CHAPTER 5

VLSI IMPLEMENTATION OF THE SRMCNN WITH ON-CHIP LEARNING AND STORAGE

5.1 INTRODUCTION

Due to the advantageous feature of local connection and simple synaptic operations, the cellular nonlinear (neural) network (CNN) introduced by Chua and Yang [55] is very suitable for VLSI implementation in many high-speed, real-time applications [56]-[58]. So far, some research works on the applications of CNNs as neural associative memories for pattern learning, recognition, and association have been explored [59]-[64], [76]-[78], [80]-[84], [102],. Among them, many innovative algorithms and software simulations of CNN associated memories were reported [76]-[78], [94], [102]. As to the hardware implementation, special learning algorithm and digital hardware implementation for CNNs were proposed in [161] to solve the sensitivity problems caused by the limited precision of analog weights. Moreover, CMOS VLSI design and chip implementation of CNN associative memory was also reported in [171], [174], [182].

In realizing CNN associative memories, the learning circuitry can be integrated on-chip with CNNs [171], [174]. The major advantages of on-chip learning [64] are: 1) No needed the host computer to perform the learning task for off-line operation. This makes the interface of neural system chips simple for many practical applications; 2) The spatial-variant template weights can be on-chip learned without being loaded from outside to the CNN chips. Thus long loading time, complex cell global interconnection, and analog weight storage elements to perform the loading operation for large numbers of spatial-variant template weights can be avoided; 3) The adaptability to the process variations of CNN chips can be enhanced.

In the thesis, the structure of the SRMCNN with **B** template and the modified Hebbian learning algorithm for auto-associative memory are proposed. The function blocks have implemented in the VLSI circuits for the TSMC 0.25 μm 1P5M n-well CMOS technology. The characteristic of the proposed circuits are correctly verified by the HSPICE software. The function of ratio memory for one bit SRMCNN with **B** template was realized in the VLSI chip and their operation was shown. The simulation results of the 18x18 SRMCNN behavior and function are demonstrated and analyzed. The capability of pattern learning and recognition is improved. The chip layout of the 9x9 SRMCNN was implemented. The conceptual design for the general architecture of the Large-Neighborhood Cellular Nonlinear (Neural) Network Universal Machine (LN-CNNUM) is described [157]-[162], [179]-[182].

The chapter is organized as follows. In Section 5.2, the model and architecture of the SRMCNN with the modified Hebbian learning algorithm for hetero-associative memory is presented. The VLSI circuits of function blocks and the characteristic of the proposed circuits are shown in Section 5.3. The learning and the embedded ratio memory operation for the **B** templar in the VLSI architecture of the SRMCNN are proposed. In Section 5.4, the VLSI chip implementation for one bit SRMCNN with **B** template in ratio memory is described. The simulation results of the 18x18 SRMCNN with **B** template are demonstrated and analyzed. The chip layout of the 9x9 SRMCNN is implemented and verified. The conceptual design for the general architecture of the Large-Neighborhood Cellular Nonlinear (Neural) Network Universal Machine (LN-CNNUM) is shown in Section 5.5. Finally, summaries are drawn.

5.2 MODEL AND ARCHITECTURE

The detailed block diagram of the two neighboring cells and their RM in the SRMCNN with **B** template is shown in Fig. 5.1. The detailed block diagram of the S block during the learning period and the recognition period are shown Fig. 5.2 (a) and (b), respectively. In Fig. 5.1, the block T1 is a V-I converter with activation function used to convert the voltage of the state into the output current of the cell. The block T2D is a V-I converter with a one-half absolute-value circuit and a sign-detection circuit to generate the absolute value of input current and detect the sign of the cell input $u_{ij}(t)$, respectively. The CNN cell *C(i,j)* is consisted with T1, R_{ij}, and C_{ij} elements [138], [166]-[172].

The block M/D [171] in Fig. 5.1, is a combined four-quadrant multiplier and a two-quadrant divider circuit, is used to generate the weights of **B** template for the modified Hebbian learning algorithm during the learning period [103], [136]. It is also used to multiply b_{ijkl} (t) by u_{kl} (t) and the ratio with the summation for five absolute neighboring weights in the recognition period. The resultant weight $Vzi_{\;ijkl}$ during the learning period is stored in the capacitor *Czi*, transfers the absolute voltage *VCzs* stored to *Czs* and stores its sign in the latch circuit in the S block of Fig. 5.2(a). Block T3 is also a V-I converter to convert the voltage of *Czs* into current. The output current of T3 is sent to the sum block and summed with the weight currents from neighboring cells. The summed current is sent to the M/D block to generate ratio-memory [171].

The *m* exemplar patterns are input in order read into the cell *C(i, j)* and the input voltage Vu_{ij}^{p} of the *p*-th input pattern is sent to T2Dto be converted into current $\mathrm{I}u_{ij}^{p}$. The desired output currents $\mathrm{I}y_{ij}^{p}$ are obtained by the decoder circuit D decoding the input p_sel corresponding to the value of m. Then the converted absolute currents $\left|\mathrm{I}y_{ij}^{p}\right|$ and

$\left|\mathrm{I}u_{kl}^{p}\right|$ from the neighboring cells with two the detected sign Vsu_{kl} and Vsy_{ij} are distinctly sent to the four-quadrant multiplier in the M/D block to charges the capacitor *Czi* for the period T_P. This operation is repeated for *m* patterns to accumulate the produced voltages in *Czi*. Finally, the weight voltage $Vzi_{ijkl}\,(0)$ stored on *Czi* at t = 0 when the learning period ends, can be written as

$$\mathrm{V}zi_{ijkl}\left(0\right)=\frac{1}{\mathrm{C}zi}\sum_{p=1}^{m}\left(\int_{T_P}\frac{\mathrm{I}y_{ij}^{p}\mathrm{I}u_{kl}^{p}}{\mathrm{Ib}}dt\right)\qquad C\left(k,l\right)\in N_{r}\left(i,\,j\right)\qquad(5.1)$$

where $\mathrm{I}y_{ij}^{p}$ is the current of the *p*-th desired output of the cell *C(i, j)*, $\mathrm{I}u_{kl}^{p}$ is the current of the *p*-th pattern sent to the cell *C(k, l)* of $N_r\,(i, j)$ neighboring cells, Ib is a constant bias current, $\mathrm{V}zi_{ijkl}\,(0)$ is the weight W_{ijkl} voltage stored on *Czi* at t = 0 sec, and T_P is the learning time of each input pattern.

The weight $W_{ijkl}(0)$ is directly generated by the current product of $\mathrm{I}y_{ij}^{p}\mathrm{I}u_{kl}^{p}$ charging on the capacitor *Czi*, $\mathrm{V}zi_{ijkl}\,(0)$ is also generated for the period *Tp*. After the learned time, the absolute value of $\mathrm{V}zi_{ijkl}\,(0)$ is transferred and stored on the capacitor *Czs*, as shown in Fig. 5.2(a).

In the elapsed period, the configuration of the S block is shown in Fig. 5.2(b), where *Czs* is disconnected from the block T2L. The leakage current $I_{leakage}$ associated with *Czs* gradually decreases $abs\,[\mathrm{V}zs_{ijkl}\,(0)]$ of *Czs* and stored the voltage after elapsed time to provide the ratio memory calculation.

In the recognition period, the voltage $\mathrm{V}u_{ij}^{t}$ of the test pattern is input to T2D and converted into the pixel current $\mathrm{I}u_{ij}^{t}$ and the sign voltage Vsx_{ij}. The absolute weight voltage $abs[Vzs_{ijkl}\,(t)]$ stored on Czs is converted into the current $abs[Izs_{ijkl}(t)]$ through T3 and summed with the currents from the other neighboring cells. The summed current, the weight current *abs*[$Izs_{ijkl}(t)$], and the cell input current $I_{ukl}(t)$ are sent to the M/D block to yield the current that corresponds to the term $b_{ijkl}(t)\ u_{kl}(t)/ \sum\limits_{C(k,l)\in N_r(i,j)} abs[b_{ijkl}(t)]$ in (5.2). The summed currents from other neighboring weights of **B** template, the input current $\mathrm{I}u_{ij}^{t}$, the cell output current $\mathrm{I}y_{ij}^{t}$, and the threshold current $\mathrm{I}z_{ij}$ to generate the cell state current $\mathrm{Ix}_{ij}(t)$. The current $Ix_{ij}(t)$ is converted into the voltage $Vx_{ij}(t)$ through resistor R_{ij}. Thus, $Vx_{ij}(t)$ can be expressed as

$$\mathrm{Vx}_{ij}(t) = \mathrm{R}_{ij}\left(\sum_{C(k,l)\in N_r(i,j)} \mathrm{K}_{\mathrm{A}} \frac{\mathrm{I}_{u_{kl}}(t)\mathrm{Iz}s_{ijkl}(t)}{\sum\limits_{C(k,l)\in N_r(i,j)} abs[\mathrm{Iz}s_{ijkl}(t)]} + \mathrm{I}y_{ij}^{t} + \mathrm{Iz}_{ij} \right) \qquad (5.2)$$

where K_{A} is the empirical gain. Ideally $\mathrm{K}_{\mathrm{A}} = 1$.

The ratioed weight $\mathrm{Iz}s_{ijkl}(t) \Big/ \sum\limits_{C(k,l)\in N_r(i,j)} abs[\mathrm{Iz}s_{ijkl}(t)]$ in (5.2) is generated by the two-quadrant divider in the M/D block with its sign equal to the sign of $Izs_{ijkl}(t)$ latched in T2L, whereas the I_{uij} *(t)* is multiplied by the ratioed weight by the four-quadrant multiplier of M/D using the latched sign of $Izs_{ijkl}(t)$ and the sign of $u_{kl}(t)$ in T2D. The current of input patterns is summed with the five weighted outputs from neighboring cells during the recognition period and converted into a voltage through the resistor R_{ij} and the parasitic capacitor C_{ij} to form the cell state $X_{ij}(t)$. The generated $Vx_{ij}(t)$ is sent to T1 to generate the output current I_{Yij} (t).

5.3 CMOS CIRCUIT REALIZATION

5.3.1 V-I Converters and Sign Detectors

The CMOS circuits of T2 is shown in Fig. 5.3 [171], where Fig. 5.3(a) show the circuits of V-I converter with the absolute-value circuit. The V-I converter which is also used in the blocks T1 and T3 is a CMOS differential amplifier M1~M7 with the source resistance to increase the linear range. The two source resisters are realized by M5 and M6 devices operated in the linear region with the gate bias voltage *Vbvic1*. The output current *Iovic* is sent to the absolute-value circuit formed by M8~M13 to generate the absolute-value current *Ioabs* with the unified flow direction. The bias *Vbvic1*, *Vbvic*, *Vbabsn*, *Vbabsp* are constant voltages.

