Time multiplexed color image processing based on a CNN with cell-state outputs

Time Multiplexed Color Image Processing Based on a CNN with Cell-State Outputs

Lei Wang, *Member, IEEE,* José Pineda de Gyvez, *Member, IEEE,*
and Edgar Sánchez-Sinencio, *Fellow, IEEE*

***Abstract*— A practical system approach for time-multiplexing cellular neural network (CNN) implementations suitable for processing large and complex images using small CNN arrays is presented. For real size applications, due to hardware limitations, it is impossible to have a one-on-one mapping between the CNN hardware cells and all the pixels in the image involved. This paper presents a practical solution by processing the input image, block by block, with the number of pixels in a block being the same as the number of CNN cells in the array. Furthermore, unlike other implementations in which the output is observed at the hard-limiting block, the very large scale integrated (VLSI) architecture hereby described monitors the outputs from the state node. While previous implementations are mostly suitable for black and white applications because of the thresholded outputs, our approach is especially suitable for applications in color (gray) image processing due to the analog nature of the state node. Experimental complementary metal–oxide–semiconductor (CMOS) chip results in good agreement with theoretical results are presented.**

***Index Terms*— Analog VLSI cellular neural networks (CNN), high-performance image processing applications, neural networks, PC interfaces.**

I. Introduction

SINCE their introduction in 1988, cellular neural networks (CNN) have shown a vast computing power, especially for image processing [1]–[4]. A number of VLSI implementations of CNN analog neural networks have been proposed in recent years [15], [18], [23]. These implementations include transconductance-mode-based processing elements [8], discrete-time implementations [5], [11], [12], switched-current signal processing elements [14], and the current-mode [13] implementation. Each kind of CNN realization has its own advantages and disadvantages. Worth mentioning is that the silicon area and power dissipation are greatly reduced because of the tradeoffs between precision and area, or power dissipation.

For example, the discrete-time CNN can yield "exact" template weights and time constant, but it often takes more area and consumes more power [11], [12]. Early CNN implementations were designed to perform one specific function in image processing or classification, such as edge detection [2], [3], connected component detection, or hole filling. More recently, the ability to change template values electronically, has been studied in detail [6], [7], [10], [23], [24]. Furthermore, in some implementations the activation function of some CNN chips is also tunable, e.g., is programmable [16], [17], the slope and the threshold of the activation function. The basic equation of a CNN cell is [1], [2]

$$C_x \frac{dx_{ij}(t)}{dt} = -\frac{1}{R_x} x_{ij}(t) + \sum_{C(k,l)\in N_r(i,j)} A(i,j;k,l) y_{kl}(t) + \sum_{C(k,l)\in N_r(i,j)} B(i,j;k,l) u_{kl}(t) + I \quad (1)$$

$$y_{ij}(t) = \tfrac{1}{2}\left[|x_{ij}(t)+1| - |x_{ij}(t)-1|\right] \quad (2)$$

where x_{ij} is the state of cell $C(i,j)$, $x_{ij}(0)$ is the initial condition of the cell, C_x and R_x conform the integration time constant of the system, and I is an independent bias constant. $A(i,j;k,l)$ and $B(i,j;k,l)$ are space invariant programming templates for all cells $C(k,l)$ in the neighborhood $N(i,j)$ of cell $C(i,j)$; u_{kl} represents the external input and y_{ij} represents the output equation, i.e., the activation function applied to the cell's state.

One common feature of currently available CNN circuits is that the output signals are the feedback outputs of the cells, and those output values are confined as binary values [1]. Hence, the output image is a black and white image even when the CNN, in its nature, is an analog and continuous signal processing system. The binary output values of the CNN are the positive or negative threshold of the activation function. Due to this nonlinear function, the feedback output of a cell can converge to either a positive or negative value under some well studied "stability conditions," [1] such as $A(i,i) > 1$. This characteristic makes the CNN very attractive for some pattern extracting applications, such as edge detection and connected element detection where a binary valued output image is acceptable. Moreover, the circuit design is relatively easy if the output is just binary rather than continuous, since the linearity, precision, and offsets of the output values are not relevant [9], [13], [15], [23] because the "x" state observed before the activation function is not of critical importance. However, in some cases, the binary output CNN does not carry information. For example, in order to solve a group of differential equations, or to build a real time control system, or to obtain an output image with multiple gray levels (color levels), a CNN with linear continuous observable outputs is

Manuscript received December 15, 1995; revised June 15, 1997. This work was supported in part by the Office of Naval Research under Grant N00014-91-0516.

L. Wang was with Texas A&M University, Department of Electrical Engineering, College Station, TX 7784 USA. He is now with Crystal Semiconductor, Inc., Austin, TX 78744 USA.

J. Pineda de Gyvez and E. Sánchez-Sinencio are with Texas A&M University, Department of Electrical Engineering, College Station, TX 77843 USA.

Publisher Item Identifier S 1063-8210(98)01318-3.

required. Since the feedback outputs are limited by thresholds, e.g., -1 and 1, state variables have relatively wider dynamic ranges than the feedback outputs and therefore can be used as continuous outputs. In some CNN theoretical research papers [25], the state variable (or state output) has already been mentioned as a useful continuous information of the CNN. Some authors also define state variables as the roots of the differential equations and hence solve differential equations with the CNN.

So far, except for the above fundamental work, there are no circuits that have been fabricated or designed for the purpose of obtaining continuous state outputs. Although in some CNN chips the state variables could be observed [13], [18], they were not used as operating outputs. In this case, additional design constraints are imposed to the CNN cell. These constraints involved improved linearity and the dynamic range of the linear components of the cell.

