Design and Analysis of a CMOS Ratio-Memory Cellular Nonlinear Network (RMCNN) Requiring No Elapsed Time

Chung-Yu Wu, *Fellow, IEEE*, Sheng-Hao Chen, and Yu Wu

***Abstract*—A CMOS ratio-memory cellular nonlinear network (RMCNN) requiring no elapsed time is proposed. The correlations between any two neighboring cells are stored in the memories. The ratio weights of each cell are generated through a comparison of the four correlations around one cell with the mean value of these four correlations. With this method, the elapsed time required by the previously existing RMCNN algorithm is no longer required and, therefore, the ratio weights can be generated individually. Moreover, the use of multi-dividers can be avoided to make the circuit simple. Based on the proposed algorithm, a CMOS RMCNN chip requiring no elapsed time has been designed and fabricated using TSMC 0.35-μm 2P4M mixed-signal technology. In the fabricated chip, three test patterns can be learned and recognized.**

***Index Terms*—Cellular nonlinear networks (CNN), CMOS, elapsed time, ratio memory cellular nonlinear networks (RMCNN).**

I. Introduction

THE cellular nonlinear (neural) network (CNN) which was proposed by Chua and Yang in 1988 [1]–[3] involves a large-scale nonlinear analogic architecture for real-time signal processing. Similar in composition to the cellular automata [4], [5], it is comprised of a massive aggregation of regularly spaced circuit clones, called cells, which communicate with each other directly and locally. With local connectivity, CNN is quite suitable for very large-scale integration (VLSI) implementation. The associated real-time and parallel-operating properties also make it popular in image processing. To date, many CNN VLSI chips have demonstrated their capabilities in realizing real-time signal and parallel processing functions [6]–[9]. In these chips, the templates, which can control the communications between cells, are programmable and the regular and local functions can be designed and applied on the entire CNN array. However, because the local characteristics are usually different in the learned patterns, the space-variant templates are more suitable than the space-invariant templates. To address this limitation, some algorithms that collect global image characteristics are proposed [10]–[12] to learn the images and some researches on pattern recognition are realized by classification [13].

To realize an on-line learning CNN with local-computing advantage, a learning algorithm called ratio-memory CNN (RMCNN) is proposed [14]–[16]. The ratio memory of the Grossberg outstar structure [17]–[19] has been used in both feedforward and feedback neural network ICs for image processing. With the proposed RMCNN, no host computer is needed to perform the off-line learning task. It can also evaluate the correlations between cells and store these correlations on the capacitors. As a result, it achieves a long template-weight storage time or equivalent pattern recognition time which is one of the advantages of RM. The charge stored on the capacitors leaks out due to the junctions from the source and drain of CMOS to the substrate. The RMCNN utilizes this leakage effect and takes the ratio of the stored values to enhance the common characteristics of the learned patterns and to raise the recognition rate. Therefore, a very long period of elapsed time is required after the learning period to make the weights of small correlations smaller or to approach zero by the leakage in order to enhance large correlations [16]. As shown in Fig. 1, the required elapsed time is dependent on learned patterns. It shows that 1500 seconds of elapsed time are required when 5 patterns are learned where 700 seconds are required when three patterns are learned. The learned correlations in different local positions of the learned patterns are distinct and the learned values have significant differences. Hence, if the elapsed time is too long, the most learned correlations will be destroyed and the recognition rate decreases. Oppositely, if the elapsed time is too short, the characteristics of large correlations cannot be enhanced and the recognition rate cannot be improved significantly. Furthermore, when the RMCNN is utilized to learn and recognize the image patterns, the stored values keep on leaking during the recognition time and this may, with time, alter the ratio weights of the RMCNN. Finally, as the weights of cells are generated by ratio memories, a precise multi-divider is required.

In this work, RMCNN architecture without elapsed time [20] is proposed and analyzed to prevent the leakage effect and to simplify the circuitry. With the new algorithm, the feature enhanced ratio weights can be generated immediately after the learning period without the requirement of elapsed time and, therefore, the circuit to generate ratio weights could be very simple and remove the need for multi-dividers in each ratio memory. An RMCNN chip not requiring elapsed time

Manuscript received March 24, 2009; revised June 28, 2009; accepted August 05, 2009. First published January 12, 2010; current version published June 09, 2010 This work was supported by the National Science Council (NSC), Taiwan, under Grant NSC 96-2221-E-009-178. . This paper was recommended by Associate Editor M. Delgado-Restituto.

C.-Y. Wu is with the Nanoelectronics and Gigascale Systems Laboratory, Institute of Electronics, National Chiao-Tung University, Hsinchu, Taiwan (e-mail: cywu@alab.ee.nctu.edu.tw).

S.-H. Chen was with Nanoelectronics and Gigascale Systems Laboratory, Institute of Electronics, National Chiao-Tung University, Hsinchu, Taiwan. He is now with MediaTek Corporation, Hsinchu, Taiwan.

Y. Wu was with Nanoelectronics and Gigascale Systems Laboratory, Institute of Electronics, National Chiao-Tung University, Hsinchu, Taiwan. He is now with eMemory Company, Hsinchu Science Park, Hsinchu, Taiwan.

Digital Object Identifier 10.1109/TCSI.2009.2031704

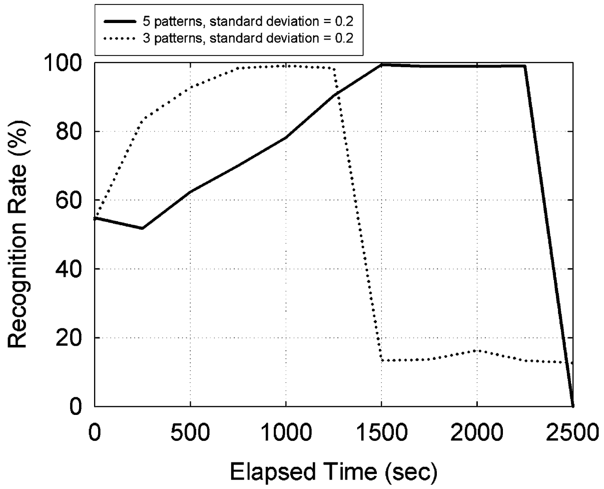


Fig. 1. Recognition rates with different elapsed time where two sets of patterns are learned and the patterns with Gaussian noise of standard deviation 0.2 are recognized.

has been designed and fabricated using TSMC 0.35-μm 2P4M mixed-signal technology. Patterns are learned and recognized with the proposed architecture and the results are analyzed and discussed. The total chip area is 4560μm×3900 μm and the area of a single cell is 400μm×250 μm. The total power consumption is 87 mW in operation with a supply voltage of 3 V.

In Section II, the models and architecture of the RN-CNN not requiring elapsed time are described. In Section III, the CMOS circuits of each block are illustrated and the HSPICE simulation results are presented to verify the functions of the blocks. In Section IV, the measurements obtained are presented and discussed. Finally, a conclusions are provided in Section V.

II. Models and Architecture

A. Model of the RMCNN Requiring No Elapsed Time

The definition of the RMCNN with elapsed time is described in [16]. The template A_{ij} of a RMCNN can be written as

$$A_{ij}(T_E) = \begin{bmatrix} 0 & a_{ij(i-1)j}(T_E) & 0 \\ a_{iji(j-1)}(T_E) & 0 & a_{iji(j+1)}(T_E) \\ 0 & a_{ij(i+1)j}(T_E) & 0 \end{bmatrix} \tag{1}$$

where $a_{ijkl}(T_E)$ is the template coefficient of the cell C_{ij} to stimulate the cell C_{kl} and is a ratio weight generated by using

$$a_{ijkl}(T_E) = \frac{\sum\limits_{p=1}^{m} \int_{Tp} u_{ij}^{p} \cdot u_{kl}^{p} dt - L_{kl}(T_E)}{\sum\limits_{kl} \left[\sum\limits_{p=1}^{m} \left| \int_{Tp} u_{ij}^{p} \cdot u_{kl}^{p} dt \right| - L_{lk}(T_E) \right]},$$
$$kl \in \{(i-1)j, i(j-1), i(j+1), (i+1)j\}. \tag{2}$$

In (2), u_{ij}^p and u_{kl}^p are the inputs of the pixels in the learned pth pattern when m patterns are learned in the learning period and L_{kl} is the intrinsic leakage of the weight $\sum_{p=1}^{m} \int_{Tp} u_{ij}^{p} \cdot u_{kl}^{p} dt$, which is different for different cells. The four ratio weights are stored in the four ratio memories around a cell. With a suitable elapsed time, the insignificant weights are decreased to zero and the pattern recognition and recovery can be performed successfully. The intrinsic leakage is utilized for characteristic enhancement.

