Accelerated Test Pattern Generation by Cone-Oriented Circuit Partitioning

T. Grüning, U. Mahlstedt, W. Daehn, C. Özcan

Institut für Theoretische Elektrotechnik, Universität Hannover
Appelstr. 9A, D-3000 Hannover 1

ABSTRACT

In this paper an efficient cone oriented circuit partitioning method is presented, which significantly speeds up automatic test pattern generation for combinational circuits. The advantages gained by the proposed partitioning method are based on the increase in the number of dominators in the circuit graph. In contrast to conventional ATPG working on the unpartitioned circuit test generation is less time consuming now and redundancies can often be identified without any backtracks. Experimental results illustrate the superiority of the cone oriented partitioning approach. Independent of the underlying ATPG algorithm the cone oriented partitioning results on average in a performance increase by more than a factor of 2.

1. INTRODUCTION

To simplify the process of test pattern generation, methods have been developed /1,2,3/ which reduce the sequential test problem to a combinational problem. Although test generation for combinational circuits is less complicated than for sequential circuits the problem of ATPG is still alive.

Most of existing state of the art algorithms /4,5,6,7/ are based on the D-Algorithm proposed by Roth /8/. All of them adopt a branch-and-bound technique **on the whole, i.e. unpartitioned, circuit** to implicitly but exhaustively examine all input combinations. Since the search space can be quite large, concepts have been developed which reduce this search problem. In this manner the test generation problem becomes more tractable. Algorithms like FAN, SOCRATES and TOPS /5,6,7/ use the concept of dominators /10/ to guide and speed up the test pattern generation process. Thus in each phase of the process as many signal values as possible can be uniquely determined so that unnecessary backtracking does not occur.

In built-in verification testing the partitioning of a large circuit into subcircuits is an appropriate method to reduce the test time /9/. This method is based on the hope that every output pin of a combinational circuit is not a function of all input pins. So the total amount of time needed for a complete test of all subcircuits is less than the time needed for the unpartitioned circuit.

Our idea is to show that the partitioning of a large circuit is an appropriate method not only to shorten the time needed for built-in verification testing but also for deterministic test pattern generation. In contrast to verification testing the speed up of the ATPG process can be expected even if a circuit output depends on all inputs.

In the next chapter we will show the benefits of the cone oriented partitioning method on the circuit properties. The third chapter deals with the resulting enhancement of redundancy identification. In the last chapters we will present a brief description of CONTEST, our cone oriented ATPG system, give some experimental results and finish with a short conclusion.

2. CIRCUIT PARTITIONING AND ITS INFLUENCE ON DOMINATORS

The test pattern generation problem is characterized as a search problem which falls into the class of NP-complete problems. To avoid unnecessary search all the nodes which must be set unambiguous to excite and propagate the fault to any primary output (PO) must be found. For example to excite a s/0 fault at one intput of an AND gate the input must be controlled to the opposite value. To propagate the fault effect all other inputs must be set to 1.

One concept to find these nodes is based on the theory of dominators /10/ in flow graphs. By definition the dominators of a faulty node consists in the faulty node itself and all nodes which the fault effect must go through to reach an observable primary output with no respect of the chosen path. These relationships are clarified by means of the example in figure 1. To reach any primary output starting at node 2 we must pass node 14 and node 17. Therefore the dominators of node 2 are node 2 itself and nodes 14 and 17. Node 7 has no dominator except for itself since there is no node which must be passed unambiguous on the way to a PO.

2024/90/0000/0418$01.00©1990 IEEE

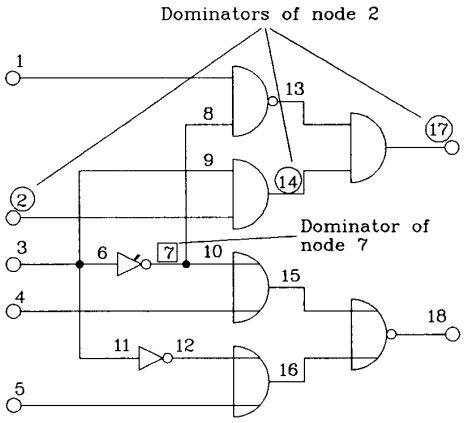

Fig. 1: Example circuit graph

Since typically there are many primary outputs to which the fault effect can be propagated the number of dominators is rather small and therefore only few nodes can be set uniquely. So our idea is to increase the number of dominators by a suited partitioning technique. This can be achieved by a cone oriented circuit partitioning. A cone is the fraction of the combinational logic network which includes all signal lines affecting one output. It is determined by a backward traversal through the circuit graph starting at the respective output and ending at the primary inputs which control this output.

To show the influence of the partitioning method on the number of dominators we will introduce the notation of the "dominator degree" and the "average dominator degree". The dominator degree of a circuit node x is defined as

$$d(x) = \text{number of dominators of node x.}$$

The dominator degree can be considered as a guideline for the number of mandatory node assignments to be performed: proportional to the number of dominators the number of uniquely determined node assignments will increase, thus speeding up the test pattern generation process.

