

# A Temperature-aware Synthesis Approach for Simultaneous Delay and Leakage Optimization

Nathaniel A. Conos and Miodrag Potkonjak

Computer Science Department  
University of California, Los Angeles  
{conos, miodrag}@cs.ucla.edu

**Abstract**—Accurate thermal knowledge is essential for achieving ultra low power in deep sub-micron CMOS technology, as it affects gate speed linearly and leakage exponentially. We propose a temperature-aware synthesis technique that efficiently utilizes input vector control (IVC), dual-threshold voltage gate sizing (GS) and pin reordering (PR) for performing simultaneous delay and leakage power optimization. To the best of our knowledge, we are the first to consider these techniques in a synergistic fashion with thermal knowledge. We evaluate our approach by showing improvements over each method when considered in isolation and in conjunction. We also study the impact of employing considered techniques with/without accurate thermal knowledge. We ran simulations on synthesized ISCAS-85 and ITC-99 circuits on a 45 nm cell library while conforming to an industrial design flow. Leakage power improvements of up to 4.54X (2.14X avg.) were achieved when applying thermal knowledge over equivalent methods that do not.

## I. INTRODUCTION

Power minimization continues to be one of the top design metrics in modern VLSI design [1][13][30]. For modern CMOS transistors, power has been primarily characterized into three main sources: 1) switching, due to the charging/discharging of load capacitance; 2) short circuit, due to the momentary short circuit state between the pull-up/down of devices; and 3) leakage, which is further broken down into gate tunneling and sub-threshold leakage currents. Sub-threshold leakage has been shown to be the dominant leakage portion for modern CMOS devices; it is strong function of input vector state and is exponentially affected by operating temperature. Gate delay is also thermally affected as rising temperatures contribute to decreased carrier mobility affecting propagation delays. However, current tools lack early and thermal analysis to better address modern and pending design issues affecting power dissipation, circuit performance, and reliability.

We propose a temperature-aware synthesis methodology combining and improving gate-level pre-silicon synthesis techniques that utilize thermal knowledge during the optimization. A summary of the considered techniques is listed below.

- Dual- $V_t$  gate sizing (GS) - utilize thermal knowledge to efficiently size and assign  $V_t$  to thermally impacting gates to minimize leakage power.
- Input vector control (IVC) - find promising input vectors for a given thermal map by placing temperature critical gates to their minimal leakage states.

- Pin reordering (PR) - improve the effectiveness of IVC by placing each gate to its optimal leakage state, relaxing IVC-imposed constraints.

The main contribution of our work is to demonstrate the vital role temperature knowledge has on modern CAD optimization techniques using industrial imposed constraints. However, until now, previous approaches have considered at most two of the mentioned techniques simultaneously with temperature (e.g., GS+PR, IVC+GS+PR). Furthermore, these techniques often assume simplistic delay/power models that operate under nominal conditions (e.g., operating temperature, average leakage). Our goal is to show that a strong interdependence exists when considering all of the enabled techniques in light of utilizing the correct temperature knowledge during the optimization process. We demonstrate that the success of considered techniques heavily depend on each other due to interacting metrics we consider such as leakage and delay.

Additional contributions include further enhancements to input vector control and gate replacement by simultaneously employing pin reordering using thermal profile knowledge, while adhering to industrial imposed design constraints, such as load capacitance and slew limits [1]. The complete flow can be performed in an iterative fashion to obtain better solutions and can be easily integrated into modern CAD flow. We also show that input vector control optimization should be considered across diverse temperature profile scenarios requiring different input vectors per temperature assumption. Section IV provides a more complete description of our contributions.

## II. MOTIVATION

The addition of temperature drastically changes the optimization search space. Figure 1 (a) illustrates how temperature impacts IVC decisions under a hot circuit condition. Under the nominal temperature condition (left), the optimal configuration places the top two nand gates to their *mls*, while trading off the *wls* for the remaining output gate. However, making a decision to the right figure would result in a significant leakage penalty of over 6.8X (Table I). Thus, under a typical system that exhibits temperature variations, it is key to provide the correct IVC while simultaneously account for the delay alterations. The same idea can be readily applied early in the synthesis phase during gate sizing and  $V_t$  assignment. Under the scenario where standby leakage is dominant, the correct input vector plays a vital role in selecting optimal sizes and threshold voltages.

