

This document is an author-formatted work. The definitive version for citation appears as:

R. F. DeMara and P. J. Wilder, "A Taxonomy of High Performance Computer Architectures for Uniform Treatment of Multiprocessor Designs," in *Proceedings of the 1999 American Association for Engineering Education Southeastern (ASEE-SE'99) Conference*, pp. 1 – 9, Clemson, North Carolina, U.S.A., April 11 – 13, 1999.

A Taxonomy of High Performance Computer Architectures for Uniform Treatment of Multiprocessor Designs

Ronald F. DeMara¹ and Paul J. Wilder²

Abstract

Seven distinct configurations of shared-memory multiprocessors are defined and parameterized in terms of the number of independent interface ports to the global shared data (P), the number of ports per memory device (D), and the number of ports supporting independent read accesses (R) and write accesses (W). The class of *B-symmetric* architectures containing N processors exhibit port utilization proportional to the quantity N/P and require physical memory capacity of size M , where M denotes the address range of the global shared-memory space. This class includes the Single-Port through N-Port memory strategies. The class of *U-symmetric* architectures includes the Allocated Dual-Port, the Concurrent Read Replicated, and the Read-Time Resolution Coherent memory strategies. U-symmetric architectures decrease port contention by supporting up to $R+W$ concurrent memory transactions per cycle. However, they require physical memory capacities ranging from $N \cdot M$ up to $M \cdot N^2$. This taxonomy has been useful in conceptualizing a large number of machine designs in terms of a unified model requiring relatively few independent parameters. This enables incremental understanding of multiprocessor architectures by progressing through different architectural classifications in a pre-defined sequence of increasing complexity.

1.0 Overview

A fundamental problem when presenting course material on high performance computer architectures to both advanced undergraduate and beginning graduate students is the wide range of sophisticated machine designs and interconnection topologies, which must be introduced. Each architecture has its own characteristics and components which must be explained in sufficient detail in order for the students to grasp the relative benefits and understand the operational characteristics

¹ Dept. of Electrical and Computer Engr., Univ. of Central Florida, Orlando FL 32816-2450.
E-mail: demara@pegasus.cc.ucf.edu

² Univ. of Southern Mississippi, Gulf Coast Campus, PO Box 850, Gautier, MS 39553.
Email: paul.wilder@usm.edu

of each design. This makes it difficult to cover a wide enough range of the processor designs currently in use while still allowing the students to comprehend how the different approaches compare to one another. The taxonomy of computer architectures presented in this paper that was developed at the University of Central Florida was found to be useful in allowing coverage of the topic material within a unified framework.

2.0 Nomenclature

To precisely describe the capability for simultaneous access of a shared-memory multiprocessor, we propose a new taxonomy for shared-memory architectures: *B-symmetric*, *U-symmetric*, and *nonsymmetric*. A *symmetric* multiprocessor has the property that the shared memory access capabilities of all processors are identical. The prefixes defined in this paper describe the input/output capabilities of all access ports as either bi-directional, unidirectional, or some combination of both.

