Incremental Timing-Driven Placement With Approximated Signoff Wire Delay and Regression-Based Cell Delay

Tai-Cheng Lee and Yih-Lang Li, *Member, IEEE*

***Abstract*— Satisfying timing requirements is the most challenging phase of the modern complex system-on-chip (SOC) design. The timing closure of the static timing analysis (STA) is a necessary but time-consuming stage before tapeout. Physical design tools are normally ineffective in obtaining accurate timing estimates, so the accurate timing calculation must be conducted in the signoff level timer after each physical modification on designs. This paper proposes a way to reduce the number of iterations between the signoff timer and the physical implementation procedures. Approximate timing models for extracting the signoff timer information and the nonlinear library are used in the optimization of the timing-driven placement (TDP). The accurate estimation of net and cell delays is integrated into TDP, so the optimal positions of cell movement can be obtained. This postoptimization algorithm was entered into the benchmark of the ICCAD15 incremental timing-driven contest, and the embodiments were obtained from the top three teams. Under the same design constraints, the proposed method yielded significant improvements in all kinds of default design chip.**

***Index Terms*— Physical design, postoptimization, timing optimization, timing-driven placement (TDP).**

I. Introduction

PHYSICAL design has become more challenging with the development of deep-submicrometer processes in the system-on-chip (SOC) design. To satisfy the signoff requirements, SOC implementation comprises several steps that involve checking against manufacturing design rules and the verification of circuit functions. The timing closure of an SOC design is critical to its functional correctness and chip efficiency because the absence of timing closure may lead to an absence of tapeout [1]. Gate sizing [2] and detailed placement have been used to address this issue in the postplacement optimization step. During the integrated circuit (IC) back-end design flow, the timing-driven placement (TDP) is one of the central stages for meeting timing demand [3]. Consistent with Moore's law, the SOC design scale increases year by year at advanced processes. This trend makes timing convergence more challenging and increases the run time for exact timing extraction. More cells in design mean more objects to be optimized and more timing constraint functions to be handled during TDP. In addition, fixing of several hundred blocks, IPs, or SRAM in the design strictly limits the cell movement and results in movable space discontinuities, which increases the challenge of finding placement solutions without large cell displacement and avoiding design rule checking (DRC) violations.

In timing signoff, processes used for the most advanced manufacture are often accompanied by highly complex timing estimations. In industrial design, a signoff timer (Primetime or Tempus) provides accurate timing information. Several iterations are required to resolve timing issues that do not arise using physical design tools. Electronic design automation (EDA) tools for producing a physical design can synthesize a layout that satisfies the timing requirement, based on the estimation using their own built-in timer. However, it is common to have several hundred timing violation paths, according to those reports calculated by a signoff timer. In the final tuning stage, iterations between physical design and signoff tools are exhausting with consequences for commercial competitiveness. Engineers must always perform extensive manual work and trial-by-error, together with time-consuming and computing resource intensive simulation.

TDP could solve the problem of timing violations by moving cells without increasing the power consumption. Linear programming has been used to address this issue by transforming all timing requirements and cell movement constraints into linear formats [4]–[6]. These methods used net-based weighting and handle huge amounts of limitations of potential cell movement. For modern designs with several million cells, such optimization seems impossible, owing to the huge runtime. The Lagrangian relaxation (LR) formulation was used to transform the global optimization into a separate local cell optimization [7]. It used the simple but fast linear cell and net model to estimate the timing. However, in modern timing models, such as the composite current source (CCS) model or nonlinear delay model (NLDM), the delay library cannot be modeled using linear formats. Moreover, cell capacitance was reduced in [8] and [9] to reduce the delay of the cells and improve the timing violations. If the routing

Manuscript received August 11, 2018; revised December 17, 2018 and March 2, 2019; accepted April 16, 2019. Date of publication June 6, 2019; date of current version September 25, 2019. This work was supported in part by the Ministry of Science and Technology of Taiwan under Grant MOST 105-2221-E-009-162-MY3. *(Corresponding author: Yih-Lang Li.)*

T.-C. Lee is with the Institute of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan (e-mail: leetaicheng.cs01g@g2.nctu.edu.tw).

Y.-L. Li is with the Department of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan (e-mail: ylli@cs.nctu.edu.tw).

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

Digital Object Identifier 10.1109/TVLSI.2019.2915254

is not based on the Steiner tree due to DRC or a concern about routing blockage, then the timing improvement may be ineffective. The works using simple linear and routing models to accelerate the timing closure likely retain timing violations on the signoff timer. Bock *et al.* [10] developed local TDP with decoupling of a specific timing model to shorten timing paths to reduce the number of violations as well as the total negative late slack. The signoff simulation seems to require more iterations than other studies. The work in [11] improved timing by shortening the wire length between cells on critical paths to reduce the timing violations of the whole design but also has the same weakness. In addition, several ECO tools designed for postoptimization are available on the commercial market. Although these tools propose timing improvement solutions for signoff designs, users encounter efficiency issues while attempting to integrate the tools into a physical design flow. This may limit the timing improvements because separate physical design tools cannot use information from ECO tools.

Linear-based models, used in previous studies, appear too simple, compared to nonlinear timing models. In this paper, accurate timer information is used in incremental TDP to reduce the timing slack. Each cell with negative slack is used to model the delay caused by the environment and the capacitance, as determined by the routing and cell libraries. This approach transforms a timing model into a relatively simple linear format with limited loss of accuracy and uses locally accurate timing information to enable local optimization while considering all timing paths.

In this algorithm, timing information is extracted using a signoff timer with high accuracy. In addition, approximate information can be obtained using a linear formula, which can be optimally efficient. The use of the approximate signoff timing model enables accurate estimation of cell movement and obtains the optimal solution, accelerating the closure of the timing signoff in the postoptimized design. Net weighting from LR is also used to ensure that improving the local timing violation does not negatively affect total timing. Ultimately, the clock tree is adjusted to help in achieving timing convergence.

The experimental results concern from given floorplannig, and each of the initial placements has different routings and timing violations. Our algorithm is applied to the ICCAD15 incremental timing-driven contest benchmark [12], the signoff timer, and the initial designs from the top three teams achieving the best timing improvements in the contest. The organizer of the competition released the improved designs, so they can be referenced by later researchers. Our methodology has significantly improved various default design chips. Our postoptimization method improved the quality scores of the top three teams by 93.04%, 167.82%, and 74.89% under a short distance constraint. With respect to the optimized design with the same restrictions on the cell movement, our method can provide further improvements using precise timer information.

Our contributions in this paper are as follows.

1) An approximate way to model local net timing—routing information—is considered in dynamic weighting. The Elmore model is combined with local routing information using each net weighting. Dynamic weighting is utilized to estimate the net delay and is especially suited to the small space of cell movement.
2) Linear regression technique to model a table-based cell library—cell information—can be modeled using linear approximate equations. These linear models are slightly less accurate than using tables but are significantly more efficient. Cell signatures are extracted as coefficient parameters, also avoiding the issue of sensitivity.
3) More efficient cell movement—refined LR weighting technique—considers more detailed path situations and provides more space for cell movement than transitional LR.
4) No impact on data paths of initial design—create useful skew—makes clock tree slightly unbalanced to solve the flip-flop with timing violation and to increase the shared length of launch and capture paths to improve the timing slack.

This paper is organized as follows. Section II introduces a basic formulation of the timing signoff requirement. Section III introduces approximate techniques for timing modeling and presents optimization flow with clock tree manipulation. Section IV presents experimental results, and Section V draws conclusions.

II. Basic Problem Formulation

Timing closure occurs when all timing paths in a design can satisfy the timing requirement at a specific clock frequency. This is needed for normal operation; eliminating all timing violations at all endpoints of timing paths is required for timing signoff. In late violation, also called the setup violation, the signal arrives at a register too soon before the previous signal, along the data path, is captured in the register after the necessary time frame. Define a and r as the arrival time and required time of a timing path. A path with a late timing violation is defined as $\min[0, r^{\text{late}} - a^{\text{late}}]$. When a register has an early violation, also called the hold violation, a signal has arrived too soon, so the previous signal cannot be successfully transformed. The violation is defined as $\min[0, a^{\text{early}} - r^{\text{early}}]$. The total negative slack (TNS) is the sum of the total negative timing slacks at all endpoints of a design. WNS is the worst negative slack. The basic form of late or early timing is

$$\begin{aligned} \text{TNS}_{\text{late}} &= \sum_{i \in F} \min\left[0, r_i^{\text{late}} - a_i^{\text{late}}\right] \\ \text{TNS}_{\text{early}} &= \sum_{i \in F} \min\left[0, a_i^{\text{early}} - r_i^{\text{early}}\right] \end{aligned} \tag{1}$$

where F is a combined set of input pins and primary output pins of all sequential cells; then, the WNSs are defined as

$$\begin{aligned} \text{WNS}_{\text{late}} &= \min_{i \in F} \left\{ \min\left[0, r_i^{\text{late}} - a_i^{\text{late}}\right] \right\} \\ \text{WNS}_{\text{early}} &= \min_{i \in F} \left\{ \min\left[0, a_i^{\text{early}} - r_i^{\text{early}}\right] \right\}. \end{aligned} \tag{2}$$

TDP aims to minimize TNS and WNS.

