Parity-Scan Design to Reduce the Cost of Test Application

Hideo Fujiwara and Akihiro Yamamoto

***Abstract*—The cost of testing consists mainly of the cost of test generation and the cost of test application. Scan design approach is a representative of those techniques that can reduce the cost of test generation for sequential circuits. However, the length of a test sequence for the scan design approach can grow quite large due to the scan operation shifting the values into the scan chain, which makes the cost of test application large. This paper proposes a design-for-testability approach called *Parity-Scan Design* which can reduce the cost of test application as well as the cost of test generation for sequential circuits. The parity-scan design approach is a combination of scan technique and parity testing. Two types of parity-scan designs, pre-parity and post-parity scan design, are presented. Experiments on ISCAS89 circuits show that as high as 91.2% (91.1%) test length reduction and 32.4% (27.0%) average reduction can be obtained for pre-parity (post-parity) scan design under the single scan chain approach. More reduction can be achieved by applying a multiple scan chain technique.**

***Index Terms*—Cost of test application, cost of test generation, design for testability, parity test, sequential circuits, scan design.**

Manuscript received October 31, 1992; revised April 12, 1993. This paper was recommended by Associate Editor K.-T. Cheng.

H. Fujiwara is with the Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma-shi, Nara 630-01, Japan.

A. Yamamoto is with the Department of Computer Science, Meiji University, Tama-Ku, Kawasaki 214, Japan.

IEEE Log Number 9212221.

I. Introduction

Testing has two main stages: the generation of tests for a given circuit and the application of these tests to the circuit. The cost of testing consists mainly of the cost of test generation and the cost of test application. Hence, design for testability should be considered to reduce both costs of test generation and test application. As a representative of those techniques that can reduce the cost of test generation, scan design approach [1], [2] can greatly reduce the cost of test generation for sequential circuits. However, the length of test sequence for scan design approach can grow quite large due to the scan operation shifting the values into and out of the scan chain, which makes the cost of test application quite large. Suppose a sequential circuit with a single scan chain of N_f flip-flops and an automatic combinational test generator generates N_t test patterns for the combinational logic part of the sequential circuit. Each generated test pattern is expanded into as many steps or shift clocks as are required to serially shift the N_f pseudo-input values into the scan chain. Hence, a set of N_t test patterns results in a test sequence of length $N_f(N_t + 1) + N_t$ where the first term $N_f(N_t + 1)$ indicates the total number of shift clocks and the second term N_t is the total number of normal clocks.

Several approaches have been proposed to reduce test application time. Parallel scan chains [1] can be used whereby the total number of scan chains are divided into K chains of length N_f divided by K. Therefore, the total number of shift clocks can be reduced from $N_f(N_t + 1)$ to $(N_f(N_t + 1))/K$. This reduction can be very high if K is large. However, K parallel scan chains require large pin overhead; additional K primary inputs and K primary outputs for K shift registers. Test set compaction [3] is considered to reduce the number of test patterns by combining some test patterns into one test pattern. However, one cannot expect high reduction of test length from the test set compaction approach. Partial scan design [4] can also reduce the length of test sequence by including only a subset of flip-flops in the scan chain. However, this requires a sequential test generator which increases the cost of test generation and again one cannot expect high reduction of test length from the partial scan approach. The test application time can be reduced by ordering the flip-flops in the scan chain [5]. However, only a low reduction in range 14% to 18.8% can be obtained according to the experimental results shown in [5]. A selectable length partial scan approach [6] is proposed where test length reductions in range 63% to 70.5% have been achieved for five large circuits. However, the test length reduction achieved depends highly on how effectively the scan flip-flops can be partitioned into a high-frequency group and low-frequency group.

This paper proposes a design-for-testability approach which can reduce the cost of test application as well as the cost of test generation for sequential circuits. The proposed method called Parity-Scan Design is a combination of scan technique and parity testing. First, we shall show that the potential testability by parity testing is very high according to the experiments on ISCAS89 benchmark circuits [7]. Then we shall introduce the parity-scan design method by combining a scan design technique and parity testing of pseudo-outputs, and present the condition of an input sequence to be a test sequence for a sequential circuit with parity-scan design under a single scan chain. Two types of parity-scan designs, pre-parity and post-parity scan design, are presented. To obtain more reduction of test length, we shall extend the parity-scan design under a single scan chain to that under a multiple scan chain. The multiple scan chain approach divides a scan into plural scan chains without pin overhead by sharing scan-in and scan-out pins among them. Experiments on ISCAS89 circuits show that as high as 91.2% (91.1%) test length reduction and 32.4% (27.0%) average reduction can be obtained for pre-parity (post-parity) scan design under the single scan chain approach. More reduction can be achieved by applying the multiple scan chain technique.

