5-2006

# The Development of a Capacitance-Based Biotelemetry System for Implantable Applications

Calum John Johnson
*University of Tennessee - Knoxville*

To the Graduate Council:

I am submitting herewith a thesis written by Calum John Johnson entitled "The Development of a Capacitance-Based Biotelemetry System for Implantable Applications." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Benjamin J Blalock, Major Professor

We have read this thesis and recommend its acceptance:

Charles L. Britton, M. Nance Ericson

Accepted for the Council:<br>
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Calum John Johnson entitled "The Development of a Capacitance-Based Biotelemetry System for Implantable Applications." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Benjamin J. Blalock

Major Professor

We have read this thesis
and recommend its acceptance:

Charles L Britton

M. Nance Ericson

Accepted for the Council:

Anne Mayhew

Vice Chancellor and Dean of Graduate Studies

(Original signatures are on file with official student records)

# The Development of a Capacitance-Based Biotelemetry System for Implantable Applications

A Thesis Presented for the

Master of Science Degree

The University of Tennessee, Knoxville

Calum John Johnson

May 2006

## Acknowledgements

I would like to thank all the teachers, both academic and non-academic, who have influenced me throughout my studies. I am especially grateful for the education I received as an undergraduate at Tennessee Technological University because it is the foundation of my career. I am also grateful to the University of Tennessee for continuing my education at the graduate level and helping me expand my horizons. I would also like to thank Oak Ridge National Laboratory for the assistantship that made this work possible and for providing an environment in which I could learn and succeed.

I would like to all the members of my committee, Dr. B. J. Blalock, Dr. C. L. Britton, and Dr. M. N. Ericson, for their efforts and patience in helping me create this thesis. I would like to especially thank Dr. M. N. Ericson for taking such a personal role in my work.

I would like to thank my family for their support, encouragement, and understanding. I would like to especially thank Leah E. Watson for her love and support.

Finally I would like to thank God for making this and all things possible.

# Abstract

Most modern biomedical implants implement some form of communications link between the implant and the outside world. This biotelemetry link has many requirements such as data bandwidth and power consumption. Designing an appropriate link that meets these requirements is one of the most significant engineering challenges associated with these implants. Communications methods that are currently used for this link include standard Radio Frequency (RF) approaches, inductively coupled approaches, and load modulation approaches. This thesis describes the development of a unique capacitance-based biotelemetry system for implantable applications. This system consists of two distinct parts: the implanted transmitter and the external body-mounted receiver. The prototype transmitter is based on a custom Application Specific Integrated Circuit (ASIC) fabricated using the AMI 1.5µ process. This ASIC encodes and transmits predetermined data packets by driving two electrodes in a slew-controlled manner, all contained within a biocompatible material. The receiver consists of charge-sensitive amplifier front end using a discriminator to distinguish individual bits. A Field Programmable Gate Array (FPGA) decodes the transmitted data and relays it to a PC-based LabVIEW interface. Test results using a saline-based human tissue model are presented.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1 INTRODUCTION

## 1.1 Telemetry in Implantable Systems

Implanted sensors have been in use in some form for almost 50 years for measuring physiological parameters such as heart rate, electrocardiogram (ECG), electroencephalogram (EEG), temperature, pH, and various pressures [1]. Clinical applications of sensor-based implants are increasing rapidly as recent advances in Low Voltage, Low Power (LVLP) integrated circuits and sensing techniques are enabling new areas of monitoring including blood perfusion and neural activity [2, 3]. The earliest example of implantable controllable actuators is externally programmable pacemakers that became commercially available in the early 1980s [4]. Newer implant types include those for implantable drug delivery systems, urinary incontinence, pain management, movement disorders like Parkinson's disease, and retinal prostheses [5].

Wireless communication with implanted systems is preferred, as transcutaneous wires significantly increase the risk of infection and may cause a high level of patient discomfort. However, communicating wirelessly through the human body is a complex problem, the solution to which depends largely on the particular application. There is no universal communication system that is the optimum solution for every type of implant. Many limitations such as the implant's size, function, power consumption, implanted location, required telemetry distance, and patient environment play an important role in determining the optimum communication technique. Designing a communication system to meet all of these requirements is one of the most challenging aspects of implant design.

This chapter reviews many of the primary design challenges associated with data transmission through tissue and provides an overview of the primary wireless communication systems employed in modern implanted systems. An alternative technique based on capacitve coupling is introduced and compared to these more commonly employed methods.

## 1.2 Biological Interaction

For optimum results, any wireless system must be designed for an explicit transmission media. When designing a wireless system typically what is examined is how the transmission media influences the communication. The most common influence examined is how the media limits the transmission distance or in other words how far reliable communication can be achieved given a fixed amount of transmitting power. When designing a wireless system for communicating through tissue however, the problem is exacerbated: not only must the effects of how the tissue influences the communication be examined, but also how the communication affects the tissue. A wireless system that causes severe damage to surrounding tissue is not a valid solution for implant communication. The problem is made harder by the fact that the electrical properties of tissues are not uniform for all types of tissues. For example, the conductivity of fat for a given frequency can be twenty times or more than that of muscle.

The following subsections discuss the complicated two-way interaction of tissue and communications.

### 1.2.1 Biological Effects on Communications

Tissue can be separated into two general classes: tissues with high fluid content and tissues with low fluid content. The fluids contained in the tissue are composed of salt ions, polar protein molecules, and polar water molecules. Thus, the response of tissues to electromagnetic fields is largely dominated by ionic conduction, a phenomenon strongly coupled to fluid content. Conduction currents in the tissue are caused by the oscillations of free charges (ions) when a time-varying electromagnetic field is applied to the tissue. The rotation of the dipole molecules causes changes in the displacement currents. Ohmic losses and dielectric losses cause the power dissipation of the conduction and displacement currents respectively [5].

Tables 1.1 and 1.2 summarize the properties of high and low fluid content tissues, respectively. From these tables several trends can be established. The most important trend gathered from the tables is that for high fluid content tissues, such as those found in the abdomen, higher frequencies have a lower depth of penetration for a given transmitting power and are thus less efficient. These tables are meant as a general guide; more exact modeling can be accomplished through the use of detailed computer models of the human body [6].

### 1.2.2 Communications Effects on Biology

The harmful effects of communication on the biological media can be broken down into three classes: interference, athermal biological effects, and thermal heating, and each of these poses certain limitations on the communication link.

**Table 1.1: Electromagnetic properties for high fluid content tissue**

| *Frequency (MHz)* | *Wavelength in Air (cm)* | *Dielectric Constant $\varepsilon_D$* | *Conductivity $\rho_L$ (mho/m)* | *Wavelength in Tissue $\lambda_L$ (cm)* | *Depth of Penetration (cm)* |
|---|---|---|---|---|---|
| ***1*** | ***30000*** | ***2000*** | ***0.400*** | ***436*** | ***91.3*** |
| ***10*** | ***3000*** | ***160*** | ***0.625*** | ***118*** | ***21.6*** |
| ***1000*** | ***30*** | ***50*** | ***1.7*** | ***2.8*** | ***2.4*** |
| ***10000*** | ***3*** | ***39.9*** | ***10.3*** | ***0.46*** | ***0.34*** |

**Table 1.2: Electromagnetic properties for low fluid content tissue**

| *Frequency (MHz)* | *Wavelength in Air (cm)* | *Dielectric Constant $\varepsilon_D$* | *Conductivity $\rho_L$ (mho/m)* | *Wavelength in Tissue $\lambda_L$ (cm)* | *Depth of Penetration (cm)* |
|---|---|---|---|---|---|
| ***1*** | ***30000*** | | | | |
| ***10*** | ***3000*** | | | | |
| ***100*** | ***300*** | ***7.45*** | ***19-76*** | ***106*** | ***60.4*** |
| ***1000*** | ***30*** | ***5.6*** | ***71-171*** | ***8.4*** | ***13.9*** |
| ***10000*** | ***3*** | ***4.5*** | ***324-549*** | ***1.4*** | ***3.4*** |

The first of these effects occurs when electrical signals are either directly interfering with a natural biological signal or when parts of the body misinterpret the electrical signal as a natural biological signal. However, because natural biological signals are generally in a very low frequency range (DC to approximately 3 KHz) this type of effect is easily preventable [7].

The second of these effects, athermal biological effects, is a general classification of non-thermal biological responses to electromagnetic fields. These effects include plasma membrane damage within cells, immune system changes, neurological effects, behavioral effects, interaction of certain drug and chemical compounds to electromagnetic fields, action potential disturbance, and effects on DNA [8]. While studies have shown that very high levels of electromagnetic fields can cause these effects [5], they have not shown conclusive evidence that low level exposure to electromagnetic fields is harmful.

The third and most common type of effect is thermal heating. The energy needed to produce thermal heating comes from the power dissipation of the electromagnetic wave in the tissue through both Ohmic and dielectric losses. The operation of Microwave ovens is dependant on this same principle. The quantity used to characterize the absorption of electromagnetic energy is known as the specific absorption rate (SAR), and is expressed in the units of watts per kilogram (W/kg).

Any dissipation of RF energy will cause some amount of thermal heating, but this heating is only harmful when the amount of heating becomes greater than that associated with the natural metabolic rate of the body, or approximately 1 Watt/kg. The FCC mandated limit for RF exposure is 0.4 W/kg for the whole body under controlled conditions, and 8 W/kg for only partial of the body exposure [8]. Exceptions are made for certain body parts such as the ankles, wrists, hands, and feet which can have higher exposure rates, and the eyes and testes, which have lower exposure rates due to their relative lack of available blood flow to dissipate excessive heating.

## 1.3 Current Wireless Implant Biotelemetry Methods

The following subsections summarize current approaches for implant communications: standard Radio Frequency (RF) techniques based on Amplitude Modulation (AM), Frequency Modulation (FM), and other modulations, inductive RF approaches for data and power transfer, load or passive reflection modulation, spread spectrum approaches, and techniques based on current injection.

### 1.3.1 RF Approaches Based on Common Modulation Techniques

Narrow band RF data transmission is one of the most common techniques employed in biotelemetry. Of the modulation techniques used in narrow band transmissions, one of the most common in implanted systems is frequency modulation (FM). FM-based approaches are attractive because they are less sensitive to received signal strength than some other approaches, thus the movement of the internal and external antennae in relation to each other does not affect signal recovery. This is one reason why an FM-based approach was popular in early implants where analog data was directly transmitted [1].

The transmission of digital signals has seen the rise of several other modulation techniques employed in implanted systems, namely on-off keying (OOK) [9], amplitude-shift keying (ASK) [10,11], phase-shift keying (PSK) [12], and frequency-shift keying (FSK) [13]. Although ASK is the simplest to implement, the FSK and PSK offer increased robustness to noise, making them attractive for certain applications.

### 1.3.2 Inductively Coupled RF Power

Inductive data and power transfer works by the principle of a time-varying magnetic flux density producing electrical energy, as described by Faraday's law:

$$\mathbf{V = -N\ \partial\Phi/\partial t}$$

Generally, a loop antenna (also known as the primary coil) is used to create the time-varying magnetic flux by having a time-varying current applied to it (Ampere's law). This in turn is coupled to another loop antenna (or the secondary coil) of *N* turns that converts the flux to a voltage.

The amount of flux that is successfully coupled depends on a number of different factors, the most important being the coil sizes and number of turns, coil separation distance, alignment of the coils, and the transmitted frequency. In an implanted system, each of these factors is limited by either the implant's physical requirements or by limitations imposed by the surrounding biological media. The limitation on the implant's size sets a maximum limit on the size of the secondary coil. However, tethering the secondary coil from the implant and positioning it on the surface can minimize the separation distance and improve alignment between the secondary coil and the primary coil placed on the skin. The transmitted frequency is a compromise: higher frequencies transfer more power because of the higher time derivative of the magnetic flux, but higher frequencies also have more loss in tissue, as described in section 1.2.1. Typically a transmitting frequency around 2MHz is used for power transfer [14].

Information from the external unit to the implant can be easily piggybacked on the transmitted power signal by modulating the signal. Information may be transmitted from the implant to the external unit through a variety of methods, but usually one of three methods is employed: an additional RF transmitter and antenna is used (see section 1.3.1), a passive load-modulation technique is used (see section 1.3.3), or a very short return pulse is sent using the secondary coil as the transmitting antenna [15]. In the last case the transmitting and receiving roles of the coils are reversed, although much less power is transmitted from the implant compared to the power transmitted from the external unit. Pulse position modulation (or varying the time when the return pulse is transmitted) is the most common modulation technique used when this method is employed.

### 1.3.3 Passive Load-Modulation

Load-modulation techniques control a change in reflected impedance between two coupled elements to transmit information passively to a receiver. In such a system, the receiver transmits a RF signal that is coupled in the near field to an implant whose load is modulated. Typically both the transmitter and receiver are inductively coupled coils. Shunting of the implanted coil to ground causes a change in reflected impedance seen by the receiver, enabling data communication. This technique is becoming more common in implant biotelemetry [5]. Application of this technique to implant biotelemetry was initially used in animal implants before becoming more common in human implants [16]. The foremost advantage this technique offers is the ability to providing power to the implant while using the backscattered RF signal for data communications. This technique is especially useful in simple, very low power implants such as RFID tags used for tracking and identification of individual animals [17]. The drawback is the same as any inductively coupled technique: reliable communication (and power transfer) can only occur if the coils are sufficiently coupled. This can be easily seen from the quadratic dependence of the coupling factor in the equation for the reflected load impedance,

$$Z_{ref} = (1-k^2)/j\omega C_P + (L_P/L_S)*R_{load}k^2$$

Here $k$ is the coupling factor, $C_P$ is the primary capacitance, $L_P$ is the primary inductance, $L_S$ is the secondary inductance, and $R_{load}$ is the secondary load [5]. The coupling factor has a large dependence on both the shape of the coil (where larger is better) and the distance between the two coils (where shorter is better). These factors together limit the implant to either shallow implantation depth or fairly large size.

One novel variation of this technique is in using thin-wire dipoles coupled to the electric field components of an external electromagnetic signal rather than using the standard inductively coupled coils [18]. This offers the advantage that the antenna dimensions are reduced to one dimension rather than two, but there is a drawback in that frequency choices become limited due to realistic limitations on antenna size. For some applications, however, a long, thin antenna could be appropriate. In order to make the antenna length reasonably short, a higher transmitting frequency (and thus shorter wavelength) must be used. This is complicated by two factors. The first of these, as was shown earlier in this chapter, is that higher frequency signals have a lower depth of penetration in a high water content biological medium. The second complication, unlike inductive backscatter modulation techniques, is that the returned signal is actually a reflected second harmonic of the transmitted signal. This compounds the problem of balancing frequency and the resulting depth of penetration versus antenna length.

### 1.3.4 Direct-Sequence Spread Spectrum

Spread spectrum communication systems can be described by three basic parameters [19]:

1) The transmitted signal is spread over bandwidth much greater than the bandwidth needed to transmit the information.
2) The transmitted bandwidth is determined by some function that is independent of the information that is transmitted.
3) The receiver uses the same function to recover the transmitted data, and does so in a synchronous manner with the transmitter.

In Direct-Sequence Spread Spectrum (DSSS) these parameters are met by first modulating the data by a pseudo-random code, otherwise known as a spreading code, operating at a much higher clock frequency than the data. The resulting data is then modulated to the desired frequency and transmitted.

The standard advantages of a spread spectrum to other applications also apply to implant applications: namely a much greater reduction in interference with other signals, especially narrow band signals. The opposite is also true; the transmitted spread-spectrum signals are themselves more resistant to interference caused by narrow band signals operating in the same frequency space. This could prove especially advantageous in situations where multiple implanted transmitters need to share the same frequency space and non-interference is imperative (such as a pacemaker coexisting with a pain management implant).

The use of DSSS in implant biotelemetry applications is currently limited. A search of current literature produced two examples. The first is an ingestible capsule used for real-time monitoring of the gastrointestinal tract that discusses a DSSS communication system for one-way data transmission from the implant [20]. However, although this work discussed a DSSS scheme, it did not actually implement this scheme but instead used a narrowband RF approach. The second example is research in progress involving real time transmission of data for an implanted tissue profusion monitor at Oak Ridge National Laboratory [21].

### 1.3.5 Directly Coupled Current Injection

One new interesting and significantly more esoteric implant biotelemetry approach is the technique of directly coupling currents to the tissue [22]. In this technique platinum electrodes are connected to the implant, with one electrode driven by a fixed amount of current and the other electrode serving as a ground. The injected current produces voltage differences when measured at the skin in multiple places by an electromyogram (EMG) amplifier. This technique directly takes advantage of the ionic conduction properties of the tissue. Unlike the approach proposed by this thesis, however, directly coupling current into the tissue has a much higher risk of biological stimulation. Indeed, in the previously cited work between 1 and 3 mA of current were injected into the tissue, which may impact the physiological measurements of interest. Likewise, the relatively high currents prevent this from being a useful low-power technique.

## 1.4 Communication Via Capacitive Coupling for Biotelemetry Applications

Unlike most of the previously described techniques, the work described in this thesis is not a modulated RF technique. Instead, an unmodulated baseband technique using a capacitive link between the implant and a body-mounted receiver was investigated. This technique treats the tissue as a lossy capacitor. The crux of this system is comprised of two transmitting plates capacitivley coupled to the tissue and two like-coupled body-mounted receiving plates, with the tissue in between these sets of plates acting as the dielectric for the capacitor. The basic model for this system is shown in Figure 1.1.

### 1.4.1 Previous Capacitive Coupling Work

While communication via capacitive coupling is certainly not novel, applying this approach to an implant is. Communicating via a capacitive link has been used in many applications, from very short-range high-speed links between microchips [23] to relatively long-range links for communication between rooms for audio/visual equipment [24].

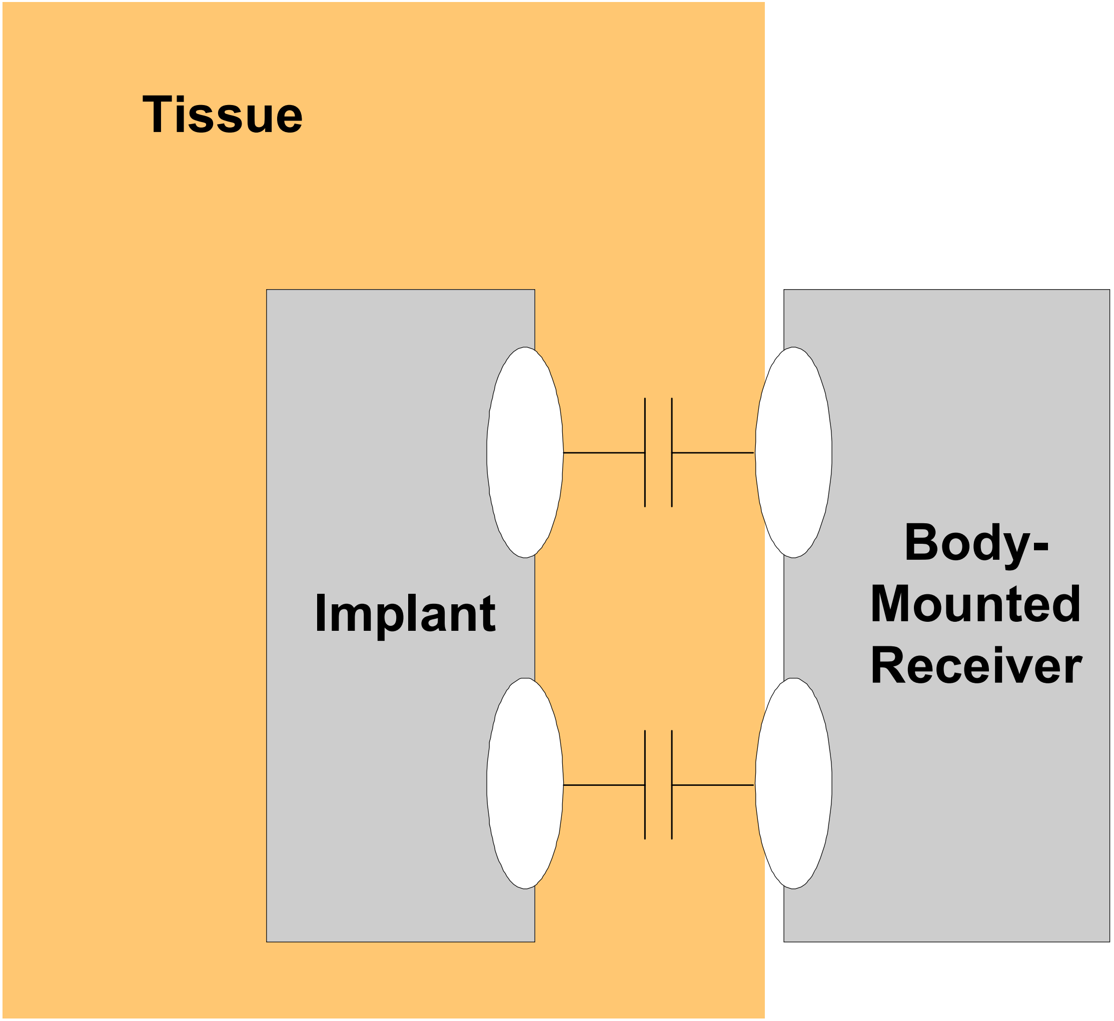


**Figure 1.1: Basic model of an implanted capacitance coupled system**

The closest application of this type of link to an implantable type is in the work done by Thomas Zimmerman in developing personal area networks where the communications backbone is a capacitive link [25]. However, in this case the data is capacitivley coupled to the skin with one electrode, and the skin is used as a conductor for coupling to another skin-mounted electrode. The ground return for this system is through the air. In this thesis both the transmission and ground links are through the tissue. One set of plates is attached to the implant and the other set is in contact with the skin, attached to a body-mounted receiver.

### 1.4.2 Capacitance-Based Approach Compared to Other Techniques

As compared to the other techniques, communication via capacitive coupling offers several distinct advantages and disadvantages. The following subsections briefly describe the advantages and disadvantages associated with this approach.

The first distinct advantage is the use of no off-chip passive components. Unlike RF systems, the only components this technique uses that are external to the chip are the plates used in the capacitive link. However, clever use of the implant package can assuage even this limitation.

The second distinct advantage is in the limited chip area needed to implement this approach. The reason for this comes primarily from the fact that this is an unmodulated baseband technique. All that is effectively needed to implement this approach is an encoder to encode the outgoing data and an analog drive circuit to shape the transmitted waveform. The logic needed for the encoder can be made relatively compact and the analog drive circuitry is essentially just an op-amp, so the size of each of these can be made quite small.

The third distinct advantage to this approach is that it is relatively simple to implement compared to RF approaches. This also comes from the fact that it is an unmodulated baseband technique. The significantly lower complexity compared to RF techniques means that its implementation can be accomplished more quickly and with fewer components.

One of the goals in the design of this technique was to make the power consumption rival that of comparable battery powered RF techniques. Obviously this technique cannot rival the power consumption (or lack thereof) of inductively powered links. The state of the art power consumption for a battery powered RF link is approximately 1µW [5]. One of the goals of this work was to evaluate if a comparable result could be achieved.

The primary limitation in terms of power consumption for a capacitive coupled link is in the slew-rate of the transmitted signal. The equation for the slew-rate of a purely capacitive system is defined as

$$dVc/dt = I/C$$

Here $dVc/dt$ is the slew-rate, $I$ is the current, and $C$ is the capacitance. For lower current and thus lower power slew-rate must therefore be minimized. If the load was purely The practical limit of this is determined by the desired transmission rate of the system; a slower transmission rate minimizing the slew-rate allows for lower power consumption. For this reason this technique is well suited for low-speed unmodulated baseband approaches.

Communication via capacitive coupling through tissue places two major limitations on implant design. The first of these is in the effective range of communication. Because of the nature of the link, the communication distance via capacitive coupling is limited to the distance from the implant to the skin. In other words, by its very nature the receiver must be mounted to the body. Also, depending on plate size and the size of the patient, the implant depth is limited. In the worst-case scenario of a morbidly obese person, the maximum implant depth could actually not extend to the desired organ.

The first of these range limitations can be overcome through the use of a body-mounted repeater using an RF link. In this case the body-mounted receiver would be nothing more than a relay station to some other station where any necessary data analysis and storage would take place. The second of these limitations can be overcome by adequate plate sizing for particularly large patients.

The second major limitation communicating via capacitive coupling places on implant design is on the size of the implant. The effective range for communicating via capacitive coupling is directly influenced by the plate size. As a function of this, the primary limitation the capacitive coupling technique places on implant size is the size of the plates used. Empirical testing results of plate sizes are presented in Chapter 2, but generally speaking implant depth for plates much smaller than 1cm diameter would be severely limited. Thus for a very small implant this technique is not a valid solution.

However, for a medium sized implant creative use of the implant casing can easily provide relatively large plate sizes.

## 1.5 Justification for Research

As demonstrated, a number of different techniques have been reported. This thesis reports the first application of a capacitance based communication technique to implanted sensors. There is no universal communication system that works for every type of implant. A communication system designed for use by a relatively large battery powered implant would not be appropriate for a small externally powered implant. In presenting the work done for this thesis the intent is to provide another option to the implant designer when choosing a communication technique. This communication technique was designed for a medium sized (on the order of the size of a pacemaker or smaller), battery powered, and short-range implant. While other techniques may offer other advantages, what this technique offers is a low number of off-chip passive components, low on-chip silicon area requirements, and relatively simple implementation.

## 1.6 Scope of Thesis

This thesis describes the development of a capacitance-based biotelemetry system for implantable applications. This system consists of two distinct parts: the implanted transmitter and the external receiver. The prototype transmitter is based on a custom

ASIC fabricated using the AMI 1.5µ process available through MOSIS. This ASIC encodes and transmits predetermined data packets by driving two electrodes in a slew-controlled manner, all contained within a biocompatible material. The receiver consists of charge-sensitive front end using a discriminator to distinguish individual bits. A FPGA decodes the transmitted data and relays it to a PC-based LABVIEW interface.

In Chapter 1, a review of many of the primary design challenges associated with data transmission through tissue are presented. Also, an overview of the primary wireless communication systems employed in modern implanted systems is provided. Finally, an alternative technique based on capacitive coupling is introduced and compared to the previously discussed techniques.

Chapter 2 details the initial sensor design and evaluation. The first approach to characterize the medium is discussed. This includes simulation and experimental results from this method. A second method for simulating the medium is then developed. From this method, optimum electrode configurations are determined. A discussion of the various electrode configurations is presented along with experimental data to conclude why the particular electrode configuration was chosen.

Chapter 3 presents the system design and implementation. The chapter begins with an overview of the complete system. Next, each of the individual blocks in the transmitter and receiver are discussed in detail. This includes a justification for why a particular design for each block was chosen and in some cases presents alternate designs that were also implemented. The design, simulation, and implementation of the transmitter as a custom ASIC is also presented. Finally, the design and implementation of the receiver is given.

Chapter 4 details the testing of the system. Three different test setups were created to test the system. Each of the three setups are first described, then the various test results such as successful data link range, slew-control results, and power consumption from each setup are explored.

Chapter 5 presents the conclusions of the testing and discusses improvements that could be made.

# Chapter 2 MODELING CAPACITIVE COUPLED BIOTELEMETRY

## 2.1 Modeling Human Tissue

Successful design of any implantable communication system requires the use of a suitable model for the human body. These models are classified into two types: computer-based models and physical experimental models. One major problem with building accurate system models for either of these types comes not in the modeling of single tissue types, but in how to model complex structures composed of multiple tissues with significantly different electrical properties. Although solutions to this problem exist for both model types, they are not always appropriate for early prototyping.

For computer-based models, highly detailed human body models and simulator packages such as the XFDTD High Fidelity Body Mesh from Remcom are available. These packages have emerged primarily in response to the need for accurate prediction of the interaction of electromagnetic fields with biological tissues specifically for evaluating compliancy to federal FCC guidelines [26]. However, after evaluation, the Remcom package was not used directly in this research for two reasons. First, the XFDTD package was specifically designed for RF simulations with waveforms of 500 MHz or more, and not low frequency baseband signals as are employed in this research. Consequently, simulation times were extremely lengthy (on the order of a day or more) and frequently produced failed simulations. The second reason was that software with this level of detail is much more useful for fine-tuning working systems than for evaluating prototype systems.

Advanced physical models suffer from the same complexity problem. Obviously the most accurate solution is to use either a live animal or a human corpse, but neither of these options is suitable for quick prototyping where repeated implantation over fairly long periods of time is desired. The common solution is to use a compound that fairly accurately imitates the electrical properties of tissue for a given frequency (otherwise known as a phantom tissue). Phantom tissues can be broken down into two different types: solid cast forms and liquids. The chemical composition for solids with the electrical characteristics of a variety of tissue types at various frequencies has been documented [26]. These can be layered to create a more detailed simulation. Very detailed body forms made up of these materials are commercially available [27]. Liquid-based phantom tissues are generally saline-based solutions of varying salt concentrations. Although it is inherently harder to layer liquid-based phantoms, they are easier to produce and being liquids naturally conform to the device under test.

For this research a single-concentration saline-based solution was used as the primary test setup. Although this only offers a first order approximation at best to modeling the human body, it is suitable for prototyping functionality rather than making subtle adjustments in performance. Saline was chosen under the assumption that blood dominated tissues will comprise most of the tissues surrounding the implant. Furthermore, it is assumed that these tissues provide the worst-case scenario for transmission, as they are the lossiest. Both saline and blood have dielectric properties that reflect their ionic conduction nature. Adjusting the salt concentration per mass in saline to approximately 0.8% produces a relative conductivity for low frequencies that is the same as blood [26].

The next section describes the equivalent circuit model extracted from an initial experimental test setup. In the third section an improved experimental setup is presented. The experimental setup used addresses some of the issues not addressed by the first setup, and helps lay the foundation of the problems the custom ASIC will need to solve.

## 2.2 Establishing an Equivalent Electrical Model

With saline chosen as the experimental medium one of the first steps that was performed was to establish an equivalent electrical model. However, what was desired was not a model based on the frequency response of the saline but on the time domain response. Characterization in the time domain rather than the frequency domain is natural when dealing with an unmodulated baseband communication scheme. This is because with an unmodulated baseband approach the designer is principally interested in is the step response of the system, not the frequency response. Thus the goal in this case was to establish an electrical model with a step response similar to that of the medium, not necessarily one with a similar frequency response.

The key assumption in building this model is that saline behaves like a capacitor with extreme dielectric absorption. Thus, the model chosen for the saline was the Dow model of a dielectric absorptive capacitor. The method chosen to verify this model was to first find the experimental step response of the system. Then, via numerical analysis, the parameters of a Dow model of a dielectric absorptive capacitor were fitted to produce a matching step response [28].

### 2.2.1 Time Domain Characterization Test Setup

The initial test setup consisted of two copper electrodes pressed against a phosphate buffered saline-based tissue phantom as shown in Figure 2.1. Each electrode was 15mm in diameter and the separation distance was approximately 20mm. One electrode was connected to a pulse generator and the other to a charge-sensitive preamplifier.

### 2.2.2 Step Response

The step response for the experimental setup is shown in Figure 2.2, where the upper trace is the output of the charge sensitive preamplifier and the lower trace is the input from the pulse generator. As expected, this response is similar to that of a capacitor with a large amount of dielectric absorption.

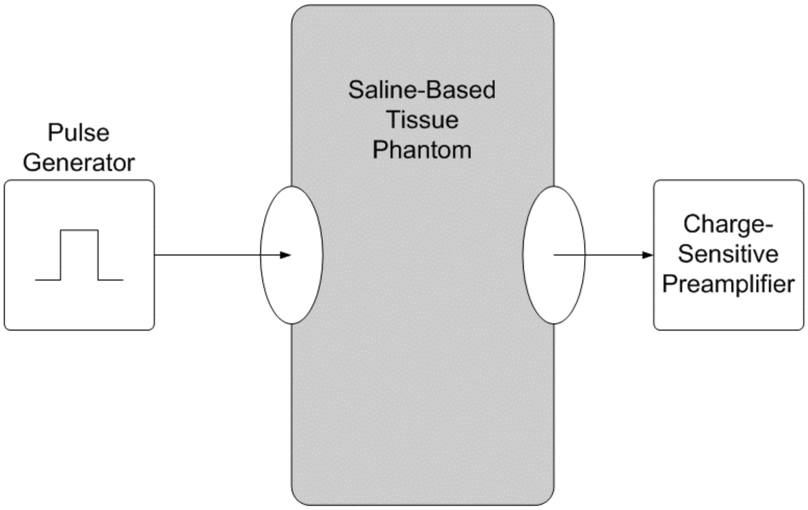


**Figure 2.1: Schematic for time domain characterization test setup**

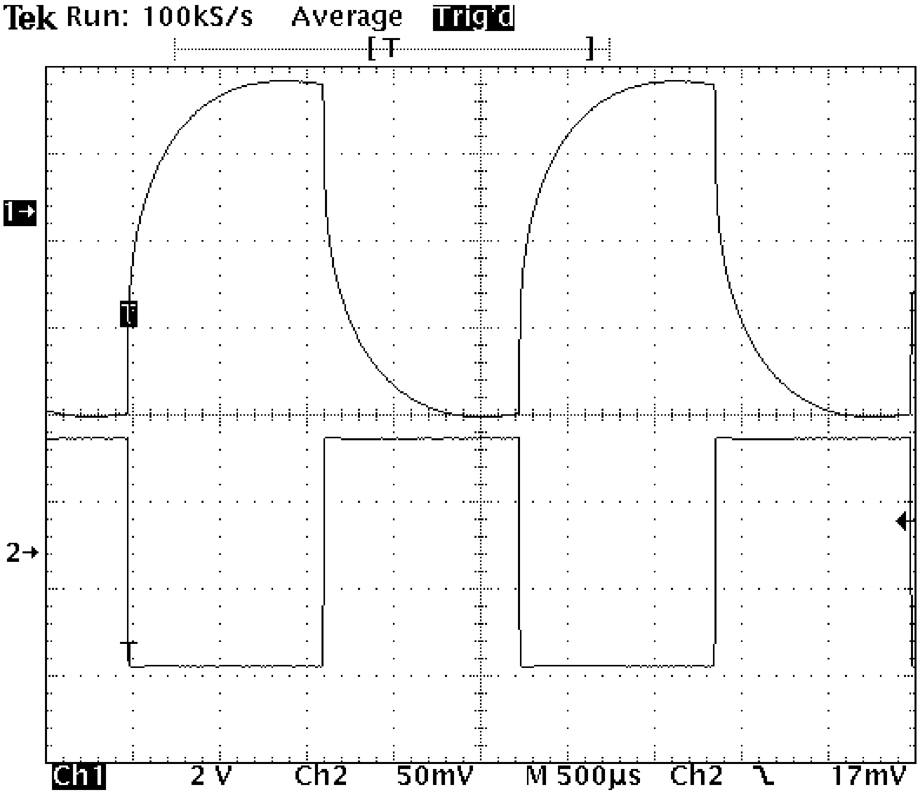


**Figure 2.2: Step response of the system**

### 2.2.3 Equivalent Electrical Model

Based on the experimental step response, the electrical model given in Figure 2.3 was developed, where the saline is modeled as a third-order absorptive capacitor. The values for the feedback capacitor and resistor of the charge sensitive preamplifier are 2.3pF and 978.3MΩ respectively. Numerical analysis was performed to fit the values of C1, C2, C3, R2, and R3 to produce an approximately matching step response. These values were 0.208pF, 1.477pF, and 0.861pF for C1, C2, and C3. Approximate fits for R2 and R3 were 1.156MΩ and 95.6MΩ. Obviously, adjusting any parameters in the experimental test setup such as plate size or transmission distance would change the fitted values, but this does verify that indeed saline (and thus human tissue) can be modeled as a extremely dielectric absorptive capacitor.

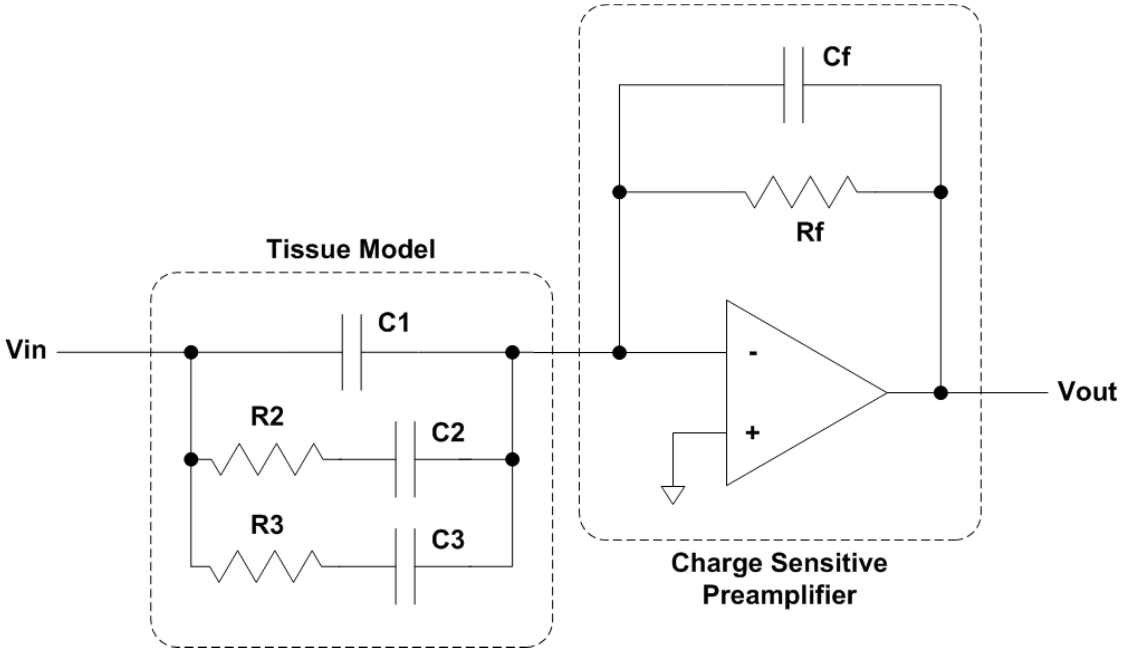


**Figure 2.3: Equivalent electrical model derived from the experimental setup**

## 2.3 Establishing a More Accurate Experimental Setup

While the experimental test setup detailed in the previous section is useful for establishing an equivalent electrical model for saline, there are two issues it fails to address. The first of these is that with the first test setup there is an explicit hardwired return path between the pulse generator and the charge sensitive preamplifier. Ideally for a real-world implant there is no transcutaneous ground wire connecting the ground of the implant with that of the receiver. The second issue is that for an actual implant the tissue does not just act as a wall in front of the implant but completely surrounds the implant.

The second test setup addressed these points in two ways. First, a battery-powered oscillator replaced the pulse generator. To establish a return path a second electrode was added to both the oscillator and receiver. One of the test oscillators used is shown in Figure 2.4. Second, a tank was constructed to completely submerge the oscillator in phosphate buffered saline. This tank volume was substantially larger than the glove used in the previous experiment, measuring slightly over 1 $ft^3$, to better approximate a human torso. Also, the oscillator could be positioned accurately at fixed distances from the receiving electrodes in the tank through the use of a marked track. This tank is shown in Figure 2.5.

### 2.3.1 Summary of Test Results

Besides just offering a proof-of-concept, the new setup allowed for the testing of several new parameters establishing better empiric characterization of the transmission method.

