Design of a DI model-based Content Addressable Memory for Asynchronous Cache

Jigjidsuren Battogtokh, Kyoung-Rok Cho
Department of Computer and Communication Engineering,
Chungbuk National University, Cheongju, Chungbuk, Korea

ABSTRACT

This paper presents a novel approach in the design of a CAM for an asynchronous cache. The architecture of cache mainly consists of four units: control logics, content addressable memory, completion signal logic units and instruction memory. The pseudo-DCVSL is useful to make a completion signal which is a reference for handshake control. The proposed CAM is a very simple extension of the basic circuitry that makes a completion signal based on DI model. The cache has 2.75KB CAM for 8KB instruction memory. We designed and simulated the proposed asynchronous cache including CAM. The results show that the cache hit ratio is up to 95% based on pseudo –LRU replacement policy.

Keywords: Cache, CAM, DCVSL, completion signal

1. INTRODUCTION

The generic concept associated with implementation of cache is to reduce the average access time of a main memory in the CPU [1]. The instruction cache is both smaller and a faster memory that stores copies of recent instructions fetched from program memory. The average latency of cache access is smaller when compared with the latency of a main memory access [2]. In general, the energy consumption of a cache accounts for significant part of the energy utilization in an embedded system. For instance, the cache employing the sleep mode approach prevents cache access until it is woken up. Within such design paradigm, asynchronous circuit techniques appear as a natural choice to realize the sleep mode functionality [3]. Thus, implementation of a handshaking protocol becomes responsible for activating active blocks, leaving other blocks in the wait-state. The asynchronous circuit employing delay-insensitive (DI) delay model is an option capable of both energy efficiency as well as performance enhancement. This paper presents a novel asynchronous instruction cache suitable for self-timed system. There are two salient features when compared with the conventional asynchronous cache designs. First, the cache design employs DI delay model, which operates correctly assuming positive, bounded but unknown delays in gates as well as wires communicating with main memory and CPU that employ bound delay model. Second, we employ content addressable memory (CAM) instead of the more conventional random access memory (RAM) for the cache tag storage which reduces the power consumption.

Section II presents the proposed asynchronous cache

architecture, CAM, completion signal generation using differential cascade voltage switch logic (DCVSL). Simulation results and conclusions are provided in sections III and IV respectively.

2. ASYNCHRONOUS CACHE ARCHITECTURE

A self-timed circuits have a number of computation blocks that are coupled through interconnection blocks. Main architecture of the asynchronous cache is shown in figure 1 that includes CAM, handshake block and related blocks. Since there is no system clock to coordinate operations between computational blocks, we employ handshake protocol [4]. The completion signal of a combinational block helps with the handshake protocol as reference signal in asynchronous circuit [5]. We design asynchronous cache using DI model for CAM and a bundled model for an external memory.
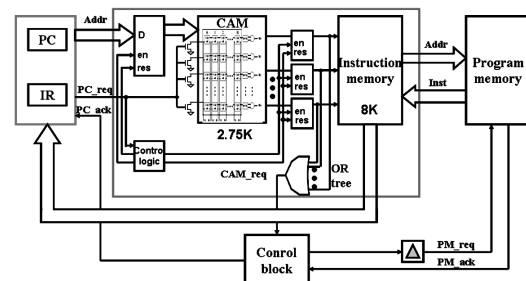

Fig. 1 Main architecture of the proposed asynchronous cache

Cache match state

First, an address in the Program Counter is loaded to the address bus. Then PC_req signal accesses the cache that loaded into CAM. If the address is in the CAM that is a match state, a completion signal is activated. Then PC_ack signal is enabled that it disables PC_req. The data in CAM memory is moved to the Instruction Register of CPU and the searching operation is finished. Timing diagram of the match state is shown in figure 2.
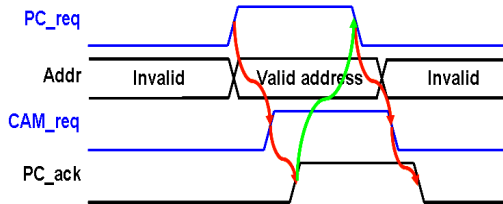

