

# A proposal of MFMCAM and its applications

Takeshi KUMAKI, Keisuke IWAI, and Takakazu KUROKAWA

Department of Computer Science,

National Defense Academy

1-10-20 Hashirimizu, Yokosuka-shi, 239-8686 Japan

Tel. +81-468-41-3810 (ext. 2219), Fax. +81-468-44-5911

e-mail: kukky@ma3.justnet.ne.jp

## Abstract

This paper proposes MFMCAM(Multi-Functional Multi-port CAM) which has several ports with same characteristic function. This device can process data faster than the conventional single port CAM. MFMCAM is superior to CAMs formed in parallel on the stand points of frequency and module resources. Two representative applications of MFMCAM, sorter and router, are also presented.

## 1. Introduction

CAM(Content addressable Memory) is a kind of functional memory, having a data table called "contents table" in itself. Data to be compared is stored in contents table. This device is able to compare an input data with all of their stored data in contents table simultaneously. If the input data matches with one of the stored data, CAM outputs the according address in which the matched data is stored. Since CAM can search data in contents table very fast, there are several practical applications of CAM such as router, filter, data compression, dictionary search and so on.

However there are several technical problems for this device. As the first problem, conventional CAM has the fixed number of processing clock cycles. For example, a fully-parallel(or word-parallel, bit-parallel) CAM needs only single-clock-cycle [1] - [3]. These architectures represent the best solution at present. However there will be some limits for the improvements of processing time by CAM in case of multiple data inputs at the same time. As the second problem, the cost of CAM is very high. Generally its cost is several times higher than RAM. [4], [5]. The problem that conventional CAM can only process "exact" match will be the next problem. There are several CAM architectures to be able to accord with "approximate" match [6] - [9]. However these CAMs have little ability for flexible or fixed "approximate" match.

In order to overcome these problems, we propose MFMCAM (Multi-Functional Multi-port CAM) and its effective applications.

## 2. Architecture of MFMCAM

### 2.1 Basic architecture

Fig. 1 shows a general model of MFMCAM. To decrease memory capacity, CAM with multiple inputs was proposed in [10]. However this CAM generates only one matched signal. Here, we will present MFMCAM which has N input ports as well as N output ports. An input port can receive m-bit input data and m-bit masking data. While an output port generates a hit bit and n-bit hit address. These N ports have one common contents table, and MFMCAM can process N data at the same time. Thus the processing time will be N times faster than the conventional single port CAM.

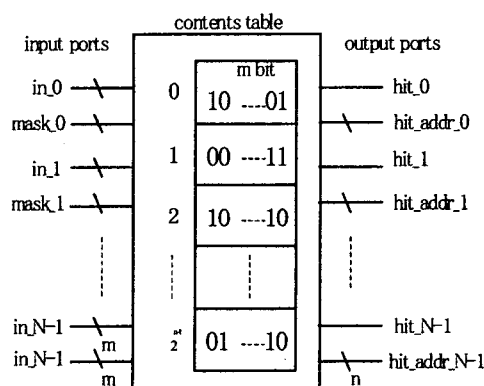


Fig.1. m-bit x 2<sup>n-1</sup>-word MFMCAM.

### 2.2 Characteristic functions

Since MFMCAM has multiple ports and multiple functions, it has many possibilities using its flexible searching ability to be applied for many applications. The representative characteristic functions of MFMCAM are shown as follows:

#### (i) Flexible number of ports

MFMCAM can change the number of ports flexibly by using FPGA. Before applying MFMCAM for a

specific application, it can be reconfigured and the number of ports as well as its function can be adjusted to the application. We can also choose a suitable device for each application.

(ii) Independent operation for each port

This device can handle different masking patterns for each port.

(iii) Comparing data according to hamming distance

By giving different hamming distance to each port before the beginning of data search, MFMCAM will be able to handle "approximate" match. In case that a hamming distance "2" is given to MFMCAM, and an input data is 00110, it will search such data as "10100", "00000", "01111", and so on. This kind of function enable MFMCAM to process "exact" match and also "approximate" match at the same time.

### 3. Design of MFMCAM

#### 3.1 Overall Block Diagram

MFMCAM with 16-bit data width  $\times$  1,024 words (10-bit address) is developed on FPGA (Xilinx Virtex II XC2V6000) using XST verilog HDL. We added multiple ports and several functions to the conventional CAM presented at Xilinx application note XAPP202. This conventional device has hierarchical structure consisting of the following four different types of modules; mlt\_module, mlt\_entry, mlt\_byte and lut\_ram. As the initial values in contents table of MFMCAM, we used Block RAM(16-bit  $\times$  1,024words) in Virtex chip named "mlt\_block\_ram". The block diagram of MFMCAM is shown in Fig. 2.