The generated $Vx_{ij}(t)$ is sent to T1 to generate the current $Iy_{ij}(t)$ and $sign[\mathrm{I}y_{ij}(t)]$ as

$$\mathrm{I}y_{ij}(t) = \begin{cases} \mathrm{Gm2d}\ \ \mathrm{V}x_{ij}(t) & \text{if } -\mathrm{V}_L \le \mathrm{V}x_{ij}(t) \le +\mathrm{V}_U \\ \mathrm{Gm2d}\ \ \mathrm{V}_U & \text{if } \mathrm{V}x_{ij}(t) > \mathrm{V}_U \\ -\mathrm{Gm2d}\ \ \mathrm{V}_L & \text{if } \mathrm{V}x_{ij}(t) < -\mathrm{V}_L \end{cases} \tag{5.3}$$

$$sign[Iy_{ij}(t)] = \begin{cases} 0\mathrm{V} & \text{if } \mathrm{Vx}_{ij}(t) < 1.25V \\ 2.5\mathrm{V} & \text{if } \mathrm{Vx}_{ij}(t) > 1.25V \end{cases} \tag{5.4}$$

where the Gm2d is the transconductance of T1, V_U is the upper saturation voltage, and V_L is the lower saturation voltage. It can be seen from (5.3) and (5.4) that the block T1 realizes $f(\mathrm{V}x_{ij}(t))$ by separating its magnitude and sign. The sign $sign[\mathrm{I}y_{ij}(t)]$ is detected in the block T1 and its voltage is $V_{SYij,}$ as shown in Fig. 5.4.

The sign of Vzi_{ijkl} is detected and latched by the CMOS dynamic latch circuits of Fig. 5.5 in the block of T2L whereas the sign of the input voltage Vx_{ij} (Vy_{ij}) is detected by the four cascaded CMOS inverters in the block of T2D and its output voltage is denoted as

V_{SXij} (V_{SYij}) in Fig. 5.4. When the input signal Vzi_{ijkl} or Vx_{ij} (Vy_{ij}) is larger than the inverter threshold voltage (1.25V), the output of the latch circuit in Fig. 5.5 or the detect circuit in Fig. 5.4 is high (2.5V). Otherwise, the circuit output becomes low (0V) when the input signal Vzi_{ijkl} or Vx_{ij} (Vy_{ij}) is smaller than the threshold voltage (0V). To avoid the effect of the inverter threshold-voltage variations, the input signal levels are kept well separated from the threshold voltage.

In the learning period, the signs of the input voltages u_{kl}^{p} and the desired output voltage y_{ij}^{p} are detected by the circuit of Fig. 5.4 in T2D and used to determine the sign of Wb_{ijkl} in (4.4) whereas the sign of the voltage Vzi_{ijkl} is detected too. In the recognition period, the sign of node si_{ijkl} or equivalently the sign of w_{ijkl} denoted as Vsw_{ijkl} in Fig. 5.5 is further latched by setting ϕ_1 low (-1.25V) and ϕ_2 high (1.25V). The latched sign is used in generating the first term in (5.4).

Fig. 5.3(b) is presented the HSPICE simulation result of the V-I converter with the absolute-value circuit, which is designed by using 0.25 μm (1P5M) n-well CMOS technology. It can be seen from Fig. 5.3(a) that the voltage Vin is converted into positive current *Ioabs*. The maximum linearity error of *Ioabs* is 15% at Vin – Vref = ± 0.8 V. It is found that this error is acceptable in the SRMCNN.

5.3.2 Analog Multiplier and Divider Circuit

The combined the four-quadrant analog multiplier and two-quadrant divider (M/D) in Fig. 5.1 can be realized in the current mode by the CMOS circuit shown in Fig. 5.6 [171]. In Fig. 5.6, the currents I_1 and I_3 for multiplication are input through the PMOS current sources M14i/M14 and M15i/M15/M16, respectively, whereas the current I_2 as the divider is input through M24i/M24. The parasitic vertical PNP bipolar junction transistors (BJTs) [178] Q1, Q2, Q3, and Q4 are adopted to perform the functions of multiplication and division by using the relation between emitter current I_E and base-emitter voltage V_{BE} as

$$I_E = I_S \exp(V_{BE}/V_T) \quad \text{or} \quad V_{BE} = V_T \ln(I_E/I_S) \tag{5.5}$$

where I_S is the emitter saturation current and V_T is the thermal voltage. The OP AMP Ao has a closed-loop feedback via the NMOS device M21. Thus, the emitter voltage V_{E3} and V_{E4} are virtually the same. With the buffered direct injection circuit [171], the output current I_4 can be readout through the PMOS current mirrors M19, M25, and M26, and the NMOS current mirror M29 and M30 to form the output current I_{omd}. Since V_{E3} = V_{E4}, than the loop voltage as

$$V_{BE1} + V_{BE3} = V_{BE2} + V_{BE4} \tag{5.6}$$

Using the equation in (5.5), the relation among I_{E1}, I_{E2}, I_{E3}, and I_{E4} can be obtained from (5.6) as

$$I_{E4} = \frac{I_{E1}\ I_{E3}}{I_{E2}} \tag{5.7}$$

Neglecting the base currents, the output current I_4 can be expressed in terms of I_1, I_2, and I_3 as

$$I_4 \cong \frac{I_1\, I_3}{I_2} \tag{5.8}$$

In above equation, only the magnitudes of the input current signals are used to form the magnitude of the output current signal. The sign of I_4 should be determined to realize the complete function of four-quadrant multiplier and two-quadrant divider. In Fig. 5.6, the signal “selpn” is used to determine the sign of the output current I_{omd}. The signal “selpn” is obtained from the XNOR gate with the three different input signs. In the learning period, ϕ_1 is high and ϕ_2 is low. The output “selpn” is determined by the sign voltages VSX_{kl} and VSy_{ij} from the block T2D to realize the sign of $Iy^{p}_{ij} Iu^{p}_{kl}$ as in (5.1). In the recognition period, ϕ_1 is low and ϕ_2 is high. “selpn” is determined by the voltage VSu_{kl} and Vsw_{ijkl} from the block T2D and T2L, respectively, to determine the sign of Iu_{kl} Izs_{ijkl} in (5.2). If the signal “selpn” is high (low), the sign is negative (positive) and the MOS device M28 (M27) is turned on to make $Iomd$ = -I_4 (+I_4).

The BJT devices used in Fig. 5.6 are the parasitical vertical BJT in the 0.25 μm n-well CMOS process. The current gain β of the parasitic BJTs is about 6~17. It is not large enough to neglect the effect of the base currents of the BJTs Q3 and Q4 to the emitter currents of the BJTs Q1 and Q2, respectively. Thus, extra circuits are needed to further bypass the base currents from entering the emitters of Q1 and Q2. In Fig. 5.6, the BJTs Q13 and Q24 have the same emitter currents I_3 and I_4 as Q3 and Q4, respectively. Thus, Q13 (Q24) has the same base current as Q3 (Q4). The current mirror circuits M17/M18 (M22/M23) are used to mirror the base current of Q13 (Q24) to Q3 (Q4). Thus the base current of Q3 (Q4) is bypassed from Q1 (Q2) and the relation $I_{E1} = I_1$ and $I_{E2} = I_2$ can be more accurately maintained to realize (5.8).

In the learning period, the M/D block functions as a multiplier to implement the multiplication function $Iy_{ij}Iu_{kl}$. The HSPICE simulation results of the multiplier function of the M/D circuit in Fig. 5.6 are shown in Fig. 5.7(a), where the device parameters of 0.25 μm 1P5M n-well CMOS technology are used. It is found that in the actual operation range of I_1 from 0.5 μA to 6 μA and I_3 from 1.2 μA to 6 μA with I_2 kept at 20 μA, the multiplication error can be kept fewer than 5.5%. In the recognition period, most of the $u_{kl}(t)$ from the neighboring cell input is kept at the maximum absolute value as in (5.3). Thus most of the corresponding input current I_1 of the M/D block becomes a constant current and the M/D block is functioned as a divider. The HSPICE simulation results of the divider function of the M/D circuit in Fig. 5.6 are shown in Fig. 5.7(b). In the actual operation range of I_3 from the 1.2 μA to 6 μA and I_2 from 0.3 μA to 6 μA with I_1 is kept at 6 μA, the output current can be as high as 60 μA. Under the condition $I_3 < I_2$ which is the actual operation condition, the division error can be kept under 10%.

The above errors of the M/D circuit are also dependent on the variations of device parameters. However, it is found from simulation results that these errors have insignificant effects on the operation of RMCNN because of on-chip learning and RM operation. This is because that the RM is sensitive to the variation of the multiplier-divider. However, the multiplier-divider is not sensitive to the variation of the processing due to that the β of BJT will not affect the precision of the multiplier-divider.

5.3.3. The CMOS Readout Circuit

A layer of the boundary cells is designed to surround the 18x18 regular cell arrays. In the boundary cells, both state $x_{ij}(t)$ and input $u_{ij}(t)$ are zero. Thus the output $y_{ij}(t)$ of the boundary cell is also zero. Since the boundary cells have to send a zero signal voltage to the neighboring regular cells or other boundary cells, it can be realized by setting the weights from boundary cell to other cells to be zero. Thus the associated RM blocks can be removed.

To readout the neuron states signal x_{ij}, suitable readout circuit shown in Fig. 5.8 is designed in the 18x18 CMOS SRMCNN. In the readout circuit, the inputs of NMOS-input CMOS single-stage OP AMPs used as unity-gain buffers are connected to the node x_{ij} in Fig. 5.8. The buffer output is connected to the input of the source-follower driver through the switch controlled by the column select control signal *CSj*. In the readout operation, *CSj* is raised to high column by column so that x_{ij} is sent to the input of NMOS source follower M31 and M32 with M32 biased by V_{BUF} as the current source. Through the source follower, the neuron state signal can be readout column by column to the output pad and the large off-chip load.

5.4 SIMULATION RESULTS

The CMOS circuit as function blocks in Section 5.4, the architecture of the SRMCNN in Fig. 5.1 is implemented with the 18x18 array. In the implemented 18x18 CMOS SRMCNN with **B** template, the capacitors C*zi* and C*zs* for absolute weight voltage storage are realized by the NMOS gate capacitors. Because the current-mode output signals used in the SRMCNN, the summing and distribution block is realized by directly connecting the output nodes of the related blocks. The collected current send to the input of the master stage in a CMOS current mirror to perform the summing function and the mirrored output current is distributed out through the multiple slave stages.

The recognition behavior of the SRMCNN with B template for associative memory application has been simulated by the Matlab software. The exemplar patterns in sequence input to the system produce the weights for the hetero-associative memory learning. The learned weights can be used to processing the noisy patterns with white-black noise for the ratio memory operation. The SRMCNN can successfully recognize the test pattern and output the desired correct pattern. The simulated results have been verified the capability of the image processing and its VLSI circuits can successful implemented.

Due to the simulation results, the architecture of the proposed SRMCNN with **B** template and r=1 in Fig. 5.1 is designed and fabricated by the 0.25μm 1P5M n-well technology. The CMOS circuits of the block in the SRMCNN are presented in the Fig. 5.3, 5.4, 5.5, and 5.6, respectively. The functions of those circuits are correctly verified by the HSPICE software.

