

Tailoring ATPG for Embedded Testing

Rainer Dorsch

Hans-Joachim Wunderlich

University of Stuttgart

Breitwiesenstr. 20-22, D-70565 Stuttgart, Germany

phone +49-711-7816-215, fax +49-711-7816-288

{rainer.dorsch,wu}@informatik.uni-stuttgart.de

Abstract— An automatic test pattern generation (ATPG) method is presented for a scan-based test architecture which minimizes ATE storage requirements and reduces the bandwidth between the automatic test equipment (ATE) and the chip under test. To generate tailored deterministic test patterns, a standard ATPG tool performing dynamic compaction and allowing constraints on circuit inputs is used. The combination of an appropriate test architecture and the tailored test patterns reduces the test data volume up to two orders of magnitude compared with standard compacted test sets.

Keywords: Test Resource Partitioning, Systems-on-a-Chip, ATPG

I. INTRODUCTION

A system-on-a-chip (SoC) typically contains various pre-designed and prevalidated embedded cores like memory cores, processor cores, interface cores, mixed-signal, and analog cores. The SoC design style increases the designers productivity, but at the same time the test complexity of the chip increases and the cores' accessibility from the chip pins is reduced. Higher test data volume is necessary to test the chip, higher ATE-chip bandwidth and more test pins are necessary to keep the test times low, and higher timing resolution is necessary to avoid yield loss.

The ITRS roadmap [1] predicts a significant cost increase of ATE capable to test the next generation of systems-on-a-chip, and investing in techniques complementing ATE becomes now attractive. Fig. 1 shows three possible areas in which the test complexity problem might be addressed: design-for-testability, design and test automation tools, and ATE architecture. The presented approach addresses all three areas concurrently.

A common test method is the application of a test set generated by an ATPG tool tuned to minimize the number of test patterns by compacting test vectors statically and dynamically [2–6] in order to reduce test application time and storage requirements for the ATE. A designer has now a tradeoff between using sophisticated ATPG tools to get very compact test sets or to use a sufficiently powerful ATE architecture. Lower limits of the minimum sizes of compacted test sets [3] show that reducing the test data volume significantly is not possible by improving the ATPG algorithms only, i.e. addressing only the area of design and test automation tools is not sufficient.

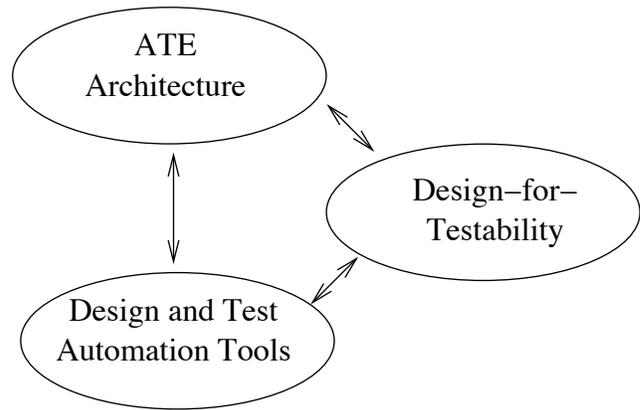


Fig. 1. Potential Techniques for SoC Testing.

Some Built-In-Self-Test (BIST) techniques generate the test patterns on-chip based on a compact representation of the test patterns [7–10]. These *encoded test data* may also be stored on an ATE in order to reduce the test data volume considerably. A tailored ATPG algorithm for these BIST schemes was introduced in order to achieve better compression of the test patterns [11].

There is a tradeoff among the size of the test pattern generator, the ATE storage requirement, the test length and the test generation time required by the tool generating an encoded representation of the deterministic patterns. The LFSR based approaches have a good compression rate and are compatible with standard-scan design, but test controllers and on-chip pattern generators are required. Moreover, a small change in the test requirements and the test set after tape out might be impossible because linear dependencies might prohibit the generation of certain test patterns with the implemented LFSR.

Recently, a new test architecture REusing Scan chains for test Pattern decompressIoN (RESPIN) was introduced [12]. In the RESPIN architecture a set of precomputed test patterns is applied to a core under test which are generated from encoded test data stored on an ATE. In terms of area and control overhead the RESPIN architecture is comparable to architectures which equip each core with a test wrapper [13] and apply compacted test sets to an embedded core using a test access mechanism (TAM). In terms of ATE storage and ATE-chip bandwidth

requirements, it is similar to BIST approaches.

The present paper describes how a standard ATPG tool must be controlled by input constraints in order to obtain a test set tailored to achieve a very low volume of encoded test data for the RESPIN test architecture. It is found that a tailored test set reduces the size of the encoded test data stored on the ATE significantly compared to a test set generated without constraining the ATPG during pattern generation.

