Architecture and Optoelectronic Implementation of the WARRP Router*

Timothy Mark Pinkston Yungho Choi Mongkol Raksapatcharawong

SMART Interconnects Group
Tel: +1-213-740-4482 Fax: +1-213-740-4418
EE-Systems Dept., University of Southern California, Los Angeles, CA 90089-2562
{tpink,yunghoc,mraksapa}.usc.edu; http://www.usc.edu/dept/ceng/pinkston/SMART.html

Abstract

The WARRP router implements progressive deadlock recovery-based true-fully-adaptive routing in an efficient architecture designed to maximize flexibility as well as speed. The WARRP II version addresses the I/O pin-out problem by integrating dense high-bandwidth optoelectronic transceivers onto the router chip capable of providing over 300 GBytes/sec per square centimeter [7]. To our knowledge, this is the first fully-functional optoelectronic network router implemented to-date.

1 Introduction

The importance of the interconnection network in high-performance parallel systems is conspicuous. The basic switching component that establishes the network's topology and determines the allowable paths packets can take through the network to guarantee deadlock freedom is the router. Previous network routers based on avoiding deadlock restrict routing freedom and do not efficiently utilize scarce network bandwidth, resulting in lower overall network performance and fault-tolerance capabilities. Restricting routing can cause unbalanced traffic loads [1] and can place constraints on design optimizations that could increase the speed of the internal router architecture [2]. Moreover, deadlock avoidance-based routers including the SGI SPIDER [3] (a static router) and the Cray T3E network router [4] (an adaptive router) preallocate considerable network resources in the form of virtual channel buffers to avoid deadlock anomalies which have empirically been shown to occur very rarely [5].

The *WARRP* router (*W*ormhole *A*daptive *R*ecovery-based *R*outing via *P*reemption) implements a progressive deadlock recovery routing algorithm which allows unrestricted *true fully adaptive routing* [6]. The *WARRP* router relaxes routing restrictions designed to avoid deadlock by allowing packets to be routed toward destinations on any virtual channel in any network dimension, achieving maximum utilization of network bandwidth and increased fault-tolerance. Deadlock freedom is guaranteed by redirecting potentially deadlocked packets to flit-sized *deadlock buffers* central to the routers which preempt physical channel bandwidth in order to synthesize recovery. Access to deadlock buffers is controlled so that, system-wide, these deadlock handling resources form what is essentially a deadlock-free lane on which potentially deadlocked packets can be delivered to resolve impending deadlock.

The *WARRP* router is implemented using a bipartite structure that separates deadlock handling resources from normal (non-deadlock) routing resources. Each can then be optimized separately to achieve maximum flexibility and efficient recovery. The internal router architecture is further optimized by employing a partitioned crossbar structure organized by virtual channel network [2]. This allows the *WARRP* router to implement the highest degree of adaptivity while not sacrificing router speed, regardless of the number of virtual channels per physical link. Further, the *WARRP II* version of the router addresses the pin-out problem by integrating dense high-bandwidth optoelectronic transceivers onto the router chip capable of providing over 300 GBytes/sec of optical I/O per square centimeter [7].

Figure 1 shows a block diagram of the router which is designed for two-dimensional torus topologies. It provides high-bandwidth communication through four bidirectional physical links directly connecting neighboring routers plus one bidirectional injection/reception link connecting its local processor node. Packets of arbitrary length can be routed but are divided into 8-bit flits. Each physical link implements three bidirectional virtual lanes. The internal datapath of the router is equal to the external link width for design simplicity. This design constraint limits the speed and number of optical I/O that can potentially be used.

The remainder of this paper is organized as follows. Section 2 gives background on the progressive deadlock recovery routing scheme implemented in WARRP. Section 3 describes the architectural design of the router. Section 4 presents the

*This research was supported by an NSF Research Initiation Award, grant ECS-9411587, and an NSF Career Award, grant ECS-9624251.

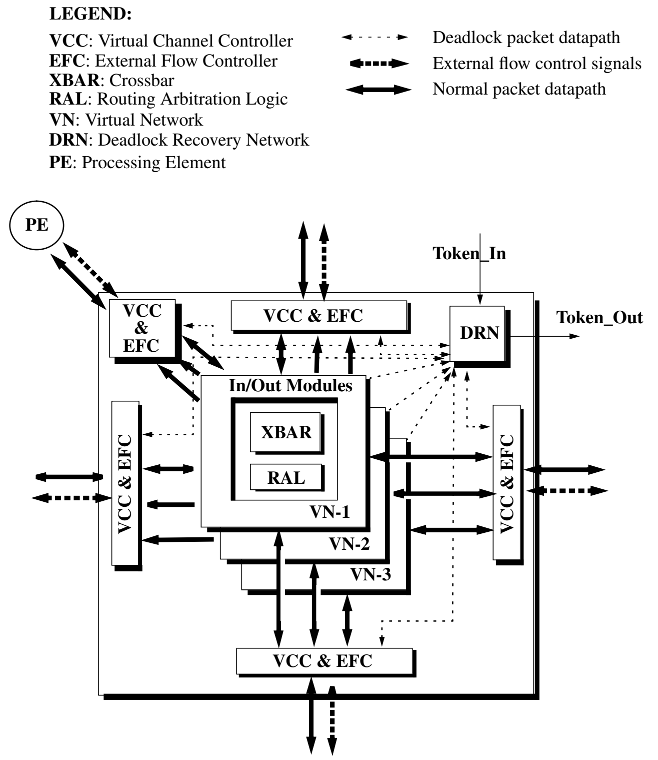

Figure 1. Block Diagram of the WARRP Router.

implementation of the router, and Section 5 concludes the paper with some performance figures.

