Design of a 3GSPS ADC board for the iBOB

by

Michael Carl Gorven

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

Bachelor of Science in Electrical and Computer Engineering

at the

University of Cape Town

October 2007

Supervisor: Professor M.R. Inggs

Abstract

This thesis describes the design, implementation, production and testing of a 3 GSPS ADC board for the iBOB reconfigurable hardware platform. The ADC board was developed for use in the KAT project.

The design phase consisted of generating a high level, block diagram of the system and then creating the detailed circuit diagram, which was implemented in the PCB layout. Firmware for the iBOB was also developed to support the ADC board.

Following the successful completion of the design phase the board was sent to be manufactured. This took longer than expected, which delayed the remainder of the project. The board was populated and some basic tests were conducted.

Due to problems with the manufacture of the board and the time constraints, thorough testing of the board did not take place. However, a testing plan for the system was developed and a list of possible improvements has been included.

Declaration

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

Michael Gorven

Cape Town
22 October 2007

Contents

List of Figures

List of Tables

Glossary

10/100 Ethernet A local area network technology, 31

10GigE A local area network technology capable of transferring 10 Gbit/s, 2, 31, 32, 52, 68

AC *alternating current*, 43

ADC *analogue to digital converter*, i, 2–7, 9–13, 21, 22, 27, 31–36, 38–41, 43–48, 52, 53, 56, 60, 64, 67–70, 73, 74, 76, 96

ADC083000 An ADC chip from National Semiconductor. See Section 2.2.1 and [14], 2, 3, 11, 12, 14, 15, 21, 22, 26, 32–34, 41–43, 47, 55, 60, 66, 68–70, 76, 91

AGC *automatic gain control*, 76

balun A passive device which converts between balanced and unbalanced signals using electromagnetic coupling[28], 4, 5, 10, 36, 38, 40–43, 46, 55, 70

BER *bit error rate*: The rate of bit errors, 12, 69

BGA *ball grid array*: A type of IC package, 12

C A programming language, 7, 31, 32, 39, 52, 56

CASPER *Center for Astronomy Signal Processing and Electronics Research*: A research group at the University of California, Berkeley. See Section 1.2, 2, 32, 52, 76

DC *direct current*, 7, 23, 57, 60, 64, 71

DDR *double data rate*: Data is clocked on both edges of the clock, 11, 12, 32

dielectric constant A measure to which a material concentrates electrostatic lines of flux, 29, 58

DxDesigner Schematic capture software from Mentor Graphics, 5, 38, 40, 43, 55, 78

EDA *electronic design automation*: Software tools used to design and produce hardware, 3, 74, 75

EDK *Embedded Development Kit*, 32

ENOB *effective number of bits*: The effective resolution of an ADC. See Section 2.11.4, 3, 5, 11, 12, 35, 69

FFT *fast Fourier transform*, 35, 68, 69

FPGA *field-programmable gate array*: A programmable logic device, 2, 15, 22, 23, 31, 32, 73, 74, 76

FR-4 *Flame Resistant 4*: A material used to make PCB substrate, 7, 29, 30, 58, 71, 74

Gerber A file format for representing PCBs and used in PCB manufacture, 6, 51, 56, 86

GPIO *general purpose input/output*: Pins on a processor which can be controlled directly by the software, 7, 39, 52, 53, 56

GSPS *giga-samples per second*, i, 2, 9, 11, 12, 60, 76

HDL *hardware description language*: A language used to program FPGAs, 31, 32

HyperLynx PCB simulation software from Mentor Graphics, 6, 39, 45, 47, 96

iADC ADC board for the iBOB, developed by CASPER, 2, 3, 7, 23, 27, 36, 39, 52, 70, 76

IBIS *input/output buffer information specification*: A file format for characterising the behaviour of an IC's inputs and outputs, 47

iBOB FPGA board designed for radioastronomy applications by CASPER, i, 1–5, 7, 9–11, 15, 21–23, 26, 27, 31, 32, 36, 38–40, 42, 52, 60, 64, 66–69, 73, 74, 76

IC *integrated circuit*, 26, 43, 46

I/O *input/output*, 2, 52

KAT *Karoo Array Telescope*, i, v, 1, 3, 58, 60

LDO *low dropout*, 25

LLP *lead-free leadless leadframe package*: A type of IC package, 15

LMX2531 A frequency synthesiser chip from National Semiconductor. See Section 2.3.3 and [16], 2, 3, 12, 14, 15, 32, 33, 42, 53, 55, 60, 66, 67, 71, 91

LQFP *low-profile quad flat package*: A type of IC package, 12

LVDS *low voltage differential signal*, 11, 12

MAX6627 A digital temperature sensor from Maxim. See Section 2.5.1 and [11], 21, 22, 32–34, 43, 60, 66, 70

meerKAT Demonstrator radio telescope for the SKA, 1

MSB *most significant bit*, 33

PADS Layout PCB layout software from Mentor Grahpics, 6, 39, 44, 45, 47, 51, 86

PADS Router PCB routing software from Mentor Graphics, 6, 39, 44, 47

PCB *printed circuit board*, i, 3–8, 10, 11, 23, 27, 29, 37–39, 44, 56, 58, 60, 64, 73, 75, 86

phase noise Noise caused by unwanted frequency variations, 12, 13

PLL *phase locked loop*, 14, 15, 33

RAM *random access memory*, 2

RF *radio frequency*, 1, 21, 23

RMS *root mean square*, 13, 41

RO4003 A PCB substrate material manufactured by Rogers Corporation and designed for use at high frequencies. See Section 2.9.4 and [20], 7, 29, 30, 57, 58, 71, 75

ROACH A reconfigurable hardware platform which supersedes the iBOB, 9, 76

RS232 A serial protocol for communication between two devices, 2, 31

SDR *single data rate*: Data is clocked on only one edge of the clock, 31

SFDR *spurious free dynamic range*: The range of measurable input powers which are free from spurious signals. See Section 2.11.5 , 5, 11, 36, 69

Simulink A graphical block diagram tool for modeling and simulating systems, 32

SINAD *signal to noise and distortion*: A measure of signal quality. See Section 2.11.3, 35, 69

SKA *Square Kilometre Array*, 1

SMA A 50 Ω RF connector, 3, 4, 10, 22, 23, 27, 39, 42, 46, 60

SN74AUC34 A hex buffer gate from Texas Instruments. See Section 2.8.1 and [22], 26, 42, 60, 66

SN74LVC2G07 A buffer/driver from Texas Instruments. See Section 2.8.2 and [23], 26, 42, 60, 66

SNR *signal to noise ratio*: Ratio of signal power to noise power. See Section 2.11.1, 3, 5, 11, 12, 14, 35, 69

SPI *serial peripheral interface*: A serial protocol used for inter-chip communication, 7, 12, 22, 39, 40, 52, 53

SRAM *static random access memory*, 31

TC1-1-13M A 1:1 balun from Mini-Circuits. See Section 2.4.2 and [12], 17, 21, 60

THD *total harmonic distortion*: A measure of the distortion in a signal. See Section 2.11.2, 5, 11, 35, 69

TPS74401 A low dropout regulator from Texas Instruments. See Section 2.7.1 and [24], 25, 60, 66

transformer A device which transfers energy using inductively coupled windings, 17

VCO *voltage controlled oscillator*: An oscillator whose frequency is set by an external control voltage, 14

VHDL *VHSIC hardware description language*: An HDL, 32, 52

via A hole drilled through the board, usually used to connect signals on different layers, 46

Virtex-II Pro 2VP50 An FPGA manufactured by Xilinx. See [31], 31

Z-DOK A connector for differential lines manufactured by Tyco Electronics. See Section 2.6.1 and [27], 2–4, 10, 21–23, 25, 27, 31, 40, 42–44, 46–48, 60

Acknowledgements

This thesis would not have been possible without the involvement of a number of people. I would like to acknowledge their contributions to this project.

Firstly I would like to thank Dr Alan Langman at KAT for giving me the opportunity to do this project in association with KAT. Thank you also for supervising my project and making sure that I was on the right track.

Very special thanks go to Alec Rust at KAT, who helped me with the technical side of the project (i.e. the majority). Your assistance was invaluable, and this project wouldn't have gotten this far without you. Thank you for guiding me through the minefield of hardware design, and for putting up with my stupid questions. Thank you for giving up your time to assist me.

I would like to thank David George at KAT for his assistance in developing the firmware. Since I did not have time to understand the firmware development process properly, I often had to draw on Dave's knowledge and ask him what my compile errors meant.

Thank you also to all the people at KAT who helped me out from time to time.

To Jason Salkinder, thank you for showing me the ropes and for helping me to figure problems out. Your input helped me to get up and running much quicker than if I had to work it out myself, and your knowledge was very useful. I learnt a lot from you, and you made my task much easier.

Thank you to my supervisor Prof. Inggs for his input and guidance during my project. Your assistance in planning this project enabled it to succeed.

To Kylie: Thank you for your continuous support and encouragement throughout this project, and for keeping me sane during this semester! I couldn't have done it without you.

I would also like to acknowledge some of the Open Source software which I used in preparing this thesis: Ubuntu, KDE, gerbv, LaTeX, Vim, Kivio, Inkscape, gnuplot, GNU Octave, gerber2pdf, Subversion, GIMP and VirtualBox.

Chapter 1

Introduction

This chapter describes the background to the project and introduces the iBOB hardware. The project requirements are then presented, followed by a review of these requirements. It then lists the stages of the project and provides a summary of what each stage entails.

1.1 Background

Radio telescopes receive RF radiation emitted by objects in the universe. This data is analysed and used by astronomers to study the universe. The telescope consists of a radio antenna which receives the actual signal, and a system to store and process the signals. As with most technology nowadays, this is done digitally. A crucial part of this system is converting the analogue signal to its digital representation. This project is concerned with the analogue to digital conversion of the signal.

The *Square Kilometre Array* (SKA) is an international project to build a radio telescope consisting of thousands of dishes with a total area of one square kilometre [21]. South Africa is one of the two[1] candidate countries to host the SKA. The SKA is the first of this scale, and requires new and improved technologies in order to succeed.

The South African team working on the SKA is the *Karoo Array Tele-*

[1]The other country is Australia.

scope (KAT) project. KAT is designing and developing a prototype for the SKA called meerKAT, which will be installed in the Northern Cape. This prototype will be used to test the new technology and equipment being developed. The KAT project includes the development of the telescope antennas, hardware, the datacentre and computing technology which will process this data, and the network to connect the telescope and the datacentre.

1.2 CASPER and iBOB

The *Center for Astronomy Signal Processing and Electronics Research* (CASPER) is a research group at the University of California, Berkeley. Their goal is to "streamline and reduce the current radio astronomy instrumentation design flow through the development of an open-source, platform-independent design approach" [1]. They developed the iBOB, which is a hardware platform designed for use in radio astronomy. The iBOB contains an FPGA, RAM and I/O (including RS232 and 10GigE).

The iBOB uses separate ADC boards to perform the actual signal digitisation. The ADC boards connect to the iBOB via the Z-DOK connectors. CASPER also developed the iADC board to perform this task. The version 1 iADC board uses an e2v[2] AT84AD001B chip, which is an 8-bit ADC capable of digitising either two channels at 1 GSPS each, or a single channel at 2 GSPS [6, p.1].

1.3 Project Requirements

The thesis project is to design, implement and test a 3 GSPS ADC board for the iBOB. The ADC chip used by the board will be the National Semiconductor ADC083000 chip. The board must be able to generate the sampling clock from a low speed reference clock using a National Semiconductor LMX2531 chip, but should also accept an external sampling clock.

[2] e2v took over Atmel's data converter products in August 2006.

The iBOB must be configured to use the ADC board. The performance of the ADC must be determined by analysing the data it produces.

1.4 Requirements Review

In order to be compatible with the iBOB, the board will have to be the same size as the iADC board and contain four mounting holes in the correct locations. It must have a Z-DOK adapter mounted on the correct edge.

The ADC083000 requires a 1.5 GHz sampling clock [14, p.1], which can be generated by the LMX2531[3] [16, p.1]. The LMX2531 requires a reference frequency of 5–80 MHz.

The board will require connectors for the input signal, reference clock and external sampling clock. The most common connector in use is the 50 Ω SMA connector. As this is used by the lab equipment at KAT, this is what should be used on the board. Power is supplied to the board through the Z-DOK connector, but the board will require voltage regulators in order to provide the correct voltages.

The project will require the selection of components and design of the circuit. The schematic must then be captured using *electronic design automation* (EDA) software, and the PCB must then be laid out. Simulation of the signals on the board must then be carried out to verify signal integrity and noise performance. Once the PCB has been manufactured, it must be built up.

Firmware for the iBOB must then be written to retrieve data from the ADC and transmit it to a computer. The frequency synthesiser and possibly the ADC chip must also be programmed. Software may have to be written in order to receive data from the iBOB and store it.

The ADC board and iBOB firmware must be tested to ensure that the system operates correctly. The system is to be tested with various input signals and the data collected. This data needs to be analysed and interpreted in order to determine the performance of the ADC board. Measures such as

[3]The lowest specified frequency of the LMX2531LQ1570E is actually 1530 MHz, but the difference of 30 MHz should be negligible.

ENOB and SNR must be calculated.

1.5 Project Overview

The project consists of three main stages, which correspond to the chapters that follow in this report.

1.5.1 Technical Research

Chapter 2 presents the results of the research done in order to carry out this project. Each requirement of the system is presented along with the components that will fulfil these requirements. Information on the various design issues that will need to be addressed is provided. Aspects regarding PCBs in general as well as issues specific to this application are then investigated. The structure of the iBOB firmware is presented, as well as the information which will be needed when implementing the firmware. Finally, the tests which will be used to evaluate the ADC board are explained.

The chapter begins with the details of the ADC chip being used, as well as another ADC chip which was initially considered. A brief introduction to phase noise and jitter is presented, followed by the requirements for the sampling clock, after which the chosen frequency synthesiser is discussed. The characteristics of the input signal are then discussed, and various methods of converting it to a differential signal are outlined. The balun chosen to do this conversion is then introduced.

The chip which will monitor the temperature of the ADC is described, followed by details on the Z-DOK and SMA connectors which will be used to interface with other systems. The setup of the power supply and the voltage regulator are then discussed. The need for level converters is explained and the two chips which will be used as level converters are then presented.

Issues regarding the PCB are then discussed. These include the form factor of the board as required by the iBOB, and the PCB manufacturer's capabilities which the board must adhere to. The required impedance matching and length matching is detailed, and then the substrate material and board

stackup are discussed.

Details of the iBOB hardware are presented, as well as the structure and development environment of the firmware. The required functions of the firmware are then presented, along with detailed information regarding the serial, control and data interfaces that the firmware will need to manage.

The analysis of ADC performance is then discussed. Definitions of the various tests to be done on the output data (ENOB, SNR, SFDR and THD) will be described, as well as their interpretation and meanings.

Although the first few weeks of the project consisted mostly of this research, the research phase extended throughout the project. Numerous issues were uncovered as the project progressed, and these issues necessitated further research. This chapter presents all the information which was directly used while implementing the project and which is needed in order to understand the project.

1.5.2 System Design

Chapter 3 details the design process that was followed during the project. The high level design is first executed, resulting in a block diagram of the system. The exact connections between the components are then determined in the circuit design. This is converted to a PCB during the board layout, and some simulations are done to verify the design. The firmware to run on the iBOB is then designed and implemented.

The chapter begins with the high level design of the system. Each requirement of the system is investigated and the functions required in order to meet these requirements are determined. Support functions (such as the power supply and temperature monitoring) are also considered, and the external interfaces are defined. The logical connections between the components and the external interfaces are determined, which results in the overall architecture of the system as shown in Figure 1.1. This architecture is conceptually simple, but is sound and capable of meeting the requirements.

This architecture is then implemented in the detailed circuit design. This determines the physical connections between all the components, as well as

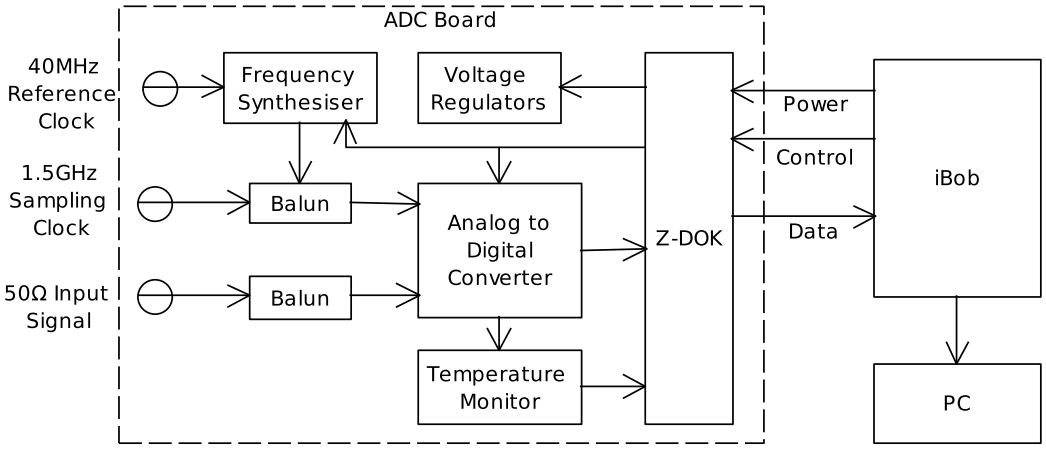


Figure 1.1: System Block Diagram

the passive components which are needed. The balun configuration is chosen, and the power supplies and level converters are designed. The frequency synthesiser, ADC and temperature monitor are then added and connected. Lastly, the schematic is captured in DxDesigner.

The circuit is then realised as a PCB which is laid out using PADS Layout. First, the design rules and layer stackup are defined, taking the required impedance matching into account. The components are then placed on the board and the power planes are defined. The board is then routed in PADS Router. The decoupling capacitors are manually routed first, then the ADC outputs are autorouted, after which the other traces are routed manually. The layout and routing process took much longer than expected. However, in the end it was completed successfully and resulted in a good design.

HyperLynx is then used to ensure that the impedances are correctly matched. Signals on the ADC outputs are also simulated in order to check the signal integrity and the influence of cross talk. The simulations were very useful while designing the board and are an essential part of designing a good system. Simulations of the final design show that the impedances are well matched and that the signal integrity of the ADC outputs is very good.

The board is then prepared for production. The components are labelled on the silkscreen. The design rule checks are run in order to verify that

the design is correct. Finally, the Gerber files to be used in production are generated and checked. All the errors which were found were fixed, and the Gerber files were successfully generated and checked.

The firmware for the iBOB is adapted from existing firmware. Since the board uses the same pin connections as the iADC, the data collection routine is compatible. The control pins are configured as GPIO and made accessible to the processors. The SPI protocol is implemented in C and used to program the ADC and frequency synthesiser. Finally, the initialisation routine is written. Integrating the GPIO core was much more difficult than expected, and prevented the firmware from being completed in time.

Overall, the design of the system was successful but did require much more time than expected. More experience with the PCB design process and the tools involved would have decreased this time significantly. The resulting design appears to be good and the system should operate as intended with good performance.

1.5.3 Integration and Testing

Chapter 4 describes the manufacture and population of the board. The board and firmware are first tested individually, and then the system is tested as a whole. Data is collected and analysed to determine the performance of the board. A list of changes to correct and improve the design is then provided.

The chapter begins with the manufacture of the board, and the problem that was encountered, which was the unavailability of RO4003 laminate. This resulted in mismatched impedances which also vary significantly with frequency due to the nature of FR-4. The manufacture was also delayed significantly, which meant that there was not enough time in which to complete the integration and testing phase. Some simulations are then done to determine the repercussions this will have on the board. Sourcing of the parts for the board is briefly described, followed by the population of the board. This was done by hand and required a non-standard technique for mounting the leadless components.

The board is then tested, beginning with a visual inspection. The form

factor of the board is checked, and some DC tests are done to check connectivity. The board is then powered up and the power supplies are tested and inspected for ripple. All the DC tests were passed. Measurements of the power supplies all agreed with the expected values, and the ripple on the power supplies is acceptable.

The rest of the system is then tested, beginning with the firmware. Unfortunately the firmware did not work as intended, and there was not sufficient time to fix it. The remainder of the chapter describes the tests which would have been done after the firmware was working. The sampling clock would be tested next, followed by the data collection routine. Lastly, the entire system would be tested.

The system's performance would then be evaluated. Three sets of data would be collected for analysis, using three different sinusoids as the input signal. This data would be processed and various measures calculated. These measures would be compared to the expected values, and plots of the data would be generated. Unfortunately no results were gathered since the system was not completed. The performance of the system therefore could not be compared and evaluated.

Some errors were found with the board, as well as a few improvements which could be incorporated in future revisions. A list of changes which should be implemented is provided. These include errors which were found during testing, as well as some possible improvements.

Even though it was not fully tested, the board certainly has the potential to work correctly and meet all the requirements. If the board manufacture hadn't been delayed, I fully expect this phase to have been completed successfully.

1.5.4 Conclusions and Future Work

Chapter 5, the final chapter, presents the outcomes and results of the project, as well as what was not achieved. Conclusions are then drawn from the experience, and possible future work is briefly outlined.

The research and design phases of the project were completed successfully,

although the board layout did take longer than expected. These phases produced good high level, circuit and PCB designs which appear to meet the requirements. The firmware was implemented but not tested. The board was manufactured and populated, after which the basic board tests were done. A plan to test the system and evaluate its performance was developed, but not executed due to a lack of time.

Various conclusions are then drawn. The difficulties of designing the ADC board lay predominately with the high speed aspects rather than the system design. A large number of skills were required for this project, and a lot of time was required to learn how to use the various tools. Simulations were a very important part of the design, and the autorouter was an extremely useful tool.

The board which was produced appears to be a successful implementation of the requirements. The project was however a very daunting task due to the time, skills and knowledge which was required. A great deal of knowledge was gained as well as new skills, and the project was a good experience.

There are a few opportunities for future work leading on from this project. The integration and testing phase can be completed by executing the test plan given in Chapter 4. A board revision could be done to fix the errors which were found and implement some new features and improvements. Developing firmware for the iBOB which is capable of handling 3 GSPS would improve the capabilities of the iBOB. Interleaving two ADC boards to achieve 6 GSPS would provide impressive sampling performance. Lastly, the board could be adapted for use with ROACH.

Chapter 2

Technical Research

This chapter presents the results of the research done in order to carry out this project. Each requirement of the system is presented along with the components that will fulfil these requirements. Information on the various design issues that will need to be addressed is provided. Aspects regarding PCBs in general as well as issues specific to this application are then investigated. The structure of the iBOB firmware is presented, as well as the information which will be needed when implementing the firmware. Finally, the tests which will be used to evaluate the ADC board are explained.

2.1 Introduction

The chapter begins with the details of the ADC chip being used, as well as another ADC chip which was initially considered. A brief introduction to phase noise and jitter is presented, followed by the requirements for the sampling clock, after which the chosen frequency synthesiser is discussed. The characteristics of the input signal are then discussed, and various methods of converting it to a differential signal are outlined. The balun chosen to do this conversion is then introduced.

The chip which will monitor the temperature of the ADC is described, followed by details on the Z-DOK and SMA connectors which will be used to interface with other systems. The setup of the power supply and the voltage

regulator are then discussed. The need for level converters is explained and the two chips which will be used as level converters are then presented.

Issues regarding the PCB are then discussed. These include the form factor of the board as required by the iBOB, and the PCB manufacturer's capabilities which the board must adhere to. The required impedance matching and length matching is detailed, and then the substrate material and board stackup are discussed.

Details of the iBOB hardware are presented, as well as the structure and development environment of the firmware. The required functions of the firmware are then presented, along with detailed information regarding the serial, control and data interfaces that the firmware will need to manage.

The analysis of ADC performance is then discussed. Definitions of the various tests to be done on the output data (ENOB, SNR, SFDR and THD) will be described, as well as their interpretation and meanings.

2.2 ADC

The ADC is the heart of this board, since it provides the fundamental function of the board. The ADC determines the main characteristics of the board (such as the sampling rate and resolution) and has a large effect on the performance of the board. There were two ADCs which were considered for this project. These are the National Semiconductor ADC083000, and the e2v EV8AQ160. Although the ADC083000 was specified in the project requirements, the EV8AQ160 is briefly mentioned for completeness.

2.2.1 National Semiconductor ADC083000

The ADC083000 is a single channel, low power, high performance 8-bit ADC manufactured by National Instruments[1]. It is rated at 3 GSPS (but can achieve up to 3.4 GSPS). It requires a 1.5 GHz[2] differential sampling clock

[1] `http://www.national.com`

[2] The ADC samples on both edges of the clock, and hence the sampling rate is twice the frequency of the clock.

and a differential input signal. The output data is provided on four *low voltage differential signal* (LVDS) buses using *double data rate* (DDR) (each bus therefore runs at 375 MHz for a sampling rate of 3 GSPS).

The chip is configured using either control pins or a serial interface (which in known as Extended Control Mode). The Extended Control Mode is more versatile, as it allows more parameters to be configured (such as the input offset, full-scale range and clock phase adjustment). The ADC083000 features a self calibration routine and an internal sample-and-hold amplifier, and can output a known test pattern for use in debugging.

The chip requires a 1.9 V power supply, and consumes approximately 1.9 W. For a 748 MHz input, the ADC083000 can supposedly achieve a *bit error rate* (BER) of 10^{-18}, an *effective number of bits* (ENOB) of 7.0 bits and a *signal to noise ratio* (SNR) of 44.5 dB. Its full power bandwidth is specified as 3 GHz. The chip comes as a 128 lead *low-profile quad flat package* (LQFP). [14, p.1]

2.2.2 e2v EV8AQ160

The EV8AQ160 is a quad 8-bit ADC manufactured by e2v[3]. It is capable of digitising either four channels at 1.25 GSPS, two channels at 2.5 GSPS, or one channel at 5 GSPS. It requires a 2.5 GHz differential sampling clock and differential input signals. The output data is provided on eight LVDS buses using DDR. It is configured using a *serial peripheral interface* (SPI), and comes as a 380 pin *ball grid array* (BGA) package. [7, p.1,6]

2.3 Sampling Clock

The sampling clock is used by the ADC to determine when to sample the input signal. Part of the performance of the board rests on the quality of the sampling clock. What follows is a discussion of phase noise and jitter, and then the requirements for the sampling clock are presented. An overview of the LMX2531 frequency synthesiser is then provided.

