Brief Contributions

Easily Testable Multiple-Valued Logic Circuits Derived from Reed-Muller Circuits

E.V. Dubrova, *Member*, *IEEE*, and
J.C. Muzio, *Member*, *IEEE*

Abstract—In 1972, Reddy showed that the binary circuits realizing Reed-Muller canonical form are easily testable. In this paper, we extend Reddy's result to multiple-valued logic circuits, employing more than two discrete levels of signal. The electronic fabrication of such circuits became feasible due to the recent advances in integrated circuit technology. We show that, in the multiple-valued case, several new phenomena occur which allow us to asymptotically reduce the upper bound on the number of tests required for fault detection, but make the generation of tests harder.

Index Terms—Multiple-valued function, Reed-Muller circuit, easily testable circuit, stuck-at fault.

1 Introduction

In recent years, there have been major advances in integrated circuit technology which have both made feasible and generated great interest in electronic circuits which employ more than two discrete levels of signal. Such circuits, called *multiple-valued logic circuits*, offer several potential opportunities for the improvement of present very-large scale integrated (VLSI) circuit designs. Serious difficulties with limitations on the number of connections of an integrated circuit with the external world (pinout problem), as well as on the number of connections inside the circuit encountered in some VLSI circuit synthesis, could be substantially reduced if signals in the circuit are allowed to assume three or more states rather than only two. Apart from this reduction in the interconnection problem, applying multiple-valued logic to logic design has also been shown to allow enhancing circuit performance in terms of chip area [4], [5], [6], [7], operation speed [12], and power consumption [10].

In memory design, a major achievement is Intel's 64-Mbit Strataflash flash memory device with multilevel storage capability, announced in 1997 [2]. Each memory cell consists of a single NOR transistor, implemented using 0.45-micron technology. Two bits of information are stored in a cell by charging the polysilicon floating gate of a transistor to four different levels. As a result, cell area is reduced and the unit cost is dropped to 47 cents per megabit, which is the lowest cost on the flash memory market at present.

Intel's announcement was followed by Mitsubishi's and SGS-Thomson's announcement on development of a 64-Mbit multilevel cell flash memory that will compete with Intel's device [3]. The production is expected to begin in the second half of 1999. This new 64-Mbit flash memory will aim to incorporate DiNOR's inherent speed and NOR's ability to scale down in voltage. The first device is expected to operate using a 3-volt power supply and achieve access speeds of 100 to 125 nanoseconds. SGS-Thomson and Mitsubishi are hoping to compete with Intel by using a 0.20-micron process technology and then migrating to 0.18-micron technology.

In logic design, among the recent relevant achievements are the multiple-valued full adder [8], multiplier [9], [10], counter [11], and A/D converter [12]. The annual *IEEE Proceedings of the International Symposium on Multiple-Valued Logic* provides the authoritative ongoing record of contributions in this area.

In this paper, we show that, besides enhancing circuit performance in terms of chip area, operation speed, and power consumption, multiple-valued logic circuits might also offer the benefit of easier testability. We study logic circuits realizing a modulo m sum-of-products canonical form of multiple-valued functions with m being a prime. So far, the applications of this canonical form to logic design have only been considered for the case $m = 2$. The circuits realizing modulo 2 sum-of-products canonical forms, usually called *Reed-Muller canonical circuits*, are known to be easily testable [13], [14], [15]. The upper bound on the number of tests required to detect all single stuck-at faults in such circuits is proven to be $3n + 4$, where n is the number of primary inputs [13]. In this paper, we investigate the case of $m > 2$, with m being a prime. We show that the upper bound on the number of tests required to detect all single stuck-at faults in a circuit realizing a function in modulo m sum-of-products canonical form is $2n + 4$. Generalizing from the two- to the m-valued case, however, is a nontrivial problem because, for $m > 2$, several new phenomena occur which allow us to reduce the upper bound on the number of tests required for fault detection, but make the generation of tests harder. We also show that, by adding to the circuit an extra multiplication modulo m gate with an observable output, the number of tests required to detect all single stuck-at faults in the circuit is reducible to four universal tests. Such a technique has been applied in the binary case as well [13], but then the number of tests can only be reduced to $n + 4$, which is *asymptotically* worse as compared to the multiple-valued case.

