Core-Level Compression Technique Selection and SOC Test Architecture Design[1]

Anders Larsson[+], Xin Zhang[+], Erik Larsson[+], and Krishnendu Chakrabarty[*]

[+]Department of Computer Science
Linköping University
SE-582 83 Linköping, Sweden

*Department of Electrical and Computer Engineering
Duke University
Durham NC 27708, USA

***Abstract* —The increasing test-data volumes needed for the testing of system-on-chip (SOC) integrated circuits lead to long test-application times and high tester memory requirements. Efficient test planning and test-data compression are therefore needed. We present an analysis to highlight the fact that the impact of a test-data compression technique on test time and compression ratio are method-dependant as well as TAM-width dependant. This implies that for a given set of compression schemes, there is no compression scheme that is the optimal with respect to test time reduction and test-data compression at all TAM widths. We therefore propose a technique where we integrate core wrapper design, test architecture design and test scheduling with test-data compression technique selection for each core in order to minimize the SOC test-application time and the test-data volume. Experimental results for several SOCs crafted from industrial cores demonstrate that the proposed method leads to significant reduction in test-data volume and test time.**

1. Introduction

Advances in semiconductor technology have led to the emergence of complex system-on-chip (SOC) integrated circuits (ICs). For timely design of such ICs, it is increasingly common to make use of a core-based design flow where pre-designed and pre-verified blocks of logic, so called cores, are used as building blocks.

All ICs must be tested for defect screening. The cost of manufacturing test is increasing rapidly since chip complexities and new process technologies are leading to higher test-data volumes. The 2007 ITRS Final Draft predicts that the test-data volume for integrated circuits will be as much as 38 times larger and the test-application time will be about 17 times larger in 2015 than it is today [1]. High test-data volumes lead to long test-application times and high Automatic Test Equipment (ATE) memory requirements.

Test planning and test-data compression address high test-data volumes and long test-application times. Modularly-designed SOCs can be tested in a modular manner. Several test planning techniques that defines the test architecture and the order of tests have been proposed [2]–[6] The main objective for these techniques is to reduce the test-application times, however, they also lead to a corresponding reduction of test-data to be stored on the ATE.

Test-data compression is efficient to reduce test-data volume and test-application time [7]–[10].The benefits of test-data compression can be further enhanced if the compressed patterns are delivered to the decompressors using an efficient SOC-level test-access architecture and test schedule. The combination of test-data compression (at the core-level), and TAM optimization and test scheduling (at the SOC-level) can therefore be used to further reduce test-data volume and test time.

A number of methods have been proposed to combine core-level test-data compression with SOC-level test planning [11, 12]. As expected, these techniques show that test-data compression leads to a reduction in test time for the core-based SOC. However, they do not provide any quantitative insights on the test-time reduction (at the SOC-level) derived from adding a decompressor for any given embedded core. Therefore, we recently proposed a method for core-level expansion of compressed test patterns and test architecture optimization at SOC-level. The work reported in [13] was, however, limited to test-application time minimization and to a single given compression technique only.

In this paper, we analyze the test time and test-data compression ratio for three compression techniques. For a given core, we find for each technique, different characteristics on compression ratio and test time. As these characteristics depend on the bitwidth assigned to a core, it is difficult to find the optimal bitwidth for each individual core in an SOC when the test architecture is to be designed. We therefore present an optimization technique that for a given SOC, finds the best test-data compression technique for each core, designs the core wrapper, defines the test architecture, and schedules the tests such that the SOC's overall test-application time and test-data volume are minimized.

The rest of this paper is organized as follows. Section 2 contains background and prior work. In Section 3, we discuss test-data compression techniques. In Section 4, we formulate the problem and in Section 5 the proposed algorithm is described. Experimental results are presented in Section 6, and finally, Section 7 concludes the paper.

2. Background and Prior Work

Modular SOCs can be tested in a modular fashion by making use of the IEEE 1500 Std. [14]. By embedding cores in wrappers, each core can be tested as an individual unit. The test architecture is responsible for the on-chip transportation of test stimuli from chip pins to individual cores, and for the transportation of test responses from cores to chip pins.

Iyengar *et al*. defined a wrapper design algorithm that for a given core, groups the scan-chains and wrapper cells into a given number of so-called wrapper chains [3]. As the test time for a core behaves

[1] The research is partially supported by the Strategic Integrated Electronic Systems Research (STRINGENT) program founded by the Swedish Foundation for Strategic Research (SSF).

1081-7735/08 $25.00 © 2008 IEEE
DOI 10.1109/ATS.2008.71

as a staircase when the number of wrapper-chains increases, it is difficult to assign the best number of wrapper-chains to each and every core in an SOC. Iyengar *et al*. [3] therefore defined a SOC-level heuristic where the test architecture and test schedule are defined. A typical output is shown in Figure 1 where the W ATE channels ($W = 32$) for test stimuli delivery are divided into three test access mechanisms (TAMs) of width w_1, w_2, and w_3, and each core is assigned to one TAM. A number of test planning approaches have been proposed [2]–[6].

