ICOT Technical Report: TR-709

TR-709

An Intelligent Cache Memory Chip Suitable for Logical Inference

by

K. Yasuda, K. Furutani, A. Maeda,
S. Wakano, H. Nakashima, Y. Takeda &
M. Yamada (Mitsubishi)

November, 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

Title An Intelligent Cache Memory Chip suitable for Logical Inference

Authors Kenichi Yasuda*, Nonmember,
Kiyohiro Furutani*, Member,
Atsushi Maeda*, Nonmember,
Shoichi Wakano***, Nonmember,
Hiroshi Nakashima**, Nonmember,
Yasutaka Takeda**, Nonmember and
Michihiro Yamada****, Member

Affiliation / Address

*LSI Laboratory,Mitsubishi Electric Corporation,
4-1 Mizuhara, Itami, 664, Japan

**Information Systems and Electronics Development
Laboratory, Mitsubishi Electric Corporation,
5-1-1 Ofuna, Kamakura, 247, Japan

***LSI Design Center, Mitsubishi Electric
Engineering Company Limited,
4-61-5 Higashino, Itami, 664, Japan

****Kita-Itami Works, Mitsubishi Electric
Corporation,
4-1 Mizuhara, Itami, 664, Japan

Acknowledgement

The authors are thankful to C.Tunetomo and M.Kanehama for their test and CAD support.

The authors also wish to thank Dr. H.Komiya, Dr. T.Nakano, and Dr. T.Yoshihara for their encouragement and support of this research program, and also thank Dr. S.Uchida and researchers in the ICOT for giving us the opportunity to conduct this research.

Running Head Intelligent Cache memory chip

Summary

We have newly developed a VLSI intelligent cache memory chip which constitutes one processor element of a Parallel Inference Machine(PIM/m) system.

This cache memory chip contains 61K transistors and 80K bits memory cells. The chip measures 14.47mm x 14.84mm and is fabricated by using 1.0um CMOS double metal technology. The cache memory chip implements a hardware support called "Trail Buffer" which is suitable for the execution of logic programming languages. We have determined the cache memory size by practical simulation taking the relationship between the chip size and hit-ratio of the cache memory into consideration. The scan test method and the special commands to access every memory cell are applied to enhance the testability. This chip itself operates at a cycle time of 30MHz. The typical power consumption is 2.5W with a 5.5V power supply at 16.7MHz operation.

With this cache memory chip, the CPU board of the PIM/m is now tuned, for 16.7 MHz operation and has attained 1.5 MLIPS(logical inference per second), which is the highest performance as an inference machine in the world.

1. Introduction

Application specific processors such as Ivory[1], ELIS[2], and Pegasus[3] have been optimized for symbolic processing languages such as Lisp and Prolog, and have played an important role in the research and development of artificial intelligence(AI). However, higher performance processors are still required for the study of AI. One solution is to operate a lot of the processors in parallel and the another is to reduce the machine cycle of the processor.

The Parallel Inference Machine(PIM/m)[4][5] system has been developed. The 256 processor elements are connected to form a two-dimensional mesh network. To create a PIM/m system, with large numbers of parallel machines in a realistic size, requires the development of VLSI. To meet these demands, we have developed a new VLSI intelligent cache memory chip which constitutes one processor element of the PIM/m.

The architectural features of the cache memory chip are an embeded program counter and a hardware support called "Trail Buffer", which is suitable for the execution of logical programming languages. The cache memory chip employs Harvard architecture, which requires increasing the number of pins. To prevent an increase in the number of pins, we embeded a program counter and a common bus for the instruction cache block shared by the address and data. It also helps control increases in the chip area. In the execution of logic programming languages, a variable assignment by unification should be reset when the inference falls into contradiction. In this process, the trail buffer is useful for reseting data quickly. Since unification is one of the most important processes in the execution of logic programming, the support of unification with hardware is effective.

It is important to take the relationship between the chip size and hit-ratio of the cache memory into consideration. We have determined the memory size by practical simulation.

This chip includes the 1K word instruction cache memory, 1K tags for accessing the off-chip 4K word data cache memory, and the DRAM controller.

Section 2 describes the novel architectures in the device configuration. In Section 3, the testability of the chip is described. Section 4 covers the hierarchical layout. A general description of the chip and the results of our evaluation are summarized in Section 5.

2.Device configuration

Fig.1 shows the configuration of one processor element in the PIM/m system. The cache memory chip, processor chip, network chip, and memory devices (SRAM and DRAM) make up the compact processor element.

Fig.2 shows a block diagram of the cache memory chip, which consists of three functional blocks; an instruction cache block, a data cache block, and a main memory interface block as a DRAM controller. The instruction and data cache blocks employ the physical address caching scheme with on-chip translation lookaside buffers (TLB's) [6].

