

Effective Memory Bandwidth of Band-Connected Partial Crossbars

Bruno R. Preiss

Department of Electrical Engineering
University of Waterloo

Abstract—This correspondence introduces the class of partially-connected multiple-bus multiprocessor configurations called band-connected partial crossbars. An analytical derivation of the effective memory bandwidth of band-connected partial crossbars, based on a simple model of such systems, is also presented.

Index Terms—Band-connected partial crossbar, effective memory bandwidth, multibus, multiple-bus, multiprocessor.

I. Introduction

A multiple-bus multiprocessor system consists of P processors sharing M memory modules using $B \leq \min(P, M)$ buses[1]. The standard interconnection scheme for a multiple-bus system has all P processors and all M memory modules connected to all B buses[2] as shown in Fig. 1 (a). However, the standard interconnection scheme contains redundant connections. Eliminating (some or all) of the redundant connections gives rise to the class of partial multibus systems as shown in Fig. 1 (b)[3]. A special case of the partial multibus is the crossbar, as shown in Fig. 1 (c). All of these systems have the common property that they are fully connected. That is, there exists a (not necessarily unique) path between every processor and memory. In addition, systems in which $B = \min(P, M)$ exhibit maximum concurrency of memory access. That is, there are no bus conflicts, only memory module conflicts[1].

Bruno R. Preiss is with the Department of Electrical Engineering, University of Waterloo, Waterloo, Ontario, CANADA, N2L 3G1.

In this correspondence, we consider systems that are not fully connected. In particular, we consider systems called partial crossbars, as shown in Fig. 1 (d). The study of such systems is motivated by the observation that certain kinds of parallel computations, e.g., relaxation algorithms, exhibit strong memory module locality of reference. That is, each processor limits its shared-memory accesses to a subset of the memory modules. In addition, the high cost of implementing crossbars due to the large number of interconnections makes such systems impractical for large numbers of processors and memory modules. Furthermore, current technology limitations are likely to set a limit of four buses per processor (or memory) for the near future[4].

In[5], Sethi and Deo present an analysis of crossbar systems in which each processor has a favourite memory; i.e., one with a higher access probability. All other memory modules are assumed to be accessed with equal probability. In this correspondence, we consider systems in which each processor accesses a band of memory modules with equal probability. All other memory modules are inaccessible.

II. Band-Connected Partial Crossbars

A band-connected partial crossbar (BCPC) is a multiple-bus multiprocessor system consisting of p processors (P_0, P_1, \dots, P_{p-1}) and M memory modules (M_0, M_1, \dots, M_{M-1}) in which processor P_i is connected to memory modules $M_{id}, M_{id+1}, \dots, M_{id+m-1}$. Note that each processor is connected to $m \leq M$ memory modules. The total number of memory modules is $M=(p-1)d+m$. The quantity d is the memory module offset between consecutive processors. That is, consecutive processors share $m-d$ memory modules. An example of a BCPC with $p=2$ processors each accessing $m=5$ memory modules with a memory module offset of $d=2$ is shown in Fig. 2 (a).

A cyclic BCPC is a multiple-bus multiprocessor system consisting of p processors (P_0, P_1, \dots, P_{p-1}) and M memory modules (M_0, M_1, \dots, M_{M-1}) in which processor P_i is connected to memory modules $M_{id}, M_{id \oplus 1}, \dots, M_{id \oplus (m-1)}$, where \oplus denotes addition modulo M . The total number of memory modules is $M=pd$. An example of a cyclic BCPC with $p=3$ processors each accessing $m=3$ memory

modules with a memory module offset of $d=1$ is shown in Fig. 2 (b).

A processor in a BCPC can only access a subset of the memory modules in the system. However, BCPC systems exhibit maximum concurrency of memory access since there are no bus conflicts, only memory conflicts.

The advantages of BCPC systems are the reduced implementation cost, as measured by the total number of connections in the system and the bus loads, and the reduction in memory interference. These cost parameters for the multiple-bus multiprocessor system configurations shown in Fig. 1 are summarized in Table I.

III. Effective Memory Bandwidth

The memory access model used in this analysis is a very simple one: Each processor generates a memory request in every memory cycle. The memory requests are independent, random, and uniformly distributed. Blocked memory requests are discarded; they are not queued. Subsequent requests are independent.

