PERFORMANCE OF A HIGH-SPEED TRANSCUTANEOUS LINK WITH ERROR CORRECTION CODING

by

David Toribio

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Electrical Engineering

Approved:

Dr. Chris Winstead Major Professor	Dr. Reyhan Baktur Committee Member
Dr. Edmund Spencer Committee Member	Dr. Mark R. McLellan Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2013

Abstract

Performance of a High-Speed Transcutaneous Link with Error Correction Coding

by

David Toribio, Master of Science

Utah State University, 2013

Major Professor: Dr. Chris Winstead
Department: Electrical and Computer Engineering

A Register Transfer Level (RTL) design that integrates a figure-8 multi-band inductive link (MIL), a bi-level pulse harmonic modulation (PHM) system, and a Gallager A decoder is presented and verified. Integrating this MIL with a PHM system can lead to optimizing power efficiency, data rate, and transmitter power consumption in near field transcutaneous wireless communication systems for cortical implants (CI). Also, a technique to increase the data rate of PHM systems based in multi-level transmission (MLT) is presented. MLT is an attractive solution to increasing data rates in PHM systems while meeting power consumption constraints, since it does not require increasing the frequency of the harmonics generated in the PHM receiver. Verilog Analog Mixed Signal (AMS) was used to verify these systems. The distance between external and implanted coils was modeled to be 10 mm. From the power interference characterization of the bi-level system, it was observed that for every value of the coupling coefficient between the power transmitter coil and data receiver coil (k_{14}), there is a different comparator reference voltage that can minimize the bit error rate (BER) of the system and optimize decoder performance. Moreover, it is shown that as k_{14} increases, this optimal reference voltage also increases. Therefore, it is expected that a control system that could adapt the comparator reference voltages to changes in (k_{14}), could enhance robustness of PHM systems against factors that can increase MIL power

interference, such as misalignments. From the verification of the MLT technique, it was observed that the optimal delay between initiation and suppression pulses (t_d) of a PHM system varies with respect to initiation pulse amplitudes, contrary to what had been stated in the past in previous PHM system verifications. Furthermore, it is concluded that, unless the pulse pattern generator (PPG) is designed to vary t_d according to the initiation pulse transmitted, MLT-based PHM systems would be too vulnerable to non-idealities, such as noise and misalignments, preventing its feasibility. From the transmitter (TX) clock jitter characterizations, it was concluded that as the PHM levels of transmission increase, systems become more sensitive to jitters. Finally, the Gallager A decoder was found powerful in enhancing the robustness of PHM systems against power interference, TX clock jitter, and noise.

(75 pages)

Public Abstract

Performance of a High-Speed Transcutaneous Link with Error Correction Coding

by

David Toribio, Master of Science

Utah State University, 2013

Major Professor: Dr. Chris Winstead
Department: Electrical and Computer Engineering

Cortical implants (CI) can be used in medicine to treat neurological disorders, and to serve as substitutes of damaged organs in the nervous system. Currently, active research is being developed in the wireless data and power transfer to CIs, in order to avoid frequent surgical interventions to replace batteries and potential paths of infection due to wires breaching the skin. In this paper, we consider performance optimization strategies for a pulse-based wireless link that have been proposed recently. We consider two enhancements that may allow for increased throughput in this system. First, a low-power error-correcting code is used to improve the system's robustness against non-ideal factors. Second, the system is adapted to a communication scheme that doubles its speed.

Acknowledgments

I would especially like to thank Dr. Chris Winstead for introducing me to the neuroprostheses research community, and for his great mentoring. I would also like to seriously thank my family down in The Dominican Republic, for calling me every day just to see how I am doing, and helping me every day to get around the obstacles of life. There is no way I would have been able to do this without their support. Finally, I would like to thank The Ministry of Higher Education of The Dominican Republic, for having provided me full financial assistance to accomplish my undergraduate and graduate academic goals.

David Toribio

Contents

List of Tables

List of Figures

Acronyms

ASK	Amplitude Shift Keying
BER	Bit Error Rate
CI	Cortical Implant
ECC	Error Correction Coding
IC	Integrated Circuits
IP	Initiation Pulse
IR	Impulse Radio
ISI	Inter-Symbol Interference
LNA	Low-Noise Amplifier
LPF	Low-Pass Filter
LPF-OV	LPF Output Voltage
MIL	Multi-Band Inductive Link
MLT	Multi-Level Transmission
MS	Mixed Signal
PPG	Pulse Pattern Generator
P-P	Peak-to-Peak
RF	Radio Frequency
RMS	Random Mean Square
SNR	Signal-to-Noise Ratio
SP	Suppression Pulse
TX	Transmitter
UWB	Ultra-Wideband

Chapter 1

Introduction

1.1 Research Context

The research presented in this thesis focuses on the development of cortical implants (CIs), which are devices that can substitute a sensory, motor, or cognitive modality that might have been damaged as a result of an injury or a disease.

1.1.1 Sensory Applications

The main sensory applications of CIs are in visual, auditory, and pain relief prostheses.

Visual prostheses can create a sense of image by electrically stimulating neurons in the visual system. They consist of an external imaging system which acquires and processes video. This imaging system then transmits the data to an implant, which maps the image across an array of electrodes. Subsequently, the electrodes stimulate the retina to create an image perception.

Auditory prostheses acquire, process, and convert sound into electrical energy for subsequent delivery to the auditory nerve. They consist of a microphone which receives sound from the external environment and sends it to a processor. The processor digitizes the sound and filters it into separate frequency bands that are sent to the appropriate tonotonic region in the cochlea that approximately corresponds to those frequencies.

CIs-based pain relief prostheses have been investigated as therapeutic solutions to chronic refractory pains. These prostheses stimulate the motor cortex to alleviate pain by means of implanted micro-electrodes, a pulse generator, and an external remote control.

1.1.2 Motor Applications

The main motor applications of CIs being studied are in the cortical control of external

limbs and devices. In this application, the implants consist of an electrode array that detect neural data, and an integrated or external signal processing circuitry that decodes the data and wirelessly sends it to an external device or limb. Researchers have already built implants that allow patients to move cursors, and active research is being pursued to develop implants that would allow people with disabilities control assistive devices such as wheelchairs and robotic arms.

1.1.3 Cognitive Applications

Cognitive applications of CIs seek to restore cognitive function to individuals with brain tissue loss due to injury, disease, or stroke by performing the function of the damaged tissue with integrated circuits. Some applications of these prostheses are in the treatment of Alzheimers Disease, hippocampal deficits, traumatic brain injury, Parkinsons disease, speech deficits, and paralysis. Some facts that can help getting a perspective of the size of the societal impact and market for cognitive applications of CIs are:

- Alzheimers disease is projected to affect more that 107 million people worldwide by the year 2050 [1];
- 1.4 million people in the United States suffer traumatic brain injury [2];
- 7.5 million people in the United States have speech deficits [3];
- 6.5 million people in the United States have suffered stroke [4].

1.2 Challenges

1.2.1 Safety

Tissue surrounding CIs are very sensitive to temperature. Hence, it is critical to minimize CIs power consumption in order to decrease CIs heat dissipation and prevent tissue damages. Many CIs applications, especially visual prostheses, require high data rates in order to operate properly; therefore, their power consumption can be very high. A percutaneous wire could be used to transfer power and data to these CIs applications by means of

breaking the skin to reach the stimulator. However, breaching the skin provides a potential path for infection to enter, risking the safety of the individual as well as the implant [5]. Another alternative is to implant a battery, however, high data rate demanding applications often require more amp-hours than a battery can provide without frequent re-implantation or recharging [6]. Therefore, several neuroscientists have been motivated to engage in research on inductive links (e.g., a pair of resonant coils separated by a distance) for power and data transmission to CIs. Active research is being developed in the design of low-power wireless communication systems for CIs. However, one tradeoff of many power optimization techniques in CIs communication systems is a lowering in their data rate.

1.2.2 Signal-to-Noise Ratio and Clock Jitter

The tissue inbetween the external transmitter and the implant's receiver coils attenuates the signal transmitted and deposits random noise on it, decreasing its signal-to-noise ratio (SNR) as the distance between the coils increase. Also, the low-noise amplifier (LNA) at the implant's receiver and the passive components of the communication systems deposit random noise in the transmitted signal. The degradation of the transmitted signal SNR due to the tissue, LNA, and passive components of the systems induce an increment in the bit-error-rate (BER) of the communication systems, and hence, a decrement in the data rate of CIs wireless systems. Wireless communication systems considered for CIs are synchronous systems; therefore, a digital clock controls their states. Digital clocks are never entirely precise, because their duty cycle randomly changes in each cycle. This instability in digital clocks is referred to as clock jitter. Increments in clock random mean square (RMS) jitter generally result in an increase of the BER, and a decrease of the data rate of CIs communication systems.

1.2.3 Radio Frequency (RF) Integrated Circuits (IC) Functional Failures

RF-ICs used in CIs, usually involves the interaction of analog and digital systems, hence, their architecture are based on mixed-signal (MS) systems. Studies have shown that, in current design flows of MS-ICs, the probability of producing a functional failure

due to design error during the first tape out is of 45% [7]. One reason that MS-ICs are at high risk of functional failure is due to the lack of tools for analog systems verification [7]. MATLAB has been used in the past to model the analog portions of wireless transceivers. However, at this high design level, implementation effects, such as, signal transients and parasitics cannot be modeled accurately [8]. Also, another disadvantage of designs at this level is that they are not supported by transistor level simulators, preventing their co-simulation with transistor level designs [9]. Co-simulations of transistor level designs with higher design levels are desired, because they allow the designer to implement only the critical design areas at the transistor level, while representing the non-critical design areas at higher design levels, speeding up MS-RF-IC verifications. Transistor level simulations also have been used to verify analog portions of wireless transceivers. However, due to the complexity (e.g., thousands of modes of operations and settings) of modern MS-ICs, transistor level simulations of a single IC mode can sometimes take a week or more, which may result highly impractical [9].

Chapter 2

Background

2.1 Multi-Band Inductive Link (MIL)

Many researchers have considered using only one inductive link, also known as single band inductive links, for simultaneous inductive transmission of power and data [10]. However, since maximizing power efficiency leads to lowering the transmitted signal frequency, while maximizing data rate leads to increasing the transmitted signal frequency, it becomes challenging to maximize these conflicting parameters by using only one link [11]. The reason that low signal frequency is desired for maximizing power efficiency in inductive links is because the tissue in between the coils of the link absorbs high frequency components [11].

Several researchers have been attracted by the idea of employing MILs for the simultaneous power and data transmission to cortical implants, in order to satisfy the conflictive requirements of maximizing power efficiency and data rate. Some of the main issues encountered in using MILs are: increment in size of the implant receiver, and a rise of BER due to interference between the power signal and the data signal.

There have been different approaches to minimize the interference between power and data in MILs. One approach consists on positioning data coils on top, and vertical to the power coils, in order to minimize their mutual inductances [11]. Another approach consists on placing figure-8 data coils on top and in parallel to power coils [5]. In these types of data coils, the direction of the windings in each loop is chosen so that when the coil is exposed to an external field symmetrical to both loops (such as the field created by the power coils), induced currents will cancel each other out. A different approach consists in using co-planar coils [10]. In this approach, data and power coils are placed in the same plane (hence, the term co-planar), separated by a distance, rather than on top of each other.

2.2 Pulse Harmonic Modulation (PHM)

There have been different communication system architectures that have been investigated for CIs. Amplitude Shift Keying (ASK) has been popular in the past because of its simple modulation and demodulation technique [11]. This method, however, is not robust against coupling variations and faces major limitations in high bandwidth data transmission. These limitations are due to the need of high order filters with sharp cutoff that cannot be easily integrated in this low frequency end of Radio Frequency (RF) applications [11]. Another data transmission technique studied has been phase-coherent frequency shift keying (pcFSK) which shapes the inductive link passband spectrum to reach 2.5 Mbps. However, this technique occupies a wide bandwidth (>5MHZ) [11]. Also, other techniques have used binary and quadrature phase shift keying, which have fewer limitations than ASK and FSK [11]. However, all of these techniques use a carrier signal to transmit data, leading to high power consumption in the data link, especially in the transmitter side.

