A Case Study on the Implementation of the Illinois Scan Architecture

Frank F. Hsu* Kenneth M. Butler Janak H. Patel†

Texas Instruments Inc.
12500 TI Boulevard MS 8645
Dallas, TX 75243

Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1308 West Main Street, Urbana, IL 61801

Abstract

Scan based test techniques offer a very efficient alternative to achieve high fault coverage when compared to functional pattern testing. As circuit sizes grow ever larger, test data volume and test application time grow unwieldy even in the very efficient scan based designs. The Illinois Scan Architecture is a low cost alternative to conventional scan. In this paper, we present the first ever reported case study of the effectiveness of the Illinois Scan Architecture on an industrial circuit.

1 Introduction

Functional pattern testing used to be the predominant method applied to large volume integrated circuit (IC) testing. In recent years, structured test techniques, primarily full scan, are rapidly supplanting functional patterns as the only way to achieve high coverages with short development cycles and short tester times. However, typical ASICs today can have digital gate counts well in excess of one million two input NAND equivalent gates. Compounding the problem is the fact that typical VLSI automatic test equipment (ATE) are very restrictive in the number and depth of scan chains they can support. This is especially true for older legacy hardware, which is even more constrained than their current generation equivalent testers, but must still be used to minimize test cost.

For such circuits and manufacturing situations, even though scan test times are far shorter than functional pattern test times for a given coverage, they are long enough to increase the overall test cost to an unacceptably high level [1]–[4]. In addition to the cost of long test application times, the cost of large test data volume has become significant. Test data volume affects the cost in different ways. When the test data exceeds the tester memory depth, additional test data requires access to slower mass storage. When one also considers transition fault testing in addition to normal stuck-at testing, the test data volume indeed becomes a major concern.

One solution to this situation is logic BIST. In a logic BIST approach, test pattern generation and output response comparison functions are built into the chip. Typically, the fault coverage achievable with logic BIST alone is insufficient and must be supplemented with conventional deterministic scan patterns [5]–[8]. However, the number of scan patterns required to "top up" the coverage is considerably smaller than the total number of scan patterns required for equally high fault coverage in the complete absence of logic BIST. Furthermore, recent adaptations of logic BIST which are a hybrid of deterministic automatic test pattern generation (ATPG) and logic BIST have shown great promise in reducing tester data volume while maintaining high fault coverage [5], [7], [8].

Unfortunately, there are several difficulties with implementing full or hybrid logic BIST. First of all, there is an incremental area penalty for the logic BIST controller, modifications to the scan-enable signal, added routing, added controllability and observability, and other design changes necessary in logic BIST which can amount to 1-2% of the overall die area [6]. This 1-2% is above the die area impact for full scan, which alone can account for 4-5% area on large ASICs with many memories. Furthermore, logic BIST can complicate the circuit's timing closure. For example, one commercial BIST vendor requires all circuitry to operate within three artificial clock domains (X, $X/2$, $X/4$) regardless of the requirements that the chip's function places on the allocation and speed of clock domains. Such restrictions can require overdesign and can complicate the already complex tasks of clock distribution and clock tree balancing.

*Frank Hsu is now with Dallas Semiconductor, 4401 S. Beltwood Parkway, Farmers Branch, TX 75244.

†Janak Patel's research was supported by the Semiconductor Research Corporation under Contract SRC 99-TJ-717.

0-7803-7169-0/01 $10.00

Probably the largest hurdle to implementing logic BIST is the increased difficulty the BIST implementation imposes on closing timing for the circuit at full rated speed. Most commercial logic BIST implementations require an at-speed scan-enable, which is not necessary when implementing scan transition fault testing in conventional scan designs and utilizing the "functional justification" or "launch from capture" technique [9]. The increased complexity of the scan-enable function plus the constraints of the additional routing of signals to and from the BIST controller(s), linear feedback shift registers (LFSRs), and the circuitry itself greatly increase the complexity of placement, routing, and the overall timing closure process. Most successful logic BIST implementations are accomplished only by engineers who are very familiar with the unBISTed circuit and/or experts in the BIST function.

An alternative to logic BIST involves storing the patterns in compressed form and decompressing them on-the-fly [10], [11]. In this approach, deterministic patterns are reordered post-ATPG to minimize the Hamming distance between the patterns so that pattern differences may be calculated and stored in compressed form using run-length encoding. This approach requires added CPU resources to reorder the patterns and some logic overhead to implement cyclical scan chains and the run-length decoder. Also, the savings in data storage are somewhat modest.

There are a variety of other schemes involving compressed test data of one form or another. Jas and Touba give an excellent summary of other work in this area [10].

