Yield Enhancement Techniques for Content-Addressable Memories

Shyue-Kung Lu, Guan-Quan Lin, and **Sy-Yen Kuo*
Dep. of Electronic Engineering, Fu Jen Catholic University, Taipei, Taiwan
sklu@ee.fju.edu.tw
*Dep. of Electrical Engineering, National Taiwan University, Taipei, Taiwan

Abstract—In this paper, *block-level* reconfiguration techniques are proposed for content-addressable memories. The CAM words and redundant words are divided into *blocks* and reconfiguration is performed at the block level instead of the conventional word level. The reconfiguration mechanism used to implement our technique requires negligible hardware overhead. According to simulation results, the hardware overhead is 3.92% for a 1-Mb CAM array. It also shows that our approach can improve fabrication yield significantly.

1. Introduction

With the trend of nanometer technology, the size of a CAM module can reach over 18 MBits [1, 2]. Due to the low accessibility of embedded CAM modules, it is difficult to test them with external testers. A promising solution is by using built-in self-test techniques for CAMs [3, 4, 5]. However, as the density and capacity of CAMs increase, it is more prone to suffer from defects. Therefore, the fabrication yield will be very low. It is inevitable to seek for efficient fault-tolerant techniques for improving the yield of CAMs.

For the traditional redundancy mechanisms, an entire redundant row (column) is used to replace the faulty row (column). Since the number of CAM cells of a row (column) in the CAM array is very large. It is not efficient to replace a faulty row (column) with an entire redundant (row) column. Therefore, in order to achieve sufficient fabrication yield, more redundant rows (columns) are required to be added into the CAM array. This will in turn increase the fabrication cost.

In order to alleviate these drawbacks, novel redundant mechanisms are proposed in this paper. Instead of replacing faulty cells with an entire CAM word, the *block-based* approach is proposed. Each row of the CAM array (including the word line and the match line) is divided into *row blocks*. We use only redundant rows as the redundancies and they are also divided into row blocks. Therefore, the replacement can be performed at the row block level instead of the conventional row level. Redundant columns are not added here for simplicity. According to simulation results, the hardware overhead is 3.92% for an experimental 1-Mb CAM array. The yield improvements over conventional *word-based* and *bank-based* techniques are significant.

The rest of this paper is organized as follows. In the next section, the basic functions of CAMs are reviewed. In Section 3, the scenario of the block-based fault-tolerant architecture is proposed. Some preliminary simulation results are also shown in this section. Finally, conclusions are given is Section 4.

2. Architecture of CAMs

The basic architecture of a content-addressable memory is shown in Fig. 1 [6]. It mainly consists of the CAM array, the mask register, the priority encoder, and the row decoder. In this example, the CAM array contains 4 rows and 6 columns. Each CAM cell is represented by the box marked *C*. During the search operation, the search data is sent to the input register and then forwarded to each row entry in the CAM array via the search lines (SL_0-SL_5). The search data is compared with the stored word in each row entry. The results of the comparisons determine the values of the match lines (ML_0-ML_3) and these values are sent to the priority encoder. The outputs of the encoder include the *hit* signal and the address of the matched row entry (*Match_address*). The content of the mask register is used to exclude the input data bits from comparing with the stored words.

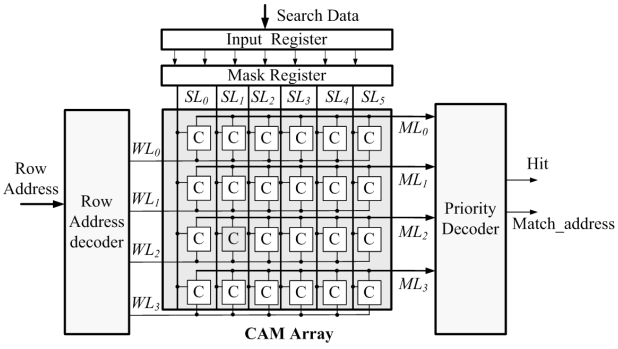

Fig. 1: The block diagram of a CAM device.

3. Architecture of BISR for CAMs

The basic concept of the proposed *block-based* replacement technique can be illustrated as shown in Fig. 2. In this example, the CAM array contains 4 words (W_0-W_3) and 16 columns. Two spare words (SR_0, SR_1) are added into this CAM array. They are placed at the top and the bottom of this array, respectively. Each word (including the spare rows) is divided into *row blocks* and each contains four CAM cells in this example. The row blocks of the same columns form a row bank. Therefore, there are four row banks (RB_0-RB_3) in Fig. 2 and each row bank contains 4 row blocks. A faulty row block is a block which contains faulty CAM cells. The CAM array in Fig. 2 contains 5 faulty row blocks (the dashed boxes). The spare words are also divided into eight row blocks which can be used to repair faulty row blocks.

