Double-Tree Scan: A Novel Low-Power Scan-Path Architecture

Bhargab B. Bhattacharya

ACM Unit
Indian Statistical Institute
Calcutta – 700 108, India
bhargab@isical.ac.in

Sharad C. Seth and Sheng Zhang

Department of Computer Sci. and Eng.
University of Nebraska-Lincoln
Lincoln, NE 68588-0115, USA
{seth, szhang}@cse.unl.edu

Abstract

In a scan-based system with a large number of flip-flops, a major component of power is consumed during scan-shift and clocking operation in test mode. In this paper, a novel scan-path architecture called double-tree scan (DTS) is proposed that drastically reduces the scan-shift and clock activity during testing. The inherent combinatorial properties of double-tree structure are employed to design the scan architecture, clock gating logic, and a simple shift controller. The design is independent of the structure of the circuit-under-test (CUT) or its test set. It provides a significant reduction both in instantaneous and average power needed for clocking and scan-shifting. The architecture fits well to built-in self-test (BIST) scheme under random testing, as well as to deterministic test environment.

1. Introduction

With the emergence of mobile devices, design of low-power VLSI systems has become a major concern in circuit synthesis. A significant component of the power consumed in CMOS circuits is caused by the *switching activity* (SA) at various circuit nodes during operation. The dynamic power consumed at a circuit node is proportional to the total number of $0 \rightarrow 1$ and $1 \rightarrow 0$ transitions that the logic signal undergoes at that node multiplied by its capacitance and the frequency of operation.

Power/energy minimization during testing has become important in the context of deep sub-micron technology because of higher device densities and clock rates. In a scan-based system, a significant amount of power is consumed during the scan operations, as the activity in the scan-path, clock tree, and in the CUT becomes very high [1]. The average-power optimization extends the battery life in mobile applications. Maximum sustained power over a specified limit, may cause excessive heating of the device, whereas, the instantaneous power may cause excessive (inductive) voltage drop in the power and ground lines because of current swing. Thus, the logic states at circuit nodes may erroneously change. Further, BIST schemes with random test patterns may need an excessive amount of energy because of longer test length.

2. Background

Existing power/energy minimization techniques include test scheduling [2], toggle suppression and blocking useless patterns [13], designing low-power TPG for BIST applications [8, 12], use of Golomb coding for scan testing [7], and power-aware ATPG [14, 15]. For deterministic testing, power reduction can be achieved by reordering scan chains and test vectors [4]. Compaction of test vectors for low power in a scan-based system was addressed in [3]. For minimizing switching activity during scan-shift, multiple scan-path architectures with selective freezing have been reported recently [6, 8, 10]. Various other scan design techniques for power minimization have also been proposed [16, 17, 18]. Further issues involving low-power gated clock design have been studied in [11].

The DTS architecture is motivated by the observation that in typical designs with large number of scan flip flops, a major component of energy is consumed in scan-shifting and clocking of the scan chain. We focus attention on designs with a single serial scan input and output. The following properties are desirable in designing such a scan-path and its associated shift control mechanism:

(1) It should be possible to load or unload the scan-path with f FFs in f shift clock cycles;
(2) It should be possible to completely overlap the loading of a new test vector with the unloading of the previous response;
(3) The scan control mechanism should be simple and it should not increase the shifting time;
(4) The design should be independent of the structure of the CUT and its test set;
(5) The input order of values shifted in, may also be preserved at the shifted output.

The simplest architecture with these properties is the classical linear scan (*Fig. 1a*), where all the flip-flops are configured as a chain in test mode. In a linear scan chain

of length f, the total worst-case switching activity during scan-shifting in terms of number of transitions is $O(T*f^2)$, where T is the number of test vectors. Similarly, the clock activity is $O(T*f^2)$. The other extreme example is the full parallel scan (*Fig. 1b*) where, each flip-flop can be loaded and unloaded independently. Between these two options lies the multiple parallel scan-path architecture, where the scan-path is decomposed into several smaller and independent linear chains (*Fig. 1c*). Recently, test architectures based on multiple scan-paths have been reported in order to reduce the shift and clock activity [6, 9, 10] and for BIST applications [5]. During shifting/clocking, each chain can be selectively chosen while freezing the remaining others. If s is the number of linear chains, the switching activity reduces to $\{s*T*(f/s)^2\} = T*f^2/s$, i.e., by a *constant factor*. Since, the power loss (both for shifting and clocking) is quadratic in the length of the chain, a fully linear scan-path consumes maximum power, whereas, a fully parallel scan-path requires minimum power. Increasing the multiplicity of the scan-paths for a given number of flip-flops, reduces the length of the paths, and hence power demand. However, the control mechanism becomes overly complex as the multiplicity increases. For a large number of scan chains, the MUX block either introduces large fan-in/fanout, or delay. The former consumes more power and the latter increases the shifting time. Thus, the fully parallel scheme, or the multiple parallel scan-path scheme beyond a certain value of multiplicity, is impractical. For example, a multiple scan-path with only 3 linear chains was considered by Saxena, Butler, and Whetsel [9]. Nicolici and Al-Hashimi [6] used at most 7 chains, and Sinanoglu and Orailoglu [10] studied up to 24 linear chains. Moreover, the design methods in [6, 10] are strongly dependent either on the structure of the CUT, or on the test set itself.