Yet another interesting solution to the test data volume problem is the recently proposed Illinois Scan Architecture (ILS) [12]. The Illinois Scan Architecture exploits the fact that a large chip may contain many subcircuits that are fairly independent from one another. These subcircuits may be tested in parallel by the same test patterns giving high fault coverage. This "independence" is exploited by rearchitecting the scan function such that portions of the scan chains are simultaneously scanned with identical scan-in data; called broadcast mode. The scan-out from multiple scan chains is routed to a MISR for signature calculation as in a full logic BIST implementation such as STUMPS [13].

Since the broadcast mode of ILS puts constraints on the test patterns, many faults become untestable. To cover these faults, the broadcast patterns must be supplemented with conventional scan patterns. However, it is not clear what are the properties of a successful implementation of ILS. How many scan partitions must be used? Where is the "knee" in the fault coverage curve when performing the broadcast function? What total tester time and data volume savings can be expected in an ILS circuit versus the same circuit with conventional scan?

In this paper, we attempt to answer these and similar questions. We apply the ILS methodology to an industrial circuit while varying the parameters of the implementation. We compare the tester time and data volume in the ILS implementations to that of the conventional scan insertion. We report these results and draw some general conclusions about the viability of the technique and the design elements necessary to ensure a net reduction in tester data volume and test application time. To our knowledge, this is the first study of the Illinois Scan Architecture on an industrial circuit. We believe this case study will help others who are contemplating using ILS in their designs.

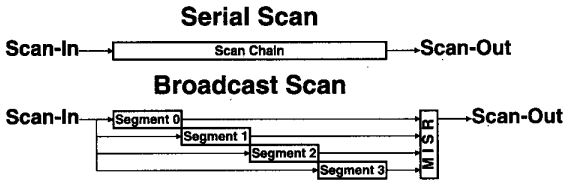

Figure 1: A diagram showing two modes of the Illinois Scan Architecture.

2 The Illinois Scan Architecture Methodology

The Illinois Scan Architecture was first described in [12]. Here we give a brief overview of that approach.

This scan architecture is shown in Figure 1. The top portion of the figure shows a normal serial scan mode, and the bottom figure shows that same scan chain in the Broadcast Mode of ILS. The scan-in pin that fed the original scan chain is routed to each of the scan chain segments. Note this shared scan-in idea is similar to the earlier work [14]. The segment scan-outs are routed to a multiple input signature register (MISR) as one might do in a logic BISTed circuit. This option of course necessitates following most logic BIST design rules - e. g., avoiding unknown states, contending internal buses, setup/hold invalidated flip-flop captures, or other BIST violations which could corrupt the signature(s). However, unlike logic BIST where the test data is pseudorandom, the deterministic ATPG used in ILS can filter out some of the offending test patterns and thus eliminate the need to follow complete logic BIST design rules. The tester need not store any information other than the reduced size scan-in data and the MISR signatures. The sum total

of this data can be significantly less than the storage requirements of a conventional scan architecture. And of course, the parallel nature of the scan chains in the broadcast mode requires significantly less test application time.

As mentioned previously, such a scan architecture will result in a fault coverage which is less than in a conventional full scan design. This drop in coverage is due to the fact that many patterns available in a conventional scan architecture can no longer be applied in the broadcast mode of ILS. The broadcast mode puts many constraints on the test patterns. These constraints make many faults untestable. Thus, it is necessary to implement a second mode of scan operation where the chains are restructured into a conventional scan configuration so that unconstrained scan patterns may be applied. The switching between broadcast mode and serial mode is relatively simple to implement in a design. The broadcast mode does not require extensive redesign of the conventional scan. The advantage here is that most of the routing changes are on scan-in and scan-out signals only, whose timing is typically not as critical as other scan signals such as clocks and scan-enables.

3 Improving Data Efficiency for Unbalanced Scan Chains

Some ASICs are designed with several modules, each having its own clock domain. Often each of these modules has its own scan chain(s). Since module sizes vary a great deal, the lengths of scan chains also vary a great deal. When unbalanced scan chains exist in the circuit and are used as parallel scan chains, the test application time for a scan pattern depends on the length of the longest scan chain. The scan patterns for the shorter chains are usually padded with 'X' values. Unfortunately, these 'X' values have to be explicitly stored in the tester memory even though they serve no purpose! Thus padding of shorter patterns with X's take up valuable tester memory. In contrast to parallel scan, the broadcast mode of Illinois Scan architecture ties all unbalanced chains to one scan input. Thus the explicit padding of shorter patterns is no longer needed and no space is wasted to store X's.

In addition, the ILS implementation divides the longer chains into multiple segments while improving the memory usage efficiency. Figure 2 (a) shows three scan chains of different lengths, with the length of the longest chain equal to 100 shifts and the other two chains of lengths 25 and 40. The parallel scan implementation of these three chains requires 300 bits of tester memory per scan pattern even though the actual test pattern requires only 100+40+25 = 165 bits.

