Highly Testable Finite State Machines Based on EXOR Logic

Ugur Kalay[1], Nagesh Venkataramaiah[2], Alan Mishchenko[1], Douglas V. Hall[1], Marek A. Perkowski[1]

[1] Dept. of Electrical and Computer Engineering
Portland State University
P.O. Box 751, Portland, OR 97207-0751
ugurkal@ee.pdx.edu

[2] Integrated Measurement Systems
9525 SW Gemini Drv.
Beaverton, OR 97008
nagesh@ims.com

Abstract: *It is well-known that AND/EXOR circuits are more easily testable than AND/OR circuits. Therefore, in this paper, we primarily propose to use AND/EXOR realizations for implementation of the combinational logic parts of finite state machines. Then, we investigate the effect of different state assignments (i.e. one-hot, grey-code, etc.) and that of using different types of registers (i.e. D-type, JK-type, etc.) on the testability of finite state machines. As the basis of our measurements, we considered two easily testable AND/EXOR realizations; one for EXOR Sum-of-Products expressions and the other for Generalized Reed-Muller expressions. We make comparisons of these realizations in terms of area and the number of test patterns as we change the state assignment and the type of registers. We also show that 2-level AND/EXOR realizations can yield less area than 2-level AND/OR realizations in the implementation of finite state machines.*

I. INTRODUCTION

The problem of designing highly testable finite state machines is known to be difficult. Many of the previous solutions suggested making the state registers partially/fully controllable and observable [2,4], or inserting extra scan registers into arbitrary nodes in the design [16]. However, these methods require difficult circuit modifications and very long pseudo-random test sequences for Built-in Self Testing (BIST). Furthermore, they usually do not guarantee 100% fault coverage for single stuck-at faults. A possible solution, not investigated until now, is to use easily testable AND/EXOR circuits for the combinational logic (CL) parts of Finite State Machines (FSM).

In general, AND-EXOR logic implementations require fewer gates and connections than AND-OR logic implementations. In addition, they require fewer test patterns and give a testing time complexity of $O(n)$, where n is the number of inputs. The AND-EXOR form has been developed into a complete hierarchy of Reed-Muller expansions using the Shannon, Positive Davio, Negative Davio expansions, and other transformations related to generalized forms [13]. Reddy showed that highly testable circuits can be realized for the Positive Polarity Reed-Muller (PPRM) expression of a function [11]. However, since a PPRM does not allow the complemented forms of input variables in an expression, it often yields large expressions. Because of this, researchers have investigated other Reed-Muller forms such as Fixed Polarity Reed-Muller (FPRM)[12], and Generalized Reed-Muller (GRM) [14], that allow more flexibility in the expression. These usually require fewer product terms, and therefore less area [13]. Although it is not a Reed-Muller form, the most flexible (non-restricted) AND/EXOR expression is an EXOR Sum-of-Products (ESOP). In most cases an ESOP yields the shortest expression for AND/EXOR circuits due to its flexibility. This form has been investigated for testability as described in [5,10].

Combinational Logic is commonly used for the *next-state* logic and the *output-decode* logic in a sequential machine. The complexity of the functions required for each of these is highly dependent on the bit encoding of the internal states (state assignment) and on the type of registers used in the realization. Therefore, these two parameters should be taken into account when investigating the overall testability of a sequential machine. For the work described in this paper, we implemented the CL parts for a wide variety of state machines with two, easily testable AND/EXOR realizations. For each, we then calculated the number of tests and the area required for these implementations with different state assignments and different types of registers.

The organization of this paper is as follows. In the next section, the two AND/EXOR realization schemes are given. Section 3 introduces the technology library referred to in the area measurements, Section 4 presents the measurement results, and Section 5 gives our conclusions.

II. EASILY TESTABLE AND/EXOR REALIZATIONS

The first CL realization and testing scheme used in this paper was proposed for the GRM form by Sasao in [14]. A GRM does not have any restrictions on input variable polarities, so it is the most flexible of the Reed-Muller forms. However, it has the limitation that an expression can have only a single product term containing a given subset of variables. Sasao's easily testable GRM realization is illustrated for the function $f = x_1 \oplus \overline{x}_1\overline{x}_2 \oplus x_2\overline{x}_3$ in Figure 1. The *literal part* provides the complements of the input variables. The AND part, the EXOR part, and the literal part implement the GRM expression. The EXOR part is implemented as an EXOR tree structure. The *check part* is designed for testability (DFT), and added to detect faults in the literal part and in the primary inputs. Overall, this easily testable implementation requires one additional input, c, and four additional outputs, $o_1 - o_4$. The test set proposed in [14] for this realization is dependent on the function being realized and is therfore not universal.

