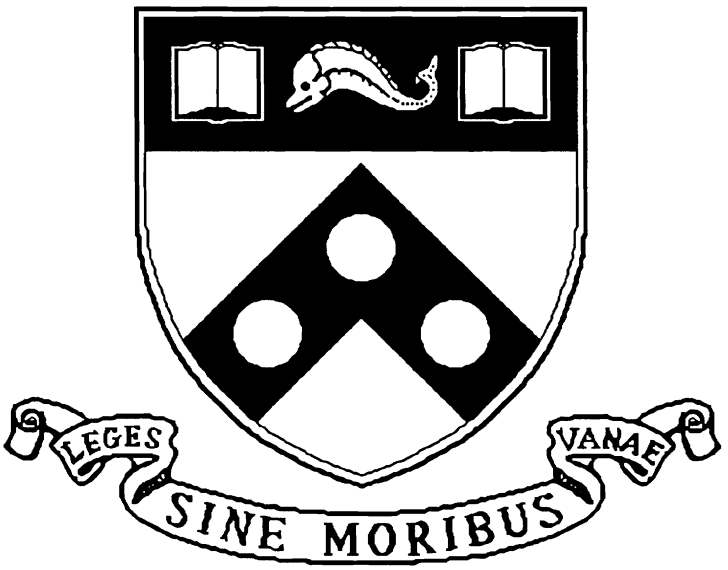
Host Interfacing at a Gigabit

MS-CIS-93-43
DISTRIBUTED SYSTEMS LAB 33

C. Brendan S. Traw



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

April 1993

Host Interfacing at a Gigabit

C. Brendan S. Traw traw@cis.upenn.edu

April 21, 1993

Abstract

A major goal of the host interface architecture which has been developed at UPenn is to be sufficiently flexible as to allow implemention using a range of technologies. These technologies can provide the performance necessary for operation in the emerging high bandwidth ATM networking environments. This paper examines the feasibility of reimplementing the current instantiation of the architecture which operates at 160 Mbps to allow for operation in the 600+ Mbps domain.

1 Introduction

The host interface architecture[6] developed at UPenn allows the interconnection of workstation hosts with high bandwidth networking environments. The basic philosophy for this work has been to develop an architecture which provides hardware support for a "common denominator" of network services, in this case, all per cell operations, in a general enough manner to allow the architecture to be implemented in a variety of technologies. This allows the implementor to select the appropriate technology on a cost/performance/ease-of-implementation basis and also to take advantage of improvements in technology.

The first incarnation of the architecture is as a 160 Mbps network interface for IBM RISC System/6000 workstations. Several different physical layers including SONET STS-3c (155 Mbps) and TAXI (125 Mbps) are supported. 160 Mbps was chosen as the initial target performance since it provided a good match between the bandwidth which the workstation could support, the bandwidth of viable physical layers, and programmable logic which was available in early 1991 when this implementation was begun.

The initial implementation has been prototyped and tested in Model 320 RISC System/6000 workstations. The host interface hardware could easily sustain 160 Mbps. Overall system performance was limited to about 140 Mbps due to an architectural problem with the Model 320's I/O Channel Controller[6]. IBM claims to have eliminated this problem in future models of the RISC System/6000 product line[4]. Many other workstation vendors including Hewlett Packard are beginning to offer machines which should be able to sustain I/O connections with bandwidths between 400 and 600 Mbps. Therefore, exploring the feasibility of updating the implementation of our host interface to match these anticipated performance levels is an interesting exercise.

The remained of this paper is organized in the following manner. Section 2, discusses the current implementation and its performance. Sections 3 and 4 present improvements to the implementation which will allow 600+ Mbps operation. The changes presented in Sections 3 and 4 will be analyzed in Section 5. Finally, Section 6 discusses the prospects for performance improvements not presented in detail in this paper.