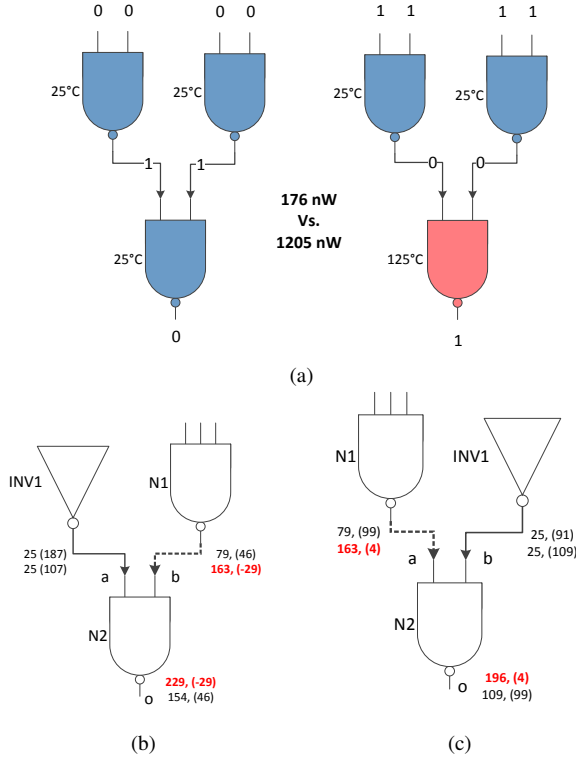


Fig. 1: Input vector control under two temperature settings (a); pre-slack (b) and post-slack (c) re-allocation via input pin reordering. For (b-c), the critical path is represented as bold-red with arrival time (left) and slack (right) terms provided.

Pin reordering can be applied to reallocate slack ( $T_{target} - T_{max}$ ) to achieve delay targets. Figure 1 (b-c) illustrates an example of maximizing slack by reordering the input pins. Four terms are shown for each *net*. The top two represent rise arrival/slack), the bottom corresponds to fall arrival/slack (ps). For the sake of simplicity, ignore the effects of slew on input pins *a* and *b* for gate *N2*, and assume all gates are operating under nominal temperature, negative unate, and that a  $T_{target}=200$  ps is set. Gate *N2* has cell rise and fall delays of (25, 33) from  $a \rightarrow o$  and (66, 84) from  $b \rightarrow o$ . Although, path  $a \rightarrow o$  is shorter than  $b \rightarrow o$ , the rise/fall arrival times and slack of input *b* are (79, 46) and (163, -46), which leads to a timing violation ( $T_{max}=229$  ps) (Figure 1 (c)). To maximize slack, the input pins of *N2* can be reordered such that the path with minimum slack is connected to the path with the smallest arrival time. Thus, after slack reallocation,  $T_{max}=194$  ps without violations. However, due to the temperature dependence on gate delay, the delay of certain paths may be totally different. Thus, it is vital that accurate temperature knowledge is applied during the synthesis flow.

### III. RELATED WORK

Temperature has increased in importance in modeling in several CAD areas, ranging from architectural [19] and behavioral synthesis [24], to the transistor/logic level pertaining to hardware security [26][27][28][29]. Leakage power has become increasingly important for modern CMOS devices. Input vector control (IVC) has been applied to minimize

TABLE I: Input vector state (*m*) dependent leakage power (*nW*) for high ( $HV_t$ ) and low ( $LV_t$ ) of 2-input nand gate at minimum size operating at room and hot temperature.

