FLOW CONTROL DESIGN FOR A FLEXIBLE AND ADAPTIVE ROUTER IN PARALLEL SYSTEMS

A. De Luca[1] & A. Jiménez[2,3]

[1] CINVESTAV-IPN. Electric Engineering. Computation Section. e-mail: dlap@delta.cs.cinvestav.mx

[2] Universidad Autónoma Metropolitana. Electronic Department. e-mail: ajf@correo.azc.uam.mx

[3] ESIME-IPN. Electric Engineering Department

Received: January 13th, 2003. Accepted April 2th, 2003

ABSTRACT

The present article contains a high performance buffer design, useful for message flow control in parallel systems. The appropriate handling of the buffers is an important activity for the flow control function. The proposed buffer, which we have denominated as Self-Compacting Buffer (SCB), reduces the communication latency through a highly efficient management of their space, and a hardware implementation. The SCB was designed with a parallel-distributed control unit, using a control cell for each storage locality. Their capacity is expandable: it can grow width and length, preserving the complexity of its control cells. The SCB allows simultaneous writing and reading operations in a single clock cycle.

RESUMEN

El presente artículo contiene el diseño de un buffer de alto desempeño, útil para el control de flujo de mensajes en sistemas paralelos. El manejo adecuado de buffers es una actividad importante de la función de control de flujo. El buffer propuesto, el cual hemos denominado como SCB (Buffer AutoCompactante), permite reducir la latencia de comunicación mediante una administración altamente eficiente de su espacio y una implantación de hardware. El SCB fue diseñado con una unidad de control distribuida paralela, utilizando una celda de control para cada localidad de almacenamiento. Su capacidad es expandible: puede crecer en ancho y largo, manteniendo constante la complejidad de sus celdas de control. El SCB permite la realización simultánea de operaciones de escritura y lectura en un solo ciclo de reloj.

KEYWORDS: Routers, Flow Control, Parallel systems, Self-compacting buffer, VHDL.

1. INTRODUCTION

Parallel processing is a future systems technological key. The main challenge of this systems type, searching for a bigger performance, is the latency time reduction in data communication, which represents an important "bottle neck" now [1-4].

From the router performance point of view in parallel systems, there are two important parameters [5]: routing delay and flow control latency. The first one refers to the lapsed time from the moment when a message arrives to router until the output port is determined by which it will be send. The second corresponds to the frequency that this message can be sent by the router. Once the output port is assigned, the message is stored in a buffer associated to arrival port, so that later on it is directed to the corresponding output port [6,7]. The handling of this information is denominated *flow control*. At this moment for several reasons, the flow control task is slower than the control route one [8,9].

The goal of this work consists on designing a quick structure that carries out the flow control function to reduce at the minimum possible the difference between flow control time and route control time.

We propose a design based on a basic reading and writing operations and their variants model, in which their main properties were characterized. The design was realized in a primitive operations set and it was proven that it matchs the proposed design. Later on, the primitive operations set is represented in a table and, using this table, the design is implemented in logical circuits. Finally, with the help of a logical simulation tool, it is verified that the conditions table fulfill the representation of the model, and so the design is justified.

The circuit was designed totally in hardware fashion, to increase the performance [10,11]. Also a parallel-distributed control is used, with a control cell for each buffer locality. The buffer capacity is expandable: it can grow in width and length, maintaining unaffected its control cells complexity. The simultaneous reading and writing operations are executed in a single clock cycle.

2. SCB DEVELOPMENT

The flow controller proposed in this project is based in the DAMQ flow controller outlined by Tamir and Frazier [12].

The main objective of this implementation is to obtain a high performance buffer with a minimum writing and reading cycle, using efficiently its storage capacity. The proposed buffer avoids empty storage spaces between the data, and its circuit is designed to reduce access times as much as possible. This device is denominated Self-Compacting Buffer, because it maintains the busy spaces adjacent permanently.

2.1 The SCB Operation Principle

The SCB is a FIFO type data multi-array where each array is independent and of variable size. It is possible to complete reading and writing operations simultaneously, and the space occupied in the SCB remains always compacted.

