Jpn. J. Appl. Phys. Vol. 39 (2000) pp. 2278–2286
Part 1, No. 4B, April 2000

Cellular νMOS Circuits Performing Edge Detection with Difference-of-Gaussian Filters

Tatsuhiko SUNAYAMA, Masayuki IKEBE, Tetsuya ASAI* and Yoshihito AMEMIYA
Department of Electrical Engineering, Hokkaido University, Kita 13, Nishi 8, Sapporo, 060-8628, Japan

(Received October 22, 1999; accepted for publication December 21, 1999)

Aiming at the development of high-speed image processors, we propose a cellular νMOS circuit that performs the processing of edge detection. The proposed circuit uses neuron MOS (νMOS) transistors for analog convolution operations with Gaussian-shaped kernel functions, which makes the circuit organization extremely simple as compared with that of conventional convolution circuits. Performances of the proposed circuit are evaluated by simulation program with integrated circuit emphasis (SPICE). The results show the usefulness of the cellular νMOS circuit in image-processing applications.

KEYWORDS: cellular automaton, difference-of-Gaussian filter, edge detection, νMOS transistor

1. Introduction

One of the promising research areas in microelectronics is the development of intelligent image sensors based on parallel processing architectures. Among various image-processing architectures, the cellular automaton (CA) is expected to provide high-speed image-processing systems because of its inherent parallel operations.[1)] Recently, a number of CA algorithms for extracting various image features have been proposed and successfully demonstrated,[2)] which show the great potential of CA in image-processing applications.

Aiming at the development of CA-based high-speed processors, a number of large-scale integrated circuits (LSIs) have been proposed and fabricated.[3–8)] The important requirement in developing such CA LSIs is that the LSI must be implemented on one chip in a fully parallel construction. Because image-processing applications need a large number of pixels, the unit processor (pixel circuit) has to be constructed as compactly as possible. To meet this requirement, a functional metal-oxide semiconductor (MOS) device, known as the νMOS transistor, was recently developed and has been applied to various image-processing applications.[9,10)] And the authors have developed a number of CA circuits using the νMOS transistors (cellular νMOS circuits) for morphological processing on binary images and demonstrated that functions required for the processing can be successfully achieved with compact pixel circuits.[11–14)]

In this paper, we propose a CA algorithm for performing difference-of-Gaussian (DoG) filtering and develop the νMOS circuit that performs edge detection on gray scale (analog) images on the basis of the DoG filtering. The cellular νMOS circuit is useful for analog image processing as well as for binary image processing. The proposed circuit uses the νMOS transistor for analog convolution operations with Gaussian-shaped kernel functions, which makes the circuit organization extremely simple as compared with that of conventional convolution circuits.

This paper is organized as follows. In §2, we outline the CA and the DoG filters for edge detection. In §3, we propose a method of edge detection by the CA. Section 4 shows the cellular νMOS circuit for edge detection. Section 5 shows the behaviors of the νMOS circuit, using a simulation program with integrated circuit emphasis (SPICE). Section 6 is devoted to a summary.

*E-mail address: asai@sapiens-ei.eng.hokudai.ac.jp

2. The Cellular Automaton and DoG Filters

2.1 Pattern transformation using the cellular automaton

A cellular automaton (CA) is a discrete dynamical system whose behavior is completely specified in terms of finite local interactions.[1,2)] The CA consists of many identical cells (processors) and local connections among the cells. These cells are regularly arrayed on a two-dimensional rectangular grid. Figure 1 shows a CA consisting of m-by-n cells with local interactions among the neighboring cells. Nine cells represented by the gray boxes are the neighboring cells of cell $C_{i,j}$, including the cell. Each cell has a binary, multiple-valued, or continuous state. All the cells change their states synchronously in an update cycle. During the cycle, the current frame, represented by all the cell-state planes, is replaced by a new frame according to a given transition rule.

The CA can be used for various image processing applications by regarding each cell and its state as a pixel and the pixel-luminance value, respectively. As an example, edge detection by means of the CA is demonstrated in Fig. 2. A binary pixel state (0 or 1) is assumed in the example. Figures 2(a) and 2(b) show an original frame and an edged frame processed by the CA, respectively. The original frame is updated to the edged frame according to the transition rule given in

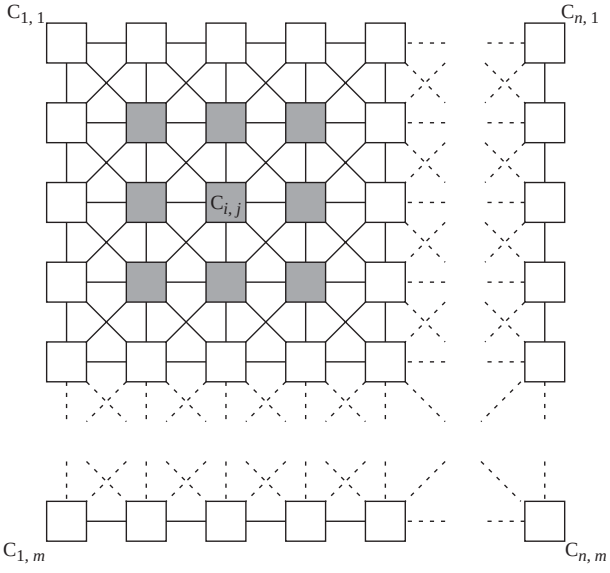


Fig. 1. A cellular automaton (CA) consisting of m-by-n cells with local connections among neighboring cells. Nine cells represented by the filled box are the neighboring cells of the cell $C_{i,j}$, including the cell.

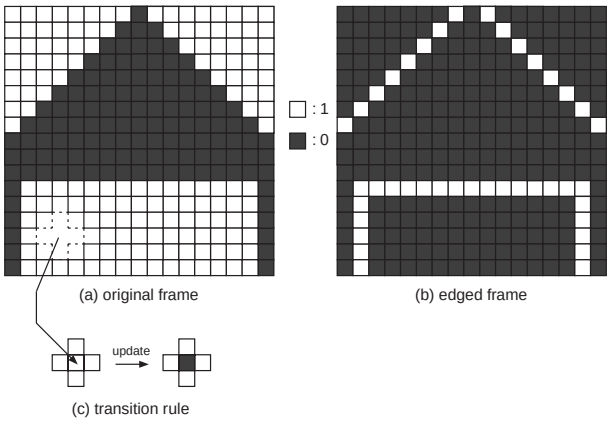


Fig. 2. Edge detection by means of the CA. (a) an original (input) frame; (b) an edged (processed) frame; (c) a transition rule for edge detection. Black and white boxes represent the cell states 0 and 1, respectively.

