Timing Yield-Aware Color Reassignment and Detailed Placement Perturbation for Bimodal CD Distribution in Double Patterning Lithography

Mohit Gupta, Kwangok Jeong, *Student Member, IEEE,* and Andrew B. Kahng, *Fellow, IEEE*

Abstract*—Double patterning lithography (DPL) is in current production for memory products, and is widely viewed as inevitable for logic products at the 32 nm node. DPL decomposes and prints the shapes of a critical-layer layout in two exposures. In traditional single-exposure lithography, adjacent identical layout features will have identical mean critical dimension (CD), and spatially correlated CD variations. However, with DPL, adjacent features can have distinct mean CDs, and uncorrelated CD variations. This introduces a new set of "bimodal" challenges for timing analysis and optimization. We assess the potential impact of bimodal CD distribution on timing analysis and guardbanding, and find that the traditional "unimodal" characterization and analysis framework may not be viable for DPL. We propose new bimodal-aware timing analysis and optimization methods to improve timing yield of standard-cell based designs that are manufactured using DPL. Our first contribution is a DPL-aware approach to timing modeling, based on detailed analysis of cell layouts. Our second contribution is an integer linear programming-based maximization of "alternate" mask coloring of instances in timing-critical paths, to minimize harmful covariance and performance variation. Third, we propose a dynamic programming-based detailed placement algorithm that solves mask coloring conflicts and can be used to ensure "double patterning correctness" after placement or even after detailed routing, while minimizing the displacement of timing-critical cells with manageable engineering change order (ECO) impact. With a 45 nm library and open-source design testcases, our timing-aware recoloring and placement optimization together achieve up to 271 ps (respectively, 55.75 ns) reduction in worst (respectively, total) negative slack, and 70% (respectively, 72%) reduction in worst (respectively, total) negative slack variation, respectively.

***Index Terms*—Bimodal critical dimension (CD) distribution, double patterning, placement perturbation, self-compensation.**

Manuscript received August 4, 2008; revised December 3, 2009. Date of current version July 21, 2010. A preliminary version this work appeared in [14], in which potential solutions for double patterning lithography bimodal-aware timing analysis and optimization were proposed. An extension beyond [14] was proposed in [9] which includes initial experimental results on timing-aware recoloring and placement perturbation. Extensions beyond [14] and [9] include an additional placement perturbation algorithm and comparison, discussions on applicable design stages, and experimental results on eight additional testcases. This paper was recommended by Associate Editor D. Z. Pan.

M. Gupta and K. Jeong are with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093-0404 USA (e-mail: mohit@vlsicad.ucsd.edu; kjeong@vlsicad.ucsd.edu).

A. B. Kahng is with the Departments of Computer Science and Engineering, and Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093-0404 USA (e-mail: abk@ucsd.edu).

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

Digital Object Identifier 10.1109/TCAD.2010.2049041

I. Introduction

DOUBLE patterning lithography (DPL) has been gaining attention as a relatively high throughput and low-cost lithography technique for deep submicrometer technologies. Next-generation patterning techniques, such as 157 nm, extreme ultraviolet, nanoimprint, e-beam direct write, etc., still face high cost, difficulty of materials and processes, or prohibitively low throughputs. Hence, double patterning with traditional lithography tools using 193 nm is regarded as a highly promising technique for 32 nm and 22 nm technologies.

DPL effectively doubles the achievable pattern density by using multiple exposure and etch steps or by using an additional spacer formation followed by multiple etch steps per layer. Double exposure [1], double patterning [2], [7], and sacrificial spacer double patterning [12] are three major types of double patterning in the International Technology Roadmap for Semiconductors [22], and have variants that enhance printability and reduce variability [3], [13].

To use double patterning, patterns in the same layer need to be decomposed into two groups, which are implemented independently [5], [17]. The pattern decomposition problem for double patterning is similar to a 2-colorability problem. Layout patterns are converted to a graph, in which a vertex represents an individual pattern and an edge represents a coloring conflict between vertices that are placed within a distance of the given technology resolution, and then assign colors to vertices, while minimizing the coloring conflicts. Routing methods considering double patterning coloring decomposition are discussed in [4].

Even with a layout decomposition and coloring solution that guarantees printability of a given design, other process variations, such as overlay or CD error, hamper easy adoption of double patterning. Overlay error can cause catastrophic open faults when a polygon pattern is split and its pieces assigned to different masks. Extensions for overlay error-tolerance are required at the interface between split patterns. Using a stitching method [20], incomplete connectivity problems can be avoided, but overlay-induced width and space variation increases on-chip variation. Notably, overlay error in double exposure/patterning with negative-tone resist for trench-first back-end of line (BEOL) process, or in spacer double patterning with positive-tone resist in spacer double patterning, results in large critical dimension (CD) errors [22]. For the most critical front-end-of-line (FEOL) layer, i.e., the

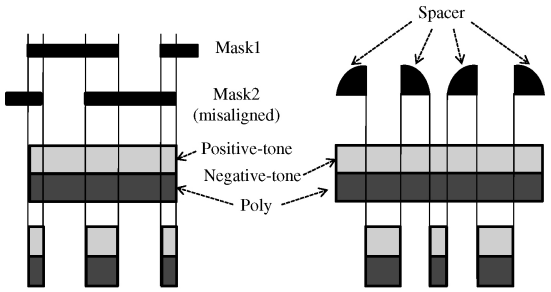

Fig. 1. Abbreviated DPL processes. CD variations caused by (a) overlay error and (b) spacer thickness variation. Figures do not convey exact DPL processes, but conceptually show CD changes from DPL processes.

poly layer, double exposure/patterning with positive-tone resist and spacer double patterning with negative-tone resist cause large CD error as shown in Fig. 1.

A number of previous works address the impact of overlay error in double patterning. Various sources of overlay error are discussed and an analytical capacitance model that comprehends overlay error is proposed in [21] and [8]. Nevertheless, overlay error in BEOL may not be a critical limiting factor for double patterning since the impact of overlay is comparable with other traditional variation sources [16]. However, in FEOL, CD variation is the most critical issue, hence we expect double patterning solutions that result in large CD variation due to overlay will not be deployable for the poly layer.

Excluding double patterning solutions causing large CD variation from overlay error, there still exist CD variations between the two poly lines that are printed in different steps of double patterning, due to the interference between successive exposures, nonuniform topography, imperfect etch biasing, etc. Dusa *et al.* [6] observe about 8% CD mean difference between two poly groups in a double patterning process, and those two CD populations are not correlated to each other.

In this paper, we assess the potential impact of bimodal CD distribution in DPL on timing analysis error guardbanding, as compared to the traditional "unimodal" characterization and analysis. To mitigate the impacts of bimodal CD distribution, we propose new bimodal-aware timing analysis and optimization methods to improve timing yield of standard-cell based designs that are manufactured using DPL.

The main contributions of our work are summarized as follows.

1) We give both analytic and empirical assessments of the potential impact of DPL on timing analysis error and guardbanding.
2) We investigate the mechanism of the delay variation due to bimodal CD distribution, and develop a new metric to represent the timing variation in double patterning.
3) We present a timing-aware optimal color assignment technique—alternate color assignment—using integer linear programming (ILP).
4) We implement a dynamic programming-based detailed placement to ensure the printability of DPL—double patterning lithography correctness (DPL-Corr)—utilizing whitespace management as in AF-Corr ([11]), after optimizing coloring using the alternate coloring assignment.

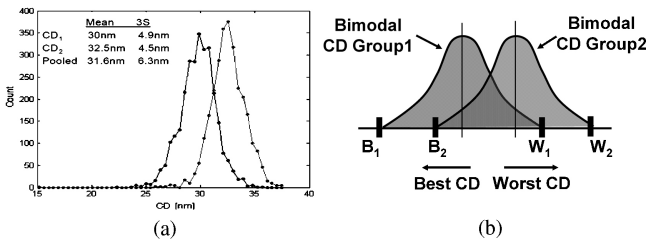

Fig. 2. Bimodal CD distribution for 32 nm technology measured from (a) 24 wafers processed by DPL, and (b) shown in a simplified illustration.

The remainder of this paper is organized as follows. In Section II, we describe the "bimodal challenge" from DPL. In Section III, we assess the impacts of bimodal CD distribution on timing delay, slack, and guardband, and suggest potential solutions to mitigate the impacts. In Section IV, we present an optimal alternate color assignment method for the timing-critical path optimization. In Section V, we describe our two dynamic programming (DP) algorithms for coloring conflict removal. Section VI discusses the experimental flow and results. We evaluate both alternate color assignment and DPL-Corr techniques separately and together. Finally, Section VII gives conclusions.

II. Bimodal Challenge

Fig. 2(a) shows a bimodal CD distribution for 32 nm technology measured from 24 wafers processed by DPL, as reported in [6]. Fig. 2(b) shows a simplified illustration of the bimodal CD distribution, in which two CD groups have independent mean and sigma values. The bimodal CD distribution affects design timing as follows.

A. Loss of Spatial Correlation

The existence of two independent CD populations in a design takes away the presumptions of spatial correlation that has always been used to reduce pessimism in corner-based timing analysis. For example, consider two closely placed, identical inverters made with different steps of double patterning DPL—i.e., one inverter is made by the first litho-etch step and the other is made by the second litho-etch step. These two inverters can have different gate CDs, so that their electrical characteristics, such as delay and power, can also be extremely different from each other despite the fact of adjacency in the same die.[1]

In general, within-die variations are taken into account by on-chip variation models or by statistical timing analysis flows. Bimodal CD distribution can also be treated as an additional variation source. However, the important problem that we address in this paper is that the size of the variation from the bimodal CD distribution can be very large, e.g., over 8% of CD mean difference between the groups, as shown in Fig. 2(a); therefore, designers must consider more extreme within-die

[1]Even though the fab will center the process to eliminate any nominal offset between the two exposures on average, any given chip will have a bimodal distribution. While it is unlikely that a technology averaged over many wafers/chips will show the behavior in Fig. 2, the lack of correlation between the two patterning steps will introduce bimodal behavior.

TABLE I
MEAN AND SIGMA OF BIMODAL AND POOLED UNIMODAL CD DISTRIBUTIONS

		G1		G2	
		Mean (nm)	3σ (nm)	Mean (nm)	3σ (nm)
Mean Diff.	Unimodal	50.00	2.00	–	–
0 nm	Pooled uni.	50.00	2.00	–	–
	Bimodal	50.00	2.00	50.00	2.00
1 nm	Pooled uni.	50.00	2.50	–	–
	Bimodal	49.50	2.00	50.50	2.00
2 nm	Pooled uni.	50.00	3.61	–	–
	Bimodal	49.00	2.00	51.00	2.00
3 nm	Pooled uni.	50.00	4.92	–	–
	Bimodal	48.50	2.00	51.50	2.00
4 nm	Pooled uni.	50.00	6.32	–	–
	Bimodal	48.00	2.00	52.00	2.00
5 nm	Pooled uni.	50.00	7.76	–	–
	Bimodal	47.50	2.00	52.50	2.00
6 nm	Pooled uni.	50.00	9.22	–	–
	Bimodal	47.00	2.00	53.00	2.00

variations during timing optimization as a direct consequence of DPL.

