Pergamon

Computers Elect. Engng Vol. 23, No. 6, pp. 439–451, 1997
© 1998 Elsevier Science Ltd. All rights reserved
Printed in Great Britain
0045-7906/97 $17.00 + 0.00

PII: S0045-7906(97)00030-X

A FUNCTIONAL νMOS CIRCUIT FOR IMPLEMENTING CELLULAR-AUTOMATON PICTURE-PROCESSING DEVICES

M. IKEBE, M. AKAZAWA and Y. AMEMIYA

Faculty of Engineering, Hokkaido University, Kita 13, Nishi 8, Sapporo 060, Japan

(Received 6 January 1997; accepted 30 March 1997)

Abstract—This paper proposes a design of cell circuits for implementing cellular-automaton devices that perform morphological picture processing. To produce the complex cell functions required for the morphological processing, we present the idea of using the silicon functional device, ν-MOS FET. We designed sample cell circuits for several morphological processings, and simulated their operation to show the expected cell operation. We also designed a sample cellular-automaton circuit using the proposed cell circuits, and demonstrated in simulation its example processing (noise cleaning and edge extraction in an image). A low dissipation of about 20 μW per cell circuit can be expected at 1 MHz operation; therefore, 10^5 or more cells that operate in parallel can be integrated into an LSI. © 1998 Elsevier Science Ltd. All rights reserved

Key words: Cellular automaton, picture processing, ν-MOS FET.

1. INTRODUCTION

The cellular automaton is a parallel processing system that is suitable for high-speed picture processing. To implement the cellular automaton into LSIs, we must first develop a cell circuit that can produce required cell functions in compact construction. This paper proposes such a cell circuit: namely, a νMOS cellular automaton circuit.

The cellular automaton is a parallel, distributed data-processing system that consists of many identical processing elements (cells) regularly arrayed on a plane. Each cell changes its state in discrete time steps through interaction with its neighboring cells. The data that the cellular automaton manipulates is a pattern of the cell states (i.e. a matrix, of which the elements represent states of the arrayed cells). The cellular automaton receives an input pattern and converts the pattern into various differing patterns with time steps; at an opportune moment, the converted pattern is retrieved as an output. With proper interaction rules, we will be able to obtain useful pattern transformations.

The cellular automaton has potential applications especially to binary picture processing, because—if each cell and its state are regarded as a *picture element* and a *black–white level of the picture element*—the operation of the cellular automaton is just the same as *morphological picture processing* on binary (two-tone) pictures. The cellular automaton can be expected to provide high-speed morphological processing devices because its operation is inherently parallel.

A difficulty in developing such cellular-automaton devices is that the device has to be implemented in one chip with *fully parallel construction* (one processing circuit for each cell). Because picture-processing applications require a large number of elements (e.g. 500 × 500 elements for television pictures), it is, therefore, required that a cell circuit be compact in construction and small in area. However, it is difficult to construct a compact cell circuit with existing transistors because many devices are required to implement the required cell functions.

To overcome this problem, we will present the idea that a compact cell circuit for picture processing applications can be constructed by use of a silicon functional MOS device known as the *ν-MOS FET (the neuron MOS FET)*. In the following sections, first, we will review the great similarity between the cellular automaton operation and the morphological picture processing (Section 2). We will then propose that a cell circuit for morphological picture processing can be constructed simply by using the ν-MOS FETs (Section 3). We will present sample cell circuits and simulate their operation to show that the circuits can produce the cell functions required for morphological processing (Section 4). We will also discuss the power dissipation of the ν-MOS FET cell circuits and present a method for designing low-power cell circuits (Section 5).

2. THE CELLULAR AUTOMATON AND ITS APPLICATION TO MORPHOLOGICAL PICTURE PROCESSING

2.1. Pattern transformation using cellular automata

A cellular automaton is a processing system that consists of a large number of simple, identical processing elements (cells) regularly arrayed on a plane (Fig. 1). Each cell has two or more states, and all the cells change their states synchronously in discrete time steps according to a given local interaction rule (a *cell function*). Each state determines its subsequent state based on the current states of its own and of its nearest-neighboring cells. The cellular automaton contains no control center, but is capable of complex state transition [1,2].

The cellular automaton operates as a transducer that produces an output information pattern in response to an input information pattern. As an example (Fig. 2), we assume a *binary* cell state (1 or 0) and consider the *eight* neighbors in determining the subsequent state of each cell, and follow the cell function illustrated in Fig. 2(a) (called the *Game-of-Life* rule). We start with an initial cell-state pattern illustrated in Fig. 2(b) (step 0), and with time steps, we will observe transition of the cell-state pattern as shown in the figure (steps 1–7). Another initial pattern will produce a different pattern change. There are many other cell functions and, therefore, various pattern transformations.

2.2. Morphological picture processing

Morphological picture processing is a type of processing in which the spacial form or structure of objects within an image is modified. The morphological processing is often used as a preliminary to image analysis; it is used to condition and modify raw image signals in a way such that structural features of objects in an image will be enhanced or accentuated [3–5].

The basic concept of the morphological processing for binary images is as follows. A small-sized mask (typically 3 × 3 pixels) is scanned over an image. If the binary pattern of the mask (called a *template*) matches the state of the pixels under the mask, then the center pixel of the 3 × 3 pixel window in the image will be subsequently set to some desired binary state. For a pat-