Fig. 2(c). When the current state of the pixel and that of the nearest four neighbors of the pixel are 1 (white pixel), the state of the pixel is changed to zero (black pixel). Otherwise, the pixel state is stationary through the update. The CA with these rules can extract the edges from the original frame in a single update.

2.2 The DoG filter

The DoG filter plays an important role in various image-processing applications.[15] Using the DoG filters, the edges can be extracted from an input image without being affected by noise, which is closely related to the mechanisms of early vision. The DoG function is obtained by taking the difference of two Gaussian functions with different spatial constants [σ_1 and σ_2 ($> \sigma_1$)] and therefore with different amplitudes [A_1 and A_2 ($< A_1$)]. Figure 3 shows two Gaussian kernels [$A_1\ g(x, \sigma_1)$ and $A_2\ g(x, \sigma_2)$] and the DoG kernel [DoG(x, A_1, A_2, σ_1, σ_2)] for an impulse input $\delta(x)$. The impulse is convoluted with the Gaussian kernels, and the resultant DoG response is obtained as the well known "Mexican hat" function.

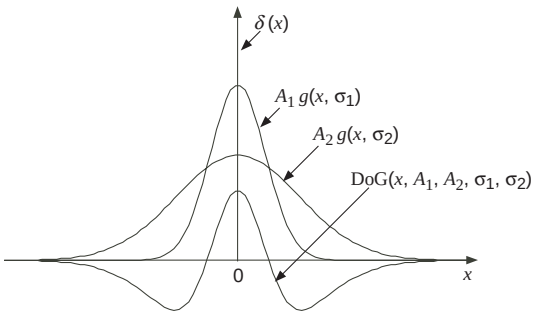


Fig. 3. Gaussian and DoG kernels for an impulse input $\delta(x)$.

When luminance values of pixels in an input image are given to the DoG filter, the edged image is obtained by convoluting input images over all pixels with the DoG function; that is, the edged images are produced by taking the difference between two Gaussian-smoothed images of the input image.

The spatial frequency of the edged image is determined by two spatial constants (σ_1 and σ_2). Noises in the input image can be eliminated by selecting an appropriate value of σ_1, while σ_2 determines the spatial resolving power in detecting the positions of edges with respect to the luminance values of the edges. Fine edges can be extracted by small values of σ_1 and σ_2, but the luminance values of the edges become small and the resultant edged image becomes noisy. In contrast, when large σ_1 and σ_2 are used, the luminance values of the edges become large and noises will be attenuated, but the edged image becomes blurry (difficult to determine the positions of the edges).

3. Edge Detection with DoG Filters on the CA

DoG filtering requires convolution operations over a wide area, while the CA is based on local interaction, as described in the preceding section. To detect edges using the DoG filters on the CA, the authors here present the concept that the Gaussian smoothing can be approximately achieved on a CA with a four-neighbor transition rule given by

$$S_{i,j}(t+\Delta t) = \frac{S_{i-1,j}(t) + S_{i+1,j}(t) + S_{i,j-1}(t) + S_{i,j+1}(t) + 4S_{i,j}(t)}{8}, \quad (i = 1, \ldots, n \quad \text{and} \quad j = 1, \ldots, m) \tag{3.1}$$

where t represents time, Δt the update time step, $S_{i,j}(t)$ the pixel state (luminance value) on a discrete position (i, j), and $S_{i,j}(t + \Delta t)$ the pixel value after the update. Luminance values of four neighboring pixels at time t are represented by $S_{i+1,j}(t)$, $S_{i,j-1}(t)$, $S_{i,j+1}(t)$, and $S_{i,j}(t)$. Figure 4 shows the transition of the pixel values under rule (3.1). At $t = 0$, an impulse input, $S_{i,j}(0) = \delta_{ij}$, was applied to the CA, as illustrated in Fig. 4(a). The subsequent changes of the pixel values are shown in Figs. 4(b) through 4(d), and the pixel-value distributions along with dashed line A-B in Fig. 4(a) are illustrated in Fig. 4(e) with update time steps as a parameter. As shown in Fig. 4(e), the pixel-value distribution gradually spreads as the number of updates increases and, in consequence, the expected Gaussian-shaped smoothing can be achieved.

The reason why we employ eq. (3.1) for the Gaussian smoothing is that the equation can be easily embodied by utilizing the weighted-sum-of-inputs operation of the νMOS transistor. Because all signs of $S_{i,j}$, $S_{i-1,j}$, $S_{i+1,j}$, $S_{i,j-1}$ and $S_{i,j+1}$ in eq. (3.1) are positive, we can easily construct a compact circuit for the convolution operation (weighted summation) using the νMOS transistors without using any subtracting operations.

In the Gaussian smoothing by means of CA, the spatial constant for smoothing is determined by the number of updates and therefore by processing time t (that is discrete in the unit of the update time step). By subtracting pixel values at $t = t_2 (> 0)$ from pixel values at $t = t_1 (< t_2)$, that is, $S_{i,j}(t_1) - S_{i,j}(t_2)$ for all values of i and j, the edges of an input image can be extracted according to the DoG mechanism discussed in §3. Note that time t_1 (corresponding to spatial constant σ_1) has to be chosen so that the image noises can be sufficiently removed, and that time t_2 (corresponding to σ_2) has to be selected with respect to the luminance values and the spatial frequency of edge patterns in the images.

