Reduction of SOC Test Data Volume, Scan Power and Testing Time Using Alternating Run-length Codes[*]

Anshuman Chandra and Krishnendu Chakrabarty
Dept. Electrical and Computer Engineering
Duke University
Durham, NC 27705, USA.
{achandra, krish}@ee.duke.edu

ABSTRACT

We present a test resource partitioning (TRP) technique that simultaneously reduces test data volume, test application time and scan power. The proposed approach is based on the use of alternating run-length codes for test data compression. Experimental results for the larger ISCAS-89 benchmarks and an IBM production circuit show that reduced test data volume, test application time and low power scan testing can indeed be achieved in all cases.

1. INTRODUCTION

Intellectual property (IP) cores are now commonly used in large system-on-a-chip (SOC) designs. Although IP cores help reduce design cycle time, they pose several test challenges. The system integrator is confronted with the problems of test data volume, test application time and power consumption during test. New techniques based on test resource partitioning (TRP) that reduce test data volume, testing time and power during testing are therefore necessary to facilitate plug-and-play SOC test automation.

Power consumption in test mode is considerably higher than in normal mode [1]. Test data volume and test application time are two additional problems faced in SOC test integration. Current techniques do not provide a unified solution to the three problems of test data volume, test application time and test power. These techniques typically provide point solutions that target at most two out of the above three objectives.

Structural methods for reducing test data volume and testing time typically require design modifications. For example, the Illinois scan architecture (ILS) [2, 3] uses a single scan-in pin to simultaneously feed multiple scan chains during broadcast mode. However, a drawback of the ILS architecture is that it does not address the problem of reducing power consumption during scan testing.

In a different approach, which can be described as an algorithmic strategy, 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 [16, 17]. These compression techniques are typically based on run-length codes and their variants, e.g. Golomb and FDR codes. A particularly attractive feature of TRP based on compression methods is that it does not require any redesign of IP cores.

Another way to reduce test data volume and testing time is to use BIST [4]. However, BIST can only be applied to SOCs if the IP cores in them are BIST-ready. Since most currently-available IP cores are not BIST-ready, the incorporation of BIST in them requires considerable redesign.

A number of techniques to control power consumption in test mode have also been presented in the literature. These can be broadly classified as (a) structural (b) algorithmic, and (c) tester-based.

1. *Structural methods*: These methods, which do not address test data volume or testing time, involve gated scan chains [5], tailored test generation circuits [6, 9], modified scan latch and vector inhibition [7, 8], and scan chain organization [10].

2. *Algorithmic methods*: These include automatic test pattern generation (ATPG) under power constraints, test data compression, and test scheduling algorithms. ATPG techniques for generating vectors for low power testing usually leads to an increase in the number of test vectors [11]. On the other hand, static compaction of scan vectors causes significant increase in power consumption during testing [13]. While compacted vectors are useless if they exceed power constraints, uncompacted vectors cannot be used as they require excessive tester memory. In order to address this problem, power minimization based on test data compression was presented in [19]. Test scheduling techniques for system integration attempt to reduce testing time by applying scan/BIST vectors to several cores simultaneously [14, 15]. Test scheduling is typically carried out under power constraints since multiple cores are tested in parallel.

3. *Tester frequency*: Reduction in power dissipation can be achieved by running the tester at a slower frequency. Although this method offers the simplest way to reduce power consumption, it leads to unacceptable testing times and is therefore impractical.

We note that structural methods for reducing test power in SOCs require modification to the embedded cores, e.g. via scan latch reordering [12], scan chain and scan cell redesign. This is usually not feasible for IP cores. ATPG techniques are also infeasible for IP cores since they require gate-level structural models. We therefore focus on test data compres-

[*]This research was supported in part by the National Science Foundation under grant number CCR-9875324.

DAC 2002, June 10-14, 2002, New Orleans, Louisiana, , USA.
Copyright 2002 ACM 1-58113-297-2/01/0006 ...$5.00.