In nanometer designs, the accuracy, coverage, and efficiency of timing analysis are essential for the timing closure. Generally, the timing of a digital design can be verified using the static timing analysis (STA) that is a static method for

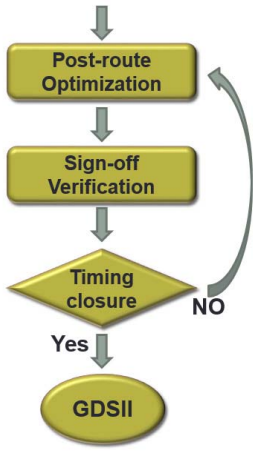

Fig. 1. Flow of iteration between the signoff verification and the postoptimization procedure.

validating the timing performance of all possible paths in a design. However, the timing estimation in traditional TDP differs from signoff level timing, especially at the advanced process node. Timing estimation of the signoff level normally requires routing information and has long runtime, hence cannot be used for placement optimization. Therefore, resolving these timing violations, which are not observed in physical design tools, takes several iterations to achieve the timing closure. Using the LR-based optimization, this paper increases timing quality during postoptimization. The proposed algorithm reduces the number of iterations between the signoff timer and physical implementation procedures (see Fig. 1).

III. Proposed Optimization Algorithm

To solve timing violations, TDP must involve more accurate timing estimation. Higher accuracy of estimated delay favors fewer iterations between the simulation of the signoff timer and the physical implementation. For a timing path, the delay can be split into two parts: cell delay and wire delay. Accurate timing information can be obtained from the signoff timer and cell models. The net weighting technique is modified and integrated into the developed optimization flow. In addition, the clock tree is adjusted to improve the timing slack.

A. Approximate Net Delay Model

The Elmore model [16] used a probability distribution function (pdf) to describe the impulse response and a cumulative distribution function to model the signal impulse response to estimate the net delay. The simplest model uses the first moment of the ideal impulse response of the signal, and it is sometimes not sufficiently accurate for a real design. To improve the accuracy of estimation, higher moments of the impulse response can be used, which requires a greater calculation burden and, normally, a need for more information, such as details about the routing situation. Therefore, order reduction-based methods have been proposed. Gupta *et al.* [13] proposed a method for extending any delay metric derived for a step input to a delay metric for a ramp input for RC trees that is valid over all input slews. Alpert *et al.* [14] have provided closed-form delay and slew approximate equations. Xu and Chowdhury [18] have used sophisticated methods and topological structural information to reduce runtime and increase accuracy, obtaining approximate results that are comparable to those of the asymptotic waveform evaluation (AWE).

During the calculation on the signoff timer, the net delay becomes more complex than previously discussed. A net will be separated into small segments and each RC value is extracted using an RC extractor, such as StarRC or QRC. A net routes through several metal layers and vias with various resistance and capacitance. Environmental noise and net crosstalk are considered at this stage. The interconnect capacitance causes different timing effects among near nets. The interconnect parasitism, the routing layer, and routing trees affect the net delay.

In the placement stage, the amount of net delay calculation markedly exceeds the number of cells, which is generally above several millions. Owing to the concerns about computability, the first-order moment of the Elmore model is very popular because of its linear closed form.

The net delay is estimated in a previous work as [6]–[9]

$$\text{delay} = k \cdot r \cdot l \cdot \left(\frac{c \cdot l}{2} + \text{cap}_{\text{pin}} \right) \tag{3}$$

where l is the Manhattan distance between pins, cap_{pin} is the capacitance of the input pin of a cell that is connected to the net, r is the unit resistance, c is the unit capacitance, and k is a factor that is set to 1 or 0.69 [6]–[9]. A net transition can be modeled similarly. However, in TDP, as a cell moves to a candidate position, routing must be estimated. If the cell has more than one output connection, the length of the net is estimated as half-perimeter wirelength (HPWL), significantly reducing the accuracy of the calculated net delay.

In the postoptimization procedure, cell movement is limited within a small area. Cell movement for timing convergence on all timing constraints must be very careful to prevent violating more DRCs or causing other timing issues. The strategy herein is to minimize the number of changes in the default design and apply a simple delay model to approximate the real timing using a dynamic coefficient. The net delay is estimated as

$$\text{delay}_{a \to b} = \alpha \times \widehat{\text{delay}}_{a \to b} \tag{4}$$

where $\widehat{\text{delay}}_{a \to b} = r \times l \times ((c \times l/2) + \text{cap}_{\text{pin\_}b})$ and α is the dynamic variable that corresponds to each net and will be updated after extracting the information from the signoff timer. Therefore, the dynamic weighting of the net between any two pins, a and b, can be calculated as

$$\alpha_{a \to b} = \text{delay}_{a \to b} / \left(r \times l \times \left(\frac{c \times l}{2} + \text{cap}_{\text{pin\_}b} \right) \right). \tag{5}$$

If the location of a cell is changed due to a timing issue, then its cell pins are also moved from their original location. The simple Elmore model with a dynamic factor can be used to approximate the net delay, influenced by the environment or the routing style.

During cell movement, the length of all connected nets will also be changed, and the net delay should be updated. The new net delay can be estimated as

$$\text{delay}_{\text{new}} = \alpha_{a\to b}\left(r \times l_{\text{new}} \times \left(\frac{c \cdot l_{\text{new}}}{2} + \text{cap}_{\text{pin\_}b}\right)\right). \quad (6)$$

If the net of a cell output pin has more than one sink pin, then the net delay can be estimated separately for every sink pin that is connected to the output pin. Using the information from the timer, the dynamic Elmore model's coefficient will be updated in every timer calculation. Accordingly, accurate results can be obtained more efficiently without using higher order moments or assuming a routing topology. The dynamic net weighting of our methodology is used to extract the timing information from the signoff timer with high accuracy. For different process technology nodes, the signoff timer can be designed accordingly. Our proposed method can be used to approximate the net delay, influenced by the environment or the routing style in various nodes for the timer. Therefore, this method can be decoupled from the timing calculation and applicable to different process technology nodes.

B. Approximate Cell Delay Model

1) Linear Model: Cell delay (or output slew) can be modeled as a linear combination of input slew at the input pin and output capacitance at the output pin. The linear delay and the slew model of standard cell Gcan be formulated as [4]–[7]

$$\begin{aligned}\text{Delay}_{\text{cell}} &= D_0 + D_1\text{slew}_{\text{in}} + D_2\text{cap}_{\text{out}}\\ \text{Slew}_{\text{cell}} &= S_0 + S_1\text{slew}_{\text{in}} + S_2\text{cap}_{\text{out}}\end{aligned} \quad (7)$$

where D_0, D_1, D_2, S_0, S_1, and S_2 are the library coefficients that depend on the library property of the standard cells. D_0 and S_0 denote the intrinsic delay and slew, respectively. slew_{in} is the transition time of the input pin of cell G. cap_{out} denotes the total downstream capacitance at the output pin of cell G. Cell model operations can be expressed in the linear format that is suitable for the LP-based optimal procedure. Every cell and net delay along a path are formatted linearly. The optimization object can be transformed into a linear programming program. The constrained functions are scalable when the limit on the cell movement is high and solutions can be efficiently obtained for timing optimization tasks.

Nowadays, the timing models of cell libraries in nanometer technologies are no longer represented by simple linear formulation. Owing to reliability considerations, cell delay models, such as NLDM or CCS format, are normally presented as a table-based model. The cell libraries provide a more accurate cell property under several conditions than the simple linear model, and NLDM or CCS presents data in a table format according to the rising or falling signals in the corresponding pins.