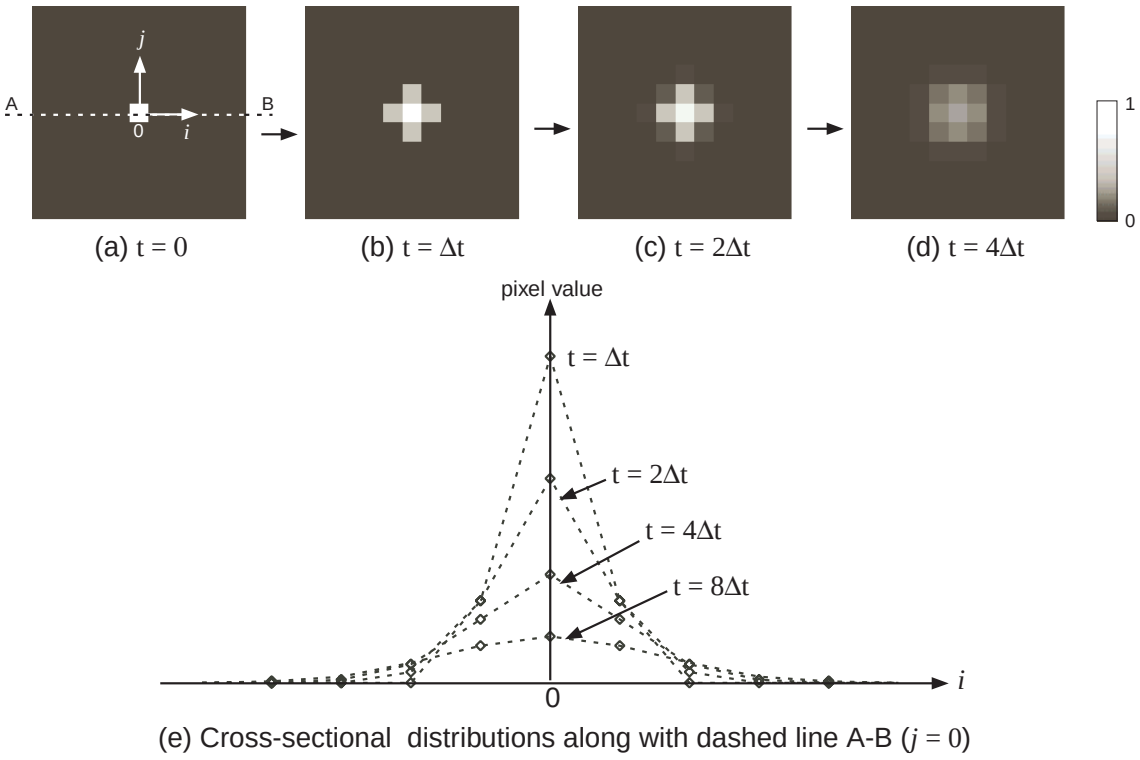


Fig. 4. Transition of the pixel values under rule (3.1). (a) an impulse input at $t = 0$; (b) smoothed pattern at $t = \Delta t$; (c) $t = 2\Delta t$; (d) $t = 4\Delta t$; (e) pixel-value distributions along with dashed line A-B in (a).

4. Circuit Implementation of the CA Systems for the Edge Detection

To present the practical form of our idea, we designed CA LSIs for edge detection. Figure 5(a) shows the floorplan of the LSI that consists of a two-dimensional array of n-by-m pixel circuits and an array of m absolute difference circuits (AbsDC). The array of pixel circuits operates as the CA for Gaussian smoothing, while the AbsDCs take the difference of the Gaussians and output the absolute values of the difference. By the absolute calculation, the edges in an incident image are represented by positive outputs of the AbsDCs, while flat patterns are represented by zero outputs.

The pixel circuit [Fig. 5(b)] consists of a photodiode, a

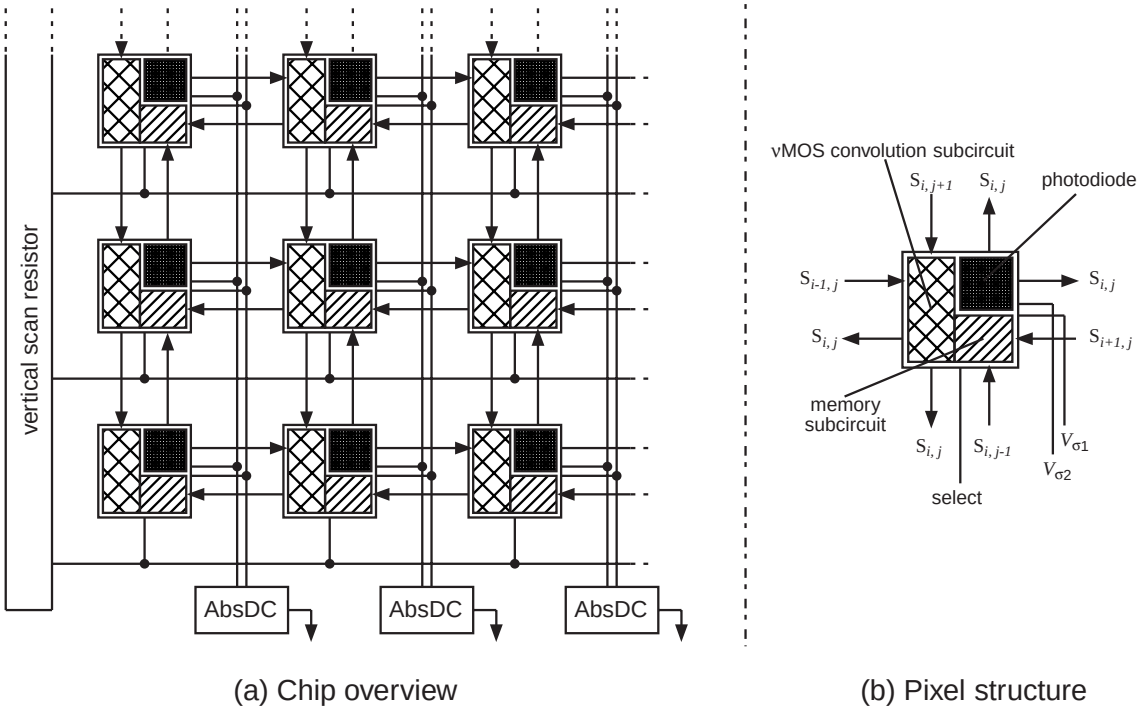


Fig. 5. Overview of the proposed LSI that consists of a two-dimensional array of pixel circuits and an array of absolute difference circuits (AbsDC).

νMOS convolution subcircuit, and a memory subcircuit. It receives local inputs ($S_{i-1,j}$, $S_{i+1,j}$, $S_{i,j-1}$, and $S_{i,j+1}$) from four-neighborhood pixel circuits and sends its pixel value ($S_{i,j}$) to the neighborhood pixel circuits. The array of pixel circuits operates in the following cycles: i) a storing cycle, ii) the first smoothing cycle and iii) the second smoothing cycle. During the storing cycle, the PD in each pixel circuit accepts an incident image and generates the corresponding photocurrent. The charge of the photocurrent is stored in a capacitor. By the end of the cycle ($t = t_0$), the image luminance on each PD has been transduced into a voltage on the capacitor. In the first smoothing cycle, the transduced voltage in each pixel circuit is distributed around the neighboring pixel circuits according to the CA operation. At the end of this cycle ($t = t_1$), all the pixel circuits store the first-distributed voltages ($\equiv V_{\sigma 1}$) in the memory. Note that the smoothing constant σ_1 is determined by the duration of this cycle. Then, in the second smoothing cycle, these voltages are further distributed through additional CA operations. At the end of this cycle ($t = t_2$), the first-distributed voltages and the second-distributed voltages ($\equiv V_{\sigma 2}$) are transferred to the AbsDC array.

When a pixel circuit (i, j) is selected by the vertical scan resistor, the circuit sends the output voltages ($V_{\sigma 1}$ and $V_{\sigma 2}$) to the i-th AbsDC on the same column. Because the vertical scan resistor accesses all the pixel circuits within the same row (j) in sequential order, all the outputs of the pixels on the j-th row are simultaneously transferred to the AbsDCs.

