Measurement and optimization of electrical process window

Tuck-Boon Chan
Abde Ali Kagalwalla
Puneet Gupta
University of California
Electrical Engineering Department
Los Angeles, California 90095
E-mail: tuckie@ucla.edu

Abstract. A process window is a collection of values of process parameters that allow a circuit to be printed and to operate under desired specifications. A conventional process window, which is determined through geometrical fidelity, geometric process window (GPW), does not account for lithography effects on electrical metrics such as delay, static noise margin (SNM), and power. In contrast to GPW, this paper introduces an electrical process window (EPW) which accounts for electrical specifications. Process parameters are considered within EPW if the performance (delay, SNM, and leakage power) of printed circuit is within desired specifications. Our experiment results show that the area of EPW is 1.5 to 8× larger than that of GPW. This implies that even if a layout falls outside geometric tolerance, the electrical performance of the circuit may satisfy desired specifications. In addition to process window evaluation, we show that EPW can be enlarged by 10% on average using gate length biasing and V_{th} push. We also propose approximate methods to evaluate EPW, which can be used with little or no design information. Our results show that the proposed approximation method can estimate more than 70% of the area of reference EPW. We also propose a method to extract representative layouts for large designs which can then be used to evaluate a process window, thereby improving the runtime by 49%. *© 2011 Society of Photo-Optical Instrumentation Engineers (SPIE).* [DOI: 10.1117/1.3545822]

Subject terms: lithographic variation; electrical process window; geometrical process window; critical dimension; design for manufacturing.

Paper 10076PR received Jul. 19, 2010; revised manuscript received Dec. 21, 2010; accepted for publication Dec. 30, 2010; published online Feb. 25, 2011.

1 Introduction

The rapid pace of semiconductor scaling over the last decades, coupled with much slower advances in lithography technology, has forced 193-nm optical lithographic printing beyond its limit. Consequently, resolution enhancement techniques (RET) such as optical proximity correction (OPC), subresolution assist features, and phase-shift masks have become a necessity to ensure the printability of such small features.

Since OPC is typically performed at a nominal lithographic setup, it fails to account for variation in exposure, focus, or overlay. To compensate for these variations, process window (PW) OPC has been proposed in Ref. 1, whereby OPCs are performed at multiple process corners. This method is, however, impractical due to long runtime. Another method, image slope OPC[2] optimizes slope of intensity, which is a measure of variation in dose, along with edge placement error (EPE). Retargeting[3,4] is a rule-based technique to modify the layout before performing OPC to improve process window and is a popular approach in industry. Although these methods address the problem of lithographic variation, accurate metrics are required to quantify their benefits.

Process window is the range of process parameters such that designs produced within this range operate under desired specifications.[5] Typical process window checks if the critical dimension (CD) of any feature deviates from its nominal value by more than a predefined tolerance[5,6] and is denoted as geometric process window (GPW) in this paper. Although GPW is easy to compute or measure, it is not an accurate representation of electrical behavior of the printed circuit.

Recently, there has been some interest in reducing the pessimism due to poor correlation between design geometry and electrical performance. In Ref. 7, *electrically driven OPC* is developed based on nonrectangular transistor models for I_{on} and I_{off}. Zhang and van Adrichem[8] developed an analytical model to account for corner rounding in printed transistors and accounted for its impact on saturation current during OPC. Gupta et. al.[9] used timing slack of critical paths to reduce the complexity of post-OPC mask shapes. These methods achieve smaller performance variation and reduced mask complexity despite large geometric errors.[10] In Ref. 11, the authors propose a design-for-manufacturing methodology to compare the static noise margin (SNM) of 6T static random access memory (SRAM) cells printed under different defocus conditions. The method provides important feedbacks for designers at early design stage, which helps to reduce design and manufacturing costs.

Inspired by the above-mentioned approaches, we propose an electrical process window (EPW), which estimates PW based on delay, SNM, and leakage deviation instead of variation in CD. In this work, we focus on PW analysis for digital VLSI circuit which has a dense geometry pattern and is susceptible to lithography variation [we do not evaluate EPW for analog circuit because the layout of analog circuit is usually guardbanded with high

1932-5150/2011/$25.00 © 2011 SPIE

margin to account for lithography variations]. To evaluate EPW, we generate post-OPC lithography contours of a given layout at different exposure, defocus, and overlay (E/F/O) process points. Then, we extract transistor shapes and their electrical performances using the model in Ref. 12. Finally, EPW is defined by process points that yield lithography contours with acceptable electrical performances.

The key contributions of this work are as follows:

1. In contrast to the conventional GPW, we propose an electrical process window defined by delay, SNM, and leakage power of a design. EPW can reduce the pessimism in process control requirements as its area is 1.5 to 8× larger than that of GPW.
2. We demonstrate that EPW can be optimized by layout transparent methods such as gate length biasing and V_{th} push during manufacturing.
3. We propose several approximations to EPW for cases where design information is incomplete.
4. We present the concept of representative layout extraction which can be used to reduce EPW evaluation runtime.

The paper is organized as follows. Section 2 gives the precise definition of various methods of evaluating GPW and EPW. Section 3 describes our experimental setup and compares EPW against GPW. Section 4 demonstrates approaches to improve EPW and discusses their impact on EPW. Section 5 introduces approximations to EPW and Sec. 6 presents our representative layout-based approach to speed up EPW evaluation. Section 7 shows the experimental result of EPW including SRAM and Sec. 8 concludes our work.

2 Definition of Process Windows

In this work, we focus on analyzing the lithography process window for a polylayer because it usually is the most critical layer in lithography. Moreover, lithography variation on polylayer has a strong correlation to electrical variation as it defines transistor gate length.

2.1 *Geometric Process Window*

GPW is defined as the range of process parameters such that deviation between the CD of printed contour and circuit layout on polylayer (gate length) is within a predefined tolerance, i.e.,

$$(E_i, F_j, O_k) \in \text{GPW} \Longleftrightarrow \text{lower bound of allowed CD deviation} \leq \text{CD} \leq \text{upper bound of allowed CD deviation}. \quad (1)$$

In our experiments, CD deviation is estimated based on an EPE histogram of all transistor segments. As illustrated in Fig. 1, EPE is defined as the displacement between printed contour and layout segments. Since EPE only measures channel length deviation on one side of a transistor channel, the following scenarios are considered and CD is defined accordingly.

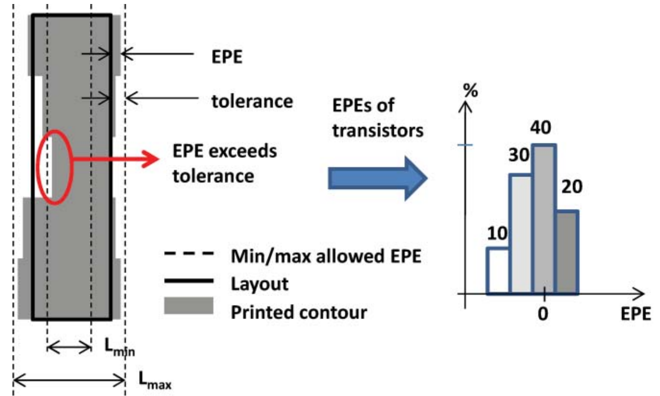

Fig. 1 Illustration of EPE histogram.

1. Maximum EPE occurs at both edges of a transistor segment. CD = nominal channel length ±2× maximum EPE (worst case).
2. Maximum EPE occurs at one edge of a transistor segment. We assume that the edge opposite the maximum EPE segment is not changed and CD = nominal channel length ± maximum EPE.

Based on the definitions for CD and GPW, we consider a process point (E_i, F_j, O_k) to be within GPW if more than 99% of EPEs are smaller than predefined CD tolerance. The 1% allowance is given to avoid pessimistic GPW due to EPE outliers, which can be fixed by a fine tuning mask in OPC. In subsequent sections, we use W-GPW to denote GPW with CD defined by scenario 1 (worst case) and A-GPW for GPW with CD defined by scenario 2.

2.2 *Electrical Process Window*

A process point (E_i, F_j, O_k) is considered within EPW if electrical performance of a printed circuit is within desired tolerance, i.e.,

$$(E_i, F_j, O_k) \in \text{EPW} \Longleftrightarrow \text{circuit performance lower bound} \leq \text{circuit performance} \leq \text{circuit performance upper bound}. \quad (2)$$

In this work, we demonstrate the evaluation of delay centric EPW (D-EPW), leakage power centric EPW (P-EPW), and SNM-EPW as they are commonly used electrical performance metrics. In Ref. 13, the impact of interconnect linewidth variation is found to be much smaller than the impact of transistor gate length variation on delay. Therefore, we do not consider interconnect linewidth variation in calculating EPWs.