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

Figure 1: The alternating run-length code.

sion for reducing test power, test data volume, and testing time simultaneously.

It was shown in [19] that test data volume and test power can be reduced simultaneously using Golomb coding. The key idea is to map the don't-cares in the test vectors to zero. This results in long runs of zeros that can be efficiently compressed using Golomb code [17]. The resulting fully-specified test set also reduces switching activity during scan shifting. Hence, significant reduction in scan power is accompanied with test data compression. However, the test power can be reduced further if we map the don't-cares to derive test sets that minimize switching activity. Unfortunately these test sets are not amenable for compression by run-length codes.

In this paper, we present a new class of codes, called alternating run-length codes, for test data compression. These codes are particularly effective for compressing test sets that lead to minimum switching activity . They also reduce testing time due to the reduction in the amount of test data that needs to be transported from the tester to the SOC.

2. ALTERNATING RUN-LENGTH CODE

We first review FDR coding and its application to test data compression. The FDR code is a data compression code that maps variable-length runs of 0s to variable-length codewords. The reader is referred to [18] for a detailed discussion and motivation for the FDR code.

It was shown in [18] that the FDR code is very efficient for compressing data that has few 1s and long runs of 0s. However, for data streams that are composed of both runs of 0s and runs of 1s, the FDR code is rather inefficient. In fact, in our initial experiments, the sizes of encoded test sets obtained for such test sets were larger than the sizes of uncompressed test sets. This provides the motivation to develop a code that can efficiently compress both runs of 0s and 1s.

Figure 1 illustrates the encoding procedure for the new alternating run-length code. The alternating run-length code is also a variable-to-variable-length code and consists of two parts—group prefix and tail. The prefix identifies the group in which the run-length lies and the tail identifies the member within the group. An additional parameter associated with this code is the alternating binary variable a. The encoding produced by the alternating run-length code for a given run-length depends on the value of a. If $a = 0$, the run-length is treated as a run of 0s. On the other hand, if $a = 1$, the run-length is treated as a run of 1s. Note that the values of a for the different runs are not added to the encoded data stream.

```
Input data stream:              000000111111000001 (18-bits)
FDR encoded data:               1100000000000000011010 (22-bits)
Alternating FDR encoded data:   11000010111010 (14-bits)
                                |  a= 0 |a =1|a= 0|
```

Figure 2: An example to compare FDR coding with alternating run-length coding.

Figure 2 shows the encoded data obtained using the two codes for a data stream composed of interleaved runs of 0s and 1s. We observe that the size of the FDR-encoded data set (22 bits) is larger than the size of the input data set (18 bits); hence the FDR code provides no compression for this case. On the other hand, the size of the alternating run-length-encoded data set (14 bits) is smaller than the size of the input data set. Therefore, we are able to achieve compression with the new code. We also note that $a = 0$ is used for compressing a runs of 0s, $a = 1$ is used for compressing a runs of 1s, and $a = 0$ is then used for compressing the next run of 0s. Hence, a is inverted after each run is encoded and it keeps alternating between 0 and 1 thereafter. In this paper, we assume a default initial value of $a = 0$, i.e., we assume that the input data stream starts with a run of 0s.

2.1 Power estimation

We use the weighted transitions metric (WTM) introduced in [13] to estimate the power consumption due to scan vectors. The WTM metric models the fact that the scan power for a given vector depends not only on the number of transitions in it but also on their relative positions. For example, consider a scan vector $v_1v_2v_3v_4v_5 = 01000$, where v_1 is first loaded into the scan chain. The 0-to-1 transition between v_1 and v_2 causes more switching activity in the scan chain than the 1-to-0 transition between v_2 and v_3.

WTM is also strongly correlated to the switching activity in the internal nodes of the core under test during scan operation. It was shown experimentally in [13] that scan vectors with higher weighted transition metric dissipate more power in the core under test. In addition, experimental results for an industrial ASIC [5] confirm that the transition power in the scan chains is the dominant contributor to test power.