Each AbsDC calculates the differential value between $V_{\sigma 1}$ and $V_{\sigma 2}$, and then outputs the absolute value of the differential value, that is, $|V_{\sigma 1} - V_{\sigma 2}|$. Because the absolute difference value has a large positive value around the edges, the edges are represented by bright lines in the output image.

4.1 A νMOS pixel circuit for the CA

Figure 6 shows a νMOS pixel circuit for the CA performing the Gaussian smoothing. The circuit consists of two subcircuits; i) a convolution subcircuit and ii) a memory subcircuit. The convolution subcircuit consists of a νMOS source follower biased by a transistor M1, a photodiode (PD), a storage capacitor (C_s), a differential amplifier (M2 through M6), a pMOS source follower (M7 and M8), transfer gates (T1 through T3), and a level shift circuit (M9 and M10).

The differential amplifier and the pMOS source follower construct an operational amplifier that amplifies a differential voltage between nodes (a) and (b) with a large negative gain. The output node (c) of the amplifier is connected to the level shift circuit, and the shifted output voltage on node (b) is fed back to the amplifier. Thus, these circuits (M2 through M10) construct a unity gain buffer that transmits the voltage on node (a) to nodes (b) and (c). Voltage V_c on node (c) is shifted up from the voltage V_b on node (b) as

$$V_c = V_b + V_T + \sqrt{\frac{I_{bias}}{\beta}}, \qquad \left(\beta = \frac{1}{2}\mu_n C_{ox}\frac{W}{L}\right) \quad (4.1)$$

where C_{ox}, V_T, I_{bias}, μ_n, W and L represent the gate capacitance of M9, the threshold voltage of M9, the source current produced by M10, the electron mobility, and the channel width and length of M9, respectively, assuming that M9 operates in a saturation region.

Before the storing cycle of the CA LSI, a "reset" signal is applied to transfer gate T3 to reset the voltage on node (a) at V_{bias}. During the storing cycle, clocks ϕ_1 and ϕ_2 are fixed at zero, and the PD accepts an input light and generates the corresponding photocurrent to transduce the luminance value to a voltage on node (a).

During the first and the second smoothing cycles, the νMOS source follower receives the transduced voltages from four-neighborhood pixel circuits ($S_{i-1,j}$, $S_{i+1,j}$, $S_{i,j-1}$, and $S_{i,j+1}$) to perform the convolution operation for Gaussian smoothing with the rule (3.1). In the pixel circuit, the convoluted voltage $V_{i,j}$ is given by

$$V_{i,j} = \frac{S_{i-1,j} + S_{i+1,j} + S_{i,j-1} + S_{i,j+1} + 4S_{i,j}}{8 + C_{ox}/C_0} - V_T - \sqrt{\frac{I_{bias}}{\beta}}, \quad (4.2)$$

where C_{ox}, C_0, V_T, I_{bias}, and β are the floating gate-substrate capacitance, the gate-floating gate capacitance, the threshold voltage of the νMOS transistor, the source current produced by M1, and the transconductance parameter of the νMOS transistor, respectively. Assuming that the transconductance parameters and the threshold voltages of the νMOS transistor and M1 are the same as those of M9 and M10, eq. (4.2) becomes equivalent to eq. (3.1) if $C_{ox} \ll C_0$ because the term $V_T + \sqrt{I_{bias}/\beta}$ in eq. (4.2) is cancelled at node (c) according to eq. (4.1).

It should be noted that we cannot completely eliminate the term $V_T + \sqrt{I_{bias}/\beta}$ in eqs. (4.1) and (4.2) due to the existence of practical device mismatches. These mismatches induce all $V_{i,j}$ to VDD (or 0) as the number of transition steps increases. However, they can be suppressed by adjusting the length of the storing cycle manually. When $V_{i,j} \rightarrow$ VDD (or 0), the length of the storing time will be shortened (or extended).

In both the first and second smoothing cycles, square-clock signals are given to ϕ_1 and ϕ_2. When $\phi_1 = 1$ ($\phi_2 = 0$), the convoluted voltage ($V_{i,j}$) is stored in C_s through T2. This voltage is shifted up and appears on node (c). When $\phi_1 = 0$ ($\phi_2 = 1$), the shifted voltage ($S_{i,j}$) on node (c) is given to four-neighborhood pixel circuits through T5. By repeating these operations, Gaussian smoothing is performed until the desired value of σ_1 is obtained. In the above discussion, the photocurrent is assumed to make only a small contribution to the convolution operation during the cycle; that is, the first and second smoothing cycles must be sufficiently shorter in duration than the storing cycle.

During the first smoothing cycle, ϕ_3 is fixed at 1. At the end of the first cycle ($t = t_1$), ϕ_3 is set at 0. At this time, the smoothed voltage $S_{i,j}(t_1)$ is stored on C_m in the memory subcircuit, through the transfer gate T4 (the voltage is denoted V_s in the figure). Then, in the second smoothing cycle, additional smoothing is performed by applying the same clocks (ϕ_1 and ϕ_2) in the same manner as in the first smoothing cycle. The smoothing operation is continued until the desired value of σ_2 is obtained.

At the end of the second smoothing cycle ($t = t_2$), both ϕ_1 and ϕ_2 are fixed at zero, and the "select" signal is applied to

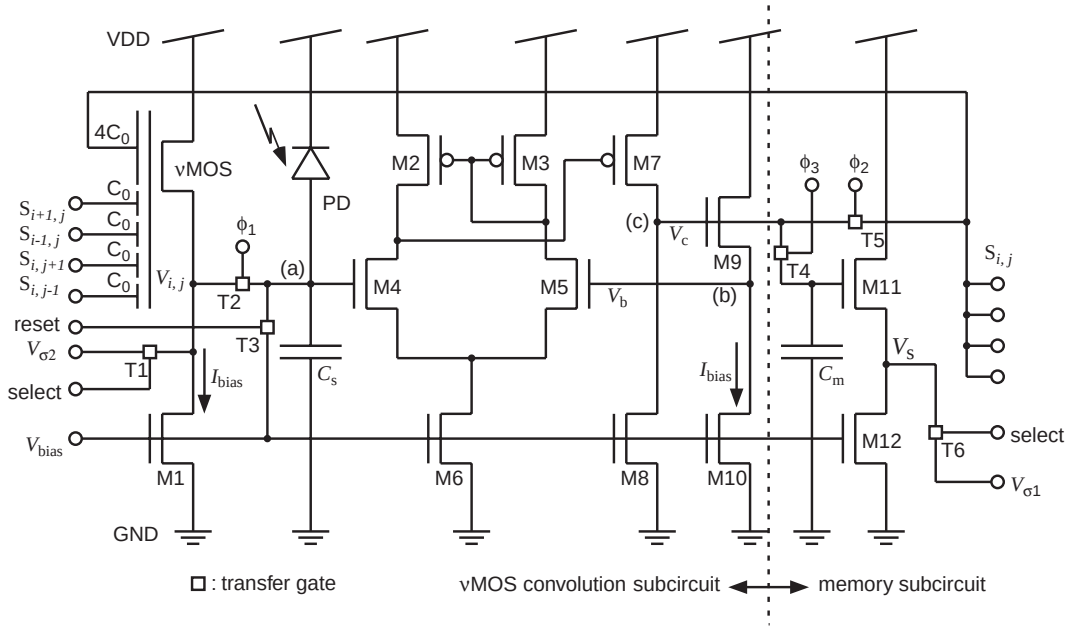


Fig. 6. νMOS pixel circuit for the CA performing the Gaussian smoothing.

transfer gates T1 and T6. Then, the first and second smoothed voltages [$V_{i,j}(t_1)$ and $V_{i,j}(t_2)$] are sent to the j-th AbsDC (the voltages are denoted by $V_{\sigma 1}$ and $V_{\sigma 2}$).

