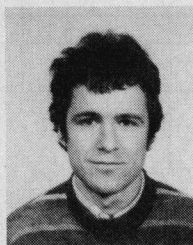


ACKNOWLEDGMENT

I wish to thank J. Bentley, who suggested this problem to me and sparked my interest in it. My gratitude also goes to one of the referees for pointing out the simplification in the procedure SLIDE, possible when rectangles are packed with decreasing widths.

REFERENCES

- [1] B. S. Baker, D. J. Brown, and H. P. Katseff, "A 5/4 algorithm for two-dimensional packing," *J. Alg.*, vol. 2, pp. 338-368, 1981.
- [2] B. S. Baker, E. G. Coffman, and R. L. Rivest, "Orthogonal packings in two dimensions," *SIAM J. Comput.*, vol. 9, pp. 846-855, 1980.
- [3] B. S. Baker and J. S. Schwarz, "Shelf Algorithms for two-dimensional packing problems," in *Proc. 1979 Conf. Inform. Sci. Syst.*, Baltimore, 1979.
- [4] E. G. Coffman, M. R. Garey, D. S. Johnson, and R. E. Tarjan, "Performance bounds for level-oriented two-dimensional packing algorithms," *SIAM J. Comput.*, vol. 9, pp. 808-826, 1980.
- [5] D. E. Knuth, *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*. Reading, MA: Addison-Wesley, 1968.
- [6] J. D. Ullman, "Complexity of sequencing problems," in *Computer and Job-Shop Scheduling Theory*, E. G. Coffman, Ed. New York: Wiley, 1975.



Bernard Chazelle received the Diplôme d'ingénieur from the Ecole Nationale Supérieure des Mines de Paris, France, in 1977, and the M.S. and Ph.D. degrees in computer science from Yale University, New Haven, CT, in 1978 and 1980, respectively.

From 1980 to 1982 he was a Research Associate in the Department of Computer Science at Carnegie-Mellon University, Pittsburgh, PA. In September 1982, he joined the Department of Computer Science at Brown University, Providence,

RI, where he is currently an Assistant Professor. His research interests include analysis of algorithms, complexity theory, computational geometry, VLSI, and graphics.

Dr. Chazelle is a member of the Association for Computing Machinery.

Reduction of Connections for Multibus Organization

TOMÁS LANG, MATEO VALERO, AND MIGUEL A. FIOI

Abstract—The multibus interconnection network is an attractive solution for connecting processors and memory modules in a multiprocessor with shared memory. It provides a throughput which is intermediate between the single bus and the crossbar, with a corresponding intermediate cost.

The standard connection scheme for the multibus connects all processors and all memory modules to all buses. This connection scheme is redundant and expensive for a relatively large number of buses.

Reduced connection schemes that produce the same throughput as the standard connection are presented. The schemes are optimal with respect to the number of connections, are easy to arbitrate, reliable when a bus fails, and expandable. The reduction is specially significant when the number of buses is relatively large, being of 25 percent when this number is half the number of memory modules.

Index Terms—Arbitration, connection reduction, interconnection network, multiple buses, multiprocessors.

Manuscript received November 11, 1981; revised December 17, 1982.

T. Lang was with the Facultat d'Informàtica, Universitat Politècnica de Barcelona, Barcelona, Spain. He is now with the Department of Computer Science, University of California, Los Angeles, CA 90024.

M. Valero is with the Facultat d'Informàtica, Universitat Politècnica de Barcelona, Barcelona, Spain.

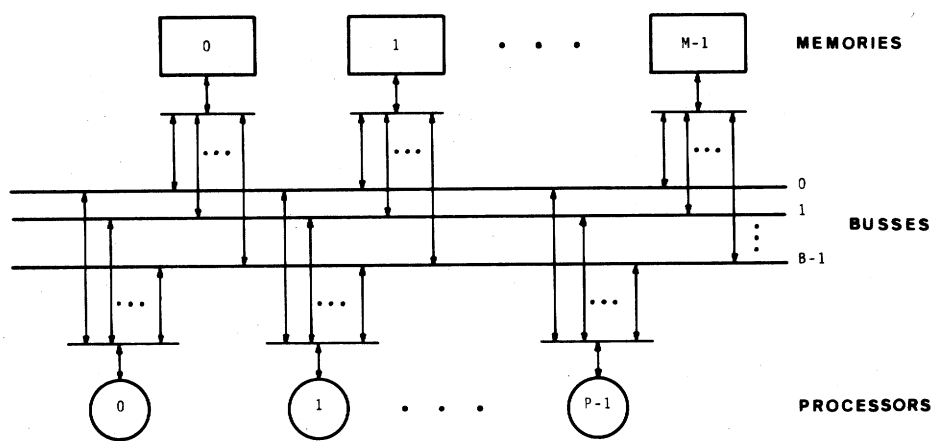
M. A. Fiol is with the School of Telecommunication Engineering, Universitat Politècnica de Barcelona, Barcelona, Spain.