The contribution hereby proposed consists of a practical solution to handle large image sizes by using 1) a multiplexing scheme and 2) a linearized CNN cell array for color (or gray) images. It should be noticed that the CNN cell array can be as small as 3×3 and still be capable of handling large image sizes. It is necessary to stress out that the state-of-the-art work in cellular neural networks has concentrated on VLSI implementations without really addressing the "systems level." While efficient implementations have been reported, no reports have been presented on the use of these implementations for processing large complex images. The work hereby presented introduces a strategy to process large images using small CNN arrays. The approach, time-multiplexing, is prompted by the need to simulate hardware models and test hardware implementations of CNN. For practical size applications, due to hardware limitations, it is impossible to have a one-on-one mapping between the CNN hardware processors and all the pixels in the image involved. We present a practical solution by processing the input image block by block, with the number of pixels in a block being the same as the number of CNN processors in the hardware.

II. Time-Multiplexing Hardware Simulation

In time-multiplexing hardware simulations one can define a block of pixels (subimage) which will be processed by an equal number of CNN cells [22]. Once convergence is achieved, a new subimage adjacent to the one just processed, is scheduled for further processing. This procedure is repeated until the whole image has been scanned using a lexicographical order, say, from left to right and from top to bottom. It is obvious that with this approach the processing of large images becomes feasible in spite of the finite number of CNN cells. Even though the approach seems simple and appealing, an important observation is necessary: the processed border pixels in each subimage may have incorrect values since they are processed without neighboring information. Hence, to cope with the previous problem, two sufficient conditions must be considered to ensure that each border cell properly interacts with its neighbors. These conditions are: 1) to have a belt of pixels from the original image around the subimage being processed and 2) to have pixel overlaps between adjacent subimages. We will go into the details of these two constraints in the next subsection.

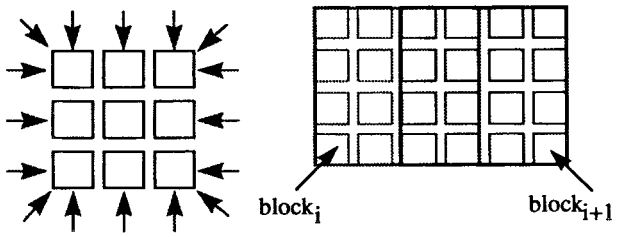


Fig. 1. Conditions for time multiplexing operation: (a) belt of inputs and (b) overlapped pixels.

A. Sufficient and Necessary Conditions for Time Multiplexing

Notice first that in the absence of template-A values the processing error is both image and template-B dependent. In other words, the steady state of a border cell may converge to an incorrect value due to the absence of it's neighbors weighted input. One can easily conclude that the error is canceled if the missing external inputs are provided to the border cells as depicted in Fig. 1(a). Since typically, the array is "embedded" in the image during operation, this condition can easily be satisfied.

Let us address now the interactions among cells. The problem in this situation is more involved because the output signals depend on the state of their corresponding cells. To minimize the error an overlap of pixels between two adjacent blocks is proposed, see Fig. 1(b). In this form, the inner cells of the CNN array will always receive weighted processing information from the border cells.

The general time-multiplexing procedure consists in processing each image block until all CNN cells within the block converge. The block with converged cells will have state output variables which are the values used for the final output image. Every time that a new subimage is processed, the physical CNN array is initialized to the initial conditions of the original image, or to black or to white as required by the template in use. In the overlapping procedure the outer overlapped cell's converged values are discarded since they were computed with incomplete neighboring information. Only the inner cell's converged values are kept as valid values. This implies that for a neighborhood radius of one, an overlap of two pixel column/rows is needed to be able to ensure correct values for pixels assigned to the border cells. With the added overlapping feature, better neighboring interactions are achieved, but at the same time, an increase in computation time is inevitable.

With the previous multiplexing scheme the image needs to be iterated several times over newly obtained states to allow the proper propagation of global effects. Multiple iterations are necessary to guarantee that all cells have converged to correct values taking into account all global effects. This can be inferred by considering a diagonal propagation of, say, a black pixel in a fully white image. Notice that without

overlaps it is impossible to propagate global effects, and that the propagation is achieved with at least one overlapped pixel.

For the purpose of better understanding the overall idea of this time-multiplexing approach, a simplified algorithm is presented below. Assume an $M \times N$ image, an $m \times n$ CNN array, pixel values E_{ij}, o overlaps, and a cell $C(p, q)$.

```
for (i = 1; i < M; i = i + m − o) {
  for (j = 1; j < N; j = j + n − o) {
    for (p = 1; p < m; p = p + 1) {
      for (q = 1; q < n; q = q + 1) {
        u(p, q) = E_{i,j}   /*load input image*/
        /*  load initial conditions,
        and let CNN run */
```

$$x_{p,q}(t_o) = \left\{ \begin{matrix} u_{ij} \\ black \\ white \end{matrix} \right\}$$

$$x_{p,q}(t_{n+1}) = x_{p,q}(t_n) + \int_{t_n}^{t_{n+1}} \cdot f[x_{p,q}(nt)]\, dt$$

```
      }
    }
  }
}
```

III. System Architecture

The CNN IC consists of a 3×3 array with shared input/output pins. Salient features of this implementation are full template programmability, a programmable integration time constant, and an external output at the state node.

Fig. 2 presents a modular view of the CNN IC along with I/O signals.

- $b_{11}, b_{12}, \cdots, b_{33}$ are the pins to set the analog template values of B_{ij} where $i, j = 1, 2, 3$.
- $a_{11}, a_{12}, \cdots, a_{33}$ are the pins to set the analog template values of A_{ij} where $i, j = 1, 2, 3$.
- IO1, IO2, $\cdots$, IO9 are the input–output pins of all nine cells. The pin of each cell is used to do the functions of setting the boundary conditions, initializing the state, and of providing external input values to the cell, as well as obtaining its state output.
- $d1$ and $d2$ are control signals to multiplex each input–output pin for different functions at different time periods.
- V_{bias} is the offset bias voltage for the templates, and V_c is a tuning voltage of the active resistor.
- 5, −5, 1, −1 V are the power supplies for the circuit and for the activation function, respectively.

