ANALYSIS AND SIMULATION OF MULTIPLEXED SINGLE-BUS NETWORKS WITH AND WITHOUT BUFFERING

José Mª Llabería Griñó (*)
Mateo Valero Cortés (*)
Enrique Herrada Lillo (**)
Jesús Labarta Mancho (*)

(*) Facultad de Informática. Barcelona. Spain.
(**) E.T.S.I. Telecomunicación. Barcelona. Spain.

ABSTRACT

Performance issues of a single-bus interconnection network for multiprocessor systems, operating in a multiplexed way, are presented in this paper. Several models are developed and used to allow system performance evaluation. Comparisons with equivalent crossbar systems are provided. It is shown how crossbar EBW values can be reached and exceeded when appropriate operation parameters are chosen in a multiplexed single-bus system. Another architectural feature is considered, concerning the utilization of buffers at the memory modules. With the buffering scheme, memory interference can be reduced so that the system performance is practically improved.

1. INTRODUCTION

In many multiprocessor systems, the shared memory is divided into independent modules, so that some type of interconnection network must exist to provide a communication path between processors and memory modules.

Both the sharing of memory modules, and the interconnection network itself, contribute to the global system loss of efficiency. This performance reduction is due to : 1) memory interference, 2) network degradation (if less than needed number of links is provided) and 3) network arbitration and link switching additional delay times.

One of the first and most widely used interconnection network is the crossbar, (1) . This network does not introduce degradation, but due to memory conflicts its bandwidth is only 0.6 n when the number of processors (n) and the number of memory modules (m) are both large and equal,(1). Because the crossbar cost becomes prohibitive in many real situations, other interconnection networks have been proposed and analyzed to optimize the design at a given cost; that is the case of several degradating networks, such as : single-bus (2,3), multiple-bus (4,5,6) and shuffle-exchange (7).

Multiplexed networks operate in a way that allows the links to be occupied only during the time interval required to the processor requests to reach memory modules, or the memory modules to turn results back to processors. This multiplexed operation makes additional hardware necessary, but allows either greater network bandwidth, or lower number of links, or reduction in memory interference when more memory modules are provided for a given network cost.

Some kind of multiplexed links management has been implemented in several computer architectures: look-ahead processors (i.e. IBM 360-91),(8); data-flow processors (i.e. LAU, (9); recent microcomputer-based systems (i.e.IAPX-432, (10); HP-32 bit,(11). Several published works deal with multiplexed networks evaluation: shuffle-exchange (12) topologies with L-M networks and S-segmented processors (13, 14), among others.
A multiplexed single-bus architecture has been proposed and implemented in the Rapid Bus system (15), where a fixed correspondence is established between the bus cycle and the processor that can use the bus in that cycle; the bus cycle is 10ns, and 25 processors are connected to it. Even higher speeds can be achieved if the memory modules are included in the system chip implementation, as proposed in (16).

In this paper, the performance evaluation of a multiplexed single-bus interconnection network for a multiprocessor system (Fig. 1), is presented. Section 2 includes the system operation hypotheses. In section 3 results obtained through simulation techniques are presented. Exact and approximate models are proposed, in sections 3 & 4, to characterize and evaluate the system under study; two different priority strategies are considered. Numerical results of the analytical models developed in the two previous sections, are provided in section 5, where simulation results are examined for comparison. The addition of buffers at the memory modules inputs and outputs is proposed in section 6, where the efficiency improvements obtained with the buffering technique are evaluated through simulation. Summary and conclusions are provided in the last section of the paper.

2.- SYSTEM OPERATION

To evaluate the system performance, the following operation assumptions are introduced:

a) The system under study is composed of n processors and m memory modules.