It may be noted that some of the other well known scan architectures that do not have power minimization as their main goal allow multiple scan chains to operate in parallel, e.g. the STUMPS architecture uses an LFSR/phase shifter (MISR) to load (unload) multiple chains in parallel, and the shared scan-in (Illinois scan) architecture allows parallel scan chains to be loaded with *identical* bit streams [19, 20]. These schemes are not directly comparable to DTS, although the scan chains they employ can be implemented with the DTS to reduce power consumption.

3. Main Results

This paper presents a novel scan-path architecture called *double-tree scan* (DTS) for low-power test applications. The structure resembles two trees glued at the leaf nodes. We report on the design methodology of DTS, clock gating logic, and shift controller. The architecture is independent of the CUT and its test set. The scheme reduces both the scan-shift and clock activity. The design is simple and hardware overhead is low. It provides very significant amount of power/energy savings over various existing schemes, and the benefit becomes more prominent as the number of scan flip-flops increases.

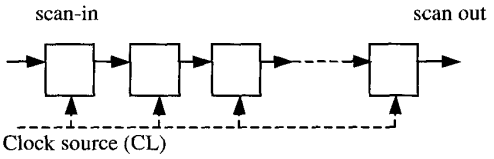

Figure 1a: Linear scan-path

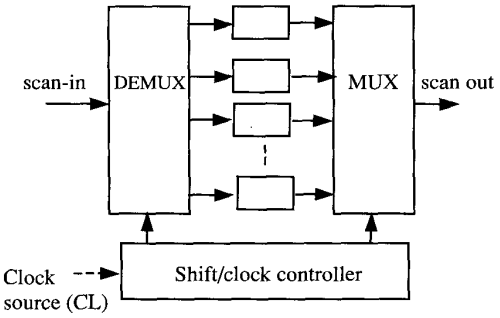

Figure 1b: Fully parallel scan-path

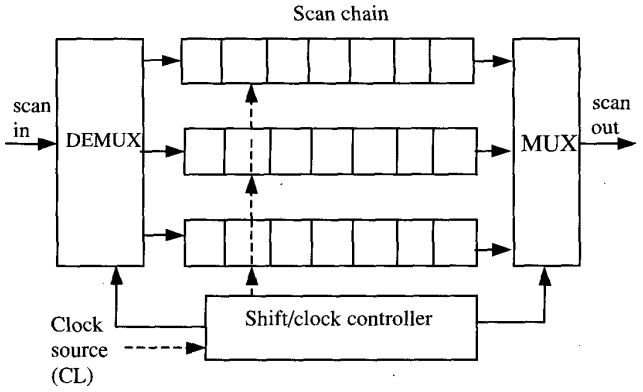

Figure 1c: Multiple parallel scan-path

4. Double-Tree Scan (DTS) Structure

We organize the scan flip-flops in a radically different fashion called *double-tree scan (DTS)* architecture (*Figs. 2a, 2b, 2c*), to achieve drastic power/energy reduction. A complete binary tree of level k (considering the root at level 0) consists of 2^k leaf nodes and $2^k - 1$ internal nodes. The proposed scan structure resembles two complete k-level binary trees whose leaf nodes are merged pair-wise. Thus, a *full double-tree* DTS(k) consists of $N = 3*2^k - 2$

(= $2^k - 1 + 2^k + 2^k - 1$) nodes. Each node of the tree represents a scan flip-flop. All edges in the tree are directed from top to bottom. A directed edge (i, j) in the DTS indicates that Q(i), i.e., the Q-output of the flip-flop i drives D(j), the D-input of the flip-flop j. For each node with in-degree 2 (i.e., a join node) in the bottom half of the DTS, a 2→1 multiplexer is needed to select the predecessor flip-flop during scan operation.

5. Full DTS

Examples of full DTS(k) for k = 1, 2, 3 are shown in *Figs. 2a, 2b, 2c* respectively. The topmost node (source) serves as the scan-in node, whereas the bottommost node (sink) serves as the scan-out node. The DTS structure is hierarchical in nature: DTS(k) can be constructed by taking two copies of DTS(k-1) and adding two more nodes as a new source and sink with corresponding edge connections. Thus, a full DTS(k) has $(3*2^k - 2)$ flip-flops organized as 2^k (overlapping) scan-paths, each of (2k + 1) nodes from the source to the sink (*Table 1*). For example, a full DTS(10) consists of 3070 FFs organized as 1024 overlapping scan-paths, each of length 21. In a conventional multiple parallel scan-path scheme (*Fig. 1c*) with the same number of FFs organized as 1024 independent linear chains, the length of each chain and consequently the power/energy loss in the scan-path would have been much smaller. However, a 1024-input MUX block lumped at the scan output will need very large fan-in/fan-out (causing additional power loss), or introduce large delay (causing increase of shifting time).