The second CL realization and testing scheme used in this paper was proposed for ESOPs, the most general, non-restricted AND/EXOR expression, by Kalay (et al.) in [5]. For a given function, an ESOP expression is always the same or shorter than a GRM form. Figure 2 shows the easily testable ESOP realization for the function $f = x_1x_5 \oplus x_1x_2x_3 \oplus x_2x_3x_4 \oplus \overline{x}_2\overline{x}_3\overline{x}_4$. In

0-7803-5582-2/99/$10.00 ©1999 IEEE

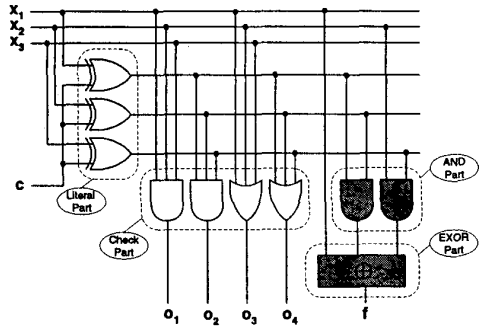

Figure 1. Easily testable GRM realization proposed by Sasao [14].

this implementation, the names of the blocks and their functions are very similar to those in Sasao's. The major difference in this implementaton is that for improved testability, the EXOR level, *linear part*, is implemented as a cascade of 2-input EXOR gates rather than as a tree of EXOR gates. For the same reason, the check part is also implemented as a cascade of EXOR gates.

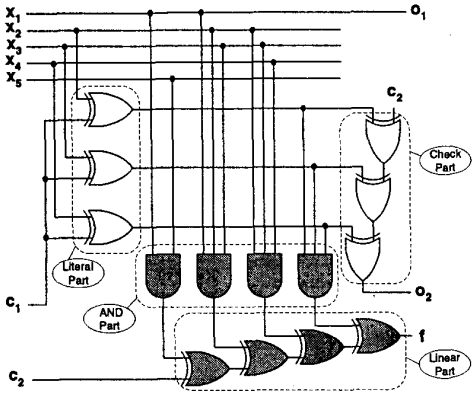

Figure 2. Easily testable ESOP realization proposed by Kalay (et al.) [5].

Table 1 compares the properties of the two implementation schemes. In the table, p represents the number of product terms, and n represents the number of primary inputs to the CL part. Note that the test set for the implementation shown in Figure is universal. Also note that this implementation requires a smaller number of tests than the implementation shown in Figure 1.

III. TECHNOLOGY LIBRARY

A 0.5 micron, array-based library developed by LSI Logic Corporation was used for synthesis [6]. We limited the number of components for our measurements according to Table 2. Referring to this table, all area measurements in this paper were expressed in cell units, and the area of interconnections was not included. Multiple input EXOR gates were not included in the library because both realization schemes use 2-input EXOR gates either in a tree or in a cascade structure. Also, for multiple input AND(OR) gates, a composite gate that is made out of 2-input AND(OR) gates is assumed. The INVERTER gate is used only in the measuments of AND/OR (SOP) implementations. This is because both

	GRM(Sasao)	ESOP(Kalay, et al.)
Additional gates and I/O pins	Yes	Yes
Universal test patterns	No	Yes
Number of tests for single stuck-at faults	$p+2n+5$	$n+6$
Realization of the EXOR level	Tree	Cascade

Table 1. Comparison of two AND/EXOR realization and testing schemes.

Component	Area (cell units)
INVERTER	1
AND2	2
OR2	2
XOR2	3

Table 2. The technology library used in measurements.

AND/EXOR realization schemes used in this paper implement an INVERTER with a 2-input EXOR gate.

