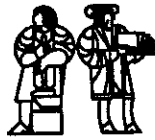


**LABORATORY FOR  
COMPUTER SCIENCE**



**MASSACHUSETTS  
INSTITUTE OF  
TECHNOLOGY**

**Design and Implementation of a High-speed  
Data Link for a Dataflow Supercomputer**

Computation Structures Group Memo 298  
June 1989

**John Santoro**

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Bachelor of Science. This report describes research done at the Laboratory of Computer Science of the Massachusetts Institute of Technology. Funding for the Laboratory is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-84-K-0099.

545 TECHNOLOGY SQUARE, CAMBRIDGE, MASSACHUSETTS 02139

# **DESIGN AND IMPLEMENTATION OF A HIGH-SPEED DATA LINK FOR A DATAFLOW SUPERCOMPUTER**

by

John Joseph Santoro

Submitted to the  
Department of Electrical Engineering and Computer Science

June 5, 1989

In Partial Fulfillment of the Requirements for the Degree of  
Bachelor of Science in Electrical Engineering and Science.

## **ABSTRACT**

The Monsoon Dataflow Supercomputer is a high speed numerical computation device under development at the Massachusetts Institute of Technology's Laboratory for Computer Science. When completed, the supercomputer will consist of 256 processors and memory devices connected by a message-handling network. This architecture by nature requires the high speed transfer of data and messages. A Packet-switched Routing Chip (PaRC) was developed to provide the communication network. To insure data integrity between PaRC's a separate transmitter and receiver were required. The object of this thesis is to design and implement a high speed digital data link to transfer information reliably throughout the network.

Thesis Supervisor:  
Title:

Dr. George A. Boughton  
Research Associate, Laboratory for Computer Science

## Acknowledgments

Special thanks to Andy Boughton for the support and friendship without which this thesis would not be possible. Thanks to Christopher Joerg, Jack Costanza, and Ralph Tiberio for their generous assistance.

## **Dedication**

To my family, for their patience, love, and support through all of my endeavors.

## Table of Contents

<b>ABSTRACT</b>	<b>2</b>
<b>Acknowledgments</b>	<b>3</b>
<b>Dedication</b>	<b>4</b>
<b>Table of Contents</b>	<b>5</b>
<b>List of Figures</b>	<b>6</b>
<b>1. Project Overview</b>	<b>7</b>
1.1 Introduction	7
1.2 Design Objectives	9
1.3 Thesis Goals	9
<b>2. Design Overview</b>	<b>11</b>
2.1 Design Objectives	11
2.1.1 Transmitter Objectives	11
2.1.2 Receiver Objectives	11
2.2 Design Strategies	11
2.2.1 Data Integrity and Encoding	11
2.2.2 Synchronization	12
2.2.3 Timing of Clock Generation	13
<b>3. Implementation</b>	<b>14</b>
3.1 CAD Tools	14
3.2 Transmitter	14
3.2.1 Data Multiplexing (MUX)	15
3.2.2 Synchronization (SYNC_CIRCUIT)	19
3.2.3 50 MHz Clock Generation (CLKTREE50)	23
3.2.4 200 MHz Clock Generation (TIMING_SETUP)	23
3.2.5 Acknowledge Signals	26
3.2.6 Drivers	27
3.3 Receiver	27
3.3.1 Data Reception	27
3.3.2 Data Demultiplexing (DEMUX)	31
3.3.3 Clock Generation	31
3.3.4 Acknowledge Signals	33
3.4 Cable	33
3.4.1 Objectives	33
3.4.2 Prototype	34
3.4.3 Testing	34
3.4.4 Conclusions	35
<b>4. Verification</b>	<b>38</b>
<b>5. Conclusions</b>	<b>39</b>

## List of Figures

<b>Figure 1-1:</b>	<b>Sixteen Processor Communication Architecture</b>	<b>8</b>
<b>Figure 1-2:</b>	<b>Communication Architecture with Data Link</b>	<b>10</b>
<b>Figure 3-1:</b>	<b>Transmitter</b>	<b>15</b>
<b>Figure 3-2:</b>	<b>Transmitter Block Diagram</b>	<b>16</b>
<b>Figure 3-3:</b>	<b>Transmitter Schematic</b>	<b>17</b>
<b>Figure 3-4:</b>	<b>MUX Schematic</b>	<b>18</b>
<b>Figure 3-5:</b>	<b>Synchronization Schematic, Part 1</b>	<b>20</b>
<b>Figure 3-6:</b>	<b>Synchronization Schematic, Part 2</b>	<b>22</b>
<b>Figure 3-7:</b>	<b>CLKTREE50 Schematic</b>	<b>24</b>
<b>Figure 3-8:</b>	<b>TIMING_SETUP Schematic</b>	<b>25</b>
<b>Figure 3-9:</b>	<b>Receiver</b>	<b>28</b>
<b>Figure 3-10:</b>	<b>Receiver Block Diagram</b>	<b>29</b>
<b>Figure 3-11:</b>	<b>Receiver Schematic</b>	<b>30</b>
<b>Figure 3-12:</b>	<b>DEMUX Schematic</b>	<b>32</b>
<b>Figure 3-13:</b>	<b>Waveform of Test Pattern</b>	<b>35</b>
<b>Figure 3-14:</b>	<b>Waveform Received over Forty Feet</b>	<b>36</b>
<b>Figure 3-15:</b>	<b>Waveform Received over Twenty Feet</b>	<b>37</b>

## Chapter 1

### Project Overview

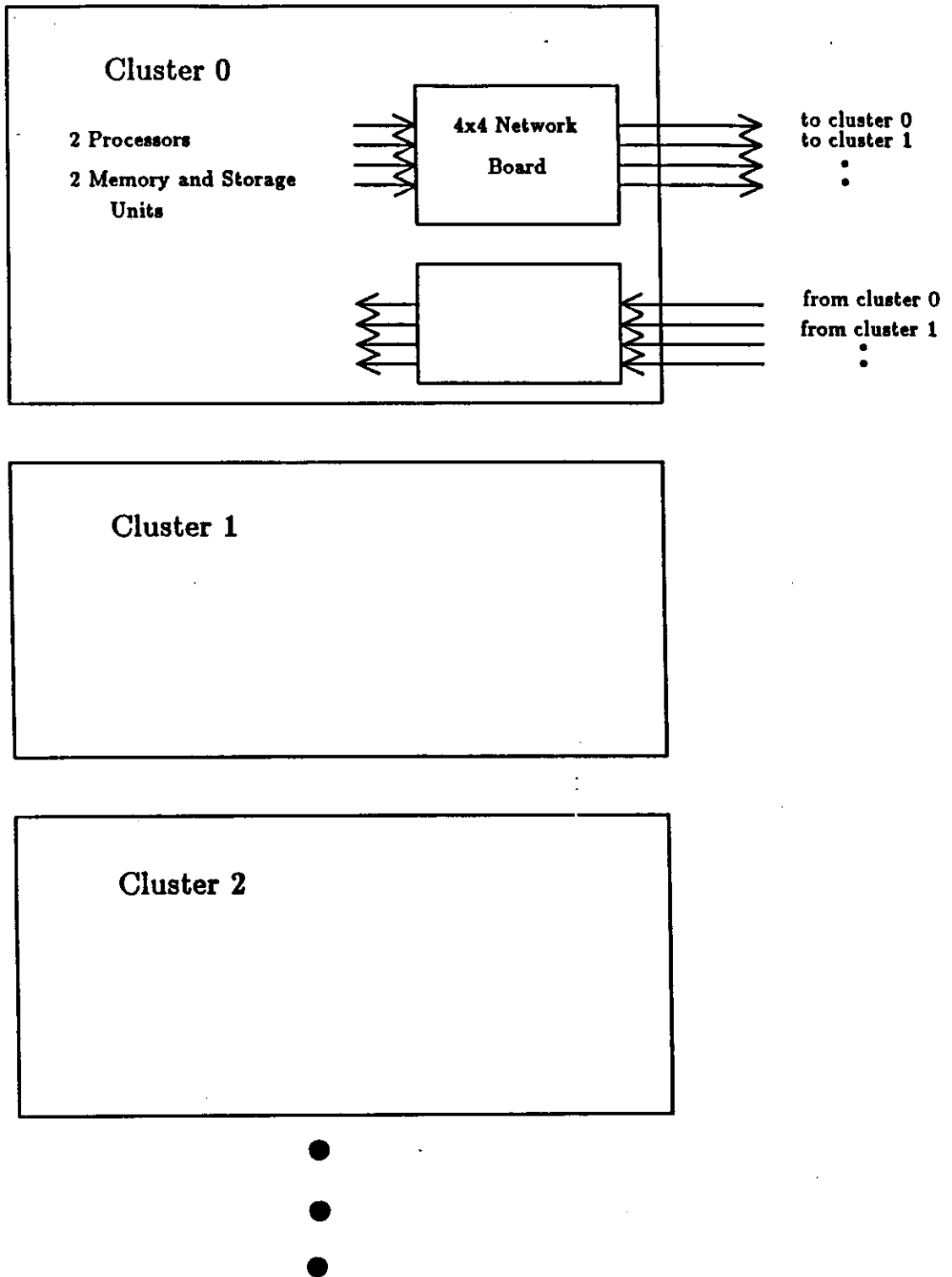
#### 1.1 Introduction

The Monsoon parallel computation device is a dataflow processor currently being designed at the Laboratory for Computer Science under Professor Arvind. In a dataflow architecture data are passed around the system as message packets. Operations thus can be distributed among the processors to be executed upon arrival of the necessary data packets, rather than waiting for unrelated previous instructions. Ultimately the device will consist of 128 processor units and 128 memory and storage units. Each processor is capable of operating at 10 MIPs. In the final form, the computer has been designed to execute instructions in the range of 500 to 1000 MIPs.