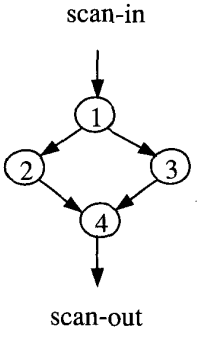

Figure 2a: DTS(1); # FFs (f) = 4

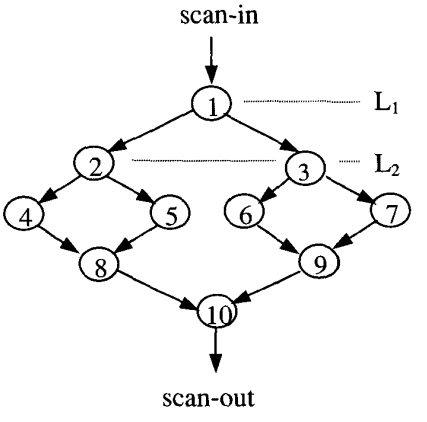

Figure 2b: DTS(2); #FFs (f) = 10

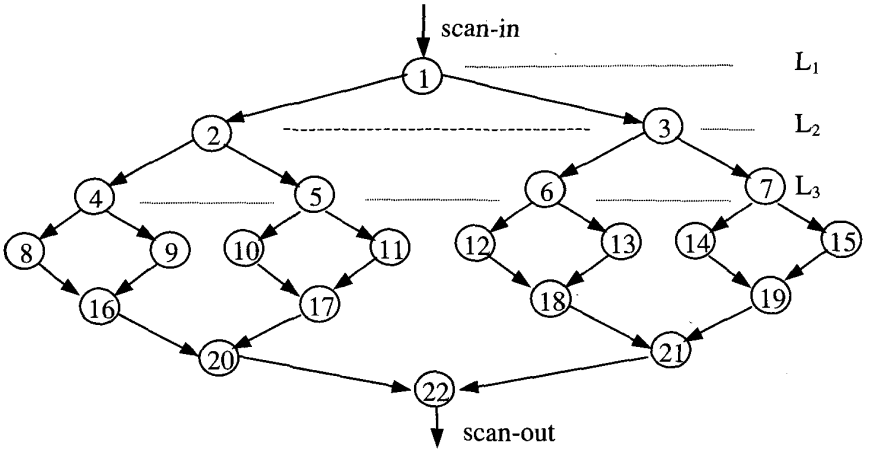

Figure 2c: DTS(3); # FFs (f) = 22

5.1 Clock Controller

During scan-shift, one should be able to select a single path from the source to sink in the DTS for loading and unloading the FFs. Paths in a DTS(k) may be designated by a k-bit vector $L_1, L_2, .. L_k$, where the path number is simply the binary number corresponding to this vector. Thus, in DTS(2) of *Fig. 2b*, we have:

$L_1 L_2 = 00 \Rightarrow$ Path-0: $1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 10$;

$L_1 L_2 = 01 \Rightarrow$ Path-1: $1 \rightarrow 2 \rightarrow 5 \rightarrow 8 \rightarrow 10$;

$L_1 L_2 = 10 \Rightarrow$ Path-2: $1 \rightarrow 3 \rightarrow 6 \rightarrow 9 \rightarrow 10$;

$L_1 L_2 = 11 \Rightarrow$ Path-3: $1 \rightarrow 3 \rightarrow 7 \rightarrow 9 \rightarrow 10$.

Table 1: Full DTS for some values of k

k	# FFs in full DTS(k)	Number of scan-paths	Length of each scan-path
0	1	1	1
1	4	2	3
2	10	4	5
3	22	8	7
4	46	16	9
5	94	32	11
6	190	64	13
7	382	128	15
8	766	256	17
9	1534	512	19
10	3070	1024	21

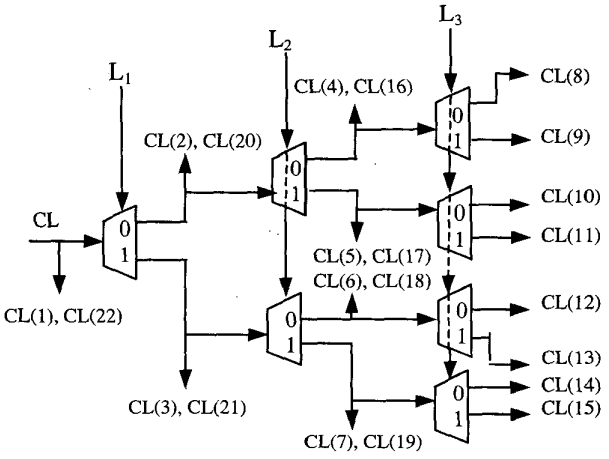

Figure 3: A naive clock gating logic for DTS(3); # DEMUX = 7

5.1.1 A Naive Design

A straightforward way to design a clock gating logic for DTS(k) would be to use k control lines $L_1, L_2, .. L_k$, and a tree circuit consisting of 2^k-1 (1→ 2) DEMUX units for clock routing mechanism. For each node of DTS with out-degree 2 (i.e., a fork node) in the top half of the DTS, a 1→2 demultiplexer is needed to route the clock to an appropriate successor flip-flop. For example, a clock control circuit of DTS(3) of *Fig. 2c* can be designed as shown in *Fig. 3*, where, CL(i) denotes the clock signal to the flip-flop i. At any instant of time during scan-shift, only a single path from the source to sink is activated by allowing the clock signal to reach to the flip-flops along the path. Clocks to other flip-flops in the remaining paths are frozen. For each of the join nodes in the bottom up of the tree, a 2→1 multiplexer is needed, which can be controlled accordingly. Thus, the architecture reduces both the shift and clock activity simultaneously. The number of MUX units needed for join nodes of the bottom half of the tree is 2^k-1. Hence, for a DTS(k) with $(3*2^k - 2)$ FFs, the additional hardware overhead would be a total of $(2*2^k - 2)$ DEMUX/MUX units. However, the maximum fanout of a control line would be 2^{k-1}, which may not be acceptable for a system with a very large number of scan flip-flops. For example, such a controller of DTS(10) with 3070 flip-flops would require a fanout of 512.

