Rose-Hulman Institute of Technology
Rose-Hulman Scholar

Graduate Theses - Electrical and Computer Engineering

Electrical and Computer Engineering

5-9-2003

Implementing an Integrated Signaling and Power Distribution Control System for Remotely Located Devices

Andrew Stephen Hintz
Rose-Hulman Institute of Technology

Follow this and additional works at: https://scholar.rose-hulman.edu/dept_electrical

 Part of the Power and Energy Commons, and the Signal Processing Commons

Recommended Citation

Hintz, Andrew Stephen, "Implementing an Integrated Signaling and Power Distribution Control System for Remotely Located Devices" (2003). *Graduate Theses - Electrical and Computer Engineering*. 1.
https://scholar.rose-hulman.edu/dept_electrical/1

This Thesis is brought to you for free and open access by the Electrical and Computer Engineering at Rose-Hulman Scholar. It has been accepted for inclusion in Graduate Theses - Electrical and Computer Engineering by an authorized administrator of Rose-Hulman Scholar. For more information, please contact weir1@rose-hulman.edu.

IMPLEMENTING AN INTEGRATED SIGNALING AND POWER DISTRIBUTION CONTROL SYSTEM FOR REMOTELY LOCATED DEVICES

A Thesis

Submitted to the Faculty

of

Rose-Hulman Institute of Technology

by

Andrew Stephen Hintz

In Partial Fulfillment of the
Requirements for the Degree

of

Master of Science in Electrical Engineering

May 2003

Rose-Hulman Institute of Technology

FINAL EXAMINATION REPORT

Name Andrew Hintz Discipline(s) Electrical Engineering

X Thesis Thesis Topic: Implementing an Integrated Signaling and Power Distribution Control System for Remotely Located Devices

____ Non thesis Topic of Relevant Experience: ____

I. EXAMINATION COMMITTEE:

PROFESSOR	DEPT.
Chairperson: Dr. Tina Hudson	ECE
Dr. Ed Doering	ECE
Dr. J.P. Mellor	CS

Requested by: Tina Hudson Chairperson

Approved by: Jared Borrey Department Head

II. FINAL EXAMINATION REPORT:

Date of Exam: May 9, 2003

X Passed

____ Failed

Advisory Committee Signatures of Approval:

Tina Hudson Chairperson

E K Doering

J P Mellor

APPROVED R.H.I.T. GRADUATE COMMITTEE

When the report is complete, the Advisory Committee Chairperson will send this form to Graduate Studies. Copies will be sent to each committee member, the department head, and the student.

ABSTRACT

Hintz, Andrew Stephen

MSEE

Rose-Hulman Institute of Technology

May 2003

Implementing an Integrated Signaling and Power Distribution Control System for Remotely Located Devices

Dr. Tina A. Hudson

A system was designed and implemented that combined the distribution of high-current power with a digital control signal over a common conductor. Two different versions of this system were implemented. Initially, a design based on of commercially available parts was created and tested to prove that the concept of combining communications and power is valid. The resulting design was then miniaturized to show that the system might be combined onto a single integrated circuit. In the miniaturization process, some circuit blocks were redesigned to take advantage of the flexibility provided by ASIC designs. Both the proof of concept and the VLSI implementations were completely designed, implemented, and fully tested. It was shown that the system can be miniaturized. The

miniaturization provided the advantages of smaller overall implementation size and higher reliability due to decreased part count. The disadvantage of the miniaturization process was that the design became fixed once it was fabricated in silicon.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

The combination of power distribution and control signals is becoming more commonplace in various industries. As remote devices require less direct user intervention, centralized control of the devices becomes the primary method of control [10]. When power distribution and the transmission of control signals are combined, then remotely controlling these devices is made more economical [12].

There are many different ways that the combination of power distribution and communications have been implemented. The most prevalent implementation is called power line communications (PLC). Electricity distribution companies have utilized PLC for the reading of its customers' meters [10]. Within the industrial automation sector, a protocol named LONWORKS has been implemented that also operates over the power distribution lines [11]. The X10 standard is common among users of home automation systems [11]. All of these protocols are very application specific, and no generic standard has been implemented.

The trucking industry could benefit from the combination of power distribution and control data, which is applicable in various parts of a tractor-trailer, or semi. One element of the semi that would benefit from this combination would be the control of the trailer lighting. Currently, individual circuits control all the lights on a trailer through a

device called a 7-Way. Combining the distribution of power to these circuits with control data replaces five individual conductors of the 7-Way with a single conductor, creating a weight and material savings. The other 2, that are not directly replaced, are a line for the Anti-Lock Braking System and a chassis ground.

The 7-Way is currently the standard for controlling semi-trailer lights. The system consists of a bundle of cabling that runs from the tractor to the lights. As the name implies, there are seven conductors, six of which are used to control lights, and one is a ground line. Applying a voltage to its corresponding conductor in the 7-Way controls each circuit. Some of the lights that are controlled by the 7-Way include the brake lights, turn signal lights, and general trailer marker lights.

The 7-Way that is currently used to provide power for lighting on semi trailers has several drawbacks that occur in the case of a failure. Should one wire of the 7-Way be damaged, producing an open-circuit condition, the lights connected on that wire no longer illuminate, creating a safety hazard. To repair a single conductor that is damaged, the entire wiring harness must be removed, which requires a great deal of time. This process incurs a great expense to the trailer owner in terms of parts, labor, and loss of the use of the trailer.

The protocols that have been discussed so far all have their drawbacks when applied to semi-trailers. Both PLC and LONWORKS are based on an AC power distribution grid [11]. Because a semi is based upon a DC power distribution grid, it would be very costly to retrofit the trailer to run off AC power that is fed from a DC power grid. The X10 protocol, which could be modulated onto a DC power grid, requires one second to transmit some commands resulting in a protocol that is too slow for this application [11]. With all of these drawbacks, a new application specific protocol needed to be developed.

The system presented in this thesis solved both of these problems. With the architecture of this system, a single point of failure can exist only at a receiver or transmitter; thus, a single break in the conductor would not cause a failure of any of the lights. With the communications and power distribution available via a single conductor, this system would be smaller in size than a state-of-the-art harness and would be less expensive to replace and easier to retrofit.

1.1. State of the Art in the Trucking Industry

Where the protocols that were discussed in the previous section are currently implemented systems that combine power distribution and control data, this section discusses the protocols that are currently available in the trucking industry. Before discussing what systems are readily available to the trucking industry, a cursory

explanation of the desired system operation is needed. As is shown in Figure 1, the purpose of this system was to combine 12 V and a control signal to generate a "modulated power signal." This modulated power signal was then received, and converted back into control data, as well as a 5 V signal to power control logic. The modulated power signal that is received is also used to power the load. A single conductor could be used to transmit power and communications by combining them into a single signal, using a chassis ground for the return.

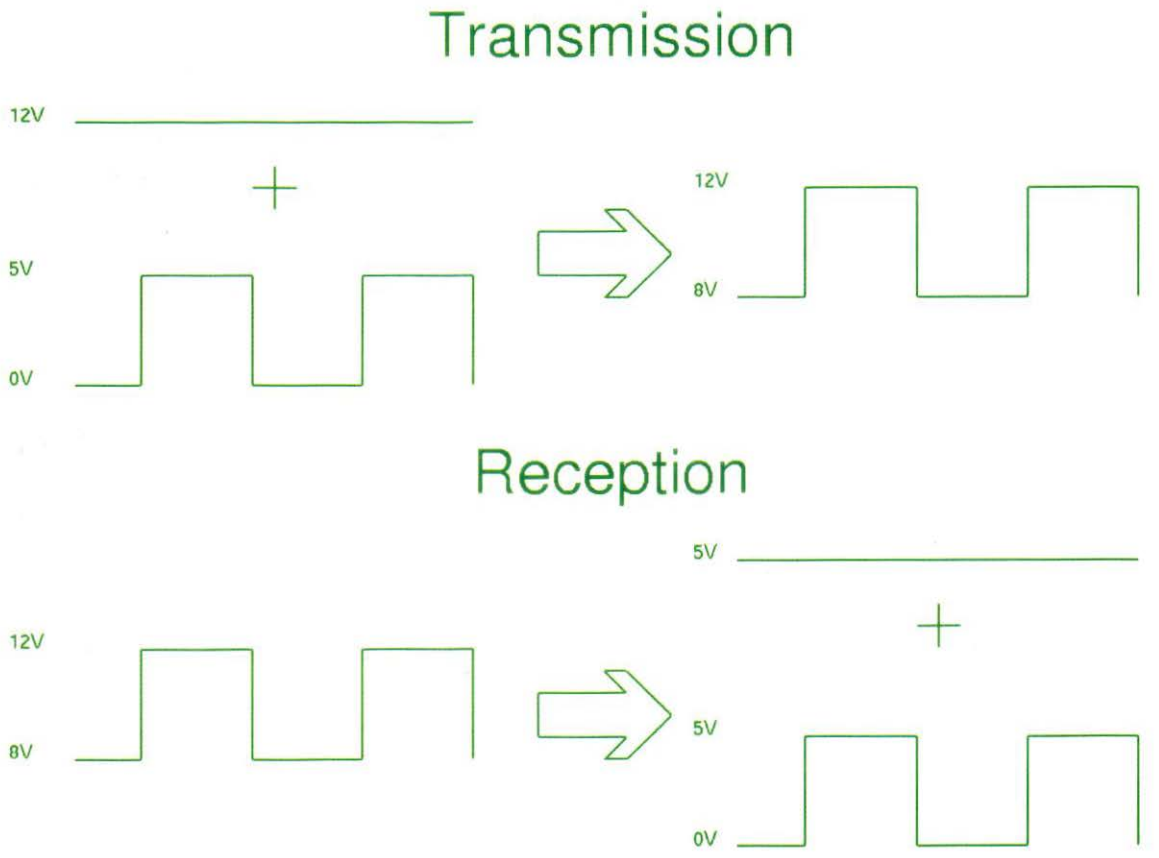

Figure 1. Conceptual Representation of the Control Signals

Currently, several protocols are in use that relate to the system presented in this thesis. SAE J1708 and Controller Area Networks are already used in the trucking industry with others emerging in limited use. Some of the standards presented here are solutions, which combine communications and power, but each has imitations.

The first protocol discussed is SAE J1708, "Serial Data Communications Between Microcomputer Systems in Heavy-Duty Vehicle Applications" [14]. This protocol is currently in use in the trucking industry, primarily moving data between sensors on the tractor and the on-board computer. Electrically, the protocol is based on the RS-485 standard. J1708 is a low bandwidth protocol running at 9600 baud, and is also a half-duplex system. The protocol implements message structure and bus access architecture, and allows higher priority devices to gain access to the bus first. J1708 is a communications-only protocol and, therefore, does not supply power to remote devices.

Controller Area Network (CAN) is another system that is becoming increasingly prevalent in the trucking industry. CAN is described for road vehicles in standard ISO 11898. CAN is designed to work among the engine control units of a vehicle. This standard provides for communication rates of 125 kilobits-per-second to 1 megabit-per-second. The system also has a standard bus-arbitration scheme that allows for lost

packets to be resent. CAN allows for error detection and correction and system-wide data consistency. In the event of a fault, the network can automatically detect the difference between a temporary and permanent failure and autonomously disconnect defective nodes on the network [5]. Like SAE J1708, CAN supplies only communications and does not provide any power for remote nodes.

An emerging standard in the trucking industry that combines communications and power is known as "PLC4TRUCKS." This protocol, referenced under SAE standard J2497, utilizes SAE standard J1708 for its message structure, and uses a spread spectrum modulation technique for sending data [13]. By modulating the communications signal on a power line -- the anti-lock braking system (ABS) signaling line in this case -- power and communications can be supplied to remote loads via a single conductor. Currently, PLC4TRUCKS is utilized only to transmit ABS status information. The J2497 standard states that some electrical control units, or ECUs, connected to the system may cause severe attenuation and distortion of the communications signal. Under that condition extra filtering hardware is necessary to isolate the ECU from the communications line. This protocol could provide both communications and power, but may require the additional installation of complex hardware.

A recent draft amendment to the Ethernet standard (IEEE 802.3) provides for "Powered Ethernet." From a communications standpoint, the protocol is no different than standard

Ethernet and is, therefore, a high-bandwidth (100 Mbps) full-duplex system. Powered Ethernet operates in conjunction with standard Ethernet data to power remote data terminal equipment (DTE) such as routers, telephones, or network hubs. These devices are all low-power devices, operating at less than six watts. The power is delivered either via two spare pairs of wires in twisted-pair cabling or through a series of coupling transformers.

1.2. Statement of Thesis Solution

The system presented in this thesis is a functionality combination not available through state-of-the-art solutions. Although solutions such as CAN and J1708 exist within the trucking industry, these systems do not allow for power distribution to be coincident with communications in the same line. SAE standard J2497 allows for coupling of a spread-spectrum signal to a power line of the existing 7-Way, but a great deal of filtering hardware is required to keep the communications from interfering with the tractor's engine control units. Additionally, noise from the tractor can affect the trailer communications. This standard is also a retrofit of a communications system over an existing conductor, rather than a fundamental design of a combined communications and power system. Finally, while powered Ethernet (IEEE 802.3af) could be adapted to the

commercial trucking industry, it was not designed for high-power loads, such as automotive lamps.

From an economic standpoint, if the process of replacing a damaged umbilical could be made cheaper, and/or less time consuming, the owner of the trailer would save money. Currently, it costs $1,000 and requires twelve hours to replace an existing 7-Way [4]. The time that a trailer is not in service can itself result in significant loss of income for a trucking company. Not only is the trailer out of service for the time required to do the maintenance, there is also the time and cost of transporting the trailer to a repair facility. It is possible to retrofit a legacy system with the solution that is presented here because of the requirement for only a single conductor. By using the existing wiring harness the speed of installations would be increased and material costs would be lower.

The system that is presented in this thesis consists of several functional blocks, which will be explained in further detail in Chapter 2. These blocks include a power and communications decoupler, a data receiver, a function decoder, and a power switching solution. The research presented focuses primarily on the development of the receiver of this system, but a transmitter designed and implemented is presented for completeness.

Chapter 3 focuses on the implementation and proof of concept of this system based upon

Consumer Off The Shelf (COTS) parts. Chapter 4 discusses the minimization of the system into a custom VLSI integrated circuit. Chapter 5 provides a comparison of the two implementations with regard to packaging, economic, and performance factors. Finally, Chapter 6 provides the conclusions that can be drawn from this research and states what further research can be conducted.

2. Design Considerations

In looking at the system that was to be designed and implemented, there were four functions that formed the basic building blocks for the receiver. As can be seen in Figure 2, these four blocks were the Power and Communications Decoupler, the Data Receiver, the Function Decoder, and the Power Switching Circuitry. These four blocks are described below in sections 2.1 through 2.4 in terms of their functional inputs and outputs, as there can be several different implementations that achieve the same functional goal.

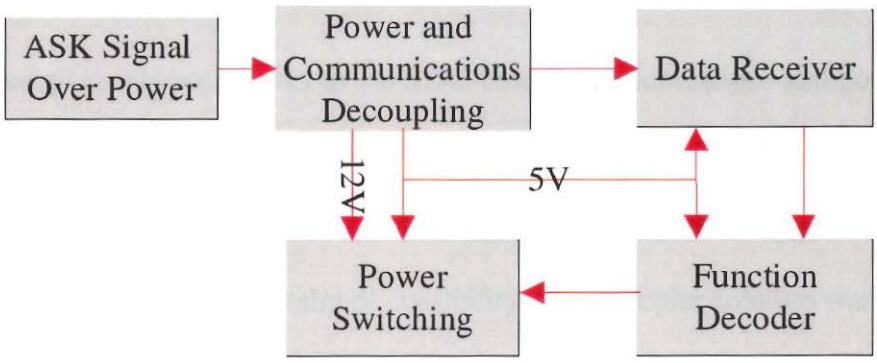

Figure 2. Conceptual System Diagram

2.1. Decoupling of Communications and Power

The primary input to the receiver comes through the Communications and Power Decoupler. This system receives the combined communications and power signal and produces a separate communications signal as well as power to drive the control logic of the receiver. One assumption made in the design of the decoupling circuitry was the need to be completely passive. Since this circuit was the one that generated the system power for the rest of the control logic, there was not necessarily a guaranteed voltage that was available to drive any active circuitry that might be used.

A second design consideration was to use a DC-based communications and power system. Direct Current power is readily available on vehicles; therefore, converting to an AC-based system would require either retrofitting the tractor or utilizing a DC-AC converter. Neither is out of the realm of possibility, but a simpler solution was to restrict the design of the receiver. If the tractor's entire power grid required an upgrade to implement this system, then it could not be retrofitted to be used on all trailers. Maintenance personnel would not have to be trained if the tractor's power grid remained unchanged.

