Analysis of Test Application Time for Test Data Compression Methods Based on Compression Codes*

ANSHUMAN CHANDRA

Synopsys, Inc., 700 E. Middlefield Rd., Mountain View, CA 94043, USA
anshuman@synopsys.com

KRISHNENDU CHAKRABARTY

Department of Electrical and Computer Engineering, Duke University, 130 Hudson Hall, Box 90291,
Durham, NC 27708, USA

Received March 16, 2003; Revised October 21, 2003

Editor: H.J. Wunderlich

Abstract. We present an analysis of test application time for test data compression techniques that are used for reducing test data volume and testing time in system-on-a-chip (SOC) designs. These techniques are based on data compression codes and on-chip decompression. The compression/decompression scheme decreases test data volume and the amount of data that has to be transported from the tester to the SOC. We show via analysis as well as through experiments that the proposed scheme reduces testing time and allows the use of a slower tester. Results on test application time for the ISCAS'89 circuits are obtained using an ATE testbench developed in VHDL to emulate ATE functionality.

Keywords: automatic test equipment (ATE), decompression architecture, embedded core testing, precomputed test sets, system-on-a-chip testing, test set encoding, testing time, variable-to-variable-length codes

1. Introduction

A typical system-on-a-chip (SOC) integrates several intellectual property (IP) cores. These cores must be tested using precomputed test sets provided by the core vendor. However, increased design complexity leads to higher test data volume for SOC's, which in turn leads to an increase in testing time [28]. One approach to reduce test time, as well as overcome memory and I/O limitations of automatic test equipment (ATE), is based

on test data compression and on-chip decompression [4, 5, 7, 9, 11, 15–18]. Test data compression is especially appealing for SOC's with IP cores for which BIST data compression techniques based on gate-level structural knowledge are not feasible [14, 20].

Test data compression is an example of a test resource management technique for handling test complexity (see Fig. 1). Test data volume and testing time are decreased by using a combination of coding techniques and faster on-chip decompression of encoded test data. The precomputed test set T_D provided by the core vendor is compressed (encoded) to a much smaller test set T_E and stored in ATE memory. An on-chip decoder is used for pattern decompression to generate T_D from T_E during pattern application. The compressed data can be transferred at a slower rate from the ATE

*This research was supported in part by the National Science Foundation under grants CCR-9875324 and CCR-0204077. This manuscript is based on the authors's paper that appeared in *Proceedings of Design, Automation and Test in Europe (DATE) Conference*, Paris, France, March 2002, pp. 598–603.

Fig. 1. Test data compression and on-chip decompression.

to the SOC. This allows the use of low-end ATEs with less memory and slower clock rates.

Test data compression techniques based on statistical coding [15, 18], variable-to-fixed length coding [17], variable-to-variable-length Golomb coding [4], FDR coding [5], EFDR coding [9], variable-length input Huffman coding (VIHC) [11], and alternating run-length coding [6] have been proposed to reduce test data volume. A number of techniques based on dictionary-based compression methods have also been proposed for SOC test data volume reduction. In [18], variable-length codes are assigned based on the frequencies of the input blocks using modified Huffman codes. The technique presented in [25] uses a dictionary with fixedlength indices to generate all the distinct output vectors. Another technique based on LZ77 compression algorithm uses a dynamic dictionary for test data compression [27].

Other test data compression methods proposed in the literature include a linear mapping network to drive a large number of internal scan chains through a small number of external pins [2], the RESPIN method [8], a technique based on geometric shapes for compressing test vectors [10], and exploiting the don't-care bits in the test vectors, such that these bits are neither stored on the ATE nor transferred to the chip [21]. While all these techniques are aimed at reducing test data volume, they present different design alternatives and their applicability to a particular design varies from case to case.

Recently a number of commercial tools have been introduced in the market to reduce the manufacturing test cost, which is primarily driven by two components of test: test data volume and test application time. These tools use different test data compression techniques to

provide over 10X reduction in test data volume and test time. For example, the SoCBIST tool from Synopsys [26], the TestKompress tool from Mentor Graphics [24], and the OPMISR [1] and SmartBIST [22] tools from Cadence Design Systems reduce test data volume and testing time using test data compression and on-chip decompression.

In this paper, we focus on test data compression based on data compression codes to reduce testing time and test data volume. The proposed scheme, which is based on frequency-directed run-length (FDR) codes for test data compression [5], allows us to readily overcome ATE memory limitation. Unlike [19], it does not require internal details of the core under test. The reduction of test data was demonstrated in earlier work [5]. Here we concentrate on the test application time. We present a testing time analysis to demonstrate that a slower ATE can be used without impacting testing time. Such testing time analysis has not been presented earlier for test data compression schemes based on compression codes.

The remainder of the paper is organized as follows. The test data compression and decompression architecture based on FDR codes is reviewed in Section 2. We present testing time analysis for the single scan chain and the multiple scan chains architectures for run-length codes. The experimental results and the decoder implementation for single scan chain are presented in Section 3, followed by conclusions in Section 4.

2. Test Data Compression

We first review FDR coding and its application to test data compression [5]. The FDR code is a data compression code that maps variable-length runs of 0s to variable-length codewords. It is constructed as follows: The runs of 0s in the data stream are divided into groups $A_1, A_2, A_3, \dots, A_k$, where k is determined by the length $l_{\max}$ of the longest run ($2^k - 3 \leq l_{\max} \leq 2^{k+1} - 3$). Note also that a run of length l is mapped to group A_j where $j = \lceil \log_2(l + 3) \rceil - 1$. The size of the i th group is equal to 2^i i.e., A_i contains 2^i members. Each codeword consists of two parts—a group prefix and a tail. The group prefix is used to identify the group to which the run belongs and the tail is used to identify the members within the group. The encoding procedure is illustrated in Table 1. As an example, consider a run of five 0s ($r_1 = 000001$) in the input stream. Thus r_1 belongs to group A_2 and it is mapped to the codeword 1011. The reader is referred

Table 1. An example of FDR coding.

