Test-Architecture Optimization and Test Scheduling for SOCs with Core-Level Expansion of Compressed Test Patterns

Anders Larsson, Erik Larsson, Krishnendu Chakrabarty*, Petru Eles, and Zebo Peng

Embedded Systems Laboratory
Linköpings Universitet
SE-582 83 Linköping, Sweden

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

Abstract[1]

The ever-increasing test data volume for core-based system-on-chip (SOC) integrated circuits is resulting in high test times and excessive tester memory requirements. To reduce both test time and test data volume, we propose a technique for test-architecture optimization and test scheduling that is based on core-level expansion of compressed test patterns. For each wrapped embedded core and its decompressor, we show that the test time does not decrease monotonically with the width of test access mechanism (TAM) at the decompressor input. We optimize the wrapper and decompressor designs for each core, as well as the TAM architecture and the test schedule at the SOC level. Experimental results for SOCs crafted from several industrial cores demonstrate that the proposed method leads to significant reduction in test data volume and test time, especially when compared to a method that does not rely on core-level decompression of patterns.

1. Introduction

Advances in technology have led to the emergence of "system chips", also referred to as system-on-chip (SOC) integrated circuits (ICs). In order to meet demands for short time-to-market, it is increasingly common to make use of a core-based design approach. Despite the widespread adoption of a core-based SOC design flow, manufacturing test remains a major problem for SOCs, due in part to high test data volume, long test time, and the need for large memory on testers. The 2005 International Technology Roadmap for Semiconductors predicted that test data volume and test time will grow 30x by 2010 [1].

Modular testing offers a low-cost SOC test solution. In modular testing, a test wrapper is used to isolate a core for testing and a test access mechanism (TAM) is used to transport test data from IC pins to a core, and test responses from a core's output terminals to an IC pins. TAMs and wrappers for core access can be combined with test scheduling at SOC level to reduce test time. The recent IEEE 1500 Std. includes a specification of the various modes of operation for a core test wrapper [2, 3]. Since TAM design and test scheduling are not addressed by IEEE 1500, a number of "test planning" techniques have been proposed recently for TAM optimization and test scheduling [4, 5, 6]. The main goal of these methods is to reduce test time for a core-based SOC. Reduced test time leads to a corresponding decrease of the amount of test data that must be stored on the tester and delivered to the SOC. However, these techniques do not address the problem of high test data volume for the embedded cores.

Test-data compression (TDC) has recently emerged as a popular technique to reduce test data volume and test time. TDC has been used in many large ICs containing tens of million of gates, and test compression tools are now common in industry [7, 8, 9]. In TDC, on-chip decompressors are used to expand compressed test patterns before they are loaded into a core's scan chains. The benefits of TDC 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 TDC (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.

Several methods have been published in recent years to combine core-level TDC with SOC-level test planning [10,

1. The research is partially supported by the Strategic Integrated Electronic Systems Research (STRINGENT) program.

978-3-9810801-3-1/DATE08 © 2008 EDAA

11, 12, 13]. As expected, these techniques show that TDC 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. Many TDC methods provide higher compression when a slight increase in test time for a core is allowed through the use of a narrow TAM. In such cases, prior work does not provide any means to trade-off the compression at the core-level with the test time for a core in the overall SOC test schedule. Another drawback of previous methods is that they lead to irregular test-access architectures, which require specialized TAM optimization and test scheduling solutions at the SOC level.

In this paper, we present a co-optimization technique that reduces the SOC test time by determining the best trade-off between TDC and the test time for each core in the SOC. The proposed SOC-level test planning method leads to regular test-access architectures and it is able to leverage the large body of work that has been developed recently for TAM optimization and test scheduling. We use the selective encoding method from [14] as a representative compression method at the core-level. The decompressor for a core is placed between its wrapper and the TAM as illustrated in Figure 1. In this way, the number of inputs to the decompressor is determined on the basis of not only the compression achieved for the test cubes for the core, but also the test data volume (and compression achieved) for the other cores in the SOC. It is the test stimuli that are subject for the test planning while the handling of test responses is beyond the scope of this work.

The rest of this paper is organized as follows. In Section 2, we discuss TDC and describe how the test time varies in a non-monotonic manner with the number of decompressor inputs. In Section 3, we describe test-time minimization using test- architecture optimization and test scheduling with core-level test-pattern expansion. Experimental results are presented in Section 4. Finally, Section 5 concludes the paper.

