Impact of Range and Precision in Technology on Cell-Based Design

John Lee
Electrical Engineering Department
University of California at Los Angeles
lee@ee.ucla.edu

Puneet Gupta
Electrical Engineering Department
University of California at Los Angeles
puneet@ee.ucla.edu

ABSTRACT

With the introduction of non-planar CMOS technologies in commercial designs, the effects of the range and precision allowed in a technology is an important. The limited range and precision (i.e. granularity) in a technology, and consequently, in a standard cell design, may result in significant penalties in the power and delay performance in a design. In this work, the impact of the range and precision is examined by providing a new framework for estimating the power suboptimality incurred by a design relative to a given library. Methods that predict the suboptimality well, both qualitatively and quantitatively, and the implications on standard cell library design are explored. While no other methods for estimating suboptimality are known, compared to a method derived from literature, our method provides a nearly 2x better estimate for v_{th} assignment and 10x improvement for gate sizing.

1. INTRODUCTION

The introduction of multi-gate devices for the 22nm technology node introduces new challenges in creating standard cell libraries. The gate sizes can no longer be arbitrary widths, and instead must come in multiples of the fins, e.g. as 1x, 2x, 3x, 4x, etc [8]. In addition, having a wide array of threshold voltages may not be possible [12], and thus, to understand the impact of these restrictions on the power and delay tradeoffs, we examine the impact of the range and precision on standard cell-based designs.

Standard cell libraries offer a wide variety of both logical functions and sequential elements. In addition to the variety of functions, a range and selection of cell parameters are provided, including changes in the threshold voltages, transistor sizes, and P/N ratios. This provides flexibility for the designer, allowing them to fine tune power, delay, robustness and noise characteristics during circuit design.

Choosing the threshold voltages, transistor sizes, transistor lengths, and P/N ("beta") ratios from the values allowable by the technology is a tradeoff between maintaining a manageable and supportable library, and providing flexibility to the circuit designer. Large libraries require substantial time to design, tune, and characterize. Commercial design teams often use libraries with hundreds of cells, and these cells must be re-characterized periodically to match silicon measurements, and changes in manufacturing process parameters.

This paper examines the suboptimality associated with any given selection of gate sizes and threshold voltages. The suboptimality is relative to the case where a continuous range of gate sizes and threshold voltages are given. More formally, the problem is: *given a design, delay target (or target clock period), and standard cell library, estimate the suboptimality in power, relative to a case where a continuous set of sizes are available.* This question is a stronger question than the library cell selection problem – if the suboptimality can be estimated, this procedure can be used to select a set of library cells, or even the technology that should be used to implement the design. In contrast, research on library cell selection [4, 10, 2] cannot estimate the suboptimality.

Prior work on this subject uses quantization error and experimental results to guide library cell selection. In [4], the authors take an error based approach that minimizes the error incurred from snapping continuous gate sizes. The gate sizes are chosen to minimize the expected error in cell area ("size-match"), delay ("delay-match") or power ("power-match"). More specifically, let s represent the gate size, c_{L} represent the capacitive load of a gate, $\mathbf{Prob}(s)$ represent the probability distribution for the optimal continuous sizes, and $\mathbf{Prob}(s, c_{\mathrm{L}})$ represent the joint probability distribution for the optimal continuous sizes and the capacitive loads. If the set of gate sizes is given as $\{s_1, ..., s_n\}$, then the errors are:

$$\mathrm{so}_{\mathcal{Q}(\mathrm{delay})} = \min_{s_i} |d(s_i, c_{\mathrm{L}}) - d(s, c_{\mathrm{L}})| \quad (1)$$

$$\mathrm{so}_{\mathcal{Q}(\mathrm{size})} = \min_{s_i} |s_i - s| \quad (2)$$

$$\mathrm{so}_{\mathcal{Q}} = \mathrm{so}_{\mathcal{Q}(\mathrm{power})} = \min_{s_i} |p(s_i) - p(s)| \quad (3)$$

where $\mathrm{so}_{\mathcal{Q}(\mathrm{delay})}$ is the delay-match, $\mathrm{so}_{\mathcal{Q}(\mathrm{size})}$ is the size match, $\mathrm{so}_{\mathcal{Q}(\mathrm{power})}$ is the power match, d is the delay of the gate, and p is the power of the gate. The work in [4] focuses on minimizing the expected value of these errors over the distribution of s and c_{L}. These metrics accurately measure the quantization, or the errors associated with snapping to the discrete sizes. However, these metrics are not related to design metrics, such as power or delay suboptimality, and it does not explain why matching the sizes ("size-match") produces the best results.

IEEE/ACM International Conference on Computer-Aided Design (ICCAD) 2012, November 5-8, 2012, San Jose, California, USA

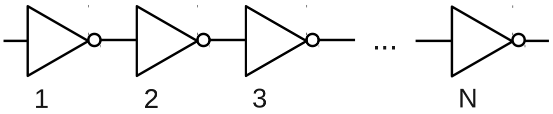


Figure 1: An inverter chain example.

[10] examines which geometric size progression provides the best tradeoffs. They find that a ratio of 1.3 with cells of size $\{1\times, 1.3\times, 1.3^2\times, ...\}$ provides results very close to that with an infinite set of gate sizes. [2] provide experimental results that evaluate different types of library selection. The experiments are performed using their continuous sizing plus branch-and-bound sizing method to evaluate different sets of gate sizes. They conclude that a compact library with sizes $\{0.5\times, 1\times, 2\times, 3\times, 4\times\}$ and 3-4 beta values work the best.

However, while prior work provides some insight into creating libraries, they lack an analytical framework to consider the power vs. delay tradeoff. The question of how the library cell selection affects delay constrained power optimization is still an open question that is very relevant with the increasing constraints on standard cell libraries and the increasing importance of power minimization.

This work shows how the library cells in a design affect the suboptimality of a delay constrained, power optimized design. In this paper, we focus on leakage power, but the framework can be used to estimate the dynamic power suboptimality as well. We also provide experimental results that show our estimates are useful in practice.

In summary, the contributions of this paper are:

- Framework for analyzing the suboptimality in power of a design due to the range and precision of $v_{\rm th}$ and sizes.
- Experimental results that show the usefulness and accuracy of these methods.
- Extensions to library cell selection and applications in multi-terminal gate based design.
- A study of the impact of the range in sizes or $v_{\rm th}$ on the achievable power and delay.