Additionally, using a DC-AC converter to power an AC receiver has its own difficulties. Since all the load power that is used on the trailer comes through the same line as the communications, the DC-AC converter would need to be able to withstand large currents. Any inefficiency in DC-AC conversion, or AC-DC conversion at the receiver, would require more capacity from the transmitter. In addition, when coupling to an AC power line, coupling transformers are almost always required. A transformer is generally going to be larger than the passive circuitry required to remove DC voltages. The transformer can also become saturated by large currents passing through them, which would cause an inability to supply enough power to a load.

One final design consideration that was necessary was that the decoupling circuitry should be as small and compact as possible. In order to be retrofitted into the same space that is utilized by a semi-trailer light, it had to be able to fit behind or within the light.

2.2. Data Receiver

The Data Receiver functional block was one of the most critical pieces of the system. There are numerous communications protocols from which to choose, so a number of design criteria are needed merely to narrow the search for an applicable communications protocol.

The communications should not be a differential type. Initially, one would figure that a differential system would be preferable for the noise immunity that is associated with differential signals over long distances. The primary reason a differential protocol could not be utilized was the desire for the communications and power to share a single conductor. A second issue preventing differential communication was that it required twice the decoupling circuitry, as each signal (Data+ and Data-) needed to be coupled and decoupled from separate power distribution lines.

The data communications had to be asynchronous. If the communications was synchronous, the synchronizing signal needed to be transmitted some way. Usually this was through another conductor, which had already been deemed not desirable for this system. One could do the synchronization through a wireless signal, but this added unneeded complexity when an asynchronous signal would be sufficient.

The communications line and data receiver had to accommodate a reasonable data baud rate. Most vehicle protocols use 9600 baud. Controller Area Networks have higher bandwidths, but are also required to send a larger volume of data than was necessary for this system.

A Serial Component Interface (SCI) data protocol will adhere to each of these design criteria. Therefore, the data receiver in the design implemented this protocol. A sample SCI timing diagram is shown in Figure 3. The primary waveform that is presented here is the topmost wave, which shows a representation of the SCI protocol. Selecting SCI as the basic communications protocol allowed the use of any standard Universal Serial Asynchronous Receiver / Transmitter (USART).

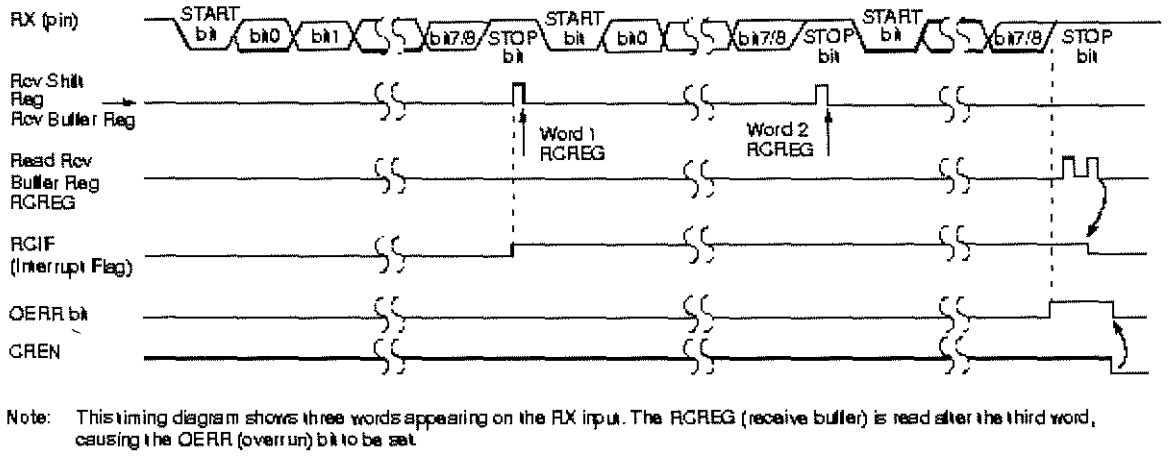

Figure 3. Sample SCI Timing Diagram [8]

2.3. Function Decoding

The Function Decoder's primary function was to perform a mapping from inputs to outputs. The decoder received data from the Data Receiver, and decoded the data bytes into control signals, depending on how the configuration bits were set. The major design factor for the function decoder was the complexity of the data protocol. The complexity

of the data protocol related to the amount of processing that was required to get the actual control data out of the communications signal. Minimization of the processing of the communications signal reduces the hardware that is required for the function decoder. The only requirement for function decoding was that the minimum functionality needed to match a direct replacement of a 7-Way umbilical. However, additional functionality may be added to enhance the system.

2.4. Power Switching

Once the data was read and decoded, the system then had to control the power that was delivered to the load. In this case, the load was a light that was attached to the semi-trailer. The lights that are utilized on a semi-trailer are either incandescent or LED-based, and the system needed to control either type of light. Therefore, the switching capacity allowed for the highest load, which in this case was incandescent, and had to allow for multiple lights connected in parallel in a single circuit.

The power-switching unit was modeled in two different ways depending upon its load. If a load required a constant current, a designer would implement a current-source power switching system. If the system required a constant voltage for its load, then a voltage source would be the appropriate type of power switching system. In this application an

incandescent bulb was considered the standard load, which required a voltage source implementation.

When looking at controlling high-power loads, two different types of switching could be utilized: hot and cold. The difference between the two is in the fact that hot switching puts the controlling switch (for human interaction) directly in line with the circuit. Cold switching uses a low-power control line to control a relay or solid-state switch, which controls the power delivered to the high-power load. This system implemented cold switching, which was inherently safer for the operator, because the high-power elements of the system could more easily be isolated from the operator.

2.5. Failure Modes

One last set of design considerations addressed was the failure modes of the system as a whole. Several modes of failure exist in the standard 7-Way that is currently utilized on semi-trailers. Short-circuit failures, where the line is shorted to ground, will render that particular circuit of lights useless because a fuse that is in line with that circuit blows, protecting the rest of the tractor's electrical system. Should one line become shorted to another line, the lights would work, but both sets of lights would be on simultaneously. If the combined load of the two circuits exceeds the fuse that is designed for short circuit protection, then eventually the fuses on both circuits blow.

An open-circuit condition is another failure that can occur. The open circuit condition is caused by one of two things. Either the wire has become damaged or the light is burned out. In the case of a burned out light, a single lighting element does not illuminate, assuming that all lights in the circuit are wired in parallel. In the case of a broken conductor, any of the lights further from the cab than the disconnection do not illuminate.

Through the network topology of the distribution lines of the system presented here, a single open-circuit failure caused by a conductor becoming damaged still allowed other lights to illuminate. This particular design consideration was implemented in the design of the transmitter. If the communications and power operated over a single conductor that ran from the front to the back of the trailer, then if that conductor became damaged, all the lights on the trailer suffered. Through the utilization of a 'looping' type of network architecture, where a loops throughout the trailer and back to the transmitter, a single break in that conductor caused the power to be sent down the other side of the loop, still supplying power and communications to all the loads.

3. Commercial Off The Shelf Implementation

The first step in the process of developing the proof of concept of the Distributed Control System was to create the system utilizing Commercial Off The Shelf (COTS) parts. The complexity of COTS parts ranges from discrete parts, like FETs and resistors, through integrated systems. All of the implementation work presented utilized discrete components that were integrated to achieve the specified purpose.

3.1. Data Transmitter

The Control System transmitter is presented for the purpose of completeness and to see how the entire system was implemented and integrated. The transmitter utilized several switches to create an input data byte that was read by a PIC 16F873 microcontroller. The data byte was sent to the modulation circuitry via the Serial Component Interface (SCI) protocol. The modulator circuit converted the SCI signal to an 8/12 V digital signal. No frequency shifting was done; the modulation circuitry merely put a DC offset on the communications line. The modulation circuitry worked by using a power P-Channel Enhancement Mode MOSFET circuit to switch between the 12 V input bus and a regulated 8 V bus. The microcontroller did not directly control the P-FET because it did not have the appropriate voltage range to turn the P-FET completely off.

The schematic, as developed by Quentin Kramer and myself for the transmitter, is shown in Figure 4. All of the COTS parts were readily available and purchased from Digi-Key, Inc. The C source code for the microcontroller is available in Appendix A.

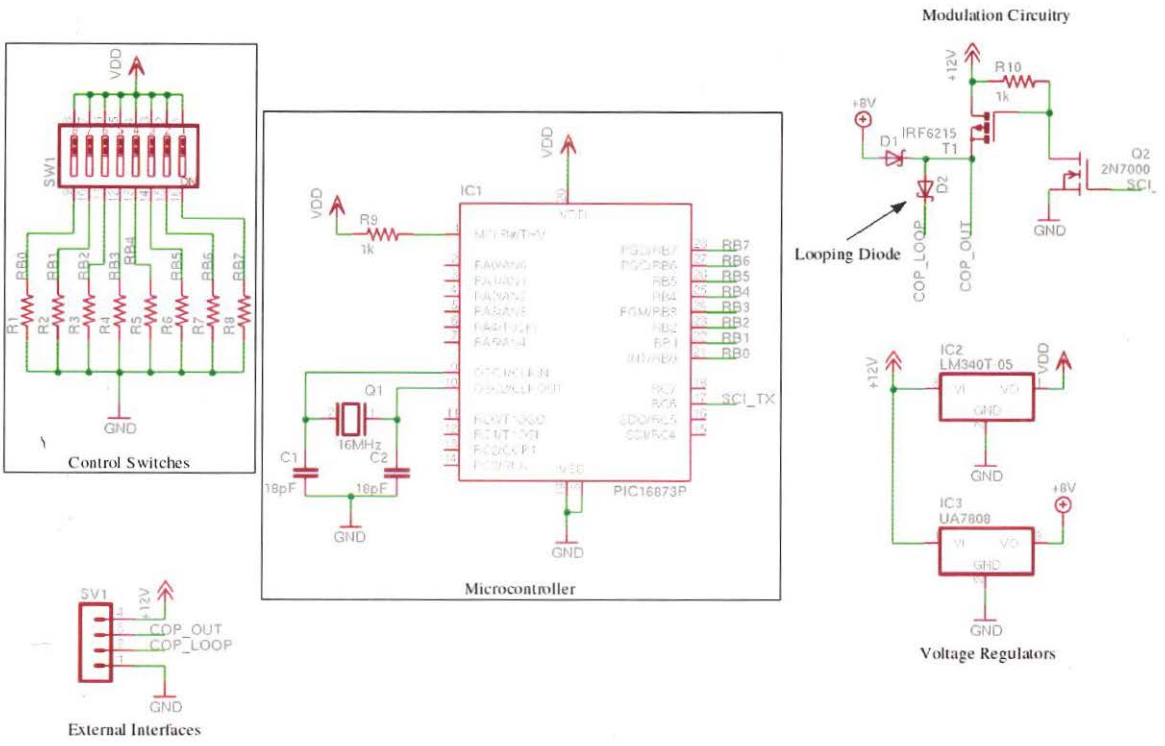

Figure 4. Transmitter Schematic Diagram

Also shown on the schematic are the 'looping' diodes. The 'looping' diodes allowed for the 'looping' network architecture because, in general, current flows one way around the loop. If the line became damaged, then the current (and signals) flowed the other

direction around the loop to the affected receivers. The use of diodes made sure that signals were not back-fed into the modulation circuitry, and minimized the voltage drop on the loop.

When tested, the transmitter worked as expected. The data rate was 9600 baud, and the voltages of the modulator were 12.00 V and 8.03 V for the logic high and low levels, respectively. Waveforms of the input SCI data signal and the output of the modulator are presented in Figure 5. Both waveforms are provided to show the fact that the modulated signal is the same as the input signal.

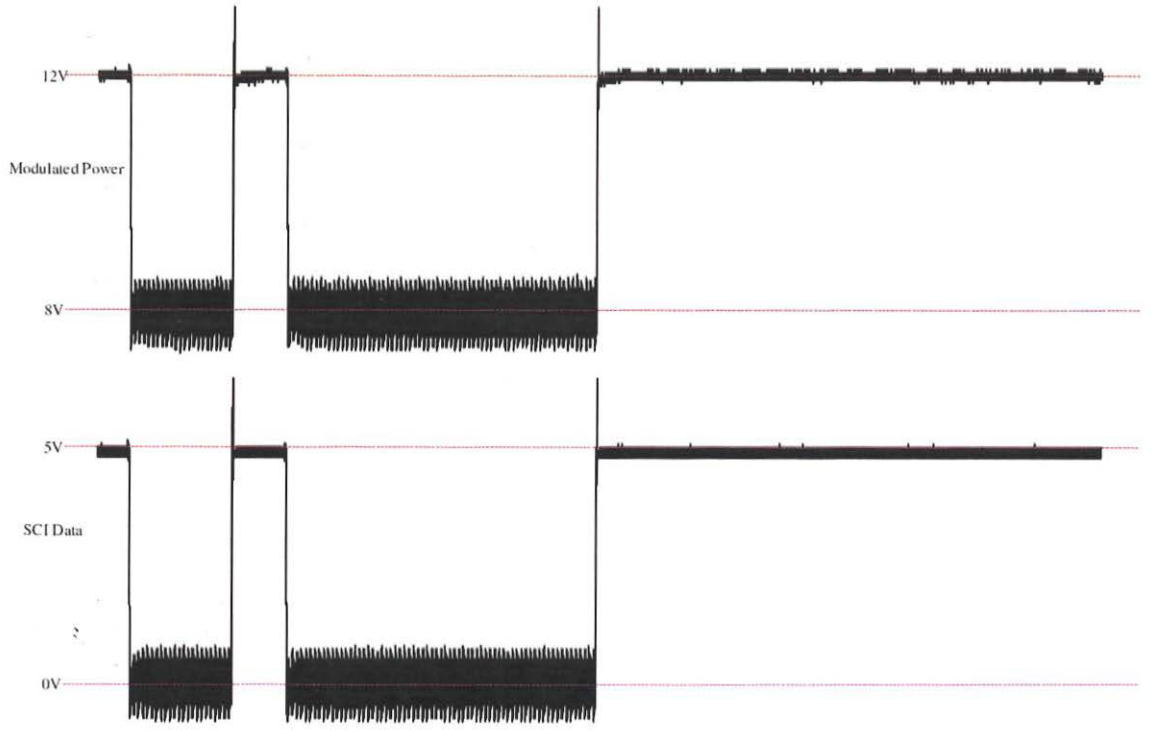

Figure 5. Comparison of Modulated Power Signal and SCI Data

3.2. Receiver: Decoupling of Communications and Power

Within the receiver circuitry, the first function that was performed on the incoming signal was to decouple the communications signal from the power. There were two types of power that were made available in the receiver. The first was power to drive the load that was on the output of the receiver. The second type of power was the lower voltage power necessary to run the control logic.

In the schematic that is presented in Figure 6, the power generation circuitry works directly from the incoming signal. An LM340-T 5.0 V regulator was used to regulate the incoming voltage down to a level that powered the control logic. The regulator had a 1.0 A maximum current; however, this power limitation was not an issue because the low-voltage components typically drew less than 275 mA.

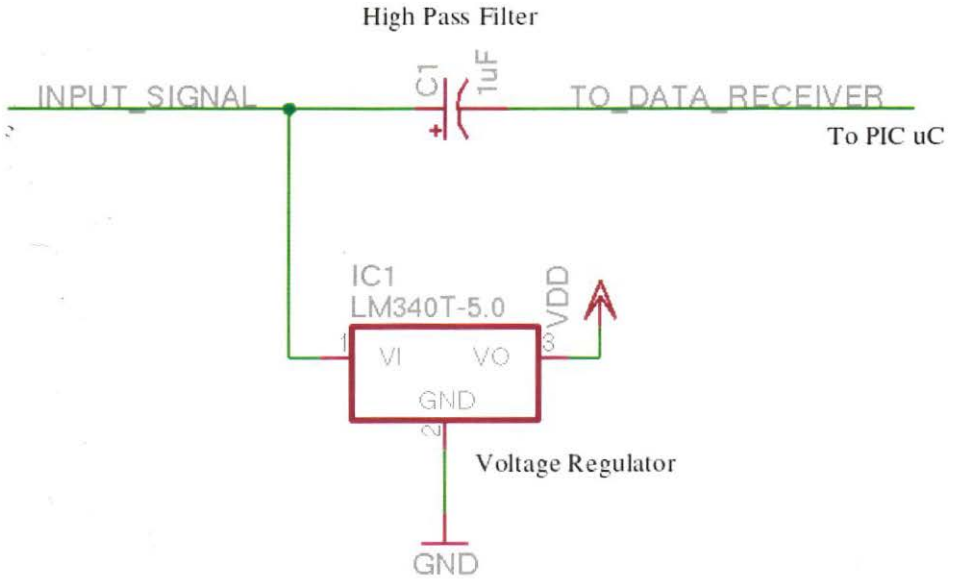

Figure 6. Schematic Diagram of Power and Communications Decoupling Circuitry

The load power was a direct connection from the incoming signal to the power-switching circuitry. When the SCI data line was idle, the voltage that was present is logic high. Consequently, this 'idle voltage' on the modulated power line was 12 V. The line is

always at 12 V, except when data is being transmitted. In the worst case of a byte of all zeros (00000000), the line was at 8 V, thus dimming the light for approximately 1ms, which was imperceptible to the human eye. Therefore, it was not necessary for the system to transmit with a higher voltage (e.g. 15 V) that was then regulated down to 12 V. The specific application of this system precluded the need for data to be sent constantly, which could generate a lower average voltage on the line.