Load Shift Keying (LSK) based passive telemetry can be used to transmit ASK signals, and can result in lower power consumption than voltage controlled oscillators based active telemetry. However, in order for LSK to function properly, strong coupling is required [11], which can increase the size of the inductive link and prevent its use in some CI applications.

After the Federal Communications Commission deregulated the use of Ultra-Wideband (UWB) for low-power communications, a technique known as Impulse Radio (IR)-UWB soon became popular for low power communication systems [12]. The reason it became popular is because this is a carrier-less type of communication system and therefore its power consumption, especially in the transmitter side, is lower than carrier based communication system. In this technique, a bit 1 or a bit 0 is transmitted depending on if a pulse is transmitted or not. Equation (2.1) shows the relationship between the center frequency (f_r) of the signal generated in the receiver data coil and the time width of the pulses (t_{pw}) transmitted [13].

$$f_r = \frac{\sqrt{2}}{\Pi t_{pw}} \tag{2.1}$$

Since this technique was intended to work in far field communication systems in the

range of 3.1-10.6 GHz it was undesirable for transcutaneous communication because signals at these frequencies are highly absorbable by water and cannot penetrate through the tissue [14].

Some researchers began applying the IR-UWB technique using inductive coupling for transferring data in applications such as chip stacking and multimedia [13]. These applications employ inductive links with very large bandwidths in order to allow the high frequency components to penetrate the tissue [15]. Otherwise, ringing that extend beyond the bit period are produced increasing inter-symbol inteference (ISI), and hence, BER. One method of maintaining a large inductive link bandwidth (in the hundreads of MHZ) requires adding parallel or series resistors which leads to lowering the Q of the link [16]. Degrading the Q of a link is undesirable because the link becomes more vulnerable to external interferences and it also increases the SNR of the transmitted signal [17].

A communication system that prevents ISI without having to maintain a large inductive link bandwidth was proposed by Inanlou and Ghovanloo [15], and it is known as Pulse Harmonic Modulation (PHM). In this technique, two pulses with specific time delays between them are transmitted every bit period, instead of just one pulse such as in IR-UWB. The second pulse that is transmitted (the suppression pulse) produces an impulse response in the link that opposes in phase to the impulse response of the first pulse (the initiation pulse). This opposition in phase between the first and the second impulse response attenuates the signal in the receiver, which prevents the ISI that would have otherwise been caused by the low inductive link bandwidth.

2.3 Error Correction Coding (ECC)

Unfortunately, there has not been extensive research up to date in evaluating the feasibility and performance of ECC in CIs communication systems. However, it has been identified that ECC could be beneficial to CIs communication systems, since ECC could contribute in the strengthening of CIs communication reliability while reducing transmitter power consumption due to coding gain [18]. Nevertheless, one clear disadvantage of including ECC in CIs communication systems is that power consumption in the receiver is

increased due to the decoder [18]. Therefore, ECC could be especially advantageous for CIs when used in the uplink communication system of CIs (e.g., transmitting from inside to the outside of the body), since in this case the receiver and decoder would be placed outside of the body, and would not be constrained by heating temperature.

A Hamming-based ECC was proposed by Arabi and Sawan [19], as an optimal type of ECC for CIs applications because of its simple hardware implementation, strong reliability, and low-power consumption. However, this design only allows for single error detection and correction per frame. It is not reported by Arabi and Sawan [19], the tradeoffs between BER improvements, rise of power consumption due to the decoder, and CI size increment that characterize this Hamming ECC design. However, it is anticipated that it would be useful to modify this design or adapt a more complex ECC to the power consumption constraints imposed by a CI environment, in order to allow CI communication systems to detect multiple errors per frame, and become more resistant to non-idealities [19].

Convolutional ECC are more complex than the decoder proposed by Arabi and Sawan [19], and they have the capability of detecting multiple error per frames. They have been widely used in low-power wireless applications because of their relatively low complexity and power consumption [20]. The most common implementations of convolutional decoders used have been purely digital; however, analog implementations can result in lower power consumption and smaller size at the expense of a reduction in speed [20]. Therefore, analog implementations may be desired for lower data rate CI applications. It is proposed by Tomatsopoulos and Demosthenous [20], an analog convolutional decoder that operates at 1Mbps, occupies $0.5mm^2$ and consumes 2.45mW, which confirms the suitability of these types of decoders for low data rate CI applications.

Analog designs of other types of decoders, such as Hamming codes [21] and iterative decoders for block codes [22] have also been implemented in the past, exhibiting very large coding gains at the expense of decoder complexity and power consumption [20]. Therefore, it is anticipated that these types of decoders would be especially advantageous in the uplink communication system of CIs.

2.4 Multi-Level Transmission (MLT)

Up to date, most of the CI wireless communication techniques that have been reported use bi-level transmission. The term bi-level arises because only two symbols are used for transmission. Therefore, only one bit per symbol is transmitted under this type of transmission.

In MLT, however, more than two symbols are used for transmission, and hence, more than one bit can be transmitted per symbol. MLT has been implemented in an LSK system for biomedical implants by Xu et al. [23]. This system incorporates one more resistance and switch than its bi-level counterpart, in order to vary the impedance across the transmitter coil up to four impedance levels, and hence allowing the transmission of two bits per level. Therefore, this four-level LSK system doubles the data rate of its bi-level version.

Even though LSK is an attractive low-power alternative to voltage controlled oscillators based transceivers, the need of strong coils coupling for LSK proper operation may still make PHM more desirable for CI applications [15].

The PHM system implemented by Inanlou and Ghovanloo [15], is able to transmit one bit per 200 ns, while generating harmonics in the receiver of 37.5 MHz. The data rate of this system was approximately doubled in the design reported by Inanlou et al. [24], by means of shortening the width of the pulses transmitted, and hence, increasing the frequency of the harmonics generated in the receiver coil. Therefore, employing this technique leads to an increment in power dissipation in the system.

Another alternative to increase the data rate of PHM is to enable MLT. One method to enable MLT in PHM systems is to, instead of using on-off keying (OOK) modulation, as in the systems described by Inanlou et al. [15,24] allow the system to employ ASK modulation. In order to use ASK in PHM systems, the initiation pulses would be allowed to take on more than two amplitudes (e.g., more than two symbols would be used). Hence, allowing PHM systems to transmit more than one bit per symbol, increasing its data rate.

2.5 RF-Register Transfer Level (RTL) Design

An MS system verification methodology was proposed by Chang and Kundert [9], in

order to reduce the time to reach an MS-IC design that meets target specifications, and to reduce the risk of MS-IC functional failures. This methodology suggests the use of hardware description languages (HDL) and a mixed-signal simulator, for the RTL design and functional verification of MS systems. An advantage of being able to verify an HDL-based RTL design of an MS system is that functional errors can be detected early in the IC development process, which can save money and time. These savings can occur since functional error corrections may be more difficult and time consuming at the transistor level design [9], and IC re-spins can be quite expensive nowadays [8]. Also, another advantage of RTL designs is that they can ease the re-use of MS-IC designs. Therefore, they can allow better communication between different engineering departments and research groups, reducing the time to reach an IC optimal design.

An HDL that is currently in use for analog RTL design is Verilog-A. For digital RTL design, Verilog-HDL and VHDL are most common, but also other languages such as SystemVerilog and SystemC could be used as well [9]. There are also HDLs that combine analog HDLs with digital HDLs, and hence, can be used for the modeling of MS systems such as Verilog-AMS, VHDL-AMS and SystemC-AMS.

MS simulators that can support MS-HDLs can be integrated or glued [25]. An integrated MS simulator bolts together an established circuit simulator with an established logic simulator. An example of an integrated MS simulator is Cadences AMS Designer which tightly merges the Spectre circuit simulator with the NC-SIM logic simulator. A glued simulator, such as Mentors ADVance MS, adds an event-driven kernel to an established circuit simulator.

Chapter 3

Research Overview

3.1 Objectives

1. To functionally verify an RTL design of an ECC-based CI communication system.
2. To characterize the BER of bi-level PHM systems with respect to noise, TX clock jitter, and MIL power interference.
3. To assess the performance of ECC in reducing the sensitivity of bi-level PHM systems to SNR, TX clock jitter, and MIL power interference.
4. To assess the feasibility and the performance of MLT in PHM systems.
5. To characterize the BER of MLT PHM systems with respect to noise and TX clock jitter.
6. To determine the performance of ECC in reducing the sensitivity of MLT-based PHM systems to SNR and TX clock jitter.

3.2 Multi-Band Inductive Link

As mentioned in the background section, three different types of MILs have been designed for the purposes of simultaneously maximizing power efficiency and data rate in CIs wireless systems. These are co-planar, vertical, and figure-8 MILs.

Although co-planar MILs may exhibit stronger direct coupling coefficients than vertical and figure-8 coils, they may experience a more limited range of CI applications due to their larger size.

From a detailed comparison of the performances of figure-8 and vertical MILs presented by Jow and Ghovanloo [5], it can be observed that when no misalignment is present, vertical

MILs show lower interference than figure-8 MILs. However, figure-8 MILs reveal stronger robustness against misalignment than vertical MILs. Therefore, the use of these types of MILs will depend on the CI application.

For example, in stable settings, such as in neuroscience research experiments, vertical MILs would be more effective. Yet, in applications where CIs will be more exposed to motion, such as visual prostheses, figure-8 MILs would be desired.

The target applications of the CI project developed in this thesis are neuro-prostheses that could be used as therapeutic devices of neurological diseases. Therefore, it is likely that these neuro-prostheses would be implanted in moving subjects, and that they would be very size constrained. For these reasons, it was decided to create a figure-8 MIL verilog-AMS model over a co-planar or vertical MIL model, and use this model for pursuing the objectives 1-6.

3.3 Pulse Harmonic Modulation

A long-term goal of the project started with this thesis is to identify CI wireless communication systems that are ideal in power consumption, size and data rate. PHM exhibits very low-power consumption, and since its transmitter and receiver are of low complexity, its size can be adequate for large range of CI applications. Moreover, multi-level transmission could allow PHM to achieve suitable data rates for high-speed CI applications such as visual prostheses. Therefore, since PHM has strong chances of being an ideal CI wireless communication system, a Verilog AMS model of a PHM system was created and attached to the figure-8 MIL model to pursue objectives 1-6.

A figure-8 MIL design has been characterized by Jow and Ghovanloo [5], and a PHM system has been presented by Inanlou and Ghovanloo [15]. Integrating these designs can lead to a description of the effect that figure-8 MIL power interferences have in PHM BER, and hence, data rates. It is interesting to learn about this MIL power interferences in the data signal since it is likely that this interference will worsen the BERs and data rates of the systems reported by Inanlou et al. [15, 24].

3.4 Error Correction Coding

It is presented in this thesis a characterization of the performance of a Gallager A decoder when attached to a figure-8 MIL-based PHM system. This decoder is of low complexity, and opposed to the Hamming ECC attached to the CI presented by Arabi and Sawan [19], it can allow the detection of multiple errors per frames.

The Gallager A Verilog AMS model is used to pursue objectives 3 and 6, which relate to assessing the performance of ECC in reducing PHM sensitivities to SNR, jitter, and MIL power interference. Improving PHM robustness against to SNR, jitter, and MIL power interference can increase PHM data rates and decrease PHM vulnerability to inductive link coils separation and coil misalignments.

3.5 Multi-Level Transmission

As part of the research presented in this thesis, the Verilog AMS PHM system attached to the figure-8 MIL and Gallager A ECC was enabled to perform MLT. Adding MLT to PHM allowed it to transmit two bits per TX pulse, doubling the data rate of the system designed by Inanlou and Ghovanloo [15]. Moreover, different to the technique described by Inanlou et al. [24], to increase the data rate of the system presented by Inanlou and Ghovanloo [15], MLT does not increase the frequency of the signal generated at the data receiver coil of the MIL, and hence MLT could be a lower power consuming technique to increase PHM data rates than the presented by Inanlou et al. [24].

The MLT Verilog AMS model was used in this thesis to assess the feasibility of MLT in PHM, to determine what conditions need to be met for MLT to work properly in PHM, and to pursue objectives 4, 5, and 6.

