

# A New Algorithm for Global Fault Collapsing into Equivalence and Dominance Sets

A. V. S. S. Prasad  
Agere Systems  
Bangalore 560066, India  
*avssp@agere.com*

Vishwani D. Agrawal  
Agere Systems  
Murray Hill, NJ 07974, USA  
*va@agere.com*

Madhusudan V. Atre  
Agere Systems  
Bangalore 560066, India  
*mvatre@agere.com*

## Abstract

Nodes in a dominance graph represent faults of a circuit. A directed edge from node  $f_i$  to node  $f_j$  means that fault  $f_j$  dominates  $f_i$ . The equivalence of faults  $f_i$  and  $f_j$  is indicated by the presence of simultaneous edges  $f_i \rightarrow f_j$  and  $f_j \rightarrow f_i$ . When local dominance and equivalence relations are included in this graph, its transitive closure provides the collapsed fault sets. Pre-collapsed fault sets of standard cells and other logic blocks can be stored in a graph library for hierarchical fault collapsing. Examples show how more compact fault sets are obtained by using functional equivalences that can be found by analysis of small cells. Benchmark circuits c432 and c499 are used to illustrate the use of functional fault collapsing within their exclusive-OR cells.

## 1. Introduction

Fault collapsing or finding fault equivalence is an essential part of any test system. The underlying algorithms are considered to be matured, can be found in any text-book [1, 5], and are included in all ATPG and fault simulation programs [6, 10, 9, 11]. We summarize the known results below and give the motivation for the present work.

Two faults are called equivalent if they are detected by exactly the same set of tests. Functionally, this requirement translates into the faulty functions corresponding to the two equivalent faults being identical. Structural equivalence can be easily analyzed for Boolean gates. For example, all single stuck-at-0 (s-a-0) faults on the inputs and output of an AND gate are equivalent. Thus, for an  $n$  input AND gate we need to consider only  $n + 2$  faults. Similar equivalences have been worked out for other gates [1, 5]. Functional equivalences involve circuits consisting of multiple gates, and are not used in general practice

due to high complexity and the lack of suitable algorithms. Structural fault collapsing alone can reduce the fault set size to about 40 to 60% of the set of all faults.

Another form of collapsing that can further reduce the fault set size is known as dominance fault collapsing. A fault, all of whose tests detect some other fault, is said to be dominated by that other fault. For an AND gate, the output s-a-1 fault dominates any single s-a-1 fault on an input. Thus, dominance fault collapsing reduces the number of faults for an  $n$  input AND gate to  $n + 1$ .

In the equivalence collapsed set when a fault is not detected the status of the entire set of faults that is equivalent to it is known. Such is not the case in the dominance collapsed set. Still there are advantages of using the latter for ATPG.

Both equivalence and dominance relations are transitive. For example, if fault  $p$  dominates  $q$  and  $q$  dominates  $r$ , then  $p$  will dominate  $r$ . Collapsing algorithms use this transitivity property to find larger equivalence or dominance sets from the local structural relations.

Most ATPG programs use only structural equivalence fault collapsing. The Fastest program, developed at the University of Wisconsin, can do both types of collapsing, but it too uses only structural collapsing [9]. Our motivation in this work is to present a new graph-theoretic algorithm for both types of fault collapsing. First, because of the transitive nature of the relations we are dealing with, our use of transitive closure graph makes the result globally optimum and not dependent on the order in which the transitivity is exercised. Second, we can incorporate functional equivalences that are pre-computed for logic cells in a library-based design. Faults in the library cells are pre-collapsed and saved for use in hierarchical collapsing procedures.