5.1.2 A Hierarchical Clock Controller

A better way to design the controller is to use a recursive structure, which routes the clock as well as the control signals in a hierarchical fashion. To implement this architecture, a different clock control mechanism is needed. For DTS(2) of *Fig. 2b*, the controller is shown in *Fig. 4*. To activate scan-shift along the path 1→ 2→ 5→ 8→ 10, we set the control lines L1 = 0 and L2 = 1. Thus, the DEMUX tree enables CL(1), CL(2), CL(5), CL(8), and CL(10). The MUX control lines for the flip-flops 8 and 10 are also set accordingly.

In this method, DTS(k+1) can be constructed recursively by using two copies of DTS(k), and combining them by adding (k+1) new (1→2) DEMUX blocks. The design for DTS(3) is shown in *Fig. 5*. Thus, if D(k) denotes the number of DEMUX units to realize DTS(k), then D(k+1) = 2*D(k) + (k+1). Solving this recurrence yields, D(k) = 2^{k+1} – (k+2). The maximum fanout of a control line for DTS(k) will be only k, instead of 2^{k-1} as in the earlier design. Thus, DTS(10) with 3070 FFs would need a maximum fanout of 10. As before, the number of MUX units needed for join nodes of the bottom half of the tree is 2^k-1. Hence, for a DTS(k) with $(3*2^k - 2)$ FFs, the additional hardware overhead would be a total of $(3*2^k - k - 3)$ DEMUX/MUX units.

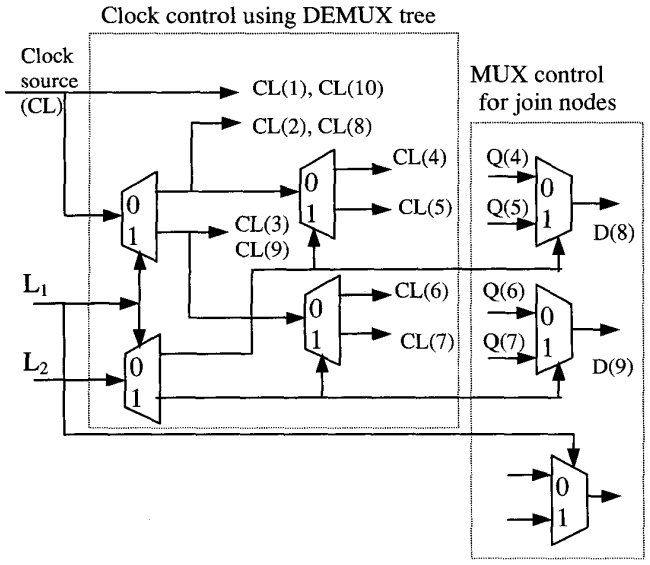

Figure 4: A hierarchical clock gating logic for DTS(2)

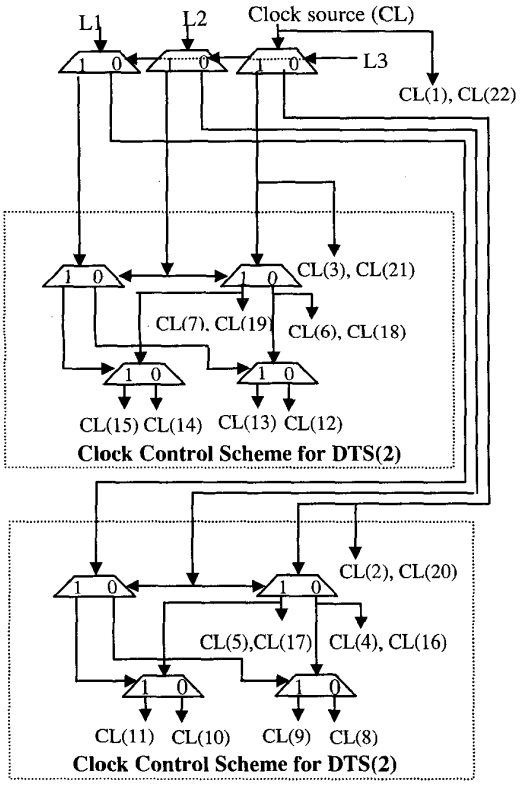

Figure 5: A hierarchical clock control logic for DTS(3) using 11DEMUX units; MUX units for join nodes are not shown.

A hierarchical controller of DTS(k) has several advantages. First, the maximum fanout of a control line is k. Second, it takes care of driving clock signals to all the scan flip-flops, and further, each output of a DEMUX unit drives at most two signals. Hence, it obviates the need for a separate clock tree for buffering. Third, for each shift clock, only a single path of length k is activated through the interior of the clock tree, and hence, the additional power loss in the tree is low. Lastly, the MUX units that are needed for the join nodes of the bottom half of DTS tree to merge them to a single scan output, are interspersed with the scan flip-flops (see *Fig. 4*). Although they appear on the shift path, they only increase latency slightly, but not the throughput (i.e., they do not need to slow down the shift clock, unlike the multiple scan-path architecture of *Fig. 1c*).

5.2 Shift Controller

We now demonstrate how DTS architecture can be employed to reduce power/energy demand drastically during scan-shifting and clocking. For simplicity of illustration, we use DTS(2) with 10 flip-flops (*Fig. 6a*).

To implement the scan architecture, a clock control is needed as shown in the *Fig. 4*. Let (p10 p9 p8 p7 p6 p5 p4 p3 p2 p1) denote a 10-bit vector to be shifted in the DTS. Assume that the current contents of the FFs are Q1, Q2, ..., Q10, where Qi denotes the content of the i-th FF. There are two ways the scan-paths can be filled: depth-first load (DFL) and breadth-first load (BFL).

