2015 International Symposium on Nonlinear Theory and its Applications
NOLTA2015, Kowloon, Hong Kong, China, December 1-4, 2015

Cellular Hardware Platform:CAM^2 on FPGA

Yosuke Hioki† Takeshi Kumaki† Tomohiro Fujita†, and Takeshi Ogura†

†Ritsumeikan University, 1-1-1 Noji-Higashi, Kusatsu, Shiga, 525-8577 Japan
Email: {ri0013hv@ed, kumaki@fc, tfujita@se, togura@se}.ritsumei.ac.jp

Abstract—In this work, cellular automata on content addressable memory (CAM^2) is implemented on FPGA, which is highly parallel two-dimensional cellular automata architecture. It can achieve various cellular automata (CA) processing such as pixel level snakes, morphological wavelet transform and pattern spectrum. These applications are currently offered on ASIC. CAM^2 on ASIC cannot change its capacity and configuration, while in contrast, CAM^2 on FPGA is flexible in terms of capacity and configuration, can perform optimum processing, and is low cost. As a result, max frequency and capacity of CAM^2 on FPGA is respectively fourth and one eighth that of, CAM^2 on ASIC. Capacity frequency product realizes one twice as much as CAM^2 on ASIC. CAM^2 on FPGA is thus poised to make a significant contribution to the development of compact, high-performance systems.

1. Introduction

CAM^2 is a highly parallel two-dimensional cellular automata architecture [1], [2], [3] that can realize various CA-based processing such as pixel level snakes [4], morphological wavelet transform [5], [6], and pattern spectrum [7], [8]. However, CAM^2 on ASIC has serious issues; namely, it capacity and configuration cannot be changed, and the cost of developing a new CAM^2 has become huge. In contrast, FPGA has continued to advance. If CAM^2 is implemented on FPGA, it becomes an accelerator of various CA-based processes. Moreover, CAM^2 on FPGA can change its capacity and configuration, and it keeps expenses below that of ASIC. Therefore, CAM^2 on FPGA is a very appealing platform.

This paper is organized as follows. First, we introduce cellular hardware platforms: CAM^2 and give an example of its application on ASIC. In sect. 3, CAM^2 on FPGA is described and the differences between CAM^2 on FPGA and CAM^2 on ASIC are discussed. In sect. 4, implementation reslts of CAM^2 on FPGA is shown. We conclude with a brief summary in sect. 5.

2. Cellular Hardware Platform

2.1. CAM^2

CAM^2 is being developed for several digital appliances and is composed of content addressable memory (CAM), that makes it possible to embed within it a large number of processing element (PEs) corresponding to Cellular Automata (CA) cells. Table .1 list the specifications of the CAM^2 processor and Fig. 3 is photograph of CAM^2 on FPGA. The CAM^2 is a compact, high-performance, flexible, and highly parallel 2-D cellular automata. In light of the above, real-time morphological pattern spectrum processing can be realized with the CAM^2 evaluation system. The CAM^2 evaluation system and a block diagram of CAM^2 are shown in Fig. 1 and the CAM cell of CAM^2 on ASIC is shown in Fig. 2. The CAM^2 evaluation system consists of a CAM^2 processor, a FPGA that controls the evaluation board, various memories, a monitor, and a host PC. The most prominent features of the configuration are the dedicated CAM blocks, included in the CAM^2 processor for the highly parallel PE array. Each CAM block performs various types of parallel data processing for CA in each word. Moreover, the CAM^2 processor can be easily controlled by command sequences generated from the FPGA. Since the FPGA can easily rewrite the logic, CAM^2 can perform various types of CA-based image processing either alone or in combination with another system.