[3] `http://www.e2v.com`

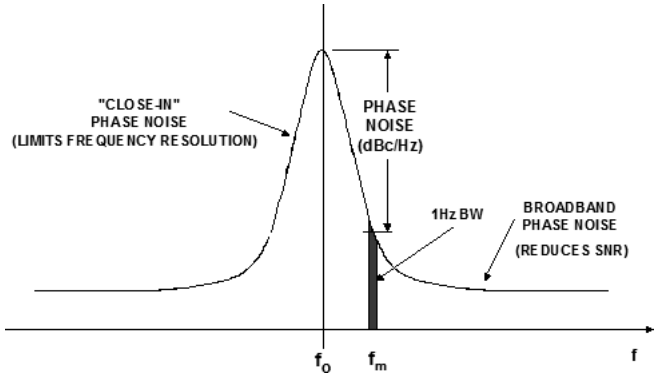


Figure 2.1: Frequency spectrum of a non-ideal oscillator showing measurement of phase noise [10]

2.3.1 Phase Noise and Jitter

An ideal clock would oscillate at one single frequency. In the frequency spectrum there would be a single component at the oscillating frequency. Real oscillators however, have a frequency which varies with time. In the frequency spectrum this appears as additional components surrounding the oscillator frequency, as shown in Figure 2.1. These deviations from the oscillator frequency are known as phase noise. [29]

In the time domain phase noise appears as cycles which are either faster or slower than they should be. In the context of a sampling clock, this means that samples get taken either earlier or later than they should. This is known as aperture uncertainty, and it creates an error in the sampled voltage (as shown in Figure 2.2). This error adds noise to the sampled signal, and reduces the performance of the ADC. [2, p.1–2]

Phase noise is measured by calculating the noise power at a certain offset from the centre frequency, and is expressed in decibels relative to the carrier (dBc). This is illustrated in Figure 2.1. Jitter is expressed as the RMS average of the difference between the ideal cycle time and the actual cycle time. It is possible to convert between phase noise and jitter.

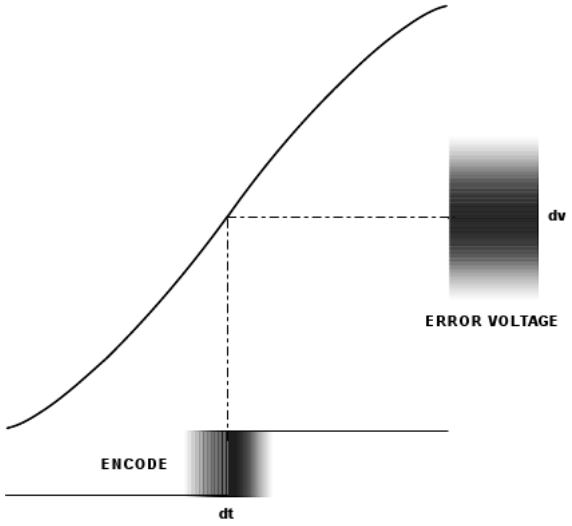


Figure 2.2: Aperture uncertainty causes an error in the sampled voltage [2, p.1]

2.3.2 Clock Requirements

In order to minimise the noise in the sampled signal, the phase noise should be as low as possible. The clock should also be stable and accurate in order to aid analysis of the sampled data.

The ADC083000 datasheet provides a formula (shown in Equation 2.1) to calculate the maximum jitter allowable in order to prevent a reduction in SNR due to jitter ($V_{\mathrm{IN(P–P)}}$ is the peak-to-peak input voltage, V_{INFSR} is the full-scale range, N is the resolution in bits, and f_{IN} is the maximum input frequency). [14, p.31]

$$t_{\mathrm{J(MAX)}} = \frac{V_{\mathrm{IN(P–P)}}}{V_{\mathrm{INFSR}}} \times \frac{1}{2^{N+1} \times \pi \times f_{\mathrm{IN}}} \tag{2.1}$$

Evaluating this equation for $V_{\mathrm{I(P–P)}} = 750\,\mathrm{mV}$, $V_{\mathrm{INFSR}} = 800\,\mathrm{mV}$ and $f_{\mathrm{IN}} = 1.5\,\mathrm{GHz}$ yields a maximum jitter of $t_{\mathrm{J(MAX)}} = 389\,\mathrm{fs}$. The sampling clock should therefore have a maximum jitter of 389 fs.

2.3.3 National Semiconductor LMX2531

According to National Semiconductor [16, p.1], the LMX2531 is a "low power, high performance frequency synthesizer system which includes a fully integrated delta-sigma PLL and VCO with fully integrated tank circuit." It uses a reference oscillator to generate a specific output frequency. It is capable of producing a very stable clock with low phase noise. The LMX2531 has a fractional-N PLL, and an adjustable loop filter. It also has a FastLock feature, which decreases the lock time.

The phase noise of the LMX2531 is specified in the datasheet [16, p.8]. This was converted to a jitter value using an online tool provided by Raltron [18]. This yielded a jitter value of 261 fs, which is less than the maximum jitter of 389 fs calculated in Section 2.3.2.

The reference oscillator must be 5–80 MHz. The LMX2531 is programmed using a serial interface (in order to configure the PLL, phase detector, loop filter and FastLock). It comes as a 36 pin *lead-free leadless leadframe package* (LLP) package. The connection diagram from the datasheet is shown in Figure 2.3. [16, p.1,6]

The LMX2531 uses a 3 V power supply, but the serial interface runs at 2.65 V. The electrical characteristics of the LMX2531 and the FPGA on the iBOB were compared to determine whether the interfaces were compatible. It was discovered that the output range of the FPGA fell within the input range of the LMX2531. The interfaces are therefore directly compatible, and a level converter is not necessary. [16, p.10] [31, p.76]

2.4 Input Signal

The input signal and the sampling clock[4] are single ended 50 Ω signals. The ADC083000 however, has differential inputs for both these inputs. Differential signals represent the signal value as the difference in voltage between

[4]Both the external sampling clock and the output of the LMX2531 are 50 Ω single ended.

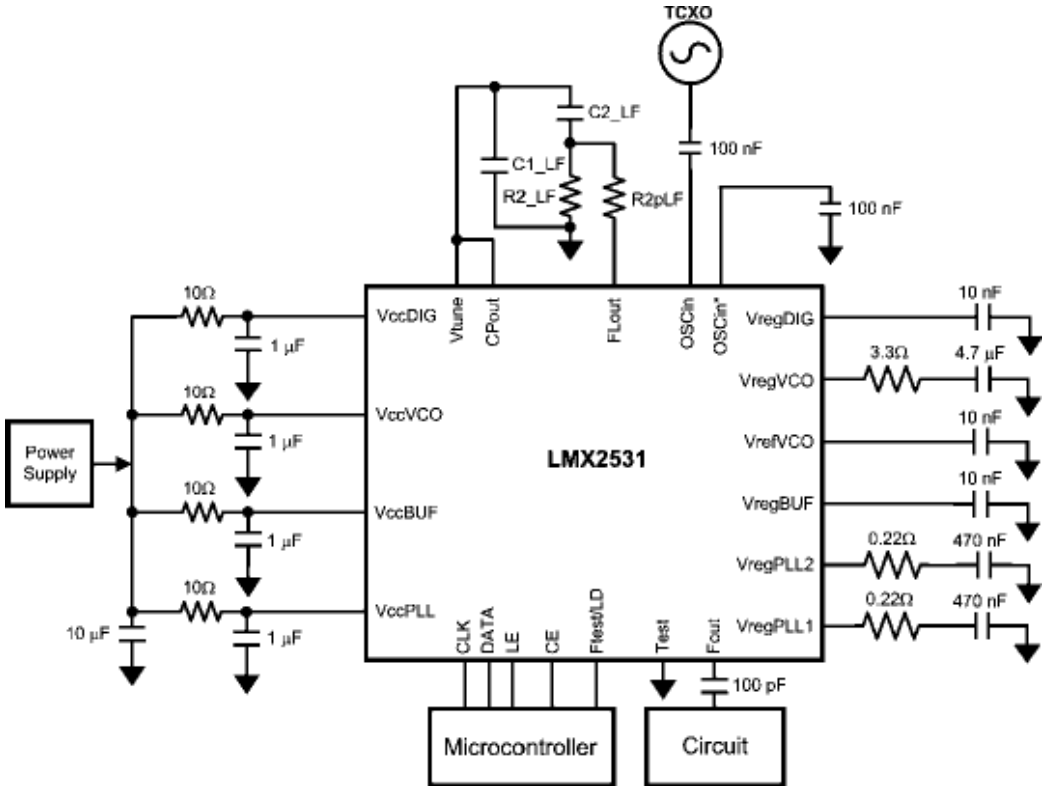


Figure 2.3: LMX2531 Connection Diagram [16, p.4]

two lines. This provides better noise immunity.[5]

It is possible to leave these signals as single ended. This is done by connecting the signal to the positive input, and connecting the negative input to ground. The problems with this method are that the impedance matching will not be good, and that it is more susceptible to noise. These two signals must therefore be converted to differential signals. Methods of doing this conversion will now be covered, and then the TC1-1-13M is presented.

2.4.1 Single to Differential Conversion

There are numerous methods of converting single ended (also known as unbalanced) signals to differential (also known as balanced) signals. All of these methods use some kind of electromagnetic coupling. Five different techniques using transformers are briefly described here, followed by two other methods.

Direct conversion using a 1:$\sqrt{2}$ transformer

The first technique uses a 1:$\sqrt{2}$ transformer, as shown in Figure 2.4. The advantages of this technique is that no power is lost in the conversion, and no other components are required. The disadvantage is that 1:$\sqrt{2}$ transformers are uncommon due to the number of windings required. [8, p.2]

Conversion using a 1:1 transformer

The second technique uses a 1:1 transformer as a normal transformer, as shown in Figure 2.5. The advantage is that 1:1 transformers are readily available. The disadvantages are the extra 100 Ω resistor required for matching and that half of the power is lost in the conversion. [8, p.3]

Conversion using a 1:1 transformer with double secondary

The third technique uses a 1:1 transformer with a double secondary, as shown in Figure 2.6. The only advantage of this configuration over the previous one

[5]The receiver calculates the signal value as $x = x^+ - x^-$. If noise is added to each line the calculation becomes $x = (x^+ + n) - (x^- + n) = x^+ - x^-$, showing that the noise is cancelled out when calculating the difference.

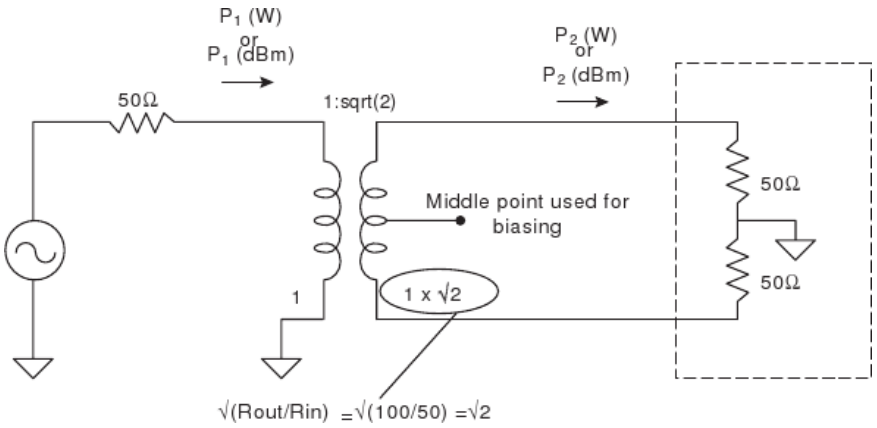


Figure 2.4: Direct conversion using a 1:$\sqrt{2}$ transformer [8, p.2]

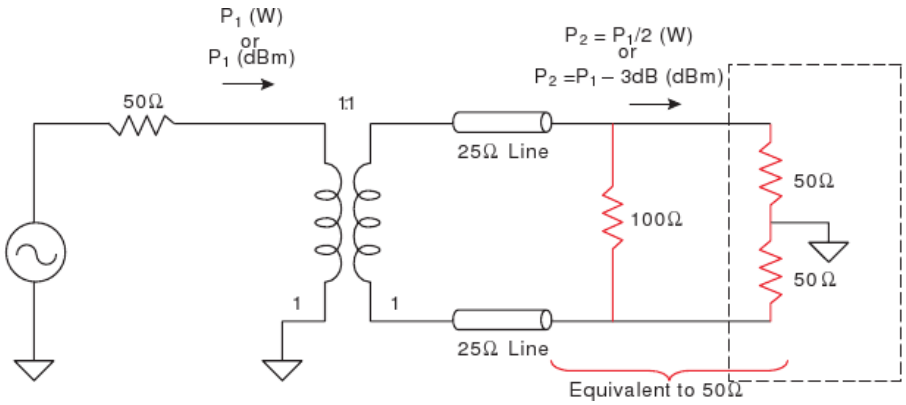


Figure 2.5: Conversion using a 1:1 transformer [8, p.3]

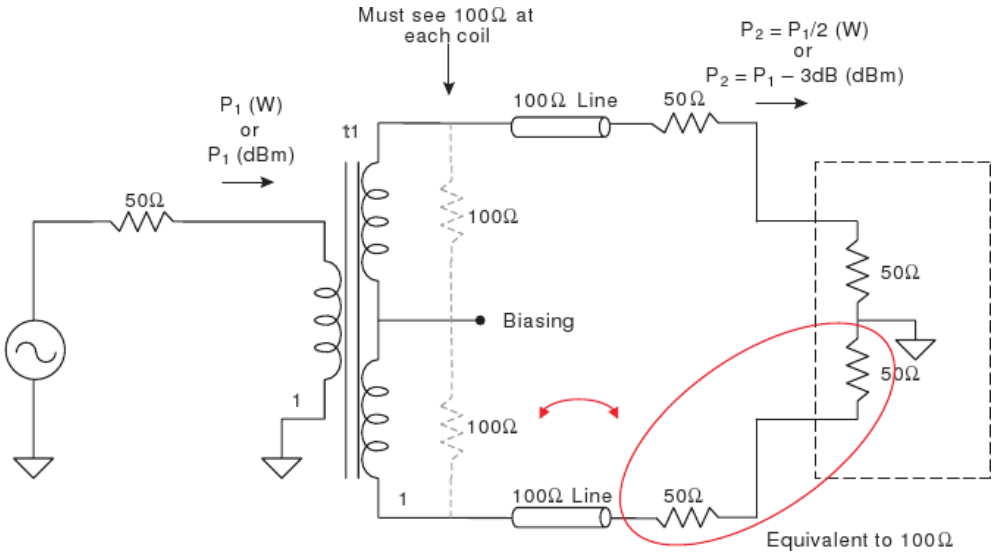


Figure 2.6: Conversion using a 1:1 transformer with double secondary [8, p.4]

is that the middle point can be used for biasing. [8, p.4]

Conversion using a 1:1 transformer with twisted cable

The fourth technique uses a 1:1 transformer on its side, as shown in Figure 2.7. The advantage is that the primary and secondary are well isolated from each other. The disadvantages are that half the power is lost in the conversion, extra resistors are needed for matching, and that symmetry at low frequencies is lost. [8, p.5]

Conversion using a 1:1 twisted pair transformer

The fifth technique uses a 1:1 twisted pair transformer, as shown in Figure 2.8. The advantage of this configuration over the previous one is that symmetry is retained at low frequencies. [8, p.6]

Transmission Line Transformers

Transmission line transformers act as a transformer but without any winding. They use transmission lines of specific lengths and impedances to create this

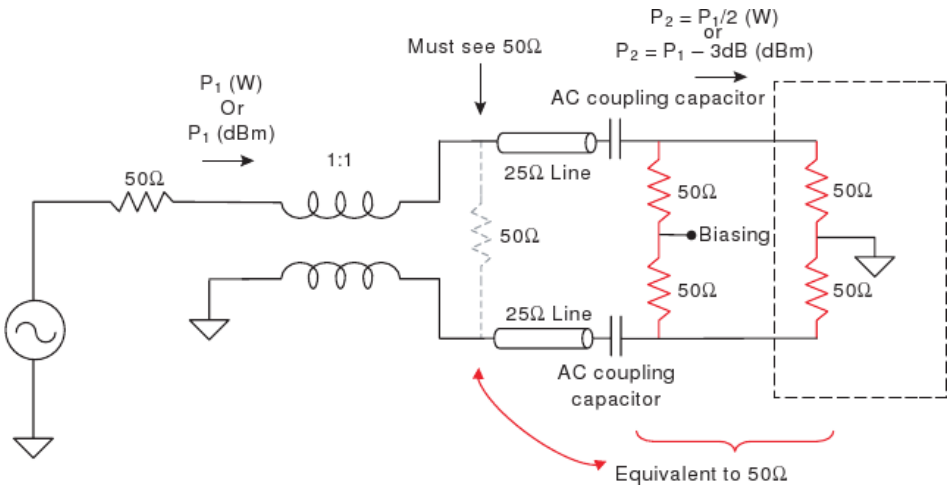


Figure 2.7: Conversion using a 1:1 transformer with twisted cable [8, p.5]

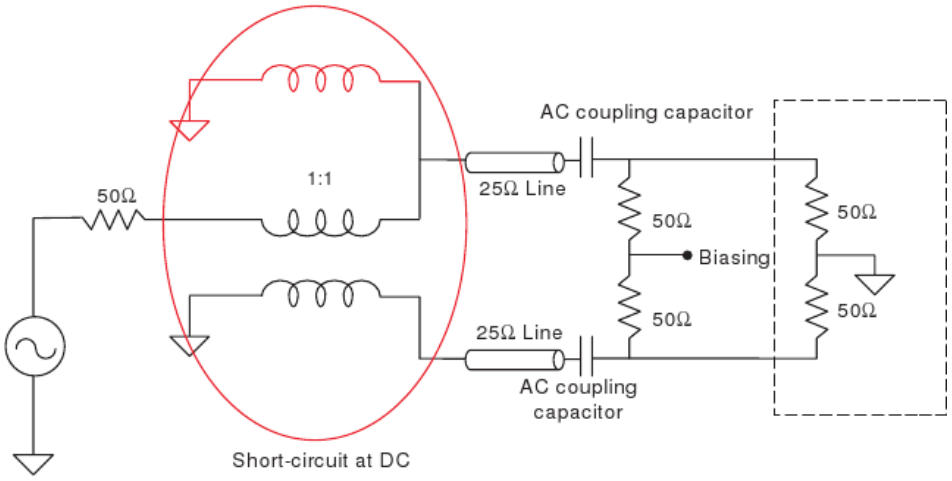


Figure 2.8: Conversion using a 1:1 twisted pair transformer [8, p.6]

effect. The disadvantage is that they have a very narrow bandwidth, and therefore are not suitable for this application. [28]

Active Devices

It is possible to convert single ended signal to differential signals using actively powered devices, such as amplifiers. The basic idea is to use one opamp as a buffer which outputs the input signal on the positive output, and use another opamp as an inverting buffer which inverts the input signal by 180° and outputs it on the negative output.

The advantage of this method is that power isn't lost because the signal is being amplified. The disadvantage is that low noise amplifiers are required in order to keep the signal clean. Since the input signal to the ADC includes high frequencies and has a large bandwidth, special amplifiers would be required.

2.4.2 Mini-Circuits TC1-1-13M

The TC1-1-13M is a 50 Ω RF transformer manufactured by Mini-Circuits[6] with a wide frequency range of 4.5–3000 MHz. At 1.5 GHz it has an insertion loss of 0.90 dB, an amplitude imbalance of 0.29 dB, and a phase imbalance of 2°. These are less than the maximum imbalances the ADC083000 can tolerate (1 dB amplitude imbalance and 2.5° phase imbalance [14, p.30]). The TC1-1-13M is surface mounted and can handle 24 dBm of RF power. [12]

2.5 Temperature Monitoring

The ADC083000 contains an on-chip temperature diode which can be used to measure the temperature of the chip. This is useful since the performance of the ADC is influenced by the temperature. The temperature is determined by measuring the voltage across the diode. Since the Z-DOK only allows for

[6] `http://www.minicircuits.com`

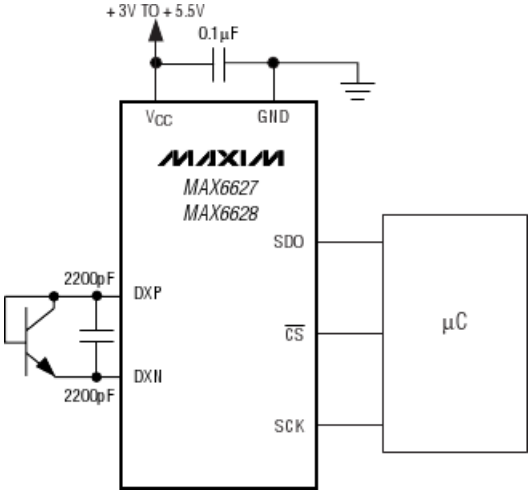


Figure 2.9: MAX6627 Typical Operating Circuit [11, p.1]

digital signals, this measurement must be done on the board and provided in digital format to the iBOB. The component chosen for this task is the MAX6627.

2.5.1 Maxim MAX6627

The MAX6627 is a remote temperature sensor manufactured by Maxim[7]. It calculates the temperature of a remote sensor (in this case, the on-chip diode on the ADC083000) and provides the temperature over SPI. It samples the temperature every 0.5 s, and has an accuracy of ±1 °C over the range 0–125 °C. The typical operating circuit from the datasheet is shown in Figure 2.9. [11, p.1]

The electrical characteristics of the serial interface were examined and determined to be compatible with the FPGA on the iBOB. This means that a level converter is not necessary. [11, p.3]

[7] `http://www.maxim-ic.com`

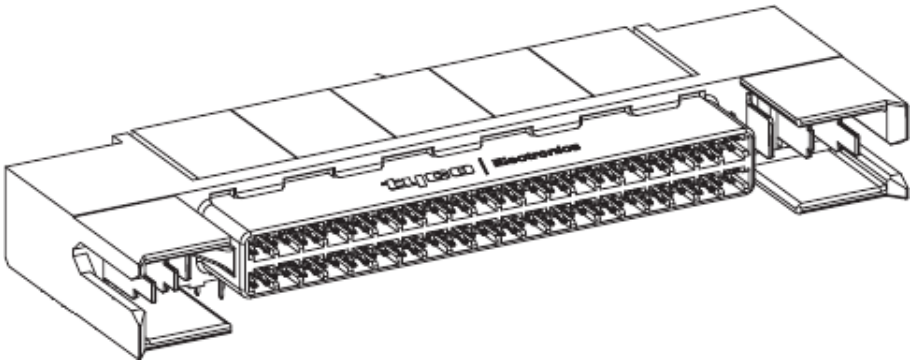


Figure 2.10: Z-DOK Adapter [27, p.2]

2.6 Connectors

The ADC board contains four connectors: one Z-DOK connector (for power and interfacing to the iBOB) and three SMA connectors (for the inputs). These connectors are now presented.

2.6.1 Z-DOK Connector

The Z-DOK is a connector manufactured by Tyco Electronics[8], and is designed specifically for connecting differential pairs with a differential impedance of 100 Ω [27, p.1]. The iBOB uses the Z-DOK connector with 40 differential pairs and 3 utility contacts per side [4]. The Z-DOK adapter is shown in Figure 2.10. The PCB land pattern is specified in the datasheet [27].

The utility contacts are used for power supplies and ground. The differential pairs are connected to the FPGA on the iBOB. These differential pairs could be used for any purpose, but in order to make writing the firmware easier they should be kept compatible with the iADC. The power and data connections are shown in Table 2.1.

2.6.2 SMA Edge Connector

SMA connectors are coaxial RF connectors with an impedance of 50 Ω. They provide excellent performance from DC to 18 GHz. [30]

[8] `http://www.tycoelectronics.com`

Table 2.1: Z-DOK Connections [4]

Pin Number(s)	Signal
A1–A16	Byte 0
B1–B20	Ground
C1–C16	Byte 2
D1–D16	Byte 1
E1–E20	Ground
F1–F16	Byte 3
W1–Z1	Ground
W2–Z2	5V
W3–Z3	3.3V
W4–Z4	2.5V
W5–Z5	1.8V
W6–Z6	Ground

Figure 2.11: SMA Edge Connector [19]

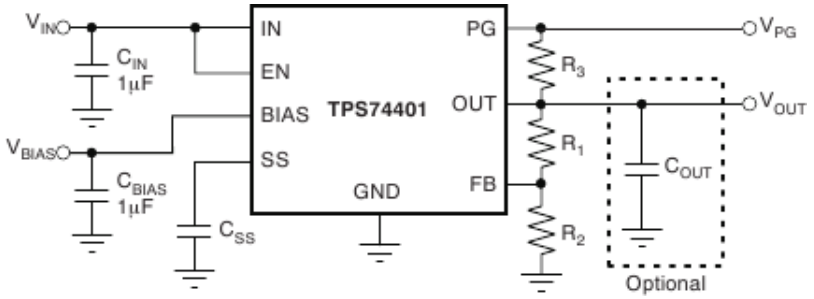


Figure 2.12: TPS74401 Typical Application Circuit [24, p.1]

Edge connectors are mounted on the edge of the board, and allow the signal to travel straight in instead of via a right angle. Figure 2.11 shows such an edge connector.

2.7 Power Supply

Power is supplied to the board through the Z-DOK connector in the form of 1.8 V, 2.5 V, 3.3 V and 5 V supplies. The components on the board require 1.9 V and 3 V supplies, and so regulators will have to be used to provide these voltages. The chosen regulator is the TPS74401.

2.7.1 Texas Instruments TPS74401

The TPS74401 is a *low dropout* (LDO) linear regulator manufactured by Texas Instruments[9]. The output voltage is adjustable from 0.8 V to 3.6 V and is accurate within 1%. It also features programmable soft-start and can provide up to 3.0 A of current. [24, p.1]

The typical application circuit from the datasheet is shown in Figure 2.12. The output voltage of the regulator is set using two resistors according to Equation 2.2. The soft-start time is set using a capacitor according to Equation 2.3. In order to achieve a dropout voltage less than 200 mV, the V_{BIAS} pin should be 1.62 V higher than the output voltage. [24, p.3–4]

[9] `http://www.ti.com`

$$V_{\mathrm{OUT}} = 0.8 \times (1 + \frac{R1}{R2}) \tag{2.2}$$

$$t_{\mathrm{SS}} = 0.8 \times \frac{C_{\mathrm{SS}}(F)}{7.3} \times 10^{-7} \tag{2.3}$$

2.7.2 Decoupling Capacitors

Decoupling capacitors are placed near the power pins of each IC. They are used to smooth the power supply. This is because the chips often draw large currents, but for an extremely short time. The capacitors store energy and then supply this energy during these brief bursts. This prevents these large currents from creating ripples on the power supply, which could negatively influence other parts of the circuit.