As indicated by the arrows, faulty row blocks are replaced by their corresponding spare row blocks. Therefore, two spare words (5 spare row blocks) are sufficient to repair the faulty CAM array successfully with our block-based replacement technique. However, if an entire row is used as the basic replacement element; four spare words are required to repair the CAM array. It is evident that our approach will suffer from less hardware overhead and the cost to achieve the specified yield or reliability level is reduced significantly. To perform the block-based reconfiguration, a *switching logic* should be integrated with each row block (including the redundancy) to steer the word lines and match lines. The switching logic can switch off the faulty row block and replace it with its upper or lower neighboring block. Therefore, the replacement can be performed at the block level instead of the traditional row level. The fabrication yield can be improved significantly due to the efficient usage of the redundancy.

The entire BISR (built-in self-repair) architecture is shown in Fig. 3. Besides of the CAM array (including the redundancy), the BIRA (built-in redundancy analysis) module, the BIST controller, and a CAM wrapper are added. The BIST module is used to perform the adopted March algorithm for testing content-addressable memories.

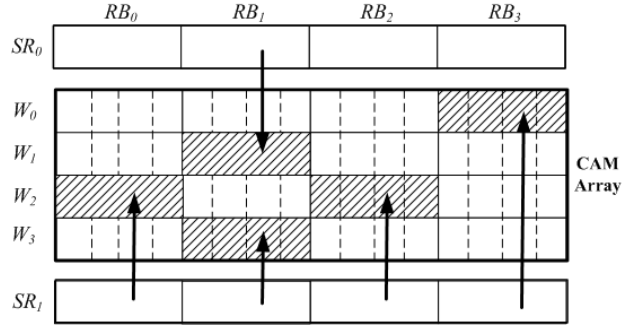

Fig. 2: The basic concept of the block-based replacement technique.

The CAM wrapper is used to switch between normal mode and the BIST mode. If a faulty row block is detected, the BIRA module is used for redundancy analysis. Control signals are generated to control the switch logic of each row block. According to simulation results, the hardware overhead is 3.92% for an experimental 1-Mb CAM array. Moreover, the fabrication yield can be improved significantly.

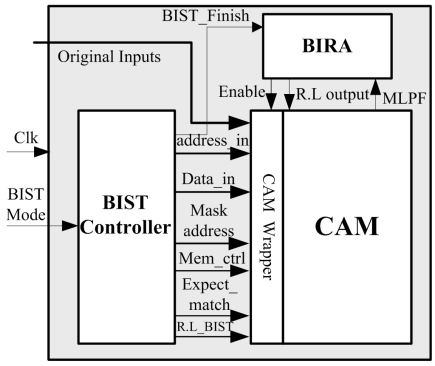

Fig. 3: Architecture of the proposed BISR techniques.

4. Conclusions

In this paper, block-based BISR techniques are proposed for content-addressable memories. According to simulation results, the hardware overhead is 3.92% for a 1-Mb CAM array. Moreover, the fabrication yield can also be improved significantly.

Reference

[1] G. Kasai,Y. Takarabe, K. Furumi, and M. Yoneda, "200 MHz/200 MSPS 3.2 W at 1.5 V Vdd, 9.4 Mbits ternary CAM with new charge injection match detect circuits and bank selection scheme," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, 2003, pp. 387–390.

[2] A. Roth, etc., "Advanced ternary CAM circuits on 0.13 _m logic process technology," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, 2004, pp. 465–468.

[3] K. J. Lin and C. W. Wu, "Testing Content-Addressable Memories Using Functional Models and March-Like Algorithm," *IEEE Trans. Computer-Aided Des. of Integrated Circuits and Systems*, vol. 19, no. 5, May 2000. pp. 577-588.

[4] R. Nadkarni, I. Arsovski, R. Wistort, and V. Chickanosky, " Improved Match-Line Test and Repair Methodology Including Power-Supply Noise Testing for Content-Addressable Memories," in *Proc. Int'l Test Conference,* Oct. 2006, pp. 1-9.

[5] Y. S. Kang, J. C. Lee, and S. Kang, "Parallel BIST architecture for CAMs," *Electronics Letters,* vol. 33, no. 1, pp. 30–31, Jan. 1997.

[6] K. Pagiamtzis and A. Sheikholeslami*,"* Content-Addressable Memory (CAM) Circuits and Architectures: A Tutorial and Survey," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 3, March 2006, pp. 712-717.