The RESPIN test architecture is revisited in the next section. Section III describes the compaction scheme using constrained ATPG. The experimental results for the ISCAS89 benchmarks presented in Section IV show a test data volume reduction up to two orders of magnitude by using the new ATPG method together with the RESPIN test architecture compared with compacted test sets.

II. RESPIN TEST ARCHITECTURE

For the convenience of the reader, this section revises the RESPIN test architecture [12]. Fig. 2 shows an example and illustrates how the CPU is tested in the RESPIN test architecture. Other cores are tested in the same way concurrently with the CPU or sequentially after or before the CPU. The scan chains of the MPEG core are reused to decompress the test patterns for the *core under test* (CUT), which is the CPU. Since the MPEG core performs a tester function embedded on the chip, it is denoted as *embedded tester core* (ETC).

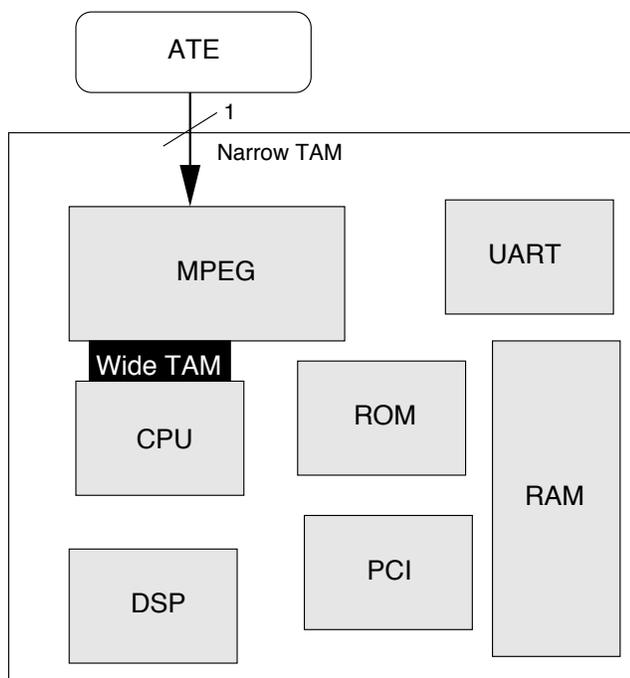


Fig. 2. RESPIN Test Architecture.

The encoded test data are stored on the ATE. These data are transported from the ATE to the chip on a narrow low bandwidth TAM. To transform the encoded test data into test patterns the scan chains of the ETC are reused. The test pattern

is transported on the wide high bandwidth TAM from the ETC to the CUT and the test responses may be compacted on chip using dedicated compaction hardware [14] or reusing cores implementing system functionality [15–18].

For each test pattern, one bit of encoded test data is transported from the ATE to the ETC. Sometimes one bit of test data is not sufficient to control the ETC to generate a test pattern, which increases the fault coverage in the CUT. In these cases a *link pattern* is generated, which does not increase the fault coverage (but it may increase defect coverage).

Both the CUT and the ETC are equipped with scan chains and isolated from their environments by a test wrapper during the test. Each core has a serial test access (see Fig. 3a), which is a mandatory element of the proposed P1500 standard for embedded core test [13], and a parallel test access (see Fig. 3b), which is common in industrial designs. Any core with a serial as well as parallel test access may be used as ETC by adding only a single multiplexer (controlled by the tc signal) and a feedback wire as shown in Fig. 3c. The parallel test access outputs of the ETC are connected to the parallel test access inputs of the CUT. The control signal tc of the ETC is connected to the scan enable signal of the CUT.

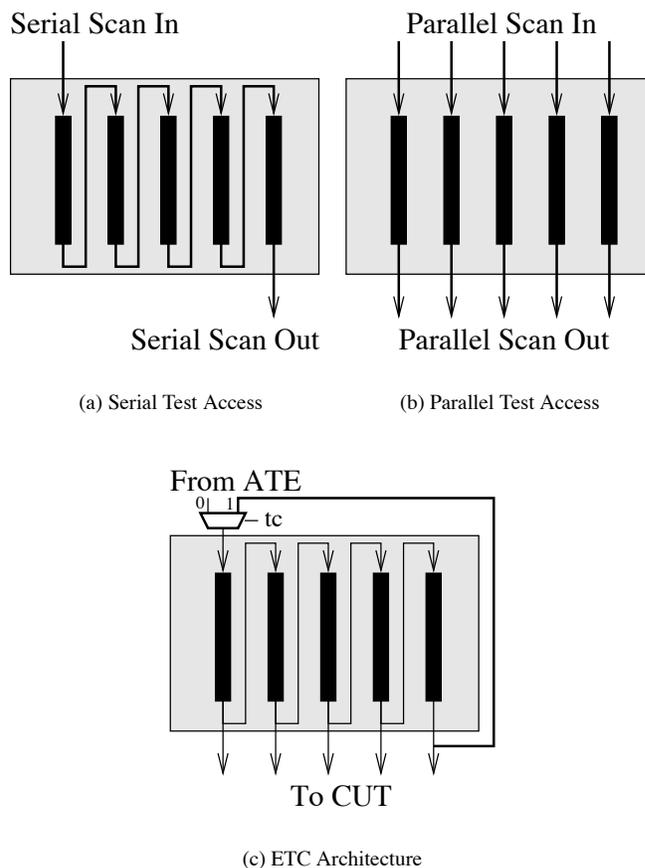


Fig. 3. RESPIN test architecture.

