VLSI DESIGN
1997, Vol. 5, No. 3, pp. 241–252
Reprints available directly from the publisher
Photocopying permitted by license only

I_{DDQ} Detectable Bridges in Combinational CMOS Circuits

E. ISERN* and J. FIGUERAS

Departament d'Enginyeria Electrònica, Universitat Politècnica de Catalunya, Diagonal 647, 08028 Barcelona. Spain

Undetectable stuck-at faults in combinational circuits are related to the existence of logic redundancy (*s-redundancy*). Similarly, logically equivalent nodes may cause some bridging faults to become undetectable by I_{DDQ} testing. An efficient method for the identification and removal of such functionally equivalent nodes (*f-redundant nodes*) in combinational circuits is presented. OBDD graphs are used to identify the functional equivalence of candidate to f-redundancy nodes. An f-redundancy removal algorithm based on circuit transformations to improve bridging fault testability, is also proposed. The efficiency of the identification and removal of f-redundancy has been evaluated on a set of benchmark circuits.

Keywords: Bridging faults, current testing, functional redundancy, functional equivalence, functional redundancy removal

1. INTRODUCTION

Bridging faults (BF), or shorts between normally disconnected circuit lines, are one of the common types of failures in CMOS circuits. Studies examining realistic faults in CMOS digital circuits suggest that bridging faults may account for from thirty to fifty percent of all faults [11, 14].

The detection of logic bridging faults (LBF) by logic testing has been widely investigated. A fixed boolean relationship (such as a wired AND or a wired OR) has been assumed to model the value of the shorted defective nodes [23, 2]. However, this *wired model* has been found inappropriate for CMOS circuits because the short might produce intermediate voltage levels with unpredictable logic behaviour [26, 29].

The testing method based on quiescent current consumption monitoring (also called I_{DDQ} testing) has been shown to be efficient in the detection of this kind of faults in CMOS circuits [19, 20, 25, 28].

The conditions for bridge detectability by I_{DDQ} testing have been investigated by many authors [19, 20, 25, 8, 27, 10]. The most commonly used fault detectability criterion consists in controlling the bridged nodes to complementary logic values in the fault-free circuit [19, 20].

In this paper, only bridging faults between logic nodes (LBFs) are considered. Throughout the work, a LBF will be considered to be testable by current testing if the shorted nodes can be controlled to opposite

*Corresponding author. Tel.: 34-3-401 66 50. Fax: 34-3-40 7785. E-mail: isern@eel.upc.es, figueras@eel.upc.es.

logic values. For some feedback LBFs the circuit may oscillate. In this case, the current consumption exhibits also periodical oscillations. The mean value of the oscillating current can be shown to be greater than zero and, therefore, the defect causing the oscillation may be testable by current testing [27]. Moreover, there may be some feedback LBFs affecting nodes of the same logic behaviour and giving rise to detectable faults. As an example, Figure 1 presents a feedback LBF in a inverter chain. Let us assume that signal D is strong whereas signal B is weak (the transistors of the inverter with output D are larger than the ones of the inverter generating B). The last two inverters constitute a sequential circuit as if they were isolated. Its state might not be changeable by the weak input B. As a consequence, its state will be determined at power-up by the relative sizes of the circuitry. Now, if an input signal is forced at A such that the voltage at B attempts to change the state of the sequential circuit, the weak B inverter will not be able to commute D and significant I_{DDQ} might appear. In this work, this kind of LBFs is not included in the set of testable bridges. The reason for this is that its testability depends on the actual electrical implementation, which may be unknown. Thus, LBFs between nodes that have the same logic behaviour for all inputs to the circuit are considered as *redundant* LBFs. In other words, the presence of undetectable shorts may be related to some kind of redundancy in the circuit, namely groups of functionally equivalent nodes.

In order to distinguish the functional redundancy from the more commonly referred stuck-at redundancy, the former will be called *f-redundancy*, whereas the later will be referred to as *s-redundancy*.

The identification and removal of the stuck-at type redundancy have been reported by many authors [1, 3, 31, 9, 24]. On the contrary, the identification of functionally equivalent nodes in combinational circuits, that is, the f-redundancy identification problem, has received less attention [16]. In this work, this class of redundancy related to undetectable bridging faults is analysed.

The problem to determine if a set of nodes are f-redundant is equivalent to the well-known NP-complete problem of finding if two logic functions are equivalent. There are several methods to determine if two boolean functions are equivalent [15, 32]. Unfortunately, their power rated in terms of manageable circuit size and processing time is not always satisfactory. Recently, the verification of the equivalence between logic functions has been approached with the aid of ordered binary decision diagrams (OBDDs) [5, 7, 13, 17]. Since OBDDs are canonical representations of logic functions [7], they may be used for detecting the equivalence between two boolean functions, thus determining the detectability of bridging faults.

In this work, the detectability of LBF by I_{DDQ} testing in combinational circuits is investigated. An efficient method for the identification of functionally equivalent groups of nodes (called *f-redundant classes*) is presented. These classes are used in the analysis of the bridging fault coverage of test vector sets. The rest of the paper is organized as follows. Section 2 provides some definitions and explores the relation between functional and stuck-at redundancy. Section 3 presents the proposed method for functional redundancy identification, based on OBDDs graphs. The experimental results of the application on a set of benchmark circuits are presented in section 4. In the same section, an analysis of the bridging fault coverage of the same set of circuits, considering current consumption test, is developed. An f-redundancy removal approach is considered in section 5, and some results of its application are reported. Finally, conclusions on the proposed bridge detectability analysis approach are stated in section 6.