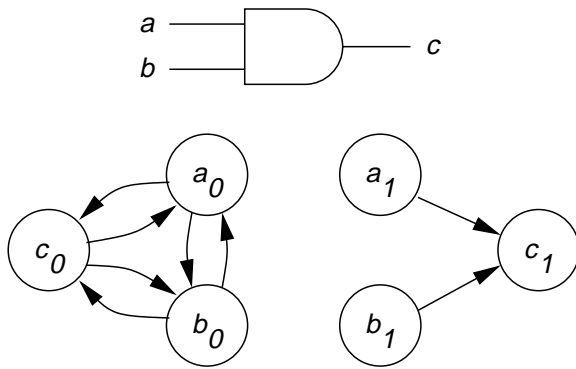


Figure 1: A two-input AND gate and its dominance graph.

	$a_0$	$a_1$	$b_0$	$b_1$	$c_0$	$c_1$
$a_0$	1	0	1	0	1	0
$a_1$	0	1	0	0	0	1
$b_0$	1	0	1	0	1	0
$b_1$	0	0	0	1	0	1
$c_0$	1	0	1	0	1	0
$c_1$	0	0	0	0	0	1

Figure 2: Connectivity matrix of dominance graph of AND gate.

## 2. A Graph Model

We represent fault equivalence and dominance relations by a directed graph. In this graph each fault is represented by a node. If a fault  $f_2$  dominates  $f_1$  then this relation is represented by a directed edge from node  $f_1$  to  $f_2$ . This edge indicates that any test for  $f_1$  must detect  $f_2$ . Clearly, the presence of edges  $f_1 \rightarrow f_2$  and  $f_2 \rightarrow f_1$  indicates that the two faults  $f_1$  and  $f_2$  are equivalent. Such graph representation of fault dominance has been used before [4].

We will call a graph representing the dominance relations among the faults of a circuit as the *fault dominance graph* or simply *dominance graph*. Figure 1 shows the dominance graph for the faults of an AND gate. We have used the subscript fault notation. Thus,  $a_0$  means the fault  $a$  stuck-at-0.

The dominance graph is conveniently represented by its connectivity matrix shown in Figure 2. This matrix contains one row and one column corresponding to each fault. A 1 entry at the intersection of a row and a column means that the fault corresponding to the column dominates the fault corresponding to the row. For example, the 1 in the second row and last column indicates that  $c_1$  dominates  $a_1$ . Equivalence of two faults is expressed by two 1's placed

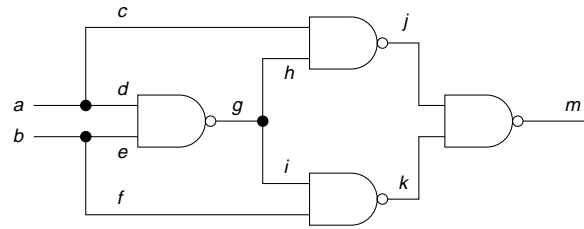


Figure 3: An xor cell.

in diagonally symmetric positions. An example of equivalence is the pair  $(a_0, b_0)$ .

## 3. Collapsing Via Transitive Closure

*Transitive closure*  $T$  of a directed graph  $G$  is another graph such that  $T$  and  $G$  have the same node set, and whenever a directed path exists from node  $x$  to node  $y$  in  $G$ ,  $T$  contains an edge  $x \rightarrow y$ . The connectivity matrix of  $T$  is known as the *reachability matrix* of  $G$ . There are efficient algorithms of computing the transitive closure and we will discuss its complexity in a later section [2, 3].

We will illustrate the present application by the example circuit of Figure 3, which implements an exclusive-OR function using four NAND gates. We also assume that it is a standard cell, which is used as a building block for designing larger circuits. This cell is sufficiently small so that we can find its functional equivalences either by Boolean analysis or exhaustive simulation of its faulty functions. Thus, we determine the following functionally equivalent fault sets:

- $(c_1, f_1)$
- $(g_1, h_1, i_1)$
- $(g_0, m_0)$

These equivalences as well as all structural equivalence and dominance relations are represented in the  $24 \times 24$  dominance matrix shown in Figure 4. The 0 entries are shown as dots for clarity. A row in this matrix corresponds to a fault, which is dominated by all faults whose columns have 1 entries in the row. Boxed entries show dominances due to the functional relationships between signals, which cannot be derived from the structure of the circuit [1, 5]. Thus, the row marked  $c_1$  on the left indicates that  $c_1$  is dominated by  $f_1$  due to functional equivalence and by  $j_0$  due to structural dominance. Of course, once added to the graph (or matrix) both functional and structural relations are treated alike.

