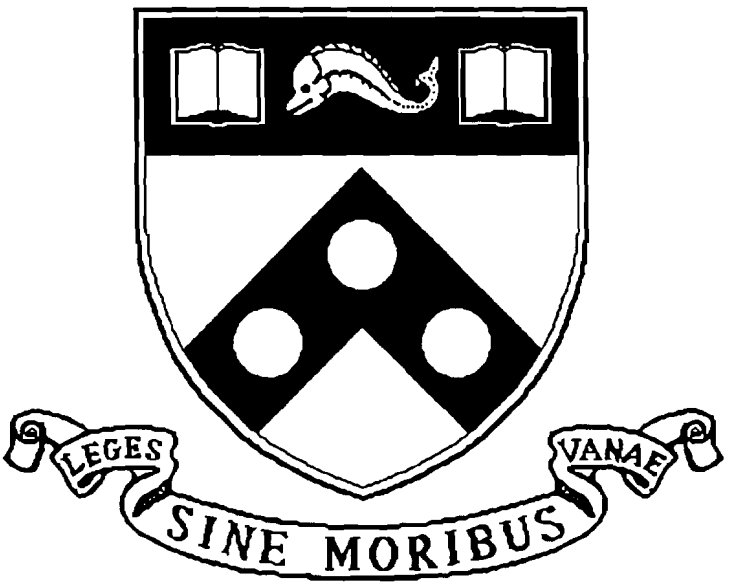
Evaluation of the GLINK Chip Set as a Data Link Layer for Local ATM

MS-CIS-93-88
DISTRIBUTED SYSTEMS LAB 35

Chris Russo



University of Pennsylvania
School of Engineering and Applied Science
Computer and Information Science Department
Philadelphia, PA 19104-6389

October 1993

Evaluation of the GLINK Chip Set as a Data Link Layer for Local ATM

Chris Russo

University of Pennsylvania
Philadelphia, Pennsylvania 19104

Abstract

The introduction of the HDMP-1000 Gigabit Rate Transmit/Receive (GLINK) Chip Set by Hewlett-Packard has provided a hardware device capable of matching SONET STS-3c up to nearly STS-24c rates over either coaxial cable or optical fiber. This capability enables high speed local communication over a low cost medium, and lends itself well to use as both the physical and the data link layer for ATM in a local (few hundred feet) environment. This paper analyzes the capabilities of the GLINK chip set in terms of link length and data rate trade-offs, and in terms of its usefulness as a physical and DLL for local ATM.

1. Introduction

This technical report describes a study and design exercise done to ascertain the feasibility of using the Hewlett-Packard HDMP-1000 Gigabit Rate Transmit/Receive (GLINK) Chip Set as the physical and data link layer (DLL) for local asynchronous transfer mode (ATM) communication. The work reported here began with an evaluation of the performance characteristics of the GLINK chip set in which maximum data rate as a function of coaxial cable link length was measured to determine the feasibility of local high speed (155 MBps and up) serial communication via low cost RG-58 coaxial cable. The experimental set up and methodology for this analysis is described in the GLINK Physical Layer Performance Characterization section of this report.

The design phase of this work was conducted to verify that a working local ATM transport system using coaxial cable and the GLINK chip set could be built. The design objective was relatively straight forward: provide a SONET-like interface to the ATM Host Interface (HI) Segmenter and Reassembler boards[1] (previously designed and constructed at the University of Pennsylvania), and use the GLINK chip set to provide the high speed physical link via coaxial cable. A description of the components in the system and some background information on those components is provided in the Local ATM/GLINK Architecture section of this report. Specifics of the interface board designed and built for the evaluation are described in the GLINK Interface Design section of this report.

The Results section of this report contains the results and analysis of the GLINK physical layer evaluation and the final design information about the interface board. Finally, the Conclusions section contains a brief conclusion statement as well as comments on future work.

2. GLINK Physical Layer Performance Characterization

The first part of the analysis of GLINK's usefulness in a high speed local ATM environment was to determine its data rate and link length capabilities. For the local environment, a low cost easy to work with medium such as coaxial cable is desirable over fiber and other more costly alternatives. Coaxial cable is readily and cheaply available, easily connectorized, and durable, making it a good choice of medium for local ATM connectivity. The GLINK chip set is capable of directly driving a 50 ohm line without additional driver circuitry.[2] This is not the case when using GLINK with a fiber link where expensive optics and driver circuitry are required in addition to the GLINK chips. The focus of the data rate vs link length study was therefore limited in this work to determining the performance curve for GLINK operating via differing lengths of RG-58 50 ohm coaxial cable.

2.1 GLINK Chip Set Description

The GLINK chip set is formally known as the HDMP-1000 Gigabit Rate Transmit Receive Chip Set produced by Hewlett-Packard. The chips take as input a 16, 17, 20, or 21 bit wide parallel data word and transmit it over a serial link, reconstructing it at the receiving end. The chips are performing a parallel to serial/serial to parallel conversion with a maximum rated serial throughput of 1.2 GBps. The chip set uses ECL technology and comes packaged as a pair of 64 pin ceramic quad flat pack (CQFP) packages, one for

transmit (HDMP-1002) and one for receive (HDMP-1004). All inputs and outputs to/from the GLINK chips are at ECL levels. The chip set is capable of driving a 50 ohm load as the serial link directly, or can be interfaced to operate over an optical link. The GLINK chip set costs about $330 per pair in quantity.[3]

The GLINK chip set is designed for operation in a full duplex configuration, but can also be run in a simplex mode. It can be configured to operate with a 16 or 20 bit wide data path, with an extra (17th or 21st) bit available via the "flag" bit, however the use of the flag bit reduces the robustness of the error detection on the link. The clock rate for the incoming parallel data is variable from approximately 5 MHz to 70 MHz (depending on the word width configuration), with the GLINK chip set receiving its clock from the data source. The GLINK TX chip internally multiplies up the input data clock (STRBIN signal) to produce the high speed clock for the serial link. Settings for the GLINK chip set corresponding to the correct multipliers for its PLL circuitry (DIV0, DIV1 inputs) must be set appropriately for the incoming data rate being used.[2]

All of the work described in this report was performed using an HDMP-100K Gigabit Rate Transmit/Receive Chip Set Evaluation board provided by HP.[4] This board contains a transmit and receive pair of GLINK chips as well as some ECL buffering circuitry and loading resistors, all configured for easy connection via headers and SMA type coaxial connectors. The Evaluation board provides a convenient vehicle for assessment of GLINK's capabilities without having to design and produce a board of one's own.

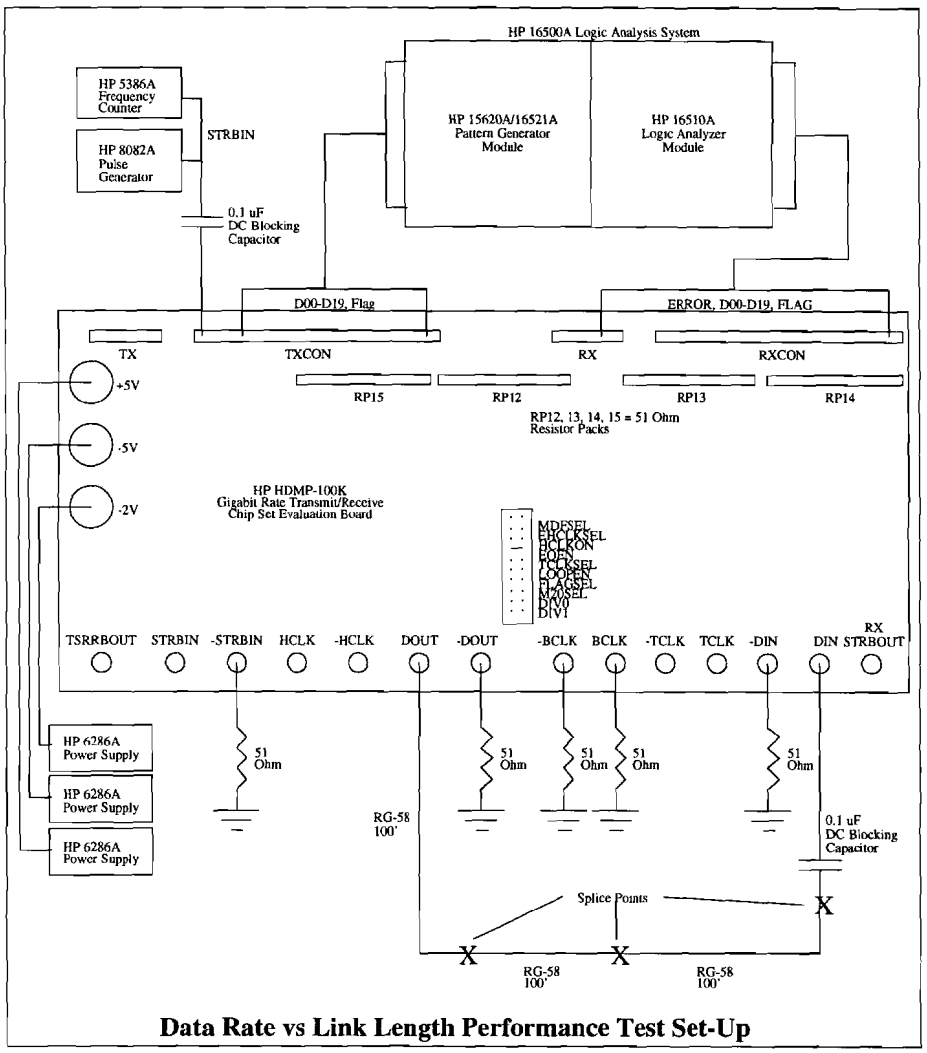
2.2 Data Rate vs Link Length Test Environment