0149-7111/85/0000/0414$01.00 © 1985 IEEE

b) The system basic cycle, as well as the bus transfer delay time, is a constant value, t. Arbitration delay is considered to be included in this value.
c) Cycle time of all memory modules is the same and equal to r·t, where r is an integer value.
d) Processor cycle is (r + 2) * t; all the processors are synchronized at the bus cycle.
e) The processors requests are independent and equally distributed among the different memory modules, (21).
f) After receiving the previous memory service, a processor inmediately issues a new request, with probability p; processors submit their request only at the beginning of the processor cycles, (1).
g) At any given instant, the bus may be required both by some processors and by some memory modules. Different perfórmance values are expected, according to bus granting policies. We shall take into account two distinct cases, namely:
g') priority is given to processors.
g'') priority is given to memories.
h) Only the requests issued by the processors toward idle memory modules are considered by the granting bus arbitration policy. A random arbitration schema is used to grant the bus to one of the (possible) several pending requests of the same type.
Accordingly with the above operation assumptions, the system effective bandwidth, EBW, is calculated using the expression:

$$EBW = Pb\ (r + 2)/2,$$

where Pb is the bus utilization, and EBW is the expected number of memory requests serviced per processor cycle.

The maximum value attainable by EBW is, obviously, (r + 2)/2, which compares advantageously with the value 1 that can be reached when the bus is not used in a multiplexed fashion.

Some other related parameters can be derived from EBW, such as the memory module utilization, the mean processor efficiency and the waiting time of every elemental access.

3.- SYSTEM EVALUATION.

By using the assumptions introduced in the previous section it is relatively simple to obtain a preliminary evaluation of the different policies through a simulation model. Figure 2 reports simulation results for the EBW obtained using different values of the parameters n, m and r. In all these cases the processors have been assumed to issue a new request at the beginning of each processor cycle, i.e, p=1. These results show that the EBWs yielded by the bus arbitration policy g') are better than those obtained using policy g''). For comparison purposes, Fig. 2 reports also the EBW yielded by the analysis of a non-multiplexed crossbar interconnection network having a basic operation cycle of length (r+2)t. As it can be seen from this, the multiplexed single bus network provides very good performance values as r increases.

Similar simulation experiments have been run also for the case of a multiprocessor systems with p<1. This situation arises when the processors are allowed to spend several cycles performing some interval processing activity. The actual load on the memory subsystem is in this case represented by n·p. Figure 3 depicts the values of EBW/np obtained for different combinations of the parameters n, m and r.

Because the computational time to run these simulations is large and because the multiplexed single-bus network may be just an element inside a bigger system, to obtain analytical models that predict performance measures of this architecture seems of interest.

3.1.- MODELS WITH PRIORITY TO MEMORY MODULES.

In this section two mathematical models, to evaluate the system EBW, are presented; in these models the bus granting policy is assumed to give priorities to requests coming from the memory modules. In Section 3.1.1 an exact Markov chain is constructed, and in Section 3.1.2. a less expensive, approximate model yielding fairly accurate results, is presented. Both models assume that p=1 (i.e., no internal processing is allowed).

3.1.1. Markov chain.

A detailed system state definition,capable of accomodate any bus granting policy and any value of p ($\leq$ 1), can be devised by using two m-component vectors as follows:

$$(n_1, n_2, \ldots n_m)\ (r_1, r_2, \ldots r_m)$$

where n: $(n_1, n_2, \ldots n_m)$ indicates the distribution of processors requests; n_i represents the number of processor requesting access to the i-th memory module. We have that $\sum_{i=1}^{m} n_i \leq n$, and $n_i \geq 0$ i=1, 2, ...m.

r: $(r_1, r_2, \ldots r_m)$ informs about which is the cycle where the memory operation is in, or about the bus ocupation status to service a request to the memory module.

As we are assuming statistically identical processors and statistically identical memory modules, all the states that are permutations of the same n (such that $n_i \geq n_j$ for any $j > i$), can be grouped together in a single state.

No closed form formula seems to exist to evaluate the exact number of states in this model, given the values of n, m and r. Nevertheless, it can be easily seen that such a model is intractable, even when n,m and r are kept to relatively low values, because the number of states grows exponentially.

In order to make this model tractable, we initially restrict ourselves to the case in which p=1 (i.e, $\sum_{i=1}^{m} n_i = n$), and in which the memory modules have priority over the processors in having the bus granted.

We can disregard the r vector to establish our model, when we consider that the memory modules have the priority over the processors to get the bus control. In this case, the n vector completely defines the state of our system. This kind of definition is coincident with those of (1) to evaluate the crossbar network, and of (5) for multiple-bus interconnection networks.

