TR-656

Fundamental Characteristics of the Snooping Cache in a Parallel Inference Machine

by
T. Tarui, T. Nakagawa, N. Ido &
M. Sugie (Hitachi)

June, 1991

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

Fundamental Characteristics of the Snooping Cache in a Parallel Inference Machine

Toshiaki Tarui, Takayuki Nakagawa, Noriyasu Ido and Mamoru Sugie
Central Research Laboratory, Hitachi, Ltd.
Higashi Koigakubo, Kokubunji, Tokyo 185, Japan

Abstract

The design concepts of the snooping cache for a Parallel Inference Machine (the PIM/c prototype) are discussed and the cache's performance is investigated in detail.

The snooping cache control mechanism was implemented on a single VLSI chip, and embedded in the PIM/c prototype to evaluate its actual performance.

As the focus of this evaluation using a real machine is to clarify the fundamental characteristics of the snooping cache, we carried out the experiments by controlling the various parameters of the cache access patterns, instead of running specific benchmark programs.

The results of the performance evaluation proved advantages of the snooping cache protocol such as an invalidation scheme and inter-cache data transfer mechanism.

Finally, the design features of the cache architecture are examined to find directions for further improvement.

1. Introduction

Japan's Fifth Generation Computer Project[1] has been driven by ICOT (the Institute for new generation COmputer Technology). The main goal of the project is to develop a knowledge and information processing system. In the system, all programs including an operating system are written in the logic

programming language KL1,[2] which has a stream-AND parallel feature and inter-process communication/synchronization mechanisms. A parallel machine customized to KL1, the parallel inference machine (PIM), is one of the most important research themes of the project. PIM is designed to exploit the fine grain parallelism of KL1 for the purpose of efficient parallel execution of KL1 programs, and is now under development.[3] Five PIM models have been developed by the participating companies in collaboration with ICOT. PIM model c (PIM/c)[4] is the machine developed by Hitachi consisting of 256 PEs (processing elements).

PIM/c is organized along a hierarchical structure, as if composed of loosely coupled clusters of TCMP (Tightly Coupled Multi-Processors). This hierarchical structure enables us to reduce the parallel processing overhead, especially communication overhead, by utilizing the locality of logic programs. In large-scale parallel machines, high speed communication among all PEs is limited by the interconnection distance. However, PEs can communicate at high speed inside a cluster composed of a limited number of PEs. In implementing the PE clusters, the cache management is crucial.

A snooping cache[5] is employed in the PIM/c clusters to support efficient communications. During the execution of KL1 programs, logical variables are shared among PEs. Therefore, fine grain communications between processors occur frequently, and they cause a cache coherence problem. From the viewpoint of parallel processing overhead reduction, as many PEs as possible should be installed in a cluster because communication overhead is much higher among clusters than in the cluster. A snooping cache solves the cache coherence problem with low overhead and, at the same time, enables the installation of many PEs in the cluster because of its inter-PE traffic reduction effect.

Many snooping cache schemes have been proposed.[5] However, none of them is appropriate for KL1 because of its frequent fine grain communications. The "five-state snooping cache protocol"[6] was proposed to efficiently support fine grain communications. We have completed developing a small scale

prototype of PIM/c in which the snooping cache mechanism is implemented by its integration into one VLSI chip.

In this paper, the basic design concepts of the snooping cache implementation are described first, and the cache's predicted performance is evaluated on the prototype. Furthermore, its features are inspected in detail.

During the first stage of the PIM development, various snooping cache schemes were examined by function level simulation disclosing their macroscopic behavior.[6] After these investigations, a five-state snooping cache protocol was developed. Many parameters determine the snooping cache behavior. Also, the detailed implementation structures may influence the behavior. So far, detailed simulated behavior of the PIM/c snooping cache with the five-state protocol, taking detailed structures such as hardware organization into account, has not been reported in the literature because a very complicated parallel simulator is necessary for such simulation. Having completed the development of real hardware in a single VLSI chip, we are able to verify the performance of the designed protocol and, at the same time, to study detailed characteristics of the designed cache in order to discover directions for further improvement.

Since the investigation of the fundamental characteristics of the designed snooping cache is of primary importance, benchmark programs of specific applications are not used. This is motivated by the fact that such benchmark programs can give us only limited data on various kinds of features and may lead to incorrect decisions. Even if a certain cache scheme is effective in some applications, it might impose significant problems on others. In this work, several fundamental cache access parameters such as locality and share ratio are selected, and the influence of those parameters on cache performance is examined by varying their values in cache access patterns.

2. The PIM/c prototype architecture

Figure 2.1 shows the organization of the PIM/c prototype. In PIM/c, the PEs are connected hierarchically. A cluster is an 8-PE TCMP and 32 clusters are loosely connected by a network. Currently, the development of a 16-PE model (2 cluster * 8 PE) has been completed, and a 256-PE model is under development.

Inside the cluster, 8 PEs, 1 CC (cluster controller) and a main memory are connected to a common-bus. The CC is a processor for network communication. PEs and the CC use a common snooping cache mechanism. A new scheme has been developed for this snooping cache.

Each PE is composed of 3 units.

(1) EU (Execution Unit) - The processor of PIM/c which executes KL1 instructions with microprograms. An EU microinstruction has a 104-bit length composed of several fields.

(2) BU (Buffer Unit) - Implements the coherent cache mechanism based on the "five-state snooping cache protocol" designed for PIM/c. It also supports shared address lock and inner-cluster communication with request broadcasting. Each PE has 2 BUs fitted to the two-way-interleaved common-bus.

(3) SU (Service processor interface Unit) - A service processor interface supporting maintenance and debugging features and clock distribution.

The cluster controller (CC) also includes the following unit besides the PE units.

NU (Network interface Unit) - An interface with the network through which inter-cluster communication is performed.

3. The PIM/c snooping cache protocol

3.1. Memory access characteristics in PIM

In KL1 programs, inter-process communication through logical variables occurs frequently. Figure 3.1 shows the typical memory access sequence in KL1 programs including memory cell allocations, a suspension/resumption

mechanism for synchronization, and incremental garbage collection. The following are the basic characteristics of PIM cache access.

(1) fine grain communication

The basic data size in logic programming is 1 word (logical variable) or 2 words (list cell).

(2) high write ratio

The KL1 system has operations for concurrent processing such as synchronization and incremental garbage collection.[7] Since memory cells are shared by all PEs in the cluster, incremental garbage collection includes synchronization. These operations are performed in each access to a logical variable, and consequently, write access to synchronization variables occurs frequently.

(3) high locality

During unification operation, access to the same address occurs frequently.

(4) frequent inter-cache communications

Most logical variables are referred by producers and consumers.

A PIM snooping cache mechanism should match the PIM cache access characteristics.

3.2. Basic features of the PIM snooping cache

Issues to be taken into consideration in the design of the PIM snooping cache are:

(1) Traffic reduction in common-bus operations caused by cache miss-hits.

(2) Overhead reduction in cache coherence maintenance consisting of common-bus operations such as data transfer and invalidation, and bus snooping.

The following is a summary of PIM snooping cache features.

(1) write-back

With high write ratios, a write-back policy is used to reduce common-bus/memory traffic.

(2) invalidation scheme

Invalidation is used to solve the cache coherence problem in the PIM based on high access locality and high write ratios in KL1 programs.

The alternative to invalidation is the broadcast scheme in which a new value is broadcast to update the contents of other caches. In a broadcast scheme, however, data on a cache is never invalidated and remains in the cache for a long time, even when it is not really used by the PE. Because of this, a high data share ratio sometimes causes needless updates in broadcast schemes. This broadcast scheme overhead might be a major portion of the overall PIM overhead.

In the invalidation scheme, data in other caches is invalidated in the first write access, and all successive write access can be accomplished locally, without common-bus transactions. However, the invalidation scheme has a drawback in the so-called "ping-pong effect," when two different PEs write and read the same cache block alternately. In this case, a pair of invalidation and fetch to the same address, replaced by one broadcasting command, occurs frequently. In PIM, however, data access with high locality prevents this problem.

(3) inter-cache data transfer

An inter-cache transfer mechanism is employed to enable fast data fetch because access to cache is faster than to main memory. Main memory is accessed only when no cache has the data.

3.3. A summary of the designed protocol

A snooping cache protocol can be specified by the cache block states and the actions to be performed in each of those states.

Figure 3.2 shows the states of the PIM/c snooping cache. Five states are defined in each cache block considering the following:

(1) exclusive/shared

An exclusive state indicates that no other cache has this same block. Wasteful invalidation can be eliminated because an invalidation command is sent only when another cache has a copy.

(2) clean/modified

A modified state indicates that data in the cache is inconsistent with memory. This state is required in order to introduce the write-back policy.

(3) valid/invalid

An invalid state indicates that data in the cache is stale. This state is required to introduce the invalidation scheme for cache coherence maintenance.

Table 3.1 lists EU commands and the actions performed in each state. A snooping cache controller changes the state of the cache block and generates common-bus commands if necessary. Common-bus commands are composed of three basic ones:

F	Fetch	read the block
I	Invalidation	invalidate the block in other caches
SO	Swap-Out	write the block back to main memory

Table 3.2 shows common-bus commands and the corresponding cache operations. Each cache snoops all the commands on the common-bus and performs the necessary actions to maintain cache coherence.

4. The PIM/c snooping cache architecture

4.1. Design issues

The following aspects should be taken into account in order to efficiently implement the snooping cache protocol in PIM/c.

