Proc. ITC-91

# BUILT-IN SELF-DIAGNOSTIC READ-ONLY-MEMORIES

Prawat Nagvajara, Department of Electrical and Computer Engineering
Drexel University, Philadelphia, Pennsylvania 19104
Mark G. Karpovsky[1], Department of Electrical, Computer and System
Engineering, Boston University, Boston, Massachusetts 02215

Abstract—The design of built-in self-diagnostic read-only-memories (ROMs) which extends the concept of the built-in self-test to provide fault-masking in the ROM normal operating mode is presented. Switch-level fault analysis of the ROMs showed that most of the single transistor stuck-open/on faults (on the order of 95% or more) will result in errors confined in the content of a single address. The presented diagnostic scheme identifies these errors and locates the address for which the data are corrupted. In the ROM operating mode the errors are corrected at the ROM outputs when the content of corresponding address is read. A VLSI implementation is presented where the area overhead is estimated to be on the order of 15% or less.

## 1 Introduction

Read-only-memories (ROMs) are often used for storing the microprograms in the structured controllers of microprocessors. The function of a control ROM (microcode ROM) is critical to the operation of a processor. To ensure the correct operation, an extensive test procedure for the control ROMs is periodically performed in the off-line mode, and when faults are detected, the microprocessors are discarded from the system. The technique that will be presented in this paper extends the ROMs off-line test procedure to provide fault-masking capability where the errors appeared at the ROMs outputs due to internal faults are identified and located in terms of the ROMs address. During the ROM normal operation, the errors are corrected at the ROM outputs when the address that has been affected by the fault is read.

Since control ROMs are embedded functional blocks the proposed test/diagnosis and error-correcting technique will be implemented by means of an additional circuits where the term "built-in self-diagnostic" is used analogous to the built-in self-test (BIST) [1]-[7]. While the scan techniques [5]-[7] can provide access to the test data the amount of time required can be too great. Alternatively, BIST design eliminates the scan test data, and consequently, achieves the minimal test time. Built-in self-test designs for embedded ROMs, RAMs and PLAs have been proposed in [1]-[4], where the techniques are basically based on exhaustive testing and test response compaction [5]-[10]. Zorian and Ivanov [1] has proposed a BIST ROM based on the "Exhaustive Enhanced Output Data Modification." (EEODM) technique which reduces the probability of an error escape due to test response compaction (aliasing probability) compared to techniques based on multiple-input signature register (MISR) [11], parity check and check-sum [6], [12]. The EEODM scheme uses MISR that can be operated in both directions of shifts to compute the polynomial division of the ROM contents by the feedback polynomial and its reciprocal polynomial, thus, obtain two separate signatures. This scheme also stores the expected quotients resulted from the polynomial division of the contents of the ROM by the feedback polynomial using an additional ROM-array column.

The proposed built-in self-diagnostic technique will be based on generalized Hamming codes. These codes had been used for design of single byte error-correcting memories [13], [6] where each memory word is encoded and the errors are concurrently corrected when a word is read. However, for this off-line test/diagnostic scheme, the redundant information of the ROM contents is stored as two additional words as opposed to encoding of each word. The authors has previously applied the generalized Hamming codes for location single faulty chips on the boundary scan boards [14]. The weighted check-sum [15] is another scheme that had been proposed for concurrent error-correction in matrix-type computations by array processors. It is suitable for matrix computation application since the arithmetical operations are available in those array processors. Unfortunately, the attempt to apply the weighted check-sum in the diagnostic technique for the ROMs revealed that the

[1] This work has been supported by NSF under Grant MIP881348

hardware required will be much greater than the algebraic scheme like Hamming codes.

In the next section the fault analysis of the ROMs based on switch-level models [16]-[18] will be presented. It will be shown that most of the single stuck-open faults and the single stuck-on faults will give rise to erroneous outputs when the content of only one address is read. This type of errors will be called "single address errors." In Section 3 the test, diagnostic and error-correcting procedures are described. In Section 4 a VLSI implementation of the built-in self-diagnostic is presented with the estimated silicon area overhead that is shown to be on order of 15% or less for typical control ROMs. In Section 5 the technique for obtaining 100% fault coverage by using the unused ROM address is discussed.

## 2 Switch-level Faults in ROMs

ROM is a two-level AND-OR structure which is commonly implemented by an NOR-NOR array. Consider the block diagram of a ROM with m inputs and n outputs shown in Fig. 1. The AND-array is implementing a decoder with m inputs and $2^m$ outputs called "Word-lines," and the OR-array is implementing the n-bit word memory cells. The input buffer and the output buffer are commonly implemented by static or dynamic D flip-flops.