In Section 2, the suboptimality of $v_{\rm th}$ assignment is examined, and in Section 3 the suboptimality of gate sizing is examined. Section 4 considers the dynamic range question, Section 5 discusses the implications of multi-gate based design, and Section 6 considers the effects of mixing technologies in a design. The paper is concluded in Section 7.

2. SUBOPTIMALITY OF $v_{\rm th}$ ASSIGNMENT

Threshold voltage assignment is a very useful tool to optimize designs for power. Increasing the threshold voltage provides an exponential decrease in leakage, with an effect on the delay proportional to [9]:

$$\mathrm{d}(v_{\rm th}) \propto \frac{1}{(V_{\rm dd} - v_{\rm th})^{\alpha}}. \qquad (4)$$

For example, in the Nangate Open Cell Library [1], $V_{\rm dd} = 1.1$ and $\alpha \approx 1.4$.

The rationale in selecting threshold voltages can be governed by Equations (1), (2), and (3). However, this is inadequate to estimate suboptimality. Consider the example in Figure 1, which has N inverters tied as a chain. Suppose a very simple model for the delay and power as a function of $v_{\rm th}$, given by:

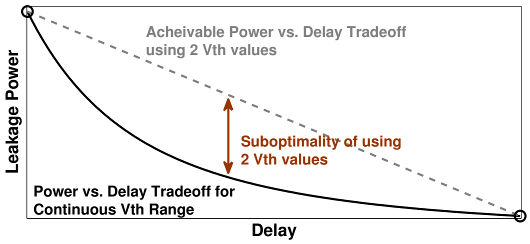


Figure 2: Suboptimality for a gate with two threshold voltages.

$$\text{Delay} = v_{\rm th}, \quad \text{Power} = 3 - v_{\rm th}, \quad 1 \le v_{\rm th} \le 2.$$

For a delay target of T (with $T > N$), an optimal set of continuous $v_{\rm th}$ is one with all gates at $v_{\rm th} = T/N$, and a corresponding power $3N - T$.

Equations (1), (2), and (3) suggest that the suboptimality associated with choosing a discrete set of sizes is related to the quantization error. In the above case, this is accurate for $N = 1$, where the optimal continuous value, $v_{\rm th} = T$, is rounded down to the nearest available discrete $v_{\rm th}$. This gives a power suboptimality of $3 - (T - v_{\rm th})$.

However, the interaction between gates is overlooked in this case. As N grows larger, the suboptimality can decrease – even with just two values: $v_{\rm th} \in \{1, 2\}$. The idea is that the suboptimality incurred by snapping down a $v_{\rm th}$ can be mitigated by snapping up in the next stage. Surprisingly, with this model, the difference between the optimal discrete and continuous solutions is at most 1, and as a percentage it is at most $1/N$. Thus, as N increases, the difference between the quantized and continuous solutions, as a percentage of the total power, continues to decrease to zero, and in the limit, the delay vs. power tradeoff is linear.

In contrast, consider the case where the delay and power are given by:

$$\text{Delay} = v_{\rm th}, \quad \text{Power} = 4 \cdot \exp(-\ln(2) \cdot v_{\rm th}) = 2^{-v_{\rm th}+2}$$

where the power as a function of $v_{\rm th}$ is slightly different. In this case, the values of delay and power at $v_{\rm th} \in \{1, 2\}$ are identical to the earlier example, but the power at the points in between are different. This leads to the situation in Figure 2. In this case, by mixing the options $v_{\rm th} \in \{1, 2\}$, the points along the upper linear tradeoff can be achieved, as in the prior example. However, using continuous $v_{\rm th}$ values can provide the tradeoff values along the lower curve. The difference between the two curves creates the suboptimality, and the $v_{\rm th}$ should be selected in a way to minimize this difference for a majority of designs.

Thus, in general, the suboptimality for a gate is derived from the difference between the curves in Figure 2. For example, for a given gate, the required delay is first determined and the point on the continuous tradeoff curve is identified. Next, the vertical distance to the upper tradeoff line is computed, and this is the suboptimality for the gate. Note that the available $v_{\rm th}$ determines the upper tradeoff line. Mathematically, the difference between the upper tradeoff line $\hat{p}(v_{\rm th}, \tau, c_{\rm L})$, and the continuous power vs. delay tradeoff curve, $p(v_{\mathrm{th}})$ is computed as:

$$\mathrm{so_c}(v_{\mathrm{th}}, \tau, c_{\mathrm{L}}) = \hat{p}(v_{\mathrm{th}}, \tau, c_{\mathrm{L}}) - p(v_{\mathrm{th}}). \tag{5}$$

Here, τ is the set of slews or input transition of the gate, and c_{L} are the capacitive loads for the gate. The upper tradeoff line $\hat{p}(v_{\mathrm{th}}, \tau, c_{\mathrm{L}})$ is defined as the line between the two neighboring v_{th} that are available in the cell library:

$$\hat{p}(v_{\mathrm{th}}) = \lambda(v_{\mathrm{th}}) \cdot p(Q^-(v_{\mathrm{th}})) + (1 - \lambda(v_{\mathrm{th}})) \cdot p(Q^+(v_{\mathrm{th}})) \tag{6}$$

with

$$\lambda(v_{\mathrm{th}}) = \frac{d(v_{\mathrm{th}}) - d(Q^-(v_{\mathrm{th}}))}{d(Q^+(v_{\mathrm{th}})) - d(Q^-(v_{\mathrm{th}}))}, \tag{7}$$

and Q^+ and Q^- are the round-up and round-down quantization functions, respectively, defined as:

$$Q^+(v_{\mathrm{th}}) = \min_i \{v_{\mathrm{th}}, i \mid \ v_{\mathrm{th}}, i \geq v_{\mathrm{th}}\} \tag{8}$$

$$Q^-(v_{\mathrm{th}}) = \max_i \{v_{\mathrm{th}}, i \mid \ v_{\mathrm{th}}, i \leq v_{\mathrm{th}}\}. \tag{9}$$

The "c" in $\mathrm{so_c}$ denotes that this suboptimality is related to the convexity of the delay vs. power curve. $\mathrm{so_c} > 0$ (e.g. there is a non-zero suboptimality) if the curve is strictly convex; as seen earlier, a linear delay vs. power tradeoff has a $\mathrm{so_c} = 0$. An interesting consequence is that if the curve is *concave* then a better tradeoff can be achieved using the linear tradeoff curve, which will lie below the continuous tradeoff curve. In this case, a different analysis applies. This is because the upper tradeoff line from Figure 2 will now lie below the power vs. delay curve. In these cases it is possible for there to be no suboptimality related to having a limited selection of sizes[1].

