Self-Compensating Design for Reduction of Timing and Leakage Sensitivity to Systematic Pattern Dependent Variation

Puneet Gupta\textsuperscript{a}, Andrew B. Kahng\textsuperscript{a}, Youngmin Kim\textsuperscript{b}, Dennis Sylvester\textsuperscript{b}

\textsuperscript{a}Blaze DFM Inc., Sunnyvale, CA, \textsuperscript{b}EECS Department, University of Michigan at Ann Arbor

\{puneet, abk\}@blaze-dfm.com, \{kimyz, dennis\}@eecs.umich.edu

ABSTRACT

Focus is one of the major sources of linewidth variation. CD variation caused by defocus is largely systematic after the layout is finished. In particular, dense lines “smile” through focus while isolated lines “frown” in typical Bossung plots. This well-defined systematic behavior of focus-dependent CD variation allows us to develop a self-compensating design methodology.

In this work, we propose a novel design methodology that allows explicit compensation of focus-dependent CD variation, either within a cell (self-compensated cells) or across cells in a critical path (self-compensated design). By creating iso and dense variants for each library cell, we can achieve designs that are more robust to focus variation. Optimization with a mixture of iso and dense cell variants is possible both for area and leakage power, with the latter providing an interesting complement to existing leakage reduction techniques such as dual-Vth. We implement both heuristic and Mixed-Integer Linear Programming (MILP) solution methods to address this optimization, and experimentally compare their results. Our results indicate that designing with a self-compensated cell library incurs ~12% area penalty and ~6% leakage increase over original layouts while compensating for focus-dependent CD variation (i.e., the design meets timing constraints across a large range of focus variation). We observe ~27% area penalty and ~7% leakage increase at the worst-case defocus condition using only single-pitch cells. The area penalty of circuits after using either the heuristic or MILP optimization approach is reduced to ~3% while maintaining timing. We also apply our optimizations to leakage, which traditionally shows very large variability due to its exponential relationship with gate CD. We conclude that a mixed iso/dense library combined with a sensitivity-based optimization approach yields much better area/timing/leakage tradeoffs than using a self-compensated cell library alone. Self-compensated design shows an average of 25% leakage reduction at the worst defocus condition for the benchmark designs that we have studied.

Keywords: ACLV, Focus, Leakage, Manufacturability, Self-Compensating, Variation, Systematic

1. INTRODUCTION

Within-die process variation has become one of the most important considerations in IC manufacturing, particularly as lithography moves into the deep-subwavelength regime. Variation can occur at the fabrication stage (intrinsic variation) or during circuit operation (dynamic variation)\textsuperscript{1}. There are two major components to intrinsic variation: random and systematic\textsuperscript{1,2,5}. Because of the strong layout dependency of the systematic component, estimation of systematic variation is impossible until layout information is available. Due to numerous variation sources and their interactions, systematic variation is difficult to predict and is often treated as random.

Effective channel length (\(L_{\text{eff}}\)) variation is one of the biggest determinants of IC performance\textsuperscript{3}. Prohibitive increases in the cost of process control necessitate relaxed control of \(L_{\text{eff}}\) from a manufacturing perspective, shifting the focus to more proactive management of \(L_{\text{eff}}\) variation from a design perspective. Across-chip linewidth variation (ACLV) control is critical to the timing and functionality of a design\textsuperscript{4}. Various RETs (Resolution Enhancement Techniques) such as SRAF (Sub-Resolution Assistant Feature), OPC (Optical Proximity Correction), and PSM (Phase Shifting Mask) are commonly used to achieve ACLV control and larger process window in current design-to-manufacturing flows\textsuperscript{8,15,16}.

One of the major sources of \(L_{\text{eff}}\) variation is focus\textsuperscript{7,9}. Focus variations can occur, for example, due to changes in wafer flatness or lens imperfections. Traditional corner-case timing analysis flows are very pessimistic in worst-casing focus impact on critical dimensions. This is because layout pitch and focus have very systematic interactions, as shown by so-
called Bossung plots (e.g., Figure 1). As shown in Figure 1, though there is little through-pitch linewidth variation at zero or best focus condition (for example, due to OPC), typically the poly linewidth at defocus increases significantly with decreasing pitch. Restricted design rules (RDRs) with only one gate pitch allowed have been recently studied in literature as a way to cope with systematic lithographic variation. Several recent works\textsuperscript{7,8,9} have examined the tradeoffs in such a design methodology. The biggest complaint with RDRs has been a possibly large increase in chip area. We compare our self-compensated design methodology with the single-pitch cells based designs later in this work.

A recent work\textsuperscript{6} has presented a study of systematic variation and its impact on static timing analysis. A novel static timing analysis approach which exploits the effect of proximity effects and focus variation is proposed\textsuperscript{6}. The authors compare the resulting systematic-variation aware timing with standard corner-based timing, and claim up to 40\% reduction of timing uncertainty using this methodology.