<i>m</i>	NAND2			
	25 °C		125 °C	
	$HV_t$	$LV_t$	$HV_t$	$LV_t$
00	<u>3.57</u>	<u>6.22</u>	<u>58.2</u>	<u>176</u>
01	24.8	44.8	285	867
10	4.09	7.32	66.3	201
11	37.2	67.5	397	1205

leakage by applying minimum leakage input vectors to leakage critical gates [12]. Leakage power is reduced due to the strong dependence of sub-threshold currents (transistor stacking) with respect to a gates applied. MUXes for internal node control was explored in [13] to drive combinational circuit sections to their minimally achievable leakage states during idle periods. Finding the minimum input vector, however, is NP-Hard [14] and requires heuristics to solve large circuit sizes. Furthermore, identifying suitable IVC is further impacted by process variation [23]. Gate replacement techniques were also covered to address internal node controllability issues. Gate replacement replaces gates in their worst-case leakage (*wls*) state with an equivalent lower leakage gate [14]

Gate sizing have become effective techniques for addressing energy and performance metrics [2][3][4][7]. The gate width is adjusted to achieve various drive strengths, enabling circuit power and timing trade-offs. In the discrete domain, gate sizing is an NP-Hard problem [5] and several well known solutions have been proposed, such as Lagrangian relaxation [20], dynamic programming [2], combinatorial relaxation [9], and sensitivity-based optimizations [3][4][7][21]. Dual threshold voltage ( $V_t$ ) combined with gate sizing has also been proposed [8]. Low  $V_t$  gates achieve greater speed, but at the expense of higher leakage and vice-versa for high  $V_t$  gates. Low  $V_t$  cells are placed on critical paths to achieve performance and high  $V_t$  gates on non critical paths to minimize leakage power. Temperature-aware dual  $V_t$  and gate sizing was explored [15] and uses heuristics to place high  $V_t$  in hot regions; however, they consider simplified leakage and delay models for accounting for power and timing values. Gate sizing has been explored in Near-threshold computing [4][25].

More recently, Intel researchers have held a yearly design contest at ISPD on the topic gate sizing and  $V_t$  assignment [1]. Leakage minimized under hard timing constraints. Industrial optimization design constraints such as gate load, and slew dependencies were used. However, only average leakage values were used for computing leakage power. In our work, we consider the gate input vector state for leakage power computation with respect to its operating temperature, while also conforming to identical industrial design constraints.

### IV. TECHNICAL APPROACH

Our leakage minimization framework consists of three major steps, as shown in Figure 2, which include: 1) initialization of cell library and circuit thermal maps; 2) leakage minimization through finding minimal input vectors combined with pin ordering (IVC+PR); 3) leakage minimization through

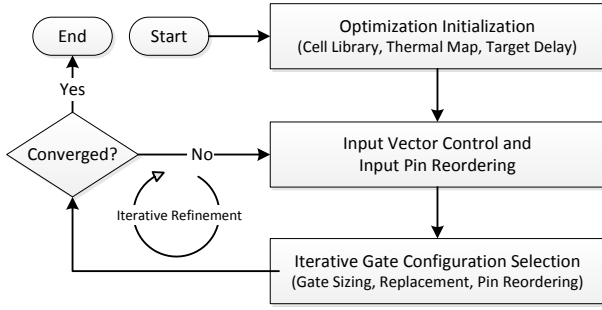


Fig. 2: Synthesis flow

simultaneous dual- $V_t$  gate sizing (GS), and input pin reordering (PR). Steps 2-3 can be repeated to obtain additional savings.

#### A. Thermal Map

Our work addresses circuit optimization during the pre-silicon phase and relies on circuit thermal simulations to generate thermal maps. Thermal maps can be generated using models found in *HotSpot* [19] where the functional-unit level temperature modeling can be extrapolated to support gate-level temperature modeling. Actual gate-level activity switching factors (obtained either through gate-level or probabilistic simulations), and its cell physical placement information can be used to generate power densities. The resultant power densities can then be used to generate a circuit-wide thermal model. However, such a process would require actual correlations to actual hardware measurements in order to generate a reliable temperature profile for use [19]. Due to this limitation in our optimization search space, we assume a static chip-wide temperature profile for each netlist as a starting point. As temperature modeling in CAD tools mature, further improvements in designs may be possible, since input vector control, pin-reordering and gate-sizing all impact power dissipation of the circuit. For our experiments, we generate temperature scenarios under two circuit-wide operating temperature assumptions (55°C as *cool* and 125°C as *hot*).

