ATPG System and Test Genaration Methods for Digital Devices

Vladimir Hahanov Olga Skvortsova Irina Sysenko Hayk Chamyan
Kharkov National University of Radio Electronics,
Lenin ave 14, 61166, Ukraine, hahanov@kture.kharkov.ua

Key words - fault simulation, digital systems, genetic algorithms, deductive method, ATPG

Abstract

Models and methods of digital circuit analysis for test generation and fault simulation are offered. The two-frame cubic algebra for compact description of sequential primitive element (here and further, primitive) in form of cubic coverings is used.Problems of digital circuit testing are formulated as linear equations. The described cubic fault simulation method allows to propagate primitive fault lists from its inputs to outputs; to generate analytical equations for deductive fault simulation of digital circuit at gate, functional and algorithmic description levels; to build compilative and interpretative fault simulators for digital circuit. The fault list cubic coverings (FLCC) allowing to create single sensitization paths are proposed. The test generation method for single stuck-at fault (SSF) detection with usage of FLCC is developed. The means of test generation for digital devices designed in Active-HDL are offered. The input description of design is based on usage of VHDL, Verilog and graphical representation of Finite State Màchine (FSM). The obtained tests are used for digital design verification in Active-HDL. For fault coverage evaluation the program implementation of cubic simulation method is used.

Also models and procedures of test generation for digital systems verification based on genetic algorithms are offered. Program implementation of test generator is oriented to processing of digital circuits with large dimensionality and it combines advantages of genetic algorithms (high speed operation of test generation with specified fault coverage) with advantages of deterministic ones (reproducibility of generated test sequences).

1 Introduction

Field Programmable Gate Arrays (FPGA) and Complex Programmable Logic Devices (CPLD) make a deserved competition to microprocessor chips. Such success is defined by usage of Hardware-Software Co-operation Design, minimum time of digital system design (4-5 months), high-speed operation (under 500 MHz), high level of gate-array chip integration.

However, there are testing problems together with advantages of CPLD (FPGA). For solving these problems it is required to create models, methods and CAD software. The mentioned means have to support:

1) digital device testing at gate, functional and algorithmic description levels, when the digital device has a high-level integration and it is specified as FSM transition graphs, Boolean equations, multi-level hierarchical structures;
2) test generation for SSF detection with fault coverage about 100%, where the test has a form of single sensitization path cubic coverings;
3) acceptable operation speed of fault simulation algorithms;
4) design verification and diagnosis for synthesis into FPGA, CPLD;
5) possibility of concurrent execution of vector operation for test generation and fault simulation;
6) VHDL standard support for digital circuit and obtained test description;
7) opportunity of the integration into existing CAD systems of world-wide leading firms (ALDEC etc.).

Deductive fault simulation method [1,2] is more preferable because of its high-speed operation. It allows to detect all SSFs by input test-vector during one iteration of digital circuit processing. But this method is oriented to the gate level of digital circuit description. It is connected with complexity of output fault list generation for non-gate primitives. The offered cubic fault simulation method allows to process digital circuits described at gate, functional and algorithmic levels. In the other side, the solution of the mentioned problem is presented as the method of test generation for SSF detection with usage of FLCC allowing to create single sensitization path.

2 Mathematical apparatus of primitive analysis

FSM model of sequential primitive is: M=<X, Y, Z, f, g>, where $X=(X_1,X_2,...,X_i,...,X_m)$, $Y=(Y_1,Y_2,...,Y_i,...,X_h)$, $Z=(Z_1,Z_2,...,Z_i,...,Z_k)$ are sets of input, internal and output State variables. The primitive is described by the cubic covering

$$C = (C_1, C_2, ..., C_i, ..., C_n), \quad (1)$$

where $C_i = (C_{i1}, C_{i2}, ..., C_{ij}, ..., C_{iq})$ is a cube including input, internal and output coordinates $C_i = (C_i^X, C_i^Y, C_i^Z)$, q=m+h+k.

