To the Graduate Council:

I am submitting herewith a thesis written by Vinton Lee James IV entitled "Functional Verification and Power Analysis of an Analog I/O Interface for the IEEE 1394-1995 High-Speed Serial Bus." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science with a major in Electrical Engineering.

Daniel B. Koch

Dr. Daniel B. Koch, Major Professor

We have read this thesis
and recommend its acceptance:

Donald W. Bouldin

M. J. Ro

Accepted for the Council:

Lee Minkel

Associate Vice Chancellor and
Dean of The Graduate School

Functional Verification and Power Analysis of an Analog I/O Interface for the IEEE 1394-1995 High-Speed Serial Bus

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Vinton Lee James IV

December 1998

ACKNOWLEDGMENTS

I wish to express my sincere appreciation to Dr. Daniel B. Koch for guiding me through the thesis process. His positive attitude is inspiring. I also wish to thank Dr. Koch for advising me during the past six years. As a distance student his support helped me complete the masters program. My appreciation also goes to Dr. Don Bouldin and Dr. Michael J. Roberts for serving on my thesis committee. My gratitude also goes to Dr. Marshall O. Pace and Dr. Chad Limbaugh for their support, flexibility, and efforts in the distance learning program. Their understanding of the difficulty of distance learning was of great value to me.

I am greatly indebted to Bill Meadows and Paul Hollis for their technical support of the thesis. As the principal designers of the circuit simulated in the thesis, their inputs and assistance was of great importance.

I take pride in the accomplishment of this thesis and the Master of Science in Electrical Engineering. I take greater pride to be a graduate of as school that is as widely know as the University of Tennessee.

ABSTRACT

This thesis simulates the functionality and performs a power analysis of the analog I/O portion of a Physical Layer device. The analog I/O circuit has been designed to comply with IEEE 1394-1995 specifications for Physical Layer devices. Applications for embedded microprocessor systems are increasing at a high rate. Communication between embedded systems processors and other systems is accomplished either with a parallel interface or a slow-speed serial interface. The IEEE 1394-1995 specification defines a high-speed, low-cost, serial bus. This bus allows efficient solutions to construct links to other nodes, with fewer wires and smaller cables, with less radio frequency interference where each link is terminated. It also has less cost than parallel interfaces. Data rates for the IEEE 1394 serial bus are 100Mbs, 200Mbs, and 400Mbs. This interface can be used to link peripherals such as a Personal Computer (PC) to a CD ROM, digital camera, hard disk drive (HDD), scanner, printer, or multi-media device, or for any time critical application. The background of the IEEE 1394-1995 high-speed serial bus specification is examined. The IEEE 1394-1995 specifications are then rigorously investigated. Various simulations are executed using various process fabrication scenarios and operating conditions. The simulations are designed for either subcircuits of the entire circuit or for port-to-port communication. A power analysis is also performed for the entire circuit. Discussion of the results and further research possibilities completes the thesis.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION.....1
1.1 The Need for High-Speed Serial Busses.....1
1.2 Current Parallel and Serial Busses.....3
1.3 High-Speed Peripheral Applications.....4
1.4 Thesis Problem Statement.....4
1.5 Thesis Outline.....5

CHAPTER 2. BACKGROUND.....7
2.1 Scope.....7
2.1.1 System Overview.....7
2.1.2 Transaction Layer Overview.....9
2.1.3 Link Layer Overview.....9
2.1.4 Physical Layer Overview.....10
2.1.5 Analog I/O Overview.....15
2.1.6 Data Coding.....18
2.2 Media Signal Interface Specifications.....23
2.2.1 Signal Amplitude.....23
2.2.2 Common Mode Voltage.....25
2.2.3 Speed Signaling.....27
2.2.4 Arbitration Signal Voltages.....27
2.2.5 Input Impedance.....29
2.2.6 Noise.....29
2.2.7 Data Rate.....30
2.2.8 Data Signal Rise and Fall Times.....30
2.2.9 Jitter and Skew.....30
2.3 Physical Devices Available.....31
2.4 Physical Devices Implementation.....32
2.5 Physical Devices Specifications.....33

CHAPTER 3. PORT INTERFACE MEDIA CABLE MODEL.....38
3.1 Cable Media Electrical Specifications.....38
3.2 Spice Model Theory.....39
3.3 Method of Implementation in Simulation.....41
3.4 Simulation and Conformation of Media Cable Model.....41

CHAPTER 4. SUBCIRCUIT SIMULATION AND ANALYSIS.....44
4.1 Arbitration Receiver Circuit.....47
4.1.1 Test Case Description.....49
4.1.2 Process and Operating Parameter Simulation Matrix.....50
4.2 Data and Strobe Transmit Driver.....53
4.2.1 Test Case Description.....56
4.2.2 Process and Operating Parameter Simulation Matrix.....57
4.3 Data and Strobe Receiver.....60
4.3.1 Test Case Description.....62
4.3.2 Process and Operating Parameter Simulation Matrix.....63

CHAPTER 5. PHYSICAL PORT MIXED MODE SIMULATION AND POWER ANALYSIS.....66
5.1 Data and Strobe Transmission and Reception.....66
5.1.1 Data and Strobe Transmission and Reception Test Case Description.....68
5.1.2 Data and Strobe Process and Operating Parameter Simulation Matrix.....69
5.2 Arbitration.....75
5.2.1 Arbitration Test Case Description.....75
5.2.2 Arbitration Process and Operating Parameter Simulation Matrix.....76
5.3 Speed Sensing.....80
5.3.1 Speed Sensing Test Case Description.....80
5.3.2 Speed Sensing Process and Operating Parameter Simulation Matrix.....81
5.4 Port Status.....84
5.4.1 Port Status Test Case Description.....86
5.4.2 Port Status Process and Operating Parameter Simulation Matrix.....86
5.5 Power Analysis.....87
5.5.1 Power Dissipation Test Case Description.....89
5.5.2 Power Dissipation Process and Operating Parameter Matrix.....91

CHAPTER 6. CONCLUSIONS.....98
6.1 Discussion.....98
6.2 Results.....99
6.3 Opportunities for Further Research.....100
6.4 Conclusion.....101

LIST OF REFERENCES.....102

APPENDIX.....104

VITA.....150

LIST OF TABLES

Table 2.1 Transaction Layer Operations Summary.....9
Table 2.2 Summary of Cable Physical Layer Services.....11
Table 2.3 Analog Circuit Pin Description.....19
Table 2.4 Differential Output Signal Amplitude.....25
Table 2.5 Differential Receive Signal Amplitude.....25
Table 2.6 Common Mode Voltage Levels.....26
Table 2.7 Port Connection Status Signal.....26
Table 2.8 Common Mode Current Values.....27
Table 2.9 Arbitration Decoding Rules.....29
Table 2.10 Arbitration Signaling Levels.....29
Table 2.11 Differential Input Impedance Specifications.....30
Table 2.12 Cable Media Data Rates.....30
Table 2.13 Output Rise Times, Fall Times, and Input Slope.....31
Table 2.14 Jitter and Skew.....31
Table 2.15 Signal Amplitude.....33
Table 2.16 Speed Signaling Common Mode Input Voltage (TPA).....34
Table 2.17 Speed Signaling Common Mode Input Voltage (TPB).....36
Table 2.18 Common Mode Speed Signaling Current.....36
Table 2.19 Arbitration Input Signal Voltages.....36
Table 2.20 S400 Maximum Jitter, Skew, Rise, Fall Times.....37
Table 3.1 Cable Characteristics.....38
Table 4.1 Simulation Matrix.....45
Table 4.2 Operating Condition Values.....45
Table 4.3 Parasitic Model Parameters.....46
Table 4.4 Transistor Model Parameters.....47
Table 4.1.1 Arbitration Truth Table.....49
Table 4.1.2 Arbitration Test Conditions.....51
Table 5.1.1 Data and Strobe Enable and Control Signals.....68
Table 5.2.1 Arbitration Enable and Control Signals.....76
Table 5.3.1 Speed Signaling Enable and Control Signals.....81
Table 5.3.2 Port Status Enable and Control Signals.....85
Table 5.5.1 Functional Power Dissipation Matrix.....90
Table 5.5.2 Minimum and Maximum Power Dissipation Matrix.....91
Table 5.5.3 Minimum and Maximum Power Dissipation Inputs.....92
Table 5.5.4 Current Dissipation Summary.....96
Table 5.5.5 Power Dissipation Summary.....97
Table 5.5.6 Minimum and Maximum Current Dissipation Simulation.....97

LIST OF FIGURES

Figure 2.1 IEEE 1394 Protocol Layers....8
Figure 2.2 Typical Physical Layer Device....13
Figure 2.3 Cable Media Interface....14
Figure 2.4 Single Analog I/O Port....16
Figure 2.5 Single Physical Device Port....17
Figure 2.6 TPA Schematic....20
Figure 2.7 TPB Schematic....21
Figure 2.8 Differential Encoding....22
Figure 2.9 Non-Return-To-Zero Format....22
Figure 2.10 Recovered Clock Signal....24
Figure 2.11 Three Level Signaling....28
Figure 3.1 Spice Model of Differential Pair Transmission Lines....39
Figure 3.2 Differential Mode Connector And Cable Spice Model....40
Figure 3.3 Common Mode Connector And Cable Spice Model....40
Figure 3.4 Connector And Cable Spice Model Schematic....42
Figure 3.5 Lump Elementary Cell Structures....42
Figure 3.6 Data Transmission And Reception Cable Simulation....43
Figure 4.1.1 Arbitration Receiver Circuit Block Diagram....48
Figure 4.1.2 Spice Arbitration Input Voltages....52
Figure 4.1.3 Spice Arbitration Simulation Results....52
Figure 4.1.4 Spice Arbitration Simulation With Low V_{dd} 54
Figure 4.2.1 Data / Strobe Driver Circuit Block Diagram....55
Figure 4.2.2 Spice Driver Circuit Input Voltages....55
Figure 4.2.3 Spice Data / Strobe Driver Simulation Results....58
Figure 4.2.4 Data / Strobe Driver Simulation With Low V_{dd} , High Temp....58
Figure 4.2.5 Spice Data / Strobe Driver Rising Edge Simulation....59
Figure 4.3.1 Data And Strobe Receiver Circuit Block Diagram....61
Figure 4.3.2 Spice Data And Strobe Receiver Input Voltages....64
Figure 4.3.3 Spice Data And Strobe Receiver Simulation Results....64
Figure 4.3.4 Spice Data And Strobe Receiver Simulation....65
Figure 5.1 Port-To-Port Simulation Model....67
Figure 5.1.1 Spice Data/Strobe Driver Circuit Input Signals....70
Figure 5.1.2 Data And Strobe Input and Output Signals....70
Figure 5.1.3 Data And Strobe Signal Propagation....72
Figure 5.1.4 Analog Data / Strobe Signals....72
Figure 5.1.5 Worst Case Data And Strobe Transmission And Reception....74
Figure 5.1.6 Worst Case Analog Data / Strobe Signals....74
Figure 5.2.1 Arbitration Simulation Input....77
Figure 5.2.2 Arbitration Simulation With Cable Model....78
Figure 5.2.3 Arbitration Simulation Without Cable Model....79
Figure 5.2.4 Arbitration Simulation Output Without Cable Model....79
Figure 5.3.1 Speed Signaling Simulation Input....82
Figure 5.3.2 Speed Signaling Simulation Input And Analog Modulation....83

Figure 5.3.3 Speed Signaling Analog Modulation And
Digital Outputs.....83
Figure 5.3.4 Speed Signaling Sub-Circuit Current Verification.....85
Figure 5.4.1 Port Status With Connected Port.....88
Figure 5.4.2 Port Status With Disconnected Port.....89
Figure 5.5.1 Simulation Power Analysis Inputs.....92
Figure 5.5.2 Typical Case Simulation Power Analysis.....93
Figure 5.5.3 Simulation Current Dissipation Summary.....94
Figure 5.5.4 Simulation Current Dissipation Across Temperature.....95

CHAPTER 1. INTRODUCTION

1.1 The Need for High-Speed Serial Busses

Applications for embedded microprocessor systems are increasing at a high rate. Customer demands for higher performance, greater system integration, and lower cost are putting greater pressures on designers and manufactures of microprocessor-based systems.

Communication between embedded systems processors (ESP) and other systems is typically accomplished either with a parallel interface or a slow-speed serial interface. The parallel interface can have a high data rate, but problems such as timing skew, impedance, power usage from drivers, space considerations, cumbersome cables, costly media, and overall cost can be serious limitations. Standard serial interfaces work well, but currently have low data rates.

Apple Computer started development on its FireWire™ high-speed serial bus to improve system performance in the middle of the 1980s [1]. Soon afterwards work began on the IEEE 1394 specification, which defined a high-speed, low-cost, serial bus. The intention is to replace older serial communication technologies such as RS232 and improve the speed of data transmission between electronic peripheral devices. The current specification under development is the IEEE 1394-1995 specification.

One of the possible FireWire™ applications is for a high-speed peripheral bus. This configuration allows good in or out-of-cabinet solutions to link to other nodes using fewer wires, smaller cables, less radio frequency interference, and terminated links. It also has less cost than parallel interfaces. Data rates for the IEEE 1394-1995 serial

bus are 100Mbs (S100), 200Mbs (S200), and 400Mbs (S400). Future data rates could include 800Mbs and 1Gbs, but these may require fiber optic media.

The system is broken down into three main parts: Transaction Layer, Link Layer, and Physical Layer. The Transaction Layer provides bus management services for serial bus management, such as write, read and lock transactions [1]. It interfaces between the Link Layer and higher layers. The details of this interaction is outside the scope of the thesis. The Link Layer provides acknowledged data transfer services between a source node and a destination node. The Physical Layer provides the actual interface between the nodes, including both the mechanical and electrical interface. It includes the physical connection, which consists of the cable media itself, the connectors, and electronic transceivers.

The Physical Layer is mostly a digital circuit, but it also has an analog portion. The analog portion (transceiver, or port) is the interface between the digital I/O of the Physical Layer and the cable. Motorola is designing this port circuit as part of a module to be used in integrated ESP designs. This module will meet IEEE 1394-1995 standards in conjunction with digital Link and Physical Layer modules.

The purpose of the thesis is to verify the port (analog I/O) portion of the Physical Layer on the functional and electrical specification levels, and to use this data to perform a power analysis.

1.2 Current Parallel and Serial Busses

Most communication between microprocessors and peripherals is done on a parallel bus. For example, when a microprocessor services a parallel or serial port, it usually does so using an 8-bit bus with various control signals. A computer may utilize a 16-bit, 32-bit, 64-bit, or even larger busses for communication to devices in the system. These systems are usually implemented on a board-level design or by using several connected boards, but are also used in communication with other systems such as a printer using a cable. Although many parallel busses provide a high rate of data transfer at low individual bit transmission speeds, they also require much more physical area than serial busses. They are susceptible to timing skews, drive cost higher, require many drivers, and require special considerations to control impedance. Some designs also transmit levels of electro-magnetic interference (EMI) which violate state or federal regulations. The system-level bus has always been a bottle-neck for microprocessor-based systems.

Serial busses allow faster individual bit transmission speeds than parallel busses. They also offer reduced space and fewer wires. EMI can be reduced by using fewer signal lines or cheaper, shielded cables. Power consumption is reduced due to fewer bus drivers. A high-speed serial communication port which can transfer data faster than a parallel bus is ideal for improving performance while reducing cost of a microprocessor-based system. This need is the main driver for the IEEE 1394-1995 high-speed serial bus as current serial technologies such as RS232 have a maximum data transfer rate of 19.2kbs. Other serial

technologies such as Asynchronous Transfer Mode (ATM), HIgh Performance Peripheral Interface (HIPPI), Scaleable Coherent Interface/Local Area MultiProcessor (SCI/LAMP), FibreChannel, Serial Express, and Universal Serial Bus (USB) offer various benefits to their intended applications. They will not be discussed in the thesis.

1.3 High-Speed Peripheral Applications

Current personal computers (PCs) have separate ports for the keyboard, mouse, serial port, parallel port, sound and game ports, LAN, video port, modem, and perhaps even more ports that can be added by the user such as an SCSI port. Internally the computer may connect the microprocessor to the hard drive, floppy drive, or video card through a parallel bus such as IDE, VESA, or PCI. The IEEE 1394-1995 bus could connect all of these peripherals onto a single bus. The PC can also link to a digital camera, video camera, hard disk drive (HDD), scanner, printer, multimedia, stereo, television, or any time critical application. In fact, all of these devices could be linked together on one bus.

1.4 Thesis Problem Statement

A port circuit has been designed as part of a module to be used in integrated ESP designs. The module will meet IEEE 1394-1995 standards in conjunction with digital Link and Physical Layer modules.

The purpose of the thesis is to verify the port (analog I/O) portion of the Physical Layer on the functional and electrical specification levels, and use this data to perform a power analysis.

1.5 Thesis Outline

The thesis consists of six chapters. Chapter 2 gives an introduction to the general operation of the IEEE 1394-1995 system with an emphasis on the Physical Layer. The analog portion of the Physical Layer circuit to be simulated and analyzed is introduced. It also covers the basics of the Transaction Layer and Link Layer. Chapter 2 also covers the IEEE 1394-1995 specification for the Physical Layer on a detailed level. The criteria defined in Chapter 2 are to be compared to the simulation results generated in the thesis. It then gives a brief overview of the Physical Layer devices currently available in industry and reviews their respective published specifications and implementations.

Chapter 3 introduces the Port Interface Media Cable Model that will be used in simulation of one port talking to another port. This model is necessary to confirm data transmission and reception under given conditions to the IEEE 1394-1995 specification.

Chapter 4 executes a detailed simulation of the three most critical subcircuits in the design. These simulations verify ranges of operation that higher level simulation cannot check. Each simulation will vary typical and worst case wafer fab processing, bias voltage, and temperature.

Chapter 5 presents the simulation and analysis of the complete analog I/O circuit. As with Chapter 4, each simulation will vary typical and worst case wafer fab processing, bias voltage, and temperature. The power dissipation analysis will consider typical and

worst case wafer fab processing at nominal and extreme operating conditions. The simulation will include port status, speed sensing, bus arbitration, data and strobe transmission and reception. The simulation will be executed using two ports connected with the Cable Media Model.

Chapter 6 completes the thesis with conclusions and recommendations. The methods, test cases, and results are discussed with possible improvements to the circuit and simulations.

CHAPTER 2. BACKGROUND

2.1 Scope

The thesis will concentrate on the analog I/O partition of the Physical Layer. IEEE 1394-1995 specification for levels, currents, and other electrical parameters will be checked against simulation results. The remainder of the IEEE 1394-1995 specification will be introduced with a level of detail sufficient to give a general understanding of the system. The IEEE 1394-1995 specification is the presiding and only document covering the high-speed serial bus. Therefore, most of the information will come directly from this specification.

2.1.1 System Overview

The IEEE 1394-1995 high-speed serial bus is designed to connect multiple digital peripherals. The requirements specify a microprocessor-based system connected to a Link device, which in turn is connected to a Physical Layer device. The three layers that use this system are the Transaction Layer, Link Layer, and Physical Layer. The Node Controller is a component within a node that provides a coordination point for management functions for a given node [1]. A node is an addressable device attached to the serial bus with at least the minimum set of control registers [1]. As with the Transaction Layer, the node controller is a software implementation instead of a dedicated device. The block diagram for the basic system is given in Figure 2.1. The Link Layer is implemented on a Link device while the Physical Layer is implemented on a Physical Layer device. A device is

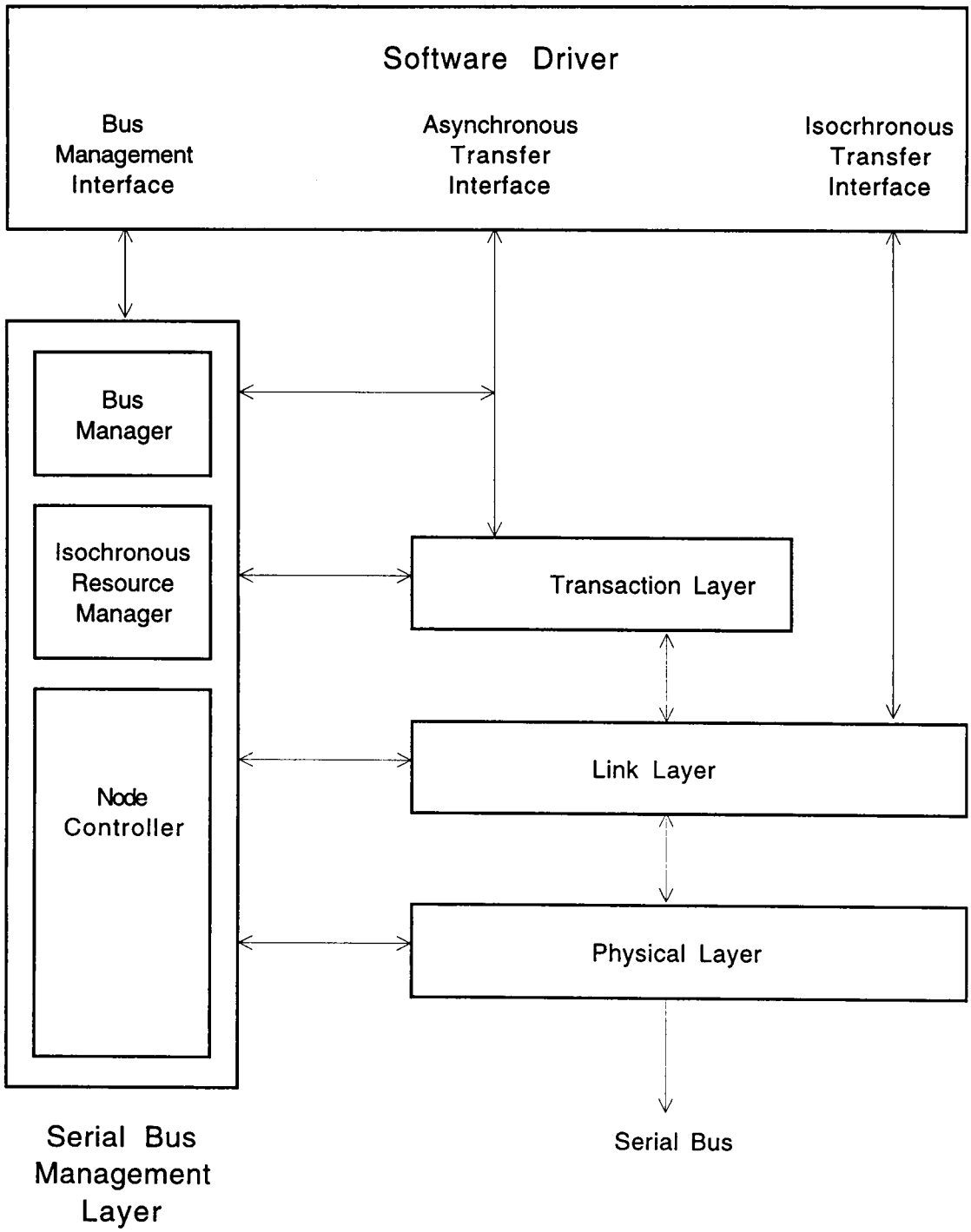


Figure 2.1 - IEEE 1394 Protocol Layers [1]

defined as a circuit on a piece of silicon, such as a computer chip. The Transaction Layer does not have a dedicated device for implementation. The physical network links are implemented with a media cable connected to the Physical Layer device in each peripheral. All three layers are compatible with Control and Status Registers (CSR) Architecture for microcomputer buses [1]. CSR Architecture is not required for high-speed serial bus operation.

2.1.2 Transaction Layer Overview

The Transaction Layer provides three operations for the transfer of data between nodes. These operations are summarized in Table 2.1.

Table 2.1 - Transaction Layer Operations Summary [1]

Transaction	**Description**
Write	Data is transferred to an address in a different node
Read	Data is retrieved from an address in a different node
Lock	Data is sent to a different node, used to perform an indivisible function, with the results returned

The Transaction Layer may perform more than one transaction at a time. It serves as the interface between the Link Layer and the Application Layer. If CSR Architecture is not used, the microprocessor-based system must support its own interface to the Link Layer.

2.1.3 Link Layer Overview

The Link Layer of the high-speed serial bus provides connectionless acknowledged data transfer services between a source node and a destination node [1]. It provides link services between the Transaction Layer and the Physical Layer. Communication to the Physical

Layer is done with a parallel bus that is electrically isolated from the Physical Layer device. The Link Layer handles all packet, formatting, transmission and reception.

There are two basic modes of communication. One mode is Asynchronous Operations. These read and write functions require acknowledgments from other nodes to ensure packets are received correctly. They are not synchronized to time, and are used for applications that do not require 100% accuracy [2]. If a packet is received incorrectly, retries are acceptable although not required. Guaranteed delivery requires Asynchronous communication.

The other mode is Isochronous Operations. In this method time delivery is more critical than accuracy, which is good for applications such as audio, video, and multimedia. Each node is guaranteed a certain amount of time to transmit on the bus. Because there is no acknowledgment or error checking, packet size is reduced [2]. Further discussion of the Transaction Layer is not critical to the scope of this thesis.

2.1.4 Physical Layer Overview

The Physical Layer provides the actual interface between bus nodes. It also provides services for the Link Layer and the Node Controller. These services are given in Table 2.2. These services are described in detail in the IEEE 1394-1995 specification. Each of these services may require the analog I/O portion of the Physical Layer device to function, but the test cases will be limited to only the electrical specifications. The basic operation is that the Link and Physical Layer

Table 2.2 - Summary of Cable Physical Layer Services [1]

Service	**Layer communicated with**	**Purpose of Service**
Cable Physical Layer control request	From the Node Controller	Configure the cable Physical Layer
Cable Physical Layer Control Confirmation	To the Node Controller	Confirm cable Physical Layer control request
Cable Physical Layer event indication	To the Node Controller	Alert Node Controller to events detected in the cable Physical Layer
Cable Physical Layer arbitration request	From the Link Layer	Cause the cable Physical Layer to request control of the bus
Cable Physical Layer arbitration confirmation	To the Link Layer	Confirm cable Physical Layer arbitration request
Cable Physical Layer clock indication	To the Link Layer	Indicate when it is time for the Link Layer to make a cable Physical Layer data request
Cable Physical Layer data request	From the Link Layer	Cause the cable Physical Layer to send one clocked symbol or to start or end a data packet
Cable Physical Layer data indication	To the Link Layer	Indicate the reception of one clocked data symbol or a bus status change

devices communicate on a parallel bus and the Physical Layer performs serial bus arbitration, speed sensing, data transmission and reception. The state machines in the Physical Layer device handle the services mentioned above and simply use the analog I/O as an interface to the other Physical Layer devices. A block diagram of a typical Physical Layer device is given in Figure 2.2. Every manufacturer of Physical Layer devices uses a similar block diagram. The Link Interface is connected directly to the Link Device. The analog portion of this device is located in the "Ports" and the "Bias Voltage and Current Generator" blocks. The ports are connected to ports on other Physical Layer devices via cable media. Simulation and analysis of these two portions with cable media are the main concern of this thesis. Each port consist two sections: Twisted Pair A (TPA) and Twisted Pair B (TPB). During normal operation TPA on one port is connected to TPB on another port via cable media. All communication is implemented with differential voltage signaling. Both TPA and TPB can send and receive data. This analog circuit is implemented with transceivers, comparators, current sources, and a bias voltage source. This circuit is called the Cable Media Interface and is shown in Figure 2.3. TPA can either transmit the data strobe or receive data while TPB can either transmit data or receive the data strobe. TPA has a comparator to test the common mode voltage appearing on the TPA terminals for the speed at which the other connected port (TPB) can transmit. TPB has a signal current generator which generates the common mode voltage levels for speed signaling to TPA. TPA also has a Schmidt Trigger buffer used to generate a signal when a cable media is plugged in while the analog

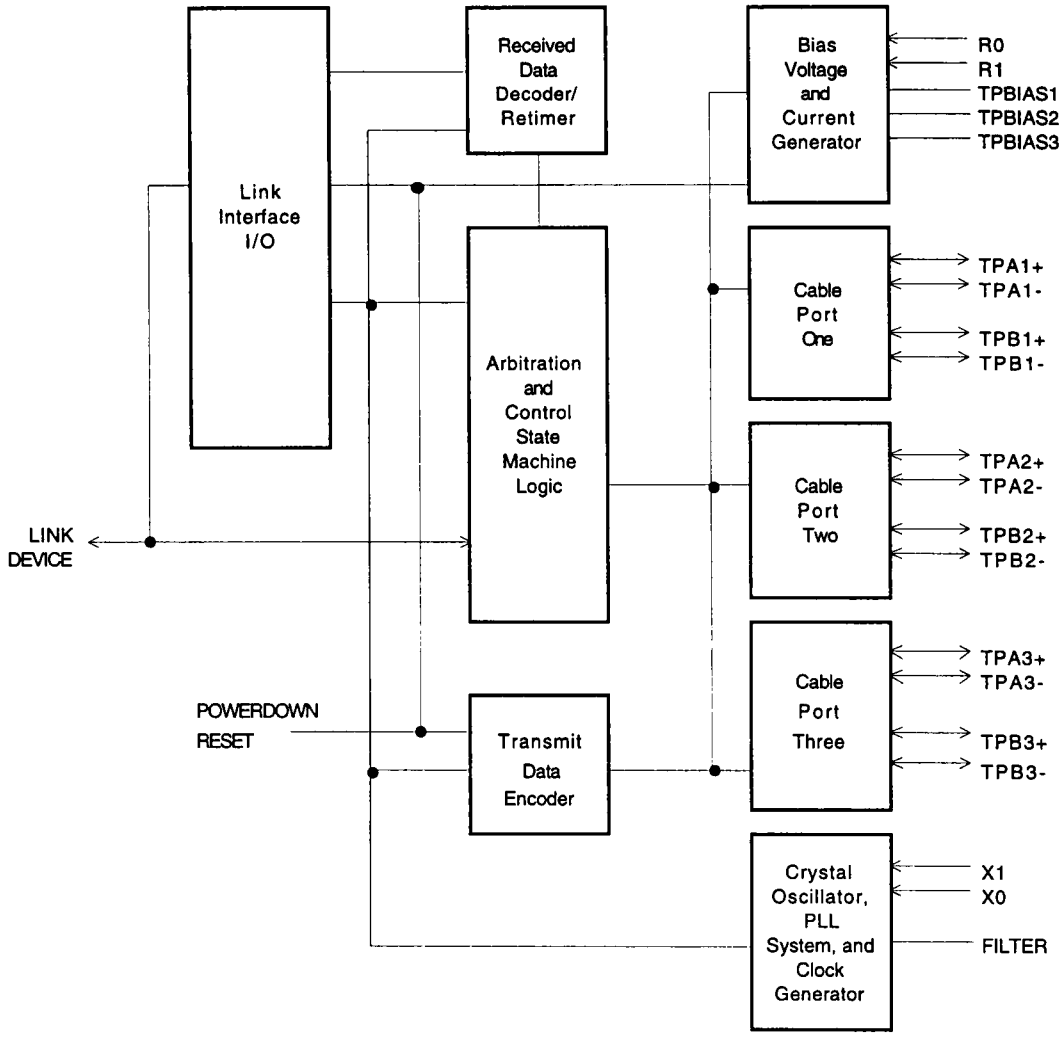


Figure 2.2 - Typical Physical Layer Device

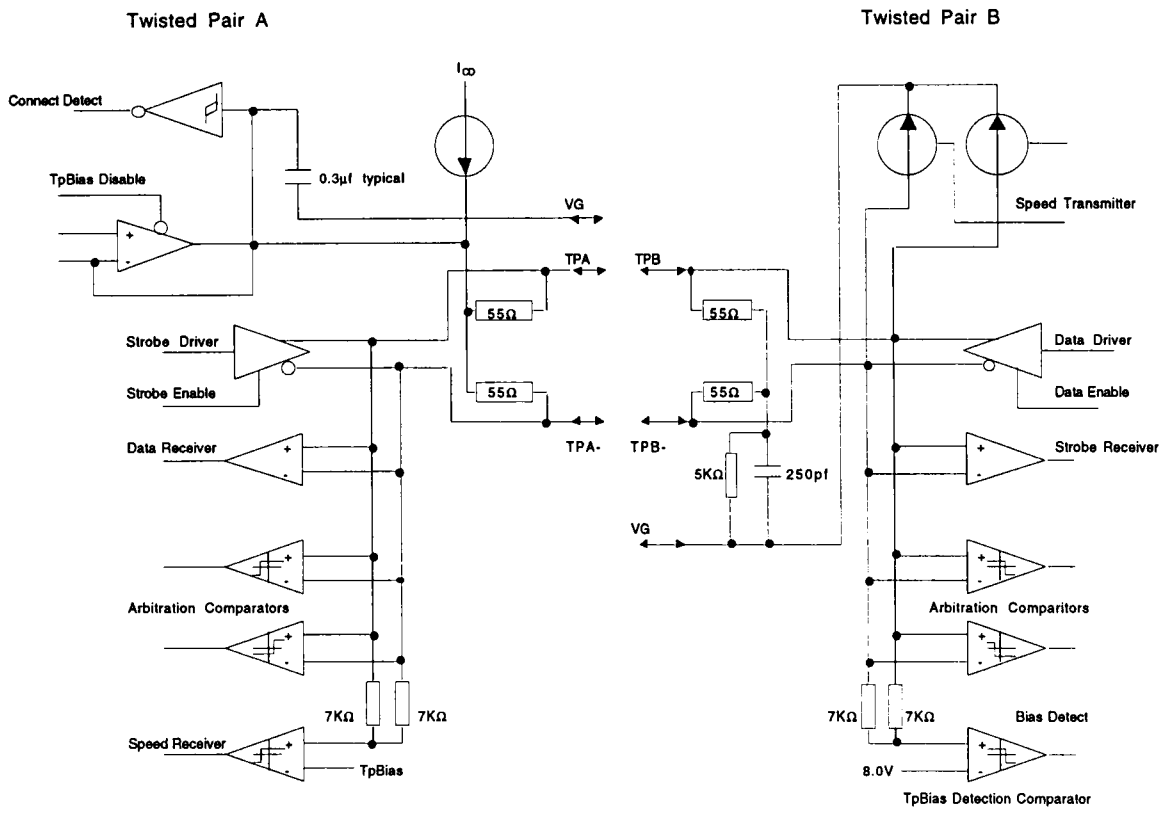


Figure 2.3 - Cable Media Interface [1]

circuit is in a power down state [1]. TPB has a comparator called Bias Detect used to determine if another port is connected. Both TPA and TPB have comparators used for bus arbitration.

The signals *tpa* and *tpa-* are generated with a combination of the strobe driver and the twisted pair bias voltage (*TpBias*). *TpBias* is a reference voltage used as a common mode voltage for *tpa* and *tpa-*. Figure 2.3 shows the strobe drive outputs connected to the *tpa* and *tpa-* terminals. *TpBias* is also shown connected the *tpa* and *tpa-* through individual 55Ω resistors. These two resistors are actually external to the Physical Layer device. When TPA is transmitting data, the common

mode voltage will be *TpBias*. The differential signal amplitudes will be centered around this common mode voltage.

The signals *tpb* and *tpb-* can be generated in two ways. If TPB is transmitting data, then the TPB data driver operates the same way as TPA. As with TPA, TPB has two external 55Ω resistors. However, TPB does not have a *TpBias*. TPB uses the *TpBias* from the TPA of a connected port. Figure 2.3 shows that TPB has two current sources connected to the TPB terminals. The common output of the current sources are configured to serve as a common mode source. If TPB is speed signaling, it will source a certain amount of current which changes the common mode voltage on TPA.

2.1.5 Analog I/O Overview

Motorola engineering staff has designed a CMOS circuit to be used to implement the analog I/O portion of a Physical Layer device. In the IEEE 1394-1995 specification this is called the Media Signal Interface. The author of this thesis did not participate in the circuit design process. The top level diagram for this circuit is given in Figure 2.4. This design actually includes three ports. Only one port is shown here. Each port has various input and output signals. The schematic for a single port and all circuits is given in Figure 2.5. The external analog connections are *tpa*, *tpab*, *tpb*, *tpbb*, and *TpBias*. Each port is serviced by a internal bias circuit which provides a reference *TpBias* voltage, a 25μA current source, and a 25μA current sink. The digital connections are used to control various input and output data and

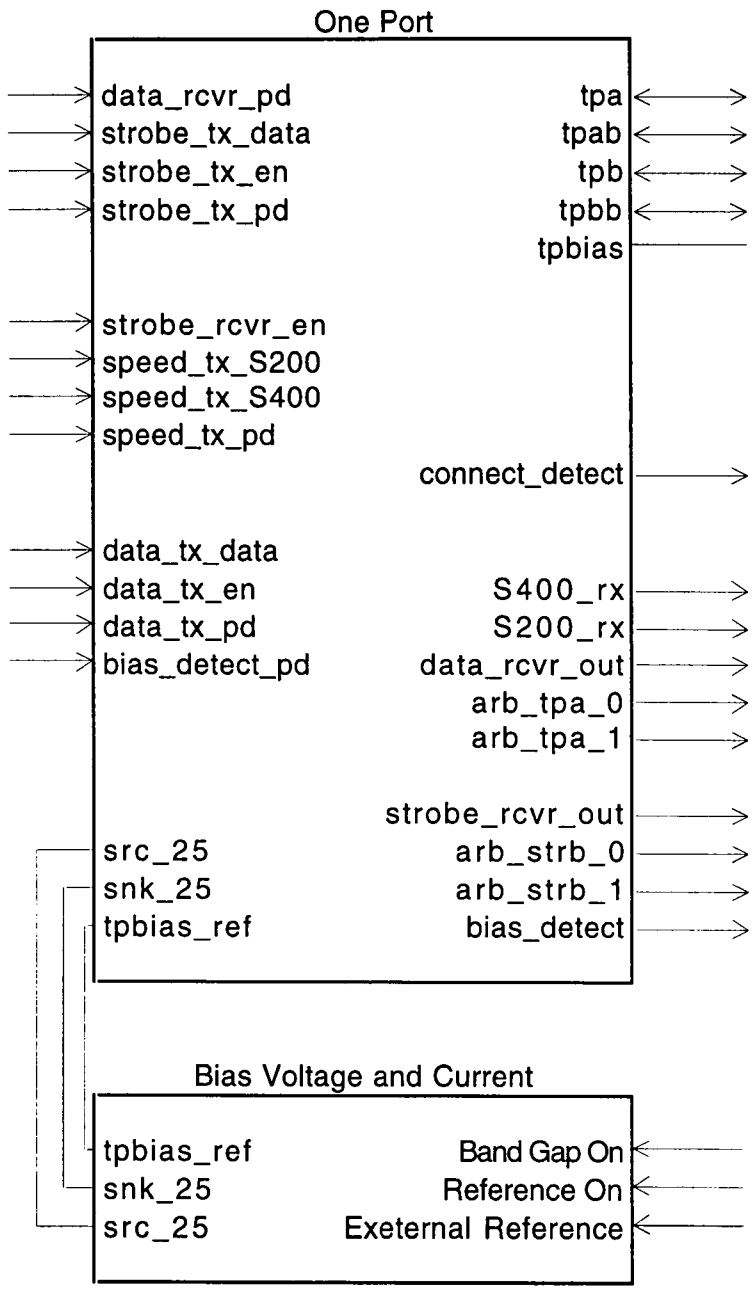


Figure 2.4 - Single Analog I/O Port

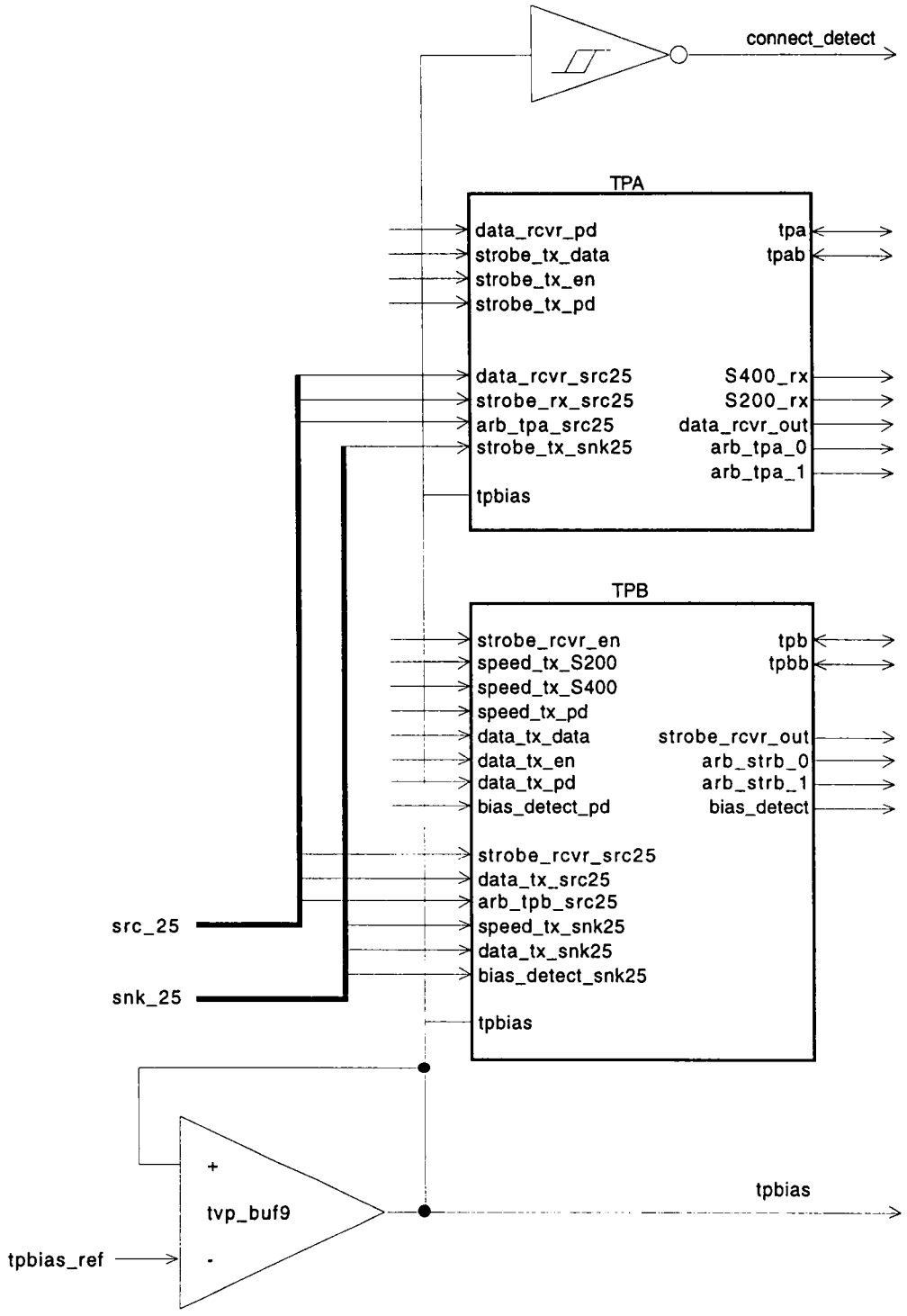


Figure 2.5 - Single Physical Device Port

control signals. A summary of these signals for Port #1 is given in Table 2.3. The I/O column indicates if the signal is an input (I) or an output (O). The A/D column indicates if the signal is analog (A) or digital (D). The digital signals will be connected to the digital modules of Physical Layer, such as the Arbitration and Control logic, Transmit Data Encoder, or Received Data Decoder / Re-Timer. The schematic for TPA is given in Figure 2.6. The operation of each module will be discussed and simulated individually in other sections of the thesis. Notice that the main functions of the TPA part of the port include data receiver, strobe driver, bus arbitration, and speed sensing. The schematic for TPB is given in Figure 2.7. As with TPA, the operation of each module in TPB will be discussed and simulated individually in other sections of the thesis. Notice that the main functions of the TPB part of the port include strobe receiver, data driver, bus arbitration, speed transmission, and port status.

