STEEL: A Technique for Stress-Enhanced Standard Cell Library Design

Brian T. Cline, Vivek Joshi, Dennis Sylvester, David Blaauw
Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI
{btcline,vivekj,dennis,blaauw}@eecs.umich.edu

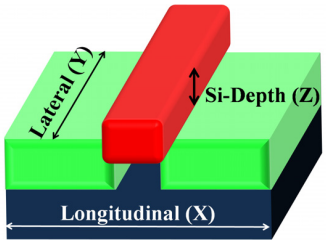
Abstract

Mobility degradation and device scaling limitations have led process engineers to develop new techniques that introduce mechanical stress in MOSFET channels, which results in enhanced carrier transport. New fabrication steps strive to increase carrier mobility which, consequently, increases both I_{on} and I_{off} in CMOS devices. However, most stress-enhancement techniques are dependent on layout parameters and their effects can be exploited within standard cell library design. In this work, we propose a new standard cell library design methodology that shares V_{DD} and V_{SS} source/drain connections across standard cell boundaries. Such sharing allows for increased channel stress in both the corresponding device as well as its neighboring devices. Using an industrial 65nm process and standard cell library, we show that our standard cell design methodology can be seamlessly integrated into current, state-of-the-art digital IC design flows. The new shared source/drain technique improves critical path delay by 11% on average over a number of benchmarks for only a ~35% increase in leakage. Furthermore, stress-enhanced standard cell libraries offer a superior power/delay tradeoff compared to dual-V_{th} across a wide range of operating points with reduced manufacturing costs. Specifically, our stress-enhanced library (with a single V_{th}) consumes ~2.5X less leakage than its dual-V_{th} counterpart.

1. Introduction

Over the past five years, numerous techniques have been developed in the semiconductor industry that combat increasing mobility degradation and allow for continued performance improvement in modern processes. These techniques typically achieve performance boosts by increasing the amount of mechanical stress induced within a MOSFET's channel. Applying mechanical stress to a MOSFET channel alters its valence and conduction bands, which results in changed carrier mobility and/or band scattering rates [1,2]. Increasing mobility by inducing stress counteracts mobility degradation and also increases the drain-to-source current (I_{DS}) in all device operating regimes. This increase in I_{DS}, however, actually depends on a number of layout parameters such as source/drain active area, contact placement, distance from STI structures, etc. Therefore, even if two different CMOS devices in a design have identical gate widths and lengths (W and L, respectively), their drive currents could vary by as much as ~20% due to the dependency of stress on other layout properties.

There are four main sources of stress in today's advanced CMOS processes, three of which exhibit strong layout dependencies. The first source, Shallow Trench Isolation (STI), was one of the earliest sources of mechanical stress to be extensively researched and modeled [3,4]. STI's impact on carrier mobility and drain current in both NMOS and PMOS devices is well known, and a number of works have examined various STI topics ranging from device-level modeling to efficient white-space management placement algorithms [5,6]. More recently, two additional sources of stress have been incorporated into semiconductor manufacturing: dual-stress nitride liners [7] and embedded SiGe (e-SiGe) [8]. Since electron and hole mobilities are improved by applying different types of stress in the X, Y, and Z directions (as shown in



	NMOS	PMOS
X	Tensile	Compressive
Y	Tensile	Tensile
Z	Compressive	Tensile

Figure 1. Preferred CMOS device stress types.

Figure 1), two separate nitride liners are required for NMOS and PMOS devices – a tensile liner and a compressive liner, respectively. In addition to the compressive nitride liner, PMOS stress is further enhanced by embedding a layer of SiGe within the source/drain regions of a device. The lattice mismatch between SiGe and Si introduces significant stress in a PMOS channel and can increase on and off currents (I_{on} and I_{off}) by as much as ~15% and ~3X, respectively [9]. The last principal stress source is the Stress Memorization Technique (SMT) used in NMOS transistors. Unlike the previous three sources discussed, the amount of stress induced by SMT is layout independent because it involves a uniform deposition, anneal, and removal of a stressed dielectric layer [10–12].

As alluded to in the previous paragraph, the first three stress sources – STI, nitride, and e-SiGe – are all dependent on common layout parameters in modern standard cells. The two most dominant layout properties that affect mechanical stress and are customizable within standard cell design are source/drain (S/D) active area and contact placement [10]. Larger S/D areas allow for greater amounts of e-SiGe (in PMOS devices) and nitride (in both types of devices), which enhances mechanical stress in the channel. Contact placement, however, disrupts the continuity of the nitride layer and, consequently, lowers the contribution of the nitride layer to channel stress. Hence, contacts placed farther away from the channel will increase the amount of nitride adjacent to the channel, enhancing channel stress [10]. Overall, the layout dependencies of stress are well documented [5,9–11,13], but to our knowledge, no work has focused on new standard cell library design techniques which exploit these dependencies. To date, the most recent work has only suggested layout guidelines for present-day standard cell library design [10,11].