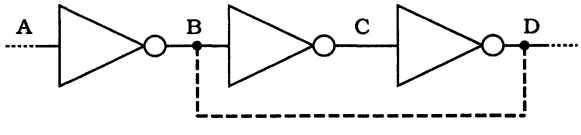

FIGURE 1 Detectability of a feedback bridging fault. If B is weak and D strong, the LBF B-D may be I_{DDQ} testable.

2. FUNCTIONAL VS. STUCK-AT REDUNDANCY

The presence of some undetectable stuck-at faults in digital combinational circuits implies that part of the

logic of the circuits is redundant and can be removed. In this paper, the redundancy related to undetectable stuck-at faults will be referred to as *s-redundancy*.

As indicated above, undetectable bridging faults by current observation can be related to another type of redundancy, namely *f-redundancy*, different from the s-redundancy.

In order to state the problem of f-redundancy identification properly, the following terminology is defined. *F-redundant nodes*: Two logic nodes α and β in a combinational logic circuit $\mathcal{C}$ are said to be *f-redundant* iff, for all primary input values of $\mathcal{C}$, both nodes take the same logic value. *F-redundant classes*: The relation "to be f-redundant" is an equivalence relation and it divides the circuit node set into disjunct subsets or equivalence classes. Henceforth, every class will be referred to as an *f-redundant class*. *LBF detectability*: Classes with one node are irredundant and any LBF between the single node of the class and any other node of the circuit will be assumed to be detectable. On the other hand, LBFs between nodes of the same class are assumed to be undetectable by current observation and, hence, f-redundant.

Let us now explore the relation between s-redundancy and f-redundancy. Simple examples are used to show that one type of redundancy does not imply the other.

A circuit irredundant in the sense of s-redundancy may contain some f-redundant nodes. Conversely, a circuit with no f-redundancy may have some untestable stuck-at faults, that is, it can be s-redundant. There is no dependence between both kinds of redundancies. Figure 2 provides some simple examples of this independence. Figure 2.a) shows a circuit that is s-redundant (b_1 s-1 and b_2 s-1 are undetectable) but is not f-redundant (all the columns in the truth table are different); Figure 2.b) shows a circuit that at the same time is s-redundant (b_1 s-1 and b_2 s-1 are undetectable) and f-redundant (C ≡ Y); and finally, Figure 2.c) shows a circuit that is not s-redundant but f-redundant (C ≡ Y). The analysis of bridge detectability is hence equivalent to the f-redundancy analysis. Once the f-redundancy classes have been determined, all the LBFs between nodes within these classes are assumed to be undetectable, whereas the remaining may be I_{DDQ} testable.

3. METHODOLOGY OF LBF DETECTABILITY ANALYSIS

In this section, a method for identifying f-redundant classes in combinational logic circuits, and hence, identifying undetectable LBFs by current testing, is

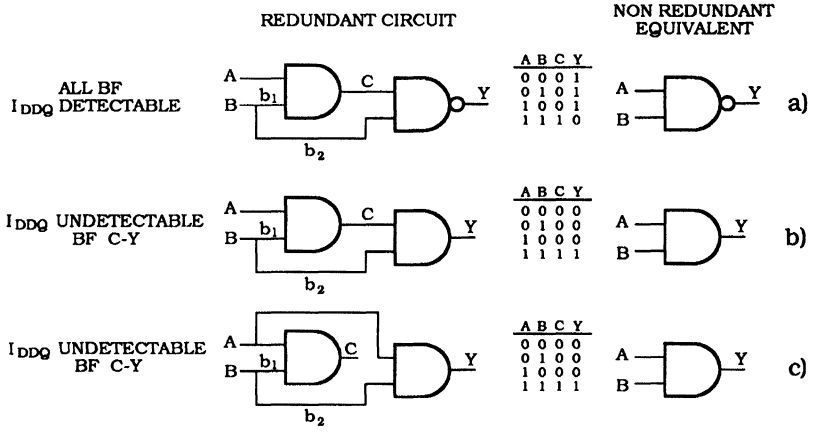

FIGURE 2 Examples of a) circuit s-redundant but not f-redundant; b) circuit s-redundant and f-redundant; c) circuit f-redundant but not s-redundant, and its effect on bridge detectability.

proposed. This method is based on OBDD comparisons.

The large number of node couples in a circuit $N(N-1)/2$ couples in a circuit with N nodes) makes it impractical to look for f-redundant classes by means of checking all the possible couples on nodes. An alternative strategy is proposed. First, *candidate to f-redundant classes* are identified and then, the f-redundancy among the nodes of these candidate classes are verified.

Following this strategy, which has been found to be highly successful in the experiments performed, the analysis consists of two distinct phases:

Phase 1: A sequence of K pseudorandom input vectors is applied to the circuit. For each node, a binary identifier of K digits is generated by using the logic values taken by the node when the input sequence is applied. Two nodes will belong to the same *candidate to f-redundant class* if and only if they have the same identifier. The size of these classes decreases as the number of input vector increases. A value of $K = 0.2 \times$ (#gates) has been found experimentally to be a good number of input vectors. This value keeps the maximum size of the candidate classes small enough to allow the construction of the OBDDs for each pair of nodes within these classes.

