

## A MORPHOLOGICAL ASSOCIATIVE MEMORY EMPLOYING SIMPLIFIED REVERSE RECALL AND ITS HARDWARE MODEL

HIDETAKA HARADA AND TSUTOMU MIKI

Graduate School of Life Science and Systems Engineering  
Kyushu Institute of Technology  
2-4 Hibikino, Wakamatsu-ku, Kitakyushu 808-0196, Japan  
harada-hidetaka@edu.brain.kyutech.ac.jp

Received August 2013; revised January 2014

**ABSTRACT.** *A morphological associative memory (MAM) has been proposed by Ritter as an associative memory model. This model has the advantages of a large memory capacity and high perfect recall rates, in comparison with other associative memory models. However, the MAM cannot handle a pattern that is completely included with other stored patterns. To overcome this problem, we proposed an MAM employing a reverse recall. However, the extra recall time of the model increases as the number of included patterns increases. Therefore, we developed an MAM employing a simplified reverse recall to reduce the extra recall time. This model is suitable for hardware implementation because the model uses simple operations and has a parallel architecture. Thus, we propose a high-speed compact hardware model of the MAM employing the simplified reverse recall in parallel architecture to accelerate the operation and for real world applications. In this paper, we introduce the MAM hardware we have developed. We confirm the performance of the MAM hardware model by comparing its operation speed with that of a software model. We expect that the MAM hardware model will be applied to pattern recognition, filters and pattern retrievals, etc.*

**Keywords:** Morphological associative memory, Reverse recall, Pattern with inclusive relation, Hardware model

**1. Introduction.** Associative memory is an important brain function and has attracted great attention. In associative memory models, the Hopfield network [1] is recognized as the most popular model. However, it has drawbacks such as a low memory capacity in relation to the number of memory units, and low convergence caused by a local minimum. As one of several associative memory models, Ritter et al. proposed morphological associative memory (MAM) [2], based on morphological neural networks [3]. In comparison with other associative memory models such as the Hopfield network, the MAM's advantages are: large memory capacity and a high perfect recall rate. In general, the model also has a drawback in that the design of the kernel image for association becomes harder as the total number of stored patterns increases. To overcome this problem, several effective methods to design the kernel image have been proposed [4-6]. Unfortunately, these methods have a problem, that is, a perfect recall cannot be achieved for an input pattern with a corrupted kernel image and stored patterns with redundancy bits are necessary for the kernel images. Therefore, we proposed an MAM using a stored pattern independent kernel image [7]. Our kernel design method offers these advantages: a simple kernel design, and good recall performance for a corrupted kernel image. However, like conventional MAMs, our model cannot recall the correct pattern for a pattern that is completely included with other stored patterns (e.g., "C and G", "E and F"). The existence of such inclusion patterns becomes problematic when the number of stored patterns increases.

As a solution, we proposed an MAM employing reverse recall [8]. In this model, we introduced the reverse recall method into the MAM using a stored pattern independent kernel image. Although the model can recall the plausible pattern for inclusion patterns without noise by using a reverse recall method, extra recall time is required and increases for reverse recall methods as the number of included patterns increases.

To reduce the extra recall time for the reverse recall method, we proposed a simplified reverse recall method that reduces extra recall time by simplifying a formula of the reverse recall for binary stored patterns [11]. The MAM employing the simplified reverse recall method is suitable for hardware implementation because its operations are simple and executable in parallel. As hardware implementation of an MAM, Guzman et al. proposed a processor-based MAM that is effective in image recognition [10]. Unfortunately, the processor cannot handle inclusion patterns because it is based on Ritter's original MAM. In our previous work, we proposed a high-speed compact hardware model of the MAM employing simplified reverse recall in parallel architecture to accelerate the operation and for real-world applications [12]. This MAM hardware model is designed to handle the inclusion patterns by refining our previous work [7]. In this paper, we introduce the MAM hardware model we have developed.

The model developed in this study can recall a plausible pattern even from a similar pattern. Furthermore, it can separate each pattern from the inclusion pattern, and recall a plausible recall pattern corresponding to an input pattern. For example, this model can be applied to real-world applications such as image recognition and character recognition, where many similar patterns exist. In addition, the developed hardware model can recognize a pattern with high-speed operation, working in parallel. We confirm the performance of the MAM hardware model by comparing the operation speed of the hardware model with that of a software model. This paper is a culmination of our previous three papers [8,11,12].

