Exploiting Setup–Hold-Time Interdependence in Static Timing Analysis

Emre Salman, *Student Member, IEEE*, Ali Dasdan, *Member, IEEE*, Feroze Taraporevala, *Member, IEEE*, Kayhan Küçükçakar, *Member, IEEE*, and Eby G. Friedman, *Fellow, IEEE*

***Abstract*—A methodology is proposed to exploit the interdependence between setup- and hold-time constraints in static timing analysis (STA). The methodology consists of two phases. The first phase includes the interdependent characterization of sequential cells, resulting in multiple constraint pairs. The second phase includes an efficient algorithm that exploits these multiple pairs in STA. The methodology improves accuracy by removing optimism and reducing unnecessary pessimism. Furthermore, the tradeoff between setup and hold times is exploited to significantly reduce timing violations in STA. These benefits are validated using industrial circuits and tools, exhibiting up to 53% reduction in the number of constraint violations as well as up to 48% reduction in the worst negative slack, which corresponds to a 15% decrease in the clock period.**

***Index Terms*—Constraint characterization, hold time, library characterization, setup–hold interdependence, setup time, static timing analysis (STA), timing constraint, timing violation.**

I. Introduction

THE CONTINUOUS scaling of complementary metal–oxide–semiconductor technology supports higher speed very large scale integration (VLSI) circuits. Operating frequencies of up to 1 GHz are common in modern deep-submicrometer application-specific integrated circuits. As the system clock period decreases, the pessimism imposed by timing verification tools becomes less acceptable. More accurate characterization and verification techniques are therefore highly desirable.

The timing verification of VLSI circuits is achieved by means of static timing analysis (STA) tools. The STA tools rely on data described in the cell libraries to analyze the circuit. The characterization of the individual cells in cell libraries is therefore highly critical in terms of the accuracy of the STA results [1]–[4]. Specifically, the setup- and hold-time constraints of the sequential cells are used to verify the timing of a synchronous circuit. Inaccurate characterization of timing constraints causes the STA results to be either highly optimistic or pessimistic. Both cases should be avoided as the optimistic case can cause a fabricated circuit to fail, whereas the pessimistic case unnecessarily degrades circuit speed.

The overoptimism or pessimism in STA is primarily due to the "independent" characterization of the timing constraints, although these constraints (including CLK-to-Q delay) are "interdependent." The constraints should therefore be characterized interdependently to remove optimism or pessimism in STA. In [5], a timing-constraint characterization that minimizes the sum of the CLK-to-Q delay and the setup time is proposed. In [6], the CLK-to-Q delay of a sequential cell is modeled, considering the dependence between the CLK-to-Q delay and the setup time. A 50–60-ps decrease in the clock period is shown if this dependence is considered during STA. These approaches, however, do not consider the interdependence between the setup time and the hold time. An approach for interdependent characterization is proposed in [7], and a solution that considers the dependence between the setup time, hold time, and CLK-to-Q delay is offered in [8] to determine the maximum operating frequency of a sequential cell. These approaches, however, do not exploit the interdependence in STA.

When the interdependence is considered during constraint characterization of a sequential cell, multiple valid constraint pairs, which are interchangeable, are obtained. These multiple pairs can be utilized in STA to significantly reduce timing violations and improve negative slack. Multiple constraint pairs, however, are currently not exploited.

A comprehensive methodology is proposed in this paper to rectify the weaknesses of the current approaches. The methodology consists of two phases. In the first phase, an interdependent characterization of the setup and hold times is described, resulting in multiple constraint pairs. In the second phase, an efficient algorithm with linear-time complexity is presented to integrate the interdependence into an STA tool. The algorithm exploits multiple constraint pairs by dynamically switching between pairs in order to remove violations.

Three main contributions are introduced in this paper, namely: 1) the existence and interchangeability of multiple constraint pairs; 2) the characterization of multiple pairs for cell libraries; and 3) a linear-time algorithm to exploit multiple pairs in STA. The methodology is validated on high-performance industrial circuits and an industrial sign-off STA tool. In particular, STA results demonstrate up to 53% reduction in the number of constraint violations as well as up to 48% reduction in the worst negative slack (WNS).

The rest of this paper is organized as follows: Relevant background material is provided in Section II. The problem formulation, the concept of interdependence, and current approaches

Manuscript received March 17, 2006; revised July 14, 2006. This paper was recommended by Associate Editor C. J. Alpert.

E. Salman and E. G. Friedman are with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627 USA (e-mail: salman@ece.rochester.edu; friedman@ece.rochester.edu).

A. Dasdan was with Synopsys, Inc., Mountain View, CA 94043 USA. He is now with Yahoo, Inc., Santa Clara, CA 95054 USA (e-mail: ali_dasdan@yahoo.com).

F. Taraporevala and K. Küçükçakar are with Synopsys, Inc., Mountain View, CA 94043 USA (e-mail: ferozet@synopsys.com; kayhan@synopsys.com).

Digital Object Identifier 10.1109/TCAD.2006.885834

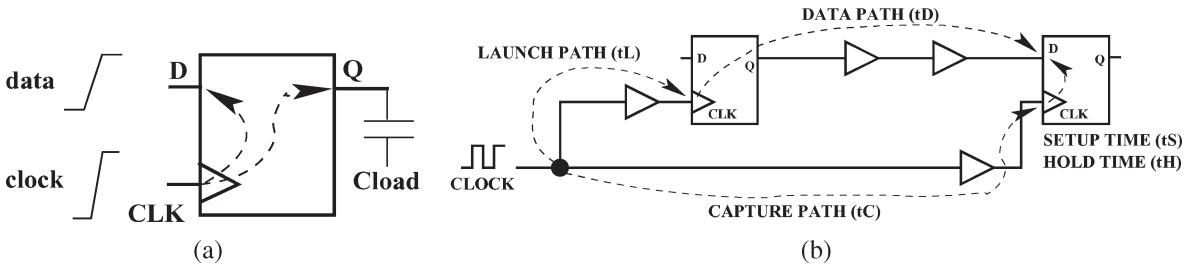

Fig. 1. Sequential cells, timing arcs, and timing paths. (a) Simplified sequential cell with two kinds of timing arcs. (b) Conceptual computation of SS and HS in STA on an example synchronous data path.

are discussed in Section III. The proposed characterization methodology and algorithm are described in Section IV. The STA results are presented in Section V. Finally, this paper is concluded in Section VI.

II. Preliminaries

Basic definitions and terminology about sequential circuits and industrial STA are reviewed in this section.

A. Sequential Circuits

A simplified sequential cell is illustrated in Fig. 1(a) and consists of a data input D, a clock input CLK, and an output Q. Examples of sequential circuits are flip flops and latches.

Two kinds of timing arcs of a sequential cell are relevant to this paper: 1) a constraining arc from the CLK input to the D input to annotate the setup and hold times and 2) a propagating arc from the CLK input to the Q output to annotate the CLK-to-Q delay. In STA, a timing path starts at CLK and ends at D. Hence, D and CLK are referred to as an endpoint and a startpoint, respectively.

B. STA

A basic STA tool reads in a circuit netlist, a cell library, and a clock period T. The tool reports whether the circuit performs as intended. This analysis is accomplished by computing the worst setup slack (SS) and worst hold slack (HS) at every endpoint. Referring to Fig. 1(b), these slacks are computed as follows:

$$\text{SS} = \min(tC + T) - \max(tL + tD + tS) \quad (1)$$

$$\text{HS} = \min(tL + tD) - \max(tC + tH) \quad (2)$$

where tC, tL, tD, tS, and tH refer to the capture path delay, launch path delay, data path delay, setup time, and hold time, respectively, as illustrated in Fig. 1(b).

If a slack is negative or nonnegative, it is said to be "violated" or "satisfied," respectively. If an SS is violated, the circuit can operate correctly by slowing the circuit down, i.e., by increasing T. If an HS is violated, the circuit will not function correctly.

C. Setup–Hold Times and Skews

Since nonnegative slacks are required to avoid violations, (1) and (2) can also be written, respectively, as follows:

$$\min(tC + T) - \max(tL + tD) \geq \max(tS) \quad (3)$$

$$\min(tL + tD) - \max(tC) \geq \max(tH). \quad (4)$$

These inequalities require a difference called a "skew" to be larger than or equal to a number called a "constraint." These inequalities, therefore, can be rewritten as

$$\min(\text{setup skew}) \geq \max(\text{setup time}) \quad (5)$$

$$\min(\text{hold skew}) \geq \max(\text{hold time}). \quad (6)$$

These skews and constraints are defined more formally in the following.