The ratio of useful data bits to the bits required in the tester memory is defined as *Data Efficiency*. The data efficiency of the three scan chain parallel implementation, as shown in Figure 2 (a), would then be 165/300=55%. After dividing the longest chain in half for ILS implementation, as shown in Figure 2 (b), the useful data bits becomes 25+40+50=115 bits, and the data efficiency ratio improves to 115/150=76.7%. Dividing the two longer chains into even shorter segments, as shown in Figure 2 (c), raises the data efficiency to 93.3%.

Further division of scan chains in the ILS implementation continues to reduce the tester memory requirement per scan pattern. However, the total test data volume may not monotonically decrease due to addition of serial patterns to cover the faults not tested in the broadcast mode.

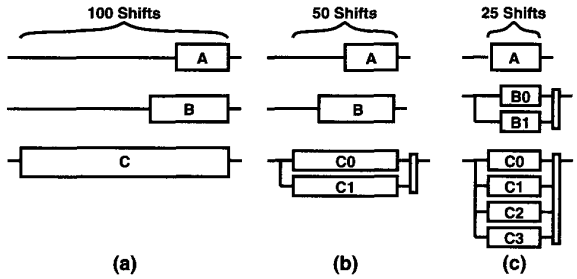

Figure 2: A diagram of an ILS implementation with unbalanced scan chains.

4 Results

In order to investigate the properties of the Illinois Scan designs, we selected a single ASIC from a third-party design house to use for detailed experiments. It is a full scan design with scan already present in the design when received from the customer. It has eight scan chains, as this is a limitation of the tester hardware used to test this device. We will report the results of ILS insertion for a number of ILS variations.

Rather than modify the design netlist many times to implement various ILS configurations on top of the conventional scan function, we took a different approach. In this work we are not actually performing a full ILS insertion, but instead simulating the presence of ILS in an unmodified netlist.

We arbitrarily chose a number of ILS segments, and we applied ATPG constraints to a commercial ATPG tool to cause it to behave as if the segments

were fed by the same scan-in pin/data, and were being scanned identically. So, for example, in a scan chain of length 100 that would be split into 5 segments, we would force the ATPG to adhere to the condition where the scan-in data of scan cells 1, 21, 41, 61, and 81 were identical, scan-in data for cells 2, 22, 42, 62, and 82 were identical, and so on. This permitted us to more easily investigate numerous ILS scenarios without having to manually create netlist representations for them all.

After constrained ATPG was complete, we would remove the constraints and apply normal ATPG to reach the desired level of fault coverage. Then, a combination of reverse and random fault simulation was performed, first using the serial scan patterns followed by the broadcast scan patterns. The reason for this fault simulation was that the serial scan patterns, while necessary, will cover a majority of the total faults and thus may make many broadcast scan patterns redundant. Lastly, we would calculate the total tester pattern storage requirements for the ILS and conventional scan data, and compare that result with the scan data requirements without ILS. We also applied the same reverse/random pattern simulation for compression purposes to the unmodified circuit to facilitate likewise comparisons.

We found that the ATPG tool had difficulty satisfying the ILS constraints if the number of segments exceeded 32, so we report results for ILS configurations with up to 32 divisions. Because of this ATPG limitation, we were also unable to establish the optimum number of divisions in our experiment. It is likely that division by numbers greater than 32 could improve the results reported below.

We should also note that the intended function of the circuit used in this study is unknown to the authors. Furthermore, no topological analysis was performed to optimize the selections of scannable flip-flops placed into each scan chain segment. The segmentation was done in a completely arbitrary manner. A more intelligent, topologically-based partitioning algorithm might also improve the results [12].

Finally, we should point out that the design we used is essentially BIST-ready, and no additional hardware would have been necessary to make the circuit compatible with a MISR scan-out scenario. However, as mentioned in Section 2, in the ILS approach, even if there are MISR-incompatible patterns, those patterns can be easily filtered out of the test set.

4.1 Single Scan Chain Configuration

The design used in our experiments contains approximately 158K two input NAND equivalent gates. We assume that only one scan chain exists in the design. The Illinois Scan Architecture implementation partitions this scan chain into N multiple equal-length segments, labeled as Div$N$ configuration in Table 1. The table also lists the pattern length and fault coverage values from broadcast scan (before reverse fault simulation), serial scan, and broadcast scan (after reverse fault simulation) for various chain configurations. Column number 1 in Table 1 lists the length of the longest scan segment in each version of the circuit. The number of ATPG untestable (AU) faults in the original design is 24142 and the fault coverage is 94.25%. AU faults are faults which are not testable by ATPG due to constraints placed on the ATPG run by the user. Column 2 lists the number of additional AU faults created in the broadcast scan configuration. As expected, the number of AU faults increases as more constraints are enforced during ATPG. The increased AU fault counts correspond to the drop in coverage discussed in Section 2.