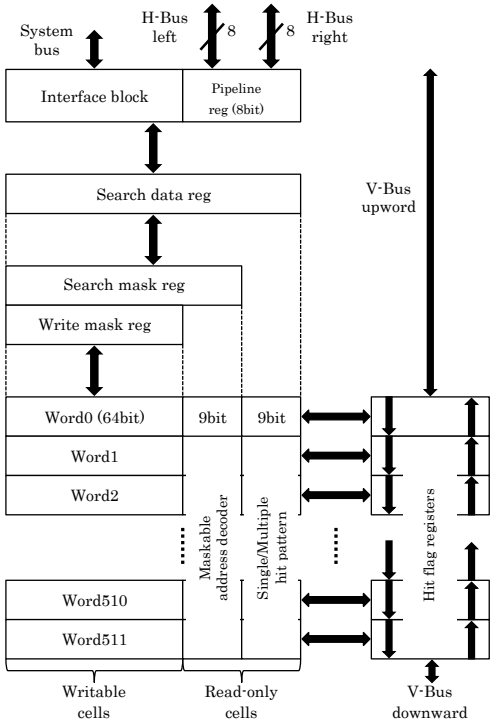

Figure 1: CAM^2 on ASIC configuration

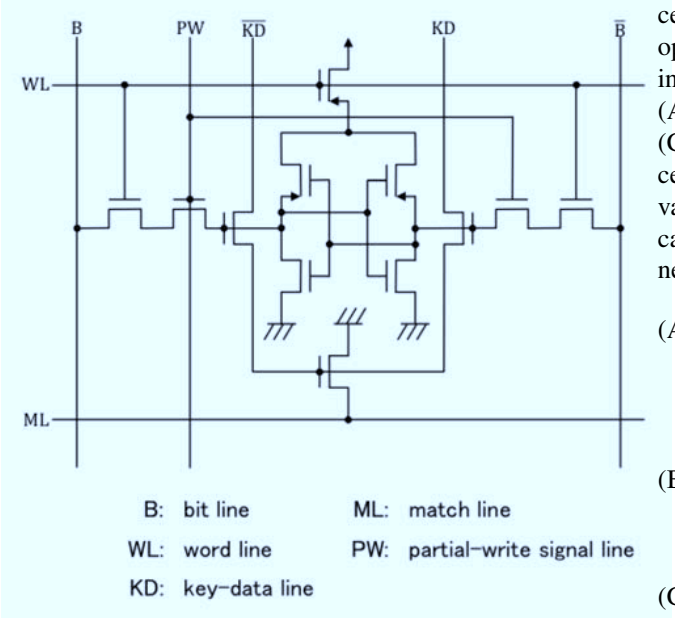

Figure 2: CAM^2 cell on ASIC

Table 1: specification of CAM^2 processor

Configuration	512 words × 64 bits × 32 blocks
Instruction set	32
Operating frequency	40 MHz
LSI process technology	0.25 μm CMOS double Al layers

Figure 3: CAM^2 on ASIC

While the CAM^2 processor is suitable for CA processing, it also enables effective parallel bit-serial cipher processing that consists of logic operations and table-lookup coding. CA and cipher processing with the CAM^2 processor are carried out by iterative data transfer and update operations. To perform the above processing, the following functions are absolutely essential: maskable OR search (A), maskable parallel write (B), and hit-flag shift up down (C). In the case of CA processing, the value of the original cell is transferred to its nearest neighboring cell. In the CA-value update process, the next value of the original cell is calculated by a transition rule using the original and nearest neighboring cell values.

(A) Maskable OR search
Search data from input are compared with word contents. The maskable search results are accumulated in hit-flag registers through OR logic.

(B) Maskable partial-parallel write
Data are written into specific bit positions of multiple word locations.

(C) Hit-flag shift up and down
The hit-flag registers are shifted to upper or lower words.

Through the iteration of these functions, the CA-value transfer and update processes can be carried out in a bit-serial, word-parallel manner. The processing drawbacks are that it takes a long time for such complex operations as the multiplication of longer bits. Moreover, the processing time for the CA-value transfer is proportional to the transfer bit length. Thus, a low-bit CA-value update is required to shorten the processing time.

2.2. Application of CAM^2

(A) Cellular neural network [9],[10]
In a cellular neural/nonlinear network (CNN), information processing is done in parallel through network dynamics, which means CNN has the potential to realize a highly parallel real-time information processing system. To demonstrate the flexibility and programmability of CAM^2, nonlinear template CNN was implemented.

(B) Pixel level snakes [4]
The snakes algorithm was originally formularized as a variational method. Using the Euler-Lagrange method, it can be transformed into a partial differential equation that can be related to cellular network models. In PLS, the authors developed this model to connect with CNN more easily, and succeeded in leading the snakes algorithm to pixel level cellular computation. In PLS, an external potential is calculated with the gradient of the original image. An active contour dynamically deforms its shape with both its internal energy and the external potential. In the initial state, an active closed curve is given to enclose objects. The shape of the active closed curve iterates the

deformation and the active closed curve contracts until both the internal energy and the external potential are equivalent. When the internal energy and the external potential are reasonable, the equilibrium point of the active closed curve is the contour image of the object.