To implement this message-passing system, a routing network is needed to enable all 256 elements to communicate. A Packet Routing Chip (PaRC), developed by Christopher Joerg as an undergraduate thesis, transmits sixteen bit long data words at a speed of 800 Megabits per second through four input and four output ports. Network boards composed of a two by four array of these chips will provide sixteen inputs and sixteen outputs. An arrangement of these boards will interconnect the processors. In the sixteen processor Monsoon prototype, each network board will contain a single PaRC. Each board will service either the inbound or outbound traffic for a cluster of two processors and two memory and storage units. The architecture for this arrangement is shown in Figure 1-1.

To address concerns for data integrity and electrical difficulties with PaRC's CMOS technology, a separate data link chip was designed to transmit and receive data between PaRCs between individual network boards and between network and processor boards. In addition to providing better drivers, the data link chip will be able to reduce the number of

Figure 1-1: Sixteen Processor Communication Architecture





signals through multiplexing and thus permit the use of higher quality connectors and cables. The architecture with the data link chip is in Figure 1-2. Board 1 represents an outbound board for a cluster of processors. The sixteen bits are passed at 50 MHz from PaRC to the data link, where they are multiplexed to four serial nibbles at 200 MHz and driven down the interboard cable. The data are received on the inbound Board 2, where they are demultiplexed to sixteen bits for the receiving PaRC.

## 1.2 Design Objectives

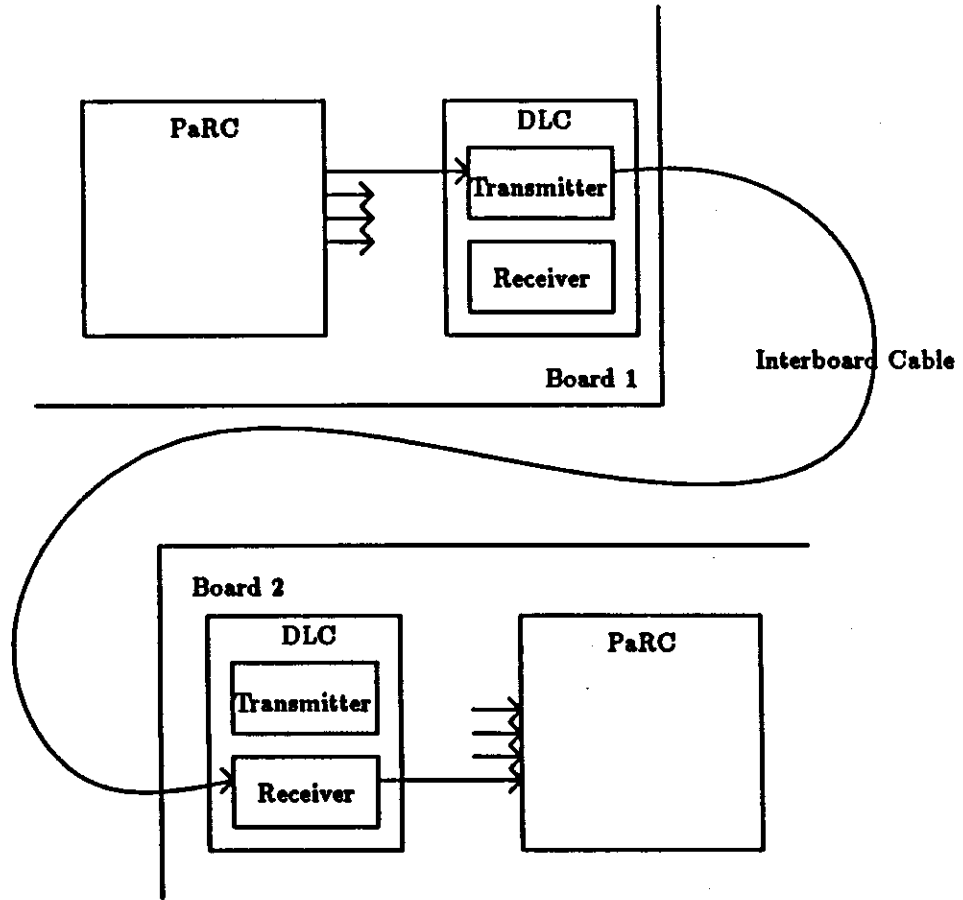
- Transmit sixteen bit data words as four nibbles at a rate of 800 Megabits per second
- Provide extremely low data error rate over forty feet of cable
- Generate processors' clocks, PaRC's local clock, and PaRC's data reception clock
- Pass acknowledge signals from PaRC receiver to PaRC transmitter
- Use ECL technology to transmit signals differentially

The preliminary design of this data link was begun by Robert Lester as an undergraduate thesis. Lester's thesis provided a block diagram of the system and an encoding scheme to provide data error-checking.

## 1.3 Thesis Goals

The object of this thesis is to design the chip and complete its implementation for fabrication. The second chapter provides a design overview and discusses strategies and decisions involved. The next chapter is the implementation, examining each component of the transmitter and receiver as well as the cable technology to be employed. The fourth chapter describes design and fabrication verification testing. The final section concludes on the success of the design and on the future of the project.

Figure 1-2: Communication Architecture with Data Link



## Chapter 2

### Design Overview

#### 2.1 Design Objectives

##### 2.1.1 Transmitter Objectives

- Multiplex 16 bit data word from PaRC at 50 MHz into 4 nibbles, to be sent differentially at 200 MHz
- Synchronize PaRC data with the local clock and report synchronization errors
- Generate a 50 MHz clock for PaRC and a 200 MHz clock for the receiver to load transmitted data
- Generate a load signal for receiver output registers
- Receive two differential acknowledge signals from the receiver

##### 2.1.2 Receiver Objectives

- Receive differentially four nibbles in series at 200 MHz and demultiplex them into one sixteen bit word at 50 MHz
- Generate a 50 MHz clock for PaRC to synchronize data reception from the data link
- Pass two acknowledge signals from receiving PaRC to data link transmitter

#### 2.2 Design Strategies

##### 2.2.1 Data Integrity and Encoding

The elimination of encoding was an early decision that was a significant deviation from Lester's original design. By translating each four bit data block into a unique combination of three ones and three zeros, the signal would be more stable electrically and would be more likely to reveal an error if bits were altered. However, the encoding scheme required that the six bits be sent serially at 300 MHz in order to maintain the 800 Mbits

flow rate for the incoming sixteen bit data word. The multiple layers of logic that the encoding required created delays that could exceed the fast clock period. An alternative had to be found to ensure data integrity.

After evaluating cable technologies with ECL drivers and receivers in a test setup, it was concluded that high quality cables do exist to meet the link's data integrity requirements (Cable, Section 3.4). If the signal is given sufficient time to settle (approximately 1.5-2 ns), the data would have the desired integrity and thus encoding could be eliminated. This had the additional benefit of requiring only a 200 MHz clock, which provided a longer period for data transitions across the link, and simplified the circuit design.

### 2.2.2 Synchronization

The variability of PaRC made synchronization a vital design issue. Process variations between PaRCs require that the data link initially tunes itself to PaRC. As a result of these variations, control signals and clocks from PaRC must be synchronized with the data link's clocks. The data from PaRC is timed by PaRC to be in the middle of the PaRC clock cycle.

One difficulty whose solution had to be considered was the possibility of metastability. A signal transition by PaRC could coincide with a clock edge in the synchronization circuitry, throwing the output into an indeterminate state. The synchronization circuitry must be able to guarantee a valid voltage level for the signal by giving it 35 ns to settle (Synchronization, Section 3.2.2).

On the basis of the PaRC's 50 MHz clock, the control circuitry must determine when the transmitter should load its internal data registers and when it should shift out the data for its serial transmission. Conversely, a signal must be generated for the receiver to determine when to load the registers and when to present the data to the receiving PaRC.

The decision was made to use a signal rather than a separate clock because the logic used by a signal would be more consistent than the variability in the oscillations of a clock (Section 3.2.2).

The CMOS technology used by PaRC is susceptible to temperature variation with use. The fluctuation can be as much as 3 ns., significant to a period of 5 ns. for the 200 MHz clock. Error detection circuitry is required to inform PaRC when the clock accompanying its data is out of synchronization with the data link's local clocks. While the design must withstand a variation of up to 3 ns, two degrees of fault will be indicated for variations larger than 3 ns.

### **2.2.3 Timing of Clock Generation**

In order to provide approximately 2 ns for the data traveling across the cable to settle, the timing of the local clock that shifts out the data across the link and the clock that loads the data in a flip-flop in the receiver is of critical concern. If the signal is given significantly less time to settle electrically, the data integrity will be endangered. In addition, the design must account for skew, caused by variations in the cable, between the transmitted clock and data signals.

## Chapter 3

### Implementation

#### 3.1 CAD Tools

The implementation of the data link chip was done with the IDEA development system by Mentor Graphics on an Apollo workstation. The system has features for schematic capture and editing, functionality testing, and timing analysis. The tool allows the designer to create a high level block diagram form and proceed through successive layers of detail to the primitive "cells." These "cells" are logic components that comprise the circuit and are particular to the library for the given gate array. The system then generates net routing lists that permits the fabrication of the chip. The actual chip is a Motorola MCA2500 semi-customized ECL gate array.

#### 3.2 Transmitter

The primary role of the transmitter is to take a sixteen bit data word from PaRC and serially transmit it in four nibbles. In addition the transmitter generates clock signals, synchronizes data, and receives acknowledge signals from the receiving PaRC to pass to the transmitting PaRC. The input and output signals required for these functions can be seen in Figure 3-1.

A block diagram of the transmitter is shown in Figure 3-2. The sixteen bit wide data word at 50 MHz from PaRC is captured into four sets of registers with a nibble each. Each set of registers then shifts the bits at 200 MHz out of the transmitter. The synchronization and clock generation circuitry are separate modules that provide timing and framing information. A schematic of the transmitter implementation is in Figure 3-3. The names in parentheses following the section titles refer to the subsystem names in the schematic.

