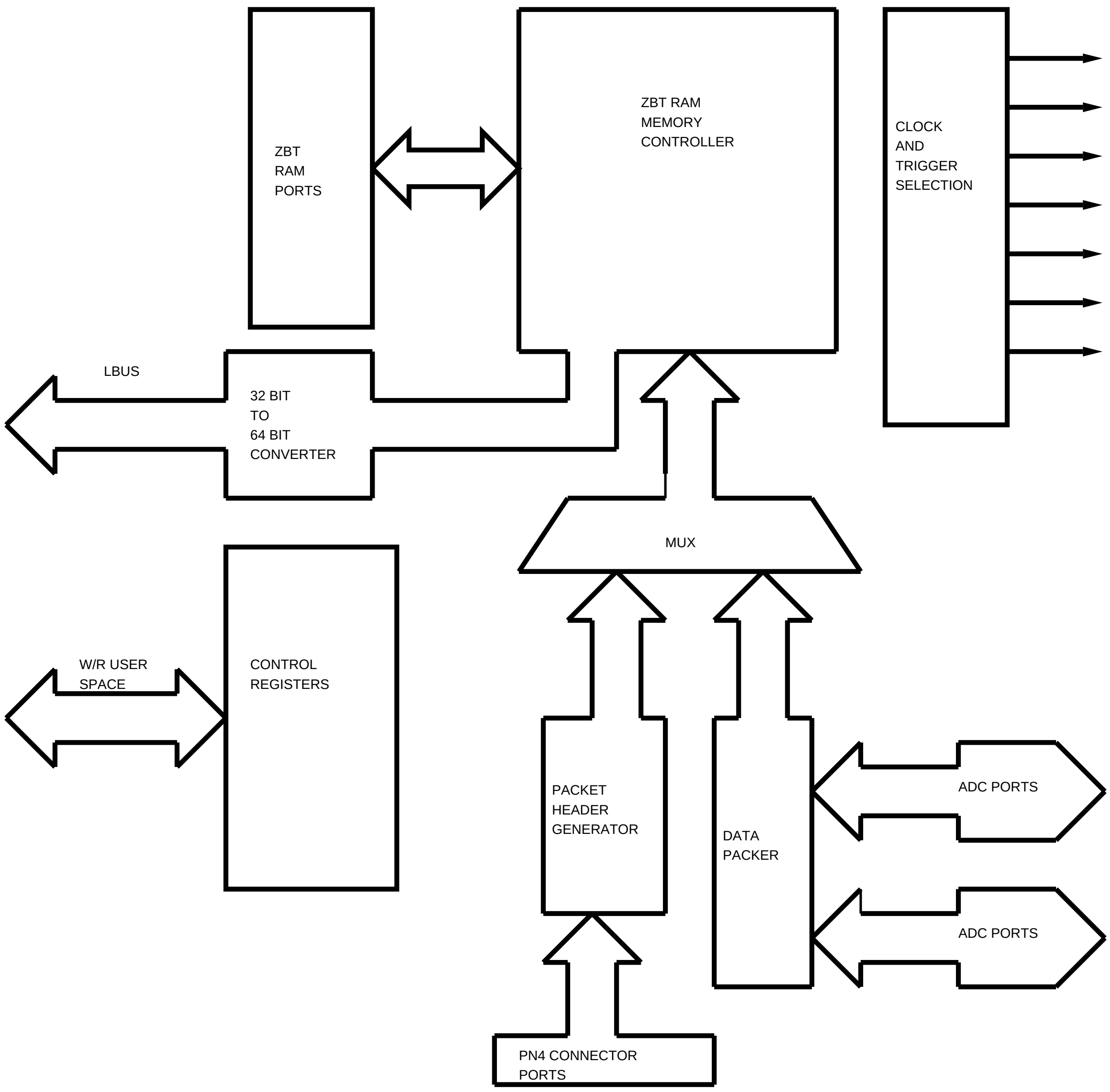

Figure 6.4: PM480 ADC FPGA firmware

SamplingCntrlReg allows the user to:

- Select the number of samples per PRI
- Enable or disable a ADC channel
- Enable or disable parity checking
- Select certain header elements to form the packet header.

GeneralCntrlReg allows the user to:

- Select SU Modes (Test or Sampling Mode)
- Control Clock and Trigger selection.

TriggerReg can be written to provide a software trigger to start sampling.

Data Packer

The Data Packer component latches the 14 bit ADC samples and packs then into a 32 bit word before writing to ZBT RAM.

Data Flow

Upon trigger assertion (software trigger or external trigger) the packet header is generated. The generation is described in the previous chapter. Once the header has been written to ZBT RAM, sampling commences. The 14-bit samples are packed into 32-bit words for easy storage. The sampled data and the header forms a packet. After the last sample has been written to ZBT memory, the packet is written to the FIFO in the PCI FPGA via LBus. Since the FIFO is smaller than 16k samples, we cannot continuously write to the FIFO, but have to stop writes once the FIFO is full. FIFO status signals are brought from the PCI FPGA to ADC FPGA to stop or start writes on LBus.

Quartus place and route report

Quartus outputs text files, which describe the compilation specifics, FPGA resources used and the expected behaviour of the logic. The Quartus report displayed the following:

- Total Logic Elements 2188/4992 (43%)
- Total Pins 313/333 (99%)
- Total Memory bits 0/49152 (0%)
- ADCClk fmax 114.94 MHz

The fimware developed for the ADC FPGA is considerably larger than the firmware developed for the DAC FPGA. A total of 43 % of logic elements were used in the ADC FPGA, while in the DAC FPGA, only 16 % of total logic elements were used. The ADC FPGA firmware was more complex due to a larger set of features and larger data path. The ADCClk timing requirement was met. The ADCs are to sample at 105 MHz, therefore the firmware timing requirement should be equivalent or greater than 105 MHz.

Multiple Clock Domains

The DPG, SU and TU all have multiple clock domains present in each FPGA. The clock domains exist due to different input clocks. The PCI write and reads to registers are synchronous to the PCI clock. The sampled data is synchronous to the sampling clock. This creates two clock domains in an FPGA. Signals or data travelling between the two domains can become misinterpreted or corrupted due to meta-stability. To overcome meta-stability the following was implemented in the FPGAs:

- A signal coming from a different clock domain was treated as an asynchronous signal. The signal was buffered using a number of registers and the rising edge or falling edge detected
- Data was moved between two clock domains using a dual clocking FIFO. The data written to the FIFO was synchronous to the write clock and the reader was synchronous to the read clock.

6.3 Timing Unit

The TU has one Altera APEX FPGA on the PMC module. The PCI Core and the TU specific firmware were written for just one FPGA. Due to the PCI Core, wrapper functions and application specific firmware being placed in one FPGA, the number of logic elements used was large, 98% of total elements. Illustrated in Figure 6.5 is the firmware implemented in the APEX FPGA.

6.3.1 Control Registers

This firmware component, which is similar to the SU and DPG Control Register component, contains five registers:

1. The General Control Register allows the user to disable or enable the output triggers and select a clock source
2. The Frequency Register allows the user to select the PRF. The PRF is calculated by dividing the input clock frequency by the value stored in the Frequency Register

3. The Pre-Pulse Register sets the position of the Pre-Pulse trigger
4. The DPG Register sets the position of the DPG trigger
5. The ADC Register sets the position of the ADC trigger.

6.3.2 Trigger Generator

At power-up or after every PRI the Frequency Register is read and the value is loaded into a down-counter. The values in the Pre-Pulse, DPG and ADC Registers are loaded into comparators. After the counter and comparators have been loaded, the counter begins to count down. When the counter reaches a value stored in one of the comparators, a pulse, is created and sent out from the FPGA to ECL circuitry.

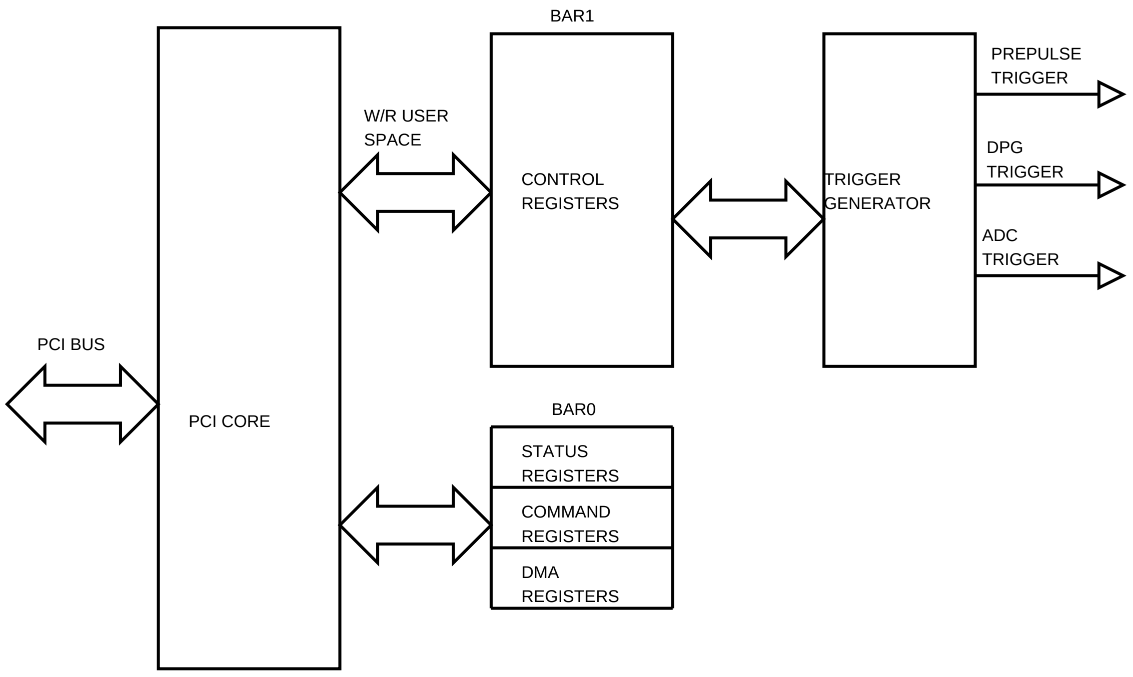

Figure 6.5: PM440 Firmware

6.3.3 Quartus place and route

The Quartus place and route report displayed the following:

- Total Logic Elements 4107/4160 (98%)
- Total Pins 181/246 (73%)
- Total Memory bits 25600/53248 (48%)
- TUClk 170.3 MHz

The total elements used in the APEX FPGA was large. The APEX FPGA contained the PCI Core, Parsec wrapper functions and the TU specific firmware. If further TU functionality was required, the firmware might not have fitted. The highest frequency digital clock used in the RDU is the DPG input clock (150 MHz). It was thus decided that the maximum timing requirement for the TU should be equivalent to or greater than the DPG clock. The timing requirement was met.

6.4 Conclusions

A description of the firmware written for the SU, DPG and TU has been given in this chapter. Altera wrote the PCI Core firmware and Parsec wrote wrapper functions, which added further functionality to the PCI Core. The PCI Core and Parsec wrapper functions were placed in three of the FPGAs. The DPG, SU and TU specific firmware were designed and developed by the author and placed in the remaining FPGAs.

Chapter 7

Firmware Testing

In this chapter the procedure used to test the firmware developed for the SU, DPG and TU, to verify correct operation, is described. Since the hardware had not been tested by Parsec, most of the debugging during the testing stage involved searching for hardware associated problems.

Testing of the modules was done over a duration of several months and was limited to the equipment available at the time. Testing progressed as equipment became available.

7.1 Sampling Unit Testing

7.1.1 Test Limitations

Testing of the SU at Parsec was not completed due to the lack of test equipment. To fully test the SU, two signal generators are required; Parsec however only had one high quality signal generator, which was used as the input signal. The SU was clocked using its on-board 30 MHz clock. To test the SU correctly it should have been clocked around 105 MHz, as this was our required input clock frequency.

After leaving Parsec, the SU could not be tested for some months, due to a lack of cables and a compact PCI rack or PMC to PCI adapter board. The cables were needed to connect a clock and input signal to the SU, whereas a rack or adapter board was needed to communicate between a PC and the PM480. Once the cables arrived and a PCI adapter board had been purchased, a second signal generator was still required. The 100 MHz TU on-board clock was used to clock the SU, but the RRSG lacked a 64-bit/66 MHz PCI machine to test data rates.

The SU test procedure was limited by the equipment available. Writes and reads to registers were tested at Parsec; the results were successful, and will not be discussed in the test procedure, because only significant tests and results will be described. The data path continuity will be tested and the emulated ADC will be tested. It is recommended that data rates be tested when a 64-bit/66 MHz PCI machine is purchased.

7.1.2 Test Procedure

To ensure the firmware is working correctly the data path from the output of the ADC to the input of the PCI Core input should operate correctly. No bits must be corrupted and there must be no lost samples. The easiest method of checking bit corruption and data path continuity was to generate a test ramp before the data packer in the ADC FPGA firmware. This would act as though virtual ADCs are sampling a saw tooth waveform. Simple up-counters were implemented. The user can enable this test mode, writing to the Sampling Control register.

The hardware modification made at Parsec, to emulate an ADC sampling at twice the normal rate, had to be tested. To verify correct operation, the output SNR and ENOB was calculated for varying input frequencies. The output of a high quality signal generator was used for the input to the SU. UCT had only one of these signal generators at the time, so the TU was used to clock the SU at 100 MHz. This resulted in the SU sampling at 200 MSPS.

7.1.3 Debugging

Several problems were encountered once the firmware was placed in the ADC FPGA and the PCI FPGA. Initially, a test ramp was generated in the ADC FPGA to emulate the ADC sampling a ramp. A DMA from the SU to PC memory returned all zeros. A test ramp was then placed after the ZBT RAM controller and proved successful. It was then concluded that a problem existed with the ZBT RAM memory component. After probing the device with an Oscilloscope it was discovered that the ZBT clock pin was unconnected. Soldering a wire from the clock pin to the correct FPGA pin (via a resistor) solved the problem.

After further inspection of the ramp data, it was discovered that certain bits were being corrupted. The corruption was taking place in three possible areas: between the ADC FPGA and PCI FPGA when a packet transfer was taking place, between the ADC FPGA and the ZBT RAM, or between the PCI FPGA and the PCI bus.

A test ramp was placed inside the PCI FPGA. It was noted that no corruption occurred. So the problem occurred either between the FPGAs or between the ZBT and the ADC FPGA.

Consequently, a test ramp was placed after the ZBT RAM controller and before the ADC FPGA interface between the FPGAs. Transferring data showed that specific bits in a 64-bit word were being corrupted. It was then concluded that the interface between the FPGAs was causing the problem. However, it was not certain, which interface was causing the problem, i.e. the ADC FPGA interface or the PCI FPGA interface. After consulting with the staff at Parsec, they suggested to alter the interface in the PCI FPGA significantly, and place Fast IO registers at the pins between the PCI FPGA and ADC FPGA. This decreased the setup times and increased the hold times. After further testing, this modification was concluded to have solved the bit corruption.