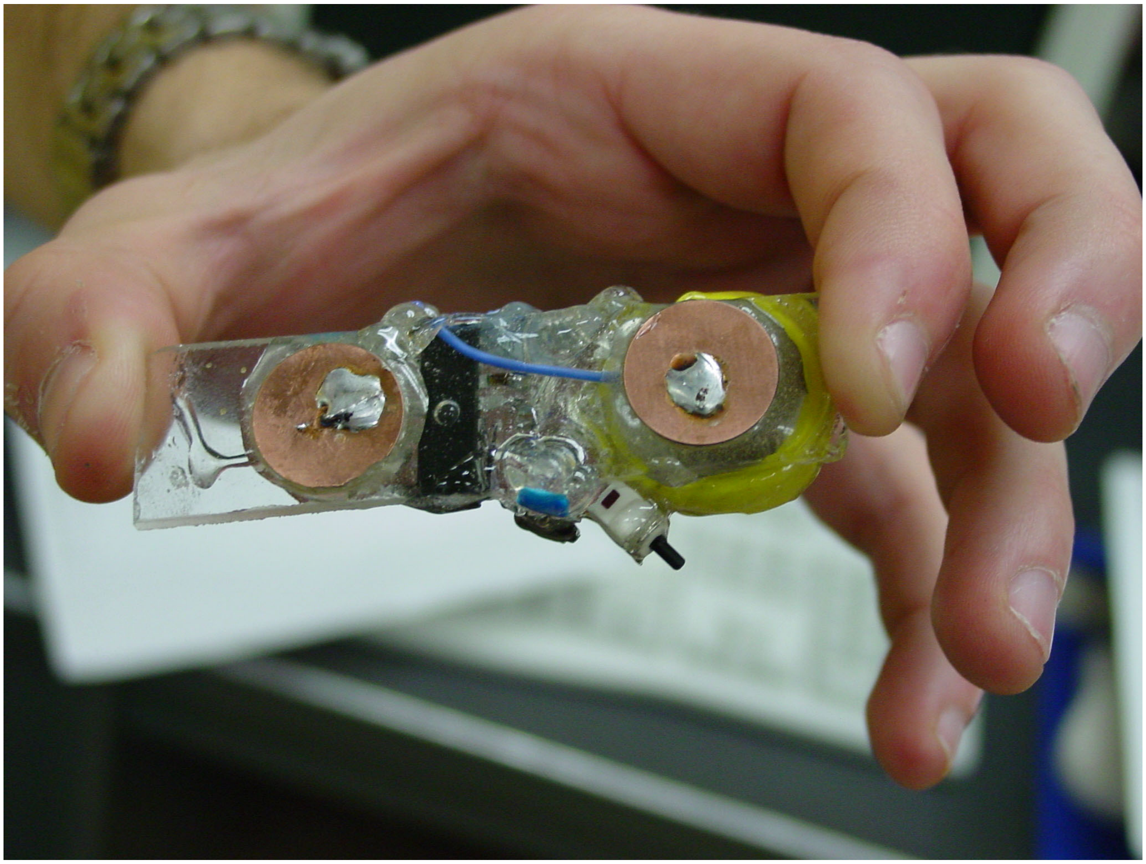


**Figure 2.4: One of the battery-powered oscillators used to simulate an implant**

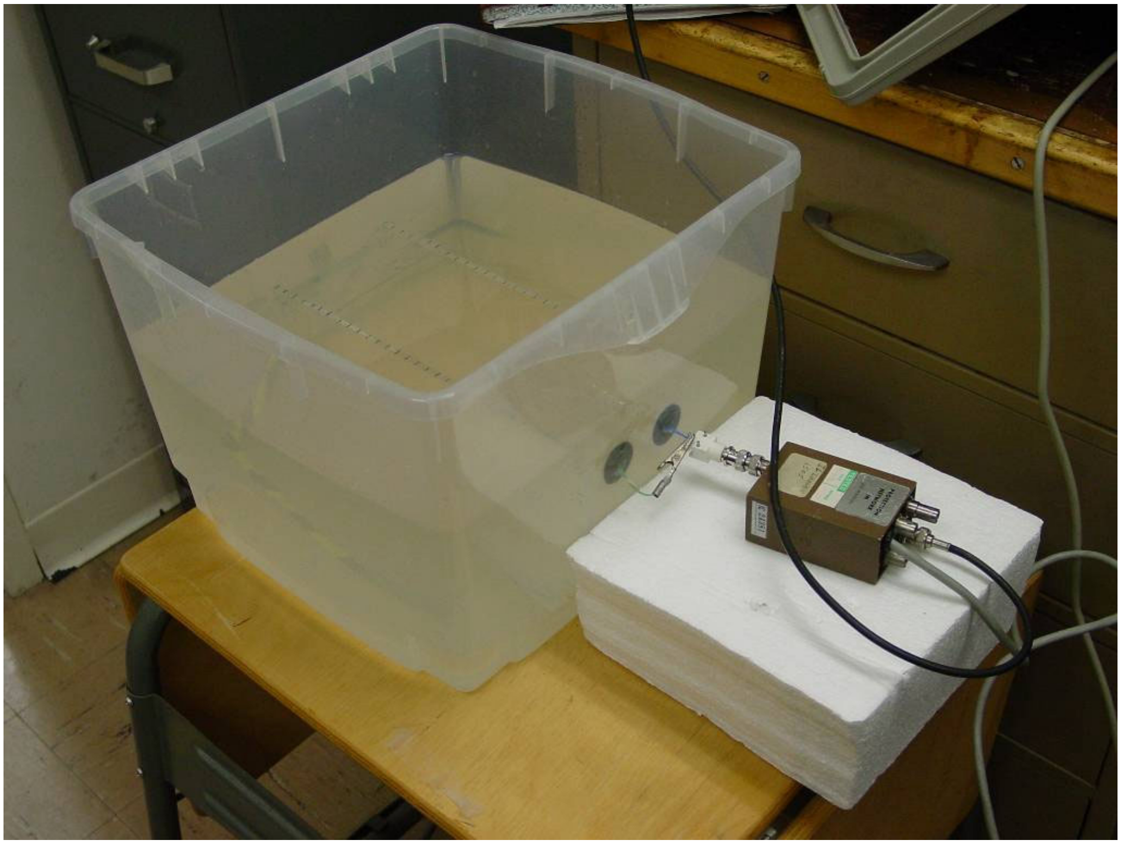


**Figure 2.5:  Tank used in the improved experimental test setup**

The first was the distance of the transmitting electrodes from the receiving electrodes. The second was the size of the transmitting and receiving electrodes. The third was the electrode isolation. To test these parameters the output of the charge sensitive preamplifier was connected to a spectrum analyzer. The received power at the fundamental harmonic was measured while each parameter was adjusted. As expected, both plate size and separation directly affects the received power. Figure 2.6 clearly shows the effects of each of these: First, doubling the receiving electrode plate diameter (or quadrupling the area) produces a 10 times increase in received signal power. Second, received signal power drops approximately with the square of the distance.

Changing the electrode isolation also revealed interesting results. Hypothetically each electrode can be either insulated from the saline or directly electrically connected to the saline. However, certain configurations are more desirable than others from a biological compatibility standpoint, namely those configurations where both implant electrodes are insulated. Also, certain configurations change the nature of the transmission. For instance, directly coupling the transmitting electrode to the saline is no longer communication via capacitive coupling but instead directly coupled current injection, as described briefly in Chapter 1. It was found that with the receiving ground electrode directly connected and the receiving electrode insulated, reliable transmission could be seen with both of the implants electrodes insulated. Exposing the transmitter's ground electrode increased the received power by 6dBm. Ideally all electrodes should be insulated, however, but in this configuration the received signal strength was found to be significantly lower than any other configuration.

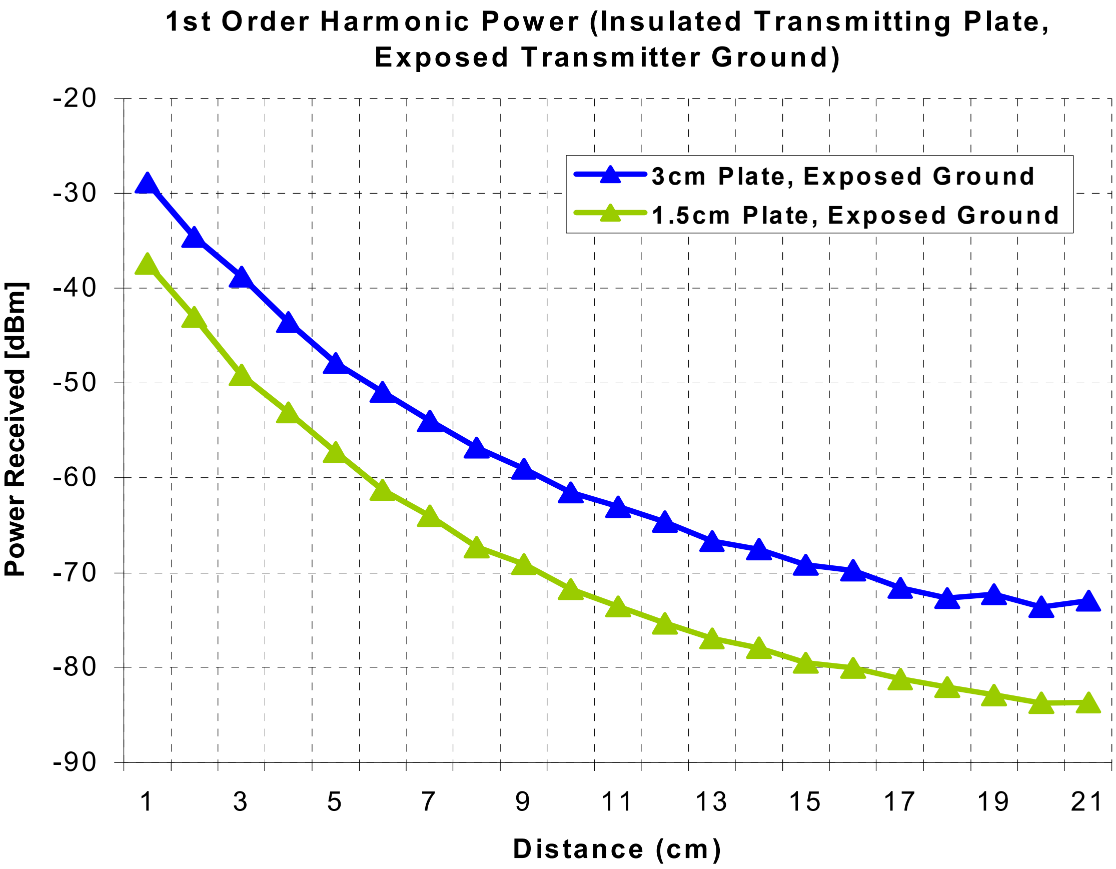


**Figure 2.6: 1$^{st}$ Order harmonic power versus distance and plate size**

### 2.3.2 Test Results and Conclusions

Although this test setup contributed valuable data, the key contribution came in the form of a better understanding of the test medium, not in exact measurements. The reason for this lies in the oscillators used. It was discovered that changing the type of oscillator changed both the shape and amplitude of the received waveform. The reason for this is twofold: First, the output stages of the oscillators are not designed to drive a capacitor with a large dielectric absorption. Second, the shape of the waveform each oscillator generates is different, especially with regard to ringing and slew-rate, and this causes a large change at the output of the charge sensitive preamplifier. These factors highlight the need for an output stage designed to both drive a capacitor with a high dielectric absorption and offer some degree of waveform shaping.

# Chapter 3 System Design and Implementation

## 3.1 System Summary - Overview

The capacitance-based telemetry system consists of two distinct parts: an implemented transmitter, and a body-mounted receiver (see Figure 3.1).

The transmitter has three basic functions: packetizing digital sensory data, encoding these packets using a DC-balanced method, and transmitting the data via an analog drive circuit incorporating pulse shaping to optimize transmission characteristics. This circuit drives a plate that is electrically insulated from the tissue, thus ensuring no direct DC coupling to the tissue. A ground-return electrode is also attached that is likewise insulated from the tissue.

The receiver module also has three distinct blocks: a charge sensitive pre-amplifier front-end, discriminator, and packet decoder. The pre-amplifier is connected to an electrode that, like the transmitting electrode, is electrically insulated from the tissue. An insulated ground-return electrode is also present. The pre-amplifier produces a pulse-train output of that is feed into a discriminator circuit that acts as a 1-bit A/D converter transforming the received analog waveforms into digital bits. The converted digital signal is then input to a packet decoder block. The decoding process involves several functions: basic packet detection, synchronization, error detection and correction, and removing the encapsulated data.

This chapter presents the design of each of the individual blocks within the transmitter and receiver.

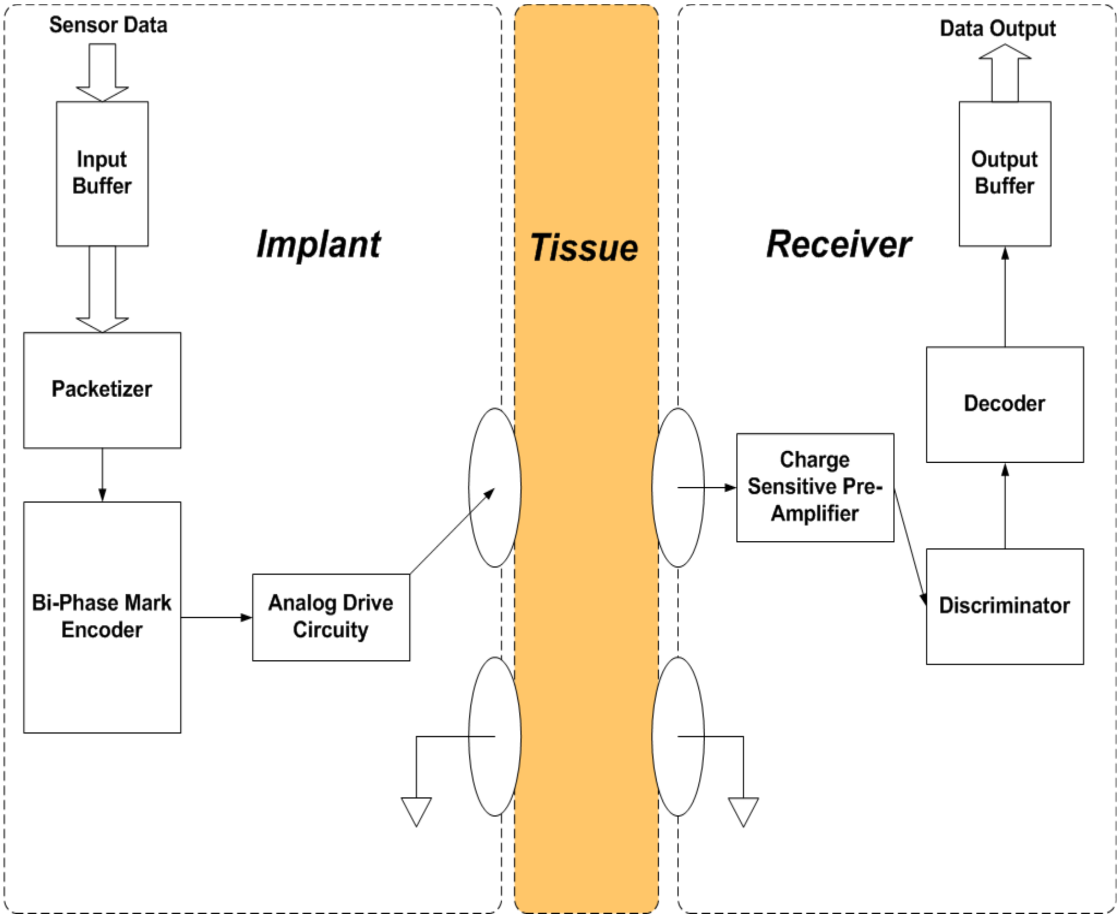


**Figure 3.1: Block diagram for complete system**

Each function is presented in the order data is transmitted and received, following the data through the system as it would be transmitted from the implant to the external body-mounted receiver. The chapter concludes with a presentation of the physical implementation of the transmitter and receiver.

## 3.2 Implant Transmitter

As was stated previously, the basic function of the transmitter is to take data from the implant, encapsulate it in packets, encode the packets in a scheme suitable for transmission through tissue, and transmit the encoded packets with each pulse shaped to maximize transmission efficiency. For testing purposes, to eliminate the need for incorporating a sensor and associated interfacing electronics, "dummy" data is generated on chip. This data is a non-random number pattern, allowing the test receiver to accurately determine when packets have been dropped or incorrectly transmitted. A basic block diagram of the transmitter is shown in Figure 3.2. The following subsections detail the block that generates the dummy data packets, and then describes the format chosen for the data packets. Example waveforms of the packets are also shown, showing in detail the structure of the packets and the generated packet train as a whole. The chosen encoding scheme and its design is detailed next, followed by a discussion of alternative encoding schemes and their implementations. Finally two different analog drive circuits are presented. The basic functionality of each is discussed, along with simulation results and associated integrated circuit layout details.

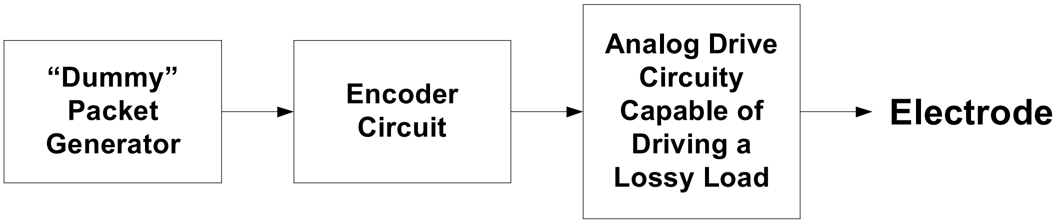


**Figure 3.2: Implant transmitter block diagram**

### 3.2.1 Dummy Packet Generator

The dummy packet generator's function is to generate data encapsulated within packets as though it was coming from a real external sensor. However, unlike real data from an external sensor, this data must be both regular and periodic allowing straightforward verification of the data link. Thus for the purposes of the transmitter the dummy packet generator has three functions: generating a series of predictable data so that missed or erroneous packets can easily be detected, encapsulation of the data into packets of the chosen format, and periodic generation of these packets. The packet transmission interval must be sufficiently long enough to prevent accidental pileup of packets and short enough to enable straightforward observation with an oscilloscope or logic analyzer. The time between packets was chosen to be 224 cycles of the system clock, or roughly 788 packets per second. This low data rate was chosen to simplify testing, and does not represent a limitation of the system. Also, a "sleep" signal is generated during the dead-time to shutdown other sections of the chip.

The dummy packet generator was designed in VHSIC Hardware Description Language (VHDL) using Altera's Quartus II software. Initial testing was done using an off-the-shelf test board incorporating an Altera Stratix FPGA. This test board was used in both the transmitter and receiver to minimize the required effort and to allow easier verification of both functional blocks. The dummy packet generator is a simple state machine consisting of four states. Each state represents a section of the packet currently being generated: header, footer, data block, and sleep (off state). Because each section has a fixed length, most of the activity is simply counting clock cycles until it is time to switch to the next state. The VHDL code for the dummy packet generator is given in Appendix 1. After verification of the design using the FPGA board, the VHDL based design was ported to an auto-place, auto-route software package (Timberwolf [30]) and an ASIC layout was produced using a library of standard CMOS primitives (this layout is shown in Figure 3.3).

### 3.2.2 Packet Format

The packet format was designed first and foremost to enable ease of decoding. For this reason a large header and footer were used. The header is composed of thirty-two "1" bits and the footer is composed of twenty-four "1" bits. A start bit and a stop bit were also used to separate the header and footer portions of the packet from the data. Both start and stop bits are 0s. The data block of the packet is composed of one 4-bit counter repeated twice, for a total of eight data bits. Figure 3.4 shows the Quartus II simulation results the of the dummy packet generator block. The first waveform is a single packet. The "position" vector in the waveform gives the current state of the state machine. The data vector is 4-bit counter that is repeated twice in the packet. The second waveform shows a close-up of the actual data in a packet.

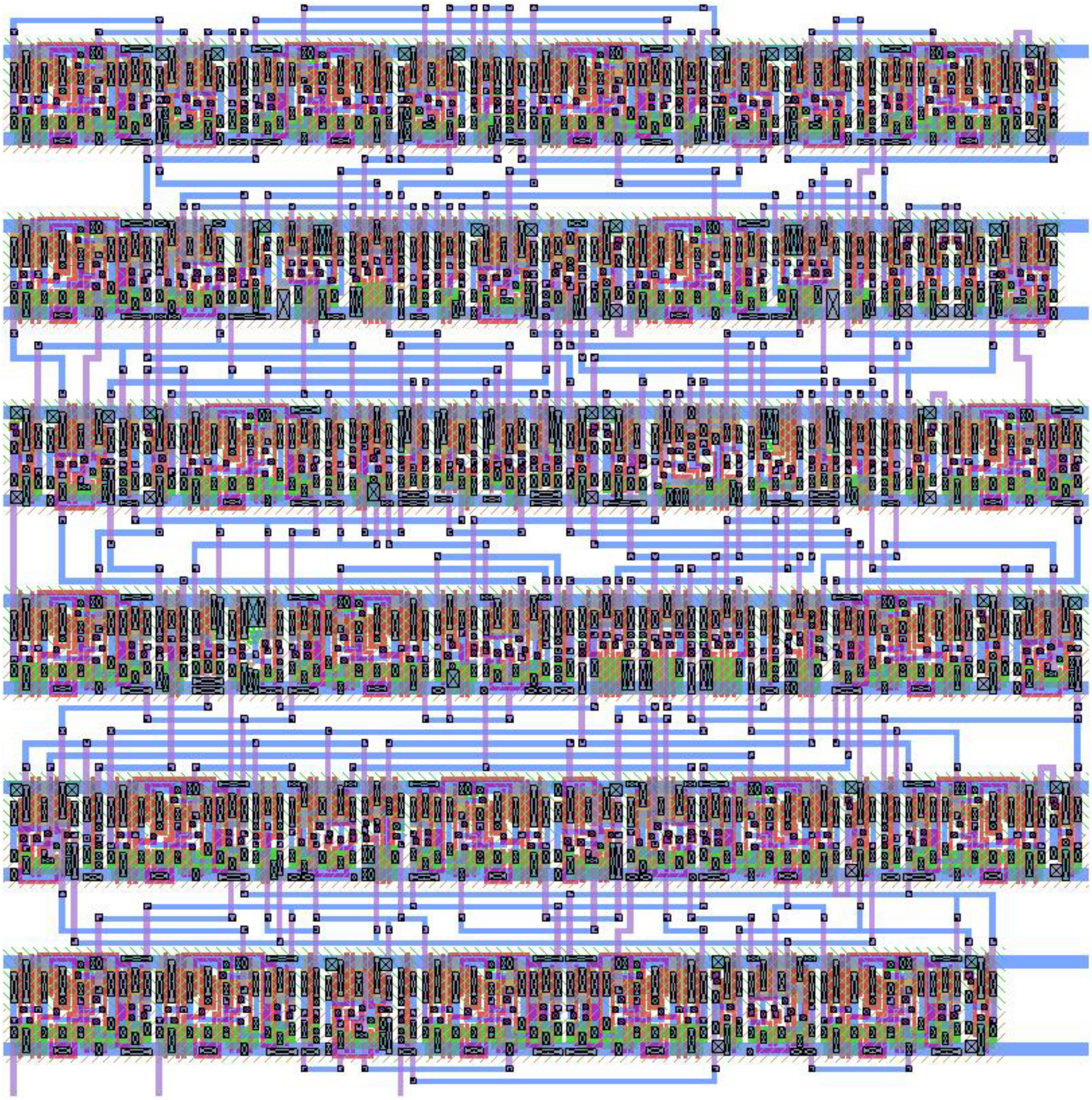
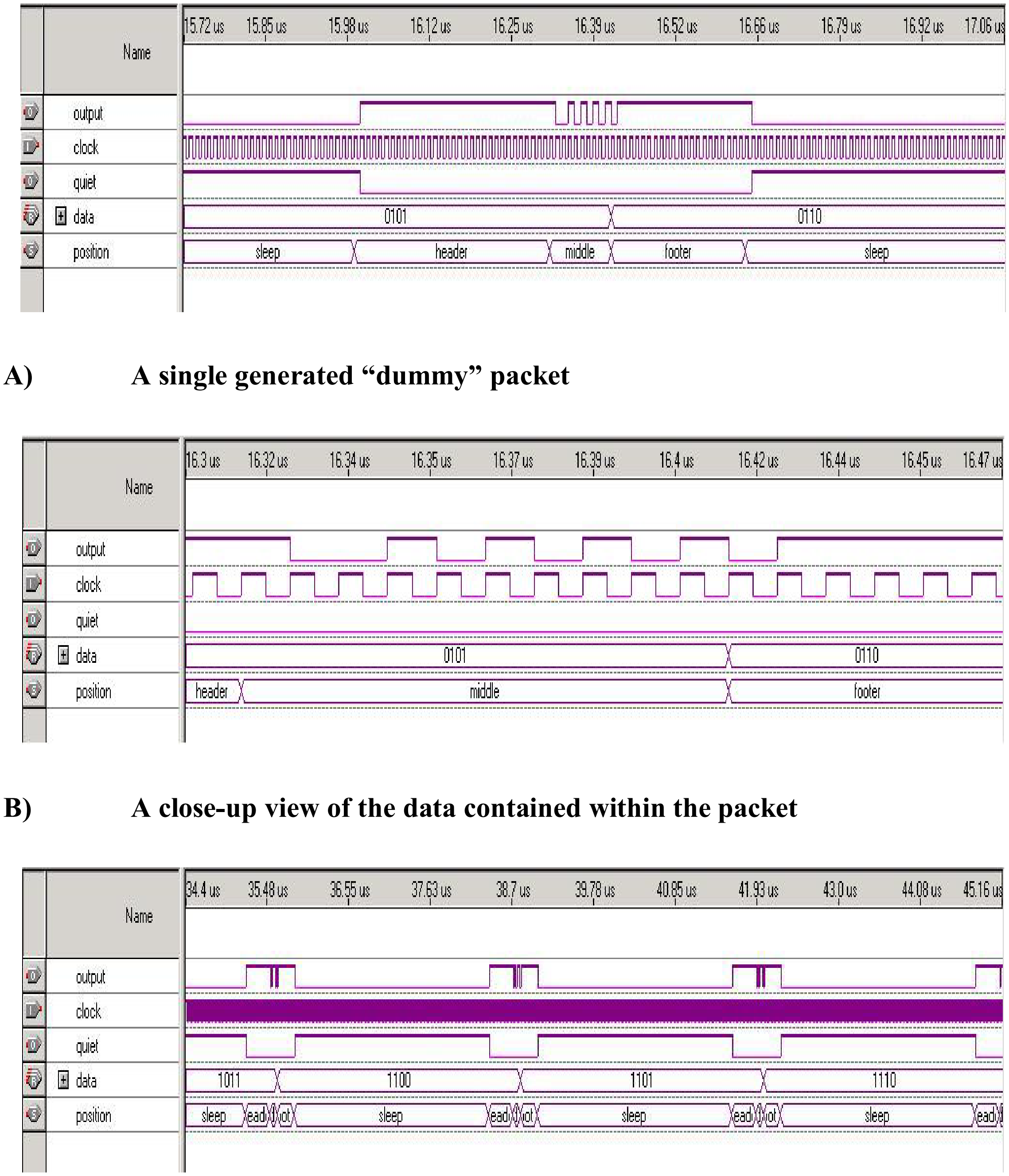


**Figure 3.3: Layout of the dummy packet generator block**



**A) A single generated "dummy" packet**

**B) A close-up view of the data contained within the packet**

**C) Multiple packets being generated**

**Figure 3.4: Simulated waveforms of "dummy" packets**

In this waveform the count value (“0101” in this case) can clearly be seen, repeated twice and surrounded by the ‘0’ start and stop bits. The third waveform shows a series of packets being generated at regular intervals.

### 3.2.3 Bi-Phase Mark Encoding Scheme

In choosing an encoding scheme three main criteria should be met [31]:

1) For biocompatibility reasons the DC component of the bit stream should be approximately zero for all possible patterns.

2) For ease of decoding the encoding scheme should be self-clocking.

3) Due to the observed differential nature of the channel for certain transmitter plate arrangements, an event should cause a transition in the input signal to create a pulse at the output. At the output of the receiver the rising-edge of an input square wave causes a positive pulse and the falling-edge causes a negative pulse.

Of the popular encoding schemes, Manchester encoding, bi-phase mark encoding, and bi-phase space encoding were chosen due to their self-clocking nature and zero DC average level. Bi-phase space is identical to bi-phase mark except ‘0’ is encoded as a mid-level bit rather than ‘1’.

Bi-phase level is encoded to represent a ‘1’ as the first half-bit high and the second half-bit low. Conversely, a ‘0’ is represented by the first half-bit low and the second half-bit high, guaranteeing a level change in the middle of each bit. For bi-phase mark there is always a level change on the beginning of each bit (i.e. the rising edge of the clock signal). A ‘1’ is encoded as a level change in the middle of the bit (or on the falling edge of the clock) and a ‘0’ is encoded by no level change. As stated earlier, bi-phase space is the inverse of this. For each of these schemes is NRZ style, that is a high is +V and a low is –V.

Each of these encoding methods meets the first two requirements. They each have one guaranteed level change within each bit allowing extraction of a clock signal, and the average DC level for each is zero. However, the bi-phase mark and bi-phase space schemes are more suitable for this application because of the third criterion. In a bi-phase level scheme a ‘1’ is dependant on the first half of the bit being high and the second half being low, and vice-versa for a ‘0’. However bi-phase mark and bi-phase space are purely dependant on transitions only and not on the actual level, making either of them more suitable for this application.

One advantage to using either bi-phase mark or bi-phase space is that there is some inherent error detection without having to use additional bits. Because there must be a transition on the beginning of every bit, any bit sent without that transition is considered an error. Because the channel transmits rising edges as a positive pulse and falling edges as a negative pulse, for any encoding scheme with only two levels there can never be two consecutive positive pulses or two consecutive negative pulses. Thus any bit containing this pattern should be considered an error.

For these reasons bi-phase mark was chosen as the primary encoding scheme. Although a VHDL encoder block was developed at the same time as the dummy packet generator (as it was needed for verification of the functionality of the receiver), it was not used in the transmitter implementation. Instead the encoder blocks for use on the chip were designed and laid out manually using standard logic. The logic used for the bi-phase mark encoder block in shown in Figure 3.5.

The layout for the bi-phase mark encoder block was designed in MAGIC using blocks from the MSU standard cell library, with the exception of the dual edge-triggered flip-flop, a previously designed ORNL custom cell. The bi-phase mark encoder block was extracted and simulated directly from the layout using IRSIM. The layout is shown in Figure 3.6 and the resulting simulation in Figure 3.7.

## 3.2.4 Alternative Encoding Schemes

Two other schemes were also designed and implemented in addition to the bi-phase mark approach: bi-phase level (Manchester encoding) and delay modulation (Miller encoding). Like the bi-phase mark implementation, the logic and layouts for both of these were implemented by hand design rather than via VHDL and an extracted layout.

### 3.2.4.1 Manchester Encoding

Manchester encoding is straightforward compared to either bi-phase mark or Miller encoding. In Manchester encoding a ‘1’ is encoded as a half-bit wide pulse in the first half of the bit and a ‘0’ as a pulse in the second half of the bit. Because there is always a transition in the middle the clock can be easily extracted. The main advantage Manchester encoding has over bi-phase mark is the simplicity of implementation, and this is why it was included in the ASIC design. Even though the bi-phase mark encoder block met the system requirements and simulated correctly, including

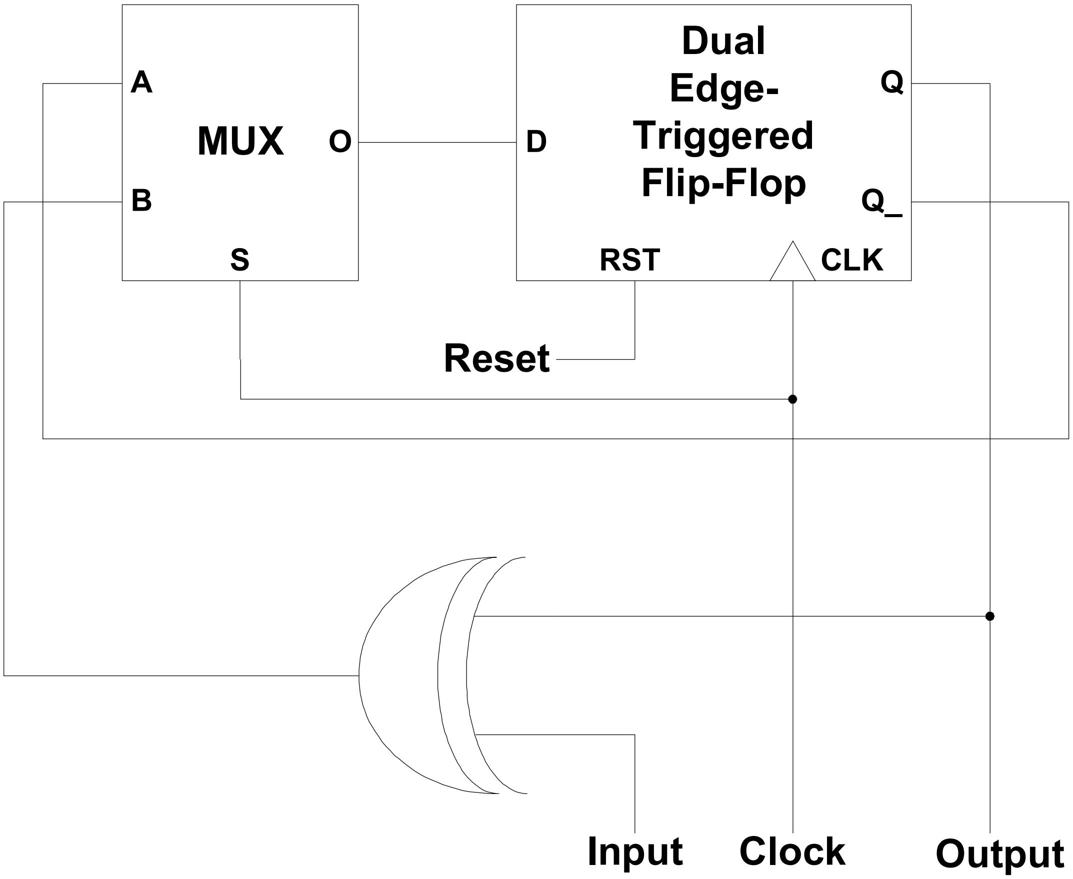


**Figure 3.5: Logic for the bi-phase mark encoder block**

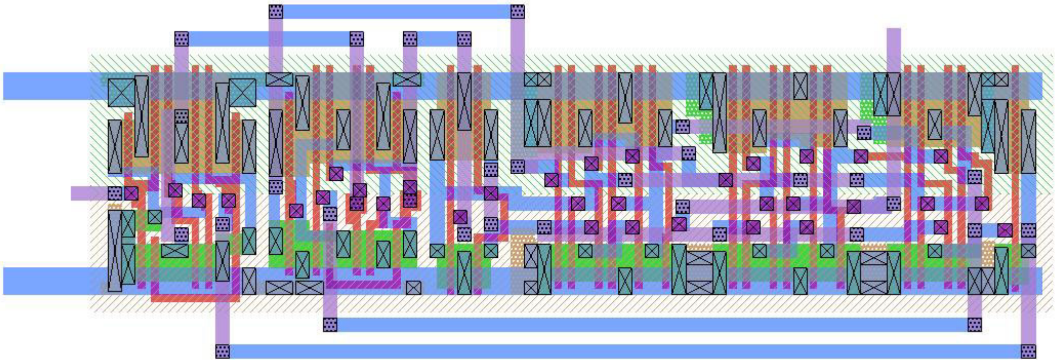


Figure 3.6: ASIC layout of the bi-phase mark encoder block

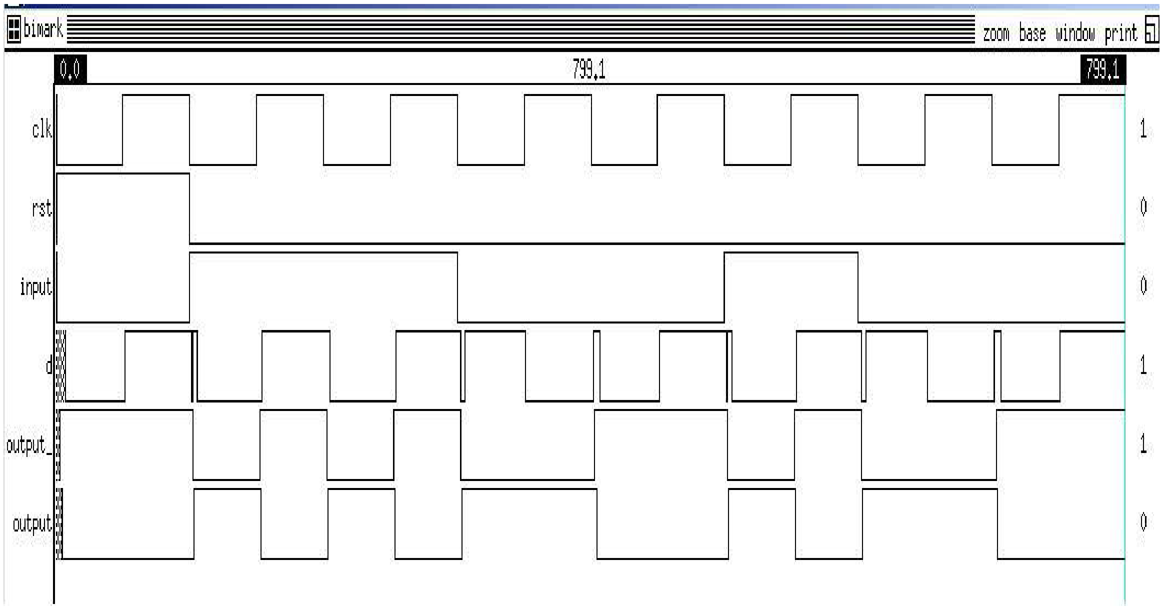


Figure 3.7: Simulation of the bi-phase mark encoder block

Manchester encoding was an easy way to add a redundant encoding type with a reasonably high level of assurance of functionality. Figure 3.8 shows the logic for Manchester encoding. Figure 3.9 shows the ASIC layout for the block. This approach can be implemented using few gates, as can be expected from the comparatively low complexity of the logic. Thus, the layout is the simplest of the encoding schemes included on the chip. Figure 3.10 shows the simulation results of this block.

#### 3.2.4.2 Miller Encoding

Unlike Manchester encoding Miller encoding is a fairly complex encoding scheme. In this scheme a ‘1’ is represented by a transition in the middle of the bit, and a ‘0’ by no transition. However, if the next bit is a ‘0’ and the current bit is also a ‘0’ then a transition is made at the end of the current bit. The increased functional complexity results in more complex logic for implementation.

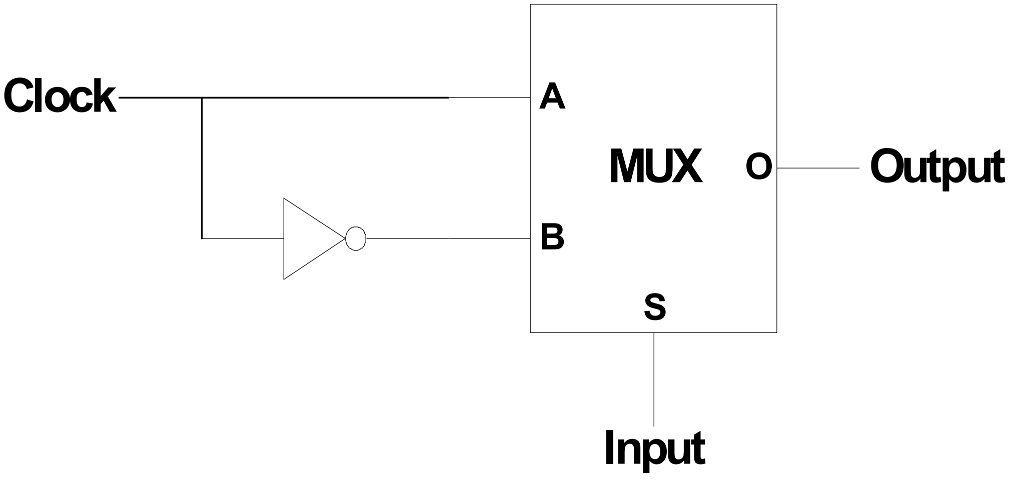


**Figure 3.8: Logic for the Manchester encoder block**

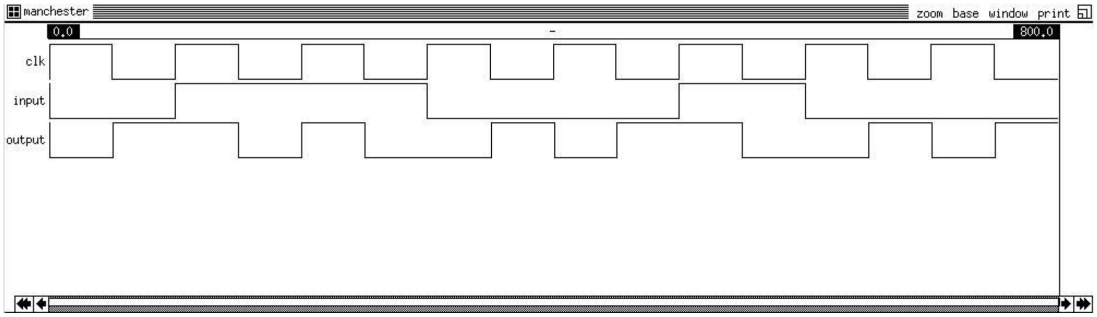


**Figure 3.9: Simulation of the Manchester encoder block**

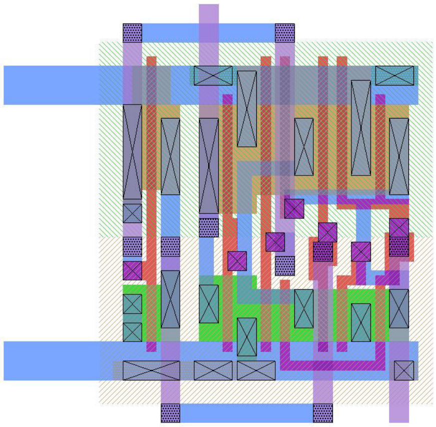


**Figure 3.10: Layout of the Manchester encoder block**

Figure 3.11 shows the logic used to implement the Miller encoding block. One technique was the use of a delay block to temporarily hold the previous state. Although this may not be the best choice for use in a robust block, it was used here because this block was intended for testing purposes and only as an extreme worst-case backup. Figure 3.12 shows the layout used for the Miller encoder block. Figure 3.13 shows the simulation results of this block.