I. INTRODUCTION

ONE of the many important aspects to consider in the design of multiprocessor systems is the structure of the network connecting the processors to the shared memory modules. Many parameters have a bearing on this choice. Among them: reliability, cost, modularity, bandwidth, number of processors, and expandability.

Several interconnection networks have been proposed, such as the crossbar [1], single bus [2], multibus [3], [4], and other special interconnection networks [5]. There are several analytic models to assess the performance of the various topologies under different processor demand patterns [3], [6], [7].

The multibus interconnection is an attractive solution for connecting processors and memory modules in a multiprocessor with shared memory. It provides a throughput which is intermediate between the single bus and the crossbar, with a corresponding intermediate cost. Moreover, if the processor requests are independent and uniformly distributed among the memory modules, the amount of memory conflicts makes the throughput obtained with the crossbar roughly the same as that obtained with the multibus with a number of buses slightly larger than half the number of processors [4].



(a)

Memory Modules

0	1	2	...	M-2	M-1
B-1	B-1	B-1	...	B-1	B-1
B-2	B-2	B-2	...	B-2	B-2
.
.
2	2	2	...	2	2
1	1	1	...	1	1
0	0	0	...	0	0

Busses

(b)

Processors

0	1	2	...	P-2	P-1
B-1	B-1	B-1	...	B-1	B-1
B-2	B-2	B-2	...	B-2	B-2
.
.
2	2	2	...	2	2
1	1	1	...	1	1
0	0	0	...	0	0

Busses

(c)

Fig. 1. (a) Complete multibus interconnection scheme. (b) Matrix representation of complete interconnection (memory-buses part). (Entries indicate the buses numbers connected to a memory module.) (c) Matrix representation of complete interconnection (processors-buses part).

The multibus interconnection has been studied in [3], [8]. A fast and modular arbiter for this network has been proposed in [9].

The standard connection scheme for the multibus case is illustrated in Fig. 1 where all P processors and all M memory modules are connected to all B buses. The resulting number of connections is $B(M + P)$, which can be large and result in a costly network. In this paper, we show that this connection is redundant and that reduced schemes produce the same throughput at a lower cost. The reduction is especially significant when the number of buses is relatively large. For example, the number of connections required is reduced by approximately 25 percent when the number of buses is half the number of processors (for $M = P$).

To show the practicality of these reduced schemes, we investigate their cost (measured by the number of connections and the memory and bus loads), the arbitration complexity and speed, the reliability (as the possibility of functioning in a degraded form when a bus fails), and the expandability (that is, the reconfiguration required to increase the number of buses, memory modules, and processors). We conclude that there are connection schemes that adequately satisfy these requirements.

II. REDUCTION IN THE NUMBER OF MEMORY-BUS CONNECTIONS

As mentioned in the Introduction, the complete interconnection of Fig. 1 is redundant. As an example, consider the

Memory Modules															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
							7	7	7	7	7	7	7	7	7
						6	6	6	6	6	6	6	6	6	6
					5	5	5	5	5	5	5	5	5	5	5
				4	4	4	4	4	4	4	4	4	4	4	4
			3	3	3	3	3	3	3	3	3	3	3	3	3
		2	2	2	2	2	2	2	2	2	2	2	2	2	2
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 2. Trapezoidal interconnection. (For 16 memory-modules, 8 buses.)

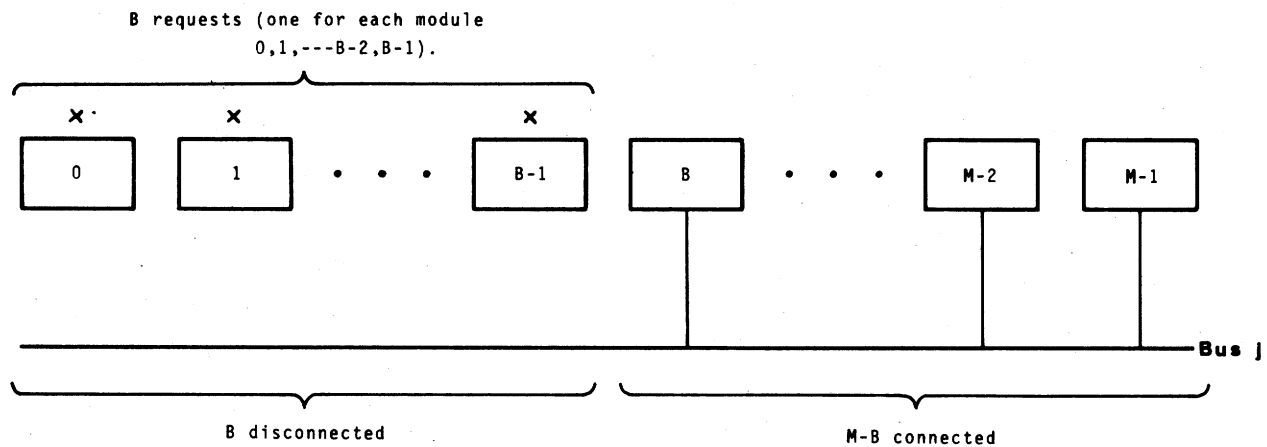


Fig. 3. Illustration for proof of Theorem 1.