2.2.1 *Delay centric electrical process window*

Due to subwavelength lithography, a printed transistor channel is not rectangular despite the use of aggressive RET techniques. This imposes difficulties in EPW extraction as electrical performance of a nonrectangular gate (NRG) transistor cannot be determined from a precharacterized library. To model the impact of NRG transistors on critical path

delay, we extract I_{on} of each NRG transistor using the method proposed in Ref. 12. As shown in Fig. 2, an NRG transistor obtained from simulated contour is sliced into narrower transistors to approximate the nonrectangular channel. Then, the effective channel length, width, and V_{th} of sliced transistors are extracted so that they can be represented as rectangular transistors [We use a SPICE-based method in Ref. 12 to calibrate parameters for an NRG transistor model]. Finally, the rectangular transistors are simulated using HSPICE[14] and their I_{on} and I_{off} are summed up to represent total I_{on} and I_{off} of the NRG transistor. After obtaining the current, the cell delay of NRG transistor is estimated by the following equation:

$$\text{Cell delay} = \frac{\sum_{j=1}^{N_i} I_{\text{on}-\text{original}-\text{j}}}{\sum_{j=1}^{N_i} I_{\text{on}-\text{simulated}-\text{j}}} \times \text{original cell delay},$$

where N_i is the total number of transistors in cell j and original cell delay is the delay of the cell specified in circuit's timing report. Subsequently, path delay of simulated contour ($D_{\text{path}-\text{simulated}}$) is represented as the sum of a delay of every cell along the path,

$$D_{\text{path}-\text{simulated}} = \sum_{i=1}^{M} (\text{Cell delay}_i), \tag{3}$$

where M is the total number of cells along a critical path. Finally, D-EPW is defined as

$$\begin{aligned}(E_i, F_j, O_k) \in \text{D-EPW} &\iff \max(\Delta D_{\text{path}}) \\ &\le \text{upper bound of allowed delay deviation.} \\ \Delta D_{\text{path}} &= \left[\frac{D_{\text{path}-\text{simulated}}}{D_{\text{path}-\text{original}}} - 1\right] \times 100\%,\end{aligned} \tag{4}$$

where $D_{\text{path}-\text{original}}$ is the delay of the critical path obtained from circuit's timing report.

2.2.2 *Leakage power centric electrical process window*

Leakage current of NRG transistors at different process points ($I_{\text{off}-\text{simulated}}$) are obtained using the method in Ref. 12. The method is also used for calculating the leakage current of each transistor in pre-OPC layout ($I_{\text{off}-\text{original}}$) to evaluate leakage power deviation of a circuit (Δpower).

$$\Delta\text{power} = \left[\frac{\sum_{j=1}^{T} I_{\text{off}-\text{simulated}-\text{j}}}{\sum_{j=1}^{T} I_{\text{off}-\text{original}-\text{j}}} - 1\right] \times 100\%, \tag{5}$$

where T denotes the total number of transistors in a design. Note that Eq. (5) does not account for cell topology. For example, a stacked transistor has less leakage power compared to nonstacked transistors. This leads to an estimation error whenever CD variations are different for stacked and nonstacked transistors. Since the P-EPW is a function of relative leakage power instead of the absolute value, the estimation error is only significant when stacked and nonstacked transistors have different CD variations. In other words, the estimation error is negligible if stack and non-stack transistors have similar CD distributions. For random digital logic, CD variation is affected by a surrounding pattern which has no direct correlation with cell topology. Therefore the estimation error due to cell topology is unlikely a major source of error.

Since there is no lower bound for leakage power, P-EPW is defined as

$$\begin{aligned}(E_i, F_j, O_k) \in \text{P-EPW} &\iff \Delta\text{power} \\ &\le \text{upper bound of allowed leakage power deviation.}\end{aligned} \tag{6}$$

2.2.3 *Signal noise margin electrical process window*

To capture the impact of lithography imperfection on a SRAM cell, we replace each NRG transistor in the cell by an equivalent transistor which has the same I_{on} as the NRG transistor. Since there can be many width and length combinations for a given I_{on}, we choose the equivalent transistor which has a channel width equal to the average width of the NRG transistor.

After obtaining the equivalent transistors for a SRAM cell, we run the spice simulation to get the voltage transfer curves of inverter pairs in a SRAM cell. We evaluate only a read noise margin, since it is typically more critical compared to hold or noise margin. The SNM of a cell is defined by the diagonal length of maximum square within butterfly curves as shown in Figure 3. Due to the regular layout of the SRAM array, the printed contours of each cell are similar. Therefore, we evaluate SNM-EPW based on the SNM value of a SRAM

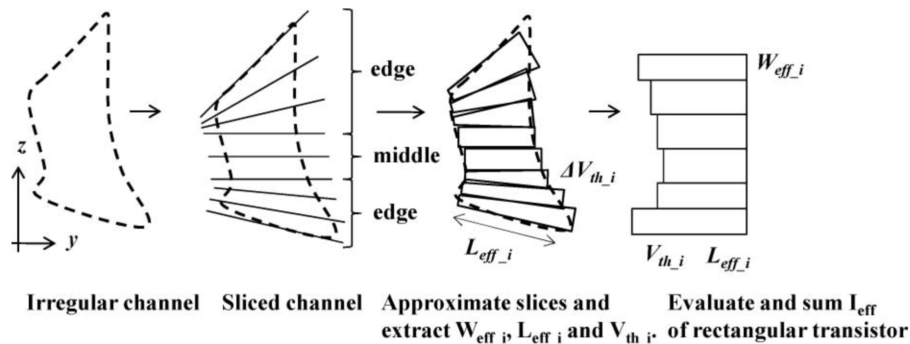

Fig. 2 Nonrectangular gate transistor I_{on} and I_{off} extraction.

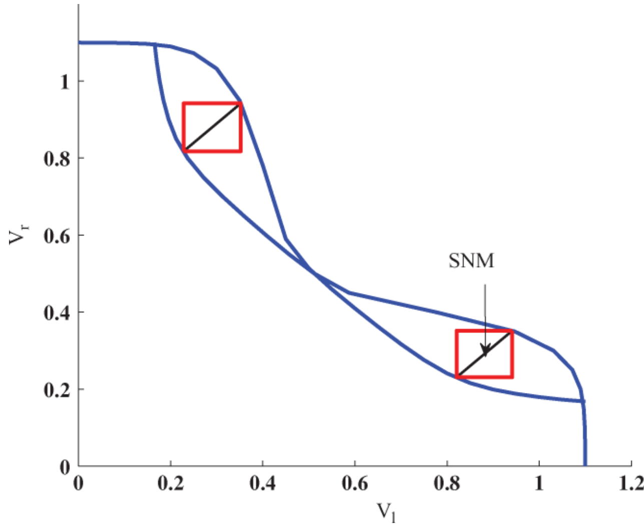

Fig. 3 SNM extraction based on voltage transfer curves of a 6T SRAM bit cell.

cell. SNM-EPW is defined as

$$(E_i, F_j, O_k) \in \text{SNM} - \text{EPW} \iff \Delta\text{SNM} \geq \text{lower bound of allowed signal noise margin deviation,} \tag{7}$$

$$\Delta\text{SNM} = \left[\frac{\text{SNM}_{\text{simulated}}}{\text{SNM}_{\text{original}}} - 1\right] \times 100\%. \tag{7}$$

2.2.4 *Combined electrical process window*

Whenever there are more than one electrical performance metrics, the combined electrical PW (C-EPW) can be easily computed by finding the intersections of the EPWs,

$$\text{C-EPW} = \bigcap_{i=1}^{Q} (\text{EPW}_i), \tag{8}$$

where Q is the total number of electrical performances. In this work, C-EPW is defined as the intersection between D-EPW and P-EPW.

2.3 *Relation Between GPW and EPW Tolerances*

Since GPWs and EPWs are defined differently, we need to figure out the relation between the two for fair comparison. To obtain the worse case corners of GPW, we simulate an inverter with four times fanout and a 6T SRAM cell at [nominal length $\pm$ (2 $\times$ EPE tolerance)](Vdd=1.1 V, Temperature = 25° C) using SPICE[14] and a transistor model provided by Nangate Open Cell Library.[15] The maximum delay, leakage power, and SNM deviations are extracted to represent D-EPW, P-EPW, and SNM-EPW tolerances, respectively. Table 1 summarizes the corresponding deviations in delay and leakage power for different EPE tolerances. For example, ±5% EPE (2.5 nm of 50 nm nominal channel length) corresponds to 11%, 54%, and –24% deviations in delay, power, and SNM, respectively. Hence, W-GPW with 2.5% EPE tolerance corresponds to A-GPW with 5% EPE tolerance, D-EPW with 11% delay tolerance, P-EPW with 54% leakage power tolerance, and SNM-EPW with –30% SNM tolerance.

When channel length deviates more than 10%, the SNM of a 6T SRAM cell reduces to zero. Therefore, the maximum allowed geometrical deviation is 10% for SRAM. The tolerance for leakage power is very high compared to channel length and EPE tolerances because leakage power increases exponentially as channel length decreases. Note that the tolerances in Table 1 are strongly dependent on the process technology.