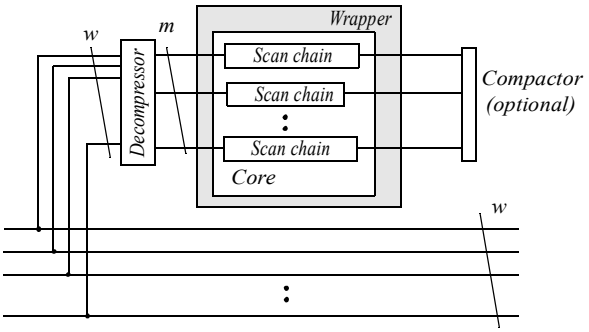

Figure 1. Test data compression for a wrapped core.

2. Analysis of test-data compression

Figure 1 shows a TDC architecture for a single wrapped core. The compressed test data volume for the core, which is stored in the ATE, is sent via w TAM wires to the decompressor at the core under test. The decompressor takes at each clock cycle w input bits and expands them into m bits ($w < m$), which feed m wrapper chains. The scan chains, and the wrapper input and wrapper output cells at the core are accessed using the m wrapper chains [15].

For a number of industrial cores, we analyzed the test time when the selective encoding method from [14] is used for TDC. For every core i, we considered all possible values of w and m and evaluated the test time $\tau_i^c(w, m)$. We found a similar behaviour for all cores but due to space limitations, we only report results on the industrial core named ckt-7.

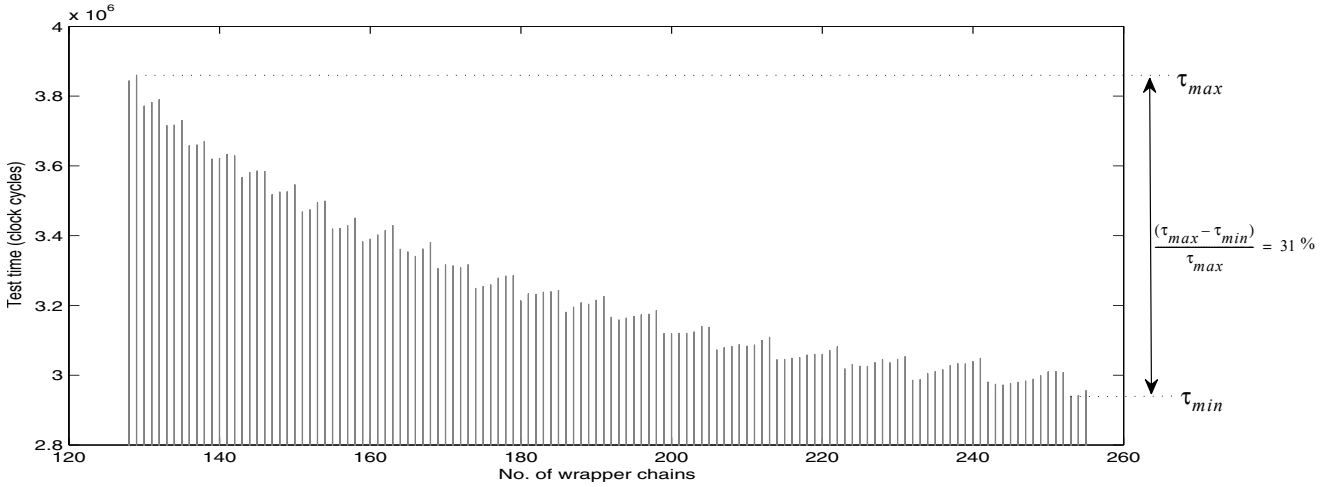

Figure 2. The non-monotonic variation of test time with the number of wrapper-chains at TAM width 10 for core ckt-7.

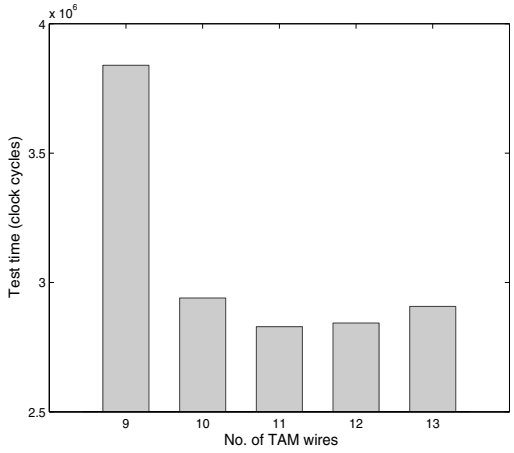

Figure 3. Lowest test time at various TAM widths for core ckt-7.