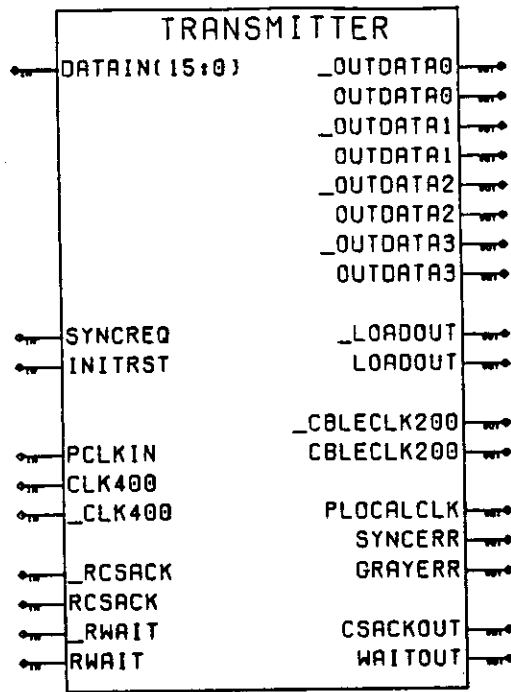


Figure 3-1: Transmitter

### 3.2.1 Data Multiplexing (MUX)

A schematic for the data transmission circuitry is shown in Figure 3-4. A block of four of PaRC's sixteen data bits enters the circuit (DATAIN in Figure 3-4) and is latched into a M291 flip-flop at 50 MHz (see Section 3.2.3 for details on the 50 MHz clock generation). The data from these flip-flops feed the next set of flip-flops according to the SHIFT signal. The signal is the inverse of REGLOAD generated by the synchronization circuitry (Section 3.2.2 below). SHIFT provides for one 200 MHz cycle low in which the data are latched followed by three 200 MHz cycles high in which the data in the flip-flops are shifted out as SHIFTOUT. SHIFTOUT is output with its complement so that it can be transmitted differentially (see Drivers, Section 3.2.6).

Figure 3-2: Transmitter Block Diagram

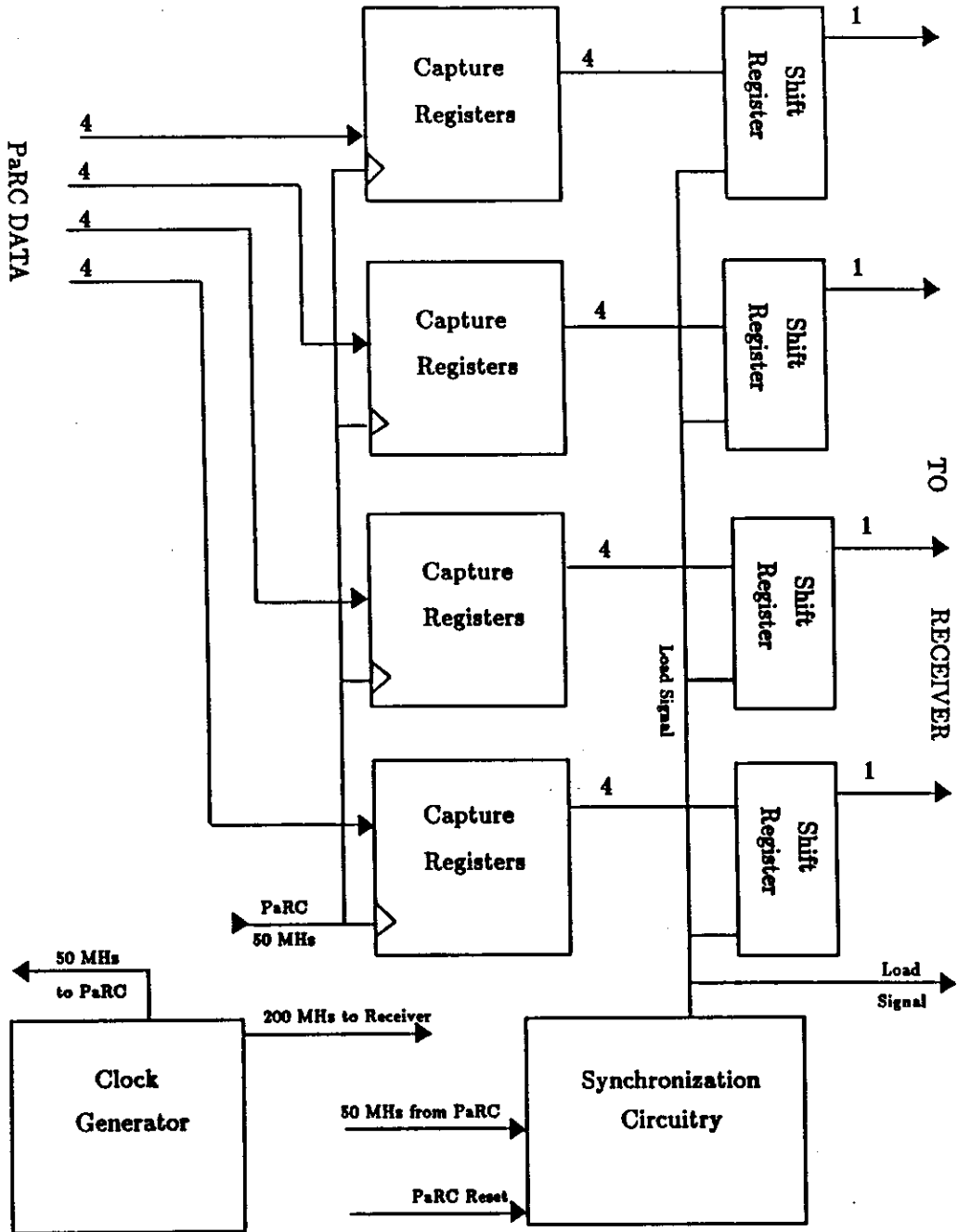




Figure 3-3: Transmitter Schematic

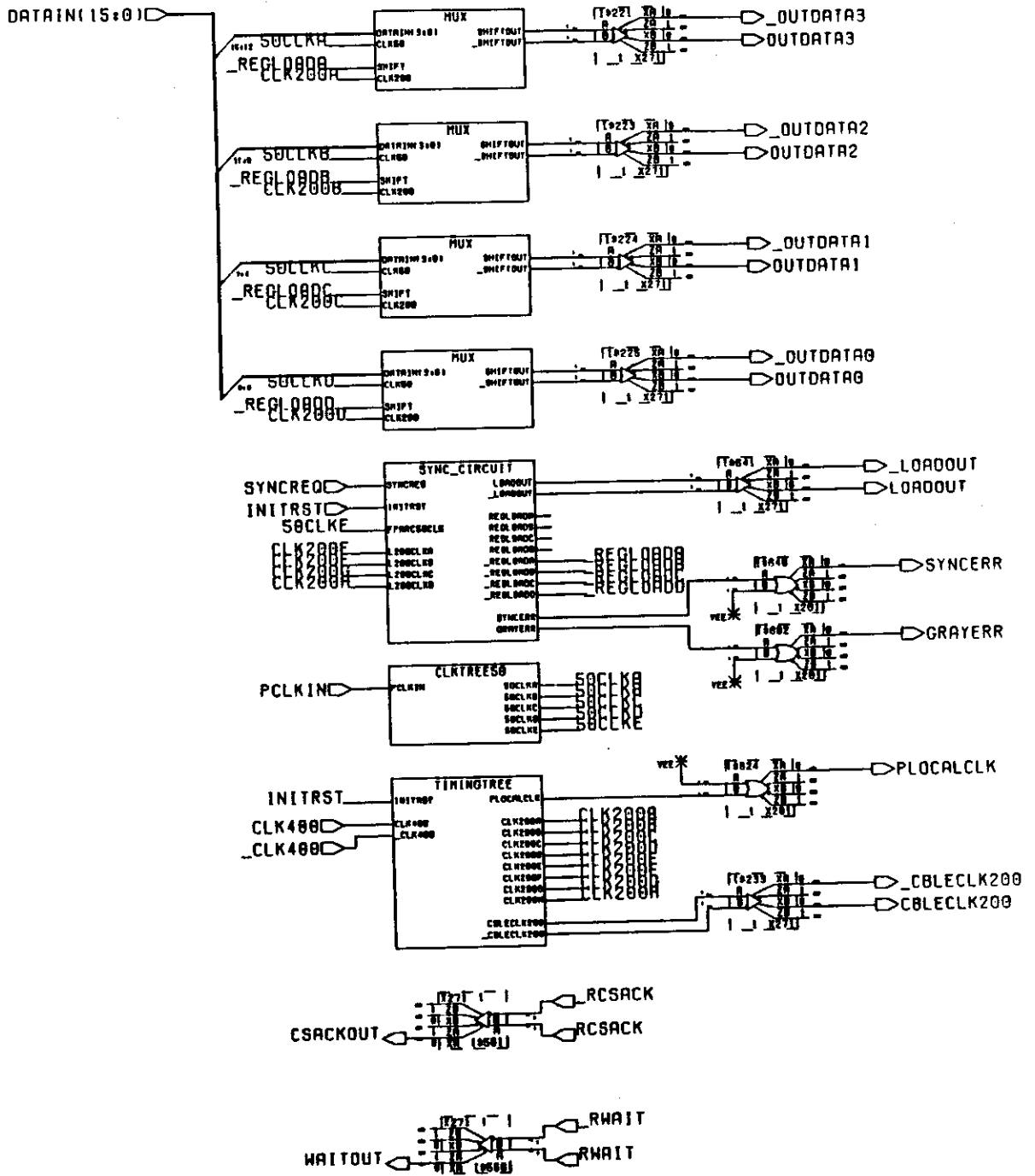
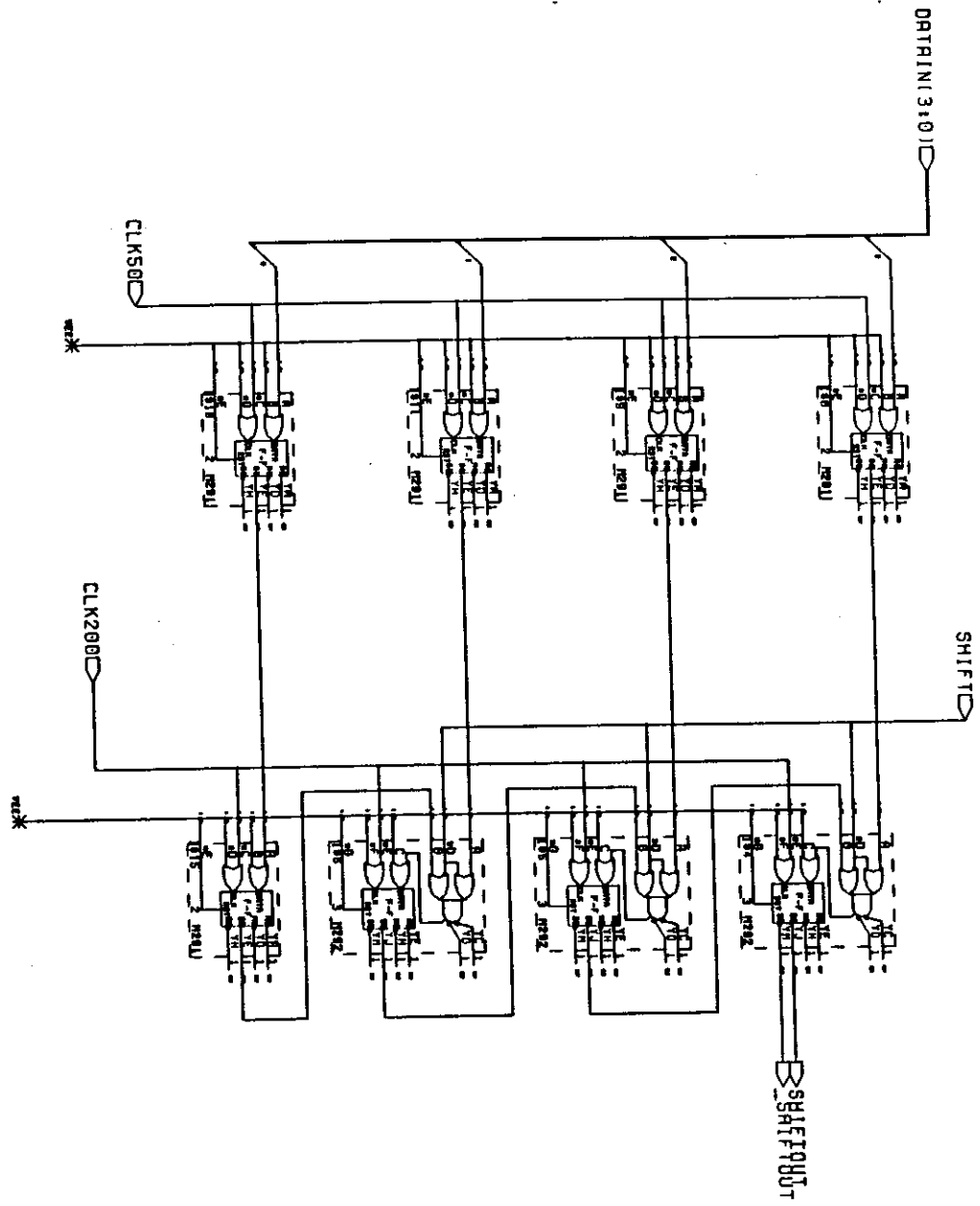