Fig. 2 Match state timing diagram of main architecture

Cache mismatch state

First, the address in the Program Counter is loaded to the address bus. Then PC_req signal accesses the cache that enables the input latch of CAM. Therefore valid address is loaded into CAM. If the search operation is missed to find the address in CAM, PM_req signal is activated after bundled delay Δt. The PM_req transfers the address to main memory. After finding data from main memory, PM_ack signal goes *high*. Thus the PC_ack loads an intstruction to the Instruction Register. Mismatch state timing diagram is shown in figure 3.
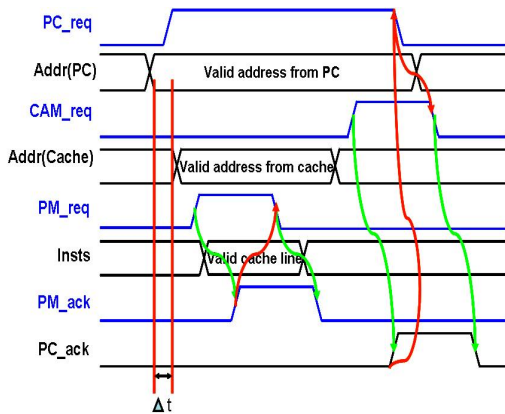

Fig. 3 Mismatch state timing diagram of main architecture

2.1 CAM design

Content addressable memory (CAM) or associative memory, is a storage device, which can be addressed by its own contents [6]. Each bit of CAM cells includes comparison logic. An input data of CAM is simultaneously compared with all the stored data [7]. Main architecture of CAM block is shown in figure 4.

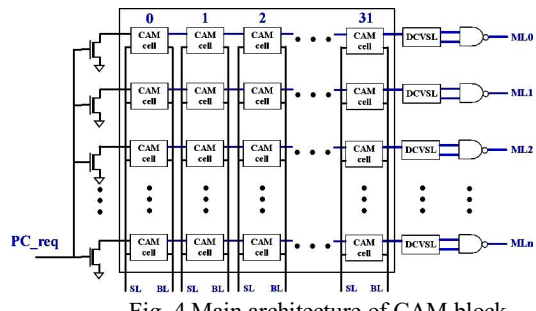enables the input latch of CAM. Therefore a valid address is

Fig. 4 Main architecture of CAM block

CAM cell has two basic functions: bit storage (as in RAM) and bit comparison (unique to CAM). Figure 5 and figure 6 show a conventional NAND-type CAM cell and a proposed NOR-type CAM cell, respectively. The bit storage in both cases is in an SRAM cell where cross-coupled inverters have data node **D** and $\overline{\mathbf{D}}$. In the figure 5 and figure 6, word line access NMOS transistors are used to read and write data in SRAM. Although some CAM cell implementations use lower area DRAM cells, typically, CAM cells use SRAM storage. The bit comparison, which is logically equivalent to an XOR of the stored bit and the search bit is implemented in a somewhat different fashion in the NOR and NAND cells [8].

Conventional NAND Cell

The NAND cell compares the stored bit, D, and corresponding search data on the corresponding search lines, ($\mathbf{SL}$, $\overline{\mathbf{SL}}$), using the tree transistors M1, $\mathbf{M_D}$, and $\mathbf{M_{\overline{D}}}$, which are all typically minimum-size to maintain high cell density. We illustrate the bit-comparison operation of a conventional NAND cell with an example.
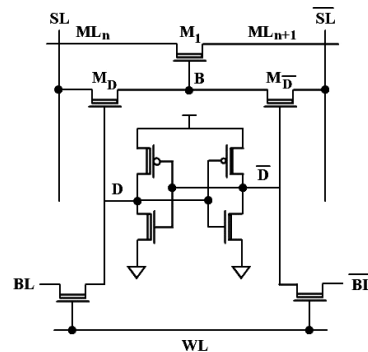