2 Adaptive Recovery-based Routing

Deadlock recovery routing is optimistic as it permits unrestricted routing of packets in the absence of deadlock and efficiently recovers from rare cases of impending deadlock [8, 9]. Recent empirical studies confirm that deadlocks are highly improbable when routing freedom abounds [5]. Due to this, the *WARRP* router implements the *Disha-Sequential* [9, 10] recovery-based routing algorithm which progressively redirects suspected deadlocked packets toward destinations rather than regressively aborting and later re-introducing these packets into the network from source nodes as in [8]. Though highly improbable, multiple deadlocks can be resolved sequentially one after another.

Progressive recovery from deadlocks is through a single flit-sized deadlock buffer central to each router accessible from all neighboring routers. System wide, these buffers form what is collectively a recovery path which is essentially a "floating" channel buffer shared by all physical dimensions of a router. Mutually exclusive access to deadlock buffers implemented by a circulating token ensures that the recovery path remains deadlock-free [9, 10]. Turn selection criteria is also used to limit the eligibility of packets to recover.

On the event of an impending deadlock, an eligible packet

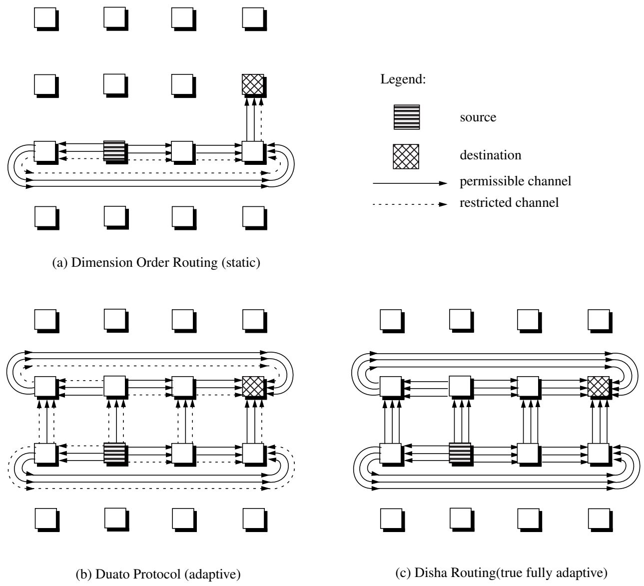

Figure 2. Routing Adaptivity Comparison.

which captures the token is switched to the deadlock-free lane starting at the current router's deadlock buffer and routed minimal adaptively along a recovery path leading to its destination where it is consumed to break the impending deadlock. Network bandwidth may be temporarily deallocated from other "normal" packets and assigned to the "deadlock" packet during the recovery process and re-allocated back to preempted normal packets when the deadlock packet's tail passes. Hence, network bandwidth is dynamically allocated to recover from deadlock only in the rare instances when impending deadlock is suspected; otherwise, all network bandwidth is preallocated to true fully adaptive routing of normal packets.

2.1 True Fully Adaptive Routing

The *WARRP* router implements true fully adaptive routing in which unrestricted routing towards the destination on all virtual channels in all network dimensions is allowed by normal packets. Because of this high degree of routing freedom, packets have a higher chance to route around congestion and faulty nodes/links in the network. This is illustrated in Figure 2 for a two-dimensional torus network with three virtual channels per physical link. The routing freedom allowed by two well-known avoidance-based routing algorithms implemented by current router chips is compared to that of the recovery-based routing algorithm implemented by the *WARRP* router: static dimension order routing [11] (used in the MRC, SPIDER, and Cray T3D routers), adaptive Duato Protocol routing [12] (used in the MIT Reliable Router and Cray T3E router), and true fully adaptive *Disha* routing [9] (used in the *WARRP* router). In the illustration, we assume that the dateline [4] crosses the source router.

Dimension order routing forces packets to traverse dimensions in some fixed order (e.g., X then Y), providing packets with at most only two virtual channel options at each intermediate router or a total of 16 different paths from source to destination, as shown in Figure 2(a). Duato's Protocol allows one additional virtual channel to be used corresponding to the adaptive channel in the non-dimension-ordered direction for a total of 36 different paths, as shown in Figure 2(b). If, however, these permissible channels all happen to be occupied by other packets enroute, packets must remain blocked even if the other (restricted) channels are free. *Disha* routing, on the other hand, allows packets to access all virtual channels along any path to the destination (only minimal ones are shown here) for as many as six virtual channel options at each intermediate router or a total of 486 different routing paths, as shown in Figure 2(c). As confirmed in previous work [6, 9], this greater routing flexibility allows for significant improvement in network performance and fault-tolerance by increasing the utilization of available network bandwidth.

