Gate-Length Biasing for Runtime-Leakage Control

Puneet Gupta, *Student Member, IEEE*, Andrew B. Kahng, *Member, IEEE*,
Puneet Sharma, *Member, IEEE*, and Dennis Sylvester, *Senior Member, IEEE*

***Abstract*—Leakage power has become one of the most critical design concerns for the system level chip designer. While lowered supplies (and consequently, lowered threshold voltage) and aggressive clock gating can achieve dynamic power reduction, these techniques increase the leakage power and, therefore, causes its share of total power to increase. Manufacturers face the additional challenge of leakage variability: Recent data indicate that the leakage of microprocessor chips from a single 180-nm wafer can vary by as much as 20×. Previously proposed techniques for leakage-power reduction include the use of multiple supply and gate threshold voltages, and the assignment of input values to inactive gates, such that leakage is minimized.**

The additional design space afforded by the biasing of device gate lengths to reduce chip leakage power and its variability is studied. It is well known that leakage power decreases exponentially and delay increases linearly with increasing gate length. Thus, it is possible to increase gate length only marginally to take advantage of the exponential leakage reduction, while impairing performance only linearly. From a design-flow standpoint, the use of only slight increases in gate length preserves both pin and layout compatibility; therefore, the authors' technique can be applied as a postlayout enhancement step. The authors apply gate-length biasing only to those devices that do not appear in critical paths, thus assuring zero or negligible degradation in chip performance. To highlight the value of the technique, the multithreshold voltage technique, which is widely used for leakage reduction, is first applied and then gate-length biasing is used to show further reduction in leakage.

Experimental results show that gate-length biasing reduces leakage by 24%–38% for the most commonly used cells, while incurring delay penalties of under 10%. Selective gate-length biasing at the circuit level reduces circuit leakage by up to 30% with no delay penalty. Leakage variability is reduced significantly by up to 41%, which may lead to substantial improvements in the manufacturing yield and the product cost. The use of gate-length biasing for leakage optimization of cell instances is also assessed, in which: 1) not all timing arcs are timing critical and/or 2) the rise and fall transitions are not both timing critical at the same time.

***Index Terms*—Biasing, channel length, design for manufacturing, gate length, leakage, optimization, sensitivity, static power, variability.**

Manuscript received December 20, 2004; revised April 10, 2005. This paper was recommended by Associate Editor S. Sapatnekar.

P. Gupta and P. Sharma are with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093-0114 USA (e-mail: puneet@ucsd.edu; sharma@ucsd.edu).

A. B. Kahng is with the Department of Computer Science and Engineering and the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093-0114 USA (e-mail: abk@ucsd.edu).

D. Sylvester is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: dmcs@eecs.umich.edu).

Digital Object Identifier 10.1109/TCAD.2005.857313

I. Introduction

HIGH-POWER dissipation in integrated circuits shortens battery life, reduces circuit performance and reliability, and has a large impact on packaging costs. Power in complementary metal–oxide–semiconductor (CMOS) circuits consists of dynamic and static (due to leakage currents) components. Leakage is becoming an ever-increasing component of the total dissipated power, with its contribution projected to increase from 18% at 130 nm to 54% at the 65-nm node [21]. Leakage is composed of three major components: 1) subthreshold leakage; 2) gate leakage; and 3) reverse-biased drain substrate and source-substrate junction band-to-band tunneling leakage [4]. Subthreshold leakage is the dominant contributor to the total leakage at 130 nm and is predicted to remain so in the future [4]. In this paper, we present a novel approach for subthreshold leakage reduction.

Leakage-reduction methodologies can be divided into two classes depending on whether they reduce standby leakage or runtime leakage. Standby techniques reduce the leakage of devices that are known not to be in operation, while runtime techniques reduce the leakage of active devices. Several techniques have been proposed for standby-leakage reduction. Body biasing or variable threshold MOS (VTMOS)-based approaches [12] dynamically adjust the device $V_{\rm th}$ by biasing the body terminal.[1] Mutithreshold CMOS (MTCMOS) techniques [13], [17], [18], [26] use high-$V_{\rm th}$ CMOS [or negative channel MOS (NMOS) or positive channel MOS (PMOS)] to disconnect Vdd or Vss or both to a logic circuit implemented using low $V_{\rm th}$ devices in standby mode. Source biasing, where a positive bias is applied in standby state to the source terminals of OFF devices, was proposed in [11]. Other techniques, such as the use of transistor stacks [33] and input vector control [10], have also been proposed.

The only mainstream approach to runtime-leakage reduction is the multi-$V_{\rm th}$ manufacturing process. In this approach, cells in noncritical paths are assigned high $V_{\rm th}$, while cells in critical paths are assigned low $V_{\rm th}$. Wei *et al.* [30] presented a heuristic algorithm for the selection and assignment of an optimal high $V_{\rm th}$ to cells on noncritical paths. The multi-$V_{\rm th}$ approach has also been combined with several other power-reduction techniques [15], [27], [32]. The primary drawback to this technique has traditionally been the rise in process costs due to the additional steps and masks. However, the increased costs have been outweighed by the resulting substantial leakage reductions, and multi-$V_{\rm th}$ processes are now standard. A new complication facing multi-$V_{\rm th}$ is the increased variability of

[1]Body biasing has also been proposed to reduce the leakage of active devices [22].

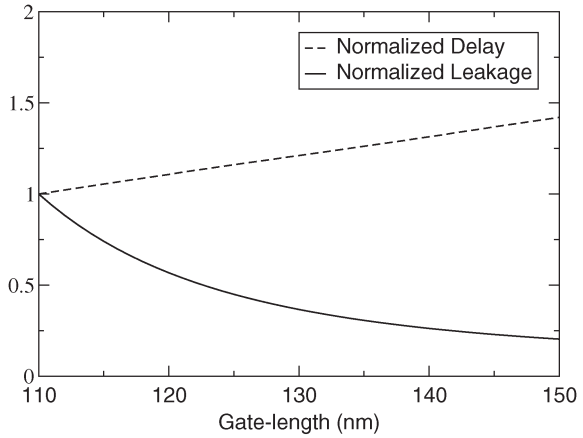

Fig. 1. Variation of leakage and delay (each normalized to 1.00) for an NMOS device in an industrial 130-nm technology.

$V_{\rm th}$ for low-$V_{\rm th}$ devices. This occurs in part due to the random doping fluctuations, as well as the worsened drain-induced barrier lowering (DIBL) and short channel effects (SCEs) in devices with lower channel doping. The larger variability in $V_{\rm th}$ degrades the achievable leakage reductions of multi-$V_{\rm th}$ and worsens with continued MOS scaling. Moreover, multi-$V_{\rm th}$ methodologies do not offer a smooth tradeoff between performance and leakage power. Devices with different $V_{\rm th}$ typically have a large separation in terms of performance and leakage, for instance, a 15% speed penalty with a $10\times$ reduction in leakage for high-$V_{\rm th}$ devices.

The use of longer gate lengths ($L_{\rm Gate}$) in devices within noncritical gates was first described in [29]. In that study, large changes to the gate lengths were considered, resulting in heavy delay and dynamic power penalties. Moreover, cell layouts with significantly larger gate lengths are not layout swappable with their nominal versions, resulting in substantial engineering change order (ECO) overheads during layout. In this paper, we propose very small increases in gate length for noncritical devices. These small increases maximize the leakage reduction since they take full advantage of the SCE and incur only very small penalties in drive current and input capacitance. Technologies at the 90-nm node and below employ super-halo doping, giving rise to reverse SCEs (RSCEs) that mitigate the traditional SCE to some extent. However, we have found the proposed technique to substantially reduce leakage for the two 130-nm and two 90-nm industrial processes that we investigated. Recent reports from leading integrated device manufacturers (IDMs) indicate that SCE continues to dominate $V_{\rm th}$ roll-off characteristics at the 65- and 45-nm technology nodes [6], [16], [19], [20]. However, we note that the $V_{\rm th}$ roll-off curve must be understood to assess the feasibility of this approach and to determine reasonable increases for the gate length.

The variation of delay and leakage with the gate length is shown in Fig. 1 for an industrial 130-nm process. Leakage current flattens out with gate lengths beyond 140 nm, making $L_{\rm Gate}$ biasing less desirable in that range. Another major advantage of $L_{\rm Gate}$ biasing is leakage variability reduction. Since the sensitivity of leakage to gate length reduces with increased gate length, a fixed level of variability in gate length translates to a reduced variability in leakage. We use the terms gate-length biasing and $L_{\rm Gate}$ biasing interchangeably to refer to the proposed technique. We use the phrase "biasing a device" to imply increasing the gate length of the device slightly.