A delay (slew) table model may consist of two dimensions with two variables: input slew and output capacitance. If the required values are not available from all entries in the table, then interpolation or extrapolation may be used to obtain the missing values to derive the corresponding delay (slew). The shortcoming of interpolation or extrapolation for a table-based model is its sensitivity. Without loss of generality, an example of a 1-D table is demonstrated. In an interpolation procedure, two endpoints in the table are checked to estimate the library coefficient. Let c be a data point and d be its corresponding value. The linear interpolation is

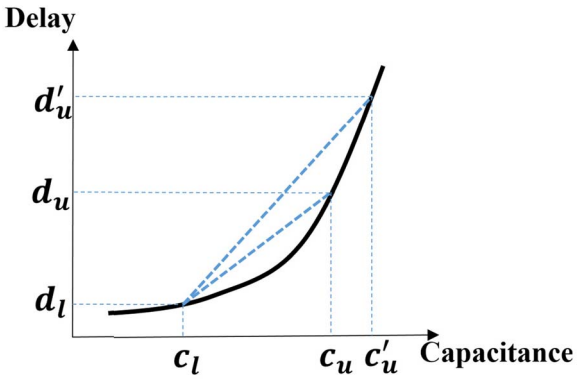

Fig. 2. Sensitivity of the interpolation method.

$$d_{\text{new}} = d_l + \frac{d_u - d_l}{c_l - c_u}(c_{\text{new}} - c_u) \quad (8)$$

and the slope of line is $(d_u - d_l)/(c_l - c_u)$.

Under this simplified cell delay model assumption, the cell delay is affected only by the cell output capacitance and is directly proportional to it. Therefore, the reduced expression is

$$\text{Delay} = D_0 + D_2 \times \text{cap}_{\text{out}} \quad (9)$$

After the cell movement, the cell delay is estimated as

$$\begin{aligned}\text{cell\_delay}_{\text{new}} &= \text{cell\_delay}_{\text{old}} + \Delta\text{cell}_{\text{delay}}\\ &= \text{cell\_delay}_{\text{old}} + \Delta\text{capacitainc} \times \text{coef}\end{aligned} \quad (10)$$

where coef is the slope. This formulation indicates that the estimated cap factor depends on the upper and lower bounds. Table-based models are, however, usually nonlinear. Choosing different boundaries may yield significantly different estimates of factors, as depicted in Fig. 2. In cell movement stage without accurate routing information, estimating the exact capacitance is difficult, potentially resulting in the unpredictability of values of the bounds. Hence, the interpolation (or extrapolation) method may fail in local approximation. In addition, regarding the complexity of computing, the timing calculation of each movable candidate location of every cell will access the timing table to calculate the corresponding estimates of library coefficients by interpolation or extrapolation. The computational burden is expected to be very high. However, it should be noted that if the delay model is extended beyond linear assumptions, as in the case of low voltages, the linear approximation will lose significant accuracy. This situation will reduce the optimization performance of all linear programming-based algorithms.

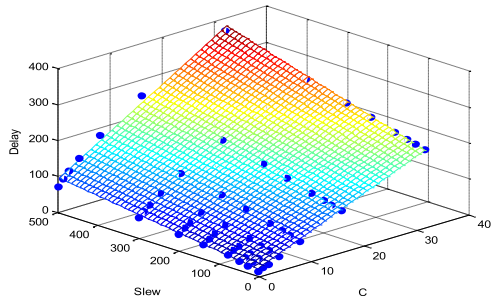

Fig. 3. Linear regression model for cell delay.

2) Linear Regression Model: To reduce the computational complexity, a timing model should be approximated as a linear model. Accordingly, the optimization task can be transformed into a linear programming-based problem. The least-square estimation of the regression coefficients is used in this paper. Table-based functions are approximated as linear functions. Each table from the cell libraries can be used to build a model that represents this table, and it can be easily integrated into a linear programming procedure. Keeping the model concise helps to prevent overfitting and reduces the error of the estimation in extrapolation. The basic optimization principle in linear regression is based on the least square. The linear regression model is as follows:

$$Y = X_1\beta_1 + X_2\beta_2 + \cdots + X_n\beta_n + e \tag{11}$$

where $\beta_1, \ldots, \beta_n$ are the parameters of the linear model.

For T sets of observed values of Y, the formula can be written in matrix notation as

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{e} \tag{12}$$

where X is an independent set, $\boldsymbol{Y}' = (y_1 \ldots y_T)$, $\boldsymbol{\beta}' = (\beta_1, \ldots, \beta_T)$, and $\boldsymbol{e}' = (e_1, \ldots, e_T)$. The least-square method is used to measure the quality of the estimate β. The procedure of this stage is to minimize the sum of the square residuals [15]

$$S(\boldsymbol{\beta}) = \sum_{t=1}^{T} e_t^2 = \boldsymbol{e}'\boldsymbol{e} = (\boldsymbol{Y} - \boldsymbol{X\beta})'(\boldsymbol{Y} - \boldsymbol{X\beta}) \tag{13}$$

The principle of estimation involves minimizing the sum of the squared residuals. If X is a full-rank matrix (rank $= K$), then the unique optimal solution is [15]

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y} \tag{14}$$

where $\widehat{\boldsymbol{\beta}}$ is an estimation vector of $\boldsymbol{\beta}$. Estimator ensures that the sum of squared residuals between this model and the actual data set is minimal [15].

The solution of (14) is applied to the table-based timing model. In a 2-D table, each value corresponds to two variables. Geometrically, two table coordinates and the corresponding value can be plotted in a 3-D space (see Fig. 3). The least-square estimation of the regression coefficients organizes a 2-D plane in a 3-D space. Consider, for instance, a cell delay table model that has the parameters input slew and output capacitance. Linear regression can approximate the relationship between the cell delays that correspond to input slew and output capacitance, β_{cap} and β_{slew}. Values in $\boldsymbol{X}$ denote the combinations of slew and output capacitance. $\boldsymbol{Y}$ is the corresponding cell delay value of $\boldsymbol{X}$. By (14), $(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}$ attains the minimum sum of the squared errors of predictions among all possible linear solutions.

The linear model in (7) can be replaced with the regression model. During optimization, the regression model can be integrated into the timing estimation procedure. For example, the original cell delay that is extracted from the signoff timer can be utilized on cell movement, and then, the number of changes in capacitance and slew can be adjusted as follows:

$$\text{delay}_{\text{cell}}^{\text{new}} = \text{delay}_{\text{cell}}^{\text{orig}} + \beta_{\text{cap}}\Delta\text{Cap} + \beta_{\text{slew}}\Delta\text{Slew} \tag{15}$$

In Fig. 3, every dot represents a datum in the table model from the cell library [11]. The plane is constructed using the linear regression model

$$\text{delay} = \beta_0 + \beta_{\text{cap}} * \text{Capacitance} + \beta_{\text{slew}} * \text{Slew} \tag{16}$$

where the x-axis represents the capacitance, the y-axis represents the slew, and the z-axis represents the delay. The difference between the estimated values and the real data is small, showing that this model is reasonably accurate.

The regression method is used in the table-based modeling. A set of data in a timing table can be approximated using a linear formulation. Cells of each type are constructed in the beginning and can be reused during the overall optimization stage.

The coefficient of determination R^2 provides a measure of how well a mathematical model accounts for variations in a data set [19]. A high coefficient of determination R^2 means a mathematical model adequately explains variations in a given data set. We use R^2 to judge the accuracy of our models for data obtained from a table-based library. In our experiments, linear regression is good enough to fit the cell library for the benchmark [12], and all of our R^2 values are above 95%.

A regression-based technique can be used to fit all kinds of table-based models in a cell library and can be integrated into the optimization flow. As far as we know, the process technology nodes that are 7 nm or less use a table-based format to describe the timing model under different conditions. When linear approximation cannot fit the data sufficiently and a linear model cannot sufficiently explain the variation of nonlinear data, the idea of regression-based methodology can still be extended to nonlinear formats and can be applied to represent nonlinear data. We demonstrate two techniques: piecewise regression and polynomial regression [19]. First, the data can be represented by the continuous piecewise regression. Without the loss of generality, we model a 1-D table. The range of X is separated into two or several intervals, and different linear equations are used to model each interval. Fig. 4(a) demonstrates one continuous piecewise regression with one knot ξ to fit the data. The artificial data set is fit by two linear models with a continuous point. This model can also be applied to the data using several separate linear models. Second, polynomial regression, a form of regression analysis, can be used to describe the nonlinear phenomena. Fig. 4(b) demonstrates that the data can be modeled with a quadratic equation. The higher order of polynomials can be

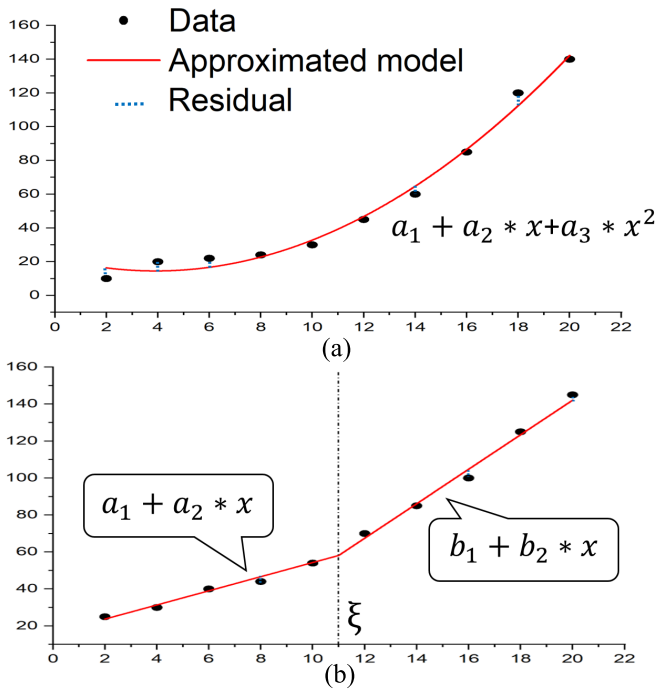

Fig. 4. (a) Continuous piecewise regression and (b) quadratic regression for artificial nonlinear data sets.