## 2. Morphological Associative Memory: MAM.

**2.1. Ritter's MAM.** Ritter's MAM consists of two-stage recall processes using memory matrices "M" and "W" in the stages [2], as shown in Figure 1. In the recall process of Ritter's MAM, a kernel image  $Z^r$  is used as an index for association. The kernel image consists of partial units of the stored pattern  $X^r$ . The elements of the memory matrices  $m_{ij}$  and  $w_{ij}$  are given as

$$m_{ij} = \bigvee_{r=1}^R (z_i^r - z_j^r), \quad (1)$$

$$w_{ij} = \bigwedge_{r=1}^R (y_i^r - z_j^r), \quad (2)$$

where  $R$  is the total number of stored patterns.  $\vee$  and  $\wedge$  denote maximum and minimum operators, respectively. The output pattern  $Y^r$  is obtained as follows:

$$z_i^r = \bigwedge_{j=1}^n (m_{ij} + x_j^r) \quad i = 1, \dots, n, \quad (3)$$

$$y_i^r = \bigvee_{j=1}^n (w_{ij} + z_j^r) \quad i = 1, \dots, m. \quad (4)$$

In Ritter's MAM, the kernel image corresponding to the input pattern can be recalled at the first recall using Equation (3), and the output pattern can be obtained at the second recall using Equation (4) for the kernel image recalled by the first recall.

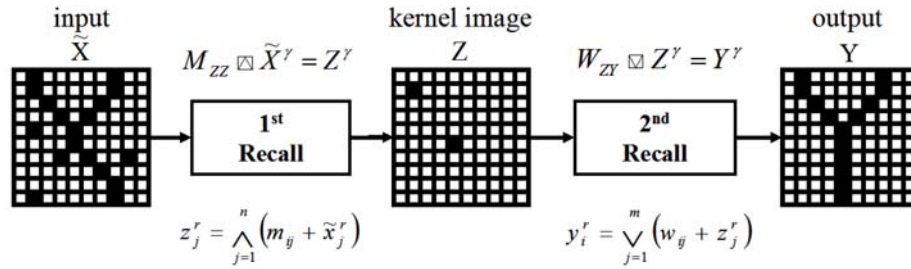


FIGURE 1. Schematic of Ritter's MAM having two-stage recall process [2]

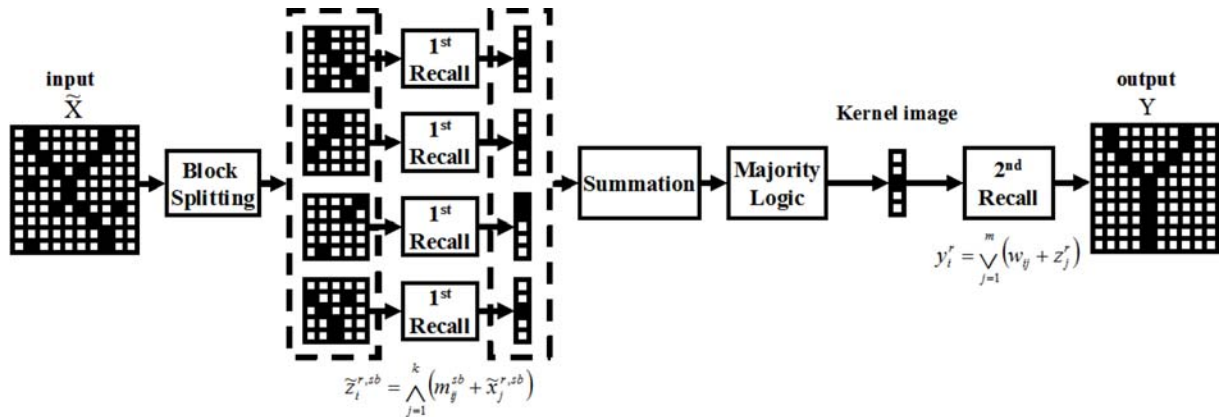


FIGURE 2. Schematic of the MAM using the stored pattern independent kernel image [7]

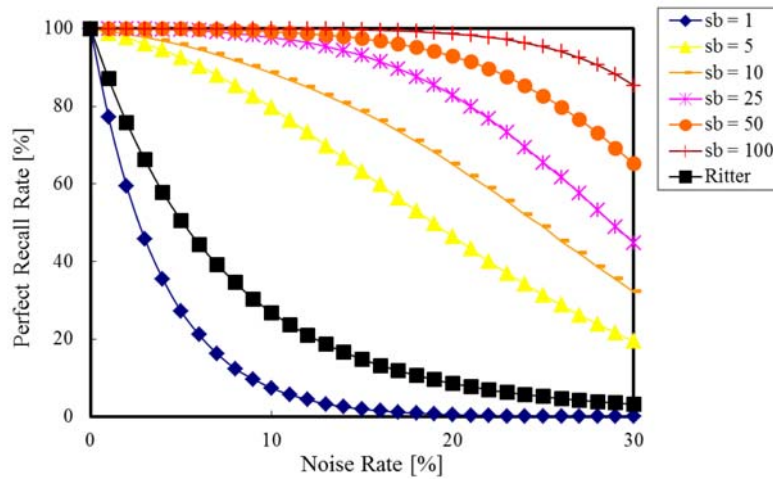


FIGURE 3. Noise tolerance of the MAM using a stored pattern independent kernel image and Ritter's MAM for twenty alphabet patterns [7]. "sb" represents the number of blocks in the block splitting step in Figure 2.