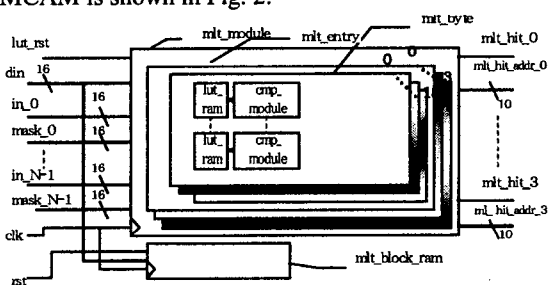


Fig. 2 Overall block diagram of MFMCAM.

#### 3.1 Mlt\_byte module

This module is the basic module of MFMCAM. One mlt\_byte module consists of 16 memory modules named "lut\_rams" and N comparison modules named "comp\_modules". A block diagram of mlt\_byte module is shown in Fig. 3.

The lut\_ram can store 1b  $\times$  16words data as a part

contents table. This module is made by Distributed Ram in Virtex.

The comp\_module includes a "func\_circuit" and a "comp\_circuit". A "func\_circuit" processes data with several kinds of functions which are described in 2.2. While a "comp\_circuit" includes a comparator that receives the output of the adjustment "func\_circuit" and generates a hit signal.

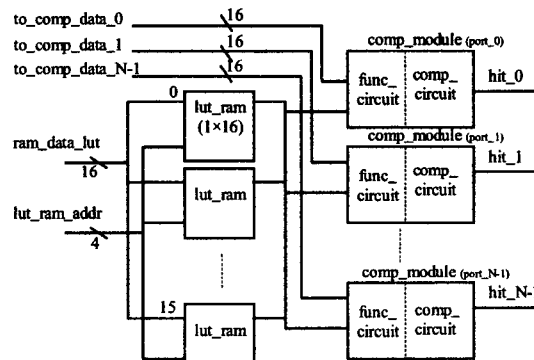


Fig. 3 Block diagram of mlt\_byte module.

### 4. Implementation of MFMCAM

#### 4.1 Device and tool for implementation

We implemented a MFMCAM using a FPGA chip (Xilinx Virtex II XC2V6000). Xilinx ISE 4.1i and XST verilog HDL are used for this implementation. Implementation results are summarized in Table. 1.

#### 4.2 Comparison with parallel CAMs

Generally multiple input data can be processed using several CAMs arranged in parallel [11]. Here we will call this kind of architecture as a "parallel CAM". We have implemented parallel CAMs for the evaluation of MFMCAM. Its implementation results are also shown in Table. 1. As the number of ports increases, the number of slices for parallel CAMs increased by 1,700, while its clock frequency decreased. On the other hand, the number of slices for MFMCAM increased one half of that of CAMs, and its clock frequency did not decrease so much. We can see the superiority of MFMCAM to parallel CAMs on the stand points of the number of slices and clock frequency

Table. 1 Implementation results of MFMCAM and parallel CAMs.

	Number of prts	1	2	3	4
MFMCAM	Number of slices	1,794	2,613	3,347	4,061
	Maximum frequency	62.158	40.933	48.056	56.363
Parallel CAMs	Number of slices	1,794	3,566	5,147	6,850
	Maximum frequency	62.158	54.690	49.312	33.835

### 4.3 Performance evaluation

In order to clarify the performance of MFMCAM, we compared the implementation results of MFMCAM with several conventional CAMs using FPGA or ASIC. These competitors are two types of XAPP CAM designed by XILINX[12], [13], embedded CAM by ALTERA APEX[14] and LANCAM-B by MUSIC SEMICONDUCTORS [15]. Comparison results are summarized in Table. 2.

**Table 2 Performance of CAMs**

Design	Number of bits	Number of depth	Latency (ns)		Frequency (MHz)	Number of ports	Processing ratio
			compare	encoder			
XAPP202	16	32	8	8	115	1	0.16
XAPP203	16	4096	16 × 20	20	50	1	0.99
APEX CAM	48	256	9.1	9.1	110	1	0.26
LANCAMB	16	1024	90		-	1	0.93
MFMCAM	16	1024	16 × 17.7	53.2	56	4	1

Here processing ratio is calculated by the following equation;

$$\left(\frac{1}{\text{Latency}}\right) \times (\text{ports}) \times \left(\frac{\text{Depth}}{1,024}\right)$$

MFMCAM is superior to the other existing CAMs concerning to the data processing speed.