This model can be improved to handle slew and other timing effects and interactions by using the slacks instead of the delays. In this case, we have:

$$\lambda(v_{\mathrm{th}}) = \frac{\mathrm{slack}(v_{\mathrm{th}}) - \mathrm{slack}(Q^-(v_{\mathrm{th}}))}{\mathrm{slack}(Q^+(v_{\mathrm{th}})) - \mathrm{slack}(Q^-(v_{\mathrm{th}}))}, \tag{10}$$

This improved expression will be used for the remainder of this section.

2.1 Suboptimality expressions

The total suboptimality of a design can be expressed by summing the individual $\mathrm{so}_{\mathrm{c}(g)}$ for each gate g:

$$\mathrm{so}(\vec{v_{\mathrm{th}}}, \vec{\tau}, \vec{c_{\mathrm{L}}}) = \gamma_{\mathrm{c}} \sum_{\forall \mathrm{gates}\ g} \mathrm{so}_{\mathrm{c}(g)}(v_{\mathrm{th}}(g), \tau(g), c_{\mathrm{L}}(g)) \tag{11}$$

where v_{th} is the optimal continuous size, τ is the set of slews or input transitions in the design, and c_{L} are the capacitive loads for each gate in the design. $\gamma_{\mathrm{c}} \approx 1$ is a fitting term used to improve the fit of the bounds.

When the optimal continuous v_{th} are not available, but the design is available, the suboptimality for each gate can be taken to be the worst-case over the slews (τ) and v_{th}:

$$\mathrm{so}(c_{\mathrm{L}}) = \max_{v_{\mathrm{th}}, \tau} \{\gamma_{\mathrm{c}} \mathrm{so_c}(v_{\mathrm{th}}, \tau, c_{\mathrm{L}})\}. \tag{12}$$

When the design itself is unavailable, the expression for the suboptimality of each gate can be rewritten to provide the maximum over all the possible loads as well:

$$\mathrm{so} = \max_{v_{\mathrm{th}}, \tau, c_{\mathrm{L}}} \{\gamma_{\mathrm{c}} \mathrm{so_c}(v_{\mathrm{th}}, \tau, c_{\mathrm{L}})\}. \tag{13}$$

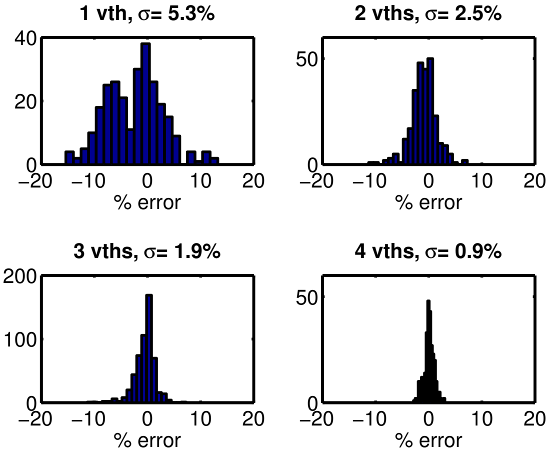


Figure 3: Histogram of the errors in predicting the discrete optimum, from a given optimal continuous solution. The errors are given as a percentage of the discrete optimum.

2.2 Small circuit experiments

To better understand the effects of v_{th} selection on the suboptimality, 30 circuit examples were randomly generated using the method in [11], each having 30 gates. The delay and power functions were modeled using a 65nm commercial library, and fit to posynomial models. These circuits were small enough to solve optimally, using a branch-and-bound method.

The optimal continuous power and discrete powers were computed for ten different delay targets between the minimum delay and maximum delay (the delay associated with the minimum power configuration). Next, the value γ_{c} in (11) is fit to its least-squares value, yielding $\gamma_{\mathrm{c}} = .974$, which shows that the $\gamma_{\mathrm{c}} \approx 1$ and that the preceding analysis is an accurate predictor of the suboptimality. The resulting fit, and errors, are shown in Figure 3.

The results are very good. The standard deviation in errors are 5.3%, 2.5%, 1.9% and .9%, for 1, 2, 3 and 4 v_{th} options, respectively (the errors are given as a percentage of the discrete optimum). In contrast, using the quantization errors in (1), (2), and (3) provide much larger errors, at 16%, 34%, 27% and 14%, respectively.

2.3 Effects on Post-layout Threshold Voltage Assignment

An experiment to measure the effect of threshold voltage assignment on post-layout designs (where the wire loads are known) was performed. Libraries with several different v_{th} choices were used (the percentages denote the deviation from the nominal v_{th}):

- v_{th}-L1: $\{-20\%, 20\%\}$
- v_{th}-L2: $\{-20\%, 0\%, 20\%\}$
- v_{th}-L3: $\{-20\%, 0\%, 10\%, 20\%\}$
- v_{th}-L4: $\{-20\%, 0\%, 10\%, 20\%\}$
- v_{th}-L5: $\{-20\%, 10\%, 0\%, 10\%, 20\%\}$

and they were compared to values for a dense library (e.g. a library with all v_{th} values between -20% and 20% of the nominal v_{th}, in increments of .1%. The library is a 32nm library from a leading EDA company where the $\{-20\%, 0\%, 20\%\}$

[1] However, note that the continuous optimum is *always* less than or equal to the discrete optimum.