### 3.2.5 Analog Slew Control Circuits

The main function of the drive circuitry is to drive a range of capacitive loads with the load having minimal affect on the shape of the driven waveform. Also, some degree of shaping of the waveform was desired to determine the optimum shape for maximum transmission distance. This shaping consists of entirely of slew-rate control. A tight slew-rate control is important due to the nature of the media. One result of early investigations was that there is a direct relation between the slew rate of the transmitted signal and the shape and level of the received signal. Therefore being able to set the slew-rate for optimum transmission is highly desired.

The center of virtually every slew-rate limiting circuit is controlling a fixed drive current into a capacitive load. Slew-rate is defined as the maximum change in voltage per amount of time, i.e. the maximum slope of the output signal for a given load. The current through a capacitor is defined as

$$Ic(t) = C\, dVc/dt.$$

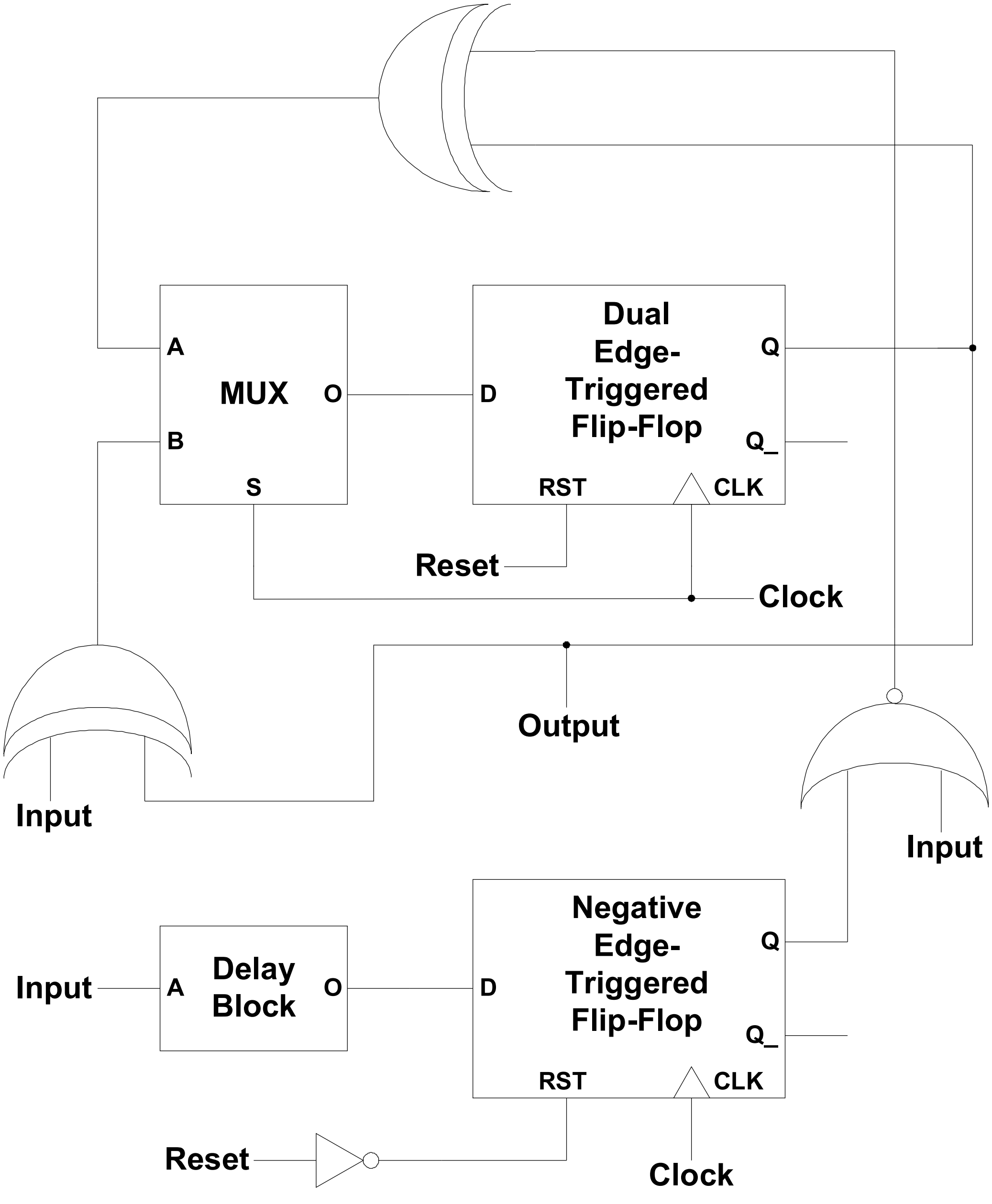


**Figure 3.11: Logic for the Miller encoder block**

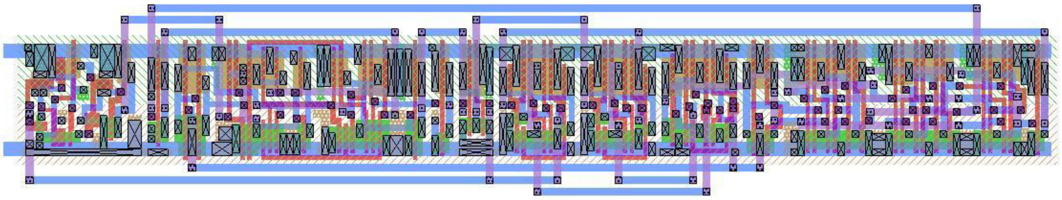


**Figure 3.12: Layout of the Miller encoder block**

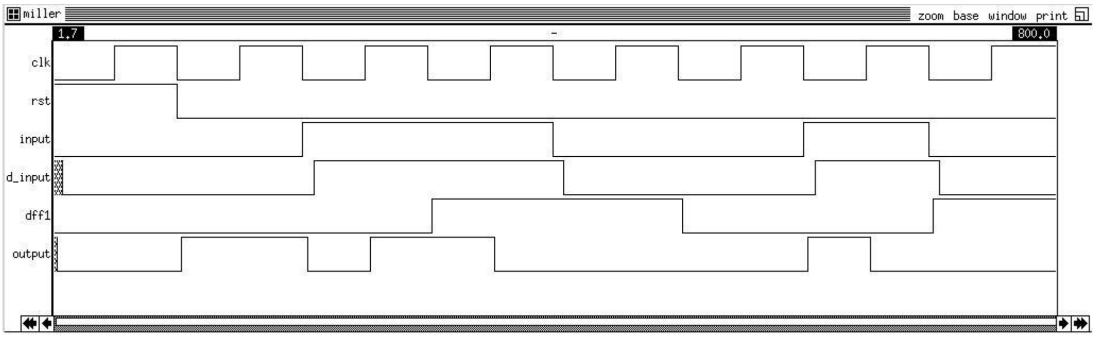


**Figure 3.13: Simulation of the Miller encoder block**

Thus for a capacitive load, the slew-rate can be expressed as

$$dVc/dt = Ic(t) / C.$$

Two different analog drive circuits were designed and implemented for slew-rate control. Optimally, the drive circuitry should have the following characteristics: ability to drive a wide range of output loads, tallow precision adjustment of the slew-rate, drive the output from rail-to-rail, and have an output waveform with maximally linear transitions. Rather than trying to meet all of the design criteria in a single design, two approaches were implemented. These two designs differ on two points: the range of settable slew-rates and the total output range. The following sections detail these two approaches.

### 3.2.5.1 Op-Amp with a Tail-Current Controlled Slew-Rate

The first slew-control approach uses an op-amp where the slew-rate is controlled by setting the tail current [32]. The maximum current through the compensation capacitor *Cc* sets the slew rate in a basic op-amp design. For a simple source coupled differential-pair input stage, this is approximately *Iss*. Thus the slew rate is

$$dVc/dt = \mathrm{Iss} / \mathrm{Cc}.$$

Figure 3.14 is a basic op-amp as described, with *Iss* indicated. *Iss* can be controlled through the current mirror that is used to sink it at the base of the differential input. Like any standard current mirror, the mirrored current is proportional to the original current by the ratio of the transistor widths, thus *Iss* = *Ir* (*W1* / *W2*).

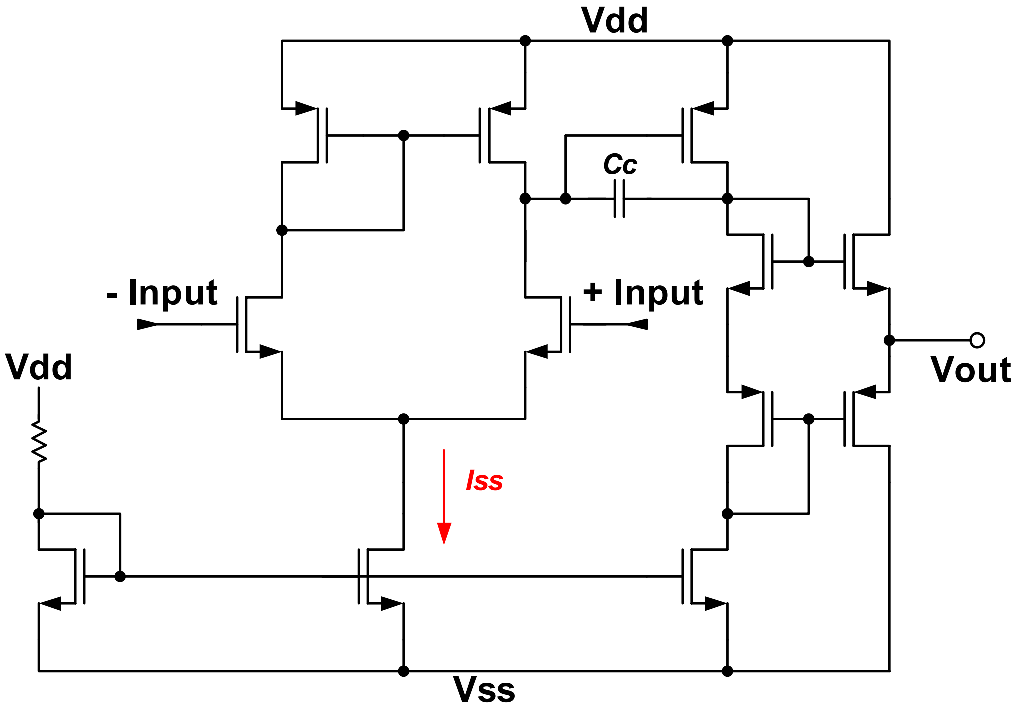


Figure 3.14: Basic operational amplifier

If M2 is a string of gate-connected transistors, each with a switch connecting its drain to the differential input pair, the bias current and thus slew rate can be easily controlled by simply switching on or off transistors.

Higher slew rates achieved by larger values of *Iss* will lower the open loop gain of the op-amp as a function of one over the square root of *Iss*. Inversely, lower *Iss* values raise the open loop gain. Also, the Vgs of the differential pair input is lowered with decreasing values of *Iss*.

Figure 3.15 shows the schematic for the final op-amp design. This design differs from the previous schematic of a basic op-amp in two points. The first of these is in this case the tail current through the differential pair is controlled externally. The transistors M41-M48 act as switches, allowing the tail current to be adjusted incrementally over eight steps. The other difference is in the output stage this design has no output buffer, but is instead directly coupled to the output. An alternative biasing scheme is also used to increase the output drive range.

A variety of SPICE simulations for several parameters of this op-amp were carried out. See Appendix 2 for the complete SPICE listing. The key simulation was the testing of the slew rate as a function of the tail current. SPICE simulation results for each incremental stepping of the tail current are shown in Figure 3.16, assuming a 100pF load. The layout used to implement this op amp is given in Figure 3.17.

#### 3.2.5.2 Feedback Controlled Current-Starved Inverter

The second design is a feedback controlled current starved inverter. The basic block diagram for this topology is shown in Figure 3.18.

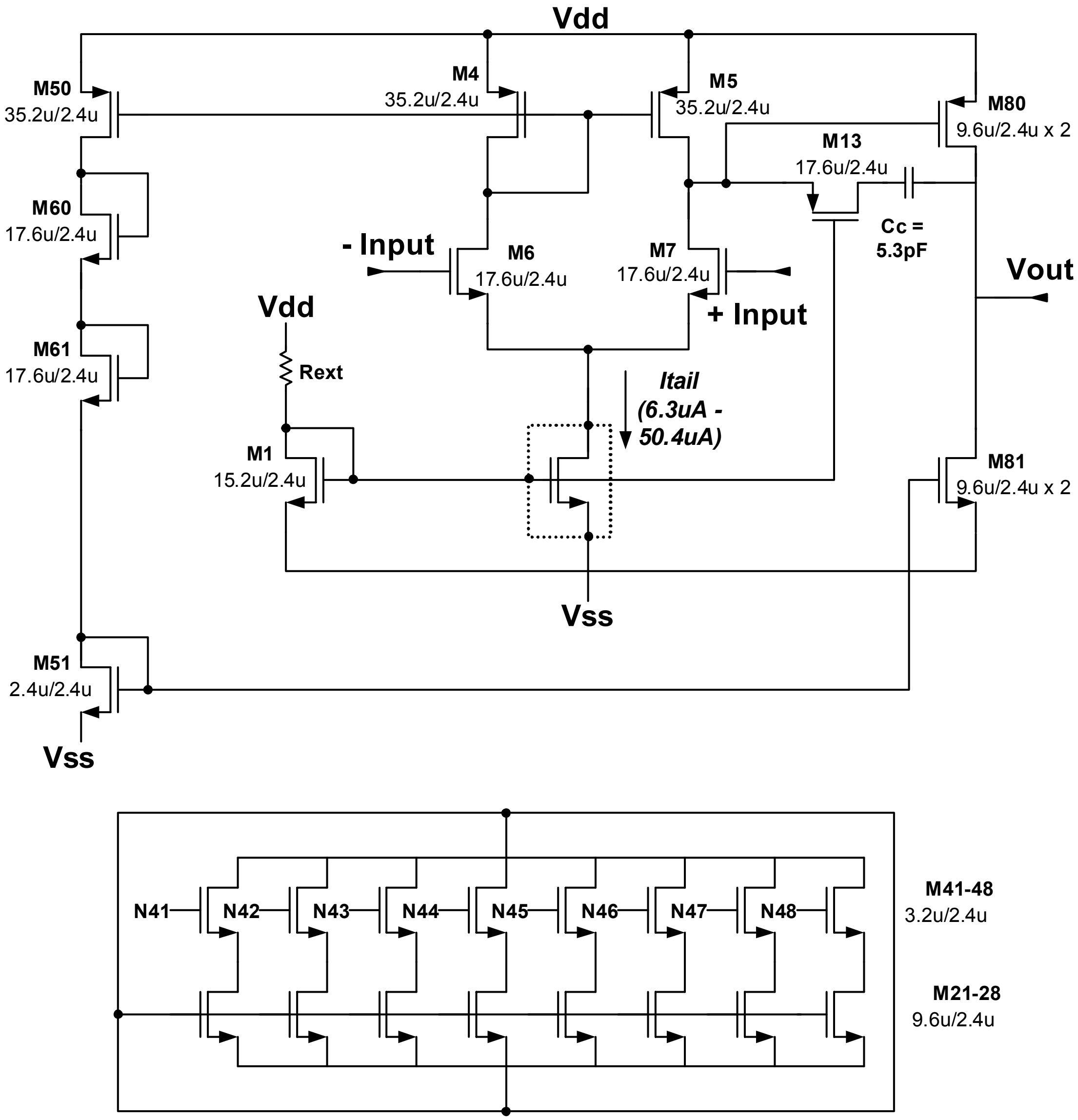


**Figure 3.15: Tail-current adjustable slew-rate controlled op-amp**

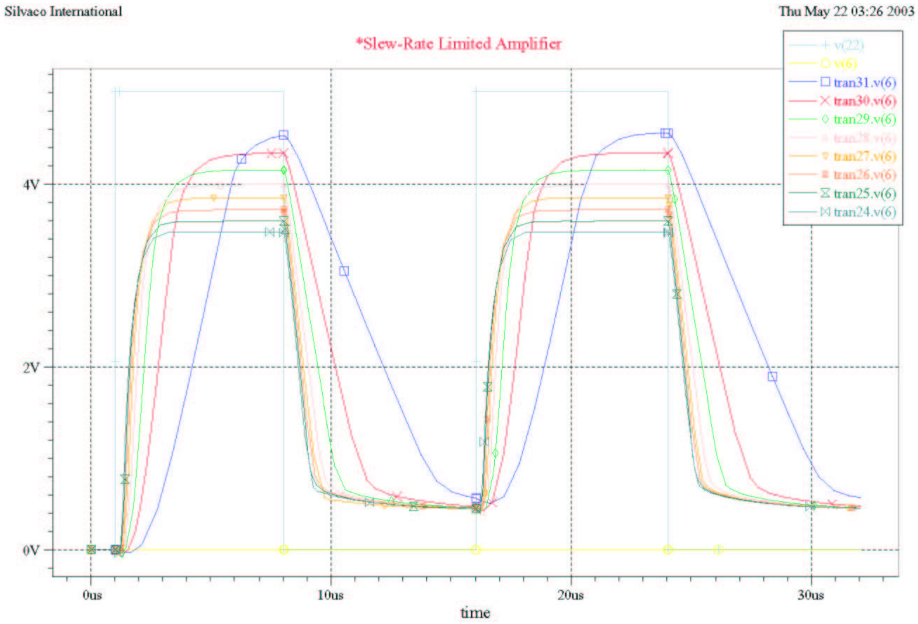


**Figure 3.16: Simulated slew range of tail-current limited amplifier (assuming a 100pF load capacitance)**

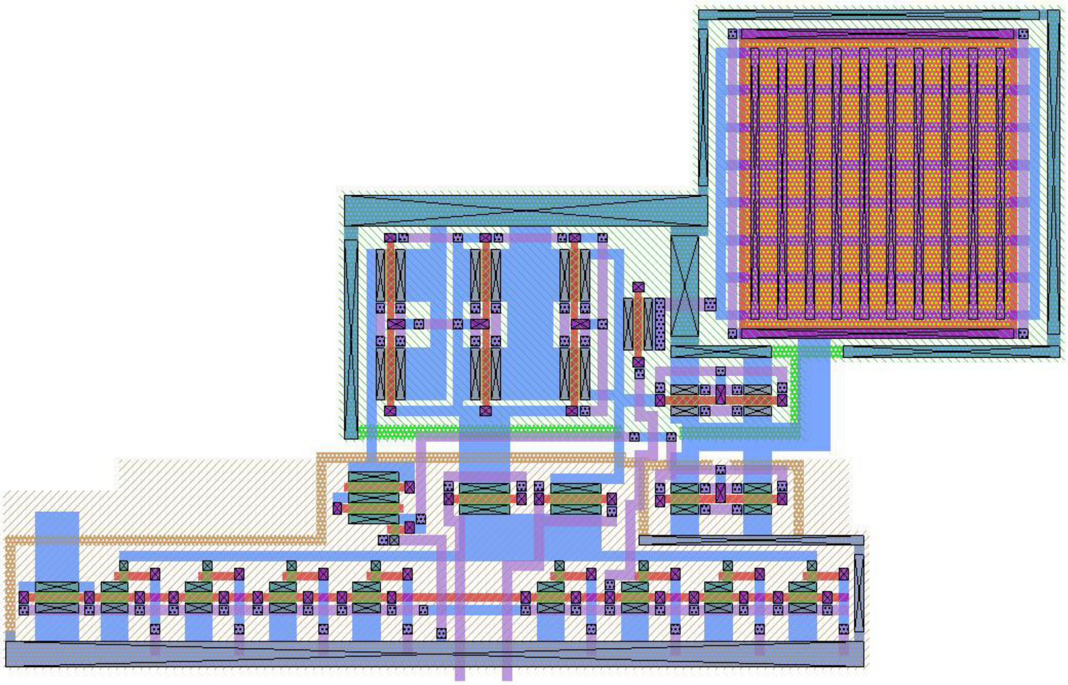


**Figure 3.17: Layout of the tail-current controlled adjustable slew-rate op-amp**

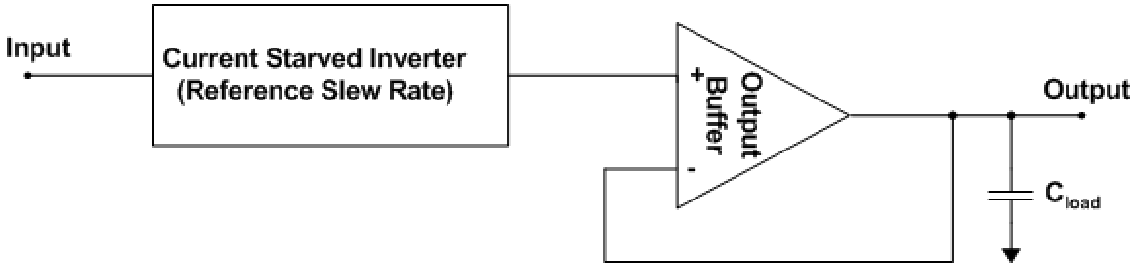


**Figure 3.18: Block diagram for a buffered current starved inverter**

Rather than have a single stage where the large signal response is adjusted, this design has one stage that sets a reference slew-rate and a second stage that buffers the first stage.

The reference slew-rate is set in the first stage by a current starved inverter. The circuit for this inverter is shown in Figure 3.19. This circuit acts as a basic inverter except the current through the load capacitance is limited by M3 and M4. Changing the load capacitance changes the slew-rate. In this case, the load capacitance is an off-chip adjustable capacitor. The only other unique feature of this circuit is the addition of Cslow. It was found through simulation that the addition of this capacitor sets a second pole resulting in smoother output curves.

The rail-to-rail output buffer is shown in Figure 3.20. This output stage can swing to either rail, although it is not linear over its full rail-to-rail output. As is seen in the simulation results in Figure 3.21, when the output signal approaches one threshold voltage of either rail, the output buffer no longer linearly tracks the input. This non-linear region is caused by either M31or M30 no longer being biased in saturation as the input approaches the positive or negative rails, respectively.

A variety of SPICE simulations for several parameters were carried out. See Appendix 3 for the complete SPICE listing used in the simulations. Again, one of the most interesting results was the slew-rate of the buffer. SPICE simulation results for a variety of reference capacitor values are shown in Figure 3.21, again assuming a load capacitance of 50pF and a bias current of 7.25µA. The layout used to implement this design is shown in Figure 3.22.

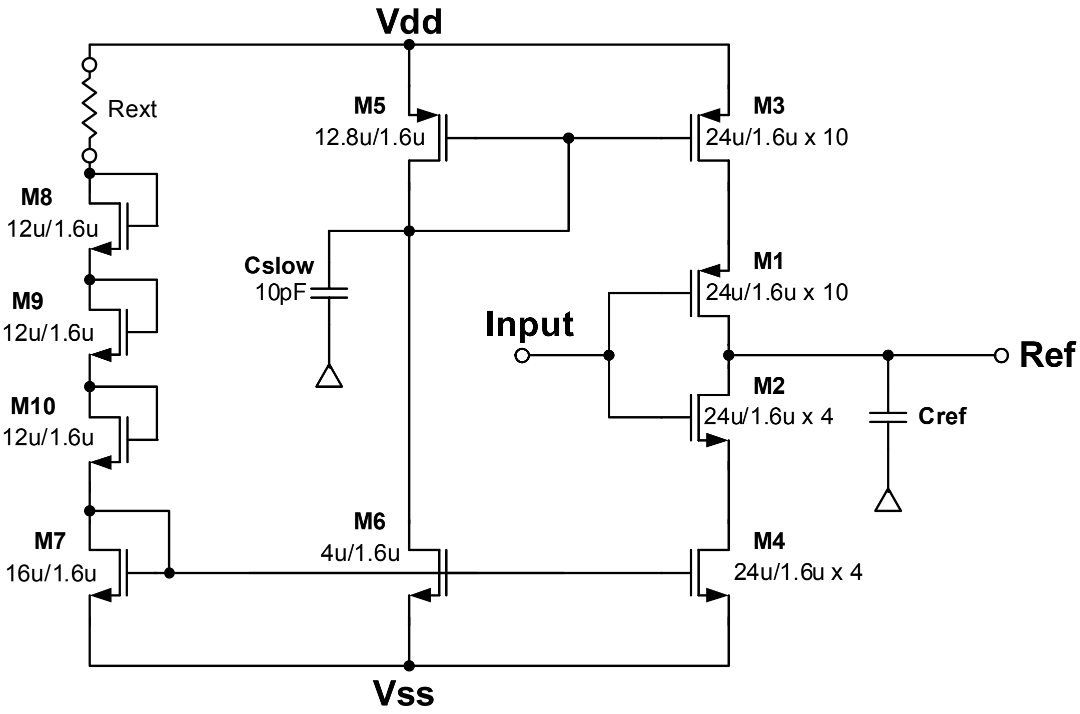


**Figure 3.19: Current-starved inverter reference slew-rate circuit**

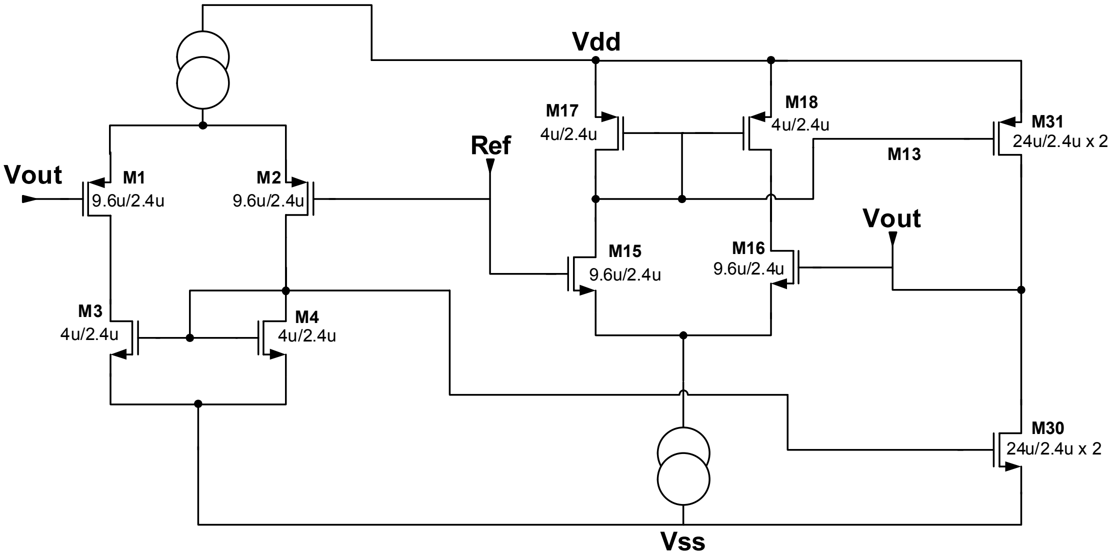


**Figure 3.20: Rail-to-rail output buffer**

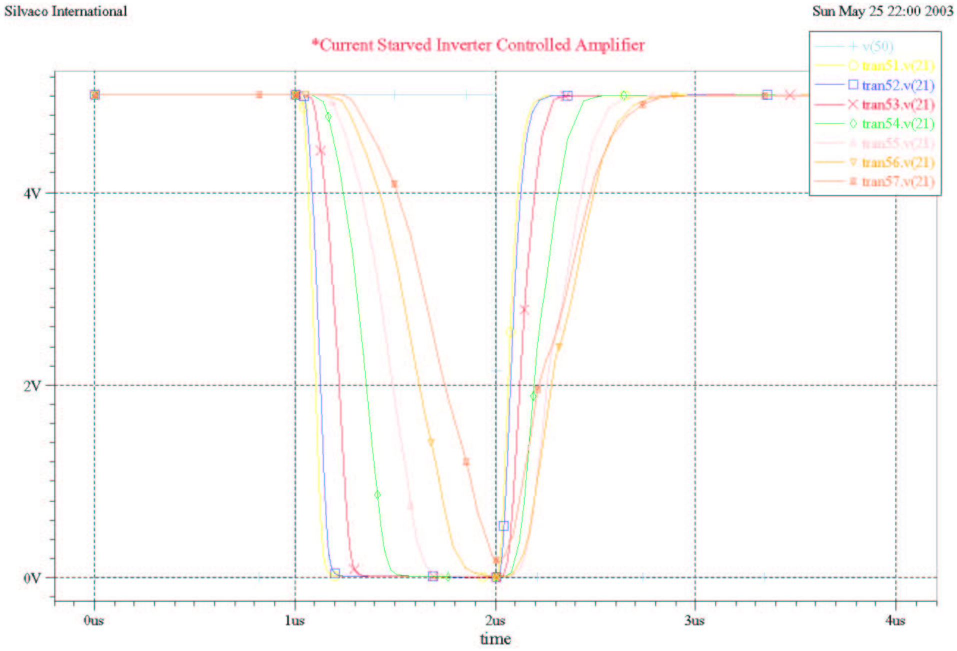


**Figure 3.21: Simulated slew range for a series of reference load capacitances and a bias current of 7.25µA**

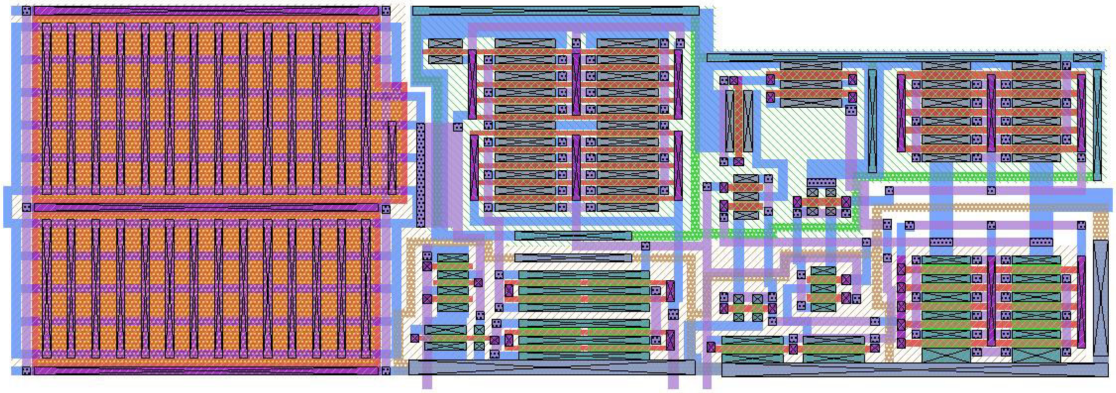


**Figure 3.22: Layout for buffered current-starved inverter**

## 3.3 External Receiver

The functions of the receiver can be broken down into three steps: detecting an individual pulse, decoding packets and detecting bit-errors within the packet, and tracking the packets to determine if there has been an error or skipped packet. Separate blocks within the receiver perform each of these functions. The block diagram for the receiver is shown in Figure 3.23. The following subsections describe the design of each of these pieces.

### 3.3.1 Charge-Sensitive Preamplifier with Discriminator Front-End

A charge sensitive preamplifier performs the front-end detection of pulses. A charge sensitive preamplifier is special case of a very low noise current integrator. An emitted pulse from the transmitter is seen as a current pulse at the input to the charge-sensitive preamplifier. The output signal produced by the charge-sensitive preamplifier is proportional to the input current pulse. A charge-sensitive preamplifier previously developed for another project, as shown in Figure 3.24, was chosen for this application. It was modified slightly by adjusting the feedback loop to reduce the low frequency gain after initial testing that showed that it had significantly more low frequency gain than was desired.

After initial testing, the output of the preamplifier was AC coupled to a gain stage with a gain of 21. This stage also incorporated a low-pass filter with a corner frequency of approximately 250 kHz. In this case a straightforward discriminator circuit was used. 20mV of hysteresis was added for noise protection and an LM311 comparator.

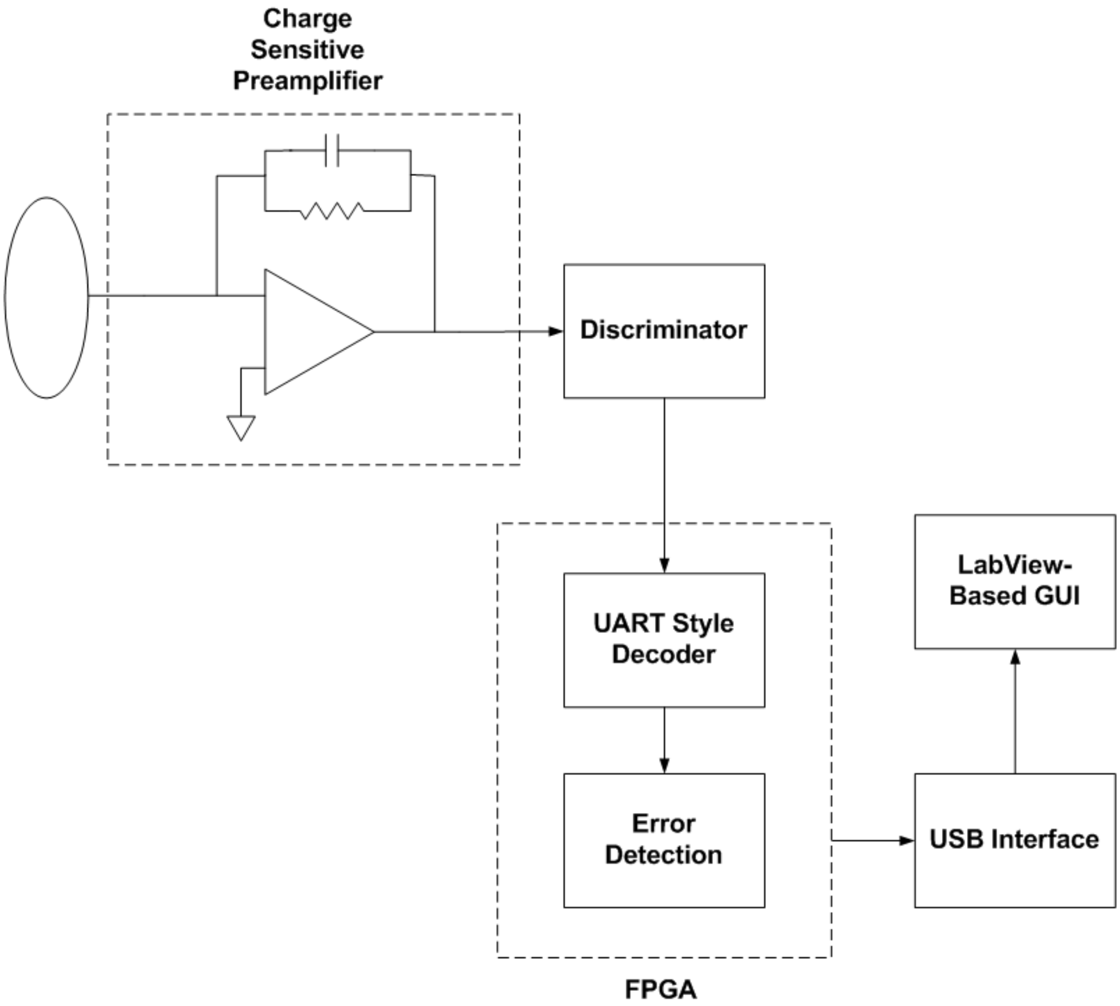


**Figure 3.23: Block diagram of the external receiver**

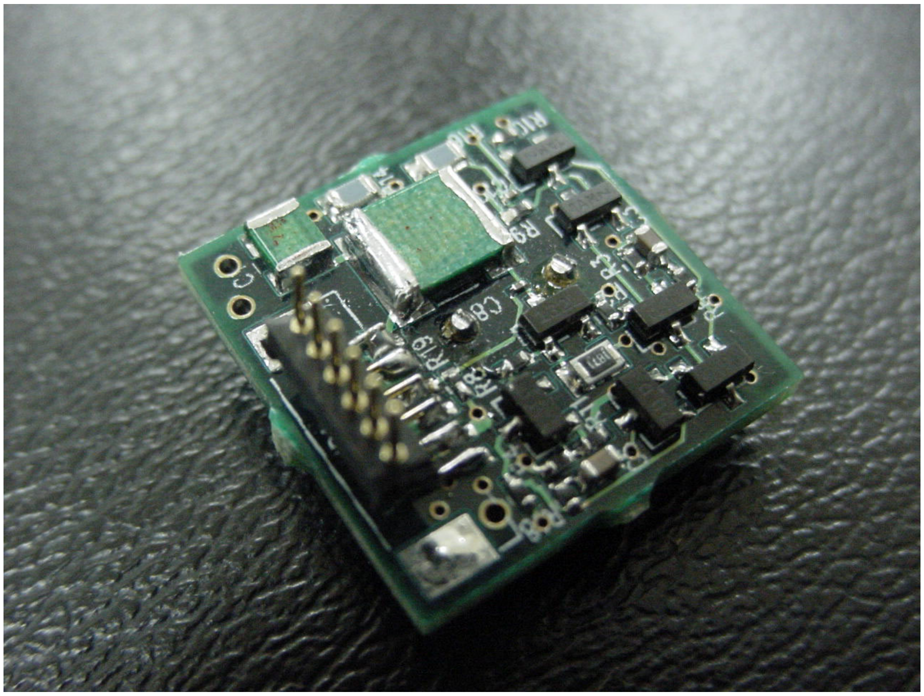


**Figure 3.24: The charge sensitive preamplifier used in the front-end**

### 3.3.2 UART-Style Decoder Block

The process of decoding the string of detected bits into a packet and extracting the data from the packet is the function of the decoder block. The decoder block was written in VHDL hand-in-hand with the transmitter block.

Information encoded in a bi-phase mark format is transmitted by the presence or absence of a mid-bit transition. Thus the basic functionality of the decoder block in decoding a single bit is that after the start of a bit the decoder polls for a transition later in the bit. If a transition occurs the bit is decoded as a ‘1’, otherwise the bit is decoded as a ‘0’. It does this by a method very similar to the way a UART works. A UART works by oversampling the input at a rate sixteen times the transmitted data clock. The decoder block works in much the same fashion, but rather than periodically polling for a fixed level it polls for a level change. It also polls for a window of eight clock cycles, allowing for jitter in the transmitter clock and also providing some error detection as the detection of multiple level-changes within a bit indicates an error. In this design any time an error is detected the entire packet is assumed to be corrupt. The process of decoding a single bit is shown in Figure 3.25.

Assuming the packet is correctly decoded, the next step is to extract the data contained within the packet. This is performed by first counting the thirty-two ‘1’s that make up the header. Next, bit is checked to see if it is the ‘0’ start bit. The following eight received bits are then put in the data register and the stop bit is checked. Finally, the twenty-four ‘1’s in the footer are counted. Any time an error is detected, the packet is assumed to be corrupt and the data is ignored.