2 Current Implementation

The host interface architecture is implemented as two separate components. The segmenter, is responsible for dividing protocol data units (PDUs) and data streams to be transmitted into ATM cell payloads, generating the appropriate cell format and control field, and then mapping the completed cells into the physical layer payload for transport. The second, the reassembler, performs the reverse set of functions necessary to reconstruct the PDUs and data streams from cells received from the network. As implemented, the segmentation and reassembly functions are performed on separate peripheral cards connected to the Micro Channel I/O bus of the RISC System/6000 workstation.

The segmenter and reassembler consist of the logic and memories necessary to implement the ATM cell processing function as well as some additional logic necessary to interface to the IBM RISC System/6000's Micro Channel I/O bus. For the purposes of this discussion, the interface logic will be ignored except when it directly affects the performance of the cell processing hardware.

The current implementation of this host interface architecture is based on the technology of the Altera 5000 series[2] of erasable programmable logic devices (EPLDs). The devices used have densities of 128 and 192 macrocells. A macrocell is a logic array combined with a configurable flip

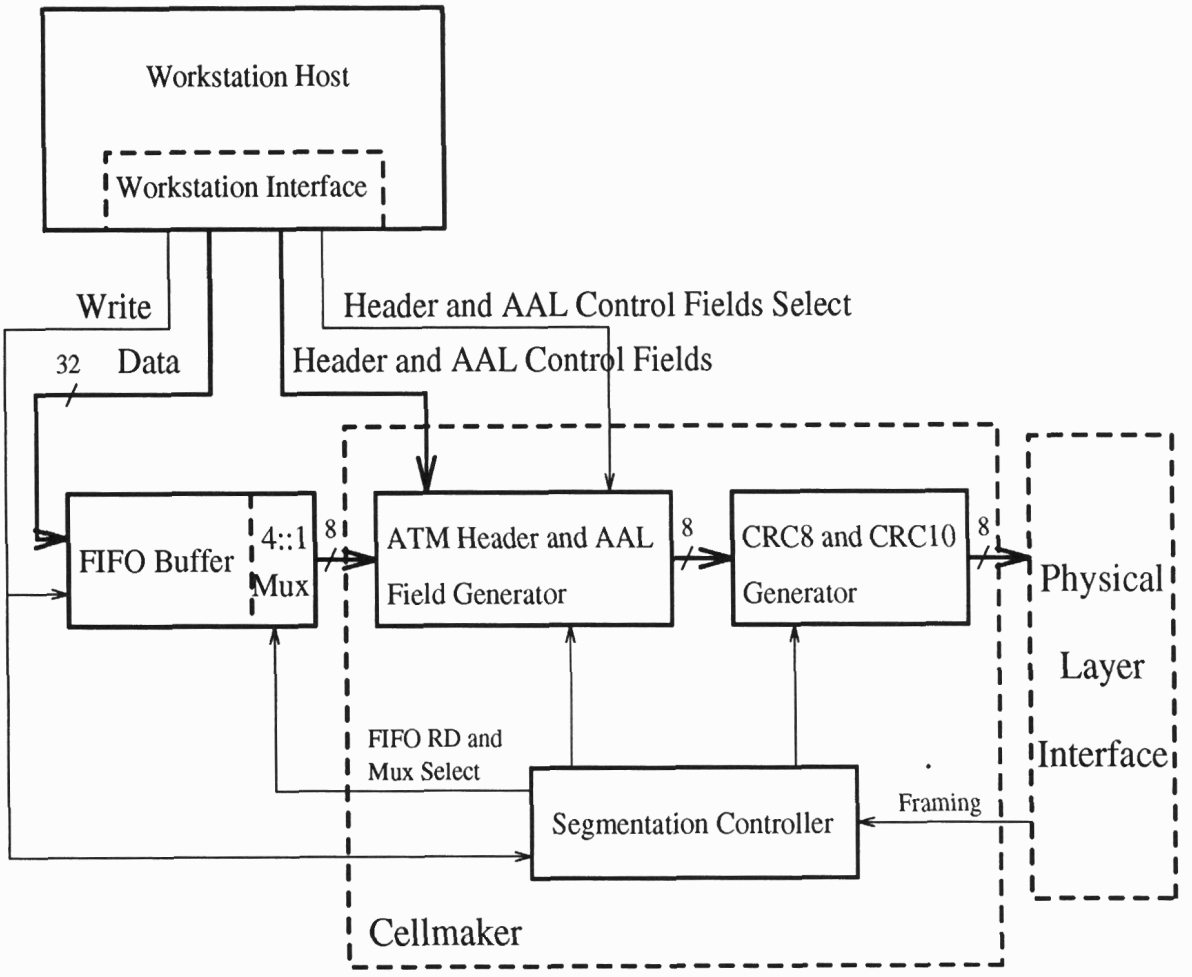


Figure 1: Current Segmenter Implementation

flop. Given the complexity of the logic, the maximum clock frequency was limited to 22 Mhz (see appendix A). A typical operation frequency when connected to an OC-3c physical layer is 19.44 Mhz which is the byte data rate of an OC-3c.

For the performance discussion, the segmenter and reassembler will be examined separately.

2.1 Segmenter Performance

A diagram of the segmenter is presented in Figure 1.

The 32 bit wide FIFO data buffer is implemented as a bank of four 9 bit wide, 25 ns, asynchronous FIFOs. The purpose of this buffer is for holding data prior to segmentation. The 32 bit

