Evaluation of the GTL Technology for Use in the *T Network

by

Satoshi Asari

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degrees of

Bachelor of Science

and

Masters of Engineering in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1994

Author .....
Department of Electrical Engineering and Computer Science
May 16, 1994

Certified by .....
Andy Boughton
Thesis Supervisor

Accepted by .....
F. R. Morgenthaler
Chairman, Departmental Committee on Graduate Students

Evaluation of the GTL Technology for Use in the *T Network

by

Satoshi Asari

Submitted to the Department of Electrical Engineering and Computer Science
on May 16, 1994, in partial fulfillment of the
requirements for the degrees of
Bachelor of Science
and
Masters of Engineering in Electrical Engineering

Abstract

*T is the latest undertaking of the Computation Structures Group at the Laboratory of Computer Science. It is an attempt to use RISC processors in a massively parallel computing environment. The interconnection network of this system will be driven using a low voltage swing CMOS technology called GTL. This thesis explores the performance of this technology in an environment comparable to that of the *T network. This was done in two stages. First, a series of simulations were performed using a simulation tool called HSPICE. Various signal forms were transmitted across a transmission environment, and the resulting waveforms were examined. The results obtained were encouraging in that the outputs showed very little distortion. Second, a test board was constructed in an attempt to confirm these results. A test chip was obtained which contained several GTL drivers and receivers. A transmission environment was constructed using these chips as part of a transmitting and receiving protocol which allowed the detection of transmission errors.

Thesis Supervisor: Andy Boughton
Title:

Acknowledgments

I would like to take this opportunity to extend my thanks to my advisor, Dr. Andy Boughton. I am very grateful for the guidance and knowledge that he has provided for me during the 4 years that I have worked for him. I would also like to thank Jack Costanza and Ralph Tiberio for all their help and patience. Last but no least, I would like to thank Gowri Rao whom I had the pleasure to work with during the first stages of this project.

Contents

List of Figures

Chapter 1

Introduction

1.1 Dataflow Model of Parallelism

The *T (pronounced 'Start') project, currently being conducted by the Computation Structures Group at the Laboratory for Computer Science, is the latest step in the evolution of dataflow architecture. Dataflow graphs are perhaps the most developed model for fine-grain parallelism. Normally, these graphs are generated by compilers from functional languages. A dataflow graph consists of nodes which are connected by arcs. Each node specifies the operation to be performed on the values that flow on tokens along the arcs. These operations need not be single instructions. They may themselves be organized as a dataflow graph or as a linear sequence of instructions. The latter case is called multithreading and each sequence of instructions is called a thread. Each token also carries a continuation consisting of a node number, an instruction pointer, and a frame pointer. During the processing of a token, a waiting/matching store is checked to see if the token's partners, if any, have arrived already. If the partners are found (successful synchronization), the node is fired (the operation specified by the node is executed). If the partner is not found, the token is placed in the waiting/matching store, and the next token is processed. It is important that a value that fails to synchronize not block the processing of other tokens. Such a situation could lead to a deadlock state.

The multithreaded dataflow model requires some hardware support for efficient

operation. Since each of the nodes can be executed on a different processor, a large amount of communication must take place amongst the processors. Given this fact, a fast processor/network interface would noticeably improve the efficiency of the system. A mechanism for the synchronization process mentioned above is also necessary. In a dataflow graph all of the inputs to a node must be synchronized before the node is fired. Whenever a synchronization fails or a thread ends, a context change must take place. The ability to perform fast context switches will increase the efficiency of the system. Thread scheduling is another important issue. Bad scheduling can have a great impact on the efficiency. It is generally too expensive to incorporate software into the scheduling process, so a hardware queue is usually used.

1.2 Monsoon

There is existing proof that dataflow hardware can indeed be built in the form of Monsoon. Monsoon is the predecessor to *T and was designed and built by the Computation Structures Group in conjunction with Motorola. It incorporated fast processor/network interfaces, fast context switching, and fast synchronization. Although Monsoon was successful in these areas, it showed some weaknesses. For example, it's single thread performance was poor. This means that it did not perform well in executing a single sequence of instructions. Another big concern regarding Monsoon was the fact that it's processors are fully custom built. The speed of commercial processors exceed that of custom processors. This fact makes the use of conventional processors more desirable.

1.2.1 Processor Considerations

In the short term, the technology and marketing factors that will drive the design and advance of individual processors will be their suitability as components in single processor workstations, PC's, and small shared memory multiprocessor servers. Unfortunately, this means that these processors are less than ideal for use as processing elements in large scale parallel architectures. The major shortcoming of these

processors is the lack of mechanisms tailored for interprocessor communication and synchronization. The result is that communication and synchronization amongst nodes will have substantial execution overhead.

1.3 *T and *T-NG

*T (pronounced 'Start') is an attempt to use RISC processors in a system that will support the multithreaded model. The *T architecture is divided into three parts: the data processor, the synchronization processor, and the remote memory processor. The synchronization processor is responsible for receiving messages from the network, synchronizing the values with other input values, and notifying the data processor when the thread is ready to be executed. The remote memory processor handles remote memory requests from other nodes, and the data processor actually executes the instructions. The original realization of this architecture involved the use of the Motorola MC88110. The MC88110 was modified to contain a network interface unit, and hardware support for thread scheduling.

A major shift in Motorola's corporate strategy presented new opportunities to the *T project. This lead to a new *T model which is called *T-NG (StarT, the Next Generation). The main difference between *T and *T-NG is the addition of caches for global shared memory and the substitution of a 64-bit processor from the IBM and Motorola PowerPC family for the MC88110. An external network interface unit will be added to the processor. The remainder of the design remains similar to that of *T.

1.3.1 Arctic

The interconnection network will be built using a routing chip called ARCTIC (A Routing Chip That Is Cool). ARCTIC is a four input four output packet switch and is based on previous work with the successful PaRC routing chip. PaRC is a four input four output routing chip used to build the network for Monsoon and was designed chiefly by Chris Joerg, a graduate student in the group. ARCTIC, however,

has added routing support for use in a fat-tree topology, the network topology to be used in *T-NG. Also, the *T-NG network will have twice the bandwidth (200 MBytes/sec/link) of the Monsoon network.

Each link in this network is 16 bits wide and data will be transmitted at 100 Mbits/sec/link. All data links will use the GTL technology (a low voltage swing CMOS technology) and each bit will be transmitted by a single ended GTL driver. The performance of this configuration, in particular that of the GTL technology, is important to the performance of the network.

1.4 Thesis Overview

This thesis will explore the use of the GTL technology in the implementation of the *T network. Its performance in an environment comparable to that of the *T network is evaluated through a series of simulations and the construction of a test board. The simulations will be done using a circuit simulation tool called HSPICE. The test board will verify the results obtained in the simulation phase. The basic idea of the test board is to transmit random bit sequences over a transmission media and to detect errors at the other end.

Chapter 2

Analytical Model

This section deals with the development and execution of a simulation model of the transmission configuration. Simulation was done using a tool called HSPICE. HSPICE is an analog circuit simulator made by Meta-Software Inc. It performs steady-state, transient, and frequency domain simulations of electrical circuits. HSPICE takes a netlist file as input. This file contains the circuit descriptions, analysis and simulation control statements, control options, and submodule routines amongst other things. Outputs can be viewed in many forms including textual and graphical forms. For the purposes of this project, the graphical form of the outputs prove to be the most useful. There are some issues which need to be addressed as the model is being developed. One of these issues is the representation of the transmission lines to be used. Work in sections 2.1 and 2.2 were done in conjunction with Gowri Rao.

2.1 Transmission Lines

2.1.1 Physical Description of Traces

There are two different traces involved in the configuration. There is the Printed Circuit Board (PCB) trace and the Multi-chip Module (MCM) trace. The PCB traces are the lengthy traces which connect the nodes of the system. The MCM traces are the short on chip traces. The PCB traces were made with copper and the MCM

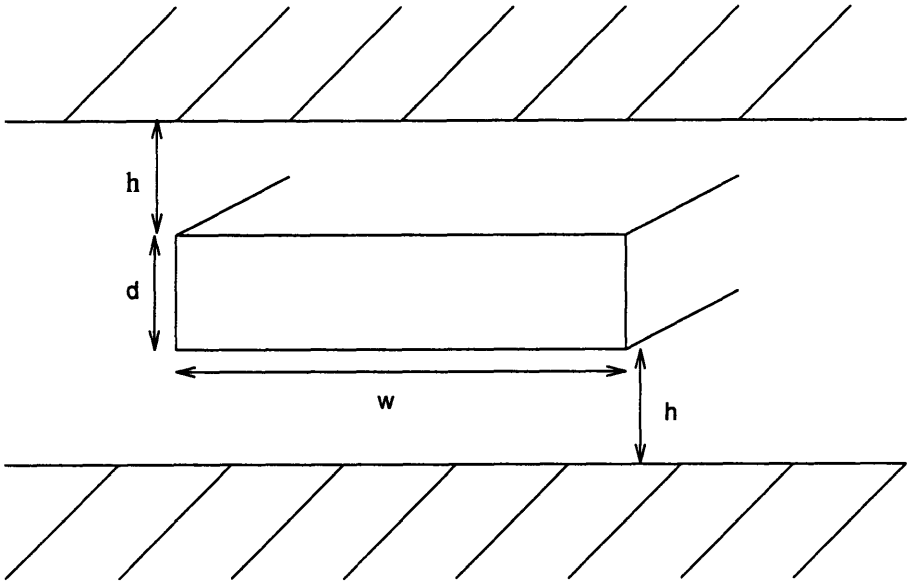

Figure 2-1: Stripline Transmission Line Configuration

traces were made with Molybdenum. Both types were matched to 50 ohms and were constructed in the stripline configuration, which is shown in Figure 2-1. It shows two ground planes which sandwich a conductor surrounded by dielectric material. Some of the factors which determine the transmission characteristics of the trace are the resistivity of the conductor, the dielectric constant of the material surrounding the conductor, the frequency of the transmitted signal, and the dimensions labeled d, w, and h in the figure. In this application, $w = 8mils$ and $h = 4mils$.

2.1.2 Skin Effect

When the frequency of the transmitted signal becomes very high, the skin effect becomes an important issue. At high frequencies, the electromagnetic fields can only penetrate a small distance into the conductors. The distance to which the fields penetrate the conductor is called the skin depth (δ) and is dependent on the frequency. The effect that this phenomenon has on the resistance of the conductor is referred to as the skin effect.

The skin depth can be calculated using the equation obtained from [4]

$$\delta^2 = \frac{2}{\omega\mu\sigma} \tag{2.1}$$

where the values for μ and σ are

$$\mu \approx \mu_0 = 4\pi \times 10^{-7} \tag{2.2}$$

and

$$\sigma = 5.8 \times 10^7 \Omega^{-1} m^{-1} \tag{2.3}$$

for copper. At a frequency of 400 MHz, $\omega = 2\pi f = 800\pi \times 10^6 rad/sec$. Substituting these values into the above equation yields $\delta = 0.13 mil$ for copper at 400MHz. The resistance per unit length can be found by, from [4],

$$R = \frac{1}{A\sigma}\Omega in^{-1} \tag{2.4}$$

where A is the area in which the current is flowing. This area is equivalent to the area in which electromagnetic fields can be found. So, $A = 2\delta w = 2.08 \times 10^{-6} in^2$. Substituting, $R = 0.3263\Omega in^{-1}$. The next step is to calculate the attenuation (α) of the line. This can be found using

$$\alpha = \frac{R}{2Z_0} + \pi f C Z_0 tan\delta \tag{2.5}$$

where $tan\delta = 0.03$ and $Z_0 = 50\Omega$. This δ is not the same δ as the one that represented the skin depth. Before this value can be evaluated, a value for C must be obtained. Take the two equations

$$Z_0 = \sqrt{\frac{L}{C}} = 50\Omega \tag{2.6}$$

and

$$\frac{1}{\sqrt{LC}} = \frac{c}{\epsilon} = 1.5 \times 10^8 \tag{2.7}$$

In this case since c is the speed of light and equals 3×10^8 and $\epsilon = 4$ for the dielectric

being used. Multiplying (2.6) and (2.7) and inverting

$$C = \frac{1}{50 \times 1.5 \times 10^8} = 1.33 \times 10^{-10} \frac{F}{m} \tag{2.8}$$

Substituting into (2.5) yields the attenuation value of $\alpha = 1.12 \frac{dB}{ft}$. Similar calculations were done for 100 MHz and 200 MHz. At 100 MHz, $\alpha = 0.393 \frac{dB}{ft}$ and at 200 MHz, $\alpha = 0.653 \frac{dB}{ft}$.