In this paper, we also assess the costs and benefits of transistor-level $L_{\rm Gate}$ biasing (TLLB). Since different transistors control different timing arcs of a cell, TLLB can individually modify delays of different timing arcs. Our hypothesis is that asymmetry in the timing criticality of different timing arcs of a cell instance in a circuit, and that of the rise and fall transitions, can be used by TLLB to yield significant leakage savings. Ketkar and Saptnekar [14], Sirichotiyakul *et al.* [28], and Wei *et al.* [31] proposed transistor-level $V_{\rm th}$ assignment for leakage-power reduction. Our approach uses $L_{\rm Gate}$ biasing instead of $V_{\rm th}$ assignment and is similar to that of [31]. The major disadvantage of TLLB (or $V_{\rm th}$ assignment) is the increase in library size and its characterization time.

The contributions of our paper include the following:

1) a leakage-reduction methodology based on a less than 10% increase in drawn $L_{\rm Gate}$ of devices;
2) a thorough analysis of the potential benefits and caveats of such a biasing methodology, including the implications of lithography and process variability;
3) experiments and results showing the potential benefits of an $L_{\rm Gate}$-biasing methodology in different design scenarios such as dual $V_{\rm th}$.

The organization of this paper is as follows. In Section II, we describe the proposed $L_{\rm Gate}$-biasing methodology for leakage reduction. Section III extends the ideas to be applied at the transistor level for further reduction of leakage at the cost of increased library size. Section IV presents the experiments and results for the validation of the proposed ideas. It also analyzes the potential manufacturing and process variation implications of biasing gate lengths. Finally, Section V concludes with a brief description of ongoing research.

II. Cell-Level Gate-Length Biasing

In this section, we describe the proposed cell-level $L_{\rm Gate}$-biasing (CLLB) methodology. Our approach extends a standard cell library by adding biased variants to it. We then use a leakage-optimization approach to incorporate slower low-leakage cells into noncritical paths, while retaining faster high-leakage cells in critical paths.

A. Library Generation

We generate a restricted library composed of variants of the 25 most commonly used cells in our test cases.[2] For each cell, we add a biased variant in which all devices have the biased gate length. We consider less than 10% biasing because of the following reasons.

1) The nominal gate length of the technology is usually very close to or beyond the "knee" of the leakage versus the $L_{\rm Gate}$ curve which arises due to SCE. For a large bias, the advantage of super-linear dependence of leakage on

[2]We first synthesize our test cases with the complete Artisan Taiwan Semiconductor Manufacturing Company (TSMC) library to identify the most frequently used cells.

the gate length is lost. Moreover, the dynamic power and delay both increase almost linearly with the gate length. Therefore, small biases give more "bang for the buck."

2) From a manufacturability point of view (discussed later in Section IV-B), having two prevalent pitches (which are relatively distinct) in the design can harm the printability properties (i.e., size of process window). We retain the same polypitch as the unbiased version of the cell: There is a small decrease in spacing between gate-poly geometries, but minimum spacing rules are not violated even when the unbiased polys are at minimum spacing, since our biases are within the tolerance margins. Since design rules check (DRC) tools first snap to grid, biases of under 10% are not detected and are considered acceptable due to margins in design rules.
3) An increase in the drawn dimension, which is less than the layout grid resolution (typically 10 nm for 130-nm technology), ensures pin compatibility with the unsized version of the cell. This is very important to ensure that multi-L_{Gate} optimizations can be done post placement or even after detailed routing without ECOs. In this way, we retain the layout transparency that has made multi-V_{th} optimization so adoptable within chip-implementation flows. Biases smaller than the layout grid pitch also ensure design-rule correctness for the biased cell layout, provided that the unbiased version is design-rule correct.

For the simulation program with integrated circuits emphasis (SPICE) models we use, the nominal gate length of all transistors is 130 nm. In our approach, all transistors in a biased variant of a cell have a gate length of 138 nm. We choose 138 nm as the biased gate length because it places the delay of the low-V_{th}-biased variant between the low-V_{th}-nominal gate-length variant and the nominal-V_{th}-nominal gate-length variant. A larger bias can lead to a larger per-cell leakage saving at a higher performance cost. However, in a resizing setup (described below) with a delay constraint, the leakage benefit over the whole design can decrease as the number of instances that can be replaced by their biased version is reduced. Larger or smaller biases may produce larger leakage reductions for some designs. Libraries, however, are not design specific and a biased gate length that produces good leakage reductions for all designs must be chosen. We have found the abovementioned approach for choosing the biased gate length to work well for all designs. We note that this value of 138 nm is highly process specific and is not intended to reflect the best biased gate length for all 130-nm processes. We discussed biasing at finer levels of granularity (i.e., having multiple biased gate lengths and independently biasing devices within a cell) in [9]. However, we did not find any significant leakage savings beyond those from the approach mentioned above.[3]

An important component of the methodology is the layout and the characterization of the dual-L_{Gate} library. Since we investigate very small biases to the gate length, the layout of the biased library cell does not need to change, except for a simple automatic scaling of dimensions. Moreover, since the bias is smaller than the minimum layout grid pitch, design-rule violations do not occur. Of course, after the slight modifications to the layout, the biased versions of the cell are put through the standard extraction and power/timing characterization process.

B. Optimization for Leakage

We perform standard gate sizing (gate width sizing) prior to L_{Gate} biasing using Synopsys Design Compiler v2003.06-SP1. Since delay is almost always the primary design goal, we perform sizing to achieve the minimum possible delay. We use a sensitivity-based downsizing (i.e., begin with all nominal cell variants and replace cells on noncritical paths with biased variants) algorithm for leakage optimization. In our studies, we have found downsizing to be significantly more effective at leakage reduction than upsizing (i.e., begin with all biased variants in the circuit and replace the critical cells with their nominal-L_{Gate} variants), irrespective of the delay constraints. An intuitive rationale is that upsizing approaches have dual objectives of delay and leakage during the cell selection for upsizing. Downsizing approaches, on the other hand, only downsize cells that do not cause timing violations and have the sole objective of leakage minimization. We note that an upsizing approach may be faster when loose delay constraints are to be met since very few transistors have to be upsized. However, delay is almost always the primary design goal and loose delay constraints are rare. A timing analyzer is an essential component of any delay-aware power optimization approach; it is used to compute the delay sensitivity to biasing of cell instances in the design. For an accurate yet scalable implementation, we use three types of timers that vary in speed and accuracy.

1) Standard static timing analysis (SSTA). Slews and actual arrival times (AATs) are propagated forward after a topological ordering of the circuit. The required arrival times (RATs) are back propagated and slacks are then computed. The slew, delay, and slack values of our timer match exactly with Synopsys PrimeTime vU-2003.03-SP2 and our timer can handle unate and nonunate cells.[4]
2) Exact incremental STA (EISTA). We begin with the fan-in nodes of the node that has been modified. From all these nodes, slews and AATs are propagated in the forward direction until the values stop changing. RATs are back propagated from only those nodes for which the slew, AAT, or RAT has changed. The slews, delays, and slacks match exactly with SSTA.
3) Constrained incremental STA (CISTA). The sensitivity computation involves temporary modifications to a cell to find the change in its slack and leakage. To make this step faster, we restrict the incremental timing calculation

[3]We have recently been informed that a major U.S. semiconductor manufacturer has started to offer its customers a cellwise L_{Gate}-biased variant of its 90-nm cell library with a 6-nm bias. Also, a recent paper by Texas Instruments describes a very similar approach used by them [24]. This not only reinforces the viability of the methodology we describe, but also suggests that our use of an 8-nm bias for a 130-nm cell library provides a practically relevant testbed.

