The Design and Analysis of a CMOS Low-Power Large-Neighborhood CNN With Propagating Connections

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

***Abstract*—The design of a large-neighborhood cellular nonlinear network (LN-CNN) with propagating connections is proposed. The propagating connections are utilized to achieve large-neighborhood templates in the shape of diamonds. Based on the propagating connections, each LN-CNN cell can only be connected to neighboring cells without interconnections to farther cells. Thus, it is suitable for very large scale integration implementation. The LN-CNN functions of diffusion, deblurring, and Müller-Lyer illusion are successfully verified. Meanwhile, the functions of erosion and dilation are expanded with the diamond-shaped LN templates. Furthermore, the simple N- and P-type synapses stop all the static current paths so that the dc power dissipation can be reduced to only 0.7 mW on standby and 18 mW in operation. An experimental LN-CNN chip with a 20 $\times$ 20 array has been fabricated using 0.18-μm CMOS technology. With the proposed LN-CNN chip, more applications and LN-CNN templates can be studied further.**

***Index Terms*—Cellular neural networks (CNNs), CMOS, large-neighborhood (LN), propagating connections.**

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 to the composition of the cellular automata [4], [5], it is composed of a massive aggregation of regularly spaced circuit clones, called cells, which communicate with each other directly and locally. In a basic CNN, each cell is connected to its nearest layer of neighboring cells. Such a CNN, called a 3 $\times$ 3 neighborhood CNN, is the most popular CNN structure. Their local connectivity makes CNNs easy to be implemented in a very large scale integration (VLSI) design. So far, many 3 $\times$ 3 neighborhood CNN VLSI chips have demonstrated their capabilities in realizing real-time signal and parallel processing functions [6]–[15].

The CNN universal machine (CNNUM) [6] is a programmable CNN, which can perform several complicated functions. Recently, research on the CNNUM has been conducted and successfully implemented. Current CNNUMs are based on the 3 $\times$ 3 neighborhood CNN structures [7]–[12] and 3 $\times$ 3 neighborhood templates. Some applications [16], [17] are verified by using the CNNUM. However, 3 $\times$ 3 neighborhood CNNs with the nearest neighborhood are restricted in their ability to solve complex problems efficiently. Although a large-neighborhood template can be transformed into several 3 $\times$ 3 neighborhood templates [18], [19], the multiple operating steps with 3 $\times$ 3 neighborhood templates require more time and power.

It is more efficient to construct a large-neighborhood CNN (LN-CNN), which can perform functions using large-neighborhood templates. In an LN-CNN, each cell is connected to more than one layer of the neighboring cells. Generally, an LN-CNN is difficult to be implemented in a VLSI design through direct wire connections among the 3 $\times$ 3 neighborhood CNN cells. Recently, however, a design for an LN-CNN has been proposed and implemented by using a new device called the neuron BJT (νBJT) [13]–[15]. Based on the νBJT, an LN-CNN with symmetric templates has been designed [13], [14]. The LN-CNN with asymmetric templates has also been proposed with some limitations in realizing large-neighborhood templates [15].

In this paper, a new improved low-power CMOS compact LN-CNN architecture with propagating synaptic connections [21], [22] is proposed and analyzed. In the proposed kernel unit, only one layer of the neighboring cells is connected, but it can realize large-neighborhood diamond-shaped templates in the first two neighboring layers. Thus, complicated wire connections to farther cells can be avoided. The propagating synaptic connections can be used not only in horizontal and vertical directions but also in diagonal directions. As a result, the circular symmetric templates can be realized. Moreover, the circuitry can be shared between templates $\boldsymbol{A}$ and $\boldsymbol{B}$ in the proposed architecture. This results in a simpler architecture and smaller chip area. To realize the proposed architecture, the low-power neuron and synapses have been designed using CMOS current-mode circuits without static current paths. In addition, an experimental chip has been designed and fabricated using 0.18-μm CMOS technology. The LN-CNN chip with the array size of 20 $\times$ 20 can realize the function of the diamond-shaped large-neighborhood templates. The LN-CNN functions of diffusion, deblurring, and Müller-Lyer illusion have been verified successfully. Meanwhile, the functions of erosion and dilation are expanded with the diamond-shaped LN templates. The total chip area is 1543 μm $\times$ 1248 μm, and the area of a single cell is 33.58 μm $\times$ 43.15 μm. The power is 0.7 mW on standby and

Manuscript received January 04, 2007; revised March 18, 2008. First published October 31, 2008; current version published February 11, 2009. This work was supported by the National Science Council (NSC), Taiwan, under Grant NSC 94-2215-E-009-042.

The authors are with the Nanoelectronics and Gigascale Systems Laboratory, Institute of Electronics, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: cywu@alab.ee.nctu.edu.tw; shawn.ee92g@nctu.edu.tw).

Digital Object Identifier 10.1109/TCSI.2008.2002115

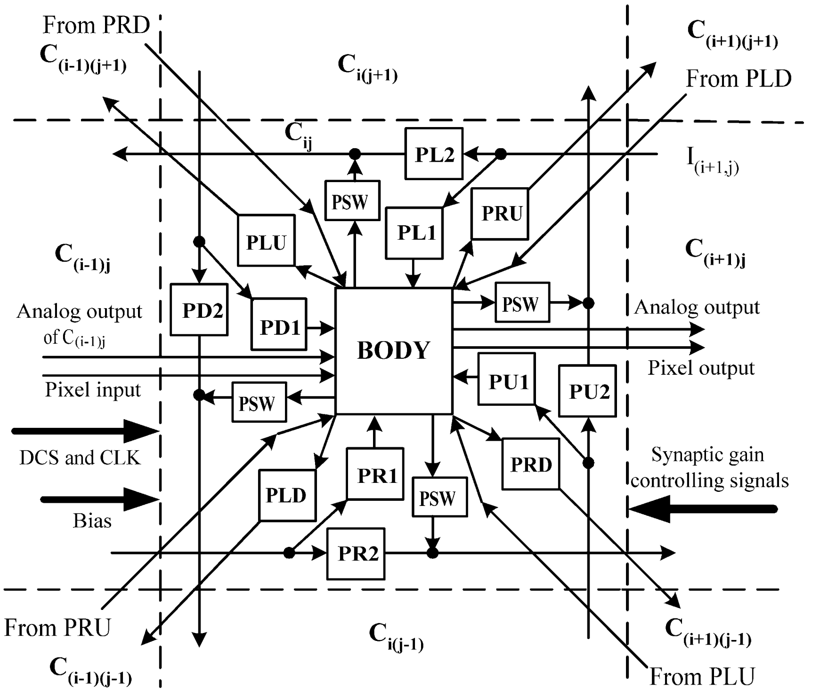


Fig. 1. Architecture of an LN-CNN kernel unit.

18 mW in operation with a 1.8-V supply voltage. As a result, the proposed kernel unit has a very simple structure, small dc power dissipation, and small chip area, which can be applied to the CMOS implementation of an LN-CNNUM with a huge kernel array size. Also, with the hardware of the proposed LN-CNN structure, many new functions or templates of LN-CNN can be explored.

In Section II, the LN-CNN model, the global architecture of the kernel unit of the LN-CNNUM, and the components of each regular cell are described. In Section III, the CMOS circuits of the neuron, synapses, PSW, and analog memory in the proposed LN-CNN are described, and HSPICE simulation results are presented to verify the circuit functions. The overall chip architecture in the design is also illustrated. In Section IV, the measurement results are shown and discussed. Finally, a concluding section is provided.

II. Architecture and Models

For a standard CNN, the state equation is written as [1]–[3]

$$\dot{x}_{ij} = -x_{ij} + Z_{ij} + \sum_{C_{kl}\in S_{ij}} A_{kl}y_{kl} + \sum_{C_{kl}\in S_{ij}} B_{kl}u_{kl} \quad (1)$$

where x_{ij}, y_{ij}, and u_{ij} are the state, output, and input of the neuron cell $\mathbf{C}_{ij}$ in a CNN array, respectively; the coefficient Z_{ij}, called the template $\boldsymbol{Z}$, is the threshold of the neuron cell $\mathbf{C}_{ij}$; and A_{kl} and B_{kl} are the coefficients, called templates $\boldsymbol{A}$ and $\boldsymbol{B}$, which are multiplied with output y_{kl} and input u_{kl} of the cell $\mathbf{C}_{kl}$, respectively, in the sphere of influence (S_{ij}) of the neuron cell $\mathbf{C}_{ij}$. The two sets of products are accumulated over all the cells $\mathbf{C}_{kl}$ in the sphere of influence (S_{ij}) of the neuron cell $\mathbf{C}_{ij}$. Where there are nonzero coefficients for templates $\boldsymbol{A}$ and $\boldsymbol{B}$ at the neighboring cells $\mathbf{C}_{(i\pm r)(j\pm r)}$, r is an integer called neighborhood of radius. If r is greater than one, it is called an LN-CNN.

