Test Data Compression: The System Integrator's Perspective

Paul Theo Gonciari and Bashir M Al-Hashimi
Dept. of ECS, University of Southampton
United Kingdom
{p.gonciari,bmah}@ecs.soton.ac.uk

Nicola Nicolici*
Dept. of ECE, McMaster University
Canada
nicola@ece.mcmaster.ca

Abstract

Test data compression (TDC) is a promising low-cost methodology for System-on-a-Chip (SOC) test. This is due to the fact that it can reduce not only the volume of test data but also the bandwidth requirements. In this paper we provide a quantitative analysis of two distinctive TDC methods from the system integrator's standpoint considering a core based SOC environment. The proposed analysis addresses four parameters: compression ratio, test application time, area overhead and power dissipation. Based on our analysis, some future research directions are given which can lead to an easier integration of TDC in the SOC design flow and to further improve the four parameters.

1. Introduction

The continuous increase in chip complexity and transistor density lead to an increased burden on manufacturing test, which is a decisive factor in chip delivery. With the increase in the cost of manufacturing test [13], new test methodologies which address the test problem from a low-cost point of view are required. In addition, in the system-on-a-chip (SOC) paradigm, the system integrator has only knowledge about the core functionality but not about its implementation details. Thus, changing the core test methodology leads to core redesign, which can defeat the very purpose of the SOC design: short time to market through core reuse. The cost of test is related to the volume of test data (VTD) [19], the test application time (TAT) [3], the channel capacity [15] and nevertheless the cost of automatic test equipment (ATE) [13]. In addition, with the large number of gates and high chip frequency, high power dissipation during test can considerably affect yield [23, 26], and thus cost. To avoid core redesign, an effective low-cost methodology should not only address the above factors, but it should also assume basic test knowledge about the cores. It is considered in this paper that the system integrator has only the mandatory test information specified by the IEEE P1500 standard [17] (i.e., the system integrator is able to perform core wrapper design and has the test set delivered with the core), and is restrained from performing fault simulation due to IP protection of hard and firm cores.

*Research supported in part by Micronet (C.6.MM2) and NSERC (RGP239-003-01)

An effective low-cost methodology in scan based designs is test data compression (TDC). Testing in TDC environments (TDCE) implies sending the compressed test data from the ATE to the on-chip decoder, decompressing the test data on-chip and sending the decompressed test data to the core under test (CUT). TDC methods can be broadly classified into two classes: (i) methods which exploit the spareness of care bits in the test set [2, 7, 16] and (ii) methods which exploit the regularities of the test set [6, 8, 9, 14, 20]. The former are embedded with automatic test pattern generation (ATPG), hence requiring an ATPG run by the system integrator. This is done to ensure the small percentage of care bits required by these methods and the required fault coverage. The latter can be employed from the system integrator's perspective without any penalties on fault coverage, however, their TDCE parameters can be affected. The TDCE has been characterized in [8] using three parameters: compression ratio, area overhead and TAT. While power dissipation during scan has been addressed in conjunction with TDC [5], the approach considered power dissipation as a parameter of the test set. Therefore, it is a test set dependent approach and can efficiently reduce power dissipation only during scan-in. If, however, scan-out power is also considered, then scan latch reordering (SLR) has to be performed [21] in order to contain the total power dissipation. In the considered core based SOC environment SLR is an intrusive technique which leads to core redesign. Thus, a solution which guarantees total power reduction during scan from the system integrator's perspective has to be non-intrusive and test set independent. This is achieved by either using a slow clock or exploiting scan chain partitioning [9, 18, 25]. To characterise the TDCE from the power dissipation point of view, in this paper we consider a fourth parameter: the power dissipation as a feature of the on-chip decoding architecture.