output of the buffer is connected to a 4 to 1 mux so that the data can be written to the byte wide ATM Header and AAL Field Generator (FG). As the data passes through the FG, the ATM header and adaptation layer (if appropriate) are added before the partially completed cell is passed to the CRC generator. The CRC generator calculates the CRCs and then places them in the appropriate fields of the ATM header and adaptation layer. Once the CRCs have been added to the cell, the cell is mapped onto the physical layer under the control of the Segmentation Controller. The FG, CRC Generator, and the Segmentation controller are all implemented in a single Altera 5192-1 EPLD called the Cellmaker.

The performance of this segmentation hardware is limited by two factors, other than the pre-determined bandwidth of the workstation bus and that of the physical layer. These are: the width of the cell generation data path and the speed at which it can be clocked. The FIFOs and mux are not bottlenecks until the desired bandwidth exceeds 32*40 Mhz = 1.28 Gbps.

2.2 Reassembler Performance

A diagram of the reassembler is presented in Figure 2.

The reassembler is composed of five major functional elements which operate in parallel to form a cell processing pipeline. Each element of the pipeline is constructed from a single Altera 5128-1 or 5192-1 and with the exception of the Cell Manager and Linked Access Controller, the memory which is associated with it. All of the logic, as in the case of the segmenter can be clocked at up to 22 Mhz. Two different clocks are used. The Cell Manager is synchronized to the clock of the physical layer interface while the rest of the reassembler is clocked at the same frequency, but not necessarily in phase with the physical layer clock. Two clocks were used so that the host workstation can continue operation with the reassembler even if the physical layer clock is temporarily lost.

Table 1 lists the performance of each of the elements in the pipeline. These performance figures are somewhat simplified in that they only represent the worst case per cell processing performed by each element. They do not include the impact of background maintenance and management activities performed by the host on the host interface which would be necessary for actual operation of the reassembler. These activities are ignored since they do not occur as frequently as the arrival of cells and when they do occur, they require few clock cycles to complete. Also, the network