#### B. Input Vector Control and Pin Reordering

IVC is an essential technique for leakage reduction in idle modes since temperature critical (e.g., hot) gates may be placed in lower leakage states and traded-off with less critical (e.g., cool) gates be placed in higher leakage states. We improve the conventional IVC by combining it with input pin reordering (PR). PR provides additional opportunity for leakage savings for each gate, since it relaxes the constraint imposed by the IVC setting of its transitive fan-in gates. Thus, a higher percentage of gates may be placed at or closer to their respective *mls*, making IVC+PR an effective technique for addressing circuits that exhibit large temperature variations.

Finding the optimal IVC is NP-Hard [10]. In our work, we utilize a statistical random-walk procedure of 10K randomly generated input vectors for obtaining a promising IVC to be used in later phases. This procedure is simple in nature and note that more sophisticated techniques can be used. However, our experience has shown this technique achieves relatively

---

#### Algorithm 1 Gate Configuration Selection Algorithm

---

- 1: Initialize library, delay target  $D_{target}$ , and *netlist*
  - 2: Obtain minimum input vector and pin reordering (*mIVC*)
  - 3: Compute difficulty metric for each gate (power, delay)
  - 4: Initially set all gates to High- $V_t$  and minimal size
  - 5: Compute difficulty and lock top  $K$  critical gates
  - 6: Lock  $K$  maximally constrained gates  $G_{lock}$
  - 7: **repeat**
  - 8:   For each gate, find least constraining move  $m$
  - 9:   Perform move  $m$  and lock chosen gate
  - 10:   If all gates are locked, unlock all gates  $\in G_{lock}$
  - 11:   If no solutions found after  $L$  iterations then relax  $K$
  - 12:   Recompute circuit difficulties
  - 13: **until** *Converge*
- 

fast convergence with most designs converging before completion. To further reduce the number of computations in this phase, we modify the look-up-table entry for each gate to only consider the minimum leakage values for respective input vector permutations. For example, the minimum IVC for a 3-input nandgate “100” can represent its respective leakage profiles of “100” and “010” (Table I).

#### C. Gate Configuration Selection

The next step in our flow is to perform iterative gate-level modifications for minimizing leakage power with respect to a given delay target. This phase combines dual- $V_t$  gate sizing (GS), and pin reordering (PR) techniques.

1) *Selection Heuristic*: Our gate configuration selection approach is based on *maximally* constrained, *minimally* constraining optimization paradigm. The procedure is constructive at each its step such that the most benefiting move in terms optimization criteria is performed. The maximally constrained principle attempts to assign the gate configurations to difficult gates early while there is still slack in the design. In addition, the early assignment of the difficult or constraining gates early provide an accurate picture of actual consequent difficulties for future moves and is recognized as early as possible [22]. In a circuit where leakage power is localized to few hot spots, determining the optimal configuration of these gates early in the optimization phase is critical. The *minimally* constraining principle states that at each step we should determine a gates move (GS and PR) in such a way that the remaining gates are as least constrained as possible.

We first define the sources of difficulty or constraining metric in determining the best configuration for a particular gate. Gates are sorted in descending order during “difficulty computation” step (Algorithm 1) in decreasing precedence: 1) Leakage power - temperature leakage impact factor; 2) Slack - gate participation on the critical paths; 3) Logic Depth - gate participation on longest paths; 4) Fan-out - affect on its transitive fan-out gates; and 5) Fan-in - affect on its transitive fan-in gates. The leakage profile of a gate is considered as the main source of difficulty, followed by its ability to potentially impact the circuit delay. Our objective is to minimize leakage consumption, thus, we first identify the top  $K$  leakage critical gates and lock them to their minimally impacting configuration.

Algorithm 1 highlights our gate configuration selection procedure. Line 1 and 2 performs all required pre-processing steps including thermal simulation to identify critically temperature impacting gates and IVC to obtain the minimum leakage input vector combined with its corresponding optimal pin-ordering structure. The key idea in this step is to maximize achievable savings by *IVC + PR* before performing gate-level adjustments. Line 3 identifies the most constrained or leakage critical gates and locks them to their initial minimal leakage configuration. For our purposes, we set  $K$  to be equal to the number of gates predicted to be temperature critical. Lines 7-13 performs iterative gate-level adjustments based on its move classification. This procedure is repeated until all gates configuration have been set. Once a configuration is determined, the gate is locked (frozen) (excluding gates in initially the locked gate set  $G_{lock}$ ) until the start of the next iteration. The locking principle prevents the algorithm from getting stuck into a local minima by requiring all gates configurations to be determined before reiterating.