Test-data compression is effective for reducing the ATE memory requirement and the test time. The general idea is to fill the large number of don't-care bits in the test stimuli such that high compression is achieved. The compressed test stimuli are stored in the ATE and decoded during test application. The produced test responses can be compacted on-chip. There are a number of different test-data compression techniques [7]–[10].

The test-data compression scheme Selective Encoding (SE) [7] makes use of on-chip decoders to expand the compressed test stimuli. The w input bits (TAM width) are expanded to m scan-chains (see Figure 2) [7]. The expansion means that more scan-chains can be used compared to when SE is not used, hence, the scan-chains are shorter and therefore the test time is lower. For SE, w is given as: $w = \lceil \log_2(m+1) \rceil + 2$. For modular SOCs, it is favourable to place the decompressor for a core near the core as it reduces routing cost as $w < m$ (see Figure 2).

SE works in two modes: single-bit-mode or group-copy-mode. In single-bit-mode, each bit in a slice (m bits) is indexed from 0 to m, and the position of the target symbol is encoded. For example, the target symbol of 1 in the slice "XXX1000" is encoded as "0011" as it is positioned at index 3. In group-copy-mode the m-bit slice is divided into $m / \lceil \log_2(m+1) \rceil$ groups. Two codewords are needed to encode one group where the first codeword specifies the index of the first bit in the group, and the second codeword contains the test-data. The hardware cost for the SE test-data compression technique is small. For larger than million-gate designs, the number of required flip-flops corresponds to a hardware cost of only 1% [7].

In the compression scheme Vector Repeat (VR), there is no decoder logic as VR tries to find overlapping patterns that can be repeated for a number of clock cycles [10]. The decoding is embedded in the ATE test program. As VR does not expand its stimuli, it is only able to achieve compression in the space domain and not in the time domain. SE performs compression both in space and time.

Methods that combine core-level test-data compression with SOC-level test planning lead to reduction in test time [5, 11, 12]. As these techniques are ad hoc and straight forward, we have in detail analyzed the test-data compression scheme SE [13].

A number of industrial cores were analyzed with respect to test time [13]. For every core, we considered all possible values of w and m and evaluated the test time $\tau(w, m)$. The results show that when the goal is to find the lowest test time for a core, it is not sufficient to simply assign a large number of wrapper chains. In fact, the test time varies much between the best w and the worst, and test time can increase with an increase in the number of wrapper chains. Hence, it is difficult to find an optimal w and m pair to design the optimal decoder for a core [13]. Furthermore, for an SOC, it is not trivial to find the best decoder for each core such that the test architecture leads to minimal test time. The proposed technique however considered only SE [13] and it was limited to the minimization of test-application time. In this paper, we analyze SE, VR, and the combination of SE and VR, and in contrast to [13] test-application time and test-data volume are minimized.

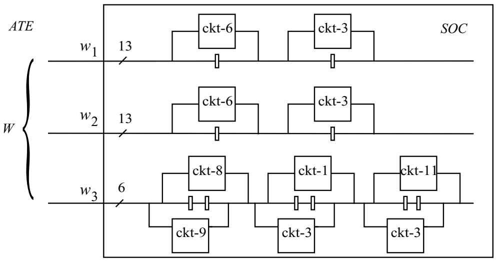


Figure 1. A typical test architecture design where each core is assigned to one TAM.

3. Analysis of Compression Techniques

In this section, we analyze the test time and test-data volume for test-data compression techniques at various TAM widths. We have made use of Selective Encoding (SE), Vector Repeat (VR), and the combination of SE and VR (SE_VR). For SE_VR, the test-data is first compressed using SE and then compressed by applying VR.

For the experiments, we have implemented the wrapper design algorithm proposed by Iyengar *et al*. [3] and for the filling of don't-care bits, we have made use of minimum-transition fill. The details are as follows. First, the wrapper chains have been formed for a given TAM width, and then, based on the wrapper design, the test stimuli bits are arranged accordingly and filled according to the minimum transition fill scheme. After that, the test stimuli are compressed. The experiments have been repeated for each of the three approaches (SE, VR, SE_VR) at each TAM width.

The experiments have been performed on the cores in d695 [15] and on the industrial cores [7]. In this paper, we show, due to space limitations, the results for the d695 core s9234 and the industrial core ckt-7. The results in test time at various bandwidths are for the two cores presented in Figure 3, and the results in test-data volume are for the two cores reported in Figure 4.

The results in Figure 3 show that the test time decreases as expected for all test-data compression techniques when the TAM width is increased. The test times for SE and SE_VR are always the same as VR does only compress in space domain and no compression is performed in the time domain. When comparing the test time for the three compression techniques, VR is better for lower TAM widths while SE and SE_VR are better for wider TAM widths. Further, SE cannot be applied to a narrow TAM as the technique requires a minimum of three TAM wires. SE requires two TAM wires for control of the decompressor, hence a minimum of three TAM wires are required for one wrapper chain. In summary, there is no compression technique that produces test times that is best for any TAM width. For example, in Figure 3(a), the test time of VR is better than that of SE and SE_VR at TAM width 4 while at TAM width 8 it is the other way around.