The aim of this paper is to provide a quantitative analysis of the two TDC categories when applied to core based SOC environments. For this purpose we need to chose two representative compression methods. Analysing the first cate-

Core	n/m	s	FFs	min_{sc}/max_{sc}	$\%cb$
s5378	35/49	4	179	44/45	20.58 – 24.64
s9234	36/39	4	211	52/53	16.28 – 28.72
s13207	62/152	16	638	39/40	6.01 – 8.88
s15850	77/150	16	534	33/34	6.77 – 15.11
s35932	35/320	32	1728	54/54	2.80 – 59.36
s38417	28/106	32	1636	51/52	2.63 – 21.73
s38584	38/304	32	1426	44/45	3.23 – 17.19

Table 1. Core specification

gory, it is found that the RESPIN architecture imposes design constraints [7] which may not suite the system integrator. The approaches in [2] and [16] are embedded in an ATPG process. Since it appears that the former is constrained to generating a limited number of output combinations, the XOR-network based architecture [16] was chosen for the performed analysis. This approach is tuned for core based SOC environments in the following section. With respect to the second category, the approach in [20] achieves considerably reduction in TAT, however the VTD is negatively affected by the required control signals. It was show in [9] that the extended distribution architecture (add-on-xDistr) has promising features when compared to [6, 8, 14], and therefore chosen for the analysis performed in this paper. To evaluate the selected architectures we considered three variables: the care bit density of the test set (i.e., the percentage of care bits), the test bus width, and the frequency ratio between the on-chip test frequency and the ATE operating frequency. The impact on these four TDCE parameters is analysed when the three variables are varied. In addition, based on our analysis some further research directions are given. It is assumed throughout the paper that the system interator has to chose the right chip level TDC infrastructure in order to ensure core level test having only basic core test information.

2. Preliminaries

To assess the two compression architectures in a core based SOC environment we choose as starting point three variables: the care bit density of the test set ($\%cb$), the test bus width (w), and the frequency ratio (α). All the TDC methods are at some extent dependent on the number of care bits in the test sets, hence, varying the care bit density will influence the amount of compression attainable by the two architectures. An important factor in reducing the TAT in core based SOCs is the width of the test bus [17]. As it will be seen in this paper this influences the TAT and the power dissipation when considered with the two compression architectures. Finally, the frequency ratio is, on one hand, a requirement for the add-on-xDistr architecture [9], and on the other hand, it has also an impact on the power dissipation for this architecture. In order to vary the above variables we considered the following experimental setup. We firstly performed controlled static compaction, then core wrapper design, and finally, the two approaches have been applied and analysed.

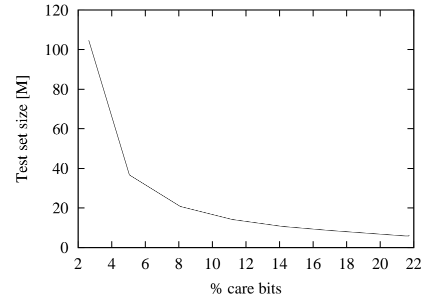


Figure 1. Test set size decrease with increase of $\%cb$

The above steps were performed on the largest ISCAS89 [4] benchmark circuits. The specifications of the circuits are given in Table 1. For each circuit, the number of inputs/outputs (n/m), the number of scan chains (s), the total number of FFs (FFs), the minimum and maximum scan chain lengths (after the assignment of FFs to scan chains), and the range of considered care bit densities per test set ($\%cb$) are listed. To model the different care bit densities, we firstly ran Atalanta [1] and generated the test sets comprising one test vector for each fault. Subsequently, we performed controlled static compaction using a simple greedy heuristic. The heuristic compacted the test set such that if the test vectors had a number of care bits smaller than a control value B, then they were merged with compatible test vectors if the resulting number of care bits did not exceed B. Controlled compaction was performed for values of $B = \{50, 100, 150, 200, 250, 300, 400, 800, 1000, 2000\}$. The variation of the initial test set size with $\%cb$ is given in Figure 1 for core s38417. The system integrator could employ such a controlled compaction scheme, since, as it will be seen in the following section, substantial reduction in VTD can be attained when combined with the two architectures. Having test sets with different care bit densities, the next step has been core wrapper design, i.e., the construction of wrapper scan chains (WSCs). The chosen core wrapper design algorithm was *mUMA* [11]. While the *mUMA* is required by the add-on-xDistr architecture, there are no penalties with respect to TAT, for the ISCAS89 benchmark circuits, when compared to other algorithms [11]. From this point, the experimental setup is dependent on the used architecture. If the XOR-network architecture is employed, then the test set is mapped to the WSCs. As it will be seen later in this section, the add-on-xDistr architecture requires two WSCs partitions, hence the test set is divided into two partitions corresponding to the WSC' ones. Core wrapper design was performed for test bus widths $w = \{8, 16, 24, 32\}$. The two approaches were implemented as described next.