Let p be the number of processors in a (non-cyclic) BCPC. Each processor is connected to exactly m memory modules. Let d be the memory module offset. The effective memory bandwidth of a (non-cyclic) BCPC, $EMBW(p, m, d)$, is defined as the expected number of busy memory modules in any given memory cycle.

In the case where $d=0$, $M=m$, the BCPC is, in fact, a fully-connected crossbar. It has been shown, [6] and [7], that the effective memory bandwidth for a crossbar is given by

$$EMBW(p, m, 0) = m \left[1 - \left(1 - \frac{1}{m} \right)^p \right]. \quad (1)$$

This is an important result as it forms the basis for the derivations to follow. Note also that

$$EMBW(p,m,m)=p \quad (2)$$

and

$$EMBW(1,m,d)=1 . \quad (3)$$

A. EMBW(2,m,d)

A BCPC with $p=2$ processors each accessing $m=5$ memory modules with a memory module offset $d=2$ is shown in Fig. 2 (a). A simplified graphical representation of this configuration is shown in Fig. 3. The BCPC can be divided into three regions of interest: Region A, wherein processor P_0 has sole access to d memory modules; region B, which consists of a $p \times (m-d)$ crossbar; and region C, wherein processor P_1 has sole access to the remaining d memory modules.

Using the regions defined above, the set of all possible permutations of processor memory accesses can be partitioned into the three memory-access patterns as shown in Table II: 1) Processor P_0 accesses a memory module in region A and P_1 in either B or C (indicated by the * in Table II). 2) Both P_0 and P_1 access modules in region B. 3) P_0 accesses in region B, P_1 in C. Also listed in Table II is the probability of each memory access pattern (prob.) and the expected number of busy memory modules given the specified memory access pattern (BW).

The effective memory bandwidth is simply the sum of the BW entries weighted by their respective probabilities. Combining (1) with the entries of Table II gives

$$EMBW(2,m,d)=2-\frac{\beta}{m}, \beta=1-\frac{d}{m}. \quad (4)$$

B. EMBW(p,m,m/2)

A BCPC with $p=4$ processors each accessing m memory modules with a memory module offset of $d=m/2$ is shown in Fig. 4. In this example, the set of memory access permutations is partitioned into five memory access patterns as shown in Table III. As in the preceding derivation, the effective memory

bandwidth is calculated by evaluating the appropriate weighted sum. Generalizing the calculation to an arbitrary number of processors, p , gives

$$\begin{aligned}
 EMBW(p,m,m/2) &= \frac{1}{2} + \sum_{k=1}^{p-2} k \left(\frac{1}{2}\right)^{k+2} + p \left(\frac{1}{2}\right)^p \\
 &+ EMBW(2,m/2,0) \sum_{k=1}^{p-1} \left(\frac{1}{2}\right)^{k+1} \\
 &+ \sum_{k=1}^{p-1} \left(\frac{1}{2}\right)^k EMBW(p-k,m,m/2).
 \end{aligned} \tag{5}$$

Using (1) and evaluating the sum of the geometric series, (5) can be simplified to the following recurrence relation

$$\begin{aligned}
 EMBW(p,m,m/2) &= 1 + \left(1 - \frac{1}{m}\right) \left(\frac{\frac{1}{2} - (\frac{1}{2})^p}{\frac{1}{2}}\right) \\
 &+ \sum_{k=1}^{p-1} \left(\frac{1}{2}\right)^k EMBW(p-k,m,m/2).
 \end{aligned} \tag{6}$$

It is shown in the appendix that the solution to (6) is

$$EMBW(p,m,m/2) = p - (p-1) \left(\frac{\frac{1}{2}}{m}\right) \tag{7}$$

Observe that (7) is a generalized form of (4).

C. $EMBW(p,m,d)$, $(m/2) \leq d \leq m$

The result of the preceding section can be easily extended to the case where $(m/2) \leq d \leq m$. This situation is shown in Fig. 5 for a BCPC with $p=4$ processors. The calculation of the effective memory bandwidth is summarized in Table IV. Generalizing the calculation to an arbitrary number of processors gives

$$\begin{aligned}
 EMBW(p,m,d) &= \alpha + \alpha\beta^{p-1}p + \sum_{k=1}^{p-2} k\beta^{k+2} \\
 &+ (1-2\beta) \sum_{k=1}^{p-2} (k+1)\beta^k \\
 &+ EMBW(2,m-d,0) \sum_{k=1}^{p-1} \beta^{k+1} \\
 &+ \alpha EMBW(p-1,m,d) \\
 &+ \sum_{k=2}^{p-1} \beta^k EMBW(p-k,m,d) \\
 &+ (1-2\beta) \sum_{k=2}^{p-1} EMBW(p-k,m,d).
 \end{aligned} \tag{8}$$