Group	Run-length	Group prefix	Tail	Codeword
A_1	0	0	0	00
	1		1	01
A_2	2	10	00	1000
	3		01	1001
	4		10	1010
	5		11	1011
A_3	6	110	000	110000
	7		001	110001
	8		010	110010
	9		011	110011
	10		100	110100
	11		101	110101
	12		110	110110
	13		111	110111
...	...	...	...	...

to [3, 5] for a detailed discussion and motivation for the FDR code.

An on-chip decoder decompresses the encoded test set T_E and produces the precomputed test set T_D . Even though T_D contains more patterns than the test sets obtained after static compaction of ATPG vectors, the testing time is reduced since pattern decompression can be carried out on-chip at higher clock frequencies. As discussed in [5], the decoder can be efficiently implemented by a $k_{\max}$ -bit counter, a $\log_2 k_{\max}$ -bit counter and a finite-state machine (FSM), where $k_{\max}$ is the maximum group size encountered during FDR coding of the test data stream. The synthesized decode FSM circuit contains only 4 flip-flops and 38 combinational gates. For any circuit whose test set is compressed using FDR code, the given logic is the only additional hardware required.

2.1. Testing Time Analysis: Single Scan Chain

We first analyze the testing time when a single scan chain is fed by the FDR decoder. Test data compression decreases testing time, and allows the use of a low-cost ATE running at a lower frequency to test the core without imposing any penalties on the total testing time. Let the ATE frequency and the on-chip scan frequency be f_{ATE} and f_{scan} , respectively, where $f_{\text{ATE}} < f_{\text{scan}}$. Since the ATE and the scan chain operate at two different fre-

Fig. 2. Block diagram of FDR decoder partitioned into two frequency domains for TRP.

quencies, the decoder also consists of two parts—one operating at f_{ATE} and the other operating at f_{scan} such that $f_{\text{ATE}} = f_{\text{scan}}/\alpha$, $\alpha > 1$. The parameter α should ideally be a power of 2 since it is easier to synchronize the ATE clock with the scan clock for such values of α [13]. If the scan chain has multiple segments operating at different clock frequencies, each segment has a dedicated decoder for test data decompression.

The proposed scheme is suited for reusing slower and older-generation testers to test chips running at a higher scan frequency than the ATE frequency. Fig. 2 outlines the decoder partitioned into two frequency domains. The decoder communicates with the ATE through the *bit_in* and *en* signals, and the two counters. These counters operate at a frequency of f_{ATE} when the data is being transferred from the ATE to the decoder and switch to a frequency of f_{scan} when the data is being decoded. The proposed scheme therefore decouples the internal scan chain(s) from the ATE via the use of a decoder interface. This decoupling implies that the scan clock frequency is no longer constrained by the ATE clock frequency limitation. Thus a low-cost ATE running at a slower frequency f_{ATE} can be used to test a circuit with a higher scan test frequency f_{scan} .

For the FDR code, let TAT_{SSC}^F (F in the superscript denotes the FDR code) be the test application time for

the entire test set with a single scan chain (SSC). Let T_{shift} be the time required to shift the encoded data from the ATE to the chip and T_{decode} be the time required to decoded the encoded test set, respectively. An upper bound on TAT_{SSC}^F can be obtained by making a pessimistic assumption that the decoding begins after the complete encoded test set is transferred from the ATE to the chip. This implies that

$$TAT_{\text{SSC}}^F \leq T_{\text{shift}} + T_{\text{decode}}.$$

Since data is transferred from the ATE to the chip at the tester frequency, the time required to transfer encoded test set is given by

$$T_{\text{shift}} = \frac{|T_E^F|}{f_{\text{ATE}}}$$

where $|T_E^F|$ is the size of the FDR encoded test set.

Similarly, the time required to decode the encoded test set on-chip is given by

$$T_{\text{decode}} = \frac{|T_D|}{f_{\text{scan}}}$$

where $|T_D|$ is the size of the test set.

Therefore, an upper bound on the test application time for the entire test set is given by

$$\begin{aligned} TAT_{\text{SSC}}^F &\leq \frac{|T_E^F|}{f_{\text{ATE}}} + \frac{|T_D|}{f_{\text{scan}}} \\ &= \frac{|T_E^F|}{f_{\text{ATE}}} + \frac{|T_D|}{\alpha f_{\text{ATE}}}. \end{aligned}$$

For FDR codes, the prefix length and the tail length of the codeword belonging to any group is equal (see Table 1). For example, the codeword for run-length 5 is 1011, where the prefix and the tail are each of size two bits. To derive a lower bound on the testing time, we make an optimistic assumption that the tail bits are shifted in while the prefix is being decompressed. Since the tail bits are now shifted from the ATE while the prefix bits are decoded, the time required to transfer the encoded test set from the ATE to the chip is given by

$$T_{\text{shift}} = \frac{|T_E^F|}{2f_{\text{ATE}}}.$$

Therefore, a lower bound on the test application time for the entire test set is given by

$$TAT_{\text{SSC}}^F \geq \frac{|T_E^F|}{2f_{\text{ATE}}} + \frac{|T_D|}{\alpha f_{\text{ATE}}}.$$

Next, we derive upper and lower bound on the testing time for the test data compression scheme based on Golomb codes that was presented in [4]. It has been shown earlier that FDR codes are more effective in reducing test data volume. Here we compare the two codes in terms of testing time. We analyze the testing time when a single scan chain is fed by the Golomb decoder.

Let TAT_{SSC}^G (G in the superscript denotes the Golomb code) be the test application time for the entire test set with a single scan chain (SSC). The time required to transfer encoded test set from the ATE to the chip at the tester frequency is given by

$$T_{\text{shift}} = \frac{|T_E^G|}{f_{\text{ATE}}}$$

where $|T_E^G|$ is the size of the Golomb encoded test set.