HSPICE is not equipped to compensate for the skin effect. Therefore, the effect must be incorporated into the model. This can be done simply by setting the parameter d to the appropriate value. This value is 2δ since that is the area in which current is actually flowing. This assertion was tested by sending pure sinusoids at 100 MHz, 200 MHz, and 400 MHz down a transmission line model with the appropriate values of 2δ. The attenuation was compared to the values of α derived above. The results were satisfactory. Unfortunately, the signals to be sent in the final model are not sinusoids, but are pulse like waveforms. These types of signals consist of a wide range of frequency components, so it is difficult to choose a value for d. It was decided that the value of δ for 400 MHz would be used in the transmission line model.

2.2 Transmission Model

The transmission process involves more than just the copper PC Board trace. MCM traces to represent the on chip traces, a driver which is where the signal originates, and capacitors to represent the various capacitances of the system must all be included. To obtain a HSPICE model for the Molybdenum MCM traces, steps in section 2.1 were followed with the conductance of Molybdenum being substituted for that of copper and the proper value of the dielectric constant. A model of this transmission process can be seen in figure 2-2. A simple switch was used to represent the GTL driver in this model.

Two types of inputs were used to run on this model. One was a 10 ns pulse and the other was a square wave with a 20 ns period. An example of a pulse input can

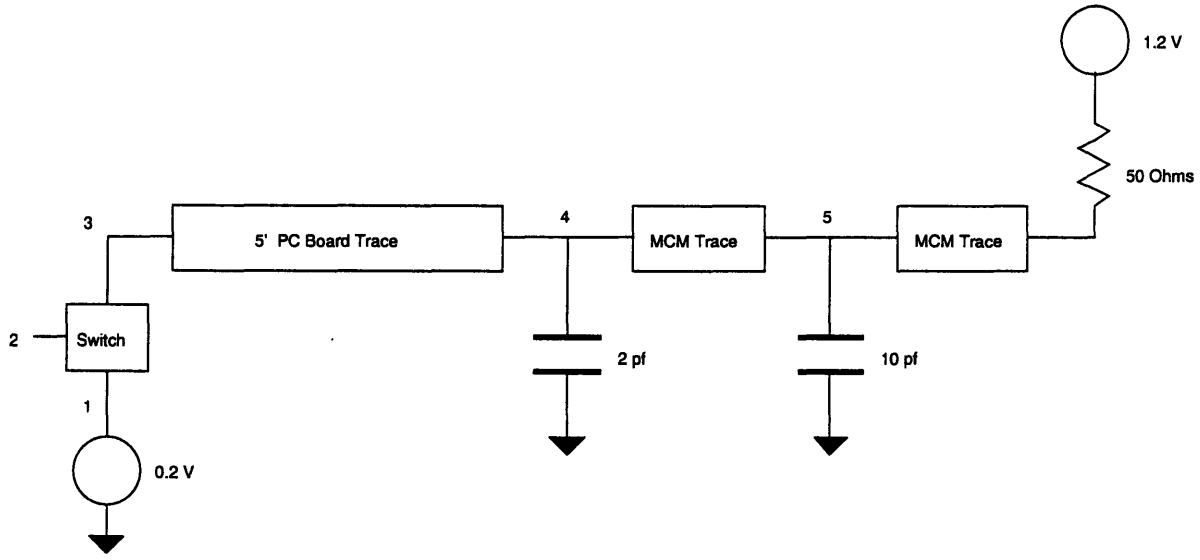

Figure 2-2: System Model with Switch Acting as Driver

be seen in figure 2-3. The transition time from the low value to the high value and vice versa was taken to be 1 ns.

The results of running a simulation on this model with a square wave input are shown in figures A-1 through A-4 in the appendix. These are plots of voltages taken at nodes 3, 4, 5, and 6. The voltage at node 5 was of most interest since that is the node at which a GTL receiver would be attached. Voltages at other nodes were plotted for the purposes of comparison and analysis. There is quite a bit of ringing evident in these plots, particularly at node 4. After careful observation of the plots and the timing of the ringing, the conclusion that the ringing was caused by the discontinuities in the transmission model was reached. A slight mismatch in the impedances on opposite sides of the node led to reflections at each of the nodes. In spite of this problem, the signal at node 5 seems to retain all of the important characteristics of the square wave input. The high and low voltage levels remain well above and below the threshold voltage of 0.8 V. In summary, the results obtained from these simulations are encouraging.

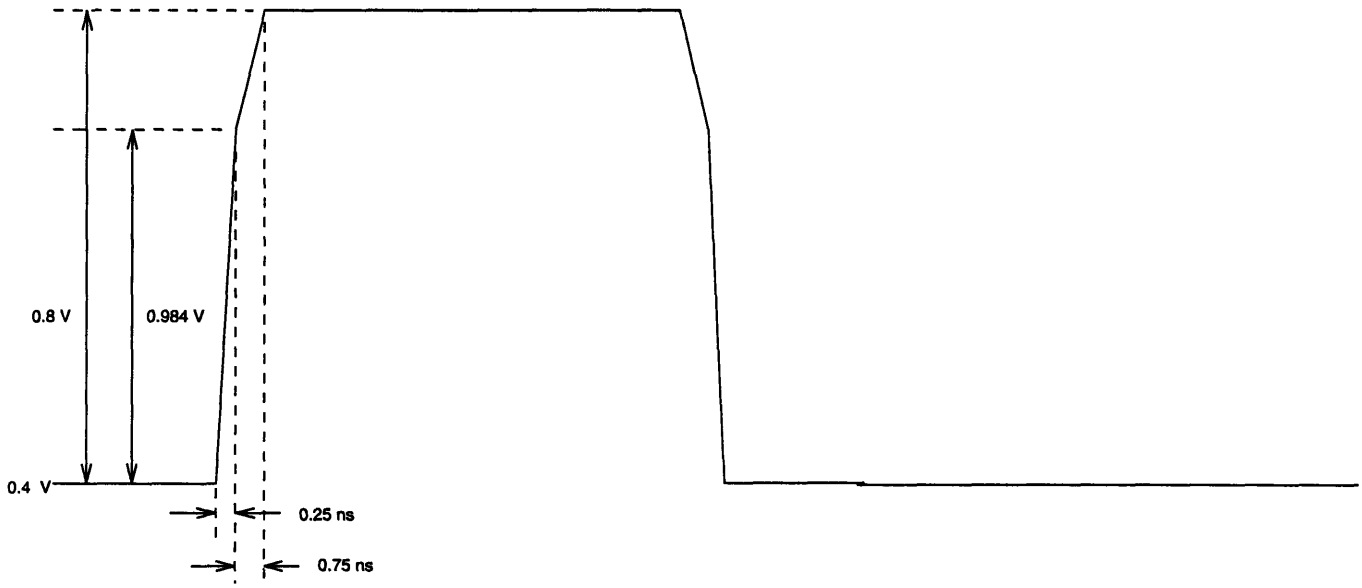

Figure 2-3: Diagram of Pulse Used in Simulation

2.3 Transmission Model with GTL Drivers

The above model used a simple switch to represent the GTL driver, and the signal was simply viewed at the point where a GTL receiver would be present. Once the appropriate HSPICE models were obtained, the switch was replaced by the model for the driver. Also, the model for the receiver was placed in the appropriate location. A graphical representation of this new model can be seen in figure 2-4. The same inputs as in section 2.2 were used to run on this model.

The result of running a simulation on this model with a square wave input to the driver is shown in figure A-5 of the appendix. In this plot, V(PAD) represents the voltage at the output of the driver, V(5) represents the input to the receiver, and V(DI2) represents the output of the receiver. V(PAD) shows some ringing, particularly at the high voltage. This is do to the reflections from the various nodes. V(5) shows less ringing and maintains the desired square wave shape. Also, it is centered at around 0.8 V which was used as the threshold voltage in the receiver. V(DI2), the output of the receiver, is a nearly perfect square wave. The period remains at 20 ns and the duty cycle is very close to 1/2. The duty cycle is the fraction of time that the square wave is at the high voltage. Since the receiver model

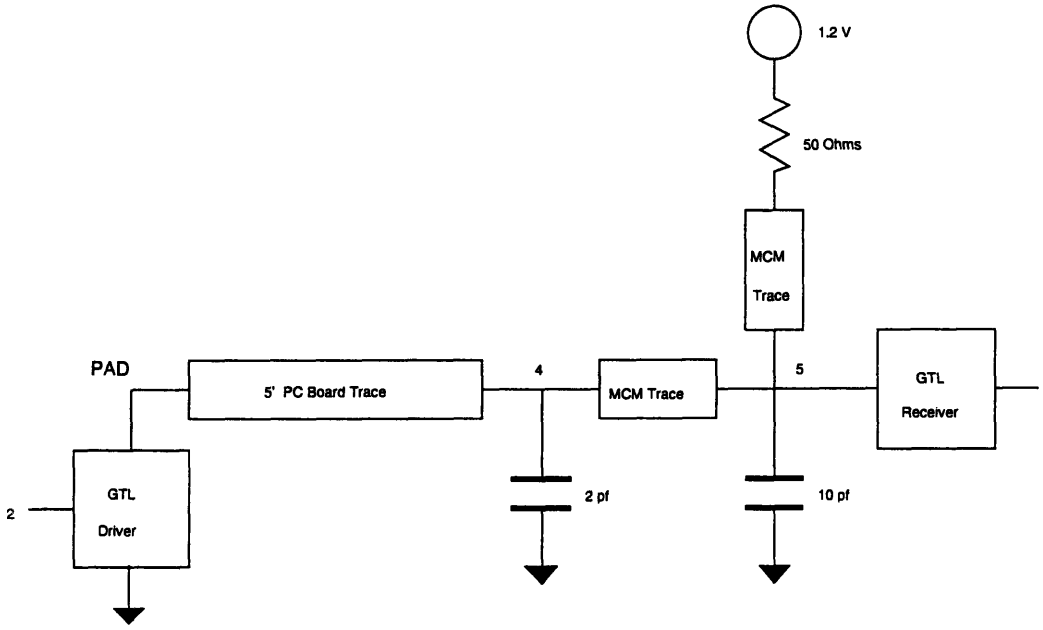

Figure 2-4: System Model with GTL Drivers and Receivers

used was for a inverting receiver, V(DI2) is inverted from V(5).

Due to these encouraging results, other simulations were run with a longer PCB trace. The results for PCB trace lengths of 8 ft, 10 ft, 12 ft, 15 ft, and 20 ft can be seen in figures A-6 through A-10 of the appendix. These plots show that, as the length of the PCB trace increases, the attenuation of the signal becomes greater. This attenuation is seen mostly at the low voltage level. The high level voltage stays near 1.2 V, whereas the low voltage steadily creeps upwards. This is due to the setup of the configuration. The voltage is pulled up to 1.2 V near the receiver. This allows for the voltage at node 5 to be restored to very near 1.2 V. This nonuniform attenuation has visible consequences. As the PCB trace length increases, the duty cycle of the receiver output begins to decrease. This means that the receiver output spends more time at the low voltage level than it does at the high voltage level. This effect becomes more obvious at lengths of 15 ft and 20 ft.

Chapter 3

Test Board

3.1 Overview

This section deals with the design and construction of a test board. The goal of this phase of the project is to verify the analytical results obtained in the previous sections. Test chips were made available by Motorola for use on this board. The test chips included several accessible GTL drivers and receivers. These drivers and receivers were used as part of a transmitting and receiving protocol which allowed transmission errors to be detected.

The basic idea of this protocol is as follows. A pseudo random sequence of numbers will be created and transmitted. The sequence will be received at the other end of the transmission line and the received sequence will be checked against the transmitted sequence. An error occurs whenever the received sequence differs from the transmitted one. There were 6 GTL receivers available and one of these was used to receive the transmitted clock signal. Therefore, there were 5 lines between the transmitter and the receiver which carried the test sequence and 1 line which carried the clock.

3.2 Overall Layout

The layout of the board is as shown in figure 3-1. The transmitter portion and the receiver portion of the board are connected by a set of 5 foot long PC Board traces.

