Comprehensive Optimization of Scan Chain Timing During Late-Stage IC Implementation

Kun Young Chung[1], Andrew B. Kahng[1,2] and Jiajia Li[2]
CSE[1] and ECE[2] Departments, UC San Diego, La Jolla, CA, USA
{k1chung, abk, jil150}@ucsd.edu

ABSTRACT

Scan chain timing is increasingly critical to test time and product cost. However, hold buffer insertions (e.g., due to large clock skew) limit scan timing improvement. Dynamic voltage drop (DVD) during scan shift further degrades scan shift timing, inducing "false failures" in silicon. Hence, new optimizations are needed in late stages of implementation when accurate (skew, DVD) information is available. We propose skew-aware scan ordering to minimize hold buffers, and DVD-aware gating insertion to improve scan shift timing slacks. Our optimizations at the post-CTS and post-routing stages reduce hold buffers by up to 82%, and DVD-induced timing degradation by up to 58%, with negligible area and power overheads.

1. INTRODUCTION

Scan chain timing is an important consideration in modern scan chain design. Setup timing of scan shift timing paths or scan paths directly affects test time and cost, and any improvement of scan shift timing or scan timing will not only reduce test time and cost, but potentially improve robustness of test as well.[1] In modern designs, the volume of test patterns is very likely to increase significantly due to (i) the increased importance of delay testing (especially in FinFET technology), (ii) increased test coverage requirements, and (iii) increased design gate counts, which in turn increase test time. To compensate the large volume of test patterns and to reduce the test time, speedup of scan shift is needed. However, fast scan shift worsens dynamic voltage drop (DVD) due to higher switching activity, and induces smaller scan timing slacks which make scan timing more vulnerable to DVD. Furthermore, due to the small number of logic instances along scan timing paths (i.e., between consecutive scan flip-flops in a given scan chain), scan timing paths are vulnerable to hold violations. Mitigation of hold timing violations along scan chains entails hold buffer insertions which induce power and area overheads.

In this work, we address two timing issues related to scan chain.

- First, we perform scan ordering that exploits knowledge of clock skew and scan cell locations, so as to reduce hold violations along the scan chain and enable the removal of hold buffers. Figure 1 shows a simple example where reordering scan cells leads to positive skews between consecutive scan cells in a scan chain, thus removing hold violations.
- Second, scan test at a high frequency (especially during scan shift) is highly likely to incur large dynamic voltage drop (DVD), which in turn degrades scan timing and causes "false failures" in silicon. To address this, we perform DVD-aware gating insertion to reduce dynamic voltage drop during scan shift and maximize scan timing slacks.

Figure 1: Illustration of skew-aware scan ordering that removes hold violation. L is clock latency.

Although gating insertion and scan ordering optimizations have been proposed by many previous workers [1] [3] [4] [5] [8] [9] [14] [17], these optimizations are performed during early-stage IC implementation (e.g., during synthesis or placement). However, due to subsequent physical implementation steps (notably, clock tree synthesis, signal routing, and buffering and gate sizing), design information such as clock skew, cell power and timing slacks can be very different at late design stages (e.g., after routing). Figures 2 and 3 respectively show that hold-critical scan timing paths and DVD hotspots vary markedly between post-placement and post-routing states of the design. As a result, an early-stage optimization might be misleading and result in poor solution quality.

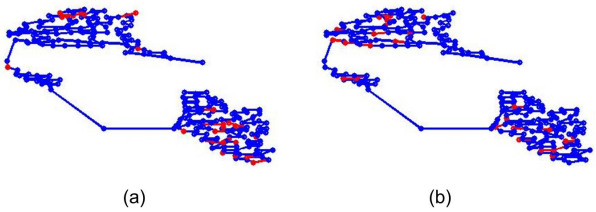

Figure 2: Hold-critical scan timing paths vary between (a) post-placement stage and (b) post-routing stage. In red are the top 10% of the hold-critical paths. In blue are the non-critical paths. Design: LEON3MP. Technology: 28LP.

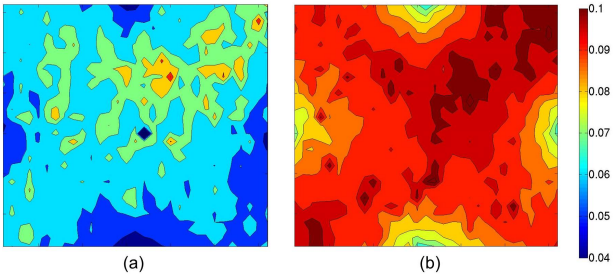

Figure 3: Dynamic voltage drop (DVD) varies between (a) post-placement stage and (b) post-routing stage. Design: LEON3MP. Technology: 28LP.

[1]Depending on the individual SoC design, structural test time can be up to 50% or even a larger proportion of the total test time. Scan time typically dominates structural test time.

DAC'16, June 05 - 09 2016, Austin, TX, USA.

http://dx.doi.org/10.1145/2897937.2897998.