c6288	ϵ-$\mathrm{so}_{\mathcal{Q}(\mathrm{power})}$			ϵ- so_{c}		
	max	avg	std	max	avg	std
v_{th}-L1	712%	197%	249%	390%	100%	123%
v_{th}-L2	33%	12%	14%	20%	5.5%	6.7%
v_{th}-L3	42%	5.5%	9.2%	17%	2.6%	4.3%
v_{th}-L4	47%	14%	19%	31%	6.9%	9.7%
v_{th}-L5	8.5%	1.9%	2.4%	8.5%	1.6%	2.3%
c7552						
v_{th}-L1	103%	39%	39%	142%	40%	37%
v_{th}-L2	23%	7.2%	9.3%	14%	4.7%	6.2%
v_{th}-L3	41%	4.3%	9.4%	11%	1.6%	2.8%
v_{th}-L4	17%	7.3%	8.7%	15%	5.1%	6.8%
v_{th}-L5	7.0%	1.4%	2.1%	6.1%	1.6%	1.7%
s35932						
v_{th}-L1	231%	69%	94%	144%	35%	48%
v_{th}-L2	22%	6.8%	8.9%	17%	3.6%	5.4%
v_{th}-L3	26%	3.6%	8.6%	16%	2.2%	4.7%
v_{th}-L4	12%	5.4%	6.4%	9.7%	3.1%	4.0%
v_{th}-L5	7.2%	1.4%	2.2%	10%	1.5%	2.9%
s38417						
v_{th}-L1	48%	3%	7.2%	48%	1.5%	4.6%
v_{th}-L2	6.3%	.3%	1.0%	5.1%	.2%	.8%
v_{th}-L3	3.9%	.1%	.5%	3.6%	.1%	.5%
v_{th}-L4	3.6%	.3%	.6%	2.7%	.1%	.4%
v_{th}-L5	1.1%	.03%	.1%	1.1%	.04%	.1%
s38584						
v_{th}-L1	20%	2.9%	4.1%	13%	3.3%	4.2%
v_{th}-L2	4.3%	.2	.59%	3.9%	.2%	.5%
v_{th}-L3	3.9%	.1%	.44%	3.9%	.1%	.4%
v_{th}-L4	1.4%	.1%	.27%	1.1%	.1%	.2%
v_{th}-L5	.9%	.01%	.13%	1.1%	.1%	.1%

*minimum error for all designs is 0%

Table 1: Errors in v_{th} suboptimality estimation (as a percentage of the total power). Statistics are over the range of delays.

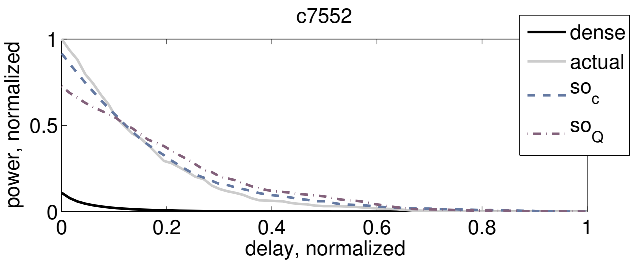


Figure 4: Error prediction on the c7552 benchmark, in the v_{th}-L1 case. The so_{c} and the $\mathrm{so}_{\mathcal{Q}}$ predictions are made using the data from the dense library result.

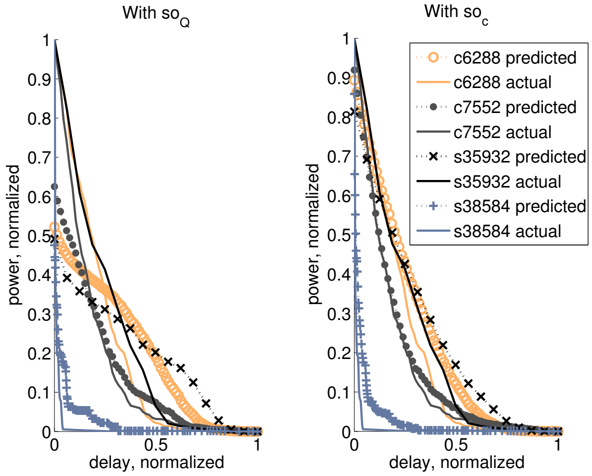


Figure 5: Suboptimality estimates with library v_{th}-L1 for the benchmarks c6288, c7552, s35932, and s38584. The $\mathrm{so}_{\mathcal{Q}}$ method is on the left, and the so_{c} is on the right. The benchmark s38417 omitted as the errors for this benchmark were small.

data points are included in the library. The remaining points are created by fitting each table entry in the Liberty file using exponential models to interpolate the power and (4) to fit the delay. The libraries v_{th}-L1, v_{th}-L2, v_{th}-L3, v_{th}-L4, v_{th}-L5 have 2, 3, 4, 4, and 5 v_{th} options, respectively.

All optimizations are performed using the widely-known and simple to implement circuit optimization method TILOS [5]. While this method is not optimal, even for the dense library, it is derived from an optimal algorithm, and therefore should provide reasonable results. In addition, [6] shows that this method provides results that are robust – they perform well over a range of benchmarks. In this section, the dense library is used as a proxy for a continuous library.

The benchmarks used in this study are widely used gate sizing and v_{th} assignment benchmarks, from the ISCAS '85 and ISCAS '89 benchmark suites. They are initially synthesized, placed and routed with all optimization flags set to high-effort and minimum delay. Once the sizing is complete, the suboptimality estimate for each library choice can be computed in a matter of seconds. A common value of $\gamma_{\mathrm{c}} = .87$ was used to estimate the suboptimality for all benchmarks.

Table 2.3 shows the statistics for the suboptimality prediction errors (max, average, and standard deviation) for this work (so_{c}), and estimates derived from work in [3] ($\mathrm{so}_{\mathcal{Q}}$). The results show that using the so_{c} metric is a better predictor than the $\mathrm{so}_{\mathcal{Q}}$, providing much smaller errors in the worst cases, such as the c6288 v_{th}-L1 library. In the c7552 v_{th}-L1 case, the errors seem worse, however, Figure 4 shows that these numbers are misleading – the fit using the so_{c} is qualitatively better than the fit using the $\mathrm{so}_{\mathcal{Q}}$. Figure 5 shows that this is true in general– the fit using $\mathrm{so}_{\mathcal{Q}}$ on the left does not follow the trend well, and the fit on the right using so_{c} is clearly superior. Overall, the so_{c} has an average error nearly half of the $\mathrm{so}_{\mathcal{Q}}$ estimates – a mean error of 8.8%, compared to 15%.

In viewing these results, it is important to remember two things. First, the suboptimality estimates are predicted from the dense v_{th} assignment. Figure 4 shows that the dense assignment is nearly an order of magnitude smaller than the discrete power, yet both the value and the trends are predicted well.

