Description of the i-out-of-m controller - Design 1 implemented in Xilinx 3195 FPGA

Hugo A. Uriona, Srini Seetharam and Vinai Sirkay

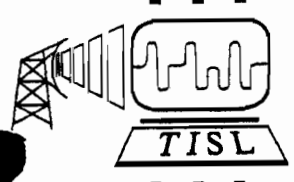
TISL Technical Report TISL-9770-26

Prepared for:

Defense Advanced Research Projects Agency/CSTO

Research on Gigabit Gateways
AARPA Order No. 8634
Issued by EDS/AVS under Contract #F19628-92-C-0080

1994



Telecommunications and Information Sciences Laboratory
The University of Kansas Center for Research, Inc.
2291 Irving Hill Drive Lawrence, Kansas 66045

The i-out-of-m flow control mechanism is a sliding window based rate control algorithm used to control the average cell transmission rates of multiple VCs. In an ATM network i-out-of-m pacing allows a single VC to transmit up to i cells in a window of size m. This paper presents a hardware implementation of the i-out-of-m flow control using a single FPGA as the rate controller. This implementation can support throughputs of up to 622 Mb/s.

1 Introduction

The i-out-of-m flow control mechanism is a sliding-window based rate control algorithm used to control the average throughput of multiple virtual circuits (VCs) [2]. This algorithm is suitable for pacing ATM traffic carried over SONET [3, 6]. In this case, the i-out-of-m algorithm allows a source to transmit up to i cells in any m consecutive cell slots. Different VC may have different values of i and m to provide more flexibility in bandwidth allocation. The average bandwidth allocated to a VC is determined by the ratio i/m and the capacity of the link. By adjusting the i and m values for the VCs the total bandwidth available can be dynamically allocated. Large values of i and a small i/m ratio allow bursty traffic, while a large i/m ratio, approaching one, results in a smooth flow of cells.

The bandwidth policer described in this paper has been implemented on a single FPGA and several memory chips. In particular, the information needed for scheduling and pacing the VCs is stored in twelve SRAM chips organized in three memory structures: a Pacing Table, a Schedule Pointer Table and a Schedule List. This implementation is able to support throughput up to 622 Mb/s while running at 25 MHz. It is currently being evaluated in the MAGIC gigabit testbed [4, 5].

An overview of the i-out-of-m control mechanism and a detailed description of the memory structures are presented in Section 2. The operation of the implementation is explained is Section 3 and a summary of the physical implementation is included in Section 4.

2 Architecture

Initially a VC is given an *account* initialized to i credits. This account is decremented by one after each cell transmission on that VC. At the same time, a new credit is scheduled to be incremented to the same account m cell slots later, where m is the window size. If the account of the VC reaches zero, the corresponding VC is stopped and it is not allowed to transmit more cells until its account has a positive balance.

Due to hardware implementation constraints, only one VC can be served per cell slot, hence there is a queue of VCs per scheduled cell slot. Figure 1 shows a block diagram of the i-out-of-m control mechanism. After a cell is transmitted on a VC, the i/m Controller reads the corresponding i and m values from the pacing table. The i value is decremented by one and stored back into the Pacing Table. The m value is used to schedule a credit update by inserting the outgoing VC at the tail of the queue of VCs which are scheduled at the same time slot. The queued VCs form the Schedule List. The Controller updates the accounts of all the VCs in a given queue, which is scheduled for cell slot k, before proceeding to update the credits of the VCs in the next queue, which are scheduled for cell slot $k+1$.

The memory structures managed by the Controller and their functions are described below.