2.8 Level Converters

The ADC083000 uses a 1.9 V supply voltage, and therefore requires 1.9 V logic on the serial interface and control pins. However, the iBOB uses 2.5 V logic. The voltage levels therefore need to be converted in order to make them compatible. The SN74AUC34 will be used to convert the 2.5 V logic from the iBOB to 1.9 V, and the SN74LVC2G07 will be used to convert the 1.9 V logic from the ADC083000 to 2.5 V.

2.8.1 Texas Instruments SN74AUC34

The SN74AUC34 is a hex buffer gate manufactured by Texas Instruments. Each of the six gates simply outputs the same logic value which is being input. The inputs are tolerant of voltages higher than the supply voltage, but the output voltage will only go as high as the supply voltage. This is why it can be used as a level converter. [22, p.1]

2.8.2 Texas Instruments SN74LVC2G07

The SN74LVC2G07 is a dual buffer/driver manufactured by Texas Instruments. Each buffer simply outputs the same logic value which is being input. The buffers are open-drain, and so a pullup resistor on the outputs can be used to set the desired output voltage. [23, p.1]

2.9 PCB

Various issues regarding the PCB will be significant for this application. These relate to compatibility with the iBOB and adherence to the manufacturer's capabilities. Due to the high frequencies on the board, impedance and length matching are important, as well as the substrate material and stackup.

2.9.1 Form Factor

In order to be compatible with the iBOB, the board must have the same form factor as the iADC board. The relevant parts of the iADC board are shown in Figure 2.13 (the origin is on the bottom left corner of the board). This means that the board must be 127 mm×85.85 mm, and contain four mounting holes with a diameter of 3.175 mm at the following coordinates (units in mm): (4.826; 20.32), (4.826; 111.76), (81.026; 20.32), (81.026; 111.76). The Z-DOK connector must be positioned in the centre of the bottom edge, and the SMA connectors must be positioned on the top edge. [3]

2.9.2 Manufacturer Capabilities

In order to be successfully manufactured, the board must adhere to the manufacturer's capabilities. These capabilities pertain to the sizes and accuracies which are obtainable. The board will be manufactured by Trax Interconnect[10], and their capabilities are listed in their Short Form Specification [25].

[10] `http://www.trax.co.za`

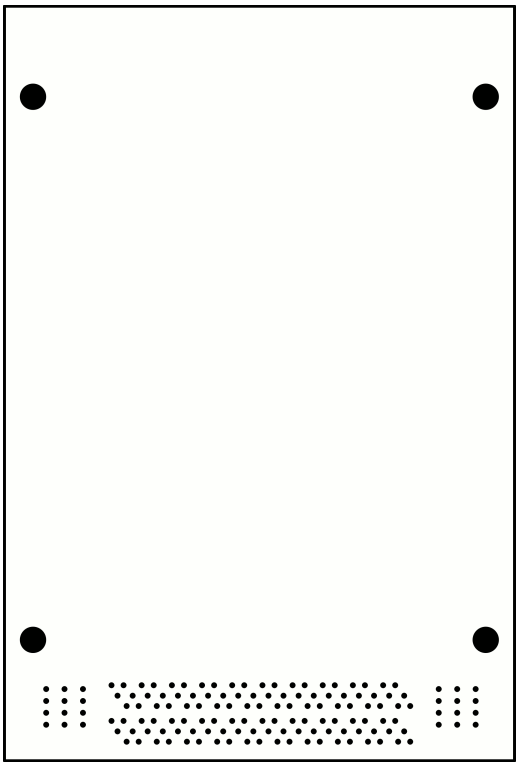


Figure 2.13: Outline, mounting holes and Z-DOK connector on iADC board [3]

2.9.3 Impedance Matching

At the high speeds and high frequencies which the ADC board operates at, the traces on the PCB act like transmission lines. If the impedance of the lines are not matched correctly, there will be reflections on the transmission line. These reflections reduce the received power and distort the signal.

The traces carrying the single ended input signal and the clock signals must have a characteristic impedance of $50\,\Omega$. The differential traces (which include the output data and the differential input and clock signals) must have a differential impedance of $100\,\Omega$.

The impedance of the traces are affected by the following four factors.

1. The width of the trace
2. The thickness of the trace
3. The dielectric constant of the substrate
4. The thickness of the substrate

The impedance of the differential traces is additionally affected by the distance between the two traces.

2.9.4 Substrate Material

The most common substrate used in PCBs is FR-4. The problem with FR-4 however is that the dielectric constant is not very constant at higher frequencies. Figure 2.14 shows the dielectric constant of FR-4 — it can be seen to decrease significantly as the frequency increases.

There are other materials available which are designed for use in high frequency PCBs. One such material is RO4003, manufactured by Rogers Corporation[11]. The dielectric constant of RO4003 is also shown in Figure 2.14, and can be seen to be much more constant than FR-4.

RO4003 is a glass reinforced hydrocarbon/ceramic laminate, and has a low dielectric constant which is stable over both temperature and frequency. It can be used in standard FR-4 PCB manufacturing processes. [20, p.1]

[11] `http://www.rogerscorporation.com`

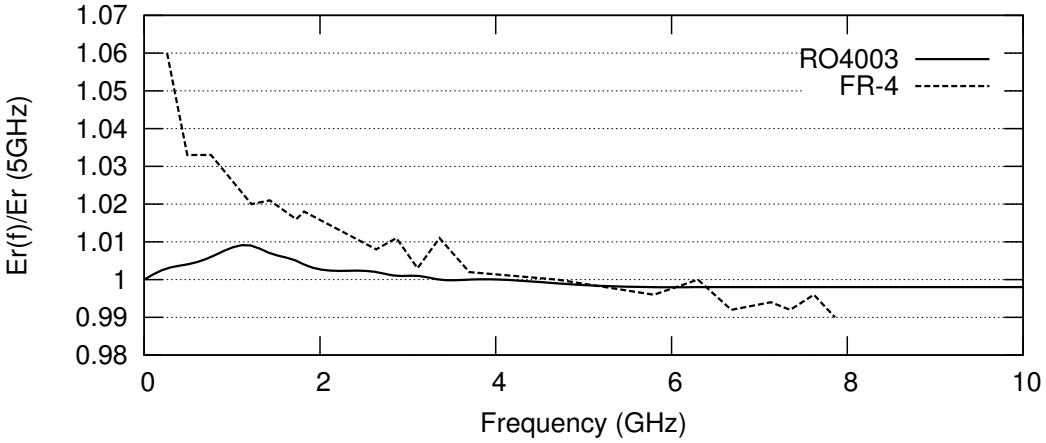


Figure 2.14: Graph of dielectric constant vs frequency for RO4003 and FR-4 (redrawn from [20, p.3])

Figure 2.15: Standard stackup for four layer PCBs

2.9.5 Stackup

The standard stackup of a four layer board is shown in Figure 2.15. Laminate is a piece of cured substrate with a layer of copper on each side. The two inner layers are etched onto the copper layers of the laminate. Copper foil is then added to each side and bonded to the laminate with prepreg (which is uncured substrate). The outer layers are etched onto the copper foil. [26, p.2]

However, it was discovered that Trax did not stock RO4003 prepreg. This was a problem since the outer layers would be the signal layers and they needed to be placed on RO4003 substrate. A non-standard stackup

Figure 2.16: Non-standard stackup used for the board

therefore had to be used (illustrated in Figure 2.16). Two pieces of laminate are used: one containing the first and second layers, and the other containing the third and fourth layers. The two pieces of laminate are then bonded with FR-4 prepreg.

2.9.6 Length Matching

Since the outputs of the ADC run at very high speeds, the delay experienced by signals on the lines is significant. The signals must reach the FPGA on the iBOB at exactly the same time otherwise the values received by the FPGA will be incorrect.

The rule of thumb is to ensure that the lines are matched within one tenth of the wavelength. In the case of *single data rate* (SDR), the outputs are running at 750 MHz. This corresponds to a wavelength of $\frac{c}{750\,\mathrm{MHz}} = 400\,\mathrm{mm}$. The output lines should therefore be matched within 40 mm.

2.10 iBOB Firmware

The hardware on the iBOB is briefly outlined, and the development kit is described. The requirements of the firmware is then presented, followed by detailed descriptions of the serial interfaces of the three chips. The format of the data buses and the control lines are provided.

2.10.1 iBOB Hardware

The iBOB board contains a Xilinx Virtex-II Pro 2VP50 FPGA, 1 MiB SRAM, an RS232 interface, a 10/100 Ethernet interface and two 10GigE interfaces. It has two Z-DOK connectors, and hence can utilise two ADC boards simultaneously. [5] The Virtex-II Pro 2VP50 contains two embedded Power PC processors, and so is programmed using both a *hardware description language* (HDL) and C.

2.10.2 Development Kit

Xilinx provide software to compile code for their FPGAs. The logic design (compiling HDL) is done by ISE, whereas the C code and overall system design is done by the *Embedded Development Kit* (EDK). These development kits provide numerous IP cores, which are pre-written functions which can be used in user designs.

CASPER have developed a Simulink-based tool to generate firmware for the iBOB. The tool allows the user to create a block diagram of the desired system, using various modules which have already been developed. The flow and control of data is set up in in this block diagram, as well as the configuration of the system. This tool then generates the VHDL and C code, which is compiled using the software provided by Xilinx.

2.10.3 Firmware Requirements

The firmware will need to receive the sampled data from the ADC, which is provided on four buses. The four buses need to be interleaved in order to restore the correct order. This data must then be transmitted over the 10GigE network to the receiving computer.

The firmware must also configure the ADC board on startup. This involves programming the ADC083000 and LMX2531 using the serial interface, and calibrating the ADC. The firmware should also monitor the temperature of the ADC by retrieving the temperature from the MAX6627.

Table 2.2: ADC083000 Registers [14, p.25]

Address (Hex)	Description
0x1	Configuration
0x2	Offset
0x3	Full-scale Voltage Adjust
0xD	Extended Clock Phase Adjust (fine)
0xE	Extended Clock Phase Adjust (coarse)
0xF	Test Pattern

2.10.4 Sampled Data

The output data of the ADC is demultiplexed and provided on four 8 bit buses. The `DCLK` output determines when the data buses must be latched. The data must be interleaved in the order `Da`, `Db`, `Dc`, `Dd`. The buses will be clocked with DDR.

2.10.5 Serial Interface

The serial interface will be used to program the ADC083000 and LMX2531, and to retrieve the temperature of the ADC from the MAX6627. All three devices use a three wire interface, consisting of a clock, data and chip select line.

ADC

The ADC083000 is programmed by accessing six 16 bit write-only registers. Each register access begins with the header `0000 0000 0001`, followed by the 4 bit register address. The 16 bits to be written to the register are then transmitted. The chip select line is active low, data is clocked on the rising edge of the clock, and data is sent *most significant bit* (MSB) first. The six registers are described in Table 2.2. The detailed register descriptions can be found in the datasheet. [14, p.25–27]

Frequency Synthesiser

The LMX2531 is programmed by accessing eleven 20 bit write-only registers. Each access begins with the data to be written followed by the 4 bit register address. The chip select line (known as latch enable) is active low, data is clocked on the rising edge of the clock, and data is sent MSB first. The register values are fairly complicated, and involve configuring the PLL, charge pump, phase detector, dithering, loop filter and FastLock. The registers must also be programmed in a specific order. The detailed register descriptions and instructions can be found in the datasheet. [16, p.13–24]

Temperature Monitor

The MAX6627 provides access to a single read-only register. Output is initiated by asserting the chip select low, after which 16 bits are output on the falling edge of the clock. The first 13 bits represent the temperature in fixed-point two's complement with four fractional bits. [11, p.6]

2.10.6 Control Lines

Calibration of the ADC083000 can be initiated by holding the `CAL` line low for 80 input clock cycles and then holding it high for another 80 input clock cycles. Eighty input clock cycles corresponds to $\frac{80}{1.5\,\text{GHz}} = 54\,\text{ns}$ at 1.5 GHz. The `CALRUN` line is high when calibration is in progress. [14, p.22]

The ADC083000 can be disabled and powered down by holding `PD` high [14, p.23]. The `DCLK_RST` line can be used to synchronise multiple ADC boards [14, p.28]. The `LD` line shows whether the frequency synthesiser has locked onto the output frequency [16, p.4].

2.11 Performance Analysis

There are various calculations which can be used to characterise and compare the performance of ADCs. The performance of an ADC is determined by how ideal its transfer characteristic is, as well as how much noise is introduced

by the system. The various values which will be calculated are defined and explained below.

In the equations below, S is the signal power, N is the noise power, and D is the distortion power (all in Watts). These values are determined by processing the data with the FFT and reading the power from the output thereof.

2.11.1 Signal to Noise Ratio

The *signal to noise ratio* (SNR), as shown in Equation 2.4, is the ratio of the signal power to the noise power. This value shows how much unwanted noise has been added to the signal, and determines the system's performance at low input powers. [9]

$$\text{SNR} = 20 \log \frac{S}{N} \qquad \text{(dB)} \tag{2.4}$$

2.11.2 Total Harmonic Distortion

The *total harmonic distortion* (THD), as shown in Equation 2.5, is the ratio of the signal power to the distortion power. The distortion is due to unwanted signals filtering into the channel. This could be due to coupling between lines, or due to inter-modulation distortion. [9]

$$\text{THD} = 20 \log \frac{S}{D} \qquad \text{(dB)} \tag{2.5}$$

2.11.3 Signal to Noise And Distortion

The *signal to noise and distortion* (SINAD), as shown in Equation 2.6, is the ratio of signal power to the combined noise and distortion power. This represents how discernible the signal is from other unwanted components. [9]

$$\text{SINAD} = 20 \log \frac{S}{N + D} \qquad \text{(dB)} \tag{2.6}$$

2.11.4 Effective Number of Bits

The *effective number of bits* (ENOB), as shown in Equation 2.7, is the resolution that an ideal ADC with the same performance would have. Due to the non-ideal transfer function of real ADCs, the effective resolution is less than the actual resolution. [9]

$$\text{ENOB} = \frac{\text{SINAD} - 1.76\,\text{dB}}{6.02} \qquad \text{(bits)} \tag{2.7}$$

2.11.5 Spurious Free Dynamic Range

The *spurious free dynamic range* (SFDR) is the difference in power between the signal and the strongest unwanted signal. This determines the weakest signal which can be differentiated from the interference. [9]

2.12 Conclusions

Although the first few weeks of the project consisted mostly of this research, the research phase extended throughout the project. Numerous issues were uncovered as the project progressed, and these issues necessitated further research. This chapter presents all the information which was directly used while implementing the project and which is needed in order to understand the project.

Gathering information on the components was simple (although sometimes tedious) since it was all contained in the datasheets. Methods of analysing ADC performance were also easily found, mostly in application notes published by component manufacturers.

Some of the issues (such as phase noise, single to differential conversion and baluns) were more complex and required more in depth research and examination. New concepts had to be learnt and applied, and this needed more time. Most of these topics are very involved, and therefore could only be addressed very briefly.

There was not, however, much documentation available for the iBOB and the iADC. This necessitated examining schematics and layouts of these

boards (and sometimes datasheets of components on the boards) in order to discover the required information. Documentation regarding the iBOB firmware is virtually non-existent, and so most of this was gathered by consulting others or by directly examining the firmware.

The aspects regarding PCBs (and high frequency PCBs in particular) were the most foreign to me, since I have had very little experience with designing PCBs. A lot of this information was gathered from other people, and required many new concepts to be grasped. Most of these issues couldn't be researched in detail due to the voluminous amount of information available (which could easily be the topic of an entire thesis).

This project required a large amount of research on many different topics, as well as dealing with and using large amounts of information. This has helped to develop my research and learning skills, and is one of the positive outcomes of this project. Although it was a lengthy process, all the information required for this project was successfully gathered.

Chapter 3

System Design

This chapter details the design process that was followed during the project. The high level design is first executed, resulting in a block diagram of the system. The exact connections between the components are then determined in the circuit design. This is converted to a PCB during the board layout, and some simulations are done to verify the design. The firmware to run on the iBOB is then designed and implemented.

3.1 Introduction

The chapter begins with the high level design of the system. Each requirement of the system is investigated and the functions required in order to meet these requirements are determined. Support functions (such as the power supply and temperature monitoring) are also considered, and the external interfaces are defined. The logical connections between the components and the external interfaces are determined, which results in the overall architecture of the system.

This architecture is then implemented in the detailed circuit design. This determines the physical connections between all the components, as well as the passive components which are needed. The balun configuration is chosen, and the power supplies and level converters are designed. The frequency synthesiser, ADC and temperature monitor are then added and connected.

Lastly, the schematic is captured in DxDesigner.

The circuit is then realised as a PCB which is laid out using PADS Layout. First, the design rules and layer stackup are defined, taking the required impedance matching into account. The components are then placed on the board and the power planes are defined. The board is then routed in PADS Router. The decoupling capacitors are manually routed first, then the ADC outputs are autorouted, after which the other traces are routed manually.

HyperLynx is then used to ensure that the impedances are correctly matched. Signals on the ADC outputs are also simulated in order to check the signal integrity and the influence of cross talk.

The board is then prepared for production. The components are labelled on the silkscreen. The design rule checks are run in order to verify that the design is correct. Finally, the Gerber files to be used in production are generated and checked.

The firmware for the iBOB is adapted from existing firmware. Since the board uses the same pin connections as the iADC, the data collection routine is compatible. The control pins are configured as *general purpose input/output* (GPIO) and made accessible to the processors. The SPI protocol is implemented in C and used to program the ADC and frequency synthesiser. Finally, the initialisation routine is written.

3.2 High Level Design

Each requirement of the system is investigated and the functions required in order to meet these requirements are determined. Support functions (such as the power supply and temperature monitoring) are also considered, and the external interfaces are defined. The logical connections between the components and the external interfaces are determined, which results in the overall architecture of the system, as shown in Figure 3.1.

The ADC implements the principal function of the board. The frequency synthesiser generates the sampling clock required by the ADC. The temperature monitor is used to read the temperature of the ADC chip.

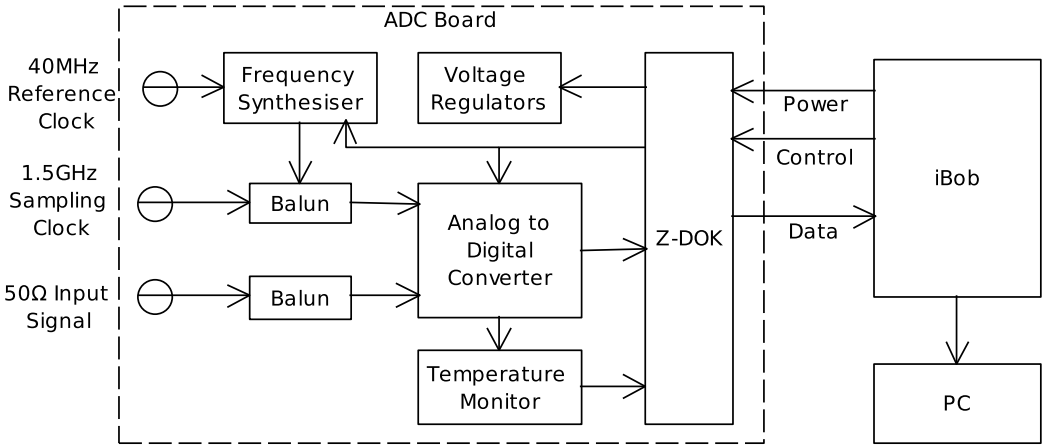


Figure 3.1: System Block Diagram

The input signal is connected to the board with an SMA connector. Since the ADC requires a differential signal, the input signal is passed through a balun in order to convert it. The frequency synthesiser requires a reference clock, which is also connected with an SMA connector. The frequency synthesiser generates the 1.5 GHz sampling clock, which is converted to a differential signal using a balun before being sent to the ADC. The board also allows an external sampling clock to be used instead of the frequency synthesiser, which is also connected with an SMA connector.

The Z-DOK connector provides the connection to the iBOB board. The output of the ADC is connected directly to the Z-DOK. Power is supplied to the board through the Z-DOK, and converted to the necessary voltages using voltage regulators. The ADC and frequency synthesiser are programmed using an SPI bus which is passed through the Z-DOK. The ADC has additional control pins which are also connected to the Z-DOK. The temperature monitor is also connected to the SPI bus.

3.3 Circuit Design

The circuit design determines the physical connections between all the components, as well as the passive components which are needed. The balun configuration is chosen, and the power supplies and level converters are de-

signed. The frequency synthesiser, ADC and temperature monitor are then added and connected. Lastly, the schematic is captured in DxDesigner. The final schematic is contained in Appendix A.

3.3.1 Balun

The baluns are used to convert single ended signals to differential signals. The fourth technique from Section 2.4.1 (shown in Figure 2.7) was chosen because 1:1 baluns are easily available and because it provides good isolation. Two baluns are used: one to convert the input signal, and the other to convert the sampling clock.

In order to select between the external sampling clock and the frequency synthesiser, $0\,\Omega$ resistors are used to connect both these signals to the balun. Only one of these resistors is actually placed on the board though, depending on which clock is being used.

3.3.2 Input Power

Now that the configuration of the front-end is known, the maximum input power can be calculated. The maximum full-scale range of the ADC083000 is $V_{\text{P–P}} = 840\,\text{mV}$ [14, p.23]. This is converted to an RMS value (Equation 3.1), and then to a power level assuming $100\,\Omega$ termination (Equation 3.2).

$$V_{\text{P–P(RMS)}} = \frac{840\,\text{mV}}{\sqrt{2}} = 594\,\text{mV} \tag{3.1}$$

$$P_{\text{ADC}} = 10 \times \log \frac{(594\,\text{mV})^2}{100\,\Omega \times 1\,\text{mW}} = 5.48\,\text{dBm} \tag{3.2}$$

Since the balun only outputs half the power, the maximum input power is twice the maximum power for the ADC. The maximum input power is thus $V_{\text{IN(MAX)}} = 5.48\,\text{dBm} + 3.01\,\text{dB} = 8.49\,\text{dBm}$. The input signal should have a maximum frequency of 1.5 GHz in order to prevent aliasing.

3.3.3 Power Supply

The ADC is powered from 1.9 V, which is split into two supplies: V_{A} (which powers the analog circuitry) and V_{DR} (which powers the digital outputs). Since the digital outputs will create noise, it was decided to use two separate regulators in order to isolate these two supplies. The frequency synthesiser, temperature monitor and level converters all use 3 V supplies. These can all be run from a single regulator.

The datasheet [24, p.4] provides a table of resistor values for common output voltages. Since the required values weren't in the table, the values were adjusted slightly using Equation 2.2. For the 1.9 V output, the values are $R1 = 3.3\,\mathrm{k}\Omega$ and $R2 = 2.4\,\mathrm{k}\Omega$. For the 3 V output, the values are $R1 = 3.3\,\mathrm{k}\Omega$ and $R2 = 1.2\,\mathrm{k}\Omega$.

A 33 µF capacitor is placed on the output of each power supply. This value was recommended by the ADC083000 datasheet [14, p.33], and for convenience was used on the 3 V supply as well. The regulators don't need a capacitor on the input since the power supplies from the iBOB should be fairly stable.

3.3.4 Level Converters

The SN74AUC34 is used to convert the 2.5 V signals from the iBOB down to 1.9 V signals for the ADC083000. The SN74AUC34 is supplied from the 1.9 V digital power supply. The following signals from the Z-DOK are connected to the inputs: `SCLK`, `SDATA`, `CS_ADC`, `CAL` and `PDOWN`. The level converted outputs are routed to the respective pins on the ADC083000.

The SN74LVC2G07 is used to convert the 1.9 V signals from the ADC083000 up to the 2.5 V signals for the iBOB. The SN74LVC2G07 is supplied from the 1.9 V digital power supply. The `CALRUN` signal from the ADC083000 is connected to the input. The output pin is connected to the relevant pin on the Z-DOK, and to the 2.5 V supply via a 100 kΩ pullup resistor.

3.3.5 Frequency Synthesiser

The LMX2531 was connected as shown in Figure 2.3. The chip enable (CE) pin is connected to 1.9 V via a 0 Ω resistor, which allows the chip to be disabled by not placing the resistor. The serial interface lines were connected directly to the Z-DOK. The reference oscillator input `OSCin` is connected directly to the SMA connector. The oscillator output Fout is connected to the balun. The values for the loop filter (`C1_LF`, `C2_LF`, `R1_LF` and `R2_LF`) were taken from the ADC083000 Evaluation Board [15, p.2].

3.3.6 ADC

The ADC083000 was connected according to the pin descriptions in the datasheet [14, p.3–7]. The output data pins were connected directly to the Z-DOK. The pins requiring level conversion are described in Section 3.3.4. The `ECE` pin was connected to half the supply voltage using a resistor divider in order to enable Extended Control Mode. The input and clock pins were connected to the output of the baluns. The differential reset pins `DCLK_RST+/-` were connected to the Z-DOK, and `DCLK_RST` left unconnected. V_{CMO} is connected to ground (since the input is AC coupled).

3.3.7 Temperature Monitor

The MAX6627 was connected as shown in Figure 2.9. The `DXP` and `DXN` pins were connected to the `Tdiode` pins on the ADC083000. The serial interface pins were connected directly to the Z-DOK.

3.3.8 Decoupling Capacitors

All power supply pins on the ADC, frequency synthesiser, temperature monitor and level converters were decoupled with 100 nF capacitors. This value was chosen because it is recommended in the ADC083000 datasheet [14, p.33] and used in the ADC083000 evaluation board [15, p.2,4].

3.3.9 Schematic Capture

The circuit was then captured using DxDesigner. The parts library was first setup, and then symbols for the ICs and connectors were created. The symbols were then added to the schematic and connected as determined in this section.

3.4 Board Layout

The circuit is then realised as a PCB which is laid out using PADS Layout. First, the design rules and layer stackup are defined, taking the required impedance matching into account. The components are then placed on the board and the power planes are defined. The board is then routed in PADS Router. The decoupling capacitors are manually routed first, then the ADC outputs are autorouted, after which the other traces are routed manually. The final board layout is contained in Appendix B.