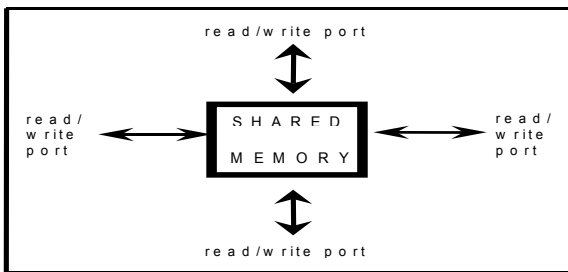


Figure 2.1. B-symmetric Architecture

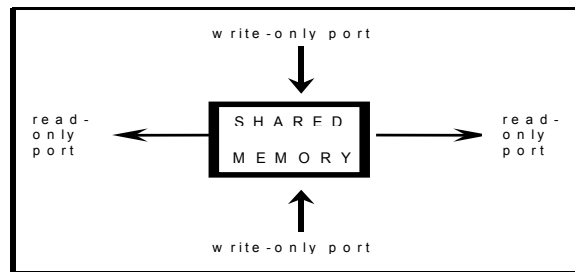


Figure 2.2. U-symmetric Architecture

More formally, a shared-memory system is said to be *bi-directional symmetric (B-symmetric)* if the total number of independent access ports to the shared data, denoted by P , is related to the number of read-capable access ports, R , and the number of write-capable access ports, W , such that $P = R = W$. A shared-memory system is said to be *Unidirectional Symmetric (U-symmetric)* if $P = R + W$. Examples of B-symmetric and U-symmetric architectures are shown in Figure 2.1 and Figure 2.2, respectively. Non-symmetric architectures are not treated explicitly in this paper as they can be represented as the superposition of one or more B-symmetric and U-symmetric configurations.

3.0 Notation and Configuration Parameters

In order to characterize a shared-memory system, the number of device ports per memory module, denoted by D , is necessary in addition to quantities R , W , P , and the characterization of access symmetry defined above. The total number of accessible ports to the shared memory is at least equal to the number of ports per device, the number of read ports, and/or write ports. Hence, the relationship $1 \leq D$, $R, W \leq P$ holds for all architectures, regardless of interconnection topology. To understand the relationship between these quantities, first define M_R and M_W as the multiplicity of read ports and write ports to the shared space, respectively. In general, M_R is given by R divided by the number of read ports per physical memory device, and an analogous relationship holds for M_W . If there is no redundancy in the use of the physical memory, then $M_R = M_W = 1$. When M_R or M_W exceed unity then there are dedicated access ports for one or more processors, or other redundant communication channels have been provided to the shared memory. These two cases impose a relationship on P such that $\max(R, W) \leq P \leq R + W$. To label multiprocessor architectures according to these parameters it is sufficient to indicate R , W , D , and the symmetry classification denoted as $C \in \{B, U\}$ for B-symmetric or U-symmetric configurations, respectively. We adopt the notation $C_D^{R,W}$ to refer to each

of the seven relevant classifications described below. For example, the architecture labeled $U_4^{8,3}$ refers to the U-symmetric architecture with $R = 8$ read ports, $W = 3$ write ports, and $D = 4$ device ports.

4.0 Multiprocessor Classifications

The following terms are used to describe the performance properties of various multiprocessor configurations. The *bandwidth*, in words per memory cycle, of a shared-memory system is a measure of the capacity to perform concurrent accesses by multiple CPUs. The *likelihood of access contention* indicates the potential for concurrent accesses to incur delays for one or more CPUs, thus reducing the delivered bandwidth below the requested bandwidth. The *contention penalty*, in memory cycles, quantifies this degradation in performance. The *contention resolution complexity* is a measure of the complexity of the techniques used to reconcile access conflicts using special-purpose hardware or protocols. The *wiring complexity* reflects the wire count required to interconnect the processors within the shared-memory system. Note that all of the following architectures may be enhanced with the addition of private memories for local data, as well as cache memories in which global data is selectively replicated. These enhancements improve throughput, but impose the requirement to maintain memory coherence.

4.1 Single-Port Memory Strategy - $B_1^{1,1}$

The architecture shown in Figure 4.1 is referred to as the $B_1^{1,1}$ configuration. It is a B-symmetric, uniplex read, uniplex write multiprocessor employing a memory module with one device port. This system supports a single memory transaction to the shared data at a time because of the time-multiplexed bus interconnection from the processors to the global memory. A memory access by any processor inhibits simultaneous memory accesses from any other processor. All inhibited processors are stalled until the bus becomes available.

