

Gate-Length Biasing for Runtime Leakage Control

Puneet Gupta Andrew B. Kahng Puneet Sharma Dennis Sylvester

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 leakage power and therefore cause its share of total power to increase. Manufacturers face the additional challenge of *leakage variability*: recent data indicates that leakage of microprocessor chips from a single 180nm wafer can vary by as much as 20×. Previously proposed techniques for leakage power reduction include use of multiple supply and gate threshold voltages, and assignment of input values to inactive gates such that leakage is minimized.

We study the additional design space afforded by biasing of device gate-lengths to reduce chip leakage power and its variability. 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 pin- and layout-compatibility; therefore, our technique can be applied as a post-layout enhancement step. We 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, we first apply the multi-threshold voltage technique which is widely used for leakage reduction, and then use gate-length biasing to show further reduction in leakage.

Experimental results show that gate-length biasing reduces leakage by 24% to 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 manufacturing yield and product cost. We also assess the use of gate-length biasing for leakage optimization of cell instances in which: (1) not all timing arcs are timing-critical, and/or (2) rise and fall transitions are not both timing-critical at the same time.

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 CMOS circuits consists of dynamic and static (due to leakage currents) components. Leakage is becoming an ever-increasing component of total dissipated power with its contribution projected to increase from 18% at 130nm to 54% at the 65nm 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 total leakage at 130nm and is forecast to remain so in the future [4]. In this work we present a novel approach for subthreshold leakage reduction.

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

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. E-mail: puneet@ucsd.edu, sharma@ucsd.edu.

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

D. Sylvester is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109. Email: dmcs@eecs.umich.edu.

Leakage reduction methodologies can be divided into two classes depending on whether they reduce *standby* leakage or *runtime* leakage. Standby techniques reduce leakage of devices that are known not to be in operation while runtime techniques reduce leakage of active devices. Several techniques have been proposed for standby leakage reduction. *Body biasing* or *VTMOS* based approaches [12] dynamically adjust the device V_{th} by biasing the body terminal¹. *Multi-threshold CMOS (MTCMOS)* techniques [17], [13], [18], [26] use high- V_{th} CMOS (or NMOS or PMOS) to disconnect Vdd or Vss or both to logic circuit implemented using low V_{th} devices in standby mode. *Source biasing*, where a positive bias is applied in standby state to source terminals of off devices, was proposed in [11]. Other techniques such as 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_{th} manufacturing process. In this approach, cells in non-critical paths are assigned a high V_{th} while cells in critical paths are assigned a low V_{th} . [30] presented a heuristic algorithm for selection and assignment of an optimal high V_{th} to cells on non-critical paths. The multi- V_{th} approach has also been combined with several other power reduction techniques [15], [32], [27]. The primary drawback to this technique has traditionally been the rise in process costs due to additional steps and masks. However, the increased costs have been outweighed by the resulting substantial leakage reductions, and multi- V_{th} processes are now standard. A new complication facing multi- V_{th} is the increased variability of V_{th} for low- V_{th} devices. This occurs in part due to random doping fluctuations, as well as worsened DIBL (Drain Induced Barrier Lowering) and short-channel effects (SCE) in devices with lower channel doping. The larger variability in V_{th} degrades the achievable leakage reductions of multi- V_{th} and worsens with continued MOS scaling. Moreover, multi- V_{th} methodologies do not offer a smooth tradeoff between performance and leakage power. Devices with different V_{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_{th} devices.

The use of longer gate-lengths (L_{Gate}) in devices within non-critical gates was first described in [29]. In that work, *large* changes to 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 ECO (Engineering Change Order) overheads during layout. In this paper, we propose very *small* increases in gate-length for non-critical 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 90nm node and below employ super-halo doping, giving rise to reverse short channel effects (RSCE) that mitigate traditional SCE to some extent. However, we have found the proposed technique to substantially reduce leakage for the two 130nm and two 90nm industrial processes that we investigated. Recent reports from leading integrated device manufacturers (IDMs) indicate SCE continues to dominate V_{th} roll-off characteristics at the 65nm and 45nm technology nodes [19], [20], [6], [16]. However, we note that the V_{th} roll-off curve must be understood to assess the feasibility of this approach and to determine reasonable increases for gate length.