As the function of ratio memory, the experimental chip of one bit SRMCNN with **B** template used to observe the feature enhancement effect. The five neighboring pixels are input in serial for six data is shown in Fig. 5.9(a), to the SRMCNN with the period 0.5μs per each charge time. The learned weights through out the six times learn are described in Fig. 5.9(b). The desired output of the cell is 1. The operation of learning and ratio state for the SRMCNN is presented, as the Fig. 5.9(c). In the Fig. 5.9(d), the zoom-out variations of the five weights versus the elapsed period from 8μs to 22μs are shown. The absolute coefficients of the learned weights of the **B** template are gradually decreased with the constant leakage current for the elapsed period. And take the weights with leakage to operate in the ratio of the absolute summation for its neighboring weights of B template. As the simulation results from the Fig. 5.9(d), the larger value of the weight is enhanced the ratioed weight to 1 and the smaller values of the weights are suppressed to zero during the elapsed period. The layout graph of the one bit SRMCNN chip operated in the learning and ratio memory for the TSMC 0.25μm 1P5M n-well technology is shown in Fig. 5.10. It includes one regular cell, five RMs, and five current mirrors. The characteristics of fabricated one bit SRMCNN with **B** template chip is summarized in the Table 5.1.

The 18x18 SRMCNN with **B** template is proposed, their capability of the recognition has been verified, and the function blocks in the architecture are also integrated with the CMOS circuits. The integrated circuits are simulated by the HSPICE software. The five exemplar patterns for the English capital letter H are sequent applied to update the voltage on the capacitor Czi of the associative memory in the SRMCNN. The learning period is 0.5μs for each input exemplar patterns. The learned weights of the **B** template are gradually decreased with constant leakage current for an elapse time. After 350 sec, the resultant weights are stored in the capacitor Czs, as the associative memory.

The voltages with leakage stored on Czs for the associative memory are divided by its absolute summation of the neighboring weights to generate the ratioed weights. The ratioed weights multiplied by the value of the test input are performed from the M/D circuit in the RMs during the recognition period. The simulation results of the learning and recognition using the HSPICE software for 18x18 SRMCNN with **B** template are shown in Fig. 5.11(a). In the Fig. 5.11(b) presents the five exemplar input patterns with zoom-out for the English capital letter H. The SRMCNN can correctly recognize the output pattern for any one exemplar input test pattern. The recognized pattern is sequentially shift-out from the column neurons in time series, as shown in the Fig. 5.11(c). Due to the simulation results, the SRMCNN with **B** template and modified Hebbain algorithm can successful recognized the noisy patterns with white and black noise for auto-associative memory applications.

The layout graph of the 9x9 SRMCNN chip operated in the learning and recognition operations for the TSMC 0.25μm 1P5M n-well technology is shown in Fig. 5.12. It includes 81 regular cells, 405 RMs and 81 current summations. The brief characteristics of fabricated 9x9 SRMCNN with **B** template chip are summarized.

5.5 LARGE NEIGHBORING STRUCTURE

The general architecture of the Large-Neighborhood Cellular Nonlinear (Neural) Network Universal Machine (LN-CNNUM) is presented in Fig. 5.13 [157]-[162], [179]-[182]. As shown in Fig. 5.13, the universal machine includes a 64x64 regular array of cells, a Global Analogic Programming Unit (GAPU), analog and digital input/output circuits, and global address decoders. The core-computing unit of the LN-CNNUM is the large-neighborhood cell with a new large-neighborhood interconnection. The powerful LN-CNNUM can also perform complex tasks with large-neighborhood templates, such as Muller-Lyer arrowhead illusion, etc.

Fig. 5.14(a) depicts the structure of LN-CNNUM. The programmable templates **A** and **B** are realized by the path stages PAR, PAL, PAIR, PAIL, PAU, PAD, PAIU, PAID, PA1, PA2, PA3, PADR, PADL, PAUR, PAUL, and PBR, PBL, PBIR, PBIL, PBU, PBD, PBIU, PBID, PB0, PB1, PB2, PB3, PBDR, PBDL, PBUR, PBUL, respectively, which are basically current amplifiers. The connected neuron array is described in Fig. 5.14 (b).

According to the LN-CNNUM cell structure of Fig. 5.14(a), the corresponding realized LN template weights are shown in Fig. 5.15 where the neighborhood layers are defined in a different way. The numbers of neurons in the r-th layer is equal to $4r$ that is different from $8r$ in the conventional CNN. The LN-CNNUM structure can realize asymmetrical large-neighborhood templates with different weight values in the first and second neighboring cell layers. The cell structure is shown in Fig. 5.16. The LN-CNNUM is composed of the LN-CNN Kernel unit, a Local Logic Unit (LLU), the Local Analog Memory (LAM), the Local Logic Memory (LLM), the Local Analog Output unit (LAOU), and the Local Communication and Control Unit (LCCU). The LN-CNN kernel unit of the cell in the LN-CNNUM consist the large-neighborhood interconnection. Fig. 5.14 shows the kernel unit's functional blocks.

A general-purposed parallel analogic processor, called the LN-CNNUM, is designed and analyzed. Similar to a digital computer, the LN-CNNUM is controlled by sets of instructions. Not only is an elementary LN-CNN structure, the processor also a platform for integrating the flow of LN-CNN operations. Furthermore, the LN-CNNUM is an important tool for organizing various kinds of CNN structures to perform complicated tasks that a SN-CNN cannot finish. Consequently, some local memories, some logic-computing units, and complex configurations of several switches are added to the LN-CNNUM, which thus becomes an analogic computer. With a powerful computing unit, the LN-CNNUM can deal with highly complicated functions.

Future research on efficient physical structures for implementing νBJT will concern nano-scale devices and integration. Additionally, research into LN-CNNUMs will also concern the nanoelectronic regime, and lead to the development of a feasible and powerful nano-scale CNNUM (Nano-CNNUM).

5.6 SUMMARY

In this chapter, the structure of the self-feedback ratio memory cellular nonlinear network with modified Hebbain algorithm is proposed and analyzed for hetero-associative memory applications. In the SRMCNN, the weights of the **B** template are generated from a set of the exemplar patterns and the desired output pattern, and then transform the learned weights into ratio weights stored on the ratio memory. The proposed network can be used the pattern learning, recognition, and recovery in hetero-associative memory for the various image processing applications. The function blocks have implemented by the TSMC 0.25 μm 1P5M n-well CMOS technology. The correctly characteristic of the proposed circuits are also verified by the HSPICE software.

As the feature of ratio memory, the experimental chip of one bit SRMCNN with **B** template is implemented to observe the feature enhancement effect. The chip area including one neuron cell and five RMs is 350μm x 400μm. The five neighboring pixels are input in serial for six data to the SRMCNN with the period 0.5μs per each charge time. The desired output of the cell is 1. The absolute coefficients of the learned weights of the **B** template are gradually decreased with the constant leakage current for the elapsed period. The weights with leakage used in the ratio of the absolute summation for its neighboring weights of **B** template. The operation of ratio state in the SRMCNN is presented the zoom-out variations of the five weights versus the elapsed period from 8μs to 22μs after learning time. As the simulation results, the larger value of the weight is enhanced the ratioed weight to 1 and the smaller values of the weights are suppressed to zero during the elapsed period.

The simulation results have successful verified the correct function of the 18x18 SRMCNN for patterns recognition. The chip layout of 9x9 SRMCNN is implemented and verified. The chip included 81 neurons and 405 RMs in the 4000x4200 μm^2 area sizes for the TSMC 0.25 μm 1P5M n-well CMOS technology. The combination of the four chips is used in an 18x18 SRMCNN form for auto-associative memory applications. Thus, the SRMCNN can correctly recognize the output pattern for any one exemplar input test pattern. Due to the simulation results, the SRMCNN with **B** template and modified Hebbain algorithm can successful recognized the noisy patterns with white and black noise for auto-associative memory applications. The conceptual design for the general architecture of the Large-Neighborhood Cellular Nonlinear (Neural) Network Universal Machine (LN-CNNUM) is introduced.

Table 5.1 The summary on the characteristics of the fabricated one bit SRMCNN chip

Technology	0.25 µm, 1P5M, n-well CMOS
Resolution	one Cells
No. of RM blocks	5 RMs
one Pixels	1 cell + 5 RMs
one bit SRMCNN area	350 µm x 400 µm
one bit SRMCNN area with I/O pad	1085 µm x 1085 µm
Transistor/gate count	821
Power supply	2.5V
Total power dissipation	0.6 mW
Minimum learning time of a pixel	0.5 µs
Elapse time	350 sec

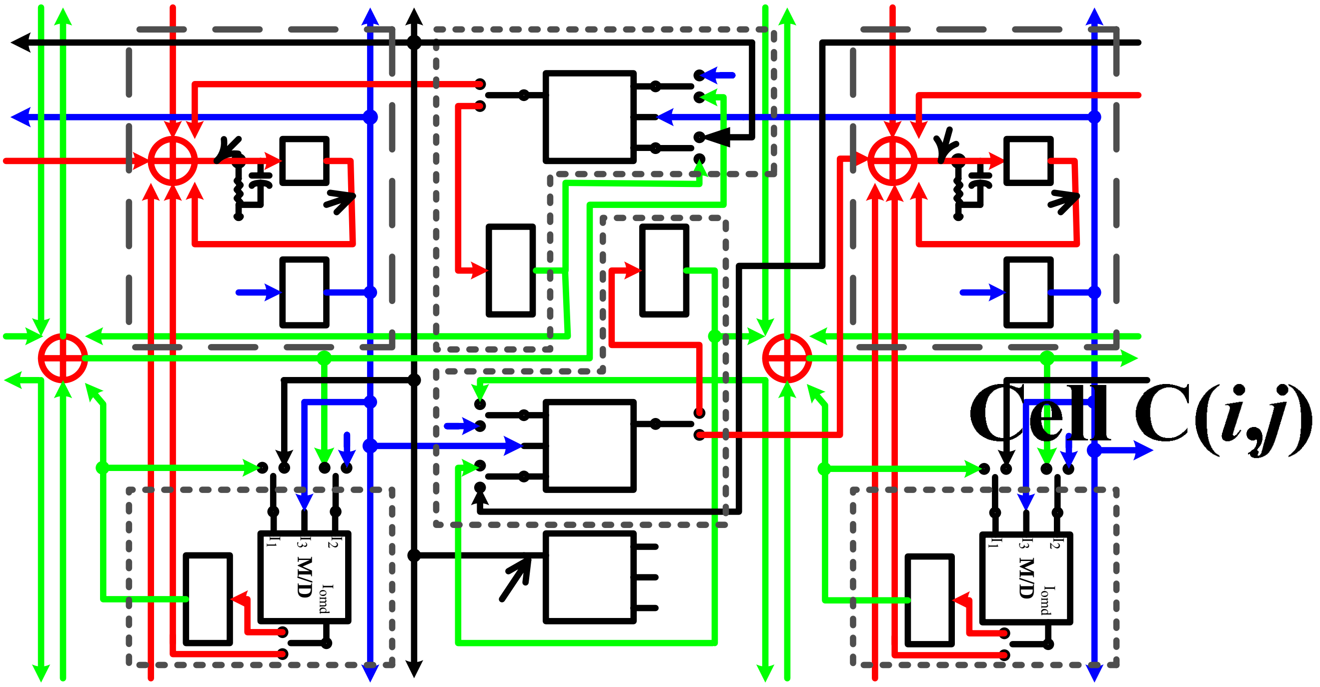
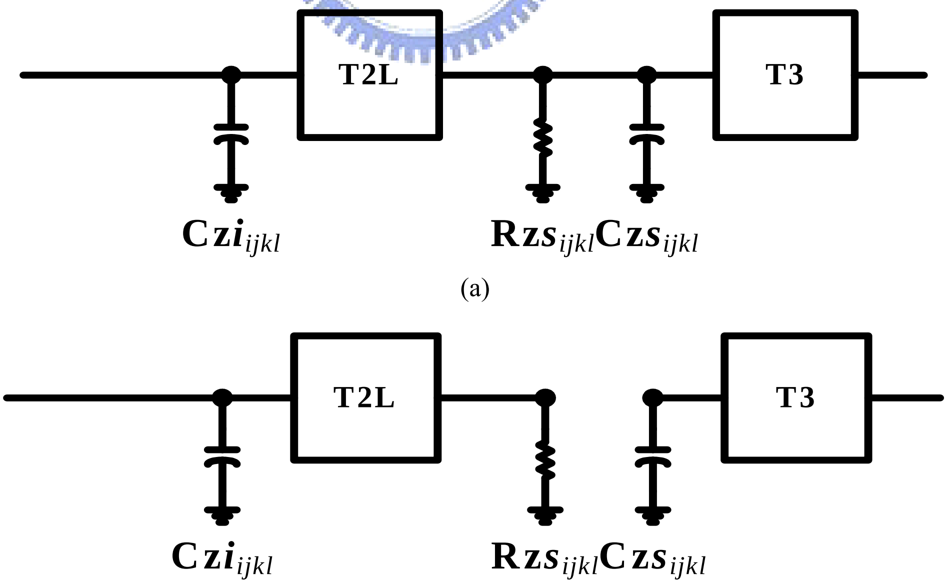


