Using a Single Input to Support Multiple Scan Chains *

Kuen-Jong Lee Jih-Jeen Chen Cheng-Hua Huang

Dept. of E.E, Nat'l Cheng-Kung U.
Tainan, Taiwan 70101, R.O.C.
E-mail : kjlee@eembox.ee.ncku.edu.tw

Abstract

Single scan chain architectures suffer from long test application time, while multiple scan chain architectures require large pin overhead and are not supported by Boundary Scan. In this paper, we present a novel method to allow a single input line to support multiple scan chains. By appropriately connecting the inputs of all circuits under test during ATPG process such that the generated test patterns can be broadcast to all scan chains when actual testing is executed, we show that 177 and 280 test patterns are enough to detect all detectable faults in all 10 ISCAS'85 combinational circuits and 10 largest ISCAS'89 sequential circuits, respectively.

Index Terms– design for testability, test generation, scan based design, Boundary scan (IEEE 1149.1) and test compaction.

1 Introduction

Scan based design is a structural design for test (DFT) technique that has been widely accepted in industry [1]. The basic idea of this method is to convert all or part of the internal registers of the circuit under test (CUT) into scan registers such that the controllability and observability of the CUT can be enhanced and the test generation complexity can be greatly reduced. However, it is well-known that the test application time of a scan system is proportional to the length of the scan chain. In a modern VLSI circuit, the number of internal registers can be in the range of thousands or even higher. Hence high product test cost may become a major concern in determining whether a scan based design should be used or not.

Multiple scan chains techniques have been developed to alleviate the long test application time problem [2][3][4]. By dividing a single serial scan chain into a number of shorter scan chains, test patterns and test results can then be shifted in/out of all scan chains in parallel to reduce the test application time. This method, however, will require a much higher number of extra I/O pins if a pair of I/O pins is used for each scan chain. One may use a multiplexer for each I/O such that the scan pin can be used for test and normal operations during different functional modes. This, however, will result in extra area overhead and may introduce further performance degradation that already exists in a scan based system.

Moreover, currently boundary scan has become a standard for board level testing [5]. Unfortunately the boundary scan architecture allows only one pin for test data input and another one for data output, hence cannot support multiple scan chains efficiently. One may use a de-multiplexer to distribute the test patterns from the single input line to multiple scan chains. Clearly this will give up the most important advantage of the multiple scan chain technique, i.e., reducing test application time via parallel loading of test patterns.

In this paper, we propose a novel method to allow one single data input to support multiple scan chains. The basic idea is to consider all the circuits driven by all scan chains as a single circuit when executing the automatic test pattern generation (ATPG) process such that the generated test patterns are effective for all circuits. With this method, we show that the total number of test patterns to test all circuits can be drastically reduced. Experimental results shows that 177 test patterns are enough to detect all detectable faults in the ten ISCAS'85 combinational circuits, while 280 are enough for the 10 largest ISCAS'89 scan-based sequential circuits. The reduction of the number of test patterns of course greatly reduce the required test application time.

2 Basic Concepts

It is well known that when generating a pattern for a specific fault in a CUT using a PODEM-like algorithm [6], usually only a subset of the primary inputs need to be specified. The *don't care* bits appear in the pattern can be further assigned some specific values to detect more faults. This "test compaction" concept has been used in some test generation algo-

* This work was supported in part by the National Science Council of R.O.C. under contract NSC-87-2215-E-006-011 and NSC-86-2262-E-009-010R.

rithms and it is shown that both test generation time and test pattern size can be reduced [7][8].

Test compaction can also be done after a set of test patterns has been generated. The basic idea here is to explore the compatibility among the generated test patterns and try to replace them with a new set of test patterns that has smaller size but still covers all faults that are detected by the original test set [9]. Previous test compaction techniques, however, only focus on generating a small test set for a single circuit. In this paper, we deal with multiple circuits and find that significant reduction of test set size can be achieved by considering not only the compatibility of test patterns, but also the property of an ATPG process.