However, the elapsed time control is difficult. With a long elapsed time, the remnants in the four ratio memories around one cell may leak out, completely. However, with a short elapsed time, the small stable ratio weights cannot be obtained and the recognition cannot be well performed. To solve the problem, a new template generating method is proposed. First, the mean M_{ij} of the learned absolute correlated weights is generated as

$$M_{ij} = \mathrm{avg}_{kl} \left(\left| \sum_{p=1}^{m} \int_{Tp} u_{ij}^{p} \cdot u_{kl}^{p} dt \right| \right),$$
$$kl \in \{(i-1)j, i(j-1), i(j+1), (i+1)j\} \tag{3}$$

where m is the number of learned patterns, T_p is the learning time of one pattern, and u_{ij}^p and u_{kl}^p are the inputs of $\mathrm{cell}(i,j)$ and $\mathrm{cell}(k,l)$, respectively, in the learned pth pattern. The ratio weight a'_{ijkl} is regenerated as following:

$$a'_{ijkl} = \begin{cases} 0, & \text{if } \left| \sum\limits_{p=1}^{m} \int_{Tp} u_{ij}^{p} \cdot u_{kl}^{p} dt \right| < M_{ij} \\ \frac{1}{\mathrm{PN}_{ij}}, & \text{if } \left| \sum\limits_{p=1}^{m} \int_{Tp} u_{ij}^{p} \cdot u_{kl}^{p} dt \right| \geq M_{ij} \end{cases}$$
$$kl \in \{(i-1)j, i(j-1), i(j+1), (i+1)j\} \tag{4}$$

where PN_{ij} means the number of the absolute correlated weights which is larger than M_{ij}. As shown in (4), the template coefficient is generated by comparing the absolute correlated weights and the mean M_{ij}. By counting PN_{ij}, the template value is set to $1/\mathrm{PN}_{ij}$ when its absolute correlated weight is larger than the mean. This retains the overall summation of absolute template coefficients a'_{ijkl} of template $\boldsymbol{A}$ at 1 to avoid any divergence in recognition.

To demonstrate why the coefficient which is larger than the mean is retained, a simple model of the absolute ratio weight can be constructed as following:

$$\frac{P-L}{(P-L)+(Q-L)+(R-L)+(S-L)} > \frac{P}{P+Q+R+S},$$
$$P > L, \quad Q > L, \quad R > L, \quad S > L \tag{5}$$

where P, Q, R, and S represent the four absolute correlated weights generated in the learning period, and L represents the average leakage in the elapsed period. After an elapsed period, the absolute ratio weight is enlarged when the coefficient is retained as following:

$$\frac{1}{4} \cdot \frac{P-L}{M-L} > \frac{1}{4} \cdot \frac{P}{M}, \quad P > M, \quad P > L,$$
$$Q > L, \quad R > L, \quad S > L \tag{6}$$

where M is the mean of P, Q, R, and S. When P is larger than M, the absolute ratio weight is enlarged after a period of elapsed time. Oppositely, when P is smaller than M, the absolute ratio weight would degrade to 0 after a period of elapsed time. To simplify the calculation, the coefficient is decided by comparing it with the mean value directly. Hence, if one of the

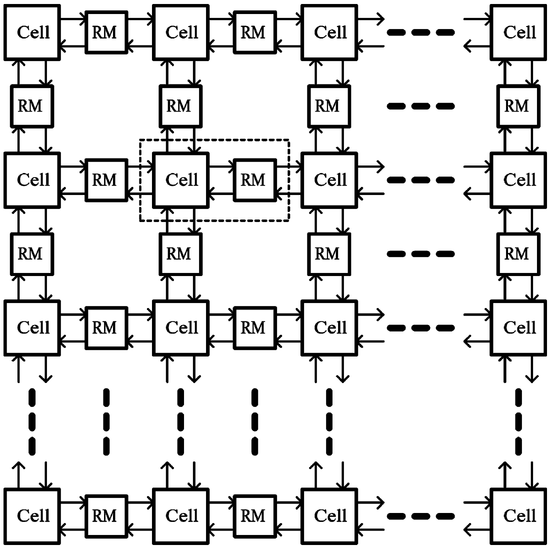


Fig. 2. General architecture of the RMCNN which contains cells and RMs.

correlated weights is smaller than the mean M, the correlation is assumed to leak out and the weight is set to zero. When the correlated weights are larger than the mean M, they are assumed to have the same weighting. In the proposed algorithm, when P is larger (smaller) than or equal to M, the absolute ratio weight is chosen to be $1/\mathrm{PN}$ (0) where PN is the number of the absolute correlated weights which is larger than M_{ij}. This makes the sum of the absolute ratio weights equal to 1 around one cell. Hence, RMCNN requiring no elapsed time is to predict the resultant ratio weight of RMCNN after elapsed time.

B. The Architecture of a RMCNN Not Requiring Elapsed Time

The general architecture of RMCNN is shown in Fig. 2. There are four RMs: up, down, right, and left sides around one cell and there are two neighboring cells around one RM. The structure of the kernel unit of the RMCNN not requiring elapsed time is separated into three parts in Fig. 3(a)–3(c). In Fig. 3(a), block Neuron is a neuron composed of a resistor and a capacitor and block VTI1 is a voltage-to-current converter to convert the input voltage into current. Block $VIT2$ is a voltage-to-absolute-current converter. In Fig. 3(b), block W is the synaptic gain block to multiply the absolute input current from VTI2 with a chosen weight of 1/4, 1/3, 1/2, or 1 and the output sign is controlled by a sign controller. The weight is controlled by block Counter_L in Fig. 3(c). The output current is injected to neuron of neighboring cells or the capacitor Cw to store the correlations in different operational periods. VTI3 converts the voltage on Cw into absolute current and the output is sent to block COMP. Block COMP is a comparator that can compare four absolute currents from VTI3 with the average of these four currents. Block Counter_L counts the number of currents which are larger than the average current. Block Counter_L can generate the signals to control the weights of blocks W by the comparing and counting results.

In the learning period, only switches $sw1$, $sw2$, $sw4$, and $sw5$ in Fig. 3(b) are open. As shown in Fig. 4(a) when pth pattern is learned, binary input u_{ij}^P of $\mathrm{cell}(i,j)$ is sent into block VTI1 and the output current is sent to block Neuron to generate the state voltage of $\mathrm{cell}(i,j)$. The positive or negative state of $\mathrm{cell}(i,j)$ is detected and the absolute current is extracted by block $VTI2$.