3.4.1 Layers

For good noise performance, it is important to have a large ground plane. Due to the number of traces between the ADC and the Z-DOK, the board will require at least two signal layers. Since PCBs must have an even number of layers, the board will consist of four layers. The layers are therefore as follows:

1. Top (signal)
2. Power plane
3. Ground plane
4. Bottom (signal)

In order to achieve the impedance matching described in Section 3.4.2, it was determined (using simulation) that the substrate on the top and bottom layers should be as thin as possible. According to the manufacturer's

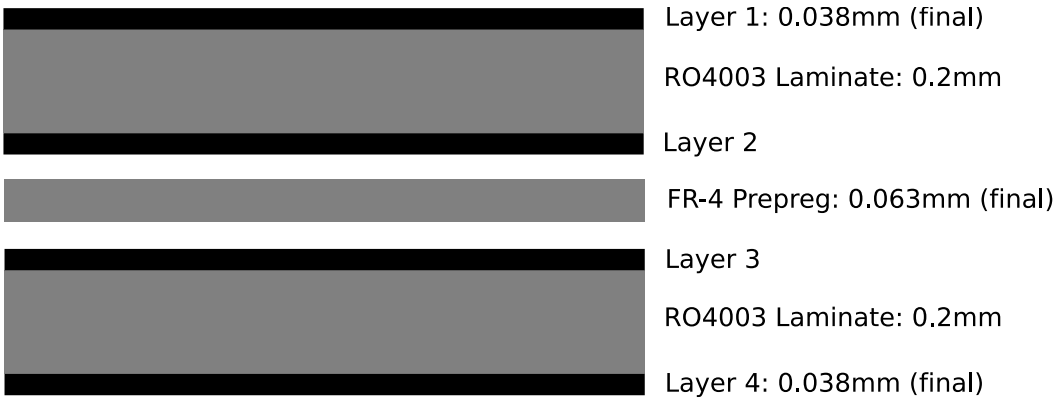


Figure 3.2: Final stackup used for the board

specifications [26, p.2], the thinnest laminate available is 0.2 mm. A thin substrate between the power and ground planes is also desirable since this increases the capacitance. The thinnest prepreg available is 0.063 mm. The thinnest copper thickness of 0.038 mm was used. The final stackup in shown in Figure 3.2.

3.4.2 Impedance Matching

As described in Section 2.9.3, the input and clock traces must have a characteristic impedance of 50 Ω, and the differential pairs must have a differential impedance of 100 Ω. HyperLynx was used to simulate the board, and the correct trace widths and separation were determined by trial-and-error. The differential lines need a width of 0.25 mm and a separation of 0.13 mm. The 50 Ω traces need a width of 0.43 mm.

3.4.3 Design Rules

The design rules were entered into PADS Layout according to the manufacturer's specifications [25]. Many values required by PADS Layout weren't found in the specification (such as some clearances), and so these had to be estimated.

Extra design rules were entered for the differential pairs, defining the trace width and separation. The ADC outputs were added to a class and a

rule added to match their lengths.

3.4.4 Placement and Planes

The outline of the board was defined according to the measurements in Section 2.9.1. The mounting holes were then created and placed, and the Z-DOK connector was placed (since these have fixed positions). The other components were then placed on the board. A large area was left between the ADC and the Z-DOK since space is needed to increase the length of short traces. The frequency synthesiser was placed close to the ADC and the SMA connector. The baluns were placed close to the single ended signals (in order to convert them to differential signals as soon as possible).

The voltage regulators were placed near the edge of the board, in line with the side of the Z-DOK connector from which their input voltage comes in. Three planes were then defined on the power layer, corresponding to each of the power supplies.

3.4.5 Decoupling Capacitors

The decoupling capacitors were placed and routed first. This is because they need to be very close to the actual pins on the ICs. The capacitors were placed on the bottom layer of the board (since space is needed on the top layer for traces to exit from the IC), usually directly underneath the IC.

In order to ensure that the current runs over the capacitor, the pins were connected directly to the capacitor using a via. This via is not connected to the power or ground planes which it passes through. A second via then connects the capacitor to the power or ground plane. This means that current must flow from the power plane to the capacitor through the second via, and then to the component pin through the first via (and in reverse for ground connections).

3.4.6 ADC Outputs

The ADC outputs were mainly routed using the autorouter. This is because the trace lengths had to be closely matched, and this is a difficult task to do manually. However, it was discovered that when all the outputs were autorouted together, the autorouter tried to obtain a much longer length than was actually required. In order to overcome this, 6–10 of the longest lines were first routed manually, and the remainder then autorouted. This achieved a significantly shorter trace length.

In order to achieve the matching, the lines were first autorouted with a matching constraint of 30 mm. The constraint was then decreased and the autorouter run again (using the Optomize command) in order to achieve the new constraint. This was repeated until the lines were matched within 3 mm.

3.4.7 Other Traces

The other traces were routed manually using interactive routing in PADS Router. There were no special requirements for these traces.

3.5 Simulation

HyperLynx is then used to ensure that the impedances are correctly matched. Signals on the ADC outputs are also simulated in order to check the signal integrity and the influence of cross talk.

3.5.1 Setup

The board was exported from PADS Layout to HyperLynx. The stackup of the board was then defined by entering the thicknesses of each substrate and copper layer and the dielectric constant of the substrate. Models for the components then had to be entered. An *input/output buffer information specification* (IBIS) model for the ADC083000 was found on the National Instruments website. Each pair of pins on the Z-DOK were defined as a 100 Ω differential terminator.

3.5.2 Impedance Matching

The impedance of the input and clock traces was determined by viewing the properties of the trace. The impedance of these lines was 50.2 Ω. The differential impedance of the differential pairs was determined by enabling cross talk simulation and choosing "Walk Coupling Regions" for the trace. The differential impedance of these lines was 99.6 Ω.

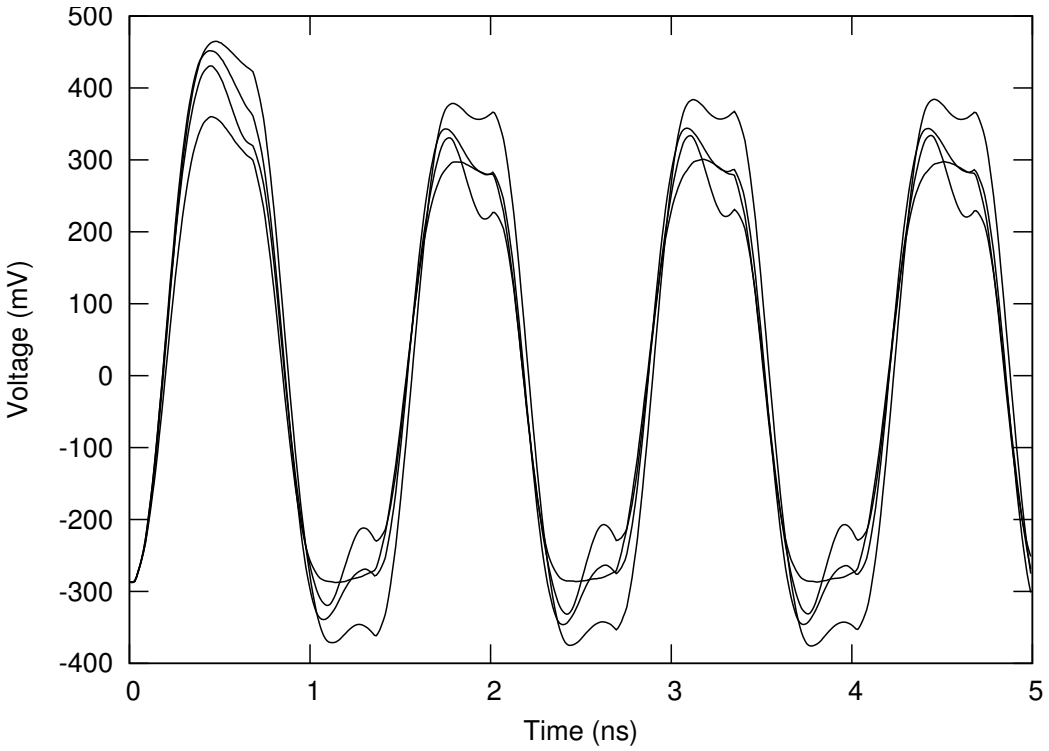
3.5.3 Signal Integrity

Signal integrity on the ADC outputs was tested by simulating some of the output lines with crosstalk simulation. The outputs were tested at a frequency of 750 MHz. One of these simulations is shown in Figure 3.3. Figure 3.3a shows the signals at the ADC pins, and Figure 3.3b shows the signals at the Z-DOK pins. Each line corresponds to a differential pair (i.e. each line represents a differential voltage).

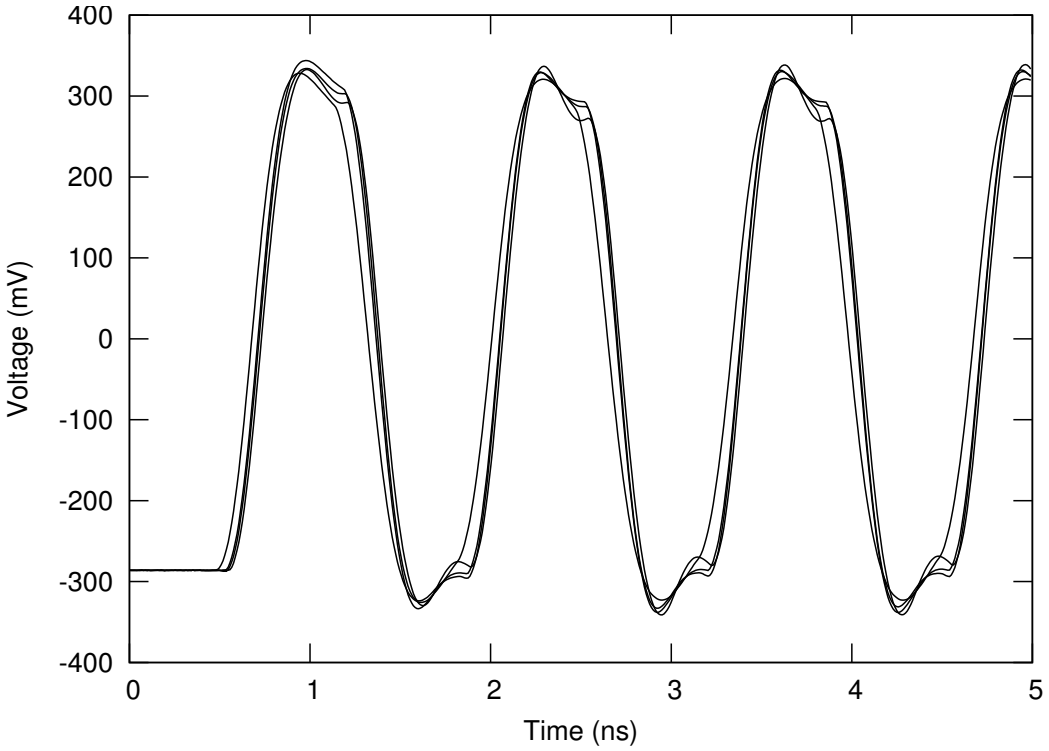
All the outputs have clear transitions (no jagged lines). The response at the peaks is not very flat (this is actually because it doesn't have time to settle before the next transition). The ripples aren't however very big, and shouldn't cause the wrong value to be received.

In order to find how much crosstalk is induced into the line, the output for that line is disabled in the ADC model. This means that the line should have a constant voltage — any variation will be due to crosstalk. Figure 3.4 shows the result of this simulation. Minor fluctuations can be seen in the line, showing that crosstalk is present. The peak-to-peak voltage of this line at the Z-DOK pins is 15.17 mV, and at the ADC pins is 30.29 mV. Since that the normal peak-to-peak voltage is approximately 660 mV, this crosstalk creates a maximum of 4.5% deviation, which should not cause any problems.

Simulations of some of the other lines were done, and no problems were found. Appendix D contains graphs of another set of simulations.

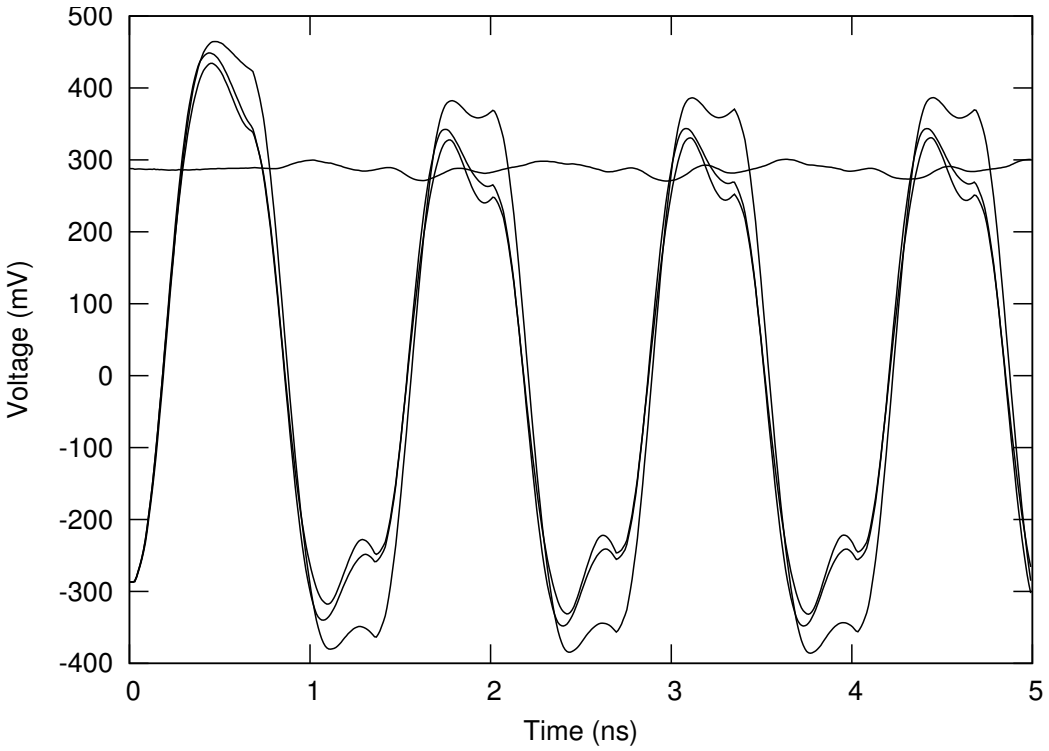


(a) At ADC pins

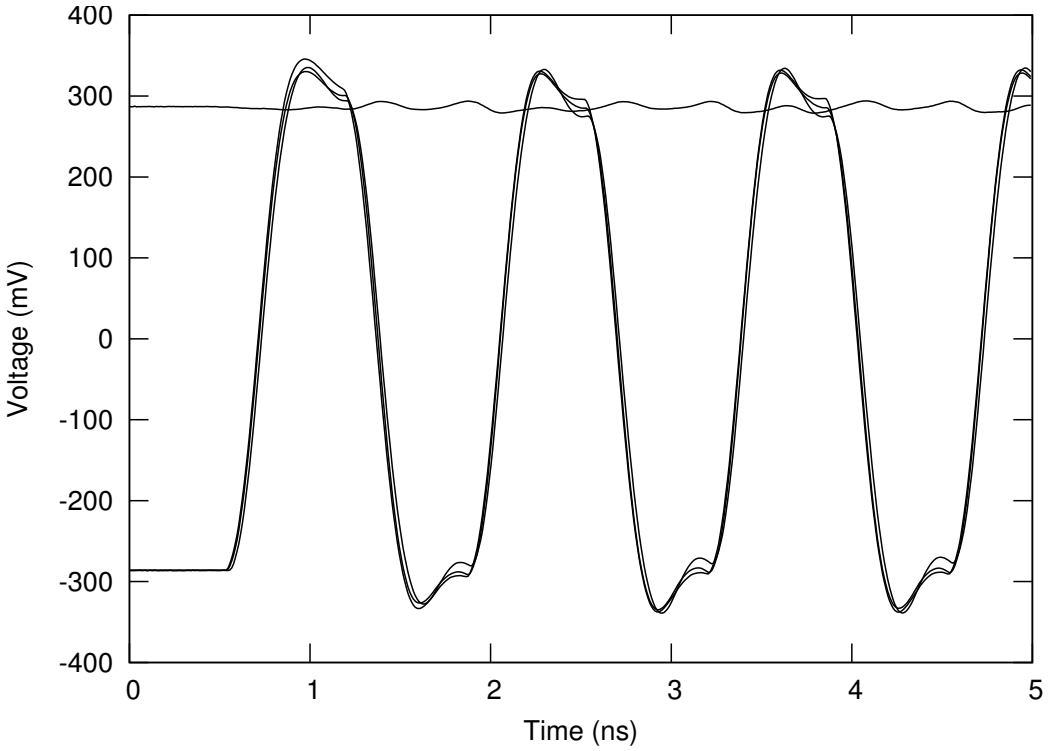


(b) At Z-DOK pins

Figure 3.3: Simulation of ADC outputs



(a) At ADC pins



(b) At Z-DOK pins

Figure 3.4: Simulation of ADC outputs showing crosstalk

3.6 Preparing for Production

The board is then prepared for production. The components are labelled on the silkscreen. The design rule checks are run in order to verify that the design is correct. Finally, the Gerber files to be used in production are generated and checked.

3.6.1 Silkscreen

The silkscreen contains text and graphics which are printed on the board. This was used to label all the components with their reference designator.

3.6.2 Design Rule Checks

Before generating the Gerber files for production, the design rule checks were run to ensure that the board was completed and adhered to all the constraints. The clearance check checked that the distances between objects were large enough. The connectivity check ensured that everything was connected as specified in the schematic. The fabrication check checked for silkscreen covering pads and that minimum pad and trace widths were adhered to.

3.6.3 Gerber Generation

Once the board was completed, the files used for manufacturing it were generated. These are in Gerber format, and are generated by PADS Layout. These files are images of what must appear on the layer they represent. The Gerbers for the electrical layers specify which parts must contain copper (except for the ground plane, which specifies were copper must not appear since it is a CAM layer[1]). The silkscreen layers specify what must be printed on the silkscreen, and the solder mask layers specify which parts must not be covered with solder mask.

[1]Since the majority of the ground plane is usually copper, it is easier to specify which parts must not contain copper. This is achieved by defining the layer as a CAM layer, which causes the Gerber output to be inverted.

In addition to the Gerber files, a drill file is required in NC Drill format. This specifies all the holes which must be drilled in the board. (It took a significant amount of time to configure PADS Layout to correctly generate the drill file using metric units.) The Gerber files were checked in a Gerber viewer to ensure that all the layers lined up and that the units were correct.

3.7 Firmware Design

The firmware for the iBOB is adapted from existing firmware. Since the board uses the same pin connections as the iADC, the data collection routine is compatible. The control pins are configured as GPIO and made accessible to the processors. The SPI protocol is implemented in C and used to program the ADC and frequency synthesiser. Finally, the initialisation routine is written. A portion of the firmware is contained in Appendix C.

3.7.1 Sampled Data

The firmware tool created by CASPER only supports the iADC boards, and couldn't be adapted very easily to support the new ADC board. However, since the data interface of the new ADC board is the same as the iADC in single channel mode, this portion of the firmware is compatible.

Existing firmware was therefore used as the base for the new firmware. This existing firmware was created with the CASPER tool, and simply receives data from the ADC and transmits it over 10GigE.

3.7.2 GPIO

The VHDL and C code generated by the CASPER tool was then edited manually to add the required control functions. The easiest approach was to implement the SPI protocol in C, and so all the control pins were set up as GPIO pins accessible from C.

In order to do this, the `system.ucf` file was first edited to add the physical pin definitions. The logical pin mappings were then defined in the

`system.mhs` file, as well as the instantiation of the GPIO core. The GPIO core presents the pins as memory mapped I/O, which are controlled by simply writing values to a specific memory address.

Most of the time spent writing the firmware was spent integrating the GPIO core into the existing firmware. Although the firmware would compile successfully, the GPIO core would not show up in the address map of the processors. This meant that it wasn't actually accessible to the processors and therefore wouldn't work. Even with much assistance, this problem was not very easily fixed, and was eventually solved three days before this thesis was due.

3.7.3 Serial Interface

The serial interface consists of a clock and data line connected to all components, and a separate chip select line for each component. (This is more or less the SPI protocol.) The protocol was implemented in software (i.e. it was "bitbanged"). The serial interface is used to program the ADC and frequency synthesiser, and to retrieve the temperature from the temperature monitor.

ADC

The first register contains general configuration settings. The settings applicable for this board are shown in Table 3.1. Functions were written to set the input offset, full-scale voltage and clock phase adjust, as well as to enable the test pattern.

Frequency Synthesiser

The LMX2531 was configured for a 40 MHz reference frequency and 1.5 GHz output frequency. The variables used to calculate the register values are shown in Table 3.2, and the register programming sequence and values are shown in Table 3.3.

Table 3.1: ADC configuration

Symbol	Description	Value
DRE	Differential Reset Enable	On
RTD	Resistor Trim Disable	Off
DCS	Duty Cycle Stabilizer	On
DCP	DDR Clock Phase	90° Phase
nDE	DDR Enable	On
OV	Output Voltage	680 mV
OE	Output Edge	N/A

Table 3.2: LMX2531 Configuration Values

Variable	Value
NUM	0
N	150
ICP	7
R	4
DEN	0
FoLD	3
ORDER	1
DITHER	3
FDM	1
DIV2	0
C3_C4_ADJ	5
R3_ADJ_FL	3
R3_ADJ	3
R4_ADJ_FL	3
R4_ADJ	3
EN_LPFLTR	1
VCO_ACI_SEL	8
XTLSEL	1
XTLDIV	2
XTLMAN	0

Table 3.3: LMX2531 Programming

Register	Value
R5	0x840005
R5	0x800005
R5	0x8007F5
R12	0x01048C
R9	0x000BA9
R7	0x000207
R6	0x18FFD6
R3	0x74C003
R2	0x400042
R1	0x2E0001

3.7.4 Initialisation

The initialisation sequence consists of the following steps.

1. Set `PD` high in order to enable the ADC083000
2. Program configuration register of ADC083000
3. Program other settings of ADC083000
4. Initiate calibration of ADC083000
5. Program LMX2531 according to Table 3.3
6. Check `CALRUN` to indicate that calibration is complete
7. Check `LD` to indicate that LMX2531 has locked onto output frequency

3.8 Conclusions

The high level design of the system was simple since large parts of the system were specified in the requirements. The design therefore mostly involved identifying the supporting functions. The overall architecture is conceptually simple, but is sound and capable of meeting the requirements.

The circuit design was fairly straightforward and was successfully accomplished. Some decisions (such as the balun configuration) had to be made, but for the most part simply consisted of consulting the datasheets and following the design procedure given. Learning how to use DxDesigner did take a little time, but thereafter was easy to use.

The layout and routing process took much longer than expected, mostly due to the time required to learn how to use the software and due to a number of issues which were discovered and which required large parts of the board to be re-routed. This was probably the longest and most complex part of the project, and required many new skills and knowledge. However, in the end it was completed successfully and resulted in a good design. There are some improvements and optimisations that can be made, and the board would definitely benefit from a revision.

The simulations were very useful while designing the board and are an essential part of designing a good system. Learning how to use the software did not require much time. Simulations of the final design show that the impedances are well matched and that the signal integrity of the ADC outputs is very good.

Preparing the board for production was straightforward but required real attention to detail. All the errors which were found were fixed, and the Gerber files were successfully generated and checked.

The design of the firmware was very straightforward since the main functionality was already implemented. Integrating the GPIO core was much more difficult than expected, and prevented the firmware from being completed in time. Writing the C code to implement the serial protocol and control lines was done very quickly, but couldn't be tested.

Overall, the design of the system was successful but did require much more time than expected. More experience with the PCB design process and the tools involved would have decreased this time significantly. The resulting design appears to be good and the system should operate as intended with good performance.

Chapter 4

Integration and Testing

This chapter describes the manufacture and population of the board. The board and firmware are first tested individually, and then the system is tested as a whole. Data is collected and analysed to determine the performance of the board. A list of changes to correct and improve the design is then provided.

4.1 Introduction

The chapter begins with the manufacture of the board, and the problem that was encountered, which was the unavailability of RO4003 laminate. Some simulations are then done to determine the repercussions this will have on the board. Sourcing of the parts for the board is briefly described, followed by the population of the board. This was done by hand and required a non-standard technique for mounting the leadless components.

The board is then tested, beginning with a visual inspection. The form factor of the board is checked, and some DC tests are done to check connectivity. The board is then powered up and the power supplies are tested and inspected for ripple.

The rest of the system is then tested, beginning with the firmware. Unfortunately the firmware did not work as intended, and there was not sufficient time to fix it. The remainder of the chapter describes the tests which would

have been done after the firmware was working. The sampling clock would be tested next, followed by the data collection routine. Lastly, the entire system would be tested.

The system's performance would then be evaluated. Three sets of data would be collected for analysis, using three different sinusoids as the input signal. This data would be processed and various measures calculated. These measures would be compared to the expected values, and plots of the data would be generated.

Lastly, a list of changes which should be implemented is provided. These include errors which were found during testing, as well as some possible improvements.

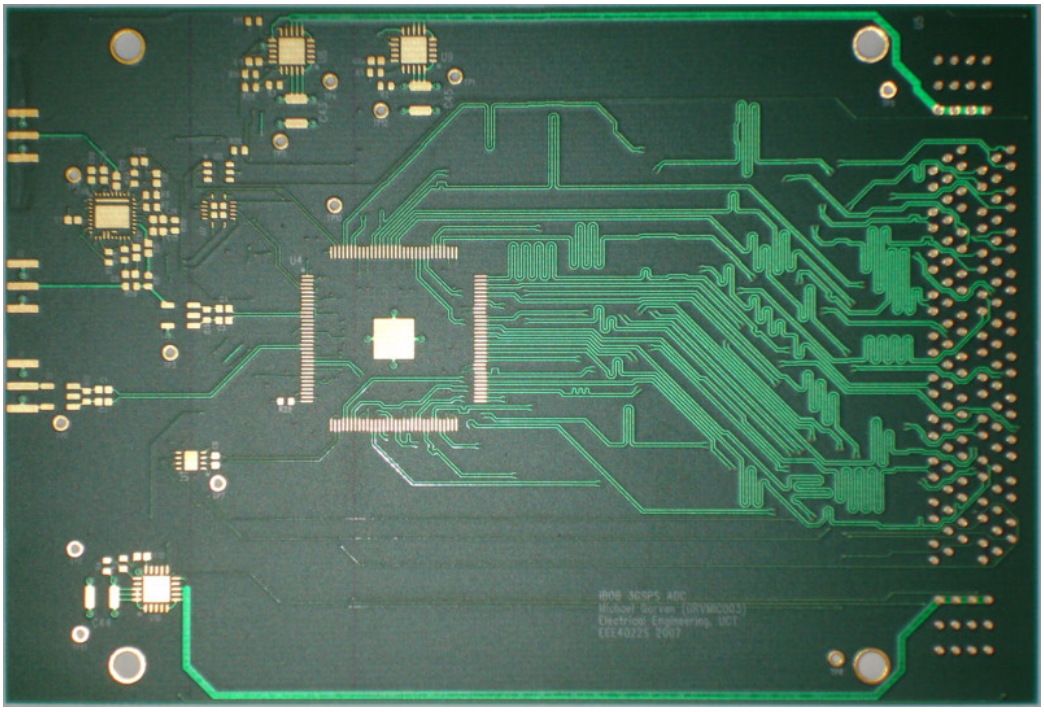
4.2 Manufacture

The board was sent to Trax Interconnect to be manufactured. Unfortunately, Trax no longer had RO4003 laminate in stock, and neither Trax nor KAT could procure the RO4003 fast enough for the project timeframe. The board therefore had to be manufactured using FR-4 laminate. This means that the impedances will no longer be matched, and will also vary significantly with frequency. This will have a negative impact on the performance of the board. The board manufacture also took longer than expected, which left very little time for the integration and testing phase. The PCB is shown in Figure 4.1.