Two control signals are used as switching signals to multiplex inputs, outputs, and cell initial conditions in order to let them use the same pins. Data lines are shared by analog inputs, boundary values, initial conditions of cell state variables, and outputs of state variables. The logic codes and sequences of pin multiplexing are shown in Table I. The pin multiplexing scheme uses capacitors (0.6 pF) to hold the input information when the circuit is switched to the output

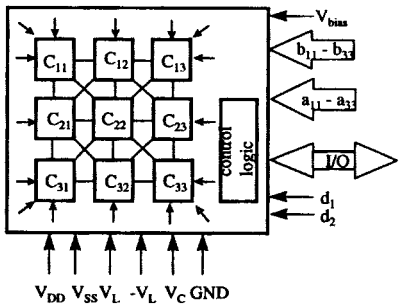


Fig. 2. System structure of the 3×3 CNN.

TABLE I
The Codes and Sequence of Pin Multiplexing Operations

sequence	code	operation
1	11	set boundary values (in)
2	10	set initial conditions (in)
3	01	set input voltages (in)
4	00	extract state output

mode. This capacitance value is designed to eliminate the feedthrough effects of complementary metal–oxide–semiconductor (CMOS) switches, and for the same purpose, all analog switches are transistors with minimal size. As a result, the output is kept unchanged when the pin is switched from the input to the output voltage. The terminal to set the initial condition is connected to the state variable node of the cell.

A. Cell Core

Fig. 3 shows the hardware realization of a CNN cell. Here the integrator time constant is composed of a capacitor C_x, an opamp and an OTA which is in the feedback path. The OTA is used to substitute an active resistor, with a value $g_x = 1/R_x$. The purpose of adding the opamp is to buffer the RC integrator from the 19 multipliers used to implement the weights of both A and B templates. Observe that when the 19 multipliers are connected in parallel a much smaller net output resistance than that of just one multiplier (divided by 19), and also a much larger net parasitic capacitance than that of one multiplier ($19\times$) appear. These two nonideal elements could reduce the effective value of R_x and increase the value of C_x in the structure of Fig. 3 because of their parallel connections with each other. However, the *virtual ground point* (noninverting input) of the opamp can isolate the output impedances of the multipliers from R_x and C_x. On the other hand, the virtual ground makes each multiplier have a "virtual" zero load, and thus eliminates the load effect on the multiplier which comes from the finite output impedance of the transconductance multiplier.

Another advantage of buffering the large aggregated parasitic capacitance of 19 multipliers is that the value of C_x can be controlled by a single capacitor, rather than by many (undetermined) parasitic values. In this way, it is possible to control the time constant of the integrator. More importantly, the mismatches of the time constants of the cells are much

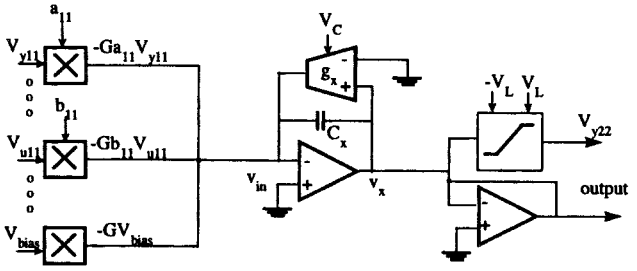


Fig. 3. Building structure of the CNN cell.

smaller if the value of C_x is determined by one monolithic capacitor rather than by many distributed parasitic values.

To prevent the integrator from entering into oscillation, the value of C_x has to be large enough to form a dominant pole, and thus, compensate for the phase shift of the integrator. Assuming a single-pole model for the OTA, this condition is satisfied when $C_x/g_x \gg p$ where g_x is the OTA's transconductance and p is the OTA's dominant pole.

Finally, the buffer in Fig. 3 is basically an opamp with unity-gain feedback connection whose function is to isolate the state variable node of the cell from the outside environment.

B. Multiplier Circuit

For the design of a continuous output CNN, the general requirements for the transconductance multiplier are the linearity and its tolerance to process mismatches.

The multiplier used here is a folded Gilbert multiplier [26] with linearity enhancement, which is illustrated in Fig. 4. PMOS transistors M1 and M2 form a current mirror pair to supply a biasing current to the input pair M3 and M4, thus $I_5 = I_6 = I_b/2$. M3 and M4 are biased in their linear regions as source degenerated resistors, whose functions are to expand the linearity of the input pair M5 and M6. V_H is a control voltage that represents the template value, e.g., a_{11}, b_{11}, etc. In the following equations, M3 and M4 are ignored for convenience of the mathematical analysis. It can be proven that the output differential current of this multiplier is

$$\begin{aligned} I_{o1} - I_{o2} &\approx kV'_{\text{in}}\left(\sqrt{2I_9/k} - \sqrt{2I_{10}/k}\right) \\ &= kV'_{\text{in}}\sqrt{2k/k'}\,(V_{gs5} - V_{gs6}) \\ &= \sqrt{2kk'}\,V_{\text{in}}V_c \end{aligned} \tag{3}$$

where $k = k_5 = k_6 = (W/2L)K_p$ and $k' = k_{11} = k_{12} = k_{13} = k_{14}$.

Equation (3) is the fundamental equation to perform four-quadrant multiplications with the input voltage V_{in} and the control voltage V_c. The linearity of the circuit can be improved by either using long channel transistors or large biasing currents, since the value of $\sqrt{2I_{10}/k'}$ can be much larger than V_{in}.