The next thing to remember is that these results are for practical optimization tools. This is because the TILOS method is not an *optimal* method, and the optimization results have some additional suboptimality involved. However, the estimates still predict the suboptimality well, showing that this method is applicable to practical optimization

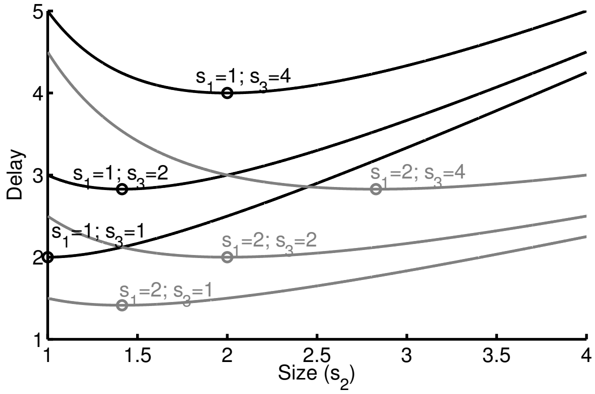


Figure 6: Arrival time at the output of gate 2 (from Figure 1), as a function of the gate sizes s_1, s_2, s_3. The 'o' marks the minimum delay value on each curve.

methods.

3. SUBOPTIMALITY IN GATE SIZING

Computing the suboptimality in gate sizing is substantially different than that of $v_{\rm th}$. This is for two reasons:

- The delays between different gates are coupled – changing the size of a gate will affect the delays of its neighboring gates due to the changing capacitive loading.
- The power tradeoff between different gates is not as dramatic. The change in leakage power as a function of $v_{\rm th}$ is exponential, while the change in leakage power as a function of size is linear.

The coupling of the delay between gates results in cases where increasing a gate's size *decreases* the delay at its output while *increasing* the delay at its inputs, because the gate's input pins increase in size and therefore in capacitance. For example, Figure 6 plots the cumulative delay at the output of gate 2 (from Figure 1), as a function of the input gate size (s_1), the size of gate 2 (s_2) and the size of the output gate 3 (s_3) with the model:

$$\text{Delay} = (\textstyle\sum_{j\in\text{fanout}(i)} s_j)/s_i, \quad \text{Power} = s_i$$

The tradeoff curves vary heavily as the input gate size changes. This is in contrast to the case in Figure 2, where the delay at the gate's input is largely independent of its $v_{\rm th}$.

An important difference between Figures 2 and 6 is that the minimum cumulative delay does not always occur at the maximum or minimum sizes – the maximum size does not always result in the minimum delay. For example, consider the plot in Figure 7, which has the delay as the x-axis, and power as the y-axis. Delay values less than three require sizes between $(1, 2)$, and therefore the choice of intermediate sizes affect whether a particular delay is achievable. When the required delay cannot be achieved by a set of library sizes, it results in a delay penalty that must be corrected by sizing other gates as well. In this process, only pareto-optimal gate sizes, in a power vs. delay sense, are considered.

This results in two cases:

1. A size within the library is able to provide the required delay.
2. No size in the library can provide the required delay.

In the first case, an analogue of (5) can be used to estimate the effect on suboptimality. This is done by finding the difference between the piecewise linear tradeoff function of the available sizes with the tradeoff curve of the continuous sizes. In Figure 8, this is the difference in power between the gray piecewise linear tradeoff function and the black curve. Mathematically, this is:

$$\text{so}_{\rm c}(s, \tau, c_{\rm L}) = \hat{p}(s, \tau, c_{\rm L}) - p(s) \tag{14}$$

where

$$\hat{p}(s, \tau, c_{\rm L}) = \lambda(s, \tau, c_{\rm L}) \cdot p(F^-(s, \tau, c_{\rm L})) + \tag{15}$$

$$(1 - \lambda(s, \tau, c_{\rm L})) \cdot p(F^+(s, \tau, c_{\rm L})) \tag{16}$$

$$\lambda(s, \tau, c_{\rm L}) = \frac{\text{slack}(s, \tau, c_{\rm L}) - \text{slack}(F^-(s, \tau, c_{\rm L}))}{\text{slack}(F^+(s, \tau, c_{\rm L})) - \text{slack}(F^-(s, \tau, c_{\rm L}))}. \tag{17}$$

where F^+ and F^- are generalizations of the quantization functions Q^+ and Q^- for sizing, that are used to handle the case where there is no available size that meets the required slack, and are defined as:

$$F^+(s, \tau, c_{\rm L}) = \min\{s_i \mid \text{slack}(s_i, \tau, c_{\rm L}) \geq \text{slack}(s, \tau, c_{\rm L})\} \tag{18}$$

$$F^-(s, \tau, c_{\rm L}) = \max\{s_i \mid s_i < F^+(s)\}. \tag{19}$$

When there are no sizes that provide the needed slack, the convention $F^+(s) = \infty$ is used. The slacks are used to account for the cumulative effect on the delay, as sizing affects both the delay at the output of the gate, and the delay at the input of the gate.

In the Case 2, the effect on suboptimality is much more difficult to characterize. Not only is the current gate affected, but the resulting delay penalty (due to the gate's inability to achieve the required delay) requires the sizes of other gates to be adjusted to achieve the required delay.

These two cases are approximated as:

$$\text{so}_{\rm c+}(s, \tau, c_{\rm L}) = \begin{cases} \text{so}_{\rm c}(s, \tau, c_{\rm L}) & \text{if } F^+(s) < \infty \\ \gamma_Q(p(Q^+(s)) - p(s)) & \text{otherwise} \end{cases} \tag{20}$$

where $p(Q^+(s)) - p(s)$ is the power penalty for rounding up to the nearest size, with

$$Q^+(s) = \min_i\{s_i \mid s_i \geq s\}. \tag{21}$$

In the first case, the suboptimality is characterized by the convexity of the power vs. delay curve, and when this cannot be applied, it is characterized by the power penalty of snapping the next higher size.