[4]Delay values from our timer match with PrimeTime only under our restricted use model. Our timer does not support several important features such as interconnect delay, hold time checks, false paths, multiple clocks, three-pin standard delay format (SDF), etc.

```
procedure L_GateBiasing
1  Run SSTA to initialize s_p  ∀cell instances, p
2  S ← {}
3  forall cell instances, p
4      P_p ← ComputeSensitivity(p)
5      S ← S ∪ P_p
6  do
7      P_p* ← max(S)
8      if(P_p* ≤ 0)
9          exit
10     S ← S − {P_p*}
11     SaveState()
12     Downsize cell instance p*
13     EISTA(p*)
14     if(TimingViolated())
15         RestoreState()
16     else
17         N ← p* ∪ fan-in and fan-out nodes of p*
18         forall q ∈ N
19             if(P_q ∈ S)
20                 P_q ← ComputeSensitivity(q)
21                 Update P_q in S
22 while(|S| > 0)

procedure ComputeSensitivity(q)
1  old_slack ← Slack on cell instance q
2  old_leakage ← Leakage of cell instance q
3  SaveState()
4  Downsize cell instance q
5  CISTA(q)
6  new_slack ← Slack on cell instance q
7  new_leakage ← Leakage of cell instance q
8  RestoreState()
9  return (old_leakage − new_leakage)/(old_slack − new_slack)
```

Fig. 2. Pseudocode for cell-level gate-length biasing for leakage optimization.

to only one stage before and after the gate being modified. The next stage is affected by the slew changes and the previous stage is affected by the pin capacitance change of the modified gate. The ripple effect on other stages farther away from the gate (primarily due to slew changes[5]) is neglected since high accuracy is not critical for sensitivity computation.

We use the phrase "downsizing a cell instance" (or node) to mean replacing it by its biased variant in the circuit. In our terminology, s_p represents the slack on a given cell instance p, and s'_p represents the slack on p after it has been downsized. ℓ_p and ℓ'_p indicate the initial and final leakages of cell instance p before and after downsizing, respectively. P_p represents the sensitivity associated with cell instance p and is defined as

$$P_p = \frac{\ell_p - \ell'_p}{s_p - s'_p}.$$

The pseudocode for our leakage-optimization implementation is given in Fig. 2. The algorithm begins with SSTA and initializes slack values s_p in Line 1. Sensitivities P_p are computed for all cell instances p and put into a set S in Lines 2–5. We select and remove the largest sensitivity P_{p^*} from the set S and continue with the algorithm if $P_{p^*} \geq 0$. In Line 11, the function SaveState saves the gate lengths of all transistors in the circuit, as well as the delay, slew, and slack values. The cell instance p^* is downsized and EISTA is run from it to update the delay, slew, and slack values in Lines 12–13. Our timing libraries capture the effect of biasing on the slew as well as the input capacitance and our static timing analyzer efficiently and accurately updates the design to reflect the changes in delay, capacitance, and slew due to the downsizing move. If there is no timing violation (negative slack on any timing arc), then this move is accepted, otherwise the saved state is restored. If the move is accepted, we also update the sensitivities of node p^*, its fan-in nodes, and its fan-out nodes in Lines 17–21. The algorithm continues until the largest sensitivity becomes negative or the size of S becomes zero. Function ComputeSensitivity(q) temporarily downsizes the cell instance q and finds its slack using CISTA. Since high accuracy is not critical for the sensitivity computation, we choose to use CISTA, which is faster but less accurate than EISTA. Table I shows a comparison of leakage and runtime when EISTA and CISTA are used for sensitivity computation.

TABLE I
COMPARISON OF LEAKAGE AND RUNTIME WHEN EISTA AND CISTA ARE USED FOR THE SENSITIVITY COMPUTATION

Circuit	Leakage (mW)		CPU (s)	
	EISTA	CISTA	EISTA	CISTA
s9234	0.0712	0.0712	4.86	2.75
c5315	0.3317	0.3359	24.18	14.99
c7552	0.6284	0.6356	55.56	43.79
s13207	0.1230	0.1228	33.43	17.15
c6288	1.8730	1.9157	508.86	305.09
alu128	0.4687	0.4857	1122.89	544.75
s38417	0.4584	0.4467	1331.49	746.79

TABLE II
ASYMMETRY IN DELAYS OF VARIOUS TIMING ARCS WITHIN A NAND2X2 CELL

Timing Arc	Propagation Delay (ps)	Transition Delay (ps)
A → Y ↑	99.05	104.31
A → Y ↓	73.07	79.12
B → Y ↑	107.20	112.98
B → Y ↓	70.65	76.37

III. TRANSISTOR-LEVEL GATE-LENGTH BIASING

We use the term timing arc to indicate an intracell path from an input transition to a resulting rise (or fall) output transition. For an n-input gate, there are $2n$ timing arcs.[6] Due to different parasitics, as well as PMOS/NMOS asymmetries, these timing arcs can have different delay values associated with them. For instance, Table II shows the delay values for the same input slew and load capacitance pair for different timing arcs of a NAND2X2 cell from the Artisan TSMC 130-nm library. Pin swapping is a common postsynthesis timing optimization step to make use of the asymmetry in delays of different input pins. To make use of asymmetry in rise–fall delays, techniques such as PMOS/NMOS (P/N) ratio perturbations have been

[5]There may be some impact due to the coupling-induced delay also, as the arrival time windows can change; we ignore this effect.

[6]There may be four timing arcs corresponding to nonunate inputs (e.g., select input of MUX).

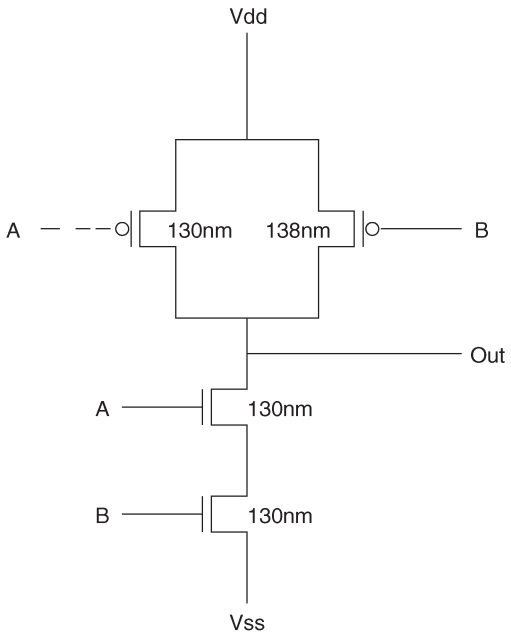

Fig. 3. Gate-length biasing of the transistors in NAND2X1 when only the rise and fall timing arcs from input A to the output are critical.

previously proposed to decrease circuit delay [5]. We propose to exploit these asymmetries using TLLB to "recover" leakage from noncritical timing arcs within a cell.

A. Library Generation

For each cell, our library contains the variants corresponding to all subsets of the set of timing arcs. A gate with n inputs has $2n$ timing arcs and, therefore, 2^{2n} variants (including the original cell). Given a set of critical timing arcs, our goal is to assign a biased L_{Gate} to some transistors in the cell and a nominal L_{Gate} to the remaining transistors such that: 1) critical timing arcs have a delay penalty of under 1% with respect to the original unbiased cell and 2) cell leakage power is minimized. Assignment of L_{Gate} to transistors in a cell, given a set of critical timing arcs, can be done by analyzing the cell topology for simple cells. However, we automate the process in the following manner. We enumerate all configurations for each cell in which nominal L_{Gate} is assigned to some transistors and biased L_{Gate} to the others. For each configuration, we find the delay and leakage under a canonical output load of an inverter (INVX1) using SPICE. Now for each possible subset of timing arcs that can be simultaneously critical, one biasing configuration is chosen based on the two criteria given earlier. Fig. 3 shows L_{Gate} biasing of the transistors in the simplest NAND cell (NAND2X1) when only the rise and fall timing arcs from input A to the output are critical. In this case, only the PMOS device with B as its input can be slowed without penalizing the critical timing arcs.

B. Optimization for Leakage

We use a sensitivity-based downsizing approach that is very similar to the one described in Section II-B. We keep track of the slack on every timing arc and compute sensitivity for each timing arc. To limit the runtime and memory requirements, we first optimize at the cell level and then optimize at the transistor level for only the unbiased cells in the circuit.

IV. Experiments and Results

We now describe our test flow for the validation of the L_{Gate}-biasing methodology, and present experimental results. Details of the test cases[7] used in our experiments are given in Table III. The test cases are synthesized with the Artisan TSMC 130-nm library using Synopsys Design Compiler v2003.06-SP1 with low-V_{th} cells only. To limit the library characterization runtime, we restrict the library to variants of the following 25 most frequently used cells: CLKINVX1, INVX12, INVX1, INVX3, INVX4, INVX8, INVXL, MXI2X1, MXI2X4, NAND2BX4, NAND2X1, NAND2X2, NAND2X4, NAND2X6, NAND2X8, NAND2XL, NOR2X1, NOR2X2, NOR2X4, NOR2X6, NOR2X8, OAI21X4, XNOR2X1, XNOR2X4, XOR2X4. To identify the most frequently used cells, we synthesize our test cases with the complete library and select the 25 most frequently used cells. The delay constraint is kept tight so that the postsynthesis delay is close to the minimum achievable delay.