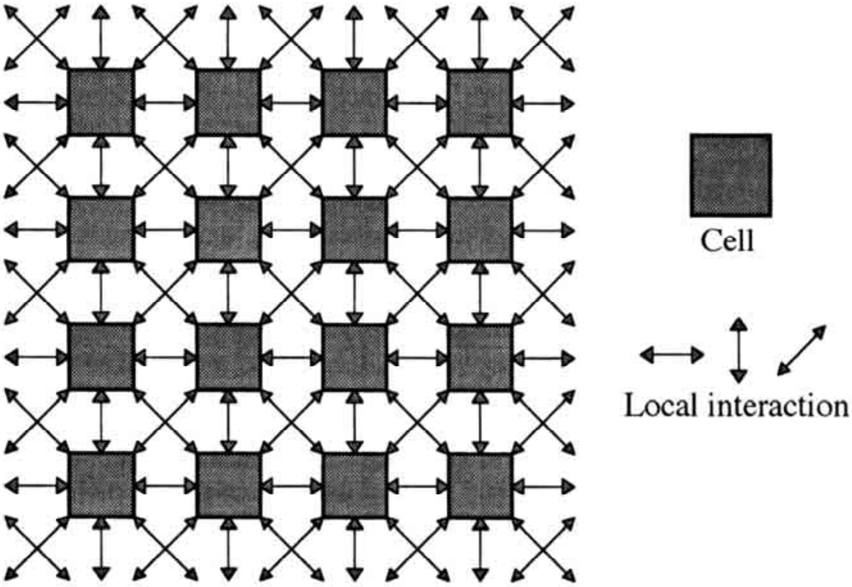
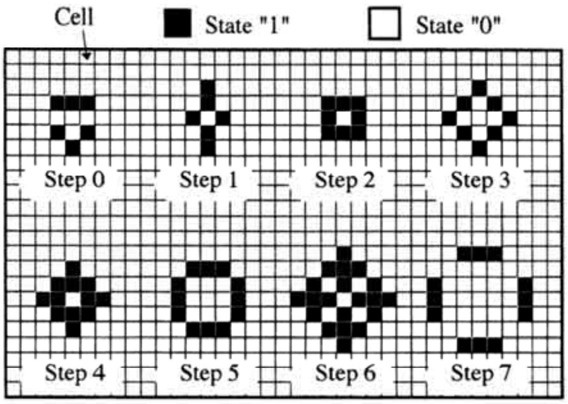


Fig. 1. Cellular automaton. An information processing system consisting of a large number of identical processing cells with local interactions.

Current state	Number of adjacent "1" cells	Subsequent state
1	2 , 3	1
	0, 1, 4, 5, 6, 7, 8	0
0	3	1
	0, 1, 2, 4, 5, 6, 7, 8	0

(a) Cell function



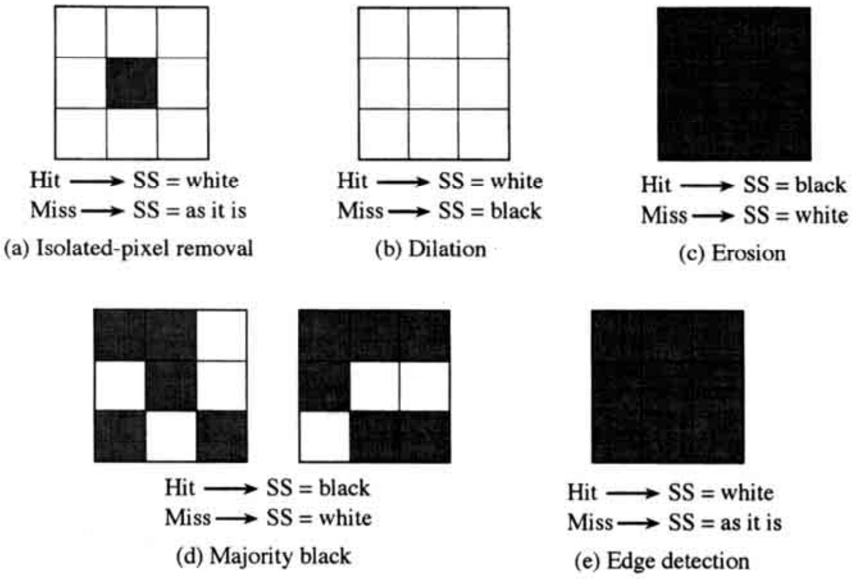
(b) Pattern transformation

Fig. 2. Example of patten transformation in a cellular automaton. The Game-of-Life operation is illustrated: (a) cell function; (b) pattern transformation.

tern mismatch, the center pixel will be set to the opposite state, or will be left as it is. After scanning over the entire image, all the pixels are converted simultaneously to their subsequent states. For example, to perform pinpoint-noise cleaning, if the 3 × 3 pixel pattern is encountered that coincides with the template given in Fig. 3(a) and then the center pixel in the 3 × 3 is changed to white; otherwise, it is left as it is. This operation removes isolated black points in an image.

For the following discussions, we give several known instances of morphological processing.

1. *Dilation*: set the center pixel to white if all the pixels in a 3 × 3 window are white; otherwise, set the center pixel to black. The corresponding template is illustrated in Fig. 3(b). With dilation, an object grows uniformly by a single-pixel-width ring of exterior pixels. This is a basic operation for morphological processing and is frequently used together with erosion, described below.



(a) Isolated-pixel removal

(b) Dilation

(c) Erosion

(d) Majority black

(e) Edge detection

Fig. 3. Example templates for morphological picture processing. "Hit" means the correspondence between the template and the pattern of the 3 × 3 pixel window, and "Miss" means the lack of correspondence. "SS" means the subsequent state of the center pixel.

2. *Erosion*: set the center pixel to black if all the pixels in a 3 × 3 window are black; otherwise, set the center pixel to white. The corresponding template is illustrated in Fig. 3(c). With erosion, an object shrinks by a single-pixel-width ring of interior pixels.
3. *Majority black*: set the center pixel to black if five or more pixels in a 3 × 3 window are black; otherwise, set the center pixel to white. There are 256 qualifying templates for this operation. Two instances of templates are shown in Fig. 3(d). The majority black is useful for filling small holes in objects and closing short gaps in strokes. (The Game-of-Life is somewhat similar to this operation, but is more complex.)
4. *Edge detection*: set the center pixel to white if all the pixels in a 3 × 3 window are black; otherwise, leave the center pixel as it is. The corresponding template is shown in Fig. 3(e). The edge detection converts the interior pixels of an object to white, but leaves the periphery pixels black.

By selecting appropriate operations, we can perform various processing on measurements of images. An example is illustrated in Fig. 4; a noisy image of an object (a letter E) is cleaned by a combination of dilation and erosion; then the edge of the object is extracted by edge detection. For many other applications, see Refs. [3–5].

2.3. Morphological processing devices using cellular automata

If each cell of a cellular automaton is regarded as a pixel and its 1–0 state as a black–white level of the pixel, then the operation of the cellular automaton is the same as the morphological processing on binary pictures; the cell functions for cellular automata corresponds to the templates for morphological processing. The cellular automaton can be expected to provide high-speed morphological processing devices because of its inherently parallel operation.