In order to save area and power dissipation, a summing current mirror (shown in Fig. 5) is utilized to collect the output currents (I_{o1} and I_{o2}) of all 19 multipliers instead of adding individual current mirrors for every multiplier at their corresponding output stage. This cascode summing current mirror is needed to minimize output offsets and make the

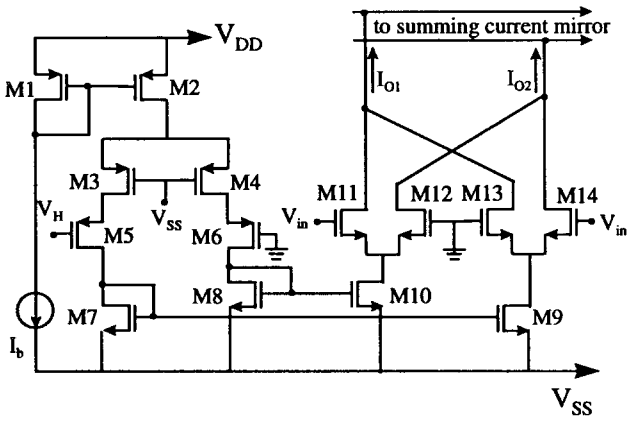


Fig. 4. Circuit of the folded Gilbert multiplier with linear expansion. V_c is the control voltage (template input), and V_{in} is the image pixel input.

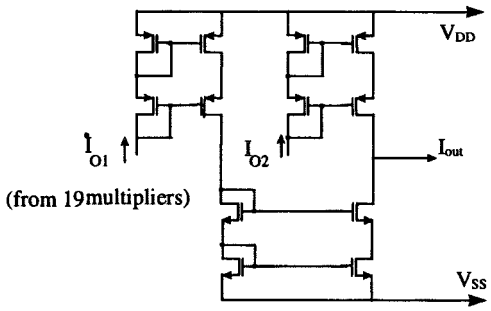


Fig. 5. Summing current mirror of all 19 multipliers.

circuit properties less sensitive to process variations. HSPICE simulations showed a total harmonic distortion (THD) of 0.5%, at $V_{\text{in}} = 1.0$ V and $V_c = 1.0$ V, and a power dissipation of about 0.75 mW using the process parameters of 2 μm n-well CMOS technology from MOSIS.

C. The Linear Tunable OTA

The linear resistor of the cell core was implemented with a tunable OTA. The tunability of the OTA is also indispensable because 1) on-chip tuning is required to compensate systematic errors raised by parameter mismatches and 2) CNN programmability involves that the value of R_x be also adjustable. The traditional method of tuning the transconductance of the OTA is by adjusting the biasing current of the input differential pair. The linearity is significantly reduced by decreasing its biasing current because of the quadratic relationship between the gate-source voltage and the drain current. Hence, this method is not suitable for our purposes. One suitable way to simultaneously improve both the linearity and tunability of the OTA is to utilize a programmable current mirror [20], [21]. This approach has been found to be very good for both linearity and tunability. It consists of adjusting the gain of the current mirrors rather than the biasing current of the input pair. In our case, a modified programmable current mirror (programmable Widlar current mirror) is presented. It has a simple structure and its performance is also good.

1) Circuit Analysis: The circuit structure of the OTA is shown in Fig. 6. The only difference of this circuit from

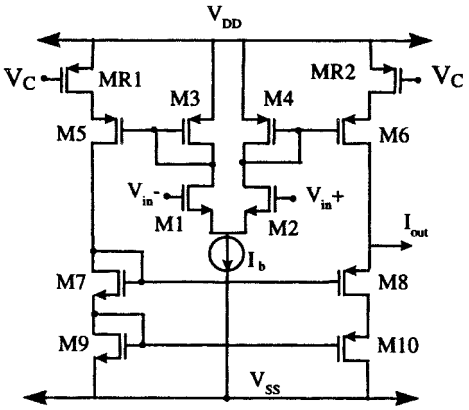


Fig. 6. Circuit of the tunable linear OTA using programmable current mirrors.

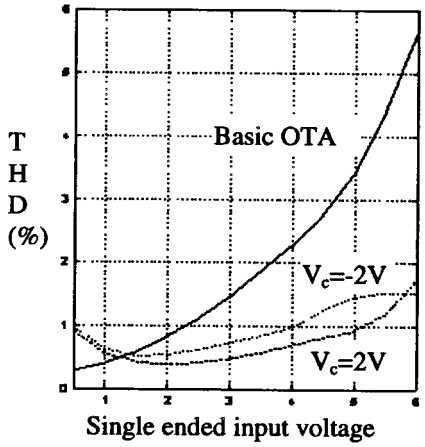


Fig. 7. Plots of THD versus input voltage (peak to peak value) of the basic OTA and the tunable OTA at $V_c = -2$ V and $V_c = 2$ V.

a *basic* CMOS OTA is that there are two transistors MR1 and MR2, which are biased in their linear regions. The functions of both active resistors are to tune the current gains of two current mirrors: M3-M5 and M4-M6, as well as to increase the linear range. Let the linear resistance of MR1 and MR2 be denoted as R. Now recall that since MR1 and MR2 are working in the linear range, their equivalent resistance is $R = 1/[k_R(V_{gs} - V_{TP})]$. Assume also, that $k_R = K_P(W/2L)_{\rm MR}k = k_1 = k_2, k' = k_3 = k_5$ and $V_d = V_{in+} - V_{in-}$.