We have earlier investigated the possibility of designs that are robust to focus variation, and we investigate the optimization of timing and area using estimated layouts and timing\textsuperscript{23}. We extend the DAC05 work\textsuperscript{23} in three ways. (1) Actual cell layout generation, parasitic extraction, and cell characterization are performed to validate the correctness of the previous DAC05 approach. (2) We change the optimization objective to leakage from area. (3) We take into account the nonlinear nature of focus-dependent CD variation and propose a method which exploits integer linear programming techniques to ensure timing correctness of the design throughout the entire range of defocus. In addition, single-pitched cells in which all the device linewidths are immune to focus variation are generated and investigated.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
\textbf{Parameters} & \textbf{Values} \\
\hline
\textlambda\text (wavelength) & 248nm \\
NA (Numerical Aperture) & 0.7 \\
Illumination type & Annular \\
Scattering bar width & 60nm \\
Scattering bar placement & 180nm \\
Linewidth (nominal) & 130nm \\
\hline
\end{tabular}
\caption{Parameters used in CalibreWB}
\end{table}

Figure 1: Linewidth variation with defocus level (nominal linewidth = 130nm).

\section{Compensating Focus-Dependent CD Variation}

Systematic variation can be mitigated to some extent by performing OPC and inserting assist features, but for various reasons (modeling errors, algorithmic inaccuracies, process variations, etc.) cannot be completely removed. The remaining linewidth variation due to layout is significant even after the use of complex RET techniques, with isolated and dense lines retaining opposite behavior under varying defocus\textsuperscript{6}. Thus, there is a possibility of compensating for systematic variation in the design itself. This compensation can be achieved in two ways.

\subsection{Self-compensated Cell Layout}

By \textit{self-compensated cell layout}, we refer to a correct-by-construction methodology that relies on within-cell compensation of CD variation caused by focus variation. For example, variation can be compensated in series-connected NMOS if one device becomes thinner (thus, faster) under defocus, and the other device becomes fatter (thus,
This can be achieved by making one device “iso” and the other device “dense”. The other way of generating self-compensated cells is to find spacing ranges in which the linewidth variation is negligible by focus variation. Each spacing between adjacent poly lines should be one of these values. In this work, we generate all the self-compensated cells by defining the both spacings of poly lines to be at compensated spacing ranges to the focus variation. We also propose the possibility of single pitched-cells in which all spacings between adjacent poly lines are equal to eliminate the focus-dependent CD variation inside cells.

2.2. Self-compensated Physical Design.

This refers to compensation across cells (e.g., in a critical path). Consider two cells G1 and G2 that lie on the critical path G1 → G2. Focus variation, if not corrected by applying expensive RETs, can cause variation in delay of the critical path and potential timing failures or parametric yield loss. However, if G1 is explicitly made “iso” while G2 is made “dense”, then focus-dependent CD variation can be compensated. Assuming that iso and dense versions of library cells are available, designs that are robust to focus variation become possible.

In this paper we compare and contrast the two approaches put forth above. For example, we seek to compare the area overheads of self-compensated libraries vs. across-cell optimizations. We generate each variant of cells from the observation of the lithography simulation, area overhead then acquired from the Place & Route (PR) step. Sensitivity-based heuristic optimization approach for the self-compensating design for timing and area aspect is proposed. Same heuristic algorithm can be applicable for the timing and leakage perspective. We also propose the possibility of using ILP technique to resolve the optimization problem.

The remainder of this paper is organized as follows. Section 3 describes the construction of a cell library that consists of each version of cells, section 4 describes self-compensating design techniques, and section 5 shows several optimization options in timing and leakage. We present experimental results in Section 6 and Section 7 provides conclusions.

3. LAYOUT GENERATION

The work which described in is based on the lithographic simulation results after OPC and SRAF insertion using Calibre WorkBench. Through-pitch and through-focus CD variation data are obtained from the five-line patterns. However, no layout generation processes for iso/dense/self-compensated versions of cells are performed. Area of each cell version and parasitics are estimated from the expected spacings. To confirm iso/dense/self-compensated vs. original layout comparisons and to get better estimates of delay and area after placement and routing, we propose to generate each version of cell by using an automatic layout generation tool.

3.1. Lithography Simulation

Lithography parameters which are used in Calibre WorkBench are shown in Table 1. Optical lithography process of 248nm wavelength with 0.7 numerical aperture is used. Optical models are generated at 5 different defocus levels (0.0, 0.1, 0.2, 0.3, 0.4um) and a constant threshold resist model is used. We assume that the optical characteristics are symmetric in defocus (i.e., +0.1um = -0.1um defocus).

