The MEGA Router: A Hardware Message Passing Gate Array Router

7L−4

Andrew FLAVELL and Yoshizo TAKAHASHI
Department of Information Science and Intelligent Systems, University of Tokushima
e-mail: flavell@n30.is.tokushima-u.ac.jp

I. INTRODUCTION

Parallel computers based on the multiple-instruction multiple-data (MIMD) message-passing paradigm have proven to be effective in the solution of a large number of problems [1]. Their direct, regular structure allows the exploitation of locality of communications and provide networks which are flexible in their ability to be scaled. Inter-process communication in a multicomputer is implemented by the passing of messages via a communications network. Thus, the performance of the communications network has a major effect on the overall performance of the system. Often a message will have to pass through a number of intermediate nodes before reaching it's destination and it is therefore desirable that the functions of message routing and computation are separate.

In this paper we present a preliminary discussion of a MEssage-passing Gate Array (MEGA) router, which is currently under development at the University of Tokushima. Fujitsu's ViewCad gate array design software is being used to design and evaluate the performance metrics possible with a semi-custom design. Section II presents a discussion of the requirements of a message router. Section III introduces the architecture of the MEGA router and conclusions are presented in Section IV.

II. ROUTER OVERVIEW

Presented in Fig. 1 is a block diagram of the basic modules of a message router.

The input unit controls the message flow into the router, buffers messages when two or more request the same output, determines the output port for an incoming message based on the routing algorithm, and requests access to the switch and output unit by queuing requests to the arbitration unit. The routing algorithm

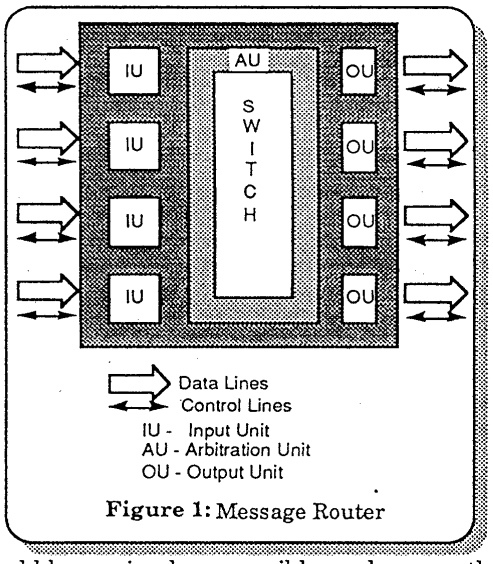

Figure 1: Message Router

should be as simple as possible, and ensure that deadlock and livelock conditions cannot exist in the network. The arbitration unit controls access to the switch and output unit so as to minimize the average delay experienced by a message in the network. The output unit controls the flow of data across the physical channel separating two routers, or a router and a processing element.

III. MEGA ROUTER

A number of researchers and commercial developers have developed hardware routers for use in multicomputer networks recently [3][4][6]. These have typically been implemented using full custom VLSI techniques which has enabled them to achieve high throughput, and low message latencies. However, a number of advantages exist in taking a semi-custom approach to the design [5]. These include a shorter design time, lower production costs for small volumes of devices, and well established design and simulation tools.

The MEGA router is being implemented using Fujitsu's ViewCad system, which is hosted on a Fujitsu FMR-70 personal computer. This system allows up to 32,000 basic cells per design, and supports schematic entry, design rule checking, functional simulation and critical path analysis.

The MEGA Router: A Hardware Message-Passing Gate Array Router.
Andrew Flavell and Yoshizo Takahashi
Department of Information Science and Intelligent Systems, University of Tokushima.