It is reasonable to assume the values of k and k' to be larger than 10^{-5} A/V^2, R larger than 100 K Ω, and the biasing current I_b large enough (e.g., 35 μA). Then within a limited input range such that I_3 and I_4 are not far from $I_b/2$, we have

$$4R\sqrt{k'I_3} \gg 1 \tag{4}$$

$$4R\sqrt{k'I_4} \gg 1. \tag{5}$$

Therefore, by obtaining I_5 and I_6 in terms of I_3 and I_4, respectively, around MR1 and MR2 the output current can be approximated as

$$I_{\rm out} = I_5 - I_6 = \frac{1}{2k'R^2}\left\{2R\sqrt{kk'}\,V_d\left[2R^{0.5}(k'I_3)^{1/4} - 2R^{0.5}(k'I_4)^{1/4}\right]\right\}. \tag{6}$$

In order to separate the linear and nonlinear terms of $I_{\rm out}$, it is better to expand it into polynomial expressions in terms of the differential input voltage V_d. All even terms vanish if it is assumed that the input voltage is differential. Disregarding high order terms we have

$$I_{\rm out} = \alpha_1 V_d + \alpha_3 V_d^3 + \cdots$$

where

$$\alpha_1 = \left.\frac{dI_{\rm out}}{dV_d}\right|_{V_d=0} \tag{7}$$

$$\alpha_3 = \left.\frac{1}{3}\frac{d^3I_{\rm out}}{dV_d^3}\right|_{V_d=0}. \tag{8}$$

Evaluating the corresponding derivatives, (7) and (8) yield, respectively

$$\alpha_1 = \frac{1}{R}\sqrt{\frac{k}{k}}\left[1 - \frac{1}{2}k'^{-5/4}R^{-1/2}\left(\frac{I_b}{2}\right)^{-1/4}\right] \tag{9}$$

$$\alpha_3 \approx \frac{1}{12}R^{-3/2}k'^{-3/4}k^{3/2}\left(\frac{I_b}{2}\right)^{-13/4}. \tag{10}$$

The linear term in (9) is the conductance of the OTA. In certain cases, such as when R is large enough, α_1 can be further simplified as

$$\alpha_1 \approx \frac{1}{R}\sqrt{\frac{k'}{k}}. \tag{11}$$

Simultaneously notice from (10), that the nonlinearity (third-order distortion) of $I_{\rm out}$ can be greatly reduced by increasing the values of R and I_b. Under the assumptions made in (4) and (5), the third-order term is much smaller than the linear term. Then, it can be concluded that

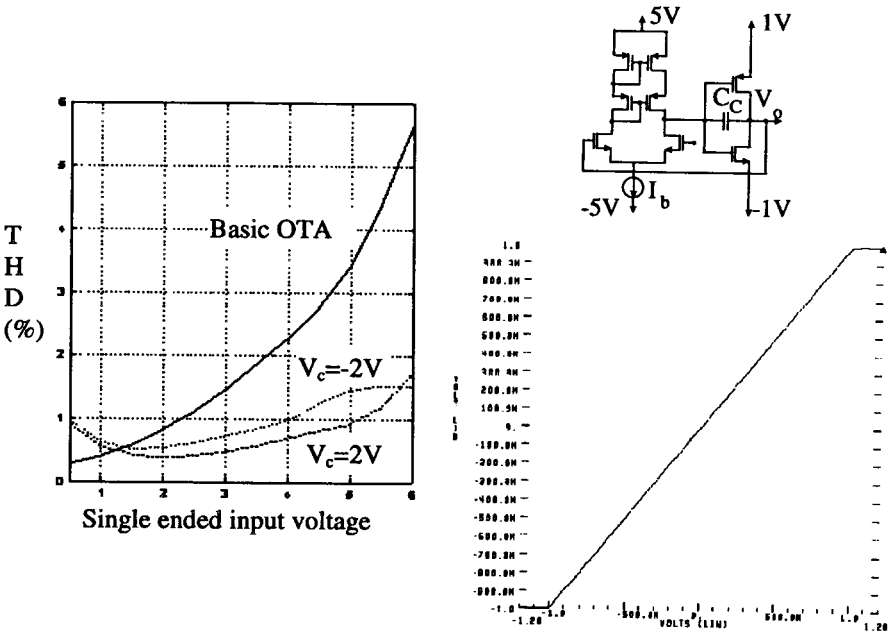
$$I_{\rm out} \approx k_R(|V_{dd} - V_c| - |V_{thP}|)V_d\sqrt{\frac{k}{k'}}. \tag{12}$$

Another advantage of this programmable current mirror is that its gain bandwidth product will not significantly change with the adjustment of its conductance. The OTA's input range is from -3–3 V. The tuning range of the conductance is from 2 to 5 μmhos. The simulated THD versus input voltages is plotted in Fig. 7. Although there are differences between different input voltages, these variations are within the tolerable range of linearity. The estimated total power dissipation of the OTA is 0.45 mW.

D. The Activation Function

The sigmoid activation function plays a very important role to control the errors and the stability of the CNN [19]. The most important aspects to consider are as follows.

1) The threshold voltages of the sigmoid function must be accurate to avoid measurement errors applied to the output voltage.



(a) (b)

Fig. 8. Circuit structure of the activation function: (a) an opamp having separate power voltages and unity feedback configuration and (b) HSPICE simulation results.

2) The slope of the linear segment should be the same among all cells in the CNN. Any mismatch may introduce stability problems.
3) The slope of the sigmoid function of each cell should be about 1.0 to make the cell more stable than other values.
4) The slew rate of the output voltage should be high to have a short settling time.

The circuit structure of the sigmoid activation function is shown in Fig. 8(a). It is basically an opamp in unity gain-feedback connection, but the power supplies of the first and second stages of the opamp are different. The supply voltages of the first stage are -5 and $+5$ V, while the voltages of the second stage are only -1 and $+1$ V. All power voltages are supplied external to the IC. The advantage of using independent power supplies in one opamp is that the threshold voltages of the sigmoid function are well defined and programmable. Therefore we do not have to use hard limiter circuits, whose threshold voltages are always significantly variable with process variations.