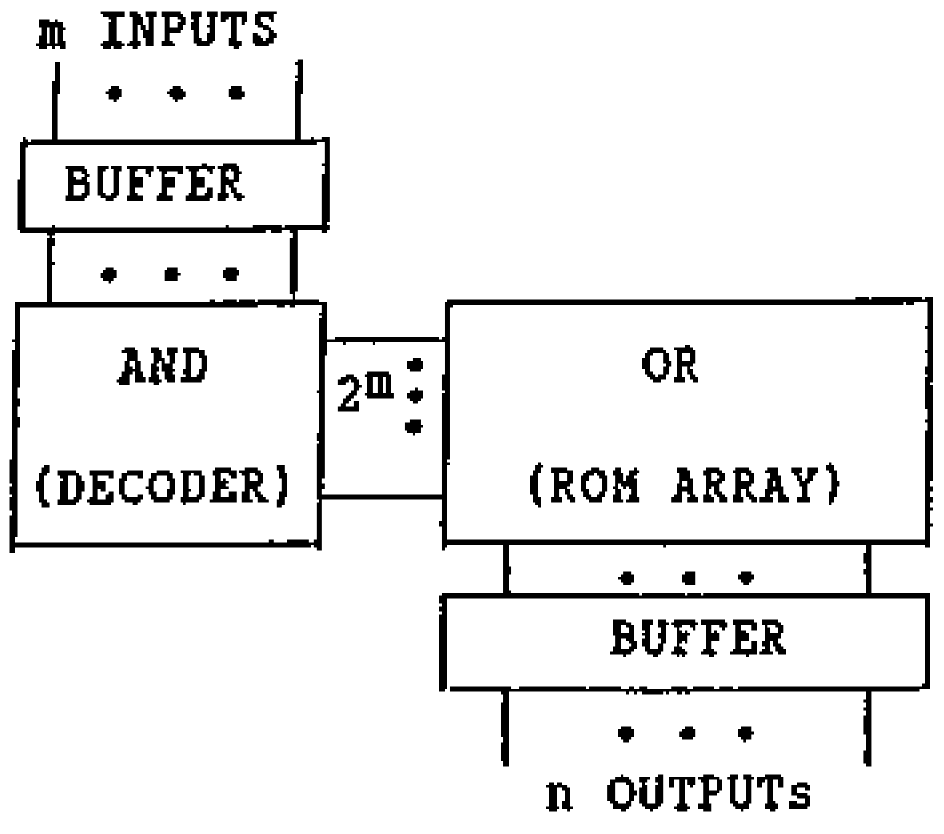
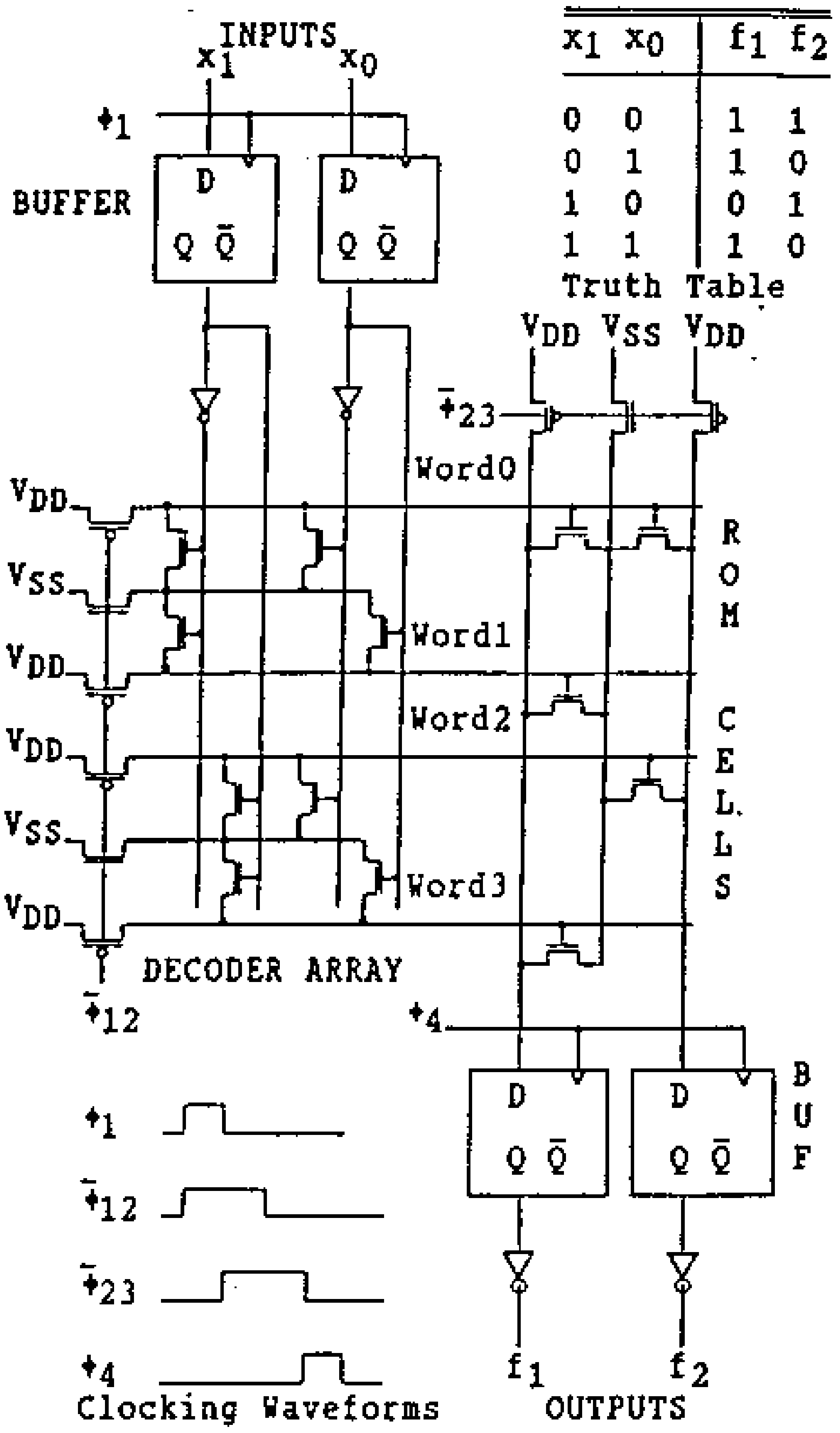


Fig. 1. ROM Block diagram

In Fig. 2 a transistor circuit of a dynamic CMOS using four-phase clocking scheme [19] with two inputs and two outputs is shown. This circuit will be used in the discussion of faults. The circuit functions as follow. The inputs are latched during $\phi_1=1$ (see the clocking waveforms in Fig. 2). The word-lines are precharged when $\phi_{12}=0$ by the p-transitors and evaluated when $\phi_{12}=1$ by the n-transistors. In this case only one of the word-line will be a 1. The output-lines are precharged when $\phi_{23}=0$ and evaluated when $\phi_{23}=1$ by the p-transistors and the n-transistors, respectively. A content of the ROM is latched into the output buffer when $\phi_4=1$. In general, the number of the decoder and the ROM-cells $V_{SS}$ switches (the n-transistors) is equal to $2^{m-1}$ and n/2 (for n is an even integer), respectively.



| $x_1$ | $x_0$ | $f_1$ | $f_2$ |
|---|---|---|---|
| 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Truth Table

Fig. 2. Example of ROM transistor circuit

A stuck-open fault is the case when a MOS transistor (switch) becomes nonconductive regardless of the gate voltage. A stuck-on fault, on the other hand, is the case when a switch becomes conductive regardless of the gate voltage [16]-[18]. There is a difficulty in predicting the value of the gate output voltage when stuck-on faults occur at the precharge transistors and $V_{SS}$ switches since there is a path between $V_{DD}$ and $V_{SS}$. In this case the word-lines or the outputs will have an intermediate voltage lies between 0V. to 5V.

The effects of single stuck-open and stuck-on faults in terms of the line stuck-at faults is discussed below using the ROM transistor-circuit given in Fig. 2.

## 2.1 Faults in the Decoder

(i) Faults in Decoder Array: A stuck-open fault in the decoder array causes two word-lines to be a 1 simultaneously for one input combination. For example, if the top left-hand corner transistor of the decoder array is stuck-open, (see Fig. 2), when the input (1 0) is applied the word-line number zero and the word-line number two are both a 1. In this case the ROM output is (1 1), the componentwise OR between (1 1) the content of address zero and (0 1) the content of the address two.

Errors appearing at the ROM outputs are defined as the componentwise exclusive-OR between the fault-free outputs and the erroneous outputs. For the above stuck-open fault, the error is equal to (1 0) which occurs when the input (1 0) is applied. Faults which only distort the content of only one address are called "Single Address Faults" (single address errors). For example, stuck-open fault in the decoder array are single address faults, moreover, there are $m2^m$ stuck-open faults in the decoder array (where m denotes the number of ROM inputs).

A stuck-on fault in the decoder array causes a word-line to stuck-at 0 which is again a single address fault. This is equivalent to the content of an address equals to a pattern of all one's. Note that in this case, the stuck-on faults do not create paths between $V_{DD}$ and $V_{SS}$ due to the clocking of precharge and $V_{SS}$ switches.

(ii) Faults in the Precharge Switches: A stuck-open fault causes a word-line stuck-at 0 which is a single address fault. The number of precharge switches stuck-open is $2^m$. A stuck-on fault, however, creates a path between $V_{DD}$ and $V_{SS}$ where the value of a word-line may be biased towards a 0 or 1.

(iii) Faults in the $V_{SS}$ Switches: A stuck-open fault causes two adjacent word-lines to stuck-at 1 and it is not a single address fault. The number of $V_{SS}$ switches is $2^{m-1}$. A stuck-on fault in a $V_{SS}$ switch is undetectable, however, there is a static power dissipation when $\phi_{12}=1$.