Phase 2: Once the possible f-redundant classes have been restricted to the set of candidate to f-redundant classes, the actual f-redundancy classes are determined. To do that, the f-redundancy between all the couples of nodes within the candidate classes is checked. Given a couple of nodes α and β, with associate functions f_α and f_β, their OBDDs are constructed. If the two graphs are equal, α and β will be f-redundant. The problem can be simplified due to the locality of many redundant logic structures. In most analysed cases, the OBDDs can be constructed from a set of internal predecessor nodes S, which uniquely determine both functions f_α and f_β, instead of using the primary inputs. To illustrate this fact, Figure 3.a) shows a portion of a circuit $\mathcal{C}$ containing two redundant nodes α and β. If the two OBDDs associated with f_α and f_β are constructed using variables a, b, c, the same graph (shown in Figure 3.b)) is obtained. On the other hand, it is not indispensable that a, b, c be primary inputs of $\mathcal{C}$; it is sufficient that a, b, c form a set of predecessor nodes S which uniquely determine both functions f_α and f_β. Thus, the OBDDs so constructed will have in general a smaller size than if they had been constructed from a set of primary inputs.

4. METHODOLOGY EVALUATION

The OBDD algorithms reported in [7] have been implemented in C. The utility HISIM of System HILO has been used for the logic simulation of the circuits and a SUN SPARCstation 2 workstation supported the development. The circuits chosen to experiment the proposed method have been the following:

- The ISCAS'85 combinational benchmark circuits [6].

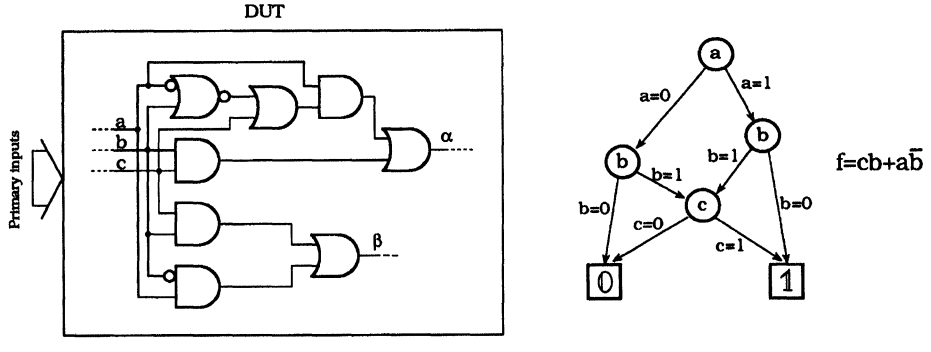

FIGURE 3 An example of two f-redundant nodes α and β (a) and their OBDD graph (b).

TABLE I Summary of the F-redundancy analysis for the benchmark circuits

Cir	LBF	T1 (sec)	T2 (min:sec)	Total time (min:sec)	DLBF	UDLBF	INDLBF	Reduced time (sec)
C432	19503	0.1	2:11	2:11	19496	7	0	18.7
C499	29890	0.1	9:19	9:19	29810	80	0	7.0
C880	98790	0.2	0:38	0:38	98718	72	0	0.5
C1355	173166	0.2	31:15	31:15	173054	112	0	16.7
C1908	418155	0.4	48:45	48:45	417208	947	0	3.1
C2670	1018878	0.9	11:47	11:48	1017944	934	0	0.8
C3540	1480060	1.2	21:00	21:01	1477865	2193	2	0.6
C5315	3091341	2.2	42:48	42:50	3089879	1462	0	1.8
C6288	3000025	1.3	3:45	3:46	2999825	183	17	1.1
C7552	6921060	5.6	120:28	120:34	6917980	2921	159	2.3
alu4	8128	0.1	0:11	0:11	8088	40	0	0.3
mul4	19701	0.1	0:17	0:17	19645	56	0	0.3
mul8	258840	0.3	12:35	12:35	258562	277	1	2.7
mul12	1223830	0.5	25:28	25:29	1223044	771	15	1.9
mulb	752151	0.8	11:08	11:09	750374	1777	0	0.4

DLBF: Detectable logic bridging faults
UDLBF: Undetectable logic bridging faults
INDLBF: Indeterminate logic bridging faults
T1: Phase 1 time
T2: Phase 2 time
Reduced time: (T1 + T2)/(UDLBF + INDLBF).

- Four circuits (referred to as mul4, mul8, mul12 and alu4) mul4, mul8 and mul12 are 4 × 4, 8 × 8 and 12 × 12 multiply/accumulate circuits respectively, generated by wiring up Am25S05 elements [4, 30]. alu4 is a four-bit ALU/function generator [4, 30].
- An 8 × 8 modified Booth multiplier (mulb) [22].

For every circuit the process described above has been carried out. The main part of the algorithm is the construction of the OBDDs graph. It has been widely reported that the OBDD size is strongly related to the choice of the variable order [7, 13, 21, 12]. In this work, the ordering heuristic proposed by Fujita et al. in [13] has been used in order to keep the size of the OBDDs reasonably bounded. The OBDDs constructor program provides, for every node couple, one of the following results:

- The two nodes are f-redundant.
- The two nodes are not f-redundant.