The structure of the feedback connection of a high-gain opamp guarantees that the slope of the input–output characteristic curve is almost ideal 1.0 and that there are sharp turning corners at the points of $V_{\text{in}} = -1.0$ V and $V_{\text{in}} = 1.0$ V. These conditions make the active function be a perfect piecewise linear function; see Fig. 8(b). However, the deep feedback connection of the opamp may introduce stability problems. A compensation capacitor C_c has to be added to compensate for the phase shift, but C_c will contribute to the time delay of the activation function. Within one chip, it is acceptable to assume that the relative mismatch (or the variation of the ratio) of the C_c's between two cells is very small, so the time delay mismatch caused by C_c is not critical. The simulated slew rate of the activation function is about 10 V/μs, for a load capacitance of 1 pF. The total power dissipation of the activation function is 0.38 mW.

TABLE II
HSPICE SIMULATED RESULTS OF THE ONE-STAGE OPAMP IN FIG. 9

Parameter	Result
DC gain	65dB
Dynamic swing range	-3V to 3V
Phase margin	50°
Gain bandwidth product	4MHz (C_L = 1pF)
Slew rate	10V/μs
Power dissipation	0.42mW

E. Op Amp

There are two op amps in each cell; one is in the cell core; another one is used as a buffer to isolate the parasitic capacitance of the outside world from the state variable node. Both op amps have a one-stage opamp structure as shown in Fig. 9. The advantage of using a one-stage opamp is that it is more stable than a two-stage opamp, and the load capacitance does not affect the stability of the opamp. The gain of the one-stage opamp is not very high, but is enough for our applications. For $V_{DD} = |-V_{SS}| = 5$ V, its HSPICE simulation results are listed in Table II.

IV. TESTING OF ELECTRICAL PARAMETERS

The 3×3 CNN chip was fabricated with MOSIS n-well 2.0 μm process. The microphotograph of the die is shown in Fig. 10 where all cells are arranged as a 3×3 array. The die area of the circuit is approximately 3.2 mm^2, and the power dissipation is less than 250 mW. The cell area is 0.19 mm^2.

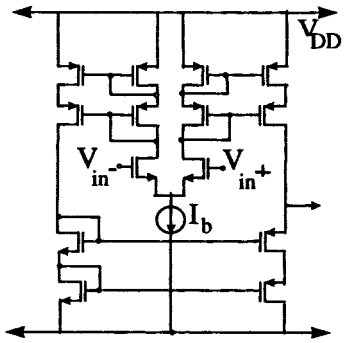


Fig. 9. Circuit structure of one-stage opamp.

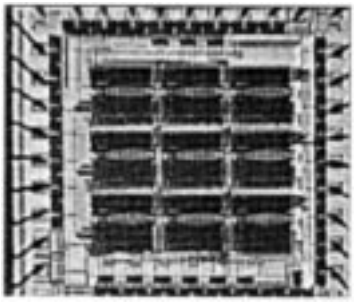


Fig. 10. Photograph of microchip.

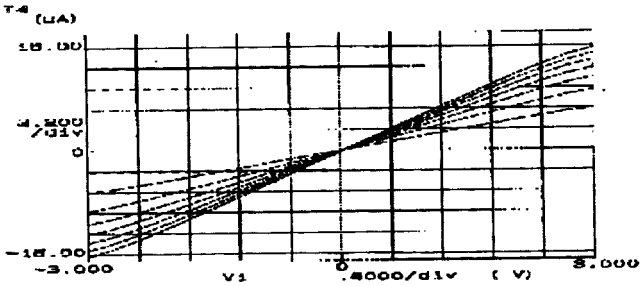


Fig. 11. DC sweep characteristic of the tunable OTA where the tuning voltage V_c is from -2 to $+2$ V.

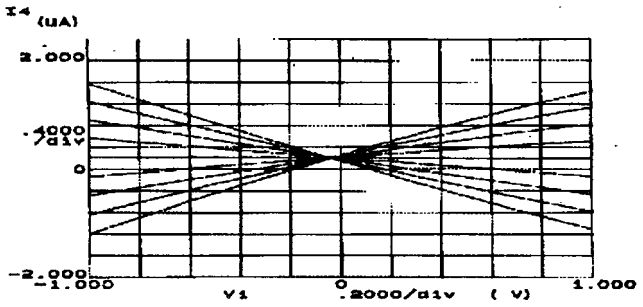


Fig. 12. DC sweep characteristics of the analog multiplier where the sweep voltage is from -1 to 1 V.

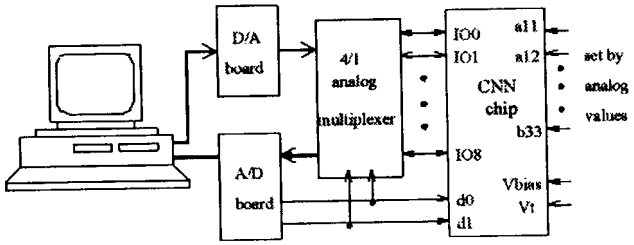


Fig. 13. General interface circuit of the CNN chip with a personal computer.

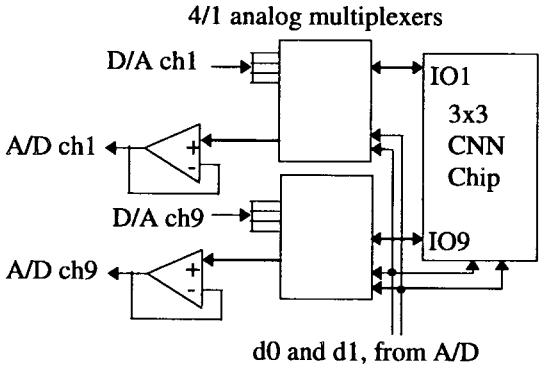


Fig. 14. Detailed connections of 4-1 analog multiplexers for CNN-PC interface.