3 Comparison Between GPW and EPW for Digital Logic

3.1 *Experimental Setup*

To show the differences between GPW and EPW for digital logic, five ISCAS-85 (Ref. 16) and a microprocessor (mips) benchmark[17] circuits were implemented using 45 nm Nangate Open Cell Library (PDK v1.2 v2008).[15] After synthesis, placement, and routing, we define the paths within 20% of setup time constraint as critical paths. The layouts of benchmark circuits were scaled to 65 nm for OPC and lithography simulation due to limitations in our optical models. After that, the simulated contours are scaled down to 45 nm for leakage and drive current extraction. To emulate variations in a lithography system, we simulate an image for polylayer with different exposure and defocus values using Mentor Calibre.[18] In this work, we only analyze the PW for polylayer. During EPW extraction, we use the active layer patterns in layout.

Overlay error is emulated by shifting a active layer along the vertical direction (Z direction in Fig. 2) during transistor shape extraction. Process parameters in our experiments are as follows:

Table 1 Tolerances of GPW and EPW.

ΔChannel length (%)	W-GPW Δ EPE (%)	A-GPW Δ EPE (%)	D-EPW Δ delay (%)	P-EPW Δ power (%)	SNM-EPW Δ SNM (%)
5	2.5	5	11	54	-24
10	5.0	10	21	311	-61
15	7.5	15	30	2476	N/A

Table 2 GPW and EPW area for ISCAS-85 benchmark circuits.

Tolerance %	W-GPW			A-GPW			D-EPW			P-EPW			C-EPW (delay, power)			Feasible area
	2.5	5	7.5	5	10	15	11	21	30	54	311	2476	(11,54)	(21,311)	(30,2476)	
c432	0	0	0	0	300	1276	1538	2086	2460	882	1720	2107	0	1086	1846	2760
c499	0	0	0	0	117	1375	1559	2105	2508	921	1718	2076	9	1103	1864	2760
c880	0	0	0	0	196	1278	1390	1956	2332	825	1464	1969	0	890	1770	2565
c1355	0	0	0	0	95	1313	1665	2204	2560	847	1569	2052	35	1052	1891	2760
c1908	0	0	0	0	139	1253	1388	1937	2309	841	1493	1988	1	900	1767	2565
mips	0	0	0	0	0	190	921	1209	1426	334	599	823	0	248	690	1590
average	0	0	0	0	141	1114	1410	1916	2266	775	1427	1836	7	880	1638	2500

- Exposure (%) $\in \{80, 90, 100, 110, 120\}$.
- Defocus (nm) $\in \{0, 40, 80, 160\}$.
- Overlay (nm) $\in \{-20, -10, 0, 10, 20\}$.

All process points for which any printed transistor is open or short are excluded from EPWs and GPWs. This defines the maximum feasible process window. To evaluate GPW, we generate the EPE histogram for each process point by comparing printed contours to original layout using Mentor Calibre.[18] To evaluate EPW, we translate the extracted channel shapes into an OpenAccess database.[19] After that, I_{on} and I_{off} of every transistor are extracted using the method in Ref. 12 to obtain deviations in delay and leakage power as mentioned in Sec. 2. The analysis of EPW (including NRG transistor current extraction) was implemented in C++ and the experiment was carried out on a 64 bit machine running at 2 GHz with 16 GB memory.

3.2 *Results*

Results in Table 2 show that W-GPW is very pessimistic as it has zero area for all tolerances. Compared to W-GPW, A-GPW has less constrained CD definition and larger PW as expected.

Figure 4 shows the scatter plots of W-GPW, A-GPW, D-EPW, P-EPW, and C-EPW for benchmark circuit c1908. Although the experiments are carried out for different E/F/O, the overlay axis is excluded in these plots because it is observed that the PW is insensitive to overlay for the layouts we have. To reduce lithography simulation runtime, we estimate delay, leakage power, and EPE values between sampled data points by interpolation. The experiment results for other circuits are not displayed but the area of the PWs are stated in Table 2. (The result of W-GPW is not included in Fig. 4 as its has zero area in all cases.) From Fig. 4,* we can clearly notice the area of A-GPW is smaller than the areas of EPWs with corresponding tolerances. This implies there are process points where printed circuit can meet electrical tolerances although its CD violates geometric constraints. GPW is a more pessimistic metric compared to EPW because:

1. GPW requires at least 99% EPE to be within tolerable range. In contrast, EPW only restricts the total power and delay of a circuit which is the average of deviation of each transistor segment. Therefore, some of the transistor segments can vary significantly but the entire transistor is still able to meet EPW tolerance due to the averaging.
2. All transistors are not equally important in EPW. For instance, delay constraints are applied only for transistors on critical paths instead of all transistors in a design.
3. Averaging across multiple transistors in a critical path for delay or all transistors for power.

It is observed that at 100% exposure and 80 nm defocus (circled in Fig. 4), A-GPW with 15% EPE tolerance is within tolerance (shaded) but P-EPW with corresponding leakage power tolerance is not. This happens whenever the actual channel length deviation (combined EPE on both edges) is larger than 7.5 nm (15% of channel length) but none of the EPEs exceeds 7.5 nm. As a result, the process point is considered valid in A-GPW but the actual leakage power is greater than predefined leakage power constraints. This example shows that A-GPW is generally pessimistic compared to EPW but it does not guarantee the electrical performance of circuit printed within its PW.

When both leakage power and delay are considered, C-EPW can be much smaller than D-EPW or P-EPW as shown in the fourth row in Fig. 4. C-EPW is valuable as it clearly defines the acceptable process range, ensuring the printed design can meet both delay and power requirements. In cases where A-GPW and C-EPW have comparable tolerances as mentioned in Table 1, the area of C-EPW is 1.5 to 8× larger than that of A-GPW.

*It is noticed that the ideal process point at 100% exposure and 0 nm defocus lies outside P-EPW at 54% tolerance. Meanwhile, process points at 90% exposure and 0 to 80-nm defocus meets the tightest delay and leakage power tolerance. We believe this is due to imperfect calibration of our OPC setup.

4 Optimization of Electrical Process Window

With EPW, the impact of process tuning on PW can be estimated from simulated contours. This enables fast and extensive exploration of process tuning approaches for maximizing PW. Since C-EPW is defined as the intersection of D-EPW and P-EPW, it is possible to improve C-EPW by increasing D-EPW or P-EPW. But any change in gate lengths or V_{th} has opposite effects on D-EPW and P-EPW. For example, P-EPW increases along with transistor gate lengths (leakage power reduced) but vice versa for D-EPW. Therefore, there is always a trade-off between D-EPW and P-EPW. As long as the sensitivities of P-EPW and D-EPW to the intentional gate length or V_{th} perturbation are different, they can be leveraged to improve C-EPW.

In this work, we assume $\pm$ 2 nm gate length biasing and $\pm$ 20 mV V_{th} push are allowed. To emulate gate length biasing, we adjust gate lengths of transistors during I_{on} and I_{off} extraction and the adjustment is conformal to the gates edges. Meanwhile the V_{th} push is implemented by adjusting the nominal V_{th} of each transistor during I_{on} and I_{off} extraction.

Figure 5 shows that reducing the gate lengths or lowering V_{th} enlarges D-EPW as expected. Meanwhile they reduce the area of P-EPW because total leakage power is increased when gate length or V_{th} of transistors are reduced. Since D-EPW only considers delay deviation on critical paths, reducing gate lengths on critical cells or all cells have an identical impact on D-EPW. For benchmark circuits *c*880 and *mips*, however, this is not true because one or more of the reduced gate lengths on noncritical cells in the circuits are smaller than the minimum acceptable gate length (30 nm). Any transistor smaller than this minimum gate length is considered as electrically shorted and it is a catastrophic circuit failure. As a result, the process points which print the shorted transistor are treated as not feasible points which reduce the D-EPW for circuit *c*880 and *mips*.