We use two circuits CUT(1) and CUT(2) to illustrate the basic concept. Assume these two circuits have their own test sets T1=< t_{11}, t_{12}, $\cdots$, t_{1k} > and T2=< t_{21}, t_{22}, $\cdots$, t_{2l} >, respectively. In the beginning of an ATPG process, usually random patterns are used until a specified fault coverage, say 85%, is reached. If the same random pattern generation process is used when generating T1 and T2, then it is likely that $t_{11} = t_{21}$, $t_{12} = t_{22}$, $\cdots$, $t_{1i} = t_{2i}$ up to some i (For simplicity, assume the two circuits have the same number of inputs). Since most faults have been detected by the first i patterns, each of the remaining test patterns generated by deterministic ATPG may be needed only for a small number of faults. Therefore it is likely that these patterns may have many $don't\ \ care$ bits. For example, when generating $t_{1(i+1)}$, many don't care bits may still exist when no more faults in CUT(1) can be detected. If we use the test pattern with bits assigned so far for faults in CUT(2), then in addition to those faults that can be detected by the current pattern, we can further assign specific values to the don't care bits in the pattern to detect more faults. We can expect that if CUT(1) and CUT(2) are independent, then the don't care bits that cannot be further assigned to detect more faults in CUT(1) now may be assigned some values to detect more faults in CUT(2).

With the concept presented above, we know that either random patterns or deterministic patterns can be shared by both CUT(1) and CUT(2). This gives us a totally new method to deal with the multiple scan chain problem as explained below. If we take into account the requirement of all circuits driven by all scan chains simultaneously when generating tests, then the same test patterns is likely to be equally effective for all circuits. Therefore when actual test application procedure is carried out, we can simply "broadcast" the same patterns to all circuits. This allows us to use one single data line to provide test patterns to all circuits.

3 Virtual Circuits for ATPG

To implement the above concept, we have to recompose the circuits under test before the ATPG process. We shall call the recomposed circuit as the "virtual circuit". It should be pointed out that physically the virtual circuit does not exist. It is just for the ATPG process to generate common test vectors that are effective for all circuits. The actual hardware configuration will be presented in the next section.

We first show how to construct a virtual circuit from two CUTs, CUT(1) and CUT(2). Assume the numbers of primary inputs of CUT(1) and CUT(2) are M and N, respectively. Without loss of generality, assume $M \geq N$. We select N pins from the M pins of CUT(1) and connect them to the N pins of CUT(2). The connection is 1-1 mapping, i.e., different pins in CUT(2) are connected to different pins in CUT(1). There exist $\frac{M!}{(M-N)!}$ different virtual circuits based on CUT(1) and CUT(2). Figure 1 shows two possible connections of two CUTs with 4 and 3 inputs respectively. In Figure 2(a), the i th pins of CUT(1) and CUT(2) are connected, i =1, 2, 3, while in Figure 2(b) the connection is more random.

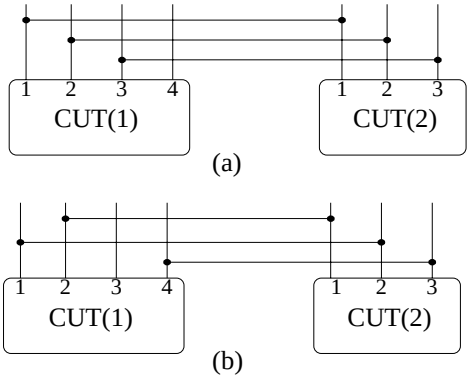

Figure 1: Two virtual circuits.

Similar construction can be extended to more circuits. Let CUT(1), CUT(2), $\cdots$, CUT(k) be k circuits with N_1, N_2, $\cdots$, N_k inputs, respectively, where $N_1 = \max_{i=1,\cdots,k} N_i$. Then we can connect the N_2 pins of CUT(2) to some N_2 pins of CUT(1), the N_3 pins of CUT(3) to some N_3 pins of CUT(1), and so on. There are totally $\frac{N_1!}{(N_1-N_2)!} \cdot \frac{N_1!}{(N_1-N_3)!} \cdots \frac{N_1!}{(N_1-N_k)!}$ different connections.

After a virtual circuit is constructed, we then apply the ATPG process to the whole circuit. Since only the inputs of all circuits are connected together, the test patterns generated can detect all detectable faults in all circuits.