Fig. 5.1 The detailed architecture of two neighboring cells and their ratio memory (RM) in the SRMCNN

$\mathbf{X}_{ij}$


T2L
T3
$\mathbf{Czi}_{ijkl}$
$\mathbf{Rzs}_{ijkl}\mathbf{Czs}_{ijkl}$
T2L
T3
$\mathbf{Czi}_{ijkl}$
$\mathbf{Rzs}_{ijkl}\mathbf{Czs}_{ijkl}$

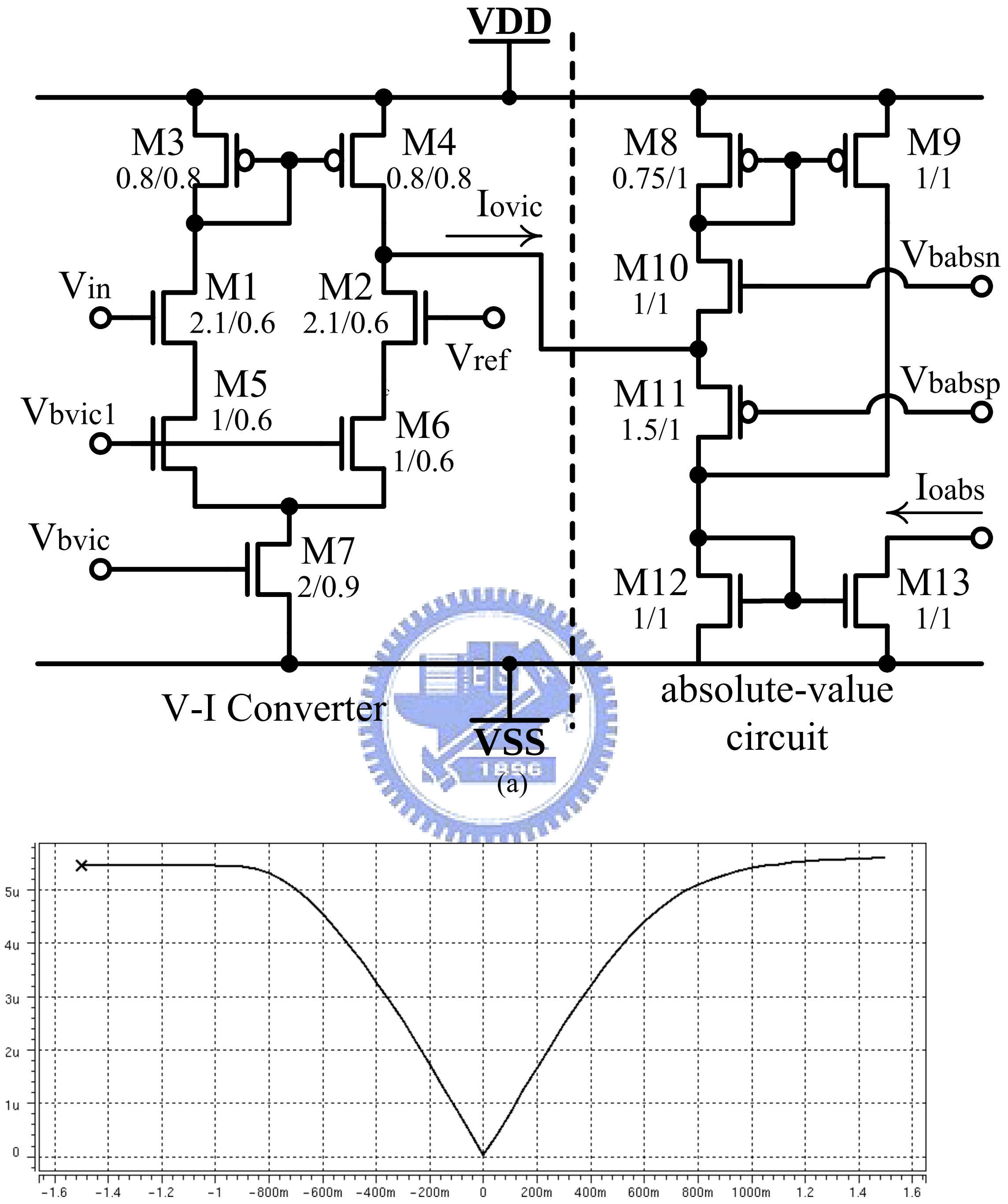
(a)

(b)

$\mathbf{R}_{ij}$

$\mathbf{u}_{ij}^{t}$

Fig. 5.2 The S block in the SRMCNN during (a) learning period and (b) recognition period



(a)

(b)

Fig. 5.3 (a) The circuit of V-I converter of the blocks T2, and the absolute-value circuit; (b) The HSPICE simulation results