Alternatively, one can improve P-EPW by increasing gate length (non-critical or all cells) or V_{th} of transistors. Figure 5 shows that the approaches have similar improvements for P-EPW but the impacts of these approaches on D-EPW vary significantly. Since increasing gate length or V_{th} of all transistors also increases critical path delays, D-EPW of these approaches are smaller compared to D-EPW of optimization approach which increases gate length of noncritical cell only. There are cases (*c*880 and *mips*) where increasing gate lengths of noncritical cells have comparable

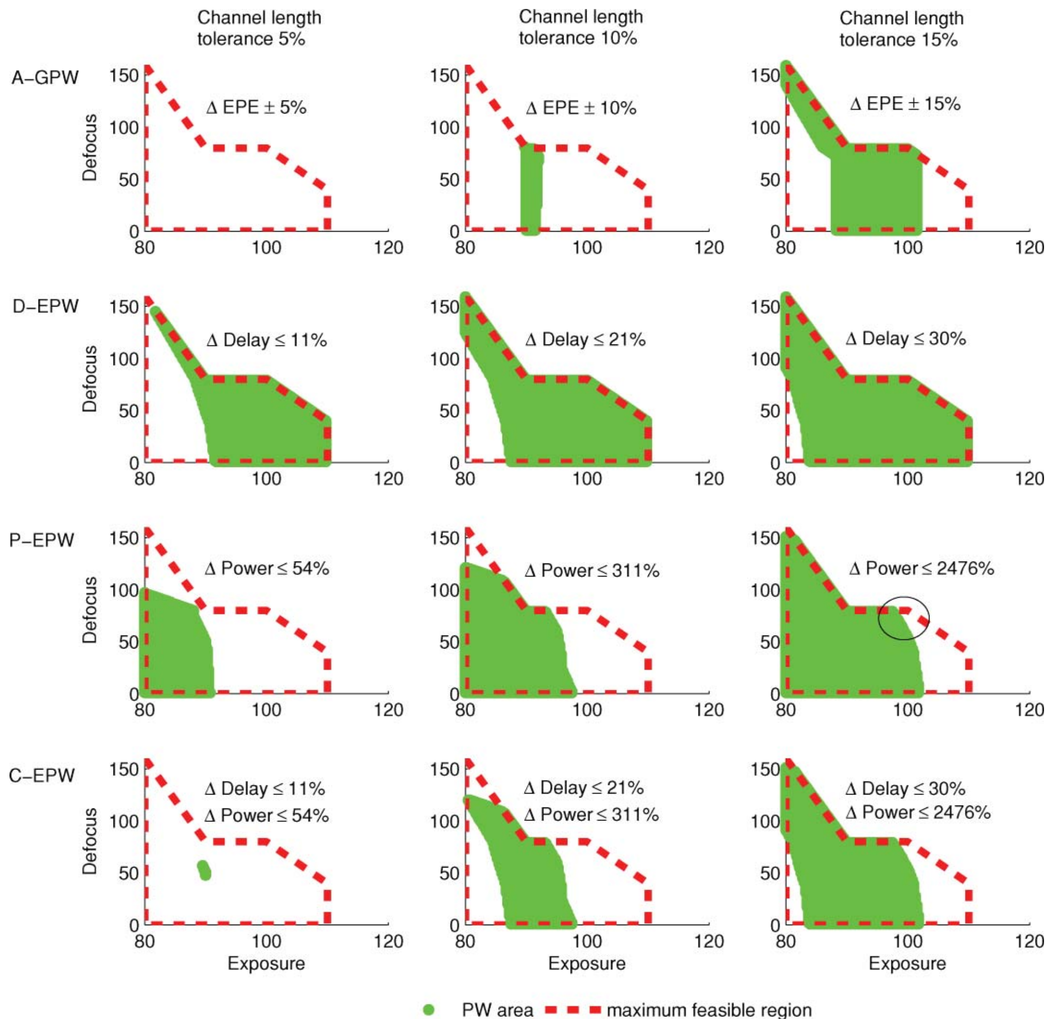

Fig. 4 Scatter plots of A-GPW, D-EPW, P-EPW and C-EPW for ISCAD-85 benchmark circuit c1908. Units for defocus and exposure are (nm) and (%).

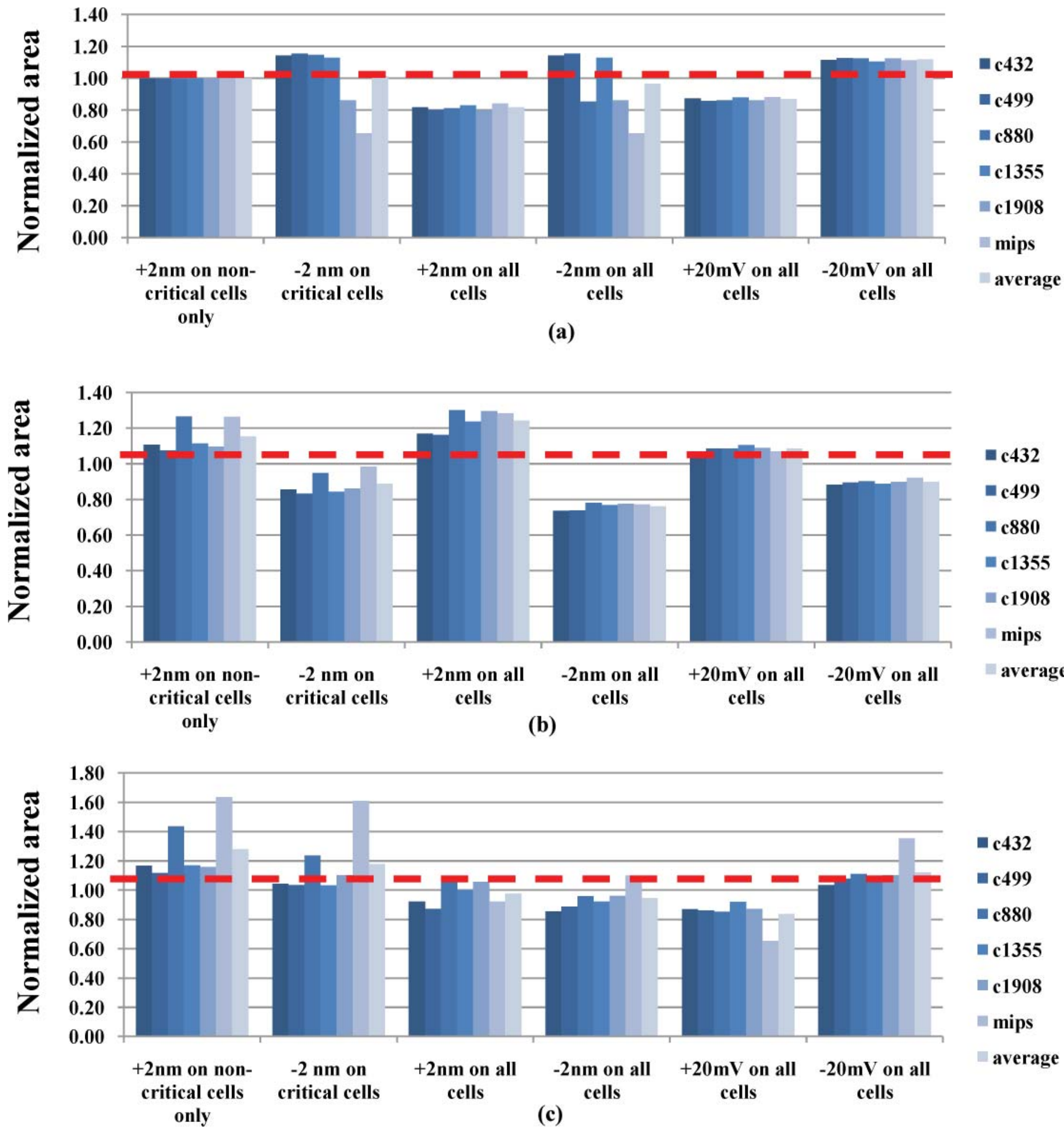

Fig. 5 Optimized EPW area normalized to unoptimized EPW area for (a) D-EPW, (b) P-EPW, and (c) C-EPW. Tolerances for delay and leakage power are 21% and 311%, respectively.

impact to increasing gate lengths of all cells because the number of critical cells is relatively small compared to the number of total cells as indicated in Table 3.

On average, biasing gate lengths selectively improves C-EPW while biasing gate lengths of all cells reduces the area of C-EPW. Besides, reducing V_{th} also improves C-EPW and vice versa for increasing V_{th}. Based on this analysis, reducing V_{th} seems to be a good approach in the absence of any design information, as it improves C-EPW consistently for all benchmark circuits. Moreover, it can be done without knowing the locations of critical cells.

5 EPW Approximations

In practice, critical paths of the design may not be available to the foundry. Instead of reverting to GPW, which is very pessimistic as already mentioned, we propose two methods to estimate EPW using purely geometric means.

Table 3 Ratio of critical cells to total cells in benchmark circuits.

Circuits	Critical cells/total cells
c432	50%
c499	24%
c880	16%
c1355	49%
c1908	26%
mips	3%
Average	24%

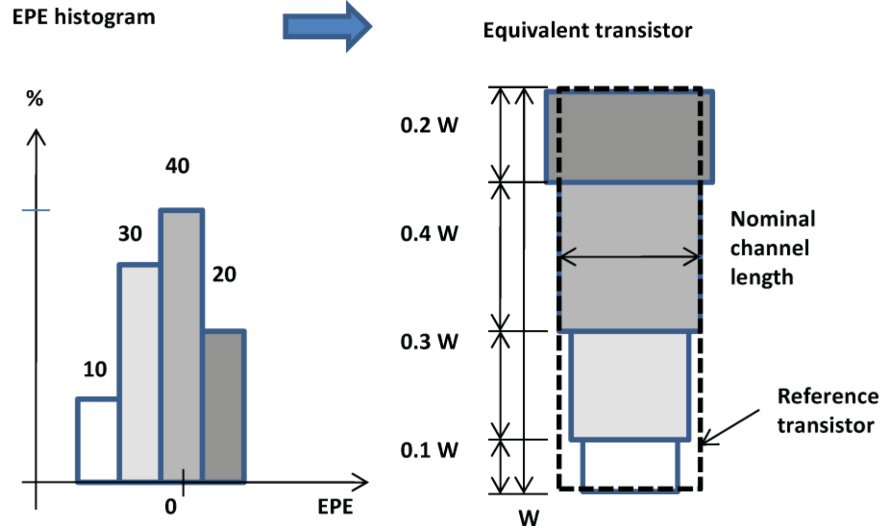

Fig. 6 Extracting equivalent transistor from EPE histogram.