The test setup used in evaluating the GLINK data rate vs link length performance is shown in Figure 1. The Evaluation board was configured for a single coaxial link and was set up to loop the transmit data out (DOUT) signal to the receive data in (DIN) input via differing lengths of RG-58 coax cable. The clock source used to provide the STRBIN signal for testing was an HP Model 8082A pulse generator. It was coupled to the STRBIN input via a 0.1 uF capacitor. The data source was an HP 16520A pattern generator, running at a slower rate than the clock source (the maximum rate for the pattern generator was 50 MHz). On the receive side, the output data and some of the system control flags were monitored using an HP 16510A logic analyzer. On the RX side of the board, 51 ohm resistors were added (RP12-RP15) from each ECL output to the -2 volt supply for proper loading. The unnecessary buffer line logic (BLL) outputs were loaded with 51 ohm resistors to ground. The cable equalization (EQEN) input was enabled for the duration of the testing.

The coaxial link from TX DOUT to RX DIN was coupled via a 0.1 uF capacitor. The maximum link length attempted was 1000 feet, and data was collected at lengths ranging from 300 feet down to 50 feet. The coax was connected to the GLINK Evaluation board via SMA connectors, and the coupling capacitor was spliced into the coax 3 inches from the RX DIN connector. The coax cable also had a splice every 100 feet, each splice being soldered both on the center conductor and braid. The testing was done by cutting off lengths of coax from the receive end, and reconnecting it at the coupling capacitor near the RX DIN connector for each length. While the splices in the coax probably caused slight impedance mis-matches and reflections, no effect in the test data was obvious.

2.3 Data Rate vs Link Length Test Procedures

The tests were performed as follows. For a given length of coax, the STRBIN frequency provided by the pulse generator was increased until the ERROR output from the RX chip was triggered. This was tracked by setting the logic analyzer to trigger off a transition in the ERROR output from the GLINK Evaluation board. The ERROR signal from the GLINK RX chip is set when the received word is not recognized as either a valid data or control word by the receiver. The data being provided as the TX input was a simple counting pattern being generated by the pattern generator (asynchronously with respect to the pulse generator providing the STRBIN signal) at a rate approximately one-tenth the STRBIN rate. For each word-length (16, 17, 20, or 21 bits, set via the M20SEL and FLAG inputs), the STRBIN frequency was increased until ERROR signals were detected, and then backed off until the ERROR signals disappeared. The point at which the ERROR signals disappeared was recorded as the maximum rate for that link length and configuration. The DIV0 and DIV1 inputs were adjusted throughout the testing to keep the STRBIN frequency within the specified range for the GLINK chip set's PLL circuitry.



Data Rate vs Link Length Performance Test Set-Up

Figure 1

First, two tests were done to get a set of upper and lower bounds on the link length vs data rate performance of GLINK. To bound the absolute maximum link length, a 1000 foot roll of RG-58 coax was inserted as the data link, and transmission at any rate was impossible. The second test performed was to determine the maximum transmission rate with an optimal link. This was done using the 50 ohm microstrip machined into the Evaluation board printed circuit board for operation of GLINK in loopback mode. When the LOOPEN input to the GLINK Evaluation board is asserted, the loopback on the board is used and the DOUT and DIN signals are ignored. Assuming the etched path on the board provides the best possible impedance match for transmission and a short link length (low loss), one can also assume that the maximum transmission rate possible with the chip set will be attained using the loopback path on the board. With LOOPEN asserted, a maximum STRBIN frequency of approximately 73 MHz was attained.

The bulk of the testing was done for link lengths varying from 300 feet down to 50 feet. At 300 feet, the maximum STRBIN frequency was near the minimum for GLINK operation (4.6 MHz), so 300 feet is the approximate maximum working link length for GLINK over coax at any rate. At 50 feet, the maximum STRBIN rate was equal to that attained in the test described above using the loopback etched into the board, so no testing for lengths shorter than this was done. Data points were taken at 20 foot increments from 300 feet down to 80 feet, and at 10 foot increments from 80 feet down to 50 feet. The raw data gathered and plots of the performance curves produced from that data appear in the Results section of this report.

3. Local ATM/GLINK Architecture

The architecture for the local ATM system using GLINK built and tested here consists of four components: the ATM HI Segmenter and Reassembler boards,[1] the PC in which they are installed, the interface board constructed as part of the evaluation, and the GLINK Evaluation board.[4] To avoid confusion between the Host Interface boards and the interface board built as part of this work, the later will be referred to as the "Glue" board (describing its function to fit and stick the GLINK technology to the Host Interface Segmenter and Reassembler boards). The configuration is shown in Figure 2, entitled "High Level Local ATM/GLINK Test Prototype Architecture."

The assumption behind this test architecture was that it would be acceptable to use a single GLINK Evaluation board to test the basic design in a loopback configuration, and then assume its expandability to the full duplex point-to-point case. During the testing, two PCs were used: an IBM PS2 and an IBM RISC System/6000. In both cases, the PC ran the software to drive the Host Interface boards, and the Host Interface boards used the Glue board and GLINK Evaluation board (together) to communicate via the coaxial cable. The test cell data was a simple counting pattern.

The ATM HI boards were designed to operate with a SONET physical layer with an eight bit wide data path, and three overhead signals which determine the way in which the data bytes should be interpreted.[5] The header enable (HEN) signal designates a byte to be one of the five header bytes of the ATM cell. The path overhead enable (PEN) and transport overhead enable (TEN) signals designate a byte as being SONET overhead, and therefore it should not be considered part of the ATM cell. The clock signals (TXBYTECLK and RXBYTECLK) to the ATM HI boards are taken from the physical layer connecting them. The primary purpose of the Glue board was to provide the appropriate timing and signals to the ATM HI boards. Secondary functions of the Glue board include ECL-TTL and TTL-ECL level translations, and multiplexing/demultiplexing the 8 bit wide outputs/inputs of the ATM HI boards for transmission via the 16 bit wide GLINK data path.