Columns 3 and 4 show the number of patterns generated in the broadcast scan configuration and the fault coverage achieved by those patterns, respectively. The number of broadcast patterns is always considerably greater than the number of serial patterns in the baseline configuration. The reason for this is obvious. The increased number of constraints on the ATPG patterns decreases compaction and thus increases the number of patterns. In particular, in the broadcast mode, all scan segments can only store identical patterns. Thus, for example, two faults that could have been detected by the same serial pattern now may require two distinct broadcast patterns.

As the number of scan segments is increased, the number of broadcast patterns decreases for a while. This situation is somewhat counterintuitive. There are two effects at work that pull the number of patterns in opposite directions. Less compaction makes the number of patterns go up. Countering this effect is the fact that additional constraints also make many faults untestable. Thus the number of patterns required to detect a smaller set of faults is also smaller. Depending on the circuit and the compaction algorithms used, the number of broadcast patterns may decrease or increase with more scan segments. We see both of these effects in our experiments.

Following the constrained ATPG in the broadcast configuration, an unconstrained (serial scan) ATPG is performed to generate patterns for the remaining faults. The scan chain length for serial scan in this design is 9351 D-type scannable flip-flops. The numbers of serial scan patterns are listed in Column 5, and these patterns are necessary to detect faults undetectable in the broadcast configuration. The fault

	Broadcast Scan (Pre-RFS)				Serial Scan		Broadcast Scan (Post-RFS)	
Col No.	1	2	3	4	5	6	7	8
Config	Max Len	Add AU	Pats	FC%	Pats	FC%	Pats	FC%
Baseline	9,351	0	0	0.00	910	94.25	0	0.00
Div4	2,338	74	20,815	94.23	13	56.17	16,245	38.08
Div8	1,169	250	17,899	94.19	14	60.24	13,368	34.01
Div12	780	239	16,430	94.19	20	63.64	11,824	30.61
Div16	585	733	15,370	94.08	53	69.68	9,677	24.57
Div20	468	582	14,459	94.11	31	65.95	10,060	28.30
Div24	390	692	13,741	94.09	38	67.46	9,454	26.79
Div28	334	1,220	11,825	91.58	166	76.02	6,627	18.23
Div32	293	1,565	12,337	93.87	59	73.02	7,928	21.23

Table 1: A table showing ATPG patterns and fault coverage for various ILS configurations.

coverage reported for the serial scan patterns is obtained during reverse/random fault simulation (RFS). As fault coverage decreases for the broadcast scan patterns when the scan chain is divided into more segments, the number of serial scan patterns increases to test more faults.

This result demonstrates that the serial scan patterns are actually capable of detecting increasingly more faults. By performing reverse/random fault simulation of the broadcast scan patterns on the remaining faults, the number of required broadcast scan patterns is significantly reduced, as is shown in Column 7. The fault coverage numbers obtained by these broadcast scan patterns, on top of serial scan patterns, are shown in Column 8. It should be reemphasized that in all different configurations of the Illinois Scan Architecture the total fault coverage is the same, namely, 94.25% in this circuit.

A close examination of both sets of patterns shows that while the number of serial scan patterns is small, those patterns are very effective in detecting easily testable faults. A normal full scan ATPG, described as baseline in this paper, would then spend a long time generating patterns to test the remaining faults. Although the broadcast scan implementation requires many more patterns to cover the remaining faults, the total scan data is reduced since the length of the broadcast patterns is a small fraction of the length of the serial patterns. Table 2 compares the memory requirements of several Illinois Scan configurations to that of the conventional full scan implementation.

Column 2 in Table 2 shows the number of serial scan patterns generated for various ILS configurations. Column 3 gives the total memory required to store this test stimuli. This data does not include the memory required to store the test responses. Column 4 shows the chain length for the broadcast scan configuration and Column 5 shows the number of patterns generated to achieve maximum fault coverage for this design. Column 6 shows the total memory requirement to store all broadcast scan patterns. Column 7 shows the total memory requirements to store both broadcast scan and serial scan patterns. Column 8 shows the percentage savings versus the 8.5 Mb of scan data required for the conventional full scan design. Since the test application time is very similar to the test storage - it is one shift cycle for each test data bit - we did not present it here in the table.

As shown from the experimental data, ILS configurations with a small number of divisions did not reduce memory requirements for all test patterns. In fact, a low division of scan chains actually increased the memory requirements in some cases due to the higher number of patterns. The benefit of ILS increases as the number of divisions increases. For example, if the scan chain were divided into 32 segments, it would require 59 serial scan patterns and 7928 broadcast scan patterns totaling about 2.87 Mb of scan data. That is a 66.2% savings versus the conventional full scan configuration. As the amount of scan data decreases, the total test application time will also decrease. Thus the ILS implementation is shown to decrease both scan memory requirements and test application time.