Memory Modules															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
							7	7	7	7	7	7	7	7	7
						6	6	6	6	6	6	6	6	6	6
					5	5	5	5	5	5	5	5	5	5	5
			3	3	3	3	3	3	3	3	3	3	3	3	3
		2	2	2	2	2	2	2	2	2	2	2	2	2	2
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

x indicates requested module; modules 0,1,2,3,4 cannot be assigned simultaneously.

Fig. 4. Connection satisfying lower bound but with degradation.

reduction of connections between memory modules and buses shown in Fig. 2 (trapezoidal connection-scheme). It is straightforward to show that any B memory modules can be connected to the B buses, and therefore that the throughput of the trapezoidal scheme is the same as that of the complete connection. A possible assignment algorithm would be to assign the buses in ascending order to memory modules also in ascending order. At this point, we are only interested in showing that an assignment algorithm exists: in Section V we consider fair algorithms which lead to simple arbiter implementations.

The trapezoidal connection is a reduced scheme which provides the same throughput as the complete connection but, as indicated by the following theorem, it is not minimal.

Theorem 1: A lower bound on the number of memory modules connected to a bus is $M - B + 1$ (assuming that all processors are connected to all buses).

Proof: By contradiction. Suppose that $M - B$ connections for bus j are sufficient. Then there are B modules not connected to that bus and, if the requests correspond to these B modules, then bus j cannot be assigned (Fig. 3). The throughput is therefore degraded.

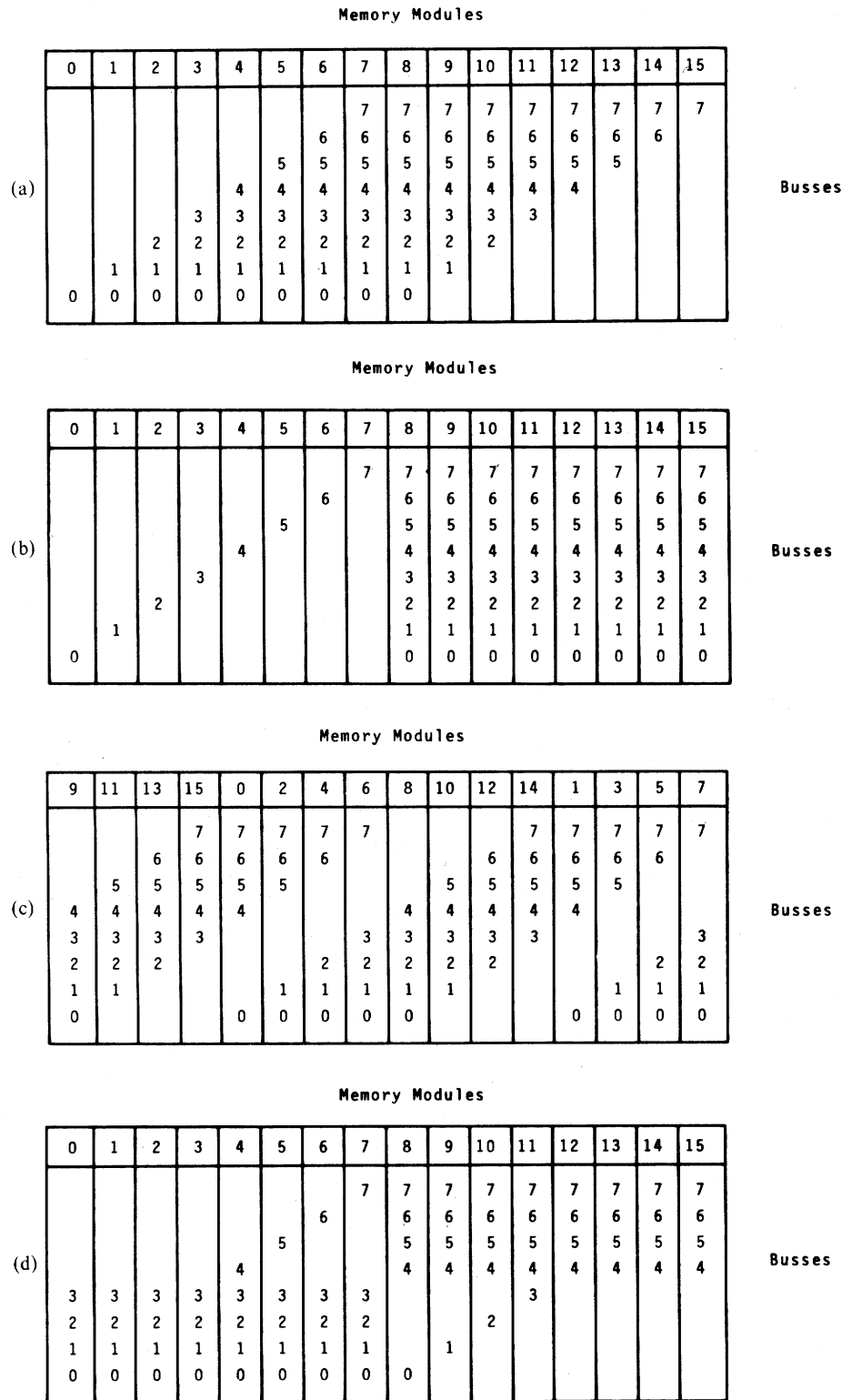


Fig. 5. (a) Rhombic interconnection. (b) Staircase connection. (c) Cyclic interconnection. (d) Balanced interconnection.