2.2 Recovery Eligibility

Since only one of the packets involved in an impending deadlock needs to be removed to resolve deadlock, the *WARRP* router implements some amount of selectivity by allowing only eligible packets to access deadlock buffers and route on the recovery lane. In addition to blockage and token capture, packet eligibility is also determined by two routing criteria: westward *turns* [13] and *dateline* [4] crossings.

Previous work [4, 13] has shown that restricting turns made in routing and restricting routes over the dateline (e.g., wraparound links in tori) can prevent the formation of cyclic dependencies necessary for deadlock. It follows that, for deadlock to form, at least one packet involved must make a turn or dateline route. The *WARRP* router uses this property to minimize the number of packets which might be mistaken for being deadlocked unnecessarily. Only those packets which make a westward turn or a dateline route is given eligibility to use the recovery lane.

This selectivity criteria allows the recovery lane to be used for special circumstances that can benefit from deadlock-free preemption of network bandwidth for fast delivery other than deadlock handling. For instance, *priority packet routing* can occur by making packets eligible to use the recovery lane upon injection into the network. Relief from network congestion can also occur by allowing packets meeting the selectivity criteria (which may not actually be involved in deadlock) to be eligible to recover.

3 Router Architecture

3.1 Overview

The *WARRP* router separates normal routing resources from deadlock handling resources to allow minimum coupling

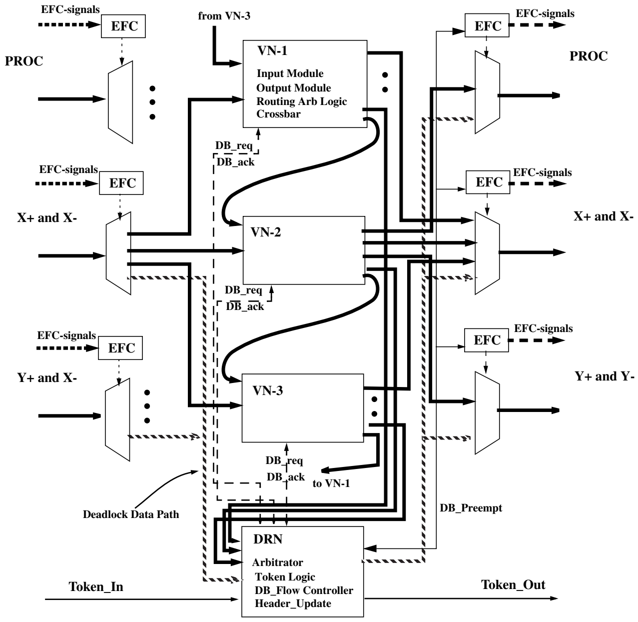

Figure 3. Organization of the WARRP router.

between the normal routing process and the deadlock handling process. This bipartite structure also allows for maximum design flexibility and exploitation of unrestricted routing using the normal routing resources while still ensuring deadlock freedom. The internal structure of the normal routing section is partitioned into three *virtual network* (VN) modules as shown in Figure 3. Incoming packets are directed to one of the virtual networks in the normal routing section or to the *deadlock module* (DRN) in the deadlock handling section, depending on the status of the control signals sent along with the packet (signifying deadlock or normal).

A normal packet entering the router is received by the *input module* of the corresponding virtual network module. Routing information is extracted from the *header* flit (first flit of the packet) by the *address decoder* which sequentially makes requests to the *routing and arbitration module* for a permissible output link by cycling through the following request types until being granted: (1) an *output module* in the same dimension of the current virtual network, (2) an *output module* in another dimension of the same virtual network, (3) a *connect channel* to the next virtual network, and (4) the recovery lane in the deadlock module (only eligible packets are allowed to make this request). If the first request type is granted, the input-buffered packet is routed over the current virtual network's *crossbar* to the allocated output module after the header is updated. If a connect channel is allocated to the packet due to all the available output links in the current virtual network being occupied by other packets, the packet is transferred to the next virtual network's input module where the same routing actions are performed. If the recovery lane request is granted due to the permissible output modules and connect channel of the vir-

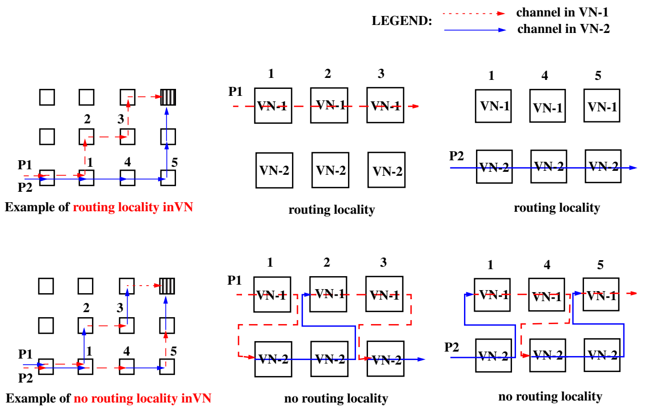

Figure 4. Routing Locality in VN: This figure shows two examples of how routing locality in virtual network (VN) can and cannot occur.