**2.2. MAM employing effective kernel design.** In Ritter's MAM, the design of the kernel image becomes difficult as the total number of stored patterns increases, because the kernel image consists of partial units of stored pattern that do not overlap. Therefore, we proposed an MAM using a stored pattern independent kernel image [7]. In this kernel design method, only one element of the kernel image is "1" while the others are "0". Figure 2 shows the recall process of the MAM using a stored pattern independent kernel image. Figure 3 shows noise tolerance of the MAM using a stored pattern independent

kernel image and Ritter’s MAM in the hetero-association experiments for twenty alphabet patterns. This model facilitates the design of the kernel image and improves the perfect recall rate when noise appears in the input images.

**2.3. MAM employing reverse recall.** However, like conventional MAMs, the MAM using a stored pattern independent kernel image cannot recall the correct pattern for a pattern completely included with other stored patterns (e.g., “C and G”, “E and F”), which is called “*inclusion pattern*” in this paper. As shown in Figure 4, an MAM using a stored pattern independent kernel image recalls the overlapped output pattern for the inclusion pattern.

To overcome this problem, we proposed a reverse recall method [8]. As shown in Figure 5, in this method, the candidates of an input pattern are reversely recalled by each separated kernel image. Subsequently, the plausible recall pattern can be determined by comparing the input pattern with the candidates obtained by the reverse recall. Figure 6 shows noise tolerance of the MAM employing the reverse recall method, and the MAM using the stored pattern independent kernel image in hetero-association experiments for twofold inclusion patterns.

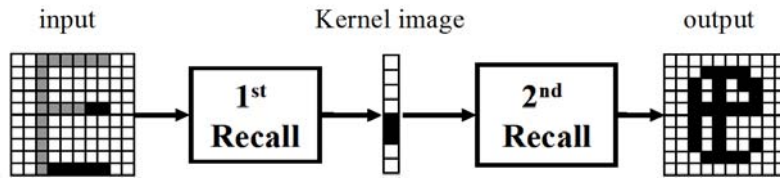


FIGURE 4. Problem of an MAM using a stored pattern independent kernel image. Pattern “F” is completely included with pattern “E”.

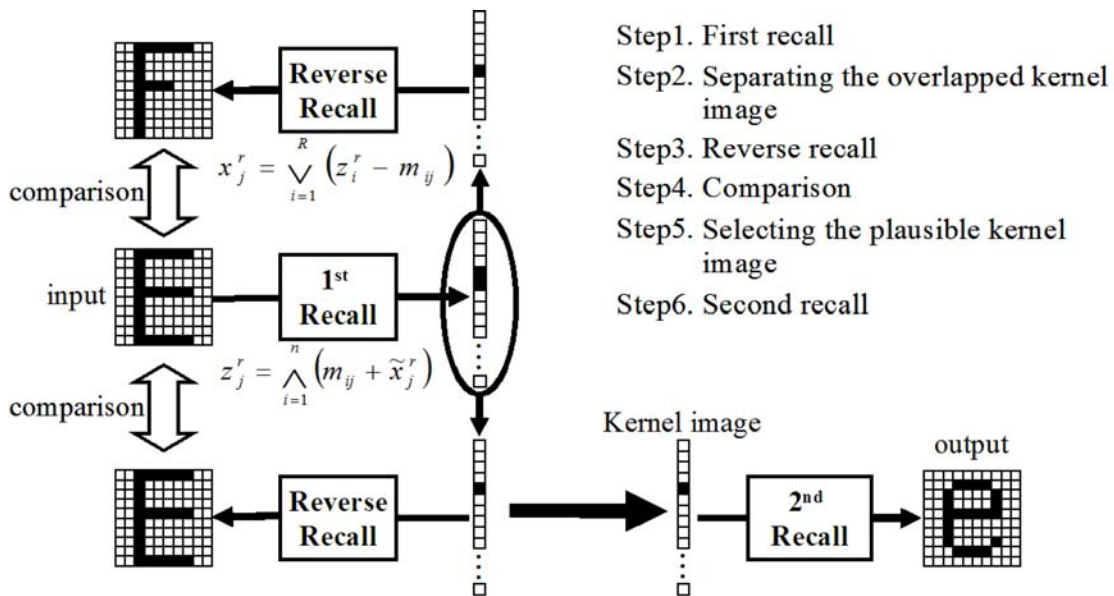


FIGURE 5. Schematic of an MAM employing reverse recall [8]