Thus, in this work we propose a new standard cell design methodology that strives to fully exploit the layout dependencies of mechanical stress. Our library design methodology differs from previous mechanical stress work in that it employs a cell-level, library-wide enhancement technique that not only increases within-cell stress, but also increases cell-to-cell stress. Since most standard cells in a typical library have source/drain V_{DD} and V_{SS} ties adjacent to one or both edges of the cell, our new, stress-enhanced libraries share these ties across cell placement and route boundaries as illustrated in Figure 2. By sharing the V_{DD} and V_{SS} nodes, stress is enhanced in both the edge devices as well as their neighbors, increasing I_{on} and I_{off} by up to ~20% and ~3.5X, respectively for PMOS devices, and 7.5% and ~2X, respectively for NMOS devices. We verify our standard cell design by comparing to a commercial standard cell library in an industrial 65nm

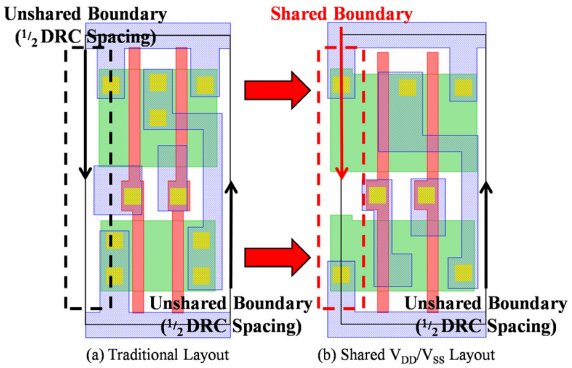


Figure 2. Traditional standard cell layout (a) versus proposed shared source/drain approach (b) for a 2-input NAND.

process. Device models and stress effects were calibrated using Tsuprem4 [14] (for simulating device fabrication) with Davinci 3D TCAD [15] simulations and industrial 65nm stress data [8]. By using our shared V_{DD}/V_{SS} approach, delay over a number of benchmarks improved by 11% on average for only a 35% increase in leakage.

The remainder of the paper is organized as follows. Section 2 describes the technique used in our proposed standard cell design methodology. Section 3 describes our standard cell design and its ease of integration within state-of-the-art VLSI design flows. Finally, Section 4 discusses our results and Section 5 concludes the paper.

2. A Technique for Enhancing Stress in Standard Cell Layouts

As stated in Section 1, mechanical stress in MOSFET channels depends on a number of layout parameters. However, the amount of mechanical stress in a typical CMOS device is not only a function of its own layout parameters (S/D area, contact placement, etc.), but also of its neighbors' parameters. Thus, NMOS and PMOS devices that share their S/D regions with other transistors have significantly higher channel stress (and, hence, drive current enhancement) than those at the edges of an active region (which are therefore bordered by STI), even for identical active area length and contact placement. For NMOS devices, this is mainly due to the fact that STI has a negative impact on the amount of tensile stress induced in the longitudinal direction, resulting in lower values of tensile stress in edge devices compared to devices towards the center. For PMOS devices, stress due to STI enhances channel stress, however, since e-SiGe has a much stronger contribution than STI, "center" PMOS devices also exhibit considerably higher channel stress as they are surrounded by more e-SiGe. Therefore, in the presence of mechanical stress, two devices with identical layout parameters (W, L, $L_{s/d}$, contact placement, etc.) may differ significantly in drive current, depending upon their positions in the layout (even when neglecting process variation).

From a standard cell design perspective, one would ideally avoid these stress-based variations and move to a more uniformly stressed standard cell to minimize context dependencies and performance uncertainty. By sharing the V_{DD} and V_{SS} source/drain ties across standard cell boundaries, we can effectively increase the number of "center" devices (devices with at least one other transistor on both sides) in a given standard cell. This results in higher channel stress in the devices of such cells, since all of the affected devices will have more neighbors (which means more e-SiGe, smaller STI regions, more nitride, etc.). Figures 3 (a) and (b) illustrate our shared V_{DD} and V_{SS} source/drain connection technique (referred to as the STEEL – STrEss Enhanced Library – technique for the remainder of the paper). Figure 3a depicts the traditional standard cell layout (for an inverter with two fingers) where the active area edge is placed at a location >=1/2 the design rule space from the standard cell boundary (the black rectangle that encapsulates the cell). However, since most standard cells in a typical library have at least one cell edge that is adjacent to a V_{DD} and V_{SS} S/D, we can share the connection between cells, effectively doubling the S/D active area and eliminating STI between the two cells. The edge devices achieve the largest increase using this approach – typically $L_{s/d}$ increases by >2X – and their induced channel stress now becomes more comparable to the stress in the "center" devices. Therefore, sharing the V_{DD} and V_{SS} connections between standard cells will not only lead to a more uniform distribution of channel stress, but will also improve the overall drive current of the standard cells (shown in the channel stress contour plots in the center of Figure 3). The actual "sharing" occurs in Figure 3b where the Metal-1 connections from V_{DD} and V_{SS} have been moved to the cell boundary. In this case, PMOS and NMOS drive currents increase by 13.5% and 6.3%, respectively, while leakage current increases by 2.8X and 1.6X. Furthermore, one of the strengths of STEEL is that it achieves these improvements in stress uniformity and drive current with no cell area increase (i.e., the area encapsulated by the black place and route boundaries in Figure 3 is identical for both cells (a) and (b)).

3. Implementation of STEEL in Standard Cell Design

In order to develop a 65nm STEEL standard cell library that accurately captured stress effects and ensured compatibility within existing