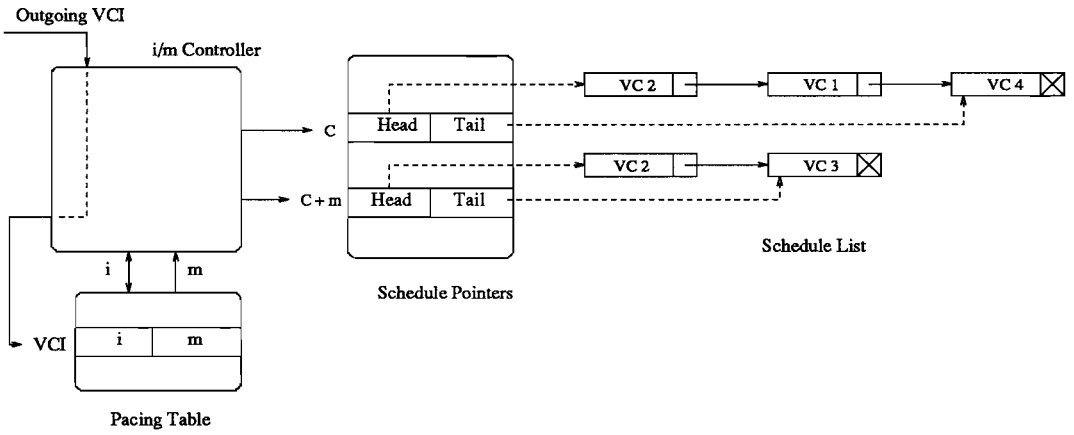


Figure 1: i/m Block Diagram

2.1 Pacing Table

The Pacing Table contains the information needed to control the bandwidth of the VC. This table is indexed by VC identifiers (VCIs). For each VCI the table contains an i value, which is the number of credits in the VC account, an m value, which is the window size in cell slots, and a flag to indicate if the VC is bandwidth controlled or not. The number of bits allocated for the i and m values are eleven and twelve, respectively. The current size of the Pacing Table is 4k entries.

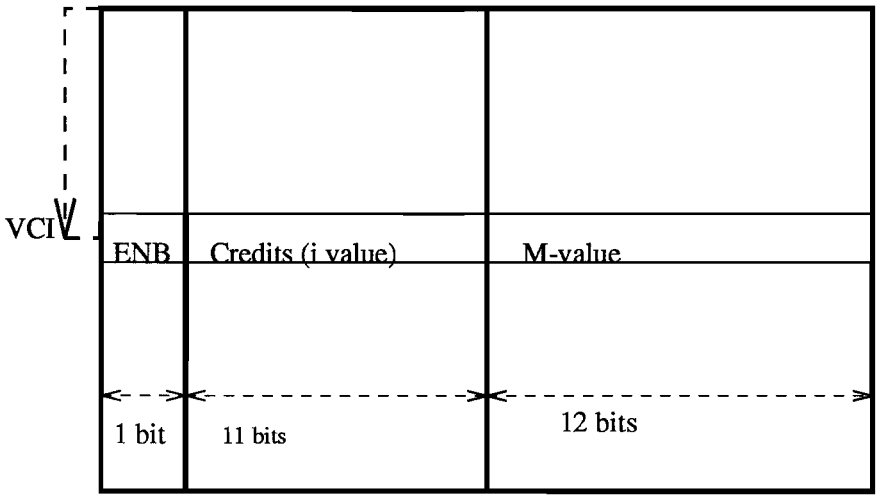


Figure 2: Pacing Table

2.2 Schedule Pointers Table

This memory keeps track of the head and tail pointers of the queues of VCs that are scheduled for credit update at different cell slots. The entries in this memory, a Head pointer and a Tail pointer, are used as indices to the Schedule List. Sixteen bits are allocated for each of the head and tail pointers of each entry. The most significant bit of each pointer is a nil indicator. The current implementation supports 4096 entries which is the maximum window size (in cell slots) allowed.