On the other hand, previous post-routing stage scan chain optimizations mainly focus on test pattern optimization. For example, these optimizations use an ATPG engine to generate low-power test patterns so as to reduce DVD and improve scan timing. However, flip-flop toggling ratios may not necessarily be an accurate indicator of circuits' switching activity or devices' power consumption, and hence may not be able to effectively mitigate the real DVD hotspot issues. Furthermore, generating power-aware test patterns suffers significantly from test pattern inflation and test coverage degradation. For instance, based on our industrial colleague's design experience with modern CPU designs containing multi-millions of flip-flops, imposing limits of 20% and 30% peak flip-flop switching activities in scan capture and scan shift, respectively, ends up with a 2 or 3 times higher test pattern count along with degradation in test coverage.

In this work, we are the first to propose late-stage scan timing optimizations for hold buffer removal and DVD-aware gating insertion. Of course, such late-stage ECO-based scan optimization risks large impacts on design quality – in particular, timing degradation and/or power and area overheads from additional gates being inserted in the functional paths as well as incremental routing due to scan reordering. Therefore, our optimizations necessarily comprehend timing impacts on datapaths in function mode, while keeping area and power overheads to negligible levels (e.g., <1%). We note that test pattern (compression) optimization is not the focus of this work. However, our optimizations can easily be combined with existing test pattern optimizations. Our contributions are as follows.

- We are the first to propose a comprehensive scan timing optimization at late-stage IC implementation.
- We propose a scan reordering optimization that is aware of clock skew and scan cell locations; this optimization removes up to 82% of hold buffers along scan chains in 28LP testcases.
- We propose an ECO-based gating insertion approach to improve DVD-aware scan timing during scan shift; this optimization reduces the DVD-induced slack degradation by up to 58% with negligible area and power overhead.
- We validate our approaches on a realistic implementation flow (including DFT and DVD-aware timing analysis) in a commercial 28LP technology; this implementation flow has been developed under the guidance of our industrial colleagues.

2. RELATED WORK

We now review previous works on scan ordering optimization and gating insertion, as well as other methods for DVD reduction during scan shift.

Scan ordering. Scan chain ordering optimizations have typically been formulated as (symmetric or asymmetric) *Traveling Salesperson Problem* (TSP) optimizations, usually with a wirelength minimization (i.e., total tour cost) objective. An early TSP-based heuristic for scan ordering, which largely ignores physical information, is due to Feuer and Koo [4]. Works such as [1] [8] [9] [14] [17] consider physical information and routing to minimize the wirelength overhead of scan chains. Gupta et al. [6] reduce wirelength overhead of scan routing by considering the availability of connection points on entire fanout routing trees (not simply at scan-out pins), and use [25] as their TSP solver. The same authors later consider timing in addition to wirelength minimization [7]. Cui et al. [2] propose pre-ordering of clusters to improve power reduction. Seo et al. [19] combine gating, clustering and reordering to minimize scan power.

The above-mentioned academic works, along with commercial tools [16], focus on constructing scan chains from scratch for given new layout. Alternatively, Kahng et al. [11] address an ECO scan chain optimization context. Tudu et al. [21] propose a graph-based optimization to reorder scan cells based on a given test pattern, such that the transitions during scan shift and capture are minimized.

For one of our present foci, namely, scan reordering to minimize hold buffers, we are aware of only the previous work embodied in the 2003 patent of Teene [20]. In Section 3.1 below, we propose a scan ordering approach for minimization of hold buffers along scan chains. As compared to the approach in [20] which is aware of skew, our approach also comprehends cell locations, wire delay and setup timing constraints along datapaths.

Gating approaches. Due to excessive switching activity during at-speed test, test power is typically a major issue during circuit design and test pattern generation. *Gating insertion* is one of the known methods to reduce scan power. To suppress the activity of fanout combinational cells from scan flip-flops during scan shifting, Gerstendörfer et al. [5] propose to insert gating logic at the output (i.e., Q pin) of scan flip-flops. To minimize the area overhead and the delay impact, Elshoukry et al. [3] propose a critical path-aware partial gating approach, in which the gating points are selected based on the number of fanout cells and their fanouts. The authors of [3] also point out that such gating approaches can incur large peak power when gating logic instances change from the gating mode to the transparent mode. They propose to assign a separate control block for each scan chain and to enable/disable the gating logic instances for different scan chains one at a time. In addition to gating at the Q pin of a scan flip-flop, Jayaraman et al. [10] propose to also gate the internal nodes inside the fanout cone of a scan flip-flop. The increased flexibility offers a better tradeoff between area overhead of gating logic and test power reduction. Although such gating approaches reduce shift power, the increased capacitance can increase capture power. Zhao et al. [22] study the tradeoff between test power and capture power reductions.

Although the above gating optimization approaches all reduce test power, they perform optimization at the post-synthesis stage, when detailed (final) layout information is not available. Therefore, these approaches cannot accurately capture the locations of DVD hotspots, and may make suboptimal or guardbanded decisions. Lee et al. [13] address such issues when removing DVD hotspots by post-layout test pattern modification. However, the updated test pattern is not efficient and can increase the test time. In another work, since the voltage drop issue typically occurs at the beginning of scan shift (where the scan shift causes a sudden increase in switching activity), Schulze et al. [18] propose to start scan shift with a lower frequency (implicitly assuming that the scan shift uses an independent clock). Once the test power supply has responded to its initial di/dt event, they increase the shift frequency.