Using (1) and evaluating the sum of the geometric series, (8) can be simplified to the following recurrence relation

$$\begin{aligned}
 EMBW(p,m,d) &= 1 + \left(1 - \frac{1}{m}\right) \left(\frac{\beta - \beta^p}{\alpha}\right) \\
 &+ \sum_{k=1}^{p-1} (\beta^{k-1} - \beta^k) EMBW(p-k,m,d).
 \end{aligned} \tag{9}$$

It is shown in the appendix that the solution to this recurrence relation is

$$EMBW(p,m,d) = p - (p-1) \left(\frac{\beta}{m}\right), \quad \beta = 1 - \frac{d}{m}. \tag{10}$$

IV. Spatial Steady State

The preceding derivations of effective memory bandwidth are based on an exhaustive enumeration of all possible memory access patterns. The problem with this approach is that the complexity of the computation increases as the degree of memory module overlap increases (i.e., as d decreases).

An alternate approach to calculating the effective memory bandwidth of a (non-cyclic) BCPC is to calculate for each processor the probability that its memory access will succeed. The effective memory bandwidth is then simply the sum of these probabilities. The advantage of this approach is that for large numbers of processors there is a spatial steady state. Processors in the spatial steady state have identical access success probabilities. Furthermore, the access success probability for a processor in spatial steady state is independent of the total number of processors in the system.

The spatial steady state results are directly applicable to the analysis of cyclic BCPC systems. Due to symmetry, all processors in a cyclic BCPC have identical access success probabilities. Consequently, all processors in a cyclic BCPC are in spatial steady state.

A. EMBW(p,m,m/2), p≥2

A BCPC with an arbitrary number of processors and a memory module offset of $d=m/2$ is shown in Fig. 6. Note that all but the end processors are in the spatial steady state. The probability, $SSS(m,d)$, that a steady-state processor succeeds in its memory access is simply

$$SSS(m,m/2) = \frac{1}{2}(1) + \frac{1}{2}(EMBW(2,m/2,0)/2) \tag{11}$$

$$= 1 - \frac{1}{2m} .$$

The probability that each of the end processors succeeds is given by

$$E_0(m,m/2) = \frac{1}{2}(1) + \frac{1}{2}SSS(m,m/2) \tag{12}$$

$$= 1 - \frac{1}{4m} .$$

The effective memory bandwidth is simply

$$\begin{aligned}
 EMBW(p, m, m/2) &= 2E_0(m, m/2) + (p-2)SSS(m, m/2) \\
 &= p - (p-1) \binom{\frac{1}{2}}{m}.
 \end{aligned} \tag{13}$$

Note that this result is the same as (7). This derivation is also much simpler because it does not require solving a recurrence relation.

The spatial steady state can be interpreted in two ways: First, note that

$$\lim_{p \rightarrow \infty} \frac{EMBW(p, m, m/2)}{p} = SSS(m, m/2). \tag{14}$$

Thus, in general, the spatial steady state processor access success probability is the limiting value of the average access success probability for a processor as the number of processors is increased. Second, the spatial steady state is the exact access success probability for cyclic BCPC systems.

We introduce a new quantity, $\overline{EMBW}(p, m, d)$, which is the effective memory bandwidth for a cyclic BCPC given by

$$\overline{EMBW}(p, m, d) = p SSS(m, d). \tag{15}$$

B. $\overline{EMBW}(p, m, m/k)$

The spatial steady state of a BCPC with a memory module offset of $d=m/3$ is shown in Fig. 7. The calculation of the spatial steady state access probability is summarized in Table V. Generalizing this computation to an arbitrary fractional memory module offset, $d=m/k$, gives

$$SSS(m, m/k) = \sum_{i=0}^{k-1} \binom{k-1}{i} \frac{(k-1)^{(k-1)-i}}{k^{k-1}} \frac{EMBW(i+1, m/k, 0)}{i+1}. \tag{16}$$

Using (1) and the binomial theorem, the sum in (16) can be evaluated to give

$$\overline{EMBW}(p,m,m/k)=\frac{pm}{k} \left[1 - \left(1 - \frac{1}{m} \right)^k \right]. \quad (17)$$

C. $\overline{EMBW}(p,m,d)$

The spatial steady state of a BCPC with each processor accessing $m=5n$ memory modules and a memory module offset of $d=2n$, where n is an arbitrary integer, is shown in Fig. 8. Note that in the general case, the number of processors competing for the memory modules in a particular region is not a constant (even in spatial steady state).

