Survey for Stuck At Fault Modelling of Digital Circuits At Register Transfer Logic (RTL)

Ms. Naina A. Udge
Assistant Professor
Faculty of Electr.&Telecom.
Sipna COET Amravati (M.H.)

Dr. S. A. Ladhake
Principle
SipnaCOET,Amravati(M.H.)

Prof. P. D. Gawande
AssosicateProfessor
Faculty of Electr.&Telecom.
SipnaCOET,Amravati(M.H.)

Abstract

This research aims to testing of digital circuits. By using fault models at the lower levels, testing becomes cumbersome and will lead to delays in the design cycle. Thus there is a need to look for a new approach of testing the circuits at higher levels to speed up the design cycle. Different methods are implemented to detect the faults such as stuck-at faults in RTL circuits. A digital circuit usually comprises a controller and datapath. The time spent for determining a valid controller behavior to detect a fault usually dominates test generation time. A validation test set is used to verify controller behavior and, hence, it activates various controller behaviors. In this paper, different methods are presentedfor detecting faults in the datapath thus, resulting in the detection of a majority of stuck-at faults in the datapath RTL modules.

Keywords-*stuck-at faults; fault coverage;datastructure;validation test sets*

1.Introduction

There are following faults in the chip namely Single stuck-at faults;Transistor open and short faults;Memoryfaults;PLA faults (stuck-at, cross-point, bridging);Functional faults (processors);Delay faults (transition, path);Analog faults.Three properties define a single stuck-at fault :

(i) Only one line is faulty
(ii) The faulty line is permanently set to 0 or 1
(iii) The fault can be at an input or output of a gate.

Advances in semiconductor and electronic design automation technology have increased the size and complexity of VLSI circuits. Various invasive [design for testability (DFT)] and noninvasive [software-based self-test (SBST), built-in self-test (BIST)] strategies have been suggested to reduce test generation time and improve fault coverage. While the invasive techniques tend to alter the characteristics of the design in terms of area and timing, the noninvasive techniques suffer from low fault coverages and/or large test application times. Validation/functional testing methods start with a functional description of the circuit and make sure that the circuit's operation corresponds to its description. various methods are geared towards extensive validation of the controller behavior of a given circuit under test.

2.Related Work

Various methods are implemented to detect the stuck-at fault in digital circuits.

A. Adding Buffer to each ports of RTL Circuits.

The very first method is adding buffer to each ports of RTL circuits to create a new faulty circuit[7].

1. Firstly test bench is developed and the simulation is first run on a good circuit and then on each of the faulty circuits using any simulator.

2. The outputs obtained in each case of the faulty circuits are compared with the output of the good circuit to determine which faults are detected. That is the new faulty circuit and the fault free circuit is simulated and the outputs so obtained are compared. The fault list is tabulated.

3. The ratio of the numbers of RTL faults detected to the total number of RTL faults gives the RTL fault coverage.

In this work Verilog Hardware Description Language is used for writing the RTL models. The basic assumption is that the components are fault free and

only their interconnections are affected. These map to the operators and variables in the RTL descriptions respectively. Gate level primitives can be instantiated in a model using gate instantiation as these are supported for synthesis. These primitive gates describe the hardware. Therefore synthesizing a gate primitive generates logic based on the gate behavior which eventually gets mapped to the target technology .Based on this the single stuck-at fault is modeled.The assumption is also that at most one fault occurs at a time in the circuit.The most common model used for logical fault is the single stuck-at fault (SSF). In this a fault in a logic gate gives a favourable outcome in one of its inputs or the output being fixed to either a logic 0(stuck-at-0) or a logic 1(stuck-at-1).

Table 1.RTL Versus Gate-Level Fault Coverage

Name of the circuit	RTL Fault Coverage	Gate-Level Fault Coverage
Full Adder	100%	100%
Multiplexer	100%	100%
Decoder	100%	100%
Comparator	100%	100%
Priority Encoder	100%	100%
JK flip-flop	100%	100%
D flip-flop	100%	100%

B. Using Validation Test Sets.

Next method is that in which validation test sets are used to generate test sequences that detect a majority of stuck-at faults in the datapath[1].