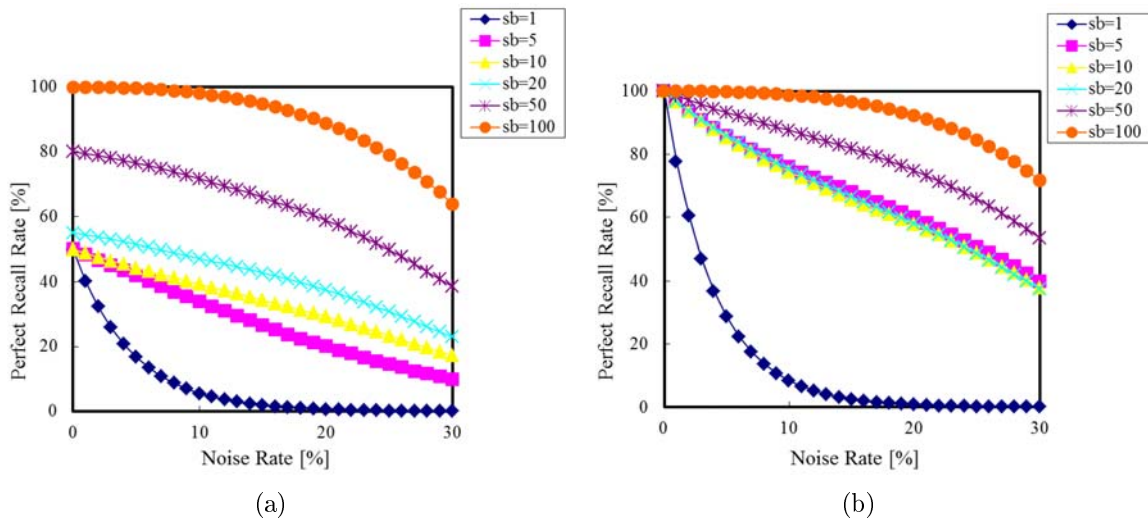


FIGURE 6. Noise tolerance in hetero-association for twofold inclusion patterns: (a) the result of the MAM employing reverse recall, and (b) the stored pattern independent kernel image method

### 3. Hardware Implementation.

**3.1. Simplified reverse recall method.** In an MAM employing a reverse recall, additional calculations are required for the reverse recall method. The extra recall time is approximately equal to  $0.5 \times N_{ip} \times T_{recall}$ , where  $N_{ip}$  is the number of included patterns, and  $T_{recall}$  is the recall time without inclusion patterns. In an MAM employing a reverse recall, the overlapped kernel image is separated into individual kernel images, and then the stored pattern corresponding to each kernel image is reversely recalled using each separated kernel image.

Here, in an MAM employing reverse recall, if  $X^r = (x_1^r, \dots, x_n^r)$  and  $Z^r = (z_1^r, \dots, z_n^r)$ , the stored pattern  $X^r$  reversely recalled for the kernel image  $Z^r$  is given as

$$x_j^r = \bigvee_{i=1}^R (z_i^r - m_{ij}), \quad (5)$$

where  $z_i$  is  $i$ -th unit of the kernel image  $Z^r$ , and  $x_j$  is  $j$ -th unit of the stored pattern  $X^r$ .  $R$  is the total number of stored patterns. For binary stored patterns, Equation (5) can be rewritten as

$$x_j^r = z_r^r - m_{rj}. \quad (6)$$

Equation (6) shows that the stored pattern corresponding to each kernel image can be reversely recalled by one element of the kernel image. Using Equation (6) instead of Equation (5), the stored pattern can be reversely recalled in one shot. In this method, if the overlapped kernel image is recalled at the first recall, the stored pattern corresponding to each kernel image is reversely recalled by the corresponding firing unit of the kernel image. The plausible kernel image is then selected by comparing the input pattern with the patterns obtained by the simplified reverse recall, as illustrated in Figure 7.

**3.2. MAM hardware model.** The block diagram of the MAM hardware we developed [12] handles thirty-two stored patterns, shown in Figure 8. It is designed by introducing a reverse recall unit into the hardware model of the MAM using the stored pattern independent kernel image [7]. Here, each stored pattern consists of 64 binary units. The MAM hardware model consists of a memory unit, first recall unit, majority logic unit, reverse recall unit, and second recall unit. In the memory unit, memory matrices “ $M$ ”