The paper is organized as follows: Section 2 gives the background for the paper. In Section 3, the upper bound on the number of tests needed to detect all single stuck-at faults in a circuit realizing a function in modulo m sum-of-products canonical form is derived. Section 4 shows that, by adding an extra multiplication modulo m gate with an observable output to the circuit, the number of tests needed to detect all single stuck-at faults can be reduced to four. Section 5 concludes the paper.

2 Preliminaries

We use the standard notation adopted in the areas of multiple-valued logic and testing. For a more detailed description, the reader is referred to [16] and [17].

2.1 Modulo m Sum-of-Products Form

A *multiple-valued logic function* $f(x_1, \ldots, x_n)$ is a mapping $f : M^n \rightarrow M$, where $M := \{0, 1, \ldots, m-1\}$ is a totally ordered set and M^n denotes the Cartesian product $M \times M \times \ldots \times M$ of n sets M. We say that $f(x_1, \ldots, x_n)$ is an n-variable m-valued function.

Cohn [19] has shown that, if m is a prime, then any n-variable m-valued function has a unique modulo m representation of the type:

$$f(x_1, \ldots, x_n) = \sum_{i=0}^{m^n-1} c_i \cdot x_1^{i_1} \cdot x_2^{i_2} \cdot \ldots \cdot x_n^{i_n}, \quad (1)$$

- *E.V. Dubrova is with the Electronic System Design Lab., Department of Electronics, Royal Institute of Technology, KTH-Electrum, Electrum 229, S-164 40 Kista, Sweden. E-mail: elena@ele.kth.se.*
- *J.C. Muzio is with the VLSI Design and Test Group, Department of Computer Science, University of Victoria, PO Box 3055, Victoria, BC, Canada V8W 3P6. E-mail: jmuzio@csr.uvic.ca.*

Manuscript received 29 June 1998; revised 27 Aug. 1999; accepted 14 Sept. 2000.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 107088.

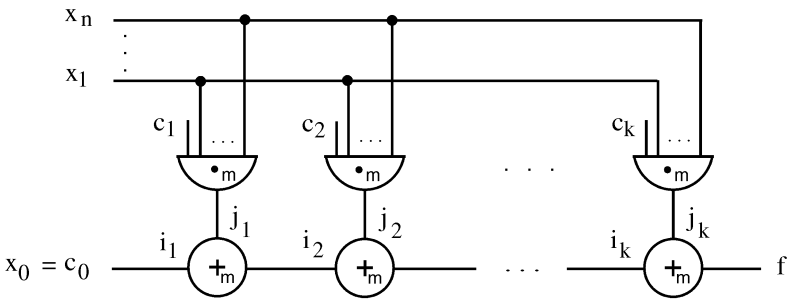

Fig. 1. Logic circuit scheme realizing the modulo m sum-of-products form.

where $c_i \in M$ are constants, "$\cdot$" stands for multiplication modulo m, and $\sum$ stands for addition modulo m. $(i_1 i_2 \dots i_n)$ is the m-ary expansion of i with i_1 being the least significant digit and the term $x_j^{i_j}$ denotes the i_jth power of the variable x_j, $j \in \{1, \dots, n\}$. Modulo m addition and multiplication form a Galois Field of order m, $GF(m)$. Modulo m sum-of-products forms are polynomials over $GF(m)$.

A modulo m sum-of-products form can be implemented by a logic circuit of the type shown in Fig. 1. It consists of a linear cascade of two-input addition modulo m gates fed by multiplication modulo m gates, one corresponding to each product term of the expansion (1) with a nonzero constant c_i, $i \in \{1, \dots, m^n - 1\}$. The input x_0 has the value of the constant c_0 during normal operation and a value different from c_0 during testing.