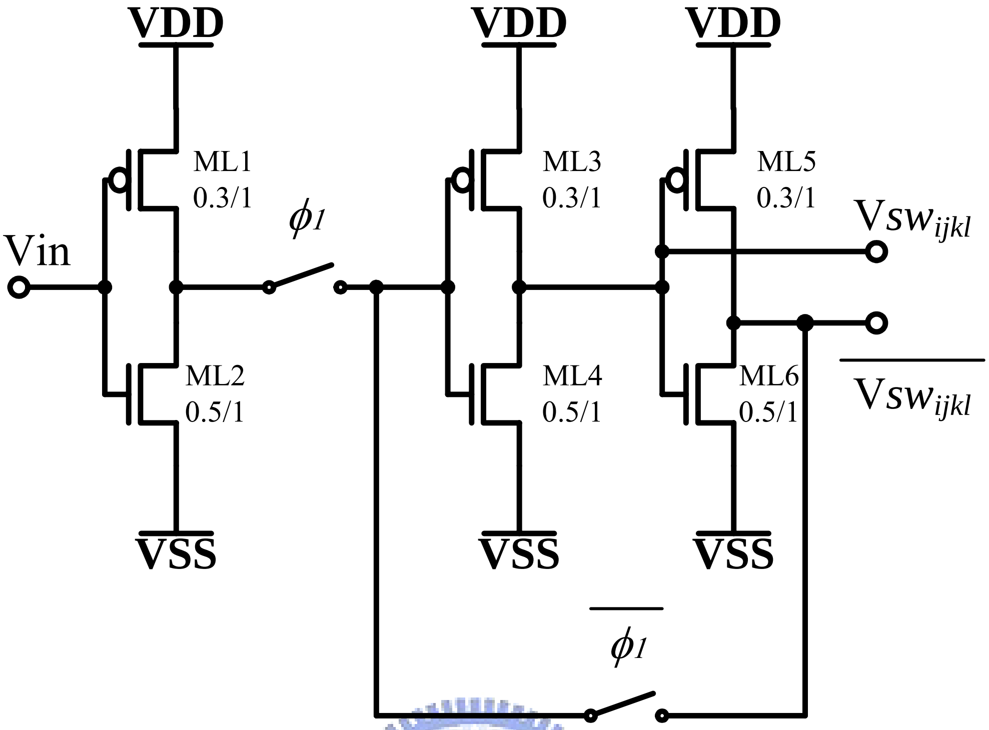


Fig. 5.4 Sign latch

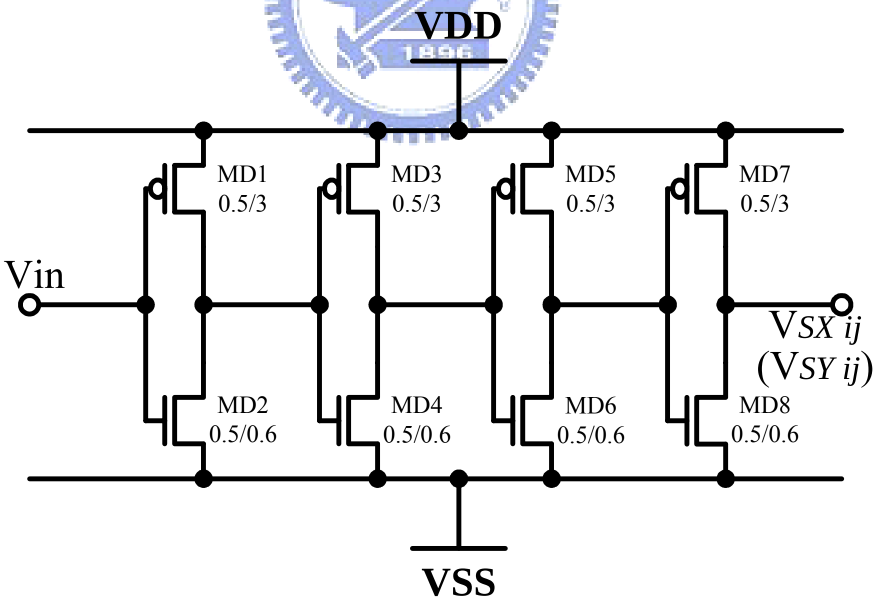
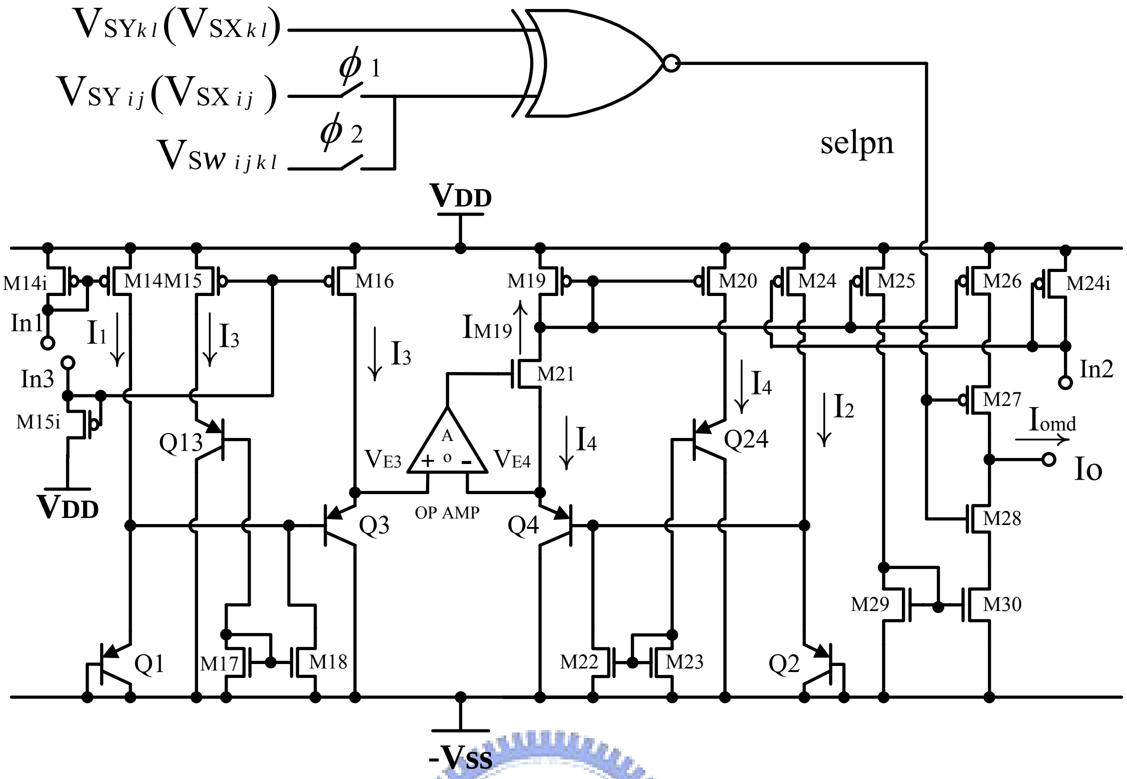
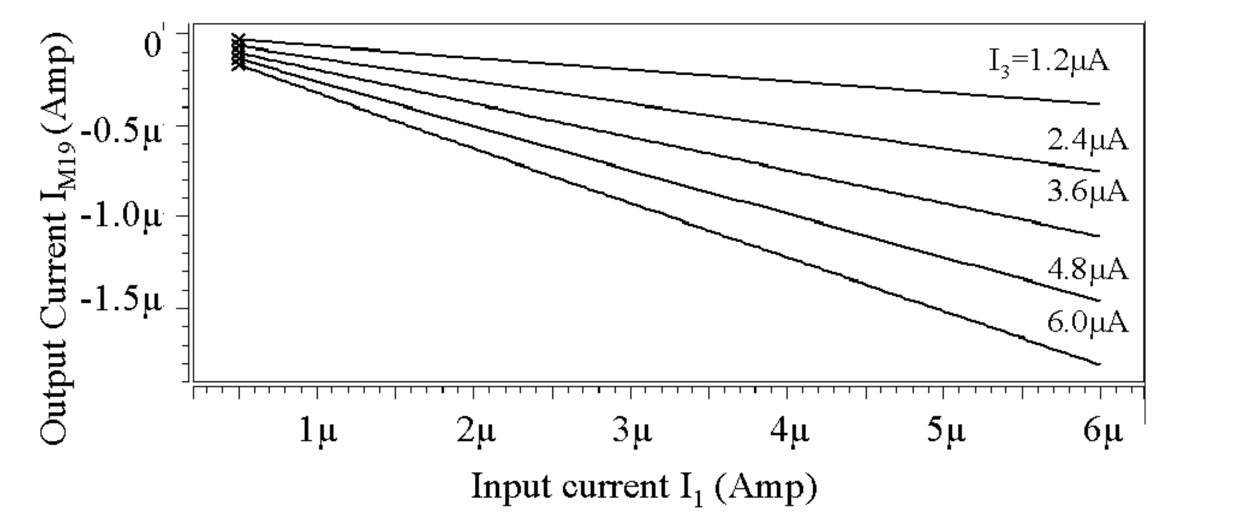


Fig. 5.5 Sign detector



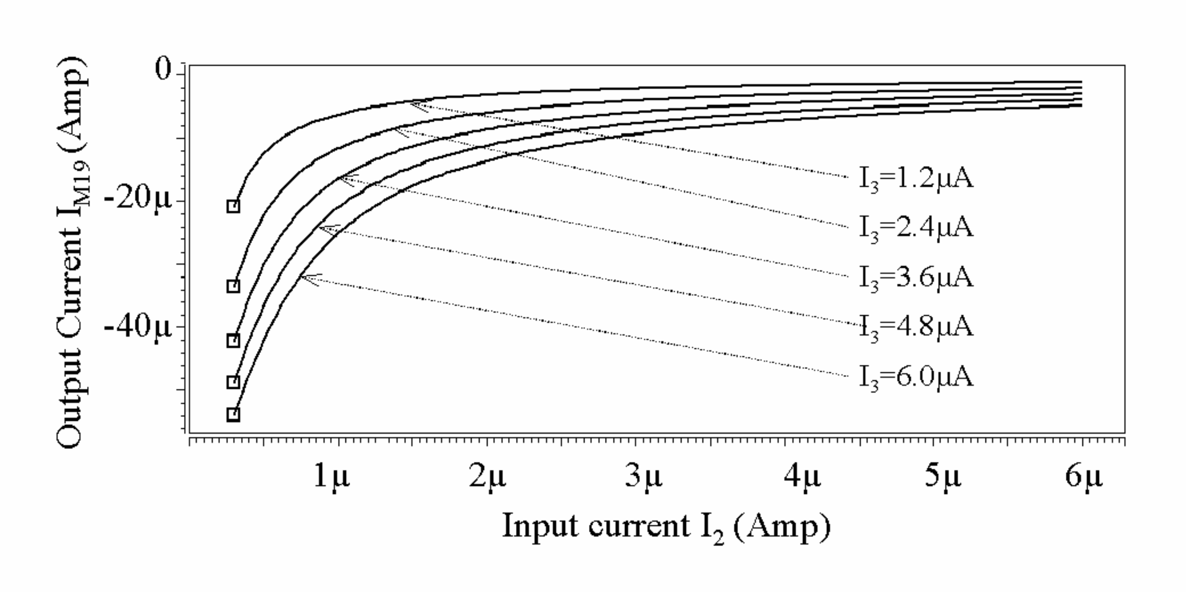
L / W	
1.0um / 5.0um	M14、M14i、M15、M15i、M16、M20、M24、M24i、M25
1.0um / 2.1um	M17、M18、M22、M23
1.0um / 2.3um	M28
1.0um / 3.9um	M19
1.0um / 4.7um	M26
1.0um / 5.5um	M27
1.0um / 6.0um	M21
2.0um / 4.1um	M29、M30

Fig. 5.6 The CMOS circuit of the block M/D



Multiplication Function I_2=20μA

(a)



Division Function I_1=6μA

(b)

Fig. 5.7 HSPICE simulation results for (a)Multiplication function with I_2=20μA (b) Division function with I_1=6μA