Algorithm: Cell movement

Input: a movable cell c, a candidate location R

1. Estimate new net length and capacitance from location R
2. $C_{up} \leftarrow \{u|\text{upstream cell } u \text{ of } c\}$
3. **foreach** upstream cell u of C_{up}
4. $t_{init} \leftarrow$arrival time from u
5. $\Delta t_{net} \leftarrow \alpha \cdot (\Delta\text{net between } u \text{ and } C)$
6. $\Delta t_{cell} \leftarrow f_u(\Delta\text{capacitance}, \Delta\text{slew})$
7. $t_{new} \leftarrow t_{init} + \Delta t_{net} + \Delta t_{cell}$
8. **end foreach**
9. $\Delta T \leftarrow \max\{t_{init} \text{ of } u | \forall u \in C_{up}\} - \max\{t_{new} \text{ of } u | \forall u \in C_{up}\}$
10. $C_{down} \leftarrow \{u|\text{downstream cell } u \text{ of } c\}$
11. $cost \leftarrow 0$
12. **foreach** downstream cell d of C_{down}
13. $t_{init} \leftarrow$ arrival time to $d + \Delta T$
14. $\Delta t_{net} \leftarrow \alpha \cdot (\Delta\text{net between } C \text{ and } d)$
15. $\Delta t_{cell} \leftarrow f_d(\Delta\text{capacitance}, \Delta\text{slew})$
16. $t_{new} \leftarrow t_{init} + \Delta t_{net} + \Delta t_{cell}$
17. $cost \leftarrow cost + t_{new} \cdot \omega_{c \to d}$
18. **end foreach**
19. **return** *cost*

Fig. 5. Pseudocode of the cell movement.

further used to fit the data. In our experiments, continuous piecewise regression and quadratic regression models obtained values above 99% for all cell libraries. If these models still cannot represent a data set, the models can be combined using the continuous piecewise polynomial regression. All the approximation models introduced earlier can be integrated into our optimization flow by using the first-order Taylor approximation [19].

C. Design Flow With Violation Path-Based Net Weighting

LR is an important technique for reducing the timing violation. It has been applied to gate sizing and to incremental TDP for timing convergence [2], [7], [20]. Weightings are incorporated into objective functions. This method can decouple the global optimization into simpler local timing object functions.

LR weighting should satisfy the following [2]:

$$\begin{aligned} &\forall n : \sum_{w \to u} n_{w \to u} = \sum_{u \to v} n_{u \to v} \\ &\forall I : n_I = \sum_{I \to v} n_{I \to v} \\ &\forall O : \sum_{v \to O} n_{v \to O} = n_O \end{aligned} \tag{17}$$

By the first constraint, for any node n, the sum of input weighting must be equal to the sum of output weighting. The other two constraints concern the weightings of the primary inputs and outputs of the design. The LR values at the endpoints of all timing paths are assigned, and the internal cells throughout the design are traced backward from the endpoints. Every node must satisfy the constraints in (17), and the net weighting is updated after every timing calculation. Based on the net weighting, the cell movement or gate sizing not only considers the optimization of local timing but also satisfies the requirement of other global timing paths.

Intuitively, the output weightings of a cell can be regarded as an impact factor of downstream branches of the output pin. These weightings can quantify the effect of a cell on its downstream endpoints. A node with a greater weighting has more downstream endpoints or fewer branches from its endpoints. A node that receives a greater weighting from its endpoints has more impact on all timing paths. Our modified LR weighting is used to reduce the need for redundant cell manipulations. All endpoints with positive timing slack are traced, and their initial weightings are set to zero; the endpoints with timing violation (negative slack) are set to a positive value. If the weighting of an endpoint is set to zero, then all cells in the design, which are related to this endpoint, have a zero from this endpoint and do not need to consider this branch path during cell movement. This step will enable optimization for every cell, ignoring all flip-flops with positive timing.

Unlike gate sizing studies, the time-driven process usually involves a tradeoff. The movable position of a cell has the property of exclusiveness. In a legalized design, two cells cannot be placed in the same position. Sometimes, the movement of a cell may not satisfy all of the downstream timing paths. In the presented weighting method, adjusting the position of a cell can occupy the space from the positive slack paths to the negative paths. Timing paths with violations have more space to find the optimized position in the design. Cells will be attracted to the nets with higher weighting. Sometimes, the cell movement will negatively influence the timing paths with positive slacks. However, more space will be extracted to let violating paths meet timing closure. This concept can be regarded as borrowing timing from positive paths for negative paths.

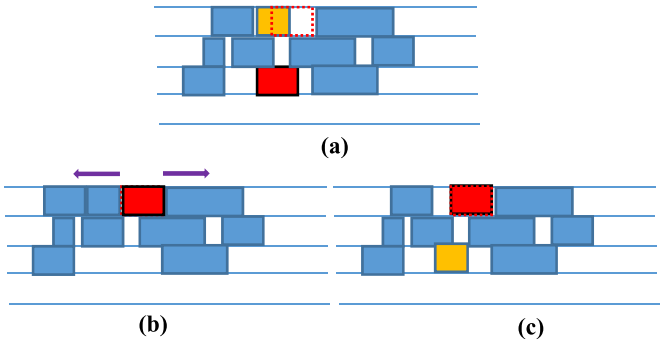

Fig. 6. Legalization. (a) Location finding for cell move. (b) Limited cell move for adjacent cells. (c) Cell swap.

Fig. 5 displays the cell movement algorithm. Every cell in the design with negative weighting optimizes its position by choosing the movable candidate position with the lowest cost function. Parameter α is obtained from the dynamic net weighting, f represents a linear regression model of the cell, and ω is the refined LR weighting. The new net length and capacitance at the candidate location are estimated (line 1). For each upstream cell of c, the new arrival time is computed from net weighting and cell regression functions (lines 4–7). The increase or decrease arrival timing at the output pin affected by the arrival time of all input pins is computed in line 9. For each downstream cell d of c, a pretiming delta is added to the initial input pin arrival time of d in line 13. All cost function values from the downstream net are accumulated (lines 13–17). The new arrival time at the input pin of d is computed (lines 14–16). The LR weighting ω is integrated in line 17. Ultimately, this algorithm returns a cost value that corresponds to its new candidate position.

Using the previously described algorithm, candidate locations will be ranked by their improvement of the timing. Candidates that do not improve timing will be ignored. In the best case, a cell can be placed at the candidate location without causing any violation. However, searching the optimal candidate location for cell movement always causes cell overlapping, especailly in a design with high utilization. If a candidate location overlaps a fixed cell or macro, it will be abandoned and the next good one will be identified. If a candidate overlaps a movable cell [see Fig. 6(a)], then legalizer will legalize this design and will try to shrink the available space to the right and left of it [see Fig. 6(b)]. This paper attempts to minimize the physical impact of the cell movement on the initial design but to maximize the timing improvement. During this procedure, every cell that has moved remains close to its original location. The legalizer herein may not be able to find enough space, so the candidate location will be abandoned. In some cases, the candidate location overlaps a movable cell whose net weighting is zero. Therefore, the movable cell has positive timing slack and room to be moved farther from its original location, negatively affecting its timing slack at some time. The legalizer will swap the locations of the cells [see Fig. 6(c)] and continue attempting to find sufficient free space. Compared to [11], our optimizer not only moves a cell to free space but also creates useful space by swapping cells on noncritical paths. Overall, our legalization can ensure that each cell placement is legal and that these cell movements will not worsen the original timing.

Algorithm: Creation of common path

Input: a register C

1. **If** C has positive slack **then return**
2. $L \leftarrow$ Pre-register of C
3. $B \leftarrow \{u|$buffer u and connect to ck pin of $C\}$
4. $B' \leftarrow \{u'|$buffer u' and connect to ck pin of $L\}$
5. **If** (B or B' is empty set) or (B equal B') **then return**
6. **If** paths have common buffer is compensated for timing
7. **If** C has early negative slack **then**
8. Connect to common buffer to increase length of launch path
9. **If** C has late negative slack **then**
10. Connect to common buffer to increase length capture path
11. **return**

Fig. 7. Pseudocode of the creation of the common path.