For example, the 3-variable 3-valued function

$$f(x_1, x_2, x_3) = 1x_1^2 x_2 \oplus 2x_1^2 x_2^2 \oplus 1x_1 x_2 x_3 \oplus 2x_1 x_2 x_3^2$$

can be implemented by the circuit shown in Fig. 2. The constant 1 is an identity element with respect to multiplication and can therefore be omitted.

In the two-valued case, the expansion (1) reduces to the *zero polarity Reed-Muller canonical form* [20], [21]. For $m = 2$, the addition modulo 2 is equivalent to the XOR and the multiplication modulo 2 is equivalent to the AND. The Reed-Muller canonical form often yields compact representations for the functions which are "hard" for the conventional sum-of-product expansion over AND, OR, and NOT. It is shown to provide a suitable basis for the implementation of some practical Boolean functions with embedded XOR-logic [22]. The odd-parity function is an example of such a function. On one hand, it needs 2^{n-1} products of n variables, each to be written as a sum-of-product expansion over AND, OR, and NOT. On the other hand, it can be expanded in the Reed-Muller canonical form using only n products of one variable each, namely $x_1 \oplus x_2 \oplus \dots \oplus x_n$.

2.2 Test Generation for Logic Circuits

A *fault* in an electrical circuit is a physical defect of one or more components, which can cause the circuit to malfunction. Many physical faults in electrical circuits can be modeled by a *stuck-at fault* logic model. In this kind of fault-model, it is assumed that any physical fault (such as, for example, short or open diode, broken wire between gates, etc.) can be modeled by a number of lines in the corresponding logic circuit permanently fixed at some logic level 0, 1, ..., or $m - 1$. In this paper, we use the *single stuck-at fault* logic model, assuming that a single line in the circuit is fixed at a logic level 0, 1, ..., or $m - 1$.

Any *n*-tuple $(a_1, \dots, a_n) \in M^n$ of values of the input variables $(x_1, \dots, x_n)$ is called *test* for a fault α if and only if

$$f(a_1, \dots, a_n) \neq f^\alpha(a_1, \dots, a_n),$$

where $f^\alpha(a_1, \dots, a_n)$ denotes the function describing the circuit in the presence of the fault α. Such a test $(a_1, \dots, a_n)$ is said to *detect* the fault α.

In the traditional methods for fault detection, tests are applied to the circuit under test and the output responses are verified one by one. Any discrepancy detects a fault. A set T of input vectors is called a *test set* for some set of faults $\mathcal{F}$ if the observation of the corresponding outputs allows the detection of every fault from $\mathcal{F}$ in the circuit.

If the circuit is not redundant, the set of m^n possible input vectors is a test set of the circuit. However, if the circuit implementation is known, it is possible to construct test sets having less than m^n elements. One of the objectives of testing is to construct minimal test sets. A test set T is called *minimal* if no other test set is properly included in T. The search for a minimal test set is usually carried out in two steps:

1. For each of the possible faults from $\mathcal{F}$, generate the corresponding tests.
2. Find a minimal cover of the faults by a set of test vectors.

It is proven by Reddy [13] that, for an *n*-input binary logic circuit implementing Reed-Muller canonical form, there exists a minimal test set for all single stuck-at faults of a maximum size

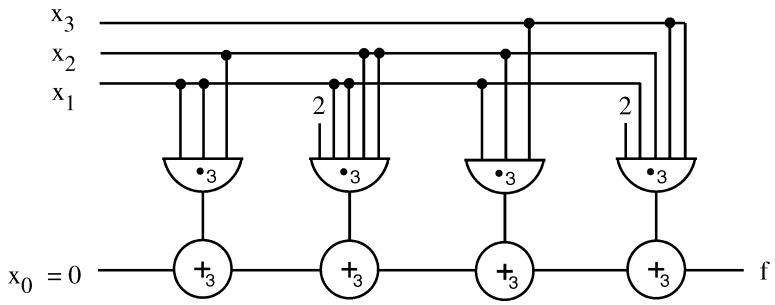

Fig. 2. Modulo m sum-of-products circuit realization of a function $f(x_1, x_2, x_3)$.