(1) EU access time in hit cases

Our trial coding of PIM/c microprograms suggested that memory access consumes almost all cycles in EU operations. This is because the horizontal microarchitecture of the EU enables other operations such as arithmetic to be executed in parallel with memory access. Thus, the reduction of memory access time is the most important design issue, especially in cache hit cases. EU itself can access memory in 2 cycles. Hence, in hit cases, the cache access time should be equal to 2 cycles.

(2) hardware resource limit

From the viewpoint of reducing machine cycle time, one PE (CC) should be integrated in one board. This is because signal delay is much greater among boards than in a board. To integrate one PE into one board, the snooping cache controller should be integrated into one VLSI chip. In the case of the 0.8-micrometer CMOS gate arrays we are using, nearly 35 kilogates can be integrated in a chip.

(3) bus command overhead

The common-bus should have low command latency and high command/data throughput in order to perform efficient communication among PEs.

(4) cache access conflict

In the snooping cache, commands arrive from the EU and the common-bus simultaneously. Consequently, the access conflicts between the EU and the bus commands must be solved to maintain cache coherence.

The following sections present the snooping cache implementation in PIM/c in detail.

4.2. Implementation

Figure 4.1 shows a block diagram of the PIM/c snooping cache. One cache bank is composed of a snooping cache control IC, which includes CAA (Cache Address Array) RAMs, and external CDA (Cache Data Array) RAMs. The control chip also contains the cache control circuits and the data buffers/selectors. This organization helps meet two requirements of the PIM/c snooping cache. First, CAA, in which addresses and states are stored, requires fast access, and second, CDA, in which data is stored, requires a relatively large capacity.

As discussed in section 4.1, the access from EU in cache hit cases must be finished in two cycles. To achieve this target, a look-ahead memory access mechanism is employed. In other words, the cache controller starts the CAA/CDA access before it has finished decoding commands issued by the EU. This is required because it takes two cycles to access the data stored in CDA. CDA access within two cycles would be impossible if the cache controller started

accessing CAA/CDA after command decoding. Another mechanism to speed up the EU access is parallel RAM access. In this scheme, in read access, CAA and CDA data in two rows are read in-parallel and data is selected after CDA access.

4.3. Common-bus features

In PIM/c a two-way-interleaved bus is used to increase the bus throughput. Results of function level simulation suggest that one common-bus can support only 4 PEs because of bus throughput limitations. From the viewpoint of software efficiency, however, approximately 8 PEs are required for each cluster. Therefore, the bus, the shared memory, and the cache are all two-way-interleaved.

A round-robin scheme is used for common-bus arbitration. This mechanism allows all PEs to receive equal priority - in other words, to have equal bus wait time. Moreover, a round-robin type of bus arbiter is used to select one of the PEs responding simultaneously to a request for inter-cache data transfer.

Table 4.1 shows the common-bus signals and Figure 4.2 shows a time chart of the typical common-bus commands.

4.4. Cache access conflict

As described in section 4.1, memory access conflicts of EU commands and bus snooping operations occur on CAA/CDA RAMs. A two-port-memory, in which EU can access the memory in parallel with the bus snooping hardware, could solve this problem. However, this would be impossible because of hardware limitations on the CAA RAM capacity. CAA access from either the EU or from bus snooping hardware must be delayed in order to avoid memory access contention.

In the snooping cache protocol used on PIM/c, bus snooping cannot be delayed because bus operation must maintain the fixed sequence as shown in Figure 4.2. An alternative is a protocol with flexible bus operation, but this would increase the bus command length and overhead related to bus operation. Furthermore, it is preferable not to increase the bus busy period by

delaying bus snooping operations. Such delays waste the important common bus resource shared by PEs.

For that reason, the EU access is delayed when bus snooping hardware and EU access the same bank of CAA or CDA RAMs at the same time. This imposes a certain overhead on access from EU, but it is not so critical as the EU can access a different memory bank distinct from the one accessed by the bus snooping hardware.

Table 4.2 summarizes the architecture of PIM/c snooping cache.

5. Investigation of the PIM/c snooping cache characteristics

5.1. Purpose of investigation

In order to investigate the fundamental characteristics of the PIM/c snooping cache, the hardware performance was measured and analyzed using a real machine. The characteristics of the designed snooping cache mechanism were studied to discover the direction to further improvement.

Conventionally, the performance of computer systems is evaluated by benchmark programs. However, with benchmark programs, only limited data can be measured, and global features covering a wide range of applications cannot be wholly clarified. The snooping cache can be promising in this wide range of applications, but its behavior is not yet sufficiently clear. At this time, we feel that global features are more important than performance in specific applications. PIM is designed for knowledge and information processing. Finally, its performance should be evaluated by benchmarks in these applications. In the next development stage, PIM/c cache performance should be evaluated by programs written in KL1.

Since in this paper we focus on fundamental characteristics which are general over a wide range of applications, artificial access patterns, in which cache access parameters can span a reasonable range, were employed.

5.2. Access parameters

Main parameters important to snooping cache behavior are locality, share ratio and write ratio. Consequently, the effects of those parameters on snooping cache performance were evaluated. The effect of working set size was not evaluated in this investigation because only coherence maintenance characteristics were of interest here.

(1) locality

The locality is determined by the frequency of access to a cache block. In other words, locality is high if much access to the same block occurs in a short period. In general, memory access locality enables the cache to work efficiently. If addresses are accessed completely at random, there is no necessity to install a cache in each PE. As a result, locality should be considered in the investigation of cache protocol. However, unlike other parameters, there is no clear definition of locality, and it cannot be controlled directly. In this investigation, the access intervals to the same block define the locality.

(2) share ratio

The share ratio is defined as the number of shared blocks in an access pattern. The measured share ratio on the real cache may be different from this share ratio in the access pattern because, with the invalidation scheme, copies in other caches are erased on write access. The number of PEs accessing the shared cache block is determined randomly. The effect of the share ratio is evaluated to investigate the characteristics of the inter-cache data transfer.

(3) write ratio

The write ratio determines the frequency of coherence maintenance operations. Several characteristics of the coherence maintenance by the snooping cache protocol can be investigated with this parameter.

Table 5.1 summarizes the cache access parameters and their default values.

5.3. The methodology of the experiments

In the snooping cache investigation, data measurements are made in the following sequence.

(1) Generate cache access pattern, on workstation, focusing on specific cache access parameters.

(2) Down-load the access pattern to local memory in each PE.

(3) Execute cache access by means of microprogram.

(4) Collect statistical data using hardware monitor.

In an experiment, 40,000 instances of cache access are made in each PE. All processors in the cluster, including CC, have their own caches, and consequently, nine snooping caches are connected to the two-way-interleaved common-bus. Since an access pattern stored in the local memory of each PE is fetched by microprogram, the effect of local memory access is corrected during the data measurement.

The end state of the same access pattern is used as the initial state of the cache since the focus is on the stable state of the snooping cache. An empty cache might be used as the initial state; however, it causes an excessively high memory access ratio. This is because, in an empty cache, the first access to each cache block always causes a memory access avoidable in the stable states by the inter-cache data transfer feature.

In PIM/c, each PE has a hardware monitor composed of a 40-bit counter and a circuit to select the event to be counted. This facility makes it possible to collect various statistical data such as the number of common-bus commands, the cache hit ratio, and the bus snooping wait time, without any overhead.

5.4. Results

5.4.1. Locality

Figure 5.1 shows the relationship between locality (controlled by the access interval) and cache throughput efficiency and number of bus commands. Figure 5.2 shows the effects of locality on the exclusive ratio in the cache. In Figure 5.2, the exclusive ratio which the access pattern itself has as its own characteristic is 10%. This means that 90% of access is to shared data.

Concerning Figure 5.1, in the low locality region, the cache throughput efficiency is low due to frequent miss-hits and invalidations. This is because the exclusive ratio is low in the case of low locality, and frequent coherence maintenance operations are necessary. This suggests that the invalidation protocol is not effective in low locality. On the other hand, as the locality increases, the exclusive ratio in the cache increases, and high exclusive ratio values in the cache can be obtained in the high locality region.(Figure 5.2) For example, values of more than 70% exclusive ratio are achieved in the cache access, even if the exclusive ratio in the access pattern itself is only 10%. This is because the invalidation mechanism erases unused copies in other caches. High exclusive ratios in the cache are desirable in order to reduce the overhead in the snooping cache because the exclusive cache block needs no coherence maintenance operation. Thus, the invalidation scheme is effective, especially in a case of high locality .

5.4.2. Share ratio

Figure 5.3 shows the relationship between share ratio and cache throughput efficiency, number of bus commands and external hit ratio. The external hit ratio is the probability with which inter-cache data transfers (as opposed to memory access) occur in cache miss cases.

A high share ratio causes frequent fetch/invalidation and tends to decrease throughput efficiency. However, the external hit ratio increases as the share ratio increases, and consequently, in the high hit ratio region, the data waiting time in cache miss cases can be reduced by inter-cache data transfers. In the high share ratio cases, the inter-cache data transfer feature prevents significant degradation of throughput efficiency which would be caused without this feature. In other words, caches in other PEs can be used as large secondary caches because the inter-cache data transfer is faster than main memory access.

5.4.3. Snooping cache overhead

Table 5.2 shows the composition of snooping cache overhead in detail. The bus waiting overhead is the most significant degradation factor in the PIM/c snooping cache. Since the bus request wait is due to access contention on the

common-bus, this is an inevitable drawback of the snooping cache architecture utilizing a common-bus.

Another major overhead factor is bus snooping. Table 5.2 suggests that 1/4 of the total overhead is caused by bus snooping. The overhead caused by conflict of CAA is greater than that caused by conflict of CDA. The reason for this is the fact that CAA conflicts may occur on every bus command. On the other hand, CDA conflicts occur only when data is transferred between caches in different PEs.