7.1.4 Results

Figure 7.1 illustrates a ramp created in the ADC FPGA before the data packer firmware component. The ramp data was extracted from the SU and plotted with Matlab. A 14-bit up counter produced the ramp shown. The counter was clocked at 105 MHz, and 8192 samples were taken. Note that the ramp is linear with no curves or glitches. If a glitch had occurred, it would have indicated a bit corruption. This testing of the data path was conducted at Parsec.

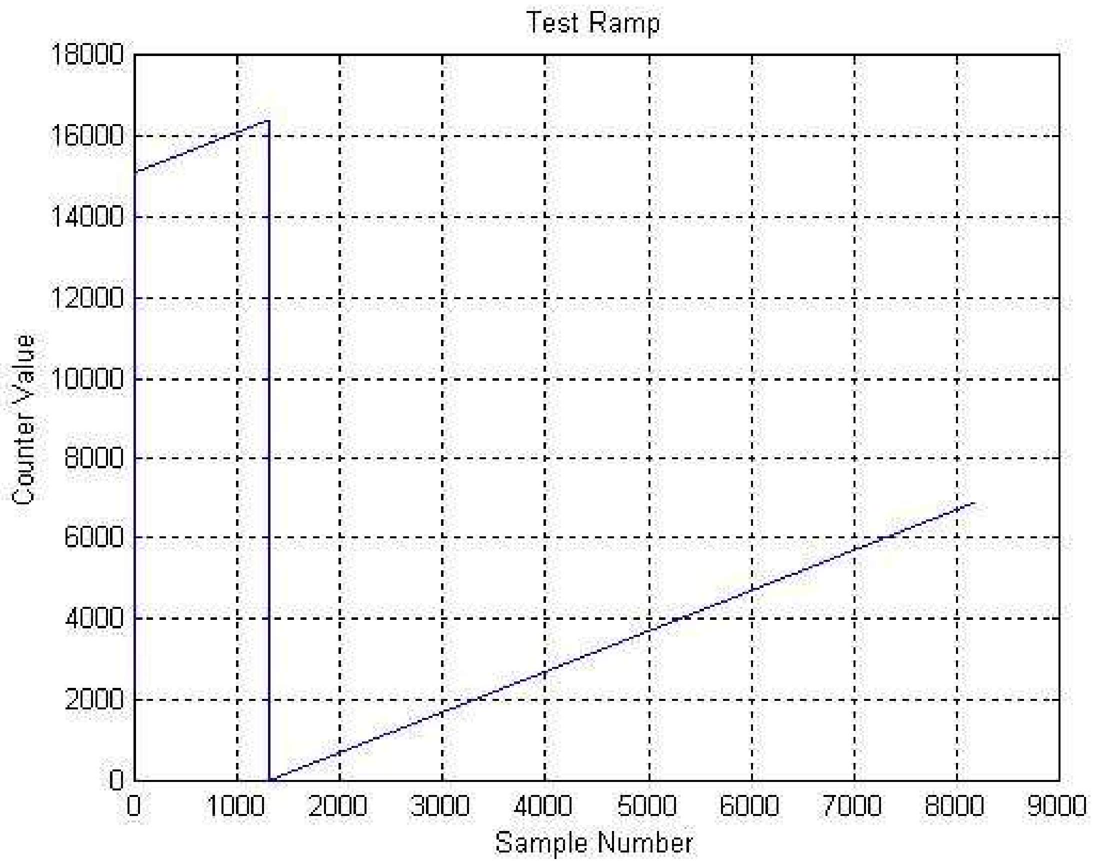

Figure 7.1: Test Ramp

The SNR and ENOB results of sampling varying sinusoids from 10 MHz to 190 MHz are given in Table 7.1. The calculated average ENOB for the SU is 9-bits, thereby satisfying one of our user requirements. FFTs were taken to view the output spectrum of the SU, which was done for various input frequencies. The results are displayed in Appendix B.

Table 7.1: SNR calculations

Frequency(MHz)	**SNR-Emulated ADC (dB)**	**SNR-ADC 1 (dB)**	**SNR-ADC 2 (dB)**
10	67	69	68
20	60	64	68
30	59	67	68
40	61	69	69
50	61	72	64
60	62	66	64
70	60	65	66
80	65	64	64
90	66	67	67
110	65	67	68
120	57	63	65
130	58	64	64
140	56	63	64
150	56	62	68
160	60	62	63
170	61	64	65
180	61	65	65
190	61	65	64
Average SNR	61	65	66
ENOB	9-bits	10-bits	10-bits

7.2 Digital Pulse Generator Testing

7.2.1 Test Limitations

The DPG was fully tested at Parsec by using a signal generator, spectrum analyzer, oscilloscope and compact PCI rack, all of which were available at Parsec. The control registers were tested individually, by writing and reading to registers; data read back was complete, and the correct DPG mode and control changes were visible. Data path continuity tests and SFDR measurements are discussed in the test procedure.

7.2.2 Test Procedure

To prove that the firmware is operating correctly, the data path from the PCI Core to the DAC output ports in the DAC FPGA should not corrupt bits or drop samples. Firstly then, the PCI Core to DAC FPGA data path must be checked. Writing data to the dual port memory and reading the correct data back will indicate correct operation.

Once the interfaces between the FPGAs are concluded to be bug free, the data path from the dual port memory to the firmware DAC output ports can be tested. Writing a sinusoid to the dual port memory (32-bits x 1024) and triggering the DPG every 1024 clock cycles, produced a continuous sinusoid to be written out of the FPGA to the two DACs. The DAC outputs should be probed with an oscilloscope, and if there are no glitches and a perfect sinusoid is viewed, it would mean that no bits have been corrupted, or samples dropped.

Further, one could view the sinusoidal output spectrum with a Spectrum Analyser. If the SFDR measured using the Spectrum Analyser is compared to the SFDR given in the DAC data sheets, and if these are more or less equal, we can conclude that the firmware is working correctly.

7.2.3 Debugging

When it was confirmed that the firmware did indeed simulate correctly and timing constraints were met, the module was programmed via JTAG. It was noted that no problems were occurring along the data path between the PCI Bus and the Dual Port Memory. The data written and read back proved to be correct.

To test the path between the dual port memory and the firmware DAC output ports, a stored sinusoid was used as a waveform. Monitoring the outputs of the DAC with an oscilloscope showed the DACs output was not working correctly, as the DAC outputs were constantly set to ground. The input pins to the DAC were checked with an oscilloscope and proved to be in the correct configuration. After consulting with a number of Parsec's staff it was found that the ac-coupling transformer at the output of the DACs was incorrectly connected. A hardware modification was thus made and this solved the problem.

7.2.4 Results

Sinusoids of varying frequencies from 1 MHz to 50 MHz were loaded into the DPG. The DPG was clocked at 150 MHz. The sinusoids were monitored with an Oscilloscope and a Spectrum Analyser. All sinusoids viewed with an Oscilloscope proved to be glitch free and perfect in shape. The spectrum of the sinusoids was displayed and the SFDR measured. The SFDR measurements obtained are illustrated in Table 7.2, and plots taken using an Aligent spectrum analyzer are displayed in Appendix C. Table 7.4 illustrates the SFDR taken from the DAC datasheet [18].

Table 7.2: DPG SFDR measurements

Frequency (MHz)	**SFDR (dB)**
1	83
5	83
10	81
20	74
30	70
40	62
50	71

When comparing Table 7.2 and Table 7.4, it can be seen that the measured values are similar. In some cases, the measured SFDR is better than the datasheet SFDR. Viewing

Table 7.4: Analog Devices SFDR measurements [18]

Frequency (MHz)	**SFDR (dB)**
5.02	82
20.02	75
50.02	65

varying sinusoids with an oscilloscope and measuring the SFDR, and comparing these values to the datasheet specifications, it is concluded that the firmware is working correctly.

7.3 Timing Unit Testing

7.3.1 Test Limitations

Complete testing of the TU was not possible. It was discovered, that an unknown problem exists with the PM440 hardware. Communication between a PC and the TU is not possible. Once Parsec has revised the hardware design or proved the problem no longer exists, testing can continue.

The TU firmware was altered to exclude the PCI Core. Trigger time jitter measurements were made with a spectrum analyzer but this proved to be unsuccessful. It was later discovered that a timing interval analyzer was required to perform the necessary tests. A time interval analyzer was not located. The test procedure described below should be done once the TU hardware bug has been solved.

7.3.2 Test Procedure

The data path should be tested to ensure that the firmware is operating correctly. The data path to be tested runs from the PCI Core to the firmware component Trigger Generator. If written data is read back correctly from the registers present in the Control Register firmware component, then we can conclude that part of the data path (PCI Core to the firmware component Control Registers) is operating correctly. Monitoring the PECL trigger outputs with an oscilloscope, and changing the PRF and position of triggers, by sending a token to the TU, confirms that the data path is operating correctly.

Trigger timing jitter measurements should be made and compared with Parsec's claimed jitter specification in their PM440 datasheet.

7.3.3 Debugging

Programming the PROM on the TU for the first time was unsuccessful. A JTAG hardware issue was located. The problem was discovered to be an incorrect resistor value placed in the circuit comprising the JTAG chain. The old resistor was taken out and a new one placed on the board. After this modification, the JTAG chain functioned correctly.

Placing the Card in a PCI slot and powering up the PC caused the PC to hang during BIOS setup. The firmware, pin assignments and constraints were re-checked, and everything seemed to be in order. Since the same PCI Core has been loaded into previous designs, and had worked correctly in these cases, the problem was most likely a hardware bug. Parsec has been notified about the problem and will contact the RRSG once the problem has been solved, but it is not known when the problem will be solved.

The card was then connected to a power supply to test the clock and trigger circuitry. The PCI Core firmware component and Parsec wrapper functions were deleted from the existing firmware. Probing the clock and trigger outputs confirmed that the PECL circuitry was working correctly.

Trigger timing jitter measurements were made with a Spectrum Analyser, and the phase noise plot was captured on disk. Parameters taken from the phase noise plot were entered into a Matlab program, which calculated the timing jitter from phase noise plots [19, 20].

7.3.4 Results

The JTAG chain, FPGA and PECL circuitry are all working correctly, but there is an issue with the interface to the PCI Bus. Testing of the data path cannot be done until Parsec has solved the problem, and will thus not be included in this dissertation. Results of the ModelSim simulation of the firmware will however be given. The ModelSim simulation, simulates tokens written to the TU. It is shown in Figure 7.2 that the tokens sent change frequency and position of the trigger pulses.

Refer to Figure 7.2, the following sequence of events occur:

1. Initially at startup no trigger pulses are output
2. The Pre-Pulse, DPG and ADC registers are written
3. Finally the Frequency register is written. When the triggers are enabled and the Frequency register contains a value greater than the values present in the Pre-Pulse, DPG and ADC register, the output triggers become active
4. The frequency and position of the pulses are changed by writing to the registers. It appears from Figure 7.2 that the triggers change frequency and position.

The timing jitter measurements have been recorded in Table 7.6. The jitter decreases as frequency increases, but the jitter should remain more or less the same in a divider circuit: “Unlike ripple counters, phase noise does not accumulate with each stage in synchronous counters. Phase noise at the output of a synchronous counter is independent of the number of stages and consists only of the noise of its clock along with the noise of the last stage” [21].

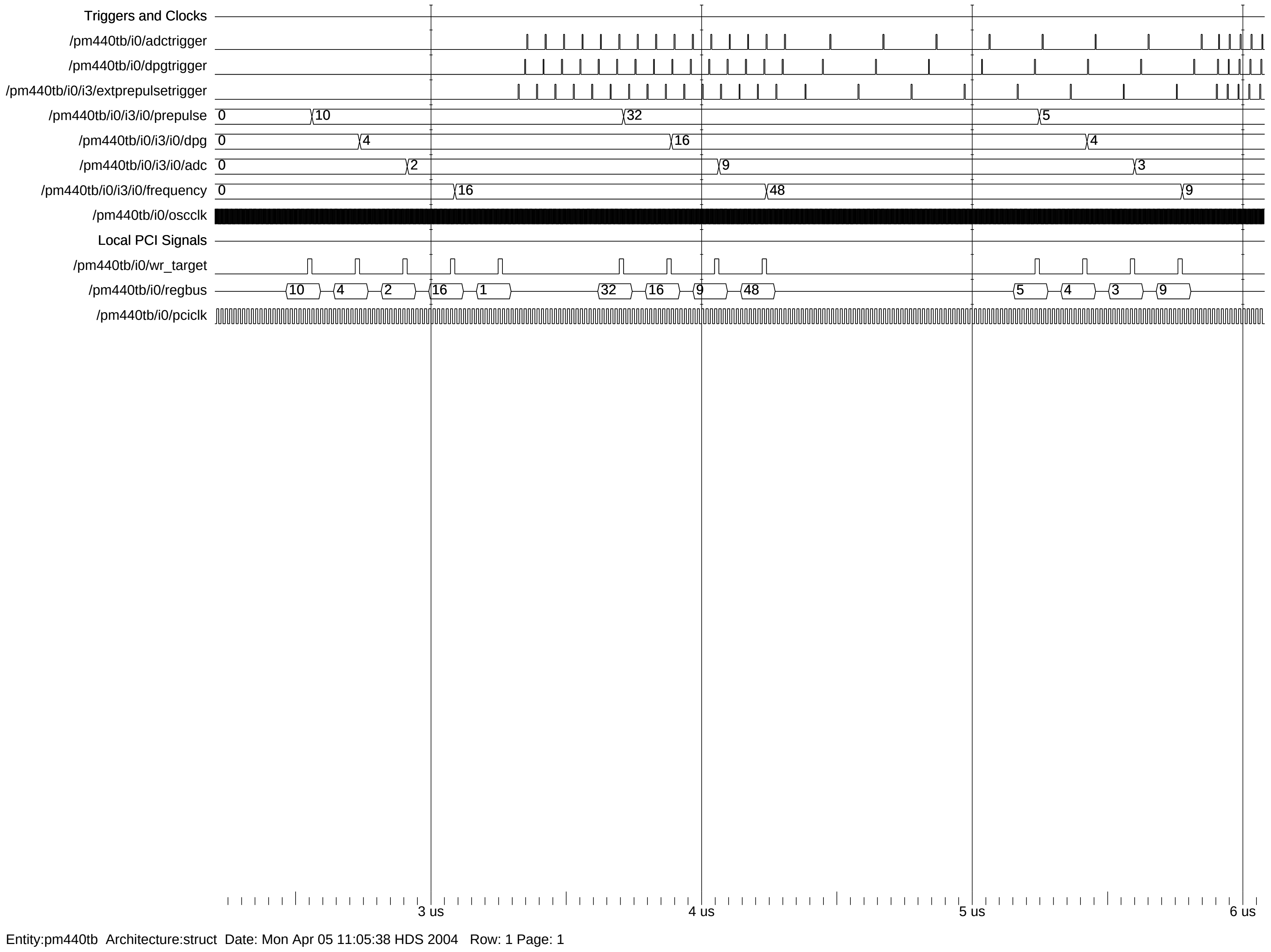

Figure 7.2: TU Simulation Results

Therefore the most probable reason for jitter increasing as frequency decreases must be the method of measurement. A new method of measurement would involve a time-interval analyzer instead of a spectrum analyzer, but it was not possible to locate a time interval analyzer.