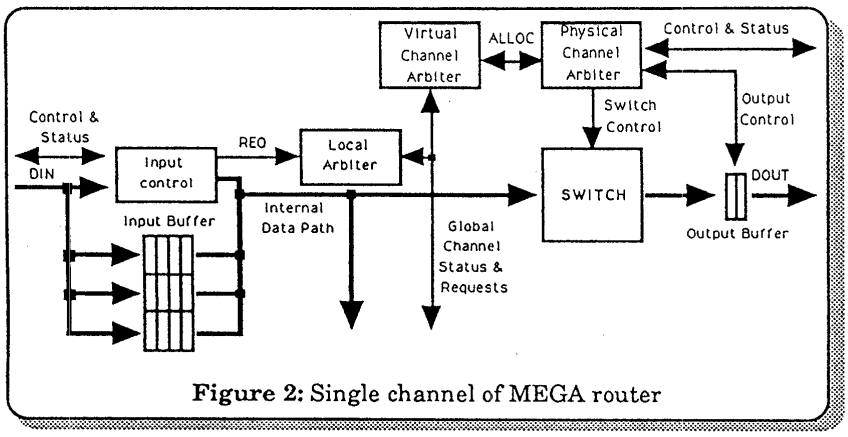

Figure 2: Single channel of MEGA router

Fig. 2 illustrates a block diagram of a single channel of the MEGA router. When a packet arrives at the input, the input controller will determine the output port that the packet requires, and signal the local arbiter that it has an output request. The output port assignment is determined by the routing algorithm, which is a function of the network topology. The local arbiter assigns one of the competing input requests to be passed to the virtual channel arbiter of the requested output port. Once an input is assigned a output virtual channel by the virtual channel arbiter, it will hold this resource until packet transmission is complete. This is required as the MEGA router implements wormhole routing, and therefore only the leading *flit*[1] [2] contains routing information. Once a packet is assigned a virtual channel it will compete with other packets, at the *flit* level, for the switch and physical channel. The physical channel arbiter maintains buffer status information for the input buffers on the current node and the receiver buffers on the next node. This ensures that the physical channel will only be assigned to a virtual channel if the receiving buffer of the next router is empty and the input buffer of the current router is full. A small output buffer is provided to decouple the timing of the switch and physical channel resources.

The first design prototype has virtual channel buffers 16 bits wide and four words deep. Each virtual channel requires 417 basic cells (BCs) to implement and a full buffer containing eight virtual channels, including input and control logic, and output multiplexers requires 3645 cells. The arbiters all implement round-robin arbitration and the router contains 10 uni-directional ports, which are formed into 5 bi-directional pairs. The initial design specification calls for a minimum data throughput of 20 Mbytes/channel, which requires a 10 MHz clock rate.

Future research will involve evaluating the performance of the router using the ViewCad functional simulator. By varying parameters such as the number of virtual channels, virtual channel depth, virtual channel allocation, network topology (via the routing algorithm), and arbitration schemes, we will be able to ascertain the performance of the router prior to actual fabrication.

IV. CONCLUSIONS

In this paper we have presented an overview of the MEGA router project, which is being undertaken to design and evaluate the performance of a dedicated hardware router for use in a MIMD message passing multicomputer. By taking advantage of the design tools available for gate array design we will be able to completely evaluate the performance metrics of our router prior to fabrication.

References

[1] W. C. Athas and C. L. Seitz, "Multicomputers: Message -Passing Concurrent Computers", *IEEE Comp.*, vol. 21, pp.9-24, Aug. 1988.

[2] W. J Dally, "Virtual Channel Flow Control", *IEEE Trans. Comp.*, vol. 3, no. 2, pp.194-205, March 1992.

[3] W. J. Dally and C. L. Seitz, "The torus routing chip", *Distributed Comp.*, vol. 1,pp.187-196, 1986.

[4] W. J. Dally and P. Song, "Design of a self-timed VLSI multicomputer communication controller", *in Proc. Int. Conf Comp. Design,* IEEE Comp. Society Press, pp.230-234, Oct. 1987.

[5] C. E. Leiserson *et al.*, "The Network Architecture of the Connection Machine CM-5", *in Proc. 4th Annual ACM Symp. on Parallel Arch. and Alg.*, ACM Press, pp.272-285, Jul. 1992

[6] Y. Tamir and G. L. Frazier, "High Performance Multi-Queue Buffers for VLSI Communication Switches", *in Proc. 19th Annual Symp. on Comp. Architecture,* IEEE Comp. Society Press, pp.343-354, June 1988.

[1] A flow control digit, or *flit*, is the smallest unit of information on which flow control is performed.