Moreover, let us now assume that our system state is $(n_1, n_2, \ldots n_m)$ such that, out of these m components, x are non-zero numbers ($0 \leq x \leq m$). That is, the state definition is expressed by the m-vector:

$$(n_1, n_2, \ldots n_x, 0, \ldots 0)$$

Because of the serialization effect induced by the bus, only a maximum of r+1 requests can be completed during a processor cycle. Of course the number of busy memory modules, x, can be smaller than r+1, so that the actual number, K, of requests that can be serviced during a processor cycle is:

$$K = MIN\ (x,\ r+1).$$

According to the relative values of x and r+1, two different situations must be considered in order to evaluate the transition probabilities from the general $(n_1, n_2, \ldots n_m)$ state.

a) $x \leq r + 1$, that is $K = x$.

b) $x > r + 1$, that is $K = r + 1$.

To calculate the state transition matrix, the same method used in (5) is proposed here. Indeed, the evaluation is similar to that presented there for the multiple-bus network, just assuming b(number of buses) to be equal to r + 1.

With a multiple-bus system, the maximum number of requests that can be serviced in a cycle is limited by the number, b, of buses. With a multiplexed single-bus system, the maximum number of requests that can be serviced in a processor cycle becomes r+1, because the bus is granted in the next cycle to the first accessed memory module.

Once the state probabilities are known,the system effective bandwith (EBW), can be expressed as:

$$EBW = \sum_{x=1}^{r+1} \frac{r+2}{r+1+x} \times P(x) + \sum_{x=r+2}^{m} \frac{r+2}{2} P(x)$$

where P(x) represents the $(n_1, n_2, \ldots n_x, 0, \ldots 0)$ state probability, and (r+2)/(r+1+x) and (r+2)/2 represent, respectively in cases a) and b), the useful cycles fraction.

3.2. Approximate Model

In this section we mention a purely combinational model valid for any n,m and r values and which obtains numerical results that approximate quite well those of the exact analysis developed in the previous section.

In order to obtain this approximation, we introduce the simplifying assumption of discarding all the requests directed to busy memory modules. What makes the behaviour of this simplified model different from that employed for the previous exact analysis, is the fact that at the beginning of each processor cycle, all the processors are assumed to submit new independent requests. This memoryless assumption has been widely used in several previous works, among others in (17) for crossbar networks, in (7) for shuffle-exchange networks and in (5) for multiple-bus interconnection networks.

Probability P(x) of serving x requests during a processor cycle can be evaluated by means of an expression as derived in (5) for multiple-bus networks.

4.- MODEL WITH PRIORITY TO PROCESSORS.

An exact Markov chain analysis of this system can be performed using the general state definition introduced at the beginning of section 3.1. However, since the detailed information of the r vector must now be explicitly considered, the state space of this model is larger than that used for the exact analysis of the other priority discipline, and makes this approach impracticable. In this section we will thus concentrate on the derivation of an approximate model, based on Markov chain analysis, under the usual hypothesis of p=1.

This new approximate method overcomes the problem of the extremely large system state space, by lumping together states of the original Markov chain that are logically related, and by devising a technique for an approximate evaluation of the state transition probability of the resulting reduced Markov chain. The source of the inaccuracy of this model is thus represented by the derivation of an aggregated chain obtained by lumping together states which are not quite lumpable and by constructing a transition probability matrix which only partially accounts for the details of the model.

The state definition of the reduced Markov chain evaluated by this model derives from that of section 3.1, introducing the following simplifications:

a) the n vector is substituted by a variable, c, that is used to indicate the number

of $n_i \neq 0$; that is to say how many different memory modules have been demanded (some of them -or all- may be servicing previous requests). With this substitution, the exact demands distribution is lost.

b) the m-component vector representing the stage of service of the different memory modules is substituted by three variables that (partially) summarize the information carried by this vector in the following way:

b1) a variable, i, meaning how many memory modules have not yet completed the requested access. The exact informations related to which are the modules being accessed, and to which is the exact present access sub-cycle, are lost.

b2) a variable, e, to indicate the number of memory modules that have accomplished the requested service, but have not been able to communicate back to the processors, because the bus was not available to them.

b3) a variable, b, signals the bus status, accordingly to the following encoding:

. b=0, means the bus is busy servicing a request issued by a memory module.

. b=1, means the bus is busy servicing a request issued by a processor.

