$$\begin{cases} I^* \leqq 2\mu_P(Q_B + Q_{ss})(-\phi_{MS} + 2\phi_F)\dfrac{W}{L} & (20') \\ V_{TP} + \sqrt{I^*}\left(\dfrac{1}{\sqrt{\beta_N}} + \dfrac{1}{\sqrt{\beta_P}}\right) \leqq E. & (21) \end{cases}$$

In order to design whole systems, the following condition, which is obtained by the condition of a multiple-fanout circuit, should be added to two conditions (20') and (21).

$$V_{TN} + \sqrt{I^*}\left(\frac{1}{\sqrt{\beta_N}} + \frac{1}{\sqrt{\beta_P}}\right) \leqq E \qquad (27)$$

where V_{TN} is the threshold voltage of an enhancement mode *n-ch* MOS FET (absolute value).

V. Conclusion

Nine basic fuzzy logic functions are all expressed only with the bounded difference and the algebraic sum. The algebraic sum is implemented only by wiring in the current mode circuits. Thus, the bounded-difference circuit is regarded as a basic logic cell.

Any complicated fuzzy hardware systems made up of nine basic fuzzy logic functions can be realized with only one kind of master slice including the bounded-difference (basic logic cell) and multiple-output n MOS current mirror array.

CMOS fuzzy logic semicustom IC presented here exhibits the following distinctive features. 1) The effect of the variation in V_{TH} and g_m on the electrical characteristics of the hardware systems can be cut off by adjusting the supply voltage to the appropriate value. Thus, the expensive ion implanter is not needed. 2) The basic logic cell exhibits good linearity which can not be easily achieved in voltage mode. 3) Since it does not need resistors nor isolation, it is suitable for a large scale fuzzy hardware system. 4) Circuits presented here can be appropriated to a binary logic and even to a 10-valued logic. 5) It exhibits the advantage over fuzzy information processing by using binary circuits, which arises from the ability to provide much more "functions per unit area." 6) It presents a low cost and a short term of design and fabrication.

Nine basic fuzzy logic function circuits presented here will be indispensable for a "fuzzy computer" in the near future.

Acknowledgment

The authors wish to thank Prof. T. Matsuo, Dr. M. Esashi, and Dr. K. Takahashi in Tohoku University, and Prof. K. Shono in Sophia University for their helpful advice on wafer processing.

References

[1] L. A. Zadeh, "Fuzzy sets," *Informat. Contr.*, vol. 8, pp. 338–358, 1965.
[2] T. Yamakawa, Y. Shirai, T. Inoue, and F. Ueno, "Implementation of fuzzy logic hardware systems—Three fundamental arithmetic circuits," *Trans. IECE*, vol. 63-C, pp. 720–721, Oct. 1980.
[3] ——, "Implementation of fuzzy logic (complement, bounded difference, bounded sum, and absolute difference) by current mode circuits," *Trans. IECE*, vol. 63-C, pp. 722–723, Oct. 1980.
[4] ——, "Construction of a programmable multifunction voltage mode fuzzy logic circuit," *Trans. IECE*, vol. 63-C, pp. 724–725, Oct. 1980.
[5] ——, "Implementation of a new analog multiplier and/or divider circuit with current mode," *Trans. IECE*, vol. 63-C, pp. 861–862, Dec. 1980.
[6] ——, "Implementation of fuzzy logic circuits and a programmable multifunction circuit with current mode," *Trans. IECE*, vol. 64-C, pp. 147–148, Feb. 1981.
[7] T. Yamakawa, T. Miki, and F. Ueno, "The fuzzy logic semicustom IC fabricated by using the standard *p* MOS process," *Trans. IECE*, vol. J67-C, pp. 600–601, July 1984.
[8] ——, "Basic fuzzy logic circuit formed by using *p* MOS current mirror circuits," *Trans. IECE*, vol. J67-C, pp. 1022–1029, Dec. 1984.
[9] Y. Shirai, T. Yamakawa, and F. Ueno, "A CAD oriented synthesis of fuzzy logic circuits," *Trans. IECE*, vol. J67-D, pp. 708–714, June 1984.
[10] T. Yamakawa, Y. Shirai, and F. Ueno, "Fuzzy logic circuits in current mode," in *The Analysis of Fuzzy Information*, vol. I, J. C. Bezdek, Ed. Gainesville, FL: CRC Press, to be published.
[11] F. Möllmer and R. Müller, "A simple model for the determination of I^2L base current components," *IEEE J. Solid-State Circuits*, vol. SC-13, pp. 899–905, Dec. 1978.
[12] H. H. Flocke, "I^2L design in a standard bipolar process," *IEEE J. Solid-State Circuits*, vol. SC-13, pp. 914–917, Dec. 1978.
[13] J. M. Koopmans and C. J. van der Meij, "Base resistance in I^2L structures: Its determination and its influence on upward current gain," *IEEE J. Solid-State Circuits*, vol. SC-17, pp. 783–786, Aug. 1982.
[14] T. T. Dao, E. J. McCluskey, and L. K. Russell, "Multivalued integrated injection logic," *IEEE Trans. Comput.*, vol. C-26, pp. 1233–1241, Dec. 1977.
[15] M. Akiya and S. Nakashima, "High-precision MOS current mirror," *Proc. IEE*, vol. 131, pt. I, pp. 170–175, Oct. 1984.