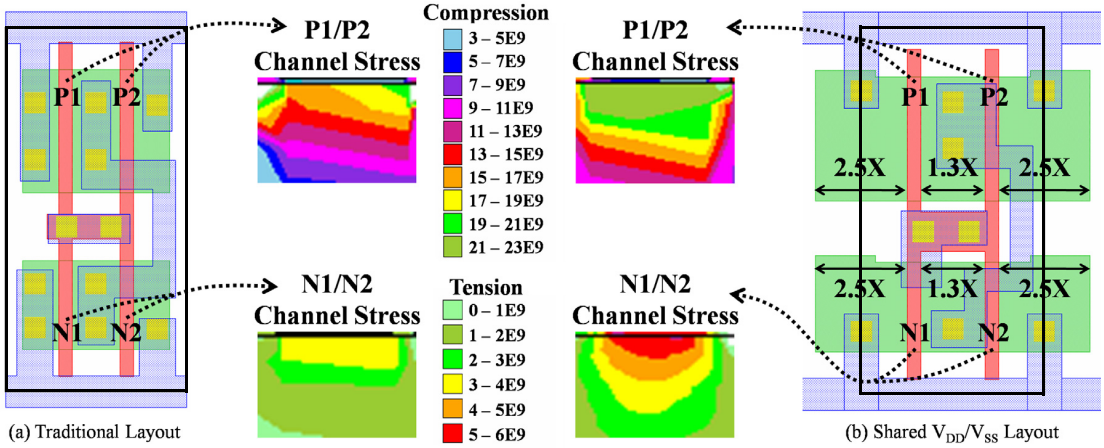


Figure 3. Impact of shared V_{DD}/V_{SS} approach on stress (measured in Pascals) in a two-finger inverter.
[Note: The NMOS and PMOS devices are symmetric here; the channel stress in N1 (P1) is identical to N2 (P2).]

VLSI design tools (e.g., synthesis tools, place and route tools, etc.), we created a design flow which is described below and illustrated in Figure 4. This design flow is executed on a cell-by-cell basis, and begins by capturing the effects of stress for each device within a cell. We use Tsuprem4 to simulate the fabrication steps and Davinci 3D TCAD to capture the stress-enhanced device parameters. Then, we calibrate our TCAD model with an HSPICE model and extract the effects of stress into one device-specific multiplication factor: the low-field mobility multiplier ($\mu_{0,STRESS\_MULT}$). This modified HSPICE model is then used within Signalstorm (a library characterization tool) to calculate the propagation delays and power consumption for a given cell, which is eventually output in Synopsys's Liberty file format. This .LIB file can be used in a number of industry standard synthesis and/or automated place and route (APR) tools.

The remainder of this section describes the STEEL standard cell design flow in more detail and concludes by describing common issues encountered and how they were resolved. We implemented our design flow on a reduced set of the most commonly used standard cells – 33 standard cells in total.

3.1. Tsuprem4 and Davinci Device Simulation

Our design flow begins by using Tsuprem4 to simulate the fabrication of a particular device and capture the process-induced stress. Davinci 3D TCAD tool is then used to capture device behavior under stress by solving for stress-based mobility enhancement equations. We used a TCAD device simulator for this work because currently, to our knowledge, there are no industry-standard device models that capture all of the layout-dependent effects of stress. BSIM4 captures only the STI-related stress impact on effective mobility (μ_{eff}), saturation velocity (v_{sat}), and threshold voltage (V_{th}). However, previous work has found that other layout parameters also play a critical role in determining the amount of mechanical stress induced in a channel [10]. Therefore, to capture these effects we simulate each standard cell in Tsuprem4 and Davinci, and extract the new, stress-enhanced low-field mobility (μ_0) at $V_{GS} = V_{DD} = 1V$ and $V_{DS} = 50mV$. By comparing a device's stress-enhanced mobility to its mobility without stress (the same TCAD simulation with the stress-analysis disabled), we can determine a device-specific scalar multiplier for μ_0: $\mu_{0,STRESS\_MULT}$. This multiplier is then used in our BSIM4 HSPICE model, described next.

3.2. Stress-Enhanced BSIM4 HSPICE Model

After calibrating Davinci device simulations to 65nm industrial HSPICE models (by matching I_{on} and I_{off}), we adjust the BSIM4 model so that the low-field mobility multiplier, $\mu_{0,STRESS\_MULT}$, is included as a possible input parameter for both PMOS and NMOS devices. We simply scale the old value of μ_0 by the multiplier: $\mu_0 = \mu_{0,OLD} \cdot \mu_{0,STRESS\_MULT}$. Simultaneously, since our Davinci models already capture all of the sources of mechanical stress, we temporarily turn off the BSIM4 stress models for μ_{eff}, v_{sat}, and V_{th} by setting the stress effect parameters for mobility degradation/enhancement (*KU0*), saturation velocity degradation/enhancement (*KVSAT*), and threshold voltage shift (*KVTH0*) to zero. The resulting *I-V* fit for minimum-sized NMOS and PMOS devices is shown in Figure 5, which verifies the accuracy of our model. For example, in these minimum-sized devices we find that our modified HSPICE device models incur an average root mean square error in saturation current of ~3μA and ~0.7μA for the NMOS and PMOS devices, respectively. These HSPICE device models eventually serve as the basis of our standard cell library characterization.

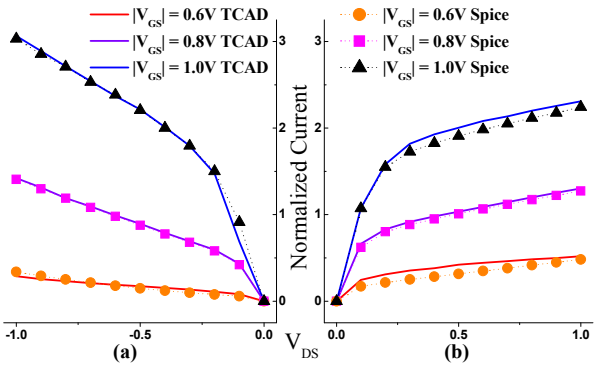


Figure 5. (a) PMOS and (b) NMOS *I-V* plots: Davinci vs. HSPICE.