B. Increase of Overall CD Variation

Unless the two CD populations have the same mean values, overall CD variation must be increased with DPL. Dusa *et al.* propose the use of a unimodal representation pooled from the bimodal CD distribution [6], specifically

$$\left(3\sigma_{\mathrm{CD,pooled}}\right)^2 = \frac{\left(3\sigma_{\mathrm{CD,G1}}\right)^2}{2} + \frac{\left(3\sigma_{\mathrm{CD,G2}}\right)^2}{2} + \left(\frac{3}{2}\left(\mu_{\mathrm{CD,G1}} - \mu_{\mathrm{CD,G2}}\right)\right)^2 \quad (1)$$

where G1 and G2 are the two different groups of CD populations. Dusa *et al.* observed about 20% 3σ CD variation, relative to the mean CD, in their pooled CD variation model for a 32 nm DPL process. Table I shows, for various CD mean differences between G1 and G2, the CD mean and sigma values for the bimodal distribution, and for the corresponding unimodal distributions as calculated using (1) for 50 nm target CD.[2] As seen in the table, overall CD variation of the unimodal representation in column 4 increases with the increasing mean difference between CD groups.

III. IMPACTS OF BIMODAL CD DISTRIBUTION

In this section, we analyze the timing problems that arise from a bimodal CD distribution. In this discussion, we refer to the different CD distributions as corresponding to the different *colorings* (i.e., mask exposures) of the gate polys in a cell layout. In DPL coloring, adjacent minimum-pitch poly lines must be colored differently. Thus, a cell can have (at least) two basic versions according to its coloring sequence, as shown in Fig. 3. To distinguish between these different colorings, when the cell is instantiated in standard "North" orientation we use C_{12} (respectively, C_{21}) to refer to a cell in which the first or leftmost poly is colored by CD group 1 (respectively, CD group 2), the second poly is colored by CD group 2 (respectively, CD group 1), and so on.[3] We discuss the key impacts of the bimodal CD distribution: on path delay variation, on timing slack variation, and on the design guardband.

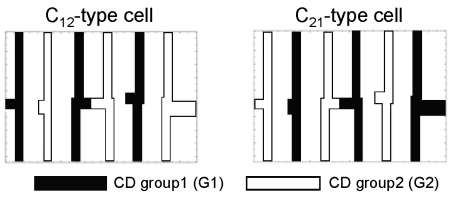

Fig. 3. Example of two different DPL colorings for a NOR3 cell.

A. Path Delay Variation in DPL

Every cell instance in a design can be colored differently according to its location and the surrounding cell instances. Therefore, instances of the same master cell in a timing path can be differently colored, and can have different electrical behaviors. As mentioned in Section II, due to the loss of the spatial correlation between differently colored cells, delays across cell types (C_{12} and C_{21}) in a path can vary randomly or with less correlation, even while cells of the same type coloring have strong correlation. Finding the path delay variation of a timing path in the presence of bimodal CD distribution requires solution of the following

Bimodal path delay variation analysis: Given m cells g_i of C_{12} type and n cells q_j of C_{21} type in a timing path, determine the delay variation of the timing path, subject to the constraints

(a) $Min_{i,j} cov(g_i, g_j) > Max_{i,j} cov(g_i, q_j)$

(b) $Min_{i,j} cov(q_i, q_j) > Max_{i,j} cov(g_i, q_j)$.

According to the constraints, the covariance between cells in the same group is larger than the covariance between cells in different groups.

The delay variation of a delay path is

$$\sigma^2(d(path)) = \sigma^2\left(\sum_i (d(g_i)) + \sum_j (d(q_j))\right) = \sum_i \sigma^2(d(g_i)) + \sum_j \sigma^2(d(q_j)) + 2\sum_{i_1,i_2} cov(g_{i_1}, g_{i_2}) + 2\sum_{j_1,j_2} cov(q_{j_1}, q_{j_2}) + 2\sum_{i,j} cov(g_i, q_j). \quad (2)$$

From (2), since $cov(g_i, q_j)$ is small (e.g., zero in the case of no correlation), the path delay variation for a path composed of uncorrelated different types of cells is smaller than that of a path composed of only correlated cells. Fig. 4 shows delay variations of a 16-stage inverter chain, normalized to mean values. Here, only four (out of 2^{16}) path colorings are studied: 1) M1-only; 2) M1-M2-M1-$\cdots$ alternation; 3) M2-M1-M2-$\cdots$ alternation; and 4) M2-only. The figure shows the

[2]As noted in the earlier review of double exposure DPL technology, since overlay control in the 45 nm node is 9 nm, it is difficult to use the negative double exposure process in light of the CD variation requirement. Hence, we do not consider the negative correlation between CD groups that would result with double exposure DPL, and we assume that CD variation is determined only by CD control capability.

[3]It is important to note that these at least two different colorings for the cell will exist regardless of whether a cell has an odd number of polys or an even number of polys, and regardless of cells' placement locations and orientations.

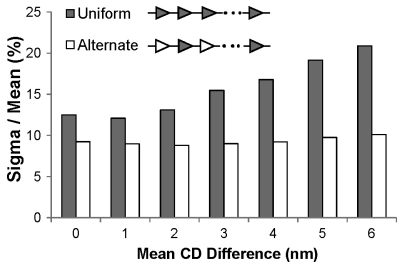

Fig. 4. Relative delay variation σ/μ (%) over all process corners.

average of 1 and 4 (dark gray), and average of 2 and 3 (clear). Correspondingly to the analytical solution in (2), alternate coloring of the timing path shows smaller delay variations.

B. Timing Slack Variation in DPL

While path delay variation can be reduced by the bimodal CD distribution, we find a very different situation with variation of timing slack—which is the most important parameter for design timing. Timing slack of the design is defined by

$$T_{\text{slack}} = T_{\text{clock}} + T_{\text{cycle}} - T_{\text{data}}. \tag{3}$$

The variation of the timing slack is calculated by

$$\sigma^2_{T_{\text{slack}}} = \sigma^2_{T_{\text{clock}}} + \sigma^2_{T_{\text{data}}} - 2cov\left(T_{\text{clock}}, T_{\text{data}}\right). \tag{4}$$

For a traditional single-exposure process, if we assume that spatial correlation is high, the covariance term in (4) will reduce the slack variation. However, in DPL, since cells in the clock path can be colored in a different way from cells in the data path, the covariance term will be reduced to zero, so that timing slack variation becomes a sum of clock path and data path variations. To meet signoff timing constraints with this increased slack variation in DPL, designs will require more stringent and difficult timing optimization.

To see more explicitly and realistically the impact of bimodal CD distribution on the timing slack, we extract a topmost critical path from the *AES* core, obtained as register-transfer level (RTL) from the open-source site http://opencores.org/ [25], which synthesizes to 40k instances, and is placed and routed with a reduced set of 45 nm library cells from the *Nangate* 45 nm *Open Cell Library* [24]. Each of the launching and capturing clock paths is composed of 14 stages of inverters. Also, the launching and capturing clock paths share the initial four stages of inverters, but differ from each other in the latter ten stages of each path. The data path is composed of 30 logic stages, e.g., 2-input NAND, NOR, OR and AND logic cells, and 1-input buffer (BUF) and inverter (INV) cells. An exhaustive design of experiments (DOE) would require $4 \cdot 2^{54}$ cases ($54 = 4 + 10 + 10 + 30$). We reduce the DOE complexity by restricting alternatives for the clock paths, the combinational data path, and registers.

First, we assume that the colorings of all cells in the data path are fixed. This allows us to evaluate the impact of bimodal CD distribution only on the clock design. Second, because the number of clock path configurations still remains very large ($4 \cdot 2^{24}$), we further limit our experiments to the five extreme cases shown in Table II.

TABLE II
COLORING CONFIGURATIONS OF THE CRITICAL PATH EXAMPLE

	Data path	Launching clock path	Capturing clock path
Case 1	$C_{12} \rightarrow C_{12} \cdots$	$C_{12} \rightarrow C_{12} \rightarrow \cdots$	$C_{12} \rightarrow C_{12} \rightarrow \cdots$
Case 2		$C_{21} \rightarrow C_{21} \rightarrow \cdots$	$C_{21} \rightarrow C_{21} \rightarrow \cdots$
Case 3		$C_{12} \rightarrow C_{12} \rightarrow \cdots$	$C_{21} \rightarrow C_{21} \rightarrow \cdots$
Case 4		$C_{21} \rightarrow C_{21} \rightarrow \cdots$	$C_{12} \rightarrow C_{12} \rightarrow \cdots$
Case 5		$C_{12} \rightarrow C_{21} \rightarrow \cdots$	$C_{12} \rightarrow C_{21} \rightarrow \cdots$

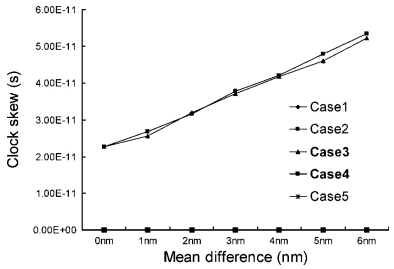

Fig. 5. Clock skew versus CD mean difference between CD groups, across combinations of process corners. Note that Cases 1, 2, and 5 are superposed on the x-axis.

For a design to operate correctly, data signals must be carried from one (launching) register to the next (capturing) register once per each clock cycle. The difference of delays between launching and capturing clock paths, i.e., *clock skew*, plays an important role in both the setup and hold time slacks. Fig. 5 shows the maximum skew that occurs as a result of the bimodal CD distribution, across the path coloring sequences shown in Table II. Note that the clock skew is originally designed to be zero. Intuitively, we can expect that there is no clock skew when the coloring sequences of both clock paths are the same, i.e., Cases 1, 2, and 5. However, even when the mean difference between two CD groups is zero, Cases 3 and 4 show substantial clock skew due to the different coloring sequences of launching and capturing clock paths, and the skew increases when the CD mean difference increases. The maximum clock skews of Cases 3 and 4 with 0 nm CD mean difference are 22.7 ps for each, and these skews increase up to 52.2 ps and 53.4 ps, respectively, with 6 nm CD mean difference.

Fig. 6 shows the setup time slack changes of each coloring sequence of clock paths versus the mean difference of the CD groups at the worst CD corner combination (*MAX–MAX*). The timing path originally has zero slack when the CD mean difference is zero (i.e., two color groups have same CD mean). For Case 4, since the delay of the capturing (respectively, launching) clock path decreases (increases), the setup time slack becomes negative;[4] this will worsen when the number of stages of the clock network increases. For Cases 1–3, and 5, delay of the capturing clock path is greater than that of the launching clock path, so that the setup time slack is still positive or even improved. We also note that the pooled unimodal CD representation shows unnecessarily pessimistic setup time slack values.