It should be noted that the movement of an input image during the storing cycle causes a motion blur on the accepted images (transduced voltages $V_{i,j}$ in the circuit). However, in both the first and second smoothing cycles, the movement does not affect the CA operation because the input photocurrent is assumed to be very small.

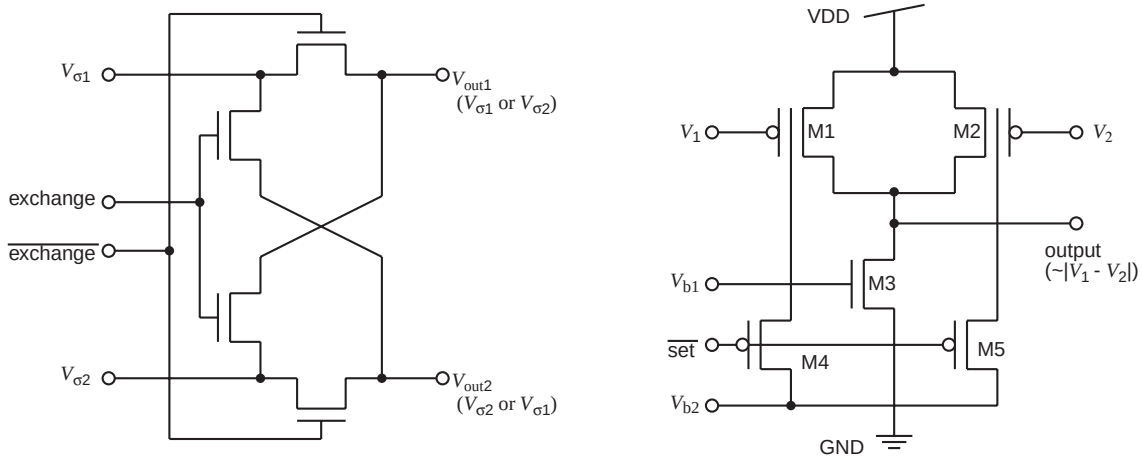
4.2 The absolute difference circuit

The absolute difference circuit (AbsDC) consists of an absolute value circuit[16] and a voltage exchanger. Figures 7(a) and 7(b) show the voltage exchanger and the absolute value circuit. The AbsDC accepts the smoothed voltages ($V_{\sigma 1}$ and $V_{\sigma 2}$) produced by pixel circuits and outputs the absolute difference voltage between $V_{\sigma 1}$ and $V_{\sigma 2}$.

In the AbsDC, the voltage exchanger receives two input voltages ($V_{\sigma 1}$ and $V_{\sigma 2}$) from the pixel circuits . When the "exchange" signal is zero, the exchanger passes the input voltages $V_{\sigma 1}$ and $V_{\sigma 2}$ to the outputs V_{out1} and V_{out2}, respectively. The outputs are exchanged when the "exchange" signal is 1, that is, $V_{\text{out1}} = V_{\sigma 2}$ and $V_{\text{out2}} = V_{\sigma 1}$.

The absolute value circuit has a differential pair consisting of double-gate MOS transistors (M1 and M2) biased by a transistor M3 with bias voltage V_{b1}. The outer gates (V_1 and V_2) of M1 and M2 are connected with the outputs of the voltage exchanger (V_{out1} and V_{out2}), while the inner gates are connected with switching transistors M4 and M5. When M4 and M5 are open ("set" = 1), the inner gate voltages are charged to V_{b2}, while these gates are floating when "set" = 0.

The operation of the AbsDC is divided into three modes: i) a setting mode, ii) a floating mode and iii) an exchanging mode. Each mode is selected by the control signals "exchange" and "set" in the AbsDC. In the setting mode ("exchange" = 0 and "set" = 1), output voltages of the ex-



(a) voltage exchanger (b) absolute value circuit[16]

Fig. 7. Absolute difference circuit (AbsDC). (a) voltage exchanger; (b) absolute value circuit.

changer (V_{out1} and V_{out2}) are applied to the inputs (V_1 and V_2) of the differential pair in the absolute value circuit. Simultaneously, the inner gates of M1 and M2 are charged up to V_{b2} through M4 and M5. In the floating mode ("exchange" = "set" = 0), M4 and M5 are open and the inner gates become floating. At this time, the floating gate voltages of M1 and M2 are charged up to $V_{b2}-V_1$ and $V_{b2}-V_2$, respectively. In the exchanging mode ("exchange" = 1 and "set" = 0), the two inputs (V_1 and V_2) for the differential pair are exchanged by the voltage exchanger. By this exchange, the floating gate voltages of M1 and M2 are charged up to $V_{b2}+V_{dif}$ and $V_{b2}-V_{dif}$, respectively, where $V_{dif} \equiv V_2 - V_1$. When V_{dif} is negative (or positive), M1 and M3 (or M2 and M3) operate as an nMOS source follower that amplifies $V_{b2} - V_{dif}$ (or $V_{b2} + V_{dif}$) with a finite gain. Thus, the absolute value of the differential voltage ($|V_{dif}|$) appears on the output node. Note that the bias voltages V_{b1} and V_{b2} must be set so that the output will be zero if $V_{dif} = 0$.

5. Simulation Results

In this section, we show the results of SPICE simulations for the proposed pixel circuits, AbsDC, and CA LSIs. In the following simulations, we used typical parameter values for all transistors, assuming a 0.6 μm CMOS process, as listed in Table I.