. b=2, means the bus is idle.

The state of the reduced Markov chain is thus represented by the vector:

(i,c,e,b)

To establish the state transition probability matrix of this model, four state classes can be distinguished:

Class 0, (i,c,0,2), with i=c
Class 1, (i,c,e,0), with 1+i+e=c
Class 2, (i,c,e,1), with 1+i+e=c
Class 3, (i,c,e,1), with 1+i+e < c

and the following relations must be accomplished:

$$i < MIN(n,m,r) \quad \text{and} \quad 1 < c < MIN(n,m)$$

The particular and important case with $r \geq MIN(n,m)$ yields to the value of the total number of states, S, as follows:

$$S = (3v^2+3v-2)/2, \qquad \text{where} \qquad v = MIN(n,m)$$

The most complicated part of the calculation of the transition probability matrix correspond to the class (i,c,e,0). From this class of states, the system evolves to subsequent states, following the schema listed below :

next states	transition probabilities
(i-1, c-1,e,0)	P1*P2*P3
(i-1,c,e+1,1)	P1*(P2*(1-P3)+(1-P2)*P4)
(i-1, c+1,e+1,1)	P1*(1-P2)*P4
(i,c-1,0,2), (i,c-1,e+1,0)	(1-P1)*P2*P3
(i,c-1,e-1,0)	(1-P1)*P2*P3
(i,c,e,1)	(1-P1)*(P2*(1-P3)+(1-P2)*(P4)
(i,c+1,e,1)	(1-P1)*(1-P2)*(1-P4)

where:

. P1 is the probability that one of the i busy memory modules completes its access during the processor cycle (at most, one memory access can complete during a single bus cycle). P1 is approximately equal to i/r.

. P2 is the probability that the most recently served request was the only one directed to a particular memory module; that is, b=0 and c decrements to c-1. To evaluate this probability, as an approximation, the following expression is used:

$$P2 = \frac{\binom{m}{c-1} \sum \frac{(n-1)!}{l_1!\ldots.l_{(c-1)}!}}{\binom{m}{c-1}\sum \frac{(n-1)!}{l_1!\ldots.l_{(c-1)}!} + \binom{m}{c}\sum \frac{(n-1)!}{l_1!\ldots.l_c!}}$$

where the summations extend to all possible values $l_1,\ldots l_k > 0$ such that $l_1+\ldots+l_k=n-1$

with k equal to c or to c-1 according to the corresponding term.

. P3 is the probability that the processor, which has just seen its last memory service completed, submits a new request directed to a memory module out of the c-1 modules which are presently being accessed. Thus, we have

$$P3 = (c-1)/m$$

. P4 is the probability that the processor which has just seen its last memory service completed, submits a new request directed to one memory module out of the c modules presently being accessed. We have

$$P4 = c/m$$

The study of the three remaining state classes leads to the following situations:

	next states	transition probabilities
(i,c,0,2)	(i-1,c,0,0)	P1
	(i,c,0,2)	1-P1
(i,c,e,1) 1+i+e=c	(i,c,e,0)	P1
	(i+1,c,e-1,0), (i+1,c,0,2)	1-P1
(i,c,e,1) 1+i+e<c	(i,c,e,0)	P1
	(i+1,c,e,1)	1-P1

5- NUMERICAL RESULTS

In this section a collection, and the corresponding discussion, of some numerical results, obtained from the previously presented models, is given. The models results have been verified, using several sets of parameters values, by comparing them to those obtained with simulations.

Table 1 shows the values of EBW calculated through an exact Markow chain, when the memory modules bus demands have priority over those from the processors. It is worthwhile noticing that the results are symmetrical on m and n. This remark suggests to make symmetric the approximate expression (n*=min(n,m) ,m*=max(n,m)) mentioned in section 3.2. In table 2, some results of the non-symmetrical expression are shown.

The observed numerical disagreements are always less than 9%, and in the interesting ranges of r> m > n , lower than 5-6%

The results obtained from the approximate Markov chain analysis developed in section 4 are shown in Table 3, that also includes simulation results. The numerical disagreements do not exceed 5% in almost any case.