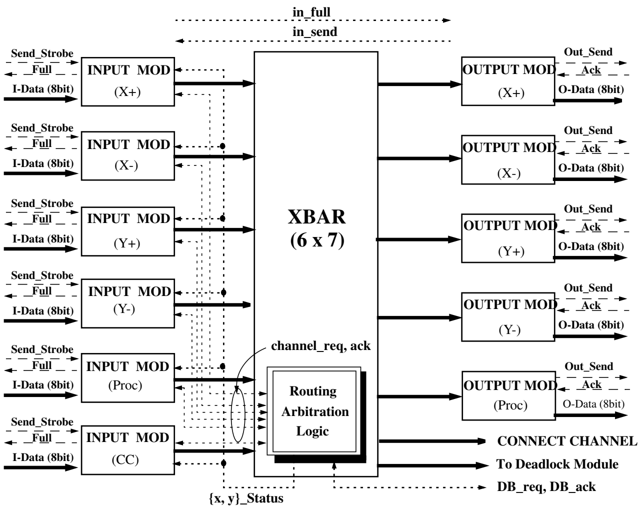

Figure 5. Virtual Network Module.

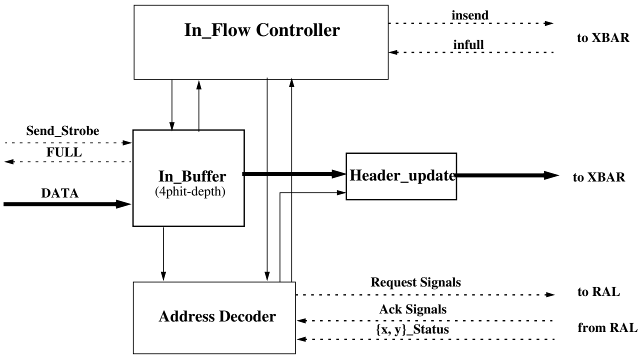

Figure 6. Block Diagram of the Input Module.

tual network being busy, the crossbar is configured to redirect the suspected deadlock packet to the deadlock module to invoke the recovery process once the token is captured. The status of the packet is then changed from "normal" to "deadlock". As a final step, packets are time-multiplexed onto the physical link from the output modules of the three virtual networks in a round-robin fashion or given priority over the link if coming from the deadlock module. The input, output, and deadlock modules are deallocated from the packet once the *tail* flit passes.

A deadlock packet entering the router is latched directly into the central deadlock module in the deadlock handling section of the router. The header flit of the deadlock packet is decoded by the address decoder in the deadlock module. This routing information is used to decide which physical link will be allocated to the deadlock packet, preempting other normal packets which may be routing over the link. The control of normal and deadlock packet flow between routers is regulated by the *flow control* module to ensure no overflow. The *WARRP* router implements these functions using the major components described below.

3.2 Virtual Network Module

The *WARRP* router implements a decoupled internal router architecture partitioned into separate virtual networks which are linked by intra-router connect channels to maximize router speed and minimize implementation cost. This organization enforces *routing locality in virtual network* in which routing options that direct packets to output links within the same virtual channel network are selected over those that change virtual network class, as illustrated in Figure 4. By partitioning the internal architecture into smaller units based on virtual network, routing arbitration complexity and crossbar size is considerably reduced without sacrificing routing freedom [2].

The virtual network modules are identical, and each has complete routing functionality implemented by the input modules, output modules, crossbar, and router and arbitration module, as shown in Figure 5. Each virtual network has an injection input module connected to the processor node to distribute traffic injection over all virtual networks and balance the load. In addition, there are four input modules from the $X+$, $X-$, $Y+$, and $Y-$ directions and an input module from the $i-1$ mod 3 virtual network connect channel. The seven output modules connect to the four directions, the processor node port, the $i+1$ mod 3 virtual network, and the deadlock module. Also, each virtual network has its own crossbar and associated crossbar routing and arbitration module so that routing decisions in all three virtual networks can occur independently and simultaneously. The 6×7 crossbar allows non-blocking connections between all input modules and output ports.

3.3 Input Module

The input module buffers packet flits, decodes header flits, generates requests for permissible output links and header up-

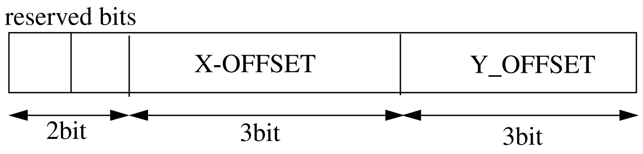

Figure 7. Header Flit Format.

