Physical-Aware Approaches for Speeding Up Scan Shift Operations in SoCs

Taehee Lee, Ik Joon Chang, Chilgee Lee, and Joon-Sung Yang

System-on-chip (SoC) designs have a number of flip-flops; the more flip-flops an SoC has, the longer the associated *scan test application time* will be. A *scan shift operation* accounts for a significant portion of a scan test application time. This paper presents physical-aware approaches for speeding up scan shift operations in SoCs. To improve the speed of a scan shift operation, we propose a layout-aware flip-flop insertion and scan shift operation–aware physical implementation procedure. The proposed combined method of insertion and procedure effectively improves the speed of a scan shift operation. Static timing analyses of state-of-the-art SoC designs show that the proposed approaches help increase the speeds of scan shift operations by up to 4.1 times that reached under a conventional method. The faster scan shift operation speeds help to shorten scan test application times, thus reducing test costs.

Keywords: Scan-based test, timing closure, test application time reduction, static timing analysis, interconnect pipelining.

I. Introduction

Recent system-on-chip (SoC) designs are larger in terms of their physical characteristics compared to previous designs. Consequently, scan test data volumes and *scan test application times* for SoCs are increasing. Large-scale SoC designs contain more scan cells and faults; consequently, more test patterns are required to achieve a high-quality SoC scan test. There has been a persistent demand to reduce scan test application time without impacting upon test quality; scan test application time can be represented as follows:

scan test application time
= (no. of test vectors × no. of scan flip-flops in a scan chain)
/ test clock frequency.

Scan test application time can be decreased by reducing both the number of test vectors and the number of scan flip-flops in a scan chain, and by increasing test clock frequency. In this regard, previous studies have attempted to reduce the number of test vectors by performing either a static or dynamic compaction depending on whether a test compaction is carried out either as a post-processing step after test generation or as part of the test generation process itself [1]–[4]. Both static and dynamic compaction methods are conducted for the generation of test sets with a small number of specified bits [5]. Earlier approaches have presented several different methods for the compression of test vectors to reduce scan test application time by partitioning a single scan chain into a number of shorter scan chains. Methods for compressing test vectors require some additional on-chip hardware, such as a decompressor (before any scan chains) and a response compressor (after any scan chains) [6], [7]. Furthermore, layout-aware scan designs,

Manuscript received Aug. 31, 2015; revised Nov. 10, 2015; accepted Dec. 30, 2015.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education (NRF-2015R1D1A1A01058856) and in part by the National Research Foundation of Korea (NRF-2013R1A1A1005832).

Taehee Lee (taehee77.lee@samsung.com) is with the Department of Semiconductor and Display Engineering, Sungkyunkwan University, Suwon, Rep. of Korea and the SOC design team in system LSI, Samsung Electronics, Hwaseong, Rep. of Korea.

Ik Joon Chang (ichang@khu.ac.kr) is with the Department of Electronics and Radio Engineering, Kyunghee University, Seoul, Rep. of Korea.

Chilgee Lee (cslee@skku.edu) and Joon-Sung Yang (corresponding author, js.yang@skku.edu) are with the Department of Semiconductor Systems Engineering, Sungkyunkwan University, Suwon, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.16.0115.0779

such as scan chain stitching and scan-flip-flop re-ordering, which aim to minimize wire lengths between flip-flops whilst giving consideration to physical locations [8], [9], have been studied. An alternative fundamental approach to enhancing scan test application time is that which seeks to speed up scan shift test clock frequency.

This paper describes practical approaches to speed up *scan shift operations*. The proposed approaches are unique in that they take into consideration the physical design step. First, we consider the physical design information related to scan input (SI) pads and scan output (SO) pads. SI and SO pads are generally located far from the first and last flip-flops in a scan chain. As the delay of an SI/SO path between SI/SO pads and first/last flip-flops exceeds a scan shift test clock period, an interconnect pipelining is required.

This paper presents a layout-aware flip-flop insertion on an SI/SO path. To achieve a fast scan shift operation speed, a scan shift operation–aware physical implementation procedure is performed between the synthesis and routing stages.

The following summarizes our proposed physical-aware approaches integrated in an SoC design flow.

1) A scan shift mode is *timing-optimized* at the same time as is the functional mode, in the placement stage.
2) Scan-chain reordering is performed to achieve a shorter routing length.
3) Multiple scan clock pads are placed close together.

The rest of this paper is organized as follows. Section II discusses a layout-aware flip-flop insertion. Section III explains in detail the scan shift operation–aware physical implementation procedure. Section IV contains the experimental results, and Section V outlines our conclusion.

II. Layout-Aware Flip-Flop Insertion for SI/SO Paths

Increasing integration density enables larger and more complex SoC designs. SoCs include hard-macro digital and analog intellectual properties (IPs); firm-macro IPs; and soft-macro IPs.

