# Solution to concurrent access of data in *i* and *m* tables by the Line Card Processor and i-out-of-m controller

Vinai Sirkay, Douglas Niehaus and Brian Buchanan

Department of Electical Engineering and Computer Science
Telecommunication & Information Science Laboratory
University of Kansas
2291 Irving Hill Road
Lawrence KS 66045-2228
(913) 864-4833

TISL -Technical Report 9770-25

August 1994

TISL

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

## Abstract

A solution to the concurrent access of information tables by three hardware components involved in the dynamic bandwidth allocation and control algorithm is proposed. This solution involves a modification of design 1 for the i-out-of-m controller and provides a hardware-software solution.

# 1 Introduction

During the software development of the dynamic bandwidth allocation (DBA) scheme, a classic concurrent programming problem was discovered. This was with respect to the concurrent access to the I out of M tables by three different components of the DBA calculation:

1. the DBA process calculating new i and m values
2. the flow control (FC) hardware
3. the credit return (CR) hardware

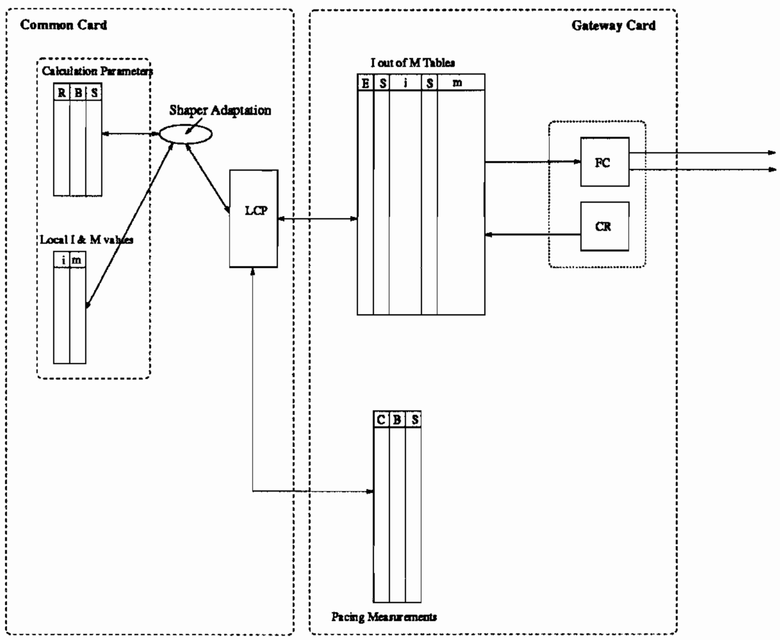


Figure 1: DBA System

# 2 Problem Statement

Once the end of a measurement interval is reached, we use the pacing measurement to determine new I and M values. Once the new I and M values are computed, we compute the difference between the new and old I and M values [1]. These differences specify the number of credits that need to be added or taken away from the current credit balance, maintained in the I field of the I/M hardware [2]. The LCP then engages in a Read-Modify-Write cycle on the data in the I out of M tables in order to update them.

This leads us to 2 scenarios:

### 2.0.1 Scenerio 1 : Addition of Credits

Step 1 : If the new-I greater than old-I then we need to add credits.
Step 2 : LCP reads the I tables to determine number of unused credits and then computes the new number to write in

In this case the number of credits could change after the LCP performs its read and before it can write it back to the I and M tables. This can happen if

1. A credit is scheduled to return after the Read portion but before the Write portion of the Read-Modify-Write cycle performed by the LCP to write the new I value

   or

2. If a cell goes by associated with the VCI being updated during the LCP's Read-Modify-Write cycle.

In this case an incorrect I value would be written, since a credit return or credit decrement would have been ignored. This error is one in which we will lose/gain credits for ever. No correction is possible, since there is no way to determine the outstanding credit balance.

Figure refaddcr shows an example of the problem of adding credits. In this case, the measurement interval ends at time t+4 and a read of the $I$ value occurs at time t+5. The credit remaining at this time is 6. Now the DBA process on the LCP calculates and new $I$ and $M$ values at time t+8. The new $I$ value is computed to be 20 giving the VC 10 extra credits to use. At this time the credits remaining are 4 (i.e. two credits have been used up in the time taken for the DBA calcualations). Based on the value read at t+6 the new credit bank entry for the VC is computed to be 16 and the value 16 is written at t+9. This give the VC two extra credit than it should actually have.