3.3. Standard Cell Library Characterization

To make our new standard cell library compatible with existing digital, integrated circuit (IC) design flows, it is essential to be able to characterize the new standard cells and determine typical gate level parameters such as pin capacitance, propagation delay, dynamic and leakage power consumption, etc. To achieve this, we input our modified HSPICE models into Cadence's Signalstorm delay calculator. Signalstorm then simulates our stress-enhanced gates over a number of output-loading and input-slew combinations and finally generates a LIBERTY characterization file (.LIB). The .LIB file generation is the last step in the STEEL standard cell design flow and it enables the use of these new libraries within synthesis and APR tools with minimum additional overhead (described in more detail in Section 4.1).

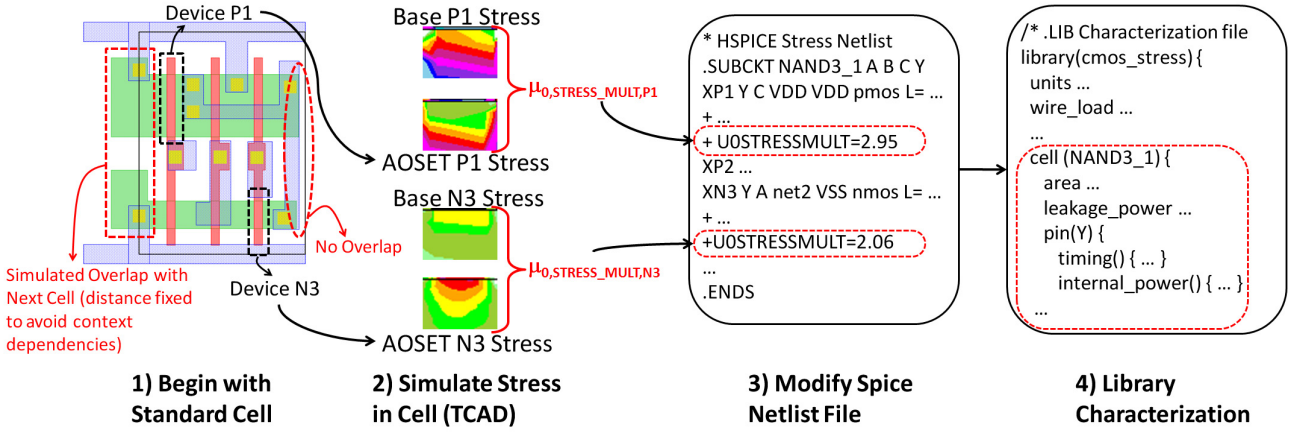


Figure 4. STEEL characterization flow.

3.4. Implementation Decisions in STEEL

There were several design decisions that needed to be resolved while creating a STEEL standard cell library. The first decision addressed the number of variants that could exist at an abutted boundary. These variants occur because many of the standard cells in a typical library cannot share the V_{DD} and V_{SS} connections at both edges of the cell. Instead, the adjacent S/D node is connected to some other net (e.g. the output node in a minimum-sized Inverter or NAND gate). For instance, refer to the 2-input NAND layout in Figure 2b. The NMOS drain on the right-hand side is tied to the output, Y. Therefore, this drain cannot be shared at the boundary with any arbitrary cell in a design whose left NMOS S/D is not connected to the same net. In this case, the PMOS source tied to V_{DD} could be shared, but only with a cell that has the same configuration (shared PMOS, unshared NMOS) or a custom "Filler" cell designed for the "shared PMOS, unshared NMOS" case. Therefore, to keep the number of edge variants small, we implemented two types of standard cell edges: shared or unshared. If either the NMOS or PMOS S/D is not connected to V_{SS}/V_{DD}, respectively, then that edge of the cell is designed to be completely unshared. STEEL consequently has three different types of cells:

- Cells with both edges "shared" (such as the one in Figure 3b).
- Cells with one "shared" edge and one "unshared" edge (previously discussed and illustrated in Figure 2b).
- Cells with both edges "unshared" (similar to the layout shown in Figure 2a).

Each standard cell in the library corresponds to only 1 of these 3 types, with the exception of inverters and buffers. To ease APR we designed two versions of inverter and buffer cells, one with the maximum number of shared connections and one with zero shared connections (both edges "unshared"). The "unshared" inverter and buffer cells reduce the placement/routing complexity involved during buffer insertion. For additional details of using STEEL libraries within APR, refer to Section 4.1.

The second design decision made was that a cell edge of a certain type (either shared or unshared) could only be abutted with an edge of the same type. In our implementation, we chose to let the APR tool handle this by passing it an additional set of constraints:

- Only abut "shared" edges with "shared" edges.
- Only abut "unshared" edges with "unshared" edges.

Details regarding the additional overhead needed to use STEEL within APR is included in Section 4.1.

The final implementation detail is a by-product of the layout dependency of stress. Since we are essentially extending the active area between standard cells, differing amounts of active overlap for different combinations of cells could significantly change the I_{on} and I_{off} currents for a given device. Therefore, context dependencies could easily arise if the STEEL library is not carefully designed. To illustrate this problem, consider the example in Figure 6, which shows two overlap cases for transistor, T_1. In the first case, the standard cell containing T_1 is placed next to a cell whose nearest device is T_2. The distance, X_{12}, between these two transistors corresponds to the active area length, $L_{s/d}$, of this source/drain region and directly affects the amount of stress induced in both T_1 and T_2. However, in the same design, the same cell type that contains T_1 is used again, but this time is placed next to T_3 and the S/D length increases by 1.3X. In this simple

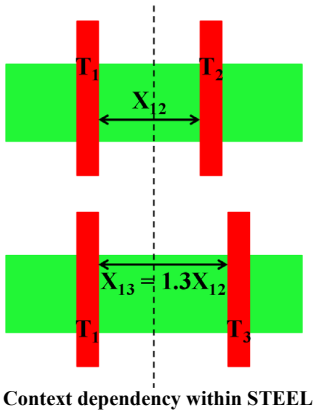


Figure 6. Context dependency within STEEL designs.

example, this 30% change will increase the drive current by ~10% (if we assume T_1, T_2, and T_3 are PMOS devices), which is substantial.