2.1.6 Data Coding

Data and strobe signals are sent over TPA and TPB. These twisted pair signals encode their data as differential voltage. A logic one or zero is encoded based on the voltage levels on a given twisted pair as shown in Figure 2.8. Notice that both voltage signals are always positive with respect to ground. The polarity of the common mode voltage defines a logic one or zero. The voltage levels are discussed in detail in another section. The data and strobe are sent in non-return-to-zero (NRZ) format. An example of this data and strobe transmission is shown in Figure 2.9.

Table 2.3 - Analog Circuit Pin Description

Signal	I/O	A/D	Description
tpa_1	I/O	A	TPA positive differential
tpab_1	I/O	A	TPA negative differential
tpb_1	I/O	A	TPB positive differential
tpbb_1	I/O	A	TPB negative differential
tpbias_1	O	A	Common mode bias voltage
src_25	I	A	25µA source for current mirror bias
snk_25	I	A	25µA sink for current mirror bias
tpbias_ref	I	A	Twisted pair bias voltage
data_rcvr_pd	I	D	Data receiver circuit power down control
strobe_tx_data	I	D	Strobe transmit
strobe_ex_en	I	D	Strobe data circuit enable
strobe_tx_pd	I	D	Strobe transmit circuit power down control
strobe_rcvr_en	I	D	Strobe receiver circuit enable
speed_tx_S200	I	D	200Mbs speed setting control
speed_tx_S400	I	D	400Mbs speed setting control
speed_tx_pd	I	D	Speed setting circuit power down
data_tx_data	I	D	Input data to be transmitted
data_tx_en	I	D	Data transmit circuit enable control
data_tx_pd	I	D	Data transmit circuit power down control
bias_detect_pd	I	D	Port status detection circuit power down
S400_rx	O	D	Indicates speed of next port is 400MPS
S200_rx	O	D	Indicates speed of next port is 200MPS
data_rcvr_out	O	D	Data received from next port
strobe_rcvr_out	O	D	Strobe received from next port
arb_tpa_0	O	D	Decoded TPA arbitration output signal
arb_tpa_1	O	D	Decoded TPA arbitration output signal
arb_strb_0	O	D	Decoded TPB arbitration output signal
arb_strb_1	O	D	Decoded TPB arbitration output signal
connect_detect	O	D	Indicates a port is connected
bias_detect	O	D	Indicates that a port has been plugged in

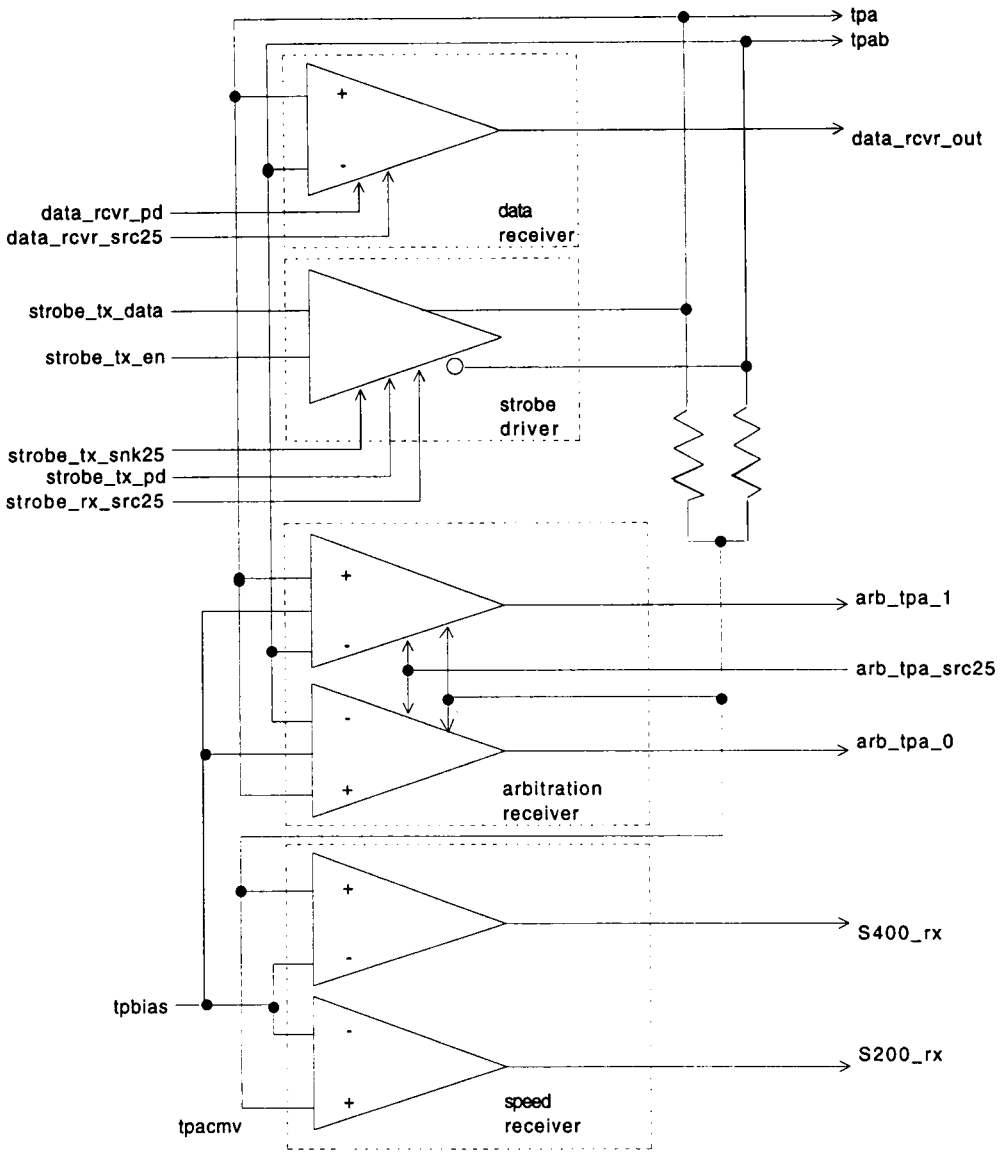


Figure 2.6 - TPA Schematic

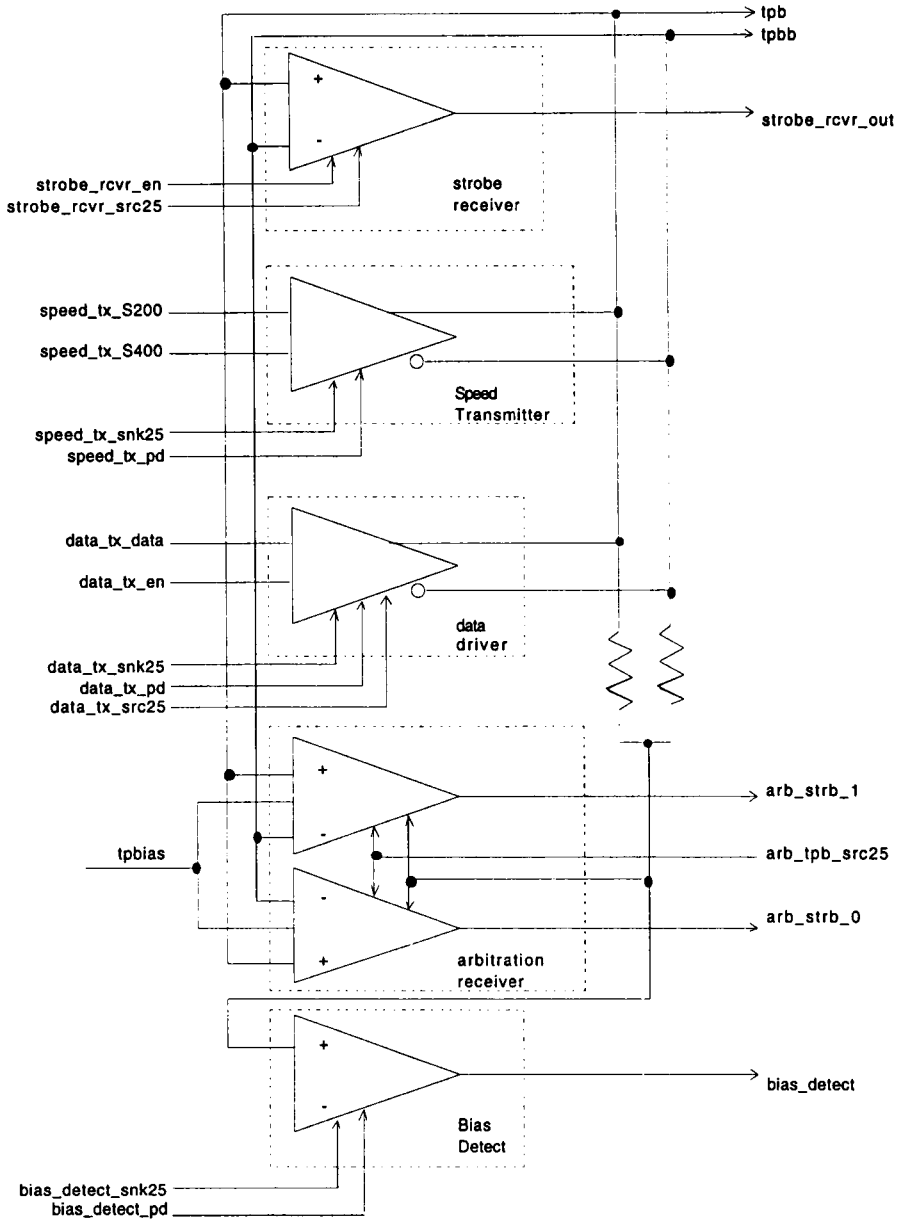


Figure 2.7 - TPB Schematic

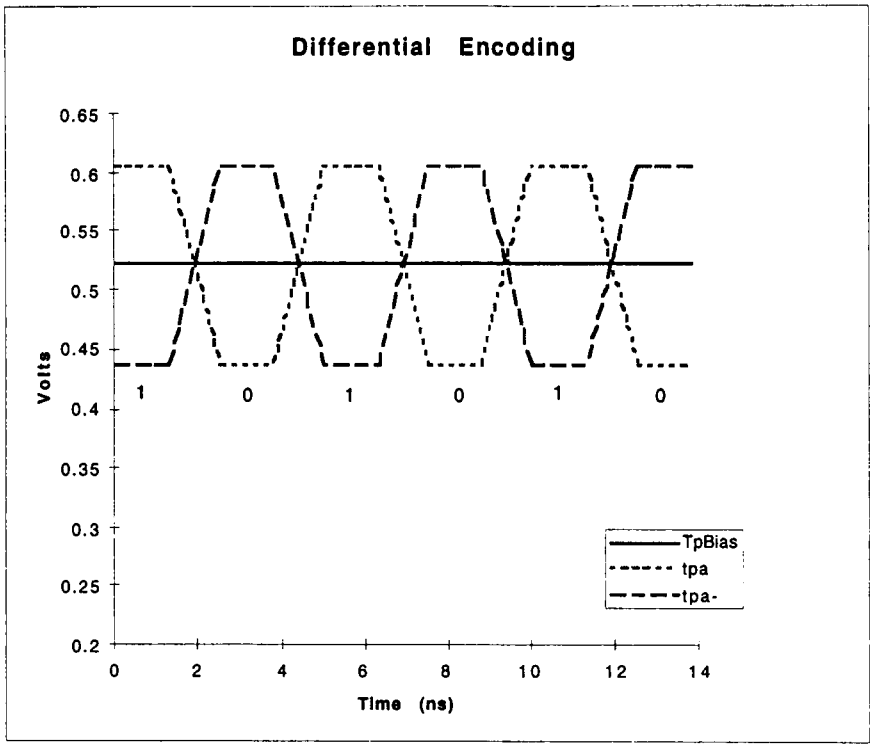


Figure 2.8 - Differential Encoding

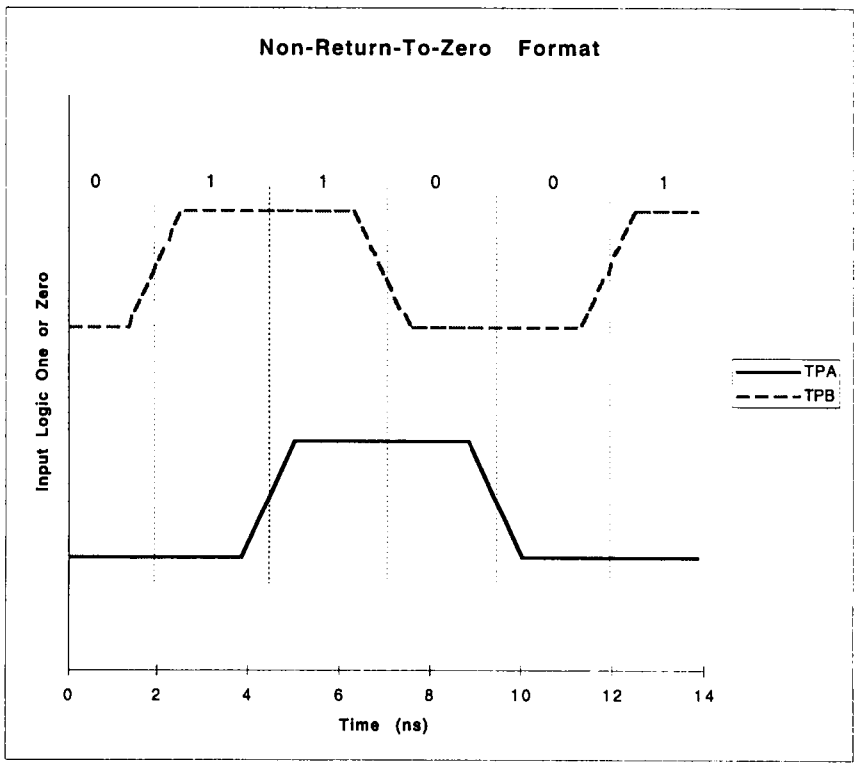


Figure 2.9 - Non-Return-To-Zero Format

Notice that the data and strobe signals do not toggle for every data bit. The only time the data signal toggles is when the data changes state. If the data does not change, then the strobe signal will toggle. Either the data or strobe toggle with each data bit, but never for the same bit. This technique allows a clock to extracted from a logical exclusive-or (XOR) of the data and strobe signal [2]. An example of data and clock recovery is shown in Figure 2.10. This method also reduces many skew problems associated from a clocked data bus [2]. Although the data coding discussed in this section is simple, it is important to remember that network speed sensing and bus arbitration uses the same TPA and TPB wires. Different common mode voltage levels are used for these operations. The Physical Layer is responsible for knowing what activity is taking place.

2.2 Media Signal Interface Specifications

This section presents precise specifications of the electrical parameters of the Media Signal Interface. The Media Signal Interface is equivalent to one port in a Physical Layer device. Most of the following data is taken directly from the IEEE 1394-1995 specification.

2.2.1 Signal Amplitude

TPA and TPB must be able to drive a differential output signal sufficient to overcome loading on the media cable and the input loading on TPB and TPA. The specified range is given it Table 2.4. The differential output voltage levels for TPA are measured across the TPA and TPA- terminals. The same condition applies to TPB. The

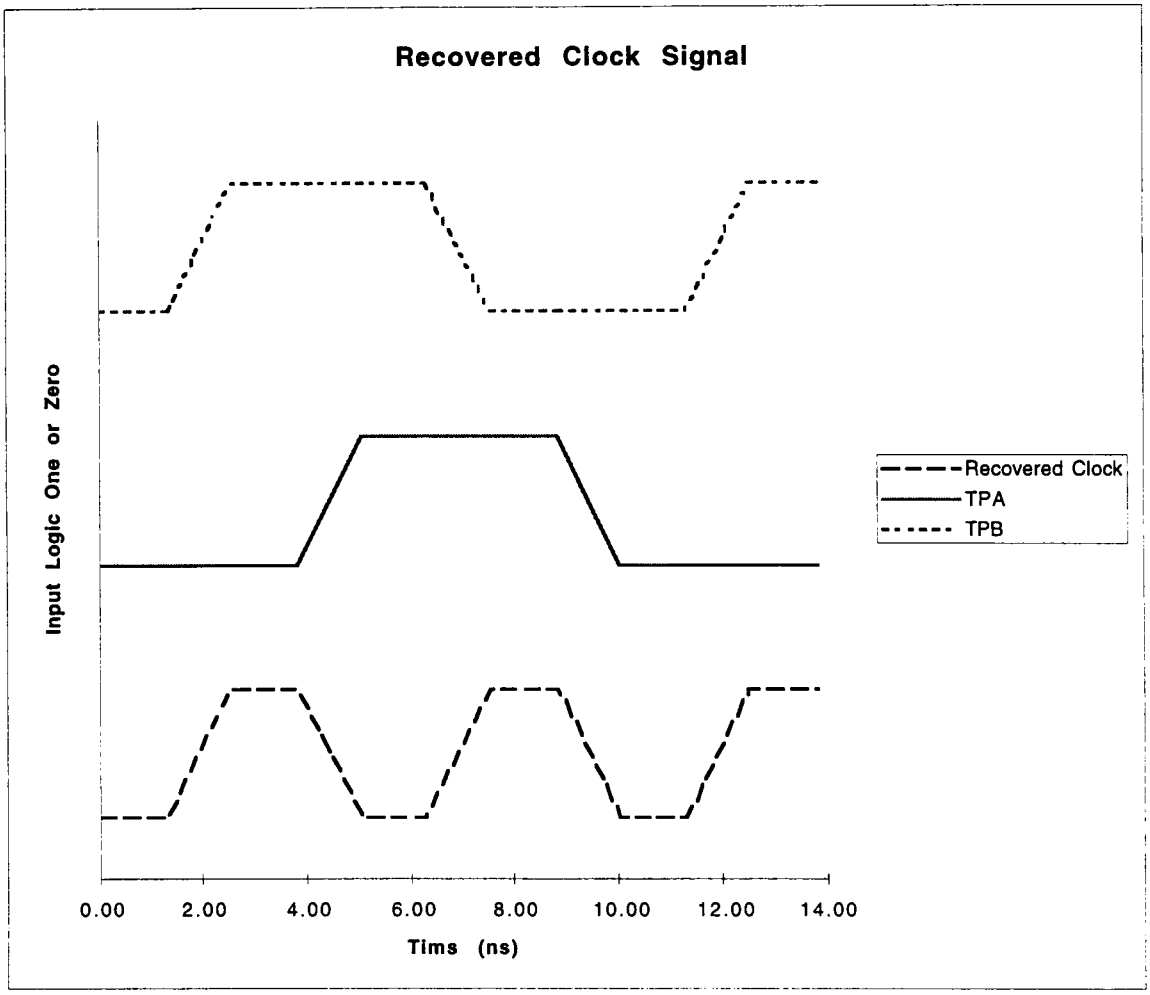


Figure 2.10 - Recovered Clock Signal

specified minimum and maximum differential voltage levels at the receiving end of the cable are given in Table 2.5.

Table 2.4 - Differential Output Signal Amplitude [1]

Maximum	Minimum
265mV	172mV

Table 2.5 - Differential Receive Signal Amplitude [1]

Signal	S100 mV Max	S100 mV Min	S200 mV Max	S200 mV Min	S400 mV Max	S400 mV Min
Arbitration	260	173	262	171	265	168
Data & Strobe	260	142	260	132	260	118

The values given for arbitration apply to the arbitration comparators in the port. The values given for data and strobe apply to the data and strobe receive comparators in the port. These two functions have different amplitude specifications because arbitration uses three differential voltage ranges while data and strobe reception use only two.

2.2.2 Common Mode Voltage

The common mode voltage is the average with respect to ground of the differential voltage on TPA or TPB. The value of the common mode voltage is used by other ports to determine if a port is connected and at what speed to transmit. The values for the common mode output voltages specifications for TPA are given in Table 2.6. If a common mode voltage of 1.665V or above is seen on TPA, then the port will assume that either no speed signaling is present or that the 100Mbs

Table 2.6 - Common Mode Voltage Levels [1]

Pair	Mode	Speed	Limit
TPA	Output	Off	1.665V
TPA	Output	S100	1.665V
TPA	Output	S200	1.438V
TPA	Output	S400	1.030V
TPB	Input	Off	1.165V
TPB	Input	S100	1.165V
TPB	Input	S200	0.935V
TPB	Input	S400	0.523V

(S100) default speed is used. If the level is between 1.438V and 1.665V, then S200 is used. If the level is between 1.030V and 1.438V, then S400 us used. The maximum TPA common mode output voltage is 2.015V while the maximum TPB common mode input voltage is 2.515V. TPB uses the same process to determine speed signaling as TPA. The maximum common mode voltage allowed is 2.015V. The common mode input voltage on TPB is also used to determine if a port is connected. Table 2.7 gives the specified common mode voltage levels used to determine if a port is connected.

Table 2.7 - Port Connection Status Signal [1]

Connection Status	Common Mode Voltage
Port disconnected	TPB < 0.6V
Undefined	0.6V < TPB < 1.0V
Port connected	1.0V < TPB

The common mode voltages given in this section are sampled at selected times during the operation of the port. Connection status is not sensed during speed signaling because the levels may produce a false port disconnected output.

2.2.3 Speed Signaling

Speed signaling is accomplished by TPB in one port by sourcing current to TPA in another port. The sourcing of current will shift the common mode voltage on the terminals at TPA. The *TpBias* generated by TPA is over-powered by the TPB current source. Table 2.8 gives the ranges of the current sourcing required in speed signaling.

Table 2.8 - Common Mode Current Values [1]

Signaling	**Speed**	**Max**	**Min**
Off	S100	0.44mA	-0.81mA
Off	S200	0.44mA	-0.81mA
Off	S400	0.44mA	-0.81mA
On	S100	0.44mA	-0.81mA
On	S200	-2.53mA	-4.84mA
On	S400	-8.10mA	-12.40mA

The comparators in the port have internal resistors as shown in Figure 2.3. These resistors are configured to produce a common mode voltage with reduced current to the speed comparator. This voltage is compared to *TpBias*. The values of the two resistors is determined by what speed the port is trying to detect. A port requires one comparator to detect S100 or S200 speed. If a port is only capable of S100 transmission, a comparator is not required because the default is S100. Two comparators are required if the port has S400 capability.

2.2.4 Arbitration Signal Voltages

Unlike data signaling which uses two differential voltage ranges, arbitration uses three-range signaling. Figure 2.11 shows how the three levels are interpreted.

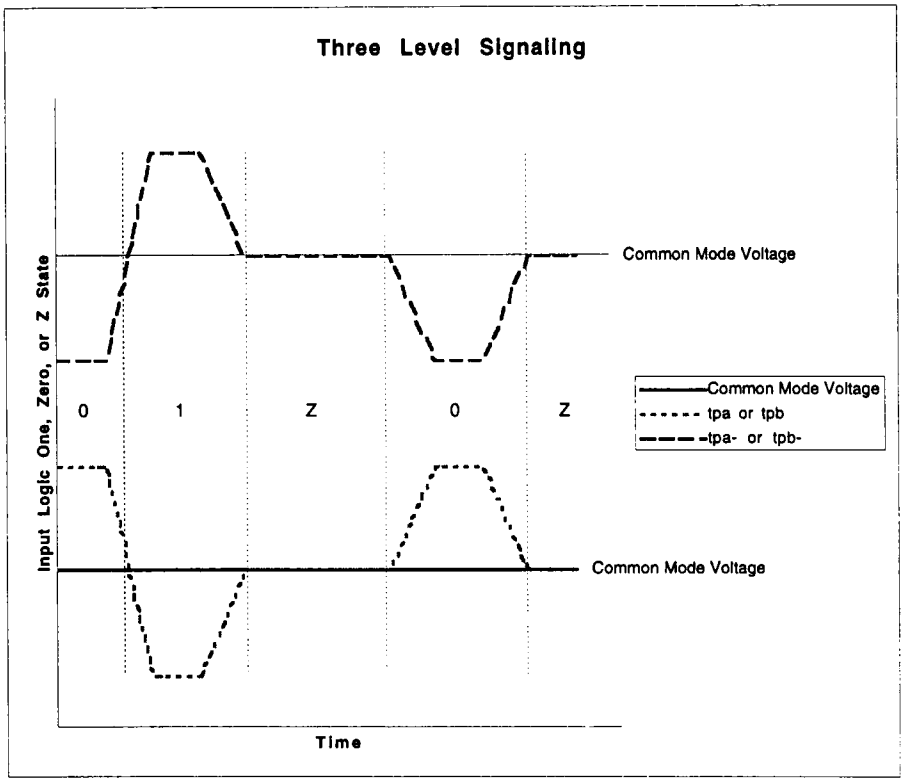


Figure 2.11 - Three Level Signaling [2]

Notice that the logic one or zero is decoded the same way as normal data signaling, but the "Z" state is decoded when the differential signal is zero. When a port is transmitting on TPA or TPB, it can also read the values that appear on the TPA or TPB terminals. For example, if a port transmits a logic one, but reads a Z, the other port must be transmitting a logic 0. The arbitration decoding rules are given in Table 2.9. TPA and TPB generate these states at the same time. The combination of TPA and TPB data are used differently depending on whether the node is a child or parent. The specified voltage levels are given in Table 2.10. Notice that there is a range of +89mV to -89mV for the Z state. This is done because reaching absolutely zero differential voltage is an impossibility. To reach a logic one, the differential

Table 2.9 - Arbitration Decoding Rules [1]

Received Comparator Value	Transmitted Comparator Value	Interpreted Arbitration Signal	Comment
Z	Z	Z	Both ports trans. a Z
0	Z	0	Other port trans. a 0
1	Z	1	Other port trans. a 1
Z	0	1	Other port trans. a 1
0	0	0	Other port trans. a 0 or Z
Z	1	1	Other port trans. a 0
1	1	1	Other port trans. a 1 or Z

Table 2.10 - Arbitration Signaling Levels [1]

Interpreted Signal	Max	Min
1	-	168mV
Z	89mV	-89mV
0	-168mV	-

voltage must be at least 168mV. To reach a logic zero, the differential voltage must be at least -168mV.

2.2.5 Input Impedance

Both TPA and TPB must meet the following input impedance requirements as given in Table 2.11. R_{in} is input resistance, $R_{in\ tx}$ is input resistance in transmit mode, and C_{in} is input capacitance.

2.2.6 Noise

The maximum output common mode noise allowed is 200mV peak-to-peak during arbitration and 20mV during data transmission. The maximum input common mode noise allowed is 225mV peak-to-peak during arbitration and

55mV during data reception. The maximum differential noise shall be 8mV peak-to-peak [1].

Table 2.11 - Differential Input Impedance Specifications [1]

Parameter	Condition	Max	Min	Units
R_{in}	Receive mode	111	109	Ω
R_{in_tx}	Transmit mode	111	105	Ω
C_{in}	S100 only port	9		pF
C_{in}	S200 only port	7		pF
C_{in}	S400 only port	4		pF

2.2.7 Data Rate

The specified data transmission rates are given in Table 2.12.

2.2.8 Data Signal Rise and Fall Times

The output rise and fall times are measured from 10% to 90% of the voltage swing. The input slope is measured at a differential offset of +25mV for rising signals and -25mV for falling signals. These rise times, fall times, and slope rates are specified in Table 2.13.

2.2.9 Jitter and Skew

The jitter and skew specifications are summarized in Table 2.14.

Table 2.12 - Cable Media Data Rates [1]

Data Rate	True Rate	Rate Delta	Units
S100	98.304 ppm	+/-100 ppm	Mbs
S200	196.608 ppm	+/-100 ppm	Mbs
S400	393.216 ppm	+/-100 ppm	Mbs

Table 2.13 - Output Rise Times, Fall Times, and Input Slope [1]

Data Rate	Max rise or fall time	Min slope rate
S100	3.2ns	60mV/ns
S200	2.2ns	110mV/ns
S400	1.2ns	175mV/ns

Table 2.14 - Jitter and Skew Rates [1]

Terminal	S100	S200	S400
Maximum transmitter skew	0.40ns	0.15ns	0.10ns
Maximum transmitter jitter	0.80ns	0.25ns	0.15ns
Maximum skew at receiver	0.80ns	0.55ns	0.50ns
Maximum jitter at receiver	1.08ns	0.50ns	0.315ns

2.3 Physical Devices Available

The final version of IEEE 1394-1995 has not yet been finalized. The devices used to implement IEEE 1394-1995 are manufactured by only a few companies because the technology has only been recently developed. Most companies that produce a Link device also manufacture a Physical Layer device. Some companies license their Link and Physical Layer designs to other companies. Companies that take this option must design the analog I/O portion (media interface) of the Physical Layer device to fit their own libraries and fabrication technology. Texas Instruments (TI), International Business Machines (IBM), Lucent Technologies, Philips, Symbios Logic, and Oxford Semiconductor either manufacture or will license the digital design for a Link or Physical Layer device. All of the devices are very similar from an external point of view. This is the result from following the IEEE 1394-1995 specification. The documentation for these devices is also very similar from company to company.

2.4 Physical Devices Implementation

Figure 2.2 shows the block diagram of a typical Physical Layer device. The thesis will concentrate on the Port, Bias Voltage, and Current Generator portion of the device as they form the media interface. For each port, five connections are brought outside of the Physical Layer device. These signals are used for the TPA and TPB cable interface connections. The typical media cable consists of two shielded twisted pair cables used for TPA and TPB. *TpBias* is used as the reference voltage for TPA. Figure 2.3 shows the electrical schematic for a single port. Notice that the 55Ω resistors are placed to allow *tpa* or *tpa-* to become biased at the same voltage. The actual voltage level on *tpa* or *tpa-* depends on the mode of operation, such as trans., receiving, speed sensing, etc. TPB does not use *TpBias* because the other end of the cable media is connected to TPA on another port, which already has its own *TpBias*. TPB also has a two 55Ω resistor network with a 5kΩ resistor and capacitor. These resistor and capacitor networks also serve as network terminators.

Each port may provide a power and ground that are connected to the cable media. A Physical Layer device will provide a common power to be used by all connected ports. Each port must also monitor this power and report to the Link Interface if the voltage level drops below a specified level of 8V.

The power and ground used by the Physical Layer device must be electrically isolated from the rest of the system, including the Link device. This is needed to prevent ground loops from damaging the system. For example, this problem can occur when one electronic device

is plugged into one AC power circuit while another electronic device is plugged into another AC power circuit. If the two devices are connected with an IEEE 1394-1995 cable, and the AC power to the electronic devices is miss-wired such that a neutral on one device power supply is a ground on another, serious problems can occur. Two common methods used to isolate Physical Layer and Link devices are the transformer and capacitive isolation barrier circuits [3]. The implementations are outside the scope of the thesis.

2.5 Physical Devices Specifications

Six companies that currently manufacture Physical Layer devices publish data sheets. Most of the operational data specified in these data sheets are not relevant to the scope of the thesis. The media interface parameters that are specified in the data sheets and IEEE 1394-1995 specification that will be examined in the thesis are compared here. Table 2.15 compares differential signal amplitude.

Table 2.15 - Signal Amplitude

Manufacturer	Device	Min	Max
IEEE 1394 Spec.	-	172mV	265mV
Texas Instruments [4]	TSB21V03A	172mV	265mV
Lucent Technologies [5]	FW810	172mV	265mV
IBM [6]	IMBS21S760/850	<90mV	265mV
Philips [7]	PDI1394P11	172mV	265mV
Symbios Logic [3]	SYM13FW403	172mV	265mV

In each case the manufacturer matched the IEEE 1394-1995 specification exactly, except IBM which shows a smaller minimum. Most of the parameters compared in this section match exactly, but the true capability of the devices is not known. Some of these devices may have

additional margin built into their design to allow for manufacturing process variances and looser application specifications. The speed signaling common mode input voltage level for TPA is given in Table 2.16.

Table 2.16 - Speed Signaling Common Mode Input Voltage (TPA)

		S100 (V)		S200 (V)		S400 (V)	
Manufacturer	**Device**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**
IEEE 1394 Spec.	-	1.665	2.015	1.438	2.015	1.030	2.015
Texas Instruments [4]	TSB21V03A	1.165	2.015	0.935	2.015	N/A	N/A
Lucent Technologies [5]	FW810	1.665	2.015	1.438	2.015	1.030	2.015
IBM [6]	IMBS21S760	N/A	N/A	1.438	2.015	1.030	2.015
Philips [7]	PDI1394P11	1.665	2.015	0.935	2.015	N/A	N/A
Symbios Logic [3]	SYM13FW403	1.665	2.015	0.935	2.015	1.030	0.523

Entries marked with N/A represent no data available or not applicable. The device data sheets call this data the TPB common-mode input voltage in the nonsource power mode. Nonsource power mode is when one node provides the power for another node to operate. This allows nodes that are powered down to pass on incoming data from the bus, thus the network is not disconnected. Most of the specifications match, but some seem to reflect values that should be in the speed signaling common mode input voltage level for TPB. The reasons for the differences are not clear, but one possibility is a simple mistake in the published data for the Texas Instruments, Philips, and Symbios Logic devices. The speed signaling common mode input voltage level for TPA is given in Table 2.17. In this table, discrepancies with Lucent S400 and IBM S200 and S400 are evident, but most of the values match that of the IEEE 1394-1995 specification. The common mode signaling current is compared in

Table 2.18. There are no differences between the Physical Layer devices and the IEEE 1394-1995 specification in this table. The arbitration input signal voltages are compared in Table 2.19. There are no significant differences between the Physical Layer devices and the IEEE 1394-1995 specification in this table, but there is no entry for the "Z" state. The IEEE 1394-1995 specification denotes the Z state as between -89mV and +89mV.

Differential input impedance in the IEEE 1394-1995 specification is given assuming the 55Ω resistor network is attached. The results are a given input impedance as specified in Table 2.11, about 110Ω. The values given in the current device data books is derived from the twisted pair terminals without this network. Therefore, the values are a minimum of 15KΩ, due to the input resistance of the input and output buffers and drivers in the interface media. These values so large in comparison with that of the resistor network that the input impedance remains virtually unchanged on the system regardless of whether the Physical Layer device is present on the circuit board or not. The IEEE 1394-1995 and current Physical Layer data will not be compared because they represent different resistances. Noise is not specified in the Physical Layer device data sheets and cannot be compared to the IEEE 1394-1995 specification. The data rate of each device matches the IEEE 1394-1995 specification as long as the Physical Layer device input clocks are within specified limits. The data jitter, skew, rise, and fall times for S400 operation are given in Table 2.20.

Table 2.17 - Speed Signaling Common Mode Input Voltage (TPB)

		S100 (V)		S200 (V)		S400 (V)	
Manufacturer	**Device**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**
IEEE 1394 Spec.	-	1.165	2.515	0.935	2.515	0.523	2.515
Texas Instruments [4]	TSB21V03A	1.165	2.515	0.935	2.515	N/A	N/A
Lucent Technologies [5]	FW810	1.165	2.515	0.935	2.515	0.532	2.515
IBM [6]	IMBS21S760/850	N/A	N/A	1.165	2.515	1.165	0.935
Philips [7]	PDI1394P11	1.165	2.515	0.935	2.515	N/A	N/A
Symbios Logic [3]	SYM13FW403	1.165	2.515	0.935	2.515	0.523	0.935

Table 2.18 - Common Mode Speed Signaling Current

		S100 (mA)		S200 (mA)		S400 (mA)	
Manufacturer	**Device**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**
IEEE 1394 Spec.	-	0.44	-0.81	-2.53	-4.84	-8.10	-12.40
Texas Instruments [4]	TSB21V03A	0.44	-0.81	-2.53	-4.84	N/A	N/A
Lucent Technologies [5]	FW810	N/A	N/A	-2.53	-4.84	-8.10	-12.40
IBM [6]	IMBS21S760/850	0.44	-0.81	-2.53	-4.84	-8.10	-12.40
Philips [7]	PDI1394P11	N/A	N/A	-2.53	-4.84	N/A	N/A
Symbios Logic [3]	SYM13FW403	0.44	-0.80	-2.53	-4.84	-8.10	-12.40

Table 2.19 - Arbitration Input Signal Voltages

		Plus (mV)		Minus (mV)	
Manufacturer	**Device**	**Min**	**Max**	**Min**	**Max**
IEEE 1394 Spec.	-	89	168	-89	-168
Texas Instruments [4]	TSB21V03A	89	168	-89	-168
Lucent Technologies [5]	FW810	89	168	-89	-168
IBM [6]	IMBS21S760/850	N/A	N/A	N/A	N/A
Philips [7]	PDI1394P11	89	168	-89	-168
Symbios Logic [3]	SYM13FW403	89	168	-89	-168

Table 2.20 - S400 Maximum Jitter, Skew, Rise, Fall Times

Manufacturer	Device	Jitter	Skew	Rise & Fall
IEEE 1394 Specification	-	0.25ns	0.15ns	1.2ns
Texas Instruments [4]	TSB21V03A	0.25ns	0.15ns	1.2ns
Lucent Technologies [5]	FW810	0.25ns	0.10ns	1.2ns
IBM [6]	IMBS21S760/850	0.25ns	0.15ns	1.2ns
Philips [7]	PDI1394P11	0.25ns	0.15ns	1.2ns
Symbios Logic [3]	SYM13FW403	0.25ns	0.10ns	1.2ns

In each case the manufacturer meets or beats the IEEE 1394-1995 specification limits. S100 and S200 are not compared here because the thesis will simulate only S400 operation. Only one manufacturer published power dissipation data. The IBM 400MHz device has a typical power dissipation of 1W and a maximum of 1.3W.

CHAPTER 3. PORT INTERFACE MEDIA CABLE MODEL

Simulation of port-to-port communication can be done without a cable model by simply connecting TPA from one port to TPB of another port. However, this method would neglect the characteristics of the cable that the media signal interface must overcome to meet IEEE 1394-1995 specifications. The differential and common mode voltages on the twisted pair wires are only a few hundred mV. The currents used to signal speed are in the mA range. Signal degradation over the cable must be included in the simulation to achieve reliable results.

3.1 Cable Media Electrical Specifications

The IEEE 1394-1995 specification covers every mechanical aspect of the media interface cable. The electrical characteristics are the main concerns here. Selected mechanical and electrical characteristics are presented in Table 3.1. The model used for simulation in the thesis will use only 400MHz parameters.

