Chapter Nine
Summary and Conclusions

The potential for large multiprocessors and multicomputers to achieve high performance can only be realized if they are provided with a high-throughput low-latency communication network. Fast small $n \times n$ switches are critical components for achieving high-speed communication. The organization of the buffers in the $n \times n$ switches is one of the most important factors in determining their performance.

The architecture of $n \times n$ switches should allow them to efficiently utilize their buffer memory as well as the raw bandwidth of their ports. The architecture is constrained by the requirement that it must be amenable to high-performance cost-effective VLSI implementation. Since both the datapath and control of the switch must operate at high clock rates, complexity must be limited. The primary contribution of this dissertation is the architecture of a new packet buffer, the *dynamically allocated multi-queue* buffer, for use in $n \times n$ switches. This buffer supports forwarding of packets in non-FIFO order and provides efficient handling of variable length packets. A critical advantage of DAMQ buffers is that flow control is simpler than with other multi-queue buffers and there is no need to ‘‘pre-route’’ packets as with the other multi-queue buffers.

We have described the micro architecture of a DAMQ buffer and its controller in the context of the ComCoBB communication coprocessor for multicomputers. We have also presented the DAMQ Buffer Chip, a stand-alone implementation of the DAMQ buffer appropriate for use as the building block of a

multi-chip switch. We demonstrated that the DAMQ buffer can be efficiently implemented in VLSI to support packet transmission and reception at the rate of one byte per clock cycle with high clock rates. With a ‘‘hardwired’’ linked list manager and a fast routing mechanism, the buffer supports virtual cut-through of messages with a latency of four cycles.

We have evaluated the DAMQ buffer by comparing its performance with that of three alternative practical buffers in the context of a synchronous store-and-forward multistage interconnection network. Both discarding and blocking switches were considered. The DAMQ buffer provides two key features: non-FIFO handling of packets and dynamic partitioning of buffer storage. For uniform traffic, our modeling and simulations show that these features result in large performance improvements over conventional FIFO buffers. The DAMQ buffer also provides a higher maximum throughput than do other practical non-FIFO buffer organizations with the same total buffer storage capacity.

In order to use large multiprocessors and multicomputers for real-time applications, it must be possible to guarantee, with a high degree of confidence, that high-priority traffic can be transmitted through the network with low specified latency. With conventional interconnection networks used in multiprocessors and multicomputers, the worst-case latency of traffic through the network can increase dramatically as the load on the network increases. This may prevent the use of these interconnection networks for critical real-time applications or force their use with very low utilization (and thus high cost/performance ratio) in order to guarantee low maximum latency.

For interconnection networks composed of small $n \times n$ switches, we have

shown that simply increasing the size of conventional buffers in the switches does not result in improved performance for high priority packets. There is thus a fundamental need for buffer organizations which support high-priority packets. We have developed a technique for efficiently supporting high-priority traffic, while maintaining good performance for normal traffic. The modifications to the DAMQ buffer in order to support high-priority traffic involve a few additional control registers and somewhat more complex arbitration of the crossbar switch. Overall these modifications are expected to require only a small percentage increase in total buffer area.

We have evaluated alternative approaches to providing support for high-priority packets in $n \times n$ switches. This evaluation was based on implementation complexity and simulation studies of a multistage interconnection network transmitting packets with two levels of priority. Our simulations have demonstrated five key points. (1) In a conventional network with a single priority level for all packets, worst-case latency for packets can be several times higher than average latency. Hence, there is a need to identify and provide preferential treatment to those packets for which fast service is particularly important. (2) Using the priority of packets as the determining factor in arbitrating contention within each switch does not provide sufficient support for high-priority traffic. Such arbitration does not reduce the 99^{th} percentile latency of the high-priority packets to the level of average normal packet latency, even under moderate network load. (3) As long as the proportion of high-priority traffic is low, dedicated queues for high priority traffic reduce the 99^{th} percentile latency of the high-priority packets to the level of the average latency for normal traffic under a

wide range of network throughputs. (4) When the proportion of high-priority traffic is large and the network is heavily loaded, multiple dedicated queues for high-priority traffic can reduce the 99^{th} percentile latency relative to the single dedicated queue approach. However, this performance advantage is marginal and the associated implementation cost is very high. (5) Dedicated buffers for the high-priority traffic can provide the same performance advantage as dedicated queues in shared DAMQ buffers. However, since DAMQ buffers are needed to maximize performance for normal traffic, the implementation cost for dedicated buffers is much higher than the cost of adding dedicated queues to DAMQ buffers. Hence, there is no reason to use dedicated buffers.

Our results indicate that the DAMQ buffer with a single dedicated queue for high-priority packets provides support for high-priority traffic which is superior to the support provided by alternate switch designs based on a dedicated high-priority queue. This resulting network performance, with respect to high-priority traffic, is very close to the performance of a network based on ‘‘ideal’’ switches for which a practical implementation is not feasible. Hence, given the low hardware overhead of the scheme based on a DAMQ buffer with a single high-priority queue, it is clearly the preferable option.