5.1 *Method I: Use EPE Histogram Of Entire Design*

This method uses the EPE histogram generated during OPC to approximate EPW without extracting the channel shape of each transistor. We assume that average delay and leakage power deviation induced by EPEs of all transistors are approximately the same as that of an artificial *equivalent transistor* with the EPE histogram of an entire design. As illustrated in Fig. 6, based on the EPE histogram extracted for an entire design, each nonzero EPE bin is translated into a transistor edge which has the corresponding EPE. Consequently the channel width of each transistor segment is proportional to the percentage count[†] of its EPE bins.

Since EPE can happen on both sides of a transistor,

$$\text{channel length} = \text{nominal channel length} + 2\times\text{EPE (worst case}^{\ddagger}). \quad (9)$$

After constructing the *equivalent transistor*, its I_{on} and I_{off} can be estimated by the NRG current extraction method mentioned earlier. Note that the histogram is mainly constructed by the EPE of the middle part of the transistor channel. These transistor sections have uniform V_{th} as they are not affected by *narrow width effects* which happens at transistor edges. During NRG current extraction, we assign each segment in the equivalent transistor to their corresponding uniform V_{th} value. Therefore the extracted current is independent of ordering of slices.[§]

Since delay is inversely proportional to I_{on}, we estimate delay deviation as the ratio of I_{on} of a reference transistor to the calculated $I_{on-equivalent-transistor}$. As shown in Fig. 6, the reference transistor has nominal channel length and its total channel width is the same as the one of *equivalent transistor*. Meanwhile leakage power deviation is estimated by the ratio of $I_{off-equivalent-transistor}$ to the I_{off} of the reference because leakage power is proportional to I_{off}. The approximated EPWs are called *histogram-EPWs* in the remaining text and their definitions are given as follows:

$$(E_i, F_j, O_k) \in \text{histogram}-\text{D}-\text{EPW} \iff \left[\frac{I_{on-reference-transistor}}{I_{on-equivalent-transistor}} - 1\right]\times 100 \le \text{upper bound of allowed delay deviation}$$

$$(E_i, F_j, O_k) \in \text{histogramP}-\text{EPW} \iff \left[\frac{I_{off-equivalent-transistor}}{I_{off-reference-transistor}} - 1\right]\times 100 \le \text{upper bound of allowed power deviation}. \quad (10)$$

In our experimental setup, EPE histogram included edge displacement of PMOS and NMOS transistors together. To estimate transistor current correctly for static CMOS, the width ratio of PMOS and NMOS is taken into account when we calculate I_{on} and I_{off},

$$I = \frac{K\times I_{PMOS} + I_{NMOS}}{K+1},$$

where K is the ratio of PMOS to NMOS channel width. In our experiments, we use the average K across different logic cells in Nangate Open Cell library[15] which is ≈ 1.7.

5.2 *Method II: Use The Shape Of Every Transistor*

Given the shape of every transistor, as mentioned in previous sections, we can extract I_{on} and I_{off}. Thus, we can calculate P-EPW based on the definitions in Eq. (6) and no approximation is required. On the other hand, exact D-EPW cannot be determined as the information of critical cells is not available. Clearly, a strict D-EPW can be defined by the worst case delay variation of all transistors. But this definition is

[†] If all edge fragments are not of equal width, the histogram can be weighed appropriately.

[‡] Based on our experiment results, defining "channel length=nominal channel length + EPE" leads to over-optimistic approximations that cover a large area out of reference EPWs. Therefore, we assume a worst case EPE condition for this approximate method.

[§] We ignore the error due to V_{th} location dependency as it is not a major source of error compared to the simple approximations in EPW definitions. A better accuracy is possible by using complex layout extraction to keep track of edge versus center EPE.

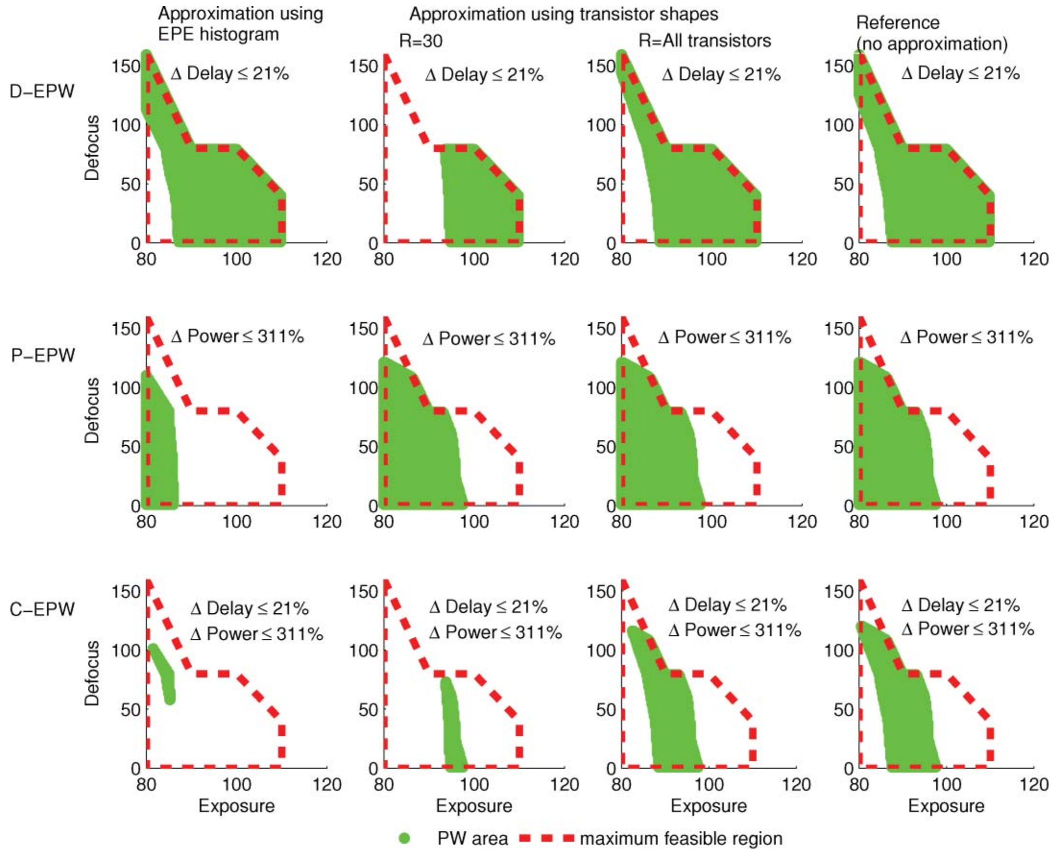

Fig. 7 Comparison between EPW and its approximations for benchmark circuit c1908. Units for defocus and exposure are (nm) and (%).

pessimistic as it ignores an averaging effect along a critical path, which usually contains more than a single cell. To reduce the pessimism, we approximate D-EPW by averaging the delay deviation of R number of transistors with slowest delay deviation. The delay deviation of each transistor is given by

$$\Delta\text{Delay} = \left[\frac{I_{\text{on}-\text{original}}}{I_{\text{on}-\text{simulated}}} - 1\right] \times 100\%,$$

where $I_{\text{on}-\text{original}}$ is the I_{on} of the pre-OPC transistor obtained from layout and $I_{\text{on}-\text{simulated}}$ is the I_{on} of NRG transistor from simulated contour. The approximated D-EPW is named as shape-D-EPW and its definition is given as follows:

$$(E_i, F_j, O_k) \in \text{shape D-EPW} \iff \frac{\sum_{n=1}^{R} \Delta\text{Delay}_n}{R} \leq \text{upper bound of allowed delay deviation.} \tag{11}$$

Based on the critical paths of our benchmark circuits, we found that the average transistor stages along a critical path is about 30. Therefore, we used R=30 in our experiment for pessimistic approximation. Note that this definition does not guarantee a strict lower bound as there might be cases where the logic stages along critical paths are less than R and they contain some of the transistors with the worst delay deviations.

Alternatively, we assume that the EPE distribution of transistors along a critical path is similar to that of all transistors in a design. In this case, we can estimate D-EPW by averaging the delay deviation of all transistors, i.e., R=total number of transistors.