Figure 2 shows, for ckt-7, the test time when the TAM width is fixed to 10 bits; hence w = 10 and m varies between 128 and 255. It is expected that the test time decreases as the number of wrapper chain increases; for example, the test time at m = 255 is lower than the test time when m = 128. However, it is not expected that the minimum test time τ_{min} is obtained for 253 wrapper chains, and not at the maximum number of wrapper chains (m = 255). Figure 2 shows 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 can increase with an increase in the number of wrapper chains.

For TAM width 10 (w = 10), Figure 2 shows that the minimum test time is at m = 253. For each TAM width w, we have determined the value of m that gives the minimum test time. The minimum test time for each TAM width is plotted in Figure 3. It is interesting to note that the test time does not necessarily decrease as the TAM width (inputs to the decompressor) increases. Actually, the test time can increase as the TAM width increases. Figure 3 clearly shows that the test time at TAM width 11 is lower than at TAM widths 12 and 13.

There are three reasons for the behavior highlighted in Figure 2 and Figure 3. First, the test data itself will be slightly different for different wrapper chain architectures due to the fact that extra bits (so called idle bits) must be added to balance wrapper chains. Second, the reorganization of the test data for a different number of wrapper chains will impact the characteristics (i.e., the distributions of 1s, 0s, and Xs) of the scan slices, and therefore also the amount of compression achieved. Third, the number of decompressor inputs w is a ceiling function for the selective-encoding method [14], and given by $w = \lceil \log_2(m+1) \rceil + 2$. As a result, the test time varies in a non-monotonic fashion with m, as shown in Figure 2.

The above analysis has shown that it is not straightforward to optimally design decompressor at core-level to minimize the test time. Furthermore, at the SOC-level, the values of w and m for each core should be determined such that the overall test data volume and corresponding test time are minimum; hence all other cores must be considered together. This problem is also not trivial.

3. Test data compression-driven TAM optimization and test scheduling

The test-architecture design and test scheduling problem for a core-based SOC involves the following goals: (i) partition the top-level (SOC) test-access wires into TAMs, (ii) define the width w_i of each TAM, (iii) assign cores to TAMs, and (iv) design a wrapper for each core, such that the test time is minimum. If TDC is taken into account, we also need to determine where to place the decompressors as well as the widths for their inputs and output interfaces, i.e., the value of w at the decompressor input and the value m at the decompressor output.

Figure 4 shows three alternatives for one industrial design. In Figure 4(a), the test architecture and the test schedule are optimized but TDC is not used. The test time is 32460913 clock cycles. Figure 4(b) shows an alternative scheme where test architecture design and test scheduling are optimized assuming a decompressor per TAM. The test time is lowered to 10711883 clock cycles. However, the TAMs used to access the cores are extremely wide. Figure 4 (c) shows a scheme where a decompressor is placed at each core. The test time is the same as in Figure 4(b); however, the on-chip TAMs are much narrower.

The problem presented in this work is similar to the test-architecture optimization and test scheduling problems, which are known to be NP-hard [16]. Therefore, we use a heuristic method in this paper. We are given a set of test cubes for each core and the SOC-level TAM width. The heuristic procedure consists of four steps:

1. *Wrapper-chain design*. For a given core and its test length, wrapper-design places the scanned elements (scan chains, input and output wrapper cells) into wrapper-chains with the corresponding test time. We have made use of the optimization heuristic from [5]. For each core, we generate a number of the wrapper design alternatives.
2. *Decompression design*. We make use of the selective encoding TDC scheme [14]. As discussed above, we generate all alternatives for the decompressor input/ output mapping. For each combination of w and m, we have for a core i the test time $\tau_i^c(w,m)$. The TDC technique in [14] encodes the test data for wrapper chain slices in every clock cycle; hence test data volume and test time are reduced. TDC is carried out by encoding the wrapper chain slices of the test data using

