LABORATORY FOR COMPUTER SCIENCE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Arctic Routing Chip

Computation Structures Group Memo 373
March 7, 1994

G. Andrew Boughton

This report describes research done at the Laboratory for Computer Science of the Massachusetts Institute of Technology. Funding for the Laboratory is provided in part by the Advanced Research Projects Agency of the Department of Defense under the Office of Naval Research contract N00014-92-J-1310.

545 TECHNOLOGY SQUARE, CAMBRIDGE, MASSACHUSETTS 02139

Arctic Routing Chip*

G. Andrew Boughton

Massachusetts Institute of Technology, Laboratory for Computer Science, Cambridge Mass 02139, USA

Abstract. Arctic is a 4x4 packet routing chip being developed for the *T multiprocessor. Arctic can be used to implement a variety of staged networks and will be used to implement a fat tree network for *T. Arctic meets the requirements of *T and of a wide class of systems. This paper discusses the key features of Arctic. These include its buffering scheme which enables very high utilization of network links and its test and control system which provides error detection, limited error handling, and in-circuit testability.

1 Introduction

Arctic is a four input four output packet router implemented on a Motorola H4CP gate array chip. Arctic has *all* the features necessary for use in a commercial multiboard multiprocessor such as *T [7, 8, 1]. It has high bandwidth (200 MBytes/sec/port), two priority levels, packet sizes up to 96 bytes, and extensive error detection; it has limited error handling, keeps statistics, can directly drive long PC traces, and provides significant testing support. While Arctic has special features to support fat tree networks [6], it can also be configured to support any of a wide variety of other staged network topologies.

Arctic uses a sophisticated buffer management scheme that greatly increases the effectiveness of its buffers and the utilization of network links. This scheme is similar to that developed by Joerg for the PaRC routing chip [4, 5] of the Monsoon multiprocessor.

Arctic has a test and control system that is significantly more sophisticated than those in most previous chips. This system is accessible through a JTAG port. It is used to configure the chip, detect errors, count packet flow statistics, and handle certain errors. It can also be used to access most of the state elements of Arctic and to run in-circuit chip verification tests.

2 Basic Structure

As is shown in Figure 1, Arctic is composed of four input sections, four output sections, the crossbar, and the test and control section. There are five clock domains. The crossbar and the output sections are in a single central clock domain.

* The research described in this paper was supported in part by the Advanced Research Projects Agency under Office of Naval Research contract N00014-92-J-1310.

Each incoming data link has its own clock which drives the corresponding input section. The central clock clocks data out of the buffers.