6- BUFFERS IN THE MEMORY MODULES

In the previously described network architecture, any processor which issued an access request to some memory module, was not allowed to take the bus until the last service performed by that same module was sent back to the demanding processor. That operation scheme potentially increases the amount of possible memory interference.

Several mechanisms may be devised to augment the global system throughput. The one proposed here (see Fig. 4) consists of adding one buffer to every memory module input and output.

With this organization, every time a memory module completes one requested service, it puts the service results in the output buffer, and becomes available for servicing a new request, taking it (if it exists) from its input buffer. The efficiency of the system will thus be increased due to the fact that a memory module can now be busy servicing different requests in contiguous bus cycles.

In order to evaluate the system performance of this new system architecture, the following hypotheses are added to those introduced in Section 2:

1) the buffers acces time is considered either negligible or included in the memory cycle time.

2) a FIFO policy is assumed to serve the buffered requests.

3) the priority to get the bus, in case of conflict, is given to the processors requests (as in previous g' hypothesis).

If random exponential variables could be used to characterize the bus and memory modules service times, the buffered system could be modeled with a product form queueing network (18) and thus its performance evaluated using standard well established techniques (19), (20).

However, in this system both service times are constant values, and by using simulation techniques we have been able to measure the numerical differences between the two service times characterizations. The results obtained show large discrepancies, which exceeded 25% difference. Pessimistic results are obtained when an exponential distribution is assumed in the model.

Presently, only simulation results to evaluate the buffered system performance have been obtained. Exact or approximate analytical models are not constructed so far.

Several EBW values are shown in Fig. 5. For comparison purposes, the corresponding EBW values for a crossbar system with basic cycle equal to r+2 units of time, are also shown in the same Fig. 5. The results corresponding to the case p<1 are shown in Fig. 6.

It can be noted that the non-buffered crossbar EBW values can be improved by the buffered single-bus system EBW values, because the amount of memory interference is decreased when using memory buffers.

However, when r increases, the buffered single-bus EBW tends to the crossbar corresponding values, because the buffering effect losses its influence as the bus utilization decreases. In other words, when r approaches ∞ , as a constant processor cycle value is assumed, the memory modules service time grows up, and consequently the bus occupation time is lower, yielding a greater memory interference and lower network degradation.

This behaviour of the EBW, obtained for buffered single bus memory system, shows also that the

effect of buffering is proportionally larger as the difference (n-m) increases, due to the larger memory interference that such an organization implies.

7- CONCLUSIONS

In this paper, we have evaluated the multiplexed single-bus interconnection network, under discrete hypotheses. Two different bus arbitration strategies have been considered. Our analysis show that, to assign the bus according to a priority scheme that favours processor requests, leads to improvements in the EBW value. The maximum network bandwidth equals (r+2)/2; this value is attainable with r < MIN (n,m), r being the ratio between the cycle time of memory modules and the bus basic cycle. For larger values of r, the crossbar EBW acts as a lower bound value to the multiplexed single-bus EBW.

Assuming that priority is given to memory modules, an exact Markov chain model, and a combinational approximate model, have been presented. With priority to processors, an approximate markovian model has been constructed. All the models allow to obtain EBW values that are fairly close to the exact ones.

In a crossbar system, n*m connections are needed; in the multiplexed single-bus, only n+m are necessary, although other factors exist that reduce the cost effectiveness.

In the multiplexed single-bus network design, trade-offs can be made between several system parameters, to obtain performance measures comparable to those of a crossbar network. The 8x8 crossbar EBW value is attained with m=14 and r=8 in the single-bus system; only a 5% degradation is suffered if m=10, and four buses are needed with a multiple-bus network, as shown in (5).

The case p < 1 (internal processing cycles) has been evaluated through simulation techniques. With p > 0.4, a value of r=8 is enough to exceed the crossbar performance,in a system with 8 processors and 16 memory modules; with lower network loads, the crossbar EBW can be reached for lower values of m and r.

In order to reduce memory interference, buffering schemes are proposed here. Simulation results show that this technique allows not to increase the number (m) of memory modules needed to obtain a given EBW; a buffered single-bus system with r=18 performs like a 16x16 crossbar.

The multiplexed single-bus with memory buffers operates in saturation (no underutilization) until r approaches the value of MIN(n,m). EBW values better than those of a crossbar system are attainable with $r \sim MIN(n,m)+2$.