5.2.1 Depth-First Load (DFL)

In this scheme, a scan-path is loaded serially up to a certain depth in the tree once for all, and then the other paths are processed. The shift controller to scan-in and scan-out a complete 10-bit vector for DTS(2) is shown in *Table 2*. When the shift-in process is completed, the contents of the FFs will look like as in *Fig. 6(b)*. The shift controller can be implemented as a simple finite-state machine that drives the control lines internally during scan-shift. Hence, no external I/O pins are needed. for these control lines. The overall scan architecture is shown in *Fig. 7*. During functional operation, all the flip-flops should receive the clock signals, and hence, the DEMUX design is slightly modified as in *Fig. 8*.

5.2.2 Breadth-First Load (BFL)

From *Table 2*, it may be observed that in the earlier DFL scheme the shifted outputs do not preserve the order in which the inputs are loaded. Although it is not a necessary criterion in test mode, one can achieve this on a DTS by loading the FFs in a breadth-first fashion.

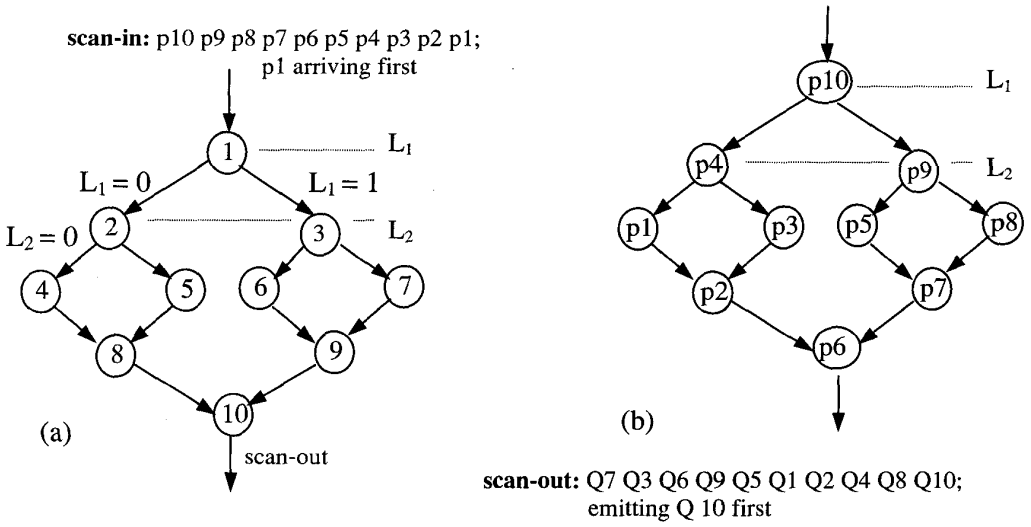

Figure 6: Scan-in and scan-out sequence for DTS(2) architecture in DFL mode

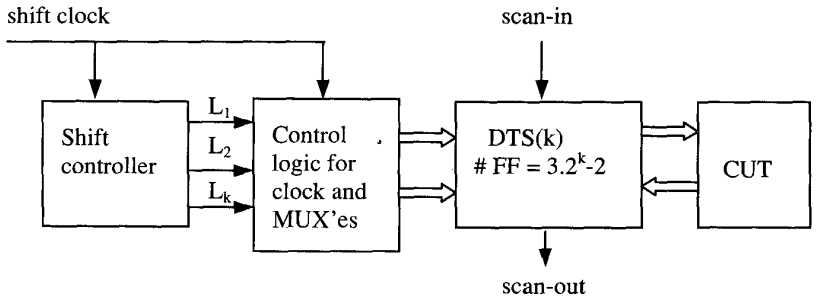

Figure 7: DTS architecture

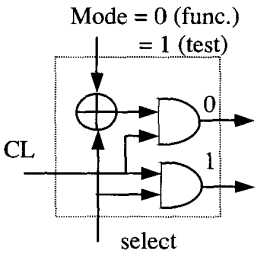

Figure 8: DEMUX

Table 2: Shift controller for the DFL mode

shift clock	L_1	L_2	active path	scanned-in bit	scanned-out bit
1	0	0	Path-0	p1	Q10
2	0	0	Path-0	p2	Q8
3	0	0	Path-0	p3	Q4
4	0	1	Path-1	p4	Q2
5	0	1	Path-1	p5	Q1
6	1	0	Path-2	p6	Q5
7	1	0	Path-2	p7	Q9
8	1	1	Path-3	p8	Q6
9	1	1	Path-3	p9	Q3
10	1	1	Path-3	p10	Q7

For a DTS(k), we run a modulo-2k counter continuously to load and unload the FFs, where the counter value is used to choose the corresponding path in the tree. For example, to load into DTS(2) with 10 FFs, we run the counter for 10 shift clocks cycling through states 0, 1, 2, 3, 0, 1, 2, 3, 0, 1. To unload from the tree (and to load the next 10-bit vector concurrently), we continue the count from the last value, i.e., run the counter for 10 cycles again as follows: 2, 3, 0, 1, 2, 3, 0, 1, 2, 3. It can be verified that the input and output order remain the same in this scheme.

The same clock control logic can be used here. Further, the shift controller can be implemented just by using a simple counter. The configuration of DTS(2) after complete shift-in is shown in *Fig. 9*.