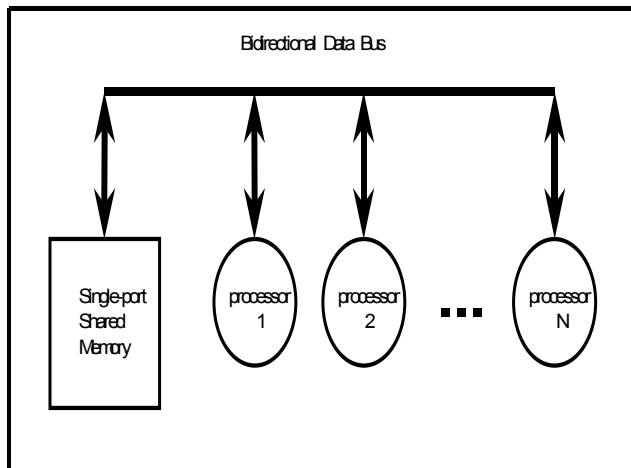


Figure 4.1. $B_1^{1,1}$ Multiprocessor System

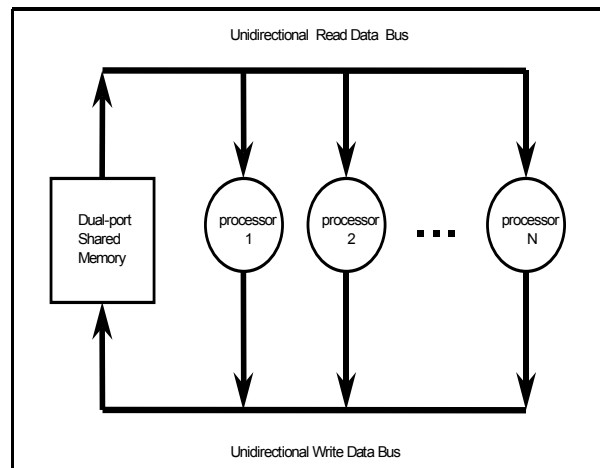


Figure 4.2. $U_2^{1,1}$ Multiprocessor System

The $B_1^{1,1}$ architecture is straightforward to implement because it requires a single bi-directional bus and uses memory modules with low design complexity and cost. The number of transactions is limited to one read or one write so concurrency may be restricted. Therefore, the likelihood of having either read or write access contention is extremely high. The resolution of memory contention is performed by the bus access logic that is provided at each CPU interface.

4.2 Allocated Dual-Port Memory Strategy - $U_2^{1,1}$

The $U_2^{1,1}$ architecture shown in Figure 4.2 is a U-symmetric, uniplex read, uniplex write shared-memory system using a dual-port memory. It supports one read transaction concurrently with a single write transaction. The most fundamental implementation of this configuration would utilize unidirectional buses. The dual-port memory device thus supports concurrent access for a single write and a single read by any processor. This architecture improves the previous configuration by allowing exactly one read to occur concurrently with a single write.

4.3 Multi-mode Dual-Port Memory Strategy - $B_2^{2,2}$

Figure 4.3 shows a B-symmetric, duplex read, duplex write multiprocessor denoted $B_2^{2,2}$. It allows two read or two write accesses, or a combination of up to two reads and writes. This strategy utilizes two bi-directional buses connecting all processors to a dual-ported memory module. Control of the assignment of pending read or write operations to an available bus is done dynamically. Since many applications perform significantly more read transactions than write transactions, these buses allow an effective doubling of read access bandwidth which may justify the complexity over a $U_2^{1,1}$ architecture.