II. Parity Testability of Flip-Flops

As mentioned in the preceding section, here we consider a design-for-testability method that can reduce both costs of test generation and test application for sequential circuits. To satisfy the former, i.e., to reduce the cost of test generation, we adopt the scan design technique. Assuming scan design we futher consider to reduce the cost of test application.

The purpose of scan process is (a) to set the flip-flops in a circuit to any desired state and (b) to observe the internal state of the flip-flops in a circuit. The former is the controllability enhancement of flip-flops, and the latter is the observability enhancement of flip-flops. To reduce the scan process, we have to consider something to substitute for it. Before resolving the controllability issue, let us first consider what can be substituted for the scan process to enhance the observability of flip-flops.

Fig. 1 shows a sequential circuit with a double-latch design. All clocked flip-flops are implemented as a set of master-slave latches L_2 and L_2. Cutting the feedback loops where the clocked flip-flops are, we can get *pseudo-inputs* and *pseudo-outputs* as shown in Fig. 2. Further, inserting a cascade of Exclusive-OR gates on pseudo-outputs, we can get a circuit with parity testing for pseudo-outputs in Fig. 2. In this circuit of Fig. 2, if an error of a fault propagates to the odd number of pseduo-outputs, the error can be observed at the output of the Exclusive-OR cascade. Here, it is an interesting issue to investigate how many percentage of faults can be detected by such a *parity testing*. To see this, we have made an experiment on the ISCAS89 circuits [7]. The ISCAS89 circuits have been changed into the structure of Fig. 2. Then the FAN algorithm [8] has been applied to the modified ISCAS89 circuits with the structure of Fig. 2. The results are given in Table I. The following columns are contained in the table:

1) Circuit Name—The assigned name for the circuit.
2) #gates—The number of logic gates. Primary inputs and primary outputs are considered gates.
3) #DFFs—The number of D flip-flops which are contained in the circuit.
4) #faults (total)—The total number of fault equivalence classes generated from the circuit.
5) #faults (redund)—Then number of redundant faults identified by FAN.

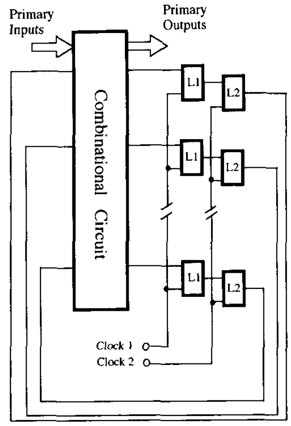

Fig. 1. A sequential circuit.

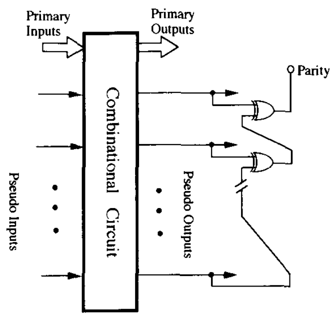

Fig. 2. Parity testing of pseudo-outputs.

6) #faults (abort)-The number of faults aborted by FAN due to a backtrack limit of 100.
7) #patterns-The number of test patterns generated by FAN.
8) #detected faults (PO)—The number of faults detected at primary outputs.
9) #detected faults (odd)—The number of faults that can be detected at the odd number of pseudo-outputs with at least one test pattern generated by FAN but that cannot be detected at primary outputs (i.e., detectable at the parity output with at least one test pattern but undetectable at the primary output).
10) #detected faults (even)—The number of faults that can be detected only at the even number of pseudo-outputs when detected with test patterns generated by FAN (i.e., detectable at the pseudo-outputs but neither detectable at the parity output nor at the primary outputs).
11) Parity Testability—The ratio of the number of detectable faults by parity testing to the total number of detectable faults, i.e.,

$$\frac{\#\text{PO} + \#\text{odd}}{\#\text{PO} + \#\text{odd} + \#\text{even}} \times 100\%$$

We can easily expect that, for each fault that can be detected at pseudo-outputs, the probability of the fault to be detected at the odd number of pseudo-outputs with *at least one* test-pattern is much higher than the probability of the fault not to be detected at the odd number of pseudo-outputs for *all* test-patterns. So, we have obtained the expected results that #odd is larger than #even for all benchmark circuits as shown in Table I.