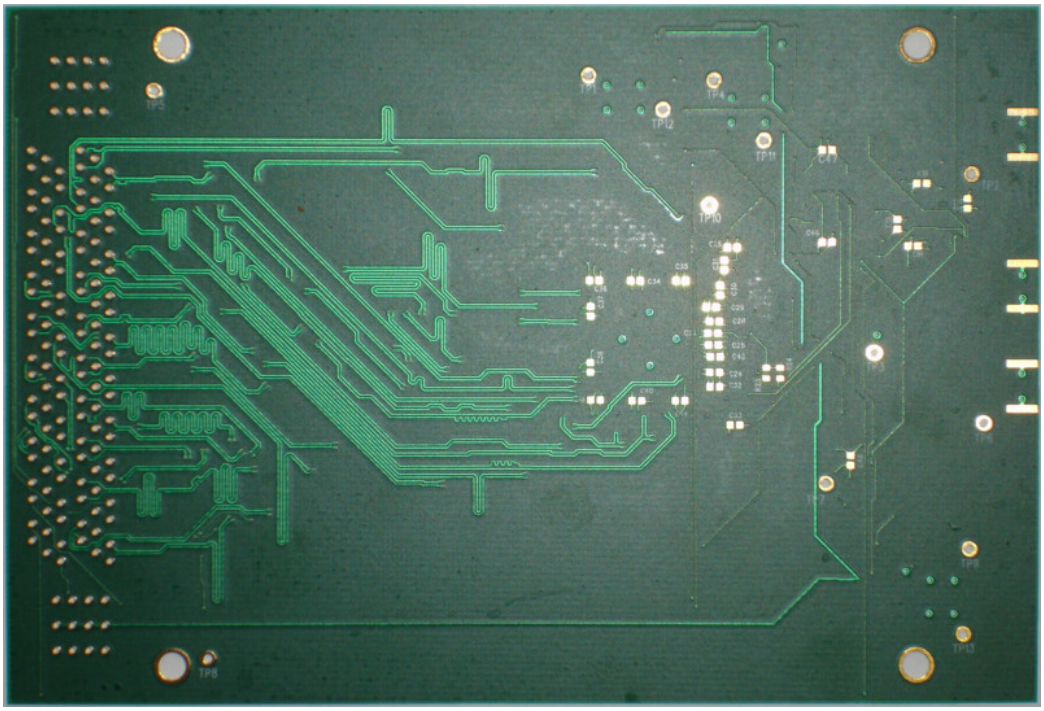
4.2.1 Simulation

The simulations in Section 3.5 were redone in order to determine what effect the FR-4 would have on the board. The dielectric constant of the substrate was changed to that of FR-4. The results show that the 50 Ω traces would now have an impedance of 45.1 Ω, and the 100 Ω differential traces would now have a differential impedance of 90.7 Ω. This represents an error of approximately 10%, which is significant enough to create some reflections on the lines.

The signal integrity simulations were also redone, the results of which are



(a) Top



(b) Bottom

Figure 4.1: Manufactured PCB

shown in Figure 4.2 and Figure 4.3. The signals still seem fairly clean, with good transitions. The peak-to-peak voltage of the crosstalk in Figure 4.3a is 28.6 mV, which is actually slightly less than the value measured in Figure 3.4a.

4.3 Parts

The passive components (resistors and capacitors) needed had been ordered from Digi-Key. Samples of the LMX2531LQ1570E and MAX6627 were requested and received from National Semiconductor and Maxim respectively. The Z-DOK connectors were couriered from Berkeley, and the TPS74401, SN74AUC34, SN74LVC2G07 and TC1-1-13M were stocked by KAT.

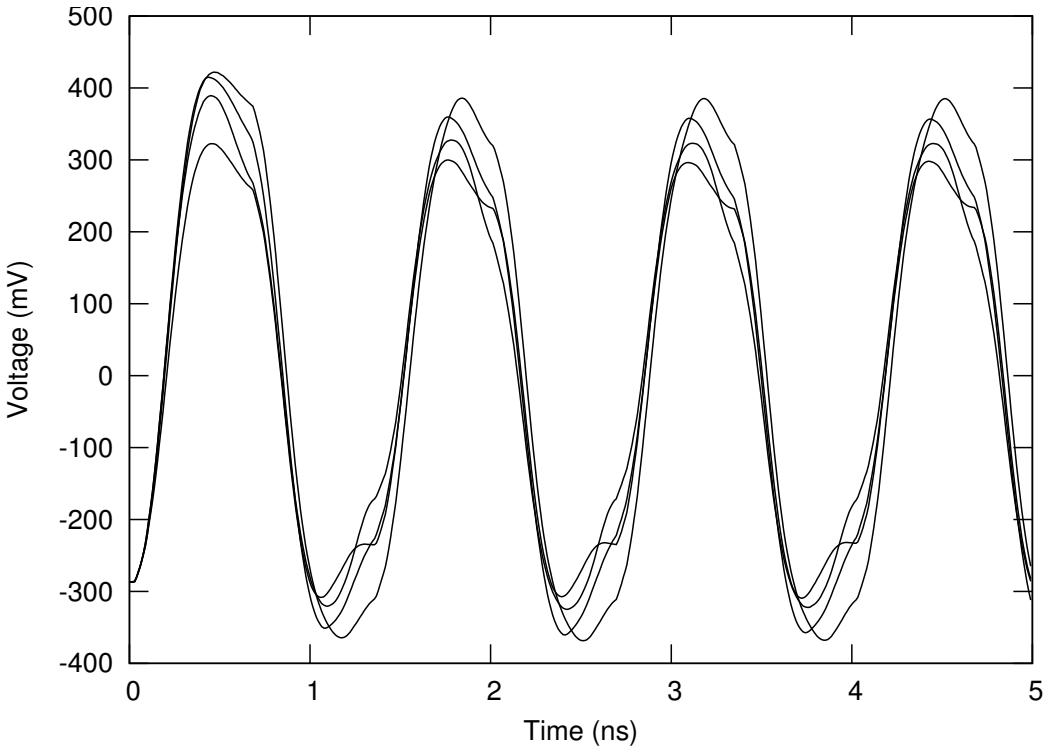
Due to the cost, we were unable to get samples of the ADC083000. Luckily KAT had samples of the ADC082500, which is a pin compatible 2.5 GSPS ADC, and so this was used instead. One sample of the SMA connector was received from RF Design, but we neglected to order the remaining connectors.

4.4 Population

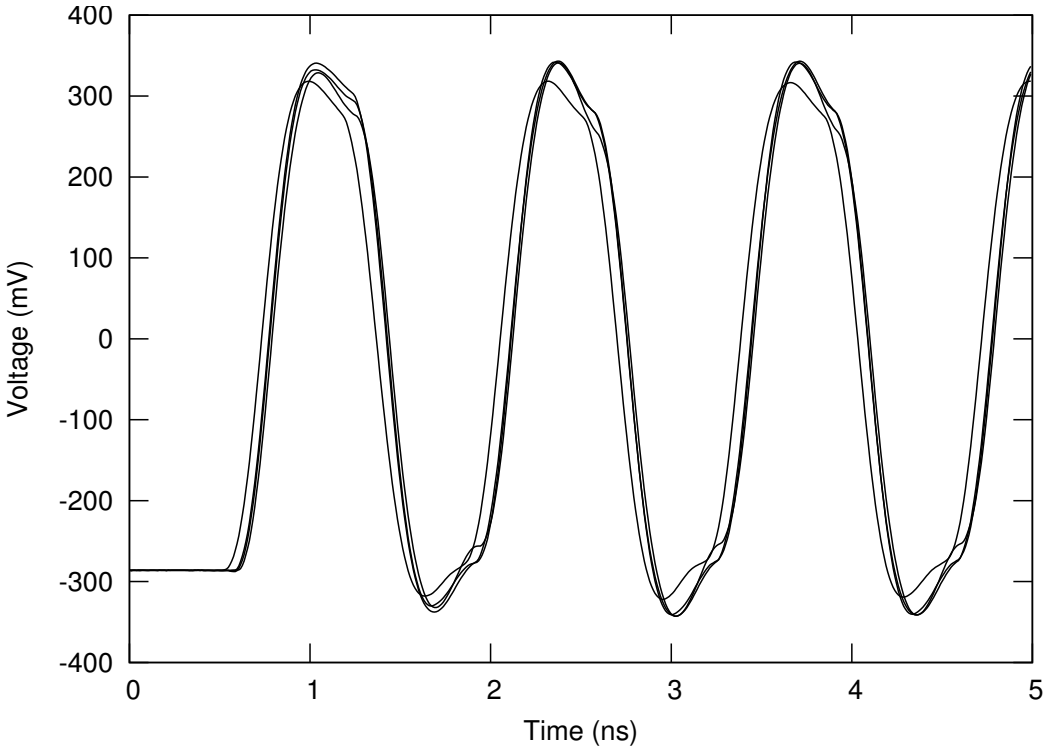
The PCB was then populated by hand with the assistance of Alec Rust. The passive components were placed first, followed by the regulator chips. The board was then powered up and the output of the regulators were checked. (This was to ensure that the other chips weren't damaged by incorrect output voltages.)

The remaining chips were then soldered onto the board. Some of these were fairly difficult since they did not have leads — only pads on the bottom. These were soldered by tinning the component and the board, placing it, and then heating it up with hot air.

Once all the chips were soldered, the board was powered up again and the power supplies were checked. Finally, the Z-DOK connector was soldered on. The completed (except for the SMA connectors) board is shown in Figure 4.4.

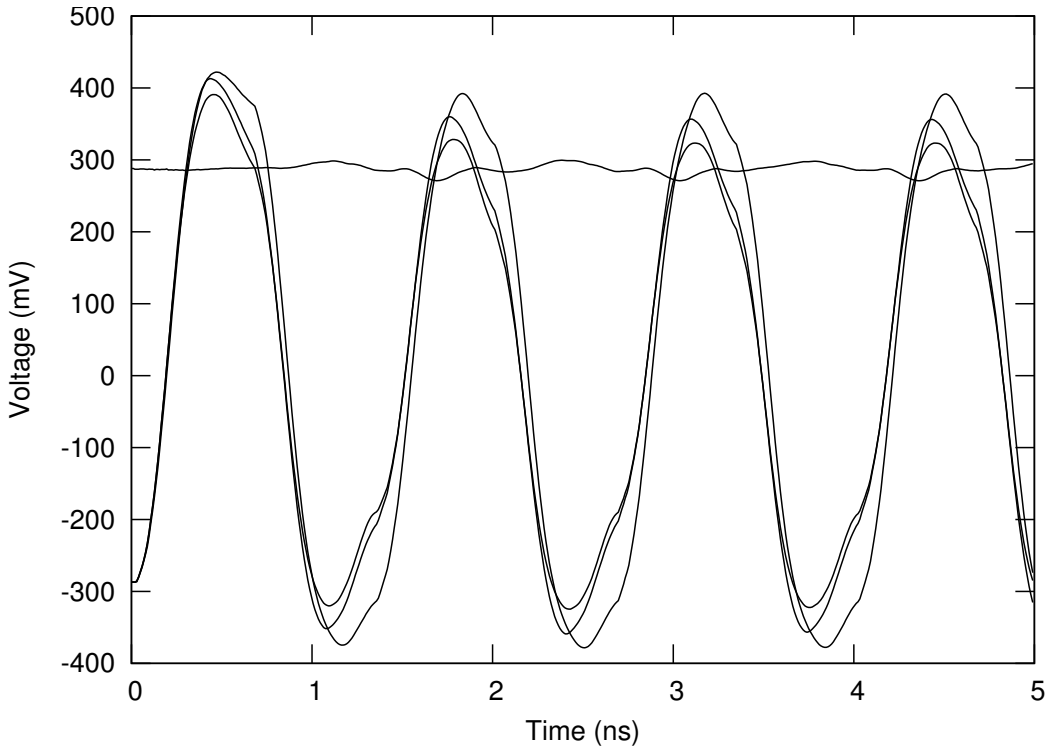


(a) At ADC pins

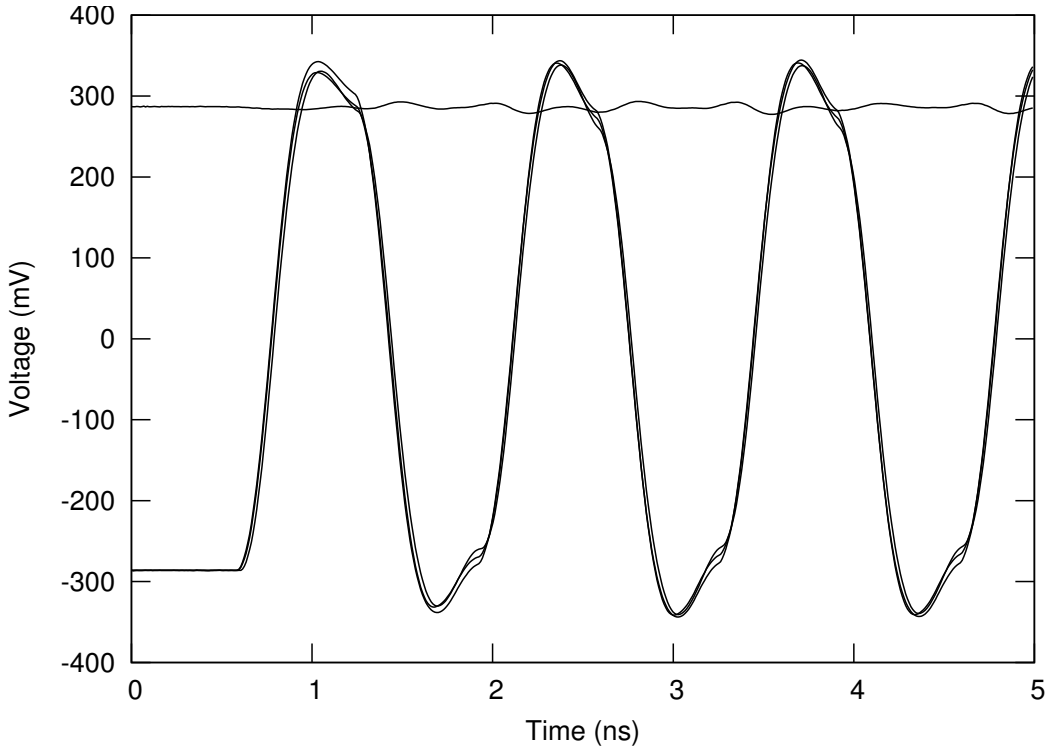


(b) At Z-DOK pins

Figure 4.2: Simulation of ADC outputs on FR-4 material

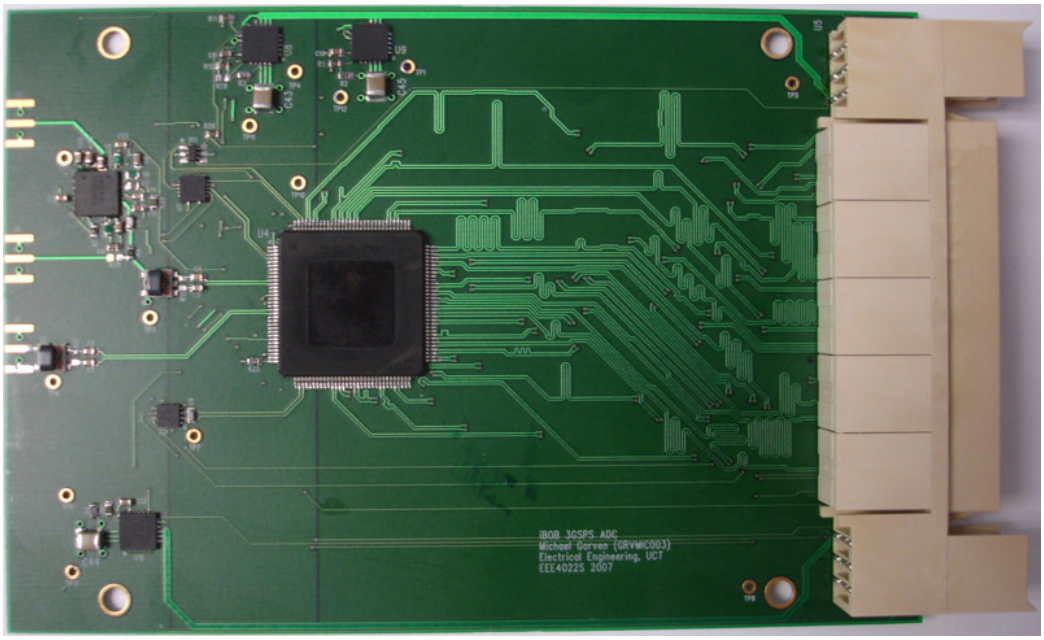


(a) At ADC pins

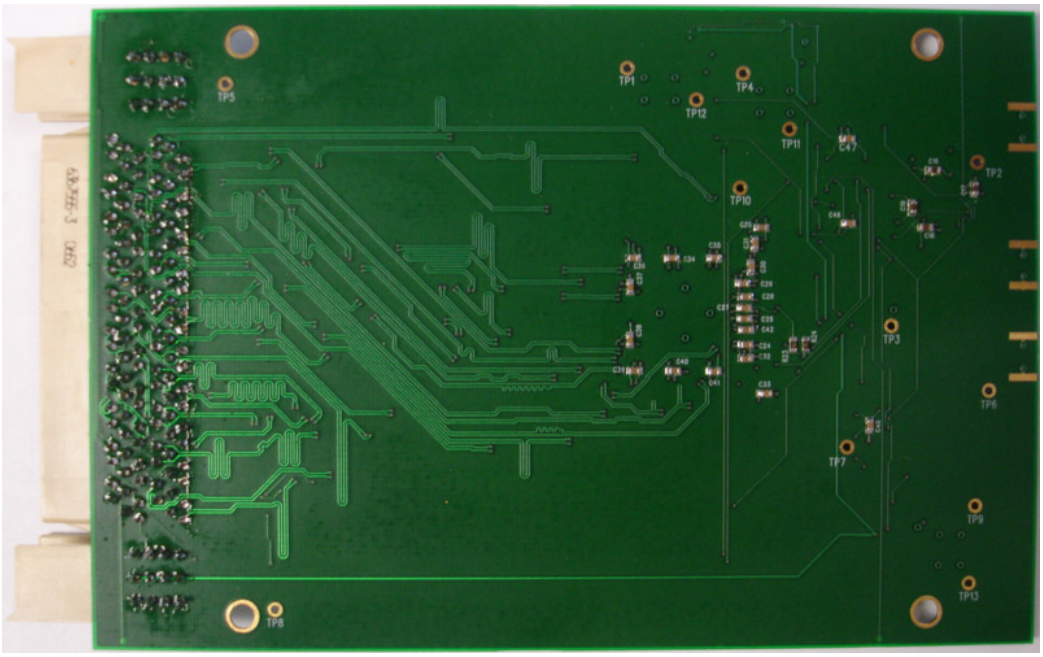


(b) At Z-DOK pins

Figure 4.3: Simulation of ADC outputs on FR-4 material showing crosstalk



(a) Top



(b) Bottom

Figure 4.4: Populated ADC board

4.5 Board Testing

The board is first tested to ensure that it is working correctly. A visual inspection is done and the form factor of the board is tested. Connectivity between the ADC board and the iBOB are tested with the DC tests, and the power supplies are then checked.

4.5.1 Visual Inspection

The board was visually inspected, to check that no components were missing, that everything was soldered and that the solder joints looked good.

4.5.2 Form Factor Test

This test verifies the requirement that the board be compatible with the iBOB. The board was connected to the iBOB as shown in Figure 4.5. The connector fitted correctly, and the mounting holes were correctly located over the mounting screws. This test was passed.

4.5.3 DC Tests

The DC tests check connectivity between various points. The PCBs were tested by the manufacturer with a flying probe machine. This ensures that the tracks on the board are properly connected. A multimeter was used to check that the power supplies weren't shorted to ground.

Connectivity between the ADC board and the iBOB was then tested. The board was connected to the iBOB. Various signals were then checked for connectivity between the ADC board and the iBOB using a multimeter. The test results are shown in Table 4.1.

4.5.4 Power Supplies

This test verifies the requirement that the board must have on-board voltage regulators, and that these are operating correctly. The board was connected

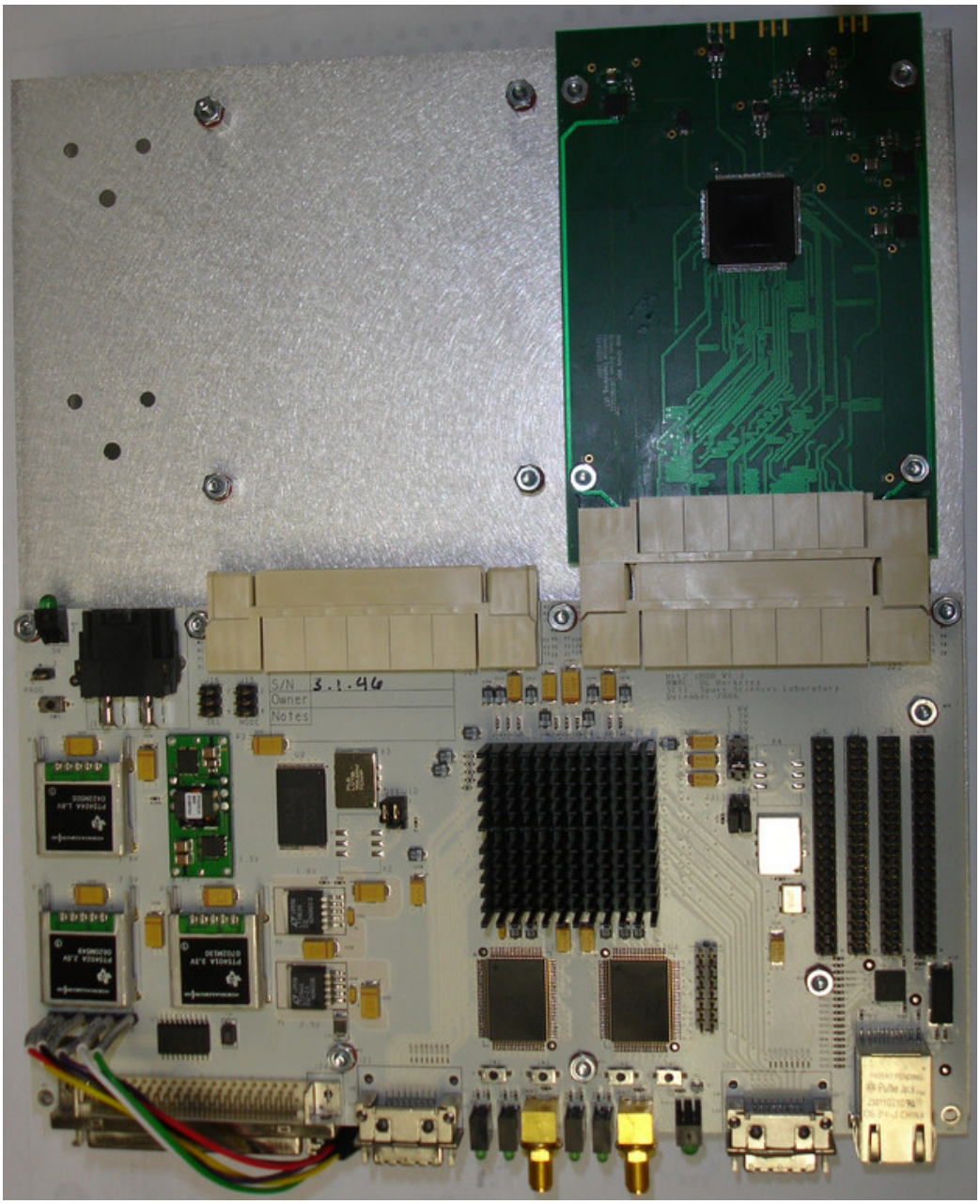


Figure 4.5: ADC board connected to the iBOB

Table 4.1: Results of Z-DOK DC Tests

Signal	Connected	Pass/Fail
Ground	Yes	Pass
5 V	Yes	Pass
3.3 V	Yes	Pass
2.5 V	Yes	Pass

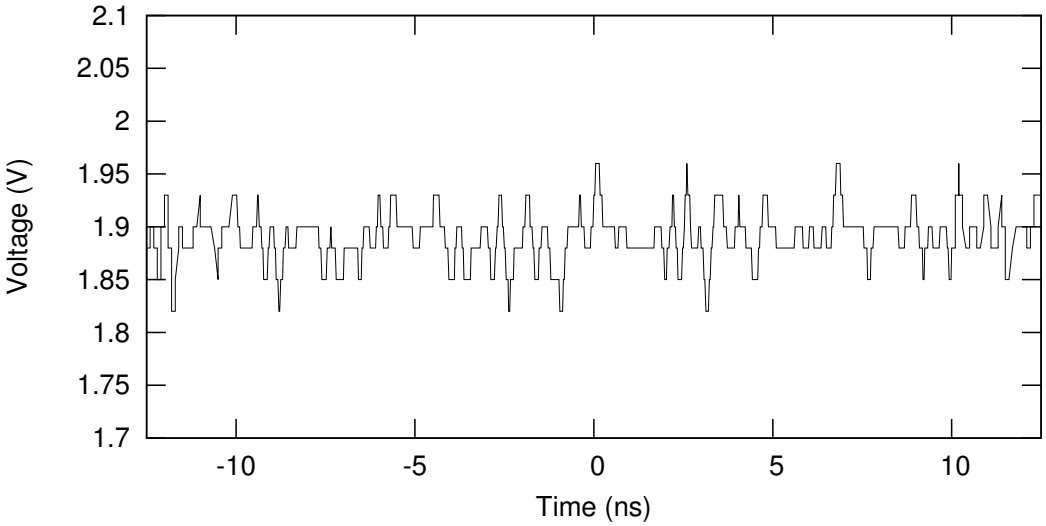


Figure 4.6: Plot of 1.9 V power supply

to the iBOB and powered up. The power supply at the pins of each component was then measured and compared to the expected value. Table 4.2 shows the results of these tests, all of which passed.