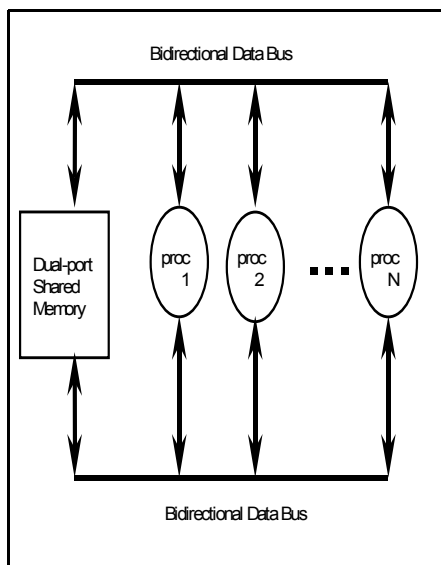


Figure 4.3. $B_2^{2,2}$ Architecture

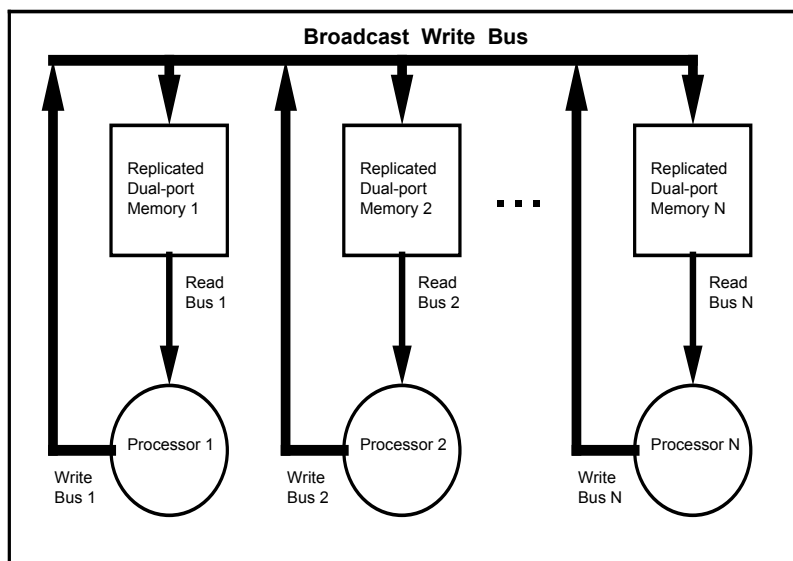


Figure 4.4. Replicated Memory $U_2^{N,1}$ Architecture

4.4 Concurrent Read Replicated Memory Strategy - $U_2^{N,1}$

The Concurrent Read Replicated Memory strategy is shown in Figure 4.4. The characterization of this shared-memory architecture is $U_2^{N,1}$. It is a U-symmetric, N -plex read, uniplex write shared-memory system using replicated dual-port memories. It allows multiple concurrent read accesses due to the use of redundant memory modules and private read buses for each processor. This strategy utilizes a unidirectional broadcast bus to write all updates into the dual-port memories that are replicated for each processor. The shared memory system allows a single access for write transactions and independent of the concurrent read transactions. Hence, only write transactions can

incur memory contention. This architecture improves upon the $B_2^{2,2}$ characterization by allowing N independent concurrent read transactions. It should be noted that the composite shared-memory is effectively an $N+1$ port memory, but is implemented using only dual-ported modules independent of the size of N .

4.5. Extended Concurrent Read Replicated Memory Strategy: $U_D^{N, D-1}$

The memory replication concept can be extended for applications with higher write ratios by utilizing more ports per memory module, as in the $U_D^{N, D-1}$ architecture shown in Figure 4.5. It is a U-symmetric, N -plex read, $(D-1)$ -plex write system, where D denotes the number of ports per memory device.