Now the question is how to select a virtual circuit such that the number of generated test patterns is minimum. This is clearly an NP-complete problem because even the test compaction problem for a single circuit is NP-complete [10]. Therefore heuristic methods must be used. In this paper, we shall use two simple methods for the selection. We shall show that even such simple selection methods will result in significant reduction in test application time.

The first connection method is shown in Figure 2 which simply connects the N_i inputs, i =2,$\cdots$,k of each CUT(i) circuit to the first N_i pins of N_1 such that all the first pins of each circuit are connected together, all the second pins of each circuit are connected together, and so on. With this configuration one can expect that the "burden" of the pins with small indices will be greater than that for the pins with large indices because the first N_{min} bits of the generated test pattern bits must be applied to all circuits, while those for the last $N_1 - N_{max}$ bits are only applied to CUT(1), where $N_{min} = \min_{i=2,\cdots,k} N_i$ and $N_{max} = \max_{i=2,\cdots,k} N_i$.

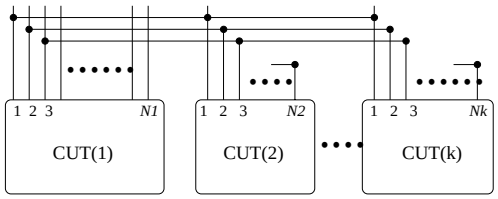

Figure 2: The first connection method.

The second selection method tries to "evenly distribute" the connection among the N_1 pins. We connect the N_2 pins of CUT(2) to the first N_2 pins of CUT(1). Then, we search the remaining circuits to see whether any one can fit the last $N_1 - N_2$ pins of CUT(1). If one exists, say CUT(i), then we connect the N_i pin of CUT(i) to the $N_2+1, \cdots, N_2+N_i$ pins of CUT(1). This process continues until no circuits can fit the remaining pins of CUT(1). Then for the remaining circuits, we resume the connection from the first pin of CUT(1). Figure 3 shows an example in which the inputs of 8 CUTs are configurated into 4 "scan chain" like structures.

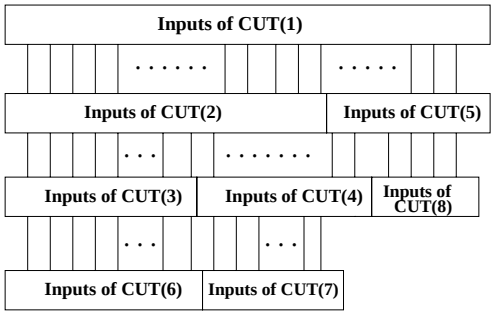

Figure 3: The second connection method.

4 Hardware Configuration

As mentioned before, the virtual circuit is just for the ATPG process. Now we describe the hardware configuration. Since all circuits will receive the same patterns, we can use one single line to broadcast test patterns to all circuits. But the configuration of the scan chains will be dependent on the selection of virtual circuits. In Figure 4, a general scan configuration based on the first method presented in the previous section is given.

In this figure all circuits under test will receive the same test patterns through Scan Input. For each pattern, after N_1 shifts, the entire pattern will appear in the scan chain of CUT(1), or SC_1, and the first N_i bits of the pattern will appear in the scan chain of CUT(i), or SC_i for $i = 2, \cdots, k$. Hence each CUT will receive its required pattern. The pattern is then applied to each circuit and the results are compressed by a multiple input signature register (MISR). With this configuration, we achieve the goal of using one single input line to support multiple scan chains. It

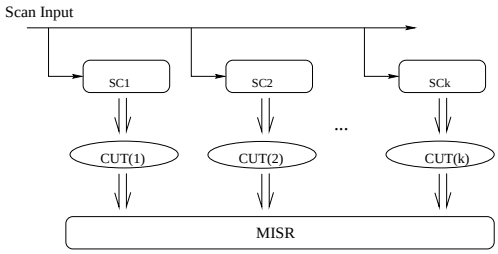

Figure 4: The scan configuration of first method.

should be pointed out that there may be some variations on the result compression scheme. For example, one may use a number of smaller MISRs instead of a single large MISR. In the case where the outputs of some circuits are on the scan chains (e.g., the pipeline structure), we may use a test structure similar to that used in the STUMPS structure [11] as shown in Figure 5.