Figure 3-4: MUX Schematic



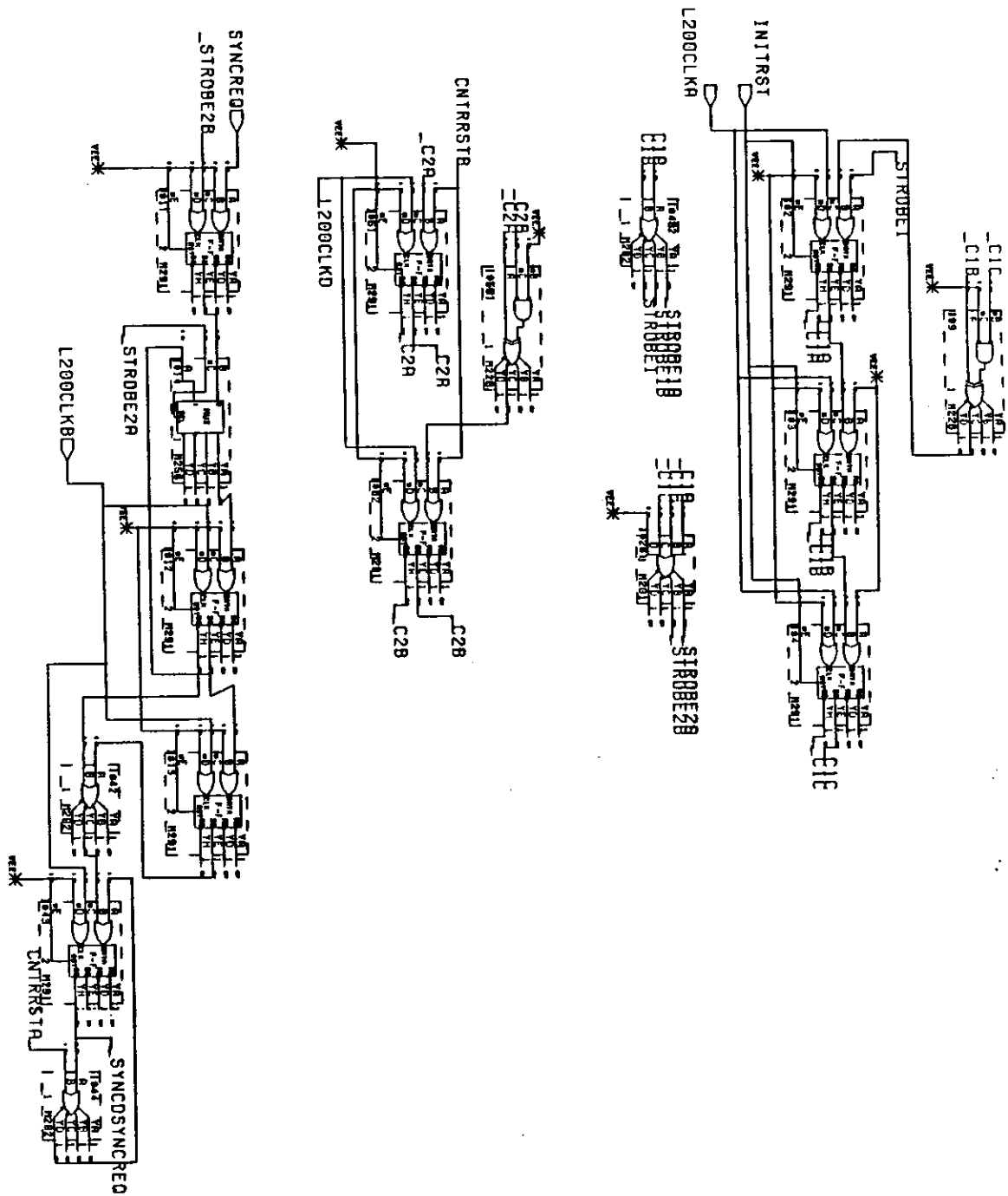
### 3.2.2 Synchronization (SYNC\_CIRCUIT)

Schematics of the synchronization circuitry are in Figure 3-5 and Figure 3-6. The purpose of the synchronization circuitry is to generate the REGLOAD signal for the transmitter's capture circuitry based on the 50 MHz clock on the PaRC that is sending the data. This 50 MHz clock fluctuates because of temperature and process variations, thus it must be considered to be asynchronous relative to the transmitter's local 200 MHz clock. After the transmitter clocks are reset with INTRST, the synchronization circuitry begins by detecting the synchronization request from PaRC (SYNCREQ in Figure 3-5). When this request is detected, the circuit begins to search for the rising edge of the PaRC clock FPARC50CLK. This clock is actually one of five copies of PCLKIN in Figure 3-1 that was distributed by CLKTREE50 (Section 3.2.3). With the location of the edge determined, the synchronization circuitry periodically generates \_REGLOADA through \_REGLOADD for the transmitter's capture circuitry (Section 3.2.1) and LOADOUT for the receiver's data demultiplexing circuitry (Section 3.3.2). The synchronization circuitry at this point begins to monitor FPARC50CLK and signal an error if its phase drifts too far relative to the 200 MHz local clock. Such phase shifts would be caused by temperature effects in PaRC.

The top two circuits in Figure 3-5 are counters that serve the other functions. The top circuit is a three bit counter that generates strobes to search for the SYNCREQ and FPARC50CLK signals. By using logic combinations of three bits, the circuit counts to seven while generating \_STROBE1A, \_STROBE1B, \_STROBE2A, and \_STROBE2B. Each strobe has six 200 MHz cycles high and one 200MHz cycle low. The lower two bit counter is the basis for the four cycle pattern for REGLOAD and LOADOUT.