The variation of delay and leakage with gate-length is shown in Figure 1 for an industrial 130nm process. Leakage current flattens out with gate-length beyond 140nm, making L_{Gate} biasing less desirable in that range. Another major

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

advantage of L_{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 reduced variability in leakage. We use the terms *gate-length biasing* and L_{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_{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 timing criticality of different timing arcs of a cell instance in a circuit, and that of rise and fall transitions, can be used by TLLB to yield significant leakage savings. [31], [28], [14] proposed transistor-level V_{th} assignment for leakage power reduction. Our approach uses L_{Gate} biasing instead of V_{th} assignment and is similar to [31]. The major disadvantage of TLLB (or V_{th} assignment) is the increase in library size and its characterization time.

Contributions of our work include the following.

- A leakage reduction methodology based on less than 10% increase in drawn L_{Gate} of devices.
- A thorough analysis of potential benefits and caveats of such a biasing methodology, including implications of lithography and process variability.
- Experiments and results showing potential benefits of an L_{Gate} biasing methodology in different design scenarios such as dual- V_{th} .

The organization of this paper is as follows. In the next section, we describe the proposed L_{Gate} biasing methodology for leakage reduction. Section 3 extends the ideas to be applied at the transistor level for further reduction of leakage at the cost of increased library size. Section 4 gives experiments and results for validation of the proposed ideas. It also analyzes the potential manufacturing and process variation implications of biasing gate-lengths. Finally, Section 5 concludes with a brief description of ongoing research.

II. CELL-LEVEL GATE-LENGTH BIASING

In this section we describe the proposed cell-level L_{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 non-critical 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 cases². 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:

- The nominal gate-length of the technology is usually very close to or beyond the “knee” of the leakage vs. L_{Gate} curve which arises due to SCE. For large bias, the advantage of super-linear dependence of leakage on gate-length is lost. Moreover, dynamic power and delay both increase almost linearly with gate-length. Therefore, small biases give more “bang for the buck”.

²We first synthesize our test cases with the complete Artisan TSMC library to identify the most frequently used cells.

- 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 printability properties (i.e., size of process window). We retain the same poly-pitch 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 DRC tools first snap to grid, biases of under 10% are not detected and are considered acceptable due to margins in design rules.
- An increase in drawn dimension that is less than the layout grid resolution (typically $10nm$ for $130nm$ 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 SPICE models we use, the nominal gate-length of all transistors is $130nm$. In our approach, all transistors in a biased variant of a cell have a gate-length of $138nm$. We choose $138nm$ as the biased gate-length because it places the delay of low- V_{th} -biased variant between the low- V_{th} -nominal gate-length variant and the nominal- V_{th} -nominal gate-length variant. Larger bias can lead to 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 above mentioned approach for choosing the biased gate-length to work well for all designs. We note that this value of $138nm$ is highly process specific and is not intended to reflect the best biased gate-length for all $130nm$ 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 above³.

An important component of the methodology is layout and 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-SPI*. Since delay is almost always the primary design goal, we perform sizing to achieve the minimum possible delay.

³We have recently been informed that a major U.S. semiconductor manufacturer has started to offer its customers a cell-wise L_{Gate} biased variant of its $90nm$ cell library with a $6nm$ 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 $8nm$ bias for a $130nm$ cell library provides a practically relevant testbed.

We use a sensitivity-based, *downsizing* (i.e., begin with all nominal cell variants and replace cells on non-critical 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 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 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, however, 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 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.