We observe that none of the above works consider timing impact of DVD and optimize DVD-aware scan timing slacks. To our knowledge, ours is the first work to propose gating insertion to minimize scan timing degradation due to DVD.

3. METHODOLOGY

We propose two basic optimization levers to improve scan timing at the post-routing stage while minimizing power and area overheads. Our optimizations comprehend both hold and setup constraints not only in scan mode, but in function mode(s) as well.

- **Post-routing scan reordering.** Since scan chains contain few combinational gates, they are typically hold-critical. As a result, hold buffers are inserted during the placement and routing stages, incurring power and area overheads. Such buffer insertion must be guardbanded since final routing, extraction and signoff timing analysis are not yet known. On the other hand, we propose scan chain reordering at the *post-routing* stage which can exploit knowledge of clock skews, exact scan cell locations, and post-routing timing slacks. This enables a more surgical improvement of hold timing and removal of hold buffers.
- **Fanout gating for dynamic voltage drop mitigation.** Dynamic voltage drop (DVD) during scan shift can degrade scan setup timing, leading to "false failures" in silicon. We propose gating insertion to reduce the DVD and improve scan setup timing slack. Again, while gating insertion is a previously-known technique, our contribution lies in demonstrating practicality and benefits when applied at the post-routing stage.

Table 1 lists notations used in the following discussion.

Table 1: Notations used in our work.

Term	Meaning
q_k	k^{th} scan chain, $(1 \leq k \leq K)$
Q	set of scan chains in design
v_k	number of hold violations along scan chain q_k
p_j	j^{th} scan timing path $(1 \leq j \leq M)$
S_j	setup timing slack of scan timing path p_j $(1 \leq j \leq M)$
c_i	i^{th} cell $(1 \leq i \leq N)$
g_i	switching power of cell c_i $(1 \leq i \leq N)$
s_i	setup timing slack of cell c_i
h_r	DVD hotspot $(1 \leq r \leq R)$
$d_{i,i'}$	Manhattan distance between cells c_i and $c_{i'}$

3.1 Scan Reordering for Hold Buffer Removal

Scan timing paths with negative clock skew values[2] and small distances between launch and capture flip-flops are prone to hold violations (as illustrated in Figure 4), whereby hold buffers are inserted along the scan chains to meet hold constraints. However, hold buffer insertions cause area and power overheads, which at least indirectly compromise functional and/or test timing.[3] Thus, we propose a scan cell reordering optimization at the post-routing stage (i) to achieve a greater incidence of positive skew values, and (ii) to slightly increase distance between consecutive scan cells so as to increase wire delay and enable buffer removal. We define the scan reordering problem for hold buffer removal as follows.

Post-Routing Scan Reordering: **Given** a design (i.e., netlist (.v), and placement, CTS, and routing solutions (.def) with scan chain(s) inserted), timing constraints (.sdc), hold buffer list, upper bounds on wirelength penalty, Liberty NLDM timing and power models (.lib), and fixed subchain ordering constraints (SCANDEF), **perform scan reordering** to maximize the number of hold buffers removed.

Our reordering optimization uses the *2-opt* local search heuristic for the traveling salesperson problem [12], as explained in Algorithm 1. We sequentially optimize each scan chain in a design. For each scan chain q_k and a given node c_i, we perform 2-opt swaps along c_i's downstream nodes and select the ordering solution with the minimum number of hold violations (this heuristically maximizes the number of hold buffer removals) (Lines 4-13).

Algorithm 1 Scan Reordering.

```
1:  for all q_k ∈ Q do
2:      q' ← q_k                          // scan chain ordering solution
3:      v' ← timingAnalysis(q')           // extract number of hold violations
4:      for i := 2 to N^k − 2 do
5:          for i' := i+1 to N^k − 1 do
6:              q'' ← 2OptSwap(q', i, i')
7:              v'' ← timingAnalysis(q'')
8:              if v'' < v' && feasible(q'') then
9:                  q' ← q''; v' ← v''
10:             end if
11:         end for
12:         q_k ← q'
13:     end for
14:     Reorder scan chain based on updated q_k
15: end for
16: ECO route
```

[2]We use the standard definition of *skew* between two sequentially adjacent flip-flops as launch clock latency minus capture clock latency. Note that a *scan timing path* is a timing path between (launch and capture) flip-flops that are consecutive in a scan chain.

[3]Larger area spreads and slows timing paths, while larger power may be compensated by reduced clock frequencies. Moreover, hold buffer insertion can potentially create new setup timing criticalities, as detailed below.

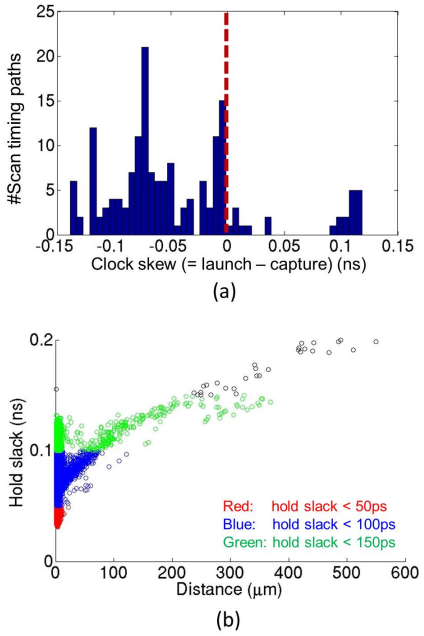

Figure 4: Causes of hold violations on scan timing paths. (a) Skew distribution of scan timing paths with hold buffers inserted. Negative skew values correlate with increased likelihood of hold violations. (b) Distances between consecutive scan cells versus hold timing slacks. Start-end pairs of scan cells separated by small distances have small hold slacks. Design: LEON3MP. Technology: 28LP.

