A Cost-Effective Multistage Interconnection Network with Network Overlapping and Memory Interleaving

KANG G. SHIN, SENIOR MEMBER, IEEE, AND JYH-CHARN LIU, STUDENT MEMBER, IEEE

Abstract — **In this paper, we propose a cost-effective design of circuit switching multistage interconnection networks (CSMIN's). Increase of the network bandwidth and reduction of the network size (and thus low costs) are both accomplished by *network overlapping* and *memory interleaving* (NOMI), instead of by increasing the number of switches or adding buffers.**

The NOMI and its control principle are described on the basis of the structure and interconnection functions of CSMIN's. Detailed accounts of both the network design and the drastic reduction in hardware costs are given. The impact of NOMI on system performance is also analyzed.

Index Terms — **Asynchronous and synchronous multiplexing, bandwidth, blocking factor, circuit switching, memory cluster, multistage interconnection network, network overlapping and memory interleaving, pass rate, processor cluster.**

I. Introduction

AMONG the many unresolved issues in the realization of multiprocessors, development of a cost-effective interconnection network is essential. Such an interconnection network must have a high bandwidth at inexpensive costs and must also be applicable to large multiprocessor systems. There are several candidate solutions to the interconnection problem. The first is the well-known crossbar network, which is usually ruled out due to its high cost. However, it offers a performance upper bound of the second candidate, that is, multistage interconnection networks (MIN's). MIN's are more attractive than the crossbar since for a system having N processors and N memory modules MIN's require $O(N \log_r N)$ $r \times r$ switch elements[1] as compared to $O(N^2)$ switches for crossbar networks. Each $r \times r$ switch used in MIN's is essentially a crossbar with r inputs and r outputs; r is defined as the *order* of the switch. Nonblocking multistage networks have capabilities close to those of crossbar networks [1], [2]. Nevertheless, their hardware costs are still too high to be practical.

There are two different switching methods that have been widely used for MIN's: *packet switching* and *circuit switching* [3], [4]. Since switches are expensive and network delay is significant, it is desirable to fully utilize the interconnection network. Packet switching networks are suitable for transfer of short messages, whereas circuit switching networks are known to be most suitable for massive data exchanges between processors and memory modules [4]. In a conventional circuit switching multistage interconnection network (CCSMIN), a unique path must be established between a source and its destination to route data or messages. Each path is established by physically connecting a set of switches and links in the network. Usually, semiconductor gates are used to direct the data/messages at switches. Assuming that no tristate components are used, different gates and links must be used for transfer of data or messages *from* processors *to* memory modules, then *from* memory modules back *to* processors. As shown in Fig. 1, a CCSMIN can then be decomposed into *forward* and *backward* (sub)networks. The forward network routes requests/data from processors to memory modules, and the backward network returns requested data to processors. In CCSMIN's, once a forward path is established for some processor–memory pair, resources in the backward path are also locked until the memory access cycle of the processor is completed. This method is straightforward, but results in underutilization of resources.

Pipelined operation of circuit switching networks has been proposed to improve the utilization problem of the CCSMIN [5], [6]. Different levels of data registers are added to the switches to form pipelines. If a path has been established successfully, a burst of data can be transferred. However, when the size of the data burst is not large, the performance of the network could be worse than nonpipelined networks due to the setup overhead. To eliminate this deficiency, the operations of the network must be *tuned* to obtain higher bandwidths. To this end, we propose a method of *network overlapping* and *memory interleaving* (NOMI), leading to a cost-effective MIN. For convenience, we term a CSMIN equipped with this NOMI method an *overlapped circuit switching multistage interconnection network* (OCSMIN).

Since modern telecommunication systems make extensive use of digital switching networks, it is worth considering the similarities and dissimilarities between the interconnection networks of multiprocessors and telecommunication systems.[2] The requirements of multiprocessor systems are different from those of telecommunication systems, although they use similar switching networks. For example, parallel transmission of data is usually required for multiprocessor systems because they have much higher bandwidth requirements than acoustic channels. Also, crossbar or nonblocking networks are often selected in telephone systems due to the

Manuscript received May 1, 1985; revised August 12, 1985. This work was supported in part by the U.S. Office of Naval Research under Contract N00014-85-K-0122. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the view of the Office of Naval Research.

The authors are with the Division of Computer Science and Engineering, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109.

[1]Henceforth, these will be abbreviated as *switches*.

[2]An anonymous referee brought this to the authors' attention.

0018-9340/85/1200-1088$01.00 © 1985 IEEE

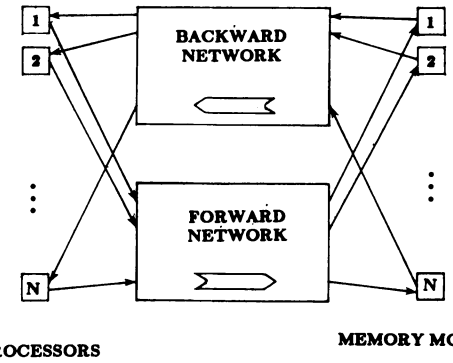


Fig. 1. An $N \times N$ multiprocessor system connected by a CCSMIN.

balking behavior of customers when a waiting queue is formed. These facts lead to different design considerations, as will be seen in this paper.

Both synchronous time division and asynchronous multiplexing have been widely used in telecommunication systems [7], [8]. The first technique is especially important for switching networks of modern telephone systems due to their high capacity requirements. Synchronous time division multiplexing (TDM) is suitable for signal sources requiring identical transmission bandwidths and encoding[3] word lengths. These signals are cyclically sampled and transmitted to their receivers. Acoustic signals are the predominant information processed by telecommunication systems. A/D and D/A converters are required at both the ends of a digitized switching network. A sufficient bandwidth must be assigned in each channel to provide a good speech quality. The above facts impose a limit on the depth of multiplexing in a communication switching network. This type of network is often called a *T–S–T* network, which represents *Time–Space–Time* multiplexing of signals and the switching network [7], [8]. A good example of this type network is the ESS-4 [9]. The underlying philosophy of the second technique, asynchronous multiplexing, is that resources are assigned *as needed*. Asynchronous multiplexing is usually used when there are random requests of resources. However, the overhead of asynchronous control increases rapidly with the size of the system, thus making it unacceptable for large systems.

In a computer system, shared resources are buses, I/O channels, or an interconnection network. For MIN-based multiprocessor systems, it is important to match bandwidths of its subsystems with each other since unmatched bandwidths lead to bottleneck or underutilization of resources. We will focus here on the design and analysis of an OCSMIN for large multiprocessor systems. NOMI can be viewed as a type of synchronous multiplexing. It results in a significant reduction of the network size, which in turn shortens the network delay and saves hardware. This is made possible by closely tuning system components with each other and is thus a novel departure from conventional methods of using more hardware to improve performance.

The rest of this paper is organized as follows. Section II describes the NOMI and its operating principles. In Section III the design complexity and the hardware costs associated with the CCSMIN and OCSMIN are comparatively analyzed. Section IV deals with a comparative performance evaluation of the two networks. The paper concludes with Section V.

[3]Pulse code modulation (PCM) is one of the most popular techniques.

II. Operational Principles

Operational functions of the CCSMIN and OCSMIN are presented first in this section. Implementation of the OCSMIN is then illustrated by an example design of network components.

A. Notation and Interconnection Functions

There are typically three major subsystems in a multiprocessor system: processor subsystem, memory subsystem, and an interconnection network. Although NOMI can be applied to networks of any topology, it is important to carefully select an underlying topology which allows for easy implementation of the system. This is obvious from the fact that the network topology is one of the most important factors in deciding the ease of system construction [10], [11].

In this paper, we add one more level of abstraction named *cluster* to the system hierarchy of the CCSMIN. A *processor cluster* pc_i is a set of processors p_{ij}, $1 \le j \le w$, and an interface unit pn_i between these processors and the network, i.e., $pc_i \equiv \{pn_i, p_{ij} \mid 1 \le j \le w\}$ for all $1 \le i \le N'$ where N is the number of processors in the system, $N' = N/w$ is the *height* of the network, and w is the number of processors in a cluster. Logically, an interface unit is the mechanism which allows the network to communicate with processors or memory modules. Its functions include voltage level transferring, control signal encoding/decoding, etc. For clarity of presentation, only one interface unit is assigned to each cluster. However, such an interface unit may be implemented with several LSI chips, each of which is attached to a processor. The *processor subsystem* can then be defined as the collection of processor clusters, $PS \equiv \{pc_i \mid 1 \le i \le N'\}$. The memory subsystem can be similarly defined. Let mc_i, $1 \le i \le N'$, be a *memory cluster*, which is defined as a set of memory modules m_{ij} and an interface unit mn_i between the network and the memory modules, i.e., $mc_i \equiv \{mn_i, m_{ij} \mid 1 \le j \le w\}$. The memory subsystem MS is the collection of memory clusters, i.e., $MS \equiv \{mc_i \mid 1 \le i \le N'\}$. Fig. 2 shows a block diagram of these components in the system.