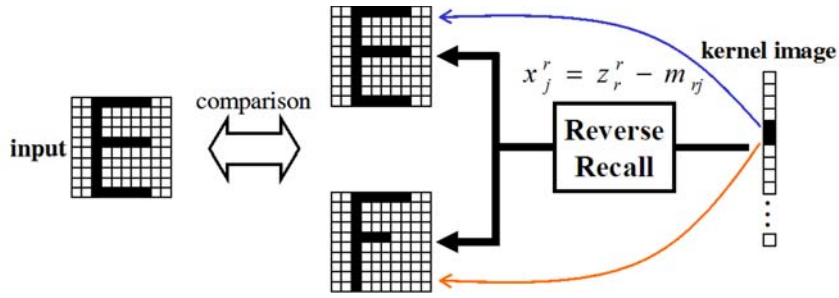


FIGURE 7. Reverse recall process of the simplified reverse recall [11]

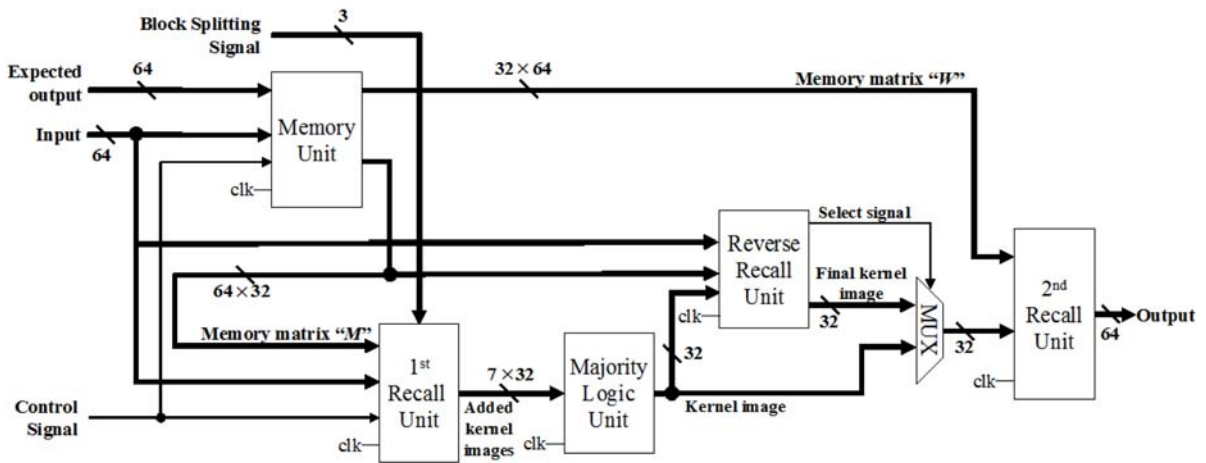


FIGURE 8. Block diagram of the MAM hardware model [12]

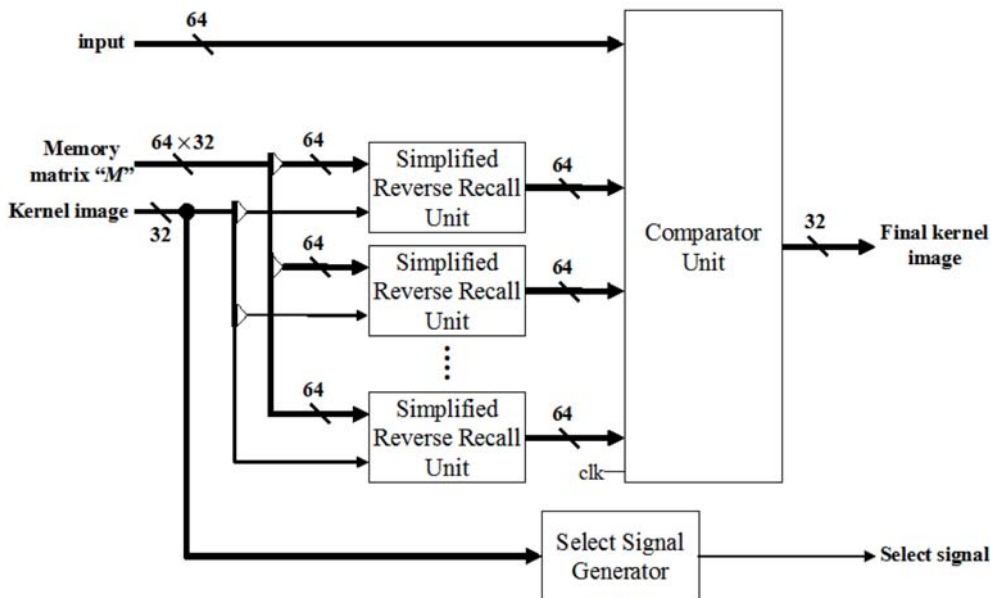


FIGURE 9. Block diagram of the reverse recall unit

and “ $W$ ” are calculated from the input and corresponding output patterns. Using the first recall and majority logic units, the kernel image corresponding to the input pattern is recalled. When the overlapped kernel image is recalled, the plausible kernel image can be determined from the overlapped kernel image at the reverse recall unit. Finally, the

second recall unit calculates the output pattern corresponding to the final kernel image. Figure 9 shows the block diagram of the reverse recall unit. In this unit, using a simplified version, the stored pattern corresponding to each kernel image is calculated using Equation (6). In the comparator unit, the final kernel image is determined by comparing the input with the stored patterns reversely recalled by the simplified reverse recall unit. In the selected signal generator, the overlapped kernel image is detected. If the overlapped kernel image cannot be detected, the kernel image of the majority logic unit is selected.