The test-data volumes, obtained using different compression techniques at various TAM widths, are shown in Figure 4. Figure 4

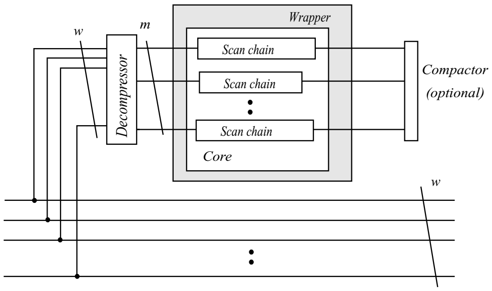


Figure 2. Test-data compression for a wrapped core.

shows that the test-data volume is not constant for various TAM widths. For VR, the compression ratio decreases and the compressed test data volume increases for wider TAMs. This is explained by the fact that it is more difficult to find overlapping vectors when the slice (TAM width) increases. The test-data volume decreases for SE as TAM width increases. However, as shown in Figure 4(b), the compression ratio gets worse for wider TAMs. To summarize, test-data compression ratio depends on TAM width, and there is no compression technique that produces the best test-data volume for any TAM width.

This analysis of test-data compression schemes behaviour in terms of test time and compression ratio shows that:

- it is not trivial to select the test-data compression scheme that produces the lowest test time and the best compression ratio, and
- for a core-based SOC, where the test architecture is to be designed and several cores are to be assigned to the same TAM, it is not trivial to find the TAM widths such that it fits all cores the best.

Therefore, there is a need to include the selection of test-data compression scheme when the test architecture and test schedule are defined in order to minimize the SOC's overall test-application time and the test-data volume.

4. Problem Formulation

We divide the problem into two parts; core-level and SOC-level. The problem at core-level is to find the number of TAM wires (w), the number of wrapper-chains (m), and the compression technique c, which gives the minimum cost in terms of test time and test-data volume. For each core i in the SOC, the following is given:

- sc_i - the number of scan chains,
- ff_{ij} - the number of flip-flops in scan chain j, where $j = \{1, 2, ..., sc_i\}$,
- wi_i - the number of input wrapper cells,
- wo_i - the number of output wrapper cells,
- $nff_i = \sum_{j=1}^{sc_i} ff_{ij}$ - the total number of flip-flops,
- $T_i = (ts_{i1}, ..., ts_{il})$ - the test stimuli consisting of a sequence of l test patterns, where ts_{ik} consists of $nff_i + wi_i$ bits and each bit can be 0, 1, or x, x is a don't-care bit.

We assume that a set of compression techniques is available:

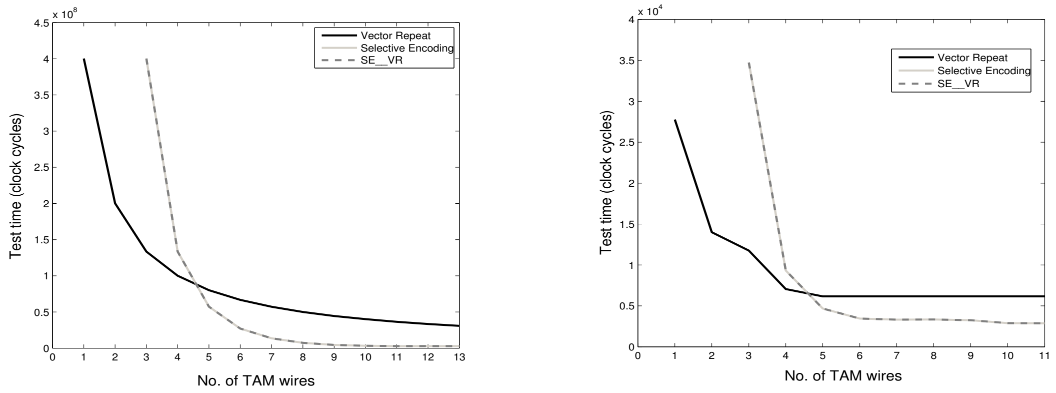
$C = \{nc, SE, VR, SE\_VR\}$ - the compression technique alternatives where nc denotes no compression, SE denotes Selective Encoding, VR denotes Vector Repeat, and SE_VR denotes Selective Encoding followed by Vector Repeat.

For each compression technique c, where $c \in C$, we can easily determine:

- $\tau_i(w, m, c)$ - the test time using test-data compression technique c at w number of TAM wires and m number of wrapper-chains.
- $\mu_i(w, m, c)$ - the compressed test-data volume using compression technique c at w number of TAM wires and m number of wrapper-chains.