3.2. CD Measurement

To find a specific spacing range for iso, dense, self-compensated, and single-pitched cells we perform lithography simulation after OPC and SRAF insertion at two defocus values: best focus and worst defocus. The resulting printed linewidth is then measured. $L_{\text{eff}}$ variation at the worst defocus value (i.e., 0.4um) is used to construct criteria for spacing range of each variant of cells (i.e., iso, dense, self-compensated, and single-pitched cells). The linewidth variation with spacing from 180nm to 660nm at 0.0um and 0.4um defocus level is shown in Figure 2. The 3D graph with different left and right spacing from 180nm to 630nm with 50nm step is shown in Figure 3. The tolerance of the self-compensated devices is set at 4nm since the 3σ for the gate CD control is 4nm in 130nm technology. The 1st scattering bar is inserted at the spacing of 420nm, and the 2nd scattering bar is inserted at the spacing of 660nm. Therefore, we define the spacing for dense devices to be 180nm (minimum spacing), 420nm (1st scattering bar point), and 660nm (2nd scattering bar point). We define the intervals 380 ~ 410nm and 600 ~ 650nm as iso spacing regions; and 260 ~ 320nm and 460 ~ 480nm as self-compensated spacing regions. Finally, we select 480nm as the spacing value for single-pitched cells from the self-compensated regions, because the minimum spacing for contacts is 420nm. Table 2 summarizes the spacing criteria for cell generation. We can set our intended spacing of poly gates within technology files (ProTech) to make
each desired version of the library cells. A lumped-C model of capacitance is extracted and added into netlists to obtain more exact timing; to this end, we use a commercial parasitic extraction tool (*CalibrePEX*)\(^{10}\).

To analyze iso/dense/self-compensated behavior with defocus, we use five-line patterns and sweep the spacing between the three center lines from 180nm to 480nm. SRAF (scattering bar) insertion and OPC are performed on these patterns using *Calibre*\(^{10}\). The average linewidth of the center line is then measured for each pattern. Figure 1 shows the variation in the critical dimension (CD) for different spacing values at nine different defocus values (-0.4μm to +0.4μm). In our study 0.0μm indicates in-focus condition and 0.4μm is the worst-case defocus level. The figure shows distinct spacing ranges where the patterns behave as iso, dense or self-compensated.

Based on Figure 1 and Figure 3, we generate a lookup table (LUT) using the function \( CD = f(LS, RS, F) \), where \( LS \) is the left spacing, \( RS \) is the right spacing, and \( F \) is defocus. This allows us to obtain the exact degree to which specific patterns act isolated, dense, or self-compensated, and also to predict CD given defocus and spacings. The tolerance of the self-compensated devices is set at 4nm. Thus, if linewidths are 4nm larger than nominal at 0.4μm defocus, we classify those patterns as “dense”; similarly, if linewidths are 4nm smaller than nominal, we classify the patterns as “iso”. Finally, if the CD variation is less than 4nm at 0.4μm defocus, we consider the pattern “self-compensated”. The first scattering bar insertion point is at a spacing of 420nm, therefore, the “most-iso” pattern has a spacing of roughly 400nm. At 420nm spacing and above, the pattern reverts to “dense” behavior as a result of scattering bar insertion. At the “most-dense” spacing (i.e., 180nm on each side), the linewidth increases 13% from nominal and in the “most-iso” case (i.e., 400nm on each side), the linewidth decreases 11% from nominal at the 0.4μm defocus point.

The optimal scattering bar placement and width depend on numerous factors such as wavelength (\( \lambda \)), numerical aperture (NA), illumination type, and others\(^{12}\). A reference\(^{13}\) provides equations for optimal size and placement (defined as SRAF to main pattern spacing) of scattering bars, which are \((0.2 \sim 0.25)\*\(\lambda/NA\)) and \((0.55 \sim 0.75)\*\(\lambda/NA\)) respectively.

![Figure 2: Linewidth variation with spacing (note that SRAFs are inserted at 420nm and 660nm spacings)](image)

![Figure 3: Linewidth variation with asymmetric spacing at two defocus values (note that the flat surface is for 0.0μm defocus value)](image)

<table>
<thead>
<tr>
<th><strong>cell version</strong></th>
<th><strong>Spacing range (nm)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>dense</td>
<td>180(min.), 420(1sb), 660(2sb)</td>
</tr>
<tr>
<td>iso</td>
<td>380 ~ 410, 600 ~ 650</td>
</tr>
<tr>
<td>self-compensated</td>
<td>260 ~ 320, 460 ~ 480</td>
</tr>
<tr>
<td>single-pitched</td>
<td>480</td>
</tr>
</tbody>
</table>
3.3. Edge Devices

Special consideration is required for the edge devices. Edge devices are the devices that are closest to the cell boundary. For example, since there is only one poly line for NMOS and PMOS in an INVX1 (minimum sized inverter) layout, these are all edge devices. We identify two different types of edge devices: case1 has no neighboring devices on either side (e.g., INVX1), while case2 has no neighboring device on exactly one side (e.g., left-most or right-most devices in cells except INVX1 and INVX2 which have no fingers). To investigate the edge effect in case1, we first sweep the spacing from 180nm to 1um symmetrically on both sides. Figure 4 shows linewidth versus spacing in case1. As can be seen from the graph, linewidth is insensitive to focus after two SRAFs are inserted on each side of poly line. For case2 edge devices, we fix one side at 180nm for dense and 380nm for iso devices. The spacing on the other side is swept up to 2um. Figure 5 shows the case2 edge effect of dense and iso cells respectively. When two adjacent poly lines are 1.2um apart (i.e., 2 SRAFs are inserted at each side), the linewidth does not vary much even if the spacing becomes larger. Since the distance from edge devices to the cell boundary for all cells is over 600nm in this technology (making the distance of two neighboring poly lines more than 1.2um), we assume that all edge devices in case2 follow the behavior seen in Figure 5.

