Power-Oriented Test Scheduling for SOCs

Wang-Dauh Tseng

Department of Computer Science & Engineering Yuan Ze University Chung-li, Taiwan, 32026, ROC

Summary

The purpose of this paper is to integrate the management of power consumption to augment the parallelism of the testing activities and to reduce the testing time of SOCs. This is achieved in three parts. First, a specific scan cell is employed to reduce the switching activities during test scan stages. Secondly, a test vector reordering approach is used to reduce the test power. Finally, a test sequence expansion technique is employed to avoid the conflict of the high-power part of cores in the same session, thus as many cores can be tested concurrently. Experimental results show that the proposed approach can reduce the SOC testing time significantly.

Keywords:

SOC, test vector reordering, power consumption, test scheduling.

Introduction

In order to shorten SOC testing time, it is efficient to put as many cores to test concurrently. However, limitations, such as power consumption and resource conflicts etc., inhibit parallel test. In general, a circuit consumes more power in test mode than in normal mode. The power consumed in test mode can be as high as twice than in the normal mode [1]. To reduce the testing time, the testing of cores should be performed in parallel to the greatest extent possible, but it will lead to higher power dissipation that is not tolerated by the system. Power dissipation in circuits is in proportion to the switching activities. During test mode, filling in the scan chain with test data requires shifting the bits of the test data one by one, thus creates switching activities in the flip-flops. The rippling effect in a scan chain reflects into the circuit, resulting in a large number of unnecessary transitions at the circuit lines. Preventing unnecessary rippling in the scan chain during the shift of the test data can reduce power dissipation. This can be achieved by careful selection of the scan cell design.

The cores are always tested by applying the test vectors. The switching activity of the core under test will reach a maximal amount when the patterns are applied [2]. The switching activity differs depending on the consecutive test pairs. If the succeeding test vector is similar to the proceeding one, the transitions between these two vectors are reduced. Several test vector reordering approaches have been proposed to increase the correlation between successive test vectors in the test sequence.

2. Proposed Approach

The purpose of this paper is to augment the parallelism of the testing activities and reduce the testing time of SOCs by integrating a power management strategy. This is achieved in three parts:

(1) Scan cell selection: A specific scan cell is employed to prevent the shifting of test vectors on scan chains from affecting the internal circuits of the core under test, and it can leave out the power consumed in the period of scan-in and scan-out.

(2) Test vector reordering: We improve the test vector reordering algorithm proposed in [3] to reflect the power cost between each pair of continuous test vectors.

(3) Power oriented test scheduling: After applying above two steps, power consumed by cores can be reduced and the shape of power profile becomes regularly. Using the power profile features, we propose a test scheduling algorithm to increase the test parallelism of cores.

2.1 Scan Cell Design

The concept of full scan design is that the normal parallel-load registers in the core are seen as the scan cells. Consider a scan chain with k scan cells. When test mode, the test vector is scanned into the scan chain completely during k clock times. During the k-th clock time, the test pattern is applied to the core. During the (k + 1)st clock time, the test response is loaded into the scan chain. During the next k clock times, the test response and the succeeding test vector are scanned simultaneously. During scan the rippling effect in a scan chain reflects into the circuit, resulting in a large number of unnecessary transitions at the circuit lines. To avoid such situation, we adopted the multiplex data shift register latch (MD-SRL) [4] as the scan cell. MD-SRL scan cell is a two-port shift register latch consisting of latches L1 and L2 along with a multiplexed input. This cell is not considered a flip-flop, since each latch has its own clock. By controlling the clock of L2, unnecessary transitions at the circuit lines during scan can be avoided. In addition, it causes the power profile regularity. Consider, for example, the scan chain with four scan cells. Fig. 1(a) shows a possible power profile without eliminating unnecessary transitions during scan. Contrarily, Fig. 1(b) shows the power profile by adopting MD-SRL. Trivially, the power will reach a local maximum when the circuit is in update mode since the switch activity of internal circuits occurs significantly at this clock cycle [2]. The cycle of power for the circuit under test to reach a local maximum is fixed and can be calculated. The cycle of the high power parts can be calculated as follows:

 $Ti(n) = n(L+1) \quad i, n = 1, 2, 3, \dots (1)$ Where i is the core number, n is the vector number, and L is the length of the scan chain of core i.



Fig. 1. Power profiles for (a) without (b) with preventing unnecessary transitions.