Of course, these minimal connections have to follow a pattern so that the buses can be assigned to any B memory modules. A necessary, but not sufficient, condition is that any set of B memory modules is connected to B buses. That this condition is not sufficient is shown by the example of Fig. 4.

In the next section, we present several schemes that produce no degradation in throughput and are minimal.

III. MINIMAL MEMORY-BUS CONNECTION SCHEMES AND THEIR COSTS

In Fig. 5 we present several connection schemes that are minimal: the rhombic, the staircase, and the balanced schemes (for 16 memory-modules, 8 buses). They all satisfy the condition that, for any B requests, the B buses can be assigned.

TABLE I
COST PARAMETERS FOR REDUCED SCHEMES

Scheme	No of connections	Max. Bus load	Max. Memory load
Complete	$B(P+M)$	$P+M$	B
Trapezoidal	$B \left[P+M - \frac{B-1}{2} \right]$	$P+M$	B
Rhombic	$B[P+M-(B-1)]$	$P+M-B+1$	B if $M > 2B$ $M-B$ if $M < 2B$
Staircase	$B[P+M-(B-1)]$	$P+M-B+1$	B
Cyclic	$B[P+M-(B-2)]$	$P+M-B+2$	$\left[\frac{(M-B+2)B}{M} \right] = 5$
Balanced	$B[P+M-(B-1)]$	$P+M-B+1$	$\frac{B}{2} + 1$

In addition, in Fig. 5(d) we show a cyclic scheme. This is not minimal (each bus is connected to ten memory modules), but it has the important characteristic that each memory module is connected to the same number of buses.

The cost of these schemes is presented in Table I, taking into account the parameters indicated in the introduction: number of connections, and memory-module and bus loads. All the minimal schemes have the same number of connections, but they differ in the loads. With respect to maximum memory loads, the rhombic is better than the staircase for large values of B . The best are the cyclic and the balanced.

IV. REDUCTION IN THE NUMBER OF CONNECTIONS BETWEEN MEMORY AND BUSES AND BETWEEN BUSES AND PROCESSORS

In Section III, we discussed minimal schemes for the memory-bus connection assuming that the processor-bus scheme is complete. Of course, the same type of reduction can be obtained in the processor-bus connection if the memory-bus scheme is complete. We now consider what happens if we reduce both connections simultaneously.

Theorem 2: Consider bus i and let m_i and p_i be the number of memory modules and processors connected to it. Then, for no degradation, m_i and p_i have to satisfy the following restrictions:

- 1) $M - B + 1 \leq m_i \leq M$
- 2) $(P + M + 1) - (B + m_i) \leq p_i \leq P$.

Proof: Part 1) has already been proved in Theorem 1. For part 2), if bus i is connected to m_i memory modules, it is disconnected from $M - m_i$. If B memory modules are requested, among which are the $M - m_i$ that are not connected to bus i , it is necessary to select one among the other $B - (M - m_i)$ to assign to bus i (note: because of 1), $B - (M - m_i) \geq 1$).

Since it is necessary to be able to connect any processor to any memory, the selection of one of the $B - (M - m_i)$ memory modules requires that no more than $[B - (M - m_i)] - 1$ processors be disconnected from bus i . Therefore

$$p_i \geq P - [B - (M - m_i)] + 1.$$

Corollary: If each bus is connected to m memory modules, the minimal total number of connections is

$$Bm + B[P + M + 1 - (B + m)] = B[P + M - B + 1].$$

Note that this number is independent of m . Therefore, a minimal solution is obtained by minimizing one set of connections (memory-bus or bus-processor) and keeping the other complete.

Due to this corollary, in the following we continue to consider the schemes presented in Section III.

V. ARBITRATION METHODS

We now discuss the arbitration algorithms that are applicable to the presented connection schemes. For some schemes the corresponding algorithms seem easy to implement and, therefore, these schemes are of practical interest. For others, no simple algorithm has yet been found.

The arbitration algorithm should be fair, in the sense that all processors should have the same probability of accessing memory, and should be easy to implement. These two requirements are sometimes conflicting; in these cases, it might be convenient to divide the arbitration process into two parts:

- 1) A fair selection process, which selects $\min(B, R)$ memory modules from the R memory modules that have at least one pending request.
- 2) An assignment process, which assigns the buses to the selected memory modules.