Therefore, an upper bound on the test application time for the entire test set is given by

$$\begin{aligned} TAT_{\text{SSC}}^G &\leq \frac{|T_E^G|}{f_{\text{ATE}}} + \frac{|T_D|}{f_{\text{scan}}} \\ &= \frac{|T_E^G|}{f_{\text{ATE}}} + \frac{|T_D|}{\alpha f_{\text{ATE}}} \end{aligned}$$

For Golomb codes, the prefix length and the tail length of the codeword belonging to any group are not equal (see Table 2). For example, the codeword for run-length 2 and Golomb parameter $m = 4$ is 010, where the prefix is one bit long and the tail is two bits long. However, a lower bound on the testing time can be obtained based on an optimistic assumption that the tail bits are shifted in while the prefix is being decompressed. Therefore,

$$TAT_{\text{SSC}}^G \geq T_{\text{prefix}} + T_{\text{decode}}.$$

where T_{prefix} is the time to shift the prefix data from the ATE to the chip.

As shown in Table 2, the size of the prefix for any group k is equal to k bits. Let the number of codewords belonging to the k th group be $q(k)$ and k_{max} be the

Table 2. An example of Golomb coding for $m = 4$.

Group	Run-length	Group prefix	Tail	Codeword
A_1	0	0	00	000
	1		01	001
	2		10	010
	3		11	011
A_2	4	10	00	1000
	5		01	1001
	6		10	1010
	7		11	1011
A_3	8	110	00	11000
	9		01	11001
	10		10	11010
	11		11	11011
...	...	...	...	...

largest group. Therefore,

$$T_{\text{prefix}} = \frac{\sum_{k=1}^{k_{\max}} kq(k)}{f_{\text{ATE}}}.$$

For Golomb codes,

$$\begin{aligned} |T_E^G| &= \sum_{k=1}^{k_{\max}} (k + \log_2 m) q(k) \\ &= \sum_{k=1}^{k_{\max}} kq(k) + \log_2 m \sum_{k=1}^{k_{\max}} q(k) \end{aligned} \quad (1)$$

Let $|C_G|$ be the total number of codewords such that

$$|C_G| = \sum_{k=1}^{k_{\max}} q(k).$$

Therefore, from Eq. (1)

$$\sum_{k=1}^{k_{\max}} kq(k) = |T_E^G| - \log_2 m |C_G| \quad (2)$$

Therefore, a lower bound on test application time for Golomb codes is given by

$$\begin{aligned} TAT_{\text{SSC}}^G &\geq \frac{|T_E^G| - \log_2 m |C_G|}{f_{\text{ATE}}} + \frac{|T_D|}{f_{\text{scan}}} \\ &= \frac{|T_E^G|}{f_{\text{ATE}}} + \frac{|T_D|}{\alpha f_{\text{ATE}}} - \frac{\log_2 m |C_G|}{f_{\text{ATE}}}. \end{aligned}$$

Table 3. An example of conventional run-length coding for $M = 7$.

Group	Run-length	Codeword
A_1	0	000
	01	001
	001	010
	0001	011
	00001	100
	000001	101
	0000001	110
A_2	00000001	111
	000000001	000111
	0000000001	001111
A_3	00000000001	010111
	...	...
	...	...

We now present an upper bound on the testing time for the test data compression scheme based on a conventional run-length code (see Table 3). Let TAT_{SSC}^R (R in the superscript denotes the conventional run-length code) be the test application time for the entire test set with a single scan chain (SSC). The time required to transfer encoded test set from the ATE to the chip at the tester frequency is given by

$$T_{\text{shift}} = \frac{|T_E^R|}{f_{\text{ATE}}}$$

where $|T_E^R|$ is the size of the conventional run-length encoded test set.

Therefore, an upper bound on the test application time for the entire test set is given by

$$\begin{aligned} TAT_{\text{SSC}}^R &\leq \frac{|T_E^R|}{f_{\text{ATE}}} + \frac{|T_D|}{f_{\text{scan}}} \\ &= \frac{|T_E^R|}{f_{\text{ATE}}} + \frac{|T_D|}{\alpha f_{\text{ATE}}}. \end{aligned}$$

Since conventional run-length code is a variable-to-fixed-length code, the codeword is not composed of a prefix and a tail. Therefore, unlike the FDR code and the Golomb code, a codeword cannot be divided into parts to do parallel decoding while shifting the data from the ATE. Hence, an expression for lower bound on testing time cannot be derived in a straightforward manner.

We next compare the testing time using the proposed scheme with that for an ATPG-compacted test

set with p patterns and an external tester operating at frequency f_{ATE}^* . The patterns used here are fully compacted and are directly applied to the circuit under test (CUT) at the frequency f_{ATE}^* . In the compression/decompression schemes presented earlier, uncompacted test sets were used that are stored in the ATE in compacted form. These are transferred to the CUT at a different ATE frequency f_{ATE} and then uncompressed on-chip at the scan frequency f_{scan} . Let the length of the scan chain be n bits. The size of the ATPG-compacted test set is pn bits and the test application time TAT_{SSC}^{ATPG} equals pn/f_{ATE}^* . We now derive the ratio $\gamma = f_{ATE}^*/f_{ATE}$ such that $TAT_{SSC}^{ATPG} = TAT_{SSC}^F$. An upper and a lower bound on γ can be derived using the upper and lower bound values of TAT_{SSC}^F . Therefore,

$$\frac{pn}{|T_E^F| + \frac{T_D}{\alpha}} \geq \gamma \geq \frac{pn}{\frac{|T_E^F|}{2} + \frac{T_D}{\alpha}}.$$

Experimental results presented in Section 3 show that the testing time is reduced considerably using the proposed method if $f_{ATE}^* = f_{ATE}$. Moreover, if the same testing time is desired using a slower ATE ($f_{ATE}^* > f_{ATE}$), the ratio γ is especially high for larger values of α . Hence FDR coding and Golomb coding allow us to decrease the volume of test data and use a slower tester without increasing testing time. Experimental results show however that, compared to Golomb codes, FDR codes are more efficient in reducing testing time.

To conclude the analysis, we note that the above bounds allow us to evaluate the testing time without a detailed analysis of the asynchronous handshaking protocol between the tester and the decoder. The exact testing time, which lies between the two bounds can be determined through a bit-by-bit analysis of the encoded test data. In Section 3, we present the methodology that we have used to determine the exact testing time for benchmark circuits. The results show that the exact testing time lies between the lower and upper bounds. Nevertheless, the formulation based on upper and lower bounds allows us to demonstrate the effectiveness of the proposed scheme without resorting to such detailed analysis.