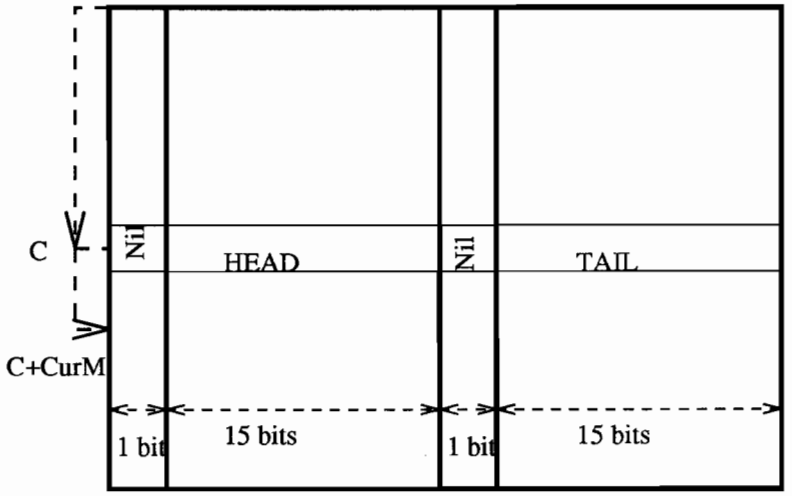


Figure 3: Schedule Pointers Table

2.3 Schedule List

The VC queues are organized in this memory in linked lists. Each element in the list contains the following: a VCI value, whose credit needs to be updated; a pointer to the next element in the queue; and a nil flag to indicate the end of the list. The VCI field is twelve bits long and the next pointer field is sixteen bits long. The current size of the Schedule List is 16k entries.

2.4 i-out-of-m Controller

The three memory structures described above are managed by the i-out-of-m Controller which is implemented in an FPGA. This controller is responsible for:

- Decrementing the credits of a VC after a cell transmission,
- Scheduling the deposit of a new credit on the VC account at a later time

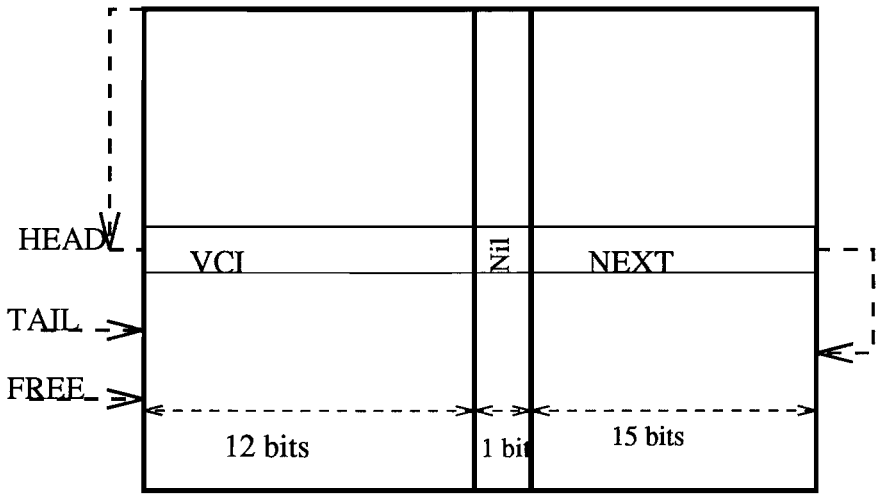


Figure 4: Schedule List

- Servicing each VC queue to increment the credits which have been previously scheduled.
- Maintaining the VC queues in the Schedule List and the queue pointers in the Schedule Pointers Table.
- Issuing Stop and Start signals according to a VC credit value.

3 Principles of Operation

There are three processes running simultaneously in the Controller:

- Process 1: Decrementing credits.
- Process 2: Adding VCs to the Schedule List.
- Process 3: Scheduling and performing credit updates.

While Process 1 operates on the VCI of the cell transmitted in slot *k*, Process 2 uses the VCI of the cell transmitted in slot *k-1*.

3.1 Decrementing Credits

When a cell is transmitted on a particular VC, the i-out-of-m Controller uses the VCI to read the *i* and *m* values off the Pacing Table. Then the *i* value is decremented by one. If the new credit value is less than a threshold value, the Controller sends a *Stop* signal back to the source to inform that it should not transmit any more cells. The Controller writes the new credit value back to the Pacing Table.

3.2 Adding a VC to Schedule List

The VC on which a cell is transmitted must be added to the Schedule List. The m value is added to the value of an internal circular counter to obtain an address into the Schedule Pointer Table. The circular counter is used as a slot index which indicates the cell slot being serviced. The Tail Pointer read off the Schedule Pointer table is used to add the VC at the end of the queue in the Schedule List.