The suboptimality over the whole design can then be estimated as:

$$\text{so}_{\rm c+}(\vec{s}, \vec{\tau}, \vec{c_{\rm L}}) = \sum_{\forall \text{gates } g} \text{so}_{{\rm c+}(g)}(s(g), \tau(g), c_{\rm L}(g)) \tag{22}$$

3.1 Post-layout experiments

Gate sizing suboptimality estimates for post-layout designs (where the wire loads are known) were performed. Libraries with several different gate size choices were used:

- s-L1: 1x, 8x
- s-L2: 1x, 2x, 8x
- s-L3: 1x, 4x, 8x

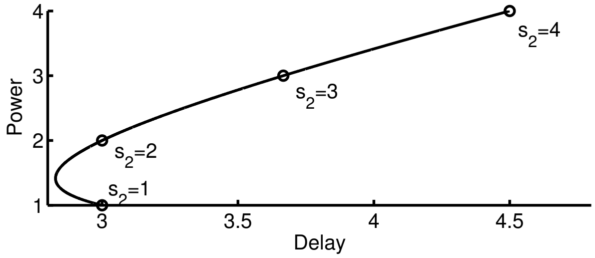


Figure 7: Power delay curve for the example in Figure 2 where the size of gate 1 is 1 ($s_1 = 1$) and the size of gate 3 is equal to 2 ($s_3 = 2$). Delay values < 3 are unachievable with the sizes $\{1, 2, 3, 4\}$.

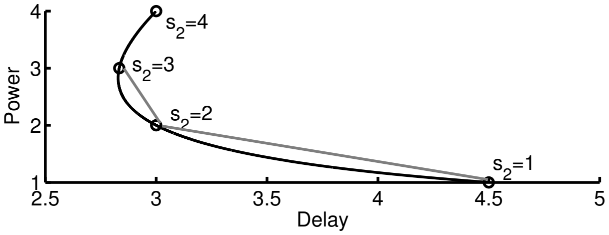


Figure 8: Power delay curve for the example in Figure 2 where $s_1 = 2$ and $s_3 = 4$. Using these gate sizes, the gray piecewise linear tradeoff curve can be approximated.

- s-L4: 1x, 2x, 4x, 8x
- s-L5: 1x, 2x, 3x, 4x, 5x, 6x, 7x, 8x

and they were compared to values for a dense library (e.g. a library with all sizes between 1x and 8x, in increments of .1x. The library is derived from a commercial 45nm library and sizes not in the library are created by fitting each table entry in the Liberty file using linear models to interpolate the power and posynomial models to fit the delay.

As in Section 2.3, all optimizations are performed using TILOS [5], and the same benchmarks are used. The designs are synthesized, placed and routed with all optimization flags set to high-effort.

Table 3.1 shows the errors in the suboptimality estimates for this work ($\mathrm{so}_{\mathrm{c}+}$), and as a reference we provide estimates ($\mathrm{so}_{\mathcal{Q}}$) derived from work in [3]. The table shows that the $\mathrm{so}_{\mathrm{c}+}$ provides a excellent estimate of the suboptimality. Compared to predicting the error using $\mathrm{so}_{\mathcal{Q}(\mathrm{power})}$, the $\mathrm{so}_{\mathrm{c}+}$ is clearly better. Overall, the so_{c} has an average error less than a tenth of the $\mathrm{so}_{\mathcal{Q}}$ estimates – a mean error of .47%, compared to 7%.

In these results, $\gamma_{\mathcal{Q}}$ is fit separately for each of the benchmarks, unlike the case of v_{th} assignment, where the same value of γ_{c} is used for all benchmarks. This is because it is difficult to predict the power penalty when no gate size can provide the needed slack. While the same $\gamma_{\mathcal{Q}}$ can be used, the error rates jump significantly– for example, in the c7552 s-L2 case, the average error becomes 3.9% with a standard deviation of 2.9%. The optimal fitted values for $\gamma_{\mathcal{Q}}$ vary as well, with $\gamma_{\mathcal{Q}} = .71$ for the c6288 to $\gamma_{\mathcal{Q}} = .94$ for the s35932 case. Determining the proper $\gamma_{\mathcal{Q}}$ for a given benchmark *a priori* is an interesting question for future research.

As in the v_{th} case, the estimates are also good qualitatively. Figure 9 shows the estimates with the actual values for the c7552 case. The estimates are very good and follow the trend well.

	ϵ-$\mathrm{so}_{\mathcal{Q}(\mathrm{power})}$			ϵ- $\mathrm{so}_{\mathrm{c}+}$		
	max	avg	std	max	avg	std
c6288						
s-L1	0.6%	0.3%	0.4%	2.2%	0.8%	1.2%
s-L2	78%	19%	30%	5.6%	1.1%	2.3%
s-L3	58%	16%	24%	9.5%	2.2%	4.4%
s-L4	90%	22%	35%	0.2%	0.1%	0.1%
s-L5	86%	21%	34%	0.3%	0.1%	0.1%
c7552						
s-L1	1.0%	0.4%	0.6%	1.1%	0.4%	0.6%
s-L2	14%	4.1%	5.8%	2.8%	0.7%	1.2%
s-L3	7.9%	3.1%	3.5%	6.1%	1.8%	3.2%
s-L4	16%	4.8%	6.9%	0.6%	0.2%	0.4%
s-L5	34%	13%	15%	3.1%	1.0%	1.4%
s35932						
s-L1	1.5%	0.9%	1.0%	2.1%	0.8%	1.1%
s-L2	48%	15%	16%	1.0%	0.4%	0.5%
s-L3	86%	24%	29%	4.7%	1.7%	1.6%
s-L4	43%	15%	15%	0.7%	0.3%	0.4%
s-L5	34%	14%	13%	0.5%	0.2%	0.3%
s38417						
s-L1	0.1%	0.1%	0.1%	0.2%	0.0%	0.1%
s-L2	0.8%	0.2%	0.3%	0.1%	0.0%	0.0%
s-L3	1.2%	0.2%	0.4%	0.4%	0.0%	0.1%
s-L4	0.8%	0.2%	0.3%	0.1%	0.0%	0.0%
s-L5	0.9%	0.2%	0.3%	0.1%	0.0%	0.0%
s38584						
s-L1	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
s-L2	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
s-L3	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
s-L4	3.2%	0.4%	1.0%	0.2%	0.0%	0.1%
s-L5	2.6%	0.3%	0.9%	0.1%	0.0%	0.0%

*minimum error for all designs is 0%

Table 2: Errors in gate sizing suboptimality estimation (as a percentage of the total power). Statistics are over the range of delays.

The analysis in this section shows why the size match library in [3] performed the best. In the above analysis on sizing, the correct sizes are needed to provide the needed slacks in the circuit, otherwise the penalty related to $p(Q^{+}(s)) - p(s)$ is incurred, which is generally larger than the penalty when a size is available. Thus, it better done by matching the continuous sizes, which is a proxy for finding gates that provide the needed slacks, than in matching the delays at the output of the gate (note that in [3], the delay match only matches the delay at the output of each gate, and ignores the impact on the delays at the inputs). This is why the size-match library in [3] provides the best results.