Treatment of subchains. To honor the fixed scan subchain ordering constraints specified by SCANDEF [27] in the input, each subchain with fixed ordering is merged into a single node in the input to our optimization. Further, we observe that hold buffers may be shared between scan timing paths and datapaths: scan timing optimization cannot remove these hold buffers if the removals will cause hold timing violations along datapaths. We therefore do not optimize the corresponding subchains, but instead also merge such subchains into single nodes in the input to our optimization.

Avoidance of setup timing violations. We observe that a given hold buffer can shield large wire capacitance and test input pin capacitance at the Q-pin of a timing-critical flip-flop. Removing the hold buffer can thus incur setup timing violations along datapaths. Therefore, timing constraints for both datapaths and scan timing paths must be comprehended when evaluating the swap (i.e., scan reordering) moves. In our optimization, we evaluate timing slack changes due to scan chain ordering based on Liberty timing models (Line 7). We estimate wire delay and capacitance based on Manhattan distance between consecutive scan cells.

Additional constraints in the local search. During the local search, we select solutions (i) that have a reduced number of hold violations, (ii) that have no timing degradation along either scan timing paths and datapaths, and (iii) that satisfy the prescribed upper bound on wirelength penalty (Line 8). More specifically, we set upper bounds on wirelength increase for each pair of consecutive scan cells and each scan chain. Note that to avoid ECO impact (e.g., buffer insertion, placement legalization), we discard solutions that create additional hold violations for pairs of consecutive scan cells which are hold timing-feasible (i.e., have non-negative hold timing slacks) in the original solution. Without such constraints, additional hold violations can be created for a hold timing-feasible scan cell pair when the total number of hold violations along the scan chain is decreased.

To summarize: $feasible()$ in Line 8 indicates a solution that is timing feasible, maintains a bounded wirelength overhead, and incurs no additional hold violations. Finally, after all scan chains are optimized, we perform ECO routing with a commercial place-and-route tool.

3.2 DVD-Aware Gating Insertion

As noted above, scan shift typically consumes high power and causes excessive dynamic voltage drop (DVD), which in turn degrades scan setup timing and leads to "false failures" in silicon. To address such DVD-aware timing degradation, we propose the post-routing application of gating insertion [3] [5] [10] to reduce switching activity of downstream cells and thereby reduce DVD impact on scan timing. We state the DVD-aware gating insertion problem as follows.

Dynamic Voltage Drop Mitigation by Gating: Given a design (i.e., netlist (.v), and placement, CTS, and routing solutions (.def) with scan chain(s) inserted), timing constraints (.sdc), Liberty timing and power models (.lib), switching activities per cell instance, and upper bounds on area/power overheads, **perform gating insertion as ECOs** to maximize the minimum slack of DVD-aware scan timing.

Figure 5 shows our proposed three-step optimization flow. (1) We first determine DVD hotspots which have the largest impact on scan timing (i.e., the DVD hotspots containing timing-critical scan cells with the worst slack degradation due to DVD), by solving an integer linear program (ILP). (2) We then allocate gating locations (in the netlist) so that the switching activities of downstream non-scan cells within the selected DVD hotspots are minimized. The reduced switching activities lead to minimized power and hence a reduction of DVD within the selected hotspots. We determine the gating locations through a netlist traversal that is guided by sensitivity functions (i.e., functions to estimate the sensitivity of dynamic power reduction in downstream cells to a gating insertion). (Note that the potential gating insertion locations are not limited to Q pins of scan cells.) We use both AND and OR gates for gating. Figure 6 shows an example with OR gate. In addition, we honor datapath timing constraints in function mode. Further, we perform sizing and VT swapping on the inserted gating cells, such that timing impact due to gating insertion is minimized. (3) Last, we insert gating cells as an ECO step: specifically, we perform a matching-based optimization to determine the whitespace suited for gating insertions. We now give additional details of these steps.

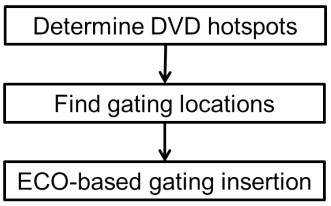

Figure 5: Optimization flow for gating insertion to optimize DVD-aware timing slacks.

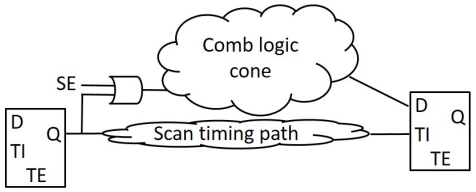

Figure 6: Illustration of gating insertion with an OR gate. TI and TE are respectively test input and test enable pins. SE is scan enable net.