The arrays in the buffer are segments whose size varies during the writing and reading operations. The SCB advantage is that their segments always stay adjacent and the following empty buffer space is available to allow increasing the size of any segment. In the SCB is possible to execute writing and reading operations at the same time, even in the same buffer locality. The segments are located inside the buffer in growing order of their address. The data written in each segment are treated like insertions. An insertion causes the displacement of the subsequent data. The reading operations are treated as extractions, that generate a movement in inverse direction. In this way, the SCB stays compacted. In Fig. 1, an example of a SCB is shown with four contiguous segments of different sizes and an empty space. In Fig. 2(a), the same SCB is shown after having written a data in segment 2. The arrow in the figure indicates that all data that are after the writing localities are displaced downwards, leaving an empty space to insert the new data. Fig. 2(b) shows the SCB after having executed a writing operation or insertion in the segment 2 and two sequential reading operations or extractions in segment 3. In this figure, the arrow indicates that all the data located after the reading address are displaced upwards one position in every reading operation, at the same time that the datum is read during the operation. With this final shift, we avoid leaving the space created during the extraction, and in this way we achieve a compact array.

A high efficiency is reached by the self-compacting technique described previously. The self-compacting is achieved using a distributed control circuit that decides simultaneously in each buffer cell, the actions that must be taken during a reading and/or writing.

The high performance is obtained using hardware in the implementation of the distributed control algorithm that manages the buffers.

In order to manage the multi-segmented buffer, a novel addressing model is used that makes the SCB scalable with a constant addresses bus width.

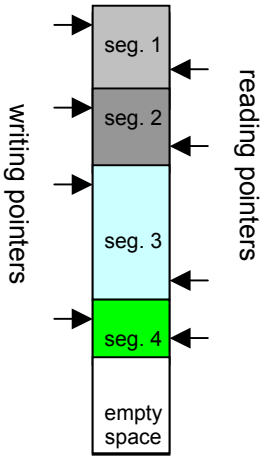

Figure 1. Self-Compacting Buffer with four segments

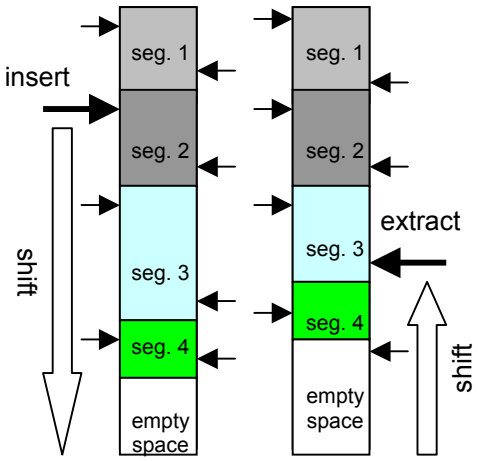

Figure 2. SCB with modifications.
a) with one writing in segment 2
b) with two reading in segment 3

2.2 SCB structure

The SCB is structured by three fundamental parts: the AR Addresses Register, the Control Circuit, and the Storage Cells. The AR register contains *n* cells, equal in number to the storage cells. The Control Circuit is also constituted by *n* cells, one for each storage locality.

The Addresses Register, called AR, contains the writing pointers WWx, and the reading pointers RRx, where WW represents the writing address, RR represents the reading address and x represents the segment number. In addition, AR contains an address of data type for each segment and an address of zero type for all the empty space. The AR register should manage four writing pointers, four reading pointers, four data addresses and a zero address. In total, they are thirteen address values, which are implemented with five bits to achieve an upward order. After resetting AR register only the four address pointers will appear (WW1 to WW4) and the pointer for empty space (ϕ).

The Memory Cells associated to the Writing Pointers correspond to the beginning of each segment and they have the specific purpose of containing only the state of each segment but no data. The state indicates if the segment is empty or not. At the beginning of the process, the state of every segment is empty, for that reason, the Writing Pointers are contiguous and they occupy only the first four cells. The data writing in a segment x is always made in the following cell to Writing Pointer of this segment. The memory cells contents, before the writing, are simultaneously displaced one position downwards.

Each bit in a data word is stored during a single clock period, in a couple of flip-flops connected in series. In the rising transition the first flip-flop is loaded. The second flip-flop is loaded with the clock slope edge. The reading operations are carried out in the second flip-flop before loading a new data.

Reading Pointers don't exist, they are created for the first time during the first writing of each segment. The Reading Pointers always point to first written data or to the oldest in each segment. Both reading and writing operations are executed in a single clock cycle. During a reading operation, an empty locality is generated and it is immediately occupied, in the same clock cycle.