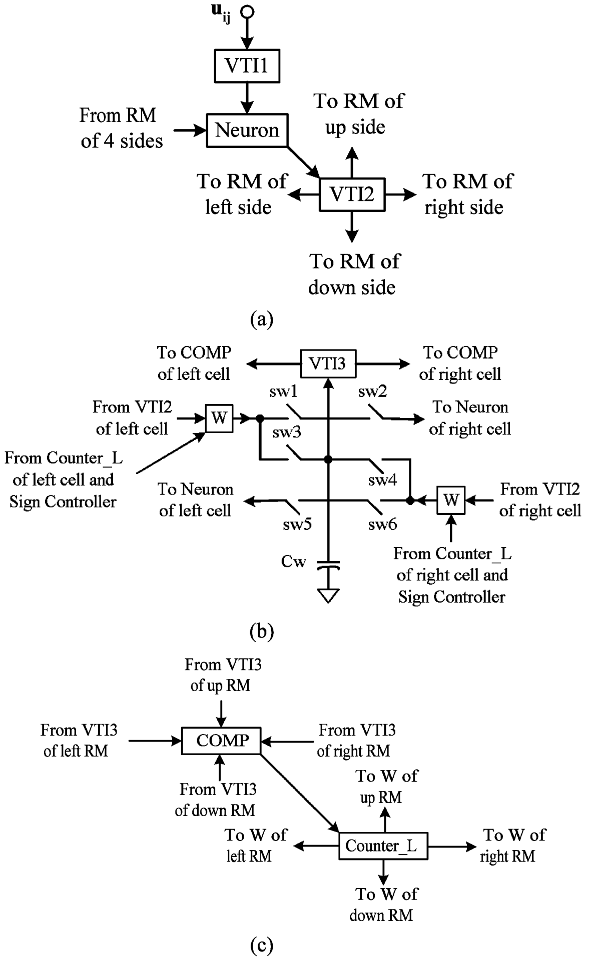
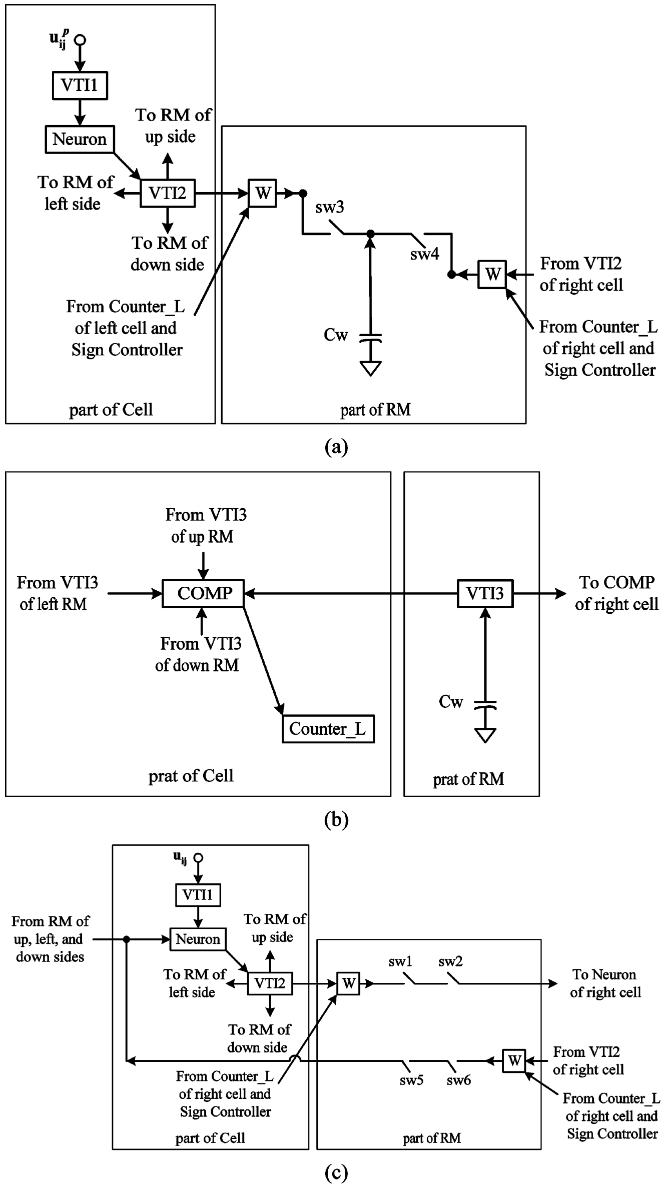


Fig. 3. Architecture of the kernel unit in RMCNN requiring no elapsed time which contains (a) Input stage and neuron, (b) RM, and (c) Comparator and counter.

The absolute current of $\mathrm{cell}(i,j)$ is sent to block W. In the learning period, the weight of block W is set to 1/4. If the states of $\mathrm{cell}(i,j)$ and its neighboring cells are the same (different), it is decided to charge (discharge) capacitor Cw with the absolute current multiplied by 1/4. The learning time of one pattern can also be adjusted slightly to prevent the voltage saturation of the capacitor.

After all the patterns are learned, the weight generating period starts. As shown in Fig. 4(b), block $VTI3$ converts the voltage stored on capacitor Cw into two absolute currents for the nearest two comparators. At the same time, the signs of the correlations are also been detected. There are four absolute currents from the neighboring RMs in one cell. The comparator generates a mean current of the four absolute currents and compares the four currents with the mean current. The results of the comparisons are counted by block Counter_L to decide the ratio weights of block W. When the $N(4-N)$ currents in neighboring RMs are larger (smaller) than the mean current, the weights of blocks W are set to $1/N(0)$ where N could be 1, 2, 3, or 4. The ratio weights are set at 1/4 for each block W only if the four currents are equal.

In the recognition period, the switches $sw1, sw2, sw5$, and $sw6$ in Fig. 3(b) are closed as shown in Fig. 4(c). The gray level input u_{ij} in Fig. 3(a) of the noisy pattern is sent into block Neuron and the operation of recognition starts.



(a)

(b)

(c)

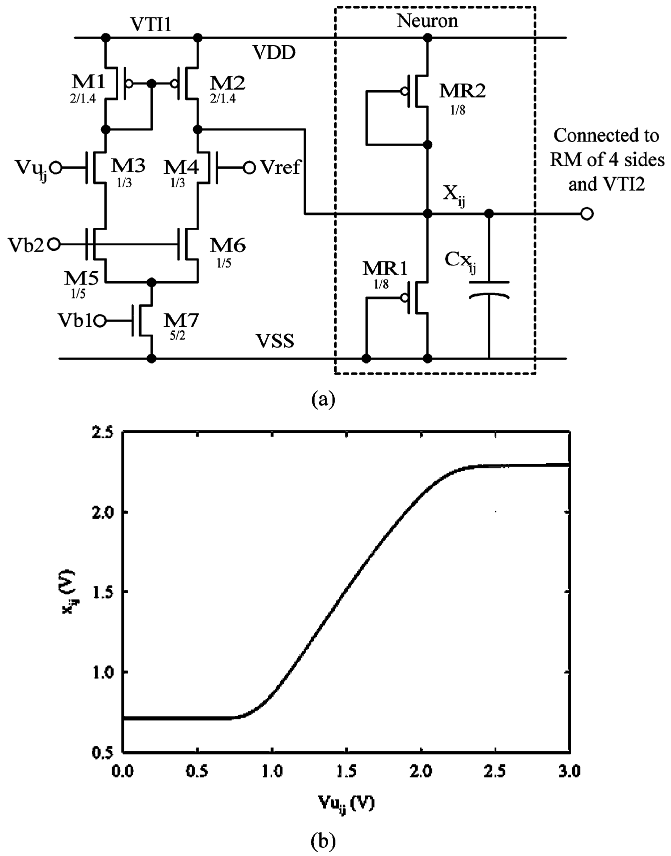
Fig. 4. Flow diagram of the operation during: (a) learning period; (b) weight generating period; and (c) recognition period.

III. Circuit Implementation With Simulation Results

A. Circuit Implementation With Simulation Results

Blocks VTI1 and Neuron are shown in Fig. 5(a). Block VTI1 is constructed by using a simple differential amplifier. M5-M6 are used to degenerate the transconductance of the amplifier and to enlarge the linear operating range. $Vref$ is set to 1.5 V and $Vb2$ at 2.5 V. $Vb1$ is controlled by a current mirror of 5.5 μA. Block Neuron is simply composed of a resistance and a capacitor. The resistance is constructed using $MR1$ and $MR2$, and the capacitor is realized by the parasitic capacitance at node X_{ij}. The transfer characteristic of $VTI1$ is shown in Fig. 5(b). The transfer curve is linear as the input voltage Vu_{ij} is between 0.9 and 2.1 V where the linearity is 94.6% and THD of the output current is 5.9% with the first harmonic of 1 kHz.