3.6 RF-RTL Design

Verilog-AMS and Cadence AMS Designer integrated MS simulator were used for the RTL design and verification of the PHM-based wireless communication system characterized in this thesis to pursue objectives 1-6. The reason an MS HDL and an MS simulator were used is because, since a long-term goal of this thesis project is to fabricate a CI transceiver,

it has been decided it would be useful to completely follow the MS IC development methodology proposed by Chang and Kundert [9]. Also, it is the intent of this thesis to motivate researchers within the CI community to follow this methodology as well. Since CI wireless transceivers design can become rather complex, especially when ECC is employed, it is anticipated that this methodology will help the CI research community to save time in reaching optimal CI-RF designs.

Chapter 4

Experimental Design

4.1 Multi-Band Inductive Link

Figure 4.1 shows a graph of the architecture that was used to implement and validate the figure-8 MIL Verilog-AMS model.

The specifications of the parameters of the architecture of Fig. 4.1 are presented in Table 4.1.

It is stated by Ghovanloo and Atluri [11], that in order to transmit power efficiently to a CI, a low-power signal frequency ($f_p < 1MHz$) would be desired, hence, f_p was selected to be $125\kappa Hz$.

The power link parameters: R1, L1, L2, and k_{12} were selected from the specifications of the power efficient link presented by Ghovanloo and Atluri [11]. Also, the power link capacitances: C1 and C2 were determined using equation (4.1), in order to allow the power link LC tanks to resonate at $125kHz$.

$$f = \frac{1}{2\Pi\sqrt{LC}} \tag{4.1}$$

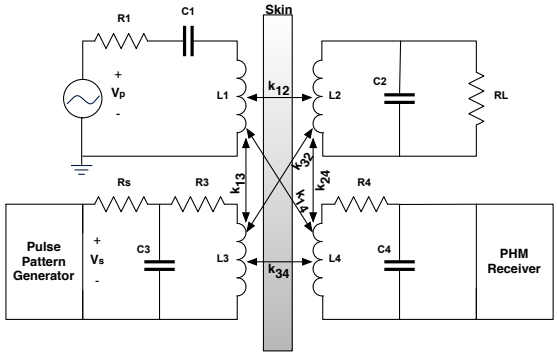

Fig. 4.1: Figure-8 multi-band inductive link.

Table 4.1: Parameters of the system architecture used to validate the figure-8 MIL.

f_p	$125\kappa Hz$	RL	50Ω	R4	$1300m\Omega$
R1	50Ω	Rs	50Ω	C4	82.7ρ
C1	$27.48\eta F$	C3	$225\rho F$	k_{12}	0.16688
L1	$59\mu H$	R3	$412m\Omega$	k_{14}	0.00012
L2	$6.55\mu F$	L3	$105\eta F$	$k_{12}k_{24}$	0.0004
C2	$247.50\eta F$	L4	$281\eta F$	k_{34}	0.011

The data link parameters: R3, C3, L3, L4, and C4 were extracted from the PHM system described by Inanlou and Ghovanloo [15], and the coupling coefficient that affect the data link: k_{14}, $k_{12}k_{24}$, and k_{34} were selected from the figure-8 MIL system presented by Jow and Ghovanloo [5].

The figure-8 MIL was modeled based on the following equations:

$$\begin{aligned} V_1(t) &= L_1\frac{dI_1}{dt} + M_{12}\frac{dI_2}{dt}, \\ V_2(t) &= L_2\frac{dI_2}{dt} + M_{12}\frac{dI_1}{dt}, \\ V_3(t) &= L_3\frac{dI_3}{dt} + M_{34}\frac{dI_4}{dt}, \\ V_4(t) &= L_4\frac{dI_4}{dt} + M_{14}\frac{dI_1}{dt} + M_{24}\frac{dI_2}{dt} + M_{34}\frac{dI_3}{dt}. \end{aligned} \tag{4.2}$$

In equations (4.2), the current that flows through L_a is represented as I_a, and in the same manner, the voltage across L_a is represented as $V_a(t)$. Also, the mutual inductances between L_a and L_b are represented as M_{ab}.

Equation (4.3) was used to obtain the mutual inductances of equation (4.2).

$$M_{ab} = k_{ab}\sqrt{L_aL_b} \tag{4.3}$$

From equation (4.2), it can be observed that the interferences of the data link in the power link were not considered, since from the analysis presented by Ghovanloo and Atluri [11], it is assumed to be minimal. Also, the interference path produced by the coupling coefficients cascade of $k_{13}k_{34}$ was not considered either since in the analysis of figure-8 MILs presented by Jow and Ghovanloo [5] is considered negligible.

The Verilog-AMS modules that were created to implement the figure-8 MIL are shown in Appendix A. Module A.1 describes the interaction between the four inductors of the MIL by using equation (4.2).

Also, a higher level module, which instantiates Module A.1, was created to add the LC tank capacitances and the MIL input and output impedances. This higher level module is listed in Module A.2.

4.2 Bi-Level Pulse Harmonic Modulation

The bi-level PHM pulse pattern generator (PPG) block shown in the system of Fig. 4.1, was implemented based on the architecture presented by Inanlou and Ghovanloo [15], and this architecture is shown in Fig. 4.2.

As it can be observed from Fig. 4.2, the pulses width is controlled by the delay t_{pw}. Likewise, the delay between the initiation and suppression pulses is controlled by t_d.

It is stated by Atluri and Ghovanloo [26], that optimal data signal frequencies for proper CI operations should range in between $25 \backsim 50MHz$. Therefore, the frequency of the signal to be generated in the receiver data coil by the PPG impulse responses (f_r) was selected to be $37.5MHz$. Hence, as it can be observed from Table 4.1, the figure-8 data link was designed to resonate at approximately this frequency. Moreover, by using equation (2.1), it was determined that the t_{pw} needed to produce this data signal frequency, was $12\eta s$.

By substituting the figure-8 MIL data link parameters of Table 4.1 and f_r in the equations derived by Inanlou and Ghovanloo [15], it was theoretically determined that t_d needed to be $106\eta s$. Also, the normalized magnitude of the suppression pulse was determined to be

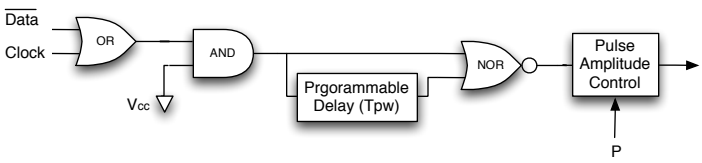

Fig. 4.2: PHM pulse pattern generator (PPG).

0.8, using the equation presented by Inanlou and Ghovanloo [15]. These PPG parameters allow the PHM system to work at a data rate of 5.2 Mbps.

The PHM receiver block shown in the system of Fig. 4.1 was designed based on the architecture presented by Inanlou and Ghovanloo [15], and this architecture is presented in Fig. 4.3.

The LNA of the PHM receiver was designed to have a gain of $26dB$, and the LPF was modeled to have a cutoff frequency of $9MHz$.

The peak voltage of the LPF output signal (LPF-Peak) when a bit 1 is transmitted was measured to be approximately $388mV$. Also, the LPF-Peak when a bit 0 is transmitted was measured to be $14mV$. Therefore, the comparator reference voltage was set to $187mV$, which is the mid-point between these peaks. Selecting the reference at this mid-point can minimize the BER of the system.

Moreover, the comparator was designed to compare the LPF output signal to the reference at the middle of the bit period (e.g., $100\eta s$ after the bit was transmitted). The reason for comparing at this time is beacause through experimental testing it was determined that at this time is when the LPF output voltage, when a bit 1 is transmitted, is the farthest to the LPF output voltage when a bit 0 is transmitted. Therefore, comparing at the middle of the bit period can minimize the BER of the system.

Even though it was theoretically determined that a t_d of $106\eta s$ would be appropriate for this bi-level system, it was experimentally observed that using a t_d of $111.7\eta s$ would produce less ISI in this system.

The behavior of the bi-level system when neither noise nor TX clock jitter was present, and when $t_d = 111.7\eta s$, is shown in Fig. 4.4. Likewise, the behavior of the system when $t_d = 106\eta s$ is presented in Fig. 4.5. If the data reciever coil signal of these two figures are compared, it can be observed that when $t_d = 111.7\eta s$ the system exhibits less ISI, and hence, a lower amount of data bits are received erroneously.

The code for the Verilog-AMS modules implemented for modelling this bi-level PHM system are presented in Appendix B. The bi-level pulse pattern generator code is shown in

Module B.1. The bi-level low noise amplifier code is presented in Module B.2. The bi-level rectifier code is shown in Module B.3. The bi-level low pass filter is presented in Module B.4, and finally the bi-level comparator is shown in Module B.5.

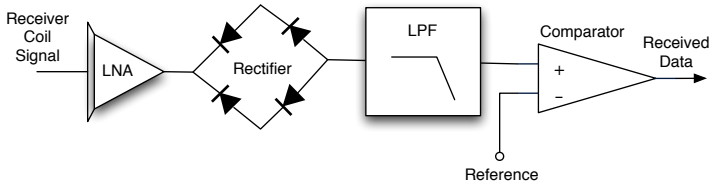

Fig. 4.3: PHM receiver.

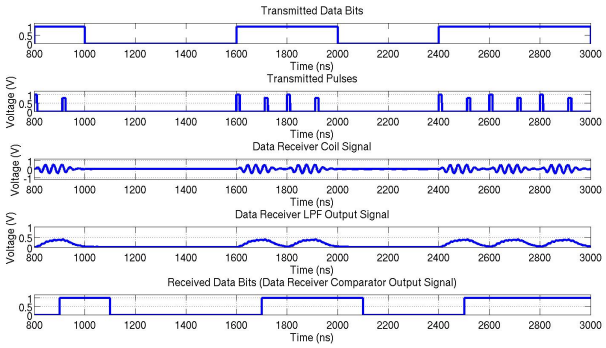

Fig. 4.4: Bi-level PHM system with $t_d = 111.7\eta s$.

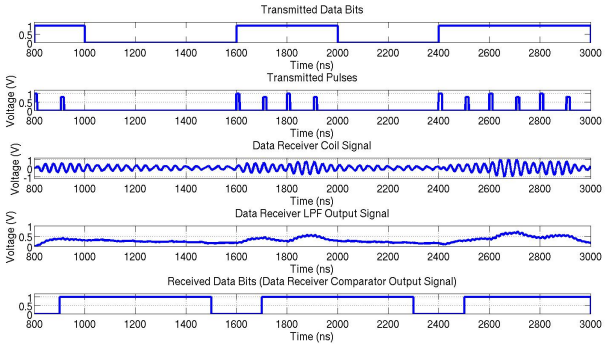

Fig. 4.5: Bi-level PHM system with $t_d = 106\eta s$.

4.3 Four-Level Pulse Harmonic Modulation

In order to model the four-level PHM system, the bi-level PPG model and the bi-level comparator were modified.

The four-level PPG model was created to output four different initiation pulses depending on the data transmitted. A truth table of the four-level PPG is shown in Table 4.2.

The suppression pulses generated by the four-level PPG were set to have a normalized amplitude of 0.8 with respect to the initiation pulse, similar to the bi-level PPG presented in Section 4.2.

The four-level comparator was modeled to make decisions based on three reference voltages. The decision making process of the four-level comparator is presented in Table 4.3.

It is suggested by Inanlou and Ghovanloo [15] that for a system using the figure-8 MIL parameters similar to the listed in Table 4.1, a t_d of $106\eta s$ would properly prevent ISI for any given initiation pulse (IP) amplitude, as long as the suppression pulse (SP) is normalized to 0.8 with respect to the IP. However, through experimental testings of the four-level PHM Verilog-AMS model, it was observed that the optimal t_d of a PHM system varies with respect to the IP amplitude. Therefore, the four-level PPG was modeled to vary t_d with respect to the IP transmitted, and hence, with respect to the data transmitted. Table 4.4 shows the t_d that the four-level PPG was modeled to use, for each IP transmitted.

Figure 4.6 presents the behavior of the four-level system implemented.