A network of k stages is needed to connect processor and memory clusters where $k \equiv \log_r N'$ is called the *width* of the network and r is the order of the switches used in the network. Denote a *forward switch* in the forward network by $FS_{ij}(m, l)$ where ij is the coordinate of the switch and m and l represent the mth input port and lth output port, respectively. (m and l will be omitted whenever they do not present any ambiguity.) The *backward switches* in the backward network can be symmetrically expressed by replacing FS with BS, i.e., $BS_{ij}(m, l)$ where m now represents the mth output port and l represents the lth input port. The entire collection of switches in the network can be represented by the set $\{FS_{ij}, BS_{ij} \mid 1 \le i \le N', 1 \le j \le k\}$.

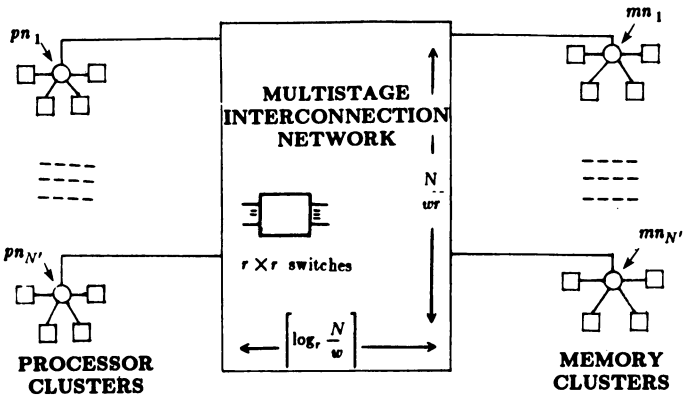


Fig. 2. The hierarchical structure of a multiprocessor system.

The same notation applies to links. Since no tristate components are assumed, separate links must be used in the forward and backward networks. The total collection of links is represented by the set $\{FL_{ij}, BL_{ij} \mid 1 \le i \le N', 1 \le j \le k+1\}$ where FL_{ij} (BL_{ij}) is a link in the forward (backward) network. Note that there are only N' interface units between processor clusters and the network, and N' interface units between the network and memory clusters. Thus, the size of the network is decided by the number of clusters, instead of by the number of processors or memory modules. Using the above model, the CCSMIN is a special case when $w = 1$ and $N = N'$. The forward and backward networks are said to be *topologically identical* if they have identical network structures except for the direction of routing.

Definition 1: In a CSMIN, which is composed of topologically identical forward and backward subnetworks, the switch BS_{ab} is called the *partner* of FS_{ef}, $BS_{ab} = \Pi(FS_{ef})$, when $a = e$, $b = f$, and BS_{ab} is set up for later servicing of all the requests passing through FS_{ef}.

Similarly, the partner of a forward (backward) link, which is also a backward (forward) link, can be defined. For unique-path CSMIN's, a processor's need for use of a switch or link in the forward network implies the same need for use of its partner in the backward network. However, we will show that the *time* of locking a forward switch's (or link's) partner plays a key role in deciding the efficiency and performance of the system.

Possible interconnections between processors and memory clusters can be represented by a relation, termed the *interconnection relation,*

$$IR \equiv \{(pc_i, {}_iFP_j, mc_j, {}_jBP_i) \mid 1 \le i,j \le N'\}$$

where

$${}_iFP_j \equiv \{(pn_i, FS_{1x_1}(i_1, o_1), FL_{1y_1}, FS_{2x_2}(i_2, o_2), FL_{2y_2}, \cdots, FS_{kx_k}(i_k, o_k), FL_{k+1y_{k+1}}, mn_j)\}$$

is the set of all possible routes from pc_i to mc_j. Similarly,

$${}_jBP_i \equiv \{(pn_i, BS_{1x_1}(l_1, r_1), BL_{1y_1}, BS_{2x_2}(l_2, r_2), BL_{2y_2}, \cdots, BS_{kx_k}(l_k, r_k), BL_{k+1y_{k+1}}, mn_j)\}$$

represents all possible paths from mc_j to pc_i.

For unique-path blocking networks there is exactly one element in each ${}_iFP_j$ (${}_jBP_i$), which is a set of switches and links. In such a case, the network resources cannot be shared by two or more requests at the same time. The relation IR is not sufficient to describe a system with operation overlapping. A dynamic model is necessary to describe the behavior of both the CCSMIN and OCSMIN. At an instant t, the request pattern $A(t)$ is represented by a set of N'-tuples

$$\{(a_1(t), a_2(t), \cdots, a_{N'}(t)) \mid a_i(t) \in \{0, 1, \cdots, N'\}, \quad 1 \le i \le N'\}$$

where $a_i(t) = j$ if pc_i makes a request for mc_j at time t, and $a_i(t) = 0$ if no request is made by pc_i. For each request $a_i(t)$, there is a corresponding interconnection in the forward network, called a *forward path,*

$${}_iFP_j(t) = \begin{cases} {}_iFP_j & \text{if resources for the path are available at time } t \\ \emptyset & \text{if } a_i(t) = 0 \text{ or the resources are not available at time } t. \end{cases}$$

Similarly, the corresponding *backward path* for this request is

$${}_jBP_i(t) = \begin{cases} {}_jBP_i & \text{if resources for the path are available at time } t \\ \emptyset & \text{if } a_i(t) = 0 \text{ or the resources are not available at time } t. \end{cases}$$

We can define the *forward interconnection pattern* as

$$FIP(t) \equiv \{{}_iFH_j(t) \mid 1 \le i,j \le N'\}, \quad \text{where } {}_iFH_j(t) = (a_i(t), {}_iFP_j(t), mc_j),$$

and the *backward interconnection pattern* as

$$BIP(t) \equiv \{{}_jBH_i(t) \mid 1 \le i,j \le N'\}, \quad \text{where } {}_jBH_i(t) = (a_i(t), {}_jBP_i(t), mc_j).$$

$FIP(t)$ and $BIP(t)$ provide services to the requests on the forward and backward networks, respectively. In the CCSMIN, both ${}_jBH_i$ and its partner ${}_iFH_j$ must be dedicated simultaneously for a read operation. Actually, $BIP(t) = \Pi(FIP(t))$ for all t in the CCSMIN. This can be represented by the *total interconnection pattern,*

$$TIP(t) \equiv FIP(t) \times BIP(t) \equiv \{(a_i(t), {}_iFP_j(t), mc_j, {}_jBP_i(t))\}.$$

Definition 2: The function $CSF: A(t) \rightarrow TIP(t)$ is the interconnection function of an operational CCSMIN, such that for all $1 \le i, j \le N'$ the following hold.

1) $CSF(a_i(t)) = {}_iFH_j(t) \times {}_jBH_i(t)$.

2) The pattern ${}_iTIP_j(t)$ is unique for each $a_i(t)$, and for each nonnull path, ${}_{i_1}FP_{j_1}(t) \cap {}_{i_2}FP_{j_2}(t) = \emptyset$ for all $i_1 \ne i_2$.

3) ${}_iBP_j(t) = \Pi({}_jFP_i(t))$ for all t.

By ${}_{i_1}FP_{j_1} \cap {}_{i_2}FP_{j_2} = \emptyset$, we mean that the requests $a_{i_1}(t)$ and $a_{i_2}(t)$ do not require the same resources. Clearly, the underutilization of the CCSMIN is due to its total interconnection requirement. The corresponding timing chart and interconnection function of a CCSMIN are shown in Fig. 3. The gates and links locked in the forward network are forced to be idle during periods $T2$ and $T3$. Similarly, the gates and