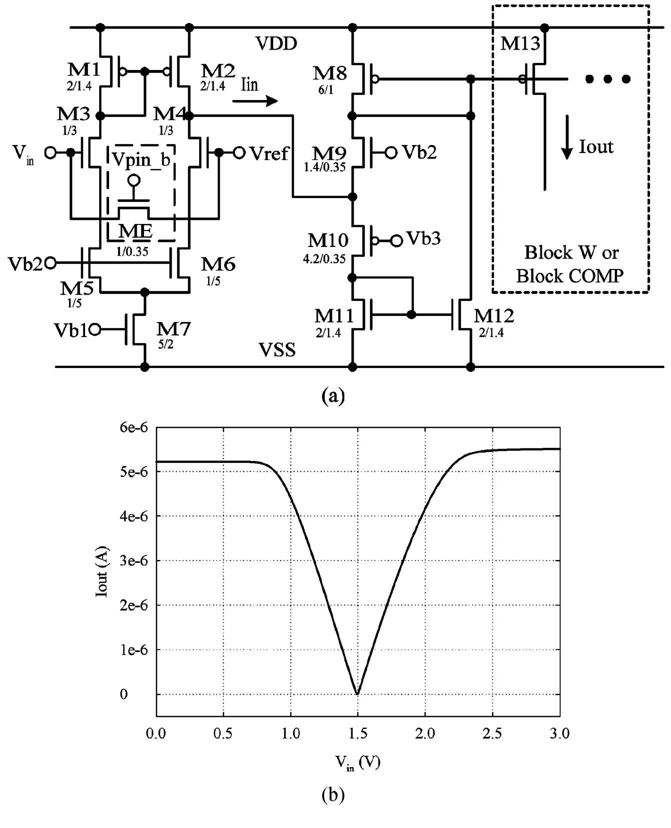
Fig. 6(a) depicts blocks $VTI2$ and $VTI3$. The circuit represented in broken lines belongs to the next stage, block W or block COMP. Block $VTI2$ is similar to block $VTI3$ except



(a)

(b)

Fig. 5. (a) Circuits of blocks VTI1 and Neuron and (b) transfer characteristic of block VTI1 with a linearity of 94.6% in the range between 0.9 V and 2.1 V.



(a)

(b)

Fig. 6. (a) Circuit of blocks $VTI2$ (with ME) and $VTI3$ (without ME) and (b) transfer characteristic of block $VTI2$ (ME is turned off) and $VTI3$.

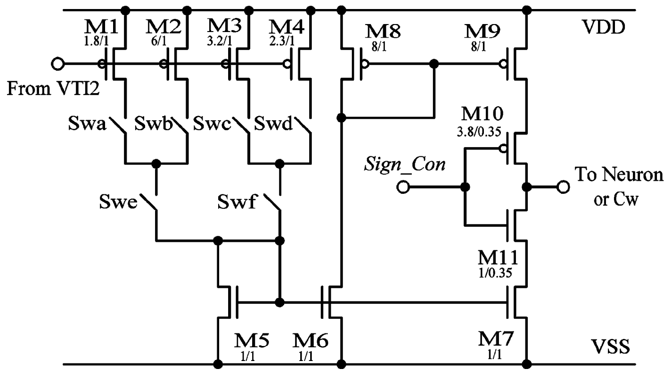


Fig. 7. Circuit of block W where the gain is decided by the ratio of M8 in Fig. 6 and M2, M3, M4, and M1, and controlled by switches Swa-Swf and the sign of output is controlled by signal Sign_Con.

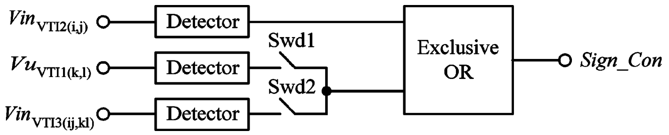


Fig. 8. Block diagram of the sign controller where the detector is composed of two cascaded inverters and the output signal Sing_Con is decided by the inputs of $VTI1$, $VTI2$ and $VTI3$ in different operational steps.

the device ME. $VTI2$ contains ME and $Vpin\_b$ is set to low, when the patterns are learned and during the recognition period, in order to turn on the function of block $VTI2$. $Vb1$ is biased by a current mirror of 5.5 μA and $Vb2$ is set to 2.5 V. $Vb3$ and $Vref$ are each set to 1.5 V. The combination of $M9$ with $Vb2$ and $M10$ with $Vb3$ can stop the static current. The differential amplifiers of $VTI2$ and $VTI3$ are the same with $VTI1$ and the properties of the differential amplifiers are similar. $VTI2$ and $VTI3$ each contain a differential amplifier and an absolute current converter. The differential amplifier generates positive and negative currents based on V_{in} as shown in Fig. 6(b). The positive (negative) current sent to the absolute current converter turns on the device $M10(M9)$ when V_{in} is larger (smaller) than 1.5 V. The positive current is inverted twice with two current mirrors, $M11/M12$ and $M8/M13$. The negative current is inverted once with the current mirror $M8/M13$. Hence, the output current $Iout$ is the absolute current of Iin.

Fig. 7 shows the circuit of block W. Switches Swa-Swf are controlled by block Counter_L to multiply the current with a gain of 1/4, 1/3, 1/2, or 1. Based on the signs of the patterns and learned correlated weights in different periods, signal *Sign_Con* is set to a proper digital code to decide the sign of block W, as shown in Fig. 8. The detector is constructed using two cascaded inverters and amplifies the input signals to achieve a digital level. During the learning period, only switch Swd1 is closed. Signal *Sign_Con* is determined by input voltage V_{in} in block $VTI2$ of cell(i,j) and input voltage Vu of the nearest neighboring cell(k,l) through an exclusive gate. During the recognition period, only switch $Swd2$ is closed. Signal *Sign_Con* is decided by input voltage V_{in} in block $VTI2$ of cell(i,j) and input voltage of block $VTI3$, which is the correlation stored on the capacitor Cw in Fig. 3(b).

The circuit of block COMP is shown in Fig. 9. The up, down, left, and right currents from blocks $VTI3$ are gathered and averaged by the current mirror as in Fig. 10.The mean current is multiplied with a weight of 0.95 to prevent incorrect decision due to the variation of the current mirror. When the absolute current is 0.05 μA less than the mean current, the decision is low after two cascaded inverters. The four directional currents from blocks $VTI3$ are compared with the averaged current I_M and the comparing results are converted into digital signals using two cascaded inverters. The four digital signals are counted by block Counter_L, which is composed of two cascaded D-flip-flops.

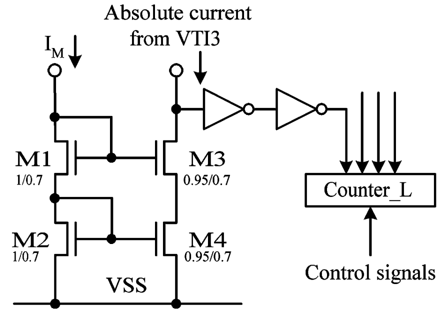


Fig. 9. Circuit of one comparator in block COMP where the output is amplified by two cascaded inverters and the result is sent to block Counter_L.

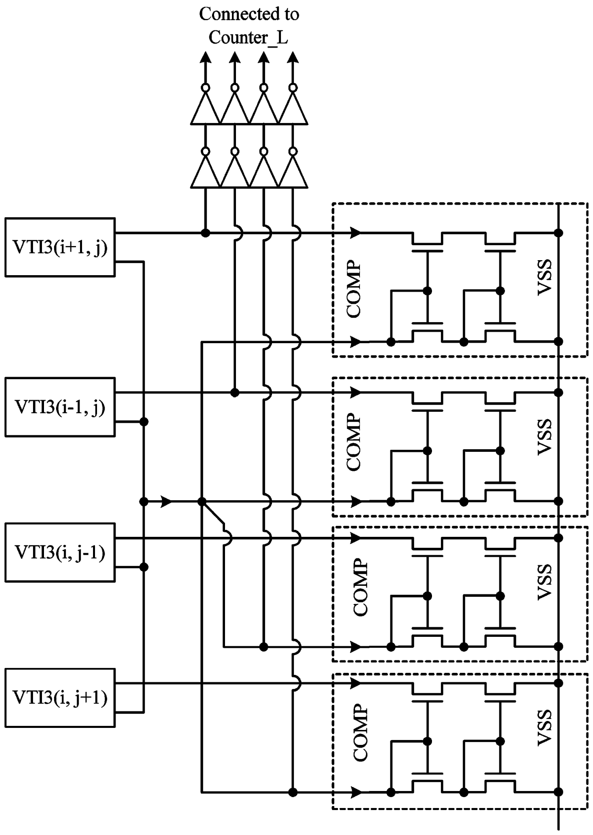


Fig. 10. Circuit of block COMP where the four absolute currents from VTI3 are compared with their mean current, and the results are amplified and sent to block Counter_L.