A series capacitor decoupled the DC offset from the communications signal. Through testing, it had been noted that the decoupling was insensitive to the value of the capacitors in the range from 100 pF to 10 μF. A single 1 μF capacitor was employed in this design, as can be seen in Figure 6.

The initial decoupling of the signal produced an output signal that ranged from -2.5 to 2.5 V, which was incompatible with the control logic that operated from 0 to 5 V. Through experimentation, it was found that simply connecting the signal to the Universal Serial Asynchronous Receiver/Transmitter, or USART, receiver pin of the PIC microcontroller added the appropriate offset. The reason that this worked was probably due to internal circuit protection that maintains the signal from going below 0 V, and possibly an internal pull-up resistor to pull the signal up to 5 V. Implementations of this

system with other microcontrollers or control logic could necessitate the implementation of a 2.5 V offset to move the logic to the appropriate voltage levels.

When implemented as a standalone system on a breadboard, the system acted as expected. The voltage output of the voltage regulator of the logic power was 5.016 V, which was within the tolerances set forth in the data sheet from National Semiconductor. Placing a 5 Ω resistor (5.86 Ω measured) across the output of the regulator verified that 899.2 mA of current could be drawn, which was sufficient to power the control logic.

Using a Tektronix FG507 function generator, a signal that oscillated between 8 V and 12 V (8.00 V and 11.75 V measured) was generated to test the communications decoupling. As expected, without the PIC connected the output, the decoupling ranged from -1.69 V to +2.00 V. When connected to the receiver pin of the PIC microcontroller, the output waveform was shifted to 0 V and 5 V, respectively. Figure 7 shows both the modulated input signal and the output of the demodulator, showing that both signals are the same, other than voltage level. Given these conditions, all the circuitry of the Communications and Power Decoupling circuitry worked as expected.

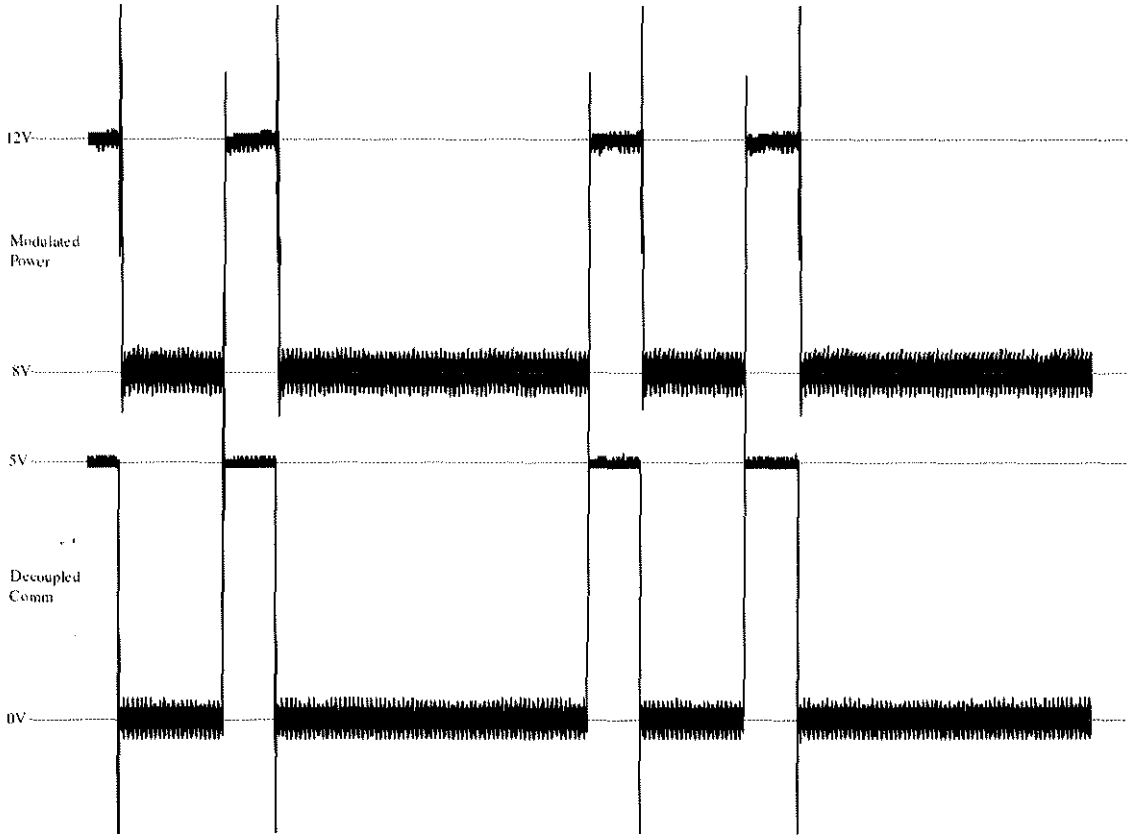

Figure 7. Modulated Power and Decoupled Communications Showing Two Data Bytes

One issue that could arise in this system is the effect of cable resistance on the power levels. A theoretical example was computed, assuming 100 feet of 12 AWG wire. In order for the lower voltage, 8 V, to remain above 5.5 V to maintain power to drive the control logic, the cable could lose 2.5 V. A semi-trailer is approximately 50 feet long, so a good approximation of the length of wire used to form the loop is 100 feet. A 2.5 V drop along 100 feet of wire (164 mΩ) resulted in 15 A of current. This allowed thirty

6 W lights to operate on this system. Most trailers contain approximately twenty 6 W lights, resulting in a nominal current draw of 10 A. A 10 A current draw caused a 1.64 V drop on the wire, which was reasonable. In a case where more than 30 lights were utilized, it was reasonable to assume that larger gauge wire would be installed.

3.3. Data Receiver

The Data Receiver portion of the system was implemented as part of the PIC 16F873 microcontroller that was used for the Function Decoder. This particular microcontroller included a hardware USART. The hardware USART received the asynchronous data and placed the received data byte into a register that the Function Decoder could access. The receiver was configured to accept a standard 9600 bit-per-second asynchronous signal.

Data buffering was the only function of the Data Receiver implemented in the microcontroller code because the USART module itself was implemented in hardware on the PIC. This code handled interrupts when a byte was received and passed the byte onto the function decoder. The schematic of the Data Receiver system can be found in Figure 8. The crystal and capacitors that are included in addition to the PIC are necessary support circuitry for the microcontroller to operate.

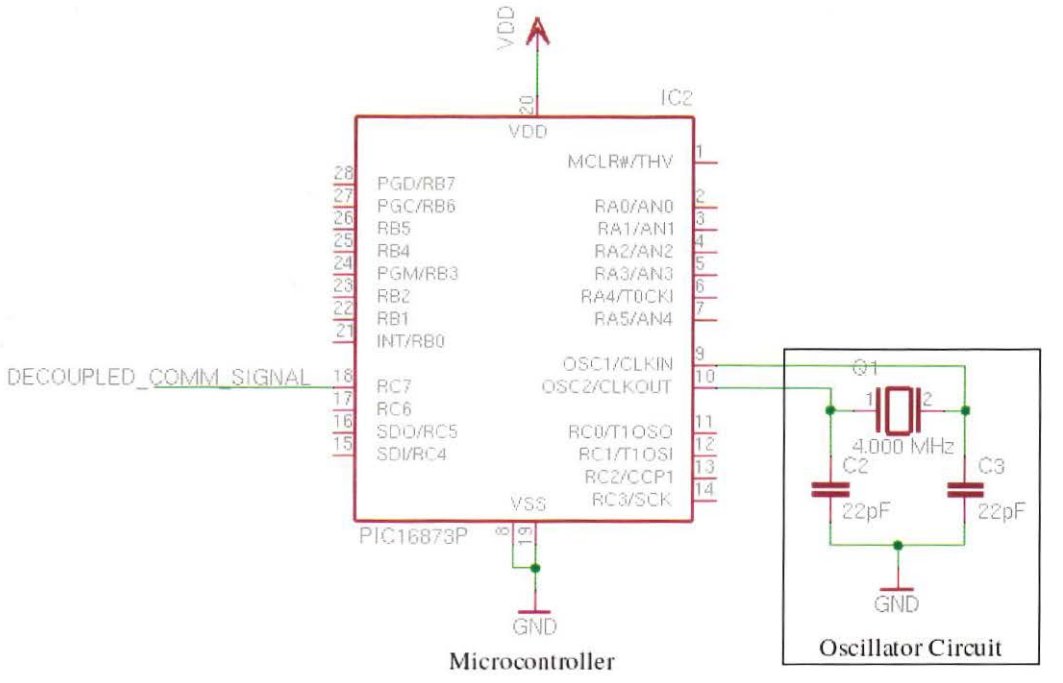

Figure 8. Schematic Diagram of COTS Data Receiver Circuit

3.4. Function Decoding

The Function Decoder of the system manages not only what sort of function was performed, but also sent control signals to the power switching circuitry. The Function Decoder took a 'function address' as well as a data byte from the Data Receiver, and determined how and when to set output lines 'high' or 'low'. Depending on the desired output, the Function Decoder could be made to perform simple on/off output or more complex multi-pin outputs. In emulation mode a light was turned on or off as if it were directly connected to the 7-Way. More complex special function modes could be used

that sequentially turn the outputs on and off to form some sort of pattern, as was shown with the Sweeping Turn Signal mode.

The implementation that was presented here utilized three input pins on a PIC 16F873 microcontroller to determine the function address. The microcontroller used eight outputs to perform the different functions. The function decoding was completely built into the microcontroller firmware and could, therefore, be modified. One receiver could utilize one set of functions, while another receiver could implement a different set of functions. The eight functions that were implemented in this design are shown in Table 1 and described below.

Table 1. List of the Addresses and Corresponding Functions

Address	***Function***
0	7-Way Emulation
1	Single Emulation Bit 0
2	Single Emulation Bit 1
3	Single Emulation Bit 2
4	Single Emulation Bit 3
5	Single Emulation Bit 4
6	Single Emulation Bit 5
7	Single Emulation Bit 6
8	Sweeping Turn Signal

Of the functions that were contained, the first, the 7-Way emulation, was a very simple function. This function simply presents the data that was contained in the data byte directly to the eight output pins. This allowed a single receiver to operate a cluster of lights that were controlled through a standard 7-Way. The 7-Way only needed functions to control lights; therefore, the three spare bits could control auxiliary lights or loads on the trailer.

The next group of functions all had the same operation, but 'listened' to a single bit in the data byte. This 'single function emulation' operated by reading a single data bit of the data byte. The value of this bit was presented to a fixed output pin to control a single set of lights. Single function emulation was used in the case that a receiver was packaged with an individual light, rather than at a centralized cluster. Placing the receiver at each light decreased the chances of a single point of failure causing more than one light to fail to illuminate. The purpose of using an 8-bit data byte was two fold. It was simpler to implement, as most systems are already designed around 8-bit data. Secondly, the use of eight bits allowed for a greater number of functions to be coded for special function use.

The last function that was implemented in this design was the sweeping turn signal. A single bit that was activated when the driver turned on a turn signal controlled the sweeping turn signal function. This function operated by turning on the output pins sequentially, such that if these outputs were all hooked up to a row of LED's they would

Library
Rose Hulman Institute of Technology

'sweep' from one end to the other. This sort of system could be utilized for a turn signal that could give other drivers additional visual cues that the truck desired to turn.

An example of an input byte and various outputs based upon function is listed in Table 2. The first column is the input data byte. The second column is the expected outputs for the system operating in 7-Way emulation mode (address 0). The third and fourth columns show the expected outputs for the receiver operating in single channel emulation modes with addresses of 3 and 4 respectively. As can be seen in 7-Way emulation mode, the function decoder acts as a pass through of the data. In single channel emulation mode, the value at bit 2 and 3 of the input byte is expressed on bit 0, columns 3 and 4 respectively.

Table 2. Example outputs of the function decoder in various modes

Bit Number	***Input Bits***	***7-Way Emulation Mode***	***Single Channel Emulation Mode (3)***	***Single Channel Emulation Mode (4)***
0	0	0	0	1
1	1	1	0	0
2	0	0	0	0
3	1	1	0	0
4	0	0	0	0
5	1	1	0	0
6	0	0	0	0
7	1	1	0	0

The function decoder was implemented on a PIC 16F873 microcontroller. The microcontroller provided flexibility necessary during the prototyping stage when it was not known what was required in the Function Decoder. This chip also provided a reasonable number of input and output pins. The entire C source code that was used in the Function Decoder and Data Receiver can be found in Appendix B for the 7-Way and Single Emulation Modes. Appendix C has the microcontroller code for the sweeping turn signal mode. The micro controller was wired as is shown in Figure 8. The choice of a micro controller with an integrated USART created the tight integration of the Function Decoder and the Data Receiver.

The Function Decoder was tested once the Data Receiver functionality was verified.

Data bytes were sent to the Function Decoder to verify that all of the functions worked correctly.

Table 3, below, summarizes the voltages present at the output of the microcontroller operating in 7-Way emulation mode. The data byte input to the function decoder is 10101010. Sample waveforms of the outputs for the sweeping turn signal are shown in Figure 9.

Table 3. Micro Controller Output Voltages

Bit (Value)	*Voltage*
0 (0)	170 mV
1 (1)	4.86V
2 (0)	175 mV
3 (1)	4.90V
4 (0)	171 mV
5 (1)	4.88V
6 (0)	160 mV
7 (1)	4.85V

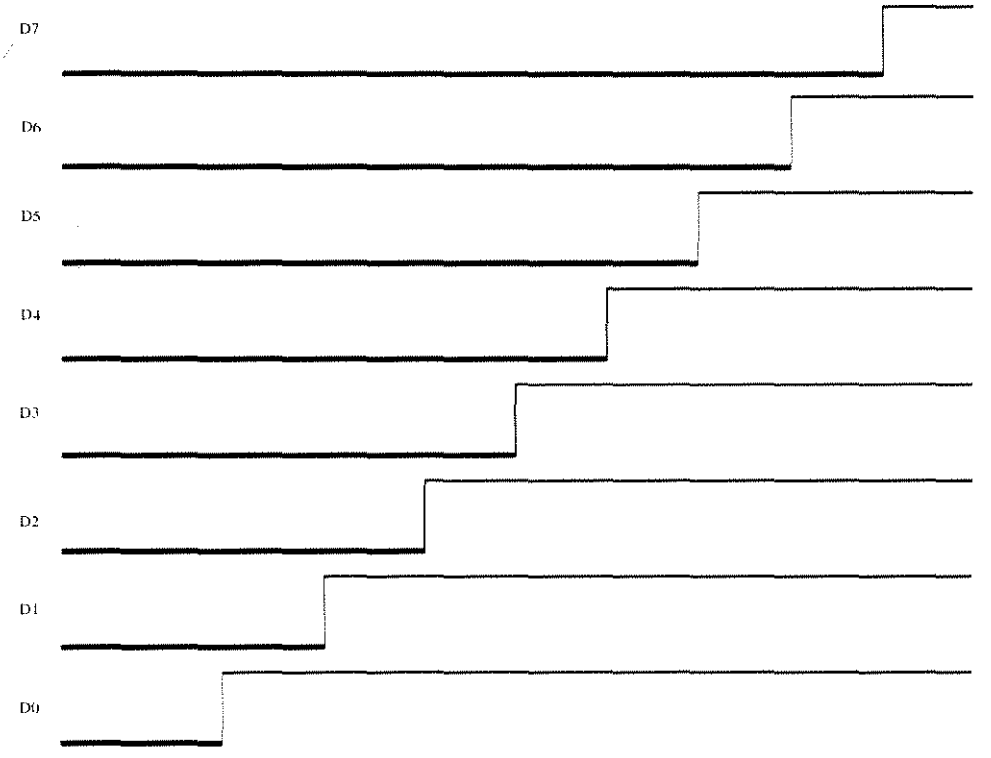

Figure 9. Outputs of the Microcontroller Operating In Sweeping Turn Signal Mode

3.5. Power Switching

The Power Switching circuitry controls the application of voltage and flow of current to the load that was being powered. Without the power switching, it was impossible to drive the high-power loads that are common on commercial semi-trailers directly with a microcontroller. The power switching methodology that was implemented is a standard high-side switching system. A high-side switching system puts the 'switch' in the system between the positive terminal of the power supply and the load. The alternative solution, low side switching, puts the 'switch' between the load and ground. High side switching is inherently safer than low-side switching, and was therefore the preferable design to be implemented.