D. Clock Tree Adjustment

Manipulating the clock tree can significantly affect timing. The timing estimation is highly accurate for the real timing of tapeout chips after performing the timing calculation on the signoff level. The timing violation is usually close to the required boundary of timing that can be eliminated by slightly changing the clock net. In the final stage of optimization, adjustments to the clock tree efficiently reduce the negative slack timing. In the following discussion, two principles are applied to reduce the timing slack by adjusting the clock tree.

1) Creation of Useful Clock Skew: Useful skew has been applied to improve the system performance in a previous study [21]. A created incremental clock tree skew can reduce the timing violation of the initial design without having any negative effect on timing or DRC. The clock tree should be modified carefully because of its high sensitivity on timing. First, late and early violations are traded off. The arrival time should be within a legal period at each corner of the process–voltage–temperature. Second, two adjacent sequential circuits, which only have combinational circuits between each other, also exhibit a timing tradeoff. Improving one timing at a sequential circuit may influence the other since the adjacent sequential circuit may be affected.

The adjacent registers are modified to reduce the negative slack, and then, the timing budget is borrowed from the preregister or postregister (adjacent flip-flops). If a register exhibits an early timing violation, the launch path will arrive too early. The clock tree can be adjusted to modify the signal path in two ways: by increasing the length of the launch path or by reducing the length of the capture path. In the case of a late violation, the method is reversed. Modifications to the signal tree should be carefully performed to ensure that they will not cause early or late timing violations in the previous- or next-level flip-flops. In addition, the algorithm balances the fan-out of the cell that we are trying to adjust. Some noncritical nets will be swapped to control the loading of the specific cell.

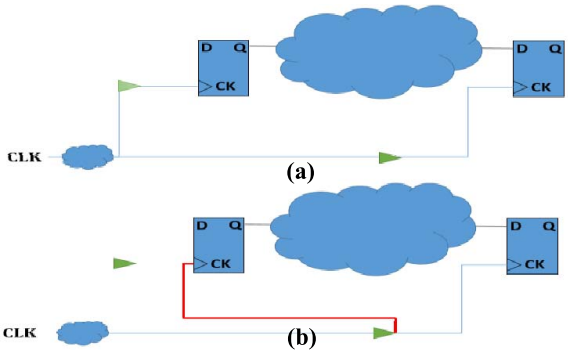

Fig. 8. Common path creation for solving early slack. (a) Identify clock buffers with compensated values that favor early slack. (b) Select buffers to increase the capture path delay.

A steep increase in the loading of a cell would be avoided in the flow. On the other hand, our algorithm properly limits the net length increase to avoid increasing the risk of IR drop. Overloading the capacitance from the fan-out net will not only cause an IR drop issue but will also cause a routing challenge. In our experiments, a large net length increase tends to cause routing failure.

2) Creation of Common Path for Cock Tree: The delay of a timing path is calculated using different corners of cell libraries and net derating factors, depending on whether the path is a launch path, using the slow (fast) corner on calculating late (early) violation, or a capture path, using the fast (slow) corner on calculating late (early) violation. However, the cells on the common path shared on capture and launch path should not use different corner model. The common data path and clock path may misestimate the timing violation.

Fig. 7 displays the pseudocode of the common path generation. This procedure is only applied to registers with negative slack (line 1). Let C present a register with negative slack. Search the preregister (L) of C, and clock buffers connected to the registers are identified (lines 3 and 4). Whether connecting them (C and L) to the same clock buffer will compensate for timing is estimated (line 6). If the compensation is positive for timing, one of the cells is chosen for reconnection to the common signal path. In early violation case, the algorithm chooses the cells close to the pre-flip-flop (C) and far from the launch flip-flop (L) (lines 7 and 8). This step increases the capture path delay and reduces the launch path delay (see Fig. 8). The same principle is applied to reduce the length of the launch path and increase the length of the capture path in late violation case (lines 8 and 9). The common path procedure can reduce small timing slack without moving any cells.

E. Flow Summary

Fig. 9 presents the overall flow of the creation of a common path. Each type of cells is built as a linear regression model at the beginning and can be reused in the overall optimization stage (line 1). In line 2, the LR weighting is updated. The dynamic net weighting for delay estimation is extracted from the timer (line 3). In lines 4 and 5, the clock tree is adjusted to reduce the timing violation close to the required timing

Algorithm: Flow of timing-driven placement

Input: a physical design D, sign-off timer information, and cell library

1. Cell model extracted by Linear regression
2. Update LR weighting
3. Update dynamic net weighting
4. Creation of common path
5. Useful Clock Skew Creation
6. **foreach** register cell u with negative lack
7. **foreach** upstream cells of u
8. Perform cell movement optimization
9. Legalize placement
10. **end foreach**
11. **end foreach**
12. Cell density reduction
13. **return**

Fig. 9. Flow of our TDP.

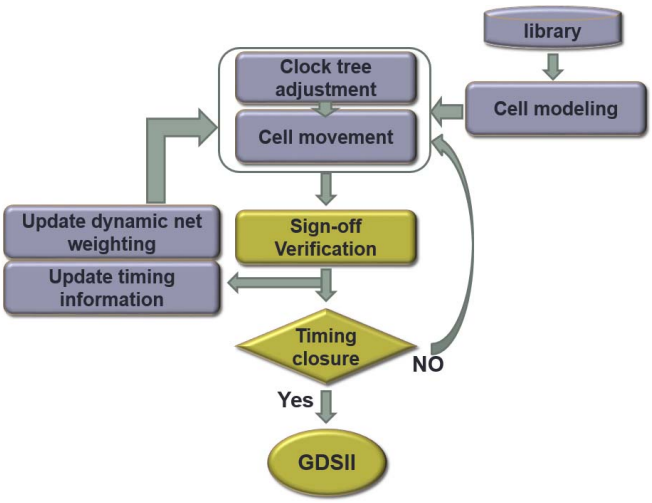

Fig. 10. Overview of the postoptimization.

boundary. Every register with negative slack and its upstream cells perform the cell movement optimization (lines 6–11). In this manner, the best candidate location for a cell will be chosen according to the cost function, and then, the cell movement is realized. If an overlap occurs, legalization is performed. BFS is performed to trace all upstream cells on timing paths. Eventually, the cell density will be reduced by the detailed placement.

Potential process, voltage, and temperature conditions should be considered in the operation of the design under different modes. In STA, several scenarios containing different process–voltage–temperature conditions should be considered concurrently. Timing closure needs to be achieved in all these scenarios. Each timing path in the multimode multicorner (MMMC) should satisfy the timing requirements. Fig. 9 demonstrates the optimization of one scenario. Our algorithm can be applied to each of them to improve timing.

At advanced process nodes, there are increasing gaps in the timing estimation between the signoff timer and the physical implementation. These gaps become worse when the number of MMMC increases and requires long runtime to overcome.

TABLE I
IMPROVEMENT RATE OF SHORT DISPLACEMENT CONSTRAINTS

Short	Benchmarks	Super.1	Super.3	Super.4	Super.5	Super.7	Super.10	Super.16	Super.18	average
cada014 (first place)	Wns Early	100.00%	16.87%	7.28%	6.70%	0.00%	48.13%	99.25%	42.56%	40.10%
	Tns Early	100.00%	79.38%	85.21%	61.95%	3.83%	54.86%	99.80%	95.90%	72.62%
	Wns Late	1.63%	2.14%	0.78%	0.62%	0.00%	-0.22%	2.26%	0.71%	0.99%
	Tns Late	1.63%	3.85%	2.77%	1.25%	5.64%	3.10%	14.84%	1.18%	4.28%
	ABU	83.33%	66.67%	40.00%	100.00%	66.67%	50.00%	75.00%	75.00%	69.58%
	StWL (um)	-0.71%	-0.39%	-0.71%	-0.55%	-0.99%	-0.64%	-0.64%	-0.53%	-0.65%
	Quality Score	25.91%	57.69%	34.01%	360.43%	62.89%	130.60%	49.91%	22.85%	93.04%
cada070 (second place)	Wns Early	45.86%	80.87%	-1.53%	7.28%	-0.10%	5.57%	42.46%	43.89%	28.04%
	Tns Early	78.78%	91.35%	75.12%	56.46%	3.39%	79.95%	77.33%	83.62%	68.25%
	Wns Late	3.84%	5.00%	3.52%	2.16%	0.00%	0.89%	5.93%	6.12%	3.43%
	Tns Late	11.37%	6.44%	7.72%	3.26%	8.56%	3.81%	28.12%	6.46%	9.47%
	ABU	83.33%	66.67%	40.00%	100.00%	66.67%	50.00%	66.67%	75.00%	68.54%
	StWL	-0.70%	-0.39%	-0.67%	-0.55%	-0.99%	-0.64%	-0.67%	-0.54%	-0.64%
	Quality Score	109.29%	169.19%	220.36%	108.08%	213.31%	313.35%	88.95%	120.00%	167.82%
cada085 (third place)	Wns Early	5.61%	66.24%	0.00%	8.00%	-0.10%	0.00%	100.00%	46.98%	28.34%
	Tns Early	85.31%	88.28%	78.86%	39.76%	2.01%	78.44%	100.00%	59.72%	66.55%
	Wns Late	0.66%	4.06%	1.43%	0.53%	0.00%	-0.02%	2.00%	1.32%	1.25%
	Tns Late	4.17%	7.72%	3.81%	2.72%	8.08%	4.07%	34.86%	5.85%	8.91%
	ABU	80.00%	66.67%	25.00%	100.00%	66.67%	50.00%	66.67%	75.00%	66.25%
	StWL	-0.73%	-0.40%	-0.72%	-0.57%	-0.99%	-0.66%	-0.69%	-0.55%	-0.66%
	Quality Score	19.93%	99.83%	71.60%	51.68%	118.70%	90.88%	93.81%	52.69%	74.89%