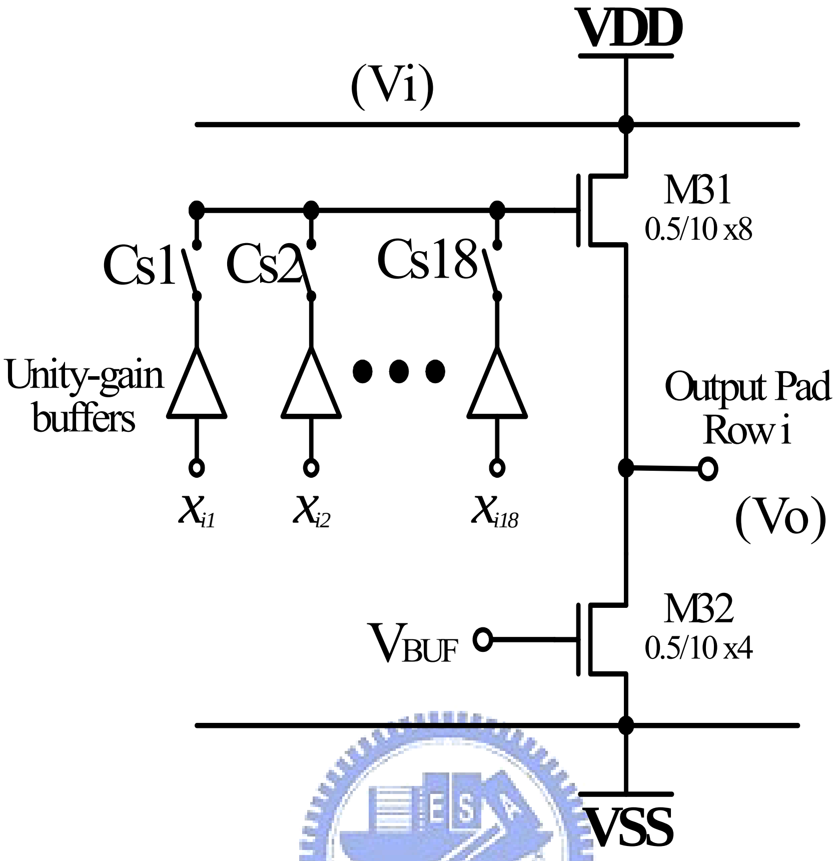
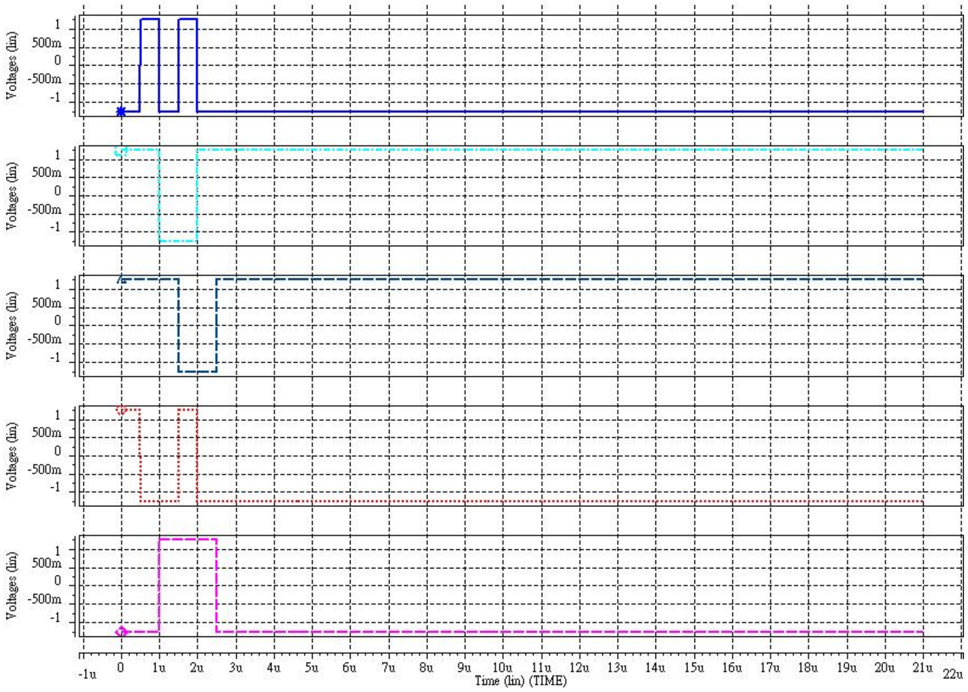
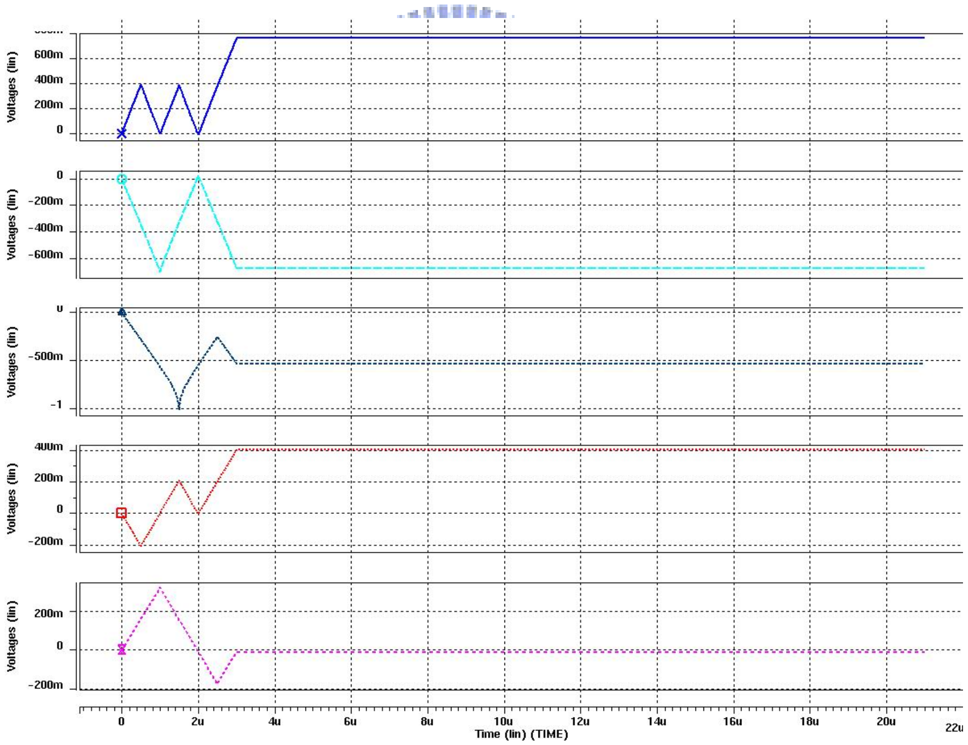


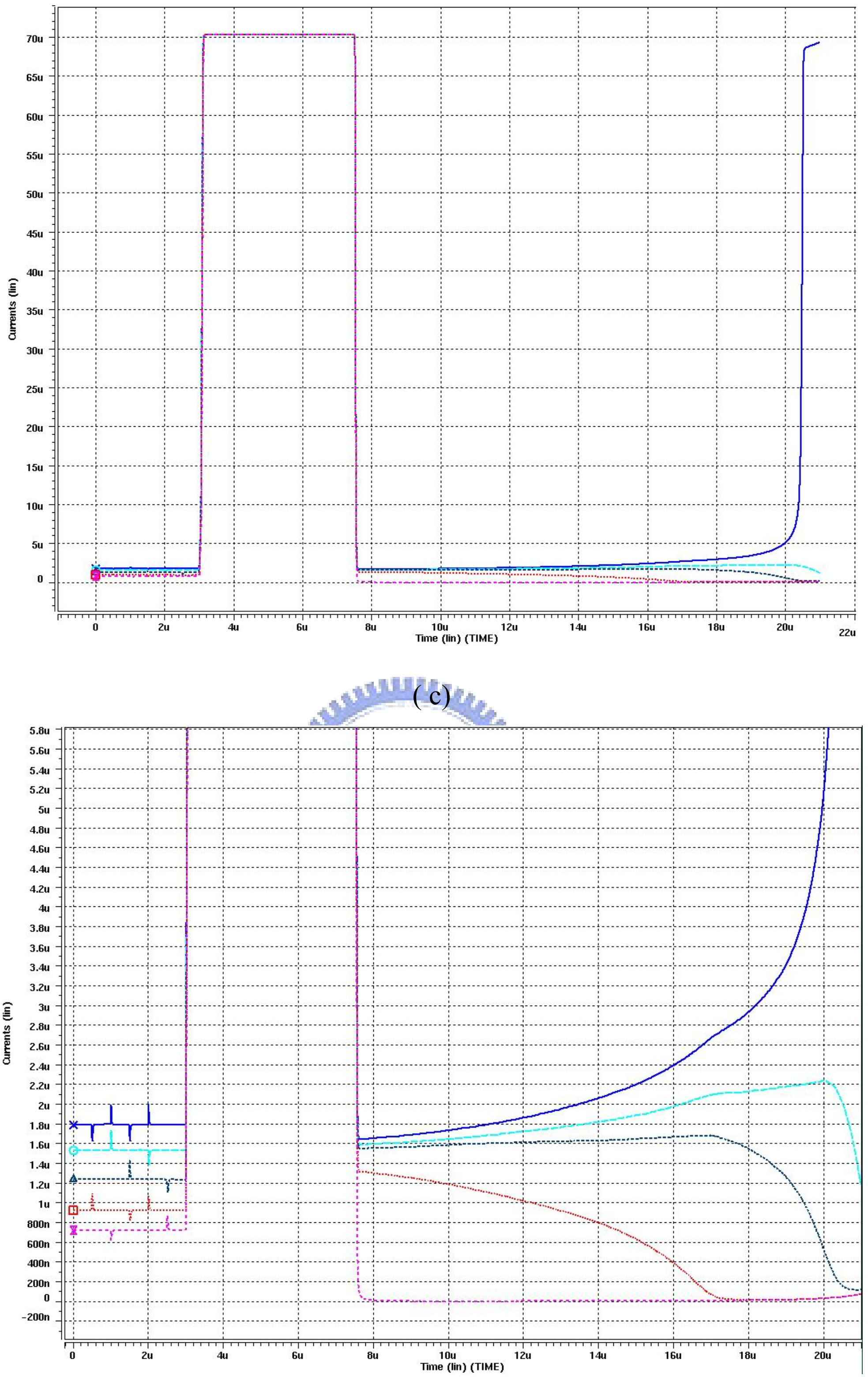
Fig. 5.8 The CMOS readout circuit for the cell output


Voltages (lin)
1
500m
0
-500m
-1
-1u
0
1u
2u
3u
4u
5u
6u
7u
8u
9u
10u
11u
12u
13u
14u
15u
16u
17u
18u
19u
20u
21u
22u
Time (lin) (TIME)

(a)


Voltages (lin)
600m
400m
200m
0
-200m
-400m
-600m
-500m
-1
400m
200m
0
-200m
200m
0
-200m
0
2u
4u
6u
8u
10u
12u
14u
16u
18u
20u
22u
Time (lin) (TIME)

(b)



(c)

(d)

Fig. 5.9 SRMCNN (a) learn data with the five neighboring cells for six data input; (b) the learned weights; (c) learning and ratio state; (d) the ratioed weights during each elapse time