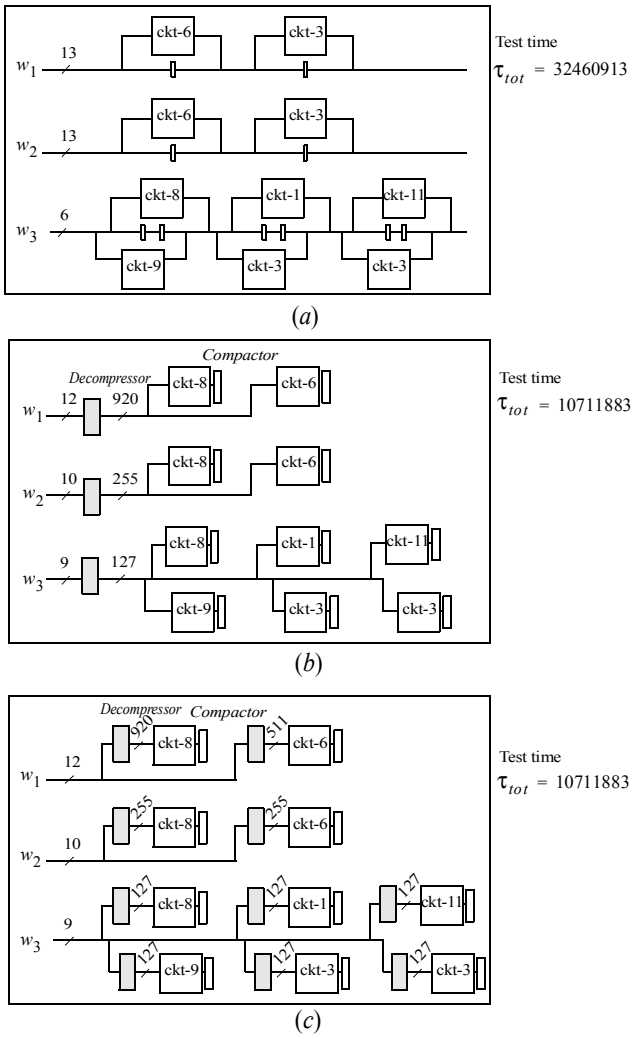

Figure 4. Test architecture design and test time (a) without using test data compression, (b) with TDC and one decompressor per TAM and (c) with TDC and one decompressor per core.

a number of w-bit slice codes, where $w = \lceil \log_2(m+1) \rceil + 2$. The coding is performed using either single-bit-mode or group-copy-mode. In single-bit-mode, each bit in a slice 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" since it is positioned at index 3. In the group-copy-mode the m-bit slice is divided into $m/\lceil \log_2(m+1) \rceil$ groups. Two code words are needed to encode one group. The first code word specifies the index of the first bit in the group, and the second code word contains the test data. The hardware cost for the selective encoding TDC technique is small. The synthesized controller part of the decompressor contains only 5 flip-flops and 23 combinational gates. The other parts of the decompressor are synthesized separately since they depend on w and m. For m = 1024 and w = 13, the synthesized decompressor contains 6409 gates and 1035 flip-flops. For larger than million-gate designs, this corresponds to a hardware cost of only 1% [14].

3. *Test architecture design*. The input for this step is a TAM width (W_{TAM}) and the output is a TAM design. Figure 4(b) shows an example where the given TAM width W_{TAM} = 31 has been partitioned into 3 TAMs (k=3) of width w_1=12, w_2=10, w_3=9.
4. *Test scheduling*. Given a test-access architecture, we sort the cores based on test time such that the core with longest test time is first, and then we traverse the set of cores and assign a core to a TAM such that resulting increase in the test time is the least. For each core, we have a lookup table from step1 and 2 to find its test time at various TAM widths, and we try each TAM width; hence the computational complexity for k TAMs and n cores is $O(nk)$.

4. Experimental results

In this section, we demonstrate the importance of integrating test architecture design, test scheduling, and TDC. We have implemented the technique described above and we have carried out experiments on the benchmark designs d695 [17] and d2758 [11], and four industrial designs.

The benchmark SOCs d695 and d2758 are composed of ISCAS'89 cores. These cores are fairly small. The number of scan-chains are in all cases less than 32, the number of test patterns are in the range 12 to 234, and the density of care bits is on average 66%. For the industrial designs, the number of scan cells ranges from 10,000 to 110,000, the test data volume is of the order of tens of Gigabits, and the care-bit density is no more than 5%.