The signal tc selects between *serial mode* ($tc = 0$) and *cir-*

cular mode ($t_c = 1$) of the ETC. Table I shows when the CUT captures the system response, the ETC is in serial mode and loads a bit from the encoded test data. Thus only one bit per test pattern is scanned into the ETC from the ATE. Whenever the CUT scans in test data, the ETC is in circular mode. For using cores as ETC both their shift and capture clocks must be compatible with the CUT clocks.

$t_c = \text{scan enable}$	0	1
CUT	capture mode	scan mode
ETC	serial mode	circular mode

TABLE I
OPERATION MODES OF ETC AND CUT

For example, assume that the CPU used as CUT contains 3 scan chains with at most 3 scan elements each. The test architecture is shown in Fig. 4. The MPEG core selected as ETC has 5 scan chains with $l_E = 21$ scan elements in total. The content of the eight bottom scan cells in the ETC is copied from the ETC into the scan chains of the CUT in the next three circular mode cycles. The number \hat{l}_C of scan elements in the longest scan chain of the CUT is in the example $\hat{l}_C = 3$.

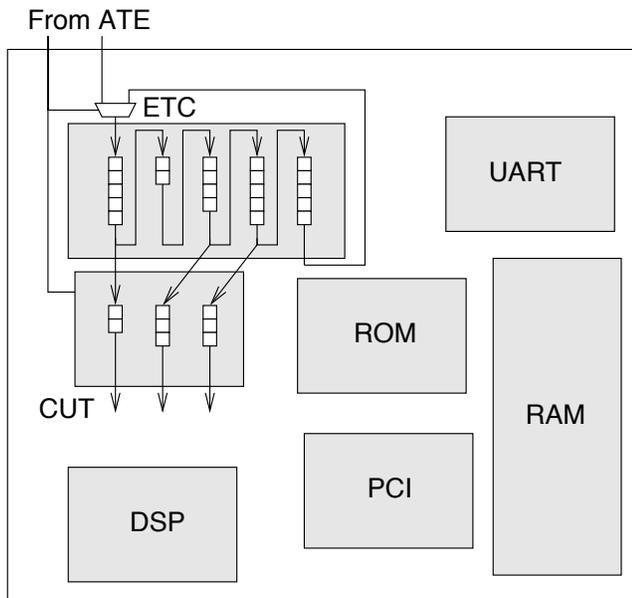


Fig. 4. Example for a test architecture with a simple ETC and CUT.

It is possible to control the ETC in a way that any pattern may be copied in the CUT, if these four conditions are satisfied:

- 1) The greatest common divisor satisfies the condition

$$g.c.d.(\hat{l}_C + 1, l_E) = 1. \quad (1)$$

- 2) The ETC has at least as many scan chains as the CUT.
- 3) No scan chain in the CUT is longer than the scan chain of the ETC it is connected to.

- 4) No scan chain of the ETC is connected to more than one scan chain of the CUT.

An upper bound for the number of clock cycles necessary to copy any pattern in the CUT is $l_E \cdot (\hat{l}_C + 1)$, experiments show that the average number is much less. It is easily verified that these conditions hold for Figure 4. Even in cases where two cores do not satisfy the conditions stated above, following a few design rules will fix the problem [12]:

- Condition 1: Either l_e or \hat{l}_C have to be increased until Eqn. (1) holds. Inserting some additional flip flops into the feedback wire of the wrapper increases l_e and is paid by some additional hardware. Applying a few more circular clock cycles increases \hat{l}_C at the cost of slightly longer test application time.
- Condition 2: Several cores may be combined to a single ETC with a sufficiently large number of scan chains.
- Condition 3: Obviously, the longer scan chains of the ETC must be used. If sufficiently long scan chains are not available, Condition 3 may be satisfied by not connecting the short scan chains of the ETC to the CUT. If a scan chain of the ETC is not connected to the CUT, the number of its scan cells is added to the length of the next scan chain. Fixing this problem may also include using more than one ETC.
- Condition 4: This is a design restriction which can always be followed.

These conditions imply that there must not be a core in the design that has more scan cells than all other cores together, in order to implement the RESPIN architecture.