6. Discussion

6.1. The invalidation protocol

In this section, the characteristics of the invalidation scheme used for maintaining cache coherence are studied using the above experimental results.

As discussed in section 3.2, there are disadvantages in the broadcast and invalidation schemes in terms of unnecessary writes in broadcast and frequent miss-hits in invalidation. In Figure 6.1, the effects of write ratio on overhead for the broadcast and invalidation schemes are illustrated. The average waiting cycle on cache access due to common-bus operations is calculated on the basis of the following assumptions.

(1) The broadcast command occupies the common-bus during two cycles (a cycle for the address and a cycle for the data).

(2) The miss-hit ratio is 0% in the broadcast scheme, which means that the cache is large enough.

(3) Concerning other parameters such as the miss-hit ratio in the invalidation scheme, experimental results are used.

The invalidation scheme shows better performance than the broadcast scheme in the region of high write ratios. This is because the broadcast scheme causes frequent updating in high write ratio cases.

The invalidation scheme does not work well in all situation. However, it is suitable for PIM because the cache access characteristics in PIM have high locality and high write ratios.

6.2. The effect of interleaving on cache

As described in section 4.4, access contention on CAA/CDA memory causes the bus snooping overhead. Therefore, in order to reduce this overhead, hardware enhancements should be made so that the EU and the bus snooping hardware can access memory for CAA/CDA in parallel. For further improvement, interleaving could be introduced into CAA/CDA RAMs without the use of a much more advanced technology than that currently used in PIM/c. For example, CAA could be eight-way-interleaved and CDA could be two-way-interleaved. The current CAA and CDA are composed of 8 and 2 banks, respectively. To interleave these RAM banks, 10% more logic circuits should be added to the current VLSI controller for the address/data selector and parallel access control. The results in table 5.2 suggest that about 20% of the total cache overhead could be reduced by interleaving in CAA/CDA RAMs.

6.3. The architecture of a TCMP

Figure 6.2 shows the effect of cache hit ratio on common-bus utilization. Bus utilization decreases as hit ratio increases because a high hit ratio causes fewer bus commands. From Figure 6.2, the upper limit of the number of PEs that can be connected through a common-bus can be estimated, assuming a constant bus throughput.

Figure 6.3 shows the relationship between hit ratio and the maximum number of PEs in TCMP. In this graph, the bus utilization is limited to 60%, so that the total cycles consumed by waiting for the release of a busy bus will not exceed the total cycles consumed by bus commands. It should be noted from Figure 6.3 that, in the PIM/c snooping cache, a 9-PE TCMP is the optimum at a 90% hit ratio. Consequently, application programs on PIM must achieve hit ratios greater than 90% in order to obtain low cache overheads. Furthermore, Figure 6.3 suggests that more than 9 PEs can be installed in TCMP for those

applications in which hit ratios higher than 90% can be expected. For example, at 95% hit ratio, about 16 PEs, could be connected in a TCMP utilizing the current snooping cache architecture and hardware.

7. Conclusions

In this paper, the design concepts for the PIM/c snooping cache are described. The snooping cache mechanism is now integrated in one VLSI cache controller. Several fundamental characteristics of the snooping cache have been investigated using a small scale prototype of the PIM/c.

The main results obtained through our investigation are the following.

① The invalidation scheme on cache coherence maintenance is effective for high locality and high write ratios. In this case, a high exclusive ratio in the cache can be achieved because unused copies in the other caches are erased by invalidation. As a result, the coherence maintenance overhead on write access can be reduced.

② The inter-cache data transfer mechanism reduces the performance degradation of high share ratios. The reason for this is the fact that the inter-cache data transfer is faster than main memory access, and caches in other PEs can be used as large secondary caches.

③ Bus snooping overhead due to CAA/CDA contention takes 1/4 of the total cache overhead. Furthermore, the use of interleaving in CAA/CDA memories will reduce cache overhead by 20%.

④ The maximum number of PEs in a TCMP is limited mainly by the hit ratio. In PIM/c, a 9-PE TCMP is optimum at a 90% hit ratio. However, about 16 PEs can be connected with the current cache architecture and hardware if a greater than 95% hit ratio is expected.

In future work, the following issues should be included.

(1) A detailed analysis of the relationship between cache access parameters and performance.

(2) An investigation of lock mechanism and cache protocol.

(3) An evaluation using KL1 benchmark programs.

Acknowledgements

The authors would like to thank Dr. Shun'ichi Uchida, manager of the research department of ICOT, for his guidance and support; and Dr. Kazuo Taki, chief of 1st ICOT Laboratory, for his helpful discussions. This research was sponsored by ICOT.

References

[1] S. Uchida, K. Taki, K. Nakajima, A. Goto and T. Chikayama, "Research and Development of the Parallel Inference System in the Intermediate Stage of the FGCS Project," Proceedings of the International Conference on Fifth Generation Computer Systems 1988, pp. 3-15, 1988.

[2] K. Ueda and T. Chikayama, "Design of the Kernel Language for the Parallel Inference Machine," Computer Journal, Vol. 33, No. 6, pp. 494-500, 1990.

[3] A. Goto, M. Sato, K. Nakajima, K. Taki and A. Matsumoto, "Overview of the Parallel Inference Machine Architecture," Proceedings of the International Conference on Fifth Generation Computer Systems 1988, pp. 208-229, 1988.

[4] A. Goto, K. Taki, T. Nakagawa and M. Sugie, "Parallel Inference Machine PIM/c - Global Structure -," Proceedings of the 40th Annual Convention IPS Japan, 2L-1, 1990 (in Japanese).

[5] J. Archibald and J. Baer, "Cache Coherence Protocols: Evaluation Using a Multiprocessor Simulation Model," ACM Transaction of Computer Systems, Vol. 4, No. 4, pp. 273-296, 1986.

[6] A. Goto, A. Matsumoto and E. Tick, "Design and Performance of a Coherent Cache for Parallel Logic Programming Architectures," Proceedings of the 16th ISCA, pp. 25-33, 1989.

[7] T. Chikayama and Y. Kimura, "Multiple Reference Management in Flat GHC," Proceedings of ICLP '87, pp. 276-293, 1987.

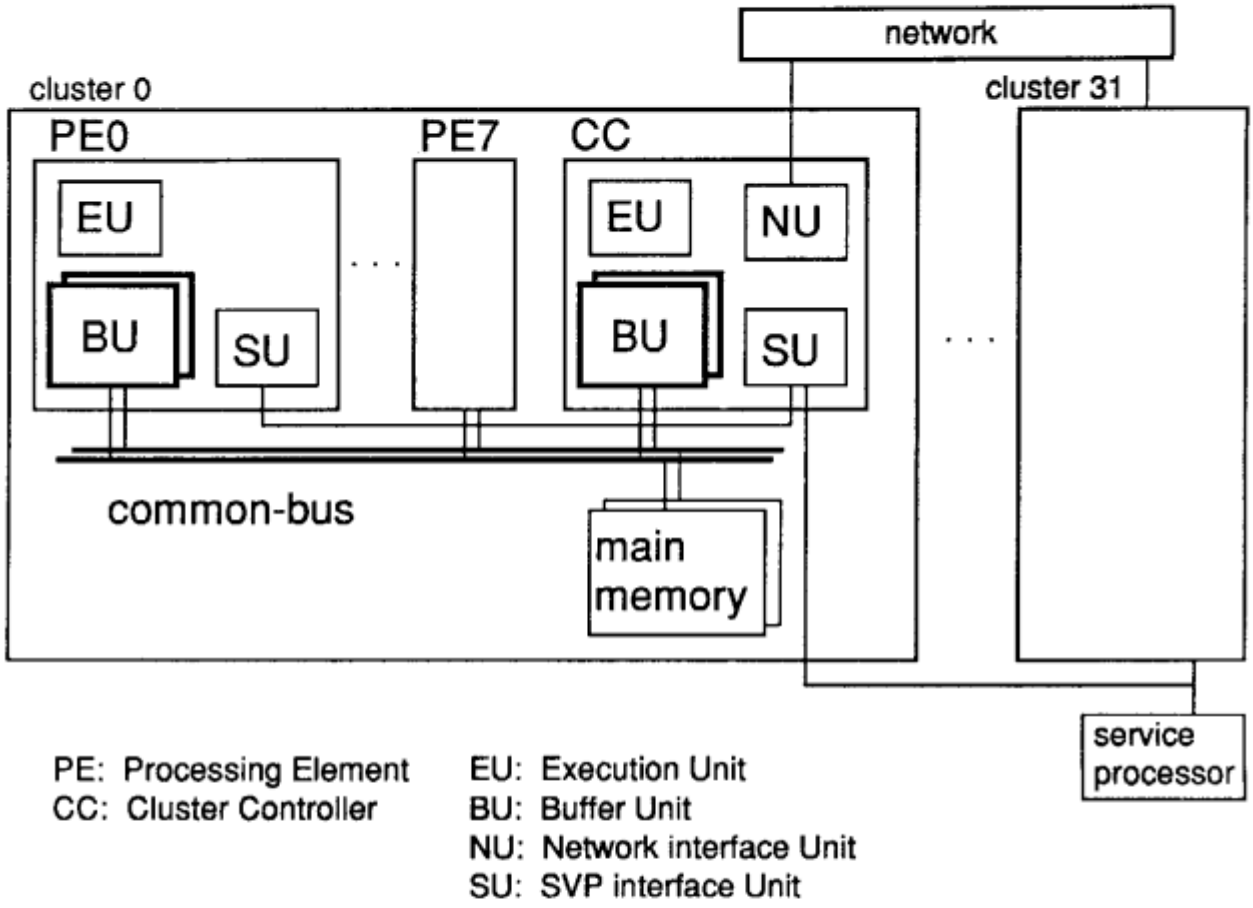


Fig. 2.1. The organization of the PIM/c prototype.