$3n+4$. Sasao [18] has extended Reddy's result for the case of binary Generalized Reed-Muller circuits, in which the input variables are allowed to appear in both complemented and non-complemented forms. In this paper, we extend Reddy's result for the multiple-valued case. First, we consider faults which occur on the primary inputs of the circuit.

3 Primary Inputs

In this section, we show that the number of tests which are needed to detect all stuck-at faults on primary inputs in a circuit realizing a modulo m sum-of-products form, as well as in an arbitrary m-valued combinational logic circuit realizing a function $f(x_1,\ldots,x_n)$, is at most $2n$.

If a function $f(x_1,\ldots,x_n)$ is not degenerate in x_i, then we can always find an assignment $(a_1,\ldots,a_{i-1},a_{i+1},\ldots,a_n)$ of values for the variables $(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n)$ and two values a_i and b_i, $a_i \neq b_i$, for x_i such that

$$f(a_1,\ldots,a_{i-1},a_i,a_{i+1},\ldots,a_n) \neq f(a_1,\ldots,a_{i-1},b_i,a_{i+1},\ldots,a_n).$$

Since the change in the value of x_i from a_i to b_i causes a change in the value of f, the input vector $t_1=(a_1,\ldots,a_{i-1},a_i,a_{i+1},\ldots,a_n)$ is a test for all x_i stuck-at faults which set the output of the circuit to a logic value different from $f(a_1,\ldots,a_{i-1},a_i,a_{i+1},\ldots,a_n)$. On the other hand, the input vector $t_2=(a_1,\ldots,a_{i-1},b_i,a_{i+1},\ldots,a_n)$ is a test for all x_i stuck-at faults which set the output of the circuit to $f(a_1,\ldots,a_{i-1},a_i,a_{i+1},\ldots,a_n)$. Since $a_i \neq b_i$, these two cases cover all m single stuck-at faults on x_i and, therefore, $T=\{t_1,t_2\}$ is a test set for all single stuck-at faults on x_i. Repeating for n inputs, a set of $2n$ tests for all single stuck-at faults on primary inputs of an arbitrary m-valued combinational logic circuit, $m>1$, can be derived.

For $m=2$, the upper bound $2n$ can be reduced to a tighter upper bound $2n_e$, where n_e is the number of primary inputs appearing in an even number of product terms in the Reed-Muller canonical form of the n-variable Boolean function being realized [13]. The following procedure is used to find the test set. For a primary input x_i, all AND gates having x_i as input are considered. From these, a gate G_i with the minimal number of other inputs is selected. Further, two tests, t_{i1} and t_{i2} are defined in the following way:

- $t_{i1}: x_i=0$, all other inputs of G_i are 1 and all other primary inputs are 0;
- $t_{i2}: x_i=1$, all other inputs of G_i are 1, and all other primary inputs are 0.

The test t_{i1} detects x_i stuck-at-1 fault and the test t_{i2} detects x_i stuck-at-0 faults. The procedure is repeated for all n_e inputs.

Unfortunately, this simple procedure cannot be used in the multiple-valued case for the following reason: In a modulo m sum-of-products form, a variable x_i can appear with $(m-1)$ different powers. For $m=2$, this makes $m-1=1$. Therefore, for any x_i, a *single* gate G_i having x_i as its input and depending on the minimal number of other inputs can be selected. By assigning all but x_i inputs of G_i to value 1, and all other primary inputs to 0, a *single* path from x_i to the output is sensitized. So, the effect of a fault on x_i can always be propagated to the output. On the other hand, if $m>2$, then $m-1>1$. Therefore, there may be more than one multiplication modulo m gate in the circuit, depending on x_i and k other primary inputs. If these k primary inputs are assigned to 1 and the rest of the primary inputs to 0, then the effect of a fault on x_i is propagated along multiple paths and thus may be canceled out by the addition modulo m cascade. For example, consider the circuit shown in Fig. 2 and suppose we generate tests for the primary input x_1. All four multiplication modulo m gates have x_1 as input, but the first and the second gates depend on the minimal number of other primary inputs (x_2 only). If we set $x_2=1$ and $x_3=0$, then the circuit implements the function $f(x_1,1,0)=1x_1^2 \oplus 2x_1^2=0$, i.e., the output is not sensitive to x_1. However, for the input assignment $x_2=2$ and $x_3=0$, the circuit implements the function $f(x_1,2,0)=2x_1^2 \oplus 2x_1^2=1x_1^2$ and, thus, the output is sensitive to x_1.