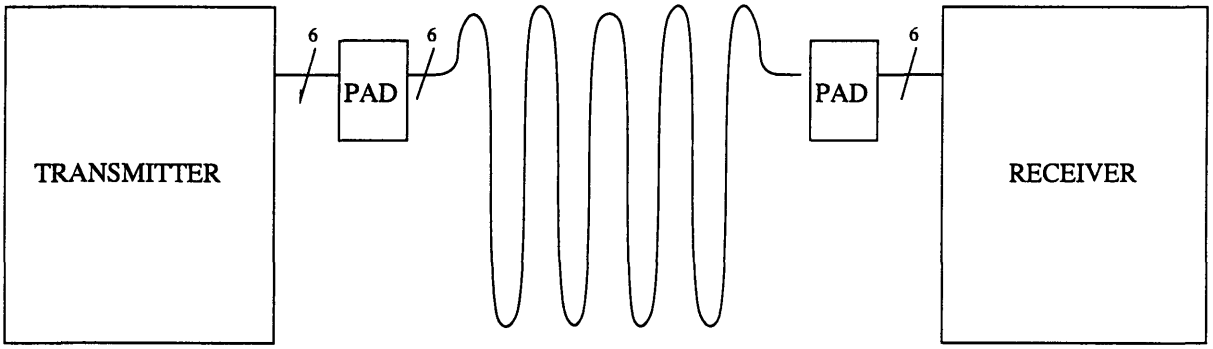

Figure 3-1: Overall Layout of the Test Board

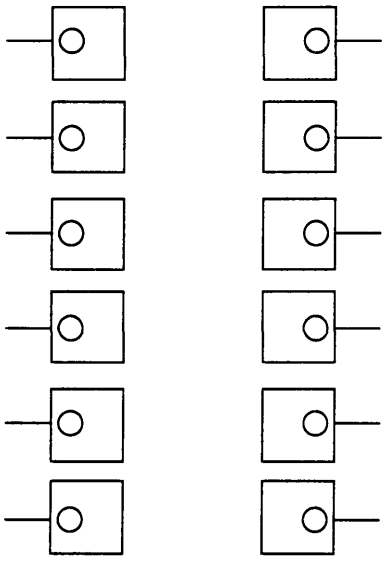

Figure 3-2: Diagram of PAD Used on Test Board

Rather than the traces be connected directly to the transmitter and the receiver, they were connected through a PAD. This was done so that transmission media, other than the PC Board trace, may be tested. A diagram of a PAD can be seen in figure 3-2. When the PC Board trace is the transmission media to be used, the left and the right columns of the PAD are connected by soldering a micro coax between them. Other transmission media may be used by soldering one end to the left column of the transmitter PAD and the other end to the right column of the receiver PAD.

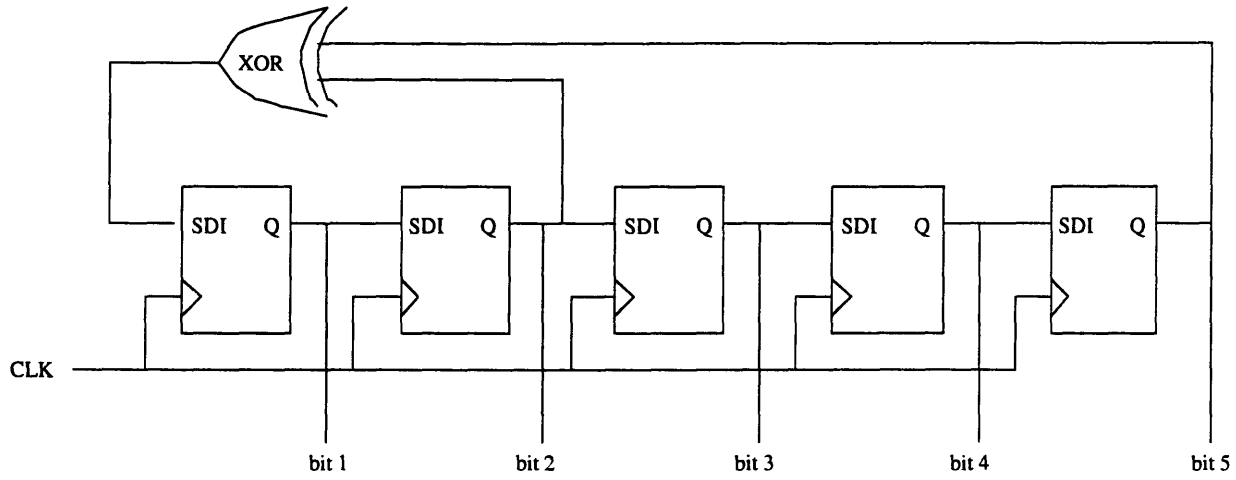

Figure 3-3: Shift Register Configuration Used to Generate 5 bit Pseudo Random Number Sequence

3.3 Transmitter

The transmitter portion of this board was responsible for generating the psuedo random sequence of 5 bit numbers and transmitting them along with the clock signal. A pseudo random sequence of 5 bit numbers can be generated with a configuration of shift registers shown in figure 3-3. A simple C program was written to ensure that such a setup produced the desired outputs. As mentioned in section 1.3.1, the data in the *T network will be transmitted at 100 Mbits/sec/link. This means that this configuration needs to generate numbers at that rate. This rate of output was unobtainable with the available discrete components. The combination of the setup times and the propagation delays of the shift registers and the XOR (exclusive or) gate were too large for such a rate to be achieved. However, PALs exist which function at the necessary speed. The above configuration was programmed into a high speed PAL. The PAL was driven with a 100 MHz oscillator which enabled the generation of pseudo random 5 bit numbers at 10^8 /sec. The code used to program the PAL can be found in the appendix. A CLR input was added to the PAL to enable the reset of the number sequence. The CLR signal was generated with a switch configuration shown in figure 3-4.

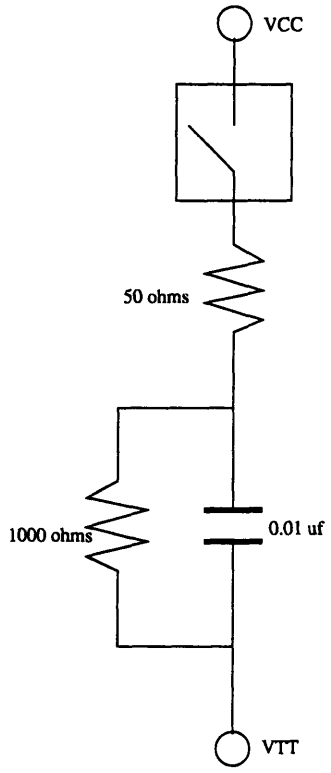

Figure 3-4: Switch Configuration Used to Generate CLR Signal

A schematic of the portion of the test chip used to drive the signals can be seen in figure A-11 of the appendix. There are components other than the driver itself present since this setup was constructed for a different purpose. The signal is input through a CMOS buffer, represented by the triangles along the left. The next component is a multiplexer. The SL (select) input to the multiplexer was set to always select input B, the signal. The output of the multiplexer goes to one of the inputs of a NAND gate. The other input to the NAND gate was set to always be high. This was done since the other input being set to be low will always produce a value of high at the output. Unfortunately, setting the input to be high inverts the signals as they go through the NAND gate. This fact will be accounted for in the error detection portion. Finally, the output of the NAND gate is connected to the GTL driver. The outputs were taken at nodes 21, 19, 18, 17, 15, and 14. A schematic of this portion of the board can be seen in figure A-12 of the appendix.

3.4 Receiver

The receiver portion of this board was responsible for receiving the data from the transmission line and detecting errors. The signals were received by the GTL receivers on the test chip and passed directly to a PAL along with the received clock signal. A diagram of the portion of the test chip used for receiving the signals is shown in figure A-13 of the appendix. These 6 components are all bidirectional GTL driver/receivers. These were all configured to act as receivers. This could be done by driving the input to the driver to the high voltage. The signals received by the PAD were connected to nodes 108, 112, 116, 122, 126, and 101. The outputs of the receivers were taken at nodes 107, 111, 115, 121, 125, and 99. The receivers present in these components are all inverting receivers. Recall that the signal was inverted in the transmitter stage as it was passed through the NAND gate. In essence, going through these inverting receivers reverses the effects of that inversion. Thus the error detection can be done in a straight forward manner, without worrying about inverted signals.

The error detection was done by taking advantage of the deterministic nature of the number generator, with the use of a PAL. Assuming that the currently received number is correct, the next correct number to be received can be determined using the same algorithm which was used to generate them at the transmitter. The subsequently received number was compared to this calculated number and whenever they did not match exactly, the error flag was raised. A clear input on the PAL lowered the error flag. The code used to program the PAL can be found in the appendix. A schematic of the receiver portion of the board is shown in figure A-14 of the appendix.

Chapter 4

Conclusion

4.1 Simulations

An HSPICE model for a transmission environment comparable to that of the *T network was developed. A series of simulations were performed in an effort to evaluate the performance of the GTL technology in such an environment. At first, a simple switch was used in place of a GTL driver, and the response to a square wave input was observed at various locations along the transmission model. The resultant waveforms showed some distortion in the form of ringing. These effects were more severe at some locations than at others. They were caused by reflections occurring at the nodes of the model. The observed waveform at the node of interest, namely the node at which the receiver would be present, seemed to retain all of the necessary characteristics.

Once the HSPICE models for the GTL driver and receiver were obtained, they were placed in the appropriate locations in the existing model. Simulations were performed using this new model and the same square wave input. The results of these runs were very encouraging. At a PCB trace length of 5 feet and a square wave period of 20 ns, the system performed very well. The output of the GTL receiver was a nearly perfect square wave with a period of 20 ns and a duty cycle of very close to 1/2. Due to these positive results, another series of simulations were performed with increasing PCB trace lengths. As expected, the attenuation of the signal increased as the trace length increased. This caused the duty cycle of the resulting output of

the receiver to decrease as the trace length increased.

Overall, the results obtained from the simulations were very positive. The system performed very well with a PCB trace length of 5 feet. The next step in the project was to design and construct a test board which would hopefully verify these results.

4.2 Test Board

A test board was designed as discussed in chapter 3. A pseudo random sequence of numbers would be transmitted by a set of GTL drivers across PCB traces. The signals will be received by a set of GTL receivers at the other end, and error detection will be performed. The error detection was done by taking advantage of the deterministic nature of the pseudo random sequence.

The design specifications were compiled and sent out to Mosher Design Services, a printed circuit board design company. Some check plots, which show the detailed layout of the board, were received. These can be found in figure A-15 of the appendix. Unfortunately, the fabrication of the board was not completed at the time that this thesis was written. The process is ongoing and is projected to be completed in the near future. Once the printed circuit board is fabricated, the board will be assembled and used to perform tests. Tests will be run using both the on board PCB traces and various cables to connect the transmitter and the receiver.