#### 4. Experimental Results.

**4.1. MAM software model.** To evaluate performance of the MAM employing the simplified reverse recall method, we investigate the perfect recall rate through hetero-association experiments. In the experiment, each pattern consists of  $10 \times 10 = 100$  binary units. The unit of pattern takes either “1” or “0”, which represents black and white, respectively.

We investigate the perfect recall rate of the MAM employing the simplified reverse recall for patterns shown in Figure 10, which shows stored patterns consisting only sets of twofold inclusion patterns. The expected patterns are different from those illustrated in Figure 10. Here, the perfect recall rate is evaluated by an average of 10,000 trials in simulation. The noise is defined as changing “1” to “0” (or “0” to “1”).

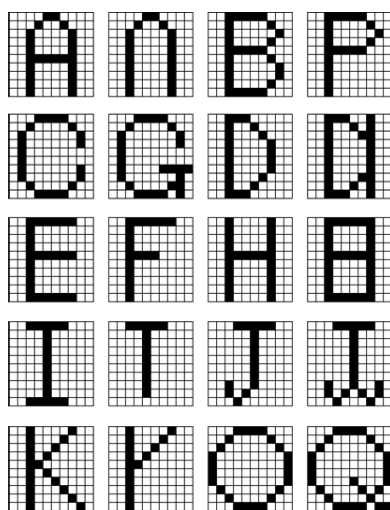


FIGURE 10. Stored patterns: twenty patterns consisting only sets of twofold inclusion patterns

Figure 11 shows noise tolerance of the MAM using the previous reverse recall and the simplified reverse recall method for patterns illustrated in Figure 10. As shown in Figure 11, the performance of the simplified reverse recall method is equivalent to the previous reverse recall method.

**4.2. MAM hardware model.** The performance of the MAM hardware model is investigated by using a logic simulator *ModelSim* [9] for hetero-association experiments. Figures 12(a) and 12(b) show the stored pattern pairs. Each stored pattern consists of  $8 \times 8 = 64$  binary units.

Figure 13 shows the simulation result of the MAM hardware model for twofold inclusion patterns. As shown in Figure 13, the output corresponding to the input pattern can be recalled in fifteen clock cycles.

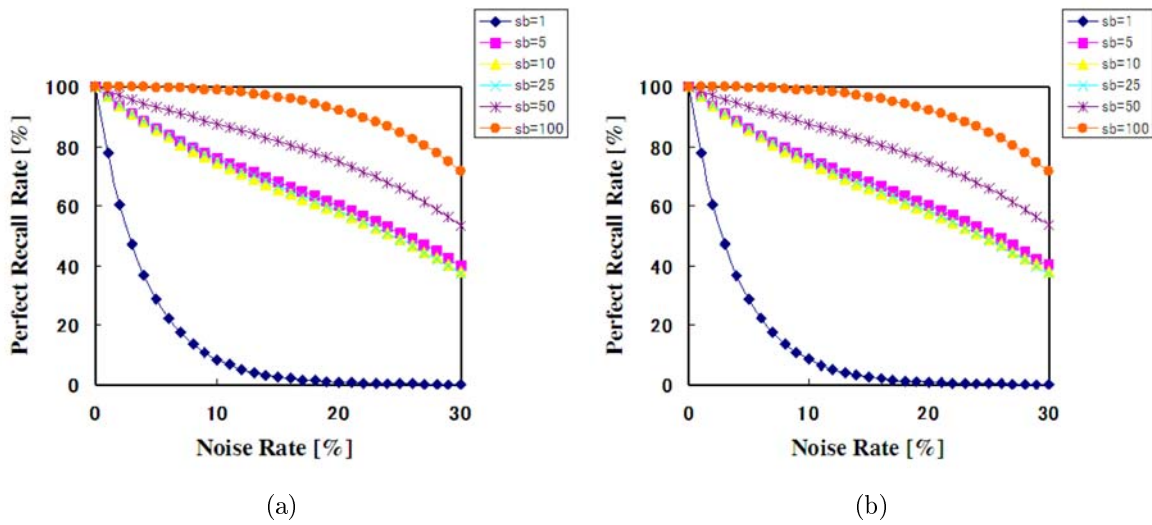


FIGURE 11. Noise tolerance in hetero-association for twofold inclusion patterns: (a) the result of the MAM employing the previous reverse recall and (b) the simplified reverse recall method