XOR-network based architecture [16] The XOR-network architecture based on the method proposed in [16] is illustrated in Figure 2 for a core with 4 WSCs. The main idea behind the method is to use one data signal to stream data into *SR*, which will then justify through the XOR-network the care bits into the WSCs.

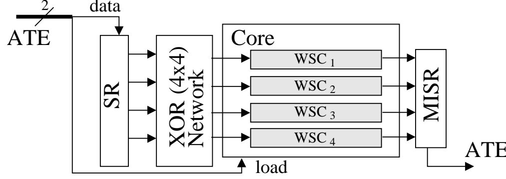


Figure 2. XOR-network based architecture [16]

However, the architecture can run into temporal pattern lockout (i.e., the inability to justify the WSC care bits at a given moment), hence one control signal to halt the WSC load temporarily is needed. Therefore, the architecture requires two ATE channels. In addition, care must be taken to avoid structural pattern lockout (i.e., when there are no *SR* values such that the XOR-network outputs will justify the correct care bits into the WSCs), since this can affect fault coverage. To account for this problem and to tune the method for the chosen core based SOC environment, we considered the XOR-network to be associated to a square matrix $A(w,w)$ with $det(A) \neq 0$. In addition in the implemented approach we used a shift register (*SR* in the figure) also of size w. The two combined, guarantee that any pattern can be justified to the inputs of the WSCs regardless of the care bit density. Hence, the implemented architecture can also be used from the system integrator's perspective at the system integration level in IP based SOCs. It is important to note that this paper does not replicate the method as proposed in [16], but rather it adapts its main idea and tunes it for core based SOC test. Next we detail the test data compression process implemented for the XOR-network based architecture.

Consider W_1 and W_2 to be the column vectors corresponding to two consecutive WSC loads. Finding the values of SR_1 and SR_2 such that W_1 and W_2 can be justified through the XOR-network basically means finding solutions to the equations: $SR_1 \times A = W_1$ and $SR_2 \times A = W_2$. In addition, in order to minimise the number of clock cycles when the WSC load is halted, the number of shifts between SR_1 and SR_2 must be minimised as well. This is possible because not all the *SR* bits are necessary to justify the care bits in one WSC load. To account for the above, we implemented a simple two step greedy heuristic. Firstly, we solved the equations using Gauss-Jordan elimination. Secondly, we determined the smallest number of shifts from SR_1 to SR_2 such that the care bits in W_2 are justified through the XOR-network. Using Gauss-Jordan elimination will give an upper bound on number of shifts required to justify W_2 starting from any SR_1, hence reducing the computational time for this step. This is explained next. After the reduced matrix corresponding to the second equation is obtained, the maximum bit position in SR_2 used in the reduced Gauss-Jordan matrix represents the maximum number of shifts required to justify W_2 through the XOR-network. For example, if for the 4 bit *SR* considered in Figure 2 the reduced matrix requires only positions 1 and 2 to justify W_2 through the XOR-network, then the maximum number of shifts from any SR_1 to SR_2 is 2. The actual number of shifts was determined incrementally starting from SR_1. As emphasised above, the attached matrix should have $det(A) \neq 0$. To fulfil this condition, we choose a prim polynomial of degree w, and constructed the attached matrix (M). In addition, we raised the matrix to the power of w. The obtained matrix is now the XOR-network attached matrix. This second step was performed to reduce the compressed test set size. While we experimented with different powers of M, we found that choosing w leads on average to $w/2$ inputs controlling one output. This allows for increased control of the XOR-network outputs hence reducing the VTD. For example, for a test set of 16.35% care bits for core s38584 when $w = 16$, using attached matrices M^8 and M^{16} yielded compressed test sets of sizes 239076 bits and 143106 bits respectively. Hence, considering M^{16} as the XOR-network matrix leads to clearly smaller test sets.