In order to make a fair comparison with prior work [11, 13, 18], we conduct two experiments. First, we use the number of ATE channels W_{ATE} as a constraint while minimizing the test time (Table 1) and, second, we use the TAM width as a constraint and we minimize the test time (Table 2). The test time results of our approach against [18] and [11] at given ATE channels (W_{ATE}) are in Table 1. Column 1 lists the design and Column 2 lists the number of ATE channels. Column 3 and Column 4 list the test time reported in [18] and [11], respectively. Column 5 shows the test time obtained using our approach. The last two columns compare the test time obtained in this work with that reported in prior work.

The results on test time at a given TAM widths (W_{TAM}) for our approach, [13] and [18] are in Table 2. The columns of Table 2 are essentially the same as the columns in Table 2 except that Column 2 in Table 2 lists the TAM width constraint. For the results on our approach against [18], it should be noted that [18] makes use of few ATE channels and many TAM wires as the decompressor is at SOC-level. It means that when comparing test time at ATE channel

Table 1. Test time comparison for ATE constraint

Design	ATE channels (W_{ATE})	[18]	[11]	Proposed approach	Comparison	Comparison
		Time ($\tau_{[18]}$) (clock cycles)	Time ($\tau_{[11]}$) (clock cycles)	Time (τ_c) (clock cycles)	$\tau_c/\tau_{[18]}$	$\tau_c/\tau_{[11]}$
d695	16	24701	n.a.[i]	29301	1.19	n.a.[i]
	24	18564	n.a.[i]	18701	1.01	n.a.[i]
	32	12192	n.a.[i]	14252	1.17	n.a.[i]
	40	10432	n.a.[i]	12269	1.18	n.a.[i]
	48	9869	n.a.[i]	12269	1.24	n.a.[i]
d2758	4	n.a.[i]	192088	499586	n.a.[i]	1.81

i. Not available

Table 2. Test time comparison for TAM width constraint

Design	TAM width (W_{TAM})	[18]	[13]	Proposed approach	Comparison	Comparison
		Test time $\tau_{[18]}$ (clock cycles)	Test time $\tau_{[13]}$ (clock cycles)	Test time τ_c (clock cycles)	$\tau_c/\tau_{[18]}$	$\tau_c/\tau_{[13]}$
d695	28	24701	n.a.[i]	16139	0.65	n.a.[i]
	42	18564	n.a.[i]	12269	0.66	n.a.[i]
	56	12192	n.a.[i]	11714	0.96	n.a.[i]
	70	10432	n.a.[i]	10437	1.00	n.a.[i]
	84	9869	n.a.[i]	9870	1.00	n.a.[i]
	98	9869	n.a.[i]	9870	1.00	n.a.[i]
	112	9869	n.a.[i]	9870	1.00	n.a.[i]
	161	n.a.[i]	11049	9870	n.a.[i]	0.89
	186	n.a.[i]	9870	9870	n.a.[i]	1.02

i. Not available

constraint (Table 1), we perform not as well as when the comparison is made at TAM wire constraint (Table 2). As discussed above, a decompressor at SOC-level leads to extensive and costly TAMs. For the results in Table 2, we perform better than [18] and in the same range as [13]. Our results are inferior to that reported in [11]; however, [11] only reports results for a single value of w, namely $w = 4$.

The designs d695 and d2758 are very small designs, the test data is small and the designs have a care-bit density of 44% on average [19]. This number is very high compared to the care-bit density of 1%-5% reported for modern industrial cores [20]. Therefore, we provide extensive experimental results using designs with industrial cores.

For the following experiment, we made use of d695 and four designs, System1, System2, System3, and System4, which are composed of industrial cores described in detail in [14]. We minimize the test time and the test data volume for a given TAM width constraint and we compare test-architecture design and test scheduling with and without test data compression.

Table 3 shows at various TAM width constraints, the test time, test data volume and CPU time (execution time to produce the solution, excluding the time for TDC) for the two cases. We compare the minimized test time τ_c and test data volume V_c obtained when TDC is used, against the minimized test time τ_{nc} and test data volume V_{nc} obtained without making use of TDC. The results in Table 3 are organized as follows. Column 1 lists the designs, Column 2 lists the number of gates and latches, and Column 3 lists the initial test data volume. Column 4 lists the TAM width constraint. The following six columns lists the test time, test data volume, and CPU time for the case without TDC and with TDC, respectively. The last three columns highlight the comparisons. Column 11 lists the comparison between the test times obtained using TDC and the test time obtained when TDC is not used. Column 12 lists the comparison between the test data volume obtained using TDC to the initial test data volume. Finally, Column 13 lists the comparison between the test data volume obtained using TDC and the test data volume obtained when TDC is not used.