Ternary Scan Design for VLSI Testability

MOU HU AND KENNETH C. SMITH

***Abstract*—In this correspondence, a new scheme is proposed in which ternary clocking signals are used to replace binary clocking signals in VLSI scan-testing designs. This scheme has the same advantage of high testability as the binary scan method [1], but it eliminates the mode-selecting signal line. Since this mode-selecting line must be routed to each flip-flop in the binary scan scheme, the saving is significant in reducing the circuit interconnection complexity and chip area. This correspondence describes the new ternary scheme in detail, and also suggests appropriate circuit designs using CMOS technology. Furthermore, comparisons are made between ternary scan and binary scan [3] and between ternary scan and a scan scheme using binary with a local decoder [2].**

***Index Terms*—Multivalued signaling, scan design, ternary logic, testability, VLSI.**

I. Introduction

Testability is a very important aspect of VLSI design. This is because the test problem becomes more and more difficult as circuit complexity increases. The testability problem is further compounded if its consideration is left until too late a stage of the design process. Among many testability design techniques proposed, the scan method [1] is a well-known and effective one. In this method, the circuit is designed to have two operating modes, namely the normal mode and the scan mode. In the scan mode, all flip-flops in the circuit are connected to form one or more shift registers so that the states of each flip-flop can be preset and tested using a serial scanning process. By virtue of the scan mode the testability of the circuit is greatly enhanced.

One disadvantage of the scan method is the requirement of an additional mode-selecting input which must be routed to every flip-flop in the circuit. This increases the control-connection complexity

Manuscript received May 26, 1985; revised September 21, 1985. This work was supported in part by the Shanghai Institute of Railway Technology and the Natural Sciences and Engineering Research Council of Canada under Grant NSERC A1753.

M. Hu is with the Department of Communications, Shanghai Institute of Railway Technology, Shanghai, China.

K. C. Smith is with the Departments of Electrical Engineering and Computer Science, University of Toronto, Toronto, Ont. M5S 1A1, Canada.

IEEE Log Number 8406765.

(by 33 to 50 percent) and requires correspondingly greater chip area. Recently, a new clocking technique has been proposed [2] which removes the need for the mode-selecting line by adding decoding logic to each flip-flop.

In this correspondence this same problem is solved using a new approach. In the new scheme a ternary-clocking technique is used; that is, one of the binary-clocking signals is replaced by a ternary-coded line. Using ternary values, the mode-selecting signal can be superimposed on the clocking signal. Thereby the mode-selecting line is eliminated as a separate entity, while its selection function remains.

II. The Ternary Scan Design

A typical master–slave flip-flop with multiplexer employed in the binary scan method is shown in Fig. 1 [3].

This flip-flop has two data inputs, namely normal data (ND) and scan data (SD). These inputs feed to the master latch (ML) through a multiplexer (MPX). A mode-selecting signal (MS) controls MPX. Table I shows the signal-coding scheme. MCK and SCK are the master and slave clock signals, respectively.