TABLE I
ISCAS89 ATPG RESULTS BY FAN

Circuit Name	# gates	# DFF's	# faults			# patterns	# detected faults			Parity Testability %
			total	redund	abort		PO	odd	even	
s208	133	8	217	0	0	44	90	124	3	98.6
s298	170	14	308	0	0	41	18	283	7	97.7
s344	225	15	342	0	0	36	33	282	27	92.1
s349	226	15	350	2	0	37	32	292	24	93.1
s382	232	21	399	0	0	48	12	369	18	98.5
s386	199	6	384	0	0	90	148	232	4	99.0
s400	239	21	424	6	0	46	12	387	19	95.5
s420	265	16	455	0	0	93	194	245	16	96.5
s444	255	21	474	14	0	55	12	429	19	95.9
s510	255	6	564	0	0	74	158	398	8	98.6
s526	265	21	555	1	0	90	18	530	6	98.9
s526n	266	21	553	0	0	91	18	528	7	98.7
s641	495	19	467	0	0	80	160	305	2	99.6
s713	508	19	581	38	0	72	168	372	3	99.4
s820	368	5	850	0	0	162	277	566	7	99.2
s832	367	5	870	14	0	156	277	572	7	99.2
s838	523	32	931	0	0	191	402	487	42	95.5
s953	521	29	1079	0	0	114	46	958	75	93.0
s1196	615	18	1242	0	0	191	550	692	0	100.0
s1238	598	18	1355	69	0	208	587	698	1	99.9
s1423	906	74	1515	14	0	126	42	1385	74	95.1
s1488	741	6	1486	0	0	170	838	644	4	99.7
s1494	735	6	1506	12	0	166	844	646	4	99.7
s5378	3400	179	4603	40	0	497	1332	2997	234	94.9
s9234	6326	211	6927	413	40	580	151	6144	179	97.2
s13207	10 167	638	9815	149	2	721	707	7360	1597	83.5
s15850	11 739	534	11 725	388	1	670	751	10 166	419	96.3
s35932	21 903	1728	39 094	3984	0	1009	2507	32 573	30	99.9
s38417	27 379	1636	31 180	161	4	2386	267	28 461	2287	92.6
s38584	24 173	1426	36 303	1500	6	1562	1600	30 853	2344	93.3

From the results shown in Table I, we can see that the *parity testability* of ISCAS89 circuits is 94.6% on average and is in the range of 83.5% to 100%. Hence, most of the detectable faults can be tested at primary outputs including the parity output. This suggests that the parity testing approach of Fig. 2 could be substituted for the scan process of observing the contents of flip-flops.

III. Parity-Scan Design

Parity-Scan Design is a variation of the scan design approach which is a combination of the scan design and parity testing. Examples of parity-scan flip-flops are shown in Fig. 3 which consists of a shift-register latch (SRL) used in LSSD [2] and an Exclusive-OR gate for parity test. The SRL consists of two latches, L_1 and L_2, which have the scan input I, the data input D, the system clock C, and two shift-control inputs, A and B. Fig. 3(a) shows a *pre-parity type* parity-scan flip-flop where the Exclusive-OR gate is located before the flip-flop to take the parity of the data input to the flip-flop. On the other hand, Fig. 3(b) shows a *post-parity* type parity-scan flip-flop where the Exclusive-OR gate is located after the flip-flop to take the parity of the flip-flop.

Using these parity-scan flip-flops, two types of general structures for double-latch parity-scan design are obtained as illustrated in Fig. 4(a) and 4(b). All storage elements are implemented as a set of master-slave latches, L_1 and L_2. Each of master–slave latches is connected in series and clocked by two nonoverlapping clocks C_1 and C_2, where C_2 is equivalent to B. Each of the Exclusive-OR gates is also connected to form an Exclusive-OR cascade with a parity output. Fig. 4(a) and 4(b) show the pre-parity type and post-parity type parity-scan designs, respectively.

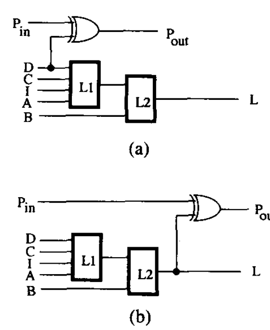

Fig. 3. (a) Pre-parity scan flip-flop. (b) Post-parity scan flip-flop.

In the shift register mode, these latches are chained to form a shift register under the control of clocks A and B. Test patterns are applied to the combinational circuit by scanning them into the shift register and applying them at the primary inputs. Then the clock C_1 is set to 1 and the response of the combinational circuit is captured in the L_1 latches and at the primary outputs. The result of the test captured in the register is then scanned out.