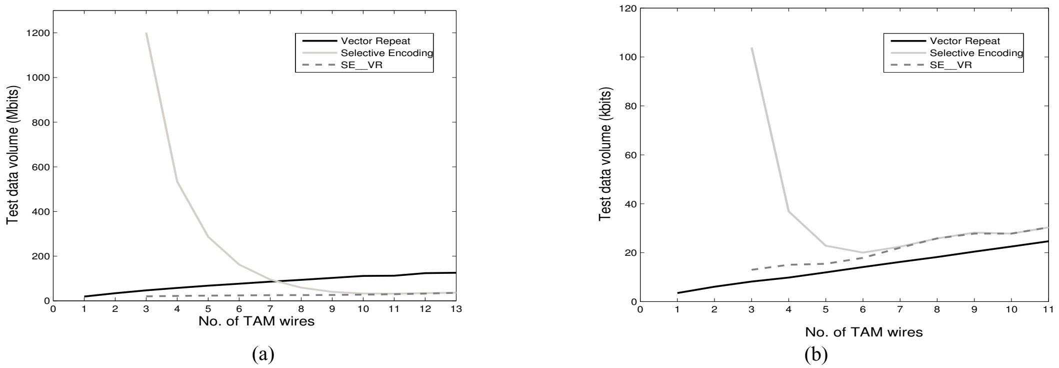
For techniques such as SE, w is the TAM width and the core's decoder input, and m is the decoder output and the number of wrapper-chains). For techniques such as VR and nc where no decoder is used, $m = w$, while for SE, m is an input parameter.

Given $\tau_i(w, m, c)$ and $\mu_i(w, m, c)$, the cost $Cost_i(w, m, c)$ for a core i at w number of TAM wires, m wrapper-chains using compression



(a) (b)

Figure 3. Test time obtained using different compression techniques at various TAM widths for (a) ckt-7 and (b) s9234



(a) (b)

Figure 4. Test-data volume using different compression techniques at various TAM widths for (a) ckt-7 and (b) s9234

technique c is:

$$Cost_i(w, c) = \alpha \times \tau_i(w, m, c) + \beta \times \mu_i(w, m, c), \qquad (1)$$

where α ($\{0 \le \alpha \le 1\}$) and β ($\{0 \le \beta \le 1\}$) are used to set the weight of the test time and the test-data volume, respectively. The value of α and β are set such that $\alpha + \beta = 1$. The minimum cost $MinCost_i$ for a core i is finally given as:

$$MinCost_i = \min\{Cost_i(w, m, c)\}, \forall w \forall m \forall c \qquad (2)$$

The core-level problem is not trivial as the lowest test time for a given m, w, and c does not necessarily result in the minimum test-data volume. At the SOC-level, the problem becomes harder. Not only must the test time and the test-data volume for each core be optimized, the test architecture design, that is the number and width of the TAMs and core assignment to a each TAM, and the test schedule must also be considered such that the overall test time and test-data volume are minimal.

The SOC-level problem Ψ_{SOC} is formulated as follows:
For a given SOC with a given TAM width W, partition the TAM and determine each TAM's width, assign the cores to the TAMs, and select a compression technique for each core, such that the system's cost is minimized. The system's cost $Cost_{SOC}$ is given by:

$$Cost_{SOC} = \alpha \times \tau_{tot} + \beta \times \mu_{tot} \qquad (3)$$

The test time τ_{tot} for a test schedule with n cores is:

$$\tau_{tot} = \max\{t_i + \tau_i(w, m, c)\}, \forall i, i \in \{1, 2, \ldots, n\}, \qquad (4)$$

where t_i is the start time when the test is applied to the core c_i, and the total test-data volume for the SOC with n cores is:

$$\mu_{tot} = \sum_{i=1}^{n} \mu_i(w, m, c) \qquad (5)$$

5. Proposed Algorithm

A heuristic technique is proposed to solve the problem Ψ_{SOC}. A pre-processing stage is used to generate, for each core, a number of wrapper and decompression design alternatives. For the wrapper design we have made use of the optimization heuristic from [3]. The don't-care bits are then filled according to minimum transition fill, and finally compression technique c is applied. For the decompressor design, we make use of the SE [7], the VR [10] and the SE_VR techniques. We generate all alternatives for the decompressor input/output mapping. For each compression technique c and each combination of w and m, we have for core i the test time $\tau_i(w, m, c)$.

The proposed algorithm consists of three procedures; initialization, compression technique selection, and test architecture design and test scheduling. The three procedures are executed as illustrated in Figure 5.

In the initialization procedure, the initial test architecture (TAM design and wrapper design) and test schedule are designed. The test transportation is sequential for each TAM. Hence, the test time τ_{tam} for a TAM connected to n cores is given as:

$$\tau_{tam} = \sum_{i=1}^{n} \tau_i(c, m, w) \qquad (6)$$

The initial number of TAMs is determined such that each core is assigned to at least one TAM wire.

In the compression technique selection procedure, a compression technique is selected for each core. Three loops are used for the selection of test compression technique alternative. The first loop i is used to iterate over the k TAMs, the second loop j iterates over the cores that are connected to TAM i, and finally, the third loop iterates over the available compression technique alternatives C. For each core, the selected compression technique is accepted if the cost is reduced. Finally, the test architecture design and test schedule procedure and the compression selection procedure are used in an optimization loop.