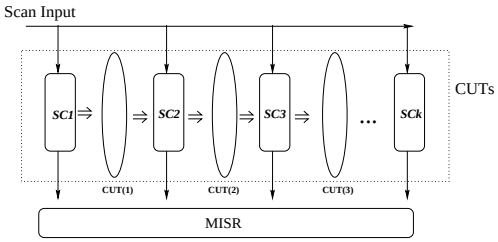

Figure 5: Broadcasting test patterns based on the STUMPS structure.

Figure 6 shows the scan chain architecture corresponding to the virtual circuit given in Figure 3 using our second virtual circuit selection method. For brevity, only the scan chains are shown. Note that in the virtual circuit, the first pin of CUT(5) is connected to the $N_2 + 1$ pin of CUT(1), while in the actual implementation, the input to the first scan cell of SC_5 is connected to the output of the last scan cell of SC_2.

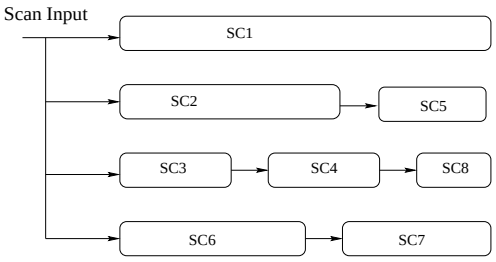

Figure 6: The scan architecture of the second virtual circuit selection method.

5 Experimental Results & Discussion

We use the 10 ISCAS'85 combinational circuits and 10 largest circuits of ISCAS'89 (S1488 is not included because it is the same as S1494) in our experiments. A commercial ATPG tool is used to generate test patterns. The fault simulation tool from Virginia

Table 1: Individual test generation results of ISCAS'85 circuits.

Circuits	# PI/PO	# Faults	# Gates	# RF	# TP	TG Time (sec)	FC (%)	TE (%)
C432	36/7	524	160	4	43	2.4	99.24	100
C499	41/32	758	202	8	54	2.5	98.94	100
C880	60/26	942	383	0	35	2.7	100	100
C1355	41/32	1574	546	8	86	8.3	99.49	100
C1908	33/25	1879	880	9	117	18.2	99.52	100
C2670	157/64	2595	1193	117	71	12.2	95.49	100
C3540	50/22	3428	1669	137	130	39.2	96	100
C5315	178/123	5350	2307	59	76	17.1	98.90	100
C6288	32/32	7744	2416	34	29	31.8	99.56	100
C7552	206/107	7548	3512	131	97	28.8	98.26	100
Total	834/470	32342	13268	507	738	163.2	97.71	100

PI/PO: Primary Input/Output, RF: Redundant Faults, TP: Test Patterns
TG Time: Test Generation Time, FC: Fault Coverage, TE: Test Efficiency

Polytec [12] is used to validate all the results. Tables 1 shows the test generation results for each individual circuit in ISCAS'85 benchmarks. From Table 1 we find that totally 738 test patterns are required to detect all 32342-507=31835 detectable faults in the 10 ISCAS'85 benchmark circuits. The total test generation time is 163.2 seconds. If all of these circuits are put into the same chip and all their inputs are connected into a single scan chain, then we will need 834*130=108420 clock cycles to apply all patterns to all circuits, where 834 is the sum of input numbers of all circuits and 130 is the largest number of required test patterns among all circuits. On the other hand, if multiple scan chains are used, then the test application time will be 206*130=26780 clock cycles (assume only one test session is used [3]), where 206 is the largest number of inputs and 130 is the largest number of test patterns for all circuits. The above two results are given in the second and third columns of Table 2.

Table 2: Experiment results for ISCAS'85.