The results show the importance of co-optimizing TDC, test architecture design, and test scheduling for SOCs. On average, our approach with TDC results in a 12.59x (15.39x) reduction in test time. (The results in parenthesis are for example SOCs that are crafted from industrial cores only.) The corresponding reduction in test-data volume is on average 12.78x (15.80x). The results also indicate that our approach with TDC is computationally effective. The CPU time was less than one minute, even for the system with the largest number of cores and a wide TAM width.

5. Conclusions

Manufacturing test is a major challenge for system-on-chip (SOC) partially due to high test data volumes, long test times and the need for large memory on testers. To reduce both test time and test data volume, we propose a co-optimization technique for test-architecture design, test scheduling and test data compression based on core-level expansion of compressed test patterns. We analyzed for a set of industrial cores the inputs and outputs of the decompressor in relation to test time and we found a non-monotonically decreasing behavior. We therefore propose a technique where we explore the trade-off between the test time and test data compression at core-level for each core and at SOC-level simultaneously. The proposed approach leads to regular test-access architectures and is able to leverage the large body of work that has been developed recently for test-architecture optimization and test scheduling. We have implemented the technique and compared it against previous work. We have also compared the approach against test architecture design and test scheduling using SOCs crafted from industrial cores and the results show that we can get a test time reduction at average 15x and a test data volume reduction at 16x.

References

[1] International Technology Roadmap for Semiconductors, *http://public.itrs.net,* 2005.

[2] F. DaSilva, Y. Zorian, L. Whetsel, K. Arabi, and R. Kapur, "Overview of the IEEE 1500 Standard," *Proceedings of International Test Conference*, pp. 988–997, 2003.

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

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

[5] V. Iyengar, K. Chakrabarty, and E. J. Marinissen, "Test

Wrapper and Test Access Mechanism Co-Optimization for System-on-Chip," *Proceedings of International Test Conference*, pp. 1023–1032, 2001.

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

[7] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee, "Embedded deterministic test," *IEEE Transactions on Computer Aided Design*, Vol. 23, pp. 776–792, 2004.

[8] B. Koenemann, C. Barnhart, B. Keller, T. Snethen, O. Farnsworth, and D. Wheater, "A SmartBIST variant with guaranteed encoding," *Proceedings of Asian Test Symposium*, pp. 325–330, 2001.

[9] M. Chandramouli, "How to implement deterministic logic built-in selftest (BIST)," *Compiler: A Monthly magazine for technologies world-wide*, Synopsys, January 2003.

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

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

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

[13] Z. Wang, K. Chakrabarty, and S. Wang, "SoC testing using LFSR reseeding, and scan-slice-based TAM optimization and test scheduling," *Proceedings of Design, Automation and Test in Europe*, pp. 201–206, 2007.

[14] Z. Wang and K. Chakrabarty, "Test Data Compression for IP Embedded Cores Using Selective Encoding of Scan Slices," *Proceedings of International Test Conference*, pp. 581–590, 2005.

[15] V. Iyengar, K. Chakrabarty, and E. J. Marinissen, "Test Wrapper and Test Access Mechanism Co-Optimization for System-on-Chip," *Journal of Electronic Testing: Theory and Applications (JETTA)*, Vol.18, No. 2, pp. 213–230, 2002.

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

[17] E. J. Marinissen, Y. Iyengar, K. Chakrabarty, "A Set of Benchmarks for Modular Testing of SOCs," *Proceedings of International Test Conference*, pp. 519–528, 2002.

[18] A. Sehgal, V. Iyengar, and K. Chakrabarty, "SOC test planning using virtual test access architectures," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 12, Issue 12, pp. 1263–1276, 2004.

[19] S. Kajihara and K. Miyase, "On Identifying Don't Care Inputs of Test Patterns for Combinational Circuits," *International Conference on Computer Aided Design*, pp. 364–369, 2001.