5.3 *Results*

Figure 7 shows that histogram D-EPW is similar to the reference D-EPW but the area of histogram P-EPW is significantly smaller than that of reference P-EPW. As a result, the approximated histogram C-EPW only covers a small region of reference C-EPW. The error in histogram P-EPW is mainly due to the definition of channel length in Eq. (9), where a worst case condition is assumed. To make matters worse, the error is exaggerated in P-EPW as leakage power grows exponentially when the channel length shrinks.

Meanwhile, Fig. 7 shows that shape D-EPW and shape C-EPW with R=30 is much smaller than that of reference EPWs. The accuracy of the approximation improves when R is increased to the total number of transistors. Since the evaluation of shape P-EPW is the same as the one for reference P-EPW, there is no difference between them.

In Fig. 8, we can see that all approximation methods cover a higher EPW area compared to A-GPW on average. When both leakage and delay are considered, shape C-EPW with *R=all transistors* has the highest area coverage among the approximated C-EPWs. Although histogram D-EPW shows the highest percentage coverage compared to shape D-EPWs, the covered EPW region for histogram C-EPW is low due to the poor coverage of histogram P-EPW. It is observed that the EPW area covered by shape D-EPW with *R=all transistors* is slightly less than histogram D-EPW although both approximations used the average delay deviation of all

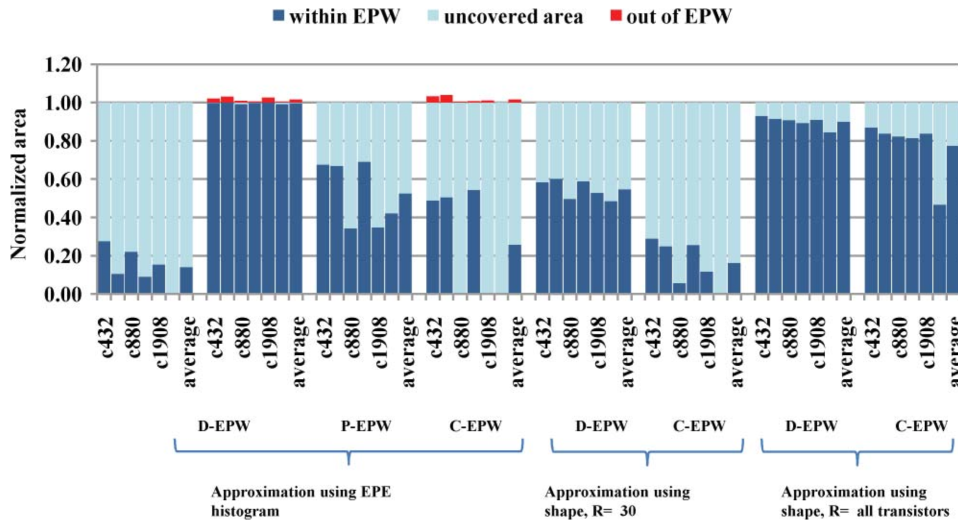

Fig. 8 Accuracy analysis for A-GPW and approximated EPWs of benchmark circuits. EPE tolerance=10%, delay tolerance = 21%, and leakage power tolerance = 311%.

transistors to define D-EPW. This discrepancy is due to the difference between the lumped EPE histogram and actual transistor shape.

It is observed that there are several cases where histogram D-EPW has a region out of D-EPW. This happens because histogram D-EPW is evaluated based on the EPE histogram of an entire design while D-EPW only considers the transistors along critical paths. In contrast, shape D-EPW with *R=all transistors* has no area out of D-EPW.

In summary, EPW extracted based on the shape of each transistor (with *R=all transistors*) is the best approximation among these approaches as it has no area out of EPW and the covered EPW areas are larger than 70% on average.

6 Runtime Reduction Through Representative Layout Extraction

The above-mentioned process window evaluation methods (GPW and EPW, including approximation methods) require lithography simulation of a single design at multiple process points, which is very slow for a large design. The problem worsens if we want to evaluate PW by considering process points at a finer level of granularity. To reduce this lithography simulation runtime, we propose an efficient PW analysis flow depicted in Fig. 9. First, we extract representative layouts (RLs) which contain relevant shapes for EPW analysis. For D-EPW, all critical cells are selected while only 5% of the total cells are selected for P-EPW analysis. Second, we check

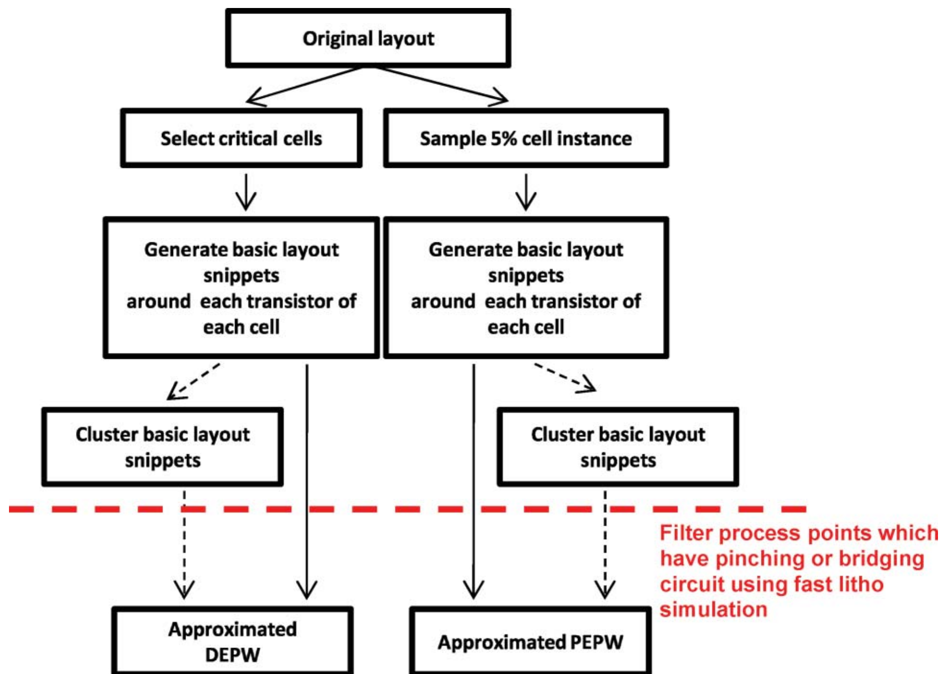

Fig. 9 Clustering flow.

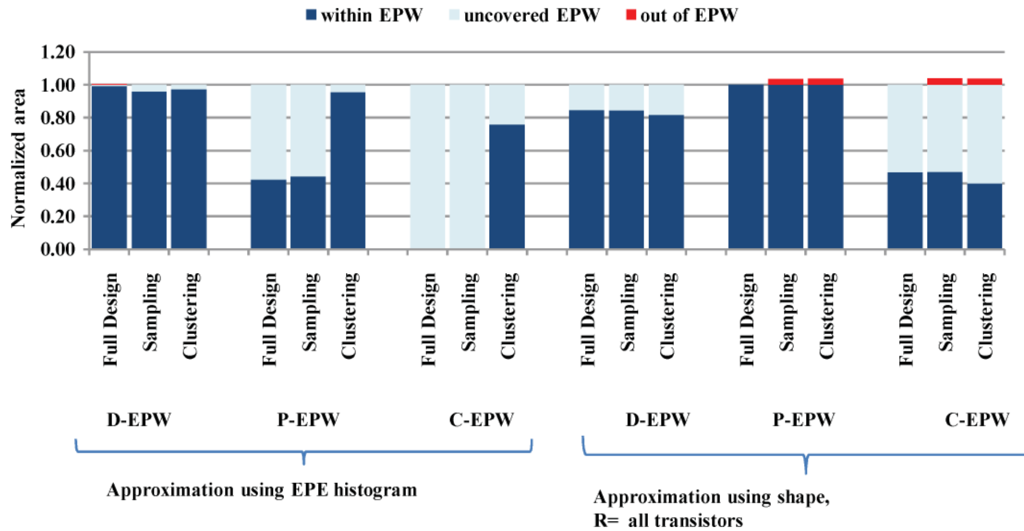

Fig. 10 Accuracy of clustering approach for benchmark design mips.

the printed image of the original layout for all process points to filter out the process points which have pinching/bridging features. This can be done efficiently by using a less accurate but fast lithography simulation setup, which is sufficient to detect pinching/bridging.[‖] In case the selected cells are too many for efficient lithography simulation, we can apply an additional clustering procedure to further reduce the total number of cells. Lithography simulation runtime is reduced because these RLs have a smaller feature count compared to the original layout.

6.1 *Representative Layout Extraction*

For constructing representative layouts, the key thing to observe is that for delay estimation we only need to consider transistors on critical cells because they are more likely to cause timing violation under process variation instead of the entire design. We take a 2 μm $\times 2$ μm square snippet centered at each transistor's channel (of each critical cell) to form basic layout snippets. The size of snippets is chosen to account for optical proximity effects on the transistor under consideration. These layout snippets are then tiled in a separate layout, which we shall call the Delay Representative Layout (DRL) of the design.