In the application to actual devices, the cellular-automaton morphological processor has to be integrated with an image sensor into a chip (one cell circuit beside each pixel-sensor element) in order to receive the input image data in parallel. This chip receives an image input, binarizes the image signal, performs the morphological preprocessing on the binary image, and then outputs the processed data in time series for the subsequent image-analysis subsystems. (To construct

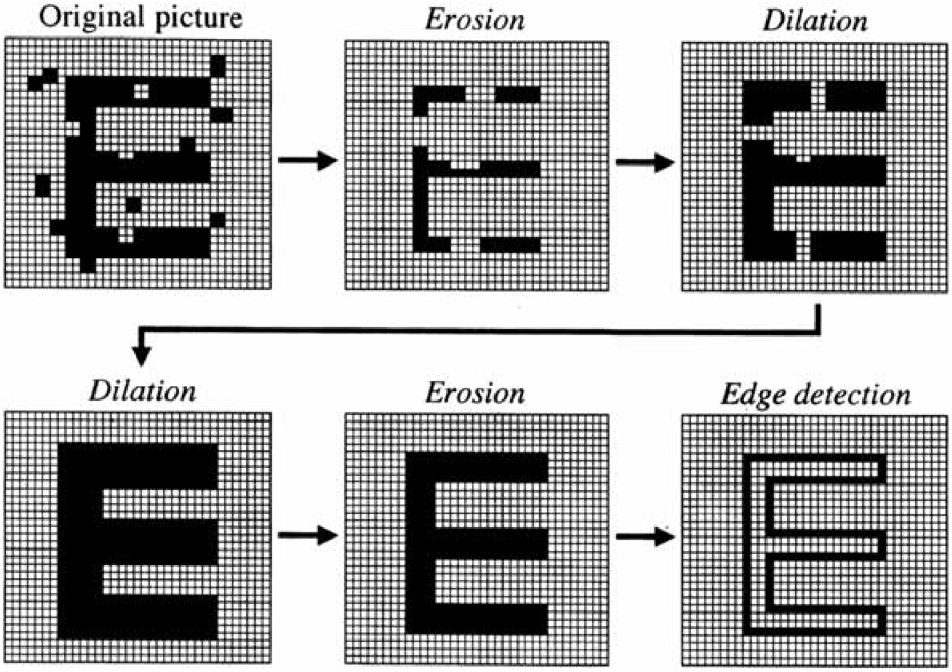


Fig. 4. An instance of morphological processing. The object in the picture is a letter E with noise. The noise is cleaned up by dilation and erosion; then the edge of the object is extracted by edge detection.

such an *intelligent* image-sensing device, a sensor array and an output circuit are required in addition to the cellular-automaton processing circuits, but we will not discuss the first two here.)

To integrate a cellular automaton into a chip, a large number of cell circuits is required (e.g. 500 × 500 elements for ordinary television pictures and millions for high-definition pictures), so the cell circuit must be compact and small-sized in its construction. The essential operation of the cell circuit is to determine its subsequent state as a function of the current states of its own and its neighboring cell circuits. The cell function depends on what processing is required, and it is not always a simple, symmetric function that can be given by a brief Boolean representation ("symmetric" means that each adjacent cell makes an equal contribution toward determining the subsequent state of the center cell). The cell function may be a more complex function, such as a majority decision, a multithreshold, or a weighted-input function. To implement such functions in compact construction, will propose the use of a variable-threshold logic device known as the ν-MOS FET. ["Cell function" is synonymous with "template" (or a set of templates).]

3. IMPLEMENTING CELL CIRCUITS BY USING ν-MOS FETS

3.1. The ν-MOS FET and its transfer characteristic

A νMOS FET is a variant of the floating-gate MOS FETs [Fig. 5(a)]. It consists of a source, a drain, a channel region, a floating gate placed above the channel region, and two or more input gates, each of which is capacitively coupled to the floating gate through input capacitance *Ci*. There are *n*-channel and *p*-channel devices, and they are usually combined in series to form a ν-CMOS inverter [Fig. 5(b)]. For details of the ν-MOS FETs, see Refs. [6–9].

When input voltages are applied to a ν-CMOS inverter, each input *Vi* induces the potential of the floating gate in proportion to C_iV_i. Hence, the total potential of the floating gate represents the weighted sum of inputs (ΣC_iV_i). The output voltage is "0" ($V_{\text{out}} = 0$) if the weighted sum of

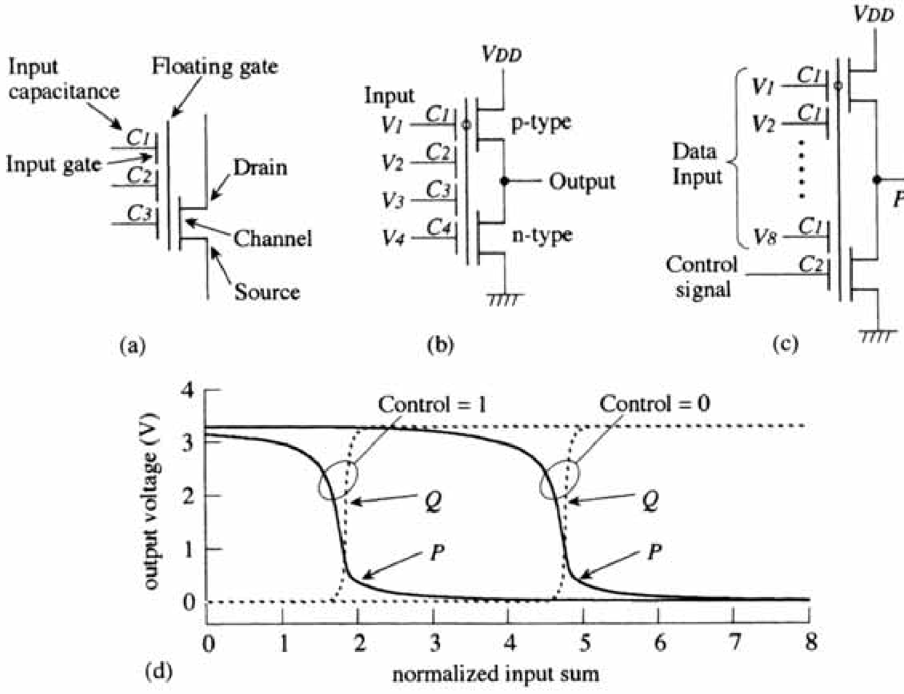


Fig. 5. ν-MOS FET and its transfer characteristics: (a) ν-MOS FET; (b) ν-CMOS inverter; and (c) transfer characteristics (i.e. the output voltage versus normalized input sum ($\Sigma V_i/V_{\text{dd}}$) of a ν-CMOS inverter with eight signal inputs and one control input. For simplicity all input gates are set to the same input voltage. Simulated assuming 0.5 μm CMOS device parameters, $C1$ = 50 fF for each signal input, and $C2$ = 130 fF for the control input.

inputs is greater than the inverter threshold, and the output is "1" ($V_{\text{out}} = V_{\text{dd}}$) if the weighted sum is less than the threshold.