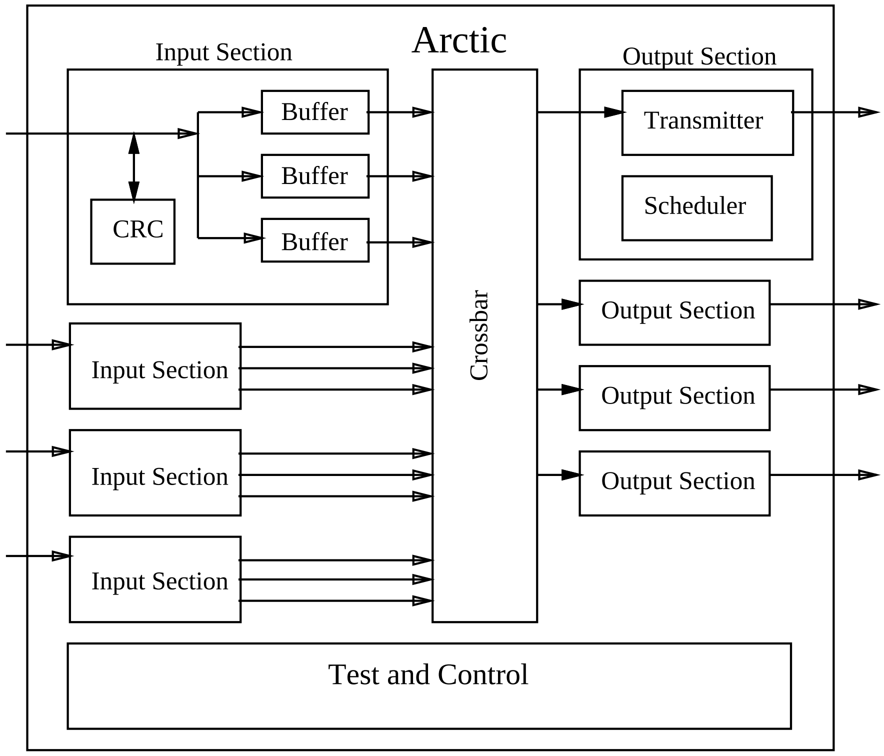

Fig. 1. Arctic Block Diagram

3 Buffering

All buffering in Arctic is located in the input sections. Each input section contains three buffers - each capable of storing a maximum size packet. Each buffer is capable of storing either a priority packet or a normal packet. The last empty buffer is reserved for a priority packet; this ensures that normal packets can not block priority packets. The head of a packet can be transferred out of a buffer before the end of the packet has been received (virtual cut-through with partial cut-throughs allowed). The crossbar has 12 inputs and 4 outputs. Each of the packet buffers in the chip is directly connected to the crossbar. The scheduling scheme used in Arctic ensures that if any buffer in the chip contains a packet destined for a given output then a packet will be transferred to that output as soon as the output becomes available. Any four of the 12 input buffers can be simultaneously transferring to output sections. Packets received on a given input and destined for a particular output are transferred in the same order as they were received, thus in order delivery of packets is maintained along a given network path. Packets received on different inputs and destined for a particular output are scheduled using a round-robin scheme which guarantees that no packet is starved for service. All priority packets waiting for a particular output are transferred before any waiting normal packet is transferred to that output.

This buffering system leads to high utilization of the network links. While an exact analysis of this system is complex, a simple analysis provides an insight of the advantages of this system over a simpler scheme. If Arctic used only a single long FIFO buffer in each of the four input sections then at most four buffered packets would be candidates for output at a given time. The chance that at least one out of four randomly addressed packets is destined for a given output is $1-(.75)^4$ which is approximately .68. However in Arctic's buffering scheme all of the 12 buffered packets are candidates for output. The chance that at least one out of 12 randomly addressed packets is destined for a given output is $1-(.75)^{12}$ which is approximately .968. Of course this analysis is not precise (for either buffering system) since it assumes that each of the buffers contains a packet and since it assumes that the buffered packets are randomly addressed, but it does to the first order identify the advantage of Arctic's buffering system over a simpler FIFO system. A longer discussion of a similar analysis of this buffering system can be found in [4].

Related buffering schemes that avoid the problems of a simple FIFO system have been studied by other researchers [2, 10]. The capabilities of these schemes differ from those of the PaRC/Arctic scheme. For example, virtual channel schemes can be used to avoid routing deadlocks in mesh networks and the PaRC/Arctic scheme does not address that issue. However the PaRC/Arctic scheme has a number of features that are not normally provided by virtual channel implementations. A single input section can simultaneously transfer buffered packets to multiple outputs. All the buffers associated with a given input are managed in a single pool rather than using several smaller disjoint pools; this provides more effective buffering. A blocked packet is stored in a single Arctic chip and does not occupy any link (or any virtual channel); this reduces the amount of tree buffering caused by a blocked packet.

For similar reasons the PaRC/Arctic buffering scheme provides better performance than the DAMQ and SAFC schemes discussed in [10, 9]. DAMQ does not allow a single input section to simultaneously transfer buffered packets to multiple outputs. SAFC does not manage in a single pool all the buffers associated with a given input. The performance advantage of these features is discussed in more detail in [4, 10]. The PaRC/Arctic scheme does require a more complex crossbar than the DAMQ and SAFC schemes. However the scheduling circuitry required by the PaRC/Arctic scheme is no more complex than that required by the DAMQ scheme.

4 Link Technology

Arctic supports a network link with a 16 bit wide data path. A clock and a frame signal are sent with the data. A flow control line runs in the reverse direction from the data. Each data bit is transmitted using a single ended GTL driver at 100 Mbits/sec. The design goal is to support printed circuit board transmission lines that are over one meter long. Encoding is used on the frame and flow control lines.

The flow control signal is actually a signal from the receiver indicating that it has freed a buffer and that the buffer is available for use. The transmitter keeps a running count of free receiver buffers. The count is decremented each time a packet is sent and incremented each time a buffer free signal is received. Of course if the transmitter thinks that no buffers are free then it will not send a packet. The count is initialized at startup. The initial value is configurable allowing Arctic to interface receivers with various numbers of buffers. This approach to flow control was suggested by Bob Greiner of Motorola and is related to the *sliding window* scheme used in TCP/IP [3]. This approach works with any amount of link delay. In that respect it is superior to the more common wait line scheme which must take into account link delay. A wait line must be asserted at least two link delays before buffering is exhausted.