Concurrent testing of analog, mixed-signal, microprocessor, DSP, and random logic cores is proposed to reduce test application time for complex systems-on-a-chip. Since the number of test pins is fixed for a given design, for each core of a concurrent test only a small subset of the total number of test pins may be used. The RESPIN architecture reduces test pin count considerably such that several logic cores may be tested concurrently.

The RESPIN test architecture requires an ATE that is able to run different ports of the ATE with different frequencies because the narrow TAM is operated with a lower frequency. The embedded cores require the DFT which avoids that ETC is damaged during pattern decompression or the link patterns damage the CUT. The same techniques as for random testing of scan designs may be used.

III. TAILORING ATPG

Usually, a test vector contains a large number of don't care bits, which may be exploited for test data compression [9]. Traditional test pattern compaction tries to merge the detection of many faults into a single pattern. Let T be a pattern in the scan chains of the CUT. If dynamic pattern compaction fails to increase fault coverage by specifying more bits of a pattern T , the unspecified bits are filled randomly. The presented test

architecture offers more degrees of freedom to fill unspecified bits. If dynamic compaction fails to specify more bits in T , after $\hat{l}_C + 1$ clock cycles another pattern T^+ is in the CUT, which requires only a single bit from the ATE. Pattern T^+ may contain some or all unspecified bits of T , and now ATPG tries to increase fault coverage by specifying these bits in the pattern T^+ , instead of filling them randomly in T .

To illustrate this with the previous example, assume that 011 is a test pattern T for CUT and specifying more bits would not detect more faults of the target fault list. Assume further that the ETC is already in a state copying T into the CUT in the next three clock cycles. Fig. 5a shows which bits have to be set in the ETC in this state. After three circular mode cycles T is copied into the CUT as shown in Fig. 5b. Fig. 5c shows the system a serial mode cycle later. The CUT captures the CUT system response on T . Since the content of the scan chains of CUT is not known after the capture cycle, it is represented by $-$. Fig. 5d shows the system after the next three circular mode cycles. The test pattern $T^+ = \mathbf{--} \ \mathbf{---} \ \mathbf{-0-}$ is now copied into the CUT. The bold bits in the pattern are present both in pattern T and T^+ .

The bold $\mathbf{0}$ represents the dependencies between T and T^+ , the bold $-$ represents the higher degree of freedom in the RESPIN architecture when filling unspecified bits. It may now be specified by ATPG for pattern T^+ . This is achieved by running ATPG on the CUT with constraints on the allowable values of the bits of a test vector. Techniques for doing such constrained ATPG have been described in [19] and [20]. These ATPG techniques consider the constraints as early as possible in the ATPG decision making process.

Compared to the first test pattern T , the pattern T^+ obtained $\hat{l}_C + 1$ clock cycles later may contain less specified bits and the pattern obtained after $(\hat{l}_C + 1) \cdot l_E$ cycles does not have any dependencies on T at all [12]. This guarantees that test patterns are generated for all testable faults.

A. Constrained ATPG Encoding Algorithm

The tailored ATPG Algorithm 1 takes the scan chain configuration of the ETC and the CUT as well as a target fault list F_0 as input. Outputs are the tailored test patterns in the encoded form (i.e. a bit sequence $E = [a_0, a_1, \dots, a_{i_e}]$).

In the RESPIN test architecture, the content of the ETC, CUT and the encoded test data have the following dependencies:

- Each bit i in the ETC was scanned in from the ATE $p(i) < l_E$ cycles in serial mode before.
- Each bit i in the CUT was scanned in the ETC from the ATE $q(i)$ cycles in serial mode before.

The mapping functions $p(i)$ and $q(i)$ depend on the configuration of the scan chains of the ETC and the CUT [12].

In Algorithm 1 the set F of undetected faults is initialized by the set F_0 of target faults. The while loop is iterated until patterns for all testable faults are computed. Each iteration corresponds to the generation of a test pattern, l counts the generated test patterns. In each cycle the constraints \mathbf{c} on the PPIs

are extracted from the the encoded test data E by

$$\mathbf{c} = a_{l-q(0)}a_{l-q(1)}a_{l-q(2)} \dots a_{l-q(N_{CUT}-1)},$$

where N_{CUT} are the number of scan cells of the CUT. The extraction is done by reference, i.e. if \mathbf{c} is changed, E is also changed. If a bit a_i in \mathbf{c} has an index $i = l - q(e) < 0$, a bit of the reset state is addressed. The bit position in the reset state is given by

$$e = (-i) \cdot (l_C + 1) \bmod l_P. \quad (2)$$

The constraints are introduced either from a previously computed pattern or from the initial state. ATPG checks now, if any faults undetected so far may be detected by a test pattern satisfying the extracted constraint. If a pattern is found, the specified bits are set in \mathbf{c} . The set of faults F' detected by this pattern are removed from the set of undetected faults F . The pattern counter l is incremented before the while loop is iterated.