The strobe allows the SYNCREQ detection circuit to sample and compare the value of SYNCREQ every seven cycles (bottom circuit in Figure 3-5). A cycle of seven is used to counter metastability. If \_STROBE2B clocks the flip-flop exactly when SYNCREQ makes a transition, the output will become metastable. \_STROBE2A either allows the

Figure 3-5: Synchronization Schematic, Part 1



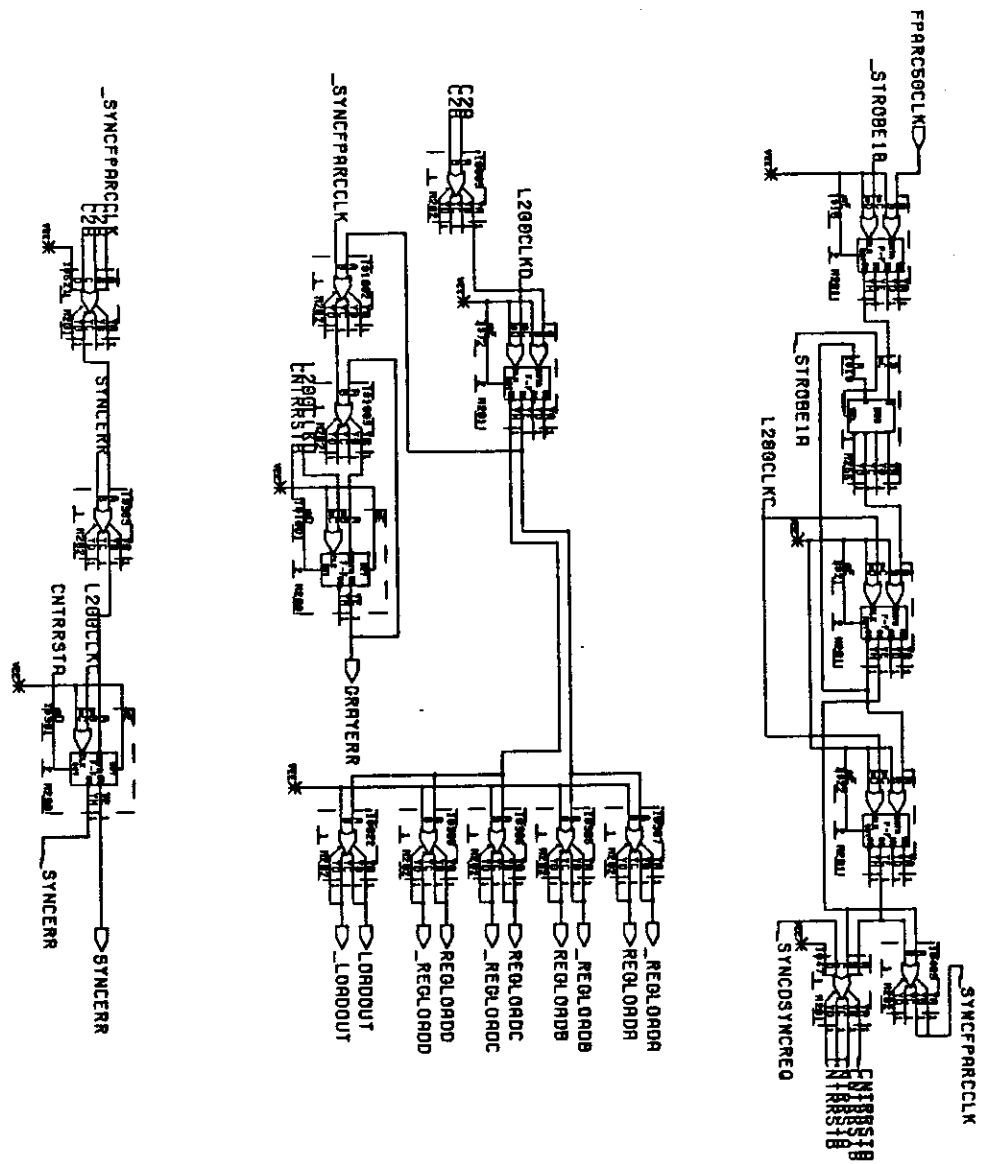
output to pass to the next flip-flop or feeds back the output from the last strobe. In this way, if the output of the first flip-flop were metastable, it would have seven cycles, or 35 ns., to settle to a discrete value. When the circuit logic sees that the result of the previous strobe was low and that from the current strobe is high, SYNCREQ has become active and `_SYNCDSYNCREQ` goes low to initiate the search for FPARC50CLK.

The detection circuitry for FPARC50CLK, located at the top of Figure 3-6, is similar to that for SYNCREQ. `_STROBE1B` and `_STROBE1A` are used to detect the rising edge of FPARC50CLK in the same fashion used to detect SYNCREQ. Since seven 200 MHz cycles are not an even multiple of the 50 ns PaRC clock, the phase of `_STROBE1` changes relative to the edge for which it is searching. The phase difference acts as a window sliding backward through the 50 MHz clock cycle. The seven cycles also allow with high probability any metastability to settle. When the result of the current strobe is low and the previous strobe was high, the circuit has determined the location of the rising edge within 5 ns. At this point the two bit counter is reset; the pattern is 00, 01, 10, 11. The rising edge corresponds with 11 on the counter.

With the location known of the rising edge of PaRC's clock, the shift and load signals, REGLOAD and LOADOUT, can be generated. To keep data transitions caused by these signals the maximum distance from the beginning of a clock cycle, REGLOAD and LOADOUT go high when the count is 01. This causes the shift registers to be loaded approximately two cycles away from the rising edge of FPARC50CLK.

Although the rising edge initially matched with the counter at 11, fluctuation in the PaRC clock may align the edge with a different value. Allowing a buffer of one cycle on each side of the initial value, the fourth value would constitute a serious error. Thus, the value of 01 with the rising edge of FPARC50CLK represents the error condition SYNCERR in the bottom circuit of Figure 3-6. Asymmetries in the allowable phase variation of FPARC50CLK variation to one border cycle preferable to the other. When the

Figure 3-6: Synchronization Schematic, Part 2



bottom circuit detects the rising edge of FPARC50CLK during 10, an intermediate error situation GRAYERR, is indicated to PaRC. When PaRC receives these error signals, it can decide to resynchronize the system by sending another SYNCREQ.

### 3.2.3 50 MHz Clock Generation (CLKTREE50)

The purpose of the 50 MHz clock circuitry is to produce five 50 MHz clocks from one PaRC clock (Figure 3-7). Each suffers an equal delay in this distribution, so that the phase relationships are the same. Four of the signals are used in one of the four multiplexing registers (Section 3.2.1). The remaining clock signal is used as the basis for synchronization with the PaRC clock input (Section 3.2.2).

### 3.2.4 200 MHz Clock Generation (TIMING\_SETUP)

The circuitry in Figure 3-8 is designed to set up the phase relationships between the clocks that transmit and receive the serial data. In addition, it generates a 50 MHz clock for PaRC's local functions (lower right corner of Figure 3-8).

Generation of the local PaRC clock PLOCALCLK is achieved by dividing by four a 200 MHz clock signal (CLK200S from clock tree at the top of Figure 3-8). The two M291 flip-flops and the M228 XOR provide the logic for a two bit counter. The rightmost flip-flop in the figure is the most significant bit, which gives it two cycles at logic one followed by two cycles of logic zero; i.e. a 50 MHz clock with a 50-50 duty cycle.

The generation of the 200 MHz clocks needed for the transmitter and receiver are based on a divided and distributed 400 MHz crystal oscillator. This 400 MHz signal is input to a M374 differential receiver (leftmost portion of Figure 3-8). Following the top branch, the clock is divided by two by feeding back into the flip-flop the complement of an initial logic level provided by INITRST (a M372 differential flip-flop is used because of performance considerations at this speed). This procedure switches the logic level every