3.4. Library Construction

The spacing between each poly line can be divided into 3 different ranges from the observation of the lithography simulations. Specific spacing values are used to generate each layout variants of the cells. An automatic layout generation tool is used to draw actual layouts in which all the spacings between poly lines are fixed to the values of each category. From the range of self-compensated spacing, one spacing value for which $L_{\text{eff}}$ variation is negligible is selected for single-pitched cells.

We consider 21 frequently used cells (INV: x1, x2, x6, x8, x12; NAND2(3) and NOR2(3): x1, x2, x4, x6). All 5 (original, dense, iso, self-compensated, and single-pitched) variants of each cell are generated. The original version is the layout generated without any spacing constraints, using the smallest possible area. The single-pitched version is generated with one fixed spacing value, using the linewidth that does not vary at the worst defocus value. Cell height of the entire layout is set to 4.2um. Table 3 shows the average area overhead comparison between the actual layout and the estimated area from $3\sigma$. As can be seen from the table, the two approaches show similar values. Self-compensated cell variants compared to the original show ~10% area increase on average over all cells. Single-pitched cells result in larger area increase than their self-compensated counterparts.
The LUT that is constructed from Figure 1 gives CD for each variant of the cells at two defocus values. Timing and power characterization is done using Synopsys Star-MTB\textsuperscript{25} on extracted netlists to generate \texttt{.lib} models for each layout version of cells.

<table>
<thead>
<tr>
<th>cell version</th>
<th>layout</th>
<th>estimated</th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>dense</td>
<td>1.02</td>
<td>1.04</td>
</tr>
<tr>
<td>iso</td>
<td>1.22</td>
<td>1.20</td>
</tr>
<tr>
<td>self-compensated</td>
<td>1.10</td>
<td>1.10</td>
</tr>
<tr>
<td>single-pitched</td>
<td>1.35</td>
<td>NA</td>
</tr>
</tbody>
</table>

### 4. SELF-COMПENSATING DESIGN

#### 4.1. Self-compensated Cell Design

By observing the lithography simulation with focus variation, we find the spacings where the linewidth vary within a certain limit (i.e., 3$\sigma$). Self-compensated devices are constructed within a cell by setting all the spacings between poly lines within a certain range. From the graphs in Figure 1 and Figure 2, the self-compensating spacing ranges are set at 260nm to 320nm and 460nm to 480nm. Self-compensated cells having these spacings on both sides are generated.

#### 4.2. Single-pitched Cell Design

Single-pitched devices are promising because of better manufacturability\textsuperscript{8,9}. We select one spacing value within the range of self-compensated spacings to generate single-pitched cells. Amongst the self-compensated spacings we select 480nm as the spacing for single-pitched cells because the minimum spacing required for contacts is 420nm. We generate a set of single-pitched cells by fixing all the spacings in the cells at this value.

### 5. OPTIMIZATION (SELF-COMПENSATED PHYSICAL DESIGN)

As can be seen from the previous section, a more robust design to focus variation is possible by using either self-compensated cells or single-pitched cells. Another option is to generate optimized circuits using both iso and dense cells to meet timing at all focus points. Optimization with a combination of iso and dense is possible both in timing constraints (i.e., critical delay) and power (i.e., leakage). It is implemented and compared both in heuristic and Mixed-Integer Linear Programming (MILP) solution methods.

#### 5.1. Timing and Area

The first optimization is possible in terms of timing and area. We can generate new circuits which meet timing requirement through all defocus values with a small increase in area having iso and dense in it. This optimization was discussed at length in our previous work\textsuperscript{21}. We reproduce some of that discussion below for the sake of completeness.

##### 5.1.1. Heuristic Algorithm

This problem of iso-dense self-compensating physical design can be solved as a sizing problem. Since dense cells are slower (at worst-case focus) and smaller iso cells are faster and bigger, we start with the circuit initially synthesized with dense cells, then swap in iso versions to meet timing at the worst-case defocus level.