Definition 1: "Setup skew" refers to the amount of time a change in the data signal arrives at the D input of a sequential cell before the arrival of the latching edge of the clock signal.

Definition 2: "Hold skew" refers to the amount of time the data signal is stable at the D input of a sequential cell after the arrival of the latching edge of the clock signal.

Definition 3: "Setup time" refers to the setup skew necessary for the clock to reliably capture the data.

Definition 4: "Hold time" refers to the hold skew necessary for the clock to reliably store the data.

Definition 5: A "setup-time violation" refers to the situation where (3) or (5) is violated.

Definition 6: A "hold-time violation" refers to the situation where (4) or (6) is violated.

Note the important difference between skews and times: Setup and hold skews refer to "any" time difference between the data and clock signals, whereas the setup and hold times refer to the time difference required to "reliably" capture and store the data. According to (1) and (2), smaller setup and hold times are required to obtain larger SS and HS. It is therefore important to decrease the setup and hold times while reliably capturing and storing the data.

III. Problem Formulation

Previous work and the concept of interdependence are discussed in this section.

The numerical results of the rest of this paper are obtained through HSPICE simulations with BSIM4/BSIM3 models under typical corner conditions. The corresponding figures are prepared using Matlab. Input data for HSPICE are taken, without modification, from a 90-nm industrial cell library.

A. Existing Characterization Approaches

A common approach to characterize setup time is to examine the setup skew versus CLK-to-Q delay relationship [5], [9] at a fixed hold skew, which is called here the "counterpart skew."

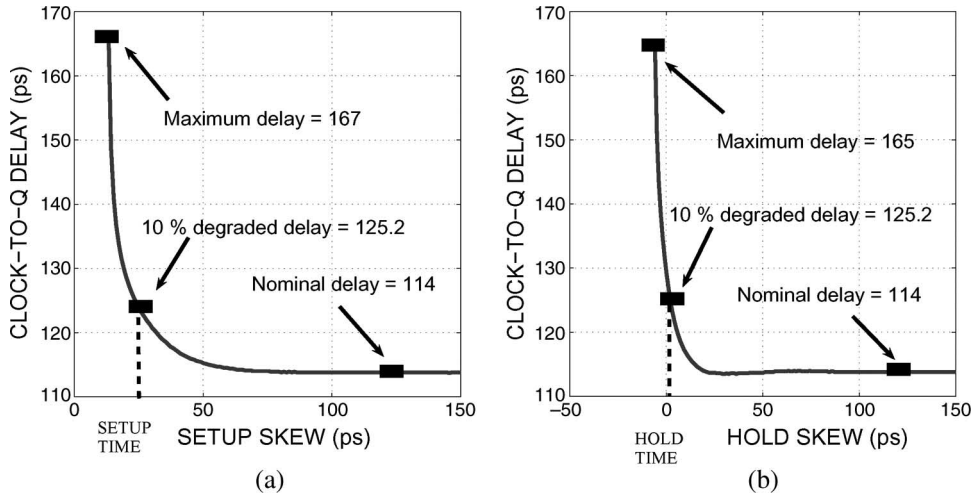

Fig. 2. Independent constraint characterization for sequential cells. (a) Setup skew versus CLK-to-Q delay for setup-time characterization. (b) Hold skew versus CLK-to-Q delay for hold-time characterization.

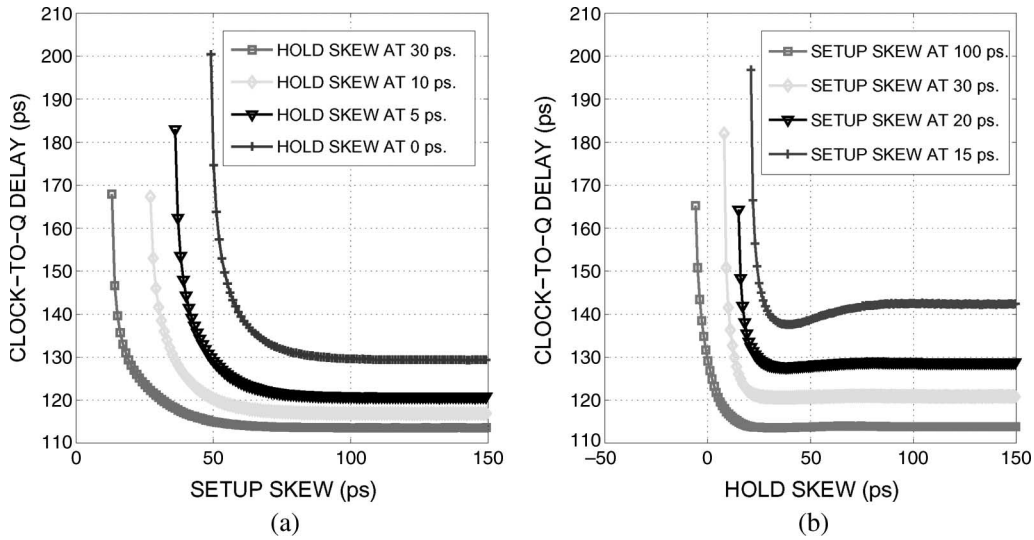

Fig. 3. Constraint characterization for sequential cells at different counterpart skews. (a) Setup skew versus CLK-to-Q delay at different hold skews. (b) Hold skew versus CLK-to-Q delay at different setup skews.

The process is similar for the hold time. These approaches are shown in Fig. 2. According to [5], three regions can be determined for both plots, namely: 1) "stable;" 2) "metastable;" and 3) "failure" regions. The stable region is defined as the region in which the CLK-to-Q delay is independent of the setup or hold skew. As the skew decreases, the CLK-to-Q delay starts to rise in an exponential fashion [10]. If the skew is excessively small, the sequential cell fails to latch the data. This region is called the failure region. The region between the stable and failure regions is referred to as the metastable region.

The setup and hold times cannot fall in the failure region since the sequential cell is unable to latch the data in that region. The setup (hold) time is usually set to the setup (hold) skew, where the stable region crosses over into the metastable region. There are different approaches to identify this "crossover point," as listed in [5]. In some approaches, the crossover point is the time where a certain amount of degradation in the CLK-to-Q delay occurs. For example, 10% degradation is assumed in this paper. In some other approaches, the crossover point is the time where the sum of the setup skew and CLK-to-Q delay is minimized.

B. Interdependence Between Setup and Hold Times

The setup and hold times are not independent [7], but rather these constraints are a function of the counterpart skews (hold skew for the setup time and setup skew for the hold time). These dependences are shown in Fig. 3. Note that the setup time decreases as the hold skew increases and that the hold time decreases as the setup skew increases. Thus, the smallest setup and hold times occur when the counterpart skews are the largest.

Existing characterization approaches typically ignore the interdependence of the setup and hold times. This strategy leads to two main issues.

Issue 1: Ignoring the interdependence of the setup and hold times results in either overoptimism or pessimism, depending on the assumption on the counterpart skew. If the counterpart

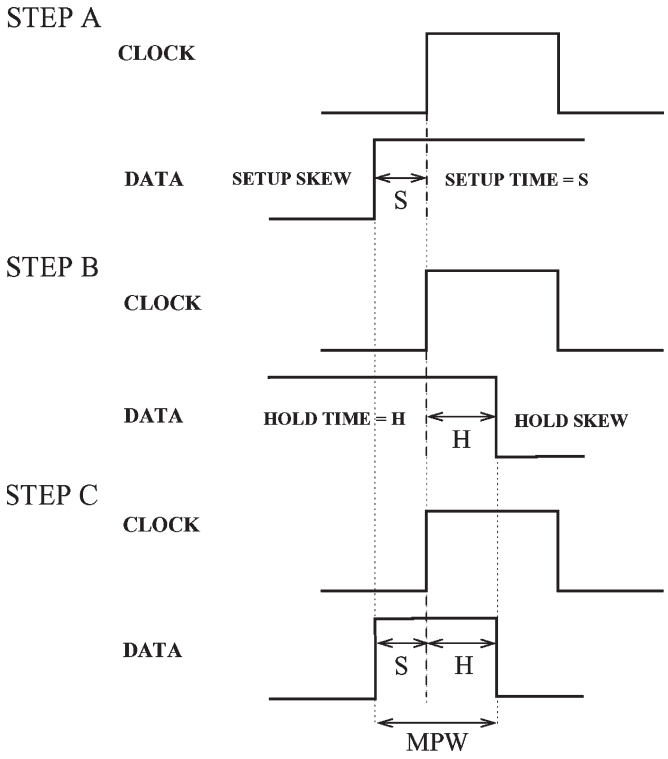

Fig. 4. Illustration of overoptimistic constraint characterization. Step A: setup-time characterization at a sufficiently large hold skew. Step B: hold-time characterization at a sufficiently large setup skew. Step C: MPW for which the sequential cell fails.

skews are assumed to be unnecessarily large, the resulting setup and hold times are optimistic. If, however, the data waveform does not satisfy large counterpart skews, the optimistic setup and hold times cause the circuit to fail, despite no violations on any of the individual constraints. This situation is illustrated in Fig. 4. In step A, the setup time is characterized as S at a sufficiently large hold skew. In step B, the hold time is characterized as H at a sufficiently large setup skew. In step C, the data waveform satisfies both of these constraints. The sequential cell, however, violates the minimum pulsewidth constraint due to the overoptimism introduced during the characterization steps of A and B.