Although Arctic's internal data paths are 32 bits at 50 MHz, 16 bit wide link data paths are used to reduce the number of off chip signals. This necessitates the use of a small amount of 100 MHz circuitry in the edges of the input and output sections.

5 Routing

Arctic networks use source based routing; the course of a packet through an Arctic network is completely determined by routing information placed in the packet header by the source. This approach has several benefits. System issues can be considered in the routing of a packet. For example, different packet types (such as I/O packets) can be routed through separate subnetworks. Source based routing is flexible since major changes in network routing policy can be accomplished simply by modifying the route generation algorithm used in the packet sources. Source based routing also supports system reconfiguration including network reconfiguration to isolate broken network hardware.

For a fat tree network, two routing fields in the packet header are used. The source processor places a random bit string in the first and the destination address in the second. As the packet travels from the leaves to the root, each Arctic checks the appropriate prefix of the destination address to determine if the packet must travel farther from the leaves. If this is the case then, since there are several acceptable paths away from the leaves, a portion of the first header field is used to choose an output (away from the leaves). Otherwise a portion of the destination address (the second header field) is used to determine the appropriate output toward the leaves.

The use of a random bit string generated at the source should provide a more globally random distribution of traffic than would be produced if each Arctic dynamically routed packets based on local flow control information. Also since this string is generated at the source, a variety of modifications can easily be made to this fat tree routing algorithm. For example, we can route a packet through a particular subnetwork by fixing certain bits in the first routing field (and placing a random bit string in the remaining portion of the field). Given that the first field determines how the packet is routed from the leaves and given the structure of the fat tree, the fixed bits mean that as the packet travels from the leaves it can only go toward a particular subset of root nodes (the subset is completely defined by the fixed bits). This means that the packet must travel in a subnetwork defined by this subset of root nodes.

Arctic can be configured to implement any of a wide variety of source based routing schemes including the fat tree routing scheme sketched above. For each incoming packet Arctic compares two subfields of header bits with two reference values. Based on these comparisons Arctic selects two bits from the packet header (or a constant and one bit from the header, or two constants). The value of these two bits determines the appropriate output port for the packet.

6 Test and Control System

The test and control system (TCS) provides a complete set of maintenance features. The TCS can be used to thoroughly test the chip. It is used to configure the chip and to control normal operation. It provides error detection and limited error handling.

A host/diagnostic processor (DP) is used to control an Arctic network. The DP controls each Arctic through the Arctic's TCS. The DP accesses an Arctic's TCS through that Arctic's JTAG port.

6.1 Test

Arctic provides more support for in-circuit chip verification than previous router chips. The TCS can be used to place Arctic in a special *low-level-test* mode. While in this mode, Arctic has a single clock domain and the TCS can be used to access scan rings that contain almost all of the chip's state elements. A test vector can be applied through the scan rings. The starting state is scanned in, the system clock is cycled, the resulting system state is scanned out, and the results are checked by the DP. This scan facility can be used to run in-circuit verification tests that have been generated by an automatic test pattern generation program. Such tests should provide a high level of stuck-at fault coverage.

The TCS can also be used to access boundary scan rings that contain all the chip inputs and outputs. These rings can be used to run any desired sequence of test packets through Arctic. In addition these rings can be used to load and and read test patterns from the packet buffers.

The statistics memory has a built-in-self-test circuit. The TCS is used to initiate the test and to read the results.

6.2 Configuration

Arctic's configuration information is primarily composed of routing configuration but it also includes some error handling configuration and some flow control configuration. The TCS is used to scan in this information in *configuration* mode.

6.3 Normal Operation

During normal packet routing operation, the TCS is used to monitor the operation of the chip, detect errors, provide some error handling, and count packet flow statistics.

Errors. The design of Arctic assumes that Arctic's links are very reliable, but Arctic has extensive circuitry for detecting link errors should they occur. Each packet contains a 16 bit CRC field. Arctic checks the CRC of each incoming packet. An idle pattern is sent on a link whenever packets are not being sent and the receiving Arctic checks for this idle pattern. Arctic checks that all incoming frame and flow control signals are correctly encoded. Lastly, Arctic checks that each incoming clock is not disconnected by checking that the clock is toggling. A two bit counter is used for each possible link error for each link. The TCS can be used to access or clear the error counters. In addition, a special error pin is also provided. The pin is asserted if any error occurs.