In a general BCPC, each processor accesses $m=kn$ memory modules with a memory module offset of $d=ln$, where $n=\text{gcd}(m,d)$. The number of processors competing with a given processor follows the pattern described in Table VI. For example, in Fig. 8. the pattern is 3 2 3 2 3. Note that the number of times that an entry appears in the pattern is exactly equal to its value. In this example, the value 3 appears three times in the pattern, the value 2 appears twice. Using this observation, it is possible to generalize (16) to give

$$\begin{aligned} SSS(kn,ln) &= \sum_{j=0}^{l-1} \frac{q}{k} \sum_{i=0}^{q-1} \binom{q-1}{i} \frac{(k-1)^{(q-1)-i}}{k^{q-1}} \frac{EMBW(i+1,n,0)}{i+1}, \quad q = \left\lfloor \frac{k-j}{l} \right\rfloor \\ &= \sum_{j=0}^{l-1} n \left[1 - \left(1 - \frac{1}{kn} \right)^q \right], \quad q = \left\lfloor \frac{k-j}{l} \right\rfloor \\ &= ln \left[1 - \left(1 - \frac{1}{kn} \right)^{\lfloor k/l \rfloor} \left(1 - \frac{k \bmod l}{kln} \right) \right]. \end{aligned} \quad (18)$$

Thus, \overline{EMBW} is given by

$$\overline{EMBW}(p,m,d)=pd \left[1 - \left(1 - \frac{1}{m} \right)^{\lfloor m/d \rfloor} \left(1 - \frac{m \bmod d}{md} \right) \right]. \quad (19)$$

The spatial steady state access success probability is shown in Fig. 9 as a function of d/m , the memory module offset ratio, for various values of m . Note that the memory module offset ratio, d/m , required to maintain a given access success probability decreases as m is increased. However, the absolute memory module offset required increases as m increases. Thus, as the number of memory modules accessed by a processors is increased, the number of processors competing for a given memory module must be decreased in order to maintain a given effective memory bandwidth.

V. Conclusions

In this correspondence, we have derived the effective memory bandwidth for the class of partially-connected multiple-bus multiprocessor systems called band-connected partial crossbar systems. We have shown that the effective memory bandwidth of these systems can be derived from the results for a fully-connected crossbar system. In addition, we have shown an alternate derivation of the effective memory bandwidth based on spatial steady state conditions. This approach is much simpler than an exhaustive enumeration of access patterns. Furthermore, the spatial steady state solution directly gives the effective memory bandwidth of cyclic BCPC systems.

Appendix

In this appendix it is shown that the solution of (9) is (10). In addition, since (7) and (8) are special cases of (9) and (10) respectively, this proof also shows that the solution of (7) is (8).

Proof: (by course-of-values induction)

1) *Base case.* $p=1$. Using (3),

$$EMBW(1,m,d)=1 \tag{20}$$

2) *Induction.* Assume (10) is true for $p=1,2,\dots,p'$. By (7),

$$\begin{aligned}
 EMBW(p'+1,m,d) &= 1 + \left(1 - \frac{1}{m}\right) \left(\frac{\beta - \beta^{p'+1}}{\alpha}\right) \\
 &+ \sum_{k=1}^{p'} (\beta^{k-1} - \beta^k) EMBW(p'+1-k,m,d) \\
 &= 1 + \left(1 - \frac{1}{m}\right) \left(\frac{\beta - \beta^{p'+1}}{\alpha}\right) \\
 &+ \sum_{k=1}^{p'} (\beta^{k-1} - \beta^k) \left(k - (k-1) \frac{\beta}{m}\right).
 \end{aligned} \tag{21}$$