The instruction cache block contains direct mapped, 1K word cache memory; 256 tags; 32-entry, 2-way set associative TLB; and a program counter(PC). In the data cache block, there are a 32-entry, 2-way set associative TLB; 1K tags for an off chip, direct mapped, 4K word cache memory(SRAM); and a trail buffer. The main memory interface block controls the DRAM refresh and the write-back for data cache coherency, and uses SEC-DED Hamming code ECC.

2-1. Embedded Program Counter

Recently Harvard architecture has been gaining ground as a style of high-performance microprocessors [7]. It is suitable for a processor which pipelines the decode and execution of the instruction, because the processor unit can fetch the instruction and operand simultaneously. However, the requirement of dedicated address, and data buses for both instruction and operand accesses, makes its implementation difficult when the processing unit and cache unit are separated on different chips. This is because the increased number of pin counts affects the chip size and packaging costs.

To solve this problem, the bus for the instruction cache block is shared by the address and data signals. The cache memory chip has a program counter, which is a copy of the one in the processor chip. The program counter in the cache memory chip is set via the instruction data bus, and incremented after the fetch operation. Because of the sequential nature of the instruction address, the processor chip needs to feed the instruction address only when there is a branch operation. Thus we were able to reduce the pin count by 32 with little less in performance.

2-2. Support for Logical Inference

To accelerate logical inference, this cache memory chip has hardware called "Trail Buffer". In logic programming language, there is an operation called "Backtrack". When a backtrack operation occurres, a variable assignment by unification should be reset. In this operation, when a processor unifies a variable, the address of the variable should be stored in "Trail Stack". The trail buffer, which can hold up to 16 addresses of assigned variables, is a cache of the trail stack [8].

The functions of the trail buffer are to store the addresses of assigned variables, to supply address data of the variable to be reset, and to remember the number of reset times.

Fig.3 shows a block diagram of elements around the trail buffer(TRB). When a variable is assigned, the address of the variable is pushed into the Trail Buffer Data Register(TRBDR), from the address bus through the selector. At the same time the data stored in TRBDR is sent to TRB, addressed by the Trail Buffer Address Register(TRBAR), and TRBAR registers +1. When backtrack occurres, the address data pops to the address bus from TRBDR, and the variable is reset by the address data. Then the TRB's data is addressed by the TRBAR and is stored in TRBDR for the next reset operation. The TRBAR then registers -1. To achieve high speed operation, we use TRBDR. The processor can determine the number of reset times by reading TRBAR data, and reset the TRB by writing "0" in the TRBAR.

It is estimated by simulation, that the trail buffer improves the speed of the logical inference by 10% (average) by accelerating backtrack.

2-3. Memory Configuration

The proper choice of the TLB and cache memory configuration is important, in order to reduce the hardware while keeping a high hit-ratio. We justified the configuration of this chip with a simulation. To simulate the device's behavior during execution, we used a Prolog compiler. Since the Prolog compiler is a large and practical program, the simulated result seemed to be a good measure of the device's performance.

Fig.4 and Fig.5 show the simulated hit-ratio versus the TLB size and cache size, respectively.

Hit-ratios of 99.83% and 99.88% are obtained by the 32-entry, 2-way, set associative TLB's for the instruction and operand address translations, respectively. Although high hit-ratios are achieved with 16 x 2 entries, we employed 32-entry, 2-way, set associative TLB's to maintain a high hit-ratio even with much larger application programs. Hit-ratio's of 94.4% and 99.2% are obtained with the 1K word instruction and direct mapped, 4K word operand cache memories, respectively. Considering performance and chip area, we decided on the memory configuration as described in the device configuration section of this papers. Therefore, this cache memory chip can successfully compensate for the speed gap between the processor chip and the DRAM.

The write-back method is used for cache coherency, . The write-back method executes the main memory access in only 15% of its miss-hit write operations, according to the simulation; while the write-through method, in general, requires a main memory access in every write operation. Since this chip has a high hit-ratio, the write-back méthod has less of a chance to access the main memory, and therefor is more effective .in spite of the increase of its hardware.

3. Testability

Testability is also an important issue in a cache memory chip with logic and memory parts. The control circuits(logic parts) are tested using the scan test method. It's test pattern is automatically generated by the in-house software "MULTES" [9]. The chip has a single scan path with 420 scan registers which increases the chip area by only 2%.

In this device there are a total of 80k bit memory cells. Normally a scan test requires a lengthy serial test pattern, however, in order to reduce the test time, we employed special commands (a miscellaneous command which is described later) to access every RAM's cell, through the 40-bit data bus shown in Fig.2. With these special commands, the test time was reduced to 12% of the conventional scan test. Furthermore, the commands allows us to perform the RAM pause test and voltage bumping test; tests which are difficult using the conventional scan test. The result was that failure detection capability was increased. For large scale logic in memory devices (like this cache memory chip), a special command which can access the RAM directly is an effective method reducing the memory test time and increasing the failure detection capability.