(1) Determine DVD hotspots. To minimize the ECO impact and corresponding area and power overheads of gating insertions while maximizing timing benefits from DVD reduction, it is important to determine the DVD hotspots that have the largest impact on the worst-slack scan timing path. To achieve this, we divide the block area into grids, where each grid is a candidate DVD hotspot. We then set up and solve the following ILP, which selects the DVD hotspots to optimize from among candidates that have maximum impact on scan timing.

$$\begin{aligned} &\textit{Maximize} && S_{min} \\ &\textit{Subject to} && S_j + \sum_{c_i \in p_j} \alpha_i \cdot \Delta_i \geq S_{min},\ \forall p_j && (1) \\ & && \alpha_i \leq L \cdot \beta_r,\ \forall c_i \in h_r && (2) \\ & && \sum \beta_r \leq R && (3) \\ &\textit{Binary} && \alpha_i,\ \beta_r && (4) \end{aligned}$$

Here, S_{min} is the minimum slack in the design; S_j is the worst slack of scan timing path p_j; Δ_i is the cell slack improvement due to DVD reduction; c_i is a cell instance; α_i is a binary indicator of whether the DVD on c_i will be optimized (i.e., whether c_i is in the selected DVD hotspot); β_r is an indicator of whether hotspot h_r will be optimized; L is a large constant number; and R defines the upper bound on the number of hotspots to be optimized. Given that the number of scan timing paths grows linearly with the number of flip-flops and the number of stages along a scan timing path is typically small, the runtime complexity of our ILP is not high. In our experiments, even for the largest design with 474K instances and 445 scan chains, the runtime is less than 1 second on a 2.5GHz Intel Xeon server. Three important considerations are as follows.

Grid size. We note that the sizes of grids (i.e., DVD hotspots) can have significant impact on solution quality. A large grid size can result in a large number of gating insertions, with corresponding area and power overheads. On the other hand, a small grid size may contain only a small number of cells; gating these cells will not effectively reduce DVD.

Grid aspect ratio. We observe that the aspect ratios of grids also impact solution quality. Given that cells within the same row share the same power and ground rails, it is more effective to define row-based hotspots (i.e., hotspots with single-row height). Our experimental results in a commercial 28LP technology confirm this hypothesis: the DVD reduction of optimization with single row-height hotspots is 1.7× that of double row-height hotspots with the same area. Thus, in the experiments reported below, we empirically define the grid size as 80μm x 1.2μm.

Need for iteration. Last, we note that there is a "chicken-and-egg" loop between the assumed DVD reduction to estimate Δ_i, versus the optimized DVD values. To address this, we perform iterative optimization such that we use the average DVD reduction value from simulation on the optimized design as the input DVD reduction assumption to the next-iteration optimization. Our experimental results show that such an iterative optimization converges (i.e., no improvement between two consecutive iterations) after the second iteration in most cases.

(2) Find Gating Locations. Based on the selected DVD hotspots, we traverse the netlist and determine the gating locations based on sensitivity functions. The objective of this optimization is to minimize the switching activities of non-scan cells within the selected DVD hotspots with minimized area and power overheads. In addition, such gating insertions must comprehend the datapath timing constraints at function mode and not create additional timing violations.

All gating cells need to connect to scan enable (SE) nets, which typically have a tree structure in a design block (shown in Figure 7(a)). Since the layout location of SE nets and the polarity of these SE net signals will affect wirelength penalty due to gating insertions as well as the types of gating cells used (e.g., AND gate versus OR gate), we extract the SE net information as input to our optimization. We divide the block area into 5μm x 5μm grids, and for a grid containing SE nets, we assign the SE net to the grid so that a gating insertion within the grid will connect to this SE net. We break ties based on the levels of the SE nets, preferring to select a net closer to the bottom level so as to minimize the delay impact on fanout SE nets. For the grids without SE routing, we execute spiral search in their neighbor grids for the SE net with minimum distance (as shown in Figure 7 (b)).

Algorithm 2 describes our gating insertion flow. We perform power simulation and use G_i to store the total switching power of cells within selected DVD hotspots H at the i^{th} iteration (Line 1).

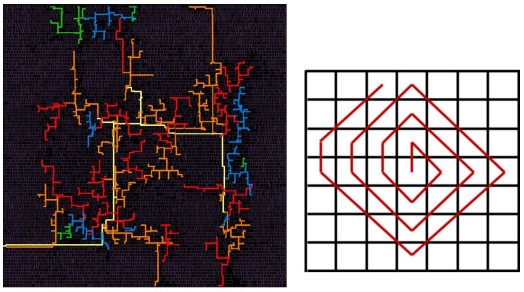

Figure 7: (a) Layout of scan enable (SE) nets. Different colors indicate different levels from the SE port. Yellow color = the topmost level; green color = the bottom level. (b) Illustration of spiral search for SE nets in neighbor grids.

