Stress Aware Layout Optimization Leveraging Active Area Dependent Mobility Enhancement

Ashutosh Chakraborty, *Student Member, IEEE,* Sean X. Shi, and David Z. Pan, *Senior Member, IEEE*

***Abstract*—Starting from the 90 nm technology node, process induced stress has played a key role in the design of high-performance devices. The emergence of source/drain silicon germanium (S/D SiGe) technique as the most important stressing mechanism for p-channel metal-oxide-semiconductor field-effect transistor devices has opened up various optimization possibilities at circuit and physical design stage. In this paper, we exploit the active area dependence of the performance improvement achievable using S/D SiGe technology for late stage engineering change order (ECO) timing optimization. An active area sizing aware cell-level delay model is derived which forms the basis of linear program based optimization of a design for achieving maximum performance or target performance under a timing budget. To control the magnitude of layout perturbation and ensure predictable timing improvement, a set of physical constraints for active area sizing is proposed. Further, an efficient minimum movement legalization algorithm is proposed to remove the overlaps caused by active area sizing of timing critical cells. Results on a wide variety of benchmarks show consistent reduction in the cycle time by up to 6.3%. Predictability of the performance improvement achievable as well as resultant minuscule layout changes make our technique very attractive for late stage ECO optimization and design closure.**

***Index Terms*—Charge carrier mobility, circuit optimization, integrated circuit layout, layout.**

I. Introduction

THE LAST FOUR DECADES have witnessed a tremendous integrated circuit (IC) performance increase and cost decrease. The most important enabler of the success of the semiconductor industry is the continuous shrinking of the transistor device size which delivers faster, cheaper, and smaller transistors in each generation. State-of-the-art technologies currently have gate lengths as small as a few tenths of a nanometer [1]. Scaling down to such small geometries brings forth a plethora of problems which make the process of device scaling exorbitantly expensive. These problems include lithography challenges, multi-million dollar mask costs, low yield ramp-up, and exponentially growing leakage current, and so on. In fact, since the introduction of the sub-100 nm technology node, further device scaling has become extremely costly and technologically challenging. The need of improved performance and the challenges of physical scaling has mandated exploration of alternate techniques which improve performance of a transistor without requiring physical scaling. These techniques include the use of III–V elements (such as Ga, As), application of advanced stress engineering, use of carbon nano-tubes (CNT), 3-D IC integration, multi-core systems, and so on. Currently, each of these techniques is a very popular field of ongoing research. Among these, advanced stress engineering has emerged as one of the most promising techniques of device performance increase because unlike techniques such as use of CNT and III–V elements, the use of stress engineering can be seamlessly integrated in existing chip manufacturing process without the need of new materials or drastic manufacturing process modifications. In fact, stress engineering has already been used by several companies such as IBM, Armonk, NY, Intel Corporation, Austin, TX, and AMD, Sunnyvale, CA, to boost their chip's performance. At the same time, stress engineering enhances *device* performance as compared to techniques such as 3-D IC integration which primarily increases the performance of a chip by reducing the *interconnect* length between communicating devices by placing them closer in the vertical dimension.

The application of mechanical stress alters the degeneracy of the energy bands in the channel of p- or n-channel metal-oxide-semiconductor field-effect transistor (PMOS or NMOS) devices which significantly changes the charge carrier mobility [2]. Mechanical stress can be of two types: compressive or tensile. In general, the mobility of a PMOS device increases when its channel is subject to compressive stress and decreases under tensile stress. For NMOS devices, an opposite trend is observed—tensile stress improves the device mobility whereas compressive stress degrades it. The objective of stress engineering is to intelligently apply the right kind of (i.e., compressive or tensile) mechanical stress to the channel of PMOS/NMOS devices in order to obtain higher performance. There are several mechanisms of imparting mechanical stress to a PMOS or NMOS device. The primary stressing mechanisms of PMOS devices are source/drain silicon germanium (S/D SiGe), the use of shallow trench isolation (STI), and stress liners. The primary stressing mechanisms for NMOS devices are STI, and stress memorization technique (SMT). A brief description of each of these stressing mechanisms is as follows. S/D SiGe techniques etch out the silicon from the source and drain (S/D) regions of

Manuscript received April 23, 2009; revised August 13, 2009 and February 24, 2010; accepted March 18, 2010. Date of current version September 22, 2010. This paper was recommended by Associate Editor N. Ranganathan.

A. Chakraborty and D. Z. Pan are with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78703 USA (e-mail: ashutosh@cerc.utexas.edu; dpan@ece.utexas.edu).

S. X. Shi is with Intel Corporation, Austin, TX 78746 USA (e-mail: sean.shi@mail.utexas.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2010.2061173

conventional transistor and epitaxially fill them with $Si_{1-x}Ge_x$ alloy where x denotes the proportion of germanium in the alloy. Larger lattice constant of the $Si_{1-x}Ge_x$ alloy compared to that of silicon creates compressive stress in the channel region. STI stress arises due to the ubiquitous method of using shallow trench of silicon oxide to isolate neighboring NMOS and PMOS devices. The stress generated due to STI is compressive in nature. Stress liners are highly stressed (compressed or tensile) silicon nitride liners which are deposited over the entire wafer to increase the carrier mobility [3]. SMT relies on depositing a stressed film on the wafer followed by rapid thermal anneal. Further details and a comprehensive review of these process-induced mechanical stress techniques can be found in excellent references such as [4], [5].

In this paper, we focus on the S/D SiGe technique which imparts compressive stress to the channel. The magnitude of the stress in the channel due to the S/D SiGe technique is critically dependent on the dimension of the active area around the device. We perform timing optimization of a design at the layout level by exploiting this active area dependent mobility of S/D SiGe type PMOS devices. In particular, we propose a new active area stretching aware cell delay model and demonstrate its linear nature for the range of current and future generation device's active area dimensions. Based on this model, our technique transforms the timing optimization problem into a linear program (LP) formulation which can be solved for minimum cycle time or for timing closure under a given timing budget. For removal of the overlaps resulting from active area stretching, a fast minimally intrusive linear time legalization algorithm is presented. Our methodology is particularly attractive for late-mode engineering change order (ECO) optimization since it directly works on the layout of the design. To enable dealing with hundreds of thousands of standard cells, this paper is performed at the standard cell level of abstraction and the internal structure of the cell (such as necessary shift in contact position, IO pin reshaping) are abstracted out. We assume that it is possible to size the active area of a cell in a continuous fashion. In practice, for standard cell based designs, the cell library can have several variants of a cell with different active area dimension. In such a case, the active area sizing results obtained from our technique can be discretized to match available cell sizes.

The rest of this paper is organized as follows. In the next section, we provide the background of the S/D SiGe technique and chip level timing closure followed by a discussion of the previous works in this field. Section III describes the derivation of our cell geometry aware active area dependent timing model. A set of rules that improve the predictability of timing optimization achieved by our method are presented in Section IV. Experimental flow and our setup are detailed in Section V. Section VI demonstrates the results of our proposed methodology. We conclude our paper in Section VII along with future research directions.

II. Background and Previous Work

The application of mechanical stress causes splitting and warpage of the six-fold degenerate conduction band for the charge carriers [6]. This causes redistribution of charge carriers in the conduction plane [7] toward the lowest energy band. Depending on the type of stress (compressive vs. tensile) and the crystal orientation, the lowest energy band can provide a path for the charge carriers to cross the device channel leading to decreased carrier transport mass and higher mobility. Increased on-current due to this higher mobility improves the performance of the device. Recently, [8] has reported the fabrication of PMOS devices with 200% improved mobility using multiple process induced stressing mechanisms. This phenomenal performance improvement is equivalent to that provided by several generations of technology scaling. The use of strain engineering is ubiquitous for high performance designs: IBM pioneered the strained silicon technology and has used it since PowerPC5; Intel has been using strain engineering from the Pentium-IV family onward; AMD 90 nm Opteron and Athlon onward have used strain engineering [9]–[11]. More recently, AMD revealed the recent advances in high-performance logic transistor engineering for their 45 nm technology node and outlined the ongoing efforts for 32 nm node and stress engineering plays a significant part in it [12].

Stress applied to a device can be either uniaxial or biaxial. As the name suggests, these correspond to stress being applied to the channel along one axis or two axes concurrently. The S/D SiGe technique causes uniaxial stress along the channel direction whereas STI and SMT techniques, which surround the device on all sides, result in biaxial stress. In general, biaxial stress is less beneficial than uniaxial stress due to several reasons: compared to uniaxial stress, biaxial stress suffers from increased misfits/dislocation in the silicon lattice, higher dopant diffusion (show stopper for shallow junction devices) and increased surface roughness. Additionally, the mobility enhancement of holes (for PMOS devices) is much lower when applying biaxial stress as compared to uniaxial stress [6]. For this reason, S/D SiGe stands out as an attractive uniaxial stressor mechanism. A S/D SiGe device can be fabricated by following the standard CMOS process with very few modifications. After the channel formation, the S/D regions of the conventional transistor are etched out. These regions are then epitaxially filled with $Si_{1-x}Ge_x$ alloy (SiGe) with the typical value of x in the range of 0.1–0.25 [13]. SiGe which now exists in drain and source region, owing to its larger lattice constant as compared to silicon lattice in the channel, compresses the channel region. The technique of S/D SiGe has an interesting property: the extent of mechanical stress imparted to the channel of the device is a function of the length of the S/D region (i.e., the active area) surrounding the channel. Increasing the active area implies a larger amount of stressor material which further increases the magnitude of compressive stress in the channel. Previous works [13] and [14] have reported the simulation and silicon measurement results which outlines the relation between the size of the active area and the improvement in the hole mobility.

