

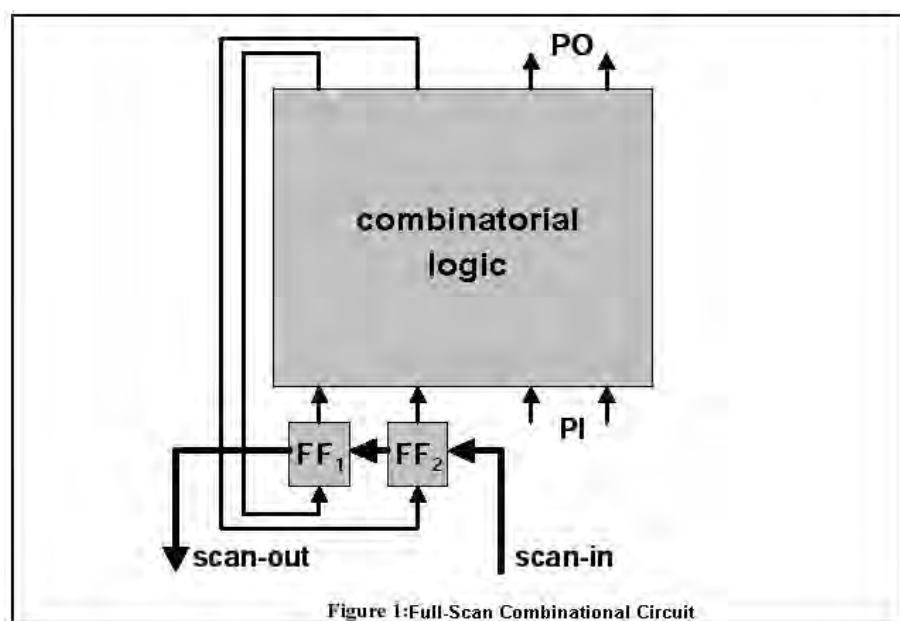
MINIMIZING TIME FOR TEST IN INTEGRATED CIRCUIT

© Andonova A.S., Dimitrov D.G., Atanasova N.G., 2004

The cost for testing integrated circuits represents a growing percentage of the total cost for their production. The former strictly depends on the length of the test session, and its reduction has been the target of many efforts in the past. This paper proposes a new method for reducing the test length by adopting a new architecture and exploiting an evolutionary optimisation algorithm. A prototype of the proposed approach was tested on ISCAS standard benchmarks and the experimental results show its effectiveness.

1. Introduction

In recent years the integrated-circuit design flow experienced radical changes. Deep sub-micron integrated-circuit manufacturing technology is enabling designers to put millions of transistors on each integrated circuit. Automatically synthesized finitestate machines (FSMs) acting, for instance, as control units, can often be found in current designs, and their complexity is skyrocketing. Clearly, industries need to check the correctness of the manufacturing process before selling these devices. Or, as electronic engineers use to say, devices have to be *tested*. In the test process, the device is stimulated with a set of input patterns and its response is compared with the expected one. Indeed, in order to fruitfully check all functionalities, test engineers need to be free in applying stimula to inputs of the circuits (called *primary inputs*) and in observing outputs (called *primary outputs*). Moreover, they must be able to set the *state* of the FSM and check it after the application of the inputs. However, these devices may be so complex that bringing them to an arbitrary state or discovering the current one are hard tasks. In the general case, these problems have been proven NP-Complete and, practically, are frequently infeasible. A standard technique adopted in the industry is to modify memory elements (figure 1), substituting simple flip-flops (e.g., *FF1* and *FF2*) with special cells that have an extra output and an extra input. Such cells are then connected forming the so-called *scan chain*. During test, first the appropriate state is loaded into memory elements; then stimula are applied to the primary inputs (PI) monitoring primary outputs (PO); and finally the new



state is read. The internal state of the device becomes both controllable and observable, but, even though, this solution is not a panacea. Since memory elements are connected in a chain, values are sequentially shifted in and out (these operations are called respectively *scan-in* and *scan-out*). The number of *clock cycles* is directly proportional to the number of memory elements and, in particular on large designs, time overhead is significant.

This paper presents a new test strategy for tackling the problem of testapplication time cost, called *interleaved scan*. Section 2 describes the approach, while section 3 illustrates its evolutionary core. Section 4 reports some preliminary experimental results and section 5 concludes the paper.