(PIM/c: Parallel Inference Machine model C)

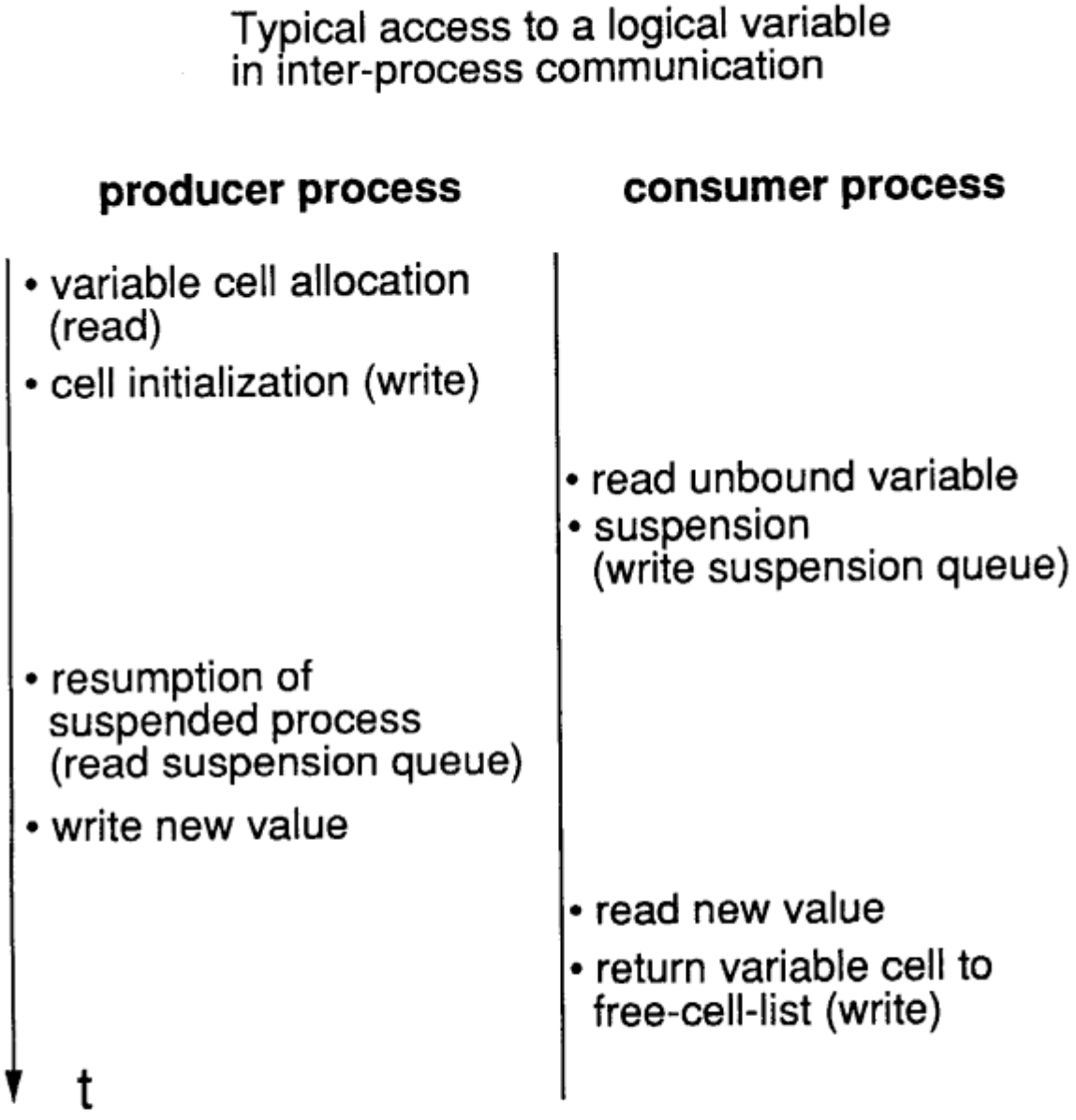


Fig. 3.1. A memory access sequence in a KL1 program.

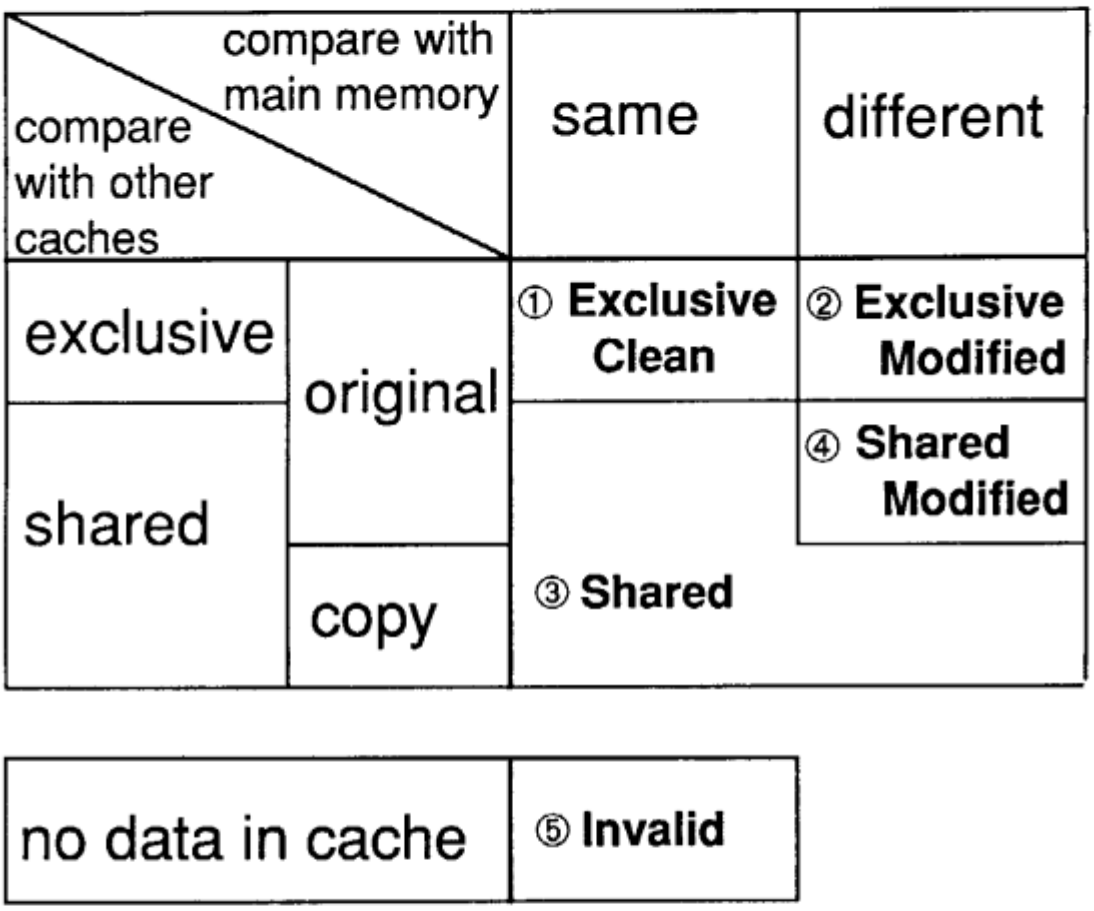


Fig. 3.2. States in PIM/c snooping cache.

Table 3.1.

EU commands and corresponding cache operations

cache state / EU commands	EM	EC	SM	S	I/MH
Read	- / EM	- / EC	- / SM	- / S	F(SO) / C->S M->EC
Write	- / EM	- / EM	I / EM	I / EM	FI(SO) / C->EM M->EM

bus command / cache next state

EU: Execution Unit

- cache state
  - EM: Exclusive Modified
  - EC: Exclusive Clean
  - SM: Shared Modified
  - S: Shared
  - I: Invalid
  - MH: Miss-Hit
- bus command
  - I: Invalidate
  - F: Fetch
  - FI: Fetch & Invalidate
  - SO: Swap Out
- next state of fetch command depend on bus operation
  - C : inter-cache transfer
  - M: memory read

Table 3.2.