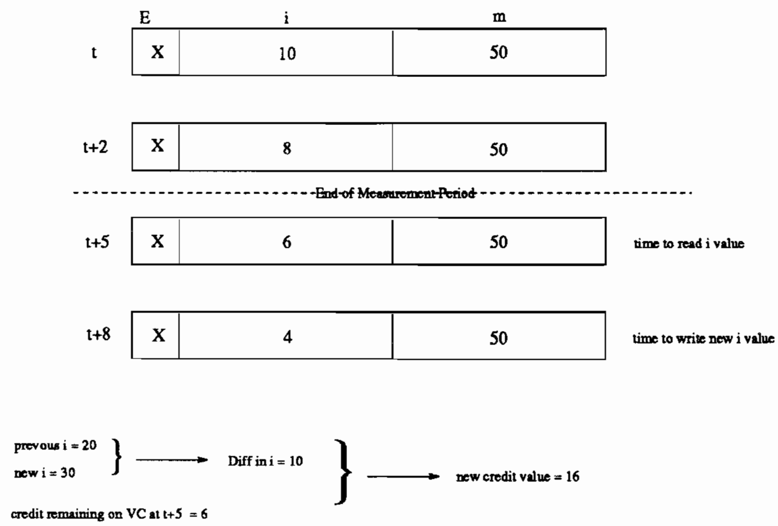


Figure 2: Scenerio 1 : Addition of Credits

### 2.0.2 Scenerio 2 : Subtraction of Credits

Step 1 : If the new-I < old-I then we need to subtract credits

Step 2 : LCP reads the I tables to determine number of unused credits and then computes the new number to write in

In this case the number of credits could change while the LCP performs its Read-Modify-Write cycle on the VC's entry in the I and M tables. This can happen if

1. A credit is scheduled to return during the time taken by the LCP to perform the Read-Modify-Write cycle or

2. If a cell goes by on that VCI during the LCP's Read-Modify-Write time.

In this case a wrong I value would be written, since a credit return or credit decrement would have been ignored. This error is one in which we will lose/gain credits for ever.

In addition, since I is an unsigned 11 bit value, we cannot write a negative number to I which means that we cannot take away credits that are scheduled to return at a later time.

Figure 3 shows an example of the problem of subtraction of credits. In this case, the measurement interval ends at time t+4 and a read of the *I* value occurs at time t+5. The credit remaining at this time is 6. Now the DBA process on the LCP calculates and new *I* and *M* values at time t+8. The new *I* value is computed to be 10 giving the VC 10 less credits to use. At this time the credits remaining are 4 (i.e. two credits have been used up in the time taken for the DBA calcualations). Based on the value read at t+6 the new credit bank entry for the VC is computed to be -4 and the value -4 is written at t+9. This give the VC two less credit than it should actually have. Note that the addition of a sign bit to the I value reduces its effective range by half.

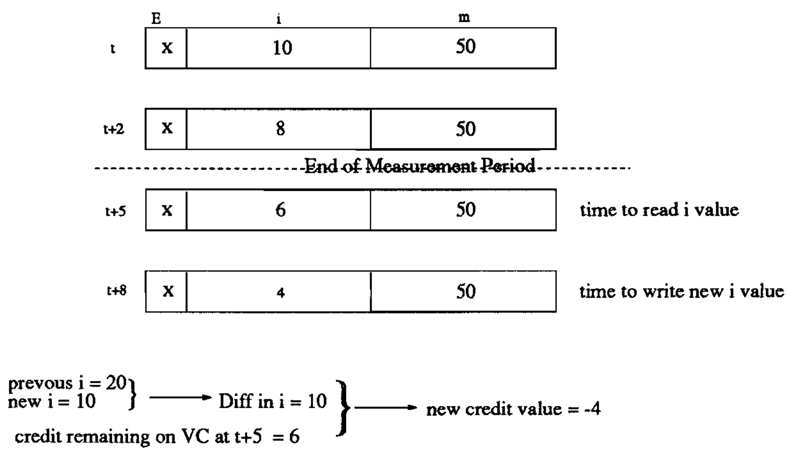


Figure 3: Scenerio 2 : Subtraction of Credits

## 2.1 Solutions

The solution to this problem was designed to have minimal hardware changes, and was selected after carefully considering a wide range of solution strategies.

One important aspect of the solution is that the credit decrements performed by the FC hardware as ATM cells go by will be stopped during the Read-Modify-Write cycle by stopping the VC. This ensures that no cells come across the cross bar for that VC and therefore no decrement will occur since the Flow Control Hardware will never see a cell on the VC in question.

The credit increment will also be disabled during the Read-Modify-Write cycle, to avoid losing a returned credit. The method for stopping credit return requires adding a semaphore bit. However, since we also effectively halved the I value by adding a sign bit, the original ratio is, at least, preserved. The semaphore bit is used to indicated to the CR hardware that the VC is being updated and that the clock needs to be frozen until the semaphore bit is unset [3]x.