2. Interleaved Scan

In the full-scan approach, engineers must devise an appropriate set of inputs to test the circuit. The set of values applied simultaneously to PI is called a *vector*, the set of these vectors is termed *test pattern*. The test pattern is generated by a special program, called *automatic test pattern generator* (ATPG). The ATPG assumes that all memory elements are controllable and observable. The application of the test can be exemplified in three steps. For each vector of the test pattern, memory elements must be set. Thus, the appropriate values are shifted in using the scan chain. Values are applied to primary inputs, the circuit produces its outputs and flip-flops latches their new state. Finally, the test machine (called *ATE*, Automatic Test Equipment) shifts out values from memory elements.

Loading and extracting values from memory elements are time-consuming processes. Not to overextend the number of input/output ports, memory elements are connected in a chain, where values are sequentially shifted in or out. As mentioned before, the number of clock cycles required is directly proportional to the number of memory elements (*nFF*). More precisely, Bushnell and Agrawal [1] elaborate a useful formula for calculating the total number of clock cycles (CC) required by the full-scan approach:

$$(3) \quad 4 \text{ ATPG } FF \text{ CC } n n = + \cdot + (1)$$

Where n_{ATPG} is the number of vectors composing the test pattern and n_{FF} is the number of flip-flops.

Several efforts have already been made to reduce the test application time. Some proposals are called *static techniques* since they have to be applied *after* test pattern generation. For instance, Niermann et al. tried to rearrange ATPG-generated vectors to reduce overall test application time by overlapping scan-in and scan-out operations [2]. On the other hand, *dynamic* approaches attempt to reduce the number of test vectors *during* test pattern generation (e.g., [3]). *Interleaved Scan*, the new approach presented in this paper, is to some extent in the middle. It starts from an ATPG-generated test pattern, but then it generates new vectors.

In the full-scan strategy, only one vector is applied on primary inputs between scan-in and scan-out operations. While in the interleaved-scan approach several different vectors can be applied before the scan-out. Thus, the FSM is first brought to a known state, and then it is evolved by applying different

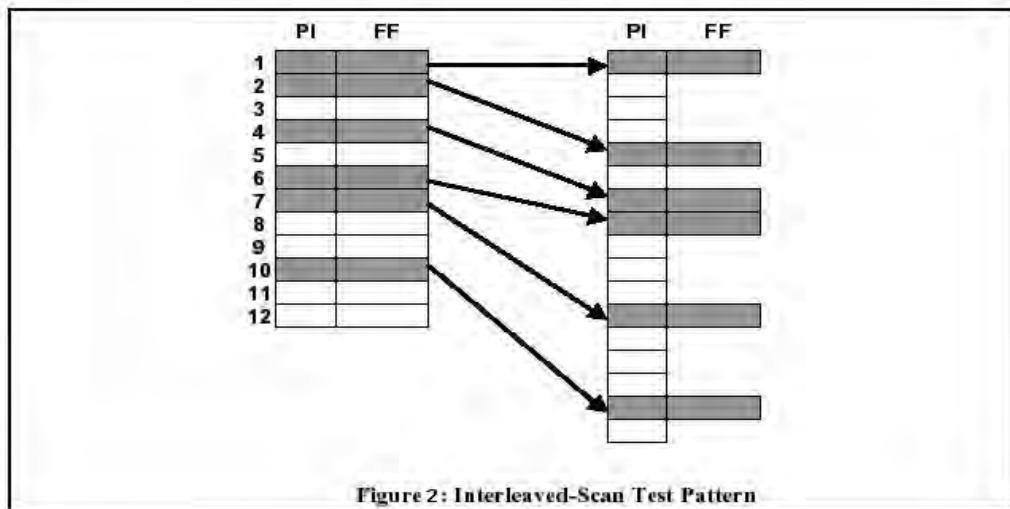
inputs. Only at the end, its state is explicitly examined through scan-out. This apparently small difference has two important consequences. On the one hand, the ATE is required to be slightly more complex. It must be able to perform a scan-in; then drive the circuit under test for several clock cycles applying primary inputs and monitoring primary outputs; and eventually perform a scan-out.

On the other hand, the ATPG must be *extremely* more versatile. The test pattern consists of some *combinational* vectors, where both primary inputs and memory elements are fully specified, and some *sequential* ones, where only primary inputs are marked. The ATPG should be able to tackle the sequential circuit, optimizing the number of scan operations. Unfortunately, no commercial tools with

such characteristics exist. Thus, we propose to generate the standard full-scan test vectors with a standard ATPG and then to extend them for the interleaved-scan approach.