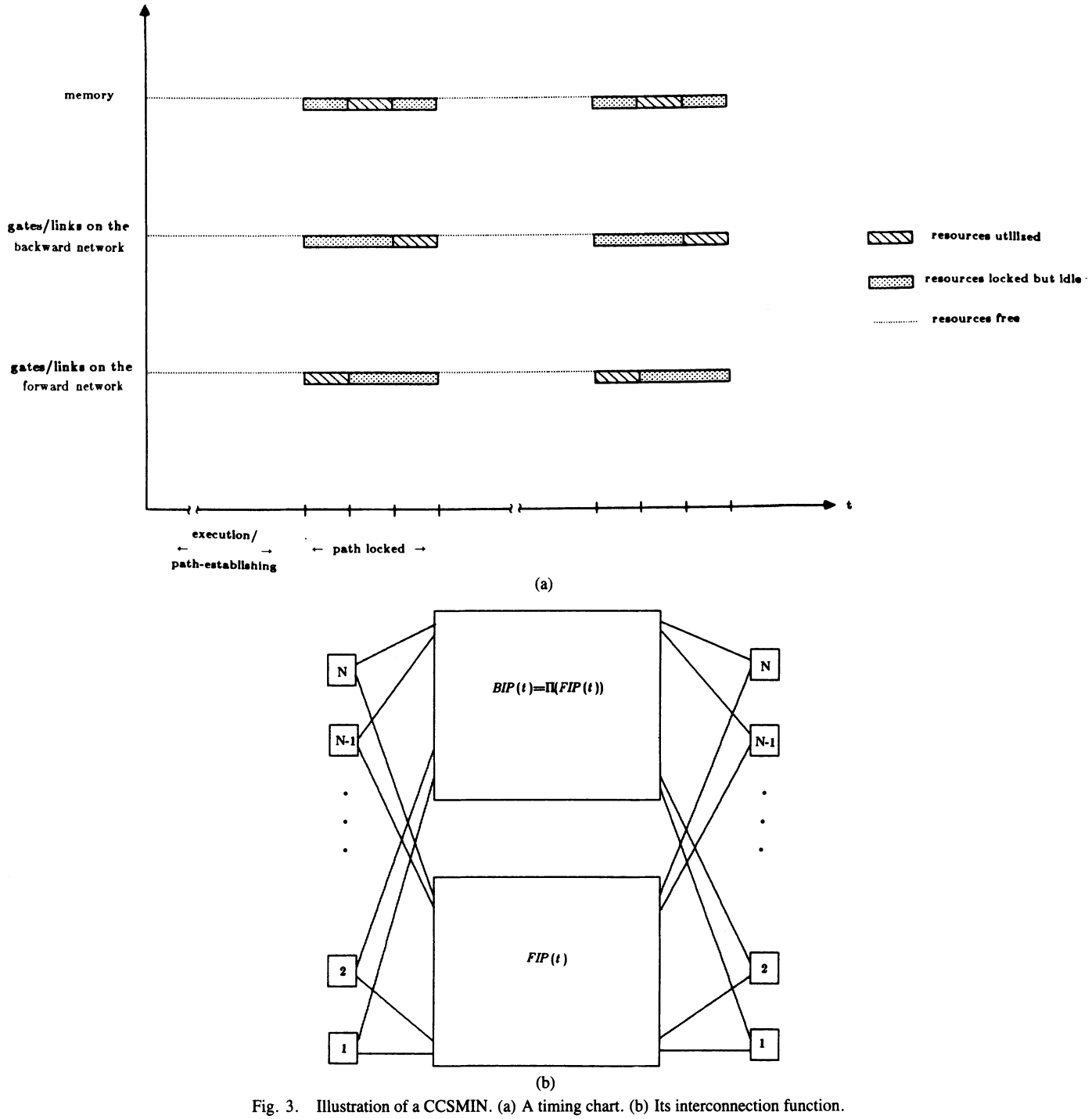


Fig. 3. Illustration of a CCSMIN. (a) A timing chart. (b) Its interconnection function.

links locked in the backward network are idle during $T1$ and $T2$. To remedy this underutilization problem, we can weaken the total interconnection requirement CSF as stated in the following definition.

Definition 3: The function $OVF: A(t) \rightarrow TIP(t)$ is an interconnection function of the OCSMIN such that for all $1 \leq i, j \leq N'$ the following hold.

1) $OVF(a_i(t)) = {}_iFH_j(t) \times {}_{j'}BH_{i'}(t)$, i and j may or may not be equal to i' and j'.

2) The pattern ${}_iTIP_j(t)$ is unique for each $a_i(t)$, and for each nonnull path, ${}_{i_1}FP_{j_1}(t) \cap {}_{i_2}FP_{j_2}(t) = \emptyset$ if $i_1 \neq i_2$.

3) ${}_jBP_i(t) = \Pi({}_iFP_j(t - \Delta t))$ for all t and some fixed Δt.

The timing chart and interconnection function of an OCSMIN are shown in Fig. 4. It will be shown that the function OVF can improve the network utilization and the network performance under certain conditions. Note that although mn_j (pn_i) can simultaneously appear more than once in ${}_iFH_j$ (${}_jBH_i$) for different requests, the operation will still be correct due to the independent input and output ports that are used in the interface units. Functions CSF and OVF describe the properties of the CCSMIN and its overlapped counterpart, OCSMIN. However, it is neither possible nor desirable to implement the OVF for all ranges of Δt. According to the definition of OVF, processors can access memory modules of the same path only after Δt. The maximum number of requests that can be routed over one path within Δt is

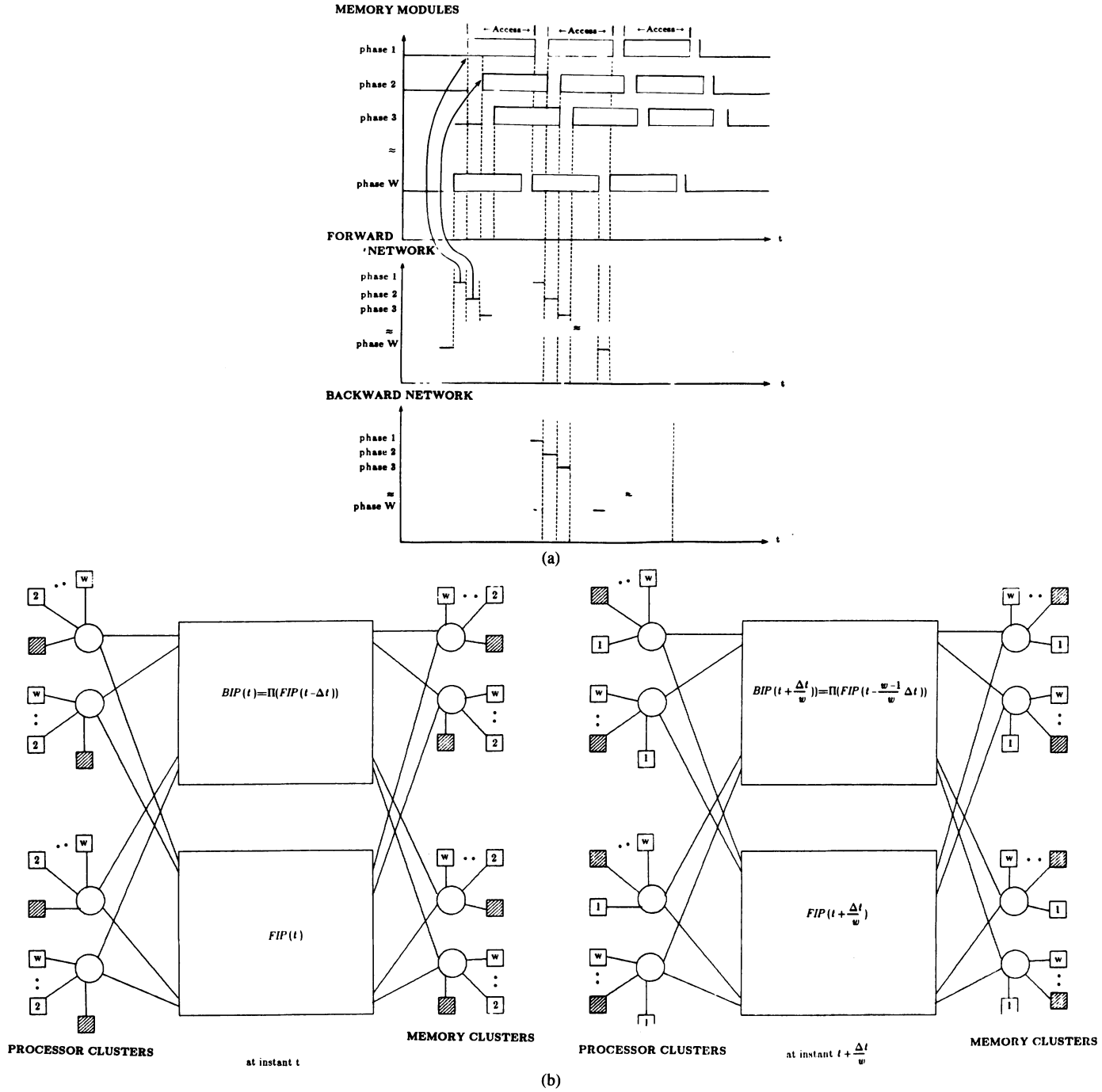


Fig. 4. Illustration of an OCSMIN. (a) A timing chart. (b) Its interconnection function.

called the number of *phases*, w; it is also the depth of NOMI, as shown in the next subsection. The *network cycle* or *network propagation delay* is the time required to route one request through the network to its destination. Conditions required to satisfy the function *OVF* will be derived in the next subsection. In addition, effects of changes in the number of phases on the cost and performance of the system will be elaborated.

B. Principles of Overlapping and Interleaving

The basic idea of the OCSMIN is that after a processor cluster has established its forward path to a memory cluster successfully, the backward partner path is not immediately locked. Instead, the resources in the backward network are just reserved for use by the request only after a delay of Δt. Likewise, the resources in the forward network are released immediately upon latching the processor's data into the destination memory module. Since the network resources for a request are not all locked during the entire service period, they can offer concurrent services to other requests, which is not possible in the CCSMIN. The necessary and sufficient condition to implement the function *OVF* is given by the following theorem.

Theorem 1: The function *OVF* can be implemented if and only if $\Delta t = T_M$, and $w \times T_D = T_M$, where T_D is the network propagation delay, T_M is the memory cycle time, and $w > 1$.

Proof: The inequality $\Delta t < T_M$ cannot hold since the system is unable to return service before a memory access is completed. When $w = 1$, the network becomes a CCSMIN and is therefore removed from consideration.