2) *Move Classification*: Gate moves are classified into three groups with respect to our delay-constrained objective. For each gate potential gate move, three ideal scenarios are considered:

- i Leakage power and delay reduction
- ii Leakage power reduction, constant delay
- iii Leakage power reduction and delay increase

It is important to account for valid moves (no load or slew violation). These valid moves are assigned priorities in the precedence class order of *i*, *ii*, and *iii*. Moves that benefit both leakage and delay (class *i*) are always selected over moves belonging in classes *ii* and *iii*, and are compared against other moves within its own class as the product of leakage and delay savings. If no class *i* moves exists, then class *ii* moves are selected by the maximum leakage energy improvement. If only class *iii* moves are found, the move that produced the maximum  $\frac{\text{benefit}}{\text{cost}}$  is selected. Note that the above objective concepts may be applied inversely when the objective is set (e.g., power-constrained delay minimization).

A major challenge during our gate-level configuration selection approach is that the initial step requires the locking of  $K$  top critical gates (line 5). Note that a condition may exist where the target delay was not achievable due to locking constraints placed on sizable gates. Under these cases, where after a specified number iterations have passed where no valid solution has been found, a gate is chosen in the locked set ( $G_{lock}$ ) to be unlocked using the maximally constrained and minimally constraining principle. The most constrained locked gate is defined using: 1) maximum frequency being on the critical path, and 2) leakage power.

3) *Epsilon Critical Tree Extraction*: We employ an epsilon critical tree ( $\epsilon$ -tree) structure to enable our algorithm to scale linearly with respect to circuit size. Determining gate configurations while maintaining accurate delay pictures, is the major challenge in sensitivity-based algorithms. A single move may require a delay re-computation of the entire circuit. Performing per-gate-wise delay update, thus, results in quadratic run time. We develop an efficient  $\epsilon$ -tree structure that performs delay updates when it is detected that gates along the critical path (or

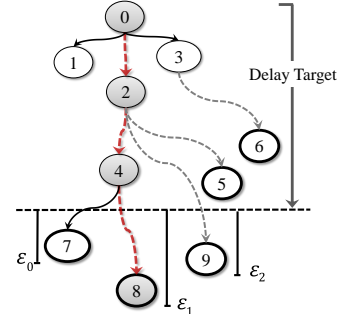


Fig. 3: An example of  $\epsilon$  critical path ( $\epsilon_{path}$ ); the critical path in red; transitive fan-out output nodes in bold outlines; and  $\epsilon_i$  corresponds to the absolute delay difference with respect to the target delay used for estimating delay cost of a move.

within some tolerance) are updated. The cost for acquiring accurate delay values of the entire circuit is significantly reduced, while minimally impacting accuracy. To further improve run-times at the expense of accuracy, groups of gates may be sized at a time as done in [3][7].

An  $\epsilon_{tree}$  consists of gates that are within  $\epsilon - \text{delay}$  of the critical path in the last accurate delay computation (shaded nodes in Figure 3). The bold-outlined nodes are primary outputs ( $P_{out}$ ), which are transitively connected to a node in the critical path. The delay impact of a gate is accounted by its transitive relation to the  $\epsilon_{path}$ . For example, a gate that is either on the critical path or an output gate of a critical path gate would cause a  $\delta$ -delay (slack) to its transitive primary output nodes. The  $\delta$ -delay is used to estimate the delay impact of each move and is defined as the sum of the squared difference with respect to each transitively affected primary output time,  $\epsilon_i$ , with respect to the delay target (Figure 3). Using an  $\epsilon_{path}$  enables the following very efficient delay estimation:

$$D_{cost} = \sum_i^{|P_{out}|} (\epsilon_i - D_{target})^2 \quad (1)$$

The run time can be reduced since the percentage of gates that make up the critical path is relatively small compared to the total gate count ( $\leq 5\%$ ). However, there can be an exponential number of paths that need to be taken into account. In order to maintain accuracy, we update the propagation and slew delays of the current gate under inspection, as well as its immediate fan-in/out neighbors. Any circuit violations (load and slew) are also fixed during accurate delay computations by employing a similar technique as in [21].