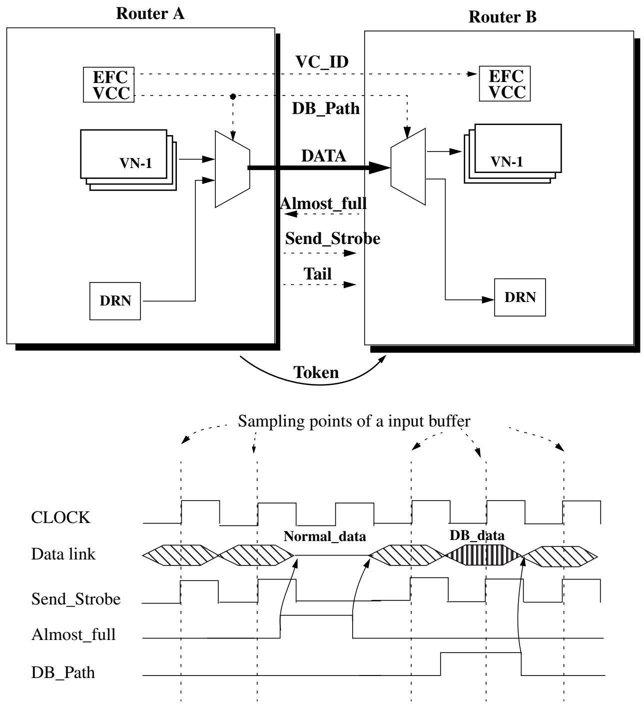

Figure 8. External Flow Control Signals and their Timing Diagram.

dates, and regulates internal flow of flits across the crossbar via handshaking with the output module and deadlock module. A block diagram is shown in Figure 6.

For simplicity of address decoding and updating, the *WARRP* router uses relative addressing in which the hop distance from the current node and the destination node is specified by a 3 bit X-offset and 3 bit Y-offset in the header. The header format is shown in Figure 7 and includes destination addressing information and two reserved bits. Each dimension offset field represents a distance range of -4 to 3 for both positive and negative direction routing across the bidirectional channels. The two reserved bits are used for extending the functionality of the *WARRP* router. The following functions are encoded: 00 = normal packet, 01 = deadlock packet eligibility, 10 = reserved, 11 = reserved. As mentioned in Section 2.2, the *WARRP* router provides selectivity in determining which packets may use the recovery lane. This is implemented by setting the first reserved bit if a packet makes a westward turn, a dateline route, or is injected as a priority packet.

True fully adaptive routing permits a broad number of routing choices. However, previous studies have indicated that allowing the input module to make multiple output link requests does not significantly improve performance [14, 15]. This is primarily due to the fact that the probability of multiple headers residing in the same router and in the same virtual network is usually very low, obviating the need for this parallelism. Hence, each input module is allowed to request at most one of several permissible output links each cycle, although multiple input modules may make requests to the routing and arbitration module in parallel. This design choice reduces the path set-up delay and simplifies the address decoder by removing the components needed to select one from a number of possible acknowledged requests. Moreover, to avoid unnecessary requests for currently occupied output ports, status information is fed back from the routing and arbitration module to the address decoder. A routing decision is made within the same clock cycle of the request. If the request is rejected, the address decoder regenerates a request signal for another appropriate output link the next successive cycle until a request is granted.

3.4 Routing and Arbitration Module

In the *WARRP* router, each virtual network has an associated crossbar routing and arbitration module. This allows routing and arbitration decisions to be made in parallel across virtual networks, and it simplifies the arbitration logic. Also, at most only a single request can be made by each input module. Hence, in the worst case, arbitration among six requests must be handled by each routing and arbitration module instead of as many as thirteen requests if the internal router architecture was not partitioned.

Arbitration is based on round-robin priority to guarantee freedom from starvation. An arbitration decision for resources in the normal routing section can be made within the same cycle of the request. However, access to resources in the deadlock routing section requires an additional cycle as explained below.

In the first cycle, arbitration among the requesting input modules within each virtual network is performed so that one input module in each network is selected. In the second cycle, the deadlock module's arbitration unit arbitrates among the possible requests by the three virtual networks so that only one input module in the router is allowed access to the recovery lane. The arbitration module in the selected virtual network configures the crossbar to connect the granted input module to the deadlock module port. This two-cycle arbitration delay for the recovery lane does not degrade router performance since recovery lane access delay is dominated by token capture (see Section 3.6) which serves as a "time-out" mechanism.

3.5 Flow Controller Module

The *WARRP* router uses a source-synchronous signaling scheme [16, 17] to control asynchronous flit flow externally

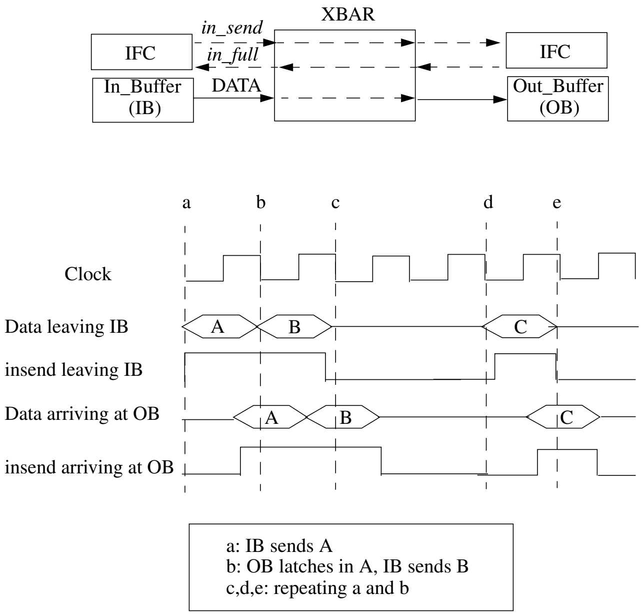

Figure 9. Internal Flow Control Signals and their Timing Diagram.