Figure 4.7 presents the four-level data receiver coil signal when t_d is fixed to either $109.9\eta s$, $108.26\eta s$, and $107.26\eta s$, and also when t_d varies according to Table 4.4. From this figure, it can be observed that although these t_ds may be optimal for a specific initiation pulse transmitted by the four-level PPG, they still produce large ISI when used with another of the pulses transmitted by the PPG. Therefore, it is assumed, that unless the PPG of an MLT-based PHM system is designed to vary between the optimal t_ds of the different initiation pulses it transmits, MLT-based PHM systems will exhibit large ISI, preventing

its feasibility.

The Verilog-AMS modules implemented for modelling this four-level PHM system are presented in Appendix C. The four-level PPG implementation is presented in Module C.1, and the four-level comparator implementation is shown in Module C.2.

Table 4.2: Four-level PPG truth table.

Data Transmitted	Initiation Pulse Amplitude (V)
00	0
01	2
10	4
11	6

Table 4.3: Four-level comparator decisions.

LPF Output Voltage (LPF-OV) Range	Four-Level Comparator Output
$LPF - OV <= 350mV$	00
$350mV < LPF - OV <= 1V$	01
$1V < LPF - OV <= 1.6V$	10
$LPF - OV > 1.6$	11

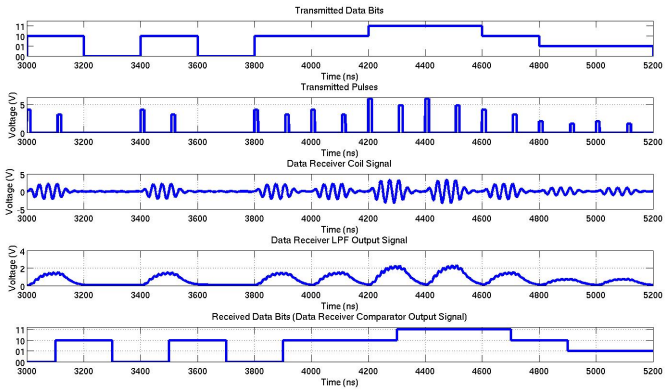

Fig. 4.6: Four-level PHM system behavior.

Table 4.4: Four-level PPG design parameters.

IP Amplitude	t_d
$0V$	N/A
$2V$	$109.9\eta s$
$4V$	$108.26\eta s$
$6V$	$107.8\eta s$

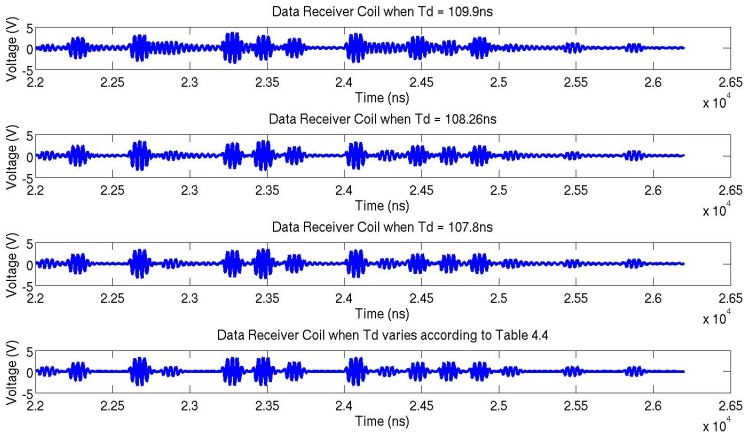

Fig. 4.7: Four-level PHM system performance with respect to t_d.

4.4 Bi-Level and Four-Level PHM Top-Level Module

Both, the bi-level and four-level PHM systems were modeled to transmit encoded data, which their PPG modules read from a file, as it can be observed in their respective PPG Verilog-AMS modules in Appendices B and C. Also, a Gallager A decoder was included in the receiver of these systems to decode the received data bits detected by their respective comparators. The codeword length used for the encoded data transmission was of 64 bits.

A Verilog-AMS top module was created to integrate the figure-8 MIL to the bi-level and four-level PHM systems, and also to perform their corresponding BER calculations. This module computes the BER of these systems for two cases: when the decoder is used to process the received data bits, and when the decoder is not used.

In order to compute the BER when the decoder is not used, the module compares each bit transmitted (e.g., the PPG output signal) to each bit received (e.g., the comparator output signal). In order to compute the BER when the decoder is used, the module latches the output of the comparator. Then, it inputs to the decoder the 64-bit latched codeword detected by the comparator, and then, waits for 20 clock cycles of decoder processing to compare the transmitted codeword with the output signal of the decoder.

A diagram of the top level module used for the bi-level and four-level systems is presented in Fig. 4.8.

The Verilog-AMS implementation of the top-level module used for the four-level and bi-level PHM systems are shown in Appendix D. The implementation of the top-level module used for modelling the bi-level PHM system is presented in Module D.1. Also, the implementation of the top-level module used for modelling the four-level PHM system is shown in Module D.2.

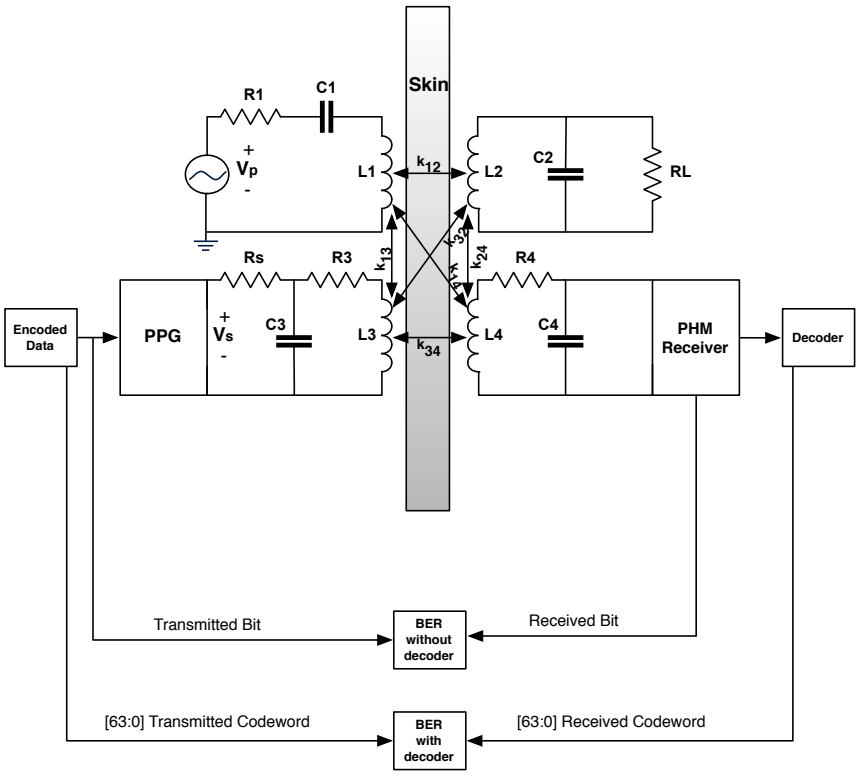

Fig. 4.8: Top-level module.

Chapter 5

Results

5.1 Bi-Level PHM Power Interference Characterization

In order to perform the bi-level power interference characterizations, resistor thermal noises were added to the systems, by using equation (5.1), and the LNA RMS noise, the power p-p amplitude, and the TX clock jitter were held constant at $2.34mV$, $100V$, and $0.3\eta s$, respectively.

$$v_n = \sqrt{4k_bTR\Delta f} \tag{5.1}$$

The main path for power interference in the data link of an MIL-based PHM system, such as the shown in Fig. 4.8, is the path between coils 1 and 4, which has the coupling coefficient of k_{14} [5]. One key factor that can cause k_{14} to increase in MILs are misalignments between the transmitter and receiver coils [5]. Figure 5.1 presents the effect of increments of k_{14} in the BER of the bi-level system and the performance of the Gallager A decoder in helping the system to whistand these increments. When the figure-8 MIL implemented in this thesis is in perfect alignmnent, its k_{14}, as calculated by Jow and Ghovanloo [5], is 1.2×10^{-4}, and this is the lowest k_{14} value evaluated in Fig. 5.1.

From Fig. 5.1, it can be observed that ECC can reduce the sensitivity of PHM systems to power interference, and hence, can increase robustness of PHM systems against factors that can affect MIL coupling coefficients, especially, misalignments.

From Fig. 5.2, it can be observed that as k_{14} increases, the modulation of the data receiver coil signal by the frequency of the power signal intensifies. Consequently, the modulation of the output signal of the receiver LPF also strengthens, as shown in Fig. 5.3. Therefore, it can be realized that the reason the BER of the system increases as the power

interference increases is because, the stronger the modulation of the LPF output signal, the higher are the chances that the LPF output signal will rise above the comparator reference voltage. Thus, increasing the chances that the comparator erroneously detect a bit 1 instead of a bit 0.

Due to this observed effect of the power interference in the receiver LPF output signal, and hence, in the BER of the system, it was then deduced that the comparator reference voltage can be raised to an optimal point, which will depend in the value of k_{14}. This optimal reference voltage would minimize the chances that the LPF modulated output signal cross the reference voltage when a bit 0 is transmitted.

Figure 5.4 presents how rising the comparator reference voltage can offset the effects of power interference by lowering the BER of the system when $k_{14} = 2.7 \times 10^{-3}$ and when $k_{14} = 3.0 \times 10^{-3}$, and enhancing decoder performance. This suggests that for every k_{14}, there is a reference voltage that can minimize the BER of the system and maximize decoder performance. Morever, it can be concluded, that the higher k_{14} the higher will be that optimal reference voltage.

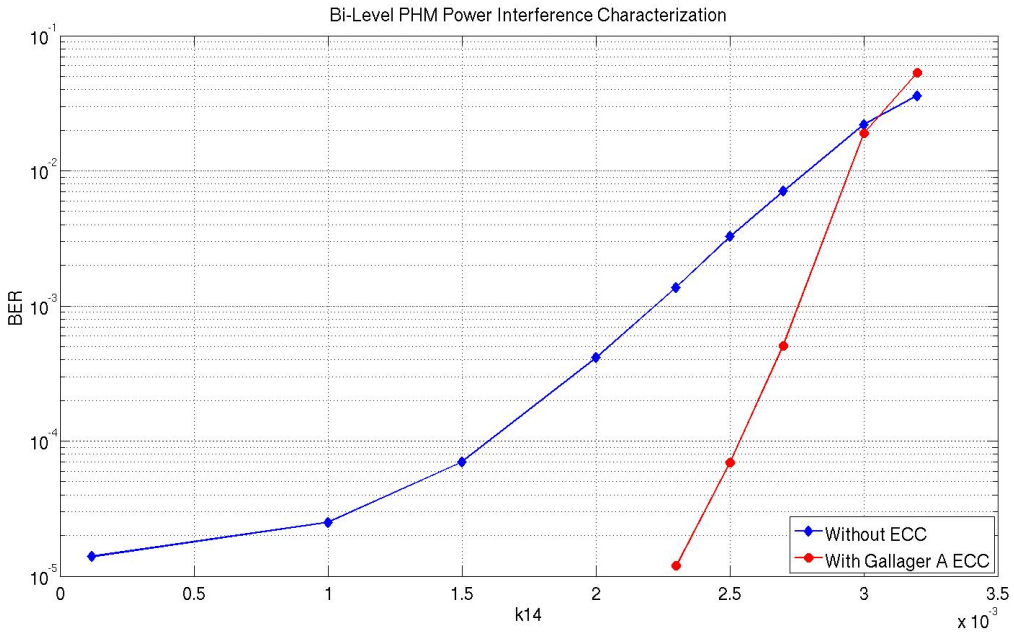

Fig. 5.1: Bi-level PHM power interference characterization.

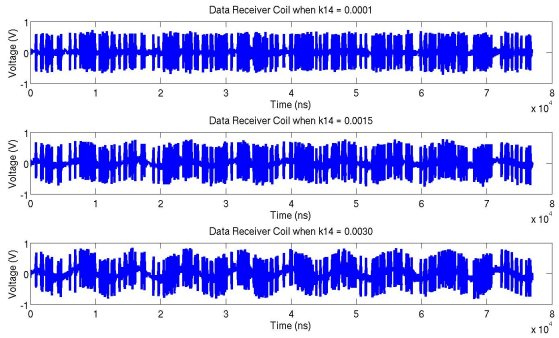

Fig. 5.2: Bi-level PHM data receiver coil modulation due to power interference.