The *Parity-Scan Design* has also the capability of parity testing for the contents captured in the L_2 latches or flip-flops before or after capturing. The pre-parity scan design can take the parity of

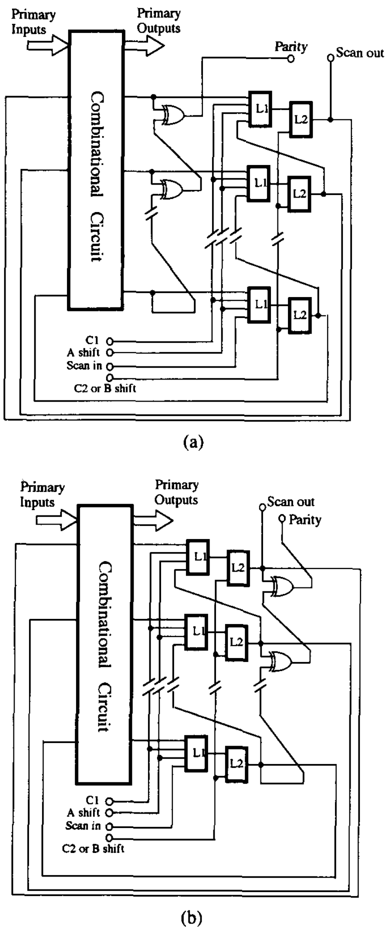

Fig. 4. (a) Pre-parity scan design. (b) Post-parity scan design.

the data input to the flip-flops and the post-parity scan design can take the parity of the flip-flops. If an error of a fault propagates to the odd number of flip-flops, the error can be observed at the parity output of the Exclusive-OR cascade. This parity test operation will be used as a substitute for the scan operation of observing the contents of flip-flops.

The area overhead due to an Exclusive-OR gate for each scan flip-flop is not so large. Since we are considering large circuits for which the cost of test application will become quite large, reducing the test application cost at the expense of extra area overhead will be a good trade-off if the reduction is guaranteed to be high. As for another issue, the cascading of Exclusive-OR gates beyond certain numbers may force slowing down the test application frequency. To alleviate this issue, one may adopt a tree of Exclusive-OR gates instead of the cascade of Exclusive-OR gates, partition the parity function into several sub-functions, choose a subset of flip-flops to get a partial parity, and so on.

IV. Test Sequence for Parity-Scan Design

Let us classify all detectable faults in the combinational logic part of a sequential circuit, i.e., combinational irredundant faults, into two groups; one is a set of faults that are detectable at primary outputs and/or a parity output, and the other is a set of faults that are detectable only at the even number of pseudo-outputs. Let us call the former *parity-testable faults* and the latter *parity-untestable faults.*

Assume two consecutive test patterns $T(i)$ and $T(i + 1)$. Suppose a fault f that is tested by the test pattern $T(i)$. If the fault f is parity-testable, it is detected at the primary outputs when $T(i)$ is applied and/or at the parity output. In case of pre-parity scan design, the fault is detected at the parity output at the same time when $T(i)$ is applied. In case of post-parity scan design, the fault is detected at athe parity output at the next time when $T(i + 1)$ is applied. For parity-testable faults in both cases, hence, the scan operation to observe the flip-flops can be omitted. Furthermore, the scan operation for setting flip-flops to the values of pseudo-inputs for the next test $T(i + 1)$ can be omitted if the values at the pseudo-outputs of the current test $T(i)$ are identical to the values at the pseudo-inputs of the next test $T(i + 1)$. Therefore, if the values at the pseudo-outputs of the current test $T(i)$ are *identical* to the values at the pseudo-inputs of the next test $T(i + 1)$ and if $T(i)$ is aimed at testing *parity-testable* faults only, then the scan operation between $T(i)$ and $T(i + 1)$ can be omitted. Let us call this state *no scan with clock*. This is illustrated in Fig. 5(a).

In case of pre-parity scan design, we can further omit scan operation. If no system clock to the flip-flops is applied between $T(i)$ and $T(i + 1)$, the flip-flops can hold the same contents and hence the values at the pseudo-inputs of the next test $T(i + 1)$ are identical to the values at the pseudo-inputs of the current test $T(i)$. In case of pre-parity scan design, hence, if the values at the pseudo-inputs of current test $T(i)$ are *identical* to values at the pseudo-inputs of the next test $T(i + 1)$ and if $T(i)$ is aimed at testing *parity-testable* faults only, then the scan operation between $T(i)$ and $T(i + 1)$ can be omitted. Let us call this state *no scan without clock*. This isillustrated in Fig. 5(b).