One way to handle this context dependency is to characterize the particular device, T_1 for every possible $X_{1,N}$ that could exist by abutting it next to any other "shared" edge in the library. However, since an industrial library typically has many hundreds of cells, this leads to an infeasibly large number of characterizations. Instead, we chose to fix the distance $X_{M,N}$, such that each device T_M and T_N are placed $0.5X_{M,N}$ away from the boundary. We selected a value for $X_{M,N}$ that achieved ~20% and ~8% increases in PMOS and NMOS I_{on} (for the edge devices) and increased I_{off} by ~4X and ~2X, respectively.

4. Experimental Results

In order to determine the strengths of the STEEL design methodology, we compared it to two industry design flows: single-V_{th} (using regular-V_{th}, or RVT, cells) and dual-V_{th} (using both RVT and low-V_{th}, or LVT, cells). These comparisons are included in Sections 4.2 and 4.3, respectively. We also describe a simple assignment technique in Section 4.4 which only applies the advantages of STEEL to critical cells, improving leakage at slower delay points or in unbalanced circuits. However, before we examine our results, we begin by briefly discussing how our place and route tools were configured to handle the STEEL library.

4.1. APR using STEEL Libraries

As mentioned previously in Section 3.4, the various standard cell edge types (either "shared" or "unshared" in our implementation) in the STEEL library add a small amount of complexity to cell placement. To minimize this complexity, we enforced a few additional constraints within the APR tool (discussed in Section 3.4). We accomplished this through a custom Tool Command Language (TCL) script that was designed and run within Cadence's APR tool, Encounter. Essentially, the script steps through each placed standard cell in the design, starting with the top, leftmost cell, and continues from left to right across a single core row before proceeding to the next row down. As the script traverses the standard cell row (from left to right), it checks the adjacent cell edges. If the edges match, the TCL script moves to the next cell. However, if the edges do not match, the script checks if the opposite side of the right cell matches the current cell edge. If it does, the script flips the cell and continues. If neither sides match, then a filler cell is placed in between the cells, to ensure that design rules are satisfied. The penalty incurred is typically minimal, and we found that even with row utilizations of up to ~85%, the STEEL library can be placed and routed using the same floorplan and dimensions as the traditional standard cell libraries.

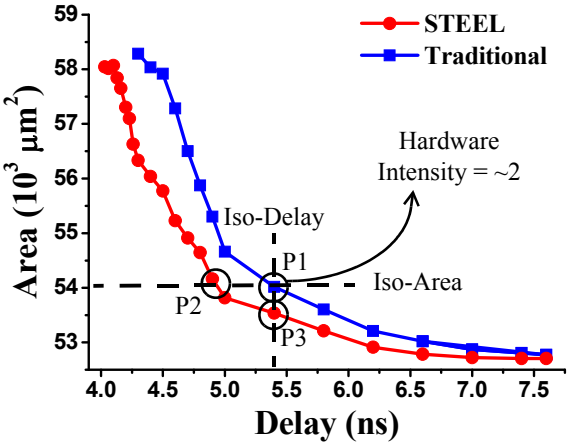


Figure 7. Area versus delay for the Viterbi decoder benchmark.

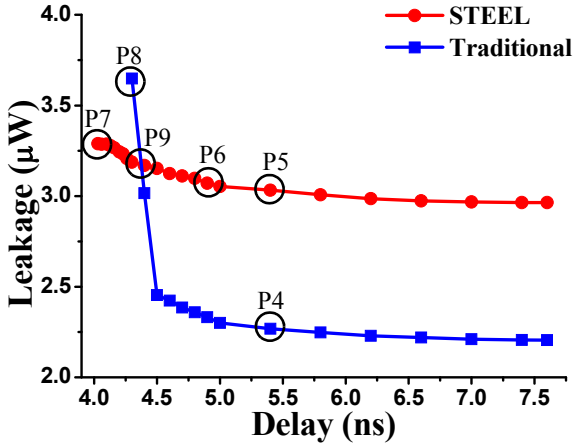


Figure 8. Leakage versus delay for the Viterbi decoder benchmark.

4.2. STEEL versus Regular-V_{th} Results

We begin our analysis by comparing the area, leakage power, and delay of STEEL designs to their traditional, single-V_{th}-based equivalent. The basis of our comparison was an industrial 65nm RVT library. Both libraries were characterized using the stress-enhancement models and flow described in Section 3 and pictured in Figure 4. With the new .LIB files, we were able to synthesize and place and route a variety of benchmarks using both libraries. In total, we implemented the physical design of 10 benchmarks whose gate count ranged anywhere from ~100 to ~60,000 standard cells. Each benchmark was synthesized at a number of different constraints to determine both the area-versus-delay tradeoff, as well as the leakage-power-versus-delay tradeoff.

For example, Figures 7 and 8 illustrate these tradeoffs for a Viterbi Decoding circuit (with ~25,000 gates). There are a few interesting points to notice from these plots. First of all, the STEEL version has a better area/delay tradeoff characteristic. Hence, for the same critical path delay, the STEEL implementation will consume less area. This improvement occurs because the STEEL cells are identical in area to the traditional cells, but have reduced propagation delays (due to the stress-enhancement achieved through active-area overlap). Consequently, the physical design tools do not have to size a given STEEL path as aggressively as its corresponding traditional path implementation, leading to reduced area consumption.