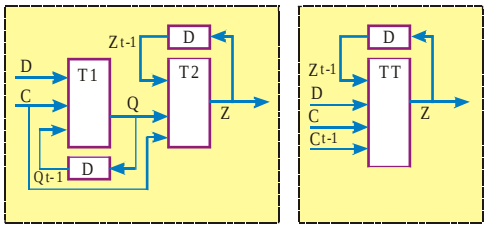

Fig.1. FSM model of flip-flop and latch

The main feature of suggested models is compactness of truth- and transition tables for complex functional primitives and FSM descriptions; universality and completness of table

models for solving problems of forward propagation and backward implication; universality and simplicity of cubic model analysis algorithms for: deterministic test generation, fault-free and fault simulation.

3 Fault simulation based on cubic algebra

Let's consider the model W= (M,L,T), where M is a primitive model represented by cubic covering C, L is a fault list cubic covering (FLCC), T is a test. Problems of digital circuit testing are formulated on condition that one of components is not defined.

FLCC L for the vector T and the primitive covering C is computed by a linear equation

$$T \oplus C = L, \qquad (2)$$

where $\oplus$ – is a binary operation XOR, which determines interaction of components T, C, L in the three-valued alphabet:

$$T_j \oplus C_{ij} = \begin{array}{|c|c|c|c|}\hline \oplus & 0 & 1 & X \\ \hline 0 & 0 & 1 & X \\ \hline 1 & 1 & 0 & X \\ \hline X & X & X & X \\ \hline \end{array}. \qquad (3)$$

The universal formula of FLCC analysis can be obtained as a result of application of (3) to the test-vector T and to covering of the multi-output primitive C. The mentioned formula can be used for definition of faults L detected at output.

4 Algorithm of cubic SSF simulation for combinational circuits

The cubic deductive SSF simulation algorithm for combinational circuit is defined according to the procedure (3):

1.First, the fault-free simulation of next primitive $P_i (i = \overline{1, M})$ at the test-vector $T_t (t = \overline{1, N})$ is executed. If $t = N$, then a detected fault list L(T) at the test T is formed. The end of simulation. Otherwise, if $t < N$, then move to point 2.

2. If all circuit elements are processed $(i = M)$, the comparison of two consecutive fault-free vectors is executed. If vectors are equal ($T_t^r = T_t^{r-1}$), then there is the end of simulation T_t and move to point 3. Otherwise, move to point 1.

3. Primary input fault lists are defined in the form of a complement to their fault-free state $L_j = \{j^{\overline{T}_j}\}$.

4. The fault simulation according to the procedure is executed for primitive $P_i (i = \overline{1, M})$. The primitive output fault identified as $j^{\overline{T}_j}$ is added to the obtained list.

5. If $(i = M)$, then detected fault list $L(T_t)$ generation and motion to point 1 are executed. Otherwise, if $i < M$, then move to point 4.

Cubic fault simulation algorithm for digital systems at functional description level has an operating speed

$$C_k^F = b^2 \times \sum_{i=1}^{M} \{q_i \times n_i \times [(0{,}01 \times L)^2 + 3] + 2\} \approx$$
$$\approx b^2 \times (0{,}01 \times L)^2 \times \sum_{i=1}^{M} (q_i \times n_i),$$

where q_i, n_i is a number of variables and cubes in cubic covering; L is a number of lines in circuit; b is a number of faults; M is a number of functional elements; $(0{,}01 \times L)$ is an average number of active adjacent patterns of circuit lines; b is a number of non-equivalent faults in circuit.

The operation speed of digital systems simulation at the gate description level is evaluated by following expression:

$$C_k^G \Big|_{(M=G;G=q_i \times n_i)} = b^2 \times L^2 \times G.$$

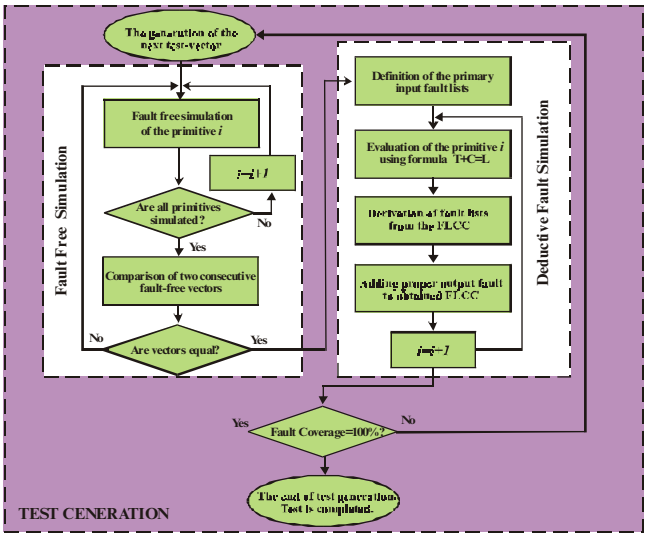

Fig. 2. A digital functional module

In [3] the operation speed evaluation for concurrent algorithm is $C_p = (b^2 / W) \times G^3$ and for deductive one (CHIEFS system) is $C_d = b^2 \times Q \times G^2 \Big|_{Q=G} = b^2 G^3$, where W is a lenght of word; G is a number of equivalent gates; Q is an average number of gates sensitized by faults.

Since a number of circuit lines is less than a number of gates at least in two times, the proposed cubic simulation method has a better operaiotn speed in comparison with dedutive algorithm. This gain will be bigger under processing of circuits at functional level, when a number of lines is less than a number of gates by dozens of times.

5 Fault simulation in sequential primitive elenemts

Output fault list is a function, which is specified according two consecutive input vectors, where each coordinate is specified in the following combinations:

$$T_t = \left[\frac{0}{0}, \frac{0}{1}, \frac{1}{0}, \frac{1}{1}, \frac{0}{X}, \frac{1}{X}, \frac{X}{0}, \frac{X}{1}, \frac{X}{X}\right].$$

The two-frame format of an input vector is directed forward the sequiential FSM analysis, since, in general case, its covering is specified in the two-frame alphabet. Hence, we

can in form of table represents $\oplus$-operation between coordinates of the test-vector and the literals of two-frame cubic covering.

Each coordinate of the table is a compact form of fault lists $L = T \oplus C$. For instance, if input test-vector T=(01) and cubic covering C=P the fault list cubic covering is following: L=S

Output coordinates of FLCC at two frames also can be represented in form of table. For instance, for the coordinate L=V, its value in previous frame is 0, and in current frame is 1, and it is specified in in that table by letter E. Determinations of symbols S, P are exceptions. There is an interpretation difference depending on the fact, whether an output variable is the function or the argument to the output, for which a detectable fault list is generated. In the first case, determination of the above-mentioned symbols gives (J, E), in the second one – (X, X), that indicate to the absence of FLCC for the specified output in the frame t-1.

6 Modified genetic algorithms for test generation

Genetic algorithms are oriented to determination of function extremum [3]. Depending on its properties of convexity, linearity and differentiability the appropriate algorithm of search of minimum or maximum is selected. If the function is rather complex, multimodal and if it has a lot of break points then finding of effective algorithm of extremum search is a difficult problem. As regards discrete functions, it's possible to find appropriate extremum here without any assumptions about function properties (linearity, monotonicity, conservability and augmentability). Here, the special deterministic algorithms [2] which use values of function in one or another point of Boolean space are necessary.

Genetic algorithms (GAs) are the algorithms of optimization based on usage of natural selection mechanism. There are the following features of GA application in contrast to other algorithms of optimization. They are:

1) GAs work with vectors or codes of set of parameters.
2) GAs execute search of extremum in the population presenting set of points of Boolean space in each iteration.
3) GAs use only the value of function in the point of discrete space without taking into account its other properties.
4) GAs aren't deterministic methods but they are probabilistic ones.

Modified models of test generation for digital systems verification in term of genetic algorithms is offered. The main purpose of improving is maximal elimination of factor of randomness from procedures of initial population definition, crossover, mutation and natural selection.

Advantage of GAs is considerable high-speed operation of test generation with ordered fault coverage in comparison with deterministic methods. Drawback of GAs is probabilistic procedures generating test, which can't be reproducible iteratively. It is unacceptable for digital systems verification. That's why, factor of randomness should be maximally excluded.