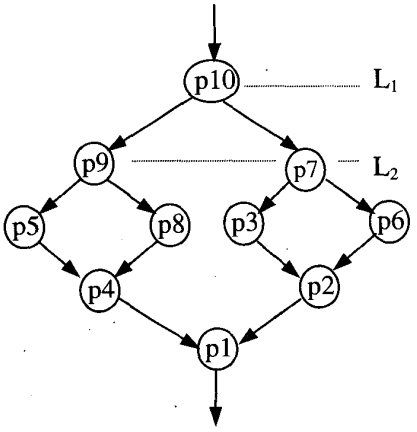

Figure 9: Loading/unloading in BFL mode;
scan-out: Q1 Q2 Q5 Q3 Q7 Q4 Q8 Q6 Q9 Q10;
emitting Q10 first

6. Arbitrary Size DTS

For a full DTS(k), the number of FFs is $(3*2^k - 2)$. In real life, if the number of flip-flops (f) in the CUT does not satisfy this equality, then we have an *incomplete DTS*. We propose two techniques to construct an incomplete DTS: (i) by pruning a full DTS of larger size, or (ii) by serial concatenation of several smaller full DTS blocks.

6.1 Pruning

We first choose a full DTS which is just larger than the given f, i.e., choose k such that $(3*2^k - 2) > f > (3*2^{k-1} - 2)$. We then delete appropriate number of nodes starting from the innermost hierarchy of the DTS(k). During the deletion process, a few additional edges may be needed to connect the pendant edges, so that the continuity of all the scan-paths is preserved.

Deleting a node: For a node with in and out degree equal to 1, we can simply delete this node and the edges connected to it. Further, if as a result of the deletion of node B, A and C become unconnected, we add an edge between A and C.

We start the pruning of DTS nodes from the innermost tree in the order of decreasing path number, and retain as many smaller complete DTS blocks as possible. The following example (*Fig. 10*) illustrates the process when the starting DTS has 10 nodes. The resulting DTS may be asymmetric in nature, and may have paths of unequal

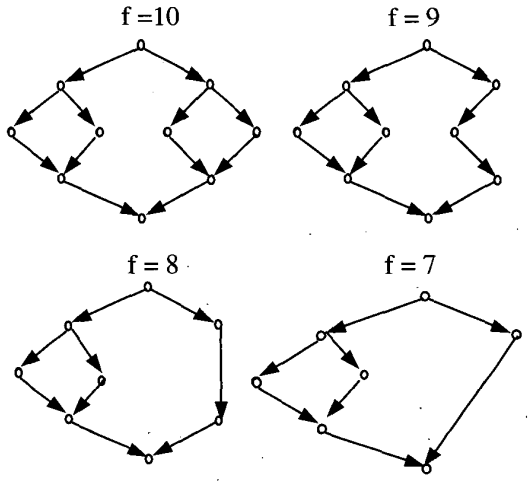

Figure 10: Designing an incomplete DTS by pruning

lengths. A more sophisticated pruning technique can be employed to preserve symmetry and equality of path lengths as much as possible.

6.2 Serial Concatenation of Full DTS Blocks

In this method, we choose a full DTS which is just smaller than the given f, i.e., choose k such that $(3*2^k - 2) < f < (3*2^{k+1} - 2)$. For the remaining nodes, we iterate the same process to obtain more DTS blocks, until their nodes add up to the given number of FF. All the full DTS blocks thus obtained are then concatenated serially.

We will demonstrate that an incomplete DTS designed by pruning is more power-efficient than the one obtained by serial concatenation technique. Since the length of the scan-paths in the pruned DTS may vary, the structure often becomes asymmetric. Thus, for overlapping load/unload operation, a DFL-based controller has to be designed. On the other hand, in the serial version, all the paths from the source to sink are of equal length, and hence, the same BFL-based shift controller can be used. Thus, its hardware implementation is simple. The following example illustrates both the techniques.

Example: Consider the circuit s35932.scan (ISCAS-89). We assume that the scan-path also includes the PI/PO for test application, and hence, its length should be #FFs + max {#PI, #PO} = 1728 + 320 = 2048.

The pruned design of a DTS with 2048 FFs is shown in *Fig. 11a*. The maximum scan-path length is 21, and the minimum is 16. The dotted links do not represent any physical connection; they indicate the pruned portions. The serial design (*Fig. 11b*) requires concatenation of DTS(9), DTS(7), DTS(5), DTS(3), DTS(2), DTS(1), one each, and two single nodes (i.e. DTS(0)). This follows