The architecture of the proposed LN-CNN kernel unit is shown in Fig. 1, where the region surrounded by the broken line represents one neural cell $\mathbf{C}_{ij}$ defined by the coordinate. In $\mathbf{C}_{ij}$ (of Fig. 1), the BODY shown in Fig. 2 consists of the neuron, analog memory, synapses, and control circuits. PU1, PD1, PL1, PR1, PRU, PRD, PLU, PLD, PU2, PD2, PL2, and PR2 are all synapses, which can multiply input signals and result in different gains which are controlled by the synaptic gain controlling signals. As a result, these synapses can be combined to realize the coefficients of templates $\boldsymbol{A}$ and $\boldsymbol{B}$, except the center coefficients A_{ij} and B_{ij}. Among these synapses, PU2, PD2, PR2, and PL2 can propagate signals to the cells farther than the neighboring cells. For example, the signal $I_{(i+1,j)}$ from $\mathbf{C}_{(i+1)j}$ can pass through PL2, be multiplied by the gain of PL2, and then reach $\mathbf{C}_{(i-1)j}$. These connections that are used to realize large-neighborhood templates are called propagating connections. PLU, PLD, PRU, PRD, PL1, PD1, PR1, and PU1 are used to connect the neighboring cells directly. These connections between the nearest neighboring cells are called direct connections. PSW is a current switch, and the gain of PSW is one. The polarities of the signals sent out of the BODY in upward, downward, leftward, and rightward directions are determined by four PSWs. The output current of PSW is combined with that sent from the synapse of the propagating connections in the former cell. Eventually, the resultant output is sent into the synapse of the next cell.

The DCS and CLK in Fig. 1 are digital controlling signal and clock signal, respectively, to control logic circuits and switches in the kernel unit. The Pixel input signal of one cell is connected to the Pixel output signal of the former cell. For example, the Pixel input of $\mathbf{C}_{ij}$ comes from the Pixel output of $\mathbf{C}_{(i-1)j}$. This

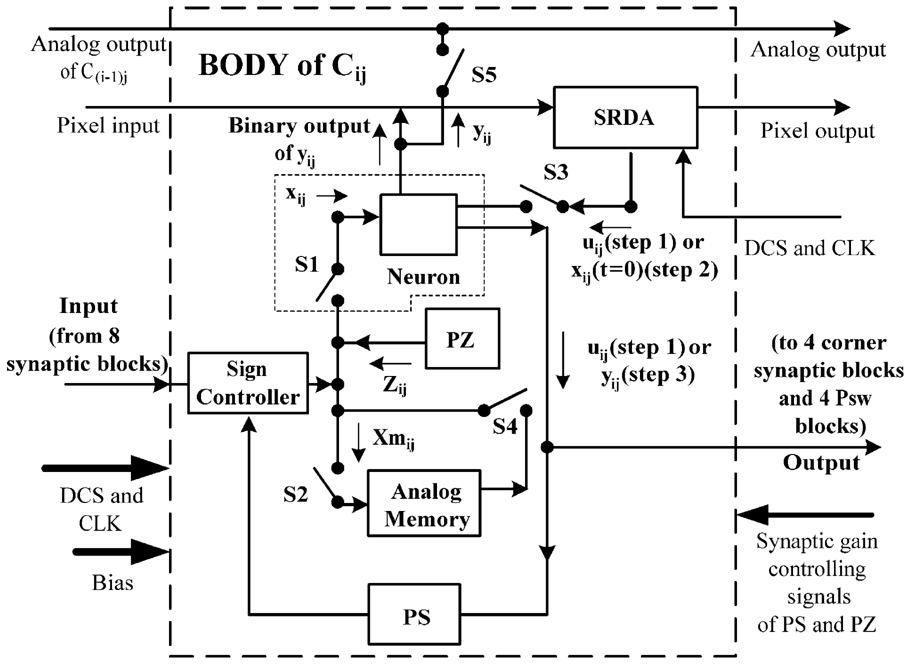

BODY of C$_{ij}$
Analog output of C$_{(i-1)j}$
Pixel input
Binary output of y$_{ij}$
y$_{ij}$
S5
SRDA
Analog output
Pixel output
S3
x$_{ij}$
S1
Neuron
u$_{ij}$(step 1) or x$_{ij}$(t=0)(step 2)
DCS and CLK
Input (from 8 synaptic blocks)
Sign Controller
PZ
Z$_{ij}$
u$_{ij}$(step 1) or y$_{ij}$(step 3)
(to 4 corner synaptic blocks and 4 Psw blocks)
S4
Xm$_{ij}$
Output
S2
Analog Memory
DCS and CLK
Bias
Synaptic gain controlling signals of PS and PZ
PS

Fig. 2. Structure of the BODY in Fig. 1.

signal transfers the input pattern to each cell and the output pattern to the output pads in series. The arrows between the cells are connected to the relative positions of each cell. For example, the arrow line from the PRU of $\mathbf{C}_{ij}$ is connected to the BODY of $\mathbf{C}_{(i+1)(j+1)}$, and similarly, the arrow line from the PLD of $\mathbf{C}_{(i+1)(j+1)}$ is connected to the BODY of $\mathbf{C}_{ij}$.

In the structure of the BODY in Fig. 2, switches S1–S4 are controlled by the signals of DCS and CLK, and switch S5 is controlled by a 5-bit decoder. An SRDA contains one shift register, digital controlling logic, and a 1-bit D/A converter (DAC) inside. The Pixel input of $\mathbf{C}_{ij}$ can be transferred to the next cell by the SRDA. The SRDA provides the binary input signal u_{ij} or the initial state value $x_{ij}(t=0)$ of each cell during the operation. After the operation, the SRDA can store the binary output of y_{ij} from the neuron, and the analog output y_{ij} can be read out by turning on switch S5.

In Fig. 2, the Neuron is a neuron with a standard piecewise linear ramp function

$$y_{ij} = f(x_{ij}) = \frac{1}{2}|x_{ij}+1| - \frac{1}{2}|x_{ij}-1|. \tag{2}$$

The input of the BODY comes from the summation of eight synaptic outputs, as shown in Fig. 1, and the output of the BODY is duplicated eight times and sent to the four PSWs and four corner synapses PRU, PRD, PLU, and PLD. PZ generates the coefficient Z_{ij}, where PS is the synapse that generates the center coefficients A_{ij} and B_{ij} of templates $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively. The Analog Memory is used to store the following equation:

$$Xm_{ij} = Z_{ij} + \sum_{C_{kl}\in S_{ij}} B_{kl}u_{kl}. \tag{3}$$

Before the Neuron, there is a Sign Controller which is used to adjust the polarities of the signals from the nine synapses.

In the first step of the operation period, only the signal of Xm_{ij} in (3) is calculated, sampled, and stored by the Analog Memory. In addition, the digital code of the input u_{ij} is sent from the Pixel input to the shift register in the SRDA and stored. Switches S2 and S3 are closed, and S1 and S4 are left open. At this time, all the synapses are set to certain gains to generate the template $\boldsymbol{B}$, and PZ is set to generate Z_{ij}. The piecewise linear ramp function of the neuron is turned off. The input signal u_{ij} from the SRDA passes through the Neuron. At this moment, the output of the neuron is the same with the input signal u_{ij} from the SRDA, multiplied with the template $\boldsymbol{B}$ and combined with Z_{ij} to form Xm_{ij}, which is instilled into the Analog Memory. After switch S2 is opened, Xm_{ij} is stored in the Analog Memory.

In the second step, the digital code of the initial state $x_{ij}(t=0)$ of the desired function is sent from the Pixel input to the shift register in the SRDA and stored. S1 and S2 are open, and S3 and S4 are closed. Xm_{ij} is read out, and the neuron is set to the initial state $x_{ij}(t=0)$ provided by the SRDA. Meanwhile, the gains of all the synapses are set to generate the template $\boldsymbol{A}$. In the third step of the operation period, the S1 switch is turned on, and the S3 switch is turned off. A feedback loop is constructed, and then, the calculation of (1) is started. After the operation is completed, the readout period commences. The output y_{ij} is converted to binary form, and the binary output is sent to and stored at the shift register in the SRDA. As the input pattern of the next operation is sent into the LN-CNN, the output pattern of the former operation can be read out from the Pixel output of the last cell.

Fig. 3 shows a large-neighborhood template, where symbols from letters a to q represent the template coefficients. The neighborhood of radius r' is redefined, as shown in Fig. 3. Here, the

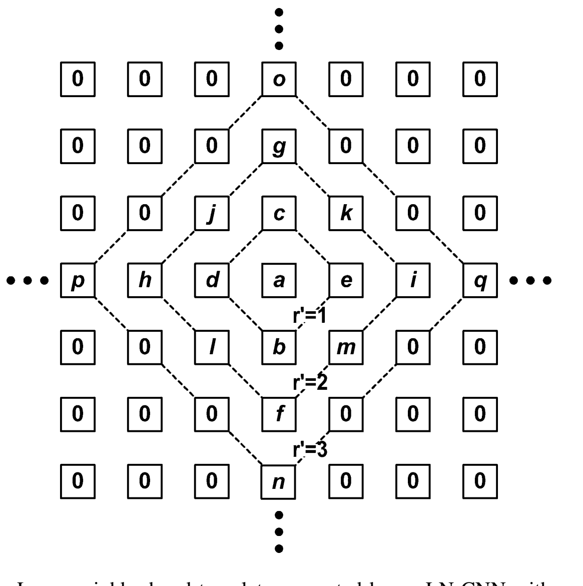


Fig. 3. Large-neighborhood template generated by an LN-CNN with propagating connections.