4. GLINK Interface (Glue) Design

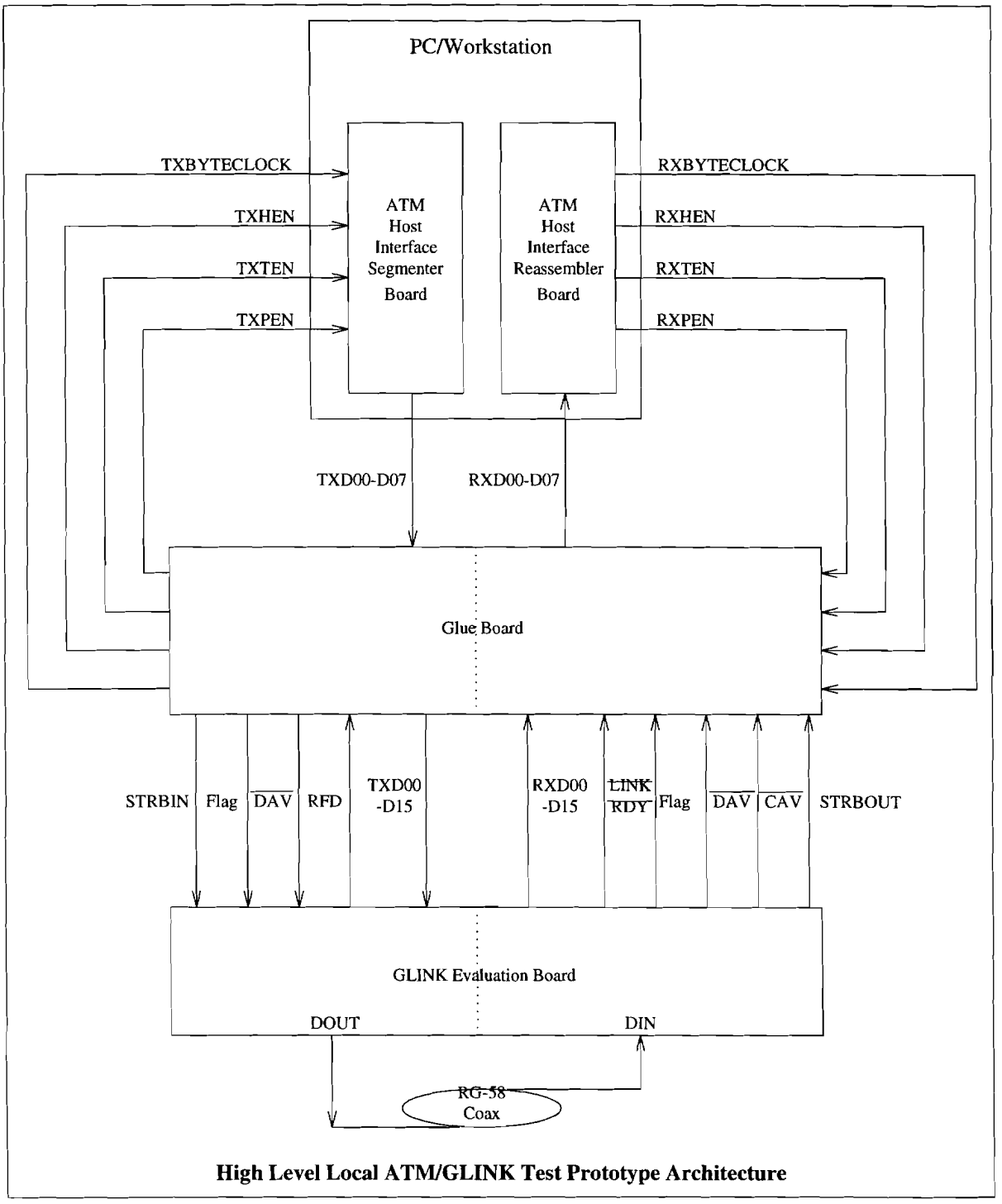
The design of the Glue board occurred in two phases, first concentrating on the TX side (getting the TX data from the ATM HI Segmenter board to the GLINK Evaluation board), then the RX side (getting the RX data from the GLINK Evaluation board to the ATM HI Reassembler board). This section of the report is similarly organized.

The mapping of the ATM cells onto the GLINK data path was done according to an application note prepared by HP on the subject[6] so as to be compatible with others' designs. Since the work of this paper was done, it is the author's understanding that minor changes have been made to the specifications in the application note referenced here.

The TX and RX portions of the Glue board both were implemented using Altera EPM5032 CMOS Electrically Programmable Logic Devices (EPLDs) in 28 pin DIP packages. These EPLDs were used to implement state machines to control the mapping of the ATM cell bytes into the GLINK frames on the TX side, and the extraction of the ATM cell bytes from the received GLINK frames at the RX side. The performance of the state machines was simulated using the Altera Max Plus 2 software on a PC, and the EPLDs were similarly programmed using Altera software and hardware and a PC. Two EPLD chips were used in the overall Glue design (as opposed to a single, larger chip) to compartmentalize the design, allowing for simpler wiring and easier troubleshooting. The EPLD code for the state machines is included in Figures A-1 and A-4 in the Appendix of this report.

4.1 Glue Transmit Portion Design

A block diagram of the design of the TX portion of the Glue board is shown in Figure 3. The design utilizes a single state machine of 56 states corresponding to the 53 bytes in an ATM cell plus three overhead bytes as specified in the HP ATM/GLINK Application Note/Specification.[6] The TX state machine provides the proper inputs to the ATM HI Segmenter (TXBYTECLOCK, TXHEN, TXTEN) to prompt it for bytes of cells, specifying to the Segmenter whether the byte expected is to be a header, payload, or overhead byte. The TX state machine controls the mapping of bytes from the Segmenter into the GLINK data word for transmission by controlling operation of the latch enable (LE) input to latch 1 (a 74F845 8 bit transparent



High Level Local ATM/GLINK Test Prototype Architecture

Figure 2

latch with tri-state outputs). Latch 1 holds the first of two bytes latched on its output while the second byte is received from the ATM HI Segmenter. When both bytes are available at the GLINK board as a 16 bit wide word, the STRBIN clock pulse is provided to the GLINK board and the data is transmitted. Latches 2 and 3 are not used, but sockets were wired into the data path when constructing the prototype to allow for their future use if correcting for timing should have required another level of latching. In the prototype, DIP headers containing jumpers were inserted in the sockets for latches 2 and 3. (Timing data drawn out as

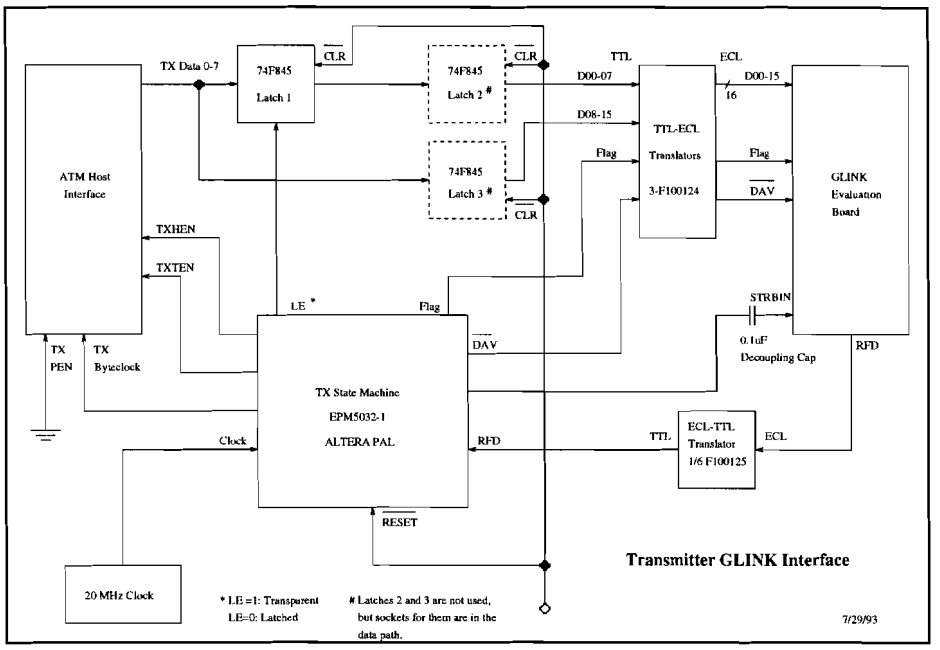


Figure 3

part of the design effort for the TX portion of the Glue board showing the relative timing/delays of signals through the components is shown in Figure A-2 in the Appendix of this report.) The TX state machine provides the Flag signal to the GLINK board, as specified in the ATM/GLINK Application Note/Specification,[6] representing the start of a new cell. The TX state machine loops the ready for data (RFD) output from the GLINK board back to it as its data available (DAV*) input.

The TX state machine provides a frequency division function since the GLINK board operates at half the speed (10 MHz) of the ATM HI Segmenter (20 MHz). The GLINK can be operated this slowly because it has a data path twice as wide as that of the ATM HI Segmenter. Operation at this rate is comparable to a SONET STS-3c connection (approximately 155 MBps). The clock for the TX portion of the Glue board is provided by a 20 MHz TTL oscillator.

A bank of three F100124 Hex TTL-ECL converters and 1/6 F100125 Hex ECL-TTL converter was necessary for level conversion between the TTL circuitry of the ATM HI Segmenter and the ECL circuitry of the GLINK Evaluation board. For the prototype, the reset signal to the Glue board is provided via a pushbutton switch.

4.2 Glue Receive Portion Design

The receive portion of the Glue board design is slightly more complex. A block diagram of the RX portion of the Glue board is shown in Figure 4. A major simplifying design specification is that the transmitter will constantly be transmitting null cells when there is no real data to be carried, thus keeping the line synchronized and avoiding the problems of restarting and synchronizing whenever a cell is to be transmitted. Even with this assumption, the following issues still complicate the design of the RX portion of the Glue board:

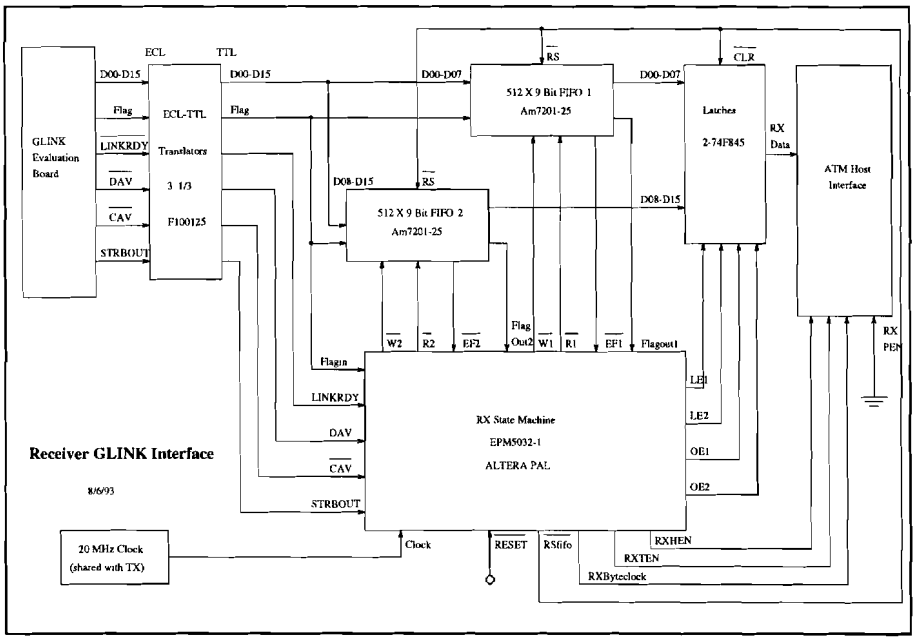


Figure 4

1. It must synchronize to the transmitter upon being powered up or reset, as well as when the transmitter is powered up or reset.
2. It must detect framing errors in the form of incomplete cells, reset itself and resynchronize accordingly.
3. The GLINK Evaluation board provides the received data as 16 bit words at a rate of 10 MHz (10 Mwords/second), but this data must be fed to the ATM HI Reassembler board as 8 bit words at a rate of 20 MHz (20 Mwords/second). To do this, either the clock coming from the GLINK Evaluation board receiver must be doubled in frequency, or buffering and storage must be added to allow asynchronous operation of the Reassembler board and the GLINK board which is feeding data to it.

To elaborate on the last item, a 20 MHz signal was necessary to clock the RX state machine and the ATM HI Reassembler board, but the STRBOUT clock signal coming from the RX portion of the GLINK board was only 10 MHz; a frequency doubler of some sort was necessary. Since a simple, reliable frequency doubling mechanism was not easily implementable here, two other alternatives were considered. The first alternative uses the high speed clock coming off the GLINK board: the BCLK signal. This signal is the clock for the serial link, operating at 20 times the STRBIN frequency. Using this signal and a cascade of flip-flops for frequency division, a 20 MHz signal synchronized to the GLINK TX clock could be generated. The other alternative was to use FIFOs to receive and hold data from the GLINK RX and store it until needed by the ATM HI Reassembler board, thus allowing for the two to operate asynchronously. The second option was chosen because of the expected complications in building a reliable frequency divider for use in the 200 MHz range, and the expected need for some sort of buffering for synchronization purposes as well as for clocking. Two 512X9 Bit (Cypress CY7C421-25) FIFO's were used, with the first eight bits of each storing the 16 data bits from the GLINK board, and the ninth bit on both FIFO's being

connected to the Flag output from the GLINK board. The write signal for the FIFO's was generated by ANDing the data available (DAV), control word available (CAV*), and STRBOUT signals from the GLINK board via the RX EPLD.

The data is alternately read from the two FIFOs at a clock rate of 10 MHz (from each) and then multiplexed into a single 8 bit signal at a 20 MHz rate using two 74F845 latches. The RX state machine controls this multiplexing process by alternately enabling the outputs of the two latches, both of which are connected to the single 8 bit data input of the ATM HI Reassembler.

The clock signal for the RX portion of the Glue board is the same 20 MHz TTL oscillator providing the clocking for the TX portion of the board. The ECL-TTL translation is performed by a block of F100125 translator chips, using the inverting outputs where convenient for a few of the control signals (LINKRDY, DAV).

4.2.1 RX State Machine Design The first two design issues for the RX portion of the Glue board (listed earlier) relate to attaining and maintaining synchronization with the transmitter, and this is done under the control of the RX state machine, the code for which appears in Figure A-4 in the Appendix of this report. The RX state machine takes as inputs signals from the GLINK board and status signals from the FIFOs, and provides the control signals to the FIFOs, the latches, and the ATM HI Reassembler. The RX state machine consists of an underlying set of state machines, one being similar to the TX machine, having 56 states representing the 53 bytes of the ATM cell and three bytes of overhead. The other state machines are involved in synchronizing the receiver to the transmitter according to the process shown in the flowchart in Figure 5 and described below.

The state machine monitors the Flagin signals coming from the GLINK board, and counts them until two flags have passed, showing that at least one full cell has been stored in the FIFOs. At this point, the data will start being read from the FIFOs (the state machine will begin pulsing the read lines to the FIFOs), and the RX state machine will monitor the Flagout1 signal from the output of the FIFOs. When the first flag signal is read from the FIFOs, the first full cell of data is beginning, and the state machine begins feeding the data to the ATM HI Reassembler by de-asserting the RXTEN signal, and correctly controlling the latch enable (LE) and output enable (OE) signals to the latches. Once the system has reached this state, the receiver is synchronized to transmitter. Until it reaches this state, the RXTEN signal to the ATM HI Reassembler is asserted, signifying that the bytes it is receiving are overhead and should be ignored.

The state machine, upon attaining synchronization with the transmitter, monitors the flagout signals from the FIFO's constantly to detect if a flag appears at the wrong time, or if a flag is dropped, showing a loss of synchronization. Should this occur (due to a power glitch, or the transmitter being inadvertently reset), the RX state machine goes back to its start up state, flushing the FIFOs, resetting the latches, and putting the ATM HI Reassembler back on standby by asserting the RXTEN signal. It then follows the same procedure described above to regain synchronization. The cell or cells that may be lost due to such a glitch or during synchronization will not be recovered at this level. It is left to higher levels in the communication protocol to detect the data loss and retransmit any missing cells.

Other signals that were wired to the RX EPLD, but not used at this point in the RX state machine logic are the empty flags from the FIFOs. These signals, as well as the FIFOs' full flags may be useful at some point in detection of system status and correction should clock slips occur between a given transmitter and receiver to the point where the FIFOs get over- or under-run.

4.3 Glue Physical Component Layout and Design

The Glue board was built using wire-wrap construction techniques on a board with an embedded ground plane to provide noise immunity. Every power supply connection to each chip on the board has a 0.1 uF capacitor in parallel with it to decouple the chips from power supply glitches and noise. The layout of components and connections on the Glue board is shown in Figure 6, entitled "ATM Host Interface to GLINK Evaluator Board Interface Physical Layout." The decoupling capacitors are not shown in Figure 6 for simplicity. Connections to the GLINK Evaluator board are made via two 60 pin headers and two ribbon cables, and the connection to the ATM HI boards is made via a single 34 pin header. A schematic diagram of the final Glue circuit is shown in Figure 7.