Fig. 5 Conventional NAND-type CAM cell

Consider the case of a match state when $\mathbf{SL}$ = 1 and $\mathbf{D}$ = 1. The pass transistor $\mathbf{M_D}$ is ON and passes the logic "1" on the $\mathbf{SL}$ to node B. Node **B** is the bit-match node, which is logic "1" if there is a match in the cell. The logic "1" on node **B** turns ON transistor M$_1$. Node that M1 is also turned ON in the

other match case when $SL = 0$ and $D = 0$. In this case, the transistor $M_{\bar{D}}$ passes logic high to raise node B. The remaining cases, where $SL \neq D$, result in a miss condition, and accordingly node B is logic "0" and the transistor M1 is OFF. Node B is a pass-transistor implementation of the XNOR function. The NAND nature of this cell becomes clear when multiple NAND cells are serially connected. In this case, the M_{Ln} and M_{Ln+1} nodes are joined to form a word. A serial NMOS chain of all the Mi transistors resembles the pull-down path of a CMOS NAND logic gate. A match condition for the entire word occurs only if every cell in a word is in the match condition. An important property of the NOR cell is that it provides a full rail voltage at the gates of all comparison transistors [9]. On the other hand, a deficiency of the NAND cell is that it provides only a reduced logic "1" voltage at node B, which can reach only V_{DD}–V_{th} when search lines are driven to V_{DD} (where V_{DD} is the supply voltage and V_{th} is the NMOS threshold voltage).

Proposed NOR Cell

The matched result of CAM is corresponding address of instruction memory. The proposed NOR cell implements the comparison between the complementary stored bit, D (and $\bar{D}$), and the complementary search data on the complementary search line, SL (and $\overline{SL}$), using four comparison transistors, M_1 through M_4, which are all typically minimum-size to maintain high cell density. These transistors implement the pull-down path of a dynamic XNOR logic gate with inputs SL and D. Each pair of transistors, M_1, M_3 and M_2, M_4 makes a pull-down path from the match line, ML, such that a mismatch of SL and D activates at least one of the pull-down paths, connecting ML to ground. A match of SL and D disables both pull-down paths, disconnecting ML from ground [10].
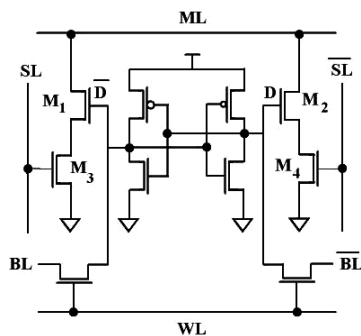

Fig. 6 Proposed NOR-type CAM cell

The NOR nature of this cell becomes clear when multiple cells are connected in parallel to form a CAM word by shorting the ML of each cell to the ML of adjacent cells. The pull-down paths connect in parallel resembling the pull-down path of a CMOS NOR logic gate. There is a match condition on a given ML only if every individual cell in the word has a match [11].

Match state timing diagram of CAM is shown in figure 7. When PC_req signal is enabled to high, following EN signal of latch is enabled. By disabling RES signal, valid address is taken into address latch. If the result of searching operation is matched, completion signal is activated. Else the result of searching operation is mismatched, completion signal is not activated.
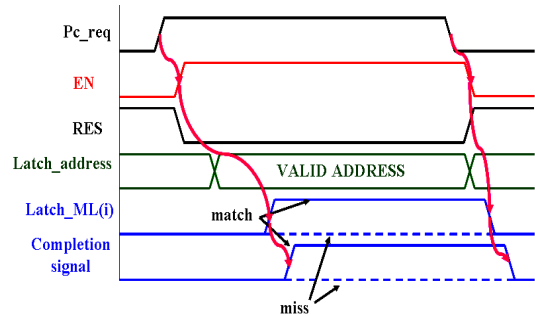