between routers. Two external flow control signals are required as shown in Figure 8. The *Send_Strobe* pulse is sent along with each packet flit to indicate that the receiving input module is to latch in the data. Buffers in the input module are used to re-synchronize incoming data with the internal clock of the router to obviate the need for additional PLL/DLL to maintain phase-synchronization between routers. To fully utilize link bandwidth, an *Almost_Full* handshaking signal controls the flow of flits between routers. This signal reflects the status of the input module's input buffer and is asserted once the buffer nears capacity. Enough buffer space is provided in the *WARRP* router to allow for round-trip propagation delay of flow control signals as multiple flits may be in flight before these signals are detected.

Two additional external control signals are required to establish the path between the output buffer of one router and the corresponding input buffer of the next router as shown in Figure 8. The *VC_ID* signal identifies to which virtual network the flit should be transmitted, and the *DB_Path* signal designates that the flit should be transmitted to the recovery lane. This signal is asserted by the router's external flow controller, possibly preempting physical channel bandwidth from normal packets. These signals are encoded in two bits (three virtual channel buffers and one deadlock buffer). The *tail* and *token* are additional control lines implemented by the router.

Internal flow between input buffers and output buffers within the *WARRP* router is controlled by the *in_send* and *in_full* control signals. As shown in Figure 9, these signals are used to represent the status of virtual network buffers to send and receive data. Internal flow is synchronized with the router

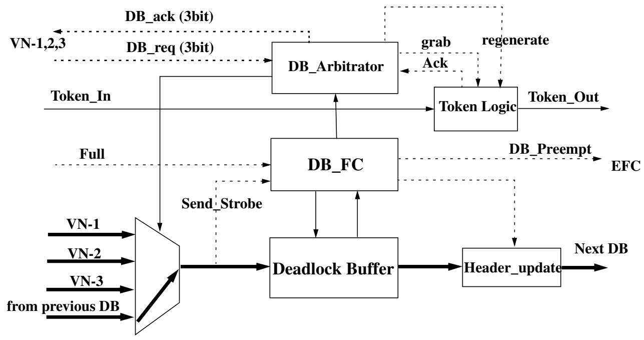

Figure 10. Block Diagram of Deadlock Module.

clock to simplify router design and to reduce synchronization overhead. The in_full signal is also used to generate the backpressure to stop internal data flow when the output buffer is preempted by the deadlock buffer.

3.6 Deadlock Module

The deadlock module resolves impending deadlock by providing a recovery lane for potential deadlock packets. A block diagram of this module is shown in Figure 10.

For the recovery lane to be free from deadlock, mutual exclusive access to the deadlock module is implemented. This is maintained by the *DB_Arbitrator* in the deadlock module and an external token that circulates among all the routers. When an eligible packet tries to access the deadlock module due to excessive blockage on normal buffer resources, two different types of arbitration for the deadlock buffer resource occurs: arbitration among eligible packets in the current router and arbitration among other routers which may have eligible deadlock packets. The first is performed by the DB_Arbitrator, as described in Section 3.4. The second is accomplished by having the eligible packet capture a circulating token. Once a packet is granted access to the recovery lane, the token stops circulating until the deadlock packet reaches its destination where the token is regenerated.

A deadlock packet partially buffered in a router's deadlock module is transmitted to the deadlock module of a neighboring router by preempting the bandwidth of the physical link along any minimal path toward the destination. The preemption operation is initiated by asserting a control signal, *DB_Preempt*, to the external flow controller of the preempted physical link. To maximize bandwidth utilization of links, preemption stops until more data flits are ready to be transmitted by the deadlock module if a bubble occurs in deadlock packet transmission.

4 Router Implementation

The *WARRP* router architecture described in this paper was designed at the behavior level using a VHDL hardware description language and implemented in RAM-based FPGA technology (see [19] for a fully detailed specification of the router design). This allowed us to quickly simulate and verify our design at the gate and architecture levels to further optimize the design and estimate performance prior to fabrication in optoelectronic technology. Preliminary simulations indicate that the router can be clock at approximately 25 MHz using this technology.

The *WARRP* router is also being implemented in CMOS-SEED optoelectronic technology which allows the integration of CMOS VLSI logic circuitry with dense surface-normal, high-speed optoelectronic transceivers (AlGaAs-based multiple-quantum-well self-electrooptic-effect devices or SEEDs) via flip-chip bonding. SEEDs are passive diode devices which change their light absorption when an applied electric field is varied across the device terminals. This results in modulation of an off-chip source controlled by an electrical signal for its use as an optical transmitter. The same device can also be biased to work as a receiver that generates a leakage current upon receiving light, thus converting an optical signal back to an electrical signal. These devices organized as an array of 20×10 pixels are flip-chip bonded on top of the $2mm \times 2mm$ router circuitry to allow a total of 200 optical I/Os (only some of which are used). Each diode is $20 \times 60 \mu m^2$ with a separation of $62.5\mu m$ and $125\mu m$ from neighboring devices. Hence, this technology is capable of supporting approximately 5,000 optical I/Os per cm^2 with each I/O capable of modulating/receiving data at more than 2 Gb/s [18] using only $300\mu W$ of optical power, for an aggregate capacity on the order order of 100s of GBytes per sec per cm^2 (even with differential signal mode assumed). Network routers can benefit from this high bandwidth I/O capability in several ways including realizing implementations with wider channels (data and control), higher node degree, full duplex links, redundant channels (for fault-tolerance), etc., to reduce latency and increase throughput.