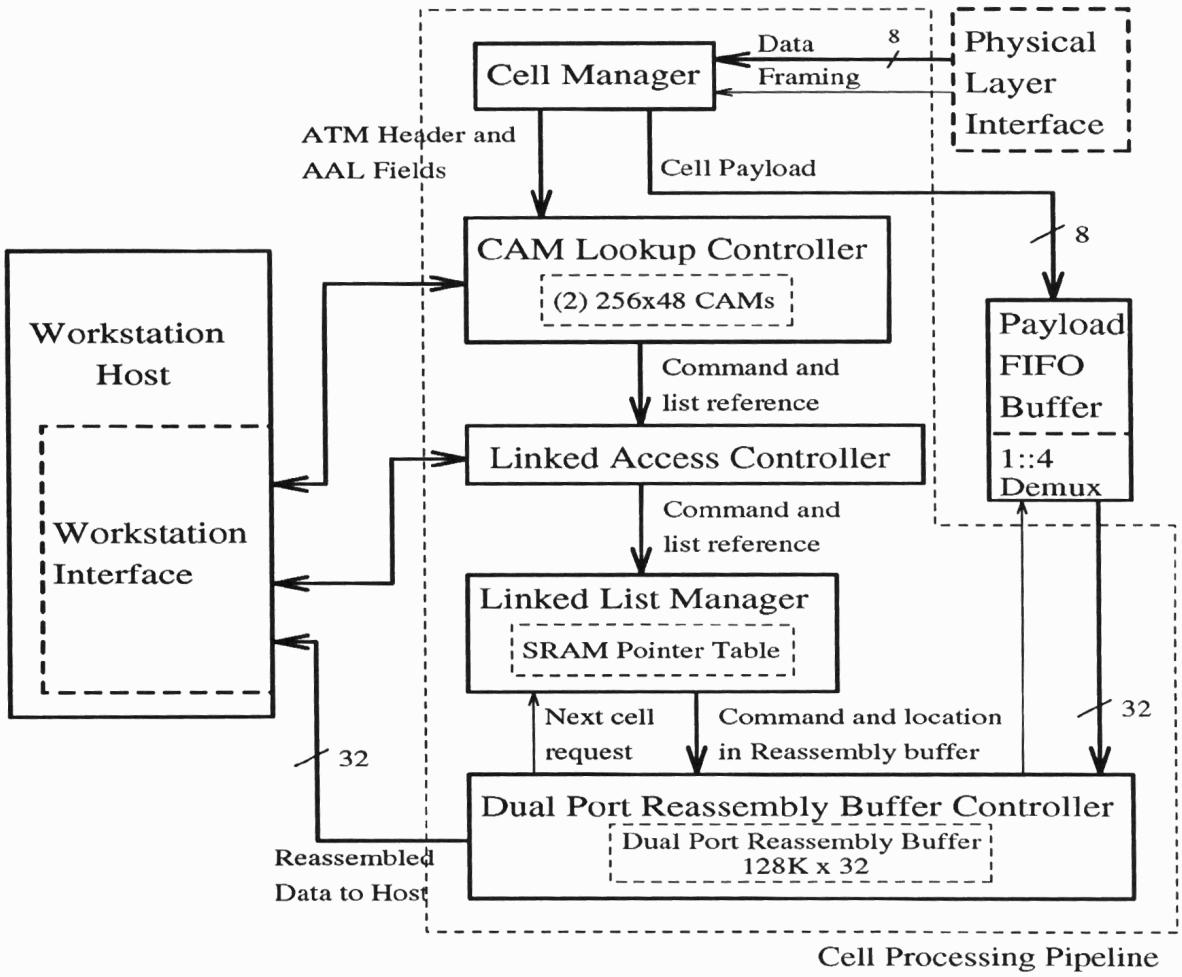


Figure 2: Current Reassembler Implementation

bandwidth figures are just for data. They do not include the additional bandwidth that is needed for physical layer framing.

2.2.1 Cell Manager Performance

In many respects, the Cell Manager is very similar to the Cellmaker, a part of the segmenter. The Cell Manager provides an interface to the physical layer, generates the CRC8 and CRC10 of a cell received from the network for integrity verification purposes, and extracts the values from various ATM header and AAL data fields. Much of the logic for these functions was copied directly from the Cellmaker; thus, the Cell Manager suffers from the same performance limitations as the

Stage	Clock Cycles	Network Bandwidth (Mbps)	Comments
Cell Manager	-	160	Matches Physical Layer Data rate
CAM Lookup Controller	18	470	
Linked Access Controller	6	1413	
Linked List Manager	28	303	
Dual Port Reassembly Buffer Controller	27	314	

Table 1: Performance of Stages in Current Reassembly Pipeline (20 Mhz Clock)

Cellmaker.