Fig. 7 Match state timing diagram of CAM

When PC_req signal is disabled, EN signal of latch is disabled. After that RES signal of latch is enabled and at this time address latch is refreshed by 0. In addition, completion signal is replaced by 0. Mismatch state timing diagram of CAM is shown in figure 8.
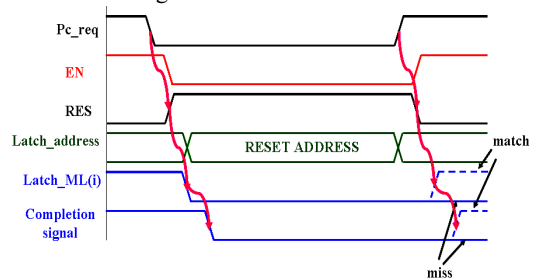

Fig. 8 Mismatch state timing diagram of CAM

2.2 Completion signal using DCVSL gate

Self-timed circuits consist of number of computation blocks coupled through interconnection blocks. The system employs a handshake protocol instead of the more classical clock signal. The combinational block generates a completion signal that indicates the combinational block is allowed to move a new operation. DCVSL is used to make a completion signal with a very simple extension of the basic circuitry. This led to self-timed designs with minimal overhead for completion circuitry. A generalized DCVSL gate is shown in figure 9. DCVSL gate has NMOS trees which implements the logic function with two output nodes Q and $\bar{Q}$.
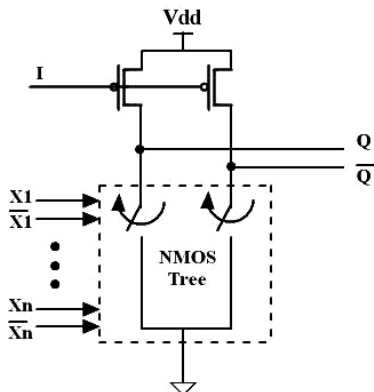
Fig. 9 DCVSL gate structure

For pre-charge phase during input **I** is low, nodes **Q** and $\overline{\textbf{Q}}$ are high. In the evaluate phase, a path of the NMOS trees will be conducting, causing one of the nodes **Q** and $\overline{\textbf{Q}}$ to be discharged. Therefore, result of evaluate phase is complementary values on the outputs. Completion signal required by the self-timed components is generated by making NAND gate of two outputs. Figure 10 shows a circuit for completion signal detecting match state. Table 1 shows truth table for CAM cell.
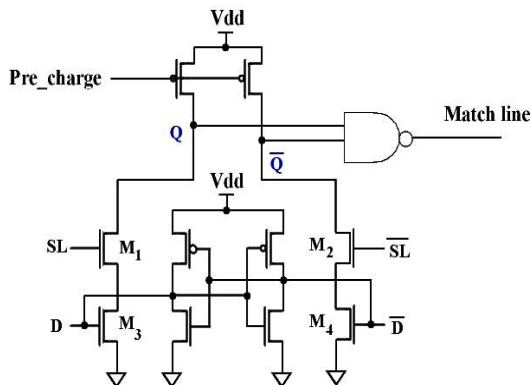

Fig. 10 Completion signal for matching

Table 1. CAM truth table

SL	D	ML	Content
0	0	1	Match
0	1	0	Mismatch
1	0	0	Mismatch
1	1	1	Match

3. SIMULATION RESULTS

The asynchronous cache is 32 bit × 2048 corresponding to 8Kbyte. CAM is decoding 2048 rows of the cache memory. Thus, the total size of CAM is, 2048 × 11bit, 2.75KB. LSB 11 bits is used to identify an address. Figure 11 shows CAM is handling the instruction memory.
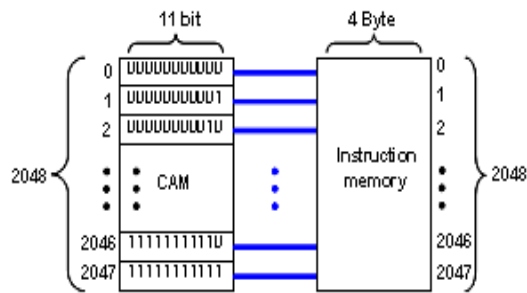