This can be reduced to

$$EMBW(p'+1,m,d) = (p'+1) - p' \left(\frac{\beta}{m}\right). \tag{22}$$

References

1. Y.-C. Liu and C.-J. Jou, “Effective Memory Bandwidth and Processor Blocking Probability in Multiple-Bus Systems,” *IEEE Trans. Comput.*, Vol. C-36, No. 6, June 1987. pp. 761-764,
2. T. Lang, M. Valero, and M. A. Fiol, “Reduction of Connections for Multibus Organization,” *IEEE Trans. Comput.*, Vol. C-32, No. 8, Aug. 1983. pp. 707-716,
3. T. Lang, M. Valero, and I. Alegre, “Bandwidth of Crossbar and Multiple-Bus Connections for Multiprocessors,” *IEEE Trans. Comput.*, Vol. C-31, No. 12, Dec. 1982. pp. 1227-1982,
4. T. N. Mudge, J. P. Hayes, and D. C. Winsor, “Multiple Bus Architectures,” *Computer*, Vol. 20, No. 6, June 1987. pp. 42-48,
5. A. S. Sethi and N. Deo, “Interference in Multiprocessor Systems with Localized Memory Access Probabilities,” *IEEE Trans. Comput.*, Vol. C-28, No. 2, Feb. 1979. pp. 157-163,
6. D. Y. Chang, D. J. Kuck, and D. H. Lawrie, “On the Effective Bandwidth of Parallel Memories,” *IEEE Trans. Comput.*, Vol. C-26, No. 5, May 1977. pp. 480-490,
7. J. H. Patel, “Performance of Processor-Memory Interconnections for Multiprocessors,” *IEEE Trans. Comput.*, Vol. C-30, No. 10, Oct. 1981. pp. 771-780,

TABLE I
Cost Parameters for
Multiple-Bus Multiprocessor Systems

configuration	number of buses	bus load	number of connections
multibus	$B \leq \min(P, M)$	$P+M$	$(P+M)B$
partial multibus	$B \leq \min(P, M)$	$P+Mb/B$	$PB+Mb$
crossbar	$\min(P, M)$	$\max(P, M)+1$	$PM+\min(P, M)$
BCPC	P	$m+1$	$P(m+1)$

TABLE II
Calculating $EMBW(2, m, d)$.
($\alpha=d/m, \beta=1-\alpha$)

P_0	P_1	prob.	BW
A	*	α	2
B	B	β^2	$EMBW(2, m-d, 0)$
B	C	$\alpha\beta$	2

TABLE III
Calculating $EMBW(4, m, m/2)$.

P_0	P_1	P_2	P_3	prob.	BW
A	*	*	*	1/2	1 + $EMBW(3, m, m/2)$
B	B	*	*	1/4	$EMBW(2, m/2, 0)$ + $EMBW(2, m, m/2)$
B	C	C	*	1/8	1 + $EMBW(2, m/2, 0)$ + $EMBW(1, m, m/2)$
B	C	D	D	1/16	2 + $EMBW(2, m/2, 0)$
B	C	D	E	1/16	4

TABLE IV
Calculating $EMBW(4, m, d)$.
($(m/2) \leq d \leq m$)

P_0	P_1	P_2	P_3	prob.	BW
A	*	*	*	α	1 + $EMBW(3, m, d)$
B	B	*	*	β^2	$EMBW(2, m-d, 0)$ + $EMBW(2, m, d)$
B	C	*	*	$\beta(1-2\beta)$	2 + $EMBW(2, m, d)$
B	D	D	*	β^3	1 + $EMBW(2, m-d, 0)$ + $EMBW(1, m, d)$
B	D	E	*	$\beta^2(1-2\beta)$	3 + $EMBW(1, m, d)$
B	D	F	F	β^4	2 + $EMBW(2, m-d, 0)$
B	D	F	G	$\beta^3(1-2\beta)$	4
B	D	F	H	β^4	4

TABLE V
Calculating $SSS(m, m/3)$.

P_{n-1}	P_n	P_{n+1}	prob.	access success prob.
\bar{A}	A	\bar{A}	(2/3)(2/3)	1
\bar{A}	A	\bar{A}	(2/3)(1/3)	$EMBW(2, m/3, 0)/2$
\bar{A}	A	A	(1/3)(2/3)	$EMBW(2, m/3, 0)/2$
A	A	A	(1/3)(1/3)	$EMBW(3, m/3, 0)/3$

TABLE VI
Memory Access Overlap Patterns.