Figure 3-7: CLKTREE50 Schematic

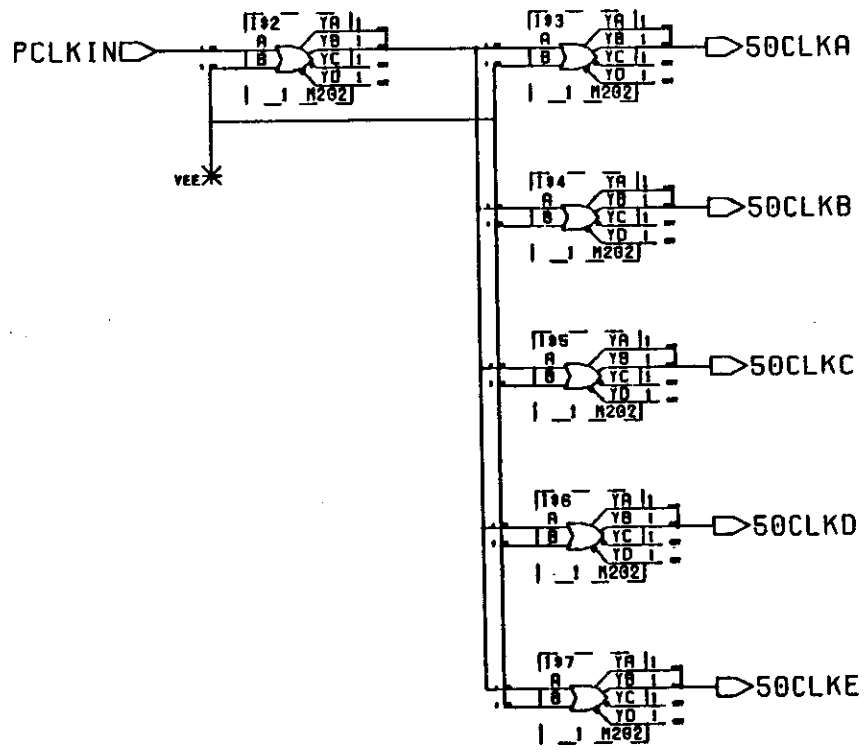
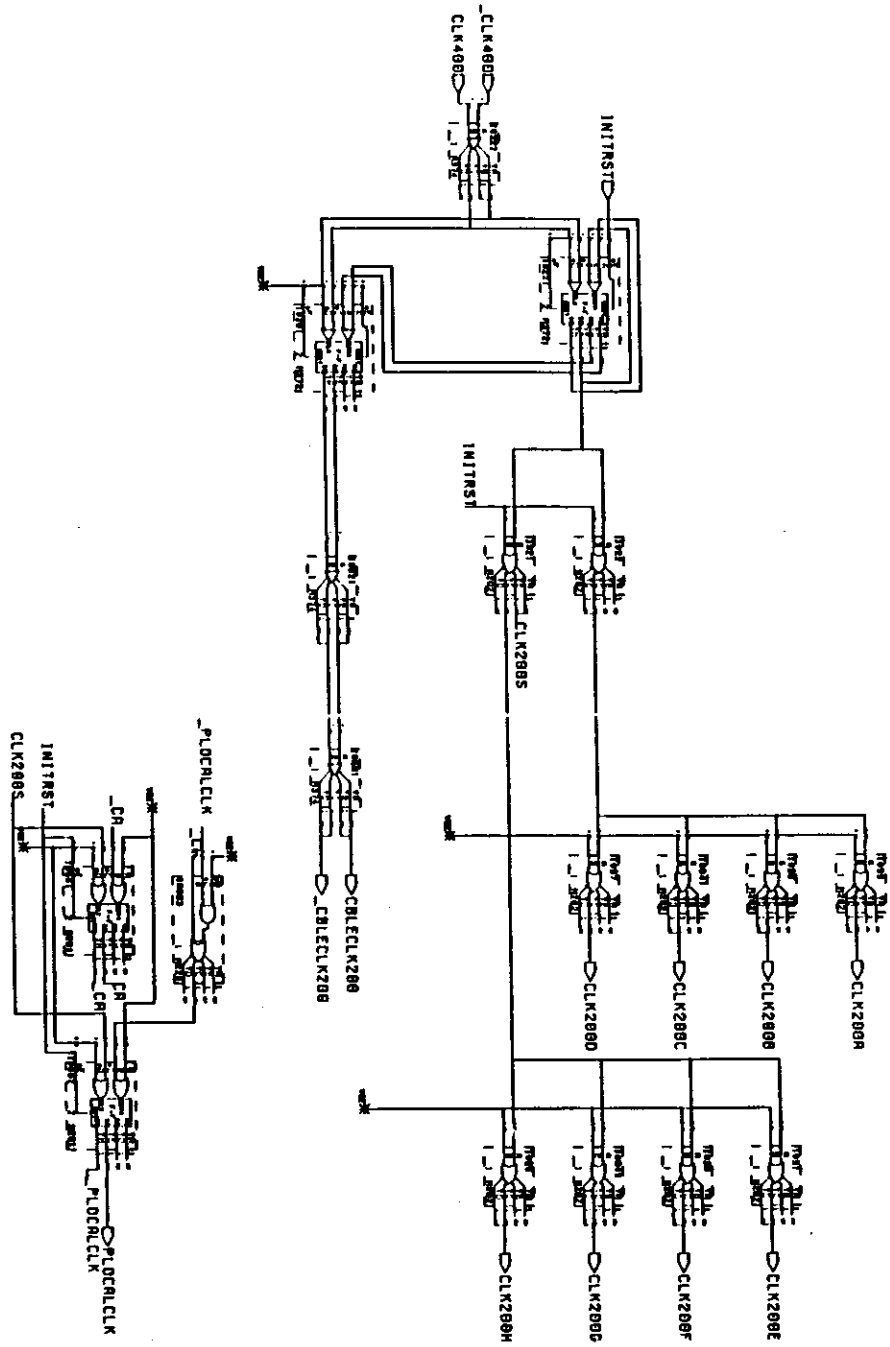




Figure 3-8: TIMING\_SETUP Schematic



two cycles. The resulting 200 MHz signal is distributed into two then six clocks. CLK200A through CLK200D are used in the shift register circuitry (Section 3.2.1). CLK200E and CLK200F are employed in the synchronization circuitry (Section 3.2.2). INTRST is used in the middle two M202 OR gates to reset the clocks to a common state.

The lower branch of the tree creates the phase relationship between the clocks used in the shift circuitry and the clocks in the data reception circuitry. This clock CBLECLK200 travels across the link and is distributed to the four flip-flops that latch the data at the receiver (Section 3.3.1). As indicated in Section 2.2.1, Data Integrity and Encoding, two nanoseconds are desired to allow the data signals to settle electrically. However, the phase may be affected by up to 500 ps of wire skew between the clock and data lines. In addition, variations in gate speed between specifications and actual performance could affect the phase. To account for these factors, CBLECLK200 has been placed approximately 3.75 ns after CLK200A-CLK200D. The 270 degree phase delay is accomplished by using another M372. This flip-flop is clocked with the inversion of the 400 MHz clock, producing a shift of ninety degrees. The inversion of the 200 MHz clock is given as its input, adding the remaining 180 degrees. Two M374 differential receivers on the output of the flip-flop serve to balance the delay of its path with the delay of the clocks' path above.

### 3.2.5 Acknowledge Signals

The signals WAITOUT and CSACKOUT are passed differentially from the receiver to the transmitter. WAITOUT is a flow control signal that tells the transmitting PaRC to stop sending data when the receiver's buffers are full. CSACKOUT acknowledges the reception of packet across the network. When a processor needs to send a datum to another processor, this signal will indicate that the datum has reached the destination successfully. As these signals are control signals for PaRC, they are merely passed along and do not affect the data link circuitry.

### 3.2.6 Drivers

To provide greater output power for transmission, the Motorola ECL array supplies special output cells. For signals traveling across the link, the X271 differential drivers are employed. For single-ended output signals to PaRC (SYNCERR, GRAYERR, and PLOCALCLK), the X201 driver is used instead.

### 3.3 Receiver

The role of the receiver is to take the four serial nibbles from the transmitter and demultiplex them into the original sixteen bit data word for the receiving PaRC. To accomplish this, data, clock, and control signals are input. In addition, the receiver takes in two acknowledge signals WAITIN and CSACKIN from the receiving PaRC and passes them backwards to the transmitter. As output, the receiver has the sixteen bits DATAOUT[15:0] and a 50 MHz clock PCLKOUT that PaRC uses to synchronize the data. All input and output signals for the receiver can be seen in Figure 3-9.

The block diagram of the design is in Figure 3-10. The four nibbles first are shifted in at 200 MHz. These data subsequently are latched and output at 50 MHz for PaRC. Additional circuitry is required for control signal and clock distribution as well as for the generation of the output PaRC clock and for the acknowledge signals. A schematic of the receiver implementation is in Figure 3-11.

#### 3.3.1 Data Reception

The transmitted data first encounters the M374 differential receivers. Since noise acquired on the link will not affect the relative voltage levels of a signal and its complement, differential transmission is preferred. Although the receivers add delays, the delays are equal and thus do not affect the phase of the signals.