There have been a lot of research works on understanding the reasons and quantifying the mobility improvement due to mechanical stress. However, there is dearth of literature which successfully *exploits this effect* for any kind of optimization beyond the device level. The conference version of our paper [15] is the first work to capture the device level S/D SiGe enhanced

mobility at the standard cell level and to perform layout-level timing optimization. The only other work that has exploited stress for chip-level layout optimization is [16] which proposes a STI width dependent SPICE model and guides the white space allocation to modulate the width of STI between different cells. Our paper is different from [16] since we deal with stress in novel *S/D SiGe* device due to stressors *inside* the cell whereas [16] deals with STI stress in conventional silicon devices arising from stressors *external* to the cell. As will be shown later, layout optimization with stressors inside the cell needs careful cell layout and geometry analysis as well as placement overlap control. More recently, [17] and [18] have analyzed the mobility improvement due to several concurrent stressing mechanisms. Based on this analysis, the authors propose a set of *cell* layout improvement guidelines such as expanding active area within the cell, relocation of contacts, and repositioning of the NMOS/PMOS device within the well. Our paper differs from these works since we perform layout optimization at the chip level considering timing paths passing through different cells whereas the focus of [17], [18] is optimization of a *single* cell without embedding it in the whole design. Active area dependent mobility has been exploited to reduce leakage power in [19]. The authors demonstrated that use of stress enhanced standard cells can achieve similar performance improvements as that are achievable with the use of low V_{TH} cell variants, without the exorbitant leakage power increase associated with the latter approach.

The motivation of our paper is as follows. One of the most formidable challenges of modern IC design flow is to achieve timing closure so that all the latch/flip-flop inputs and primary outputs of the design have non-negative slack. Due to rising interconnect delay and timing model inaccuracies at placement and physical synthesis stages, there are situations where the design does not meet timing requirements at the post-layout level. Such a design can be fixed by either re-synthesis, or by improving placement or routing of the design, but all these steps require a long turn-around-time. A reasonable assumption is that the design team has put the required optimization efforts during all phases so that at the post-layout level, the timing violation (observed vs. predicted) is sufficiently small. For such a practical case, there is critical need of late-mode optimization methodology which can handle ECO changes without requiring re-synthesis, placement and routing. Performance increase by active area size modulation presents itself as an exciting option for such a scenario. If at the post-layout stage there are timing violations, the active areas of cells lying on failing paths can be increased with minute penalty in area and negligible design time to obtain timing closure. Routing congestion for modern designs mandates leaving approximately 20%–40% white space in the chip layout. This white space is used in several steps such as filler cells, decap insertion. A part of this white space can be utilized to accommodate the increased active area. Using active area stretching as a late ECO fix has several benefits as compared to other techniques.

1) Active area stretching of a cell has minuscule impact on the timing of its fanin cell due to the fact that the gate size of the active area sized cell remains the same, thus the capacitive load seen by the fanin cell remains unchanged. This is in stark contrast with techniques like gate-sizing which produce a ripple effect on the timing of the fanin gates and in turn require sizing up the fanin gate.
2) Increasing the active area of a few cells introduces negligible power consumption increase (which is mainly due to diffusion area capacitance increase) and does not use extra routing resources as opposed to techniques such as aggressive buffer insertion which adds significant power consumption and uses extra nets.

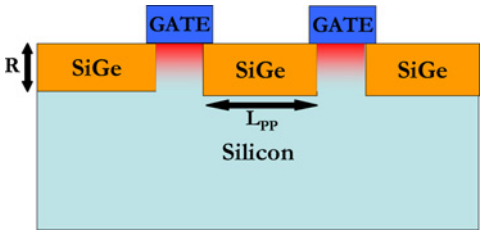

Fig. 1. Side view of a SiGe S/D device. Source/drain regions are epitaxially filled with SiGe which compresses the channel.

Increasing the active area of some of the devices can break the fixed poly-pitch paradigm under which all the poly gates are equidistant. For the design scenarios where this paradigm is strict, our optimization method should be applied only for poly gates at the two extremes of the cell so as to maintain fixed poly-pitch inside the cell. For more flexible scenarios, one may allow non fixed-pitches for very few timing critical cells and apply heavy resolution enhancement techniques for their printability.

III. Stretching Aware Timing Model

Consider the side-view of two (series connected) SiGe S/D PMOS devices in Fig. 1 shown without spacer (needed for lightly doped drain region to suppress short channel effects) for the sake of clarity. Note that the only difference as compared to a traditional PMOS device is that the S/D regions are filled with SiGe alloy which imparts compressive stress in the channel under the gate poly. The length of active area adjacent to the device channel is equal to the poly to poly distance (L_{pp}) between adjacent poly gates, therefore we will use the term L_{pp} to denote the dimension of the active area in the rest of the paper. The work in [13] and [14] has demonstrated that the stress in the device channel can be modulated by changing the active area dimension. In particular, [13] presented the transconductance improvement for isolated PMOS transistors as well as simulation results for densely packed transistors. Their results show significant improvement in transconductance and channel stress as a function of L_{pp}.

The hole mobility of PMOS devices has a near linear dependence on the uniaxial stress in the channel [20] and [21]. Using this, we transformed the increase in the stress values to the increase in mobility. SPICE simulations were then performed with the modified mobility values to obtain the delay of an inverter as a function of the channel stress.

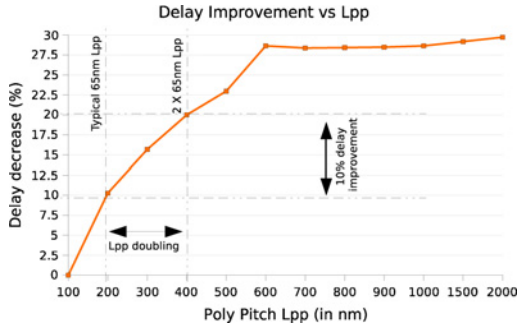

Fig. 2. Delay improvement of PMOS device as a function of active area dimension (L_{pp}). Delay improves by 10% when L_{pp} is doubled from its nominal (for 45 nm DRC) value of 100 nm to 200 nm.

The nominal value of L_{pp} for the 45 nm technology transistor was taken as 100 nm. Fig. 2, shows the improvement in the delay of the PMOS device as compared to a device without any increase in L_{pp}. We refer readers to the cited work for process related details of the experiments carried out. The near-linear dependence of the performance improvement on L_{pp} is evident from Fig. 2 which saturates at L_{pp} exceeding 600 nm. The design rules for 65 nm, 45 nm, and 32 nm gate lengths require the contacted L_{pp} to be approximately 150 nm, 100 nm, and 70 nm, respectively [1]. Even if we allow tripling of these L_{pp} dimensions for performance improvement, it is still below the saturation limit of 600 nm. We performed a linear fit (with $R^2 = 0.96$) of the performance improvement vs. L_{pp} increase curve in the range of interest of 100 nm to 200 nm and 200 nm to 400 nm and we observed that the doubling the active area size roughly corresponds to 10% better PMOS performance. We would like to point out that using S/D SiGe only increases the mobility of PMOS devices which improves the rise time of the cell without impacting its fall time. Since the delay of a cell is the average of fall and rise time, whenever we report reduction in the delay of a cell, we do it after scaling by a factor of 0.5 to compensate for only PMOS's improvement. This method is reasonable for most cases since in a timing path, nearly half of the cells are undergoing 0 => 1 transition while the rest are switching the other way. For a rare design whose critical paths have mostly all rising transition or all falling transition cells, the timing improvement should be obtained by performing static timing analysis.

Consider the top view of a transistor before and after L_{pp} resizing in Fig. 3 such that the L_{pp} after resizing is twice its original value. Each of the vertical tall (red) bars are poly gates, and the horizontal (yellow) tile is the active area. On increasing the L_{pp}, the delay through the gates can be reduced. Previously we saw that doubling the L_{pp} of a transistor can decrease its delay by 10%. Since in our methodology we stretch standard cells, in this section we will derive the effective active area increase as a function of increase in standard cell width (instead of increase in L_{pp} distance). It is visually evident that the increase in standard cell width is not 2X, but less than 2X when the L_{pp} is increased to 2X of its original size. Conversely, doubling the width of a standard cell effectively increases the L_{pp} dimension by more than 2X.

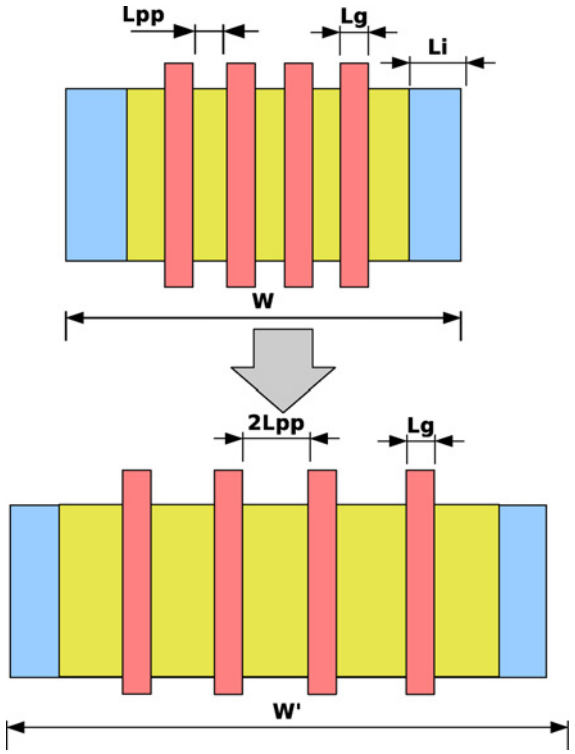

Fig. 3. Scenario of doubling of L_{pp} of a cell and its effect on cell width. Notations: gate length (L_g), active area (L_{pp}), insulation oxide thickness (L_i), W and W' are widths before and after active area doubling.