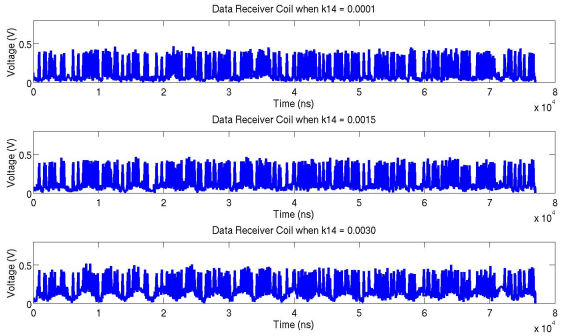

Fig. 5.3: Bi-level PHM receiver LPF output signal modulation due to power interference.

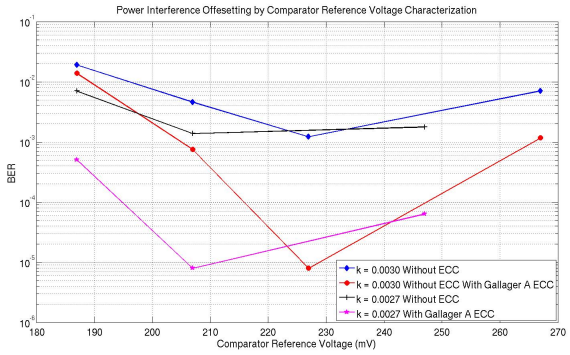

Fig. 5.4: Offsetting the rise in BER due to power interference by tuning the comparator reference voltage.

5.2 Bi-Level and Four-Level PHM Transmitter (TX) Clock Jitter Characterization

In order to perform the bi-level and four-level jitter characterizations, the LNA RMS noise, the power P-P amplitude, and k_{14} were held constant at $2.34mV$, $100V$, and 1.2×10^{-4}, respectively.

The TX clock jitter was modeled as a normally distributed value, which randomnly changes the TX clock duty cycle, and hence, randonmly varies t_d in each bit period, as it can be seen in the bi-level and four-level PPG Verilog-AMS modules presented in Appendices B and C, respectively.

Figure 5.5 presents the TX clock jitter characterizations of the bi-level and four-level systems.

It is noted from Fig. 5.5 that the four-level PHM systems is more sensitive to TX clock jitters than bi-level systems. This characteristic was expected, since this four-level system employs pulses that are higher in amplitude than the bi-level system pulses, which in turn generate greater ISI due to clock jitters.

It may be thought that if the four-level system employs pulses with lower amplitudes, its sensitivity to TX clock jitter could be decreased, since lower ISI would be generated due to clock jitter. However, lowering pulse amplitudes leads to lowering the difference between comparator reference voltages, which counteract the benefits of lower ISI. Therefore, it is not expected that lowering pulse amplitudes in four-level would significantly alter its sensitivity to jitter, however, tests should be performed to confirm this hypothesis. Based on this hypothesis and the results from Fig. 5.5, it is concluded that increasing the levels of a PHM system, also increases its sensitivity to TX clock jitter.

It can also be observed from Fig. 5.5 that both, bi-level and four-level PHM systems, would be robust against TX clock jitter below $200\rho s$. Recently, there have been reports of TX designs exhibiting clock jitters as low as $1.8\rho s$ and $1.9\rho s$, as presented by Sim et al. [27], and Demirkan and Spencer [28], respectively. Therefore, it can be deduced that both, bi-level and four-level PHM systems may tolerate clock jitter, especially when using

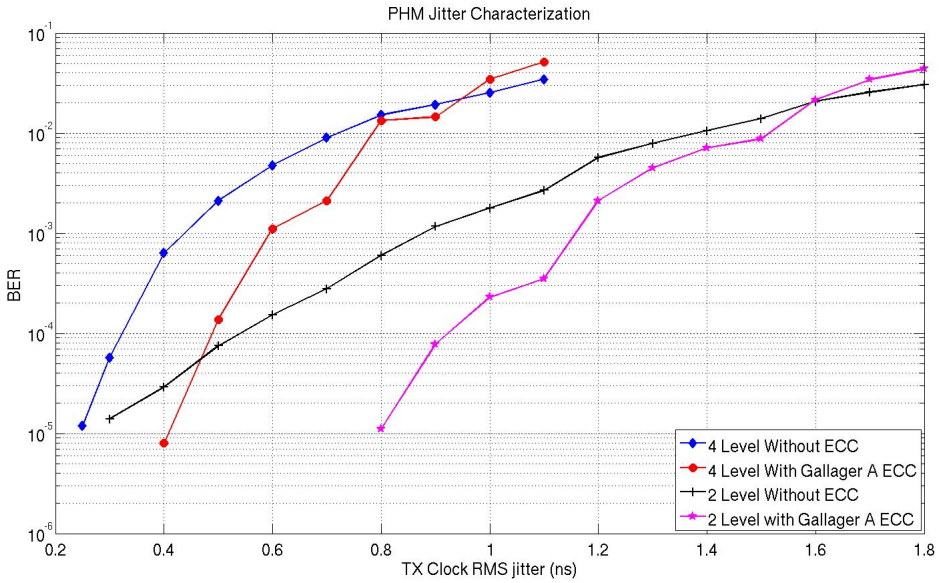

Fig. 5.5: PHM TX clock jitter characterization.

current state of the art TXs. Moreover, it can be assumed that multi-level PHM systems could be designed to have more than four levels of transmission, while still beging able to tolerate TX clock jitters.

5.3 Bi-Level and Four-Level PHM Noise Characterization

In order to perform the noise characterization of the bi-level and four-level systems, the power p-p amplitude, TX clock jitter, and k_{14} were set to $100V$, $0.3\eta s$, and 1.2×10^{-4}, respectively.

As it can be observed from Fig. 5.6, ECC can reduce the sensitivity to noise in PHM systems. Also, in Fig. 5.6, it can be realized that the four-level system is less sensitive to noise than the bi-level system.

This noise characterization was expected for the particular systems implemented in this thesis, because the difference in amplitude of the initiation pulses transmitted by the four-level PPG is $2V$, whereas, is only $1V$ for the pulses transmitted by the bi-level PPG.

This greater initiation pulse amplitude difference allows the four-level receiver comparator to be set with reference voltages that are more separated, hence, better preventing

the noise to cross the reference voltages and producing bit errors.

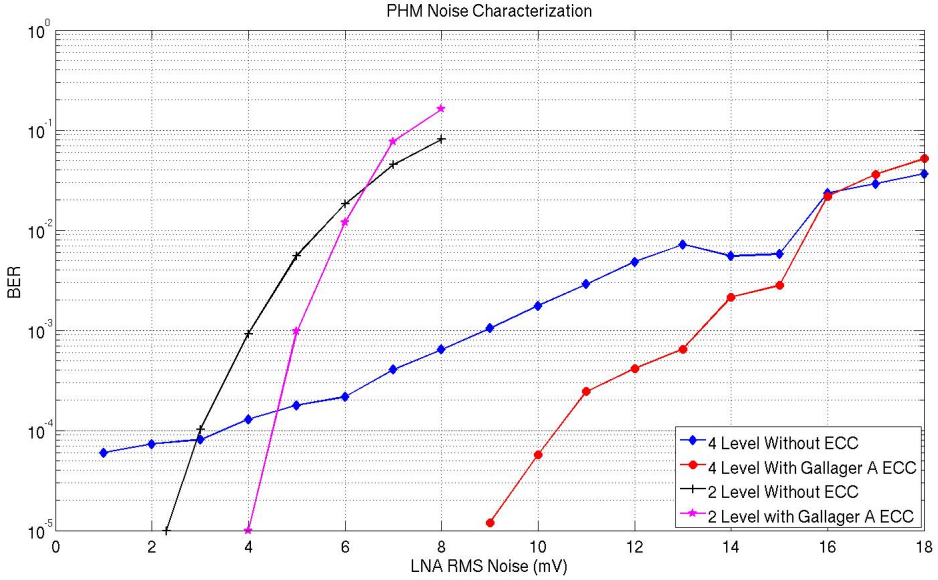

Fig. 5.6: PHM noise characterization.

Chapter 6

Conclusion and Future Work

This thesis has introduced the use of MS RTL verification tools, such as Verilog AMS and Cadence AMS Designers, in the validation of CI wireless systems. From the work presented in this thesis, it can be appreciated how useful can be MS RTL verification tools for validating a complex MS architecture such as an ECC-based wireless communication system, and for rapidly observing the effects of MS systems augmentations.

After evaluating the effectiveness of these tools through the system characterizations obtained from the models and experiments presented in this thesis, it is believed that MS RTL tools will be very useful for the CI research community in order to save time in reaching an optimal ECC based wireless solution. The reason being that MS architectures, especially ECC designs, can be rather complex, and RTL tools can speed up their functional verification and reduce IC functional failures. In the future, it is expected that some or all RTL models presented in this thesis will be used in conjunction with transistor level designs of ECC-based CI wireless systems, in order to follow the MS-IC design methodology proposed by Chang and Kundert [9], and for efficiently fabricating optimal CI wireless solution.

An RTL design of the integration of a figure-8 based MIL with an ECC-based bi-level PHM system has been introduced and validated in this thesis. The power interference in the received data signal of this system was characterized by observing the effect of increasing the main power interference path with coupling coefficient k_{14}, in the received data signal. It was observed that increments in k_{14} which are normally caused by MIL misalignments, result in increments in BER. In addition, it was determined that the reason for this rise in BER, is because as the power interference increases, the modulation of the receiver LPF output signal by the power signal carrier strengthens, which increases the chances of the

LPF output signal to cross the reference voltage when a bit 0 is transmitted. This crossing of the reference voltage by the LPF output signal then makes the receiver comparator erroneouly detect a bit 1, producing bit error.

It was then concluded that there is a level to which the comparator reference voltage can be raised, that minimizes the reference voltage crossings of the LPF output signal. Therefore, raising the reference to this level can minimize the BER and can further enhance the performance of ECC. It was determined that the higher k_{14} the higher needs to be this optimal reference voltage. Consequently, it is believed that a control system that would adapt the receiver comparator reference voltage to changes in k_{14}, could enhance robustness of PHM systems to factors that affect k_{14}, such as misalignments.

An RTL design of the integration of a figure-8 MIL with an ECC-based four-level PHM system has also been introduced and verified, as part of this thesis. It was shown that this system is able to double the data rate of the bi-level PHM system proposed by Inanlou and Ghovanloo [15]. This MLT-based technique to increase PHM data rate has the potential of being less power consuming than the pulse shaping technique used by Inanlou et al. [24], since it does not rely in increasing the frequency of the harmonics generated in the receiver. While testing the RTL design of this four-level system, it was also observed that, opposed to what is suggested by Inanlou and Ghovanloo [15], the optimal t_d of a PHM system varies with respect to initiation pulse amplitude. Morever, it was concluded that an MLT-based PHM system cannot be realizable, unless the PPG is designed to vary t_d depending on the initiation pulse transmitted.

TX clock jitter characterizations were performed on the bi-level and four-level systems. It was concluded that as levels of transmission increase, PHM systems become more vulnerable to jitter. It was also observed that both, bi-level and four-level PHM systems, would be robust against TX clock jitter below $200\rho s$. Recently, there have been reports of TX designs exhibiting clock jitters as low as $1.8\rho s$ and $1.9\rho s$, as presented by Sim et al. [27], and Demirkan and Spencer [28], respectively. Therefore, it was deduced that both, bi-level and four-level PHM systems may tolerate clock jitter, especially when using current state of the

art TXs. Moreover, it can be assumed that multi-level PHM systems could be designed to have more than four levels of transmission, while still being able to tolerate TX clock jitters. Future steps should include identifying the maximum level of transmission that PHM could employ while still tolerating clock jitter.

Finally, an RTL design of Gallager A ECC was implemented and its performance when applied to the bi-level and four-level MIL-based PHM systems was evaluated. This decoder is more powerful than the one used for the CI communication system presented by Arabi and Sawan [19], since it can detect multiple errors per frame. It was demonstrated that it can provide robustness to PHM systems against power interference, TX clock jitter, and SNR. Next steps will include estimating and optimizing the power consumption of this decoder.