2.2. Testing Time Analysis: Multiple Scan Chains

Let us consider a core under test with s scan chains, each of length l (see Fig. 3). The FDR decoder receives data as a single stream of bits and outputs the decoded data in a single bit stream. Therefore, a limita-

Fig. 3. Decompression architecture for a core under test with multiple scan chains.

tion of the scheme based on FDR codes is that an extra shift register is required to drive multiple scan chains. However, the number of input channels required to feed these chains is reduced to a single input channel. For the Illinois scan architecture, the FDR decoder can be used to feed the scan chains without using an extra scan register, thus saving the extra test clock cycles needed to fill the register before shifting the data into the scan chains. For our analysis, we assume that the scan chains in the embedded cores cannot be redesigned. The cores are available to the system integrator as optimized layouts.

A shift register SR is used to serially shift data from the decoder and then load the scan chains in parallel (SR can be configured out of the first scan cell of each scan chain) and a multiple-input signature register (MISR) is used for logging the captured responses. The MISR compresses the accumulated responses into a signature. Two counters are used to indicate that SR and the scan chains are loaded. The counter used for SR and the scan chains are $\log_2 s$ bits and $\log_2 l$ bits long, respectively. SR is loaded with a new bit only when the decoder output is valid and $\log_2 s$ bit counter is incremented for every valid bit. When SR is fully loaded, the scan clock is enabled to shift the data from SR to the scan chains and from the scan chains into the MISR. For every bit loaded into the scan chains, the $\log_2 l$ -bit counter is incremented. When the scan chain is completely loaded, the functional clock is enabled to apply the pattern to the core.

The contents of the MISR at the completion of the test provide a signature. This signature is compared with the expected signature to determine the test outcome. We assume here that aliasing can either be

neglected or made insignificant by increasing the MISR size.

Let the number of test patterns to be applied to the core under test shown in Fig. 3 be p . The number of cycles required to fill SR and shift the data from SR to the scan chains is given by $(s + 1)$. The number of cycles required to shift a single pattern in the l scan chains is given by $\sum_{i=1}^l (s + 1) = (s + 1)l$. Therefore, for p patterns, the number of cycles required is given by $(s + 1)lp = lps + lp$. This is lp cycles more than the time needed to shift in the test pattern if a single scan chain is used. We can now apply the analysis for single scan chain architecture described in the previous section. Let TAT_{SSC} and TAT_{MSC} be the testing time for the core with a single scan chain and multiple scan chains, respectively. Here, TAT_{SSC} corresponds to the time required for testing a core with a single scan chain using an on-chip decoder. Since the data is shifted into SR from the decoder at the scan frequency, the testing time for the multiple scan chains architecture is given by the sum of testing time of the single scan chain and the time required for the extra lp shift cycles, i.e.,

$$TAT_{MSC} = TAT_{SSC} + \frac{lp}{f_{scan}}. \quad (3)$$

The bounds on the testing time for the multiple scan chain architecture can now be obtained by substituting the expressions corresponding to the lower and upper bounds for the single scan chain architecture, derived in the previous section, for TAT_{SSC} in (3).

The test set has to be reorganized for the above test architecture before applying test data compression. Let the test pattern j for the i th scan chain be $t_j^i = b_{j1}^i b_{j2}^i b_{j3}^i \dots b_{jl}^i$, where b_{jk}^i is the k th bit of the j th pattern and the i th scan chain. Let t_j^{MSC} be the j th pattern of the modified test set for a core with multiple scan chains (MSC). Therefore, t_j^{MSC} is obtained by forming a single stream of bits by interleaving the bits of the j th pattern of each scan chain i.e., $t_j^{MSC} = b_{j1}^1 b_{j1}^2 b_{j1}^3 \dots b_{j1}^l b_{j2}^1 b_{j2}^2 b_{j2}^3 \dots b_{j2}^l, \dots, b_{jl}^1 b_{jl}^2 b_{jl}^3 \dots b_{jl}^l$. Fig. 4 illustrates the procedure of obtaining the new test pattern for four scan chains. The testing time analysis of Section 2.1 can now be directly applied to the FDR or Golomb coding of the modified test data sequence. Experimental results comparing the testing times for the single scan chain and the multiple scan chains architectures are presented in Section 3.

Fig. 4. Generating the interleaved data stream for a circuit with four scan chains.

3. Experimental Results

In this section, we present experimental results on the testing time for the proposed scheme based on FDR coding, Golomb coding and conventional run-length coding. We also present the decoder design and the experimental setup used for determining the actual testing time using the FDR code. The effectiveness of FDR coding for test data volume reduction was shown in [5]; these results are therefore not presented here. We have used uncompacted and compacted test sets for the large ISCAS-89 benchmark circuits obtained using the Mintest ATPG program, which is known to yield the most compact test sets for the benchmark circuits.

3.1. Experimental Validation for a Single Scan Chain

The decoder design for the FDR code was described in [5]. In this section we present modifications to the decoder design required to determine the testing time for a single scan chain. We developed an ATE test-bench in VHDL to emulate ATE functionality. We also developed VHDL models for the different benchmark circuits with an on-chip decoder to conduct our experiments. The decoder presented in [5] had only one control signal en communicating with the ATE. To develop a better handshaking scheme between the ATE and the FDR decoder, another control signal en_DEC has been added and minor modifications have been made to the FSM.

The decoder block diagram and the FSM are shown in Figs. 5 and 6, respectively. As seen in Fig. 5, the inputs to the FSM are en_FSM , which signals that the next bit from the ATE is available; bit_in , the data from the ATE; $rs1$ and $rs2$, which signal when the k -bit and $\log_2 k$ -bit counters of the decoder, respectively, are initially zero. The signal en_DEC , from the ATE, changes

Fig. 5. FDR decoder block diagram.