B. Operational Steps

The operation is separated into three periods: the learning, weight generating, and recognition periods. In the learning period, blocks VTI1, Neuron, VTI2, and W are active. Input

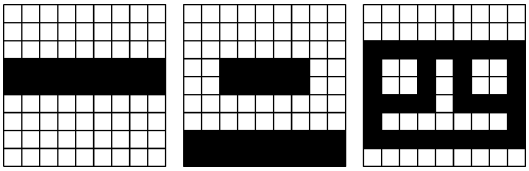


Fig. 11. Input patterns which are learned in the learning period.

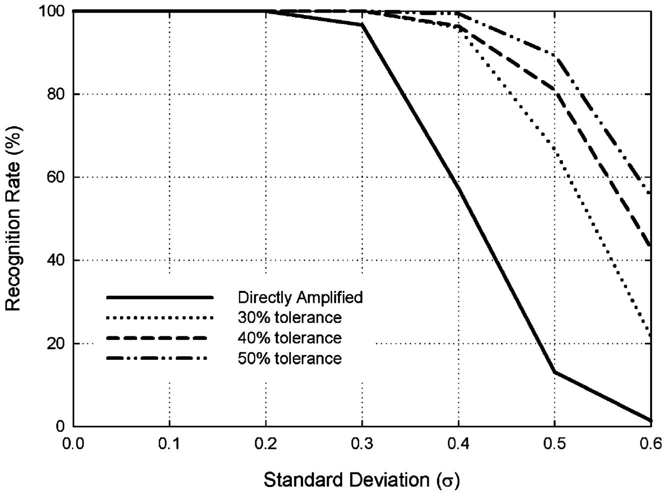


Fig. 12. Recognition rates by direct amplification and using proposed RMCNN with different tolerance of gray level inaccuracy.

TABLE I
COMPARISONS OF TEMPLATES A IN CELL(4, 5), CELL(5, 3), CELL(8, 5), AND CELL(7, 5) BETWEEN RMCNN WITH AND WITHOUT ELAPSED TIME

Templates A	With Elapsed Time	Without Elapsed Time
9 x 9 RMCNN r = 1	$A_{4,5}(800s)= \begin{bmatrix} 0 & -0.5 & 0 \\ 0 & 0 & 0 \\ 0 & 0.5 & 0 \end{bmatrix}$	$A_{4,5} = \begin{bmatrix} 0 & -0.5 & 0 \\ 0 & 0 & 0 \\ 0 & 0.5 & 0 \end{bmatrix}$
	$A_{5,3}(800s)= \begin{bmatrix} 0 & 0.944 & 0 \\ 0.018 & 0 & 0.018 \\ 0 & -0.018 & 0 \end{bmatrix}$	$A_{5,3} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
	$A_{8,5}(800s)= \begin{bmatrix} 0 & 0 & 0 \\ 0.5 & 0 & 0.5 \\ 0 & 0 & 0 \end{bmatrix}$	$A_{8,5} = \begin{bmatrix} 0 & 0 & 0 \\ 0.5 & 0 & 0.5 \\ 0 & 0 & 0 \end{bmatrix}$
	$A_{7,5}(800s)= \begin{bmatrix} 0 & 0.33 & 0 \\ 0.33 & 0 & 0.33 \\ 0 & 0 & 0 \end{bmatrix}$	$A_{7,5} = \begin{bmatrix} 0 & 0.33 & 0 \\ 0.33 & 0 & 0.33 \\ 0 & 0 & 0 \end{bmatrix}$

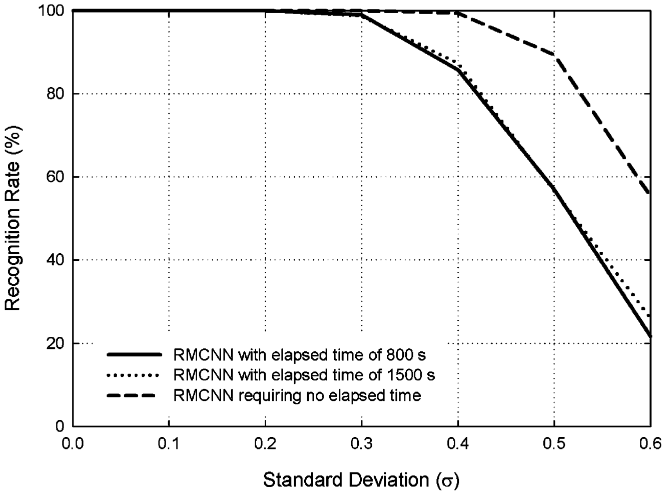


Fig. 13. Recognition rates of RMCNN with elapsed time of 800 and 1500 s, and RMCNN requiring no elapsed time.

voltage Vu_{ij} of the learned pattern is transferred into the current signal and the current signal is applied on block Neuron to produce an output state voltage Vx_{ij}. With state voltage Vx_{ij}, block VTI2 generates the absolute current and this is multiplied by 1/4 through controlling the switches Swa-Swf. The polarity of the output current of block W is controlled by signal *Sign_Con* which is generated by the sign controller in Fig. 6 where only switch Swd1 is closed. With the output current of block W, capacitor Cw is charged or discharged in an interval Tp. After all the patterns are learned, the weight generating period starts.

In the weight generating period, the switches in Fig. 3(b) are all open. The voltage on Cw is applied to $VTI3$ to generate two absolute currents while the sign is also detected, simultaneously. With four absolute RM currents in four directions, mean current I_M is generated and compared with the four absolute currents as shown in Fig. 10. The four comparators outputs are counted by block Counter_L, which sends the control signal to four blocks W to generate corresponding weights. As a result, the RMCNN is ready for recognition. In the third period, the noisy patterns are sent into the RMCNN. Switches $sw1$, $sw2$, $sw5$, and $sw6$ in Fig. 3(b) are closed and so is $Swd2$ in Fig. 8. At the same time, device ME in Fig. 6(a) is turned off to commence the operation of recognition. A set of patterns in Fig. 11 are learned with Matlab by the proposed algorithm of a 9×9 RMCNN. The resultant recognition rate is shown in Fig. 12 which is compared by directly amplifying the noisy patterns with an inverter. As can be seen from Fig. 12, when the tolerance level is 50%, the recognition rate is better than that of direct amplification. However, when the tolerance is sterner, the recognition rate is reduced. Due to the fact that template A is a non-self-feedback template, the recognized output patterns can not be pulled to a saturated state and, hence, the recognition rate is degraded. The template values of cell(4, 5), cell(5, 3), cell(8, 5), and cell(7, 5) are listed in Table I and are compared with RMCNN [16] with an elapsed time of 800 sec. As can be seen in Table I, the templates are almost the same except for some negligible coefficients. The comparison of recognition rates are also shown in Fig. 13. It shows that RMCNN has similar recognition rates after 800 seconds and 1500 seconds while the recognition rate of RMCNN requiring no elapsed time is better than RMCNN with elapsed time.

The recognition rate is shown in Fig. 14 where the self-feedback RMCNN without elapsed time is simulated and the result is compared with that of direct amplification. Since the closed loop in the self-feedback RMCNN saturates the output, the recognition rate can be raised and is better than that without self-feedback and that of direct amplification. In this paper, only the test

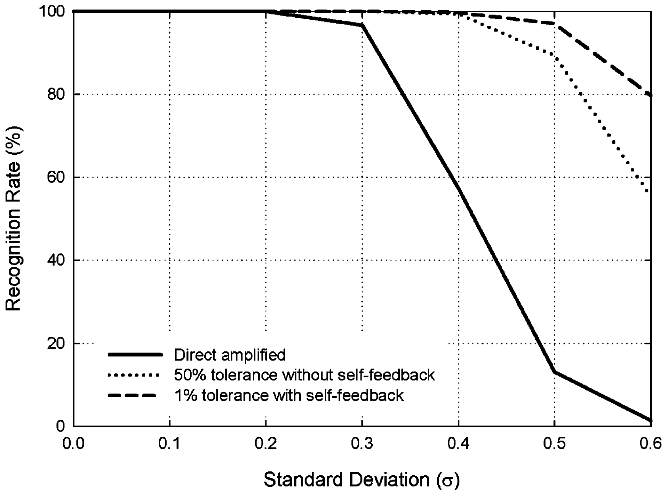


Fig. 14. Recognition rates by direct amplification, using the proposed RMCNN with self-feedback, and using the proposed RMCNN without self-feedback of 50% tolerance.