Several dozen iterations are necessary to resolve these timing issues that do not arise when using physical design tools. In industrial design, engineers must perform extensive manual work and trial-by-error under tight schedules for tapeout. Our methodology integrates accurate timing information for the optimization of the physical implementation. Fig. 10 presents the embedding of the algorithm into a standard signoff flow before tapeout.

IV. EXPERIMENTAL RESULTS

The implementation is written in C++ and compiled with g++ 4.9.2. All benchmarks are executed on a quad-core 3.00-GHz Intel Xeon CPU E5450 server with 48-GB RAM. The operating system is CentOS release 5.3 (Linux kernel 2.6.18-128.2.1.el5).

The metrics for evaluating an incremental design include quality improvement and routing penalty. The judgment is provided by the International Conference on Computer-Aided Design (ICCAD) 2015 contest [12]. The improving rate could be defined as improving rate of late TNS $= (\text{TNS}'_{\text{late}}/\text{TNS}_{\text{late}})$.

The total quality improvement metric is

$$w_{\text{TNS}}\left(w_{\text{late}}\left(1-\frac{\text{TNS}'_{\text{late}}}{\text{TNS}_{\text{late}}}\right)+w_{\text{early}}\left(1-\frac{\text{TNS}'_{\text{early}}}{\text{TNS}_{\text{early}}}\right)\right) + w_{\text{WNS}}\left(w_{\text{late}}\left(1-\frac{\text{WNS}'_{\text{late}}}{\text{WNS}_{\text{late}}}\right) + w_{\text{early}}\left(1-\frac{\text{WNS}'_{\text{early}}}{\text{WNS}_{\text{early}}}\right)\right) \quad (18)$$

where the weighting is $\{w_{\text{TNS}}, w_{\text{TNS}}, w_{\text{late}}, w_{\text{early}} = \{2, 5, 1, 2\}$.

The slack improvement score of the difference design with the overfill penalty is obtained from

$$\max\{\text{slac\_improve} \times (1 - \Delta\text{OverBinUtility}), 0\}. \quad (19)$$

A. Benchmark Suite

The ICCAD 2015 contest organizers released a realistic benchmark suite and provided a framework for timing evaluation in academic environments [12]. This suite consists of eight designs, each with industry-standard formats of physical design (.v,.lef,.def,.sdc, and.lib). Each design includes the specific initial floorplannig and placement. The main goal of the contest is to reduce the timing violation under long and short maximum cell displacement constraints. All cells in a design must be aligned on rows and sites of the floorplan. Some of the cells and macros are fixed. No overlap is permitted. After placement optimization, real routers are applied to route and extract the parasitic RC of the routed wires, which reflects the impact of routing on various cell placement solutions. An academic signoff timer with common path pessimism removal (CPPR) is also provided. Timing information is presented using industry-standard interconnect modeling, a Standard Parasitic Exchange Format (.spef) file, circuit element modeling, and nonlinear delay models that contain multidimensional tables, which are recorded in a Liberty (.lib) file.

B. Analysis of Results

Eight initial preoptimization realistic benchmarks were released by the ICCAD incremental TDP contest organizers. Each design was judged based on the improvement in quality score, obtained using (18) and (19). The algorithm presented herein followed the contest rules, including the constraint on the cell placement distance and all legal checks.

TABLE II
IMPROVEMENT RATE OF LONG DISPLACEMENT CONSTRAINTS

Short	Benchmarks	Super.1	Super.3	Super.4	Super.5	Super.7	Super.10	Super.16	Super.18	average
cada014 (first place)	Wns Early	100.00%	54.62%	100.00%	100.00%	0.00%	100.00%	100.00%	100.00%	81.83%
	Tns Early	100.00%	96.85%	100.00%	100.00%	4.57%	100.00%	100.00%	100.00%	87.68%
	Wns Late	0.78%	0.85%	0.46%	0.22%	0.00%	0.38%	1.93%	0.65%	0.66%
	Tns Late	-0.87%	2.60%	2.91%	1.07%	4.42%	3.21%	11.06%	-0.29%	3.01%
	ABU	83.33%	66.67%	40.00%	100.00%	66.67%	50.00%	75.00%	75.00%	69.58%
	StWL (um)	-0.71%	-0.41%	-1.01%	-0.62%	-0.94%	-0.83%	-0.68%	-0.74%	-0.74%
	Quality Score	73.72%	13.32%	30.99%	184.91%	24.91%	120.99%	24.07%	5.80%	59.84%
cada070 (second place)	Wns Early	100.00%	100.00%	35.38%	100.00%	-0.10%	100.00%	100.00%	100.00%	79.41%
	Tns Early	100.00%	100.00%	92.40%	100.00%	4.43%	100.00%	100.00%	100.00%	87.10%
	Wns Late	2.37%	2.97%	3.71%	1.81%	0.00%	1.46%	5.06%	3.37%	2.59%
	Tns Late	8.78%	4.01%	7.04%	2.42%	5.80%	4.24%	23.77%	5.17%	7.65%
	ABU	80.00%	66.67%	25.00%	100.00%	66.67%	50.00%	66.67%	66.67%	65.21%
	StWL	-0.71%	-0.38%	-1.01%	-0.55%	-0.95%	-0.75%	-0.67%	-0.76%	-0.72%
	Quality Score	178.32%	26.15%	69.69%	40.21%	172.16%	54.45%	86.66%	35.99%	82.95%
cada085 (third place)	Wns Early	100.00%	100.00%	100.00%	99.77%	-0.10%	67.87%	29.90%	100.00%	74.68%
	Tns Early	100.00%	100.00%	100.00%	99.98%	2.37%	97.58%	8.70%	100.00%	76.08%
	Wns Late	5.16%	2.77%	3.79%	0.40%	0.00%	-0.31%	2.41%	1.02%	1.91%
	Tns Late	11.22%	5.55%	7.75%	2.31%	6.09%	4.04%	34.25%	3.86%	9.38%
	ABU	80.00%	66.67%	25.00%	100.00%	66.67%	50.00%	66.67%	75.00%	66.25%
	StWL	-0.79%	-0.42%	-1.17%	-0.57%	-0.92%	-0.82%	-0.69%	-0.75%	-0.77%
	Quality Score	*(434/0)	23.03%	*(384/0)	62.77%	47.60%	119.29%	55.68%	19.96%	-

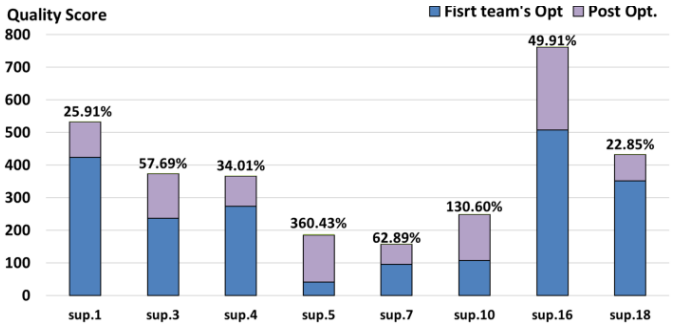

Fig. 11. Improvement in the postoptimization quality score for the team in the first place.

1) Analysis of Postoptimization Results: In this section, initial placement solutions are obtained using the optimized designs of the top three teams. Forty-eight designs satisfied the short and long distance constraints. This paper aims to perform postoptimization. Tables I and II present the details of performance improvement. Tables I and II also present the WNS, TNS, average bin utilization (ABU), and the quality metrics obtained by each of the top three teams in the contest. Table II presents the runtime information, and Table III presents detailed information about the initial design solutions of the top team.

