

Design and Implementation of Quaternary NMOS Integrated Circuits for Pipelined Image Processing

著者	亀山 充隆
journal or publication title	IEEE Journal of Solid-State Circuits
volume	22
number	1
page range	20-27
year	1987
URL	http://hdl.handle.net/10097/46837

Design and Implementation of Quaternary NMOS Integrated Circuits for Pipelined Image Processing

MICHTAKA KAMEYAMA, MEMBER, IEEE, TAKAHIRO HANYU,
AND TATSUO HIGUCHI, SENIOR MEMBER, IEEE

Abstract—A new pipelined image processor using multiple-valued logic is effectively employed for systematic image processing without encoding and decoding because each pixel can be directly expressed by a single multiple-valued digit for images having several gray levels or several colors. Furthermore, from the viewpoint of hardware implementation, reduction in wiring complexity and reduction in chip area can be achieved in multiple-valued logic system.

In this paper, a new pattern matching procedure for performing four-valued image processing based on cellular logic operation is proposed, allowing two different templates to be processed simultaneously in a pipelined manner. Based on these double pattern matching cells, a compact NMOS image processing chip has been implemented. It is demonstrated that the compactness comes from reduced interconnections in the double pattern matching cells using a quaternary multiplexer or T gates, realized with pass transistors and multiple ion implants.

I. INTRODUCTION

IT HAS long been recognized that the use of multiple-valued logic in conventional digital systems has potential advantages [1], [2]. One of the most important advantages of a multiple-valued logic system is reduction in interconnections [3]. However, very few types of chips based on multiple-valued logic have been fabricated for practical applications [4]–[8].

This paper presents an implementation of a new quaternary NMOS integrated circuit for four-valued pipelined image processing using multiple ion implant technique. It has been demonstrated that not only two-valued image processing but also several-valued image processing is essential in applications such as robotics and medical image processing with several colors [9], [10]. The image processing algorithm employed here is based on cellular logic operations which perform digitally to transform an array of four-valued input data into a new data array. With images having four levels or colors, each pixel can be directly expressed by a single quaternary digit. We have shown that by the use of quaternary logic, systematic image processing can be effectively achieved without going back and forth between the actual image and the binary

data [11]. In four-valued image processing, cellular logic operations can be generalized by template or pattern matching. A new pattern matching cell is designed so that two different templates can be processed simultaneously in a pipelined manner. The ease of this double matching procedure is due to the full use of the quaternary information in the matched result [12].

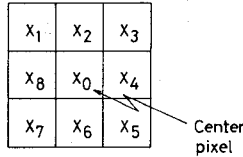
The image processing hardware implemented is a linear array of the pattern matching cells which has been fabricated based on the usual 10- μm NMOS process with the unit length $\lambda = 5 \mu\text{m}$ [13], [14]. The basic building block of the cell is a quaternary multiplexer which is also called a T gate. The use of the T gate enables a structured design of any quaternary logic system because both combinational and sequential circuits can be constructed using only T gates. The T gate consists of E/D NMOS transistors which have different threshold voltages realized by multiple ion implants. Since pass transistors are used for multiplexing the quaternary input signals, chip interconnections can be greatly reduced in comparison with other implementations. The integrated T gate is proved, by measurement, to have almost the desired characteristics with the different threshold voltages, 1.2, 2.7, and 3.6 V, and a propagation delay of 150 ns. A two-phase dynamic shift register which can be used for the pipelined operation in the array is also implemented using T gates. Finally, it is confirmed that the pattern matching cell operates with a 2-MHz data rate.

Since the data flow in the array of the pattern matching cells is completely in the form of quaternary signals, the processing capability per unit cell is greatly increased. In fact, the number of cells is reduced to 50 percent of a conventional binary implementation because of the direct processing on input pixels and the double matching procedure. This implies that the interconnections between the cells are greatly reduced.

II. IMAGE PROCESSING ALGORITHM USING MULTIPLE-VALUED LOGIC

The digital image discussed here is uniformly sampled and quantized to several levels.

Manuscript received December 31, 1985; revised May 18, 1986.
The authors are with the Department of Electronic Engineering, Tohoku University, Aoba, Aramaki, Sendai 980, Japan.
IEEE Log Number 8611326.

Fig. 1. Near-neighbor variables in a 3×3 window.

Suppose that a digitized image is approximated by equally spaced samples arranged in the form of an $M \times N$ array, where each element of the array is a discrete quantity:

$$A = \begin{pmatrix} A(0,0) & A(0,1) & \cdots & A(0,N-1) \\ A(1,0) & A(1,1) & \cdots & A(1,N-1) \\ \vdots & \vdots & & \vdots \\ A(M-1,0) & A(M-1,1) & \cdots & A(M-1,N-1) \end{pmatrix}. \quad (1)$$

Let the set of discrete quantities in the four-valued images be $L = \{0,1,2,3\}$. In the following discussion, the correspondence between the four-valued pixels and gray levels, or colored levels, is assumed predefined.

The image processing algorithm using multiple-valued logic is based on near-neighbor logic operations. Fig. 1 shows a configuration of a center pixel and its near-neighbor variables in the case of a 3×3 window.