Figures 8(a) and 8(b) show the input-output characteristics of the operational amplifier in the pixel circuit and the simulated circuit, respectively. In the simulation, the supply voltage (VDD) was set at 5 V, and the bias voltage (V_{bias}) was set at 0.65 V so that the bias current $I_{bias} \approx 1$ μA. For the given physical parameters, the linear characteristics were obtained in the input range of 0.3 V $< V_{in} <$ 3.8 V. Because the upper-shifted voltage (V_c) was saturated with the supply voltage at $V_{in} \approx 3.8$ V, the output of the operational amplifier (V_b) was limited at the voltage ($\approx$ 3.8 V). These results indicate that the circuit operates correctly according to eq. (4.2) only if the pixel voltages ($V_{i,j}$) are in the range of 0.3 V $< V_{i,j} <$ 3.8 V.

Figure 9 shows the input-output characteristics of the AbsDC. In the simulation, the bias voltages V_{b1} and V_{b2} were set at 0.8 V and 4.0 V, respectively. The "exchange" and "set" voltages were set at 0 V or 5 V that correspond to the signal "0" and "1", respectively. The linear characteristics was obtained in the input range of 0.2 V $< |V_{in}| <$ 0.8 V. Although a slight offset exists around $|V_{in}| <$ 0.2 V, the offset is useful for eliminating the influence of device mismatches such as in zero-bias currents of the PDs.

We combined 15 $\times$ 15 pixel circuits into a CA LSI and observed the responses of the LSI for a given input pattern [Fig. 10(a)]. In this simulation, we applied an input image to

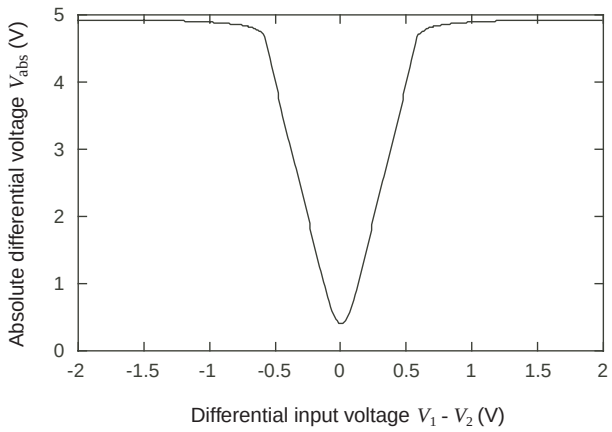


Fig. 9. Input-output characteristic of the absolute difference circuit.

Table I. Circuit constants and physical parameters assumed in the simulations: the ratio of the channel width to the channel length of MOS transistors W/L, gate oxide thickness T_{ox}, Zero-bias threshold voltage of nMOS and pMOS transistors $V_{Tn,p}$, and mobility of electrons (μ_n) and holes (μ_p).

W (μm)	L (μm)	T_{ox} (Å)	V_{Tn}(V)	V_{Tp}(V)	μ_n (cm^2/Vs)	μ_p (cm^2/Vs)
1.2	0.6	110	0.55	−0.6	440	178

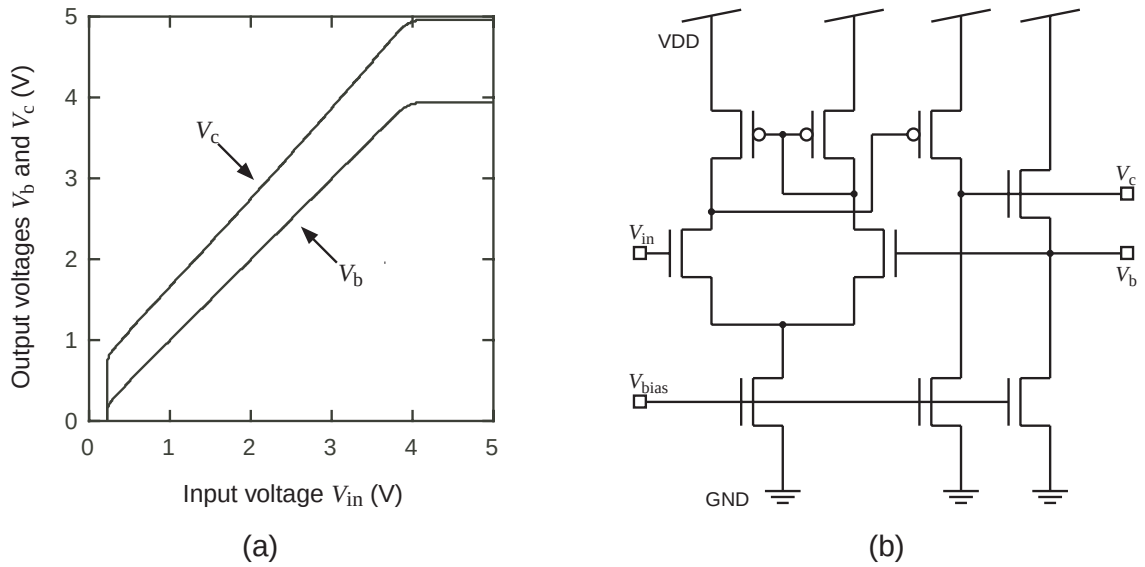
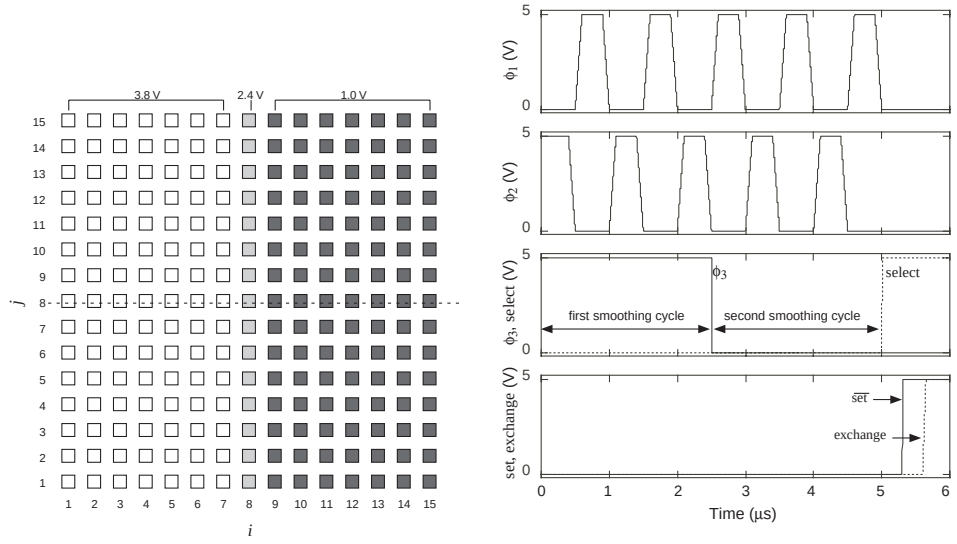
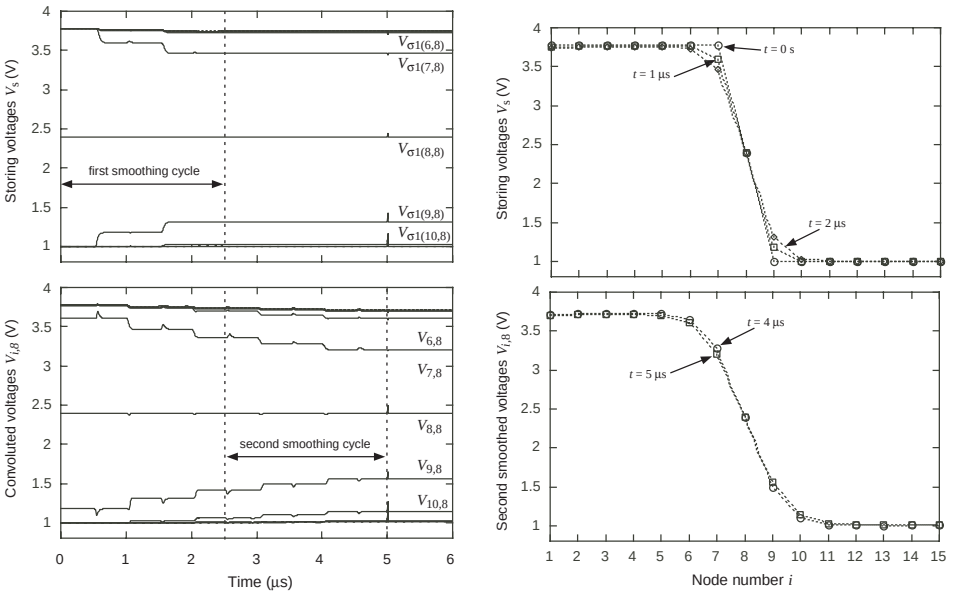