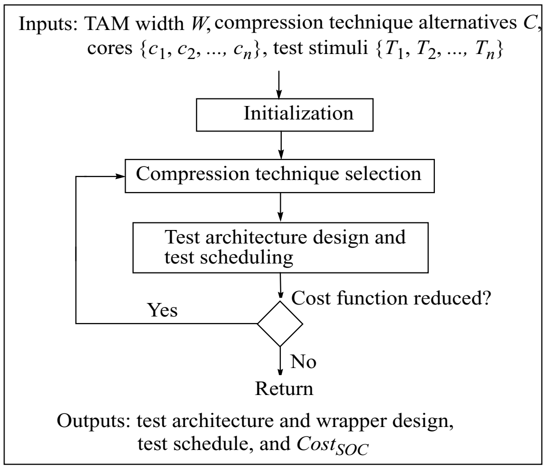

Inputs: TAM width W, compression technique alternatives C,
cores $\{c_1, c_2, ..., c_n\}$, test stimuli $\{T_1, T_2, ..., T_n\}$
Initialization
Compression technique selection
Test architecture design and test scheduling
Cost function reduced?
Yes
No
Return
Outputs: test architecture and wrapper design,
test schedule, and $Cost_{SOC}$

Figure 5. Flowchart of the proposed algorithm.

For all iterations in the optimization loop, a modified solution is generated and evaluated. From the current TAM architecture a new TAM architecture alternative is generated by merging TAMs. The merging of TAMs is carried out as follows. Consider two TAMs i and TAM j as candidates for a merge. A new TAM, with w_{new} TAM wires, will be generated where $w_{new} = w_i + w_j$. All cores, that previous to the merge were assigned to TAMs i and TAM j, will after the merge be assigned to the new TAM.

The optimization loop is stopped when no new TAM architecture alternative and test schedule can be generated such that the system's cost is reduced.

6. Experimental Results

We have implemented the proposed algorithm described above and we have carried out experiments on the benchmark design d695 [15], and on three designs, System1, System2, and System3, crafted using industrial cores, which are described in detail in [7]. The characteristic of each design is presented in Table 1. Column 1 lists the design and Column 2 lists the number of cores. Column 3 and Column 4 list the number of flip-flops and the initial given test-data volume, respectively.

We minimize each system's cost $Cost_{SOC}$ for various TAM widths W. For each TAM-width constraint, we run three different experiments: $\alpha = 1$ and $\beta = 0$, corresponding to test time minimization; $\alpha = 0$ and $\beta = 1$, corresponding to test-data volume minimization; $\alpha = 0.5$ and $\beta = 0.5$.

We compare our proposed algorithm (PA) with four other approaches; no compression (nc), only Vector Repeat (VR), only Selective Encoding (SE), and only combined Selective Encoding and Vector Repeat (SE_VR).

The results from the experiments for System1 to System3 and for d695 are collected in Table 2 to Table 5, respectively. Tables 2-5 show the test time τ_{tot} and test-data volume μ_{tot} for each system at various TAM widths. The results in Tables 2-5 are organized as follows. Column 1 lists the compression technique used, Column 2 lists the test time and data factors, and Column 3 lists the TAM width constraint. Column 4 and Column 5 list the test time and test-data

TABLE 1
Design characteristics

Design	No. of cores	No. of flip-flops	Initial given (uncompressed) test-data volume (Mbits)
System1	30	739,743	62,404
System2	60	1,479,486	124,808
System3	100	2,465,810	208,014
d695	10	6,348	0.34

volume for each compression technique. The last two columns highlight the comparisons. Column 6 lists the comparison between the test times τ_{tot} obtained when test-data compression is used and the test time τ_{nc} obtained when test-data compression is not used is not used. Column 7 lists the comparison between the test-data volumes μ_{tot} obtained using test-data compression to the test-data volume μ_{nc} obtained when test-data compression is not used.

The results highlight the importance of co-optimizing test-data compression selection, test architecture design, and test scheduling for SOCs. On average, our proposed approach, with test-data compression selection, results in a 6.14x reduction in test time (when only the test time is considered in the optimization). The corresponding reduction in test-data volume is on average 26.56x. When both the test time and the test-data volume were optimized, the test time was reduced by 4.29x and the test-data volume was reduced by 13.84x.

We have also preformed experiments that show the test time, test-data volume, and system's cost at various values of α (and β). The results of the experiments for System1 at TAM width 32 are presented in Figure 6. The test time is long when $\alpha = 0$, corresponding to test-data volume minimization. The test time is decreased as α is increased. In contrast to the test time, the results for the test-data volume show an increased test-data volume as α is increased. The overall system's cost is slightly increased as α is increased and reaches a maximum when $\alpha = 0.2$. For values larger than 0.2 the system's cost is reduced and, as expected, converges to the test-time when $\alpha = 1$.

The proposed algorithm assumes that wrapper design are available for all compression techniques, and for all TAM/wrapper chains alternatives for the different cores. The process of generating these alternatives is quite time consuming. However, once this information is available, our algorithm is computationally effective. The cpu-time (execution time to produce the solutions, excluding the time for core wrapper design and test-data compression) is very short. For the largest design, System3 and the widest TAM width constraint W = 32, the CPU-time was less than 1 second.