TABLE I
DERIVED EQUATIONS OF TEMPLATE COEFFICIENTS AND GAINS OF SYNAPSES

Connection Type	Template Coefficients Constructed by the Gains of Synapses		Gain of Each Synapse by Template Coefficients	
Direct Connection	a = PS,	b = PU1	PS = a,	PU1 = b
	c = PD1,	d = PR1	PD1 = c,	PR1 = d
	e = PL1,	j = PRD	PL1 = e,	PRD = j
	k = PLD,	l = PRU	PLD = k,	PRU = l
	m = PLU		PLU = m	
Propagating Connection	f = PU1×PU2		$\text{PR2} = \frac{h}{d} < 1$	
	g = PD1×PD2			
	h = PR1×PR2		$\text{PL2} = \frac{i}{e} < 1$	
	i = PL1×PL2			
	n = PR1×PR2^2		$\text{PU2} = \frac{f}{b} < 1$	
	o = PU1×PU2^2			
	p = PL1×PL2^2		$\text{PD2} = \frac{g}{c} < 1$	
	q = PD1×PD2^2			

sphere of influence $\mathbf{S}_{ij}$ of a large neighborhood is not considered as a 5×5 matrix but is defined as a diamond-shaped matrix in Fig. 3 with neighborhood of radius $r' = 2$. Each coefficient can be derived from the gains of the synapses in Fig. 1 and the PS in Fig. 2. The derived equations are listed in Table I, where the template coefficients in Fig. 3 are expressed by the gains of the synapses, and the gain of each synapse is expressed by the template coefficients. Thus, the architecture in Figs. 1 and 2 can be used to generate the large-neighborhood templates with $r' = 2$ shown in Fig. 3.

According to Table I, the gains of synapses PD2, PU2, PL2, and PR2 of propagating connections should be less than one for each. If the synaptic gain of a propagating connection is larger than or equal to one, then the signal coming from the cells along one direction would diverge. The gains of these synapses of propagating connections can be determined from the template coefficients f, g, h, and i, as listed in Table I. Because of the propagating connections, if the template coefficients f, g, h, and i are not equal to zero, the coefficients o, q, n, and p would not equal zero also, respectively. However, if the template coefficients n, o, q, and p are to be set to zero, the template values f, g, h, and i would be small enough when compared with the template values b, c, d, and e, respectively.

The four corner coefficients j, k, l, and m are determined directly by synapses PRD, PLD, PRU, and PLU, respectively, of direct connections. Similarly, the coefficient a can be generated directly by the PS in Fig. 2.

III. Circuit Implementation and Simulation Results

It has already been established that current-mode signals can be easily combined. In addition, current-mode circuits are faster and consume less power than voltage-mode circuits. Therefore, the proposed LN-CNN has been implemented by using current-mode circuits. In all the current-mode circuit realizations, the signals represented in Figs. 1 and 2 and transferred inside the kernel unit are all in current mode, except the DCS, CLK, synaptic gain controlling signals and the digital logic circuits signals.

A. Neurons and PZ

Fig. 4 shows the circuit of the PZ and the Neuron inside the BODY, as indicated by dotted lines in Fig. 2. The PZ is implemented by devices M_{Z1} and M_{Z2}. The gate bias voltages V_{ZP} and V_{ZN} directly control the current through M_{Z1} and M_{Z2}, respectively, to generate the threshold current I_Z. The circuitry of M_{N1}–M_{N6} is the neuron core with the piecewise linear ramp function. The gate bias voltages $V1$ and $V2$ are used not only to maintain the static current of the neuron zero with devices M_{N4} and M_{N3} but also to limit the currents through M_{N1} with M_{N2} and M_{N6} with M_{N5}, respectively. Furthermore, M_{N3} and M_{N4} also act as switch S1 in Fig. 2. The gate bias voltages $V1$ and $V2$ are controlled by the external bias current $I\text{bias}$. The transfer characteristic of the neuron is simulated, as shown in Fig. 5. The low and high limit currents of the piecewise linear ramp function range from 351.8 to 487.8 nA and from 389.5 to 534.3 nA, respectively, when the $I\text{bias}$ is in the range of 250–360 nA and the supply voltage is 1.8 V. When the neuron is on standby or there is no input current, the leakage current is less than 1 nA. In the first and second steps of the operation period, S1 is turned off, i.e., $M_{N2}-M_{N5}$ are turned off. In this way, the neuron core acts as two current mirrors. As the input current I_u, shown in Fig. 4, is provided by the SRDA in the first step, the current I_{Xm} is calculated, and in the second step, the initial value $I_x(t = 0)$ is also introduced by the SRDA. Moreover, M_{N7} and M_{N8} are used to send the binary outputs to the SRDA or send the transient currents to the analog outputs through S5.

B. Synapses and Sign Controller

The circuit diagrams of the synapses are shown in Fig. 6(a)–(c) and are indicated by broken lines, whereas the circuit diagram of the Sign Controller is demonstrated by broken lines in Fig. 6(d). The circuit of Fig. 6(a) is used to realize synapses PL2, PR2, PD2, and PU2 of propagating connections. There are two paths, i.e., N and P types, in one synapse to deal with the bidirectional current inputs. If an LN-CNN is on standby or there are no input currents, the synapses consume no power. The device pairs $M_{\text{sa}1}/M_{\text{sa}3}$ and

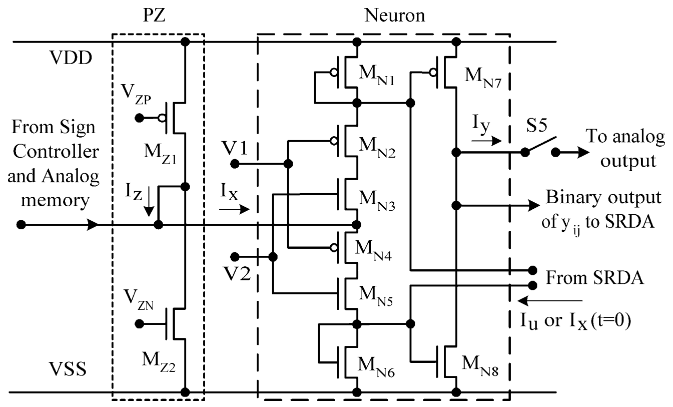

PZ
Neuron
VDD
VSS
From Sign Controller and Analog memory
V1
V2
Iy
S5
To analog output
Binary output of y_ij to SRDA
From SRDA
Iu or Ix(t=0)

Fig. 4. Circuit diagram of the Neuron and PZ in Fig. 2.

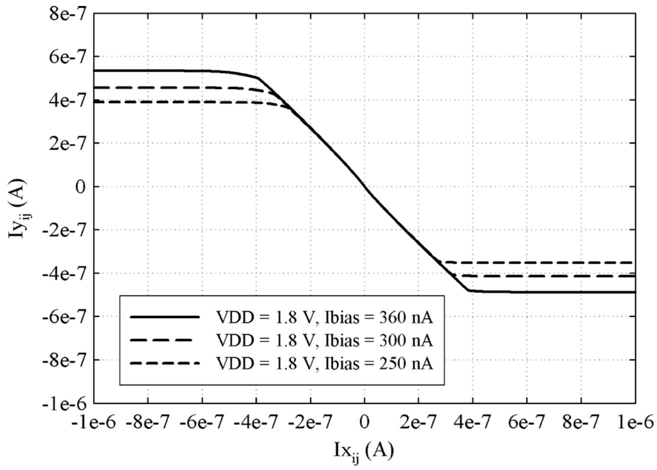


Fig. 5. Transfer characteristic of a neuron with different external bias currents Ibias's.