Alternatively, if you analyze the circuits at the same value of area (iso-area), STEEL typically reduces delay by 11% (again, due to the stress-enhancement achieved without increasing area). Notice that even at the minimum delay point on the traditional curve, the STEEL library still provides ~9% improvement. Furthermore, if you examine the leakage tradeoff in Figure 8, leakage power in the Viterbi decoder increases rapidly on the left side of the plot (toward smaller values of delay). This is due to the fact that to meet these tight timing constraints, the synthesis tool must size up the majority of the gates in the design, which increases leakage dramatically. Since stress-enhanced gates are designed to primarily give improvements in I_{on} (and therefore, delay), this region of the curve is where the STEEL library prefers to operate.

The full set of benchmark results compared to the single-RVT library is included in the seven leftmost columns of Table 1. This table was constructed using the following procedure. For each benchmark, we analyzed the area/delay tradeoff curve for the traditional 65nm implementation to determine the delay where hardware intensity was ~2. Hardware intensity was originally proposed in [16] as a power versus delay metric. In this work we use a modified version of hardware intensity that compares area and delay. Thus, for the remainder of the paper, hardware intensity is defined as the percentage change in area over the percentage change in delay. Next, the corresponding values of area and delay (whose hardware intensity is ~2) were used to determine the iso-area and iso-delay comparisons made against the STEEL implementation. For example, in the Viterbi decoder benchmark, the point on the area/delay curve (for the traditional implementation) where the hardware intensity was equal to 2 is labeled point "P1" in Figure 7. The corresponding delay improvement that we achieve using STEEL is given in Column 3 of Table 1. For the Viterbi decoder, this value is calculated by comparing the delays at "P1" and "P2" (in Figure 7). Similarly, area improvement – Column 4 in Table 1 – is calculated by comparing the areas at "P1" and "P3". Next, Columns 5 and 6 include the leakage power increase incurred by the STEEL implementation. These values are calculated for the Viterbi circuit by comparing the leakage values at "P4" and "P5" (from Figure 8) for the iso-delay case, and comparing "P4" with "P6" for the iso-area column. Finally, the decrease in the minimum critical path delay is noted in Column 7. This value for the Viterbi decoder is determined by comparing the delay at points "P7" and "P8" in Figure 8. The remainder of Table 1 is discussed in Section 4.3.

Generally, we discovered that for iso-area, the STEEL implementation achieves average delay improvements of 11% while leakage only increases by 35% on average. Additionally, we found that the STEEL-based benchmarks successfully synthesized at a minimum delay value that was, on average, 9.1% less than the traditional minimum delay.

4.3. STEEL versus Dual-V_{th} Results

In addition to a significantly improved area-delay tradeoff for STEEL versus a single-V_{th} standard library, we now demonstrate that *STEEL provides superior performance with a single-V_{th} over a traditional dual-V_{th} library* for the majority of operating points where dual-V_{th} would be of interest. This arises due to the improved I_{on} vs. I_{off} tradeoff using stress enhancement compared to using low-V_{th} devices [11] and indicates that STEEL simultaneously offers a better power/performance envelope and lower manufacturing costs over dual-V_{th}. Figure 9, for example, illustrates the leakage/delay curve for the dual-V_{th} implementation of the Viterbi decoder (notice its similarity to Figure 8). The slower part of the curve (delay > 4.26ns) is actually identical to Figure 8, due to the fact that only RVT cells are used in the design until the delay constraint becomes less than or equal to 4.26ns. In the region of interest for STEEL, we found that the leakage crossover point (where dual-V_{th} leakage becomes greater than STEEL) typically occurred between the most tightly constrained RVT design (with

Table 1. Design improvement obtained using STEEL (compared against single-V_{th} and dual-V_{th} implementations).

Circuit	Number of Gates	% Delay Improvement (Iso-area)	% Area Improvement (Iso-delay)	Leakage Increase (Iso-delay)	Leakage Increase (Iso-area)	% Delay Improvement Beyond Min. Critical Path	$\frac{\text{Dual-Vth Leakage}}{\text{STEEL Leakage}}$ †
c432	143	**18.6%**	2.4%	1.41	1.46	12.5%	**2.95**
c1908	265	**6.00%**	6.7%	1.11	1.22	9.4%	**4.88**
c880	291	**16.5%**	2.6%	1.34	1.39	8.1%	**2.37**
c2670	489	**9.2%**	1.1%	1.35	1.34	4.4%	**0.85**
c3540	921	**9.0%**	2.1%	1.33	1.36	9.0%	**2.08**
c7552	1264	**11.1%**	0.9%	1.27	1.28	12.5%	**2.97**
c5315	1275	**15.5%**	1.5%	1.33	1.34	13.3%	**2.78**
c6288	1703	**7.1%**	0.4%	1.27	1.28	8.2%	**3.52**
Viterbi Dec.	25287	**8.0%**	1.1%	1.33	1.35	6.3%	**2.06**
Ethernet	66310	**8.6%**	0.1%	1.50	1.50	7.5%	**0.79**
AVERAGE		**11.0%**	1.9%	1.32	1.35	9.1%	**2.53**

† The dual-V_{th} leakage increase over STEEL is calculated at iso-delay for the minimum critical path delay of the STEEL design.

zero LVT cells) and the dual-V_{th} implementation that used the minimum number of LVT cells needed to satisfy timing. Since the LVT cells in our industrial library typically increased leakage by ~20X, the minimum leakage for the dual-V_{th} case occurred at the timing constraint that used the minimum number of LVT cells. Even at this minimum leakage point for dual-V_{th} (where the number of LVT cells is only a small percentage of the total number of cells, <5%), the substantial leakage increase per low-V_{th} cell caused this minimum-leakage, dual-V_{th} implementation to almost match the leakage increase incurred by STEEL. Over all of the benchmarks, we found that even at the minimum dual-V_{th} leakage, dual-V_{th} only showed a 2.9% average savings in leakage over STEEL. Furthermore, by the time the STEEL implementations reached their minimum delay, the dual-V_{th} leakage had increased to ~2.5X the average value of STEEL leakage (displayed in the last column of Table 1). An example point for the Viterbi decoder circuit for this value is shown in Figure 9.