Consider a scan chain of length l and a scan vector $t_j = t^{\star}_{j,1}t^{\star}_{j,2}\ldots t^{\star}_{j,l}$, with $t^{\star}_{j,1}$ scanned in before $t^{\star}_{j,2}$, and so on. As shown in [19], the *weighted transitions metric* for t_j, denoted WTM_j, is given by $WTM_j = \sum_{i=1}^{l-1}(l-i)\cdot(t^{\star}_{j,i} \oplus t^{\star}_{j,i+1})$. If the test set T_D contains n vectors $t_1, t_2, \ldots, t_n$ then the average scan in power P_{avg} and peak scan in power P_{peak} are estimated as $P_{avg} = \sum_{j=1}^{n} WTM_j/n$ and $P_{peak} = \max_{j\in\{1,2,\ldots,n\}}\{WTM_j\}$.

It was shown in [19] that Golomb coding can be used to simultaneously reduce the volume of test data, P_{avg} and P_{peak}. The don't-care bits were mapped to 0s and the resulting test data was compressed using the Golomb code. While this approach provides significant reductions in power consumption, and at the same time, decreases the test data volume considerably, it does not minimize either P_{avg} or P_{peak}.

Here we improve upon [19] by using the alternating run-length code. Table 1 shows a partially-specified scan vec-

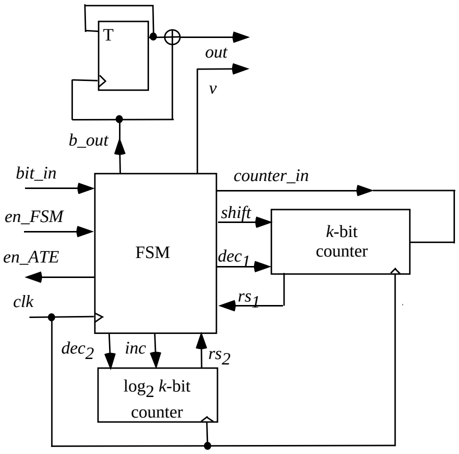

Figure 3: The decoder block diagram for the alternating run-length code.

tor $t_i = 01XXX10XXX01$ with scan chain length $l = 12$, where X denotes a don't-care bit. If the don't-cares are mapped to appropriate binary values to minimize the weighted transition metric, then a sequence $dXXXXd'$, $d \in \{0, 1\}$, must be mapped to $ddddd d'$. Similarly, a sequence $dXXXX$ must be mapped to $ddddd$. This ensures that the few unavoidable transitions occur "late" during scan in.

Table 1 shows the impact of don't-care mapping on the weighted transitions metric WTM_i and compression for a given test vector t_i. Since the FDR code has been shown to be more efficient than the Golomb code [18], we only consider the FDR code here. The WTM value is clearly higher if the don't-cares are always mapped to 0. However, FDR coding is much more effective in reducing test data volume if this strategy is used. On the other hand, while FDR coding is ineffective for the fully-specified test vector that minimizes WTM, the alternating run-length code provides the same compression as achieved with the FDR code with all don't-cares mapped to 0s.

2.2 Decompression architecture

An on-chip decoder decompresses the encoded test set T_E and produces T_D. Let r_{max} be the longest run of 0s in T_D and let $k = \lceil \log_2 r_{max} \rceil$. As discussed in [18], the FDR decoder can be efficiently implemented by a k-bit counter, a $\log_2 k$-bit counter and a finite-state machine (FSM), and it is independent of the precomputed test set and the circuit under test. The synthesized decode FSM circuit contains only 4 flip-flops and 34 combinational gates. The decoder for alternating run-length code can be implemented by making a small modification to the FDR decoder. The block diagram of the alternating run-length decoder is shown in Figure 3. An additional toggle flip-flop and an exclusive-OR gate are required to switch between $a = 0$ and $a = 1$.