Figure 2 illustrates the proposed approach. A full-scan test pattern (left) is extended to an interleaved-scan one (right). Vectors are composed of two parts: the first one contains values for the primary inputs (PI), while the second one contains the state of the FSM, i.e., flip-flop values (FF). Full-scan test vectors

are copied into the new test pattern (vectors number 1, 2, 4, 6, 7 and 10), while new vectors (shown in white) are generated to complete them. These new vectors will be applied between scan operations, thus they do not specify memory-element values and contains only primary-input data. Not all vectors need to be copied from the full-scan test pattern since the newly generated ones may have made some useless (in the example, vectors number 3, 5, 8, 9, 11 and 12 were discarded). Some full-scan vectors do not need any extension (e.g., vector number 4). It is important to note that the overall number of interleaved-scan vectors is likely to be larger to the number of full-scan ones.



However, since the number of scan operations is reduced, the total number of clock cycles required to run the test is lower. This paper also proposes a method for building an interleaved-scan test pattern starting from a full-scan one. The method is based on an evolutionary algorithm and is detailed in the next section.

3. Interleaved-Scan Test Pattern Generation

Starting from a full-scan test pattern, the proposed interleaved-scan test pattern generation consists in two steps: select a subset of full-scan vectors and extend them by possibly adding new vectors. The two steps are performed iteratively: first, a vector is heuristically chosen. Then it is expanded. The process is reiterated until required. In the first step, the vector able to detect the larger amount of faults is selected. The second step exploits a local-search technique that mixes the explicit neighborhood exploration of hill-climbing with the evolutionary concept of population. Our optimization algorithm exploits a steepest-step hill climber, where each solution represents a whole population of individuals [5]. In more details, at the beginning of the evolution process, a population of individuals is randomly built. This population represents the initial starting point. All individuals are evaluated and the population is assigned a value equals to the fitness of its best individual (its champion). Then, as in Genetic Algorithms (GA), genetic operators are applied to generate a new population. These operators include standard crossover operators and mutations. However, differently from a standard GA, np new populations are concurrently built and evaluated. Finally, as in a hill-climber, the best new population is selected as the new starting point, and the process is iterated. Figure 3 exemplifies this approach. Starting from a population of sequences (on the left), $np = 3$ new populations are built. The third one contains the champion (the fittest individual), thus it is selected as new starting point.

Thus, the evolutionary concept of *fitness* is used two times in the optimisation algorithm. Firstly, as in standard GA, the fitness is used to select individuals for reproduction when building a new population. Secondly, the fitness of the best individual in a population is used to rank the population against neighbour populations for the hill-climbing step. In the proposed framework, the goal of the optimisation is to devise a set of vectors to extend a full-scan one. This set can be empty, because the best choice may be not to extend the current vector. Individuals in populations are *sequences*, i.e., binary vectors to be applied to the

circuit primary inputs in consecutive clock cycles starting from the initial state specified in the full-scan vector. The *fitness* of an individual simply measures the number of previously undetected faults that the sequence is able to discover. A penalty term slightly penalizes longer sequences. In each step, new populations are generated from the current one through crossover operators, where the new sequence is generated from two parents, and mutations, where it is generated by modifying an existing one. Four crossover operators are defined and they are chosen with equal probability:

- **Horizontal 1-cut crossover:** the new sequence is composed of some vectors coming from either parent, according to the position of one cut point randomly generated in the first individual, and another one randomly generated in the second.

- **Horizontal 2-cut crossover:** the new sequence is composed of some vectors coming from either parent, according to the position of two cut points randomly generated in the first individual. The length of the new sequence is the longest between the two parent ones.

- **Horizontal uniform crossover:** each vector in the new sequence is taken randomly from the first or from the second parent. The length of the new sequence is the longest between the two parent ones.

- **Vertical uniform crossover:** each vector in the new sequence inherits some bit columns from the first parent and some from the second. The length of the new sequence is the longest between the two parent ones: inputs taken from the shortest parent are completed with random values where needed.