2.2.2 CAM Lookup Controller Performance

The CLC provides a means of associating cells received from the network with the appropriate reassembly data structure according to their VCI and MID (if AAL3/4 data). The major factor limiting the performance of the CLC is the performance of the CAMs themselves. In this implementation, Am99C10-70 256 by 48 CAM devices are used. This implementation needs 11 clock cycles to perform the required read, write, and match operations on the CAM. The other clock cycles are needed to pass the results of the CAM search off to the next stage of the pipeline, the LAC, and synchronize the CLC with the CM since they are clocked by potentially out of phase sources.

2.2.3 Linked Access Controller

The LAC arbitrates between the Host and CLC for access to the LLM. It is necessary since there were insufficient I/O resources available on the LLM EPLD. The speed at which the EPLD can be clocked and the method of exchanging data with the LLM limits the performance of the LAC.

2.2.4 Linked List Manager

The Linked List Manager manages the data structure responsible for reassembling and allocating space in the Dual Port Reassembly Buffer. The memory used for the pointer table is 32K by 16 of

20 nS SRAM. The performance of this stage of the reassembly pipeline is particularly critical since two time consuming operations (insert into a list and remove from a list) must be performed for each cell received from the network. The insert is used when the cell is initially received from the network while the remove is used when the cell is to be move to the host workstation's memory. The LLM's performance is currently limited by the speed at which the LLM EPLD can be clocked.

2.2.5 Dual Port Reassembly Buffer Controller Performance

The DPRBC effectively dual ports a 128K by 32 bank of standard 20 ns SRAM, thereby allowing cell bodies to be simultaneously written to and read from the locations specified by the LLM in the Dual Port Reassembly Buffer. Again, the speed of this stage is currently limited by the speed at which the DPRBC EPLD can be clocked.

3 Proposed Segmenter Performance Improvements

To obtain 600+ Mbps performance, a factor of four improvement in bandwidth is needed. This factor of four will be accomplished by two modifications: doubling the clock rate and doubling the width of the data path. Each of these modifications will increase performance by a factor of two.

3.1 Doubling the Clock Rate

To double the clock rate, Altera has provided a solution in the form of their newest line of EPLD technology[3], the 7000 series. The main differences between the 5000 series used for the first implementation and the new 7000 series are that the 7000 series has a greater macrocell density, lower propagation delays, and a more flexible interconnection scheme. Unlike the 5000 series, there is no performance difference between connecting the output a macrocell to the input of another in the same logical array block (LAB) or in a different LAB since all macrocell to macrocell connections must be made through the programmable interconnect array (PIA). For logical structures like those used in the Cellmaker, clock speeds of about 52 Mhz could be supported (see Appendix A). This easily provides a factor of two performance improvement for the Cellmaker's internal logic. Also, because the structure of these two families of EPLDs is so similar, it will not be difficult to transfer the current designs to the 7000 series.

3.2 Doubling the Data Path Width

The main consideration when doubling the width of the data path is whether or not the CRC calculations can be performed 16 bits at a time with the present logic resources. See Appendix B for the expressions necessary to generate both the CRC8 and CRC10 eight and sixteen bits at a time.

Generating the CRCs 16 bits at a time requires at most only an additional 32 and 27 XOR terms for the CRC8 and CRC10 respectively. These additional terms can easily be supplied by the 7000 series EPLD since there is no penalty for spreading the CRC generation logic across several LABs.

4 Proposed Reassembler Performance Improvements

To increase the network bandwidth which can be supported by the Reassembler to 600+ Mbps, a number of improvements must be made to the implementation of the various stages of the cell reassembly pipeline.

4.1 Changes to the Cell Manager

A four fold increase in the performance of the CM is needed to support 600+ Mbps. This can be accomplished using the same techniques of doubling the width of the data path and doubling the clock frequency as were used for the Segmenter's Cellmaker.

4.2 Changes to the CAM Lookup Controller

The overall performance of this stage cannot be improved significantly unless the CAM is replaced with a faster device. Since the current CAM is still sufficient for this the new performance target, this issue will not be addressed. By moving the CLC's logic to a 7000 series EPLD and doubling the clock rate, a slight improvement in performance can be achieved since the read and write cycles for the CAM can be performed more closely to the specifications. Also, the resulting command and list reference could be passed off to the LAC more quickly.