Bus commands and corresponding cache operations

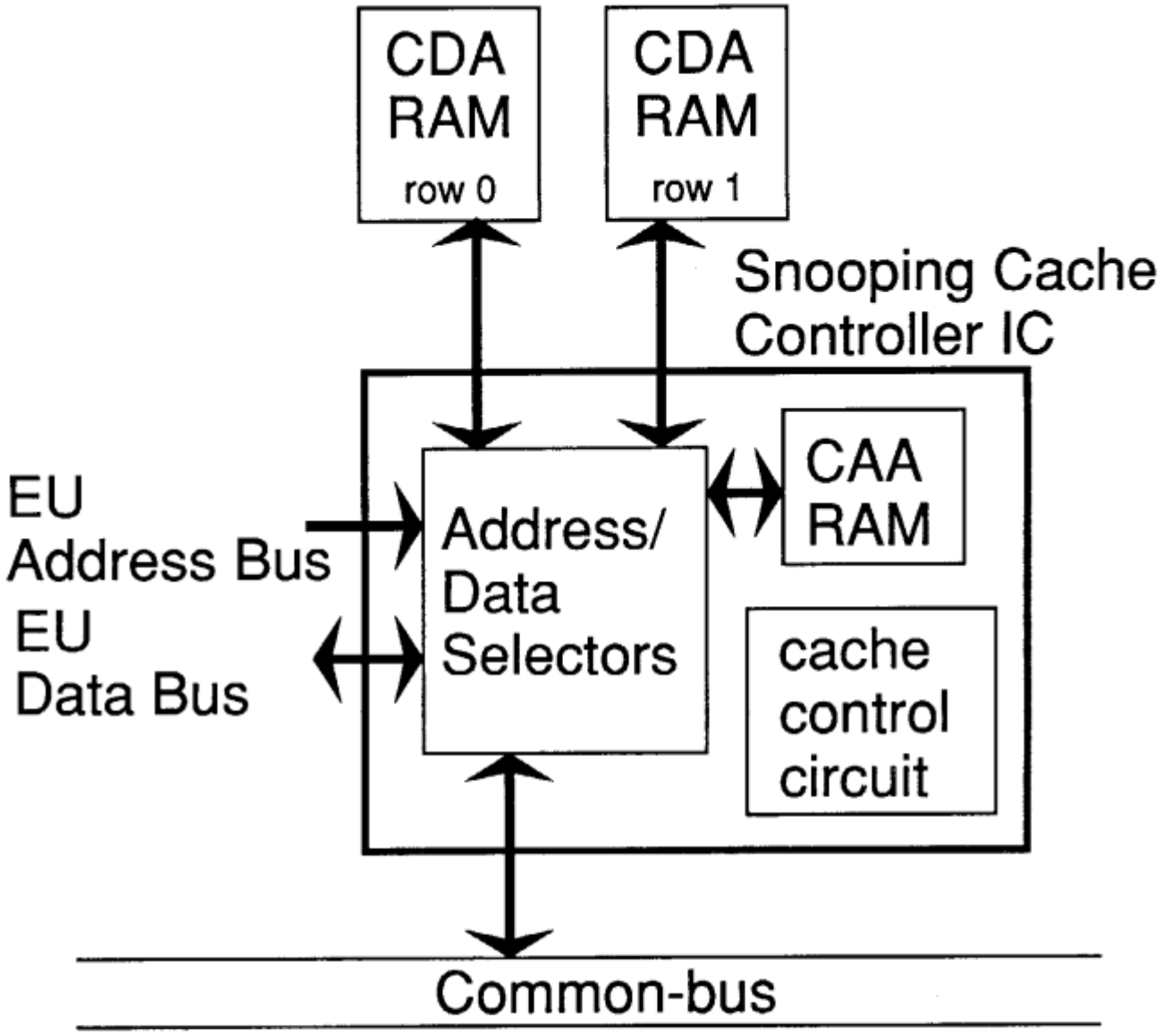
cache state / bus commands	EM	EC	SM	S	I/MH
F F - SO	H / SM	H / S	H / SM	H / S	- / I
FI FI - SO	H / I	H / I	H / I	H / I	- / I
I	- / I	- / I	- / I	- / I	- / I

bus reply / cache next state

- cache state
  - EM: Exclusive Modified
  - EC: Exclusive Clean
  - SM: Shared Modified
  - S: Shared
  - I: Invalid
  - MH: Miss-Hit

- bus command
  - I: Invalidate
  - F: Fetch
  - FI: Fetch & Invalidate
  - SO: Swap Out

- bus reply
  - H: hit reply

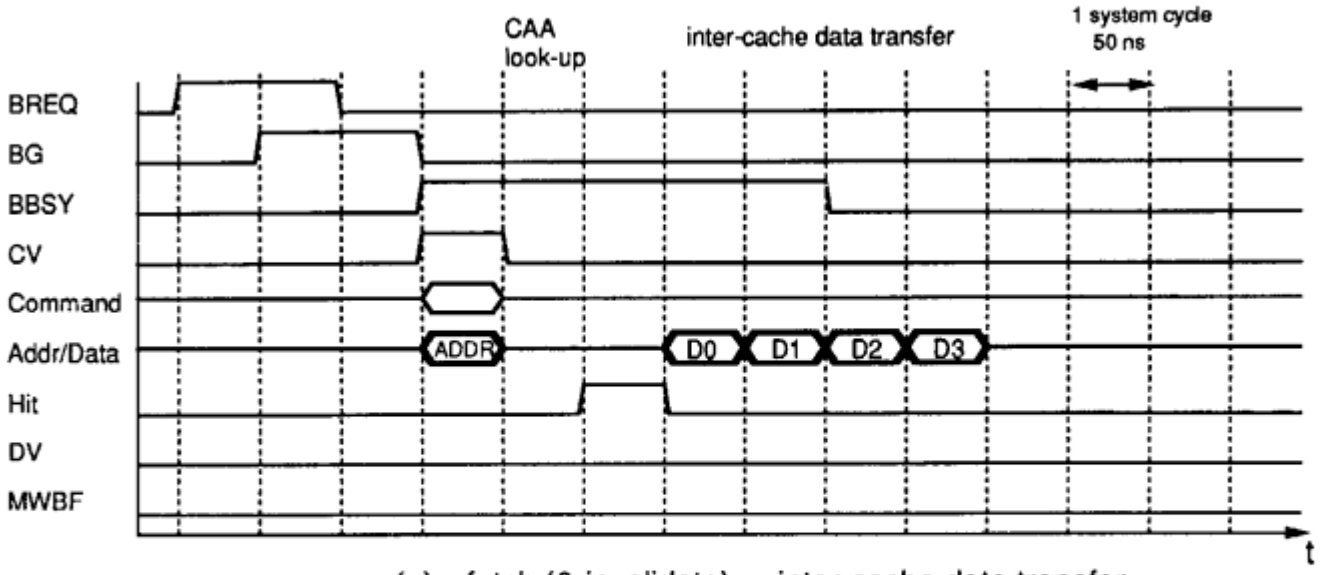


CAA: Cache Address Array
CDA: Cache Data Array
EU: Execution Unit

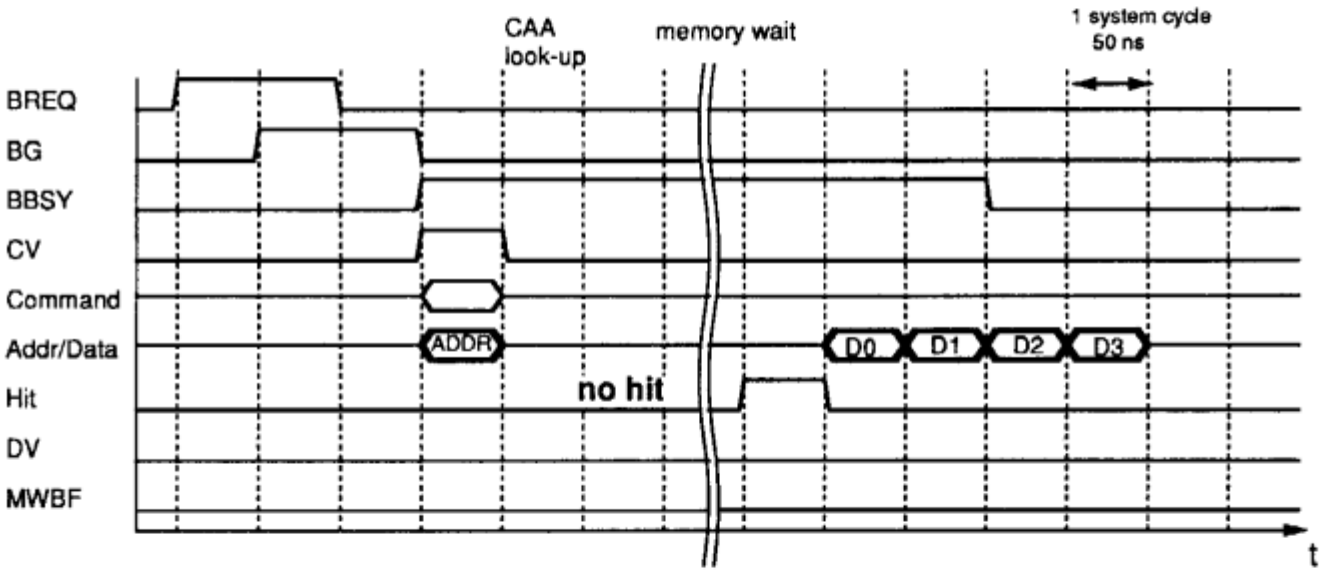
Fig. 4.1. Snooping cache organization.