	0	1	2	...	$k-1$
$l=1$	$\left[\begin{array}{c} k \\ 1 \end{array} \right]$	repeat	...		
$l=2$	$\left[\begin{array}{c} k \\ 2 \end{array} \right]$	$\left[\begin{array}{c} k-1 \\ 2 \end{array} \right]$	repeat	...	
$l=3$	$\left[\begin{array}{c} k \\ 3 \end{array} \right]$	$\left[\begin{array}{c} k-1 \\ 3 \end{array} \right]$	$\left[\begin{array}{c} k-2 \\ 3 \end{array} \right]$	repeat	...
...					
$l=k$	$\left[\begin{array}{c} k \\ k \end{array} \right]$	$\left[\begin{array}{c} k-1 \\ k \end{array} \right]$	$\left[\begin{array}{c} k-2 \\ k \end{array} \right]$...	$\left[\begin{array}{c} 1 \\ k \end{array} \right]$

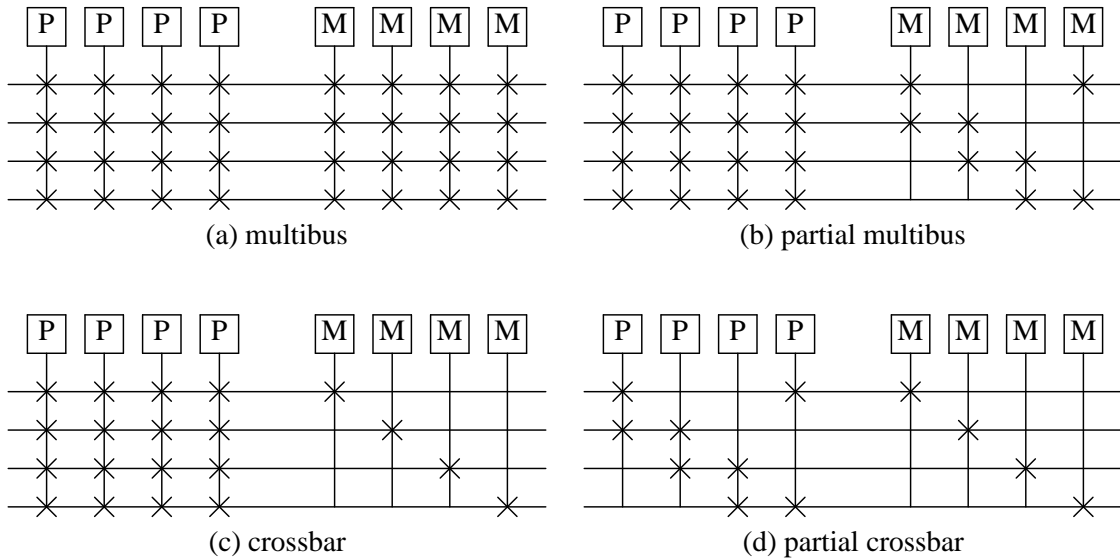


Fig. 1. Multiple-bus multiprocessor configurations.

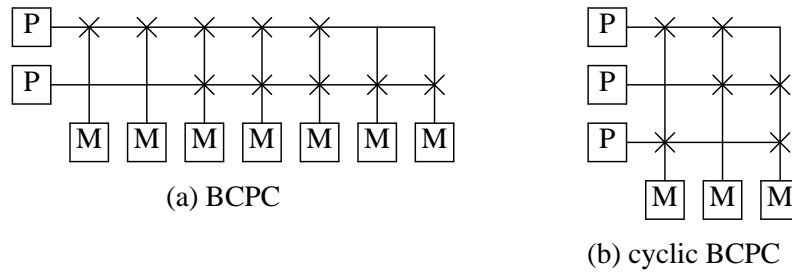


Fig. 2. (a) BCPC with $p=2$ processors each accessing $m=5$ memory modules with a memory module offset of $d=2$. (b) Cyclic BCPC with $p=3$ processors each accessing $m=5$ memory modules with a memory module offset of $d=1$.

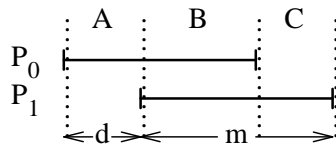


Fig. 3. Simplified graphical representation of a BCPC with $p=2$ processors.