From the above results it is clear that one could have obtained very different memory savings for the above configurations by increasing the number of serial scan patterns beyond the minimum required. For example, in the Div4 configuration, it is clear that any number of broadcast patterns greater than 4 times the original full scan serial pattern will always require more memory. That number for Div4 is 3640 pat-

	Serial Scan			Broadcast Scan			Total	% Savings
Col No.	1	2	3	4	5	6	7	8
Config	Max Len	Pats	Megabits	Max Len	Pats	Megabits	Megabits	
Baseline	9351	910	8.51	9,351	0	0	8.51	0.0
Div4	9351	13	0.12	2,338	16,245	37.98	38.10	-347.8
Div8	9351	14	0.13	1,169	13,368	15.63	15.76	-85.2
Div12	9351	20	0.19	780	11,824	9.22	9.41	-10.6
Div16	9351	53	0.50	585	9,677	5.66	6.16	27.6
Div20	9351	31	0.29	468	10,060	4.71	5.00	41.3
Div24	9351	38	0.36	390	9,454	3.69	4.04	52.5
Div28	9351	166	1.55	334	6,627	2.21	3.77	55.7
Div32	9351	59	0.55	293	7,928	2.32	2.87	66.2

Table 2: A table showing tester memory usage for various ILS configurations.

terns. By restricting the ATPG run for the broadcast mode to about 2500 patterns we still leave room for about 285 serial scan patterns without going over the original memory requirements. Of course, there is no guarantee that one would not need more than 285 serial patterns to complete the fault coverage. While we cannot predict the optimum split between broadcast and serial patterns, it is clear that some simple guidelines can be followed during the ATPG run.

Let the number of full scan baseline patterns be F. Then designate an ILS configuration with n-way division of the baseline scan as Div(n). Let the number of broadcast patterns generated be B and the number of serial patterns generated to obtain complete fault coverage be S. Then for ILS to have smaller tester memory bits than the convention full scan memory bits, the following relationship must be satisfied.

$$S + B/n < F \quad (1)$$

Let us say the maximum number of serial patterns S is 1/5th of original serial patterns F. Then the relationship reduces to

$$B < 4nF/5 \quad (2)$$

Thus the initial ATPG run in the broadcast mode must be restricted to generating a maximum of $4nF/5$ patterns. For example in our circuit we should have restricted the number of broadcast patterns for Div4, Div8, Div12 and Div16 to 2912, 5824, 8736 and 11648 respectively. In each case, we have room for 910/5 = 182 serial scan patterns without going over the original memory requirements. As mentioned earlier, there is no guarantee that a complete fault coverage will be obtained in each case with a ceiling of 182 serial scan patterns. In this study we did not run these experiments with the above guidelines, but instead we completed the fault coverage in every case.

4.2 Multiple Scan Chain Configuration

In the preceding example we assumed that there was only one scan chain in the design. It is much more common for designs to have several scan chains each with separate scan-in and scan-out pins. In the following experiments, we studied the effectiveness of the ILS technique in a design with multiple scan chains. We utilize the same design used in the preceding sections.

Although a typical goal during scan insertion is to implement balanced scan chains, the 8 scan chains are unbalanced in this particular circuit in our experiment. The longest chain contains 1883 flip-flops and the shortest chain only 213. For ILS implementation, these 8 parallel chains are divided into several segments depending on the length of the longest scan chain, similar to the diagram shown in Figure 2 (c). Each original chain has its own broadcast organization. Therefore in this study we have eight broadcast groups each with its own scan-in pin.

Table 3 gives the results for the multiple scan chain configuration. The chain division listed on Column 1 of Table 3 is the broadcast configuration of the longest chain; all the other chains are divided into N segments of M flip-flops when necessary, where N is the minimum number of divisions that allows M to be less than or equal to the segment length of the longest chain. The column labels and contents in Table 3 correspond exactly to the labels and contents in Table 1. As we have shown in the methodology description, a side benefit of the ILS technique is that the data efficiencies for the scan patterns are greatly improved in a design with unbalanced scan chains.