Table 4.1.
Common-bus signals

Group	Signal Name	Number of bits	Function
Common-bus	Data/Address	45	5 bytes Address/Data (with parity bits) (1 byte tag and 4 bytes data)
Snooping Cache Control	CV (Command Valid)	1	Specifies the timing of common-bus commands
	Command	5	Common-bus command
	Hit	9	The accessed block is stored in specified cache
Memory Access Control	DV (Data Valid)	1	Specifies the timing of data transferred from memory
	MWBF (Memory Write Busy Flag)	1	Flag showing full write buffer in main memory
Bus Arbitration	BREQ (Bus Request)	9	Common-bus request
	BG (Bus Grant)	9	Reply to bus request
	BBSY (Bus Busy)	1	Common-bus busy signal

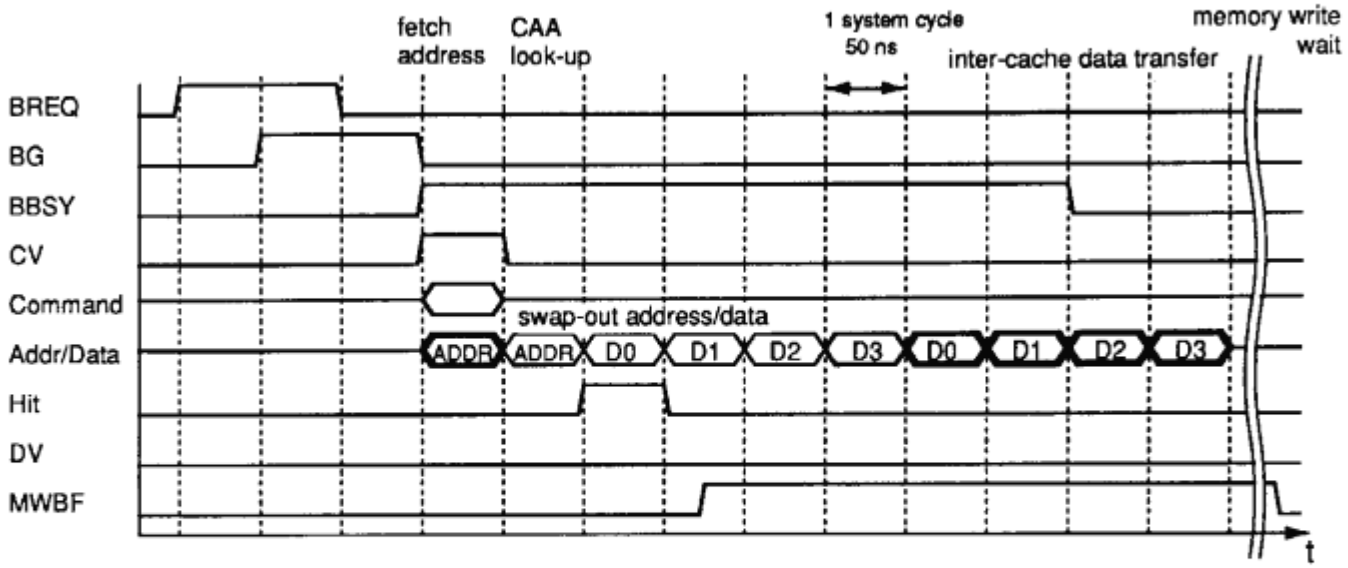


(a) fetch (& invalidate) -> inter-cache data transfer

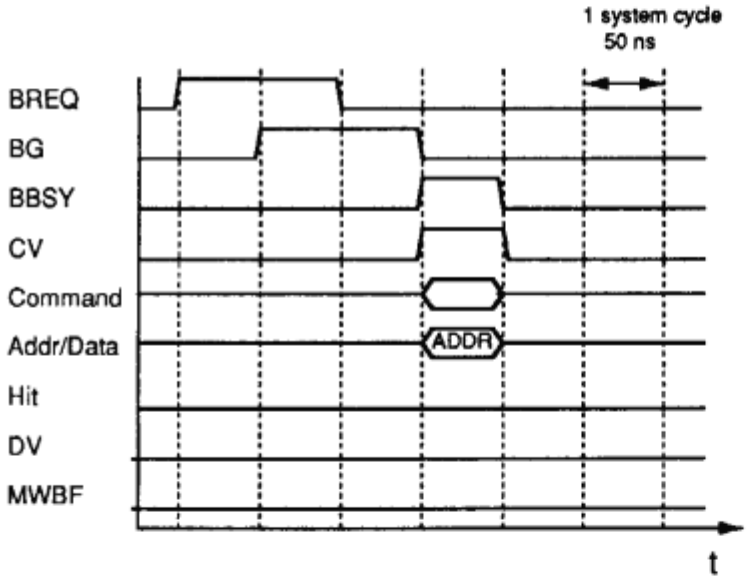


(b) fetch (& invalidate) -> main memory access

Fig. 4.2. (a) & (b) Common-bus operation time charts.



(c) fetch (& invalidate) with swap out -> inter-cache data transfer



(d) invalidation

Fig. 4.2. (c) & (d) Common-bus operation time charts.

Table 4.2.
PIM/c snooping cache architecture summary

Item	Specification
Protocol	"five-state snooping cache protocol" (invalidation scheme)
Size	2 sets * 1 k columns * 2 banks = 4 k blocks 1 block = 4 words = 20 bytes
Controller LSI	30 k gates CMOS LSI (320 I/O pins) 50 ns system cycle (4 phase clock)
CAA RAM	36 bits * 1 k columns/bank on-chip RAM (15 ns access time)
CDA RAM	45 bits * 8k words/bank 64 k bits static-RAM (15 ns access time)
Common-BUS	2-way-interleaved structure throughput: 20 M words/s for each bus

Table 5.1.
Cache Access Parameters

Parameter	Range	Default Value	Notes
Access interval to a block	1 - 2048 (cycle)	8 (cycle)	Controls locality
Share ratio	12.2 - 97.6 (%)	91.3 (%)	
Write ratio	10 - 90 (%)	30 (%)	

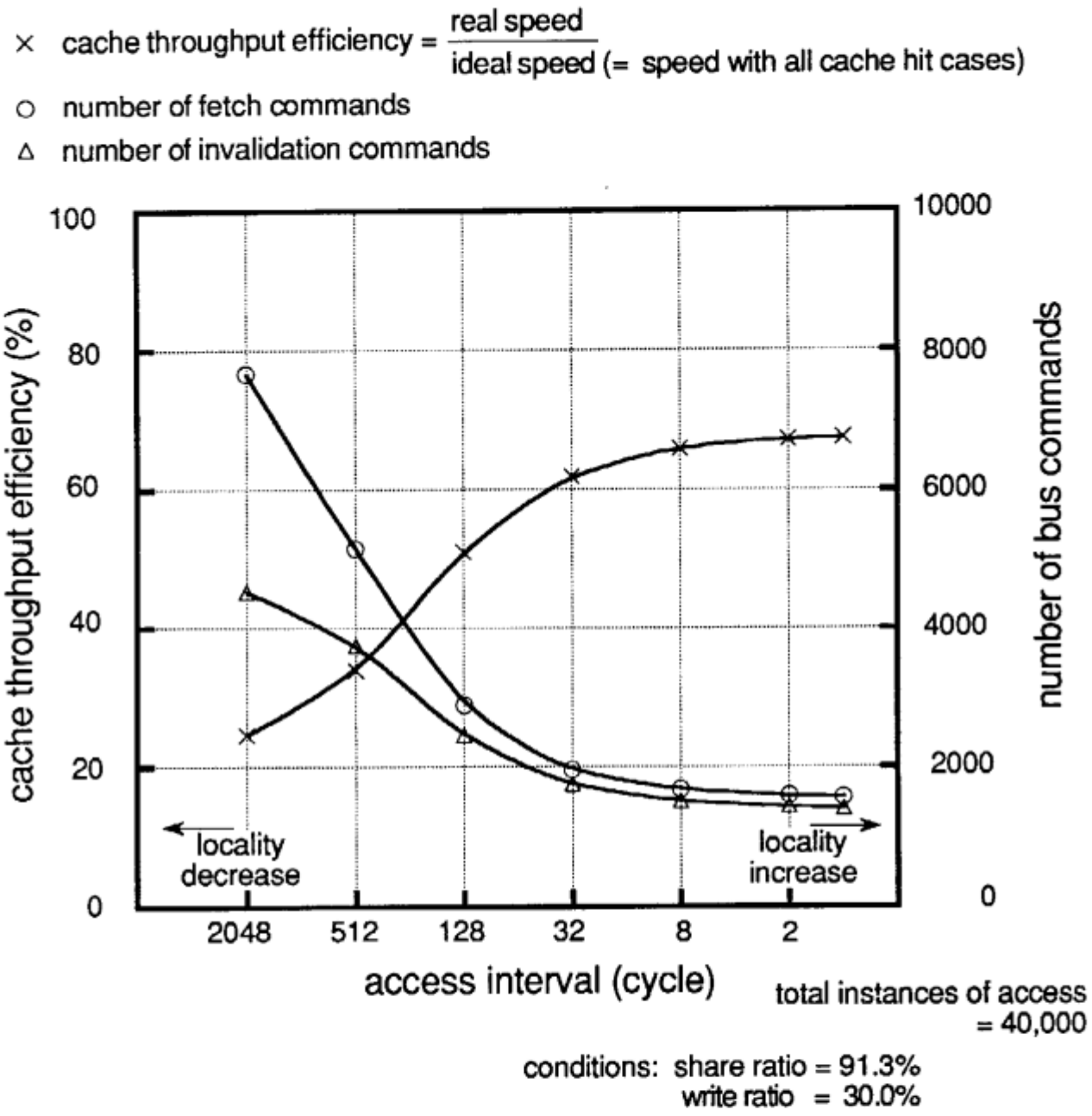


Fig. 5.1. Cache throughput efficiency and number of bus commands as a function of locality.