3. TESTING TIME ANALYSIS

We now analyze the testing time when a single scan chain is fed by the alternating run-length 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 $f_{ATE} < f_{scan}$. Since the ATE and the scan chain operate at two different frequencies, the decoder also consists of two parts—one operating at f_{ATE} and the other operating at f_{scan} such that $f_{ATE} = f_{scan}/\alpha$, $\alpha > 1$. 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 [21]. The proposed TRP 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 f_{scan} can now be made much larger than f_{ATE}.

For the alternating run-length code, let $t(k, i)$ be the total time required to decompress a codeword that is the i^{th} member of the k^{th} group, and $t_{shift}(k, i)$ and $t_{decode}(k, i)$ be the time required to transfer the data from the ATE to the chip and to decode the codeword, respectively. An upper bound on $t(k, i)$ can be obtained by assuming that decoding begins after the complete codeword is transfered from the ATE. This implies that $t(k, i) \leq t_{shift}(k, i) + t_{decode}(k, i)$.

The prefix length and the tail length of the codeword belonging to the k^{th} group is each equal to k bits; see Figure 1. Since data is transfered from the ATE to the chip at the tester frequency, the time required to transfer any codeword of the k^{th} group is given by $t_{shift}(k, i) = 2k/f_{ATE}$.

For any codeword, the prefix is identical to the binary representation of the run-length corresponding to the group's first element. As shown in Figure 1, the number of 0s in the prefix of a codeword belonging to the k^{th} group is equal to $2^k - 2$. The decoder has to output $(2^k - 2)$ 0s before the tail decoding starts. The time $t_{prefix}(k)$ required to decompress the prefix of any codeword from the k^{th} group is therefore given by $t_{prefix}(k) = (2^k - 2)/f_{scan}$.

Similarly, the time $t_{tail}(k, i)$ required to decompress the tail of the i^{th} member of the k^{th} group is equal to the sum of time required to output $(i - 1)$ 0s and a single 1. Hence, $t_{tail}(k, i) = ((i - 1) + 1)/f_{scan} = i/f_{scan}$.

Therefore, the total decoding time $t_{decode}(k, i)$ is given by

$$\begin{aligned} t_{decode}(k, i) &= t_{prefix}(k) + t_{tail}(k, i) \\ &= (2^k - 2)/f_{scan} + i/f_{scan}. \end{aligned}$$

The total time needed to decompress the codeword is given by

$$\begin{aligned} t(k, i) &\leq t_{shift}(k, i) + t_{decode}(k, i) \\ &= 2k/f_{ATE} + (2^k - 2)/f_{scan} + i/f_{scan} \\ &= \frac{1}{f_{ATE}}(2k + \frac{2^k - 2 + i}{\alpha}) \qquad (1) \end{aligned}$$

where $f_{scan} = \alpha\ f_{ATE}$.

Let $q(k, 1), q(k, 2), q(k, 3), \ldots, q(k, 2^k)$ be the absolute frequencies of the members of the k^{th} group. Therefore, the decompression time $\tau(k)$ for the runs belonging to k^{th} group is given by

$$\begin{aligned} \tau(k) &= \frac{1}{f_{ATE}} \sum_{i=1}^{2^k} (2k + \frac{2^k - 2 + i}{\alpha}) q(k, i) \\ &= \frac{1}{f_{ATE}} (2k \sum_{i=1}^{2^k} q(k, i) + \frac{1}{\alpha} \sum_{i=1}^{2^k} (2^k - 2 + i) q(k, i)). \end{aligned}$$

Let us assume that k_{max} is the largest group. The test application time TAT_{SSC} for the entire test set with a single scan chain (SSC) is given by

$$TAT_{SSC} \leq \sum_{k=1}^{k_{max}} \tau(k) = \frac{|T_E| + \frac{1}{\alpha} \sum_{k=1}^{k_{max}} \sum_{i=1}^{2^k} (2^k - 2 + i) q(k, i)}{f_{ATE}}$$

Partially-specified scan vector (test cube)	Fully-specified vector (Minimum WTM)	Fully-specified vector (Don't-cares mapped to 0s)
$t_i = 01XXX10XXX01$ (12-bits)	011111000001 FDR code length: 14 bits Alternating FDR code length: 10 bits $WTM_i = 18$	010001000001 FDR code length: 10 bits $WTM_i = 25$

Table 1: Mapping of don't-cares in a test cube to binary values.