	$a_0$	$a_1$	$b_0$	$b_1$	$c_0$	$c_1$	$d_0$	$d_1$	$e_0$	$e_1$	$f_0$	$f_1$	$g_0$	$g_1$	$h_0$	$h_1$	$i_0$	$i_1$	$j_0$	$j_1$	$k_0$	$k_1$	$m_0$	$m_1$
$a_0$	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$a_1$	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$b_0$	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$b_1$	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$c_0$	.	.	.	.	1	.	.	.	.	.	.	.	.	.	1	.	.	.	.	1	.	.	.	.
$c_1$	.	.	.	.	.	1	.	.	.	.	.	<b>1</b>	.	.	.	.	.	.	1	.	.	.	.	.
$d_0$	.	.	.	.	.	.	1	.	1	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.
$d_1$	.	.	.	.	.	.	.	1	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.
$e_0$	.	.	.	.	.	.	1	.	1	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.
$e_1$	.	.	.	.	.	.	.	.	.	1	.	.	1	.	.	.	.	.	.	.	.	.	.	.
$f_0$	.	.	.	.	.	.	.	.	.	.	1	.	.	.	.	.	1	.	.	.	.	1	.	.
$f_1$	.	.	.	.	.	<b>1</b>	.	.	.	.	.	1	.	.	.	.	.	.	.	.	.	1	.	.
$g_0$	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.	.	.	.	.	.	.	<b>1</b>	.
$g_1$	.	.	.	.	.	.	1	.	1	.	.	.	.	1	.	<b>1</b>	.	<b>1</b>	.	.	.	.	.	.
$h_0$	.	.	.	.	1	.	.	.	.	.	.	.	.	.	1	.	.	.	.	1	.	.	.	.
$h_1$	.	.	.	.	.	.	.	.	.	.	.	.	.	<b>1</b>	.	1	.	<b>1</b>	1	.	.	.	.	.
$i_0$	.	.	.	.	.	.	.	.	.	.	1	.	.	.	.	.	1	.	.	.	.	1	.	.
$i_1$	.	.	.	.	.	.	.	.	.	.	.	.	.	<b>1</b>	.	<b>1</b>	.	1	.	.	1	.	.	.
$j_0$	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.	1	.	1
$j_1$	.	.	.	.	1	.	.	.	.	.	.	.	.	.	1	.	.	.	.	1	.	.	1	.
$k_0$	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.	1	.	1
$k_1$	.	.	.	.	.	.	.	.	.	.	1	.	.	.	.	.	1	.	.	.	.	1	1	.
$m_0$	.	.	.	.	.	.	.	.	.	.	.	.	<b>1</b>	.	.	.	.	.	.	.	.	.	1	.
$m_1$	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.	1	.	.	1

Figure 4: Dominance matrix of *xor* cell (boxed numbers show functional equivalences.)

In the dominance matrix, 1's placed symmetrically about the diagonal correspond to fault equivalence. An example is the fault pair  $(d_0, e_0)$ . When only a one-way dominance exists, a 1 appears only in the upper or lower triangular part of the matrix. An example is  $c_1$  dominated by  $j_0$ .

We computed the transitive closure of the dominance matrix. Although more efficient computation is possible, it was done using the Floyd-Warshall algorithm, which is of complexity  $O(n^3)$  [7]. The result is shown in Figure 5. This matrix contains many more 1's since it includes the global implications of the relations represented in the dominance matrix. Both equivalence and dominance collapsed fault sets can be directly extracted from the transitive closure.