The design that was implemented could be modeled as a switch in series between the voltage source and the load. This allowed for any current load to be connected, as long as the solid-state switch could handle the current requirements. The transmitter modulation circuitry limited the amount of power drawn by all of the receivers simultaneously. The current transmitter implementation limited the total power draw to 11 A, based upon current component selection. Each power-switching circuit was capable of supplying 1 A of current to a load. The high current requirement for the

system will slowly decline, as more and more LED-based lights, which draw less than 500 mA of current, are installed.

The final schematic for the power switching circuitry is included in Figure 10. The control signal, which was a digital signal of either 0 V or 5 V, did not have the voltage range necessary to drive the P-FET that was required for the high-side switching. The P-FET requires a V_{SG} of less than 0.7 V to turn the system off, and directly from the microcontroller, only a V_{SG} of 7 V could be applied, because the source of the P-FET is at 12 V and the maximum voltage the microcontroller can produce is 5 V. Therefore, the control signal was used to switch a N-FET, which when wired as shown can achieve the 12 V to 0 V voltage range. This system was a non-inverting control system, where logic high on the control signal closed the 'switch' (P-FET) and logic low opened the switch.

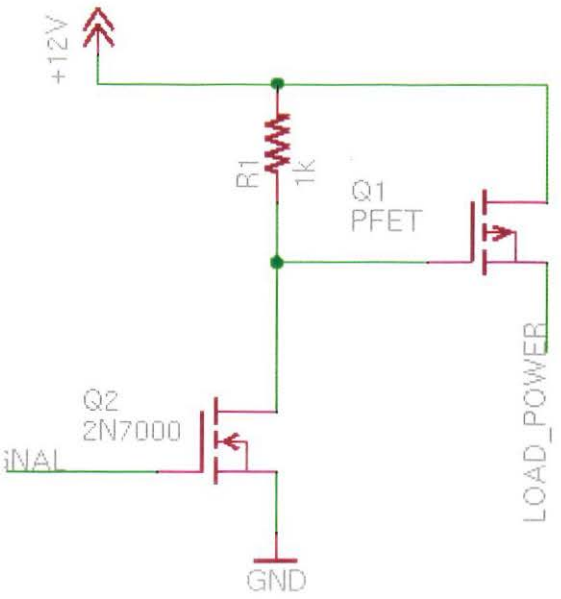

Figure 10. Schematic Diagram of COTS Power Switching Circuitry

The power switching circuitry could be tested in various ways. The simplest test was to make sure that the P-FET is actually operating as a switch. Using an ohmmeter, the resistance across the P-FET was measured. When 0 V was applied to the control line, the resistance was infinite (showing an open circuit). When the control voltage was switched to 5 V, the resistance went down to 0.27 Ω, which was effectively a short circuit. This resistance change showed that the P-FET was in fact operating like a switch.

The second test was to place an ammeter in series with the P-FET to measure the short circuit current. In an effort not to destroy the P-FET, an 11 Ω resistor was placed in

series as well to limit the current. The P-FET was indeed capable of sourcing 1015.9 mA of current when the control voltage was high and 0 mA when the control voltage was low.

3.6. Complete System Implementation

A schematic of the integrated receiver is shown in Figure 11. The design was centered on the PIC micro controller that implemented the Data Receiver and the Function Decoder. This schematic only demonstrates single channel emulation for simplicity. The design implemented for either the sweeping turn signal or the 7-Way emulation would require the Power Switching circuitry to be replicated seven more times to handle all the outputs.

This implementation of the Distributed Control System did not address many forms of faults, such as short circuit failures. Further development is required to address these issues, because a simple fuse system, similar to what is currently implemented in a semi-trailer, may be inappropriate. The failures that were compensated for, though, include open-circuit faults. If the signal distribution line became damaged, the 'looping' circuitry in the transmitter allowed for the signal to be rerouted automatically. This handled single points of failure, but any form of multiple open-circuit failures caused a failure in the lights on the affected line.

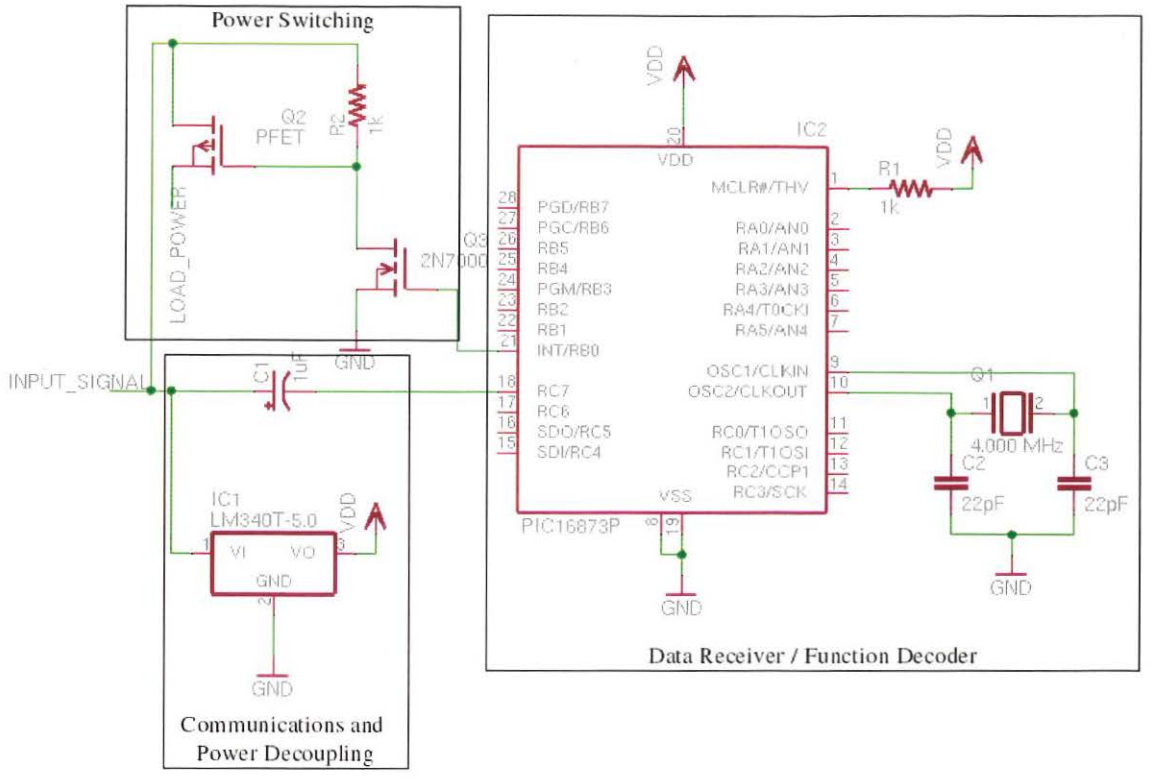

Figure 11. Schematic Diagram of COTS Integrated System

To test the system, a transmitter and three receivers were built. The receivers that were built included a 7-Way emulator, a single-channel emulator, and a sweeping turn signal receiver. To test the ability for the system to drive a high-current load, a headlight was hooked up on the single-channel receiver. The transmitter and receivers were connected via two-conductor twelve-gauge wire. A second test system was built with a transmitter and two special function receivers. One of the receivers was a sweeping turn signal, and the second special function receiver was a bi-color turn signal/brake light demonstration

receiver. Both of the receivers drove low current LED loads. All of the measurements that were done were taken from one of the two test systems.

Through the use of several switches at the transmitter all of the receivers were tested and worked appropriately. Simple toggle switches were used to emulate the controls of the driver and to set control bits high or low. All of the receivers responded to the correct input signals and functioned correctly. The single channel receiver was fully capable of driving a standard headlight as its load.

In addition to testing basic functionalities, the ability to tolerate a single point of failure was tested. When one end of the loop was unplugged, all of the receivers that were connected to the tested system still functioned without any noticeable degradation.

Given all of this data, the concept of the Distributed Control System was shown to work. It was shown that a transmitter could be constructed to combine distribution of high-current power distribution and a digital control signal. In addition to the transmitter, a receiver was built that was capable of decoupling the control signal from the power distribution line. The receiver also had the ability to control either high or low current loads.

4. Full-Custom VLSI Implementation

The second implementation of this Distributed Control System centered on a full-custom VLSI-based implementation of the receiver. In an effort to minimize the size of the receiver control circuitry, as many of the functional blocks were integrated onto a single chip as was reasonably possible. As a result, the Data Receiver, Function Decoder, and portions of the Power Switching circuitry were put onto a single integrated circuit. The integrated circuit was designed and simulated at Rose-Hulman Institute of Technology, and then fabricated by MOSIS on the AMI C5N 0.6 μm process using the scalable sub-micron CMOS design rules. All of the transistors could be designed at their minimum size because the system did not have to operate at extremely high speeds.

4.1. Decoupling of Communications and Power

During the design of the integrated circuit, it was determined for several reasons that the decoupling of the communications and power signals needed to be done externally. The first major reason for moving the decoupling off-chip was the fact that a large capacitance was needed to perform the data decoupling. In the Chapter 3, it was determined that a 1 μF capacitor was the appropriate value to use for the decoupling. Implementing this capacitor on the chip required 1.1 mm^2 of area on the chip, based on a sample of the MOSIS parameters for this process. With the rest of the circuitry that was on the chip, it was determined that it was difficult to find enough area on the chip to

accommodate the capacitor. In addition, if the capacitor were put on the chip, it would preclude the use of different decoupling schemes, as discussed in Chapter 6.

The other portion of the decoupling circuitry was the generation of the control circuitry voltages. There were several methods of implementing a voltage regulator on the chip, but all of these methods had drawbacks. A standard method of regulating voltage is the use of a Zener diode. Unfortunately, the fabrication of Zener diodes was not available in this process. A Zener diode operates through tunneling breakdown, which is due to very high semiconductor doping [15]. Because the AMI C5N process only allows for 'standard' doping levels, this solution was not a possibility for integration onto the chip.

Another possible solution was to use a resistor divider to drop the voltage. This method suffered from two drawbacks. Due to the current-to-voltage relationship of resistive elements, the output voltage of this solution varies linearly with power draw. The only way to solve this problem is to push so much current through the resistor divider that the current draw of the control circuitry appears negligible. This solution requires a large amount of power, preventing the receiver from being a low power device. One other problem that cannot be solved with this system is the fact that the output voltage is linearly proportional to the input voltage. So, if the input signal drops to 8 V from an idle of 12 V (as is the case during data transmission), the output voltage drops from a standard of 5 V to 3.3 V. This voltage drop would cause havoc in the control circuitry.

The final method that was investigated was to put a series chain of standard diodes (or diode connected MOSFETs). This would allow a fixed voltage on the output, given any input larger than the required output, but suffered from having fixed power dissipation. By using a resistor in series with the diodes, as is shown in Figure 12, the current was limited through the diodes. This limiting resistor also limited the maximum amount of current that could be drawn by the control circuitry. Although limiting the current to the control logic was not a major problem, there was still a fixed power draw no matter how little current the control logic used.

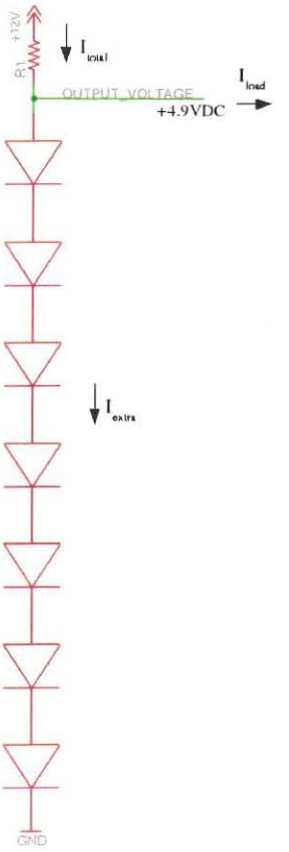

Figure 12. Sample of Constant Current Draw Voltage Reference

Given all of the factors that precluded the Communications and Power Decoupling circuitry from being integrated with the rest of the circuitry, it was determined to just utilize the same implementation as the COTS solution. The schematic of this implementation is shown in Figure 6 from the Chapter 3.

4.2. Data Receiver

The Data Receiver was the most intricate portion of the integrated circuit that had to be designed. The major portion of what made the receiver so intricate was that it was designed to receive an asynchronous signal, meaning that the beginning of reception was not necessarily aligned with any sort of clock signal internal to the chip. The receiver also had to be able to receive only 8 bits of data at a time, and then 'reset' itself and wait for the next data byte to begin. The fundamental building blocks of the Data Receiver were a shift register, a ring oscillator, control logic, and an output buffer to latch the data that was received. The output buffer was used so that when the next data byte was being received, the data being sent to the function decoder did not change until the complete byte was received.

4.2.1. Ring Oscillator for Bit Clock

To generate the bit-clock for the shift register, a ring oscillator was implemented. A ring oscillator is an odd number of inverters put in series that feeds back from the last inverter to the first, creating an oscillation. The frequency of the oscillator is based on the propagation delay from the first inverter to the last. Due to the slow bit rate of data for this system, it was determined that 'current-starved' inverters should be used in lieu of normal inverters [17]. The current-starved inverter adds a second P-FET and N-FET in series with the standard two of an inverter, as can be seen in Figure 13. The purpose of the additional transistors is to use them as current limitation devices for the inverter. By

limiting the current that is available to the inverter, the propagation delay can be increased, and therefore slow down the clock speed of the ring oscillator. The current starving transistors were mirrored to limit the current to 1.142 μA on the PFET side of the inverter and 652 nA on the NFET side. The currents were not symmetrical because both the NFET and PFET were the same size. There is different carrier mobility for N and P-type material and, therefore, with transistors that are the same size, the currents are different.

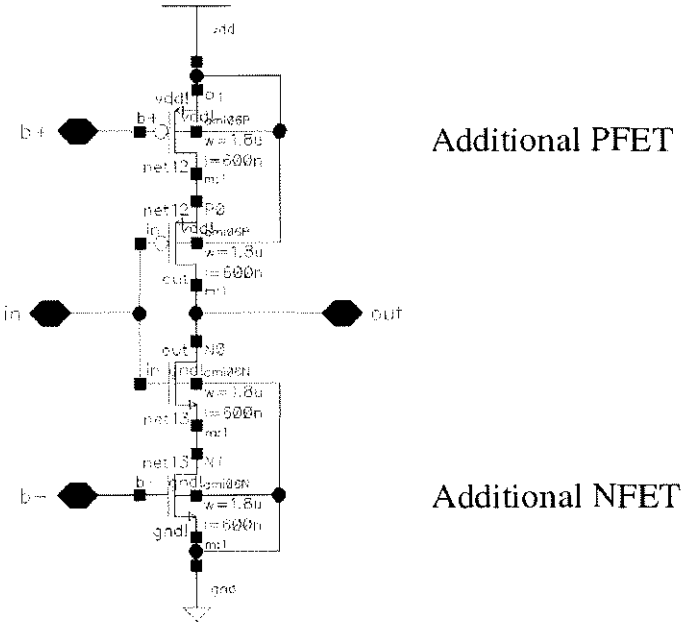

Figure 13. Schematic Diagram of Current Starved Inverter

With the first simulation of the ring oscillator, it was noted that the oscillator could not maintain enough voltage variation in the output signal to drive the shift register for all 8 bits of data. The voltage oscillated approximately 200 mV about 2.3 V, which did not pass V_{IL} or V_{IH} for the CMOS gates. To combat this problem, a small analog comparator (see Figure 14) was added to drive the output voltages closer to the supply rails. The comparator operated by comparing the difference of the input voltages. By setting one of the inputs to a fixed voltage, then if the other input was higher or lower than the fixed voltage, then the output swung high, or low, respectively. The transistors of the comparator were made with a W/L ratio of 6/2, which was sufficient since speed was not a primary concern in this design.