Let us assume that the gate length is L_g and the oxide isolation thickness on each side of the active area is L_i. Note that the gate length or the insulation oxide thickness does not increase while sizing up the active area. The total width of the standard cell before stretching is W. We considered the case where there are n poly gates (i.e., $n = 2$ for 2-input NAND/NOR, $n = 3$ for 3-input NAND/NOR, and so on.) in the cell. The original width of the standard cell W in terms of constituent dimensions can be computed as

$$W = (n+1)L_{pp} + nL_g + 2L_i. \tag{1}$$

After increasing the width of the cell K times, if the L_{pp} increases to L'_{pp}, then the following equation holds:

$$KW = (n+1)L'_{pp} + nL_g + 2L_i. \tag{2}$$

Taking the ratio of the above, we can derive that

$$L'_{pp} = KL_{pp} + \frac{(K-1)\left(nL_g + 2L_i\right)}{(n+1)}. \tag{3}$$

The above equation shows that increasing the width of a cell by K times, L_{pp} increases by more than K times. For example, in a typical case where insulation (i.e., L_i) and total gate length (i.e., $n \times L_g$) are 20% and 30% of W, respectively, we can achieve 3X increase in L_{pp} by doubling the width of the standard cell. For each standard cell in our library, we stored the mapping between the increase in standard cell's

width and the corresponding increase in its L_{pp} dimension which in turn was used to compute the delay decrease. Using (1), (2), and (3), if the delay of the cell when its width is increased to K times is $D(KW)$, then it can be represented in terms of its nominal delay $D(W)$ at original width of W as

$$D(KW) = D(W)\left(\frac{1-\alpha(K-1)W}{(n+1)L_{pp}}\right) \tag{4}$$

where the parameter α is the decrease in delay by doubling the L_{pp} which has been found to be 10% (i.e., 0.1). Note that the above expression only depends on the layout, the original width and the original L_{pp} of the device and thus is readily known once the cell library analysis is complete. In the above discussion, we have assumed that all PMOS devices inside a cell benefit equally due to the active area sizing. In practice, the devices on the boundary of the cell improve more than the device in the center of the cell because the nominal value of stress near the center of the cell is already higher than that at the cell boundary. For higher accuracy, the above analysis can be done for each input pin of a cell rather than for the whole cell.

IV. Timing Closure by Cell Stretching

In the next three subsections, we present our layout level active area stretching based timing closure scheme. First we discuss the constraints on L_{pp} stretching which we enforce to bring *predictability* to the achieved timing improvement. The formulation of LP is tackled next. Finally, we discuss the legalization algorithm to efficiently remove overlaps caused due to L_{pp} stretching.

A. Stretching Constraints for Predictability

Increasing the width of a critical cell[1] introduces overlap with non-critical cell adjacent to it and needs to be legalized. If this overlap is not properly controlled, the legalization step would cause substantial change in the location of non-critical cells. This increases the chances of turning a non-critical path into a critical path and can lead to several iterations of timing improvement without any guarantee of convergence. To control this effect, we propose the following constraints which our optimization flow must adhere to, while deciding about the extent of the critical cell's stretching. Due to the stretching of critical cells:

1) no critical cell can move (thus, can only stretch);
2) no non-critical cell can jump over a critical cell;
3) no cell (critical or not) should leave its row;
4) a critical cell can stretch only until a particular limit.

The above rules are graphically represented in terms of valid and invalid movement in Fig. 4. The original layout for the first three cases is given in Fig. 4(a).

The first constraint enforces that the interconnect delay between any pair of critical cells remains unchanged after expansion. In Fig. 4(b), the coordinates of cell A and C have changed, but that of critical cells B and D have not (even though these cells have stretched). Therefore, if there is an interconnect between cells B and D, the delay on that interconnect would not change.

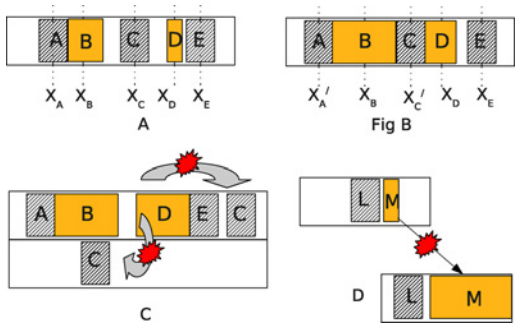

Fig. 4. Graphical representation of the rules which determine the stretching of critical cells. Long horizontal slabs are circuit rows. Cells in bold orange color are critical cells. Non-critical cells are in hatched slanted line. Position X_i represents the center of mass of cell i. See adjoining text below for a description of the four figures.

The second constraint prevents "wide" movement of a non-critical cell which circumvents the potential problem of large increase in interconnect delay of nets incident on non-critical cell. The intra-row movement in Fig. 4(c) represents this invalid move since the non-critical cell C needs to jump over critical cell D to allow D to expand.

The third constraint is like the second constraint with an added advantage that the modifications done in a row cannot induce detrimental changes to neighboring rows. The inter-row movement in Fig. 4(c) shows an example of invalid move since the non-critical cell C need to moved into another row. This constraint also removes any chain effect of overlaps in one row causing further overlaps in other rows.

The fourth constraint is a trade-off between the extent of layout modifications and the cycle time improvement in light of the saturation of improvement for large L_{pp} values (see Section III). The stretching depicted in Fig. 4(d) is invalid if the cell M is allowed to stretch up to 2X original size. In our experiments we enforce that no cell can be stretched to more than twice its original size.

B. LP Formulation for Timing Optimization

Consider a timing failing path p in the circuit. This path (as any other path) consists of an alternating sequence of standard cells and interconnects. Since the path is failing, all the cells will be critical (and thus, under purview of being stretched). Let the set $C_i = c_i^0, c_i^1 \,..\, c_i^m$ be the cells in the path i. Since the critical cells never move from their original location (as per Rule 1 above), the interconnect delays between these cells can be summed up and taken as fixed before and after stretching. Let d_i^I represent the total interconnect delay for path i. Further let the delay of cell c be D_c (pin-to-pin delay). Thus, the total delay of the path i (= $DELAY_i$) can be written as

$$DELAY_i = d_i^I + \sum_{j \in C_i} D_j. \tag{5}$$

[1]We define a cell lying on a failing path as a *critical* cell here onward. A cell which is not critical is referred to as non-critical cell.

Recall that the delay of cell c can be written according to (4). Let the width of the standard cell c with n_c poly gates be increased by ΔW_c. The path delay consisting of such cells can thus be expressed as

$$DELAY_i = d_i^I + \sum_{j \in C_i} \left(D_j(W) \times \left(1 - \frac{0.1 \times \Delta W}{(n_c + 1) L_{pp}^c} \right) \right) \quad (6)$$

where $D_j(W)$ is the original (i.e., without any increase in L_{pp}) of cell j. Note that the expression $(n_c + 1) \times L_{pp}^c$ is constant for a given type of cell and can be pre-computed and stored in a look-up table beforehand.

1) *Achieving Highest Performance:* If the optimization target is to achieve the highest possible performance while satisfying all the active area stretching constraints of Section IV-A, a LP can be formulated as follows. Let P_{crit} denote all failing paths, the white space available between two consecutive critical cells a and b in a row excluding the space used up by non-critical cells between them be denoted by WS_{ab} and each such consecutive pair of critical cells be part of the set $PAIRS$. The set of all critical cells is denoted as $CRIT$. The LP is

$$\begin{aligned} \max \quad & M \\ \text{s.t.} \quad & DELAY_i + M \leq 0 \quad (i \in P_{crit}) \\ & \Delta W_a + \Delta W_b \leq WS_{ab} \, (ab \in PAIRS) \\ & \Delta W_x \leq W_x \quad (x \in CRIT). \end{aligned}$$

The dummy variable, M, when maximized such that the first set of constraints are satisfied minimizes the delay of the longest path in the circuit. The second set of equations force a solution in which there is always enough space between two critical cells so that cells lying between them can be locally moved without overlap, thus never violating Rules 2 and 3 of Section IV-A. Third set of equation is to prevent any cell to become more than twice its original size: in adherence to Rule 4. The objective function, when maximized, is equivalent to choosing the right values of amount of expansion for each cell (basically ΔW_x for each cell x that is critical) so that the circuit is the fastest possible.