The near-neighbor operations are essentially generalized by a template matching, and are described by the transition δ of a center pixel $x_0 \in A(i, j)$. The transition δ is determined by an appropriate combination of a center pixel x_0 and its near-neighbor variables x_1, \dots, x_8 as follows:

$$\delta: \begin{matrix} \begin{matrix} V_1^1 & V_2^1 & V_3^1 \\ V_8^1 & V_0^1 & V_4^1 \\ V_7^1 & V_6^1 & V_5^1 \end{matrix} & \rightarrow V^1, \dots, & \begin{matrix} V_1^k & V_2^k & V_3^k \\ V_8^k & V_0^k & V_4^k \\ V_7^k & V_6^k & V_5^k \end{matrix} & \rightarrow V^k \end{matrix}$$

where $V_\eta^\mu, V^\mu (1 \leq \mu \leq k, 0 \leq \eta \leq 8) \in L$.

The above notation is interpreted in the following way. If for any $\mu (1 \leq \mu \leq k)$ a center pixel $x_0 \in A(i, j)$ and its near-neighbor variables x_1, x_2, \dots, x_8 are equal to $V_0^\mu, V_1^\mu, \dots, V_8^\mu$, respectively, then x_0 changes to the new state V^μ where the transition will be made. Otherwise, x_0 does not change.

The near-neighbor operations are divided broadly into two types: nonrecursive (or simple) near-neighbor operations and recursive near-neighbor operations.

A. Simple Near-Neighbor Operation

In the simple near-neighbor operation, when the center pixel $A(i, j)$ of an input image A and its near-neighbor pixels $A(i-1, j-1), \dots, A(i+1, j+1)$ are equal to a_0, a_1, \dots, a_8 , respectively, $A(i, j)$ is changed to r_0 .

Otherwise, $A(i, j)$ remains unchanged. A simple near-neighbor operation can be formulated by a template with an input image A as follows:

$$\begin{matrix} A(\text{input}) & R(\text{output}) \\ \begin{matrix} a_1 & a_2 & a_3 \\ a_8 & a_0 & a_4 \\ a_7 & a_6 & a_5 \end{matrix} & \rightarrow r_0 \end{matrix}$$

Using the near-neighbor operations, the number of templates required for the state transition often becomes large in a multiple-valued logic system. In this case, compression of the templates is achieved by using a multiple-valued minimization technique [11].

B. Recursive Near-Neighbor Operation

This operation is also represented by the state transition of R using templates as follows:

$$\begin{matrix} R(\text{present state}) & A(\text{input}) & R'(\text{next}) \\ \begin{matrix} r_1 & r_2 & r_3 \\ r_8 & r_0 & r_4 \\ r_7 & r_6 & r_5 \end{matrix} & a_0 & \rightarrow r'_0 \end{matrix}$$

where a_0 and r_0 are the center pixel of the input image A and the next state R , respectively. The recursive operation is effectively repeated until R reaches a constant value. Sufficient conditions for the convergence of the state vector R within a finite repetition are given in [11].

III. QUATERNARY NMOS T GATE

A. Design of a Quaternary NMOS T Gate

A quaternary T gate can be used as a basic building block in a four-valued pipelined image processing hardware [12]. A quaternary T gate is a multiplexer function defined by

$$T_{\text{out}} = T(p_0, p_1, p_2, p_3; x) = p_i, \quad \text{if } x = i \quad (2)$$

where

$$p_0, p_1, p_2, p_3, x \in L = \{0, 1, 2, 3\}.$$

Fig. 2 shows a quaternary T -gate circuit using NMOS transistors. The threshold voltage of each transistor and the logical levels at the design stage are shown in Table I. The transistors $M_1 - M_8$ are used as the components of binary inverters and NOR circuits with different threshold voltages realized by multiple ion implants. The voltages V_a, V_b, V_d , and V_f at the points a, b, d , and f in Fig. 2 become V_{DD} if the logical value x is 0, 1, 2, and 3,

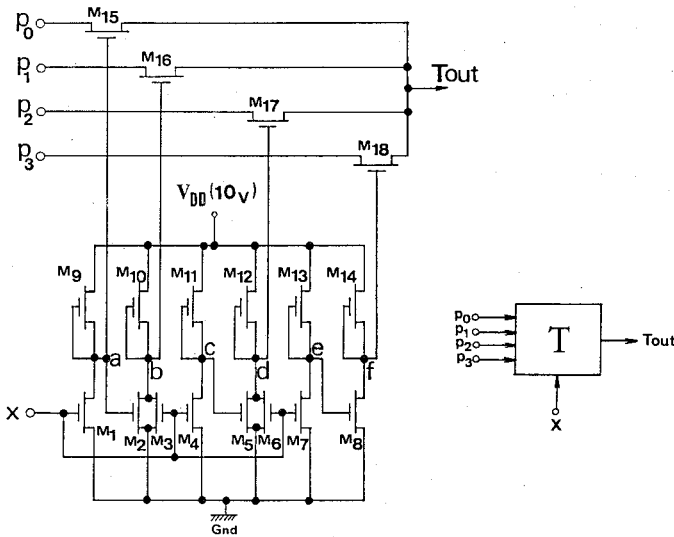


Fig. 2. Quaternary NMOS T-gate circuit.