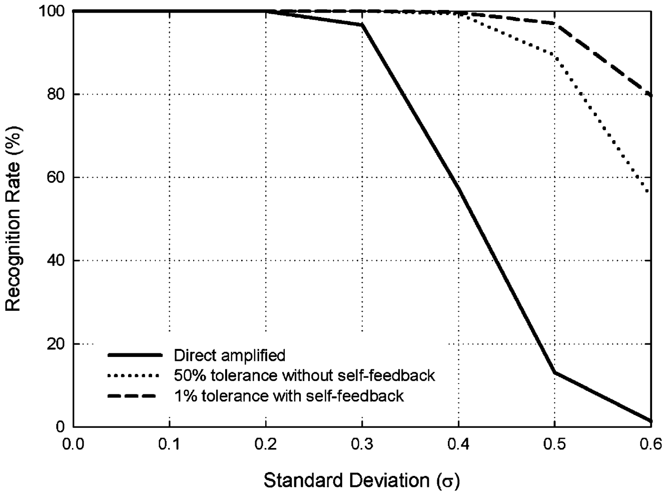


Fig. 15. Recognition rates of common time constant and time constant varying with Gaussian probability density of standard deviation 0.1 where the common time constant is normalized to 1.

chip of the non-self-feedback RMCNN without elapsed time is designed and measured to verify the proposed RMCNN algorithm not requiring elapsed time. A self-feedback RMCNN not requiring elapsed time can be designed similarly.

Furthermore, the capacitance of neural RC model is implemented by the parasitic capacitance and it may cause large variation of time constant. However, CNNs essentially have great tolerance to variations. Fig. 15 shows the recognition rates of common time constant and time constant varying with Gaussian probability density of standard deviation 0.1 where the common time constant is normalized to 1. It is found that, even with 10% standard deviation, the recognition rate is almost the same.

With the proposed structure, the patterns of characters "X", "Y", and "Z" and characters "X", "Y", and "M", which contains oblique lines, are also learned and recognized. The recognition rates are shown in Fig. 16 and it shows that the recognition rates degrade and are dependent on learned patterns. To learn patterns with oblique lines correctly, more RMs can be inserted between diagonal cells.

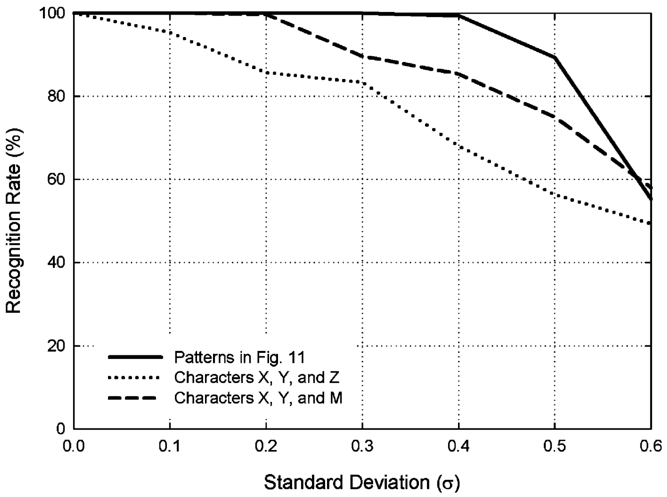


Fig. 16. Recognition rates where the patterns in Fig. 11, characters X, Y, and Z, and characters X, Y, and M are learned and recognized where the patterns of the characters contain many oblique lines.

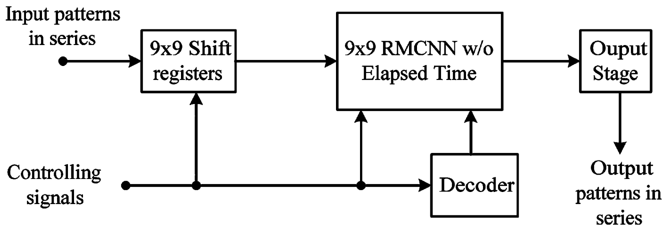


Fig. 17. Architecture of 9 × 9 RMCNN not requiring elapsed time where contains 9×9 shift registers for image storage, decoder and output stage for readout, and 9×9 RMCNN requiring elapsed time for operation.

IV. Experimental Results and Discussion

To verify the concept of RMCNN requiring no elapsed time, a simple experimental chip is designed and fabricated. The experimental RMCNN chip has only RMs located in any two horizontal or vertical cells. It is of the architecture of 9 × 9 type as shown in Fig. 17. The input patterns for learning and recognition are sent serially into 9 × 9 shift registers. The decoder can select the cells in the proposed RMCNN not requiring elapsed time to be read out in series. The controlling signals are listed in Table II with a controlling timing diagram shown in Fig. 18. The learning and recognition periods are controlled by signals *clk1* and *clk2*, respectively. Signal *Reset* is used to reset the charge on the capacitor Cw. Signal *newp* enables the shift registers and then, signal *DFF* can trigger the D-flip-flops in the shift registers to transfer the pixels of the input pattern in series. Signal *pin* generates the patterns which are sent into the neural network. Signals *Con_L* and *Con_G* trigger the local and global counters. The local counter counts the number of the currents which are larger than the mean current in the cell. The global counter generates the signals to control which comparative results in the cell should be counted by the local counter. Signal *noi* can introduce the noise into the input patterns. With such architecture, an RMCNN chip not requiring elapsed time has been designed and fabricated using TSMC 0.35-μm 2P4M mixed-signal technology. Fig. 19 shows the photograph of the fabricated chip of an RMCNN not requiring elapsed time.

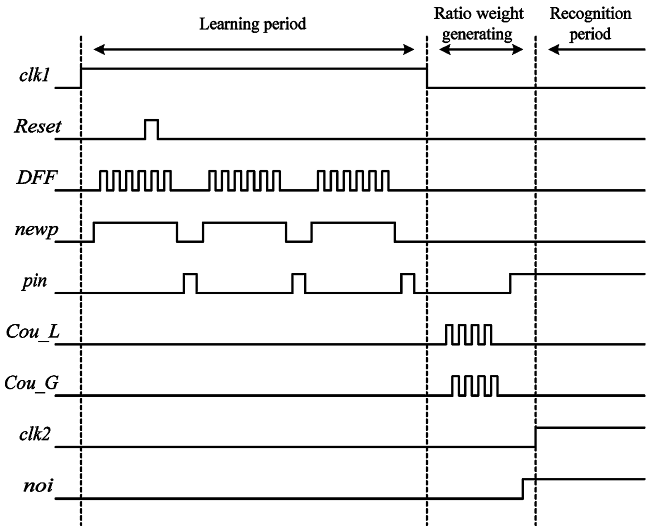


Fig. 18. Timing diagram of the control signals where the explanation of each signal is list in Table II.