4. Layout

This device is laid out by cell based methodology, integrating 610K transistors in a 14.47mm x 14.84mm device area, as shown in Fig.~~5~~6. The RAM's and PLA's are automatically generated.

In the layout of large chips, the reduction of signal propagation delay is important. This chip is laid out hierarchically to minimize the wire length. This device is divided into three functional blocks and every block is formed by some functional parts. In order to pay special attention to the critical path, we first lay out the functional parts like the hit-check part as the basic level, then we lay out three functional blocks as the upper level. At last, as the top level, we adjust three blocks to one chip. With this hierarchical layout we gain a high operating speed of 30MHz.

This hierarchical layout also shortens the time of the layout. The layout design and the verification are completed in one week.

5. Chip General Description and Performance Evaluation

5-1 Command Set

Table 1 shows the command set of this cache chip. The command code is supplyed to the Data Cache Block and the Instruction Cache Block through two independent command buses, DCCOM and ICCOM, respectively, The operation of each commands is described as follow;

(For Instruction Cache)

(1)nop

no operation

(2)fetch

read data from Instruction Cache according to the Instruction Program Counter(IPC), and register +1 in IPC for the next fetch operation.

(3)set_PC

set Instruction address to IPC.

(For Data Cache)

(1)nop

no operation

(2)undo

pop Trail Buffer(TRB); then data and address buses input data to Data Cache Memory(DCM). register -1 in {TRBAR}.

(3)reset_TRB

write "0" to TRBAR.

(4)read_TRBAR

read TRBAR

(5)read

read DCM according to data from address bus.

(6)set_misc_command

set misc_command code in Command Register (MCR).

(7)cancel

cancel the read command which is set just before the cycle.

(8)write

write the data from data bus in DCM using address data from address bus.

(9)unify_write

write the data from data bus in DCM according to address data from address bus, write same address data in TRB; +1 is registered in TRBAR.

(10)execute_misc_read

execute command which is set in MCR using address data from address bus.

(11)execute_misc_write

execute command which is set in MCR using address data from address bus and data from data bus.

5-2 Miscellaneous Command

For the maintenance of resources such as registers and TLB's, this cache chip supports miscellaneous commands. The MCR designates the object of the operation, and the execution of the operation is triggered by execute_misc_read or executc_misc_write commands. Part of the miscellaneous command set is shown in Table 2, and their functions are described as follows:

(1)re_execute_D_cache

reexecute read or write operation of Data cache which has been suspended by TLB miss.

(2)clear_D_Cache

invalidate date of DCM according to data of address bus, if the data of the block is renewed, the data is written back to main memory.

(3)pop TRB

read data from TRB; -1 is registered in TRBAR.

(4)push TRB

write data of data bus to TRB. TRBAR is incremented.

(5)re_execute_I_Cache

reexecute fetch command which has been suspended by TLB miss.

(6)clear_I_Cache

invalidate data of Instruction Cache Memory(ICM) addressed by IPC.

(7)read_ICM

read data of ICM addressed by IPC.

(8)write_ICM

write data from data bus in ICM addressed by IPC.

(9)read_IPC

read data from IPC.

(10)write_IPC

write data from data bus in IPC.

The miscellaneous command can also access other registers and memories, such as DTLB(Data cache TLB) which is designated by MCR.

The processor element can maintain all resources by these miscellaneous commands, and we can also test resources by using these commands.

5-3 Chip Production Result And Performance Evaluation

Fig.6 shows a microphotograph of this chip. The chip is fabricated by using 1.0um CMOS double metal technology. The package is a 361 pin PGA.

This chip achieves an instruction cache data access of 17ns. The instruction cache ready signal and the data cache ready signal are output at 27ns and 4ns(next cycle), respectively, from the beginning of the clock cycle (falling edge of the clock). Fig.7 shows a Instruction-Cache ready signal and Instruction-Cache data signal. The chip itself operates at a cycle time of 30MHz. Fig.8 shows the shmoo plot of the operating cycle time. The typical power consumption is 2.5W with a 5.5V power supply, at a 16.7 MHz operation. These features are summarized in Table 3.

Table 4 shows the append performance of the CPU board of the PIM/m and other machines for parallel and sequential logic programming languages. The performance of the PIM/m is about 4 times greater than its predecessor; PSI-II [8].