If MS = 0, ND connects to ML and the circuit is in the normal mode; if MS = 1, SD connects to ML and the circuit is in the scan mode. The master-latch clocking signal MCK and the slave-latch clocking signal SCK are two nonoverlapping clock signals as shown in Fig. 2. When MCK = 1, and SCK = 0, ML is activated. When SCK = 1 and MCK = 0, the slave latch SL is activated. If MCK = SCK = 0, both ML and SL are inactive. Thus, the data stored in the flip-flop remain unchanged. Since MCK and SCK are nonoverlapping pulse signals, the state MCK = SCK = 1 does not occur.

From Figs. 1, 2, and Table I, it is seen that the mode-selecting signal (MS) is effective only during the period in which MCK = 1. When MCK = 0, MS has no effect. On the other hand, SCK is effective only when MCK = 0. Thus, it is possible to superimpose the MS signal on SCK and to eliminate the MS line.

To achieve this, a ternary logic technique will be used as follows: the logic value set of the ternary logic used is

$$Q = \{0, \tfrac{1}{2}, 1\}$$

where 0, ½, and 1 correspond to the lower, middle, and upper levels, respectively [4].

Fig. 3 is a ternary-scan master–slave flip-flop. The two data inputs ND and SD are the same as for the binary scan flip-flop. However, only two control signals are required, that is, MCK and SCK. The mode-selecting signal is superimposed on SCK. MCK is a binary signal while SCK is a ternary signal. The coding scheme is shown in Table II. The waveform for each signal is shown in Fig. 4.

In normal mode, both MCK and SCK operate conventionally, being 0- and 1-level pulses. When MCK = 1 and SCK = 0, ND connects to ML through MPX. When MCK = 0 and SCK = 1, the data stored in ML are gated into SL.

In scan mode, MCK remains as a sequence of 0 to 1 pulses, while SCK pulses range from the ½ to 1-levels. When MCK = 1 and SCK = ½, SD feeds to ML through MPX. When MCK = 0 and SCK = 1, the data stored in ML are gated into SL.

To guarantee the correct operation of SL, a level-conversion circuit (LC) is added at the control input of SL. This circuit will convert the ½-to-1-level pulses to 0-to-1-level pulses, leaving the 0-to-1-level pulses unaffected.

A detailed circuit design will be presented in the next section.

III. An Implementation Using CMOS Circuits

First let us consider the implementation of the ternary-scan flip-flop as shown in Fig. 3. Since ML and SL are the same as in the binary scheme, we will consider MPX and LC only.

MPX can be implemented using a (ternary) T gate with signal connections as shown in Fig. 5.

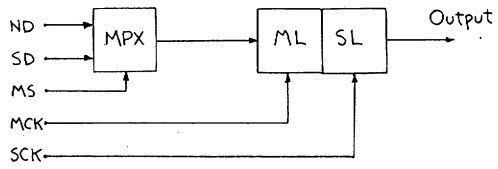

Fig. 1. Binary scan design storage and control.

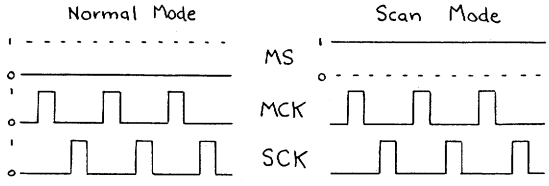

Fig. 2. Scan-control binary signaling.

TABLE I
Scan-Control Binary Coding

MS	MCK	SCK	Mode	Operation
0	0	0	Normal	None Active
0	1	0		ML Active
0	0	1		SL Active
0	1	1		Not Allowed
1	0	0	Scan	None Active
1	1	0		ML Active
1	0	1		SL Active
1	1	1		Not Allowed

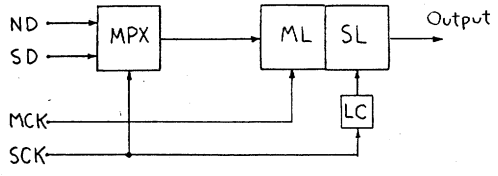

Fig. 3. Ternary scan design and control.

TABLE II
Scan-Control Ternary Coding

MCK	SCK	Mode	Operation
0	0	Normal	None Active
1	0		ML Active
0	1		SL Active
0	½	Scan	None Active
1	½		ML Active
0	1		SL Active

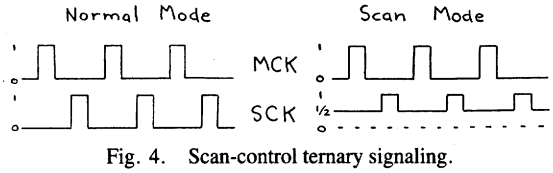

Fig. 4. Scan-control ternary signaling.