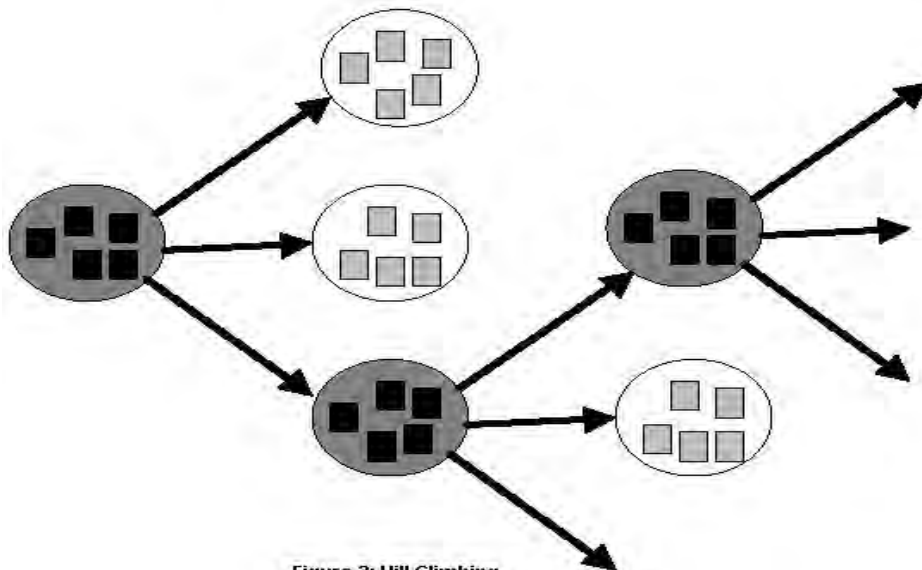


Figure 3: Hill Climbing

Three mutation operators are implemented and are selected with equal probability:

- **Change mutation:** a vector in the sequence is replaced with a new randomly generated one.
- **Add mutation:** a random vector is added in a random position, shifting forward the subsequent vectors.

- **Delete mutation:** a randomly selected vector is removed from the sequence, shifting backward the subsequent vectors. Individuals are selected for applying genetic operators using the roulette wheel technique on their linearized fitness: sequences with higher fitness are likelier to be selected for crossover or mutation, so that good sequences are given more chances to generate better ones.

4. Experimental Results

A prototype implementing the proposed approach was built in C++. The prototype was tested on ISCAS89 circuits [4], a standard set of benchmarks in the electronic test field.

Table show the parameter values exploited in the experimental evaluation. The prototype required about 24 hours of CPU time, table 2 details its results. The first two columns report the name of the circuit [Circuit] and the number of memory elements [nFF]. Next, figures for the full-scan [Full-Scan] approach are shown: the number of test vectors composing the whole test pattern [# Vec], and the corresponding number of clock cycles [CC]. The same information is reported for the interleaved scan [Interleaved-Scan]. Last column summarizes the gain in percentage attained by the proposed approach. Results are also summarized in figure 4.

Table 1

Experimental Parameters

Parameter	Value	Meaning
N_p	10	Number of neighbour population
S_p	50	Population size
S_o	30	Offspring size (new individuals generated)
M	10	Mutation probability

Table 2

Experimental Results

Circuit	nFF	Full-Scan		Interleaved-Scan		GAIN
		# Vec	CC	# Vec	CC	
s27	3	6	31	9	24	22.58%
s208	8	33	292	58	177	39.38%
s298	14	29	452	58	299	33.85%
s635	32	59	1988	59	1771	10.92%
s641	19	32	669	90	619	7.47%
s713	19	32	669	90	529	20.93%
s820	5	122	629	203	474	24.64%
s832	5	127	654	204	475	27.37%
s838	32	170	5540	208	3098	44.08%
s938	32	173	5636	223	2989	46.97%
s953	29	98	2933	155	974	66.79%
s967	29	101	3020	182	1225	59.44%
s991	19	28	593	69	418	29.51%
s1196	18	164	3010	244	1390	53.82%
s1238	18	177	3244	333	1649	49.17%
s1269	37	37	1484	149	660	55.53%
s1423	74	44	3482	156	2683	22.95%
s1488	6	138	850	193	470	44.71%
s1494	6	135	832	238	510	38.70%
s1512	57	83	4906	118	3765	23.26%
s3271	116	61	7428	206	5273	29.01%
s3330	132	224	29968	389	18998	36.61%
s3384	183	80	15193	157	11630	23.45%
s4863	104	60	6556	174	4610	29.68%
s4863	104	60	6556	174	4610	29.68%
s6669	239	43	10998	105	3682	66.52%
s9234	228	238	54952	395	50569	7.98%
s35932	1728	40	74308	70	29436	60.39%