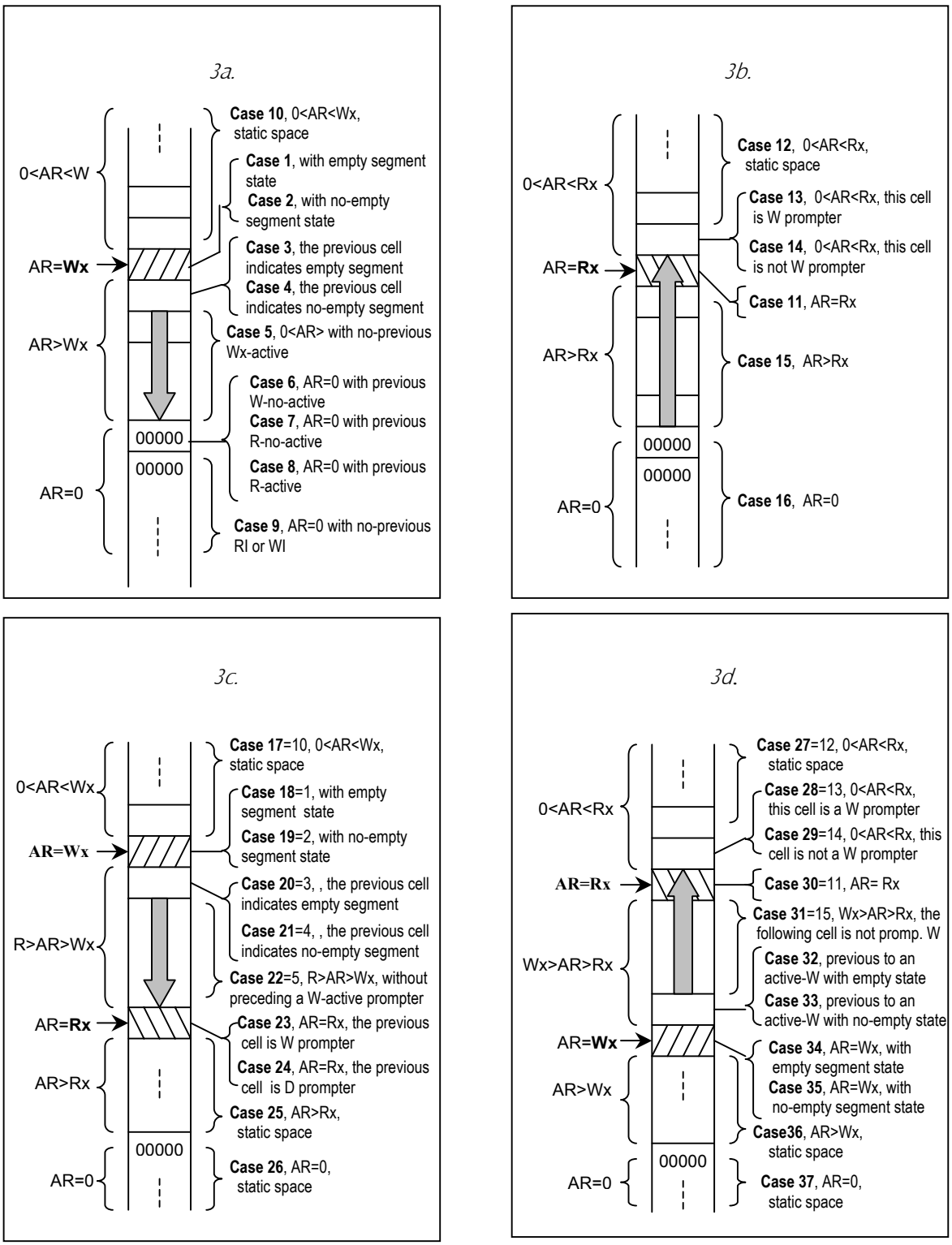

Figure 3. a) Cases with a single writing operation,
b) Cases with a single reading operation
c) Cases with simultaneous reading and writing operations(W < R)
d) Cases with simultaneous reading and writing operations (W > R)

The SCB is controlled by a distributed control circuit, where each memory cell MC, is associated to a Local Control Cell: LCC. Each MC is controlled by its own LCC. The commands generated by each Control Cell are specific for each Memory Cell associated, and they are generated simultaneously in all the cells. The commands that receive the Memory Cells are for the execution of any of the following operations:

- writing
- reading
- writing/reading
- shift down
- shift up
- segment state setting
- no action

The control decisions for each LCC are made starting from the current conditions of the cell and from the conditions of the adjacent cells. There exist 37 different states, according to their operation and location. In Figures 3a, 3b, 3c and 3d all the possible cases and the conditions for each one are described.

In each LCC signs are received and generated to know and let know the cells status. To execute the necessary operation, the control also takes into account the relative position of the writing and reading addresses, when both of them are given simultaneously. Fig. 4 shows the interrelation among the AR register cells, the Local Control Cells, the Memory Cells, and the adjacent cells (upper and lower).