Table 3.1 - Cable Characteristics [8]

Characteristic or Parameter	**Maximum Values**
Cable length	4.5 meters
Differential mode characteristic impedance	110Ω
Common mode characteristic impedance	33Ω
Signal Velocity	5.05 ns/meter
100MHz signal pair attenuation	2.3dB
200MHz signal pair attenuation	3.2dB
400MHz signal pair attenuation	5.8dB
100MHz Relative propagation skew	400ps
200MHz Relative propagation skew	Not Specified
400MHz Relative propagation skew	100ps
TPA to TPB crosstalk	-26dB

3.2 Spice Model Theory

The cable to be modeled consists of 4.5 meters of twisted pair cable with a 25mm connector on each end. The connectors and cable can be simulated using Spice lossy transmission line (LTRA) elements along with basic inductors and capacitors. The LTRA must be tuned to match the specified characteristics in Table 3.1. Research in modeling for the IEEE 1394-1995 cable has produced a model that can be used to simulate common mode and differential mode signal propagation. It was developed by engineers at the Intel Corporation's Media & Interconnect Technology Lab and presented to the IEEE. This model has been modified to fit the version of spice used by the thesis. The block diagram of the Spice model of the differential pair transmission lines is given in Figure 3.1.

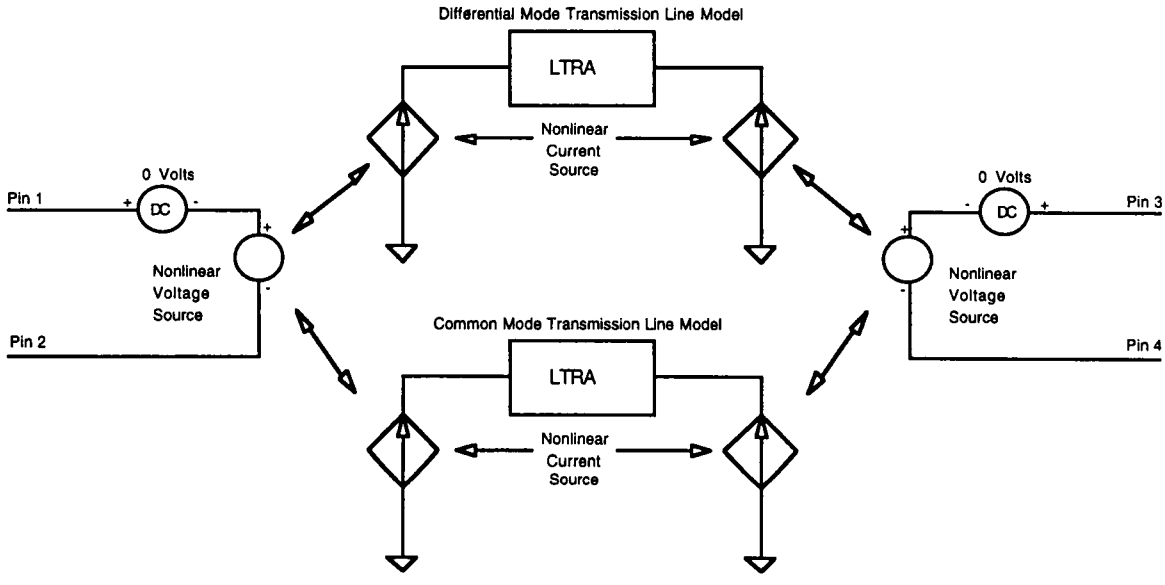


Figure 3.1 - Spice Model of Differential Pair Transmission Lines [9]

The Differential and Common Mode Transmission Line models are separated because electrical characteristics are different at DC and AC.

Nonlinear sources are used to permit modeling arbitrary combinations of excitation currents into the transmission line model [9]. The connectors are modeled differently for differential and common mode operation. Figure 3.2 shows the model for a connector and transmission line in the differential mode.

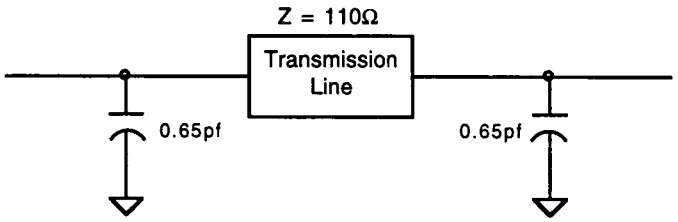


Figure 3.2 - Differential Mode Connector And Cable Spice Model [9]

The capacitor values are chosen to match that of the connector at DC operation. The impedance values matches the IEEE 1394-1995 specification. Figure 3.3 shows the model for a connector and transmission line in the common mode.

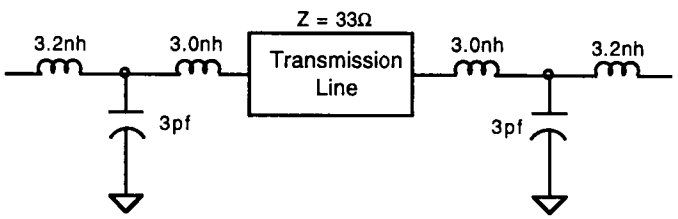


Figure 3.3 - Common Mode Connector And Cable Spice Model [9]

The inductance and capacitor values are chosen to match that of the connector at AC operation. The impedance values matches the IEEE 1394-1995 specification.

3.3 Method of Implementation in Simulation

The schematic of the Spice model used to simulate the two connectors and cable is given in Figure 3.4. The common mode and differential model use separate paths. The connectors also have a short transmission line added to the model to simulate the 25mm connector length. Terminals V1, V2, V3, and V4 denote where TPA and TPB will be connected. The external resistors and capacitors that are specified in the media signal interface schematic have been added to the model. The 4.5m cable is simulated using Spice transmission line models. The transmission line model is made from a LUMP cell, which is shown in Figure 3.5. The LUMP option specifics how many of these cells are connected serially to model the lossy transmission line. The number of cells plays a major role in the trade-off between accuracy and run time efficiency. The default number of cells was chosen because it generally yields the most accurate results [10].

3.4 Simulation and Conformation of Media Cable Model

The Spice net list and control file for the cable model are given in the Appendix as Cable Model Spice Net List and Cable Model Spice Control File. Two voltage sources are used to drive TPA with a simulated switching of 400MHz burst of eight data bits. The common mode voltage was set 1.84V. The differential amplitude was set to be 165mV, which is the worst case differential output signal amplitude allowed by the IEEE 1394-1995 specification. The input and output of this simulation are given in Figure 3.6.

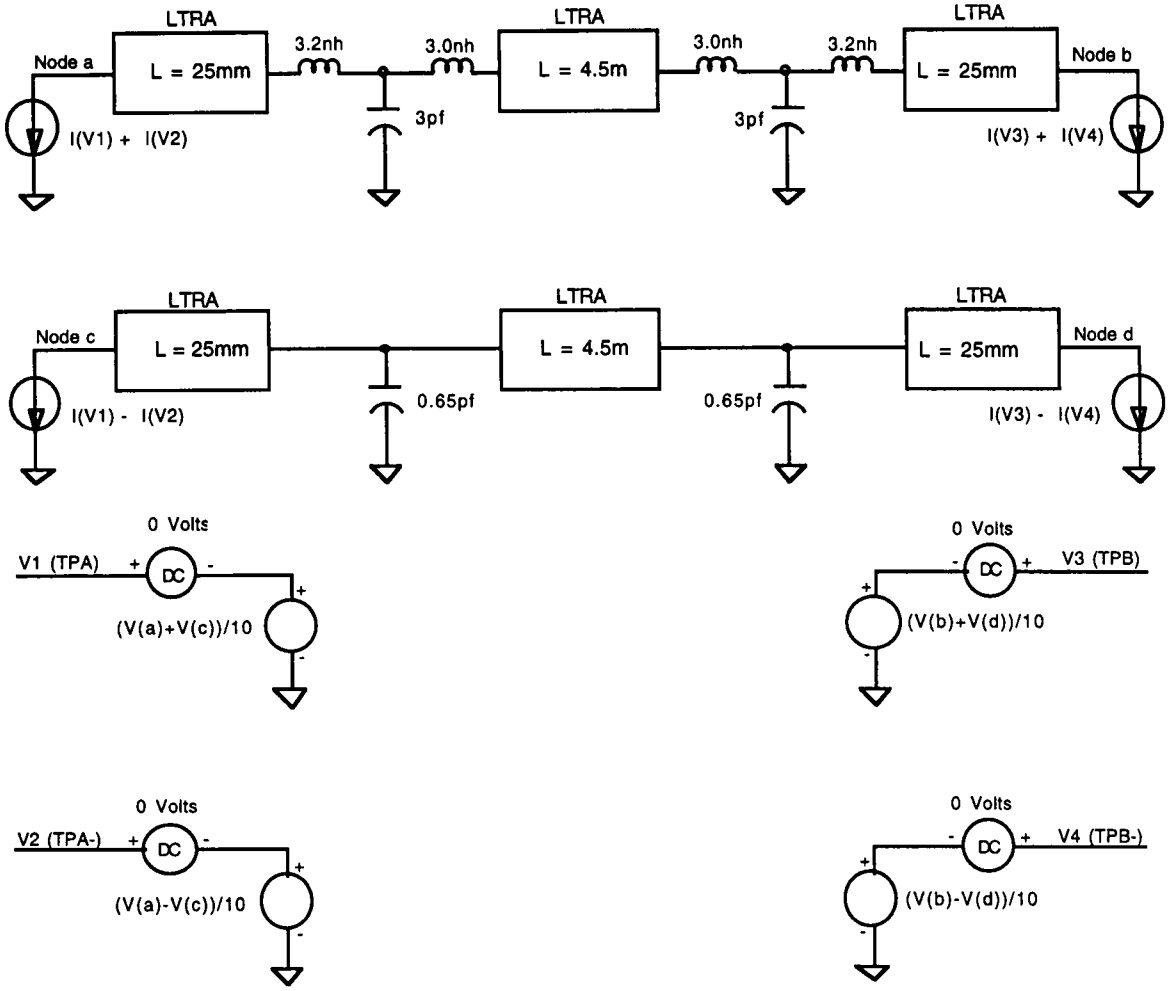


Figure 3.4 - Connector And Cable Spice Model Schematic

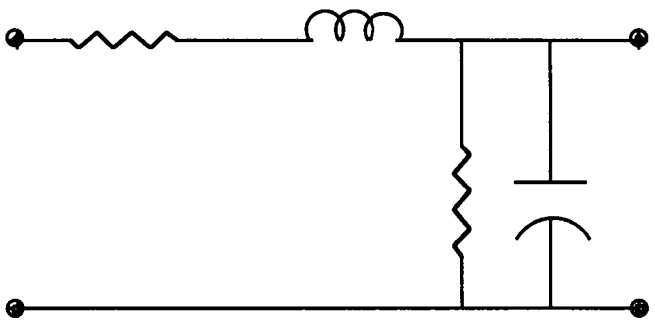


Figure 3.5 - Lump Elementary Cell Structures [10]

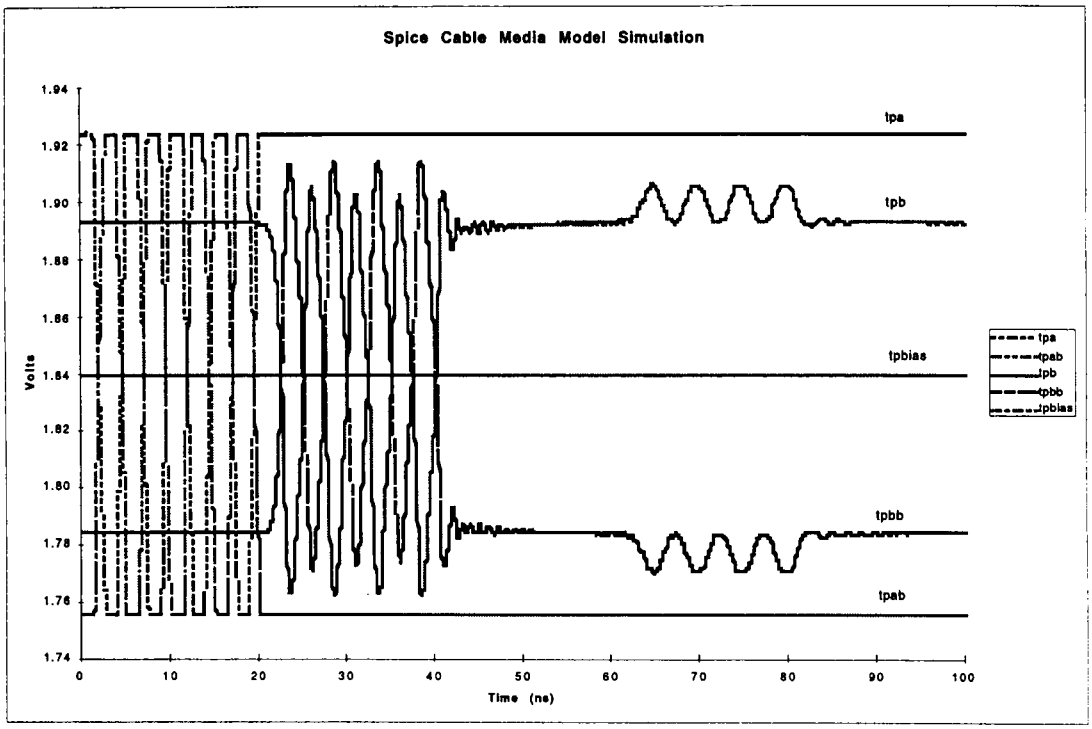


Figure 3.6 - Data Transmission And Reception Cable Simulation

The signal delay of 23ns matches the maximum specified 23ns allowed. This is calculated by the maximum of 5.05ns/meter over 4.5 meters. The attenuation is measured as the differential gain of output to input voltage. The simulation yielded 3.27dB attenuation based on 165mV differential input and 110mV differential output. The maximum allowed value is 5.8dB at 400MHz. The rise time is 1.75ns based on 10% to 90%. The skew and crosstalk will not be simulated in the thesis. The cable meets the specifications as given in Table 3.1.

CHAPTER 4. SUBCIRCUIT SIMULATION AND ANALYSIS

The electrical media interface that the thesis will simulate consists of fifteen different subcircuits. Each circuit is designed to perform either a direct interface to the cable media or to provide various reference or bias voltages or currents. Three of these circuits are critical to the operation of the entire circuit. They are the Arbitration Receiver, Data and Strobe Transmit Driver, and the Data and Strobe Transmit Receiver circuits. The remaining twelve subcircuits operate at DC levels only and can be verified in the complete port simulation. The simulation of these three subcircuits is designed to cover all possible inputs and outputs at different operating conditions as well as circuits that are manufactured with different process characteristics. This cannot be accomplished with a connected port and cable media because in the input and output levels cannot be controlled in the precise ranges needed to verify operation across all specifications. The output from each simulation will be compared against the IEEE 1394-1995 specification. The simulation will include an operating temperature, device bias voltage, parasitic model, and transistor model. The parasitic and transistor models are used to set wafer fab process parameters to a best, typical, and worst-case scenario. The simulation matrix to be used on each subcircuit is given in Table 4.1. Only one of the simulations is done on a typical device at nominal operating conditions. The remaining simulations are performed on worst-case devices at various operating conditions.

Table 4.1 - Simulation Matrix

Parasitic Model	Transistor Model	Operating Voltage	Operating Temperature
Typical Case	Typical Case	Nominal	Nominal
Worst-Case	Worst-Case	Nominal	Nominal
Worst-Case	Worst-Case	Nominal	High
Worst-Case	Worst-Case	Nominal	Low
Worst-Case	Worst-Case	High	Nominal
Worst-Case	Worst-Case	High	High
Worst-Case	Worst-Case	High	Low
Worst-Case	Worst-Case	Low	Nominal
Worst-Case	Worst-Case	Low	High
Worst-Case	Worst-Case	Low	Low

This methodology verifies that each subcircuit design is robust and can still function at degraded conditions and process characteristics. The operating voltage and temperature values are given in Table 4.2.

Table 4.2 - Operating Condition Values

Parameter	Condition	Value
Voltage	Nominal	3.3V
Voltage	High	3.70V
Voltage	Low	2.90V
Temperature	Nominal	25C
Temperature	High	100C
Temperature	Low	0C

These values are chosen to verify performance over a broad range of operating conditions. The parasitic model varies parameters such as the transistor gate poly-silicon width (poly gate width), the resistances, and the capacitances of transistors. The selection of the interconnect width is based on the assumption that the parasitics are dominated by the interconnect capacitances rather than the resistances. Table 4.3 shows what parameters are adjusted for each model.

Table 4.3 - Parasitic Model Parameters [11]

Parameter	Typical-Case	Worst-Case
Interconnect Geometry Bias	Nominal	Increase
N-well Geometry Bias	Nominal	Increase
Inter Layer Dielectric Thickness	Nominal	Decrease
Conducting Layer Thickness	Nominal	Increase
N-well Sheet Resistance	Nominal	Decrease
Interconnect Sheet Resistance	Nominal	Decrease
Contact/Via Resistance	Nominal	Increase
Junction Leakage	Nominal	Increase
Junction Capacitance	Nominal	Increase

N-well is the area in silicon that is implanted with n-type material such as arsenic or phosphorous. A CMOS transistor is built inside of this N-well area. The worst-case values are chosen to induce parasitic capacitance into the circuit. This will change the characteristics of the transistors to be slower and induce timing skew. Analog performance may also change, such as bias voltages, bias currents, output voltages, and input ranges.

For the transistor models, the typical and worst-case parameters are selected to produce the typical or weakest current drive capabilities in the transistors. These parameters are summarized in Table 4.4. The worst-case parameters clearly reduce the speed and drive capabilities of the transistors. Spice simulations that use worst-case parasitic and transistor models under extreme operating temperature and voltage levels will verify the performance of each subcircuit.

Table 4.4 - Transistor Model Parameters [11]

Parameter	Typical Case	Worst-Case NMOS	Worst-Case PMOS
Gate Oxide Thickness	Nominal	Increase	Increase
Threshold Voltage	Nominal	Increase	Decrease
Source/Drain Diffusion Under Gate	Nominal	Decrease	Decrease
Poly Gate Width	Nominal	Decrease	Decrease
Active Channel Length	Nominal	Increase	Increase
Tank Diffusion	Nominal	Increase	Increase
Poly Sheet Resistance	Nominal	Increase	Increase

4.1 Arbitration Receiver Circuit

The Arbitration Receiver circuit is used to detect what voltage levels are being recieved into the port. These signals are discussed in Section 2.2.4. The functional diagram for the Arbitration Receiver circuit is included in the Electrical Media Interface shown in Figure 2.3. The Arbitration Receiver circuit consist of a set of comparators. The block diagram of the circuit used for this implementation is shown in Figure 4.1. There are two sets of comparators in the design. This is necessary because the input common mode voltage has a specified range from 0.523V to 2.515V for all speeds. For 400MHz operation, the minimum common mode input voltage is 0.523V. The dimensions of the transistors limit the operational range of a comparator to a lower or higher voltage of this common mode range. Hence, two separate sets of comparators are needed to cover the entire range. A fifth comparator is used to determine which set of comparators should be selected. It is denoted as "Common Mode Voltage Range Detector" in Figure 4.1.1. If the input common mode voltage is below the *offtpbias* level, then the two comparators denoted as "Lower Common Mode Voltage Range Comparator" in Figure 4.1.1 are selected. The input common mode voltage is denoted as

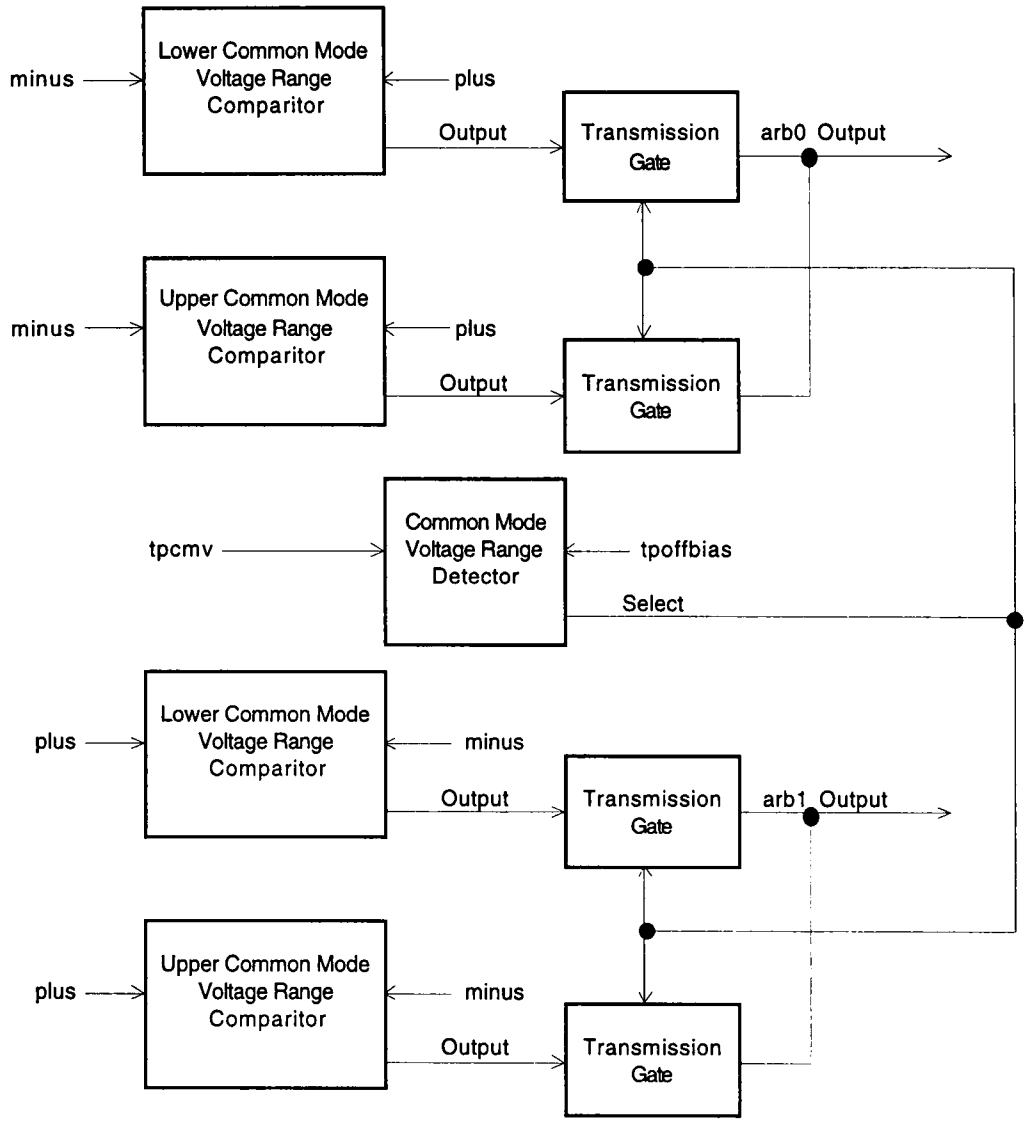


Figure 4.1.1 - Arbitration Receiver Circuit Block Diagram

tpcmv in Figure 4.1.1. When *tpcmv* reaches a level above *tpbias*, then the two comparators denoted as "Upper Common Mode Voltage Range Comparator" in Figure 4.1.1 are selected. The schematic for the CMOS circuit used for this implementation is given in the Appendix as the Arbitration Receiver Schematic. Each comparator can be described as a current mirror within a differential amplifier. The output of each comparator is selected by enabling a transmission gate. The *select* signal that controls the transmission gates in Arbitration Receiver circuit is the output from the Common Mode Voltage Range Detector comparator. The *offtpbias* signal is generated by an internal voltage divider as shown in the Arbitration Receiver circuit. The 25μA current sources and sinks are provided by bias circuitry from the Reference Current Amplifier subcircuit.

4.1.1 Test Case Description

The inputs to the Arbitration Receiver circuit are either TPA or TPB. They are connected to the *plus* and *minus* signals on each comparator. The differential amplitudes generated by TPA or TPB will meet the requirements of Table 2.10. The digital outputs are *arb0* and *arb1*. The expected *arb0* and *arb1* outputs for each input differential amplitude regardless of the common mode voltage is given in Table 4.1.1.

Table 4.1.1 - Arbitration Truth Table

Interpreted Physical Signal	**Maximum Differential Amplitude**	**Minimum Differential Amplitude**	**arb0 Digital Output**	**arb1 Digital Output**
1	-	168mV	Low	High
Z	89mV	-89mV	Low	Low
0	-168mV	-	High	Low

The differential amplitudes in Table 4.1.1 will be centered around the TPA or TPB common mode voltage. The common mode voltage will match that of Table 2.6. The test input conditions are generated using these two tables. The common mode voltage range combined with the differential amplitude produce the test conditions given in Table 4.1.2. These test conditions cover the entire specified range. The Select Control Signal column indicates when the select line enables the transmission gates to use the output from the lower to upper range common mode input voltage. The *arb0* and *arb1* digital outputs remain at zero unless the differential input voltage reaches a certain amplitude.

The *plus* and *minus* voltages given in Table 4.1.2 are implemented in a Spice program to simulate the Arbitration Receiver circuit. The Spice control file is given in the Appendix as the Arbitration Spice Control File. The input voltage levels are is plotted in Figure 4.1.2. The input common mode voltage starts at 0.5V and steps up by 0.5V after the four differential amplitude cases are executed. The final common mode voltage is 3.0V. This is above the specified range of 2.515V but it is included here to test for the range of the circuit operation.

4.1.2 Process and Operating Parameter Simulation Matrix

The simulation matrix given in Table 4.1 was applied to the Arbitration circuit. The input voltage levels, *select* control line, *arb0* and *arb1* outputs for the typical processing and nominal operating conditions are plotted in Figure 4.1.3.

Table 4.1.2 - Arbitration Test Conditions

Common Mode Voltage (V)	Plus Input (V)	Minus Input (V)	Select Control Signal	arb0 Digital Output	arb1 Digital Output
0.5	0.545	0.445	Low	Low	Low
0.5	0.445	0.545	Low	Low	Low
0.5	0.584	0.416	Low	High	Low
0.5	0.416	0.584	Low	Low	High
1.0	1.045	0.955	Low	Low	Low
1.0	0.955	1.045	Low	Low	Low
1.0	1.084	0.916	Low	High	Low
1.0	0.916	1.084	Low	Low	High
1.5	1.545	1.455	Low	Low	Low
1.5	1.455	1.545	Low	Low	Low
1.5	1.584	1.416	Low	High	Low
1.5	1.416	1.584	Low	Low	High
2.0	2.045	1.955	High	Low	Low
2.0	1.955	2.045	High	Low	Low
2.0	2.084	1.916	High	High	Low
2.0	1.916	2.084	High	Low	High
2.5	2.545	2.455	High	Low	Low
2.5	2.455	2.545	High	Low	Low
2.5	2.584	2.416	High	High	Low
2.5	2.416	2.584	High	Low	High
3.0	3.045	2.955	High	Low	Low
3.0	2.955	3.045	High	Low	Low
3.0	3.084	3.916	High	High	Low
3.0	3.916	3.084	High	Low	High

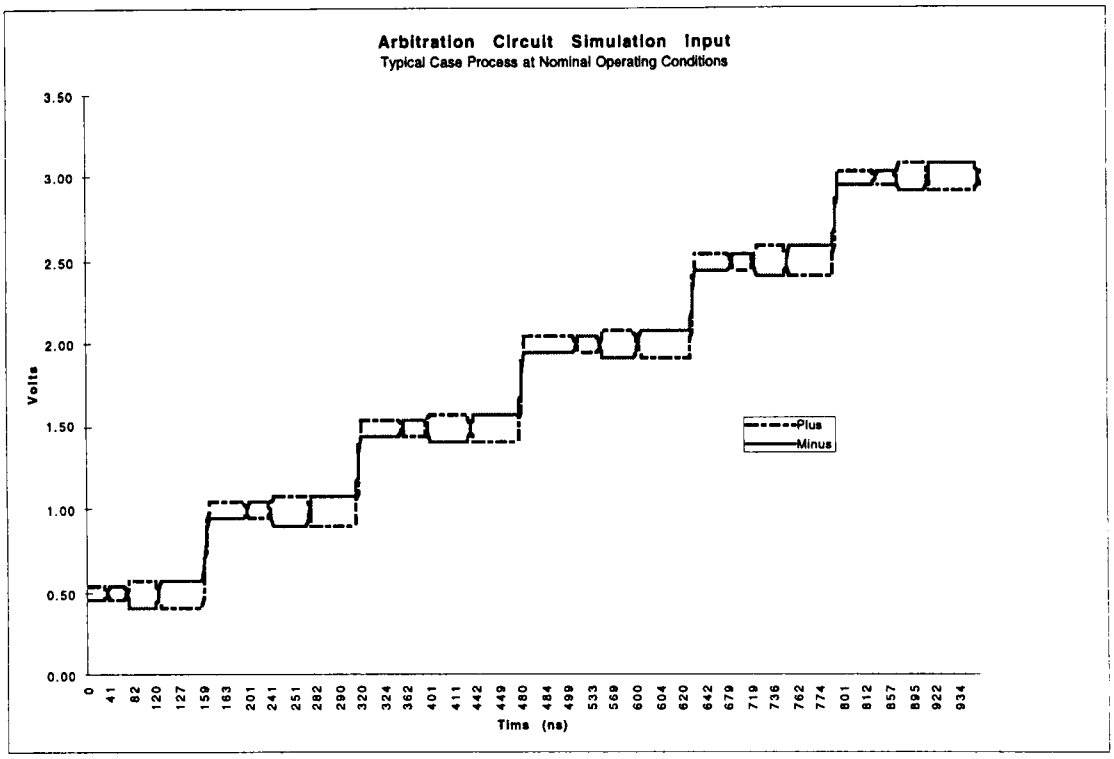


Figure 4.1.2 - Spice Arbitration Input Voltages

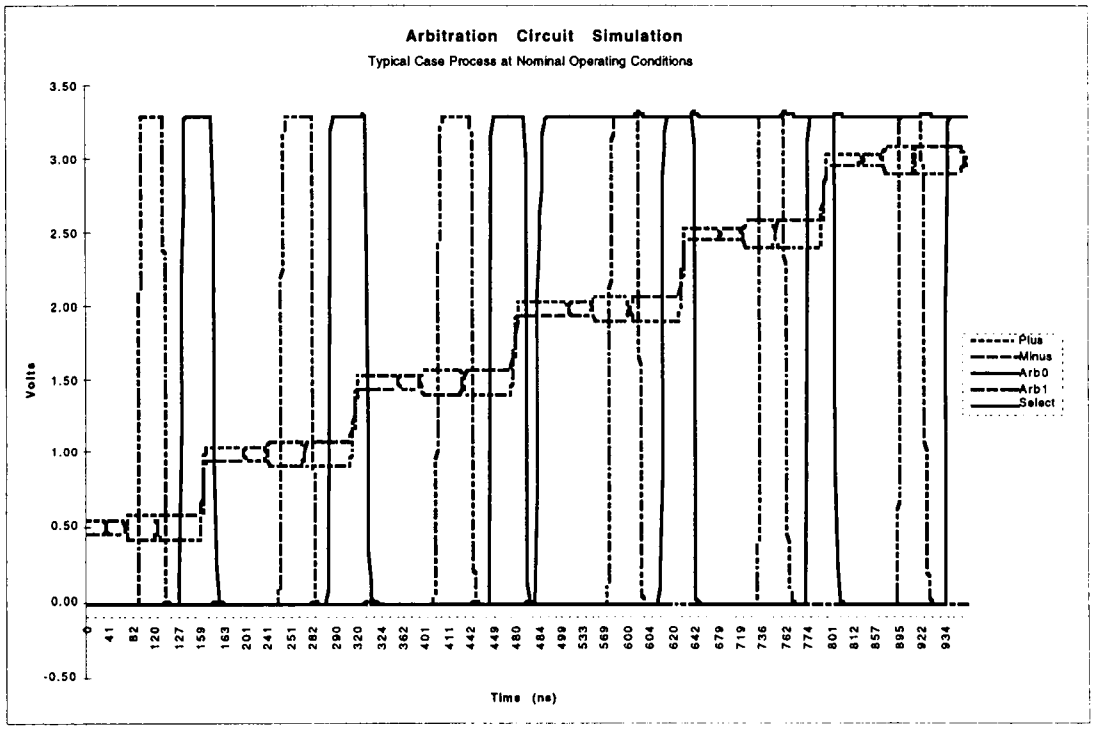


Figure 4.1.3 - Spice Arbitration Simulation Results

In every case *arb0* and *arb1* have the correct output. The select signal toggles between the common mode voltages of 1.5V and 2.0V. The simulation was repeated at all corners of the matrix. The only simulations with failing results was when the operating voltage was below the common mode input voltage. This is when V_{dd} is 2.9V while the common mode voltage was 3.0V. The plot of this output is given in Figure 4.1.4. The maximum allowed common mode voltage is 2.515V so the circuit still passes all specifications. These results are the same for any simulation with V_{dd} set to 2.9V. All simulations with V_{dd} set to 3.3V or 3.7V passed for all common mode ranges regardless of processing corner or temperature.

4.2 Data and Strobe Transmit Driver

The Data and Strobe Driver Circuit is used to transmit strobe for TPA or transmit data for TPB. The functional diagram for the Data and Strobe Transmit Driver Circuit is included in the Electrical Media Interface in Figure 2.3. The output of the driver is driven with a series of P-channel transistors for high voltage output. Describing the output as high voltage is relevant to *tpbias*. The outputs *tpa* and *tpab* should not be more than a few hundred milli-volts above *tpbias*. The *tpbias* reference should be about 1.84V. The P-channel transistors are sourcing current in this mode. The low voltage output is driven by a N-channel transistor, which is sinking current into the circuit in this mode.

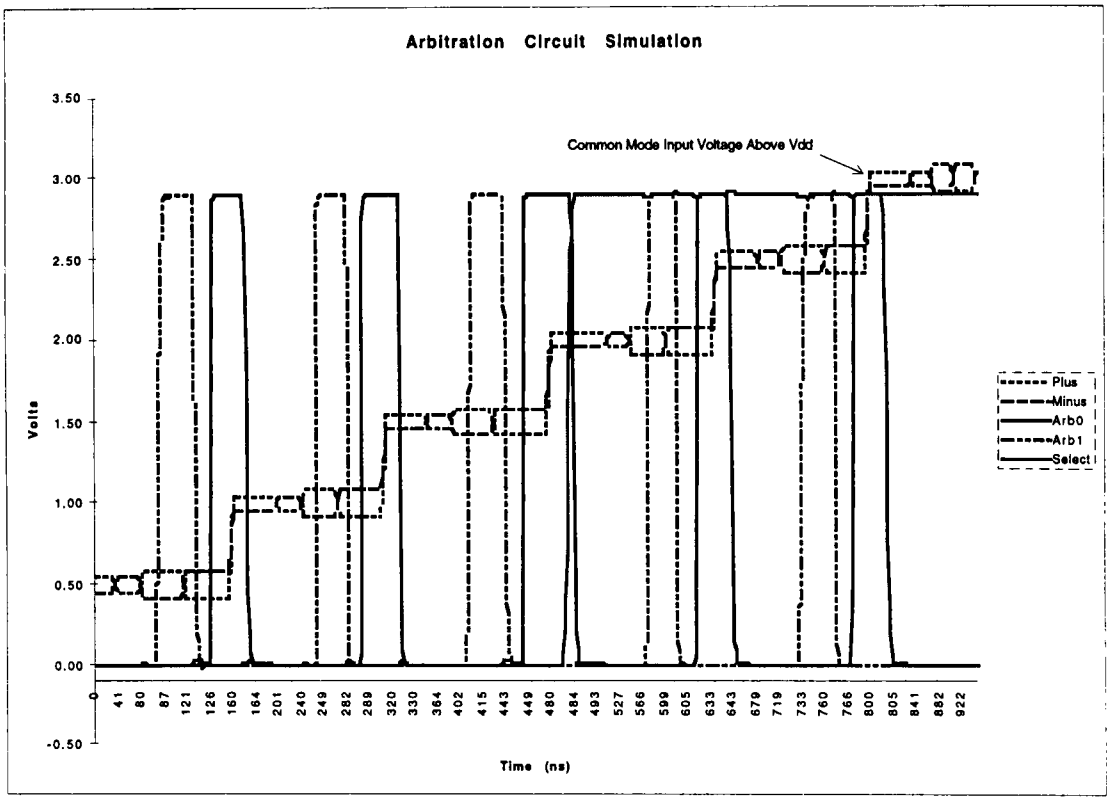


Figure 4.1.4 - Spice Arbitration Simulation With Low V_{dd}

The block diagram of the circuit used for this implementation is shown in Figure 4.2.1. The three stages of P-channel drivers have delayed inputs. This allows a rising edge with the specified 1.2ns rise time. This will also reduce over-shoot on the *tpa* or *tpab* outputs. Only one N-channel transistor is necessary to implement a smooth falling edge. This is due to the smaller active area required by an N-channel to reach the same current drive capability as a larger P-channel transistor. The schematic for the CMOS circuit used for this implementation is given in Appendix as the Data and Strobe Driver Schematic. The delay paths are implemented with a combination of inverters and capacitors, although most of the delay is due to the inverters. The capacitors are P-channel transistors that have their source and drain nodes connected to *tpa* or

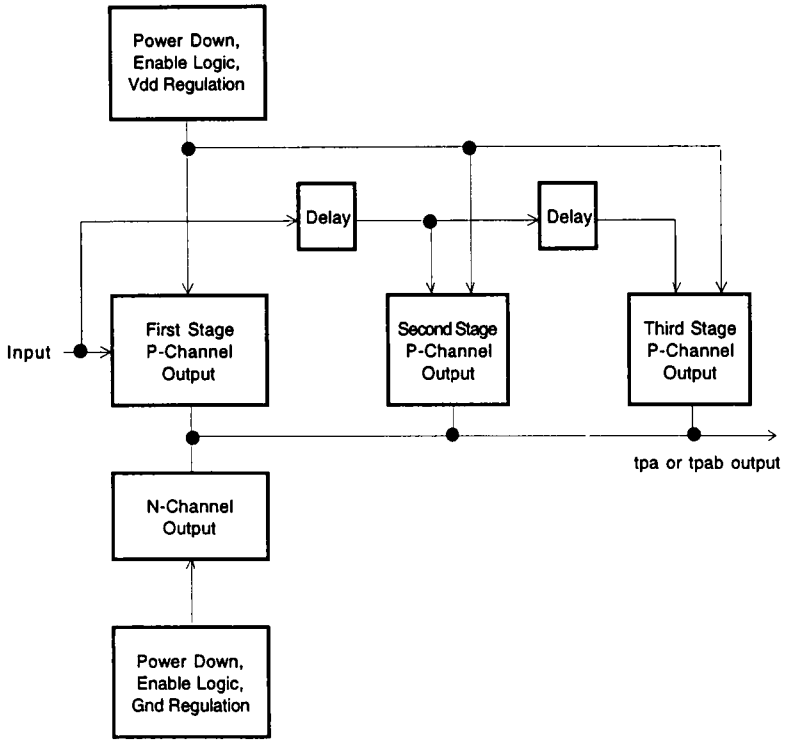


Figure 4.2.1 - Data / Strobe Driver Circuit Block Diagram

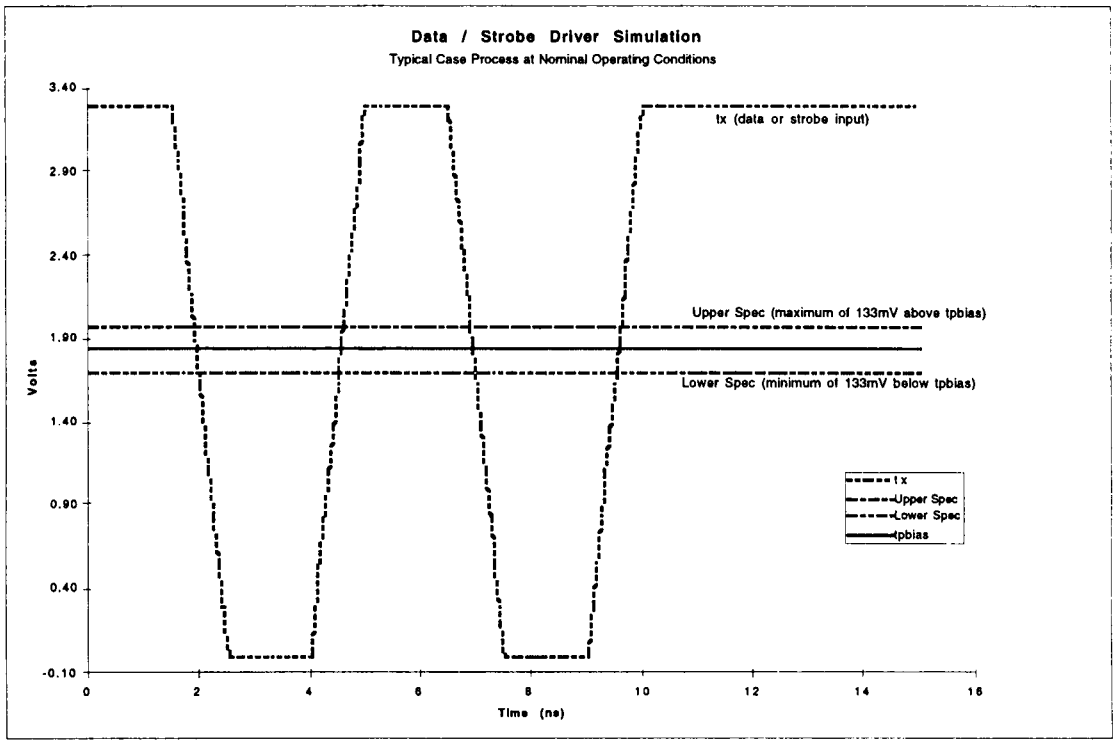


Figure 4.2.2 - Spice Driver Circuit Input Voltages

tpab. The gate of the transistor is connected to the delayed transmit signal. These capacitors also serve to dampen over-shoot. The 25μA current sources and sinks are provided by bias circuitry from the Reference Current Amplifier subcircuit.