We consider up to two gate lengths and two threshold voltages. We perform experiments for the following scenarios: 1) single-V_{th}, single-L_{Gate} (SVT-SGL); 2) dual-V_{th}, single L_{Gate} (DVT-SGL); 3) single-V_{th}, dual-L_{Gate} (SVT-DGL); and 4) dual-V_{th}, dual L_{Gate} (DVT-DGL). The dual-V_{th} flow uses nominal and low values of V_{th} while the single-V_{th} flow uses only the low value of V_{th}. STMicroelectronics 130-nm device models are used with the two V_{th} values each for PMOS and NMOS transistors (PMOS: -0.09 and -0.17 V; NMOS: 0.11 and 0.19 V). We use Cadence SignalStorm v4.1 (with SYNOPSYS HSPICE) for delay and power characterization of cell variants. Synopsys Design Compiler is used to measure the circuit delay, dynamic power, and leakage power. We assume an activity factor of 0.02 for the dynamic power calculation in all our experiments. We do not assume any wire-load models, as a result of which, the dynamic power and delay overheads of L_{Gate} biasing are conservative (i.e., overestimated). All experiments are run on an Intel Xeon 1.4-GHz computer with 2 GB of RAM.

A. Leakage Reduction

Table IV shows the leakage savings and delay penalties due to L_{Gate} biasing for all the cells in our library. The results strongly support our hypothesis that small biases in L_{Gate} can afford significant leakage savings with a small performance impact. To assess the maximum impact of biasing, we explore the power-performance envelope obtained by replacing every device in the design by its device-level biased variant.

We now use our leakage-optimization approach to selectively bias cells on the noncritical paths. Table V shows the leakage reduction, dynamic power penalty, and total power reduction for our test cases when L_{Gate} biasing is applied without the

[7]To handle sequential test cases, we convert them to combinational circuits by treating all flip-flops as primary inputs and primary outputs.

TABLE III
TEST CASES USED IN OUR EXPERIMENTS AND THEIR DETAILS

Test Case	Source	#Cells	Delay (ns)	Leakage (mW)	Dynamic (mW)
s9234	ISCAS'89	861	0.437	0.7074	0.3907
c5315	ISCAS'85	1442	0.556	1.4413	1.5345
c7552	ISCAS'85	1902	0.485	1.8328	2.0813
s13207	ISCAS'89	1957	0.904	1.3934	0.6296
c6288	ISCAS'85	4289	2.118	3.5994	8.0316
alu128	Opencores.org[2]	7536	2.306	5.1571	4.4177
s38417	ISCAS'89	7826	0.692	4.9381	4.2069

TABLE IV
LEAKAGE REDUCTION AND DELAY PENALTY DUE TO GATE-LENGTH BIASING FOR ALL 25 CELLS IN OUR LIBRARY

Cell	Low V_{th}		Nominal V_{th}	
	Leakage Reduction (%)	Delay Penalty (%)	Leakage Reduction (%)	Delay Penalty (%)
CLKINVX1	30.02	5.59	34.12	5.54
INVX12	30.28	4.70	36.27	6.87
INVX1	29.45	5.08	33.63	5.12
INVX3	30.72	5.68	35.67	5.52
INVX4	30.01	5.36	35.38	6.28
INVX8	29.97	6.75	35.73	5.25
INVXL	24.16	4.91	28.05	4.79
MXI2X1	23.61	5.45	27.26	5.97
MXI2X4	27.77	6.28	33.27	6.76
NAND2BX4	29.86	7.70	34.07	7.52
NAND2X1	33.19	5.32	37.03	5.58
NAND2X2	32.55	6.13	36.64	6.47
NAND2X4	32.21	6.54	36.95	6.63
NAND2X6	31.76	11.37	37.09	6.75
NAND2X8	31.70	6.07	37.14	7.29
NAND2XL	28.81	5.39	29.86	5.50
NOR2X1	27.42	5.47	32.58	5.39
NOR2X2	28.54	5.92	34.06	5.66
NOR2X4	28.85	6.61	34.25	8.21
NOR2X6	28.78	7.29	34.18	7.47
NOR2X8	28.76	6.51	34.40	6.96
OAI21X4	32.89	6.98	37.63	6.82
XNOR2X1	28.22	5.75	33.06	7.59
XNOR2X4	30.96	4.86	37.99	7.76
XOR2X4	30.87	7.92	37.98	6.85

TABLE V
IMPACT OF GATE-LENGTH BIASING ON LEAKAGE AND DYNAMIC POWER (ASSUMING AN ACTIVITY OF 0.02) FOR SINGLE-THRESHOLD-VOLTAGE DESIGNS. DELAY-PENALTY CONSTRAINT IS SET TO 0%, 2.5%, AND 5% FOR EACH OF THE TEST CASES. (NOTE: DELAY PENALTY FOR SVT-SGL IS ALWAYS SET TO 0% DUE TO THE NONAVAILABILITY OF $V_{\rm th}$ AND $L_{\rm Gate}$ KNOBS. SVT-DGL IS SLOWER THAN SVT-SGL FOR DELAY PENALTIES OF 2.5% AND 5%.)

Test	Delay (ns)	SVT-SGL			SVT-DGL			Reduction			CPU (s)
		Leakage (mW)	Dynamic (mW)	Total (mW)	Leakage (mW)	Dynamic (mW)	Total (mW)	Leakage (%)	Dynamic (%)	Total (%)	
s9234	0.437	0.7074	0.3907	1.0981	0.5023	0.4005	0.9028	28.99	-2.50	17.79	1.81
	0.447	0.7074	0.3907	1.0981	0.5003	0.4006	0.9008	29.28	-2.52	17.96	1.79
	0.458	0.7074	0.3907	1.0981	0.4983	0.4006	0.8988	29.56	-2.51	18.15	1.79
c5315	0.556	1.4413	1.5345	2.9758	1.2552	1.5455	2.8007	12.91	-0.72	5.88	5.60
	0.570	1.4413	1.5345	2.9758	1.0415	1.5585	2.6000	27.74	-1.56	12.63	5.80
	0.584	1.4413	1.5345	2.9758	1.0242	1.5604	2.5846	28.94	-1.69	13.15	5.79
c7552	0.485	1.8328	2.0813	3.9141	1.4447	2.0992	3.5439	21.18	-0.86	9.46	10.97
	0.497	1.8328	2.0813	3.9141	1.3665	2.1042	3.4707	25.44	-1.10	11.33	11.08
	0.509	1.8328	2.0813	3.9141	1.3177	2.1084	3.4261	28.10	-1.30	12.47	10.89
s13207	0.904	1.3934	0.6296	2.0230	0.9845	0.6448	1.6293	29.35	-2.42	19.46	11.46
	0.927	1.3934	0.6296	2.0230	0.9778	0.6449	1.6226	29.83	-2.42	19.79	11.31
	0.949	1.3934	0.6296	2.0230	0.9758	0.6446	1.6204	29.97	-2.39	19.90	11.27
c6288	2.118	3.5994	8.0316	11.6310	3.3391	8.0454	11.3845	7.23	-0.17	2.12	70.51
	2.171	3.5994	8.0316	11.6310	2.8461	8.0931	10.9392	20.93	-0.77	5.95	74.79
	2.224	3.5994	8.0316	11.6310	2.7415	8.1051	10.8466	23.83	-0.92	6.74	70.11
alu128	2.306	5.1571	4.4177	9.5748	4.5051	4.4429	8.9480	12.64	-0.57	6.55	270.00
	2.363	5.1571	4.4177	9.5748	3.5992	4.4818	8.0810	30.21	-1.45	15.60	212.97
	2.421	5.1571	4.4177	9.5748	3.5900	4.4826	8.0726	30.39	-1.47	15.69	211.47
s38417	0.692	4.9381	4.2069	9.1450	3.4847	4.2765	7.7612	29.43	-1.65	15.13	225.18
	0.710	4.9381	4.2069	9.1450	3.4744	4.2778	7.7522	29.64	-1.69	15.23	225.68
	0.727	4.9381	4.2069	9.1450	3.4713	4.2779	7.7492	29.70	-1.69	15.26	221.35

TABLE VI
IMPACT OF GATE-LENGTH BIASING ON LEAKAGE AND DYNAMIC POWER (ASSUMING AN ACTIVITY OF 0.02) FOR DUAL-THRESHOLD-VOLTAGE DESIGNS. DELAY-PENALTY CONSTRAINT IS SET TO 0%, 2.5%, AND 5% FOR EACH OF THE TEST CASES