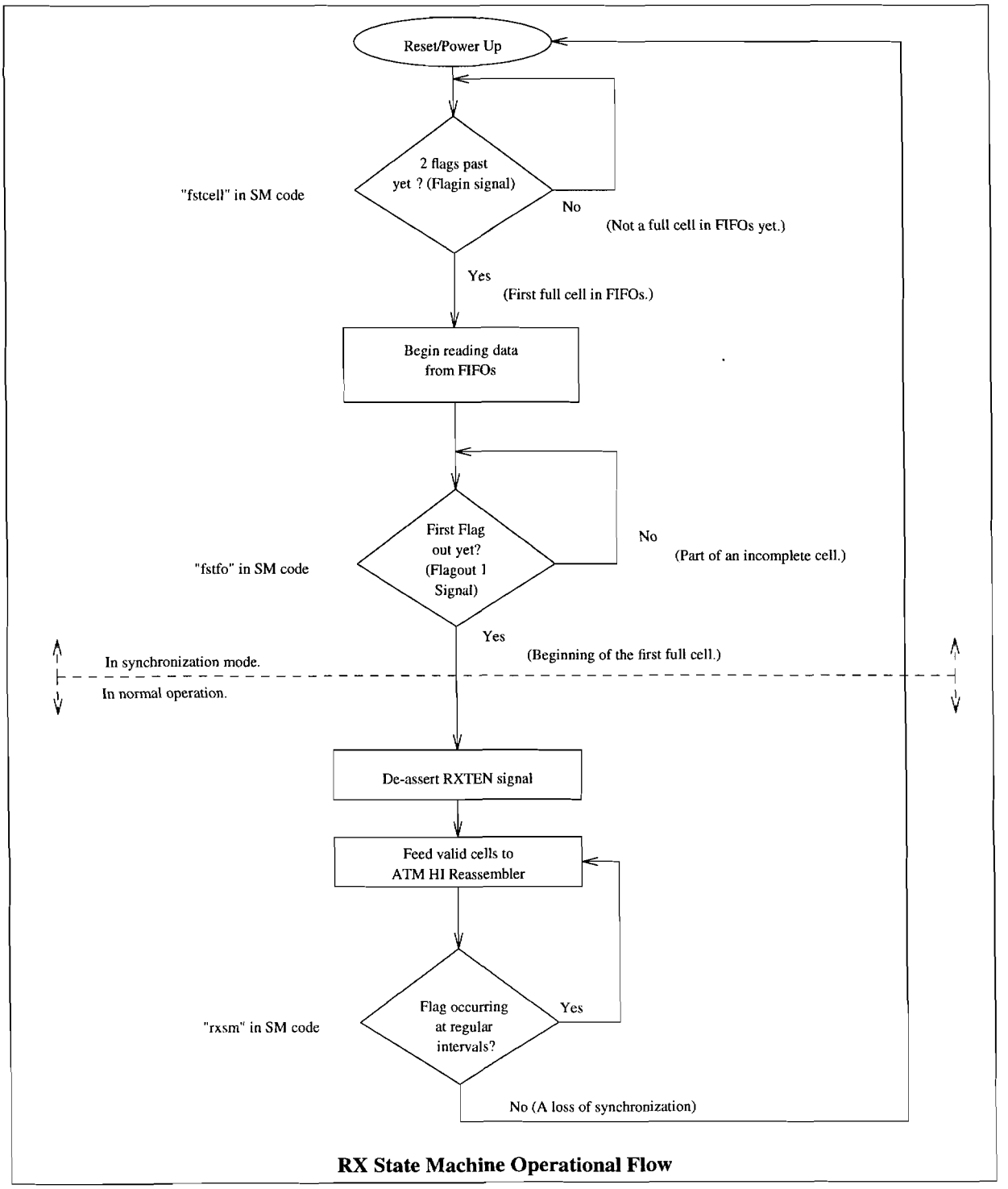


Figure 5

The physical layout was done with the design goals of space conservation on the board in case unforeseen details would require additional components, and design subsystem compartmentalization for short, simplified wiring and troubleshooting. For these reasons, the components for the TX and RX portions are clustered separately on the board as shown in Figure 6.

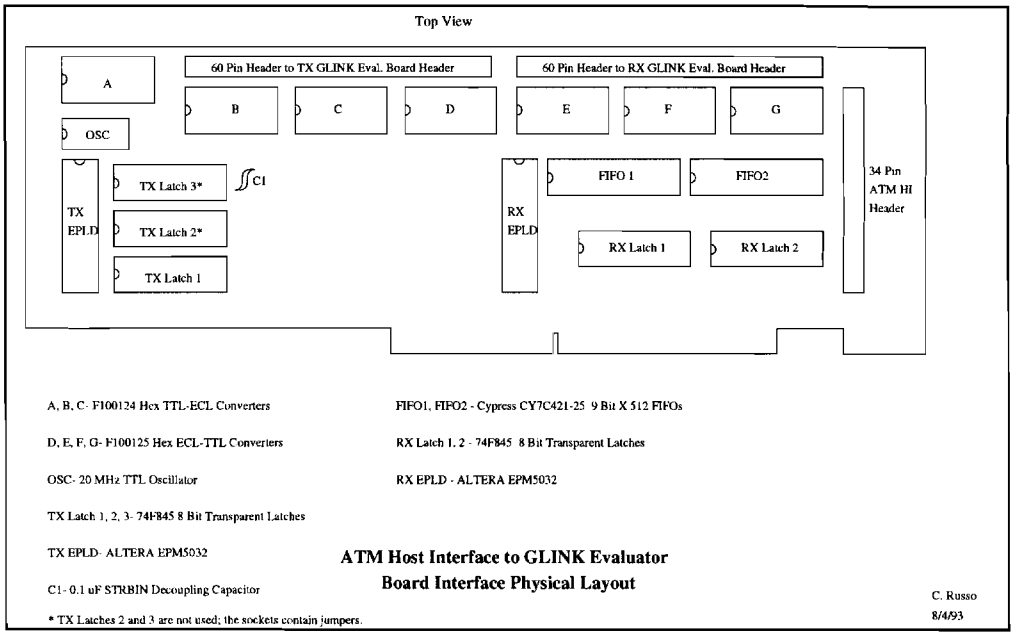


Figure 6

5. Results

This section contains the results of the data rate vs link length analysis of the GLINK chip set and the final design information and analysis of the Glue board.

5.1 Data Rate vs Link Length Test Data

Table 1 contains the data rate vs link length data gathered as described in section 2.3 of this report. Performance curves were generated from this data. The first plot shown in Figure 8 entitled "STRBIN Frequency vs Link Length" contains a simple plot of the raw data from Table 1. Figure 9, entitled "Bit Rate vs Link Length" was generated by simply multiplying the STRBIN frequency by the data word length for each mode tested.

Analysis of Figure 8 reveals a few things about the performance of the GLINK chip set, as well as a few things about the experimental design used in the analysis. The use of three 100 foot sections of cable spliced together did not have an obvious adverse affect on the test results; no abrupt change in the slope of the plots occurs at the splice points. It appears that the 17 and 21 bit modes are capable of supporting a slightly higher STRBIN frequency, even though the general trend is that the smaller the word length, the higher the maximum STRBIN frequency. This antiintuitive result may be explained by the experimental test set-up. The maximum STRBIN rate was determined by the occurrence of the ERROR flag at the receiver. The robustness of error detection on the link is diminished when the Flag bit is used to carry data; in the 17 and 21 bit modes. With this in mind, it can be theorized that the maximum STRBIN frequency in 17/21 vs 16/20 bit modes may be slightly inflated due to errors going undetected. From Figure 9, it can be seen that the link length for operation at 155 MBps can be as long as approximately 250 feet, and for operation at 622 MBps it can be as long as approximately 110 feet.