2) *Achieving Target Performance:* If the optimization target is to meet a given timing requirement with least possible active area increase, the LP can be formulated as follows. let the target path delay to be achieved be D_{target}, which is set by the required frequency of operation demand. The LP is

$$\begin{aligned} \min \quad & \textstyle\sum_{ab} (\Delta W_a + \Delta W_b) \quad (ab \in PAIRS) \\ \text{s.t.} \quad & DELAY_i \leq D_{target} \, (i \in P_{crit}) \\ & \Delta W_a + \Delta W_b \leq WS_{ab} \quad (ab \in PAIRS) \\ & \Delta W_x \leq W_x \quad (x \in CRIT). \end{aligned}$$

This formulation is different from that for achieving the highest performance because the cost function now tries to achieve the targeted performance with minimum increase in total active area. This helps to minimize the power penalty of active area sizing or to reduce the perturbation in the layout. In case a particular part of the layout is very sensitive and should not be disturbed, the ΔW's of the cells in that region can be set to zero.

Algorithm 1 LegalizeDesign

```
for all critical cell C in the design do
    RemoveOverlapOn ( right ) edge of ( C )
    RemoveOverlapOn ( left ) edge of ( C )
end for
```

Algorithm 2 RemoveOverlapOn [direction DIR, Cell C]

```
1: Identify neighboring cell N on $DIR of $C
2: Let $WS_CN be pre-stretching white space b/w $C and $N
3: Let ΔW_C be the increase in width of $C
4: Overlap $OL = ΔW_C/2 - $WS_CN
5: while $OL > 0 do
6:     Shift $N by $OL to $DIR
7:     $C ⇐ $N
8:     $N ⇐ neighboring cell on $DIR of $N
9:     Recompute $WS_CN using pre-stretching location
10:    $OL = $OL - $WS_CN
11: end while
```

Typically, a design requires some white space which subsequently is used by spare cells, decaps and filler cells. To accommodate these cells, an extra constraint can be added to the above two LP formulations to limit the maximum amount of active area growth to a particular fraction of the available white space.

3) *LP Formulation Complexity Analysis:* The total number of path delay constraints (first set) in both the above LP formulations is linear in the number of failing paths. The set $PAIRS$ only has pairs of *critical* cells which are in the *same row* of the layout such that there is no other critical cell between them (for example cells B and D in Fig. 4). Therefore, the number of pairs in this set is linear in the number of critical cells. Similarly number of last set of constraints is also linear in the number of critical cells. Therefore, the overall number of constraints in the LP formulation is $O(C_{crit} + P_{fail})$ where C_{crit} is the number of cells which are critical and P_{fail} is the number of paths that are violating timing requirement. The output of the LP is the amount of increase in the width of various critical cells.

C. Minimum Perturbation Legalization

After the LP has been solved and the critical cells enlarged, there may be overlaps between the expanded critical cells and their neighboring non-critical cells. This overlap needs to be removed through legalization. In view of the general philosophy of perturbing the minimum amount of interconnects and cells, we propose Algorithm 1 which remove overlaps while shifting the least number of cells by minimum displacement to get a legalized placement.

The above algorithm makes two calls to RemoveOverlapOn (described in Algorithm 2) for each timing critical cell in the layout: the first time to remove overlap from the left edge of the critical cell and the second time to do it from the right edge. Next we describe the algorithm RemoveOverlapOn.

Consider the case in which Algorithm 2 is invoked to remove overlaps from the left edge of a critical cell. This

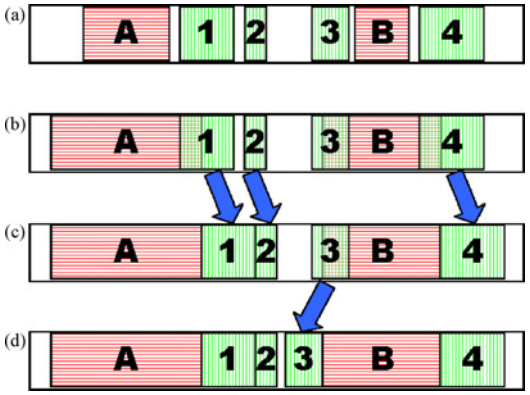

Fig. 5. Sequence of four images depicting a simple standard cell row with two critical and four non-critical cells undergoing overlap removal. Critical cells are shown in red with horizontal lines. Non-critical cells are shown in green with vertical lines. (a) Original layout. (b) After critical cells have expanded. (c) After removing overlap on right-hand side of each critical cell. (d) After overlap removal on left-hand side of each critical cell.

means that the argument *DIR* is set as *left*[2] The explanation of the complimentary case for the *right* direction is similar in nature. First, we identify the cell lying immediately on the left of the critical cell. In case the critical cell is the first cell of the circuit row, the beginning of the row acts as a dummy neighboring cell. Then, based on the increase in the width of the critical cell (Line 3) and the existing white space between the critical cell and the neighboring cell (Line 2), we compute the current value of overlap (Line 4). This value is the distance the neighboring cell must move in order to become non-overlapping with the critical cell. Next, we iterate until the value of overlap is non-negative. In each iteration, we shift the neighbor cell by the amount of overlap. We then assign the neighbor cell as the new critical cell whereas the cell next to the neighbor cell becomes the new neighbor cell (Line 8). Further, we update the value of overlap by subtracting the white space which originally existed between the new critical and the new neighbor cell. The loop terminates when the overlap is non-positive meaning that the extra width has been successfully consumed by the white space available. Fig. 5 depicts the sequence of our legalization flow: the original layout and the layout after critical cell expansion are shown in (a) and (b), respectively. In Fig. 5(c) and (d), the overlap on right edge and left edge of each critical cells are resolved, respectively.

V. Experimental Flow and Setup

Fig. 6 depicts our overall flow. Given a benchmark, we compile the RTL on to a standard cell library to technology dependent netlist using Synopsys Design Compiler. The next step is to perform place and route of this design followed by parasitic extraction performed using Cadence SoC Encounter [22] (version 6.2) tool. At this stage, we generate an estimate of the active and leakage power dissipation of the design using the same tool (edge marked *Original Power* in the figure) based on the probabilistic switching activities at the circuit input. Parasitics aware timing analysis is then

[2]To better understand Algorithm 2, read it while replacing every occurrence of "DIR" with "left."

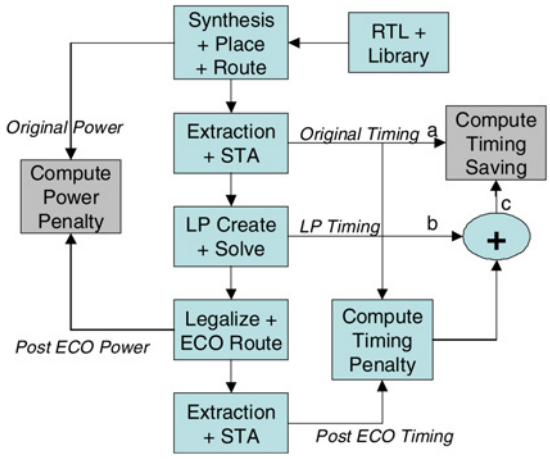

Fig. 6. Flow used in our experiments.

performed using Synopsys Primetime tool to identify the paths which need to be fixed as well the design cycle time (edge marked *Original Timing* in the figure). In our flow, we considered all paths whose delay is more than 80% of the maximum delay as failing.

A C++ program was developed to analyze the delay through the cells on the failing path along with their geometry and location in the chip layout to generate the set of constraints outlined in Section IV-A. The maximum allowed post-stretching width of each cell is constrained to twice of its original width. Based on these constraints, the two LP formulation for minimum cycle time and minimum perturbation timing closure are generated as outlined in Section IV-B. These two LP are solved using the open-source *GNU Linear Programming Toolkit* [23] to compute the new width of cells on critical cells as well as the achieved cycle time of the design (edge marked *LP Timing* in the figure). Another C++ program was used to implement the active area sizing solution generated by the LP using the legalization procedure presented in Algorithm 1. In addition, this program un-routes all the interconnects which are incident on the cells that are moved due to the legalization process. ECO routing of the resultant design is then performed using the Encounter tool to complete the routing of these nets followed by parasitic extraction of the changed layout.

To estimate the power dissipation of the circuit after active area sizing, we first use Cadence Encounter [22] tool to report individual gate's nominal dynamic and leakage power. These power numbers are post-processed as follows. The dynamic power of the cells which undergo active area sizing are scaled by the increase in their active area to account for larger diffusion capacitance that needs to be charged/discharged while switching. For leakage power estimate, we generated a look-up table which maps the increase in mobility of a PMOS device to the increase in its leakage power through SPICE simulations. The leakage power numbers of the cells which undergo active area sizing are then scaled by the factor corresponding to its mobility increase from the look-up table generated. The resulting power dissipation is depicted as the edge marked *Post ECO Power* in the figure which is compared with the original power dissipation to compute the increase in power dissipation.

After the ECO routing is complete, we perform a parasitics aware timing analysis using Primetime to report the new timing of the design (edge marked *Post ECO Timing* in the figure). The difference between the timing achieved at this step and that of the original layout represents the penalty paid due to perturbation of the original solution generated by placer/router. This timing penalty is added to the timing reported within the C++ timing engine (marked *LP Timing*) to compute the total effective timing of the design and thus the timing saving achieved by our method is computed.

Example 1: Imagine a circuit which has been placed and routed. Parasitic aware timing analysis yields the maximum delay through the circuit as 10 ns. We run the critical paths (say which have a delay more than 9 ns) through the LP formulation for active area stretching. The solution of LP reports the improved delay through the circuit as 9.5 ns. We perform legalization and ECO routing on the solution provided by the LP and rerun parasitic aware timing analysis which reports the delay as 10.1 ns. Comparing 10 ns and 10.1 ns, we compute the timing penalty due to legalization and ECO routing as 0.1 ns. This penalty is added to 9.5 ns to get the corrected delay of optimized circuit as 9.6 ns which when compared to original delay of 10 ns boils down to 4% reduction in the delay of the circuit.