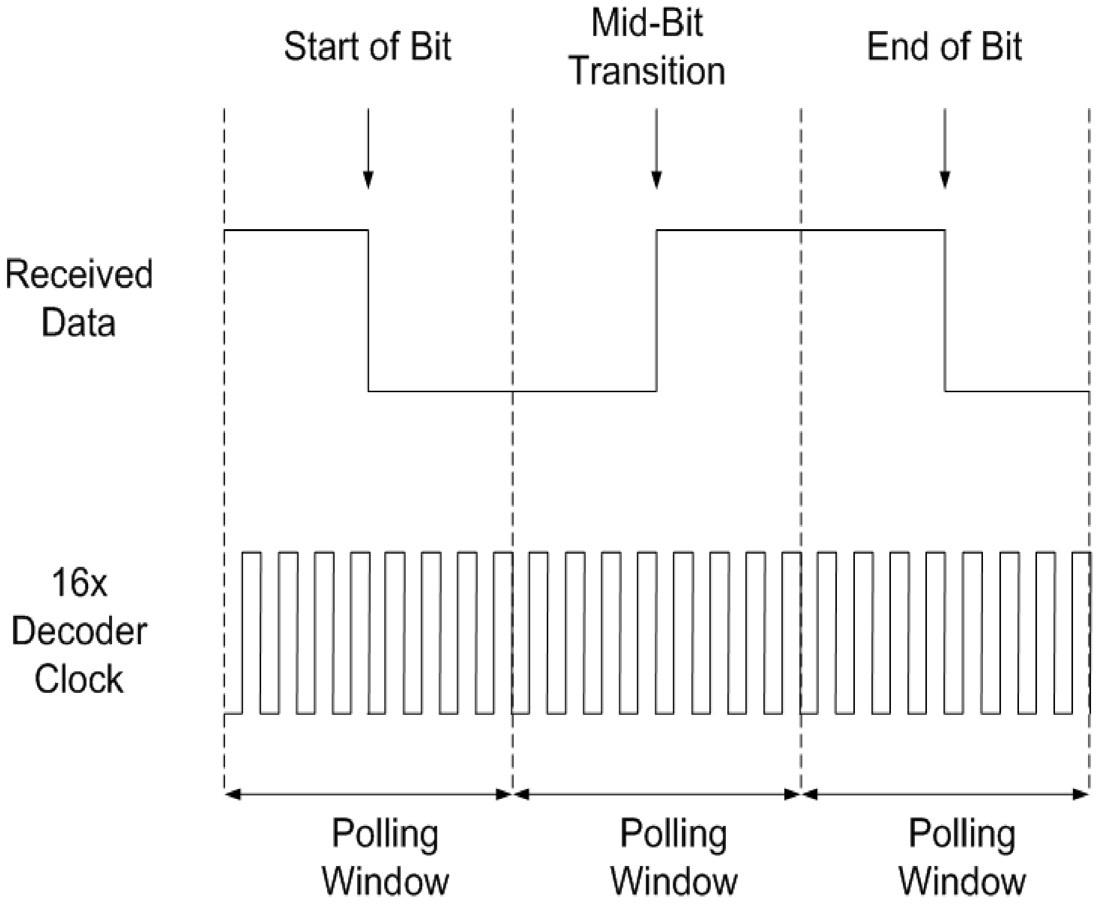


**Figure 3.25: The process of decoding a bit**

### 3.3.3 PC-based LabVIEW Graphical User Interface (GUI)

A LabVIEW Graphical User Interface (GUI) was designed to show the user if packets are currently being received and also how many packets have been either dropped or been incorrectly decoded. These results are displayed giving error statistics for the data link. It performs the first function by measuring the amount of time between received packets. If the amount exceeds 1ms it assumes that the connection has been lost, which is indicated as a timeout error. The second function is performed in two parts. First, if the decoder block detects individual bit errors it alerts the LabVIEW program and this error is in turn counted. Second, it checks the decoded data to make sure it is the correct data in the received sequence. If not, it counts this as a bad packet and reports this to the user.

## 3.4 Implant Transmitter Implementation

The core of the transmitter was implemented as a custom ASIC (CapTran1 chip) using a 1.5µ AMI process available through MOSIS. The implementation of this chip is described in the first subsection. The second subsection describes the transmitter PCB boards that were developed.

### 3.4.1 CapTran1 Chip

The CapTran1 chip was fabricated via MOSIS in a 1.5µ AMI process. The 'Tiny Chip' die size used is a relatively small 2.20 x 2.20 mm. In addition to the blocks described earlier, the chip included a few other features and necessary functions. These sections were needed to add functionality to the chip and to conserve pins. The number of pins was limited to 40 and all pins were used. Indeed pin count proved to be the only real limiting factor in the chips design, and as is seen in Figure 3.26 silicon space was not a limitation. One reason for the high pin count is that separate digital and analog power rails were used. Although using a single supply rail would free up some pins, separate rails were chosen in an effort to limit noise coupling from the digital blocks to the analog blocks.

The first of these additional features was the inclusion of two additional logic blocks. Both of these blocks were combinatorial switching blocks, where the first of these selects which encoding scheme is used, choosing from the schemes described earlier in this chapter or setting no encoding dependant on the position of two external switches.

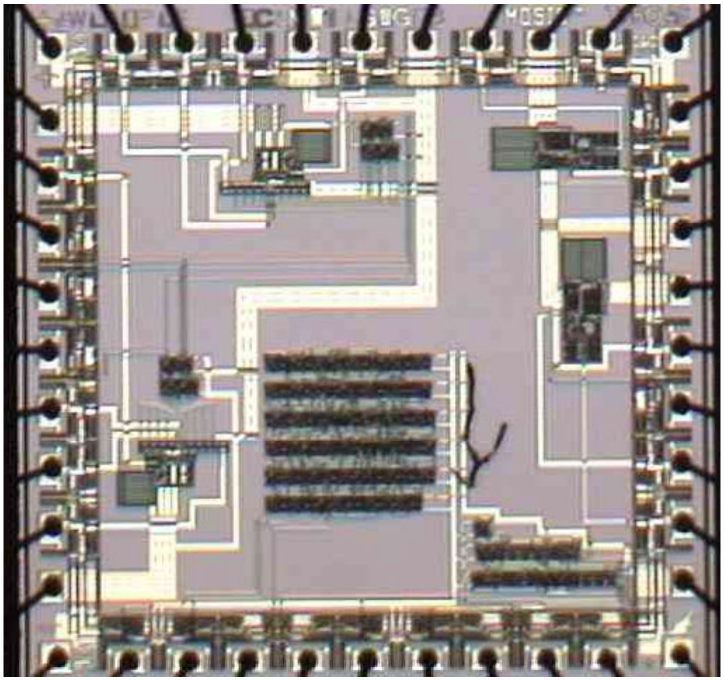


**Figure 3.26: The fabricated CapTran1 chip**

The second block sets the tail current in the tail current controlled slew-rate limited op-amp. This tail current has eight different settings that are selected via three external switches in a binary code.

The other important feature of the CapTran1 chip is the redundant analog blocks. Each analog driver is implemented twice; there are two slew-rate limited op-amps and two feedback controlled current-starved inverter based drivers. The reason for this was that one of the configurations that was tested was driving the plates differentially rather than single-ended to investigate both single and differential drive techniques. Results of this testing can be found in Chapter 4.

Examining the fabricated chip several observations can be made. First, the digital portions of the chip are considerably larger than the analog blocks, even with double implementations of the analog blocks. Indeed, the dummy packet generator is the largest single section on the chip. Second, better matching could have been done between the analog blocks. When testing the transmitter in differential output mode, one item of key importance is that the both plates must be driven as close to symmetrical as possible. Having the identical analog blocks implemented at right angles with each other rather than in the same orientation means that their matching will not be optimum, and thus the symmetry of their outputs may suffer. The third observation is that the layout has not been optimized to conserve space. A more space-conscience layout would allow a significantly greater amount of space for other blocks and functions. The final observation is that care must be taken in the handling of these ICs with the protective cover removed, or dust particles will contaminate the IC, as is clearly evident.

## 3.4.2 Test Transmitter PCB

Two test transmitter boards were fabricated to both test the ASIC functionality and to facilitate encapsulation for tests in the saline tank. The first of these was a large board referred to as the 'transmitter functionality' test board. This board was designed to test all features of the CapTran1 chip. Although it is battery powered and used in certain test setups, this board was not designed for encapsulation. The second test board was a much smaller board, referred to as the 'mini-transmitter' test board, and was designed specifically for encapsulation. These two boards are detailed in the following subsections.

### 3.4.2.1 Transmitter Functionality Test Board

The transmitter functionality test board was designed for testing each individual component of the CapTran1 chip. As such, this board was fairly large, measuring approximately 4" x 5" with an additional 2" x 2" daughter board containing the battery. This board is shown in Figure 3.27, and the corresponding schematic is in Figure 3.28.

The transmitter functionality test board implemented some additional logic beyond that contained on the CapTran1 chip. The reason for this was a slight oversight in the design of the CapTran1 chip. When the chip transmits, the packets should be encoded, but when it is not transmitting the output should be turned off. The problem with this lies in the encoder block. When the dummy packet generator is attached directly to the encoder, the spaces between the packets are also encoded. With a bi-phase mark encoding scheme this results in an undesirable transmission between packets (i.e. the encoder sees a string of zeros and encodes appropriately).

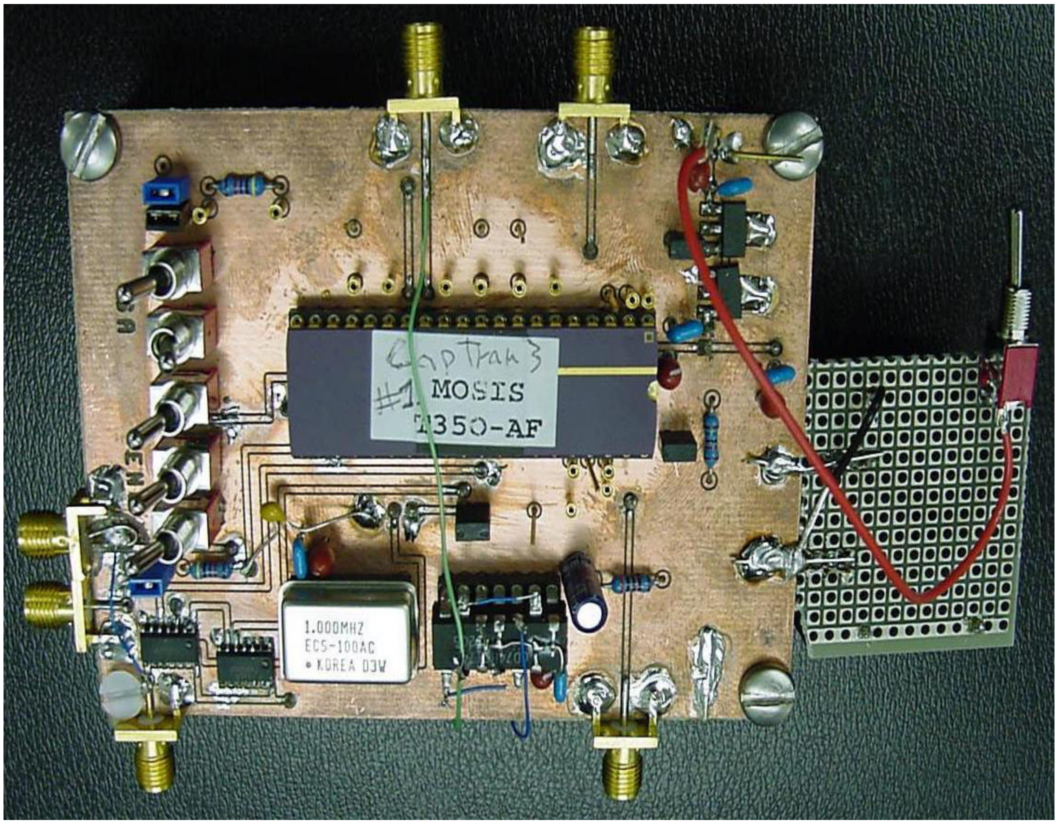


**Figure 3.27: The transmitter functionality test board**

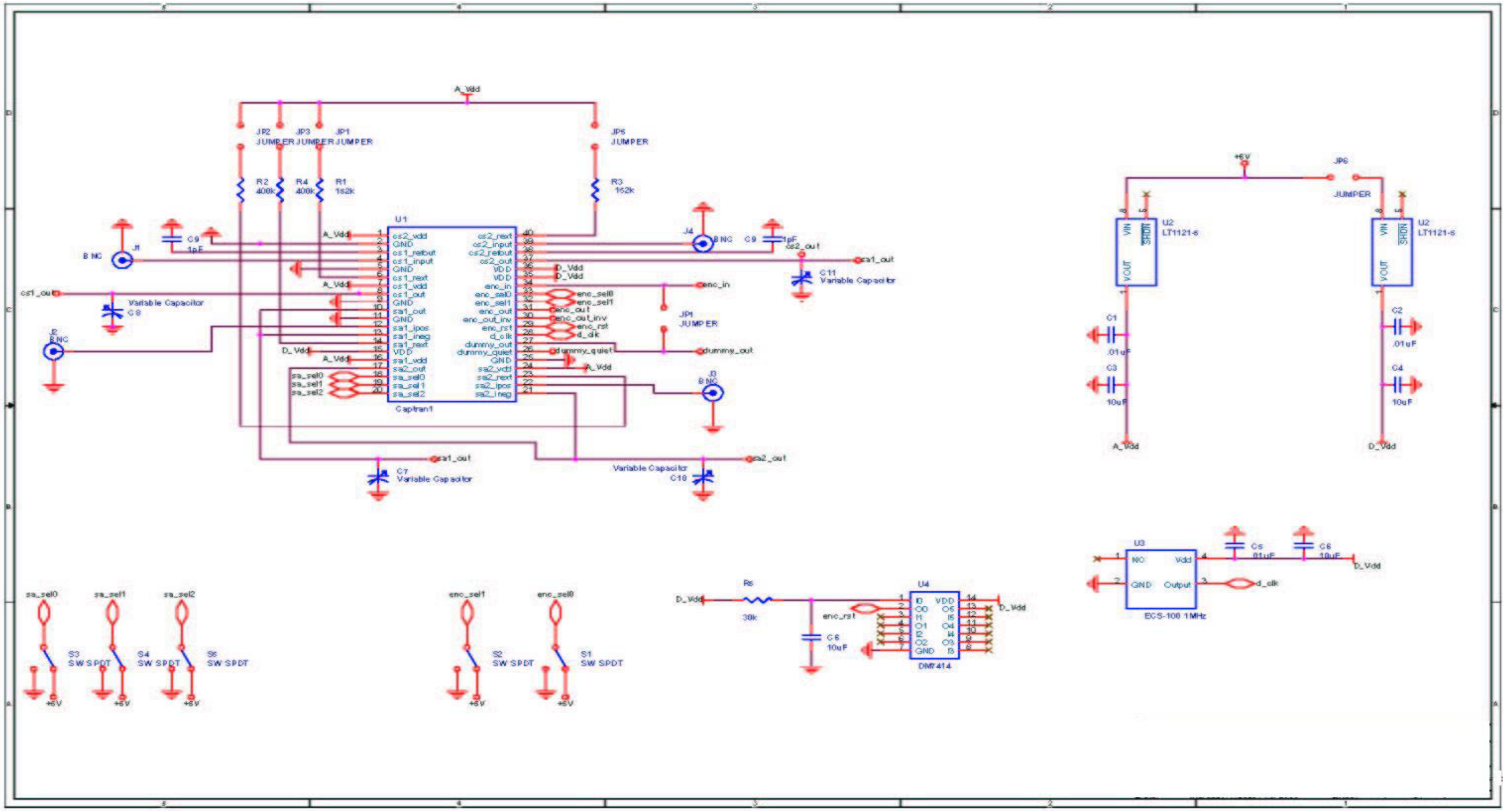


**Figure 3.28: Schematic of the transmitter functionality test board**

This problem arose because the encoder blocks and the dummy packet generator were designed using different methods (schematic style hand done logic implemented directly in MAGIC versus VHDL). The two blocks were simulated separately and never connected as a single entity prior to fabrication. The solution to this was to use external logic to turn the output off in between packets, made possible through the use of the 'sleep' output from the dummy packet generator block.

Another additional part needed was a Schmitt-trigger inverter. This was used to reset the flip-flops in the encoder block on initial power-up. In future implementations, both of these functions should be implemented on chip.

#### 3.4.2.2 Encapsulated "Mini-Transmitter" Test Board

As its name implies, the mini-transmitter test board was designed primarily to be small. The reason for this was that a smaller board would be easier to package and encapsulate. This board is significantly smaller than the transmitter functionality test board, measuring approximately 2" by 3". The complete unit is approximately 1" thick including the batteries and transmitting plates. The transmitting plates are mounted directly under the circuit board with the batteries in between, as shown in Figure 3.29. The schematic for the mini-transmitter is shown in Figure 3.30. This board is also modular in nature and made to be stacked. The CapTran1 chip and additional logic is contained on one board with the batteries positioned between this board and another board containing the transmitting plates. The entire assembly is mounted to a long (approximately 4"), thin piece of plastic. This piece is used to mount the assembly on the test fixture in the saline tank.

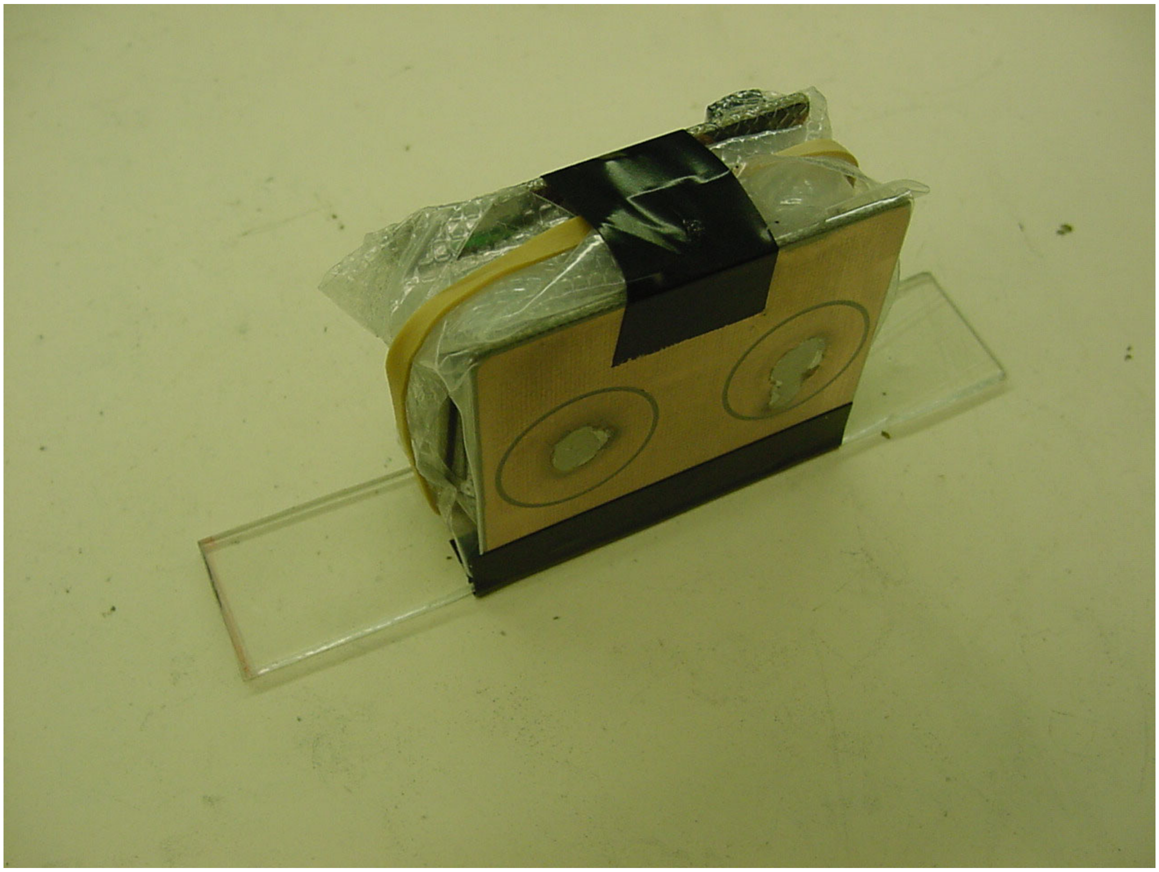


**Figure 3.29: The sealed mini-transmitter test unit**

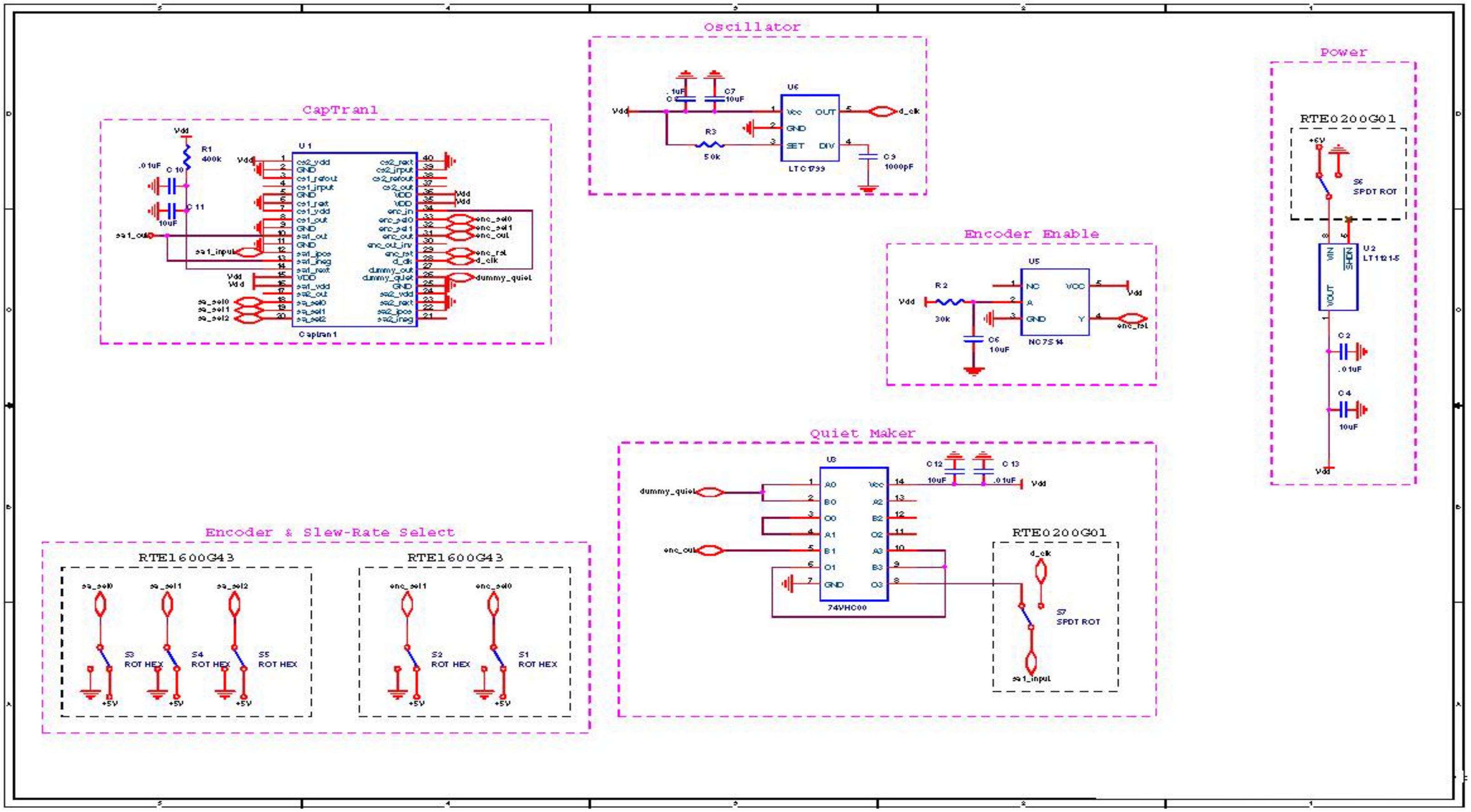


**Figure 3.30: Schematic of the mini-transmitter test board**

These implemented functions were the dummy packet generator, the bi-phase mark encoder, and only one tail current controlled slew-rate limited op-amp analog drive circuit. This not only reduced the size of the board, but low-power consumption components were also selected thus reducing the overall power consumption.

## 3.5 Receiver Implementation

Unlike the custom nature of the transmitter, the receiver was implemented using almost all of the off-the-shelf components in a modular fashion. Doing so allowed testing of each component without the whole receiver system having to be operational.

The receiver has five separate modules: the charge sensitive pre-amplifier front-end, a discriminator circuit, the FPGA based packet decoder, the USB communication board, and finally the PC based LabVIEW GUI. The following subsections briefly describe each of these except the charge sensitive pre-amplifier, as that has been sufficiently described in section 3.3.1.

### 3.5.1 Gain and Discriminator Board

The discriminator board transforms the analog waveform from the charge sensitive pre-amplifier into a digital output. As described earlier, the actual discriminator circuit is implemented via a Linear Technology LM311 discriminator chip. A hysteresis loop with a hysteresis of 20mV was also added as well as the ability to adjust the mid-point of the discriminator. This value of hysteresis was chosen by evaluating the noise seen at the output of the charge-sensitive pre-amplifier when no input signal was applied. The gain stage was implemented using an OP37 op-amp circuit with a low frequency gain of 21. The OP37 was selected for its high slew-rate (17V/µS) and gain-bandwidth product (40 MHz, giving a 1.9 MHz bandwidth at the chosen gain).

The implemented circuit has slightly more functionality than as described previously in the design section (section 3.3.1). One test configuration used was testing the transmitter in a differential-drive configuration rather than a single-ended configuration. To implement this on the receiver side, a Burr-Brown INA105 unity gain differential amplifier front-end was used. The same circuit is used in single-ended mode by grounding one input to the differential amplifier.

### 3.5.2 FPGA Implementation

All VHDL was implemented using Altera Stratix FPGAs shown in Figure 3.31. Rather than use a custom board, an off-the-shelf Stratix 672 SmartPack board from

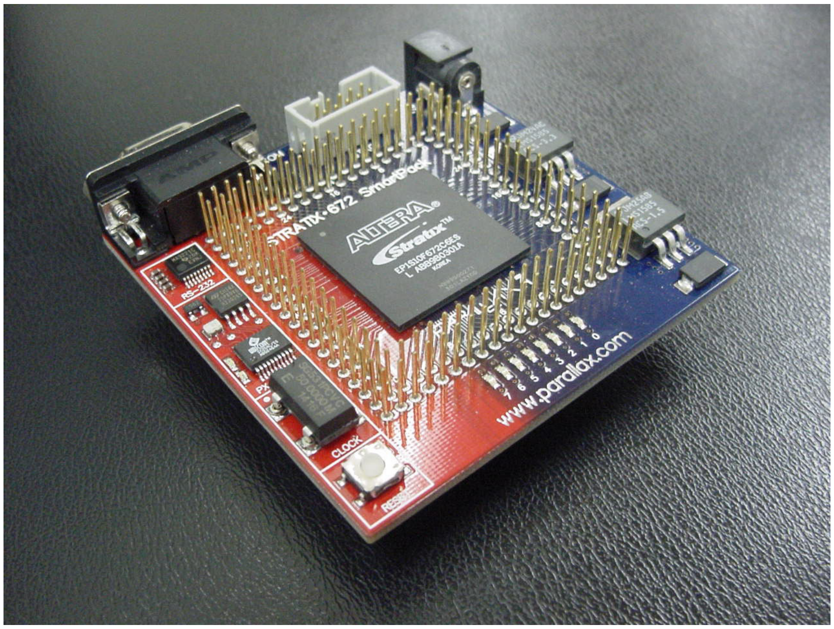


**Figure 3.31: The Stratix 672 SmartPack FPGA board used in the receiver**

Parallax was used. Two of these boards were purchased to help design and test the transmitter and receiver blocks. In this mode, one board was used as a test transmitter and the other as the UART receiver block.

### 3.5.3 USB Communication Board

The USB communication board handles the communication between the Stratix 672 SmartPack FPGA board and the PC. There were three choices in the type of link to implement between the FPGA board and the PC. The first of these was to do a standard RS-232 serial link directly from the FPGA to the PC. Although easy to implement, a serial link has some significant disadvantages when compared with a USB link, namely speed and the ability to daisy chain multiple devices. However, in this application a USB link was chosen primarily because the experience gained with its implementation will be leveraged in future applications. The second choice was to implement a USB interface, but have the interface be implemented directly on the FPGA board. This could be accomplished as there is both plenty of space on the FPGA and ample pins physically available on the board. However, although this would be the most elegant approach, due to the complicated nature of the USB interface this option was not feasible without sacrificing considerable time and effort. Instead the option that was chosen was to use a USB link but implement it using an off the shelf parallel-to-USB interface board. The board used was the DLE-USB245M by DLPdesign, based on the FTDI FT245BM chip, as shown in Figure 3.32. Interface with this board consists of an 8-bit data bus and 4 control lines. However, because data is sent in a non-bi-directional form from the FPGA to the PC, only 2 of the 4 control lines are used in this application.

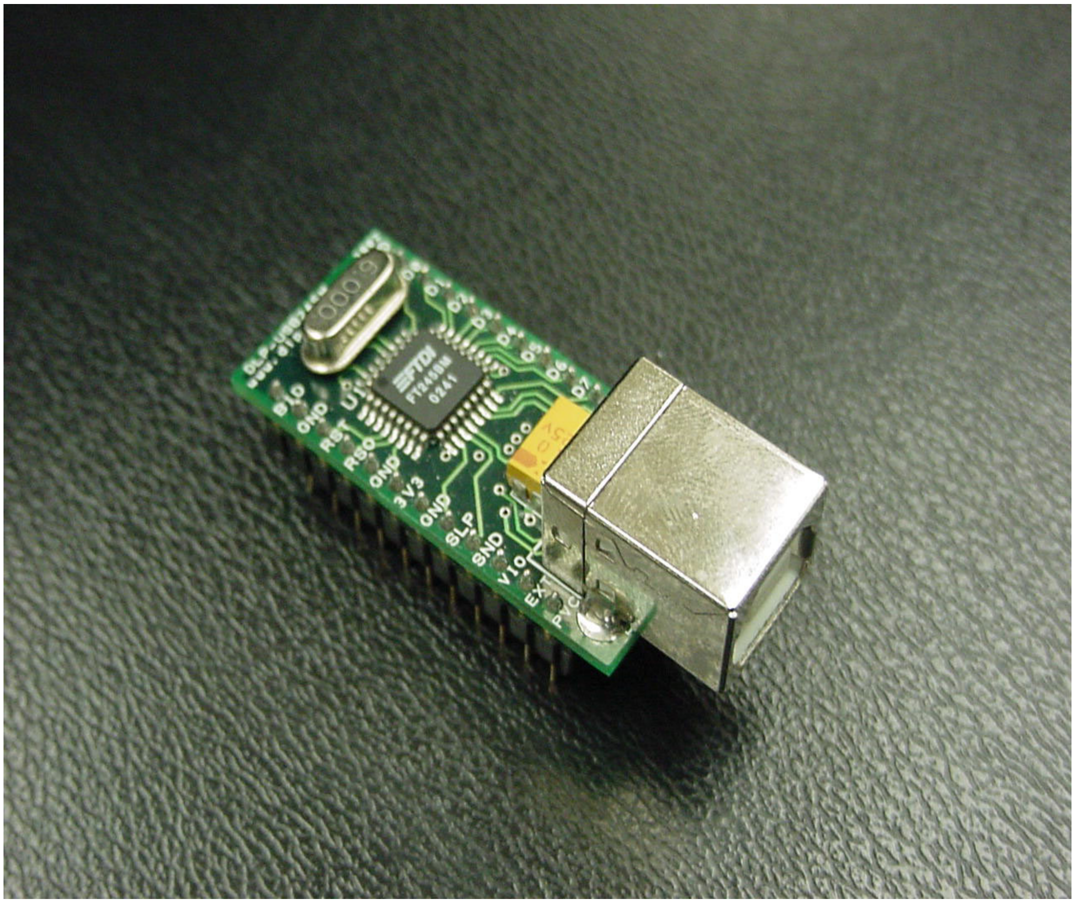


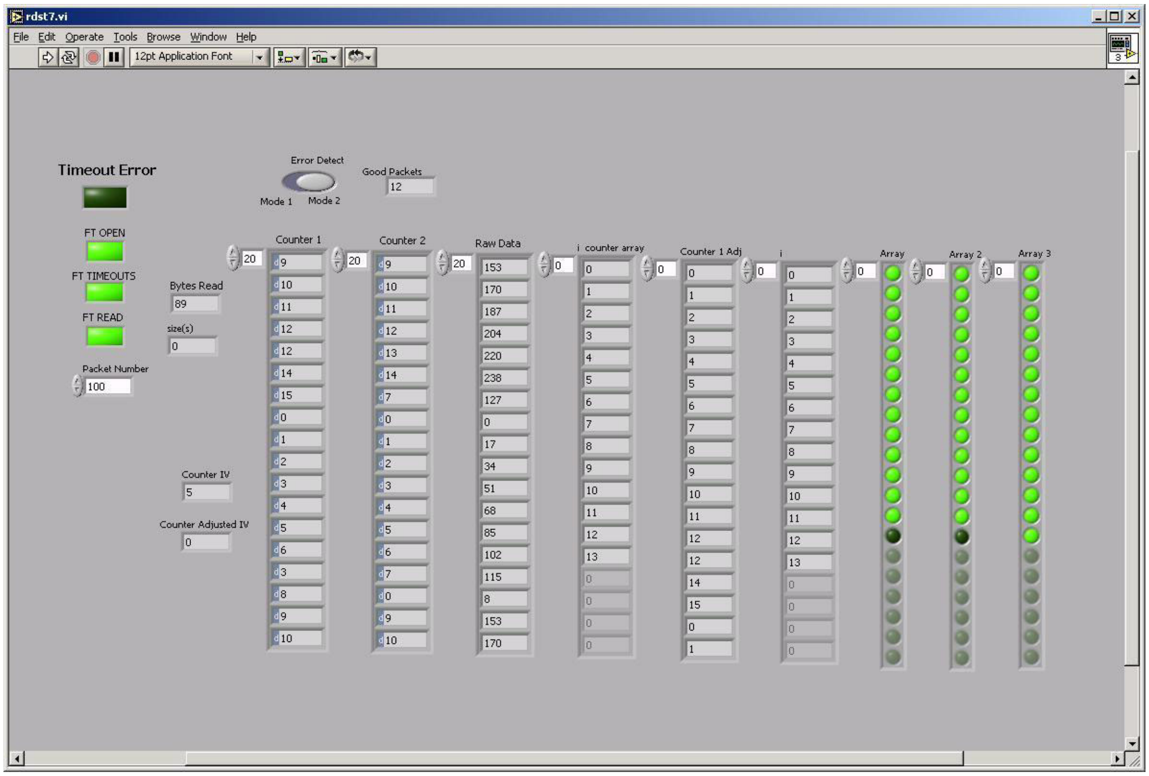
**Figure 3.32: Parallel-to-USB interface board**

The VHDL code used to implement the communication between the USB board and the FPGA is given in Appendix 4.

### 3.5.4 LabVIEW GUI Implementation

The function of the LabVIEW GUI is to enable statistical analysis of the link's stability. This is performed by counting the number of successful packets that are received before an erroneous packet is received. The LabVIEW program detects erroneous packets using the following rules. First, all packets must be received in the proper order. Verification of the proper order is accomplished by tracking the data that is contained in the packets. This data consists of two identical 4-bit counters. If any packet is received in which the value of the counters does not correspond to the previous value incremented by one. Second, the value of the two counters must be identical. If either of these conditions occurs the LabVIEW program records the error and displays the number of successful packets transmitted before the error. Running the LabVIEW program repeatedly, a good statistical analysis of the link is achieved by averaging the results.

Figure 3.33 shows the LabVIEW GUI detecting an erroneous packet. In this case an error occurred in packet number 13. Examining the columns marked 'Counter 1' and 'Counter 2', note that the value contained in 'Counter 2' is incorrect. The box marked 'Good Packets' displays the number of packets successfully received before this error.



rdst7.vi

Timeout Error ○

FT OPEN ●

FT TIMEOUTS ●

FT READ ●

Packet Number: 100

Error Detect: Mode 1 / Mode 2

Good Packets: 12

Bytes Read: 89

size(s): 0

Counter IV: 5

Counter Adjusted IV: 0

| Counter 1 | Counter 2 | Raw Data | i counter array | Counter 1 Adj | i | Array | Array 2 | Array 3 |
|---|---|---|---|---|---|---|---|---|
| 20 | 20 | 20 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 9 | 153 | 0 | 0 | 0 | ● | ● | ● |
| 10 | 10 | 170 | 1 | 1 | 1 | ● | ● | ● |
| 11 | 11 | 187 | 2 | 2 | 2 | ● | ● | ● |
| 12 | 12 | 204 | 3 | 3 | 3 | ● | ● | ● |
| 12 | 13 | 220 | 4 | 4 | 4 | ● | ● | ● |
| 14 | 14 | 238 | 5 | 5 | 5 | ● | ● | ● |
| 15 | 7 | 127 | 6 | 6 | 6 | ● | ● | ● |
| 0 | 0 | 0 | 7 | 7 | 7 | ● | ● | ● |
| 1 | 1 | 17 | 8 | 8 | 8 | ● | ● | ● |
| 2 | 2 | 34 | 9 | 9 | 9 | ● | ● | ● |
| 3 | 3 | 51 | 10 | 10 | 10 | ● | ● | ● |
| 4 | 4 | 68 | 11 | 11 | 11 | ● | ● | ● |
| 5 | 5 | 85 | 12 | 12 | 12 | ● | ● | ● |
| 6 | 6 | 102 | 13 | 12 | 13 | ○ | ○ | ● |
| 3 | 7 | 115 | 0 | 14 | 0 | ○ | ○ | ○ |
| 8 | 0 | 8 | 0 | 15 | 0 | ○ | ○ | ○ |
| 9 | 9 | 153 | 0 | 0 | 0 | ○ | ○ | ○ |
| 10 | 10 | 170 | 0 | 1 | 0 | ○ | ○ | ○ |

**Figure 3.33: The LabVIEW program detecting an incorrect packet**

# Chapter 4 VERIFICATION AND CHARACTERIZATION

## 4.1 Testing Overview

A prototype communication system (detailed in Chapter 3) was fabricated to enable validation of the capacitance-based communications technique. This chapter summarizes the verification and characterization of this prototype system. The testing of the system was done in three parts: testing the functionality of each of the blocks and the system as a whole, evaluation of the telemetry link using various mediums, and finally statistical reliability measurements of the link. The purpose of this testing was not only to test the performance of the system, but also test the feasibility of this technique in an implanted application. The following sections present the results of these tests and conclude with a discussion of alternate tests, factors affecting link reliability, and the feasibility of this technique in an implanted application.

## 4.2 Functionality Testing

Before evaluating the performance of the system, the functionality of each block was tested and then the entire system as a whole was tested to verify complete end-to-end communication. This testing and verification is presented in the following subsections and details the following: testing the functionality of each block on the custom ASIC-based transmitter, evaluating the charge sensitive pre-amplifier, testing the functionality of the VHDL-based packet decoder and USB link, measuring the power consumption of the transmitter, and finally verifying end-to-end communication. In each section measured values are compared with simulated results where appropriate.

## 4.2.1 Testing the Functionality of the CapTran1-Based Transmitter

As described in Chapter 3, the transmitter functionality test board was designed for the testing and evaluation of each block implemented on the CapTran1 chip. This board also contained some additional logic necessary to interface the different blocks to create a fully functional transmitter. Functionality testing of the CapTran1 chip was broken down into verification of the digital sections and evaluating the performance of the analog sections for two different capacitive loads.

### 4.2.1.1 Verification of Digital Blocks

Testing of the digital blocks consisted of two parts: verifying that the packet generator was performing as expected, in both the formatting of individual packets and the sequencing of packets, and then second that the encoder block was operational. Each block was found to function correctly. Figure 4.1 shows a complete packet generated by the dummy packet block after it has been encoded in the bi-phase mark scheme by the encoder block. Note the regular bits at the header and end trailer of the packet.

### 4.2.1.2 Evaluating the Analog Blocks

The analog blocks were designed to drive a range of different capacitive loads with externally programmed slew-rates. Testing was performed with two fixed capacitive loads (10pF load and 100pF) representing the minimum and maximum expected load.