Test	Delay	DVT-SGL			DVT-DGL			Reduction			CPU
	(*ns*)	Leakage (*mW*)	Dynamic (*mW*)	Total (*mW*)	Leakage (*mW*)	Dynamic (*mW*)	Total (*mW*)	Leakage (%)	Dynamic (%)	Total (%)	(*s*)
s9234	0.437	0.0984	0.3697	0.4681	0.0722	0.3801	0.4523	26.60	-2.81	3.37	1.86
	0.447	0.0914	0.3691	0.4604	0.0650	0.3798	0.4448	28.81	-2.90	3.39	1.89
	0.458	0.0873	0.3676	0.4549	0.0609	0.3784	0.4393	30.20	-2.95	3.41	1.83
c5315	0.556	0.3772	1.4298	1.8070	0.3391	1.4483	1.7874	10.11	-1.29	1.09	5.74
	0.570	0.2871	1.4193	1.7064	0.2485	1.4390	1.6875	13.45	-1.39	1.11	6.21
	0.584	0.2401	1.4119	1.6520	0.1986	1.4328	1.6314	17.27	-1.48	1.24	6.14
c7552	0.485	0.6798	1.9332	2.6130	0.6655	1.9393	2.6048	2.10	-0.32	0.31	10.40
	0.497	0.4698	1.9114	2.3812	0.4478	1.9210	2.3689	4.68	-0.50	0.52	10.51
	0.509	0.3447	1.8994	2.2441	0.3184	1.9107	2.2291	7.63	-0.59	0.67	10.55
s13207	0.904	0.1735	0.5930	0.7664	0.1247	0.6069	0.7316	28.09	-2.35	4.54	11.59
	0.927	0.1561	0.5920	0.7481	0.1066	0.6060	0.7127	31.68	-2.37	4.73	11.73
	0.949	0.1536	0.5919	0.7455	0.1027	0.6060	0.7087	33.14	-2.39	4.93	11.76
c6288	2.118	1.9733	7.7472	9.7205	1.9517	7.7572	9.7089	1.09	-0.13	0.12	79.25
	2.171	1.2258	7.5399	8.7657	1.1880	7.5574	8.7454	3.08	-0.23	0.23	79.25
	2.224	0.8446	7.4160	8.2606	0.8204	7.4283	8.2487	2.87	-0.17	0.14	77.28
alu128	2.306	0.6457	3.9890	4.6347	0.5184	4.0353	4.5537	19.73	-1.16	1.75	240.09
	2.363	0.6151	3.9837	4.5988	0.4970	4.0242	4.5212	19.21	-1.02	1.69	262.37
	2.421	0.5965	3.9817	4.5782	0.4497	4.0378	4.4875	24.62	-1.41	1.98	277.99
s38417	0.692	0.5862	3.8324	4.4186	0.4838	3.8680	4.3518	17.46	-0.93	1.51	238.62
	0.710	0.5637	3.8309	4.3946	0.4189	3.8861	4.3050	25.69	-1.44	2.04	238.99
	0.727	0.5504	3.8306	4.3810	0.4067	3.8849	4.2916	26.11	-1.42	2.04	234.94

TABLE VII
IMPACT OF GATE-LENGTH BIASING ON THE SUBTHRESHOLD LEAKAGE AND GATE TUNNELING LEAKAGE OF 90-nm PMOS AND NMOS DEVICES 1-μm WIDTH AT DIFFERENT TEMPERATURES. TOTAL LEAKAGE REDUCTIONS ARE HIGH EVEN WHEN GATE LEAKAGE IS CONSIDERED

Device	Temp (^oC)	Subthreshold Leakage (*nW*)			Gate Tunneling Leakage (*nW*)			Total Leakage (*nW*)		
		Unbiased	Biased	Reduction	Unbiased	Biased	Reduction	Unbiased	Biased	Reduction
PMOS	25	6.45	4.21	34.73%	2.01	2.03	-1.00%	8.46	6.24	26.24%
NMOS	25	12.68	8.43	33.52%	6.24	6.25	-0.16%	18.92	14.68	22.41%
PMOS	125	116.80	79.91	31.58%	2.17	2.20	-1.38%	118.97	82.11	30.98%
NMOS	125	115.90	83.58	27.89%	6.62	6.69	-1.05%	122.52	90.27	26.32%

dual-$V_{\rm th}$ assignment. Table VI shows results when $L_{\rm Gate}$ biasing is applied together with the dual $V_{\rm th}$ approach. To show the effectiveness of $L_{\rm Gate}$ biasing with loose delay constraints, results when the delay constraint is relaxed are also shown for each circuit. The leakage reductions primarily depend on the slack profile of the circuit. If a lot of paths have near-zero slacks, then the leakage reductions are smaller. As the delay penalty increases, more slack is introduced on paths and larger leakage reductions are seen. We observe that leakage reductions are smaller when the circuit has already been optimized using dual-$V_{\rm th}$ assignment. This is expected because the dual-$V_{\rm th}$ assignment consumes slack on noncritical paths, reducing the slack available for $L_{\rm Gate}$ optimization. We also observe larger leakage reductions in sequential circuits; this is because circuit delay is determined by the slowest pipeline stage and the percentage of noncritical paths is typically higher in sequential circuits.

Our leakage models do not include gate leakage, which can marginally increase due to biasing. Gate leakage is composed of gate-length dependent [gate-to-channel $(I_{\rm gc})$ and gate-to-body $(I_{\rm gb})$ tunneling] and independent components [edge direct tunneling $(I_{\rm gs}+I_{\rm gd})$]. The gate-length independent component, which stems from the gate–drain and gate–source overlap regions, is not affected by biasing. To assess the change in gate-length dependent components due to biasing, we perform SPICE simulations to report the gate-to-channel leakage[8] for both nominal and biased devices. We use 90-nm BSIM4 device models from a leading foundry that model all five components of gate leakage described in BSIM v4.4.0. Table VII shows the gate and subthreshold leakage for biased and unbiased nominal $V_{\rm th}$ NMOS and PMOS devices of 1-μm width at 25 °C and 125 °C. The reductions in subthreshold and gate leakage as well as the total leakage reduction are shown. Based on these results, we conclude that the increase in gate leakage due to biasing is negligible. Furthermore, since biasing is a runtime-leakage-reduction approach, the operating temperature is likely to be higher than room temperature—in this scenario, gate leakage is not a major portion of the total leakage. When the operating temperature is elevated, the reduction in total leakage is approximately equal to the reduction in the subthreshold leakage, and total leakage reductions similar to the results presented in Tables V and VI are expected.[9] Gate leakage is predicted to increase with technology scaling; technologies under 65 nm, however, are likely to adopt high-k gate dielectrics, which

[8]The gate-to-body component is two orders of magnitude smaller than the gate-to-channel component and it is therefore excluded from this analysis.

[9]We report subthreshold leakage at 25 °C. Although the subthreshold leakage itself increases significantly with temperature, the percentage reduction in it due to biasing does not change much.

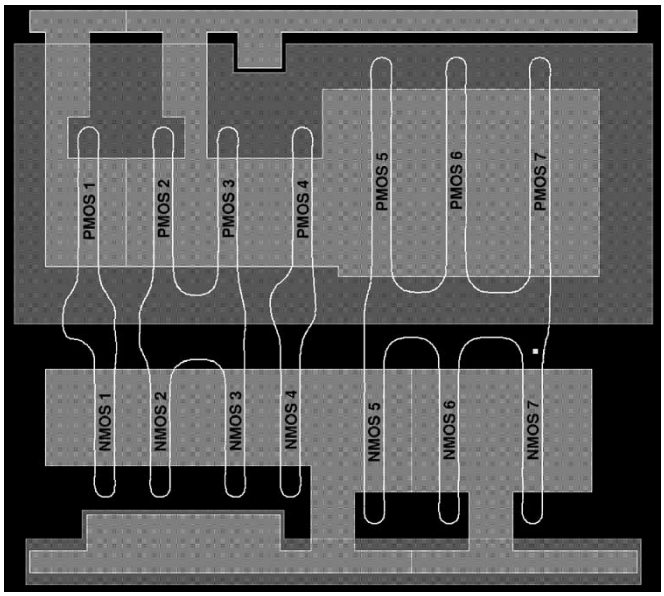

Fig. 4. Cell layout of a generic AND2X6 with simulated printed gate lengths.

will tremendously reduce gate leakage. Therefore, in terms of scalability, subthreshold leakage remains the key problem at high operating temperatures. We also note that because the vertical electric fields do not increase due to biasing, negative-bias thermal instability (NBTI) is not expected to increase with biasing [25].