TABLE II Frequencies of the size (number of nodes) of the OBDDs generated for f-redundancy identification

OBDD size range	Circuit														
	c432	c499	c880	c1355	c1908	c2670	c3540	c5315	c6288	c7552	alu4	mul4	mul8	mul12	mulb
0–50	100%	100%	100%	100%	77.21%	65.07%	94.00%	79.37%	74.19%	78.04%	46.15%	100%	97.06%	95.92%	86.17%
50–100	—	—	—	—	13.49%	19.54%	5.70%	12.30%	25.81%	12.74%	—	—	2.94%	4.08%	7.85%
100–150	—	—	—	—	5.81%	5.61%	—	1.64%	—	1.68%	—	—	—	—	0.56%
150–200	—	—	—	—	1.86%	3.33%	—	2.46%	—	2.24%	53.85%	—	—	—	—
200–250	—	—	—	—	0.47%	1.46%	—	1.37%	—	1.13%	—	—	—	—	—
250–300	—	—	—	—	0.47%	0.62%	—	0.41%	—	0.88%	—	—	—	—	0.19%
300–350	—	—	—	—	0.47%	0.62%	—	0.54%	—	0.88%	—	—	—	—	—
350–400	—	—	—	—	0.22%	0.22%	—	0.41%	—	0.25%	—	—	—	—	—
400–450	—	—	—	—	—	—	—	—	—	0.32%	—	—	—	—	0.56%
450–500	—	—	—	—	—	0.41%	—	0.28%	—	—	—	—	—	—	0.75%
500–1000	—	—	—	—	—	0.83%	0.15%	0.54%	—	0.72%	—	—	—	—	2.99%
1000–5000	—	—	—	—	—	1.46%	—	0.68%	—	—	—	—	—	—	—
>5000	—	—	—	—	—	0.83%	0.15%	—	—	1.12%	—	—	—	—	0.93%

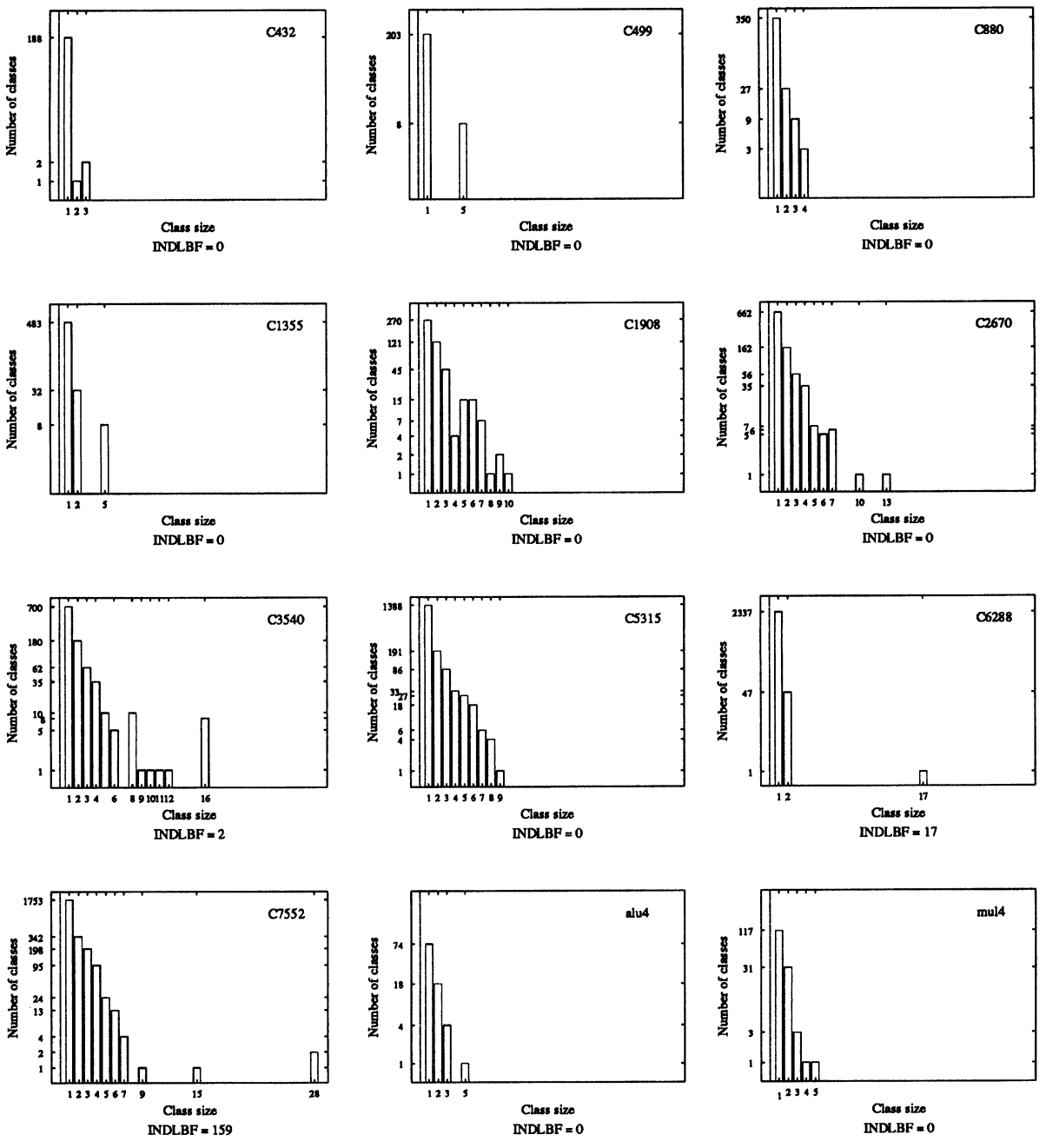

FIGURE 4 F-redundant classes of the benchmark circuits.