To show the variable-threshold operation of a ν-CMOS inverter, we illustrate in Fig. 5(d) the transfer characteristics (i.e. the output voltage vs. normalized input sum ($\Sigma V_i/V_{\text{dd}}$ curve) calculated for the circuit in Fig. 5(c). (For the device parameters, see the figure caption and Appendix). In this instance, we assumed nine input gates and used eight of them for data input and the other one for threshold control. The threshold characteristic of a ν-CMOS inverter (solid curves) is not very steep, but if necessary, it can be reshaped by adding an inverter as illustrated by dashed curves. A single-threshold function is, thus, easily obtained with one ν-CMOS inverter, and its threshold can be changed by a control input. By combining a number of ν-CMOS inverters we can implement a variable-multithreshold logic operation.

3.2. Constructing templates using ν-CMOS inverters

The basic operation of the morphological processing is to check the window pattern of an image against a given template to see whether they are consistent with each other. This operation can be implemented simply by using a ν-CMOS inverter, as illustrated in Fig. 6. Suppose, for instance, that a template is given that has the pattern [Fig. 6(a)] in which the center pixel 0 is white, pixels 1, 4 and 7 are black, pixels 0, 3, 5 and 8 are white, and pixels 2 and 6 are black or white (don't care). To compare a 3×3 window pattern with this template, an eight-input ν-CMOS inverter is prepared [Fig. 6(b)] and pixel signals from the window pattern are applied to

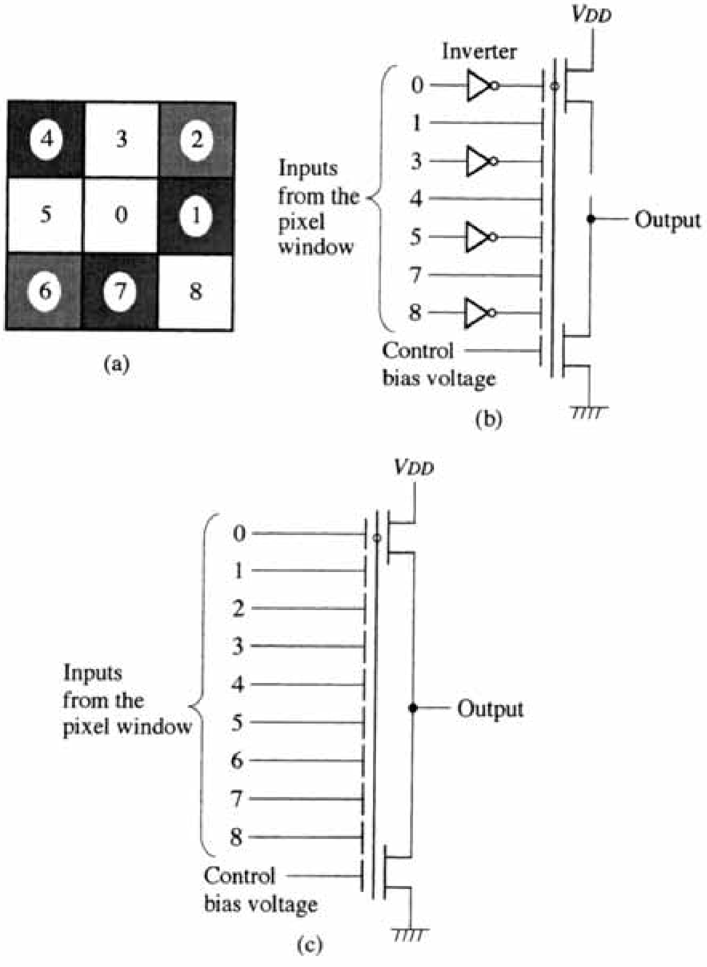


Fig. 6. Implementation of templates by using ν-CMOS circuits: (a) sample template; (b) a ν-CMOS circuit for the sample template, and (c) a ν-CMOS circuit for the majority black template.

the input gates of the inverter (signals from black pixels are "1" and from white are "0"). The function of the template is given by the way in which the signal is connected: signals from pixels 1, 4, and 7 are connected directly to the input gates, signals from 3, 5, and 8 are connected through inverters, and signals from 2 and 6 are *not* connected. The eighth gate is used to control the threshold of the ν-CMOS inverter. In this circuit construction, all the seven inputs for the ν-CMOS inverter will become "1" if and only if the 3 × 3 pixel pattern of the window matches the template pattern. By setting the ν-CMOS inverter at an appropriate threshold, we will be able to know the pattern matching by observing the output change from 1 to 0.

As another example, we present a circuit construction for the majority black templates [Fig. 6(c)]. A ten-input ν-CMOS inverter is used for this purpose; nine pixel signals are applied to the nine inputs, and a bias voltage is applied to the tenth input to control the threshold such that the ν-CMOS inverter will change its output from 1 to 0 if five or more pixel signals are 1. The majority black corresponds to a majority-decision cell function and requires a set of many templates, but can be implemented compactly by using a single ν-CMOS inverter. If necessary, we can give a differing weight to each pixel signal by changing the value of each input capacitance.

3.3. Game-of-Life cell circuits

The Game-of-Life is the best known example of cellular-automaton rules (cell functions), so we take up first the circuit that can implement this function. Required is a double-threshold, symmetric cell function as illustrated in Fig. 7(a); the input to the cell circuit is the number of adjacent cells that are in state "1", and the output is the subsequent state to which the cell will change one time-step later. The threshold of the function has to be changed according to the current state of the cell.