	Broadcast Scan (Pre-RFS)				Serial Scan		Broadcast Scan (Post-RFS)	
Col No.	1	2	3	4	5	6	7	8
Config	Max Len	Add AU	Pats	FC%	Pats	FC%	Pats	FC%
Baseline	1,883	0	0	0.00	910	94.25	0	0.00
Div4	471	172	16,929	94.21	16	62.19	11,484	32.06
Div8	236	778	15,649	94.07	37	67.65	9,904	26.60
Div12	157	1,797	14,056	93.82	131	73.57	8,268	20.68
Div16	118	2,355	12,787	93.69	86	74.33	7,543	19.92
Div20	95	5,334	11,233	92.98	123	75.34	6,560	18.91
Div24	79	4,060	10,755	93.25	165	77.36	6,290	16.89
Div28	68	4,075	9,985	93.17	193	77.98	5,834	16.27
Div32	59	6,129	9,383	92.79	174	79.79	5,430	14.46

Table 3: A table showing ATPG patterns and fault coverage for various ILS configurations using eight broadcast groups.

	Serial Scan			Broadcast Scan			Summary		
Col No.	1	2	3	4	5	6	7	8	9
Config	Max Len	Pats	Megabits	Max Len	Pats	Megabits	Megabits	% Savings	% Data Efficiency
Baseline	1,883	910	13.71	1,883	0	0	13.71	0.0	62.1
Div4	1,883	16	0.24	471	11,484	43.27	43.51	-217.4	87.7
Div8	1,883	37	0.56	236	9,904	18.70	19.26	-40.5	93.3
Div12	1,883	131	1.97	157	8,268	10.38	12.36	9.8	90.6
Div16	1,883	86	1.30	118	7,543	7.12	8.42	38.6	93.5
Div20	1,883	123	1.85	95	6,560	4.99	6.84	50.1	93.7
Div24	1,883	165	2.49	79	6,290	3.98	6.46	52.9	95.3
Div28	1,883	193	2.91	68	5,834	3.17	6.08	55.6	96.1
Div32	1,883	174	2.62	59	5,430	2.56	5.18	62.2	97.2

Table 4: A table showing test memory usage for various ILS configurations using eight broadcast groups.

Similar to the results shown for the single chain design, the number of additional AU faults caused by the constraints increases as the chains are divided into more segments. The number of raw patterns and the fault coverage decrease as the amount of broadcast constraints increases. Again, the few serial scan patterns, while necessary, were able to detect a majority of faults in the circuit. The number of broadcast scan patterns was reduced after reverse fault simulation and these are applied to maximize the fault coverage after the application of serial scan patterns. Table 4 shows the memory usage of the ILS implementations compared to the baseline configuration. The memory usage for the conventional full scan (baseline) implementation consists of only serial scan patterns. The memory bits are computed as maximum chain length times number of patterns times the number of broadcast groups. Columns 1, 2, and 3 in Table 4 give the scan chain length, the number of scan patterns generated, and the total number of memory bits for the serial scan case.

Next, we move to the broadcast scan calculation. In this design, the number of broadcast groups is eight. Columns 4, 5, and 6 in the table correspond to columns 1, 2, 3, except for broadcast scan mode. Finally, columns 7, 8, and 9 give the total bits for serial scan plus broadcast scan, the percent savings realized versus the baseline case, and data efficiency per our definition in Section 3, respectively.

The memory usages for ILS implementations with low divisions are dominated by broadcast scan patterns. The benefit of the ILS technique becomes apparent when the length of the longest scan segment decreases. For example, the Div32 ILS configuration provided 62.2% saving on the total memory requirement compared to that of the conventional full scan implementation. Again, with the reduction in total data volume, the total test application time would also be significantly reduced. The total shift cycles can be estimated as (total bits)/(8 broadcast groups).

Column 9 shows the improvement of data efficiency due to the implementation of ILS. As we increase the parallelism in the design and effectively shorten the longer scan chains, the amount of 'X' values padded to the end of the shorter chains is drastically reduced. As shown in the table, the data efficiency quickly improves from 62.1% to the high 90's percentile.

If the chains were balanced in the original circuit, the test data volume for the baseline entry in Table 4 would be 910 scan patterns times 9351, the total number of flips-flops, which is 8.5 million data bits. This data volume is still higher than the data volumes for the unbalanced ILS versions Div20 thru Div32 in Table 4. If the ILS implementation was on the balanced chain circuit, the data volume would be even less. We did not perform such a simulation but one can infer the worst case volume from Table 2. Therefore, we can conclude that the ILS is effective in reducing test data volume in multiple scan circuits with or without balanced scan chains.

4.3 Hardware Overhead for ILS

When evaluating any new DFT technique, one important consideration is the amount of area overhead inherent in the methodology. If we examine the ILS overhead in the context of our study circuit, in the largest experiment we have 32 muxes and 32 stages in the MISR. If we assume pessimistically four gates/mux and four more gates/MISR stage, then the total overhead would be 256 more gates, or about 0.16% overhead in terms of gate count. Of course there would be additional routing overhead as well.