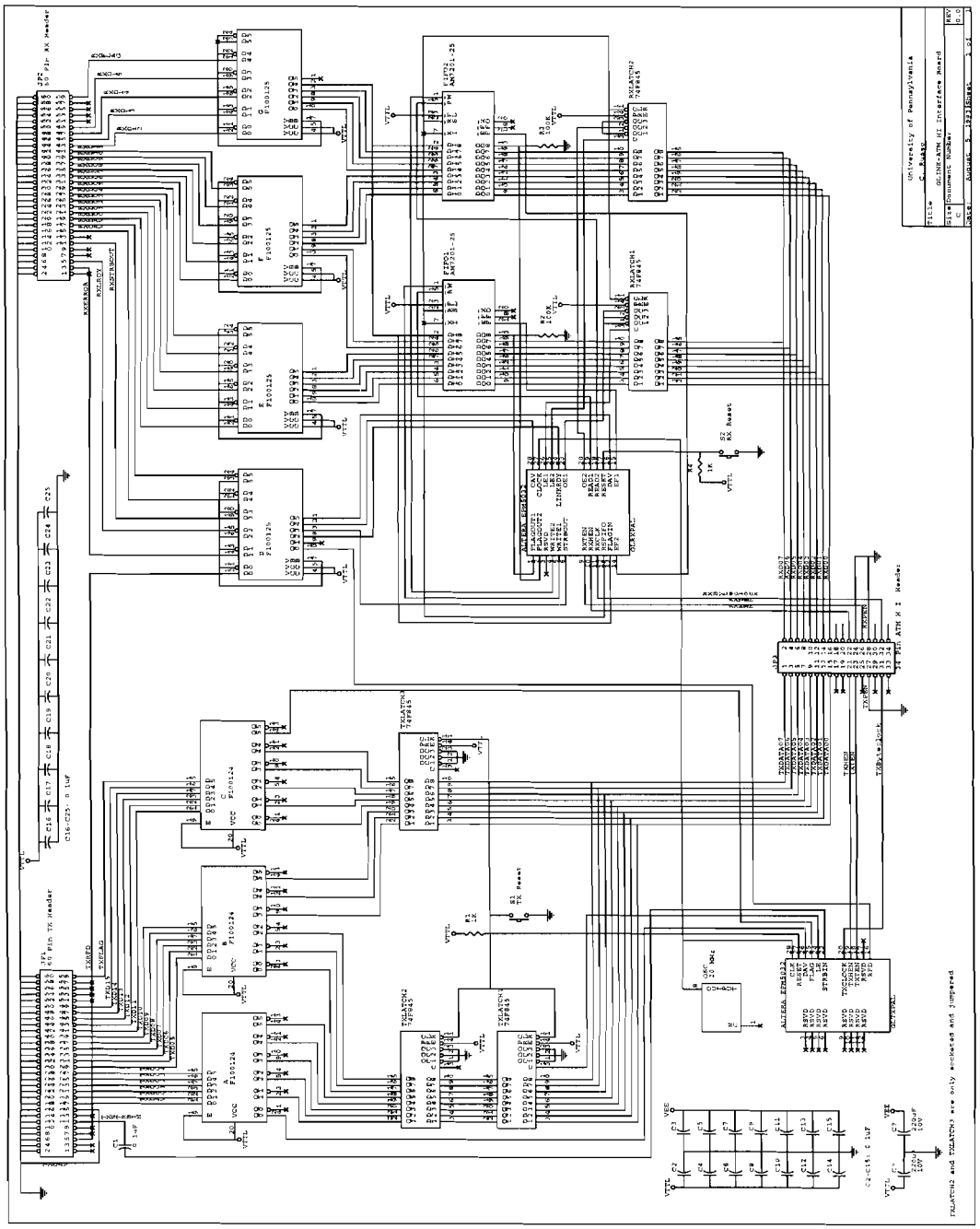


Figure 7

Table 1: Data Rate vs Link Length				
Link Length (ft.)	Maximum STRBIN Frequency (MHz)			
	16 Bit	17 Bit	20 Bit	21 Bit
300	7.07	7.01	5.236	5.128
280	8.20	7.97	6.239	6.046
260	9.474	9.218	6.752	6.498
240	10.976	11.007	8.363	8.385
220	12.73	12.91	9.510	9.479
200	15.93	16.22	10.997	11.278
180	20.86	21.06	14.813	14.61
160	22.12	23.33	17.504	17.553
140	24.62	25.12	20.97	21.54
120	33.89	34.56	23.93	25.09
100	47.98	48.91	36.28	38.91
80	58.11	58.89	45.34	49.22
70	66.18	66.45	56.5	57.76
60	68.8	72.21	59.65	62.67
50	73.8	73.8	62.54	62.16

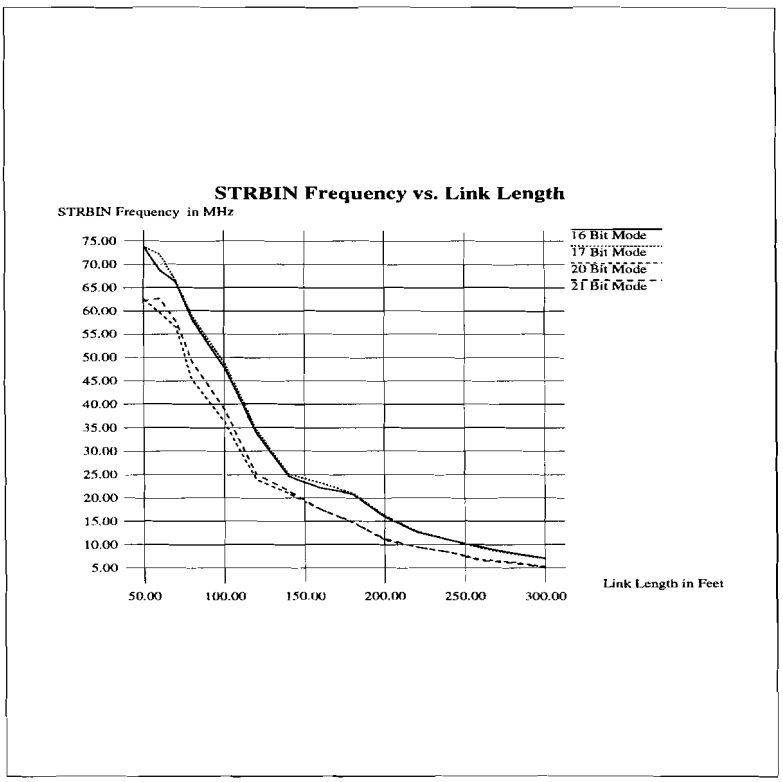


Figure 8

One other notable point discovered during the data rate vs link length testing concerns the setting of the DIV0 and DIV1 inputs to the GLINK chips for operation over the STRBIN frequency range tested. In the GLINK chip set technical data,[2] minimum and maximum clock frequencies are specified for each of the four combinations of DIV0 and DIV1 inputs, and some gaps appear between the cutoff frequencies for each combination of settings. In the actual testing, sharp cutoffs were not evident; in fact, overlapping of the frequency ranges occurred.

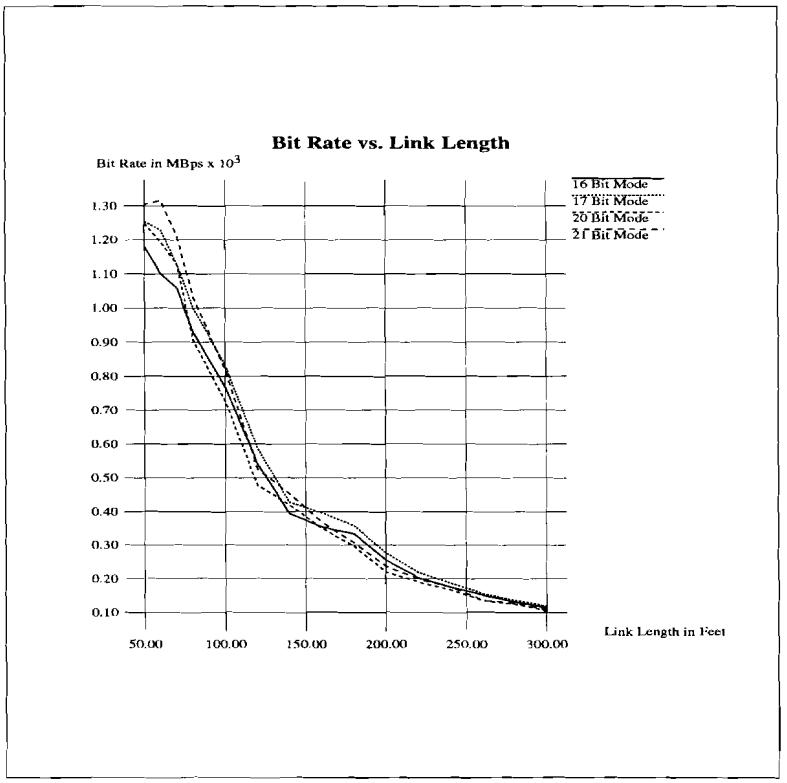


Figure 9

5.2 Glue Board Design Results

The Glue board design exercise described in section 4 of this report led to the construction and testing of the board, and to the transmission of ATM cells via GLINK as described in the ATM/GLINK Application Note/Specification.[6] With the components arranged as shown earlier in Figure 2, a simple counting pattern was sent as the data for the ATM cells with a rate approximately 155 MBps. The system was tested with respect to random resets on both the TX and RX sides, as well as loss of the physical link (disconnecting the coax), and it recovered synchronization every time. The system was tested with a 50 foot coaxial link and it operated fine at that length. No other link length tests were performed with the actual system and data being transferred.

The prototype Glue board and the GLINK Evaluation board had the following power consumption characteristics:

Table 2: Power Consumption Data		
	+5V Supply	-5V Supply
GLINK Board	0.125A	1.06A
Glue Board	0.78A	0.43A
Total	0.905A	1.49A

6. Conclusions

The main conclusion of this work is that the GLINK chip set is a viable means of providing local ATM transport via low cost coaxial cable at a rate of 155 MBps, and can be adapted to work well with the ATM Host Interface equipment previously constructed at Penn designed for a SONET physical layer. For a link of up to 300 feet with no external repeaters and using standard (low cost) RG-58 coaxial cable, this rate can be easily reached and maintained using simple support circuitry like that designed and tested here. With

higher speed components in the logic portion of the Glue board (or a different architecture), rates surpassing 622 MBps can be reached over a 100 foot link.

6.1 Next Steps

There are a few near term modifications and additions to this basic design which would simplify its use, many of which are packaging and power supply related. First, the arrangement in the prototype had the GLINK chips and basic support circuitry on one board, with the Glue board being separate; a cumbersome arrangement at best. Combining the GLINK circuitry and interface/control circuitry onto a single printed circuit board would be more convenient and in the long term, more reliable. Combining the circuitry into a single plug-in board for a workstation would also allow some of the components (like the fire-wall of buffer chips on the GLINK Evaluation board) to be eliminated, thus lowering the power and thermal loads the board would create. The addition of a DC-DC converter and some regulation circuitry to provide the -5V and -2V supplies would eliminate the need for bench supplies to power the circuit, allowing it all to operate from the +5V supply from the computer the board is used in. Wiring the reset signals for the Glue board into a software controlled reset would eliminate the need for manual reset switches.

One aspect of the design described here that was not explored was operation using two separate communicating entities; all testing done here was strictly transmitting and receiving the same signal (looping it back). This type of testing may not have been rigorous enough to bring timing related bugs in the design to the surface. An example which comes to mind is the clock in the RX portion of the Glue board. As tested here, one clock was being used to control the data transmission, and the same clock was being used to operate the RX circuitry. If two Glue boards and two GLINK Evaluation boards were available, a full duplex circuit with two clock sources (ideally of the same frequency) could have been tested to see if differences in clock speeds manifested themselves as overflowing or underflowing FIFOs in the receivers. Were this to be a problem, two solutions come to mind. The more logically complex would be to monitor the FIFOs and the amount of data they are holding and insert overhead into the data path to compensate for any rate differences before an over- or under-run occurs. Alternatively, one could return to the design mentioned previously in which the serial clock at the RX side of the GLINK is divided down to provide the clock for the RX circuitry, a clock which would necessarily be synchronized with the clock on the TX side of the link.