There are three reasons why the proposed layout-aware flip-flop insertion approach is needed. Firstly, in large-scale SoCs, distances between SI/SO pads and the first/last flip-flops are considerably large. Secondly, because many hard-macro IPs are reused or provided by third parties, there is often little flexibility with regards to floor plans and placement; they are generally placed at the side of an SoC. SI or SO pads shared with functional I/Os (that is, SI or SO pads that are multiplexed with functional I/Os) cannot be placed where hard-macro IPs are to be found. Thirdly, SI or SO pads cannot be shared with all I/O pads in an SoC. High-speed IOs, such as DDR interface and high speed NAND Flash interface, are not shared with SI/SOs, since multiplexers selecting either SI/SO or function I/O can cause problems such as delays and skews. Analog I/O cannot be shared with SI/SOs either. Only low-speed IOs, such as I^{2}S, I^{2}C, and SPI, are typically shared with SI/SOs. The degrees of freedom available for the optimal location of SI/SO pads is limited. The aforementioned three reasons serve to highlight why SI/SO pads and the first/last flip-flops in a scan chain are placed far apart from one another. A long-distance SI/SO path thus becomes a timing-critical path in a scan shift operation. Flip-flops can be inserted on a SI/SO path if the delay of the SI/SO path exceeds a scan shift clock cycle.

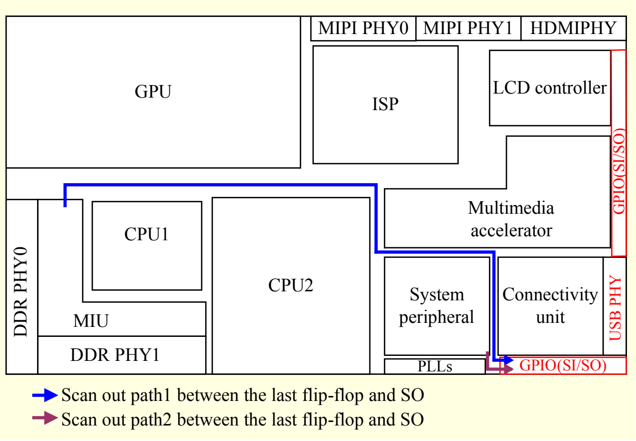

Fig. 1. Example of SoC floor plan.

Figure 1 shows an example of a floor plan from a state-of-the-art SoC. There are two SO paths — “scan out path 1” (blue line) and “scan out path 2” (red line). Scan out path 1 has its final flip-flop located in MIU, and it is connected to an SO pad (the whole path requires a very long wire). On the other hand, scan out path 2 has a shorter path to an SO pad; hence, scan out path 1 may cause a delay problem. To avoid timing delays, scan out path 1 needs to be partitioned by flip-flops based on the layout design.

Recent studies [10], [11] have proposed methods to determine flip-flop insertion locations for interconnect planning. This paper focuses on interconnect optimization between an SI/SO pad and multiple flip-flop interconnections. The proposed layout-aware flip-flop insertion approach finds the optimal number of flip-flops and their insertion locations along multiple SI/SO paths from a single SI/SO pad.

Figure 2 shows examples of SI/SO paths. An SI pad is connected to multiple flip-flops, such as “First FF_1” and “First FF_2,” through pipes. “Last FF_1” and “Last FF_2” are connected to an SO pad. An SI/SO path includes an I/O multiplexer, which is composed of a mode selection module and a general purpose input/output (GPIO) module. The mode selection module controls I/O functionality depending on operational modes such as test mode or functional mode. To avoid a timing

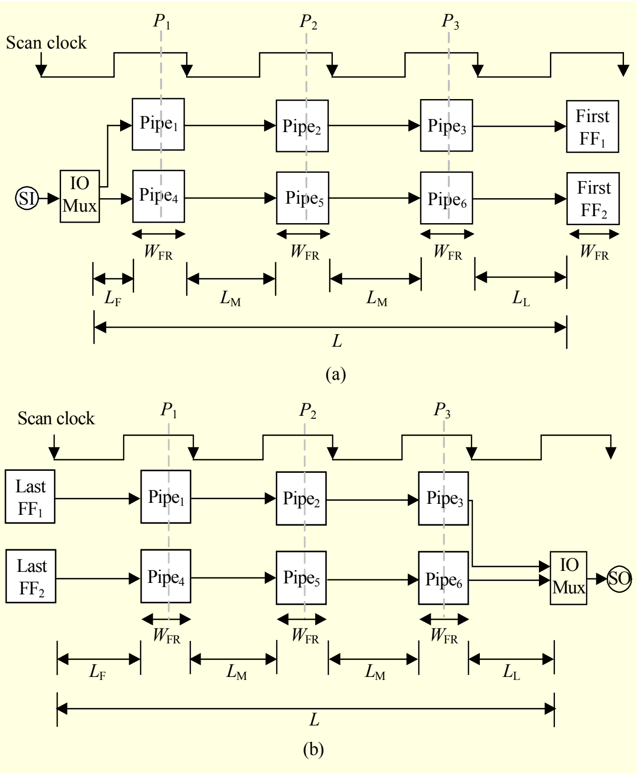

Fig. 2. Scan I/O paths between SI/SO paths and flip-flops: (a) SI path from SI pad to first flip-flop and (b) SO path from last flip-flop to SO pad.

delay problem along a long path, the example in Fig. 2 has six inserted flip-flops — Pipe$_1$, Pipe$_2$, Pipe$_3$, Pipe$_4$, Pipe$_5$, and Pipe$_6$. The maximum wire distances of the first segment, middle segment, and last segment are denoted as L_F, L_M, and L_L, respectively. The width of a feasible region where a flip-flop can be placed is denoted by W_{FR}. The maximum wire length without pipelining by flip-flop insertion is denoted by L_{MAX}.