Table 7.6: Trigger Frequency vs. Jitter

Frequency (MHz)	**Jitter (ps)**
100	6
25	14
6.25	74
1.56	334
0.39	1303
0.097	5440

7.4 Conclusions

The DPG and SU are operating correctly. Performance tests were done for the SU to examine whether IF sampling, using two ADC emulating a single ADC, produces a fair SNR. The calculated ENOB produced 9 bits, and satisfies our user requirement. Performance tests for the DPG showed the measured SFDR to be similar to the datasheet SFDR. The TU firmware simulates correctly, and the results confirm correct operation. The PECL clock and trigger circuitry of the TU were tested and viewed with an oscilloscope, which showed correct operation. However, when placing the TU in a PC, the machine does not get past the BIOS setup and freezes. Parsec will inform the RRSG when this problem is solved.

Chapter 8

Conclusions and Future Recommendations

This chapter discusses the conclusions drawn based upon the results gained from the previous chapter on the Firmware Testing. It is shown that the requirements have been achieved by purchasing generic Data Acquisition hardware and implementing SASARII specific firmware.

8.1 Sampling Unit

The Firmware Testing chapter showed that our data path worked correctly, by generating a test ramp in the ADC FPGA close to the ADC input ports in the FPGA. Transferring the ramp to PC memory showed that no bit corruption or loss of samples occurred. The digital circuitry of the SU functioned correctly. The hardware modification made to emulate a single ADC of sampling at 210 MSPS was tested. The performance measurements SNR and ENOB were calculated, and an average ENOB of 9-bits showed sampling performance greater than 8-bits.
The requirements have been satisfied:

- The 66 MHz / 64 bit PCI bus interface will handle the expected data rate
- IF Sampling at 210 MSPS satisfies the Nyquist criterion
- Control registers allow the user to select packet header element, number of samples per packet, trigger and clock sources and more
- ENOB of 9-bits of resolution is more than required
- PECL clock and trigger circuitry ensures low jitter.

8.2 Digital Pulse Generator

The data path was tested as described in the Firmware Testing chapter. From the results obtained, it is concluded that the firmware is indeed working correctly. During testing a sinusoid was loaded into the DPG, and testing with an oscilloscope showed no glitches, but a perfect sinusoid. The SFDR of the DACs was taken using a spectrum analyser and compared to the Analog Devices DAC datasheet. The measured SFDR was similar to the datasheet SFDR. Purchasing the PM488 and programming SASARII specific firmware satisfied the requirements:

- A theoretical bandwidth of 160 MHz is possible
- 1024 x 32-bit words can be stored to be output on the rising edge of the external trigger. This amounts to 1024 x 14-bit samples per channel
- PECL clock and trigger circuitry ensures low jitter.

8.3 Timing Unit

The results from the Firmware testing chapter showed that the firmware does simulate correctly, but due to a possible hardware bug the hardware could not be tested properly. During BIOS setup the card would cause the PC, into which the TU was placed, to freeze. It is not known what causes this problem, but Parsec has been notified, and will inform the RRSG when the problem has been solved. Firmware without the PCI Core was placed in the FPGA to test the PECL clock and trigger circuitry and it worked correctly.

8.4 Future Recommendations

At the time of writting up the dissertation, a lack of test equipment and a TU hardware related issue prevented the proper testing of the RDU. In this section we will briefly layout recommendations and the expected outcome of further TU testing.

8.4.1 Timing Unit Testing

The TU hardware or the Parsec wrapper functions may undergo revision and updating, to resolve the hardware problem. The TU firmware will have to be updated to allow for the changes to be made. Somebody knowledgeable in VHDL and the SASARII project should make these changes.

The following tests should be performed:

- All registers must be written and read to test for data path continuity. If the value written to a register is corrupted when read back, the firmware or hardware must be debugged

- All register writes must change TU operation correctly. Writing to the GeneralCntrlReg should change the TU mode of operation as expected. Writing to FrequencyReg should change the system PRF. Writing to PrePulseReg, DPGReg or ADCReg should change the trigger timing position

- Trigger PRF and position changes must be monitored with an oscilloscope or logic analyzer

- Trigger jitter measurements must be taken to ensure the TU operates to specification. The RRSG must locate a time interval analyzer and perform the necessary measurements.

Further TU testing should prove to be simple if equipment is available. The firmware architecture has been laid out in a simple manner and commented on extensively. The only expected difficulty would be in locating a time interval analyzer to test trigger jitter.

8.4.2 Radar Digital Unit testing

Testing of the RDU will only be possible when the SASARII Frequency Distribution Unit, Radar Frequency Unit and Data Storage Unit become available. The testing of the RDU integrated in the SASARII system is not within the scope of this dissertation.

Bibliography

[1] David C. Buchanan, “Performance of an IF sampling ADC in receiver applications”, Application Note, Analog Devices, Rev. 0

[2] M. Nguyen, J. Devlin, J. Whitting, “Investigation of different digital implementation methods for the TIGER radar receiver”, Paper, Department of Electronic Engineering, La Trobe University Bundoora

[3] Leon G. Melkonian, “Dynamic Specification for Sampling A/D Converters”, Application Note, National Semiconductor, May 1991

[4] Mark Sauerwald, “Designing with High-Speed Analog-to-Digital Converters”, Application Note, National Semiconductor, May 1988

[5] “Fundamentals of Sampled Data Systems”, Application Note, Analog Devices

[6] Brad Brannon, “Aperture Uncertainty and ADC System Performance”, Application Note, Analog Devices

[7] Darren Coetzer, “SASARII Transmitter Design Considerations”, SASARII Design Document, Radar and Remote Sensing Group, University of Cape Town, October 2003

[8] Ajmal I. Mohungoo, “SASARII Reciever Design Document”, SASARII Deisgn Document, Radar and Remote Sensing Group, University of Cape Town, August 2003

[9] Darren Coetzer, Thomas Bennett, “SASARII Frequency Distribution Unit Design Considerations”, SASARII Design Document, Radar and Remote Sensing Group, University of Cape Town, October 2003

[10] Gage Applied Technologies, http://www.gage-applied.com/, website

[11] Acqiris, http://www.acqiris.com/, website

[12] Strategic Test, http://www.strategic-test.com/, website

[13] Parsec, http://www.parsec.co.za/, website

[14] Peralex, http://www.peralex.com/, website, Jan 2003

[15] Altera, “ACEX 1K Programmable Logic Device Family”, Datasheet, revision 3.4, May 2003

[16] Altera, ”PCI MegaCore Function User Guide”, User Guide, version 2.0.0, August 2001

[17] Micron Technology, “Pipelined ZBT SRAM”, Datasheet, Rev. 2/02

[18] Analog Devices, “AD9772A Product datasheet”, Datasheet, Rev. A

[19] Neil Roberts, “Phase Noise and Jitter - A primer for Digital Designers”, Paper ZARLINK semiconductor, July 2003

[20] Joseph V Adler, “Clock-source jitter: A clear understanding aids oscillator selection”, Paper, Vectron International, Feburary 1999

[21] Ken Kundert, “Predicting the Phase Noise of PLL-Based Frequency Synthesizers”, Paper, The Designers Guide, November 2003

[22] Envisat, www.evisat.esa.int/, website, Oct 2004

[23] John Curlander, Robert McDonough, “Synthetic Aperture Radar: Systems and Signal Processing”, Wiley, New York, 1991

Appendix A

Registers

Digital Pulse Generator Registers

This section shows the registers present in the DPG that are memory mapped to Base Address 1.

SamplingCntrlReg

Bit Number	31	30	(29:28)	27
Function	ChannelB_en	ChannelA_en	Sample_Read_Test_Mode 00 = Sample Mode 01 = W/R Mode 10 = Test Mode	Clock Sel 0 = Int Osc 1 = Ext Osc

Bit Number	26:25	24:23	(22:21)	20:19
Function	Trigger_Sel 00 = Ext Trig 01 = Pn4 Trig 10 = SW Trig 11 = Test Trig	MODA_Sel Always set to 00	MODB_Sel Always set to 00	DIVA_Sel Always set to 00

Bit Number	18:17	16	15:10	9:0
Function	DIVB_Sel Always set to 00	TestTrigger_en	Nothing	Burst Size 0 = Int Osc

Sample_Read_Test_Mode Notes:

- Test Mode = DAC outputs a sinusoid stored in ROM
- W/R Mem Mode = Memory has been allocated in the DAC FPGA. One can write 1024x32 bit words to RAM. 14 bit samples must be stored in positions (13:0) and (29:16). **PCI reads/writes to memory must be 32 bit single cycle**
- Sample Mode = in this mode at every trigger assertion the memory pointer starts at address zero and cycles down to location (Burst Size). The stored wave form is output the DACs

Trigger_Sel Notes:

- SW Trig = Software trigger enable. Writing to TriggerReg(0) allows one to trigger the module via PCI

- Test Trig = When Trigger_Sel = “11” and TestTriggerEn = ’1’ the module is trigger by an internal trigger that is asserted every 1024 clock cycles

TriggerReg

Bit Number	31:1	0
Function	Nothing	SwTrigger

Sampling Unit Registers

This section shows the registers present in the SU that are memory mapped to Base Address 1.

SamplingCntrlReg

Bit Number	31	30	29	28:23
Function	ChannelB_en	ChannelA_en	ParityBit_en	Nothing

Bit Number	22:17	16:0
Function	Header_Sel	SampleBlockSize

GeneralCntrlReg

Bit Number	31	30:29	28	27	26:25
Function	Pn4IO_en	Nothing	Sample_Test_Mode 0 = Sample Mode 1 = Test Mode	Clock_Sel 0 = Int Clk 1 = Ext Clk	Trigger_Sel 00 = Ext Trig 01 = Pn4 Trig 10 = Sw Trig

Bit Number	24:1	0
Function	Nothing	Sampling Status 0 = idle 1 = sampling

TriggerReg

Bit Number	31:1	0
Function	Nothing	SwTrigger

Timing Unit Registers

This section shows the registers present in the DPG that are memory mapped to Base Address 1.

GeneralCntrlReg

Bit Number	6:5	4	3
Function	ClockOut_Sel 00 = OscClk 01 = ExtClk 10 = TTLClk	Pn4PrePulseTrig_Sel	ExtPrePulseTrig_Sel

Bit Number	2	1	0
Function	DPGTrigger_Sel	ADCTrigger_Sel	Triggers_en

ClockOut_Sel Notes:

- OscClk = The on board Oscillator Clk is output on the 3 front panel connectors
- ExtClk = The input external clock is output on the 3 front panel connectors
- TTLClk = The TTL signal from the output of the FPGA is output on the 3 front panel connectors

FrequencyReg

Bit Number	31:0
Function	Counter value

The value loaded in FrequencyReg is counted down to zero. The time taken for the values to reach zero, sets the system PRI.

PrePulseReg

Bit Number	31:0
Function	Comparator value

When the value stored in PrePulseReg is equivalent to the value in FrequencyReg a pulse is output the TU.

DPGReg

Bit Number	31:0
Function	Comparator value

When the value stored in DPGReg is equivalent to the value in FrequencyReg a pulse is output the TU.

ADCReg

Bit Number	31:0
Function	Comparator value

When the value stored in ADCReg is equivalent to the value in FrequencyReg a pulse is output the TU.