The DAMQ buffer architecture was designed to support variable-length packets and virtual cut-through — features which are not present under synchronous transmission protocols. In Ch. 6, we demonstrated the value of non-FIFO buffer and dynamic buffer allocation in asynchronous communication networks. We showed that, under uniform traffic conditions, for both constant and variable-length packets, the DAMQ buffer supports a higher saturation network

throughput over multistage interconnection networks than do FIFO buffers or statically allocated non-FIFO buffers. For constant-length packets, we showed that networks operating at saturation throughput *self-synchronize.* For a network of FIFO or DAMQ switches, this results in network behavior similar to that of a synchronous network — the number of connections that can be made during each crossbar arbitration is maximized and wasted network bandwidth is minimized. For SAMQ and SAFC switches (multi-queue buffers whose memory is statically allocated among their queues), the presence of virtual cut through reduces the impact of their inefficient buffer utilization, resulting in a slight improvement in their performance relative to the synchronous protocol. With variable-length packets, however, the statically allocated buffers suffered a significant reduction in their performance due to the amount of buffer memory lost to fragmentation. The DAMQ buffer, by virtue of its dynamic buffer allocation, experienced minimal fragmentation and maintained its high performance.

Chapter 6 also presented the use of the DAMQ buffer in two-dimensional torus networks. It is shown how the deadlock-free dimensional routing algorithm presented in [Dall87a] can be implemented using multi-queue buffers. The key factor of this implementation is that no single queue within a buffer be allowed to occupy the entire buffer space. Statically allocated buffers (and the Torus Routing Chip [Dall86]) accomplish this by physically separating the buffer memory associated with each queue. The DAMQ buffer prevents individual queues from occupying an entire buffer via maximum usage flow control. The simulation results for 11×11 and 21×21 torus networks indicate that the DAMQ buffer supports a higher saturation throughput than do statically allocated multi-queue

buffers, for small buffer sizes. Further, the throughput of buffered torus networks is shown to be ≈ 100% higher than that of networks with wormhole routing and minimal buffering [Dall90b]. We also show that buffered torus networks are highly reactive (in contrast to unbuffered torus networks[Dall90b]). This is due to the fact that congestion can propagate around the ring within a given dimension, creating a positive-feedback loop which causes the network throughput to drop dramatically when the applied load exceeds the point of maximum network throughput.

The DAMQ Chip, presented in Ch. 7, is a VLSI implementation of a stand-alone version of the DAMQ buffer. As an interface chip in the node of a multicomputer, it asynchronously receives packets from and transmits packets to other DAMQ Chips. As an example implementation of the DAMQ buffer, it demonstrates that a non-FIFO buffer can be efficiently implemented in VLSI. We presented details of the implementation, including the floorplan and critical path timing.

We have demonstrated that the DAMQ buffer can operate at high clock rates, despite its complex control. Hence, other factors, such as the buffer memory or the inter-chip links, will limit the raw bandwidth. For a DAMQ buffer with four queues and a packet size of 32 bytes, it was shown that the DAMQ buffer will have one less packet slot available to it than a FIFO buffer occupying approximately the same chip area. As was shown in Chs. 4 and 6, with as few as two packet slots (64 bytes), a network with DAMQ buffers can outperform a FIFO buffer network which has an additional packet slot. With three packet slots, the DAMQ buffer network will saturate at a throughput approximately 23% higher than a FIFO buffer

network with four packet slots. The DAMQ buffer is thus shown to be a highly effective building block for packet switches, demonstrating that for VLSI switches, increased control complexity may lead to higher performance.

Chapter 8 examines the performance of hop-level flow control mechanisms for scalable multicomputers. A significant contribution of this chapter is our methodology for measuring the relative performance of flow control mechanisms via simulation. A congestion benchmark suite was developed, where each benchmark in the suite creates a different pattern of congestion using stochastically-generated communication traffic. The benchmark suite was chosen to ‘‘stress’’ the flow control mechanisms in a number of different ways such that any flow control which preserves the network performance through all of the benchmarks is likely to perform well under the conditions which occur in a general-purpose multicomputer computer.

The chapter contains the description and analysis of a number of hop-level flow control mechanisms. Their performance is evaluated via the congestion benchmark suite described above. We also examined the complexity of their implementation. There was a single flow control / buffer architecture combination whose performance was superior to all other switching architectures examined in this dissertation for every benchmark in the congestion suite — destination-based flow control with statically allocated, fully-connected (SAFC) buffers. SAFC buffers are multi-queue buffers in which the buffer memory is statically partitioned among the queues, and each queue has its own read port. This allows multiple packets to be transmitted from a single buffer to different output ports simultaneously. While this superior connectivity gives SAFC buffers the potential

to support a higher throughput than DAMQ buffers, the results presented in Chs. 4 and 6 suggest that DAMQ buffers offer higher performance for moderately-sized buffers. However, when SAFC buffering is combined with destination-based flow control, a significant improvement in network performance results. Even under uniform traffic conditions, SAFC buffers with destination-based flow control supports a higher maximum throughput and a lower average latency for given throughputs than do DAMQ buffers with blocking, destination-based, or any other flow control mechanism that we examined.