### 3.1. Equivalence Collapsed Fault Set

Corresponding to a fault, the transitive closure contains a row vector and a column vector of 1's and 0's. A 1 in the row vector indicates that the fault is dominated by some other fault, and a 1 in the column vector indicates that the fault dominates the other fault.

As an example, consider the fault  $f_1$  in Figure 5. The 1's in the row of  $f_1$  show that it is dominated by the set  $(c_1, f_1, j_0, k_0, m_1)$ . Examining the column under  $f_1$ , we find that it dominates the set  $(c_1, f_1)$ . Since equivalence means two-way dominance, the common faults in the two sets form the equivalence set of  $f_1$ . The transitive closure ensures that this is the largest equivalence set that can be obtained for  $f_1$  with the given dominance relations. The following algorithm finds an equivalence collapsed set.

**Algorithm Equivalence:** Begin with  $F$  (set of all faults) and  $E$  (equivalent set, initially empty). Execute the following steps until  $F$  becomes empty:

1. Arbitrarily select a fault from  $F$ .
2. Intersect (bit-by-bit logical AND) the row and column vectors of the transitive closure matrix corresponding to the selected fault.
3. The set of faults corresponding to 1's in the intersected vector is the equivalent set. Add any one fault from this set to  $E$  and delete all faults of this set from  $F$ . ■

	$a_0$	$a_1$	$b_0$	$b_1$	$c_0$	$c_1$	$d_0$	$d_1$	$e_0$	$e_1$	$f_0$	$f_1$	$g_0$	$g_1$	$h_0$	$h_1$	$i_0$	$i_1$	$j_0$	$j_1$	$k_0$	$k_1$	$m_0$	$m_1$	
$a_0$	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$a_1$	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$b_0$	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$b_1$	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
$c_0$	.	.	.	.	1	.	.	.	.	.	.	.	1	.	1	.	.	.	.	1	.	.	1	.	
$c_1$	.	.	.	.	.	1	.	.	.	.	.	1	.	.	.	.	.	.	1	.	1	.	.	1	
$d_0$	.	.	.	.	.	.	1	.	1	.	.	.	.	1	.	1	.	1	1	.	1	.	.	1	
$d_1$	.	.	.	.	.	.	.	1	.	.	.	.	1	.	.	.	.	.	.	.	.	.	1	.	
$e_0$	.	.	.	.	.	.	1	.	1	.	.	.	.	1	.	1	.	1	1	.	1	.	.	1	
$e_1$	.	.	.	.	.	.	.	.	.	1	.	.	1	.	.	.	.	.	.	.	.	.	1	.	
$f_0$	.	.	.	.	.	.	.	.	.	.	1	.	1	.	.	.	1	.	.	.	1	1	.	.	
$f_1$	.	.	.	.	.	1	.	.	.	.	.	1	.	.	.	.	.	.	1	.	1	.	.	1	
$g_0$	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.	.	.	.	.	.	.	1	.	
$g_1$	.	.	.	.	.	.	1	.	1	.	.	.	.	1	.	1	.	1	1	.	1	.	.	1	
$h_0$	.	.	.	.	1	.	.	.	.	.	.	.	1	.	1	.	.	.	.	1	.	.	1	.	
$h_1$	.	.	.	.	.	.	1	.	1	.	.	.	.	1	.	1	.	1	1	.	1	.	.	1	
$i_0$	.	.	.	.	.	.	.	.	.	.	1	.	1	.	.	.	1	.	.	.	.	1	1	.	
$i_1$	.	.	.	.	.	.	1	.	1	.	.	.	.	1	.	1	.	1	1	.	1	.	.	1	
$j_0$	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.	1	.	.	1
$j_1$	.	.	.	.	1	.	.	.	.	.	.	.	1	.	1	.	.	.	.	1	.	.	1	.	
$k_0$	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.	1	.	.	1
$k_1$	.	.	.	.	.	.	.	.	.	.	1	.	1	.	.	.	1	.	.	.	.	1	1	.	.
$m_0$	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.	.	.	.	.	.	.	1	.	.
$m_1$	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.	1	.	.	1

Figure 5: Transitive closure (reachability matrix) of *xor* cell.