The algorithm used by the DBA process to implement a safe the READ-MODIFY-WRITE cycle on the shared data is:

1. spl_high (raise DBA process priority so that no other ELX process can interrupt)

2. Stop VC using the an_ec_t lcs_stop_invci (an_port_t inport, an_invci_t invci, an_outvci_t outvci);

3. Set the semaphore bit in the M tables by writing to the M tables

4. Busy wait for 2 major Gateway time cycle to allow the values in the I and M to settle down

5. DO THE READ-MODIFY-WRITE

6. Upon completion of the RMW (4) cycle, the semaphore bit is unset in the M table entry for the VC

7. According to the number that we entered into the I tables, the LCP needs to decide if the VC needs to be restarted or should continue to be stopped

   (a) If the VC need to started up again using the an_ec_t lcs_start_invci (an_port_t inport, an_invci_t invci, an_outvci_t outvci);

8. spl_x (return the DBA process to the original priority level)

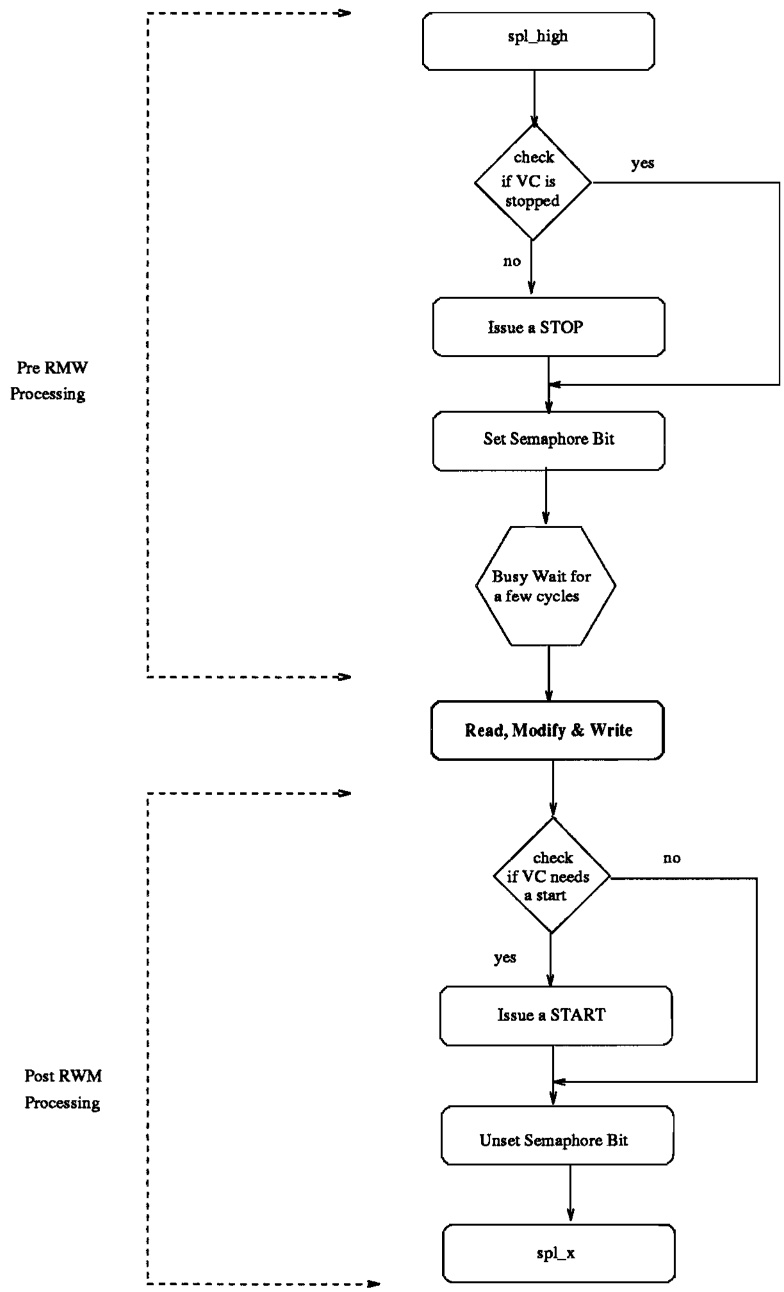


Figure 4: DBA Solution

## References


[1] C. Braun. Performance evaluation of dynamic and static bandwidth management methods for ATM networks. Master's thesis, University of Kansas, August 1994.

[2] Srini Seetharam and Hugo Uriona. Desciption of the i-out-of-m controller - Design 1 implemented in Xilinx 3195 FPGA. TISL Technical Report TISL-9770-26, Telecommunication and Information Science Laboratory, University of Kansas, July 1994.

[3] V. Sirkay. Real-time implementation of an adaptive shaper for dynamic bandwidth allocation in atm networks. Master's thesis, University of Kansas, December 1994.