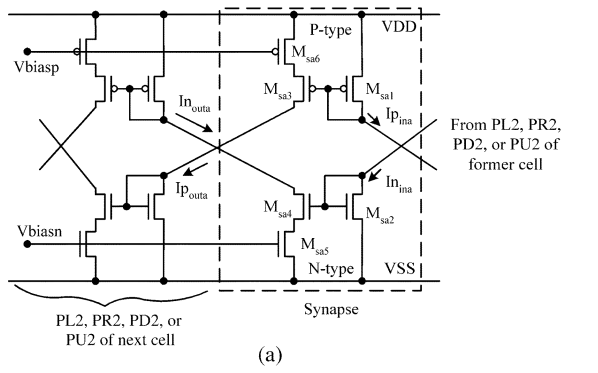
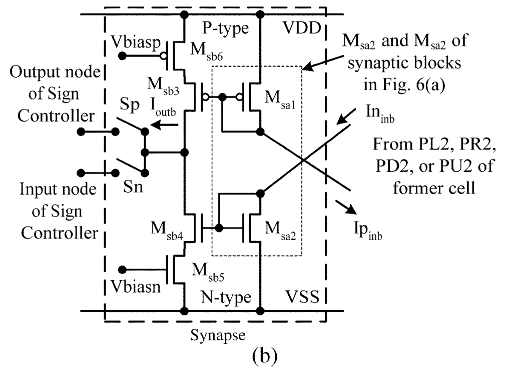
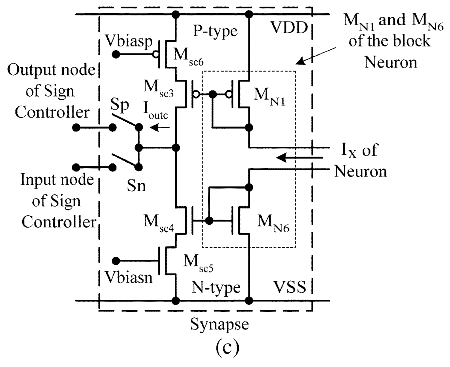
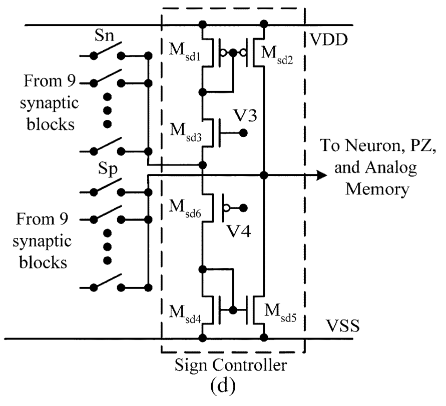




(a)
(b)
(c)
(d)
Synapse
Sign Controller
From PL2, PR2, PD2, or PU2 of former cell
PL2, PR2, PD2, or PU2 of next cell
To Neuron, PZ, and Analog Memory

Fig. 6. Circuit diagrams of (a) synapses PL2, PR2, PD2, and PU2; (b) synapses PL1, PR1, PD1, and PU1; (c) synapses PRU, PRD, PLU, PLD, and PS; and (d) the sign controller.

$M_{\text{sa2}}/M_{\text{sa4}}$ can be seen as two sets of current mirrors, and the maximum gains are determined by the ratios of $M_{\text{sa1}}/M_{\text{sa3}}$ and $M_{\text{sa2}}/M_{\text{sa4}}$. M_{sa6} and M_{sa5} with gate bias voltages $V_{\text{bias}p}$ and $V_{\text{bias}n}$ are operated in the linear region to control the current mirror gains of $M_{\text{sa1}}/M_{\text{sa3}}$ and $M_{\text{sa2}}/M_{\text{sa4}}$, respectively. All the gate bias voltages $V_{\text{bias}p}$ and $V_{\text{bias}n}$ of synapses combined with the gate bias voltages V_{ZP} and V_{ZN} of the PZ form the synaptic gain controlling signals, as shown in Fig. 1. Furthermore, the gate bias voltages $V_{\text{bias}p}$ and $V_{\text{bias}n}$ are generated by using an on-chip 4-bit DAC. There are 16 different values for $V_{\text{bias}p}$ and $V_{\text{bias}n}$. An HSPICE-simulated Inouta versus Inina diagram of the N-type synapse with differing gate bias voltages $V_{\text{bias}n}$'s ranging from 34.4 to 737 mV is shown in Fig. 7. The corresponding N- and P-type current gains of the input current ranging from 300 to 500 nA are shown in Fig. 8, where $M_{\text{sa1}}-M_{\text{sa4}}$ are operated in the subthreshold region with a supply voltage of 1.8 V. The N-type synaptic gains with different $V_{\text{bias}n}$ values range from 0 to 1.54 in the input current range of 300–500 nA, while the P-type synaptic gains with different $V_{\text{bias}p}$ values range from 0 to 1.42. The N-type synaptic gain has an average variation of ±6.38%, while the P-type synaptic gain has that of ±7.72%, as indicated by short bars over the input current range of 300–500 nA. It can be seen that the synapses can generate the desired templates with a tolerable level of error by setting the codes for the $V_{\text{bias}n}$ and $V_{\text{bias}p}$ voltages with proper values.

The circuits of the synapses of direct connections are shown in Fig. 6(b) and (c), and it can be seen that the circuits and operations are similar to those of the synapses of propagating connections. The circuit in Fig. 6(b) realizes synapses PL1, PR1, PD1, and PU1, while that in Fig. 6(c) realizes PLU, PLD, PRU, PRD,

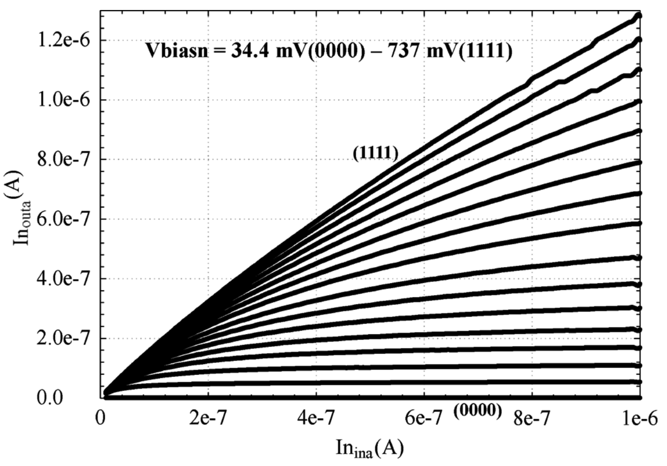


Fig. 7. HSPICE-simulated Inouta versus Inina diagram of the N-type synapse in Fig. 6(a) with 16 different values for $V_{\mathrm{bias}n}$.

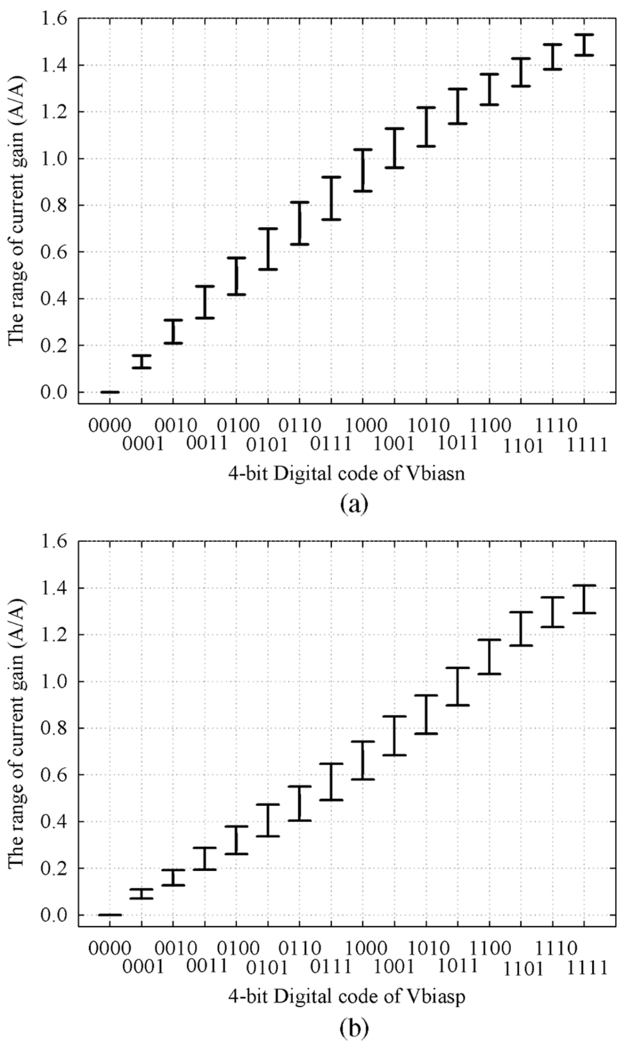


Fig. 8. Range of (a) the N-type and (b) P-type current gains of the synapses with an input current range of 300–500 nA.