To account for routing overhead, we could make a pessimistic assumption that it would be as large at the gate overhead, then the total design area impact would be approximately 0.32%, which we believe is an acceptable result. This result compares favorably with recently published industrial case studies of conventional BIST insertion [6] and also deterministic BIST results [7], [8].

4.4 Transition Fault ATPG with ILS

As circuit timing becomes more and more critical in designs, circuit delay testing becomes crucial in screening speed related defects. Thus, the effectiveness of ILS in reducing test time and data volume for transition faults was evaluated in this experiment. Table 5 gives the ATPG results for the same multiple scan chain design as in the previous section. We employ the ATPG's default test generation strategy, which is the "launch from shift" method [15]. In launch from shift, the second pattern in the transition fault test sequence is derived as a one bit shift of the first pattern. We are also not forcing the ATPG to hold primary input pins constant between the first and second patterns in the pattern pair. As for stuck-at fault ATPG, reverse/random simulations were performed to remove redundant broadcast scan patterns.

Columns 1 through 4 give the chain length, pattern count, total memory bits consumed, and the transition fault coverage for the serial scan configuration. Columns 5 through 8 give those same results for broadcast scan configurations. Columns 9 and 10 give the summed bit counts and percent savings versus serial scan alone, respectively.

Note that the number of these serial scan patterns is significantly larger than that of the stuck-at fault patterns, while the number of broadcast scan patterns stays roughly the same. The reason is that transition fault detection requires a stringent fault activation and propagation sequence that may be harder to meet in an ILS implementation. However, the results shows that the smaller size of the scan segments in ILS is able to greatly reduce test time and data volume of transition fault test patterns by the same magnitude as for the stuck-at fault patterns, as shown in the last column.

5 Multiple Clock Domains

The example circuit used in this paper is an ideal learning example because it contains a single clock domain. However, it is commonplace in large ASIC and semicustom designs to have circuits containing multiple, sometimes dozens of clock domains. One might question the applicability of ILS to these more complex designs.

It is the experience of the authors that due to tester limitations or the necessity to control timing complexity, ASIC designs containing a large number of clock domains may be required to "combine" clock domains during test mode to achieve a relatively small number of test clocks. So, it is frequently the case that the number of clock domains during test mode is

	Serial Scan				Broadcast Scan				Summary	
Col No.	1	2	3	4	5	6	7	8	9	10
Config	Max Len	Pats	Megabits	FC%	Max Len	Pats	Megabits	FC%	MegaBits	% Savings
Baseline	1,883	3,347	50.42	93.08	1,883	0	0	0.00	50.42	0.0
Div4	1,883	1,071	16.13	71.62	471	11,557	43.55	21.46	59.68	-18.4
Div8	1,883	1,300	19.58	76.14	236	10,844	20.47	16.94	40.06	20.6
Div12	1,883	1,393	20.98	79.32	157	9,518	11.95	13.76	32.94	34.7
Div16	1,883	1,479	22.28	80.83	118	8,331	7.86	12.25	30.14	40.2
Div20	1,883	1,897	28.58	83.21	95	6,300	4.79	9.87	33.36	33.8
Div24	1,883	1,729	26.05	81.27	79	7,013	4.43	11.81	30.48	39.6
Div28	1,883	1,313	19.78	77.98	68	5,855	3.19	15.10	22.96	54.5
Div32	1,883	1,205	18.15	79.79	59	5,038	2.38	13.29	20.53	59.3

Table 5: Transition fault ATPG results for various configurations of ILS with eight broadcast groups.

a small subset of the total number of clock domains in the design. Regardless of whether ILS is used or not, designers must still perform test insertion and clock tree synthesis to have a working scan solution in test mode.

Once this task has been accomplished, ILS may still be applied to the reduced number of clock domains. In this scenario, there is the added constraint that the segment lengths in each of the ILS-treated clock domains should be made as close to balanced as possible. The need for balance is purely for tester time minimization, and is simply a guideline, not a requirement.

6 Test Tool Implications

In order to reap the full benefits of an ILS implementation and to simplify the overall design methodology, design tools would have to automate the use of ILS. The modifications to the scan insertion process are relatively minor since muxing logic is a frequent need during scan insertion. The synthesis of the MISR is a fairly straightforward task.

The changes to ATPG, however, are more radical. ATPG would need to be able to perform all of its design for test rule checking, scan chain tracing, and other functions in the presence of the somewhat more complex ILS logic. Furthermore, the fault ordering/targeting algorithm would probably be impacted in order to target faults for test generation in an efficient manner while still taking advantage of the broadcast mode.

There are no current commercial ATPG tools that can cope with ILS completely today. Our experimentation was merely a modeling of ILS implementations which could be accomplished with ATPG constraints which simulate the presence of ILS in the circuit. The presence of these constraints slowed the ATPG process considerably. As an example, the full run time including dynamic and static compression for stuck-at ATPG for the baseline case was approximately 1700 CPU seconds on an UltraSPARC^{TM} 10 class machine. The Div32 run time for the same test case was nearly 16,000 CPU seconds on the same machine.