- *Standard static timing analysis (SSTA)*. Slews and actual arrival times (AATs) are propagated forward after a topological ordering of the circuit. Required arrival times (RATs) are back-propagated and slacks are then computed. Slew, delay and slack values of our timer match exactly with *Synopsys PrimeTime vU-2003.03-SP2* and our timer can handle unate and non-unate cells ⁴.
- *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. Slews, delays and slacks match exactly with SSTA.
- *Constrained incremental STA (CISTA)*. Sensitivity computation involves temporary modifications to a cell to find change in its slack and leakage. To make this step faster, we restrict the incremental timing calculation to only one stage before and after the gate being modified. The next stage is affected by 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 changes⁵) 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 Figure 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 $P_{p^*} \geq 0$.

⁴Delay 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, 3-pin SDFs, etc.

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

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 slew as well as 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 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 cell instance q and finds its slack using CISTA. Since high accuracy is not critical for 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.

III. TRANSISTOR-LEVEL GATE-LENGTH BIASING

We use the term *timing arc* to indicate an intra-cell path from an input transition to a resulting rise (or fall) output transition. For an n -input gate there are $2n$ timing arcs⁶. 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 130nm library. Pin swapping is a common post-synthesis 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 P/N ratio perturbations have been previously proposed to decrease circuit delay [5]. We propose to exploit these asymmetries using TLLB to “recover” leakage from non-critical timing arcs within a cell.

A. Library Generation

For each cell, our library contains 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 biased L_{Gate} to some transistors in the cell and 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. Figure 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.

⁶There may be four timing arcs corresponding to non-unate inputs (e.g., select input of MUX).

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 validation of the L_{Gate} biasing methodology, and present experimental results. Details of the test cases⁷ used in our experiments are given in Table III. The test cases are synthesized with the *Artisan TSMC 130nm library* using *Synopsys Design Compiler v2003.06-SP1* with low- V_{th} cells only. To limit library characterization runtime, we restrict the library to variants of the following 25 most frequently used cells: CLKINX1, 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 post-synthesis delay is close to 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 130nm* device models are used with the two V_{th} values each for PMOS and NMOS transistors (PMOS: -0.09V and -0.17V; NMOS: 0.11V and 0.19V). We use *Cadence SignalStorm v4.1* (with *Synopsys HSPICE*) for delay and power characterization of cell variants. *Synopsys Design Compiler* is used to measure circuit delay, dynamic power and leakage power. We assume an activity factor of 0.02 for 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 1.4GHz computer with 2GB of RAM.

A. Leakage Reduction

Table IV shows the leakage savings and delay penalties due to L_{Gate} biasing for all cells in our library. The results strongly support our hypothesis that small biases in L_{Gate} can afford significant leakage savings with 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 non-critical 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 dual- V_{th} assignment. Table VI shows results when L_{Gate} biasing is applied together with the dual V_{th} approach. To show the effectiveness of L_{Gate} biasing with loose delay constraints, results when the delay constraint is relaxed are

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

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_{th} assignment. This is expected because dual- V_{th} assignment consumes slack on non-critical paths reducing the slack available for L_{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 non-critical 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_{gc}) and gate-to-body (I_{gb}) tunneling) and independent components (edge direct tunneling ($I_{gs} + I_{gd}$)). 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 leakage⁸ for nominal and biased devices. We use 90nm *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_{th} NMOS and PMOS devices of $1\mu m$ width at $25^\circ C$ and $125^\circ C$. 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 total leakage. When the operating temperature is elevated, the reduction in total leakage is approximately equal to the reduction in subthreshold leakage and total leakage reductions similar to the results presented in Tables V and VI are expected⁹. Gate leakage is predicted to increase with technology scaling; technologies under $65nm$, however, are likely to adopt high-k gate dielectrics which will tremendously reduce gate leakage so 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 subsection, we investigate the manufacturability and process variability implications of our L_{Gate} biasing approach. As our method relies on biasing of drawn gate-length, it is important to correlate this with actual printed gate-length on the wafer. This is even more important as the bias we introduce in gate-length is of the same order as the typical critical dimension (CD) tolerances in 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 post-manufacturing setup, we follow a reticle enhancement technology (RET) and process simulation flow for an example cell master.