	Single	Multiple	Method 1	Method 2
TE (%)	100%	100%	100%	100%
# TP	130	130	195	177
Scan chain length	834	206	206	206
TG Time (sec)	163.2	163.2	122.2	130.3
TA Cycles	108420	26780	40170	36462
Normalized TA Cycles	4.05	1	1.50	1.36
	1	0.25	0.37	0.34

TE: Test Efficiency, TP: Test Pattern
TG Time: Test Generation Time, TA: Test Application

The fourth and fifth columns of Table 2 show the results of our first method (Method1) and second method (Method2), respectively. Totally 195 and 177 test patterns are required to detect all faults in 10 ISCAS'85 circuits using Method1 and Method2, respectively. Clearly these numbers are significantly smaller than the total number of patterns required for all circuits (738). The test generation time (122.2 and 130.3 seconds) is also less than the sum of that for each individual circuit (163.2 seconds). The test application time is calculated by $206\times195=40170$ ($206\times177=36462$) cycles for Method1 (Method2), which is about 37% (34%) of the single scan chain method and 150% (136%) of the multiple scan method.

For the sequential benchmark circuits, we assume that only the flip-flops of the circuits are chained together. The results for individual circuit process is given in Table 3, where we find that totally 144027-6786=137241 faults can be detected by 1356 patterns in 1293.3 CPU seconds (test generation time). The results of our methods and their comparison with single and multiple scan schemes are given in Table 4. The meanings of the second and third columns are the same as those in Table 2, where we can see that the total test application times are 1850947 and 485568 for single and multiple scan chain scan chains, respectively.

The 4th and 5th columns respectively show the results of Method1 and Method2. An interesting result is that the number of test patterns using Method2 is even smaller than the largest number of patterns required for a single circuit S13207, and hence the test application time using this method is even shorter than that using multiple scan chains. This can be explained as follows. Since the number of test patterns required for S13207 is much larger than the maximum of those required for other circuits (281 v.s. 174), and the number of FFs of S13207 is much smaller than the maximum number of FFs of other circuits (669 v.s. 1728), it is likely that the number of patterns required for S13207 will dominate the final total number of patterns for all circuits. Also since in general test compaction is NP-complete and can only be dealt with using heuristic methods, it is possible that the final total number of test patterns for all circuits is slightly smaller than that required for the single "dominating" circuit. For the combinational ISCAS'85 circuits, the difference of the numbers of test patterns between the "dominating" circuit (C3540) and other circuits is not as great as that for ISCAS'89. Hence the number of final total number of test patterns for all circuits (177) is large compared to the number 130 required for C3540. However we believe that by other virtual circuit selection method, this number (177) can be further reduced. The experiment on the ISCAS'89 circuits also suggest that it may be totally unnecessary to use multiple scan chains if test application time is of the main concern; using a single line to broadcast test patterns to all flip-flops may achieve the same fault coverage.

Unlike the case of ISCAS'85 circuits, the test generation time in our methods for ISCAS'89 circuits is

Table 3: Individual test generation results of ISCAS'89 circuits.

Circuits	# PI/PO	# FF	# Faults	# Gates	# RF	# TP	TG Time (sec)	FC (%)	TE (%)
S1238	14/14	18	1355	598	69	149	9.9	94.91	100
S1423	17/5	74	1515	906	14	43	7.4	99.08	100
S1494	8/19	6	1506	735	12	117	4.9	99.20	100
S5378	35/49	179	4603	3400	40	134	29.7	99.13	100
S9234	19/22	228	6927	6326	452	159	88.2	93.47	100
S13207	31/121	669	9815	10167	153	281	200.4	98.44	100
S15850	14/87	597	11727	11739	391	158	239.1	96.67	100
S35932	35/320	1728	39094	21903	3984	28	164.1	89.81	100
S38417	28/106	1636	31180	27379	165	113	224.9	99.47	100
S38584	12/278	1452	36305	24173	1506	174	324.7	95.85	100
Total	213/1021	6587	144027	107326	6786	1356	1293.3	95.28	100

PI/PO: Primary Input/Output, FF: Flip-Flops, RF: Redundant Faults, TP: Test Patterns
TG Time: Test Generation Time, FC: Fault Coverage, TE: Test Efficiency

longer than the sum of time for each individual circuit. We have found that this is because when the virtual circuit is very large (107,326 gates in our case), memory swapping activity will consume a great amount of CPU time during ATPG process.