The selection process does not depend on the connection scheme. A fair cyclic algorithm is easily implementable with a fast and modular circuit.

The bus-assignment process depends on the connection scheme. We now investigate assignment algorithms that lead to a simple implementation.

For the complete and trapezoidal schemes, the assignment can be performed by assigning an increasing bus number to an increasing module number. This is formalized in the following algorithm. $S(j) = 1$ indicates that the j th module has been selected by the selection process and the value of $B(i)$ specifies the memory module to which bus i is assigned:

```

i = 0
FOR j = 0 until M - 1 DO
  If S(j) = 1 THEN
    BEGIN
      B(i) = j;
      i = i + 1;
    END.

```

This algorithm is easy to implement and results in a modular and fast network [9].

For the rhombic scheme, a possible algorithm is

```

i = 0
FOR j = 0 to M - B DO
  IF S(j) = 1 THEN
    BEGIN
      B(i) = j;
      i = i + 1;
    END
FOR j = M - B + 1 to M - 1 DO
  IF S(j) = 1 THEN
    BEGIN
      If i > (j - (M - B))
      THEN
        B(i) = j;
        i = i + 1;
      ELSE
        B(j - (M - B)) = j;
        i = j - (M - B) + 1
      END.

```

This algorithm is also simple to implement. It represents a small variation with respect to the implementation for the trapezoidal case.

For the staircase scheme an algorithm is

```

k = 0
FOR j = 0 to B - 1 DO
  IF S(j) = 1 THEN B(j) = j
  ELSE
    A(k) = j;
    k = k + 1;
k = 0
FOR j = B to M - 1 DO
  IF S(j) = 1 THEN B(A(k)) = j;
    k = k + 1.

```

This algorithm is a little more complex to implement because the information of which buses are not assigned in the first part has to be transmitted to the second.

For the cyclic scheme, as illustrated by Fig. 5(c), there does not seem to exist a simple algorithm. It appears to be necessary to inspect all selected requests to make the assignments. Nevertheless, for an adequate renumbering of the memory modules, an algorithm has been obtained that is acceptable. At present we are working on the improvement and generalization of this algorithm.

For the balanced scheme, the assignment algorithm seems to be complex to implement in hardware.

As a conclusion, we see that from the point of view of arbi-

tration, the rhombic and the staircase schemes are acceptable.

VI. RELIABILITY

We now consider the reliability of the connection schemes. We modify the connections so that the system can operate in a degraded form when a bus fails.

The complete connection does not require any modification. If a bus fails, the arbiter would assign the remaining $B - 1$ buses. As these buses are connected to all memory modules, no connection problem arises.

The trapezoidal connection has to be modified to permit this degraded operation. This can be seen by the fact that if there are $B - 1$ buses, Theorem 1 requires that every bus be connected to at least $M - B + 2$ memory modules, and this is not the case for bus number $B - 1$ (which is connected to $M - B + 1$ memory modules). Also memory module 0 is connected only to bus 0 so that it would be completely disconnected if bus 0 fails. Consequently, the minimum modification required is to connect bus $B - 1$ also to module 0, resulting in the connection scheme indicated in Fig. 6(a). It is easy to see that in this case, the assignment of the $B - 1$ buses can always be done. The arbitration procedure has to be changed as follows: assign bus $B - 1$ to memory module 0 (unless this bus fails), and then assign from bus 0 in ascending order. Of course, it is necessary to skip the failed bus.

The rhombic connection has to be modified similarly. To satisfy the theorem, one additional connection has to be included for each bus. Fig. 6(b) shows a modified connection scheme that allows the degraded operation. The arbitration algorithm has to be changed, as for the trapezoidal case.

The staircase connection is modified as indicated in Fig. 6(c) in order to satisfy the theorem and assure that a memory module is connected, at least, to two buses. In this case, the arbitration has to be modified considerably. To avoid this, a somewhat redundant connection is shown in Fig. 6(d), which operates in degraded fashion with the same type of modification to the arbitration procedure as for the rhombic case.

We do not show the modification of the cyclic and balanced schemes due to their complicated arbitration algorithms.

We conclude that the rhombic and staircase schemes can be easily modified to provide the required reliability.

VII. EXPANDABILITY

We consider now the reconfiguration that is necessary when the system is expanded by adding a memory module or a bus.

For the complete connection no reconfiguration is necessary, that is, no changes in the connections to other memory modules or buses is required. The new memory module is connected to all buses and the new bus to all memory modules.

Similarly, no reconfiguration is necessary in the trapezoidal case: the new memory module is connected to all buses and the new bus is connected to the memory modules required to form the expanded trapezoidal (Fig. 7).