Alternatively, if the counterpart skews are assumed to be unnecessarily small, the resulting setup and hold times are pessimistic. An example that illustrates pessimism is shown in Fig. 5. This example is proposed in [7] to overcome the overoptimism. The hold-time characterization is shown in Fig. 5(a). In step A, a temporary setup time, which is used in step B, is characterized at a sufficiently large hold skew. In step B, this temporary setup time is used as the counterpart skew to characterize the hold time. The hold time is therefore highly pessimistic. The process is similar for the setup-time characterization shown in Fig. 5(b), which produces a highly pessimistic result. Note that the temporary setup and hold times are used only to overcome overoptimism of the characterization process and do not reflect the final setup and hold times. Both of the overoptimistic and pessimistic cases should be avoided as the optimistic case can cause circuit failures after fabrication, whereas the pessimistic case can cause false violations during STA.

Issue 2: The interdependence between the setup time and hold time can be used to improve the timing analysis results of a circuit. Therefore, if this dependence is considered but not exploited in STA, the opportunity to reduce the number of timing violations and improve the slack is lost.

In [7], the first issue is resolved by considering this interdependence. However, "only one" interdependent pair of setup and hold times is considered. The interdependence is therefore not exploited to improve the slacks in STA. In this paper, a comprehensive methodology that solves both issues and offers reliable integration into an industrial STA tool is proposed, demonstrating up to 53% reduction in the number of constraint violations as well as up to 48% reduction in the WNS.

IV. Proposed Methodology

The proposed methodology is described here in five steps and also illustrated in the flowchart shown in Fig. 6. Each step is explained in a separate subsection. Steps 1)–4) describe the first phase of the methodology, which is related to the interdependent characterization of the timing constraints. Step 5) describes the second phase, which is related to efficient integration into an STA tool such that by using the proposed algorithm, the interdependence is exploited to remove timing violations.

Step 1) For each data and clock slew combination, the circuit is simulated to obtain CLK-to-Q delay surfaces, where each is a function of "independently varying" setup and hold skews.

Step 2) A contour, which is set to a constant CLK-to-Q delay, is obtained for each surface. Each point on the contour represents an interdependent pair of setup and hold times.

Step 3) Critical pairs are identified on each contour. A cell library is created for each of these pairs.

Step 4) At least two of the critical pairs are used to obtain a piecewise linear (PWL) curve. This curve approximates the contour with slightly pessimistic setup and hold times, allowing a potentially infinite number of pairs.

Step 5) A cell library is used for STA. If a violation occurs, the other libraries and the PWL curve are utilized to resolve the violation using the proposed algorithm. The process of violation resolution is therefore automated within STA.

Note that in Step 3), the creation of a cell library per critical pair is advocated only to simplify the exposition. In practice, a cell library per critical pair is not required, and the details are provided in Section IV-D.

The limitations of this methodology, e.g., library characterization time and STA runtime increase, are discussed in Section IV-F.

A. CLK-to-Q Delay Surface

For a given data and clock slew combination, the CLK-to-Q delay is a function of both the setup skew and the hold skew. A typical delay surface is shown in Fig. 7(a). Note that in this

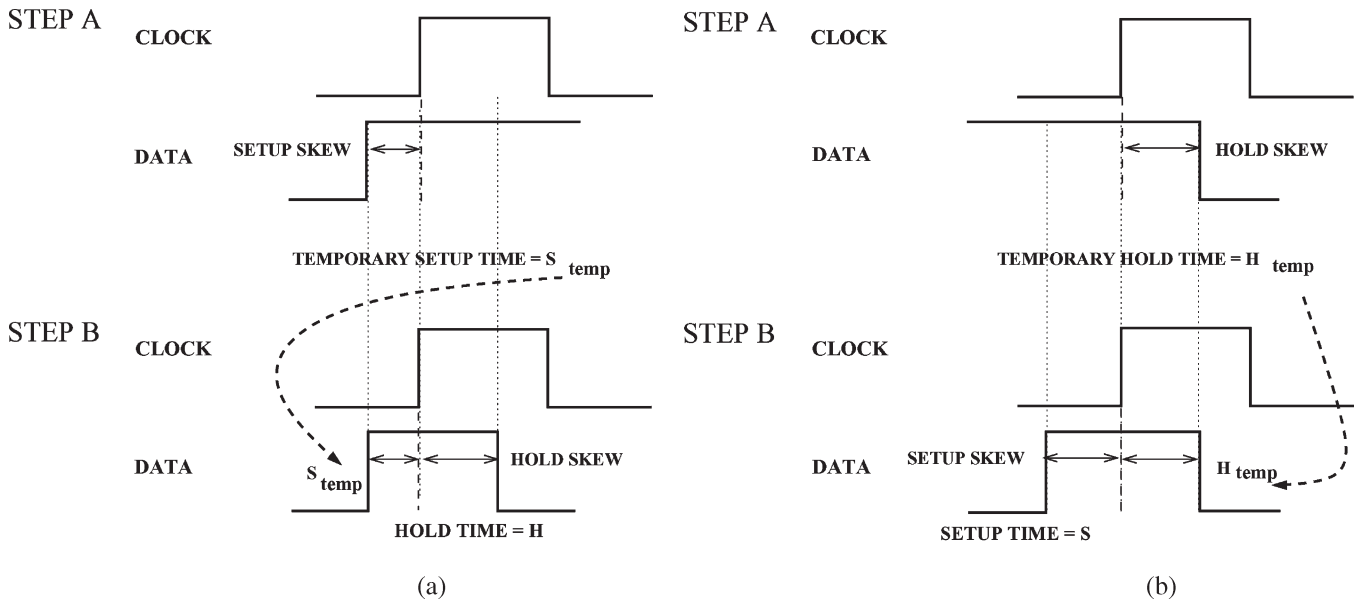

Fig. 5. Illustration of pessimistic constraint characterization. (a) Hold-time characterization. Step A: temporary setup-time characterization at a sufficiently large hold skew. Step B: hold-time characterization when the setup skew is equal to the temporary setup time found in step A. (b) Setup-time characterization. Step A: temporary hold-time characterization at a sufficiently large setup skew. Step B: setup-time characterization when the hold skew is equal to the temporary hold time found in step A.