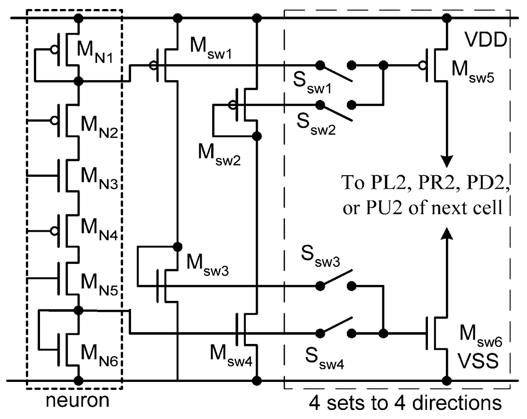


Fig. 9. Circuit diagram of the PSW.

and PS. The P- and N-type synaptic gains of one synapse of direct connections can be set to different values to perform more functions. The synapses shown in Fig. 6(b) share two master devices $M_{\mathrm{sa1}}/M_{\mathrm{sa2}}$ with the synapses of propagating connections, while those shown in Fig. 6(c) share M_{N1}/M_{N6} with the Neuron. The output currents of Fig. 6(b) and (c) are sent to the Sign Controller using switches S_n and S_p to decide the polarities of the signals. The maximum gains of synapses PLU, PLD, PRU, and PRD are set to two, and those of PL1, PR1, PU1, and PD1 are set to four. The gain of synapse PS is set to eight. Through this design, this LN-CNN can generate the templates, as shown in Fig. 3, where the center coefficient a is smaller than eight and the coefficients b, c, d, and e are smaller than four, while the coefficients j, k, l, and m are smaller than two.

The circuitry of the Sign Controller is shown in Fig. 6(d), where switches S_n and S_p of the nine synapses used to adjust the polarity of the signals from the synapses are also drawn. Devices M_{sd3} and M_{sd6} with gate bias voltages $V3$ and $V4$, respectively, maintain the static current from M_{sd1} to M_{sd4} at zero level. $M_{\mathrm{sd1}}/M_{\mathrm{sd2}}$ and $M_{\mathrm{sd4}}/M_{\mathrm{sd5}}$ are the current mirrors used to invert the direction of the current flow. If the polarity of the input signal from the synapses is negative, S_p is turned off, and the input signal enters the neuron or analog memory through switch S_n and the Sign Controller. However, in the same situation, if the input signal is positive, S_n is turned off, and the signal enters the neuron through the S_p switch.

C. PSWs

Each of the synapses contains one pair of switches S_n and S_p to control the signal polarities, except the synapses of propagating connections. Hence, to confirm that the output signals sent out of the BODY and those sent out of the synapses of propagating connections have the same polarities, the PSW has been added to achieve this purpose.

Fig. 9 shows the circuit diagram of the PSW. The output currents of the neuron are mirrored through M_{sw1} and M_{sw4} to generate the gate voltages on M_{sw2} and M_{sw3}, respectively. The current through M_{sw5}, where the gate is connected to the gate of $M_{N1}(M_{\mathrm{sw2}})$, is opposite to the current through M_{sw6}, whose gate is connected to the gate of $M_{\mathrm{sw3}}(M_{N6})$. The polarity of the output current in the PSW is selected using switches $S_{\mathrm{sw1}}-S_{\mathrm{sw4}}$. For a positive (negative) output of the PSW, switches S_{sw1} and S_{sw4} (S_{sw2} and S_{sw3}) are closed, and at the same time, switches S_{sw2} and S_{sw3} (S_{sw1} and S_{sw4}) are opened. There are four PSWs containing switches $S_{\mathrm{sw1}}-S_{\mathrm{sw4}}$

TABLE II
COMPARISON OF DEVICE NUMBERS AND INTERCONNECTION LINES

		This Work with Propagating Connection	LN-CNN (r' = 2) with Direct Connection Using the Circuit Structure in [9]
Device Number	Synapse of Direct Connections	8.67	8
	Synapse of Propagating Connections	13	-
	Neuron Core	8	10
	Neuron Cell	130	104
Interconnection Lines cross One Cell		8	12

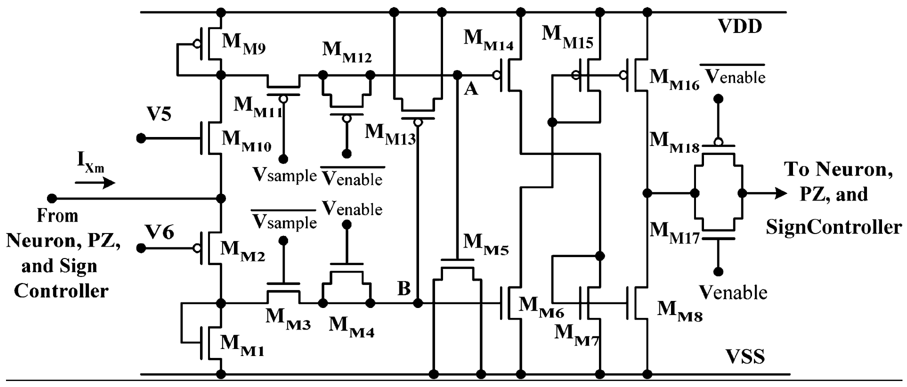


Fig. 10. Circuit diagram of the analog memory.

and $M_{\text{sw5}}-M_{\text{sw6}}$, as shown in Fig. 1, and these four PSWs share the circuits of $M_{\text{sw1}}-M_{\text{sw4}}$.

A comparison of the device numbers and interconnection lines of the kernel unit between the proposed structure and the LN-CNN with direct connection using the circuit structure in [9] is given in Table II. As can be seen from Table II, the LN-CNN with direct connections needs 12 connections, including four connections to the farther neighboring cells. In the proposed structure, more devices are required; however, as each cell only has eight connections to the nearest eight neighboring cells, this facilitates the IC implementation.

D. Analog Memory

Fig. 10 shows the circuit diagram of the analog memory, where M_{M1} and M_{M9} are used to generate the gate voltages of M_{M6} and M_{M14}, respectively, from the input current I_{Xm}. The gate voltages are stored at node A (B) by turning off $M_{M11}(M_{M3})$ with the signal Vsample (with the complementary signal of Vsample). M_{M4} and M_{M12} are used to compensate for the charge injections and the clock feedthrough from M_{M3} and M_{M11}, respectively. M_{M5} and M_{M13} are used to increase the gate–source capacitance C_{gs} of M_{M6} and M_{M16}, respectively, in order to suppress the sampling error. The current mirror $M_{M7}/M_{M8}(M_{M15}/M_{M16})$ is used to isolate storage node A (B) from the output node of analog memory so that the stored voltage is not affected by the voltage change at the output node. As the analog memory is read out, the signal Venable (the complementary signal of Venable) turns on $M_{M17}(M_{M18})$, and at the same time, it also turns on the compensational function of $M_{M4}(M_{M12})$. Furthermore, devices M_{M2} and M_{M10} with gate bias voltages $V5$ and $V6$, respectively, maintain the static current from M_{M9} to M_{M1} at zero level and also act as switch S2, as can be seen in Fig. 2.

E. Overall Chip Architecture

Fig. 11 shows the architecture of the whole system, where the size of the kernel unit array is 20 × 20. There are 5 × 54 shift registers to store the digital codes of synaptic gain controlling signals. The digital codes of each synaptic gain controlling signal are stored in a 4-bit shift register for absolute value and a 1-bit shift register for polarity. However, one synapse requires two synaptic gain controlling signals, and the signals have different values when templates $\boldsymbol{A}$ and $\boldsymbol{B}$ are generated. Hence, there are 5 × 52 shift registers required for templates $\boldsymbol{A}$ and $\boldsymbol{B}$. For the synaptic gain controlling signals of template $\boldsymbol{Z}$, a 6-bit register is required for the absolute value of template $\boldsymbol{Z}$, and a 1-bit shift register is used for its polarity. Thus, 5 × 2 1-bit shift registers are required for template $\boldsymbol{Z}$. The signal from the Digital Controlling Circuit determines whether the Generation Circuit for templates $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{Z}$, which has 28 DACs, generates synaptic gain controlling signals for either template $\boldsymbol{A}$ or templates $\boldsymbol{B}$ and $\boldsymbol{Z}$. The external bias current Ibias generates the bias currents and voltages required in the system, particularly the bias voltages $V1-V6$ inside the Neuron, Sign Controller, and Analog Memory, as shown in Fig. 1. The signals ***Input_Enable*** and ***Weight_Enable*** with external clock signal

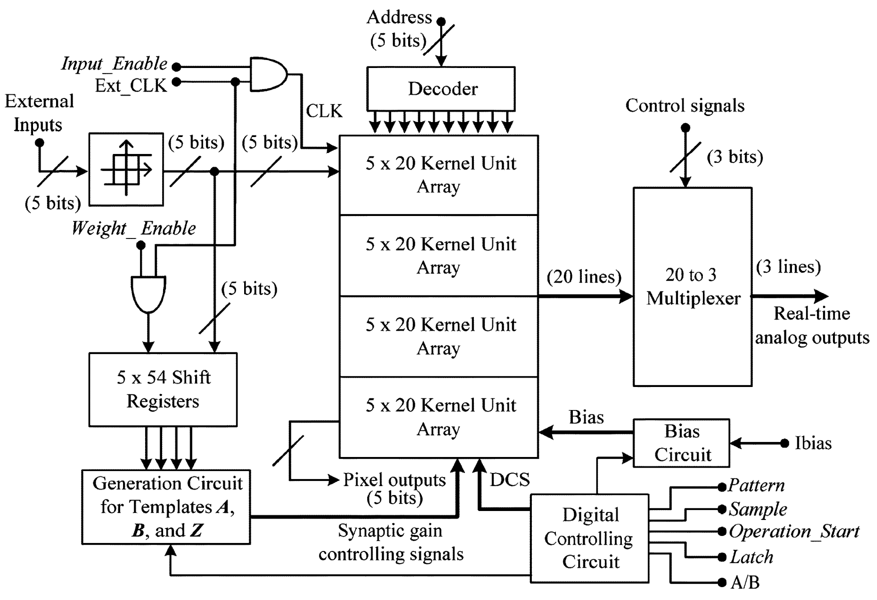


Fig. 11. Architecture of the 20 × 20 LN-CNN system.