4.2.1 Test Case Description

The inputs to the driver circuit are the transmit, enable, and power down inputs. The enable (*en*) signal must be pulled to V_{dd} while the power down (*pd*) signal must be grounded to enable data to be transmitted. If either *en* is low or *pd* is high, V_{dd} and ground will be disconnected from the driver circuit. This will cause *tpa* and *tpab* to float to the voltage of *tpbias* through the external 55Ω transistor network. If the driver circuit is enabled, *tpa* and *tpab* would either rise to V_{dd} or fall to ground if the resistor network was not connected. This simulation includes the external resistor network.

The Spice control file is given in the Appendix as Data and Strobe Transmission Spice Control File. The input voltage levels are plotted Spice Control File. The input voltage levels are plotted in Figure 4.2.2. The input signal is applied to *tx*. It is a four bit pulse train with 1ns rise and fall times toggling at 400MHz. The effective bit time is 2.5ns. The levels of *tpbias* along with the maximum and minimum differential output levels are also shown on this plot. The enable and power down signals are tied to the state which allows the driver circuit to be enabled. The simulated outputs should match the requirements of Tables 2.4 and 2.11.

4.2.2 Process and Operating Parameter Simulation Matrix

The simulation matrix given in Table 4.1 was applied to the driver circuit. The *tx*, *tpa*, and *tpab* signals for the typical processing and nominal operating conditions are plotted in Figure 4.2.3. The *tpa* and *tpab* outputs easily meet the minimum requirements of a 1.2ns rise time and 265mV differential amplitude. These levels are simulated without the cable media interface and a connected port. The simulations with the cable and connected port in Chapter Five will check these levels under more stringent conditions. All of the simulations showed no degradation of performance across operating temperature or voltage assuming the worst-case process. The expected worst-case performance of the driver circuit would be at low V_{dd}, high temperature. The simulation results are plotted in Figure 4.2.4. Although the driver circuit outputs pass specifications, all cases showed a slight dip in the output rising edge. The dip becomes more pronounced at high temperatures. A closer look at selected internal signals in the simulation is plotted in Figure 4.2.5. Each step in *tpa* correlates to either a positive voltage being applied to a capacitor or a P-channel transistor being turned on. The first ramp in *tpa* is when signal *ppupos* applies positive voltage to capacitor I$273 (shown in the Data / Strobe Driver Circuit Schematic). Then after *tx* propagates in parallel through two inverters, *ppuposb* and *ppuposb2* turn on the first two P-channel transistors at the same time. This correlates with the largest ramp in the rising edge.

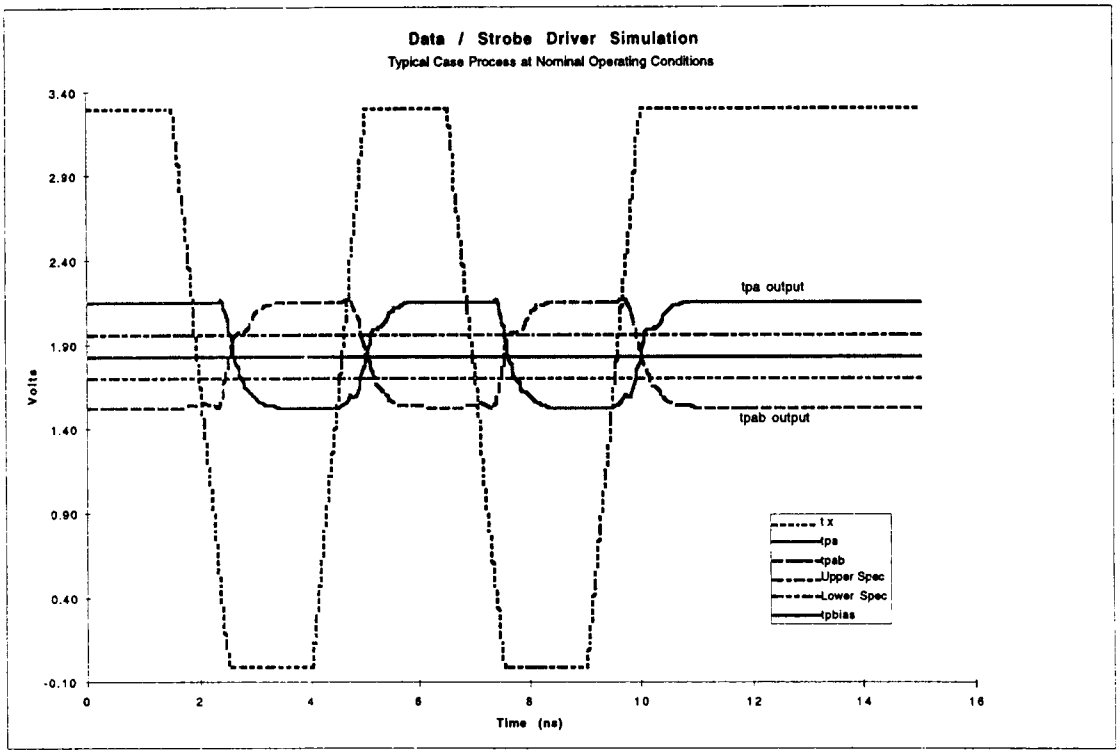


Figure 4.2.3 - Spice Data / Strobe Driver Simulation Results

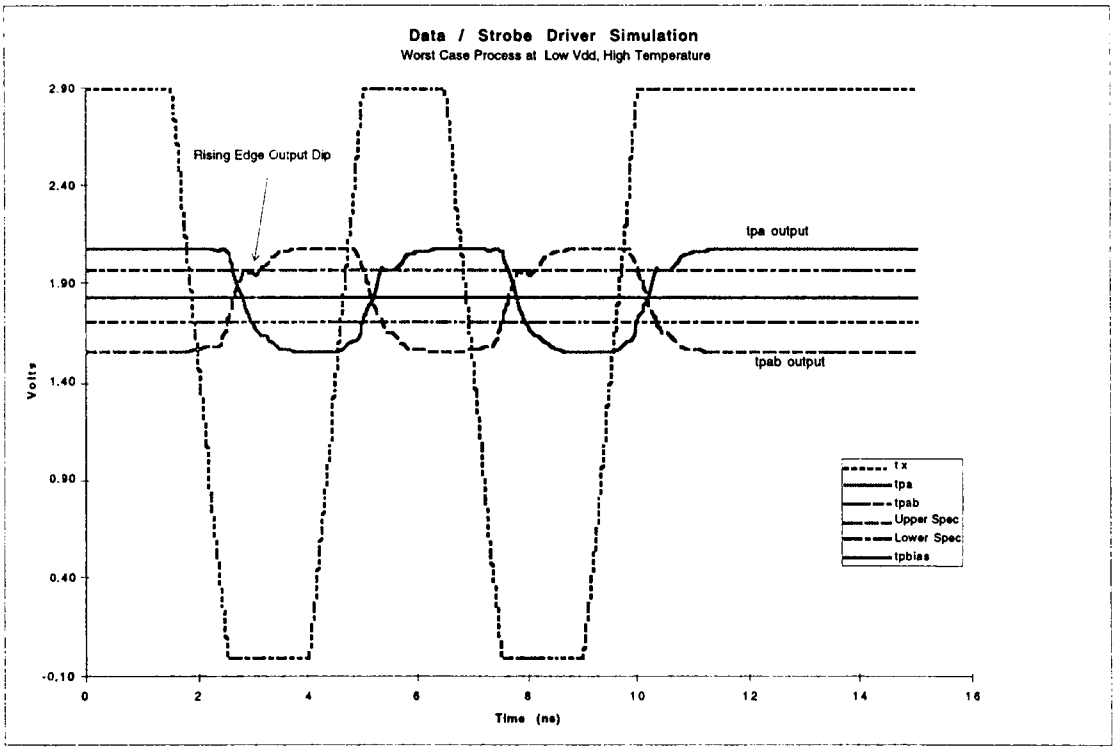


Figure 4.2.4 - Data / Strobe Driver Simulation With Low V_{dd}, High Temp

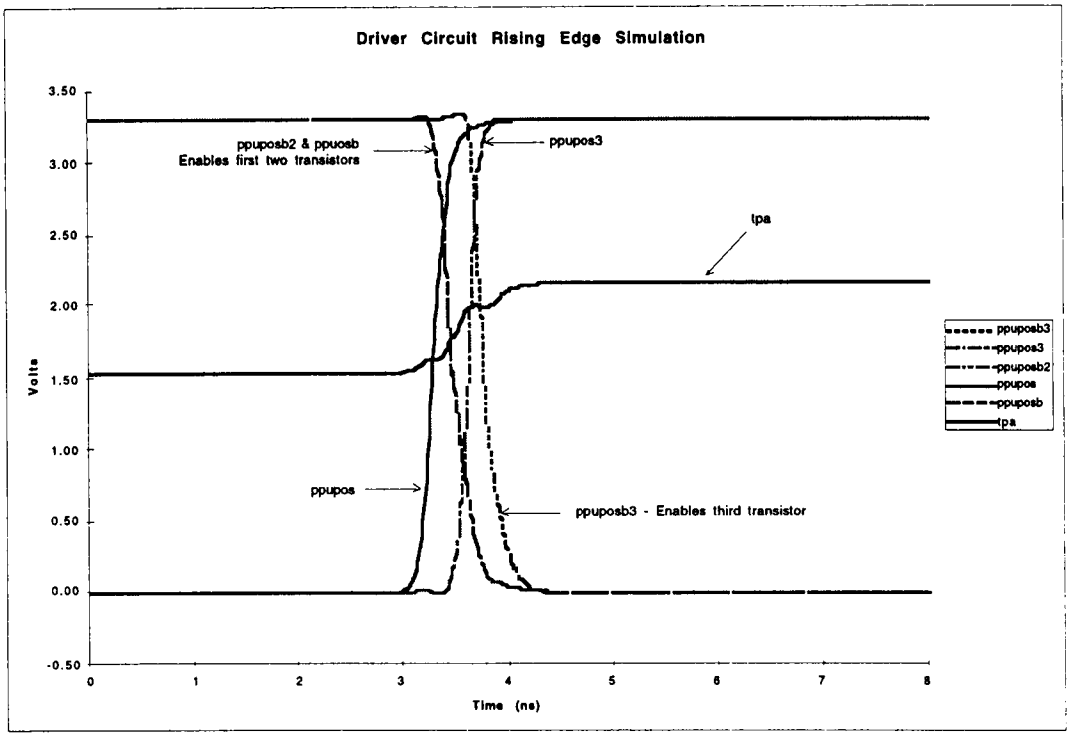


Figure 4.2.5 - Spice Data / Strobe Driver Rising Edge Simulation

After *tpa* reaches its maximum voltage using the first two P-channel transistors, *ppuposb3* turns on the third P-channel transistor, which raises the output to the final level above *tpbias*. This does not appear to be a smooth transition from a low to high state, but the amplitude from a low to high transition is less than 0.7V, and with only the external resistive load. Normal rising edges are from 10% to 90% of V_{dd}. The addition of the simulated cable and a connected port may also provide enough loading to make the transition smoother. This circuit must regulate the rising edge in a narrow amplitude range. This characteristic does not present a problem because the differential voltage of *tpa* and *tpab* determines the transmitted data or strobe.

4.3 Data and Strobe Receiver

The Data And Strobe Receiver Circuit is used to decode the TPA or TPB differential voltage inputs into a logic one or zero. The functional diagram for the Data And Strobe Receiver Circuit is included in the Electrical Media Interface in Figure 2.3. The differential inputs must meet the minimum specified requirements in Tables 2.5 and 2.6. The block diagram for the receiver is given in Figure 4.3.1. The *plus* and *minus* terminals are connected to TPA or TPB. The differential input voltage is applied to the dual output differential amplifier. The purpose of this stage is to eliminate the common mode voltage component of the input voltage. It has a positive and negative output which is fed into the single output differential amplifier. This second stage converts the differential output from the first stage into a single

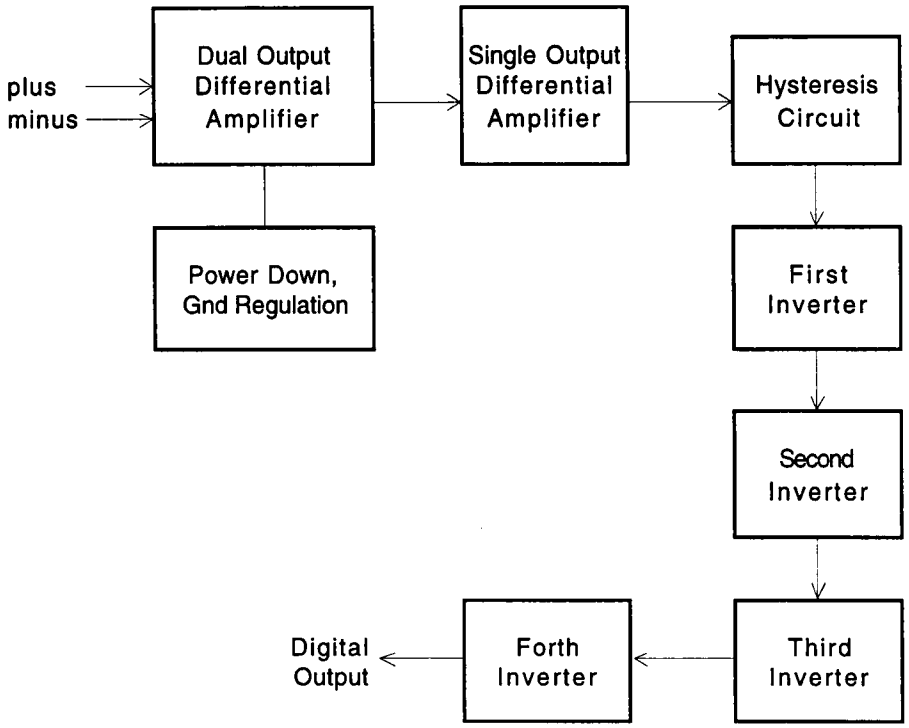


Figure 4.3.1 - Data And Strobe Receiver Circuit Block Diagram

ended output and provides most of the amplification. For example, if the differential voltage is a positive 200mV, the output should be high. if the differential voltage is negative, the output will be low. The output from the second stage is fed into a hysteresis circuit. This third stage provides feed back to the output of the second stage that allows the second stage output voltage to settle closer to V_{dd} or ground. The four inverters are designed to increase the digital output drive capability. Each inverter is larger than the next so that input and output capacitances can be overcome at high-speed operation. The schematic for the CMOS circuit used for this implementation is given in the Appendix as the Data and Strobe Receiver Schematic. The outputs of the first stage are *a* and *b*. They drive into the second stage, which as only *b2* as an output. The output *b2* is used to drive the input to an inverter. This inverter is part of the hysteresis circuit and consist of transistors *p3* and *n3*. The output from this inverter, *b3*, is used as feedback to drive transistors *pfdbk* and *nfdbk*. These transistors source and sink the current needed to complete the hysteresis. The output *b2* is also used to drive the first of four inverters. The digital output of the circuit is denoted as *out*.

4.3.1 Test Case Description

The inputs to the Data and Strobe Receiver circuit are either TPA or TPB. They are connected to the *plus* and *minus* signals on each comparator. The differential amplitudes generated by TPA or TPB will meet the requirements of Table 2.2.

The Spice control file is given in the Appendix as Data and Strobe Reception Spice Control File. The spice simulation input voltages are plotted in Figure 4.3.2. The differential voltages test the minimum and maximum differential amplitude specifications. They are designed to simulate 400MHz operation. The input common mode voltage starts at 0.5V and steps up by 0.5V after the four differential amplitude cases are executed. The final common mode voltage is 3.0V. This is above the specified range of 2.515V, but it is included here to test for the range of the circuit operation. The power down signal is pulled low while the *src25* signal is connected to a 25μA current source during the simulation.

4.3.2 Process and Operating Parameter Simulation Matrix

The simulation matrix given in Table 4.1 was applied to the Data And Strobe Receiver Circuit. The input and output voltage levels for typical processing and nominal operating conditions are plotted in Figure 4.3.3. The digital output begins to operate correctly when the common mode voltage is between 0.5V and 1.0V. The simulation was repeated at all corners of the matrix. The only simulations with failing results was when the operating voltage was below the common mode input voltage. This is when V_{dd} is 2.9V while the common mode voltage was 3.0V. The maximum specified input common mode voltage for S400 speed is 0.523V. The simulation was run again with one difference.

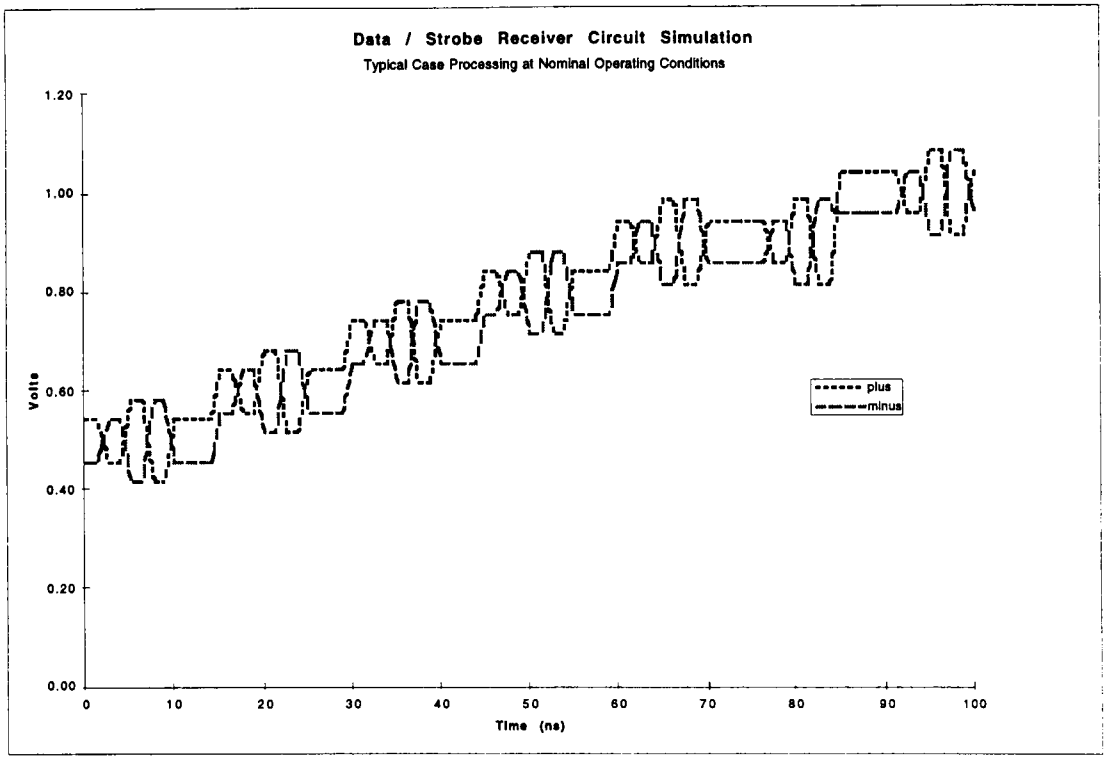


Figure 4.3.2 - Spice Data And Strobe Receiver Input Voltages

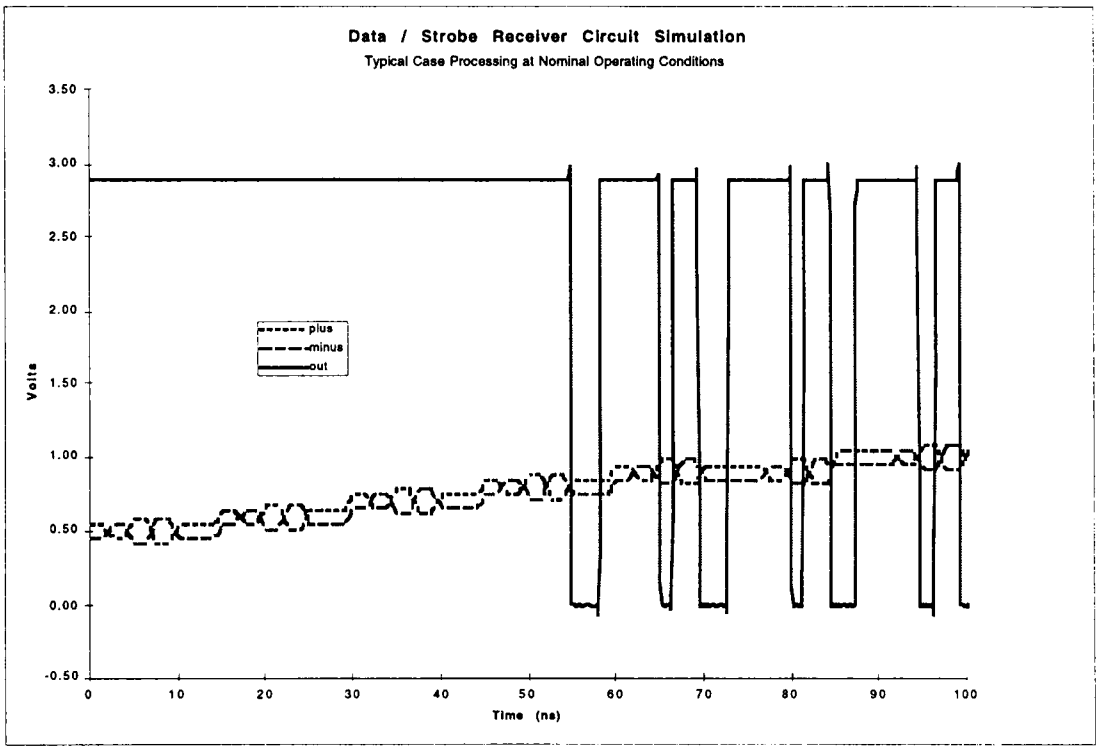


Figure 4.3.3 - Spice Data And Strobe Receiver Simulation Results

The common mode input voltages were changed to increase by only 0.1V instead of 0.5V. The purpose of this was to determine lower end of the operating range of the receiver circuit. The plot of one output is given in Figure 4.3.4. In each case of process, temperature, and operating voltage, the circuit does not function reliably until the common mode voltage is a minimum of 0.8V. This is because the to the first stage differential inputs are N-channel transistors. The threshold of these transistors is between 0.7V and 0.8V. Even if the inputs were P-channel transistors with a threshold voltage of 0.0V, the margin would still be less than a volt. The circuit does function correctly in higher ranges of common mode voltage. This circuit does not pass the minimum common mode input voltage specification of 0.523V.

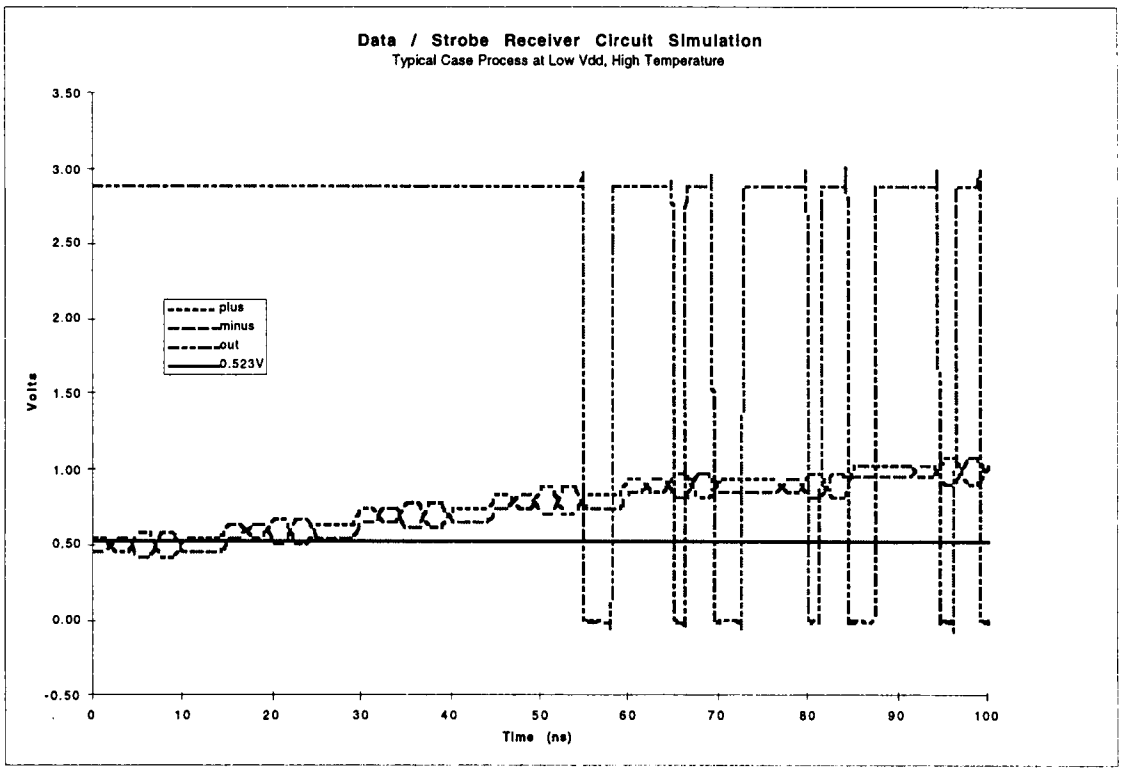


Figure 4.3.4 - Spice Data And Strobe Receiver Simulation

CHAPTER 5. PHYSICAL PORT MIXED MODE SIMULATION AND POWER ANALYSIS

The simulation of the electrical media interface circuit is mixed mode because the inputs and outputs to the complete circuit are digital signals while the analog signals are internal to the port and from one port to another across the cable media interface. The model for port-to-port simulation is given in Figure 5.1. A description of each pin is available in Table 2.3. The simulation inputs will be designed to verify port status, speed signaling, arbitration, and data and strobe transmission and reception. Each simulation will follow the matrix given in Table 4.1. The operating conditions and processing parameters for the matrix are given in Tables 4.2, 4.3, and 4.4. The output from each simulation will be compared to the IEEE 1394-1995 specification to verify correct functionality. The power dissipation analysis will consider typical and worst case processing at nominal and extreme operating conditions.

5.1 Data and Strobe Transmission and Reception

Each port in Figure 5.1 is identical. Therefore, when sending data from one port to the other, transmission and reception is simulated during the same simulation. This reduces the number of test cases needed for a complete simulation. Port#1 will transmit a strobe from TPA into the cable media. Port#2 will receive the strobe through TPB, which is connected to the same cable media as TPA on Port#1. Port#1 will send data through TPB to the cable media.

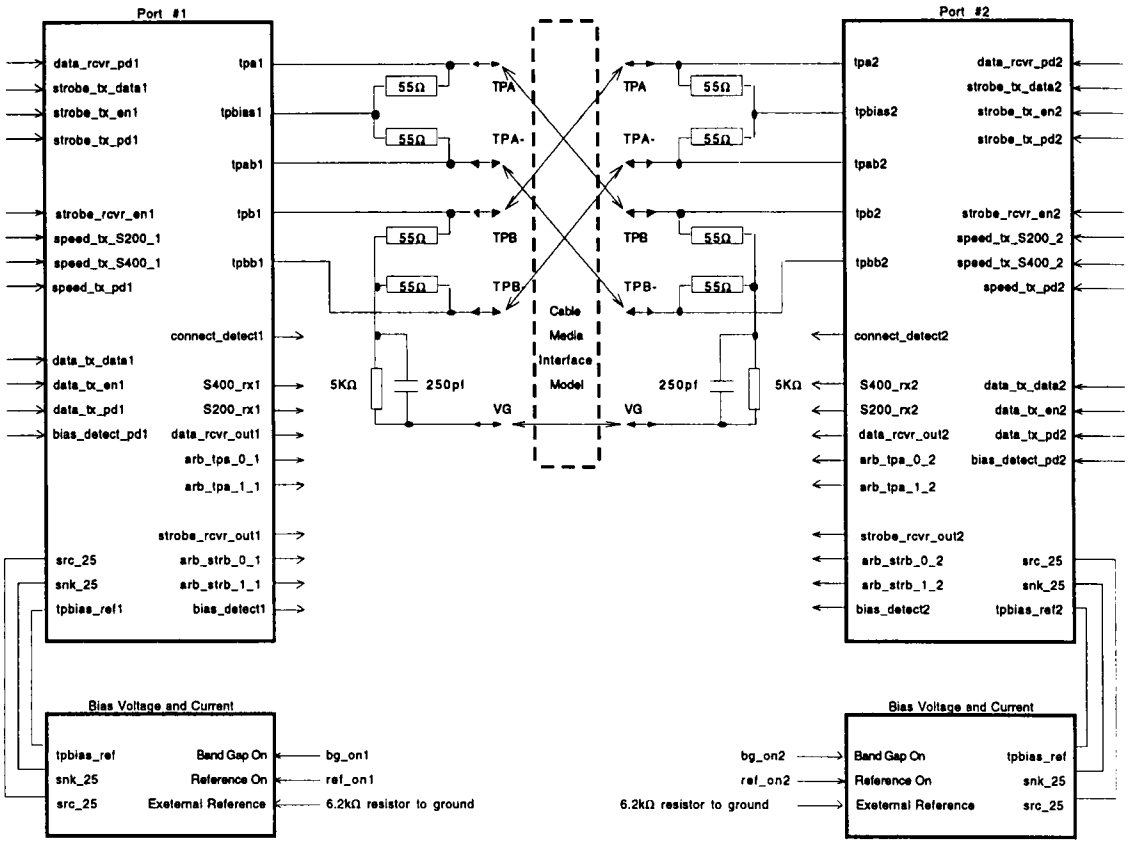


Figure 5.1 - Port-To-Port Simulation Model

Port#2 will receive the data through TPA, which is connected to the same cable media as TPB on Port#1.

5.1.1 Data and Strobe Transmission and Reception Test Case Description

For data and strobe transmission and reception, the first step is to enable or disable the necessary subcircuits in the port. This is accomplished by biasing or grounding certain control lines as denoted in Table 5.1.1. This configuration powers up and enables the transmission circuits in Port#1. All other functions are disabled in Port#1. Port#2 has only the reception circuits enabled and powered up.

Table 5.1.1 - Data and Strobe Enable and Control Signals

Signal	**Bias**	**Port**	**Description**
data_rcvr_pd1	V_{dd}	1	Data receiver power down
strobe_tx_en1	V_{dd}	1	Strobe data enable
strobe_tx_pd1	Gnd	1	Strobe transmit power down
strobe_rcvr_en1	V_{dd}	1	Strobe receiver enable
data_tx_en1	V_{dd}	1	Data transmit enable
data_tx_pd1	Gnd	1	Data transmit power down
bias_detect_pd1	V_{dd}	1	Port status detection power down
speed_tx_pd1	V_{dd}	1	Speed transmit power down
bg_on1	V_{dd}	1	Band gap starter
ref_on1	V_{dd}	1	Reference voltage
data_rcvr_pd2	Gnd	2	Data receiver power down
strobe_tx_en2	Gnd	2	Strobe data enable
strobe_tx_pd2	V_{dd}	2	Strobe transmit power down
strobe_rcvr_en2	Gnd	2	Strobe receiver enable
data_tx_en2	Gnd	2	Data transmit enable
data_tx_pd2	V_{dd}	2	Data transmit power down
bias_detect_pd2	V_{dd}	2	Port status detection power down
speed_tx_pd2	V_{dd}	2	Speed transmit power down
bg_on2	V_{dd}	2	Band gap starter
ref_on1	V_{dd}	2	Reference voltage

Both ports have the voltage and current generators enabled by V_{dd} to the *bg_on* and *ref_on* control lines. The *rext* input is connected to a 6.2kΩ

resistor which is grounded at the other end. Data and strobe transmission can now be accomplished by toggling the *data_tx_data1* and *strobe_tx_data1* digital inputs on Port#1. The data will be output from the *data_rcvr_out2* and *strobe_rcvr_out2* signals on Port#2. The enable and power down control signals as well as the data and strobe transmission inputs are controlled by the Spice control file given in the Appendix as the Port-to-Port Data and Strobe Transmission and Reception Spice Control File. The data and strobe inputs are plotted in Figure 5.1.1. The input signal represents a short 400MHz burst. The rise times are set to only one nanosecond. The pulse duration is only 2.5ns at 400MHz.

5.1.2 Data and Strobe Process and Operating Parameter Simulation Matrix

The simulation was executed across the process and operating conditions matrix. The strobe and data input signals on Port#1 were toggled. The strobe and data signals were recieved correctly and converted back into digital signals by Port#2. The plot of the input and output signals are given in Figure 5.1.2. The input signal resembles a sine wave at this high of a frequency. The plots of the input signals *data_tx_data1* and *strobe_tx_data1* to Port#1 are the same, hence the appearance of one waveform. The output from Port#2 will also be the same for both *data_rcvr_out2* and *strobe_rcvr_out2*. Notice that there is approximately a 23ns delay from the input to output. This is due to the delay induced in the cable media model as discussed in Chapter Three.

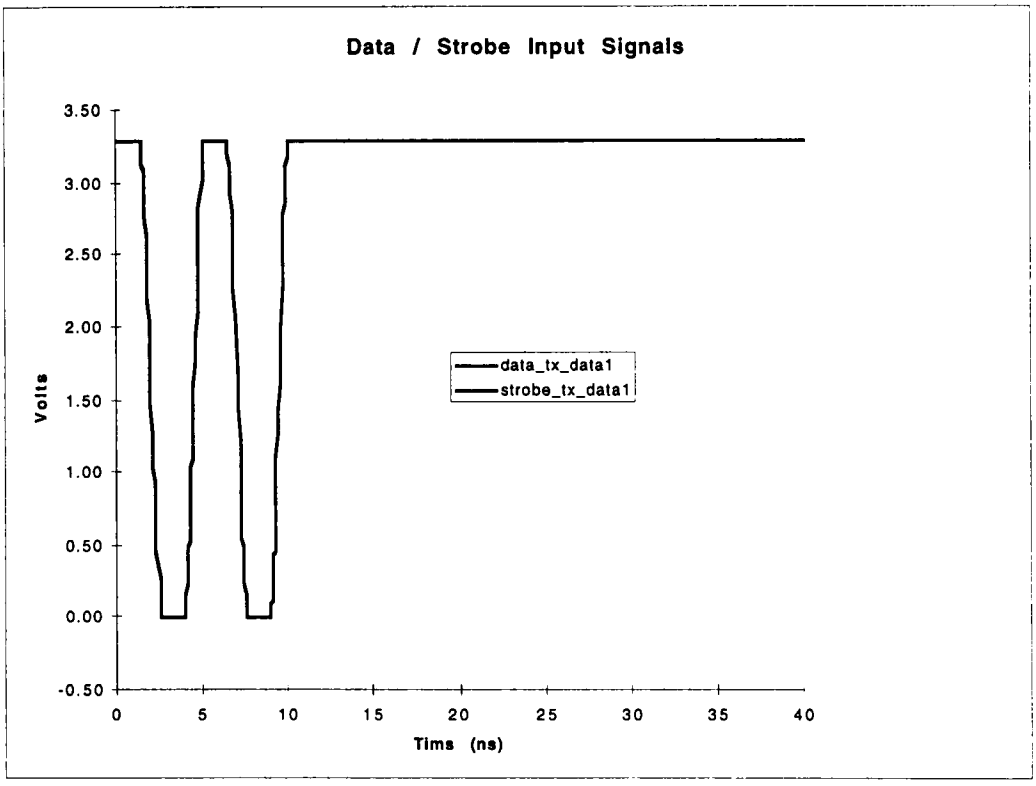


Figure 5.1.1 - Spice Data/Strobe Driver Circuit Input Signals

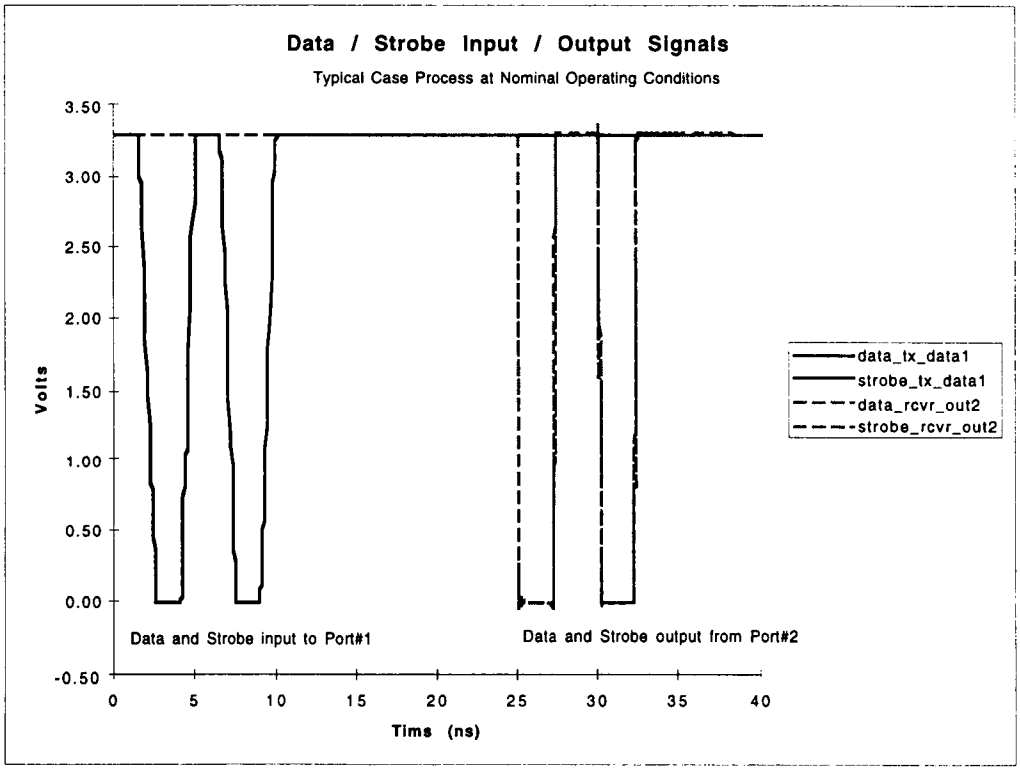


Figure 5.1.2 - Data And Strobe Input and Output Signals

The plot of the analog signals through the cable media are given in Figure 5.1.3. Here the signal propagation delay through the cable is observed by the delay from *tpa1* and *tpab1* modulating to *tpb2*and *tpbb2* modulating. Notice that the delay from the digital inputs *strobe_tx_data1* to *tpa1* and *tpab1* modulating is very short. Also notice that the delay from when the data passes from the cable to digital output conversion is also very short. This is observed by delay from *tpb2* and *tpbb2* modulating to the *strobe_rcvr_out2* output. A closer look at the analog signals is given in Figure 5.1.4. Notice that the outputs *tpb2* and *tpbb2* look like sine waves during modulation. This is the result of the 400MHz burst input. The output appears to be severely attenuated, but the differential voltage is still above the minimum specified 168mV and below the maximum of 265mV. The *tpa1* and *tpab1* rise and fall times are less than 1ns, which passes the maximum 1.2ns specification. The data and strobe input and output circuits function correctly during port to port communications and meet IEEE 1394-1995 specifications. The *tpbias1* reference voltage remains constant at 1.84V during the simulation.