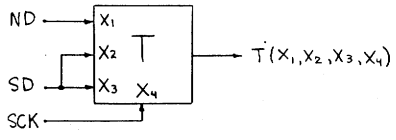

Fig. 5. A ternary T gate implementing MPX.

The operation of T gate [5] is defined as follows:

$$T(x_1, x_2, x_3, x_4) = x_1, \quad \text{if } x_4 = 0$$
$$= x_2, \quad \text{if } x_4 = \tfrac{1}{2}$$
$$= x_3, \quad \text{if } x_4 = 1.$$

In Fig. 5, if SCK = 0, the output is ND; if SCK = ½, the output is SD. Thus, the requirements of the coding scheme shown in Table II are met. When SCK = 1, the output of the T gate will be SD again. In Table II, this is a DON'T CARE term. Accordingly, Fig. 5 constitutes a correct logic design for MPX.

Many circuit implementations for the T gate are possible. However, since there are only two inputs for MPX in Fig. 3, a more economic design is possible as shown in Fig. 6.

In Fig. 6, SCK is connected to an inverter to generate the control signal C. C is connected to another inverter to generate control signal $\overline{C}$. The number beside each MOS transistor shows the relationship of the ratios of width to length of each transistor (assuming the n mobility to be twice that of p). Since the beta of T_2 is twice the beta of T_1, the threshold of the first inverter is between 0 and ½, resulting in the truth table for C as shown in Table III. Correspondingly, the threshold of the second (binary) inverter is near ½. The truth table for $\overline{C}$ is also shown in Table III.

C and $\overline{C}$ control the transistors T_5–T_8. The operation of the circuit of Fig. 6 is shown in Table III, Column 4.

The truth table for the level-conversion circuit LC is shown in Table IV. A CMOS circuit for LC is shown in Fig. 7. The relationship of the ratios of beta for T_1, T_2, T_3, and T_4 is 2:1:½:1. Thus, the threshold of the first inverter is between ½ and 1, while the threshold of the second inverter is lower, between 0 and ½. The latter choice has been made to increase the noise margin in the signal space between the ½- and 1-levels.

Now we will discuss the circuit design for generating the required ternary-clocking signals. Fig. 8 shows a possible ternary clocking-signal generator. Here MAX is a ternary gate whose output is the larger (more positive) of its inputs. The waveforms for each signal in Fig. 8 are shown in Fig. 9.

In Fig. 8, a binary clock-generator generates two nonoverlapping clocking signals CL_1 and CL_2. CL_1 is used directly as MCK. CL_2 is converted to SCK by a MAX gate under the control of mode-selecting signal MS.

The definition of the MAX function is as follows [6]:

$$\text{MAX}(x, y) = x \quad \text{if } x \geq y$$
$$= y \quad \text{if } y > x.$$

In normal mode, MS = 0, and SCK = CL_2. In scan mode, MS = ½, and the 0- to 1-level pulse of CL_2 is converted to the ½- to 1-level pulse of SCK. Many possible circuit designs for the MAX gate can be used here. Alternatively, a special circuit for which detailed performance data is not yet available, can be used as shown in Fig. 10.

In Fig. 10, the control signal NORMAL is a 1 in normal mode and a 0 in scan mode. In normal mode the circuit operates quite conventionally. In scan mode this (simple) circuit consumes additional power in order to establish SCK at the ½ level required.

IV. Comparisons

In comparing the ternary scan design to the conventional binary scan design [3], one can see that the master latch and slave latch are the same in both, while the multiplexers for each have the same level of complexity. However, the ternary scheme saves one signal line MS at the cost of adding a level conversion circuit LC for each SL flip-flop. The circuit LC is relatively low-cost being a locally connected circuit and implemented using only 4 transistors in CMOS technology. On the other hand, the normal MS line must be routed to every flip-flop in the circuit. It is well known that in VLSI designs, long routing paths are relatively more expensive in terms of chip area than a few locally connected devices. Thus, the advantage of the new scheme seems clear.