Fig. 11 CAM size

We implemented CAM block in VHDL coding. A CAM cell described the latch for bit-line, the latch for search-line and XNOR gate for comparing bit-line and search-line. If all data of search-lines and bit-lines are equal, that will be matched. All CAM cell's match-lines connect the AND gates. After that outputs of AND gates connected to DCVSL gates. Finally, outputs of DCVSL gates connected to OR gate, that will be generate completion signal. Figure 12 shows the RTL architecture of CAM. Figure 13 shows VHDL coding simulation results of the asynchronous CAM.
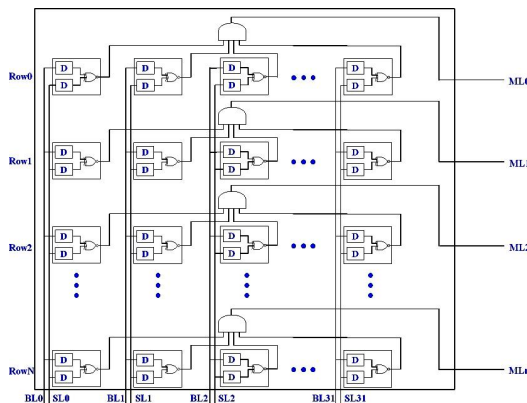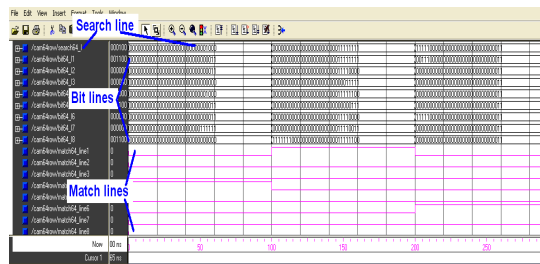

Fig. 12 The RTL architecture of CAM

Fig. 13 VHDL coding simulation result of the asynchronous CAM.

All input vector are tested. In matched state as shown in figure 14, **SL** and **D** are ON, then **Q** line connect to ground. That time $\overline{\textbf{Q}}$ keeps high state, then output of NAND gate goes to high. Figure 14 shows simulation result of CAM cell in HSPICE tool that is supported by IDEC. Figure 15 shows simulated circuit CAM with DCVSL.
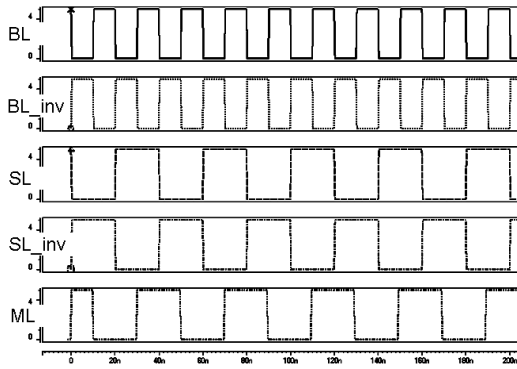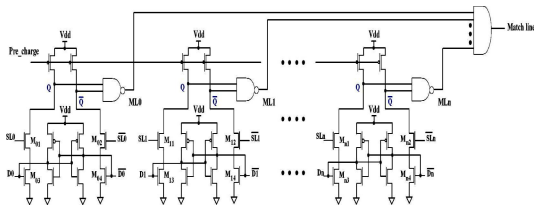
Fig. 14 Simulation result of CAM cell

Fig. 15 Simulated circuit CAM with DCVSL

Designed cache was tested using Michigan (MI) bench program as shown in table 2.

Table 2. Hit ratio of the designed cache using MI bench program

MI Bench Program	Hit	Miss	Hit ratio
BITC	723991	89	99%
STRS	186040	82	99%
QSTR	147966	391	99%
DHRY	1150981	268	99%
DIJK	5324518	200	99%

4. CONCLUSIONS

This paper proposed a CAM based on DI model for asynchronous cache. We get a completion signal of CAM by using DCVSL that acts as a reference signal in the system. We designed and simulated a cache with 2.75KB CAM for 8KB instruction memory. The results show that the cache hit ratio is up to 95% based on pseudo-least recently used (LRU) replacement policy.