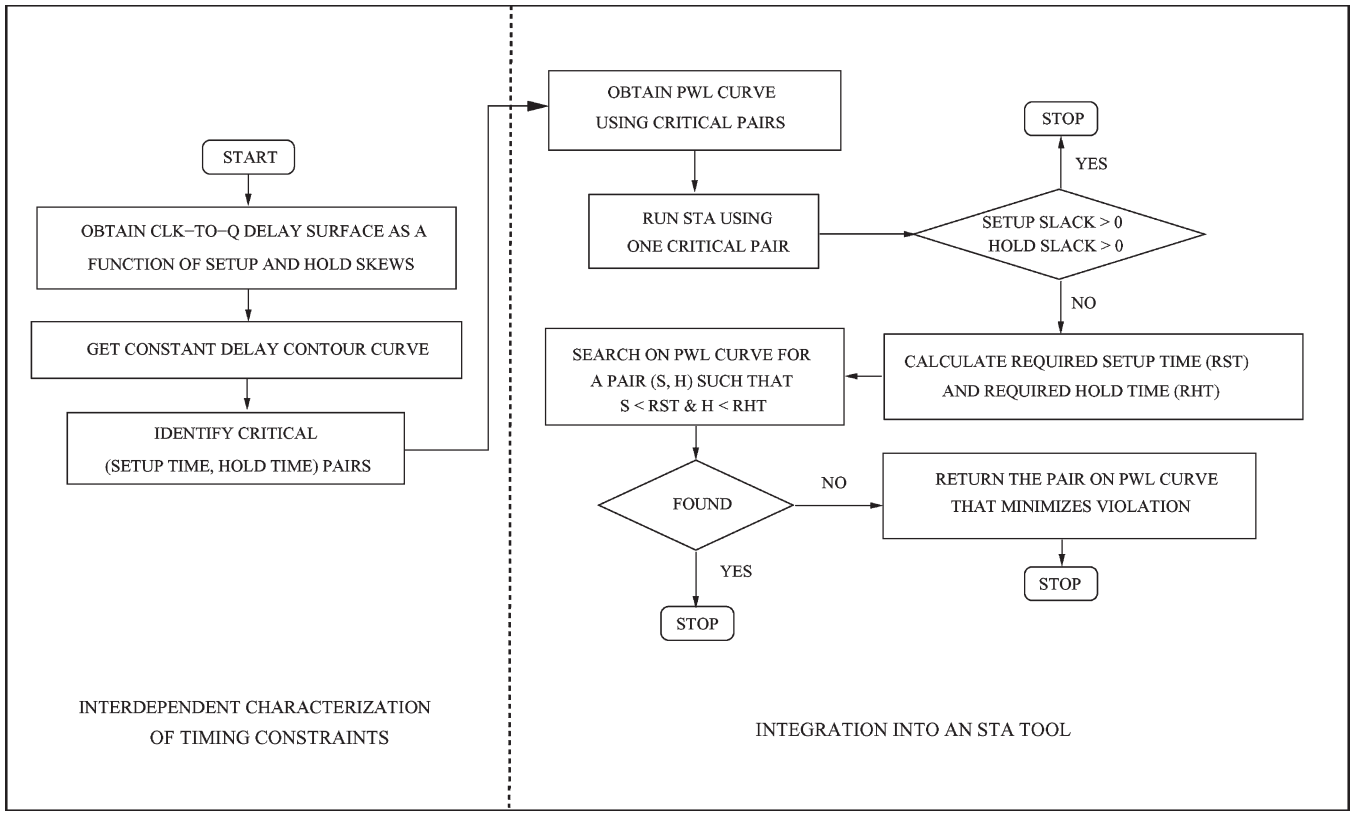

Fig. 6. Flowchart summarizing the proposed methodology including interdependent characterization of timing constraints and integration into an STA tool.

figure, the CLK-to-Q delay increases when the skews either independently or simultaneously decrease. The multiple peaks on the surface mark the boundary beyond, where the skews are excessively small and the sequential cell can no longer latch the data.

A delay surface is generated by independently varying the setup and hold skews. Independent variations allow the generation of the "actual" delay surface and simplify the library characterization process at the expense of additional circuit simulations.

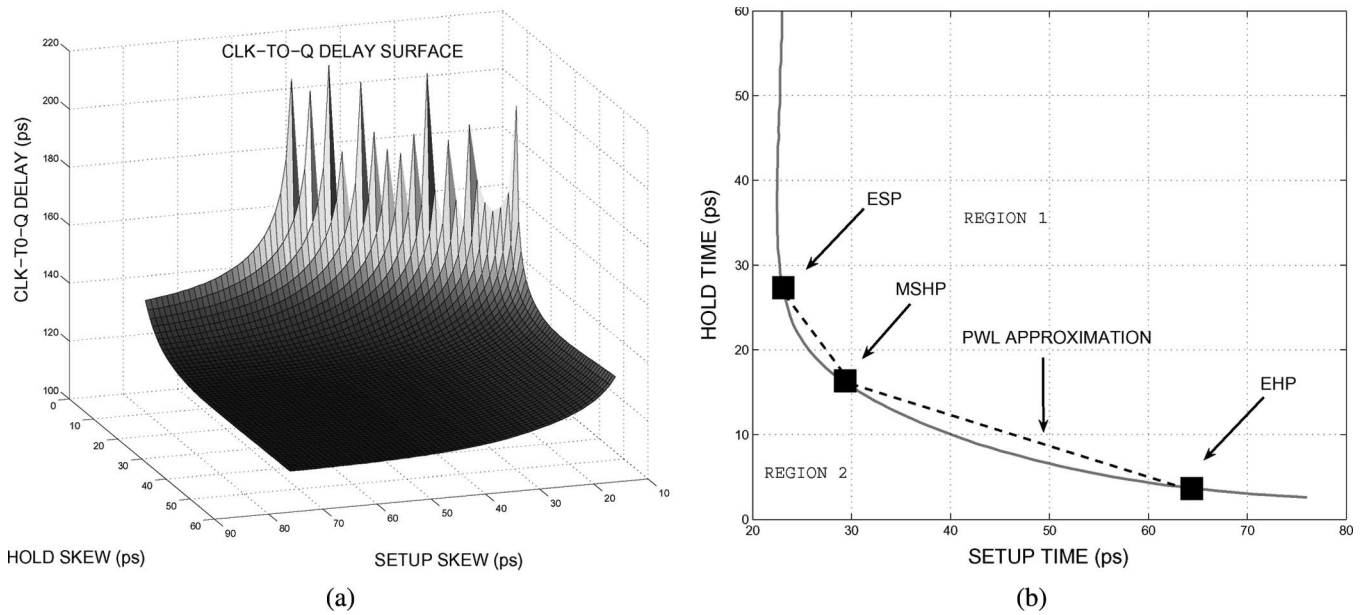

Fig. 7. Interdependent constraint characterization of a sequential cell. (a) CLK-to-Q delay surface as a function of independently varying setup skew and hold skew. (b) Contour at a 10%-degraded CLK-to-Q delay. The contour includes the critical pairs as well as a PWL approximation. Regions 1 and 2 are the pessimistic and optimistic regions, respectively.

Every point on the CLK-to-Q delay surface corresponds to a skew pair, which is denoted as (setup skew, hold skew). If a particular pair on this surface is identified as the final (setup time, hold time) pair, issue 1 in Section III-B is resolved because the setup and hold times at this point are now interdependent.

Different approaches exist to select a final pair on the surface, depending on the definition of the crossover point [7]. Irrespective of the approach used, it is highly likely that there will be multiple final pairs that satisfy this definition.

B. Constant Delay Contour

The definition of a common crossover point is a specific per cent degradation in the CLK-to-Q delay. Once the CLK-to-Q delay surface in three dimensions is obtained, all of the final pairs can be extracted from the constant delay contour as a per cent of the crossover point.

The contour obtained at a 10%-degraded CLK-to-Q delay is depicted in Fig. 7(b). Each (setup time, hold time) pair on this contour is interdependent and valid. Furthermore, any pair in region 1 is also valid with additional pessimism, whereas any pair in region 2 is invalid, as the pairs in this region are optimistic.

Two important conclusions can be drawn from this contour.

1) Rather than individual and independent setup and hold times, there are multiple and interdependent (setup time, hold time) pairs. Any pair can be chosen depending on the potential to remove timing violations.
2) As indicated in Fig. 7(b), the setup and hold times are inversely proportional, which can also be verified by least square regression analysis. Hence, a small setup time can be obtained at the expense of a large hold time (or vice versa).

C. Critical Pairs on Contour

The following pairs on the contour are defined as critical pairs because these pairs are appropriate candidates to include in a cell library. For each pair X, the notation $X = (s, h)$, where $s[X] = s$ and $h[X] = h$, is used.

Definition 7: P is defined as the set of all (s, h) pairs on the contour, where s is the setup time and h is the hold time.

Definition 8: S and H are defined as the set of all setup times s and hold times h on the contour, respectively.

Definition 9: The "minimum setup pair (MSP)" is defined as the pair (s, h) in P such that s is minimum in S. More formally,

$$\text{MSP} = (s, h) \in P \text{ such that } s = \min_{\forall s \in S}(s). \tag{7}$$

Definition 10: The "minimum hold pair (MHP)" is defined as the pair (s, h) in P such that h is minimum in H. More formally,

$$\text{MHP} = (s, h) \in P \text{ such that } h = \min_{\forall h \in H}(h). \tag{8}$$

The setup (hold) time of MHP (MSP) can be impractically large to minimize the corresponding hold (setup) time. If a slightly larger but bounded increase (controlled by a parameter ε) in hold (setup) time is allowed, the setup (hold) time of MHP (MSP) can be reduced to an acceptable level. This reduction is achieved by the effective hold (setup) pair.

Definition 11: The "effective setup pair (ESP)" is formally defined as

$$\text{ESP} = (s, h) \in P \text{ such that } s = s[\text{MSP}] + \varepsilon_s * |s[\text{MSP}]| \tag{9}$$

where ε_s is a user-controlled nonnegative parameter.