Appendix A

Figures

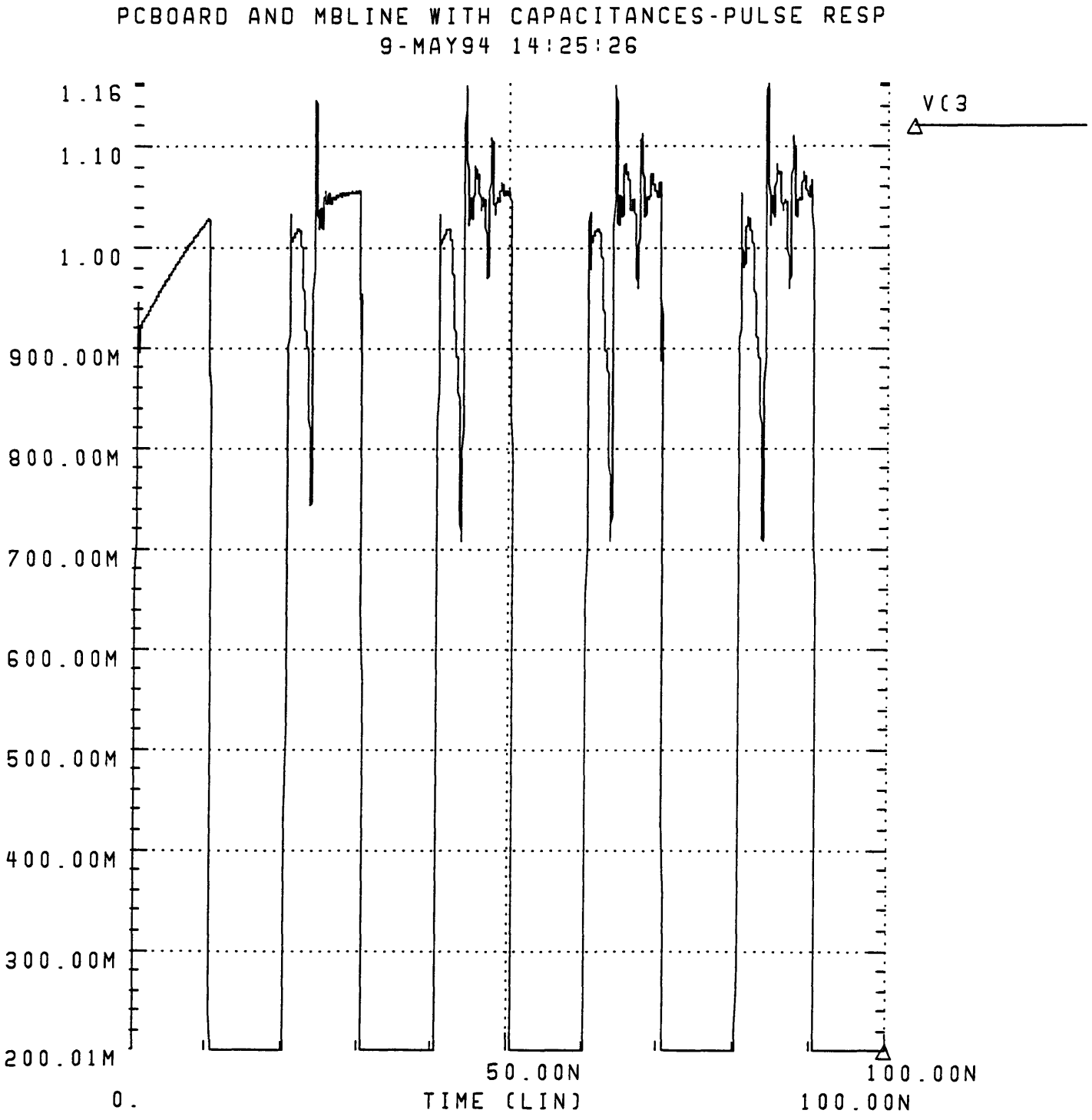

Figure A-1: Voltage at Node 3 for Simulation from Section 2.2

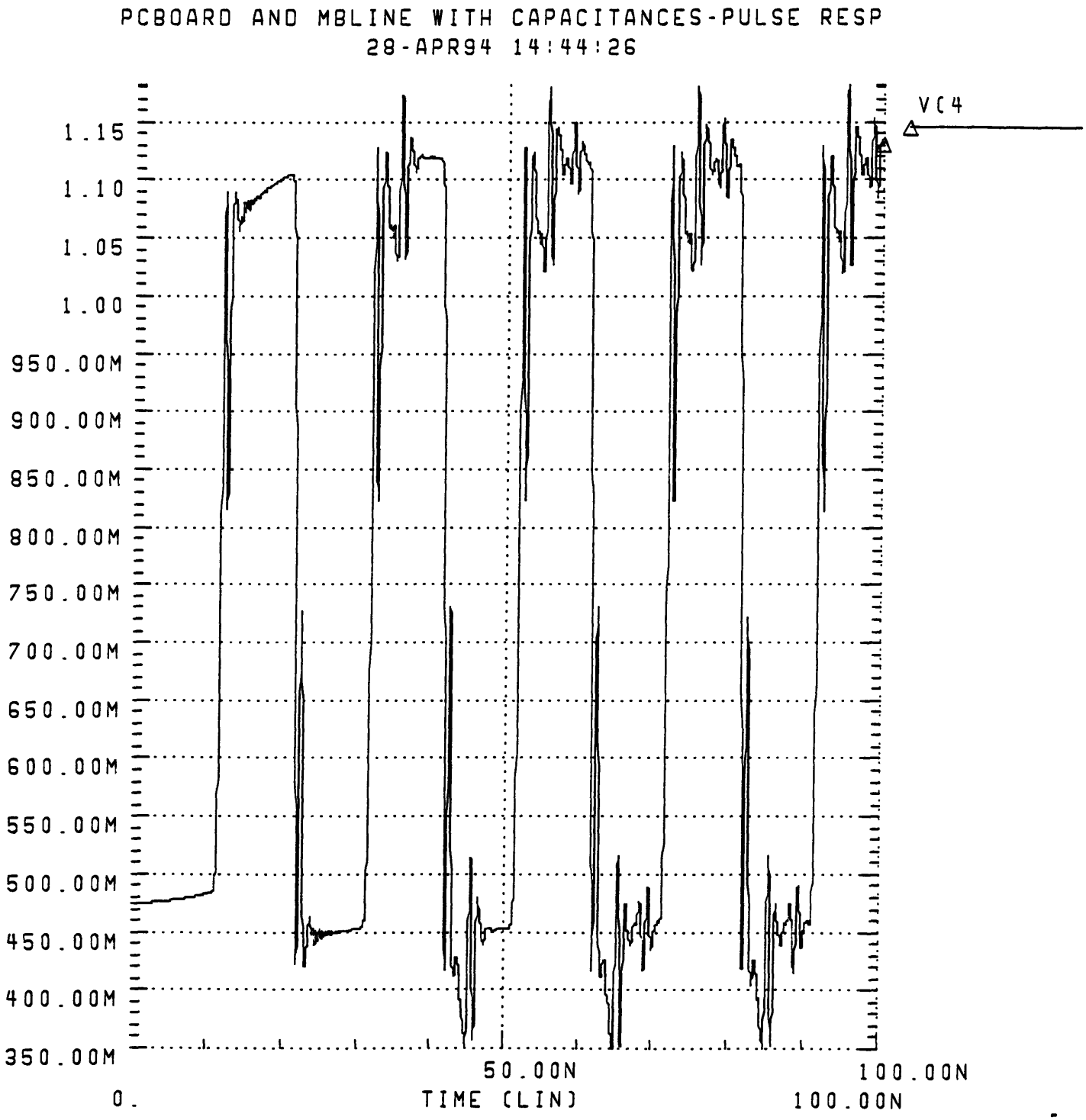

Figure A-2: Voltage at Node 4 for Simulation from Section 2.2

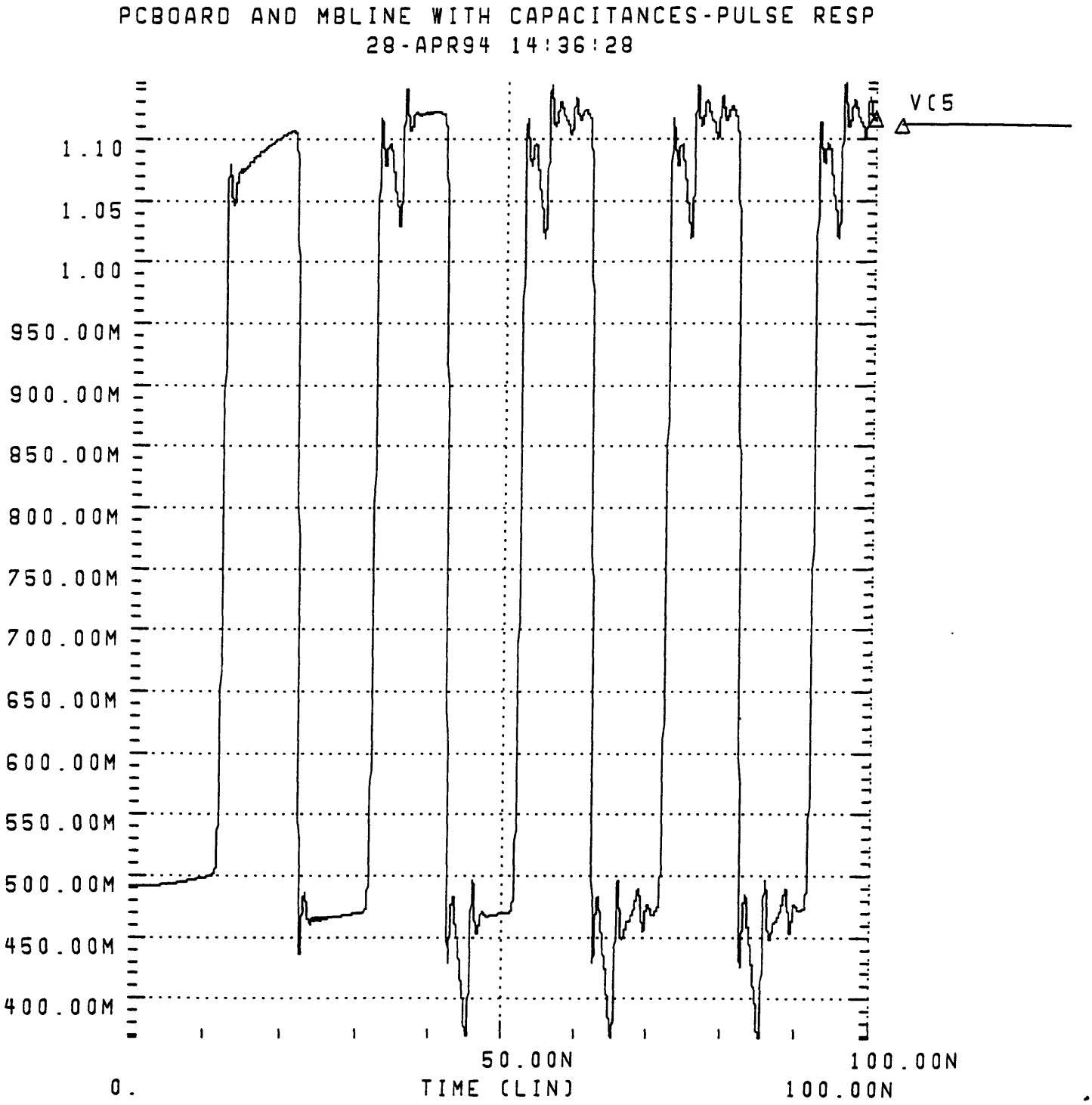

Figure A-3: Voltage at Node 5 for Simulation from Section 2.2

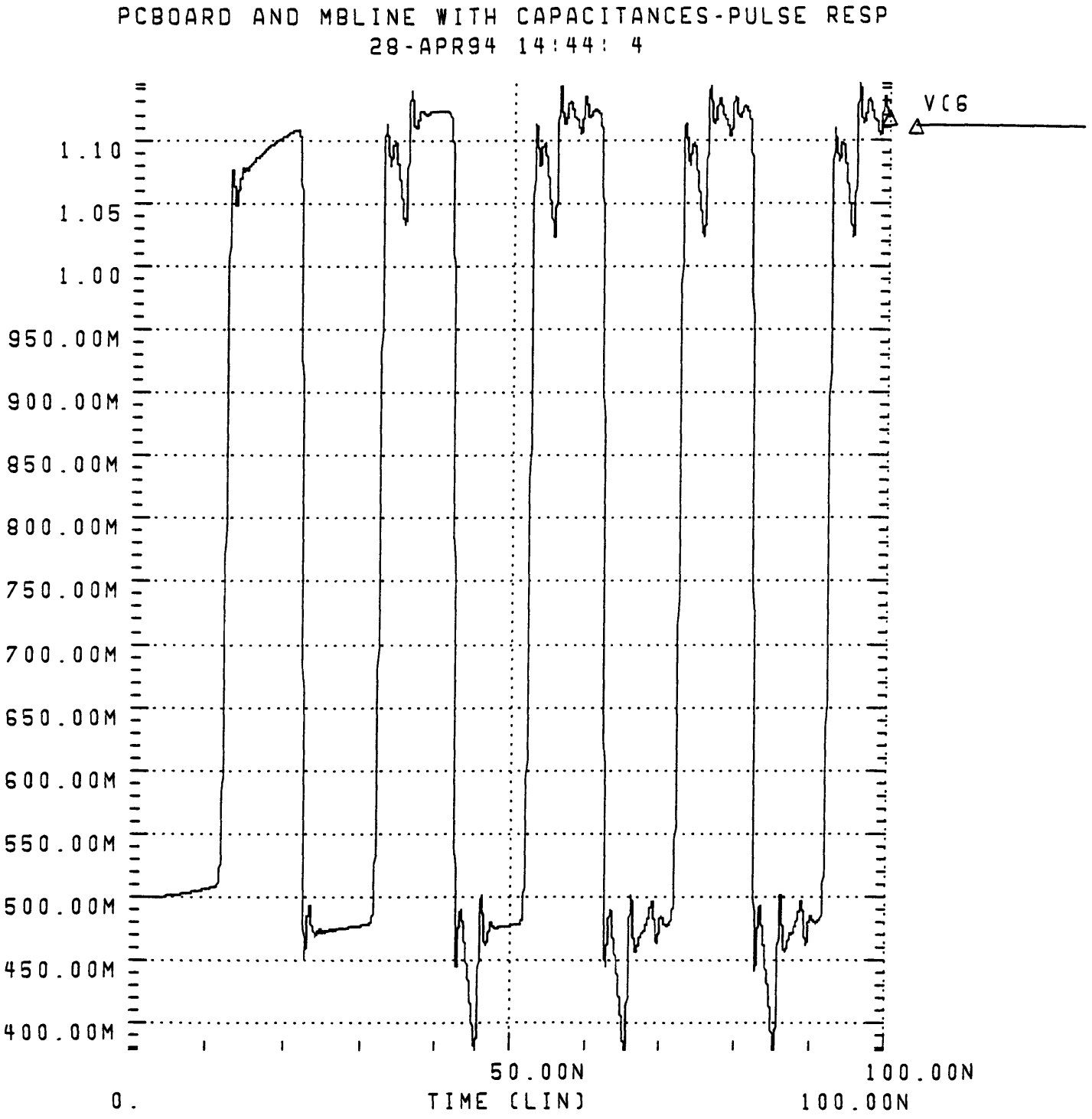

Figure A-4: Voltage at Node 6 for Simulation from Section 2.2

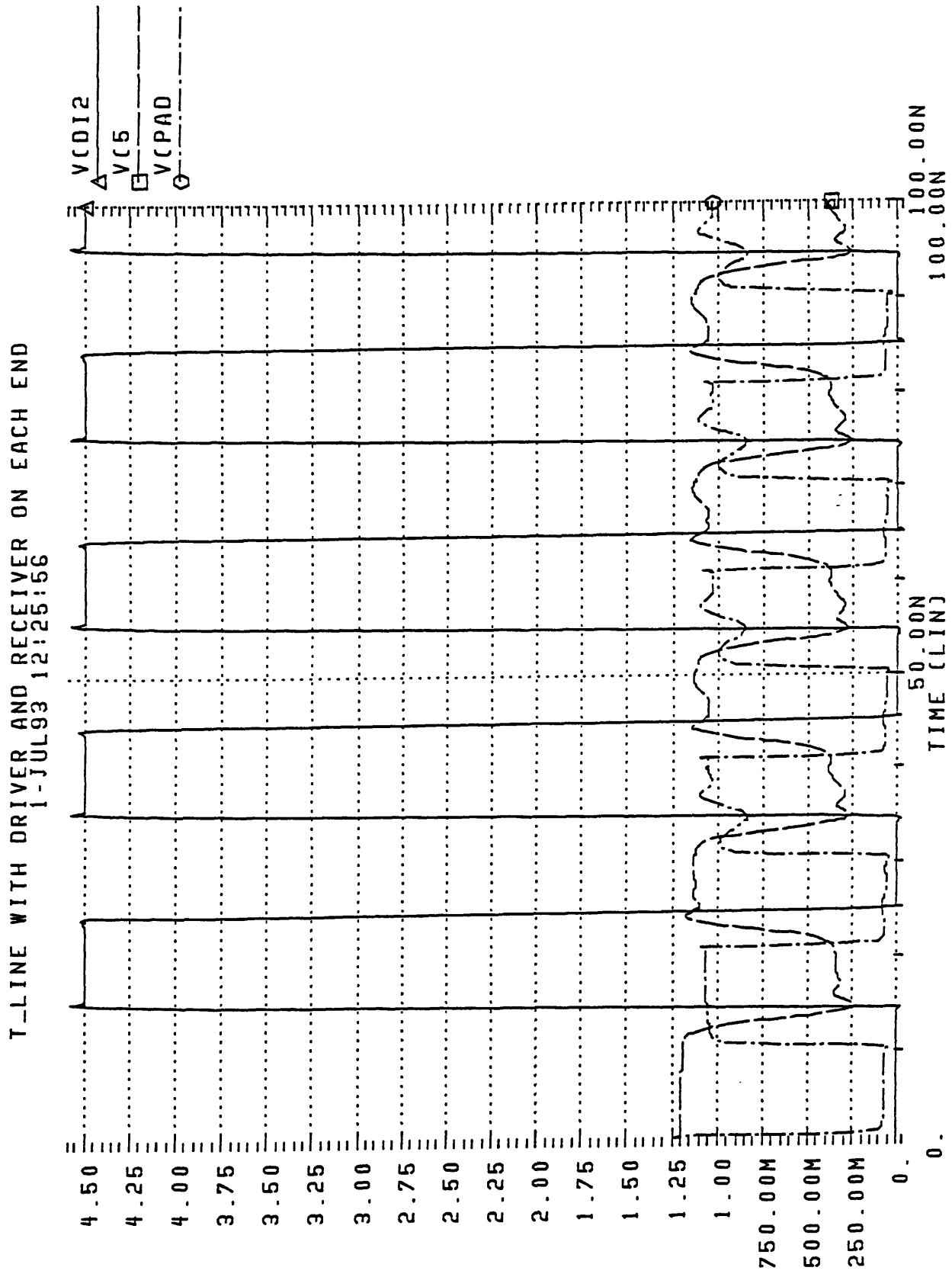

Figure A-5: Plot of Results for Simulation from Section 2.3

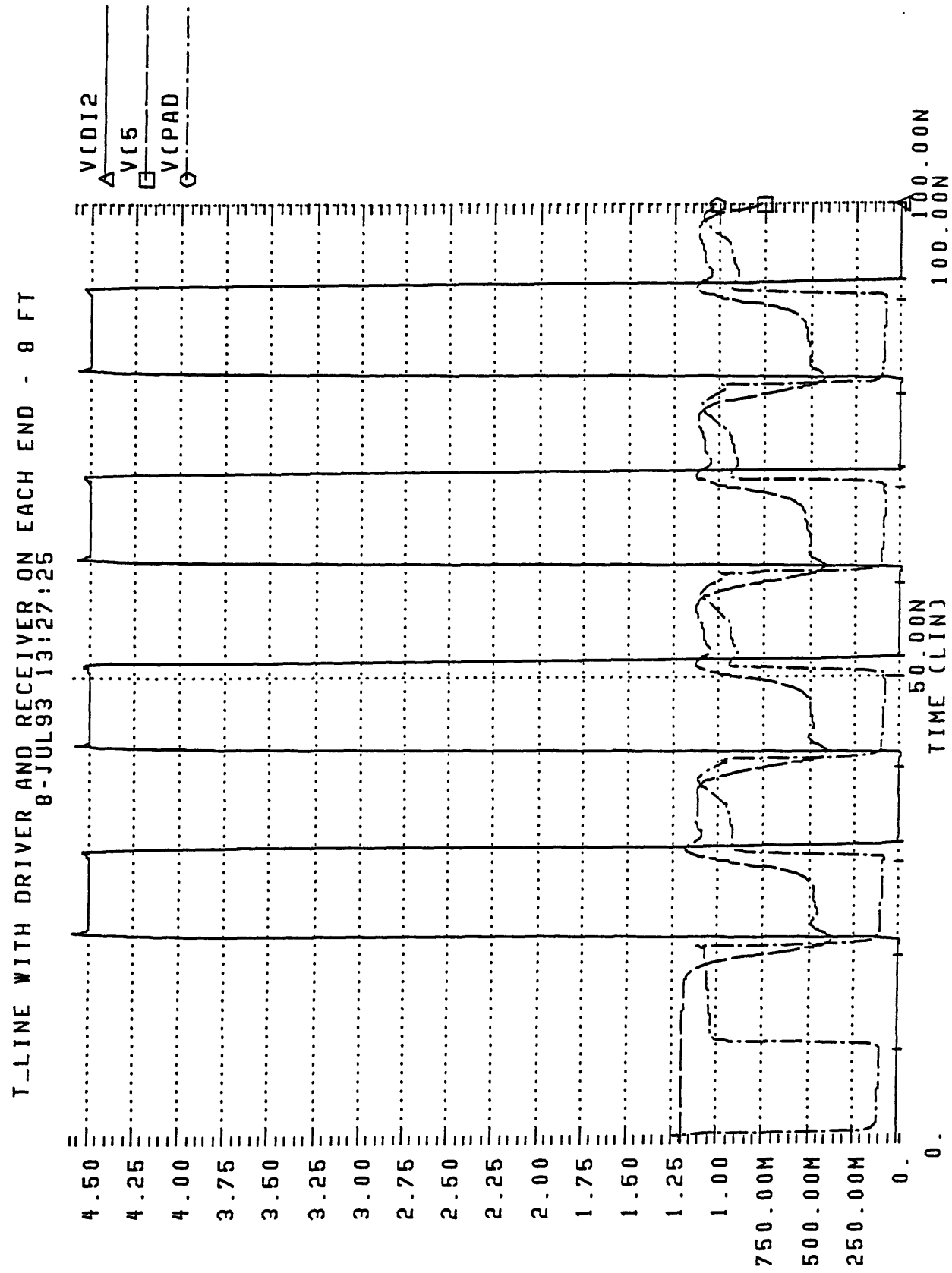

Figure A-6: Plot of Results with PCB Trace Length of 8 ft

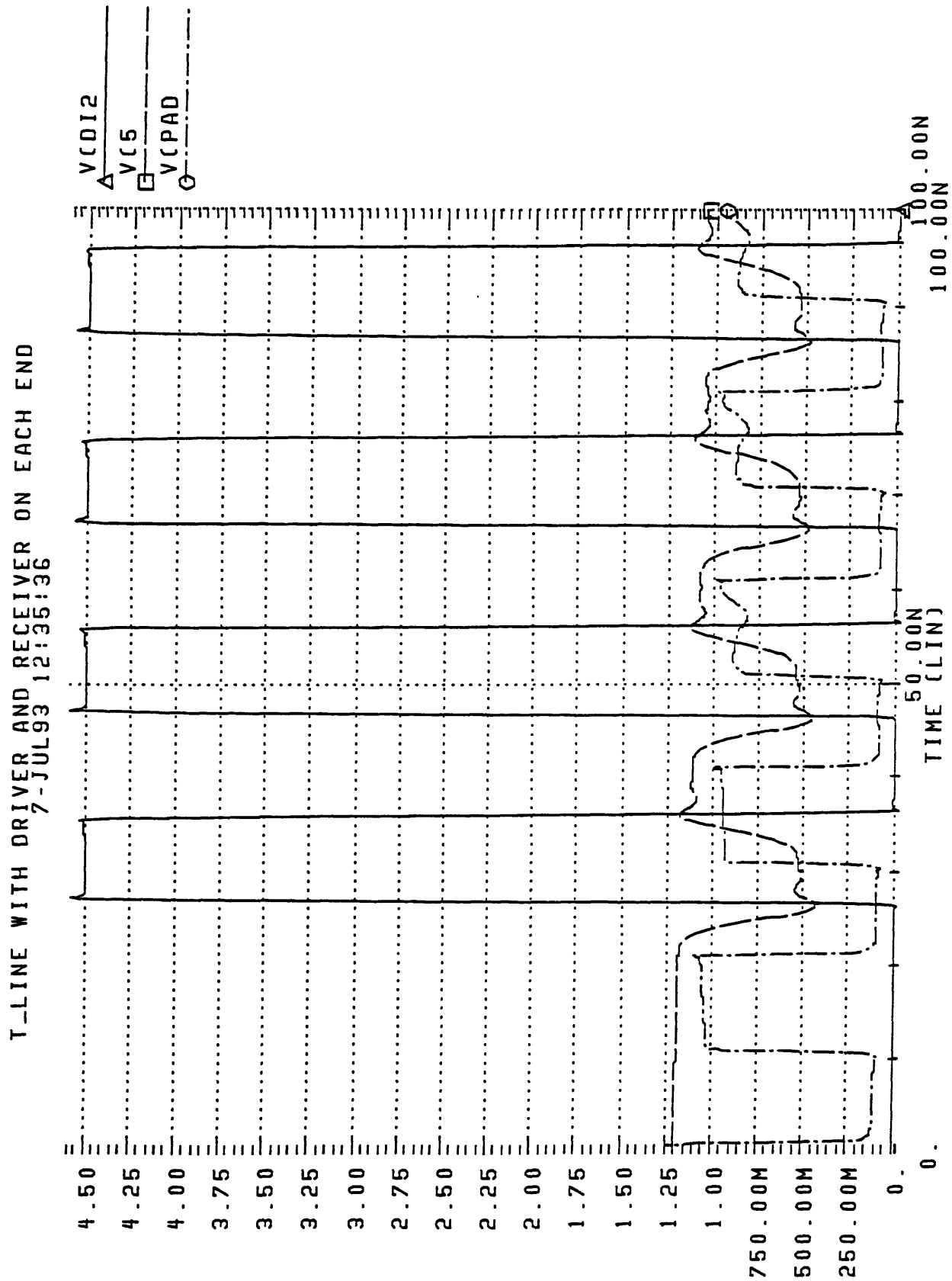

Figure A-7: Plot of Results with PCB Trace Length of 10 ft

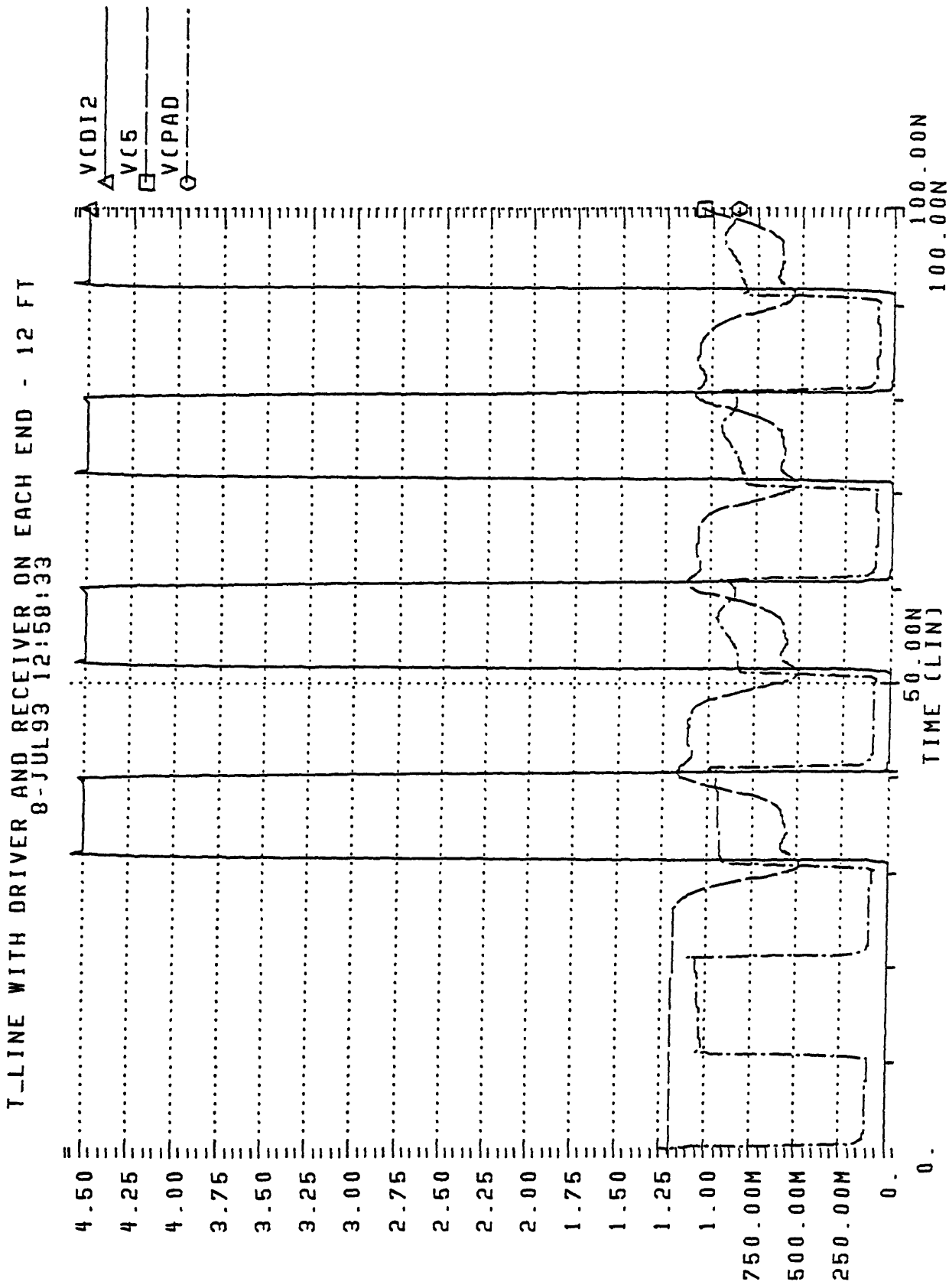

Figure A-8: Plot of Results with PCB Trace Length of 12 ft

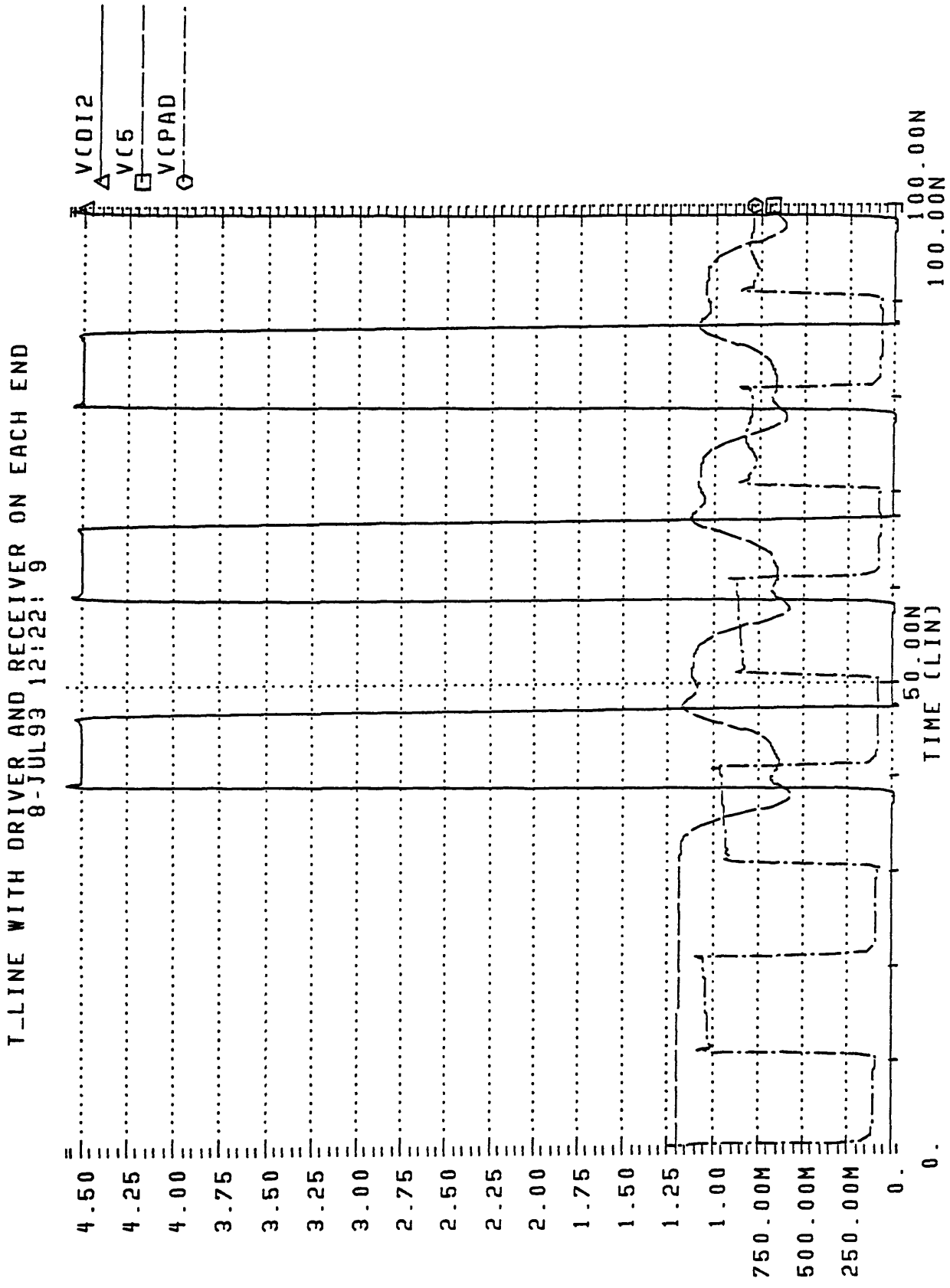

Figure A-9: Plot of Results with PCB Trace Length of 15 ft

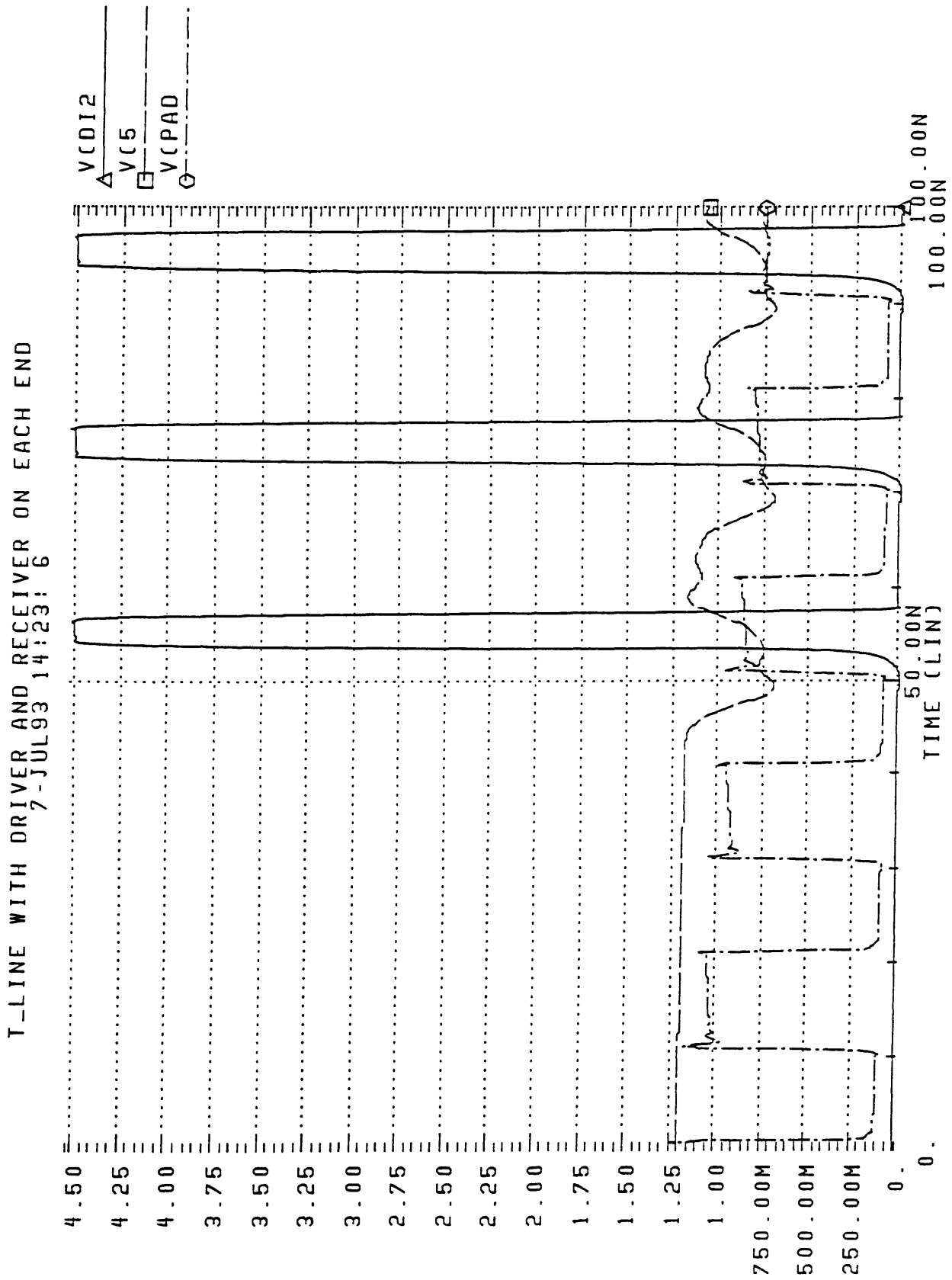

Figure A-10: Plot of Results with PCB Trace Length of 20 ft

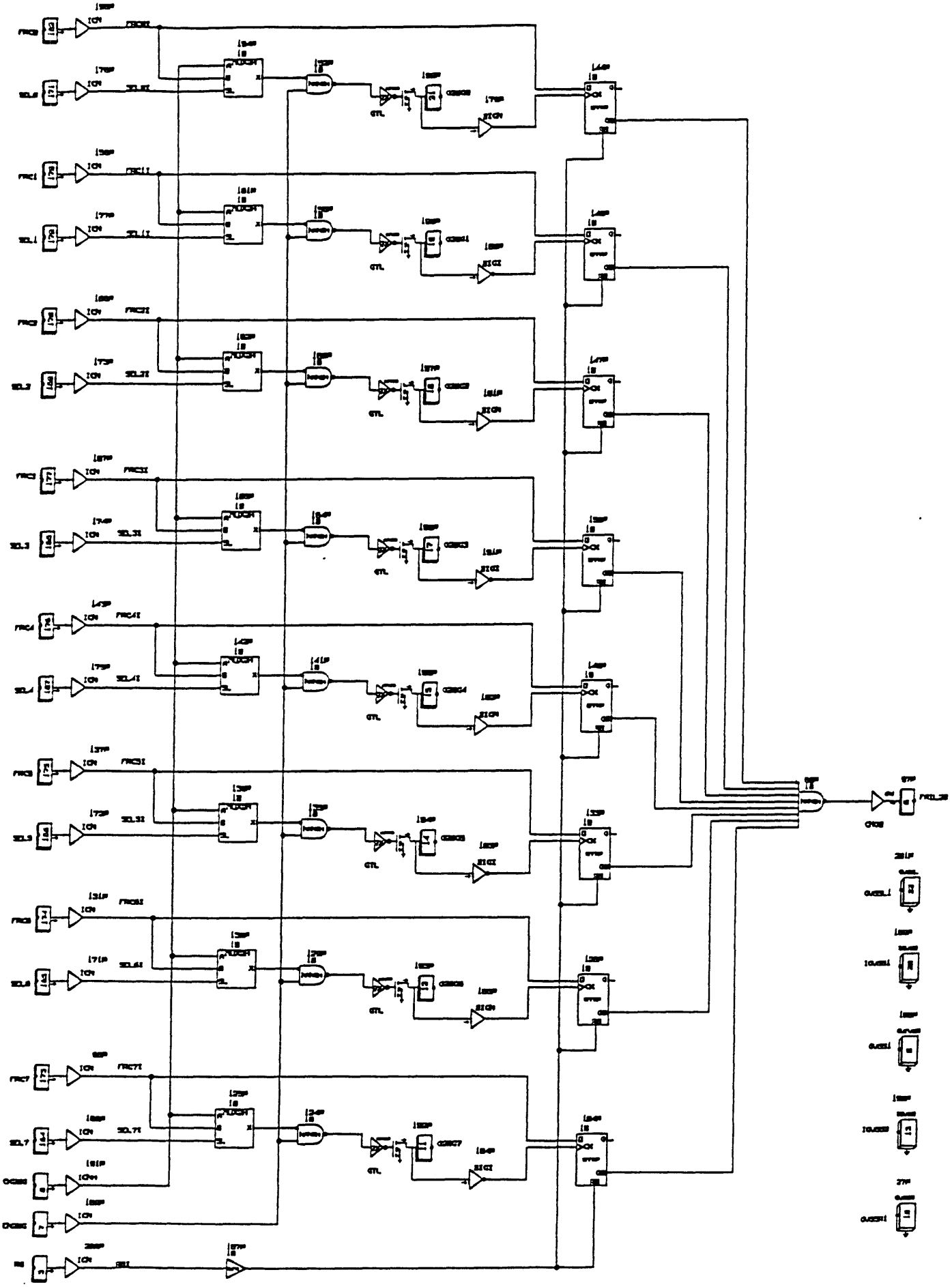