All experiments were performed on 64-bit 4-CPU 8-GB RAM machines running Linux operating system. Our code was implemented in the C++ language. We performed two sets of experiments to demonstrate the efficacy of our optimization technique. The first set of experiment is targeted to explore the dependency of our optimization method on the choice of benchmark circuits. The second set of experiment deals with the optimization of one particular benchmark for a variety of different design parameters. The method of estimating routing effort of a design and the setup of these two set of experiments are explained next.

A. Estimating Routing Effort

To faithfully capture the detrimental impact of our optimization flow on the timing due to increased interconnect length, it is important to consider the routing *effort* of the design. Routing *effort* can be defined as the difficulty a router faces to successfully route all the nets and is a function of average number of fanouts of cells and degree of connectivity among the cells. A design with high routing effort is very susceptible to movement of cells during the legalization step since the resulting ECO routing may introduce significant detour to re-route it, possibly generating a high delay path. Traditionally, routing effort is measured by routing congestion which is computed on each global routing grid cell perimeter as the ratio of interconnects crossing the grid perimeter to the number of tracks available. However, this per-grid routing congestion is not suitable for our purpose since we want to explore the impact of higher overall routing effort of the design. To capture this, in our experiments we used the routing layers available in the router (during the initial routing phase and ECO routing phase) as a proxy for modulating the holistic routing effort of the design. The reduction in available routing layers can be considered as reducing track capacity per grid cell summed over all routable layers thus measuring the resources available to the router.

TABLE I
CHARACTERISTICS OF THE BENCHMARK CIRCUITS USED IN THE FIRST SET OF EXPERIMENTS

Bench Ckt	Num Cells	Num Nets	Num IO	Avg Fanout	Row Util	Route Lyrs
des	88 566	90 660	298	2.10	71.1%	7
vga	4128	4761	214	2.26	70.4%	7
dlx	13 471	14 678	170	2.47	72.0%	7
ethernet	49 427	50 460	210	2.36	69.3%	7
i8051	9752	10 432	100	2.63	70.7%	7
JpgComp	12 294	13 324	77	2.38	69.4%	7

The columns denote the number of cells, nets, IO pins, average fanout of each cell, layout aspect ratio, row utilization, and number of layers allowed for routing. Aspect ratio of die for layout was set as 1.00.

B. Experiment Set 1

In the first set of experiments, our aim is to quantify the optimization potential of our methodology on a variety of different benchmark circuits. Since these benchmarks have different characteristics such as cell/net count, path delay distribution, number of inputs/outputs, and interconnect structure, this experiment proves the generic nature of our optimization method. All the properties of the optimization flow were kept similar for all the benchmark circuits in order to avoid any bias. These properties include layout row utilization, available routing layers, aspect ratio of the die, and buffers/repeaters available to the physical synthesis tool. Table I shows the primary characteristics of the different benchmarks we used for this set of experiments. The geometrical aspect ratio of all layouts was fixed to be 1.00 and the circuit row utilization was set to be 70%. The technology independent RTL of these benchmarks are obtained from the open-source *Opencores* [24] design repository. All designs were technology mapped to the open-source 45 nm non-linear delay model library from *Nangate* [25] and were routed using seven routing layers which is the maximum available layers.

C. Experiment Set 2

In the second set of experiments, we narrowed our focus to a single benchmark circuit and performed an in-depth analysis of how the timing results achieved differ due to various typical design choices. The parameters that were varied for this benchmark are standard cell library (high or low V_{th} variants), routing effort, and core row utilization. The choice of threshold voltage determines the path delay distribution. We used two variants of a commercial 90 nm standard cell library for this set of experiments: high V_t (350 mV) and low V_t (200 mV). The nominal supply voltage of the library was 1V. The choice of core row utilization defines the amount of white space available for cell stretching. We performed circuit layout with row utilization values ranging from 20% to 85% in steps of 5%. The high routing effort version of the layout was allowed only 4 routing metal layers to complete all the routing whereas the low routing effort version had the freedom to route with 7 routing metal layers. Table II shows the primary characteristics of the benchmark we used for this set of experiments.

TABLE II
CHARACTERISTICS OF THE BENCHMARK CIRCUIT USED IN THE SECOND SET OF EXPERIMENTS

Bench Ckt	Num IO	Num Cells	Num Nets	Avg FO	Num Cells	Num Nets	Avg FO
		Low V_{th}			High V_{th}		
wims	112	12 238	22 492	3.67	11 454	21 209	3.70

The columns denote the number of IO pins and the tuple {number of cells, nets and average fanout of each cell} for high and low V_{th} library mapped variants.

VI. EXPERIMENTAL RESULTS

A. Results for Experiment Set 1

We report the results for the two formulations: achieving the highest performance where the objective is to make the circuit as fast as possible and for achieving target performance where the objective is to obtain a particular performance with minimum stretching (corresponding to minimum dynamic power increase).

Table III shows the results achieved by our flow for various benchmarks where the objective is to make the circuit as fast as possible. Columns *a*, *b*, and *c* tabulate the cycle time of original design, LP reported cycle time before legalization/ECO routing (legalization here onward), and final timing considering the correction due to perturbation arising from legalization. In Fig. 6, these correspond to edges marked *a*, *b*, and *c*, respectively. The timing improvement achieved before legalization is in Column *d* and their corresponding values after legalization is in Column *e*. Column *f* shows the difference between these two timing improvement values. Columns *g*, *h*, *i*, *j*, and *k* show the number of critical cells, number of cells stretched, number of cells moved during legalization, increase in dynamic power consumption, and increase in leakage power consumption of the chip, respectively. Column *l* shows the number of nets that need to be ECO routed. Finally, the combined CPU time for the LP solver, legalization and ECO routing is reported in the last column.

From Table III we observe that across different benchmarks, our technique can improve the cycle time by as much as 6.77% and as little as 4.5%. Averaging across different benchmarks, we obtain 5.74% cycle time improvement which corresponds to more than 6% increase in the design's frequency of operation, which is remarkable at post placement stage without requiring any change in the netlist structure.

We note that the difference between the delay improvement predicted by the solution of LP (i.e., before legalization) and the actual saving after legalization is very minute with an average value of 0.06%. In fact, in two of the six benchmark, there is no difference between the two timing improvements. This is a key observation because the very small difference means that the designer can get immediate feedback of the potential timing improvement right after solving the LP without the need of legalization or ECO routing. In case the timing achieved is not sufficient, the maximum size to which a cell is allowed to stretch can be increased and our optimization flow can be re-run. On the contrary, if the timing achieved is significantly high, the criteria of failed path definition can be changed to capture additional paths

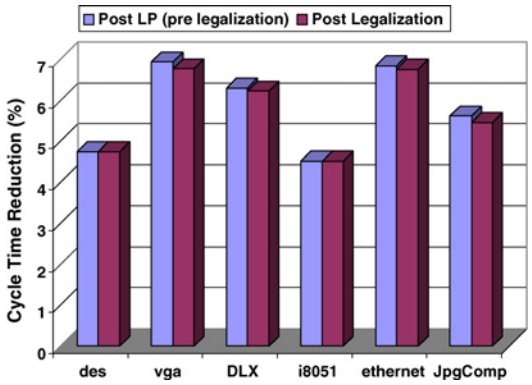

Fig. 7. Cycle time reduction achieved by our flow just after LP solving and after (legalization+ECO routing). Up to 6.5% cycle time reduction is observed. Minute difference between the saving numbers allows early prediction of the optimization potential.

and our flow can be run again. In this way, our flow is very suitable as an efficient *what-if* analysis tool. The very small difference due to legalization and ECO routing is enabled due to the rigid stretching constraints we propose in Section IV-A. Without these constraints, the LP reported timing improvement would be higher but there would be much more uncertainty regarding the final timing achieved due to larger perturbations during legalization and ECO routing. Fig. 7 shows the savings achieved before and after legalization for different benchmarks (also shown in Table III).

From Table III, we also observe that very few of the critical cells undergo active area stretching. On an average, only 0.42% of n their width which increase the dynamic power of the design by 0.58%. The average increase in leakage power of the design is less than 5%. Notice that this is 10X more than the impact on dynamic power because mobility enhancement has a much stronger impact on leakage than dynamic power or delay. The number of cells that move during legalization because of stretching of critical cells is only 0.37%. Due to the movement of cells during legalization, on an average 2.4% of the nets need to be ECO routed. Except the benchmark `JpgComp`, the LP solving, legalization and ECO routing for all other benchmarks finished within 15 minutes. For `JpgComp`, the router spends a longer time due to a larger number of nets (5.46%) that need to be re-routed.