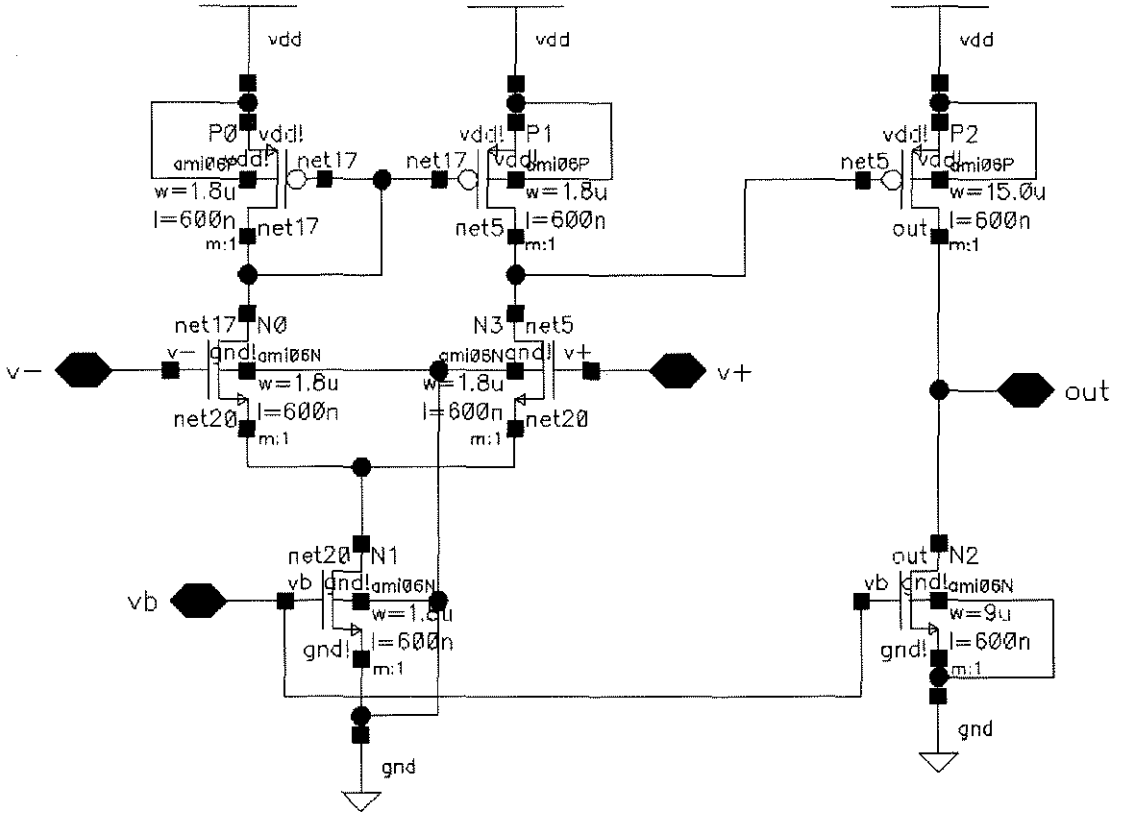

Figure 14. Schematic Diagram of the Comparator Used in the Ring Oscillator

When laid out, this section was treated more as an analog circuit than a digital circuit, as shown in Figure 15. With the current of the current mirrors so small, the mirrors were highly sensitive to substrate noise. This fact, and the addition of the analog comparator, were the factors that led to treating the ring oscillator as an analog circuit. To account for the circuit's noise susceptibility, guard rings were added surrounding the ring oscillator. Separate guard rings surrounded the oscillator and the current sources. The guard rings are rings of active region that surround the circuitry, that are designed to create a

forward-biased diode between the substrate and ground to draw stray carriers away from the sensitive circuitry [3].

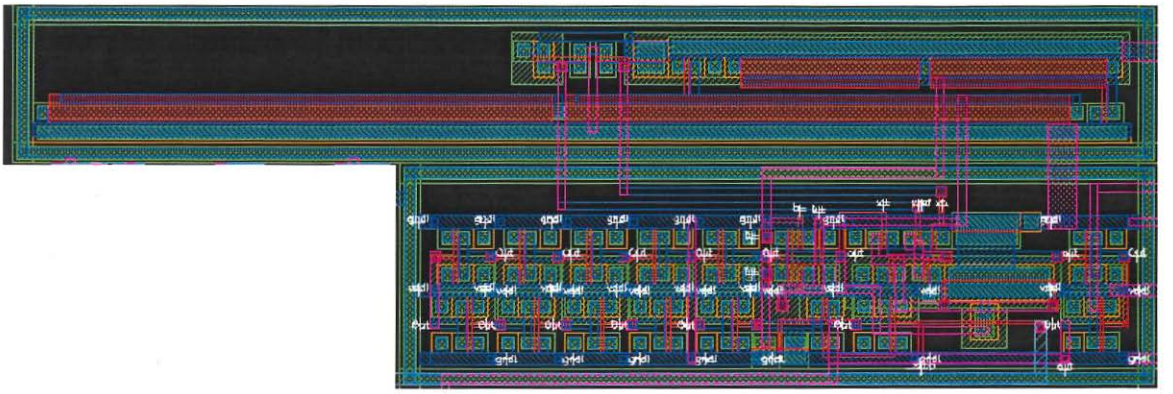

Figure 15. Layout of Ring Oscillator and Supporting Circuitry

4.2.2. Output Buffer

A simple digital output buffer was added to isolate the data that was being presented to the Function Decoder from the shift register. Due to the nature of the serial in / parallel out shift register, the data sent to the Function Decoder would change as a new data byte is received if this buffer were not present. However, the Function Decoder should not be active until the entire byte is received. The structure of the buffer had eight D-Flip Flops in parallel (one for each of the outputs) that were all clocked by a single line. The clock line comes from the control circuitry and was strobed once eight bits had been received.

4.2.3. Control Circuitry

The control circuitry for the data receiver was the portion of the circuit that handles asynchronous data reception. The first part of the control circuitry was the start-bit detector. This circuit needed to be able to detect a high to low transition, signifying the SCI data stream start bit, during the time that the receiver was not receiving any data. The reception of the start bit was accomplished by inverting the incoming data signal. The inverted data signal was connected to the clock of a D-Flip Flop. The D-Flip-Flop was wired up to pass logic high to the output when the start bit was detected. The output of the flip-flop was high until all eight bits had been received, and then the flip-flop was reset to logic low. The logic high signal from the start-bit detector enabled the ring oscillator. Figure 16 shows a simplified schematic of the start bit detector.

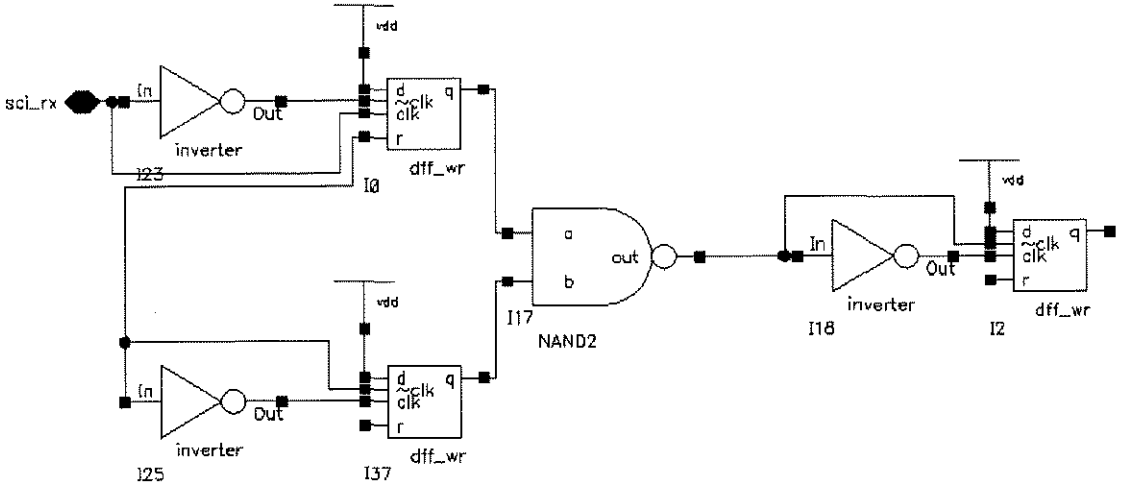

Figure 16. Schematic of the Start Bit Detector

A bit counter was connected to the ring oscillator to determine when the eight data bits had been received. The counter design used multiplexers for the next-state logic. This was different from standard counters, which typically use combinational logic to determine the next state of the counter. Typically, T-Gate logic, upon which this was based, is smaller in layout than combinational logic. Also the layout of the counter may be simpler because of the modularity of the building blocks of the counter. For timing considerations, the counter was modified to work on a two-phase (2-ϕ) non-overlapping clock signal. The 2-ϕ non-overlapping clock was used to form a double-buffered output for the counter. The first buffer holds the value that was used to compute the next state. The second buffer was used to buffer the output, once the first buffer had settled. Once eight bits were counted, a reset signal was sent to the control logic to get it ready for the next data byte.

A delay line was added to reset the counter. Due to the 2-ϕ nature of the counter, if the reset signal did not occur at the right time, the counter did not get fully reset and, therefore, counted seven instead of eight bits on subsequent receptions of data. Figure 17 (a) shows what the output would be like before the delay line was added, and Figure 17 (b) shows the result after the delay line was included. It is apparent that the reset signal happens before the second phase clocks the counter in the first diagram. The delay line utilized an even number of serially linked Schmitt Trigger inverters. These inverters

were based on a design presented by Uyamura in his book *Circuit Design for CMOS VLSI* [16]. With the addition of the delay line, the counter was reset correctly.

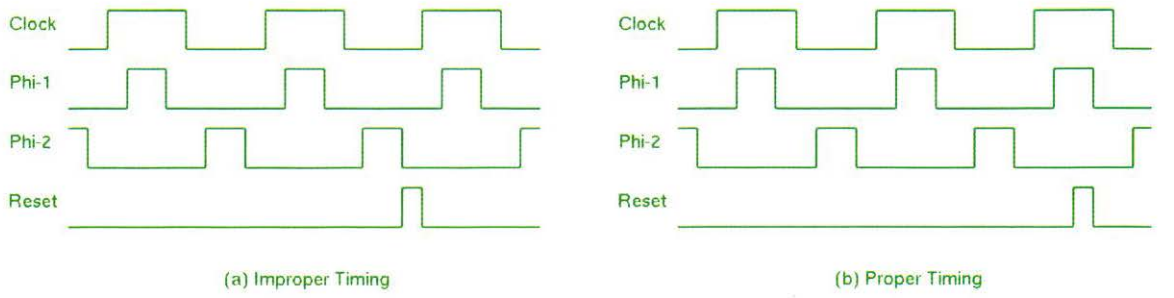

Figure 17. Timing Diagrams for 3-Bit Counter

4.2.4. Shift Register

The shift register that was implemented was based on a standard D-Flip Flop-based shift register [6]. The design of a standard D-Flip Flop-based shift register utilizes 8 D-Flip Flops that are cascaded together to form an 8-bit serial in / parallel out shift register. A clock signal is used to shift the data, and a reset signal is available to reset all the values to logical low. This particular shift registers differed from a standard shift register in that it utilized a 2-ϕ clock and double-buffered flip-flops for each stage. A schematic of the shift register can be seen in Figure 18 (a), which utilizes the D-Flip Flop as is shown in Figure 18 (b).

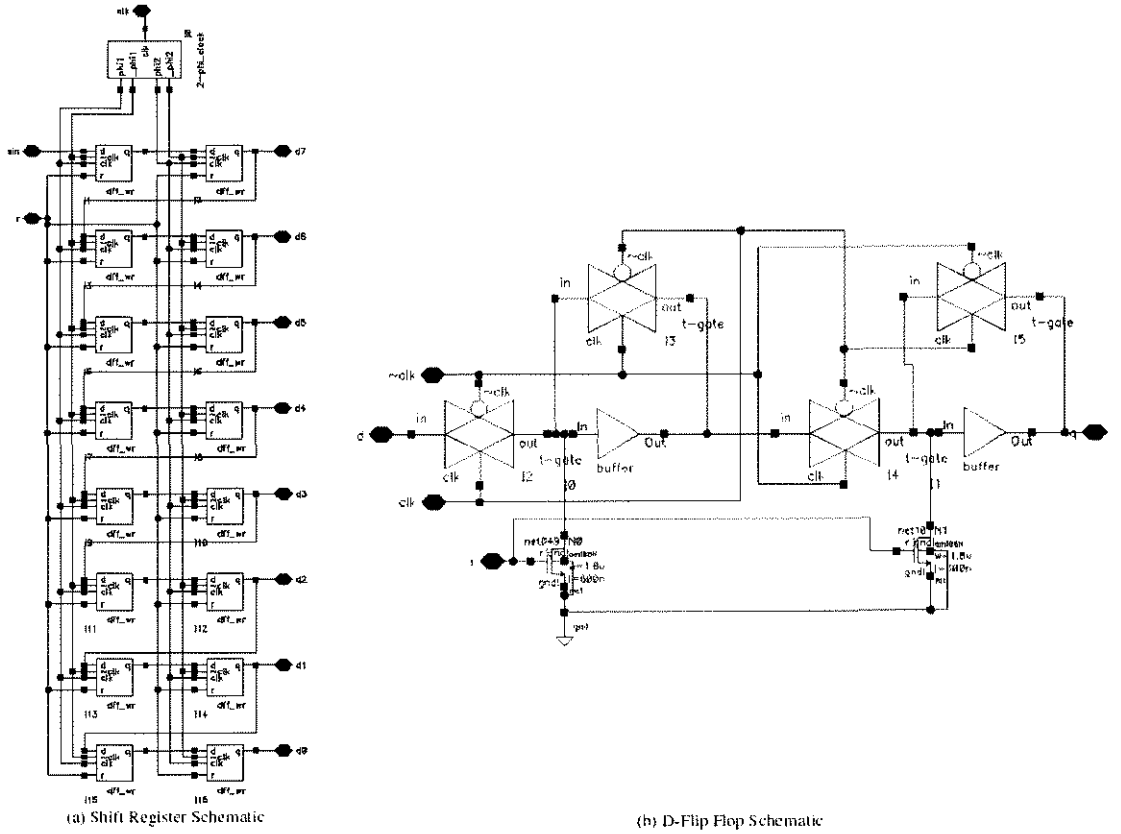

Figure 18. Schematic Diagram of Shift Register and D-Flip Flop

4.2.5. Receiver Testing and Results

A schematic of the VLSI Implementation of the Data Receiver is presented in Figure 19 to show how the building blocks were combined. Most of the control circuitry is shown as discrete logic elements. The larger blocks, such as the shift register, the ring oscillator, and the bit-counter are shown as integrated 'black-boxes' on the schematic.

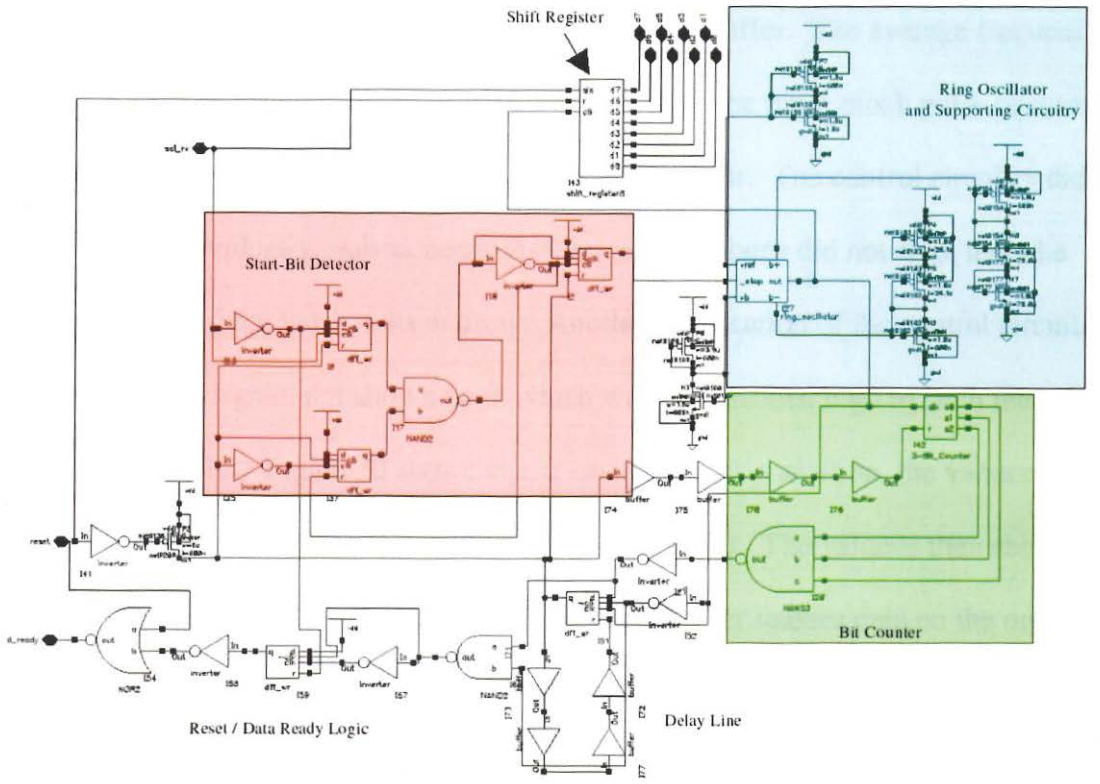

Figure 19. Schematic Diagram of VLSI SCI Receiver Module

Upon simulating the Data Receiver, the system worked as expected. A series of waveforms collected from the simulation is presented in Figure 20. The signals from top to bottom are bit7, bit6, bit5, bit4, bit3, bit2, bit1, bit0, data in, and the bit clock. The stream that was presented on the data input was a simple oscillation that when sampled will have 11111000 as its bit pattern. As is seen, when the data in first transitions from high to low, the ring oscillator started up, and oscillated until it had generated nine clock

pulses. The first eight clock pulses shifted the data into the system, and the ninth sent a data ready signal that latched the bit pattern to the output buffer. The average frequency of the bit clock for this simulation was 68.21 MHz. Once the ninth clock pulse occurred an internal reset signal was sent to turn off the ring oscillator. The control circuitry did not need to be individually probed, because if the ring oscillator did not stop, then the control circuitry did not work in its entirety. Another verification of the control circuitry was the data ready signal, not shown here, which was also strobed high to latch the data into the output buffer. Figure 20 shows that at each bit clock transition, the value on the data input signal is shifted into the topmost waveform (bit7). The value is then shifted down through the rest of the shift register. The output buffer latches data on the output of the shift register when the last transition (Data Ready) occurs.