B:

-0.1	-0.1	-0.1	-0.1	-0.1
-0.1	-0.1	-0.1	-0.1	-0.1
-0.1	-0.1	1.3	-0.1	-0.1
-0.1	-0.1	-0.1	-0.1	-0.1
-0.1	-0.1	-0.1	-0.1	-0.1

A:

0	0	0	0	0
0	0	0	0	0
0	0	1.3	0	0
0	0	0	0	0
0	0	0	0	0

Z:

-2.8

Fig. 12. 5 × 5 templates B, A, and Z for Müller-Lyer illusion [3].

B:

0	0	0	-0.06	0	0	0
0	0	0	-0.26	0	0	0
0	0	-0.83	-1.18	-0.83	0	0
-0.06	-0.26	-1.18	+2.73	-1.18	-0.26	-0.06
0	0	-0.83	-1.18	-0.83	0	0
0	0	0	-0.26	0	0	0
0	0	0	-0.06	0	0	0

A:

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	+2.96	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Z:

-8

Fig. 13. Extracted values of the diamond-shaped template from an HSPICE postlayout simulation.

Ext_CLK are used to determine whether the external input signals are input and initial patterns or the digital codes of synaptic gain controlling signals, respectively. In the array, 5-bit binary signals in one clock cycle are sent into the LN-CNN and read out from the Pixel outputs. From 20 neuron analog output signals of one column selected by a 5-bit decoder, three real-time neuron analog output signals can be read out using a 20-to-3 multiplexer.

In the first step, both input pattern and digital codes of the templates $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{Z}$ are ready for operation, and the signal $\boldsymbol{A/B}$ is set to Low first to cause the template generation circuit to generate synaptic gain controlling signals of the templates $\boldsymbol{B}$ and $\boldsymbol{Z}$. Meanwhile, the function of the neuron in the kernel unit is turned off. The signal ***Pattern*** goes to High in order to inject the input pattern into all the neurons. The result of the first step is sent into the analog memory and stored after the signal ***Sample*** is enabled, and then, the signal ***Pattern*** returns to Low.

In the second step, the pattern in the shift register of the SRDA is replaced by the initial pattern of the desired function. The initial pattern in the shift register is then sent to the neurons as the initial values by enabling the signal ***Pattern*** again. Meanwhile, the signal $\boldsymbol{A/B}$ is set to High so that the template $\boldsymbol{A}$ is generated by the synapses as the template generation circuit generates the synaptic gain controlling signals of the template $\boldsymbol{A}$. In the third

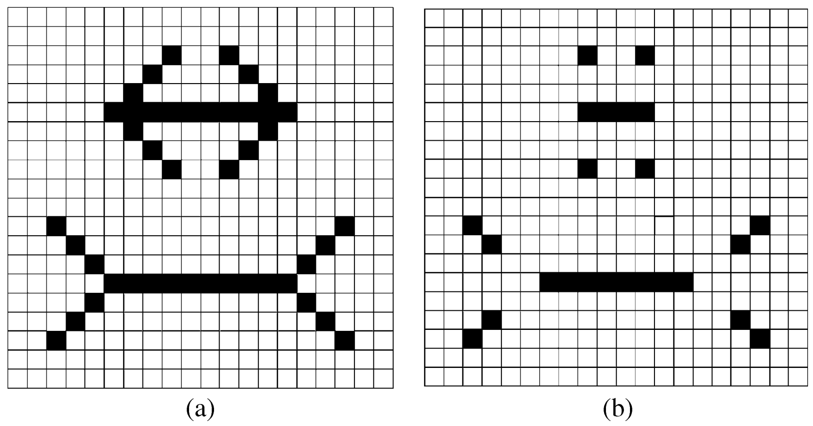


Fig. 14. (a) Input patterns of Müller-Lyer illusion. (b) Resultant output pattern of Müller-Lyer illusion from the HSPICE simulation result.

5 x 5 template

0.01	0.03	0.0425	0.03	0.01
0.03	0.045	0.06	0.045	0.03
0.0425	0.06	0.13	0.06	0.0425
0.03	0.045	0.06	0.045	0.03
0.01	0.03	0.0425	0.03	0.01

diamond-shaped template

0	0	0	0.03	0	0	0
0	0	0	0.0425	0	0	0
0	0	0.045	0.06	0.045	0	0
0.03	0.0425	0.06	0.13	0.06	0.0425	0.03
0	0	0.045	0.06	0.045	0	0
0	0	0	0.0425	0	0	0
0	0	0	0.03	0	0	0

(a)

5 x 5 template

-0.05	-0.2	-0.3	-0.2	-0.05
-0.2	-0.5	-0.6	-0.5	-0.2
-0.3	-0.6	0	-0.6	-0.3
-0.2	-0.5	-0.6	-0.5	-0.2
-0.05	-0.2	-0.3	-0.2	-0.05

diamond-shaped template

0	0	0	-0.08	0	0	0
0	0	0	-0.3	0	0	0
0	0	-0.3	-0.5	-0.3	0	0
-0.08	-0.3	-0.5	0	-0.5	-0.3	-0.08
0	0	-0.3	-0.5	-0.3	0	0
0	0	0	-0.3	0	0	0
0	0	0	-0.08	0	0	0

(b)

Fig. 15. Template B of 5 $\times$ 5 and diamond-shaped templates of (a) diffusion [19] and (b) deblurring [3].

step, the signal ***Operation_Start*** is enabled, and the signal ***Pattern*** is disabled to turn off the initial values. The function of the neurons is turned on to start the overall calculation of template ***A*** with the signals read out from the analog memories. After the outputs are stable, the binary output pattern can be stored in the SRDA as the signals ***Latch*** and ***Input_Enable*** are set to High. When the next input pattern comes in, all the digital signals are disabled, except the signal ***Input_Enable***, and the output pattern can be read out from the 5-bit Pixel outputs.

F. HSPICE Simulation Results

The proposed LN-CNN circuit was designed using 0.18-μm CMOS technology. The HSPICE postlayout simulation was performed with a 20 $\times$ 20 kernel cell array to verify the circuit functions. The function of Müller-Lyer illusion with the 5 $\times$ 5 large-neighborhood template [3], as shown in Fig. 12, was adopted. According to the original 5 $\times$ 5 template, the predicted signs of each diamond-shaped template are set in Fig. 13. Only the center coefficients A_{ij} and B_{ij} are positive, while the others are negative in Fig. 12, so it is reasonable that only the center coefficients in the diamond-shaped template are set to positive. The input pattern of Müller-Lyer illusion is shown in Fig. 14(a). After the HSPICE simulation, the resultant output pattern is shown in Fig. 14(b), where the upper line with outward arrows becomes shorter than the lower line with inward arrows after illusion. The function cannot be realized by a 3 × 3 neighborhood template. The coefficients of the diamond-shaped large-neighborhood template in Fig. 3 were extracted from the postlayout simulation results directly and are shown in Fig. 13, which has the same signs as those in Fig. 12.

The simulated standby power consumption is about 1.148 mW, where a 1.8-V supply voltage is used. The external bias current is 360 nA. The kernel unit array only consumes 1 μW, which accounts for about 0.087% of the overall standby power consumption. As the array is extended to 128 × 128, the standby power consumption is about 7.35 mW and is dominated by the peripheral circuits.