C. Guardband and Design Process in DPL

The simplest way to consider the bimodal CD distribution in the design process is to model bimodal as unimodal.

[4]With 6 nm CD mean difference, −18 ps of slack violation occurs. This value is about 10% of the clock path delay of our test case.

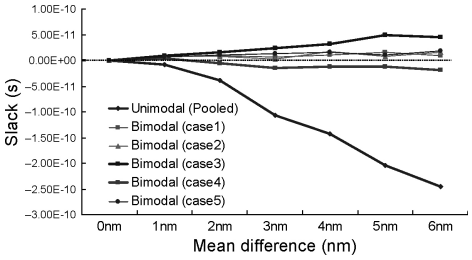

Fig. 6. Setup time slack versus CD mean difference between CD groups across combinations of process corners.

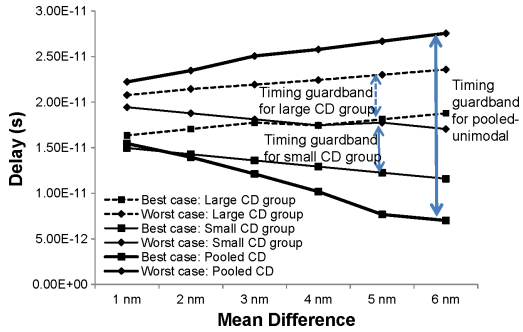

Fig. 7. Timing guardband for each characterization method.

The already-cited pooled unimodal CD model from [6] can be useful, and today's conventional flow can still be used. However, the pooled unimodal model gives a too-pessimistic guardband, which can lead to significant overdesign.

Fig. 7 shows the best-case and worst-case delays of the 45 nm INV cell for each of pooled unimodal and bimodal, with mean difference between the two CD groups on the x-axis. Delay difference between worst and best shows the size of the guardband. As seen in the figure, simple unimodal modeling will lead to more than 2× increase of guardband, even for the small mean difference cases; according to the recent study on guardband impact in [15], this will lead to over 15%, 39%, and 14% of area, runtime and wirelength increase, respectively.

From the above results and discussion, we can conclude that a pooled unimodal representation with pessimistic corner values will not suffice in the future of DPL, and furthermore, as we demonstrated above, the pooled unimodal model cannot capture the potential timing problems caused by uncorrelated data and clock delay variations.

D. Implications for Bimodal-Aware Timing Analysis and Optimization

Our results strongly suggest that to incorporate DPL into production, the bimodal CD distribution must be dealt with accurately in both analysis and optimization. In this section, we briefly outline courses of action by which the industry can respond to uncorrelated distributions of device behaviors in the DPL context.

1) *Timing Analysis:* DPL requires (1) more guardbanding and/or (2) a new methodology to characterize electrical properties such as delay, power, etc. of DPL-manufactured circuits. Existing methodology and infrastructure allows modeling via a pooled unimodal CD model, but our timing analyses in Sections III-A–III-C show that the pooled unimodal description is likely too pessimistic.

The device parameters of the process must capture the reality of bimodal CD variation, and production methodology should permit each printed transistor (finger) to independently reference the appropriate model card.

Given bimodal-aware SPICE modeling, bimodal-aware timing libraries can be generated. To begin with, poly patterns in a cell are decomposed into two groups. Pattern decomposition in double patterning can be performed at *design-level* or at *cell-level*. We use the term "coloring" when referring to the pattern decomposition and assignment of patterns to masks.

- *Design-level coloring* is "flat": the color of each pattern is decided considering colors of the surrounding patterns, so that a master cell can be colored in a number of ways according to the different surrounding colors and distances from them.
- *Cell-level coloring*, on the other hand, is "hierarchical": there is an optimal pattern decomposition for a given master cell, so that each master can have only two differently-colored possible instantiations in silicon (C_{12} type and C_{21} type), as illustrated by Fig. 3.

Each colored version can have multiple timing models according to the CD values of each poly group. Each worst-/best-case is split into two different scenarios. For the worst-case timing analysis, both $MAX_{G1} \geq MAX_{G2}$ and $MAX_{G1} \leq MAX_{G2}$ must be considered, since we cannot presume which poly group will have larger or smaller gate length than the other.

For accurate timing analysis, each differently-colored master cell must be characterized using SPICE simulations, and each cell instance must refer to the timing model defined for each colored version and each CD value. Hence, the number of worst-case timing models for a given master cell will be the number of different colorings multiplied by two CD scenarios. This explosion of library complexity in design-level coloring increases the complexity of sizing optimization, which is essentially a problem of selection from within a predefined cell library. Hence, to maintain tractable complexity, we assume cell-based coloring in the rest of this paper.

2) *Timing Optimization:* For bimodal-aware timing optimization, we can split a timing graph into three pieces: data path, clock path, and sequential cells. As is implied by the $cov(g_i, q_i)$ term in (2) and our experiments in Section III (e.g., Fig. 4), the alternate coloring reduces delay variation as well as the worst-case delay of both clock and data paths. Thus, we suggest the use of "self-compensation" proposed by Gupta *et al.* [10]—in the sense of deliberate balancing of cell colorings in timing paths—to reduce delay variation. For clock paths, we suggest the use of the same type coloring for all cells in the clock network (thus exploiting spatial correlation maximally). This will further reduce clock variation as well as slack variation. According to the experiment in Fig. 5, several tens of ps skew reduction will result. (*Note:* A "methodological" approach to bimodality mitigation for clock skew is based on improved cell library design; see footnote 7.)

However, we note that coloring based on the timing, i.e., alternate coloring or same type coloring of paths, can increase coloring conflicts with neighboring cells that are already colored [Fig. 8(b)]. To avoid DPL coloring conflicts between

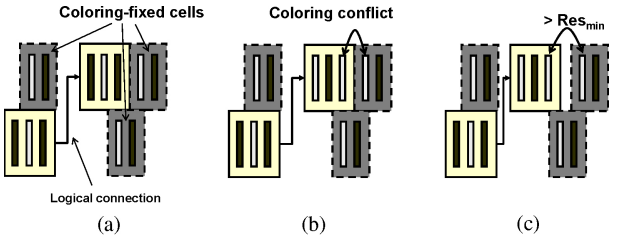

Fig. 8. Conflict removal with intelligent whitespace management. (a) Find the timing-critical path. (b) Determine best coloring. (c) Remove coloring conflicts with whitespace management.

adjacent cell instances, it may be necessary to develop larger-sized cells, in which all critical features can be colored independent of the colorings of other neighboring cells. To avoid increase in cell area, we suggest fixing coloring conflicts by placement perturbation using remaining whitespace, i.e., increasing distance between conflicting cells, as shown in Fig. 8.

For sequential cells, e.g., latches and registers, it is unclear which process combination gives the worst or best behavior. Measurement and optimization of delay and hazard timing margin across process corners appears to be an open and challenging research topic.

IV. Alternate Coloring of Timing Critical Paths

Given the implications discussed in Section III-D, we now propose a methodology for optimal coloring of timing paths.

A. New Metric: Coloring Sequence Cost

We begin by quantifying "balance" in the coloring of timing paths. The impact of DPL's bimodal CD distribution on cell delay varies according to the number of poly lines in a cell, the topology of the circuit, assigned color for each poly gate, and the specific transistors that are activated during signal transitions. A path consisting of only buffers (each buffer comprising two cascaded inverters) experiences small impact from the bimodal CD distribution, since the CD change of the first inverter can be compensated by that of the second inverter. On the other hand, a path consisting of only one-stage inverters can have two different worst-case delay values when all inverters are assigned the same color: the inverters will have either all positive CD changes or all negative CD changes, so that there is no compensation.

For cells that are more complex than an inverter or buffer, the impact on delay of bimodal CD distribution is complicated. Table III shows SPICE simulation results on NAND2 shown in Fig. 9(a), according to transitive inputs (A1 and A2), switching direction (rise and fall), and bimodal CD variation. Comparing the second and fifth rows, we observe that the CD of A2 has negligible impact on the rise delay due to the transition of A1 ($rise_{\mathrm{A1}}$). Similarly, the CD of A1 does not affect rise delay due to the transition of A2 ($rise_{\mathrm{A2}}$). We can conclude that rise delay of the NAND2 depends only on the CD of the PMOS device.

However, both fall delays—due to the transition of A1 ($fall_{\mathrm{A1}}$) or of A2 ($fall_{\mathrm{A2}}$)—are affected by both CD values

TABLE III
Delay Changes Due to the CD Changes of the Transitive Input Gates of a NAND2 Which Has Two Input Poly Lines Corresponding to Two Input Pins A1 and A2

A1 CD (nm)	A2 CD (nm)	$fall_{\mathrm{A1}}$ (s)	$rise_{\mathrm{A1}}$ (s)	$fall_{\mathrm{A2}}$ (s)	$rise_{\mathrm{A2}}$ (s)
51	51	4.979e−11	9.730e−11	5.465e−11	1.131e−10
49	49	4.823e−11	8.825e−11	5.148e−11	1.021e−10
50	50	4.830e−11	9.290e−11	5.308e−11	1.076e−10
51	49	4.905e−11	9.726e−11	5.232e−11	1.022e−10
49	51	4.889e−11	8.828e−11	5.379e−11	1.130e−10

of A1 and A2. For series transistors MN1 and MN2, coloring of MN2 affects cell delay triggered by MN1, and vice versa. We observe that fall delay values in the fifth and sixth rows are in between the values in the second and third rows, which are the slowest and fastest delays when both MN1 and MN2 have smaller or larger CD. Therefore, average CD change of MN1 and MN2 can be used to represent fall delay change.[5]

1) *Coloring Sequence Cost (CSC) for a Timing Arc:* To account for the different impact of bimodal CD changes on the cell delay, we define a *coloring sequence cost* (*CSC*) that scores how poly lines are colored alternately from input to output, i.e., we use *CSC* as a quantitative measure of the alternate coloring of timing paths. The smaller the *CSC* value, the more alternate coloring is in the signal propagation path in a cell, which implies smaller delay variation due to the bimodal CD distribution. For a single N channel metal-oxide-semiconductor (NMOS) or PMOS device, we assign *CSC* value of either 1 or −1 according to the color of the transistor. *CSC* for a network of transistors is calculated as follows.

- *Parallel transistors:* 1 or −1 of the transitive input poly.
- *Series transistors:* Average *CSC* of all series transistors.
- *Fingered transistors:* Average *CSC* of all finger transistors.
- *Cascaded transistors: CSC* of each stage is added up.

Based on the above rules, examples of the *CSC* calculation for BUF, NAND2, and AND2 cells are shown below. We calculate *CSC* for each timing arc for each coloring version of a master cell. For example, $CSC_{C_{12},rise_{\mathrm{A1}}}$ denotes the coloring sequence cost for rise delay due to a transitive input *A* of a coloring version C_{12}. We use "1" and "−1" for the *CSC* of the transistors formed by black and white poly lines in Fig. 9, respectively. For C_{21} cells, the colors of poly lines are inverted.