## 2.2 Faults in the ROM Array

(i) Faults in the ROM Cells: A stuck-open fault or a stuck-on fault causes one bit in the ROM content to become erroneous (a single-address fault). The number of the transistors in the ROM cells is equal to the number of one's in the ROM contents and assumed to be $(n/2)2^m$ (where n is a number of bits in a word).

(ii) Faults in the Precharge Switches: A stuck-open fault causes an output line to stuck-at 1 and it is not a single address faults. A stuck-on fault gives rise to an output biased towards a 0 or 1, and it is not a single address fault. There are 2n non-single address faults in the ROM array.

(iii) Faults in the $V_{SS}$ Switches: A stuck-open fault causes two adjacent output lines to stuck-at 0 and a stuck-on fault is undetectable. There are n/2 $V_{SS}$-switches in the ROM array.

## 2.3 Faults in the Buffers

Switch-level faults in the buffers (static DFFs or dynamic D latches) have one of the effects that can be modeled as the input-lines and the bit-lines stuck-at faults. Other effects are the incorrect timing of the latched inputs, the path between $V_{DD}$ and $V_{SS}$, and undetected faults. A stuck-open fault in an inverter eventually causes a input-line or bit-line(s) or output-line to stuck-at 0 or 1 depending on the occurrence of fault in the pMOS or the nMOS transistor. A stuck-on fault causes an inverter output to be biased towards a 0 or 1. Faults in the ROM buffers and the inverters are not single address faults and the number of these faults will be estimated as the input-lines and bit-lines stuck-at faults.

## 2.4 The Number of ROM Single Address Faults

A general expression for the total number $F_t$ of detectable faults (excluding the intermediate gate-voltage faults in the decoder), and the total number $F_s$ of single address faults in the four-phase dynamic CMOS ROM with m inputs and n outputs (e.g., see Fig. 2) are given as follows:

$$F_t = \underset{\text{Decoder}}{[2m2^m+2^m+2^{m-1}]} + \underset{\text{ROM array}}{[2(n/2)2^m+2n+(n/2)]} + \underset{\text{Buffer\&Inverters}}{2[4(m+n)+(m+n)]} = (2m+1.5+n)2^m + 12.5n + 10m, \tag{1}$$

$$F_s = \underset{\text{Decoder}}{[2m2^m+2^m]} + \underset{\text{ROM array}}{[2(n/2)2^m]} = (2m + 1 + n)2^m. \tag{2}$$

From (1) and (2) we see that most of the single switch-level faults affect the ROM outputs as the single address errors. For the typical sizes of microcode ROMs (e.g., 9K×38 ROM in the HP 9000 CPU chip [20]) and signal processor program ROMs (e.g., 1536×16 ROM in the TI TMS32010 and 1K×20 ROM in the Fujitsu MB8764 [22]), the percentages of the single address faults for (m,n) = (9,60), (10,20) and (13,40) is 97%, 98% and 99%, respectively. If the assumption that the intermediate output-voltage faults in the decoder will result in non-single address faults is made, $F_t$ will have the additional $2^m$ faults, and the percentage of single address faults is 96%, 96% and 98% for the above (m,n) parameters. Therefore, a built-in self-diagnostic scheme which identifies the magnitudes and addresses of the single address errors is desirable.

## 3 Built-in Self-Diagnostic Technique

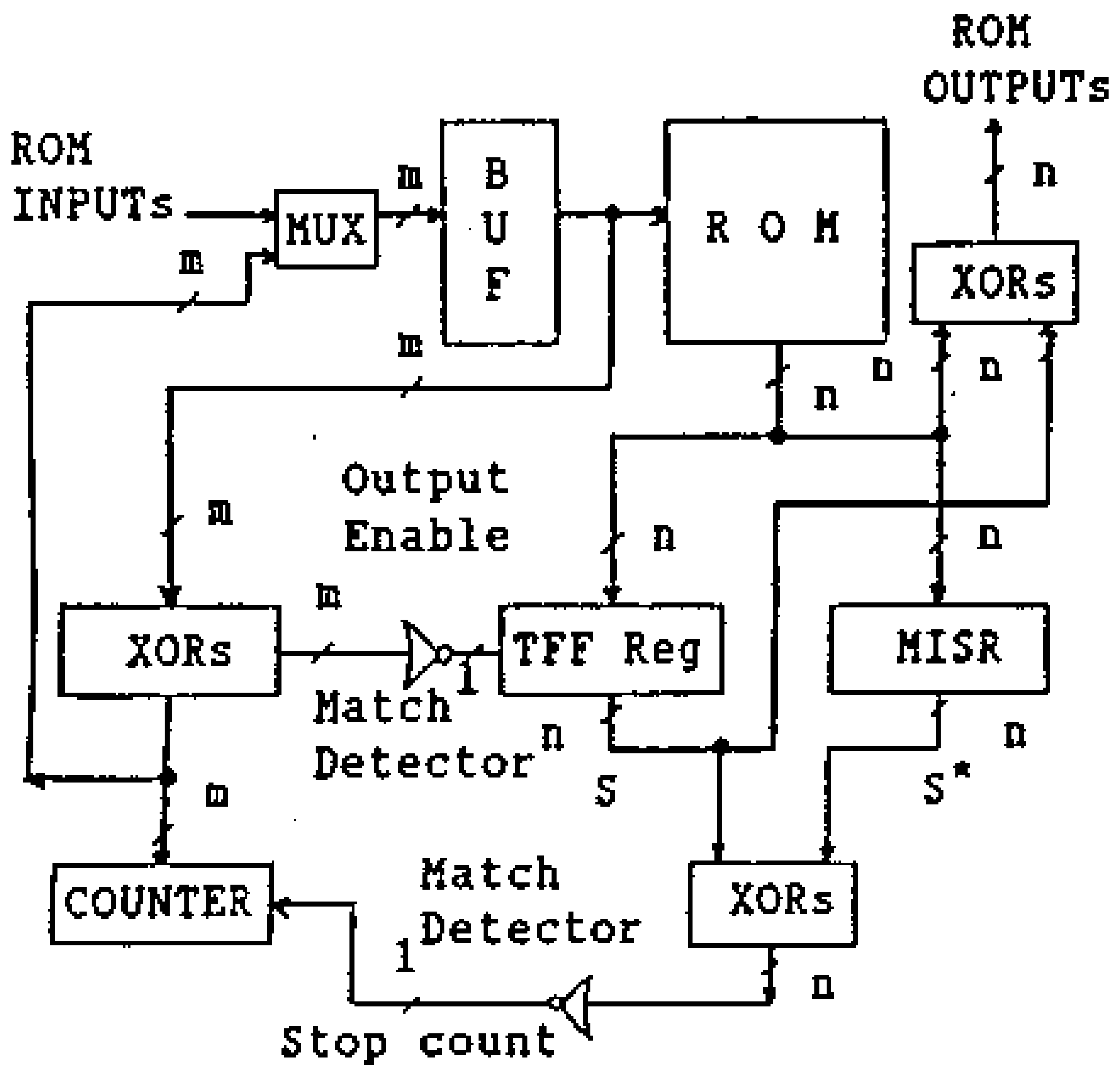


Fig. 3 Built-in Self-Diagnostic ROM

### 3.1 Test Procedure

Fig. 3 shows the block diagram of the built-in self-diagnostic ROM where the bold lines and regular lines indicate the components involved in the normal operating mode and the test/diagnostic mode, respectively. In the test mode the counter generates all $2^m$ input patterns starting from the address zero up to the address $2^m-1$. For each input applied the content of the corresponding address is parallelly loaded into T flip-flops register (TFF Reg.) and simultaneously into MISR (parallel-input LFSR) which is shifted to its next state. This procedure is repeated for the next input pattern until all $2^m$ contents of the ROM are read.