TABLE I
PARAMETERS OF THE DESIGNED
T-GATE CIRCUIT

Tr.	Depletion/ Enhancement	V _{th}	Tr.	Depletion/ Enhancement	V _{th}
M1	Enhancement	0.0v	M10	Depletion	-3.0v
M2	∥	2.0v	M11	∥	∥
M3	∥	∥	M12	∥	∥
M4	∥	∥	M13	∥	∥
M5	∥	∥	M14	∥	∥
M6	∥	4.0v	M15	Enhancement (Pass Tr.)	1.0v
M7	∥	∥	M16	∥	∥
M8	∥	2.0v	M17	∥	∥
M9	∥	-3.0v	M18	∥	∥

Logical Value	0	1	2	3
Voltage	0 V	2 V	4 V	6 V

respectively. Otherwise, these voltages are almost zero. The pass transistors $M_{15}-M_{18}$ are used as analog switches, so that the data input p_i appears at the output T_{out} if the gate voltage of the corresponding pass transistor becomes high. By using the pass transistors, quaternary signals can be transferred to the output without decoding to binary signals, which makes reduction in complexity of the interconnections possible.

Four different enhancement-mode transistors and a depletion-mode transistor are formed by using three implant steps. The enhancement-mode transistors having the threshold voltages $V_{th} = 4$ and 2 V receive boron implants N_{DS1} and N_{DS2} , respectively. The depletion-mode transistors having the threshold voltage $V_{dep} = -3$ V receive phosphorus implant N_{DS3} . The pass transistors having the threshold voltage $V_{th} = 1$ V receive both the boron implant N_{DS1} and the phosphorus implant N_{DS3} , thus reducing the number of implant steps. Compared with the standard NMOS process, the increase in the number of the implant steps is only one. The change of the threshold voltage ΔV_{th} can be controlled by dose control according to

$$\Delta V_{th} = q \cdot N_{DS} / C_{ox} \quad (3)$$

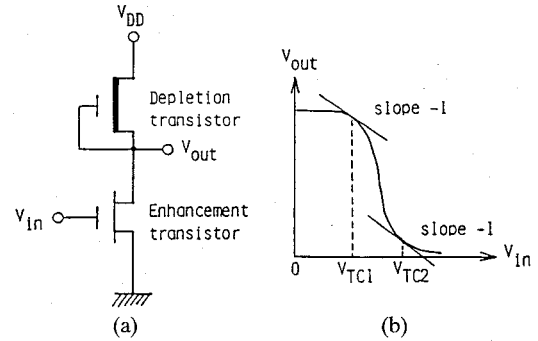


Fig. 3. Basic inverter. (a) E/D-type NMOS inverter circuit. (b) Static transfer characteristic.

TABLE II
EXAMPLE OF V_{TC1} AND V_{TC2}

V_{th}	0V	2V	4V
V_{TC1}	0.52V	2.52V	4.52V
V_{TC2}	1.50V	3.50V	5.50V
V_{inv}	1.01V	3.01V	5.01V

where q is an electron charge, C_{ox} is a gate capacitance per unit area, and N_{DS} is the net implant dose per unit area. If N_{DS} becomes very large, some nonlinear effects appear in (3).

Fig. 3 shows a basic inverter circuit used in the T gate. Let the depletion-mode pull-up transistor have a threshold voltage V_{dep} , and let the enhancement-mode pull-down transistor have a threshold voltage V_{th} . The switching voltages where the slope of the curve becomes -1 in Fig. 3(b) are given by

$$V_{TC1} = V_{th} + |V_{dep}| / \sqrt{\beta_R(\beta + 1)} \quad (4)$$

$$V_{TC2} = V_{th} + 2|V_{dep}| / \sqrt{3\beta_R}$$

where $\beta_R = \{(W_{pd}/L_{pd})/(W_{pu}/L_{pu})\} \cdot (\mu_{pd}/\mu_{pu})$, and where μ_{pu} , W_{pu} , and L_{pu} are, respectively, surface mobility, channel width and length in the gate region of the depletion-mode transistors. Similarly, μ_{pd} , W_{pd} , and L_{pd} are the corresponding parameters in the enhancement-mode transistors. The logical threshold voltage V_{inv} of the inverter is assumed to be $V_{inv} = (V_{TC1} + V_{TC2})/2$. In the Custom LSI Design and Development System at Tohoku University, the surface mobilities are measured to be $\mu_{pd} = 500 \text{ cm}^2 \cdot \text{s}^{-1} \cdot \text{V}^{-1}$ and $\mu_{pu} = 375 \text{ cm}^2 \cdot \text{s}^{-1} \cdot \text{V}^{-1}$. For the length-to-width ratio $Z = 4$ and $V_{dep} = -3$ V, the voltages V_{TC1} , V_{TC2} , and V_{inv} are given in Table II.