Hence, in the case of $m>2$, all m^k possible combinations of values for k primary inputs (not assigned to 0) should be examined to find out which one of them makes the output sensitive to x_i. Such an assignment always exists, provided the circuit doesn't have redundant multiplication modulo m gates. The smaller the value of k, the easier it is to find the test for x_i. In [1], we proved that, for a random circuit implementing an m-valued n-variable function in modulo m sum-of-products form, the probability that $k\leq 1$ is greater than 99.99 percent for any x_i, provided $n\geq 3$ and $m\geq 3$. We have also described in [1] a procedure for test generation for primary inputs, handling the specifics of the m-valued case.

In the next section, we consider the detection of internal faults, which, in the context of our circuit structure, consists of all faults other than those considered above on the primary input lines. This means that all of the inputs to the individual gates comprise the set of internal faults that we consider.

4 Internal Lines

It is proven in [13] that, in the binary Reed-Muller circuit realization of an n-variable Boolean function, at most $n+4$ tests are required to detect all internal single stuck-at faults. The proof is constructive by showing that, independently of the function being realized, a set $T=T_1\cup T_2$ detects all internal single stuck-at faults. T_1, consisting of four tests, is defined by the following table:

x_0	x_1	x_2	...	x_n
0	0	0	...	0
0	1	1	...	1
1	0	0	...	0
1	1	1	...	1

It detects all single faults on the inputs of XOR gates and all stuck-at-0 faults on the inputs of AND gates. T_2 is defined by $T_2:=\{t_{21},t_{22},\ldots,t_{2n}\}$ with the test t_{2i} having $x_i=0$ and $x_j=1$ for all $i\neq j$, where $i,j\in\{1,\ldots,n\}$. It detects all stuck-at-1 faults on the inputs of AND gates. So, the $n+4$ tests in the test set $T=T_1\cup T_2$ detect all internal single stuck-at faults in a binary Reed-Muller circuit. It is also shown in [13] that, for XOR gates, these $n+4$ tests detect not only single stuck-at faults, but also other faults, changing the functionality of the XOR gate to any the 15 other two-variable functions.

We use a similar approach to prove that, in the modulo m sum-of-products circuit realization of an m-valued function, four tests are required to detect all internal single stuck-at faults. It might appear surprising that the multiple-valued case requires fewer tests than the two-valued one. Before giving the result, we explain the intuition behind this phenomenon.

Consider an n-input multiplication modulo m gate G with m being a prime. Let $a_i\in M$ be the value of the input variable x_i, for $i\in\{1,\ldots,n\}$. Since the cancellation law of multiplication holds for $GF(m)$ [23], for any $a,b,c\in M$, it holds that if $a\neq 0$ and $b\neq c$, then $ab\neq ac$.

Hence, if an input vector, $(a_1,\ldots,a_n)$ such that $a_i\neq 0$ for all i, is applied to G, then a change in the value of any single input x_i causes a change in the value on the output. This implies that $(a_1,\ldots,a_n)$ is a test for all x_i stuck-at-$\overline{a}_i$ faults, where $\overline{a}_i$ denotes any value but a_i, i.e., $\overline{a}_i\in M-\{a_i\}$.

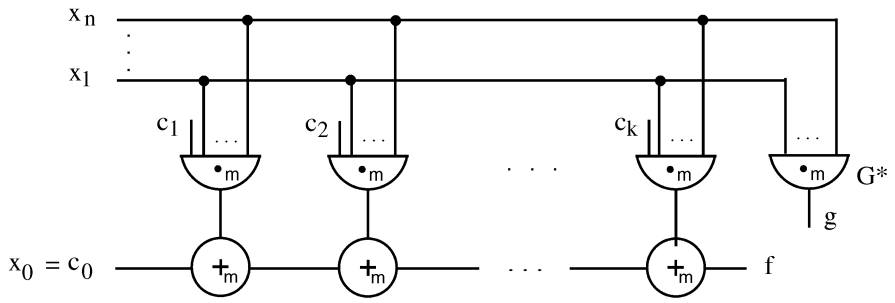

Fig. 3. Logic circuit with an extra multiplication modulo m gate G^*.