Initially, synthesis with the “dense” library results in the slowest timing at worst defocus conditions with small area. The optimization of delay versus area is implemented using a sensitivity-based approach to minimize area penalty while instantiating “iso” counterparts of “dense” cells in the circuit to meet timing constraints. In our experiments, the required time at the primary outputs is set to be the worst-case delay with the original library at 0.0um defocus. The sensitivity of all gates with respect to a change from “dense” to “iso” variants can be defined as\textsuperscript{17}:
\[ \text{Sensitivity} = \frac{1}{\Delta A + K_1 \sum_{\text{arcs}} \text{slack}_{\text{arc}} - S_{\text{min}} + K_2} \Delta D \]  

(1)

Where \( \Delta A \) is the change in area and \( \Delta D \) is the change in delay due to swapping “dense” with “iso”. \( S_{\text{min}} \) is the worst slack in the circuit when synthesized using the “dense” library, and the \( \text{arcs} \) consist of all rise and fall transitions from each input to output of the gate. The term \( \text{slack}_{\text{arc}} \) is the difference between arrival and required times of the timing arc, and \( K_1 \) and \( K_2 \) are small positive numbers to ensure stability of the equation. Pseudocode for our optimization process is as follows:

```plaintext
Optimization {
    Input: focus, Output: optimized circuits
    While worst_slack is negative
        Calculate sensitivities of all gates in the circuit
        Sort sensitivities in non-increasing order
        Swap the “dense” version with “iso” cell, based on the order of sensitivities
        Calculate new_delay of circuit
        Update worst_slack
    }

Post-processing {
    If worst_slack at intermediate defocus value is not negative
        Finish optimization and exit
    Else
        Find the maximum-delay defocus point
        Perform optimization at the maximum-delay defocus point
    }
```

As the pseudocode indicates, we first sort sensitivities in non-increasing order. The gate with maximum sensitivity is then swapped with its corresponding “iso” version. Incremental timing analysis updates the \( \text{worst_slack} \) value and new sensitivities are then calculated if the timing is not met. Figure 6 shows an example of the swapping. The numbers in the illustration represent the sensitivity of each gate when changing from dense to iso counterparts. In this case, the highest sensitivity is 20, hence we swap the gate with an iso version with a small area penalty. Since all gates are dense at first, the design may not meet timing at worst-case defocus. Changing from dense to iso will compensate for the focus along critical paths. The optimization iterates this swapping until timing constraints are met.

Even after the above optimization procedure (which ensures timing correctness at both best and worst focus conditions), the circuit may not meet timing constraints at intermediate values of focus since delay variation with focus may be highly nonlinear and even non-monotone. Thus, the timing constraint needs to be checked across the defocus range. We sweep the optimized circuits over all defocus values to find the maximum-delay point. If the maximum-delay defocus point is out of the permissible focus range or the maximum delay is less than the required time, no more steps are needed. However, if the maximum-delay defocus point is within the permissible focus range, a post-processing step as described above is required to globally meet the timing constraint. At the maximum-delay defocus point, we can apply the same sensitivity-based optimization process shown above to ensure that the optimized circuit meets timing throughout the expected defocus range. Delay at intermediate focus values (e.g., 0.11, 0.24, 0.37\text{um}, etc.) is calculated by interpolation from pre-characterized cell delays at a small set of focus values (e.g., 0.1, 0.2, 0.3\text{um}, etc.). In the interpolation, we assume CD to be a quadratic function of focus. Also we assume that cell delay is a linear function of gate length for small perturbations of gate length.

![Figure 6: Illustration of optimization process (D denotes "dense" and I denotes "iso" cell; numbers are example sensitivities of gates to swapping to “iso” counterparts)](image-url)
5.1.2. Mixed-Integer Linear Programming (MILP)

Sensitivity-based heuristic optimization using iso and dense cells to make designs self-compensating results in good solutions; however, post-processing steps are indispensable for the compensation to be valid throughout the defocus range (i.e., in-focus through ~ 0.4um DOF range). Because of the nonlinearity of delay (or CD) and focus relations, optimization solutions should guarantee the compensation not only at extreme defocus values but also at in-between values.

To include the post-processing step into the optimization phase, a new optimization approach that exploits integer linear programming is proposed. For each gate \( i \), the area of component \( i \) is \( A_i \), \( P \) is the set of all possible paths and \( n \) is the number of gates in circuits. The problem of minimizing total area subject to maximum delay bound (i.e., required time) can be formulated as\(^{19} \):

\[
\begin{align*}
\text{Minimize} & \quad \sum_{i=1}^{n} A_i \\
\text{Subject to} & \quad \sum_{p} D_i \leq D_{\text{max}} \quad \forall p \in P \\
& \quad A_i \in \{A_i^{\text{dense}}, A_i^{\text{iso}}\} ; i = 1, \ldots, n
\end{align*}
\]

The number of possible paths from primary inputs to primary outputs should be exponential in \( n \). Therefore, transforming the constraints on path delay into constraints on delay across components (e.g., arrival time) is widely accepted as the practical technique. \( a_i \) represents the arrival time at each node \( i \). \( D_{\text{max}} \) means the maximum delay bound (e.g., required time of the circuits). To include the delay variation with defocus we discretize the defocus into 5 levels (i.e., 0.0, 0.1, 0.2, 0.3, 0.4um). The ILP problem can then be rewritten by:

\[
\begin{align*}
\text{Minimize} & \quad \sum_{i=1}^{n} A_i \\
\text{Subject to} & \quad a_{i,f} \leq D_{\text{max}} \quad ; j \in \text{outputs} \\
& \quad a_{i,f} + D_{i,f} \leq a_{i,f} \quad ; i = 1, \ldots, n \text{ and } \forall j \in \text{input}(i) \\
& \quad D_{i,f} \leq a_{i,f} \quad ; i = n+1, \ldots, n+s : \text{inputs} \\
& \quad A_i \in \{A_i^{\text{dense}}, A_i^{\text{iso}}\} \quad ; i = 1, \ldots, n \\
& \quad f \in \{0.0,0.1,0.2,0.3,0.4\} : \text{defocus}
\end{align*}
\]