We express the delays corresponding to L_{MAX}, L_L, L_M, and L_F by D_{MAX}, D_L, D_M, and D_F, respectively. They are given below as

$$D_{MAX} = T_{clk} - T_{pad} - T_{FF\_setup} - T_{I/OMux} - T_{unc}, \quad (1)$$

$$D_F\,(\text{Scan In Path}) = T_{clk} - T_{pad} - T_{FF\_setup} - T_{I/OMux} - T_{unc}, \quad (2)$$

$$D_L\,(\text{Scan In Path}) = T_{clk} - T_{FF\_setup} - T_{prop} - T_{unc}, \quad (3)$$

$$D_F\,(\text{Scan Out Path}) = T_{clk} - T_{FF\_setup} - T_{prop} - T_{unc}, \quad (4)$$

$$D_L\,(\text{Scan Out Path}) = T_{clk} - T_{IO\_setup} - T_{I/OMux} - T_{prop} - T_{unc}, \quad (5)$$

$$D_M = T_{clk} - T_{FF\_setup} - T_{prop} - T_{unc}, \quad (6)$$

where T_{clk} is a clock period, $T_{I/OMux}$ is a delay of an I/O multiplex module, T_{pad} is a pad delay, T_{FF_setup} is a flip-flop setup time, T_{IO_setup} is the time required by the external automatic test equipment (ATE), T_{prop} is a flip-flop D to Q propagation delay after positive clock edge, and T_{unc} is a clock uncertainty, such as skew or jitter.

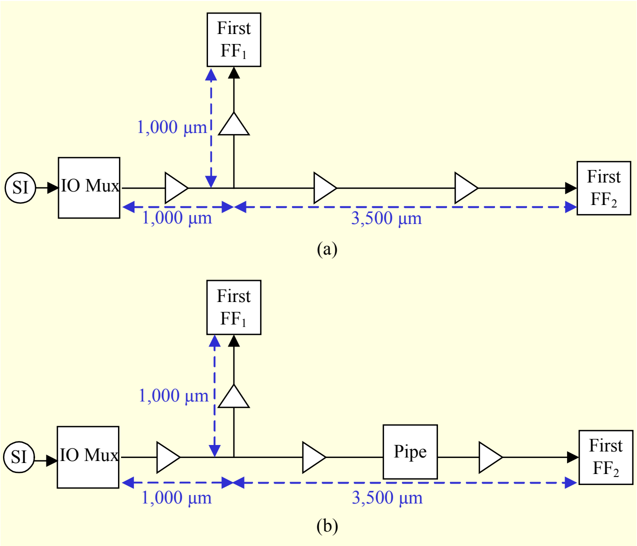

Fig. 3. Example of flip-flop insertion on SI path: (a) SI path structure before pipelining by flip-flop insertion and (b) SI path structure after pipelining by flip-flop insertion.

The delays D_{MAX}, D_M, D_L, and D_F are calculated by equations (1) to (6). The wire lengths corresponding to D_{MAX}, D_M, D_L, and D_F can be calculated by a CAD tool, such as IC Compiler [12]. We assume the following conditions: a scan shift clock, pad delay, setup time of a flip-flop, delay of IO Mux module, and clock uncertainty are 10 ns, 1 ns, 0.1 ns, 5.6 ns, and 0.4 ns, respectively. The delay D_{MAX} is calculated by (1) and is 2.9 ns. The wire length corresponding to the delay of 2.9 ns, including repeaters, is 4,000 µm under a 28 nm technology process, which is the maximum wire length L_{MAX}. The wire length can vary depending on the size of wire pitch, wire width, and repeater drive strength.

As shown in Fig. 3(a), the length between "IO Mux" and "First FF$_1$" is 2,000 µm. Because the length is shorter than 4,000 µm, L_{MAX}, this path does not need a flip-flop insertion. However, the other path between "SI" and "First FF$_2$" requires a flip-flop insertion since its length is 4,500 µm (which is larger than L_{MAX}). Figure 3(b) shows that a "Pipe" flip-flop is inserted between "SI" and "First FF$_2$." After "Pipe" insertion, a static timing is checked to validate whether the scan shift timing constraint is satisfied by flip-flop insertion.