7. Conclusion

The high amount of test-data volumes needed for manufacturing test of System-on-chip (SOC) integrated circuits (ICs) leads to long test times and high Automatic Test Equipment (ATE) memory requirements. In this paper, we have analyzed the test time and test-data compression ratio for the test-data compression schemes Selective Encoding, Vector Repeat and the combination of Selective Encoding and Vector Repeat for a number of ISCAS cores and industrial cores. The analysis shows that the test time and the test-data compression ratio are method dependant as well as TAM width dependant. It is therefore not trivial to select the optimal compression scheme for a core. Further, the behavior on test time and test-data compression ratio are independent; hence it is difficult to select the optimal TAM width for a given core such that both test time and compressed test-data are minimal. The problem becomes even more difficult for a core-based SOC as cores assigned to the same TAM must have the same bandwidth. We therefore proposed a technique to integrate test-data compression selection with test planning. Our technique selects test-data compression technique for each core, designs the core wrapper, defines the number and widths of each TAM, and schedules the testing of the cores on the test architecture such that the test-application time and the test-data volume are minimized. We have performed experiments on several SOCs that are crafted from industrial cores. The experimental results demonstrate that the proposed method leads to significant reduction in test-data volume, on average 26.56x, and test time, on average 6.14x.

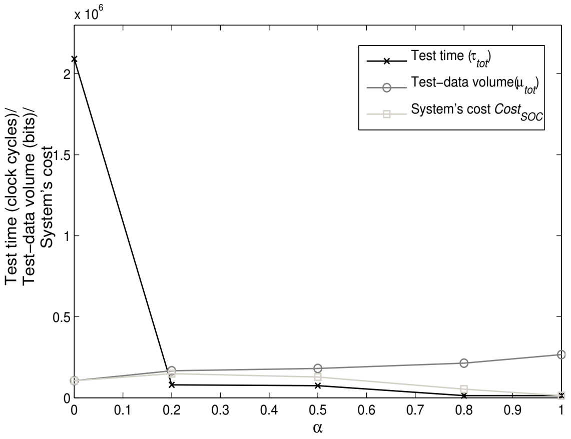


Figure 6. Test time, test-data volume, and system's cost for System1 using the proposed algorithm at various values of α at TAM width 32, ($\beta = 1 - \alpha$).

References

[1] Semiconductor Industry Association. (2007) International Technology Roadmap for Semiconductors (ITRS), http://www.itrs.net/Links/2007ITRS/Home2007.htm.

[2] V. lyengar, K. Chakrabarty, and E. J. Marinissen, "Test access mechanism optimization, test scheduling, and tester data volume reduction for system-on-chip," *IEEE Trans. on Comp.*, Vol. 52, Issue 12, pp. 1619–1632, 2003.

[3] V. Iyengar, K. Chakrabarty, and E. J. Marinissen, "Test Wrapper and Test Access Mechanism Co-Optimization for System-on-Chip," *Proc. Int. Test Conf.*, pp. 1023–1032, 2001.

[4] S. K. Goel and E. J. Marinissen, "SOC test architecture design for efficient utilization of test bandwidth," *ACM Trans. Design Automation of Electronic Systems*, Vol. 8, Issue 4, pp. 399–429, 2003.

[5] X. Qiang and N. Nicolici, "Multi-frequency test access mechanism design for modular SOC testing," *Proc. Asian Test Symp.*, pp. 2–7, 2004.

[6] V. Iyengar, K. Chakrabarty, and E. J. Marinissen, "Test access mechanism optimization, test scheduling, and tester data volume reduction for system-on-chip," *IEEE Trans. Comp.*, Vol. 52, Issue 12, pp. 1619–1632, 2003.

[7] Z. Wang and K. Chakrabarty, "Test Data Compression for IP Embedded Cores Using Selective Encoding of Scan Slices," *Proc. Int. Test Conf.*, pp. 581–590, 2005.

[8] A. Chandra and K. Chakrabarty, "A unified approach to reduce SOC test data volume, scan power and testing time," *IEEE Trans. CAD*, Vol. 22, Issue 3, pp. 352–363, 2003.

[9] S. Kajihara and K. Miyase, "On Identifying Don't Care Inputs of Test Patterns for Combinational Circuits," *Int. Conf. CAD*, pp. 364–369, 2001.

[10] C. Barnhart, V. Brunkhorst, F. Distler, O. Farnsworth, B.Keller, and B. Koenemann, "OPMISR: The Foundation for Compressed ATPG Vectors", *Proc. Int. Test Conf.*, pp. 748–757, 2001.

[11] P. T. Gonciari and B. M. Al-Hashimi, "A compression-driven test access mechanism design approach," *Proc. European Test Symp.*, pp. 100–105, 2004.

[12] V. Iyengar and A. Chandra, "Unified SOC test approach based on test data compression and TAM design," *IEE Proc. Computer and Digital Techniques*, Vol. 152, Issue 1, pp. 82–88, 2005.

[13] A. Larsson, E. Larsson, K. Chakrabarty, P. Eles, and Z. Peng, "Test Architecture Optimization and Test Scheduling for SOCs with Core Level Expansion of Compressed Test Patterns", *Proc. Design, Automations and Test in Europe*, pp. 188–193, 2008.

[14] IEEE 1500 Standard for Embedded Core Test (SECT), *http://grouper.ieee.org/groups/1500*, 2007.

[15] E. J. Marinissen, Y. Iyengar, K. Chakrabarty, "A Set of Benchmarks for Modular Testing of SOCs," *Proc. Int. Test Conf.*, pp. 519–528, 2002.