Figure A-11: Schematic of Portion of Test Chip Used in Transmitter

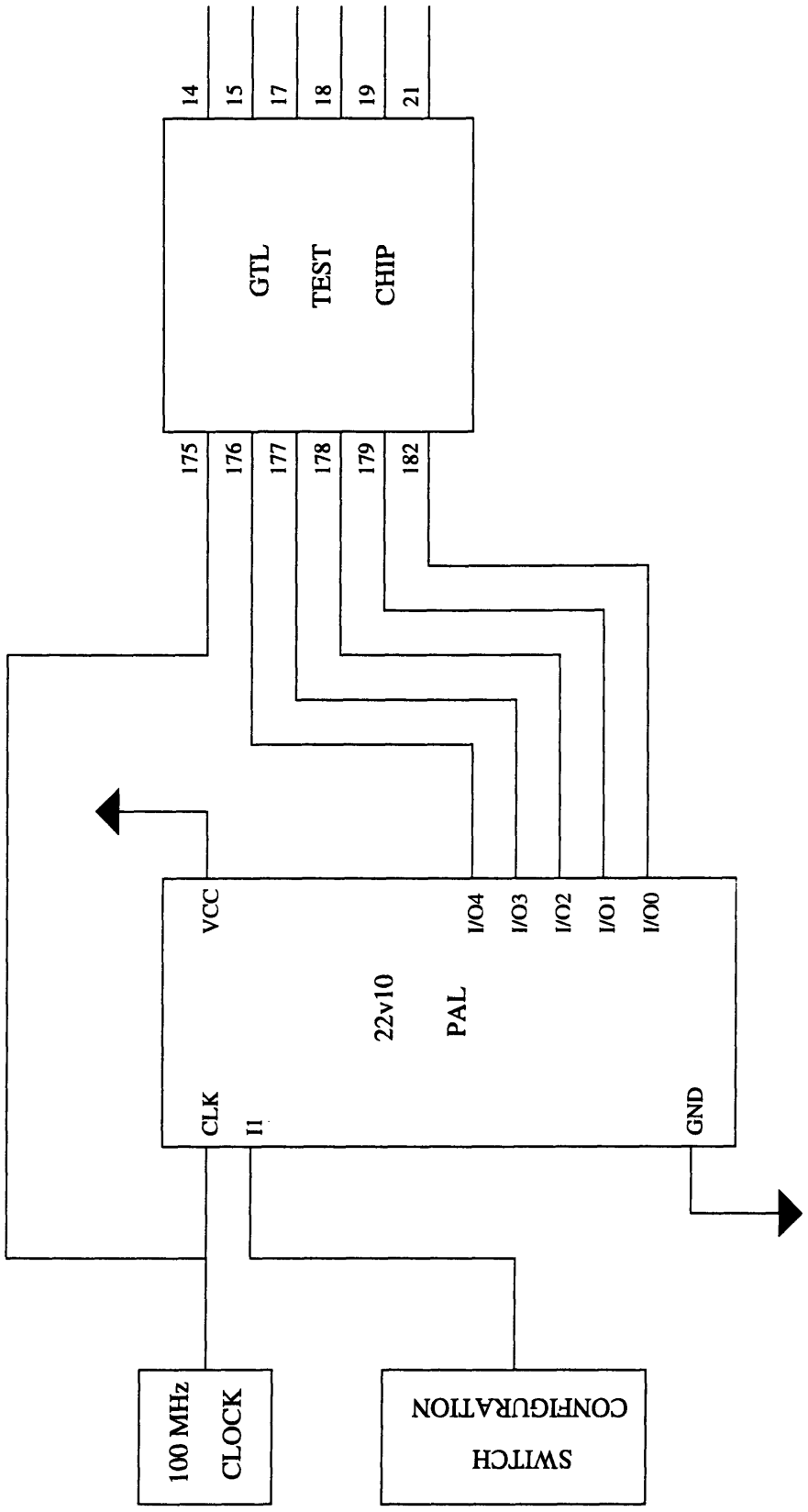

Figure A-12: Schematic of Transmitter Portion of Test Board

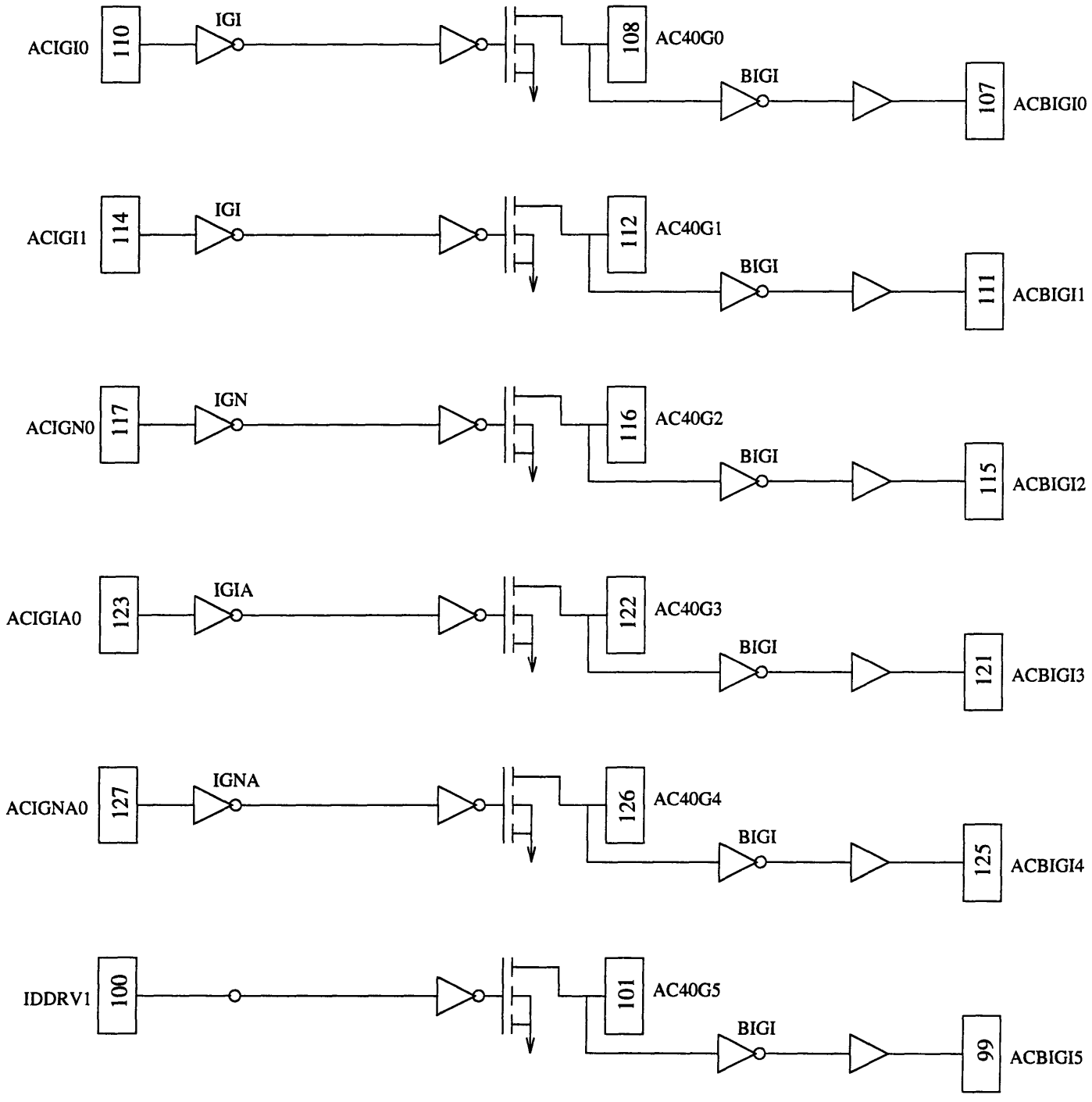

Figure A-13: Diagram of Portion of Test Chip Used in Receiver

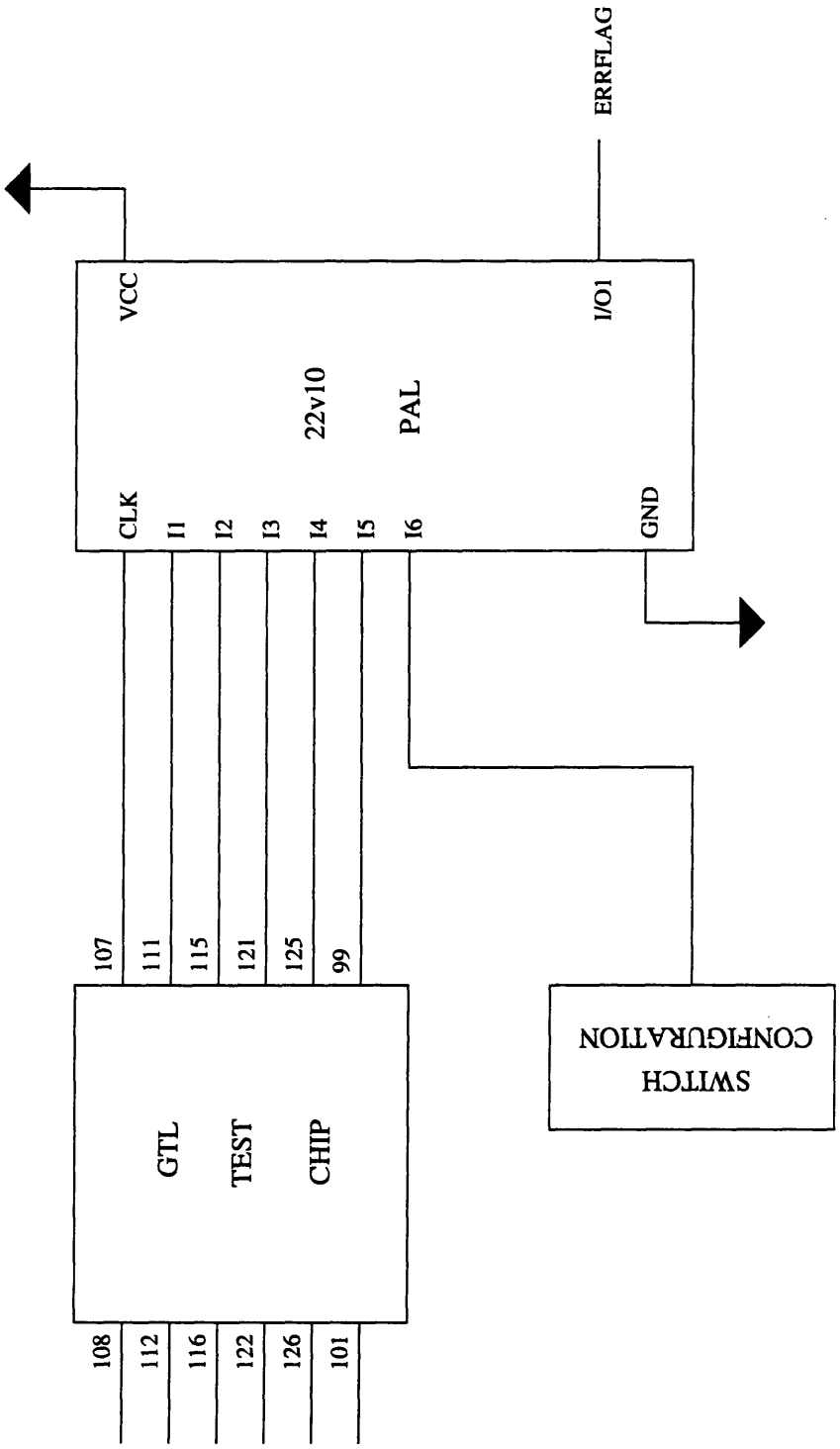

Figure A-14: Schematic of Receiver Portion of Test Board

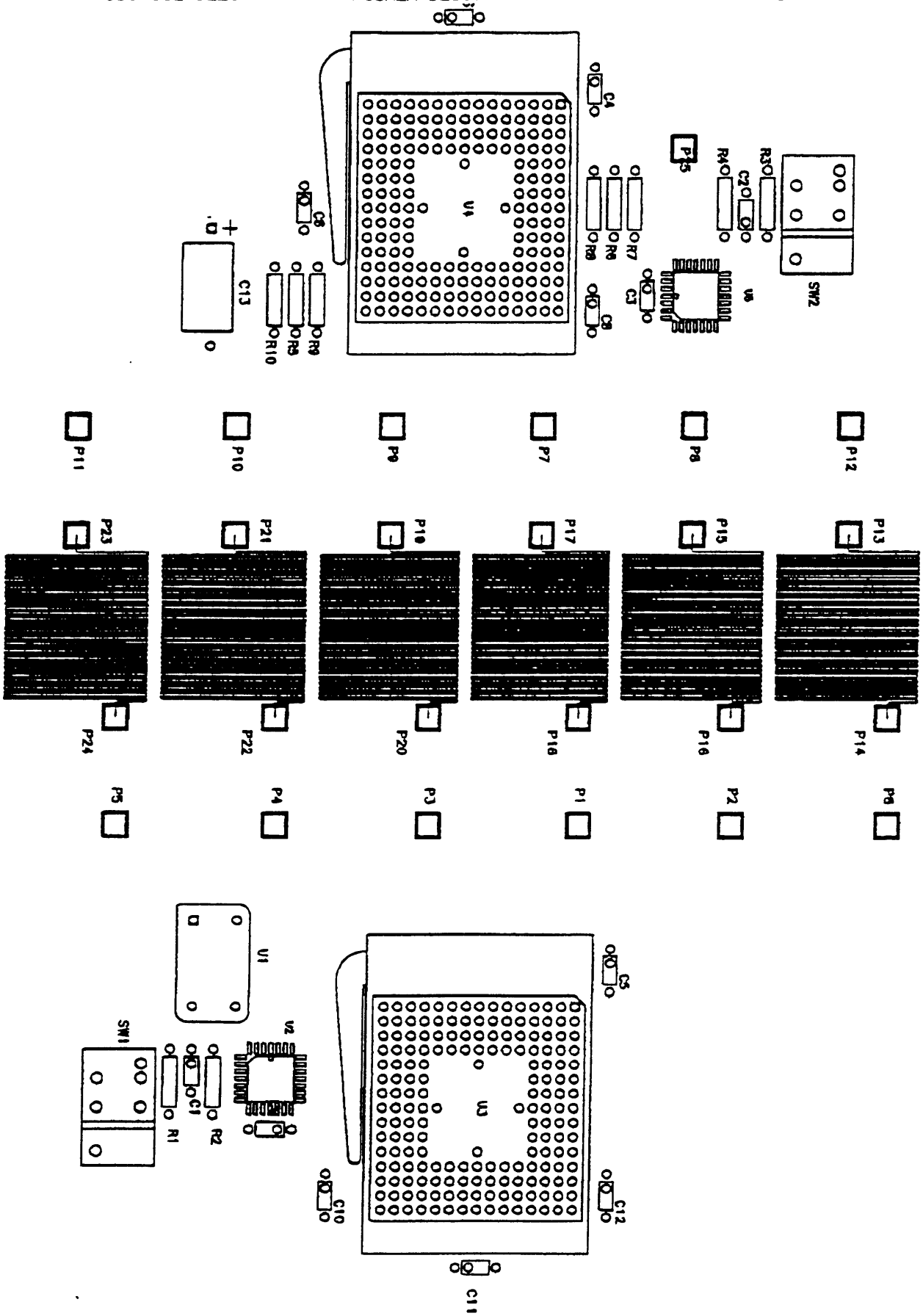

Figure A-15: Checkplot Received from Mosher Design Services

Appendix B

PAL Code

```
*** This is the code used to program the PAL at the transmitter

MODULE _sr5numgen
FLAG '-r3'
TITLE '5-bit psuedo random number generator'
sr5numgen_dev DEVICE 'P22v10';

" This module implements a 5-bit psuedo random number generator
" It uses a 5-bit shift register with an xor feedback
" The feedback bit gets shifted in at every cycle
" It clears to all 1's since clearing to all 0's leaves us deadlocked
" in that state

"control inputs
CLK PIN 1;
CLR PIN 2;

"parallel data outputs
Q0, Q1, Q2, Q3, Q4 PIN 14, 15, 16, 17, 18;

"feedback bit
FDBK PIN 19;

"constants and definitions