4.3 Changes to the Linked Access Controller

It would be helpful to change this stage from a 5000 to 7000 series EPLD and doubling the clock rate to speed the passing of commands to the LLM.

4.4 Changes to the Linked List Manager

The performance of the LLM can roughly be doubled by changing to the 7000 series EPLDs which allow the clock rate to be doubled. Since the accesstime of the SRAMs cannot easily be reduced enough to match the shortened clock period, loading data from the memory into the LLM would be made slightly more complicated. An external latch could be added between the memory and the LLM to store the data for a clock cycle. Since the latch can deliver data to the LLM much faster than the access time of the SRAM, the data can then be successfully loaded on the next clock cycle. This latch could possibly add one additional clock cycle to each LLM cell operation. Write operations on the SRAM will not be affected by the reduced clock period. Two to three clock cycles can be saved if the command and data passing mechanism between the LLM and other stages can be overlapped with pointer table LLM operations. This was not done on the initial implementation since the extra performance was not needed, but it should not be too difficult to implement.

4.5 Changes to the Dual Port Reassembly Buffer Controller

To increase the bandwidth of the Dual Port Reassembly Buffer by a factor of two, the DPRBC could be implemented in a 7000 series EPLD at double the clock rate. The 20 ns SRAM will be enough to handle the faster clock with no modifications since the DPRBC does not need to touch the data coming to and from the SRAM as the LLM did.

5 Overall Performance after Improvements

5.1 Segmenter

The improvements suggested for the segmenter will provide a factor of four performance improvement. This will easily raise the bandwidth it can sustain to well over 600 Mbps.

Stage	Clock Cycles	Network Bandwidth (Mbps)
Cell Manager	-	640
CAM Lookup Controller	23	737
Linked Access Controller	6	2827
Linked List Manager	26	652
Dual Port Reassembly Buffer Controller	27	628

Table 2: Performance of Stages in Improved Reassembly Pipeline (40 Mhz Clock)

5.2 Reassembler

The performance of the stages in the reassembly pipeline would be affected by the suggested changes according to table 2.

With the changes presented here, overall performance of the reassembler would exceed 600 Mbps.

6 Future Directions

The possibilities for further augmentation of the performance of this architecture are not limited to the methods outlined in this paper. Future improvements in CAM, EPLD, and SRAM technology should allow for continued increases in performance. Also, the application of somewhat more implementationally demanding and capital intensive technologies such as gatearrays or semicustom VLSI could yield additional performance gains. Architecturally, the data paths in the Cellmaker and Cell Manager logic could be widened further. The pointer table could also be widened or interleaved to allow the Linked List Manager to perform multiple pointer operations in a single memory cycle.

All of these performance enhancing options reaffirm that we have fulfilled our original goal[5] of developing a host interface architecture which is sufficiently flexible and powerful to provide a range of performances across many implementation technology choices.

References

[1] Advance Micro Devices, "Am99C10 256 x 48 Content Addressable Memory," 1989.

[2] Altera Corporation, 1992 Data Book.

[3] Altera Corporation, EPM7256 EPLD Data Sheet.

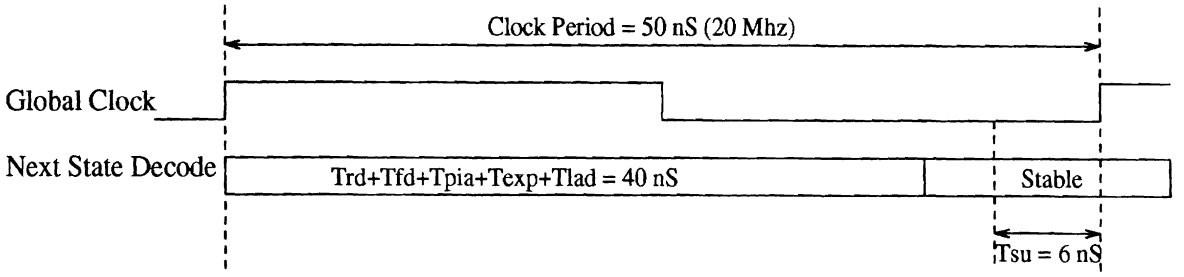
[4] IBM Corporation, IBM RISC System/6000 POWERserver 970 Product Information.