Suppose $w \times T_D = \Delta t = T_M$ and there are w requests sent to memory modules and serviced during the period T_M. Services in one memory cluster will be completed in the same order as routed through the forward network to the cluster. Obviously, the completed services will then be routed in the same order through the backward network to the originating processors. Thus, ${}_jBP_i(t + T_M)$ is the partner of ${}_iFP_j(t)$ for all t. When one completed service is being routed through the backward path, its partner (forward) path is available for other requests, thus satisfying all three conditions of the function *OVF*.

Assume now that the network is overlapped in w' ways and no buffer is used in the network. For the *OVF* function the forward network must be made to support a full utilization of memory modules which have the cycle time T_M and are interleaved in w ways. That is, it must be able to deliver at least w requests to the memory modules during T_M. This requires a forward network with the propagation delay T_D to satisfy $T_D \leq (T_M/w)$ (or $w' \geq w$). If $w' > w$, there will be an excessive number of requests to be serviced by the memory modules; those requests will be balked, thereby making it impossible to meet the third condition of the function *OVF*. Therefore, $w' = w$.

When the memory subsystem is fully utilized, one memory cluster must be able to return w data/messages via the backward network to processor clusters during T_M. Thus, $\Delta t = T_M = w \times T_D$ is necessary. ■

Accurate estimation of the bandwidth of CPU's or processors is still an open problem. The estimation should be based on the analysis of system workload, which is known to be very difficult for general-purpose systems. There are, however, two simpler and more practical design methods available. The first is the use of the well-known benchmark programs, and the second is the use of the worst case estimation. Let $1/T_P$ be the required bandwidth of processors, where T_P could represent i) the mean memory request cycle of processors for the first approach, and ii) the shortest memory request cycle for the second approach. The second approach obviously requires a higher bandwidth than the first. Note that processors make access requests randomly to memory modules.

A useful corollary follows immediately from the above observation and Theorem 1.

Corollary 1: The maximum number of processors within one processor cluster is $N_P \equiv (T_P/T_D)$ where T_P is the memory request cycle of processors and T_D is the network propagation delay.

Proof: Let w_P be the depth of processor interleaving, which is identical to the number of processors within one cluster. To have a matched bandwidth between processors and the network, $(w_P/T_P) = (w/T_M)$ must hold. Thus, $w_P = (T_P/T_D)$ by Theorem 1. ■

This corollary implies that processors can be interleaved only when $T_P > T_D$. In view of the complexities of microprocessor and switching elements, the condition $T_P > T_D$ is much easier to meet by using contemporary semiconductor technology. It is also possible to have heterogeneous units in a processor or memory cluster for perfectly (bandwidth) matched systems. This leads to an asymmetric network structure, i.e., consisting of switches with different numbers of input and output ports. However, homogeneous clusters will be used throughout this paper for clarity of presentation.

For the multiprocessor system under consideration, one memory cycle is divided into w network cycles. If the system has N processors and N memory modules, both processors and memory modules are grouped into (N/w) processor clusters and (N/w) memory clusters, respectively. Each cluster is composed of w processors or memory modules. By the definition of the function *OVF*, one and only one processor (or memory module) within a cluster is allowed to be connected to the network at a time. That is, processors and memory modules in a cluster must be *interleaved* for NOMI. Memory modules in a cluster must by *physically* interleaved and operate in phases ph_i, $1 \leq i \leq w$, where $ph_i \neq ph_j$ for $i \neq j$, i.e., a w-phase clock is used in each cluster. Processors within a cluster can use any clock phase(s), provided that they attempt to access memory modules of different phases. Thus, immediate conflicts among these processors are prevented since only one processor can access the network and memory modules in each phase. This operation must be controlled by proper scheduling so as to maximize the performance. However, such *logical* interleaving of processors cannot always be achieved due to the random nature of memory accesses by processors. The impact of direct conflicts within one cluster on performance is analyzed in Section IV.

One potential complexity in the network operation is the routing of microprocessors' control signals. Compared to memory access operations, many of the control signals on microprocessor chips are rarely used, e.g., interrupts and their acknowledgments. Routing of these signals complicates the timing control of the network. However, commercial translators between different control protocols are available for a similar purpose, e.g., dynamic RAM (DRAM) controllers, bus controllers of some commercial products, etc. Thus, we assume that control signals on microprocessor chips can be encoded/decoded by the processor–network interface into the conventional data/addresses format with different identification tags. This encoding/decoding process can simplify the control signals of the network.

In conventional designs, a feedback (or asynchronous) protocol, i.e., handshaking, is often used for a service acknowledgment. To obtain a low propagation delay, handshaking is not adopted in the OCSMIN design. Instead, the processor's phase number can be transmitted from pn_i to the destination mn_j. For both *read* and *write* operations, the source's phase number can be returned to acknowledge the completion of service. On the other hand, if the phase number is not echoed within Δt, the source cluster will regard that its request is dislodged, and the request must be retransmitted. A comparison between the handshaking and nonhandshaking methods is shown in Fig. 5. The feedback delay reported in [12] can be eliminated in our design.

The *switch arbitrator* (*control unit*) interprets routing tags and controls the routing of data according to the switch's status. There are two modes of operation for the arbitrator: *predefined priority* and *asynchronous*. In the predefined priority mode, the priority of each request is fixed *a priori*. Requests of higher priority must always be honored prior to requests of lower priority. This is the simplest mechanism and can be implemented by combinational circuitry. *Asyn-*

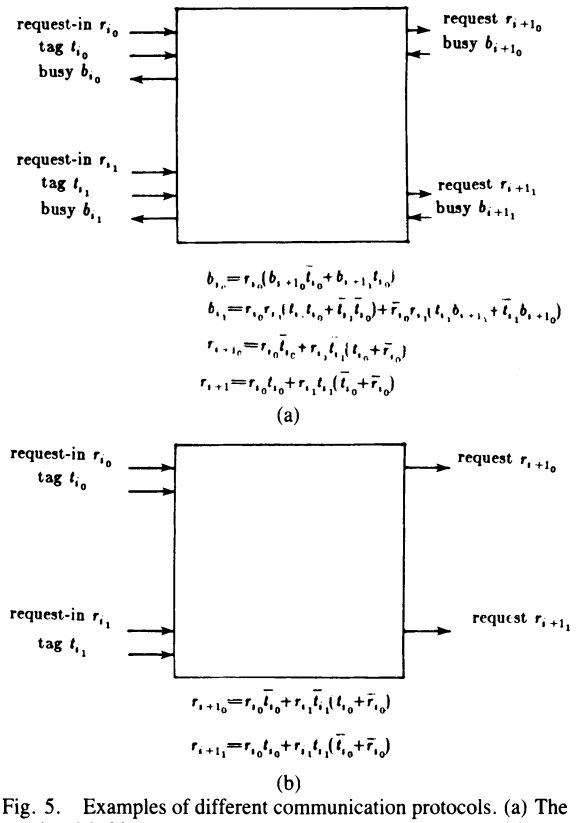


Fig. 5. Examples of different communication protocols. (a) The handshaking protocol. (b) The nonhandshaking protocol.

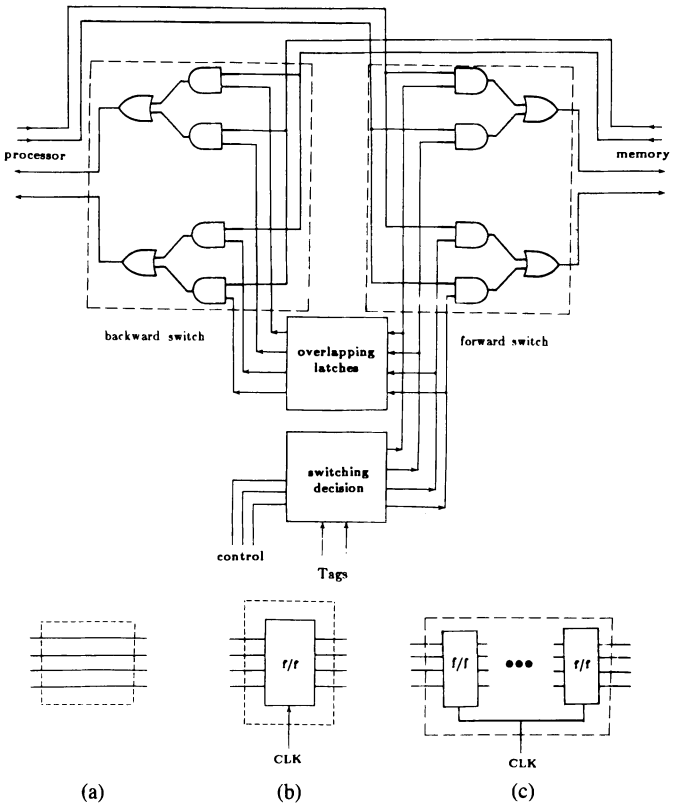


Fig. 6. Required circuitry for support. (a) CCSMIN. (b) Two-way overlapped network. (c) w-way overlapped network.

chronous arbitrators outperform the other in a large system, but their design complexity is directly proportional to the number of their input signals [13]. When the number of input signals is large, the required circuitry is too complicated to be practical. Thus, the asynchronous mode can only work well for low-order switches. The predefined priority mode is chosen in our design to achieve a low network delay.