			FDR coding using T_D		Alternating FDR coding using T_D		
Circuit	**No. of bits in T_D**	**Size of Mintest test set (bits)**	**Comp-pression (percent)**	**No. of bits in T_E**	**Comp-pression (percent)**	**No. of bits in T_E**	**Improvement over Mintest (percent)**
s9234	39273	25935	43.59	22152	44.96	21612	16.66
s13207	165200	163100	81.30	30880	80.23	32648	79.98
s15850	76986	57434	66.22	26000	65.83	26306	54.19
s38417	164736	113152	43.26	93466	60.55	64976	42.57
s38584	199104	161040	60.91	77812	61.13	77372	51.95
Average	—	—	57.21	—	60.57	—	48.16

Table 2: Experimental results on test data compression using the FDR and the alternating run-length code.

where, $|T_E|$ is the size of the encoded test set. Next, to derive a lower bound on the testing time, suppose the tail bits are shifted in while the prefix is being decompressed. Since, the tail bits are now shifted in parallel while the prefix bits are decoded, a lower bound on decoding time using (1) is given by:

$$t(k,i) \geq k/f_{ATE} + (2^k - 2)/f_{scan} + i/f_{scan} = \frac{k + \frac{2^k-2+i}{\alpha}}{f_{ATE}}.$$

Therefore,

$$TAT_{SSC} \quad \geq \quad \frac{1}{f_{ATE}}(\frac{|T_E|}{2} + \frac{1}{\alpha}\sum_{k=1}^{k_{max}}\sum_{i=1}^{2^k}(2^k - 2 + i)q(k,i)).$$

We next compare the testing time using the proposed TRP scheme with that for an ATPG-compacted test set with p patterns and an external tester operating at frequency $f^{\star}_{ATE}$. 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 $pn/f^{\star}_{ATE}$. Experimental results presented in Section 4 show that the testing time is reduced considerably using the proposed method if $f^{\star}_{ATE} = f_{ATE}$.

4. EXPERIMENTAL RESULTS

In this section, we evaluate the effectiveness of alternating run-length coding for reducing test data volume, testing time and power consumption during scan testing. We carried out experiments for the larger ISCAS-89 benchmark circuits and a production circuit from IBM. The experiments were conducted on a Sun Ultra 10 workstation with a 333 MHz processor and 256 MB of memory.

Table 2 presents the experimental results for test sets obtained from the Mintest ATPG program [20]. We compare the compression obtained using the FDR code and the alternating run-length code. In order to compare with [19], we first assigned all don't-cares to 0s and compressed T_D using the FDR code. We then carefully mapped the don't-cares to minimize WTM and compressed the resulting T_D using the alternating run-length code. Table 2 shows the sizes of T_D, the size of the smallest encoded test set obtained after static compaction using Mintest, the size of compressed test set obtained with all don't-cares mapped to 0 ($|T_{E1}|$) and size of compressed test set obtained with an optimal mapping of don't-cares to minimize WTM ($|T_{E2}|$).

As is evident from Table 2, the alternating run-length code yields better compression than the FDR code for four out of the six benchmark circuits. This is particularly remarkable since, as we show later in this section, high compression with the alternating run-length code is accompanied with significant reduction in testing time and test power. In all cases, the size of the encoded test set is less than the smallest ATPG-compacted test sets known for these circuits. On average, the size of T_E is 48.16% less than that of the compacted test sets obtained using Mintest.