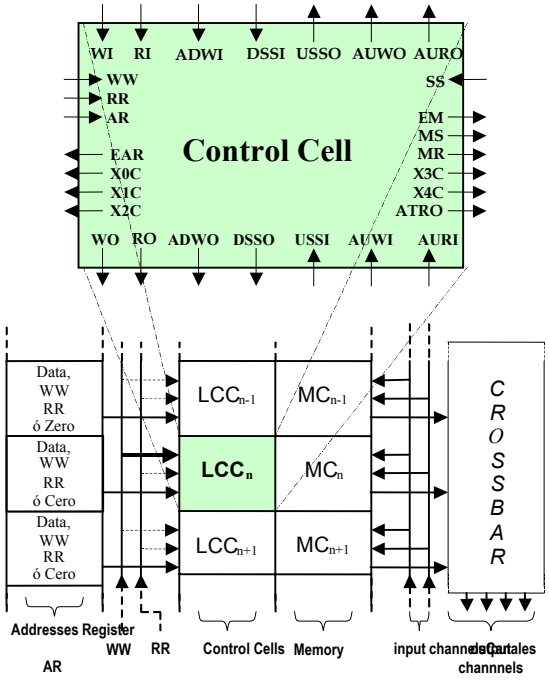

Figure 4. Interrelation of AR register with Control and Memory Cells, in the SCB

The SCB performs the flow control function required for the router structure. The crossbar in Fig. 4 has the function of sending the memory data toward the output channels, in accordance with an output algorithm whose function is part of the flow control of the messages received in memory.

3. CONCLUSION

The high performance offered by the SCB with operation cycles in one clock cycle, besides its high efficiency degree, reached by means of the self-compacting, make it very useful in communication applications where the latency is critical. Particularly, we proposed the SCB to be used as message buffer in multicomputer parallel systems where the

high performance is limited by the "bottle neck" that represents the communication latency. The messages routing in parallel systems is limited mainly in the flow control function. The SCB helps to solve this problem in an important way.

The SCB was designed with a simulation tool provided by FOUNDATION Series Software Co. through XILINX Programmable Logic Co [13]. The tool is based on the VHDL language (VHSIC Hardware Description Language), and it allows the design implementation in an integrated circuit of FPGA type (Field Programmable Gate Array) [10,13,14].

4. REFERENCES

[1] Hwang K. Advanced Computer Architecture: Parallelism, Scalability, Programmability. McGraw-Hill, 1993.

[2] Thin-fries J. G., Vassiliadis S., Pechanek G. G., Johnson H. D. & Green D. M. To Processing Unit Flexible for Multiprocessor Machine Organizations. Instrumentation and Development, vol. 3, not. 4, pages 20-32, 1994.

[3] Jiménez-Flores A. & De Luca P. A., Design of a Flexible and Adaptive Router for Parallel Systems. Computation International Congress CIC´99. IPN, México.

[4] Delgado-Frias J., Vassiliadis S, Johnson H. D., Summerville D. & De Luca A. A Processing Unit for Multiprocesor Organizations. Department of Electrical Engineering, State University of New York at Binghamton. 1996.

[5] Duato, J. & Sudhakar Y. Interconnection Networks: an engineering approach. IEEE Computer Society Press, 1997.

[6] McHugh J. A. Algorithmic Graph Theory. Prentice-Hall, 1990.

[7] Park J., Vassiliadis S. & Cold Thin J. G. Router Architecture for Oblivious Routing Algorithms. Parallel Computing Technologies (PaCT-93), Obninsk, Russia, 1993.

[8] Culler D. E., Singh J. P., and Gupta A. Parallel Computer Architecture: A hardware/Software Approach. Morgan Kaufmann Publishers Inc., 1999, pp. 1025.

[9] Tanenbaum A. S., Structured Computer Organization. Prentice Hall Inc., 1999.

[10] Jay C. VHDL and Synthesis Tools Provide to Generic Design Entry Platform Into FPGAs, PLDs and ASICs. Microprocessors and Microsystems, Volume 17, Issue 7, September 1993, 391-398

[11] Dally W. J., Chao L., Chien A. A., Hassoun S., Horwat W., Kaplan J., Song P., Scott B. T. Architecture of to Message-Driven Processor. 25 Years ISCA: Retrospectives and Reprints, pp 337-344, 1998.

[12] Tamir Y. and Frazier G.L. Dynamically Allocated Multi-queue Buffers for VLSI Communication Switches. IEEE Trans. On Computers, volume 41, Not. 6 pp. 725-737, June, 1992.

[13] VHDL Reference Guide. Xilinx Inc., 1999.

[14] Pellerin D., Taylor D. VHDL. Made Easy. Prentice Hall PTR., 1997.