In the rhombic connection scheme, the connections have to be reconfigured as indicated in Fig. 8. As shown, when a

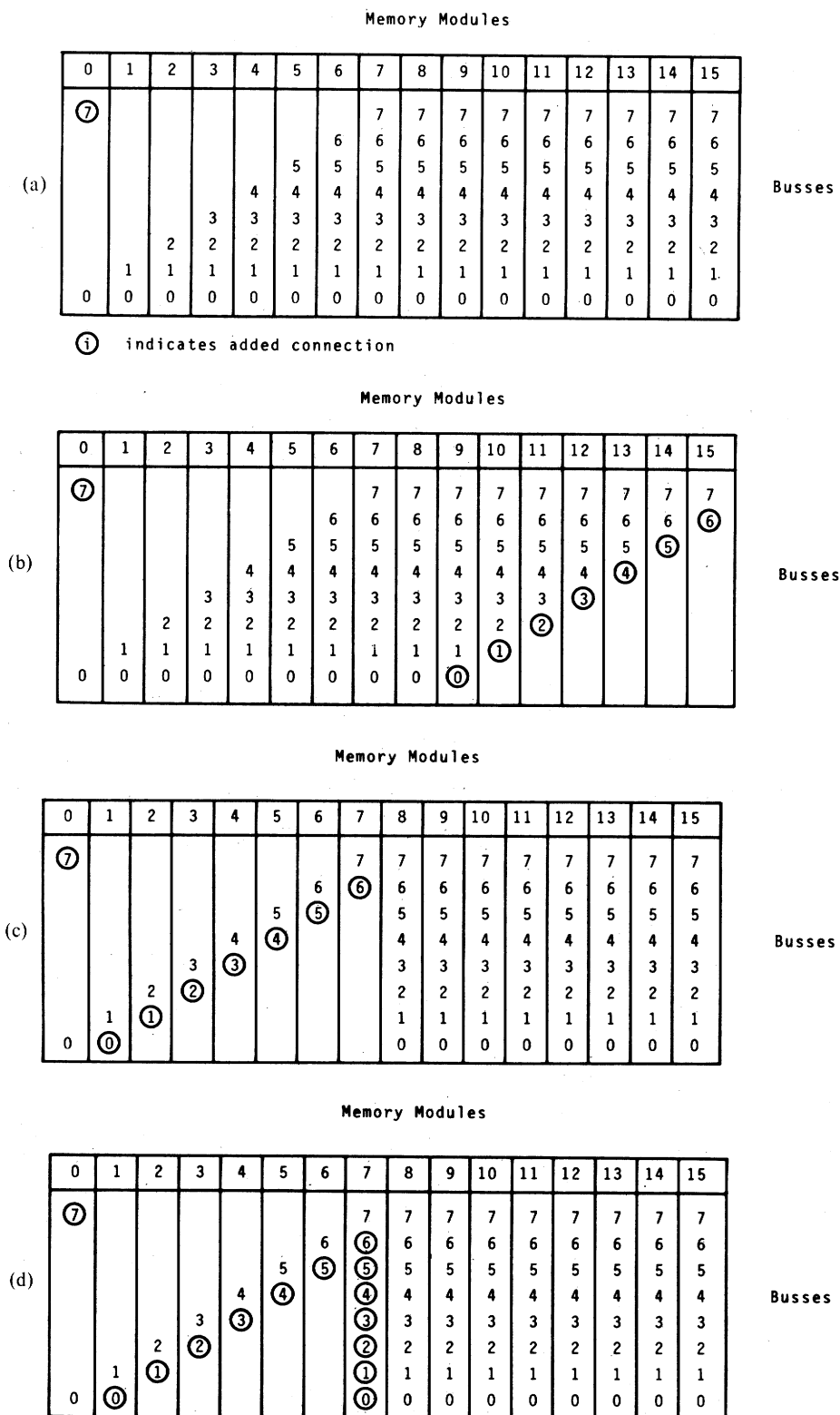


Fig. 6. (a) Trapezoidal interconnection with addition for reliability. (b) Rhombic interconnection with addition for reliability. (c) Staircase interconnection scheme with addition for reliability. (d) Staircase interconnection with addition for reliability and simple arbitration.

memory module is added, connections have to be added to $M - B$ memory modules. On the other hand, when a bus is added, some connections to the other busses can be eliminated.

The reconfiguration required for the staircase scheme is

indicated in Fig. 9. The situation is similar to the trapezoidal scheme when a memory module is added, and to the rhombic case when a bus is added.

We do not discuss the reconfiguration for the cyclic and

Memory Modules															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
								8	8	8	8	8	8	8	8
							7	7	7	7	7	7	7	7	7
						6	6	6	6	6	6	6	6	6	6
					5	5	5	5	5	5	5	5	5	5	5
				4	4	4	4	4	4	4	4	4	4	4	4
			3	3	3	3	3	3	3	3	3	3	3	3	3
		2	2	2	2	2	2	2	2	2	2	2	2	2	2
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(a)

Memory Modules															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
							7	7	7	7	7	7	7	7	7
						6	6	6	6	6	6	6	6	6	6
					5	5	5	5	5	5	5	5	5	5	5
				4	4	4	4	4	4	4	4	4	4	4	4
			3	3	3	3	3	3	3	3	3	3	3	3	3
		2	2	2	2	2	2	2	2	2	2	2	2	2	2
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(b)

Busses

Fig. 7. (a) Adding one bus to the trapezoidal interconnection. (b) Adding one memory module to the trapezoidal interconnection.

balanced case, considering the problems in arbiter implementation.