6. Conclusion

We developed the VLSI intelligent cache memory chip to create high performance inference machines. The instruction cache block, data cache block and main memory interface block as the DRAM controller are integrated on the chip. The embedded program counter reduces the pin count, and the trail buffer memory facilitates execution of the logic programming language. The cache configuration is optimized by practical simulation to achieve a high hit-ratio. Special commands are employed for the RAM's test which is included on the chip. The hierarchical layout provides a high operating speed and reduce the design time.

With this cache memory chip, the CPU board of the PIM/m is ready to handle 16.7 MHz operation and it has attained 1.5MLIPS(logical inference per second), waking it the highest performance inference machine in the world.

Reference

[1] C.Baker, D.Chan, J.Cherry, A.Corry, G.Efland, B.Edward, M.Maton, et al., "The Symbolics Ivory Processor: A 40 Bit Tagged Architecture Lisp Microprocessor," in Proc. of Intl. Conf. on Computer Design, pp.512-515, 1987.

[2] K.Watanabe, A.Ishikawa, Y.Yamada and Y.Hibino,
"A 32b LISP Processor," in Proc. of Intl. Solid-State Circuits Conf. (1987)

[3] K.Seo and T.Yokota, "Design and Fabrication of Pegasus Prolog Processor," in Proc. of Intl. Conf. on VLSI, 1989.

[4] S.Uchida, K.Taki, K.Nakajima, A.Goto, T.Chikayama, "Research and Development of the Parallel Inference System in the Intermediate Stage of the FGCS Project," Proc. of the

Intl.Conf. on Fifth Generation Computer Systems, pp.16-36 ,1988.

[5] H.Nakashima, Y.Takeda, K.Nakajima, H.Andou, K.Furutani, "A Pipelined Microprocessor for Logic Programming Languages," Proc. of Intl. Conf. on Computer Design, pp.355-359,1990.

[6] J.Cho et al., "A 40K Cache Memory and Memory Management Unit," ISSCC, pp.50-51, 1986

[7] Les Kohn et al., "Introducing the Intel i860 64-Bit Microprocessor," IEEE Micro, vol.9 no.4, pp.15-30, 1989.

[8] H.Nakashima and K.Nakajima, "Hardware Architecture of The Sequential Inference Machine : PSI-II," Proc. of 4th IEEE Simposium on Logic Programming, pp.104-113. (1987)

[9] T.Ogihara, Y.Koseko, G.Yonemori and H.Kawai, "Testable Design and Support Tool for Cell Based Test," Digest of International Test Conference, pp.1065-1071. (1990)

Figure Captions

Fig. 1 Configuration of one processor element

Fig. 2 Block diagram of the cache memory chip

Fig. 3 Block diagram of elements
around the Trail Buffer

Fig. 4 Estimation of TLB hit-ratio

Fig. 5 Estimation of Cache hit-ratio

Fig. 6 Photomicrograph of the cache memory chip

Fig. 7 Waveforms of signals

Fig. 8 Shmoo plot of the operating cycle

Table Captions

Table 1 Command set

Table 2 Part of the miscellaneous command set

Table 3 Features

Table 4 Append performance

code	mnemonic
00000	nop
00101	undo
00110	reset_TRB
00111	read_TRBAR
01000	read
01110	set_misc_command
01111	cancel
10000	write
11000	unify_write
11101	execute_misc_read
11111	execute_misc_write

(for Data-Cache)

code	mnemonic
00	nop
01	fetch
10	set_PC
11	nop

(for Instruction-Cache)

Table 1 K. YASUDA et. al

code	resources	exec_misc_read	exec_misc_write
00000		(nop)	re_execute_D_Cache
00010		(nop)	clear_D_Cache
01001	TRB	pop TRB	push TRB
10000		(nop)	re_execute_I_Cache
10010		(nop)	clear_I_Cache
10111	ICM	read_ICM	write_ICM
11001	IPC	read_IPC	write_IPC

Table 2 . K . YASUDA et. al

Configurations

- Instruction Cache Block
  - TLB : 32 entry x 2 way set associative
  - Tag : 22 bit x 256 entry, direct mapping
  - Cache : 40 bit x 1024 word
- Data Cache Block
  - TLB : 32 entry x 2 way set associative
  - Tag : 22 bit x 1024 entry, direct mapping
  - Cache (Off-chip) : 40 bit x 4096 word
- Main memory interface Block
  - Write back for data cache coherency
  - SEC-DED Hamming code ECC
  - DRAM refresh control
- Trail Buffer
  - 16 entry x 32 bit

Operating Speed (Vcc = 4.5V, Ta = 25 ℃)

Operating cycle time	: 32 nsec
Instruction cache ready from clock	: 27 nsec
Instruction cache data from clock	: 17 nsec
Data cache ready from clock	: 4 nsec (next cycle)