- *NAND2:* There are two poly lines and one logic stage as shown in Fig. 9(a).
  - $CSC_{C_{12},rise_{\mathrm{A1}}}$ (= MP1: on) = 1.
  - $CSC_{C_{12},fall_{\mathrm{A1}}}$ (= MN1: on, MN2: on) = (1+(−1))/2 = 0.
  - $CSC_{C_{12},rise_{\mathrm{A2}}}$ (= MP2: on) = −1.
  - $CSC_{C_{12},fall_{\mathrm{A2}}}$ (= MN1: on, MN2: on) = (1+(−1))/ 2 = 0.
  - $CSC_{C_{21},rise_{\mathrm{A1}}}$ (= MP1: on) = −1.
  - $CSC_{C_{21},fall_{\mathrm{A1}}}$ (= MN1: on, MN2: on) = (−1+1)/2 = 0.
  - $CSC_{C_{21},rise_{\mathrm{A2}}}$ (= MP2: on) = 1.
  - $CSC_{C_{21},fall_{\mathrm{A2}}}$ (= MN1: on, MN2: on) = (−1 + 1) /2 = 0.

[5]Using the average of MN1 and MN2 may not be accurate, since delay impacts of MN1 and MN2 are different due to different charge-sharing effects for MN1 and MN2. We can extend the methods presented here to obtain and use accurate delay values via cell characterization with different CD combinations.

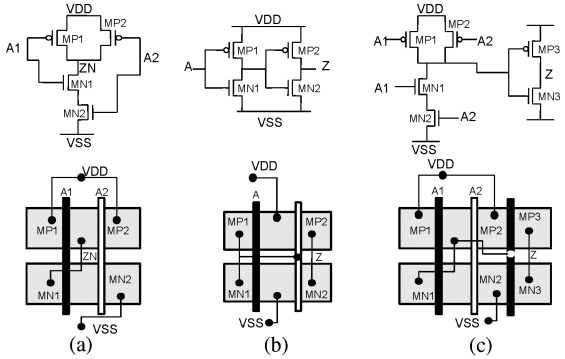

Fig. 9. Schematic and layout of C_{12} type (a) NAND2, (b) BUF, and (c) AND2 cells.

- *BUF:* There are two poly lines which are cascaded (INV followed by INV) as shown in Fig. 9(b).
  - $CSC_{C_{12},rise_A}$ (= MN1: on, MP2: on) = 1 + (−1) = 0.
  - $CSC_{C_{12},fall_A}$ (= MP1: on, MN2: on) = 1 + (−1) = 0.
  - $CSC_{C_{21},rise_A}$ (= MN1: on, MP2: on) = −1 + 1 = 0.
  - $CSC_{C_{21},fall_A}$ (= MP1: on, MN2: on) = −1 + 1 = 0.
- *AND2:* AND2 consists of a NAND2 and an INV as shown in Fig. 9(c). *CSC*s of NAND2 and INV are added. We show only example *CSC* calculations for the rise and fall delay of the C_{12} cell by A1.
  - $CSC_{C_{12},rise_{A1}}$ (= (MN1: on, MN2: on + MP3: on) = (1 + (−1)) / 2 + 1 = 1.
  - $CSC_{C_{12},fall_{A1}}$ (= MP1: on + MN3: on) = 1 + 1 = 2.

In our experimental testbed, we analyze the schematic and layout of all cell masters used in our testcases, and calculate the *CSC* value for each timing arc of each coloring version.

2) *Coloring Sequence Cost for a Path (CSCP):* Given the *CSC* values of all timing arcs, we define the coloring sequence cost of a path (*CSCP*) as a weighted sum of the *CSC* values of its timing arcs. Since the impact of *CSC* is relative to each timing arc delay, the weight is given by the delay value D_l of the timing arc l

$$CSCP_i = \sum_{l \in i} CSC_l \cdot D_l.$$

To verify the correlation between *CSCP* and the actual delay variation, we extract a timing path from a design which has 22 stages of logic cells.[6] We assign a color to each cell in the path randomly, then calculate *CSCP* for each path coloring, and measure the path delay using two bimodal-aware timing libraries, $G1L - G2S$ and $G1S - G2L$.[7]

Fig. 10 shows the correlation between the calculated *CSCP* and the delay difference for 1300 random colorings of the timing path. We observe that *CSCP* and timing variation have a strong positive correlation, i.e., correlation coefficient is 0.902, and rank correlation is 0.900.

[6]The timing path consists of two buffer, three inverter, three 2-input NOR, two 2-input OR, nine 2-input NAND, one 3-input NAND, one 3-input OR–AND, and one 3-input AND–OR gates.

[7]Suppose each group of poly lines in a cell has CD value either of CD1 or of CD2, according to the color of the poly. $G1L - G2S$ ($G1S - G2L$) represents the case that CD1 (CD2) is larger than CD2 (CD1). Bimodal-aware timing libraries have been discussed in detail in Section III-D.

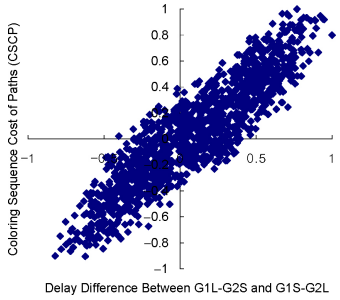

Fig. 10. Correlation between *CSCP* and the delay difference between the two bimodal-aware timing libraries $G1L - G2S$ and $G1S - G2L$, for 1300 different colorings of the timing path.

B. Optimal Color Assignment Problem

Due to the high correlation between *CSCP* and delay variation, we seek to minimize delay variation by minimizing *CSCP* of timing-critical paths.

Optimal timing path coloring problem:

- Given P a set of timing-critical paths.
- Assign coloring of each cell in the timing paths so as to minimize $Max_{i \in P} \mid CSCP_i \mid$, where $CSCP_i$ is the coloring sequence cost of path i. For the top-k timing-critical paths in a design, this can be formulated as an ILP using an indicator variable M for the maximum magnitude of any *CSCP*, and binary variables x_j and y_j to capture the color of a cell j.

Minimization of maximum CSCP:

- Objective
  Minimize M.
- Subject to

$$M \geq CSCP_i, \quad 1 \leq i \leq k$$
$$M \geq -CSCP_i, 1 \leq i \leq k$$
$$CSCP_i = \sum_{l \in i} CSC_{C_{12},l}(j) \cdot x_j + CSC_{C_{21},l}(j) \cdot y_j$$
$$x_j + y_j = 1, \quad x_j \in \{0, 1\}, \quad y_j \in \{0, 1\}$$

where P is a set of k timing-critical paths, and a path i is a set of timing arcs. $CSC_{C_{12},l}(j)$ and $CSC_{C_{21},l}(j)$ are the two different *CSC* values of a timing arc l of a cell j with respect to the two coloring versions of the cell, C_{12} and C_{21}, respectively.

An ILP solver (*ILOG CPLEX v10.110* [27]) is used to solve this problem, and returns the optimal color values for cells in the top-k timing paths. As shown in Table VII, ILP runtimes are reasonable and scale well—e.g., 5.23 s for 1000 timing-critical paths.

V. Coloring Conflict Removal: DPL-Corr

Our timing optimization by alternate color assignment within timing-critical paths, presented in Section IV, can introduce coloring conflicts between neighbors in a row for which colors have already been determined.

Given a cell coloring solution that maximizes the alternate coloring of timing-critical paths, we solve resulting coloring conflicts by placement perturbation within available whitespace using a DP formulation [11]. In other words, we exploit the whitespace available in a standard-cell row to solve coloring conflicts through detailed placement perturbation.

TABLE IV
APPLICABLE DESIGN STAGES

Design Stage	Alternate Coloring	Conflict Removal
Pre-placement	△	×
Post-placement	△	△
Post-clock-synthesis	△	△
Post-routing	◯	◯

Our approach is aware of timing-critical cells and seeks to minimize perturbation of these cells to preserve timing goals of the design.

Table IV summarizes potential applicable design stages of our methodology. △ and ◯ represent applicable design stages, and × represents inapplicable design stages. Each alternate coloring and conflict removal can be used separately and together. Alternate coloring can be performed at every timing optimization stage in the design implementation flow, even before placement, while coloring conflict removal is applicable after placement. However, timing can change significantly at every design optimization stage, so that timing-critical paths can change continually, and new coloring conflicts can occur whenever placement locations or master cells of cell instances change. Consequently, alternate coloring may optimize nontiming critical paths, if used at early design stages, and conflict removal may be performed unnecessarily, if used before any significant timing optimizations, not guaranteeing DPL-correctness at the signoff. Hence, we suggest using our methodology at near-final timing signoff stages—in particular, after detailed routing (denoted by ◯ in Table IV) when timing improvement has "saturated," and most timing-critical paths have been fixed or else clearly identified.

We further note that use of the placement perturbation approach as a post-processing step after cell coloring may not converge to a complete conflict removal in designs with high utilization of layout area. To counter this, we introduce a recoloring approach which recolors the cells during whitespace optimization to remove additional coloring conflicts. We make sure that this recoloring introduces minimum change to the colors of the cells in the alternate color assignment solution obtained in Section IV. We describe these dynamic programming formulations in the rest of this section.

A. Dynamic Programming Formulation for DPL-Corr (SHIFT)

We use the following notation in our problem formulation.

1) L_j^{PS} (R_j^{PS}), respectively, denote the space between the leftmost poly (rightmost poly) and the cell outline of a cell j.
2) L_j^{PC} (R_j^{PC}), respectively, represent the color of the leftmost poly (rightmost poly) for a cell j.
3) x_j denotes the left x-coordinate of a cell j.
4) w_j represents the width of a cell j.
5) δ_j denotes the displacement of a cell j from its original left x-coordinate.
6) The sites in a standard-cell row are indexed from left to right. A cell occupies multiple placement sites in a standard-cell row. s_j denotes the leftmost placement site index for cell j.

Fig. 11 provides a view of the notations used for two adjacent cells a and $a-1$ in a standard-cell row. We consider

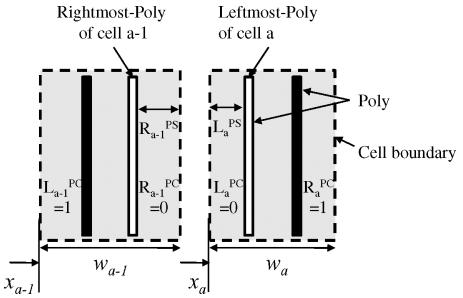

Fig. 11. Variables used in the DP problem formulation.

only the boundary poly lines in the cells (i.e., those with external x-coordinates), as the internal poly lines in a cell are assumed to have been colored alternately and therefore do not have a bearing on the neighboring cells.