4. DYNAMIC RANGE AND PRECISION SELECTION

The theory developed in this paper is useful in understanding the question of dynamic range – what should the maximum and minimum gate sizes and v_{th} values that should be used? – and the question of library precision selection – which v_{th} and sizes should be used?

The minimum size and maximum v_{th} should be set to the minimum values allowable by the technology. This is due to

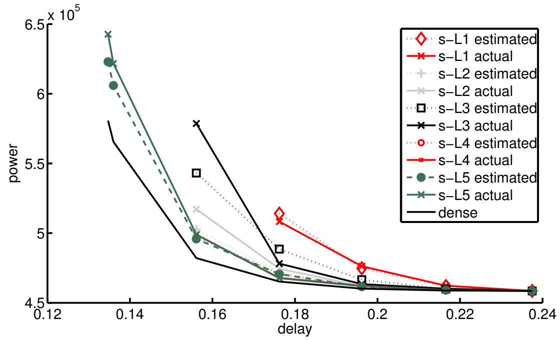


Figure 9: Estimation results for c7552. The estimate is predicted using from the dense sizing using the suboptimality results in this paper, and the actual power refers to the actual power achieved by the TILOS method.

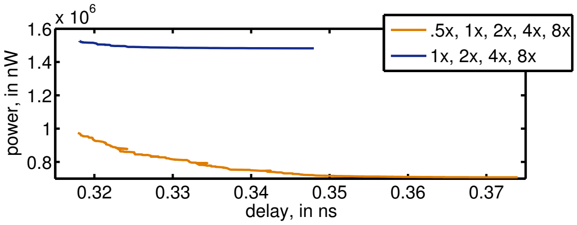


Figure 10: Effect of adding an additional minimum size of .5x on the s35932 benchmark in a 32nm library. The overall power decreases significantly, without affecting the delay range.

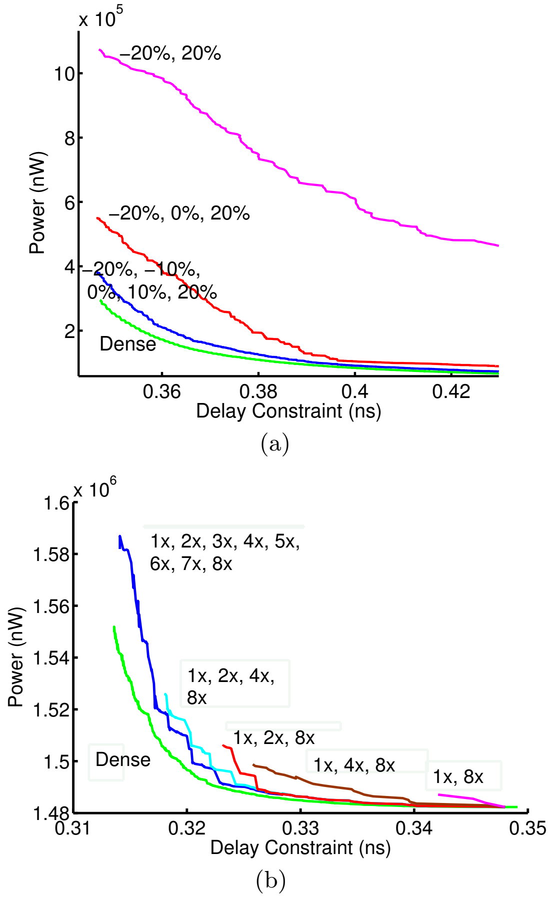
three reasons:

1. The power vs. delay tradeoff is convex and decreasing, thus lower power options improve the achievable power because they have a better power vs. delay tradeoff.
2. In most designs, there are a significant number of gates with positive slack and at their minimum power option. Thus, improving the power of these gates will create significant power savings, without a delay penalty. However, increasing the v_{th} and size choices can incur a technological as well as a library design cost.
3. In gate sizing, reducing a gate to a smaller gate size can reduce the capacitive load of the fanin gates.

For example, Figure 10 shows the effects of adding an additional gate with size 0.5x. The power drop is dramatic, and it is clear that smaller gate sizes make a large impact. This power drop can be roughly estimated by assuming that a percentage of gates at minimum size (or maximum v_{th}) are switched to the newly available lower power option.

The maximum size should be chosen to improve the delay range. Large gates sizes are need to drive large capacitive loads in the design, due to wireloads and gate fanouts. This load dictates the power vs. delay tradeoff curve, as in Figure 6. When there are large load, the need for larger gate sizes increases (as in the $s_1 = 2$, $s_3 = 4$ case), but when larger loads are not present, then large gate sizes are not needed (as in the $s_1 = 1$, $s_3 = 1$ case).

However, it is important to note that the maximum gate size does not itself dictate the delay range. This is because a wide selection of gate sizes are needed to achieve the minimum delay, as shown in Figures 7 and 8. The maximum size may not provide the minimum delay, and it is important



(a)

(b)

Figure 11: Effects of library cell granularity on the achievable delay range for v_{th} assignment (a) and gate sizing (b). The optimization results for the s35932 benchmark in a 32nm library are shown.

to have the right size to achieve the delay that is necessary. The effects of different granularities on the delay range is shown in Figure 11(b). The achievable range using just 1x and 8x is quite poor.

The minimum v_{th} should be selected mainly for delay feasibility. Providing a lower v_{th} decreases the delay that is achievable in a design by providing faster gates. However, because the power tradeoff is convex and decreasing, adding a lower v_{th} is unlikely to improve the power, as the power vs. delay tradeoff is worse for lower v_{th} cells. However, the power may be improved in cases where there are cells that fanout to many other cells, that are all critical.

Regarding library precision, the analysis in this paper shows how to compute the suboptimality from a given design. This can be done using a continuous model of the library, or by creating a dense library. Once information from a dense solution is created, library choices can be analyze quickly – each library choice can be analyzed in a matter of seconds. Also, the dense library need not contain a large range of sizes and v_{th}; it must only contain the candidate sizes and v_{th} that are under consideration.

If a design optimization is not possible or desired, then the library precision can be selected by using (13) and (20) for the expected input slews and output loads. The library designer can use these expressions to minimize the worst-case suboptimalities or the average suboptimalities.