[20] T. Hiraide, K. O. Boateng, H. Konishi, K. Itaya, M. Emori, H. Yamanaka, and T. Mochiyama, "BIST-aided scan test - a new method for test cost reduction," *Proceedings of VLSI Test Symposium*, pp. 359–364, 2003.

Table 3. Test time minimization at TAM width constraint

Design	No. of gates, no. of latches (1M, 100k)	Initial given test data volume V_i (Mbits)	TAM width (W_{TAM})	Without test data compression			With test data compression			Time reduction factor τ_{nc}/τ_c	Data volume reduction factor V_i/V_c	Data volume reduction factor V_{nc}/V_c
				Test time τ_{nc} (1000 clock cycles)	Test data volume V_{nc} (Mbits)	CPU time (s)	Test time τ_c (1000 clock cycles)	Test data volume V_c (Mbits)	CPU time (s)			
d695	n.r.[i], 0.06	0.340	16	44.745	0.696	3.91	29.301	0.475	0.05	1.53	0.72	1.47
			24	30.107	0.699	6.25	18.701	0.425	0.20	1.61	0.80	1.65
			32	22.538	0.699	65.11	14.252	0.425	0.64	1.58	0.80	1.65
			40	18.058	0.705	158.79	12.269	0.482	1.72	1.47	0.71	1.46
			48	15.474	0.719	336.09	12.269	0.454	0.13	1.26	0.75	1.58
			56	13.163	0.708	1442.85	11.714	0.580	0.04	1.12	0.59	1.22
			64	12.034	0.723	16013.3	10.437	0.589	0.09	1.15	0.58	1.23
System1	7.13, 5.39	6547	16	409351	6548	0.29	28323	432	0.12	14.45	15.16	15.16
			24	272942	6548	0.42	15160	345	0.46	18.00	18.96	18.96
			32	204745	6548	3.68	10697	338	0.31	19.14	19.35	19.36
			40	163923	6549	5.42	9857	368	4.16	16.63	17.81	17.82
			48	136532	6549	0.81	7580	345	0.01	18.01	18.96	18.97
			56	117013	6548	0.94	7580	348	0.04	15.44	18.81	18.81
			64	102431	6550	13.27	7580	345	0.14	13.51	18.96	18.97
System2	16.74, 10.74	7967	16	498618	7973	1.27	42684	667	0.50	11.68	11.95	11.96
			24	332471	7973	27.81	22190	529	0.40	14.98	15.05	15.07
			32	249309	7973	148.75	15547	493	1.33	16.04	16.16	16.17
			40	199511	7972	315.96	13033	517	3.98	15.31	15.41	15.42
			48	166306	7974	1495.87	11150	529	1.26	14.92	15.05	15.07
			56	142603	7977	265.40	9086	502	3.83	15.70	15.86	15.88
			64	124789	7978	57.96	7855	496	10.28	15.89	16.07	16.09
System3	40.24, 26.46	20731	16	1296889	20740	2.52	112766	1777	1.03	11.50	11.67	11.67
			24	864201	20734	198.99	58069	1391	0.84	14.88	14.91	14.91
			32	648403	20739	298.26	41621	1325	2.77	15.58	15.65	15.66
			40	518662	20736	4193.85	34056	1357	8.28	15.23	15.28	15.28
			48	432238	20736	3041.56	29067	1391	2.56	14.87	14.91	14.91
			56	370625	20739	5805.18	23724	1321	7.67	15.62	15.70	15.71
			64	324320	20740	10260.10	20737	1320	20.72	15.64	15.70	15.71
System4	48.58, 33.64	24853	16	1554672	24861	32.35	133277	2112	1.56	11.66	11.77	11.77
			24	1036684	24864	183.26	68572	1641	1.25	15.12	15.14	15.15
			32	777643	24871	465.53	49168	1564	4.15	15.82	15.89	15.90
			40	622073	24869	6443.33	40704	1619	12.46	15.28	15.35	15.36
			48	518413	24864	9594.34	34365	1641	3.86	15.09	15.14	15.15
			56	444525	24874	3215.82	28102	1564	11.43	15.82	15.89	15.91
			64	388931	24873	16090.20	24687	1567	30.93	15.75	15.86	15.87
									Average for all designs	12.59	12.78	12.78
									Average for industrial designs only	15.39	15.80	15.80

i. Not reported