Both the ternary scheme and the binary-with-decoder scheme [2] eliminate the MS line by adding a few locally connected devices. The additional local circuit in each scheme is also of the same order

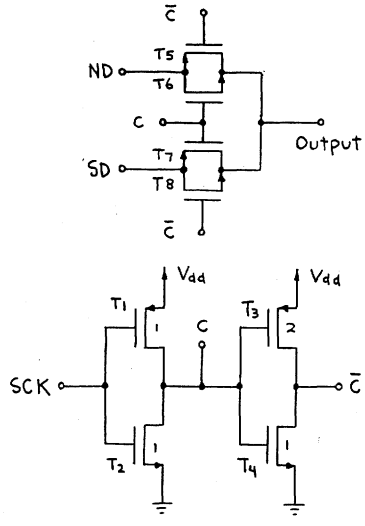

Fig. 6. A CMOS implementation of MPX as a two-input T gate with ternary control by SCK.

TABLE III
MULTIPLEXOR CONTROL CODING

SCK	C	$\overline{C}$	Operation
0	1	0	Output ND
½	0	1	Output SD
1	0	1	Output SD

TABLE IV
LEVEL CONVERTER TRUTH TABLE

SCK	Output
0	0
½	0
1	1

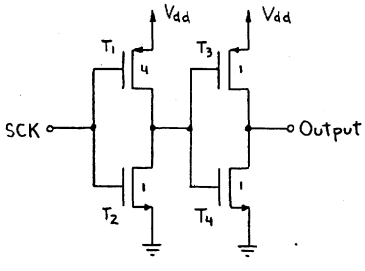

Fig. 7. A level-converter circuit.

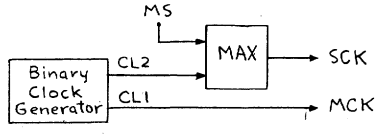

Fig. 8. Ternary clock-signal generation.

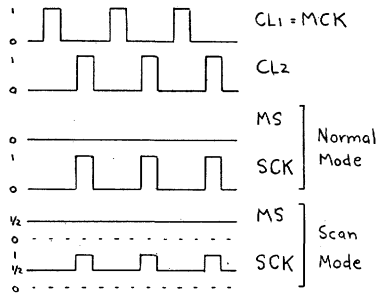

Fig. 9. Ternary clock-signal levels and timing.

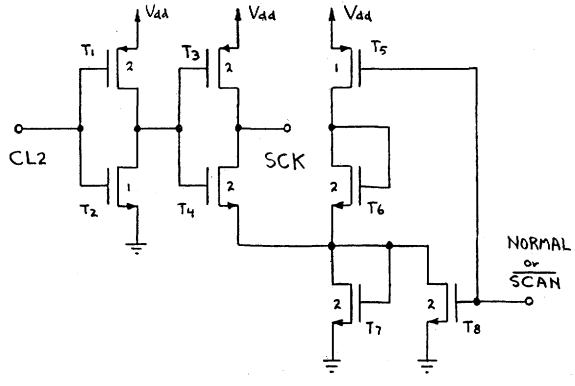

Fig. 10. A ternary slave-clock generator circuit.

of complexity. However for the binary-with-decoder scheme, two-clock level-sensitive operation is used in the normal mode only. For the scan mode, single-clock operation is employed. In contrast, for the ternary scheme proposed in this correspondence, the preferred two-clock level-sensitive mode is used in both normal and scan modes. This implies that the ternary-scan scheme can be made inherently more reliable than can the binary-with-decoder scheme.

REFERENCES

[1] E. B. Eichelberger and T. W. Williams, "A logic design structure for LSI testability," in *Proc. 14th Design Automat. Conf.*, 1977, pp. 462–468.
[2] M. R. Mercer and V. D. Agrawal, "A novel clocking technique for VLSI circuit testability," *IEEE J. Solid-State Circuits.*, vol. SC-19, pp. 207–212, Apr. 1984.
[3] M. J. Y. Williams and J. B. Angell, "Enhancing testability of LSI circuits via test points and additional logic," *IEEE Trans. Comput.*, vol. C-22, pp. 46–60, Jan. 1973.
[4] M. Hu and K. C. Smith, "On the use of CMOS ternary gates to realize a self-checking binary logic system," in *Proc 11th Int. Symp. Multiple-Valued Logic*, May 1981, pp. 212–217.
[5] T. Higuchi and M. Kameyama, "Ternary logic system based on T gate," in *Proc. 5th Int. Symp. Multiple-Valued Logic*, May 1975, pp. 290–304.
[6] K. C. Smith, "The prospects for multivalued logic: A technology and applications view," *IEEE Trans. Comput.*, vol. C-30, pp. 619–634, Sept. 1981.