5. MULTI-GATE LIBRARIES

FinFETs and other multi-gate devices are considered to be an alternative for conventional CMOS devices as they offer improved leakage power, reduced parasitic capacitances, and improved resistance to parametric variability [12, 8]. These devices feature fully-depleted channels, and have gates that surround the channel on three sides, offering superior control of the channel.

However, these novel devices pose two challenges to standard library cell designers. Firstly, the gate widths must be quantized– different gate widths are created by placing multiple "fins" in parallel. This means that gate widths will only be available at multiples of the minimum gate width, and fine granularities, such at 1.3x, may not be possible.

From the results above, this should not pose a large problem. As shown in Figure 11 the power penalty between a fully dense library and one with integer sized gates is not large. This is because the power vs. gate width tradeoff is linear, and therefore the power penalty of omitting fractional gate sizes will not be large. However, this may limit the achievable delay range, as in Figure 11, though this is minor.

On the other hand, it is not clear what kinds of v_{th} options will be available in FinFETs and multi-gate device. The v_{th} can be adjusted by changing the gate workfunction [12], or by using the device as a three terminal device [7]. Researchers at the commercial foundry state that multi-v_{th} options can be provided by the FinFET family [13], however the limited availability of v_{th} options to a designer will limit the design space very significantly, as in Figure 11(a).

6. MIXING TECHNOLOGIES

The results in this paper provide an interesting perspective on mixing technologies in a design. Consider the case in Figure 12, where there are two technologies with different power vs. delay curves. Device 1 provides a better tradeoff for large delays, and Device 2 provides better tradeoffs for smaller delays. The benefits of using both technologies can be analyzed in three cases: (1) when the technologies are used independently, (2) the technologies are mixed but optimized independently, and (3) the technologies are mixed and optimized interchangeably.

In the first case of using a single technology, either of the Device 1 or Device 2 tradeoff curves can be achieved, but not a mix of the two. In other words, once the device type is chosen, then the design is restricted to the corresponding power vs. delay tradeoff. This can lead to a large suboptimality in when Device 1 is used near its minimum delay or Device 2 is used at its maximum delay. This suboptimality is avoided in case two, where the technologies are mixed. The mixture of technologies can achieve the pareto-frontier– the left-hand portion of Device 2 and right-hand portion of the Device 1's tradeoff curve.

However, there is a third case, where the devices are mixed and optimized interchangeably. In this case, gates in a single critical path can use devices from either technology, and this can provide a power savings that is greater than if each technology was used individually. For example, in Figure 12, this can provide the bottom dotted-tradeoff line. The tradeoff provided by this type of optimization is better than the best tradeoff of either technology, and there is an interesting synergy than can be exploited by mixture.

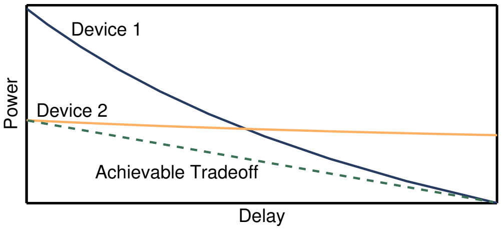


Figure 12: Achievable power delay range using two different types of devices. By mixing the device types, a better power vs. delay tradeoff can be achieved than by either of the device types alone.

7. CONCLUSION AND FUTURE WORK

This work examined the impact of the range and precision in standard cell libraries by proposing methods for estimating the suboptimality in power. The suboptimality is primarily due to the convexity of the power vs. delay tradeoff. Compared to a method derived from literature [3], our method provides a nearly 2x better estimate for v_{th} assignment and 10x improvement for gate sizing. This also leads to insights on selecting the range and precision of the standard cell libraries, especially with the new advances in multi-gate devices. Future work will provide improved analysis for gate sizing suboptimality estimates, and in determining the feasible range of a design, for a given set of gate sizes.

8. ACKNOWLEDGMENTS

This work was supported by the IMPACT UC Discovery Grant and NSF CAREER Award No. 0846196.

9. REFERENCES

[1] Nangate Open Cell Library v1.3. Available from http://www.si2.org/openeda.si2.org/projects/nangatelib.

[2] R. Afonso, M. Rahman, H. Tennakoon, and C. Sechen. Power efficient standard cell library design. In *IEEE Dallas Circuits and Systems Workshop,(DCAS)*, pages 1–4. IEEE, 2009.

[3] F. Beeftink, P. Kudva, D. Kung, R. Puri, and L. Stok. Combinatorial cell design for cmos libraries. *Integration, the VLSI Journal*, 29(1):67–93, 2000.

[4] F. Beeftink, P. Kudva, D. Kung, and L. Stok. Gate-size selection for standard cell libraries. *Proc. ICCAD*, pages 545–550, Nov 1998.

[5] J. Fishburn and A. Dunlop. TILOS: A Posynomial Approach to Transistor Sizing. *Proc. ICCAD*, 1985.

[6] J. Lee and P. Gupta. *Discrete Circuit Optimization: Library Based Gate Sizing and Threshold Voltage Assignment*. Now Publishers, 2012.

[7] P. Mishra, A. Muttreja, and N. Jha. Finfet circuit design. *Nanoelectric Circuit Design*, page 23, 2010.

[8] C. Pacha et al. Circuit design issues in multi-gate fet cmos technologies. In *IEEE ISSCC*, pages 1656–1665. IEEE, 2006.

[9] T. Sakurai and A. Newton. Alpha-power law mosfet model and its applications to cmos inverter delay and other formulas. *IEEE JSSC*, 25(2):584–594, apr 1990.

[10] V. Singhal and G. Girishankar. Optimal gate size selection for standard cells in a library. In *IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software*, pages 47–50, 2006.

[11] D. Stroobandt, P. Verplaetse, and J. van Campenhout. Generating synthetic benchmark circuits for evaluating cad tools. *IEEE TCAD*, 19(9):1011 –1022, sep 2000.

[12] J. Warnock. Circuit design challenges at the 14nm technology node. In *Proc. DAC*, pages 464 –467, june 2011.

[13] C. Yeh et al. A low operating power finfet transistor module featuring scaled gate stack and strain engineering for 32/28nm soc technology. In *IEEE Int. Electron Devices Meeting (IEDM)*, pages 34–1. IEEE, 2010.