We use the layout of a generic AND2X6 cell and perform model-based optical proximity correction (OPC) on it

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

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

using *Calibre v9.3.2.5* [1].¹⁰ The printed image of the cell is then calculated using *dense* simulation in Calibre. The layout of the cell along with printed gate-lengths of all devices in it is shown in Figure 4. We measure the L_{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 Figure 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 *printed* dimensions of biased and unbiased versions of the cells shows that the benefits of biasing estimated using *drawn* dimensions will not be lost after RET application and the manufacturing process.

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

C. Process Variability

A number of sources of variation can cause fluctuations in gate-length, and hence in performance and leakage. This has been a subject of much discussion in the recent literature (e.g., [23], [8]). Up to $20\times$ variation in leakage has been reported in production 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_{th} vs. L_{Gate} curve. To validate this intuition, we study the impact of gate-length variation on leakage and performance both pre- and post-biasing using a simple worst-case approach. We assume the CD variation budget to be $\pm 10nm$. The performance and leakage of the test case circuits is measured at the worst-case, nominal and best-case process corners which consider just gate-length variation. This is done for the DVT-DGL approach in which biasing is done along with dual V_{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 benefits of alternative leakage control techniques. We note that the corner case analysis only models the inter-die component of 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 $\sigma_{WID} = \sigma_{DTD} = 3.33nm$. The variations are assumed to follow a Gaussian distribution with no correlations. We compare the results for three dual V_{th} scenarios: unbiased (DVT-SGL), biased (DVT-DGL) and uniformly biased (when gate-lengths of all transistors in the design are biased by 8nm). Leakage distributions for the test case *alu128* are shown in Figure 5. Note that in uniform biasing all devices are biased and the circuit delay no longer meets timing.

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

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

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 over CLLB. Since TLLB only biases devices of 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. CONCLUSIONS AND ONGOING WORK

We have presented a novel methodology that uses selective, *small* L_{Gate} biases to achieve an *easily manufacturable* approach to runtime leakage reduction. For our test cases we have observed the following.

- 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 design flow unlike sizing-based methods.
- With a biasing of $8nm$ in a $130nm$ process, leakage reductions of 24% to 38% are achieved for the most commonly used cells with a delay penalty of under 10%.
- 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. 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.
- 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 needs 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 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.
- The devices with biased gate-length 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.
- L_{Gate} biasing leads to more process-insensitive designs with respect to leakage current. Biased designs have up to 41% less leakage worst-case variability in presence of inter-die variations as compared to nominal gate-length designs. In presence of both inter- and intra-die CD variations, selective L_{Gate} biasing can yield designs less sensitive to variations.

Our ongoing work is along the following directions.

- 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.
- Assessment of leakage savings from the use of more than two gate-lengths for more frequently used and leaky cells in the library, such as inverters and buffers. Also, development of better approaches to reduce cell library size.
- Evaluation of the impact of biasing on leakage at future technology nodes for which leakage is a much bigger issue than it is at 130nm.