## 5. Applications

### 5.1 Overview

We will propose several applications using MFMCAM in section. Since MFMCAM can process multiple inputs and flexible search, these applications will make the best use of characteristics of MFMCAM.

### 5.2 MFMCAM for Switching or routing

#### 5.2.1 Background

Recently networks require the great progress of their speed up for their communication to send huge amount of IP packets. The process of classifying IP packets depending on information in the header part is a characteristic process of communication using IP packets. Switch or router is the typical equipment to process this classification at present. These equipments are using software or cache for this classification. However, their processing speed is becoming tight according to the wide spread of Internet technology. Therefore CAM has been used for equipments to process IP headers [16].

#### 5.2.2 Applying of MFMCAM

We propose a new architecture of IP routing using MFMCAM. Its example is shown in Fig. 4.

Usually when IP packets arrive at a switch or a router, header parts are sent to the search engine and the other parts are stored in queuing memory. MFMCAM transforms the address information of the incoming header by comparing with its contents table. If this comparison matches with any data in contents table, MFMCAM outputs the according address to RAM. Then RAM generates the destination address for the next header by receiving a matched address. In case of using MFMCAM, this device can process multiple headers at the same time. Thus a high throughput of transaction can be realized using only one MFMCAM.

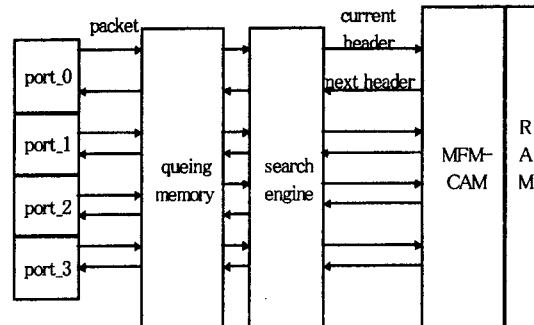


Fig. 4: IP header switching using MFMCAM

### 5.3 MFMCAM for sorting

Here we propose an application of MFMCAM to sorting data in contents table. Before the beginning sorting process, an approximate number and the bit position are decided at each port. An example of sorting processing using MFMCAM is shown in Table. 3 and Fig. 5.

Table. 3: Approximate number and bit position.

ports	Approximate number	bit position
0	0	MSB x_x_x_x_x LSB
1	1	MSB x_x_x_x_o LSB
2	1	MSB x_x_x_o_x LSB
3	2	MSB x_x_x_o_o LSB

o: assign bit

At first, a start bit pattern "00000" is inputted to MFMCAM. Since each port can search data which has different hamming distance from the input data according to the dedicated pattern shown in Table. 3. In case of this example, the matching results of four ports are presented as follows;

$$(\text{port0}, \text{port1}, \text{port2}, \text{port3}) = (1, 6, -, 8)$$

After that, the inputs to MFMCAM increase by 4 as; 00100, 01000, 01100.... The output of MFMCAM as the sorting process is shown in Table. 4. Generally if the contents table consists of data with m bit width, the maximum number of steps is expressed as  $2^m / (\text{ports})$ .

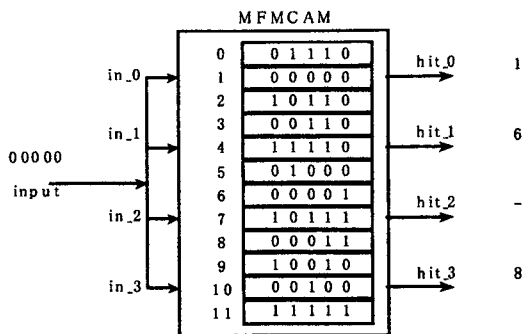


Fig. 5 Example of sorting process using MFM CAM.

Table. 4 Output of MFM CAM as sorting.