B. Characteristic of the Implemented T Gate

The process is based on the multiple ion implant steps shown in Table III. The dc transfer characteristic is shown in Fig. 4, where the output function $T_{out} = T(0, 1, 2, 3; x)$ is used as a quantizer function. The logical threshold voltages V_{inv} between the logical values 1-2 and 2-3 are smaller than the designed ones because of the nonlinear effect in

TABLE III
MULTIPLE ION IMPLANT QUATERNARY LOGIC

$V_{th}(v)$	Implant	Dose (cm^{-2})	Transistor
2.6	Boron	$N_{DS}=12.5 \times 10^{11}$	M_6, M_7
1.7	Boron	$N_{DS}=6.3 \times 10^{11}$	$M_2 \sim M_5, M_8$
0.2	None	None	M_1
-3.0	Phosphorus	$N_{DS}=9.5 \times 10^{11}$	$M_9 \sim M_{14}$ (Depletion Tr.)
0.8	Boron & Phosphorus	$N_{DS1} \text{ \& } N_{DS3}$	$M_{15} \sim M_{18}$ (Pass Tr.)

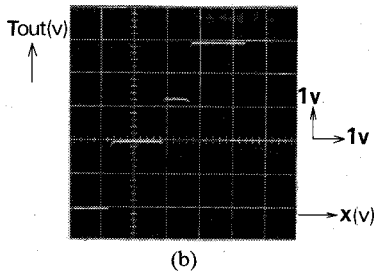
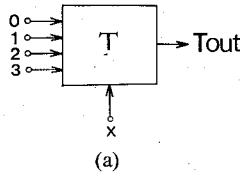


Fig. 4. (a) Quantizer function. (b) Static transfer characteristic of the quantizer.

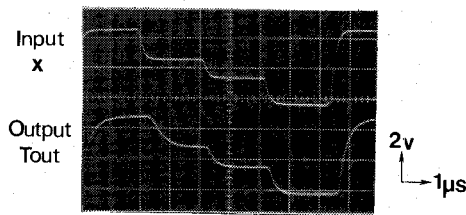


Fig. 5. Input and output response of the quantizer.

dose control due to leakage of implants. This can be easily improved by precise implants after the measurement of the nonlinear effects. Fig. 5 shows a transient response. The maximum propagation delay is about 150 ns, when the output is connected to the control terminal x of another T gate.

C. Dynamic Shift Register

Not only combinational but also sequential circuits can be constructed using T gates. Fig. 6 shows a dynamic shift register constructed by T gates. Using the two-phase non-overlapping clock ϕ_1 and ϕ_2 , the dynamic shift register employs charge storage on capacitive node at the gate connected to the control input x to retain logic levels

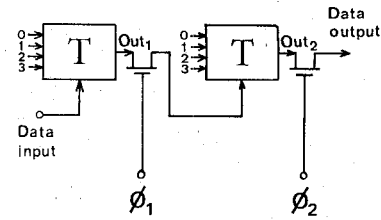


Fig. 6. Quaternary dynamic shift register.

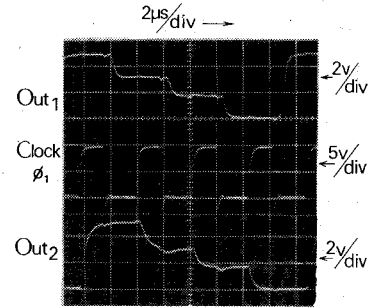


Fig. 7. Input and output waveforms of the dynamic shift register.

between clock periods. Fig. 7 shows a transient response. The minimum clock frequency of this circuit is limited to 1 kHz due to charge leakage from the soft node.

Although the quaternary dynamic shift register itself is not as simple as a corresponding binary implementation, it can be effectively used in a modified form such that the T gate is operated as both a dynamic shift register element and a logic gate. These examples are shown in the following section.

IV. HARDWARE STRUCTURE OF A PIPELINED IMAGE PROCESSOR

The four-valued image processor proposed here consists of the data shifter generating input image streams with different phase and commonly clocked near-neighbor processing stages in a pipelined manner as shown in Fig. 8.

A. Design of a Pattern Matching Cell

A quaternary pattern matching (PM) cell for double matching has been proposed [12]. Let us denote the input stream as $a_1 a_2 \cdots a_i \cdots$, the two finite pattern streams as $p_1 p_2 \cdots p_i \cdots$ and $q_1 q_2 \cdots q_i \cdots$, and the output stream as $c_1 c_2 \cdots c_i \cdots$, where $a_i, p_i, q_i, c_i \in L$, and the output digit c_i is defined as follows:

$$c_i = \begin{cases} 1, & \text{if the subsequence } a_1 a_2 \cdots a_i \text{ matches only the} \\ & \text{pattern } p_1 p_2 \cdots p_i \\ 2, & \text{if the subsequence } a_1 a_2 \cdots a_i \text{ matches only the} \\ & \text{pattern } q_1 q_2 \cdots q_i \\ 3, & \text{if the subsequence } a_1 a_2 \cdots a_i \text{ matches both the} \\ & \text{patterns } p_1 p_2 \cdots p_i \text{ and } q_1 q_2 \cdots q_i \\ 0, & \text{otherwise.} \end{cases}$$

(5)

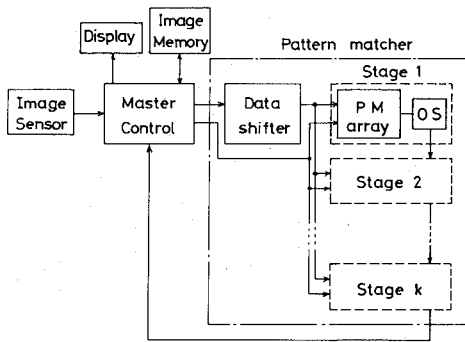


Fig. 8. Block diagram of the pipelined image processor.