For that the modification of listed procedures of crossover, mutation and natural selection is executed with substitution of probabilistic components by deterministic ones.

7 ATPG Systems using developed fault simulation and genetic algorithms methods

Test generation system for boolean equation TESTBUILDER.

The program is intended for ATPG with respect to SSFs of digital designs described in language of Boolean equations.

Program operations:

1. Pseudo-random test generation in term of built-in binary code generators and decimal code generators.
2. Deterministic binary test-vector generation, where the mentioned test-vectors sensitize single logical paths in circuit.
3. Single stuck-at fault simulation with purposes of fault coverage evaluation of obtained test.
4. Test formatting in standard of VHDL - Testbench.

The program has processed:

– 10 combinational circuits from list ISCAS'85; average time of deterministic test generation is 28 minutes.

–140 combinational and sequential circuits from PRUS; 45 sequential circuits with large complexity from PRUS;

– 22 sequential circuits from list ITC'99; average time of deterministic test generation is 2 hours.

– 216 sequential circuits; average time of deterministic test generation is 14 seconds.

– 72 combinational circuits; average time of deterministic test generation is 57 seconds.

Average complexity of design is 1000 lines. Average time of pseudo-random test generation is 5 minutes. Test coverage is more than 90 %.

Test generation system used GA.

Program implementation of the method has proved its effectiveness in the view of speed operation and quality of generated tests. The strategy of processing of digital device presented in form of circuit contains next steps:

- algorithmic tests selection for initial population depending on structural and functional complexity of object;
- definition of fitness-function of population by fault simulation where fitness-function is test quality;
- execution of procedures of crossover and mutation in term of deterministic procedures usage;
- forming of individuals-offsprings for the next population which doesn't substitute previous one but it's always expansion to it.

Fault simulator.

Fault simulator is intended for single stuck-at fault simulation of digital circuit, where the digital circuit is described at functional level in form of cubic coverings.

The problems solved by the program:

1. SSF simulation on cubic coverings of functional primitive elements.
2. Simulation of complements to states of circuit lines on cubic coverings of functional elements.
3. Algorithmic and pseudo-random test generation.
4. Length test optimization by improving its quality.
5. Optimization of number of algorithmic generators by coverage problem solving.

Initial descriptions of testing object are VHDL and representation of circuit in form of Boolean equations.

Result of program work is the test for digital design represented in VHDL (Testbench) format.

8 Experimental results

Two test generation systems (TestBuilder uses deterministic and genetic algorithms, Nemesis) have been evaluated by three parameters: the test generation time, test quality and test size. Test-examples are selected from ISCAS85 combinational circuits, maximum size is 7552 equivalent gates. In general, the developed

Table 1

test	ATPG	algorithm	time	FC[%] PRT	test size
c1355	testbuilder	genetic algorithm	0,12	99,99	104
		deterministic	3,11	99,06	70
	nemesis	without random	4,11	99,05	
c17	testbuilder	genetic algorithm	0	100	7
		deterministic	0	100	4
	nemesis	without random	0	100	
c1908	testbuilder	genetic algorithm	5,11	98,1	76
		deterministic	8,91	96,93	44
	nemesis	without random	7,809	99,05	
c3540	testbuilder	genetic algorithm	4,3	96,3	169
		deterministic	10,5	96,16	88
	nemesis	without random	8,677	95,7	
c432	testbuilder	genetic algorithm	0,20	99,23	45
		deterministic	0,5	99,23	22
	nemesis	without random	0,649	99,2	
c499	testbuilder	genetic algorithm	0,80	100	65
		deterministic	0	100	49
	nemesis	without random	0,104	98,9	
c5315	testbuilder	genetic algorithm	4,16	99,9	121
		deterministic	3,43	99,8	70
	nemesis	without random	3,8	98,8	
c6288	testbuilder	genetic algorithm	4,11	99,65	48
		deterministic	6,545	99,65	21
	nemesis	without random	6,238	99,5	
c7552	testbuilder	genetic algorithm	9,45	96,5	128
		deterministic	15,95	94,06	55
	nemesis	without random	22,93	98,2	
c880	testbuilder	genetic algorithm	0,40	99,3	57
		deterministic	0,4	99,44	31
	nemesis	without random	0,332	100	

tools of test synthesis based on genetic algorithm method have advantages comparing with Nemesis and deterministic method:

1. In time. On average 49% saving of time have been received (fig.3). (In general for big size circuits)).
2. In quality of test 0,06 % (fig.4). Genetic has advantage.