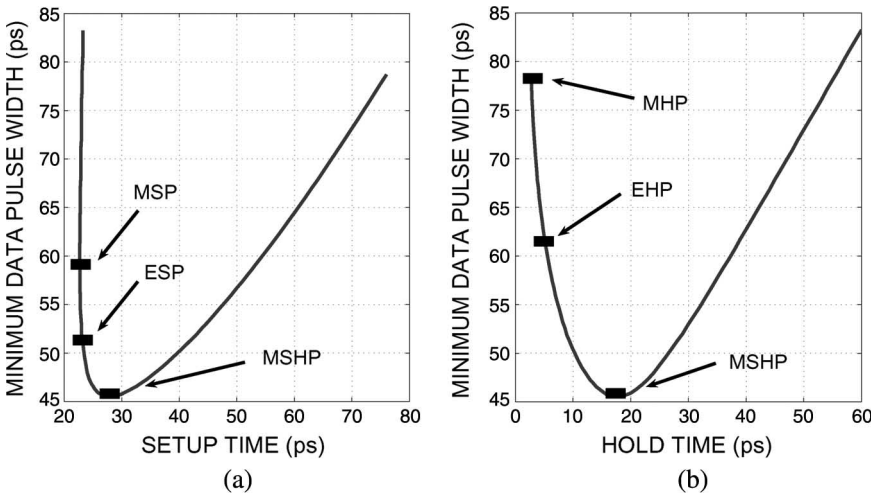

Fig. 8. Relationship between the constraints and the minimum pulsewidth. (a) Setup time versus minimum pulsewidth. (b) Hold time versus minimum pulsewidth.

Definition 12: The "effective hold pair (EHP)" is formally defined as

$$\text{EHP} = (s,h) \in P \text{ such that } h = h[\text{MHP}] + \varepsilon_h * |h[\text{MHP}]| \quad (10)$$

where ε_h is a user-controlled nonnegative parameter.

Definition 13: The "minimum setup–hold pair (MSHP)" is defined as the pair (s,h) in P such that the summation of s and h is the minimum. More formally,

$$\text{MSHP} = (s,h) \in P \text{ such that } s+h = \min\left\{\sum_{\forall (s,h)\in P} (s+h)\right\}. \quad (11)$$

Note that MSHP corresponds to the minimum data pulsewidth (MPW) possible that can be captured and stored by the sequential cell.

The distinction between the minimum and effective pairs can be illustrated by evaluating the minimum pulsewidth of the data signal. The minimum pulsewidth of the data signal is determined by summing the setup and hold times.

The variation of the minimum pulsewidth with respect to the setup and hold times is shown in Fig. 8. If the minimum constraints are used, rather than effective constraints, the minimum pulsewidth increases significantly. Note that at zero ε_s and ε_h, the effective constraints are equal to the minimum constraints.

D. PWL Approximation of Contour

In order to fully exploit this interdependence, the STA tool should use at least two (setup time, hold time) pairs on the contour. Since library characterization is expensive in time and memory, it may be impractical to generate more than two or three pairs. These pairs may, however, be insufficient to remove all violations. An improvement is to generate critical pairs and connect these critical pairs using a PWL curve to approximate the contour. For example, an approximation with two line segments, i.e., linear, can be obtained by connecting ESP and EHP, and an approximation with three line segments can be obtained by connecting ESP, MSHP, and EHP.

To avoid optimism in the PWL approximation, the contour should be convex with respect to the PWL approximation, i.e., as illustrated in Fig. 7(b), the line segments of the PWL approximation of the contour should remain in region 1.

The linear representation of the contour at three different data and clock slew pairs is shown in Fig. 9. The slews are computed with respect to 10% and 90% thresholds of the signal voltages. Each linear curve is obtained using ESP and EHP on the contour when $\varepsilon_s = 0$ and $\varepsilon_h = 0.2$. Note that the number of critical pairs used in the PWL curve represents a tradeoff between accuracy and complexity.

To represent critical pairs in cell libraries for sequential cells, the following is proposed. Current cell libraries generally contain two lookup tables for setup time and two tables for hold time (one table is defined for the rising edge data signal, and the other table is defined for the falling-edge data signal). Therefore, if two critical pairs are used, there are two possible options depending on the flexibility of the library format. If a slight modification is allowed on the library format, the proposed methodology does not require more tables: The existing tables for setup and hold times should be modified to contain interdependent (setup time, hold time) pairs instead of independent setup time and hold times. The library, therefore, still contains four tables, all of which consist of (setup time, hold time) pairs. If the current library format cannot be modified, the proposed methodology requires four additional tables: two tables for setup time and two tables for hold time to sufficiently represent (setup time, hold time) pairs. For both options, each table should be identified with the corresponding critical pair name. Note that if more than two critical pairs are used, the corresponding modifications can be reasoned similarly.

E. Integration of Interdependent Characterization Into STA

The FIND-BEST-PAIR algorithm shown in Fig. 10 is proposed to exploit the interdependence of the setup and hold times in STA.

1) FIND-BEST-PAIR Algorithm: FIND-BEST-PAIR reads in the PWL representation P of the contour as an input. This

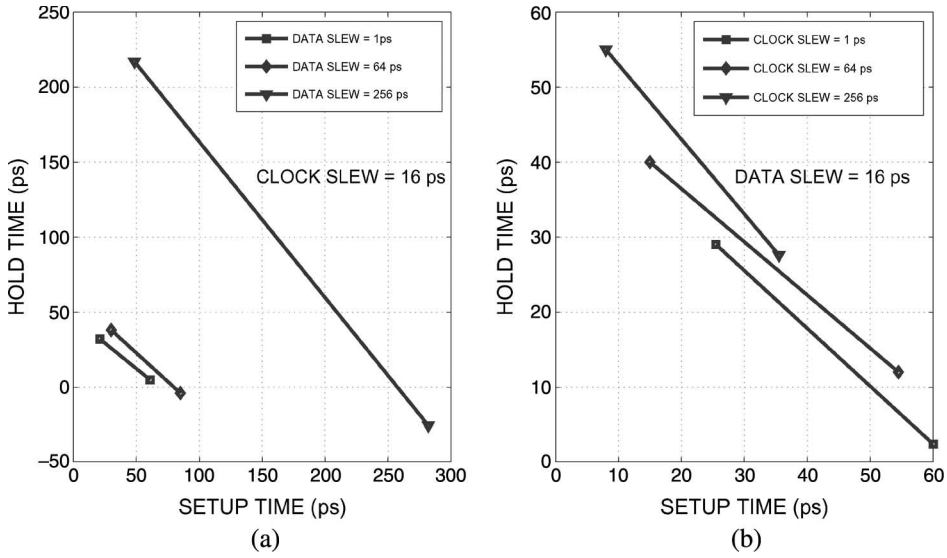

Fig. 9. Linear representations using two pairs, ESP and EHP, where $\varepsilon_s = 0$ and $\varepsilon_h = 0.2$. (a) At different data slews. (b) At different clock slews.

FIND-BEST-PAIR(Ordered pairs P)
1. Select a valid pair (s_0, h_0) from P
2. Calculate hold slack = HS (as in Section II-B)
3. Calculate setup slack = SS (as in Section II-B)
4. **if** ($HS \geq 0$ and $SS \geq 0$) **then**
5. **return** $<(s_0, h_0), \text{found}>$
6. Calculate maximum required setup time = $RST = s_0 + SS$
7. Calculate maximum required hold time = $RHT = h_0 + HS$
8. **for** each (s_i, h_i) in P where $i = 1, \cdots, (|P| - 1)$ **do**
9. **if** ($s_i \leq RST$) **then**
10. $h = \frac{(h_i - h_{i-1})}{(s_i - s_{i-1})} * (RST - s_{i-1}) + h_{i-1}$
11. **if** ($h \leq RHT$) **then**
12. **return** $<(RST, h), \text{found}>$
13. **return** $<(RST, RHT), \text{not found}>$

Fig. 10. FIND-BEST-PAIR algorithm to determine the (setup time, hold time) pair that removes the violation using the pairs P on the PWL approximation of the contour curve. This algorithm is run only for sequential cells with violations.

representation is obtained using critical (setup time, hold time) pairs, as described in Section IV-D. The critical pairs are sorted in descending order of setup times such that two successive pairs imply a line segment. The PWL representation P therefore contains a set of connected line segments.

At line 1 of FIND-BEST-PAIR, the (setup time, hold time) pair with the largest setup time is selected from the input. Note that this pair can be any of the critical pairs on P, but it is suggested here to use EHP, as hold times are typically more critical. The SS and HS are determined as described in Section II-B. Both slacks are checked for violations. If both are nonnegative, the algorithm terminates, returning the pair as the "best" pair. If one or both of the slacks are negative, these slacks are used to compute the "required setup time (RST)" and "required hold time (RHT)" to remove the violations. The loop at line 8 determines if such a pair actually exists in P. This line enables the tool to dynamically switch between interdependent (setup time, hold time) pairs to determine the best pair. If a pair that can remove the violations is found, that pair is returned as the best pair at line 12. If no such pair exists, the RST and RHT are returned at line 13 with a warning that no solution is possible. Note that the RST and RHT can be used to search for a pair that minimizes the violations.