Fig. 6. State transition diagram of the modified FDR decoder FSM diagram.

value when the ATE sends a new bit through *bit_in*. A flip-flop and an exclusive-or gate detect the change in *en_DEC*, causing *en_FSM* to go high on an edge of *en_DEC*. The outputs *out* and *v* are outputs of the decoder, used as input to the scan chain. The signal *en_ATE* indicates to the ATE that the FSM is busy.

When *en_ATE* = 1, the FSM indicates that it is able to accept another bit from the ATE; otherwise, the FSM is busy and cannot accept the next bit. The signal *en_ATE* is zero only in states S3 and S7 when the FSM needs to output zeros. Outputs *shift* and *counter_in* feed the *k*-bit counter for reading in the group prefix in states S0 through S3, and for reading in the tail in states S4 thru S7. Outputs *dec2* and *inc* feed the $\log_2 k$ -bit counter to keep track of the number of bits in the group prefix.

The FSM has initial state S0, which indicates to the ATE that it is ready for a bit by setting *en_ATE* = 1. This signal remains high until the complete group prefix is received. The group prefix is read in and decoded in states S0 through S3. A transition from state S0 to state S4 is made if the group prefix is a single bit, i.e., it is zero. Otherwise, the FSM transitions to state S1, where it oscillates with state S2 until the final bit of the group prefix is received. The FSM reaches state S2 if it needs to wait for the next bit from the ATE; otherwise if the next bit is already available, it stays in state S1. From states S1 and S2, the FSM moves to state S3 when *bit_in* = 0, which is the final bit of the group prefix. The FSM then remains in state S3 and brings *dec1* high while outputting 0s based on the group prefix that was shifted into the *k*-bit counter in states S0 through S2.

When *rs1* indicates that the *k*-bit counter is zero, the FSM transitions to state S4. States S4 through S7 handle and decode the tail of the FDR codeword. In state S4, the FSM awaits the next bit from the ATE. In states S4 through S6, the tail is shifted into the *k*-bit counter. From states S4 through S6, a transition to state S5 occurs if *bit_in* = 0, and a transition to S6 occurs if *bit_in* = 1. State S7, similar to state S3 for

Table 4. Comparison of testing time using the proposed method with traditional scan-based external testing.

Circuit	f_{ATE} (MHz)	α	Upper bound on TAT_{SSC}^R (ms)	Lower bound on TAT_{SSC}^G (ms)	Upper bound on TAT_{SSC}^G (ms)	Lower bound on TAT_{SSC}^F (ms)	Upper bound on TAT_{SSC}^G (ms)	Simulation results for the FDR code (ms)	TAT_{SSC}^{NTPG} (ms)
s5378	20	4	0.956	0.571	0.846	0.526	0.756	0.691	1.037
		8	0.807	0.423	0.698	0.378	0.551	0.596	1.037
		16	0.733	0.349	0.623	0.303	0.533	0.491	1.037
s9234	20	4	1.480	0.885	1.279	0.877	1.263	1.153	1.296
		8	1.234	0.639	1.033	0.631	1.018	0.911	1.296
		16	1.111	0.516	0.911	0.509	0.895	0.816	1.296
s13207	20	4	5.472	2.712	3.359	2.574	3.083	2.962	8.155
		8	4.439	1.679	2.326	1.541	2.050	1.929	8.155
		16	3.923	1.163	1.810	1.025	1.534	1.435	8.155
s15850	20	4	2.804	1.606	2.249	1.502	2.041	1.892	2.871
		8	2.323	1.129	1.768	1.020	1.560	1.418	2.871
		16	2.082	0.884	1.528	0.780	1.320	1.214	2.871
s38417	20	4	6.439	3.785	5.512	3.485	4.912	4.516	5.657
		8	5.409	2.756	4.482	2.456	3.882	3.537	5.657
		16	4.895	2.241	3.967	1.941	3.368	3.113	5.657
s38584	20	4	7.613	4.499	6.509	4.247	6.005	5.498	8.052
		8	6.369	3.254	5.265	3.002	4.760	4.299	8.052
		16	5.747	2.632	4.643	2.380	4.138	3.799	8.052

the group prefix, decodes the tail by outputting zeros based on the value in the k -bit counter. If the tail is zero and the group prefix is only a single bit, then the FSM transitions to state S0 from state S5 instead of going to state S7, and a 1 is output since the FDR codeword is 00. If the group prefix was only a single bit but the tail is one, then the FSM goes to state S7 from state S6 so that a zero will be output before a one.

Table 4 presents test application time for the proposed method and for traditional scan-based testing with $f_{ATE}^* = f_{ATE}$. We note that in all the cases the upper bounds on test application time using the proposed scheme for the Golomb code and the FDR code are lower than that for scan-based external testing. We also note that the actual test application time for the proposed scheme lies between the lower and upper bounds. For example, the test application time for s38584 with $\alpha = 8$, and $f_{ATE}^* = f_{ATE} = 20$ MHz is 4.299 ms, and it lies between 3.002 ms and 4.760 ms, which is lower than the time of 8.052 ms required for external testing. This reduction in testing time is obtained without increasing the ATE frequency.

Table 5 presents test application time for the proposed method for the single scan chain and the multiple scan chains architectures, and for traditional scan-based testing with $f_{ATE}^* = f_{ATE}$. The table lists the number of scan chains s , length of each scan chain l

Table 6. Lower and upper bounds on $\gamma = f_{ATE}^*/f_{ATE}$ for the FDR code.

Circuit	$(\gamma_{\min}, \gamma_{\max})$		
	$\alpha = 4$	$\alpha = 8$	$\alpha = 16$
s5378	(1.37, 1.97)	(1.70, 2.74)	(1.94, 3.41)
s9234	(1.02, 1.47)	(1.27, 2.05)	(1.44, 2.54)
s13207	(2.64, 3.16)	(3.97, 5.28)	(5.31, 7.95)
s15850	(1.46, 1.91)	(1.88, 2.81)	(2.2, 3.68)
s38417	(1.15, 1.62)	(1.45, 2.30)	(1.67, 2.91)
s38584	(1.34, 1.89)	(1.69, 2.68)	(1.94, 3.38)

and the number of patterns p applied to the core. We note that except for the case of s9234 ($\alpha = 4$), in each case the upper bound on test application time using the proposed scheme for the FDR code is lower than that for scan-based external testing. Note that the testing time tends to increase slightly for the multiple scan chains architecture (the lower and upper bounds are higher). This is expected since we are using only a single ATE I/O channel and serializing test data at the scan inputs.