### 3.2 Finite-Field Algebra and MISRs (LFSRs)

Prior to the discussion of the diagnostic scheme, a finite-field algebra of a MISR is presented. An example of a MISR and the corresponding algebra of $GF(2^n)$, the Galois Field of $2^n$ elements, is given below.

Example 1. Consider a primitive parallel-input LFSR (MISR) with feedback polynomial $x^3+x+1$ over $\{0,1\}$ where + denotes modulo-2 sum shown in Fig. 4.

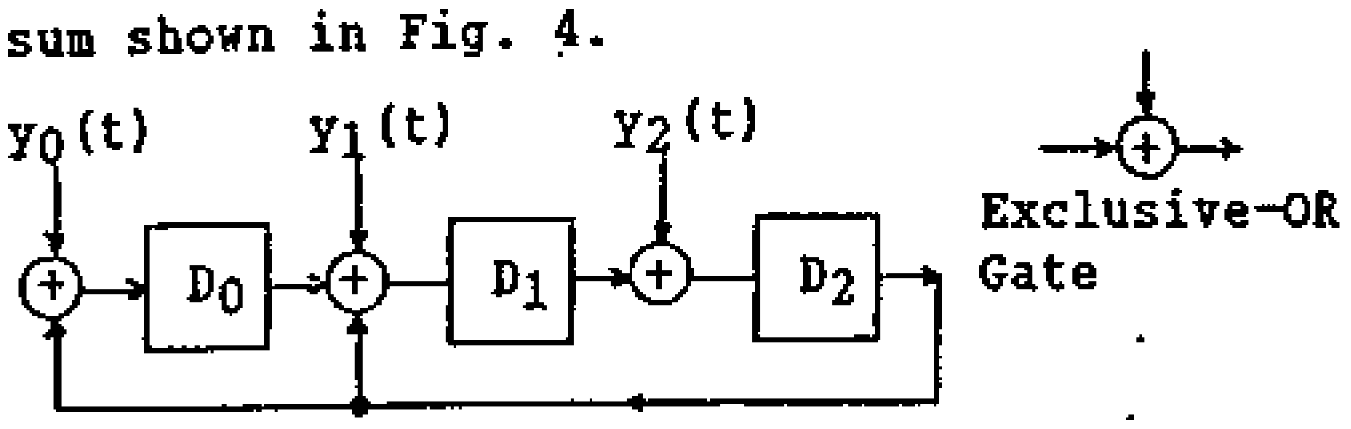


Fig. 4. MISR: Primitive Parallel-input LFSR

An LFSR is called "primitive" if its autonomous (zero input) state transition goes through all $2^n-1$ nonzero states (patterns) for any nonzero initial state. In fact, the nonzero elements of $GF(2^n)$ can be defined in terms of internal states of an n-stage primitive LFSR. In this example, nonzero elements of $GF(2^n)$, n=3, can be constructed using the autonomous LFSR shown in Fig. 4 with $y_0(t)=y_1(t)=y_2(t)=0$, and are given in Table 1 below.

| t \ | s(t) | |
|---|---|---|
| 0 | 1 0 0 | 1 |
| 1 | 0 1 0 | $\alpha$ |
| 2 | 0 0 1 | $\alpha^2$ |
| 3 | 1 1 0 | $\alpha^3$ |
| 4 | 0 1 1 | $\alpha^4$ |
| 5 | 1 1 1 | $\alpha^5$ |
| 6 | 1 0 1 | $\alpha^6$ |

Table 1. Nonzero Elements of $GF(2^3)$ Generated by the Autonomous LFSR with Feedback Polynomial $x^3+x+1$

In Table 1 the states s(t), $t=0,1,\dots,2^n-2$ of the LFSR are listed in terms of the contents of the DFFs. Moreover, the corresponding exponential representation $\alpha^t$, $t=0,1,\dots,2^n-2$, of the nonzero elements are also given where $\alpha$ = (010) is the primitive element, $\alpha^i \neq \alpha^j$, $i \neq j$; $i,j=0,1,\dots,2^n-2$. The addition of $GF(2^n)$ is the componentwise modulo-2 sum, e.g., $\alpha^3+\alpha^5=(110)+(111)=(001)=\alpha^2$. The multiplication is defined as $\alpha^i \cdot \alpha^j=\alpha^{i+j}$, i+j mod $2^n-1$, for example, $\alpha^4 \cdot \alpha^6=\alpha^{10}=\alpha^3$, (10 mod 7 = 3). The vector of all zero's is the element 0. The state transition of an n-stage primitive LFSR can be described by the following first-order linear difference equation over the corresponding $GF(2^n)$,

$$s(t+1) = \alpha s(t),\ t\ge 0,\ s(t),\alpha\in GF(2^n). \qquad (3)$$

A reciprocal LFSR can be defined by the LFSR shifted in the opposite direction. Its state transition is described by

$$s(t+1) = \alpha^{-1}s(t),\ t\ge 0. \qquad (4)$$

Table 2 (below) shows state transitions of the LFSR given in Fig. 4 when shifted in the opposite direction.

| t\ | s(t) | |
|---|---|---|
| 0 | 1 0 0 | 1 |
| 1 | 1 0 1 | $\alpha^6=\alpha^{-1}$ |
| 2 | 1 1 1 | $\alpha^5=\alpha^{-2}$ |
| 3 | 0 1 1 | $\alpha^4=\alpha^{-3}$ |
| 4 | 1 1 0 | $\alpha^3=\alpha^{-4}$ |
| 5 | 0 0 1 | $\alpha^2=\alpha^{-5}$ |
| 6 | 0 1 0 | $\alpha=\alpha^{-6}$ |

Table 2. State Transition of an Autonomous LFSR when running in an opposite direction–Reciprocal LFSR.

The state transition of a primitive parallel-input LFSR (MISR) can also be described by a difference equation over the corresponding $GF(2^n)$ as follows,

$$s(t+1) = \alpha s(t) + y(t),\ t\ge 0, \qquad (5)$$

where $s(0)$ is an initial state and $y(t)\in GF(2^n)$. Table 3 shows an example of the state transition of the parallel-input LFSR given in Fig. 4 with input $y(t)$, $t = 0, 1, 2, 3$ and initial state (0 1 1).

| t\ | y(t) | s(t) |
|---|---|---|
| 0 | $0\ 1\ 0 = \alpha$ | $0\ 1\ 1 = \alpha^4$ |
| 1 | $1\ 0\ 0 = 1$ | $1\ 0\ 1 = \alpha^6$ |
| 2 | $0\ 1\ 1 = \alpha^4$ | $0\ 0\ 0 = 0$ |
| 3 | $1\ 1\ 1 = \alpha^5$ | $0\ 1\ 1 = \alpha^4$ |
| 4 | | $0\ 0\ 0 = 0$ |

Table 3. State Transition of MISR (Parallel-input LFSR)