B. Manufacturability and Process Effects

In this section, we investigate the manufacturability and process variability implications of our $L_{\rm Gate}$-biasing approach. As our method relies on the biasing of the drawn gate length, it is important to correlate this with the actual printed gate length on the wafer. This is even more important as the bias we introduce in the gate length is of the same order as the typical critical dimension (CD) tolerances in the manufacturing processes. Moreover, we expect larger gate lengths to have better printability properties, leading to less CD—and hence leakage—variability. To validate our multiple gate-length approach in a postmanufacturing setup, we follow a reticle enhancement technology (RET) and process a simulation flow for an example cell master.

We use the layout of a generic AND2X6 cell and perform a model-based optical proximity correction (OPC) on it using Calibre v9.3_2.5 [1].[10] The printed image of the cell is then calculated using a *dense* simulation in Calibre. The layout of the cell, along with printed gate lengths of all devices in it, is shown in Fig. 4. We measure the $L_{\rm Gate}$ for every device in the cell, for both biased and unbiased versions. The printed gate lengths for the seven NMOS and PMOS devices labeled in Fig. 4 are shown in Table VIII. As expected, biased and unbiased gate lengths track each other well. There are some outliers that may be due to the relative simplicity of the OPC model being used. High correlation between the printed dimensions of the biased and unbiased versions of the cells show that the benefits of biasing estimation using the drawn dimensions will not be lost after the RET application and the manufacturing process.

TABLE VIII
COMPARISON OF PRINTED DIMENSIONS OF UNBIASED AND BIASED VERSIONS OF AND2X6. THE UNBIASED NOMINAL GATE LENGTH IS 130 nm WHILE THE BIASED NOMINAL IS 138 nm. NOTE THE HIGH CORRELATION BETWEEN UNBIASED AND BIASED VERSIONS

Device Number	Gate Length (*nm*)					
	PMOS			NMOS		
	Unbiased	Biased	Diff.	Unbiased	Biased	Diff.
1	128	135	+7	129	135	+6
2	127	131	+4	126	131	+5
3	127	131	+4	127	131	+4
4	124	131	+7	126	133	+7
5	124	131	+7	124	132	+8
6	124	132	+8	124	132	+8
7	127	135	+8	127	135	+8

TABLE IX
PROCESS-WINDOW IMPROVEMENT WITH GATE-LENGTH BIASING. THE CD TOLERANCE IS KEPT AT 13 nm. ELAT = EXPOSURE LATITUDE

Defocus (μm)	ELAT (%) for 130*nm*	ELAT (%) for 138*nm*
-0.2	4.93	5.30
0.0	6.75	7.26
0.2	5.69	6.24

Another potentially valuable benefit of slightly larger gate lengths is the possibility of improved printability. Minimum poly spacing is larger than the poly gate length, so that the process window (which is constrained by the minimum resolvable dimension) tends to be larger as the gate length increases even though the poly spacing decreases. For example, the depth of focus for various values of exposure latitude with the same illumination system as above for 130- and 138-nm lines is shown in Table IX.[11]

C. Process Variability

A number of sources of variation can cause fluctuations in the gate length, and hence, in performance and leakage. This has been a subject of much discussion in the recent literature (e.g., [8] and [23]). Up to $20\times$ variation in leakage has been reported in the production of microprocessors [7]. For leakage, the reduction in variation post biasing is likely to be substantial as the larger gate length is closer to the "flatter" region of the $V_{\rm th}$ versus the $L_{\rm Gate}$ curve. To validate this intuition, we study the impact of gate-length variation on leakage and performance both pre- and postbiasing using a simple worst case approach. We assume the CD variation budget to be ± 10 nm. The performance and leakage of the test-case circuits is measured at the worst case, nominal, and best case process corners, which consider only the gate-length variation. This is done for the DVT-DGL approach in which the biasing is done along with the dual $V_{\rm th}$ assignment. The results are shown in Table X. For the seven test cases, we see up to a 41% reduction in leakage-power uncertainty caused by linewidth variation. Such large reductions in uncertainty can potentially outweigh the benefits of alternative leakage-control techniques. We note that the corner case analysis only models the inter-die component

[10]Model-based OPC is performed using annular optical illumination with $\lambda = 248$ nm and NA $= 0.7$.

[11]The process simulation was performed using Prolith v8.1.2 [3].

TABLE X
REDUCTION IN PERFORMANCE AND LEAKAGE-POWER UNCERTAINTY WITH BIASED GATE LENGTH IN THE PRESENCE OF INTER-DIE VARIATIONS. THE UNCERTAINTY SPREAD IS SPECIFIED AS A PERCENTAGE OF NOMINAL. THE RESULTS ARE GIVEN FOR DUAL $V_{\rm th}$ AND THE BIASING IS 8 nm

Circuit	Circuit Delay (ns)						
	Unbiased (DVT-SGL)			Biased (DVT-DGL)			% Spread
	BC	WC	NOM	BC	WC	NOM	Reduction
s9234	0.504	0.385	0.436	0.506	0.387	0.436	-0.53
c5315	0.642	0.499	0.556	0.643	0.501	0.556	0.71
c7552	0.559	0.433	0.485	0.559	0.433	0.485	0.46
s13207	1.029	0.797	0.904	1.031	0.800	0.904	0.35
c6288	2.411	1.888	2.118	2.411	1.889	2.118	0.13
alu128	2.631	2.045	2.305	2.640	2.053	2.306	-0.10
s38417	0.793	0.615	0.692	0.793	0.616	0.692	0.03
Circuit	Leakage (mW)						
	Unbiased (DVT-SGL)			Biased (DVT-DGL)			% Spread
	BC	WC	NOM	BC	WC	NOM	Reduction
s9234	0.0591	0.1898	0.0984	0.0467	0.1268	0.0722	38.76
c5315	0.2358	0.6883	0.3772	0.2176	0.5960	0.3391	16.38
c7552	0.4291	1.2171	0.6798	0.4226	1.1825	0.6655	3.57
s13207	0.1036	0.3401	0.1735	0.0807	0.2211	0.1247	40.65
c6288	1.2477	3.5081	1.9733	1.2373	3.4559	1.9517	1.85
alu128	0.3827	1.2858	0.6457	0.3229	0.9641	0.5184	29.00
s38417	0.3526	1.1453	0.5862	0.3038	0.8966	0.4838	25.22

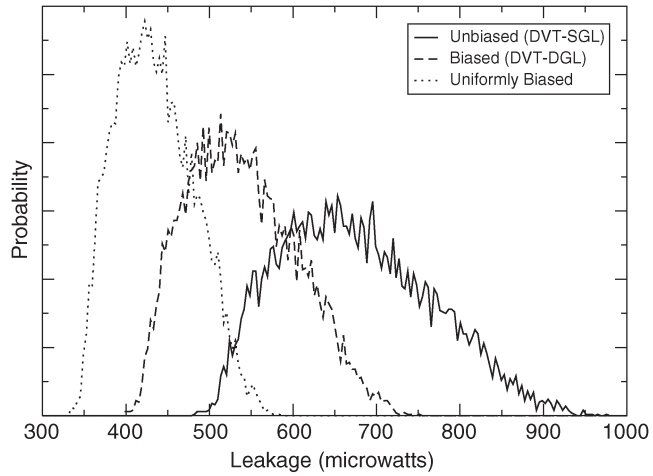

Fig. 5. Leakage distributions for unbiased, uniform-biased, and technology level selectively biased alu128. Note the "left shift" of the distribution with the introduction of biased devices in the design.

of the variation, which typically constitutes roughly half of the total CD variation.

To assess the impact of both within-die (WID) and die-to-die (DTD) components of variation, we run 10 000 Monte Carlo simulations with $\sigma_{\rm WID} = \sigma_{\rm DTD} = 3.33$ nm. The variations are assumed to follow a Gaussian distribution with no correlations. We compare the results for three dual $V_{\rm th}$ scenarios: 1) unbiased (DVT-SGL); 2) biased (DVT-DGL); and 3) uniformly biased (when gate lengths of all transistors in the design are biased by 8 nm). Leakage distributions for the test case alu128 are shown in Fig. 5. Note that in uniform biasing, all devices are biased and the circuit delay no longer meets timing.