The average dominator degree of a circuit is defined as

$$d_{av} = \frac{\sum_{i=1}^{n} d(i)}{n}$$

where n is the number of circuit nodes. The value d_{av} reflects the correlation between the number of dominators and mandatory node assignments corresponding to the whole circuit.

To illustrate the benefits of the cone oriented partitioning method look again to faulty node 7 in figure 1. Considering the circuit as a whole the only dominator is node 7 itself.

Now consider figure 1a. The bold lines mark the cone belonging to primary output 17 of our example circuit. The partitioning causes the propagation of the fault effect through the nodes 8, 13 and 17. Thereby the dominator degree is by a factor of four greater than for the unpartitioned circuit.

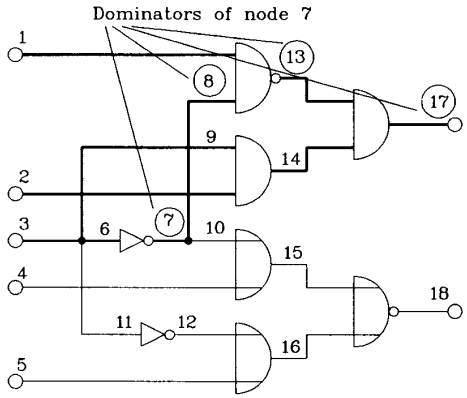

Fig. 1a: Circuit graph after partitioning

The influence of the circuit partitioning on the ISCAS'85-Benchmark circuits is shown in figure 2. It can be seen that on average d_{av} for the partioned circuit (hatched bars) is by a factor of 2.1 greater than for the unpartitioned circuit (black bars).

3. CIRCUIT PARTITIONING AND ITS INFLUENCE ON REDUNDANCY IDENTIFICATION

In many cases when the cone oriented circuit partitioning is employed redundant faults can be identified simply by assigning only the mandatory signal values to the fault site and its dominators. Leaving the circuit unpartitioned massive backtracking would be required to identify a redundant fault.

Assuming a s/0 fault at node 7 in figure 1. To provide a faulty signal at the fault site node 3 must be controlled to 0. Since there are no other dominators no further mandatory assignments can be made.

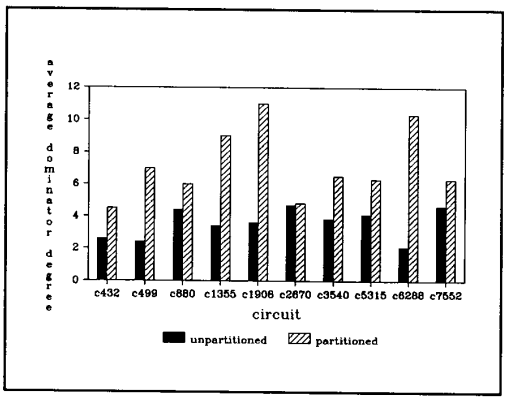

Fig 2: Influence of the cone-oriented partitioning on the ISCAS'85 Benchmark Circuits

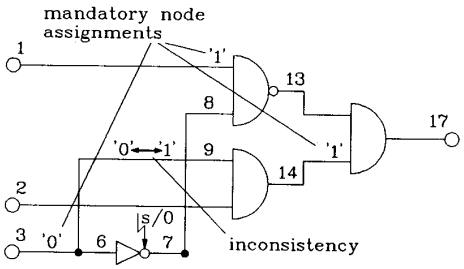

Fig. 3a: Cone belonging to primary output 17

Now take a look at figure 3a and 3b which show the cones belonging to the primary output 17 and 18 respectively. Based on the partitioning we get node 13 and node 17 in the upper cone as dominators of the faulty node. To propagate the fault effect the dominator inputs, which cannot be reached from the fault site must be set to the non controlling value. Therefore a 1 is assigned to both node 1 and node 14. Performing the implication reveals an inconsistency at node 9. In this manner the fault has been identified as redundant in the considered cone since no backtrack choice exists. For the second cone we get node 15 and 18 as dominators of the fault site. Performing the sensitization and implications an inconsistency occurs at node 12. Therefore the fault is redundant in this cone too.

To identify a 'global redundancy' a fault must be identified as 'local' redundant in all cones. Due to the greater number of dominators in the partitioned case this task can be performed often without backtracking and therefore faster than in the unpartitioned case.

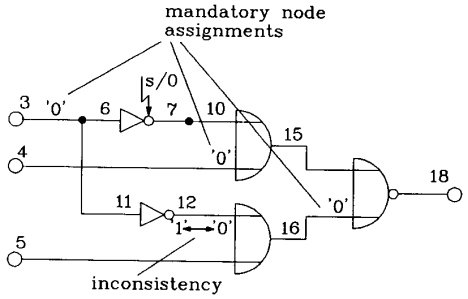

Fig 3b: Cone belonging to primary output 18

4. EXPERIMENTAL RESULTS

To investigate the approach of cone oriented test pattern generation CONTEST, a FAN-based algorithm using nine-valued logic /11/, has been implemented on an Apollo DN4000 workstation. Figure 4 scetches the program flow in principle.

```
FOR ALL (primary outputs)
{
        mark_cone ()
        FOR ALL (faults in cone not yet detected)
        {
                CONTEST ()
                fault_dropping ()
        }
}
```

Fig 4: Principal program flow

To gather the results of the circuit oriented test pattern generation we modified our cone oriented approach in a simple way. This was achieved by inserting a fault transparent 'dummy'-gate at the primary outputs of a circuit.