Algorithm 1 Constrained ATPG Encoding Algorithm

```

 $F = F_0$ ; // all faults in the target fault list are undetected
Remove all untestable faults from  $F$ ;
 $l = 0$ ;
while  $F \neq \emptyset$  do
   $\mathbf{c} = a_{l-q(0)}a_{l-q(1)}a_{l-q(2)} \dots a_{l-q(N_{CUT}-1)}$ ;
  Run ATPG with  $\mathbf{c}$  and  $F$ ;
  if Pattern found then
    Select one pattern randomly covering the faults set  $F'$ ;
    Write specified bits in pattern in  $\mathbf{c}$ ;
     $F = F \setminus \{F'\}$ ;
  end if
   $l = l + 1$ ;
end while

```

B. Test Flow

The test flow shown in Fig. 6 is similar to a standard test flow of a structural test for a design with multiple scan paths. After scan chain insertion for each CUT, one or several cores are selected that serve as ETC for the CUT. Finally, the ATPG tool generates tailored test patterns for the target faults which are stored as encoded test data.

IV. EXPERIMENTAL RESULTS

Experiments were performed with the ISCAS89 benchmarks [21] as CUT. For the ETC, it was assumed that each scan chain had a length of 100 bit. We only report results for benchmarks that are not testable by random patterns and a few additional deterministic patterns, which could be applied without encoding.

A commercial ATPG tool and fault simulator were used in the constrained ATPG compaction algorithm. The run times including this overhead was between a few seconds for small benchmarks and ten hours for the s38417 benchmark. The long runtime for this benchmark is caused by overhead in the

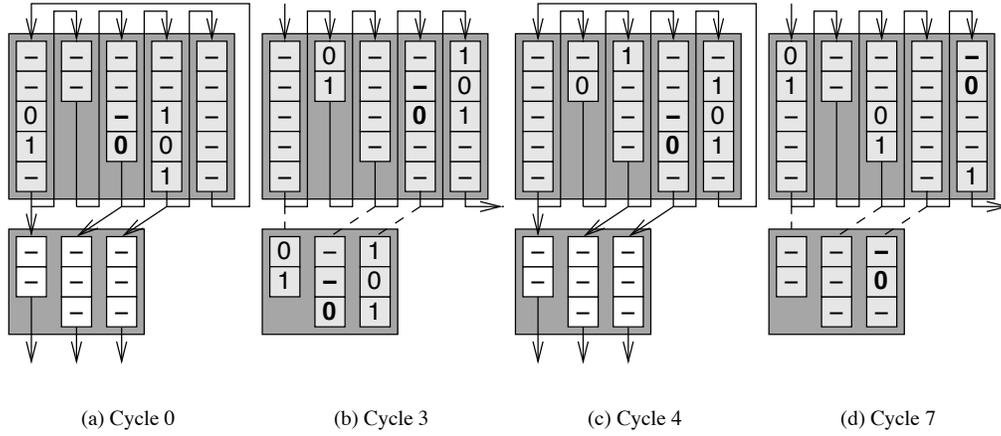


Fig. 5. Dependencies between Test Vector T and T^+ .

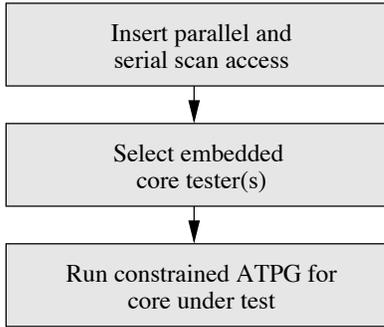


Fig. 6. RESPIN Test Flow.

ATPG tool. It could be avoided, if the source code of the tool would have been available: ATPG and fault simulator are called with different constraints in each iteration. In this commercial framework each call requires to start a new unix process, do a network access to check out a license, and parse the netlists again.

A. Impact of Tailored ATPG Patterns

In this section we compare the impact of the ATPG patterns on the volume of the encoded test data for the RESPIN architecture. Table II shows the volume in bits of the encoded test data for the largest ISCAS89 benchmarks when the RESPIN encoding algorithm [12] is applied to a statically and dynamically compacted test set (compact), a test set not compacted (non-compacted), and the tailored test set (tailored). The same ATPG tool was used for the computation of the tailored test set encoded by RESPIN architecture and the other test sets.

An non-compacted test set for faults not detected by 10,000 random patterns was generated and encoded for RESPIN in [12] (column 5, non-compact). To compare the tailored ATPG with these results, the tailored ATPG was run with a set of faults not detected by 10,000 random patterns (column 6, tailored). For

the most difficult to test benchmark (s38417), the encoded test data volume is reduced by 73% when the tailored test set is used instead of a standard test set as in [12].