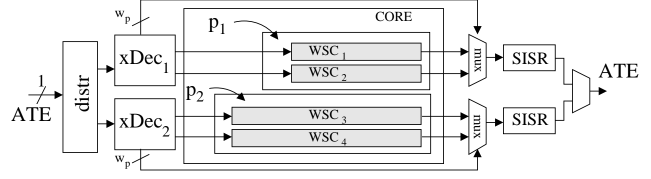
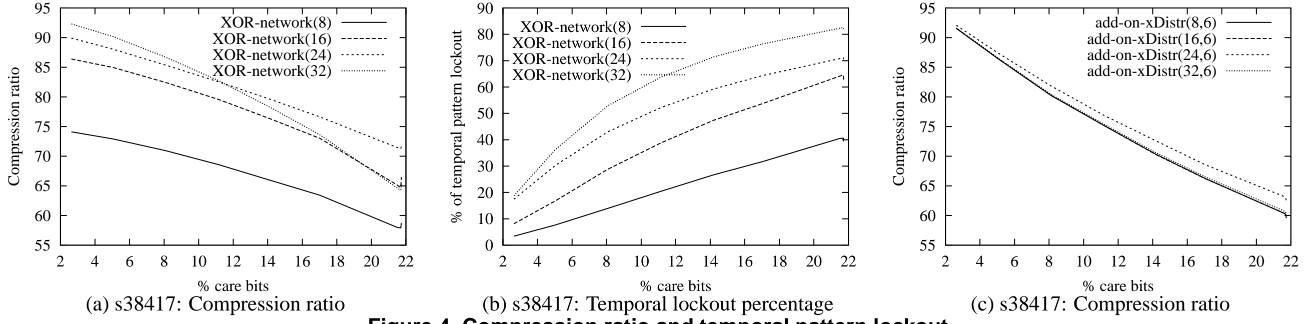


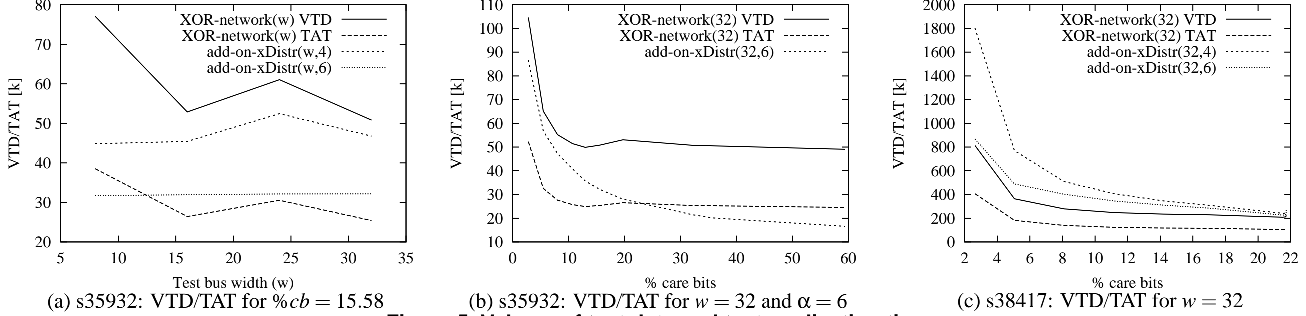
Figure 3. add-on-xDistr architecture [9]