Cells can be shifted in multiples of the placement site width, which is the finest positional granularity in the standard-cell row structure. For a given cell a, we formulate the minimum perturbation placement problem for removing coloring conflicts as follows.

$$\text{Minimize} \sum |\delta_i|$$

Subject to

$$x_a + \delta_a - x_{a-1} - \delta_{a-1} - w_{a-1} + L_a^{PS} + R_{a-1}^{PS} \geq Res_{min}$$

when L_a^{PC} and R_{a-1}^{PC} are equal.

We solve this problem via a DP recurrence. The cost function for placing a cell a at placement site b is as follows.

$$\begin{aligned} &Cost(a,b) = \lambda_a |s_a - b| + \\ &Min_{i=x_{a-1}-SRCH}^{x_{a-1}+SRCH}\{Cost(a-1,i) + HCost(a,b,a-1,i)\} \\ &Cost(1,b) = \lambda_1 |s_1 - b| \end{aligned} \tag{5}$$

where λ_a ($= e^{-\alpha slack_a}$) defines the weight of cell a according to its timing criticality using its slack value $slack_a$. This weight determines the relative importance of preserving the initial placement as opposed to displacing the cell to placement site b. The value of α is chosen such that it allows positive-slack cells to move while restricting the movement of timing-critical cells. $SRCH$ is the range over which the cell may be displaced.[8]

HCost denotes the cost of displacing cell a to site b, relative to a's immediate left neighbor in the row and depending on the distance between the corresponding boundary poly lines and their colors. The method for computing the *HCost* is shown in Fig. 12. For each of the displacements made for cell a, we compute the cost of displacing the cell a by computing the cost incurred in displacing the cell by virtue of its criticality. The other cost incurred is the minimum of the summation of cost of displacing its left neighbor over the set of displacements in the search range whose cardinality is $(2\times SRCH + 1)$ and the corresponding *HCost*. We take care of the flipped orientation of cells in the calculation of *HCost*.

[8]As noted above, the displacement is made in multiples of site width, and so the runtime for our algorithm is contained using this $SRCH$ parameter. High-utilization designs may require a large range of displacements to utilize the whitespace available in selective pockets, with increased runtime due to a larger $SRCH$ value.

HCost(a,b,a-1,i) of Cell a

Input:
Displacement of cell a corresponding to site index b: δ_b
Displacement of cell $a-1$ corresponding to site index i: δ_i
x-left coordinate and length of cell a: x_a and w_a
x-left coordinate and length of cell $a-1$: x_{a-1} and w_{a-1}

Output:
Value of *HCost*

Algorithm:
01.**Case** $a = 1$: $HCost(1,b) = 0$
02.**Case** $a > 1$
/* Calculate the spacing between cell a and $a-1$ according to new x-coordinates */
03. $spacing = x_a + \delta_b - x_{a-1} - \delta_i - w_{a-1} + L_a^{PS} + R_{a-1}^{PS}$
04.**If** $((spacing - Res_{min} < 0)$ && $(L_a^{PC} == R_{a-1}^{PC}))$
$HCost(a,b,a-1,i) = \infty$
05.**else**
$HCost(a,b,a-1,i) = 0$

Fig. 12. *HCost* algorithm for coloring conflict removal.

TABLE V
BIMODAL-AWARE TIMING LIBRARIES

CD Mean Diff. (→)	2 nm		4 nm		6 nm	
	CD (nm) of each CD group $G1$ and $G2$					
Corner Name (↓)	G1	G2	G1	G2	G1	G2
Unimodal	53.61		56.32		59.22	
$G1L - G2S$	53	51	54	50	55	49
$G1S - G2L$	51	53	50	54	49	55

For instance, if the cell a is placed in flip-south (FS) or flip-north (FN) orientation—that is, if a is mirrored about the y-axis—then L_a^{PS} corresponds to R_a^{PS} and vice-versa. Hence, R_a^{PC} is used in cost calculation instead of L_a^{PC}.

B. SHIFT with a different objective (MINMAX)

We applied another objective for whitespace management to remove coloring conflicts. In this objective, we tried to minimize the maximum displacement made by a cell in a row rather than minimizing sum of displacements of all cells as done in previous section. We tried this objective since the one pursued in Section V-A can have high standard deviation of the displacements of individual cells from the mean displacement. The objective is defined as follows.

$$\text{Minimize}\{Max\ |\delta_i|\}.$$

Subject to

$$x_a + \delta_a - x_{a-1} - \delta_{a-1} - w_{a-1} + L_a^{PS} + R_{a-1}^{PS} \geq Res_{min}$$

when L_a^{PC} and R_{a-1}^{PC} are equal.

This objective leads to a minor modification in the DP formulation of Section V-A.

C. DP with Recoloring (SHIFT + RECOLOR)

To maintain the timing goals of the design, the timing-critical cells need to be locked in their position while performing detailed placement perturbation. This can lead to very little or no reduction in the number of coloring conflicts as these timing-critical cells block the movement of nontiming critical cells. The lack of whitespace available for perturbation in very high-utilization designs may also lead to nonconvergence of the above-stated DP approach. The only alternate available at our disposal to remove coloring conflicts is the recoloring of the cells which are not fixed by the alternate coloring algorithm in Section IV. Therefore, we need to include the recoloring of the cells into our DP recurrence and assign a cost for recoloring the cell in our DP formulation. We believe that removing coloring conflicts is far more important and should be achieved (even if there is slight degradation in timing) since this is required for printability of the layout patterns. Therefore, we allow recoloring of fixed-color cells too, but they are given high weight as compared to other cells (as is done for timing-critical cells in placement perturbation). We include the color of the cell as a new dimension for formulating DP for this approach. Without loss of generality, we assume that cell a has original color C_{12} and recolored to C_{21}. Therefore, the cost of placing cell a at placement site b when recoloring of cells is allowed is as follows.

$$\begin{aligned}
Cost(a,b,C_{12}) &= \lambda_a\,|s_a - b| + Min[\\
&\quad \{Min_{i=x_{a-1}-SRCH}^{x_{a-1}+SRCH}\{Cost(a-1,i,C_{12}) \\
&\qquad + HCost(a,b,C_{12},a-1,i,C_{12})\}\},\\
&\quad \{Min_{i=x_{a-1}-SRCH}^{x_{a-1}+SRCH}\{Cost(a-1,i,C_{21}) \\
&\qquad + HCost(a,b,C_{12},a-1,i,C_{21})\}\}]\\
Cost(a,b,C_{21}) &= \lambda_a\,|s_a - b| + \lambda_c^a + Min[\\
&\quad \{Min_{i=x_{a-1}-SRCH}^{x_{a-1}+SRCH}\{Cost(a-1,i,C_{12}) \\
&\qquad + HCost(a,b,C_{21},a-1,i,C_{12})\}\},\\
&\quad \{Min_{i=x_{a-1}-SRCH}^{x_{a-1}+SRCH}\{Cost(a-1,i,C_{21}) \\
&\qquad + HCost(a,b,C_{21},a-1,i,C_{21})\}\}]\\
Cost(1,b,C_{12}) &= \lambda_1\,|s_1 - b|\\
Cost(1,b,C_{21}) &= \lambda_1\,|s_1 - b| + \lambda_c^1.
\end{aligned}$$

$Cost(a,b,C_{12})$ denotes the minimum cost of placing the cell a with original color C_{12} at placement site b when the color of cell $a-1$ can be either C_{12} or C_{21}. Similarly, $Cost(a,b,C_{21})$ denotes the minimum cost of placing the recolored cell a with color C_{21} at placement site b when the color of cell $a-1$ can be either C_{12} or C_{21}. The recoloring weight of cell a is defined as λ_c^a $(= e^{-\beta slack_a})$ where β can take different values to assign different recoloring weights to the cells. Fixed-color cells have high-recoloring weight for high values of β because they also happen to be timing-critical cells with negative slack values. As demonstrated below, this SHIFT + RECOLOR DP formulation can achieve conflict-free design by combining (1) displacement of cells ("SHIFT") to add spaces between coloring-conflicting cells with (2) inverting the given coloring of cells ("RECOLOR") to remove coloring conflicts without additional spaces.

VI. EXPERIMENTS

A. Experimental Setup

Library Preparation. We use $G1L - G2S$ and $G1S - G2L$ as the names of bimodal-aware timing libraries corresponding to the scenarios $MAX_{G1} \geq MAX_{G2}$ and $MAX_{G1} \leq MAX_{G2}$, respectively. Each scenario can further be split based on the CD mean difference values between the two groups.

We choose the most commonly used 36 standard cells from *Nangate* 45 nm *Open Cell Library* [24]. We create two coloring versions for each standard cell, e.g., $NAND2\_C12$ and $NAND2\_C21$ for a $NAND2$ cell. We characterize delay of all coloring versions of cells using predictive technology

TABLE VI
TESTCASE INFORMATION

	#Instance		Area	Utilization	Unimodal		Bimodal	
	#-C_{12}	#-C_{21}	(μm^2)	(%)	WNS (ns)	TNS (ns)	WNS (ns)	TNS (ns)
$AES70$	21350	4676	44848	69.10	−0.428	−73.8	−0.190	−14.9
$AES80$	19300	7336	38735	81.26	−0.460	−79.6	−0.197	−19.1
$AES90$	13388	7396	29765	91.15	−0.489	−67.4	−0.239	−12.8
$JPEG70$	77807	15091	175742	68.85	−0.613	−208.3	−0.331	−23.1
$JPEG80$	66928	25742	152430	79.47	−0.641	−191.9	−0.232	−16.1
$JPEG90$	60136	32483	137571	88.31	−0.613	−192.7	−0.229	−20.6
$EXU70$	23764	5933	68066	68.71	−0.446	−173.0	−0.235	−25.1
$EXU80$	19669	9898	58705	79.40	−0.548	−130.5	−0.251	−14.6
$EXU90$	18008	10007	51663	88.37	−0.449	−149.9	−0.199	−15.4
$LSU70$	30831	4673	106385	70.11	−0.448	−89.9	−0.136	−4.8
$LSU80$	27444	7638	93154	79.97	−0.486	−108.4	−0.207	−5.0
$LSU90$	23165	11926	82909	90.20	−0.466	−120.2	−0.212	−8.5

model of Arizona State Univ. [23], with respect to the two scenarios $G1L-G2S$ and $G1S-G2L$ and the three cases of CD mean difference equal to 2, 4, and 6 nm. Table V summarizes timing libraries we generate for bimodal-aware timing analysis. Unimodal CD values corresponding to the bimodal CD values follow the calculation used in [6] and Table I. With the unimodal timing models, we require one timing signoff, while with the bimodal timing models, we need two timing signoffs for each $G1L-G2S$ or $G1S-G2L$ case.

We define the minimum spacing between same-colored poly lines (Res_{min}) to be 330 nm, which is calculated by subtracting the poly width (50 nm) from twice the defined poly pitch (2×190 nm) in the *Nangate* 45 nm library. Separately, all 72 (36×2) standard cells are analyzed and CSC values are recorded.