Where \( a_{i,f} \) is the arrival time of gate \( i \) at \( f \) defocus level, and \( D_{i,f} \) represents the gate delay of \( i \) component at \( f \) defocus level. Finally if we have two choices (i.e., dense and iso) of gates and want to optimize the problem at all defocus values, the problem can be transformed into integer (binary) linear optimization problem:

\[
\begin{align*}
\text{Minimize} & \quad \sum_{i=1}^{n} [A_i^{\text{dense}} (1-x_i) + A_i^{\text{iso}} (x_i)] \\
\text{Subject to} & \quad a_{i,f} \leq D_{\text{max}} \quad ; j \in \text{outputs} \\
& \quad a_{i,f} + [D_{i,f}^{\text{dense}} (1-x_i) + D_{i,f}^{\text{iso}} (x_i)] \leq a_{i,f} \quad ; i = 1, \ldots, n \text{ and } \forall j \in \text{input}(i) \\
& \quad D_{i,f}^{\text{dense}} (1-x_i) + D_{i,f}^{\text{iso}} (x_i) \leq a_{i,f} \quad ; i = n+1, \ldots, n+s : \text{inputs} \\
& \quad x_i \in \{0,1\} : \text{binary} \quad ; i = 1, \ldots, n (0 = \text{dense}, 1 = \text{iso}) \\
& \quad f \in \{0.0, 0.1, 0.2, 0.3, 0.4\} : \text{defocus}
\end{align*}
\]

The integer (binary) linear problems then feed into a commercial solver to obtain optimum solutions. The mixed-integer optimizer of \textit{CPLEX}\textsuperscript{20} is used to solve these instances.
5.2. Timing with Leakage

Leakage is very sensitive to the linewidth variation. A through-focus leakage optimization can hence significantly improve power-limited yield. A new sensitivity which includes the leakage variation when we swap a dense cell with a iso counterpart can be formulated as Eq. (5). As can be seen in the Bossung plot (Figure 1), the linewidth of dense cells increase as defocus level increases, hence they have less leakage, on the other hand, the linewidth of iso cells decrease as the out-of-focus level increases thus they have higher leakage. The same heuristic algorithm is applicable with this new sensitivity:

\[
Sensitivity = \frac{1}{\Delta \text{Leak}} \sum_{\text{slack}_{\text{arc}}} \frac{\Delta D}{S_{\text{min}} + K_z}
\]  

(5)

Where \( \Delta \text{Leak} \) is the leakage variation from dense cells to iso counterparts at the worst defocus condition.

6. RESULTS

Table 4 summarizes the normalized delay and area overhead of each of the ISCAS85 benchmark circuits with various libraries. As can be seen from the table, the original library shows 13% slow-down in timing at 0.4um defocus because the cells are more dense-like. On the other hand, delay decreases by 16% on average with the iso library. Both self-compensated and single-pitched cells represent the robustness in defocus. The normalized leakage power information is shown in the right side of the Table 4. As expected, the original and dense library at the worst defocus value have 50% less leakage than the original library at perfect focus, since the linewidth becomes larger. On the other hand, leakage power with iso cells increases 3 times more than original cells at 0.4um defocus value. It is clearly seen that the leakage power variation in both self-compensated and single-pitched cells is within 6%.

The normalized area overhead for each of the cell versions and proposed optimization approaches are shown in Table 5. The gate distribution and run time of these optimization options are shown in the right side of the table. Heu1 means the heuristic optimization of timing and area, heu2 represents the heuristic optimization of timing and leakage. Even if self-compensated and single-pitched libraries provide good solutions for compensating designs for defocus, the area overhead is 11% and 27% respectively. ILP optimization shows the best results in area overhead because it gives optimal solution. Two sensitivity-based heuristics show approximately 4% area increase while meeting timing requirement throughout the defocus range. The gate distribution and run time of those two optimizations are shown in the right side of the table. As can be seen, ILP optimization technique shows less iso cells swapping than the other two heuristic optimizations. As anticipated, heuristic optimization with leakage power (i.e., heu2) results in less iso cell swapping than heuristic with timing and area(i.e., heu1) because iso cells have higher leakage than dense cells. However heuristic with leakage power shows slightly larger area increase from the original cells. As can be seen from the run time of heuristics and ILP optimization, the heuristic technique shows much faster execution time with reasonable solutions.