Next, we present the results for the formulation which achieves a given timing target with minimum increase in dynamic power. Note that the value given in Table III is the upper bound on the improvement of timing that can be obtained which corresponds to letting all the cells increase their width as long as the timing is improved while satisfying stretching constraints. We incrementally tightened the targeted timing (T_{tgt}) of the design in steps varying from 1% to 5% and observed very minute difference between the timing reported before legalization (i.e., after LP solving) and after legalization (<0.04%). Therefore only the final delay numbers after legalization and ECO routing are reported. Table IV tabulates the results obtained: the second and the third column show the original timing of the design and the number of critical cells in the failing paths, respectively. Each set of three columns thereafter *a*, *b*, and *c* reports the number of cells stretched,

TABLE III
RESULTS OF OUR FLOW TO ACHIEVE FASTEST POSSIBLE CIRCUIT

Design	Orig Timing *a*	Post LP Timing *b*	Final Timing *c*	Post LP Improv *d*	Final Improv *e*	Δ Improv *d* − *e*	# Cells Critical *g*	# Cells Stretch *h*	# Cells Move *i*	ΔPower Dynamic *j*	ΔPower Leak *k*	ECO Nets *l*	CPU Time (s) *m*
des	6.153	5.569	5.570	4.74%	4.74%	0%	13 092	48	5	0.39%	2.53%	0.09%	781
vga	2.081	1.792	1.799	6.94%	6.77%	0.17%	116	34	23	0.61%	5.24%	1.91%	45
dlx	3.343	2.922	2.927	6.29%	6.22%	0.07%	548	172	104	0.94%	7.33%	3.41%	238
i8051	11.265	10.248	10.249	4.51%	4.51%	0%	131	62	95	0.89%	4.97%	2.79%	902
ethernet	28.343	24.463	24.507	6.84%	6.76%	0.08%	2406	299	92	0.34%	2.88&	0.36%	325
JpgComp	7.238	6.425	6.447	5.61%	5.46%	0.15%	273	80	296	0.70%	4.86%	5.46%	2438
Avg.				5.82%	5.74%	0.06%		0.42%	0.37%	0.58%	4.63%	2.36%	12

See accompanying text for columns' description.

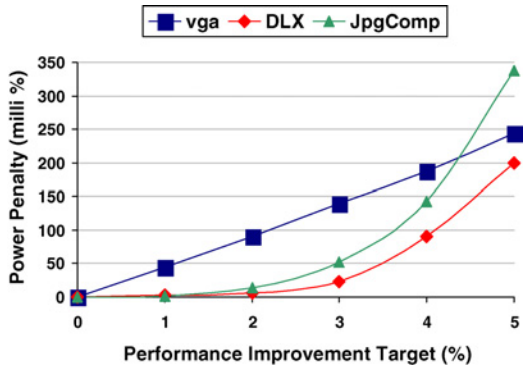

Fig. 8. Dynamic power penalty (due to larger diffusion capacitance of increase active area) as a function of performance improvement for some of the benchmark circuits. *y*-axis is in milli% = 0.001%. Values of other benchmarks can be read off Table IV.

the number of cells that move during legalization, and the increase in the dynamic power of the design, respectively, while achieving the corresponding T_{tgt}. The layout parameters such as aspect ratio, row utilization and the number of routing layers are the same as that for Table III.

The first observation we make from Table IV is that *very* few cells need to stretch and move during legalization as compared to the case when the objective is to achieve maximum performance (in Table III). This is due to the explicit objective function which minimizes the stretching for achieving a target performance. For example, in benchmark des, only 4 cells need to stretch in order to achieve 96% target delay (meaning 4% performance improvement) whereas to achieve a performance improvement of 4.74%, 48 cells need to stretch, a 12X increase. Similarly, for benchmark JpgComp, the power penalty to achieve 5.46% delay improvement is 2X (0.70% from Table III vs. 0.338% from Table IV) as compared to achieving 5% delay improvement. Therefore, we conclude that one should use the formulation for achieving maximum performance when absolutely necessary as the resultant layout modification can be much more for small incremental performance improvement. For benchmarks des and i8051, it was impossible to achieve 5% reduction in cycle time as the maximum saving for these benchmarks were 4.74% and 4.51%, respectively. In Fig. 8, the dynamic power penalty of some of the benchmarks are plotted as a function of the different target performance. There is a smooth trade-off between the target performance and the power penalty.

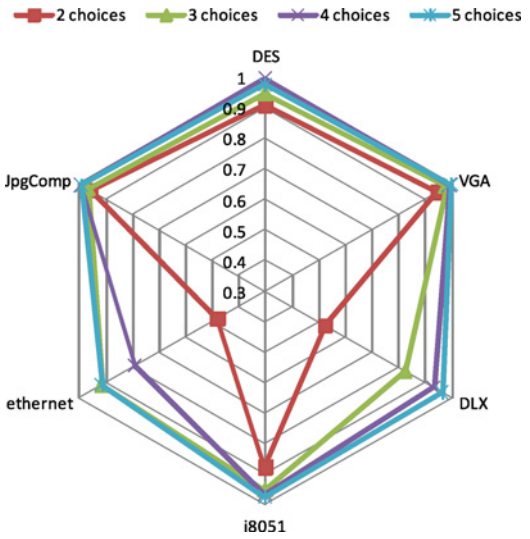

Fig. 9. Performance improvement achieved when active area sizes are restricted to 2, 3, 4, or 5 discrete variants normalized to that achieved assuming continuous active area sizes. In each experiment, all the available variants are equally spaced between 1X and 2X nominal active area sizes.

1) *Continuous vs. Discrete Active Area Sizing:* For standard cell based designs, the choice of active area available during layout is restricted to the variants available in the library. The performance improvement shown in this section has so far assumed that the active area of a cell can be sized in a continuous fashion. To understand how the availability of only a few discrete active area sizes of each cell in the library well affect the optimization potential, we repeated our experiment after restricting the choice of cells available. This effectively changes our formulation from linear programming to integer linear programming. Fig. 9 shows a radar plot of performance improvements achieved using discrete choice of active area sizes normalized w.r.t. the savings achievable by using continuous choice of active area sizes. Each radial line represents a benchmark. The data is shown for the cases when there are 2, 3, 4, or 5 variants of each standard cell available in the library. For the case with n variants, the available expanded active area sizes are assumed to be equally spaced between 1X (i.e., the nominal size) to 2X (i.e., twice the nominal size). For example, for the case when 3 variants are present, the active area size of these variants are 1.0X, 1.5X, and 2.0X times the nominal size. From Fig. 9, we observe that some benchmarks

TABLE IV
RESULTS FOR ACHIEVING TARGET TIMING (T_{tgt}) WITH LEAST POWER INCREASE

Design	Orig Timing	# Crit Cells	T_{tgt} = 99% Orig.			T_{tgt} = 98% Orig.			T_{tgt} = 97% Orig.			T_{tgt} = 96% Orig.			T_{tgt} = 95% Orig.		
			a	*b*	*c*	*a*	*b*	*c*	*a*	*b*	*c*	*a*	*b*	*c*	*a*	*b*	*c*
des	6.153	13 092	1	0	0.3m%	1	1	0.6m%	2	2	1m%	4	3	1.3m%	NA	NA	NA
vga	2.081	116	3	3	45m%	4	3	91m%	6	4	139m%	8	4	188m%	12	15	244m%
dlx	3.343	548	1	0	3m%	1	0	6m%	8	5	23m%	20	5	90m%	39	15	200m%
i8051	11.26	131	6	7	37m%	14	26	132m%	31	40	340m%	47	67	676m%	NA	NA	NA
ethernet	28.34	2406	2	0	0.2m%	4	0	0.8m%	6	0	2m%	7	0	3m%	7	0	5m%
JpgComp	7.238	273	1	6	2m%	4	19	14m%	11	27	52m%	27	54	143m%	56	151	338m%

T_{tgt} varied through 99%, 98%, 97%, 96%, and 95% of original timing. Columns *a*, *b*, and *c* show no. of cells stretched, no. of cells moved during legalization, and dynamic power increase (in milli percent = 0.001%) for each T_{tgt}. Entries marked NA mean the corresponding T_{tgt} was not achievable.

such as Jpgcomp and vga are not very sensitive to how many variants of each cell exists in the library and even if there are only 2 active area sizes available (1X and 2X), the savings achieved is almost as good as that achievable with continuous sizing. On the other hand, the timing improvements achievable for benchmarks DLX and ethernet are very sensitive to the number of variants available in the library. In particular, for the benchmark ethernet, restricting to only two variants (i.e., 1X and 2X only) reduces the performance improvement to only 40% of what is achievable in presence of continuous sizes. As a rule, we observe that having more variants enhances the performance improvement. One exception to this rule is observed for benchmark des for which the performance improvement when using four choices of active areas (i.e., 1.0X, 1.33X, 1.67X, and 2.0X) is better than having five choices (i.e., 1.0X, 1.25X, 1.5X, 1.75X, and 2.0X) of active areas. We believe this is due to different discretization of the solution space given that the sizes available for four choices are not strictly subset of those available for five choices. Overall, we observe that having three variants (i.e., 1X, 1.5X, and 2X) is a good trade off between performance improvement and cell library sizes.

B. Results for Experiment Set 2

The above results are very encouraging as the timing improvement achieved is consistent among all different benchmarks. Next, we focus on one benchmark and perform in-depth analysis of the impact of the routing layers, library threshold voltage, and row utilization as outlined in Section V-C. For all these runs, the objective function is to make the circuit as fast as possible. Table V tabulates some of the representative observation points. The first two columns show the row utilization and number of metal layers available to the router. All delay values are presented in adjacent pair of columns for design mapped using low and high V_{th} library variants. The cycle time of original layout after parasitic extraction appears under heading *Orig Timing*. Columns under *Post LP Timing* show the cycle time reported after LP solution when the critical cells are expanded, but before any legalization and ECO routing is done. Columns under *Final Timing* show the final cycle time after legalization and ECO routing. Column *Improvement* tabulates the timing improvement thus achieved (by comparing *Orig Timing* and *Final Timing*). The set of columns under *Cells Legalized* show the number of cells moved due to legalization after the expansion of the critical cells.