Since the STEEL implementations can typically provide up to ~10% delay improvements over single-V_{th} designs while consuming only a fraction of the leakage power of dual-V_{th}, STEEL can provide more optimal designs in two ways. First, for designs that only need moderate delay improvements – less than 10% – STEEL can be used to achieve these improvements. By utilizing the STEEL standard cells, the designer would not only reduce leakage (as compared to the dual-V_{th} implementation), but would also dramatically reduce manufacturing costs, since the second threshold voltage mask would not be needed. Alternatively, STEEL could also be used in conjunction with the dual-V_{th} approach to achieve more optimal designs (in terms of area and power). Since typical dual-V_{th} processes only provide coarse-grain threshold voltage values, some standard cells in a path might be sub-optimally assigned if they do not need the full performance enhancement provided by moving to a lower V_{th} value. For these cells, the STEEL versions would be more appropriate, since they can obtain more fine-grained performance improvements and will fill some of the performance space between V_{th} values. Additionally, by designing LVT STEEL cells, delay improvement can be extended beyond the performance of dual-V_{th}.

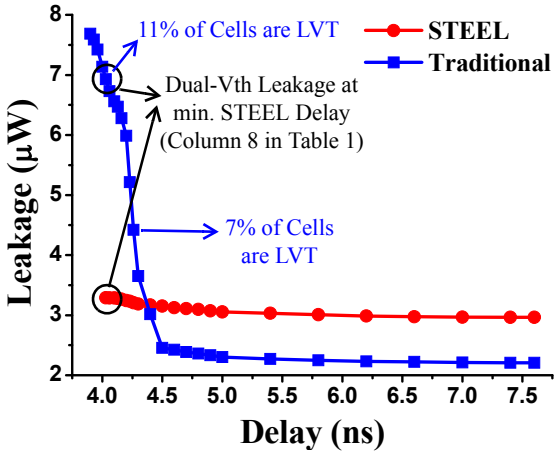


Figure 9. Viterbi decoder leakage vs. delay for dual-V_{th} case.

4.4. Intelligent STEEL-Cell Assignment

One interesting discrepancy that we found during this work was the fact that in our largest circuit, an ethernet controller, the STEEL design did not outperform the dual-V_{th} implementation. In fact, out of the 10 benchmarks, the ethernet circuit was the only case where we did not obtain improvements in leakage versus dual-V_{th}. To understand this phenomenon, we analyzed the structure of the ethernet controller and made some interesting observations:

- Even though the ethernet controller used a large number of standard cells, its paths were not balanced and the number of critical paths only represented a small fraction of the total number of paths.
- Out of ~66,000 standard cells, the dual-V_{th} design only used 285 LVT cells (<1% of the total) to meet the minimum timing constraint achieved using STEEL.

With this knowledge, it was clear why the STEEL implementation did not improve upon the dual-V_{th} case. Since we had not previously employed any delay/leakage optimization in our approach, the ~1.3X STEEL average leakage increase per standard cell occurred in each of the ~66,000 standard cells, whereas the ~20X leakage increase per LVT cell only occurred in <1% of the total cells. Therefore, while the STEEL designs outperformed dual-V_{th} in the majority of our experi-

ments, it was clear that exploring intelligent assignment schemes would be beneficial to our work, both to improve the STEEL leakage performance in unbalanced designs (as compared to dual-V_{th}), as well as achieve leakage values closer to the RVT-based designs.

So far, we have reported the STEEL results for the case where we use our stress-enhanced library uniformly across a given design (i.e, every gate in the circuit is assigned to its stress-enhanced version). However, not all of the gates in a circuit need performance enhancement to meet timing for a given delay constraint. These non-critical gates only add to the leakage overhead, and as a result we observed that the STEEL designs had larger leakage than their single-V_{th} counterpart, even at larger values of delay (more relaxed delay constraints). Thus, there is ample scope for intelligent assignment of stress-enhanced cells, where the traditional RVT library is used in conjunction with STEEL, and the STEEL cells are only assigned to timing critical gates. An intelligent cell assignment scheme will substantially reduce the leakage overhead but maintain similar improvements in delay. The benefits of this technique derive from the fact that only a fraction of total number of gates in a circuit are timing critical. Replacing only the critical gates with the leakier, higher-performance versions will result in significantly lower leakage increases, as compared to the case where all of the gates are replaced.

As a further investigation into the scope of intelligent assignment, we perform a simple experiment where we replace only the top ~10%, timing critical gates in a circuit with their stress-enhanced versions. We perform this experiment at the same hardware intensity point (discussed previously) on the area-versus-delay curve for the traditional RVT library, and compare the delay improvement and leakage overhead numbers to the case where stress enhancement was used in every cell (Column 3 and Column 6 of Table 1, respectively). Figure 10 shows the percentage improvement that we observe using intelligent assignment, as compared to the uniform-replacement ("Original" STEEL) scheme. Ideally, we would prefer to obtain all of the delay improvement achieved in the previous section (i.e., achieve 100% of the typical 11% delay improvement over RVT), while reducing the percentage leakage increase to 0% (i.e., matching the RVT leakage). As shown in the figure, we can get >80% of the "Original" delay improvement through selective replacement, while incurring a much smaller increase in leakage. The selective scheme typically reduces the uniform STEEL leakage increase by ~90%. From Figure 10, observe that the leakage number for the ethernet benchmark is exceptionally small because, despite its large size (~66,000 gates), the number of timing critical gates is very small (as mentioned previously). Thus, to achieve 80% of the "Original" improvement, only 625 gates need to be replaced with their stress-enhanced version (less than 1% of the total gates), which results in substantial leakage savings that is comparable with dual-V_{th}.