TABLE 2
Experimental results for System1

Technique	Test time, data factor (α, β)	TAM width (W)	Test time τ_{tot} (1000 clock cycles)	Test-data volume μ_{tot} (Mbits)	Comparison of test time τ_{nc}/τ_{tot}	Comparison of test-data volume μ_{nc}/μ_{tot}
nc	1, 0	8	396,478	3,170	1.00	1.00
VR			396,478	3,170	1.00	1.00
SE			59,428	474	6.67	6.68
SE_VR			59,428	190	6.67	16.67
PA			59,428	190	6.67	16.67
nc	0, 1	8	396,478	3,170	1.00	1.00
VR			3,170,080	308	0.13	10.28
SE			59,428	474	6.67	6.68
SE_VR			3,077,010	141	0.13	22.46
PA			3,169,840	106	0.13	30.05
nc	0.5, 0.5	8	396,478	3,170	1.00	1.00
VR			396,636	622	1.00	5.10
SE			59,428	474	6.67	6.68
SE_VR			59,428	190	6.67	16.67
PA			59,428	190	6.67	16.67
nc	1, 0	16	198,461	3,172	1.00	1.00
VR			198,461	3,172	1.00	1.00
SE			181,948	931	1.09	3.41
SE_VR			181,948	919	1.09	3.45
PA			22,158	275	8.96	11.54
nc	0, 1	16	198,461	3,172	1.00	1.00
VR			3,170,080	308	0.06	10.29
SE			181,948	923	1.09	3.44
SE_VR			2,038,160	141	0.10	22.48
PA			3,169,840	106	0.06	30.07
nc	0.5, 0.5	16	198,461	3,172	1.00	1.00
VR			217,030	565	0.91	5.62
SE			181,948	923	1.09	3.44
SE_VR			497,355	190	0.40	16.72
PA			33,459	201	5.93	15.81
nc	1, 0	32	99,349	3,173	1.00	1.00
VR			99,349	3,173	1.00	1.00
SE			24,284	383	4.09	8.29
SE_VR			24,284	243	4.09	13.07
PA			13,522	266	7.35	11.93
nc	0, 1	32	99,452	3,174	1.00	1.00
VR			2,091,340	308	0.05	10.29
SE			24,284	370	4.10	8.58
SE_VR			1,412,550	141	0.07	22.49
PA			2,091,120	106	0.05	30.08
nc	0.5, 0.5	32	63,729	3,170	1.00	1.00
VR			133,497	548	0.48	5.78
SE			7,513	256	8.48	12.39
SE_VR			20,994	192	3.04	16.49
PA			27,281	174	2.34	18.24

TABLE 3
Experimental results for System2

Technique	Test time, data factor (α, β)	TAM width (W)	Test time τ_{tot} (1000 clock cycles)	Test-data volume μ_{tot} (Mbits)	Comparison of test time τ_{nc}/τ_{tot}	Comparison of test-data volume μ_{nc}/μ_{tot}
nc	1, 0	8	792,751	6,339	1.00	1.00
VR			792,751	6,339	1.00	1.00
SE			118,856	949	6.67	6.68
SE_VR			118,856	380	6.67	16.67
PA			118,856	380	6.67	16.67
nc	0, 1	8	792,956	6,340	1.00	1.00
VR			6,340,150	617	0.13	10.28
SE			118,856	949	6.67	6.68
SE_VR			6,154,010	282	0.13	22.46
PA			6,339,670	211	0.13	30.05
nc	0.5, 0.5	8	792,751	6,339	1.00	1.00
VR			792,751	1,243	1.00	5.10
SE			118,856	949	6.67	6.68
SE_VR			118,856	380	6.67	16.67
PA			118,856	380	6.67	16.67
nc	1, 0	16	396,478	6,340	1.00	1.00
VR			396,478	6,340	1.00	1.00
SE			396,115	1,988	1.00	3.19
SE_VR			396,115	1,962	1.00	3.23
PA			44,315	550	8.95	11.53
nc	0, 1	16	396,922	6,345	1.00	1.00
VR			6,340,150	617	0.06	10.29
SE			396,115	1,974	1.00	3.21
SE_VR			4,658,400	282	0.09	22.48
PA			6,339,670	211	0.06	30.07
nc	0.5, 0.5	16	396,636	6,339	1.00	1.00
VR			396,636	1,243	1.00	5.10
SE			396,115	1,975	1.00	3.21
SE_VR			527,337	378	0.75	16.77
PA			66,918	401	5.93	15.79
nc	1, 0	32	198,461	6,345	1.00	1.00
VR			198,461	6,345	1.00	1.00
SE			74,375	913	2.67	6.95
SE_VR			74,375	510	2.67	12.44
PA			28,547	526	6.95	12.05
nc	0, 1	32	198,461	6,345	1.00	1.00
VR			3,525,450	617	0.06	10.29
SE			74,375	867	2.67	7.32
SE_VR			2,771,640	282	0.07	22.48
PA			3,525,400	211	0.06	30.07
nc	0.5, 0.5	32	215,936	6,340	1.00	1.00
VR			217,030	1,129	0.99	5.61
SE			74,375	867	2.90	7.31
SE_VR			60,996	392	3.54	16.17
PA			43,970	401	4.91	15.79