If the fault f is parity-untestable, the error caused by the fault is propagated only to the even number of pseudo-outputs when the test pattern $T(i)$ is applied. So, in order to detect the fault f at this time by the test pattern $T(i)$, the values with error captured at flip-flops have to be shifted out to observe at the scan output. Therefore, either if the values at the pseudo-outputs of the current test $T(i)$ are *not* identical to the values at the pseudo-inputs of the next test $T(i + 1)$, or if $T(i)$ is aimed at testing some *parity-untestable* fault, then the scan operation between $T(i)$ and $T(i + 1)$ *cannot* be omitted. Let us call this state *scan*. This is illustrated in Fig. 5(c).

Without loss of generality, a test sequence for a circuit with parity-scan design is represented by

Test-Patterns:		$T(1)$	$T(2)$	$\cdots$		$T(n)$
States:	$S(0)$	$S(1)$	$S(2)$	$\cdots$	$S(n-1)$	$S(n)$

where each $S(i)$ indicates scan with clock, scan without clock, or scan. Let us abbreviate this test sequence as $T(1)T(2) \cdots T(n)/S(0)S(1) \cdots S(n)$.

Here, we define the consistency of a test sequence for parity-scan design. A test sequence is *consistent with parity-scan design* if

1) $S(0) = S(n) =$ scan, and
2) for each $T(i)$ $(i = 2, \cdots, n - 1)$, $S(i)$ is no scan with clock only when $SO(T(i)) = SI(T(i + 1))$ and $S(i)$ is no scan without clock only when $SI(T(i)) = SI(T(i + 1))$.

Further, we consider two types of testing for each fault under test; (a) *parity testing* which detects the fault at either the parity output or primary outputs, and (b) *scan testing* which detects the fault at the scan output by scan operation. For parity-scan designed circuits, we can hence define a test sequence as follows:

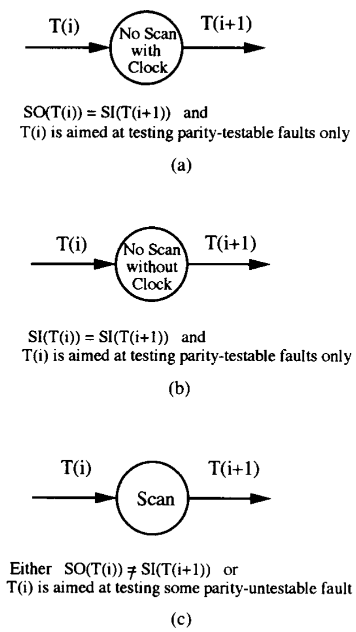

Fig. 5. Scan/no scan states. (a) No scan with clock. (b) No scan without clock. (c) Scan.

A test sequence T satisfying the following conditions is called a *parity-scan test sequence*:

1) T is consistent with parity-scan design, and
2) each detectable fault is tested by parity testing or scan testing in some sub-sequence of T.

Let us evaluate the length of a parity-scan test sequence. Let N_f and N_t be the number of flip-flops and test patterns, respectively. Let N_s and N_{ns} be the number of scan states and no scan (with/without clock) states, respectively. If each test pattern appears only once in the parity-scan test sequence, $N_s + N_{ns} = N_t + 1$. Then the length of the parity-scan test sequence becomes

$$\begin{aligned}&(\text{number of flip-flops}) \times (\text{frequency of scan})\\ &\quad + \text{number of test-patterns})\\ &= N_f N_s + N_t.\end{aligned}$$

The test length reduction of the parity-scan design to the orginal scan design is

$$1 - \frac{N_f N_s + N_t}{N_f(N_t + 1) + N_t} = \frac{N_f N_{ns}}{N_f(N_t + 1) + N_t}$$

For given N_f and N_t, this reduction is maximized when N_{ns} is maximized or N_s $(= N_t + 1 - N_{ns})$ is minimized. Since the values N_{ns} and N_s are influenced by the ordering of test patterns in a parity-scan test sequence, the problem is how to determine an ordering of test patterns that leads to the minimum test sequence. For N_t test patterns, however, the number of possible orderings is $N_t!$, which makes it almost impossible to get the optimum length when N_t is large. It is also important to construct a parity-scan test sequence with low time overhead compared to the test generation time for the combinational logic part of the circuit under test. So, we shall consider a simple procedure for searching an ordering of test patterns that reduces the number of scan operations.

The procedure for constructing a parity-scan test sequence $T(1)T(2)\cdots T(n)/S(0)S(1)S(2)\cdots S(n)$ operates as follows.