G. Software Simulation Results

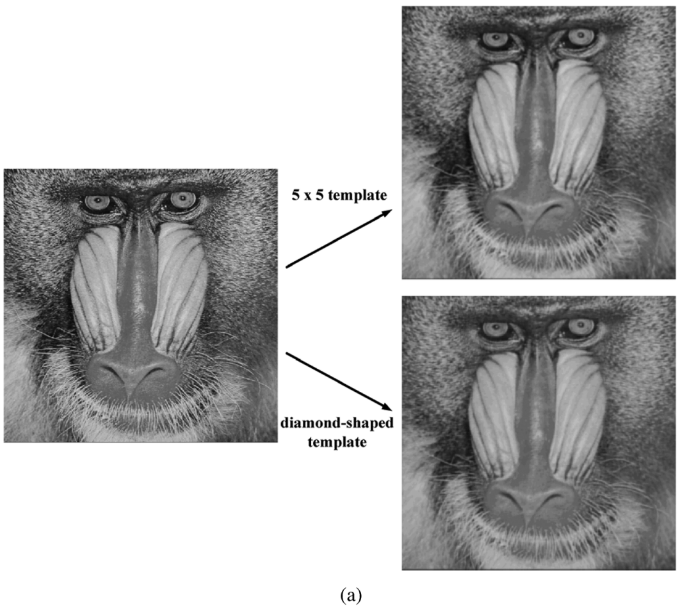
The published large-neighborhood templates are limited. Among the four published LN-CNN templates [3], [18], [19], only one template [18] cannot be implemented by using the proposed structure, since it violates the constraint. Other LN templates for diffusion, deblurring, and Müller-Lyer illusion have been successfully verified. The 5 × 5 template **B** of diffusion [19] and deblurring [3] are approximated by the proposed diamond-shaped templates, as shown in Fig. 15(a) and (b). The coefficients of templates A and Z are zero for diffusion. For deblurring, the center coefficient of template A of 5 × 5 template is ten, and that of diamond-shaped template is seven. Templates **Z** of both functions are zero. The input pattern and simulation results of diffusion and deblurring are shown in Fig. 16(a) and (b), respectively. It is shown that the diamond-shaped template can realize the function of 5 × 5 templates correctly.

The diamond-shaped LN templates can also realize some operations of binary images in one step which can be realized by the 3 × 3 neighborhood templates in two steps. The erosion and dilation functions with 3 × 3 neighborhood templates can contract and expand the edges of images by one pixel, respectively. However, the diamond-shaped LN templates can reinforce the functions to contract or expand the edges by two pixels. Fig. 17 shows the function of erosion where the boundary cells are set to be white (−1). For dilation, it can be realized by the same templates of erosion by making template Z positive. In addition, these two functions with the diamond-shaped LN templates cannot be achieved with 3 × 3 neighborhood templates in one step. Two iterations with 3 × 3 neighborhood templates are required to realize the same functions. Thus, it takes more time and energy.

IV. Experimental Results

An experimental LN-CNN chip has been fabricated using 0.18-μm CMOS technology. The whole chip area is 1543 μm × 1248 μm, where the unit cell is 33.58 μm × 43.15 μm. Fig. 18 shows the photograph of the fabricated LN-CNN chip.

The input image pattern in Fig. 14(a) was used to verify the illusion function of the fabricated LN-CNN. The digital codes of the synaptic gain were adjusted to achieve the suitable value. The binary output pattern was read out from the 5-bit pixel outputs, as shown in Fig. 11. The analog current-mode transients can be read out from the three real-time analog outputs in Fig. 11 using the transimpedance amplifiers outside of the chip. When the analog output current is zero, the output voltage of the transimpedance amplifier is 0.9 V. Since most pixels in the input pattern shown in Fig. 14(a) are in white and all the white pixels remain in white after processing, the N-type synaptic gain of the PS is set to a larger value than the P-type synaptic gain in the measurement. In this way, the problems of variation in the process can be overcome.



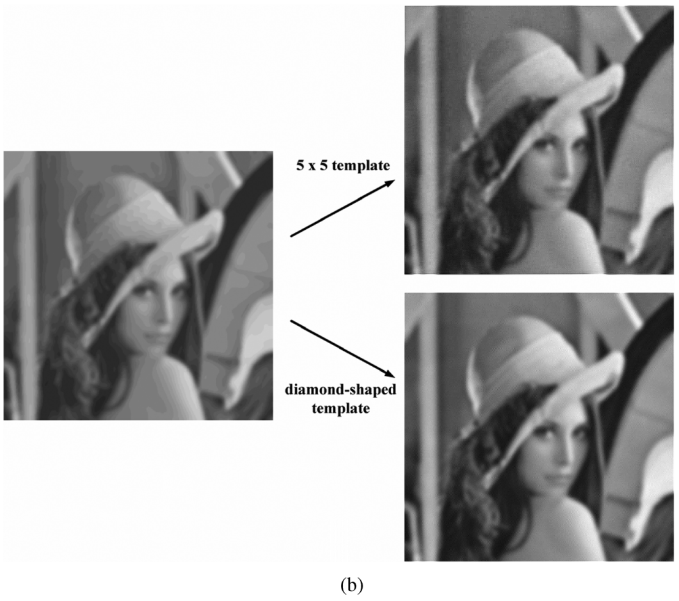
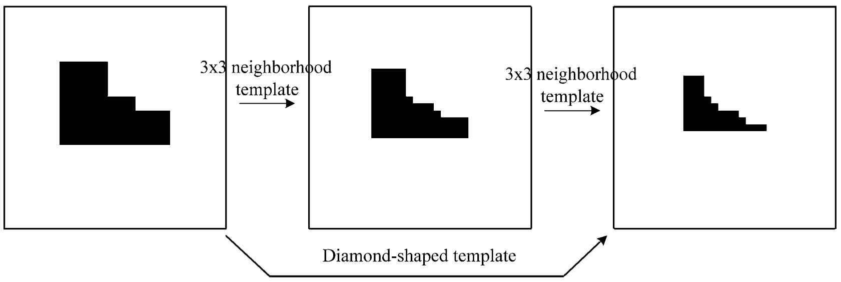


Fig. 16. Input patterns and simulation results of (a) diffusion and (b) deblurring.

The measured binary output pattern is shown in Fig. 19. The experimental result is the same with the postlayout simulation result, except Pixel A which is black in the simulation results of Fig. 14(b). The reason for the error is that the bias current of Pixel A is too small due to process variation. Thus, the self-feedback of Pixel A cannot keep Pixel A in the black state.

The measured analog output voltage of Pixel B through the transimpedance amplifier is shown in Fig. 20. The step signal is the signal ***Operation_Start***, as shown in Fig. 11. As the signal ***Operation_Start*** rises, the analog output remains nearly at 0 V within about 1 μs. Then, it starts to rise and reaches 0.9 V at



Functions	Erosion		
	B Template	A Template	Z
3 x 3 Neighborhood Template	$\begin{array}{ccc} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{array}$	$\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{array}$	-4.5
Diamond Template	$\begin{array}{ccccccc} 0 & 0 & 0 & 0.13 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 & 0 \\ 0.13 & 0.5 & 2 & 1 & 2 & 0.5 & 0.13 \\ 0 & 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.13 & 0 & 0 & 0 \end{array}$	$\begin{array}{ccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$	-15.5

Fig. 17. Input and output patterns of erosion with 3 $\times$ 3 neighborhood templates [3] in two iterations and with diamond-shaped templates in one iteration.

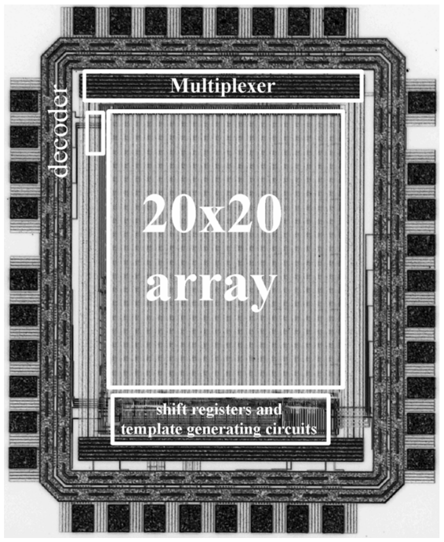


Fig. 18. Photograph of the fabricated 20 $\times$ 20 LN-CNN chip.

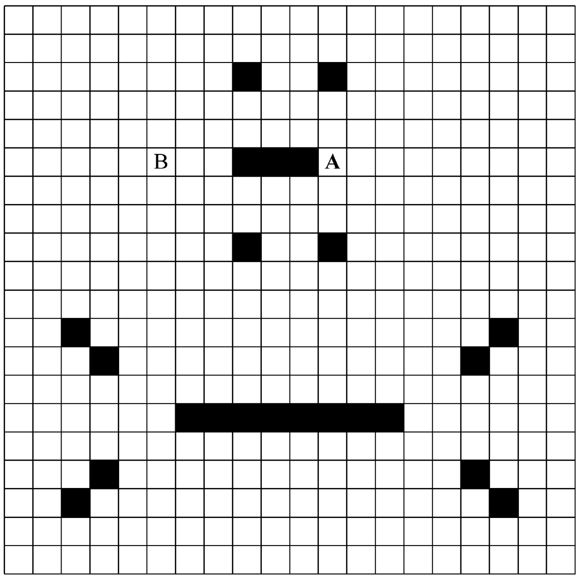


Fig. 19. Experimental resultant output pattern of Müller-Lyer illusion.

about 2 μs. Finally, it takes 3 μs to achieve the overall operation from the black to the white state. The measured transient response time is 3 μs. From the result of post simulation, the transient response operation time is less than 0.1 μs without the transimpedance amplifier. The difference is due to the large loading effect of the transimedance amplifier.