It is found that starting from traditionally compacted ATPG patterns increases the test data volume to be stored. The compression ratio for tailored ATPG is even significantly better than the one for non-compacted test sets. Due to the high number of ATPG patterns it is not always possible to start from a non-compacted test set. If tailored ATPG is compared with encoding test patterns for the RESPIN architecture starting from compacted test patterns, we find e.g. for the s35932 benchmark, encoding tailored test patterns reduces the volume of the encoded test data up to two orders of magnitude when comparing it to the results of encoding a traditionally compacted test set.

B. Encoding Efficiency

The standard method for reducing memory and bandwidth requirements is applying compact test sets. In this subsection we compare the memory requirements of compacted test sets with the requirements of RESPIN with tailored ATPG. Both test sets were generated by the same commercial ATPG tool.

In Table III, column (Benchmark) shows the benchmark used as CUT, column (PPI) shows the number of pseudo primary inputs of the benchmark, column (#patterns compact ATPG) shows the number of compacted ATPG pattern necessary to test all testable faults, column (#Bits in patterns (specified)) gives the size of the compacted test set in bits, which is the product of the Column 2 and 3; the number of specified bits in the test set are given in brackets. Column (#Bits RESPIN+tailored ATPG) contains the size of the encoded test data of RESPIN applied to a tailored test set. Column (test data reduction) shows the reduction of the volume of the encoded test data for RESPIN compared to the volume of a statically and dynamically compacted test set. Column (test application time) shows the test application time $(= \text{Column 5} \cdot (\overline{I_C} + 1) / 2 / (\text{Column 2} \cdot \text{Column 3}))$ compared to the application of a compacted test set with one

Bench- mark	no random patterns			10,000 random patterns	
	compact	non-compact	tailored	non-compact [12]	tailored
s5378	9,148	3,539	1,449	554	501
s9234.1	17,423	12,738	3,647	7,932	2,346
s13207.1	20,558	5,211	3,522	1,963	1,672
s15850.1	29,994	9,365	4,651	5,244	2,872
s35932	23,270	2,207	239	0	0
s38417	111,447	44,260	9,163	31,656	8,412
s38584.1	84,317	12,248	5,383	3,466	2,927

TABLE II
IMPACT OF ATPG ON NUMBER OF BITS TO BE STORED IN RESPIN ARCHITECTURE.

Bench- mark	#PPIs	#patterns compact ATPG	#Bits in patterns (specified)	#Bits RESPIN +tailored ATPG	test data reduction	test application time
s420.1	34	76	2,584 (1,187)	513	80%	19.8
s838.1	66	159	10,494 (4,049)	1179	89%	11.3
s1196	32	141	4,512 (1,957)	737	85%	15.7
s1238	32	153	4,896 (2,095)	706	84%	15.1
s5378	214	117	25,038 (6,173)	1,449	94%	5.80
s9234.1	247	148	36,556 (9,857)	3,647	90%	10.2
s13207.1	700	240	168,000 (11,003)	3,522	98%	2.11
s15850.1	611	119	72,709 (12,470)	4,651	94%	6.41
s35932	1,763	17	29,971 (16,883)	239	99%	5.92
s38417	1,664	93	154,752 (38,723)	9,163	94%	2.81
s38584.1	1,464	133	194,712 (33,168)	5,383	97%	0.760

TABLE III
COMPARISON WITH COMPACTED TEST SET.

external pin. In the RESPIN architecture, the number of scan chains per core is given by $\lceil \#PPIs/100 \rceil$.

When testing only one core at a time, the test application time is larger for the RESPIN architecture than for compact test sets. We observed, however, that the test time overhead decreased for the larger benchmarks circuits. Benchmark s38584.1 requires even less test time than the compact test. Since up to $\hat{I}_C + 1$ cores (101 in the experiments) may be tested concurrently with the same one bit narrow TAM, the RESPIN architecture always reduces test application time, if a sufficient number of cores are tested concurrently. Thus the last column in Table III gives also the minimum number of cores which need to share their narrow TAMs in order to have a reduction of test application time.

For the large ISCAS89 benchmark circuits the fraction of bits specified in a compacted test set is less than 25%, i.e. 75% of the test data volume is randomly filled. These random data are stored on the ATE and transported from ATE to the chip through test pins.

The volume of the RESPIN encoded test data in bit is even about 60% below the number of specified bits in a compacted test set using traditional compaction. For the larger benchmarks

the volume of a compacted test set generated by a commercial ATPG tool is reduced by significantly more than 90%, if the RESPIN architecture with tailored ATPG is used.

C. Comparison with BIST Generators

RESPIN requires only minor changes of the SoC hardware and test wrapper architecture. The ETC acts as a simple filter and dedicated hardware pattern generators and control are not needed. Despite its very simple structure, RESPIN outperforms even classical BIST techniques in terms of test application time and chip area necessary to store the encoded test data.