TABLE IV
ACCUMULATOR FUNCTION

		c_{i-1}			
		0	1	2	3
b_i	0	0	0	0	0
	1	0	1	0	1
	2	0	0	2	2
	3	0	1	2	3

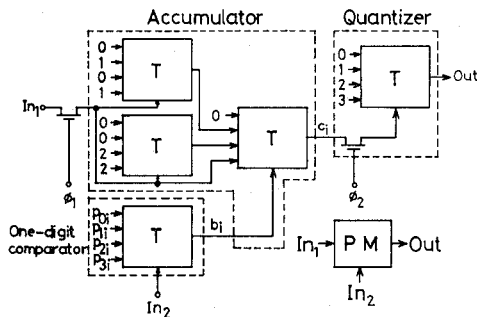


Fig. 9. Quaternary pattern matching cell.

According to this definition, two kinds of elements are necessary to construct a quaternary PM cell. One element is a one-digit comparator and the other is an accumulator.

The one-digit comparator corresponding to a pattern matching of an input pixel can be designed easily using a T gate:

$$b_i = T(\alpha_0, \alpha_1, \alpha_2, \alpha_3; a_i) \quad (6)$$

where a_i is the input digit operated on, and where each α_i can be determined according to the definition given in (5) as

$$\begin{cases} \alpha_{p_i} = \alpha_{q_i} = 3, & \text{if } p_i = q_i \\ \alpha_{p_i} = 1 \text{ and } \alpha_{q_i} = 2, & \text{if } p_i \neq q_i \\ \text{all other constants } \alpha_j = 0. \end{cases} \quad (7)$$

The accumulator receives the inputs c_{i-1} (the output from the previous cell) and b_i (the result from the comparator above), and it stores the accumulated matching results of c_{i-1} and b_i . The dynamic shift register element is also used in the accumulator for pipelined operation in the linear array of the PM cells. The function and the circuit diagram for this accumulator are shown in Table IV and Fig. 9, respectively. The accumulator function can be

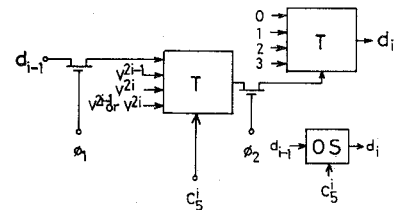
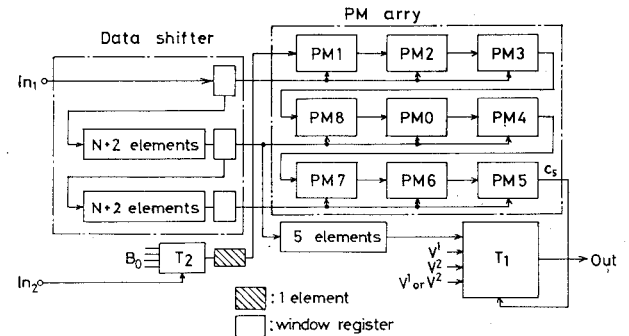


Fig. 10. Output selector.

Fig. 11. Basic structure of the quaternary image processing hardware for 3×3 near-neighbor operations.

written as

$$c_i = T(0, T(0, 1, 0, 1; c_{i-1}), T(0, 0, 2, 2; c_{i-1}), c_{i-1}; b_i). \quad (8)$$

Since three-stage T gates are contained in Fig. 9, the maximum clock frequency of the cell becomes about 2 MHz. The two-phase clock ϕ_1 and ϕ_2 is used for charge transfer. When ϕ_1 is high and ϕ_2 is low, the charge corresponding to the logic level is retained at the control input of the T gate of the accumulator, and when ϕ_1 is low and ϕ_2 is high, the charge is transferred to the quantizer.

B. Design of an Output Selector

The specified state transition within a PM array is performed by an output selector (OS), and the result is stored in the dynamic shift register in the OS for the continuous pipelined operation. Fig. 10 shows the structure of the OS using two quaternary T gates and two pass transistors. The two-phase clock ϕ_1 and ϕ_2 enables this system to perform the image processing in a pipelined manner synchronized with the operation of the PM cell.