References

[1] R. Brookmeyer, E. Johnson, K. Ziegler-Graham, and H. Arrighi, "Forecasting the global burden of alzheimer's disease," *Alzheimer's and Dementia 3*, vol. 3, p. 18691, 2007.

[2] Center for Disease Control and Prevention. Traumatic brain injury. [Online]. Available: http://www.cdc.gov/TraumaticBrainInjury/index.html, 2006.

[3] National Institute on Deafness and Other Communication Disorders, National Institutes of Health. Communication disorders. [Online]. Available: http://www.nidcd.nih.gov/health/statistics/vsl.asp, 2009.

[4] National Center for Health Statistics. Cerebrovascular disease or stroke. [Online]. Available: http://www.cdc.gov/nchs/FASTATS/stroke.htm, 2009 .

[5] U. Jow and M. Ghovanloo, "Optimization of data coils in a multiband wireless link for neuroprosthetic implantable devices," *Biomedical Circuits and Systems, IEEE Transactions*, vol. 4, no. 5, pp. 301–310, Oct. 2010.

[6] D. C. Galbraith, M. Soma, and R. L. White, "A wide-band efficient inductive transdennal power and data link with coupling insensitive gain," *Biomedical Engineering, IEEE Transactions*, vol. BME-34, no. 4, pp. 265–275, Apr. 1987.

[7] Y. Wang, S. Joeres, R. Wunderlich, and S. Heinen, "Modeling approaches for functional verification of rf-socs: Limits and future requirements," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions*, vol. 28, no. 5, pp. 769–773, May 2009.

[8] M. Hujer, R. Manasek, J. O'Mahony, P. Feerick, M. Barry, and B. Walsh, "Nanometer wireless transceiver modeling using verilog-ams and systemc," *Behavioral Modeling and Simulation Workshop, Proceedings of the 2006 IEEE International*, pp. 150–155, Sept. 2006.

[9] H. Chang and K. Kundert, "Verification of complex analog and rf ic designs," *Proceedings of the IEEE*, vol. 95, no. 3, pp. 622–639, Mar. 2007.

[10] G. Simard, M. Sawan, and D. Massicotte, "High-speed OQPSK and efficient power transfer through inductive link for biomedical implants," *Biomedical Circuits and Systems, IEEE Transactions*, vol. 4, no. 3, pp. 192-200, June 2010.

[11] M. Ghovanloo and S. Atluri, "A wide-band power-efficient inductive wireless link for implantable microelectronic devices using multiple carriers," *Circuits and Systems I: Regular Papers, IEEE Transactions*, vol. 54, no. 10, pp. 2211–2221, Oct. 2007.

[12] A. P. Chandrakasan, F. S. Lee, D. D. Wentzloff, V. Sze, B. P. Ginsburg, P. P. Mercier, D. C. Daly, and R. Blazquez, "Low-power impulse uwb architectures and circuits," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 332–352, Feb. 2009.

[13] N. Miura, D. Mizoguchi, M. Inoue, T. Sakurai, and T. Kuroda, "A 195-gb/s 1.2-w inductive inter-chip wireless superconnect with transmit power control scheme for 3-d-stacked system in a package," *Solid-State Circuits, IEEE Journal*, vol. 41, no. 1, pp. 23–34, Jan. 2006.

[14] J. C. Lin, C. C. Johnson, and A. W. Guy, "Power deposition in a spherical model of man exposed to 1-20 mhz em fields," *Microwave Symposium, 1973 IEEE G-MTT International*, pp. 257–259, June 1973.

[15] F. Inanlou and M. Ghovanloo, "Wideband near-field data transmission using pulse harmonic modulation," *Circuits and Systems I: Regular Papers, IEEE Transactions*, vol. 58, no. 1, pp. 186–195, Jan. 2011.

[16] G. J. Suaning and N. H. Lovell, "Cmos neurostimulation asic with 100 channels, scaleable output, and bidirectional radio-frequency telemetry," *Biomedical Engineering, IEEE Transactions*, vol. 48, no. 2, pp. 248–260, Feb 2001.

[17] D. Guermandi, S. Gambini, and J. Rabaey, "A 1 v 250 kpps 90 nm cmos pulse based transceiver for cm-range wireless communication," *Solid State Circuits Conference. 33rd European*, pp. 135–138, Sept. 2007.

[18] N. Sadeghi, S. Howard, S. Kasnavi, K. Iniewski, V. C. Gaudet, and C. Schlegel, "Analysis of error control code use in ultra-low-power wireless sensor networks," pp. 3558–3561, May 2006.

[19] K. Arabi and M. Sawan, "A secure communication protocol for externally controlled implantable devices," *Engineering in Medicine and Biology Society, IEEE 17th Annual Conference*, vol. 2, pp. 1661–1662, Sept. 1995.

[20] B. Tomatsopoulos and A. Demosthenous, "A cmos hard-decision analog convolutional decoder employing the mfda for low-power applications," *Circuits and Systems I: Regular Papers, IEEE Transactions*, vol. 55, no. 9, pp. 2912–2923, Oct. 2008.

[21] C. Winstead, J. Dai, S. Yu, C. Myers, R. R. Harrison, and C. Schlegel, "Cmos analog map decoder for (8,4) hamming code," *Solid-State Circuits, IEEE Journal*, vol. 39, no. 1, pp. 122–131, Jan. 2004.

[22] C. Winstead, N. Nguyen, V. C. Gaudet, and C. Schlegel, "Low-voltage cmos circuits for analog iterative decoders," *Circuits and Systems I: Regular Papers, IEEE Transactions*, vol. 53, no. 4, pp. 829–841, Apr. 2006.

[23] W. Xu, Z. Luo, and S. Sonkusale, "Fully digital bpsk demodulator and multilevel lsk back telemetry for biomedical implant transceivers," *Circuits and Systems II: Express Briefs, IEEE Transactions*, vol. 56, no. 9, pp. 714–718, Sept. 2009.

[24] F. Inanlou, M. Kiani, and M. Ghovanloo, "A 10.2 mbps pulse harmonic modulation based transceiver for implantable medical devices," *Solid-State Circuits, IEEE Journal*, vol. 46, no. 6, pp. 1296–1306, June 2011.

[25] K. Kundert and O. Zinke, *The Designer's Guide to Verilog AMS*. New York: Springer, 1971.

[26] S. Atluri and M. Ghovanloo, "Design of a wideband power-efficient inductive wireless link for implantable biomedical devices using multiple carriers," *Neural Engineering. Conference Proceedings. 2nd International IEEE Engineering in Medicine and Biology Society*, pp. 533–537, Mar. 2005.

[27] S. Sim, D. Kim, and S. Hong, "A cmos uwb pulse generator for 6-10 ghz applications," *Microwave and Wireless Components Letters, IEEE*, vol. 19, no. 2, pp. 83–85, Feb. 2009.

[28] M. Demirkan and R. R. Spencer, "A pulse-based ultra-wideband transmitter in 90-nm cmos for wpans," *Solid-State Circuits, IEEE Journal*, vol. 43, no. 12, pp. 2820–2828, Dec. 2008.

Appendices

Appendix A

Multi-Band Inductive Link

Two Verilog AMS modules were created in order to implement the figure-8 MIL.

The first module describes the interaction between the four inductors of the MIL by using equation (4.2), and it is presented in Module A.1.

Module A.1: Figure-8 MIL coils interaction.

```
1  ////////////////////////////////////////////////
2  // Figure-8 MIL Coils Interaction
3  ////////////////////////////////////////////////
4  // By David Toribio and Chris Winstead
5  // Dept of Electrical and Computer Engineering
6  // Utah State University
7  // June, 2011
8  ////////////////////////////////////////////////
9  `include "disciplines.vams"
10 module coupled_inductor (P1, N1, P2, N2, P3, N3, P4, N4);
11   parameter real L1   = 59e-6;
12   parameter real L2   = 6.55e-6;
13   parameter real L3   = 105e-9;
14   parameter real L4   = 281e-9;
15   parameter real k12  = 0.16688;
16   parameter real k14  = 0.00012;
17   parameter real k1224  = 0.0004;
18   parameter real k34  = 0.011;
19
20   real M12 = k12*sqrt(L1)*sqrt(L2);
21   real M14 = k14*sqrt(L1)*sqrt(L4);
22   real M1224 = k1224*sqrt(L2)*sqrt(L4);
23   real M34 = k34*sqrt(L3)*sqrt(L4);
24
```

```
25    inout P1, N1, P2, N2, P3, N3, P4, N4;
26    electrical P1, N1, P2, N2, P3, N3, P4, N4;
27
28    analog begin
29
30     V(P1,N1)  <+ L1*ddt(I(P1,N1)) + M12*ddt(I(P2,N2));
31     V(P2,N2)  <+ L2*ddt(I(P2,N2)) + M12*ddt(I(P1,N1));
32     V(P3,N3)  <+ L3*ddt(I(P3,N3)) + M34*ddt(I(P4,N4));
33     V(P4,N4)  <+ L4*ddt(I(P4,N4)) + M14*ddt(I(P1,N1)) + M1224*ddt(I(P1,N1)) +
           M34*ddt(I(P3,N3));
34
35    end
36  endmodule
```

Then, a higher level module, which instantiates Module A.1 was created to add the LC tank capacitances and the MIL input and output impedances. This higher level module is listed in Module A.2.

Module A.2: Figure-8 MIL top level.

```
1  ////////////////////////////////////////////////
2  // Figure-8 MIL Top Level
3  ////////////////////////////////////////////////
4  // By David Toribio and Chris Winstead
5  // Dept of Electrical and Computer Engineering
6  // Utah State University
7  // June, 2011
8  ////////////////////////////////////////////////
9  `include "disciplines.vams"
10 `include "constants.vams"
11
12 module coupled_inductor_top(L3a,L4b,gnd);
13  parameter real Tp = 8e-6;    //125 kHz
14  parameter real Td = 3.03e-8;  //33 MHz
15  parameter real R1 = 412e-3;
16  parameter real R2 = 1300e-3;
```

```
17  parameter real C1 = 225p;
18  parameter real C2 = 82.7p;
19
20  inout L3a, L4b, gnd;
21  electrical L1a,gnd,L1b,L1c,L2a,L3a,L3b,L4a,L4b;
22  ground gnd;
23
24  //Power Link Circuitry
25  vsource   #(.type("sine"), .freq(1/Tp), .ampl(50)) vpower(L1a, gnd);
26  nres   #(.r(50))          res1(L1a,L1b);
27  capacitor #(.c(27.48e-9))    cap1(L1b,L1c);
28  capacitor #(.c(247.50e-9))   res2(L2a,gnd);
29  nres   #(.r(50))          cap2(L2a,gnd);
30
31 //Data Link Circuitry
32  nres   #(.r(R1))          res3(L3a,L3b);
33  capacitor #(.c(C1))         cap3(L3a,gnd);
34  nres   #(.r(R2))          res4(L4a,L4b);
35  capacitor #(.c(C2))           cap4(L4b,gnd);
36
37 //Figure-8 MIL integration
38 coupled_inductor link(L1c, gnd, L2a, gnd, L3b, gnd, L4a, gnd);
39 endmodule
```