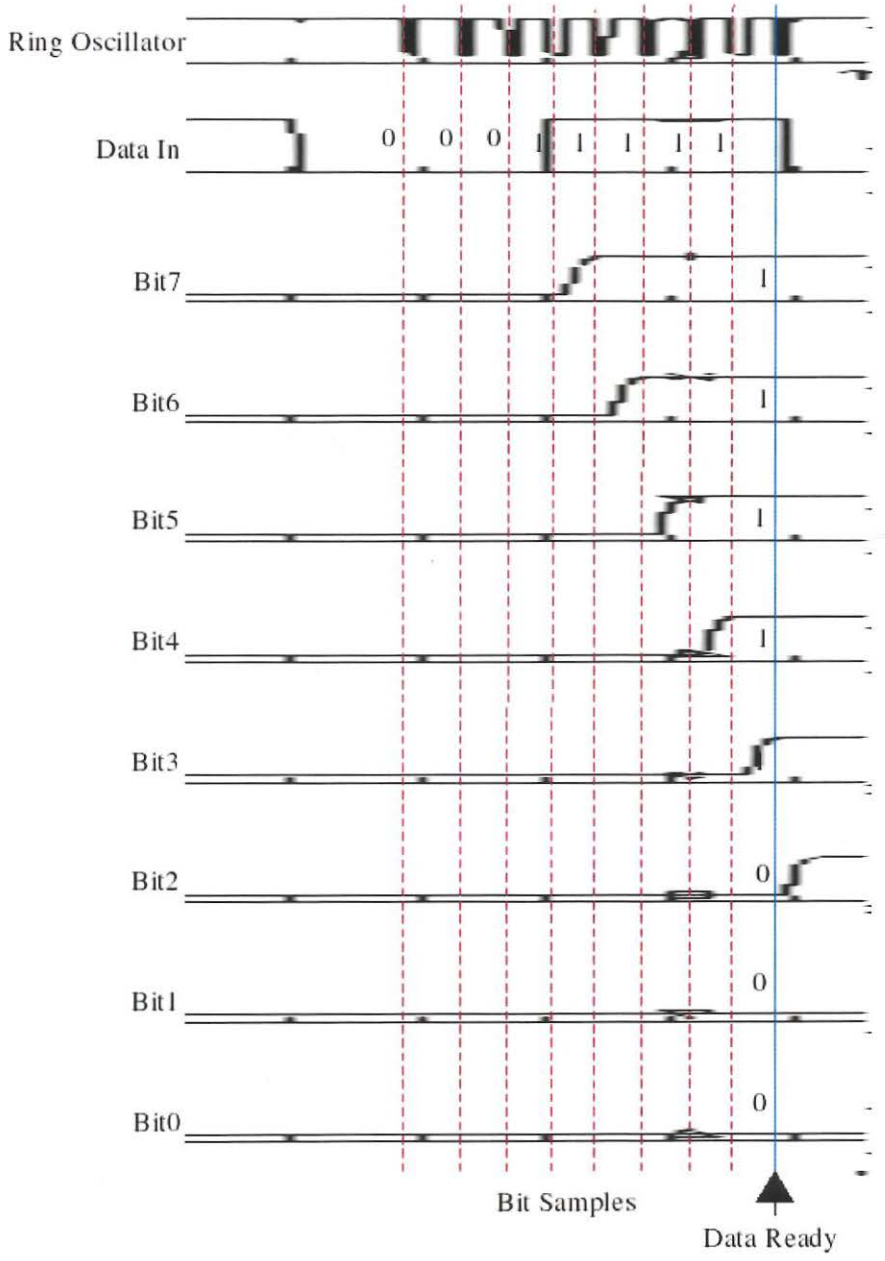

Figure 20. Simulation of Data Receiver

4.3. Function Decoding

The Function Decoder, as with the COTS implementation, mapped the data from the Data Receiver to the Power Switching circuitry. This implementation utilized transmission-gates (T-Gates) to move the data to the outputs. Through the use of combinational logic that was controlled by the function address different, T-Gates were opened or closed to connect specific outputs of the Data Receiver to the inputs of the Power Switching circuitry. The functions that were implemented for the VLSI implementation were the same as for the COTS implementation. They included the 7-Way Emulations, Single-Channel Emulation, and the Sweeping Turn Signal. Although the emulations were a simple data pass through, the Sweeping Turn Signal did require some additional logic. The Sweeping Turn Signal utilized an unbuffered shift-register, similar to the one used for the Data Receiver, and shifted logic high across the outputs to achieve the sweeping effect. The shift register was clocked by an external clock source to allow the greatest flexibility in the speed of the sweeping. A schematic of the Function Decoder circuitry is included as Figure 21.

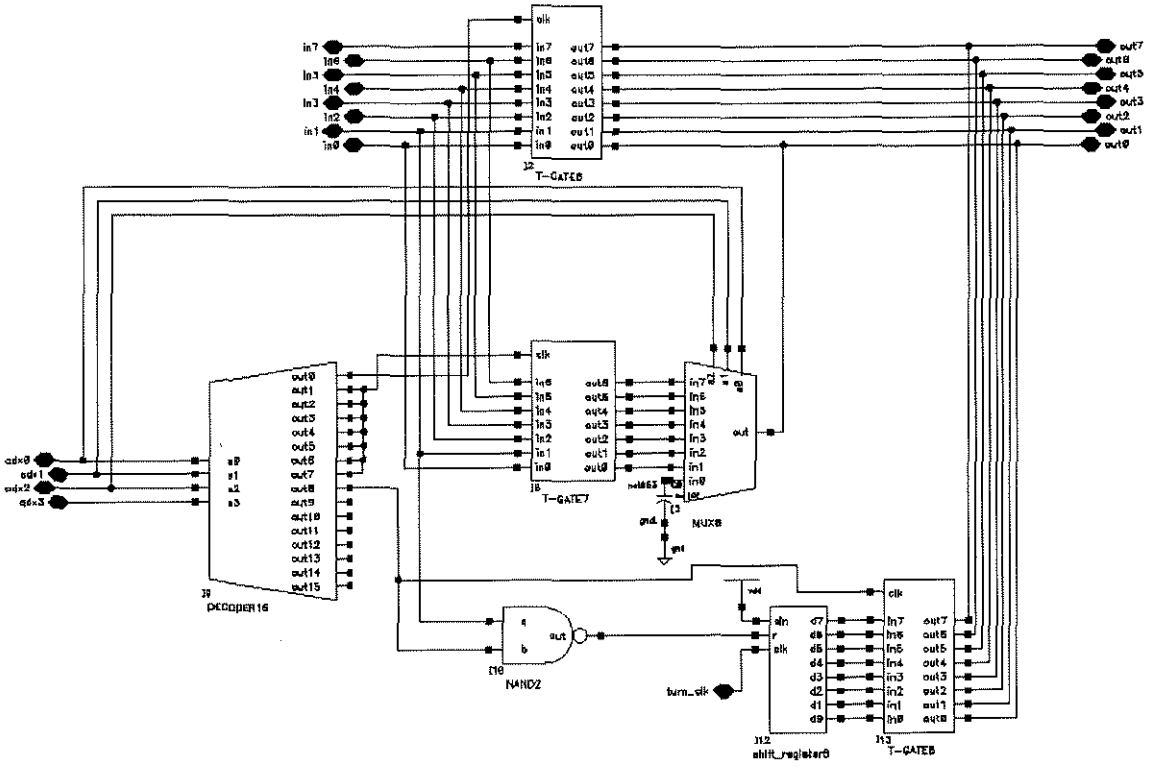

Figure 21. Schematic Diagram of VLSI Implementation of Function Decoder

When the layout of the Function Decoder was simulated, it worked as intended.

Table 4 presents the output voltage of the function decoder for the 7-Way emulation mode, which demonstrates proper operation with very little resistive drop across the T-Gates. Each of the Single Channel Emulations connected the appropriate channel to the output. Finally, the Sweeping Turn Signal swept the outputs as can be seen in Figure 22. These results indicate that the layout of the Function Decoder was correct.

Table 4. Simulated Output Voltages for VLSI Function Decoder

Bit Number	*Input Bit*	*Output Voltage*
0	0	26.7 nV
1	1	5 V
2	0	-190 pV
3	1	5 V
4	0	102 pV
5	0	-101 pV
6	1	5 V
7	1	5 V

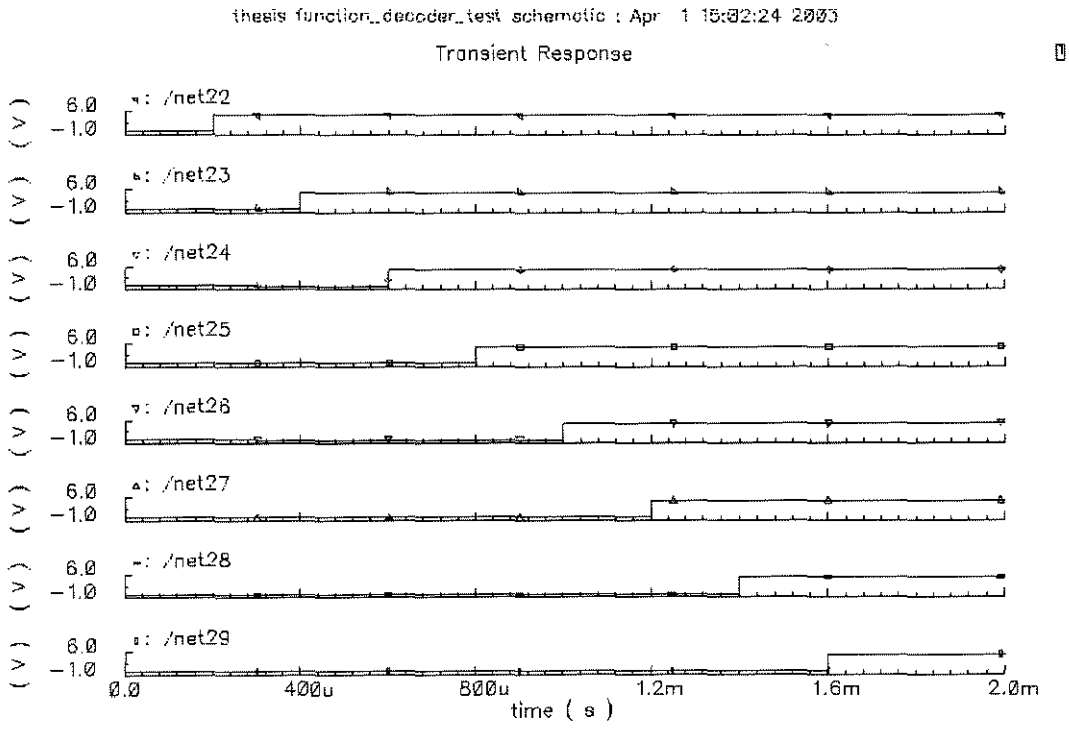

Figure 22. Sweeping Turn Signal Mode Outputs from Function Decoder

4.4. Power Switching

The final portion of the Distributed Control System was the Power Switching module. The Power Switching module for the VLSI implementation was designed to have the greatest flexibility possible. This module had two control pins; one set whether the output pin was in voltage mode or current mode, and the other determined 'active high' or 'active low' mode. In voltage mode, the active high and active low mode determined whether 5 V or 0 V, respectively, was made available at the output for a logic high input. In current mode, the module supplies current when logic high was input, and the active high or active low mode determines whether current was sourced or sunk on the output, respectively. This allowed for high power switching systems to be connected to this chip externally or low power (< 20 mA) loads to be driven directly.

The voltage source was a simple design and can be found in Figure 23. A P-FET tied the output to V_{dd}, and a N-FET tied the output to ground (0 V). These could be connected with some additional circuitry to power MOSFETs, allowing for higher power loads to be driven.

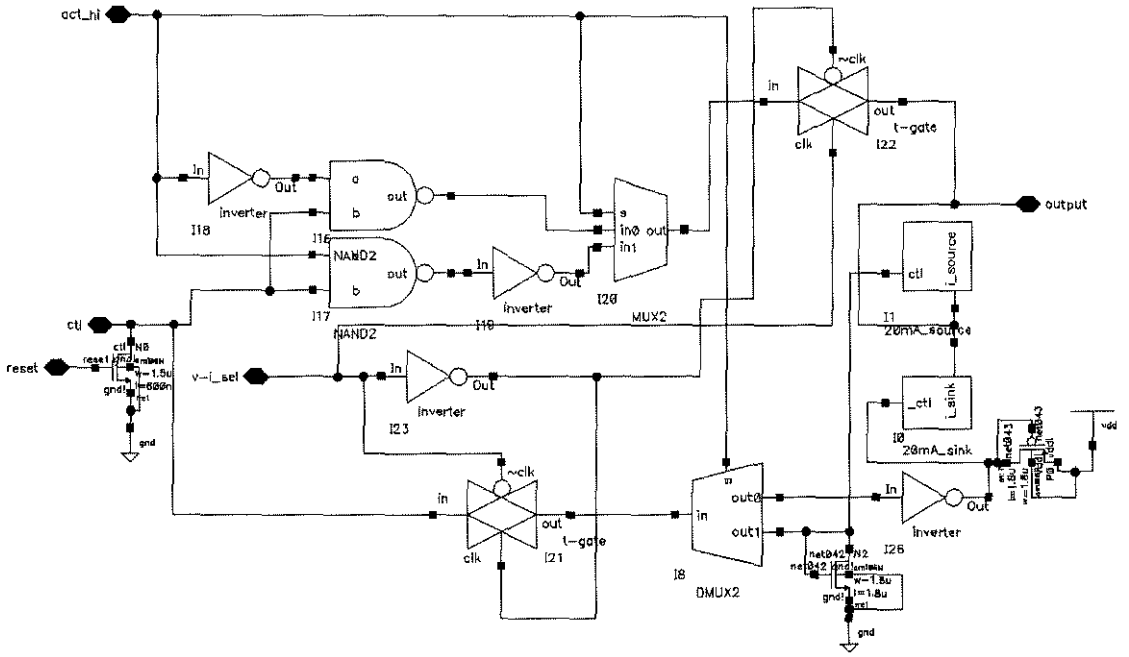

Figure 23. Schematic Diagram of VLSI Implementation of a Single Power Switching Module

The output stage could also act as a current source or sink. The current source was designed to power a 20 mA load. This design utilized a current reference, which was then mirrored and scaled to the output FETs. The current driving output was meant for operating loads where the absence or presence of current was the primary factor that affected the load's operation, such as a LED-based light.

The schematic of the Power Switching Module that is included as Figure 23 shows the control logic and current sources replaced by black boxes. The discrete transistors that are included to control the voltage sources were sized for minimal current capacity because they were designed to drive gates of further stages. The control logic contained a few parts that managed the data flow through the module. T-Gates were used to control whether the input control signal passed to the voltage source or the current source modules. The rest of the logic elements were used for the active high-selection. The NAND gates are for the voltage sources, and do the active high and active low conversions. The demultiplexer passed the control signal to the appropriate current source or sink module in current mode.