The equations for calculating the number of inserted flip-flops are shown in (7). The central position and feasible region for the flip-flop insertion are described in (8) and (9), respectively. We denote P_i to be the center position of the feasible region of the ith flip-flop, as shown in Fig. 2. Equation (8) determines P_i. A flip-flop can be inserted into a feasible region to resolve a delay problem. A feasible region excludes blockages such as hard-macro IPs such that flip-flops are not inserted into a blockage region. A placement density constraint should be considered when the location of an inserted flip-flop is identified, because the final placement density after flip-flop insertion must not violate the placement density constraint.

$$N_{\text{Flip-Flop}} = \begin{cases} 0 & \text{if } L \le L_{\text{MAX}}, \\ 1 & \text{if } L_{\text{MAX}} < L < L_{\text{L}} + L_{\text{F}}, \\ \{(L - L_{\text{L}} - L_{\text{F}}) / L_{\text{M}} + 1\} & \text{otherwise,} \end{cases} \tag{7}$$

$$P_i = L_F + (i-1)L_{\text{M}} - (i - 1/2)W_{\text{FR}}, \tag{8}$$

$$W_{\text{FR}} = [|\, L_{\text{F}} + L_{\text{L}} + (n-1)L_{\text{M}} - L \,|] / n. \tag{9}$$

Figure 4(a) shows an SI path composed of one SI pad and three "First Flip-Flops" (First FFs). Each "First FF" connected to the SI pad is visited. The number of inserted flip-flops, central positions, and feasible regions are then determined using (7), (8), and (9), respectively. To address the interconnect latency problem of the SI path, seven "Pipe" flip-flops are inserted, as shown in Fig. 4(b). The feasible region for each inserted flip-flop is illustrated by a blue rectangle. Pipe$_1$ and Pipe$_4$ have an overlapping feasible region. For power reduction, Pipe$_1$ and Pipe$_4$ are merged and replaced by a single flip-flop. The merged single flip-flop, New Pipe$_1$, is placed in the overlapping region so as not to violate scan shift timing constraints, as shown in Fig. 4(c). New Pipe$_1$ has a higher driving strength to increase its load-driving capability and shorten the delay.

Algorithm 1. Layout-aware flip-flop insertion in scan path.

Input : Scan IO pads.

Output : New scan chain between SI/SO pads and their connected flip-flop.

```
For SI/SO pads = 0, 1, … , $L$ ($L$ is the number of SI/SO pads)
    For every flip-flop that is connected to each SI/SO pad
        Find the distance between SI/SO pad and its connected flip-flop
        Calculate $D_{\text{max}}$, $D_{\text{L}}$, $D_{\text{F}}$, and $D_{\text{M}}$ according to (1), (2), (3), and (6), respectively.
        Obtain $L_{\text{max}}$, $L_{\text{L}}$, $L_{\text{F}}$, and $L_{\text{M}}$.
        Determine the number of flip-flops inserted between SI/SO pad and its connected flip-flop. It is given by (7).
        Determine the central position and feasible region of the newly inserted flip-flops based on equations (8) and (9).
        They are then placed.
        Re-stitch between the flip-flop and SI/SO pad through the newly inserted flip-flops.
        If common overlapping feasible region exists between newly inserted flip-flops.
            The flip-flops are replaced by a single flip-flop.
            Re-stitch between the replaced flip-flop and its neighboring flip-flops.
        End if
    End for
End for
```

Algorithm 1 describes the proposed layout-aware flip-flop insertion method, which can be applied to an SI/SO path. Once the number of flip-flops, along with central positions and feasible regions, is determined, scan chains between the newly inserted flip-flops and SI/SO pads are re-stitched. If there is an overlapping feasible region between the inserted flip-flops,