## V. SIMULATION FRAMEWORK

We evaluate our synthesis technique on 10 industrial benchmarks included in ISCAS-85 and ITC-99, synthesized using Cadence Encounter in order to retrieve net/wire capacitance. The Nandgate 45 nm cell library [6] is used and 3 gate sizes are assumed (1X, 2X, and 4X). We extend the cell library to support dual- $V_t$  optimization by using EKV formulas in [18] to fit against the base library and set ( $LV_t=0.55V$ ,  $HV_t=0.6V$ ,  $V_{dd}=1.0V$ ). We implement an in-house power and delay timer in C++ and was correlated within 1E-3 within the Synopsys PrimeTime industrial tool. We extend our look-up table model

to support continuous temperature indexing so that both leakage power and delay can be referenced using the driving load, size/type, and rise/fall input slews, and temperature. Leakage power is indexed using its IVC. The minimum leakage IVC is obtained through simulation (Section IV). Two chip-wide thermal operating settings are used: 1) 55°C as *cool* and 2) 125°C as *hot*. We limit the three gate configuration iterative refinement phases as additional phases resulted with marginal improvements at the expense of additional simulation run-time.

## VI. EXPERIMENTAL RESULTS

We evaluate the effect of utilizing temperature knowledge during the optimization process when considering input vector control (IVC), dual- $V_t$  gate sizing (GS), and input pin reordering (PR). We report the leakage improvements across two enabled optimization modes corresponding to their enabled techniques: 1) O1 (IVC+GS) and 2) O2 (IVC+GS+PR). The optimization objective is to minimize leakage consumption (stand-by mode), while meeting delay targets.

Table II shows the impact of using temperature knowledge across considered leakage optimization techniques. Results are grouped (row-wise) with respect to the circuit, and further sub-grouped (row-wise) with respect to the correct and wrong temperature assumption (columns 4 and 5), respectively. The correct temperature knowledge is used when the *actual* temperature under “Act.” matches the *predicted* temperature “Pred.” For example, consider benchmark *c2670* where a *hot* temperature scenario is predicted. The leakage power achieved when making the correct “Hot” temperature knowledge is 141 uW in contrast to using the wrong temperature knowledge that result with a leakage power of 522 uW. Subsequent leakage improvements are provided for the remaining techniques enabled in O2. For benchmark *c2670*, correct temperature knowledge achieved leakage improvements (leakage reduction factor) 3.70X (O1) and 3.99X (O2), showing additional leakage savings as more techniques were enabled.

The impact of temperature knowledge in placing gates in their minimal leakage state can be clearly observed by determining the % of gates (post optimization) in their minimum leakage state (*mls*) and worst leakage state (*wls*), listed under the “% Gates.” columns. It is important to note that *wls* and *mls* are only two of the  $2^{fi}$  leakage states considered, where *fi* corresponds to gates fan-in size.

Using accurate temperature knowledge enables superior solutions over an equivalent methods using the incorrect temperature assumption (Table II). For example, the result for “c2670” shows that the correct temperature knowledge enabled 37% gates to be placed in *mls* in contrast to 24% when the wrong temperature knowledge is used. Additionally, using the correct temperature knowledge placed a lower percentages of gates in their *wls* (17% vs 25%). As the number of available gate configurations increases (from O1 to O2), more gates were able to be placed in their *mls*, and less in their *wls*. Improvements using the correct temperature knowledge are greater since they enable more gate selection candidates to be selected among temperature-leakage critical gates. For instance, circuit the optimization of “c2670” under the correct temperature prediction (59% *mls*, 14% *wls*) outperformed the wrong temperature prediction (41% *mls*, 15% *wls*).