In previous work [20], *WARRP*'s core deadlock handling mechanisms were implemented in monolithic GaAs MESFET VLSI technology integrated with GaAs OPFET photodetectors and GaAs/InGaP double heterostructure LEDs. Here, we implement a fully-functional router. However, due to limited chip real-estate (only $2mm \times 2mm$), a scaled-down CMOS-SEED optoelectronic version of the *WARRP* router is implemented. This version, referred to as *WARRP II*, implements a unidirectional 4-bit wide 1-D torus topology with one virtual channel network and associated deadlock recovery mechanisms. It is implemented using $0.5\mu m$ (with $0.7\mu m$ drawn gate length), 3-metal layer, 3.3V CMOS, and it contains approximately 26,000 transistors, of which 3,000 are used for transceiver, testing, and pad driver circuits. Only 60% of the chip area ($1.2mm \times 1.2mm$) is allocated for the router datapath logic which consists of 4-flit-deep input buffers, 3-flit-deep output buffers, address decoder, 2×3 subcrossbar, crossbar arbitrator, and deadlock module (i.e., deadlock buffer and its associated flow controller and channel preemption logic). The remaining 40% of the chip area is allocated for optical

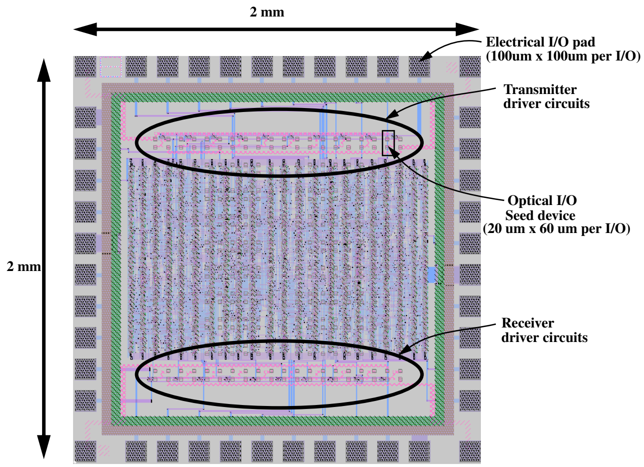

Figure 11. The VLSI Layout of the WARRP II Router.

I/O transceiver circuits, testing circuits, bonding pads, etc. It is packaged in a 44-pin PGA which uses only 22 electrical signals and only 36 SEED devices (18 dual-rail optical signals). A layout of the chip is shown in Figure 11.

5 Performance

Figure 12 gives the pin-to-pin latency estimation for the two router implementations and the various components composing the latency. The router clock cycle time is determined by the maximum of two major router (pipeline) stages in which flits may be operated upon: routing (only header flits) and internal flow-thru. The first stage consists of address decode, header update, routing arbitration and crossbar set-up time. The second stage consists of internal router flow-thru time (each virtual network crossbar traversal adds one clock cycle for header flits only). For both implementations, internal flow-thru dominates.

The optoelectronic WARRP II implementation is about the same speed as the FPGA implementation of the 2-D 3-virtual channel version of the WARRP router. This is due to design constraints in the WARRP II rather than technological limitations. For one, only two metal layers were available for gate interconnects as the third metal layer was reserved for wiring between drivers and optical I/Os. This significantly increased the average net length between standard cells as well as the size of their drivers. Second, an automated synthesis tool (EPOCH) was used to generate the VLSI layout of the WARRP II router as opposed to using a customized layout design tool. Furthermore, many of the optical I/Os were not used due to the scaled down router design. Altogether, these design constraints limit the WARRP II router to operate at 25MHz and yield an aggregate bandwidth of 25MB/s, well below the capabilities of this technology.

Nevertheless, WARRP II is the first fully-functional opto-

	FPGA (2D,3VCs)	OPTO (1D,1VC)
Stage for Routing	35ns	35ns
Stage for Internal data flow	39ns	40ns
Max Clock Speed	25MHz	25Mhz

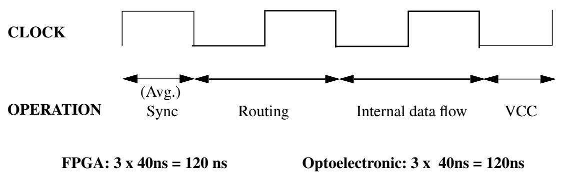

Figure 12. Pin-to-Pin Latency for the two router implementations and its components.

electronic network router chip to demonstrate that near-term optoelectronic technology is feasible for the design and implementation of advanced network routers requiring dense high-bandwidth connectivity.