(C) Morphological wavelet transform [5], [6]
Morphological wavelet transform is wavelet transform based on max-plus algebra. Morphological wavelet transform is not affected by any restrictions regardless of the type of sampling windows. Morphological processing on the CAM^2 using various structure elements has already been proposed.

(D) Morphological pattern spectrum-based image manipulation detector [7], [8]
The CAM^2 can realize real-time parallel image-manipulation detection and 1,024-parallelism AES processing. For verifying the effectiveness of the proposed image-manipulation method with the CAM^2, two types of benchmark images are analyzed by the proposed detection method. The number of clock cycles of the CAM^2 is up to 58% lower than that of conventional processors. Consequently, the proposed image-manipulation detector with the CAM^2 implementation is very effective for the investigation of crimes and photographic evidence.

3. CAM^2 on FPGA

The main benefits of CAM^2 on FPGA are its flexibility and low cost. In the case of CAM^2 on ASIC, bit length, number of words, and number of blocks are respectively fixed at 64 bits, 512 words, and 32 blocks. In contrast, CAM^2 on FPGA can change its configuration and capacity. For example, word can be changed from 64 bits to 32 bits and the number of blocks can be changed to from 16 to 32.

This flexibility can result in better processing. FPGA can also change the configuration and capacity if we change the IP. The CAM^2 cell on FPGA is shown in Fig. 4. The big difference between CAM^2 on ASIC and CAM^2 on FPGA is the CAM cell: on ASIC, it has little flexibility, and on FPGA, it has excellent flexibility, as shown in Fig. 4. The function of CAM^2 on FPGA was implemented as follows. The maskable OR search takes exclusive OR of the key data and cell data and takes logical conjunction of its result and the search mask. Next, the maskable partial-parallel write takes the logical conjunction of the word line and partial-write signal line. Hit-flag shift up and down is constant to ASIC. There functions are realized easily.

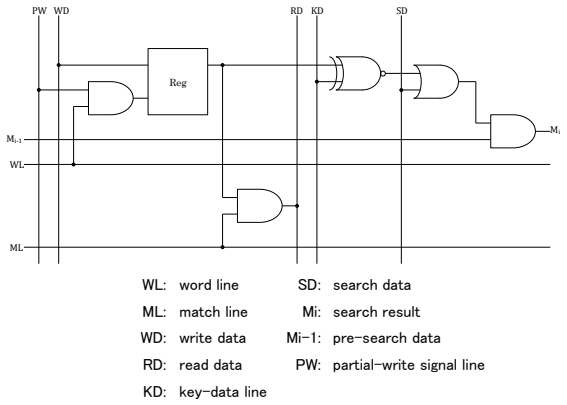

Figure 4: CAM^2 cell on FPGA

4. Implementation results

CAM^2 on FPGA is estimated to use three types of FPGA: Spartan-6, Kintex-7, and Virtex-7. A picture of Virtex-7 is shown in Fig. 5. The CAM^2 on ASIC is 40 MHz and has a 32-block configuration. The maximum capacities of each FPGA are 128 words $\times$ 8 blocks (Spartan-6), 512 word $\times$ 4 blocks (Kintex-7), and 512 words $\times$ 4 blocks (Virtex-7). The maximum implemented capacity of FPGA is smaller by one eighth than ASIC.
The max frequency of FPGA is smaller by 10% when the configuration becomes double blocks and smaller by 20% when it becomes double words. The maximum frequency of FPGA is fourth bigger than ASIC. From these results, multiply max frequency and capacity are about one twice.