### 3.3 Diagnostic Procedure

Once all the contents of the ROM are read, the signature S which is the content of TFF register (see Fig. 3) is given by

$$S = S_0 + \bar{y}(0) + \bar{y}(1) + \dots + \bar{y}(2^m-1), \qquad (6)$$

where $S_0$ denotes the initial state, $\bar{y}(t)\in GF(2^n)$, $0\le t\le 2^m-1$, is the possibly distorted content (n-bit word) of the address t, $0\le t\le 2^m-1$, and the addition is defined in $GF(2^n)$. Moreover, $\bar{y}(t)$ can be expressed as the addition in $GF(2^n)$ between fault-free content $y(t)$ and error pattern $e(t)$,

$$\bar{y}(t) = y(t) + e(t),\ 0\le t\le 2^m-1. \qquad (7)$$

For the case that errors only occurs at an address i, $0\le i\le 2^m-1$, that is, $e(t)=0$, $e(t)\in GF(2^n)$, for all $t\neq i$ (single address error) and the initial state $S_0$ of TFF register is

$$S_0 = y(0) + y(1) + \dots + y(2^m-1), \qquad (8)$$

we have, the signature

$$S = e(i), \qquad (9)$$

being the error pattern in the content of address i, $0\le i\le 2^m-1$. Next the address i can be located by the following.

The signature $S^*$, the content of the MISR (see Fig. 3) after $T = 2^m-1$ state transistions, is given by

$$S^* = \alpha^{T+1}S^*_0+\alpha^{T}\bar{y}(0)+\alpha^{T-1}\bar{y}(1)+\dots+\bar{y}(T), \qquad (10)$$

where $S^*_0$ denotes the initial state of the MISR. (We assume that $T\le 2^n-1$ which is often the case). For the case of single address errors and

$$S^*_0 = \alpha^{-1}y(0)+\alpha^{-2}y(1)+\dots+\alpha^{-T-1}y(T), \qquad (11)$$

substituting (11) into (10) with $e(t) = 0$ for all $t\neq i$, we have,

$$S^* = \alpha^{T-i}e(i),\ 0\le i\le T. \qquad (12)$$

The following theorem summarizes the condition of the proposed signature scheme for identifying a single address fault.

Theorem 1. The content of address i contains error pattern $e(i)\neq 0$ if and only if

$$S^* = \alpha^{T-i}S,\ 0\le i\le T, \qquad (13)$$

where $S = e(i)$, (9), is the signature appearing at TFF register and $S^*$ is the signature appearing at the MISR. □

To compute i, $0\le i\le 2^m-1$, based on (13), the MISR is switched to function as the reciprocal autonomous LFSR with the initial state $S^* = \alpha^{T-i}S$, (see Fig. 3). Moreover, for each state transition the content of the reciprocal LFSR is compared with the content of TFF register while the downcounter keeps track of the number of the state transitions. The comparison between the contents of these registers is done by XORs and n-input NOR gate (see Fig. 3). When a match occurs the content of the downcounter indicates the address i. This can be seen by considering (4), $s(t+1) = \alpha^{-1}s(t)$, in which the state of the reciprocal LFSR is equal to S after T-i state transitions.

Example 2. Consider a ROM with two input address lines, 3-bit words and the contents shown in Table 4. For the decoder transistor circuit identical to the one in Fig. 2, the stuck-open fault at the bottom right-hand corner transistor in the decoder array causes a single address error (OR between the contents of address 1 and 3 when the content of address 1 is read). The faulty content is also shown in Table 4.

| Address | Content | faulty Content $\bar{y}(t)$ |
|---|---|---|
| 0 | 0 1 0 | 0 1 0 |
| 1 | 1 0 0 | 1 1 1 ← faulty |
| 2 | 0 1 1 | 0 1 1 |
| 3 | 1 1 1 | 1 1 1 |

Table 4. Content of ROM in Example 2

The initial states for TFF register and MISR (Fig. 4) computed based on (8) and (11) are $s_0$ = (010) and $S^*_0$ = (011), respectively. Note that, $S^*_0$ = (011) can be verified as shown in Table 3. Table 5, below, shows the state transitions of TFF register and MISR.

| t \ $\bar{y}(t)$ | | TFF Reg. | MISR |
|---|---|---|---|
| 0 | 0 1 0 | 0 1 0 | 0 1 1 |
| 1 | 1 1 1 | 0 0 0 | 1 0 1 |
| 2 | 0 1 1 | 1 1 1 | 0 1 1 |
| 3 | 1 1 1 | 1 0 0 | 1 0 0 |
| 4 | | S = 0 1 1 = $\alpha^4$ | $S^*$ = 1 0 1 = $\alpha^6$ |

Table 5. State Transition of Signature Registers

Table 6, below, shows the state transition of the reciprocal LFSR with the initial state equal to $S^*$ = (101) = $\alpha^6$ together with the content of the downcounter.

| Downcounter | Reciprocal LFSR |
|---|---|
| 1 1 | 1 0 1 |
| 1 0 | 1 1 1 |
| Faulty = 0 1<br>Address | 0 1 1 = S = Error |

Table 6. Identification of Faulty Address

## 3.4 Error Correction

After the diagnostic process the magnitude of an error is the content of TFF register (9), and the corresponding error address is the content of the counter (see Fig. 3). In the normal operating mode the ROM input is monitored for a possible match with the error address. When the address appears as the input, the errors are corrected by the componentwise exclusive-ORing between the content of TFF register and the ROM output. This action is controlled by the signal from the match detecter to the output enable of TFF register.

## 3.5 Error-Correcting Code

The diagnostic scheme presented in this section was based on the single error-correcting code (Hamming code) over GF($2^n$) where the check matrix H equals

$$H = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^2 & \cdots & \alpha^T \end{pmatrix} . \qquad (14)$$

## 4. VLSI Implementation

In this section the components in the built-in self-diagnostic circuit (not including the controller) and an estimated area overhead based on the number transistors required in the design are presented.

### 4.1 Reference Signatures $S_0$ and $S^*_0$

The initial states $S_0$ and $S^*_0$ are stored as the two additional words in the ROM array with the word enable signals Int1 and Int2, respectively. The area required is approximated by $L_1$ = 2n (transistors).