Application of this algorithm to the transitive closure matrix of Figure 5 produced an equivalent collapsed set of *twelve* faults for the *xor* cell:  $a_0, a_1, b_0, b_1, c_0, c_1, d_0, d_1, e_1, f_0, g_0$  and  $j_0$ .

### 3.2. Dominance Collapsed Fault Set

For a fault the 1's in the corresponding row of the transitive closure matrix provide the dominating set. Thus, when a fault is placed in the collapsed fault set, we can assume that the entire dominating set is covered. A simple dominance collapsing algorithm is given below.

**Algorithm Dominance:** Begin with  $E$  (equivalent collapsed fault set obtained from Algorithm Equivalence). Execute the following steps:

1. Reduce the transitive closure matrix by deleting the rows and columns corresponding to all faults that are not in  $E$ .
2. All faults whose columns in the reduced transitive closure matrix have any off-diagonal 1's are

now removed from  $E$ . The remaining set  $E$  is the dominance collapsed fault set. ■

This algorithm removed  $g_0$  and  $j_0$  from the equivalence set, leaving *ten* faults.

When functional equivalence relations are not used, the sizes of equivalence and dominance sets are 16 and 13, respectively.

### 3.3. Library Description

We can now reduce the transitive closure matrix to be used when the *xor* cell is placed in a circuit. Only the rows and columns corresponding to the equivalence collapsed set are retained. In addition, faults on input and output lines, if they are not already in the collapsed set, are retained. This is necessary to provide reachability to the internal faults of the cell when the cell is connected to other cells. This matrix is called the *reduced dominance matrix*. For the *xor* cell it is a  $14 \times 14$  matrix as shown in Figure 6.

Many standard cells in the MOS technology contain non-Boolean gates like buses, pass transistors, and tristate devices. Structural dominances are dif-

	$a_0$	$a_1$	$b_0$	$b_1$	$c_0$	$c_1$	$d_0$	$d_1$	$e_1$	$f_0$	$g_0$	$j_0$	$m_0$	$m_1$
$a_0$	1	.	.	.	.	.	.	.	.	.	.	.	.	.
$a_1$	.	1	.	.	.	.	.	.	.	.	.	.	.	.
$b_0$	.	.	1	.	.	.	.	.	.	.	.	.	.	.
$b_1$	.	.	.	1	.	.	.	.	.	.	.	.	.	.
$c_0$	.	.	.	.	1	.	.	.	.	.	1	.	1	.
$c_1$	.	.	.	.	.	1	.	.	.	.	.	1	.	1
$d_0$	.	.	.	.	.	.	1	.	.	.	.	1	.	1
$d_1$	.	.	.	.	.	.	.	1	.	.	1	.	1	.
$e_1$	.	.	.	.	.	.	.	.	1	.	1	.	1	.
$f_0$	.	.	.	.	.	.	.	.	.	1	1	.	1	.
$g_0$	.	.	.	.	.	.	.	.	.	.	1	.	1	.
$j_0$	.	.	.	.	.	.	.	.	.	.	.	1	.	1
$m_0$	.	.	.	.	.	.	.	.	.	.	.	.	1	.
$m_1$	.	.	.	.	.	.	.	.	.	.	.	.	.	1

Figure 6: Fault collapsing library entry of a  $14 \times 14$  reduced dominance matrix for *xor* cell.

difficult to identify in these cases. However, cells of reasonable sizes can be exhaustively simulated to determine functional fault dominances, which can be included in the library description of the cell, in the same way as we did for the *xor* cell.

#### 4. A Hierarchical Adder Circuit

Figure 7 shows an eight-bit ripple-carry adder circuit with two levels of hierarchy. The circuit consists of eight full-adder subnetworks, which are constructed with *xor*, AND and OR cells.