The blue bars in Fig. 11 present the results for eight of the designs from the top team. Postoptimization may have a negative effect on an original optimized design. However, the postoptimization methodology herein increased all of the quality scores of the top three teams (see Table IV). With respect to improving the timing slack (early/late on the worst and total paths), the result has a positive effect in 31 cases and a slightly negative effect in only one. The proposed postoptimization process shifted the eight designs of the best team significantly closer to the state of timing closure. Notably, for each improvement optimization, the default displacement constraints are satisfied. The improved percentages of quality scores are between 22.85% and 360% (see Table I). Overall, the solutions of the top team were improved by 93%, indicating that even under the strict constraint, our optimization procedures do not jeopardize initial timing but find further solutions to improve timing. Similar results are obtained with respect to the initial designs of other teams. In all of the experimental cases, significantly better solutions are obtained. In cases with short distance constraints, this postoptimization method improves the scores of the top three teams by 93.04%, 167.82%, and 74.89% on average, respectively. During these 24 different optimized placement designs, our method can make designs much closer to meeting signoff requirements. Adjustment of the clock tree to solve the problem of early slack and cell movement for improving late slack is almost getting improved. Even though the slack on a few of the worst paths is slightly increased, the overall improvement is significant.

In cases with long distance constraints, our postoptimization method further improves the total scores of the top three teams. Our proposed algorithm and approximate models for extracting signoff timer information are especially beneficial for local optimization with the limited moving distance of 20 μm to each cell and the designs with minimal physical modification, such as the cases with short distance constraints. The average rates of improvement exceed 50%, 59.84%, and 82.95%; the third improving rate in two of the cases is unavailable because the denominator is zero. Our algorithm improved the scores in all 24 designs. Note that some improved scores cannot be calculated because of division by zero, and these are ignored in calculating the average.

TABLE III
RUNTIME OF ALL DEFAULT DESIGNS (SECONDS)

Benchmarks		Super.1	Super.3	Super.4	Super.5	Super.7	Super.10	Super.16	Super.18
cada014	Short	292	403	1243	1483	1758	9718	1259	350
	Long	220	331	1092	1322	1573	9677	715	319
cada070	Short	308	416	1235	1404	1842	9880	1420	355
	Long	296	381	1301	1370	1817	9867	1302	352
cada085	Short	276	419	1254	1405	1772	9761	1535	355
	Long	286	393	1274	1428	1642	9843	1374	338

TABLE IV
DETAILED INFORMATION OF POSTOPTIMIZATION ON THE FIRST PLACE TEAM UNDER SHORT CONSTRAINTS

Benchmark	Solution	Wns Early	Tns Early	Wns Late	Tns Late	ABU	StWL (um)	Quality Score
sup.1	team	-3.827	-41.56	-4669.6	-374312	0.06	9.60E+07	423.00
	ours	0.000	0.00	-4593.6	-368210	0.01	9.67E+07	532.62
sup.3	team	-65.719	-683.51	-9436.6	-1373120	0.03	1.14E+08	236.68
	ours	-54.632	-140.94	-9234.5	-1320300	0.01	1.15E+08	373.23
sup.4	team	-6.080	-173.92	-5942.8	-3195330	0.05	7.15E+07	273.60
	ours	-5.637	-25.72	-5896.8	-3106760	0.03	7.20E+07	366.66
sup.5	team	-36.769	-585.78	-25075.8	-6779950	0.02	1.08E+08	40.18
	ours	-34.305	-222.89	-24920.9	-6695390	0.00	1.08E+08	185.00
sup.7	team	-6.753	-1943.74	-15216.3	-1703780	0.03	1.40E+08	96.03
	ours	-6.753	-1869.23	-15216.3	-1607720	0.01	1.42E+08	156.42
sup.10	team	-8.619	-361.06	-16187.1	-32514400	0.04	2.05E+08	107.52
	ours	-4.471	-163.00	-16222.0	-31506800	0.02	2.07E+08	247.94
sup.16	team	-8.377	-30.67	-4363.7	-514250	0.04	9.36E+07	507.84
	ours	-0.063	-0.06	-4265.3	-437926	0.01	9.42E+07	761.31
sup.18	team	-3.806	-69.38	-4121.3	-943640	0.04	5.77E+07	351.36
	ours	-2.186	-2.85	-4092.1	-932520	0.01	5.80E+07	431.64

Within two iterations in the signoff simulation experiment under the same design constraints, updated timing information is demonstrated to improve the timing quality. Fig. 12 shows two optimization iterations using the academic signoff timer with the long distance constraint. In different initial designs, the mean quality scores are from optimizing eight circuits of the three teams. Increasing the quality score from initial design to the first two iterations of postoptimization indicates that the proposed method contributes to improving timing in each iteration. However, cell movements are limited to displacement, so scores tend to increase monotonously and become saturated under the contest rules.

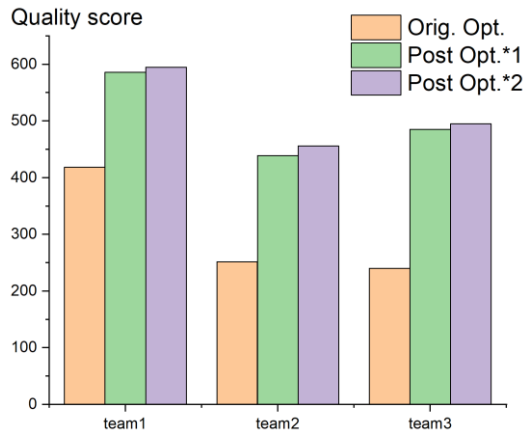

Fig. 12. Double postoptimization.

2) Comparison Results: In this section, one iteration of the postoptimization flow is applied to the initial placement solutions from the contest benchmark. Fig. 13 shows the quality scores obtained in this paper and in a previous work [9] under short distance constraints. The average quality scores [see (3)] do not differ significantly, being 311.38 and 311.16, respectively. Our slightly better results compared to [9] may be because we integrated more information from the signoff timer. This makes our estimation of timing better than theirs and also makes it easier to achieve a better solution for a short distance. In addition, under the same constraints, the optimized solutions of the top three teams can be further improved, and their postoptimized quality scores are 387.04, 350.785, and 369.33, respectively (see Fig. 14). Table V presents detailed information on the improvements. Each row records a quality score with the initial score in brackets. Before postoptimization, only five of 24 designs (17%) had a better score than the score in previous work [9]. After postoptimization, 20 of the 24 designs (83%) had a higher quality score. We believe that applying the designs of the previous study as our initial design, the timing quality can also be improved. In addition, [11] only demonstrated the results of late timing improvement. They improved WNS by 2.5% and TNS by 6.1% on average. In this paper, we improve WNS and TNS by 5.4% and 13.95%, respectively.

TABLE V
DETAILED INFORMATION OF POSTOPTIMIZATION PROCEDURE ON TOP3 TEAM VERSUS PREVIOUS WORK

Case	Super.1	Super.3	Super.4	Super.5	Super.7	Super.10	Super.16	Super.18	Average
Optimize First Place	538(450)	377(244)	378(288)	185(41)	158(99)	253(112)	769(529)	436(366)	368.75(266.13)
Optimize Second Place	465(234)	422(160)	251(80)	220(109)	132(43)	247(61)	685(370)	384(180)	350(154.625)
Optimize Third Place	473(411)	419(214)	304(179)	220(148)	150(70)	273(146)	733(386)	382(258)	369.25(226.5)
Previous work	391	351	276	149	141	142	735	302	310.875

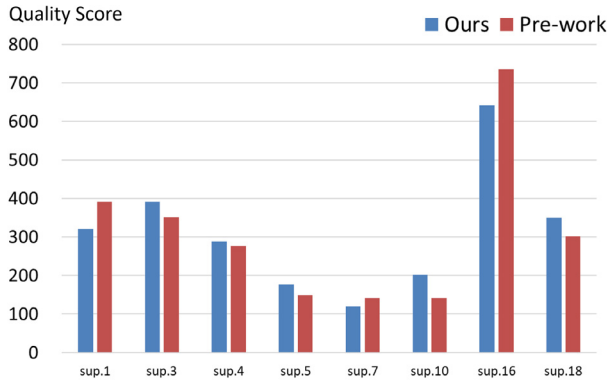

Fig. 13. Postoptimization for initial design.