### 4.2 The Front-end Circuit

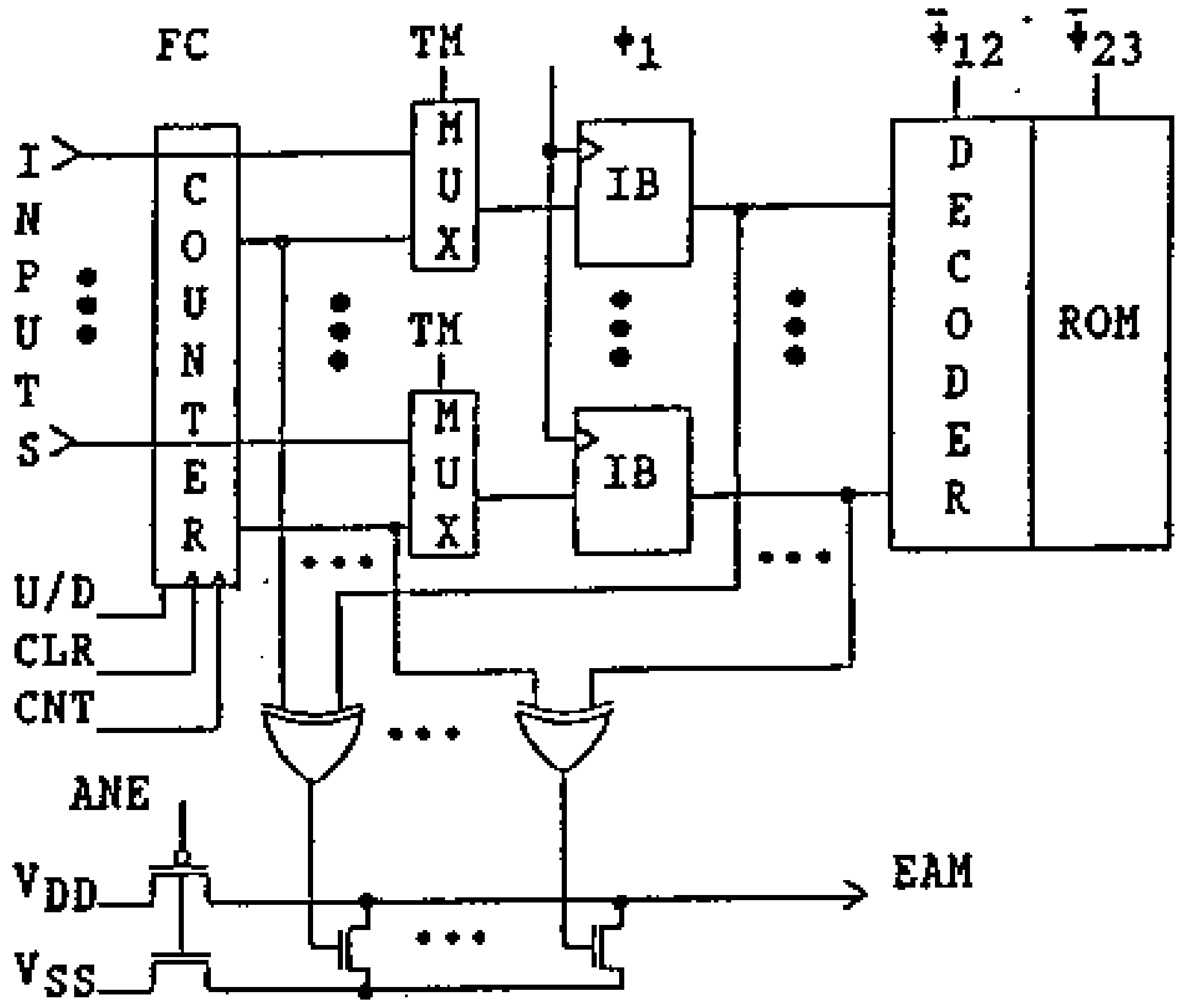


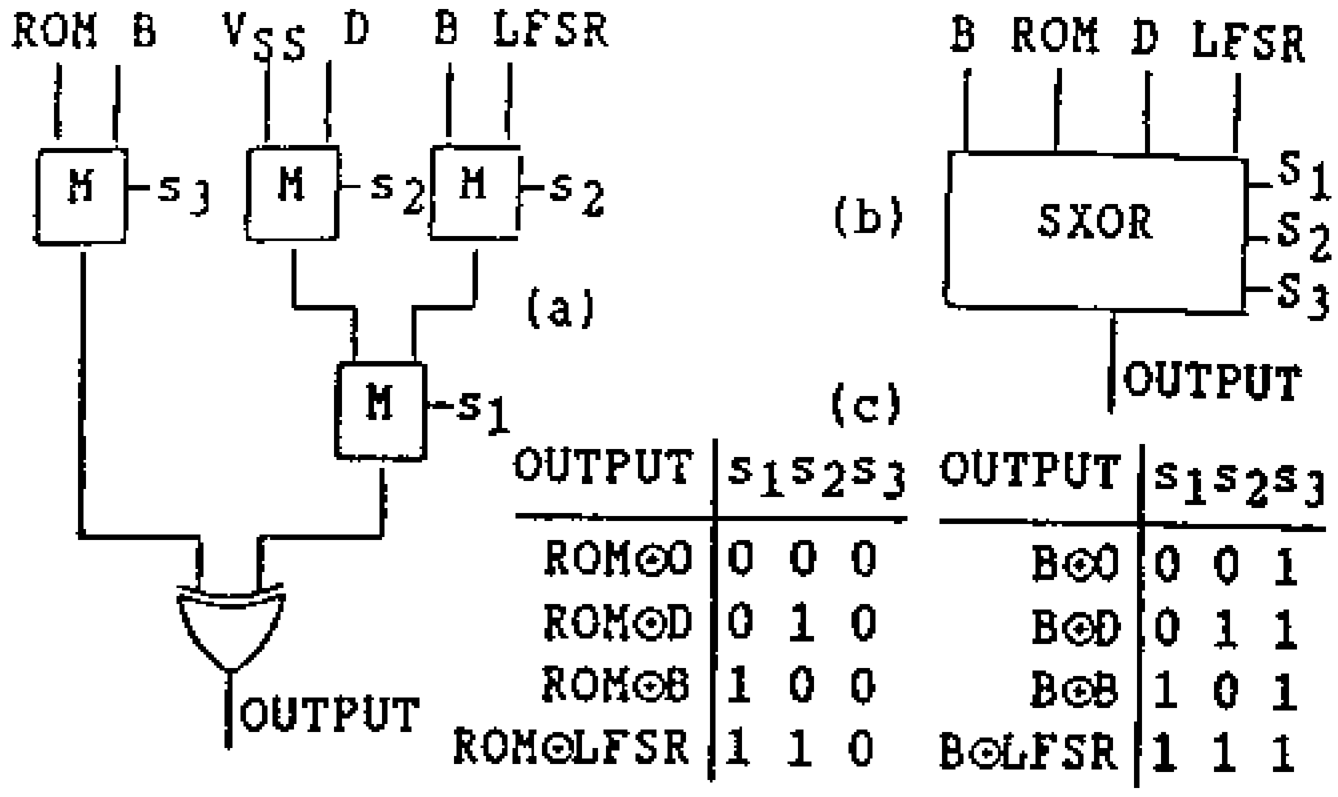
Fig. 5 Front-end Circuit: Test Generation and Error-Address Register

Fig. 5 shows the design of the "Front-end" circuitry which includes the up/down lookahead counter (up-count for test generation, down-count for error location, and parallel register for storing error address), the match detector (XOR-gates and NOR-gate), and data path 2×1 multiplexers with control signal TM=1 for the test mode, and TM=0 for the normal mode. The symbols used in Fig. 5 are given as follows; $\phi_i$'s: clock signals, IB: ROM input buffers, U/D: Up/downcount select, CLR: Clear enable, CNT: Count enable, ANE: Address-NOR evaluate, EAM: Error address match signal, and FC: Full-count signal. The number of transistors used for the CMOS components are listed as follows; 2×1 MUX: 4, XOR-gate: 6, DFF: 14. The up/down lookahead counter consists of m+1 DFFs (the additional DFF for the full-count signal) and m lookahead, up/down select and clear circuit blocks where each block consists of an AND-gate (6 transistors), 3 MUXs and a NAND-gate (4 transistors). An estimated area in terms of the number $L_2$ of transistors required to implement the front-end block is

$$L_2 = \underbrace{14(m+1)+22m}_{\text{counter}} + \underbrace{4m}_{\text{MUXs}} + \underbrace{6m}_{\text{XORs}} + \underbrace{m+2}_{\text{NORs}} + \underbrace{2m}_{\text{Wires}} = 49m + 16. \qquad (15)$$