Appendix B

FFTs of Sampled Data by the SU

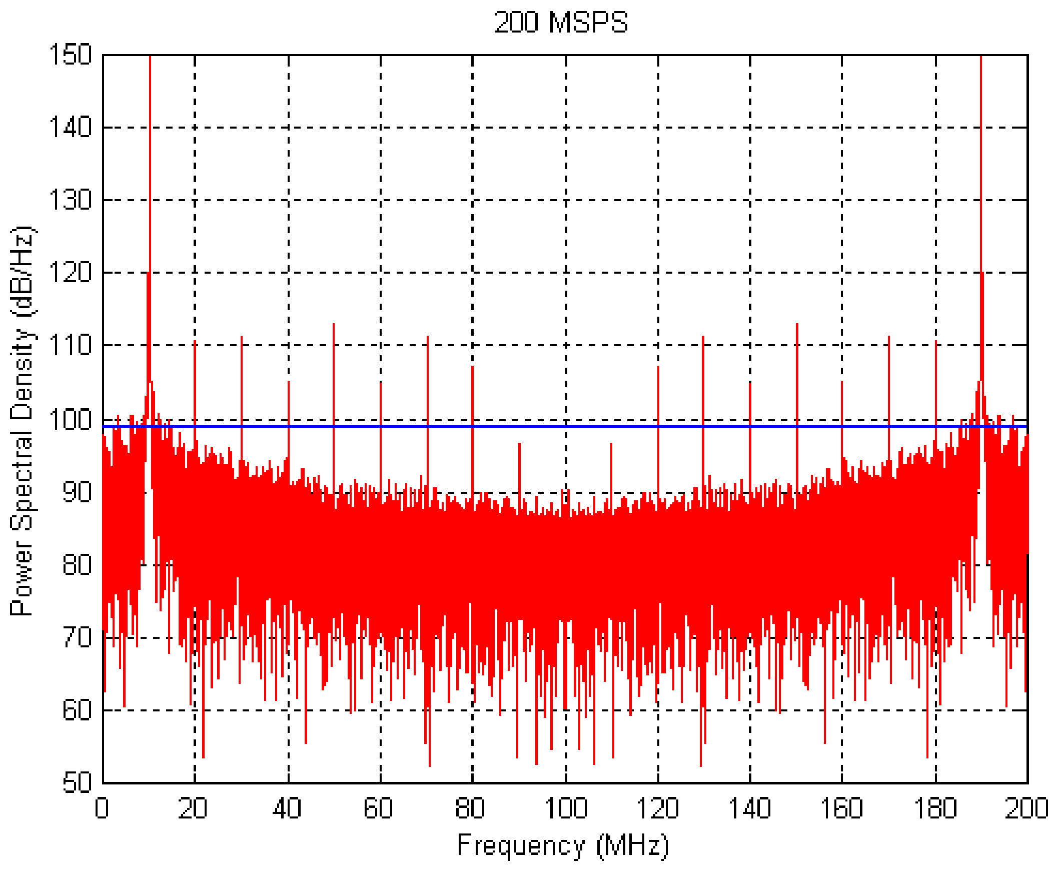

Figure 1: Spectrum for input frequency of 10 MHz

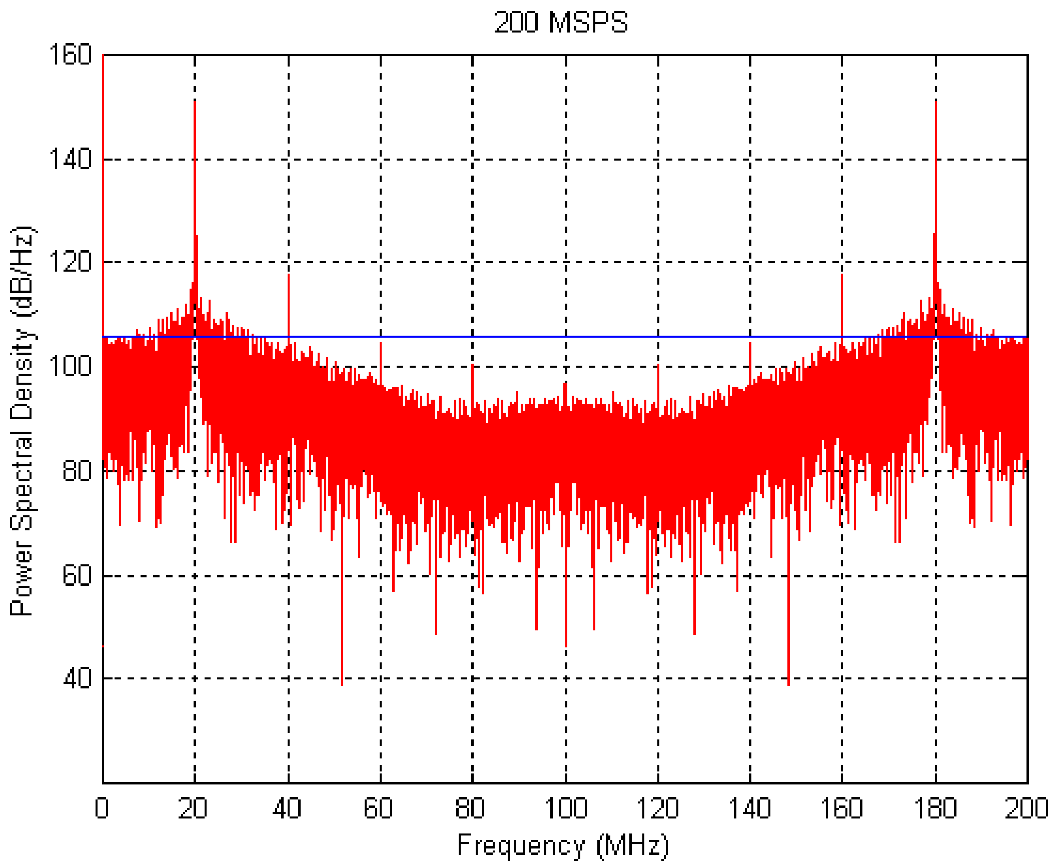

Figure 2: Spectrum for input frequency of 20 MHz

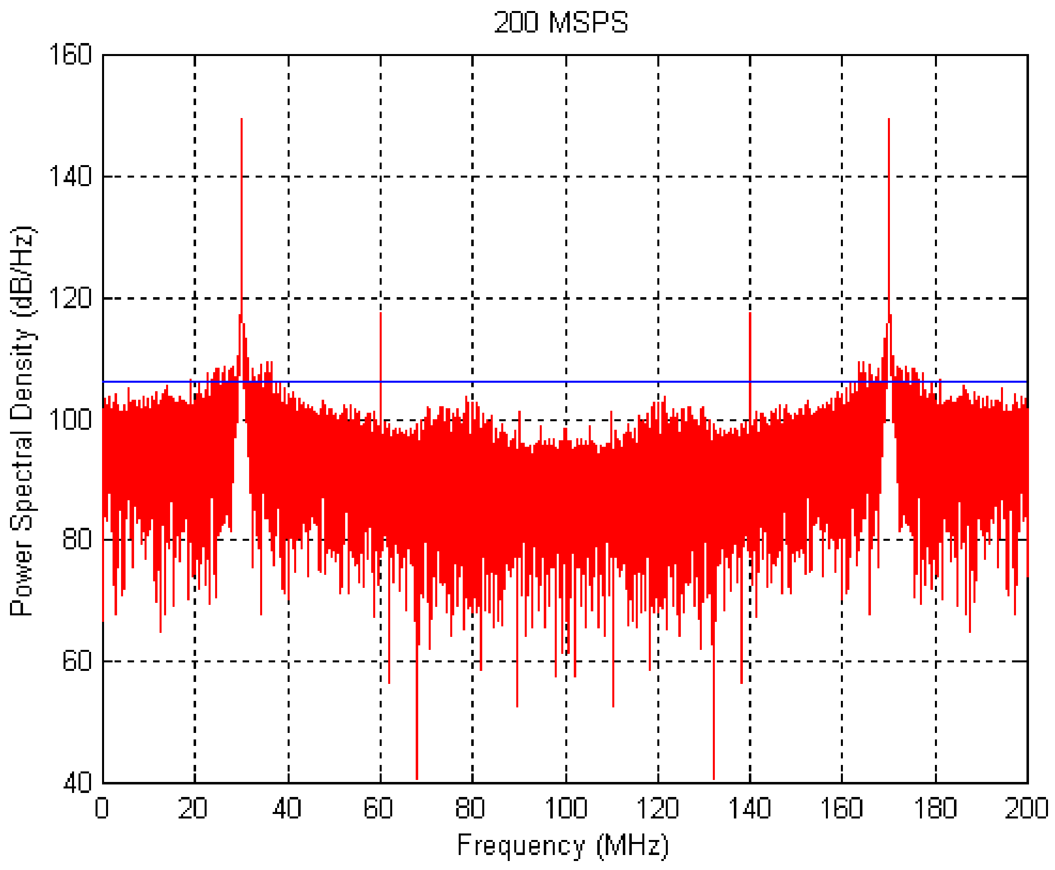

Figure 3: Spectrum for input frequency of 30 MHz

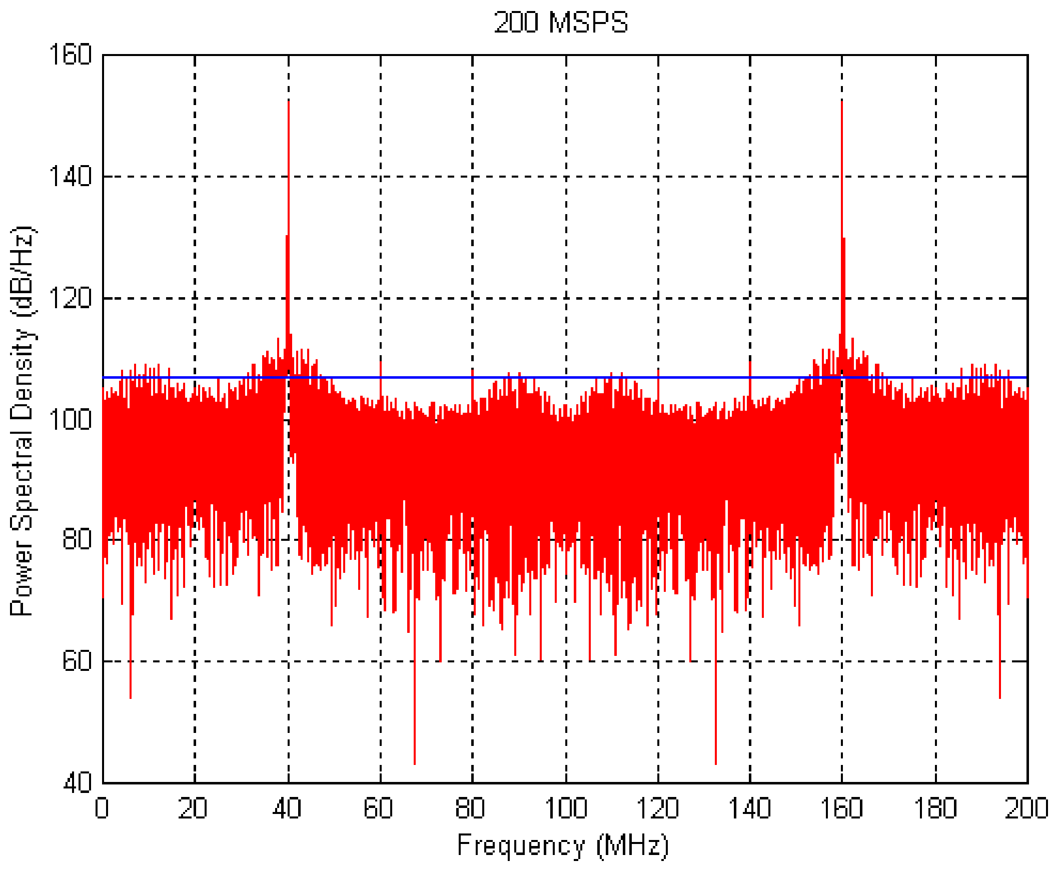

Figure 4: Spectrum for input frequency of 40 MHz

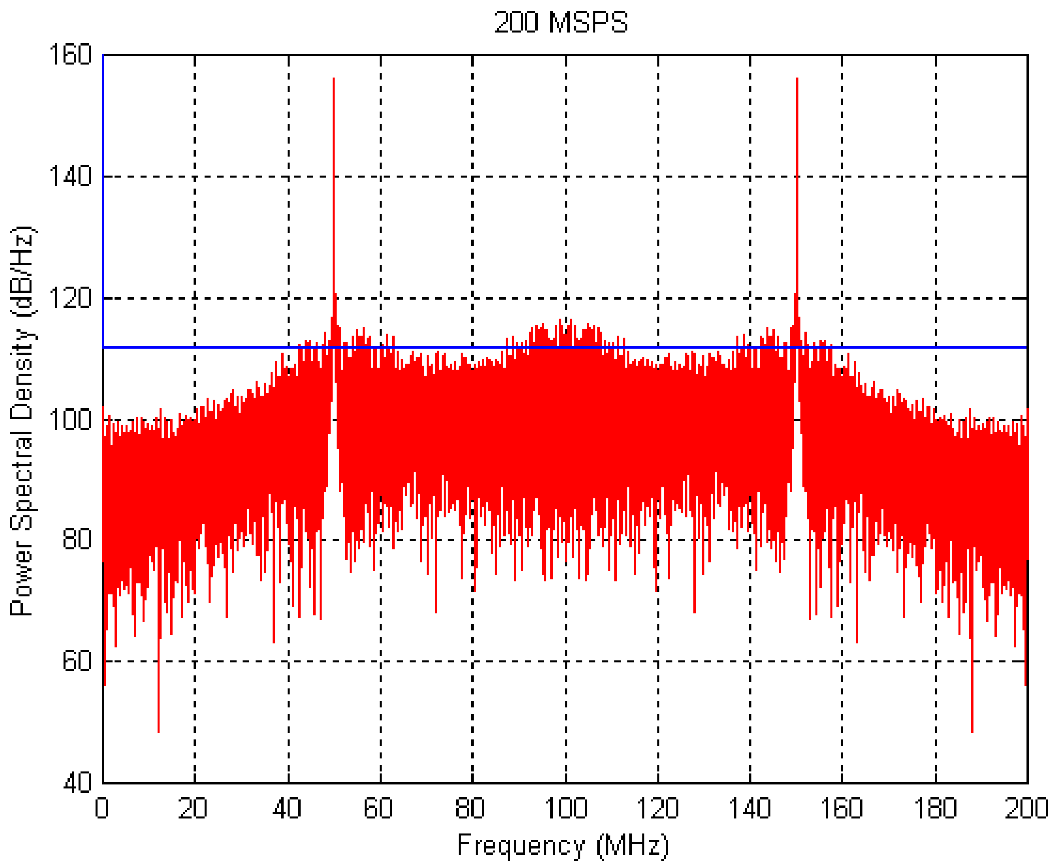

Figure 5: Spectrum for input frequency of 50 MHz