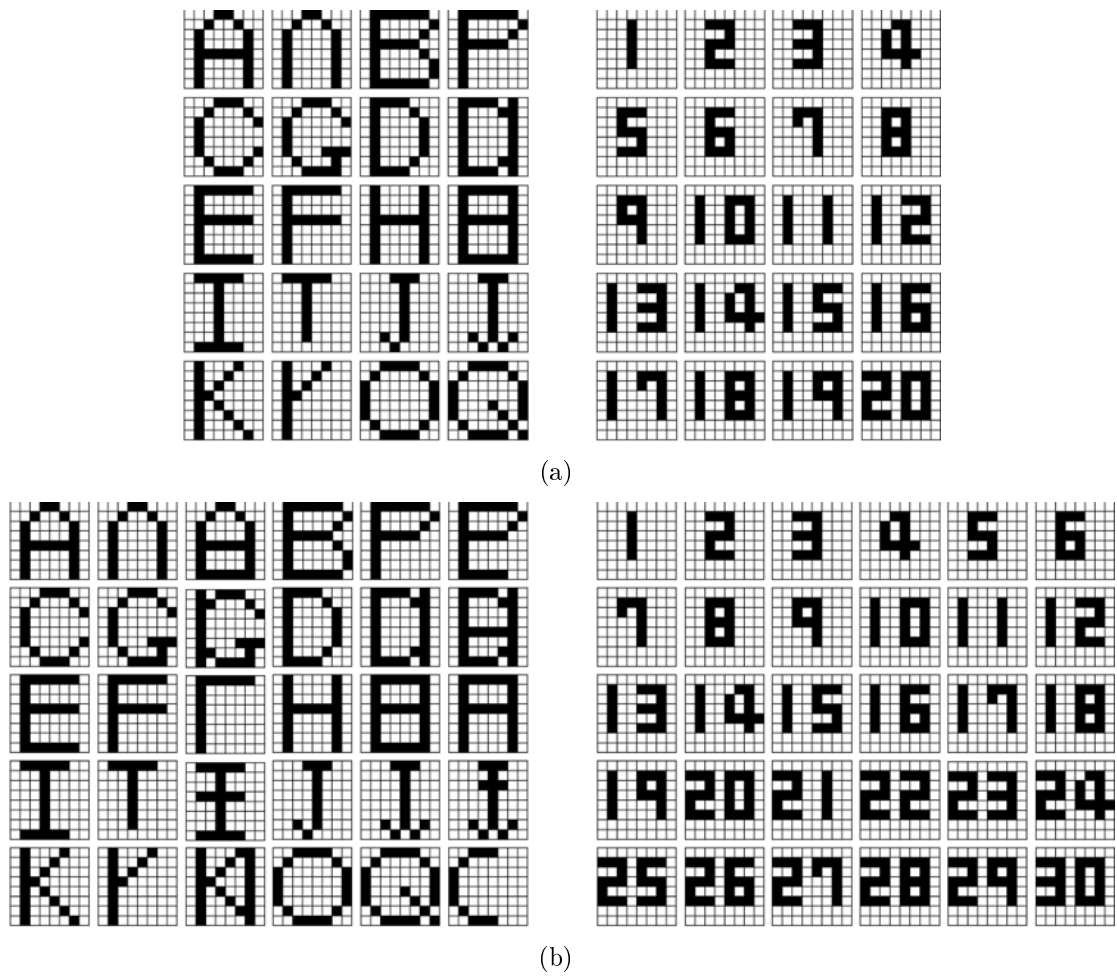


FIGURE 12. Stored pattern pairs (input and expected patterns): (a) twofold inclusion patterns and (b) threefold inclusion patterns