On the Diagnosability of a General Model of System with Three-Valued Test Outcomes

A. SENGUPTA AND A. SEN

Abstract **— The problem of diagnosability of a system with three-valued test outcomes was considered in earlier works [1]–[3]. However all these works assume the system to be modeled as in [4]. In this correspondence, we consider a more general model of the system and study the diagnosability criteria in presence of three-valued test outcomes. In this model, each unit is tested jointly by a number of other units of the system as opposed to each test being carried out by a single unit of the system as in [4]. Necessary and sufficient conditions for the diagnosability of a system under this general model have been presented in this correspondence. Throughout the correspondence, diagnosability without repair has been considered.**

Index Terms **— Connection assignment, diagnosability, fault tolerance, graph models, multiple-valued test outcome, self-diagnosable system, syndrome.**

I. INTRODUCTION

The need of reliable computing systems has motivated the researchers to investigate about diagnosable systems where the different units of the system test each other. Preparata *et al.* [4] first introduced the concept of diagnosable systems and diagnosability of systems where, in presence of failures, the faulty units can be uniquely identified. Since then, considerable research has been reported in the literature on diagnosable systems in connection with their diagnosability [1]–[13]. Most of these (except [5], [7], and [11]) are concerned with the model of the system introduced in [4], assuming only binary test outcomes to be present (except [1]–[3]). Butler [1] presented some properties of a system in connection with its diagnosability in case of three-valued test outcomes. The necessary and sufficient conditions for the diagnosability of a system with three-valued test outcomes were presented in [3]. However, all these works [1]–[3] involving three-valued test outcomes assume the system being modeled as in [4]. In this correspondence, we consider a more general model of the system proposed in [14] and study its diagnosability in presence of three-valued test outcomes. We will first describe the model briefly and compare it to the other existing models.

The model of the system can be represented as follows. A system is assumed to be composed of a number of units and to check whether the units are functioning properly, each unit is tested by a number of tests. Each of these tests is carried out (or contributed) by a number of other units of the system (the model [4] effectively assumes that each test is contributed only by a single unit of the system). Depending on the status (i.e., faulty or fault-free) of the units contributing to a test, the test might be a valid or an invalid test. The outcome of each such test is three-valued. We will consider two cases with respect to the test outcomes: a) incomplete test outcomes and b) incorrect test outcomes. In case a, the outcome is 0 if the tested unit passes the test, 1 if it fails the test and 2 if the result of testing is somehow missing possibly because of faulty transmission. Whether an outcome of a test represents the true status of the tested unit depends on whether the test is valid. We will assume that the "finding" of a valid test is always correct so long as the outcome is not missing, whereas the outcome of a test does not convey any information about the status of the tested unit if the outcome is missing or the test is invalid. Thus, if the outcome of a valid test is 0, then the tested unit is fault-free and if it is 1, the tested unit is faulty. The case b arises in case of intermittent failures. If a unit is intermittently faulty, then a valid test might discover it to be fault-free possibly because the unit was working in a fault-free manner when the testing was carried out. Accordingly, a three-valued test outcome $0, 1, 0'$ was considered in [2] where the test outcomes 0 and 1 represent the "finding" of the tested unit exactly as before, whereas the outcome $0'$ represents the incorrect pass of an intermittently faulty from a valid test. For diagnosis purpose, of course, $0'$ behaves just like a 0. In both cases a and b, the diagnosability problem arises because of the fact that two distinct sets of faulty units may produce the same set of test outcomes. In case a, we will say a system is t/x diagnosable if all the faulty units can be uniquely identified from the set of test outcomes whenever there are no more than t faulty units and no more than x missing test outcomes. In case

Manuscript received May 28, 1985; revised September 13, 1985.
The authors are with the Department of Computer Science, University of South Carolina, Columbia, SC 29208.
IEEE Log Number 8406764.