First, choose a fault f from the fault table of the circuit under test. Generate a test pattern T_f for the fault f by using the combinational test generation algorithm such as FAN [8]. Specify all undetermined values or X's at primary inputs and/or pseudo inputs of T_f by arbitrary values 0 or 1 and perform fault simulation to find all detectable faults by the test pattern T_f and to classify those detected faults into parity-testable and parity-untestable faults. Then, select T_f as the first test pattern $T(1)$ for the parity-scan test sequence. Let $S(0)$ be the *scan* state to initialize the pseudo-inputs of T_f. Those parity-testable faults can be detected by T_f even when $S(1)$ is non-scan. Those parity-untestable faults, however, can be detected by T_f only when $S(1)$ is scan, i.e., by scan-out processing.

Next, for some fault g which has not yet been detected, find a test pattern T_g such that the values at the pseudo-inputs of T_g are identical to those at the pseudo-outputs of $T(1)$. In case of pre-parity scan design, find a test pattern T_g such that the values at the pseudo-inputs of T_g are identical to those either at the pseudo-inputs or at the pseudo-outputs of $T(1)$. Here, the test pattern T_g is either generated by FAN or taken from a hash table which stores test patterns generated by FAN for reuse. For each fault, FAN is used at most once to avoid wasting CPU time. If such a test pattern T_g exists, put it on the second test pattern $T(2)$ and make the state $S(1)$ between $T(1)$ and $T(2)$ be *no scan* (*with/without clock*). If such a test pattern T_g does not exist, put a test pattern, which can detect the fault g, on the second test pattern $T(2)$ and make the state $S(1)$ between $T(1)$ and $T(2)$ be *scan*. During this process of searching a test pattern T_g, all test patterns, which have been newly generated by FAN and have not been adopted as a test pattern, are stored in the hash table to reuse them afterward. After test pattern generation, fault simulation is performed to find all other detectable faults by the test pattern T_g and to classify those faults into parity-testable and parity-untestable faults. Again, those parity-testable faults can be detected by T_g even when $S(2)$ is non-scan. On the other hand, those parity-untestable faults can be detected by T_g only when $S(2)$ is scan, i.e., by scan-out processing. Note that, before the fault simulation of a test-pattern $T(i)$, all fault lists on the pseudo-outputs of the previous test-pattern $T(i - 1)$ are set to the pseudo-inputs of the current test-pattern $T(i)$.

Continue the above process until all faults are detected. The test sequence $T(1)T(2)\cdots T(n)/ S(0)S(1)S(2)\cdots S(n)$ generated by the process mentioned above always satisfy the condition of the parity-scan test sequence.

V. Experimental Results

We have made experiments on the ISCAS89 circuits [7]. The ISCAS89 circuits contain flip-flops which are assumed to be fully scannable and parity-testable as shown in Fig. 4(a) and 4(b). The FAN algorithm [8] was used to generate test patterns for each combinational logic part of ISCAS89 circuits. From the generated test patterns, parity-scan test sequences were constructed for both pre-parity and post parity types. The experiments ran on a SUN 4/60, a 12 MIPS machine with 12MB of memory. The results for pre-parity and post-parity types are given in Tables II and III, respectively. The following columns are contained in the tables:

1. Circuit Name—The assigned name for the circuit.
2. #scan frequency (scan)—The frequency of scan operation for scan design.
3. #scan frequency (parity-scan)—The frequency of scan operation for parity-scan design.

TABLE II
ISCAS89 TEST LENGTH REDUCTION FOR PRE-PARITY TYPE

Circuit Name	Scan Frequency		Test Sequence Length		Test Length Reduction	CPU Time (sec)	
	scan	Parity-Scan	scan	Parity-Scan	%	scan	Parity-Scan
s208	45	29	404	279	30.9	1	1
s298	42	32	629	495	21.3	1	1
s344	37	28	591	461	22.0	1	1
s349	38	27	607	443	27.0	1	1
s382	49	37	1077	828	23.1	1	2
s386	91	48	636	380	40.3	1	3
s400	47	37	1033	829	19.7	1	2
s420	94	75	1597	1296	18.8	2	2
s444	56	42	1231	936	24.0	1	1
s510	75	47	524	354	32.4	2	3
s526	91	77	2001	1707	14.7	2	3
s526n	92	78	2023	1730	14.5	1	4
s641	81	26	1619	579	64.2	1	3
s713	73	34	1459	729	50.0	2	5
s820	163	66	977	487	50.2	6	12
s832	157	63	941	471	49.9	6	11
s838	192	169	6335	5602	11.6	7	15
s953	115	78	3449	2391	30.7	5	9
s1196	192	9	3647	359	90.2	11	18
s1238	209	8	3970	351	91.2	15	21
s1423	127	105	9524	7894	17.1	6	11
s1488	171	75	1196	617	48.4	13	22
s1494	167	88	1168	702	39.9	13	25
s5378	498	426	89 739	76 739	14.4	67	133
s9234	581	552	123 171	117 074	5.0	293	439
s13207	722	666	461 357	425 622	7.7	278	364
s15850	671	537	358 984	287 429	19.9	309	607
s35932	1010	305	1 746 289	527 385	69.8	1996	1827
s38417	2387	2096	3 907 518	3 431 203	12.2	2231	4251
s38584	1563	1384	2 230 400	1 975 036	11.4	2064	3444