IV. EXPERIMENTAL RESULTS

A synthesis tool, SYNTHA [7], was used to synthesize a FSM based on three state assignment schemes (1-hot, grey-code, and polynomial[8]), and three types of registers (D, JK, and T). The benchmark FSM descriptions were selected from *LGSynth93* benchmark set. SYNTHA accepts a FSM description in *kiss* format and generates a 2-level AND/OR (SOP) expression for the CL part of the FSM in *blif* (ESPRESSO) format. The obtained SOP expression was applied to another tool, DISJOINT [3], to obtain an equivalent 2-level AND/EXOR expression. To obtain the minimal ESOP expression, this expression was applied to an ESOP minimizer, EXORCISM [15]. Similarly, to obtain the minimal GRM expression, the output of DISJOINT was applied to a GRM minimizer, CGRMIN [9]. Next, the additional gates required for the DFT parts of the aforementioned GRM and ESOP realization schemes were designed.

Table 3 gives the number of product terms, the area (function and DFT), and the number of tests required for the easily testable GRM implementation of the benchmark circuits. Similarly, Table 4 gives the same measurement data for the easily testable ESOP implementation of the same benchmark circuits. Each entry in Table 3 and Table 4 is in the $a/b(c)/d$ format, where a, b, c, and d are the number of product terms, the area of the functional part, the area of the DFT part, and the number of test patterns, respectively. Some entries in the table were left blank since the minimization tool reached its capacity and quit due to too many product terms or too many inputs.

Our last measurement was performed for the 2-level Sum-of-Products (SOP) implementation of the CL part to analyze the area differences from the AND/EXOR implementations. Two-level implementations are important for PLDs or EXOR PLDs (XPLDs), not only for the total area, but also for the number of

product terms and literals that the functional expression yields. After the non-minimal SOP expression for the CL part was obtained from SYNTHA, ESPRESSO [1] was used to obtain the minimal SOP expression. It is known that AND/OR realizations generally require non-universal deterministic or random test patterns, which also require fault simulation. In addition, they require much longer test sequences than the 2-level ESOP implementation proposed by Kalay (et al.) and shown in [5]. Table 5 gives the number of product terms and the area for the same benchmark circuits used for the ESOP/GRM implementations. Each entry in Table 5 is in the a/b format, where a and b are the number of product terms, and the area of the functional part, respectively.

With three different state assignments and three different types of registers applied to nine benchmark functions, 81 different functions were analyzed for comparisons. The summary of the observations can be given as follows:

- in 78 cases, the ESOP realization scheme yielded less area than the GRM realization scheme, and in 5 of them they yielded very close areas,
- in 26 cases, the ESOP scheme yielded significantly smaller area than the GRM scheme, and in 9 of them the GRM scheme yielded 3-4 times larger area.
- in all cases, the ESOP scheme required much fewer test patterns. This is because the test set of the GRM scheme is not only dependent on the number of inputs but also on the number of product terms. Also, the GRM test set is twice as much more sensitive to the number of inputs than the ESOP test set.
- in all cases, the DFT area of the ESOP scheme is about 50% of the DFT area of the GRM scheme.
- in 40 cases, ESOP realizations yielded fewer product terms than the SOP realizations, in 17 of them ESOP yielded less area. Also in 18 cases, both realizations yielded the same number of product terms.
- in general, AND/EXOR implementations yielded less area with grey-code encoding and T or D type flip-flops, whereas AND/OR implementations yielded less area with grey-code or polynomial encoding and JK type flip-flops.

V. CONCLUSIONS

In this paper, we proposed implementing the combinational logic part of a finite state machine with AND/EXOR logic to make it more testable. Two easily testable AND/EXOR realizations were compared with different state assignment schemes and different register types. Three state assignment schemes (1-hot, grey-code, and polynomial), and three types of registers (D, JK, and T) were considered. The AND/EXOR realizations were compared in terms of area (in number of cells) and in terms of the number of test patterns required for each. Our results show that the ESOP realization scheme shown in Figure 2 is far more efficient than the GRM realization scheme in terms of both area and the required number of test patterns for single stuck-at faults. The only potential advantage of the GRM scheme is that it may have less propagation delay because it uses an EXOR tree instead of an EXOR cascade. (For this paper we did not investigate this.) A further point demonstrated in this paper was the fact that two-level AND/EXOR (ESOP) implementations can require fewer product terms and smaller areas than the equivalent two-level AND/OR (SOP) implementations. Remember that the ESOP test scheme requires a minimal and universal test set, which makes it superior for BIST as explained in [5].