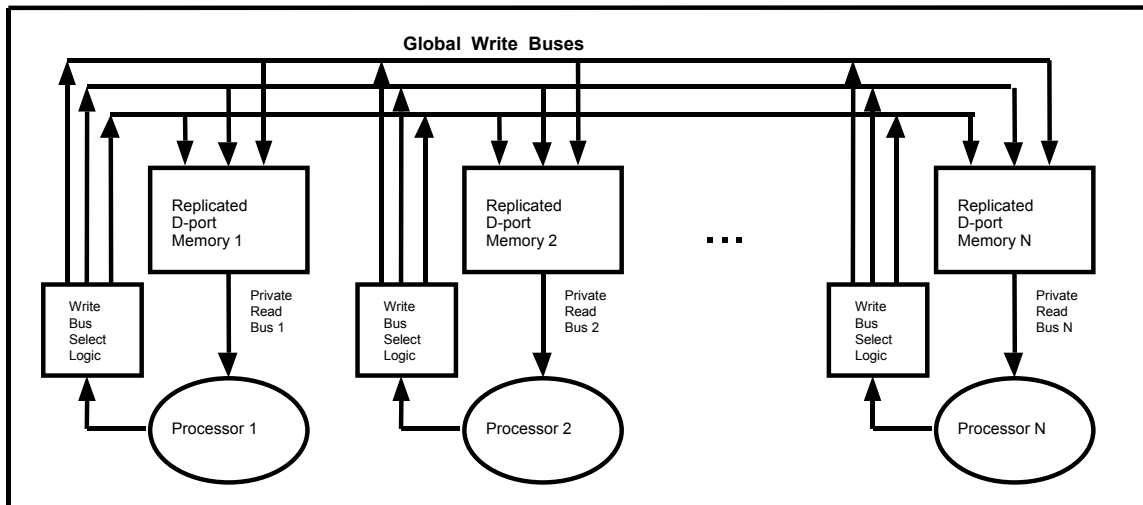


Figure 4.5. Extended Concurrent Read Replicated Memory $U_D^{N, D-1}$ Multiprocessor System

This strategy utilizes $D-1$ unidirectional write buses supporting write transactions to a D -ported memory replicated N times. All N processors have a private read bus that allows all read transactions to be concurrent. The shared memory allows $D-1$ concurrent accesses for write transactions and independent concurrent read transactions by all processors. So only memory write transactions with degree greater than $D-1$ can impact the processor utilization. The aggregate shared-memory system is effectively an $(N+D-1)$ -port memory. Any simultaneous write accesses greater than $D-1$ will inhibit write access to the remaining processors. This places a burden on the system designer to choose D correctly based on a-priori knowledge of the memory access characteristics of the target workloads. The resolution of write conflicts is moderately complex. Bus arbitration and idle bus selection logic is required to allocate and de-allocate write buses to any processor requesting write access. The wiring complexity is proportional to $(N+D-1)$. The memory is replicated N times, therefore its size is proportional to the product mN . The memory devices are special-purpose multi-port memory modules. For values of $D > 4$, the components become prohibitively expensive or are currently not feasible to construct with sufficiently large storage capacities. Also, as D increases, the memory capacity per device decreases and thus greater numbers of the devices are required to implement this architecture.

4.6 Read-Time Resolution Data-Coherent Memory Strategy - $U_2^{N,N}$

The Read-Time Resolution Coherent Memory multiprocessor is shown in Figure 4.6. The characterization of this shared-memory architecture is $U_2^{N,N}$. It is a U-symmetric, N -plex read, N -plex write shared-memory system. This strategy utilizes N unidirectional write buses supporting write transactions to a two-ported memory replicated N^2 times. All N processors have a private read bus that allows all read transactions to be concurrent. The memory system allows independent concurrent accesses for write transactions and independent concurrent read transactions by all processors. This strategy operates by bank-selecting the most recently written replicated memory bank with respect to the word of memory being read. This architecture improves upon the $U_D^{N,D-1}$ characterization by allowing N independent concurrent write transactions to eliminate all access time penalties due to contention. It is feasible to construct this architecture for a small shared-memory space using dual-port memory devices.