The data are first loaded into M291 flip-flops by the 200 MHz CBLECLK200 (see

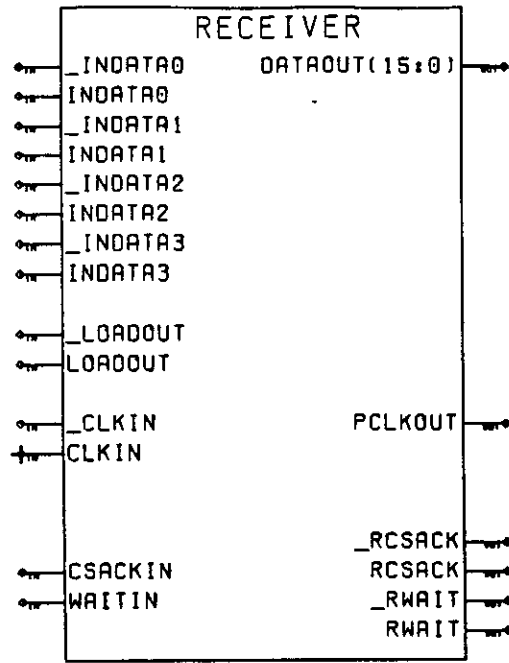


Figure 3-9: Receiver

Figure 3-10: Receiver Block Diagram

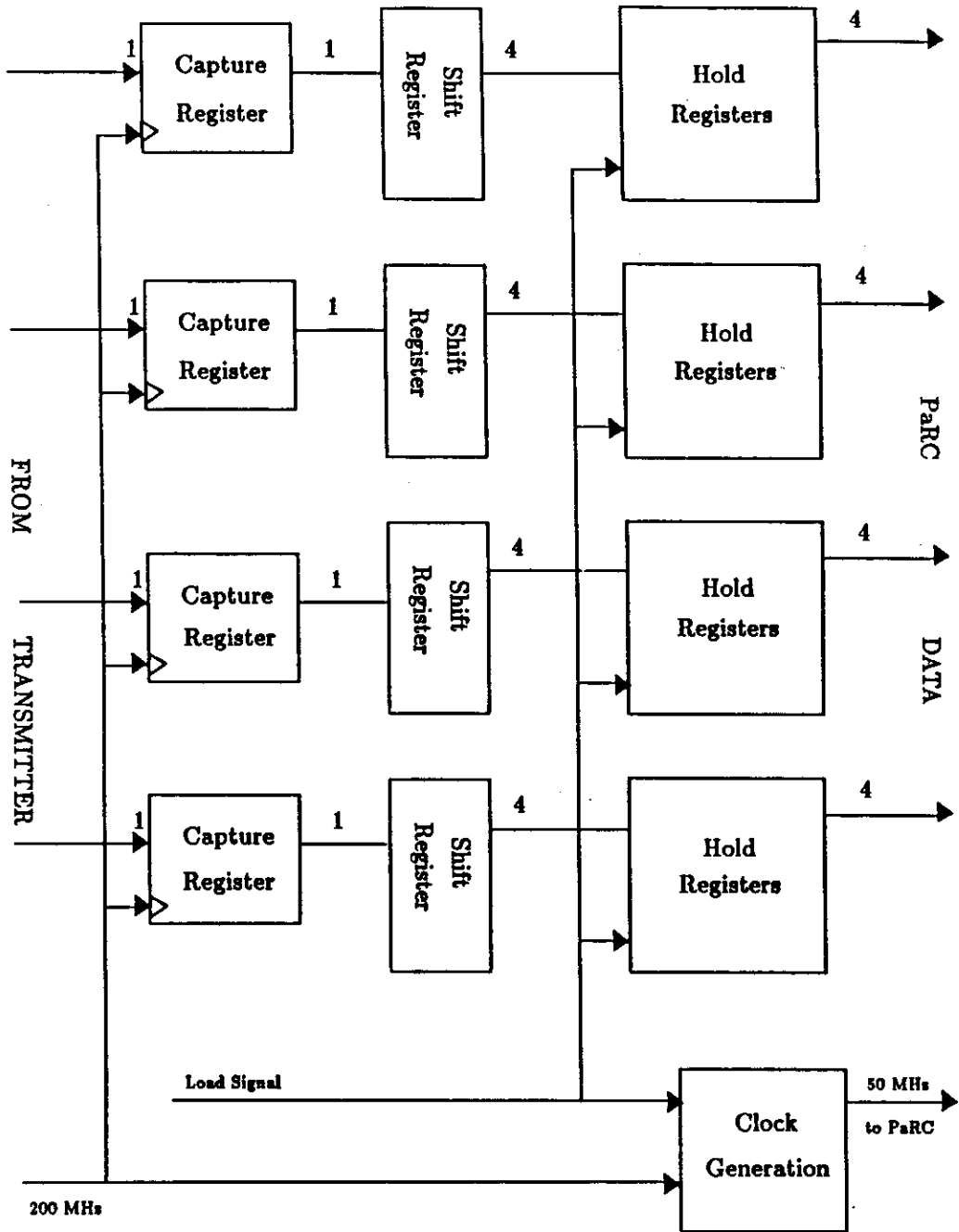
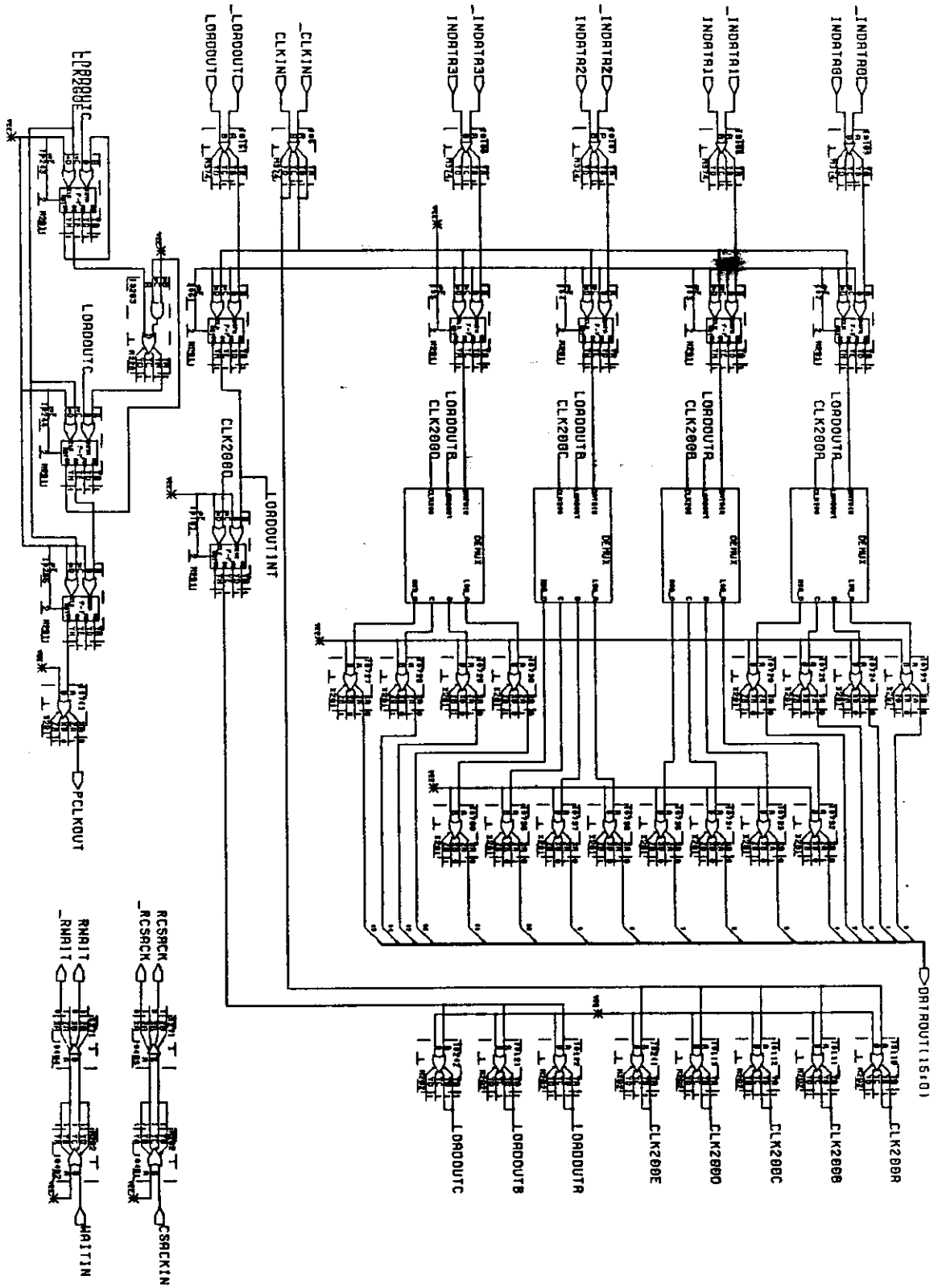


Figure 3-11: Receiver Schematic



200 MHz Clock Generation, Section 3.2.4). This latching is done at this first stage to lessen the chance for skew factors to alter the phase relationship sought in Section 3.2.4. After the data are latched, they pass to the data demultiplexing circuitry (Section 3.3.2 below). In addition to clocking these data, CBLECLK200 is distributed into six by using M202 OR gates (Figure 3-11). CLK200A through CLK200D are used for the demultiplexing.

### 3.3.2 Data Demultiplexing (DEMUX)

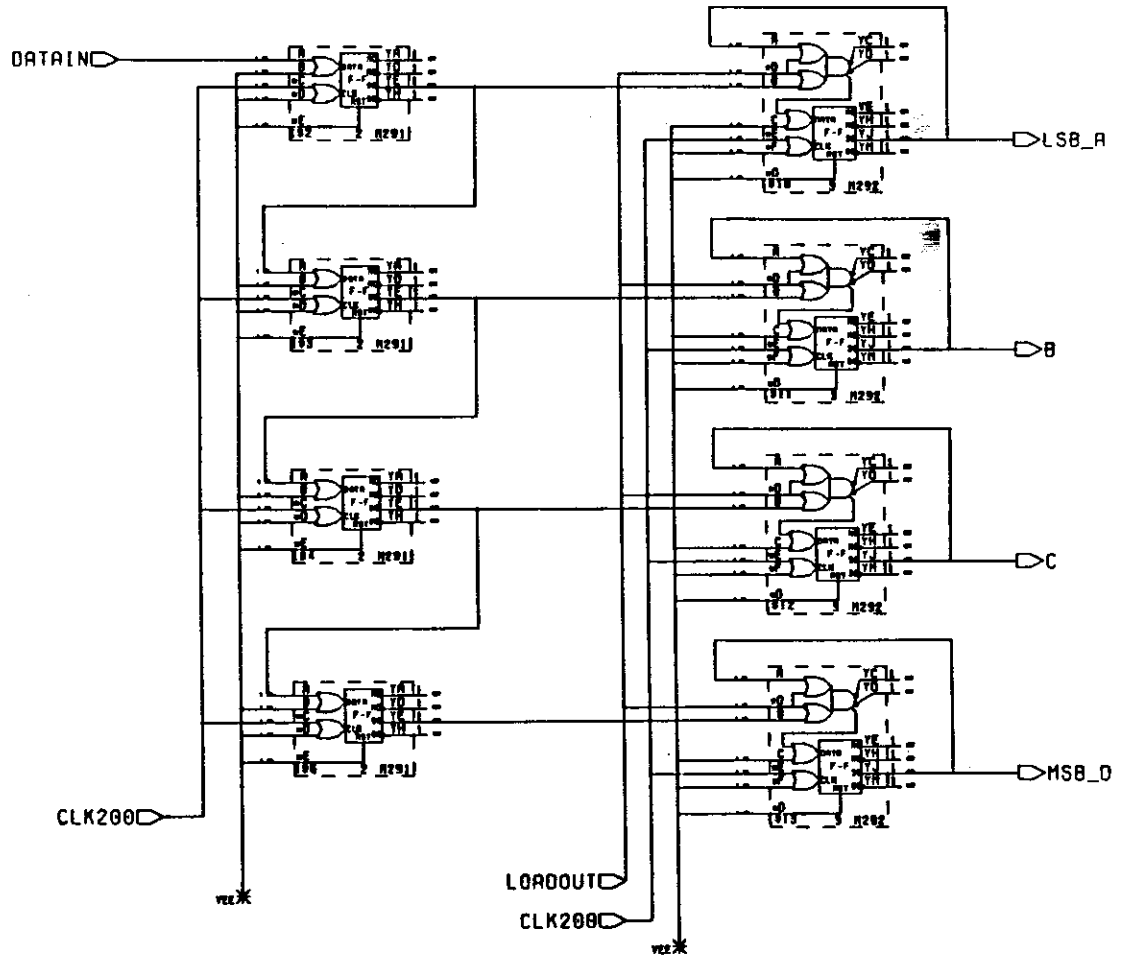
A schematic for the data demultiplexing circuitry is shown in Figure 3-12. The data received from the transmitter enters the circuitry as DATAIN. The four serial bits from the receiver then are shifted down with successive cycles. To present these four bits to PaRC in parallel, the bits are latched into four parallel registers and output for four 200 MHz cycles, i.e. at 50 MHz. The control signal that determines whether to latch the bits or present them is LOADOUT. This signal is sent across the link from the transmitter and is distributed to the demultiplexing circuitry as LOADOUTA and LOADOUTB (see Figure 3-11). LOADOUT is generated in the transmitter's synchronization circuitry and is the inverse of SHIFT (Section 3.2.1): three 200 MHz cycles low, one cycle high. Thus, the data from the shift flip-flops are loaded into the second set of flip-flops and remain for three more cycles, giving the 50 MHz needed for PaRC. The resulting sixteen bit word is presented as DATAOUT in Figure 3-11.