There are w requests to be routed during one memory cycle. Thus, $w - 1$ switching states must be memorized by each switch. Fig. 6(a)–(c) shows the different circuitry required to support OCSMIN's and CCSMIN's. A two-way overlapped network can be designed by using one set of registers, as shown in Fig. 6(b). When the network is overlapped in w ways, $w - 1$ cascaded shift registers are needed for each switch. In addition, the network clock must be w times faster than the memory clock to route w requests in one memory cycle. As can be seen from Fig. 6, these registers are *not* for buffering data/messages, but are for storing switching decisions.

Static RAM's (SRAM's) and DRAM's are the most popular main memory components for commercial machines. DRAM's are more attractive than SRAM's for these machines due to their relatively lower price. They have about the same access time, but DRAM's have a higher density than SRAM's. Although early DRAM and SRAM chips required only one phase operation, contemporary commercial DRAM chips are standardized to use multiplexed two-phase addressing: row (*ras*) and column selection (*cas*). On the other hand, one-phase operation is still a standard for commercial SRAM's [14]. In the discussion to follow, by SRAM we mean *single*-phased RAM's, and by DRAM we mean *double*-phased RAM's. Some of the most popular chips in the market clearly exhibit this fact, e.g., 4K × 8 SRAM's have a 150 ns access time, and the operation of 64K × 1 DRAM's is composed of two suboperations, row address select *ras* and *ras precharge* (i.e., *cas*). Each suboperation requires 150 ns under its normal operational mode. The operational principle of memory interleaving is well known and is not repeated here. The NOMI techniques using DRAM's and SRAM's are essentially the same. A minor modification is needed though, when DRAM's have an identical time period for both *ras* and *cas* operations. Figs. 7 and 8 illustrate the difference of two-way NOMI techniques using DRAM's and SRAM's.

III. Cost Analysis

There are two problems associated with a network requiring an excessive number of switches and stages. One is its enormous wiring, cooling, and power requirement. Such physical requirements can cause system performance degradation as well, e.g., a network delay. The other is the cost of the system despite the continuing advances and lowering costs in the related device technology. Gates and links to route data/messages are the predominant components of the network.

Definition 4: The *cost* of a CCSMIN (or an OCSMIN) is defined to be the number of data gates and links. The *cost factor* is the ratio of the number of data gates to the number of processors.

The number of data gates and links is a better network cost index than the number of switches since it expresses the network cost at the lowest level. Network delay in a large

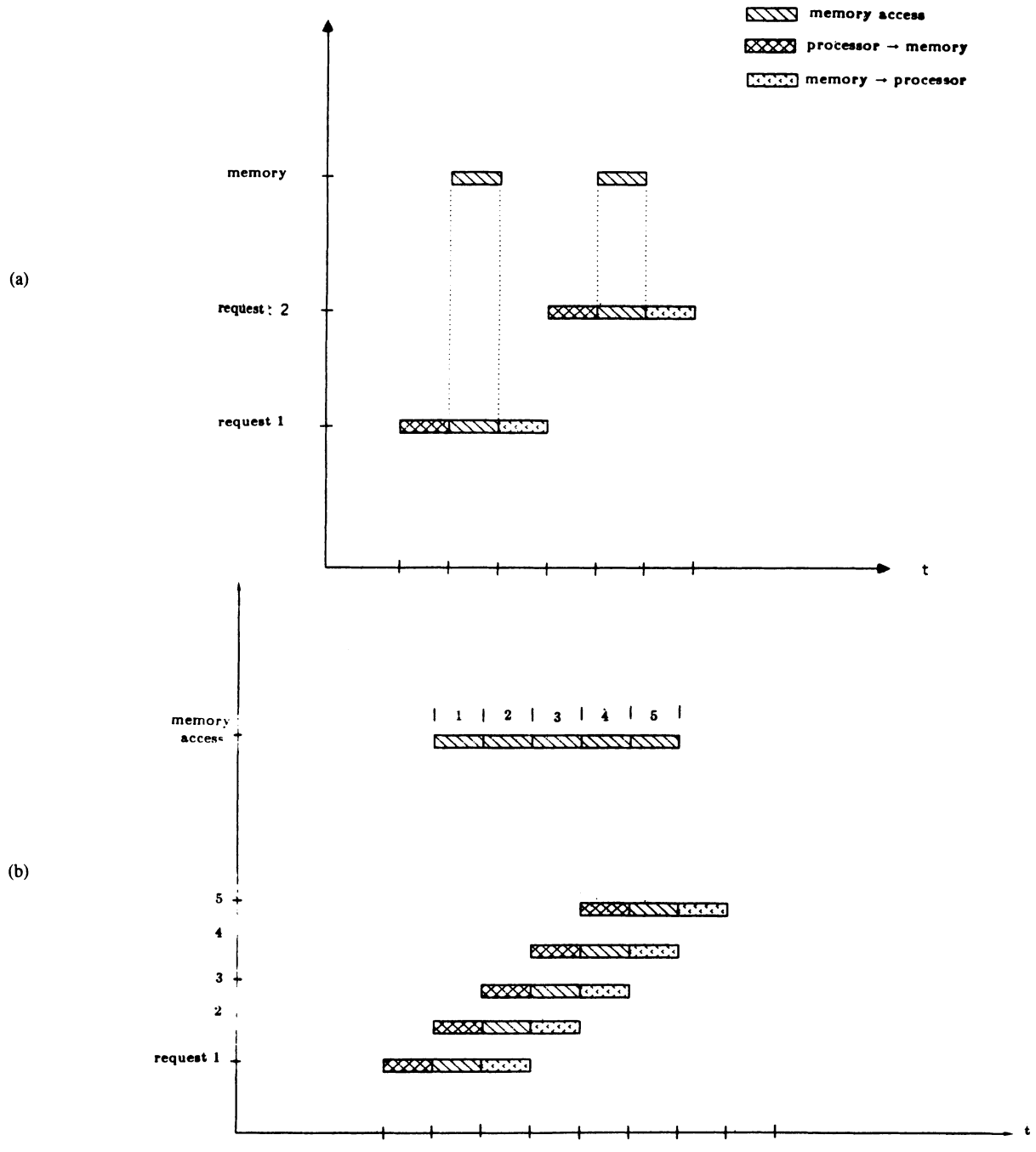


Fig. 7. Comparison of SRAM operations. (a) CCSMIN. (b) OCSMIN.

multiprocessor system is known to be significant due to the cable propagation delay. For example, in a system with 512 processors, 521 memory modules, and a 10-stage network, the network delay is shown to be 120 ns and the memory access time is shown to be 250 ns [15]. Due to its strong dependence on the physical construction, the cost of data links should be evaluated in terms of the number and length of links. It is shown in [16] that although the switch complexity of multistage networks grows as $O(N \log N)$, the link length still grows in $O(N^2)$ complexity. The total number of links in a network is proportional to the number of input and output ports in each switch used. For simplicity the network link length is ignored in our analysis, but the number of links is implicitly included in that of data gates. Thus, the result shown here is a lower (conservative) bound of performance improvement made by the NOMI technique. Note that r^2 data gates are required for one $r \times r$ switch. The cost and cost factor of a CCSMIN using $r \times r$ switches are $rN \log_r N$ and $r \log_r N$, respectively. For a fixed N, the cost factor is a monotonically increasing function of r. When the depth of NOMI is w, only N' processors and N' memory modules have to be connected in each phase. This implies that the network can be reduced in height and/or width. For an overlapped network without changing the order of switches, the total number of gates is $rN' \log_r N'$. The total number of gates is reduced by $rN\lceil \log_r N \rceil - rN'\lceil \log_r N' \rceil$, and the cost factor of the overlapped network becomes $(r/w)\lceil \log_r(N/w) \rceil$. This reduction in the number of gates by NOMI is significant. For