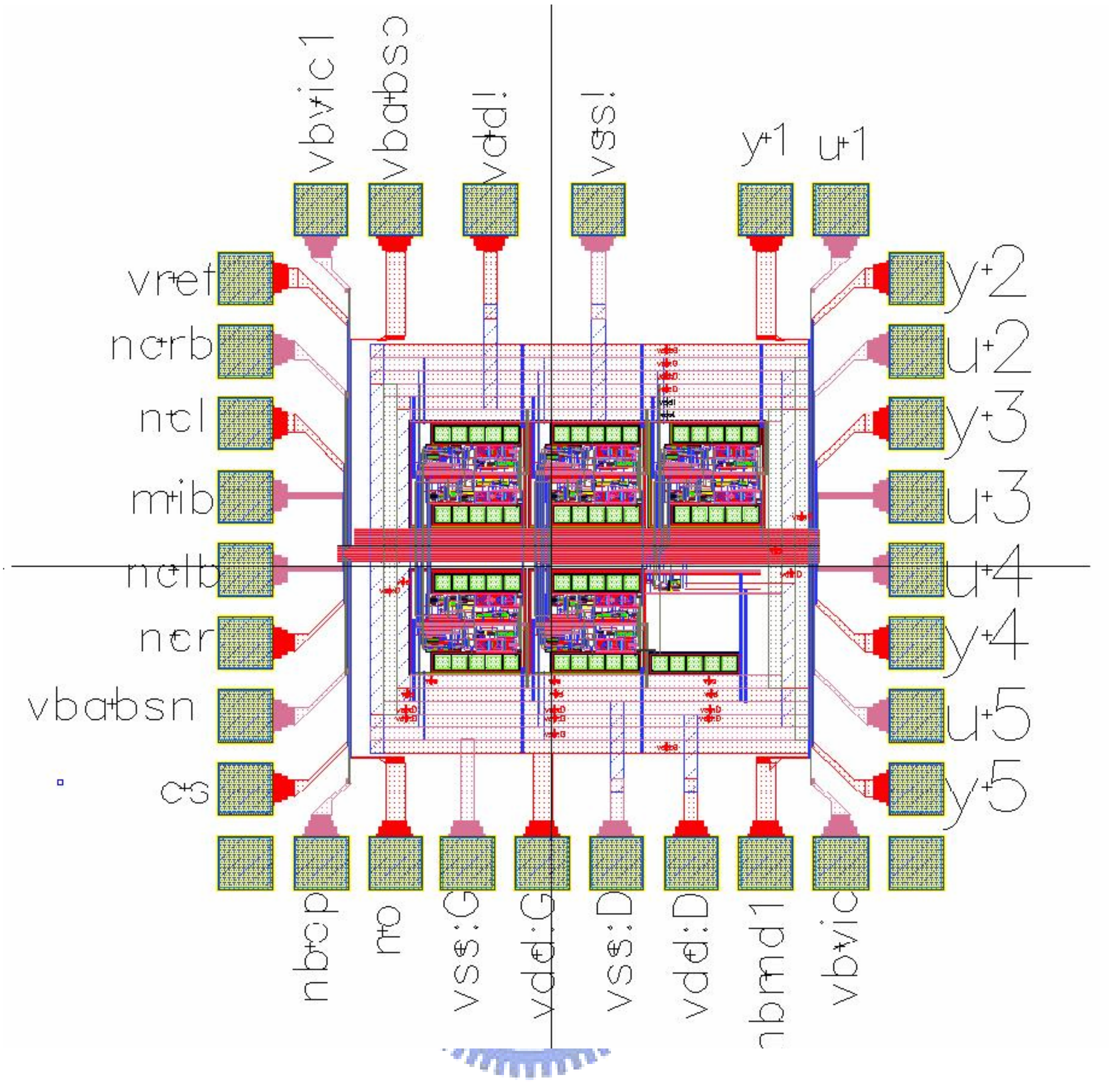
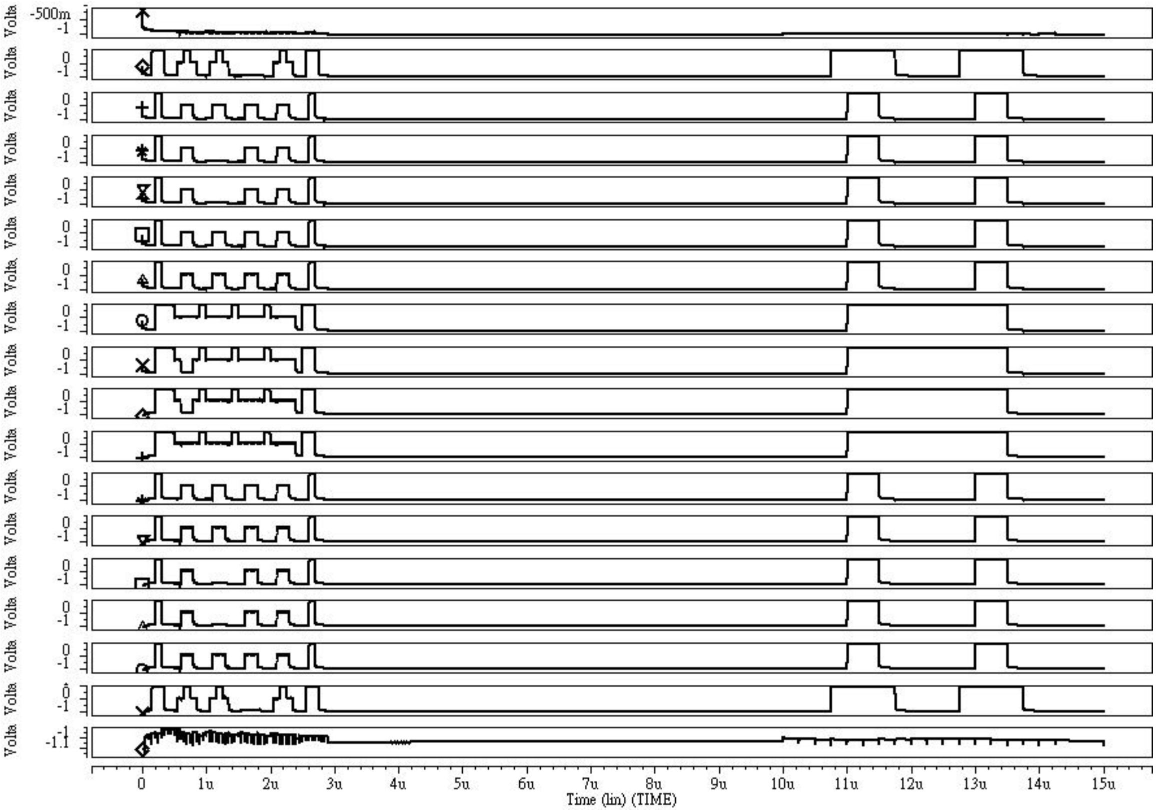
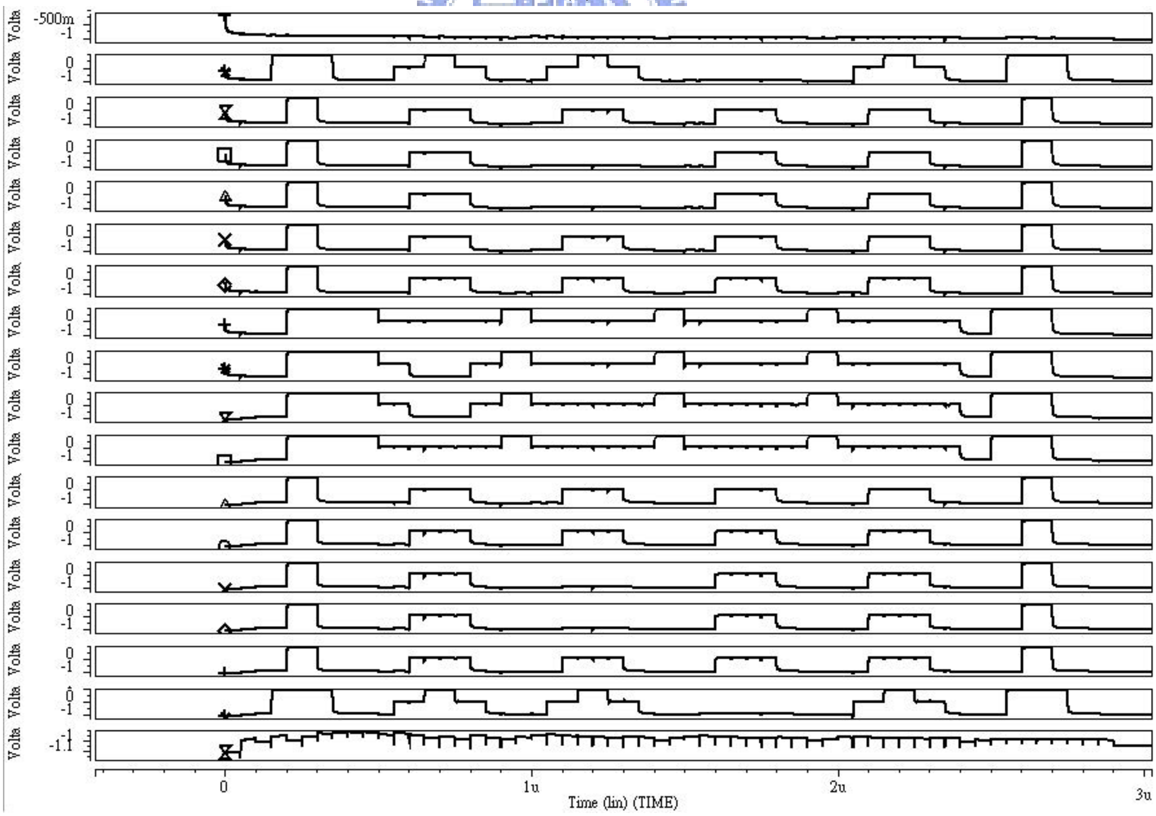


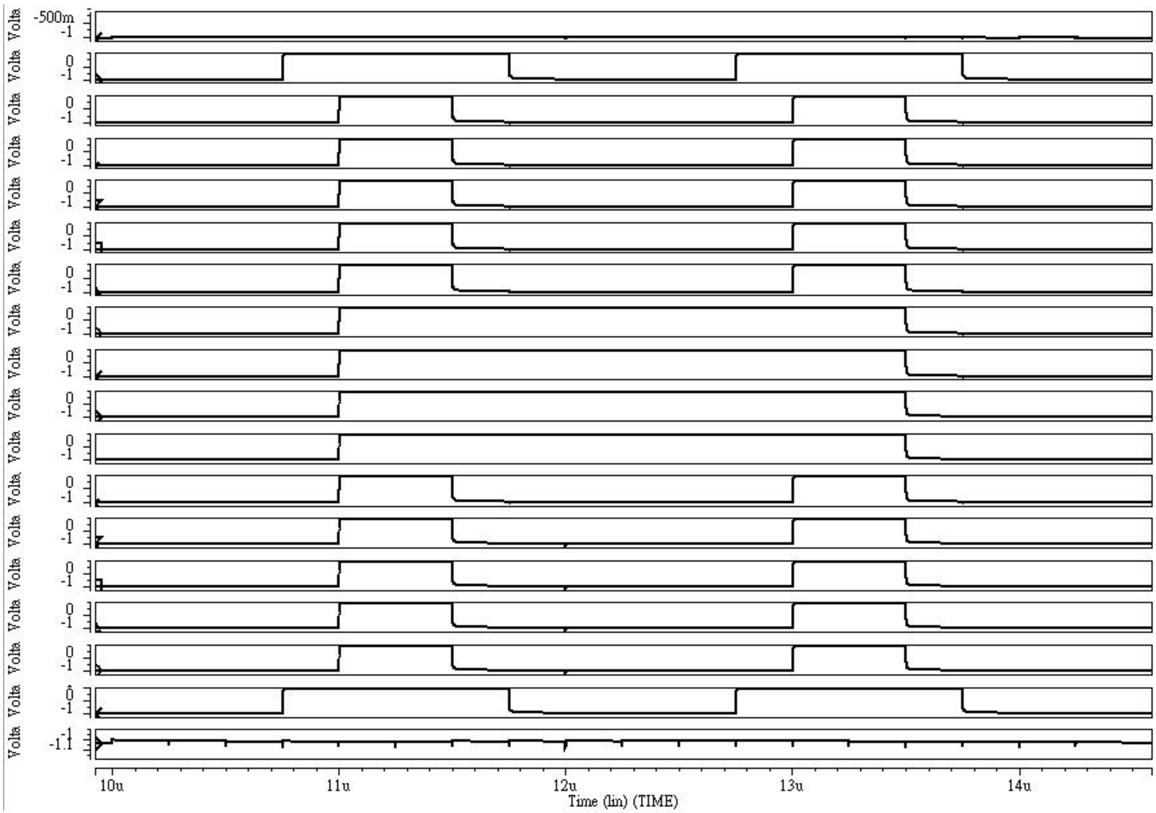
Fig. 5.10 The layout graph of one bit SRMCNN for learning and ratio


Volta
-500m
-1
0
-1
-1.1
0
1u
2u
3u
4u
5u
6u
7u
8u
9u
10u
11u
12u
13u
14u
15u
Time (lin) (TIME)

(a)