REFERENCES

- [1] “Mentor calibre,” <http://mentor.com/calibre/datasheets/opc/html/>.
- [2] “Opencores.org,” <http://www.opencores.org/projects/>.
- [3] “Prolith,” <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 Conference*, 2004, pp. 6–11.
- [5] F. Beeftink, P. Kudva, D. Kung and L. Stok, “Combinatorial Cell Design for CMOS Libraries,” *Integration, the VLSI Journal*, vol. 29, no. 4, pp. 67–93, 2000.
- [6] F. Boeuf et al., “A conventional 45nm CMOS Node Low-Cost Platform for General Purpose and Low Power applications,” *IEEE International Electron Devices Meeting*, pp. 425–428, 2004.
- [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 Conference*, 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*, 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 Conference*, 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 Conference*, 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 Journal of Solid-State Circuits*, vol. 28, no. 11, pp. 1131–1135, 1993.
- [12] I. Hyunsik, T. Inukai, H. Gomyo, T. Hiramoto and T. Sakurai, “VTCMOS Characteristics and its Optimum Conditions Predicted by a Compact Analytical Model,” in *International Symposium on Low Power Electronics and Design*, 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 Conference*, 1998, pp. 495–500.
- [14] M. Ketkar and S. Sapatnekar, “Standby Power Optimization via Transistor Sizing and Dual Threshold Voltage Assignment,” in *Proc. IEEE International Conference on Computer Aided Design*, 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 Conference*, 2003, pp. 192–194.
- [16] Z. Luo, “High Performance and Low Power Transistors Integrated in 65nm Bulk CMOS Technology,” *IEEE International Electron Devices Meeting*, pp. 661–664, 2004.
- [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 Journal of Solid-State Circuits*, vol. 30, no. 8, pp. 847–854, 1995.
- [18] S. Mutoh, S. Shigematsu, Y. Matsuya, H. Fukada, T. Kaneko and J. Yamada, “1V Multithreshold-Voltage CMOS Digital Signal Processor for Mobile Phone Application,” *IEEE Journal of Solid-State Circuits*, vol. 31, no. 11, pp. 1795–1802, 1996.
- [19] Y. Nakahara et al., “A Robust 65-nm Node CMOS Technology for Wide-Range Vdd Operation,” *IEEE International Electron Devices Meeting*, pp. 11.2.1–11.2.4, 2003.
- [20] S. Nakai et al., “A 65 nm CMOS Technology with a High-Performance and Low-Leakage Transistor, a 0.55 μm^2 /sup 2/ 6T-SRAM

- Cell and Robust Hybrid-ULK/Cu Interconnects for Mobile Multimedia Applications,” *IEEE International Electron Devices Meeting*, pp. 11.3.1–11.3.4, 2003.
- [21] S. Narendra, D. Blaauw, A. Devgan and F. Najm, “Leakage Issues in IC Design: Trends, Estimation and Avoidance,” in *Proc. IEEE International Conference on Computer Aided Design*, 2003, tutorial.
- [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 Journal of Solid-State Circuits*, vol. 37, no. 3, pp. 413–419, 2002.
- [23] R. Rao, A. Srivastava, D. Blaauw and D. Sylvester, “Statistical analysis of subthreshold leakage current for VLSI circuits,” *IEEE Transactions on Very Large Scale Integrated Systems*, vol. 12, no. 2, pp. 131–139, 2004.
- [24] P. Royannez et al., “90 nm Low Leakage SoC Design Techniques for Wireless Applications,” in *IEEE International Solid-State Circuits Conference*, 2005, pp. 138–.
- [25] D. K. Schroder and J. A. Babcock, “Negative Bias Temperature Instability: Road to Cross in Deep Submicron Silicon Semiconductor Manufacturing,” vol. 94, no. 1, pp. 1–18, 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 Journal of Solid-State Circuits*, vol. 32, no. 6, pp. 861–869, 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 Transactions on Very Large Scale Integrated Systems*, vol. 10, pp. 79–90, April 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 Conference*, 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 *International Conference on Computer Design*, 2000, pp. 227–232.
- [30] L. Wei, Z. C. andn 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 Conference*, 1998, pp. 489–494.
- [31] L. Wei, Z. Chen and K. Roy, “Mixed- V_{th} CMOS Circuit Design Methodology for Low Power Applications,” in *Proc. ACM/IEEE Design Automation Conference*, 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 Conference*, 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. Symposium on VLSI Circuits*, 1998, pp. 40–41.