1. The scheme first derives the controller behaviors from validation test sequences and reuses them for simplifying justification/propagation analysis corresponding to precomputedtest vectors/responses of datapath RTL modules.

2. A heuristic is used to identify controller behaviors that are compatible with a given set of precomputed test vectors/responses. It requires only a single pass through the CDFG corresponding to a validation test sequence and is accu- rate, resulting in a small number of test generation runs.

3. Test generation is performed at the RTL and the controller behavior is prespecified,which results in very small ltest generationtimes.First step is identification of compatible controller behaviors consisting of Augmented Fault Simulation to Derive Activation-Detection Time Frame Pair and Analysis of Requirements to Identify Compatible Faults. Next step is SAT-based RTL ATPG is used to obtain a test sequence that reuses the controller behavior to justify and propagate the precomputed test vector and response to primary inputs and outputs, respectively.

SAT- based RTL ATPG uses an ILA model of the circuit under test.The circuit is unrolled for a predetermined number of cycles determined by the number of vectors in the validation test sequence from which the controller behavior is extracted. Test generation is performed on the entire circuit description comprising the controller and datapath by first identifying the paths from the inputs of the module under test to primary inputs and from the output of the module under test to primary outputs in the ILA model. These paths are then translated into Boolean clauses by translating the functionality of the individual modules in these paths. Once the Boolean clauses that capture the RTL test generation problem and the controller behavior are generated, a SAT solver is invoked to resolve these clauses. If the solver returns with a satisfiable solution, then a test sequence can be extracted from the Boolean variable assignments corresponding to the primary inputs. This sequence is guaranteed to deliver the precomputed test vector to the inputs of the corresponding RTL module and propagate the fault effect from the output of the module to a primary output. If the Boolean clauses are not satisfiable, then test generation fails, indicating that the targeted precomputed test vector/response cannot be justified/propagated by reusing the controller behavior that was found to be compatible by using the heuristic.

Table 2. Reduction In Number Of Test Generation RunsDue To TheProposedHeuristic

Circuit	Number of test generation runs	
	Without heuristic (worst case)	With heuristic
Sqr_mul	744	79
Sqr_add	1024	75
ex1	3605	156
Bessel	2133	105
Paulin	6027	193
ASPP4	6820	238
Parwan	3198	170

C. **Implementation of Automatic Test Paterrn Generation**.

Next method for detection of stuck-at fault consisting of an algorithm for generating test patterns automatically from functional register-transfer level (RTL) circuits that target detection of stuck-at faults in the circuit at the logic level.

1. In order to do this, a data structure named assignment decision diagram are used[5]. The algorithm is very versatile and can tackle almost any type of single- clock design, although performance varies according to the design style. The first step is to convert each process and concurrent RTL statements in each leaf component of the circuit into ADDs.
2. After that, each ADD is selected and its I nternals targeted for testing. First the arithmetic operations are targeted, then logic arrays, then untagged registers, latches, and memories, then untagged ADNs, and finally random logic blocks and black boxes.

During testing of each RTL element, a nine-valued symbolic RTL justification and propagation is done to trace outpaths from the PIs to the element inputs and element outputs to POs to obtain a symbolic test environment for the module. The search is a branch and bound type of search with backtracking and has a backtrack limit and search time limit that may be adjusted. It requires hierarchy traversal in case of a hierarchical design. The transformations across operations are based on the nine-valued RTL algebra and placed in a look-up table. If a search fails to obtain a test environment, then some heuristics are used to increase the coverage. Once the test environment is found, a precomputed test set is used (only for arithmetic operations) from a test set library to get a system-level test set for the RTL element. All system-level test sets for all RTL elements are concatenated together to get the complete RTL circuit test set