H, L, C = 1, 0, .C.;

Q = [Q4, Q3, Q2, Q1, Q0];
Qi_1 = [Q3, Q2, Q1, Q0, FDBK];
```

```
EQUATIONS

Q := !CLR & Qi_1
   #  CLR & [1,1,1,1,1];

FDBK = Q4 $ Q1;

TEST_VECTORS
([CLK,CLR] -> [    Q    ])
 [ C , H ] -> [^h1f];
 [ C , L ] -> [^h1e];
 [ C , L ] -> [^h1c];
 [ C , L ] -> [^h19];
 [ C , L ] -> [^h13];
 [ C , L ] -> [^h06];
 [ C , L ] -> [^h0d];
 [ C , L ] -> [^h1a];
 [ C , L ] -> [^h14];
 [ C , L ] -> [^h09];
 [ C , L ] -> [^h12];
 [ C , L ] -> [^h04];
 [ C , L ] -> [^h08];
 [ C , L ] -> [^h10];
 [ C , L ] -> [^h01];
 [ C , L ] -> [^h02];
 [ C , L ] -> [^h05];
 [ C , L ] -> [^h0a];
 [ C , L ] -> [^h15];
 [ C , L ] -> [^h0b];
 [ C , L ] -> [^h17];
```

```
 [ C , L ] -> [^h0e];
 [ C , L ] -> [^h1d];
 [ C , L ] -> [^h1b];
 [ C , L ] -> [^h16];
 [ C , L ] -> [^h0c];
 [ C , L ] -> [^h18];
 [ C , L ] -> [^h11];
 [ C , L ] -> [^h03];
 [ C , L ] -> [^h07];
 [ C , L ] -> [^h0f];
 [ C , L ] -> [^h1f];
 [ C , H ] -> [^h1f];

END _sr5numgen
```

```
*** This is the code which was used to program the PAL at the receiver

MODULE _errdetect
FLAG '-r3'
TITLE 'error detector'