Supply voltage	5 volt
Power	2.5 watt (Vcc = 5.5V, @16.7MHz)
Package	361 pin PGA
Chip size	14.47 x 14.84 mm
Transistor Count	610 K
Technology	Psub Twin well 1 µm CMOS single poly-Si & double Al
Memory cell	CMOS 6Tr SRAM cell

Table 3 K.YASUDA et al

	KLIPS (PLPL/SLPL)
PIM/m (one processor element)	833/1510
PSI-II	179/ 430
Pegasus	/~350

PLPL:Parallel Logic Programming Language

SLPL:Sequential Logic Programming Language

Table 4. K. YASUDA

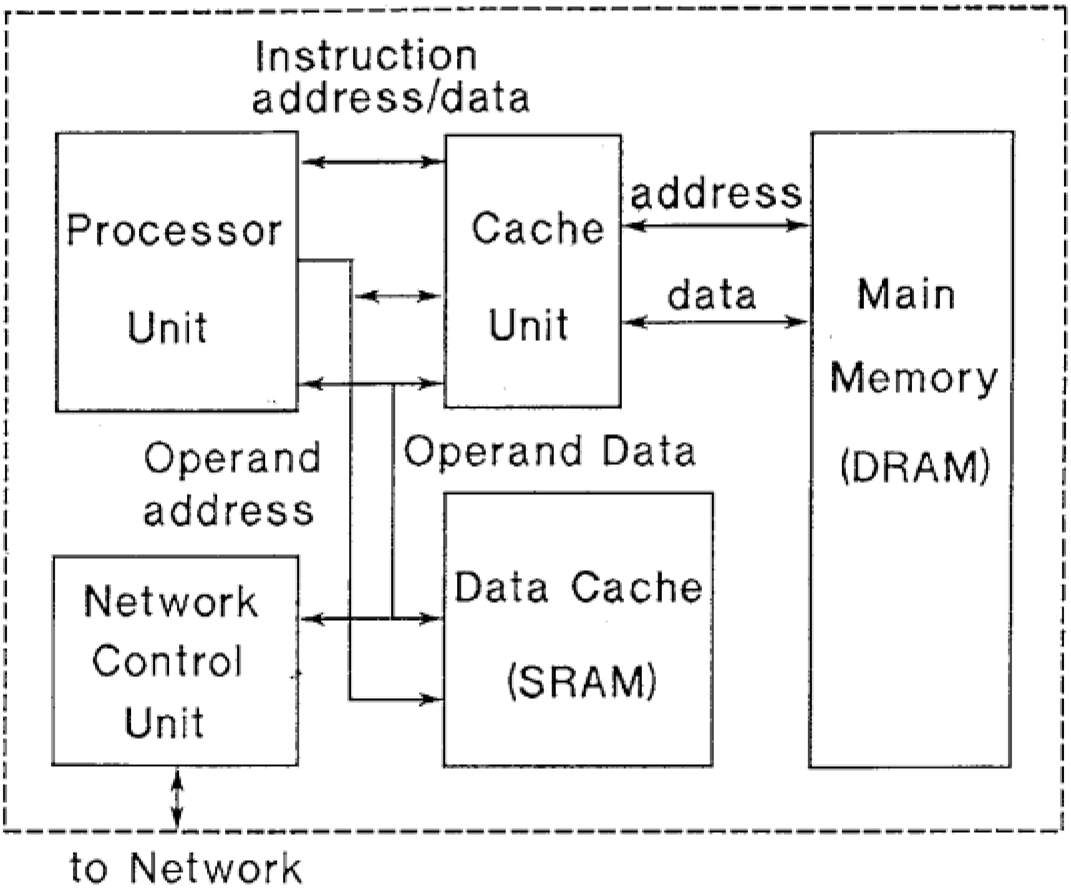

Fig. 1 K. YASUDA et. al

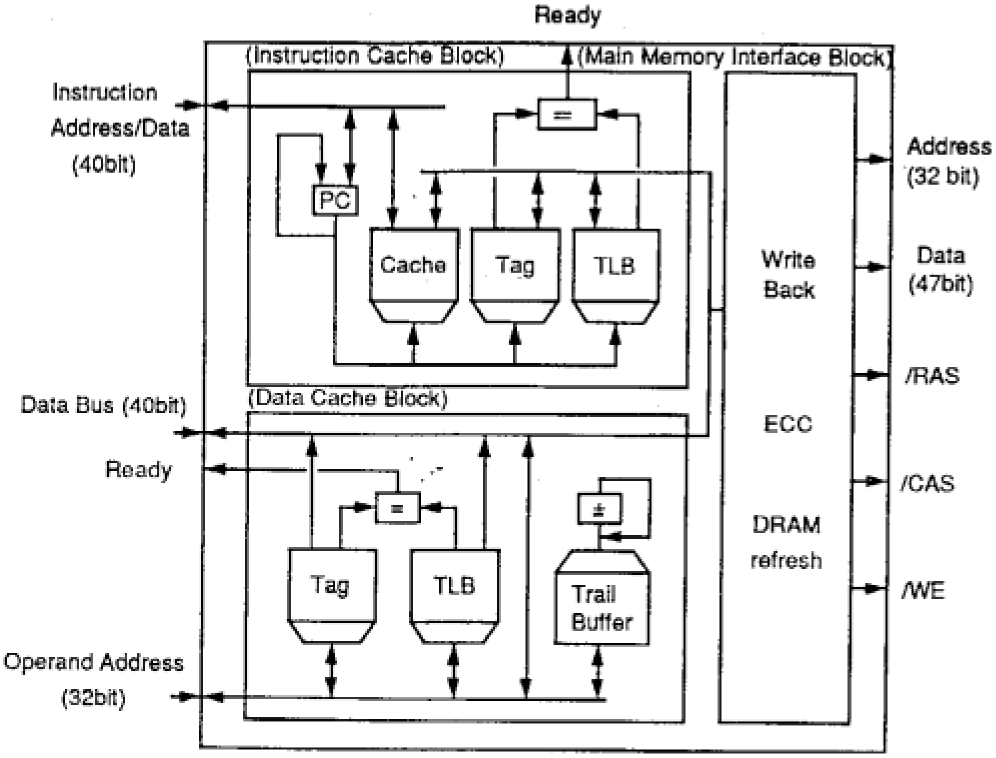

Fig. 2. K. YASUDA. et.al.

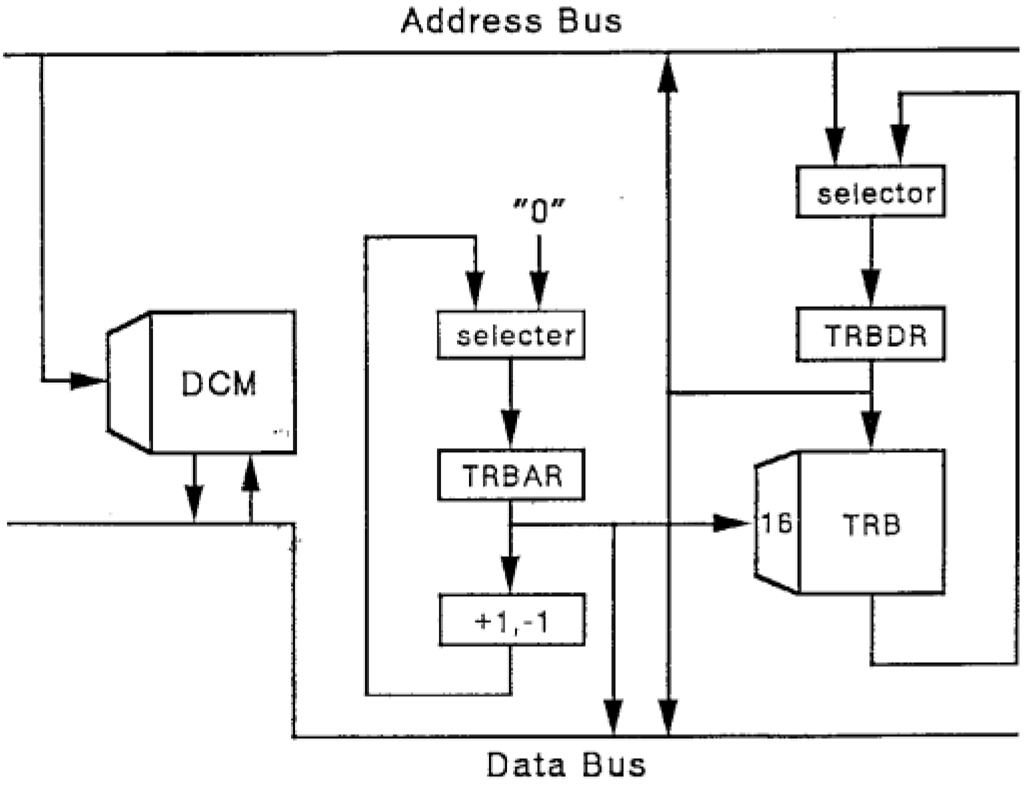

Fig. 3 K. YASUDA et. al.