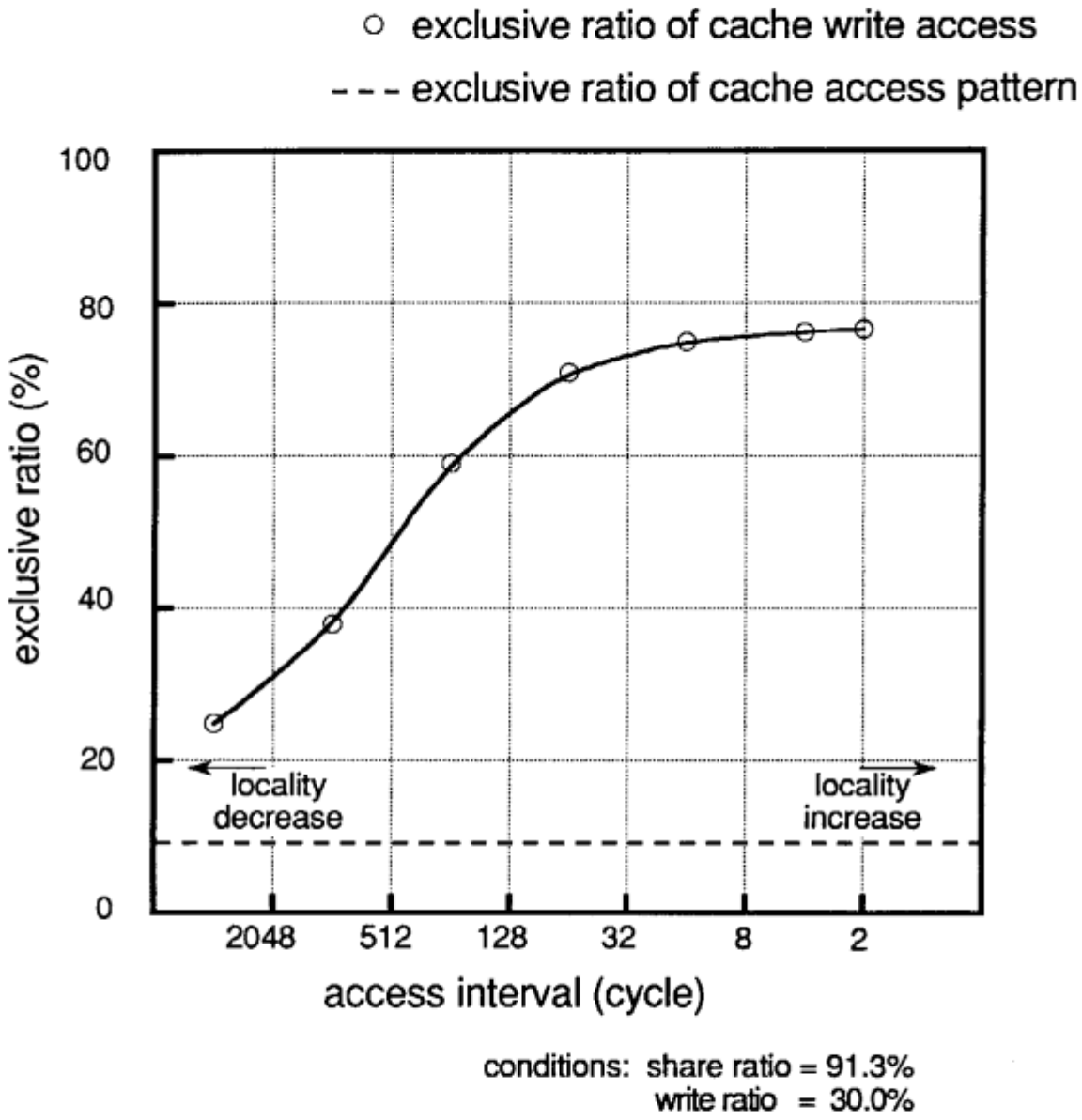


Fig. 5.2. The exclusive ratio as a function of locality.

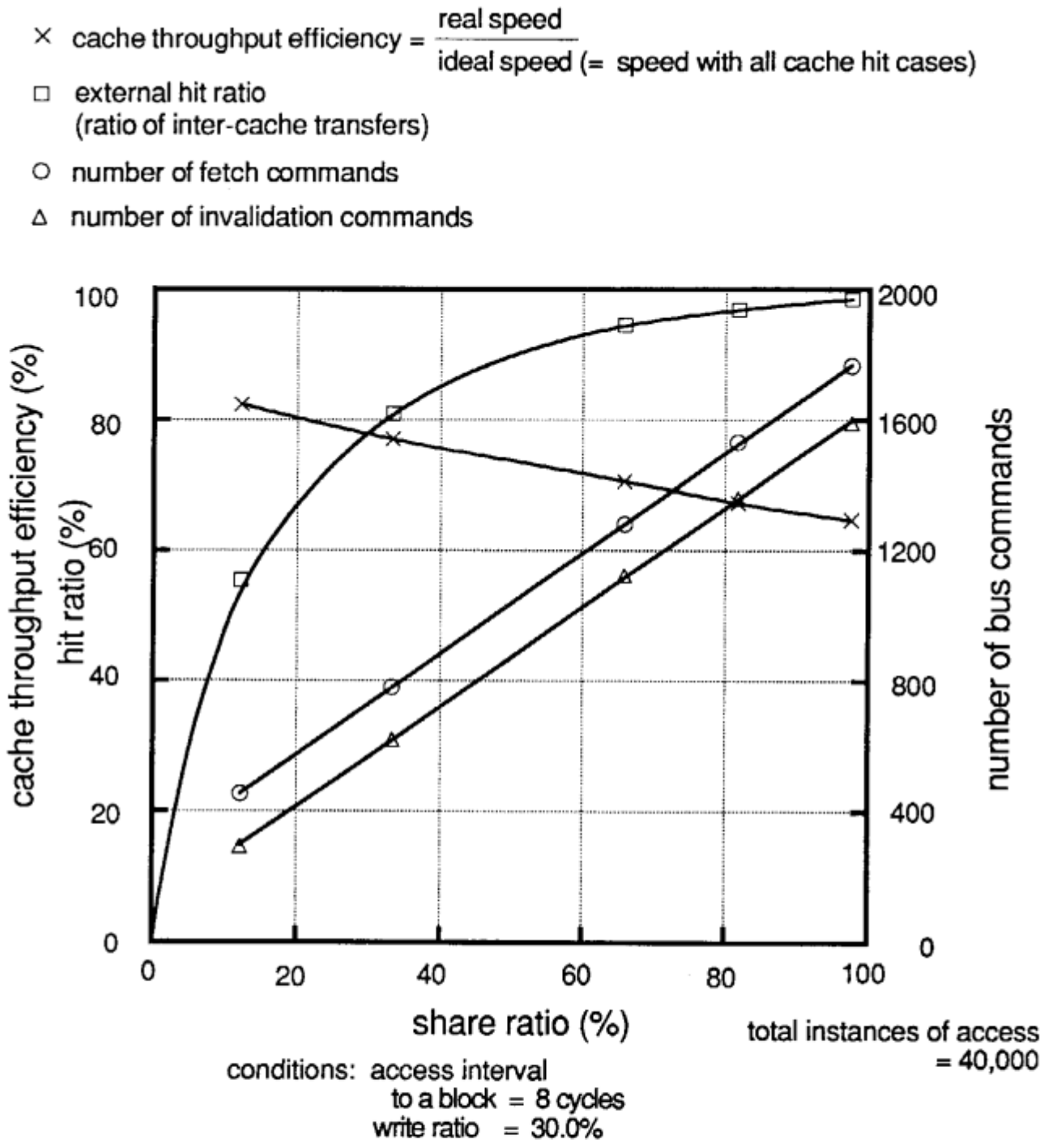


Fig. 5.3. The cache throughput efficiency, number of bus commands and the external hit ratio as a function of the share ratio.

Table 5.2.

The composition of snooping cache overhead

Overhead	Waiting cycle	Speed reduction ratio (%)
Access conflict of CAA	10124	4.2
Access conflict of CDA	6596	2.7
Bus request waiting	29219	12.0
Bus command execution	15480	6.4
Main memory waiting	1403	0.6

CAA: Cache Address Array
CDA: Cache Data Array

total instances of cache access = 40,000

conditions: share ratio = 91.3%
write ratio = 30.0%
access interval
to a block = 8 cycles