Testcase Preparation. We implement open-source cores AES and $JPEG$ obtained from the open-source site *opencores.org* [25] along with two sub-blocks of the *OpenSparcT1* design, i.e., LSU (load and store unit) and EXU (execution unit), obtained from the *Sun OpenSPARC Projects* site [26]. We synthesize the cores using *Cadence RTL Compiler v5.2* [28] with the original (non-bimodal) timing library which does not have coloring information and which assumes worst-case CD values of 50 nm for all transistors. We use *Cadence SoC Encounter v7.2* [29] to place and route with three different placement utilizations (70%, 80%, and 90%), to vary the difficulty of coloring conflict removal.

We then assign color C_{12} or C_{21} to all cell instances by replacing the original master cell names, e.g., $NAND2$ with one of its colored master cell names, $NAND2\_C12$ or $NAND2\_C21$. For this initial coloring assignment, the only objective is to not create coloring conflicts which is the only constraint for the traditional DPL pattern decomposition. We first assign a color to the leftmost cell in each cell row and assign a color of the next cell so as to not create a coloring conflict with the first cell, and then iterate this simple assignment method to the end of the cell row. For each initially colored design, we extract resistance and capacitance (RC) parasitics from *SoC Encounter* and then perform timing analysis with *Synopsys PrimeTime vB-2008.12-SP2* [30].

Table VI summarizes design and timing information of our testcases when the 2 nm CD mean difference library is used. WNS and TNS, respectively, represent the worst negative slack of the design and the total negative slack (which is the sum of all negative slacks) over all the end points of timing paths.

TABLE VII
AVERAGE $CSCP$ OF THE TOP-k CRITICAL PATHS AND TNS (NS) REDUCTION VIA THE ALTERNATE COLOR ASSIGNMENT

#Critical Timing Paths (k)		100	200	300	400	1000
Runtime (s)		0.46	4.20	19.38	6.19	5.23
Initial Coloring	Avg. $CSCP$	2.019	2.010	2.016	2.012	2.000
	TNS at 2 nm	−14.91	−14.91	−14.91	−14.91	−14.91
	TNS at 4 nm	−24.38	−24.38	−24.38	−24.38	−24.38
	TNS at 6 nm	−36.09	−36.09	−36.09	−36.09	−36.09
Alternate Coloring	Avg. $CSCP$	1.992	1.962	1.974	1.959	1.957
	TNS at 2 nm	−14.43	−13.40	−12.97	−12.48	−11.91
	TNS at 4 nm	−23.37	−21.68	−20.51	−19.51	−17.08
	TNS at 6 nm	−34.48	−32.87	−31.07	−29.45	−26.29

WNS can be regarded as the feasibility of timing closure at the given clock cycle time, and TNS can be regarded as the required effort to fix all timing violations of the design. Timing with the original (single CD distribution) timing library is met at the given clock cycle times. However, due to the bimodal CD distribution, timing of the double patterning-applied designs is significantly degraded. It must be noted that the unimodal timing analysis is more pessimistic than the bimodal-aware timing analysis. We also observe that the use of the bimodal-aware timing library can by itself directly improve timing significantly, due to the "intrinsic" alternate coloring within a cell, as we observe in the fall delay of a NAND2 in which CD variation of MN1 is compensated by opposite CD variation of MN2 (Fig. 9).

B. Experimental Flow

Fig. 13 shows our design optimization framework for double patterning. Major steps are in the left-hand side and output data are in the right-hand side. Solid arrows show the design flow and dashed arrows show the data flow.

- *Step 1 (initial design):* For the initial testcase preparation, we use a traditional timing-driven design implementation flow. The design starts with RTL netlists and timing constraints, and is synthesized, placed and routed with the original (traditional) worst-case timing library of single CD distribution.
- *Step 2 (initial coloring):* The framework performs initial coloring for double patterning, in which no coloring conflicts are allowed. The output is a design exchange format (initial_colored.def).
- *Step 3 (timing analysis):* Based on the coloring information and bimodal-aware timing libraries, a static timer analyzes timing, and generates an ILP problem instance for top-k timing-critical paths.

TABLE VIII

WNS (NS) AND TNS (NS) COMPARISON BEFORE AND AFTER ALTERNATE COLORING FOR DIFFERENT TESTCASES

CD diff.	2 nm				4 nm				6 nm			
Coloring	Initial Coloring		Alternate Coloring		Initial Coloring		Alternate Coloring		Initial Coloring		Alternate Coloring	
Timing Slack	WNS	TNS	WNS	TNS	WNS	TNS	WNS	TNS	WNS	TNS	WNS	TNS
*AES*70	−0.189	−14.91	−0.154	−12.48	−0.257	−24.38	−0.192	−19.51	−0.388	−36.09	−0.293	−29.45
*AES*80	−0.197	−19.07	−0.190	−16.10	−0.234	−26.41	−0.220	−22.65	−0.309	−36.52	−0.299	−32.53
*AES*90	−0.239	−12.83	−0.186	−11.17	−0.301	−21.68	−0.250	−18.52	−0.430	−32.47	−0.386	−28.09
*JPEG*70	−0.331	−23.08	−0.173	−13.20	−0.498	−50.76	−0.229	−22.34	−0.663	−105.15	−0.391	−49.29
*JPEG*80	−0.232	−16.15	−0.207	−10.68	−0.371	−33.03	−0.337	−15.50	−0.561	−68.52	−0.484	−34.27
*JPEG*90	−0.229	−20.56	−0.210	−12.12	−0.334	−38.77	−0.274	−18.49	−0.532	−78.97	−0.393	−40.26
*EXU*70	−0.235	−25.11	−0.206	−18.31	−0.266	−42.99	−0.197	−22.38	−0.348	−75.46	−0.265	−37.88
*EXU*80	−0.251	−14.57	−0.247	−12.04	−0.308	−21.49	−0.276	−15.99	−0.379	−35.00	−0.351	−26.72
*EXU*90	−0.199	−15.38	−0.173	−14.75	−0.255	−23.69	−0.211	−21.41	−0.358	−39.31	−0.297	−34.00
*LSU*70	−0.136	−4.78	−0.057	−0.91	−0.230	−13.33	−0.093	−2.78	−0.330	−30.31	−0.169	−7.32
*LSU*80	−0.207	−4.98	−0.206	−4.22	−0.245	−10.51	−0.220	−6.59	−0.307	−21.04	−0.273	−13.71
*LSU*90	−0.212	−8.52	−0.161	−6.94	−0.288	−15.58	−0.251	−10.34	−0.405	−26.52	−0.343	−17.95

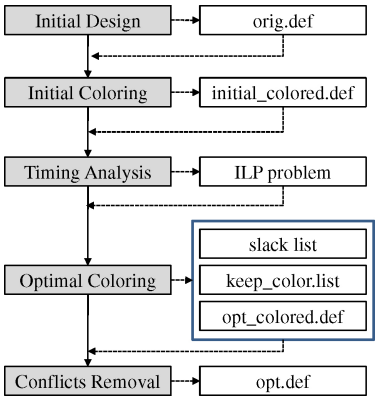

Fig. 13. Design framework for bimodal-aware timing optimization.

- *Step 4 (optimal coloring):* The ILP solver finds the optimal alternate coloring solution for the selected timing paths, and at the same time, a pre-defined color is assigned to all clock buffers.[9]
- *Step 5 (conflict removal):* DPL-Corr solves the coloring conflicts caused during Step 4, subject to the coloring constraints (keep_color.list) and timing constraints (slack.list). Partially disconnected nets due to the placement perturbation in DPL-Corr are ECO-routed, and a final design (opt.def) that does not have coloring conflicts is generated.

The orig.def is not suitable for double patterning. After Step 2, initial_colored.def is applicable for double patterning, since this design does not have coloring conflicts. We use this initial_colored.def as the reference double patterning-applied design for our comparison. The opt_colored.def is used to show the pure effect of our optimal coloring method, since the placement locations of cells and the routing are not disturbed from the reference design. We finally compare the timing quality of the final design (opt.def) with the reference design (initial_colored.def). We note that our optimization framework does not increase the given die area at all, since it perturbs the placement using existing whitespace between placed cells.

[9]Although we suggested in Section III-D to use the same color for all clock buffers, this can introduce a large number of coloring conflicts when the number of clock buffers is large (indeed, in some complex SoCs with many disjoint clock trees, more than 10% of total cell instances can be clock buffers). In this case, a methodological approach of *intrinsic* alternate coloring within block buffer and inverter cells can be exploited: if clock buffers/inverters have only an even number of fingers in every transistor, then the *CSC* values of those cells become zero. Hence, we can use either C_{12} or C_{21} cells arbitrarily in the clock network without causing any bimodality-induced clock skew problem. This methodological approach would change the designs of library cells.

TABLE IX

PERFORMANCE COMPARISON OF MINMAX AND SHIFT FOR *AES*70

	MD	SOMR	SODTCC	SODNTCC
SHIFT	3.42	134.14	880.08	2835.94
MINMAX	3.42	115.90	1131.26	3401.95

MD denotes maximum displacement for any standard-cell row in the design.

C. Experimental Results

Our first experiment is to verify the quality of our alternate color assignment method, varying the number of timing-critical paths taken into account. We apply the alternate color assignment to the timing-critical paths and uniform color assignment to the clock paths, but do not apply DPL-Corr that may introduce other timing uncertainty from placement perturbation and ECO-routing. Table VII shows $CSCP$ and TNS reduction from the alternate color assignment on different top-k critical paths of $AES70$. Runtime is listed in Row 2. Average $|CSCP|$ of the top-k critical paths (Avg. $CSCP$) of the initially colored design ("Initial Coloring") and of the alternately colored design ("Alternate Coloring") is given in rows 3 and 7, respectively. TNS is calculated using the two worst-corner bimodal-aware timing libraries $G1L - G2S$ and $G1S - G2L$. Between the two timing results, the worse one is reported. We observe that as the number of timing paths increases, the average $CSCP$ of the optimized design decreases. As a result, the total negative slack decreases.

Our second experiment is to verify the quality of our alternate color assignment method on different testcases. Table VIII summarizes the WNS and TNS reduction from the alternate color assignment for different testcases.[10] We observe WNS is reduced by more than 150 ps and TNS is improved by around 10 ns (from −23.1 ns to −13.2 ns) in $JPEG70$, even with 2 nm CD mean difference libraries.

Our third experiment is to measure the performance of the proposed DPL-Corr in Section V. Table X shows results of the DP-based coloring conflict removal algorithm on different testcases. The table shows the performance statistics for DP implementation on testcases with "Random Coloring" and "Alternate Coloring" respectively. The experiments are performed for different values of α to highlight the performance

[10]For this experiment, we consider top-400 critical paths for AES testcases, top-1000 critical paths for EXU and LSU testcases, and top-2000 critical paths for $JPEG$ testcases.