add-on-xDistr architecture [9] In the following we briefly describe the add-on-xDistr architecture, the detailed description can be found in [9]. The architecture is illustrated in Figure 3. The architecture comprises a distribution unit (*distr*) and two extended decoders (*xDec*). The WSCs were partitioned into two partitions p_1 and p_2. The *xDecs* are variable length input Huffman coding (VIHC) decoders [8] augmented with counters to account for the length of the maximum WSC in the partition (*length*), the number of WSCs in the partition (w_p) and the difference in size between the two partitions ($diff$). The add-on-xDistr receives data from the ATE at the ATE clock frequency and generates data to the WSCs at the on-chip test frequency. The main idea behind the architecture is to have each *xDec* feeding exactly one WSC at any time. To keep the figure simple, the WSC select logic is not displayed, however, there are maximum two active WSCs. This is explained next. Upon reset, the p_1 is active and the $distr$ unit will direct the data form the ATE to $xDec_1$. When the $xDec_1$ is busy generating data into one WSC in p_1, the $distr$ will activate $xDec_2$, which will then begin generating data for the active WSC in p_2. This process is repeated until both partitions are full. At this point the test vector is applied and the process repeated. Having only two active WSCs at any time, will influence the power dissipation as detailed in Section 3. Compressing the test set for add-on-xDistr implies compressing the test set for each of the partitions, then tailoring the test sets for the corresponding frequency ratio [10], and generating the composite test set [9]. Tailoring the test sets for the cor-



(a) s38417: Compression ratio (b) s38417: Temporal lockout percentage (c) s38417: Compression ratio

Figure 4. Compression ratio and temporal pattern lockout



(a) s35932: VTD/TAT for $\%cb = 15.58$ (b) s35932: VTD/TAT for $w = 32$ and $\alpha = 6$ (c) s38417: VTD/TAT for $w = 32$

Figure 5. Volume of test data and test application time

responding frequency ratio reduces any extra ATE control channels due to the elimination of synchronisation overhead [10], transforming add-on-xDistr into a self synchronous architecture. The add-on-xDistr was employed for $\alpha = \{4, 6\}$.

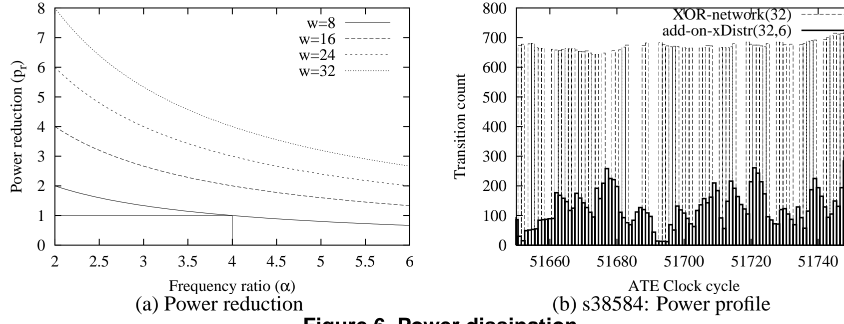
The experimental setup illustrated at the beginning of the section was implemented for each of the cores from Table 1. The quantitative analysis based on experimental data is given in the following section.

3. Quantitative analysis

Varying the care bit density ($\%cb$), the test bus width (w) and the frequency ratio (α), in the following we analyse the influences on the four TDCE parameters (compression ratio, test application time, area overhead and power dissipation). Due to limited space we will use only selected data to outline the various advantages and disadvantages of the two architectures. As noted in the previous section, the XOR-network requires two ATE channels, hence the number of bits representing volume of test data (VTD) will be double the number of clock cycles representing TAT. The add-on-xDistr requires only one ATE channel, and thus, the VTD equals the TAT, which is given in ATE clock cycles. In the figures shown in this section we denoted "XOR-network(w)" the XOR-network when used for a test bus width of w; and "add-on-xDistr(w,α)", the add-on-xDistr when used for a test bus width of w and a frequency ratio of α.