For power analysis, there is no obvious selection scheme to extract "critical" shapes as each transistor contributes to total leakage power. To avoid analyzing the entire layout, we sample 5% cell instances from each cell type. We can adjust the sampling rate for obtaining better accuracy. 2 μm $\times 2$ μm snippets for each transistor of each of the chosen cells are then used to construct a Power Representative Layout (PRL) in a similar manner to DRL. This approach of sampling cells for PRL construction reduces runtime while minimizing estimation error because standard cells with the same cell type are likely to have similar leakage power deviation.

Only DRL and PRL of a design layout then undergo lithography simulation at different process corners to evaluate EPW. Note that we use neighboring shapes of a transistor during RL extraction but we only perform EPW analysis on the transistor in the middle of the snippet for both DRL/PRL. We apply the approximate EPW methods discussed earlier to the representative layouts because complete EPW analysis requires detailed information of the critical path. The total lithography simulation runtime of these two RLs of a design is still substantially less than that of the entire design layout as shown in Table 4 [the runtime values are the CPU TIME as reported by Mentor Calibre (Ref. 18)] for one large mips processor layout. We can further reduce total transistor shapes that need to undergo lithography simulation by clustering the chosen layout snippets using the method outlined in Ref. 20. The runtime improvement due to clustering is also shown in Table 4 (clustering runtime is not included, but depending on implementation it can be expensive).

Figure 10 shows the accuracy of our DRL+PRL extraction method compared to evaluation of EPW for the entire design. Both EPE-histogram and shape approximation methods were tried for the RLs. The results show that the PW estimated using representative the layout method is similar to the one which uses an entire design. The shape approximation method is slightly optimistic and overestimated P-EPW.

Table 4 Lithography runtime for representative layouts.

Benchmark circuit	Total cells	Critical cells	Lithography runtime (h): Full design	Lithography runtime (h): Representative layout	Lithography runtime (h): Postclustering
mips	11577	382	198	101	93

[‖]Note that identifying PW to avoid bad pinching/bridging patterns is not sufficient as there are patterns which can only tolerate small errors due to timing and they are design dependent.

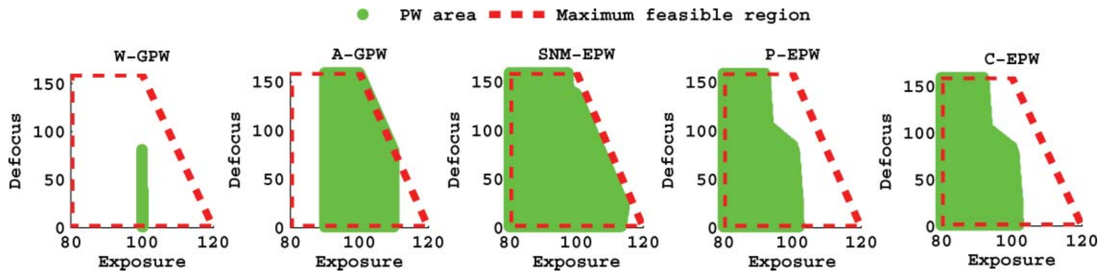

Fig. 11 SRAM GPW versus EPW. Units for defocus and exposure are (nm) and (%).

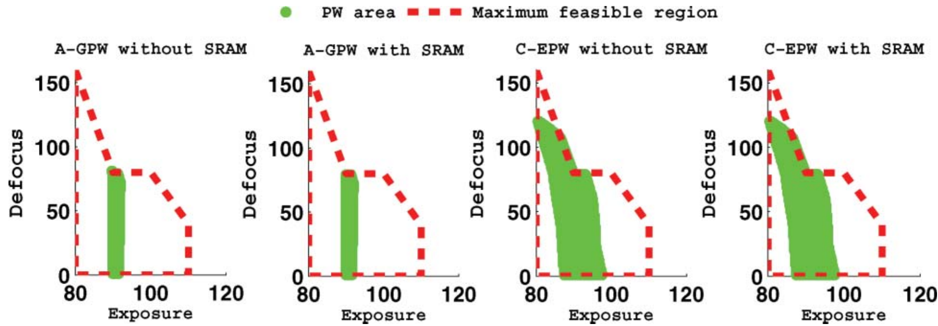

Fig. 12 GPW versus EPW for benchmark circuit c1908. Units for defocus and exposure are (nm) and (%).

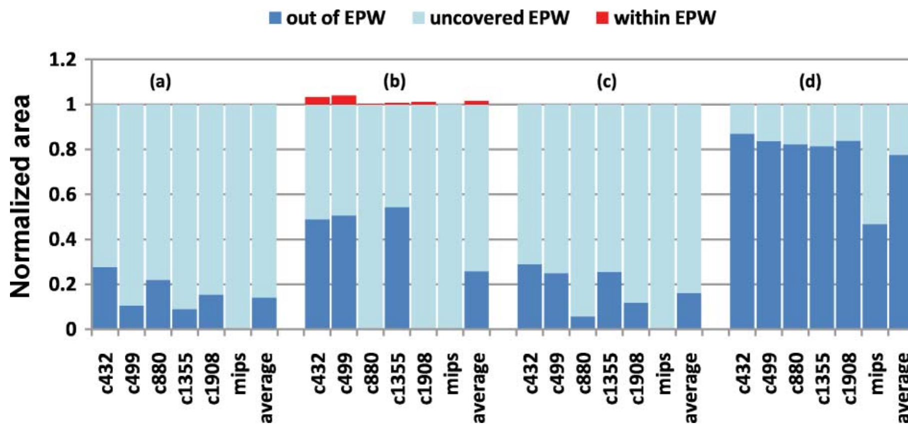

Fig. 13 Accuracy of (a) A-GPW, (b) C-EPW using histogram approximation, (c) C-EPW using shape approximation with R=30, and (d) C-EPW using shape approximation with *R*=total.

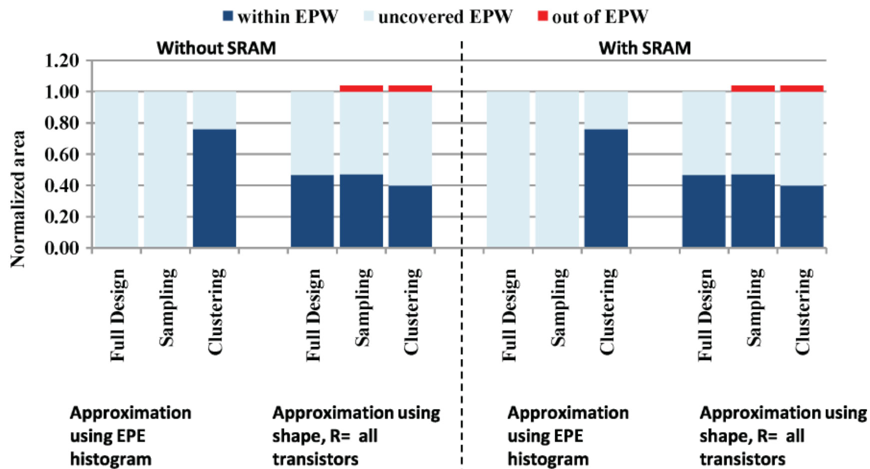

Fig. 14 Accuracy of clustering approach including SNM-EPW for benchmark design mips.

This is due to the fact that the random sampling misses out on some critical patterns that cause leakage power failure. Note that there is no area out of EPW for the histogram method. This happens because the error in sampling is compensated for by the pessimistic estimation of the histogram method as mentioned in Sec. 5. In summary, the RL extraction method reduces lithography simulation runtime significantly at the cost of some loss in EPW accuracy as some EPW critical geometries are not captured due to sampling/clustering i.e., the representative snippets do not have all the features of the critical geometries.

7 EPW Including SRAM

To evaluate the EPW of digital circuits, we need to consider the PW for random logic as well as memory cells. Since the original benchmark circuit does not have memory cells, we draw the layout of the SRAM according to the geometrical dimensions in Ref. 21. After that we optimize the bit cell for Nangate devices by sizing up the pull down transistors from 80 to 120 nm. This improves the static noise margin from 163 to 213 mV. The area of the bit cell is 2.9 μm^2 (0.785 μm $\times$ 0.370 μm).

In our experiment, we duplicate the layout of a 6T SRAM cell to form a memory array for lithography simulation. During PW analysis, we evaluate the bit cell in the middle of the array, which is not affected by empty patterns around layout boundaries.

7.1 *GPW versus EPW*

Figure 11 shows that SNM-EPW is much larger compared to the GPW because:

1. SNM is affected by the relative "drive strength" of transistors instead of absolute critical dimension deviation. For example, when the channel length of all transistors increases due to lithography variation, the impact of I_{on} reduction in a pull down transistor is compensated by I_{on} reduction of an access transistor. As a result, the SNM of a SRAM cell may still be within the desired specification even though the printed contour violates geometrical tolerance.
2. There is an averaging effect across the transistor channel.