Consequently, the rhombic and staircase schemes are easily expandable.

VIII. CONCLUSIONS

We have presented several multibus interconnection schemes that are minimal in the number of connections and have adequate characteristics with respect to arbitration, reliability, and expandability. The reduction in the number of connections is significant if the number of buses is relatively large, being of 25 percent for the case in which $B = M/2$ (for $M = P$).

We have shown that the minimal number of connections can be obtained by reducing only the memory-bus part of the network and keeping the bus-processors part complete.

With respect to memory loads, the best schemes are the balanced and the cyclic, but these are difficult to arbitrate. The rhombic and staircase schemes are simple to arbitrate and the rhombic scheme is better with respect to memory loads for $M < 2B$.

REFERENCES

- [1] W. A. Wulf and C. G. Bell "C.mmp—A multi-mini-processor," in *Fall Joint Comput. Conf. AFIPS Conf. Proc.*, vol. 41, pt. 2, 1972, pp. 765–777.
- [2] R. J. Swan *et al.*, "CM*—A modular multimicroprocessor," in *AFIPS Conf. Proc.*, vol. 46, 1977, pp. 637–644.
- [3] M. Ajmone and F. Gregoretti, "Markov models for multiple bus multiprocessor systems," Rep. no. CSD 810304, Univ. Calif. at Los Angeles, Feb. 1981.
- [4] T. Lang *et al.*, "Bandwidth of crossbar and multibus connections for multiprocessors," *IEEE Trans. Comput.*, vol. C-31, pp. 1227–1234, Dec. 1982.
- [5] H. Siegel *et al.*, "A survey interconnection methods for reconfigurable parallel processing systems," in *Proc. 1979 NCC, AFIPS*, vol. 48, June 1979, pp. 529–542.
- [6] 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.
- [7] M. Ajmone *et al.*, "A study on processor-memory interconnection in multimicroprocessor systems," *Alta Freq.*, vol. 50, pp. 120–130, May–June 1981.
- [8] T. Lang and M. Valero, "Reducción de conexiones en organización multibus y arbitrajes asociados," RR 81/07, Facultat d'Informatica de Barcelona, Spain.
- [9] —, "M-users, B-servers arbiter for multibus multiprocessor," *Microprocessing, Microprogramming, Euromicro J.*, vol. 10, pp. 11–18, Aug. 1982.



Tomás Lang received the Engineering degree from the Universidad de Chile in 1964, the M.S. degree from the University of California, Berkeley, in 1966, and the Ph.D. degree from Stanford University, Stanford, CA, in 1974.

He was Professor of Engineering at the Universidad de Chile from 1965 to 1973, was with the faculty of the University of California, Los Angeles, from 1974 to 1978, then taught computer science at the Universitat Politècnica de Barcelona, Spain, from 1978 to 1981, and currently is a

Visiting Associate Professor of Computer Science at U.C.L.A. His teaching and research interests are in computer architecture, with current emphasis in multiprocessor and architectural support for operating systems functions.

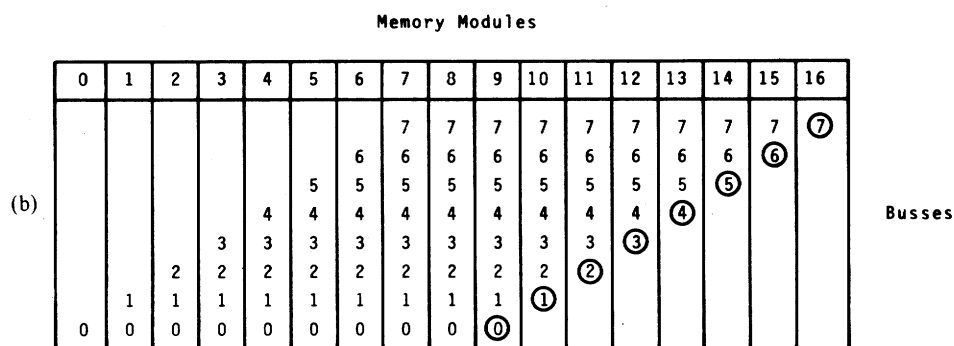
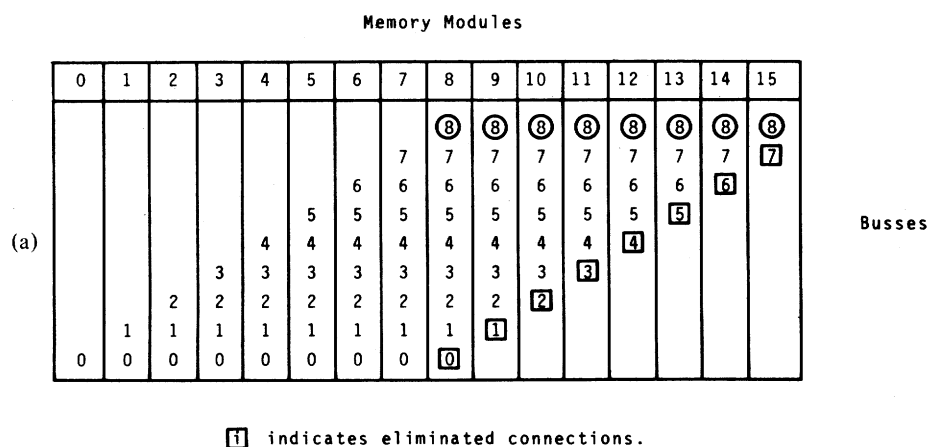