REFERENCES

[1] R. Brayton, G. Hachtel, L. Hemachandra, A. Newton and A. S-Vincentelli, "A Comparison of Logic Minimization Strategies Using ESPRESSO - An APL Program Package for Partitioned Logic Minimization", in *Proc. Int. Symp. Circuits and Systems*, (Rome), pp. 43-49, May 1982.

[2] E. B. Eichelberger and T. W. Williams, "A Logic Design Structure for LSI Testing", in *Proc. Design Automation Conf.*, pp. 462-468, June 1977.

[3] B. J. Falkowski and M. A. Perkowski, "An Algorithm for the Generation of Disjoint Cubes for Completely and Incompletely Specified Boolean Functions", *Int. J. Electronics*, vol. 70, no. 3, pp. 533-538, 1991.

[4] R. Gupta and M. A. Breuer, "BALLAST: A Methodology for Partial Scan Design", in *Proc. Int. Symp. Fault-Tolerant Computing*, pp. 118-125, June 1989.

[5] U. Kalay, M. A. Perkowski and D. V. Hall, "A Minimal Universal Test Set for Self Test of EXOR-Sum-of-Products Circuits", *submitted to IEEE Trans. Comp.*, 1998.

[6] LSI Logic Corporation, *LCA/LEA500K Array-Based Products Databook*, fourth ed., May 1997.

[7] A. A. Mishchenko, "A CAD System for Automated Synthesis of Controlling Automata", *Kibernetika i Sistemnyj Analiz*,(in Russian), no. 3, pp. 23-30, 1997.

[8] A. A. Mishchenko and M. A. Perkowski, "TRACE: A Visual Software System to Explore Properties of Reed-Muller Movement Functions" in *Proc. Reed-Muller Conf.*, Aug. 1999.

[9] M. A. Perkowski, L. Csanky, A. Sarabi and I. Schaefer, "Fast Minimization of Mixed-Polarity AND/XOR Canonical Networks", in *Proc. Reed-Muller Conf.*, pp. 32-36, Oct. 1992.

[10] D. K. Pradhan, "Universal Test Sets for Multiple Fault Detection in AND-EXOR Arrays", *IEEE Trans. Comp.*, vol. 27, pp. 181-187, 1978.

[11] S. M. Reddy, "Easily Testable Realizations for Logic Functions", *IEEE Trans. Comp.*, vol. C-21, pp. 1183-1188, 1972.

[12] A. Sarabi and M. A. Perkowski, "Design for Testability Properties of AND-EXOR Networks", in *Proc. Reed-Muller Conf.*, pp. 418-424, Sept. 1993.

[13] T. Sasao, *Logic Synthesis and Optimization*, Norwell, MA: Kluwer Academic Publishers, 1993.

[14] T. Sasao, "Easily Testable Realizations for Generalized Reed-Muller Expressions", *IEEE Trans. Comp.*, vol. 46, no. 6, pp. 709-716, 1997.

[15] N. Song and M. A. Perkowski, "Fast Look-Ahead Algorithm for Approximate ESOP, Minimization of Incompletely Specified Multi-Output Boolean Functions", in *Proc. Reed-Muller Conf.*, pp. 61-72, Sept. 1997.

[16] J. H. Steward, "Future Testing of Large LSI Circuit Cards", in *Proc. Semiconductor Test Symp.*, pp. 6-17, 1977.

Circuit	1-Hot			Grey-code			Polynomial		
	D	JK	T	D	JK	T	D	JK	T
bbara	45/1288(108)/80	45/1234(108)/80	45/1228(108)/80	38/532(52)/61	38/597(52)/61	35/485(52)/58	51/790(52)/74	44/709(64)/67	47/710(52)/70
cse	97/4148(180)/150		97/4226(180)/150	132/2804(76)/161	138/3088(76)/167	135/2605(72)/164	132/2757(80)/161	120/2387(72)/149	133/2911(60)/162
dk27	18/299(68)/41	18/320(68)/41	18/335(68)/41	14/122(28)/29	15/159(40)/30	15/150(40)/30	14/130(32)/29	16/155(28)/31	14/146(40)/29
donfile	98/5331(212)/157		96/1592(60)/117	81/1085(56)/102	90/1340(56)/111	81/1083(56)/102	108/1610(44)/129	90/1340(56)/111	108/1642(52)/129
lion	18/231(52)/37	18/243(52)/37	11/100(32)/26	11/100(32)/26	14/123(28)/29	14/123(28)/29	11/100(32)/26	14/123(28)/29	11/86(36)/26
modulo12	27/723(108)/60	27/689(108)/60	26/627(108)/59	23/228(32)/40	15/182(28)/32	13/142(32)/30	23/232(44)/40	16/186(32)/33	13/113(32)/30
s386	62/2204(156)/109		62/2264(156)/109	72/1206(84)/101	76/1357(84)/105	231/5344(80)/264	83/1472(84)/112	78/1439(76)/107	82/1526(84)/111
train11	36/992(112)/69	37/1095(112)/70	37/1056(112)/70	36/537(52)/55	41/554(52)/60	40/519(44)/59	43/603(44)/62	38/606(52)/57	40/571(56)/59
s1488				447/9548(100)/482	448/10233(92)/483	448/9951(92)/483	570/12457(92)/605	572/13398(92)/607	471/10299(92)/506