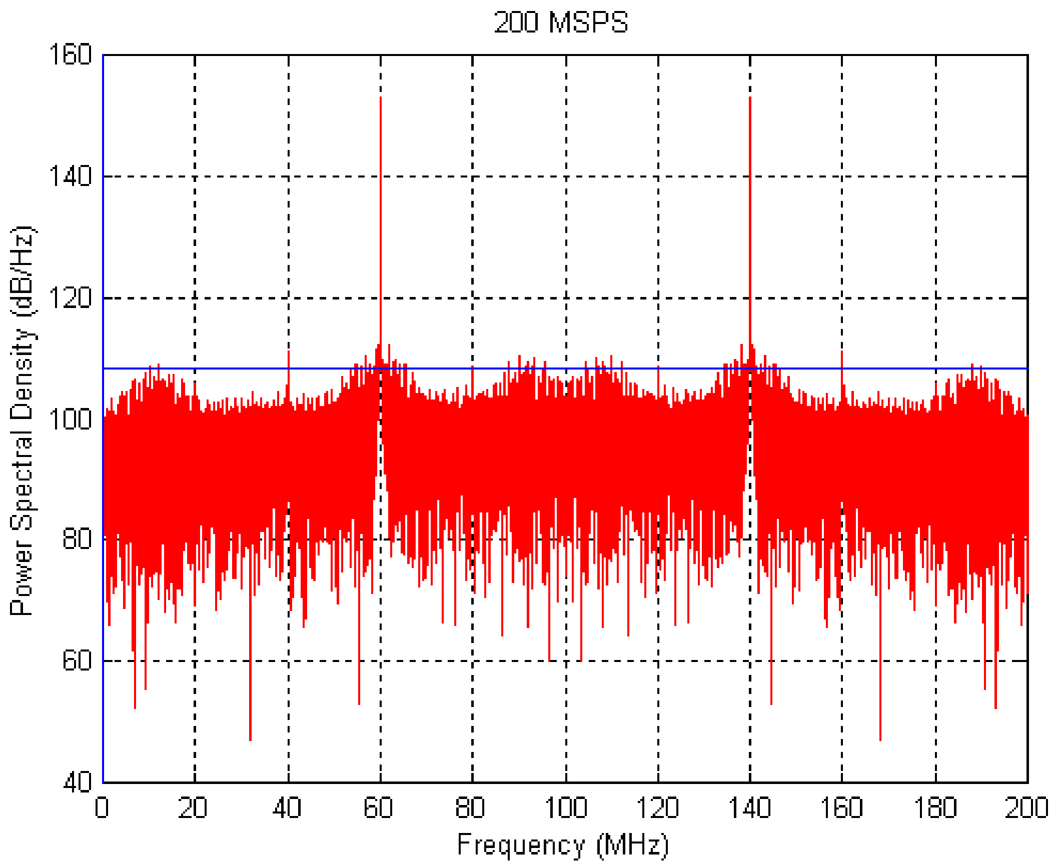

Figure 6: Spectrum for input frequency of 60 MHz

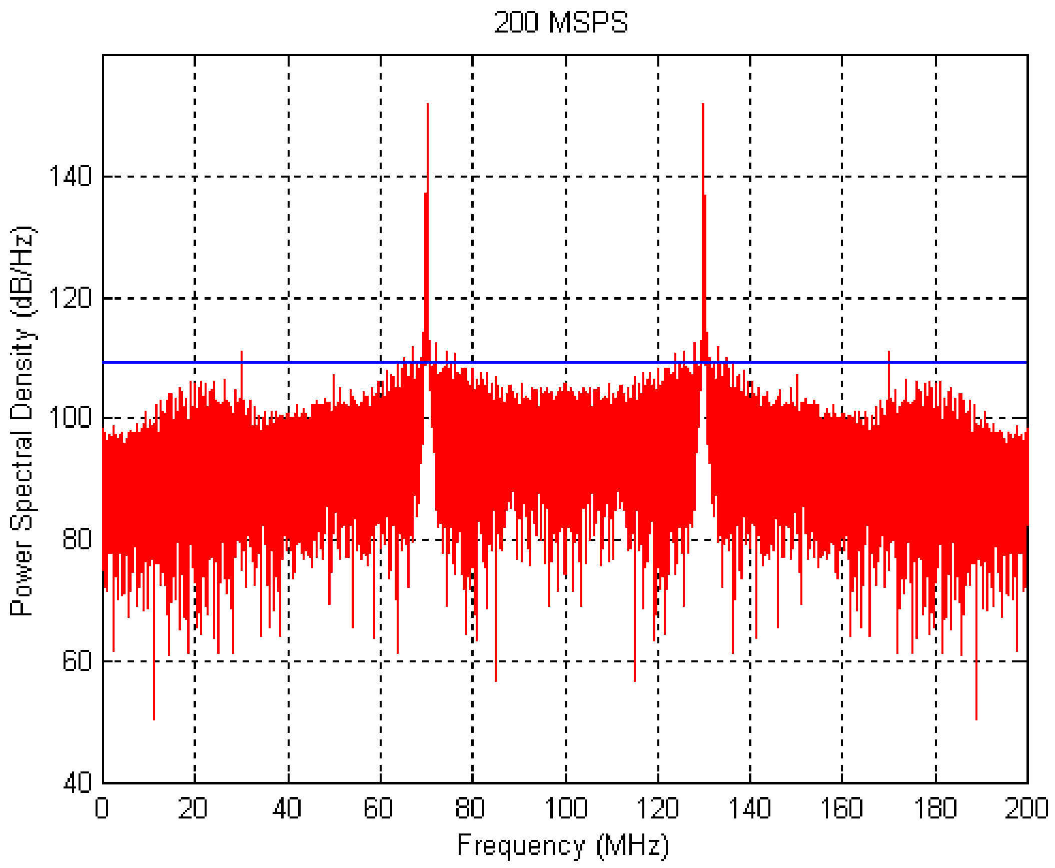

Figure 7: Spectrum for input frequency of 70 MHz

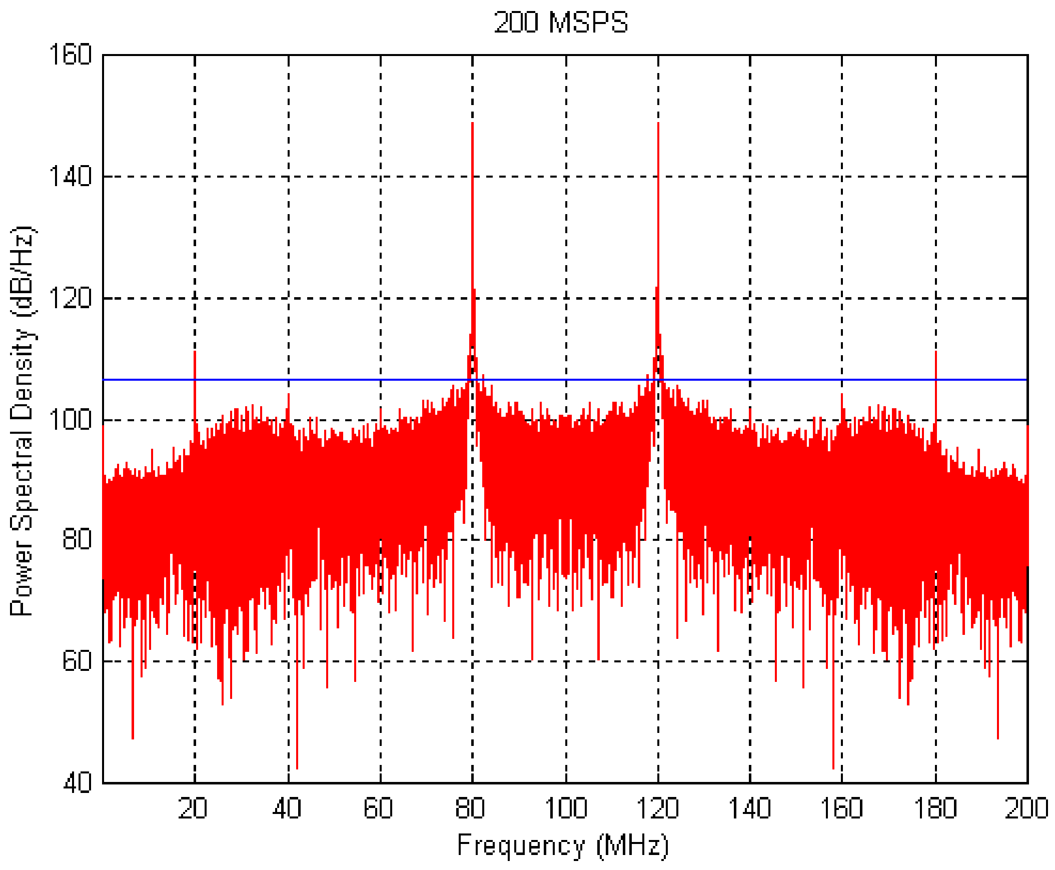

Figure 8: Spectrum for input frequency of 80 MHz

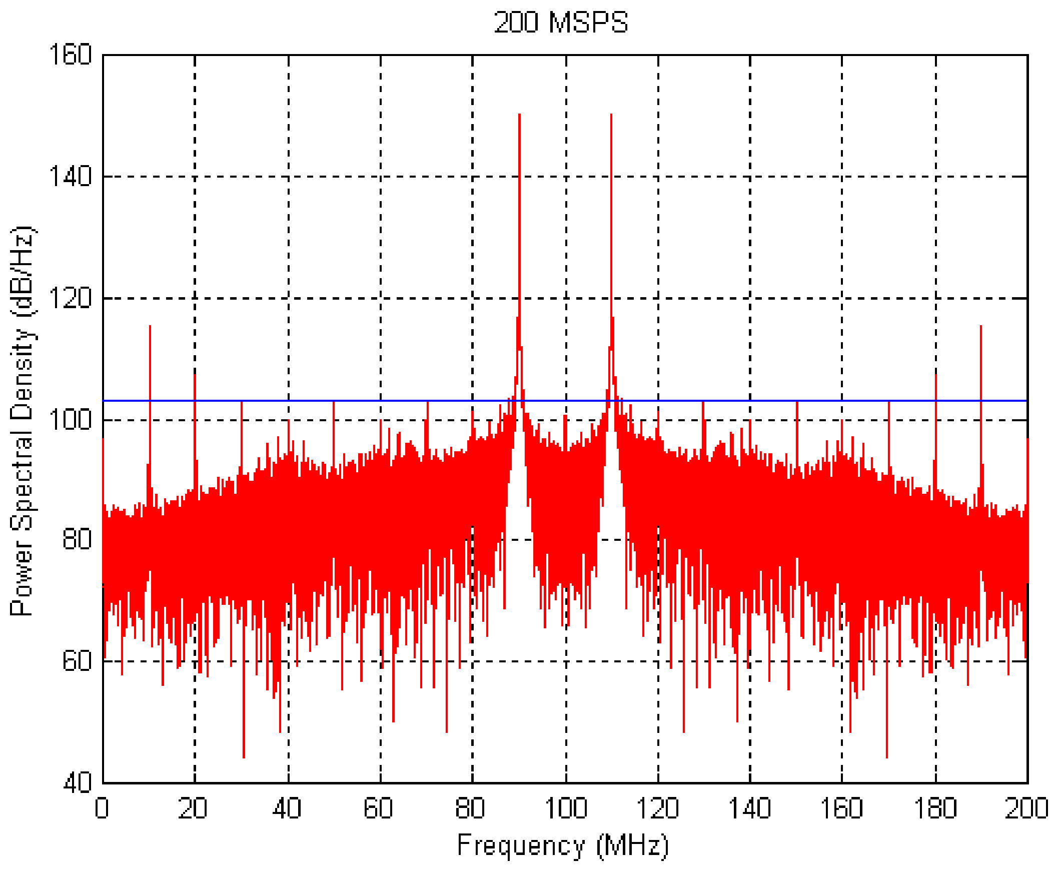

Figure 9: Spectrum for input frequency of 90 MHz

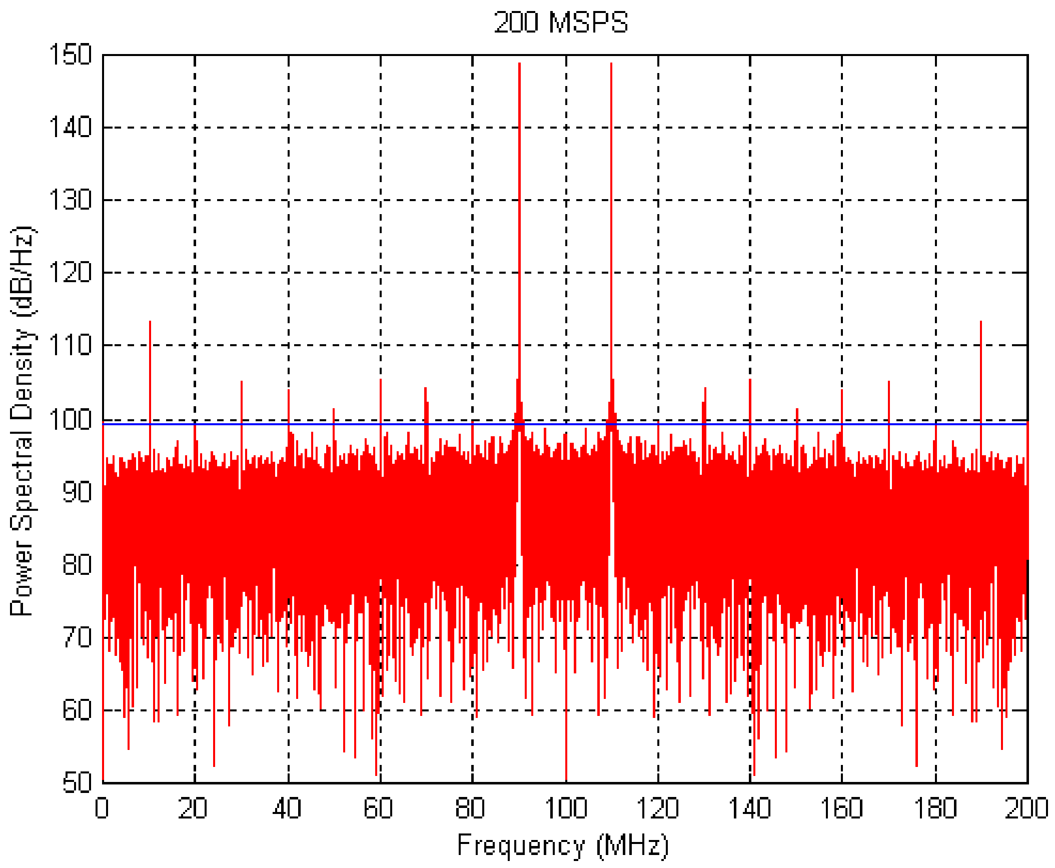

Figure 10: Spectrum for input frequency of 110 MHz

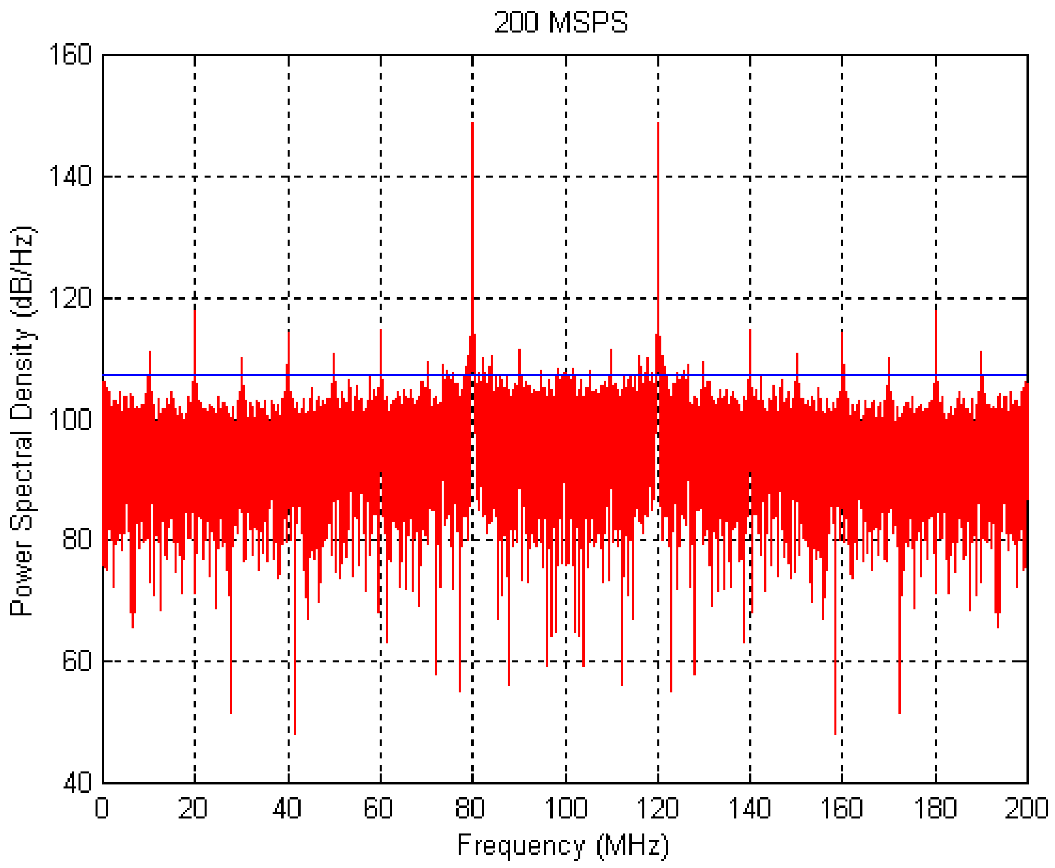