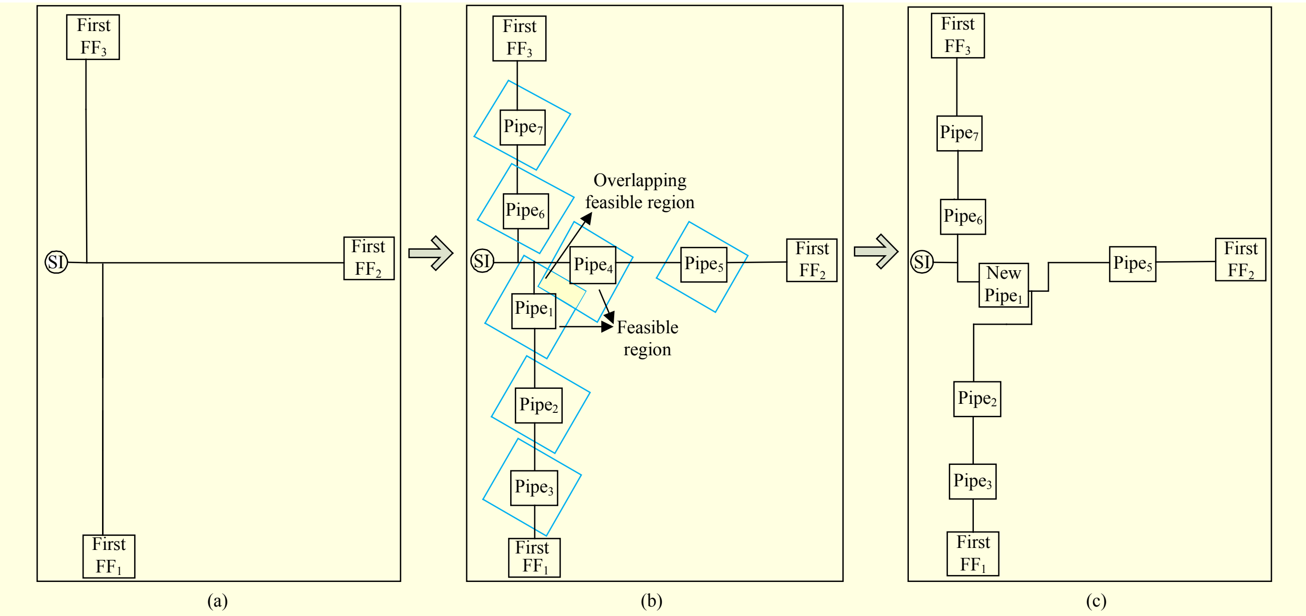

Fig. 4. (a) SI path with three first flip-flops, (b) SI path with pipes, and (c) merging Pipe$_1$ and Pipe$_4$ into New Pipe$_1$.

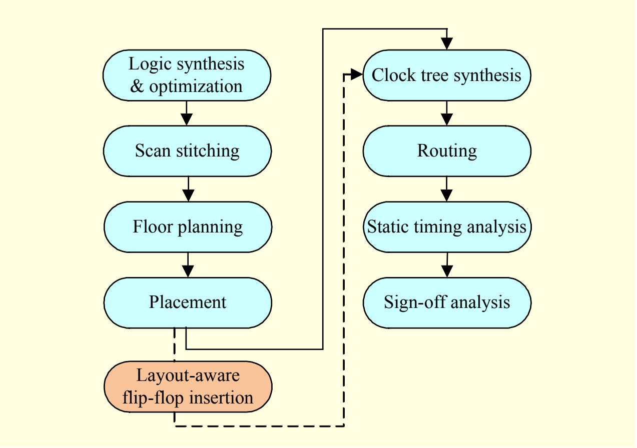

Fig. 5. Proposed design flow with layout-aware flip-flop insertion.

then they are merged and replaced by a single flip-flop. Scan chains are then re-stitched between the newly replaced flip-flop and its neighboring flip-flops.

Figure 5 shows a proposed design flow with timing-driven physical optimization steps for timing closure. Layout-aware flip-flop insertion flow is highlighted in a red box. As can be seen, the flow is added between the "Placement" and "Clock tree synthesis" stages.

III. Scan Shift Operation–Aware Physical Implementation Techniques

The worst cases of negative slack lie between SI/SO pads and the first/last flip-flops in a scan chain. A layout-aware flip-flop insertion method, described in Section II, is carried out to fix the worst cases of negative slack on SI/SO paths. To fix other timing violations on a scan shift path, not an SI/SO path, a scan shift operation–aware physical implementation procedure is proposed.

The following subsections describe the scan shift operation–aware physical implementation procedure used for speeding up a scan shift operation, which is one of the key factors in reducing scan test application time.

1. Placement of Scan Clock Pads

As SoC size becomes larger and its complexity increases, a multiple-clock-domain design style is required, and indeed has now become commonplace. A DFT architecture handles multiple scan clocks from external input pads [13], [14]. In functional mode, the asynchronous paths can commonly be treated as false paths. If the paths are treated as false paths in scan capture mode, then test coverage might be decreased due to unchecked timing violations.

In this work, instead of treating them as false paths, two different clocks are assigned to the asynchronous paths. The reason for assigning two clocks instead of a single clock is to avoid a hold violation on the asynchronous paths in scan capture mode. Once the different clocks, *CLK_A* and *CLK_B*, are assigned to a "launch flip-flop" and a "capture flip-flop," respectively, *CLK_B* toggles while *CLK_A* does not toggle, as shown in Fig. 6. In scan capture mode, the capture flip-flop, FF_2, has enough time to capture the combinational circuit output without any timing violations since *CLK_A* does not toggle. Hence, it can protect against a hold timing violation. However, in scan shift mode, timing violations should be checked and fixed between scan cells whose clock sources are different. To prevent timing violations, scan clock pads should be placed together at the floor planning stage to minimize clock skew. Otherwise, the clock skew between *CLK_A* and *CLK_B* can cause timing violations.

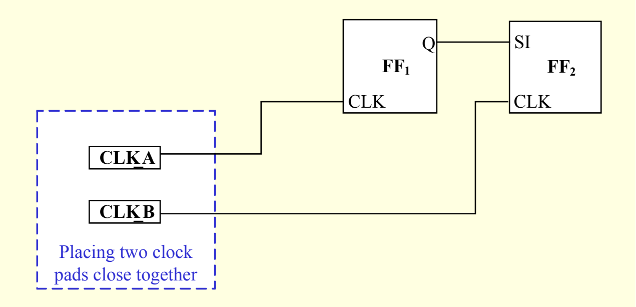

Fig. 6. Asynchronous scan shift path containing different clock sources.