The performance of the cone oriented approach was compared with the conventional approach using the ten ISCAS'85 benchmark circuits /12/. The following experiments have been performed:

1) Conventional circuit oriented test pattern generation without partitioning.
2) Cone oriented test pattern generation.

In all cases the backtracking limit for test pattern generation was set to 10 and as soon as a test pattern was generated a fault simulation was carried out to drop out not yet detected faults.

Table 1 shows the results of the experiments. The performance comparison will be depicted in terms of three parameters: the number of aborted faults, the number of identified redundancies and the CPU-time needed for test pattern generation.

Circuit	Faults	cone oriented			circuit oriented		
		abo	red	time[s]	abo	red	time[s]
c432	463	3	1	7	8	1	18
c499	708	3	8	21	23	8	24
c880	815	0	0	3	0	0	4
c1355	1316	0	8	82	6	8	157
c1908	1684	12	7	83	26	7	161
c2670	2334	34	81	171	29	86	124
c3540	3020	0	135	111	1	135	310
c5315	4903	7	59	27	6	59	130
c6288	6294	0	36	186	2	36	435
c7552	6793	38	93	284	63	69	678

abo – # faults aborted red – # identified redundancies

Table 1: Comparison between cone-oriented and circuit-oriented test pattern generation

Table 2 shows how many of the redundancies got in experiment 2 could be identified without any backtracks.

Circuit	red	red with 0 backtracks
c432	1	1
c499	8	8
c1355	8	8
c1908	7	7
c2670	81	77
c3540	135	127
c5315	59	59
c6288	36	36
c7552	93	81

Table 2: Redundancies identified without any backtracks

The results clearly underline the superiority of the cone oriented approach. The cone oriented approach yields for eight of ten circuits the same or a lower number of aborted faults than the conventional circuit-oriented approach. With respect to the number of identified redundancies there is no significant difference between the two approaches but in the cone oriented approach nearly 100 % could be identified without any backtracks. For nine circuits the tests could be generated faster than for the unpartitioned circuit. On average the test generation is accelerated by a factor of 2. The same speed up has been measured for an alternative implementation of a PODEM-algorithm illustrating the general applicability of the cone oriented partitioning method.

5. CONCLUSION

An efficient cone oriented circuit partitioning method has been presented, significantly improving the performance of existing ATPG algorithms. The cone oriented partitioning increases the number of dominators by a factor of approximately 2 thus resulting in a faster test generation and redundancy identification. Compared to conventional ATPG CONTEST, an implementation of the cone-oriented approach, performs better by an average factor of 2.

Current work is concerned with making use of structural and functional information to further improve test pattern generation and redundancy identification.

6. REFERENCES

/1/ Williams, M.J.Y., Angell, J.B.: "Enhancing Testability of Large Scale Integrated Circuits via Test Points and Additional Logic", IEEE Trans. Comput., C-22 (1), pp. 46-60, 1973

/2/ Kobayashi, T., Matsue, T., Shiba, H.: "Flip-flop Circuit with FLT Capability", Proc. IECEO, p. 692, 1968

/3/ Eichelberger, E.B., Williams, T.W.: "A Logic Design Structure for LSI Testability", Proc. 14th Design Automation Conference, pp. 462-468, 1977

/4/ Goel, P.: "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic", IEEE Trans. Comput., Vol. C-30 (3), pp. 215-222, 1981

/5/ Fujiwara, H., Shimono, T.: "On the Acceleration of Test Generation Algorithms", Proc. 13th Int. Symp. Fault-Tolerant Computing, pp. 98-105, 1983

/6/ Schulz, M.H., Auth, E.: "Advanced Automatic Test Pattern Generation and Redundancy Identification Techniques", Proc. 18th Int. Symp. Fault-Tolerant Computing, pp. 30-35, 1988

/7/ Kirkland, T., Mercer, M.R.: "A Topological Search Algorithm for ATPG", Proc. 24th Design Automation Conference, pp. 502-508, 1987

/8/ Roth, J.P.: "Diagnosis of Automata Failures: A calculus and a method", IBM Journal Res. Dev. 10, pp. 278-281, 1966

/9/ Mc Cluskey, E.J.: "Verification Testing", Proc. Intern. Test Conf. 1982, pp. 183-190, October 1982

/10/ Tarjan, R.: "Finding Dominators in Directed Graphs", SIAM Journal of Computing, Vol. 3, pp. 62-89, 1974

/11/ Muth, P.: "A Nine-Valued Logic Model for Test Generation", IEEE Trans. Comput., Vol. C-25, pp. 630-636, 1976

/12/ Brglez, F., Fujiwara, H.: "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran", special session on ATPG and fault Simulation, Proc. 1985 IEEE Int. Symp. on Circuits and Systems, Kyoto (Japan), 1985