Figure 11: Spectrum for input frequency of 120 MHz

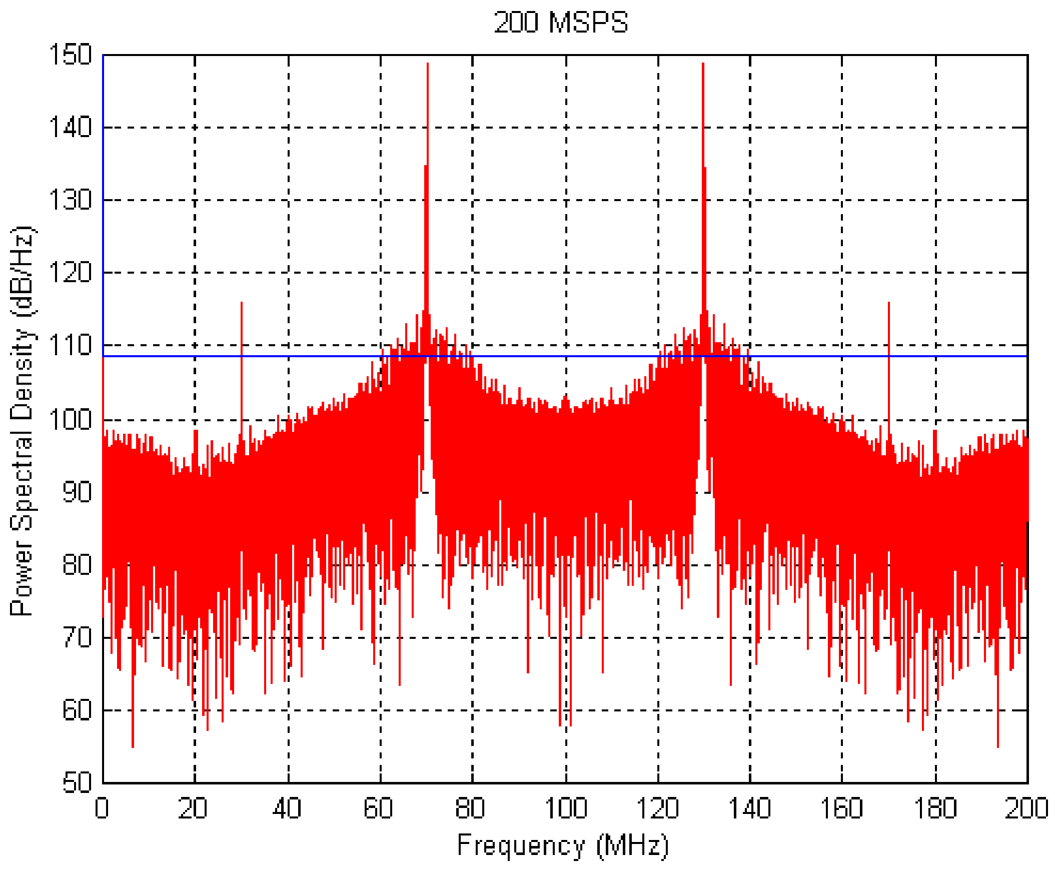

Figure 12: Spectrum for input frequency of 130 MHz

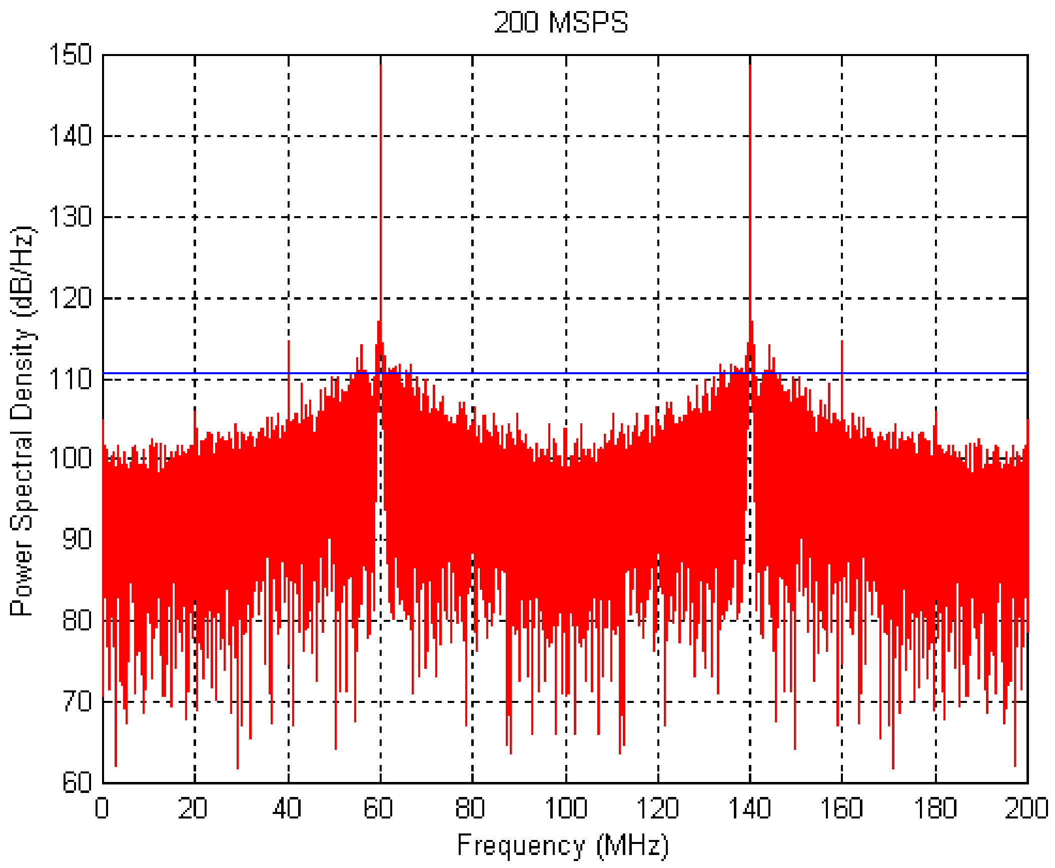

Figure 13: Spectrum for input frequency of 140 MHz

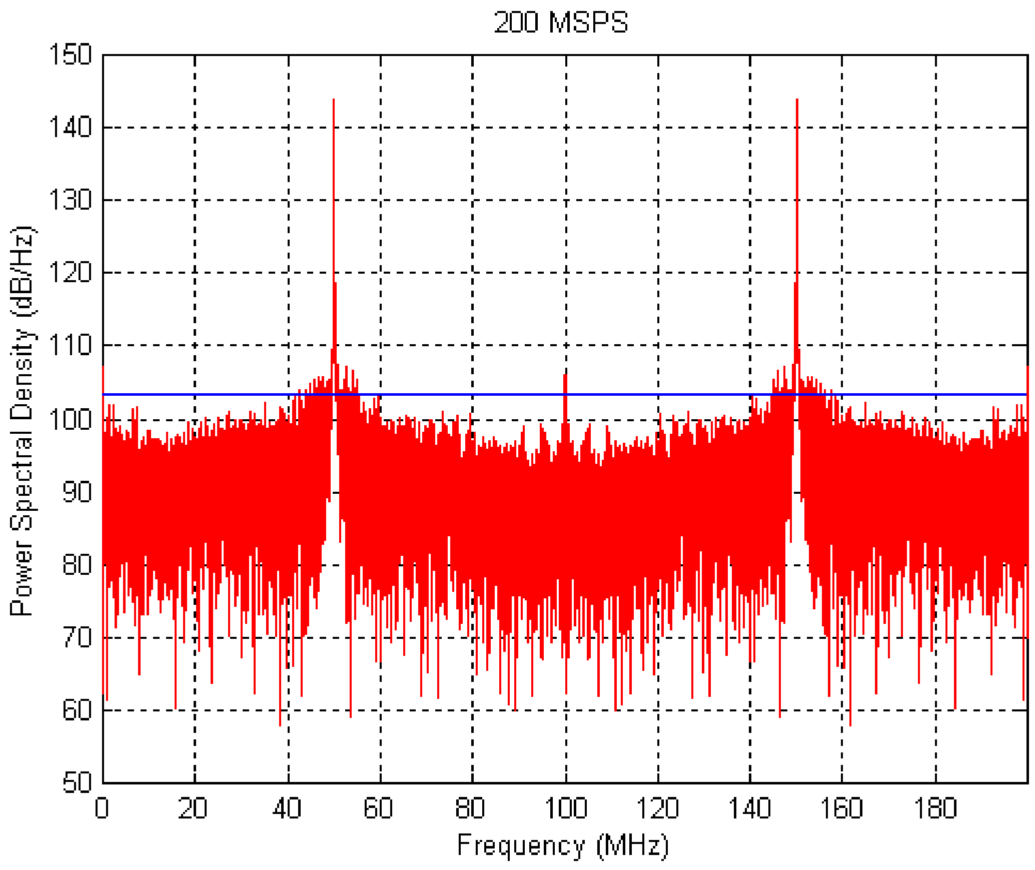

Figure 14: Spectrum for input frequency of 150 MHz

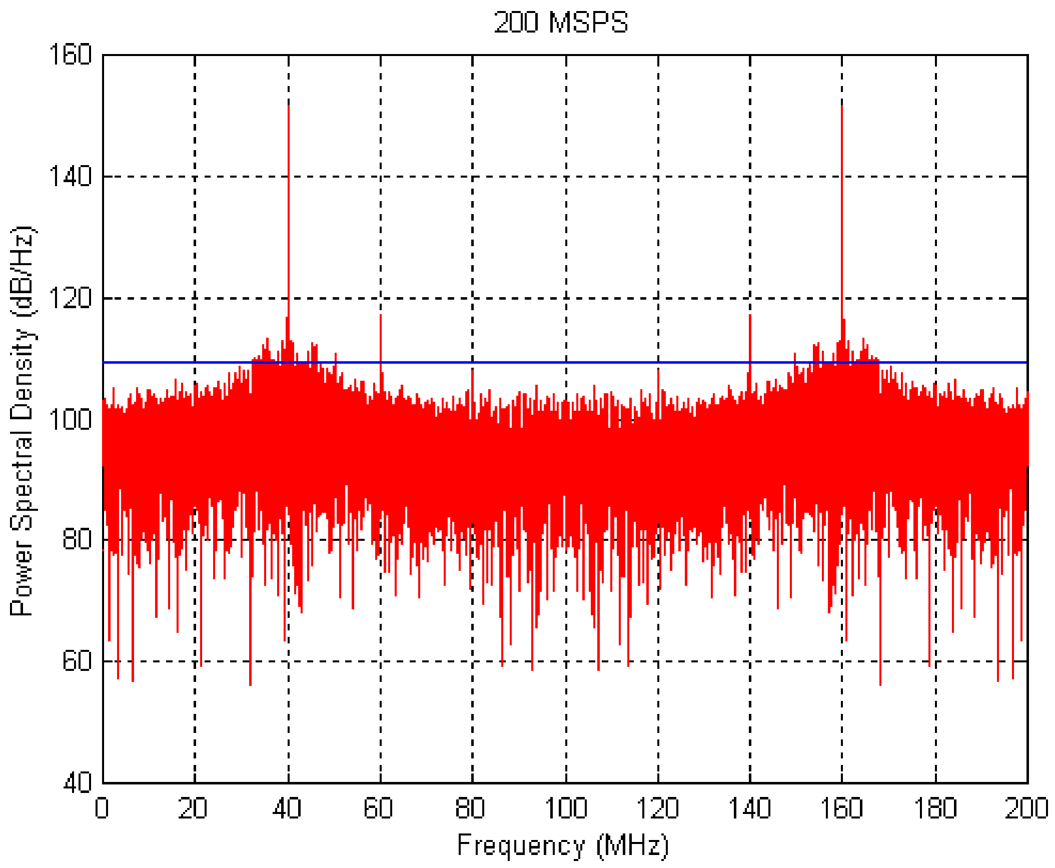

Figure 15: Spectrum for input frequency of 160 MHz

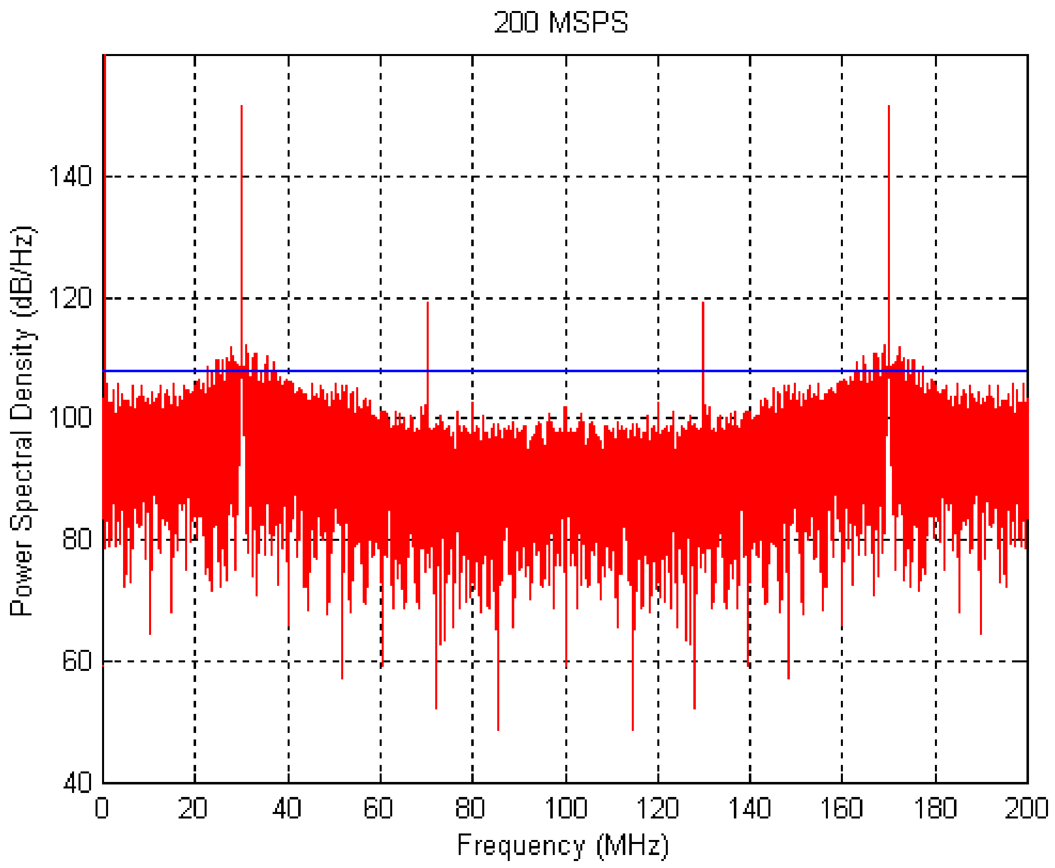