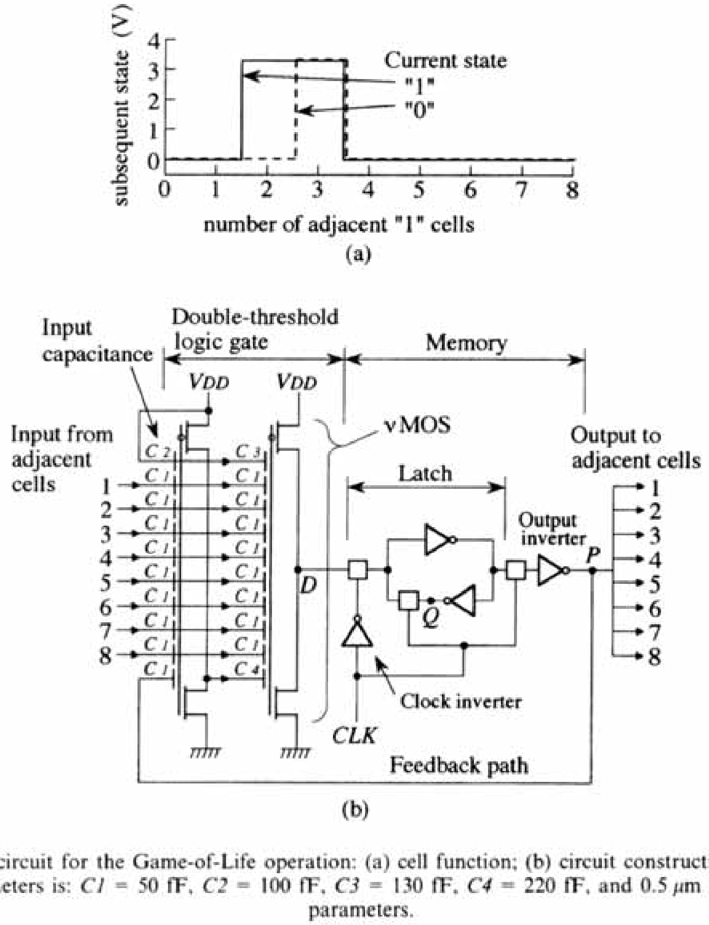


Fig. 7. Cell circuit for the Game-of-Life operation: (a) cell function; (b) circuit construction. A sample set of parameters is: $C1$ = 50 fF, $C2$ = 100 fF, $C3$ = 130 fF, $C4$ = 220 fF, and 0.5 μm CMOS device parameters.

The circuit construction we propose is illustrated in Fig. 7(b). It consists of two subcircuits: a double-threshold logic gate and a cell-state memory. The logic gate receives the current cell-state data (voltages of 1 or 0) from adjacent cells to determine the subsequent state of its own cell. The memory stores the current cell-state data (a voltage of 1 or 0) and sends the data to the adjacent cells. The same data is also sent back to the logic gate through a feedback path in order to control the threshold. The cell state is updated according to a clock signal (*CLK*). For a sample set of the input-capacitance parameters (*C1–C4*), see the caption of Fig. 7.

The double-threshold logic gate consists of two ν-CMOS inverters combined in cascade and produces the Game-of-Life cell function [see Fig. 2(a)]. Each ν-CMOS inverter has eight data inputs (coupling capacitance *C1*) to receive the cell-state data of eight neighbors and also has a bias input (coupling capacitance *C2* or *C3*) to set the threshold to an optimum value. The first ν-CMOS inverter has a feedback input (coupling capacitance *C1*) to change its threshold according to the current cell state, and the successive ν-CMOS inverter receives through *C4* the output of the first inverter to produce the double-threshold characteristic. The memory consists of a D-type latch (two inverters and two CMOS switches) and an output buffer (one inverter and a switch) that are driven by the clock *CLK*. The circuit operation is as follows:

1. The logic gate produces the subsequent cell-state data from the current cell-state data of the adjacent cells and of its own cell, and outputs the data to node *D*.
2. When $CLK = 1$ (holding mode), a loop around the latch inverters is established to store the current cell state. The current cell state (the voltage of node *P*) is equal to the state of the node *Q*. The latch is disconnected from node *D* during this holding mode.
3. When $CLK = 0$ (updating mode), the latch opens to receive the subsequent cell state *D*. The voltage of node *Q* is set equal to the voltage of node *D*. During this mode the current cell-state data is stored by the input capacitance of the output inverter. If necessary, an edge-triggered register can be used instead of a latch for fully static operation.

3.4. Dilation–erosion cell circuit

The functions dilation and erosion [Fig. 3(b) and (c)] are frequently used for morphological processing in combination with each other. They are single-threshold, symmetric functions, as illustrated in Fig. 8(a).

It is advisable in actual construction that both operations be switchable by a control signal. This is achieved as illustrated in Fig. 8(b). The circuit construction is basically the same as that of the Game-of-Life cell; the circuit also consists of a threshold logic gate and a memory, but the threshold logic gate consists of a single ν-CMOS inverter. The logic gate has a control input to switch between dilation and erosion; the switching is selected by a control signal that changes the threshold of the ν-CMOS inverter. The circuit operates in dilation if the control signal is 1 (*Vdd*), and erosion starts if the control signal is 0. For the input-capacitance parameters, see the figure caption.

3.5. Edge-detection cell circuit

The operation of edge detection [Fig. 3(e)] can be also implemented using a single-threshold gate and a memory. We considered merging an edge detection circuit with the dilation–erosion circuit mentioned above. The resultant construction is illustrated in Fig. 9. The circuit has two control signal inputs, *CS1* and *CS2*. If $CS1 = 1$ (V_{dd}), the circuit operates as an edge–detection cell, and if $CS1 = 0$, then it operates as a dilation–erosion cell. The dilation and erosion can be switched by *CS2*. For the input-capacitance parameters, see the figure caption.