## VII. CONCLUSION

We have developed a synergistic temperature-aware delay and leakage optimization approach using enhanced synthesis techniques including: input vector control (IVC), pin reordering (PR) combined with dual-threshold voltage  $V_t$  gate sizing (GS). We study the impact of temperature knowledge using these techniques under *cool* and *hot* operating conditions and report up to 4.54X leakage improvements (2.14X avg.) when utilizing the correct temperature knowledge. We evaluate our approach on a comprehensive set of benchmarks included in ISCAS-85, and ITC-99 on a 45 nm dual- $V_{th}$  cell library while conforming to industrial imposed constraints.

## REFERENCES

- [1] M. M. Ozdal et al., “The ISPD -2012 Discrete Cell Sizing Contest and Benchmark Suite”, *ISPD*, 161-164, 2012.
- [2] M. M. Ozdal et al., “Gate sizing and device technology selection algorithms for high-performance industrial designs,” *ICCAD*, 724-731, 2011.
- [3] N. A. Conos et al., “Gate sizing in the Presence of Switching Activity and Input Vector Control,” *VLSI-SOC*, 2013.
- [4] N. A. Conos et al., “Maximizing Yield in Near-Threshold Computing under the Presence of Process Variation,” *PATMOS*, 2013.
- [5] W. N. Li, “Strongly NP-hard discrete gate-sizing problems,” *ICCD*, 1045-1051, 1993.
- [6] Nangate FreePDK45-nm Library, <http://www.si2.org/>, 2011.
- [7] O. Coudert, “Gate sizing for constrained delay/power/area optimization,” *VLSI*, 465-472, 1997.
- [8] A. Srivastava et al., “Power minimization using simultaneous gate sizing, dual-Vdd and dual-Vth assignment,” *DAC*, 783-787, 2004.
- [9] Y. Liu, “A New Algorithm for Simultaneous Gate Sizing and Threshold Voltage Assignment,” *TCAD*, 223-234, 2010.
- [10] K. Roy et al., “Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits,” *IEEE*, 305-327, 2003.
- [11] A. Calimera et al., “Temperature-insensitive dual-Vth synthesis for nanometer CMOS technologies under inverse temperature dependence,” *VLSI*, 1608-1620, 2009.
- [12] J. P. Halter et al., “A gate-level leakage power reduction method for ultra-low-power CMOS circuits,” *CICC*, 475-478, 1997.
- [13] A. Abdollahi et al., “Leakage current reduction in CMOS VLSI circuits by input vector control,” *VLSI*, 140-154, 2004.
- [14] L. Yuan et al., “A combined gate replacement and input vector control approach for leakage current reduction,” *VLSI*, 173-182, 2006.
- [15] J. Gu et al., “Enhancing dual-Vt design with consideration of on-chip temperature variation,” *ICCD*, 542-547, 2010.
- [16] L. Yuan et al., “Simultaneous input vector selection and dual threshold voltage assignment for static leakage minimization,” *ICCAD*, 4-8, 2007.
- [17] A. K. Sultania et al., “Transistor and pin reordering for gate oxide leakage reduction in dual  $t_{ox}$  circuits,” *ICCD*, 228-233, 2004.
- [18] D. Markovic et al., “Ultralow-power design in near-threshold region,” *IEEE*, 237-252, 2010.
- [19] W. Huang et al., “Accurate, pre-RTL temperature-aware design using a parameterized, geometric thermal model,” *IEEE*, 1277-1288, 2008.

TABLE II: Leakage power and improvement factors (row-wise) shows results when optimizing under actual (Act.) and predicted (Pred.) temperatures with respect to the addition of each enabled technique (column-wise): (O1) IVC+GS; and (O2) IVC+GS+PR. Also shown are the % of total gates in their minimum leakage state (*mls*) and worse leakage state (*wls*) with respect to O1 and O2 as well as the achieved clock delay.