Similarly, to detect the remaining stuck-at-a_i faults on each input x_i, another input assignment $(b_1, \ldots, b_n)$ such that $b_i \neq 0$ and $a_i \neq b_i$ for all i has to be applied to G.

Hence, any two input assignments $(a_1, \ldots, a_n)$ and $(b_1, \ldots, b_n)$ such that $a_i, b_i \neq 0$ and $a_i \neq b_i$ for all $i \in \{1, \ldots, n\}$, detect all single stuck-at faults on the inputs of a multiplication modulo m gate for m being a prime greater than two. It is easy to see why $m = 2$ is an exception. In the two-valued case, there exists only one input assignment with all entries different from zero, namely $(11 \ldots 1)$. Therefore, an *n*-input multiplication modulo 2 gate (AND gate) cannot be tested for all single stuck-at faults with only two input vectors.

Since the cancellation law of addition also holds for $GF(m)$, we can similarly show that any two input assignments $(a_1, \ldots, a_n)$ and $(b_1, \ldots, b_n)$ such that $a_i \neq b_i$ for all $i \in \{1, \ldots, n\}$ detect all single stuck-at faults on the inputs of an *n*-input addition modulo m gate.

Now, we give the main result of the section.

Theorem 1. *There exists a universal set of four tests which detects all single stuck-at faults on internal lines in the modulo m sum-of-products circuit realization of an m-valued function, with m being a prime greater than two.*

Proof. The proof is constructive. Consider the set T consisting of four tests defined by the following table:

x_0	x_1	x_2	...	x_n
0	0	0	...	0
0	1	1	...	1
1	0	0	...	0
0	$m-1$	$m-1$	...	$m-1$

Let us denote by i_k and j_k the inputs of the *k*th addition modulo m gate in the cascade, as shown in Fig. 1.

1. The first test of T results in applying $(0, 0)$ to each pair (i_k, j_k), detecting all stuck-at-a, $a \in M - \{0\}$, faults on i_k and j_k.
2. The second test of T results in applying $(*, c_k)$ to each pair (i_k, j_k), where c_k is the constant (nonzero) which is fed into the *k*th multiplication modulo m gate and $*$ denotes any value from M. It detects all j_k stuck-at-0 faults. This test also detects all stuck-at-a, $a \in M - \{1\}$, faults on the inputs of the multiplication modulo m gates.
3. The third test of T results in applying $(1, 0)$ to each pair (i_k, j_k), detecting all i_k stuck-at-0 faults.
4. The fourth test of T applies the value $(m-1)$ to the inputs of multiplication modulo m gates, detecting all stuck-at-1 faults on them.

Hence, the four tests completely test the internal lines for all single stuck-at faults. □

The above theorem gives us the number of tests which are sufficient to detect all internal single stuck-at faults in a circuit realizing a modulo m sum-of-products form. Since the proof is constructive, it shows how to generate the test set itself. This test set is universal, i.e., independent of the function being realized.