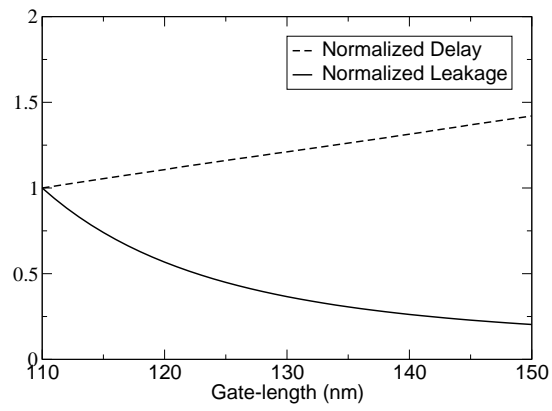


Fig. 1. Variation of leakage and delay (each normalized to 1.00) for an NMOS device in an industrial 130nm technology.

```

procedure  $L_{Gate}Biassing$ 
1  Run SSTA to initialize  $s_p \forall$  cell instances,  $p$ 
2   $S \leftarrow \{\}$ 
3  forall cell instances,  $p$ 
4       $P_p \leftarrow ComputeSensitivity(p)$ 
5       $S \leftarrow S \cup P_p$ 
6  do
7       $P_{p^*} \leftarrow \max(S)$ 
8      if ( $P_{p^*} \leq 0$ )
9          exit
10      $S \leftarrow S - \{P_{p^*}\}$ 
11      $SaveState()$ 
12     Downsize cell instance  $p^*$ 
13      $EISTA(p^*)$ 
14     if ( $TimingViolated()$ )
15          $RestoreState()$ 
16     else
17          $N \leftarrow p^* \cup$  fan-in and fan-out nodes of  $p^*$ 
18         forall  $q \in N$ 
19             if ( $P_q \in S$ )
20                  $P_q \leftarrow ComputeSensitivity(q)$ 
21                 Update  $P_q$  in  $S$ 
22 while ( $|S| > 0$ )

procedure  $ComputeSensitivity(q)$ 
1   $old\_slack \leftarrow$  Slack on cell instance  $q$ 
2   $old\_leakage \leftarrow$  Leakage of cell instance  $q$ 
3   $SaveState()$ 
4  Downsize cell instance  $q$ 
5   $CISTA(q)$ 
6   $new\_slack \leftarrow$  Slack on cell instance  $q$ 
7   $new\_leakage \leftarrow$  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.

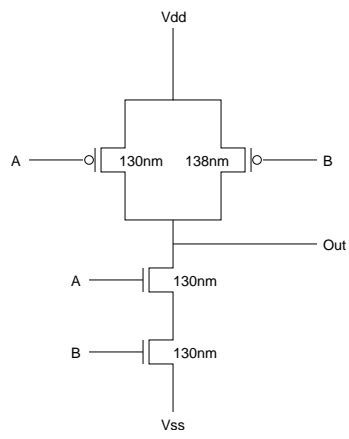


Fig. 3. Gate-length biasing of the transistors in NAND2X1 when only the rise and fall timing arcs from input A to the output are critical.

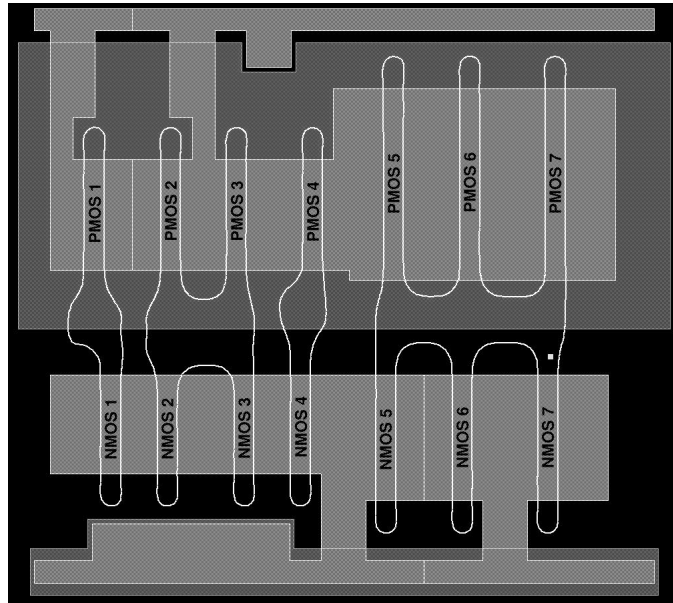


Fig. 4. Cell layout of a generic AND2X6 with simulated printed gate-lengths.

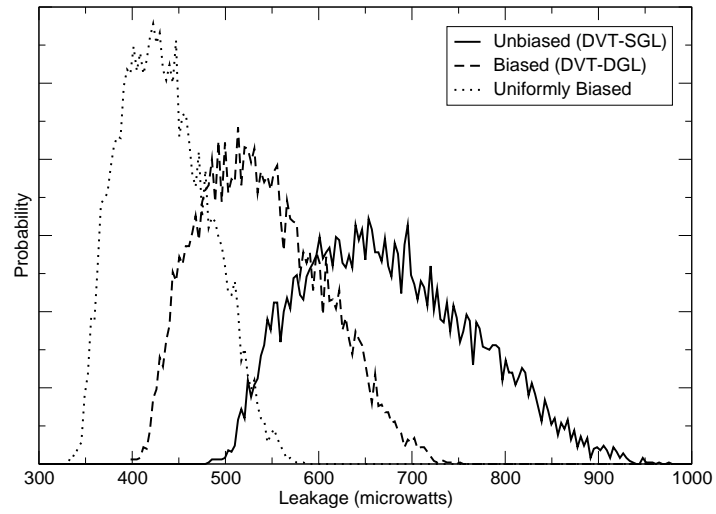


Fig. 5. Leakage distributions for unbiased, uniform-biased and technology-level selectively-biased alu128. Note the “left-shift” of the distribution with the introduction of biased devices in the design.

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

Circuit	Leakage (<i>mW</i>)		CPU (<i>s</i>)	
	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

TABLE III
TEST CASES USED IN OUR EXPERIMENTS AND THEIR DETAILS.

Test Case	Source	#Cells	Delay (<i>ns</i>)	Leakage (<i>mW</i>)	Dynamic (<i>mW</i>)
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 NON-AVAILABILITY OF V_{th} AND L_{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 (ns)	DVT-SGL			DVT-DGL			Reduction			CPU (s)
		Leakage (mW)	Dynamic (mW)	Total (mW)	Leakage (mW)	Dynamic (mW)	Total (mW)	Leakage (%)	Dynamic (%)	Total (%)	
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 SUBTHRESHOLD LEAKAGE AND GATE TUNNELING LEAKAGE OF $90nm$ PMOS AND NMOS DEVICES OF $1\mu m$ WIDTH AT DIFFERENT TEMPERATURES. TOTAL LEAKAGE REDUCTIONS ARE HIGH EVEN WHEN GATE LEAKAGE IS CONSIDERED.

Device	Temp ($^{\circ}C$)	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%

TABLE VIII

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

Device Number	Gate Length (<i>nm</i>)					