C. Pipelined Image Processing Hardware

Fig. 11 shows the basic structure of quaternary image processing hardware (pattern matcher) for 3×3 near-neighbor logic operations consisting of the data shifter, the pattern matching cells, and the output selector. The image data transformed into serial data according to the scanning of two-dimensional image are entered into the data shifter through the master control processor. Shift registers in the data shifter store two contiguous $(N+3)$ pixel scan lines, and each window register puts the image data into the three PM cells simultaneously. All neighborhood transfor-

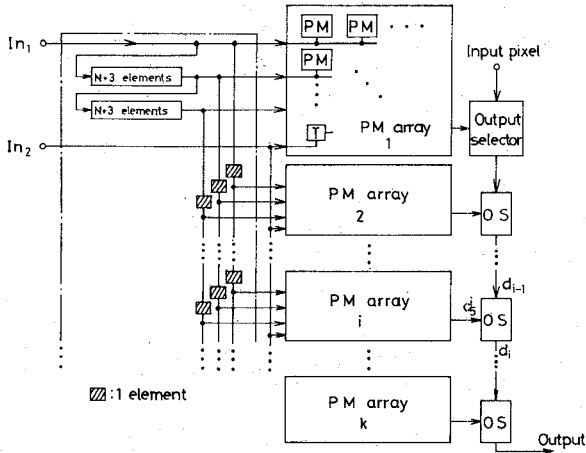


Fig. 12. Structure of the quaternary image processor.

mations and data transfers are performed within a nine-clock period.

The output image stream can be obtained at the same rate as the input scanning. The final pattern matching result of the input image is achieved by PM5 shown in Fig. 11. Using the matching result c_5 of PM5, the specified output transition for two templates can be performed simultaneously by the output T -gate T_1 as follows:

$$Out = \begin{cases} A, & \text{if } c_5 = 0 \\ & \text{(no match)} \\ V_1, & \text{if } c_5 = 1 \\ & \text{(match to the first template)} \\ V_2, & \text{if } c_5 = 2 \\ & \text{(match to the second template)} \\ V_1 \text{ or } V_2, & \text{if } c_5 = 3 \\ & \text{(match to both)} \end{cases} \quad (9)$$

where A , V_1 , and V_2 are the input pixel and the state transition values of the center pixels for two templates, respectively.

The structure of the simple near-neighbor operation with more than three templates is constructed by arranging multiple PM arrays as shown in Fig. 12. The matching result in each PM array is transformed to the transition value specified by each template in the OS which is connected to a PM array. The output d_i of the i th OS is determined by the matching result c_5^i from the i th PM array as follows:

$$d_i = \begin{cases} d_{i-1}, & c_5^i = 0 \\ V^{2i-1}, & c_5^i = 1 \\ V^{2i}, & c_5^i = 2 \\ V^{2i-1} \text{ or } V^{2i}, & c_5^i = 3 \end{cases} \quad (10)$$

where d_{i-1} is the output of the $(i-1)$ th OS and where V^{2i-1} and V^{2i} are the transition values for two templates in the i th PM array.

In the case of the recursive near-neighbor operation, the output data of the OS in the final state are fed back to the

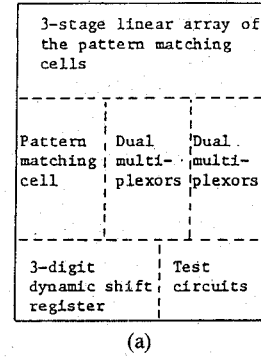


Fig. 13. Fabricated NMOS chip. (a) Chip functions. (b) Chip photomicrograph.

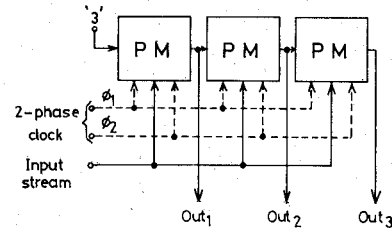


Fig. 14. Three-stage linear array of the pattern matching cells.

master control and the timing for sending these data is controlled by the master control processor. Then the output data and the input image data are transferred into the input nodes In_1 and In_2 in each stage.

By the extension of the hardware arrays, pipelined image processing for larger neighborhoods can easily be performed.

V. IMPLEMENTATION OF THE IMAGE PROCESSING HARDWARE

Fig. 13 shows the photomicrograph of the NMOS chip using the usual rules with the unit length $\lambda = 5 \mu\text{m}$ [13]. The chip consists of a three-stage linear array of the pattern matching cells, dual T gates, a pattern matching cell, and the three-digit dynamic shift register. The chip size is $5 \times 4 \text{ mm}^2$ with a total of 452 NMOS transistors.

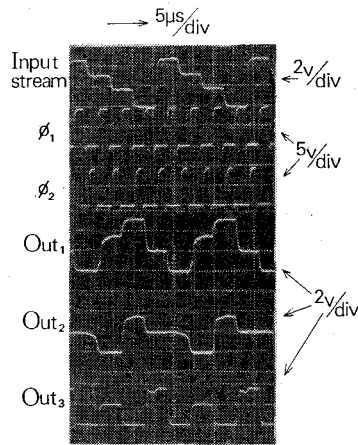


Fig. 15. Input and output waveforms of the pattern matching array.

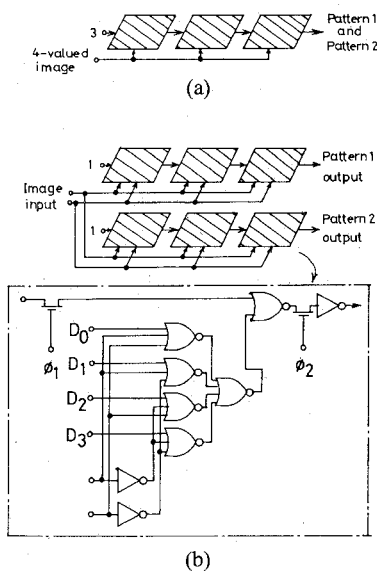


Fig. 16. Cellular arrays of pattern matching cells based on quaternary and binary logic. (a) Quaternary hardware. (b) Binary hardware.