ATPG runs with ILS constraints also produced less compacted test sets. If the ATPG was run with the ILS implementation reflected in the netlist, it should have produced even more compact patterns, resulting in substantial improvement in the volume reduction than in the results we report here.

In summary, we believe that direct support of ILS in commercial ATPG tools would not incur such a large run time penalty and would result in better compaction. However, much work remains before ILS could become a commercially supported mainstream DFT method.

7 Conclusions

As design sizes grow ever larger, the tester memory and test application time often become bottlenecks in the manufacturing process. It is becoming more common to need multiple tester loads to apply all test patterns to test a device with a conventional full scan approach. The proposed Illinois Scan Architecture provides a mechanism to reduce test application time and data storage requirements. The hardware overhead required to implement this technique is a negligibly small fraction of the overall circuit size. By applying identical scan patterns to multiple scan

chain segments simultaneously, we were able to reduce test data volume and test application time. As an added side benefit, the data efficiency for multiple scan chain designs has also been improved. The experimental results have shown the effectiveness of ILS in reducing test data volume while maintaining high fault coverage.

All previous ILS published data has been reported for circuits of 40K gates and 1500 scan flip-flops or less. The circuit in this case study thus represents the largest circuit on which ILS has been studied to date. When compared to typical ASICs being designed today, this design might be considered modest in size. However, the ILS methodology is general enough to be applicable to circuits of virtually any size, and the results realized with ILS are not expected to be diminished with increasing gate count.

8 References

[1] S. Y. Lee and K. K. Saluja, "An algorithm to reduce test application time in full scan designs," in *Dig. Tech. Papers, 1992 IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1992, pp. 17–20.

[2] S. Narayanan and M. A. Breuer, "Reconfiguration techniques for a single scan chain," *IEEE Trans. Computer-Aided Design*, vol. CAD-14, no. 6, pp. 750–765, June 1995.

[3] Y. Zorian, "Testing the monster chip," *IEEE Spectrum*, vol. 36, no. 7, pp. 55–60, Jul. 1999.

[4] K. M. Butler, "A study of test quality/tester scan memory trade-offs using the SEMATECH test methods data," in *Proc. 1999 IEEE Int. Test Conf.*, Sept. 1999, pp. 839–847.

[5] R. Dorsch and H-J. Wunderlich, "Accumulator based determinstic BIST," in *Proc. 1998 IEEE Int. Test Conf.*, Oct. 1998, pp. 412–421.

[6] G. Hetherington, T. Fryars, N. Tamarapalli, M. Kassab, A. Hassan and J. Rajski, "Logic BIST for large industrial designs: Real issues and case studies," in *Proc. 1999 IEEE Int. Test Conf.*, Sep. 1999, pp. 358–367.

[7] G. Kiefer, H. Vranken, E. J. Marinissen and H-J. Wunderlich, "Applications of deterministic logic BIST on industrial circuits," in *Proc. 2000 IEEE Int. Test Conf.*, Oct. 2000, pp. 105–114.

[8] D. Das and N. Touba, "Reducing test data volume using external/LBIST hybrid test patterns," in *Proc. 2000 IEEE Int. Test Conf.*, Oct. 2000, pp. 115–122.

[9] K-T. Cheng, S. Devadas and K. Keutzer, "Robust delay-fault test generation and synthesis for testability under standard scan design methodology," *IEEE Trans. Computer-Aided Design*, vol. CAD-12, no. 8, pp. 1217–1231, Aug. 1993.

[10] A. Jas and N. A. Touba, "Test vector decompression via cyclical scan chains and its application to core-based designs," in *Proc. 1998 IEEE Int. Test Conf.*, Oct. 1998, pp. 458–464.

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

[12] I. Hamzaoglu and J. H. Patel, "Reducing test application time for full scan embedded cores," in *Dig. Papers, 29th Int. Symp. Fault-Tolerant Comp.*, June 1999, pp. 260–267.

[13] P. H. Bardell and W. H. McAnney, "Self-testing of multichip logic modules," in *Proc. 1982 IEEE Int. Test Conf.*, Nov. 1982, pp. 200–204.

[14] K-J. Lee, J-J. Chen and C-H. Huang, "Using a single input to support multiple scan chains," in *Dig. Tech. Papers, 1998 IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1998, pp. 74–78.

[15] J. A. Waicukauski, E. Lindbloom, B. Rosen and V. Iyengar, "Transition fault simulation," *IEEE Design and Test of Comput.*, vol. 4, no. 2, pp. 32–38, Apr. 1987.