- Unknown f-redundancy relation. The program is not able to construct the OBDDs associated with the nodes. This result may appear in two cases:

a) The abort condition to avoid exponential execution time has been reached (for the experiments this condition has been fixed to 15 minutes).
b) Problems with the memory overflow. If an OBDD grows excessively the reserved place for its storage and handling becomes insufficient (more than 150,000 OBDD nodes).

Table I shows a summary of the results of the f-redundancy analysis for the benchmark circuits. DLBF denotes the number of LBFs that have been found detectable. UDLBF gives the number of LBFs between nodes within an f-redundancy class and, therefore, undetectable by I_{DDQ} testing. The column labeled INDLBF shows the number of LBFs whose detectability has not been determined due to excessive time or memory requirements. Finally, the *Reduced Time* normalize the total time to the number of undetectable and indeterminate LBFs. Table II presents statistics of the size (number of nodes) of the OBDDs generated to determine the f-redundancy classes.

Graphics in Figure 4 present the f-redundancy classes of the experimented circuits determined by using the proposed method. All the LBFs between nodes in the same f-redundancy class are undetectable by current testing.

From the experimentation the following facts are highlighted:

a) The proposed method seems to be efficient in the detection of f-redundancies in combinational logic structures. Only a small number of nodes remains without classification (labeled *indeterminate nodes* in Table I and Figure 3). Generally, the circuits present a large number of nodes within f-redundant classes. One cause is the use of identity gates (buffers), which increases the f-redundancy.
b) The execution times and memory requirements keep reasonably bounded. The time to execute Phase 1 grows in the order of N^2, being N the number of nodes, and the *total time per redundant node detected* (labeled *Reduced time* in Table I) has been found uncorrelated to the size N of the circuit.
c) The size of the OBDDs has been found relatively small. The higher percentage in all the circuits corresponds to OBDD sizes less than 200 nodes. Only for a few circuits, large OBDDs (more than 5000 nodes) have been necessary, and they are always a very small percentage of the total number of OBDDs.

4.1 F-redundancy Identification and I_{DDQ} Test Pattern Generator

In this subsection, the f-redundancy classes are used to help an I_{DDQ} test pattern generator for LBFs. The knowledge of the f-redundancy classes may prevent the TPG program from trying to generate a test for an undetectable LBF. On the other hand, if the number of detectable LBFs is known, the coverages can be given in terms of the detectable LBFs in addition to the usual way as a fraction of the total number of LBFs.

Table III shows the results of a TPG for the same benchmark circuits used above. The test generation method uses stuck-at vectors generated by considering current consumption monitoring with a standard ATPG program. A complete description of the method can be found in [18].

The targeted LBFs have been the set of all the LBF between logic nodes, V_{DD} and G_{ND}. Thus, if the circuit under test has N logic nodes, there are $(N + 1)(N + 2)/2$ bridges in this set. When knowing which LBF are detectable, coverage results in Table III are given in two coverage parameters:

C_D, defined as the ratio of detected LBF over the total number of possible LBF.
C_L, defined as the ratio of detected LBF over the subtraction of the number of UDLBF from the total number of possible LBF.

Another application of the identification of the f-redundancy is to transform the circuit in order to enhance the bridge testability. The next section is devoted to illustrate this application.

TABLE III I_{DDQ} Test coverage for logic bridging faults (LBFs)

Cir	LBF	#vec	LBF_D	UDLBF	C_D	C_L
c432	19503	15	19496	7	99.96%	100%
c499	29890	32	29810	80	99.73%	100%
c880	98790	23	98718	72	99.93%	100%
c1355	173166	90	173054	112	99.94%	100%
c1908	418155	107	417208	947	99.78%	100%
c2670	1018878	32	1017944	934	99.91%	100%
c3540	1480060	63	1477866	2193	99.95%	99.9999%
c5315	3091341	38	3089879	1462	99.95%	100%
c6288	3000025	34	2999825	200	99.99%	100%
c7552	6921060	54	6918139	2921	99.96%	100%
alu4	8128	12	8088	40	99.51%	100%
mul4	19701	15	19645	56	99.72%	100%
mul8	258840	27	258562	277	99.89%	99.9996%
mul12	1223830	35	1223044	771	99.94%	99.9988%
mulb	752151	25	750374	1777	99.76%	100%

LBF_D: Number of detected LBFs.
UDLBF: Number of undetectable LBFs.
$C_D = LBF_D/LBF$
$C_L = LBF_D/(LBF\text{-}UDLBF)$

5. CIRCUIT TRANSFORMATION TO INCREASE LBF TESTABILITY

If the nodes that produce the f-redundancy were removed from the circuit, the new circuit would have both C_D and C_L coverage parameters equal to the C_L coverage of the initial circuit. Therefore, the transformed circuit would have a LBF coverage equal or very near to 100%.

After the f-redundancy classes have been identified, the structure of the combinational circuit may be changed to remove some of this f-redundancy. This is done by reducing all the nodes in each f-redundant class to one unique representative node of the class.