5. ACKNOWLEDGMENT

6. REFERENCES

[1] Jih-Kwon Peir, W.W. Hsu, A.J.Smith, "Functional implementation techniques for CPU cache memories," *IEEE Computers, Vol.48*, Feb. 1999, pp. 100-110.

[2] J. Tuominen, T. Santti, J. Plosila, "Comparative Study of Synthesis for Asynchronous and Synchronous Cache Controllers," *IEEE Norchip conference*, March. 2006, pp. 11-14.

[3] S.S. Guillory, D.G. Saab, A.Yang, "Fault modeling and testing of self-timed circuits," *IEEE Chip-to-System Test Concerns for the 90's*, April. 1991, pp. 62-66.

[4] Jin-Fu Li, "Testing Ternary Content Addressable Memories With Comparison Faults Using March-Like Tests," *IEEE Computer-Aided design of integrated circuits and systems, Vol.26, No. 5*, May. 2007, pp. 919-931.

[5] Kuo-Hsing Cheng, Chia-Hung Wei, Shu-Yu Jiang, " Static divided word matching line for low-power content addressable memory design," *IEEE Circuits and Systems ISGAS, Vol.2*, May. 2004, pp. 629-632.

[6] Jien-Chung Lo, " Fault-tolerant content addressable memory," *IEEE Proc. ICCD*, Oct. 1993, pp.193-196.

[7] R.E. Aly, B.R. Nallamilli, M.A. Bayoumi, " Variable-way set associative cache design for embedded system applications," *Circuits and Systems MWSCAS, Vol.3*, Dec. 2003, pp.1435-1438.

[8] V. Chaudhary, T.-H. Chen, F. Sheerin, L.T. Clark, " Critical race-free low-power nand match line content addressable memory tagged cache memory," *Computers and Digital Techniques, IET, Vol.2*, Jan. 2008, pp.40-44.

[9] Kuo-Hsing Cheng, Chia-Hung Wei, Shu-Yu Jiang, " Static divided word matching line for low-power content addressable memory design," *Circuits and Systems ISCAS*, Vol.2, May. 2004, pp.929-632.

[10] J.G. Delgado-Frias, A. Yu, J. Nyathi, "A dynamic content addressable memory using a 4-transistor cell," *Design of Mixed-Mode Integrated Circuits and Applications, Vol.2*, Aug. 1999, pp.110-113.

[11] T. Kumaki, Y. Kouno, M. Ishizaki, T. Koide, H.J Mattausch, "Application of Multi-ported CAM for Parallel Coding," *Circuits and Systems APCCAS, Vol.3*, Dec. 2006, pp.1859-1862.

Jigjidsuren Battogtokh
received the B.S. and M.S. degrees Electronics Engineer-ing from information techno-logy school of Mongolian National University, Ulaan-baatar, Mongolia in 2005 and 2007, respectively. He is currently pursuing the M.S. degree at the graduate school of the Chungbuk National University, Korea. His research interests are microprocessor and VLSI design.
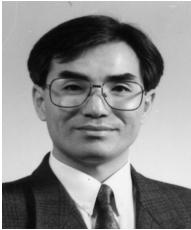
Kyoung-Rok Cho
received the B.S. degree in Electronic Engineering from Kyoung-pook National University, Taegu, Korea in 1977, and M.S. and Ph.D. degrees in Electrical Engineering from the University of Tokyo, Tokyo, Japan, in 1989 and 1992, respectively. From 1979 to 1986, he was with TV research center of Gold Star Company in Korea. He is currently a Professor in Dept. of Computer and Communication Eng. of Chungbuk National University, Korea, since August 1992. His research interests are in the field of high-speed and low-power circuit design, and ASIC design for communication system. From 1999 to 2000, he was a visiting scholar at Oregon State University, OR. He is a member of Institute of Electrical and Electronics Engineer (IEEE), Institute of Electronics, Information and Communication Engineers (IEICE), and Korea Institute Tele-communication Electronics (KITE).