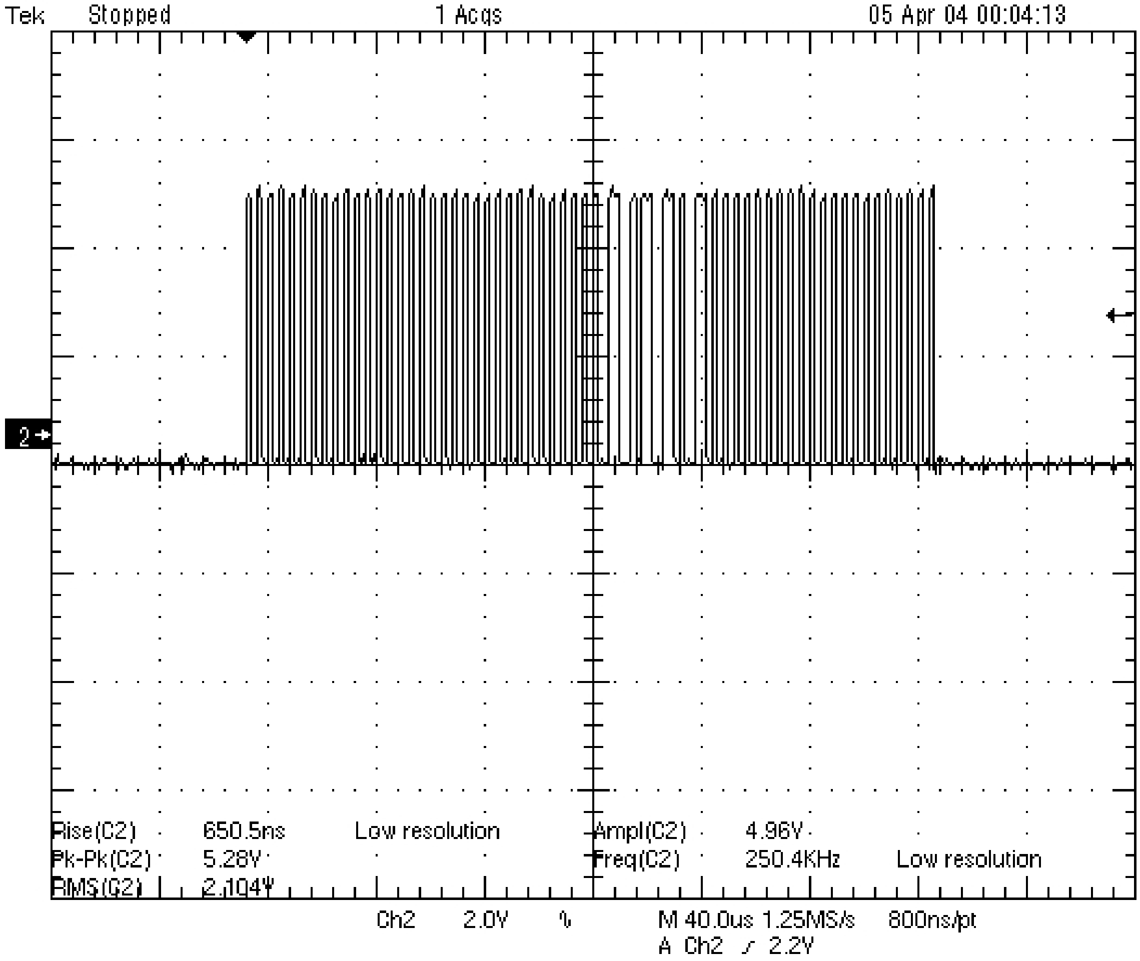


**Figure 4.1: An encoded packet generated by the "dummy" packet generator block**

The results from this testing are presented and compared to simulation results for first the tail-current controlled op-amp and then the feedback controlled current-starved inverter in Table 4.1and Table 4.2 respectively.

When initially testing the feedback controlled current-starved inverter, it was found that the slew-rate and associated control was significantly different than simulations predicted. The slew-rate was significantly slower than expected and the waveform had severe distortion versus the desired waveform. The problem of distortion is directly related to the problem of the smaller slew rate. As was mentioned in Chapter 3, one of the known flaws in this design was that as the output approaches one threshold voltage of either rail, the output buffer begins to lose the ability to accurately track the reference waveform. This distortion is most apparent for slower slew-rates.

The solution to this problem was to increase the bias current for the reference circuit until the circuit achieved the same level of functionality as the simulation. The results shown in both and Figure 4.2 and Table 4.2 are after increasing the bias current with a 100pF load. Because the output buffer is also biased by the same reference current source, but power consumption was also increased dramatically. Therefore it was decided that the tail-current controlled slew-rate adjustable op-amp would be used on the battery powered mini-transmitter board.

Unlike the feedback controlled current-starved inverter approach, the tail-current controlled op-amp has a much tighter slew-rate control for different load capacitances. However, the slew-rate adjustability was found to be significantly different than was simulated.

**Table 4.1: Slew-rates for the tail-current controlled operational amplifier**

| **Tail Current Stepping** | *Simulated Slew-Rate [V/μs]* | | *Tested Slew-Rate [V/μs]* | |
|---|---|---|---|---|
| | **10 pF Load** | **100 pF Load** | **10 pF Load** | **100 pF Load** |
| 1 | 1.27 | 1.26 | 1.2 | 1.2 |
| 2 | 2.55 | 2.47 | 2.3 | 2.3 |
| 3 | 3.82 | 3.15 | 3.4 | 3.3 |
| 4 | 5.05 | 4.44 | 4.6 | 4.4 |
| 5 | 6.14 | 5.27 | 6.0 | 5.8 |
| 6 | 7.44 | 6.69 | 6.9 | 6.5 |
| 7 | 8.43 | 8.15 | 8.3 | 7.9 |
| 8 | 9.51 | 8.91 | 9.3 | 8.6 |

**Table 4.2: Slew-rates for feedback controlled current-starved inverter**

| **Reference Capacitor Value** | *Simulated Slew-Rate [V/μs]* | | *Tested Slew-Rate [V/μs]* | |
|---|---|---|---|---|
| | **10pF Load** | **100pF Load** | **10pF Load** | **100pF Load** |
| 1 pF | 90.4 | 32.9 | 15.9 | 13.9 |
| 4.7 pF | 21.8 | 20.5 | 9.6 | 9.1 |
| 10 pF | 14.0 | 14.2 | 7.3 | 6.0 |
| 15 pF | 6.48 | 6.80 | 5.6 | 4.9 |
| 22 pF | 5.33 | 5.57 | 4.4 | 3.9 |

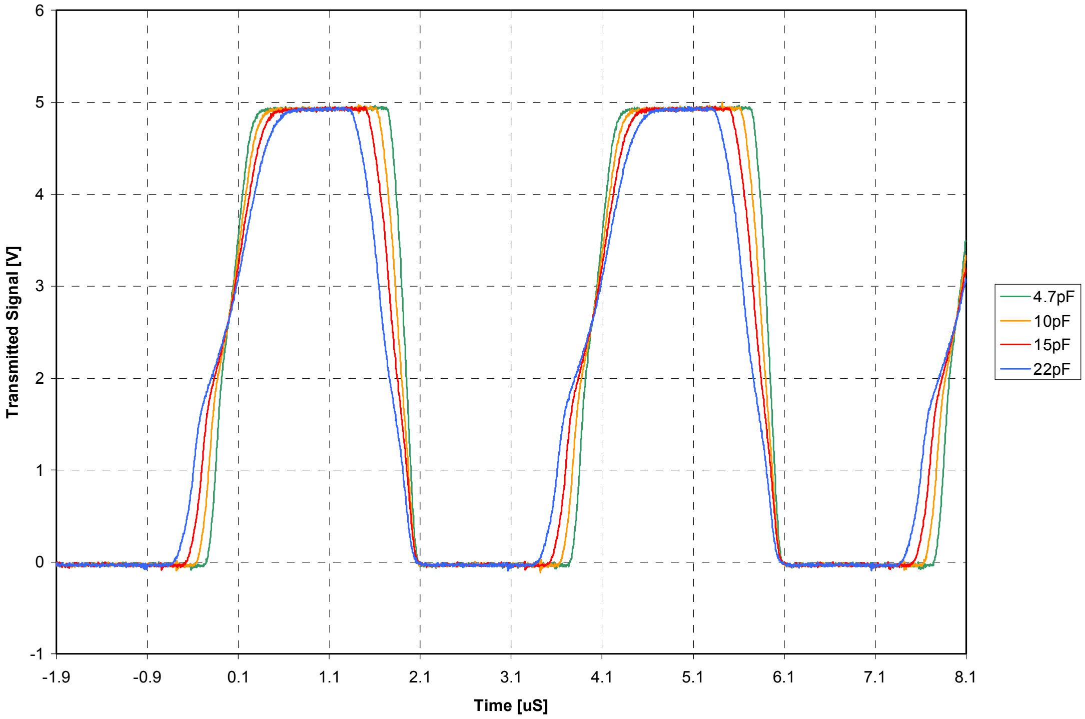


**Figure 4.2: Measured waveforms demonstrating the slew range of the feedback controlled current-starved inverter**

From testing, the slew-rate adjustability has a very high transition from slow to fast slew-rates. In other words there is large difference between setting 1 and 2, but increasingly less difference between the faster slew rates such as settings 7 and 8. While this may somewhat limit the link evaluation tests, it was decided that the tail current controlled op-amp was still more desirable for implementation on the mini-transmitter test board. Measured waveforms using a 100pF load are shown in Figure 4.3.

### 4.2.2 Power Consumption

Connecting a Hewlett-Packard 34401A multimeter inline with the battery on the transmitter functionality test board, a variety of power consumption measurements were taken. The average current was measured and multiplied by the battery voltage as different sections of the chip were enabled, providing an average power measurement of each section. These tests were compared with power usage when the CapTran1 chip was removed completely from the test board, thus isolating the actual power consumption of each block. Tests were then performed to determine the average power consumption for a constantly transmitted 250 kHz clock versus the transmission of data packets using both of the analog drive circuits. The data packets were transmitted at a rate of approximately 788 packets per second in regularly spaced intervals. These results are presented in Table 4.3.

From the results presented in Table 4.3 it is seen that the majority of the power consumption on the test board is being used by the off chip components. The oscillator generating the clock signal is the greatest source of power consumption, with very minor consumption by the other components.

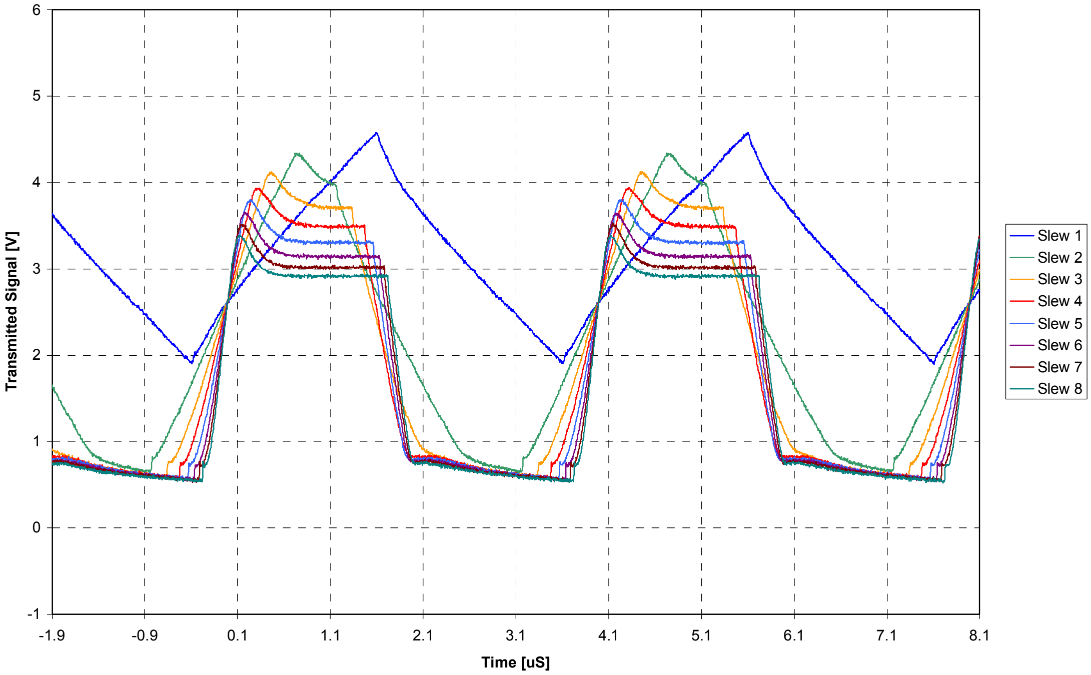


**Figure 4.3: Slew range of the tail-current controlled op-amp measured using a load capacitance of 100pF**

**Table 4.3: Average power for various blocks**

| Component Tested | Measured Average Power |
|---|---|
| Complete fully functional transmitter board with slew-rate limited op-amp transmitting data packets | 246.4mW |
| Off-chip components only (oscillator, pull-up resistors, necessary logic circuits) | 241.7mW |
| On chip digital blocks (Dummy Packet Generator and Encoder) | 5.13mW |
| Current-starved inverter based amp transmitting a constant clock into free-air on the fastest slew-rate setting | 6.45mW |
| Current-starved inverter based amp transmitting data packets into free-air on the fastest slew-rate setting | 1.05mW |
| Tail current controlled slew-rate limited op-amp transmitting a constant clock into free-air on the lowest slew-rate setting | 1.34mW |
| Tail current controlled slew-rate limited op-amp transmitting data packets into free-air on the lowest slew-rate setting | 218µW |

Testing the mini-transmitter board, which uses a lower power oscillator, an average power consumption of 27mW was measured when the board was configured to transmit a constant clock via the tail current controlled slew-rate limited op-amp into free-air on the lowest slew-rate setting.

## 4.3 Telemetry Link Evaluation

One of the features implemented on the CapTran1 chip was the ability to drive a load of relatively unknown capacitance with a fixed slew-rate. The following tests characterize the effects of varying the slew rate versus the received signal strength through various mediums. The following five test setups with different transmitter loads were used: in free air over a range of distances, through tissue at a fixed distance, through saline solution at a fixed distance in a partially submerged bath, completely encapsulating the transmitter in saline, and finally completely encapsulating the transmitter in hamburger meat at fixed distances. Each of the fixed distance tests used a range of slew rates while the tests varying distance used what was found to be the optimum slew rate. In each test the received waveform was measured after the gain stage in the receiver. In the free-air and partially submerged saline tests the transmitted waveform is also presented to show the effects of loading on the analog drive circuit. The mini-transmitter test unit, as described in Chapter 3, was used for each of these tests. It was configured to transmit a continuous signal of approximately 202 kHz. As mentioned in the previous section, only the tail-current controlled op-amp was used as the analog drive circuit.

### 4.3.1 Transmission in Free Air

Figure 4.4 shows the free air test setup. This test setup consists of two un-insulated transmitting plates aimed directly at two un-insulated receiving plates. One transmitting plate was connected to the output driver of the mini-transmitter and the other to the mini-transmitter ground. Likewise, the opposite receiving plate was connected to the input of the charge sensitive preamplifier and the other to the receiver's ground. The transmitting plates were adjusted to varying distances from the receiving plates. One possible source of error that must be considered when interpreting the results of this test is parasitic coupling of the signal to the wires connecting the sensor plates to the drive circuit. To verify that their effects were minimal the transmitting plates were removed and the connecting wires were left in place. In this configuration a signal was still received, though it was much smaller than with the plates attached. The signal was measured to be approximately 15mV peak-to-peak. From this it can be concluded that the effect of the wires is marginal compared to the scale of the measured signals.

The range of transmitted slew rates measured at the mini-transmitter is shown in Figure 4.5. Note the large change in slew rate between the second and third setting. As expected, when loaded by just free air the transmitter produced waveforms with slew rates comparable to simulation.

Figure 4.6 shows the measured waveforms at the receiver after the gain stage for the various slew rates. These tests demonstrated that slew-rate has a minimal effect on the amplitude of the received signal when transmitting through air. See section 4.3.3 for a complete comparison of these measurements made using a saline Load.

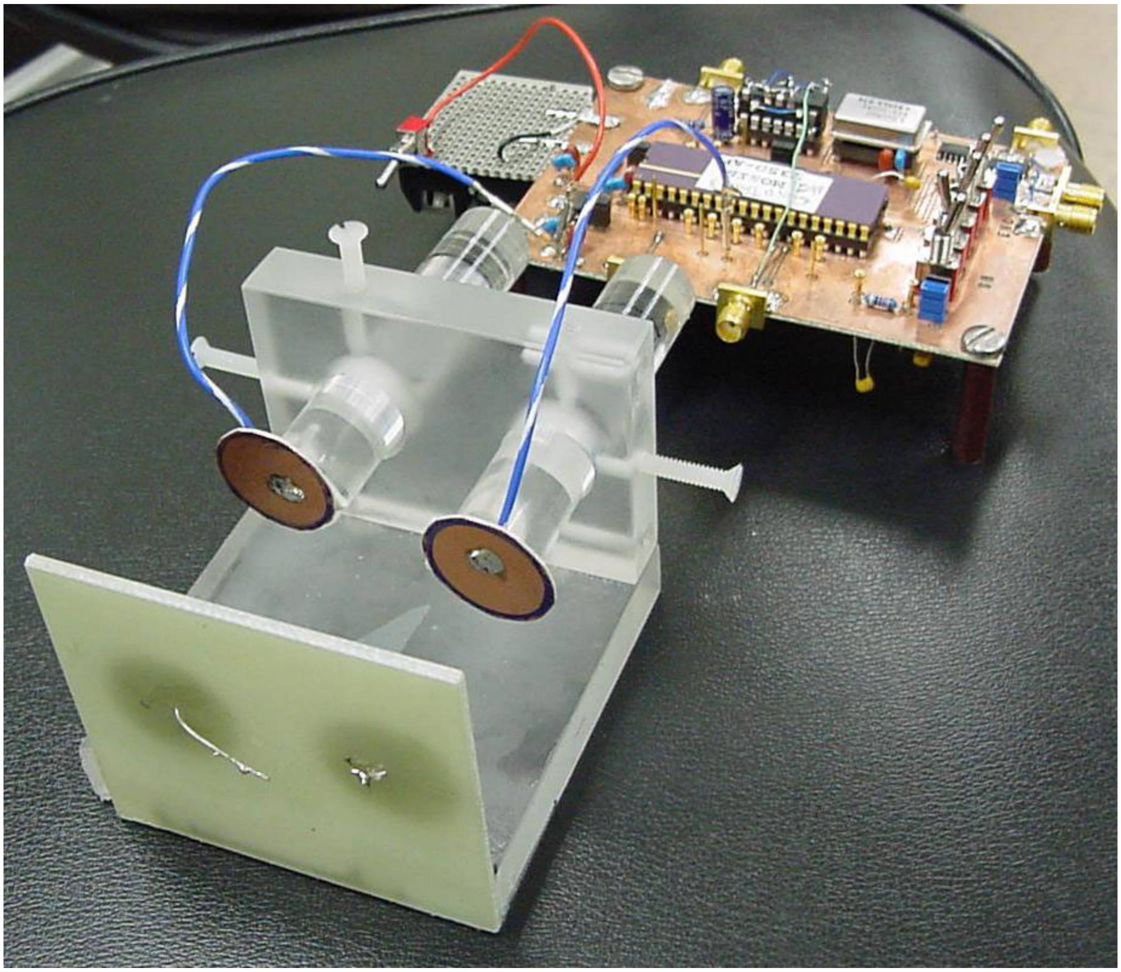


Figure 4.4: Free-air test setup

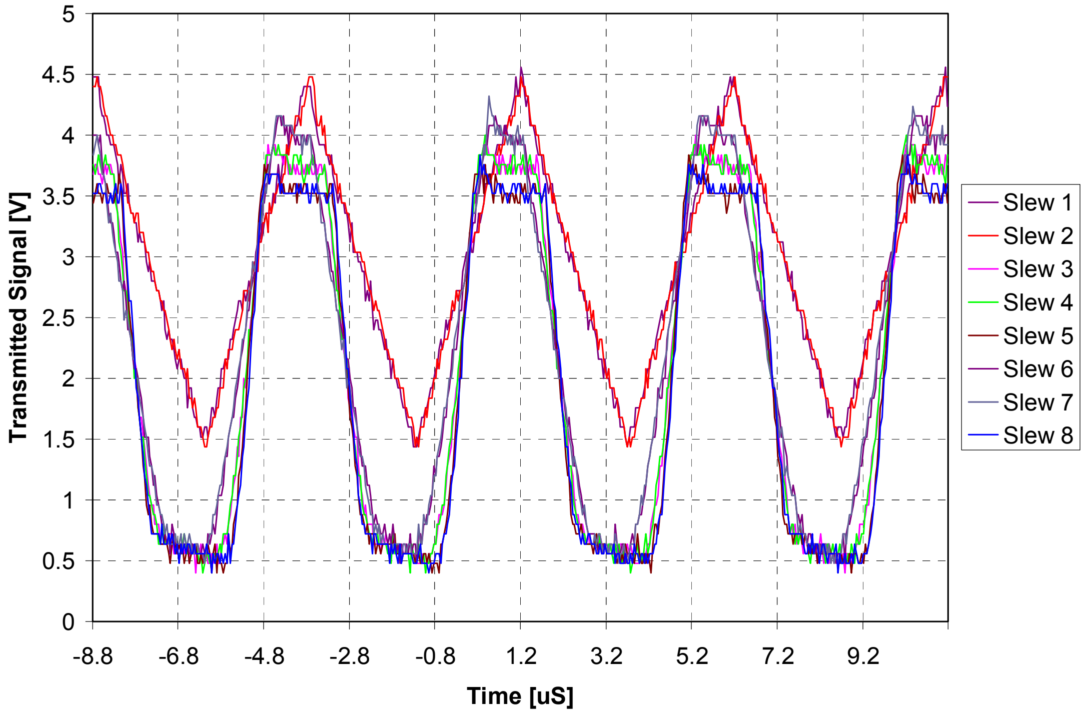


**Figure 4.5: Measured transmitter output waveforms showing adjustable slew range when loaded by free-air**

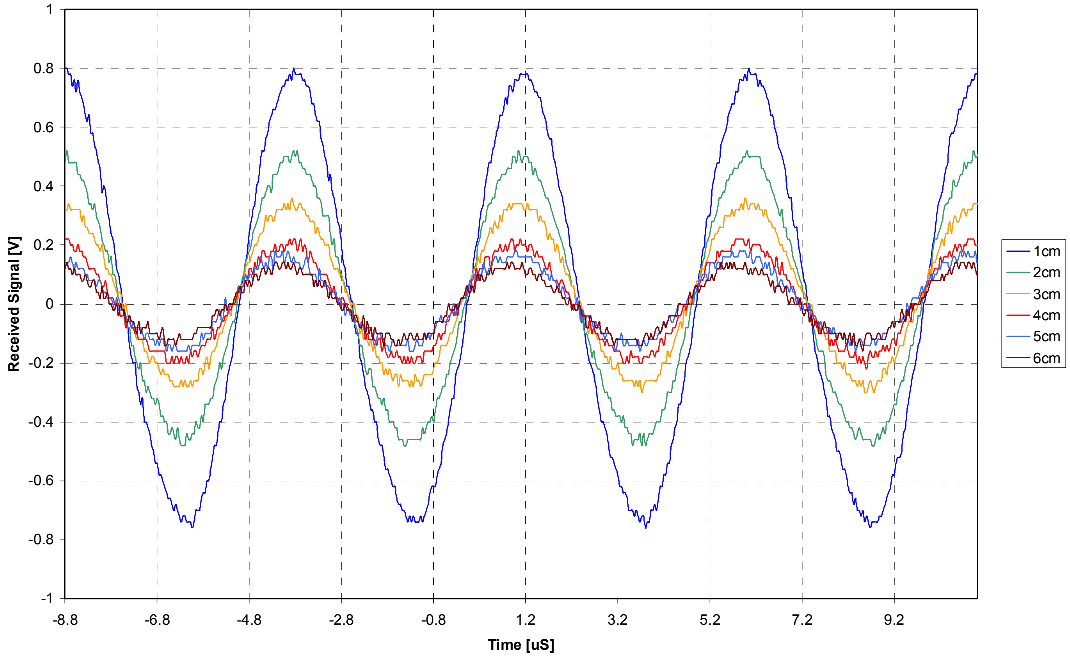


**Figure 4.6: Measured received waveforms for various slew-rates in free-air**

### 4.3.2 Transmission Through Tissue

The test setup for transmitting through tissue is shown in Figure 4.7. This test consisted of the free air test setup, but with a forearm inserted into the testing apparatus. This was an interesting test in that beyond just providing empirical results, this test offered useful insights into the actual transmission properties of the link. Unlike free air or saline, with this test repeatable results were difficult to obtain. Arm orientation, sensor to skin pressure, and the particular region on the arm chosen played a large role in affecting received signal strength. Transmitting through regions with a large amount of hair produced particularly low signal strengths, probably due the hair preventing adequate coupling with the skin. Testing on the lower, thinner section of the forearm also produced better results than the upper, thicker region.

Figure 4.8 shows the worst-case measured waveforms at the receiver after the gain stage for the various slew rates when transmitting through tissue. These signals are presented for two reasons. First, this data indicates that even in the worst-case scenario successful data transmission is possible. Second, obtaining repeatable worst-case signals proved to be significantly easier than obtaining repeatable best-case signals, although the received signal still did not maintain stable amplitude but instead had a slight amplitude modulation. This can be seen in the waveform for the first slew-rate setting. This setting did not have better performance than the other settings, but was instead captured at the peak of the modulation to show the range of amplitude fluctuation. This amplitude modulation is caused by environmental noise at the input of the charge-sensitive preamplifier, and is dominated by 60 Hz pick-up noise.

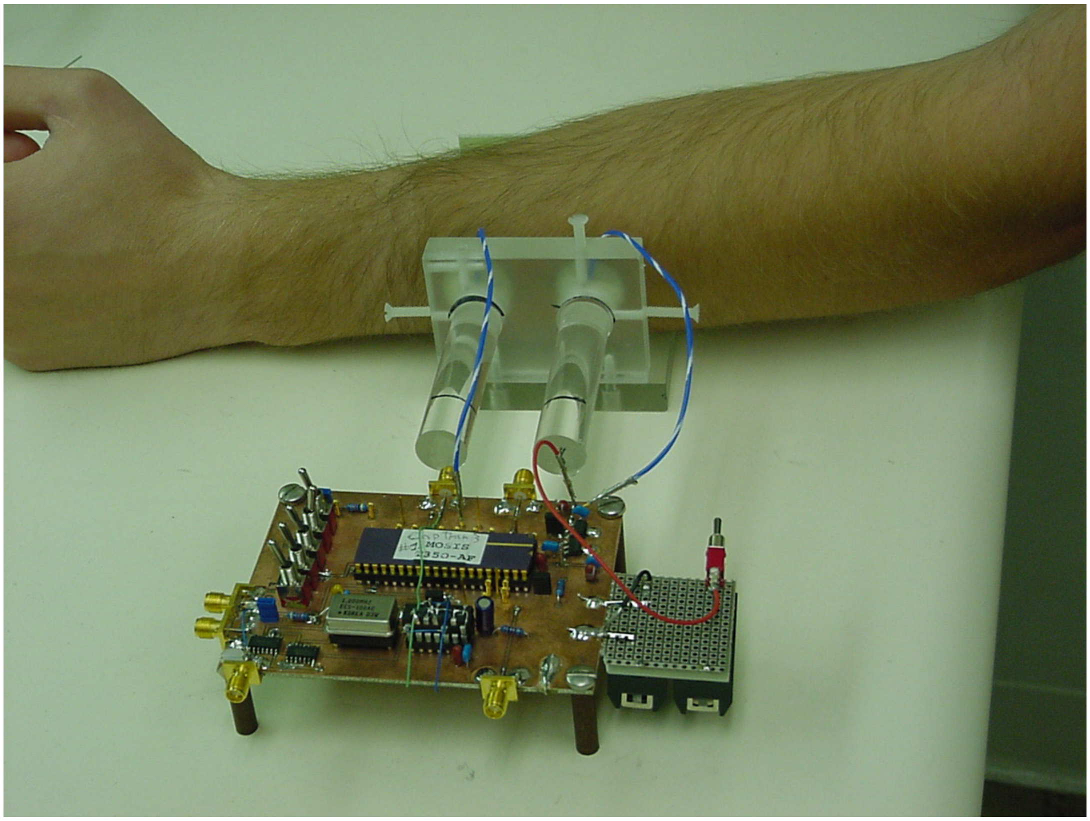


**Figure 4.7: Transmission through tissue test setup**

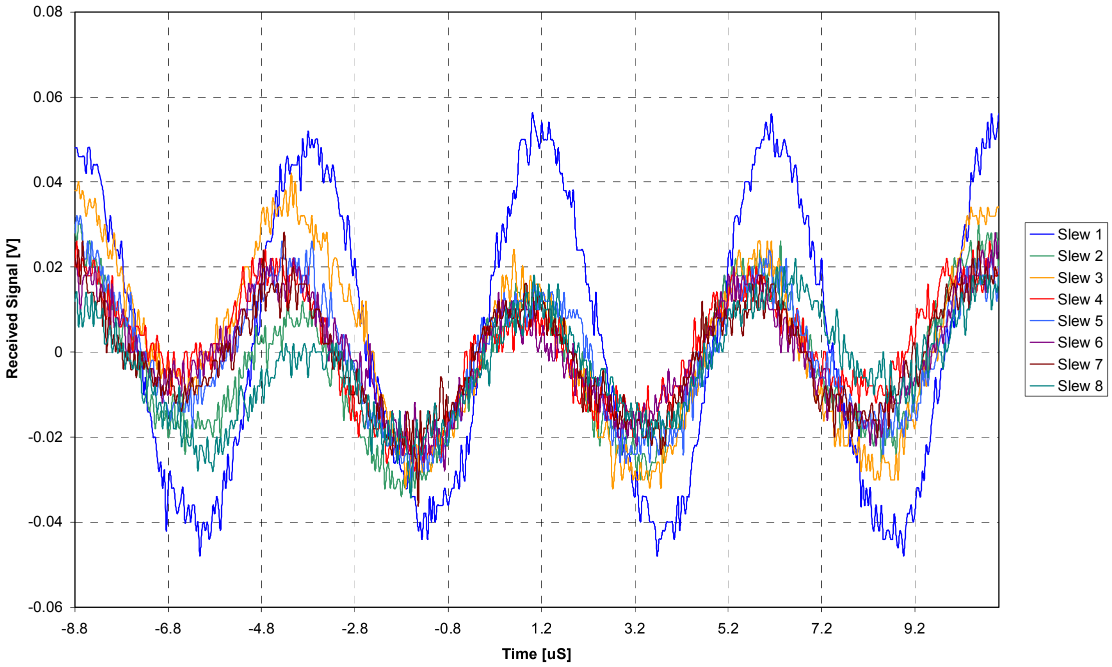


**Figure 4.8: Measured worst-case through-tissue received signals**

### 4.3.3 Partially Submerged Saline Bath

The third waveform test performed involved loading the transmitter with saline. This test was performed by submerging two plates in a saline bath; one plate was connected to the output driver and the other to the transmitter's ground. These plates were held at a fixed distance of 4 centimeters from the receiving plates. All plates were electrically isolated from the saline solution. This test setup is shown in Figure 4.9.

This test produced both useful empirical results and insights. What was particularly interesting about this test was that plate alignment and distance did not produce noticeable changes in signal strength of the received waveform. However, the transmitted signal was passing through the saline medium because removing the plates from the saline produced no received signal. This test produced two useful empirical results. The first of these was determining that the transmitter could successfully transmit into a saline load with minimal effect on the controlled slew rate. Figure 4.10 shows the waveforms of the transmitted signal as measured at the transmitter. Note they are approximately the same as transmitting through free air.

The second useful empirical result is comparing what effect slew rate has on the received signal when transmitting through a saline load. Figure 4.11 shows the received signals for the various slew rates after the gain stage on the receiver. Ideally, using Figure 4.10 and Figure 4.11 a comparison could be made to the free air testing comparing received signal strengths for the different load types with the same plate separation distances. However, because this test produced distance insensitive results, this comparison does not offer any feasible data.

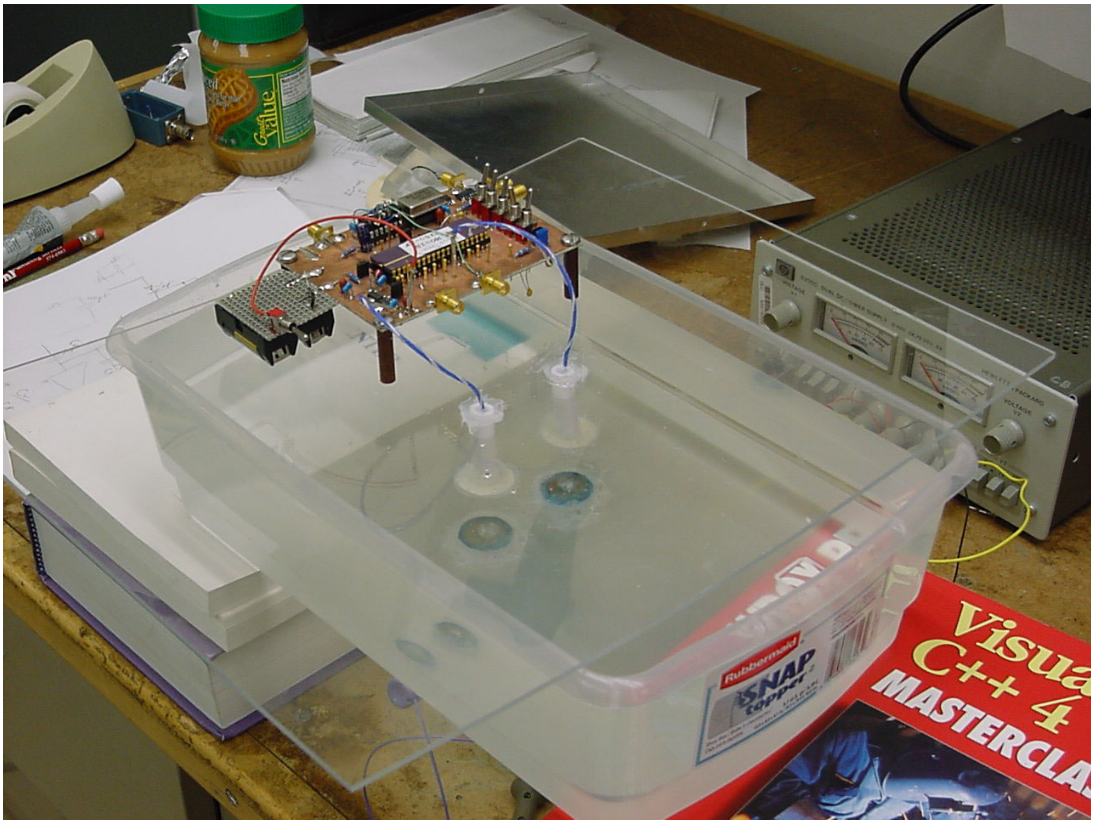


**Figure 4.9: Partially submerged saline bath test setup**

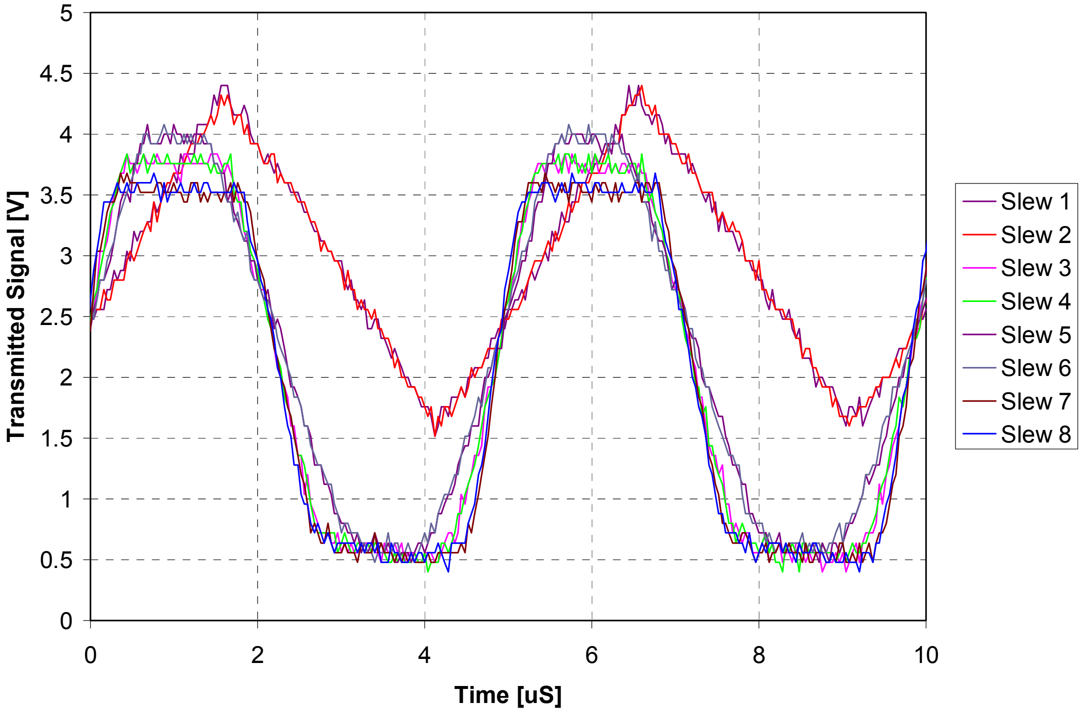


**Figure 4.10: Measured transmitting waveforms demonstrating output slew range when loaded by saline**

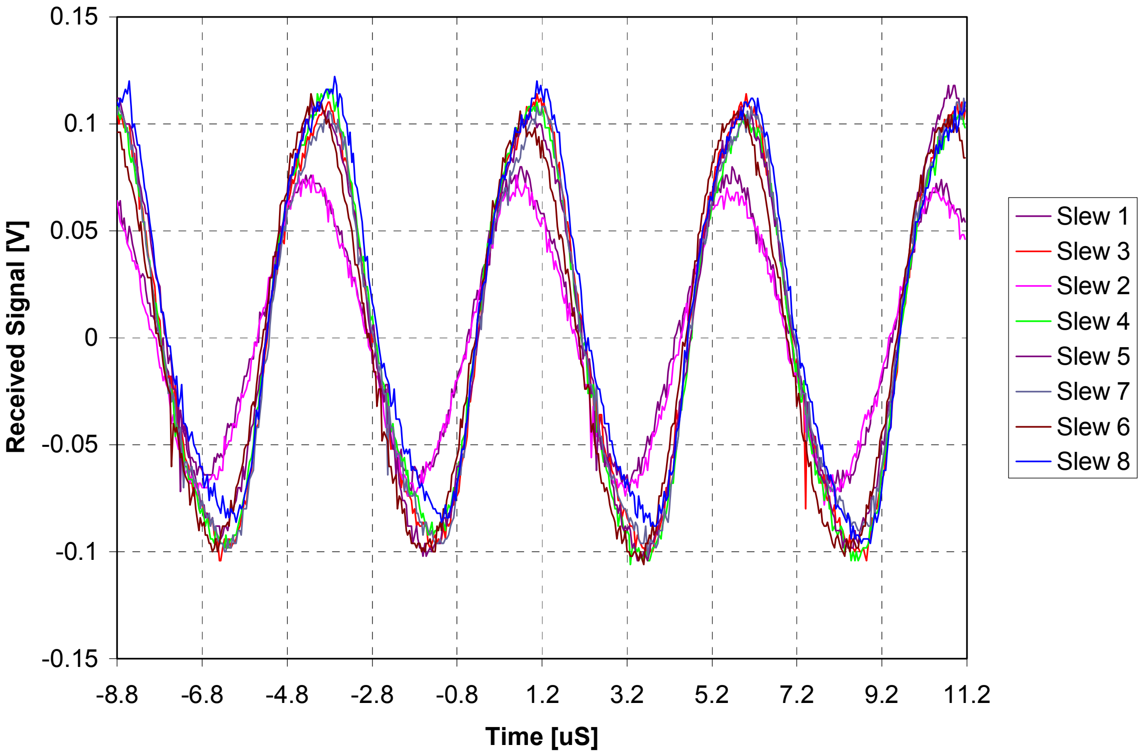


**Figure 4.11: Received waveforms for various slew rates in partially submerged saline bath**

What this test does offer is insight into how the slew rate affects the received signal strength. Because transmitted signal strength varies with slew rate due to the output driver configuration, the effect of slew-rate must be analyzed indirectly by comparing the ratio of transmitted signal amplitude versus received signal amplitude at each different slew rate. Figure 4.12 shows these results, and compares them to the results of the free air test.

Two conclusions can be made from the results of these tests. The first is affirmation that the drive circuit can successfully transmit into a saline load while maintaining a fixed slew rate. The second is that for the tests performed the slew rate has a marginal effect on the received signal strength beyond the reduced output swing associated with limiting the bias current of the driving operational amplifier. The idea behind transmitting with a slower slew rate comes out of the information presented in Chapter 2. When transmitting through tissue, higher frequency signals theoretically have a lower depth of penetration for a fixed transmitting power versus lower frequency signals, thus faster slew-rates with higher frequency components should also have a lower depth of penetration. In explaining the results of these tests however, the conclusion that must be made is that the range of slew rates tested was not large enough to noticeably change the received signal strength.

### 4.3.4 Fully Encapsulated Tests

The final link evaluation test was to fully encapsulate the mini-transmitter in a medium, simulating the actual implantation of a device. For these tests only the tail-current controlled slew rate limited op-amp was used.