To illustrate the difference between heuristic and ILP optimizations, the slack vs. defocus for test case c7552 is shown in Figure 7. As can be seen from the graph, the original circuit fails to meet the required time in out-focus, but both heuristic and ILP optimization solutions can meet timing requirement (i.e., positive slack) in all defocus range. In heuristic optimization, timing requirement is met both at in-focus and at the extreme defocus (i.e., 0.4um) after optimization. However the timing failures happen in some points of defocus because of the nonlinearity of delay by defocus. Post-processing can handle the problems and guarantee the positive slack in all defocus range.

Monte-Carlo simulation with 1000 trials is used to investigate the impact of focus variation on delay distribution. A normal distribution of focus with mean = 0.0um and \( 3\sigma = 0.4um \) is assumed. Figure 8 shows 1000 Monte-Carlo simulation results for the c6288 circuit. Self-compensated, single-pitched, and two dense + iso optimization options (i.e., area and leakage) meet timing requirement at all randomly chosen defocus points.

Table 6 shows the leakage power variation at the worst defocus conditions with several self-compensating design options. As can be seen, both self-compensated cells and single-pitched cells designs options shows ~7% leakage increase at 0.4um defocus value. And dense with iso cells optimization with timing and area shows 10% less leakage than the original at 0.4um defocus. As expected, the optimization of timing and leakage shows 25% less leakage than original and 15% less than timing and area optimization option as there are more dense cells in the circuits, leading to a decrease in leakage when out-of-focus level increases.
A novel design technique to compensate for CD variation with lithographic focus variation is proposed in this paper. Given self-compensated cells, which modify devices to cancel expected focus-dependent CD variation, designs that are much more robust to focus variation become possible. Compensated design for CD variation caused by focus variation is achievable with small area penalties, assuming that iso and dense counterpart cells are also available. We observe that with iso and dense library options we can achieve a compensated design with ~3% area overhead compared to ~11% and ~27% in a self-compensated and single-pitched library based design respectively. Self-compensated cells design shows ~5% leakage increase over original layout at 0.4um defocus. 27% area penalty and ~6% leakage increase result at the worst-case defocus condition for designs with only single-pitched cells. Iso with dense optimization designs show 10% and 25% less power using self-compensated design with area and leakage objectives respectively. Our results are based on a 130nm technology; we believe that compensation in more advanced technologies such as 65nm is worth investigating as the impact is expected to be greater.
Table 4. Normalized delay and leakage power of each version of cells for ISCAS85 benchmark circuits

<table>
<thead>
<tr>
<th>benchmarks</th>
<th>original 0.4 defocus</th>
<th>normalized delay</th>
<th>normalized leakage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>original iso dense self-comp. single</td>
<td>original iso dense self-comp. single</td>
<td></td>
</tr>
<tr>
<td>c432</td>
<td>1.00 1.13 0.84 1.13 1.00 0.99</td>
<td>0.57 3.17 0.57 1.05 1.07</td>
<td></td>
</tr>
<tr>
<td>c499</td>
<td>1.00 1.12 0.83 1.13 1.00 0.99</td>
<td>0.58 3.17 0.57 1.06 1.07</td>
<td></td>
</tr>
<tr>
<td>c880</td>
<td>1.00 1.12 0.84 1.12 1.00 0.99</td>
<td>0.57 3.20 0.57 1.06 1.07</td>
<td></td>
</tr>
<tr>
<td>c1355</td>
<td>1.00 1.13 0.84 1.13 1.00 0.99</td>
<td>0.58 3.15 0.57 1.06 1.07</td>
<td></td>
</tr>
<tr>
<td>c1908</td>
<td>1.00 1.13 0.84 1.13 1.00 0.99</td>
<td>0.58 3.12 0.57 1.05 1.06</td>
<td></td>
</tr>
<tr>
<td>c2670</td>
<td>1.00 1.14 0.83 1.13 0.99 0.99</td>
<td>0.58 3.12 0.58 1.05 1.06</td>
<td></td>
</tr>
<tr>
<td>c3540</td>
<td>1.00 1.12 0.84 1.13 0.99 0.99</td>
<td>0.57 3.18 0.57 1.05 1.07</td>
<td></td>
</tr>
<tr>
<td>c5315</td>
<td>1.00 1.13 0.83 1.14 0.99 0.99</td>
<td>0.58 3.11 0.57 1.05 1.06</td>
<td></td>
</tr>
<tr>
<td>c6288</td>
<td>1.00 1.13 0.83 1.13 1.00 0.99</td>
<td>0.58 3.14 0.57 1.05 1.06</td>
<td></td>
</tr>
<tr>
<td>c7552</td>
<td>1.00 1.12 0.83 1.13 1.00 0.99</td>
<td>0.58 3.09 0.57 1.05 1.06</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>1.00 1.13 0.84 1.13 0.99 0.99</td>
<td>0.58 3.15 0.57 1.05 1.06</td>
<td></td>
</tr>
</tbody>
</table>