4. Test Sequence Length (scan)—The length of test sequences for scan design.
5. Test Sequence Length (parity-scan)—The length of test sequences for parity-scan design.
6. Test Length Reduction—The ratio of the reduced test length for parity-scan design to the test length for scan design, i.e.,

$$\left(1 - \frac{\text{Length of Parity-Scan Test Sequence}}{\text{Length of Scan Test Sequence}}\right) \times 100\%$$

7. CPU time (scan)—The total CPU time in seconds required to generate test patterns for scan design.
8. CPU time (parity-scan)—The total CPU time in seconds required to construct a parity-scan test sequence for parity-scan design.

From the results shown in Table II, as high as 91.2% test length reduction (32.4% on average) is achieved for pre-parity scan designs of ISCAS89 benchmarks. From Table III, as high as 91.1% test length reduction (27.0% on average) is achieved for post-parity scan designs of ISCAS89 benchmarks. The test set size or the number of test-patterns generated by FAN for parity-scan testing is nearly equal to that for conventional scan testing. So, the time overhead or the extra CPU time required to construct a parity-scan test sequence from generated test patterns is not so high. Both of the pre- and post-parity scan design approaches can reduce the cost of test application.

Tables II and III show the results for parity-scan designs under a single scan chain technique. One can easily extend the single scan chain approach to a multiple scan chain. More reduction can be expected to achieve by applying the multiple scan chain technique, which is described in the next section.

VI. Multiple Scan Chain

Let N_f be the number of flip-flops and N_t be the number of test patterns for the combinational logic part of a circuit under test. For a circuit with a single scan chain, the length of the test sequence is $N_f(N_t + 1) + N_t$. To reduce the length of the test sequence, parallel scan chains can be used whereby the total number of scan chains are divided into K chains of length N_f divided by K (see Fig. 6(a)). The total number of shift clocks can be reduced from $N_f(N_t + 1)$ to $(N_f(N_t + 1)/K$. However, K parallel scan chains require large pin overhead, i.e., additional K primary inputs and K primary outputs for K shift registers. To overcome this pin overhead, we consider a variation of the parallel scan chain approach. Fig. 6(b) shows the approach called *a multiple scan chain*. In this scan design, a long scan chain is divided into K scan chains without pin overhead by sharing one scan-in pin and one scan-out pin among them. Parallel scan chains can get K differential serial input sequences from K scan-in pins and shift in and out them in parallel. The multiple scan chain, however, cannot get K different input sequences in parallel but in series. So, the multiple scan chain approach cannot be expected to achieve the same high reduction of the number of shift clocks as the parallel scan chain approach.

For the conventional scan design approach, the multiple scan chain cannot contribute to the reduction of the number of shift

TABLE III
ISCAS89 Test Length Reduction for Post-Parity Type

Circuit Name	Scan Frequency		Test Sequence Length		Test Length Reduction %	CPU Time (sec)	
	scan	Parity-Scan	scan	Parity-Scan		scan	Parity-Scan
s208	45	29	404	275	31.9	1	1
s298	42	33	629	507	19.4	1	1
s344	37	32	591	519	12.2	1	1
s349	38	27	607	442	27.2	1	1
s382	49	37	1077	827	23.2	1	1
s386	91	71	636	515	19.0	1	3
s400	47	38	1033	849	17.8	1	1
s420	94	76	1597	1309	18.0	2	2
s444	56	45	1231	1002	18.6	1	1
s510	75	59	524	430	17.9	2	4
s526	91	84	2001	1860	7.0	2	4
s526n	92	80	2023	1772	12.4	1	6
s641	81	35	1619	742	54.2	1	3
s713	73	48	1459	997	31.7	2	5
s820	163	91	977	606	38.0	6	9
s832	157	104	941	685	27.2	6	11
s838	192	172	6335	5696	10.1	7	13
s953	115	74	3449	2266	34.3	5	11
s1196	192	7	3647	324	91.1	11	18
s1238	209	9	3970	370	90.7	15	21
s1423	127	107	9524	8038	15.6	6	14
s1488	171	135	1196	981	18.0	13	20
s1494	167	142	1168	1027	12.1	13	23
s5378	498	429	89 639	77 262	13.8	67	133
s9234	581	588	123 171	124 674	−1.2	293	483
s13207	722	674	461 357	430 716	6.6	278	497
s15850	671	591	358 984	316 256	11.9	309	618
s35932	1010	222	1 746 289	383 872	78.0	1996	1595
s38417	2387	2084	3 907 518	3 411 544	12.7	2231	4219
s38584	1563	1416	2 230 400	2 020 675	9.4	2064	3499