2. Considering Timing Optimization for Scan Shift Mode

If all flip-flops are connected back-to-back on a scan shift path, then the scan shift operation speed can be easily enhanced. However, this may not be possible, due to the significant test pattern volume. Compression of test data is required to reduce the dependency on ATE memory requirement.

As shown in Fig. 7, for compression of test data, the MUX-based decompressor and XOR-based compressor are placed at SI and SO to expand the test patterns and to compact the responses, respectively. A higher compression ratio generally increases the logic depth of a decompressor and compressor. In addition, multiple multiplexers are required to support various scan test modes. If the decompressor, compressor, and multiplexers are not timing-optimized, then they can cause setup timing violations. In many product designs, due to a very short time-to-market request, the scan shift mode is not generally considered as a timing-optimization scenario, since the scan shift speed is slow. However, there is a significant request to minimize the test cost, and the scan shift operation needs to be accelerated. To speed up a scan shift operation, an

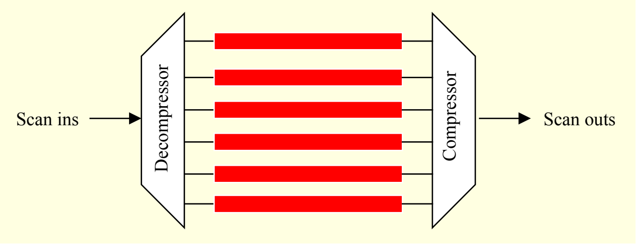

Fig. 7. SoC test compression architecture.

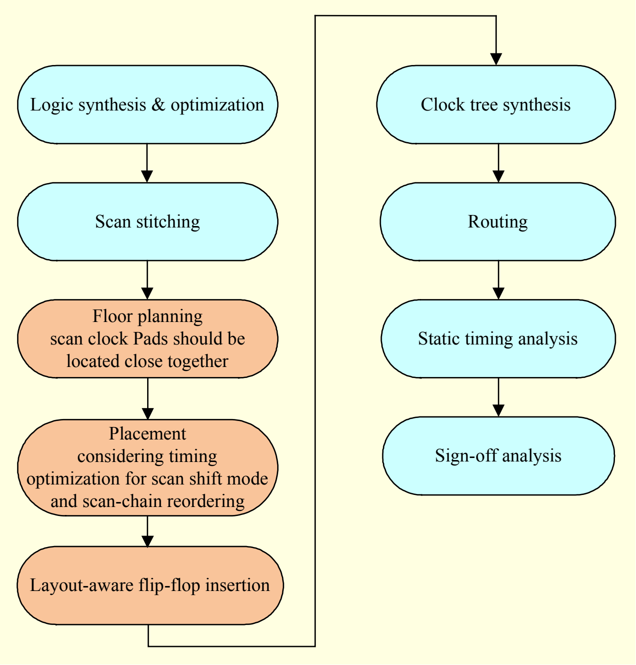
Logic synthesis & optimization
Scan stitching
Floor planning
scan clock Pads should be
located close together
Placement
considering timing
optimization for scan shift mode
and scan-chain reordering
Layout-aware flip-flop insertion
Clock tree synthesis
Routing
Static timing analysis
Sign-off analysis

Fig. 8. Scan shift operation–aware design flow.

SoC design should enhance the timing of the scan shift mode in the placement stage. Once scan-shift-mode timing is added in the placement stage, a small number of timing violations may occur in functional mode. The timing violations disappear after routing is completed, since the drive strength of logic cells are refined to fix the timing violations at the routing optimization stage.

3. Scan-Chain Reordering

The minimization of timing violations on scan paths can be significantly enhanced by scan-chain reordering. Routing also can cause a delay. Scan-chain reordering helps to reduce the routing congestion caused by scan chains. Flip-flops are reordered at the placement stage by reducing the length of nets between scan cells and wiring congestions. Scan-chain reordering has been investigated in [8] and [15]. We use a scan-chain reordering method for speeding up scan shift operations.

Figure 8 illustrates the proposed design flow used for improving scan shift speed in this paper. The scan shift operation–aware implementation techniques addressed in this section are added at the floor planning and placement stages and are highlighted in red. Once the scan shift operation–aware implementation techniques are completed, the layout-aware flip-flop insertion for SI and SO paths is performed and scan chains in SI/SO paths are then re-stitched, as described in Section II.

IV. Experimental Results

To analyze and compare scan shift operation speeds, the proposed physical-aware approaches are implemented in state-of-the-art industrial SoC designs under 28 nm technology.