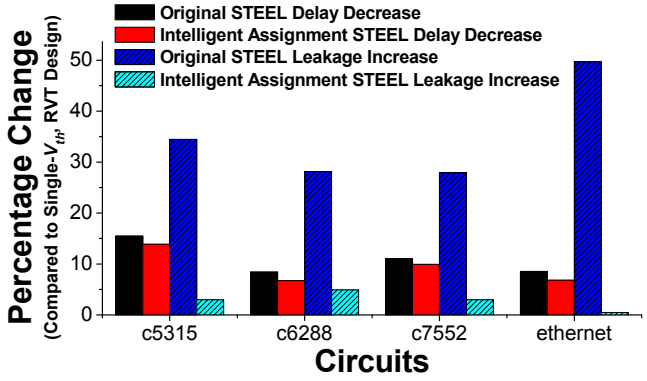


Figure 10. Impact of intelligent STEEL assignment on delay and leakage.

Intelligent replacement schemes like this approach allow STEEL to maintain its advantage over dual-V_{th}, even for designs that are extremely unbalanced (such as the ethernet benchmark). Additionally, this approach can be used to improve leakage power consumption within any STEEL design (especially for relaxed delay constraints). This means that the leakage for the STEEL technique will approach that of the traditional RVT library, especially at delay constraints located to the right of the leakage crossing point (e.g., all of the STEEL leakage values to the right of point "P9" in Figure 8 will be much closer to RVT).

5. Conclusion

In this work, we proposed STEEL, a new standard cell library design technique for modern stress-enhanced semiconductor processes. STEEL fully exploits the layout dependencies of stress. By designing the STEEL standard cells to share the V_{DD} and V_{SS} source/drain connections across cell boundaries, one can achieve drive current improvements of up to 20%. While implementing the proposed standard cell approach in a number of benchmark circuits, we demonstrated average delay reductions of 11% with only a 35% average increase in leakage, compared to single-V_{th} implementations. Additionally, STEEL-based circuits typically achieved a ~2.5X reduction in leakage when compared to dual-V_{th} designs. This implies that for designs requiring an 11% delay improvement (or less) beyond a single-V_{th} implementation, STEEL can provide this improvement for a smaller leakage penalty as well as much lower manufacturing costs compared to dual-V_{th}. Orthogonally, STEEL can also be used in conjunction with dual-V_{th} to provide more optimal designs (in terms of both leakage and delay).

References

[1] F. Andrieu et al., "Experimental and Comparative Investigation of Low and High Field Transport in Substrate- and Process-Induced Strained Nanoscale MOSFETs," in *Proc. VLSI Tech. Symp. Tech. Dig.*, pp. 176–177, June 2005.

[2] K. Mistry et al., "Delaying Forever: Uniaxial Strained Silicon Transistors in a 90nm CMOS Technology," in *Proc. VLSI Tech. Symp. Tech. Dig.*, pp. 50–51, June 2004.

[3] G. Scott et al., "NMOS Drive Current Reduction Caused by Transistor Layout and Trench Isolation Induced Stress," in *IEDM Tech. Digest*, pp. 827–830, 1999.

[4] R. A. Bianchi et al., "Accurate modeling of trench isolation induced mechanical stress effects on MOSFET electrical performance," in *Proc. IEDM*, pp. 117–120, 2002.

[5] M. V. Dunga et al., "Modeling Advanced FET Technology in a Compact Model," in *IEEE Trans. on Elect. Dev.*, Vol. 53, pp. 1971–1978, Sept. 2006.

[6] A. Kahng et al., "Exploiting STI Stress for Performance," in *Proc. ICCAD*, pp. 83–90, Nov. 2007.

[7] H. S. Yang et al., "Dual stress liner for high performance sub-45nm gate length SOI CMOS manufacturing," in *Proc. IEDM*, pp. 1075-1077, 2004.

[8] Z. Luo et al., "Design of high performance PFETs with strained si channel and laser anneal," in *Proc. IEDM*, pp. 489–492, 2005.

[9] V. Chan et al., "Strain for CMOS performance Improvement," in *Proc. IEEE CICC*, pp. 667–674, Sept. 2005.

[10] V. Joshi et al., "Stress Aware Layout Optimization," in *Proc. ISPD 2008*, pp. 168–174, April 2008.

[11] V. Joshi et al., "Leakage Power Reduction Using Stress-Enhanced Layouts," in *Proc. 45th Design Automation Conference*, pp. 912–917, June 2008.

[12] K. Ota et al., "Novel locally strained channel technique for high performance 55nm CMOS," in *Proc. IEDM*, pp. 27–30, 2002.

[13] K. Su et al., "A Scaleable Model for STI Mechanical Stress Effect on Layout Dependence of MOS Electrical Characteristics," in *Proc. IEEE 2003 CICC*, pp. 245–248, Sept. 2003.

[14] Manual, Synopsys TSUPREM4, Version 2007.03.

[15] Manual, Davinci 3D TCAD, Version 2005.10.

[16] V. Zyuban et al., "Unified Methodology for Resolving Power–Performance Tradeoffs at the Microarchitectural and Circuit Levels," in *Proc. ISLPED*, pp. 166–171, Aug. 2002.