Our cell library consists of the reduced dominance matrices for these cells. We first analyze the full-adder subnetwork using the reduced dominance matrices from the cell library. Using the transitive closure, we then reduce the dominance matrix of the subnetwork. Next, eight copies of this matrix are combined for the ripple-carry adder. In this way, the entire circuit is never flattened and the full-adder subnetwork data, analyzed once, is repeatedly reused. Also, functional fault equivalences, incorporated in the *xor* cell, are automatically used in the analysis of the larger circuit.

These results are shown in Table 1. The flat fault collapsing is conventional and is done by flattening the hierarchy to the Boolean gate level. The total number of faults, listed as “all faults” is counted at this level. Collapsing in this case is structural only. Equivalent collapsed faults (Equ.) were obtained by ATPG programs, Gentest [6], Hitec [10] and Fastest [9], all of which gave identical results. Dominance fault collapsing numbers (Dom.) were obtained

Table 1: Fault collapsing in 8-bit ripple-carry adder.

Circuit name	All faults	Number of collapsed faults			
		Flat structural only		Hierarchical with functional	
		Equ.	Dom.	Equ.	Dom.
<i>xor</i> cell	24	16	13	12	10
full-adder	60	38	30	30	24
8-bit adder	466	290	226	226	178

from Fastest. The same equivalence and dominance numbers were obtained when the graph method was applied to the flat gate-level circuits.

Hierarchical fault collapsing, both equivalence and dominance, were done by the graph method with functional equivalences incorporated in the *xor* cell. Functional equivalences provided smaller fault sets and we observed a 35% reduction in the CPU time over that needed for collapsing at the flat level.

#### 5. ISCAS’85 Benchmark Examples

Results on several ISCAS’85 benchmark circuits are given in Table 2. The “other program” results were obtained by Gentest [6], Hitec [10], and Fastest [9]. The results of the “graph method” of this paper are shown in boldface.

We analyzed two versions of the c432 combinational benchmark circuit. This circuit has 36 primary inputs, 7 primary outputs, and 160 gates including 18 exclusive-OR gates.<sup>1</sup> The first version is the original c432 in which exclusive-OR gates are not expanded. Here no fault collapsing across those gates is possible [5].

Since an exclusive-OR function is implemented with several gates, proper testing requires that logic level faults inside the function should also be considered. In the second version, which we call c432exp, exclusive-OR blocks have been replaced by the *xor* cell of Figure 3. Fault collapsing results are shown in Table 2. The graph method is compared with other programs, Gentest [6], Hitec [10], and Fastest [9].

In the case of equivalence collapsing we clearly see the advantage of functional equivalences that can be easily found in logic cells. That reduces the size of the fault set from 632 to 560. Similarly, the number of dominance collapsed faults is reduced from 503 to 449. Interestingly, this number is the same as the

<sup>1</sup>See website [http://www.cbl.ncsu.edu/CBL\\_Docs/Bench.html](http://www.cbl.ncsu.edu/CBL_Docs/Bench.html).