Algorithm 2 Gating Insertion.

```
1: G_0 ← Σ_{c_i∈H} g_i;  n ← 0
2: while n == 0 || G_n < G_{n−1} · (1 − θ) do
3:    for all i := 1 to N do
4:       f(c_i) ← 0
5:    end for
6:    for all c_i ∈ H do
7:       f'(c_i) = g_i;  queue ← ∅;  queue.push(c_i)
8:       while queue ≠ ∅ do
9:          c_i ← queue.pop()
10:         for all c_i' ∈ fanin(c_i) do
11:            f'(c_i') ← SF(f'(c_i'))
12:            if c_i' ≠ flip-flop then
13:               queue.push(c_i')
14:            end if
15:         end for
16:      end while
17:      for all i := 1 to N do
18:         f(c_i) ← f(c_i) + f'(c_i);  f'(c_i) ← 0
19:      end for
20:   end for
21:   c ← cell with maximum f(c_i)  ∀i ∈ [1, N]
22:   Insert gating at output pin of c;  Update power
23:   Size/VT-swap c to minimize power w.r.t. timing constraints
24:   n++;  G_n ← Σ_{c_i∈H} g_i
25: end while
```

Note that here the cells are non-scan cells. Since we will not insert gating cells along the scan chain, we only visit non-scan cells when we traverse the netlist. In other words, we only traverse along the nets which are candidates for gating insertion. At each iteration, we first initialize to zero the gain value $f(c_i)$ of each cell c_i (Lines 3-5). We then start from each cell within the selected DVD hotspots and traverse backwards to calculate the gain values of their fanin cells, based on a sensitivity function $SF()$ (Lines 7-16). Due to large runtime of power simulation, it is practically infeasible to perform exhaustive search within the fanin cone of cells in a given DVD hotspot to search for the gating insertion locations.[4] In our work, we search for gating locations based on gain values of cells, which are calculated based on sensitivity functions. We study different sensitivity functions based on netlist structure, logic function of cell instances, etc. Figure 8 shows the performance of various sensitivity functions (their sensitivity functions are also shown in the figure) and an example of sensitivity function-based gain value propagation. Each gating location of the optimal gating insertion solution is achieved with an exhaustive search. We observe that sensitivity function $SF = f \cdot duty\_cycle$ offers the best solution quality, where *duty_cycle* indicates the probability that a particular input signal is not masked by any other signal. The duty cycle can be estimated by logic function and static probability of other input signals of the gate. For example, having a '0' signal on one input of an AND gate will mask the other input signal. Therefore, *duty_cycle* of an input signal of an AND gate is the probability of a '1' signal at the other input of the AND gate. We accumulate the gain values calculated based on cells within the selected DVD hotspots (Line 18). Last, we select the cell with maximum gain value to insert gating. Note that at this stage, we only insert gating cells in the netlist; ECO implementation steps (placement, routing) are executed only when all gating locations are determined. We perform sizing and VT-swapping of the inserted gating cells to minimize power overhead while satisfying timing constraints (Line 23). We terminate the iterative optimization when the power reduction ratio is less than θ (Line 2).

[4]One iteration of exhaustive search in the fanin cone of 100 cells in a small design (e.g., 15K instances) can take more than four hours.

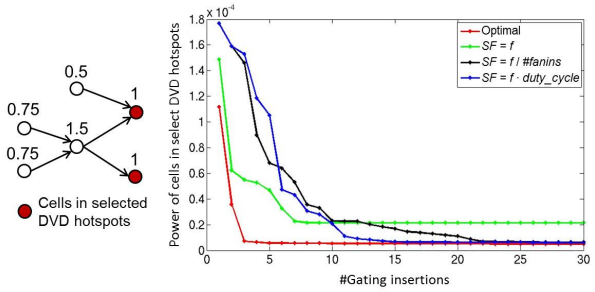

Figure 8: Left: an example of sensitivity function $SF = f/\#fanins$, where each cell within selected DVD hotspots has one unit of power. Right: performance of different sensitivity functions.

(3) ECO Optimization. Based on the identified gating insertion locations (in the netlist) and the design placement, we search for whitespaces near the gating cells and perform a matching optimization to determine the geometric locations for gating cell insertions.[5] We formulate the cost matrix using distances between the output pins and/or their fanout wire segments, and the nearby whitespaces. We then apply the Hungarian method [23] to perform the matching optimization. We perform ECO placement legalization and routing after gating insertion.

4. EXPERIMENTAL RESULTS

We perform experiments at 28nm LP foundry technology with dual-VT libraries. The supply voltage is 0.9V. We use four designs – DES, VGA, LEON3MP, NETCARD – from the ISPD-2012 contest [15] as our benchmarks. The benchmark information is shown in Table 2.[6] These designs are synthesized using *Synopsys Design Compiler vH-2013.03-SP3* [26] and then placed and routed using *Synopsys IC Compiler vI-2013.12-SP1* [28]. We use *Synopsys DFT Compiler vH-2013.03-SP3* to perform scan chain insertion. We set the maximum length of each chain to 250. We also perform scan compression in our implementation. We enable the DFT optimization options during placement and clock tree synthesis stages in *IC Compiler* to generate our initial scan chain solutions. We further use *Synopsys PrimeTime-SI vH-2013.06-SP2* and *Synopsys PT-PX vH-2013.06-SP2* [29] for timing and power analysis, with wire parasitics (SPEF) obtained from *IC Compiler*.[7]

We perform vectorless dynamic voltage drop (DVD) analysis using *ANSYS RedHawk* [24]. As inputs to DVD simulation,

[5]When there is no whitespace that satisfies the required minimum width, we either increase the region area to search for available whitespaces, or select whitespaces with smaller widths. However, these options will incur larger timing impact due to longer wirelength and/or placement legalization.