errdetect_dev DEVICE 'P22v10';

" This module checks for transmission errors.
" By performing the same xor feedback as in the generator,
" it predicts the next incoming set of bits.
" It compares these bits to the actual incoming bits.
" When a difference is detected, it raises the error flag.

"control inputs
CLK PIN 1;
CLR     PIN 7;

"parallel data inputs
D0, D1, D2, D3, D4 PIN 2, 3, 4, 5, 6;

"feedback bit
FDBK  PIN 19;

"predicted next input
Q0, Q1, Q2, Q3, Q4 PIN 14, 15, 16, 17, 18;

"error flag
ERR PIN 20;
```

```
"constants and definitions
H, L, C, X = 1, 0, .C., .X.;

D = [D4, D3, D2, D1, D0];
Q = [Q4, Q3, Q2, Q1, Q0];
Qi_1 = [D3, D2, D1, D0, FDBK];

EQUATIONS
Q := Qi_1;

FDBK = D4 $ D1;

ERR := !CLR & ((Q4 $ D4) # (Q3 $ D3) # (Q2 $ D2) # (Q1 $ D1) # (Q0 $ D0) # ERR);

TEST_VECTORS
([CLK,CLR,  D  ] -> [ERR])
 [ C , H , 'h1f] -> [ L ];
 [ C , L , 'h1e] -> [ L ];
 [ C , L , 'h1c] -> [ L ];
 [ C , L , 'h19] -> [ L ];
 [ C , L , 'h13] -> [ L ];
 [ C , L , 'h07] -> [ H ];
 [ C , L , 'h0d] -> [ H ];
 [ C , L , 'h1a] -> [ H ];
 [ C , H , 'h18] -> [ L ];

END _errdetect
```

Bibliography

[1] B. S. Ang, Arvind, and D. Chiou. Start the next generation: Integrating global caches and dataflow architecture. Computation Structures Group Memo 354, Massachusetts Institute of Technology, MIT Laboratory for Computer Science, February 1994.

[2] H. B. Bakoglu. *Circuits, Interconnections and Packaging for VLSI.* Addison-Wesley Publishing Company, Inc., 1990.

[3] Jr. C. F. Coombs. *Printed Circuits Handbook.* McGraw-Hill Book Company, third edition, 1988.

[4] H. A. Haus and J. R. Melcher. *Electromagnetic Fields and Energy.* Prentice Hall, Inc., Englewood Cliffs, NJ, 1989.

[5] R. Massey, C. Makata, and JoEllen Brock. *H4C Series Design Reference Guide.* Motorola, Inc., 1992.

[6] R. S. Nikhil, G. M. Papadopoulos, and Arvind. *t: A multithreaded massively parallel architecture. Computation Structures Group Memo 325-2, Massachusetts Institute of Technology, MIT Laboratory for Computer Science, March 1992.

[7] G. M. Papadopoulos, G. A. Boughton, R. Greiner, and M. J. Beckerle. *t: Integrated building blocks for parallel computing. Technical report.