9 Conclusions and future work

The cubic fault simulation method is a new technology of digital circuits processing at gate, functional and algorythmic description level. It allows to simulate all single stuck-at faults detected by test-vector during one iteration. The application condition consists in usage of digital circuits description in terms of cubic coverings of primitive elements. The proposed method effectively processes sequiential digital circuits described by two-frame cubic coverings as well. The last ones formalize algorithm descriptions in the form of primitives corresponding to transition graphs, FSM-charts, state tables of digital circuits.

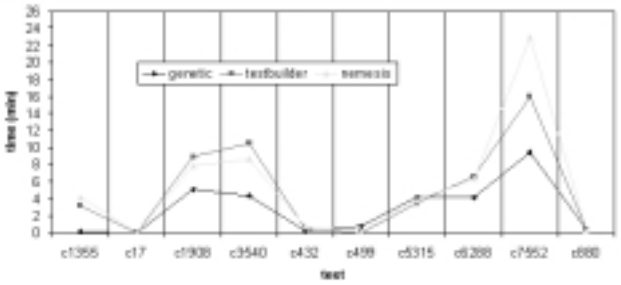

Fig. 3. Comparative analysis of test generation time

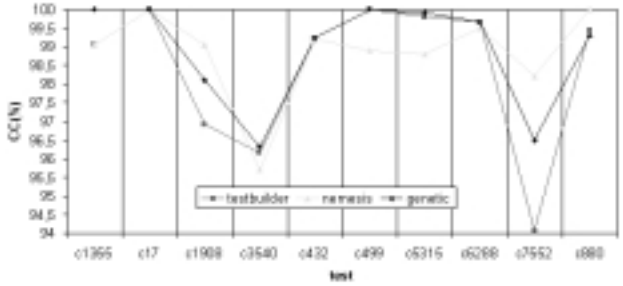

Fig. 4. Comparative analysis of test pattern quality

The proposed technology of testing by the equation $T \oplus C = L$ provides the possibility of fault simulation on the basis of the cubic covering analysis. It also allows to obtain deductive formulas for any typical functional element, to design compilative simulators for processing of digital circuits at optional description level, to generate tests for digital circuits on the basis of FLCC usage, to verify results of fault simulation and test generation and to design high-speed hardware simulators.

The practical implementation of genetic algorithms method for test generation has been proposed. It combines advantages of genetic algorithms (high-speed operation of test generation with reserved fault coverage) with advantages of deterministic ones (reproducibility of generated input sequences).

Program realization of method has proved effectiveness of proposed models and methods of test generation. It takes intermediate position between algorithmic and deterministic methods. For effective usage of software containing implementation of genetic algorithms method it's necessary to define parameters of circuit processing skilfully. These parameters depend on structure of tested digital device (Unit Under Test).

The proposed models and methods are realized in the form of program applications. The last ones are used for test generation of digital designs based on FPGA and CPLD. The class of processed structures is FSM in the form of transition graph and Boolean equations on flip-flop circuit. Digital circuit description language is VHDL. Program applications are directed toward their use in design systems: Aldec, Xilinx.

References

[1] Armstrong D.B. A deductive method of simulating faults in logic circuits. IEEE Trans. on Computers. Vol. C-21. No. 5. 1972. P. 464-471.

[2] Abramovici M., Breuer M.A. and Friedman A.D., Digital System Testing and Testable Design, Computer Science Press, 1998. 652 p.

[3] Goldberg D.E. "Genetic Algorithm in Search, Optimization and Machine Learning".– Addison Wesley.– 1989.– 342p.