Compression ratio The variation in compression ratio with the increase in $\%cb$ is analysed in Figure 4. For core s38417, Figure 4(a) shows the influence of $\%cb$ on the compression ratio when different values of w are considered. Since the compression ratio for the XOR-network is directly related to the percentage of temporal pattern lockouts ($\%tpl$), Figure 4(b) shows their variation when $\%cb$ increases. It is interesting to note that there is an improvement in compression ratio with the use of a greater w, however, the $\%tpl$ increases as well. This is due to a greater number of care bits which have to be justified through the XOR-network when w increases. At the same time, however, the number of WSC loads decreases for greater w's – e.g., for core s38584 with $\%cb = 17.19$, there are 53599, 26923, 21983, 13585 WSC loads for $w = 8$, 16, 24, and 32 respectively. Hence, as long as the decrease in the number of WSC loads is greater than the increase in $\%tpl$, there is an improvement in compression ratio when increasing w as illustrated in Figure 4(a). Figure 4(c), shows the influence of w and $\%cb$ for $\alpha = 6$ on the compression ratio attainable with add-on-xDistr. What is interesting to note here is that, even though the compression ratio decreases when $\%cb$ increases, it is approximately constant for all the values of w for a given α. These are due to the runs of 0s based coding scheme used with the add-on-xDistr architecture. Also note that, while not shown in the figure, for smaller α's the compression ratio tends to decrease. In the following we analyse the influence of the three variables on the TAT.

Test application time The TAT for the two architectures is analysed in Figure 5. Since in the case of the XOR-network, the VTD is twice the TAT, the VTD is also plotted in the figure. Figure 5(a) shows how VTD and TAT vary with w for core s35932. As also noted previously, the compression attainable by add-on-xDistr is almost constant for different w's for the same α. On the other hand, the



(a) Power reduction (b) s38584: Power profile

Figure 6. Power dissipation

XOR-network has different VTDs and TATs with the variation of w. In this particular case the TAT obtained with XOR-network is smaller than the one for add-on-xDistr when $w \geq 16$, however the VTD is greater. This varies pending the circuit, α and $\%cb$. Two cases are illustrated with Figures 5(b) and 5(c). For example, for core s35932 in Figure 5(b), the TAT and VTD obtained by add-on-xDistr for $w = 32$ and $\alpha = 6$ are overall smaller than both the TAT and the VTD obtained by XOR-network for the same w. While for core s38417 in Figure 5(c) for $w = 32$, the VTD and TAT obtained with add-on-xDistr is greater than both the VTD and TAT obtained with XOR-network. Even though, the two architectures have different behaviours when w varies, if $\alpha = 6$ and $w = 32$ the TATs are similar. It is also interesting to note that the TAT and VTD tend to have a steep fall for $\%cb$ in the first half of the considered $\%cb$ interval. However, for the second half the curve is almost linear. This is contrary to the compression ratio behaviour, which tends to decrease in a linear manner (see Figures 4(a) and 4(c)). This is due to the steep decrease in the compacted test set size as illustrated in Figure 1. Therefore, the system integrator could consider using controlled compaction to reduce TAT and VTD regardless of the employed compression architecture. The analysis with respect to area overhead is given next.

Area overhead To give an estimation of the area overhead variation with w we used Synopsis Design Compiler [22]. The results in technology units for lsi_10k are tabulated below.

$w =$	8	16	24	32
XOR-network	370	698	1023	1570
add-on-xDistr	1886	1813	1813	1742

For the XOR-network based architectures, the area overhead is linearly dependent on the size of the test bus. This is because, in order to ensure good TAT and VTD, the XOR-network was chosen with an average of $w/2$ inputs driving one XOR-network output. For add-on-xDistr, however, the changes are negligible. This is because by increasing w, the size of the WSCs get smaller. Hence, the size of the *length* counter decreases, while the size of the w_p counter increases (see Section 2).

Power dissipation The last parameter is analysed in the following. Firstly it is interesting to note that due to the usage of maximum two WSCs at any time, the add-on-xDistr architecture can considerably reduce power dissipation, especially when the number of WSCs is large. This, despite the fact that the WSCs are driven by the on-chip test frequency. Assuming the same switching probability and all the WSCs of the same length, it can be easily derived that the power reduction factor when compared to XOR-network is $p_r = w/(2 * \alpha)$. This is illustrated in Figure 6(a) for different w's. Hence, while the add-on-xDistr has almost constant TAT and VTD for different values of w, it can considerably reduce power dissipation when increasing w. Also, the power reduction is guaranteed for $w \geq 16$. For $w = 8$, reduction is attainable only for $\alpha \leq 4$. Another notable observation is the power profile of the two architectures. This is illustrated in Figure 6(b). It can be easily seen that while the XOR-network has almost constant number of transitions, the add-on-xDistr tends to have short periods of higher number of transitions. These, however, considerably smaller than XOR-network's ones. This is due to the WSCs inputs generated by the two architectures. The XOR-network generates the WSCs inputs with almost random properties, hence the transition probability is high, while the add-on-xDistr generates runs of 0s with lower transition probability.