When removing s-redundancies, the straightforward approach of identifying all redundant nodes and thereafter, removing all these nodes from the circuit, may lead to incorrect results. When a redundant region for an s-redundant fault is removed from a circuit, other s-redundant nodes may become s-irredundant and some s-irredundant nodes may become s-redundant [1, 31]. Some examples of this fact can be found in [31]. On the contrary, the removal of an f-redundant region cannot produce new f-redundant nodes.

From what has been stated above, it follows that a set of f-redundancy classes in a circuit can be removed once its identification is concluded. Only one node in each f-redundant class survives the removal process. If this node belongs to the f-redundancy region of another f-redundant node that is going to be removed, an f-redundancy class can be totally removed. The survivor node has been chosen to be the one with lower level in the f-redundancy class.

A f-redundancy removal algorithm has been used to remove the f-redundancy classes previously identified in the circuits. When performing the circuit transformation, it has been taken into account the removal process not to increase the maximum fanout of the initial circuit. This way, f-redundancies which prevent the fanout from being greater than the maximum fanout have been preserved. Table IV presents the number of lines and gates of the experimented circuits after and before the f-redundancy removal process. In this table, the number of gates equals the number of nodes minus the number of primary inputs. The number of lines equals the number of primary inputs, gate outputs and fanout branches. A circuit name with "nfr" denotes a simplified circuit. The

TABLE IV Comparison between benchmark circuits before and after the f-redundancy removal

Circuit	#gates	% gate reduction	#lines	% line reduction	time (s)
c432	160	1.9%	432	0.9%	<0.01
c432nfr	157		428		
c499	202	15.8%	499	4.8%	0.01
c499nfr	170		475		
c880	383	16.2%	880	8.6%	0.01
c880nfr	321		804		
c1355	546	11.7%	1355	4.1%	0.01
c1355nfr	482		1299		
c1908	880	49.1%	1908	37.9%	0.06
c1908nfr	448		1184		
c2670	1193	40.4%	2670	33.1%	0.08
c2670nfr	711		1786		
c3540	1669	41.9%	3540	30.3%	0.10
c3540nfr	969		2469		
c5315	2307	31.4%	5315	23.5%	0.13
c5315nfr	1605		4066		
c6288	2416	2.6%	6288	2.2%	0.03
c6288nfr	2353		6147		
c7552	3512	36.6%	7552	30.7%	0.25
c7552nfr	2225		5233		
alu4	112	28.6%	261	11.5%	<0.01
alu4nfr	80		231		
mul4	185	21.6%	404	10.9%	<0.01
mul4nfr	145		360		
mul8	694	25.5%	1512	15.4%	0.03
mul8nfr	517		1279		
mul12	1527	27.1%	3324	17.4%	0.08
mul12nfr	1113		2744		
mulb	1208	48.3%	2440	39.5%	0.03
mulbnfr	625		1475		

reduction percentage in a circuit parameter (lines or gates) was calculated by the change in this parameter divided by its initial value.

The same ATPG, whose results for the initial, circuits have been shown in Table III, was applied on the set of "nfr" circuits. Table V presents the new coverages and number of test vectors.

Preliminary experiments on some of the ISCAS'85 circuits without s-redundancy [31] have shown that removing f-redundant subcircuits may produce new s-redundancies. Conversely, if a s-redundancy removal technique is applied to a circuit that does not have any f-redundant class, some new f-redundancies may appear. This fact seems to point out that if we want the circuits not to have neither s-redundancy nor f-redundancy, both removal techniques have to be iteratively applied until the circuit does not present any type of redundant logic. This final circuit, in apart from having a reduced gate count, will not present undetectable bridges or stuck-at faults and will be highly testable. However, these circuits may violate some of the technological constraints.

6. CONCLUSIONS

Just as undetectable stuck-at faults in combinational circuits are related to some logic redundancy, so may functionally equivalent nodes cause some bridging faults to be I_{DDQ} undetectable. In this paper, this functional redundancy (called f-redundancy) has been considered and analysed. F-redundancy has been shown to be different from the usual stuck-at type

TABLE V I_{DDQ} test coverage for logic bridging faults (LBFs) after the f-redundancy removal process

Cir	LBF	#vec	LBF_D	C_D	C_L
c432nrf	18915	15	18913	99.9894%	100%
c499nrf	22578	32	22578	100%	100%
c880nrf	73153	23	731513	100%	100%
c1355nrf	137550	90	137550	100%	100%
c1908nrf	116403	107	116401	99.9983%	100%
c2670nrf	445096	32	445079	99.9962%	100%
c3540nrf	520710	63	520703	99.9987%	100%
c5315nrf	1592220	38	1592180	99.9975%	100%
c6288nrf	2847691	34	2847691	100%	100%
c7552nrf	2960961	44	2960887	99.9975%	100%
alu4nrf	4560	12	4560	100%	100%
mul4nrf	12561	15	12557	99.9682%	100%
mul8nrf	147153	23	147146	99.9952%	99.9993%
mul12nrf	661825	33	661811	99.9979%	99.9991%
mulbnrf	207046	25	207042	99.9981%	100%

LBF_D: Number of detected LBFs.
$C_D = LBF_D/LBF$
$C_L = LBF_D/(LBF\text{-}UDLBF)$.

redundancy. A method for the identification of f-redundant classes (classes of nodes that present the same logic behaviour) has been developed. This method uses OBDD graphs in order to determine the equivalence among boolean functions of nodes in candidate to f-redundant classes. Its efficiency has been evaluated on a set of benchmark circuits, including all the ISCAS'85 circuits. An approach for the removal of the f-redundant nodes has also been presented. This method has been used to remove the f-redundant nodes previously found in the set of benchmark circuits, thus increasing their LBF testability. Some results have been discussed on the possibility of using the proposed f-redundancy identification in conjunction with an I_{DDQ} TPG strategy for LBFs.