2.2 Test Vector Reordering

Test vector reordering technique is a way to decrease the power consumption during test. We are to improve the test vector reordering algorithm which is proposed in [3]. We use the input transition graph (ITG) to model the test vector reordering problem. For example, given a test sequence with M test vectors, the ITG = (Ψ, E) can be constructed as a complete directed graph with $|\Psi| =$ M nodes and |E| = M(M-1) edges. Each node Ni $\in \Psi$ include two parts: vi a test vector in the test sequence and ri the test response obtained after applying vi to the CUT. The edge $(Ni, Nj) \in E$ represents a transition at the primary inputs from Ni to Nj. The ITG edges are labeled with an estimation of the power dissipated in the circuit by the corresponding transition. The edge weights are computed according to the test-per-scan scheme. Edge (Ni, Nj) in ITG is weighted with the power consumed by the simultaneous scan-out of ri and scan-in of vj. It was shown in [5] that the weighted transition count (WTC) is very well correlated with the real power dissipation. The WTC values corresponding to vi scan-in and ri scan-out,

respectively, are given by

$$TC_{scanin}(N_{i}) = \sum_{j=1}^{m-1} (v_{i}(j) \oplus v_{i}(j+1))(m-j)$$
(2)
$$TC_{scanout}(N_{i}) = \sum_{j=1}^{m-1} (r_{i}(j) \oplus r_{i}(j+1))j$$
(3)

where vi(j) represents the j-th bit from vector vi. Finally, ITG edge weights for test-per-scan are computed by

$$Weight(v_i, v_j) = TC_{scanout}(N_i) + TC_{scanin}(N_j)$$
(4)

From above equations, only the switching activity between the bits in the test vector or in the test response is considered. It does not consider the switching activity between the last bit of current test response and the first bit of the next test vector. For example, suppose that there is a scan chain with four scan cells. After applying the first test vector, the test response obtained in the scan chain is 1001. When scan out the test result from the scan chain, the next test vector say, 1000, is scanning in simultaneously. The direction of the scan is provided from left to right. 4 transitions are produced due to the last bit of current test response and the first bit of the next test vector are different. Therefore, equation (4) should be modified by adding the following power factor.

$$TC_{outin}(r_i, v_j) = (r_i(m) \oplus v_j(1))m$$
⁽⁵⁾

In Equation (5), m is the length of scan chain, ri(m) is the last bit of test result of the preceding vector in the scan chain, and vj(1) is the first bit of the succeeding vector. ITG edge weights for test-per-scan can be modified as the following.

Weight(vi, vj) =

$$TC_{scanout}(N_i) + TC_{scanin}(N_j) + TC_{outin}(r_i, v_j)$$
 (6)

Having computed the ITG edge weighted, the test sequence reordering problem is reduced to the problem of finding in ITG a low cost Hamiltonian tour.

The modified WTC model in equation (6) makes the estimated power consumed during scan stage more precise. After applying the test vector reordering process, the power dissipation is decreased and the test concurrency can be increased.

2.3 Power Oriented Test Scheduling

By proper selecting the scan cell design, the switching activities of internal logic during scan stage can be restrained and the power profile of the core will become regularity as shown in Fig. 1(b). By following the two local peak powers model, the power profile in Fig. 1(a) can be simplified as two parts, the higher and the lower power part, as shown in Fig. 2. The purpose of the power management test scheduling algorithm is to avoid the conflict of peak power among cores during test. Since the power consumption of the SOC under test in any time instance is the sum of power consumed by the cores which are tested concurrently. If the peak power of each core can be staggered, more cores can be tested



Fig. 2. Two local peak power model of Fig. 1(b).

concurrently under the fixed power constraint, thus the SOC total testing time can be reduced. Like other scheduling algorithm, constraints, such as test resource conflict, should be taken into account. Here we employ the algorithm proposed in [6] as a preprocessor of the proposed scheduling algorithm to obtain the test compatibility graph (TCG), In TCG, a test compatibility clique (TCC) represents that all nodes in the same clique can be executed concurrently since they are time compatible. In other words, there are no resource conflicts for all nodes in the same clique. The cores are arranged according to the descending order of the test length. We use equation (1) to calculate the cycle of the high power parts of each core. After that, the conflict points of high power part among cores in the same clique can be calculated by using the following formulas.