Figs. 14 and 15 show the circuit diagram and the input-output waveforms of the three-stage linear array of the pattern matching cells, where the patterns (templates) are specified as follows:

$$\text{Pattern 1: } (p_1, p_2, p_3) = (0 \text{ or } 1, 0 \text{ or } 3, 2)$$

$$\text{Pattern 2: } (q_1, q_2, q_3) = (1 \text{ or } 2, 1 \text{ or } 3, 0).$$

With these patterns, the one-digit comparators become

$$b_1 = T(1, 3, 2, 0; a_1) \text{ for PM1}$$

$$b_2 = T(1, 2, 0, 3; a_2) \text{ for PM2}$$

and

$$b_3 = T(2, 0, 1, 0; a_3) \text{ for PM3}$$

according to (7). The operating frequency of the two-phase clock is 250 kHz in the experiment. The input stream is the periodic waveform of Fig. 15. The output of PM3 detecting Pattern 1 for (0, 3, 2) and Pattern 2 for (2, 1, 0) shows clearly that the double matching is performed by the three-stage linear array of the pattern matching cells.

TABLE V
COMPARISON OF THE IMAGE PROCESSING CELLULAR ARRAYS
(3×3 NEAR-NEIGHBOR OPERATION)

	Binary logic	Quaternary logic
Number of cells	18	9
Interconnections between cells	72	18
Number of transistors	576	414
Data rate (MHz)	2.7	2.0

TABLE VI
PARAMETERS OF THE T-GATE CIRCUIT WITH THE POWER SUPPLY
VOLTAGE $V_{DD} = 5$ V

V_{th} (V)	Implant	Transistors
-2.26	Phosphorus	$M_9 \sim M_{14}$
-0.31	"	M_1
0.90	Boron	$M_2 \sim M_5, M_8$
2.10	"	M_6, M_7
0.20	"	$M_{15} \sim M_{18}$

Logical value	0	1	2	3
Voltage	0.0V	1.2V	2.4V	3.6V

TABLE VII
COMPARISON OF THE PATTERN MATCHING CELLS WITH THE
POWER SUPPLY VOLTAGE $V_{DD} = 5$ V

	Binary logic	Quaternary logic
Static power dissipation ($\times 10^{-4}$ W)	11.5	5.3
Data rate (MHz)	2.7	1.8

Fig. 16 shows a cellular logic array based on typical binary logic components. The function of the binary array is equivalent to that of Fig. 14. The hardware based on binary components requires a 2-bit input for direct representation of an input pixel and a 2-bit output for parallel processing corresponding to the double pattern matching procedure.

From the above discussion, the feature of the quaternary image processing hardware is summarized as shown in Table V. The number of cells and interconnections between cells can be reduced to 50 and 25 percent, respectively. Moreover, great reduction of the number of transistors can be achieved. However, the speed will be rather slower than the binary array. From Table V, it is clear that a highly compact chip can be realized using the quaternary logic.

The recent dose control technique enables us to get the threshold voltage with high precision [15]. If such precise control is available, the power supply voltage can be made smaller in a usual environment because the designed threshold voltages become uniform as expected and the noise margins can be improved. For example, we can choose $V_{DD} = 5$ V which is the same as the power supply in a typical binary circuit. Table VI shows the logical levels and the threshold voltage of each transistor thus specified.

Under these conditions, the performance of the pattern matching cell is simulated using SPICE 2 as shown in Table VII. It is clear that the static power dissipation can be reduced to about half of the binary one, because of the compactness.

VI. CONCLUSION

A new pattern matching cell which can be used in image processing is designed and implemented using a quaternary T gate as a basic building block. The chip is proved, by measurement, to have almost the desired characteristics. One of the most important advantages of the quaternary image processing hardware is that the number of the cells can be reduced to 50 percent of the corresponding binary implementation because of the direct processing of input pixels and the double matching procedure. So, a highly compact image processor can be realized by using quaternary logic because the interconnection between the cells can be greatly reduced, thus opening up new possibility for future VLSI using multiple-valued logic.

ACKNOWLEDGMENT

The authors wish to thank Prof. T. Ito and Associate Prof. M. Esashi of Tohoku University for very helpful comments.