Table 4: Experiment results for ISCAS'89.

	Single	Multiple	$Method1$	$Method2$
TE (%)	100	100	100	100
# TP	281	281	287	280
Scan chain length	6587	1728	1728	1728
TG Time (sec)	1293.9	1293.9	1802.0	1893.7
TA Cycles	1850947	485568	495936	483840
Normalized TA Cycles	3.81	1	1.02	0.99
	1	0.26	0.27	0.26

TE: Test Efficiency, TP: Test Pattern
TG Time: Test Generation Time, TA: Test Application

6 Conclusions

In this paper we present a concept of virtual circuits for ATPG to generate test patterns that can be shared by all circuits driven by multiple scan chains. With this concept the supporting of a single scan input line to multiple scan chains becomes possible. Compared with conventional single and multiple scan chain architectures, great test time reduction and much less hardware overhead are obtained, respectively. The surprising result that a small number of test patterns can detect all faults in many circuits is due not only to the exploration of test pattern compatibility but also to the advantage taking of the ATPG feature in both random and deterministic test generation procedures. Finally, we would like to comment on the future work. We strongly believe that even better results are possible because in this paper only two out of the $\frac{N_1!}{(N_1-N_2)!} \cdot \frac{N_1!}{(N_1-N_3)!} \cdots \frac{N_1!}{(N_1-N_k)!}$ virtual circuit selection methods are used. We also believe that, though in this paper the presented methodology is only applied to disjoint circuits, its application to a single, large circuit is also possible. The pseudo-exhaustive methods [13] used in some built-in self-test systems that take advantage of the independence property of signal lines or inputs give a clue in this research direction.

References

[1] R.S. Fetherston, I.P. Shaik and S.C. Ma, "Testability Features of $AMD-K6^{TM}$ Microprocessor" *Proc. of the Intn'l Test Conf.,* 1997, pp.406-413.

[2] Zhang and R.D. McLeod, "An Efficient Multiple Scan Chain Testing Scheme" *Sixth Great Lakes Symposium on VLSI,* March. 1996, pp.294-297.

[3] S. Narayanan and M.A. Breuer, "Asynchronous Multiple Scan Chains" *Proc. of VLSI Test Symposium,* May. 1995, pp.270-276.

[4] S. Lee and K.G. Shin, "Design for Test Using Partial Parallel scan" *IEEE Trans. on Computers.* 1990, pp.203-211.

[5] "IEEE Standard 1149.1-1990. IEEE Standard Test Access Port and Boundary Scan Architecture." *IEEE Standard Board, New York, N.Y., 1990.*

[6] P. Goel, "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuit," *IEEE Trans. on Computers,* Vol C-30, No.3, pp.215-222, March, 1981.

[7] P. Goel and B.C. Rosales, "Test Generation & Dynamic Compaction of tests," *in digest of Test Conference,* pp.189-192, Oct. 1979.

[8] M. Abramovici and J.J. Kulikowski, "Smart and Fast: Test generation for VLSI scan-design circuits," *IEEE Design and Test of Computers,* pp.43-54, Aug. 1986.

[9] J.S. Chang and C.S. Lin, "Test Set Compaction for Combinational Circuits" *IEEE Trans. on Computers,* Nov, 1995, pp.1370-1378.

[10] B. Krishnamurthy and S.B. Akers, "On the complexity of estimating the size of a test set" *IEEE Trans. on Computers,* Aug. 1984, pp.750-753.

[11] P.H. Bardell and W.H. McAnney, "Self-Testing of Multichip Logic Modules," *Proc. Intn'l. Test Conf.,* Nov, 1982, pp.200-204.

[12] H.K. Lee and D.S. Ha, "An Efficient Forward Fault Simulation Based on the Parallel Pattern Single Fault Propagation," *Proc. of Intn'l Test Conf.,* Oct. 1991.

[13] E.J. McCluskey, "Verification Testing- A Pseudoexhaustive Test Technique," *IEEE Trans. on Computers,* Vol. C-33, No. 6, June, 1984, pp.541-546.