TABLE 4
Experimental results for System3

Technique	Test time, data factor (α, β)	TAM width (W)	Test time τ_{tot} (1000 clock cycles)	Test-data volume μ_{tot} (Mbits)	Comparison of test time τ_{nc}/τ_{tot}	Comparison of test-data volume μ_{nc}/μ_{tot}
nc	1, 0	8	1,321,250	10,566	1.00	1.00
VR			1,321,250	10,566	1.00	1.00
SE			198,093	1,581	6.67	6.68
SE_VR			198,093	634	6.67	16.67
PA			198,093	634	6.67	16.67
nc	0, 1	8	1,321,590	10,567	1.00	1.00
VR			10,566,900	1,028	0.13	10.28
SE			19,8093	1,581	6.67	6.68
SE_VR			10,256,700	470	0.13	22.46
PA			10,566,100	352	0.13	30.05
nc	0.5, 0.5	8	1,321,250	10,566	1.00	1.00
VR			1,321,250	2,072	1.00	5.10
SE			198,093	1,581	6.67	6.68
SE_VR			198,093	634	6.67	16.67
PA			198,093	634	6.67	16.67
nc	1, 0	16	660,630	10,566	1.00	1.00
VR			660,630	10,566	1.00	1.00
SE			660,622	3,299	1.00	3.20
SE_VR			660,622	3,257	1.00	3.24
PA			73,860	916	8.94	11.53
nc	0, 1	16	661,537	10,575	1.00	1.00
VR			10,566,900	1,028	0.06	10.29
SE			660,622	3,280	1.00	3.22
SE_VR			7,645,020	470	0.09	22.48
PA			10,566,100	352	0.06	30.07
nc	0.5, 0.5	16	660,630	10,566	1.00	1.00
VR			660,630	2,072	1.00	5.10
SE			660,622	3,280	1.00	3.22
SE_VR			880,523	634	0.75	16.67
PA			111,530	669	5.92	15.79
nc	1, 0	32	330,768	10,575	1.00	1.00
VR			330,768	10,575	1.00	1.00
SE			134,263	1,575	2.46	6.72
SE_VR			134,263	854	2.46	12.39
PA			42,843	916	7.72	11.54
nc	0, 1	32	330,768	10,575	1.00	1.00
VR			6,060,690	1,028	0.05	10.29
SE			134,263	1,496	2.46	7.07
SE_VR			4,344,410	470	0.08	22.48
PA			6,059,890	352	0.05	30.07
nc	0.5, 0.5	32	333,200	10,567	1.00	1.00
VR			333,254	1,816	1.00	5.82
SE			134,263	1,496	2.48	7.06
SE_VR			123,269	648	2.70	16.30
PA			67,183	669	4.96	15.79

TABLE 5
Experimental results for d695

Technique	Test time, data factor (α, β)	TAM width (W)	Test time τ_{tot} (1000 clock cycles)	Test-data volume μ_{tot} (kbits)	Comparison of test time τ_{nc}/τ_{tot}	Comparison of test-data volume μ_{nc}/μ_{tot}
nc	1, 0	8	85	667	1.00	1.00
VR			85	667	1.00	1.00
SE			47	349	1.82	1.91
SE_VR			47	345	1.82	1.93
PA			47	345	1.82	1.93
nc	0, 1	8	87	678	1.00	1.00
VR			634	49	0.14	13.98
SE			51	322	1.71	2.11
SE_VR			647	221	0.13	3.07
PA			634	49	0.14	13.98
nc	0.5, 0.5	8	87	678	1.00	1.00
VR			85	100	1.01	6.78
SE			49	323	1.75	2.10
SE_VR			64	258	1.35	2.62
PA			85	100	1.01	6.78
nc	1, 0	16	46	701	1.00	1.00
VR			46	701	1.00	1.00
SE			33	442	1.40	1.58
SE_VR			33	373	1.40	1.88
PA			33	373	1.40	1.88
nc	0, 1	16	51	787	1.00	1.00
VR			634	49	0.08	16.22
SE			49	316	1.04	2.49
SE_VR			647	221	0.08	3.57
PA			634	49	0.08	16.22
nc	0.5, 0.5	16	51	787	1.00	1.00
VR			59	117	0.86	6.73
SE			46	318	1.10	2.48
SE_VR			35	259	1.43	3.04
PA			59	117	0.86	6.73
nc	1, 0	32	26	728	1.00	1.00
VR			26	728	1.00	1.00
SE			16	401	1.60	1.81
SE_VR			16	400	1.60	1.82
PA			16	400	1.60	1.82
nc	0, 1	32	31	872	1.00	1.00
VR			372	49	0.08	17.98
SE			27	316	1.12	2.76
SE_VR			374	221	0.08	3.95
PA			372	49	0.08	17.98
nc	0.5, 0.5	32	27	787	1.00	1.00
VR			42	130	0.65	6.05
SE			23	318	1.21	2.47
SE_VR			20	255	1.35	3.08
PA			42	130	0.65	6.05