When the layout was simulated, all of the outputs operated correctly. The voltage sources provided 4.999 V with 371 μA of current when passing V_{dd}. When passing GND, the voltage source provided 187 nV with 244 μA of current.

The current sources did not achieve exactly 20 mA of current. The W/L ratios of the sources were computed based on estimates of the carrier mobility to achieve this value, but the current source was able to achieve 16.6 mA of current through a 1 Ω resistor to ground. The current sink was capable of sinking 19.01 mA of current through a 1 Ω

resistor to the digital voltage source. The inequality in source and sink capability is due to the inequality in the carrier mobility of P-FETs and N-FETs.

4.5. System Analysis

Once each of the individual blocks was designed and simulated, the functional blocks were integrated. Only one problem arose during simulation of the integrated system. The delay line for the counter in the Data Receiver needed to be lengthened, because the same timing issues that became apparent in the individual simulation of the Data Receiver again appeared. The timing issues that appeared in the integrated system simulation were most likely due to parasitic capacitances that the integrated system adds. A final schematic of the integrated system is provided to show the interconnection between functional blocks in Figure 24.

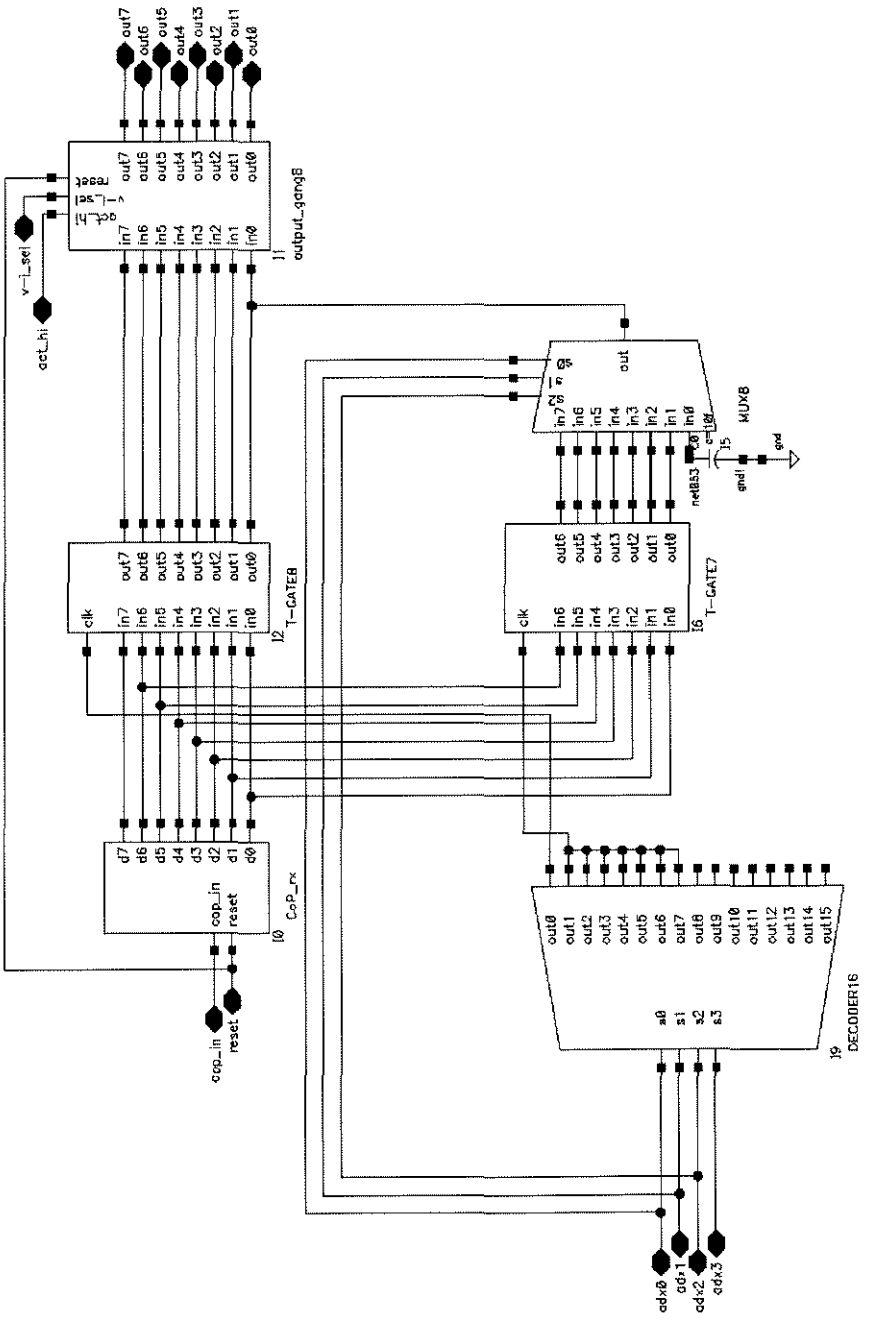

Figure 24. Full Schematic of VLSI Implementation

In converting the schematic to a prototype integrated circuit, several diagnostic signals were connected to pins for testing purposes. In addition to the standard inputs and outputs of the system, the intermediate data values of the Function Decoder connected to pins to be used as an input to the Function Decoder or an output of the Data Receiver. The ring oscillator was also pinned out to verify its correct operation. Other than standard layout techniques to integrate the modules together, only one other consideration was made. A great deal of the free space on the chip was converted into capacitors between V_{dd} and GND to help alleviate any noise issues on the power supply lines. A view of the complete layout of the chip is made available in Figure 25. The five large blocks in the lower left-hand corner of the chip are the power supply capacitors. To the right of the capacitor that is horizontal on the chip is the Data Receiver. On the right side of the chip near the pad frame are the Power Switching modules. The center block of circuitry between the Data Receiver and the Power Switching Modules is the Function Decoder circuitry and the interconnections between all of the modules.

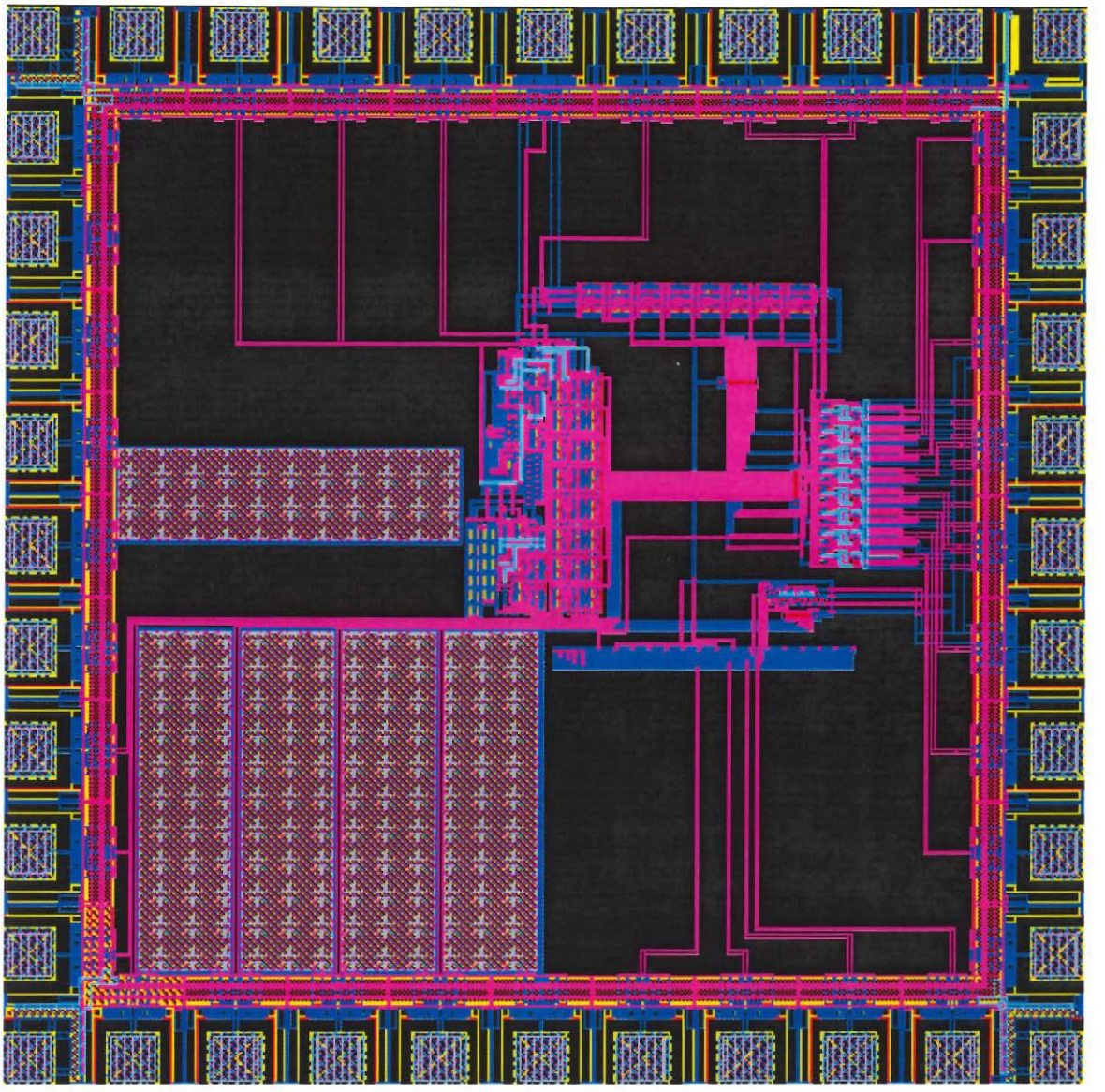

Figure 25. Layout of Complete Integrated Circuit that Was Fabricated

When looking at the failure modes that were included with the VLSI implementation of the Distributed Control System, the chip itself did not provide any further fault tolerance than the COTS solution. The looping that was accounted for in the transmitter was still

present with this implementation. It was reasonable to think that the proper design of off-chip circuitry, such as a fuse, should protect the chip from short-circuit conditions.

The complete layout was simulated under various function addresses and output stage configurations, and all configurations simulated correctly. To verify correct integration, when the complete layout was simulated in the 7-Way emulation mode, the voltages of the outputs of the chip matched the voltages expected based on the simulation of the Power Switching circuitry. Also, as shown in Figure 26, the sweeping turn signal function operated correctly while driving current sources.

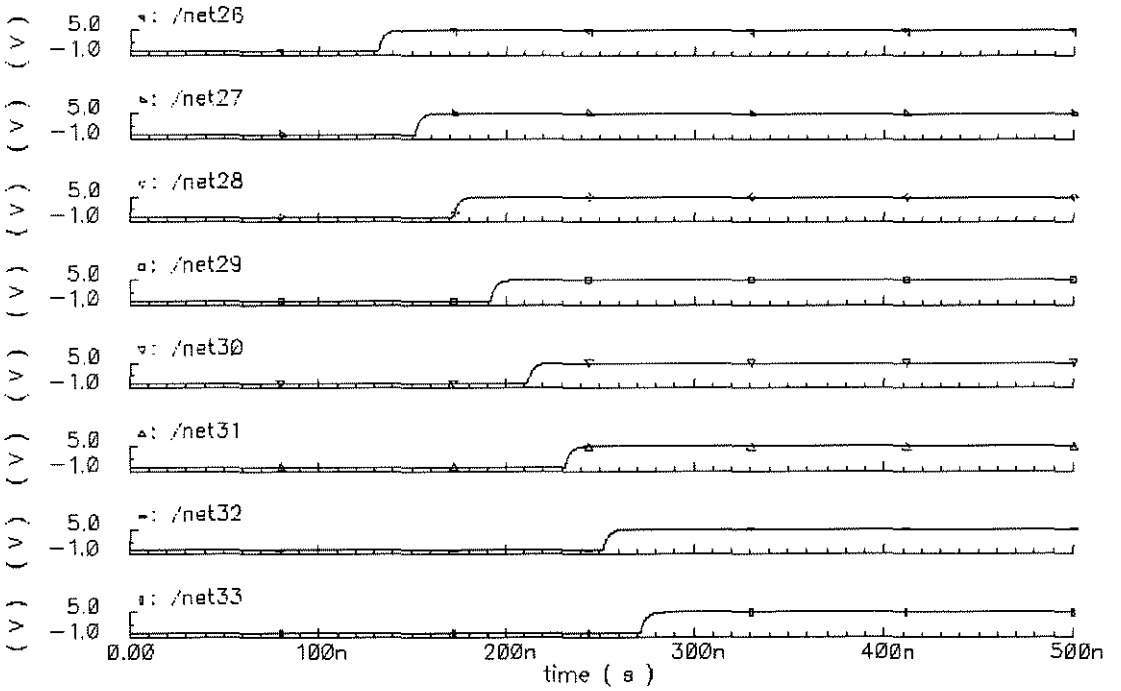

Figure 26. Simulated Output of Integrated CIrcuit in Sweeping Turn Signal Mode

4.6. Testing of Integrated Circuit Implementation

Upon testing the custom integrated circuit that was designed and fabricated for this thesis, a majority of the chip was shown to operate correctly. It was expected that the voltages and currents did not exactly match the simulated results. Table 5 shows the voltages and currents at the output of the chip from the Power Switching Module. The table shows the voltage conditions as measured at the output of the chip and the current that was either sourced or sunk by the Power Switching module.

Table 5. Measured Voltage and Current Outputs of Custom IC

Mode	*Voltage at Output*	*Current*
Voltage (High output)	4.996 V	0.280mA
Voltage (Low output)	0.965 V	6.8 mA
Current Source	0.440 V	13.208mA
Current Sink	4.63 V	17.069mA

The function decoder worked correctly in the 7-Way and Single Emulation modes, but did not work in the Sweeping Turn Signal mode. It was determined later that the Sweeping Turn Signal mode did not work because a wire had become disconnected in the final layout of the chip. This error must have occurred after the final simulations of the entire chip. To test the 7-Way and Single Channel emulations, control voltages were put to the pins that were intended to view the output of the Data Receiver, and consequently the input of the Function Decoder. By putting a specific data byte into the

Function Decoder, it was possible observe and measure the output. The actual voltages and currents at the output of the chip have been shown in Table 5 above.

The testing of all the input control signals proved successful. A full-chip reset was performed, the function address was changed, and selection between current and voltage modes and switching between active high and active low modes was tested. These signals comprise all the setup controls signals. The next signal that was tested was the output pin of the ring oscillator, which also worked correctly. Upon a falling edge of the input data signal, the oscillator started, and produced nine clock pulses before returning to its idle high state. The only other control signal, which was an internal control signal that was connected to a pin on the chip, was the d_ready signal. This signal is used to latch data to the output buffer of the Data Receiver. Unfortunately, this signal did not work like it did in the simulations. There was a visible pulse that could be viewed on an oscilloscope, but the peak was about 1 V. Since the voltage was not reaching 5 V at the output, it was most likely because enough current could not be sourced to drive the pad 'high' in the short time allotted for the signal to be pulsed. The cause for the problem with the d_ready signal was probably due to the gates not being wide enough. Since the pulse was being seen, it could be deduced that the gates were being triggered. The Data Receiver as a whole could not be completely verified, due to the fact that the d_ready line was not able to latch data from the shift register to the output buffer.

It should be mentioned that simulation results showed the bit clock to be approximately 70 MHz, and the measured bit clock on the chip was approximately 5 MHz. This variation was due to added parasitic capacitances and variation of the process parameters from those that were simulated. With such a variety in clock rates, it was necessary to discuss some issues of bit error rate. Due to the fact that the receiver had absolutely no error checking built into it, the system was going to be very susceptible to incorrect bit timings on the data. Although a microcontroller may have a fairly low susceptibility to slight variations in bit clock because intelligence can be built in to detect these errors; this particular chip had no way to tell an error from good data.

Given all of the data that were collected, from implementation and simulation, the chip could work properly, if the few errors were fixed. Based on simulation data, the Data Receiver was able to receive an asynchronous data signal that conformed to the SCI protocol. Given that the data were received, the Function Decoder was capable of translating the input data byte into the control signals required by the Power Switching module. The final outputs of the chip were also within expected limits. If the d_ready signal operated to the levels that it did in simulation, then the chip as a whole would have worked. Therefore, a custom VLSI implementation, utilized to minimize the final design, could be developed, once all of the specific functionalities were determined.

5. Comparison of COTS versus VLSI Implementations

To make a valid comparison of the implementations, both the COTS and VLSI implementations needed to be able to provide the same functionality. Both receivers were wired up with power switching circuitry for high-power loads.