Appendix B

Bi-Level PHM

Module B.1: Bi-level pulse pattern generator.

```
1  ////////////////////////////////////////////////
2  // PHM Generator
3  ////////////////////////////////////////////////
4  // By David Toribio and Chris Winstead
5  // Dept of Electrical and Computer Engineering
6  // Utah State University
7  // June, 2011
8  ////////////////////////////////////////////////
9  `timescale 1ns/1ps
10 `include "disciplines.vams"
11 module phm_generator (outP,outN,data,tx_register, frame_num, jittrms);
12   output data;
13   output tx_register;
14   parameter real tr = 1p;
15   parameter real tf = 1p;
16   parameter real start = 0.0;
17   parameter real period = 1.0;
18   integer x;
19   parameter N=64;                    // Codeword size
20   parameter M=1;              // Number of bits per sample.
21   reg [N-1:0] coded_bits;            // Output from the encoder
22   reg [N-1:0] reversed_bits;
23   reg [N-1:0] tx_register;
24   reg more_data;
25   reg data, clk, p2, p4, state;
26   wire p1, p3, p5,p6;
27
```

```
28  inout outP,outN;
29  input frame_num;
30  input jittrms;
31
32
33  wreal jittrms;
34  wreal frame_num;
35  electrical outP,outN;
36  integer tx_index;
37  integer frame_ct;
38  integer seed;
39  integer file;
40  integer status;
41  integer _seed;
42
43  real jitt;
44  real jitter;
45  real jitbal;
46  real td_cal;
47
48  initial begin
49  // Open the file and read the data:
50  file = $fopen("randomseed","r");
51  status = $fscanf(file, "%d",seed);
52  $fclose(file);
53  // Report the value read:
54  $display("Seed obtained: %d\n",seed);
55    _seed = seed;
56    jitt = 0;
57    jitter = 0;
58    jitbal = 0;
59    td_cal = 5.7;
60    data = 0;
61    clk = 0;
62    state = 0;
```

```
63      p2 = 1;
64      p4 = 1;
65      x  = 1;
66      tx_index = -1;
67      frame_ct = 0;
68      $loadStrobeFileB(coded_bits, "./data.enc",more_data,1);
69      $strobeStimB(coded_bits);
70      #1 //to allow for reverse_bits to be able to detect clean data from
            coded_bits
71      reverse_bits(coded_bits,reversed_bits);
72    end
73
74    always begin
75  //    data = {$random} % 2;
76      jitt = $dist_normal(_seed,0,jittrms*1000)*0.001;
77      jitter  = 106+jitt+td_cal;
78      jitbal = 94-jitt-td_cal;
79      clk  =#jitter ~clk;
80      clk  =#jitbal ~clk;
81      tx_index = tx_index + 1;
82      if (tx_index > N-1)
83       begin
84          tx_register = reversed_bits;
85          tx_index = 0;
86          $strobeStimB(coded_bits);
87          frame_ct = frame_ct + 1;
88          reverse_bits(coded_bits,reversed_bits);
89                  if (frame_ct == frame_num) // sending an additional frame, so
                        the last frame output of the decoder gets processed
90                      $finish;
91       end
92      data = reversed_bits[tx_index];
93    end
94
95    not(p6,data);
```

```
96    or(p1, p6, clk);
97    and(p3, p1, p2);
98    xor(p5, p3, p4);
99
100   always @(p3) begin
101     #12 p4 = p3;
102   end
103
104   always @(posedge p5) begin
105     state = ~state;
106   end
107
108   analog begin
109     if (state)
110         V(outP,outN) <+ transition(p5,0,tr,tf)*1;
111     else
112         V(outP,outN) <+ transition(p5,0,tr,tf)*0.8;
113   end
114
115    task reverse_bits;
116       input [(M*N-1):0] forward;
117       output [(M*N-1):0] reverse;
118
119       integer            k;
120       for (k=0; k<(M*N); k=k+1)
121         reverse[(N*M-1-k)] = forward[k];
122    endtask // reverse_bits
123
124 endmodule
```

Module B.2: Low noise amplifier.

```
1  ////////////////////////////////////////////////
2  // Low Noise Amplifier
3  ////////////////////////////////////////////////
4  // By David Toribio
```

```
5  // Dept of Electrical and Computer Engineering
6  // Utah State University
7  // September, 2011
8  ////////////////////////////////////////////////
9  `timescale 1ns/1ps
10 `include "disciplines.vams"
11 `include "constants.vams"
12
13 module lna(p1,p2,gnd,siglna);
14  parameter real k = 19.95;
15  inout p1,p2,gnd;
16  input siglna;
17  wreal siglna;
18  electrical p1,p2,gnd;
19  integer seed;
20  integer file;
21  integer status;
22  integer _seed;
23  real     nval;
24
25 initial begin
26  // Open the file and read the data:
27    file = $fopen("randomseed","r");
28    status = $fscanf(file, "%d",seed);
29    $fclose(file);
30    // Report the value read:
31    $display("Seed obtained: %d\n",seed);
32 end
33
34
35  analog begin
36       @(initial_step) begin
37           _seed = seed+1;
38       end
39         V(p2,gnd) <+ k*(V(p1,gnd) + $rdist_normal(_seed,0,siglna));
```

```
40   end
41  endmodule
```

Module B.3: Rectifier.

```
1  ////////////////////////////////////////////////
2  // Rectifier
3  ////////////////////////////////////////////////
4  // By David Toribio
5  // Dept of Electrical and Computer Engineering
6  // Utah State University
7  // September, 2011
8  ////////////////////////////////////////////////
9  `timescale 1ns/1ps
10 `include "disciplines.vams"
11 `include "constants.vams"
12
13 module rectifier(p1,p2,gnd);
14
15  inout p1,p2;
16  electrical p1,p2,gnd;
17
18  analog begin
19  if (V(p1,gnd) > 0)
20          V(p2,gnd) <+ V(p1,gnd);
21  else
22          V(p2,gnd) <+ abs(V(p1,gnd));
23  end
24
25 endmodule
```

Module B.4: Low pass filter.

```
1  ////////////////////////////////////////////////
2  // Low Pass Filter
3  ////////////////////////////////////////////////
```

```
4  // By David Toribio
5  // Dept of Electrical and Computer Engineering
6  // Utah State University
7  // September, 2011
8  ////////////////////////////////////////////////
9  `timescale 1ns/1ps
10 `include "disciplines.vams"
11 `include "constants.vams"
12
13 module lpf(p1,p2,gnd);
14  parameter real r = 17.683e3;
15  parameter real c = 1e-12;
16  inout p1,p2;
17  electrical p1,p2,gnd;
18
19
20  nres #(.r(r)) r1(p1,p2);
21  capacitor #(.c(c)) c1(p2,gnd);
22
23 endmodule
```

Module B.5: Comparator.

```
1  ////////////////////////////////////////////////
2  // Comparator
3  ////////////////////////////////////////////////
4  // By David Toribio and Chris Winstead
5  // Dept of Electrical and Computer Engineering
6  // Utah State University
7  // June, 2011
8  ////////////////////////////////////////////////
9  `timescale 1ns/1ps
10 `include "disciplines.vams"
11 `include "constants.vams"
12
13 module comparator(p1,p2,gnd, rec_samples);
```

```
14  parameter N = 64;
15  parameter M = 1;
16  output [N-1:0] rec_samples;
17  output p2;
18  inout p1,gnd;
19  electrical p1,gnd;
20  reg p2;
21  ground gnd;
22  reg [N-1:0] rec_storage;
23  reg [N-1:0] rec_samples;
24  integer rx_index;
25
26 initial begin
27          p2 = 0;
28          rx_index = -1;
29        end
30
31 always begin
32
33          #100;
34
35          if(V(p1,gnd) > 0.187)
36             p2 = 1;
37          else
38             p2 = 0;
39
40          if(rx_index != -1)
41                  rec_storage[rx_index] = p2;
42          #100;
43
44          rx_index = rx_index + 1;
45          if (rx_index > N-1)
46           begin
47                  rx_index = 0;
48                  rec_samples = rec_storage;
```

```
49                end
50  end
51  endmodule
```

Appendix C

Four-Level PHM

Module C.1: Four-level pulse pattern generator.

```
1  ////////////////////////////////////////////////
2  // PHM Generator
3  ////////////////////////////////////////////////
4  // By Chris Winstead and David Toribio
5  // Dept of Electrical and Computer Engineering
6  // Utah State University
7  // June, 2011
8  ////////////////////////////////////////////////
9  `timescale 1ns/1ps
10 `include "disciplines.vams"
11
12
13 module phm_generator (outP,outN,data_multi,tx_register,jittrms,frame_num);
14
15 output data_multi;
16   output tx_register;
17   parameter real tr = 1p;
18   parameter real tf = 1p;
19   parameter real start = 0.0;
20   parameter real period = 1.0;
21   integer x;
22   parameter N=64;                    // Codeword size
23   parameter M=1;                 // Number of bits per sample.
24   reg [N-1:0] coded_bits;            // Output from the encoder
25   reg [N-1:0] reversed_bits;
26   reg [N-1:0] tx_register;
27   reg more_data;
```

```
28    reg clk, p2, p4, state, emit_pulse;
29    reg [1:0] data_multi;
30    wire p1, p3, p5,p6;
31
32    inout outP,outN;
33    input frame_num;
34    input jittrms;
35
36    wreal jittrms;
37    wreal frame_num;
38    electrical outP,outN;
39    integer tx_index_multi;
40    integer frame_ct;
41    integer seed;
42    integer file;
43    integer status;
44
45    real jitter;
46    real tdelay;
47    real periodend;
48    real td_calibration;
49    real pulse_amplitude;
50
51    initial begin
52   // Open the file and read the data:
53    file = $fopen("randomseed","r");
54    status = $fscanf(file, "%d",seed);
55    $fclose(file);
56    // Report the value read:
57    $display("Seed obtained: %d\n",seed);
58      jitter = 0;
59      tdelay = 0;
60      periodend = 0;
61      td_calibration = 0;
62      data_multi = 2'b00;
```

```
63      emit_pulse = 1;
64      clk = 0;
65      state = 0;
66      p2 = 1;
67      p4 = 1;
68      x  = 1;
69      tx_index_multi = -2;
70      frame_ct = 0;
71      $loadStrobeFileB(coded_bits, "./data.enc",more_data,1);
72      $strobeStimB(coded_bits);
73      #1 //to allow for reverse_bits to be able to detect clean data from
            coded_bits
74      reverse_bits(coded_bits,reversed_bits);
75    end
76
77    always begin
78      jitter = $dist_normal(seed,0,jittrms*1000)*0.001;
79      tdelay    = 106+td_calibration+jitter;
80      periodend = 94-td_calibration-jitter;
81      clk  = #tdelay ~clk;
82      clk  = #periodend ~clk;
83      tx_index_multi = tx_index_multi + 2;
84    if (tx_index_multi > N-1)
85       begin
86          tx_register = reversed_bits;
87          tx_index_multi = 0;
88  //        tx_index = 0;
89          $strobeStimB(coded_bits);
90          frame_ct = frame_ct + 1;
91          reverse_bits(coded_bits,reversed_bits);
92                  if (frame_ct == frame_num) // sending an additional frame, so
                        the last frame output of the decoder gets processed
93                      $finish;
94       end
95   //      data = reversed_bits[tx_index];
```

```
96      data_multi = reversed_bits[tx_index_multi +: 2];
97      case (data_multi)
98          2'b00 : pulse_amplitude = 0;
99          2'b01 : pulse_amplitude = 2;
100         2'b10 : pulse_amplitude = 4;
101         2'b11 : pulse_amplitude = 6;
102     endcase // case (data)
103     case (data_multi)
104         2'b00 : td_calibration  = 0;
105         2'b01 : td_calibration  = 3.900;
106         2'b10 : td_calibration  = 2.260;
107         2'b11 : td_calibration  = 1.800;
108     endcase // case (data)
109   end
110
111   not(p6,emit_pulse);
112   or(p1, p6, clk);
113   and(p3, p1, p2);
114   xor(p5, p3, p4);
115
116   always @(p3) begin
117     #12 p4 = p3;
118   end
119
120   always @(posedge p5) begin
121     state = ~state;
122   end
123
124   analog begin
125     if (state)
126       V(outP,outN) <+ transition(p5,0,tr,tf)*1*pulse_amplitude;
127     else
128       V(outP,outN) <+ transition(p5,0,tr,tf)*0.8*pulse_amplitude;
129   end
130
```

```
131     task reverse_bits;
132         input [(M*N-1):0] forward;
133         output [(M*N-1):0] reverse;
134 
135         integer             k;
136         for (k=0; k<(M*N); k=k+1)
137           reverse[(N*M-1-k)] = forward[k];
138     endtask // reverse_bits
139 
140 endmodule
```