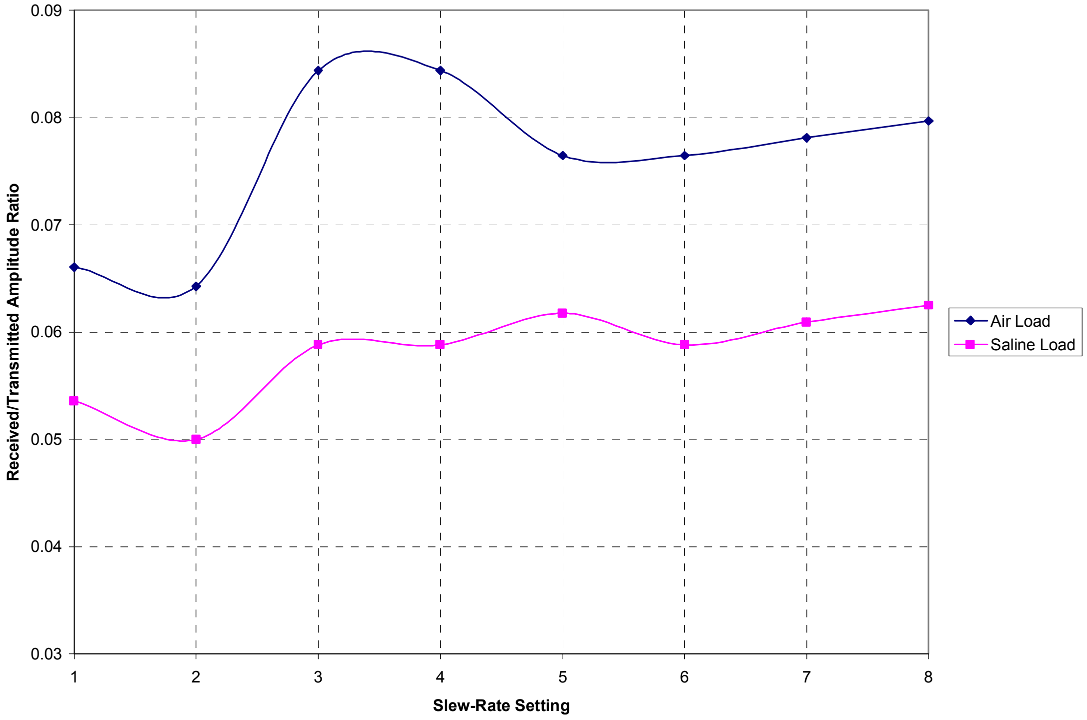


**Figure 4.12: Ratio of received signal amplitude to transmitted signal amplitude for free-air load and saline load**

Only one slew rate (slew rate “6”) was employed as adjusting the slew rate once the mini-transmitter was sealed is not practical. Originally saline was planned as the only implant medium, but poor results prompted the use of ground beef as a more accurate tissue simulator. Figure 4.13 shows the preliminary saline test setup used. This test setup was very similar to the test setup used for the initial testing detailed in Chapter 2.

The first testing was performed with all electrodes insulated from the saline. This configuration was chosen because it is the optimum configuration for an implantable device in terms of ease of packaging and optimized biocompatibility. The promising results from the partially submerged saline bath test setup as described in section 4.3.3 were achieved using this configuration. However, this insulated configuration when tested via full encapsulation in saline produced no discernable signal even at minimum distances. In an effort to determine the suitability of saline as a tissue simulator in this approach, a test setup encapsulating the mini-transmitter in ground beef was constructed. This test setup is shown in Figure 4.14. This test setup also produced no discernable signal when the fully insulated configuration was used.

In the next test the ground-return electrodes were exposed on both the transmitter and receiver, as was the case on the early testing performed for Chapter 2. In this configuration a discernable received signal was observed using the ground beef encapsulation test setup but not the saline encapsulation test setup. These waveforms are presented in Figure 4.15. However, even this setup proved to be significantly more unstable and limited compared to free-air. Compared to free-air, the range in which a signal could be successfully transmitted was quite short, limited to two centimeters.

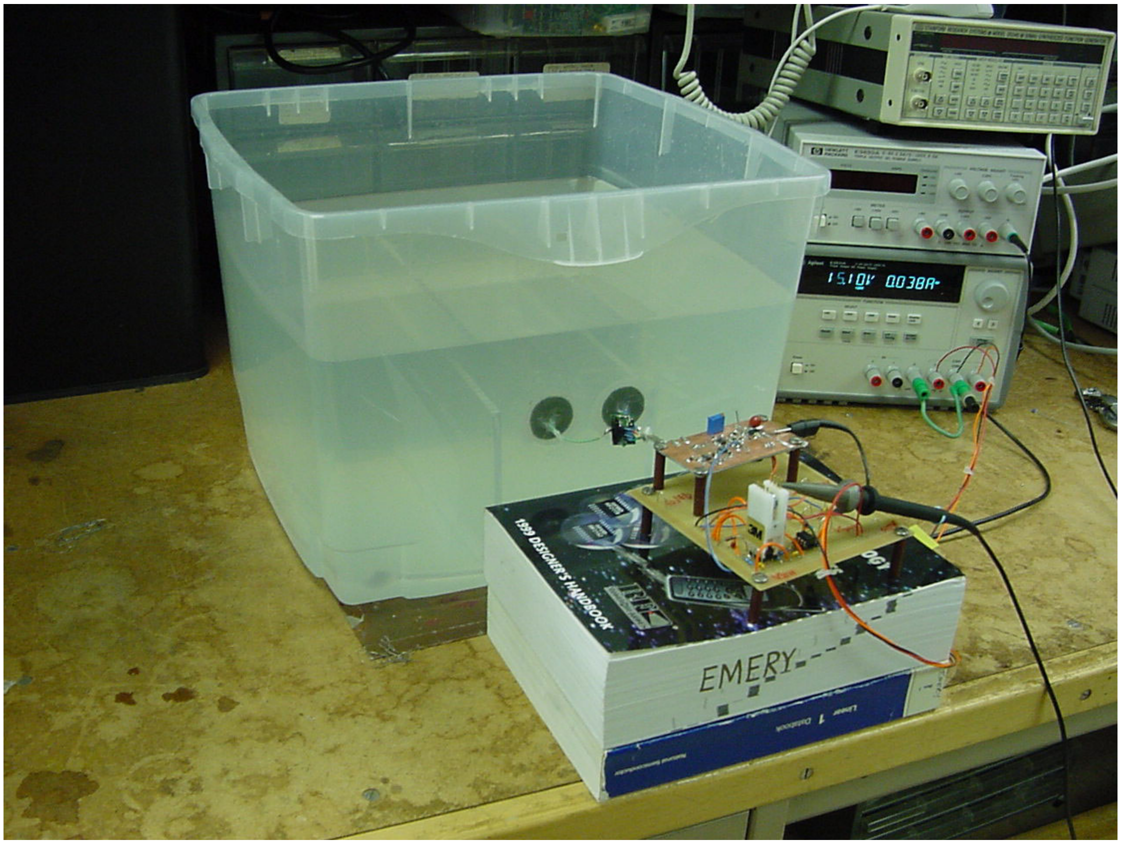


**Figure 4.13: Saline encapsulation test setup**

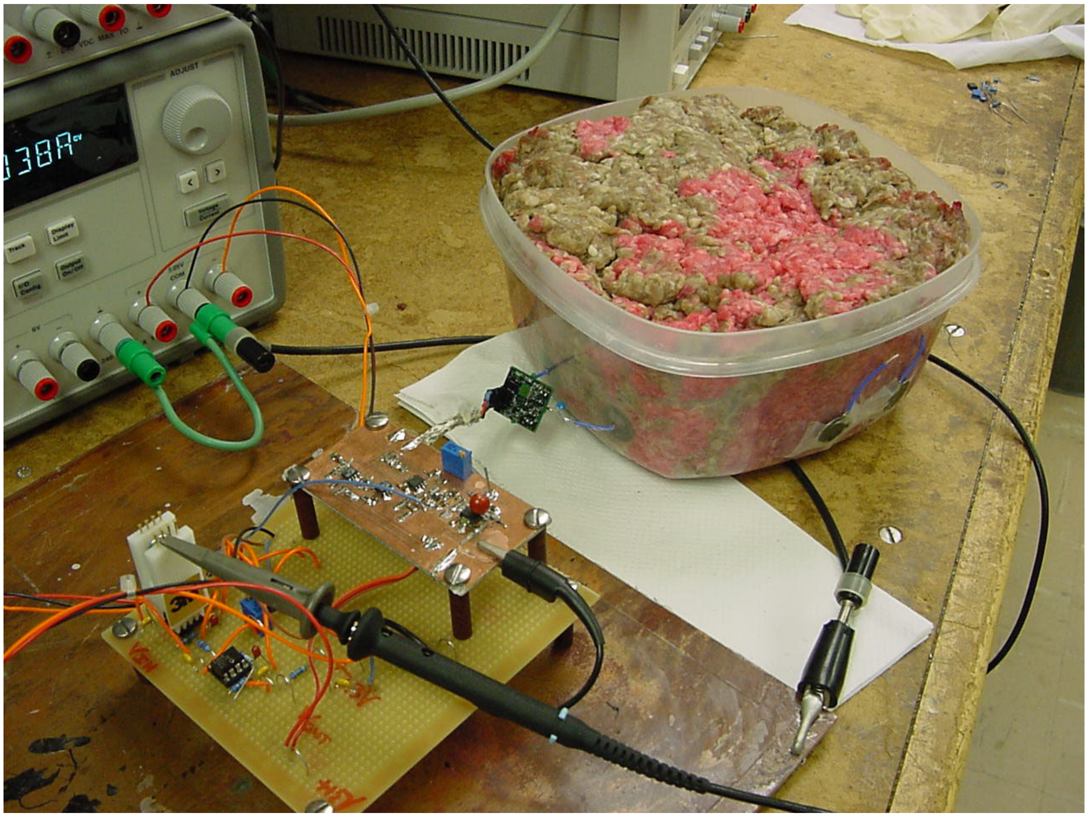


**Figure 4.14: Tissue encapsulation test setup using ground beef**

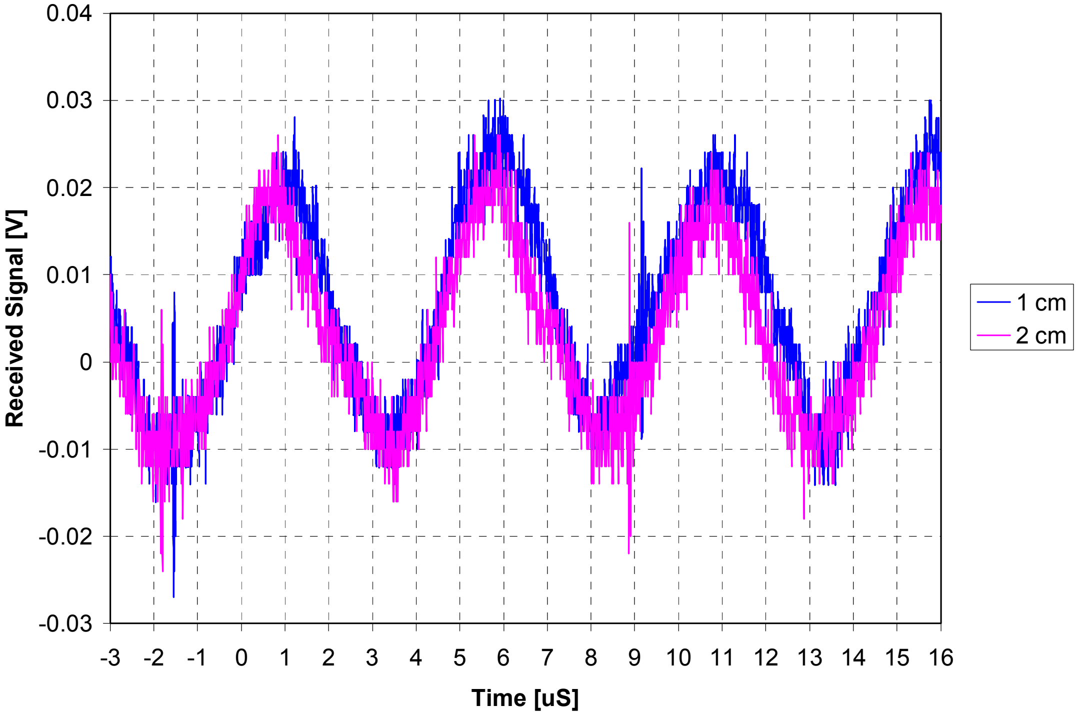


**Figure 4.15: Received waveforms for the fully encapsulated in tissue test setup**

Also, as shown in Figure 4.15, there was not a significant decrease in amplitude at different distances until approximately two centimeters, at which point there was no successful received signal. Part of the lack of amplitude dependence can be attributed to the unreliability of the link with this test setup. Unlike the free-air test setup, this setup was extremely susceptible to errors caused by misalignment and positioning; small changes would result in large variations in the received signal.

## 4.4 Statistical Reliability Measurements

The statistical reliability measurements were performed using the complete system, including the LabVIEW program described in sections 3.3.3 and 3.5.4. This program counts the number of packets received until an incorrect packet is received. To perform an accurate statistical analysis of the link, this test was repeated ten times at various distances. This test was performed first on the free-air test setup and then on the tissue encapsulated setup.

As noted in section 0, a successful link could not be established beyond 2 centimeters in the encapsulated tissue setup, so only results at 1 and 2 centimeters are presented for this test. In free-air, transmission was successful over the full range of test distances to the maximum test distance of 6 centimeters. Figure 4.16 shows the average number of successful packets before failure for each distance tested in both test setups. Examining the results from these tests reveals two trends. First is, as expected the average number of successful packets before failure decreases with distance in the free-air test setup.

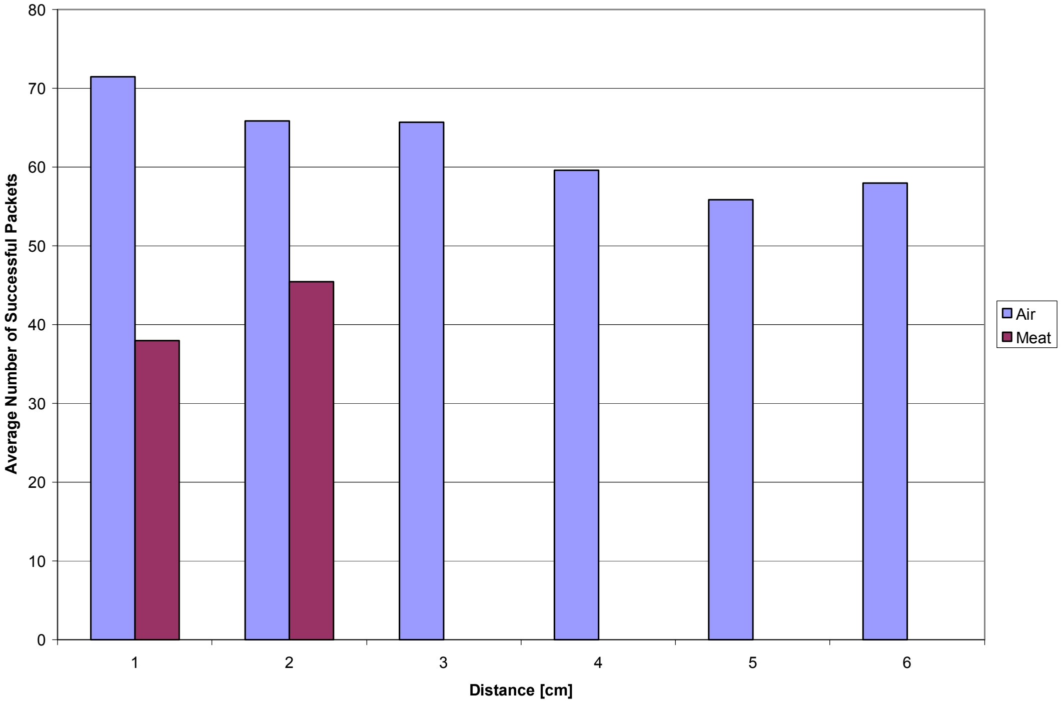


**Figure 4.16: Average number of successful packets received for various distances, as tested in free-air and fully encapsulated in tissue**

It is also apparent that the distance limitation imposed by the free-air test setup is not optimum for this type of testing. From work performed in free-air capacitive communication, it has been established that data has been successfully transmitted up to approximately 1 meter [33]. While this system has not been optimized for free-air transmission over long distances, test results indicate that link distances beyond 6 centimeters could be established and a better reliability trend established with further system optimization.

The second trend is from the results of the tissue encapsulated test setup. As was noted in section 0, the received signal amplitude did not vary with distance in this test setup. This is reflected in the statistical results. Indeed, instead of decreasing with distance the average number of successful packets actually increased slightly. This increase, however, probably indicates a higher margin of error for this test setup rather than a more stable link.

## 4.5 Additional Findings and Discussion of Limitations

The previous sections highlighted the direct results of the various tests performed, but there are several additional points of interest found through these tests that where not directly introduced. This section presents these points and also provides a brief summary of limitations for implementing this communication technique in an implanted application. It is important to note that the testing in this chapter served two purposes: evaluating the performance of the system, and evaluating the feasibility of this system for use in an implanted application. Ideally these purposes would be mutually exclusive (i.e.

the performance of the test system should ideally not be a limiting factor in determining application feasibility) but in the real world weaknesses in the system implementation are perceived as weaknesses in the feasibility of this type of communication for an implanted application. Thus care must be taken not to confuse limitations of the technique with weaknesses in the implemented system.

### 4.5.1 Environmental Noise

The greatest external noise contribution in this system was environmental noise. This noise is the greatest limitation of this system. Early testing showed the charge-sensitive preamplifier was extremely sensitive to several sources of environmental noise, but most notable noise produced by Cathode Ray Tubes (CRTs) and fluorescent light fixtures. In an effort to minimize this noise, all testing was performed in an environment with no CRTs or fluorescent lighting present. Substantial environmental noise was still observed however. In several of the previous tests references are made to a 'minimum discernable signal'. The hysteresis of the discriminator is 20mV, so theoretically the minimum discernable signal by the receiver should be approximately 20mV after the gain stage. Environmental noise, however, easily kept this from being realistic. This noise was different depending on the conditions, but a good example of common environmental noise is given in Figure 4.17. Unfortunately this noise cannot be easily filtered as it lies in the bandwidth of the transmitted signal. Ultimately this noise does not represent a limitation of this communication technique but is instead a weakness in the implemented system as improvements could be made to improve the noise immunity. Chapter 5 details some of these suggestions.

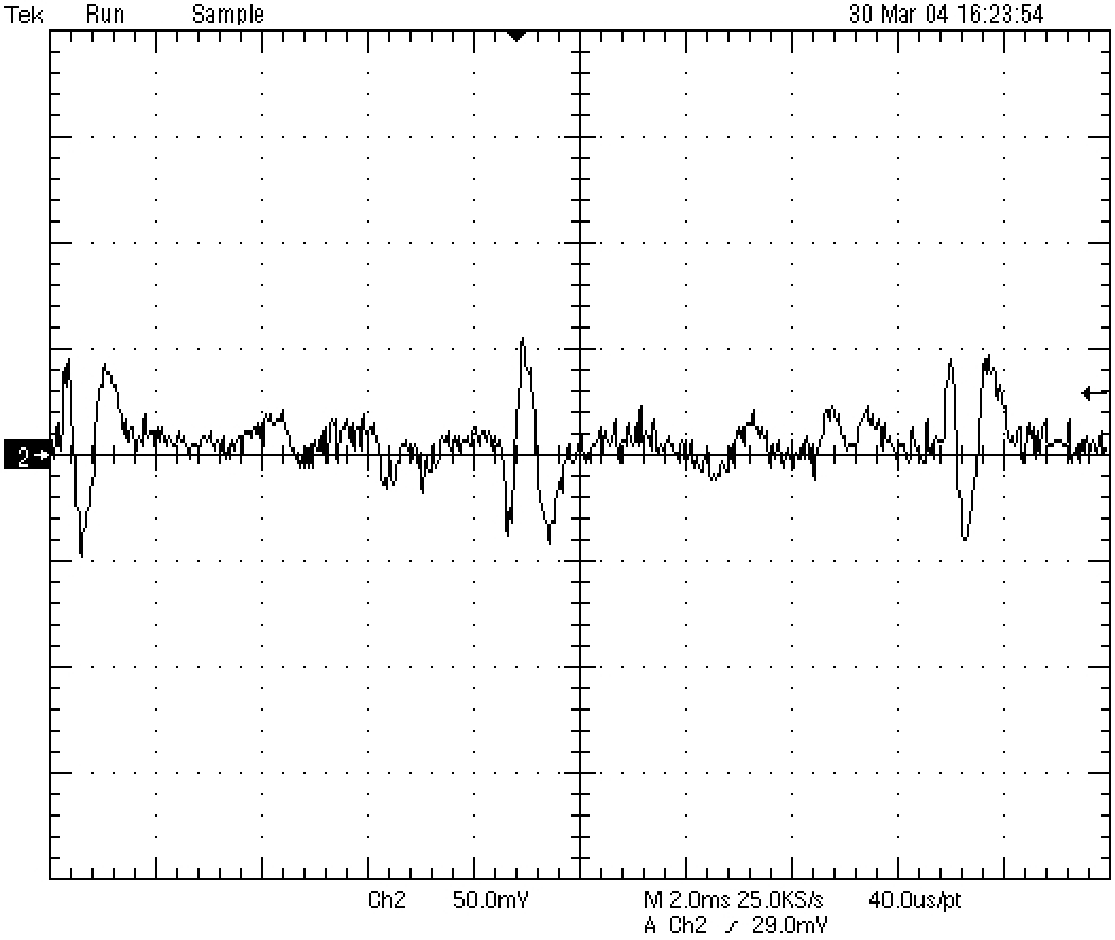


**Figure 4.17: Common environmental noise as seen at the output of the receiver's gain stage with no transmitted signal present**

### 4.5.2 Differential Drive Technique

All testing performed in this chapter used a single-ended drive configuration, where one electrode was driven and the other used as a ground-return path. Both the transmitter and receiver were designed, however, with differential transmission capabilities. This was fully implemented on both the transmitter functionality test board and the receiver board. Theoretically the differential drive technique should provide an effective method of doubling the amplitude of the transmitted signal. As was found in the tests documented earlier in this chapter, increasing the transmitted signal amplitude produces an increase in the received signal amplitude and provides for an increased distance over which a link can be successfully established. From previous work [34], it has been shown to be effective in free-air capacitive communication systems.

Unfortunately when this system with the differential drive configuration in an encapsulated environment no discernable signal was received. The problem lies in that in this configuration there were no exposed plates, as exposing either drive plate would directly couple current into the medium. Evaluating this technique further with exposed mid-point plates (thus using three transmitting plates and three receiving plates) was not performed, but may offer a successful link at the cost of additional plates.

### 4.5.3 Plate Orientation and Alignment

Although no specific empirical testing of plate alignment and orientation was performed, several conclusions were drawn from the various tests performed. From these tests, it was found that each test setup could be classified into one of three types of configurations, and that plate alignment and orientation affected the performance of each of these configurations differently.

In the first configuration all plates are in a free-air environment with no contact with any other medium and no medium other than free-air separating the plates. In this test configuration alignment affected the received signal strength in a fashion very similar to an increase in distance. For example, shifting the plates 1 centimeter off-axis produced approximately the same result as moving the plates 1 centimeter farther apart. In terms of orientation, one particular finding of interest was that relatively large signals could be received even if the plates were oriented exactly out of phase with each other (i.e. the transmitting plate oriented towards receiving ground return plate and the transmitter ground return plate oriented towards the receiving plate). This type of configuration has been used in the past to establish communication links via capacitive coupling through free-air [34].

In the second type of configuration the transmitter is situated in a free-air medium, but the transmitting plates are in contact with a lossy medium. An example of this configuration is both the partially submerged saline bath test setup and the non-encapsulated transmission through tissue test setup. In this setup plate orientation and alignment does not affect the received signal strength. Similarly, plate separation distance does not affect the received signal strength. Previous work leveraged this advantage create a novel communication scheme for various skin-mounted devices [25].

In the third type of configuration the transmitter and both transmitting plates are fully encapsulated in a lossy medium. Examples of this configuration are the test setups where the mini-transmitter was fully encapsulated in either saline or hamburger meat. In this configuration, plate orientation and alignment play a large role in received signal strength. Increasing the plates off-axis distance results in a drastic decrease of the received signal amplitude. Coupled with the limited transmitting distance when fully encapsulated, this implies that for an implanted application the depth must be relatively shallow.

### 4.5.4 Limitations for Implanted Applications

As mentioned previously, the purpose of the tests in this chapter was twofold: first to test the performance of the system, and second to test the limitation of applying this communication technique to an implanted application. In terms of the performance of the system, the tests produced positive results. Each component in the system was verified for full functionality, a complete communication link was successfully established from the transmitter to the receiver, and the mini-transmitter was successfully operated at a relatively low 27mW when continuously transmitting data.

Evaluating the feasibility of directly applying this technique to an implanted application, however, revealed two key limitations. The first of these is that the transmitter and receiver both must have exposed ground return electrodes. This is a significant limitation in terms of packaging for an implantable application. The second limitation is that when fully encapsulated, the transmission distance must be kept short (under 2 centimeters) compared to free-air transmission (tested to 6 centimeters with a longer distance very possible) or non-encapsulated skin-coupled applications (tested over a distance of approximately 1 meter). Even with these limitations, however, there may be applications where this communication technique is still attractive. One example of this is a retinal prosthesis where the transmission distance between the implant and a head-mounted receiver can be limited to less then two centimeters. Furthermore, in applications where the model is reversed and the implant acts as the receiver and the body-mounted device the transmitter, transmitting signal amplitude could be increased significantly as power consumption is less of a design issue, thus allowing greater implant depths. Chapter 5 expands upon additional techniques that may further reduce these limitations.

# Chapter 5 CONCLUSION AND FUTURE WORK

## 5.1 Conclusion

This thesis presents the design and implementation of a unique capacitance-based biotelemetry system for implantable applications, from theory to application. This system was implemented as a custom ASIC transmitter fabricated using the AMI 1.5µ process available through MOSIS. Encapsulated in biocompatible material, this transmitter encodes and transmits predetermined data packets at a controlled slew rate through tissue to a body-mounted receiver. The body-mounted receiver is made up of a charge sensitive front end with a discriminator to detect individual bits. An FPGA decodes the actual packets and transmits the data to a PC-based LabVIEW data collection program for statistical analysis of packet loss.

This system was tested and characterized using a variety of different test setups. Tests performed included power consumption, analysis of the waveform shape based on transmission media, slew rate, and distance, and finally statistical analysis of packet loss for a fixed system. These tests demonstrated a reliable link in free-air, but results when fully encapsulated in tissue or saline were less promising. For very short ranges a successful link was achieved when encapsulated in tissue but not in saline.

This novel system offers some competitive advantages versus standard RF based approaches. Because it is a baseband approach the silicon real estate used is minimal and no off-chip passive components are uses. Power consumption proved to be comparable to RF approaches as well. There are, however, several areas that could be improved, possibly allowing successful data transmission beyond the current maximum shallow implanted depth.

## 5.2 Future Work

The two foremost areas for potential future work are further testing and improvements to the system. The following subsections detail these areas.

### 5.2.1 Further Testing

One of the further tests that could be performed is *in vitro* animal testing. This testing would give verification that the system works reliably through complex tissues. Particularly, performing this testing at different locations and alignments both internally and externally and through different tissue thicknesses would offer further insight to the real-world functionality.

A second future test is to optimize the transmission frequency, which is a strong function of the surrounding tissue composition. One way to implement this with the current hardware would be to use the existing output driver in conjunction with a frequency generator and spectrum analyzer. Loading the output driver with a saline bath, you could then drive the output stage while varying the input frequency and measure the received signal strength after the charge-sensitive pre-amplifier with the spectrum analyzer. The limitation here would be the bandwidth of the output driver and testing beyond this limit would require a redesigned output stage. Performing a similar test *in vitro* using an animal subject could expand this.

### 5.2.2 System Improvement

The main area where the system could be improved is reducing the affect of environmental noise seen by the charge-sensitive pre-amplifier. Through testing it was found that the charge-sensitive pre-amplifier was particularly sensitive to both noise from CRTs and noise generated by florescent light fixtures. This problem is aggravated by the fact that much of this noise is in the bandwidth of the transmitted signal. The solution when testing was to turn off all CRTs and florescent lights near the test area, but this is not feasible for most biomedical approaches. Moving to a higher transmission frequency (the tested frequency was approximately 202 kHz) and adding a band pass filter could eliminate a large portion of this environmental noise.

In conjunction with this, another improvement would be to switch from a baseband approach to a modulated approach. Because modulated approaches can operate with much lower signal-to-noise ratios then the current baseband approach, moving to a modulated approach should increase the maximum implantable depth considerably. In an attempt to maintain simplicity, one good modulation technique that could be implemented on-off keying (OOK). Implementing this technique would require the addition of an oscillator on the transmitting side while the receiver could be implemented using standard demodulation techniques. This approach could potentially reduce noise limitations enough to allow implementation using a lower voltage process, thus further improving power consumption.

# References

# References

[1] S. Tang, B. Smith, J. H. Schild, P. H. Heckman, “Data Transmission from an Implantable Biotelemeter by Load-Shift Keying Using Circuit Configuration Modulator,” *IEEE Transactions on Biomedical Engineering*, vol. 42, no. 5, pp. 524-528, 1995

[2] M. N. Ericson, B. L. Ibey, G. L. Cote, J. S. Baba, J. B. Dixon, M. S. Hileman, C. L. Britton, M. A. Wilson, “In vivo application of a minimally invasive oximetry based perfusion sensor,” *Proceedings of the 2nd Joint EMBS-BMES Conference*, vol. 3, pp. 1789-1790, 2002

[3] T. Akin, K. Najafi, R. M. Bradley, “A Wireless Implantable Multichannel Digital Neural Recording System for a Micromachined Sieve Electrode,” *IEEE Journal of Solid-State Circuits*, vol. 33, no. 1, pp. 109-118, 1998

[4] “Significant Advances in Pacing and Beyond [online],” Medtronic Inc., available at [URL : http://www.medtronic.com/corporate/1971_1980.html#significant], accessed April 2004

[5] J. P. Piella, “Energy Management, Wireless and System Solutions for Implantable Devices”, PhD. Thesis, Universitat Autònoma de Barcelona, 2001

[6] “High Fidelity Human Meshes [online],” Remcom Inc., available at [URL : http://www.remcom.com/xfdtd6/HiFi.htm], accessed April 2004

[7] K. R. Foster, H. P. Schwan, “Dielectric Permittivity and Electric Conductivity of Biological Materials,” Handbook of Biological Effects of Electromagnetic Fields, Second Edition, Chapter 1, CRC Press, Boca Raton, FL, 1995

[8] R. F. Clevland, J. F. Ulcek, “Questions and Answers about Biological Effects and Potential Hazards of Radiofrequency Electromagnetic Fields, 4th Edition,” Federal Communications Commission Office of Engineering and Technology, Bulletin no. 56, 1999

[9] L. Wang, T. B. Tang, E. Johannessen, A. Astaras, A. F. Murray, J. M. Cooper, S. P. Beaumont, D. R. Cumming, “An Integrated Sensor Microsystem for Industrial and Biomedical Applications,” *Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference*, vol. 2, pp. 1717-1720, 2002

[10] J. H. Schild, "A Mulitchannel Implantable Telemeter for Acquisition of Physiological Data," M.S. Thesis, Case Western Reserve University, 1988

[11] B. C. Towe, "Passive Biotelemetry Using Frequency Keying," *IEEE Transactions on Biomedical Engineering*, vol. 33, no. 10, pp 905-909, 1986

[12] K. W. Fernald, J. J. Paulos, B. A. Stackhouse, R. A. Heaton, "An Implantable Digital Telemetry Integrated Circuit Using an Automatic Resonant-Frequency Search Technique," *Digest of Technical Papers of the 39th International Solid-State Circuits Conference*, pp. 68-69, 1992

[13] M. Ghovanloo, K. Najafi, "A High-Rate Frequency Shift Keying Demodulator Chip For Wireless Biomedical Implants," *Proceedings of the 2003 International Symposium on Circuits and Systems*, vol. 5, pp. 45-48, 2003

[14] W. J. Heetderks, "RF Powering of Millimeter- and Submillimeter-Sized Neural Prosthetic Implants," *IEEE Transactions on Biomedical Engineering*, vol. 35, no. 5, pp. 323-327, 1988

[15] N. N. Donaldson, "Passive Signaling via Inductive Coupling," *IEE Medical and Biological Engineering and Computing*, vol. 24, pp. 223-224, 1986

[16] P. A. Neukomm, H. Kündig, H. Baggenstos, K. Zerobin, "Passive Telemetry by Absorption Modulation: a New Principle for Long-Term Transabdominal Monitoring of Pressure and EMG of the Uterus of Cows," *Proceedings of the 10th International Symposium on Biotelemetry*, pp. 488-496, 1988

[17] "Livestock ID [online]," Texas Instruments Inc., available at [URL : http://www.ti.com/tiris/docs/solutions/animal/livestock.shtml], accessed April 2004

[18] A. Neukomm, I. Roncoroni, D. Nanz H. H. Quick, "Passive E-field Telemetry: A New Wireless Transmission Principle in Minimally Invasive Medicine," *Proceedings of the 15th International Symposium on Biotelemetry*, pp. 609-617, 1999

[19] M. Hendry, "CDMA Overview [online]," available at [URL : http://www.bee.net/mhendry/vrml/library/cdma/cdma.htm], accessed April 2004

[20] N. Aydin, T. Arsland, D. R. S. Cumming, "Design and Implementation of a Spread Spectrum Based Communication System for an Ingestible Capsule,"

*Proceedings of the 2$^{nd}$ Joint EMBS-BMES Conference*, vol. 3, pp. 1773-1774, 2002

[21] M. N. Ericson, M.A. Wilson, G.L. Coté, J.S. Baba, W. Xu, M. Bobrek, C.L. Britton, M.S. Hileman, M.R. Moore, M.S. Emery, R. Lenarduzzi, "Implantable sensor for blood flow monitoring after transplant surgery," *Minimally Invasive Therapy and Allied Technologies*, vol. 13, no. 1, pp. 1-8, 2004 (invited)

[22] D. P. Lindsey, E. L. McKee, M. L. Hull, S. M. Howell, "A New Technique for Transmission of Signals from Implantable Transducers," *IEEE Transactions on Biomedical Engineering*, vol. 45, no. 5, pp. 614-619, 1998

[23] R. J. Drost, R. D. Hopkins, I. E. Sutherland, "Proximity Communication," *Proceedings of the 2003 IEEE Custom Integrated Circuits Conference*, pp. 469-472, 2003

[24] S. Yoneda, M. Hiramoto, A. Kanayama, S. Kaisha, "Capacitive Coupling Type Data Transmission Circuit for Portable Electronic Apparatus," U.S. Patent No. 4763340, 1988

[25] T. Zimmerman, "Personal Area Networks: Near-field Intrabody Communication," *IBM Systems Journal*, Vol. 35, No. 3&4, pp. 609-617, 1996

[26] "XFDTD Example: Phone SAR [online]," Remcom Inc, available at [URL : http://www.remcom.com/examples/exampleDetailUpdated.php?exampleID=67&thisProduct=XFDTD60], accessed April 2004

[27] C. H. Durney, H. Massoudi, M. F. lskander, Radiofrequency Radiation Dosimetry Handbook, 4th Edition, United States Air Force Research Laboratory Technical Report USAFSAM-TR-85-73, Brooks Air Force Base, TX, 1986

[28] "SAM-Phantom [online]," Microwave Consultants Limited, available at [URL : http://www.sam-phantom.com], accessed April 2004

[29] K. Kundert, "Modeling Dielectric Absorption in Capacitors, ver. 2b [online]," The Designers Guide Publishing, available at [URL : http://www.designers-guide.com/Modeling/da.pdf], accessed April 2004

[30] C. Sechen, A. Sangiovanni-Vincentelli, "The Timberwolf Placement and Routing Package," *IEEE Journal of Solid-State Circuits*, vol. 20, no. 2, pp. 510-522, 1985

[31] G. Miller, Modern Electronic Communication, 6th Edition, Prentice-Hall, Upper Saddle River, NJ, 1999

[32] R. Baker, H. Li, D. Boyce, CMOS Circuit Design, Layout, and Simulation, The Institue of Electrical and Electronic Engineers, New York, NY, 1998

[33] L. Lyten, M. Bijil, "System for Wireless Information transmission Between Two Different Rooms," U.S. Patent No. 5625883, 1997

# Appendices

# Appendix 1 – Source Listing for Dummy Packet Generator

**SENDER.VHDL**

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_arith.ALL;
USE ieee.std_logic_unsigned.ALL;

ENTITY sender IS PORT(
        osc:                                         IN std_logic;
        sender_out:                                  OUT std_logic := '0';
        sender_lock_out:                     OUT std_logic := '0';
    sender_quiet_out:                 OUT std_logic := '1';
        bi_mark_out:                         OUT std_logic := '0';
        bi_mark_out_dupe:                    OUT std_logic := '0'
        );
END sender;

ARCHITECTURE behavioral OF sender IS

        COMPONENT dummy_board
        PORT(
        osc_in:                                      IN std_logic;
        pll_lock_out:                        OUT std_logic := '0';
        output:                                      OUT std_logic := '0';
    quiet:                                           OUT std_logic := '0';
        pll_clock_out:                       OUT std_logic;
        track_clock_out:                     OUT std_logic);
        END COMPONENT;

        COMPONENT bimark
        PORT(
        clock:                                       IN std_logic;
        data_in:                                     IN std_logic;
        quiet_in:                                    IN std_logic;
        data_out:                                    OUT std_logic := '0');
        END COMPONENT;

        SIGNAL sender:                std_logic;
        SIGNAL sender_lock:           std_logic;
        SIGNAL sender_quiet:  std_logic;
        SIGNAL meg_clock:             std_logic;
        SIGNAL fast_clock_in: std_logic;
        SIGNAL bi_mark :              std_logic;

BEGIN

sender_unit : dummy_board PORT MAP (
               osc_in                        => osc,
               pll_lock_out           => sender_lock,
               output                        => sender,
               quiet                         => sender_quiet,
               pll_clock_out          => meg_clock,
               track_clock_out        => fast_clock_in);

encoder : bimark PORT MAP (
               clock                         => meg_clock,
               data_in                       => sender,
               quiet_in                      => sender_quiet,
               data_out                      => bi_mark);

bi_mark_out <= bi_mark;
bi_mark_out_dupe <= bi_mark;
sender_out <= sender;
```

```
sender_lock_out <= sender_lock;
sender_quiet_out <= sender_quiet;
END behavioral;
```

**DUMMY.VHDL**

```
--Dummy packet maker for CapTran1, with pll generated clock for use on Altera board
--
--Calum Johnson
--
--This block creates a dummy packet made up of a 32 bit all 1s header,
--a 0 start-bit, 8 data bits consisting of a snake-eyes counter, a 0
--stop bit, and a 32 bit all 1s footer.  Uses an on-board PLL for clock
--generation.
--

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_arith.ALL;
USE ieee.std_logic_unsigned.ALL;

ENTITY dummy_board IS PORT(
        osc_in:                              IN std_logic;
        pll_lock_out:                        OUT std_logic := '0';
        output:                              OUT std_logic := '0';
        quiet:                               OUT std_logic := '0';
        pll_clock_out:                       OUT std_logic := '0';
        track_clock_out:                     OUT std_logic := '0'
        );
END dummy_board;

ARCHITECTURE behavioral OF dummy_board IS

        COMPONENT clock
        PORT
        (
                inclk0        : IN  STD_LOGIC := '0';
                c0            : OUT STD_LOGIC ;
                c1            : OUT STD_LOGIC ;
                locked        : OUT STD_LOGIC
        );
        END COMPONENT;


        TYPE state IS (sleep,header, middle, footer);
        SIGNAL pause_cnt :            integer RANGE 0 TO 7 := 0;
        SIGNAL data :                 integer RANGE 0 TO 15 := 0;
        SIGNAL counter :              integer RANGE 0 TO 31 := 0;
        SIGNAL position :             state := header;
        SIGNAL data_bits :            std_logic_vector (3 downto 0);
        SIGNAL pll_clock :            std_logic;
        SIGNAL pll_lock :             std_logic := '0';