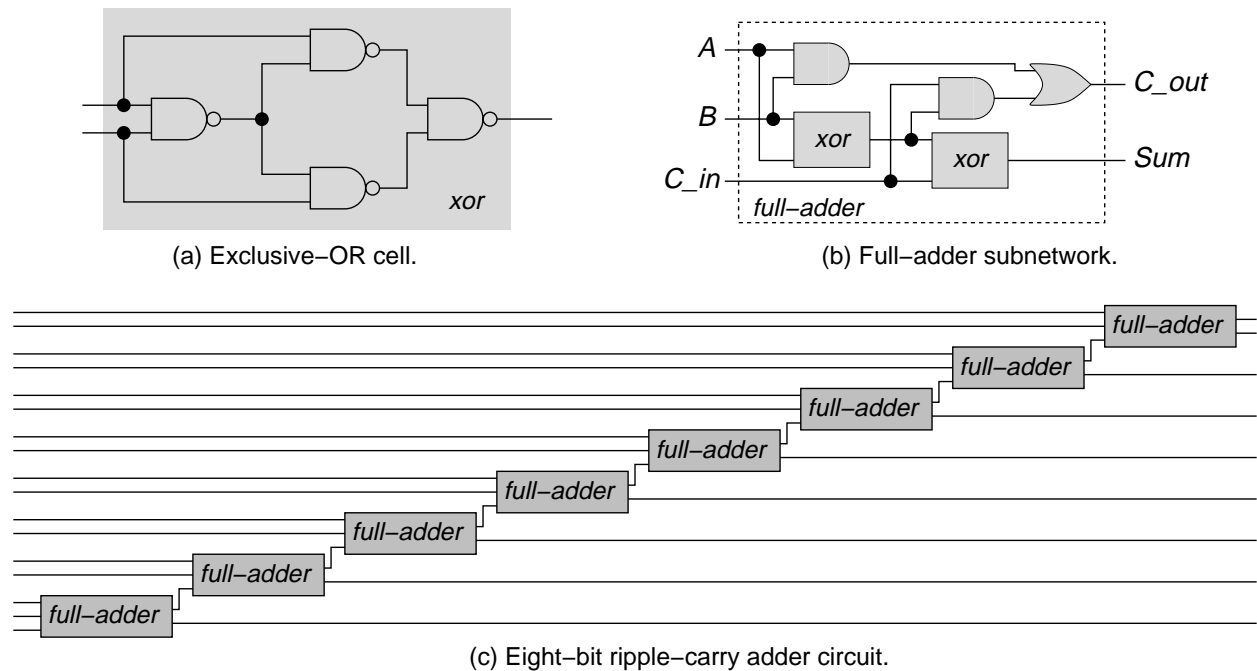


Figure 7: Hierarchical design of an 8-bit ripple-carry adder.

corresponding number for c432 with unexpanded *xor* gates.

The circuit c499 has 104 exclusive-OR gates. Expanding those as four NAND gates each gives us the circuit c499exp. Once again we observe a significant reduction in the number of collapsed faults due to functional equivalences. The circuit c1355 is the expanded version of c499. However, once we flatten the hierarchy by removing the *xor* cell boundaries, it becomes almost impossible to identify functional equivalences.

At the flat level, i.e., with the exception of c432exp and c499exp, all results of the graph method for equivalence collapsing match with those of other programs. For dominance collapsing, however, the graph method sometimes produced smaller fault sets than Fastest [9]. These differences are being investigated.

Unfortunately, ISCAS'85 circuits only provide flat description. Only in the case of *xor* we could exploit the hierarchy and functional equivalence. For “real” ASIC circuits that use standard cell libraries, once the library is characterized for functional equivalences, the advantages of the present technique can be significant.

## 6. Computational Complexity

Computationally, transitive closure is the most complex part in these procedures. The worst-case

complexity of Warshall’s matrix multiplication procedure is  $O(n^3)$ , where  $n$  is the number of nodes in the graph [3]. Also, many elements in our dominance matrix are 0. Considerable time and memory savings are possible by path tracing type of algorithms for computing transitive closure. For example, the transitive closure computations for implication graphs of logic circuits have been found to be empirically linear in the number of nodes [8]. We are currently exploring efficient implementations.

Another venue we may explore is to use the computation of strongly-connected components (SCC) of the graph instead of computing the transitive closure. An SCC is a subgraph in which every node is reachable from every other node. The equivalence collapsing problem partitions the dominance graph into SCC’s. The complexity of finding SCC’s is  $O(n + e)$  for a graph with  $n$  nodes and  $e$  edges [2].

We have shown the possibility of using the circuit hierarchy in the fault collapsing problem. Our exploration is proceeding toward building transitive closure graph library of cells and larger subnetworks. The fault lists within these blocks can be pre-collapsed, as long as certain additional faults at their boundaries are included in the library description.