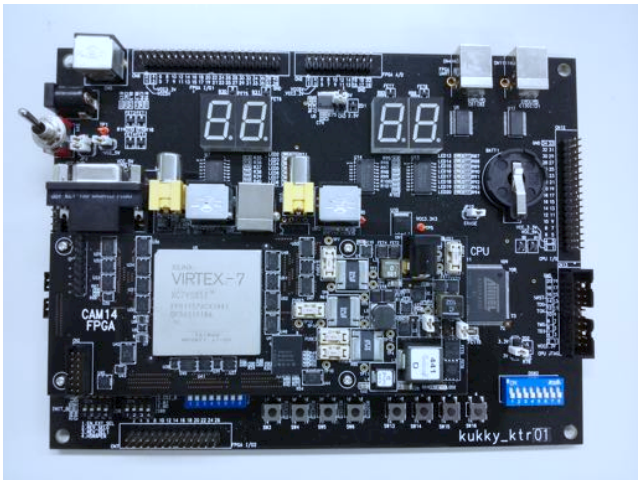

Figure 5: CAM^2 on FPGA

Table 2: Implementation results

Spartan-6(XC6SLX150)		
word × block	used resource	Max frequency(MHz)
128 × 2	44%	89
128 × 4	84%	82
128 × 8	87%	71

Kintex-7(XC7K480T)		
word × block	used resource	Max frequency(MHz)
256 × 8	90%	205
512 × 4	93%	177

Virtex-7(XC7V585T)		
word × block	used resource	Max frequency(MHz)
256 × 8	73%	205
512 × 4	75%	177

5. Conclusion

In this paper, we discussed our implementation of CAM^2 on FPGA. CAM^2 has previously been used to conduct processes such as pixel level snake, morphological wavelet transform, and pattern spectrum. CAM^2 is implemented on FPGA, on which it can change its capacity and configuration and realize a low cost. The biggest difference between CAM^2 on ASIC and on FPGA is the CAM cell. Simulations showed that the CAM^2 on FPGA achieves approximately four times the frequency compared with on ASIC, and the capacity became one eighth. Hence, using state of the art FPGA, CAM^2 on FPGA becomes larger and larger. CAM^2 implemented on FPGA has great potential as the accelerator of various CA-based processes.

Acknowledgments

Part of this work has been supported by a Grant-in-Aid for Scientific Research(C)(NO.24500287) from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

[1] T. Ikenaga and T. Ogura, "CAM^2: A Highly -Parallel Two-Dimensional Cellular Automaton Architecture", IEEE Trans. on Computers, vol. 47, no. 7, pp. 788-801, Jul. 1998.

[2] T. Ikenaga and T. Ogura, "A Fully Parallel 1-Mb CAM LSI for Real-Time Pixel-Parallel Image Processing", IEEE Journal of Solid-State circuits, vol. 35, no. 4, pp. 536-544, Apr. 2000.

[3] M. Nakanishi and T. Ogura, "A real-time CAM-based hough transform algorithm and its performance evaluation", Proc of 13th Int. Conf. Pattern Recognition(ICPR'96), vol. 2, pp. 516-521, 1996

[4] T. Matsui, T. Fujita, Y. Tsuji, T. Kumaki, M. Nakanishi, and T. Ogura, " Evaluation of Advanced Pixel-Level Snakes on Cellular Hardware Platform", NEWCAS, Jun. 2013.

[5] S. Shirai, M. Nakai, T. Kumaki, T. Fujita, M. Nakanishi, and T.Ogura, "Morphorogical wavelet transform using multiple directional sampling windows on cellular hardware platform", NEWCAS, Jun. 2011.

[6] T. Ikenaga and T. Ogura, "Real-time morphological processing using highly parallel 2-D cellular automata CAM^2 ", IEEE Trans. Image Processing, vol. 9, no. 12,pp. 2018-2026, Dec. 2000

[7] T. Kumaki, Z. B. Rafii, T. Fujita, M. Nakanishi, and T. Ogura, "Morphological pattern spectrum-based image manipulation detector", NOLTA2012, pp. 5-8, Oct. 2012.

[8] Z. B. Rafii, et al., "Real-time morphological pattern spectrum analyzer on cellular-automata hardware platform", Proc. of IEEE International Workshop on Nonlinear Circuit, Computer and Signal Processing, pp. 360-362, Mar. 2011.

[9] T. Fujita, T. Okamura, M. Nakanishi, and T. Ogura, "CAM^2-Universal Machine: A DTCNN Implementation for Real-Time Image Processing" 11th International Workshop on Cellular Neural Networks and their Applications, pp.219-223, Jul. 2008.

[10] T. Matsui, T. Fujita, M. Nakanishi, and T. Ogura, "Nonlinear Image Processing for Multiple Object Tracking on Cellular Hardware Platform", NOLTA2010, pp. 469-472, Sep. 2010.