Table 6 shows lower and upper bounds on $\gamma = f_{ATE}^*/f_{ATE}$. Recall that we are attempting to use a slower ATE with frequency f_{ATE} , yet have the same testing time as that for a faster ATE with frequency f_{ATE}^* , which

Table 5. Comparison of testing time using the proposed method for the single scan chain and the multiple scan chains architectures.

Circuit	(l, s, p)	f_{ATE} (MHz)	α	Lower bound		Upper bound		
				TAT_{SSC}^F (ms)	TAT_{MSC}^F (ms)	TAT_{SSC}^F (ms)	TAT_{MSC}^F (ms)	TAT_{SSC}^{ATPG} (ms)
s9234	(19, 13, 159)	20	4	0.877	0.915	1.263	1.301	1.296
			8	0.631	0.650	1.018	1.037	1.296
			16	0.509	0.518	0.895	0.905	1.296
s13207	(35, 20, 236)	20	4	2.574	2.677	3.083	3.186	8.155
			8	1.541	1.593	2.050	2.102	8.155
			16	1.025	1.051	1.534	1.560	8.155
s15850	(47, 13, 126)	20	4	1.502	1.576	2.041	2.115	2.871
			8	1.020	1.057	1.560	1.597	2.871
			16	0.780	0.798	1.320	1.338	2.871
s38417	(104, 16, 99)	20	4	3.485	3.614	4.912	5.041	5.657
			8	2.456	2.520	3.882	3.947	5.657
			16	1.941	1.973	3.368	3.400	5.657
s38584	(183, 8, 136)	20	4	4.247	4.558	6.005	6.316	8.052
			8	3.002	3.158	4.760	4.916	8.052
			16	2.380	2.458	4.138	4.216	8.052

applies compacted test patterns to the core under test. We assume a single scan chain for each of the benchmark circuits, and use the analytical results of Section 2.1. The bounds on γ are listed for various values of α , the ratio between the on-chip frequency f_{scan} and f_{ATE} . For example, if $\alpha = 8$ for the benchmark s13207, i.e. $f_{\text{scan}}/f_{\text{ATE}} = 8$, and the same testing time is desired for the two methods, we need an ATE that runs between 3.97 and 5.28 times faster than the ATE required with the proposed scheme. In other words, if $f_{\text{scan}} = 200$ MHz and $f_{\text{ATE}} = 25$ MHz, a faster ATE that runs at a frequency between 99 MHz and 132 MHz will be needed if test data compression is not used.

We now present a lower bound on the ratio of f_{scan} and f_{ATE} , $\alpha_{\min} = f_{\text{scan}}/f_{\text{ATE}}$, that results in maximum reduction in test application time. Minimum test application time is obtained when the ATE does not need to wait for the decoder to finish decompressing data. Since, data is transferred from the ATE at a much slower rate than the rate at which decoding takes place, minimum testing time can be obtained by running the decoder at a rate such that it is always ready to receive data from the tester. This can be achieved for each decoding method depending on the generic parameters associated with the corresponding coding algorithm.

As presented in [16], the lower bound on α is determined by parameter m for Golomb coding. Since, the decoder has to output m zeros for every 1 in the prefix, $\alpha_{\min} \geq m$ ensures that the tester never has

to wait for the decoder to finish decoding. Similarly, for FDR coding with parameter k (where k represents the group to which the longest run of zeros belongs), decoder is busy decoding for 2^k cycles in the worst case. Hence, the lower bound can be achieved for $\alpha_{\min} \geq 2^k$. For conventional run-length codes with block size $b = \log_2(M + 1)$, the decoder is busy decoding M zeros for the last member of the group in the worst case. Therefore, lower bound can be achieved for $\alpha_{\min} \geq M$.

Table 7 shows the lower bound for α corresponding to each code and the upper bound on the test application time. We notice that for all cases, value of $\alpha_{\min}$ is very high for FDR code. This implies that we need to run the chip at a much faster frequency than the ATE to achieve minimum testing time using FDR code. However, if we look at the testing times reproduced from Table 5, the upper bound on test application time using FDR code is lower than the run-length code and Golomb code even if $\alpha \leq \alpha_{\min}$. This is because very few runs of 0s lie in the last group and the decoding time corresponding to these runs contributes a very small amount to the total test application time. This can also be attributed to the fact that the compression obtained using FDR code is higher than Golomb and conventional run-length code. Hence, FDR code provides a lower testing time for an α that provides minimum test application time for conventional run-length and Golomb code.

Table 7. Comparison of lower bound $\alpha_{\min}$ required to achieve minimum test application time with the α used for computing the upper bound on TAT_{SSC} .

Circuit	$\alpha_{\min}$			Upper bound on TAT_{SSC} (ms)		
	Run-length code	Golomb code	FDR code	Run-length code	Golomb code	FDR code
s5378	7	4	128	0.807 ($\alpha = 8$)	0.846 ($\alpha = 4$)	0.551 ($\alpha = 8$)
s9234	7	4	64	1.234 ($\alpha = 8$)	1.279 ($\alpha = 4$)	1.018 ($\alpha = 8$)
s13207	7	16	512	4.439 ($\alpha = 8$)	1.810 ($\alpha = 16$)	1.534 ($\alpha = 16$)
s15850	7	8	512	2.323 ($\alpha = 8$)	1.768 ($\alpha = 8$)	1.560 ($\alpha = 8$)
s38417	7	4	1024	5.409 ($\alpha = 8$)	5.512 ($\alpha = 4$)	3.882 ($\alpha = 8$)
s38584	7	8	512	6.369 ($\alpha = 8$)	5.265 ($\alpha = 8$)	4.760 ($\alpha = 8$)

Table 8. Test application time obtained for CKT1 from IBM.