The loop at line 8 iterates over each pair (s_i, h_i) in P (except the first pair since the first pair has already been checked before the loop) in order to determine if the line segment from (s_i, h_i) to (s_{i-1}, h_{i-1}) contains any pairs that can resolve both setup- and hold-time violations. The condition at line 9 determines if s_i is smaller than or equal to the RST. If this condition is satisfied, line 10 computes hold time h such that the pair (RST, h) is on the line segment from (s_{i-1}, h_{i-1}) to (s_i, h_i). The computation of hold time h is achieved using the equation of the line segment. If the condition at line 11 is also satisfied, the pair (RST, h) can resolve both violations. The algorithm terminates, returning this pair as the best pair at line 12. If line 12 is not reached during the iterations of the loop at line 8, the algorithm terminates with no solution, which is indicated by the returned value at line 13.

2) Examples: The behavior of the algorithm is illustrated for two different cases in Fig. 11. The situation for which FIND-BEST-PAIR determines a solution and removes the violations is illustrated in Fig. 11(a). The situation for which FIND-BEST-PAIR fails to determine a solution is illustrated in Fig. 11(b). For both figures, the original contour curve is represented by the function $h(s)$, which is the hold time as a function of the setup time. Note that for simplicity, only two pairs are used in both cases for the PWL approximation. Therefore, only one line segment, which is defined by the pairs (s_0, h_0) and (s_1, h_1), exists, as shown in Fig. 11.

The shaded regions in Fig. 11 represent all of the pairs whose setup time is smaller than or equal to the RST and whose hold time is smaller than or equal to the RHT. The pairs that are at the intersection of the shaded region and the PWL approximation can therefore resolve both violations. For Fig. 11(a), FIND-BEST-PAIR exits at line 12, returning pair B since pair B is at the intersection. For Fig. 11(b), FIND-BEST-PAIR exits at line 13, returning pair D because the PWL approximation does

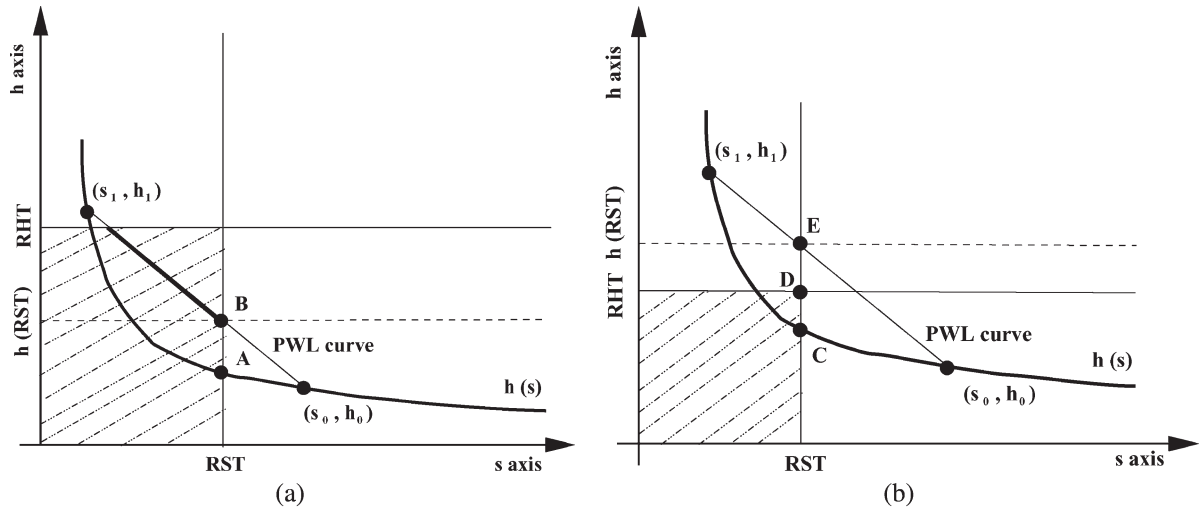

Fig. 11. Illustration of the cases for which FIND-BEST-PAIR (a) determines a solution and (b) cannot determine a solution. The actual contour curve is represented as $h(s)$, which is the hold time as a function of the setup time (RST, required setup time; RHT, required hold time). For (a), the algorithm returns point B as the best pair; the bold region of the PWL curve represents all of the pairs that can be returned. For (b), the algorithm returns point D because the PWL curve does not intersect with the shaded region, indicating that a solution is not possible.

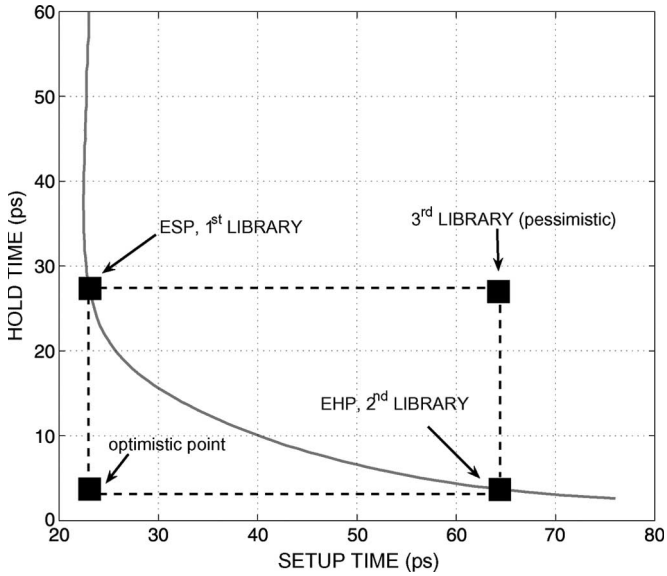

Fig. 12. Constant delay contour curve illustrating the characterization points of the three prototype libraries.

not intersect with the shaded region. In this case, therefore, no pair that can remove the setup- and hold-time violations exists.

Pairs A and C, which are shown in Fig. 11(a) and (b), respectively, illustrate the tradeoff between accuracy and complexity in terms of the number of pairs included in the PWL approximation. Pair A is on the actual contour in Fig. 11(a) and produces a larger HS value. Pair A, however, cannot be returned because it is not included in the PWL approximation. Similarly, for Fig. 11(b), pair C is also on the actual contour and can remove both violations since it is inside the shaded region. Pair C, however, cannot be returned because it is not included in the PWL approximation. A tradeoff therefore exists between accuracy and complexity in terms of the number of pairs included in the PWL approximation.

As shown in Fig. 11(a), FIND-BEST-PAIR returns the pair that minimizes the hold time. However, any pair that is at the intersection of the PWL curve and the shaded region [shown as the darker portion in Fig. 11(a)] can be returned. Note that adapting the algorithm to return any other valid pair is possible.

3) Complexity Analysis: In order to evaluate the time complexity of FIND-BEST-PAIR, the total number of sequential cells in the circuit is assumed to be N, which is a small fraction of the total number of cells in the circuit. FIND-BEST-PAIR executes for each sequential cell that violates a setup- or hold-time constraint. In the worst case, all of the N sequential cells have a timing violation. FIND-BEST-PAIR, therefore, executes N times in the worst case. For each execution, the algorithm requires time that is proportional to the number of pairs in input P because the loop at line 8 iterates P times and each iteration takes constant time. The time complexity of FIND-BEST-PAIR is therefore $O(|P|)$, and the complexity of resolving N violations using this algorithm is $O(N|P|)$. In practice, P is expected to be two or three pairs. FIND-BEST-PAIR therefore executes in constant time, and the N iterations of FIND-BEST-PAIR execute in $O(N)$ time. This time complexity is optimal since an optimal algorithm for violation resolution should execute in constant time per violation.

F. Limitations

The primary limitations of the proposed methodology are twofold: 1) The constraint characterization time increases for sequential cells. 2) The STA runtime increases if there is a timing violation. The first limitation is due to the generation of the delay surfaces, and the second limitation is due to the use of multiple constraints during STA.