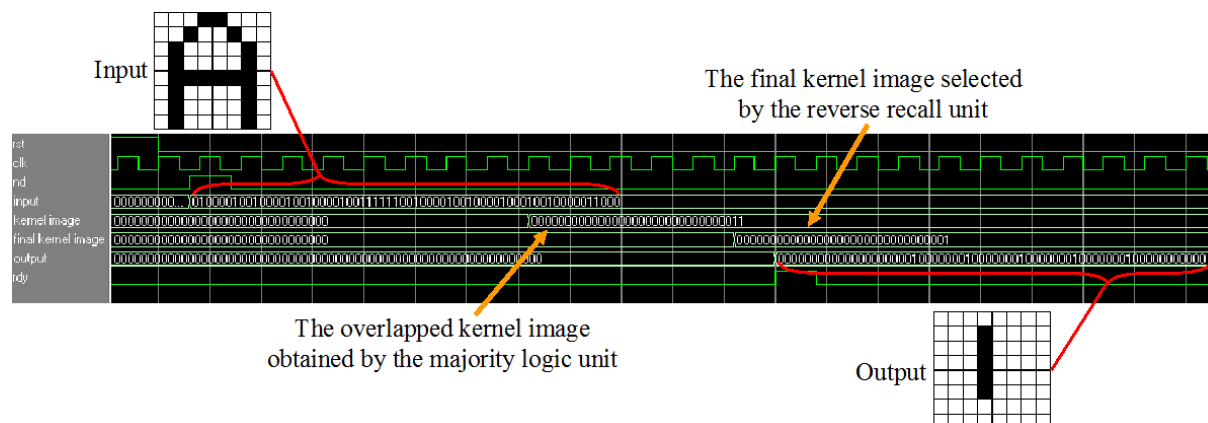


FIGURE 13. Simulation result of the MAM hardware model

TABLE 1. Performances of the MAM hardware model

Method		Maximum clock frequency (GHz)	Operation speed ( $\mu\text{sec}$ )	
			Twofold inclusion	Threifold inclusion
Software	Previous reverse recall	2.8	43.2	54.1
	Simplified reverse recall	2.8	31.4	32.1
Hardware		0.14	0.11	0.11

Table 1 shows comparison of the performances on recalls for inclusion patterns between the MAM hardware and software models. Here, the CPU used is an Intel Core i5@2.8 GHz. As shown in Table 1, the MAM hardware model achieved approximately 280 times speedup in comparison with software execution of the simplified method. Furthermore, the recall time of the MAM hardware model does not increase even if the number of inclusion patterns increases. Here, the Virtex5 (xc5vsx95t) with 58,880 logic slices is assumed as a target device.

**5. Conclusion.** In this paper, we introduced an MAM hardware model of an MAM employing the simplified reverse recall. In our previous work, we proposed an MAM employing the simplified reverse recall to reduce the extra recall time for a reverse recall method. This model is suitable for hardware implementation because of its simple and parallel architecture. Therefore, to accelerate the operation, we proposed the hardware model of the MAM employing a simplified reverse recall for binary stored patterns. This MAM hardware model achieved more than 280 times speedup in comparison with software execution.

The MAM hardware model can recall the associated pattern at high speed for noisy and inclusion patterns as well. We expect this hardware model to work effectively in real-world applications of pattern recognition, filters and pattern retrievals, etc., where many similar patterns exist. Moreover, the compactness and high-speed operations in the MAM hardware model are expected to contribute to vision systems for autonomous mobile robots. In our future work, we will extend the handling pattern to more complicated values and tackle applying our hardware model for practical applications.

## REFERENCES

- [1] J. J. Hopfield, Neural network and physical systems with emergent collective computational abilities, *Proc. of the National Academy of Sciences of the United States of America*, vol.79, no.8, pp.2554-2558, 1982.
- [2] G. X. Ritter, P. Sussner and J. L. Diaz-de-Leon, Morphological associative memory, *IEEE Trans. on Neural Networks*, vol.9, no.2, pp.281-293, 1998.
- [3] J. L. Davidson and G. X. Ritter, A theory of morphological neural networks, *SPIE Digital Optical Computing II*, vol.1215, pp.378-388, 1990.
- [4] T. Ida, S. Ueda, M. Kashima, T. Fuchida and S. Murashima, On a method to decide kernel patterns of morphological associative memory, *IEICE, Trans. on Information and Systems, Part 2*, vol.J83-D-II, no.5, pp.1372-1380, 2000.
- [5] P. Sussner, Associative morphological memories based on variations of the kernel and dual kernel methods, *Neural Networks*, vol.16, no.5-6, pp.625-632, 2003.
- [6] M. Hattori, A. Fukui and H. Ito, A fast method to decide kernel patterns for morphological associative memory, *IEEJ Trans. on Electronics, Information and Systems*, vol.123, no.10, pp.1830-1838, 2004.
- [7] H. Harada, T. Saeki and T. Miki, A morphological associative memory employing a stored pattern independent kernel image and its hardware model, *IWCIA*, pp.219-224, 2009.
- [8] H. Harada and T. Miki, A morphological associative memory employing a reverse recall, *The 17th International Conference on Neural Information Processing, Part II*, pp.321-328, 2010.
- [9] *Mentor Graphics*, <http://www.model.com/>.
- [10] E. Guzman, S. Alvarado, O. Pogrebnyak and C. Yanez, Image recognition processor based on morphological associative memories, *CERMA*, pp.260-265, 2007.
- [11] H. Harada and T. Miki, Recall time reduction of a morphological associative memory employing a reverse recall, *The 18th International Conference on Neural Information Processing, Part III*, pp.332-339, 2011.
- [12] H. Harada and T. Miki, A hardware model of a morphological associative memory employing a simplified reverse recall, *ICIC Express Letters*, vol.6, no.3, pp.833-838, 2012.