Scan vector	f_{ATE} (MHz)	α	Lower bound on TAT_{SSC}^G (ms)	Upper bound on TAT_{SSC}^G (ms)	Lower bound on TAT_{SSC}^F (ms)	Upper bound on TAT_{SSC}^F (ms)
1	20	4	14.944	17.000	14.088	15.288
		8	8.500	10.556	7.644	8.844
		16	5.278	7.334	4.422	5.622
2	20	4	14.886	16.884	14.128	15.369
		8	8.442	10.440	7.684	8.924
		16	5.220	7.218	4.462	5.702
3	20	4	14.760	16.633	14.063	15.237
		8	8.316	10.189	7.618	8.793
		16	5.094	6.967	4.396	5.571
4	20	4	14.713	16.539	13.973	15.508
		8	8.269	10.094	7.529	8.614
		16	5.047	6.872	4.307	5.391

Finally, we present testing time results for a production circuit (CKT1) from IBM. This microprocessor design consists of 1.2 million gates and 32200 latches. Results on test data volume for this circuit were presented in [7]. Table 8 shows the bounds on the test application time for CKT1 using the Golomb and FDR encoded test sets. T_D for this consists of a set of 4 scan vectors (a total of 1031072 bits of test data per vector). Since the statically-compacted vectors for CKT1 were not available, we were unable to compare the bounds on the test application time for the proposed scheme with the test application time for the traditional scan-based testing. Therefore, we compare the lower and upper bounds on the test application time for the Golomb and the FDR encoded data. We note that for all the scan vectors, the bounds on the test application time for the FDR encoded data are lower than for the Golomb encoded data. The actual testing time using the proposed scheme lies between the upper and lower bound. These results show that the proposed technique can be employed for reducing test application time for industrial designs and that the test application time for FDR-encoded data is lower than that for the Golomb-encoded data.

4. Conclusion

In this paper, we have shown that test data compression is an efficient technique to increase test effectiveness, reduce testing time and decrease test cost. The proposed scheme makes use of conventional run-length, Golomb

and FDR codes, internal scan chain and boundary scan chain of the core under test and enables the use of low-cost testers. Previous work on test data compression has shown that FDR code reduces the ATE memory required for testing. In this paper, we have shown that it also reduces the testing time, and helps achieve high quality test using a slower tester without any penalty on testing time.

Experimental results for the ISCAS benchmarks show that the proposed scheme is very efficient for reducing testing time. The on-chip decompression of test pattern decouples the internal scan chain(s) from the ATE, thereby allowing an ATE running at a lower frequency to test a circuit running at a higher scan clock frequency. We have presented a testing time analysis for compression/decompression based on conventional run-length, Golomb and FDR codes for single and multiple scan chain architectures. Experimental results for the ISCAS-89 benchmark circuits show that a slower ATE can often be used with no adverse impact on testing time. Therefore, the proposed approach not only decreases test data volume and the amount of data that must be transferred from the ATE, but it also reduces testing time and facilitates the use of less expensive ATEs.

Acknowledgments

We thank Brion Keller of Cadence Design Systems (previously with IBM Corporation) for providing scan

vectors for the production circuit. We also thank Sharon Stewart Schweizer for helping us in our experiments to determine the testing time for the single scan chain architecture.