The second limitation is not significant as the STA runtime increases only if there is a timing violation after analyzing the first (setup time, hold time) pair. Furthermore, if there is a violation, any other resolution method will also increase the overall

TABLE I
ABSOLUTE (ABS) AND RELATIVE (REL) IMPROVEMENTS OF TWO CIRCUITS WITH RESPECT TO LIBRARY 3. WNS IS THE WORST NEGATIVE SLACK, ΔWNS IS THE INCREASE IN WNS, AND ΔN IS THE DECREASE IN THE NUMBER OF VIOLATIONS

Circuit	Period (ps)	Library	WNS (ps)		Number of violations		ΔWNS (ps): abs (rel)		ΔN: abs (rel)	
			setup	hold	setup	hold	setup	hold	setup	hold
Circuit A	1500	3	-1003	-488	361	1868	-	-	-	-
		1	-790	-488	285	1868	213 (21.2%)	-	76 (21.0%)	-
		2	-1003	-307	361	1684	-	181 (37.1%)	-	184 (9.9%)
Circuit B	2500	3	-766	-26	1977	1	-	-	-	-
		1	-397	-26	924	1	369 (48.2%)	-	1053 (53.3%)	-
		2	-766	0	1977	0	-	26 (100.0%)	-	1 (100.0%)

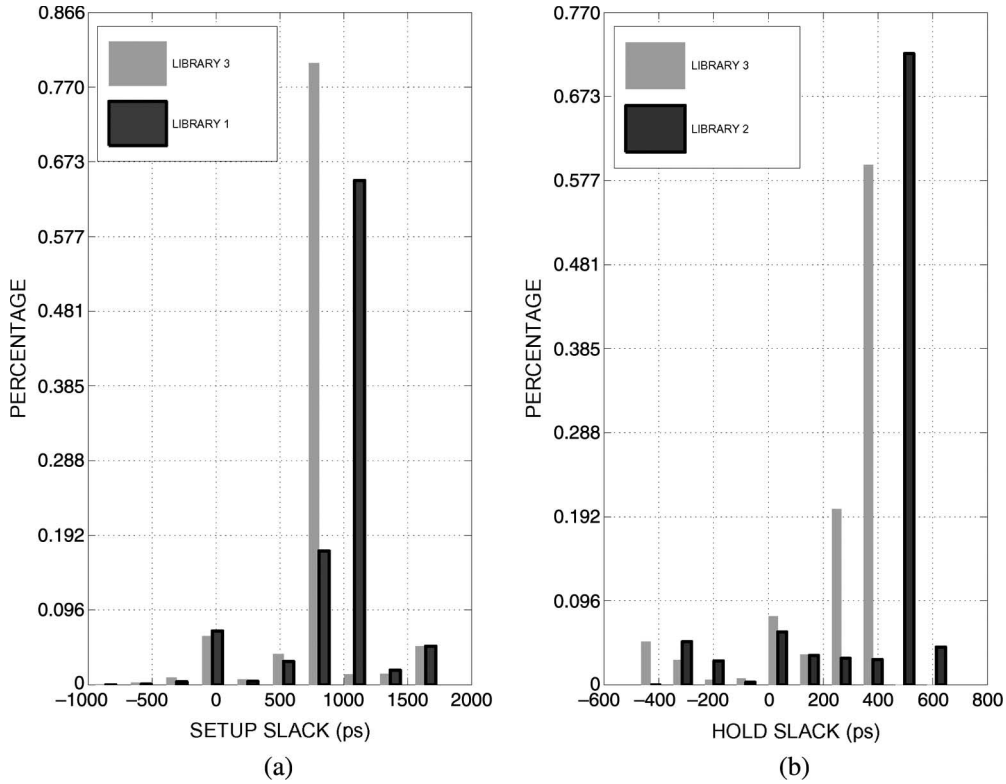

Fig. 13. Slack histograms for circuit A. (a) SS histograms of library 3 and library 1. (b) HS histograms of library 3 and library 2.

time. Since the current resolution methods are not automated, as proposed in this paper, the time for violation resolution may actually decrease with the proposed methodology. Note that the second limitation can be mitigated by using linear approximation.

In order to quantify the first limitation, assume that N_s and N_h denote the number of setup skews and hold skews to sweep over for characterization. For the overoptimistic characterization approach illustrated in Fig. 4, the total number of simulations to characterize both setup and hold times is equal to $N_s + N_h$. For the pessimistic characterization approach illustrated in Fig. 5, the total number of simulations doubles to $2(N_s + N_h)$ due to the additional steps. For the proposed characterization approach, the total number of simulations increases to $N_s N_h$ in the worst case to generate the entire CLK-to-Q delay surface, as illustrated in Fig. 7(a).

To reduce the number of simulations for the proposed approach, a simple heuristic is used to reduce the number of skews to be swept by eliminating those skew pairs that do not change the delay surface. For example, the location of the critical pairs on the contour shown in Fig. 7(b) indicate that it is not necessary to sweep the skew pairs falling in region 1.

V. STA RESULTS

A 90-nm library is used as a template to generate three new cell libraries: library 1, library 2, and library 3. As discussed in Section IV and IV-D, separate libraries are generated to simplify the evaluation of the methodology with an industrial STA environment. The sequential cells of each library are characterized using HSPICE with BSIM4/BSIM3 models under typical corner conditions.

The library characterization points for these three libraries are illustrated on a contour at 10%-degraded CLK-to-Q delay in Fig. 12. Both libraries 1 and 2 are on the contour: Library 1 is at ESP, and library 2 is at EHP. Library 3 is not on the contour; this last library uses setup times from EHP and hold times from ESP and, as such, is an example of independent and

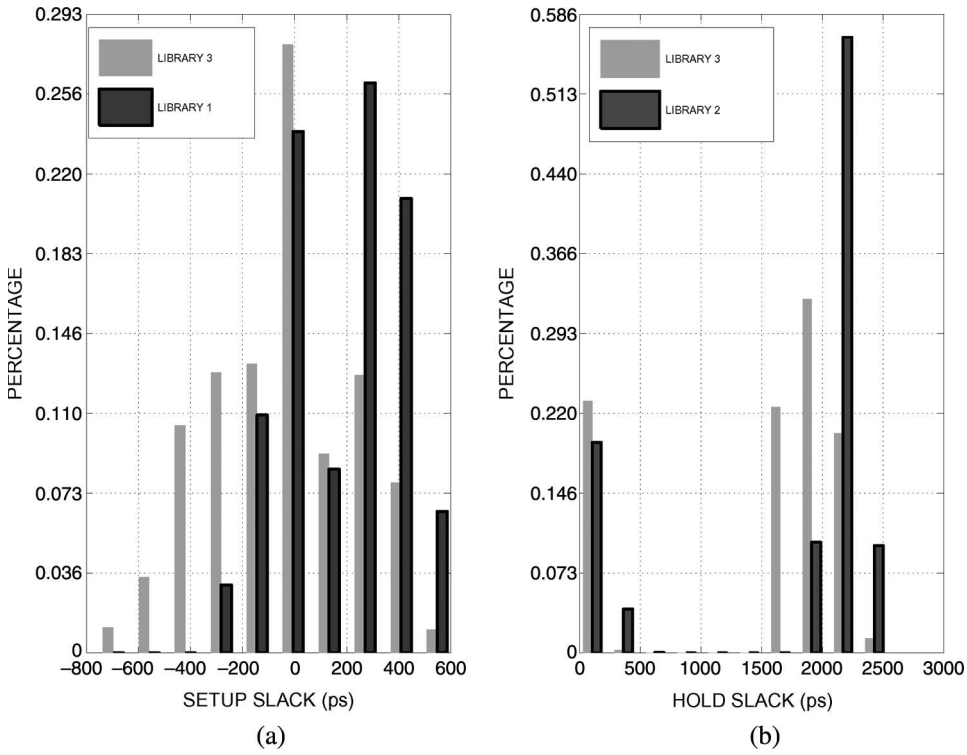

Fig. 14. Slack histograms for circuit B. (a) SS histograms of library 3 and library 1. (b) HS histograms of library 3 and library 2.

pessimistic characterization. The optimistic point that results from using relatively large counterpart skews is also shown in the figure. Note that the contour shown in Fig. 12 represents a single data-and-clock-slew combination of 25 ps each. For STA results, the same contour is obtained for each data-and-clock-slew combination that exists in the cell library. Different gains are obtained by using library 1 or library 2 over library 3, depending on these slew combinations.

An industrial sign-off STA tool, PrimeTime, is used to evaluate each prototype library on two industrial circuits: circuit A and circuit B. Both circuits are networking cores with nearly 20 000 cells. The clock frequencies of circuit A and circuit B are set to 666 and 400 MHz, respectively.

From STA, the smallest negative slack value, which is referred to as the WNS, and the number of violations are obtained for each of the endpoints. The STA results are listed in Table I. Each row corresponds to one simulation with one circuit and one library.