Because the amount of memory interference decreases when p < 1, the possitive influence of buffering becomes less effective as p decreases. However, if the value of p equals 0.3, r=12 is enough to get equal or better results than the crossbar in a 8x16 system.

ACKNOWLEDGMENT

The authors are indebted to Prof. G. Balbo for his helpful suggestions and discussions.

REFERENCES

1.- D.P. BHANDARKAR
"Markov chain models for analizing memory interference in multiprocessor computer systems". Proceedings 1st. Annual Sym. Comp. Architecture, Dec. 1973, pp 1-6.

2.- M.A. MARSAN et al.
"Comparative performance analysis of single bus multiprocessor architecture". IEEE Trans. on Computers, Vol. C-31, n 12, Dec. 1982, pp 1179-1191.

3.- E. SANVICENTE et al.
"Exact and aproximate models for multiprocessors with single bus and distributed memory". Int. Symposium MIMI, Paris, Jul. 1982, pp 15-18.

4.- M.A. MARSAN Y M. GERLA
"Markov models for multiple bus multiprocessor systems". IEEE Trans. on Computers, Vol. C-31, n 3, Mar. 1982, pp 239-248.

5.- M.VALERO, J.M.LLABERIA et al.
"A performance evaluation of the multiple bus networks for multiprocessor systems". ACM Sigmetrics Conf. on Measurement and Modeling of Comp. Systems, Aug. 1983.

6.- T.LANG, M.VALERO and I.ALEGRE.
"Bandwith of crossbar and multiple-bus connections for multiprocessors". IEEE Trans. on Computer, Vol. C-31, n. 12, Dec. 1982, pp. 1227-1234.

7.- J.H.PATEL.
"Processor-memory interconnections for multiprocessors". Proc. 6th Symp. on Comp. Architecture, 1979, pp. 168-177.

8.- L.J. BOLAND et al.
"The IBM system/360 Model 91: Storage system". IBM Journal, Jan. 1967, pp. 54-68.

9.- D. COMTE. et al.
"Les présentations dy système LAU en France et à l'étranger". Centre d'études et recherches de Toulouse.

10.-INTEL.
"iAPX 43203 VLSI interface processor".

11.-A. GUPTA and H.D. TOONG.
"Microprocessors-The first twelve years". Proceedings of the IEEE, Vol. 71, n. 11, Nov. 1983, pp. 1236-1256.

12.-D.M.DIAS and J.R. JUMP.
"Analysis and simulation on buffered delta networks". IEEE Trans. Computer, Vol. C-30, Apr. 1981, pp. 273-282.

13.- W.J. KAMINSKY AND E.S. DAVIDSON.
"Developing a multiple-instruction-stream single-chip processor". IEEE Computer, Vol. 12, Dec. 1979, pp 66-76

14.- H.F.JORDAN
"Performance measurements on HEP, a pipelined MIMD computer". Proc. 10th Symp. Comp. Arch. 1983, pp 207-212.
15.-M. P. ZOMBI AND A. C. SANDERSON.
"Rapid bus multiprocessor system". Computer Design, Nov. 1981, pp 189-200.
16.-D.A. PATTERSON AND C.H. SEQUIN.
"Design Considerations for single-chip computers of future". IEEE Trans. on Computers, Vol. C-29, no.2, Feb.1982, pp. 108-115.
17.- W.D.STRECKER.
"Analysis of the instruction execution rate in certain computer structures". Ph.D. dissertation., Carnegie - Mellon, 1970.
18.-F.BASKETT, K.M. CHANDY et al.
"Open closed and mixed networks of queues with different classes of customers". JACM 22, 2, pp. 248-260.
19.-J.P.BUZEN.
"Computational algorithms for closed queueing networks with exponential servers". CACM 16, 9, September 1973, pp. 527-531.
20.-M. REISER and S.S. HAVEMBERG.
"Mean value analysis of closed multichain queueing networks". JACM 27, 2, April 1980, 313-322.
21.-BASKETT,F. and SMITH, A.
"Interference in multiprocessor systems with interleaved memory". CACM, Vol. 19, n 6, June 1976, pp 327-334.