Table 1 shows the experimental results. The first column shows five industrial SoCs — their combinational gate and flip-flop information are given in the second and third columns, respectively. Three different design approaches are chosen to evaluate the speed of a scan shift operation. The first design approach is the conventional design flow, which does not consider any physical design information. The second design approach includes the layout-aware flip-flop insertion on SI and SO paths, as explained in Section II. The last implementation is done with both the layout-aware flip-flop insertion on SI and SO paths and the scan shift operation–aware physical implementation procedure described in Section III. The results from each design approach are given in the fourth, fifth, and sixth columns, respectively. For each design approach, the worst cases of negative slack and the number of timing violations during the setup time are measured and counted when the target scan shift speed is 100 MHz. Under a conventional design flow, none of the considered SoC designs is able to meet the required scan shift speed specification; thus, all designs suffer from large amounts of negative slack. When the layout-aware flip-flop insertion method is used, the SI and SO paths are pipelined by flip-flops. As can be seen, this method significantly reduces the worst cases of negative slack. This helps to improve the speed of scan shift operations. However, upon comparing the number of paths having timing violations between the conventional design method and the layout-aware flip-flop insertion method, we see that the number of such paths is not significantly reduced under the layout-aware flip-flop insertion method. The number of paths having timing violations is reduced by around 5%: 95% of violations among all those paths that contain violations in the conventional design flow are left as unfixed. We can conclude that the worst cases of negative slack are caused from SI/SO paths to the first/last flip-flop in a scan chain. The last design flow, considering the layout-aware flip-flop insertion and scan shift operation–aware physical implementation procedure, solves all timing-related problems

Table 1. Timing comparisons with scan shift operation specification of 100 MHz.

SoC design	Number of combinational gates	Number of flip-flips	Conventional design flow		Design flow with layout-aware flip-flop insertion for SI and SO paths		Design flow with layout-aware flip-flop insertion for SI and SO paths and scan shift test–aware physical implementation techniques	
			Setup time worst negative slack (ns)	Number of violating paths	Setup time worst negative slack (ns)	Number of violating paths	Setup time worst negative slack (ns)	Number of violating paths
1	234,878,154	3,530,666	–24.30	34,934	–7.73	33,143	0	0
2	258,365,969	3,883,733	–26.73	38,427	–8.00	36,457	0	0
3	223,134,246	3,354,133	–23.09	33,187	–7.59	31,486	0	0
4	281,853,785	4,236,799	–28.67	43,668	–8.25	41,429	0	0
5	211,390,339	3,177,599	–21.63	30,742	–7.43	29,166	0	0

and meets the required scan shift speed specification. This tells us that the timing optimization for scan shift mode, asynchronous path impact minimization by close placement of scan clocks, and scan-chain reordering are necessary for speeding up a scan shift operation.

Accelerating a scan shift operation in an SoC is key to reducing the SoC's test cost. This paper shows how important the proposed physical-aware approaches are to speeding up scan shift operations in SoC designs.

The faster the scan test application time of an SoC design, the more power the SoC is required to consume. To reduce the power increment, the following low-power scan test methods can be employed:

- A multiphase scan shifting technique can be used to create a certain amount of skew between shifting of scan chains. The technique avoids simultaneous shifting operations such that the peak power dissipation is reduced during a scan shift operation.
- During a scan test, a scan test pattern shifts through scan chains. The functional logic driven by the scan flip-flops also toggles. This can increase power dissipation during a scan test. Gating the functional output of the scan flip-flop can avoid the switching activity of combinational logic on the functional output path during a scan shift.
- An X-filling technique can be employed. One can assign "don't care" bits to each test pattern to minimize the occurrence of transitions within a scan chain.
- Static compaction minimizes the number of test patterns generated by ATPG For example, the two patterns 01XX0 and 0X0X0 are merged to 010X0.This gives a low toggle rate.
- Scan-chain reordering, described in Section III, reduces wire length, which results in power reduction.
- Decoupling capacitors are inserted to stabilize the power grid voltage in scan shift mode as well as functional mode before tape out.

Any power increment induced by speeding up a scan shift operation can be offset by employing any of the above low-power scan test methods.

V. Conclusion

In modern SoC designs, reduction of scan test application time is an important issue. To achieve a reduction in scan test application time, physical-aware approaches are discussed in this paper. Layout-aware flip-flop insertion for SI/SO paths and a scan shift operation–aware physical implementation procedure are performed to achieve a shorter scan test application time. Experimental results are targeted to increase scan shift operation speeds from 24 MHz to 100 MHz. By employing the proposed physical-aware approaches, scan shift operation speeds can be improved by up to 4.1 times that of a conventional approach.

References

[1] K.O. Boateng, H. Kawasaki, and T. Nakata, "A Method of Static Compaction of Test Stimuli," *Asian Test Symp.*, Kyoto, Japan, Nov. 19–21, 2001, pp. 137–142.

[2] I. Pomeranz, "Static Test Compaction for Scan Circuits by Using Restoration to Modify and Remove Tests," *IEEE Trans. Comput.-Aided Des. Integrated Circuits Syst.*, vol. 33, no. 12, Dec. 2014, pp. 1955–1964.

[3] B. Ayari and B. Kaminska, "A New Dynamic Test Vector Compaction for Automatic Test Pattern Generation," *IEEE Trans. Comput.-Aided Des. Integrated Circuits Syst.*, vol. 13, no. 3, Mar. 1994, pp. 353–358.