Faster links could be implemented quite simply with respect to the GLINK chip set, but faster logic, control, and storage will be bottlenecks. One possible way to speed things up would be to design the control circuitry using discrete, high-speed ECL components instead of using the EPLDs. While this would alleviate the speed bottleneck at the control level, it would make for complex hardware design and construction. Alternatively, the use of multiple GLINKs together to provide a wider data path could enable faster overall communication without increasing the speed requirements of all the support components, much in the way that in this design, the GLINK can operate at half the speed of the ATM HI boards because its data path is twice as wide. The drawbacks to this solution are that it would require multiple coaxial links (2 for each full duplex GLINK arrangement utilized), and it would be a deviation from the current specifications in the ATM/GLINK Application Note from HP on the subject.[6]

Finally, a more in-depth study of the maximum usable link length may be in order. Since the usable link lengths via optical fiber are much larger than those for coaxial cable,[3] one may assume that the distance limitation is primarily driven by signal attenuation in the coax. A simple, low cost analog amplifier may enable a greater distance range from coaxial cable, still at a lower cost than that for using fiber and its associated optics. A study of rate vs link length characteristics in which bit error rates are measured would give a clearer view of the GLINK chip set's performance, with greater accuracy than that possible with the data gathering methods used here. Rate vs link length testing with more than a single GLINK chip set would lend more statistical weight to the data as well.

7. Acknowledgements

Thanks to Dr. Jonathan Smith and Brendan Traw for all their help. Thanks to Mark Laubach at HP for providing the GLINK Evaluation board used in this work and answering countless questions concerning both the GLINK chip set and the application note on ATM over GLINK.

8. Appendix

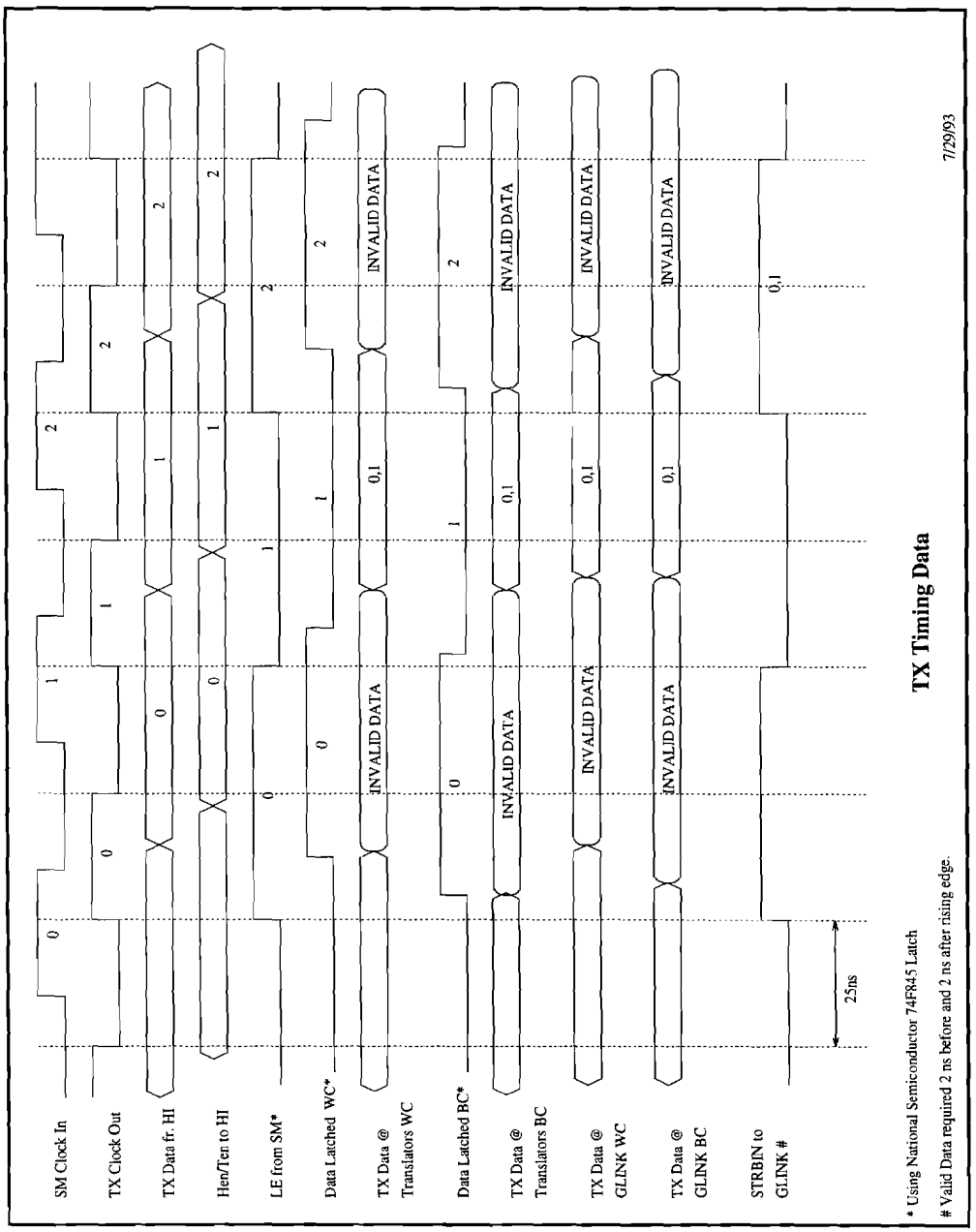


Figure A-1

```
DESIGN IS "txglue"
BEGIN
    DEVICE "txglue" IS "EPM5032-1"
    BEGIN
        clock                       @  28     : INPUT ;
        reset                       @  27     : INPUT ;
        rfd                         @  16     : INPUT ;
        dav                         @  26     : OUTPUT; %LC31%
        flag                        @  25     : OUTPUT; %LC29%
        le                          @  24     : OUTPUT; %LC27%
        strbin                      @  23     : OUTPUT; %LC25%
        txclk                       @  20     : OUTPUT; %LC23%
        txhen                       @  19     : OUTPUT; %LC21%
        txten                       @  18     : OUTPUT; %LC19%
        txsm0                       @ LC32    : BURIED;
        txsm1                       @ LC30    : BURIED;
        txsm2                       @ LC28    : BURIED;
        txsm3                       @ LC26    : BURIED;
        txsm4                       @ LC24    : BURIED;
        txsm5                       @ LC22    : BURIED;
    END;
END;

SUBDESIGN txglue
    (
    rfd,clock,reset:INPUT;
    txclk,txten,txhen,le,flag,dav,strbin:OUTPUT;
    )

VARIABLE
    txsm[5..0], hlf:DFF;

BEGIN
    txsm[].clk=(clock);
    txsm[].clrn=reset;
    hlf.clk=clock;
    hlf.clrn=reset;
    txsm[]=(txsm[]+1 & (txsm[]!=55));
    txsm[]=(0 & (txsm[]==55));
    txten=(txsm[]==0)#(txsm[]==1)#(txsm[]==7);
    txhen=(txsm[]>=2)&(txsm[]<=6);
    flag=(txsm[]==0)#(txsm[]==1);
    dav=!(rfd);
    txclk=(clock);
    hlf=!(hlf);
    strbin=(hlf==0);
    le=(hlf==0);
END;
```