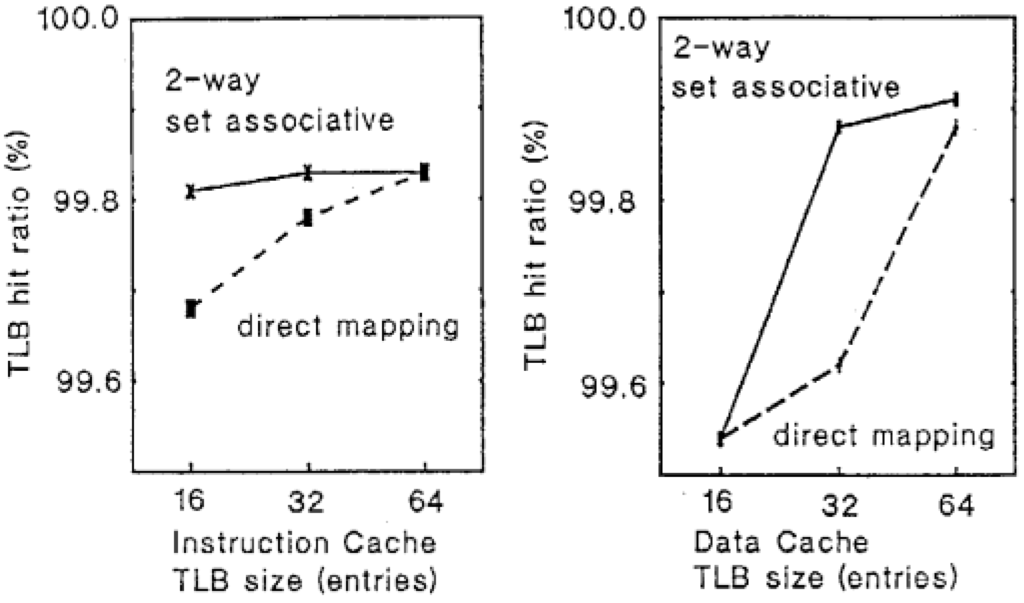

Fig. 4. K. YASUDA. et. al

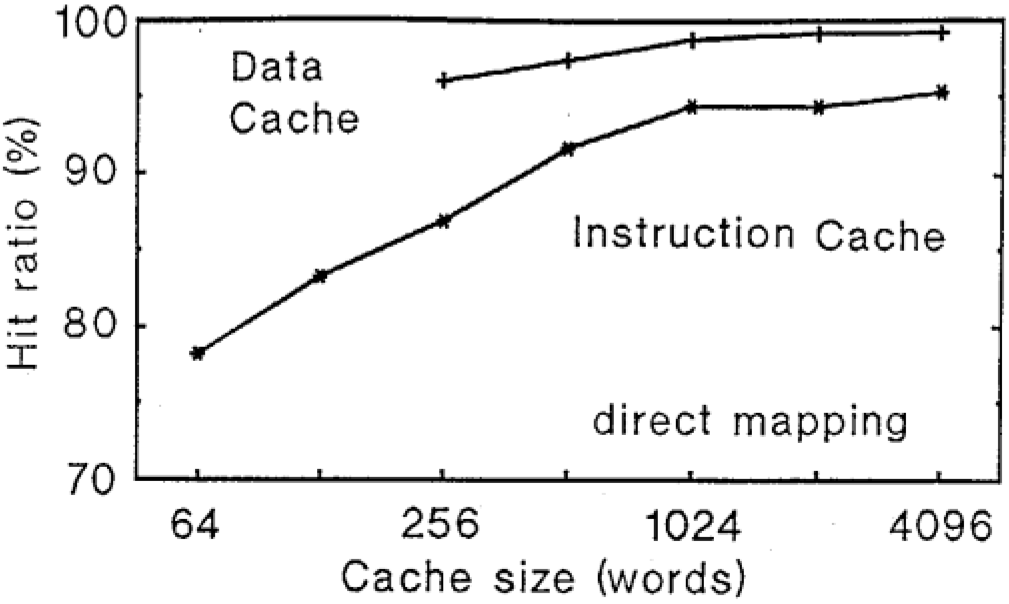

Fig. 5 K. YASUDA et. al.