The WNS and the number of violations from library 3 are taken as a baseline, and the absolute and relative improvements are computed in the WNS and the number of violations with respect to library 3. Improvements in the WNS and the number of violations correspond to an increase in the WNS and a decrease in the number of violations, respectively. Note that library 1 illustrates improvements in the setup time without affecting the hold time and that library 2 illustrates improvements in the hold time without affecting the setup time. This result is because library 1 is characterized at ESP and library 2 is characterized at EHP, as shown in Fig. 12.

As listed in Table I, the improvement in the setup WNS is 369 ps (or 48.2%). This improvement corresponds to nearly 15% of the clock period. The improvement in the hold WNS is 181 ps (or 37.1%). In terms of the number of violations, the improvement in the setup case is 53.3%, and that in the hold case is 9.9%. Note that for hold-time improvements, the case represented by the last row, where only hold-time violation is removed, is ignored. The improvement in the setup WNS provides 14% and 15% increase in performance for circuits A and B, respectively. Furthermore, the improvement in the hold WNS reduces the required circuit modifications to remove the hold-time violations.

These improvements can also be illustrated by means of slack histograms over all the endpoints rather than a single number such as WNS. The histograms for the two circuits are shown in Figs. 13 and 14. For both histograms, there is a shift toward the positive side, indicating improvements in "almost all" of the slack values. The baseline is the slacks from library 3.

VI. Conclusion

A two-phase methodology is presented to exploit setup–hold-time interdependence in STA. The issues related with independent characterization, i.e., overoptimism and pessimism, are discussed and illustrated. Interdependent constraint characterization to remove these problems is proposed in the first phase of the methodology. In the second phase, an efficient algorithm is presented to integrate this interdependence into an STA tool such that multiple constraint pairs are exploited to reduce timing violations. The proposed algorithm automates the violation resolution process with linear-time complexity. The methodology is validated using industrial circuits and an industrial sign-off STA tool. The results show up to 53%

reduction in the number of constraint violations as well as up to 48% reduction in the WNS, corresponding to a 15% decrease in the clock period.

REFERENCES

[1] W. Roethig, "Library characterization and modeling for 130 nm and 90 nm SOC design," in *Proc. IEEE Int. SOC Conf.*, Sep. 2003, pp. 383–386.

[2] D. Patel, "CHARMS: Characterization and modeling system for accurate delay prediction of ASIC designs," in *Proc. IEEE Custom Integr. Circuits Conf.*, May 1990, pp. 9.5.1–9.5.6.

[3] R. W. Phelps, "Advanced library characterization for high-performance ASIC," in *Proc. IEEE Int. ASIC Conf.*, Sep. 1991, pp. 15-3.1–15-3.4.

[4] M. A. Cirit, "Characterizing a VLSI standard cell library," in *Proc. IEEE Custom Integr. Circuits Conf.*, May 1991, pp. 25-7.1–25-7.4.

[5] V. Stojanovic and V. G. Oklobdzija, "Comparative analysis of master-slave latches and flip-flops for high-performance and low-power systems," *IEEE J. Solid-State Circuits*, vol. 34, no. 4, pp. 536–548, Apr. 1999.

[6] A. M. Jain and D. Blaauw, "Modeling flip flop delay dependencies in timing analysis," presented at the ACM/IEEE Timing Issues (TAU) Workshop, Austin, TX, Feb. 2004.

[7] G. Rao and K. Howick, "Apparatus for optimized constraint characterization with degraded options and associated methods," U.S. Patent 6 584 598 B2, Jun. 24, 2003.

[8] E. K. Howick, "Conquering the high-frequency domain with predictable sequential models," in *Proc. Electr. and Phys. Des. Conf.*, Jan. 2002, CD-ROM.

[9] N. Weste and D. Harris, *CMOS VLSI Design*. Reading, MA: Addison-Wesley, 2004.

[10] M. Shoji, *Theory of CMOS Digital Circuits and Circuit Failures*. Princeton, NJ: Princeton Univ. Press, 1992.

Emre Salman (S'02) received the B.S. degree in microelectronics engineering from Sabanci University, Istanbul, Turkey, in 2004, and the M.S. degree in electrical and computer engineering from the University of Rochester, Rochester, NY, in 2006. He is currently working toward the Ph.D. degree at the University of Rochester.

In 2005, he interned at Synopsys Inc., Mountain View, CA, where he worked on pessimism reduction in static timing analysis and cell library characterization. In 2006, he was an Intern with Freescale Semiconductor Inc., Tempe, AZ, where he developed methodologies for the analysis and reduction of substrate noise coupling for large-scale mixed-signal circuits. His research interests include analysis, modeling, and design methodologies for high-performance digital and mixed-signal integrated circuits, focusing on signal integrity, noise coupling, global interconnects such as power and clock networks, and timing analysis.

Ali Dasdan (S'93–M'00) received the Ph.D. degree in computer science from the University of Illinois, Urbana-Champaign, in 1999.

From 1997 to 1999, he was a Visiting Scholar with the Center for Embedded Computer Systems, University of California, Irvine. From 1999 to 2003, he was a member of Technical Staff with the Advanced Technology Group, Synopsys, Inc., where he contributed to the first releases of SystemC library and SystemC Compiler, and to the ATG-SI test chip to study multithreshold voltage and signal integrity effects on 0.13-μm and below process technologies. From 2003 to 2005, he was a Technical Lead with the PrimeTime Static Timing Analysis Group, Synopsys, Inc., where he led the variation-aware/statistical PrimeTime tool (PrimeTime VX) from its conception to its beta release. He also made key contributions to variation-aware cell/net models and parameterized parasitics extraction. Since January 2006, he has been a Technical Lead for various projects at the Search Engine Group, Yahoo, Inc., Santa Clara, CA.

Feroze Taraporevala (S'86–M'88) received the B.E. degree in electrical engineering from Birla Institute of Technology and Science, Pilani, India, in 1985 and the M.S. degree in computer engineering from University of Texas, Austin, in 1988.

From 1988 to 2004, he was with LSI Logic Company, ViewLogic Company, and Monterey Design Systems. Since 2004, he has been with Synopsys, Inc., Mountain View, CA. His research interests include a variety of topics on CAD and VLSI.

Kayhan Küçükçakar (M'91) received the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, in 1987 and 1991, respectively.

He is currently the R&D Director of Synopsys, Inc., Mountain View, CA, and is responsible for PrimeTime and PrimeTime VX. Before joining PrimeTime in 1999, he was with Motorola for six years, managing an internal architectural synthesis tool development. He is the author of 19 papers and the holder of seven U.S. patents.

Dr. Küçükçakar has served on the technical program committees of the International Workshop on Timing Issues, the International Workshop on Hardware/Software Co-Design, the International Symposium on System Synthesis, Design, Automation and Test in Europe, and the European Design and Test Conference.

Eby G. Friedman (S'78–M'79–SM'90–F'00) received the B.S. degree from Lafayette College, Easton, PA, in 1979 and the M.S. and Ph.D. degrees from the University of California, Irvine, in 1981 and 1989, respectively, all in electrical engineering.

From 1979 to 1991, he was with Hughes Aircraft Company, rising to the position of Manager of the Signal Processing Design and Test Department, and was responsible for the design and test of high-performance digital and analog ICs. Since 1991, he has been with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY, where he is currently a Distinguished Professor, the Director of the High Performance VLSI/IC Design and Analysis Laboratory, and the Director of the Center for Electronic Imaging Systems. He is also a Visiting Professor at the Technion-Israel Institute of Technology, Haifa, Israel. He is the author of more than 300 papers and book chapters, and several patents, and the author and editor of eight books in the fields of high-speed and low-power CMOS design techniques, high-speed interconnect, and the theory and application of synchronous clock and power distribution networks. His current research and teaching interests include high-performance synchronous digital and mixed-signal microelectronic design and analysis, with application to high-speed portable processors and low-power wireless communications.

Dr. Friedman is the Regional Editor of the *Journal of Circuits, Systems and Computers*, a member of the Editorial Boards of the *Analog Integrated Circuits and Signal Processing, Microelectronics Journal*, the *Journal of Low Power Electronics*, and the *Journal of VLSI Signal Processing*, the Chair of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS steering committee, and a member of the technical program committee of a number of conferences. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, a member of the Editorial Board of the Proceedings of the IEEE and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II—ANALOG AND DIGITAL SIGNAL PROCESSING, a member of the Circuits and Systems Society Board of Governors, and the Program and Technical Chair of several IEEE conferences. He is a recipient of the University of Rochester Graduate Teaching Award and the College of Engineering Teaching Excellence Award. He is a Senior Fulbright Fellow.