One problem with the simulation is that the input signals *tpa1* and *tpab1* are not switching as far in one direction as they are in the opposite direction. The *tpa1* signal starts at 2.09V and drops only to 1.75V instead of 1.64V. The *tpab1* signal starts at 1.64V but rises to only 1.97V instead of 2.09V. Simulation of port-to-port communication without the cable model showed that the cable model impedes DC performance, but works well at AC. The output waveforms from the cable

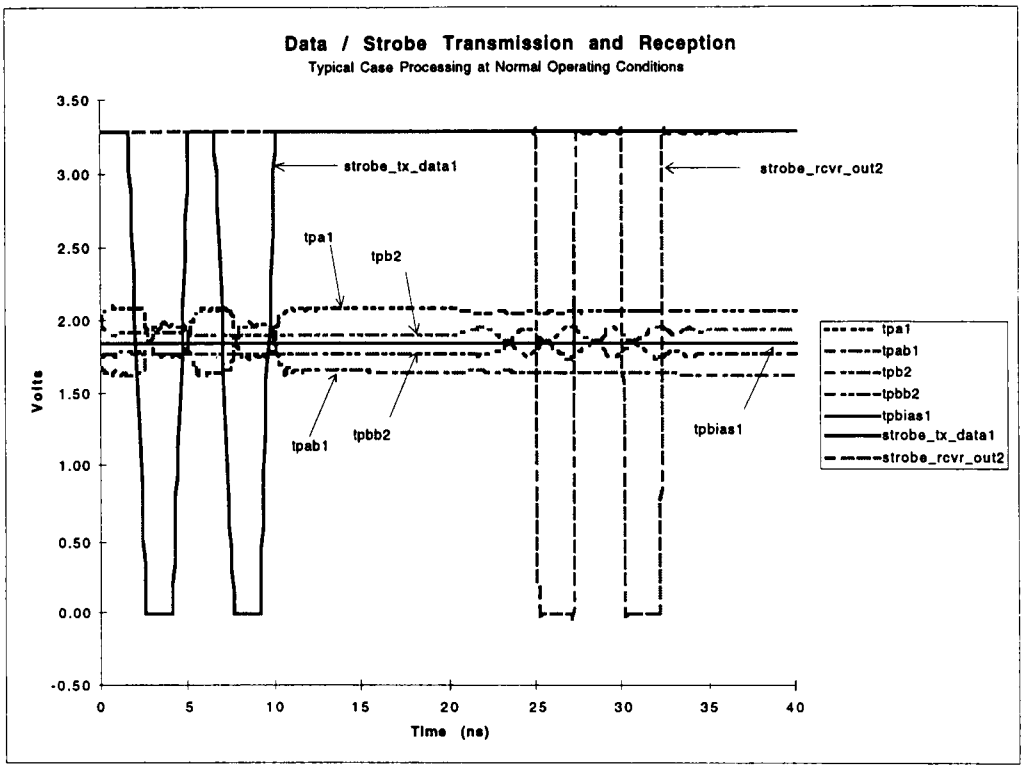


Figure 5.1.3 - Data And Strobe Signal Propagation

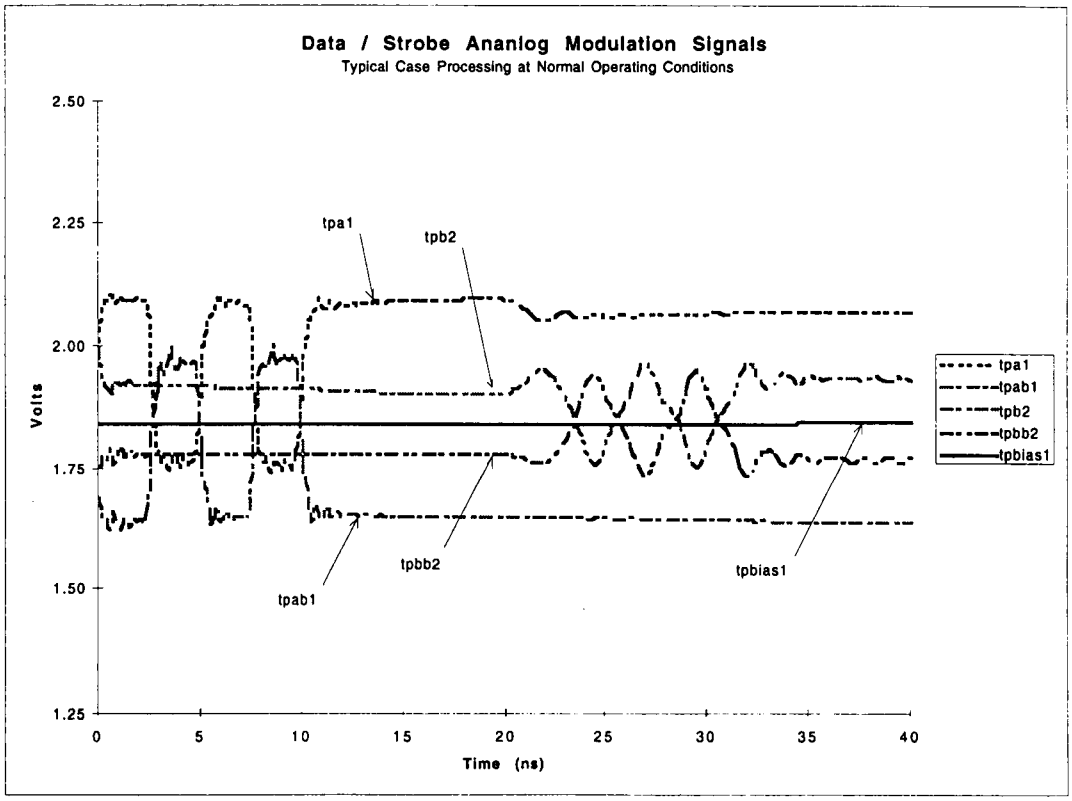


Figure 5.1.4 - Analog Data / Strobe Signals

model are degraded as expected and resemble a real cable at high frequency. However, any voltage connected to the *tpab* or *tpbb* terminals on the cable model will not rise to the same level as *tpa* or *tpab*. The same is true for *tpa* and *tpb* not falling to the same input as *tpab* and *tpbb*. This does not affect data and strobe transmission simulation because the levels are still slightly above the maximum differential input level of 265mv.

The simulation showed the same results for the typical and worst case processing across operating voltage and temperature. The assumed worst case condition is worst case processing at low V_{dd} and high temperature. The results are plotted in Figure 5.1.5. The data and strobe outputs from Port#2 correspond directly to the inputs at Port#1. The analog outputs from Port#1 (*tpa1* and *tpab1*) are slightly degraded from the typical processing nominal operating conditions case. The waveforms still meet IEEE 1394-1995 specifications. The digital outputs from Port#2 also appear to be less sharp at the start or end of transitions, but the output data matches the data input to Port#1. A closer look at the analog modulation is given in Figure 5.1.6. This output shows the same characteristics as the normal case, but with less swing on the inputs to the cable. This is due to current drive reduction at high temperatures and low V_{dd} voltages for CMOS technology. The results of the simulations show port-to-port communication is robust with variances in process and temperature.

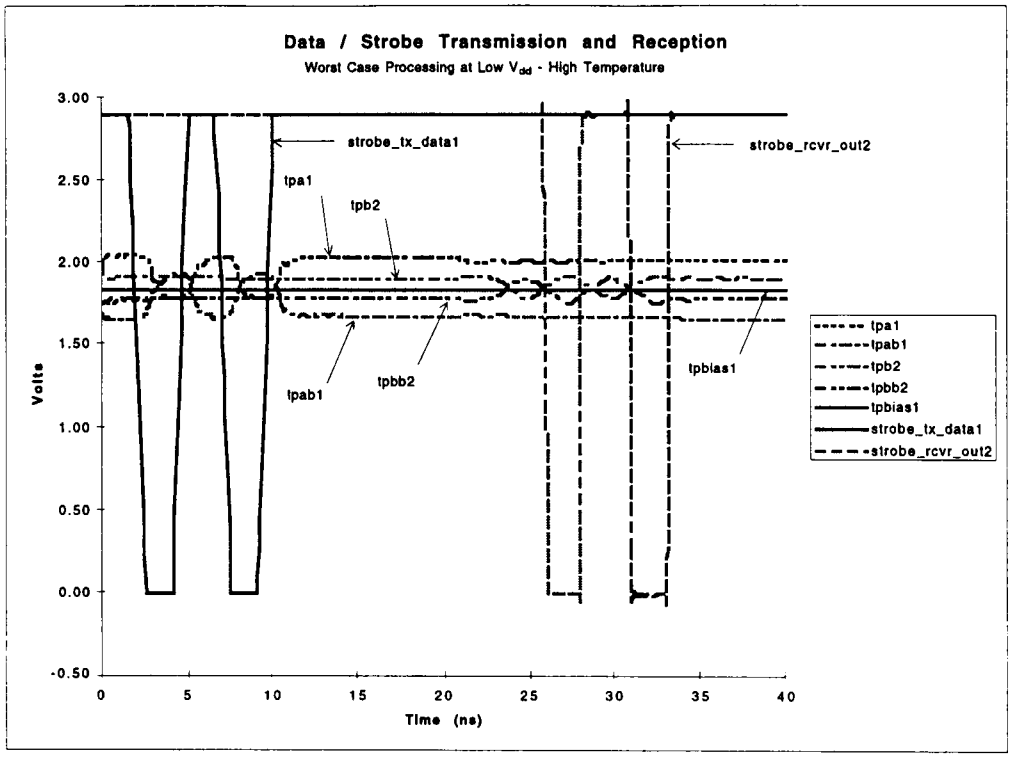


Figure 5.1.5 - Worst Case Data And Strobe Transmission And Reception

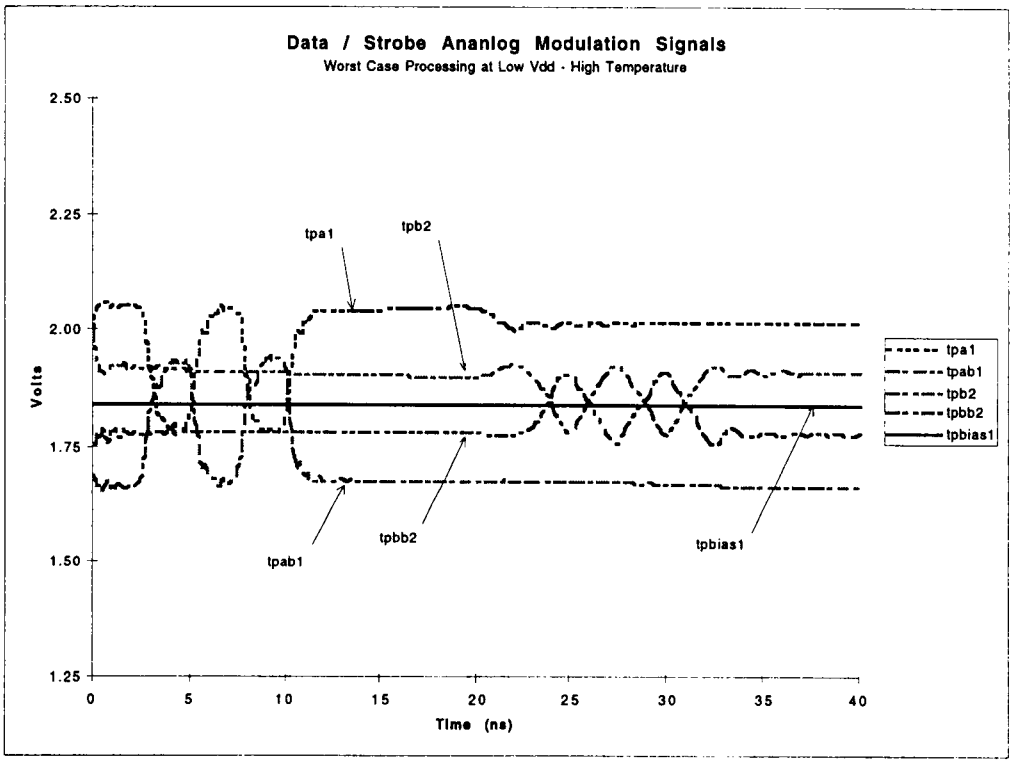


Figure 5.1.6 - Worst Case Analog Data / Strobe Signals

5.2 Arbitration

The specifications and operation of arbitration for the IEEE 1394-1995 high-speed serial bus is discussed in Section 2.2.4 and Section 4.1.Simulations will be executed in this section to verify port-to-port functionality and compliance with the IEEE 1394-1995 specification. The data and strobe transmit and the arbitration decoding subcircuits are the same for both ports, so only one set of test conditions are required to complete the simulation. Arbitration will be simulated by toggling *strobe_data_tx1* on Port#1 and *data_tx_data2* on Port#2. The *strobe_data_tx1* signal controls the *tpa1* and *tpab1* signals. The *tpa1* and *tpab1* signals are connected to the *tpb2* and *tpbb2* on Port#2 through the cable model. The outputs are *arb_tpa_0_1* and *arb_tpa_1_1* for Port#1, and *arb_strb_0_2* and *arb_strb_1_2* for Port#2.

5.2.1 Arbitration Test Case Description

For Arbitration simulation, the first step is to enable or disable the necessary subcircuits in the port. This is accomplished by biasing or grounding certain control lines as denoted in Table 5.2.1. This configuration powers up and enables the strobe transmission circuits in Port#1 and data transmission circuits in Port#2. All other functions are disabled in Port#1 and Port#2. Both ports have the voltage and current generators enabled by applying V_{dd} to the *bg_on* and *ref_on* control lines. The *rext* input is connected to a 6.2kΩ resistor which is grounded at the other end. The enable and power down control signals as well as the strobe and data transmission inputs are controlled by the Spice control file given in the Appendix as the Port-to-Port Arbitration

Table 5.2.1 - Arbitration Enable and Control Signals

Signal	Bias	Port	Description
data_rcvr_pd1	V_{dd}	1	Data receiver power down
strobe_tx_en1	V_{dd}	1	Strobe data enable
strobe_tx_pd1	Gnd	1	Strobe transmit power down
strobe_rcvr_en1	Gnd	1	Strobe receiver enable
data_tx_en1	Gnd	1	Data transmit enable
data_tx_pd1	V_{dd}	1	Data transmit power down
bias_detect_pd1	V_{dd}	1	Port status detection power down
speed_tx_pd1	V_{dd}	1	Speed transmit power down
bg_on1	V_{dd}	1	Band gap starter
ref_on1	V_{dd}	1	Reference voltage
data_rcvr_pd2	V_{dd}	2	Data receiver power down
strobe_tx_en2	Gnd	2	Strobe data enable
strobe_tx_pd2	V_{dd}	2	Strobe transmit power down
strobe_rcvr_en2	Gnd	2	Strobe receiver enable
data_tx_en2	V_{dd}	2	Data transmit enable
data_tx_pd2	Gnd	2	Data transmit power down
bias_detect_pd2	V_{dd}	2	Port status detection power down
speed_tx_pd2	V_{dd}	2	Speed transmit power down
bg_on2	V_{dd}	2	Band gap starter
ref_on1	V_{dd}	2	Reference voltage

Spice Control File. The data and strobe inputs are plotted in Figure 5.2.1. The inputs in Figure 5.2.1 should cause the differential voltage on TPA (Port#1) and TPB (Port#2) to fall within the ranges given in Table 4.1.1. The arbitration outputs should also match that of Table 4.1.1 where *arb0* from Table 4.1.1 corresponds with *arb_tpa_0_1* for Port#1 and *arb_strb_0_2* for Port#2. The output *arb1* corresponds with *arb_tpa_1_1* for Port#1 and *arb_strb_1_2* for Port#2.

5.2.2 Arbitration Process and Operating Parameter Simulation Matrix

The simulation was executed across the process and operating conditions matrix. The *strobe_data_tx1* and *data_tx_data2* input signals were toggled as specified in the test conditions. The plot of the input

signals and analog modulation are given in Figure 5.2.2. The same problem encountered with the cable model in Section 5.1 is causing a major problem with the Arbitration simulation. At DC operation, the cable model does not perform well. When the *strobe_data_tx1* input is

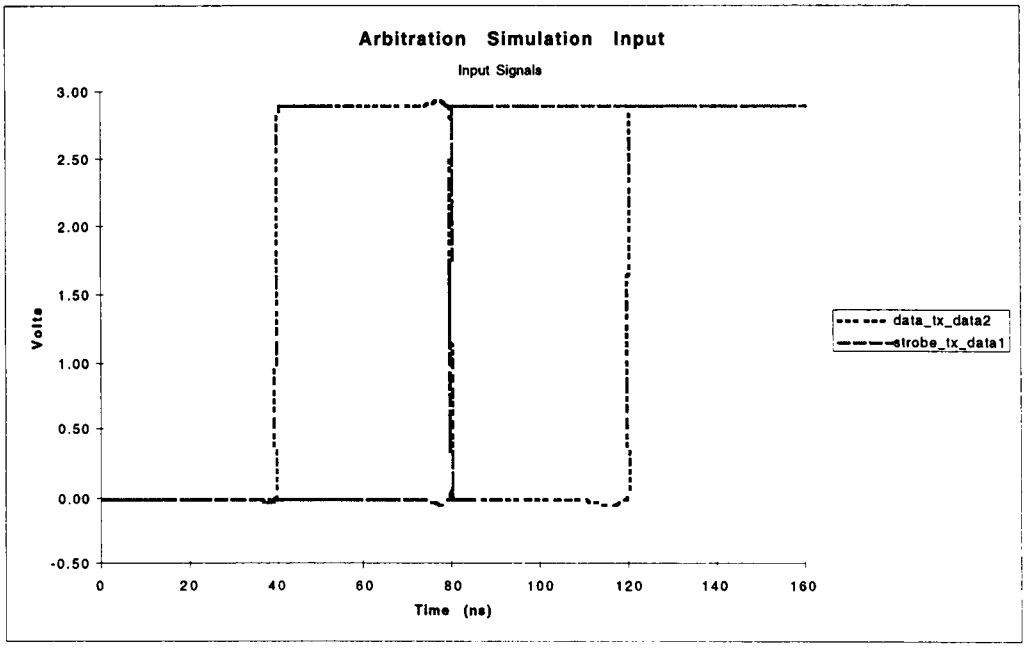


Figure 5.2.1 - Arbitration Simulation Input

high and the *data_tx_data2* input is low, the differential voltage on the *tpa1* and *tpab1* terminals should be close to zero. The same condition applies when the *strobe_data_tx1* input is low and the *data_tx_data2* input is high. Either of these input states would force one port to drive with a positive differential voltage while the other drives at a negative differential voltage as shown in Figure 2.11. The simulation output clearly shows the differential voltage is settling at +/-240mV instead of the specified +/-89mV for the Z state. The data in Figure 5.2.2 was generated with a simulation using a modified cable model to allow full voltage swings on *tpa1* and *tpab1* and allow more attenuation

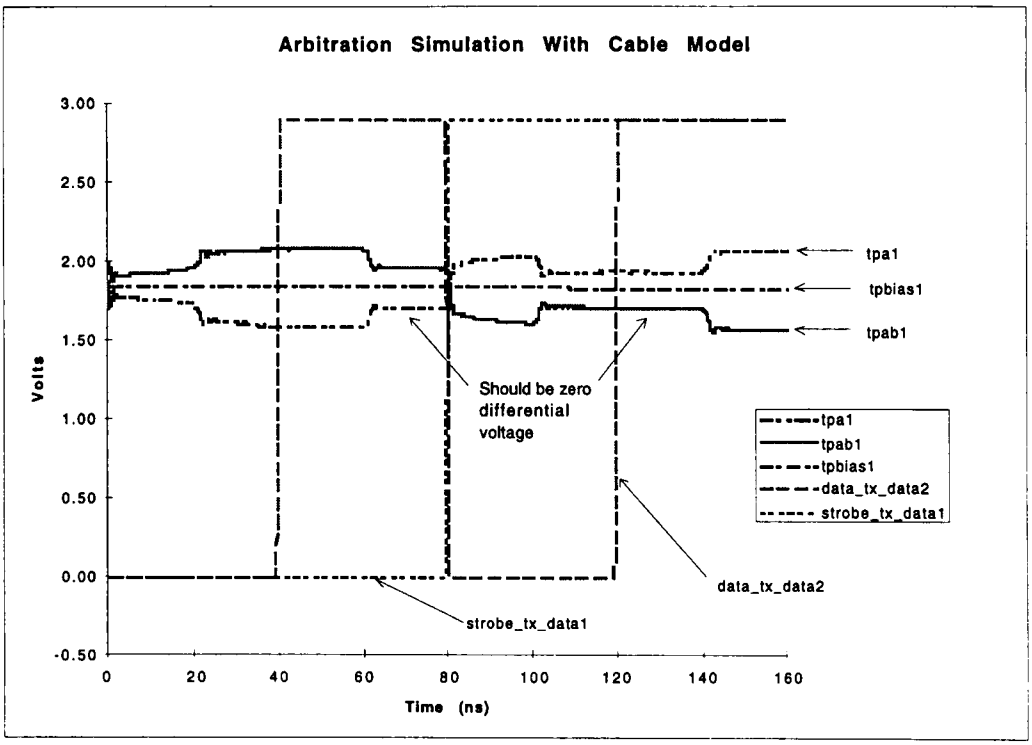


Figure 5.2.2 - Arbitration Simulation With Cable Model

to help with this problem, but the cable model is best suited for AC operation. A real cable will not force a positive voltage on a terminal when a negative voltage is connected on the other end. The simulation was modified to connect the ports directly without the cable to verify functionality. The simulation then assumes an idea cable with no voltage attenuation or current loss. The plot of the input signals and analog modulation are given in Figure 5.2.3. Notice that without the cable model the differential voltage reaches zero volts as required. The plot of the output signals and analog modulation are given in Figure 5.2.4. The Port#1 *arb_tpa_0_1* and *arb_tpa_1_1* digital outputs follow the negative, zero, or positive differential voltage as specified in Table 4.1.1. The Port#2 outputs also follow as specified but are not

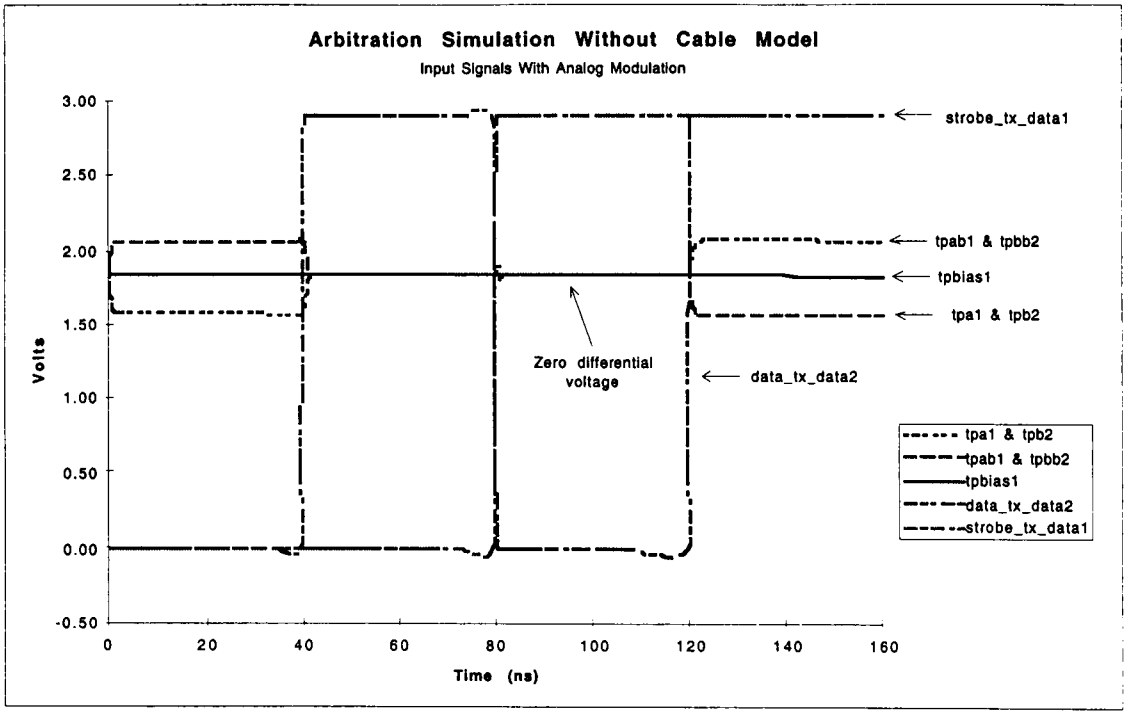


Figure 5.2.3 - Arbitration Simulation Without Cable Model

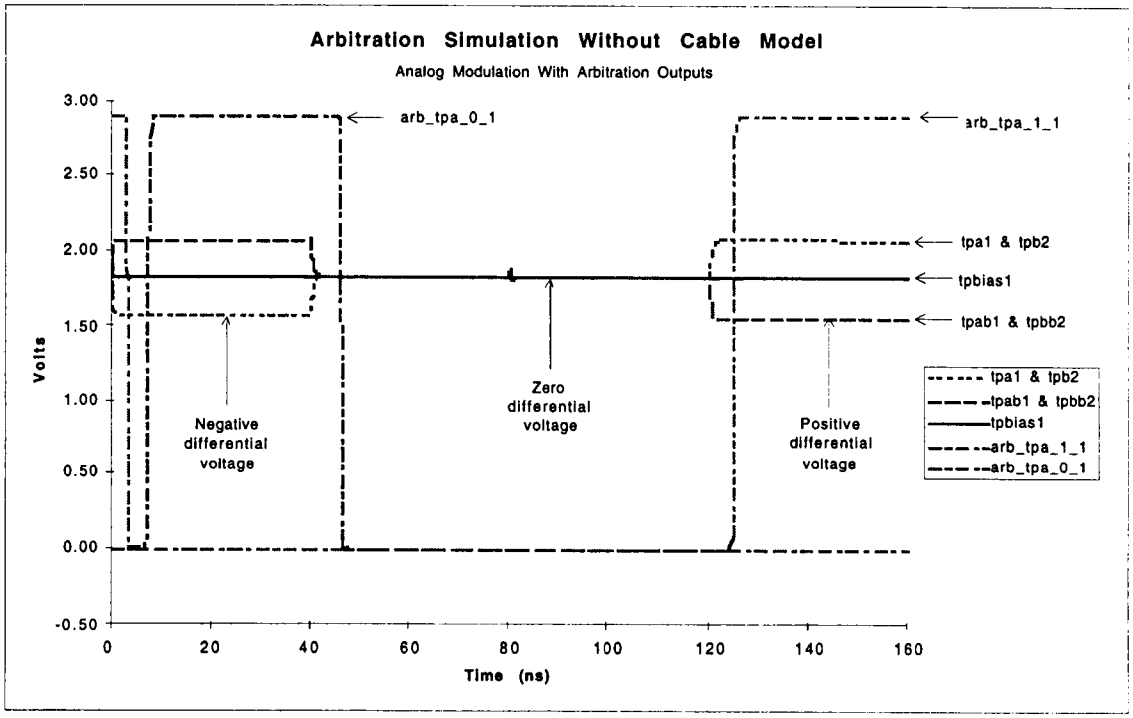


Figure 5.2.4 - Arbitration Simulation Output Without Cable Model

plotted here. All of the simulations across process and operating conditions showed the same robust operation. Port-to-port Arbitration simulation passes IEEE 1394-1995 specifications.

5.3 Speed Sensing

The specifications and operation of speed sensing for the IEEE 1394-1995 high-speed serial bus is discussed in Section 2.2. Simulation will be executed in this section to verify port-to-port speed signaling functionality and compliance with the IEEE 1394-1995 specification. The speed transmission and reception subcircuits are the same for both ports, so only one set of test conditions are required to complete the simulation. Speed signaling will be simulated by toggling the *speed_tx_s200_2* and *speed_tx_S400_2* inputs on Port#2 and examining the *s200_rx1* and *s400_rx1* outputs on Port#1.

5.3.1 Speed Sensing Test Case Description

For Speed Signaling simulation, the first step is to enable or disable the necessary subcircuits in the port. This is accomplished by biasing or grounding certain control lines as denoted in Table 5.3.1. This configuration disables all subcircuits in Port#1 and Port#2 except the speed transmission subcircuit in Port#2. Both ports have the voltage and current generators enabled by applying a V_{dd} to the *bg_on* and *ref_on* control lines. The *rext* input is connected to a 6.2kΩ resistor which is grounded at the other end. The enable and power down control signals as well as the speed transmission control signal inputs are controlled by the Spice control file given in the Appendix as the

Port-to-Port Speed Signaling Spice Control File. The speed transmission inputs are plotted in Figure 5.3.1. The inputs are very simple. The *speed_tx_s200_2* signal turns on transistors in the speed transmission

Table 5.3.1 - Speed Signaling Enable and Control Signals

Signal	Bias	Port	Description
data_rcvr_pd1	V_{dd}	1	Data receiver power down
strobe_tx_en1	V_{dd}	1	Strobe data enable
strobe_tx_pd1	V_{dd}	1	Strobe transmit power down
strobe_rcvr_en1	Gnd	1	Strobe receiver enable
data_tx_en1	Gnd	1	Data transmit enable
data_tx_pd1	V_{dd}	1	Data transmit power down
bias_detect_pd1	V_{dd}	1	Port status detection power down
speed_tx_pd1	V_{dd}	1	Speed transmit power down
bg_on1	V_{dd}	1	Band gap starter
ref_on1	V_{dd}	1	Reference voltage
data_rcvr_pd2	V_{dd}	2	Data receiver power down
strobe_tx_en2	V_{dd}	2	Strobe data enable
strobe_tx_pd2	V_{dd}	2	Strobe transmit power down
strobe_rcvr_en2	Gnd	2	Strobe receiver enable
data_tx_en2	V_{dd}	2	Data transmit enable
data_tx_pd2	V_{dd}	2	Data transmit power down
bias_detect_pd2	V_{dd}	2	Port status detection power down
speed_tx_pd2	Gnd	2	Speed transmit power down
bg_on2	V_{dd}	2	Band gap starter
ref_on1	V_{dd}	2	Reference voltage

circuit which sink current through TPB in Port#2. The *speed_tx_s400_2* turns on additional transistors to sink more current. The specified values of current that the speed transmission circuit must sink are given in Table 2.8.

5.3.2 Speed Sensing Process and Operating Parameter Simulation Matrix

The simulation was executed across the process and operating conditions matrix. The *speed_tx_s200_2* and *speed_tx_s400_2* were toggled as specified in the test conditions.

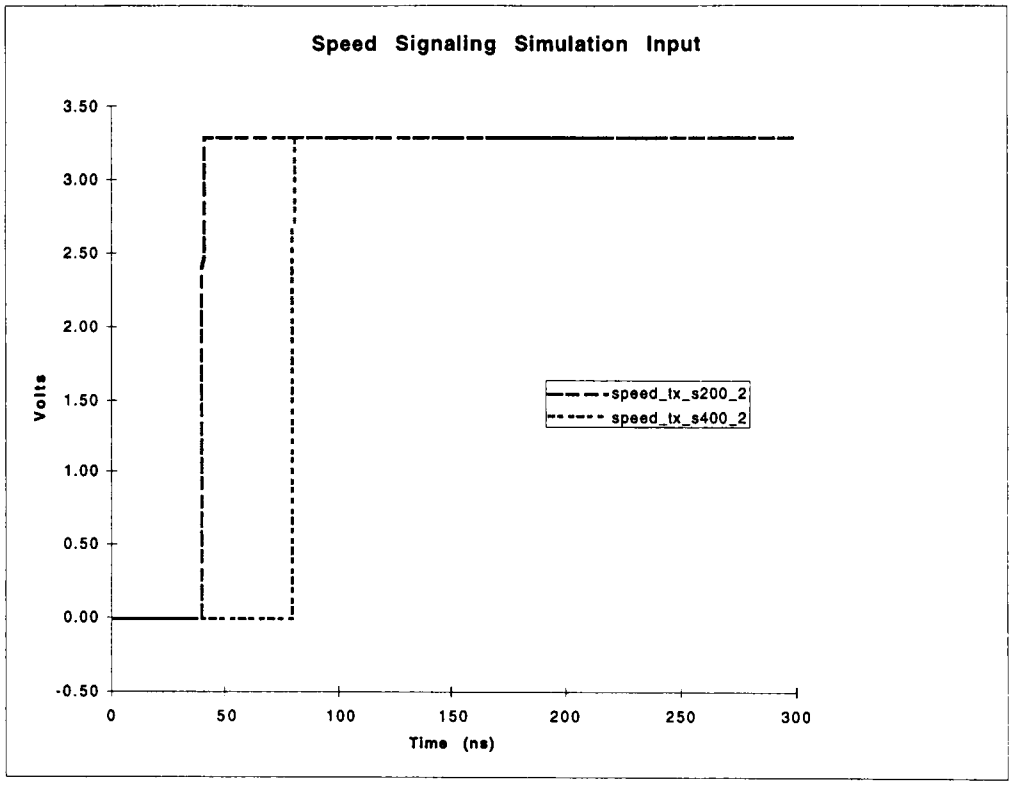


Figure 5.3.1 - Speed Signaling Simulation Input

The plot of the input signals and analog modulation are given in Figure 5.3.2. In this case cable model is performing well enough at DC to allow current to be drawn for proper speed signaling. The analog modulation signals and digital outputs from Port#1 are given in Figure 5.3.3. The Port#2 outputs *tpb2* and *tpbb2* correspond to the Port#2 inputs *speed_tx_s200_2* and *speed_tx_s400_2*. When *speed_tx_s200_2* is asserted high, the common mode voltage of *tpa1* and *tpab1* drops from the value of *tpbias1* (1.84V) to 1.75V. When *speed_tx_s400_2* is asserted high, the common mode voltage of *tpa1* and *tpab1* drops further to about 1.5V. Notice that the common mode voltage of *tpa1* and *tpab1* is pulled down by the current drain into TPB on Port#2. This difference in

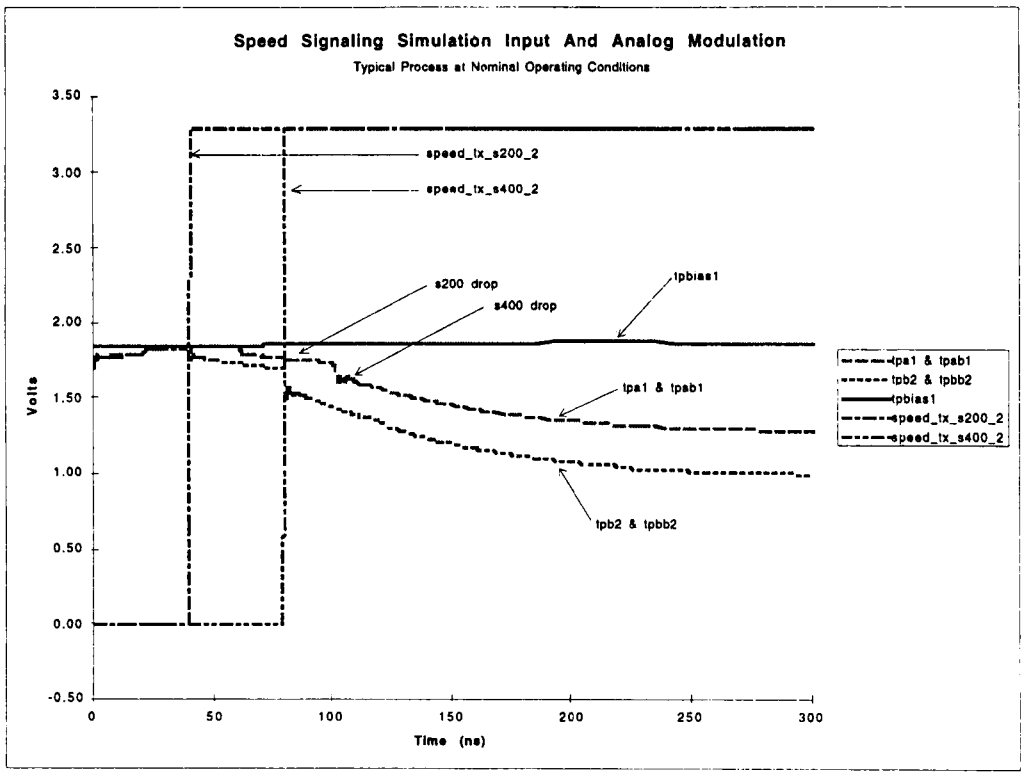


Figure 5.3.2 - Speed Signaling Simulation Input And Analog Modulation

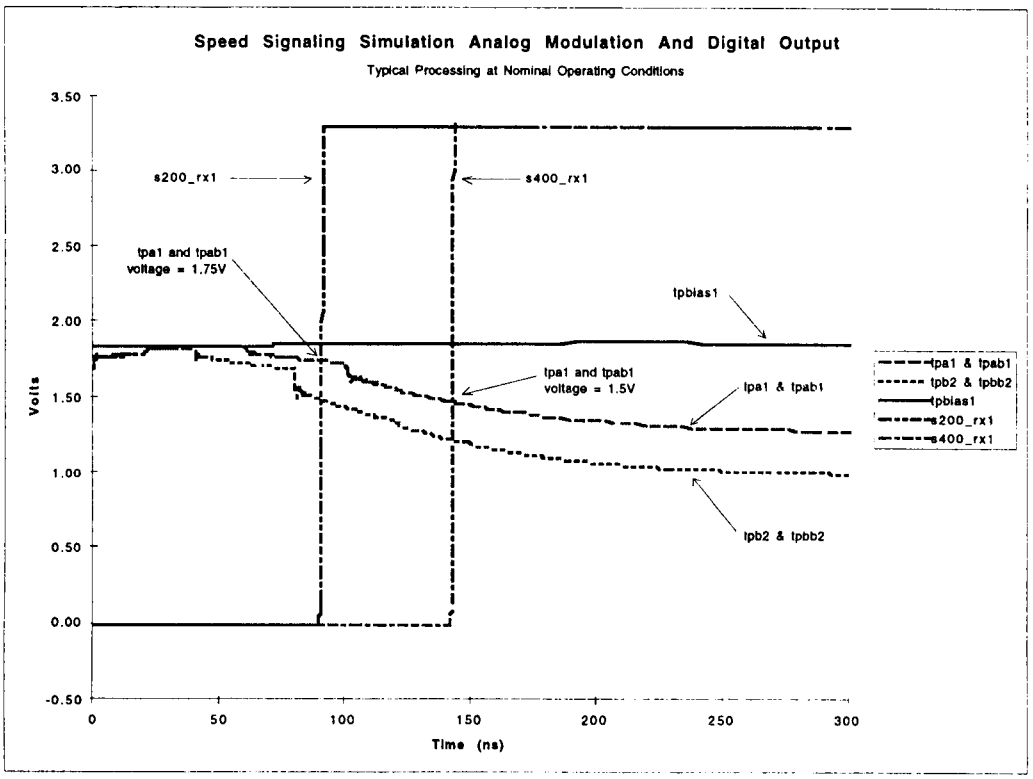


Figure 5.3.3 - Speed Signaling Analog Modulation And Digital Outputs

voltage is noted by *tpb2* and *tpbb2* being lower than *tpa1* and *tpab1*. The digital outputs show a slight time lag from the drops in common mode voltage, but the outputs are correct based on the digital inputs from Port#2. All of the simulations across process and operating conditions showed the same results. The specifications given in Table 2.8 require that speed signaling be implemented with current sinking from TPA through TPB. A check of the current required to interpret 200MHz or 400MHz operation is given in Figure 5.3.4. A quick check of the current sink required to indicate 200MHz or 400MHz speed signaling shows that the *S200rx* signal is asserted with 3.4mA and that the *S400rx* signal is asserted with 8.5mA. This meets the ranges specified in Table 2.8. Port-to-port Speed Signaling simulation passes IEEE 1394-1995 specifications.

5.4 Port Status

The specifications and operation of detecting a connected port for the IEEE 1394-1995 high-speed serial bus is discussed in Section 2.2.2. Simulation will be executed in this section to verify that a connected or disconnected port can be detected as required by the IEEE 1394-1995 specification. There are no controlled inputs to determine if a port is connected. The control bias voltages are given in Table 5.3.2. The only output is the *bias_detect* signal. The *bias_detect* signal is processed through the TPB circuit. If a port is connected, the voltage on the *bias_detect* signal will reach V_{dd}. The voltage on the *tpb* and *tpbb* will follow the *tpbias* signal from TPA on the connected port.