Fig. 8. (a) Adding one bus to the rhombic interconnection. (b) Adding one memory module to the rhombic interconnection.

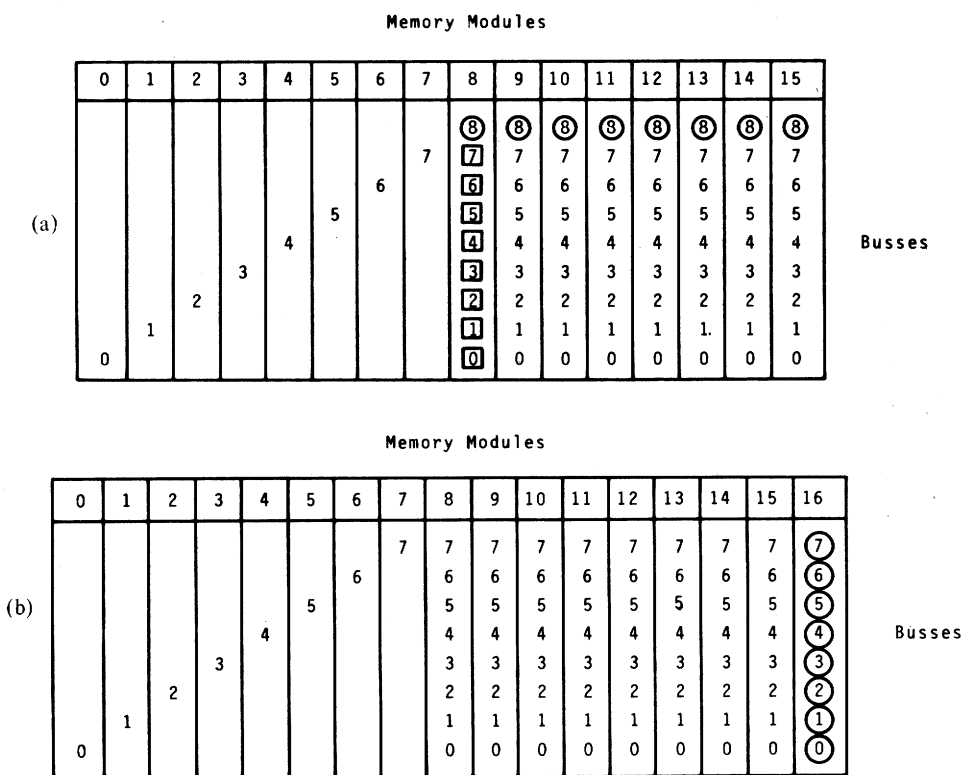
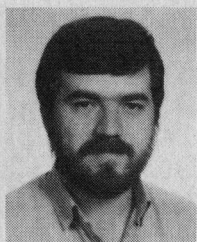


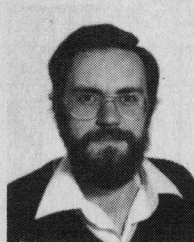
Fig. 9. (a) Adding one bus to the staircase interconnection. (b) Adding one memory module to the staircase interconnection.



Mateo Valero was born at Alfamen-Zaragoza, Spain in 1952. He received the Telecommunication Engineering degree from the Universitat Politecnica de Madrid, Spain, in 1974, and the Ph.D. degree from the Universitat de Barcelona, Spain, in 1980.

He was Professor of Engineering at the Universitat Politecnica de Barcelona (Escuela de Telecomunicacion) from 1974 to 1980. Since 1980 he has been Professor of Computer Architecture at the Facultat d'Informatica de Barcelona (UPB).

His teaching and research interests are in computer architecture, with emphasis in the design and evaluation of interconnection networks for multiprocessor systems and local computer networks.



Miguel A. Fiol was born at Palma de Mallorca, Spain, in 1949. He received the Telecommunication Engineering degree from the Universitat Politecnica de Barcelona, Spain, in 1979, and the Ph.D. degree from the same university in 1982.

Since 1979, he has taught mathematics at the School of Telecommunication Engineers. He has done research on the applications of graph theory to computer science. His current interest is in the design of interconnection networks for multiprocessors and local networks.