Table 3.Test Generation Results

Circuit	*HITEC*			*STRATEGATE*			*RTL ATPG*		
	FC (%)	TGen (sec)	TApp (cycles)	FC (%)	TGen (sec)	TApp (cycles)	FC (%)	TGen (sec)	TApp (cycles)
Paulin	97.92	147002	752	99.71	101071	4499	99.72	138	4124
Tseng	98.43	52139	366	99.63	18481	2689	99.68	216	3429
Dct	90.01	74805	1696	89.83	106056	2528	96.50	739	3965
GCD	49.16	43964	258	85.73	192384	55823	94.31	498	4568
Barcode	63.41	9799	759	57.58	232336	24764	88.78	876	4080
X25	36.31	93592	151	57.27	131897	20817	85.35	1046	3561
Am2910	73.86	18723	1317	94.48	15125	4742	95.32	2765	3952
GPIO	99.41	57	1396	98.30	11078	5292	93.56	5543	690
ALM	22.53	58600	589	29.53	67854	1563	36.52	85654	1430
EXE	21.32	27300	992	44.12	399333	26722	40.83	585700	5689

3. Conclusions

It isobservedthat the RTL FaultCoverageobtained by the very first faultmodelingmethodology has a close match to the Gate-levelFaultCoverage for the tested digital circuits.A novel approach is presented for using a validationtest set to generate test sequences that have good stuck-atfault coverage for datapath RTL modules. The scheme first derivesthe controller behaviors from validation test sequences andreuses them for simplifying justification/propagation analysiscorresponding to precomputed test vectors/responses of datapathRTL modules. A heuristic is used to identify controller behaviors that are compatible with a given set of precomputedtest vectors/responses. It requires only a single pass through the CDFG corresponding to a validation test sequence and is accurate,resulting in a small number of test generation runs. Testgeneration is performed at the RTL and the controller behavioris prespecified, which results in very small test generation times.

A versatileRTL-ATPG algorithm is presented that can generate test vectors for almostany type of single-clock functional RTL design. The algorithmuses a data structure called ADD that helps it to tackle controland data flow in an unified fashion and a nine-valued algebrathat helps it to do justification and propagation at the RTL. Thealgorithm degenerates to an inefficient logic-level ATPG algorithmif it is fed a Boolean network. The performance of the algorithm degenerates as thecircuit description becomes more and more logic type. Currentefforts are to map effective logic-level ATPG heuristics into thealgorithm so that the performance is comparable to logic-levelATPG even for logic type designs.

Finally analysing each and every methods which are mentioned above on the basis of fault coverage percentage, a suitable method will be implemented to improve the fault coverage of stuck-at fault to greater extend.

4. ACKNOWLEDGMENT

This Reasearch is done under the guidance of Dr. S.A. Ladhake, Principle,Sipna COET Amravati (M.H.) and Prof.P.D.Gawande, Associate Professor, Department Of Electronics and Telecommunication, Sipna COET Amravati (M.H.)

5. REFERENCES

1) *R. C. Ho and M. A. Horowitz*, "Validation coverageanalysis for complex digital designs," in Proc. Int. Conf. Comput.-Aided Des., Nov. 1996.

2) *I. Ghosh, A. Raghunathan, and N. K. Jha*, "A design for testability technique of RTL circuits

using control/data flow extraction," in Proc. Int. Conf. Comput.-Aided Des., Nov. 1996.

3) *D. J. Moundanos, J. A. Abraham, and Y. V. Hoskote*, "Abstraction techniques for validation coverageanalysis and test generation," IEEE Trans. Comput. Jan. 1998.

4) *IndradeepGhosh and Srivaths Ravi,* "On AutomaticGeneration of RTL Validation Test BenchesUsing Circuit Testing Techniques" Fujitsu Laboratories of America, Sunnyvale, CA 94086,NECLaboratoriesAmerica, Princeton, NJ 08540, 2001.

5) *IndradeepGhosh and MasahiroFujita*, "Automatic Test Pattern Generation for FunctionalRegister-Transfer Level Circuits UsingAssignmentDecisionDiagrams" ieee transactions on computer-aided design of integrated circuits and systems, vol. 20, no. 3, march 2001.

6) *L. Lingappan and N. K. Jha*, "Unsatisfiabilitybasedefficient design for testability solution for register-transferlevel circuits," in Proc. VLSI Test Symp., May 2005.

7) *SumaM.S. ,K.S.Gurumurthy "*FaultCoverage for digital circuits at RTL*"2011*.

8) SarveshPrabhu, Michael S. Hsiao, LoganathanLingappan and VijayGangaram, "A SMT-based Diagnostic Test GenerationMethod for Combinational Circuits" ieee 30th vlsi test symposium (vts) 2012.