[6]We use the same clock period for scan shift. We note that the much smaller clock period for scan shift as compared to those of industrial designs is due to the simple clock tree structure and single-block implementations.

[7]We emphasize that our work does not entail any benchmarking of the commercial tools used for our experiments. No value judgment is intended by, or to be inferred from, our reported results. The tool versions used in our work are the latest available to us via university program subscription.

we report rise and fall arrival timing windows of all signal pins using *PrimeTime-SI* and report instance toggle rate and power information using *PT-PX*. Our DVD IR analysis is vectorless (with assumed 50% switching activity at test inputs), due to lack of open-source representative simulation vectors. To our understanding, this reflects common industry practice. We also note that our approach can be applied to scenarios with vector-based DVD analysis. We place power pads uniformly along the block periphery. Our scan ordering optimization is implemented in C++. Gating insertion flow is implemented in Tcl using *PT-PX* and *IC Compiler*. We conduct our experiments on a 2.5GHz Intel Xeon server.

Table 2: Benchmark information.

design	clock period (ns)	#inst	#scan chain
DES	0.85	74035	45
VGA	1.1	80412	78
LEON3MP	2.0	474108	445
NETCARD	1.8	428974	358

4.1 Scan Ordering

We perform scan reordering optimization at the post-routing stage to minimize the number of hold buffers. Table 3 shows our experimental results which include the number of hold buffers along scan chains, worst negative slack (WNS), total negative slack (TNS), total hold slack (THS) and total wirelength of the initial designs (orig) and of our optimized designs (opt). We observe that our optimization can remove up to 82% of hold buffers along scan chains (i.e., for the LEON3MP case). The optimized solution incurs negligible wirelength and timing penalties. We also observe wirelength reduction for large designs (e.g., for the NETCARD case).

Table 3: Scan ordering results.

		#Hold buffers	WNS (ps)	TNS (ns)	THS (ns)	Wirelength (*mm*)
DES	orig	**1296**	-21	-0.089	-0.101	765.9
	opt	**487**	-21	-0.081	-0.121	766.3
VGA	orig	**202**	-6	-0.019	-1.222	3087.9
	opt	**89**	-6	-0.018	-1.223	3089.7
LEON3MP	orig	**25581**	30	0	-0.734	11088
	opt	**4538**	30	0	-0.705	11084
NETCARD	orig	**30864**	-4	-0.004	-13.317	12729
	opt	**26887**	-4	0.0	-13.304	12720

4.2 Gating Insertion

We perform gating insertion to minimize the timing slack degradation due to DVD. Table 4 shows our experimental results, where ΔSlack indicates the scan timing slack degradation due to DVD; #Gating cells indicate the number of inserted gating cells; and DVD is the maximum DVD of the design. We observe that our optimization achieves up to 58% reduction of the slack degradation due to DVD (i.e., for the DES case). Our solution inserts only a small number of gating cells, and has minimal area overhead (e.g., <1%). Since the number of inserted gates is small, the corresponding power and area overheads, along with impact on DVD in function mode, are negligible. We further observe that the worst DVD value of a design is not necessarily correlated with slack degradation. In other words, it is the DVD on timing-critical scan cells, rather than the worst DVD in the design, that is more critical to optimize; this has not been captured by previous works.

Table 4: Gating insertion results.

		ΔSlack (ps)	WNS (ps)	TNS (ns)	#Gating cells	DVD (mV)	Area (μm^2)
DES	orig	**60**	-21	-0.089	-	85	79662
	opt	**25**	-21	-0.079	36	84	79705
VGA	orig	**159**	-6	-0.019	-	93	120832
	opt	**118**	-6	-0.029	42	82	120873
LEON3MP	orig	**471**	30	0	-	129	699885
	opt	**383**	30	0	62	121	699969
NETCARD	orig	**576**	-4	-0.004	-	163	575869
	opt	**496**	-8	-0.008	111	147	576022

5. CONCLUSION AND FUTURE WORKS

In this work, we propose a comprehensive scan timing optimization during late-stage IC implementation. We develop two optimization approaches: (i) scan reordering that is aware of clock skew and scan cell locations to remove hold buffers along scan chains, and (ii) gating insertion to minimize the DVD impact on scan timing slack. Our optimizations achieve up to 82% hold buffer reduction and 58% improvement of scan timing degradation due to DVD. Our future works include: (i) a more comprehensive scan ordering optimization to explore the tradeoff among wirelength, hold and setup timing of scan chains, (ii) a more comprehensive formulation of scan ordering cost (e.g., considering lockup latch insertion/removal when multiple scan clocks are driving cells along a scan subchain), (iii) co-optimization of gating insertion, scan ordering and test pattern generation to minimize the DVD impact on scan timing, (iv) DVD optimization during capture stage, and (v) a predictive model to determine DVD hotspots.

Acknowledgments

We thank Samsung Electronics for support of this research.