BEGIN

        onemeg_clock : clock PORT MAP (
                inclk0        => osc_in,
                c0            => track_clock_out,
                c1            => pll_clock,
                locked        => pll_lock
        );

        pll_clock_out <= pll_clock;
        pll_lock_out <= pll_lock;

        upcount: PROCESS( pll_clock ) BEGIN

         IF ( pll_clock'event AND pll_clock = '1' ) THEN
                IF pll_lock = '1' THEN
                        IF position = header THEN
                                quiet <= '0';
                                output <= '1';
                                counter <= counter + 1;
                                IF counter = 31 THEN
                                        counter <= 0;
                                        position <= middle;
                                END IF;
                        ELSIF position = middle THEN
                                quiet <= '0';
                                data_bits <= conv_std_logic_vector(data, 4);
                                CASE counter IS
                                        WHEN 0 => output <= '0';
                                        WHEN 1 => output <= data_bits(3);
                                        WHEN 2 => output <= data_bits(2);
                                        WHEN 3 => output <= data_bits(1);
                                        WHEN 4 => output <= data_bits(0);
                                        WHEN 5 => output <= data_bits(3);
                                        WHEN 6 => output <= data_bits(2);
                                        WHEN 7 => output <= data_bits(1);
                                        WHEN 8 => output <= data_bits(0);
                                        WHEN OTHERS =>  output <= '0';
                                                        counter <= 0;
                                                        data <= data + 1;
                                                        position <= footer;
                                END CASE;
                                counter <= counter + 1;
                        ELSIF position = footer THEN
                                quiet <= '0';
                                output <= '1';
                                counter <= counter + 1;
                                IF counter = 31 THEN
                                        counter <= 0;
                                        position <= sleep;
                                END IF;
                        ELSE
                                quiet <= '1';
                                output <= '0';
                                counter <= counter + 1;
                                IF counter = 31 THEN
                                        pause_cnt <= pause_cnt + 1;
                                        IF pause_cnt = 7 THEN
                                                counter <= 0;
                                                pause_cnt <= 0;
                                                position <= header;
                                        ELSE
                                                counter <= 0;
                                        END IF;
                                END IF;
                        END IF;
                                END IF;
        END IF; -- IF( pll_clock'event AND pll_clock = '1' )
    END PROCESS upcount;
END behavioral;
```

**BIMARK.VHDL**

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_arith.ALL;
USE ieee.std_logic_unsigned.ALL;

ENTITY bimark IS PORT(
       clock:                                      IN std_logic;
       data_in:                                    IN std_logic;
       quiet_in:                                   IN std_logic;
       data_out:                                   OUT std_logic := '0'
       );
END bimark;

ARCHITECTURE behavioral OF bimark IS

--SIGNAL to_out :                    std_logic := '0';
SIGNAL data_track :            std_logic := '0';
SIGNAL clock_track :   std_logic := '1';

BEGIN

data_out <= (NOT quiet_in) AND (clock_track XOR data_track);

rising_edge: PROCESS BEGIN
WAIT UNTIL (clock'event and clock = '1');
       clock_track <= NOT clock_track;
END PROCESS rising_edge;

falling_edge: PROCESS BEGIN
WAIT UNTIL (clock'event AND clock = '0' );
       IF data_in = '1' THEN
              data_track <= NOT data_track;
       END IF;
END PROCESS falling_edge;

END behavioral;
```

**CLOCK.VHDL**

```
-- megafunction wizard: %ALTCLKLOCK%
-- GENERATION: STANDARD
-- VERSION: WM1.0
-- MODULE: altpll

-- ============================================================
-- File Name: clock.vhd
-- Megafunction Name(s):
--                    altpll
-- ============================================================
-- ************************************************************
-- THIS IS A WIZARD-GENERATED FILE. DO NOT EDIT THIS FILE!
-- ************************************************************


--Copyright (C) 1991-2003 Altera Corporation
--Any megafunction design, and related netlist (encrypted or decrypted),
--support information, device programming or simulation file, and any other
--associated documentation or information provided by Altera or a partner
--under Altera's Megafunction Partnership Program may be used only
--to program PLD devices (but not masked PLD devices) from Altera. Any
--other use of such megafunction design, netlist, support information,
--device programming or simulation file, or any other related documentation
--or information is prohibited for any other purpose, including, but not
--limited to modification, reverse engineering, de-compiling, or use with
--any other silicon devices, unless such use is explicitly licensed under
--a separate agreement with Altera or a megafunction partner. Title to the
--intellectual property, including patents, copyrights, trademarks, trade
--secrets, or maskworks, embodied in any such megafunction design, netlist,
--support information, device programming or simulation file, or any other
--related documentation or information provided by Altera or a megafunction
--partner, remains with Altera, the megafunction partner, or their respective
--licensors. No other licenses, including any licenses needed under any third
--party's intellectual property, are provided herein.


LIBRARY ieee;
USE ieee.std_logic_1164.all;

LIBRARY altera_mf;
USE altera_mf.altera_mf_components.all;

ENTITY clock IS
        PORT
        (
                inclk0          : IN STD_LOGIC  := '0';
                c0              : OUT STD_LOGIC ;
                c1              : OUT STD_LOGIC ;
                locked          : OUT STD_LOGIC
        );
END clock;


ARCHITECTURE SYN OF clock IS

        SIGNAL sub_wire0      : STD_LOGIC_VECTOR (5 DOWNTO 0);
        SIGNAL sub_wire1      : STD_LOGIC ;
        SIGNAL sub_wire2      : STD_LOGIC ;
        SIGNAL sub_wire3      : STD_LOGIC ;
        SIGNAL sub_wire4_bv   : BIT_VECTOR (0 DOWNTO 0);
        SIGNAL sub_wire4      : STD_LOGIC_VECTOR (0 DOWNTO 0);
        SIGNAL sub_wire5      : STD_LOGIC_VECTOR (5 DOWNTO 0);
        SIGNAL sub_wire6_bv   : BIT_VECTOR (0 DOWNTO 0);
        SIGNAL sub_wire6      : STD_LOGIC_VECTOR (0 DOWNTO 0);
        SIGNAL sub_wire7      : STD_LOGIC ;
        SIGNAL sub_wire8      : STD_LOGIC_VECTOR (1 DOWNTO 0);
        SIGNAL sub_wire9        : STD_LOGIC_VECTOR (3 DOWNTO 0);



        COMPONENT altpll
        GENERIC (
                clk1_divide_by          : NATURAL;
                bandwidth_type          : STRING;
                clk1_phase_shift                : STRING;
                clk0_duty_cycle                 : NATURAL;
                lpm_type                : STRING;
                clk0_multiply_by                : NATURAL;
                lock_low                : NATURAL;
                invalid_lock_multiplier                 : NATURAL;
                inclk0_input_frequency          : NATURAL;
                gate_lock_signal                : STRING;
                clk0_divide_by          : NATURAL;
                clk1_duty_cycle                 : NATURAL;
                pll_type                : STRING;
                valid_lock_multiplier           : NATURAL;
                clk1_multiply_by                : NATURAL;
                clk0_time_delay                 : STRING;
                spread_frequency                : NATURAL;
                operation_mode          : STRING;
                lock_high               : NATURAL;
                compensate_clock                : STRING;
                clk1_time_delay                 : STRING;
                clk0_phase_shift                : STRING
        );
        PORT (
                        clkena  : IN STD_LOGIC_VECTOR (5 DOWNTO 0);
                        inclk   : IN STD_LOGIC_VECTOR (1 DOWNTO 0);
                        extclkena       : IN STD_LOGIC_VECTOR (3 DOWNTO 0);
                        locked  : OUT STD_LOGIC ;
                        clk     : OUT STD_LOGIC_VECTOR (5 DOWNTO 0)
        );
        END COMPONENT;

BEGIN
        sub_wire4_bv(0 DOWNTO 0) <= "0";
        sub_wire4    <= NOT(To_stdlogicvector(sub_wire4_bv));
        sub_wire6_bv(0 DOWNTO 0) <= "0";
        sub_wire6    <= To_stdlogicvector(sub_wire6_bv);
        sub_wire2    <= sub_wire0(1);
        sub_wire1    <= sub_wire0(0);
        c0    <= sub_wire1;
        c1    <= sub_wire2;
        locked    <= sub_wire3;
        sub_wire5    <= sub_wire6(0 DOWNTO 0) & sub_wire6(0 DOWNTO 0) & sub_wire6(0 DOWNTO
0) & sub_wire6(0 DOWNTO 0) & sub_wire4(0 DOWNTO 0) & sub_wire4(0 DOWNTO 0);
        sub_wire7    <= inclk0;
        sub_wire8    <= sub_wire6(0 DOWNTO 0) & sub_wire7;
        sub_wire9    <= sub_wire6(0 DOWNTO 0) & sub_wire6(0 DOWNTO 0) & sub_wire6(0 DOWNTO
0) & sub_wire6(0 DOWNTO 0);

        altpll_component : altpll
        GENERIC MAP (
                clk1_divide_by => 50,
                bandwidth_type => "AUTO",
                clk1_phase_shift => "0",
                clk0_duty_cycle => 50,
                lpm_type => "altpll",
                clk0_multiply_by => 16,
                lock_low => 5,
                invalid_lock_multiplier => 5,
                inclk0_input_frequency => 20000,
                gate_lock_signal => "NO",
                clk0_divide_by => 25,
                clk1_duty_cycle => 50,
                pll_type => "AUTO",
                valid_lock_multiplier => 1,
                clk1_multiply_by => 1,
                clk0_time_delay => "0",
                spread_frequency => 0,
                operation_mode => "NORMAL",
                lock_high => 1,
                compensate_clock => "CLK1",
                clk1_time_delay => "0",
                clk0_phase_shift => "0"
        )
        PORT MAP (
                clkena => sub_wire5,
                inclk => sub_wire8,
                extclkena => sub_wire9,
                clk => sub_wire0,
                locked => sub_wire3
        );



END SYN;

-- ============================================================
-- CNX file retrieval info
-- ============================================================
-- Retrieval info: PRIVATE: DIV_FACTOR9 NUMERIC "1"
-- Retrieval info: PRIVATE: USE_CLKENA2 STRING "0"
-- Retrieval info: PRIVATE: MIRROR_CLK0 STRING "0"
-- Retrieval info: PRIVATE: PHASE_SHIFT_UNIT0 STRING "deg"
-- Retrieval info: PRIVATE: INCLK1_FREQ_UNIT_COMBO STRING "MHz"
-- Retrieval info: PRIVATE: SPREAD_USE STRING "0"
-- Retrieval info: PRIVATE: SPREAD_FEATURE_ENABLED STRING "1"
-- Retrieval info: PRIVATE: GLOCKED_COUNTER_EDIT_CHANGED STRING "1"
-- Retrieval info: PRIVATE: GLOCK_COUNTER_EDIT NUMERIC "10000"
-- Retrieval info: PRIVATE: USE_CLKENA3 STRING "0"
-- Retrieval info: PRIVATE: MIRROR_CLK1 STRING "0"
-- Retrieval info: PRIVATE: PHASE_SHIFT_UNIT1 STRING "deg"
-- Retrieval info: PRIVATE: DUTY_CYCLE0 STRING "50.00000000"
-- Retrieval info: PRIVATE: PHASE_SHIFT0 STRING "0.00000000"
-- Retrieval info: PRIVATE: MULT_FACTOR0 NUMERIC "16"
-- Retrieval info: PRIVATE: SPREAD_PERCENT STRING "0.500"
-- Retrieval info: PRIVATE: LOCKED_OUTPUT_CHECK STRING "1"
-- Retrieval info: PRIVATE: PLL_ARESET_CHECK STRING "0"
-- Retrieval info: PRIVATE: USE_CLKENA4 STRING "0"
-- Retrieval info: PRIVATE: MIRROR_CLK2 STRING "0"
-- Retrieval info: PRIVATE: PHASE_SHIFT_UNIT2 STRING "deg"
-- Retrieval info: PRIVATE: DUTY_CYCLE1 STRING "50.00000000"
-- Retrieval info: PRIVATE: PHASE_SHIFT1 STRING "0.00000000"
-- Retrieval info: PRIVATE: MULT_FACTOR1 NUMERIC "1"
-- Retrieval info: PRIVATE: JUMP2PAGE0 STRING ""
-- Retrieval info: PRIVATE: TIME_SHIFT0 STRING "0.00000000"
-- Retrieval info: PRIVATE: STICKY_CLK0 STRING "1"
-- Retrieval info: PRIVATE: BANDWIDTH STRING "1.000"
-- Retrieval info: PRIVATE: BANDWIDTH_USE_CUSTOM STRING "0"
-- Retrieval info: PRIVATE: USE_CLKENA5 STRING "0"
-- Retrieval info: PRIVATE: MIRROR_CLK3 STRING "0"
-- Retrieval info: PRIVATE: PHASE_SHIFT_UNIT3 STRING "deg"
-- Retrieval info: PRIVATE: DUTY_CYCLE2 STRING "50.00000000"
-- Retrieval info: PRIVATE: PHASE_SHIFT2 STRING "0.00000000"
-- Retrieval info: PRIVATE: MULT_FACTOR2 NUMERIC "1"
-- Retrieval info: PRIVATE: JUMP2PAGE1 STRING "□"
-- Retrieval info: PRIVATE: TIME_SHIFT1 STRING "0.00000000"
-- Retrieval info: PRIVATE: STICKY_CLK1 STRING "1"
-- Retrieval info: PRIVATE: SPREAD_FREQ STRING "300.000"
-- Retrieval info: PRIVATE: BANDWIDTH_FEATURE_ENABLED STRING "1"
-- Retrieval info: PRIVATE: LONG_SCAN_RADIO STRING "1"
-- Retrieval info: PRIVATE: USE_CLKENA6 STRING "0"
-- Retrieval info: PRIVATE: MIRROR_CLK4 STRING "0"
-- Retrieval info: PRIVATE: PHASE_SHIFT_UNIT4 STRING "deg"
-- Retrieval info: PRIVATE: DUTY_CYCLE3 STRING "50.00000000"
-- Retrieval info: PRIVATE: PHASE_SHIFT3 STRING "0.00000000"
-- Retrieval info: PRIVATE: MULT_FACTOR3 NUMERIC "1"
-- Retrieval info: PRIVATE: JUMP2PAGE2 STRING "General/Modes"
-- Retrieval info: PRIVATE: TIME_SHIFT2 STRING "0.00000000"
-- Retrieval info: PRIVATE: STICKY_CLK2 STRING "0"
-- Retrieval info: PRIVATE: USE_CLK0 STRING "1"
-- Retrieval info: PRIVATE: INCLK1_FREQ_EDIT_CHANGED STRING "1"
-- Retrieval info: PRIVATE: SCAN_FEATURE_ENABLED STRING "1"
-- Retrieval info: PRIVATE: ZERO_DELAY_RADIO STRING "0"
-- Retrieval info: PRIVATE: PLL_PFDENA_CHECK STRING "0"
-- Retrieval info: PRIVATE: USE_CLKENA7 STRING "0"
-- Retrieval info: PRIVATE: MIRROR_CLK5 STRING "0"
-- Retrieval info: PRIVATE: PHASE_SHIFT_UNIT5 STRING "deg"
-- Retrieval info: PRIVATE: DUTY_CYCLE4 STRING "50.00000000"
-- Retrieval info: PRIVATE: PHASE_SHIFT4 STRING "0.00000000"
-- Retrieval info: PRIVATE: MULT_FACTOR4 NUMERIC "1"
-- Retrieval info: PRIVATE: JUMP2PAGE3 STRING "General/Modes"
-- Retrieval info: PRIVATE: TIME_SHIFT3 STRING "0.00000000"
-- Retrieval info: PRIVATE: STICKY_CLK3 STRING "0"
-- Retrieval info: PRIVATE: USE_CLK1 STRING "1"
-- Retrieval info: PRIVATE: CREATE_CLKBAD_CHECK STRING "0"
-- Retrieval info: PRIVATE: INCLK1_FREQ_EDIT STRING "50.000"
-- Retrieval info: PRIVATE: CUR_DEDICATED_CLK STRING "c1"
-- Retrieval info: PRIVATE: PLL_FASTPLL_CHECK NUMERIC "0"
-- Retrieval info: PRIVATE: USE_CLKENA8 STRING "0"
-- Retrieval info: PRIVATE: MIRROR_CLK6 STRING "0"
-- Retrieval info: PRIVATE: PHASE_SHIFT_UNIT6 STRING "deg"
-- Retrieval info: PRIVATE: DUTY_CYCLE5 STRING "50.00000000"
-- Retrieval info: PRIVATE: PHASE_SHIFT5 STRING "0.00000000"
-- Retrieval info: PRIVATE: MULT_FACTOR5 NUMERIC "1"
-- Retrieval info: PRIVATE: JUMP2PAGE4 STRING "General/Modes"
-- Retrieval info: PRIVATE: TIME_SHIFT4 STRING "0.00000000"
-- Retrieval info: PRIVATE: STICKY_CLK4 STRING "0"
-- Retrieval info: PRIVATE: USE_CLK2 STRING "0"
-- Retrieval info: PRIVATE: ACTIVECLK_CHECK STRING "0"
-- Retrieval info: PRIVATE: BANDWIDTH_FREQ_UNIT STRING "MHz"
-- Retrieval info: PRIVATE: INCLK0_FREQ_UNIT_COMBO STRING "MHz"
-- Retrieval info: PRIVATE: USE_CLKENA9 STRING "0"
-- Retrieval info: PRIVATE: MIRROR_CLK7 STRING "0"
-- Retrieval info: PRIVATE: PHASE_SHIFT_UNIT7 STRING "deg"
-- Retrieval info: PRIVATE: DUTY_CYCLE6 STRING "50.00000000"
-- Retrieval info: PRIVATE: PHASE_SHIFT6 STRING "0.00000000"
-- Retrieval info: PRIVATE: MULT_FACTOR6 NUMERIC "1"
-- Retrieval info: PRIVATE: JUMP2PAGE5 STRING "General/Modes"
-- Retrieval info: PRIVATE: TIME_SHIFT5 STRING "0.00000000"
-- Retrieval info: PRIVATE: STICKY_CLK5 STRING "0"
-- Retrieval info: PRIVATE: USE_CLK3 STRING "0"
-- Retrieval info: PRIVATE: GLOCKED_MODE_CHECK STRING "0"
-- Retrieval info: PRIVATE: NORMAL_MODE_RADIO STRING "1"
-- Retrieval info: PRIVATE: CUR_FBIN_CLK STRING "e0"
-- Retrieval info: PRIVATE: MIRROR_CLK8 STRING "0"
-- Retrieval info: PRIVATE: PHASE_SHIFT_UNIT8 STRING "deg"
-- Retrieval info: PRIVATE: DUTY_CYCLE7 STRING "50.00000000"
-- Retrieval info: PRIVATE: PHASE_SHIFT7 STRING "0.00000000"
-- Retrieval info: PRIVATE: MULT_FACTOR7 NUMERIC "1"
-- Retrieval info: PRIVATE: JUMP2PAGE6 STRING "□"
-- Retrieval info: PRIVATE: TIME_SHIFT6 STRING "0.00000000"
-- Retrieval info: PRIVATE: STICKY_CLK6 STRING "0"
-- Retrieval info: PRIVATE: USE_CLK4 STRING "0"
-- Retrieval info: PRIVATE: DIV_FACTOR0 NUMERIC "25"
-- Retrieval info: PRIVATE: INCLK1_FREQ_UNIT_CHANGED STRING "1"
-- Retrieval info: PRIVATE: EXT_FEEDBACK_RADIO STRING "0"
-- Retrieval info: PRIVATE: MIRROR_CLK9 STRING "0"
-- Retrieval info: PRIVATE: PHASE_SHIFT_UNIT9 STRING "deg"
-- Retrieval info: PRIVATE: DUTY_CYCLE8 STRING "50.00000000"
-- Retrieval info: PRIVATE: PHASE_SHIFT8 STRING "0.00000000"
-- Retrieval info: PRIVATE: MULT_FACTOR8 NUMERIC "1"
-- Retrieval info: PRIVATE: JUMP2PAGE7 STRING "General/Modes"
-- Retrieval info: PRIVATE: TIME_SHIFT7 STRING "0.00000000"
-- Retrieval info: PRIVATE: STICKY_CLK7 STRING "0"
-- Retrieval info: PRIVATE: USE_CLK5 STRING "0"
-- Retrieval info: PRIVATE: DIV_FACTOR1 NUMERIC "50"
-- Retrieval info: PRIVATE: CLKLOSS_CHECK STRING "0"
-- Retrieval info: PRIVATE: BANDWIDTH_USE_AUTO STRING "1"
-- Retrieval info: PRIVATE: SHORT_SCAN_RADIO STRING "0"
-- Retrieval info: PRIVATE: DUTY_CYCLE9 STRING "50.00000000"
-- Retrieval info: PRIVATE: PHASE_SHIFT9 STRING "0.00000000"
-- Retrieval info: PRIVATE: MULT_FACTOR9 NUMERIC "1"
-- Retrieval info: PRIVATE: JUMP2PAGE8 STRING "General/Modes"
-- Retrieval info: PRIVATE: TIME_SHIFT8 STRING "0.00000000"
-- Retrieval info: PRIVATE: STICKY_CLK8 STRING "0"
-- Retrieval info: PRIVATE: USE_CLK6 STRING "0"
-- Retrieval info: PRIVATE: DIV_FACTOR2 NUMERIC "1"
-- Retrieval info: PRIVATE: CLKSWITCH_CHECK STRING "0"
-- Retrieval info: PRIVATE: SPREAD_FREQ_UNIT STRING "KHz"
-- Retrieval info: PRIVATE: PLL_ENA_CHECK STRING "0"
-- Retrieval info: PRIVATE: INCLK0_FREQ_EDIT STRING "50.000"
-- Retrieval info: PRIVATE: JUMP2PAGE9 STRING "General/Modes"
-- Retrieval info: PRIVATE: TIME_SHIFT9 STRING "0.00000000"
-- Retrieval info: PRIVATE: STICKY_CLK9 STRING "0"
-- Retrieval info: PRIVATE: USE_CLK7 STRING "0"
-- Retrieval info: PRIVATE: DIV_FACTOR3 NUMERIC "1"
-- Retrieval info: PRIVATE: CNX_NO_COMPENSATE_RADIO STRING "0"
-- Retrieval info: PRIVATE: INT_FEEDBACK__MODE_RADIO STRING "1"
-- Retrieval info: PRIVATE: USE_CLK8 STRING "0"
-- Retrieval info: PRIVATE: DIV_FACTOR4 NUMERIC "1"
-- Retrieval info: PRIVATE: JUMP2PAGE STRING "Clock switchover"
-- Retrieval info: PRIVATE: USE_CLK9 STRING "0"
-- Retrieval info: PRIVATE: DIV_FACTOR5 NUMERIC "1"
-- Retrieval info: PRIVATE: PRIMARY_CLK_COMBO STRING "inclk0"
-- Retrieval info: PRIVATE: CREATE_INCLK1_CHECK STRING "0"
-- Retrieval info: PRIVATE: SACN_INPUTS_CHECK STRING "0"
-- Retrieval info: PRIVATE: DEV_FAMILY STRING "Stratix"
-- Retrieval info: PRIVATE: DIV_FACTOR6 NUMERIC "50"
-- Retrieval info: PRIVATE: SWITCHOVER_COUNT_EDIT NUMERIC "1"
-- Retrieval info: PRIVATE: SWITCHOVER_FEATURE_ENABLED STRING "1"
-- Retrieval info: PRIVATE: BANDWIDTH_PRESET STRING "Low"
-- Retrieval info: PRIVATE: GLOCKED_FEATURE_ENABLED STRING "1"
-- Retrieval info: PRIVATE: DIV_FACTOR7 NUMERIC "1"
-- Retrieval info: PRIVATE: USE_CLKENA0 STRING "0"
-- Retrieval info: PRIVATE: DIV_FACTOR8 NUMERIC "1"
-- Retrieval info: PRIVATE: USE_CLKENA1 STRING "0"
-- Retrieval info: PRIVATE: CLKBAD_SWITCHOVER_CHECK STRING "0"
-- Retrieval info: PRIVATE: BANDWIDTH_USE_PRESET STRING "0"
-- Retrieval info: PRIVATE: DEVICE_FAMILY NUMERIC "9"
-- Retrieval info: LIBRARY: altera_mf altera_mf.altera_mf_components.all
-- Retrieval info: CONSTANT: CLK1_DIVIDE_BY NUMERIC "50"
-- Retrieval info: CONSTANT: BANDWIDTH_TYPE STRING "AUTO"
-- Retrieval info: CONSTANT: CLK1_PHASE_SHIFT STRING "0"
-- Retrieval info: CONSTANT: CLK0_DUTY_CYCLE NUMERIC "50"
-- Retrieval info: CONSTANT: LPM_TYPE STRING "altpll"
-- Retrieval info: CONSTANT: CLK0_MULTIPLY_BY NUMERIC "16"
-- Retrieval info: CONSTANT: LOCK_LOW NUMERIC "5"
-- Retrieval info: CONSTANT: INVALID_LOCK_MULTIPLIER NUMERIC "5"
-- Retrieval info: CONSTANT: INCLK0_INPUT_FREQUENCY NUMERIC "20000"
-- Retrieval info: CONSTANT: GATE_LOCK_SIGNAL STRING "NO"
-- Retrieval info: CONSTANT: CLK0_DIVIDE_BY NUMERIC "25"
-- Retrieval info: CONSTANT: CLK1_DUTY_CYCLE NUMERIC "50"
-- Retrieval info: CONSTANT: PLL_TYPE STRING "AUTO"
-- Retrieval info: CONSTANT: VALID_LOCK_MULTIPLIER NUMERIC "1"
-- Retrieval info: CONSTANT: CLK1_MULTIPLY_BY NUMERIC "1"
-- Retrieval info: CONSTANT: CLK0_TIME_DELAY STRING "0"
-- Retrieval info: CONSTANT: SPREAD_FREQUENCY NUMERIC "0"
-- Retrieval info: CONSTANT: OPERATION_MODE STRING "NORMAL"
-- Retrieval info: CONSTANT: LOCK_HIGH NUMERIC "1"
-- Retrieval info: CONSTANT: COMPENSATE_CLOCK STRING "CLK1"
-- Retrieval info: CONSTANT: CLK1_TIME_DELAY STRING "0"
-- Retrieval info: CONSTANT: CLK0_PHASE_SHIFT STRING "0"
-- Retrieval info: USED_PORT: c0 0 0 0 0 OUTPUT VCC "c0"
-- Retrieval info: USED_PORT: @clk 0 0 6 0 OUTPUT VCC "@clk[5..0]"
-- Retrieval info: USED_PORT: c1 0 0 0 0 OUTPUT VCC "c1"
-- Retrieval info: USED_PORT: inclk0 0 0 0 0 INPUT GND "inclk0"
-- Retrieval info: USED_PORT: locked 0 0 0 0 OUTPUT GND "locked"
-- Retrieval info: USED_PORT: @extclk 0 0 4 0 OUTPUT VCC "@extclk[3..0]"
-- Retrieval info: USED_PORT: @inclk 0 0 2 0 INPUT VCC "@inclk[1..0]"
-- Retrieval info: CONNECT: @clkena 0 0 1 1 VCC 0 0 0 0
-- Retrieval info: CONNECT: locked 0 0 0 0 @locked 0 0 0 0
-- Retrieval info: CONNECT: @inclk 0 0 1 0 inclk0 0 0 0 0
-- Retrieval info: CONNECT: @extclkena 0 0 1 1 GND 0 0 0 0
-- Retrieval info: CONNECT: @clkena 0 0 1 4 GND 0 0 0 0
-- Retrieval info: CONNECT: c0 0 0 0 0 @clk 0 0 1 0
-- Retrieval info: CONNECT: c1 0 0 0 0 @clk 0 0 1 1
-- Retrieval info: CONNECT: @extclkena 0 0 1 2 GND 0 0 0 0
-- Retrieval info: CONNECT: @clkena 0 0 1 5 GND 0 0 0 0
-- Retrieval info: CONNECT: @clkena 0 0 1 2 GND 0 0 0 0
-- Retrieval info: CONNECT: @clkena 0 0 1 0 VCC 0 0 0 0
-- Retrieval info: CONNECT: @inclk 0 0 1 1 GND 0 0 0 0
-- Retrieval info: CONNECT: @extclkena 0 0 1 3 GND 0 0 0 0
-- Retrieval info: CONNECT: @extclkena 0 0 1 0 GND 0 0 0 0
-- Retrieval info: CONNECT: @clkena 0 0 1 3 GND 0 0 0 0
```

# Appendix 2 – SPICE listing for Current Starved Inverter Amplifier

**CS_AMP.SPICE**

```
*Current Starved Inverter Controlled Amplifier
*Calum Johnson
*
.INCLUDE 'n66c.l3'
.INCLUDE 'test.spice'
.options NOMOD
.probe
*.op
*.print i(vdd)
.print v(21)
*.probe v(21)

*Transient Slew Test
.tran  .01E-6 4E-6

       Vin     50      0       DC      5
*+AC    1
+PWL (0 0 1E-6 0 1.0001E-6 5 2E-6 5 2.0001E-6 0 4E-6 0
*+4.0001E-6 5 24E-6 5 24.0001E-6 0 32E-6 0)
*+sin 2.5 2.5 75k

Vdd     1      0       DC      5
*External Parts
*refrence load
Cref    20     0       20E-12
*unkown load
Cload   21     0       100E-12


X1 1 50 20 6 test
M1 4 21 3 1 CMOSP W=9.60U L=2.40U AD=38.40P PD=27.20U AS=31.57P PS=18.84U
M2 5 20 3 1 CMOSP W=9.60U L=2.40U AD=38.40P PD=27.20U AS=31.57P PS=18.84U
M3 4 5 0 0 CMOSN W=4.00U L=2.40U AD=16.00P PD=16.00U AS=13.52P PS=7.07U
M4 5 5 0 0 CMOSN W=4.00U L=2.40U AD=16.00P PD=16.00U AS=13.52P PS=7.07U
M11 2 2 1 1 CMOSP W=24.00U L=2.40U AD=57.60P PD=28.80U AS=72.22P PS=41.50U
M12 3 2 1 1 CMOSP W=24.00U L=2.40U AD=78.93P PD=47.11U AS=72.22P PS=41.50U
M13 10 10 0 0 CMOSN W=24.00U L=2.40U AD=96.00P PD=56.00U AS=81.14P PS=42.43U
M14 30 10 0 0 CMOSN W=24.00U L=2.40U AD=78.93P PD=47.11U AS=81.14P PS=42.43U
M15 31 20 30 0 CMOSN W=9.60U L=2.40U AD=38.40P PD=27.20U AS=31.57P PS=18.84U
M16 32 21 30 0 CMOSN W=9.60U L=2.40U AD=38.40P PD=27.20U AS=31.57P PS=18.84U
M17 31 31 1 1 CMOSP W=4.00U L=2.40U AD=16.00P PD=16.00U AS=12.04P PS=6.92U
M18 32 31 1 1 CMOSP W=4.00U L=2.40U AD=16.00P PD=16.00U AS=12.04P PS=6.92U
M30 21 5 0 0 CMOSN M=10 W=19.20U L=2.40U AD=52.22P PD=28.48U AS=64.91P PS=33.94U
M31 21 31 1 1 CMOSP M=10 W=19.20U L=2.40U AD=52.22P PD=28.48U AS=57.77P PS=33.20U
M110 10 10 2 1 CMOSP W=24.00U L=2.40U AD=96.00P PD=56.00U AS=57.60P PS=28.80U
C1 5 0 90.7FF
C2 4 0 4.7FF
C3 3 0 8.9FF
C4 2 0 16.0FF
C5 1 0 64.2FF
C6 50 0 4.9FF
C7 32 0 3.0FF
C8 31 0 79.1FF
C9 30 0 5.8FF
C10 21 0 78.0FF
C11 20 0 23.7FF
C12 10 0 33.2FF
C13 50 6 1.4FF
C14 31 21 5.8FF
C15 10 30 1.5FF
```

```
C16 5 21 5.8FF
C17 21 0 6.4FF
C18 2 1 2.5FF
C19 5 30 1.3FF
C20 5 0 2.9FF
C21 31 1 3.3FF
C22 1 10 1.0FF
C23 21 1 2.6FF
.END



.SUBCKT test 1 2 3 7
C1 6 0 CAPAMI1r6 SCALE=17242.88
* C1=10345.73FF
C_C1 0 0 CAPAMI1r6P SCALE=17242.88
* C_C1=224.16FF
M1 3 2 4 1 CMOSP M=10 W=24.00U L=1.60U AD=65.28P PD=34.24U AS=57.60P PS=28.80U
M2 3 2 5 0 CMOSN M=4 W=24.00U L=1.60U AD=101.12P PD=56.00U AS=60.16P PS=28.80U
M3 4 6 1 1 CMOSP M=10 W=24.00U L=1.60U AD=57.60P PD=28.80U AS=66.84P PS=35.70U
M4 5 7 0 0 CMOSN M=4 W=24.00U L=1.60U AD=60.16P PD=28.80U AS=100.24P PS=57.93U
M5 6 6 1 1 CMOSP W=12.80U L=1.60U AD=51.20P PD=33.60U AS=35.65P PS=19.04U
M6 6 7 0 0 CMOSN W=4.00U L=1.60U AD=16.00P PD=16.00U AS=16.71P PS=9.66U
M7 7 7 0 0 CMOSN W=16.00U L=1.60U AD=64.00P PD=41.14U AS=66.82P PS=38.62U
M8 10 8 8 0 CMOSN W=12.00U L=1.60U AD=28.80P PD=16.80U AS=48.00P PS=32.00U
M9 9 10 10 0 CMOSN W=12.00U L=1.60U AD=28.80P PD=16.80U AS=28.80P PS=16.80U
M10 7 9 9 0 CMOSN W=12.00U L=1.60U AD=48.00P PD=30.86U AS=28.80P PS=16.80U
C2 9 0 2.8FF
C3 8 0 10.5FF
C4 7 0 15.2FF
C5 6 0 287.2FF
C6 5 0 4.2FF
C7 4 0 26.2FF
C8 3 0 31.4FF
C9 2 0 67.2FF
C10 1 0 59.3FF
C11 10 0 2.8FF
C12 6 1 21.6FF
C13 8 9 1.1FF
C14 4 6 1.8FF
C15 6 0 18.0FF
C16 4 2 1.8FF
C17 3 2 4.0FF


.MODEL CAPAMI1r6P C CAP=0.013FF
.MODEL CAPAMI1r6 C CAP=0.600FF
*additional information
R1      8      1       75k
*Except this, add this in later
*Cslow  6      0       10E-12
.ENDS
```

# Appendix 3 – SPICE Listing for Slew Rate Limited Amplifier

**SLEW_AMP.SPICE**

```
*Slew-Rate Limited Amplifier
*Calum Johnson
*

.INC 'n66c.l3'
*.INC 'ami_c5n_typical.txt'
.INC 'slub_amp.spice'
.OPTIONS NOMOD
.PROBE
.OP

*Stability Tests
*.MEASURE AC aol_phase FIND vp(6) WHEN vdb(6)=0
*.MEASURE AC aol_phase_deg PARAM='aol_phase*(180/pi)'
*.MEASURE AC GBW WHEN v(6)=1
*.MEASURE AC phase_margin PARAM='180+aol_phase_deg'
*AC Tests
*.AC    DEC     100     1       1E8
*Transient Slew Test
.TRAN  .1E-6 32E-6
.PRINT TRAN V(6)

*DC Input Range Test
*.DC    Vin     0       5       .1


Vin     22      0       DC      2.5
*+AC    1
+PWL (0 0 1E-6 0 1.0001E-6 5 8E-6 5 8.0001E-6 0 16E-6 0 16.0001E-6 5
+24E-6 5 24.0001E-6 0 32E-6 0)
*+sin 2.5 2.5 75k

*Vopp   105     0       DC      2.5
*+AC    1
*+PWL (0 5 1E-6 5 1.0001E-6 0 8E-6 0 8.0001E-6 5 16E-6 5 16.0001E-6 0
*+24E-6 0 24.0001E-6 5 32E-6 5)

Vdd     1       0       DC      5

*Base Bias Current
R1      1       5       400E3

*slew-amp
XSA     1       5       21      22      6       1       1
+       1       1       1       1       1       1       0       SLEWAMP

*Unkown Load
Cload   6       0       100E-12

*Unity Gain Configuration for Transient Tests
Rfb     6       21      0

*Test Schematic for Stability Tests
*Rfb    6       21      100E6
*Cfb    21      0       10E-3

.END
```

**SUB_AMP.SPICE**

```
*Sub_amp extracted spice file, parsed for simulation

*Connections: Vdd, Bias Current, Inv Input, Non-Inv Input, Output,
*              8 Slew Controls, Gnd
*
.SUBCKT SLEWAMP               1     5        21      22      6       41
+42    43      44     45      46     47      48      0
************BEGIN EXTRACTED SPICE FILE************
* slew_amp.spice
*
* File Location        /users2/students/johnsoncj/magic/analog_fun/slew_amp
* File Created         Thu May 22 02:36:55 2003
* Ext2spice Version    ORNL 3.2.8 <=> Wed Apr 23 14:28:38 EDT 2003

* Option Settings:

*    hspice                           = ON
*    Output_Substrate_Node            = OFF
*    Output_Well_Diodes               = OFF
*    Output_Distributed_Models        = OFF
*    Output_Models_In_Spice           = ON
*    Create_Circuit_File              = ON
*    Add_Node_Indicators              = OFF
*    Use_Proper_Node_Names            = ON
*    Star_Node_Model                  = OFF
*    Zero_Subckt_Stray_Capacitance    = OFF
*    Zero_Parasitic_Capacitance       = OFF
*    CAZM_format                      = OFF
*    Output_Transistor_Lambda         = OFF
*    Output_Transistor_NParams        = OFF
*    Output_Transistor_GeoValue       = OFF
*    Merge_Parallel_Transistors       = ON
*    Match_Merge_Parallel_Transistors = ON
*    Bin_Transistor_Values            = OFF
*    Short_allS_Type_Resistance       = OFF
*    Short_S_Type_Resistance          = OFF
*    Merge_Series_Resistance          = ON
*    Merge_Parallel_Capacitance       = ON

** *****************************************************************************
** *****************************************************************************
**
****** top level cell is slew_amp

** Extraction file is /users2/students/johnsoncj/magic/analog_fun/slew_amp/slew_amp.ext
**
** *****************************************************************************
** *****************************************************************************
Cc 60 6 CAPAMI1r6 SCALE=8908.80
* Cc=5345.28FF
C_Cc 6 1 CAPAMI1r6P SCALE=8908.80
* C_Cc=115.81FF
M1 5 5 0 0 CMOSN W=15.20U L=2.40U AD=60.80P PD=38.40U AS=61.14P PS=43.46U
M4 2 2 1 1 CMOSP M=2 W=17.60U L=2.40U AD=70.40P PD=43.20U AS=70.40P PS=44.23U
M5 3 2 1 1 CMOSP M=2 W=17.60U L=2.40U AD=70.40P PD=43.20U AS=70.40P PS=44.23U
M6 2 21 4 0 CMOSN W=17.60U L=2.40U AD=70.40P PD=43.20U AS=70.40P PS=58.36U
M7 3 22 4 0 CMOSN W=17.60U L=2.40U AD=70.40P PD=43.20U AS=70.40P PS=58.36U
M13 3 5 60 1 CMOSP W=17.60U L=2.40U AD=70.40P PD=43.20U AS=70.40P PS=43.20U
M21 31 5 0 0 CMOSN W=9.60U L=2.40U AD=30.72P PD=21.60U AS=38.62P PS=27.45U
M22 32 5 0 0 CMOSN W=9.60U L=2.40U AD=30.72P PD=21.60U AS=38.62P PS=27.45U
M23 33 5 0 0 CMOSN W=9.60U L=2.40U AD=30.72P PD=21.60U AS=38.62P PS=27.45U
M24 34 5 0 0 CMOSN W=9.60U L=2.40U AD=30.72P PD=21.60U AS=38.62P PS=27.45U
M25 35 5 0 0 CMOSN W=9.60U L=2.40U AD=30.72P PD=21.60U AS=38.62P PS=27.45U
M26 36 5 0 0 CMOSN W=9.60U L=2.40U AD=30.72P PD=21.60U AS=38.62P PS=27.45U
M27 37 5 0 0 CMOSN W=9.60U L=2.40U AD=30.72P PD=21.60U AS=38.62P PS=27.45U
M28 38 5 0 0 CMOSN W=9.60U L=2.40U AD=30.72P PD=21.60U AS=38.62P PS=27.45U
```

```
M41 4 41 31 0 CMOSN W=3.20U L=2.40U AD=12.80P PD=10.61U AS=10.24P PS=7.20U
M42 4 42 32 0 CMOSN W=3.20U L=2.40U AD=12.80P PD=10.61U AS=10.24P PS=7.20U
M43 4 43 33 0 CMOSN W=3.20U L=2.40U AD=12.80P PD=10.61U AS=10.24P PS=7.20U
M44 4 44 34 0 CMOSN W=3.20U L=2.40U AD=12.80P PD=10.61U AS=10.24P PS=7.20U
M45 4 45 35 0 CMOSN W=3.20U L=2.40U AD=12.80P PD=10.61U AS=10.24P PS=7.20U
M46 4 46 36 0 CMOSN W=3.20U L=2.40U AD=12.80P PD=10.61U AS=10.24P PS=7.20U
M47 4 47 37 0 CMOSN W=3.20U L=2.40U AD=12.80P PD=10.61U AS=10.24P PS=7.20U
M48 4 48 38 0 CMOSN W=3.20U L=2.40U AD=12.80P PD=10.61U AS=10.24P PS=7.20U
M50 50 2 1 1 CMOSP M=2 W=17.60U L=2.40U AD=70.40P PD=43.20U AS=70.40P PS=44.23U
M51 52 52 0 0 CMOSN W=2.40U L=2.40U AD=8.68P PD=5.38U AS=9.65P PS=6.86U
M60 50 50 51 0 CMOSN W=17.60U L=2.40U AD=70.40P PD=43.20U AS=42.24P PS=22.40U
M61 51 51 52 0 CMOSN W=17.60U L=2.40U AD=42.24P PD=22.40U AS=63.64P PS=39.42U
M80 6 3 1 1 CMOSP M=4 W=9.60U L=2.40U AD=38.40P PD=27.20U AS=38.40P PS=24.12U
M81 6 52 0 0 CMOSN M=2 W=9.60U L=2.40U AD=38.40P PD=27.20U AS=38.62P PS=27.45U


C1 6 0 156.6FF
C2 5 0 101.8FF
C3 4 0 46.0FF
C4 3 0 39.1FF
C5 2 0 79.1FF
C6 1 0 71.7FF
C7 60 0 129.3FF
C8 48 0 7.1FF
C9 47 0 7.1FF
C10 52 0 36.6FF
C11 46 0 7.1FF
C12 51 0 3.2FF
C13 45 0 7.1FF
C14 50 0 18.4FF
C15 44 0 7.1FF
C16 43 0 7.1FF
C17 42 0 7.1FF
C18 41 0 7.1FF
C19 22 0 4.6FF
C20 21 0 4.6FF
C21 6 52 3.3FF
C22 6 1 11.0FF
C23 3 2 1.5FF
C24 0 4 5.6FF
C25 60 6 11.5FF
C26 2 52 2.0FF
C27 2 1 7.6FF
C28 5 0 8.5FF
C29 3 1 3.5FF
C30 6 0 2.9FF
*** Node Listing for subckt:  slew_amp
** N1                [U=14]    ==  N1
** N2                [U=8]     ==  N2
** N3                [U=5]     ==  N3
** N4                [U=10]    ==  N4
** N5                [U=11]    ==  N5
** N6                [U=3]     ==  N6
** N21               [U=1]     ==  N21
** N22               [U=1]     ==  N22
** N31               [U=2]     ==  N31
** N32               [U=2]     ==  N32
** N33               [U=2]     ==  N33
** N34               [U=2]     ==  N34
** N35               [U=2]     ==  N35
** N36               [U=2]     ==  N36
** N37               [U=2]     ==  N37
** N38               [U=2]     ==  N38
** N41               [U=1]     ==  N41
** N42               [U=1]     ==  N42
** N43               [U=1]     ==  N43
** N44               [U=1]     ==  N44
** N45               [U=1]     ==  N45
** N46               [U=1]     ==  N46
** N47               [U=1]     ==  N47
** N48               [U=1]     ==  N48
** N50               [U=3]     ==  N50
** N51               [U=3]     ==  N51
** N52               [U=5]     ==  N52
** N60               [U=2]     ==  N60
************************************************************
*
* Model Definitions for HSPICE
*
************************************************************
.MODEL CAPAMI1r6P C CAP=0.013FF
.MODEL CAPAMI1r6 C CAP=0.600FF
************************************************************

* Design Summary
*          0 Documentation Errors
*          0 Extraction Errors
*          0 Warnings
************************************************************
************************END EXTRACTED SPICE FILE************
.ENDS
```