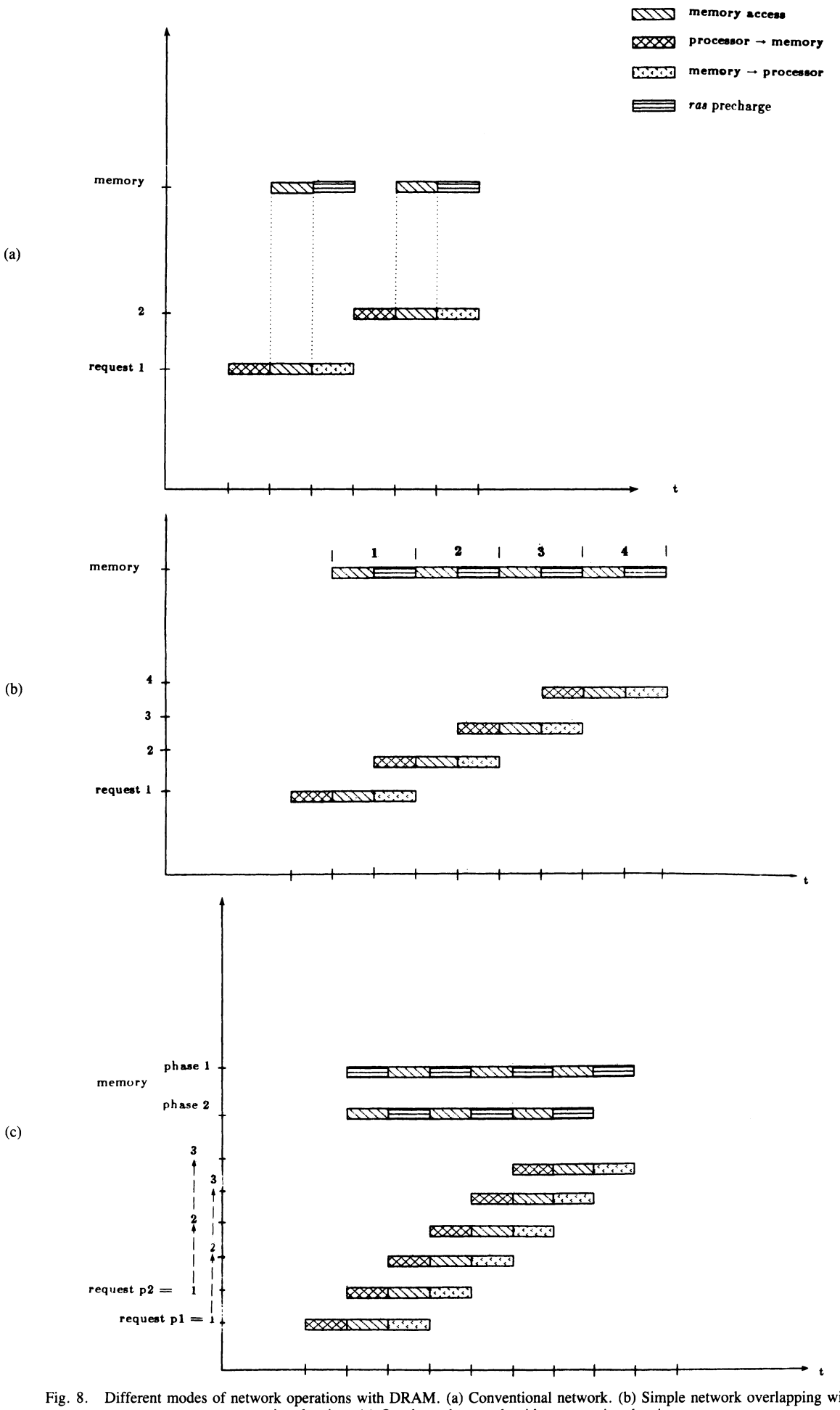


Fig. 8. Different modes of network operations with DRAM. (a) Conventional network. (b) Simple network overlapping with memory interleaving. (c) Overlapped network with memory interleaving.

example, 5120 2 × 2 switches (20 480 gates) are required for a conventional 10-stage MIN, while only 1024 2 × 2 switches (4096 gates) are needed to meet the same requirement, if the network is overlapped in four ways. On the other hand, a slight increase in hardware complexity is needed to support NOMI: 1) additional $w - 1$ shift registers at each switch, 2) a network clock w times faster than the *memory clock*, 3) a w-phase clock for each memory cluster, and 4) an interface unit capable of matching up the speed of the network and that of processors or memory modules.

An increase in the switch order generates two mutually conflicting effects: higher network cost and better performance. However, NOMI can neutralize the increase in the network cost without sacrificing the network performance since it in general reduces the network size. The following theorem states the possible benefit obtainable from a combination of these two mutually conflicting factors.

Theorem 2: The cost of a CSMIN can be made monotonically decreasing when

1) the CCSMIN is replaced by an OCSMIN,
2) higher order switches are used, and
3) the relative increase in the overlapping (or interleaving) depth is greater than that in the switch order.

Proof: The cost C_{orig} is $rN\lceil \log_r N \rceil$ for an $N \times N$ system connected by a network using $r \times r$ switches. If a new network with the overlapping depth w and $r' \times r'$ switches is used to connect processors and memory modules, then the cost of this new network, C_{new}, is $r'(N/w)\lceil \log_{r'}(N/w) \rceil$. Since $r' > r$, the logarithmic factor in C_{new} is less than or equal to that of C_{orig}. We can now consider the factors rN in C_{orig} and $r'(N/w)$ in C_{new}. Applying the third condition to these factors, we get $(r/r') < (w/w')$, and thus $(r'/w')N < (r/w)N$. It follows immediately that $C_{\text{orig}} \geq C_{\text{new}}$. ■

Corollary 2: The cost of an OCSMIN satisfying Theorem 2 is always the same as or lower than that of a bidirectional CCSMIN.

Proof: A bidirectional network uses only one network. It is composed of tristate data gates and bilateral links for bidirectional data transmission.[4] Thus, the cost of a bidirectional CCSMIN is fixed to be $(rN/2)\lceil \log_r N \rceil$.

The lowest possible NOMI depth is 2 for an OCSMIN. The cost of the two-way overlapped OCSMIN is $(rN/2)\lceil \log_r N \rceil$ when identical switches are used. Since the cost of networks satisfying Theorem 2 monotonically decreases, the corollary obviously holds. ■

IV. Performance Evaluation

In this section, we comparatively evaluate both the CCSMIN and OCSMIN. In addition to the conventional performance parameters, i.e., blocking factors and network bandwidths, we will introduce and evaluate two new parameters called the *mean system access time* and *execution/access parallelism* as metrics for assessing the network performance.

A. Blocking Factors and Bandwidths

There are numerous models proposed in the literature for evaluating the blocking factors of the CCSMIN [17–21]. Most of these models, with minor modifications, can be used to compare the blocking factors of the CCSMIN and OCSMIN since the blocking factor of the OCSMIN is independent of the overlapping depth *within* each phase. For simplicity, we have adopted the same basic assumptions in [17] for an $N \times N$ system.

A1. Requests generated by processors are independent and uniformly distributed over all memory modules.
A2. The network operates synchronously, and every path establishment takes one network cycle. Each processor generates a new request with probability P. Thus, P is also the average number of requests generated per cycle by a processor.
A3. A blocked request is ignored during the current cycle, but a new request will be generated at the next cycle.

It is easy to derive directly the blocking factor equation as $P_{i+1} = 1 - (1 - (P_i/r))^r$ where P_i is the probability that a request presents at an output port of an $r \times r$ switch located at stage i, and P_0 is the request rate of a processor cluster. The *network pass rate* is the ratio of requests which pass through the network to those made by processors. Let $k \equiv \log_r N$ be the number of stages and P_{C_k} (or P_{O_k}) be the probability that a request can pass through the k stages of the CCSMIN (OCSMIN). The respective network pass rates for the CCSMIN and OCSMIN can be calculated by $P_{A_C} = (P_{C_k}/P)$ and $P_{A_O} = (P_{O_k}/P)$.

The memory bandwidth of a large multiprocessor system can be improved by NOMI. To compare the performance of a CCSMIN to that of a corresponding OCSMIN, each path establishment is assumed to take one cycle, and all memory modules operate synchronously, i.e., assumption A2. In the CCSMIN's, the second subcycle (*ras precharge*) of a DRAM can occur during the period of routing data back to the processor. Thus, memory bandwidths of DRAM's and SRAM's can be essentially the same as shown in Fig. 9. The memory bandwidths are $1/(2T_D + T_{M_S})$ and $1/\max(T_{M_S}, T_D)$ for the CCSMIN and OCSMIN, respectively, where T_D is the network delay and T_{M_S} is the memory cycle time of an SRAM.

A relative timing chart for an OCSMIN with DRAM's was given in Fig. 8. in a simple OCSMIN without interleaving memory modules, the maximum efficiency of the network is one half of SRAM's due to the two subcycles required for DRAM's. In such a case, the memory bandwidths of both the networks become $1/(2T_D + T_{M_D})$ and $1/(2 \max(T_{M_D}, T_D))$ where T_{M_D} is the memory cycle time of a DRAM. Clearly, not much improvement is made via this mode of operation.

Consider now a k-stage CCSMIN and an l-stage OCSMIN where $l = \lceil \log_r(N/w) \rceil$. Then their respective network bandwidths are

$$BW_C = \frac{N \times P_{A_k}}{T_M + 2T_D}$$

$$BW_O = \frac{1}{T_M} = \frac{N \times P_{A_l}}{w} \times w = \frac{N}{T_M} \times P_{A_l}$$

where BW_C and BW_O are the respective bandwidths of the CCSMIN and OCSMIN, P_{A_l} is the pass rate of the l-stage network, T_M is the memory cycle time, T_D is the network delay, w is the depth of interleaving, and N is the number of memory modules. When the network is fully loaded, the

[4]Note that the NOMI technique *cannot* be applied to bidirectional networks.