[4] S. Neophytou, S. Hadjitheophanous, and M.K. Michael, "On the Impact of Fault List Partitioning in Parallel Implementations for Dynamic Test Compaction Considering Multicore Systems," *Int. Des. Test Symp.*, Marrakech, Morocco, Dec. 16–18, 2013, pp. 1–6.

[5] S.N. Neophytou and M.K. Michael, "Test Set Generation with a Large Number of Unspecified Bits Using Static and Dynamic Techniques," *IEEE Trans. Comput.*, vol. 59, no. 3, Mar. 2010, pp. 301–316.

[6] N.A. Touba, "Survey of Test Vector Compression Techniques," *IEEE Des. Test Comput.*, vol. 23, no. 4, Apr. 2006, pp. 294–303.

[7] M. Chloupek, O. Novak, and J. Jenicek, "On Test Time Reduction Using Pattern Overlapping, Broadcasting and on-Chip Decompression," *IEEE Int. Symp. Des. Diagnostics Electron. Circuits Syst.*, Tallinn, Estonia, Apr. 18–20, 2012, pp. 300–305.

[8] P. Gupta, A.B. Kahng, and S. Mantik, "Routing-Aware Scan Chain Ordering," *Asian South Pacific Des. Autom. Conf.*, Jan. 21–24, 2003, pp. 857–862.

[9] S. Banerjee et al., "Layout-Aware Illinois Scan Design for High Fault Coverage," *Int. Symp. Quality Electron. Des.*, San Jose, CA, USA, Mar. 22–24, 2010, pp. 683–688.

[10] R. Lu et al., "Flip-Flop and Repeater Insertion for Early Interconnect Planning," *Des. Autom. Test Europe Conf. Exhibition*, Paris, France, Mar. 4–8, 2002, pp. 690–695.

[11] J. Xu, "*Strategies of Handling Global Signals in Nanometer Integrated Circuits*," M.S. thesis, Dept. Electr. Comput. Eng., University of Illinois, Chicago, USA, 2009.

[12] Synopsys IC Compiler Datasheet, Accessed Mar. 1, 2016. http://www.synopsys.com/Tools/Implementation/PhysicalImplementation/Documents/iccompiler_ds.pdf

[13] V. Jain and I. Waicukauski, "Scan Test Data Volume Reduction in Multi-Clocked Designs with Safe Capture Technique," *Proc. Int. Test Conf.*, Oct. 2002, pp. 148–153.

[14] L.T. Wang et al., "VirtualScan: A New Compressed Scan Technology for Test Cost Reduction," *Proc. Int. Test Conf.*, Oct. 26–28, 2004, pp. 916–925.

[15] P.C. Tsai and S.J. Wang, "Multi-mode Segmented Scan Architecture with Layout-Aware Scan Chain Routing for Test Data and Test Time Reduction," *IET Comput. Digit. Techn.*, vol. 2, no. 6, Nov. 2008, pp. 434–444.

Taehee Lee received his BS degree from Chungnam National University, Daejeon, Rep. of Korea and his MS degree from the University of Southern California, Los Angeles USA, in 1997 and 2003, respectively. He joined the SoC Design Team of Samsung Electronics, Hwaseong, Rep. of Korea, in 2003, where he was engaged in the development of Samsung mobile SOC products. From 2006 to 2010, he was dispatched to Apple, Cupertino, CA, USA and joined the Application Processor Development. From 2011 to 2014, he was engaged in the physical implementation of Samsung application processors. He is currently a senior engineer at Samsung Electronics.

Ik Joon Chang received his BS degree, with summa cum laude, in electrical engineering from Seoul National University, Rep. of Korea. He acquired his MS and PhD degrees from the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA, in 2005 and 2009, respectively. Currently, he is an assistant professor at Kyunghee University, Seoul, Rep. of Korea. In 2005, he was awarded the Samsung Scholarship Foundation.

Chilgee Lee received his BS degree from Sungkyunkwan University, Suwon, Rep. of Korea, in 1980; his MS degree from Arizona State University, Tempe, USA, in 1985; and his PhD degree from the University of Arizona, Tucson, USA, in 1990, all in electrical and computer engineering. Upon graduation, he worked for Samsung Electronics Co., Hwaseong, Rep. of Korea, for five years. He is currently a professor at Sungkyunkwan University, Suwon, Rep. of Korea. His main research interests include system-level computer simulation and manufacturing optimization.

Joon-Sung Yang received his BS degree from Yonsei University, Seoul, Rep. of Korea, in 2003 and his MS and PhD degrees from the University of Texas, Austin, USA, in 2007 and 2009, respectively, all in electrical and computer engineering. Upon graduation, he worked for the Intel Corporation, Austin, TX, USA, for four years. He is currently an assistant professor at Sungkyunkwan University, Suwon, Rep. of Korea. His research interests include VLSI testing, silicon-debug and nanometer-scale test and design methodologies. He was a recipient of the Korea Science and Engineering Foundation Scholarship, in 2005. He received a Best Paper Award at the 2008 IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems and was nominated for Best Paper Award at the 2013 IEEE VLSI Test Symposium.

http://dx.doi.org/10.4218/etrij.16.0115.0779