The DP can make network modifications in response to a static error. While normal packet routing is occurring, the DP can issue commands through an Arctic's TCS to enable or disable a port, block an output port, or invoke flush on an output port. When the DP determines that an error has occurred, it can read the error counters of the Arctics and determine the faulty link or the faulty router. The DP can then disable the input ports connected to the unusable link(s) and invoke flush on the output ports connected to the unusable link(s). Flush causes those output ports to discard all packets sent to them. In *T the DP can also instruct the processing elements to update their route generation tables so that no new packet is routed over an unusable link(s). Thus Arctic's features allow the DP to provide some error handling. Of course this scheme causes packets to be flushed and data to be potentially lost. However it is possible for the processing elements to use an end-to-end acknowledgment/retry protocol on top of the packet transport layer provided by the Arctic network and such a protocol can avoid the loss of data at the cost of additional messaging complexity.

Statistics. Arctic records various statistics about the flow of packets out of each of its output ports. These include the number of packets transmitted, the number of priority packets transmitted, and a special statistic for use in fat tree networks. The fat tree statistic indicates either the number of packets output on the port that came from above this router or the number that came from below this router, depending on the configuration of the output. A 36 bit count is kept of each statistic. The TCS can be used to access or clear the statistics counters.

7 Conclusion

Arctic is a high bandwidth packet routing chip. It provides a more comprehensive set of features for implementing a staged network than previous router chips. The routing circuitry can support a wide variety of staged networks and is particularly well suited for fat tree networks. The buffer management scheme provides high utilization of network links. The ports can be directly connected to long PC traces that cross clock domain boundaries. Thorough error detection and statistics gathering are provided. The novel scan and test facilities make Arctic testable both in a tester and in circuit. Arctic has all the capabilities required for use in a commercial multiprocessor such as *T.

Acknowledgments

I would like to thank Greg Papadopoulos, Bob Greiner, and Chris Joerg for many of the concepts used in Arctic. I would like to thank Jack Costanza, Richard Davis, Tom Durgavich, Doug Faust, and Ralph Tiberio for concepts and design work. I would like to thank Yuval Koren, and Elth Ogston for testing the design. I would like to thank Mike Beckerle, Ed Greenwood, Jamey Hicks, and John Morris for reviewing the design. Finally I would like to thank Satoshi Asari, Steve Chamberlin, Thomas Deng, Eric Heit, Tom Klemas, Jimmy Kwon, Wing Chi Leung, and Gowri Rao for their contributions.

References

1. B. S. Ang, Arvind, and D. Chiou. StarT the Next Generation: Integrating Global Caches and Dataflow Architecture. CSG Memo 354, Computation Structures Group, LCS, MIT, February 1994.
2. W. J. Dally. Virtual Channel Flow Control. In *Proceedings of the 17th International Symposium on Computer Architecture*, May 1990.
3. Information Sciences Institute, University of Southern California, Marina del Rey, Calif. *Transmission Control Protocol, DARPA Internet Program, Protocol Specification*, September 1981. RFC: 793.
4. C. F. Joerg. Design and Implementation of a Packet Switched Routing Chip. TR 482, Laboratory for Computer Science, MIT, Cambridge, Mass., 1990.
5. C. F. Joerg and G. A. Boughton. The Monsoon Interconnection Network. In *Proceedings of the 1991 IEEE International Conference on Computer Design*, October 1991.
6. C. E. Leiserson. Fat Trees: Universal Networks for Hardware-Efficient Supercomputing. In *Proceedings of the 1985 IEEE International Conference on Parallel Processing*, August 1985.
7. R. S. Nikhil, G. M. Papadopoulos, and Arvind. *T: A Multithreaded Massively Parallel Architecture. In *Proceedings of the 19th International Symposium on Computer Architecture*, May 1992.
8. G.M. Papadopoulos, G.A. Boughton, R. Greiner, and M.J. Beckerle. *T: Integrated Building Blocks for Parallel Computing. In *Proceedings of Supercomputing '93*, November 1993.

9. Y. Tamir and H. C. Chi. Symmetric Crossbar Arbiters for VLSI Communication Switches. *IEEE Transactions on Parallel and Distributed Systems*, 4(1), January 1993.
10. Y. Tamir and G. L. Frazier. High-Performance Multi-Queue Buffers for VLSI Communication Switches. In *Proceedings of the 15th International Symposium on Computer Architecture*, 1988.