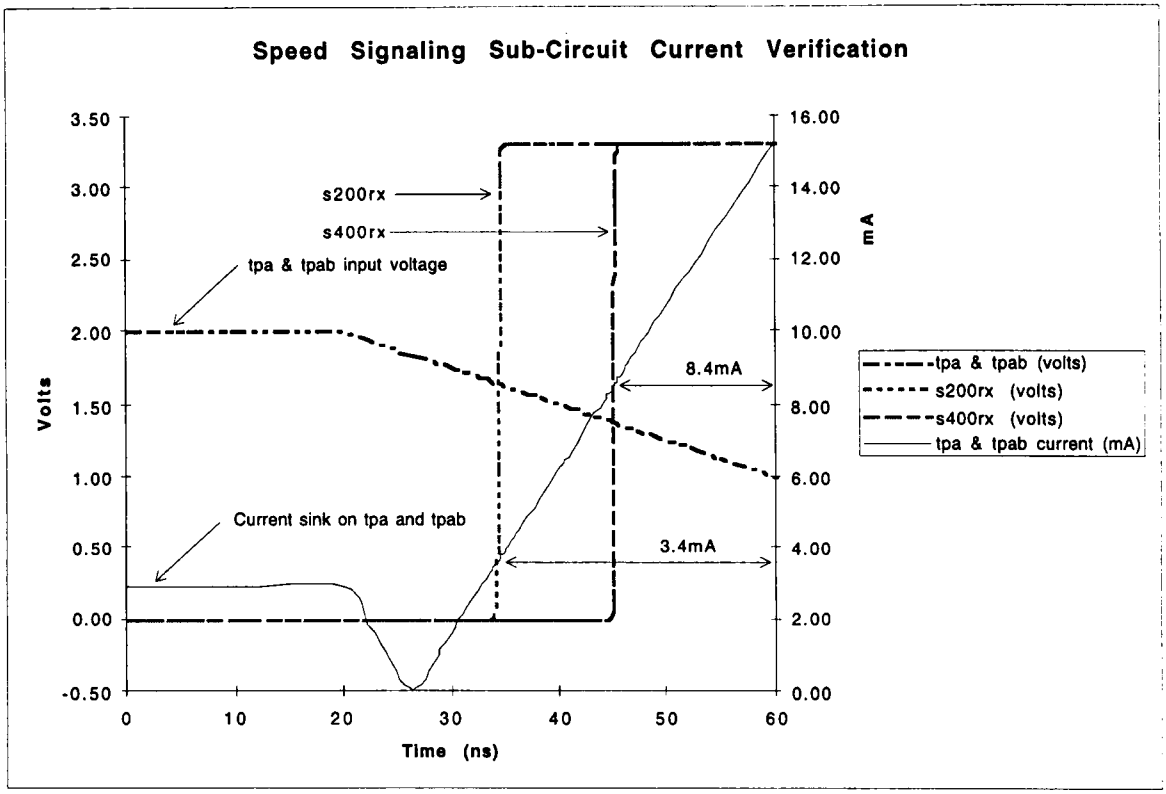


Figure 5.3.4 - Speed Signaling Sub-Circuit Current Verification

Table 5.3.2 - Port Status Enable and Control Signals

Signal	Bias	Port	Description
data_rcvr_pd1	V_{dd}	1	Data receiver power down
strobe_tx_en1	V_{dd}	1	Strobe data enable
strobe_tx_pd1	V_{dd}	1	Strobe transmit power down
strobe_rcvr_en1	Gnd	1	Strobe receiver enable
data_tx_en1	Gnd	1	Data transmit enable
data_tx_pd1	V_{dd}	1	Data transmit power down
bias_detect_pd1	Gnd	1	Port status detection power down
speed_tx_pd1	V_{dd}	1	Speed transmit power down
bg_on1	V_{dd}	1	Band gap starter
ref_on1	V_{dd}	1	Reference voltage
data_rcvr_pd2	V_{dd}	2	Data receiver power down
strobe_tx_en2	V_{dd}	2	Strobe data enable
strobe_tx_pd2	V_{dd}	2	Strobe transmit power down
strobe_rcvr_en2	Gnd	2	Strobe receiver enable
data_tx_en2	V_{dd}	2	Data transmit enable
data_tx_pd2	V_{dd}	2	Data transmit power down
bias_detect_pd2	V_{dd}	2	Port status detection power down
speed_tx_pd2	V_{dd}	2	Speed transmit power down
bg_on2	V_{dd}	2	Band gap starter
ref_on1	V_{dd}	2	Reference voltage

If a port is not connected, the voltage on the *bias_detect* signal will fall to ground. The voltage on the *tpb* and *tpbb* may also fall to ground but will fall at least below the 0.6V specified in Table 2.8.

5.4.1 Port Status Test Case Description

For Port Status simulation, the first step is to enable or disable the necessary subcircuits in the port. This is accomplished by biasing or grounding certain control lines as denoted in Table 5.3.1. This configuration disables all subcircuits in Port#1 and Port#2 except the bias detect subcircuit in Port#1. Both ports have the voltage and current generators enabled by applying V_{dd} to the *bg_on* and *ref_on* control lines. The enable and power down control signals are controlled by the Spice control file given in the Appendix as the Port-to-Port Port Status Spice Control File. No inputs are toggled during the simulation. The only output observed is the *bias_detect1* signal. The simulation to check for a connected port was executed without the cable model due to the DC performance of the cable model. The common mode voltage of the cable model does not allow operation to test the bias detect subcircuit. The problem with the cable is discussed in Section 5.2. The simulation to detect a disconnected port assumes the cable is removed from the port. This allows the common mode voltage to drop to zero volts.

5.4.2 Port Status Process and Operating Parameter Simulation Matrix

The simulation for a connected and disconnected port was executed across the process and operating condition matrix.

The results for detection of a connected port are plotted in Figure 5.4.1. When a port is connected, the common mode voltage is above 1.0V. For this simulation, the common mode voltage settles at *tpbias* (1.84V). The output signal *bias_detect1* remains at ground level. The results for detection of a disconnected port are plotted in Figure 5.4.2. The common mode voltage in Figure 5.4.2 has fallen to zero volts. If this voltage is below 0.6V, the *bias_detect1* signal will have the value of V_{dd} as shown in Figure 5.4.2.

The simulation results are the same across process variance and operating conditions. The Port Sensing capability of the circuit passes IEEE 1394-1995 specifications.

5.5 Power Analysis

Power dissipation is measured as the product of the circuit bias voltage (V_{dd}) and the current sourced through V_{dd}. Spice simulations automatically measure the current sourced by a voltage source, so the data required to analyze power dissipation can be obtained by standard simulations.

For the purpose of power dissipation analysis, port-to-port simulation is not required. Only a single port is necessary to determine power consumption on a port by port basis. No cable or loading other than the external resistor networks will included in the analysis.

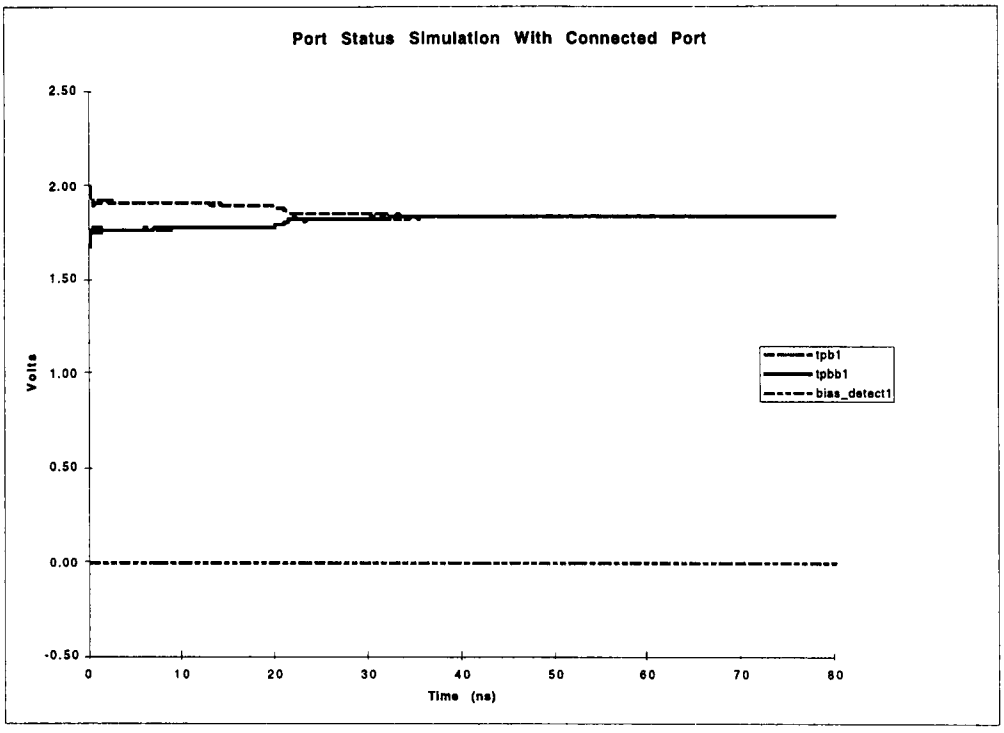


Figure 5.4.1 - Port Status With Connected Port

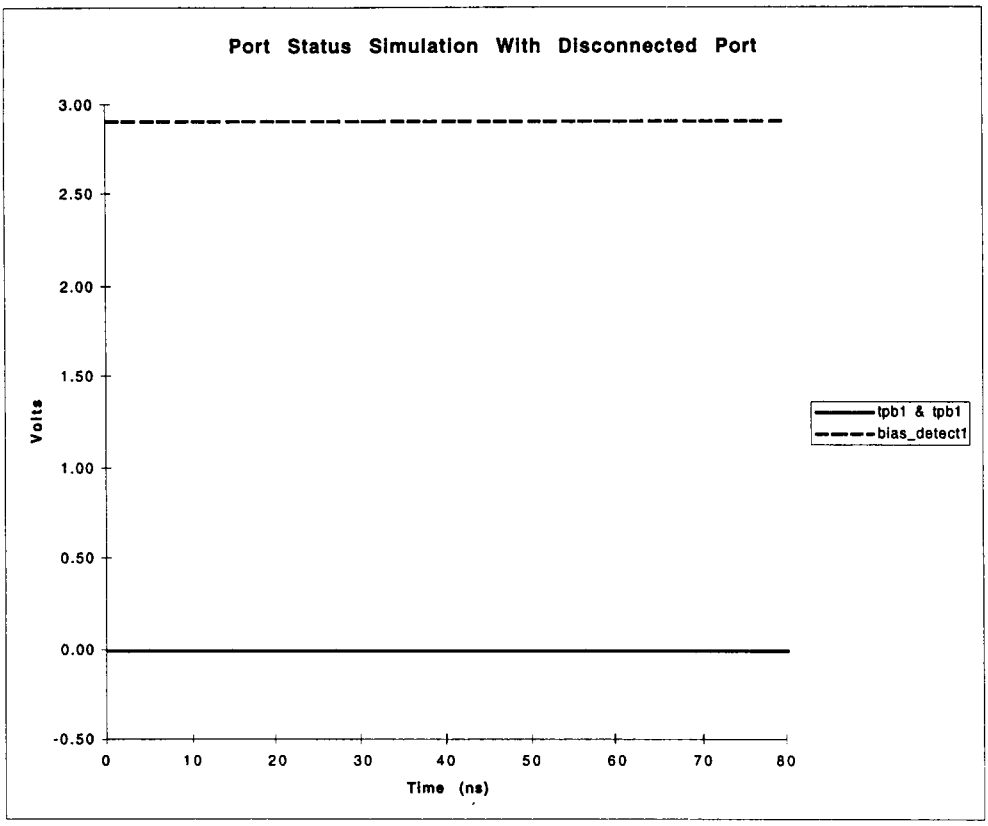


Figure 5.4.2 - Port Status With Disconnected Port

5.5.1 Power Dissipation Test Case Description

There will be two main test cases. The first is to enable the current and voltage reference circuits and disable every other subcircuit in the port. Then each of the six main input output functions of the port will be enabled one at a time. The current will be measured during each mode of operation. The six main functions to be enable are data transmission, data reception, strobe transmission, strobe reception, speed sensing, and port status. These simulation will consist of the six cases as denoted in Table 5.5.1.

Table 5.5.1 - Functional Power Dissipation Matrix

Parasitic Model	Transistor Model	Operating Voltage	Operating Temperature
Typical Case	Typical Case	Nominal	Nominal
Typical Case	Typical Case	Nominal	High
Typical Case	Typical Case	Nominal	Low
Typical Case	Typical Case	High	High
Worst-Case	Worst-Case	Nominal	Nominal
Worst-Case	Worst-Case	High	High

Power dissipation for a typical CMOS microprocessor will increase slightly at high temperatures, but will increase much more at high bias voltages.

The second test case will be to enable and disable every I/O function of the circuit at the same time to determine the maximum and minimum power dissipation expected. The reference sources will also be enabled and disabled to determine the maximum and minimum power dissipation possible. The three test cases are given in Table 5.5.2.

Table 5.5.2 - Minimum and Maximum Power Dissipation Matrix

Reference Sources	**All I/O Functional Circuits**
Enabled	Enabled
Enabled	Disabled
Disabled	Disabled

The cases in Table 5.2.2 will be done only at typical case processing and nominal operating conditions. The Spice control file used for the power dissipation simulation is given in the Appendix as the Power Dissipation Spice Control File. The control signals for the first test case are plotted in Figure 5.5.1. Notice that only one I/O function is enabled at a time. This allows the power dissipation to be determined for the entire circuit during each major function. The control inputs for the second case require only DC voltages without toggling. Table 5.5.3 shows the voltages applied to each signal for each simulation. The name and description for each I/O power down and enable control line is given in Table 5.1.1.

5.5.2 Power Dissipation Process and Operating Parameter Matrix

The six simulations from the first test case were executed. The output from the typical process and nominal operating conditions is plotted in Figure 5.5.2. The current is denoted as $V_{dd}\#i$. Because V_{dd} is held constant through the simulation, $V_{dd}\#i$ is directly proportional to the power dissipation. Notice that the highest values of $V_{dd}\#i$ are during data or strobe transmission. Data transmission was enabled twice during the simulation to check for the repeatability of the simulation measurement.

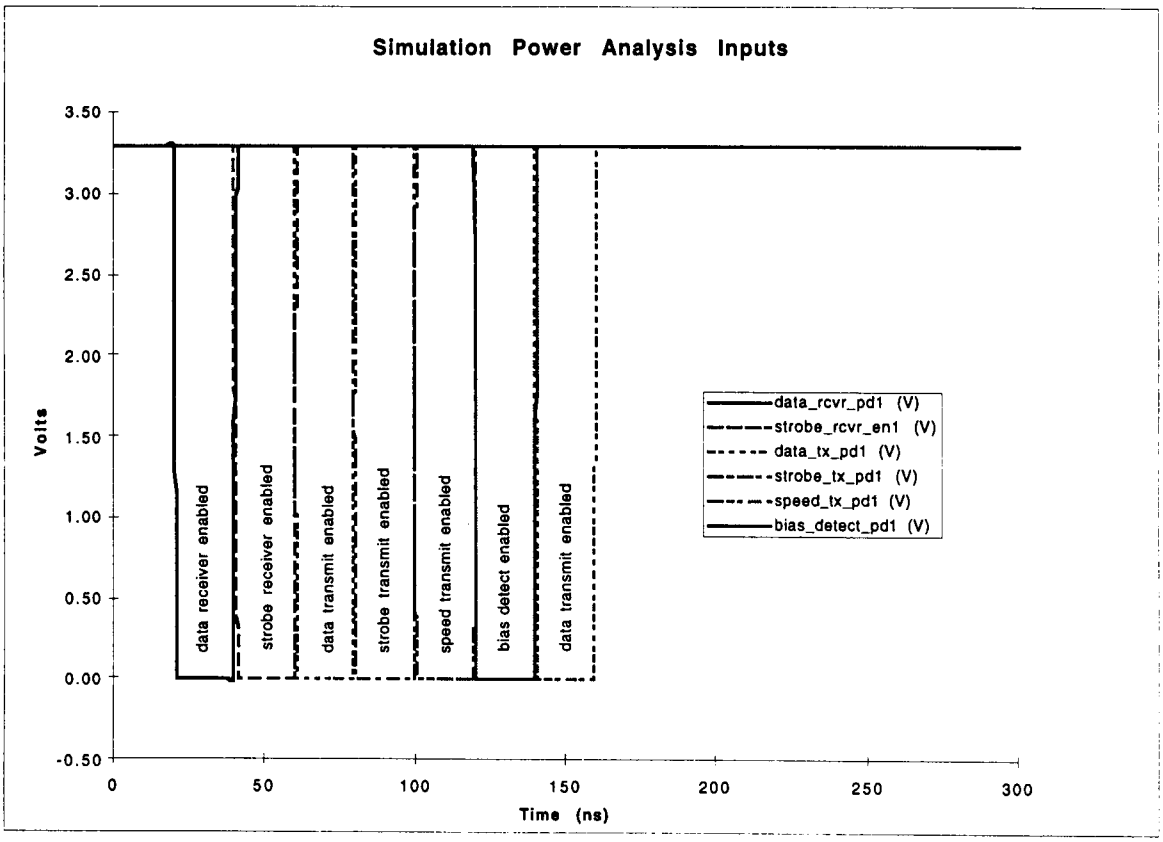


Figure 5.5.1 - Simulation Power Analysis Inputs

Table 5.5.3 - Minimum and Maximum Power Dissipation Inputs

bg_on1	***ref_on1***	**All I/O Power Down and Enable Control Signals**
V_{dd}	V_{dd}	Enabled
V_{dd}	V_{dd}	Disabled
Ground	Ground	Disabled

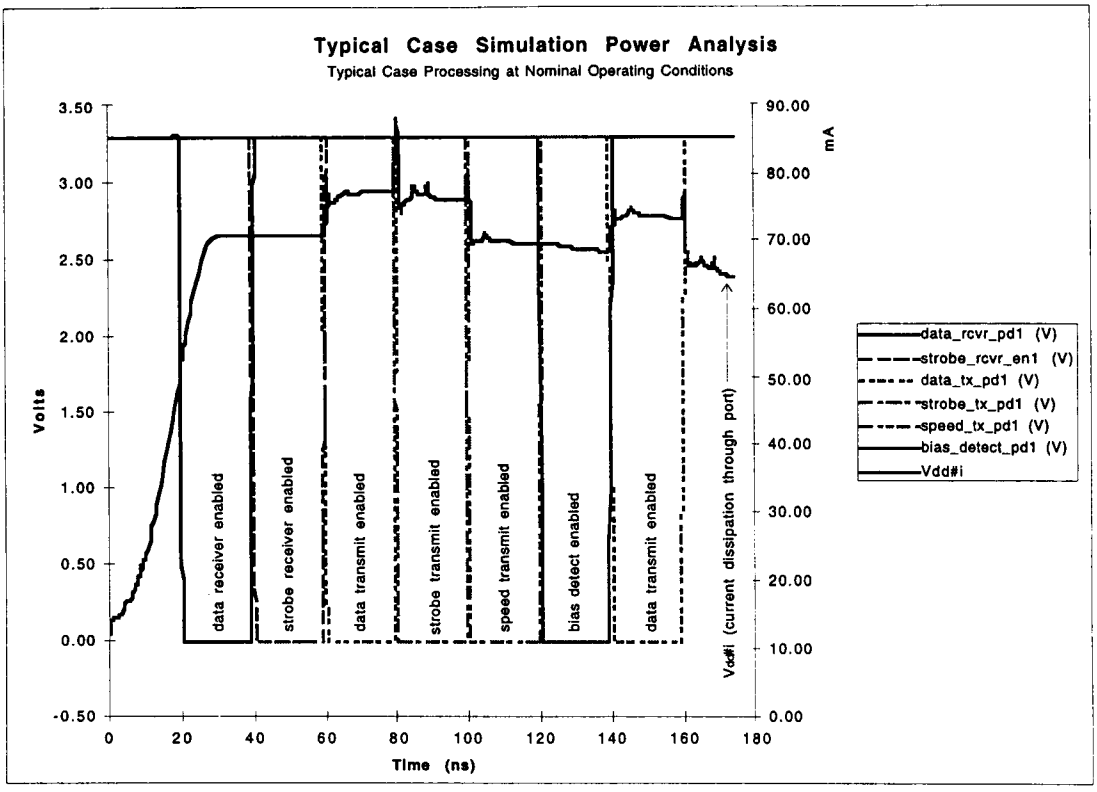


Figure 5.5.2 - Typical Case Simulation Power Analysis

The results of four of the simulations are plotted in Figure 5.5.3. The simulation shows a increase in the current dissipation at high voltage and high temperature. The Spice model increases resistance through all transistors at high temperatures, so the current should decrease as temperature increases. In this case, V_{dd} and temperature were both increased. The plots of current across temperature with V_{dd} held constant are shown in Figure 5.5.4. This data shows the predicted decrease in current. The Spice model does include the leakage current and thermal noise that increase with temperature in a real CMOS circuit as part of the current through V_{dd}. The maximum

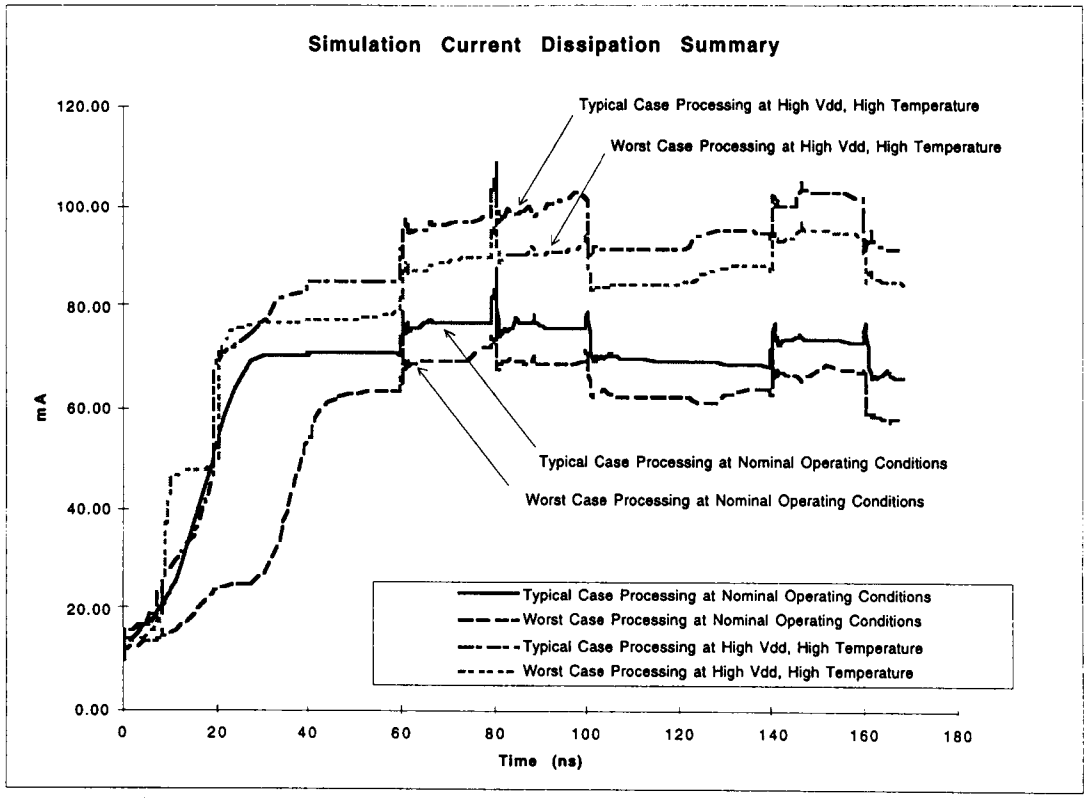


Figure 5.5.3 - Simulation Current Dissipation Summary

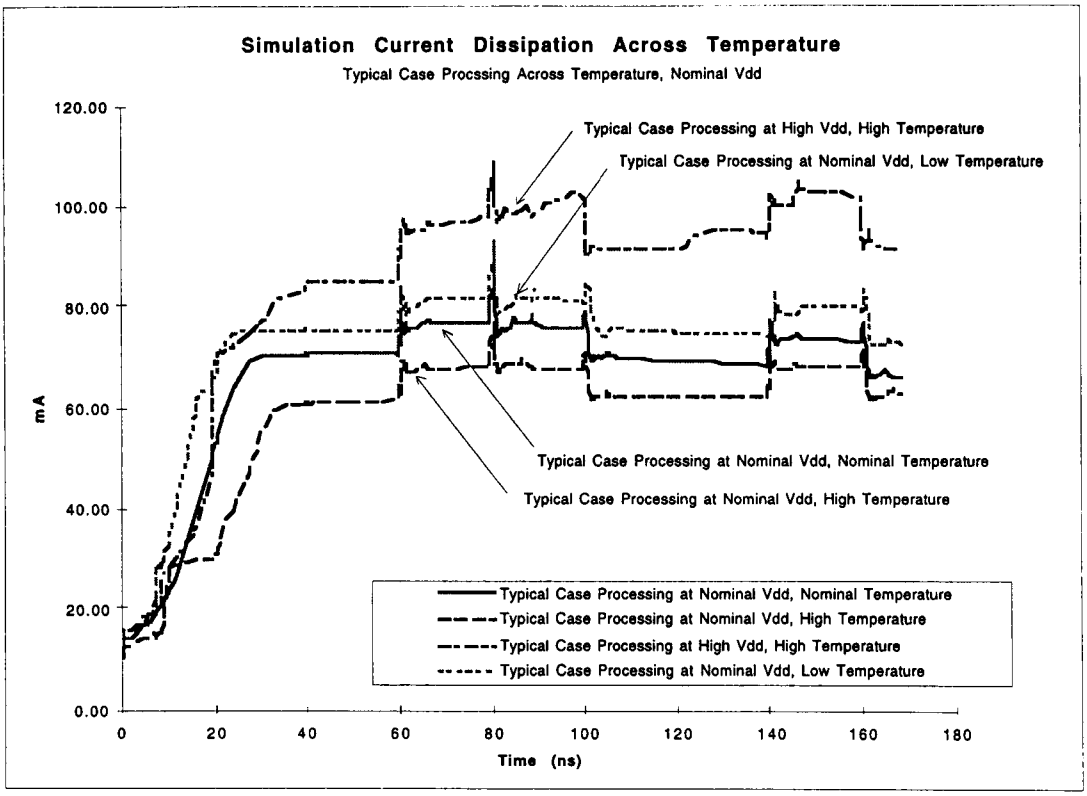


Figure 5.5.4 - Simulation Current Dissipation Across Temperature

current dissipation for each I/O function for the first four simulations is given in Table 5.5.4. The values for data and strobe reception are different because the external resistor networks are different for *tpa1* and *tpab1* than for *tpb1* and *tpbb1*. Data is transmitted through *tpb1* and *tpbb1*, which has a resistive path to ground with the external resistor and capacitor network. This causes the current dissipation to rise slightly higher than the strobe transmission. The strobe transmission has a resistive network to *tpbias*, not ground. Also, as there is no cable connected to another port, no current can flow through the cable media. This would cause a slight increase in current flow during data and strobe transmission. The other I/O functions do not source current into the cable. Speed transmission sinks current from the cable into *tpb1* and *tpbb1*. This current is not included in the simulation because it is not sourced through V_{dd} and is therefore not considered as power dissipation. Note that the data is for a single port. A physical device will have at least three ports plus digital logic to control the ports and interface with the Link device. The power dissipation for selected cases is given in Table 5.5.5. This data shows that power dissipation increases with voltage and temperature. This is typical of real CMOS devices. The simulation output for the second case is given in Table 5.5.6.

Table 5.5.4 - Current Dissipation Summary

Process	**Typical**	**Worst**	**Typical**	**Worst**
Temperature and V_{dd}	**Nominal**	**Nominal**	**High**	**High**
Data Reception	71mA	27mA	77mA	77mA
Strobe Reception	71mA	63mA	85mA	78mA
Data Transmission	77mA	69mA	97mA	89mA
Strobe Transmission	81mA	75mA	102mA	94mA
Speed Transmission	78mA	71mA	102mA	94mA
Port Sensing	70mA	63mA	92mA	85mA

Table 5.5.5 - Power Dissipation Summary

Process Temperature and V_{dd}	Typical Nominal	Worst Nominal	Typical High	Worst High
Data Reception	233mW	89mW	287mW	284mW
Strobe Reception	234mW	209mW	314mW	288mW
Data Transmission	255mW	229mW	359mW	331mW
Strobe Transmission	268mW	246mW	376mW	347mW
Speed Transmission	256mW	234mW	377mW	349mW
Port Sensing	231mW	207mW	340mW	313mW

Table 5.5.6 - Minimum and Maximum Current Dissipation Simulation

Reference Circuits	All I/O Circuits	Current Dissipation	Power Dissipation
Enabled	Enabled	83mA	272mW
Enabled	Disabled	71mA	234mW
Disabled	Disabled	0.88mA	3mW

The results show that most of the power dissipation is within the reference voltage and current generator circuits. When all subcircuits and reference circuits are disabled, the power dissipation reduces from 272mW to 3mW. The current reference generators provide many current sources and sinks for other subcircuits. The voltage reference generators also requires large amount of current through ten bipolar junction transistors.

The IBM21S851 specifies a typical power dissipation of 1.0W. It is a three port device. Typical power consumption for three ports as simulated in this thesis would be 816mW. The addition of power dissipation for the digital logic and an allowance for specified data margin would put the number for the simulated design close to 1.0W. However, the process for the IBM physical device may be very different than the simulation process parameters used in this thesis. Most of the power dissipation will be in the I/O circuit, so the simulation results for a three port Physical device are reasonable for power dissipation.

CHAPTER 6. CONCLUSIONS

6.1 Discussion

Research into the IEEE 1394-1995 high-speed serial bus showed that there are few sources of information in this new area. Most information presented in the thesis comes from the IEEE 1394-1995 specification. Only one book is currently in publication for this topic, and it is used as a source in the thesis. Other information comes from data books from various manufacturers. Determining the specifications was a straightforward yet time consuming process.

The simulation of the cable model yielded promising results. The input signals in the Spice control file were ramped voltage sources with values within the IEEE 1394-1995 specification. Only later was it determined that the cable restricts DC performance. Some common mode simulations were unable to use the cable model. The data transmission and reception simulations were not affected because the problem with the model did not move any signals outside of specified ranges.

The simulation of individual subcircuits did not pose any problems and required much less computer time to complete than Port-to-Port simulations. Port-to-port simulation was more complicated. The addition of the current and voltage reference generators required many nodes to be initialized to allow DC transient convergence before the simulation could begin. The DC cable also induced DC convergence problems that required that some nodes be initialized. Once these problems were solved the simulations proceeded without any more problems.

6.2 Results

The simulation of the data and strobe receiver circuit showed that the circuit does not properly receive data when the input common mode voltage is less than 0.8V. Although the IEEE 1394-1995 specification requires that the common mode input voltage be as low as 0.523V, 400MHz data and strobe reception can still be accomplished. The specification allows any common mode input voltage below 0.935V and above 0.523V to serve as the low-end of the range for 400MHz operation. The critical operation that requires 0.523V is speed signaling, not data and strobe reception. Speed signaling was verified in the port-to-port simulations.

Port-to-port simulation of the data and strobe transmission and reception circuits showed very good performance within the IEEE 1394-1995 specifications. The rise-times, fall-times, operations at various process corners and operating conditions, and operation over various input voltage ranges showed this circuit to be very robust. In no simulation was data ever received incorrectly.

The simulation of the arbitration circuit was complicated by the problem with the cable media interface model. The modified simulation without the cable showed that port-to-port arbitration is robust across process and operating conditions. The arbitration decoding subcircuit also works very well across the entire range of common mode voltage as required by specifications.

Port-to-port simulation of Speed Sensing was successful using the cable model. Simulation of the speed receiver subcircuit was used to verify correct operation within specifications. The simulations showed

Speed Sensing operations are robust across process and operating conditions.

The simulation of Port Status detection was a success during port-to-port simulation. The cable model was not used in the simulation because of the DC performance problems. One test case required that the cable be removed to simulate a disconnected port. The simulations show port status detection operations are robust across process and operating conditions.

The simulation to predict power dissipation showed that most of the power is consumed in the current and voltage reference generators. The values suggest that they are comparable with physical devices manufactured by other companies. The results also follow expected shifts over process and operating condition variance.

6.3 Opportunities for Further Research

Improvements in the cable media model would allow better power dissipation and functional simulations, especially in the Arbitration and Port Status mode of operation. Another opportunity would be to inject noise into the differential voltage signals at various stages during the simulations. Other considerations not addressed by the thesis are when one port is operating under one set of process or operating conditions while a connected port is operating under different conditions. For example, Port#1 could be simulating a typical process, low voltage, and high temperature device while Port#2 could be simulating a worst case process, nominal voltage, and nominal temperature device. In fact, a simulation could be designed to cover

all of these conditions in a matrix format. This would be more complicated and is outside the scope of the thesis.

The lack of a test vehicle provides the greatest opportunity for further research as a mean of verifying the simulation results in the thesis. Unfortunately, the test vehicle was not available for this purpose.

6.4 Conclusion

Simulation of the analog I/O circuit is required before implementing the circuit on silicon. The results of the simulations show good performance of the analog I/O circuit across process variability and operating conditions. Although more simulations under the conditions discussed in Section 6.2 would improve the level of confidence in the circuit design, the IEEE 1394-1995 specifications can be met with the current design.

LIST OF REFERENCES

LIST OF REFERENCES

[1] IEEE Computer Society, IEEE Standard for a High Performance Serial Bus, IEEE 1394-1995 Specification.

[2] KnowledgeTek 1394 Serial Bus Course Notes, Hugh Curley.

[3] Symbios Logic, SYM13FW403 Physical Cable Interface Specification.

[4] Texas Instruments TSB21LV03A IEEE 1394-1995 Triple-Cable Transceiver/Arbiter Specification.

[5] Lucent Technologies FW810 Phy IEEE 1394 Six-Cable Transceiver / Arbiter Device Specification.

[6] International Business Machines IBMS21S760 & IBMS21S850 1394 S200 & S400 Physical Layer Tansceiver Specification.

[7] Philips Semiconductors PDI1394P11 3-Port Physical Layer Interface Specification.

[8] Don Anderson, FireWire System Architecture, Addison Wesley ,1998.

[9] Eric Hannah, IEEE 1394 std. Electrical Issues (non published), Intel Coporation Media & Interconnect Technology Lab, 1997.

[10] Motorola MCSPICE User's Manual, RF & Analog Simulation, Volume I.

[11] Motorola MCSpice Process Library Documentation, Non Published.

APPENDIX

APPENDIX TABLE OF CONTENTS

Cable Model Net List File.....106
Cable Model Spice Control File.....108
Arbitration Spice Control File.....110
Data and Strobe Transmission Spice Control File.....113
Data and Strobe Reception Spice Control File.....115
Port-To-Port Data and Strobe Transmission and Reception
Spice Control File.....118
Port-To-Port Arbitration Spice Control File.....122
Port-To-Port Speed Sensing Spice Control File.....126
Port-To-Port Port Status Spice Control File.....130
Power Dissipation Spice Control File.....133
Top Sheet Schematic.....136
Port_1 and Port_2 Schematic.....137
TPA Schematic.....138
TPB Schematic.....139
Bias Top Schematic.....140
Band Gap Schematic.....141
Reference Voltage Generator #2 Schematic.....142
Reference Current Generator #2 Schematic.....143
Tpbias Voltage Buffer Schematic.....144
Reference Voltage Generator Schematic.....145
Reference Current Generator Schematic.....146
Speed Reception Schematic.....147
Bias Detection Schematic.....148
Data and Strobe Driver Schematic.....149
Data and Strobe Receiver Schematic.....149
Arbitration Reciever Schematic.....149
Speed Transmission Schematic.....149
Band Gap Starter Circuit.....149

CABLE MODEL NET LIST FILE

```
.SUBCKT cable_model tpa tpab tpb tpbb

* .SUBCKT COPPER_CNT 1 2 3 4
* MODEL OF TWO 1394-1995 CONNECTORS + 4.5 METERS OF CABLE
* PINS 1 AND 2 ARE ONE SIDE, PINS 3 AND 4 ARE THE OTHER SIDE
* PINS 1 AND 3 ARE + DIFF VOLTAGE, PINS 2 AND 4 ARE - DIFF VOLTAGE 20
* DEFINE AMMETERS:
VX1 tpa 5 DC gndlevel
VX2 tpab 6 DC gndlevel
VX3 tpb 7 DC gndlevel
VX4 tpbb 8 DC gndlevel

.PARAM gndlevel = 0V

* DEFINE VOLTAGE GENERATORS FOR PINS:
BVPIN1 5 0 V=(V(11)+V(9))/10
BVPIN2 6 0 V=(V(11)-V(9))/10
BVPIN3 7 0 V=(V(16)+V(10))/10
BVPIN4 8 0 V=(V(16)-V(10))/10

* DEFINE CURRENT SOURCES FOR THE TRANSMISSION LINES:
BIDIFF12 0 9 I=I(VX1)-I(VX2)
BIDIFF34 0 10 I=I(VX3)-I(VX4)
BICOM12 0 11 I=I(VX1)+I(VX2)
BICOM34 0 16 I=I(VX3)+I(VX4)

* DEFINE DIFFERENTIAL MODE CONNECTORS+CABLE
* CONNECTOR: 1 INCH TRANS LINE + CAPACITOR:

* THE NUMBERS IN THE ORIGINAL DECK WERE IN INCHES, WHICH WOULD
* BE OK EXCEPT I HAVE TO COMPENSATE FOR SHRINK & SCALE, MIGHT AS
* WELL CONVERT THE WHOLE THING TO MKS WHILE ALSO COMPENSATING
* FOR SCALE & SHRINK CARDS

* PARAMETER "INVSCALE" INVERTS ALL OF THE SCALE AND SHRINK OPTIONS
.PARAM INVSCALE=1E6

.MODEL LOSSYDIFF LTRA RPERL=38.97 GPERL=0 LPERL=533.9E-9 CPERL=52.09E-12
CPERL=52.09E-12
ODIFF12 9 0 109 0 LOSSYDIFF METHOD=LUMP L=(25.4E-3 * INVSCALE)

CCONN1 109 0 0.65PF

ODIFF34 10 0 110 0 LOSSYDIFF METHOD=LUMP L=(25.4E-3 * INVSCALE)
CCONN2 110 0 0.65PF
```

CABLE MODEL NET LIST FILE (CONTINUED)

```
* DIFF TRANSMISSION LINE
* MODEL FOR 4.5 METER DIFFERENTIAL MODE TRANSMISSION LINE:
* MODEL HAND TUNED AGAINST 5.8 DB LOSS AT 400 MHZ,
* BUT FIGURED FOR 200 MHZ


ODIFF 109 0 110 0 LOSSYDIFF METHOD=LUMP L=(4.5 * INVSCALE)


* DEFINE COMMON MODE CONNECTORS+CABLE
* CONNECTOR: 1 INCH TRANS LINE + INDUCTOR-CAPACITOR-INDUCTOR:

.MODEL LOSSYCOMMON LTRA RPERL=11.8 GPERL=0 LPERL=157E-9 CPERL=177E-12
OCOM12 11 0 111 0 LOSSYCOMMON METHOD=LUMP L=(25.4E-3 * INVSCALE)
LCONN1 111 12 3.2NH
CCONN3 12 0 3PF
LCONN2 12 13 3NH

OCOM34 16 0 116 0 LOSSYCOMMON METHOD=LUMP L=(25.4E-3 * INVSCALE)
LCONN3 15 116 3.2NH
CCONN4 15 0 3PF
LCONN4 14 15 3NH

* COMMON MODE TRANSMISSION LINE
* MODEL FOR 4.5 METER COMMON MODE TRANSMISSION LINE:
* MODEL HAND TUNED AGAINST 5.8 DB LOSS AT 400 MHZ,
* BUT FIGURED FOR 200 MHZ

OCOM 13 0 14 0 LOSSYCOMMON METHOD=LUMP L=(4.5 * INVSCALE)

.ENDS
```

CABLE MODEL SPICE CONTROL FILE

```
*************************************************************************
* Libraries and Include Files
*************************************************************************
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs BASE_HP
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs SSIM
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs RESISTOR
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs DIODE
.lib /usr/local/Mcspice/proc/cdr2mos11.mcfcns ssdt_base_fcns
.lib /usr/local/Mcspice/proc/cdr2mos11.mcfcns ssdt_cap_fcns
.include '/usr/local/Mcspice/macros_mclibs/function.mc'
.include '/usr/local/Mcspice/macros_mclibs/mdevices.mc'
.include '/usr/local/Mcspice/macros_mclibs/mgate.mc'
.include '/usr/local/Mcspice/macros_mclibs/complex.mc'
.include '/usr/local/Mcspice/macros_mclibs/mexor.mc'
.include '/usr/local/Mcspice/macros_mclibs/mload.mc'
.include '/usr/local/Mcspice/macros_mclibs/sizes.mc'

*********************************************************
* set sigma
*********************************************************
.param z.sig=3

*********************************************************
* Process and Parasitic Libraries
*********************************************************
* Change BOTH for desired corner: BCS/TYP/WCS/etc.
*       If you are using this file without runcorners.ctl, you may want
*       to use the following two lines and comment-out the ".include"
*       card for corner.inc.
*.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs WCS_FET
*.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs WCS_PARA
.include corner.inc

.ground 0
.include cable.mcspice
xbill1 tpa tpab tpb tpbb cable_model

*************************************************************************
* Stimulus files and/or Sources
*************************************************************************

vtpa tpa 0 pwl (0ns 2.100
+ 10ns 2.100
+ 12ns 1.300
+ 14ns 1.300
+ 16ns 2.100
+ 18ns 2.100
+ 20ns 1.300
+ 30ns 1.300)
```

CABLE MODEL SPICE CONTROL FILE (CONTINUED)

```
vtpab tpab 0 pwl (0ns 1.300
+ 10ns 1.300
+ 12ns 2.100
+ 14ns 2.100
+ 16ns 1.300
+ 18ns 1.300
+ 20ns 2.100
+ 30ns 2.100)

*r1 tpa 0 100
*r2 tpab 0 100

r3 tpa tpbias 55
r4 tpab tpbias 55
vtpbias tpbias 0 dc 1.86V

r5 tpb tpb_cm 55
r6 tpbb tpb_cm 55
r7 tpb_cm 0 5000
ccm tpb_cm 0 250pf


*********************************************************************
* Analysis statements
*********************************************************************
.tran 0.01ns 30ns
.end
```