REFERENCES

- [1] G. Epstein, G. Frieder, and D. C. Rine, "The development of multiple-valued logic as related to computer science," *Computer*, vol. 7, pp. 20-32, Sept. 1974.
- [2] Z. G. Vranesic and K. C. Smith, "Engineering aspects of multiple-valued logic systems," *Computer*, vol. 7, pp. 34-41, Sept. 1974.
- [3] K. C. Smith, "The prospects for multivalued logic: A technology and applications view," *IEEE Trans. Comput.*, vol. C-30, pp. 619-634, Sept. 1981.
- [4] T. T. Dao, E. J. McClusky, and L. K. Russell, "Multivalued integrated injection logic," *IEEE Trans. Comput.*, vol. C-26, pp. 1233-1241, Dec. 1977.
- [5] K. W. Current, "High density integrated computing circuitry with multiple valued logic," *IEEE J. Solid-State Circuits*, vol. SC-15, pp. 127-131, Feb. 1980.
- [6] M. Stark, "Two bits per cell ROM," in *Proc. COMPON*, Feb. 1981, pp. 209-216.
- [7] M. Brillman, D. Etienne, J. L. Oursel, and P. Tatareau, "A 4-valued ECL encoder and decoder circuit," *IEEE J. Solid-State Circuits*, vol. SC-17, pp. 547-552, June 1982.
- [8] D. A. Rich, K. C. Naiff, and K. G. Smalley, "A four-state ROM using multilevel process technology," *IEEE J. Solid-State Circuits*, vol. SC-19, pp. 174-179, Apr. 1984.
- [9] K. Preston, Jr. *et al.*, "Basis of cellular logic with some applications in medical image processing," *Proc. IEEE*, vol. 67, pp. 826-856, May 1979.
- [10] G. J. Agin, "Computer vision systems for industrial inspection and assembly," *Computer*, vol. 13, pp. 11-20, May 1980.
- [11] M. Kameyama, K. Suzuki, and T. Higuchi, "Image processing algorithms for a multiple-valued array processor," in *Proc. 13th Int. Symp. MVL*, May 1983, pp. 236-241.
- [12] M. Kameyama and T. Higuchi, "A new architecture of a pipelined image processor based on quaternary logic circuits," in *Proc. 14th Int. Symp. MVL*, May 1984, pp. 92-97.
- [13] C. A. Mead and L. A. Conway, *Introduction to VLSI Systems*. Reading, MA: Addison-Wesley, 1980.
- [14] M. Kameyama *et al.*, "An NMOS pipelined image processor using quaternary logic," in *Dig. IEEE Int. Solid-State Circuits Conf.*, 1985, WPM 8.2, pp. 86-87.
- [15] A. B. Wittkower, P. H. Rose, and G. Ryding, "Advances in ion implantation production equipment," *Solid-State Technol.*, vol. 18, no. 12, p. 41, Dec. 1975.



Michitaka Kameyama (M'79) was born in Utsunomiya, Japan, on May 12, 1950. He received the B.E., M.E., and D.E. degrees in electronic engineering from Tohoku University, Sendai, Japan, in 1973, 1975, and 1978, respectively.

He is currently an Associate Professor in the Department of Electronic Engineering, Tohoku University. His general research interests include multiple-valued logic systems, VLSI-oriented special-purpose processors, highly reliable digital

systems, and robotics.

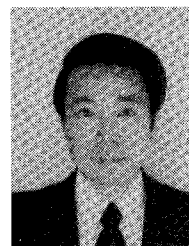
Dr. Kameyama is a member of the Institute of Electronics and Communication Engineers of Japan, the Society of Instrument and Control Engineers of Japan, the Information Processing Society of Japan, and the Robotics Society of Japan. He received the Awards for Excellence at the 1984 and 1985 IEEE International Symposiums on Multiple-Valued Logic (with T. Higuchi *et al.*) and the Technically Excellent Award from the Society of Instrument and Control Engineers of Japan in 1986 (with T. Higuchi *et al.*). He was the Program Co-chairman of the 1986 IEEE International Symposium on Multiple-Valued Logic.



Takahiro Hanyu was born in Hokkaido, Japan, on May 28, 1961. He received the B.E. and M.E. degrees in electronic engineering from Tohoku University, Sendai, Japan, in 1984 and 1986, respectively.

He is currently working towards the D.E. degree at Tohoku University. His main interests and activities are in the multiple-valued logic system and its applications to artificial intelligence.

Mr. Hanyu is a member of the Institute of Electronics and Communication Engineers of Japan. He received the Award for Excellence at the 1985 IEEE International Symposium on Multiple-Valued Logic (with M. Kameyama *et al.*).



Tatsuo Higuchi (M'70-SM'83) was born in Sendai, Japan, on March 30, 1940. He received the B.E., M.E., and D.E. degrees in electronic engineering from Tohoku University, Sendai, Japan, in 1962, 1964, and 1969, respectively.

He is currently a Professor with the Department of Electronic Engineering, Tohoku University. His research interests include design of 1-D and 2-D finite word-length digital filters, multiple-valued logic systems, fault-tolerant computing, and VLSI computing structure for signal

processing and image processing.

Dr. Higuchi is a member of the Institute of Electrical Engineers of Japan, the Institute of Electronics and Communication Engineers of Japan, and the Society of Instrument and Control Engineers of Japan. He received the Awards for Excellence at the 1984 and 1985 IEEE International Symposiums on Multiple-Valued Logic (with M. Kameyama *et al.*), the Outstanding Transactions Paper Award from the Society of Instrument and Control Engineers of Japan in 1984 (with M. Kwamata), and the Technically Excellent Award from the Society of Instrument and Control Engineers of Japan in 1986 (with M. Kameyama *et al.*). He was the Program Chairman of the 1983 IEEE International Symposium on Multiple-Valued Logic, and he is the Chairman of the Japan Research Group on Multiple-Valued Logic.