Fig. 8. Input-output characteristic of the operational amplifier in the pixel circuit (a) and the simulated circuit (b).





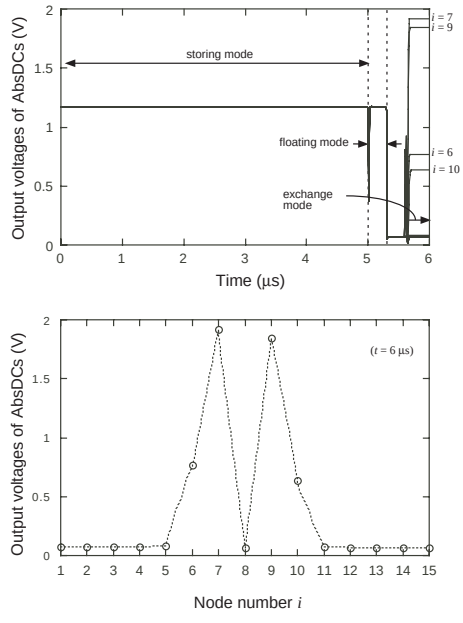


Fig. 10. Simulation results for the CA LSI consisting of 15 × 15 pixel circuits. (a) input pattern; (b) clocks and control signals; (c) transient responses of storing voltages V_s (upper) and convoluted voltages $V_{i,8}$ (lower); (d) output distributions of the pixel circuits during the first smoothing (upper) and the second smoothing cycle (lower); (e) time courses of the output voltages in the AbsDCs (upper) and output distributions of the AbsDCs.

the LSI by specifying the initial value of the voltage on the storage node [node (a) in Fig. 6] for each pixel circuit. The initial voltages were set at

$$V_{i,j}(0) = \begin{cases} 3.8\ \text{V}, & (0 < i < 8) \\ 2.4\ \text{V}, & (i = 8) \\ 1.0\ \text{V}, & (8 < i < 16) \end{cases} \tag{5.1}$$

for all j, as shown in Fig. 10(a). In Fig. 10(a), an edge exists between the nodes of i = 7 and i = 9. Figure 10(b) shows the waveforms of clocks and control signals (ϕ_1, ϕ_2, ϕ_3, "select", "exchange" and "set") for the smoothing operation cycles. The maximum photocurrent and the storing cycle were assumed to be 10 nA and 190 μs, respectively. The storing cycle is not shown in the results because the input image was applied by the node voltages. Floating gate capacitances of the νMOS transistors (C_0), storage capacitance (C_s) and memory capacitance (C_m) were set at 100 fF, 500 fF and 500 fF, respectively. The first smoothing was performed during 0 s $< t <$ 2.5 μs (two updates on the CA), and the second smoothing was performed during 2.5 μs $< t <$ 5.0 μs (two updates). Figure 10(c) shows transient responses of the pixel circuits along the dashed line in Fig. 10(a). During the first smoothing cycle, the pixel values were distributed around the neighborhood pixel circuits with increasing time [Figs. 10(d)]. At the end of the cycle, when $t = 2.5$ μs, ϕ_3 was set at zero and the pixel values were stored in C_m. Then, the pixel values were further smoothed in the second smoothing cycle, while the stored voltages (V_s) remained stationary. At the end of the cycle ($t = 5.0$ μs), the "select" signal was applied to the pixel circuits, and pixels along the dashed line in Fig. 10(a) were selected. Then, the stored voltages (V_s) and current pixel values of the selected pixel circuits ($V_{i,j}$) were transferred to the AbsDCs through transfer gates T1 and T6 (the transferred voltages are denoted $V_{\sigma 1(i,j)}$ and $V_{\sigma 2(i,j)}$). During both the first and second smoothing cycles, the "exchange" signal to the AbsDC was set at zero and the "set" signal was set at 1, that is, all the AbsDCs operate in the setting mode. After the second smoothing cycle ($t = 5.3$ μs), the "set" signal was set at zero (the floating mode). Then, the "exchange" signal was set at 1 at $t = 5.6$ μs (the exchange mode). Figure 10(e) shows the time courses of the output voltages in the AbsDCs and the resultant output distributions at $t = 6.0$ μs. Large positive output voltages were obtained around the edges (i = 6, 7, 9, and 10), while the outputs corresponding to flat planes were approximately 0 V.

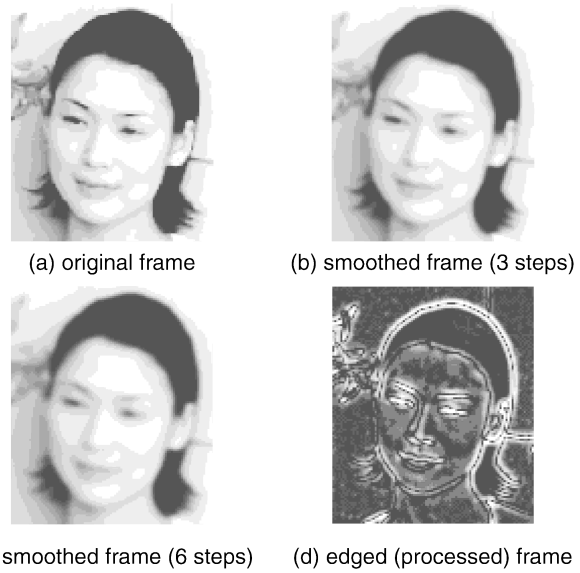


Fig. 11. Edge detection by means of the CA LSI for an analog image. (a) original frame; (b) smoothed frame after three updates; (c) smoothed after with six updates; (d) absolute difference frame between (b) and (c).