D. Leakage Reduction From Transistor-Level Gate-Length Biasing

Table XI presents the leakage-power reductions from TLLB over CLLB. We see up to a 10% reduction in leakage power

TABLE XI
LEAKAGE POWER FROM TRANSISTOR-LEVEL GATE-LENGTH BIASING

Circuit	Delay	Leakage			CPU (s)	
		CLLB	TLLB	Reduction	CLLB	TLLB
	(ns)	(mW)	(mW)	(%)	(s)	(s)
s9234	0.437	0.0722	0.0712	1.41	1.86	2.75
	0.447	0.0650	0.0628	3.39	1.89	2.38
	0.458	0.0609	0.0596	2.28	1.83	2.31
c5315	0.556	0.3391	0.3359	0.95	5.74	14.99
	0.570	0.2485	0.2368	4.71	6.21	15.29
	0.584	0.1986	0.1918	3.42	6.14	13.44
c7552	0.485	0.6655	0.6356	4.49	10.40	43.79
	0.497	0.4478	0.4438	0.89	10.51	43.22
	0.509	0.3184	0.2993	6.02	10.55	38.90
s13207	0.904	0.1247	0.1228	1.58	11.59	17.15
	0.927	0.1066	0.1055	1.08	11.73	15.62
	0.949	0.1027	0.1021	0.61	11.76	14.28
c6288	2.118	1.9517	1.9157	1.84	79.25	305.09
	2.171	1.1880	1.1555	2.74	79.46	289.56
	2.224	0.8203	0.8203	0.00	77.28	291.44
alu128	2.306	0.5184	0.4857	6.31	240.09	544.75
	2.363	0.4970	0.4492	9.62	262.37	609.13
	2.421	0.4497	0.4184	6.95	277.99	534.68
s38417	0.692	0.4838	0.4467	7.67	238.62	746.79
	0.710	0.4189	0.3982	4.93	238.99	507.62
	0.727	0.4067	0.3765	7.42	234.94	525.06

over CLLB. Since TLLB only biases devices of the unbiased cells, it performs well over CLLB when CLLB does not perform well (i.e., when CLLB leaves many cells unbiased). The leakage savings from TLLB come at the cost of increased library size. As described in Section III-A, the library is composed of all 2^{2n} variants of each n-input cell. For the 25 cells, our library for TLLB was composed of a total of 920 variants. From the small leakage savings at the cost of significantly increased library size, we conclude that TLLB should only be performed for single- and double-input cells that are frequently used.

V. CONCLUSION AND ONGOING STUDIES

We have presented a novel methodology that selectively used small $L_{\rm Gate}$ biases to achieve an easily manufacturable approach to runtime-leakage reduction. For our test cases, we have observed the following.

1) The gate-length bias we propose is always less than the pitch of the layout grid; this avoids design-rule violations. Moreover, it implies that the biased and unbiased cell layouts are completely pin compatible, and hence, layout swappable. This allows biasing-based leakage optimization to be possible at any point in the design flow, unlike in sizing-based methods.
2) With a biasing of 8 nm in a 130 nm process, leakage reductions of 24%–38% are achieved for the most commonly used cells with a delay penalty of fewer than 10%.
3) Using simple sizing techniques, we are able to achieve up to 33% leakage savings with less than 3% dynamic power overhead and no delay penalty. The use of more than two gate lengths for the most commonly used cells, along with improved sizing techniques, is likely to yield better leakage savings.
4) We compared gate-length biasing at the cell level and at the transistor level. Transistor-level gate-length biasing

can further reduce leakage by up to 10%, but requires a significantly larger library. Therefore, transistor-level biasing should be done for only the most frequently used cells such as inverters, buffers, NAND, and NOR gates. Fortunately, the most frequently used cells have one or two inputs and hence, only a small number of transistor-level biasing variants need to be characterized for them. For cells with three or more inputs, no transistor-level biasing variants may be created (i.e., only cell-level biasing variants are created). To further reduce the library size, only one of the cell variants in which different logically equivalent inputs are fast may be retained, and pin-swapping techniques can be used during leakage optimization.

5) The devices with biased gate lengths are more manufacturable and have a larger process margin than the nominal devices. Biasing does not require any extra process steps, unlike multiple-threshold-based leakage-optimization methods.
6) L_{Gate} biasing leads to more process-insensitive designs with respect to the leakage current. Biased designs have up to 41% less worst case leakage variability in the presence of inter-die variations, as compared to the nominal gate-length designs. In the presence of both inter- and intra-die CD variations, selective L_{Gate} biasing can yield designs less sensitive to variations.

Our ongoing study is along the following directions.

1) Construction of effective biasing-based leakage-optimization heuristics. To increase scalability, we plan to investigate "batched" moves in which several independent cells or transistors are biased in every iteration.
2) Assessment of leakage savings from the use of more than two gate lengths for the more frequently used and leaky cells in the library, such as inverters and buffers. Also, the development of better approaches to reduce the cell library size.
3) Evaluation of the impact of biasing on leakage at future technology nodes for which leakage is a much bigger issue than it is at 130 nm.

ACKNOWLEDGMENT

A preliminary version of this paper appeared in [9].

REFERENCES

[1] *Mentor Calibre*. [Online]. Available: http://mentor.com/calibre/datasheets/opc/html/
[2] *Opencores.org*. [Online]. Available: http://www.opencores.org/projects/
[3] *Prolith*. [Online]. Available: http://www.kla-tencor.com
[4] A. Agarwal, C. H. Kim, S. Mukhopadhyay, and K. Roy, "Leakage in nano-scale technologies: Mechanisms, impact and design considerations," in *Proc. ACM/IEEE Design Automation Conf.*, San Diego, CA, 2004, pp. 6–11.
[5] F. Beeftink, P. Kudva, D. Kung, and L. Stok, "Combinatorial cell design for CMOS libraries," *Integr. VLSI J.*, vol. 29, no. 4, pp. 67–93, Feb. 2000.
[6] F. Boeuf *et al.*, "A conventional 45 nm CMOS node low-cost platform for general purpose and low power applications," in *IEEE Int. Electron Devices Meeting*, San Francisco, CA, 2004, pp. 425–428.
[7] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impact on circuits and microarchitecture," in *Proc. ACM/IEEE Design Automation Conf.*, Anaheim, CA, 2003, pp. 338–342.
[8] Y. Cao, P. Gupta, A. B. Kahng, D. Sylvester, and J. Yang, "Design sensitivities to variability: Extrapolations and assessments in nanometer VLSI," in *Proc. IEEE ASIC/SOC*, Rochester, NY, 2002, pp. 411–415.
[9] P. Gupta, A. B. Kahng, P. Sharma, and D. Sylvester, "Selective gate-length biasing for cost-effective runtime leakage control," in *Proc. ACM/IEEE Design Automation Conf.*, San Diego, CA, 2004, pp. 327–330.
[10] J. Halter and F. Najm, "A gate-level leakage power reduction method for ultra low power CMOS circuits," in *IEEE Custom Integrated Circuits Conf.*, Santa Clara, CA, 1997, pp. 475–478.
[11] M. Horiguchi, T. Sakata, and K. Itoh, "Switched-source-impedance CMOS circuit for low standby sub-threshold current giga-scale LSI's," *IEEE J. Solid-State Circuits*, vol. 28, no. 11, pp. 1131–1135, Nov. 1993.
[12] I. Hyunsik, T. Inukai, H. Gomyo, T. Hiramoto, and T. Sakurai, "VTC-MOS characteristics and its optimum conditions predicted by a compact analytical model," in *Proc. Int. Symp. Low Power Electronics and Design*, Huntington Beach, CA, 2001, pp. 123–128.
[13] J. Kao, S. Narendra, and A. Chandrakasan, "MTCMOS hierarchical sizing based on mutual exclusive discharge patterns," in *Proc. ACM/IEEE Design Automation Conf.*, San Francisco, CA, 1998, pp. 495–500.
[14] M. Ketkar and S. Saptnekar, "Standby power optimization via transistor sizing and dual threshold voltage assignment," in *Proc. IEEE Int. Conf. Computer-Aided Design*, San Jose, CA, 2002, pp. 375–378.
[15] D. Lee and D. Blaauw, "Static leakage reduction through simultaneous threshold voltage and state assignment," in *Proc. ACM/IEEE Design Automation Conf.*, Anaheim, CA, 2003, pp. 192–194.
[16] Z. Luo, "High performance and low power transistors integrated in 65 nm bulk CMOS technology," in *Proc. IEEE Int. Electron Devices Meeting*, San Francisco, CA, 2004, pp. 661–664.
[17] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, "1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS," *IEEE J. Solid-State Circuits*, vol. 30, no. 8, pp. 847–854, Aug. 1995.
[18] S. Mutoh, S. Shigematsu, Y. Matsuya, H. Fukada, T. Kaneko, and J. Yamada, "1 V multithreshold-voltage CMOS digital signal processor for mobile phone application," *IEEE J. Solid-State Circuits*, vol. 31, no. 11, pp. 1795–1802, Nov. 1996.
[19] Y. Nakahara *et al.*, "A robust 65-nm node CMOS technology for wide-range Vdd operation," in *Proc. IEEE Int. Electron Devices Meeting*, Washington, DC, 2003, pp. 11.2.1–11.2.4.
[20] S. Nakai *et al.*, "A 65 nm CMOS technology with a high-performance and low-leakage transistor, a 0.55 μm^2 6T-SRAM cell and robust hybrid-ULK/Cu interconnects for mobile multimedia applications," in *Proc. IEEE Int. Electron Devices Meeting*, Washington, DC, 2003, pp. 11.3.1–11.3.4.
[21] S. Narendra, D. Blaauw, A. Devgan, and F. Najm "Leakage issues in IC design: Trends, estimation and avoidance" Tutorial presented at IEEE Int. Conf. Comp.-Aided Des., San Jose, CA, 2003.
[22] K. Nose, M. Hirabayashi, H. Kawaguchi, S. Lee, and T. Sakurai, "V_{th} hopping scheme to reduce subthreshold leakage for low-power processors," *IEEE J. Solid-State Circuits*, vol. 37, no. 3, pp. 413–419, Mar. 2002.
[23] R. Rao, A. Srivastava, D. Blaauw, and D. Sylvester, "Statistical analysis of subthreshold leakage current for VLSI circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 2, pp. 131–139, Feb. 2004.
[24] P. Royannez *et al.*, "90 nm low leakage SoC design techniques for wireless applications," in *Proc. IEEE Int. Solid-State Circuits Conf.*, San Francisco, CA, 2005, pp. 138–589.
[25] D. K. Schroder and J. A. Babcock, "Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing," *J. Appl. Phys.*, vol. 94, no. 1, pp. 1–18, Jul. 2003.
[26] S. Shigematsu, S. Mutoh, Y. Matsuya, Y. Tabae, and J. Yamada, "A 1-V high-speed MTCMOS circuit scheme for power-down application circuits," *IEEE J. Solid-State Circuits*, vol. 32, no. 6, pp. 861–869, Jun. 1997.
[27] S. Sirichotiyakul, T. Edwards, C. Oh, R. Panda, and D. Blaauw, "Duet: An accurate leakage estimation and optimization tool for dual-V_{th} circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 10, no. 2, pp. 79–90, Apr. 2002.
[28] S. Sirichotiyakul, T. Edwards, C. Oh, J. Zuo, A. Dharchoudhury, R. Panda, and D. Blaauw, "Stand-by power minimization through simultaneous threshold voltage selection and circuit sizing," in *Proc. ACM/IEEE Design Automation Conf.*, New Orleans, LA, 1999, pp. 436–441.
[29] N. Sirisantana, L. Wei, and K. Roy, "High-performance low-power CMOS circuits using multiple channel length and multiple oxide thickness," in *Proc. Int. Conf. Computer Design*, Austin, TX, 2000, pp. 227–232.