Circuit	f_{ATE} (MHz)	α	Lower bound on TAT_{SSC} (ms)	Upper bound on TAT_{SSC} (ms)	TAT_{SSC}^{ATPG} (ms)
s9234	20	4	0.589	0.881	1.037
		8	0.785	1.326	1.296
		16	0.663	1.203	1.296
s13207	20	4	2.881	3.697	8.155
		8	1.848	2.664	8.155
		16	1.332	2.148	8.155
s15850	20	4	1.619	2.277	2.871
		8	1.138	1.796	2.871
		16	0.898	1.555	2.871
s38417	20	4	3.683	5.308	5.657
		8	2.654	4.278	5.657
		16	2.139	3.763	5.657
s38584	20	4	4.423	6.357	8.052
		8	3.178	5.113	8.052
		16	2.556	4.490	8.052

Table 3: Comparison of testing time using the proposed TRP method with traditional scan-based testing.

Table 3 presents test application time for the proposed method using the alternating run-length codes and for traditional scan-based testing with $f^{\star}_{ATE} = f_{ATE}$. We note that in all the cases the upper bound on test application time using the proposed scheme is lower than that for scan-based external testing. For example, the test application time for s13207 with $\alpha = 8$, and $f^{\star}_{ATE} = f_{ATE} = 20$ MHz lies between 1.848 ms and 2.664 ms, which is lower than the time of 8.155 ms required for conventional scan testing using ATPG-compacted patterns derived using Mintest. Furthermore, let us assume that the desired testing time for s38417 is the average of lower and upper bounds i.e., 2.256 ms. In this case, an external tester operating at $f^{\star}_{ATE} = 72.29$ MHz will be required, as opposed to a 20 MHz tester with the proposed TRP scheme.

We next present results on the peak and average power consumption during the scan-in operation. These results show that test data compression can also lead to significant

	Uncompacted test sets with don't-cares mapped to 0s				Uncompacted test sets with don't-cares mapped to minimize *WTM*			
Circuit	Peak power P^F_{peak}	Peak power reduction (percent)	Average power P^F_{avg}	Average power reduction (percent)	Peak power P^A_{peak}	Peak power reduction (percent)	Average power P^A_{avg}	Average power reduction (percent)
s9234	12994	25.72	5692	61.09	12060	31.06	3466	76.30
s13207	101127	25.42	12416	89.82	97606	28.02	7703	93.68
s15850	81832	18.35	20742	77.18	63478	36.66	13381	85.27
s35932	172834	75.56	73080	87.47	125490	82.25	46032	92.11
s38417	505295	26.10	172665	71.31	404617	40.82	112198	81.35
s38584	531321	7.21	136634	74.50	479530	16.25	88298	83.52
Average	—	28.98	—	75.89	—	37.72	—	84.32

Table 4: Impact of the mapping of don't-cares to binary values on scan-in power consumption.

	Mintest test sets after static compaction		Uncompacted test sets with don't-cares mapped to minimize *WTM*			
Circuit	Peak power P^C_{peak}	Average power P^C_{avg}	Peak power P^G_{peak}	Peak power reduction (percent)	Average power P^G_{avg}	Average power reduction (percent)
s9234	16777	14555	16927	-0.89	12957	10.97
s13207	132129	116695	120992	8.42	97004	16.87
s15850	99647	88385	88445	11.24	67064	24.12
s35932	745282	390622	125298	83.18	50351	87.11
s38417	619929	553491	549950	11.28	311072	43.79
s38584	577365	521882	529315	8.32	437339	16.19
Average	—	—	—	20.25	—	33.17

Table 5: Experimental results on peak and average scan-out power consumption.

savings in power consumption. As discussed in Section 2, we estimate power using the weighted transitions metric. Let P^C_{peak} (P^C_{avg}) be the peak (average) power with compacted test sets obtained using Mintest, and let P^F_{peak} (P^F_{avg}) be the peak (average) power when the FDR code is applied to T_D after mapping the don't-cares to 0s. Similarly, let P^A_{peak} (P^A_{avg}) be the peak (average) power when the alternating run-length code is applied to T_D after mapping the don't-cares to minimize the weighted transitions metric. Table 4 compares the average and peak power consumption when FDR coding and alternating run-length coding are used.