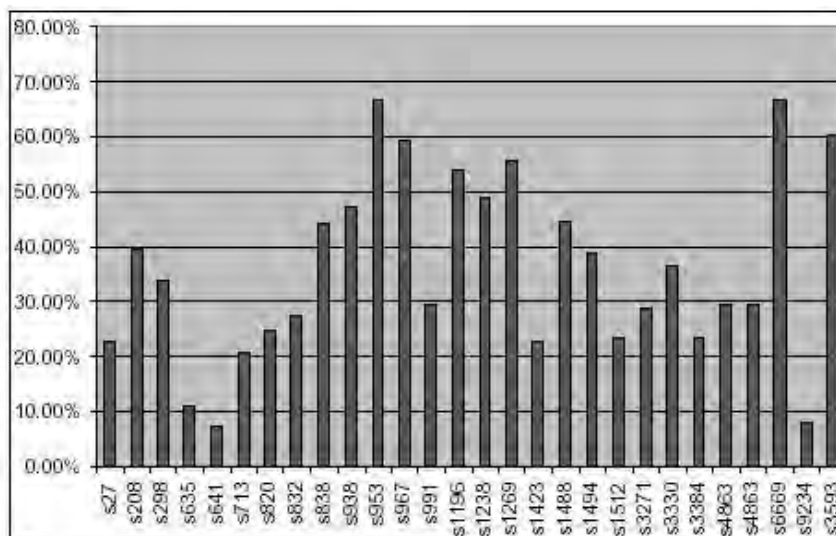


Figure 4: Test Application Time Gain

Experimental data show a significant reduction in test application time for most of the circuits, an average of 36% with two peak gains of 66%. It can also be noted that, as the number of memory elements increase, the improvement for removing a scan-in set operation step up. Thus, the interleaved scan is expected to scale well with design complexity.

Few benchmarks, like s635 and s641, are so *hard-to-test* that almost all ATPG-generated full-scan vectors are required in the interleaved-scan test pattern, thus, gain is strongly limited. On the other hand, benchmark s9234 deserves special consideration and is currently under study.

5. Conclusions

Interleaved scan is a promising test architecture for reducing the number of clock cycles required for a test session. Interleaved scan requires an effective optimisation algorithm to extend the test vectors generated by a standard ATPG. A local-search technique has been presented to address this issue, based on explicit neighborhood exploration and exploiting the concept of population. A prototype was

tested on the standard ISCAS benchmarks and the experimental results show the effectiveness of the proposed approach. Authors are experimenting the interleaved scan on a larger set of circuits to better assess its applicability. Simultaneously, they are working on applying the evolutionary core proposed in this paper to different problems.

1. *Bushnell M.L., Agrawall V.D. Essentials of Electronic Testing for Digital, Memory & Mixed Signals VLSI Circuits, Kluwer Academic Publishing, 2000.*

2. *Niermann T.M., Roy R.K., Pasel J.H., Abraham J.A. Test Compaction for Sequential Circuits // IEEE Transactions on Computer-Aided Design, February 1992, pp. 260–267.*

3. *Pomerantz I., Reddy L N., Reddy S.M. COMPACTEST: A Method to Generate Compact Test Sets for Combinational Circuits // IEEE Transactions on Computer-Aided Design, November 1993, pp. 1361-1371.*

4. *Brglez F., D. Bryant, Kozminski K. Combinational profiles of sequential benchmark circuits // Proc. Int. Symp. on Circuits And Systems, 1989, pp. 1929-1934.*

5. *Goldberg E. Genetic Algorithms in Search, Optimization, and Machine Learning // Addison-Wesley, 1989.*

I. Bolshakova¹, V. Brudnyi², N. Kolin³, P. Koptsev¹, Ya. Kost¹,
N. Kovaleva¹, O. Makido¹, T. Moskovets¹, F. Shoorigin¹

¹ Lviv Polytechnic National University, Lviv, Ukraine,

² Siberian Physical Technical Institute, Tomsk, Russia,

³ Obninsk Branch State Research Center “Karpov Institute of Physical Chemistry”, Obninsk, Russia

THE BEHAVIOUR OF InSb UNDER THE IRRADIATION WITH REACTOR NEUTRONS

© Bolshakova I., Brudnyi V., Kolin N., Koptsev P., Kost Ya.,
Kovaleva N., Makido O., Moskovets T., Shoorigin F., 2004

In the paper, is investigated the influence of the irradiation with full reactor neutron spectrum up to the fluence of $F=3 \cdot 10^{16} \text{ cm}^{-2}$ upon the electrophysical properties of complex doped InSb microcrystals and thin film InSb samples with charge carrier concentration of $n=(9 \cdot 10^{16} + 3 \cdot 10^{18}) \text{ cm}^{-3}$. The influence of the initial doping level on the radiation resistance is determined. The optimal charge carrier concentration for the manufacturing of the radiation resistant magnetic field microsensors is determined and is equal to $n=(6 + 7) \cdot 10^{17} \text{ cm}^{-3}$ for the complex doped InSb microcrystals, and $n=3 \cdot 10^{17} \text{ cm}^{-3}$ for thin film samples.

Introduction

The InSb semiconductor material, owing to its electrophysical properties (high charge carrier mobility, performance etc) has found its practical application as a sensitive element in magnetic field sensors, intended for operation in extreme conditions.