In the experimental result, the overall power consumption was about 0.7 mW on standby and 18 mW during the operation in the third step. The comparisons of power dissipation and energy consumption per cell in the proposed LN-CNN with those in CNNUC3 [10] and ACE16K [11] are listed in Table III. As may be seen in Table III, the cell in the LN-CNN has lower power dissipation and energy consumption.

V. Conclusion

In this paper, a new architecture of LN-CNN has been proposed. In the proposed LN-CNN, the propagating connections are utilized to generate diamond-shaped large-neighborhood templates. In such a connected network, each neuron cell only needs to contact the neighboring cells without the need for

TABLE III
COMPARISON OF LN-CNN WITH CNNUC3 [10] AND ACE16K [11]

	This Work with Propagating Connection	[10]	[11]
Technology @ Supply	0.18μm 6M -1P @ 1.8V	0.5μm 3M -1P @ 3.3V	0.35μm 5M -1P @ 3.3V
Power Dissipation (per cell)	45μW	250μW	180μW
Processing Speed	20 MHz	10 MHz	30 MHz
Energy Consumption (per cell)	2.25 pJ	25 pJ	6 pJ
Cell Size	33.58 x 43.15μm^2	102.2 x 120μm^2	73.3 x 75.7μm^2

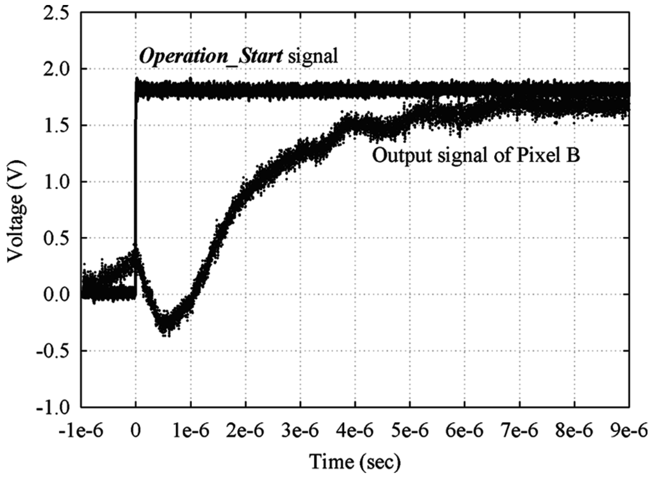


Fig. 20. Experimental results of Pixel B with the signal ***Operation_Start***.

farther interconnections. Therefore, such network architecture is suitable for VLSI implementation. Moreover, by separating the synapses into N- and P-type parts without static currents, the static power dissipation can be reduced to a minimum level. Moreover, during such an operation, the synapses of direct connections with different N- and P-type synaptic gains can also offer more functions. The connections can also be implemented in both horizontal and vertical directions and in diagonal directions to realize the circular symmetric templates. Furthermore, the LN-CNN functions of diffusion, deblurring, and Müller-Lyer illusion have also been verified successfully. With the proposed LN-CNN structure using propagating connections, many new applications and LN-CNN templates can be explored.

A CMOS LN-CNN chip with a 20 $\times$ 20 kernel unit array has been fabricated in 0.18-μm CMOS technology. From the experimental results of this paper, it can be seen that the 5 $\times$ 5 template of Müller-Lyer illusion is reconstructed into a diamond-shaped LN template, and the function has been successfully realized using the LN-CNN and with a chip power consumption of 0.7 mW on standby and 18 mW in operation.

Further research on the universal machine for LN-CNN needs to be conducted for various applications to be realized.

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, no. 10, pp. 1257–1272, Oct. 1988.

[2] L. O. Chua and L. Yang, "Cellular neural networks: Applications," *IEEE Trans. Circuits Syst.*, vol. 35, no. 10, 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. Interdiscip. 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, no. 3, pp. 163–173, Mar. 1993.

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

[8] S. Espejo, R. Carmona, R. Domínguez-Castro, and A. Rodríguez-Vázquez, "A CNN universal chip in CMOS technology," *Int. J. Circuit Theory Appl.*, vol. 24, no. 1, 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] A. Rodriguez-Vazquez, G. Linan-Cembrano, L. Carranza, E. Roca-Moreno, R. Carmona-Galan, F. Jimenez-Garrido, R. Dominguez-Castro, and S. E. Meana, "ACE16k: The third generation of mixed-signal SIMD-CNN ACE chips toward VSoCs," *IEEE Trans. Circuits Syst.*, vol. 51, no. 5, pp. 851–863, May 2004.

[11] G. Linan, S. Espejo, R. Dominguez-Castro, E. Roca, and A. Rodriguez-Vazquen, "CNNUC3: A mixed-signal 64 $\times$ 64 CNN universal chip," in *Proc. Conf. 7th Int. Microelectron. Neural, Fuzzy Bio-Inspired Syst. MicroNeuro*, Apr. 7–9, 1999, pp. 61–68.

[12] G. Linan, S. Espejo, R. Dominguez-Castro, and A. Rodriguez-Vazquen, "The CNNUC3: An analog I/O 64 $\times$ 64 CNN universal machine chip prototype with 7-bit analog accuracy," in *Proc. 6th IEEE Int. Workshop CNNA*, May 23–25, 2000, pp. 201–206.

[13] C. Y. Wu and W. C. Yen, "The neuron-bipolar junction transistor (v-BJT)—A new device structure for VLSI neural network implementation," in *Proc. IEEE Int. Conf. Electron., Circuits Syst.*, Sep. 1998, vol. 3, pp. 277–280.

[14] W. C. Yen and C. Y. Wu, "A new compact neuron-bipolar cellular neural network structure with adjustable neighborhood layers and high integration level," in *Proc. IEEE Int. Conf. Electron., Circuits Syst.*, Sep. 1999, vol. 2, pp. 713–716.

[15] W. C. Yen and C. Y. Wu, "The design of neuron-bipolar junction transistor (νBJT) cellular neural network (CNN) structure with large neighborhood templates," in *Proc. 6th IEEE Int. Workshop Cellular Netw. Appl.*, Catania, Italy, May 2000, pp. 195–200.

[16] G. Timar and C. Rekeczky, "A real-time multitarget tracking system with robust multichannel CNN-UM algorithms," *IEEE Trans. Circuits Syst.*, vol. 52, no. 7, pp. 1358–1371, Jul. 2005.

[17] C.-T. Lin, C.-H. Huang, and S.-A. Chen, "CNN-based hybrid-order texture segregation as early vision processing and its implementation on CNN-UM," *IEEE Trans. Circuits Syst.*, vol. 54, no. 10, pp. 2277–2287, Oct. 2007.

[18] K. Slot, "Large-neighborhood templates implementation in discrete-time CNN universal machine with a nearest-neighbor connection pattern," in *Proc. 3rd IEEE Int. Workshop CNNA*, Dec. 18–21, 1994, pp. 213–218.

[19] N. A. Fernandez, D. L. Valarino, V. M. Brea, and D. Cabello, "On the emulation of large-neighborhood templates with binary CNN-based architectures," in *Proc. 9th Int. Workshop Cellular Neural Netw. Appl.*, May 28–30, 2005, pp. 274–277.

[20] C. Y. Wu and W. C. Yen, "A new compact neuron-bipolar junction transistor (νBJT) cellular neural network (CNN) structure with programmable large neighborhood symmetric templates for image processing," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 48, no. 1, pp. 12–27, Jan. 2001.

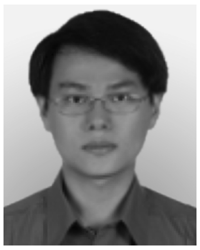
[21] C. H. Cheng, S. H. Chen, L. J. Lin, K. H. Huang, and C. Y. Wu, "A new structure of large-neighborhood cellular nonlinear network (LN-CNN)," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2003, pp. 1497–1501.

[22] S. H. Chen and C. Y. Wu, "A low power design on diffusive interconnection large-neighborhood cellular nonlinear network for giga-scale system application," in *Proc. 11th IEEE Int. Conf. Electron., Circuits Syst.*, Dec. 2004, pp. 179–182.

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, where he is currently with the Nanoelectronics and Gigascale Systems Laboratory, Institute of Electronics. From 1991 to 1995, he was the Director of the Division of Engineering and Applied Science, National Science Council (NSC), 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 postdoctoral research with the University of California, Berkeley. He has authored or coauthored over 250 technical papers in international journals and conferences. He is the holder of 19 patents, including nine U.S. patents. His research interests include nanoelectronics, biomedical devices and system, neural vision sensors, RF circuits, and computer-aided design and analysis.

Dr. Wu is a member of Eta Kappa Nu and Phi Tau Phi. He was the recipient of the 1998 IEEE Fellow Award and the 2000 Third Millennium Medal. He has also been the recipient of numerous research awards presented by the Ministry of Education, 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, where he is currently working toward the Ph.D. degree in the Nanoelectronics and Gigascale Systems Laboratory, Institute of Electronics.

His current research interests include the design of CMOS cellular nonlinear network circuits.