Table IV and V compare the RESPIN+ATPG with BIST techniques. A multi-polynomial, multi-seed random BIST scheme emulated on a CPU [8] applies 10,000 random patterns and encodes patterns for the remaining faults in seeds for LFSRs. The scheme is emulated on a microprocessor, the resulting test data volume in bits is shown in column JETTA98. Methods which reuse existing accumulator structures to generate test patterns are shown in column DATE00 [22] and ITC98 [23]. DATE00 implements a genetic algorithm, which computes the seeds for an accumulator. When the accumulator is seeded and

running, it generates patterns which achieve full coverage for the CUT. The results for an LFSR-based TPG using variable length seeds and reusing part of the scan chain of the CUT are shown in column TOC98. In contrary to the previous methods, the pattern generator is added as a dedicated hardware structure to the CUT [7]. Additionally the scan chain of the CUT is partly reused for pattern decompression. Column RESPIN+ATPG shows the experimental results when ATPG is used to generate tailored test patterns for RESPIN.

Bench-mark	JETTA98	DATE00	ITC98	TOC98	RESPIN+ATPG
s420.1	503	408	476	-	513
s838.1	3,246	1,452	3,432	-	1,179
s5378	759	3,262	3,424	-	1,449
s9234.1	11,766	24,700	7,410	5,346	3,647
s13207.1	1,796	-	12,600	5,877	3,522
s15850.1	11,826	-	15,886	6,316	4,651
s38417	71,491	-	73,216	16,797	9,163
s38584.1	11,529	-	43,920	3,996	5,383

TABLE IV

ENCODED TEST INFORMATION IN BITS OF SEVERAL DECOMPRESSION METHODOLOGIES.

Table IV shows the test data volume of the encoded test data in bits for the different techniques. The RESPIN approach with tailored ATPG is only outperformed in terms of test data volume in few cases. But there is no single technique, which performs significantly better in most cases.

Table V shows the applied test patterns corresponding to the test application time for the compared techniques. For one benchmark (s9234.1) and technique (DATE00) out of 25 comparisons, the proposed technique requires a larger number of test patterns, but in this case the proposed approach requires a by a factor of five lower encoded test data volume. For all the other comparison the proposed technique requires a shorter test length.

For 8 out of 25 cases, the proposed technique has either the lowest volume of encoded test data or the lowest test application time. For the remaining 17 out of 25 comparisons, RESPIN has both the lowest test application time and the lowest volume of the encoded test data at the same time. E.g. for s13207.1 the test application time is more than 60% less than for the other listed techniques and the test data volume is reduced by 40 through 70%.

The experimental results of this section show a drastic reduction of the test data volume to be passed from the ATE to the serial input of the ETC. In addition, the number of test inputs is reduced compared to applying compacted test sets to standard multiple scan path design. On the other hand, test time may increase compared with compact test sets while it is still

Bench-mark	JETTA98	DATE00	ITC98	TOC98	RESPIN+ATPG
s420.1	>10,000	3,000	10,000	-	513
s838.1	>10,000	2,000	10,000	-	1,179
s5378	>10,000	5,000	10,000	-	1,449
s9234.1	>10,000	2,000	10,000	>10,000	3,647
s13207.1	>10,000	-	10,000	>10,000	3,522
s15850.1	>10,000	-	10,000	>10,000	4,651
s38417	>10,000	-	10,000	>10,000	9,163
s38584.1	>10,000	-	10,000	>10,000	5,383

TABLE V

TEST LENGTH IN NUMBER OF APPLIED PATTERNS.

lower than the test lengths of the other encoding schemes. The reduction of both test data and bandwidth requirements allow concurrent testing of multiple cores. Putting this together will give benefits in all the three terms: bandwidth, memory requirement, and test time.

V. CONCLUSIONS

In this paper, a technique which uses standard ATPG tools to generate a tailored test set for the RESPIN architecture was presented. This tailored test set reduces the test data volume required by RESPIN up to 99% of a statically and dynamically compacted test set. Even if this new test architecture is compared with BIST techniques, the encoded test data volume and the test application time is significantly lower compared to nearly all the examined benchmarks and techniques.

ACKNOWLEDGEMENT

The authors wish to thank our students Pattara Kiatisevi and Georgi Outcharov for implementing the programs and performing the experiments.