5.1. Final Implementation Sizes

When comparing the implementations of the receiver systems, the implementation sizes were fundamentally the same. Although the VLSI implementation was in an integrated circuit package that was physically larger, the larger package was not required. Based upon this fact, a smaller package, similar to the package of the PIC micro controller, could be used in commercial implementations. When it came to commercial implementations, all of the parts could be further minimized to the point that the limiting factor will quickly become the high-current power switching circuitry. The final size of both prototype implementations was a 3" by 1.5" circuit board.

5.2. Final Implementation Specifications

When comparing the final output specifications of both the COTS and the VLSI implementations of the system, I could see that both systems were very similar in capability. If you leave off the high-current power switching circuitry, which was

required in either case, the output voltages were both effectively the same. In looking at the current drive capabilities, it was determined previously that the VLSI implementation can source and sink approximately 16 mA. The integrated circuit could be redesigned to increase its current drive capabilities. According to the PIC microcontroller data sheet, the chip can source or sink 25 mA per pin, with a maximum of 200 mA for combined ports [8]. So, the microcontroller had a little bit more capability than this chip implementation in driving current based loads, but careful design must be done to make sure that the per port current limit was not exceeded. A difference between the microcontroller and the custom IC was that the custom IC was designed to saturate the current outputs, whereas the microcontroller could exceed the maximum rating and destroy itself.

As for input requirements, both systems were based on 5 V CMOS logic and, therefore, had the same voltage input requirements. The only difference in the two systems was that the VLSI implementation required less current in a no-load condition when the only thing being powered was control logic.

5.3. Final Implementation Flexibility

In looking at the flexibility of the designs, the two implementations were the opposites of each other. The VLSI implementation had a fixed implementation that was on the

silicon. To make a VLSI implementation with a different set of functions required a complete redesign and the relayout of the chip.

The microcontroller, on the other hand, was reprogrammable. The reprogrammability allowed the functions to be changed by downloading a new program to the micro controller. Through the design of the receiver board, the same circuit board implementation of the COTS system provided different functionalities with only a change in the microcontroller code. The microcontroller could be programmed when the receiver was installed into a trailer after the functionality of the board had been decided, further adding to the flexibility of the system.

Another flexibility issue came in the form of the number of functions that could be implemented. If the custom chip were built on the same process and die size as the microcontroller, then the integrated circuit could host more separate functions simultaneously than the microcontroller could. This was due to the fact that the specialized circuitry on the custom chip would be able to be packed tighter than the circuitry on the microcontroller. With the fact that the microcontroller is reprogrammable, then the number of functions that could be implemented on the microcontroller is limited only by the amount of memory available for the program.

5.4. Final Implementation Costs

Another factor considered when comparing the implementations was the economic factor. The cost of the COTS implementation that was presented in this thesis is included in Table 6. Because academic granting funded the development of the VLSI implementation, both implementations were approximately the same in price since they utilized approximately the same supporting circuitry (cost difference would be approximately one dollar). Given a nonacademic situation where the integrated circuit prototyping would have to be paid for, the cost of the VLSI implementation becomes quite large. For example, a 2.2 mm x 2.2 mm die would cost $187.50 for each chip in a quantity of 40 chips through MOSIS [9]. The cost of the supporting circuitry becomes negligible against the cost of the custom chip.

Table 6. Single Unit Costing of COTS Board [3]

Part	*Quantity*	*Cost (ea)*	*Cost (total)*
PIC16F873 Micro controller	1	$7.63	$7.63
16MHz Crystal	1	$0.73	$0.73
18pF Capacitors	2	$0.07	$0.15
1kΩ Resistor	2	$0.11	$0.22
LM340T-5.0	1	$0.90	$0.98
1 μF Capacitor	1	$0.42	$0.42
IRF6215 PFET	1	$2.09	$2.09
2N7000 NFET	1	$0.31	$0.31
Manufacturing	1	$10.00	$10.00
	COTS Total		$22.53

In a manufacturing scheme, where economies of scale take place, the costs were quite different. Extrapolating the cost of the COTS solution for a quantity of 100 units is shown in Table 7. The price for the actual manufacturing of the board was assumed to be around $1 per board. Implementing a One-Time Programmable microcontroller could generate additional savings. This removed the reprogrammability of the microcontroller, but reduced the cost from $7.63 to $7.23 per unit for the microcontroller [5].

Table 7. Cost Estimates for Unit Production in Quantities of 100 for COTS Implementation

Part	*Quantity*	*Cost (ea)*	*Cost (total)*
PIC16F873 Micro controller	1	$4.95	$4.95
16MHz Crystal	1	$0.52	$0.52
18pF Capacitors	2	$0.06	$0.12
1kΩ Resistor	2	$0.05	$0.10
LM340T-5.0	1	$0.47	$0.47
1 μF Capacitor	1	$0.29	$0.29
IRF6215 PFET	1	$1.04	$1.04
2N7000 NFET	1	$0.16	$0.16
Manufacturing	1	$3.02	$3.02
	COTS Total		$10.67

The VLSI implementation ended up being very inexpensive. In large enough economies of scale, the cost of the custom chip should be able to be reduced to approximately $1. This low of a unit pricing would only be available in the case where dedicated high volume, or complete wafer, runs were done. The volume costing of the VLSI implementation is also presented in Table 8. Utilizing the same per board assumption for manufacturing cost, I saw that the VLSI implementation was cheaper for quantity volumes.

Table 8. VLSI Implementation Cost Estimates for Quantities of 100

Part	*Quantity*	*Cost (ea)*	*Cost (total)*
Custom Integrated Circuit	1	$1.00	$1.00
1kHz Oscillator	1	$0.52	$0.52
1kΩ Resistor	1	$0.05	$0.05
LM340T-5.0	1	$0.47	$0.47
1 μF Capacitor	1	$0.29	$0.29
IRF6215 PFET	1	$1.04	$1.04
2N7000 NFET	1	$0.16	$0.16
Manufacturing	1	$3.02	$3.02
	VLSI Total		$6.55

6. Conclusions and Further Research

Through this thesis, it was shown that a Distributed Control System could be implemented that integrated communications with power distribution. Two of the many methods of implementing this system were investigated. Both implementations functioned with the same fundamental specifications but have a few differences.

The final implementation sizes, while not completely minimized, were the same for either implementation. One advantage that was gained with the VLSI implementation in terms of size was the reduced part count of the final circuit. By reducing the final part count, the receivers will be easier to manufacture. With the ability of testing a majority of the circuitry before it is even packaged, a higher yield could be expected. Also, the reduced number of interconnecting wires between discrete parts reduces the number of interconnections that can get damaged through shock or vibration damage. The primary advantage that was gained from the microcontroller implementation was the amount of flexibility that was available to the designer.

Furthering the research presented here can take various forms. The VLSI implementation is the system that requires the greatest amount of further development. Currently, the Data Receiver implements no error checking logic; adding error checking in a future revision will significantly increase the reliability of the system. More functions can be

designed and implemented on the chip as well. The receiver can further be enhanced in future revisions by modifying the system that samples the input data signal. By using an oversampled clock, the system will be less sensitive to variation in the bit clock. Decreasing the sensitivity to variation in the bit clock will also make the communications more reliable.

In addition to increasing the functionality of the existing systems, different methods of data coupling and decoupling can be investigated. The next logical step in the communications coupling is to look at inductive coupling instead of capacitive coupling. One of the advantages of the inductive coupling would be an increase in transmitter power. Currently the FETs and voltage regulators on the transmitter limit the system power. Should the system move to an inductive coupling methodology, the transformer that would be used would limit the power. Different types of transformers and cores can be optimized to allow the maximum amount of power to be delivered. At the receiver, a solution for an on-chip transformer can be investigated to decouple the communications in as small a size as possible. Although integrated inductors in the past have been discouraged, new research is being done to place small inductors on a chip [1].

The last area of further development would improve fault tolerance. With the concept of the looping, the receiver should be able to detect the direction of the current flow through the loop. The receiver can, therefore, be designed to detect when current is not flowing,

and then can alert the operator in some manner to show that there is a fault in the system. Finally, the system can be expanded to allow devices other than semi-trailer lights to be powered.

LIST OF REFERENCES

LIST OF REFERENCES

1. Burghartz, J.N., et al. "Monolithic Spiral Inductors Fabricated Using a VLSI Cu-damascene Interconnect Technology and Low-Loss Substrates." *Electronic Devices Meeting* (1996): 99-102.

2. Digi-Key, Inc. *Digi-Key Catalog T031.* Thief River Falls, Minn.: Digi-Key, Jan 2003.

3. Hastings, Alan. *The Art of Analog Layout.* Upper Saddle River, N.J.: Prentice-Hall, Inc., 2001.

4. Helmuth, J. B. Telephone call, 15 April, 2003.

5. International Organization for Standardization. *Road Vehicles – Interchange of digital information – Controller area network (CAN) for high-speed communication.* ISO 11898. Geneva, Switzerland: ISO, 1993.

6. Katz, Randy H. *Contemporary Logic Design.* Redwood City, CA: Benjamin/Cummings Publishing Company, Inc., 1994.

7. IEEE Computer Society LAN/MAN Standards Committee. *Data Terminal Equipment (DTE) Power via Media Dependent Interface (MDI).* Draft supplement to standard IEEE 802.3. New York: IEEE, 2002.

8. Microchip Technology. PIC 16F87x Data Sheet. Chandler, Ariz.: Microchip Technology, 2001. http://www.microchip.com/download/lit/pline/picmicro/families/16f87x/30292c.pdf (7 July 2002).

9. MOSIS Service. Mosis Price List -- Domestic. Marina Del Rey, Calif.: http://www.mosis.org/Orders/Prices/price-list-domestic.html (17 April 2003).

10. Newbury, John. "Communication Requirements and Standards for Low Voltage Mains Signalling." *IEEE Transaction on Power Delivery* **13** (1998): 46-52.

11. Newbury, John E., and Kerry J. Morris. "Power Line Carrier Systems for Industrial Control Applications." *IEEE Transactions on Power Delivery* **14** (1999): 1191-1196.

12. Patrick, Adrian, John Newbury, and Sean Gargan. "Two-Way Communications System in the Electricity Supply Industry." *IEEE Transactions on Power Delivery* **13** (1998): 53-558.

13. Society of Automotive Engineers. *Power Line Carrier Communications in Commercial Vehicles*. SAE J2497. Warrendale, Penn.: SAE International, 2002.

14. Society of Automotive Engineers. *Serial Data Communications between Microcomputer Systems in Heavy-Duty Vehicle Applications*. SAE J1708. Warrendale, Penn.: SAE International, 1993.

15. Sze, S. M. *Physics of Semiconductor Devices*. New York: Wiley-Interscience, 1969.

16. Uyemura, John P. *Circuit Design for CMOS VLSI*. Boston: Kluwer Academic Publishers, 1992.

17. Weste, Neil H. E., and Kamran Eshraghian. *Principles of CMOS VLSI Design, 2nd Ed.* Reading, Mass.: Addison-Wesley Publishing Company, 1993.

APPENDICES

Appendix A: COTS Implementation Transmitter Microcontroller Code

```
#include <pic.h>
#include "sci.h"
#include "delay.h"
#include "sundial.h"

/*#define HIGH_SPEED*/

void main (void) {
  unsigned char OldSevenWay = 128;
  int count;

          TRISB = 255;
          sci_Init(SUNDIAL_BAUD, SCI_EIGHT);
#ifdef HIGH_SPEED
          BRGH = 1;
          SPBRG = 0;
#endif
          count = 20000;
          while (1 == 1) {
                    if (OldSevenWay != PORTB) {
                              sci_PutByte(PORTB);
                              OldSevenWay = PORTB;
                              count = 0;
                    } else if (count > 10000) {
                              sci_PutByte(OldSevenWay);
                              count = 0;
                    }
                    count++;
                    DelayMs(1);
          }
}
```

Appendix B: COTS Implementation Receiver Microcontroller Code

```
#include <pic.h>
#include "sci.h"
#include "delay.h"
#include "sundial.h"

/*  State Definitions:
0 - Direct 7-Way Rx
1 - Respond to Line 1 of 7-Way
2 - Respond to Line 2 of 7-Way
3 - Respond to Line 3 of 7-Way
4 - Respond to Line 4 of 7-Way
5 - Respond to Line 5 of 7-Way
6 - Respond to Line 6 of 7-Way
7 - Respond to Line 7 of 7-Way
8 - Headlight Control (Line 8 from 7-Way Tx)
255 - Unaddressed
*/

/*#define HIGH_SPEED*/

char state = 255;

void interrupt isr(void) {
  unsigned char ReceivedByte;
  unsigned char temp[8];

          if(RCIF)
          {
                    ReceivedByte = sci_GetByte();
                    if (state == 0) {
                              PORTB = ReceivedByte;
                    } else if (state == 255) {
                              ;
                    } else {
                              temp[0] = (ReceivedByte & 1);
                              temp[1] = (ReceivedByte & 2) >> 1;
                              temp[2] = (ReceivedByte & 4) >> 2;
                              temp[3] = (ReceivedByte & 8) >> 3;
                              temp[4] = (ReceivedByte & 16) >> 4;
                              temp[5] = (ReceivedByte & 32) >> 5;
                              temp[6] = (ReceivedByte & 64) >> 6;
                              temp[7] = (ReceivedByte & 128) >> 7;
                              if (temp[state - 1] == 1) {
                                        PORTB = 255;
                              } else {
```

```
                                        PORTB = 0;
                                }
                        }
                        sci_CheckOERR();
                }
}

void main (void) {

                TRISB = 0;
                ADCON1 = 6;
                TRISA = 63;
                PORTB = 0;
                PORTC = 192;
                DelayMs(10);
                state = PORTA;
                state = state & 63;
                sci_Init(SUNDIAL_BAUD, SCI_EIGHT);
#ifdef HIGH_SPEED
                BRGH = 1;
                SPBRG = 0;
#endif
                PIR1 = 0;  /* clear any pending interrupts */
                PEIE = 1;  /* enable perhipheral interrupts */
                GIE  = 1;  /* global interrupts enabled */

                /* perform other setup */

                RCIE = 1;  /* unmask receiver interrupts... */
                while (1==1) {
                }

}
```

Appendix C: Receiver Code for Sweeping Turn Signal Functionality

```
#include <pic.h>
#include "sci.h"
#include "delay.h"
#include "sundial.h"

#define MARK_MASK 170
#define TURN_MASK 85
#define DELAY 2500
#define DEBUG

unsigned char temp;
char state = 0;

void RunTurn () {
  int i;
  char j;
  char k;
          if (temp == 2 & state == 0) {
                    PORTB = 1;
                    i = 0;
                    while (i < DELAY) {
                              i ++;
                    }
                    PORTB = 3;
                    i = 0;
                    while (i < DELAY) {
                              i ++;
                    }
                    PORTB = 7;
                    i = 0;
                    while (i < DELAY) {
                              i ++;
                    }
                    RA2 = 0;
                    PORTA = 1;
                    i = 0;
                    while (i < DELAY) {
                              i ++;
                    }
                    PORTA = 3;
                    i = 0;
                    while (i < DELAY) {
                              i ++;
                    }
                    PORTA = 15;
```

```
                    i = 0;
                    while (i < DELAY) {
                              i ++;
                    }
                    PORTC = 1;
                    i = 0;
                    while (i < DELAY) {
                              i ++;
                    }
                    PORTC = 3;
                    i = 0;
                    while (i < DELAY) {
                              i ++;
                    }
                    PORTC = 7;
                    state = 1;
          } else  if ((temp != 2) & state == 1) {
                    PORTA = 0;
                    PORTB = 0;
                    PORTC = 0;
                    state = 0;
          }
}

void interrupt isr(void)
{
          PEIE = 0;  /* disable perhipheral interrupts */
          if(RCIF)
          {
                    temp = sci_GetByte();
                    sci_CheckOERR();
          }
          PEIE = 1;  /* enable perhipheral interrupts */
}

void main() {
  int i;

          TRISB = 0;
          TRISC = 192;
          ADCON1 = 6;
          TRISA = 0;
          PORTA = 0;
          PORTB = 0;
          PORTC = 0;
          sci_Init(SUNDIAL_BAUD, SCI_EIGHT);
          PIR1 = 0;  /* clear any pending interrupts */
          PEIE = 1;  /* enable perhipheral interrupts */
          GIE  = 1;  /* global interrupts enabled */

          /* perform other setup */
```

```
            RCIE = 1;  /* unmask receiver interrupts... */

            while (1 == 1) {
#ifndef DEBUG1
                        RunTurn();
#endif
/*                      SunDial_CheckOERR();*/
            }
}
```