Peak power and average power are significantly less if the alternating run-length code is used for test data compression and decompressed patterns are applied during testing. On average, the peak (average) power is 37.72% (84.32%) less in this case than for the Mintest test sets. (Note that the power values for the Mintest test sets are shown in Table 5.) Thus our results demonstrate that substantial reduction in test data volume and testing time are also accompanied by significant reduction in power consumption during scan testing. We next present results on the peak and average power consumption during the scan-out operation. Table 5 shows that the peak power and average power are significantly less in five out of six cases if alternating run-length coding is used for test data compression. The peak power during scan out operation was only slightly higher for the s9234 benchmark circuit. On average, the peak (average) power is 20.25% (33.17%) less than for the Mintest test sets. Thus our results demonstrate that the substantial reduction in test data volume is also accompanied by significant reduction in power consumption during scan testing. The reduction in scan-out power is an important added advantage since we do not directly target scan-out power in our compression scheme.

Table 6 presents the experimental results when the FDR code and the alternating run-length code are applied to scan vectors for a production circuit from IBM. This circuit contains 1.2 million gates, 32200 latches and a total of 30 scan chains. We were provided with four sets of scan vectors for this circuit. Each vector set consists of 32 patterns (a total of 1031072 bits of test data per vector set). We find that the compression obtained using the alternating run-length code is comparable to that obtained using the FDR code. The slight (1–2%) decrease in compression is offset by the significant savings in test power as shown next.

Table 7 presents the average and peak power values for the IBM circuit when the FDR code and the new alternating run-length code are applied to the scan vectors. We find that the alternating run-length code is extremely efficient for reducing test power. Compared to the FDR code, as much as 43.36% greater reduction is obtained in peak power and as much as 42.38% greater reduction is obtained in average power.

Finally we present results on test application time for the IBM circuit when the TRP scheme based on the new alternating run-length codes is applied to the scan vectors. We assume that the circuit consists of a single scan chain. The minimum (maximum) testing time for the four vectors using TRP and a tester with f_{ATE} = 50 MHz is 12.757 ms (15.205 ms). On the other hand, the test application time based on traditional scan-based testing using the same external tester is 61.863 ms.

5. CONCLUSIONS

We have shown that test resource partitioning (TRP) based on test data compression can be used to reduce SOC test data volume, testing time and test power simultaneously. The proposed TRP method is based on the use of a new code, which we call the alternating run-length code. Experimental results for the ISCAS-89 benchmark circuits and for

		FDR coding using T_D		Alternating run-length coding using T_D	
Scan vector set	No. of bits in T_D	Compression (percent)	No. of bits in T_E	Compression (percent)	No. of bits in T_E
1	1031073	95.08	50648	93.90	62862
2	1031073	94.71	54476	93.98	62030
3	1031073	95.02	51286	93.75	64354
4	1031073	95.63	45026	94.60	55596
Average	—	95.11	—	94.05	—

Table 6: Results on test data compression for an IBM circuit.

	Uncompacted test sets with don't-cares mapped to 0s		Uncompacted test sets with don't-cares mapped to minimize *WTM*			
Scan vector set	Peak power P^F_{peak}	Average power P^F_{avg}	Peak power P^A_{peak}	Peak power reduction (percent)	Average power P^A_{avg}	Average power reduction (percent)
1	9825669	3587758	6113059	37.78	2101363	41.42
2	10964254	3892459	6255619	42.94	2151415	44.72
3	8996102	3611662	5171926	42.50	2187382	39.43
4	7712574	3161122	3836810	50.25	1771358	43.96
Average	—	—	—	43.36	—	42.38

Table 7: Power reduction associated with the compression of IBM test data.

an IBM production circuit 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 transfered from the ATE, but it also reduces test power and testing time, and it allows the use of slower ATEs.

Acknowledgment

We thank Brion Keller of IBM Corporation for providing scan vectors for the production circuit.

6. REFERENCES

[1] Y. Zorian, "A distributed BIST control scheme for complex VLSI devices", *Proc. VLSI Test Symp.*, pp. 4-9, 1993.