The highlights of the work can be summarized in the following points:

- By means of the comparison of OBDDs, almost all f-redundancies in the investigated circuits can be detected. Only a small percentage of nodes remains without classification. That is, it cannot be determined if they are independent nodes or if they belong to some f-redundancy classes with two or more nodes.

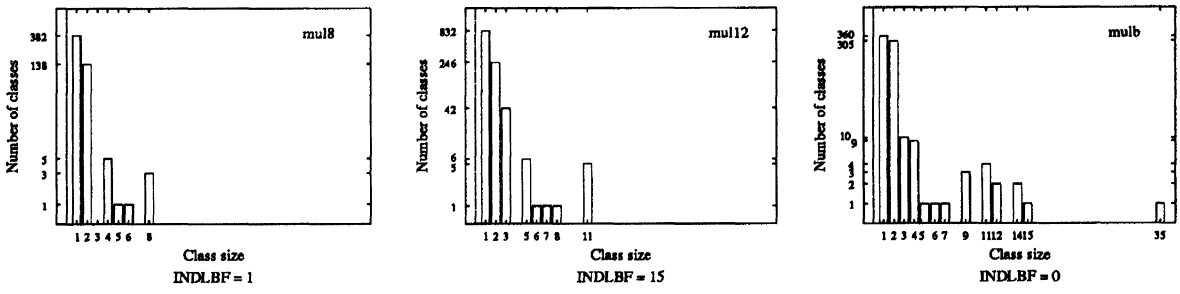

FIGURE 5 F-redundancy removal algorithm.

- The algorithm execution times and memory requirements remain reasonably bounded, although they grow with the circuit size. The size of the OBDDs needed to perform the analysis was relatively small (less than 200 nodes) except for a small percentage of the total number of OBDDs.
- The time to detect the f-redundancy between two nodes has been found to be highly dependent on the particular functionally redundant structure analysed and with no correlation with the global circuit size. This is due to the locality of the redundant logic structures in the circuits investigated.
- The knowledge of the f-redundancy classes may allow TPG strategies not to waste time in trying to generate a test for an undetectable LBF. Furthermore, if the number of detectable LBFs is known, a new coverage parameter, defined as the ratio of detected LBFs over detectable LBFs, can be used. This parameter, in addition to the classical coverage parameter that refers to all possible LBFs, may provide a deeper knowledge of the quality of a test sequence.
- The execution time for the f-redundancy removal is negligible in comparison to the identification execution time.
- In the removal process, the gate and line number reduction were relatively high in comparison to similar s-redundancy removal results [1, 31]. Nevertheless, it has to be noted that the f-redundancyremoval did not consider that some f-redundancies may be necessary in order to satisfy technology or performance requirements.

As a general conclusion it might be stated that the identification and removal of f-redundancy in combinational circuits may increase significantly the detectability of logic bridging faults using current testing.

Acknowledgment

This work has been partially supported by the project ARCHIMEDES (ESPRIT III BRA Program, No 7107) and by the CICYT project No. TIC 1024/90. The authors would like to thank Dr. W. Debany for encouraging and fruitful discussions, for providing the netlist of some of the experimented circuits (alu4, mul4, mul8 and mul12) and for validating the equivalence classes and I_{DDQ} test vectors for the circuit alu4.

References

[1] Abramovici, M. and Iyer, M. (1992). "One-pass redundancy identification and removal". *Proceedings International Test Conference*, pp. 807–815.

[2] Abramovici, M. and Menon, P. R. (1985). "A practical approach to simulation and test generation for bridging faults". *IEEE Transactions on Computers*, Vol C-34(7), (July), pp. 658–663.

[3] Abramovici, M., Miller, D. T. and Roy, R. K. (1992). "Dynamic redundancy identification in automatic test generation". *IEEE Transactions on Computer-Aided Design*, Vol 11(3), (March), pp. 404–407.

[4] Advanced Micro Devices, Inc. (ADM). (1977). *Schottky and Low-Power Schottky Data Book*, second edition.

[5] Akers, S. B. (1978). "Binary decision diagrams". *IEEE Transactions on Computers*, Vol C-27(6), (June), pp. 509–515.

[6] Brglez, F. and Fujiwara, H. (1985). "A Neural netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran". *IEEE Int'l Symp. on Circuits and Systems*. Special session on ATPG and Fault Simulation.

[7] Bryant, R. E. (1986). "Graph-based algorithms for boolean function manipulation". *IEEE Transactions on Computers*, Vol 35(8), (August), pp. 677–691.

[8] Chakravarty, S. and Thadikaran, P. (1993). "Simulation and Generation of I_{DDQ} Tests for Bridging Faults in Combinational Circuits". *Proceedings 11th IEEE VLSI test Symposium*, (April), pp. 25–32, Atlantic City.

[9] Cho, H., Hachtel, G., Jacoby, R. and Mocevunas, P. (1989). "Test pattern generation in logic optimization". *International Conference on Computer Aided Design*, (November).

[10] Debany, W. H. (1991). "Measuring the coverage of node shorts by internal access methods". *IEEE VLSI Test Symposium*, (April), pp. 215–220.