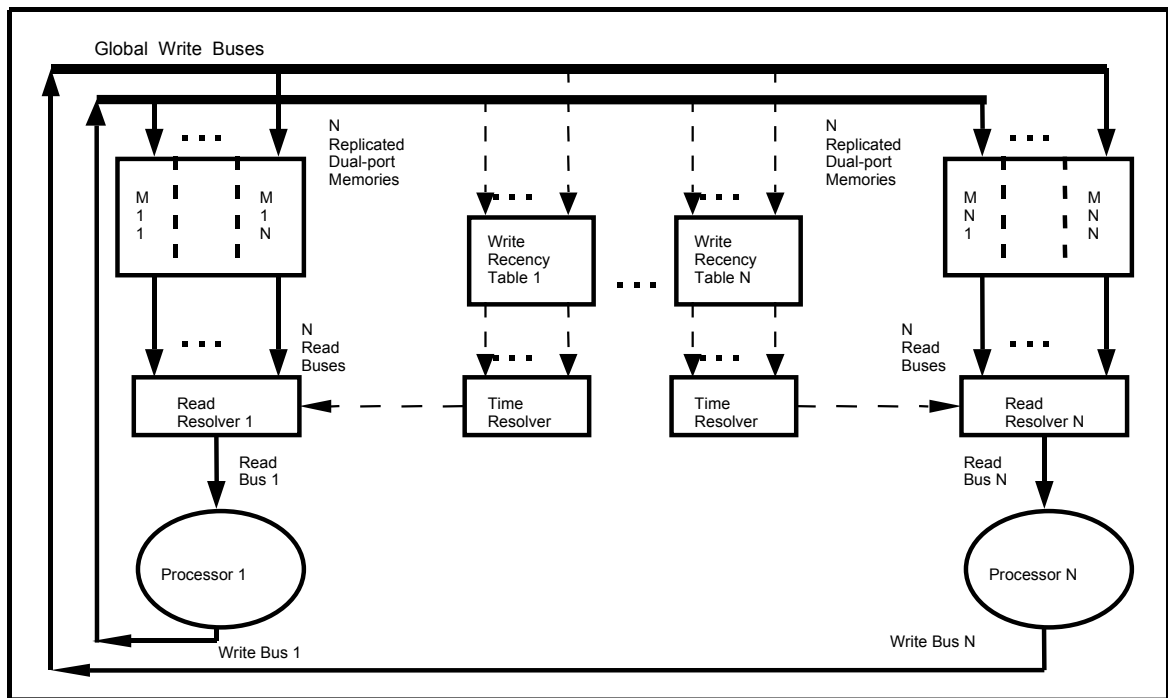


Figure 4.6. Read-Time Resolution Coherent $U_2^{N,N}$ Multiprocessor System

4.7 N-Port Memory Strategy - $B_N^{N,N}$

The system architecture shown in Figure 4.7 is $B_N^{N,N}$ which denotes a B-symmetric, N -plex read, N -plex write memory system using an N -port memory. This strategy utilizes N bi-directional buses supporting concurrent read and write transactions. This configuration is the benchmark for comparison of multiprocessor architectures because no latency is encountered due to memory access contention. The resolution of memory access conflicts is internal to the memory module. Due to pin-out, power, wiring, and capacity constraints this configuration is only feasible for a small number of processors requiring a small shared memory space.

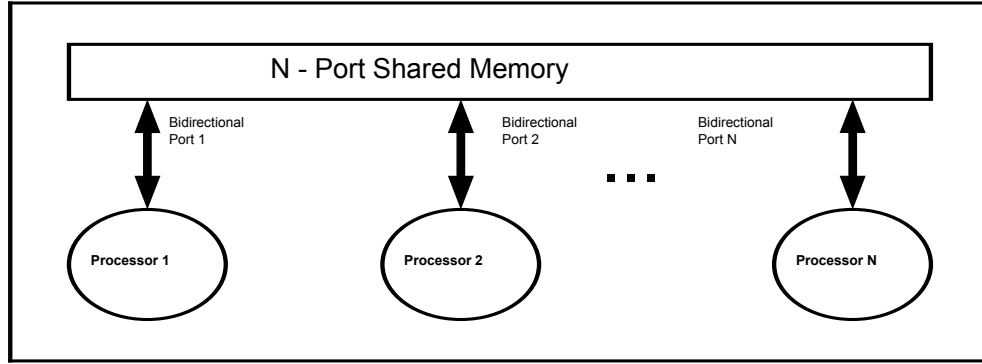


Figure 4.7. N -port Memory $B_N^{N,N}$ Multiprocessor System

4.8 Multiprocessor Architecture Summary

The relevant architectural parameters for each of the above architectures are summarized in Table 4.1, where b denotes the width of the memory buses in bits, N denotes the number of processors, and k denotes the product of the number of addresses and the width of the write-recency table entry.