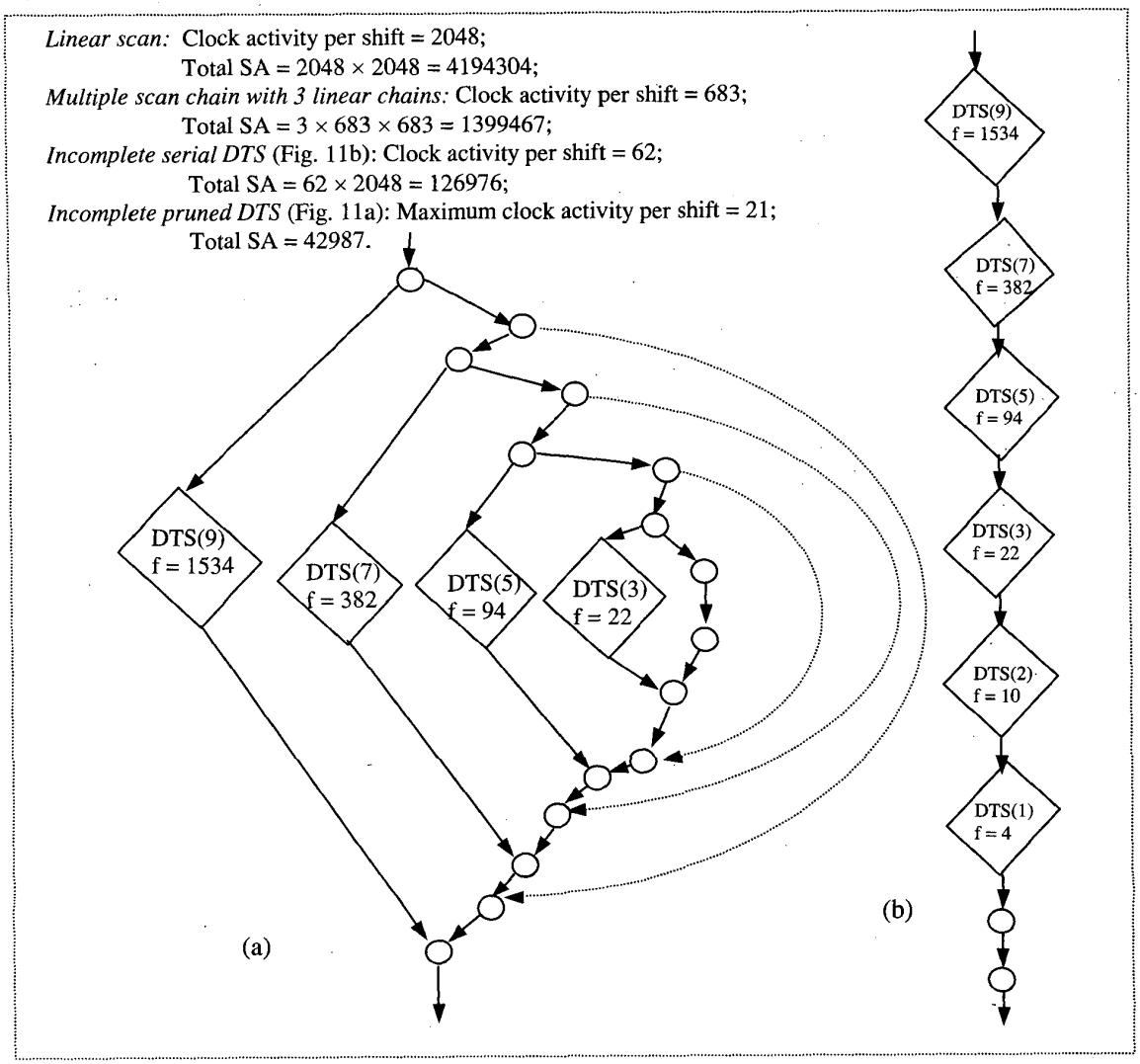

Figure 11: Incomplete DTS of 2048 FFs: (a) by pruning, (b) by serial concatenation

from the fact that 2048 = 1534 + 382 + 94 + 22 + 10 + 4 + 1 + 1 (see *Table 1*). All paths in the structure from scan-in to scan-out are of equal length (19 + 15 + 11 + 7 + 5 + 3 + 1 + 1 = 62).

Remark: A DTS architecture with DFL-based shift controller satisfies the first 4 desirable properties of scan-path design as stated in Section 4. A BFL-based implementation satisfies all of them.

6.3 Power/Energy Savings during Shifting and Clocking

It is easy to show that in a DTS architecture, for a test length T and a scan chain with f flip-flops, the total shift as well as the clock activity is $O(T*f*\lceil \log f \rceil)$ including the additional energy consumption in the control circuit. Thus, the percentile energy savings of DTS for both shifting and clocking over a linear scan chain is $(1 - (\lceil \log f \rceil / f)$, which asymptotically approaches 100% when f becomes large. As an example, for the s35932.scan circuit, we compare switching activity for loading/unloading under various schemes (see *Fig. 11*). The computed SA refers to shifting of only one vector, and is a measure of the worst-case scan-shift activity, and also the number of FFs that are to clocked during shifting. The activity per shift clock is determined by the length of

Table 3: Scan-shift switching activity

Circuit	# FFs	PIs	# Test patterns	Scan-shift switching activity (SA) and % energy savings						
				Fully linear scan chain	Pruned DTS		Serial DTS		3 linear chains	
					SA	savings	SA	savings	SA	savings
s208.scan	8	11	31	5015	1917	61.77	2688	46.40	1871	63
s713.scan	19	35	53	77610	13536	82.56	25608	67.00	25554	67
s838.scan	32	35	67	139428	21954	84.25	33837	75.73	47986	66
s953.scan	29	16	96	91436	19088	79.12	35170	61.53	30566	67
S1196.scan	18	14	151	77958	19952	74.41	34442	55.82	29192	63
S1238.scan	18	14	156	79686	20550	74.21	35256	55.76	29888	62
S1423.scan	74	17	67	269550	33160	87.70	54090	79.93	92237	66
S5378.scan	179	35	269	6187659	382791	93.81	662756	89.29	2083529	66
S9234.1.scan	211	36	287	8763780	494283	94.36	1061960	87.88	2947305	66
s13207.1.scan	638	62	428	104410365	2542069	97.56	8509093	91.85	34899244	67
s15850.1.scan	534	77	330	61520276	1693672	97.25	4726576	92.32	20643807	66
S35932.scan	1728	35	72	111434168	1231818	98.89	3209995	97.19	37219775	67
S38417.scan	1636	28	593	819768164	9517503	98.84	24628905	96.99	273917845	67
s38584.1.scan	1426	38	695	742811515	9655405	98.70	37564000	94.94	247588139	67

the currently active path and is a measure of instantaneous power. The total SA is a measure of energy needed to complete the shift.