The power supplies were also inspected using an oscilloscope, to determine the amount of ripple. The 1.9 V supply is shown in Figure 4.6, and has a peak-to-peak ripple of 140 mV. The ripple on the other supplies was also acceptable.

4.6 System Testing

Once the board has been tested, the other parts of the system are tested. The firmware is tested first, and then the sampling clock generated by the frequency synthesiser is checked. The data collection process is then tested,

Table 4.2: Results of Power Supply Tests

Ref. Des.	Device	Expected	Measured	Pass/Fail
U8	TPS74401	2.5 V	2.513 V	Pass
U9	TPS74401	2.5 V	2.513 V	Pass
U10	TPS74401	3.3 V	3.320 V	Pass
U1	MAX6627	3.0 V	3.006 V	Pass
U4	ADC083000 V_{A}	1.9 V	1.908 V	Pass
U4	ADC083000 V_{DR}	1.9 V	1.899 V	Pass
U6	LMX2531	3.0 V	3.006 V	Pass
U7	SN74AUC34	1.9 V	1.899 V	Pass
U11	SN74LVC2G07	1.9 V	1.899 V	Pass

and lastly the system as a whole is tested.

4.6.1 Firmware

In order to test the firmware, it was modified to continually run the initialisation routine. It was compiled and then programmed into the iBOB. An oscilloscope was then used to check for activity on the serial interface lines.

The firmware was compiled and successfully programmed into the iBOB. However, there was no activity on the serial interface, which means that the firmware does not work correctly, and this test failed. There was no time left in which to debug the problem.

Due to the firmware not working, the tests which follow could not be conducted. The remainder of this chapter therefore presents the tests that would have been done should the firmware have worked.

4.6.2 Frequency Synthesiser

This test verifies the requirement that the board must generate the sampling clock from a low speed reference clock, as well as the requirement that the firmware must program the frequency synthesiser. The board is connected to the iBOB and powered up. The frequency synthesiser is then programmed by the iBOB firmware. The output frequency of the LMX2531 is measured with an oscilloscope.

4.6.3 ADC

This test verifies that the ADC is operating. The board is powered up and the ADC and frequency synthesiser are programmed by the firmware. The outputs of the ADC are examined with an oscilloscope to check for the presence of data.

4.6.4 External Sampling Clock

This test verifies the requirement that the board accept an external sampling clock. The board is configured to use the external sampling clock, connected to the iBOB and powered up. A signal generator is used to provide a 1.5 GHz square wave as the external sampling clock. The outputs of the ADC are examined with an oscilloscope to check for the presence of data.

4.6.5 Data Collection

This test verifies the requirement that the firmware must collect data from the ADC and transmit it to a computer. In order to test the data collection and transmission, the test pattern feature of the ADC083000 is used. This feature causes the ADC to ignore the input signal, and output a preset pattern on the data outputs [14, p.28]. The firmware is configured to enable this test pattern, and data is then collected and transmitted to a computer. The received data is inspected by hand in order to identify the test pattern.

4.6.6 System Test

This test verifies the fundamental requirement of the system. The ADC board is connected to the iBOB and powered up. A signal generator is used to provide a 40 MHz sinusoidal signal as the reference clock, with a power of 10 dBm. A second signal generator is used to provide a 100 MHz sinusoidal signal as the input signal, with a power of 5 dBm.

The firmware on the iBOB is executed, which configures the ADC board and then collects the sampled data. The data is transmitted over 10GigE to a computer. The data is then processed with an FFT and plotted. The plot

is inspected and the frequency of the strongest signal is determined. This frequency should correspond to the 100 MHz input signal.

4.7 Performance Testing

The performance testing aims to characterise the performance of the ADC board. Sampled data is collected for different input signals and then analysed. Various values are calculated and compared to the expected values and to other ADCs.

4.7.1 Data Collection

The ADC board is connected to the iBOB, and a signal generator is used to provide the 40 MHz sinusoidal reference clock to the board. A second signal generator is used to provide the input signal to be sampled. The signal generator is configured to output a single tone sinusoid with a power of 5 dBm. A set of 1 000 000 samples are collected for each input frequency. Three different input frequencies were used: 375 MHz, 750 MHz and 1.5 GHz.

4.7.2 Data Analysis

The collected data is then analysed. An FFT of the data is computed and plotted. The plot is checked to ensure that it corresponds to the correct input frequency. The calculations defined in Section 2.11 are calculated.

The measured values are then compared to the values quoted in the datasheet [14, p.1], as shown in Table 4.3. The values in the datasheet are quoted for a 748 MHz input signal, and so the values measured for the 750 MHz input are used. For comparison, the values quoted for the BenADC-3G (which also uses the ADC083000) are provided. [13, p.1]

Table 4.3: ADC Performance Comparison

Variable	Measured	Datasheet	BenADC-3G
ENOB		7.0 bit	7.2 bit
SNR		44.5 dB	45 dB
BER		10^{-18}	
SINAD			
SFDR			58 dBc
THD			−53 dBc

4.8 Design Changes

During the testing phase and while reviewing the board layout after it had begun manufacture, a few errors were found on the board, as well as some possible improvements. The following is a list of changes which would correct and improve the board.

1. The 100 Ω resistor `R28` on the output of the 1.9 V power supply is connected in the wrong place. It should be connected between the 1.9 V output and ground.

2. The balun configuration (Figure 2.7) was implemented incorrectly. The capacitors should be placed before the 100 Ω resistor, and these should be placed closer to the ADC inputs.

3. Guard traces (traces connected to ground) should be placed around the clock and input lines to provide better noise immunity.

4. The traces connecting the temperature diode on the ADC083000 to the MAX6627 should be thicker and be surrounded with guard traces, as recommended in the datasheet. [11, p.7]

5. A ground trace should be placed around the edge of the board, as done on the iADC board, to provide better noise shielding from external interference. [3]

6. The four mounting holes should be connected to ground as done on the iADC board, in order to allow the board to be grounded to the

mounting chassis. [3]

7. The bottom pads of the LMX2531 should be connected to ground.

8. The serial data line should not be connected to all three components. There should be two separate data lines for the write-only and read-only components.

4.9 Conclusions

The manufacturing of the board was not ideal since it had to be done on FR-4 instead of RO4003. This resulted in mismatched impedances which also vary significantly with frequency due to the nature of FR-4. The manufacture was also delayed significantly, which meant that there was not enough time in which to complete the integration and testing phase.

Almost all of the parts for the board were readily available, and the board was populated by hand. Some of the components were difficult to place by hand, and possibly are not soldered properly. When selecting components for a board that will be populated by hand, leadless components should be avoided due to this difficulty. Alternatively the board should be populated professionally.

The completed board was inspected and the DC tests all passed. The power supplies were tested, and the measurements all agreed with the expected values. The ripple on the power supplies is acceptable. The firmware was then tested, but did not work as intended. The firmware could not be fixed due to a lack of time, and the remainder of the tests could therefore not be done.

Should the system have worked, the performance testing should have been fairly straightforward. Unfortunately no results were gathered since the system was not completed. The performance of the system therefore could not be compared and evaluated. Some errors were found with the board, as well as a few improvements which could be incorporated in future revisions.

Even though it was not fully tested, the board certainly has the potential to work correctly and meet all the requirements. If the board manufacture hadn't been delayed, I fully expect this phase to have been completed successfully.

Chapter 5

Conclusions and Future Work

This chapter presents the outcomes and results of the project, as well as what was not achieved. Conclusions are then drawn from the experience, and possible future work is briefly outlined.

5.1 Outcomes

The technical research phase of the project was completed successfully, and all the information necessary for the project was gathered. This included details of the components to be used, analysis of ADCs, phase noise, single to differential conversion, PCB design, high frequency and high speed digital issues, FPGA firmware development, and the iBOB board. This phase helped to develop my research and learning skills.

The design phase was also completed successfully. The high level design was easily done, and the circuit design was straightforward. The layout and routing process took much longer than expected, but eventually resulted in a good layout with well matched impedances and lengths. Most of the firmware was implemented, but could not be completed or tested due to a lack of time.

The project encountered the biggest barrier in the manufacturing phase. This was due to the unavailability of the ideal PCB material, resulting in a sub-optimal PCB. Manufacturing also took longer than anticipated, which prevented the project from being completed. The board was populated with

the components, and the basic board tests were passed.

The integration and testing phase of the project was not completed due to a lack of time. A plan to test the system and verify that it meets the requirements was developed, as well as a method for evaluating the performance of the system. These plans could not however be implemented, and so the hardware remains untested.

5.2 Conclusions

The system which was designed is conceptually very simple, and at first glance seems almost trivial. The difficulties lay in the high speed digital and high frequency analogue aspects of the system. These needed careful attention as they were crucial to the performance of the system. These aspects can become very complicated and are specialised fields in and of themselves.

The use of separate ADC boards on the iBOB (as opposed to having the ADC directly on the iBOB) was a very good design decision. It enables the ADC boards to be upgraded separately, and reduces the complexity that has to be dealt with.

A large number of skills were required in order to execute this project. Learning how to use the EDA tools took a significant amount of time, and was a hurdle to many of the tasks which made up the project. Developing firmware for FPGAs is also a very complicated task which requires a lot of learning. Since only minor modifications were done to existing firmware this wasn't a major issue for this project, but more knowledge of this would certainly have helped.

The simulations were an extremely important part of the design process. They allowed the impedances to be matched correctly without resorting to complicated mathematics. The autorouter was also a very useful tool because it managed to match the lengths of the output lines within 3 mm. Doing this manually would be extremely difficult.

Despite the few errors found with the board, it appears to be a successful implementation of the requirements. Completion of the testing would

verify this, but I am confident that the board will operate as intended. Its performance will not however achieve the expected values because it was manufactured on FR-4 and because of these errors. Correcting these errors and manufacturing it properly would yield a better board which should reach the expected performance.

Designing, implementing, producing and testing hardware for an undergraduate thesis is a very daunting task. This is due to the time, skills and knowledge which are required. More exposure to the PCB design process during the degree would make a significant difference to this.

I have gained a great deal of knowledge and new skills through this project, and overall it was a good experience. Knowing how to use the EDA tools and knowledge of the tool flow will certainly help me in my future career as an Electrical Engineer. Going through the process of hardware design has shown me what is really involved in the process, and will assist me when working on a design team.

5.3 Future Work

Due to the time available in which to complete the project, the scope was fairly narrow. There are a number of ways in which this project could be extended and improved upon.

5.3.1 Completion of Integration and Testing

The integration and testing phase was not completed due to a lack of time. The system is not working as intended, and results were not gathered. Completing this phase should yield an operational system, and the performance analysis will allow it to be evaluated and compared.

5.3.2 Board Revision

There were some design errors found with the board, and some possible improvements were discovered after the board had begun manufacture (these

are listed in Section 4.8). Performing the remaining tests may reveal additional errors with the board. Correcting these errors and manufacturing a new board would yield a better system. The new board should also be manufactured on RO4003.

New features which could be added to the board include an amplifier/attenuator for the input signal and *automatic gain control* (AGC). Different methods for performing the single to differential conversion could also be investigated.

5.3.3 Firmware

Due to time constraints, not much time could be spent on the iBOB firmware. The firmware had to be adapted from existing firmware for the current iADC boards. This firmware doesn't support the full 3 GSPS of the ADC083000 because it would require the FPGA to be clocked at 375 MHz, which is not possible.

It is theoretically possible to support the full sampling rate on the iBOB, but this requires significant changes to the firmware. The basic idea is to add a SERDES modules at each data input line, which would deserialise the data by converting each line into two lines running at half the rate. The 32 bit data bus would therefore be converted to a 64 bit bus running at half the speed. The FPGA would need to be clocked at $\frac{375\,\text{MHz}}{2} = 187.5\,\text{MHz}$, which is attainable.

5.3.4 Interleaving 2 ADC Boards

Since the iBOB can use two ADC boards simultaneously, it should be possible to interleave two such boards to achieve an overall sampling rate of 6 GSPS. This has recently been achieved by Raffanti [17] at CASPER, but with the ROACH board (which is a successor to the iBOB).

An external 1.5 GHz sampling clock would be applied to one ADC board, and a 90° delayed version applied to the other board. The same input signal would be applied to both ADC boards. The samples from the two ADCs would then be interleaved sample by sample.

5.3.5 Adapting for ROACH

ROACH is the successor to the iBOB, and performs the same function in the system (although it can perform more signal processing than the iBOB can). Adapting the ADC board to work with ROACH would probably only involve writing firmware for the ROACH board since the same connector is used for the ADC board.

References

[1] *CASPER Website*, Sep 2007. URL http://casper.berkeley.edu. 2

[2] Brad Brannon and Allen Barlow. Aperture Uncertainty and ADC System Performance, 2006. URL http://www.analog.com/UploadedFiles/Application_Notes/494064912342505447175991257024546937062255921511183854180687755AN501_a.pdf. 13, 14

[3] CASPER. *iADC v1.1 Layout*, Jun 2007. URL http://casper.berkeley.edu/hardware/adc_rev1_layout.zip. 27, 28, 70, 71

[4] CASPER. *iADC v1.1 Schematic*, Feb 2006. URL http://casper.berkeley.edu/hardware/adc_v1_1_sch.pdf. 23, 24

[5] CASPER. *iBOB Block Diagram*, Oct 2006. URL http://casper.berkeley.edu/hardware/ibob_block_diagram.pdf. 32

[6] e2v. *AT84AD001B Datasheet*, 2007. URL http://www.e2v.com/download.cfm?type=document&document=1060. 2

[7] e2v. *EV8AQ160 Datasheet*, 2007. URL http://www.e2v.com/download.cfm?type=document&document=1159. 12

[8] e2v. *Single-to-differential Conversion in High-frequency Applications*, 2004. URL http://www.e2v.com/download.cfm?type=document&document=732. 17, 18, 19, 20

[9] Walt Kester. Understand SINAD, ENOB, SNR, THD, THD + N, and SFDR so You Don't Get Lost in the Noise Floor,

Mar 2005. URL http://www.analog.com/en/content/0,2886,760%255F788%255F91250%255F0,00.html. 35, 36

[10] Walt Kester. Converting Oscillator Phase Noise to Time Jitter, 2005. URL http://www.analog.com/en/content/0,2886,760%255F788%255F91502,00.html. 13

[11] Maxim. *MAX6627 Datasheet*, 2006. URL http://datasheets.maxim-ic.com/en/ds/MAX6627-MAX6628.pdf. xii, 22, 34, 70

[12] Mini-Circuits. *TC1-1-13M Datasheet*, 2007. URL http://www.minicircuits.com/pdfs/TC1-1-13M.pdf. xiv, 21

[13] Nallatech. *BenADC-3G Product Brief*, May 2007. URL http://www.nallatech.com/mediaLibrary/images/english/6463.pdf. 69

[14] National Semiconductor. *ADC083000 Datasheet*, 2007. URL http://www.national.com/ds.cgi/DC/ADC083000.pdf. x, 3, 12, 14, 21, 33, 34, 41, 42, 43, 68, 69

[15] National Semiconductor. *ADC083000 Evaluation Board*, Jun 2007. URL http://www.national.com/appinfo/adc/files/ADC08(B)3000RB_Schematic.pdf. 43

[16] National Semiconductor. *LMX2531 Datasheet*, 2007. URL http://www.national.com/ds.cgi/LM/LMX2531LQ1570E.pdf. xii, 3, 15, 16, 34

[17] Rick Raffanti. 6 Gsps ADC Interleave Test, Sep 2007. URL http://casper.berkeley.edu/memos/mini_roach_sine_sweep.pdf. 76

[18] Raltron. Convert SSB phase noise to jitter, 2007. URL http://www.raltron.com/cust/tools/osc.asp. 15

[19] RF Design. *SMA Edge Connector*. Retrieved on Oct 2, 2007 from http://www.rfdesign.co.za/pages/5645456/Products/RF-Connectors/SMA-connectors.asp. 24

[20] Rogers Corporation. *RO4003 Datasheet*. URL `http://www.rogerscorporation.com/mwu/pdf/ro4000ds_4.pdf`. xiii, 29, 30

[21] SKA South Africa. About the SKA, Oct 2007. URL `http://www.ska.ac.za/aboutska/index.shtml`. 1

[22] Texas Instruments. *SN74AUC34 Datasheet*, 2005. URL `http://www.ti.com/lit/gpn/sn74auc34`. xiv, 26

[23] Texas Instruments. *SN74LVC2G07 Datasheet*, 2007. URL `http://www.ti.com/lit/gpn/sn74auc34`. xiv, 27

[24] Texas Instruments. *TPS74401 Datasheet*, 2007. URL `http://www.ti.com/lit/gpn/tps74401`. xiv, 25, 42

[25] Trax Interconnect. *Short Form Specification for Standard Product*, Nov 2005. URL `http://www.trax.co.za/supdocs/ShortF-1115380106.pdf`. 27, 45

[26] Trax Interconnect. *Multilayer Material Lay Up Tables*, 2002. URL `http://www.trax.co.za/supdocs/Materi-1115131197.pdf`. 30, 45

[27] Tyco Electronics. *Z-DOK Datasheet*, Aug 2005. URL `http://ecommas.tycoelectronics.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Specification+Or+Standard%7F108-1985%7FA%7Fpdf%7FEnglish%7FENG_SS_108-1985_A.pdf`. xv, 23

[28] Wikipedia. *Balun*, Oct 2007. URL `http://en.wikipedia.org/wiki/Balun`. x, 21

[29] Wikipedia. Phase Noise, 2007. URL `http://en.wikipedia.org/wiki/Phase_noise`. 13

[30] Wikipedia. *SMA connector*, Oct 2007. URL `http://en.wikipedia.org/wiki/SMA_connector`. 23

[31] Xilinx. *Virtex-II Pro 2VP50 Datasheet*, Mar 2007. URL `http://direct.xilinx.com/bvdocs/publications/ds083.pdf`. xiv, 15

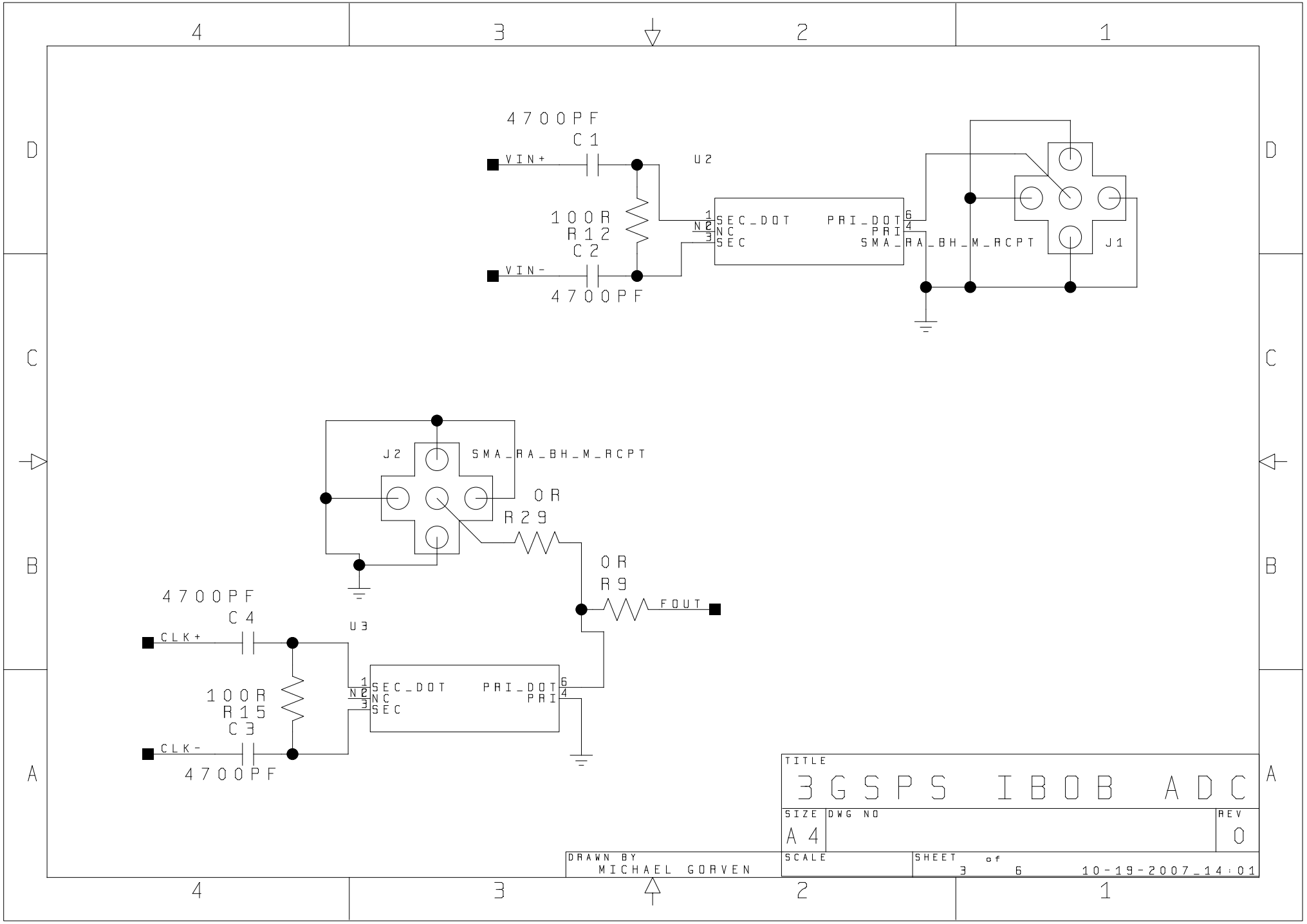
Appendix A

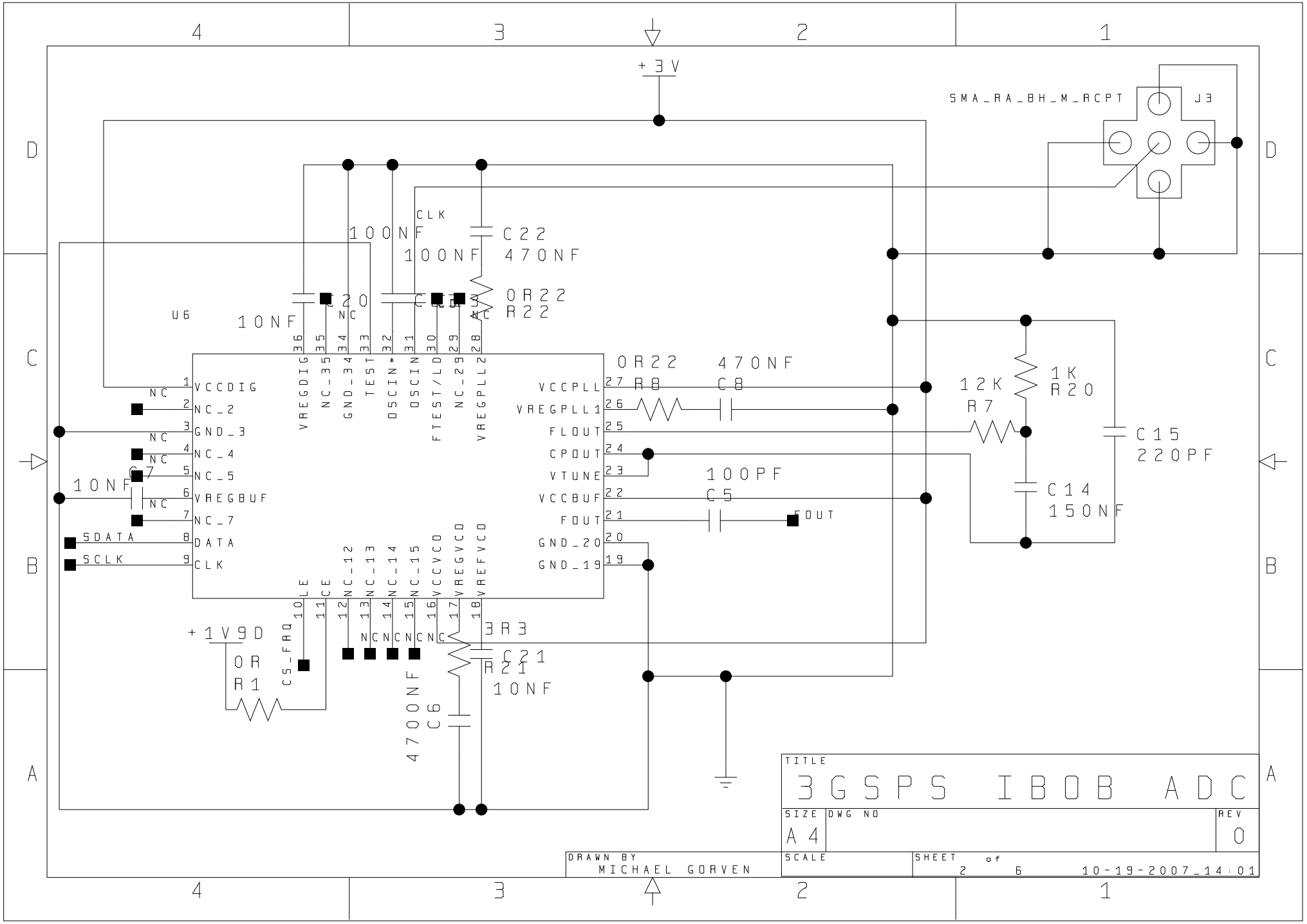
Schematic

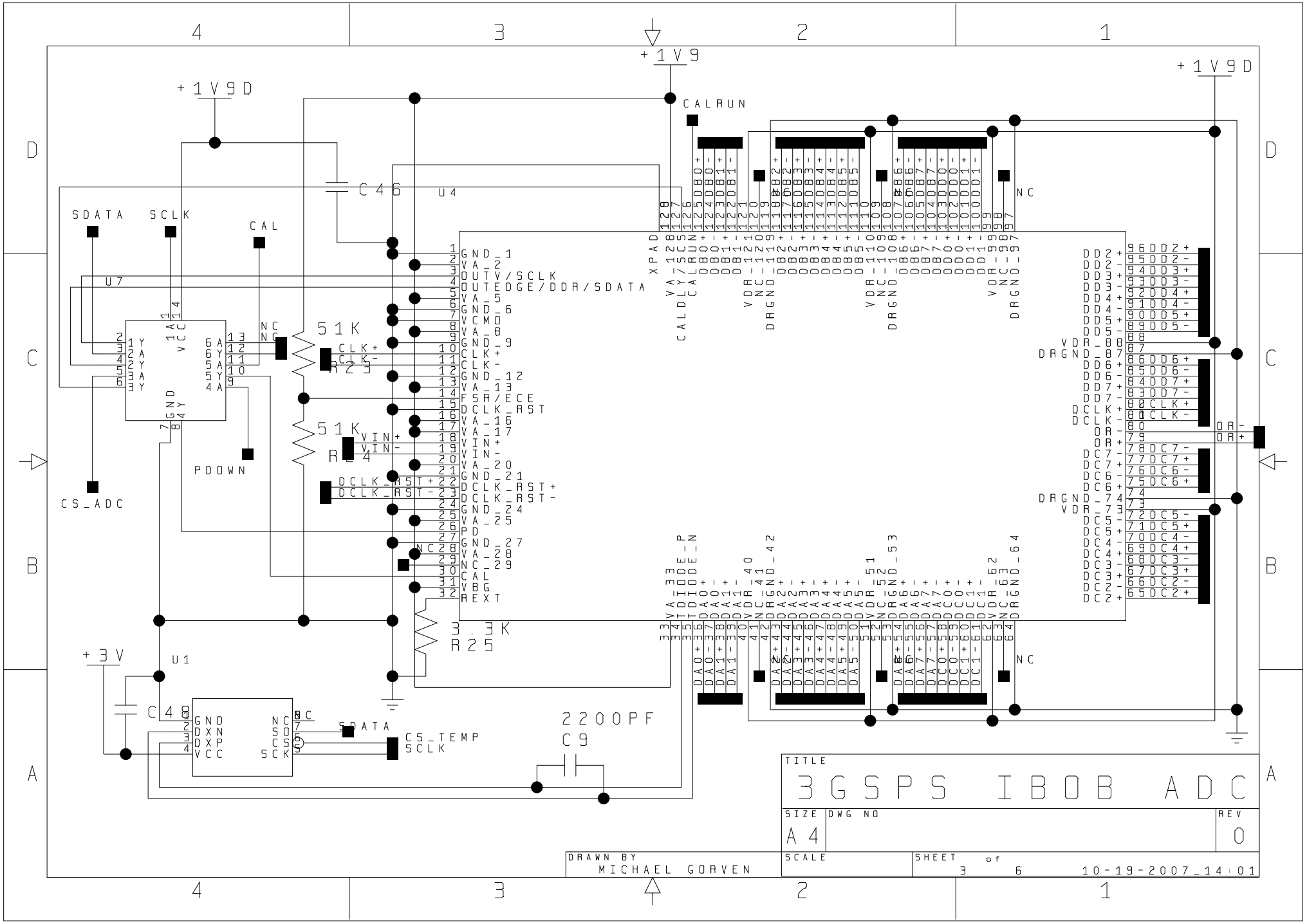
This appendix contains the outcome of the circuit design described in Section 3.3. Table A.1 lists all the components and parts used on the board, and the following six pages contain the circuit diagram. The schematic is available on the accompanying CD in DxDesigner format, and can be found in the `schematic` directory.