TABLE X

DP-BASED COLORING CONFLICT REMOVAL. SODTCC AND SODNTCC DENOTE SUM OF DISPLACEMENTS (μm) OF TIMING-CRITICAL AND NON-TIMING-CRITICAL CELLS, RESPECTIVELY

	Testcase	#Conflicts	α	SHIFT				SHIFT + RECOLOR				
				#Conflicts	SODTCC	SODNTCC	Runtime (s)	#Conflicts	SODTCC	SODNTCC	FCCD	Runtime (s)
Random	*AES*90	6574	0	4394	995.22	8961.73	52.22	0	12.35	13.68	27	205.76
			20	4394	971.09	9638.51	52.05	0	11.59	42.18	29	209.42
			∞	6516	0	88.16	53.76	0	0	4824.67	32	218.02
	*JPEG*90	27296	0	16668	4785.34	49532.05	347.23	0	64.98	71.25	152	607.88
			20	16668	4660.13	53842.2	347.73	0	54.72	213.75	158	607.86
			∞	27296	0	0	348.04	0	0	22005.04	199	678.69
	*EXU*90	8841	0	5344	5990	12101	69.79	0	909	690	11	191.02
			20	5344	6098	12914	69.89	0	510	589	63	191.26
			∞	8841	0	0	72.67	0	0	1559	138	210.74
	*LSU*90	10114	0	3275	10603	17623	84.79	0	3329	2413	0	229.77
			20	3275	10601	18723	84.80	0	2994	2630	44	229.81
			∞	10112	0	0.85	91.44	0	0	939	136	264.91
Alternate	*AES*90	158	0	42	276.26	323.76	49.98	0	94	78	2	145.12
			20	42	303.43	382.47	51.57	0	21	27	15	145.25
			∞	151	0	5.89	53.43	0	0	69	25	158.59
	*JPEG*90	2036	0	0	2523.20	4751.52	213.88	0	594	446	86	609.3
			20	0	2664.94	5568.14	213.78	0	245	334	298	609.05
			∞	2026	0	12.16	241.39	0	0	373	689	693.12
	*EXU*90	443	0	0	535.61	969.00	66.80	0	211	99	47	193.54
			20	0	580.26	1214.1	66.73	0	139	99	78	192.75
			∞	439	0	3.61	72.86	0	0	264	148	210.61
	*LSU*90	411	0	0	523.64	399.19	80.55	0	289	136	44	231.18
			20	0	605.91	530.67	80.28	0	193	124	90	231.48
			∞	403	0	11.21	90.45	0	0	124	169	266.11

FCCD represents the number of recolored cells during DPL-Corr. Results for other testcases are available in [31].

of the algorithm as the movement of timing-critical cells is restricted progressively from unrestricted movement to no movement. We also compare the performance of SHIFT and MINMAX on testcase $AES70$ with "Random Coloring" in Table IX. The value of α is 0. The value of the sum of maximum displacements of over all rows (SOMR) is lower for MINMAX than for SHIFT, as expected, but the improvement is not significant. On the contrary, the sum of displacements of timing-critical cells (SODTCC) and the sum of displacements of nontiming critical cells (SODNTCC) increases substantially for MINMAX as compared to the values reported for SHIFT. Therefore, we have chosen SHIFT for SHIFT + RECOLOR which seeks complete conflict removal.

In Table X, performance data are reported for both DP variants, i.e., SHIFT and SHIFT + RECOLOR. The value of $SRCH$ is taken as 50 and the value of β for determining recoloring weight is 30. We report the number of conflicts after placement optimization in columns 5 and 9, SODTCC in columns 6 and 10, and SODNTCC in columns 7 and 11 with respect to SHIFT and SHIFT + RECOLOR. The number of recolorings of fixed-color cells (FCCD) is in column 12. SODTCC (SODNTCC), decreases (increases) with increase in the value of α as the movement of timing-critical cells is restricted and movement of noncritical cells is needed to compensate for that restriction to achieve the same results. Runtime in columns 8 and 13 confirms linear scaling of our algorithms with respect to instance count.

The first DPL-Corr algorithm SHIFT is able to solve all the coloring conflicts for testcases with a given random coloring at 70% and 80% placement utilization, when α values are 0 and 20. For high-utilization testcases ($AES90$ and $JPEG90$), the algorithm is able to remove around 33% and 39%, respectively, of conflicts without any recoloring. Understandably, the algorithm performance suffers drastically when α is taken as ∞, since this means that all timing-critical cells are locked in their positions. The number of conflicts can be reduced by increasing the $SRCH$ range for displacements since the whitespace is often not distributed evenly over the entire standard-cell row. This approach can reduce the number of conflicts but the runtime can increase substantially. It should be noted that whitespace management alone cannot guarantee complete conflict removal in high-utilization designs because the algorithm is restricted by the lack of whitespace needed for complete conflict removal.

TABLE XI

COMPARISONS OF WNS AND TNS, BEFORE AND AFTER APPLYING ALTERNATE COLORING AND DPL-CORR SHIFT

Stage	Timing	Mean CD Difference		
		2 nm	4 nm	6 nm
Initial Coloring	WNS (ps) w/ $G1L - G2S$	−331	−498	−663
	WNS (ps) w/ $G1S - G2L$	−122	−155	−224
	Worst WNS (ps)	−331	−498	−663
	WNS diff. (ps)	209	343	439
	TNS (ns) w/ $G1L - G2S$	−23.08	−50.76	−105.15
	TNS (ns) w/ $G1S - G2L$	−4.03	−3.44	−6.57
	Worst TNS (ns)	−23.08	−50.76	−105.15
	TNS diff. (ns)	19.05	47.32	98.58
Alternate Coloring + DPL-Corr (SHIFT)	WNS (ps) w/ $G1L - G2S$	−165	−230	−392
	WNS (ps) w/ $G1S - G2L$	−174	−186	−260
	Worst WNS (ps)	−174	−230	−392
	WNS diff. (ps)	9	44	132
	TNS (ns) w/ $G1L - G2S$	−13.27	−22.40	−49.40
	TNS (ns) w/ $G1S - G2L$	−9.45	−12.89	−22.17
	Worst TNS (ns)	−13.27	−22.40	−49.40
	TNS diff. (ns)	3.82	9.51	27.23

The goal of complete conflict removal can be realized fully with our second DPL-Corr algorithm, SHIFT + RECOLOR. We apply this DP algorithm to the alternate coloring results since this is the case of interest from a flow and methodology standpoint. The number of conflicts reduces to 0 for all values of α, albeit with a slight penalty incurred in recoloring the fixed color cells obtained from alternate coloring results. SODTCC and SODNTCC are both reduced considerably with this approach, implying less effort in subsequent ECO-routing.

TABLE XII
COMPARISONS OF WNS AND TNS BEFORE AND AFTER APPLYING ALTERNATE COLORING AND DPL-CORR SHIFT + RECOLOR

Testcase	Stage	Timing	Mean CD Difference		
			2 nm	4 nm	6 nm
*AES*90	Initial Coloring	WNS (ns)	−0.239	−0.301	−0.430
		TNS (ns)	−12.83	−21.68	−32.47
	Alternate Coloring Only	WNS (ns)	−0.186	−0.250	−0.386
		TNS (ns)	−11.17	−18.52	−28.09
	DPL-Corr (SHIFT + RECOLOR)	WNS (ns)	−0.209	−0.316	−0.398
		TNS (ns)	−11.19	−8.58	−28.14
*JPEG*90	Initial Coloring	WNS (ns)	−0.229	−0.334	−0.532
		TNS (ns)	−20.56	−38.77	−78.97
	Alternate Coloring Only	WNS (ns)	−0.210	−0.274	−0.393
		TNS (ns)	−12.12	−18.49	−40.26
	DPL-Corr (SHIFT + RECOLOR)	WNS (ns)	−0.198	−0.231	−0.357
		TNS (ns)	−12.45	−18.43	−29.70
*EXU*90	Initial Coloring	WNS (ns)	−0.199	−0.255	−0.358
		TNS (ns)	−15.38	−23.69	−39.31
	Alternate Coloring Only	WNS (ns)	−0.173	−0.211	−0.297
		TNS (ns)	−14.75	−21.41	−34.00
	DPL-Corr (SHIFT + RECOLOR)	WNS (ns)	−0.173	−0.212	−0.296
		TNS (ns)	−14.44	−20.75	−32.72
*LSU*90	Initial Coloring	WNS (ns)	−0.212	−0.288	−0.405
		TNS (ns)	−8.52	−15.58	−26.52
	Alternate Coloring Only	WNS (ns)	−0.161	−0.251	−0.343
		TNS (ns)	−6.94	−10.34	−17.95
	DPL-Corr (SHIFT + RECOLOR)	WNS (ns)	−0.176	−0.252	−0.344
		TNS (ns)	−6.57	−9.43	−16.00

We also observe that there is a slight degradation in timing when DP-based recoloring is applied; this is attributable to the fact that the colors of some fixed cells are changed. The runtime is of the order of several minutes, since the number of conflicts is smaller. In general, we observe that DP with recoloring can handle all of our testcases with ease and can achieve complete conflict removal with practical runtime.

Finally, we compare timing slack and timing slack variation before and after coloring optimization, including the effect of the placement perturbation and ECO-routing due to the conflict removal. Table XI shows timing of the initial coloring and timing after coloring optimization with DPL-Corr SHIFT, for *JPEG*70 testcase. We use timing criticality weight $\alpha = 20$. "Worst WNS" and "worst TNS" represent the worse timing slack between the two corner libraries, and "WNS diff." and "TNS diff." represent the slack difference between the two corner libraries. Due to the placement perturbation and ECO-routing, the worst negative slack is degraded by at most 1 ps from the results of the alternate color assignment only. However, we still observe up to 157 ps, 268 ps and 271 ps of WNS reduction, and 9.81 ns, 28.36 ns and 55.75 ns of TNS reduction for 2 nm, 4 nm and 6 nm CD mean difference in bimodal CD distribution, respectively, in the *JPEG*70 testcase. In addition, the maximum variation of the worst (total) negative slack between two corner libraries is reduced from 439 ps (98.58 ns) to 132 ps (27.23 ns) for 6 nm CD mean difference libraries. This implies more robust timing of the design with respect to CD distribution changes of $G1$ and $G2$.

In the case that the design is very congested, our second DPL-Corr algorithm SHIFT + RECOLOR can completely solve the coloring conflicts without significant timing degradation. Table XII summarizes timing slack changes due to alternate color assignment and DPL-Corr, including ECO-routing and recoloring. Again, the worse WNS and TNS values are chosen from the results of the two bimodal-aware timing analyses. We report the highest utilization cases for each design.

VII. CONCLUSION AND ONGOING WORK

DPL is an inevitable solution and is being adopted for 32 nm and below technologies. However, due to bimodality, i.e., two CD populations within a die, on-chip timing variability increases substantially beyond the variability that occurs with traditional single-exposure lithography. As shown in our analysis in Fig. 7 and Table VI, design guardband and timing slack in double patterning can each degrade by up to 2×, and this will significantly hinder the 13% per year of device performance improvement expected in [22] (see studies of "cost of guardband" in [15]).