7. Experimental Results

Experiments are carried out on several ISCAS-89 scan circuits, and the results are reported in *Table 3*. We assume that the primary inputs (PI) are also scanned in, so the effective number of scan flip-flops is increased accordingly. A 25-bit LFSR is used to generate 20,000 pseudorandom test vectors. After doing forward and reverse fault simulation for single stuck-at faults, only those vectors that contribute to fault dropping are retained, the number of which is shown in column 4. The total scan-shift switching activity (SA) for loading the test vectors and shifting out response vectors are then computed over the entire test session, and the results are reported for fully linear scan chain (column 5), pruned DTS architecture (column 6), serial DTS architecture (column 8), and multiple scan-path with 3 linear chains (column 10) as in [9]. Energy savings for the last three structures over fully linear scan-path are shown in columns 7, 9, 11 respectively. It may be observed that for a 3-chain multiple scan-path [9], the expected savings would be $[\{T*f^2 - s*T*(f/s)^2\}/T*f^2] = (1- 1/3) = 66.6\%$, which is a constant independent of the number of FFs. This is reflected in column 11. On the other hand, energy savings will tend to increase significantly on DTS architecture, as the number of FFs increases. This is depicted in columns 7 and 9. The pruned DTS provides more savings over the serial version. As discussed earlier, the instantaneous power both for shifting and driving clock signals (not shown in the table) will also reduce in the same fashion.

8. Conclusion and Future Problems

A novel scan-path architecture called double-tree scan (DTS) is proposed for low-power testing. A hierarchical clock controller is designed that activates only a single scan chain of DTS per shift clock in test mode, while blocking clock signals to all other flip-flops. In functional mode, the clock signal reaches all flip-flops for normal operation. Depending on loading strategy, an appropriate shift controller based on depth-first load (DFL) or breadth-first load (BFL) may be designed. For a pruned (serial) DTS, a DFL(BFL) controller may fit well. The architecture is simple and provides substantial power/energy savings for both scan-shift and clock activity. Issues in layout design and clock synchronization problem for DTS will be addressed in a future work. Minimization of test application time on DTS is another open area for further investigation.

References

[1] N. Nicolici and B. M. Al-Hashimi, Power-Constrained Testing of VLSI Circuits. *Kluwer Academic Publishers*, Boston, MA, 2003.

[2] R. M. Chou, K. K. Saluja, and V. D. Agrawal, "Scheduling tests for VLSI systems under power

constraints," *IEEE Trans. on VLSI Systems,* vol. 5, pp. 175-185, June 1997.

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

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

[5] G. Kiefer and H. -J. Wunderlich, "Deterministic BIST with multiple scan chains," *Proc. ITC*, pp. 1057-1064, 1998.

[6] N. Nicolici, B. M. Al-Hashimi, "Multiple scan chains for power minimization during test application in sequential circuits," *IEEE Trans. on Computers*, pp. 721-734, June 2002.

[7] A. Chandra and K. Chakrabarty, "Low-power scan testing and test data compression for system-on-a-chip," *IEEE Trans. on CAD*, pp. 597-604, May 2002.

[8] S. Wang and S.K. Gupta, DS-LFSR: "A BIST TPG for low switching activity," *IEEE Trans. on CAD*, pp. 842-851, July 2002.

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

[10] O. Sinanoglu and A. Orailoglu, "A novel scan architecture for power efficient, rapid test," *Proc. ICCAD*, Nov. 2002.

[11] J. Oh and M. Pedram, "Gated clock routing for low-power microprocessor design," *IEEE Trans. on CAD*, vol. 20, pp. 715-722, June 2001.

[12] B. B. Bhattacharya, S. C. Seth, and S. Zhang, "Low-energy BIST design for scan-based logic circuits," *Proc. International Conference on VLSI Design*, pp. 546-551, 2003.

[13] S. Gerstendoerfer and H. -J. Wunderlich, "Minimized power consumption for scan-based BIST," *Proc. ITC,* pp. 77-84, 1999; (also in *JETTA,* January 2000).

[14] S. Wang and S. K. Gupta, "ATPG for heat dissipation minimization during test application," *IEEE Trans. on Computers*, pp. 256-262, 1998.

[15] S. Wang, "Generation of low-power-dissipation and high-fault-coverage patterns for scan-based BIST," *Proc. ITC,* 2002.

[16] O. Sinanoglu, I. Bayraktaroglu, and A. Orailoglu, "Scan power reduction through test data transition frequency analysis," *Proc. ITC,* 2002.

[17] O. Sinanoglu, I. Bayraktaroglu, and A. Orailoglu, "Test power reduction through minimization of scan chain transitions," *Proc. VLSI Test Symposium*, 2002.

[18] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, "Efficient scan chain design for power minimization during scan testing under routing constraint," *Proc. 8th IEEE European Test Workshop,* May 2003.

[19] K.-J. Lee, J.-J. Chen, and C.-H. Huang, "Using a single input to support multiple scan chains," *Proc. ICCAD*, pp. 74-78, 1998.

[20] S. Samaranayake, F. Neuveux, E. Gizdarski, R. Kapur, N. Sitchinava, and T. W. Williams, "A reconfigurable shared scan-in architecture," *Proc. VTS*, 2003.