[2] I Hamzaoglu and J. H. Patel, "Reducing test application time for full scan embedded cores", *Proc. Int. Symp. Fault-Tolerant Comp.*, pp. 260-267, 1999.

[3] F. F. Hsu, K. M. Butler and J. H. Patel, "A case study on the implementation of Illinois scan architecture", *Proc. Int. Test Conf.*, pp. 538-547, 2001.

[4] S. Hellebrand, J. Rajski, S. Tarnick, S. Venkataraman and B. Courtois, "Built-in test for circuits with scan based on reseeding of multiple-polynomial linear feedback shift registers", *IEEE Trans. Computers*, vol. 44, pp. 223-233, February 1995.

[5] J. Saxena, K. Butler and L. Whetsel, "An analysis of power reduction techniques in scan testing", *Proc. Int. Test Conf.*, pp. 670-677, 2001.

[6] P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch and H. -J. Wunderlich, "A modified clock scheme for a low power BIST test pattern generator", *Proc. VLSI Test Symp.*, pp. 306-311, 2001.

[7] S. Gerstendörfer and H.-J. Wunderlich, "Minimized power consumption for scan-based BIST", *Proc. Int. Test Conf.*, pp.77-84, 1999.

[8] P. Girard, L. Guiller, C. Landrault and S. Pravossoudovitch, "A test vector inhibiting technique for low energy BIST design", *Proc. VLSI Test Symp.*, pp. 407-412, 1999.

[9] F. Corno, M. Rebaudengo and M. S. Reorda, "Low power BIST via non-linear hybrid cellular automata", *Proc. VLSI Test Symp.*, pp. 29-34, 2000.

[10] L. Xu, Y. Sun and H. Chen, "Scan solution for testing power and testing time", *Proc. Int. Test Conf.*, pp. 652-659, 2001.

[11] S. Wang and S. K. Gupta, "ATPG for heat dissipation minimization during scan testing", *Proc. DAC*, pp. 614-619, 1997.

[12] V. Dabholkar, S. Chakravarty, I. Pomeranz and S. M. Reddy, "Techniques for minimizing power dissipation in scan and combinational circuits during test application", *IEEE Trans. CAD*, vol. 17, pp. 1325-1333, Dec. 1998.

[13] R. Sankaralingam, R. R. Oruganti and N. A. Touba, "Static compaction techniques to control scan vector power dissipation", *Proc. VLSI Test Symp.*, pp. 35-40, 2000.

[14] V. Iyengar and K. Chakrabarty, "Precedence-based, preemptive, and power-constrained test scheduling for system-on-a-chip", *Proc. VLSI Test Symp.*, pp. 368-374, 2001.

[15] K. Chakrabarty, "Test scheduling for core-based systems using mixed-integer linear programming", *IEEE Trans. CAD*, vol. 19, pp. 1163-1174, October 2000.

[16] A. Jas and N. A. Touba, "Test vector decompression via cyclical scan chains and its application to testing core-based design", *Proc. Int. Test Conf.*, pp. 458-464, 1998.

[17] A. Chandra and K. Chakrabarty, "System-on-a-chip test data compression and decompression architectures based on Golomb codes", *IEEE Trans. CAD*, vol. 20, pp. 355-368, March 2001.

[18] A. Chandra and K. Chakrabarty, "Frequency-directed run-length (FDR) codes with application to system-on-a-chip test data compression", *Proc. VLSI Test Symp.*, pp. 42-47, 2001.

[19] A. Chandra and K. Chakrabarty, "Combining low-power scan testing and test data compression for system-on-a-chip", *Proc. DAC*, pp. 166 -169, 2001.

[20] I. Hamzaoglu and J. H. Patel, "Test set compaction algorithms for combinational circuits", *Proc. Int. Conf. CAD*, pp. 283-289, 1998.

[21] D. Heidel, 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", *Proc. VLSI Test Symp.*, pp. 234–238, 1998.