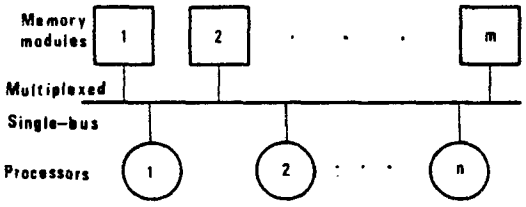

Fig. 1 Single-bus network topology

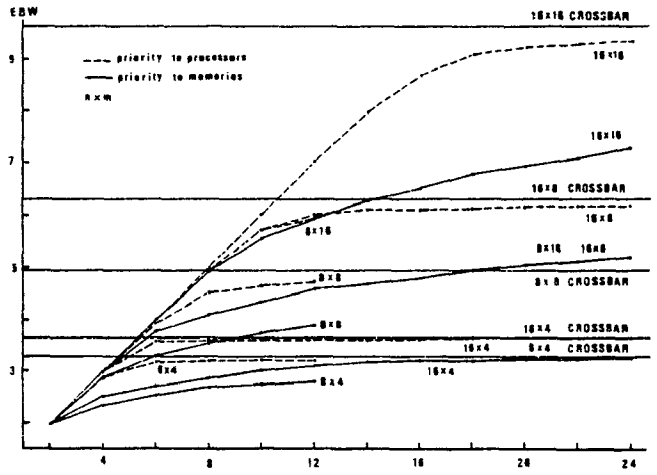

Fig. 2 Multiplexed Single-bus. Effective Bandwidth, (p=1).

m / n	2	4	6	8
2	1.417	1.625	1.694	1.729
4	1.625	2.308	2.603	2.761
6	1.694	2.603	3.164	3.469
8	1.729	2.761	3.469	3.988

Table 1. EBW exact values, with priority to memory modules. r=min(n,m)+7

m / n	2	4	6	8
2	1.417	1.625	1.694	1.729
4	1.729	2.392	2.653	2.792
6	1.807	2.778	3.305	3.570
8	1.827	2.987	3.692	4.178

Table 2. EBW approximate values, with priority to memory modules. r=min(n,m)+7

r / m	2	4	6	8	10	12
4	1.998	2.867	3.155	3.287	3.205	3.220
6	2.000	2.986	3.766	4.033	4.083	4.117
8	2.000	2.999	3.934	4.523	4.650	4.722
10	2.000	3.000	3.983	4.766	5.102	5.144
12	2.000	3.000	3.996	4.878	5.367	5.464
14	2.000	3.000	4.000	4.947	5.569	5.732
16	2.000	3.000	4.000	4.977	5.698	5.959

a) simulation model.

r / m	2	4	6	8	10	12
4	1.994	2.727	2.992	3.089	3.133	3.156
6	1.999	2.956	3.582	2.854	3.973	4.033
8	2.000	2.994	3.848	4.344	4.577	4.692
10	2.000	2.999	3.947	4.633	5.000	5.184
12	2.000	2.999	3.981	4.794	5.288	5.546
14	2.000	3.000	3.992	4.880	5.480	5.810
16	2.000	3.000	3.997	4.927	5.608	6.000

b) approximate model.

Table 3. EBW values with priority to processors

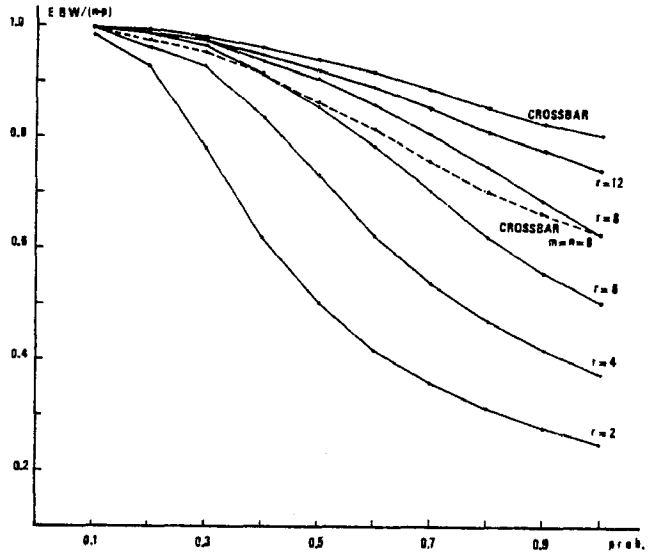

Fig. 3 Processors utilization. Multiplexed Single-bus, (p<1). n=8,m=16 systems.