To mitigate the timing variability in double patterning, we have proposed a new metric that quantifies the delay variation of timing paths, and implemented an optimal cell-based timing-aware color assignment technique for double patterning that reduces both timing delay as well as timing variation. To address the increased coloring conflicts due to this intentional timing-aware coloring, we have also proposed a dynamic programming-based detailed placement algorithm that minimizes coloring conflicts by perturbing placement and exploiting whitespace in the given placement.

With this new methodology, we effectively reduce the timing delay as well as timing variation for DPL-patterned designs. We achieve maximum 271 ps (55.75 ns) reduction in the worst (total) negative slack and 70% (72%) reduction in the worst (total) negative slack variation in double patterning-applied designs.

Our next goals are: (1) to analyze the net benefits of adopting double patterning with consideration of bimodality, so that designers and lithographers can best trade off design and process margins; (2) to seek more accurate metrics (objective functions) for further enhancement of timing quality through timing path balancing; (3) to explore different objectives for the placement perturbation; (4) to investigate the tradeoff between recoloring and displacement in terms of impact on timing quality; and (5) to develop a simultaneous timing-aware coloring and conflict removal methodology as a golden timing and placement optimizer for double-patterning lithography in the presence of bimodality.

ACKNOWLEDGMENT

Authors would like to thank A. Hoover at Freescale Semiconductor for suggesting the usage of even number of fingers in all devices of the clock distribution (buffers, inverters) to mitigate bimodality.

REFERENCES

[1] A. M. Biswas, J. Li, J. A. Hiserote, and L. S. Melvin, III, "Extension of 193 nm dry lithography to 45-nm half-pitch node: Double exposure and double processing technique," *Proc. SPIE Photomask Technol.*, vol. 6349, pp. 63491P-1–63491P-9, 2006.

[2] G. Capetti, P. Cantu, E. Galassini, A. V. Pret, C. Turco, A. Vaccaro, P. Rigolli, F. D'Angelo, and G. Cotti, "Sub k1 = 0.25 lithography with double patterning technique for 45 nm technology node flash memory devices at 193 nm," *Proc. SPIE Opt. Microlithogr.*, vol. 6520, pp. 65202K-1–65202K-12, 2007.

[3] K. Chen, W. Huang, W. Li, and P. Varanasi, "Resist freezing process for double-exposure lithography," *Proc. SPIE Adv. Resist Mater. Process. Technol.*, vol. 6923, pp. 69230G-1–69230G-10, 2008.
[4] M. Cho, Y. Ban, and D. Z. Pan, "Double patterning technology friendly detailed routing," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, 2008, pp. 506–511.
[5] M. Drapeau, V. Wiaux, E. Hendrickx, S. Verhaegen, and T. Machida, "Double patterning design split implementation and validation for the 32 nm node," *Proc. SPIE Design Manufacturability*, vol. 6521, pp. 652109-1–652109-15, 2007.
[6] M. Dusa, J. Quaedackers, O. F. A. Larsen, J. Meessen, E. van der Heijden, G. Dicker, O. Wismans, P. de Haas, K. van Ingen Schenau, J. Finders, B. Vleeming, G. Storms, P. Jaenen, S. Cheng, and M. Maenhoudt, "Pitch doubling through dual-patterning lithography: Challenges in integration and litho budgets," *Proc. SPIE Opt. Microlithography*, vol. 6520, pp. 65200G-1–65200G-10, 2007.
[7] J. Finders, M. Dusa, and S. Hsu, "Double patterning lithography: The bridge between low k1 ArF and EUV," *Microlithography World*, Feb. 2008.
[8] R. S. Ghaida and P. Gupta, "Design-overlay interactions in metal double patterning," *Proc. SPIE Design Manufacturability Through Design-Process Integr.*, vol. 7275, pp. 727514-1–727514-10, 2009.
[9] M. Gupta, K. Jeong, and A. B. Kahng, "Timing yield-aware color reassignment and detailed placement perturbation for double patterning lithography," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, 2009, pp. 607–614.
[10] P. Gupta, A. B. Kahng, Y. Kim, and D. Sylvester, "Self-compensating design for focus variation," in *Proc. ACM/EDAC/IEEE Design Autom. Conf.*, 2005, pp. 365–368.
[11] P. Gupta, A. B. Kahng, and C.-H. Park, "Detailed placement for improved depth of focus and CD control," in *Proc. Asia South Pacific Design Autom. Conf.*, 2005, pp. 343–348.
[12] B. Hwang, N. Lim, J.-H. Park, S. Jin, M. Kim, J. Jung, B, Kwon, J. Hong, J. Han, D. Kwak, J. Park, J. Choi, and W.-S. Lee, "Development of 38 nm bit-lines using copper damascene process for 64-giga bits NAND flash," in *Proc. Adv. Semicond. Manuf. Conf.*, 2008, pp. 49–51.
[13] M. Hori, T. Nagai, A. Nakamura, T. Abe, G. Wakamatsu, T. Kakizawa, Y. Anno, M. Sugiura, S. Kusumoto, Y. Yamaguchi, and T. Shimokawa, "Sub-40-nm half-pitch double patterning with resist freezing process," *Proc. SPIE Adv. Resist Mater. Process. Technol.*, vol. 6923, pp. 69230H-1–69230H-8, 2008.
[14] K. Jeong and A. B. Kahng, "Timing analysis and optimization implications of bimodal CD distribution in double patterning lithography," in *Proc. Asia South Pacific Design Autom. Conf.*, 2009, pp. 486–491.
[15] K. Jeong, A. B. Kahng, and K. Samadi, "Quantified impacts of guardband reduction on design process outcomes," in *Proc. Int. Symp. Quality Electron. Design*, 2008, pp. 890–897.
[16] K. Jeong, A. B. Kahng, and R. O. Topaloglu, "Is overlay error more important than interconnect variations in double patterning," in *Proc. ACM Int. Workshop Syst. Level Interconnect Prediction*, 2009, pp. 3–10.
[17] A. B. Kahng, C.-H. Park, X. Xu, and H. Yao, "Layout decomposition for double patterning lithography," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, 2008, pp. 465–472.
[18] S.-M. Kim, S.-Y. Koo, J.-S. Choi, Y.-S. Hwang, J.-W. Park, E.-K. Kang, C.-M. Lim, S.-C. Moon, and J.-W. Kim, "Issues and challenges of double patterning lithography in DRAM," *Proc. SPIE Opt. Microlithography*, vol. 6520, pp. 65200H-1–65200H-7, 2006.
[19] M. Maenhoudt, J. Versluijs, H. Struyf, J. van Olmen, and M. van Hove, "Double patterning scheme for sub-0.25 k1 single damascene structures at NA = 0.75, λ = 193 nm," *Proc. SPIE Opt. Microlithography*, pp. 1508–1518, 2005.
[20] K. Yuan, J.-S. Yang, and D. Z. Pan, "Double patterning layout decomposition for simultaneous conflict and stitch minimization," in *Proc. ACM Int. Symp. Phys. Design*, 2009, pp. 107–114.
[21] J.-S. Yang and D. Z. Pan, "Overlay aware interconnect and timing variation modeling for double patterning technology," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, 2008, pp. 488–493.
[22] *International Technology Roadmap for Semiconductors*, 2007 ed. [Online]. Available: http://public.itrs.net/
[23] *Predictive Technology Model* [Online]. Available: http://www.eas.asu.edu/∼ptm
[24] *NANGATE* [Online]. Available: http://www.nangate.com/
[25] *OPENCORES* [Online]. Available: http://www.opencores.org/
[26] *Sun OpenSPARC Project* [Online]. Available: http://www.sun.com/processors/opensparc/
[27] *ILOG CPLEX* [Online]. Available: http://www.ilog.com/products/cplex/
[28] *Cadence RTL Compiler User's Manual* [Online]. Available: http://www.cadence.com/eu/Pages/rtl_compiler.aspx
[29] *Cadence SoC Encounter User's Manual* [Online]. Available: http://www.cadence.com/products/di/soc_encounter/Pages/
[30] *Synopsys PrimeTime User's Manual* [Online]. Available: http://www.synopsys.com/Tools/Implementat-ion/SignOff/Pages/PrimeTime.aspx
[31] *Table X: DP-Based Coloring Conflict Removal for All Testcases (full version)* [Online]. Available: http://vlsicad.ucsd.edu/DPLCorr-Results/TableX.pdf

Mohit Gupta received the B.Tech. degree in electrical engineering from the Indian Institute of Technology Bombay, Mumbai, India, in 2005. He is currently pursuing the Ph.D. degree from the VLSI CAD Laboratory, Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla.

From 2005 to 2008, he was a Senior Research and Development Engineer with Mirafra Technologies, Bangalore, India, an electronic design automation product consulting company. His current research interests include physical design and very large scale integration design-manufacturing interface.

Kwangok Jeong (S'07) received the B.S. and M.S. degrees in electrical engineering from Hanyang University, Seoul, Korea, in 1997 and 1999, respectively. He is currently pursuing the Ph.D. degree from the VLSI CAD Laboratory, Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla.

From 1999 to 2006, he was with the Computer-Aided Engineering Team, Samsung Electronics, Seoul, Korea. His current research interests include physical design and very large scale integration design-manufacturing interface.

Mr. Jeong received three Best Paper Awards at the Samsung Technical Conference in 2001, 2002, and 2004, respectively, and the Honorable Outstanding Researcher Award in 2005.

Andrew B. Kahng (F'10) received the A.B. degree in applied mathematics (physics) from Harvard College, Cambridge, MA, and the M.S. and Ph.D. degrees in computer science from the University of California at San Diego (UCSD), La Jolla.

He was an Assistant Professor, an Associate Professor, and a Full Professor with the Computer Science Department, University of California, Los Angeles, in 1989, 1994, and 1998, respectively. In 2001, he joined UCSD as a Professor in the Computer Science and Engineering (CSE) and Electrical and Computer Engineering Departments. He has published over 300 journal and conference papers. Since 1997, his research in integrated circuit design for manufacturability has pioneered methods for automated phase-shift mask layout, variability-aware analyses and optimizations, chemical-mechanical polishing fill synthesis, and parametric yield-driven, cost-driven methodologies for chip implementation. In 2004, he co-founded Blaze DFM, Inc., Sunnyvale, CA, and served as CTO of the company until resuming his duties with UCSD in 2006.

Dr. Kahng served as an Associate Chair of the UCSD CSE Department from 2003 to 2004. He was the founding General Chair of the 1997 ACM International Symposium on Physical Design and co-founder of the ACM Workshop on System-Level Interconnect Prediction, and defined the physical design roadmap as a member of the Design Tools and Test Technology Working Group (TWG) in 1997, 1998 and 1999 renewals of the International Technology Roadmap for Semiconductors. From 2000 to 2003, he was the Chair of both the U.S. Design TWG and the Design International Technology Working Group (ITWG), and continues to serve as a co-chair of the Design ITWG. Since 1998, he has been an executive committee member of the MARCO Gigascale Systems Research Center. He has received NSF Research Initiation and Young Investigator Awards, 13 Best Paper Nominations, and six Best Paper Awards.