4. OPERATIONS OF ν-MOS CELL CIRCUITS

4.1. Operation speed

Taking the Game-of-Life cell as an example, we simulated the speed capability of the designed cell circuits. The operation speed of the cell circuit is limited by the time taken to update the

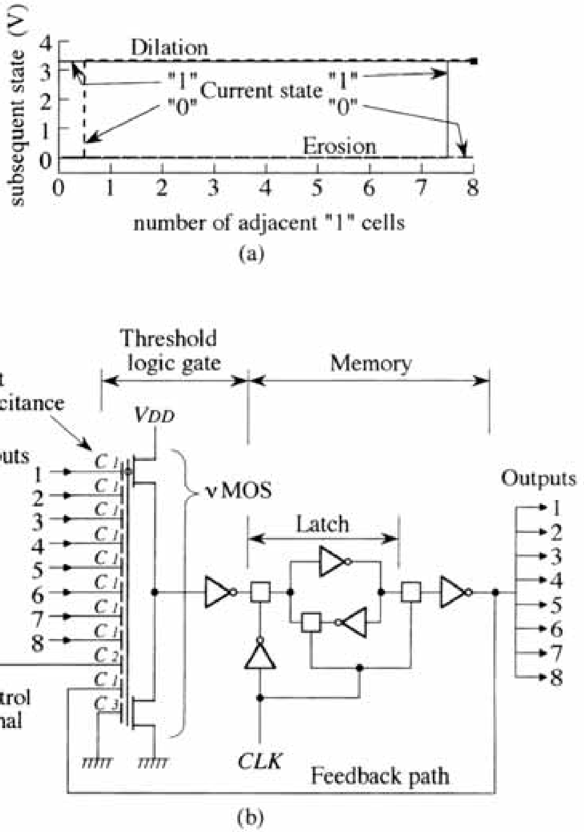


Fig. 8. Cell circuit for dilation–erosion operation: (a) cell function; (b) circuit construction. A sample set of parameters is: $C1$ = 50 fF, $C2$ = 400 fF, $C3$ = 150 fF, and 0.5 μm CMOS device parameters.

cell state. The transition time is defined as the delay between the clock input and the cell transition from the current state to the subsequent state.

To calculate the transition time, we considered setting up the cell circuit such that its state alternates between 0 and 1 with each tick of the clock (as in a T-type flip-flop). This can be performed using the test circuit of Fig. 10(a); three inputs are set at 1, four inputs at 0, and the

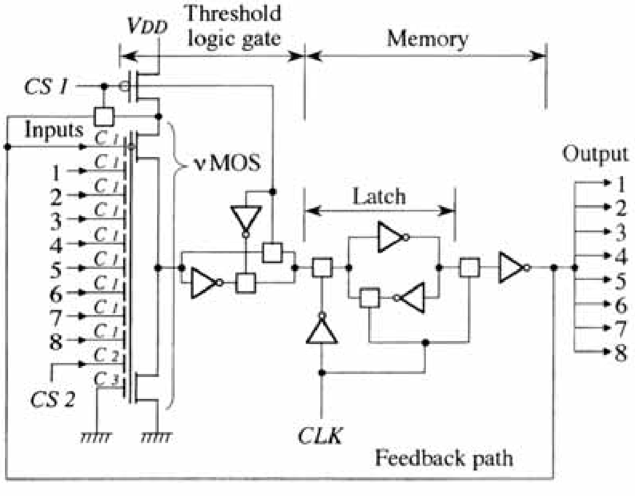


Fig. 9. Cell circuit for dilation–erosion plus edge-detection operation. A sample set of parameters is: $C1$ = 50 fF, $C2$ = 400 fF, $C3$ = 150 fF, and 0.5 μm CMOS device parameters.

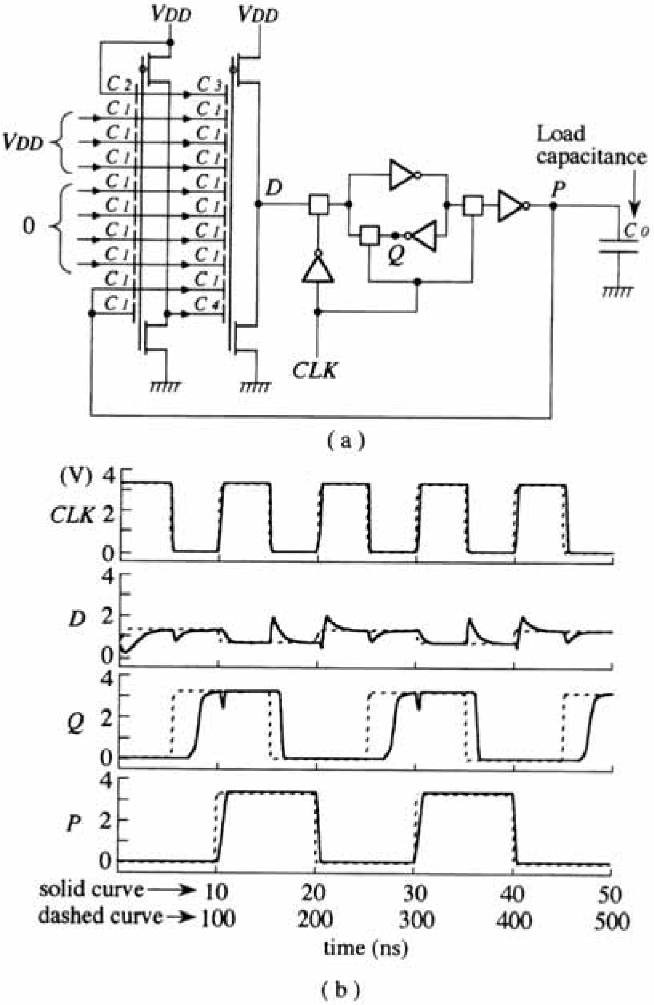


Fig. 10. Operation speed of the Game-of-Life cell circuit: (a) test circuit; (b) voltage waveforms of nodes *D* (the subsequent cell state), *Q* (current cell state), *P* (output to neighbors), and clock *CLK*. Simulated assuming the same parameters as in Fig. 7. The clock frequency is 10 MHz (dashed curves) and 100 MHz (solid curves).

other input is set equal to the current cell state. The simulated results are depicted in Fig. 10(b). Plotted are the voltage waveforms of nodes *D* (the subsequent cell state), *Q* (the stored state), *P* (the current cell state), and the clock *CLK*. In this example, the transition time was limited by the speed of the threshold logic gate and was about 5 ns, and the maximum operation frequency was 120 MHz.

4.2. Collective operations—noise removal and edge detection

We designed a sample cellular automaton by arranging the cell circuits of Fig. 9 (a dilation–erosion plus edge-detection cell) and operated it in simulation to perform noise cleaning and edge detecting. A large cell matrix cannot be dealt with because simulation time increases rapidly with the matrix size. Given here is an instance of a 15 × 15 cell matrix.