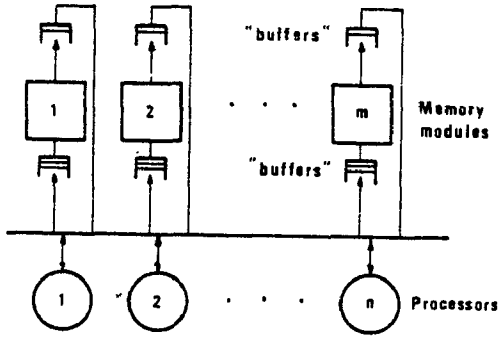

Fig. 4 Memory modules with buffers.

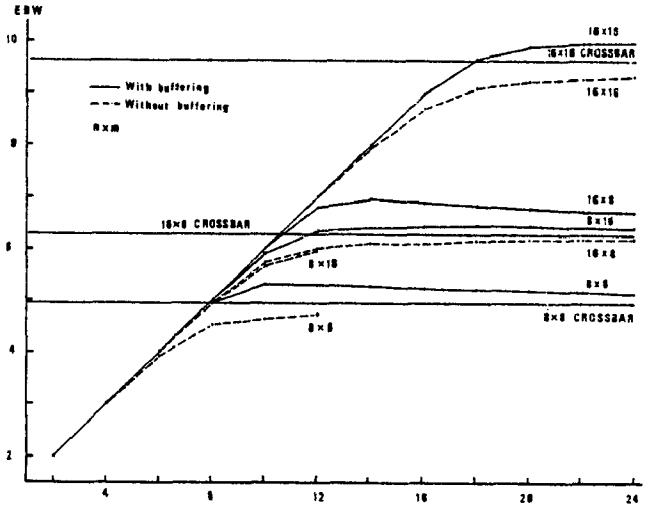

Fig. 5 EBW values to appreciate the effects of buffers in the memory modules.

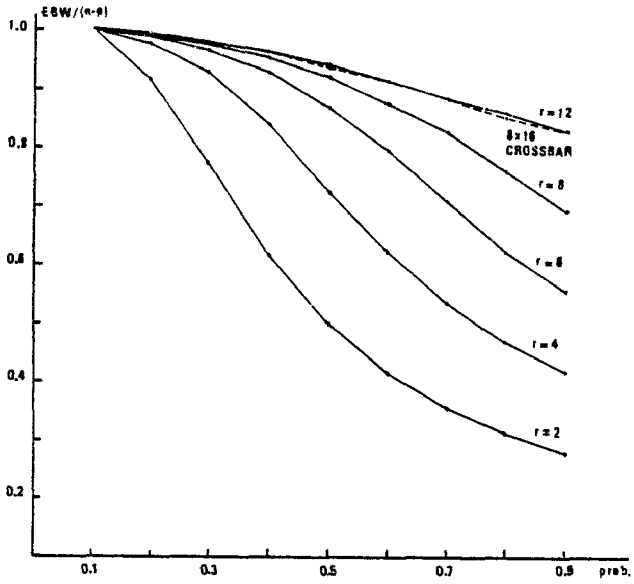

Fig. 6 Processors utilization values. n=8, m=16 systems.

r / m	6	8	10	12	14	16	18	20	22	24
4	3.915	3.938	3.815	3.731	3.661	3.617	3.575	3.541	3.523	3.499
6	3.997	4.747	4.795	4.734	4.674	4.630	4.588	4.560	4.529	4.506
8	4.000	4.943	5.312	5.312	5.275	5.239	5.206	5.180	5.155	5.136
10	4.000	4.984	5.608	5.724	5.725	5.709	5.685	5.666	5.647	5.633
12	4.000	4.994	5.778	5.987	6.020	6.019	6.010	5.997	5.983	5.970
14	4.000	4.998	5.867	6.178	6.237	6.246	6.245	6.232	6.223	6.217
16	4.000	4.999	5.912	6.325	6.405	6.426	6.429	6.421	6.414	6.410

Table 4. EBW values, with priority to processors, in a buffered system. n=8