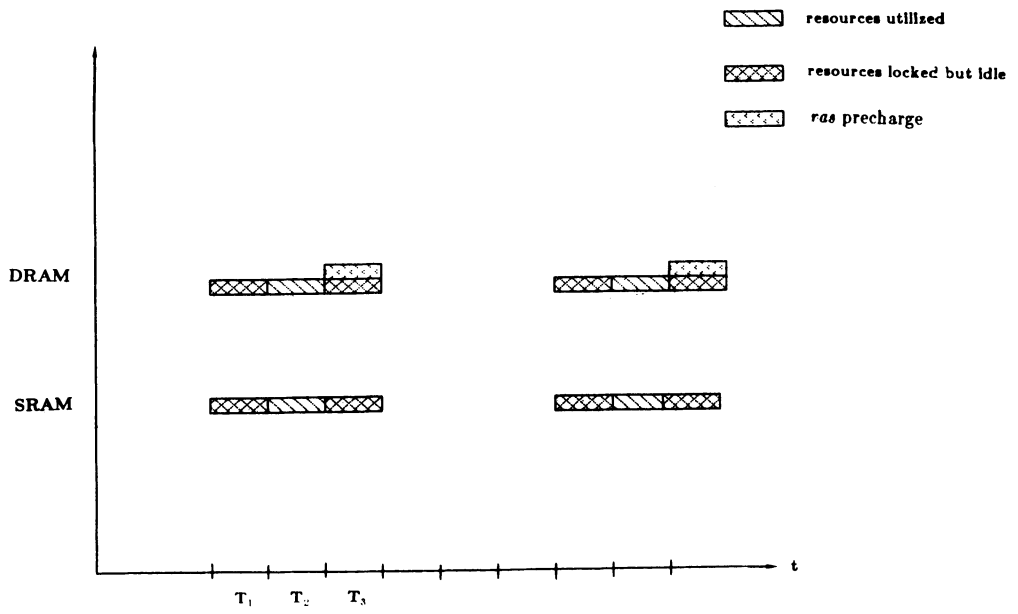


Fig. 9. Comparison of DRAM and SRAM operations in a CCSMIN.

utilization factors are $2T_D/(T_M + 4T_D)$ and 100 percent for the CCSMIN and OCSMIN, respectively. Essentially, these performance indexes describe the system's capability.

B. Mean System Access Time

NOMI can shorten the network delay with a minor hardware overhead. However, this is *not* a proper way to evaluate the design since requests could be blocked when resource contention occurs. Thus, the evaluation must also include the blocking factor to indicate the network's blocking property.

Definition 5: The *mean system access time* is defined as sum of the mean time for establishing a path and the fixed memory access time.

Assume that i) the system has reached steady state and the network pass rate also reaches some fixed value, and ii) when a processor request fails to establish its path in some cycle, it will try again at subsequent cycles until it succeeds. Usually, operation of the CCSMIN is not synchronized with memory modules. After a path has been established successfully, there is a delay before memory can actually be accessed, which is assumed to be $T_M/2$. Networks with assumption ii) are shown to provide better performance than those with blocked requests held on switches [21]. There exists a similar delay of $T_M/2$ in the OCSMIN before the desired phase occurs. Thus, the mean system access time can be expressed as

$$\begin{aligned}\overline{T}_O &= \int_0^1 \Big[2T_D + \frac{1}{2}T_M + P_A T_M + (1 - P_A)P_A 2T_M \\ &\qquad + (1 - P_A)^2 P_A 3T_M \cdots \Big] dF(P) \\ &= \int_0^1 \Big[2T_D + \frac{1}{2}T_M - T_M P_A \frac{\partial}{\partial P_A} \sum_{n=1}^{\infty} (1 - P_A)^n \Big] dF(P) \\ &= \int_0^1 \Big[2T_D + \Big(\frac{1}{P_A} + \frac{1}{2}\Big) T_M \Big] dF(P) \qquad (1)\end{aligned}$$

and

$$\begin{aligned}\overline{T}_C &= \int_0^1 \Big[T_M + \frac{1}{2}T_M + P_A 2T_D + (1 - P_A)P_A(2T_D + T_D) \\ &\qquad + (1 - P_A)^2 P_A(3T_D + T_D) \cdots \Big] dF(P) \\ &= \int_0^1 \Big[\frac{3}{2}T_M + \frac{P_A}{1 - P_A} T_D \Big\{ \sum_{n=1}^{\infty} n(1 - P_A)^{n-1-1} \Big\} \Big] dF(P) \\ &= \int_0^1 \Big[\frac{3}{2}T_M + T_D \frac{1 + P_A}{P_A} \Big] dF(P) \qquad (2)\end{aligned}$$

where F is the distribution function of the request rate.

The integration in (1) and (2) can be eliminated when the request rate P is fixed. In blocking multistage networks, P_A is a monotonically decreasing function of the network width. T_D is determined by both the switch delay and the cable delay. As shown in (1) and (2), the mean system access times $\overline{T}_O$ and $\overline{T}_C$ are very sensitive to the network pass rate. For example, assuming the memory access time to be 320 ns and the network delay to be 120 ns, with a 30 percent pass rate for a random memory access pattern, the mean system access time becomes 1000 ns. The system parameters such as the network delay, network size, etc., interact with one another in a complicated form. NOMI can always be applied to crossbar networks without increasing the mean system access time. On the other hand, the system parameters of blocking networks can be tuned so that the mean system access time may not increase. This can be done as follows:

$$\begin{aligned}\overline{T}_C - \overline{T}_O &= \frac{3}{2}T_{M_C} + T_{D_C}\frac{1 + P_C}{P_C} - \Big(2T_{D_O} + \frac{T_{M_O}}{P_O}\Big) \\ &= T_{D_C}\Big(1 + \frac{1}{P_C} - 2d\Big) + T_{M_C}\Big(\frac{3}{2} - \frac{m}{P_O}\Big)\end{aligned}$$

where $d \equiv (T_{D_O}/T_{D_C})$ and $m \equiv (T_{M_O}/T_{M_C})$. The mean system access time of the OCSMIN becomes superior to that of the CCSMIN if $d < ((1 + P_C)/2P_C)$ and $m < (3P_O/2)$. For example, in crossbar switching networks the pass rate is almost

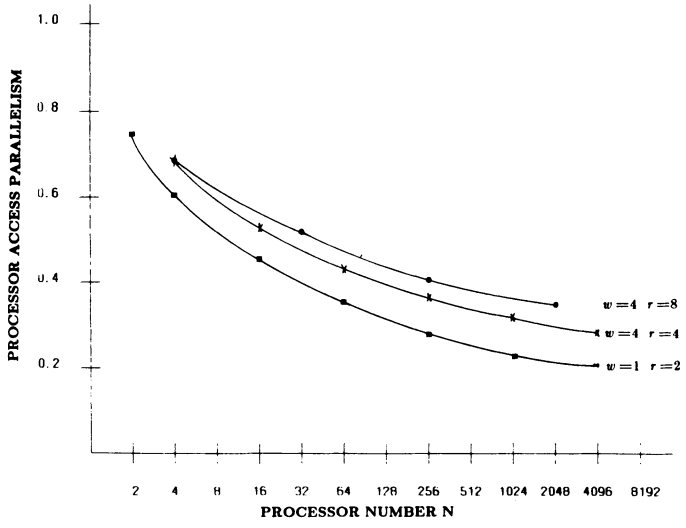


Fig. 10. Comparison of access parallelisms between CCSMIN and OCSMIN when $P = 1$.

constant (≈ 0.7) [17], and $P_C < 1, d < 1$, thereby satisfying the above condition. Since the pass rate of blocking networks varies widely, the NOMI design must be tuned to avoid the performance penalty.

C. Execution/Access Parallelism

The parallelism among processors of an MIMD machine must be examined from the viewpoints of the processor and memory subsystems. From the memory subsystem's point of view, memory accesses within one cluster are interleaved. Simultaneous accesses from a processor cluster to two or more memory modules are not allowed. Since every memory module can be accessed immediately once its present cycle is completed, there is no loss in performance of the memory subsystem.

From the processor subsystem's point of view, there are two problems to be addressed. The first is the *execution parallelism,* which represents the processors' capability of simultaneous execution of instructions. The second is the *access parallelism,* i.e., simultaneous memory access capability. Clearly, processors with CCSMIN's and OCSMIN's have an identical execution parallelism due to their mutual independence.

Three access patterns of processors within a cluster are used below to compare the access parallelism of OCSMIN's and CCSMIN's.

Case 1: When a memory module is requested by more than one processor, then only one processor is permitted.

Case 2: Each processor accesses memory modules of different phases.

Case 3: All processors need memory modules of the same phase.

The addressed memory module in Case 1 must be accessed by processors serially in both cases of CCSMIN's and OCSMIN's. The *logical interleaving* of processors described in Section II-B is achieved when the access pattern is the same as in Case 2; thus, the access parallelism among processors is maximized. When Case 3 alone is considered, the OCSMIN appears to lose the access parallelism to $1/w$ of the CCSMIN. This, of course, is not true since the effect of the pass rate on the parallelism must also be included, i.e., $(N/w)P_{A'}$ has to be compared to $(N/w)P_A$. Note that for the access pattern corresponding to Case 3 in CCSMIN's, there are only (N/w) processors allowed to access memory modules simultaneously. Thus, the ratio $(P_{A'}/P_A)$ indicates the actual loss/benefit that is obtained from the design of an OCSMIN.