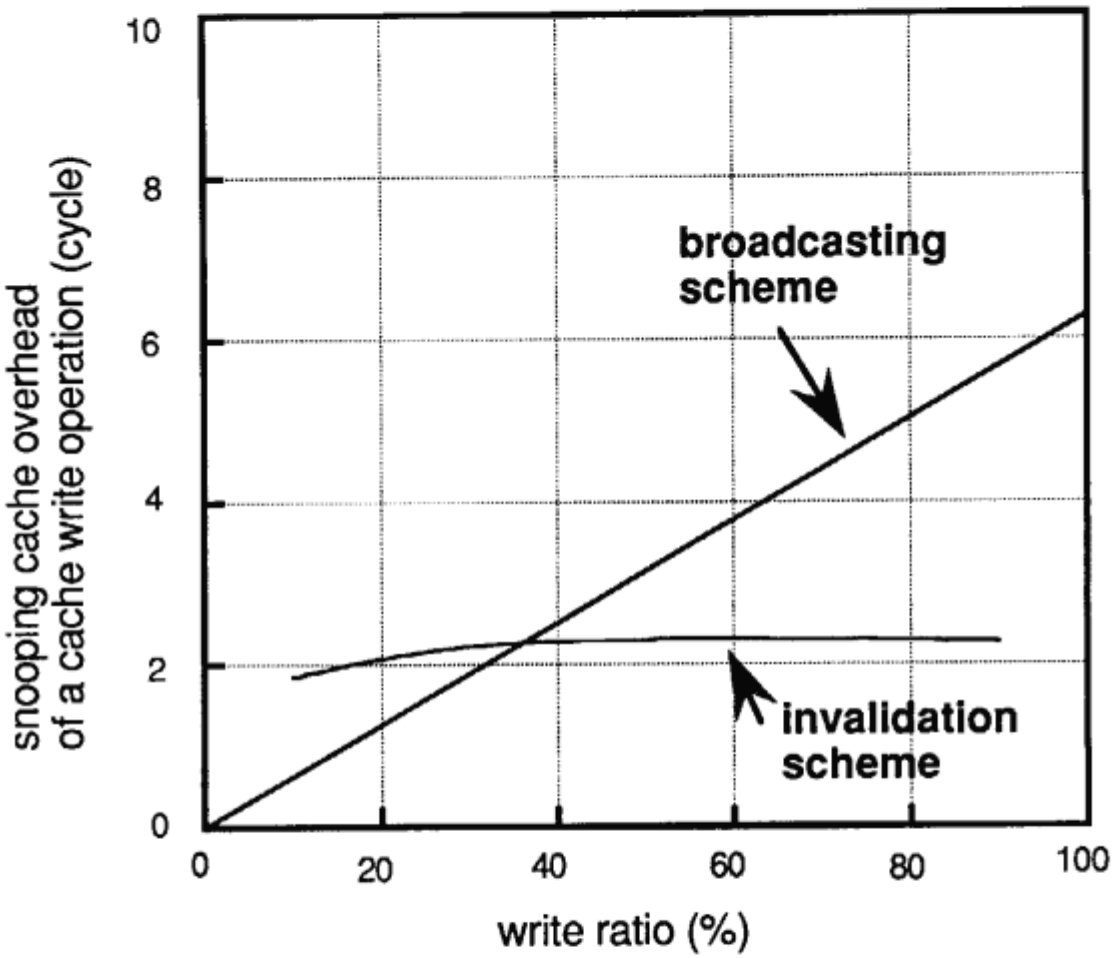


Fig. 6.1. The snooping cache overhead as a function of the write ratio.

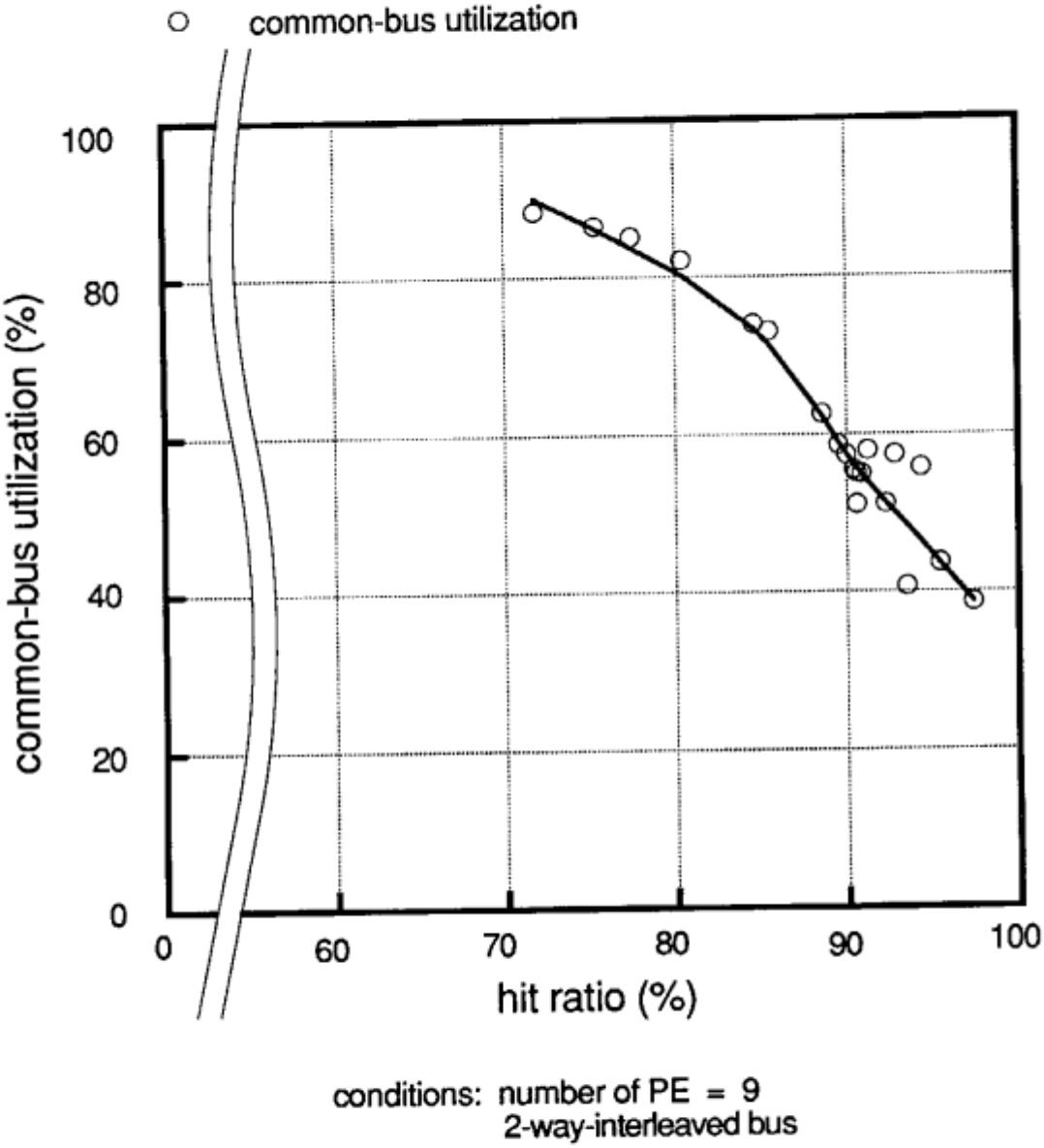


Fig. 6.2. Common-bus utilization as a function of hit ratio.

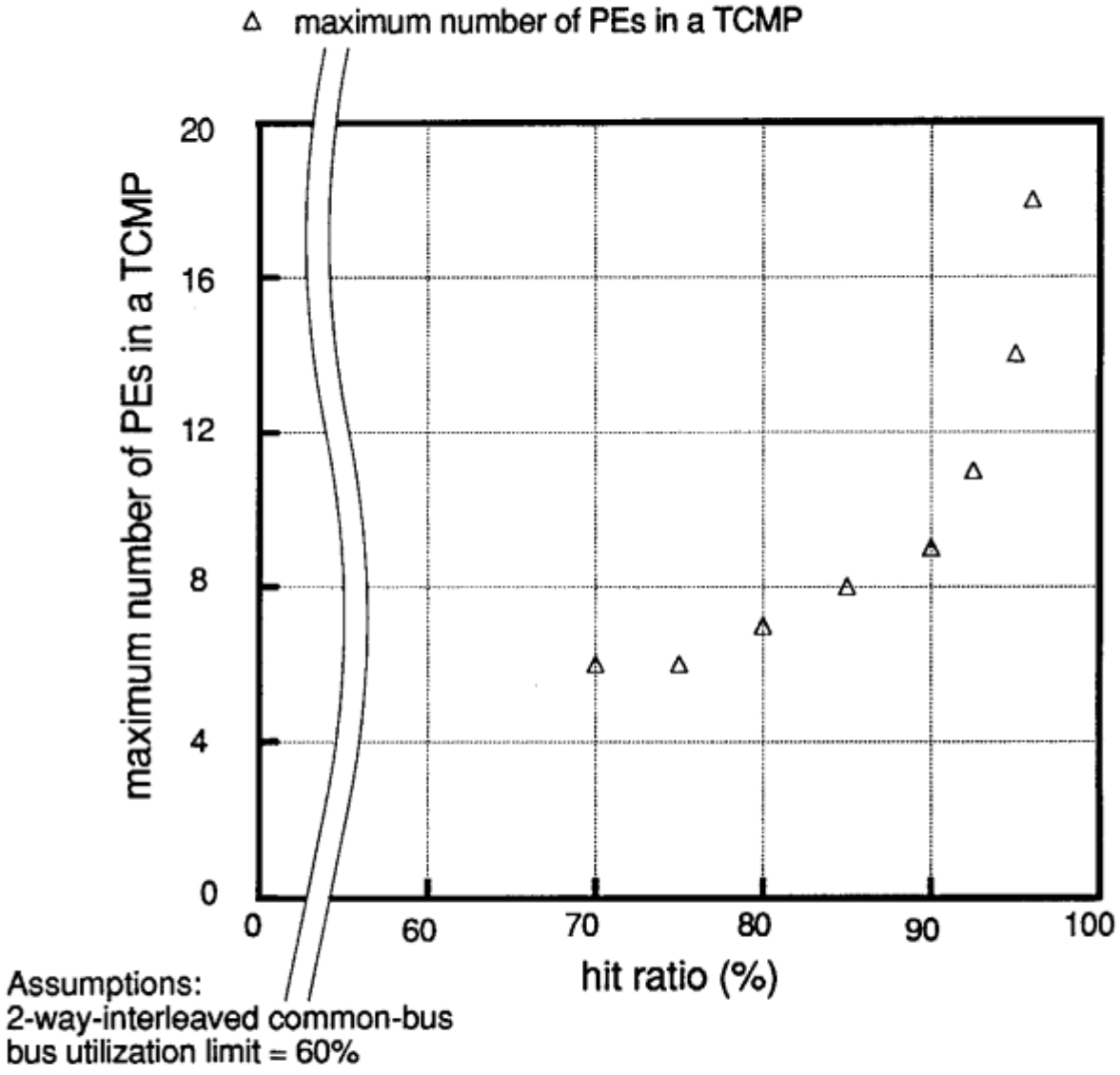


Fig. 6.3. The maximum number of PEs in a TCMP as a function of hit ratio.