Table 5. Normalized area and gate distribution of each library and optimization results

<table>
<thead>
<tr>
<th>benchmarks</th>
<th>Total # of gates</th>
<th>normalized area</th>
<th>gate distribution</th>
<th>run_time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>orig. dense iso self-comp. single</td>
<td>heuristic1 (area) heuristic2 (leakage) ILP</td>
<td>dense iso dense iso dense iso</td>
<td></td>
</tr>
<tr>
<td></td>
<td>heu1 (area) heu2 (leakage)</td>
<td>heur. ILP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c432</td>
<td>339 1.00 1.02 1.17 1.12 1.26 1.09 1.09 1.08</td>
<td>233 106 318 21 317 22</td>
<td>0.04 0.19</td>
<td></td>
</tr>
<tr>
<td>c499</td>
<td>682 1.00 1.00 1.17 1.11 1.27 1.00 1.02 1.00</td>
<td>581 101 569 113 584 98</td>
<td>0.09 1.70</td>
<td></td>
</tr>
<tr>
<td>c880</td>
<td>575 1.00 1.02 1.18 1.11 1.27 1.02 1.02 1.01</td>
<td>560 15 561 14 562 13</td>
<td>0.07 0.35</td>
<td></td>
</tr>
<tr>
<td>c1355</td>
<td>680 1.00 1.00 1.17 1.11 1.27 1.05 1.08 1.04</td>
<td>536 144 516 164 564 116</td>
<td>0.39 11.21</td>
<td></td>
</tr>
<tr>
<td>c1908</td>
<td>645 1.00 1.01 1.16 1.12 1.26 1.04 1.05 1.04</td>
<td>554 91 584 61 566 79</td>
<td>0.08 13.79</td>
<td></td>
</tr>
<tr>
<td>c2670</td>
<td>1040 1.00 1.01 1.15 1.11 1.25 1.05 1.05 1.04</td>
<td>1017 23 1020 20 1010 30</td>
<td>0.20 11.61</td>
<td></td>
</tr>
<tr>
<td>c3540</td>
<td>1313 1.00 1.01 1.17 1.10 1.27 1.01 1.01 1.01</td>
<td>1279 34 1287 26 1280 33</td>
<td>0.32 27.28</td>
<td></td>
</tr>
<tr>
<td>c5315</td>
<td>2028 1.00 1.00 1.16 1.11 1.27 1.01 1.01 1.00</td>
<td>1490 538 1978 50 1981 47</td>
<td>1.51 29.30</td>
<td></td>
</tr>
<tr>
<td>c6288</td>
<td>4102 1.00 1.00 1.16 1.11 1.26 1.06 1.07 1.05</td>
<td>3631 471 3820 282 3693 409 7.80 913.32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c7552</td>
<td>2700 1.00 1.00 1.15 1.11 1.25 1.01 1.01 1.00</td>
<td>2610 90 2658 42 2648 52</td>
<td>2.02 358.03</td>
<td></td>
</tr>
<tr>
<td>average</td>
<td>1.00 1.01 1.16 1.11 1.27 1.03 1.04 1.02</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6. Leakage power variation of self-compensating design and two heuristic optimization results at 0.4um defocus value

<table>
<thead>
<tr>
<th>at 0.4 defocus</th>
<th>c432</th>
<th>c499</th>
<th>c880</th>
<th>c1355</th>
<th>c1908</th>
<th>c2670</th>
<th>c3540</th>
<th>c5315</th>
<th>c6288</th>
<th>c7552</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>self-comp.</td>
<td>5.08%</td>
<td>5.70%</td>
<td>5.51%</td>
<td>5.60%</td>
<td>5.46%</td>
<td>5.03%</td>
<td>5.48%</td>
<td>5.48%</td>
<td>5.25%</td>
<td>5.37%</td>
<td>5.40%</td>
</tr>
<tr>
<td>single-pitched</td>
<td>6.55%</td>
<td>6.93%</td>
<td>7.02%</td>
<td>6.70%</td>
<td>6.23%</td>
<td>5.94%</td>
<td>6.82%</td>
<td>6.20%</td>
<td>6.41%</td>
<td>5.91%</td>
<td>6.47%</td>
</tr>
<tr>
<td>heu1 (area)</td>
<td>31.41%</td>
<td>-4.77%</td>
<td>-36.30%</td>
<td>11.32%</td>
<td>-6.45%</td>
<td>-36.54%</td>
<td>-34.51%</td>
<td>17.43%</td>
<td>-10.47%</td>
<td>-33.57%</td>
<td>-10.25%</td>
</tr>
<tr>
<td>heu2 (leakage)</td>
<td>-25.65%</td>
<td>0.55%</td>
<td>-36.89%</td>
<td>14.80%</td>
<td>-22.21%</td>
<td>-37.58%</td>
<td>-37.16%</td>
<td>-36.57%</td>
<td>-25.94%</td>
<td>-39.04%</td>
<td>-24.57%</td>
</tr>
</tbody>
</table>
REFERENCES