# Appendix 4 – Source Listing for FPGA-to-USB Interface

**USB.VHDL**

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_arith.ALL;
USE ieee.std_logic_unsigned.ALL;

ENTITY usb IS PORT(
        osc:                                         IN std_logic;
        TXEB_IN :                                    IN std_logic;
    RXFB_IN :                                IN std_logic;
        USB_DOUT_OUT :                               OUT std_logic_vector(7 DOWNTO 0);
        RDB_OUT :                                    OUT std_logic;
    WR_OUT :                                 OUT std_logic;
        WR_OUT_dupe :                        OUT std_logic;
        done_out :                                   OUT std_logic;
        pusher_out :                         OUT std_logic;
        pinger :                                     OUT std_logic
        );
END usb;

ARCHITECTURE behavioral OF usb IS

        COMPONENT dummy_board
        PORT(
        osc_in:               IN std_logic;
        pll_lock_out:         OUT std_logic := '0';
        output:               OUT std_logic_vector(7 DOWNTO 0);
        quiet:                OUT std_logic := '0';
        pll_clock_out:        OUT std_logic;
        track_clock_out:      OUT std_logic);
        END COMPONENT;

        COMPONENT small_usb
        PORT(
        CLK:                  in std_logic;  -- CLK is 1 MHz
        TXEB:                 in std_logic;
        RXFB:                 in std_logic;
        DATA:                 in std_logic_vector(7 downto 0);
        PUSH:                 in std_logic;
        FINISHED:             out std_logic;
        USB_DOUT:             out std_logic_vector(7 downto 0);
        RDB:                  out std_logic;
        WR:                   out std_logic;
        PING_OUT:             out std_logic);
        END COMPONENT;


        SIGNAL sender:        std_logic_vector(7 downto 0);
        SIGNAL sender_lock:   std_logic;
        SIGNAL sender_quiet:  std_logic;
        SIGNAL meg_clock:     std_logic;
        SIGNAL fast_clock_in: std_logic;
        SIGNAL bi_mark :      std_logic;
        SIGNAL pusher :               std_logic;
        SIGNAL done :         std_logic;
        SIGNAL WR_duper :     std_logic;
        SIGNAL last :         std_logic;

BEGIN

sender_unit : dummy_board PORT MAP (
               osc_in                        => osc,
               pll_lock_out          => sender_lock,
               output                        => sender,
               quiet                         => sender_quiet,
               pll_clock_out         => meg_clock,
               track_clock_out       => fast_clock_in);

usb_writer : small_usb PORT MAP (
                CLK           => fast_clock_in,
         TXEB          => TXEB_IN,
         RXFB          => RXFB_IN,
                DATA          => sender,
                PUSH          => pusher,
                FINISHED      => done,
                USB_DOUT      => USB_DOUT_OUT,
                RDB           => RDB_OUT,
         WR            => WR_duper,
                PING_OUT      => pinger);

--pusher <= '1';

WR_OUT <= WR_duper;
WR_OUT_dupe <= WR_duper;

PROCESS (fast_clock_in)
BEGIN
IF fast_clock_in'event AND fast_clock_in = '1' THEN
        --IF (done = '1') THEN
    --  pusher <= '0';
        IF (sender_quiet = '0') THEN
                last <= '0';
                pusher <= '0';
        ELSIF (sender_quiet = '1') THEN
                IF (last = '0') THEN
                        pusher <= '1';
                ELSE
                        pusher <= '0';
                END IF;
                last <= '1';
        ELSE
                pusher <= '0';
        END IF;
END IF;
END PROCESS;

pusher_out <= pusher;
done_out <= done;

END behavioral;
```

**SMALL_USB.VHDL**

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;


entity small_usb is
    Port (
         CLK : in std_logic;  -- CLK is 10 MHz
         TXEB : in std_logic;
         RXFB : in std_logic;
                DATA : in std_logic_vector(7 downto 0);
                PUSH : in std_logic;
                FINISHED : out std_logic;
                USB_DOUT : out std_logic_vector(7 downto 0);
                RDB : out std_logic;
         WR : out std_logic;
                PING_OUT : out std_logic);
end small_usb;

architecture Behavioral of small_usb is

signal USB_DOUT1,USB_DOUT0:std_logic_vector(7 downto 0):=X"00";
signal RDBS:std_logic:='1';
signal WRSL,WRS,RESET,RST:std_logic:='0';
signal PUT_DATA,PUT_DATA1:std_logic:='0';
signal PING,PING_DONE:std_logic:='0';
signal DIR1:std_logic:='0';
signal FIN:std_logic:='0';
--signal KEEP_READ,COUNT_EN,TX_EN,LVL1S,SER_EN1,DREN,LATCH:std_logic:='0';

begin

RST <= '0';
RDB <= RDBS;
USB_DOUT <= USB_DOUT0;


RESET <= '0';
FINISHED <= FIN;
RDBS <= '1';
PING_OUT <= PING;



----------------------------------------
------------- USB Write -----------------
----------------------------------------

process(CLK,RDBS)

begin

   if RDBS = '0' then
      WRS <= '0';
   elsif CLK'event and CLK = '1' then
     if WRS = '0' and PUT_DATA = '1' then --TXEB = '0' and
            WRS <= '1';
         else
            WRS <= '0';
         end if;
   end if;

end process;

process(CLK)
```

```
begin

   if CLK'event and CLK = '1' then
      if WRS = '1' then
            PUT_DATA <= '0';
         else
            PUT_DATA <= PUT_DATA1;
         end if;
   end if;

end process;

process(CLK)

begin

   if CLK'event and CLK = '0' then
     WR <= WRSL;
         WRSL <= WRS;
   end if;

end process;


----------------------------------------
------------- USB_DOUT -----------------
----------------------------------------

process(CLK)

begin

        if CLK'event and CLK = '1' then
               if DIR1 = '1' then
            USB_DOUT0 <= USB_DOUT1;
        else
            USB_DOUT0 <= USB_DOUT0;
        end if;
end if;

end process;

--process(CLK)

--begin

--   if CLK'event and CLK = '1' then
--      EN <= DIR1;
--   end if;

--end process;



----------------------------------------
------------- Main Ctrl ----------------
----------------------------------------

process(CLK)

variable STATE:integer range 0 to 2;

begin

   if CLK'event and CLK = '1' then
      if RESET = '1' then
            STATE := 0;
            PING <= '0';
               FIN <= '1';
      elsif STATE = 0 then
            PING <= '0';
               if PUSH = '1' then
                       STATE := 1;
                       FIN <= '0';
            else
               STATE := 0;
                       FIN <= '1';
            end if;
         elsif STATE = 1 then
               FIN <= '0';
               PING <= '1';
               STATE := 2;
         elsif STATE = 2 then
            PING <= '0';
            if PING_DONE = '1' then
               STATE := 0;
                       FIN <= '1';
            else
               STATE := 2;
                       FIN <= '0';
            end if;
         else
            STATE := 0;
            PING <= '0';
               FIN <= '1';
         end if;
   end if;

end process;


----------------------------------------
---------------- Ping ------------------
----------------------------------------

process(CLK)

variable STATE:integer range 0 to 7;

begin

   if CLK'event and CLK = '1' then
     if RESET = '1' then
            STATE := 0;
            PUT_DATA1 <= '0';
            USB_DOUT1 <= X"00";
            PING_DONE <= '0';
            DIR1 <= '0';
         elsif STATE = 0 then
            USB_DOUT1 <= X"00";
            PUT_DATA1 <= '0';
            PING_DONE <= '0';
            DIR1 <= '0';
            if PING = '1' then
              STATE := 1;
            else
              STATE := 0;
            end if;
         elsif STATE = 1 then
            if WRS = '1' then
             STATE := 3;
                 PUT_DATA1 <= '0';
                 USB_DOUT1 <= DATA;--calum code
                 DIR1 <= '1';
            else
```

```
              STATE := 1;
                 PUT_DATA1 <= '1';
                 USB_DOUT1 <= X"00";
                 DIR1 <= '0';
            end if;
         elsif STATE = 3 then
            PING_DONE <= '1';
            STATE := 2;
            DIR1 <= '0';
         elsif STATE = 2 then
            PING_DONE <= '0';
            STATE := 0;
         else
            STATE := 0;
         end if;
   end if;

end process;

END Behavioral;
```

**DUMMY_BOARD.VHDL**

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_arith.ALL;
USE ieee.std_logic_unsigned.ALL;

ENTITY dummy_board IS PORT(
        osc_in:                                 IN std_logic;
        pll_lock_out:                           OUT std_logic := '0';
        output:                                 OUT std_logic_vector(7 DOWNTO 0);
    quiet:                                      OUT std_logic := '0';
        pll_clock_out:                          OUT std_logic := '0';
        track_clock_out:                        OUT std_logic := '0'
        );
END dummy_board;

ARCHITECTURE behavioral OF dummy_board IS

        COMPONENT clock
        PORT
        (
                inclk0          : IN STD_LOGIC  := '0';
                c0              : OUT STD_LOGIC ;
                c1              : OUT STD_LOGIC ;
                locked          : OUT STD_LOGIC
        );
        END COMPONENT;


        TYPE state IS (sleep,header, middle, footer);
        SIGNAL pause_cnt :              integer RANGE 0 TO 7 := 0;
        SIGNAL data :                   integer RANGE 0 TO 15 := 0;
        SIGNAL counter :                integer RANGE 0 TO 31 := 0;
        SIGNAL position :               state := header;
        SIGNAL data_bits :              std_logic_vector (3 downto 0);
        SIGNAL pll_clock :              std_logic;
        SIGNAL pll_lock :               std_logic := '0';


BEGIN

        onemeg_clock : clock PORT MAP (
                inclk0          => osc_in,
                c0              => track_clock_out,
                c1              => pll_clock,
                locked          => pll_lock
        );

        pll_clock_out <= pll_clock;
        pll_lock_out <= pll_lock;

        upcount: PROCESS( pll_clock ) BEGIN

         IF ( pll_clock'event AND pll_clock = '1' ) THEN
                        IF pll_lock = '1' THEN
                        IF position = header THEN
                                quiet <= '0';
                                output <= X"00";
                                counter <= counter + 1;
                                IF counter = 31 THEN
                                        counter <= 0;
                                        position <= middle;
                                END IF;
            ELSIF position = middle THEN
                                quiet <= '0';
                                data_bits <= conv_std_logic_vector(data, 4);
                                CASE counter IS
```

```
                                      WHEN 0 => output <= data_bits & data_bits;
                              WHEN OTHERS =>  counter <= 0;
                                                                    data <= data + 1;
                                                                    position <= footer;
                              END CASE;
                              counter <= counter + 1;
                      ELSIF position = footer THEN
                quiet <= '0';
                              counter <= counter + 1;
                              IF counter = 31 THEN
                                      counter <= 0;
                                      position <= sleep;
                              END IF;
                      ELSE
                              quiet <= '1';
                              counter <= counter + 1;
                              IF counter = 31 THEN
                                      pause_cnt <= pause_cnt + 1;
                                      IF pause_cnt = 7 THEN
                                              counter <= 0;
                                              pause_cnt <= 0;
                                              position <= header;
                                      ELSE
                                              counter <= 0;
                                      END IF;
                              END IF;
                      END IF;
                      END IF;
        END IF;
    END PROCESS upcount;
END behavioral;
```

**CLOCK.VHDL**

```
-- megafunction wizard: %ALTCLKLOCK%
-- GENERATION: STANDARD
-- VERSION: WM1.0
-- MODULE: altpll

-- ============================================================
-- File Name: clock.vhd
-- Megafunction Name(s):
--                   altpll
-- ============================================================
-- ************************************************************
-- THIS IS A WIZARD-GENERATED FILE. DO NOT EDIT THIS FILE!
-- ************************************************************


--Copyright (C) 1991-2003 Altera Corporation
--Any  megafunction  design,  and related netlist (encrypted  or  decrypted),
--support information,  device programming or simulation file,  and any other
--associated  documentation or information  provided by  Altera  or a partner
--under  Altera's   Megafunction   Partnership   Program may be  used  only
--to program  PLD  devices (but not masked  PLD  devices) from  Altera.   Any
--other  use  of such  megafunction  design,  netlist,  support  information,
--device programming or simulation file,  or any other  related documentation
--or information  is prohibited  for  any  other purpose,  including, but not
--limited to  modification,  reverse engineering,  de-compiling, or use  with
--any other  silicon devices,  unless such use is  explicitly  licensed under
--a separate agreement with  Altera  or a megafunction partner.  Title to the
--intellectual property,  including patents,  copyrights,  trademarks,  trade
--secrets,  or maskworks,  embodied in any such megafunction design, netlist,
--support  information,  device programming or simulation file,  or any other
--related documentation or information provided by  Altera  or a megafunction
--partner, remains with Altera, the megafunction partner, or their respective
--licensors. No other licenses, including any licenses needed under any third
--party's intellectual property, are provided herein.


LIBRARY ieee;
USE ieee.std_logic_1164.all;

LIBRARY altera_mf;
USE altera_mf.altera_mf_components.all;

ENTITY clock IS
        PORT
        (
                inclk0          : IN STD_LOGIC  := '0';
                c0              : OUT STD_LOGIC ;
                c1              : OUT STD_LOGIC ;
                locked          : OUT STD_LOGIC
        );
END clock;


ARCHITECTURE SYN OF clock IS

        SIGNAL sub_wire0        : STD_LOGIC_VECTOR (5 DOWNTO 0);
        SIGNAL sub_wire1        : STD_LOGIC ;
        SIGNAL sub_wire2        : STD_LOGIC ;
        SIGNAL sub_wire3        : STD_LOGIC ;
        SIGNAL sub_wire4_bv     : BIT_VECTOR (0 DOWNTO 0);
        SIGNAL sub_wire4        : STD_LOGIC_VECTOR (0 DOWNTO 0);
        SIGNAL sub_wire5        : STD_LOGIC_VECTOR (5 DOWNTO 0);
        SIGNAL sub_wire6_bv     : BIT_VECTOR (0 DOWNTO 0);
        SIGNAL sub_wire6        : STD_LOGIC_VECTOR (0 DOWNTO 0);
        SIGNAL sub_wire7        : STD_LOGIC ;
        SIGNAL sub_wire8        : STD_LOGIC_VECTOR (1 DOWNTO 0);
        SIGNAL sub_wire9        : STD_LOGIC_VECTOR (3 DOWNTO 0);



        COMPONENT altpll
        GENERIC (
                clk1_divide_by          : NATURAL;
                bandwidth_type          : STRING;
                clk1_phase_shift                : STRING;
                clk0_duty_cycle                 : NATURAL;
                lpm_type                : STRING;
                clk0_multiply_by                : NATURAL;
                lock_low                : NATURAL;
                invalid_lock_multiplier                 : NATURAL;
                inclk0_input_frequency          : NATURAL;
                gate_lock_signal                : STRING;
                clk0_divide_by          : NATURAL;
                clk1_duty_cycle                 : NATURAL;
                pll_type                : STRING;
                valid_lock_multiplier           : NATURAL;
                clk1_multiply_by                : NATURAL;
                clk0_time_delay                 : STRING;
                spread_frequency                : NATURAL;
                operation_mode          : STRING;
                lock_high               : NATURAL;
                compensate_clock                : STRING;
                clk1_time_delay                 : STRING;
                clk0_phase_shift                : STRING
        );
        PORT (
                        clkena  : IN STD_LOGIC_VECTOR (5 DOWNTO 0);
                        inclk   : IN STD_LOGIC_VECTOR (1 DOWNTO 0);
                        extclkena       : IN STD_LOGIC_VECTOR (3 DOWNTO 0);
                        locked  : OUT STD_LOGIC ;
                        clk     : OUT STD_LOGIC_VECTOR (5 DOWNTO 0)
        );
        END COMPONENT;

BEGIN
        sub_wire4_bv(0 DOWNTO 0) <= "0";
        sub_wire4    <= NOT(To_stdlogicvector(sub_wire4_bv));
        sub_wire6_bv(0 DOWNTO 0) <= "0";
        sub_wire6    <= To_stdlogicvector(sub_wire6_bv);
        sub_wire2    <= sub_wire0(1);
        sub_wire1    <= sub_wire0(0);
        c0    <= sub_wire1;
        c1    <= sub_wire2;
        locked    <= sub_wire3;
        sub_wire5    <= sub_wire6(0 DOWNTO 0) & sub_wire6(0 DOWNTO 0) & sub_wire6(0 DOWNTO
0) & sub_wire6(0 DOWNTO 0) & sub_wire4(0 DOWNTO 0) & sub_wire4(0 DOWNTO 0);
        sub_wire7    <= inclk0;
        sub_wire8    <= sub_wire6(0 DOWNTO 0) & sub_wire7;
        sub_wire9    <= sub_wire6(0 DOWNTO 0) & sub_wire6(0 DOWNTO 0) & sub_wire6(0 DOWNTO
0) & sub_wire6(0 DOWNTO 0);

        altpll_component : altpll
        GENERIC MAP (
                clk1_divide_by => 50,
                bandwidth_type => "AUTO",
                clk1_phase_shift => "0",
                clk0_duty_cycle => 50,
                lpm_type => "altpll",
                clk0_multiply_by => 1,
                lock_low => 5,
                invalid_lock_multiplier => 5,
                inclk0_input_frequency => 20000,
                gate_lock_signal => "NO",
                clk0_divide_by => 50,
                clk1_duty_cycle => 50,
                pll_type => "AUTO",
                valid_lock_multiplier => 1,
                clk1_multiply_by => 1,
                clk0_time_delay => "0",
                spread_frequency => 0,
                operation_mode => "NORMAL",
                lock_high => 1,
                compensate_clock => "CLK1",
                clk1_time_delay => "0",
                clk0_phase_shift => "0"
        )
        PORT MAP (
                clkena => sub_wire5,
                inclk => sub_wire8,
                extclkena => sub_wire9,
                clk => sub_wire0,
                locked => sub_wire3
        );



END SYN;

-- ============================================================
-- CNX file retrieval info
-- ============================================================
-- Retrieval info: PRIVATE: DIV_FACTOR9 NUMERIC "1"
-- Retrieval info: PRIVATE: USE_CLKENA2 STRING "0"
-- Retrieval info: PRIVATE: MIRROR_CLK0 STRING "0"
-- Retrieval info: PRIVATE: PHASE_SHIFT_UNIT0 STRING "deg"
-- Retrieval info: PRIVATE: INCLK1_FREQ_UNIT_COMBO STRING "MHz"
-- Retrieval info: PRIVATE: SPREAD_USE STRING "0"
-- Retrieval info: PRIVATE: SPREAD_FEATURE_ENABLED STRING "1"
-- Retrieval info: PRIVATE: GLOCKED_COUNTER_EDIT_CHANGED STRING "1"
-- Retrieval info: PRIVATE: GLOCK_COUNTER_EDIT NUMERIC "10000"
-- Retrieval info: PRIVATE: USE_CLKENA3 STRING "0"
-- Retrieval info: PRIVATE: MIRROR_CLK1 STRING "0"
-- Retrieval info: PRIVATE: PHASE_SHIFT_UNIT1 STRING "deg"
-- Retrieval info: PRIVATE: DUTY_CYCLE0 STRING "50.00000000"
-- Retrieval info: PRIVATE: PHASE_SHIFT0 STRING "0.00000000"
-- Retrieval info: PRIVATE: MULT_FACTOR0 NUMERIC "1"
-- Retrieval info: PRIVATE: SPREAD_PERCENT STRING "0.500"
-- Retrieval info: PRIVATE: LOCKED_OUTPUT_CHECK STRING "1"
-- Retrieval info: PRIVATE: PLL_ARESET_CHECK STRING "0"
-- Retrieval info: PRIVATE: USE_CLKENA4 STRING "0"
-- Retrieval info: PRIVATE: MIRROR_CLK2 STRING "0"
-- Retrieval info: PRIVATE: PHASE_SHIFT_UNIT2 STRING "deg"
-- Retrieval info: PRIVATE: DUTY_CYCLE1 STRING "50.00000000"
-- Retrieval info: PRIVATE: PHASE_SHIFT1 STRING "0.00000000"
-- Retrieval info: PRIVATE: MULT_FACTOR1 NUMERIC "1"
-- Retrieval info: PRIVATE: JUMP2PAGE0 STRING "□"
-- Retrieval info: PRIVATE: TIME_SHIFT0 STRING "0.00000000"
-- Retrieval info: PRIVATE: STICKY_CLK0 STRING "1"
-- Retrieval info: PRIVATE: BANDWIDTH STRING "1.000"
-- Retrieval info: PRIVATE: BANDWIDTH_USE_CUSTOM STRING "0"
-- Retrieval info: PRIVATE: USE_CLKENA5 STRING "0"
-- Retrieval info: PRIVATE: MIRROR_CLK3 STRING "0"
-- Retrieval info: PRIVATE: PHASE_SHIFT_UNIT3 STRING "deg"
-- Retrieval info: PRIVATE: DUTY_CYCLE2 STRING "50.00000000"
-- Retrieval info: PRIVATE: PHASE_SHIFT2 STRING "0.00000000"
-- Retrieval info: PRIVATE: MULT_FACTOR2 NUMERIC "1"
-- Retrieval info: PRIVATE: JUMP2PAGE1 STRING "□"
-- Retrieval info: PRIVATE: TIME_SHIFT1 STRING "0.00000000"
-- Retrieval info: PRIVATE: STICKY_CLK1 STRING "1"
-- Retrieval info: PRIVATE: SPREAD_FREQ STRING "300.000"
-- Retrieval info: PRIVATE: BANDWIDTH_FEATURE_ENABLED STRING "1"
-- Retrieval info: PRIVATE: LONG_SCAN_RADIO STRING "1"
-- Retrieval info: PRIVATE: USE_CLKENA6 STRING "0"
-- Retrieval info: PRIVATE: MIRROR_CLK4 STRING "0"
-- Retrieval info: PRIVATE: PHASE_SHIFT_UNIT4 STRING "deg"
-- Retrieval info: PRIVATE: DUTY_CYCLE3 STRING "50.00000000"
-- Retrieval info: PRIVATE: PHASE_SHIFT3 STRING "0.00000000"
-- Retrieval info: PRIVATE: MULT_FACTOR3 NUMERIC "1"
-- Retrieval info: PRIVATE: JUMP2PAGE2 STRING "General/Modes"
-- Retrieval info: PRIVATE: TIME_SHIFT2 STRING "0.00000000"
-- Retrieval info: PRIVATE: STICKY_CLK2 STRING "0"
-- Retrieval info: PRIVATE: USE_CLK0 STRING "1"
-- Retrieval info: PRIVATE: INCLK1_FREQ_EDIT_CHANGED STRING "1"
-- Retrieval info: PRIVATE: SCAN_FEATURE_ENABLED STRING "1"
-- Retrieval info: PRIVATE: ZERO_DELAY_RADIO STRING "0"
-- Retrieval info: PRIVATE: PLL_PFDENA_CHECK STRING "0"
-- Retrieval info: PRIVATE: USE_CLKENA7 STRING "0"
-- Retrieval info: PRIVATE: MIRROR_CLK5 STRING "0"
-- Retrieval info: PRIVATE: PHASE_SHIFT_UNIT5 STRING "deg"
-- Retrieval info: PRIVATE: DUTY_CYCLE4 STRING "50.00000000"
-- Retrieval info: PRIVATE: PHASE_SHIFT4 STRING "0.00000000"
-- Retrieval info: PRIVATE: MULT_FACTOR4 NUMERIC "1"
-- Retrieval info: PRIVATE: JUMP2PAGE3 STRING "General/Modes"
-- Retrieval info: PRIVATE: TIME_SHIFT3 STRING "0.00000000"
-- Retrieval info: PRIVATE: STICKY_CLK3 STRING "0"
-- Retrieval info: PRIVATE: USE_CLK1 STRING "1"
-- Retrieval info: PRIVATE: CREATE_CLKBAD_CHECK STRING "0"
-- Retrieval info: PRIVATE: INCLK1_FREQ_EDIT STRING "50.000"
-- Retrieval info: PRIVATE: CUR_DEDICATED_CLK STRING "c1"
-- Retrieval info: PRIVATE: PLL_FASTPLL_CHECK NUMERIC "0"
-- Retrieval info: PRIVATE: USE_CLKENA8 STRING "0"
-- Retrieval info: PRIVATE: MIRROR_CLK6 STRING "0"
-- Retrieval info: PRIVATE: PHASE_SHIFT_UNIT6 STRING "deg"
-- Retrieval info: PRIVATE: DUTY_CYCLE5 STRING "50.00000000"
-- Retrieval info: PRIVATE: PHASE_SHIFT5 STRING "0.00000000"
-- Retrieval info: PRIVATE: MULT_FACTOR5 NUMERIC "1"
-- Retrieval info: PRIVATE: JUMP2PAGE4 STRING "General/Modes"
-- Retrieval info: PRIVATE: TIME_SHIFT4 STRING "0.00000000"
-- Retrieval info: PRIVATE: STICKY_CLK4 STRING "0"
-- Retrieval info: PRIVATE: USE_CLK2 STRING "0"
-- Retrieval info: PRIVATE: ACTIVECLK_CHECK STRING "0"
-- Retrieval info: PRIVATE: BANDWIDTH_FREQ_UNIT STRING "MHz"
-- Retrieval info: PRIVATE: INCLK0_FREQ_UNIT_COMBO STRING "MHz"
-- Retrieval info: PRIVATE: USE_CLKENA9 STRING "0"
-- Retrieval info: PRIVATE: MIRROR_CLK7 STRING "0"
-- Retrieval info: PRIVATE: PHASE_SHIFT_UNIT7 STRING "deg"
-- Retrieval info: PRIVATE: DUTY_CYCLE6 STRING "50.00000000"
-- Retrieval info: PRIVATE: PHASE_SHIFT6 STRING "0.00000000"
-- Retrieval info: PRIVATE: MULT_FACTOR6 NUMERIC "1"
-- Retrieval info: PRIVATE: JUMP2PAGE5 STRING "General/Modes"
-- Retrieval info: PRIVATE: TIME_SHIFT5 STRING "0.00000000"
-- Retrieval info: PRIVATE: STICKY_CLK5 STRING "0"
-- Retrieval info: PRIVATE: USE_CLK3 STRING "0"
-- Retrieval info: PRIVATE: GLOCKED_MODE_CHECK STRING "0"
-- Retrieval info: PRIVATE: NORMAL_MODE_RADIO STRING "1"
-- Retrieval info: PRIVATE: CUR_FBIN_CLK STRING "e0"
-- Retrieval info: PRIVATE: MIRROR_CLK8 STRING "0"
-- Retrieval info: PRIVATE: PHASE_SHIFT_UNIT8 STRING "deg"
-- Retrieval info: PRIVATE: DUTY_CYCLE7 STRING "50.00000000"
-- Retrieval info: PRIVATE: PHASE_SHIFT7 STRING "0.00000000"
-- Retrieval info: PRIVATE: MULT_FACTOR7 NUMERIC "1"
-- Retrieval info: PRIVATE: JUMP2PAGE6 STRING "□"
-- Retrieval info: PRIVATE: TIME_SHIFT6 STRING "0.00000000"
-- Retrieval info: PRIVATE: STICKY_CLK6 STRING "0"
-- Retrieval info: PRIVATE: USE_CLK4 STRING "0"
-- Retrieval info: PRIVATE: DIV_FACTOR0 NUMERIC "50"
-- Retrieval info: PRIVATE: INCLK1_FREQ_UNIT_CHANGED STRING "1"
-- Retrieval info: PRIVATE: EXT_FEEDBACK_RADIO STRING "0"
-- Retrieval info: PRIVATE: MIRROR_CLK9 STRING "0"
-- Retrieval info: PRIVATE: PHASE_SHIFT_UNIT9 STRING "deg"
-- Retrieval info: PRIVATE: DUTY_CYCLE8 STRING "50.00000000"
-- Retrieval info: PRIVATE: PHASE_SHIFT8 STRING "0.00000000"
-- Retrieval info: PRIVATE: MULT_FACTOR8 NUMERIC "1"
-- Retrieval info: PRIVATE: JUMP2PAGE7 STRING "General/Modes"
-- Retrieval info: PRIVATE: TIME_SHIFT7 STRING "0.00000000"
-- Retrieval info: PRIVATE: STICKY_CLK7 STRING "0"
-- Retrieval info: PRIVATE: USE_CLK5 STRING "0"
-- Retrieval info: PRIVATE: DIV_FACTOR1 NUMERIC "50"
-- Retrieval info: PRIVATE: CLKLOSS_CHECK STRING "0"
-- Retrieval info: PRIVATE: BANDWIDTH_USE_AUTO STRING "1"
-- Retrieval info: PRIVATE: SHORT_SCAN_RADIO STRING "0"
-- Retrieval info: PRIVATE: DUTY_CYCLE9 STRING "50.00000000"
-- Retrieval info: PRIVATE: PHASE_SHIFT9 STRING "0.00000000"
-- Retrieval info: PRIVATE: MULT_FACTOR9 NUMERIC "1"
-- Retrieval info: PRIVATE: JUMP2PAGE8 STRING "General/Modes"
-- Retrieval info: PRIVATE: TIME_SHIFT8 STRING "0.00000000"
-- Retrieval info: PRIVATE: STICKY_CLK8 STRING "0"
-- Retrieval info: PRIVATE: USE_CLK6 STRING "0"
-- Retrieval info: PRIVATE: DIV_FACTOR2 NUMERIC "1"
-- Retrieval info: PRIVATE: CLKSWITCH_CHECK STRING "0"
-- Retrieval info: PRIVATE: SPREAD_FREQ_UNIT STRING "KHz"
-- Retrieval info: PRIVATE: PLL_ENA_CHECK STRING "0"
-- Retrieval info: PRIVATE: INCLK0_FREQ_EDIT STRING "50.000"
-- Retrieval info: PRIVATE: JUMP2PAGE9 STRING "General/Modes"
-- Retrieval info: PRIVATE: TIME_SHIFT9 STRING "0.00000000"
-- Retrieval info: PRIVATE: STICKY_CLK9 STRING "0"
-- Retrieval info: PRIVATE: USE_CLK7 STRING "0"
-- Retrieval info: PRIVATE: DIV_FACTOR3 NUMERIC "1"
-- Retrieval info: PRIVATE: CNX_NO_COMPENSATE_RADIO STRING "0"
-- Retrieval info: PRIVATE: INT_FEEDBACK__MODE_RADIO STRING "1"
-- Retrieval info: PRIVATE: USE_CLK8 STRING "0"
-- Retrieval info: PRIVATE: DIV_FACTOR4 NUMERIC "1"
-- Retrieval info: PRIVATE: JUMP2PAGE STRING "Clock switchover"
-- Retrieval info: PRIVATE: USE_CLK9 STRING "0"
-- Retrieval info: PRIVATE: DIV_FACTOR5 NUMERIC "1"
-- Retrieval info: PRIVATE: PRIMARY_CLK_COMBO STRING "inclk0"
-- Retrieval info: PRIVATE: CREATE_INCLK1_CHECK STRING "0"
-- Retrieval info: PRIVATE: SACN_INPUTS_CHECK STRING "0"
-- Retrieval info: PRIVATE: DEV_FAMILY STRING "Stratix"
-- Retrieval info: PRIVATE: DIV_FACTOR6 NUMERIC "50"
-- Retrieval info: PRIVATE: SWITCHOVER_COUNT_EDIT NUMERIC "1"
-- Retrieval info: PRIVATE: SWITCHOVER_FEATURE_ENABLED STRING "1"
-- Retrieval info: PRIVATE: BANDWIDTH_PRESET STRING "Low"
-- Retrieval info: PRIVATE: GLOCKED_FEATURE_ENABLED STRING "1"
-- Retrieval info: PRIVATE: DIV_FACTOR7 NUMERIC "1"
-- Retrieval info: PRIVATE: USE_CLKENA0 STRING "0"
-- Retrieval info: PRIVATE: DIV_FACTOR8 NUMERIC "1"
-- Retrieval info: PRIVATE: USE_CLKENA1 STRING "0"
-- Retrieval info: PRIVATE: CLKBAD_SWITCHOVER_CHECK STRING "0"
-- Retrieval info: PRIVATE: BANDWIDTH_USE_PRESET STRING "0"
-- Retrieval info: PRIVATE: DEVICE_FAMILY NUMERIC "9"
-- Retrieval info: LIBRARY: altera_mf altera_mf.altera_mf_components.all
-- Retrieval info: CONSTANT: CLK1_DIVIDE_BY NUMERIC "50"
-- Retrieval info: CONSTANT: BANDWIDTH_TYPE STRING "AUTO"
-- Retrieval info: CONSTANT: CLK1_PHASE_SHIFT STRING "0"
-- Retrieval info: CONSTANT: CLK0_DUTY_CYCLE NUMERIC "50"
-- Retrieval info: CONSTANT: LPM_TYPE STRING "altpll"
-- Retrieval info: CONSTANT: CLK0_MULTIPLY_BY NUMERIC "1"
-- Retrieval info: CONSTANT: LOCK_LOW NUMERIC "5"
-- Retrieval info: CONSTANT: INVALID_LOCK_MULTIPLIER NUMERIC "5"
-- Retrieval info: CONSTANT: INCLK0_INPUT_FREQUENCY NUMERIC "20000"
-- Retrieval info: CONSTANT: GATE_LOCK_SIGNAL STRING "NO"
-- Retrieval info: CONSTANT: CLK0_DIVIDE_BY NUMERIC "50"
-- Retrieval info: CONSTANT: CLK1_DUTY_CYCLE NUMERIC "50"
-- Retrieval info: CONSTANT: PLL_TYPE STRING "AUTO"
-- Retrieval info: CONSTANT: VALID_LOCK_MULTIPLIER NUMERIC "1"
-- Retrieval info: CONSTANT: CLK1_MULTIPLY_BY NUMERIC "1"
-- Retrieval info: CONSTANT: CLK0_TIME_DELAY STRING "0"
-- Retrieval info: CONSTANT: SPREAD_FREQUENCY NUMERIC "0"
-- Retrieval info: CONSTANT: OPERATION_MODE STRING "NORMAL"
-- Retrieval info: CONSTANT: LOCK_HIGH NUMERIC "1"
-- Retrieval info: CONSTANT: COMPENSATE_CLOCK STRING "CLK1"
-- Retrieval info: CONSTANT: CLK1_TIME_DELAY STRING "0"
-- Retrieval info: CONSTANT: CLK0_PHASE_SHIFT STRING "0"
-- Retrieval info: USED_PORT: c0 0 0 0 0 OUTPUT VCC "c0"
-- Retrieval info: USED_PORT: @clk 0 0 6 0 OUTPUT VCC "@clk[5..0]"
-- Retrieval info: USED_PORT: c1 0 0 0 0 OUTPUT VCC "c1"
-- Retrieval info: USED_PORT: inclk0 0 0 0 0 INPUT GND "inclk0"
-- Retrieval info: USED_PORT: locked 0 0 0 0 OUTPUT GND "locked"
-- Retrieval info: USED_PORT: @extclk 0 0 4 0 OUTPUT VCC "@extclk[3..0]"
-- Retrieval info: USED_PORT: @inclk 0 0 2 0 INPUT VCC "@inclk[1..0]"
-- Retrieval info: CONNECT: @clkena 0 0 1 1 VCC 0 0 0 0
-- Retrieval info: CONNECT: locked 0 0 0 0 @locked 0 0 0 0
-- Retrieval info: CONNECT: @inclk 0 0 1 0 inclk0 0 0 0 0
-- Retrieval info: CONNECT: @extclkena 0 0 1 1 GND 0 0 0 0
-- Retrieval info: CONNECT: @clkena 0 0 1 4 GND 0 0 0 0
-- Retrieval info: CONNECT: c0 0 0 0 0 @clk 0 0 1 0
-- Retrieval info: CONNECT: c1 0 0 0 0 @clk 0 0 1 1
-- Retrieval info: CONNECT: @extclkena 0 0 1 2 GND 0 0 0 0
-- Retrieval info: CONNECT: @clkena 0 0 1 5 GND 0 0 0 0
-- Retrieval info: CONNECT: @clkena 0 0 1 2 GND 0 0 0 0
-- Retrieval info: CONNECT: @clkena 0 0 1 0 VCC 0 0 0 0
-- Retrieval info: CONNECT: @inclk 0 0 1 1 GND 0 0 0 0
-- Retrieval info: CONNECT: @extclkena 0 0 1 3 GND 0 0 0 0
-- Retrieval info: CONNECT: @extclkena 0 0 1 0 GND 0 0 0 0
-- Retrieval info: CONNECT: @clkena 0 0 1 3 GND 0 0 0 0
```

# Vita

Calum Johnson was born on September 11, 1979, in Iowa City, Iowa. His family moved to Nashville, Tennessee when Calum was 1 year old where his father pursued a Doctorate of Philosophy at Vanderbilt University. Calum graduated from Martin Luther King Jr. Magnet High School for Health Sciences and Engineering in 1997 and went on to attend Tennessee Technological University. While at TTU his studies emphasized digital design and micro electrical mechanical systems. He graduated from TTU in 2001 with a Bachelor of Science degree in Electrical Engineering.

After receiving his bachelor's degree, Mr. Johnson continued with graduate studies at TTU for one semester. He then spent a summer as an intern at the Engineering Science and Technology division of Oak Ridge National Laboratory. The following semester he transferred to the University of Tennessee and began studying analog systems design while continuing to work at ORNL as a graduate research assistant.

In 2004 Mr. Johnson went to work for Atmospheric Glow Technologies in Knoxville, Tennessee, where he is currently still employed. There he is responsible for the design and implementation of data acquisition and control systems for many different test apparatuses involving atmospheric plasma. He is also responsible for firmware development for the company's commercial atmospheric plasma power supplies.