Figure 16: Spectrum for input frequency of 170 MHz

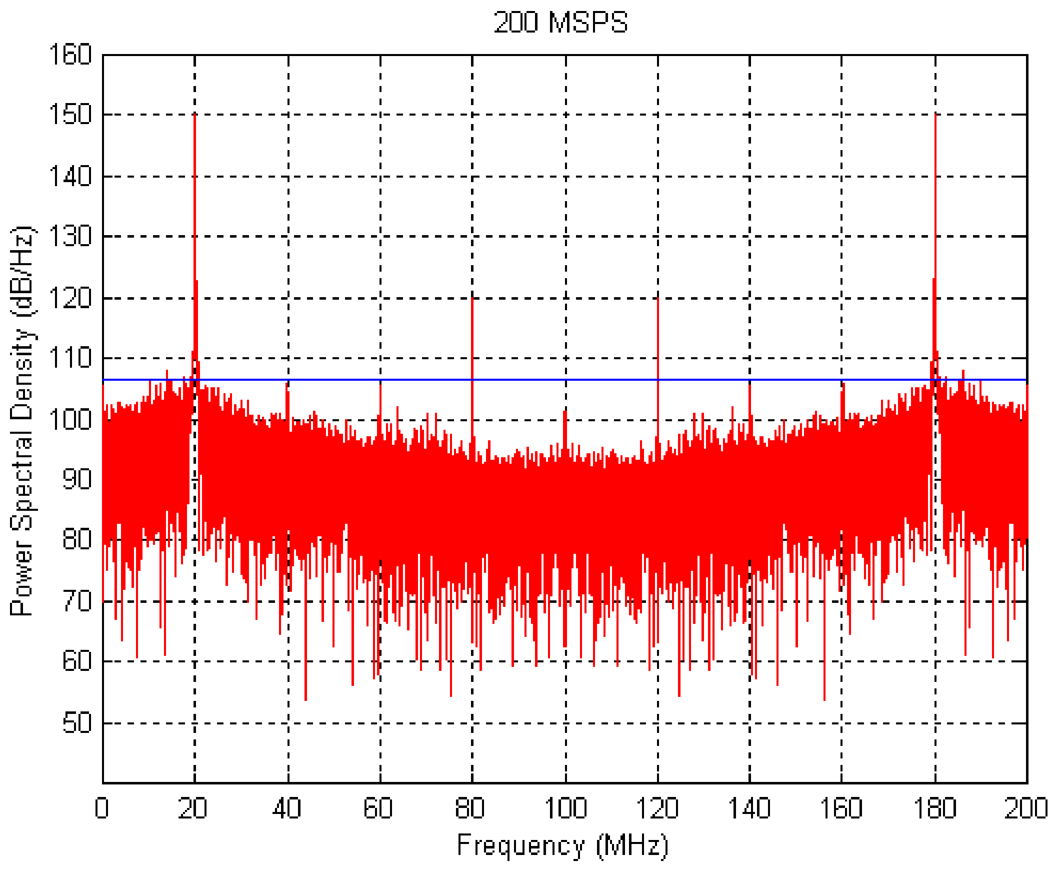

Figure 17: Spectrum for input frequency of 180 MHz

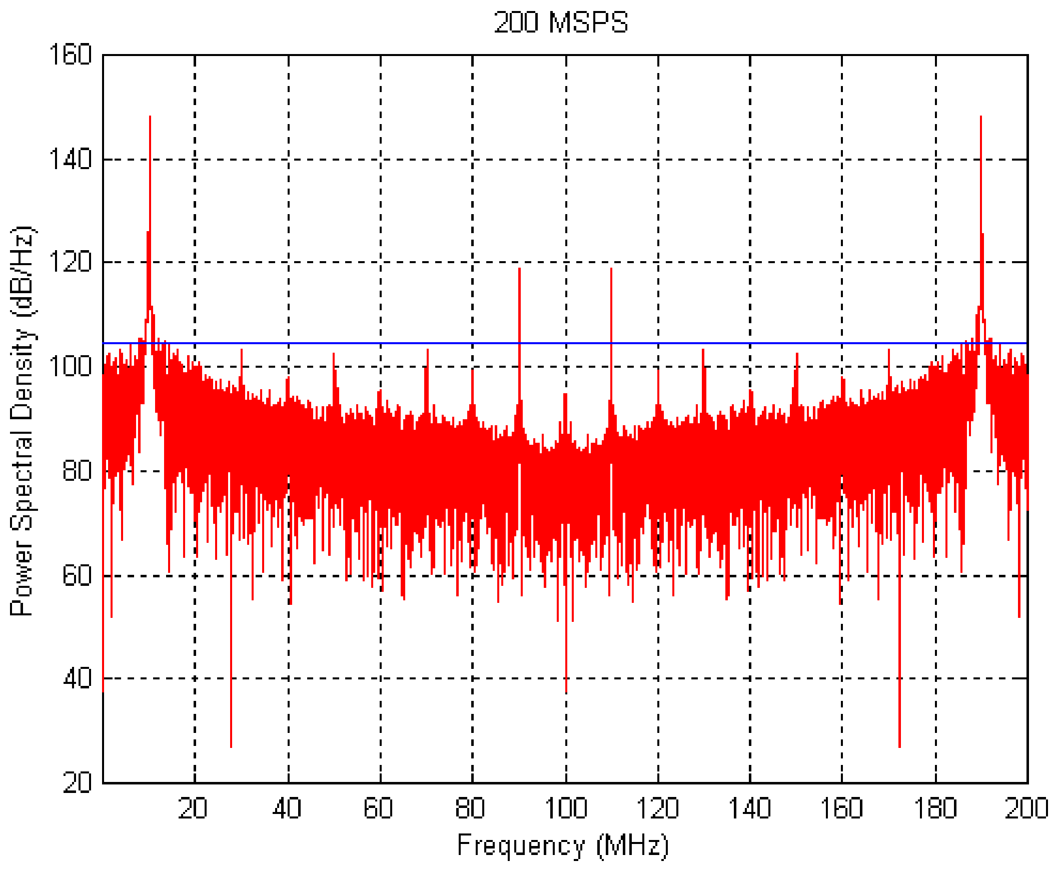

Figure 18: Spectrum for input frequency of 190 MHz

Appendix C

Spectrum Analyzer plots of DPG output sinusoids

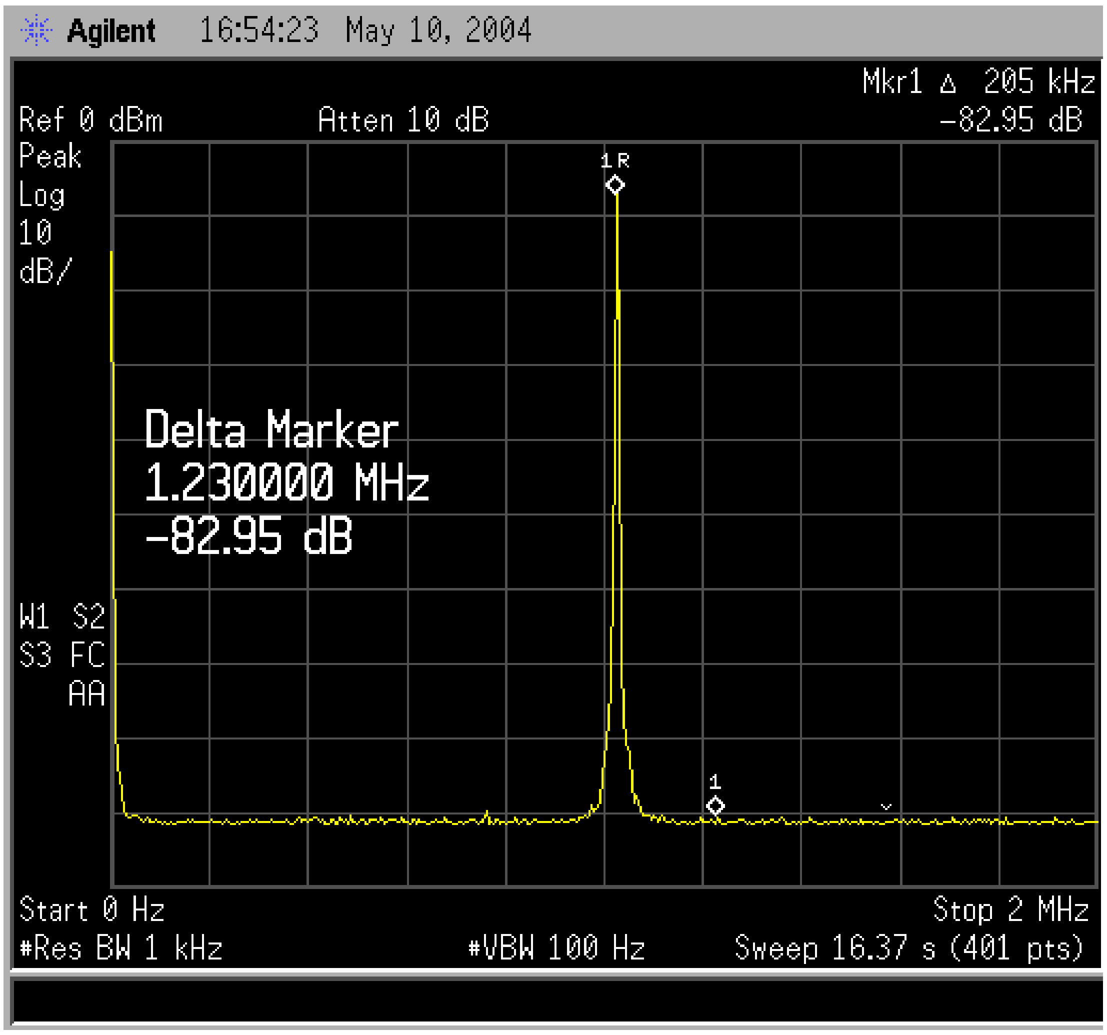

Figure 19: Spectrum for DPG output sinusoid of 1 MHz

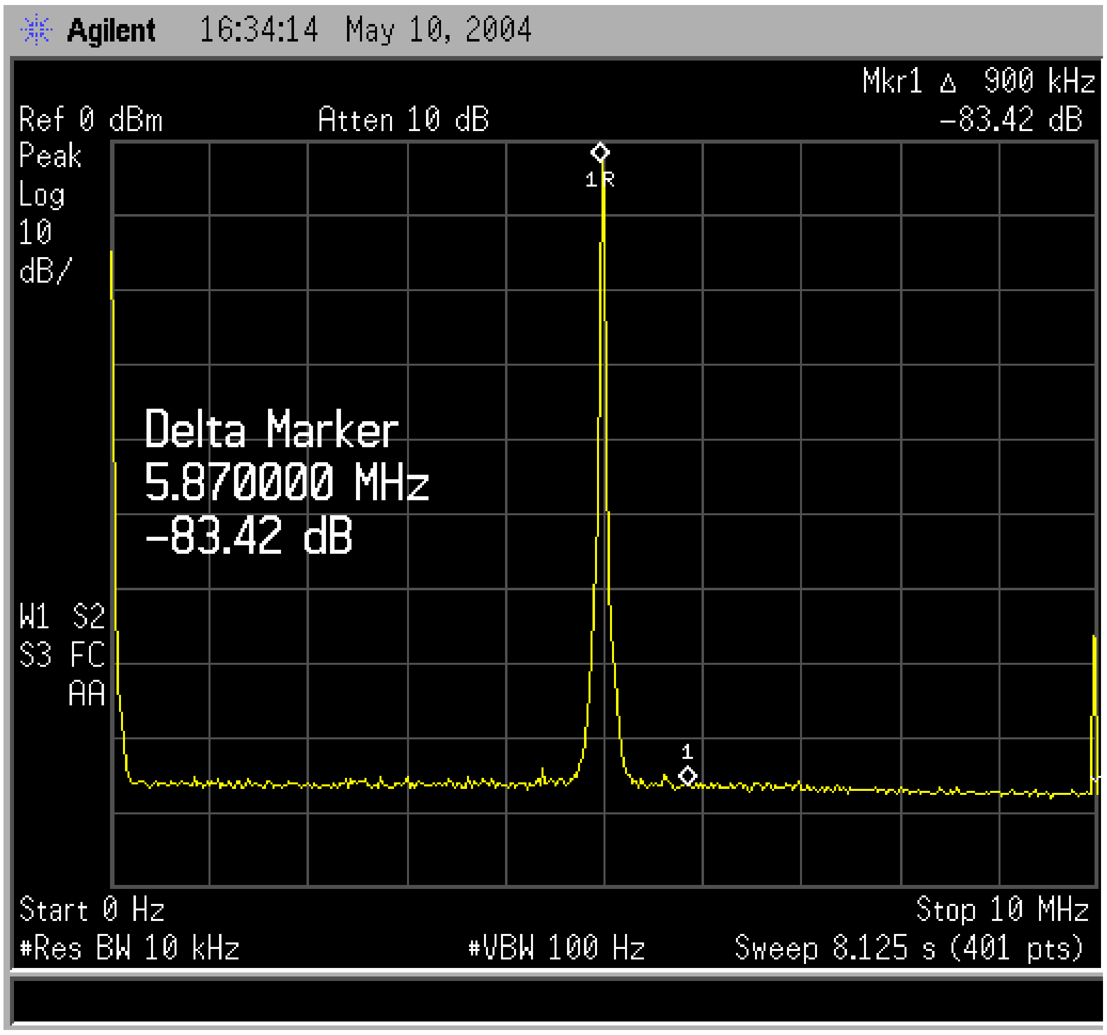

Figure 20: Spectrum for DPG output sinusoid of 5 MHz