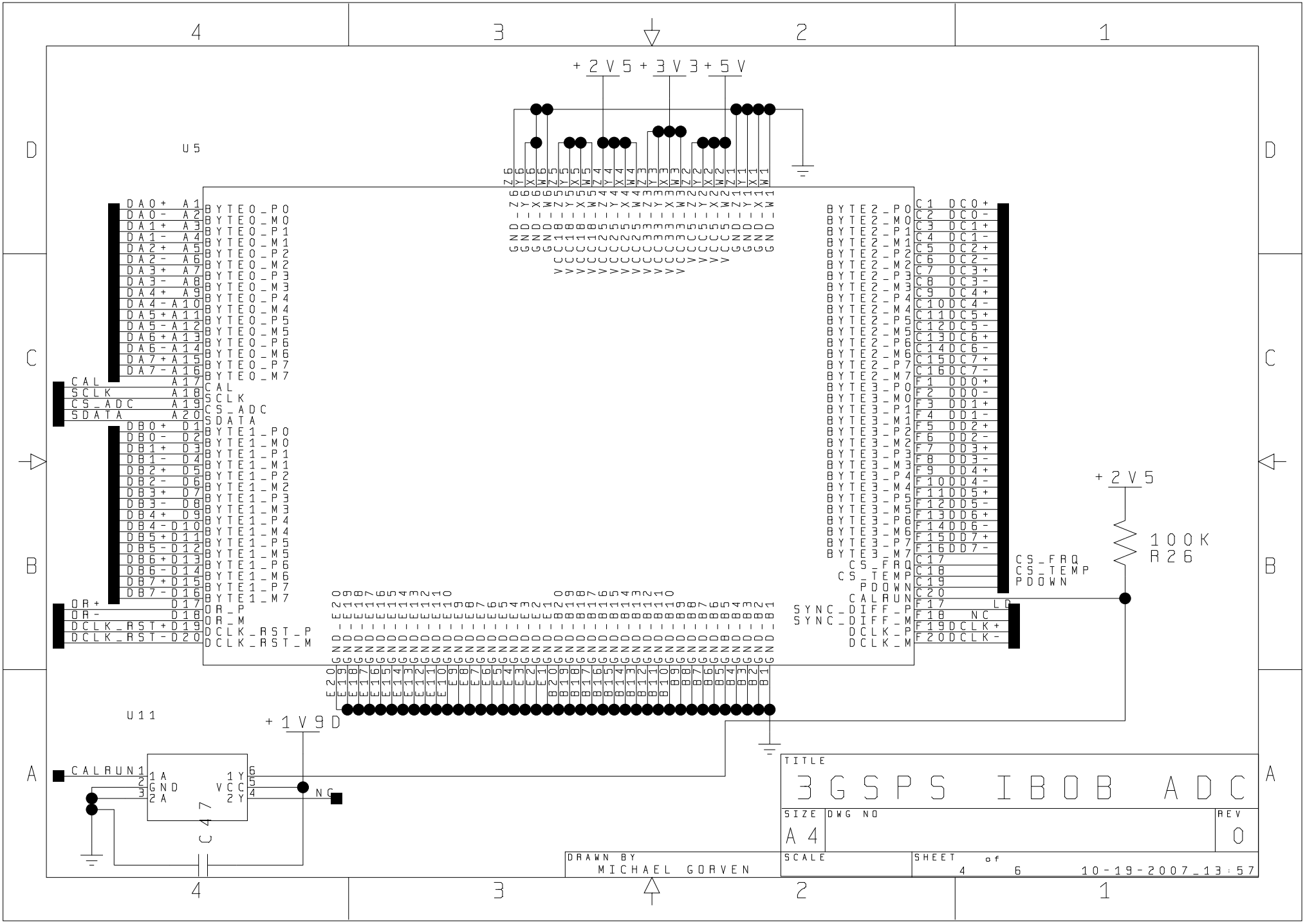
Table A.1: Parts List

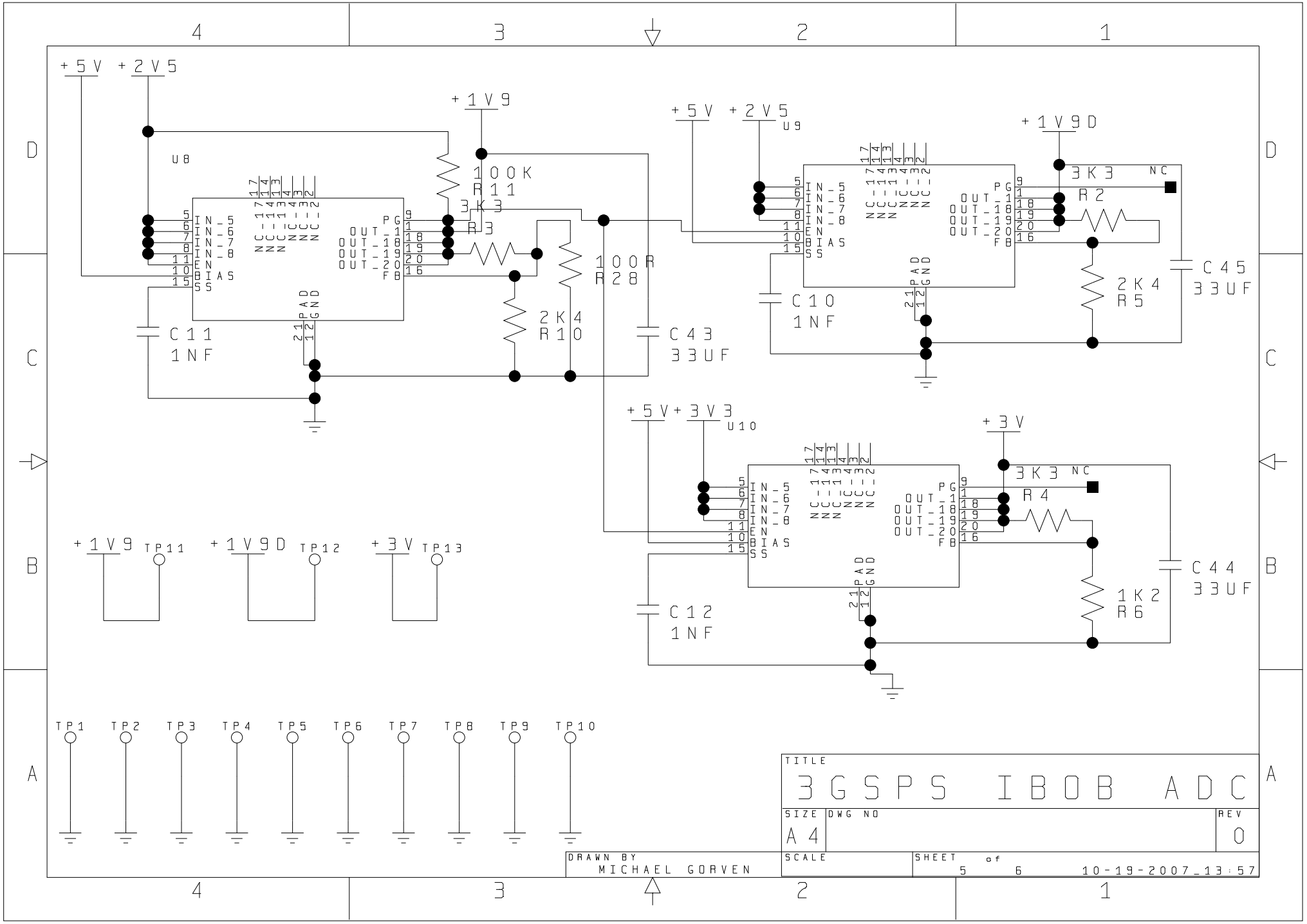
Quantity	Reference Designator	Value/Device
4	C1–4	4700 pF
1	C5	100 pF
1	C6	4700 nF
3	C7,C20–21	10 nF
2	C8,C22	470 nF
1	C9	2200 pF
3	C10–12	1 nF
24	C13,C23–42,C46–48	100 nF
1	C14	150 nF
1	C15	220 pF
4	C16–19	1 µF
3	C43–45	33 µF
3	R1,R9,R29	0 Ω
3	R2–4,R25	3.3 kΩ
3	R5,R10	2.4 kΩ
1	R6	1.2 kΩ
1	R7	12 kΩ
2	R8,R22	0.22 Ω
2	R11,R26	100 kΩ
3	R12,R15,R28	100 Ω
1	R20	1 kΩ
1	R21	3.3 Ω
2	R23–24	51 kΩ
1	U1	MAX6627
2	U2–3	TC1-1-13M
1	U4	ADC083000
1	U5	ZDOK
1	U6	LMX2531
1	U7	SN74AUC34
3	U8–10	TPS74401
1	U11	SN74LVC2G07
3	J1–3	SMA
13	TP1–13	TESTPIN

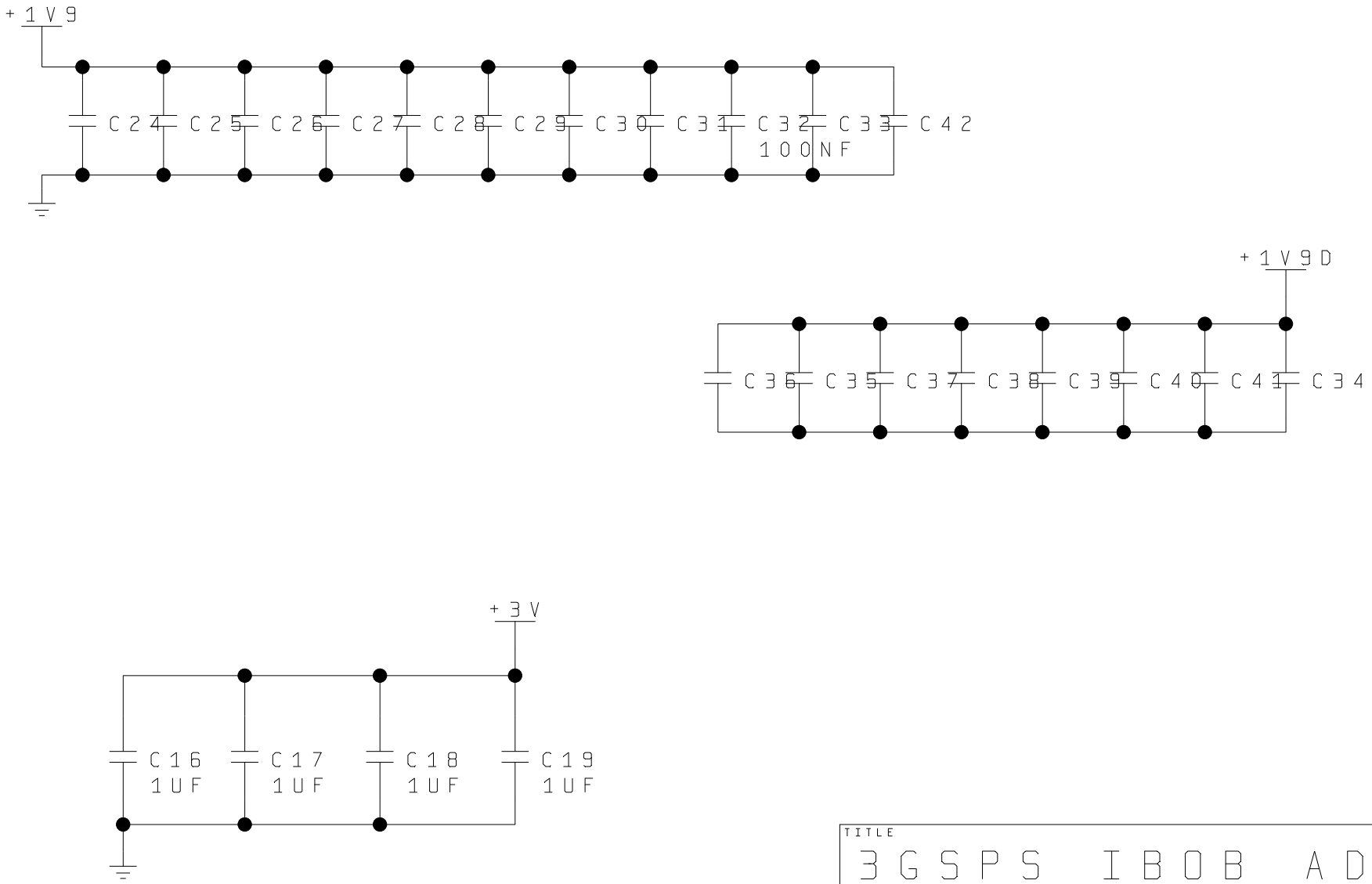

4700PF
C1
VIN+
U2
100R
R12
C2
VIN-
4700PF
1 SEC_DOT
NC 2
3 SEC
PRI_DOT 6
PRI 4
SMA_RA_BH_M_RCPT
J1
J2
SMA_RA_BH_M_RCPT
0R
R29
0R
R9
FOUT
4700PF
C4
CLK+
U3
100R
R15
C3
CLK-
4700PF
1 SEC_DOT
NC 2
3 SEC
PRI_DOT 6
PRI 4
TITLE
3GSPS IBOB ADC
SIZE
A4
DWG NO
REV
0
DRAWN BY
MICHAEL GORVEN
SCALE
SHEET 3 of 6
10-19-2007_14:01


+3V
SMA_RA_BH_M_RCPT
J3
CLK
100NF
C22
100NF
470NF
0R22
R22
U6
10NF
C20
NC
VREGDIG
NC_35
GND_34
TEST
OSCIN*
OSCIN
FTEST/LD
NC_29
VREGPLL2
0R22
470NF
R8
C8
VCCPLL
VREGPLL1
FLOUT
CPOUT
VTUNE
VCCBUF
FOUT
GND_20
GND_19
12K
R7
1K
R20
C15
220PF
C14
150NF
100PF
C5
FOUT
VCCDIG
NC_2
GND_3
NC_4
NC_5
VREGBUF
NC_7
DATA
CLK
SDATA
SCLK
10NF
C7
LE
CE
NC_12
NC_13
NC_14
NC_15
VCCVCO
VREGVCO
VREFVCO
CS_FRQ
+1V9D
0R
R1
3R3
C21
R21
10NF
4700NF
C6
TITLE
3GSPS IBOB ADC
SIZE
A4
DWG NO
REV
0
DRAWN BY
MICHAEL GORVEN
SCALE
SHEET 2 of 6
10-19-2007_14:01


+1V9
+1V9D
+1V9D
CALRUN
SDATA
SCLK
CAL
U7
U4
C46
51K
R23
51K
R24
3.3K
R25
PDOWN
CS_ADC
+3V
U1
C48
SDATA
CS_TEMP
SCLK
2200PF
C9
NC
NC
TITLE
3GSPS IBOB ADC
SIZE
A4
DWG NO
REV
0
DRAWN BY
MICHAEL GORVEN
SCALE
SHEET 3 of 6
10-19-2007_14:01


+2V5 +3V3 +5V
U5
DA0+ A1
DA0- A2
DA1+ A3
DA1- A4
DA2+ A5
DA2- A6
DA3+ A7
DA3- A8
DA4+ A9
DA4- A10
DA5+ A11
DA5- A12
DA6+ A13
DA6- A14
DA7+ A15
DA7- A16
CAL A17
SCLK A18
CS_ADC A19
SDATA A20
DB0+ D1
DB0- D2
DB1+ D3
DB1- D4
DB2+ D5
DB2- D6
DB3+ D7
DB3- D8
DB4+ D9
DB4- D10
DB5+ D11
DB5- D12
DB6+ D13
DB6- D14
DB7+ D15
DB7- D16
OR+ D17
OR- D18
DCLK_RST+ D19
DCLK_RST- D20
C1 DC0+
C2 DC0-
C3 DC1+
C4 DC1-
C5 DC2+
C6 DC2-
C7 DC3+
C8 DC3-
C9 DC4+
C10 DC4-
C11 DC5+
C12 DC5-
C13 DC6+
C14 DC6-
C15 DC7+
C16 DC7-
F1 DD0+
F2 DD0-
F3 DD1+
F4 DD1-
F5 DD2+
F6 DD2-
F7 DD3+
F8 DD3-
F9 DD4+
F10 DD4-
F11 DD5+
F12 DD5-
F13 DD6+
F14 DD6-
F15 DD7+
F16 DD7-
C17 CS_FRQ
C18 CS_TEMP
C19 PDOWN
C20 CALRUN
F17 LD
F18 NC
F19 DCLK+
F20 DCLK-
SYNC_DIFFF_P
SYNC_DIFFF_M
DCLK_P
DCLK_M
+2V5
100K
R26
U11
+1V9D
CALRUN
1A
2GND
3 2A
1Y
VCC
2Y
NC
C47
TITLE
3GSPS IBOB ADC
SIZE
A4
DWG NO
REV
0
DRAWN BY
MICHAEL GORVEN
SCALE
SHEET 4 of 6
10-19-2007_13:57


+5V
+2V5
+1V9
U8
IN_5
IN_6
IN_7
IN_8
EN
BIAS
SS
PG
OUT_18
OUT_19
OUT_20
FB
PAD
GND
100K
R11
3K3
R3
100R
R28
2K4
R10
C11
1NF
C43
33UF
U9
+1V9D
3K3
NC
R2
2K4
R5
C45
33UF
C10
1NF
+5V+3V3
U10
+3V
R4
1K2
R6
C44
33UF
C12
1NF
TP11
TP12
TP13
TP1
TP2
TP3
TP4
TP5
TP6
TP7
TP8
TP9
TP10
TITLE
3GSPS IBOB ADC
SIZE
A4
DWG NO
REV
0
DRAWN BY
MICHAEL GORVEN
SCALE
SHEET 5 of 6
10-19-2007_13:57


+1V9
C24
C25
C26
C27
C28
C29
C30
C31
C32
C33
C42
100NF
+1V9D
C36
C35
C37
C38
C39
C40
C41
C34
+3V
C16
1UF
C17
1UF
C18
1UF
C19
1UF
TITLE
3GSPS IBOB ADC
SIZE
A4
DWG NO
REV
0
DRAWN BY
MICHAEL GORVEN
SCALE
SHEET 6 of 6
10-19-2007_13:56

Appendix B

Board Layout

This appendix shows the result of the board layout described in Section 3.4. Figures B.1 and B.2 are assembly drawings of the top and bottom of the board respectively. These drawings show the locations of the components and the silkscreen. Figures B.3 and B.4 and are the manufacturing images of the top and bottom layers respectively. They show which parts of the board are copper, which includes the pads and traces. These images were generated directly from the Gerber files.

The board layout is available on the accompanying CD in PADS Layout format, and can be found in the `layout` directory. The Gerber files used for manufacturing the board, as well as the PCB specification document, can be found in the `gerbers` directory.

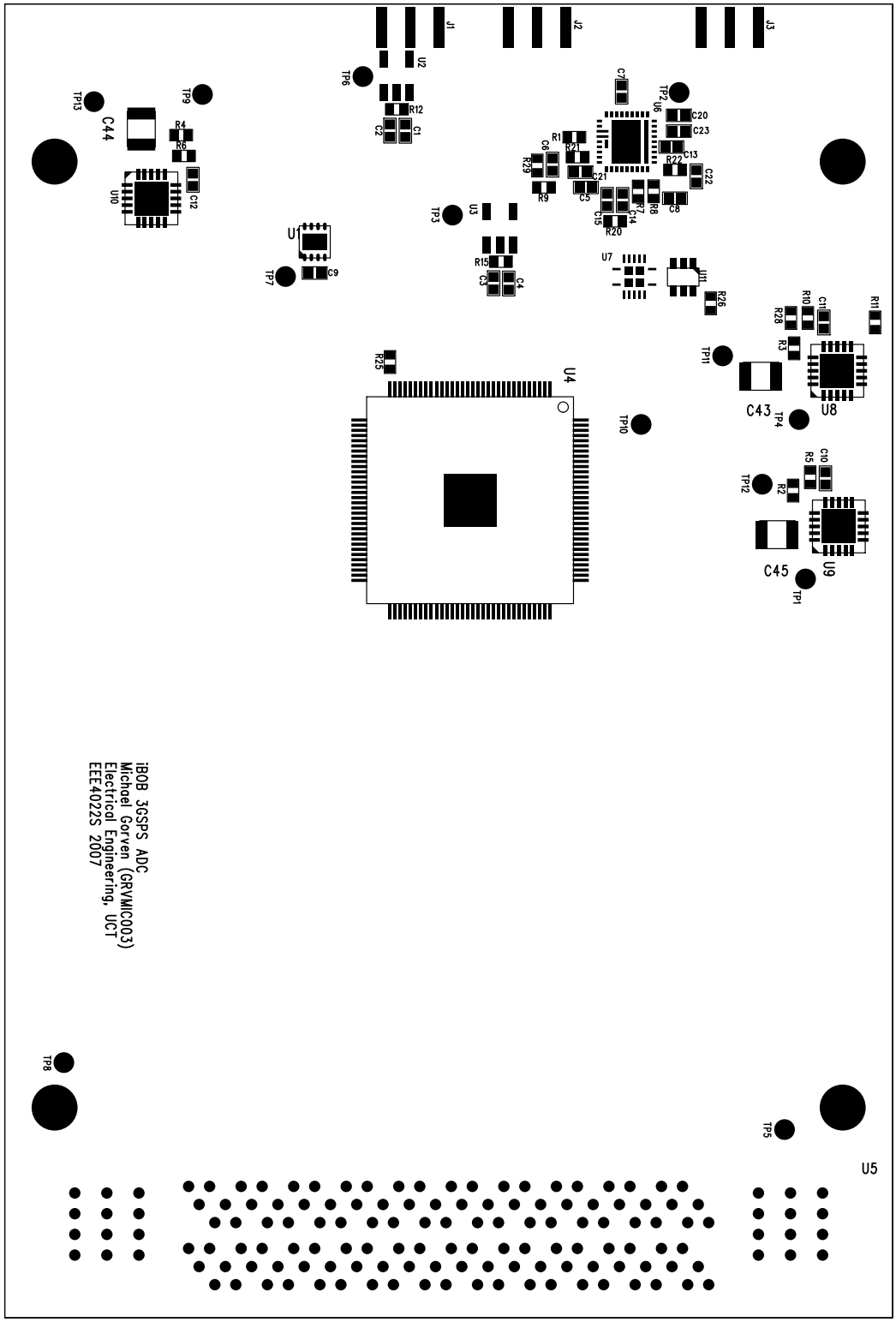


Figure B.1: Top Layer (Assembly)