References

1. C. Barnhart, V. Brunkhorst, F. Distler, O. Farnsworth, B. Keller, and B. Koenemann, "OPMISR: The Foundation for Compressed ATPG Vectors," in *Proceedings of International Test Conference*, 2001, pp. 748–757.
2. I. Bayraktaroglu and A. Orailoglu, "Test Volume and Application Time Reduction through Scan Chain Concealment," in *Proceedings of ACM/IEEE Design Automation Conference*, 2001, pp. 151–155.
3. A. Chandra and K. Chakrabarty, "Test Resource Partitioning for SOC's," *IEEE Design & Test of Computers*, vol. 18, pp. 80–91, 2001.
4. A. Chandra and K. Chakrabarty, "System-on-a-Chip Test Data Compression and Decompression Architectures Based on Golomb Codes," *IEEE Transactions on Computer-Aided Design of Integrated Circuits & Systems*, vol. 20, pp. 355–368, 2001.
5. A. Chandra and K. Chakrabarty, "Test Data Compression and Test Resource Partitioning for System-on-a-Chip Using Frequency-Directed Run-Length (FDR) Codes," *IEEE Transactions on Computers*, vol. 52, May 2003, to appear.
6. A. Chandra and K. Chakrabarty, "A Unified Approach to Reduce SOC Test Data Volume, Scan Power and Testing Time," *IEEE Transactions on Computer-Aided Design of Integrated Circuits & Systems*, vol. 22, March 2003, accepted for publication.
7. A. Chandra, K. Chakrabarty, and R.A. Medina, "How Effective are Compression Codes for Reducing Test Data Volume?," in *Proceedings of VLSI Test Symposium*, 2002, pp. 91–96.
8. R. Dorsch and H.-J. Wunderlich, "Tailoring ATPG for Embedded Testing," in *Proceedings of International Test Conference*, 2001, pp. 530–537.
9. A. El-Maleh and R. Al-Abaji, "Extended Frequency-Directed Run-Length Codes with Improved Application to System-on-a-Chip Test Data Compression," in *Proceedings of International Conference of Electronics, Circuits and Systems*, 2002, pp. 449–452.
10. A. El-Maleh, S. al Zahir, and E. Khan, "A Geometric-Primitives-Based Compression Scheme for Testing Systems on-Chip," in *Proceedings of VLSI Test Symposium*, 2001, pp. 54–59.
11. P.T. Gonciari and B. Al-Hashimi, "Improving Compression Ratio, Area Overhead, and Test Application Time for System-on-a-Chip Test Data Compression/Decompression," in *Proceedings of Design, Automation and Test in Europe (DATE) Conference*, 2002, pp. 604–611.
12. I. Hamzaoglu and J.H. Patel, "Test Set Compaction Algorithms for Combinational Circuits," in *Proceedings of International Conference on Computer-Aided Design*, 1998, pp. 283–289.
13. D. Heide, S. Dhong, P. Hofstee, M. Immediato, K. Nowka, J. Silberman, and K. Stawiasz, "High-Speed Serializing/De-Serializing Design-for-Test Methods for Evaluating a 1 GHz Microprocessor," in *Proceedings of VLSI Test Symposium*, 1998, pp. 234–238.
14. S. Hellebrand, B. Reeb, S. Tarnick, and H.J. Wunderlich, "Pattern Generator for a Deterministic BIST Scheme," in *Proceedings of International Conference on Computer-Aided Design*, 1995, pp. 88–94.
15. V. Iyengar, K. Chakrabarty, and B.T. Murray, "Deterministic Built-in Pattern Generation for Sequential Circuits," *Journal of Electronic Testing: Theory and Applications*, vol. 15, pp. 97–114, 1999.
16. A. Jas, J. Ghosh-Dastidar, M. Ng, and N.A. Touba, "Efficient Test Vector Compression Scheme Using Selective Huffman Coding," *IEEE Transactions on Computer-Aided Design of Integrated Circuits & Systems*, vol. 22, June 2003, to appear.
17. A. Jas and N.A. Touba, "Test Vector Decompression via Cyclical Scan Chains and its Application to Testing Core-Based Design," in *Proceedings of International Test Conference*, 1998, pp. 458–464.
18. A. Jas and N.A. Touba, "Scan Vector Compression/Decompression Using Statistical Coding," in *Proceedings of VLSI Test Symposium*, 1999, pp. 114–120.
19. S. Kajihara, K. Taniguchi, I. Pomeranz, and S. M. Reddy, "Test Data Compression Using Don't-Care Identification and Statistical Encoding," in *Proceedings of International Workshop on Electronic Design, Test and Applications*, 2002, pp. 413–416.
20. G. Kiefer and H.-J. Wunderlich, "Using BIST Control for Pattern Generation," in *Proceedings of International Test Conference*, 1997, pp. 347–355.
21. A. Khoche, E. Volkerink, J. Rivoir, and S. Mitra, "Test Vector Compression Using EDA-ATE Synergies," in *Proceedings of VLSI Test Symposium*, 2002, pp. 97–102.
22. B. Koenemann et al., "A SmartBIST Variant with Guaranteed Encoding," in *Proceedings of Asian Test Symposium*, 2001, pp. 325–330.
23. H.K. Lee and D.S. Ha, "An Efficient Forward Fault Simulation Algorithm Based on the Parallel Pattern Single Fault Propagation," in *Proceedings of International Test Conference*, 1991, pp. 946–955.
24. J. Rajski et al., "Embedded Deterministic Test for Low Cost Manufacturing Test," in *Proceedings of International Test Conference*, 2002, pp. 301–310.
25. S.M. Reddy, K. Miyase, S. Kajihara, and I. Pomeranz, "On Test Data Volume Reduction for Multiple Scan Chain Designs," in *Proceedings of VLSI Test Symposium*, 2002, pp. 103–108.
26. Synopsys, Inc., http://www.synopsys.com/products/test/dft_socbist_ds.pdf.
27. F. G. Wolff and C. Papachristou, "Multiscan-Based Test Compression and Hardware Decompression Using LZ77," in *Proceedings of International Test Conference*, 2002, pp. 331–339.
28. Y. Zorian, "Testing the Monster Chip," *IEEE Spectrum*, vol. 36, no. 7, pp. 54–70, 1999.

Anshuman Chandra is a Research and Development Engineer at Synopsys. His research interests focus on VLSI testing. He received the B.E. degree in electrical engineering from the University of Roorkee, Roorkee, India, in 1998, and the M.S. and Ph.D. degrees in electrical and computer engineering from Duke University, Durham, NC, in 2000 and 2002, respectively.

Dr. Chandra is a member of IEEE. He received the Test Technology Technical Council James Beausang Student Paper Award for a paper in Proc. 2000 IEEE VLSI Test Symposium. He is also a recipient of a Best Paper Award for the 2001 Design Automation and Test in Europe (DATE) Conference.

Krishnendu Chakrabarty received the B. Tech. degree from the Indian Institute of Technology, Kharagpur, in 1990, and the M.S.E. and Ph.D. degrees from the University of Michigan, Ann Arbor, in 1992 and 1995, respectively, all in Computer Science and Engineering. He is now Associate Professor of Electrical and Computer Engineering at Duke University. During 2000–2002, he was also a Mercator Visiting Professor at University of Potsdam in Germany.

Dr Chakrabarty is a recipient of the National Science Foundation Early Faculty (CAREER) award and the Office of Naval Research Young Investigator award. His current research projects include: design and testing of system-on-chip integrated circuits; embedded real-time systems; distributed sensor networks; modeling, simulation and optimization of microfluidic systems; microfluidics-based chip cooling. Dr Chakrabarty is a co-author of two books: *Microelectrofluidic Systems: Modeling and Simulation* (CRC Press, 2002) and *Test Resource Partitioning for System-on-a-Chip* (Kluwer, 2002), and the editor of SOC (*System-on-a-Chip*) *Testing for Plug and Play*

Test Automation (Kluwer, 2002). He has published over 150 papers in journals and refereed conference proceedings, and he holds a US patent in built-in self-test. He is a recipient of a best paper award at the 2001 Design, Automation and Test in Europe (DATE) Conference. He is also a recipient of the Humboldt Research Fellowship, awarded by the Alexander von Humboldt Foundation, Germany.

Dr Chakrabarty is an Associate Editor of *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, an Editor of *Journal of Electronic Testing: Theory and Applications (JETTA)*, and a member of the editorial board for *Sensor Letters* and *Journal of Embedded Computing*. He has also served as an Associate Editor of *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*. He is a senior member of IEEE, a member of ACM and ACM SIGDA, and a member of Sigma Xi. He serves as Vice Chair of Technical Activities in IEEE's Test Technology Technical Council, and is a member of the program committees of several IEEE/ACM conferences and workshops.