ARBITRATION SPICE CONTROL FILE

```
*************************************************************************
* Libraries and Include Files                                           *
*************************************************************************
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs BASE_HP
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs SSIM
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs RESISTOR
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs DIODE
.lib /usr/local/Mcspice/proc/cdr2mos11.mcfcns ssdt_base_fcns
.lib /usr/local/Mcspice/proc/cdr2mos11.mcfcns ssdt_cap_fcns
.include /usr/local/Mcspice/macros_mclibs/mdevices.mc
.include '/usr/local/Mcspice/macros_mclibs/function.mc'
.include '/usr/local/Mcspice/macros_mclibs/mgate.mc'
.include '/usr/local/Mcspice/macros_mclibs/complex.mc'
.include '/usr/local/Mcspice/macros_mclibs/mexor.mc'
.include '/usr/local/Mcspice/macros_mclibs/mload.mc'
.include '/usr/local/Mcspice/macros_mclibs/sizes.mc'

*************************************************************
* set sigma
*************************************************************
.param z.sig=3

*************************************************************
* Process and Parasitic Libraries
*************************************************************
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs WCS_FET
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs WCS_PARA

*************************************************************************
* temperature and power supply setup                                    *
*************************************************************************

.options temp=100

.param nomvoltage=3.3
.param bias = nomvoltage
.param gndlevel=0.0
.ground 0 gnd gndlevel
.param arbcmv89 = 0.0445
.global vdd vss gnd

.param arbcmv89 = 0.0445
.param arbcmvn89 = -0.0445
.param arbcmv168 = 0.084
.param arbcmvn168 = -0.084

.param arbbias_start 0.5
.param arbbias_shift 0.5
```

ARBITRATION SPICE CONTROL FILE (CONTINUED)

```
vdd vdd gnd dc bias
vss vss gnd dc gndlevel



*******************************************************************************
* Circuit Description (Netlist)
*******************************************************************************
.include arb_rx/Arb_Rx.mcspice

*******************************************************************************
* Stimulus files and/or Sources
*******************************************************************************
isrc gnd src25 25u
isink sink25 gnd 25u

r1 plus tpcmv 55
r2 minus tpcmv 55
vtpbias tpbias 0 dc 1.84V

varbbias arbbias gnd pwl(
 +    0ns  arbbias_start
 + 159ns  arbbias_start
 + 160ns (arbbias_start + 1*arbbias_shift)
 + 319ns (arbbias_start + 1*arbbias_shift)
 + 320ns (arbbias_start + 2*arbbias_shift)
 + 479ns (arbbias_start + 2*arbbias_shift)
 + 480ns (arbbias_start + 3*arbbias_shift)
 + 639ns (arbbias_start + 3*arbbias_shift)
 + 640ns (arbbias_start + 4*arbbias_shift)
 + 799ns (arbbias_start + 4*arbbias_shift)
 + 800ns (arbbias_start + 5*arbbias_shift))

vplus plus arbbias pwl (
+    0ns arbcmv89
+   39ns arbcmv89
+   41ns arbcmvn89
+   79ns arbcmvn89
+   81ns arbcmv168
+  119ns arbcmv168
+  121ns arbcmvn168
+  159ns arbcmvn168
+  160ns arbcmv89) R
```

ARBITRATION SPICE CONTROL FILE (CONTINUED)

```
vminus arbbias minus pwl (
+    0ns arbcmv89
+   39ns arbcmv89
+   41ns arbcmvn89
+   79ns arbcmvn89
+   81ns arbcmv168
+   119ns arbcmv168
+   121ns arbcmvn168
+   159ns arbcmvn168
+   160ns arbcmv89) R



*********************************************************************
* Analysis statements
*********************************************************************
.tran 0.01ns 960ns
.end
```

DATA AND STROBE TRASMISSION SPICE CONTROL FILE

```
*************************************************************************
* Libraries and Include Files                                           *
*************************************************************************
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs BASE_HP
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs SSIM
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs RESISTOR
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs DIODE
.lib /usr/local/Mcspice/proc/cdr2mos11.mcfcns ssdt_base_fcns
.lib /usr/local/Mcspice/proc/cdr2mos11.mcfcns ssdt_cap_fcns
.include /usr/local/Mcspice/macros_mclibs/mdevices.mc
.include '/usr/local/Mcspice/macros_mclibs/function.mc'
.include '/usr/local/Mcspice/macros_mclibs/mgate.mc'
.include '/usr/local/Mcspice/macros_mclibs/complex.mc'
.include '/usr/local/Mcspice/macros_mclibs/mexor.mc'
.include '/usr/local/Mcspice/macros_mclibs/mload.mc'
.include '/usr/local/Mcspice/macros_mclibs/sizes.mc'

*********************************************************
* set sigma
*********************************************************
.param z.sig=3

*********************************************************
* Process and Parasitic Libraries
*********************************************************
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs TYP_FET
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs TYP_PARA

*************************************************************************
* temperature and power supply setup                                    *
*************************************************************************
.options temp=100

.param nomvoltage=2.9
.param bias = nomvoltage
.param gndlevel=0.0
.ground 0 gnd gndlevel
.global vdd vss gnd


vdd vdd gnd dc bias
vss vss gnd dc gndlevel

*************************************************************************
* Circuit Description (Netlist)
*************************************************************************
.include ds_driver/ds_driver.mcspice
```

DATA AND STROBE TRASMISSION SPICE CONTROL FILE (CONTINUED)

```
*********************************************************************
* Stimulus files and/or Sources
*********************************************************************
isrc gnd src25 25u
isink sink25 gnd 25u
vpd pd gnd dc gndlevel
ven en gnd dc nomvoltage

r10 tx gnd 100000000

r1 tpa tpbias 55
r2 tpab tpbias 55
vtpbias tpbias gnd dc 1.84V

r5 tpb tpb_cm 55
r6 tpbb tpb_cm 55
r7 tpb_cm 0 5000
ccm tpb_cm 0 250pf

vtx tx gnd pwl (
+   0ns nomvoltage
+   1.5ns nomvoltage
+   2.5ns gndlevel
+   4.0ns gndlevel
+   5.0ns nomvoltage
+   6.5ns nomvoltage
+   7.5ns gndlevel
+   9.0ns gndlevel
+   10.0ns nomvoltage)

*********************************************************************
* Analysis statements
*********************************************************************
.tran 0.01ns 20ns
.end
```

DATA AND STROBE RECEPTION SPICE CONTROL FILE

```
*************************************************************************
* Libraries and Include Files                                           *
*************************************************************************
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs BASE_HP
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs SSIM
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs RESISTOR
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs DIODE
.lib /usr/local/Mcspice/proc/cdr2mos11.mcfcns ssdt_base_fcns
.lib /usr/local/Mcspice/proc/cdr2mos11.mcfcns ssdt_cap_fcns
.include /usr/local/Mcspice/macros_mclibs/mdevices.mc
.include '/usr/local/Mcspice/macros_mclibs/function.mc'
.include '/usr/local/Mcspice/macros_mclibs/mgate.mc'
.include '/usr/local/Mcspice/macros_mclibs/complex.mc'
.include '/usr/local/Mcspice/macros_mclibs/mexor.mc'
.include '/usr/local/Mcspice/macros_mclibs/mload.mc'
.include '/usr/local/Mcspice/macros_mclibs/sizes.mc'

*********************************************************
* set sigma (and pray)
*********************************************************
.param z.sig=3

*********************************************************
* Process and Parasitic Libraries
*********************************************************
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs TYP_FET
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs TYP_PARA
*.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs WCS_FET
*.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs WCS_PARA

*************************************************************************
* temperature and power supply setup                                    *
*************************************************************************

.options temp=25

.param nomvoltage=3.3
.param bias = nomvoltage
.param gndlevel=0.0
.ground 0 gnd gndlevel
.global vdd vss gnd

.param arbcmv89 = 0.0445
.param arbcmvn89 = -0.0445
.param arbcmv168 = 0.084
.param arbcmvn168 = -0.084

.param arbbias_start 0.5
.param arbbias_shift 0.5
```

DATA/STORBE RECEPTION SPICE CONTROL FILE (CONTINUED)

```
vdd vdd gnd dc bias
vss vss gnd dc gndlevel

*********************************************************************
* Circuit Description (Netlist)
*********************************************************************
.include ds_receiver/ds_receiver.mcspice

*********************************************************************
* Stimulus files and/or Sources
*********************************************************************
isrc gnd src25 25u
vpd pd gnd dc gndlevel

varbbias arbbias gnd pwl(
+ 0ns  arbbias_start
+ 14ns  arbbias_start
+ 15ns (arbbias_start + 1*arbbias_shift)
+ 29ns (arbbias_start + 1*arbbias_shift)
+ 30ns (arbbias_start + 2*arbbias_shift)
+ 44ns (arbbias_start + 2*arbbias_shift)
+ 45ns (arbbias_start + 3*arbbias_shift)
+ 59ns (arbbias_start + 3*arbbias_shift)
+ 60ns (arbbias_start + 4*arbbias_shift)
+ 84ns (arbbias_start + 4*arbbias_shift)
+ 85ns (arbbias_start + 5*arbbias_shift))

vplus plus arbbias pwl (
+  0ns arbcmv89
+  1.3ns arbcmv89
+  2.5ns arbcmvn89
+  3.8ns arbcmvn89
+  5.0ns arbcmv168
+  6.3ns arbcmv168
+  7.5ns arbcmvn168
+  8.8ns arbcmvn168
+  10.0ns arbcmv89
+  15.0ns arbcmv89) R
vminus arbbias minus pwl (
+  0ns arbcmv89
+  1.3ns arbcmv89
+  2.5ns arbcmvn89
+  3.8ns arbcmvn89
+  5.0ns arbcmv168
+  6.3ns arbcmv168
+  7.5ns arbcmvn168
+  8.8ns arbcmvn168
+  10.0ns arbcmv89
+  15.0ns arbcmv89) R
```

DATA/STORBE RECEPTION SPICE CONTROL FILE (CONTINUED)

```
*******************************************************************************
* Analysis statements
*******************************************************************************
.tran 0.01ns 100ns
.end
```

PORT-TO-PORT DATA AND STROBE TRASMISSION AND RECEPTION SPICE CONTROL FILE

```
*************************************************************************
* Libraries and Include Files                                           *
*************************************************************************
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs BASE_HP
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs SSIM
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs RESISTOR
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs DIODE
.lib /usr/local/Mcspice/proc/cdr2mos11.mcfcns ssdt_base_fcns
.lib /usr/local/Mcspice/proc/cdr2mos11.mcfcns ssdt_cap_fcns
.include /usr/local/Mcspice/macros_mclibs/mdevices.mc
.include '/usr/local/Mcspice/macros_mclibs/function.mc'
.include '/usr/local/Mcspice/macros_mclibs/mgate.mc'
.include '/usr/local/Mcspice/macros_mclibs/complex.mc'
.include '/usr/local/Mcspice/macros_mclibs/mexor.mc'
.include '/usr/local/Mcspice/macros_mclibs/mload.mc'
.include '/usr/local/Mcspice/macros_mclibs/sizes.mc'

*********************************************************
* set sigma
*********************************************************
.param z.sig=3

*********************************************************
* Process and Parasitic Libraries
*********************************************************
*.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs TYP_FET
*.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs TYP_PARA
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs WCS_FET
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs WCS_PARA

*************************************************************************
* temperature and power supply setup                                    *
*************************************************************************
*.options temp=0
.options temp=100
*.options temp=25

*.param nomvoltage=3.7
*.param nomvoltage=3.3
.param nomvoltage=2.9
.param bias = nomvoltage
.param gndlevel=0.0
.ground 0 gnd gndlevel
.global vdd vss gnd

vdd vdd gnd dc bias
vss vss gnd dc gndlevel
```

PORT-TO-PORT DATA AND STROBE TRASMISSION AND RECEPTION SPICE CONTROL FILE (CONTINUED)

```
*******************************************************************************
* Circuit Description (Netlist)
*******************************************************************************
.include cable.mcspice
.include Top.mcspice

*******************************************************************************
* Stimulus files and/or Sources
*******************************************************************************
.IC v#tpa1 = 2.006
.IC v#tpb1 = 2.006
.IC v#tpab1 = 1.690
.IC v#tpbb1 = 1.690
.IC v#tpb_bias1 = 1.838
.IC v#tpa2 = 2.006
.IC v#tpb2 = 2.006
.IC v#tpab2 = 1.690
.IC v#tpbb2 = 1.690
.IC v#tpb_bias2 = 1.838
.IC v#xi_1519:p_bias = 2.5
.IC v#xi_1519:n_bias = 0.611
.IC v#xi_1519:bgv = 1.2417
.IC v#xi_1519:bgv_2 = 0.6203
.IC v#xi_1519:nbias_ref = 0.6594
.IC v#xi_1519:xc_gen:cr_bias = 1.602
.IC v#tpbias_ref1 = 1.841
.IC v#tpbias1 = 1.844
.IC v#xi_2817:p_bias = 2.5
.IC v#xi_2817:n_bias = 0.611
.IC v#xi_2817:bgv = 1.2417
.IC v#xi_2817:bgv_2 = 0.6203
.IC v#xi_2817:nbias_ref = 0.6594
.IC v#xi_2817:xc_gen:cr_bias = 1.602
.IC v#tpbias_ref2 = 1.841
.IC v#tpbias2 = 1.844
vspeed_tx_s200_1 speed_tx_s200_1 gnd dc gndlevel
vspeed_tx_s200_2 speed_tx_s200_2 gnd dc gndlevel
vspeed_tx_s400_1 speed_tx_s400_1 gnd dc gndlevel
vspeed_tx_s400_2 speed_tx_s400_2 gnd dc gndlevel

rrext1 rext1 gnd 6200
rrext2 rext2 gnd 6200

vbg_on1 bg_on1 gnd dc bias
vbg_on2 bg_on2 gnd dc bias
```

PORT-TO-PORT DATA AND STROBE TRASMISSION AND RECEPTION SPICE CONTROL FILE (CONTINUED)

```
vref_on1 ref_on1 gnd dc bias
vref_on2 ref_on2 gnd dc bias

vrgnd1 rgnd1 gnd dc gndlevel
vrgnd2 rgnd2 gnd dc gndlevel

vdata_rcvr_pd1 data_rcvr_pd1 gnd dc bias
vstrobe_tx_pd1 strobe_tx_pd1 gnd dc gndlevel
vspeed_tx_pd1 speed_tx_pd1 gnd dc bias
vdata_tx_pd1 data_tx_pd1 gnd dc gndlevel
vbias_detect_pd1 bias_detect_pd1 gnd dc bias
vstrobe_rcvr_en1 strobe_rcvr_en1 gnd dc bias
vstrobe_tx_en1 strobe_tx_en1 gnd dc bias
vdata_tx_en1 data_tx_en1 gnd dc bias

vdata_rcvr_pd2 data_rcvr_pd2 gnd dc gndlevel
vstrobe_tx_pd2 strobe_tx_pd2 gnd dc bias
vspeed_tx_pd2 speed_tx_pd2 gnd dc bias
vdata_tx_pd2 data_tx_pd2 gnd dc bias
vbias_detect_pd2 bias_detect_pd2 gnd dc bias
vstrobe_rcvr_en2 strobe_rcvr_en2 gnd dc gndlevel
vstrobe_tx_en2 strobe_tx_en2 gnd dc gndlevel
vdata_tx_en2 data_tx_en2 gnd dc gndlevel

vstrobe_tx_data2 strobe_tx_data2 gnd dc gndlevel
vdata_tx_data2 data_tx_data2 gnd dc gndlevel

vdata_tx_data1 data_tx_data1 gnd pwl (
+   0ns nomvoltage
+   1.5ns nomvoltage
+   2.5ns gndlevel
+   4.0ns gndlevel
+   5.0ns nomvoltage
+   6.5ns nomvoltage
+   7.5ns gndlevel
+   9.0ns gndlevel
+   10.0ns nomvoltage)

vstrobe_tx_data1 strobe_tx_data1 gnd pwl(
+   0ns nomvoltage
+   1.5ns nomvoltage
+   2.5ns gndlevel
+   4.0ns gndlevel
+   5.0ns nomvoltage
+   6.5ns nomvoltage
+   7.5ns gndlevel
+   9.0ns gndlevel
+   10.0ns nomvoltage)
```

PORT-TO-PORT DATA AND STROBE TRASMISSION AND RECEPTION SPICE CONTROL FILE (CONTINUED)

```
*********************************************************************
* Analysis statements
*********************************************************************
.tran 0.01ns 40ns
.end
```

PORT-TO-PORT ARBITRATION SPICE CONTROL FILE

```
*************************************************************************
* Libraries and Include Files                                           *
*************************************************************************
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs BASE_HP
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs SSIM
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs RESISTOR
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs DIODE
.lib /usr/local/Mcspice/proc/cdr2mos11.mcfcns ssdt_base_fcns
.lib /usr/local/Mcspice/proc/cdr2mos11.mcfcns ssdt_cap_fcns
.include /usr/local/Mcspice/macros_mclibs/mdevices.mc
.include '/usr/local/Mcspice/macros_mclibs/function.mc'
.include '/usr/local/Mcspice/macros_mclibs/mgate.mc'
.include '/usr/local/Mcspice/macros_mclibs/complex.mc'
.include '/usr/local/Mcspice/macros_mclibs/mexor.mc'
.include '/usr/local/Mcspice/macros_mclibs/mload.mc'
.include '/usr/local/Mcspice/macros_mclibs/sizes.mc'
*
*********************************************************
* set sigma
*********************************************************
.param z.sig=3

*********************************************************
* Process and Parasitic Libraries
*********************************************************
*.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs TYP_FET
*.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs TYP_PARA
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs WCS_FET
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs WCS_PARA

*************************************************************************
* temperature and power supply setup                                    *
*************************************************************************
*.options temp=0
.options temp=100
*.options temp=25

*.param nomvoltage=3.7
*.param nomvoltage=3.3
.param nomvoltage=2.9
.param bias = nomvoltage
.param gndlevel=0.0
.ground 0 gnd gndlevel
.global vdd vss gnd

vdd vdd gnd dc bias
vss vss gnd dc gndlevel
```

PORT-TO-PORT ARBITRATION SPICE CONTROL FILE (CONTINUED)

```
*********************************************************************
* Circuit Description (Netlist)
*********************************************************************
*.include cable.mcspice
.include Top.mcspice

*********************************************************************
* Stimulus files and/or Sources
*********************************************************************
.IC v#tpa1 = 2.006
.IC v#tpb1 = 2.006
.IC v#tpab1 = 1.690
.IC v#tpbb1 = 1.690
.IC v#tpb_bias1 = 1.838
.IC v#tpa2 = 2.006
.IC v#tpb2 = 2.006
.IC v#tpab2 = 1.690
.IC v#tpbb2 = 1.690
.IC v#tpb_bias2 = 1.838

.IC v#xi_1519:p_bias = 2.5
.IC v#xi_1519:n_bias = 0.611
.IC v#xi_1519:bgv = 1.2417
.IC v#xi_1519:bgv_2 = 0.6203
.IC v#xi_1519:nbias_ref = 0.6594
.IC v#xi_1519:xc_gen:cr_bias = 1.602
.IC v#tpbias_ref1 = 1.841
.IC v#tpbias1 = 1.844
.IC v#xi_2817:p_bias = 2.5
.IC v#xi_2817:n_bias = 0.611
.IC v#xi_2817:bgv = 1.2417
.IC v#xi_2817:bgv_2 = 0.6203
.IC v#xi_2817:nbias_ref = 0.6594
.IC v#xi_2817:xc_gen:cr_bias = 1.602
.IC v#tpbias_ref2 = 1.841
.IC v#tpbias2 = 1.844

vspeed_tx_s200_1 speed_tx_s200_1 gnd dc gndlevel
vspeed_tx_s200_2 speed_tx_s200_2 gnd dc gndlevel
vspeed_tx_s400_1 speed_tx_s400_1 gnd dc gndlevel
vspeed_tx_s400_2 speed_tx_s400_2 gnd dc gndlevel

rrext1 rext1 gnd 6200
rrext2 rext2 gnd 6200

vbg_on1 bg_on1 gnd dc bias
vbg_on2 bg_on2 gnd dc bias
vref_on1 ref_on1 gnd dc bias
vref_on2 ref_on2 gnd dc bias
```

PORT-TO-PORT ARBITRATION SPICE CONTROL FILE (CONTINUED)

```
vrgnd1 rgnd1 gnd dc gndlevel
vrgnd2 rgnd2 gnd dc gndlevel

vdata_rcvr_pd1 data_rcvr_pd1 gnd dc bias
vstrobe_tx_pd1 strobe_tx_pd1 gnd dc gndlevel
vspeed_tx_pd1 speed_tx_pd1 gnd dc bias
vdata_tx_pd1 data_tx_pd1 gnd dc gndlevel
vbias_detect_pd1 bias_detect_pd1 gnd dc bias
vstrobe_rcvr_en1 strobe_rcvr_en1 gnd dc bias
vstrobe_tx_en1 strobe_tx_en1 gnd dc bias
vdata_tx_en1 data_tx_en1 gnd dc bias
vdata_rcvr_pd2 data_rcvr_pd2 gnd dc bias
vstrobe_tx_pd2 strobe_tx_pd2 gnd dc gndlevel
vspeed_tx_pd2 speed_tx_pd2 gnd dc bias
vdata_tx_pd2 data_tx_pd2 gnd dc gndlevel
vbias_detect_pd2 bias_detect_pd2 gnd dc bias
vstrobe_rcvr_en2 strobe_rcvr_en2 gnd dc bias
vstrobe_tx_en2 strobe_tx_en2 gnd dc bias
vdata_tx_en2 data_tx_en2 gnd dc bias

vstrobe_tx_data1 strobe_tx_data1 gnd pwl(
    0ns gndlevel
   79ns gndlevel
   80ns nomvoltage)

vdata_tx_data1 data_tx_data1 gnd pwl (
    0ns gndlevel
   39ns gndlevel
   40ns nomvoltage
   79ns nomvoltage
   80ns gndlevel
   119ns gndlevel
   120ns nomvoltage
   159ns nomvoltage)

vstrobe_tx_data2 strobe_tx_data2 gnd pwl(
    0ns gndlevel
   79ns gndlevel
   80ns nomvoltage)

vdata_tx_data2 data_tx_data2 gnd pwl (
    0ns gndlevel
   39ns gndlevel
   40ns nomvoltage
   79ns nomvoltage
   80ns gndlevel
   119ns gndlevel
   120ns nomvoltage
   159ns nomvoltage)
```

PORT-TO-PORT ARBITRATION TRASMISSION SPICE CONTROL FILE
(CONTINUED)

```
*********************************************************************************
* Analysis statements
*********************************************************************************
.tran 0.01ns 160ns
.end
```

PORT-TO-PORT SPEED SENSING SPICE CONTROL FILE

```
*************************************************************************
* Libraries and Include Files                                           *
*************************************************************************
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs BASE_HP
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs SSIM
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs RESISTOR
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs DIODE
.lib /usr/local/Mcspice/proc/cdr2mos11.mcfcns ssdt_base_fcns
.lib /usr/local/Mcspice/proc/cdr2mos11.mcfcns ssdt_cap_fcns
.include /usr/local/Mcspice/macros_mclibs/mdevices.mc
.include '/usr/local/Mcspice/macros_mclibs/function.mc'
.include '/usr/local/Mcspice/macros_mclibs/mgate.mc'
.include '/usr/local/Mcspice/macros_mclibs/complex.mc'
.include '/usr/local/Mcspice/macros_mclibs/mexor.mc'
.include '/usr/local/Mcspice/macros_mclibs/mload.mc'
.include '/usr/local/Mcspice/macros_mclibs/sizes.mc'

*********************************************************
* set sigma
*********************************************************
.param z.sig=3

*********************************************************
* Process and Parasitic Libraries
*********************************************************
*.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs TYP_FET
*.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs TYP_PARA
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs WCS_FET
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs WCS_PARA

*************************************************************************
* temperature and power supply setup                                    *
*************************************************************************
*.options temp=0
.options temp=100
*.options temp=25

*.param nomvoltage=3.7
*.param nomvoltage=3.3
.param nomvoltage=2.9
.param bias = nomvoltage
.param gndlevel=0.0
.ground 0 gnd gndlevel
.global vdd vss gnd

vdd vdd gnd dc bias
vss vss gnd dc gndlevel
```

PORT-TO-PORT SPEED SENSING SPICE CONTROL FILE (CONTINUED)

```
*********************************************************************
* Circuit Description (Netlist)
*********************************************************************
.include cable.mcspice
.include Top.mcspice

*********************************************************************
* Stimulus files and/or Sources
*********************************************************************
.IC v#tpa1 = 2.006
.IC v#tpb1 = 2.006
.IC v#tpab1 = 1.690
.IC v#tpbb1 = 1.690
.IC v#tpb_bias1 = 1.838
.IC v#tpa2 = 2.006
.IC v#tpb2 = 2.006
.IC v#tpab2 = 1.690
.IC v#tpbb2 = 1.690
.IC v#tpb_bias2 = 1.838
.IC v#xi_1519:p_bias = 2.5
.IC v#xi_1519:n_bias = 0.611
.IC v#xi_1519:bgv = 1.2417
.IC v#xi_1519:bgv_2 = 0.6203
.IC v#xi_1519:nbias_ref = 0.6594
.IC v#xi_1519:xc_gen:cr_bias = 1.602
.IC v#tpbias_ref1 = 1.841
.IC v#tpbias1 = 1.844
.IC v#xi_2817:p_bias = 2.5
.IC v#xi_2817:n_bias = 0.611
.IC v#xi_2817:bgv = 1.2417
.IC v#xi_2817:bgv_2 = 0.6203
.IC v#xi_2817:nbias_ref = 0.6594
.IC v#xi_2817:xc_gen:cr_bias = 1.602
.IC v#tpbias_ref2 = 1.841
.IC v#tpbias2 = 1.844

rrext1 rext1 gnd 6200
rrext2 rext2 gnd 6200

vbg_on1 bg_on1 gnd dc bias
vbg_on2 bg_on2 gnd dc bias
vref_on1 ref_on1 gnd dc bias
vref_on2 ref_on2 gnd dc bias

vrgnd1 rgnd1 gnd dc gndlevel
vrgnd2 rgnd2 gnd dc gndlevel
```

PORT-TO-PORT SPEED SENSING SPICE CONTROL FILE (CONTINUED)

```
vdata_rcvr_pd1 data_rcvr_pd1 gnd dc bias
vstrobe_tx_pd1 strobe_tx_pd1 gnd dc bias
vspeed_tx_pd1 speed_tx_pd1 gnd dc gndlevel
vdata_tx_pd1 data_tx_pd1 gnd dc bias
vbias_detect_pd1 bias_detect_pd1 gnd dc bias
vstrobe_rcvr_en1 strobe_rcvr_en1 gnd dc bias
vstrobe_tx_en1 strobe_tx_en1 gnd dc gndlevel
vdata_tx_en1 data_tx_en1 gnd dc gndlevel

vdata_rcvr_pd2 data_rcvr_pd2 gnd dc bias
vstrobe_tx_pd2 strobe_tx_pd2 gnd dc bias
vspeed_tx_pd2 speed_tx_pd2 gnd dc gndlevel
vdata_tx_pd2 data_tx_pd2 gnd dc bias
vbias_detect_pd2 bias_detect_pd2 gnd dc bias
vstrobe_rcvr_en2 strobe_rcvr_en2 gnd dc bias
vstrobe_tx_en2 strobe_tx_en2 gnd dc gndlevel
vdata_tx_en2 data_tx_en2 gnd dc gndlevel

vstrobe_tx_data1 strobe_tx_data1 gnd dc nomvoltage
vstrobe_tx_data2 strobe_tx_data2 gnd dc nomvoltage
vdata_tx_data1 data_tx_data1 gnd dc nomvoltage
vdata_tx_data2 data_tx_data2 gnd dc nomvoltage

vspeed_tx_s400_1 speed_tx_s400_1 gnd pwl (
+    0ns gndlevel
+   39ns gndlevel
+   40ns gndlevel
+   79ns gndlevel
+   80ns nomvoltage
+   119ns nomvoltage
+   120ns nomvoltage
+   159ns nomvoltage)

vspeed_tx_s200_1 speed_tx_s200_1 gnd pwl (
+    0ns gndlevel
+   39ns gndlevel
+   40ns nomvoltage
+   79ns nomvoltage
+   80ns gndlevel
+   119ns gndlevel
+   120ns nomvoltage
+   159ns nomvoltage)
```

PORT-TO-PORT SPEED SENSING SPICE CONTROL FILE (CONTINUED)

```
vspeed_tx_s400_2 speed_tx_s400_2 gnd pwl (
+     0ns gndlevel
+    39ns gndlevel
+    40ns gndlevel
+    79ns gndlevel
+    80ns nomvoltage
+   119ns nomvoltage
+   120ns nomvoltage
+   159ns nomvoltage)


vspeed_tx_s200_2 speed_tx_s200_2 gnd pwl (
+     0ns gndlevel
+    39ns gndlevel
+    40ns nomvoltage
+    79ns nomvoltage
+    80ns gndlevel
+   119ns gndlevel
+   120ns nomvoltage
+   159ns nomvoltage)


*********************************************************************
* Analysis statements
*********************************************************************
.tran 0.01ns 300ns
.end
```

PORT-TO-PORT PORT STATUS SPICE CONTROL FILE

```
*************************************************************************
* Libraries and Include Files                                           *
*************************************************************************
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs BASE_HP
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs SSIM
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs RESISTOR
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs DIODE
.lib /usr/local/Mcspice/proc/cdr2mos11.mcfcns ssdt_base_fcns
.lib /usr/local/Mcspice/proc/cdr2mos11.mcfcns ssdt_cap_fcns
.include /usr/local/Mcspice/macros_mclibs/mdevices.mc
.include '/usr/local/Mcspice/macros_mclibs/function.mc'
.include '/usr/local/Mcspice/macros_mclibs/mgate.mc'
.include '/usr/local/Mcspice/macros_mclibs/complex.mc'
.include '/usr/local/Mcspice/macros_mclibs/mexor.mc'
.include '/usr/local/Mcspice/macros_mclibs/mload.mc'
.include '/usr/local/Mcspice/macros_mclibs/sizes.mc'

*********************************************************
* set sigma
*********************************************************
.param z.sig=3

*********************************************************
* Process and Parasitic Libraries
*********************************************************
*.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs TYP_FET
*.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs TYP_PARA
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs WCS_FET
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs WCS_PARA

*************************************************************************
* temperature and power supply setup                                    *
*************************************************************************
*.options temp=0
.options temp=100
*.options temp=25

*.param nomvoltage=3.7
*.param nomvoltage=3.3
.param nomvoltage=2.9
.param bias = nomvoltage
.param gndlevel=0.0
.ground 0 gnd gndlevel
.global vdd vss gnd

vdd vdd gnd dc bias
vss vss gnd dc gndlevel
```

PORT-TO-PORT PORT STATUS SPICE CONTROL FILE (CONTINUED)

```
*********************************************************************
* Circuit Description (Netlist)
*********************************************************************
*.include cable.mcspice
.include Top.mcspice

*********************************************************************
* Stimulus files and/or Sources
*********************************************************************
.IC v#tpa1 = 2.006
.IC v#tpb1 = 2.006
.IC v#tpab1 = 1.690
.IC v#tpbb1 = 1.690
.IC v#tpb_bias1 = 1.838
.IC v#tpa2 = 2.006
.IC v#tpb2 = 2.006
.IC v#tpab2 = 1.690
.IC v#tpbb2 = 1.690
.IC v#tpb_bias2 = 1.838

.IC v#xi_1519:p_bias = 2.5
.IC v#xi_1519:n_bias = 0.611
.IC v#xi_1519:bgv = 1.2417
.IC v#xi_1519:bgv_2 = 0.6203
.IC v#xi_1519:nbias_ref = 0.6594
.IC v#xi_1519:xc_gen:cr_bias = 1.602
.IC v#tpbias_ref1 = 1.841
.IC v#tpbias1 = 1.844
.IC v#xi_2817:p_bias = 2.5
.IC v#xi_2817:n_bias = 0.611
.IC v#xi_2817:bgv = 1.2417
.IC v#xi_2817:bgv_2 = 0.6203
.IC v#xi_2817:nbias_ref = 0.6594
.IC v#xi_2817:xc_gen:cr_bias = 1.602
.IC v#tpbias_ref2 = 1.841
.IC v#tpbias2 = 1.844

vspeed_tx_s200_1 speed_tx_s200_1 gnd dc gndlevel
vspeed_tx_s200_2 speed_tx_s200_2 gnd dc gndlevel
vspeed_tx_s400_1 speed_tx_s400_1 gnd dc gndlevel
vspeed_tx_s400_2 speed_tx_s400_2 gnd dc gndlevel

rrext1 rext1 gnd 6200
rrext2 rext2 gnd 6200

vbg_on1 bg_on1 gnd dc bias
vbg_on2 bg_on2 gnd dc bias
vref_on1 ref_on1 gnd dc bias
vref_on2 ref_on2 gnd dc bias
```

PORT-TO-PORT PORT STATUS SPICE CONTROL FILE (CONTINUED)

```
vrgnd1 rgnd1 gnd dc gndlevel
vrgnd2 rgnd2 gnd dc gndlevel

vdata_rcvr_pd1 data_rcvr_pd1 gnd dc bias
vstrobe_tx_pd1 strobe_tx_pd1 gnd dc bias
vspeed_tx_pd1 speed_tx_pd1 gnd dc bias
vdata_tx_pd1 data_tx_pd1 gnd dc bias
vbias_detect_pd1 bias_detect_pd1 gnd dc gndlevel
vstrobe_rcvr_en1 strobe_rcvr_en1 gnd dc bias
vstrobe_tx_en1 strobe_tx_en1 gnd dc gndlevel
vdata_tx_en1 data_tx_en1 gnd dc gndlevel

vdata_rcvr_pd2 data_rcvr_pd2 gnd dc bias
vstrobe_tx_pd2 strobe_tx_pd2 gnd dc bias
vspeed_tx_pd2 speed_tx_pd2 gnd dc bias
vdata_tx_pd2 data_tx_pd2 gnd dc bias
vbias_detect_pd2 bias_detect_pd2 gnd dc gndlevel
vstrobe_rcvr_en2 strobe_rcvr_en2 gnd dc bias
vstrobe_tx_en2 strobe_tx_en2 gnd dc gndlevel
vdata_tx_en2 data_tx_en2 gnd dc gndlevel


*********************************************************************
* Analysis statements
*********************************************************************
.tran 0.01ns 80ns
.end
```

POWER DISSIPATION SPICE CONTROL FILE

```
*************************************************************************
* Libraries and Include Files                                           *
*************************************************************************
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs BASE_HP
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs SSIM
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs RESISTOR
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs DIODE
.lib /usr/local/Mcspice/proc/cdr2mos11.mcfcns ssdt_base_fcns
.lib /usr/local/Mcspice/proc/cdr2mos11.mcfcns ssdt_cap_fcns
.include /usr/local/Mcspice/macros_mclibs/mdevices.mc
.include '/usr/local/Mcspice/macros_mclibs/function.mc'
.include '/usr/local/Mcspice/macros_mclibs/mgate.mc'
.include '/usr/local/Mcspice/macros_mclibs/complex.mc'
.include '/usr/local/Mcspice/macros_mclibs/mexor.mc'
.include '/usr/local/Mcspice/macros_mclibs/mload.mc'
.include '/usr/local/Mcspice/macros_mclibs/sizes.mc'

*********************************************************
* set sigma
*********************************************************
.param z.sig=3

*********************************************************
* Process and Parasitic Libraries
*********************************************************
*.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs TYP_FET
*.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs TYP_PARA
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs WCS_FET
.lib /usr/local/Mcspice/proc/cdr2mos11.mclibs WCS_PARA

*************************************************************************
* temperature and power supply setup                                    *
*************************************************************************
*.options temp=0
.options temp=100
*.options temp=25

*.param nomvoltage=3.7
*.param nomvoltage=3.3
.param nomvoltage=2.9
.param bias = nomvoltage
.param gndlevel=0.0
.ground 0 gnd gndlevel
.global vdd vss gnd

vdd vdd gnd dc bias
vss vss gnd dc gndlevel
```

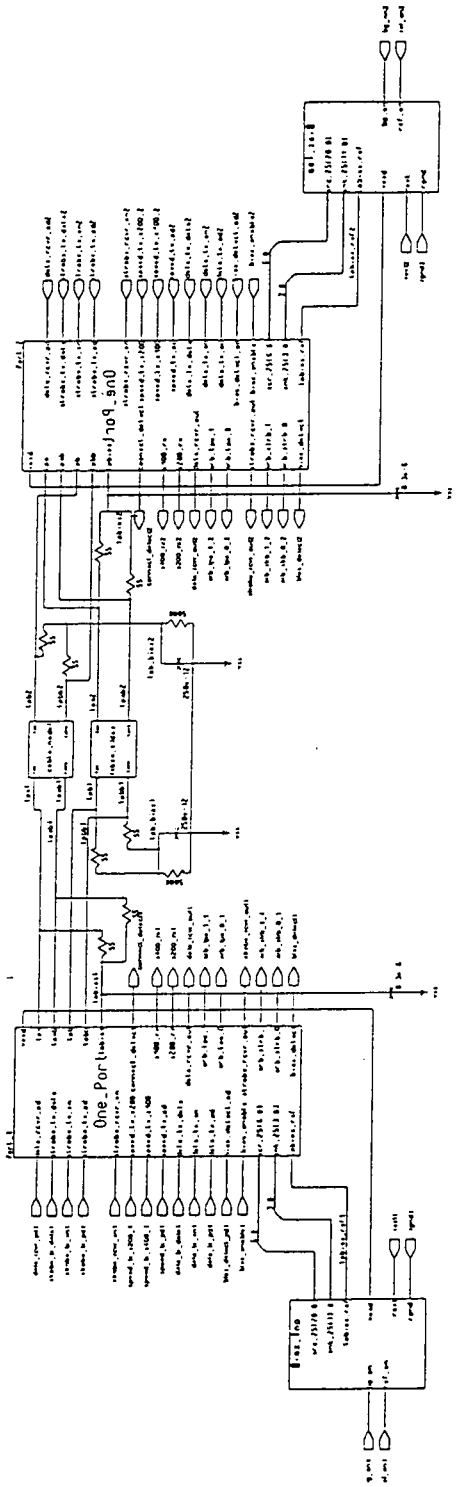
POWER DISSIPATION SPICE CONTROL FILE (CONTINUED)

```
*******************************************************************************
* Circuit Description (Netlist)
*******************************************************************************
.include topnocable.mcspice

*******************************************************************************
* Stimulus files and/or Sources
*******************************************************************************

.IC v#tpa1 = 2.006
.IC v#tpb1 = 1.690
.IC v#tpab1 = 1.690
.IC v#tpbb1 = 1.690
.IC v#tpb_bias1 = 1.838
.IC v#xi_56:p_bias = 2.5
.IC v#xi_56:n_bias = 0.611
.IC v#xi_56:bgv = 1.2417
.IC v#xi_56:bgv_2 = 0.6203
.IC v#xi_56:nbias_ref = 0.6594
.IC v#xi_56:xc_gen:cr_bias = 1.602
.IC v#tpbias_ref1 = 1.841
.IC v#tpbias1 = 1.844

vspeed_tx_s200_1 speed_tx_s200_1 gnd dc gndlevel
vspeed_tx_s400_1 speed_tx_s400_1 gnd dc gndlevel

rrext1 rext1 gnd 6200

vbg_on1 bg_on1 gnd dc bias
vref_on1 ref_on1 gnd dc bias

vdata_rcvr_pd1 data_rcvr_pd1 gnd pwl (0ns bias 19ns bias 20ns gndlevel
39ns gndlevel 40ns bias)

vstrobe_rcvr_en1 strobe_rcvr_en1 gnd pwl (0ns bias 39ns bias 40ns
gndlevel 59ns gndlevel 60ns bias)

vdata_tx_pd1 data_tx_pd1 gnd pwl (0ns bias 59ns bias 60ns gndlevel 79ns
gndlevel 80ns bias 139ns bias 140ns gndlevel 159ns gndlevel 160ns bias)

vdata_tx_en1 data_tx_en1 gnd pwl (0ns gndlevel 59ns gndlevel 60ns bias
79ns bias 80ns gndlevel 139ns gndlevel 140ns bias 159ns bias 160ns
gndlevel)

vstrobe_tx_pd1 strobe_tx_pd1 gnd pwl (0ns bias 79ns bias 80ns gndlevel
99ns gndlevel 100ns bias)
```