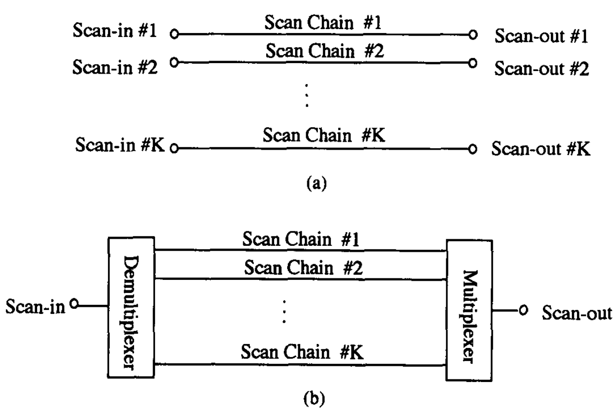

Fig. 6. (a) Parallel scan chains versus (b) multiple scan chains.

clocks. For the parity-scan design approach proposed in this paper, however, the multiple scan chain can contribute to reduce the number of shift clocks. When we construct a parity-scan test sequence from the generated test patterns, we have to put scan operation between two consecutive test patterns if the values at the pseudo-outputs of the current test pattern is not equal to the values at the pseudo-inputs of the next test pattern. However, this inequality of values occurs usually at very few scan locations or flip-flops, i.e., most of the flip-flops retain their state. Hence, in the multiple scan chain, only the sub-chains at which the inequality occurs need to be updated by scan operation while the remaining sub-chains retain their state without scan operation. This partial scan operation in the multiple scan chain can thus reduce the number of shift clocks or the number of scan operations. The advantage or this partial scan operation can also be applied in the case that we have to observe the contents of flip-flops when a parity-untestable fault is tested (see the conditions of a parity-scan test sequence).

VII. Conclusions

A new design for testability approach called *parity-scan design* which can reduce the cost of test application as well as the cost of test generation has been presented. This technique is useful in reducing the scan operations or the number of shift clocks, and hence the overall test sequence length. Significant reductions of test sequence length have been achieved on ISCAS89 benchmark circuits, as high as 91.2% (91.1%) reduction, and 32.4% (27.0%) on average for pre-parity (post-parity) scan design under the single scan chain approach. More reduction can be achieved by applying the proposed multiple scan chain technique.

References

[1] H. Fujiwara, *Logic Testing and Design for Testability.* Cambridge, MA: The MIT Press, 1985.
[2] E. B. Eichelberger and T. W. Williams, "A logic design structure for LSI testability," *Proc. Design Automation Conf.*, pp. 462–468, June 1977.
[3] P. Goel and B. Rosales, "Test generation and dynamic compaction of tests." in *Proc. Intl. Test Conf.*, pp. 189–192, 1979.

[4] V. D. Agrawal, K. Cheng, D. Johnson, and T. Lin, "Designing circuits with partial scan," *IEEE Design and Test of Computers*, pp. 8–15, Apr. 1988.
[5] R. Gupta and M. A. Breuer, "Ordering storage elements in a single scan chain," in *Proc. Intl. Conf. on Computer-Aided Design*, pp. 408–411, 1991.
[6] S. P. Morley and R. A. Marlett, "Selectable length partial scan: a method to reduce vector length," in *Proc. Intl. Test Conf.*, pp. 385–392, 1991.
[7] F. Brglez, D. Bryan, and K. Kozminski, "Combinational profiles of sequential benchmark circuits," in *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 1929–1934, May 1989.
[8] H. Fujiwara and T. Shimono, "On the acceleration of test generation algorithms," *IEEE Trans. Comput.*, vol. C-32, pp. 1137–1144, Dec. 1983.