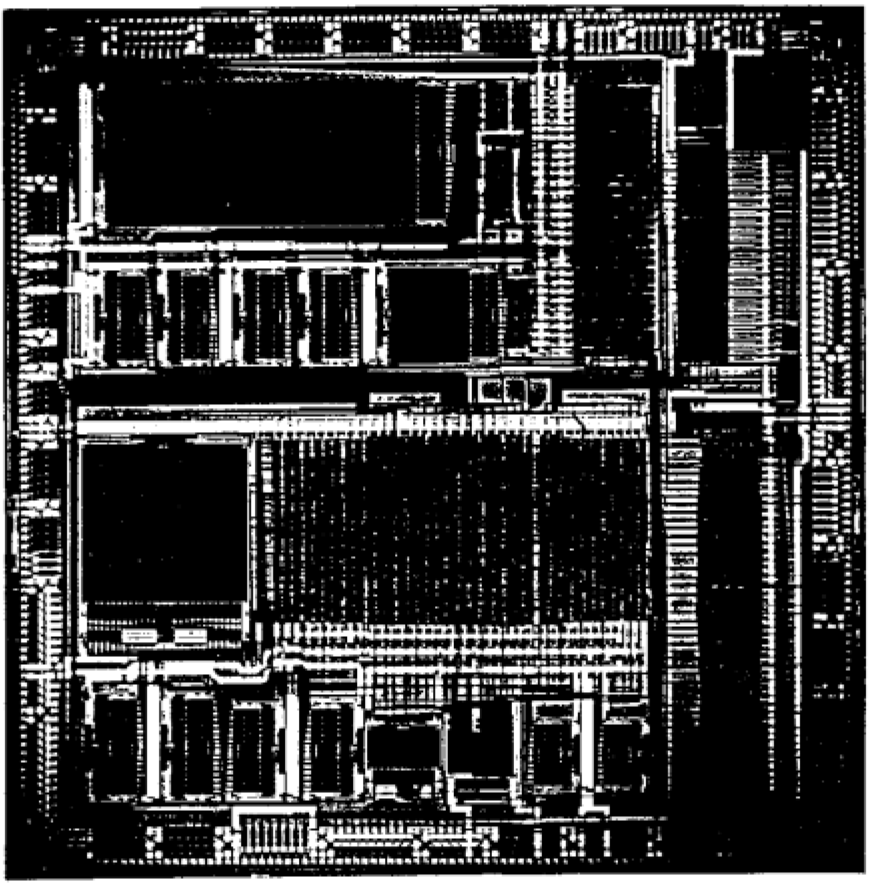

Fig. 6 K. YASUDA et. al.

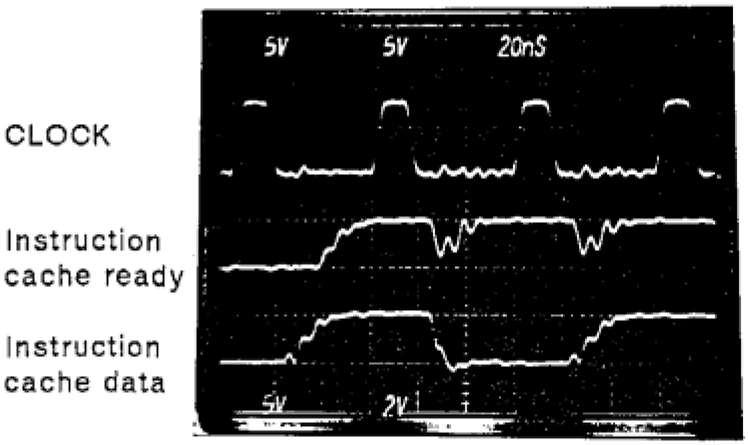

Fig. 7. K. YASUDA et. al

```
SHM1
  rate(ns)       7 6 6 6 6 6 5 5 5 5 5 4 4 4 4 4 3 3 3 3 3 2
                 0 8 6 4 2 0 8 6 4 2 0 8 6 4 2 0 8 6 4 2 0 8
VCC =  06.000 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP..
VCC =  05.900 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP..
VCC =  05.800 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP..
VCC =  05.700 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP...
VCC =  05.600 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP...
VCC =  05.500 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP....
VCC =  05.400 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP...
VCC =  05.300 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP...
VCC =  05.200 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP...
VCC =  05.100 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP....
VCC =  05.000 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP....
VCC =  04.900 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP....
VCC =  04.800 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP....
VCC =  04.700 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP....
VCC =  04.600 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP.....
VCC =  04.500 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP.....
VCC =  04.400 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP......
VCC =  04.300 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP......
VCC =  04.200 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP......
VCC =  04.100 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP.......
VCC =  4.0000 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP.......
VCC =  3.9000 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP.........
VCC =  3.8000 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP..........
VCC =  3.7000 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP.............
VCC =  3.6000 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPPPP..............
VCC =  3.5000 V  PPPPPPPPPPPPPPPPPPPPPPPPPPPP................
VCC =  3.4000 V  PPPPPPPPPPPPPPPPPPPPPPPPPPP.................
VCC =  3.3000 V  PPPPPPPPPPPPPPPPPPPPPPPPP...................
VCC =  3.2000 V  PPPPPPPPPPPPPPPPPPPPPPP.....................
VCC =  3.1000 V  PPPPPPPPPPPPPPPPPPPPP.......................
VCC =  3.0000 V  PPPPPPPPPPPPPPPPP...........................
```

Fig. 8 K. YASUDA et.al