[30] L. Wei, Z. Chen, M. Johnson, K. Roy, and V. De, "Design and optimization of low voltage high performance dual threshold CMOS circuits," in *Proc. ACM/IEEE Design Automation Conf.*, San Francisco, CA, 1998, pp. 489–494.
[31] L. Wei, Z. Chen, and K. Roy, "Mixed-$V_{\rm th}$ CMOS circuit design methodology for low power applications," in *Proc. ACM/IEEE Design Automation Conf.*, New Orleans, LA, 1999, pp. 430–435.
[32] L. Wei, K. Roy, and C. K. Koh, "Power minimization by simultaneous dual-V_{th} assignment and gate-sizing," in *Proc. ACM/IEEE Design Automation Conf.*, Los Angeles, CA, 2000, pp. 413–416.
[33] Y. Ye, S. Borkar, and V. De, "A new technique for standby leakage reduction in high-performance circuits," in *Proc. Symp. VLSI Circuits*, Honolulu, HI, 1998, pp. 40–41.

Puneet Gupta (S'01) received the Bachelor's degree in electrical engineering from the Indian Institute of Technology, Delhi, India in 2000. He is currently working toward the Ph.D. degree at the University of California at San Diego, La Jolla.

He is a cofounder of Blaze DFM Inc. His research interests lie primarily in the design–manufacturing interface. He has previously copresented an embedded tutorial on Manufacturing Aware Physical Design at the International Conference on Computer-Aided Design (ICCAD) 2003. He is the author/coauthor of over 30 papers and is the holder of seven patents.

Mr. Gupta is a recipient of the IBM Ph.D. fellowship.

Andrew B. Kahng (A'89–M'03) received the A.B. degree in applied mathematics from Harvard College, Cambridge, MA, in 1986, and the M.S. and Ph.D. degrees in computer science from the University of California at San Diego, La Jolla in 1989.

He was a member of the University of California, Los Angeles (UCLA) Computer Science faculty from 1989 to 2000. He is a Professor at the Computer Science and Engineering and the Electronics and Computer Engineering Departments at the University of California. His research is mainly in physical design and performance analysis of very large scale integration (VLSI), as well as the VLSI design–manufacturing interface. Other research interests include combinatorial and graph algorithms, and large-scale heuristic global optimization. Since 1997, he has defined the physical design roadmap for the International Technology Roadmap for Semiconductors (ITRS), and has chaired the U.S. and international working groups for design technology for the ITRS since 2001. He has been active in the Microelectronics Advanced Research Corporation (MARCO) Gigascale Silicon Research Center since its inception. He has authored/coauthored over 200 papers in the VLSI Computer Aided Design (VLSI CAD) literature.

Dr. Kahng was the founding General Chair of the Association for Computing Machinery/IEEE (ACM/IEEE) International Symposium on Physical Design, and cofounded the ACM Workshop on System-Level Interconnect Planning. He has received three Best Paper awards and a National Science Foundation (NSF) Young Investigator Award.

Puneet Sharma (M'05) received the Bachelor of Technology degree in computer science and engineering from the Indian Institute of Technology, Delhi, India, in 2002. He has been working toward the Ph.D. degree at the University of California at San Diego, La Jolla, since September 2002.

His research interests include leakage-power reduction and design for manufacturing.

Dennis Sylvester (S'96–M'97–SM'04) received the B.S. degree in electrical engineering (*summa cum laude*) from the University of Michigan, Ann Arbor, in 1995, and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, in 1997 and 1999, respectively.

He worked at Hewlett-Packard Laboratories, Palo Alto, CA, from 1996 to 1998. After working in the Advanced Technology Group of Synopsys, Mountain View, CA, he is now an Associate Professor of Electrical Engineering at the University of Michigan. He has authored/coauthored numerous articles along with a book and several book chapters in his field of research, which includes low-power circuit design and design-automation techniques, design for manufacturability, and on-chip interconnect modeling. He also serves as a consultant and technical advisory board member for several electronic design automation firms in these areas. He also helped define the circuit and physical design roadmap as a member of the International Technology Roadmap for Semiconductors (ITRS) U.S. Design Technology Working Group from 2001 to 2003. He has served on the technical program committee of numerous design automation and circuit design conferences and was General Chair of the 2003 ACM/IEEE System-Level Interconnect Prediction (SLIP) Workshop and 2005 ACM/IEEE Workshop on Timing Issues in the Synthesis and Specification of Digital Systems (TAU).

Dr. Sylvester received a National Science Foundation (NSF) CAREER award, the 2000 Beatrice Winner Award at the International Solid-State Circuits Conference (ISSCC), a 2004 IBM Faculty Award, and several best paper awards and nominations. He is the recipient of the 2003 Association for Computing Machinery Special Interest Group on Design Automation (ACM SIGDA) Outstanding New Faculty Award, the 1938E Award from the College of Engineering for teaching and mentoring, and the Henry Russel Award, which is the highest award given to faculty at the University of Michigan. His dissertation research was recognized with the 2000 David J. Sakrison Memorial Prize as the most outstanding research in the UC-Berkeley Electrical Engineering and Computer Sciences Department. He is an Associate Editor for IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS. Dr. Sylvester is a member of ACM, American Society of Engineering Education, and Eta Kappa Nu.