Acknowledgements

We acknowledge the assistance of David Brooks, Nuwan Jayasena, and Sangeeta Mudnal in helping with design implementation and verification.

References

[1] Bolding, K. “Non-Uniformities Introduced by Virtual Channel Deadlock Prevention”, *Technical Report UW-CSE-92-07-07*, University of Washington, Seattle, WA, July 1992.

[2] Yungho Choi and Timothy Mark Pinkston. “Crossbar Analysis for Optimal Deadlock Recovery Router Architecture”. In *Proceedings of the 10th International Parallel Processing Symposium*, pages 583-588, IEEE Computer Society, April 1997.

[3] Mike Galles. “Scalable Pipelined Interconnect for Distributed Endpoint Routing: The SGI SPIDER Chip”. *Proceedings of the Symposium on Hot Interconnects*, pages 141-146, IEEE Computer Society, August 1996.

[4] Steven L. Scott and Gregory M. Thorson. “The Cray T3E Network: Adaptive Routing in a High Performance 3D Torus”. *Proceedings of the Symposium on Hot Interconnects*, pages 147-156, IEEE Computer Society, August 1996.

[5] Timothy Mark Pinkston and Sugath Warnakulasuriya. “On Deadlocks in Interconnection Networks”. In *Proceedings of the 24th International Symposium on Computer Architecture*, pages 39-49, IEEE Computer Society, April 1997.

[6] Timothy Mark Pinkston, “Flexible and Efficient Routing Based on Progressive Deadlock Recovery”, Submitted to IEEE Transactions on Computers.

[7] K.W.Goossen, et al. “GaAs MQW modulators integrated with silicon CMOS”. *IEEE Photonic Technology Letters.*, 7:360-362, 1995.

[8] J. Kim, Z. Liu, and A. Chien. “Compressionless Routing: A Framework for Adaptive and Fault-tolerant Routing”. In *Proceedings of the 21st International Symposium on Computer Architecture*,IEEE Computer Society, pages 289-300, April 1994.

[9] Anjan K. V. and Timothy Mark Pinkston. “DISHA: A Deadlock Recovery Scheme for Fully Adaptive Routing”. In *Proceedings of the 22nd International Symposium on Computer Architecture*, pages 201-210, IEEE Computer Society, June 1995.

[10] Anjan K.V. and Timothy M. Pinkston, DISHA: A Deadlock Recovery Scheme for Fully Adaptive Routing, In *Proceedings of The 9th International Parallel Processing Symposium*, pages 537-543, Santa Barbara, CA, April 1995.

[11] W. Dally and C. Seitz. “Deadlock-free Message Routing in Multiprocessor Interconnection Networks”. *IEEE Transactions on Computers*, 36(5):547-553, May 1987.

[12] J. Duato. “A New Theory of Deadlock-free Adaptive Routing in Wormhole Networks”. *IEEE Transactions on Parallel and Distributed Systems*, 4(12):1320, 1331 1993.

[13] C. Glass and L. Ni, “The turn model for adaptive routing”, in *Proceedings of the 19th Int’l Symposium on Computer Architecture*, pages 278-287, May 1992.

[14] Kevin Bolding. “Chaotic Routing - Design and Implementation of an Adaptive Multicomputer Network Router”. *Ph.D Thesis, Department of Computer Science and Engineering, University of Washington.* July 1993.

[15] Yungho Choi and Timothy Mark Pinkston. “Router Design Analysis for Optimal Deadlock Recovery Router Architecture”. CENG Technical Report, Department of Electrical Engineering - Systems, University of Southern California, Los Angeles, CA 90089-2562, September 1996.

[16] Andrew A. Chien and J. H. Kim. “Planar-Adaptive Routing: Low-Cost Adaptive Networks for Multiprocessors”. In *Proceedings of the 19th Symposium on Computer Architecture*, pages 268-277. IEEE Computer Society, May 1992.

[17] Joseph Carbonaro, Frank Verhoorn. "Cavallino: The Teraflops Router and NIC". *Hot Interconnects IV Symposium Record*, pages 157-160, August 1996.

[18] T. K. Woodward, A. L. Lentine, K. W. Goossen, J. A. Walker, B. T. Tseng, S. P. Hui, J. Lothian, R. E. Leibenguth, Demultiplexing 2.48 Gb/s Optical Signals with a Lower-Speed Clocked-Sense-Amplifier- Based Hybrid CMOS/MQW Receiver Array, In *Spatial Light Modulators Technical Digest*, 1997 Spring Topical Meeting, pages 52-54, March 1997.

[19] Yungho Choi and Timothy Mark Pinkston. "The Architecture of the WARRP router". CENG Technical Report, Department of Electrical Engineering - Systems, University of Southern California, Los Angeles, CA 90089-2562, August 1997.

[20] Timothy Mark Pinkston, Mongkol Raksapatcharawong, and Yungho Choi. "Smart-Pixel Implementation of Network Router Deadlock Handling Mechanisms". In *Optics in Computing Technical Digest*, Optical Society of America, pages 159-161, March 1997.