6. REFERENCES

[1] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch and A. Virazel, "Design of Routing-Constrained Low Power Scan Chains", *Proc. DATE*, 2004, pp. 62-67.

[2] A. Cui, T. Yu, G. Qu and M. Li, "An Improved Scan Design for Minimization of Test Power under Routing Constraint", *Proc. ISCAS*, 2015, pp. 629-632.

[3] M. Elshoukry, M. Tehranipoor and C. P. Ravikumar, "A Critical-Path-Aware Partial Gating Approach for Test Power Reduction", *ACM TODAES* 12(2) (2007), pp. 17:1-17:22.

[4] M. Feuer and C. C. Koo, "Method for Rechaining Shift Register Latches Which Contain More Than One Physical Book", *IBM Technical Disclosure Bulletin* 25(9) (1983), pp. 4818-4820.

[5] S. Gerstendörfer and H.-J. Wunderlich, "Minimized Power Consumption for Scan-Based BIST", *Journal of Electronic Testing* 16(3) (2000), pp. 203-212.

[6] P. Gupta, A. B. Kahng and S. Mantik, "Routing-Aware Scan Chain Ordering", *Proc. ASP-DAC*, 2003, pp. 857-862.

[7] P. Gupta, A. B. Kahng and S. Mantik, "A Proposal for Routing-Based Timing-Driven Scan Chain Ordering", *Proc. ISQED*, 2003, pp. 339-343.

[8] M. Hirech, J. Beausang and X. Gu, "A New Approach to Scan Chain Reordering Using Physical Design Information", *Proc. ITC*, 1998, pp. 348-355.

[9] L.-C. Hsu and H.-M. Chen, "On Optimizing Scan Testing Power and Routing Cost in Scan Chain Design", *Proc. ISQED*, 2006, pp. 451-456.

[10] D. Jayaraman, R. Sethuram and S. Tragoudas, "Gating Internal Nodes to Reduce Power During Scan Shift", *Proc. GLSVLSI*, 2010, pp. 79-84.

[11] A. B. Kahng, I. Kang and S. Nath, "Incremental Multiple-Scan Chain Ordering for ECO Flip-Flop Insertion", *Proc. ICCAD*, 2013, pp. 705-712.

[12] E.L. Lawler, J.K. Lenstra, A. Rinnooy-Kan and D. Shmoys, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, Wiley, 1985.

[13] J. Lee, S. Narayan, M. Kapralos and M. Tehranipoor, "Layout-Aware, IR-drop Tolerant Transition Fault Pattern Generation", *Proc. DATE*, 2008, pp. 1172-1177.

[14] L. Y.-Z. Lin, C. C.-H. Liao and C. H.-P. Wen, "Synthesizing Multiple Scan Chains by Cost-Driven Spectral Ordering", *Proc. ASP-DAC*, 2013, pp. 540-545.

[15] M. M. Ozdal, C. Amin, A. Ayupov, S. M. Burns, G. R. Wilke and C. Zhuo, "ISPD-2012 Discrete Cell Sizing Contest and Benchmark Suite", *Proc. ISPD*, 2012, pp. 161-164, `http://archive.sigda.org/ispd/contests/12/ispd2012_contest.html`.

[16] N. Parimi, "Leveraging Physically Aware Design-for-Test to Improve Area, Power, and Timing", `https://www.cadence.com/rl/Resources/white_papers/PhysicallyAware_DFT_wp.pdf`

[17] J.-C. Rau, C.-H. Lin and J.-Y. Chang, "An Efficient Low-Overhead Policy for Constructing Multiple Scan-Chains", *Proc. ATS*, 2004, pp. 82-87.

[18] J. Schulze and R. Tally, "Mitigating Voltage Droop during Scan with Variable Shift Frequency", *Proc. ITC*, 2014, pp. 1-8.

[19] S. Seo1, Y. Lee, J. Lee and S. Kang, "A Scan Shifting Method based on Clock Gating of Multiple Groups for Low Power Scan Testing", *Proc. ISQED*, 2015, pp. 162-166.

[20] Andres R. Teene, "Clock Skew Insensitive Scan Chain Reordering", *US Patent* 6539509 B1, May 2003.

[21] J. T. Tudu, E. Larsson, V. Singh and H. Fujiwara, "Graph Theoretic Approach for Scan Cell Reordering to Minimize Peak Shift Power", *Proc. GLSVLSI*, 2010, pp. 73-78.

[22] W. Zhao, M. Tehranipoor and S. Chakravarty, "Power-Safe Test Application Using An Effective Gating Approach Considering Current Limits", *Proc. VTS*, 2011, pp. 160-165.

[23] Hungarian Algorithm, `http://www.informatik.uni-freiburg.de/stachnis/index.html`

[24] RedHawk User Guide. `https://www.apache-da.com/products/redhawk`

[25] ScanOpt. `http://vlsicad.ucsd.edu/GSRC/Bookshelf/Slots/ScanOpt/.`

[26] Synopsys Design Compiler User Guide, `http://www.synopsys.com`

[27] Synopsys DFT Compiler User Guide, `http://www.synopsys.com`

[28] Synopsys IC Compiler User Guide, `http://www.synopsys.com`

[29] Synopsys PrimeTime User Guide, `http://www.synopsys.com`