Circ.	#Gates	#Inputs	Act. (°C)	Pred. (°C)	Lk. Pwr. (uW)		Lk. Imprv. (X)		% of gates. ( <i>mls</i> , <i>wls</i> )		Clk (ns)
					O1	O2	O1	O2	O1	O2	
c1355	628	41	Nm.	Nm. Hot	43.4 47.8	28.3 36.2	1.10 -	1.28 -	- -	- -	17.0
			Hot	Nm. Hot	69.6 48.1	45.0 26.4	1.45 -	1.70 -	17, 17 11, 8	17, 17 11, 8	19.4
c2670	1575	233	Nm.	Nm. Hot	95.1 117	88.8 102	1.23 -	1.15 -	- -	- -	15.7
			Hot	Nm. Hot	522 141	499 125	3.70 -	3.99 -	24, 25 37, 17	26, 21 41, 11	17.1
c3540	1750	50	Nm.	Nm. Hot	135 162	124 146	1.20 -	1.18 -	- -	- -	28.2
			Hot	Nm. Hot	556 191	532 172	2.91 -	3.10 -	40, 31 41, 23	47, 27 50, 17	28.3
c6288	2490	32	Nm.	Nm. Hot	25.9 70.1	25.6 69.2	2.70 -	2.70 -	- -	- -	173
			Hot	Nm. Hot	219 155	217 153	1.41 -	1.41 -	14, 10 14, 10	14, 10 14, 10	175
c7552	3836	207	Nm.	Nm. Hot	155 315	152 281	2.03 -	1.85 -	- -	- -	24.8
			Hot	Nm. Hot	734 382	722 339	1.92 -	2.13 -	33, 27 38, 23	39, 25 41, 17	25.3
b11	633	38	Nm.	Nm. Hot	15.7 34.2	13.9 27.9	2.18 -	2.01 -	- -	- -	22.3
			Hot	Nm. Hot	114 54.6	96.6 43.9	2.08 -	2.20 -	15, 25 18, 14	17, 23 20, 14	22.3
b12	1183	126	Nm.	Nm. Hot	19.7 39.6	17.3 32.2	2.01 -	1.86 -	- -	- -	17.5
			Hot	Nm. Hot	129 60.5	115 48.3	2.13 -	2.37 -	30, 22 30, 8	30, 22 32, 7	17.5
b13	375	62	Nm.	Nm. Hot	2.00 9.10	1.78 7.22	4.05 -	4.54 -	- -	- -	6.82
			Hot	Nm. Hot	17.2 15.9	15.0 13.5	1.08 -	1.11 -	13, 31 26, 8	18, 31 34, 8	6.82
b17	28911	275	Nm.	Nm. Hot	191 269	171 198	1.41 -	1.16 -	- -	- -	40.6
			Hot	Nm. Hot	1230 506	1110 390	2.43 -	2.84 -	25, 27 28, 21	27, 26 32, 20	40.7
b19	85188	485	Nm.	Nm. Hot	235 404	213 343	1.72 -	1.61 -	- -	- -	43.6
			Hot	Nm. Hot	1350 530	1260 449	2.54 -	2.79 -	35, 22 38, 13	39, 21 41, 12	49.9

- [20] L. Li et al., "An efficient algorithm for library-based cell-type selection in high-performance low-power designs." *ICCAD*, 226-232, 2012.
- [21] J. Hu et al., "Sensitivity-guided metaheuristics for accurate discrete gate sizing." *ICCAD*, 233-239, 2012.
- [22] Z. Zhang et al., "Gradual relaxation techniques with applications to behavioral synthesis." *ICCAD*, 529-536, 2003.
- [23] Y. Alkabani et al., "Input vector control for post-silicon leakage current minimization in the presence of manufacturing variability." *DAC*, 606-609, 2008.
- [24] Y. Alkabani et al., "N-version temperature-aware scheduling and binding." *ISLPED*, 331-334, 2009.
- [25] J. B. Wendt et al., "Improving energy efficiency in sensing subsystems via near-threshold computing and device aging." *IEEE Sensors*, 2013.
- [26] S. Wei et al., "Integrated circuit digital rights management techniques using Physical level characterization." *DRM*, 3-14, 2011.
- [27] S. Wei et al., "Malicious circuitry detection using thermal conditioning." *TIFS*, 1136-1145, 2011.
- [28] S. Wei et al., "Gate characterization using singular value decomposition: foundations and applications." *TIFS*, 765-773, 2012.
- [29] S. Wei et al., "Scalable hardware trojan diagnosis." *VLSI*, 1049-1057, 2012.
- [30] S. Wei et al., "Low power FPGA design using post-silicon device aging." *FPGA*, 2013.