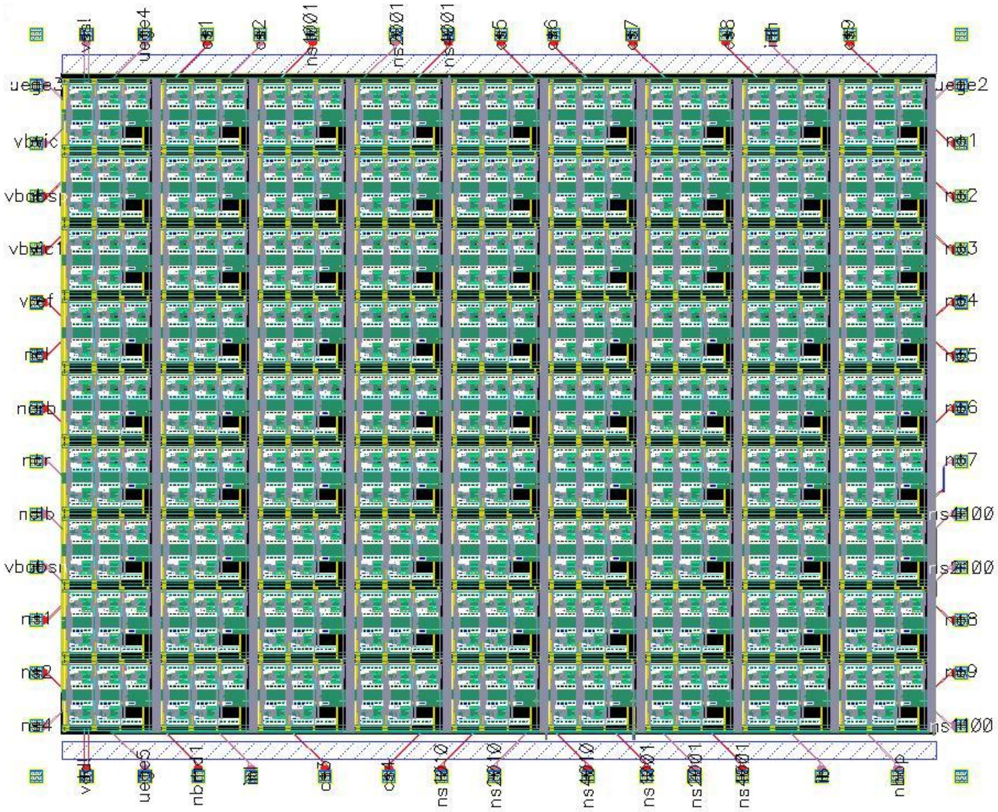

Volta
-500m
-1
0
-1
-1.1
0
1u
2u
3u
Time (lin) (TIME)

(b)



(c)

Fig. 5.11 The HSPICE simulation (a) learn five exemplar patterns with white-black noise and the recognized output pattern; (b) zoom-out the data of exemplar patterns; (c) zoom-out the recognized pattern



Technology	TSMC 0.25 μm, 1P5M, n-well CMOS
Resolution	9x9 Cells
one Pixels	1 cell + 5 RMs
SRMCNN	81 cell + 405 RMs
Chip area	4000 μm x 4200 μm

Fig. 5.12 The layout graph and characteristics for 9x9 SRMCNN system

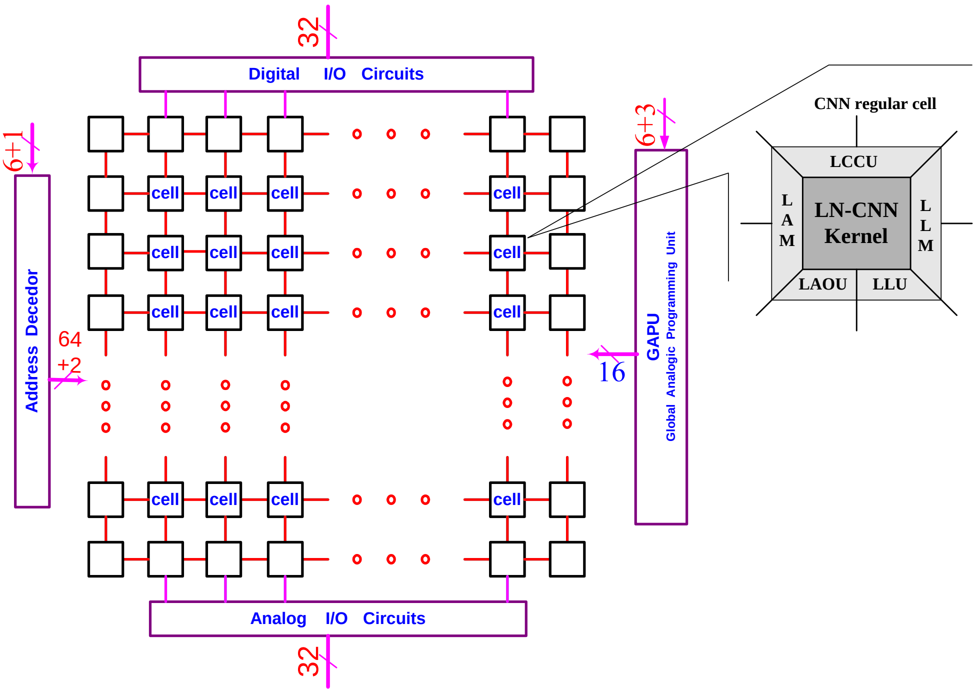
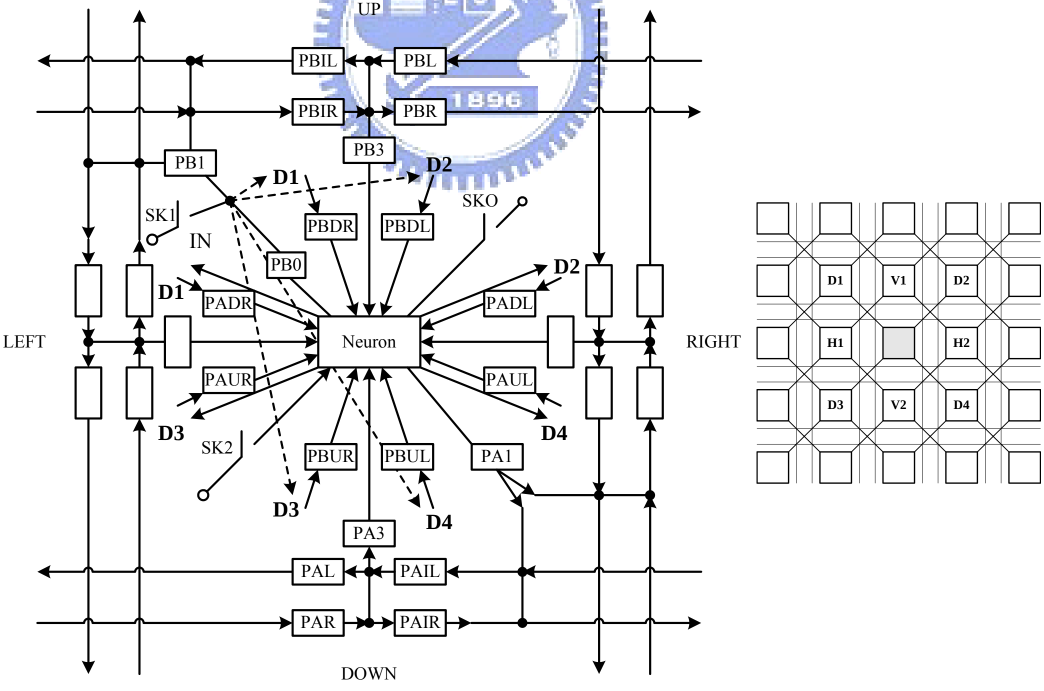


Fig. 5.13 Global architecture of LN-CNNUM.



(a) (b)

Fig. 5.14 (a) Structure of LN-CNN kernel unit. (b) Connections between neighboring cells and

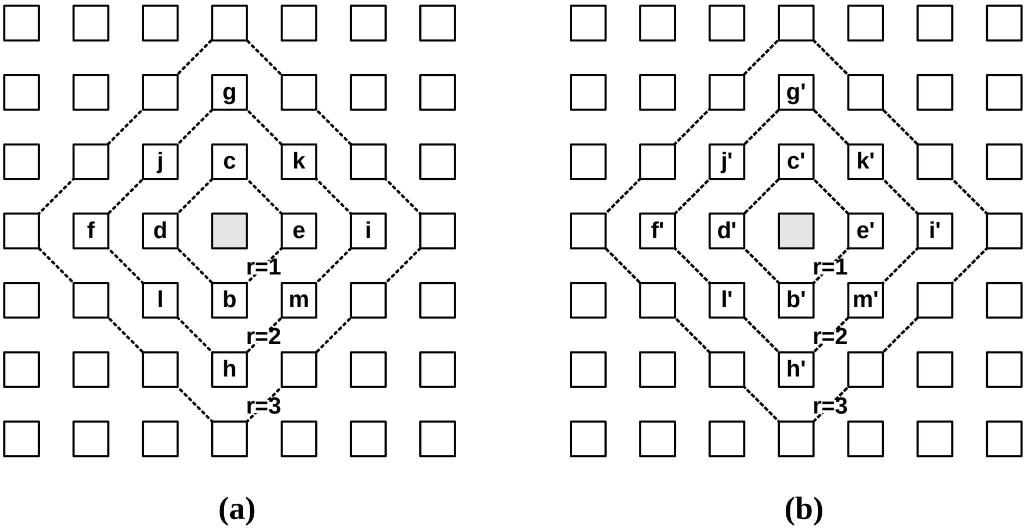


Fig. 5.15 Weights of the LN-CNNUM in (a) template **A** and (b) template **B**.

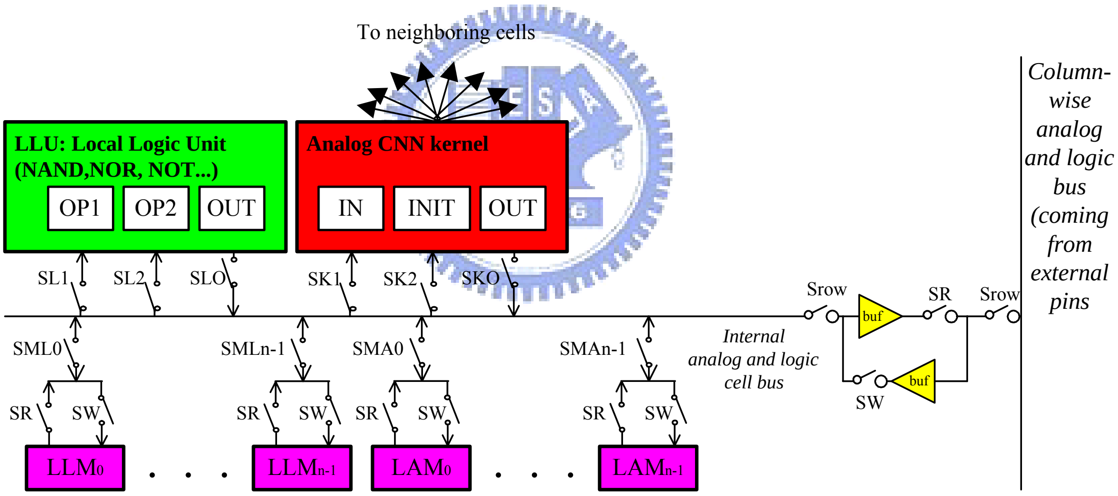


Fig. 5.16 Architecture of one cell in the LN-CNNUM.