Module C.2: Four-level comparator.

```
1  ////////////////////////////////////////////////
2  // Comparator
3  ////////////////////////////////////////////////
4  // By David Toribio and Chris Winstead
5  // Dept of Electrical and Computer Engineering
6  // Utah State University
7  // June, 2011
8  ////////////////////////////////////////////////
9  `timescale 1ns/1ps
10 `include "disciplines.vams"
11 `include "constants.vams"
12 
13 module comparator(p1,p2,gnd, rec_samples);
14  parameter N = 64;
15  parameter M = 1;
16  output [N-1:0] rec_samples;
17  output [1:0] p2;
18  inout p1,gnd;
19  electrical p1,gnd;
20  reg [1:0] p2;
21  ground gnd;
22  reg [N-1:0] rec_storage;
23  reg [N-1:0] rec_samples;
```

```
24  integer rx_index;
25
26 initial begin
27          p2 = 2'b00;
28          rx_index = -2;
29        end
30
31 always begin
32
33          #100;
34
35          if(V(p1,gnd) <= 0.35) p2 = 2'b00;
36          else if ((V(p1,gnd) > 0.35) && (V(p1,gnd) <= 1)) p2 = 2'b01;
37          else if ((V(p1,gnd) > 1) && (V(p1,gnd) <= 1.6)) p2 = 2'b10;
38          else if (V(p1,gnd) > 1.6) p2 = 2'b11;
39
40          if(rx_index != -2)
41                  rec_storage[rx_index +: 2] = p2;
42          #100;
43
44          rx_index = rx_index + 2;
45          if (rx_index > N-1)
46           begin
47                  rx_index = 0;
48                  rec_samples = rec_storage;
49           end
50 end
51 endmodule
```

Appendix D

Top-Level Modules

Module D.1: Top-level module used for the bi-level system.

```
1  ////////////////////////////////////////////////
2  // PHM Top-level
3  ////////////////////////////////////////////////
4  // By Chris Winstead & David Toribio
5  // Dept of Electrical and Computer Engineering
6  // Utah State University
7  // June, 2011
8  ////////////////////////////////////////////////
9  `timescale 1ns/1ps
10 `include "disciplines.vams"
11
12 module phm_top;
13
14   parameter M = 1;
15   parameter N = 64;
16   parameter iterations = 20; // Iterations per frame.
17   parameter framesent  = 400;
18
19   reg clk;
20   reg rst;
21   reg [N-1:0]     error_mask;          // Error locations
22
23   wire [N-1:0]  decoded_bits;       // Output from the decoder
24   wire [N-1:0]  tx_register;       // Output from the decoder
25   wire [N-1:0]  rec_samples;       // Output from the decoder
26   wire [1:0] data;
27   wire [1:0] recdata;
```

```
28
29
30  electrical pulse,gnd,p1a,p1b,recsig,ampsig,rectsig,filtsig;
31  ground gnd;
32
33  real sig;
34  real frame_num;
35  real jittrms;
36
37  wreal wr_frame_num;
38  wreal wr_sig;
39  wreal wr_jittrms;
40
41  assign wr_jittrms = jittrms;
42  assign wr_sig = sig;
43  assign wr_frame_num = frame_num;
44
45  integer        idx;   // for-loop index variable
46  integer        clock_count;  // Total clocks since start of simulation
47  integer        error_count;  // Total errors observed
48  integer        bit_count;    // Total bits observed
49  integer        count;
50  integer        ecount;
51  integer        sentbits_index;
52  integer        iter_flag;
53  integer        frame_count;
54  integer        file;
55  integer        file1;
56  integer        status;
57  integer        status1;
58  integer        jitter;
59  integer        snr;
60
61
62
```

```
63    initial begin
64  // Open the file and read the data:
65    file = $fopen("jitter","r");
66    status = $fscanf(file, "%d",jitter);
67    $fclose(file);
68    $display("Jitter obtained: %d\n",jitter);
69    file1 = $fopen("snr","r");
70    status1 = $fscanf(file1, "%d",snr);
71    $fclose(file1);
72    $display("snr obtained: %d\n",snr);
73
74    frame_num = framesent + 1;
75    jittrms = jitter*0.01; //in ns
76    sig = snr*0.00001;
77    ecount = 0;
78    count  = 0;
79    idx = 0;
80    clock_count = 0;
81    error_count = 0;
82    bit_count = 0;
83    frame_count = 0;
84    rst = 1'b0;
85    clk = 1;
86    sentbits_index = -2;
87    iter_flag = 0;
88    forever #100 clk = ~clk;
89    end
90
91    phm_generator #(.M(M),.N(N)) pg1 (pulse,gnd,data,tx_register,wr_jittrms,
          wr_frame_num);
92    nres #(.r(50)) Rs (pulse,p1a);
93    coupled_inductor_top cit1(p1a,recsig,gnd);
94    lna lna1(recsig, ampsig, gnd, wr_sig);
95    rectifier rec1(ampsig,rectsig,gnd);
96    lpf lpf1(rectsig,filtsig,gnd);
```

```
97    comparator #(.M(M),.N(N)) comp1(filtsig, recdata, gnd, rec_samples);
98    decoder #(.M(M)) D1 (.channel_in(rec_samples), .decisions(decoded_bits), .
          clk(clk), .rst(rst));
99
100   always @(posedge clk)
101       begin
102           sentbits_index = sentbits_index + 2;
103
104           if (sentbits_index > N-1 || iter_flag)
105            begin
106                   if (iter_flag == 0)
107                   begin
108                           sentbits_index = 0;
109                           frame_count = frame_count + 1;
110                   end
111
112                   if ((clock_count == 0) && (~rst))
113                           rst = 1'b1;
114                   if ((clock_count == 1) && (rst))
115                           rst = 1'b0;
116
117                   clock_count = clock_count+1;
118                   iter_flag = 1;
119
120                   if (clock_count > iterations+1)
121                    begin
122                           clock_count = 0;
123                           iter_flag = 0;
124                           error_mask = tx_register ^ decoded_bits;
125
126   //                      // Update total bit count:
127                           bit_count = bit_count + N;
128
129   //                      // If one or more errors happened, count them:
130                           if (| error_mask)
```

```
131                             begin
132                                     for (idx = 0; idx < N; idx = idx + 1)
133                                     if (error_mask[idx])
134                                     error_count = error_count + 1;
135                             end
136
137                             if (frame_count == 400)
138                             begin
139                             file = $fopen("testbench_output","w");
140                             $fwrite(file,"%d\n",error_count);
141                             $fwrite(file,"%d\n",ecount);
142                             $fwrite(file,"%d\n",bit_count);
143                             $fclose(file);
144                             end
145                     end
146             end
147     end
148
149 //************************************************
150 //       BER Calculation (Without Decoder Processing)
151 //************************************************
152 always begin
153  #150;
154         if(sentbits_index != -2)
155          begin
156
157                 if(frame_count < framesent)
158                  begin
159                         count = count + 2;
160                         if(data[0] != recdata[0])
161                         ecount = ecount + 1;
162                         if(data[1] != recdata[1])
163                         ecount = ecount + 1;
164                  end
165          end
```

```
166    #50;
167    end
168  endmodule
```

Module D.2: Top-level module used for the four-level system.

```
1  ////////////////////////////////////////////////
2  // PHM Top-level
3  ////////////////////////////////////////////////
4  // By David Toribio & Chris Winstead
5  // Dept of Electrical and Computer Engineering
6  // Utah State University
7  // June, 2011
8  ////////////////////////////////////////////////
9  `timescale 1ns/1ps
10 `include "disciplines.vams"
11
12 module phm_top;
13
14   parameter M = 1;
15   parameter N = 64;
16   parameter iterations = 20; // Iterations per frame.
17   parameter framesent  = 400;
18
19   reg clk;
20   reg rst;
21   reg [N-1:0]     error_mask;           // Error locations
22
23   wire [N-1:0]  decoded_bits;        // Output from the decoder
24   wire [N-1:0]  tx_register;        // Output from the decoder
25   wire [N-1:0]  rec_samples;        // Output from the decoder
26   wire [1:0] data;
27   wire [1:0] recdata;
28
29
30   electrical pulse,gnd,p1a,p1b,recsig,ampsig,rectsig,filtsig;
```

```
31    ground gnd;
32
33    real sig;
34    real frame_num;
35    real jittrms;
36
37    wreal wr_frame_num;
38    wreal wr_sig;
39    wreal wr_jittrms;
40
41    assign wr_jittrms = jittrms;
42    assign wr_sig = sig;
43    assign wr_frame_num = frame_num;
44
45    integer         idx;    // for-loop index variable
46    integer         clock_count;  // Total clocks since start of simulation
47    integer         error_count;  // Total errors observed
48    integer         bit_count;    // Total bits observed
49    integer         count;
50    integer         ecount;
51    integer         sentbits_index;
52    integer         iter_flag;
53    integer         frame_count;
54    integer         file;
55    integer         file1;
56    integer         status;
57    integer         status1;
58    integer         jitter;
59    integer         snr;
60
61
62
63    initial begin
64  // Open the file and read the data:
65    file = $fopen("jitter","r");
```

```
66    status = $fscanf(file , "%d",jitter);
67    $fclose(file);
68    $display("Jitter obtained: %d\n",jitter);
69    file1 = $fopen("snr","r");
70    status1 = $fscanf(file1 , "%d",snr);
71    $fclose(file1);
72    $display("snr obtained: %d\n",snr);
73
74    frame_num = framesent + 1;
75    jittrms = jitter*0.01; //in ns
76    sig = snr*0.00001;
77    ecount = 0;
78    count  = 0;
79    idx = 0;
80    clock_count = 0;
81    error_count = 0;
82    bit_count = 0;
83    frame_count = 0;
84    rst = 1'b0;
85    clk = 1;
86    sentbits_index = -2;
87    iter_flag = 0;
88    forever #100 clk = ~clk;
89    end
90
91    phm_generator #(.M(M),.N(N)) pg1 (pulse,gnd,data,tx_register,wr_jittrms,
          wr_frame_num);
92    nres #(.r(50)) Rs (pulse,p1a);
93    coupled_inductor_top cit1(p1a,recsig,gnd);
94    lna lna1(recsig, ampsig, gnd, wr_sig);
95    rectifier rec1(ampsig,rectsig,gnd);
96    lpf lpf1(rectsig,filtsig,gnd);
97    comparator #(.M(M),.N(N)) comp1(filtsig, recdata, gnd, rec_samples);
98    decoder #(.M(M)) D1 (.channel_in(rec_samples), .decisions(decoded_bits), .
          clk(clk), .rst(rst));
```

```
99
100 always @(posedge clk)
101     begin
102         sentbits_index = sentbits_index + 2;
103
104         if (sentbits_index > N-1 || iter_flag)
105          begin
106                 if (iter_flag == 0)
107                 begin
108                         sentbits_index = 0;
109                         frame_count = frame_count + 1;
110                 end
111
112                 if ((clock_count == 0) && (~rst))
113                         rst = 1'b1;
114                 if ((clock_count == 1) && (rst))
115                         rst = 1'b0;
116
117                 clock_count = clock_count+1;
118                 iter_flag = 1;
119
120                 if (clock_count > iterations+1)
121                  begin
122                         clock_count = 0;
123                         iter_flag = 0;
124                         error_mask = tx_register ^ decoded_bits;
125
126 //                      // Update total bit count:
127                         bit_count = bit_count + N;
128
129 //                      // If one or more errors happened, count them:
130                         if (| error_mask)
131                          begin
132                                 for (idx = 0; idx < N; idx = idx + 1)
133                                 if (error_mask[idx])
```

```
134                                     error_count = error_count + 1;
135                                 end
136
137                                 if (frame_count == 400)
138                                 begin
139                                 file = $fopen("testbench_output","w");
140                                 $fwrite(file,"%d\n",error_count);
141                                 $fwrite(file,"%d\n",ecount);
142                                 $fwrite(file,"%d\n",bit_count);
143                                 $fclose(file);
144                                 end
145                         end
146                 end
147             end
148
149     //************************************************
150     //        BER Calculation (Without Decoder Processing)
151     //************************************************
152     always begin
153      #150;
154             if(sentbits_index != -2)
155              begin
156                     if(frame_count < framesent)
157                      begin
158                             count = count + 2;
159                             if(data[0] != recdata[0])
160                             ecount = ecount + 1;
161                             if(data[1] != recdata[1])
162                             ecount = ecount + 1;
163                      end
164              end
165      #50;
166      end
167 endmodule
```