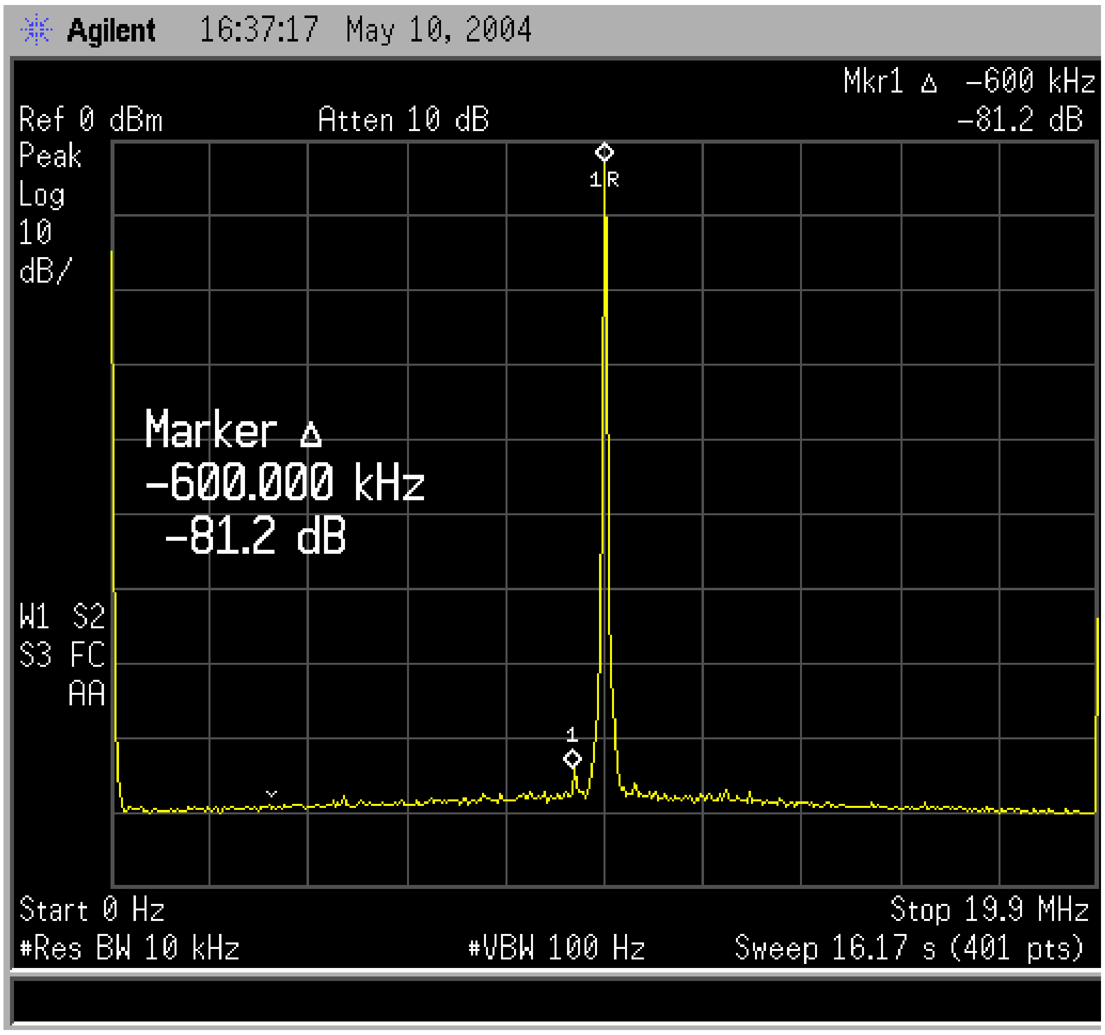

Figure 21: Spectrum for DPG output sinusoid of 10 MHz

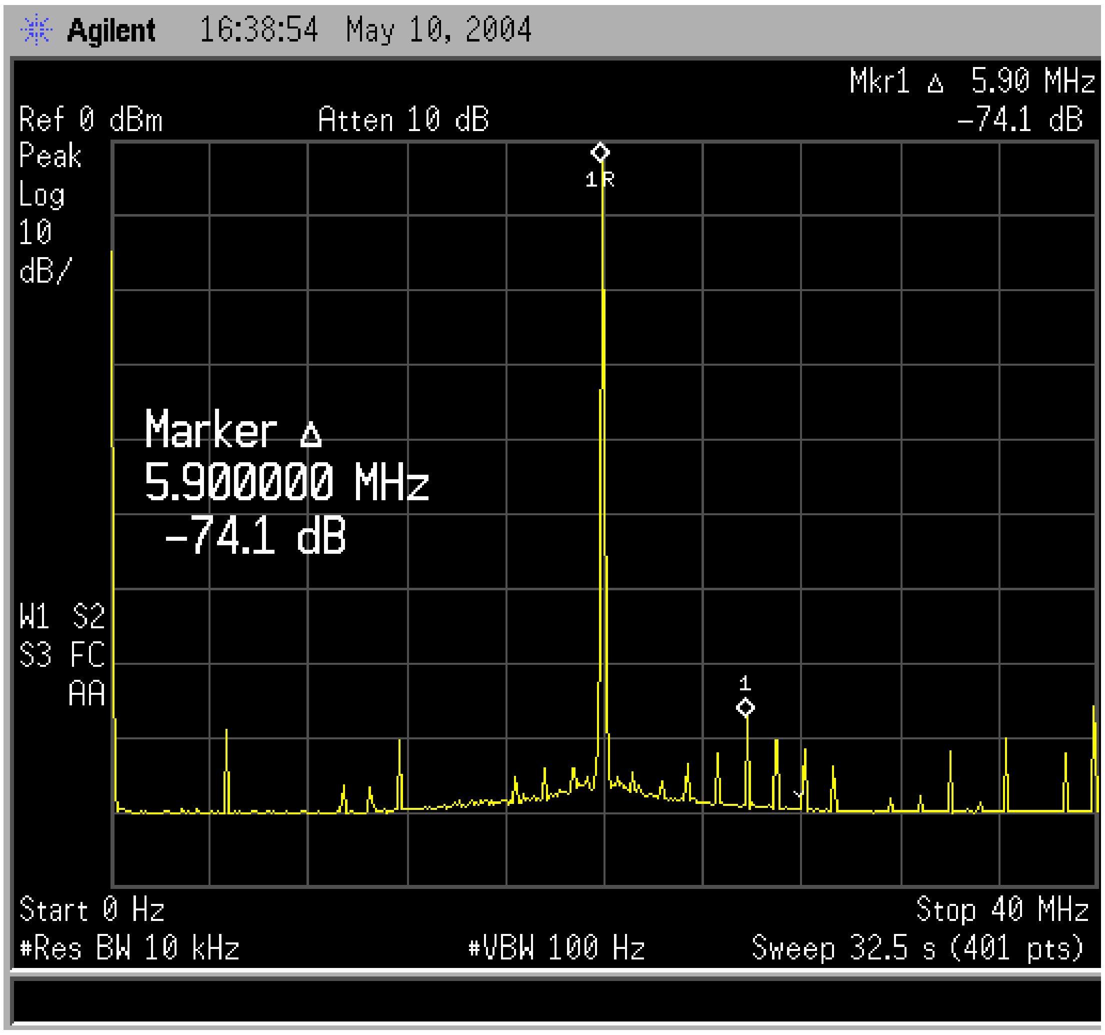

Figure 22: Spectrum for DPG output sinusoid of 20 MHz

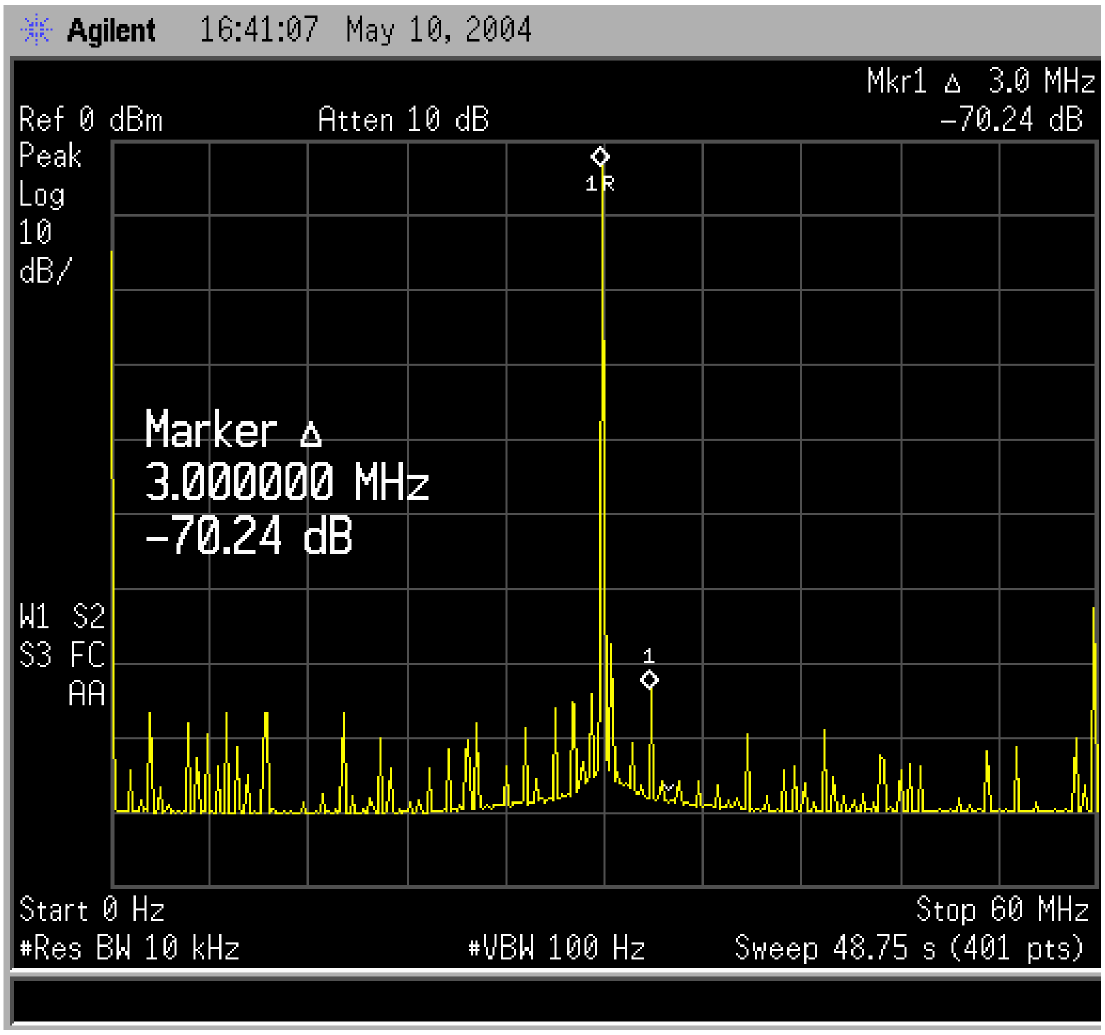

Figure 23: Spectrum for DPG output sinusoid of 30 MHz

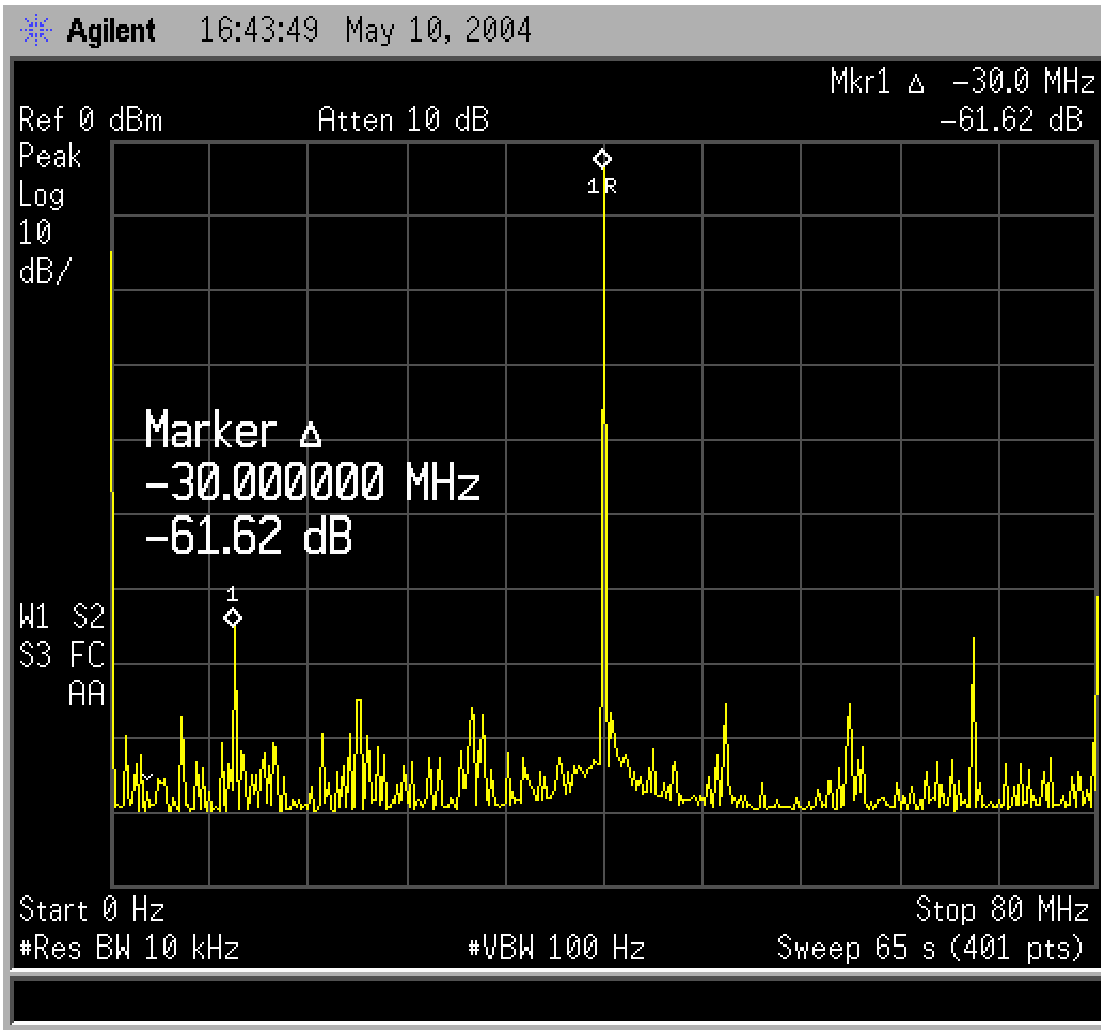

Figure 24: Spectrum for DPG output sinusoid of 40 MHz

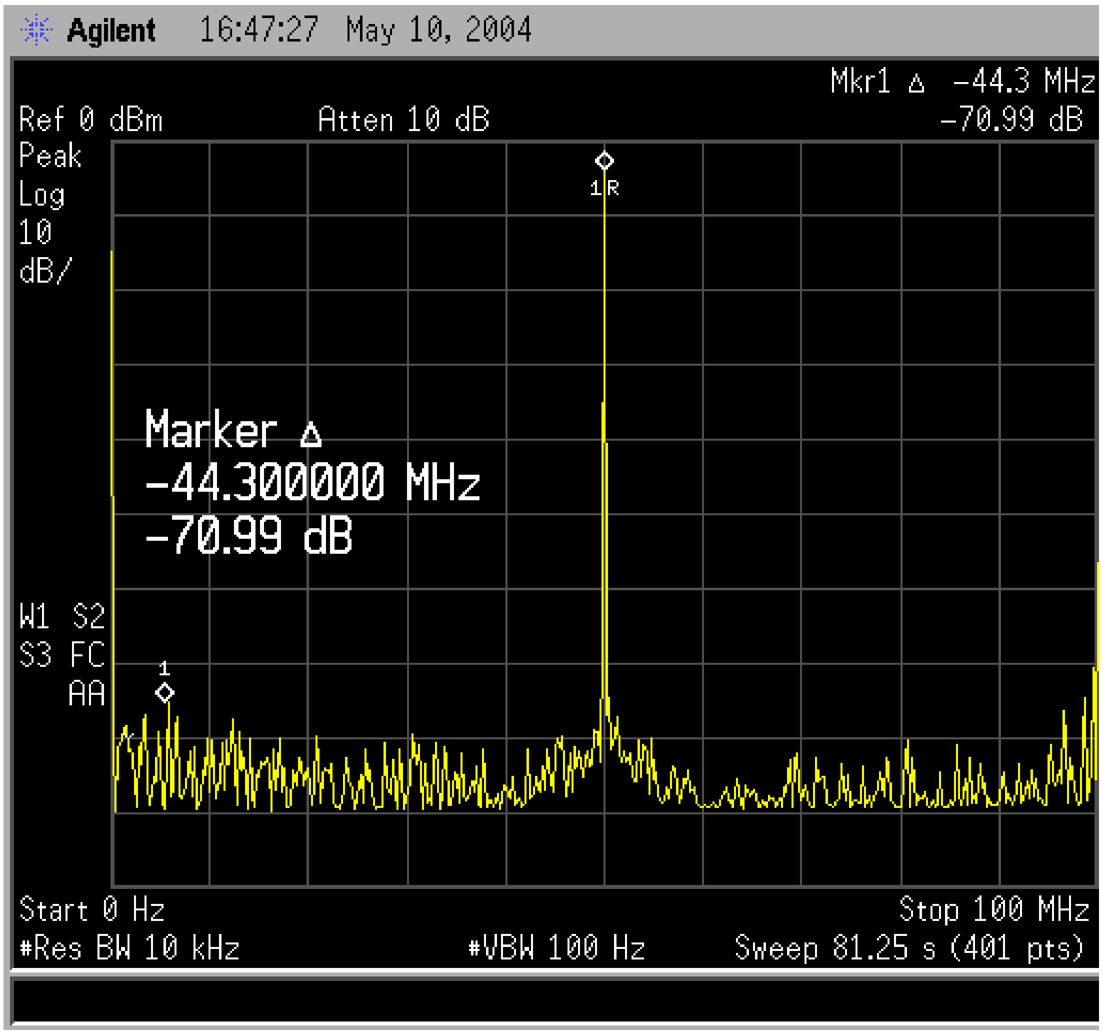

Figure 25: Spectrum for DPG output sinusoid of 50 MHz

Appendix D

Parsec Datasheets