	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 $13nm$. ELAT=EXPOSURE LATITUDE.

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

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

Circuit	Circuit Delay (<i>ns</i>)						% Spread Reduction
	Unbiased (DVT-SGL)			Biased (DVT-DGL)			
	BC	WC	NOM	BC	WC	NOM	
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 (<i>mW</i>)						% Spread Reduction
	Unbiased (DVT-SGL)			Biased (DVT-DGL)			
	BC	WC	NOM	BC	WC	NOM	
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

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

Circuit	Delay (<i>ns</i>)	Leakage			CPU (<i>s</i>)	
		CLLB (<i>mW</i>)	TLLB (<i>mW</i>)	Reduction (%)	CLLB (<i>s</i>)	TLLB (<i>s</i>)
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

LIST OF FIGURE AND TABLE CAPTIONS

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

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

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.

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

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.

Table I. Comparison of leakage and runtime when EISTA and CISTA are used for sensitivity computation.

Table II. Asymmetry in delays of various timing arcs within a NAND2X2 cell.

Table III. Test cases used in our experiments and their details.

Table IV. Leakage reduction and delay penalty due to gate-length biasing for all 25 cells in our library.

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 non-availability of V_{th} and L_{Gate} knobs. SVT-DGL is slower than SVT-SGL for delay penalties of 2.5% and 5%.)

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.

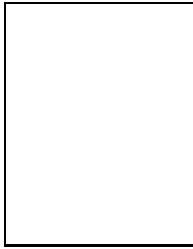
Table VII. Impact of gate-length biasing on subthreshold leakage and gate tunneling leakage of 90nm PMOS and NMOS devices of $1\mu m$ width at different temperatures. Total leakage reductions are high even when gate leakage is considered.

Table VIII. Comparison of printed dimensions of unbiased and biased versions of AND2X6. The unbiased nominal gate-length is 130nm while the biased nominal is 138nm. Note the high correlation between unbiased and biased versions.

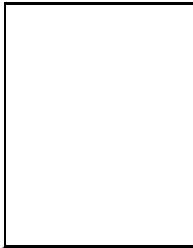
Table IX. Process window improvement with gate-length biasing. The CD tolerance is kept at 13nm. ELAT=Exposure latitude.

Table X. Reduction in performance and leakage power uncertainty with biased gate length in presence of inter-die variations. The uncertainty spread is specified as a percentage of nominal. The results are given for dual V_{th} and the biasing is 8nm.

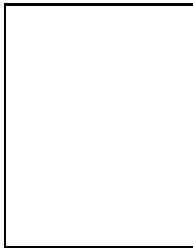
Table XI. Leakage power from transistor-level gate-length biasing.



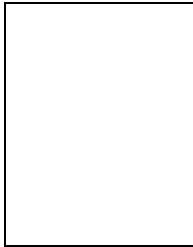