REFERENCES

- [1] Semiconductor Industry Association, *The International Technology Roadmap for Semiconductors (ITRS)*. Austin, TX: International SEMATECH, 1999.
- [2] P. Goel and B. C. Rosales, "Test Generation and Dynamic Compaction of Tests," in *Digest of Papers Test Conference*, pp. 189–192, 1979.
- [3] I. Hamzaoglu and J. H. Patel, "Test Set Compaction Algorithms for Combinational Circuits," in *Proceedings of the International Conference on CAD (ICCAD)*, November 1998.
- [4] G. Tromp, "Minimal Test Sets for Combinatorial Circuits," in *Proceedings of the IEEE International Test Conference (ITC)*, (Washington, DC), pp. 204–209, IEEE, IEEE Computer Society Press, 1991.
- [5] I. Pomeranz, L. N. Reddy, and S. M. Reddy, "Compactest: A Method to generate Compact Test Sets for Combinational Circuits," in *Proceedings of the IEEE International Test Conference (ITC)*, (Washington, DC), pp. 194–203, IEEE Computer Society Press, 1991.
- [6] H. Huguichi, N. Ishiura, and S. Yajima, "Compaction of Test Sets Based on Symbolic Fault Simulation," in *Synthesis and Simulation Meeting and International Interchange*, pp. 253–262, 1992.

- [7] J. Rajski, J. Tyszer, and N. Zacharia, "Test Data Decompression for Multiple Scan Designs with Boundary Scan," *IEEE Transactions on Computers*, vol. 47, pp. 1188–1200, November 1998.
- [8] S. Hellebrand, H.-J. Wunderlich, and A. Hertwig, "Mixed-Mode BIST Using Embedded Processors," *Journal of Electronic Testing Theory and Applications (JETTA)*, vol. 12, no. 1/2, pp. 127–138, 1998.
- [9] B. Koenemann, "LFSR-Coded Test Patterns for Scan Design," in *Proceedings of the European Test Conference (ETC)*, (München), pp. 237–242, 1991.
- [10] S. Hellebrand, S. Tarnick, J. Rajski, and B. Courtois, "Generation of Vector Patterns through Reseeding of Multiple-Polynomial Linear Feedback Shift Registers," in *Proceedings of the IEEE International Test Conference (ITC)*, (Washington, DC), pp. 120–129, IEEE, IEEE Computer Society Press, 1992.
- [11] S. Hellebrand, B. Reeb, S. Tarnick, and H.-J. Wunderlich, "Pattern Generation for a Deterministic BIST Scheme," in *Proceedings of the International Conference on CAD (ICCAD)*, pp. 88–94, ACM/IEEE, November 1995.
- [12] R. Dorsch and H.-J. Wunderlich, "Reusing scan chains for test pattern decompression." Accepted for presentation at ETW01, <http://www.ra.informatik.uni-stuttgart.de/~rainer/etw01.pdf>, May 2001.
- [13] E. J. Marinissen, Y. Zorian, R. Kapur, T. Taylor, and L. Whetsel, "Towards a Standard for Embedded Core Test: An Example," in *Proceedings of the IEEE International Test Conference (ITC)*, pp. 616–627, IEEE, 1999.
- [14] R. Frohwerk, "Signature Analysis, A New Digital Field Service Method," *Hewlett Packard Journal*, vol. 28, no. 9, pp. 2–8, 1977.
- [15] A. P. Ströle, "Bit Serial Pattern Generation and Response Compaction Using Arithmetic Functions," in *Proceedings of the VLSI Test Symposium (VTS)*, pp. 78–84, IEEE, 1998.
- [16] J. Rajski and J. Tyszer, "Test Responses Compaction in Accumulators with Rotate Carry Adders," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, pp. 531–539, April 1993.
- [17] A. P. Ströle, "Test Response Compaction using Arithmetic Functions," in *Proceedings of the VLSI Test Symposium (VTS)*, pp. 380–386, IEEE, 1996.
- [18] J. Rajski and J. Tyszer, "Accumulator-Based Compaction of Test Responses," *IEEE Transactions on Computers*, vol. 42, pp. 643–650, June 1993.
- [19] M. Konijnenburg, J. van der Linden, and V. van de Goor, "Test Pattern Generation with Restrictors," in *Proceedings of the IEEE International Test Conference (ITC)*, pp. 598–605, IEEE, 1993.
- [20] P. Wohl and J. Waicukauski, "Test generation for ultra-large circuits using ATPG constraints and test-pattern templates," in *Proceedings of the IEEE International Test Conference (ITC)*, pp. 13–20, October 1996.
- [21] F. Brglez, D. Bryan, and K. Kozminski, "Combinatorial Profiles of Sequential Benchmark Circuits," in *Proceedings of the International Symposium on Circuits and Systems (ISCAS)*, pp. 1229–1234, IEEE, 1989.
- [22] S. Cataldo, S. Chiusano, P. Prinetto, and H.-J. Wunderlich, "Optimal Hardware Pattern Generation for Functional BIST," in *Proceedings of the Design Automation and Test in Europe (DATE)*, 2000.
- [23] R. Dorsch and H.-J. Wunderlich, "Accumulator Based Deterministic BIST," in *Proceedings of the IEEE International Test Conference (ITC)*, pp. 412–421, 1998.