Unfortunately, unlike the two-valued case, the test set T, given by Theorem 1, cannot guarantee the detection of other than stuck-at type faults in addition modulo m gates. For example, consider the addition modulo m gate in Fig. 1, which is first in the cascade. If its functionality is changed to *truncated sum* function, defined by $\mathrm{TSUM}(x_1, x_2) = MIN(x_1 + x_2, m-1)$, with "+" being the regular arithmetic addition, then T will not detect such a fault. The first test of T applies $(0, 0)$ to (i_1, j_1). However, the addition modulo m function is equivalent to truncated sum function for these values, i.e., $\mathrm{TSUM}(0, 0) = 0 \oplus 0 = 0$, so the first test does not detect this fault. The second test applies $(0, c_1)$ to (i_1, j_1), where c_1 is the constant fed into the first multiplication modulo m gate. Since, for any $0 \leq c_1 \leq m-1$, $\mathrm{TSUM}(0, c_1) = 0 \oplus c_1 = c_1$, the fault is not detected. The third test applies $(1, 0)$ to (i_1, j_1), but, similarily, $\mathrm{TSUM}(1, 0) = 1 \oplus 0 = 1$. The fourth test applies $(0, d_1)$ to (i_1, j_1), where $d_1 = c_1 \cdot (m-1)^r$, with r being the number of the inputs of the first multiplication modulo m gate. Since, for any $0 \leq d_1 \leq m-1$, $\mathrm{TSUM}(0, d_1) = 0 \oplus d_1 = d_1$, the fault will not be detected. Thus, none of the four tests of T detect the fault changing the functionality of addition modulo m gate to truncated sum function.

5 Testability by Hardware Redundancy

It is shown in [13] that, by providing a binary Reed-Muller circuit with an extra AND gate having an observable output, $n + 4$ tests for internal lines also detect all single stuck-at faults on primary inputs. We show that a similar technique can be used to ensure that the four tests for internal lines given by Theorem 1 also detect all single stuck-at faults on primary inputs of a circuit realizing a modulo m sum-of-products form. Notice, that this is asymptotically better as compared to the binary case.

Consider a modulo m sum-of-products circuit realization of an *m*-valued function $f(x_1, \ldots, x_n)$ having an extra multiplication modulo m gate G^* depending on all input variables $x_1, \ldots, x_n$ and with an output g (Fig. 3). If g is also observable, then two input assignments $(a_1, \ldots, a_n)$ and $(b_1, \ldots, b_n)$ such that $a_i, b_i \neq 0$ and $a_i \neq b_i$ for all i, detect all single stuck-at faults on the inputs of G^*. These two tests also detect all single stuck-at faults on primary inputs $x_1, \ldots, x_n$ since a single path is sensitized from each x_i to the output g. Observing the second and the fourth tests from the test set T from Theorem 1:

x_0	x_1	x_2	...	x_n
0	0	0	...	0
0	1	1	...	1
1	0	0	...	0
0	$m-1$	$m-1$	...	$m-1$

we see that the assignments for $x_1, \ldots, x_n$ satisfy the requirements $a_i, b_i \neq 0$, and $a_i \neq b_i$ for all $i \in \{1, \ldots, n\}$. Thus, the test set T detects all single stuck-at faults on primary inputs as well as on the inputs of G^*.

So, by adding to the circuit an extra multiplication modulo m gate with an observable output, the number of tests needed to detect all single stuck-at faults is reducible to four.

6 Conclusion

In this paper, we extended the result of Reddy [13] to m-valued case, for m being a prime greater than two. We show that, for $m > 2$, several new phenomena occur which allow us to reduce the upper bound on the number of tests required for fault detection to $2n + 4$, but make the generation of tests harder. We also show that, by adding to the circuit an extra multiplication modulo m gate with an observable output, the number of tests required to detect all single stuck-at faults in the circuit is reducible to four universal tests, which is asymptotically better compared to the reduction obtained in binary case from applying a similar technique.

References

[1] E.V. Dubrova and J.C. Muzio, "Testability of Generalized Multiple-Valued Reed-Muller Circuits," *Proc. 26th Int'l Symp. Multiple-Valued Logic,* pp. 56-61, May 1996.

[2] Intel Corp., "MultiLevel Cell Technology," 1998, http://developer.intel.com/design/flcomp/isfbgrnd.htm.

[3] A. Cataldo, "Flash Vendors Team Up to Take on Intel," *EE Times*, 26 Feb. 1998.

[4] K.W. Current, "Current-Mode CMOS Multiple-Valued Logic Circuits" *IEEE J. Solid-State Circuits,* vol. 29, no. 2, pp. 95-107, Feb. 1994.

[5] T. Hanyu, K. Teranishi, and M. Kameyama, "Design of One-Transistor-Cell Multiple-Valued CAM," *IEEE J. Solid-State Circuits,* vol. 31, no. 11, pp. 1,669-1,674, Nov. 1996.