Figure A-2: TX State Machine Code

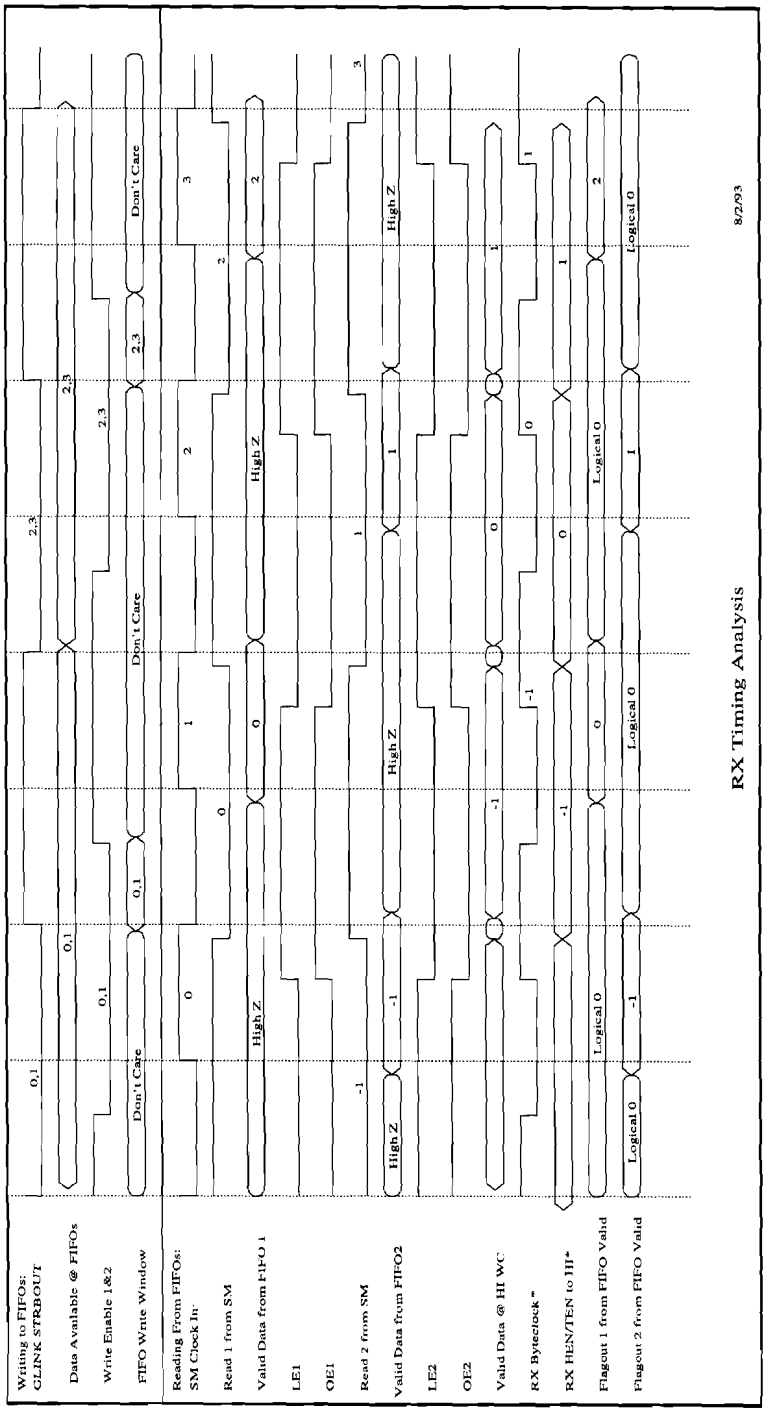

Writing to FIFOs:
GLINK STROBOUT
Data Available @ FIFOs
Write Enable 1&2
FIFO Write Window
Reading From FIFOs:
SM Clock In:
Read 1 from SM
Valid Data from FIFO 1
LE1
OE1
Read 2 from SM
Valid Data from FIFO2
LE2
OE2
Valid Data @ HI WC
RX Byteclock *
RX HEN/TEN to HI*
Flagout 1 from FIFO Valid
Flagout 2 from FIFO Valid
High Z
Logical 0
Don't Care
RX Timing Analysis
8/2/93

Figure A-3

```
DESIGN IS "rxglue"
BEGIN
    DEVICE "rxglue" IS "EPM5032-1"
    BEGIN
        cav                             @  28     : INPUT ;
        clock                           @  27     : INPUT ;
        dav                             @  16     : INPUT ;
        ef1                             @  15     : INPUT ;
        ef2                             @  14     : INPUT ;
        flagin                          @  13     : INPUT ;
        flagout1                        @   1     : INPUT ;
        flagout2                        @   2     : INPUT ;
        linkrdy                         @  24     : INPUT ; %LC27%
        reset                           @  17     : INPUT ; %LC17%
        strbout                         @   6     : INPUT ; %LC7%
        le1                             @  26     : OUTPUT; %LC31%
        le2                             @  25     : OUTPUT; %LC29%
        oe1                             @  23     : OUTPUT; %LC25%
        oe2                             @  20     : OUTPUT; %LC23%
        read1                           @  19     : OUTPUT; %LC21%
        read2                           @  18     : OUTPUT; %LC19%
        rsfifo                          @  12     : OUTPUT; %LC15%
        rxclk                           @  11     : OUTPUT; %LC13%
        rxhen                           @  10     : OUTPUT; %LC11%
        rxten                           @   9     : OUTPUT; %LC9%
        write1                          @   5     : OUTPUT; %LC5%
        write2                          @   4     : OUTPUT; %LC3%
        fstcell0                        @ LC32    : BURIED;
        fstcell1                        @ LC30    : BURIED;
        fstfo                           @ LC28    : BURIED;
        rxsm0                           @ LC26    : BURIED;
        rxsm1                           @ LC24    : BURIED;
        rxsm2                           @ LC22    : BURIED;
        rxsm3                           @ LC20    : BURIED;
        rxsm4                           @ LC18    : BURIED;
        rxsm5                           @ LC16    : BURIED;
    END;
END;

SUBDESIGN rxglue
    (
    reset,clock,strbout,cav,dav,linkrdy,flagin,flagout1,flagout2,ef1,ef2:INPUT;
    % reset, cav, ef1, ef2 are all active low at the PAL %
    write1,write2,read1,read2,rsfifo,le1,le2,oe1,oe2,rxhen,rxten,rxclk:OUTPUT;
    % write1, write2, read1, read2, oe1, oe2, rsfifo are all active low outputs
      from the PAL %
    )

VARIABLE
    rxsm[5..0],fstcell[1..0],hlf,fstfo:DFF;
    inrs:NODE;

BEGIN
    rsfifo=(reset & inrs); % when either go low, things get reset %
    write1=(!((linkrdy) & (dav) & (cav) & !(strbout)) # !(reset & inrs));
    write2=(!((linkrdy) & (dav) & (cav) & !(strbout)) # !(reset & inrs));
    % write and read inputs should be high during reset %

    fstcell[].clrn=(reset & inrs);
    fstcell[].clk=flagin;
    fstcell[]=(fstcell[]+1 & (fstcell[]<2));
    fstcell[]=(2 & (fstcell[]>=2));
    % fstcell keeps the FIFO from being read until two flags have gone by,
      meaning that a full cell is in the FIFO.%


    hlf.clrn=(fstcell[]==2 & inrs);
    hlf.clk=clock;
    hlf=!(hlf);

    read1=((hlf==0 & ef1 & fstcell[]==2)) # !(reset & inrs);
```

Figure A-4: RX State Machine Code

```
    read2=((hlf==1 & ef2)) # !(reset & inrs);
    % write and read inputs should be high during reset %

    fstfo.clrn=(reset & inrs);
    fstfo.clk=!clock;
    fstfo.d=((flagout1) # (fstfo));
    % fstfo holds rxsm in reset until the first flag is read from the FIFO,
      signifying the beginning of the first full cell.%

    rxsm[].clk=clock;
    rxsm[].clrn=(fstfo.q);
    rxsm[]=(rxsm[]+1 & (rxsm[]<55));
    rxsm[]=(0 & (rxsm[]>=55));

    IF (((rxsm[]!=0)&(rxsm[]!=1)&(rxsm[]!=2)&(flagout2)) #
    ((rxsm[]==0) & !(flagout1) & (fstfo))) THEN
        inrs=GND;
    ELSE
        inrs=VCC;
    END IF;

    % The rxsm will be free from reset once the first flag has been read
      from the FIFO. The receiver will reset itself if a flag doesn't occur
      when it should or does occur when it shouldn't. %

    le1=(hlf==1);
    oe1=(hlf==1);
    le2=(hlf==0);
    oe2=(hlf==0);
    rxclk=clock;
    %rxhen=((rxsm[]>=2) & (rxsm[]<=6));
     For some reason the simulator showed a glitch for this line,
     so I replaced it with the following one and the glitch disappeared.%
    rxhen=((rxsm[]==2)#(rxsm[]==3)#(rxsm[]==4)#(rxsm[]==5)#(rxsm[]==6));
    rxten=((rxsm[]==0)#(rxsm[]==1)#(rxsm[]==7)# !(fstfo));
END;
```

Figure A-4: RX State Machine Code

(continued)"

REFERENCES

1. C.B.S. Traw, & J.M. Smith, "Hardware/Software Organization of a High Performance ATM Host Interface," *IEEE Journal on Selected Areas in Communications,* Vol. 11 (2), 1993.

2. *Gigabit Rate Transmit Receive Chip Set Technical Data.* Hewlett-Packard, 1992.

3. M. Leonard, "Silicon Chips Simplify Gigabit Data-Link Design," *Electronic Design,* July 9, 1992.

4. *Reference Guide for the Gigabit Rate Transmit Receive Chip Set (HDMP-1000) Evaluation Board Tentative Data.* Hewlett-Packard, October 1992.

5. C.B.S. Traw, *Physical Layer Interface Specification for Upenn ATM Host Interface.* Draft, March 18, 1993.

6. M. Laubach, *Gigabit Rate Transmit/Receive Chipset ATM Framing Specification Application Note.* Hewlett-Packard, Revision 2.1, March 15, 1993.