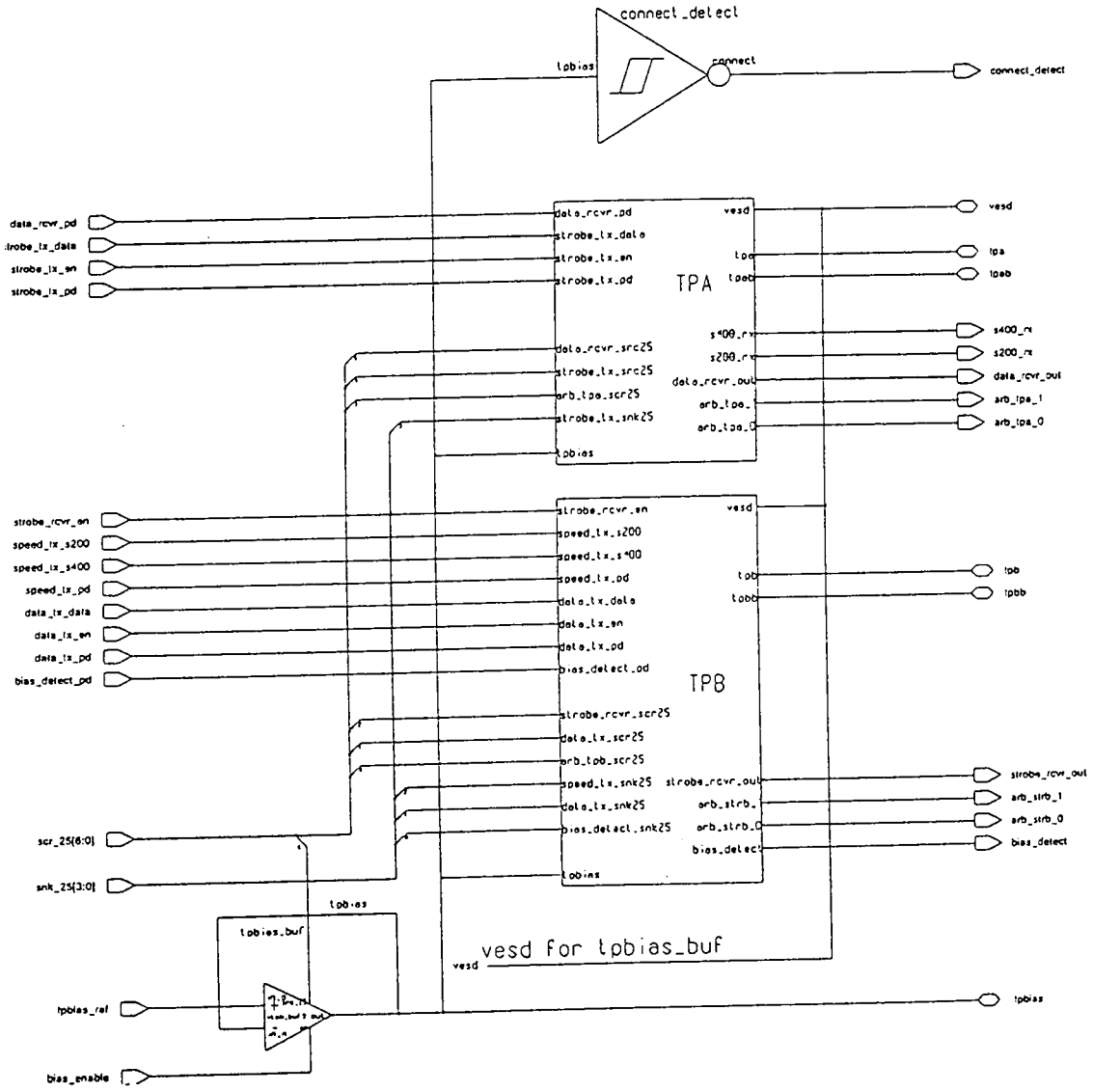
POWER DISSIPATION SPICE CONTROL FILE (CONTINUED)

```
vstrobe_tx_en1 strobe_tx_en1 gnd pwl (0ns gndlevel 79ns gndlevel 80ns
bias 99ns bias 100ns gndlevel)

vspeed_tx_pd1 speed_tx_pd1 gnd pwl (0ns bias 99ns bias 100ns gndlevel
119ns gndlevel 120ns bias)

vbias_detect_pd1 bias_detect_pd1 gnd pwl (0ns bias 119ns bias 120ns
gndlevel 139ns gndlevel 140ns bias)

vstrobe_tx_data1 strobe_tx_data1 gnd dc gndlevel
vdata_tx_data1 data_tx_data1 gnd dc nomvoltage


**********************************************************************
* Analysis statements
**********************************************************************
.tran 0.01ns 300ns
                                        .end
```

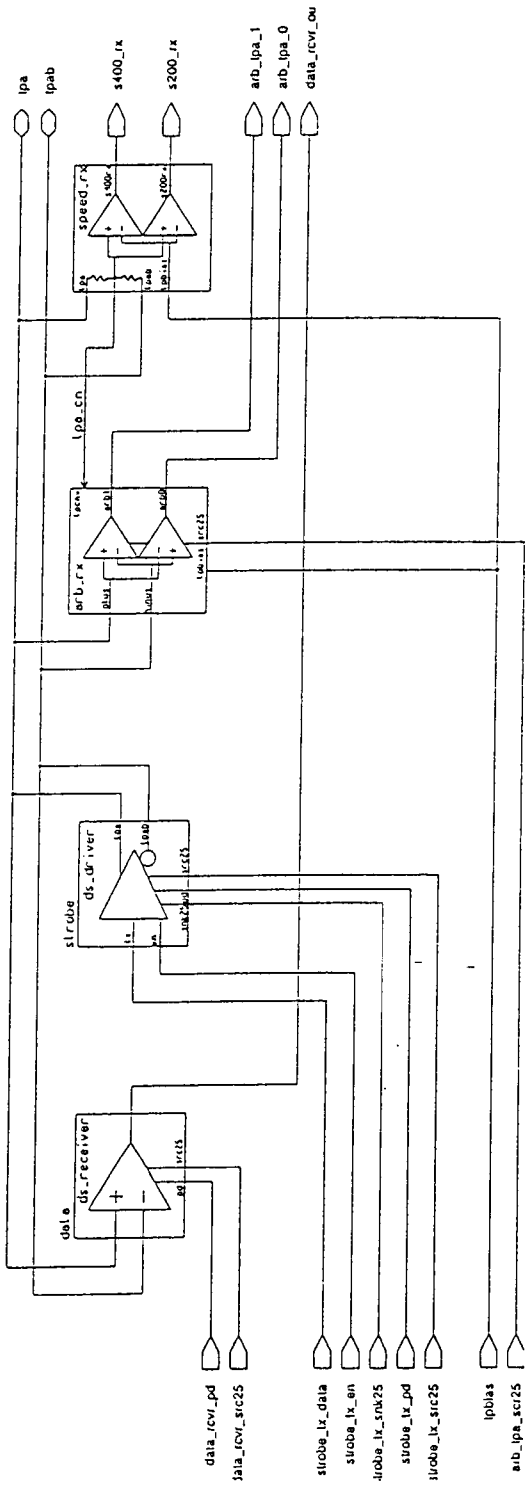
Top Sheet Schematic



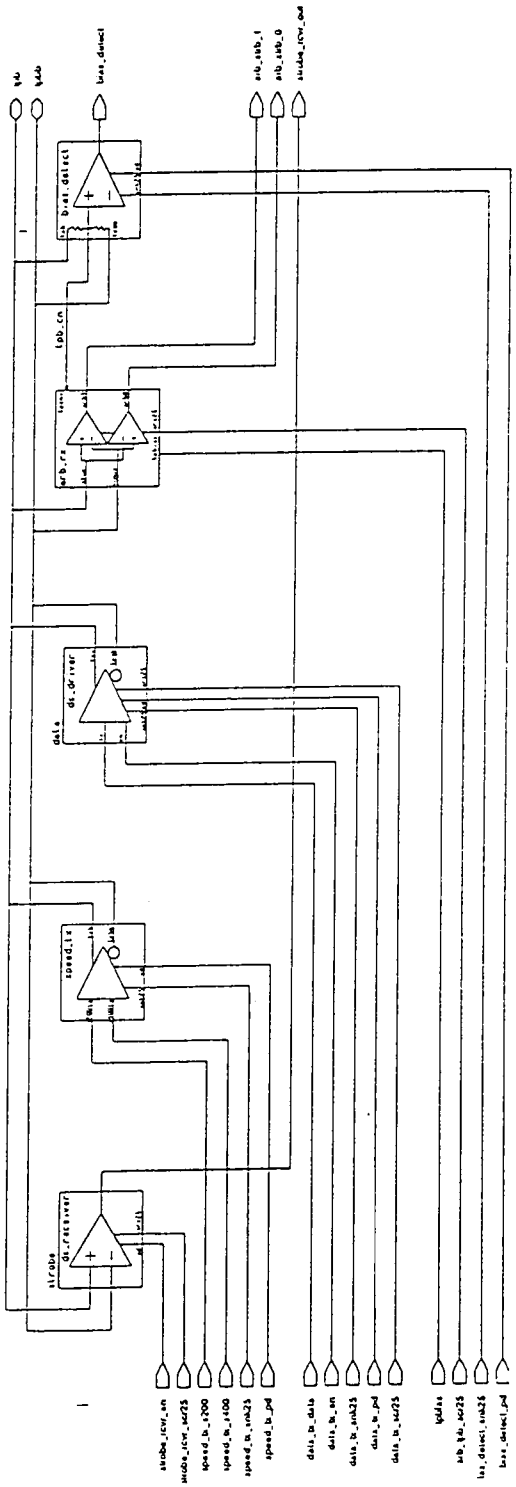
Port_1 and Port_2
(One_Port)



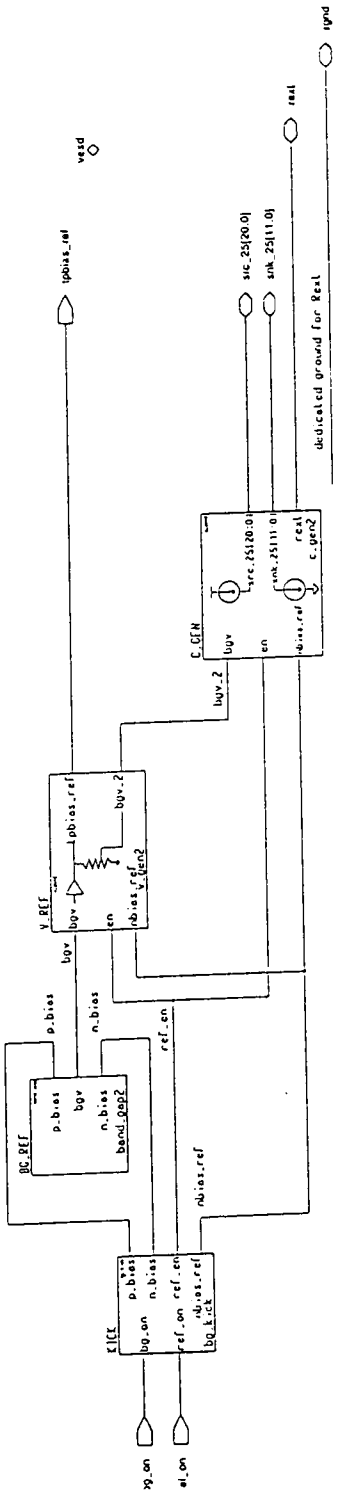
TPA Schematic



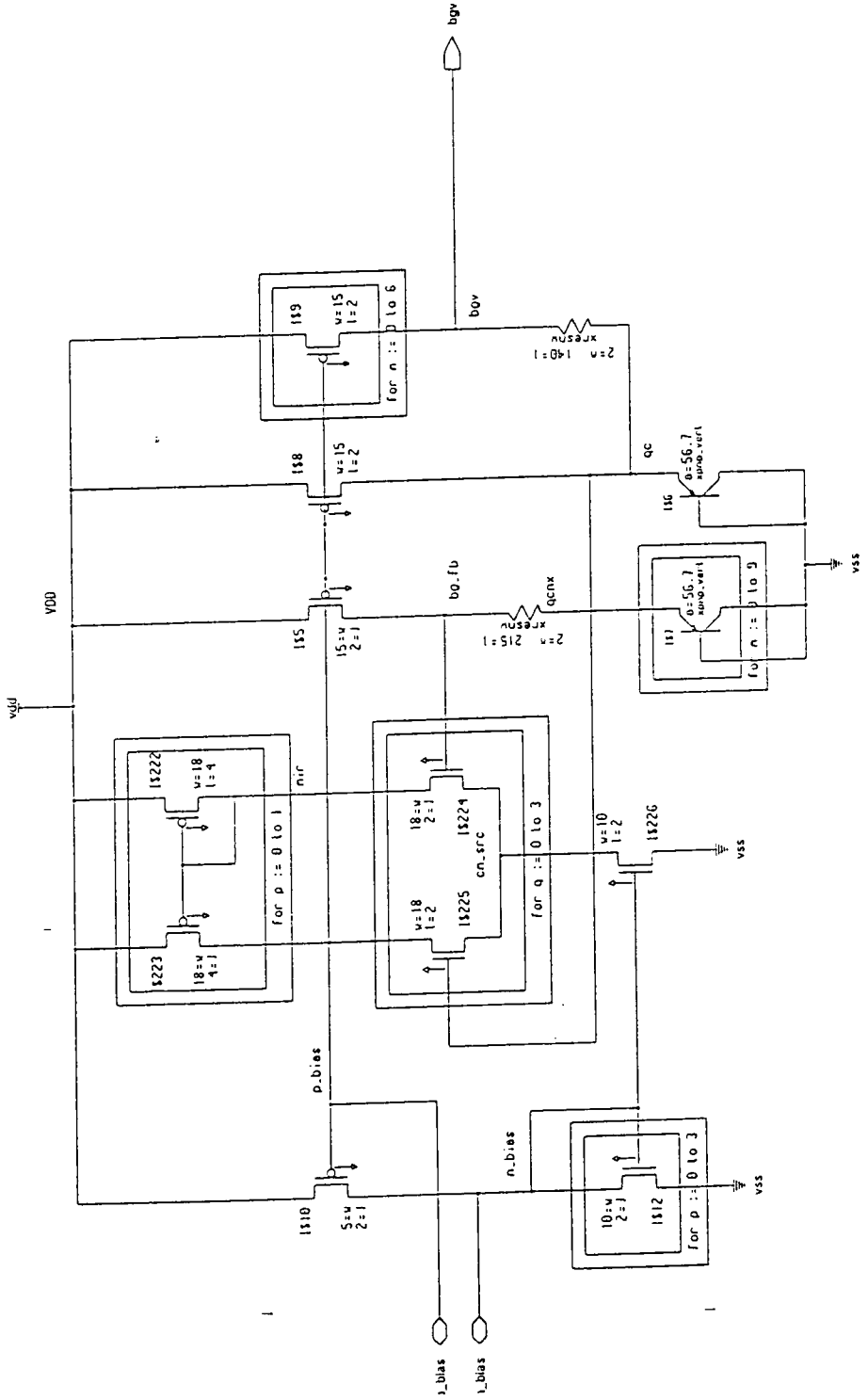
TPB Schematic



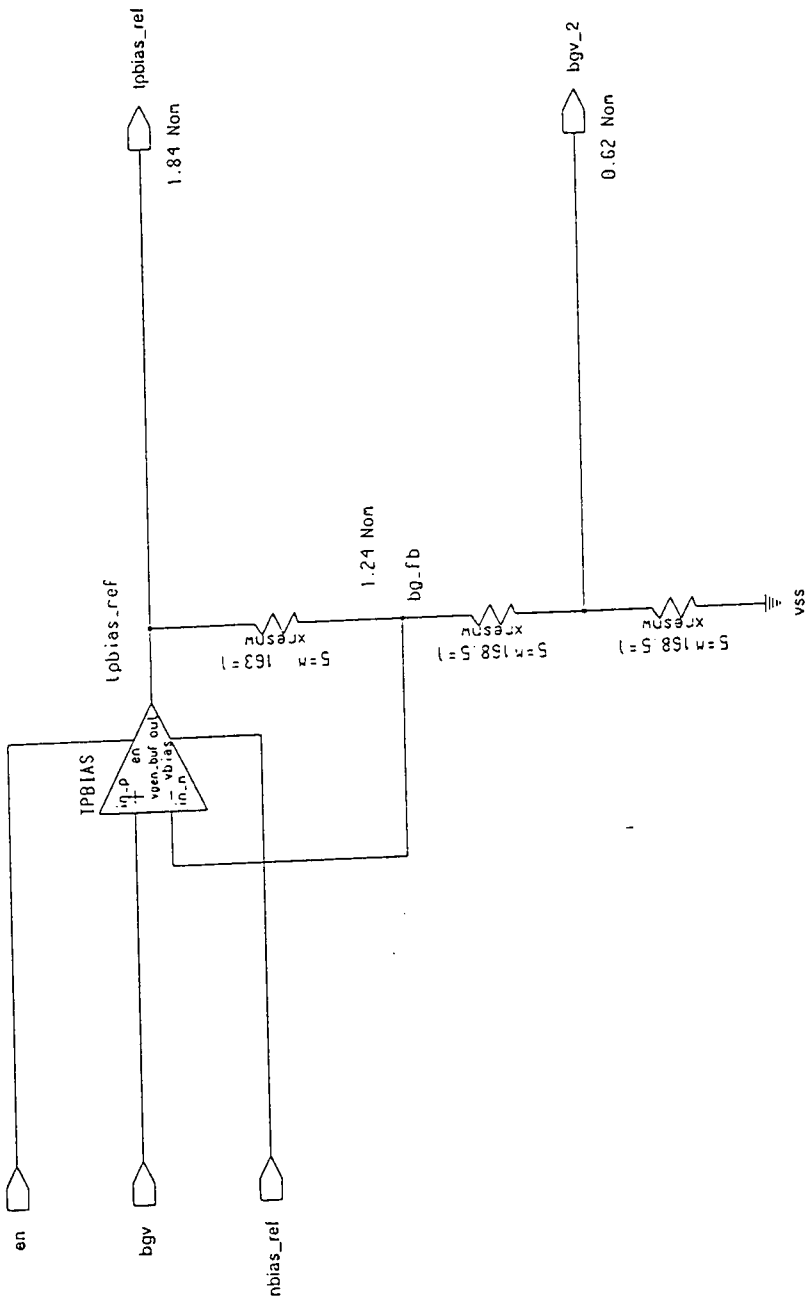
Bias Top Schematic
(bias_top)



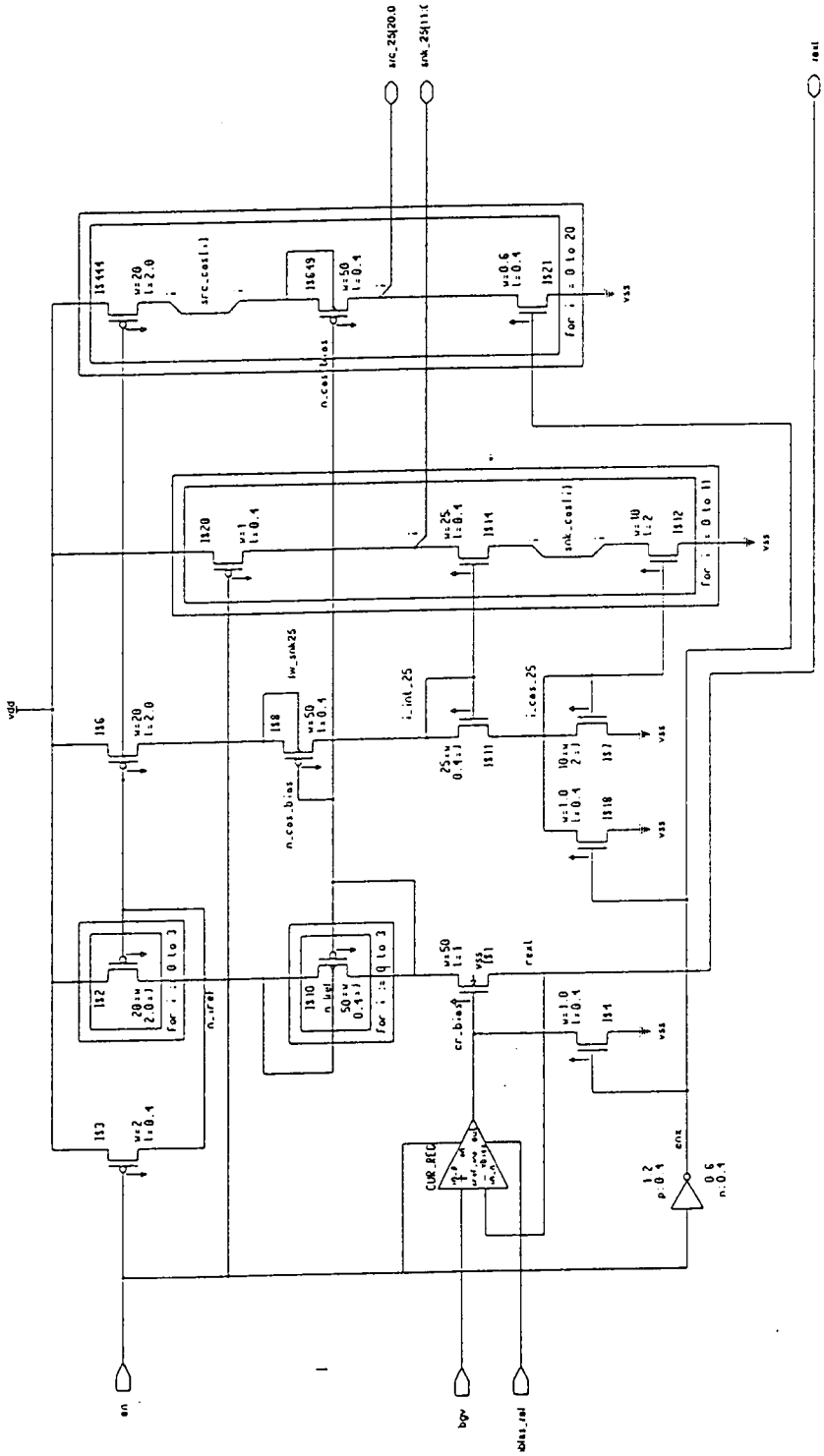
Band Gap Schematic #2
(band_gap2)



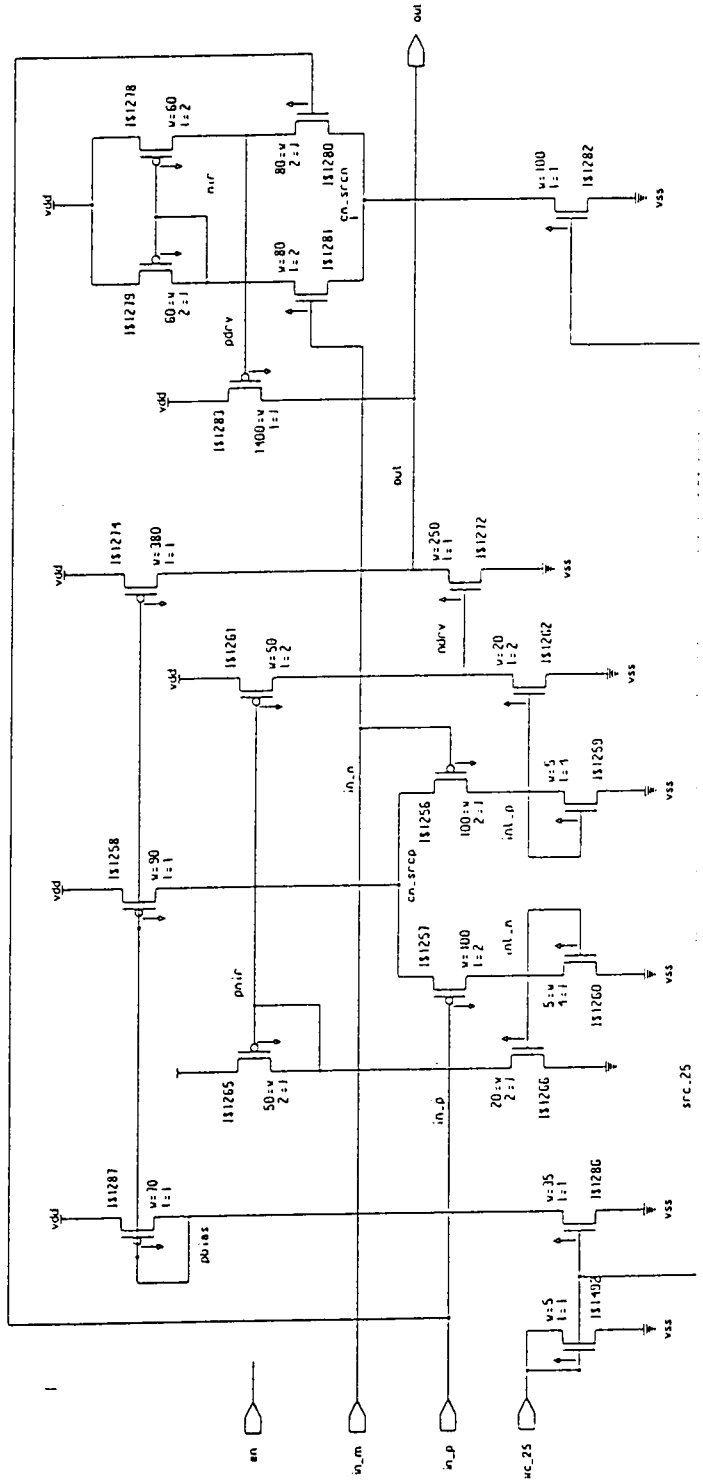
Reference Voltage Generator #2 Schematic
(v_gen2)



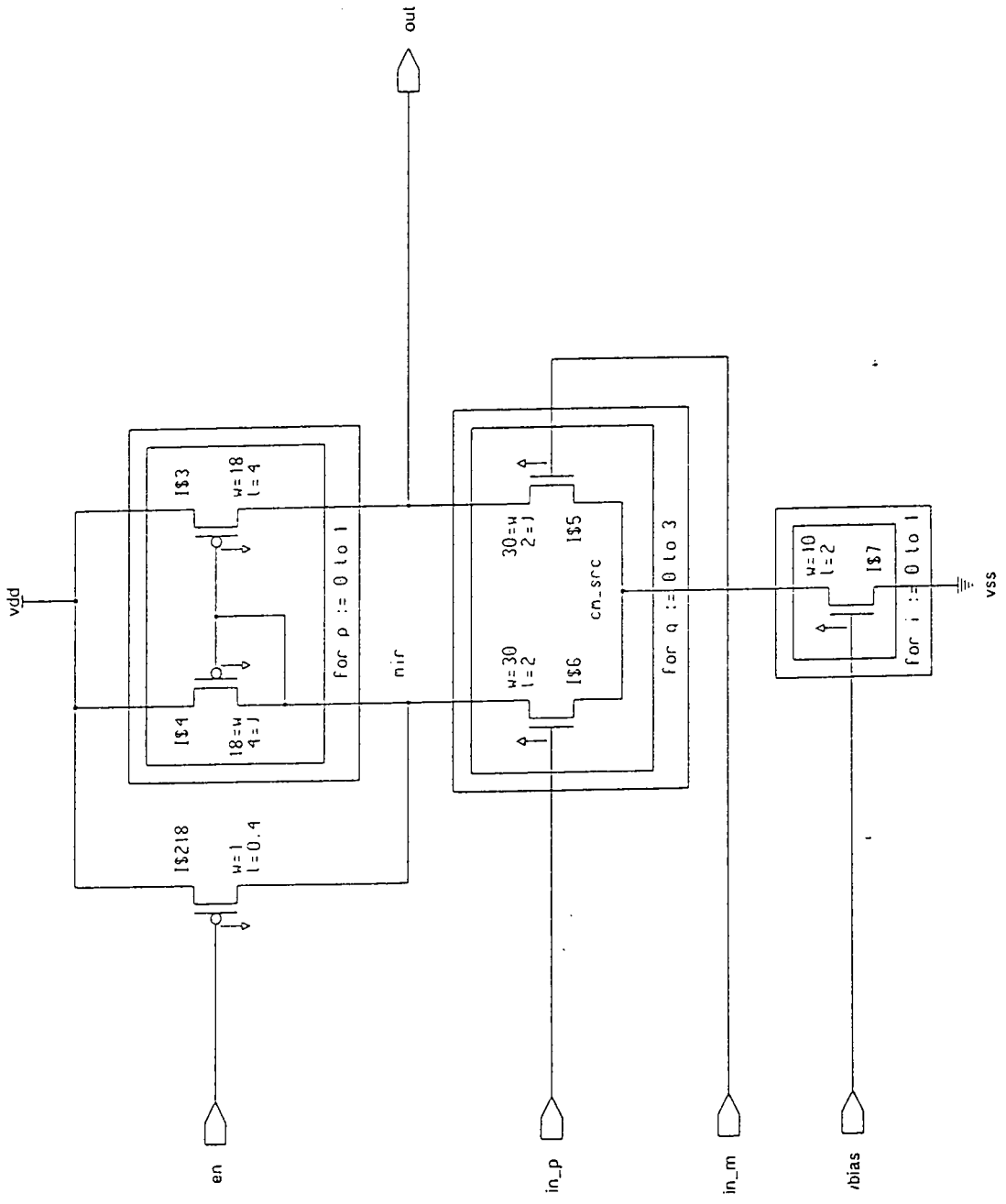
Reference Current Generator #2 Schematic
(c_gen2)



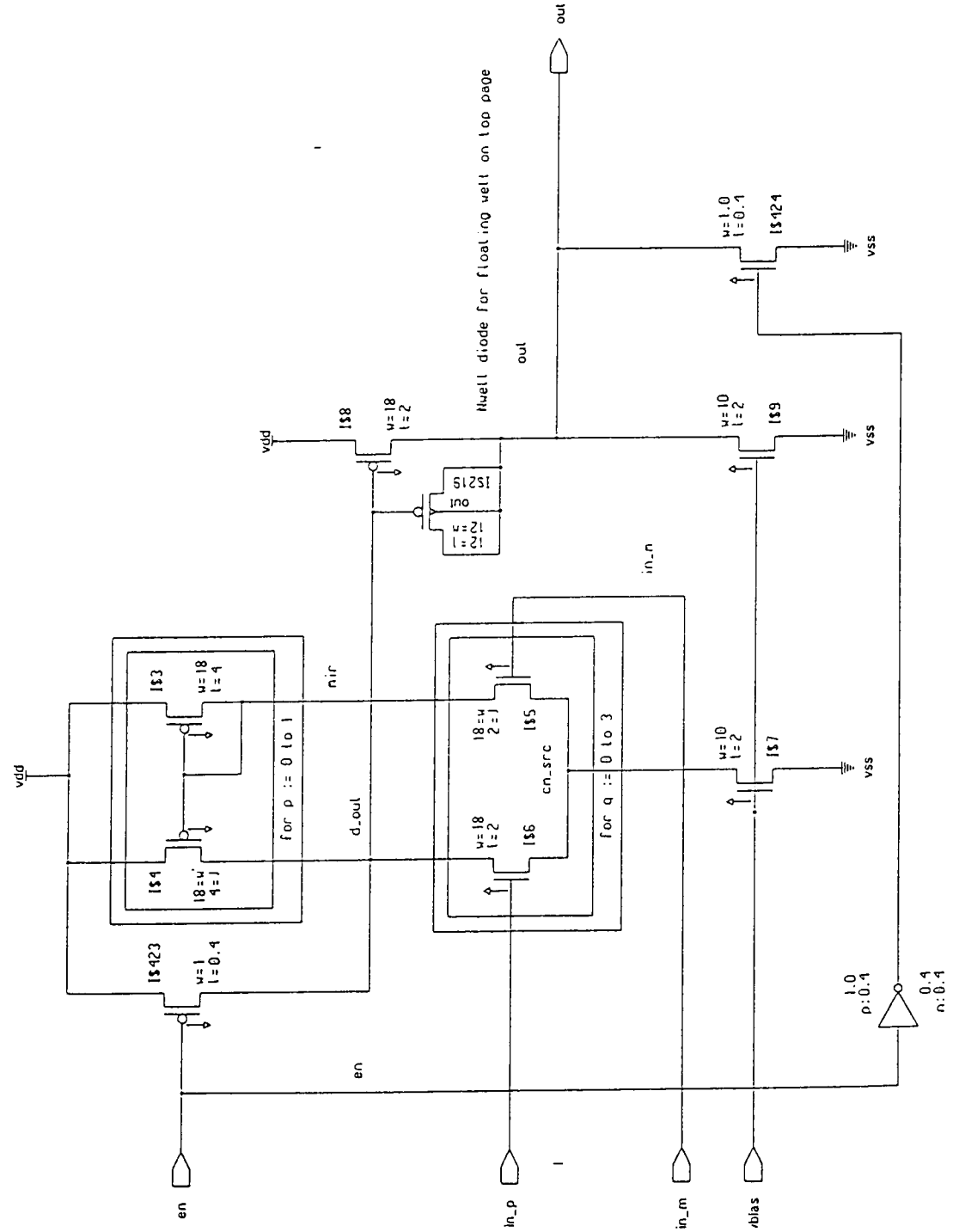
Tpbias Voltage Buffer Schematic
(vtpb_buf9)



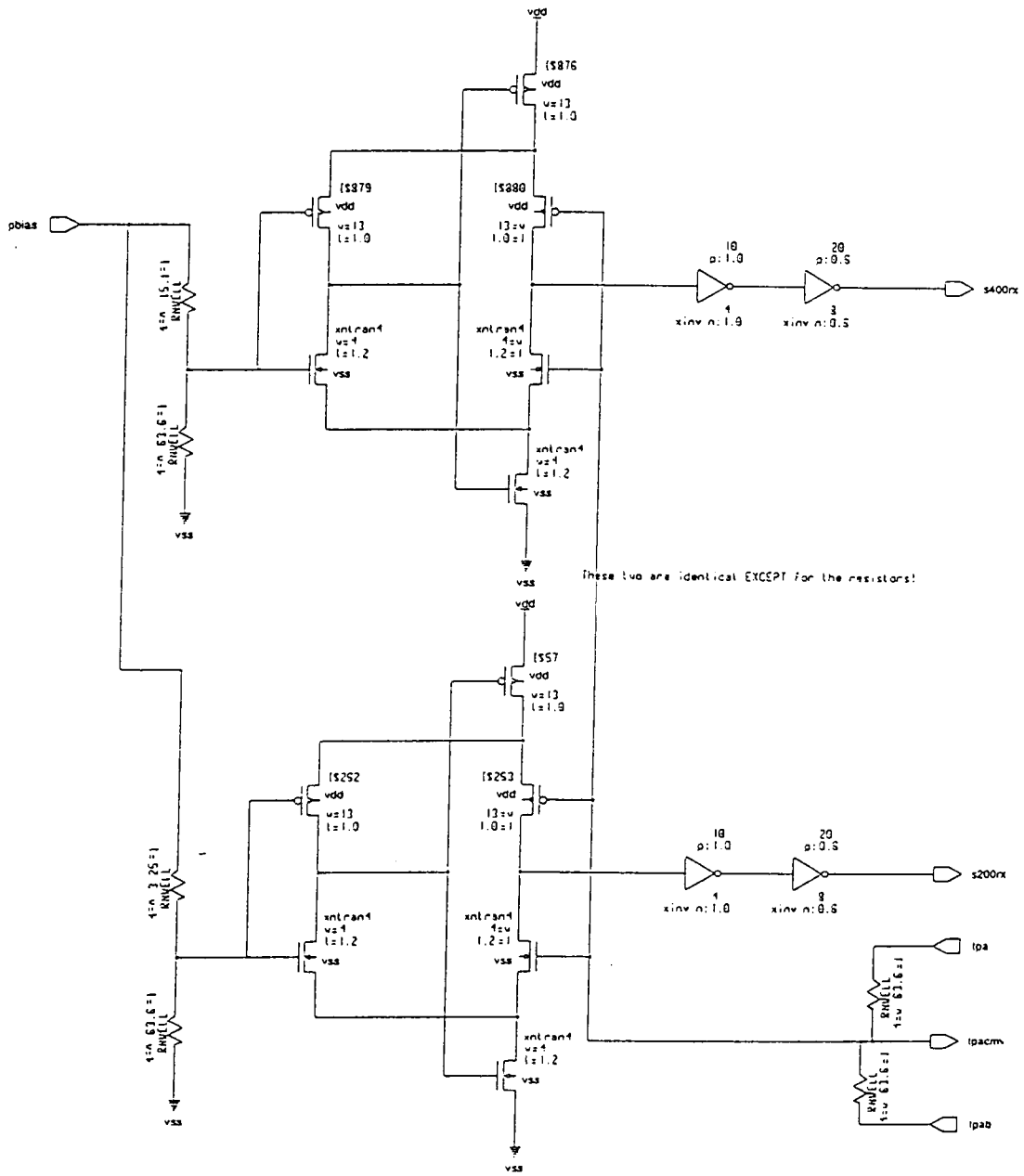
Reference Voltage Generator Schematic
(vgen_buf)



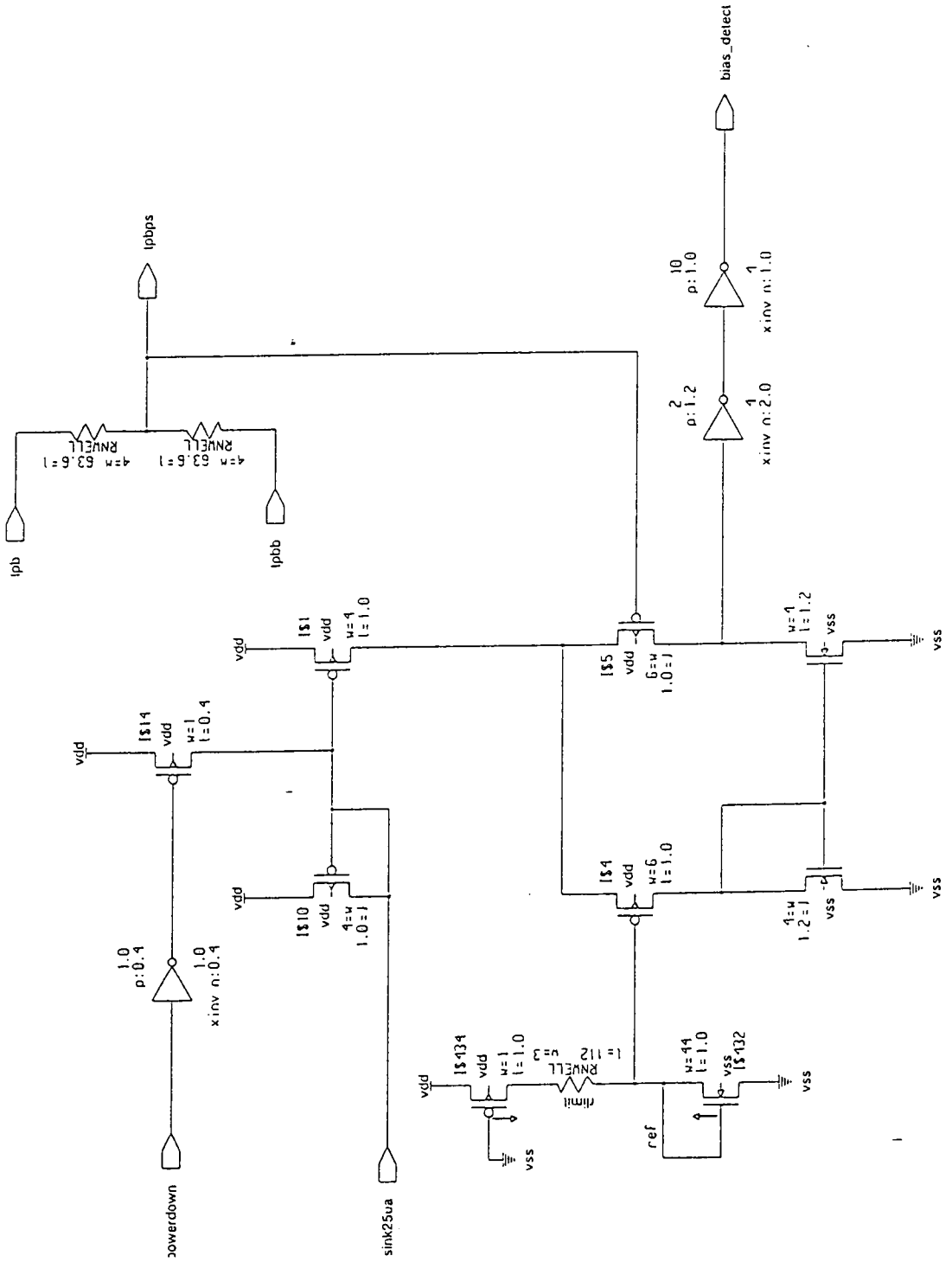
Reference Current Generator Schematic
(cref_amp)



Speed Reception Schematic
(speed_rx)



Port Status Detection Schematic
(bias_detect)



The following schematics may be found in the Motorola Firewire Analog I/O Schematics document:

Data and Strobe Driver Schematic (ds_driver)
Data and Strobe Receiver Schematic (ds_receiver)
Arbitration Reciever Schematic (arb_rx)
Speed Transmission Schematic (speed_tx)
Band Gap Starter Circuit (bg_kick)

VITA

Vinton Lee James IV was born in Austin, Texas, on May 20, 1964. He graduated from Port Aransas High School in Port Aransas, Texas, in May 1983. In September 1983, he enrolled at Texas A&M University, College Station, and received a Bachelor of Science degree of Electrical Engineering in August 1987. In September 1990, he enrolled at Texas Tech University, Lubbock, and received a Master of Business Administration in August of 1992. In September 1993, he enrolled in the distance learning program at the University of Tennessee, Knoxville, while living in Chattanooga. He graduated with a Master of Science of Electrical and Computer Engineering in December of 1998. Mr. James is registered as a Professional Engineer in the state of Texas.

Mr. James was employed with Texas Instruments from 1983 to 1992. He worked with microprocessor-based systems in Temple, Texas, and microprocessor device fabrication in Lubbock, Texas. He has also been employed with the Card Monroe Corporation (CMC) in Chattanooga, Tennessee, from 1992 to 1994. He held the positions of Research and Development Engineer, then Electrical Department Manager. CMC produces carpet tufting machines. Mr. James worked with motor control technology and manufacturing operations. In September of 1994, Mr. James entered employment with Motorola in Austin, Texas. He serves as a Product Engineering Team Leader on Motorola's 68000 and Coldfire architecture core-based devices.