### 4.3 Test/Diagnostic and Error-correcting Circuit

Fig. 6 shows a design of the test/diagnostic circuit for n = 3, where the storage cells of TFF register is implemented using the ROM output buffer ($B_i$=1,...,n) and the storage cells of the MISR, the reciprocal LFSR and the error register are implemented by the DFFs ($D_i$, i=1,...,n). These registers use the common XOR-gates where the selects of the operands is done by the control signals $s_1$, $s_2$ and $s_3$ of the block "SXOR." (The design of the SXOR and the selects of the output function are shown in Fig. 7). The symbols used in Fig. 6 are given as follows; ROM: the output lines from the ROM array, M: 2×1 MUX, F/R: Select signal for MISR (Bold lines) and the reciprocal LFSR (dashed lines), LD: DFF resgister load enable, LB: Buffer Load enable, NE: NOR-gate evaluate enable, SC: Stop-count signal, and FB: Feedback loop breaker.



| OUTPUT | $s_1s_2s_3$ | OUTPUT | $s_1s_2s_3$ |
|---|---|---|---|
| ROM⊕0 | 0 0 0 | B⊕0 | 0 0 1 |
| ROM⊕D | 0 1 0 | B⊕D | 0 1 1 |
| ROM⊕B | 1 0 0 | B⊕B | 1 0 1 |
| ROM⊕LFSR | 1 1 0 | B⊕LFSR | 1 1 1 |

Fig. 7. SXOR: (a) Circuit, (b) Block Diagram and (c) Select Functions

An estimated area in terms of the number $L_3$ of transistors required for the test, diagnostic and error-correcting circuit is given by

$$L_3 = \underbrace{14n}_{\text{DFFs}} + \underbrace{4n}_{\text{MUXs}} + \underbrace{22n}_{\text{SXOR}} + \underbrace{(n+2)}_{\text{NOR}} + \underbrace{5n}_{\text{Wires}} + 10W = 46n + 10W + 2, \qquad (16)$$

where W is the number of internal feedback taps, e.g., W=1 for trinomial feedback polynomial.

### 4.4 Control Unit

The functions of the control unit for the test/diagnostic and error-correcting procedures based on the circuits given in Fig. 5 and 6 are described on the register transfer language level.

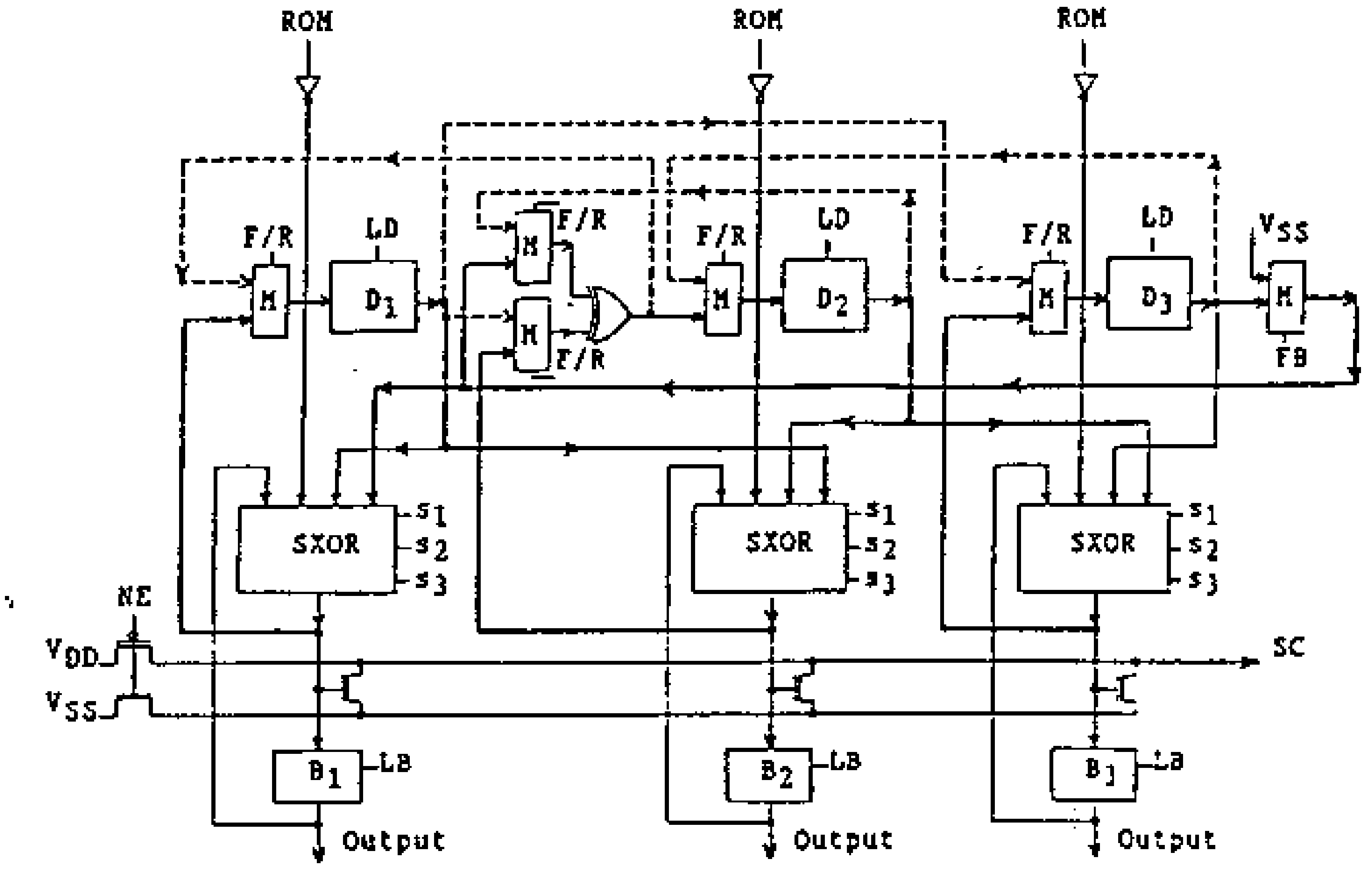


Fig. 6 Test/Diagnostic and Error-Corecting Circuit (n=3)

```
1. Load Initial states
  1.1 load $S_0$ into TFF reg
  1.2 load $S^*_0$ into DFF reg.
2. Test Application
  2.1 TM=1, clear counter, set up-count
  2.2 latch test pattern
  2.3 B ← ROM$\oplus$B (implement TFF Reg)
  2.4 D ← ROM$\oplus$LFSR (implement MISR)
  2.5 if Full-Count(FC)=1 go to 3
      else count (next test patterns)
      and go to 2.2 (loop).
3. Error Dectection
  3.1 select B$\oplus$0 and evaluate NOR-gate
  3.2 if B=0 select B$\oplus$D and evaluate
      NOR-gate else go to 4
  3.3 if D=0 go to 6 (ROM fault-free)
      else go to 7 (ROM inoperative)
4. Error Location
  4.1 select downcount and set counter
  4.2 select B$\oplus$D and evaluate NOR-gate
  4.3 if Stop-Count(SC)=1 (faulty address
      located), go to 5 (error correcting);
      if FC=1 go to 7 (ROM inoperaive);
      else shift reciprocal LFSR, count
      and go to 4.2 (loop).
5. Error Correcting
  5.1 operative status to system and
      Test-Mode(TM)=0
  5.2 compare ROM input with error address
      if ROM input matches DFF Reg. content
      then correct errors by selecting
      ROM$\oplus$D else select ROM$\oplus$0
6. Fault-Free; operative status to system,
   TM=0, and select ROM$\oplus$0.
7. Inoperative status to system.
```