Two important circuit building blocks; the analog multiplier and the tunable linear OTA, were also fabricated in separate chips for testing purposes. The dc sweep curves of the OTA at different values of V_c are shown in Fig. 11. The dc sweep characteristics of the multiplier are shown in Fig. 12.

Another import function of tuning g_m is to expand the adjustable range of template values. For example, reducing g_m can increase the value of $B(i, j)(= G/g_m \times b_{ij})$.

The linearity of the central cell $C(2, 2)$ can be calibrated by adjusting the g_m of the tunable OTA using the procedure previously described. It is necessary here to deduct an amount of -0.51 V from V_{bias} since this value is used to cancel the output offset of $C(2, 2)$ and cannot be counted in the calculation of linearity.

The CNN chip is connected to a personal computer (PC) through an A/D and a D/A interfacing board. The system connections are described in Fig. 13. The operations of setting inputs and getting outputs from the CNN chip are multiplexed externally by 4-1 analog multiplexer chips (ADG509A). External operations are synchronized with the multiplexing operations inside the chip. The type of A/D board was AT-MIO-16F, which has 12 A/D channels; the type of D/A board was AT-AO-6/10, which has 10 D/A channels. Both are products of National Instruments. The pin multiplexing control code, is generated by a computer program and interfaced through the digital I/O port in the AT-MIO-16F board.

The detailed connections of the analog multiplexers are shown in Fig. 14 where the opamps are added as A/D output buffers to isolate the output node from the parasitic capacitance of the wires and the A/D board.

The operating sequence is listed next as follows.

1) Initialize the A/D and D/A boards. Set the required template values by providing the corresponding analog voltages for the template values.
2) Set the boundary values of the 3×3 CNN, and the initial values of all the CNN cells.

(a) (b) (c)

Fig. 15. Processed images. (a) Unprocessed image, (b) edge detection, and (c) hole filler.

3) Map the pixel values (0–255) of the input image into CNN input voltage values 1.0 to −1.0 V, and send them to the chip.
4) Extract the steady-state values (3.0 to −3.0 V) of the state variables of all cells and map them to pixel values (0–255) of the output image.
5) For time multiplexing applications [22], move to another position in the input image and repeat at Step 2).

V. Image Processing Applications

The following comprises several examples of image processing applications using this 3×3 CNN chip, with output pixels at the state outputs (see Fig. 15). The size (number of pixels) of the input and output image are 256×256. The actual picture is in color, but is displayed here with 255 gray levels. The image processing is realized by using the *time multiplexing* method. The CNN chip only processes a 3×3 pixel array of the image, but the border cells of the CNN are overlapped between the two neighbor arrays to have correct boundary dynamics. Therefore, only cell, $C(2, 2)$ gives the output pixel value. The edge detection templates are the following [2]:

B

0	−0.48	0
−0.48	2	−0.48
0	−0.48	0

A

0	0	0
0	2	0
0	0	0

$$V_{\text{bias}} = -0.15 \text{ V}$$

In our experiments, the value V_{bias} affects the results very much. In order to obtain a good edge, several adjustments of V_{bias} may be needed. The processed image is shown in Fig. 15(b). The hole filling function can be used in contrasting operations and noise removal. This function is realized by the mutual feedback of output pixel values. The corresponding output image is shown in Fig. 15(c). The corresponding templates are as follows:

B

0	0	0
0	2	0
0	0	0

A

0	1	0
1	2	1
0	1	0

$$V_{\text{bias}} = -0.85 \text{ V}$$

VI. Conclusion

This paper demonstrated the feasibility of processing large images using a time-multiplexing approach involving small CNN arrays. For practical image size applications, due to current state-of-the-art technological limitations, it is impossible to have a one-on-one mapping between the CNN hardware cells and all the pixels in the image involved. It is thus a key issue the proper use of time-multiplexing implementations in common-day situations. Additionally, it was shown that a state-node output approach is especially suitable for color image processing and applications involving continuous-time output signals.

References

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

[2] ———, "Cellular neural networks: Applications," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 1273–1289, Oct. 1988.

[3] L. O. Chua and T. Roska, "The CNN paradigm," *IEEE Trans. Circuits Syst.*, vol. 40, pp. 147–155, Mar. 1993.

[4] T. Roska and L. O. Chua, "The CNN universal machine: An analogic array computer," *IEEE Trans. Circuits Syst.*, vol. 40, pp. 163–172, Mar. 1993.

[5] J. A Nossek and G. Seiler, "Cellular neural networks: Theory and circuit design," *Int. J. Circ. Theory Appl.*, vol. 20, pp. 533–553, Sept. 1992.

[6] G. F. Betta, S. Graffi, Z. M. Kovács, and G. Masetti, "CMOS implementation of an analogically programmable cellular neural network," *IEEE Trans. Circuits Syst., Pt. II*, vol. 40, pp. 206–215, Mar. 1993.

[7] P. Kinget and M. Steyaert, "A programmable analog cellular neural network CMOS chip for high speed image processing," *IEEE J. Solid-State Circuits*, vol. 30, pp. 235–243, Mar. 1995.

[8] T. Kacprzak and K. Slot, "Multiple-input OTA based circuit for cellular neural network implementation in VLSI CMOS technology," in *Proc. IEEE CNNA'92*, 1992, pp. 157–162.

[9] A. Rodríguez-Vázquez, R. Dominguez-Castro, and J. L. Huertas, "Accurate design of analog CNN in CMOS digital technologies," in *Proc. IEEE CNNA'90*, 1990, pp. 273–280.