TABLE II
DESCRIPTIONS OF EACH CONTROL SIGNAL

Control signal	Description
clk1	High : Learning period starts Low : Learning period stops
Reset	High : Reset the capacitor in the block RM Low : Normal Operation
DFF	Trigger the D-flip-flop in shift register (negative triggered) used to stored the patterns.
newp	High : Enable the transfer of the shift register Low : Disable the transfer of the shift register
pin	High : The pattern in shift register is sent to neural cells
Cou_L	Trigger the block Counter_L in each cell
Cou_G	Trigger the global counter for controlling the blocks COMP and Counter_L
clk2	High : Recognition period starts Low : Recognition period stops
noi	High : Make the pattern in shift register noisy

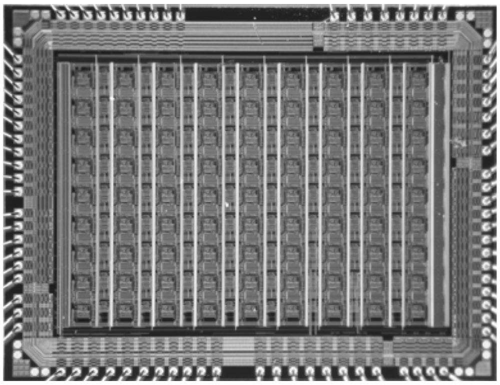


Fig. 19. Photograph of the proposed RMCNN not requiring elapsed time chip.

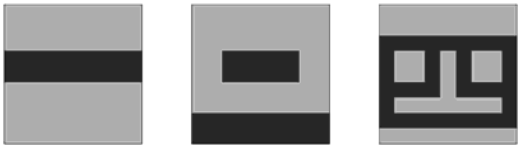


Fig. 20. Noise patterns for measurement with an equal noise level of 0.5.

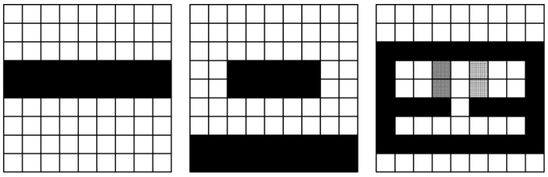


Fig. 21. Experimental results of the recognized patterns in the recognition period after the recognition of a set of patterns with equal noise level 0.25 are recognized.

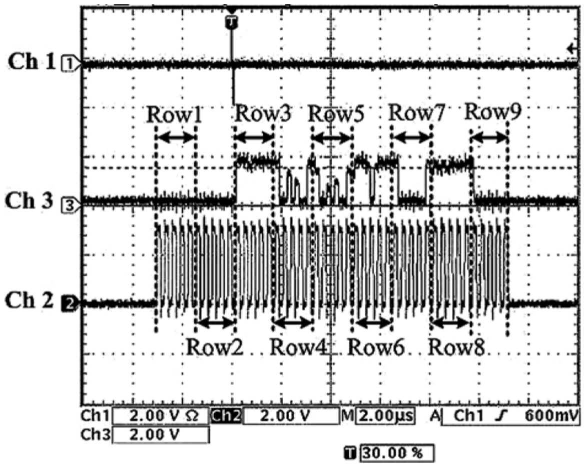


Fig. 22. Experimental output waveform of the third recognized pattern where $Ch1$ is the trigger signal, $Ch2$ is the LSB of decoder, and $Ch3$ is the output signals of the third patterns in Fig. 21.

During the learning period, three Chinese characters in Fig. 11 are learned. With 9×9 RMCNN structure, the maximum number of patterns that can be learned is 3 by using Matlab simulation. However, 18×18 RMCNN is also simulated and the maximum number of learnable patterns can be raised to 5. After the ratio weights are generated, these Chinese characters are sent again and combined with a controllable equal noise from 0 to 0.5 as shown in Fig. 20 where the noise level is set to 0.5. The noise pattern cannot be programmed individually on each pixel because the number of pads is limited. Hence, the pattern with equal noise is used in the experimental work. For the first two Chinese characters, the correct patterns could be recognized. However, the last Chinese character is recognized unsuccessfully as shown in Fig. 21 where the equal noise level is 0.25, and the output waveform is shown in Fig. 22. $Ch1$ is a trigger signal and there is no signal during the readout period. $Ch2$ is LSB of the decoder to choose which cell is read out and $Ch3$ is the output signals of cells. In Fig. 22, the output waveform of $Ch2$ is the recognized result of third pattern in Fig. 20. However, some outputs of $Ch3$ in the intervals of $Row4$ and $Row5$ cannot reach a saturated point. There are four stable pixels at the gray level as the third pattern in Fig. 20 is recognized. To discuss the reason for this,

TABLE III
THE COMPARISON OF THE ABSOLUTE WEIGHTS A44 WITH MATLAB AND HSPICE IN DIFFERENT CONDITIONS

Simulator (C ondition)	Absolute weight of cell(4,4)	Mean	ratio weight of cell(4,4)
Matlab	$A_{44}=\begin{bmatrix}0 & 0.33 & 0\\ 0.33 & 0 & 0.33\\ 0 & 1 & 0\end{bmatrix}$	0.5	$A_{44}=\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}$
HSPICE (TT)	$A_{44}=\begin{bmatrix}0 & 0.15 & 0\\ 0.28 & 0 & 0.28\\ 0 & 0.78 & 0\end{bmatrix}$	0.3725	$A_{44}=\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}$
HSPICE (FS)	$A_{44}=\begin{bmatrix}0 & 0.116 & 0\\ 0.41 & 0 & 0.41\\ 0 & 0.70 & 0\end{bmatrix}$	0.409	$A_{44}=\begin{bmatrix}0 & 0 & 0\\ 0.33 & 0 & 0.33\\ 0 & 0.33 & 0\end{bmatrix}$

TABLE IV
THE COMPARISON BETWEEN RMCNN[16] AND RMCNN REQUIRING NO ELAPSED TIME

	RMCNN [15]	RMCNNrequiring no elapsed time
Technology	0.35 mm 1P4M Mixed-Signal Process	0.35 mm 2P4M Mixed-Signal Process
Array Size	9 x 9	9 x 9
No. of RMs	144	144
Area of Single Pixel	350μm x 350μm	400μm x 250μm
Chip Area	3800μm x 3900μm	4560μm x 3900μm
Power Supply	3 V	3 V
Power Dissipation	120 mW	87 mW
Readout Time (of one pixel)	1 ms	80 ns
Weight Generating Time (3 patterns are learned)	850 sec	1.7μs

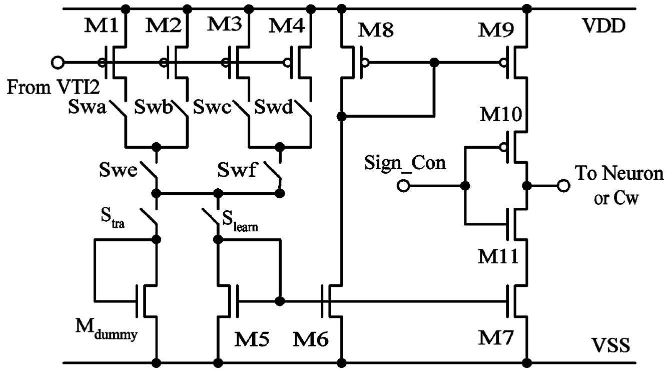


Fig. 23. Modified circuit of block W where switch Stra and Mdummy are added to prevent charging Cw when RMCNN is loading the learned patterns.

the absolute weights of the post simulation at $cell(4,4)$, which is recognized unsuccessfully in the third pattern, are listed in Table III where the simulation with Matlab and two post simulations with HSPICE in different conditions are compared. RMCNN requiring no elapsed time is designed with HSPICE in the TT condition. As can be seen, the ratio weights with Matlab and HSPICE (TT) are the same and the noisy patterns can be recognized correctly. However, with HSPICE (FS), an incorrect ratio weight is generated and leads to an unsuccessful result. When three patterns are learned, block W charges or discharges the capacitor Cw according to the input pixel of two neighboring cells. During this time, the device ME in Fig. 6(a) of block $VTI2$ is turned off. However, when a new pattern is sent into the chip after the former one is learned, device ME is turned on and the output current should be 0. However, there is still a small output current due to the asymmetric structure and the mismatch. Meanwhile, the input of the sign detector is connected to $Vref$ because device ME is turned on and this makes it impossible to predict signal *Sign_Con*. As a result, the capacitor Cw is charged or discharged unpredictably by the small current when the learned patterns are transmitted to the 9×9 shift registers. Hence, the ideal absolute weights cannot be achieved.