The result is illustrated in Fig. 11(a). To define a boundary condition, we enclosed the cell matrix with 64 bias cells, each of which was fixed at 0. Assuming the illustrated initial cell pattern (the object is a letter "A" with noise), we simulated the morphological processing with clocks. As shown in the figure, the noisy picture (step 0) is cleaned up by dilation and erosion (steps 1–4); then the edge of the object is extracted successfully (step 5). Plotted in Fig. 11(b) are the waveforms of the clock *CLK*, the control signals *CS1* and *CS2*, and the cell-output tran-

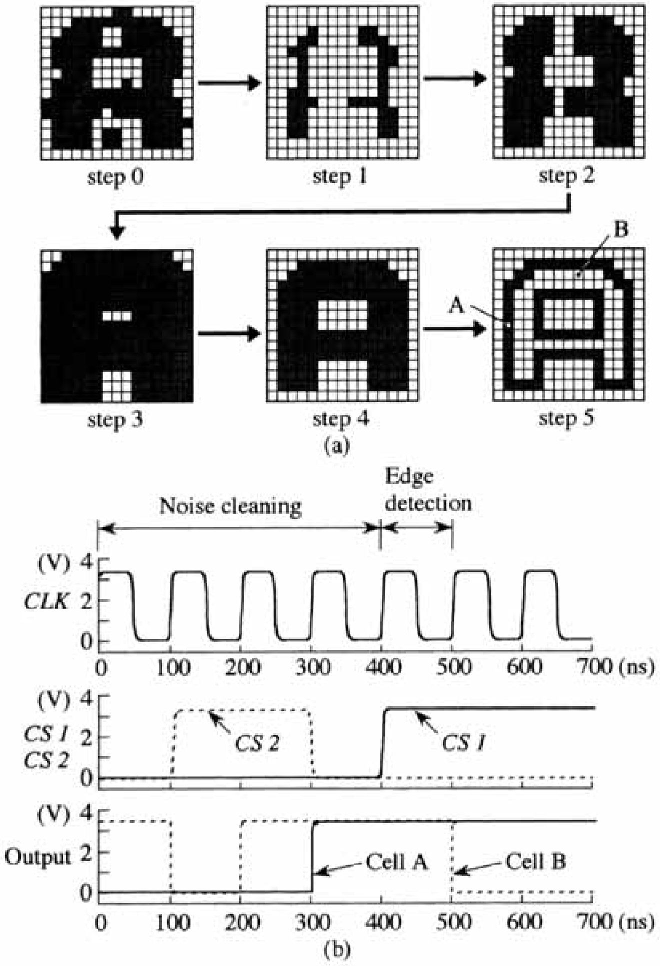


Fig. 11. Collective operation of the dilation–erosion plus edge-detection cell circuit of Fig. 9. A 15 × 15 cell array is simulated: (a) pattern transition of the cell array (the object in the picture is a letter A with noise); (b) voltage waveforms of clock *CLK*, control signals *CS1* and *CS2*, output voltages of the cells marked in step 5 of (a) by *A* and *B*. Simulated assuming a 10 MHz clock frequency and the same parameters as in Fig. 9.

sition for two cells marked by *A* and *B* [in step 5 of Fig. 11(a)]. This cellular automaton circuit operated at clock frequencies up to 110 MHz.

4.3. *Power dissipation*

There are three factors in power dissipation of CMOS circuits: the load-charging dissipation due to charging and discharging of load capacitances, the short-current dissipation caused by transient short current through the *n*-type and *p*-type MOSFETs, and the leakage dissipation due to subthreshold leak current in the MOSFETs. In ordinary CMOS circuits the first term is dominant, but in the present cell circuits the short-current dissipation can form the greater part of total dissipation because the ν-CMOS inverters are frequently operated with intermediate gate voltages.

The short-current dissipation is transient and minute in ordinary CMOS inverters because they are always used under the condition that the static gate voltage is completely 1 or completely 0. This is not true of ν-CMOS inverters because they are used for threshold logic applications and, therefore, their floating gates are frequently set at intermediate voltages between 1 and 0. Taking the Game-of-Life cell circuit [Fig. 7(b)] as an example, we simulated the short-current dissipation as a function of a number of "1" inputs. The result is illustrated in Fig. 12

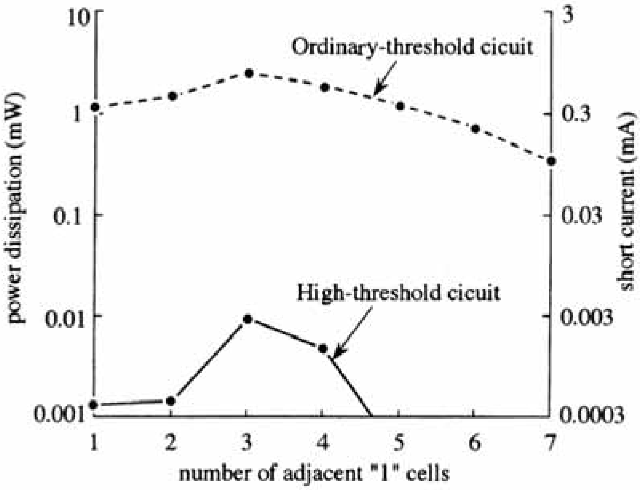


Fig. 12. Short-current power dissipation of the Game-of-Life cell circuits. Plotted as a function of the number of "1" inputs. The current state is assumed to be "0". The high-threshold circuit (solid curve) is compared with the ordinary-threshold circuit (dashed curve). For the threshold-voltage parameters, see Appendix.

by a dashed curve. It can be seen that large dissipation occurs for most input situations. This is quite inconvenient for LSI applications.

The short-current dissipation of the ν-CMOS inverters can be lowered by setting the threshold voltage of the constituent MOSFETs to a *high* value. The MOSFET threshold we assumed in the previous sections is 0.52 V for *n*-channel MOSFETs and −0.64 V for *p*-channel MOSFETs (see Appendix). We here introduce a *high-threshold* MOSFET that has a higher threshold than that of ordinary MOSFETs; as an instance, we assumed thresholds of 1.32 V (for *n*-channel) and −1.86 V (for *p*-channel). We designed a Game-of-Life cell circuit that used the ν-CMOS inverters composed of the high-threshold MOSFETs (the memory subcircuit was composed of ordinary-threshold MOSFETs). The simulated power dissipation of this cell circuit is illustrated in Fig. 12 by a solid curve as compared with the ordinary-threshold cell circuit. The short-current dissipation can be lowered by two orders or less than that of the ordinary-threshold cell circuit.