[10] M. Anguita, F J. Pelayo, A. Prieto, and J. Ortega, "Analog CMOS implementation of a discrete time CNN with programmable cloning templates," *IEEE Trans. Circuits Syst., Pt. II*, vol. 40, pp. 215–218, Mar. 1993.

[11] H. Harrer and J. Nossek, "Discrete-time cellular neural networks," *Int. J. Circuit Theory Applications*, vol. 20, pp. 453–467, Sept. 1992.

[12] H. Harrer, J. A. Nossek, and R. Stezl, "An analog implementtion of discrete-time cellular neural networks," *IEEE Trans. Neural Networks*, vol. 3, pp. 466–476, May 1992.

[13] J. E. Varrientos, E. Sánchez-Sinencio, and J. Ramírez-Angulo, "A current-mode cellular neural network implementation," *IEEE Trans. Circuits Syst., Pt. II*, vol. 40, pp. 147–155, Mar. 1993.

[14] S. Espejo, A. Rodríguez-Vázquez, R. Domínguez-Castro, and J. L. Huertas, "Switched-current techniques for image processing cellular neural networks in MOS VLSI," in *Proc. IEEE Int. Symp. Circuits Syst.*, 1992, pp. 1537–1540.

[15] J. L. Huertas, A. Rodríguez-Vázquez, and S. Espejo, "Analog VLSI implementation of cellular neural networks," in *Proc. CNNA'92*, 1992, pp. 141–150.

[16] B. J. Sheu, S. H. Bang, and W.-C. Fang, "Optimal solutions of selected cellular neural network applications by the hardware annealing method," in *Proc. IEEE CNNA'94*, 1994, pp. 279–284.
[17] _____, "Analog VLSI design of cellular neural networks with annealing ability," in *Proc. IEEE CNNA'94*, 1994, pp. 387–391.
[18] J. M. Cruz and L. O. Chua, "Design of high-speed, high-density CNNS in CMOS technology," *Int. J. Circ. Theory Appl.*, vol. 20, pp. 555–572, Sept. 1992.
[19] K. Halonen and J. Vaananen, "The nonidealities of the IC-realization and the stability of CNN-networks," in *Proc. IEEE CNNA'90*, 1990, pp. 226–234.
[20] W. J. Adams and J. Ramírez-Angulo, "Extended transconductance adjustment/linearization technique," *Electron. Lett.*, vol. 27, pp. 842–844, 1991.
[21] J. Ramírez-Angulo and I. Grau, "Wide g_m adjustment range, highly linear OTA with linear programmable current mirrors," in *Proc. IEEE Int. Symp. Circuits Syst.*, 1992, vol. 3, pp. 1372–1375.
[22] C.-C. Lee and J. Pineda de Gyvez, "Time-multiplexing CNN simulator," in *Int. Symp. Circuits Syst.*, London, May 1994, pp. 407–410.
[23] A. Rodríguez-Vázquez, R. Domínguez-Castro, and S. Espejo, "Design of CNN universal chips: Trends and obstacles," in *Proc. CNNA'94*, 1994, pp. 59–60.
[24] J. M. Cruz and L. O. Chua, "A fast, complex and efficient test implementation of the CNN universal machine," in *Proc. IEEE CNNA'94*, 1994, pp. 61–66.
[25] C.-C. Lee and J. Pineda de Gyvez, "Color image processing in a cellular neural network environment," *IEEE Trans. Neural Networks*, vol. 7, pp. 1086–1098, Sept. 1996.
[26] B. Gilbert, "A precise four quadrant multiplier with subnanosecond response," *IEEE J. Solid-State Circuits*, vol. SC-3, pp. 365–373, Dec. 1968.

Lei Wang (M'97) received the B.S. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 1985, and the M.S. degree in electrical/biomedical engineering from Tsinghua University/Peking Union Medical College, Beijing, China, in 1988. He is also working towards the Ph.D. degree at the Department of Electrical Engineering, Texas A&M University, College Station, TX.

He is currently working as a mixed-signal integrated circuit design engineer for Crystal Semiconductor, Inc., Austin, TX. His research interests are in analog circuit design, including analog neural networks, and high-resolution A/D and D/A converters.

José Pineda de Gyvez (M'95) received the degree in electronic systems engineering from ITESM, Mexico, in 1982. He received the M.Sc. degree from INAOE, Mexico, and the Ph.D. degree from the Eindhoven University of Technology, The Netherlands, 1984 and 1991, respectively.

He is the author of *Integrated Circuit Defect Sensitivity: Theory and Computational Models* (New York: Kluwer Academic).

Dr. Pineda was an Associate Editor for Cellular Neural Networks in the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS PART I (1995–1997) and was also an Associate Editor for Technology in the IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING.

Edgar Sánchez-Sinencio (F'92) received the communications and electronics engineering degree from DPN, Mexico, in 1966, the M.S.E.E. degree from Stanford University, Stanford, CA, in 1970, and the Ph.D. degree from the University of Illinois, Urbana-Champaign, in 1973.

He is currently a Professor and Co-Director of the VLSI Telecommunications Research Centre at Texas A&M University. His current research interests are in the fields of RF communication circuits and mixed-signal subsystems. He has published nearly 200 scientific papers, one pioneer book on "switched-capacitor circuits" and has edited three books.

Dr. Sánchez-Sinencio has received the IEEE Guillemin-Cauer Award and the IEEE Darlington Award for his contributions on current-mode low voltage techniques. For his work in integrated filters he became, in 1992, an IEEE Fellow. He is currently the Editor-in-Chief (1997–1999) of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS PART II. For his overall research contributions, he also received the Texas Senate Proclamation #373 for Outstanding Accomplishments by Senator J. Turner on March 25, 1996.