	00000	00100	01000	01100	10000
0	01110	01110	01110	01110	prt2 hit②
1	00000	prt0 hit①	00000	00000	00000
2	10110	10110	10110	10110	10110
3	00110	00110	prt2 hit⑤	00110	00110
4	11110	11110	11110	11110	11110
5	01000	01000	01000	prt0 hit④	01000
6	00001	prt1 hit②	00001	00001	00001
7	10111	10111	10111	10111	10111
8	00011	prt3 hit③	00011	00011	00011
9	10010	10010	10010	10010	10010
10	00100	00100	prt0 hit④	00100	00100
11	11111	11111	11111	11111	11111

	10100	11000	11100
01110	01110	01110	01110
00000	00000	00000	00000
10110	prt2 hit⑥	10110	10110
00110	00110	00110	00110
11110	11110	11110	prt2 hit④
01000	01000	01000	01000
00001	00001	00001	00001
10111	prt3 hit④	10111	10111
00011	00011	00011	00011
prt2 hit⑥	10010	10010	10010
00100	00100	00100	00100
11111	11111	11111	prt3 hit②

## 6. Conclusion

In this paper, we presented an architecture of MFM CAM and its applications. Usually some applications have to use several conventional CAMs and several logic gates for flexible search. However there exist several applications such as switching and sorting for which MFM CAM can be applied. Furthermore these applications can be realized with less resources using MFM CAM.

Since MFM CAM can process multiple input data arriving at the same time. If these input data arrives at different timing, there will be several data to be waited for the next search. Now we are improving the architecture of MFM CAM for asynchronous search. Improvements of the presented architecture of MFM CAM and its applications are still rested for the future.

## References

- [1] F. Shafai, K. J. Schultz, G. F. Randall Gibson, A. G. Bluschke and D. E. Somppi, "Fully Parallel 30-MHz, 2.5-Mb CAM," IEEE J. Solid-State Circuits, vol. 33, no. 11, pp. 1690-1696, Nov. 1998.
- [2] K. J. Schultz and P. G. Gulak, "Fully Parallel Integrated CAM/RAM Using Preclassification to Enable Large Capacities," IEEE J. Solid-State Circuits, vol. 31, no. 5, pp. 689-699, May. 1996.
- [3] T. Yamagata, M. Mihara, T. Hamamoto, Y. Murai, T. Kobayashi, M. Yamada and H. Ozaki, "A 288-kb Fully Parallel Content Addressable Memory Using a Stacked-Capacitor Cell Structure," IEEE J. Solid-State Circuits, vol. 27, no. 12, pp. 1927-1933, Dec. 1992.
- [4] <http://www.music-ic.com/>
- [5] <http://www.k-micro.com/>
- [6] H. Yamada, M. Hirata, H. Nagai and K. Takahashi, "A High-Speed String-Search Engine," IEEE J. Solid-State Circuits, vol. SC-22, no. 5, pp. 828-834, Oct. 1987.
- [7] M. Hirata, H. Yamada, H. Nagai and K. Takahashi, "A Versatile Data String-Search VLSI," IEEE J. Solid-State Circuits, vol. 23, no. 2, pp. 329-334, Apr. 1988.
- [8] M. Motomura, J. Toyoura, K. Harata, H. ooka, H. Yamada and T. Enomoto, "A 1.2-Million Transistor, 33-MHz, 20-b Dictionary Search Processor(DISP) ULSI with a 160-kb CAM," IEEE J. Solid-State Circuits, vol. 25, no. 5, pp. 828-834, Oct. 1990.
- [9] Y. Oike, M. Ikeda and K. Asada, "High-Speed Content -Addressable memory Using Synchronous Haming Distance Search Circuits," IEICE Technical Report, ICD2002-4, pp. 19-24, Apr. 2002.
- [10] M. Hariyama and M. Kameyama, "Collision Detection VLSI Processor for Highly - Safe Intelligent Vehicles Using a Multiport Content - Addressable Memory," Interdisciplinary Information Sciences, vol. 5, no. 2, pp. 109 - 116, 1999.
- [11] <http://cco-sj-2.cisco.com/japanese/warp/public>
- [12] M. Defossez, "Content Addressable Memory (CAM) in ATM Applications," Xilinx Application Note XAPP 202, ver. 1.2, Feb.2000.
- [13] J. L. Brelet and B. New, "Designing Flexible, Fast CAMs with Virtex Slices," Xilinx Application Note XAPP 203, ver. 1.1, Sep.1999.
- [14] "CAM Comparison: APEX 20KE vs. Virtex-E Devices," Technical Brief 61, ver. 1, Dec 1999.
- [15] "LANCAM B Family," MUSIC Semiconductors Data Sheet Draft, Rev. 1, Oct. 2000.
- [16] H. Shinoura and K. Okada, "Information Technology Illustrated Broadband Textbook," IE institute, pp.178-213, Jul 2001.