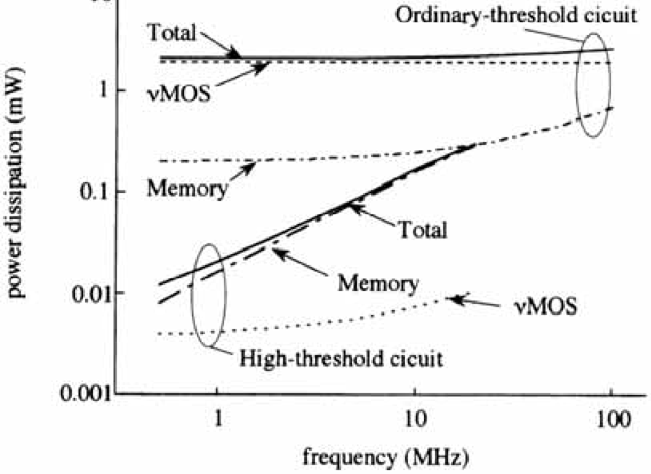


Fig. 13. Power dissipation versus clock frequency for the Game-of-Life cell circuits. The high-threshold circuit is compared with the ordinary-threshold circuit. Simulated assuming the same capacitance parameters as in Fig. 7.

The power dissipation depends on the clock frequency. We calculated the dissipation of the Game-of-Life cell circuit that uses the high-threshold MOSFETs as a function of the clock frequency. The results are illustrated in Fig. 13 as compared with the ordinary threshold circuit. The dissipation in the ordinary threshold circuit is large and almost independent of frequency because the short-current dissipation in the ν-CMOS gate circuit is dominant. In contrast, the dissipation in the high-threshold circuit has a low value; the short-current dissipation can be almost ignored and the power dissipation in the memory subcircuit is dominant.

The operation speed of circuits, in general, decreases with an increase in the threshold voltage of MOSFETs. The maximum operation frequency of the Game-of-Life cell circuit using the high-threshold MOSFETs was 20 MHz (in contrast with 120 MHz for the circuit using ordinary-threshold MOSFETs), but this is not a problem because all the cell circuits operate in parallel. In operation with a 1 MHz clock frequency (a sufficient speed for most applications), the power dissipation will be only 20 μW per cell circuit. Therefore, assuming a permissible power dissipation of 10 W per chip, we will be able to fabricate 5×10^5 cells in an LSI chip—a sufficient integration for many picture-processing applications.

5. SUMMARY

This paper proposed the construction of a cell circuit for implementing cellular-automaton devices that perform morphological picture processing. To produce the morphological processing, complex cell functions are generally required, such as variable-threshold, multithreshold, majority-decision, and weighted-input functions. To implement such functions in compact construction, we presented the idea of using the silicon functional device, the ν-MOS FET.

We designed sample cell circuits for several morphological processings, and simulated their operation to show that the expected cell functions can be obtained adequately. We also designed a sample cellular-automaton circuit (for processing of dilation-erosion plus edge-detection) using the proposed cell circuits, and demonstrated in simulation its operation of noise cleaning and edge extraction. The cell circuits can be designed into low power dissipation by use of the high-threshold MOSFETs. For instance, a low dissipation of about 20 μW per cell circuit can be expected at 1 MHz operation; therefore 10^5 or more cells that operate in parallel can be integrated into an LSI. The proposed ν-MOS cell circuits can be expected to provide compact cellular automaton devices for high-speed morphological processing.

Acknowledgements—This work was supported by the Ministry of Education, Science, Sports, and Culture under Grant-in-Aid for Scientific Research on Priority Areas, "Ultimate Integration of Intelligence on Silicon Electronic Systems" (head investigator: Tadahiro Ohmi, Tohoku University).

REFERENCES

1. Preston, K. and Duff, M., *Modern Cellular Automata*. Plenum Press, New York, 1984.
2. Toffori, T. and Margolus, N., *Cellular Automata Machines*. MIT Press, Cambridge, 1987.
3. Serra, J., *Image Analysis and Mathematical Morphology*. Academic Press, San Diego, 1982.
4. Pratt, W. K., *Digital Image Processing*. Wiley-Interscience, New York, 1991.
5. Nadler, M. and Smith, E. P., *Pattern Recognition Engineering*. Wiley, New York, 1993.
6. Shibata, T. and Ohmi, T., A functional MOS transistor featuring gate-level weighted sum and threshold operations. *IEEE Transactions on Electron Devices*, 1992, **39**(6), 1444–1455.
7. Shibata, T. and Ohmi, T., Neuron MOS binary-logic integrated circuits: part I, design fundamentals and soft-hardware-logic circuit implementation. *IEEE Transactions on Electron Devices*, 1993, **40**(3), 570–576.
8. Shibata, T. and Ohmi, T., Neuron MOS binary-logic integrated circuits: part II, simplifying techniques of circuit configuration and their practical applications. *IEEE Transactions on Electron Devices*, 1993, **40**(5), 974–979.
9. Weber, W., Prange, S. J., Thewes, R., Wohlrab, E. and Luck, A., On the application of the neuron MOS transistor principle for modern VLSI design. *IEEE Transactions on Electron Devices*, 1996, **43**(10), 1700–1708.

APPENDIX

Simulated results in this paper are for a set of 0.5 μm CMOS device parameters. The ν-MOS FET was modeled as a combination of input capacitor elements and an ordinary MOS FET. The gate width was set at 5 μm for both n-channel and p-channel MOS FETs—except 10 μm in the output buffer inverters. A 3.3 V power supply was assumed.

The MOS FET threshold voltage is a function of several parameters. We defined the threshold voltage as the gate voltage that gives a 1 μA drain current at a 3.3 V drain voltage and a 5 μm gate width. The threshold voltage was set at 0.52 V for n-channel MOS FETs and −0.64 V for p-channel MOS FETs. The threshold voltage of the high-threshold MOS FETs was set at 1.32 V (for n-channel) and −1.86 V (for p-channel).