[6] T. Waho, K.J. Chen, and M. Yamamoto, "Resonant-Tunneling Diode and HEMT Logic Circuits with Multiple Thresholds and Multilevel Output," *IEEE J. Solid-State Circuits,* vol. 33, no. 2, pp. 268-274, Feb. 1998.

[7] T. Baba, "Development of Quantum Functional Devices for Multiple-Valued Logic Circuits," *Proc. 29th Int'l Symp. Multiple-Valued Logic,* pp. 2-8, May 1999.

[8] A. Kazeminejad, K. Navi, and D. Etiemble, "CML Current Mode Full Adders for 2.5 V Power Supply," *Proc. 24th Int'l Symp. Multiple-Valued Logic,* pp. 10-15, May 1994.

[9] O. Ishizuka, A. Ohta, K. Tanno, Z. Tang, and D. Handoko, "VLSI Design of a Quaternary Multiplier with Direct Generation of Partial Products," *Proc. 27th Int'l Symp. Multiple-Valued Logic,* pp. 169-175, May 1997.

[10] T. Hanyu and M. Kameyama, "A 200 MHz Pipelined Multiplier Using 1.5 V-Supply Multiple-Valued MOS Current-Mode Circuits with Dual-Rail Source-Coupled Logic," *IEEE J. Solid-State Circuits*, vol. 30, no. 11, pp. 1,239-1,246, Nov. 1995.

[11] T.-H. Kuo, H.C. Lin, R. Potter, and D. Schupe, "Multiple-Valued Counter," *IEEE Trans. Computers,* vol. 42, no. 1, pp. 106-109, Jan. 1993.

[12] T. Waho and M. Yamamoto, "Application of Resonant-Tunneling Quaternary Quantizer to Ultrahigh-Speed A/D Convertor," *Proc. 27th Int'l Symp. Multiple-Valued Logic,* pp. 35-41, May 1997.

[13] S.M. Reddy, "Easy Testable Realizations for Logic Functions," *IEEE Trans. Computers,* vol. 21, no. 11, pp. 1,183-1,188, Nov. 1972.

[14] K.K. Saluja and S.M. Reddy, "Fault Detecting Test Sets for Reed-Muller Canonic Networks," *IEEE Trans. Computers,* vol. 24, no. 1, pp. 995-998, Jan. 1975.

[15] D.K. Pradhan, "Universal Test Sets for Multiple Fault Detection in AND-XOR Arrays," *IEEE Trans. Computers,* vol. 27, no. 2, pp. 181-187, Feb. 1978.

[16] J.C. Muzio and T.C. Wesselkamper, *Multiple-Valued Switching Theory.* Bristol and Boston: Adam Hilger Ltd., 1986.

[17] M.A. Breuer and A.D. Friedman, *Diagnosis and Reliable Design of Digital Systems.* Computer Science Press, Digital Systems Design Series, 1976.

[18] T. Sasao, "Easy Testable Realizations for Generalized Reed-Muller Expressions," *IEEE Trans. Computers,* vol. 46, no. 6, pp. 709-716, June 1997.

[19] M. Cohn, "Switching Function Canonical Form over Integer Fields," PhD thesis, Harvard Univ., Cambridge, Mass., Dec. 1960.

[20] I.S. Reed, "A Class of Multiple-Error-Correcting Codes and Their Decoding Scheme," *IRE Trans. Information Theory,* vol. 4, pp. 38-42, 1954.

[21] D.E. Muller, "Application of Boolean Algebra to Switching Circuit Design and to Error Detection," *IRE Trans. Electronic Computers,* vol. 3, pp. 6-12, 1954.

[22] T. Sasao and P. Besslich, "On the Complexity of mod-2 Sum PLA's," *IEEE Trans. Computers,* vol. 39, no. 2, pp. 262-266, Feb. 1990.

[23] G. Birkhoff and S. MacLane, *Brief Survey of Modern Algebra,* fourth ed. New York: Macmillan, 1977.