4. Conclusions and future research directions

Critically viewed, both architectures have their advantages and disadvantages. However, we consider the following to differentiate the two.

XOR-network based architecture [16] by exploiting the spareness of specified bits in the test set can considerably reduce the TAT, especially when w increases. As also stipulated in [16], due to the randomness of the generated output, the test set applied to the CUT might have greater chance to accidentally detect faults not targeted by the care bits. In addition, for small test buses, the area overhead is small, increasing the system integrator's flexibility in these cases.

add-on-xDistr architecture [9] has almost constant area overhead regardless of the value of w. This is an advantage for very large w's. By using WSC partitioning and selection the add-on-xDistr can considerably reduce the average

power dissipation even though the WSCs are driven at the on-chip frequency. In addition, it is a self-synchronous architecture (i.e., all the synchronisation is within the architecture, and no external data lines are required), hence, only one ATE channel is needed when compared to two ATE channels for the XOR-network architecture.

Founded on the quantitative analysis given in Section 3, in the following we provide some guidelines for future research. Accounted for, the issues raised below will improve the four TDCE parameters and benefit the system integrator when considering core based SOC test.

XOR-network based architecture [16] has an area overhead increase almost linear with w. Hence, for large values of w, smaller XOR-networks which provide savings similar to the ones obtained in this paper are required. In addition, the average power dissipation of the XOR-network is almost always greater than the add-on-xDistr' one. It is important to note that these results were obtained while the XOR-network architecture was driven at the ATE clock frequency. Therefore, non-intrusive and test set independent methods which address this problem are also needed.

add-on-xDistr architecture [9] exploits WSC partitioning, however, it does not exploit the test bus width. This is due to the runs of 0s based compression method used with the architecture. However, since the add-on-xDistr is generic enough to be combined with any other parallel decoders [8], it could be merged with compression methods which exploit the test bus width. Due to the fact that add-on-xDistr drives all the WSCs at the on-chip test frequency, it could be used to increase chip dynamics leading to at speed structural test without the problems imposed by the traditional slow/fast clock approach [24].

An important conclusion from the previous analysis is that even if the test sets are compacted there is still a considerable amount of compression. However, both architectures suffer from reduced compression ratios when $\%cb$ is considerably increased. This can be also attributed to the static compaction heuristic used in this paper, since it does not account for any of the particularities of the two methods. In the case of XOR-network, compaction could be performed considering the test bus width – i.e., the number of temporal pattern lockouts is minimum for the considered test bus width. In the case of add-on-xDistr, compaction could be performed considering the runs of 0s, property exploited by the VIHC scheme. Recently, in [12], the test set property has been considered in test set compaction, however, fault simulation was also required. When the above are accounted for, the compression ratio could be improved when $\%cb$ increases, hence further reducing VTD and TAT.

Finally, based on the provided analysis it may be concluded that both architectures are suitable for core based SOC environments. However, addressing the above issues will further improve their TDCE parameters and facilitate the system integrator's selection of the right TDC architecture considering the SOCs architectural' characteristics as well as the system integrator's resources and priorities.

References

[1] Virginia Polytechnic Institute and State University. http://www.ee.vt.edu/˜ha/cadtools/cadtools.html.

[2] I. Bayraktaroglu and A. Orailoglu. Test Volume and Application Time Reduction Through Scan Chain Concealment. In *DAC*, volume 38, pp 151–155, June 2001.

[3] B. Bottoms. The third millennium's test dilemma. *IEEE Design & Test of Computers*, 15(4):7–11, Oct. 1998.

[4] F. Brglez, D. Bryan, and K. Kozminski. Combinational profiles of sequential benchmark circuits. In *ISCAS*, pp 1929–1934, May 1989.