Table 3. Measurements on GRM implementations (Number of Product Terms/Functional Area (DFT Area)/Number of Test Patterns).

Circuit	1-Hot			Grey-code			Polynomial		
	D	JK	T	D	JK	T	D	JK	T
bbara	37/1142(39)/21	29/930(39)/21	29/960(39)/21	23/387(21)/15	24/467(21)/15	18/340(21)/15	27/456(21)/15	23/432(21)/15	23/453(21)/15
cse	65/2849(69)/30	61/2767(69)/30	61/2783(69)/30	45/1003(33)/18	52/1055(33)/18	50/1000(33)/18	47/1019(33)/18	51/1002(33)/18	52/1036(33)/18
dk27	14/272(27)/15	14/290(27)/15	14/320(27)/15	9/113(15)/11	10/118(15)/11	10/122(15)/11	9/104(15)/11	12/149(15)/11	11/126(15)/11
donfile	100/5709(81)/33	99/5973(81)/33	56/897(27)/14	43/652(27)/14	50/769(27)/14	45/683(27)/14	63/1076(27)/14	50/769(27)/14	65/1123(27)/14
lion	10/179(21)/13	9/147(21)/13	6/79(15)/11	6/79(15)/11	8/96(15)/11	8/96(15)/11	6/79(15)/11	8/96(15)/11	7/81(15)/11
modulo12	24/738(45)/20	12/351(39)/20	12/423(39)/20	13/164(21)/12	10/119(15)/12	8/126(15)/12	14/188(21)/12	11/169(15)/12	10/136(15)/12
s386	43/1585(63)/27	39/1501(63)/27	39/1535(57)/27	31/590(33)/18	35/641(33)/18	39/1087(39)/20	32/593(39)/18	35/661(39)/18	33/644(39)/18
train11	25/787(45)/20	24/721(45)/20	24/764(45)/20	14/269(21)/13	20/333(21)/13	19/340(21)/13	18/370(21)/13	21/361(21)/13	20/349(21)/13
s1488	170/18760(171)/63	170/18742(171)/63	170/18727(171)/63	127/2872(45)/21	147/3590(45)/21	143/3498(45)/21	132/3054(45)/21	155/4575(45)/21	151/4014(45)/21

Table 4. Measurements on ESOP implementations (Number of Product Terms/Functional Area (DFT Area)/Number of Test Patterns).

Circuit	1-Hot			Grey-code			Polynomial		
	D	JK	T	D	JK	T	D	JK	T
bbara	50/1351	33/963	33/983	29/359	23/323	28/415	32/413	24/363	29/431
cse	91/3619	69/2825	69/2857	51/921	50/809	57/869	53/965	47/751	57/903
dk27	14/251	14/263	14/277	11/91	10/73	11/85	11/93	11/81	13/105
donfile	96/5353	96/5161	53/764	49/636	41/550	53/706	74/1120	41/550	67/996
lion	11/155	9/123	6/51	6/51	7/56	7/56	6/51	7/56	7/59
modulo12	24/662	12/325	12/349	13/116	10/85	10/113	12/114	10/103	8/77
s386	54/1797	47/1619	47/1645	36/521	32/478	46/1016	38/512	33/526	40/604
train11	25/726	24/652	24/674	13/199	16/221	21/287	17/297	19/255	20/273
s1488	182/19683	181/19727	181/19823	156/2954	145/2781	153/3055	158/3056	152/2907	168/3281

Table 5. Measurements on SOP implementations (Number of Product Terms/Functional Area).