3.3 Scheduling and Credit Update

This process updates the Schedule List and increments the credits of the VCs scheduled for the cell slot indicated by the circular counter. The value of this counter is the address of the Head pointer to the VC queue to be serviced. Since only one VC can be serviced per cell slot, the Controller increments by one the credits of the VC at the head of the queue and then removes the VC from the queue. The second VC in the queue is serviced in the next cell slot, and so on, until the queue is empty. When one queue is emptied, the Controller starts updating the credits of the VCs in the next queue.

4 FPGA Implementation

The i-out-of-m Controller has been implemented in a Xilinx XC3195 FPGA. A layout of the CLBs (Configurable Logic Blocks) and the internal connections are shown in Figure 5. This implementation uses 166 out of 484 CLBs and 93 out of 176 I/O pins. The interface to the Controller includes: one 16-bit I/O data bus and a 12-bit address bus for the Pacing Table; one 16-bit I/O data bus and a 16-bit address bus shared between the Schedule List and Schedule Pointers Table; and control signals for all the memory structures. The Controller runs at 25 MHz with a cell slot duration of 520 ns. The three memory structures are stored in twelve SRAM chips: four 4k $\times$ 8 SRAMs for the Pacing Table, four 4k $\times$ 8 SRAMs for the Schedule Pointers Table, and four 16k $\times$ 8 SRAMs for the Schedule List. The maximum throughput supported by the Controller is 622 Mbps.

5 Conclusion

This paper has described an i-out-of-m pacing implementation developed for the MAGIC gigabit testbed. This implementation supports traffic at rates of up to 622 Mb/s (OC-12 or OC-12c) and requires only a single Xilinx FPGA and a few memory support chips.

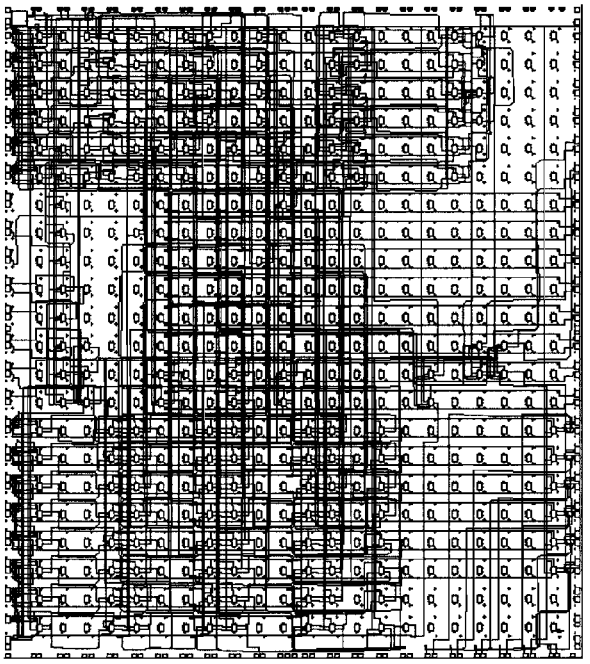


Figure 5: Layout of the i/m Controller on XC3195 FPGA

References

[1] A. Murat Bog. I-out-of-m performance. personal communication, University of Kansas, 1993.

[2] ATM Forum. ATM User-Network Interface Specification, Version 2.0. ATM Forum, June 1992.

[3] 1990 CCITT Study Group XVIII Recommendation I.150. B-ISDN Asynchronous Transfer Mode Functional Characteristics. CCITT, Geneva, 1990.

[4] G. Minden, J. Evans, D. Petr, and V. Frost. An ATM WAN/LAN gateway architecture. In *2nd IEEE Symp. High Perf. Dist. Comp.*, July 1993.

[5] I. Richer. The magic project. In *Proc. Fourth Gigabit Testbed Workshop*, Reston, Virginia, June 1993.

[6] Bellcore Technical Reference TR-NWT-000253. Synchronous Optical Network (SONET) Transport Systems: Common Generic Criteria. Bellcore Technical Reference Issue 2, Bellcore, Dec 1991.