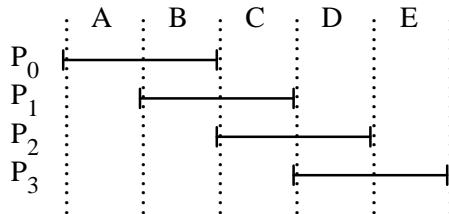


Fig. 4. BCPC with $p=4$ processors and memory module offset of $d=m/2$.

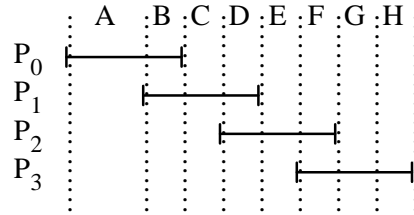


Fig. 5. BCPC with $p=4$ processors and memory module offset d , $m/2 \leq d \leq m$.

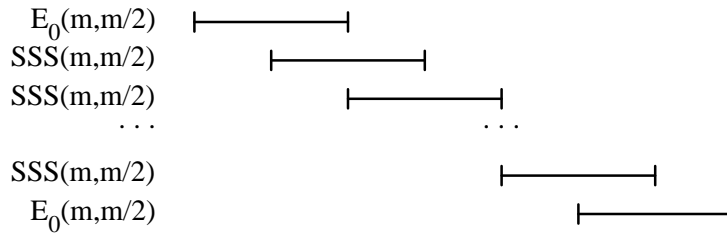


Fig. 6. BCPC with memory module offset of $d=m/2$ illustrating spatial steady state (SSS) and end conditions (E_0).

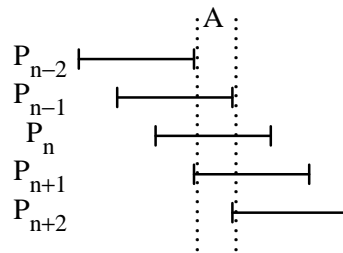


Fig. 7. Spatial steady state of a BCPC with memory module offset $d=m/3$.

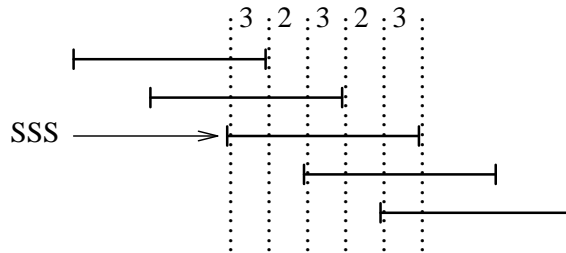


Fig. 8. Spatial steady state of a BCPC with each processor accessing $m=kn$, $k=5$, memory modules and memory module offset of $d=ln$, $l=2$.

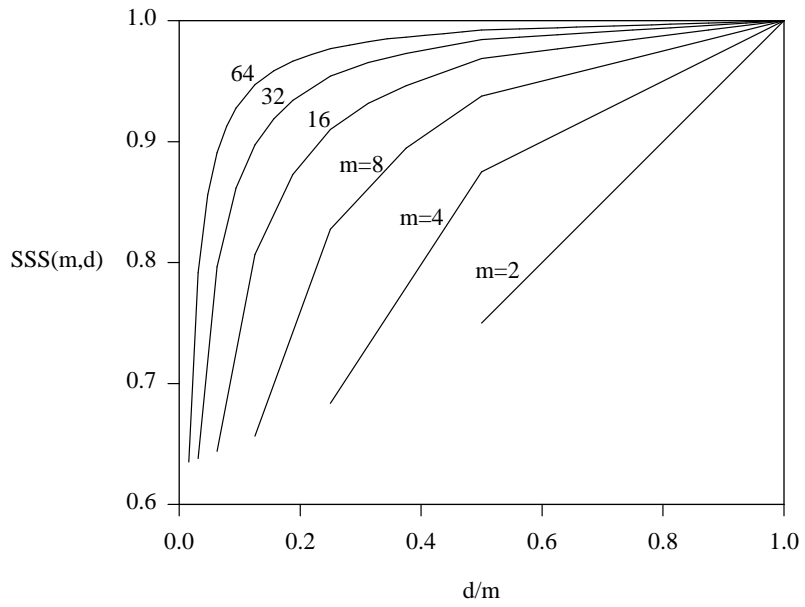


Fig. 9. Graph of spatial steady state access success probability, $SSS(m,d)$, vs. memory module offset ratio, d/m , for various values of m .