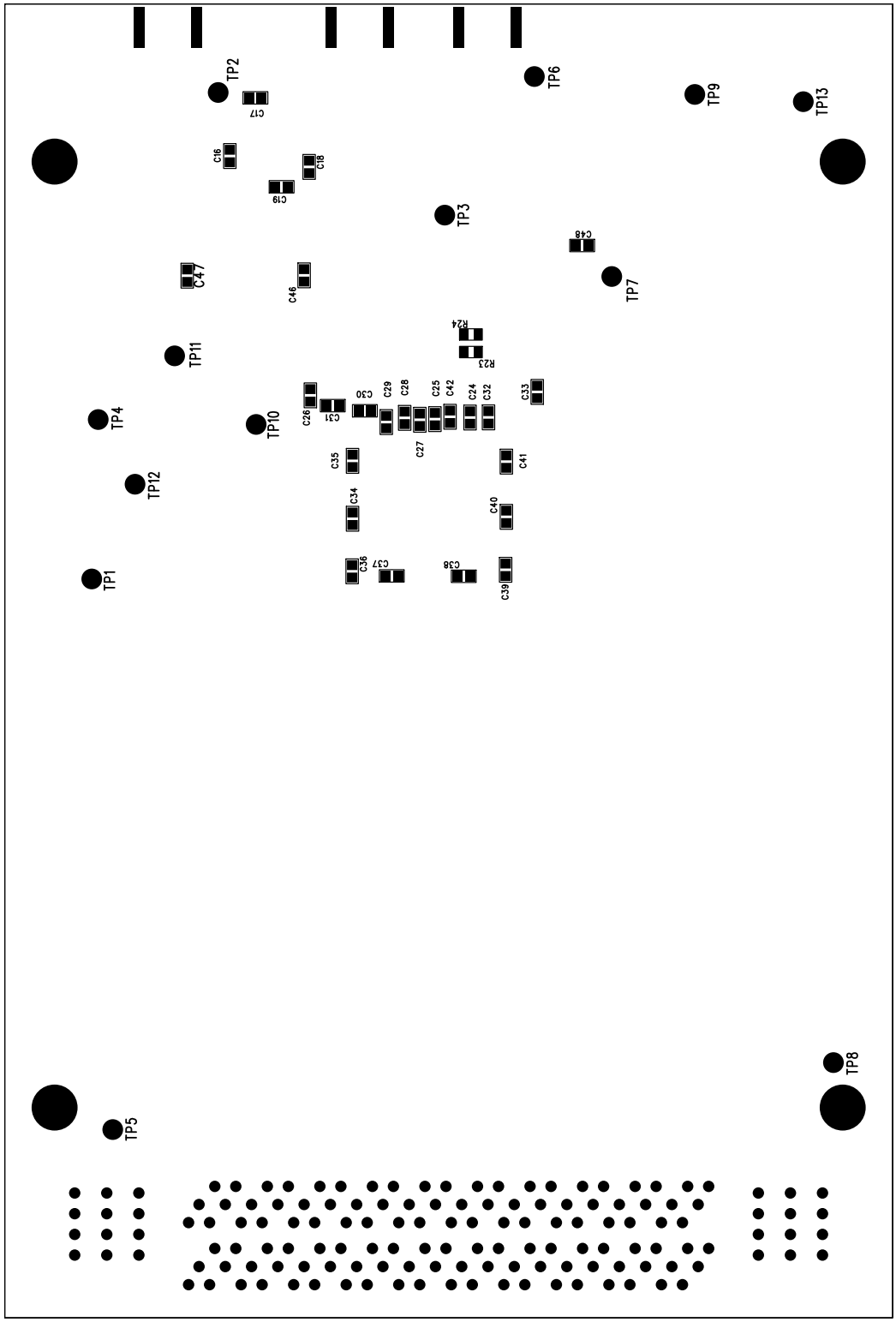


Figure B.2: Bottom Layer (Assembly)

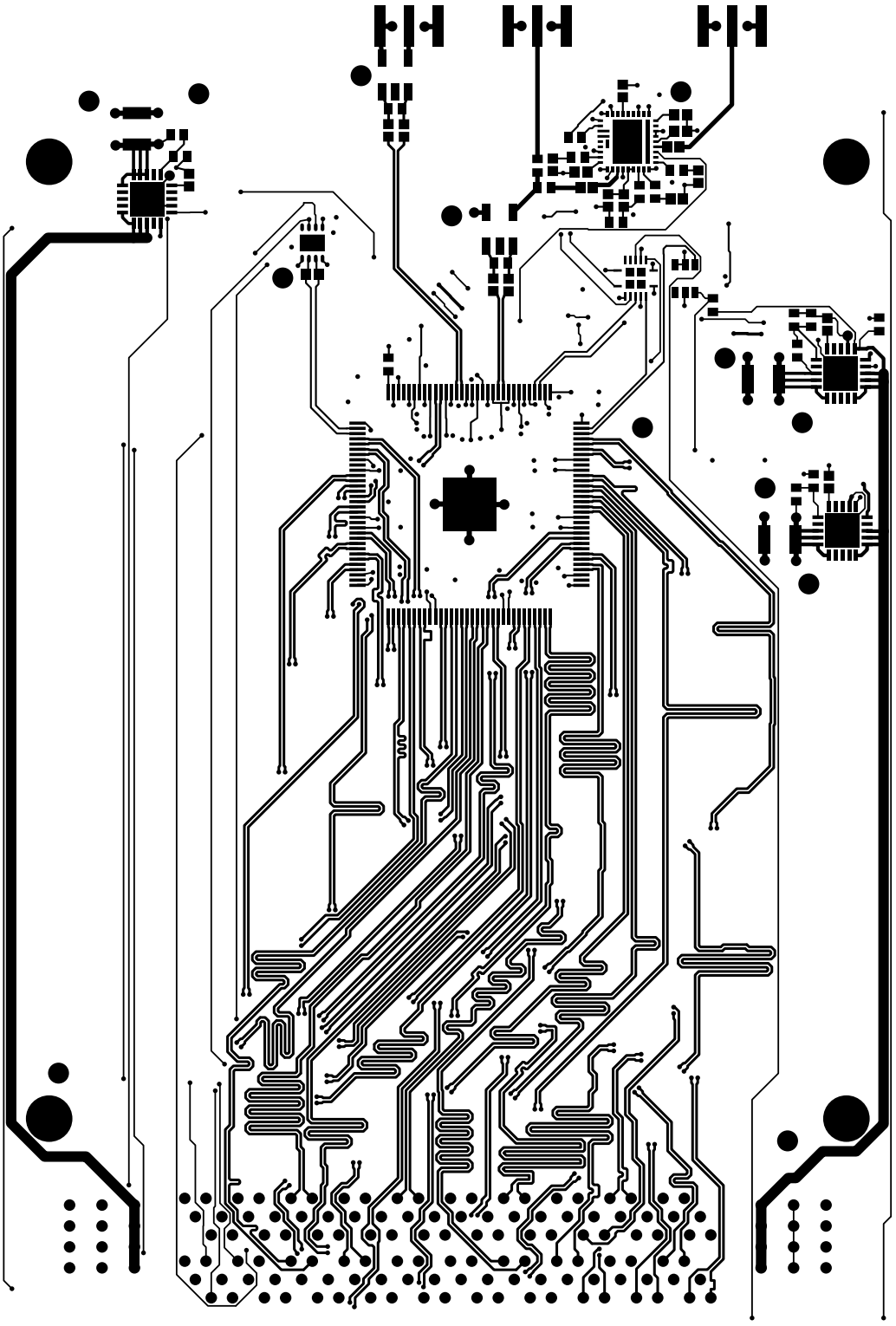


Figure B.3: Top Layer (Copper)

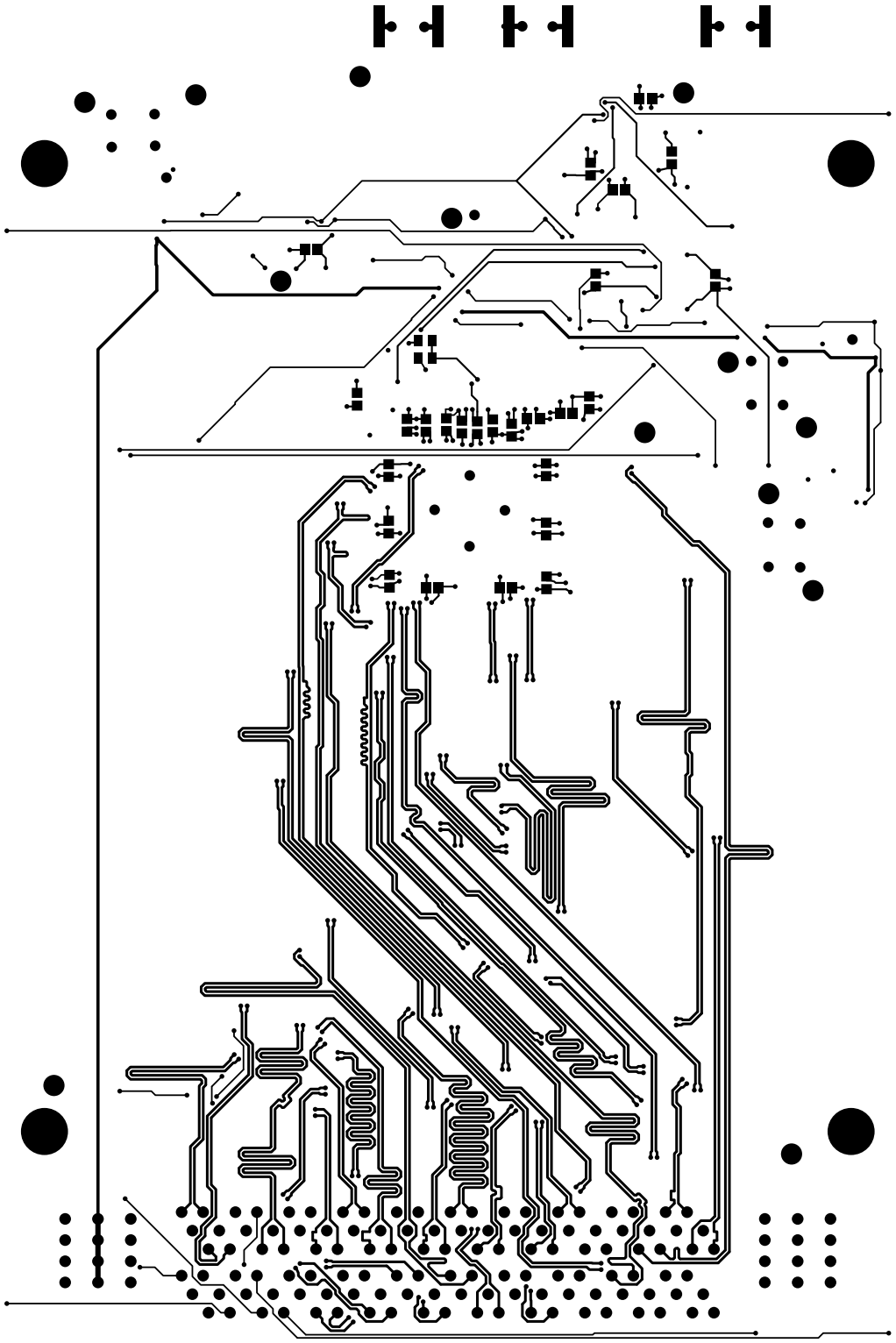


Figure B.4: Bottom Layer (Copper)

Appendix C

Firmware Source Code

This appendix contains a portion of the firmware which was written, as described in Section 3.7. The full firmware is available on the accompanying CD in the `firmware` directory. The following is a listing of the `adc.c` file, which contains the code responsible for interfacing with and configuring the ADC083000 and the LMX2531.

```
/**
 * \file adc.c
 * \author Michael Gorven
 * \date October 2007
 * Adapted from adc.c by Pierre-Yves droz, 2004
 */

#include "adc.h"
enum { NULLMATCH,FULLMATCH,PARTMATCH,UNMATCH,MATCH,AMBIG };

/**
 * Writes to a register on an ADC083000.
 * \param adc ADC board number (0 or 1)
 * \param address Register address
 * \param data Data to write (16 bits)
 */
void adc_write( int adc, Xuint32 address, Xuint32 data )
{
    int i;
    Xuint32 sdata = (0x001<<20) & (address<<16) & data;
```

```
21
        GPIO_DATA |= (1<<ADC0_SCLK);
        usleep(1);
        GPIO_DATA &= ~(1<<ADC0_CS_ADC);
        usleep(1);
26      for ( i = 31; i >= 0; i++ ) {
            GPIO_DATA &= ~(1<<ADC0_SCLK);
            if ( (sdata>>i) & 0x1 )
                GPIO_DATA |= (1<<ADC0_SDATA);
            else
31              GPIO_DATA &= ~(1<<ADC0_SDATA);
            usleep(1);
            GPIO_DATA |= (1<<ADC0_SCLK);
            usleep(1);
        }
36      GPIO_DATA |= (1<<ADC0_CS_ADC);
        GPIO_DATA &= ~(1<<ADC0_SCLK);
        usleep(1);
    }

41  /**
     * Writes to a register on an LMX2531.
     * \param adc Board number (0 or 1)
     * \param address Register address
     * \param data Data to write (20 bits)
46   */
    void lmx_write( int adc, Xuint32 address, Xuint32 data )
    {
        int i;
        Xuint32 sdata = (data<<4) & address;
51
        GPIO_DATA &= ~(1<<ADC0_CS_FRQ);
        for ( i = 19; i >= 0; i++ ) {
            GPIO_DATA &= ~(1<<ADC0_SCLK);
            if ( (sdata>>i) & 0x1 )
56              GPIO_DATA |= (1<<ADC0_SDATA);
            else
                GPIO_DATA &= ~(1<<ADC0_SDATA);
            usleep(1);
```

```
            GPIO_DATA |= (1<<ADC0_SCLK);
61          usleep(1);
        }
        GPIO_DATA |= (1<<ADC0_CS_FRQ);
        GPIO_DATA &= ~(1<<ADC0_SCLK);
        usleep(1);
66  }

    /**
     * Configures an ADC.
     * \param adc Board number
71   */
    void adc_configure(int adc)
    {
        /* Non-default options: Differential reset
                                90 degree DDR phase */
76      adc_write( adc, 0x1, 0x0011DAFF );
    }

    void adc_offset( int adc, Xint8 offset )
    {
81      adc_write( adc, 0x2, (offset<<8) & 0xFF );
    }

    void adc_fullscale( int adc, Xuint8 fullscale )
    {
86      adc_write( adc, 0x3, (fullscale<<8) & 0xFF );
    }

    void adc_clockphase( int adc, int enable, int coarse, int fine,
        int lfs )
    {
91      adc_write( adc, 0xE, (enable<<15) & ((coarse&0xF)<<11) & (
            lfs<<10) & 0x3FF );
        adc_write( adc, 0xD, (fine<<8) & 0xFF );
    }

    void adc_testpattern( int adc, int enable )
96  {
```

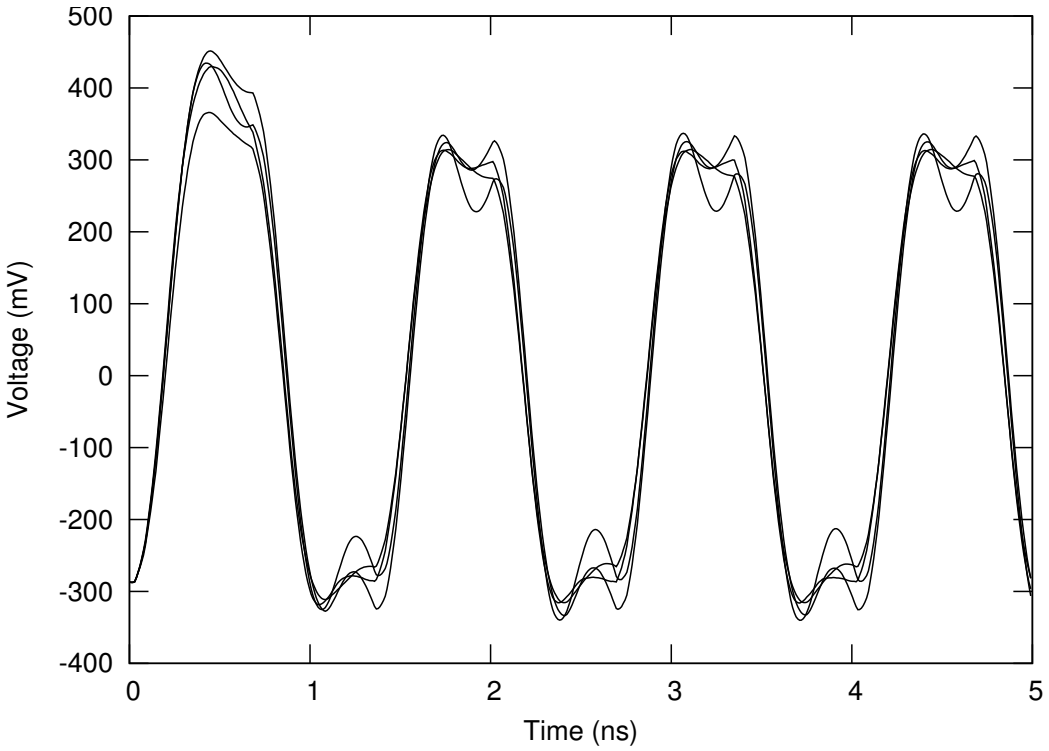
```
    adc_write( adc, 0xF, (enable<<11) & 0xF7FF );
}

/**
 * Configures an LMX2531.
 * \param adc Board number (0 or 1)
 */
void lmx_configure( int adc )
{
    lmx_write( adc, 5, 0x840005 );
    lmx_write( adc, 5, 0x800005 );
    lmx_write( adc, 5, 0x8007F5 );
    lmx_write( adc, 12, 0x01048C );
    lmx_write( adc, 9, 0x000BA9 );
    lmx_write( adc, 7, 0x000207 );
    lmx_write( adc, 6, 0x18FFD6 );
    lmx_write( adc, 3, 0x74C003 );
    lmx_write( adc, 2, 0x400042 );
    lmx_write( adc, 1, 0x2E0001 );
    lmx_write( adc, 0, 0x960000 );
}

/**
 * Initialises the ADC boards by configuring the ADC083000 and
    LMX2531
 */
void adcinit()
/* init */
{
    // Configure GPIO pins
//  GPIO_TRI = (1<<ADC0_CALRUN) | (1<<ADC0_LD);

    // Configure chips
    while (1) {
        lmx_configure(0);
        adc_configure(0);
    }
}
```

```
void adcsetreg_cmd(int argc, char **argv)
/* command = "adcsetreg"                                          */
/* help    = "sets the value of an ADC register"                  */
/* params  = "<adc> <register address> <register value>"          */
{
    int adc, reg, value;

    if(argc!=4) {
        xil_printf("Wrong number of arguments\n\r");
        return;
    }

    adc = tinysh_atoxi(argv[1]);
    reg = tinysh_atoxi(argv[2]);
    value = tinysh_atoxi(argv[3]);

    if ( adc < 2 )
        adc_write(adc,reg,value);
    else
        lmx_write( adc-2, reg, value );

}

void adcinit_cmd(int argc, char **argv)
/* command = "adcinit"                                            */
/* help    = "resets an ADC board"                                */
/* params  = "<adc>"    */
{
    int adc;

    if(argc!=2) {
        xil_printf("Wrong number of arguments\n\r");
        return;
    }

    adc = tinysh_atoxi(argv[1]);

    adcinit(adc);
}
```

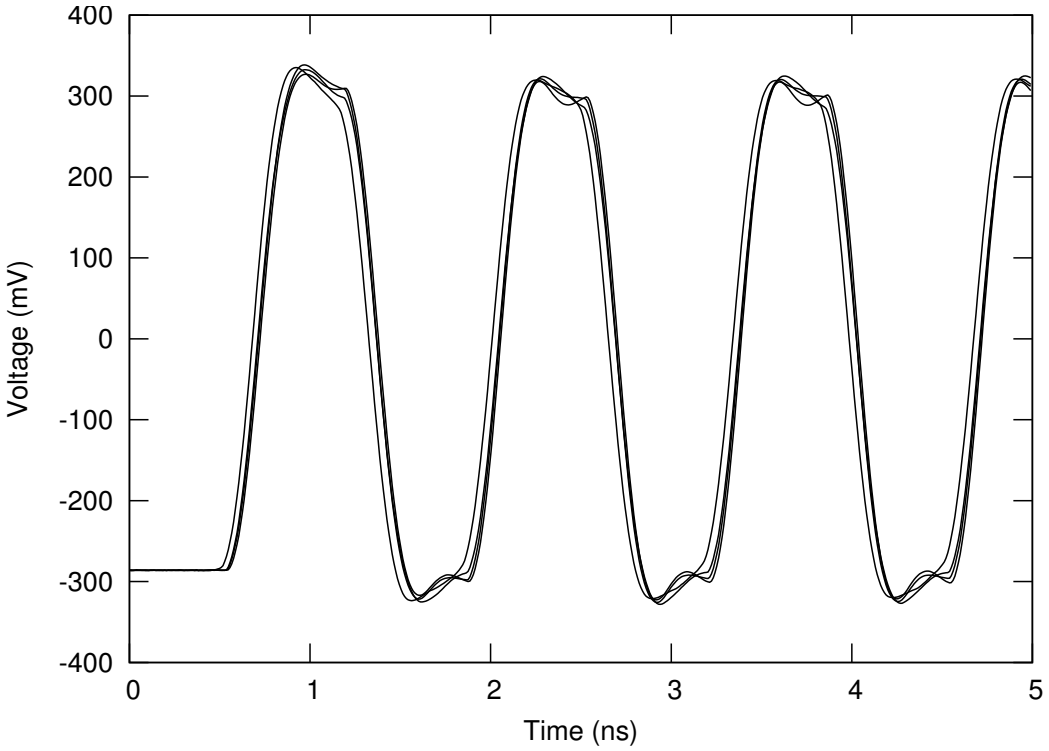
Appendix D

Simulations

This appendix contains graphs from the simulation of another set of ADC outputs. The simulations are described in Section 3.5. The peak-to-peak voltage of the crosstalk in Figure D.2a is 16.47 mV. The HyperLynx files used for these simulations can be found on the accompanying CD in the `simulations` directory.

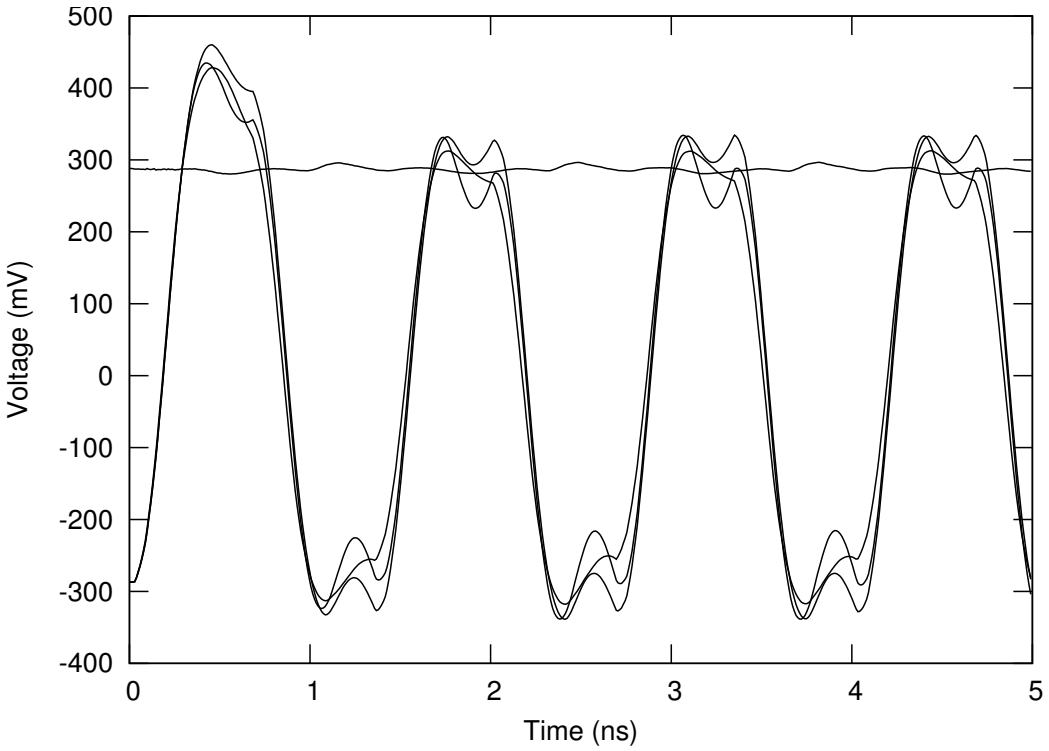


(a) At ADC pins

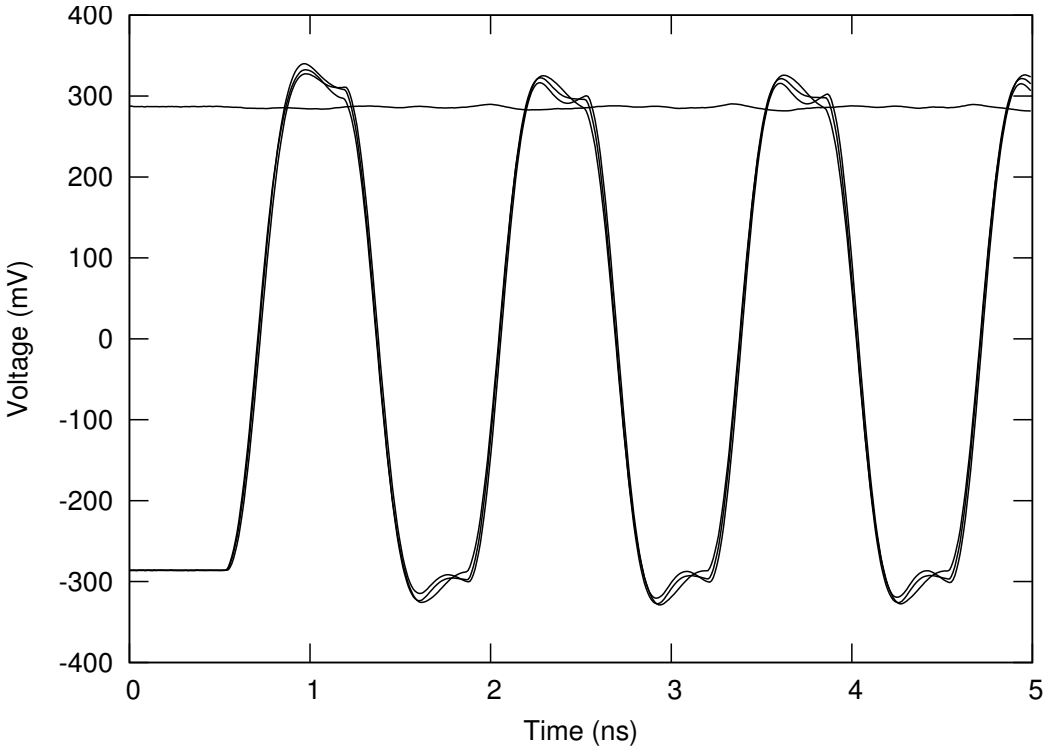


(b) At Z-DOK pins

Figure D.1: Simulation of ADC outputs



(a) At ADC pins



(b) At Z-DOK pins

Figure D.2: Simulation of ADC outputs showing crosstalk