### 3.3.3 Clock Generation

The receiver is responsible for generating a 50 MHz clock to send to PaRC. This clock allows PaRC to synchronize the data it is given by the receiver. By placing the rising edge halfway between the changes in the data, the receiver provides maximum assurance that the data will not be clocked into PaRC while they are changing.

The clock is generated by using the most significant bit of a two bit counter. One copy of the input 200 MHz clock, CLK200E, clocks the flip-flops. A copy of LOADOUT

Figure 3-12: DEMUX Schematic





initializes the clock when it goes high. LOADOUT starts the clock by setting both bits high at the same time that it loads the demultiplexed data to the output registers. Two cycles later, the most significant bit goes from low to high, placing the rising edge in the desired location halfway between data transitions.

### 3.3.4 Acknowledge Signals

WAIT and CSACK are control signals for PaRC that are passed backwards from the receiver to the transmitter (see Section 3.2.5 for more information on the function of the signals). Like other signals traveling across the link, they are driven differentially.

## 3.4 Cable

### 3.4.1 Objectives

As the medium through which signals will travel across the network, the cable must meet the following requirements:

- Pass data at 200 Mbits over a maximum distance of forty feet
- Pass reliably a 200 MHz clock
- Have excellent noise immunity

To meet these goals, standard ECL voltage level drivers were employed. Standard ECL was chosen because of the availability of such drivers on the chip. To improve noise immunity, the signals will be sent differentially. Differential mode provides better high and low frequency noise rejection than single-ended transmission. By requiring more energy to cause errors, a standard one-hundred ohm differential impedance provides better noise immunity than that provided by a higher impedance.

### 3.4.2 Prototype

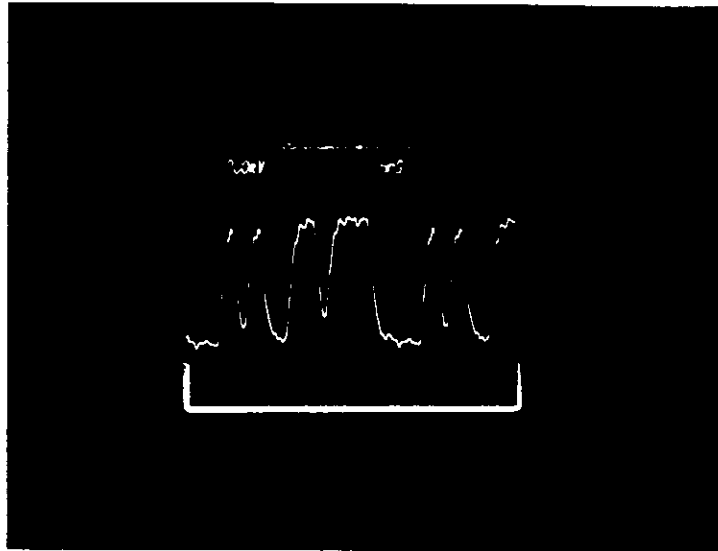
A suitable cable was found and a prototype was produced. The cable is a shielded parallel pair with a controlled impedance of 100 ohms. The conductors are twenty-six gauge silver-plated copper with a dielectric of expanded PTFE. The twenty-six gauge conductor was chosen because of its low attenuation at the required frequencies, permitting the use of conventional differential receivers. To provide greater noise immunity, an additional silver-plated copper braid bundles all of the cables of a given link.

### 3.4.3 Testing

To prove the integrity of the prototype, a test circuit was designed. Constructed from gallium arsenide components, the circuit sought to analyze the signals transmitted at 400 MBits per second through forty feet of cable. A thirteen bit test pattern was driven cyclically and received over the cable; the received signal was compared to the original signal and errors were counted. These tests were performed for periods of several hours.

The waveform of the test pattern is shown in Figure 3-13 and the waveform after reception is in Figure 3-14. These waveforms demonstrate that a rise time of 2 ns is required for the signal to reach a stable level. This condition improves considerably at twenty feet, the maximum cable length in the Monsoon prototype (resulting waveform in Figure 3-15).

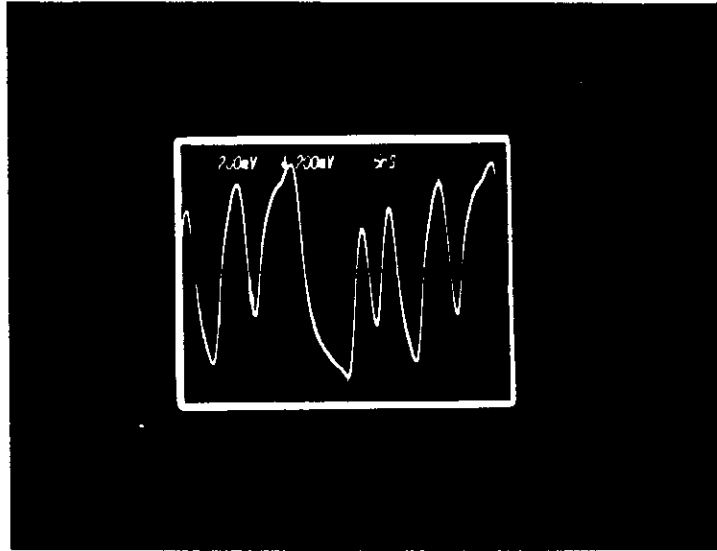
To evaluate the cable's resistance to external electro-magnetic influences, the cable was subjected to noise and the effects were noted with the test circuitry. A static discharge gun was used to generate strong broad-band electro-magnetic noise. The expanded PTFE cable was able to withstand without errors 15 kV discharges at a distance of one inch. The cable was able to withstand 1.5-2 kV discharges direct to the cable shield.



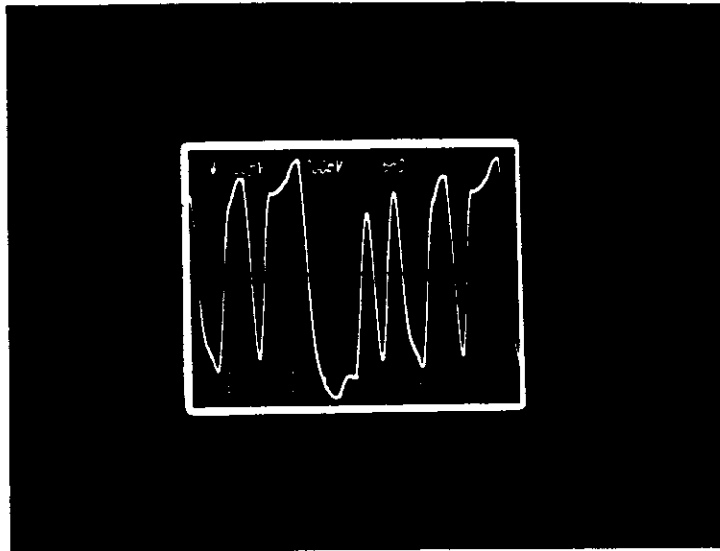
**Figure 3-13: Waveform of Test Pattern**

#### **3.4.4 Conclusions**

The expanded PTFE cable appears to fulfill its design goals for the lengths needed in Monsoon. Furthermore, as mentioned in Section 2.2.1, the immunity is such that data integrity can be assured without encoding and error checking, simplifying the data link design. However, while a prototype has been fabricated, more extensive tests must be conducted before the manufacture of the final product.



**Figure 3-14: Waveform Received over Forty Feet**



**Figure 3-15: Waveform Received over Twenty Feet**

## Chapter 4

### Verification

Upon completion of the design, testing was required to verify the functionality and the fulfillment of design issues as detailed in Chapter 2. With the IDEA development system, simulations were run to analyze individual sub-systems and their interfaces to form the total system. Timing issues were scrutinized, taking into account the conservatism of the computer models and their absence of analog considerations. Logically, the circuit does meet its design objectives. Metastability, wire delay (the simulator assumes a median wire length), and cable skew (timing variability between signals arriving over different cables) were considered implicitly in the design but are known only in the worst case.

After fabrication of the chip, Motorola must verify the absence of errors in the manufacturing process that would affect performance. Test routines were written to search for the presence of nodes in the design that are stuck in a high or low state. If a signal is stuck in either state, the output of the routine would be altered noticeably and the chip would be rejected. No software exists to guarantee that the test routines detect all stuck-at faults. However, after the routines were verified by hand, it is believed that full coverage is provided.

## Chapter 5

### Conclusions

As of the completion of this thesis in May 1989, the data link chip has been implemented and the design is ready to be handed to Motorola for fabrication. The design has performed well in simulation and successfully has fulfilled its initial objectives. The elimination of encoding did not prove to sacrifice data integrity; on the contrary, it permitted a longer clock period that gave the signals additional time to settle to reliable levels. The synchronization circuitry, so critical to the data link's interface with PaRC, appears to guarantee within acceptable bounds changes in phase from analog considerations. After much concern about the quality of the link's cable, a suitable cable has been chosen that adequately resists outside noise.

With the successful fabrication of the data link chip, the Monsoon project will have reached another milestone toward its ultimate completion. A complete architectural definition of a 4x4 network switch board (see Chapter 1) is expected by the fourth quarter of 1989. The first prototype of the network board should follow in the first quarter of 1990. Finally, a Monsoon prototype with eight processor units and eight memory and storage units should be unveiled in the third quarter of 1990.