Table 2: Fault collapsing for ISCAS'85 benchmark circuits.

Circuit name	Total faults <sup>2</sup>	Equivalence fault set size		Dominance fault set size	
		Graph method	Other programs	Graph method	Fastest [9]
c17	34	<b>22</b>	22	<b>16</b>	16
c432	864	<b>524</b>	524	<b>449</b>	449
c432exp	1044	<b>560</b>	632	<b>449</b>	503
c499	998	<b>758</b>	758	<b>706</b>	706
c499exp	2710	<b>1158</b>	1574	<b>898</b>	1210
c1355	2710	<b>1574</b>	1574	<b>1210</b>	1210
c1908	3816	<b>1879</b>	1879	<b>1566</b>	1566
c2670	5276	<b>2747</b>	2747	<b>2317</b>	2318
c3540	7080	<b>3428</b>	3428	<b>2786</b>	2794
c5315	10630	<b>5350</b>	5350	<b>4492</b>	4500
c6288	12576	<b>7744</b>	7744	<b>5824</b>	5824
c7552	15012	<b>7550</b>	7550	<b>6132</b>	6134

## 7. Conclusion

This paper makes two contributions. First, a global fault collapsing algorithm is presented. The graph representation of fault dominances allows consideration of all structural and many functional equivalences. To our knowledge the problem of using functional equivalences to collapse faults in large circuits has never been addressed in a practical sense. Transitive closure provides the global nature to the collapsing process. The second contribution is in the use of hierarchy for this problem. Traditionally, fault collapsing has been done after the circuit hierarchy is flattened. In our method, collapsed fault graphs can be saved and used for fault collapsing of higher level networks. Both ideas of using pre-collapsed graph libraries and of including functional equivalences in logic cells are new and hold promise in the modern design environment.

**Acknowledgment** – The authors are thankful to K. K. Saluja and Y. C. Kim of University of Wisconsin for providing the results of Fastest and Hitec programs.

## References

- [1] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design*. Piscataway, New Jersey: IEEE Press, 1990.
- [2] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*. Reading, Massachusetts: Addison-Wesley, 1974.
- [3] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *Data Structures and Algorithms*. Reading, Massachusetts: Addison-Wesley, 1987. Chapter 6.
- [4] S. B. Akers, C. Joseph, and B. Krishnamurthy, "On the Role of Independent Fault Sets in the Generation of Minimal Test Sets," in *Proc. International Test Conf.*, 1987, pp. 1100–1107.
- [5] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Boston: Kluwer Academic Publishers, 2000.
- [6] W.-T. Cheng and T. J. Chakraborty, "Gentest: An Automatic Test Generation System for Sequential Circuits," *Computer*, vol. 22, no. 4, pp. 43–49, Apr. 1989.
- [7] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge, Massachusetts: MIT Press, second edition, 2001.
- [8] V. Gaur, V. D. Agrawal, and M. L. Bushnell, "A New Transitive Closure Algorithm with Application to Redundancy Identification," in *Proc. 1st Int. Workshop on Electronic Design, Test and Applications (DELTA'02)*, (Christ Church, New Zealand), Jan. 2002, pp. 496–500.
- [9] T. P. Kelsey, K. K. Saluja, and S. Y. Lee, "An Efficient Algorithm for Sequential Circuit Test Generation," *IEEE Trans. Computers*, vol. 42, no. 11, pp. 1361–1371, Nov. 1993.
- [10] T. M. Niermann and J. H. Patel, "Hitec: A Test Generation Package for Sequential Circuits," in *Proc. European Design Automation Conf.*, Feb. 1991, pp. 214–218.
- [11] Synopsys, Inc., 700 East Middlefield Rd., Mountain View, CA 94043, *TetraMAX ATPG User Guide*, v2000.11 edition, November 2000. Document Order Number: 37043-000 TBD.

<sup>2</sup>Some numbers differ from other published data due to modeling differences.