To overcome the small output current from block $VTI2$, a new path can be inserted into block W, as shown in Fig. 23. Only one of switches $S_{\rm tra}$ and $S_{\rm learn}$ is turned on and the other is turned off. As the learned patterns are transmitted to the shift register, switch $S_{\rm tra}$ is turned on. Hence, capacitor Cw would not be charged or discharged by the small current from block VTI2. Switch $S_{\rm learn}$ is turned on when the pattern in the shift register is sent to the neuron and can be learned or recognized correctly. Dummy load $M_{\rm dummy}$ is the same with M5. This can cause the current source M1-M4 to have a similar load and retain the current stable during switching.

The comparison between RMCNN [16] and RMCNN requiring no elapsed time is listed in Table IV. The overall chip area of RMCNN requiring no elapsed time is slightly larger than RMCNN with elapsed time. However, the power consumption is smaller because the multi-divider is replaced by a simple comparator and counter. Furthermore, the required weight generating time of 850 seconds is reduced to 1.7 μs.

V. CONCLUSION

In this paper, a new algorithm of a RNCNN not requiring elapsed time has been proposed. In the proposed RMCNN, a new ratio weight generating method is also proposed. The use of this method avoids long elapsed time about 800 seconds while retaining the correlation of each cell. An experimental chip of 9 × 9 RMCNN not requiring elapsed time has been implemented and fabricated using TSMC 0.35-μm CMOS 2P4M technology. With the proposed RMCNN chip not requiring elapsed time, the patterns can be learned and recognized. Further applications of the proposed RMCNN not requiring elapsed time will be developed in the future.

ACKNOWLEDGMENT

The authors would like to thank the Chip Implementation Center (CIC), National Science Council, Taiwan, for the fabrication of the testing chip.

REFERENCES

[1] L. O. Chua and L. Yang, "Cellular neural networks: Theory," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 1257–1272, Oct. 1988.

[2] L. O. Chua and L. Yang, "Cellular neural networks: Applications," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 1273–1290, Oct. 1988.

[3] L. O. Chua, *CNN: A Paradigm for Complexity*, ser. World Scientific Series on Nonlinear Science. Singapore: World Scientific, 1998, vol. 31.

[4] D. Farmer, T. Toffoli, and S. Wolfrman, "Cellular automata," in *Proc. Interdisciplinary Workshop*, New York, 1984.

[5] T. Toffoli*, Cellular Automata Machines: A New Environment for Modeling*, ser. Series in Scientific Computation. Cambridge, MA: MIT Press, 1987.

[6] T. Roska and L. O. Chua, "The CNN universal machine: An analogic array computer," *IEEE Trans. Circuits Syst II, Analog Digit. Signal Process.*, vol. 40, pp. 163–173, Mar. 1993.

[7] T. Roska, "Analogic algorithms running on the CNN universal machine," in *Proc. 3rd IEEE Int. Workshop Cellular Neural Networks Appl. (CNNA-94)*, Rome, Italy, Dec. 1994, pp. 3–8.

[8] S. Espejo, R. Carmona, R. Domínguez-Castro, and A. R.-. Vázquez, "A CNN universal chip in CMOS technology," *Int. J. Circuit Theory and Appl.*, vol. 24, pp. 93–109, Mar. 1996.

[9] R. Domínguez-Castro, S. Espejo, A. Rodríguez-Vázquez, R. A. Carmona, P. Foldesy, A. Zarandy, P. Szolgay, T. Sziranyi, and T. Roska, "A 0.8-μm CMOS two-dimensional programmable mixed-signal focal-plane array processor with on-chip binary imaging and instructions storage," *IEEE J. Solid-State Circuits*, vol. 32, no. 7, pp. 1013–1026, Jul. 1997.

[10] K. Nakamura, K. Arimura, and T. Yoshikawa, "Recognition of object orientation and shape by a rotation spreading associative neural network," in *Proc. Int. Joint Conf. on Neural Netw., (IJCNN '01)*, Jul. 15–19, 2001, vol. 1, pp. 565–570.

[11] M. Namba and Z. Zhang, "Cellular neural network for associative memory and its application to braille image recognition," in *Proc. Int. Joint Conf. on Neural Netw., (IJCNN '06)*, Jul. 16–21, 2006, pp. 2409–2414.

[12] E. David, P. Ungureanu, and L. Goras, "On the feature extraction performances of CNN Gabor-type filters in texture recognition applications," in *10th Int. Workshop on Cellular Neural Netw. and Their Appl., (CNNA '06)*, Aug. 28–30, 2006, pp. 1–6.

[13] K. H. Lim, K. P. Seng, L.-M. Ang, and S. W. Chin, "Lyapunov theory-based multilayered neural network," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 4, pp. 305–309, Apr. 2009.

[14] C.-Y. Wu and J.-F. Lan, "CMOS current-mode neural associative memory design with on-chip learning," *IEEE Trans. Neural Netw.*, vol. 7, no. 1, pp. 167–181, 1996.

[15] J.-F. Lan and C.-Y. Wu, "CMOS current-mode outstar neural networks with long-period analog ratio memory," in *Proc. IEEE Int. Symp. on Circuits and Syst.*, 1995, vol. 3, pp. 1676–1679.

[16] C.-Y. Wu and C.-H. Cheng, "A learnable cellular neural network structure with ratio memory for image processing," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 49, no. 12, pp. 1713–1723, Dec. 2002.

[17] S. Grossberg, "Nonlinear difference-differential equations in prediction and learning theory," in *Proc. Natl. Acad. Sci. USA*, 1967, vol. 58, pp. 1329–1334.

[18] J. A. Feldman and D. H. Ballard, "Connectionist models and their properties," *Cognitive Science*, vol. 6, pp. 205–254, 1982.

[19] S. Grossberg*, The Adaptive Brain I: Cognition, Learning, Reinforcement, and Rhythm*. Amsterdam, the Netherlands: Elsevier/North-Holland, 1986.

[20] W. Chung-Yu and W. Yu, "The design of CMOS nonself-feedback ratio memory cellular nonlinear network without elapsed operation for pattern learning and recognition," in *2005 9th Int. Workshop on Cellular Neural Networks and Their Appl.*, May 28–30, 2005, pp. 282–285.

Chung-Yu Wu (S'76–M'76–SM'96–F'98) was born in 1950. He received the M.S. and Ph.D. degrees in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, , in 1976 and 1980, respectively.

Since 1980, he has been a consultant to high-tech industry and research organizations and has built up strong research collaborations with high-tech industries. From 1980 to 1983, he was an Associate Professor with National Chiao Tung University. From 1984 to 1986, he was a Visiting Associate Professor with the Department of Electrical Engineering, Portland State University, Portland, OR. Since 1987, he has been a Professor with National Chiao Tung University, Hsinchu, Taiwan. From 1991 to 1995, he served as the Director of the Division of Engineering and Applied Science, National Science Council, Taiwan. From 1996 to 1998, he was honored as the Centennial Honorary Chair Professor with National Chiao Tung University. He is currently the President and Chair Professor of National Chiao Tung University. In summer 2002, he conducted Post-Doctoral research with the University of California at Berkeley. He has authored or coauthored over 250 technical papers in international journals and conferences. He holds 19 patents, including nine U.S. patents. His research interests are nanoelectronics, biomedical devices and system, neural vision sensors, RF circuits, and computer-aided design (CAD) and analysis.

Dr. Wu is a member of Eta Kappa Nu and Phi Tau Phi. He was a recipient of 1998 IEEE Fellow Award and a 2000 Third Millennium Medal. He has also been the recipient of numerous research awards presented by the Ministry of Education, National Science Council (NSC), and professional foundations in Taiwan.

Sheng-Hao Chen was born in ChangHua, Taiwan, in 1980. He received the B.S. degree from the Department of Electronics Engineering, National Chiao Tung University, Hsinchu, Taiwan, in 2002, and the M.S. and Ph.D. degrees in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2003 and 2009, respectively.

Recently, he is working with MediaTek on the analog integrated circuit design.

Yu Wu was born in Taipei, Taiwan, in 1981. He received the B.S. degree from the Department of Electrical Engineering, National Central University, Jhongli, Taiwan, in 2003 and M.S. degree in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2004.

Recently, he is working in eMemory Company about the analog integrated circuit design.