[5] A. Chandra and K. Chakrabarty. Combining Low-Power Scan Testing and Test Data Compression for System-on-a-chip. In *DAC*, volume 38, pp 113–120, June 2001.

[6] A. Chandra and K. Chakrabarty. Frequency-Directed Run-Length (FDR) Codes with Application to System-on-a-Chip Test Data Compression. In *VTS*, pp 114–121, Apr. 2001.

[7] R. Dorsch and H.-J. Wunderlich. Tailoring ATPG for Embedded Testing. In *ITC*, pp 530–537, Oct. 2001.

[8] P. T. Gonciari, B. Al-Hashimi, and N. Nicolici. Improving Compression Ratio, Area overhead, and Test Application Time in System-on-a-Chip Test Data Compression/Decompression. In *DATE*, pp 604–611, Mar. 2002.

[9] P. T. Gonciari, B. Al-Hashimi, and N. Nicolici. Integrated Test Data Decompression and Core Wrapper Design for Low-Cost System-on-a-Chip Testing. In *ITC*, pp 64–73, Oct. 2002.

[10] P. T. Gonciari, B. Al-Hashimi, and N. Nicolici. Reducing Synchronization Overhead in Test Data Compression Environments. In *Digest of Papers ETW*, pp 147–152, May 2002.

[11] P. T. Gonciari, B. Al-Hashimi, and N. Nicolici. Useless Memory Allocation: Problems and Solutions. In *VTS*, pp 423–430, Apr. 2002.

[12] H. Ichihara, K. Kinoshita, I. Pomeranz, and S. Reddy. Test Transformation to Improve Compaction by Statistical Encoding. In *VLSI Design*, pp 294–299, Jan. 2000.

[13] ITRS. The International Technology Roadmap for Semiconductors, 2001 Edition. http://public.itrs.net/.

[14] A. Jas, J. Ghosh-Dastidar, and N. A. Touba. Scan Vector Compression/Decompression Using Statistical Coding. In *VTS*, pp 114–121, Apr. 1999.

[15] A. Khoche and J. Rivoir. I/O Bandwidth Bottleneck for Test: Is it Real ? In *Proceedings of Test Resource Partitioning Workshop*, pp 2.3–1–2.3–6, Nov. 2000.

[16] B. Koenemann, C. Barnhart, B. Keller, T. Snethen, O. Farnsworth, and D. Wheater. A SmartBIST Variat with Guaranteed Encoding. In *ATS*, pp 325–330, Nov. 2001.

[17] E. J. Marinissen et al. On IEEE P1500's Standard for Embedded Core Test. *JETTA*, 18(4), Aug. 2002.

[18] N. Nicolici and B. M. Al-Hashimi. Multiple Scan Chains for Power Minimization during Test Application in Sequential Circuits. *IEEE Computer*, 51(6):721–734, June 2002.

[19] J. Rajski. DFT for High-Quality Low Cost Manufacturing Test. In *ATS*, pp 3–8, Nov. 2001.

[20] S. Reda and A. Orailoglu. Reducing Test Application Time Through Test Data Mutation Encoding. In *DATE*, pp 387–393, Mar. 2002.

[21] P. M. Rosinger, P. T. Gonciari, B. Al-Hashimi, and N. Nicoli. Analysing trade-offs in scan power and test data compression for system-on-a-chip. *IEE Proceedings, Computers and Digital Techniques*, 149(4):188–196, July 2002.

[22] Synopsys Inc. Design compiler reference manual, 2001.

[23] S. Wang and S. Gupta. ATPG for heat dissipation minimization during test application. *IEEE Computer*, 47(2):256–262, Feb. 1998.

[24] B. G. West. At-Speed Structural Test. In *ITC*, pp 795–800, Sept. 1999.

[25] L. Whetsel. Adapting Scan Architectures for Low Power Operation. In *ITC*, pp 863–872, Oct. 2000.

[26] Y. Zorian. A distributed BIST control scheme for complex VLSI devices. In *VTS*, pp 4–9, 1993.