$$T_{i} = \{n(L_{i}+1) \mid n \in N\}, T_{i} \neq \phi, i \in A$$

$$\forall i, j \in A, i \neq j,$$

$$T_{ij} = T_{i} \cap T_{j} = \{x \mid x \in T_{i} \text{ and } x \in T_{j}\}$$
(8)

Where A is the set of cores under test, n is the vector number, Li is the length of the scan chain of core i, and Ti is the set of clock cycles which specifies the high power part of core i. According to the conflict point information among cores in the same clique, we duplicate each test vector which causes conflict and insert the duplicated test vector behind it. This process is called test sequence expansion. The purpose of test sequence expansion is to eliminate the peak power of the core under test at the conflict points. Test sequence expansion can not reduce the power consumed during scan. But by applying the same test vector twice, the test expansion process can eliminate the transition of internal logic in update stage and thus the high power part can be removed. Therefore, by the test sequence expansion process the power dissipation only occurs during scan mode, and the high power parts never overlap during update mode. Notably, in order to preserve the total test application time, only the test sequences which do not influence the test session length are expand.

In the following we use an example to illustrate the proposed method. Consider a SOC with three cores (C1, C2, and C3) which are in the same TCC. The information related to the cores are shown in the Table 1 which contains the scan chain length, the peak power value of the low power part, the peak power value of the

high power part, the number of test vectors, the test length, and the number of additional test vector for each core.

Table 1. The specification for a sample SOC with three cores.							
Core	Scan chain length	Plo	Phi	No. of test vectors	Test length	No. of added vectors	
C1	6	2	4	6	62	2	
C2	4	2	6	19	99	0	
C3	5	1	4	10	77	2	

Table 1. The specification for a sample SOC with three cores.

Fig. 3 shows that the power profile of each core after applying the test sequence expansion. Besides, the cores (C1, C2, C3) sorted in descending order of their test length is (C2, C3, C1). Fig. 4 shows the power profile of the summation of the cores (C1, C2, C3). Where PC = 10 is the power limit of the SOC under test. In Fig. 4, we can observe that the maximum value of the resulting power profile is 9, which is scheduled in the same test session, which length is 99 clock cycles. While using the algorithm proposed in [6], two test sessions and total 161 clock cycles are required to cover all tests in the clique.



Fig. 3. The power profile for each core in Table 1.



Fig. 5. The power profile by using the approach in [6].

Fig. 5 shows the power profile and total testing time by using the algorithm in [6]. It should be noted that if the length of scan chain is long enough, the number of the test sequence expansion required is very small. The proposed test scheduling algorithm which is modified from paper [6] is shown in Fig. 6.

Algorithm Power-oriented test scheduling

Begin

Obtain the test compatibility graph (TCG) from the resource graph;
Obtain all cliques of the TCG;
For each clique {
Arrange the tests according to the descending order of the Phi;
derive the power compatible set (PCS) of tests;
}
For each PCS {
Obtain the power compatible list (PCL) by arranging the
tests according to the descending order of the test length;
Generate recursively all the derived PCLs (DPCLs);
}
Obtain the reduced DPCL set (RDPCL);
Apply weighted covering table procedure to the RDPCL set;
Obtain the minimum cost cover;
Apply test sequence expansion procedure to the minimum
cost cover set:

End (Algorithm)

Fig. 6. Power-oriented test scheduling algorithm.

3. Experimental Results

The proposed algorithm is implemented in C++ and executed on SunOS 5.7. The benchmarks SOC d695 and SOC g1023 are used as the sample circuits to compare the testing time between the proposed approach and the approach in [6]. SOC d695 consists of two ISCAS'85 and eight ISCAS'89 benchmark circuits. SOC g1023 consists of two ISCAS'85 and twelve ISCAS'89 benchmark circuits. The information related to the SOC d695 and g1023 are shown in Table 3 and Table 4, respectively, which contain the scan chain length, the peak power value of the high power part, the number of test vectors, and the test length for each core. The experimental results for SOC d695 and g1023 are shown in Table 5 and Table 6, respectively, where Pmax represents the maximum power limit. We use the globe peak power model to approximate the power consumed in the original PCTS algorithm of [6] and the resulting test application time under the power limit is reported in column 2.

The resulting test application time of the proposed method are reported in column3. Suppose that Plo = N * Phi, where N < 1 is a constant to adjust the proportion between low power and high power part. The constant varies from 0.1 to 0.5. For example, in Table 5, if the maximum power dissipation limit is 1200 mW, then the testing time for the method in [6] is 26153 clock cycles. The testing time for the proposed method is only 13817 clock cycles when the Plo is 10% of Phi. The experimental results show that while using the proposed method, test application time can be reduced with up to 47%.