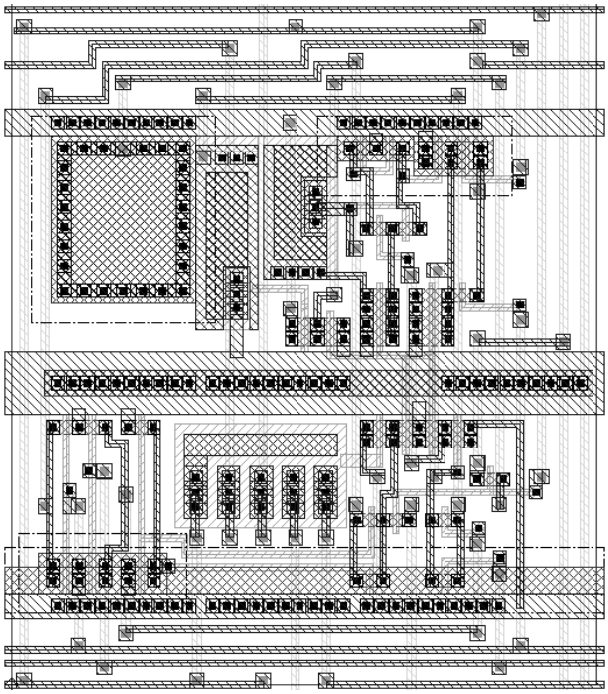


Fig. 12. Pattern layout of the proposed pixel circuit.

By combining 120 $\times$ 100 pixel circuits and 100 AbsDCs, we confirmed the image processing operation of the proposed CA LSI, as shown in Fig. 11. The original frame [Fig. 11(a)] was smoothed by the CA LSI. Figures 11(b) and 11(c) show the smoothed frames after three and six updates, respectively. From the first smoothed (three updates) and the second smoothed (six updates) frames, the CA LSI successfully produced an edged image, as shown in Fig. 11(d).

A prototype CA LSI was designed by assuming a 1.2 μm double-poly double metal CMOS process. Figure 12 shows a layout pattern of the proposed pixel circuits. A single pixel circuit occupied an area of only 107 μm $\times$ 132 μm, including the area of a photodiode (30 μm $\times$ 28 μm). The resultant fill factor was approximately 6%.

6. Summary

Aiming at the development of high-speed image processors, we proposed a cellular νMOS circuit that performs the process of edge detection. The performance of the proposed νMOS circuit was evaluated by extensive SPICE simulations. With a small number of transistors and a compact construction, the circuit successfully extracted edge patterns from incident images.

The important feature of the proposed CA LSI is that the operation speed does not decline even if the number of pixels increases. If a very large-scale integration is attained, the CA LSI significantly improves the performance of the image processing by replacing the conventional digital signal processors.

Because the proposed circuit outputs analog voltages, it can be directly connected to subsequent analog circuits, such as analog motion detectors for velocity sensing,[17] analog feature extraction[18] and analog vector quantitation for pattern matching.[19] This means that additional analog processors can be integrated on one chip, which leads to the development of high-performance analog image processors.

Acknowledgement

This work was supported by the Semiconductor Technology Academic Research Center (STARC).

1) T. Toffoli and N. Margolus: *Cellular Automata Machines* (MIT Press, Cambridge, 1987) p. 37.
2) K. Preston, Jr. and M. J. B. Duff: *Modern Cellular Automata: Theory and Applications* (Plenum Press, New York, 1984) p. 17.
3) E. Macii and M. Poncino: IEEE Trans. Reliab. **46** (1997) 173.
4) S. Nandi, S. Chattopadhyay and P. P. Chandhuri: Proc. 9th Int. Conf. VLSI Design (IEEE Comput. Soc. Press, Los Alamitos, 1995) p. 61.
5) E. D. Adamides, P. Tsalides and A. Thanailakis: Proc. 6th Mediterranean Electrotechnical Conf. (IEEE, New York, 1991) p. 335.
6) J. E. Varrientos, J. R. Angulo and E. S. Sinencio: Proc. 33rd Midwest Symp. Circuits and Systems (IEEE, New York, 1991) p. 277.
7) P. Tsalides, A. P. Marriott, P. J. Hicks and A. Thanailakis: Proc. VLSI and Computer Peripherals (IEEE Comput. Soc. Press, Washington, 1989) p. 3.
8) D. Sawh, J. Loewen, W. Lehn, H. C. Card, M. Pawlak, D. M. Burek and R. D. McLeod: Technical Digest. Proc. Canadian Conf. Very Large Scale Integration (McGill Univ., Montreal, Que., Canada, 1986) p. 133.
9) T. Shibata and T. Ohmi: IEEE Trans Electron Devices **382** (1992) 1444.
10) S. Jung, R. Thewes, T. Scheiter, K. F. Goser and W. Weber: IEEE J. Solid-State Circuits **34** (1999) 978.
11) M. Ikebe, K. Kameishi and Y. Amemiya: Electr. Eng. Jpn. **126** (1999) 41.
12) M. Ikebe, M. Akazawa and Y. Amemiya: Computers & Electr. Eng. **23** (1997) 439.
13) M. Ikebe, M. Akazawa and Y. Amemiya: Proc. 1998 2nd Int. Conf. Knowledge-Based Intelligent Electronic Systems (IEEE, New York, 1998) p. 447.
14) M. Ikebe and Y. Amemiya: Proc. the Int. Symp. Future Intellectual Integrated Electronics (Tohoku Univ., Sendai, 1999) p. 377.
15) D. Marr: *Vision* (W. H. Freeman and Company, New York, 1982) p. 62.
16) T. Shibata: Ext. Abstr. 1998 Int. Conf. Methodologies for the Conception, Design and Application of Soft Computing (World Scientific, Singapore, 1998) p. 29.
17) T. Asai, M. Ohtani and H. Yonezu: Jpn. J. Appl. Phys. **38** (1999) 2256.
18) T. Asai, M. Ohtani and H. Yonezu: IEEE Trans. Neural Networks **10** (1999) 1222.
19) M. Konda, T. Shibata and T. Ohmi: Proc. 6th Int. Conf. Microelectronics for Neural Networks, Evolutionary and Fuzzy Systems (Univ. Technol. Dresden, Dresden, Germany, 1997) p. 174.