Puneet Gupta(S'01) is a co-founder of Blaze DFM Inc. He completed his Bachelors in Electrical Engineering from Indian Institute of Technology, Delhi. He is a Ph.D. candidate at UC San Diego and also the recipient of the IBM Ph.D. fellowship. He has published over 30 papers and has 7 patents. He has previously co-presented an embedded tutorial on Manufacturing Aware Physical Design at ICCAD 2003. His research interests lie primarily in the design-manufacturing interface.



Andrew B. Kahng is professor of CSE and ECE at UC San Diego. He has published over 200 papers in the VLSI CAD literature, receiving three Best Paper awards and an NSF Young Investigator award. His A.B. in applied mathematics is from Harvard College and his M.S. and Ph.D. degrees in computer science are from UC San Diego. From 1989 to 2000, he was a member of the UCLA computer science faculty. His research is mainly in physical design and performance analysis of 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, Professor Kahng has defined the physical design roadmap for the International Technology Roadmap for Semiconductors (ITRS), and since 2001 has chaired the U.S. and international working groups for Design technology for the ITRS. He has been active in the MARCO Gigascale Silicon Research Center since its inception. He was also the founding General Chair of the ACM/IEEE International Symposium on Physical Design, and co-founded the ACM Workshop on System-Level Interconnect Planning.



Puneet Sharma(S'02) is a Ph.D. student in ECE Department at UC San Diego since September 2002. He received his Bachelor of Technology in Computer Science and Engineering from Indian Institute of Technology, Delhi in 2002. His research interests include leakage power reduction and design for manufacturing.



Dennis Sylvester(S '95, M '00, SM '04) received the B.S. degree in electrical engineering summa cum laude from the University of Michigan, Ann Arbor, in 1995. He received the M.S. and Ph.D. degrees in electrical engineering from University of California, Berkeley, in 1997 and 1999, respectively. He worked at Hewlett-Packard Laboratories in Palo Alto, CA, from 1996 to 1998. His dissertation research was recognized with the 2000 David J. Sakrison Memorial Prize as the most outstanding research in the UC-Berkeley EECS department. 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, Ann Arbor. He has published 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. Dr. Sylvester received an NSF CAREER award, the 2000 Beatrice Winner Award at ISSCC, a 2004 IBM Faculty Award, and several best paper awards and nominations. He is the recipient of the 2003 ACM SIGDA Outstanding New Faculty Award, the 1938E Award from the College of Engineering Award for teaching and mentoring, and the Henry Russel Award, which is the highest award given to faculty at the University of Michigan. 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). He is an Associate Editor for IEEE Transactions on VLSI Systems. 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 is a member of ACM, American Society of Engineering Education, and Eta Kappa Nu.