Core	Scan chain	Phi	No. of	Test
core	length	1	test vectors	length
C1	0	660	12	12
C2	0	602	73	73
C3	32	823	75	2507
C4	54	275	105	5829
C5	45	690	110	5105
C6	41	354	234	9869
C7	34	560	95	3359
C8	46	753	97	4605
C9	54	641	12	714
C10	55	1144	68	3863

Table 4. SOC g1023 specifications.

Table 3. SOC d695 specifications.

	Core	Scan chain	Dha	No. of	Test
	Jore	length	PIII	test vectors	length
	C1	43	338	134	5939
	C2	84	502	74	6374
	C3	53	547	57	3131
	C4	54	609	268	14794
	C5	32	482	51	1715
	C6	47	496	36	1775
	C7	47	717	34	1679
	C8	52	606	31	1695
	C9	64	507	68	4484
(C10	13	814	29	419
(C11	9	945	15	159
(C12	13	1059	16	237
(C13	0	822	512	512
	C14	0	585	1024	1024

Dmax	Approach in [6]	Propo	osed ap	Test time	
(mW)		Plo-N*Phi		Clocks	reduction
	(clocks)	1 10-1			(%)
1200	26153	N =	0.1	13817	47.17
			0.2	16324	37.58
			0.3	17176	34.32
			0.4	19124	26.88
			0.5	20929	19.97

Table 5. Experimental results for SOC d695.

Table 6	Experimental	results for	r SOC	σ1023
Table 0.	Experimental	results to	ISUC	21025.

Table 0. Experimental results for SOC g1025.							
Dmox	Approach	Proposed approach			Test time		
(mW)	in [6] (clocks)	Plo=N*Phi		Clocks	reduction (%)		
			0.1	16567	41.02		
1200	28089	N =	0.2	18169	35.32		
			0.3	18501	34.13		
			0.4	22134	21.20		
			0.5	24015	14.50		

References

- Y. Zorian, "A Distributed BIST Control Scheme for Complex VLSI Devices," *Proceedings of IEEE VLSI Test Symposium*, Apr. 1993, pp.4-9.
- [2] K-J. Lee, T-C. Huang and J-J. Chen, "Peak-Power Reduction for Multiple-Scan Circuits during Test Application," *IEEE Asian Test Symposium*, pp. 453-458, Dec. 2000.
- [3] Paul M. Rosinger, Bashir M. Al-Hashimi, and Nicola Nicolici, "Power Profile Manipulation: A New Approach for Reducing Test Application Time under Power Constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Volume 21 Issue: 10, Oct. 2002, pp.1217 –1225.

- [4] M. Abramovici, M. A. Breuer and A. D. Friedman. Digital Systems Testing and Testable Design, ISBN 0-7167-8179-4.
- [5] R. Sankaralingam, R. R. Oruganti and N. A. Touba, "Static compaction techniques to control scan vector power dissipation," *Proceedings of IEEE VLSI Test Symposium*, 2000, pp.35–40.
- [6] R. M. Chou, K. K. Saluja, and V. D. Agrawal, "Scheduling tests for VLSI systems under power constraints," *IEEE Transactions on VLSI System*, vol. 5, pp.175–184, Jun. 1997.
- [7] R. Sankaralingam and N. A. Touba, "Reducing Test Power Dissipation During Test Using Programmable Scan Chain Disable," Proceedings of IEEE Electronic Design, Test and Applications, 2002, pp.159 –163.
- [8] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault and S. Pravossoudovitch, "A Gated Clock Scheme for Low Power Scan Testing of Logic ICs or Embedded Cores," IEEE Asian Test Symposium, Nov. 2001.
- [9] P. Girard, "Survey of low-power testing of VLSI circuits," IEEE Design & Test of Computers, Volume: 19 Issue: 3, May/Jun. 2002, pp.80 –90.



Wang-Dauh Tseng received the B.S. degree in computer science from Soochow University, Taiwan, and the M.S. and Ph.D. degrees in computer and information science from National Chiao Tung University, Taiwan. He is currently an Assistant Professor in the Department of Computer Science and Engineering, Yuan Ze University, Taiwan. His current research interests

include fault-tolerant computing, VLSI design and testing