To perform a full EPW analysis on benchmark circuits, we define C-EPW as the intersection of delay, power, and SNM-EPW. We use $\pm 10\%$ CD tolerance for SRAM and $\pm 10\%$ CD tolerance for random logic in our experiments.

Figure 12 shows that both GPW and C-EPW do not change after intersecting the digital logic and SRAM PWs. This implies that the SRAM bit cell is not a limiting factor for PW. Also, we notice that A-GPW shows some feasible area which is not covered by SNM-EPW or P-EPW. This happens when actual channel length deviation is larger than 10% but none of the EPE exceeds 10%. In other words, these process points are considered valid if we use the definition of A-GPW but the actual SNM and leakage power violate predefined specifications. The results in Table 5 show that C-EPW is about 8$\times$ larger than GPW on average for digital logic and SRAM circuits.

Table 5 GPW and EPW area with SRAM.

	A-GPW	C-EPW (delay, power, SNM)	Feasible area
c432	300	1086	2760
c499	117	1103	2760
c880	196	890	2565
c1355	95	1052	2760
c1908	139	900	2565
mips	0	248	1590
average	109	839	2448

7.2 *Impact of SRAM on Approximation Methods*

We also study the impact of including SNM-EPW to the approximation methods mentioned in Secs. 5 and 6. Figure 13 shows that the C-EPWs (including SRAM C-EPW) of approximation methods are greater than the PW of GPW. Including SNM-EPW in the C-EPW does not change the result of approximation methods (Sec. 5) because the SNM-EPW is not the limiting PW in this case. Similarly, Fig. 14 shows that including SNM-EPW does not change the result of our representative layout approaches.

8 Conclusions

In this work, we have proposed an electrical process window which is a better measure of process window than the conventional geometric process window. The area of EPW is found to be 1.5 to 8 $\times$ larger than the GPW for our benchmark circuits because it removes the inherent pessimism of GPW by averaging the impact of geometric variation on electrical parameters. We have also analyzed various layout transparent methods to enlarge EPW. Based on our experimental results, we found that gate length biasing and V_{th} push can improve EPW by about 10%. Calculation of delay centric EPW requires information of critical cells in design which is often not available to foundries. Hence, two approximations to EPW, one based on EPE histogram and another based on transistor shape analysis, have been proposed. Our results show that the EPW estimated using transistor shape covers more than 70% of the area of reference EPW on average. We also proposed a method to extract representative layouts which can be used to reduce simulation runtime for process window extraction. The method was able to reduce process window evaluation runtime by 49% with limited impact on accuracy. Though we demonstrate the process window analysis under defocus and exposure variations, other lithography imperfections such as mask error can be included in lithography simulation.

In this work, we measure the EPW at a process point and a supply voltage. Though averaging across, the reported

EPW will be smaller if we consider v_{th} and V_{dd} fluctuation. To account for additional process variation and supply voltage fluctuation, we can evaluate EPW at worst case corners, which gives a more pessimistic estimation. If probability distribution of a process parameters are available, we can reduce the pessimism by simulating the circuit with the statistical information.

Acknowledgments

This work was generously supported in part by IMPACT UC Discovery Grant (http://impact.berkeley.edu) Contract no. ele07-10291, SRC and NSF CAREER Award NO. 0846196. The authors would like to thank the valuable inputs and contributions from Mr. Ranier Yap.

References

1. A. Krasnoperova, J. A. Culp, I. Graur, S. Mansfield, M. Al-Imam, and H. Maaty, "Process window OPC for reduced process variability and enhanced yield," *Proc. SPIE* **6154**, 1200–1211 (2006).
2. N. B. Cobb, and Y. Granik, "Using OPC to optimize for image slope and improve process window," *Proc. SPIE* **5130**, 838–846 (2003).
3. E. Yang, C. H. Li, X. H. Kang, and E. Guo, "Model-based retarget for 45 nm node and beyond," *Proc. SPIE* **7274** 727428 (2009).
4. A. E. Rosenbluth, S. Bukofsky, C. Fonseca, M. Hibbs, K. Lai, A. F. Molless, R. N. Singh, and A. K. K. Wong, "Optimum mask and source patterns to print a given shape," *J. Microlithogr. Microfabr. Microsyst.* **1**(1), 13–30 (2002).
5. C. A. Mack, D. A. Legband, and S. Jug, "Data analysis for photolithography," *Microelectron. Eng.* **46**(1-4), 65–68 (1999).
6. L. Liebmann, S. Mansfield, G. Han, J. Culp, J. Hibbeler, and R. Tsai, "Reducing dfm to practice: the lithography manufacturability assessor," *Design and Process Integration for Microelectronic Manufacturing IV Proc. SPIE* **6156**(1), 61560K (2006).
7. S. Banerjee, P. Elakkumanan, L. W. Liebmann, and M. Orshansky, "Electrically driven optical proximity correction based on linear programming," *Proc. IEEE/ACM ICCAD*, 473–479, IEEE Press, Piscataway, NJ (2008).
8. Q. C. Zhang, and P. van Adrichem, "Determining OPC target specifications electrically instead of geometrically," *Proc. SPIE* **6730**, 67303V (2007).
9. P. Gupta, A. B. Kahng, D. Sylvester, and J. Yang, "Performance-driven optical proximity correction for mask cost reduction," *J Micro/Nanolith. MEMS MOEMS* **6**(3), 031005 (2007).
10. D. Reinhard and P. Gupta, "On comparing conventional and electrically driven opc techniques," *Proc. SPIE* **7488**(1), 748838 (2009).
11. V. Axelrad, A. Shibkov, G. Hill, H.-J. Lin, C. Tabery, D. White, V. Boksha, and R. Thilmany, "A novel design-process optimization technique based on self-consistent electrical performance evaluation," *Proc. SPIE* **5756**(1), 419–426 (2005).
12. T.-B. Chan and P. Gupta, "On electrical modeling of imperfect diffusion patterning," *VLSID*, pp. 224–229, 2010 23rd International Conference on VLSI Design (2010).
13. T.-B. Chan, R. S. Ghaida, and P. Gupta, "Electrical modeling of lithographic imperfections," *VLSID*, pp. 423–428, 2010 23rd International Conference on VLSI Design (2010).
14. "Synopsys HSPICE." http://www.synopsys.com/ (2008).
15. "Nangate Open Cell Library." https://www.nangate.com/ (2010).
16. "Iscas-85 Benchmark Circuits Verilog Files." http://www.pld.ttu.ee/maksim/benchmarks/iscas85/verilog/ (1985).
17. "OpenCores." http://www.opencores.org (2010).
18. "Mentor Calibre." https://www.mentor.com (2008).
19. "Openaccess API." http://www.si2.org/ (2010).
20. J. Ghan, N. Ma, S. Mishra, C. Spanos, K. Poolla, N. Rodriguez, and L. Capodieci, "Clustering and pattern matching for an automatic hotspot classification and detection system," *Proc. SPIE* **7275**, 727516 (2009).
21. F. Boeuf, F. Arnaud, C. Boccaccio, and al., "0.248 μm^2 and 0.334 μm^2 conventional bulk 6t-sram bit-cells for 45nm node low cost - general purpose applications," *Dig. Tech. Pap. -Symp. VLSI Technol.* 130–131 (2005).

Tuck-Boon Chan received his MS degree in electronics engineering from National Taiwan University in 2007. Currently, he is a PhD student in the Electrical Engineering Department of University of California Los Angeles. His research interests are device modeling, design for manufacturing, and computer aided design for VLSI.

Abde Ali Kagalwalla is a graduate student and researcher in the Electrical Engineering Department at UCLA. He works in the NanoCAD lab under the guidance of Professor Puneet Gupta.

Puneet Gupta is currently a faculty member of the Electrical Engineering Department at UCLA. He received his BTech degree in electrical engineering from Indian Institute of Technology, Delhi, in 2000 and PhD in 2007 from University of California, San Diego. He co-founded Blaze DFM Inc. (acquired by Tela Inc.) in 2004 and served as its product architect until 2007. He has authored over 60 papers, ten U.S. patents, and a book chapter. He is a recipient of NSF CAREER award, ACM/SIGDA Outstanding New Faculty Award, European Design Automation Association Outstanding Dissertation Award, and IBM PhD fellowship. Dr. Gupta has given tutorial talks at DAC, ICCAD, Intl. VLSI Design Conference, and SPIE Advanced Lithography Symposium. He has served on the Technical Program Committee of DAC, ICCAD, ASPDAC, ISQED, ICCD, SLIP, and VLSI Design. He served as the Program Chair of IEEE DFM&Y Workshop 2009, 2010. Dr. Gupta's research has focused on building high-value bridges across application-architecture-implementation-fabrication interfaces for lowered cost and power, increased yield and improved predictability of integrated circuits and systems.