The more general cases are when request patterns are random. Note that a phase in the OCSMIN is equivalent to one *output port* of a switch. The analytical models in Section IV-A can still be adapted, where a processor cluster is treated as a $w \times w$ switch. We further assume that each phase ph_j is equally requested by processors. Then, at each phase ph_j, the probability a request appears on the interface pn_i is $P_{ph_i} = 1 - (1 - (P/w))^w$. If P_{ph_i} is the actual request rate, then the rest of the analysis is identical to that of the blocking factor. Proper depths of NOMI can improve the access parallelism. Consider a simple example system consisting of 1024 processors with a four-way NOMI using 4×4 switches. Its performance is better than a 2×2 configuration, but about the same as that of the 4×4 configuration. Comparison among different configurations is shown in Fig. 10. Usually, system performance heavily depends on the software arrangement. As shown in [3], the data storage in SIMD machines is often *skewed* to avoid the direct access conflict. The same principle could be applied to NOMI.

V. Conclusion

We have proposed in this paper a new approach to improving the cost-effectiveness of multistage interconnection networks. A significant improvement of the cost–performance is made by the overlapping and interleaving technique. This technique shows its usefulness in matching the different bandwidth requirements of subsystems in a large multiprocessor system. The drastic reduction in the number of switches/gates makes the use of high-order switches economically acceptable. It also shortens the network delay because of the smaller number of stages required.

The cost analysis of an OCSMIN in this paper has only provided a lower bound for the improvement made by NOMI since it did not include the length of data links. The link length can be evaluated only via modeling the physical construction of the network, and such an analysis is closely related to the issue of system optimization. In addition to its impact on the system cost and performance that we studied here, the NOMI technique appears to have a significant influence on the system's fault tolerance, reliability, etc. These are all interesting and a matter of our future research.

Notation

pc_i	processor cluster i,
mc_i	memory cluster i,
p_{ij}	processor located in processor cluster i with phase number j,
m_{ij}	memory module located in memory cluster i with phase number j,
pn_i	network interface unit of processor cluster i,
mn_i	network interface unit of memory cluster i,
r	order of a switch,
w	depth of overlapping, which is always associated with the memory interleaving of the same depth,
CSMIN	circuit switching multistage interconnection network,
CCSMIN	conventional circuit switching multistage interconnection network,
OCSMIN	overlapped circuit switching multistage interconnection network,
NOMI	network overlapping and memory interleaving,
$FS_{ij}(k, m)$	a forward switch located at (i, j) and its input port k and output port m are used to form an interconnection,
$BS_{ij}(k, m)$	a backward switch located at (i, j) and its output port k and input port m are used to form an interconnection,
FL_{ij}	a forward link located at (i, j),
BL_{ij}	a backward link located at (i, j),
$A(t)$	the request vector appears on the processor subsystem at time t,
$\Pi(E)$	the partner of resource E on the forward (or backward) network,
${}_iFP_j$	the set of resources on the forward network to support a path from processor cluster i to memory cluster j,
${}_iBP_j$	the set of resources on the backward network to support a path from processor cluster i to memory cluster j,
Δt	the time interval before the partner of a forward switch (link) is locked after the forward switch (link) is locked,
$CSF(t)$	interconnection function of a conventional circuit switching network,
$OVF(t)$	interconnection function of an overlapped circuit switching network,
T_M	memory subsystem (memory modules and interface) cycle time,
T_D	network propagation delay,
T_P	processor's memory–request cycle time,
N	number of processor (or memory) modules,
$N' = (N/w)$	number of processor (or memory) clusters,
P	processor request rate, which is a random variable, $0 \leq P \leq 1$ and $\int_0^1 dF(P) = 1$.
P_{A_r}	acceptance rate for a network of configuration r.[5]

Acknowledgment

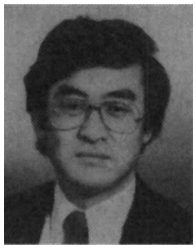
The authors wish to thank L. O. Tiffany and M. H. Woodbury for their comments on the initial draft of this paper. They are also indebted to D. Mizell at the Office of Naval Research for assistance given to the work reported in this paper.

References

[1] C. D. Thompson, "Generalized connection networks for parallel processor intercommunication," *IEEE Trans. Comput.*, vol. C-27, pp. 1119–1125, Dec. 1978.
[2] C. Clos, "A study of nonblocking switching networks," *Bell Syst. Tech. J.*, vol. 32, pp. 406–424, 1953.
[3] K. Hwang and F. A. Briggs, *Computer Architecture and Parallel Processing*. New York: McGraw-Hill, 1984.
[4] T.-Y. Feng, "A survey of interconnection network," *Computer*, pp. 12–27, Dec. 1981.
[5] H. J. Siegel and R. J. McMillen, "The multistage cube: A versatile interconnection network," *Computer*, pp. 65–76, Dec. 1983.
[6] M. A. Franklin, S. A. Kahn, and M. J. Stucki, in "Design issue in the development of a multiprocessor communication network," in *Proc. 6th Annu. Symp. Comput. Architecture*, 1979, pp. 182–187.
[7] H. Inose, *An Introduction to Digital Integrated Communication Systems*. Tokyo, Japan: Univ. Tokyo Press, 1979.
[8] J. Bellamy, *Digital Telephony*. New York: Wiley, 1982.
[9] Special Issue on No. 4 ESS, *Bell Syst. Tech. J.*, vol. 56, Sept. 1979.
[10] R. Bianchini and R. Bianchini, Jr., "Wireability of an ultracomputer," New York Univ., New York, Ultracomputer Note 43, 1982.
[11] A. Gottlieb, R. Grishman, C. P. Kruskal, K. P. McAuliffe, L. Rudolph, and M. Snir, "The NYU ultracomputer—Designing an MIMD shared memory parallel computer," *IEEE Trans. Comput.*, vol. C-32, pp. 175–189, Feb. 1983.
[12] D. F. Wann and M. A. Franklin, "Asynchronous and clocked control structures for VLSI based interconnection networks," *IEEE Trans. Comput.*, vol. C-32, pp. 284–293, Mar. 1983.
[13] Z. Kohavi, *Switching and Finite Automata Theory*. New York: McGraw-Hill, 1978.
[14] B. Prince and G. Due-Gundersen, *Semiconductor Memories*. New York: Wiley, 1983.
[15] G. H. Barnes and S. F. Lundstrom, "Design and validation of a connection network for many-processor multiprocessor systems," *Computer*, pp. 31–41, Dec. 1981.
[16] J. B. Dennis, "Data flow supercomputers," *Computer*, pp. 48–56, Nov. 1980.
[17] J. H. Patel, "Performance of processor–memory interconnections for multiprocessors," *IEEE Trans. Comput.*, vol. C-31, pp. 771–780, Nov. 1981.
[18] D. W. L. Yen, J. H. Patel, and E. D. Davidson, "Memory interference in synchronous multiprocessor systems," *IEEE Trans. Comput.*, vol. C-31, pp. 1116–1121, Nov. 1982.
[19] B. R. Rau, "Interleaved memory bandwidth in a model of a multiprocessor computer system," *IEEE Trans. Comput.*, vol. C-28, pp. 678–681, Sept. 1979.

[5] It is independent of the value of w.

[20] D. P. Bhandarkar, "Analysis of memory interference in multiprocessors," *IEEE Trans. Comput.*, vol. C-24, pp. 897–908, Sept. 1975.
[21] M. Lee and C. L. Wu, "Performance analysis of circuit switching baseline interconnection network," in *Proc. 10th Annu. Symp. Comput. Architecture*, June 1984, pp. 82–90.



Kang G. Shin (S'75–M'78–SM'82) received the B.S. degree in electronics engineering from Seoul National University, Seoul, Korea, in 1970 and the M.S. and Ph.D. degrees in electrical engineering from Cornell University, Ithaca, NY, in 1976 and 1978, respectively.

From 1970 to 1972 he served in the Korean Army as an ROTC officer and from 1972 to 1974 he was on the research staff of the Korea Institute of Science and Technology, Seoul, Korea, working on the design of VHF/UHF communication systems. From 1974 to 1978 he was a Teaching/Research Assistant and then an Instructor in the School of Electrical Engineering, Cornell University. From 1978 to 1982 he was an Assistant Professor at Rensselaer Polytechnic Institute, Troy, NY. He was also a visiting scientist at the U.S. Air Force Flight Dynamics Laboratory in the summer of 1979 and at Bell Laboratories, Holmdel, NJ, in the summer of 1980 where his work was concerned with distributed airborne computing and cache memory architecture, respectively. He also taught short courses for the IBM Computer Science Series in the area of computer architecture. Since September 1982 he has been with the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, where he is currently an Associate Professor. His current teaching and research interests are in the areas of distributed and fault-tolerant computing, computer architecture, and robotics and automation.

Dr. Shin is a member of the Association for Computing Machinery, Sigma Xi, and Phi Kappa Phi.

Jyh-Charn Liu (S'84) was born in Kaohsiung, Taiwan, Republic of China, on December 6, 1956. He received the B.S. and M.S. degrees in electrical engineering from the National Cheng Kung University, Tainan, Taiwan, in 1979 and 1981, respectively.

Since 1984 he has been a Research Assistant at the University of Michigan, Ann Arbor, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. His research interests include fault-tolerant computing, high-performance architecture, and multiprocessor systems.

Mr. Liu is a student member of the IEEE Computer Society.