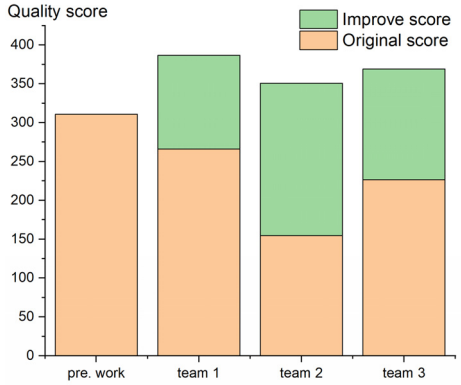

Fig. 14. Average quality score of previous work and postoptimization of top three teams.

	LR basic flow	Dynamic net weighting	Clock adj.
Improvement Percentage	47.1%	33.1%	19.8%

Fig. 15. Improvement percentage of LR weighting, dynamic net, and clock adjustment.

Fig. 15 plots the timing improvements from using LR weighting flow with linear regression timing model, dynamic net weighting, and clock adjustment. The percentage is an average over eight initial placement solutions. However, the performance of our algorithm is limited by the initial position of the cells. The difference of the individual timing improvement between these designs comes from the variation of the initial design, especially for the cell movement. Different initial designs favor different methodologies for reducing timing violations. In general, clock skew adjustment only solves slight timing violation and, in our experiments, does not contribute to large improvement percentage (19.8%). For cell movement-related techniques, the cell movement range for legalization impacts the degree of improvement of the benefit of LR-based flow and net weighting. Each cell has limited movement distance from its original design on these benchmarks. Our postoptimizer focuses on the more restricted situation because each initial layout design can be restricted in a relatively small space to move. Furthermore, in some cases, cells on critical paths are by default fixed and cannot be improved anymore by cell movement. The proposed methodology supports more flexible cell movement and, thus, more benefits in timing improvement, which results in significant improvements in most cases (average 80.1%). Reducing timing slack by cell movement depends on the available free or the swapped space. The accuracy of delay estimation using dynamic net weighting could be reduced after several iterations. Therefore, different techniques developed in this paper improve this percentage based on the initial placement.

V. CONCLUSION

This paper proposes approximation models for signoff timer information and nonlinear cell libraries, which can be used in TDP optimization. Routing information is considered in the Elmore model with dynamic weighting, and a multiregression method is applied to multiple-dimensional tables. This regression-based technique can be used to fit all kinds of table-based models in a cell library and can be integrated into the optimization flow. Adjustment of the clock tree and the modified weight of LR help to ensure that the timing violations of design are diminished. Our work integrates accurate timing information into the optimization of TDP and reduces the error of estimation during physical implementation, compared to the simulation of the signoff timer. This postoptimization algorithm was applied to the ICCAD15 benchmark. Implementation in 48 default designs from the top three teams significantly improved their timing. The postoptimization method proposed herein enables further timing improvements of the optimized designs with limited displacement constraints of cell movement by providing precise timer information. The goal of reducing the number of iterations between the signoff timer and physical implementation is, thus, achieved.

REFERENCES

[1] A. B. Kahng, "New game, new goal posts: A recent history of timing closure," in *Proc. 52nd ACM/EDAC/IEEE Design Autom. Conf.*, Jun. 2015, pp. 1–6.

[2] M. M. Ozdal, S. Burns, and J. Hu, "Algorithms for gate sizing and device parameter selection for high-performance designs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 10, pp. 1558–1571, Oct. 2012.
[3] C. Alpert, Z. Li, G.-J. Nam, C. N. Sze, N. Viswanathan, and S. I. Ward, "Placement: Hot or not," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2012, pp. 283–290.
[4] A. Chowdhary *et al.*, "How accurately can we model timing in a placement engine," in *Proc. 42nd Annu. Design Autom. Conf.*, Jun. 2005, pp. 801–806.
[5] T. Luo, D. Newmark, and D. Z. Pan, "A new LP based incremental timing driven placement for high performance designs," in *Proc. 43rd Annu. Design Autom. Conf.*, Jul. 2006, pp. 1115–1120.
[6] H. Ren, D. Z. Pan, C. J. Alpert, G.-J. Nam, and P. Villarrubia, "Hippocrates: First-do-no-harm detailed placement," in *Proc. Asia South Pacific Design Autom. Conf.*, Jan. 2007, pp. 141–146.
[7] C. Guth, V. Livramento, R. Netto, R. Fonseca, J. L. Güntzel, and L. Santos, "Timing-driven placement based on dynamic net-weighting for efficient slack histogram compression," in *Proc. Symp. Int. Symp. Phys. Design*, Mar. 2015, pp. 141–148.
[8] V. Livramento, C. Guth, R. Netto, J. L. Güntzel, and L. C. D. Santos, "Exploiting non-critical Steiner tree branches for post-placement timing optimization," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2015, pp. 528–535.
[9] G. Flach, M. Fogaça, J. Monteiro, M. Johann, and R. Reis, "Drive strength aware cell movement techniques for timing driven placement," in *Proc. Int. Symp. Phys. Design*, Apr. 2016, pp. 73–80.
[10] A. Bock, S. Held, N. Kämmerling, and U. Schorr, "Local search algorithms for timing-driven placement under arbitrary delay models," in *Proc. 52nd Annu. Design Autom. Conf.*, Jun. 2015, p. 29.
[11] J. Jung, G.-J. Nam, L. N. Reddy, I. H.-R. Jiang, and Y. Shin, "OWARU: Free space-aware timing-driven incremental placement with critical path smoothing," in *Proc. IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, Sep. 2017, pp. 1825–1838.
[12] M.-C. Kim, J. Hu, J. Li, and N. Viswanathan, "ICCAD-2015 CAD contest in incremental timing-driven placement and benchmark suite," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2015, pp. 921–926.
[13] R. Gupta, B. Tutuianu, and L. T. Pileggi, "The Elmore delay as a bound for RC trees with generalized input signals," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 16, no. 1, pp. 95–104, Jan. 1997.
[14] C. J. Alpert, F. Liu, C. V. Kashyap, and A. Devgan, "Closed-form delay and slew metrics made easy," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 23, no. 12, pp. 1661–1669, Dec. 2004.
[15] A. Dey, C. R. Rao, and H. Toutenburg, *Linear Models: Least Squares and Alternatives*. New York, NY, USA: Springer, 1997.
[16] W. C. Elmore, "The transient response of damped linear networks with particular regard to wideband amplifiers," *J. Appl. Phys.*, vol. 19, no. 1, pp. 55–63, Jan. 1948.
[17] C. V. Kashyap, C. J. Alpert, F. Liu, and A. Devgan, "Closed-form expressions for extending step delay and slew metrics to ramp inputs for RC trees," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 23, no. 4, pp. 509–516, Apr. 2004.
[18] J. Xu and M. H. Chowdhury, "Fast waveform estimation (FWE) for timing analysis," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 5, pp. 846–856, May 2011.
[19] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*, vol. 10. New York, NY, USA: Springer, 2001.
[20] G. Wu and C. Chu, "Two approaches for timing-driven placement by Lagrangian relaxation," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 12, pp. 2093–2105, Dec. 2017.
[21] V. Nawale and T. W. Chen, "Optimal useful clock skew scheduling in the presence of variations using robust ILP formulations," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2006, pp. 27–32.

Tai-Cheng Lee received the B.S. degree from the Department of Applied Mathematics, National Sun Yat-sen University, Kaohsiung, Taiwan, and the M.S. degree from the Institute of Statistics, National Tsing Hua University, Hsinchu, Taiwan. He is currently working toward the Ph.D. degree at the Department of Computer Science, National Chiao Tung University.

He has been a CAD engineer with semiconductor-related companies at Hsinchu, for several years. His current research interests include statistical learning, simulation data analysis, and optimization of physical design.

Yih-Lang Li (M'06) received the B.S. degree in nuclear engineering and the M.S. and Ph.D. degrees in computer science, majoring in designing and implementing a highly parallel cellular automata machine for fault simulation, from National Tsing Hua University, Hsinchu, Taiwan.

In 2003, he joined the Faculty of the Department of Computer Science, National Chiao Tung University (NCTU), Hsinchu, where he is currently a Professor. From 1995 to 1996 and from 1998 to 2003, he was a Software Engineer and an Associate Manager with Springsoft Corporation, Hsinchu, where he first completed the development of design rule checking (DRC) tool for the custom-based layout design and then established and led a routing team for developing a block-level shape-based router for the custom-based layout design. His current research interests include physical synthesis, parallel architecture, vehicle navigation, and deep learning.

Dr. Li joined the Technical Committee of the first CAD contest in Taiwan and served as a committee Member for ten years. He has also been serving as the Compensation Committee Member and the Independent Director of Board of Directors for AMICCOM Electronics Corporation since 2012. He was a recipient of the Japan Society for the Promotion of Science Faculty Invitation Fellowship. He was the Contest Chair of the first CAD contest at ICCAD in 2012 and the Technical Program Committee Member for ASPDAC and DAC.