The comparison of entries under Column *Post LP Timing* and *Final Timing* shows that the cycle time before and after the legalization and ECO routing stages are usually exactly the same which means that legalization and re-routing did not degrade a non-critical path into critical path. When these entries are not the same, they are very close to each other with a maximum difference of 0.01% in the row utilization range of 0.4–0.7. Similar to the results of experiment set 1, we note that *very* few cells ($\leq$0.7%) were moved during legalization. The total number of nets which need to be ECO routed was found to be under 0.4% of the total nets for all the different configurations. We observe in Table V that our technique is able to reduce the cycle time of the design by nearly 5.25% averaged over all the benchmarks.

We next discuss the impact of the various design choices such as V_{th} values, number of routing layers used, row utilization of a design on the achieved cycle time reduction using our optimization methodology. Fig. 10 shows the cycle time reduction achieved using our technique for our benchmark routed in 4 vs. 7 metal layers for different row utilization. Our hypothesis was that a design with high routing effort (i.e., with lesser number of metal layers available) would offer less timing improvement due to possible detours during ECO routing of nets ripped by legalizer. The results of the experiments show this to be true most of the times, even though the difference observed is small. We believe that the difference is tiny due to the minuscule number of nets that need to be re-routed owing to constraints in Section IV-A. Overall, an average of 5.1% cycle time reduction was achieved over row utilization ratio between 0.4 and 0.7.

Fig. 11 shows the variation of timing improvement for the low threshold design and high threshold design. In general, high threshold voltage designs are slightly more amenable to active sizing expansion based timing optimization method. The reason being that the timing improvement we achieved comes from the decrease in cell delay and high threshold voltage cells have higher ratio of cell delays to interconnect delays. Overall, an average of 5.2% cycle time reduction was achieved for various values of row utilization between 0.4 and 0.7.

The amount of white space directly impacts our technique, because in essence, our technique consumes white space to improve cycle time. Looking at Figs. 11 and 10, we can observe how the timing improvement varies as row utilization is changed for different V_{th} values and routing layers used. As expected, higher row utilization of the design (=lower

TABLE V
RESULTS FOR ACHIEVING FASTEST POSSIBLE CIRCUIT FOR BENCHMARK wims FOR DIFFERENT ROW UTILIZATION, THRESHOLD VOLTAGES (LVT: LOW V_{th}, HVT: HIGH V_{th}), AND ROUTING LAYERS

Design	Row	Rout	Orig. Timing (ns)		Post LP Timing (ns)		Final Timing (ns)		Improvement (%)		No. Cells Legalized	
	Util	Layr	LVT	HVT	LVT	HVT	LVT	HVT	LVT	HVT	LVT	HVT
wims	0.20	4	9.879	40.171	8.727	35.553	8.727	35.553	5.83%	5.75%	78	88
wims	0.20	7	9.883	40.180	8.731	35.271	8.731	35.271	5.83%	6.11%	80	89
wims	0.30	4	9.389	32.889	8.352	29.260	8.352	29.260	5.50%	5.53%	83	79
wims	0.30	7	9.379	32.811	8.349	29.182	8.349	29.182	5.52%	5.53%	79	80
wims	0.40	4	9.257	31.923	8.089	28.261	8.088	28.264	6.31%	5.73%%	79	74
wims	0.40	7	9.241	31.918	8.074	28.257	8.074	28.260	6.32%	5.73%	80	78
wims	0.50	4	9.060	32.110	8.068	28.322	8.068	28.322	5.47%	5.90%	76	69
wims	0.50	7	9.062	31.423	8.071	28.423	8.071	28.423	5.47%	5.68%	84	69
wims	0.60	4	8.865	31.242	7.969	26.452	7.969	26.455	5.05%	4.95%	67	78
wims	0.60	7	8.862	29.066	7.969	25.954	7.969	25.958	5.05%	5.35%	77	73
wims	0.70	4	8.354	31.734	7.584	29.019	7.580	29.019	4.65%	4.28%	80	69
wims	0.70	7	8.343	31.703	7.567	28.981	7.567	28.981	4.63%	4.29%	78	78
wims	0.80	4	8.161	30.823	7.558	27.846	7.558	27.842	3.67%	4.12%	72	84
wims	0.80	7	8.106	29.672	7.511	26.582	7.573	26.582	3.70%	5.20%	81	76
Avg.									5.21%	5.29%	78	77

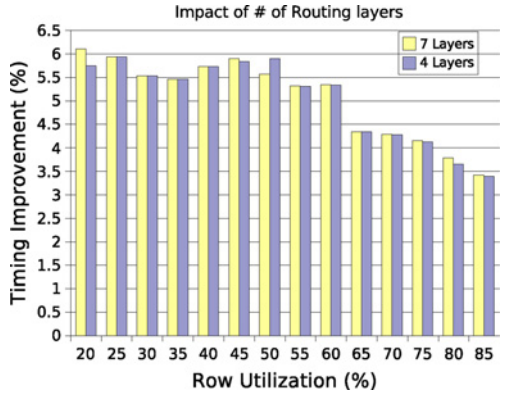

Fig. 10. Timing improvement vs. row utilization for routing using four and seven layers. Benchmark: wims high V_{th}.

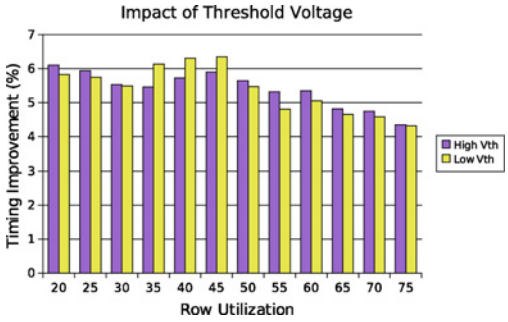

Fig. 11. Timing improvement vs. row utilization for low and high V_{th} cell library wims variants.

white space), leaves less room for our flow to improve cycle time. Overall, in the practical working range of row utilization of 0.4–0.7, our technique achieved an average cycle time reduction of 5.3%.

VII. CONCLUSION AND FUTURE WORK

In this paper, we have analyzed the impact of active area sizing on mobility improvement for S/D SiGe type devices. An active area stretching aware cell delay model was proposed based on the mobility improvement of the PMOS devices. For the first time, a methodology to exploit this phenomenon inside the conventional design flow is proposed incorporating the active area sizing based optimization during timing closure. A set of constraints was proposed whose adherence results in impeccable predictability of timing improvement after legalization and ECO routing. A legalization algorithm with minimum perturbation to existing cells was also proposed. On several benchmark circuit and on a wide range of design parameters (such as threshold voltages, row utilization, routing congestion), our technique achieves up to 6.3% cycle time reduction which is very attractive considering that the optimization is done at a very late stage in design cycle.

Future works would include analysis of other stressing mechanisms for optimizing a layout. Another line of research would explore concurrent usage of gate sizing and active area sizing to achieve timing closure.

REFERENCES

[1] ITRS, "International Technology Roadmap for Semiconductor," *Tech. Rep.*, 2007 [Online]. Available: http://www.itrs.net/Links/2007ITRS/Home2007.htm

[2] F. Andrieu, T. Ernst, F. Lime, F. Rochette, K. Romanjek, S. Barraud, C. Ravit, F. Boeuf, M. Jurczak, M. Casse, O. Weber, L. Brevard, G. Reimbold, G. Ghibaudo, and S. Deleonibus, "Experimental and comparative investigation of low and high field transport in substrate and process-induced strained nanoscaled MOSFETs," in *Proc. Symp. VLSI Technol. Dig. Tech. Papers*, Jun. 2005, pp. 176–177.

[3] H. Nii, T. Sanuki, *et al.*, "A 45nm high performance bulk logic platform technology (CMOS6) using ultra high NA(1.07) immersion lithography with hybrid dual-damascene structure and porous low-k BEOL," in *Proc. IEDM*, Dec. 2006, pp. 1–4.

[4] M. Wiatr, T. Feudel, A. Wei, A. Mowry, R. Boschke, P. Javorka, A. Gehring, T. Kammler, M. Lenski, K. Frohberg, R. Richter, M. Horstmann, and D. Greenlaw, "Review on process-induced strain techniques for advanced logic technologies," in *Proc. RTP*, Oct. 2007, pp. 19–29.

[5] P. Morin, "Mechanical stress in silicon based materials: Evolution upon annealing and impact on devices performances," in *Proc. 14th IEEE Int. Conf. Adv. Thermal Process. Semiconductors (RTP)*, Oct. 2006, pp. 93–102.

[6] K. Mistry, M. Armstrong, C. Auth, S. Cea, T. Coan, T. Ghani, T. Hoffmann, A. Murthy, J. Sandford, R. Shaheed, K. Zawadzki, K. Zhang,

S. Thompson, and M. Bohr, "Delaying forever: Uniaxial strained silicon transistors in a 90 nm CMOS technology," in *Proc. VLSI Technol. Dig. Tech. Papers*, Jun. 2004, pp. 50–51.