Table 4.1. Summary of Performance Characteristics

	Likelihood of Read Access Contention	Likelihood of Write Access Contention	Read Contention Penalty (cycles)	Write Contention Penalty (cycles)	Read Contention Complexity	Write Contention Complexity	Wiring Complexity	Memory Requirement	Memory Module Cost
$B_1^{1,1}$	very high	very high	high	High	very low	very low	b	m	low
$U_2^{1,1}$	very high	very high	high	High	very low	very low	$2b$	m	moderate
$B_2^{2,2}$	high	high	high	High	low	low	$2b$	m	moderate
$U_2^{N,1}$	none	very high	none	High	none	moderate	$b(N+1)$	mN	moderate
$U_P^{N,P-1}$	none	moderate	none	moderate	none	high	$b(N+P-1)$	mN	high
$U_2^{N,N}$	none	none	none	none	moderate	none	$2bN$	$(m+k)N^2$	low
$B_N^{N,N}$	None	none	none	none	none	none	bN	m	high

6.0 Conclusion

This paper provides a unified treatment of a wide range of multiprocessor memory architectures. The classifications defined are based on the memory access capability of each processor, the complexity of the memory devices used, the degree of memory replication, and the symmetry of bus usage. This provides a useful mechanism for the comparison of multiprocessing systems from a memory access perspective. It reveals fundamental cost/performance tradeoffs of the representative architectures to indicate which configurations will provide favorable performance. Our experience using this taxonomy in an advanced undergraduate / beginning graduate (i.e. 5000-level) Computer Architecture course, as well as graduate (6000-level) Multiprocessor Architecture course, has been favorable from two perspectives. First, the students are able to increase their knowledge incrementally in a connected fashion, rather than being overwhelmed with a collection of seemingly disjoint implemented machines. Second, the students are able to analytically model the approximate performance of these architectures using elementary counting arguments and probability theory.

References

Barroso and Dubois (1993) "The Performance of Cache-Coherent Ring-based Multiprocessors," *Proceedings of the 20th International Symposium on Computer Architecture*, IEEE Press, San Mateo, CA.

Bisiani and Ravishankar (1990) "PLUS: A Distributed Shared-Memory System," *15th Annual International Symposium on Computer Architecture*, IEEE Press, San Mateo, CA.

Hagersten, Landin, and Haridi (1992) "DDM-A Cache-Only Memory Architecture," *IEEE Computer*, 25(9), pp 44-54, September 1992.

Integrated Device Technology (1995) *1995 Specialized Memories & Modules Data Book*, Integrated Device Technology, Inc.

Lenoski and Weber (1995) *Scalable Shared-Memory Multiprocessing*, Morgan Kaufmann Publishers, Inc., San Francisco, CA.

Rottsoolk and Smith (1995) "Very High Performance Shared Memory MIMD Computer," Tera Computer Company, access <http://www.csto.arpa.mil/summaries>.

Ronald F. DeMara

Ronald F. DeMara is a full-time faculty member in the Electrical and Computer Engineering Department at the University of Central Florida in Orlando, Florida. Dr. DeMara received the B.S. in Electrical Engineering degree from Lehigh University in 1987, the M.S. in Electrical Engineering degree from the University of Maryland, College Park in 1989, and the Ph.D. in Computer Engineering degree from the University of Southern California, Los Angeles in 1992. His research interests include parallel processing, computer architecture, and memory system design.

Paul J. Wilder

Paul J. Wilder received the Bachelor of Science degree in Engineering from the University of Vermont in 1986 and the Master of Science degree in Computer Engineering from the University of Central Florida in 1996. He is currently the Computer and Electronic Engineering Technology program coordinator and an assistant professor of Engineering Technology at the University of Southern Mississippi's Gulf Coast campus in Gautier, MS. From 1989-1991 he held a faculty position at Vermont Technical College in Randolph Center, VT. His research interests include multiprocessor design, multiport memory design, and embedded microcontroller systems.