[11] Ferguson, F. J. and Shen, J. P. (1988). "A CMOS fault extractor for Inductive Fault Analysis". *IEEE Transactions on Computer-Aided Design*, Vol 7(11), (November), pp. 1181–1194.

[12] Fujita, F., Matsunaga, Y. and Kakuda, T. (1991). "On variable ordering of binary decision diagrams for the application of multi-level logic synthesis". *European Conference on Design Automation*, (February), pp. 50–54.

[13] Fujita, M., Fujisawa, H. and Kawato, N. (1988). "Evaluation and improvements of boolean comparison method based on Binary Decision Diagrams". *IEEE International Conference on Computer-Aided Design*, pp. 2–5.

[14] Galiay, J., Crouzet, Y. and Vergniault, M. (1980). "Physical versus Logical Fault Model MOS LSI Circuits: Impact on Their Testability". *IEEE Transactions on Computers*, Vol , (June), pp. 527–531.

[15] Hachtel, G. D. and Jacoby, R. M. (1985). "Algorithms for

multi-level tautology and equivalence". *International Symposium on Circuits and Systems*, Kyoto, Japan, (June).

[16] Hayes, J. P. (1976). "On the properties of irredundant logic networks". *IEEE Transactions on Computers*, Vol 25, (September 2), pp. 884–892.

[17] Isern, E. and Figueras, J. (1993). "Analysis of redundant structures in combinational circuits". *Proceedings 11th IEEE VLSI Test Symposium*, (April), pp. 21–23, Atlantic City.

[18] Isern, E. and Figueras, J. (1993). "Test Generation with High Coverages for Quiescent Test of Bridging Faults in Combinational Circuits". *Proceedings International Test Conference*, (October), pp. 73–82, Baltimore.

[19] Levi, M. W. (1981). "CMOS is most testable". *Proceedings International Test Conference*, pp. 217–220.

[20] Malaiya, Y. K. and Su, S. Y. H. (1982). "A new Fault Modeling and Testing Technique for CMOS Devices". *Proceedings International Test Conference*, pp. 25–34.

[21] Malik, S., Wang, A. R., Brayton, R. and Sangiovanni-Vincentelli, A. (1988). "Logic verification using binary decision diagrams in logic synthesis environment". *International Conference on Computer Aided design*, (November), pp. 6–9.

[22] Manich, S., Ruiz, C. and Figueras, J. (1993). "A Fault-Secure Modified Booth multiplier". *EDAC-EURO ASIC, User Forum EURO ASIC Prizes Proceedings*, pp. 180–182.

[23] Mei, K. C. Y. (1974). "Bridging and Stuck-at Faults". *IEEE Transactions on Computers*, Vol C-23, (July), pp. 720–727.

[24] Menon, P. R. and Ahuja, H. (1992). "Redundancy removal and simplification of combinational circuits". *IEEE VLSI Test Symposium*, (April), pp. 268–273.

[25] Nigh, P. and Maly, W. (1990). "Test generation for current testing". *IEEE2 Design & Test of Computers*, Vol 7(2), (February), pp. 26–38.

[26] Rajsuman, R., Malaiya, Y. K. and Jayasumana, A. P. (1987). "On accuracy of switch-level modeling of bridging faults in complex gates". *Proceedings Design Automation Conference*, pp. 244–250.

[27] Roca, M. and Rubio, A. (1993). "I_{DDQ} testing of oscillating bridging faults in CMOS circuits". *IEE Proceedings on Circuits Devices and Systems*, Vol 140(1), (February), pp. 39–44.

[28] Rodriguez-Montañés, R., Segura, J. A., Champac, V. H., Figueras, J. and Rubio, J. A. (1991). "Current vs. logic testing of gate oxid short, floating gate and bridging failures in CMOS". *Proceedings International Test Conference*, (October), pp. 510–519, Nashville.

[29] Soden, J. M. and Hawkins, C. F. (1989). "Electrical properties and detection methods for CMOS IC defects". *Proceedings of the 1st European Test Conference*, pp. 159–167.

[30] Texas Instruments (TI). (1989). *The TTL Data Book, Volume 1.*

[31] Tromp, G. and van de Goor, A. (1991). "Logic synthesis of 100-percent testable logic networks". *Proceedings of International Conference on Computer Design*, pp. 428–431.

[32] Wei, R. and Sangiovanni-Vincentelli, A. L. (1986). "PROTEUS: a logic verification system for combinational circuits". *Proceedings International Test Conference*, pp. 350–359.

Authors' Biographies

Joan Figueras received his Industrial Engineering (Electrical) degree from the Universitat Politecnica de Catalunya (U.P.C.) and he Ms.C. (Electrical) and Ph.D. (Computer Information and Control) degrees from the University of Michigan were was a Fulbright Scholar and worked in the Systems Engineering Laboratory (1968–71). He has been visiting scholar in UCLA (1979) and UCI (1992). Since (1983) is Professor at the Electronics Dept. of the U.P.C. with teaching and research responsibilities in the area of Digital Systems and VLSI design. His current research interests include Test and Design for Testability, Defect and Fault Models and new Testing and Fault Tolerance methodologies.

Eugeni Isern holds a Physics degree from the University of the Balearic Islands and is now finishing his PhD. degree at the Polytechnical University of Catalunya (U.P.C.) in Barcelona. His current research interests are in Digital Testing based on realistic Fault Models and design for Testability.