[7] M. D. Giles, M. Armstrong, C. Auth, S. M. Cea, T. Ghani, T. Hoffmann, R. Kotlyar, P. Matagne, K. Mistry, R. Nagisetty, B. Obradovic, R. Shaheed, L. Shifren, M. Stetter, S. Tyagi, X. Wang, C. Weber, and K. Zawadzki, "Understanding stress enhanced performance in Intel 90 nm CMOS technology," in *Proc. VLSI Technol. Dig. Tech. Papers*, Jun. 2004, pp. 118–119.

[8] L. Washington, F. Nouri, S. Thirupapuliyur, G. Eneman, P. Verheyen, V. Moroz, L. Smith, Xu Xiaopeng, M. Kawaguchi, T. Huang, K. Ahmed, M. Balseanu, Li Qun Xia, Meihua Shen, Yihwan Kim, R. Rooyackers, Kristin De Meyer and R. Schreutelkamp, "PMOSFET with 200% mobility enhancement induced by multiple stressors," *IEEE Electron Device Lett.*, vol. 27, no. 6, pp. 511–513, Jun. 2006.

[9] *Advanced Micro Devices* [Online]. Available: http://www.amd.com

[10] *Intel Corporation* [Online]. Available: http://www.intel.com

[11] *IBM* [Online]. Available: http://www.ibm.com

[12] T. Feudel and M. Horstmann, "Recent advances in stress and activation engineering for high-performance logic transistors," in *Proc. 16th IEEE Int. Conf. Adv. Thermal Process. Semiconductors (RTP)*, 2008, pp. 1–34.

[13] G. Eneman, P. Verheyen, R. Rooyackers, F. Nouri, L. Washington, R. Schreutelkamp, V. Moroz, L. Smith, An De Keersgieter, M. Jurczak, and Kristin De Meyer, "Scalability of the $Si_{1-x}Ge_x$ source/drain technology for the 45-nm technology node and beyond," *IEEE Trans. Electron Devices*, vol. 53, no. 7, pp. 1647–1656, Jul. 2006.

[14] R. Bianchi, G. Bouche, and O. Roux-dit Buisson, "Accurate modeling of trench isolation induced mechanical stress effects on MOSFET electrical performance," in *Proc. Dig. IEDM*, 2002, pp. 117–120.

[15] A. Chakraborty, S. Shi, and D. Pan, "Layout level timing optimization by leveraging active area dependent mobility of strained-silicon devices," in *Proc. DATE*, Mar. 2008, pp. 849–855.

[16] A. Kahng, P. Sharma, and R. Topaloglu, "Exploiting STI stress for performance," in *Proc. IEEE/ACM ICCAD*, Nov. 2007, pp. 83–90.

[17] V. Joshi, B. Cline, D. Sylvester, D. Blaauw, and K. Agarwal, "Stress aware layout optimization," in *Proc. ISPD*, 2008, pp. 168–174.

[18] B. T. Cline, V. Joshi, D. Sylvester, and D. Blaauw, "Steel: A technique for stress-enhanced standard cell library design," in *Proc. IEEE/ACM ICCAD*, 2008, pp. 691–697.

[19] V. Joshi, B. Cline, D. Sylvester, D. Blaauw, and K. Agarwal, "Leakage power reduction using stress-enhanced layouts," in *Proc. DAC*, 2008, pp. 912–917.

[20] L. Shifren, X. Wang, P. Matagne, B. Obradovic, C. Auth, S. Cea, T. Ghani, J. He, T. Hoffman, R. Kotlyar, Z. Ma, K. Mistry, R. Nagisetty, R. Shaheed, M. Stettler, C. Weber, and M. D. Giles, "Drive current enhancement in p-type metal–oxide–semiconductor field-effect transistors under shear uniaxial stress," *Appl. Phys. Lett.*, vol. 85, no. 25, pp. 6188–6190, Dec. 2004.

[21] B. Obradovic, P. Matagne, L. Shifren, E. Wang, M. Stettler, J. He, and M. D. Giles, "A physically-based analytic model for stress-induced hole mobility enhancement," *J. Comput. Electron.*, vol. 3, nos. 3–4, pp. 161–164, 2004.

[22] *User Manuals for Encounter Tool Version 6.2*, Cadence, Inc., 2007.

[23] *GNU Linear Programming Toolkit* [Online]. Available: http://www.gnu.org/software/glpk

[24] *Opensource Core Design Benchmarks* [Online]. Available: http://www.opencores.org

[25] *Opensource 45-nm Digital Cell Library* [Online]. Available: http://www.nangate.com

Ashutosh Chakraborty (S'04) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology Delhi, New Delhi, India, in 2002, and the M.S. degree in computer engineering from the University of Texas at Austin, Austin, in 2008. He is currently pursuing the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Texas at Austin.

From 2002 to 2004, he was a Senior Technical Staff Member with Mentor Graphics (India), Noida, India, where he worked on emulation products. From 2004 to 2006, he was a Research Assistant with Politecnico di Torino, Torino, Italy. He was with AMD, Sunnyvale, CA, in 2007 and Synopsys, Inc., Mountain View, CA, in 2009 as an Intern. He has published over 18 refereed papers in international conferences and journals. His current research interests include physical design considering the impact of mechanical stress and device reliability on nanometer integrated circuits.

Mr. Chakraborty received the Best Interactive Presentation Award at DATE 2009, the Best Paper Nomination at ISPD in 2010, the eASIC Placement Contest Grand Prize in 2009, the University of Texas Continuing Fellowship in 2009, the Government of Italy Fellowship in 2004, and the Rajiv Bhambhavle Best Undergraduate Thesis Award in 2002. He has served as a reviewer for several international conferences and journals, including DAC, SLIP, TCAD, TVLSI, TCAS-I, and TCAS-II.

Sean X. Shi received the B.S. degree in physics and the M.S. degree in microelectronics both from Peking University, Beijing, China, and the M.S.E. and Ph.D. degrees in computer engineering from the University of Texas at Austin, Austin, in 2001, 2004, 2008, and 2009, respectively.

From 2005 to 2009, he had been pursuing the Ph.D. degree from the Design Automation Laboratory, Department of Electrical and Computer Engineering, University of Texas at Austin. Currently, he is a Reliability Verification Design Automation Engineer with Intel Corporation, Austin, where he works on low power central processing unit design. He has published about six journal papers and 20 conference papers with three patents issued and one patent pending. His current research interests include modeling and optimization for different areas, including process simulation, device modeling, physical design, timing analysis, reliability verification, and lithography.

Dr. Shi has been awarded the IBM Ph.D. Scholarship, the Intel Scholarship on Information Science and Technology, the Yang Fuqing and Wang Yangyuan Academicians Scholarship, and the Honor of Innovation in Peking University.

David Z. Pan (S'97–M'00–SM'06) received the Ph.D. degree in computer science from the University of California, Los Angeles, in 2000.

From 2000 to 2003, he was a Research Staff Member with the IBM T. J. Watson Research Center, Yorktown Heights, NY. He is currently an Associate Professor and Director of the Design Automation Laboratory, Department of Electrical and Computer Engineering, University of Texas at Austin, Austin. He has published over 120 refereed papers in international conferences and journals, and is the holder of seven U.S. patents. His current research interests include nanometer very large scale integration (VLSI) physical design, design for manufacturing, vertical integration of technology, design and architecture, and design/computer-aided design for emerging technologies.

Dr. Pan has served as an Associate Editor for the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS (TCAD), the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-PART I, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-PART II, the IEEE CAS Society Newsletter, and the *Journal of Computer Science and Technology*. He was a Guest Editor of the TCAD Special Section "International Symposium on Physical Design" in 2007 and 2008. He serves as the Chair of the IEEE CANDE Committee and the ACM/SIGDA Physical Design Technical Committee. He is in the Design Technology Working Group of International Technology Roadmap for Semiconductors. He has served in the technical program committees of major VLSI/CAD conferences, including ASPDAC (Topic Chair), DAC, DATE, ICCAD, ISPD (Program Chair), ISLPED (Exhibits Chair), ISCAS (CAD Track Chair), ISQED (Topic Chair), GLSVLSI (Publicity Chair), SLIP (Publication Chair), ACISC (Program Co-Chair), ICICDT (Award Chair), and VLSI-DAT (EDA Track Chair). He was the General Chair of ISPD 2008 and ACISC 2009. He is a member of the Technical Advisory Board of Pyxis Technology, Inc., Austin, TX. He has received a number of awards for his research contributions and professional services, including the ACM/SIGDA Outstanding New Faculty Award in 2005, the NSF CAREER Award in 2007, the SRC Inventor Recognition Award thrice from 2000 to 2008, the IBM Faculty Award thrice from 2004 to 2006, the UCLA Engineering Distinguished Young Alumnus Award in 2009, the Best Paper Award from ASPDAC 2010, the Best Interactive Presentation Award from DATE 2010, the Best Student Paper Award from ICICDT 2009, the IBM Research Bravo Award in 2003, the SRC Techcon Best Paper in Session Award in 1998 and 2007, the Dimitris Chorafas Foundation Research Award in 2000, the ISPD Routing Contest Awards in 2007, the eASIC Placement Contest Grand Prize in 2009, five Best Paper Award Nominations (from ASPDAC, DAC, ICCAD, ISPD), and the ACM Recognition of Service Award in 2007 and 2008. He was an IEEE CAS Society Distinguished Lecturer from 2008 to 2009.