The control unit is a sequential circuit (e.g., using counter and PLA). There are 8 input signals (including system test-mode and clock signals), 18 output control signals (including clock signals) and approximately 20 internal states. An estimated number $L_3$ of transistors required for the controller (with 50 transistors per next-state logic functions and outputs) is given by

$$L_3 = \underset{\text{State storage}}{14\times5} + \underset{\text{Next-state logic}}{50\times5} + \underset{\text{Outputs}}{50\times18} + \underset{\text{wires}}{750} \cong 2,000. \qquad (17)$$

4.5 The Percentage Area Overhead

An estimated area $L_0$ of the original ROM is given by

$$L_0 = \underset{\text{Decoder}}{[2m+1.5]2^m} + \underset{\text{ROM}}{[2^m+1.5]n} + \underset{\text{Buffers}}{14(n+m)} + \underset{\text{Inv}}{2(n+m)} \qquad (18)$$
$$= (2m+n+1.5)2^m+16.5n+16m.$$

An estimated area of the additional circuit is given by

$$L_X = L_1+L_2+L_3+L_4 = 49m+48n+10W+2014, \qquad (19)$$

(see Sec. 4.1 and Eq (15), (16) and (17)).

For the case of ROMs with $(m,n) = (9,60)$, $(10,20)$ and $(13,40)$, the estimated percentage area overhead is 13%, 8%, and 0.8%, respectively. (The number $W$ of internal feedback taps is assumed to be ten). The proposed built-in self-diagnostic design is more efficient as the size of the ROM increases since the original ROM area grows exponentially with $m$, (18), whereas the additional circuit grows linearly with $m$ and $n$, (19).

## 5. Fault Detection

The proposed technique detects all single address errors and uses an n-bit signature scheme. With the typical sizes of the control ROMs, fault simulation can be performed in an affordable amount time. If some of the stuck-at faults at the inputs, bit-lines, output-lines and the word-line stuck-at 1 faults escaped, an additional word can be stored in an unused address of the ROM such that the faults previously escaped are detected [21], therefore, 100% fault coverage is easily attainable.

## 6. Conclusions

The switch-level fault analysis of the ROMs showed that most of the single stuck-open/on faults (on the order of 95% or greater) will result in errors confined in the content of a single address. The presented design of built-in self-diagnostic ROMs based on exhaustive testing and the signature scheme by the Hamming codes over GF($2^n$) identifies these errors and their corresponding addresses in the off-line test mode. In the ROM normal operating mode these errors are corrected as the content of the address is read and, thus, the ROM will continue to perform its correct function despite the presence of faults. The VLSI implementation was presented where the estimated silicon area overhead was shown be on the order of 15% or less for typical microcode ROM sizes.

## 7. References

[1] Y. Zorian and A. Ivanov, "EEODM: An efficient BIST scheme for ROMs," Proc. Int. Test Conf. 1990.

[2] K.K. Saluja, S.H. Sng, and Kinoshita, "Built-in self-testing RAM: A practical alternative," IEEE Design & Test, pp. 42-51, Feb. 1987.

[3] S.K. Jain and C.H. Stroud, "Built-in self-testing of embedded memories," IEEE Design & Test, pp. 27-37, Oct. 1986.

[4] R. Treuer, H. Fujiwara and V.K. Agarwal "Implementing a built-in self-test PLA design," IEEE Design & Test, pp. 37-48, Apr. 1985.

[5] P.H. Bardell, W.M. McAnney and J. Savir, Built-in Test for VLSI: Pseudorandom Techniques, New York: Wiley Interscience, 1987.

[6] D.K. Pradhan (Editor), Fault-Tolerant Computing: Theory and Techniques, Vol. 1, Englewood Cliffs, NJ: Prentice-Hall, 1986.

[7] T.W. Williams and K.P. Parker, "Design for testability—A survey," IEEE Trans. on Comput., Vol. C-31, No. 1, pp 2-15, Jan. 1982.

[8] S.R. Ready, K.K. Saluja and M.G. Karpovsky, "A data compression technique for test responses," IEEE Trans. on Comput., Vol. C-38, pp. 1151-1156, Sep. 1988.

[9] J.P. Robinson and N.R. Saxena, "A Unified view of test compression methods," IEEE Trans. on Comput., Vol. C-35, Apr. 1986.

[10] M.G. Karpovsky and P. Nagvajara, "Optimal robust compression of test responses," IEEE Trans. on Comput., Vol. 39, No. 1, pp. 138-141, Jan. 1990.

[11] D.K. Pradhan, S.K. Gupta and M.G. Karpovsky, "Aliasing probability for multiple input signature analyzer," IEEE Trans. on Comput., Sep. 1989.

[12] D.P. Siewiorek and R.S. Swarz, The Theory and Practice of Reliable System Design, Bedford, MA: Digitat Press, 1982.

[13] D.C. Bossen, "b-Adjacent error correction," IBM J. Res. Dev., Vol. 14 pp. 402-408, July 1970.

[14] M.G. Karpovsky and P. Nagvajara, "Design of built-in self-diagnostic boards by signature analysis," IEEE Trans. on Indust. Electronics., Vol. 36, No. 2, pp. 241-245, May 1989.

[15] J.Y. Jou and J.A. Abraham, "Fault-tolerant matrix arithmetic and signal processing on highly concurrent computing structures," Proc. IEEE, pp. 732-741, Mar. 1986.

[16] R.L. Wadsack, "Fault modeling and logic simulation of CMOS and MOS integrated circuits," Bell System Tech. J., May-June 1978, pp. 1449-1473.

[17] Y.K. Malaiya and S.Y. Su, "A new fault model and testing technique for CMOS devices," Proc. IEEE Int. Test Conf., 1982, pp. 25-34.

[18] D.L. Liu and E.J. McCluskey, "Designing CMOS logic circuits for switch-level testability," IEEE Design & Test, pp. 42-49, Aug. 1987.

[19] N. Weste and K. Eshraghian, Principle of CMOS VLSI Design: A System Perspective, Reading,MA: Addison-Wesley, 1985.

[20] L.A. Glasser and D.W. Dobberpuhl, The Design and Analysis of VLSI Circuit, Reading, MA: Addison-Wesley, 1985.

[21] S.Z. Hassan and E.J. McCluskey, "Testing PLA's using multiple parallel signature analysis," Dig. Int. Symp. Fault-Tolerant Comput., pp. 422-425, 1983.

[22] S.Y. Kung, VLSI Array Processors, Englewood Cliffs, NJ: 1988.