[5] C. B. S. Traw and J. M. Smith, *A High-Performance Host Interface for ATM Networks,* Proceedings ACM SIGCOMM '91, Zurich, September 1991.

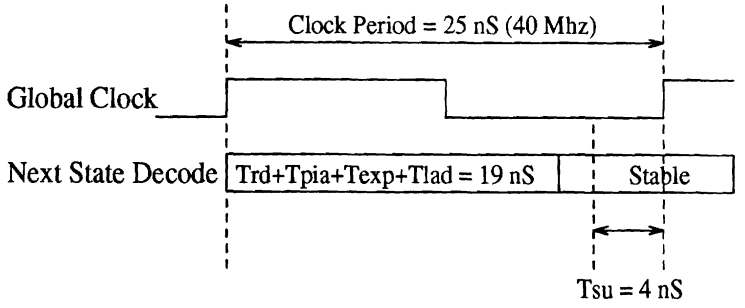
[6] C. B. S. Traw and J. M. Smith, *Hardware/Software Organization of a High-Performance ATM Host Interface,* to appear in IEEE Journal on Selected Areas in Communications, February 1993.

Appendix A: Timing Comparison of Altera 5000 Series and 7000 Series EPLDs

Altera 5000 Series



Altera 7000 Series



Appendix B: Generation of Cyclic Redundancy Checks

```
CRC8 8 bits at a time   Up to 26 XOR are required

c7 = c7$c6$c5$d7
c6 = c6$c5$c4$d6
c5 = c5$c4$c3$d5
c4 = c4$c3$c2$d4
c3 = c7$c3$c2$c1$d3
c2 = c6$c2$c1$c0$d2
c1 = c6$c1$c0$d1
c0 = c7$c6$c0$d0

CRC8 16 bits at a time Up to 58 XOR are required

c7 = c7$c5$c3$d15$d14$d13$d7
c6 = c6$c4$c2$d14$d13$d12$d6
c5 = c7$c5$c3$c1$d13$d12$d11$d5
c4 = c7$c6$c4$c2$c0$d12$d11$d10$d4
c3 = c6$c5$c2$c1$d15$d11$d10$d9$d3
c2 = c7$c5$c4$c2$c0$d14$d10$d9$d8$d2
c1 = c7$c6$c5$c4$c1$d14$d9$d8$d1
c0 = c6$c4$c0$d15$d14$d8$d0

CRC10 8 bits at a time Up to 30 XOR are required

c9 = c5$c4$c3$c2$c1
c8 = c9$c5$c0
c7 = c9$c8$c4$d7
c6 = c8$c7$c3$d6
c5 = c7$c6$c2$d5
c4 = c6$c4$c3$c2$d4
c3 = c9$c4$d3
c2 = c8$c3$d2
c1 = c7$c2$d1
```

```
c0 = c6$c5$c4$c3$c2$d0
```

CRC10 16 bits at a time Up to 57 XOR are required

```
c9 = c9$c8$c2$d13$d12$d11$d10$d9
c8 = c7$c2$d13$d8
c7 = c9$c6$c1$c0$d12$d7
c6 = c9$c8$c5$c0$d15$d11$d6
c5 = c9$c8$c7$c4$d15$d14$d10$d5
c4 = c9$c7$c6$c3$c2$d14$d12$d11$d10$d4
c3 = c6$c5$c1$d12$d3
c2 = c5$c4$c0$d11$d2
c1 = c9$c4$c3$d15$d10$d1
c0 = c9$c3$d14$d13$d12$d11$d10$d0
```

The author appreciates Bruce Davie's assistance in generating these equations.