Total Power Optimization By Simultaneous Dual-Vt Allocation and Device Sizing in High Performance Microprocessors

Tanay Karnik, Yibin Ye, James Tschanz, Liqiong Wei, Steven Burns[¶],
Venkatesh Govindarajulu, Vivek De, Shekhar Borkar
Circuit Research, Intel Labs, 5200 NE Elam Young Parkway, Hillsboro, OR 97124.
[¶]Strategic CAD, Intel Labs, 5200 NE Elam Young Parkway, Hillsboro, OR 97124.
(503) 696 1975, 001
tanay.karnik@intel.com

ABSTRACT
We describe various design automation solutions for design migration to a dual-Vt process technology. We include the results of a Lagrangian Relaxation based tool, iSTATS, and a heuristic iterative optimization flow. Joint dual-Vt allocation and sizing reduces total power by 10+% compared with Vt allocation alone, and by 25+% compared with pure sizing methods. The heuristic flow requires 5x larger computation runtime than iSTATS due to its iterative nature.

Categories and Subject Descriptors
B.7 INTEGRATED CIRCUITS
B.7.1 Types and Design Styles – *Microprocessors and microcomputers, VLSI.*

General Terms
Algorithms, Performance, Design, Experimentation, Verification.

Keywords
Dual-Vt design, multiple threshold, sizing, optimization.

1. INTRODUCTION
Total power consumption of a high performance microprocessor (μP) is exceeding 100 Watts. Power density has exceeded that of a hot plate. μP's used for mobile applications have an additional battery life constraint. Total power consists of switching and leakage components. These two components can be traded off to minimize total active power consumption.

Leakage power minimization has prompted various IC manufacturers to employ multiple threshold voltage (Vt) processes. Most of the process technologies provide dual threshold voltage transistors (dual-Vt). Low-Vt transistors are used in performance-critical blocks to meet target clock frequency requirements, and high-Vt transistors are used in blocks with delay slacks to minimize overall leakage power. Various techniques [2,3,6-9,11,12] have been published for dual-Vt design, synthesis or process migration of libraries. To improve performance, upsizing of a high-Vt transistor, which increases switching power and die area, can be traded off against using a low-Vt transistor, which increases leakage power. Few techniques consider transistor sizing and Vt allocation as an integrated problem [4,5,10]. However, the reported techniques are either not evaluated for complex industrial designs [4,5], or focus on reduction of standby leakage power only [10], not total active power of μP's.

In this paper, we describe two methods for optimized migration of an existing single-Vt design database to a dual-Vt process technology: (1) a deterministic global optimizer for simultaneous Vt allocation and transistor sizing (iSTATS), and (2) a heuristic iterative optimization flow using an industry-standard sizing tool (AMPS) and a Transistor-level Dual-Vt Allocator (TA-DVT) tool. An existing μP database in single-Vt process is used for testing the two methods, and for understanding the trade-offs offered by the different Vt allocation and sizing options.

The paper is organized as follows. Section 2 provides a detailed description of our Lagrangian Relaxation (LR) based iSTATS tool with parameter estimation, constraints, assumptions, algorithms, implementations and results. Section 3 contains similar details of the iterative heuristic method. Results are discussed and analyzed in section 4, and conclusions are presented in Section 5.

2. SIMULTANEOUS VT ALLOCATION AND TRANSISTOR SIZING BY iSTATS

2.1 Estimation
The iSTATS optimizer performs gate-level timing analysis internally. The circuit blocks contain continuously sizeable CMOS logic gates of various types. The granularity of allocation is limited to half gate-level, that is, mixed Vt's are not allowed in the pull-up PMOS tree or pull-down NMOS tree inside a gate. This results in four different versions of a static CMOS logic gate - *hphn*, *hpln*, *lphn* and *lpln*. The first/third letter indicates whether the P/N tree is high-Vt (*h*) or low-Vt (*l*).

2.1.1 Gate Delays
The gate library is simulated for varying conditions of input slope, normalized load (output load capacitance divided by driving conductance) and P/N transistor skew. All library gates are designed to have two independent size variables per gate. Rise/fall delay and signal slope are modeled by separate equations for different Vt versions of a gate.

Conference '02, June 10-14, 2002, New Orleans, Louisiana.

2.1.2 Path Delays

iSTATS includes a static timing analyzer that supports time borrowing across transparent latches. The results are correlated with Pathmill, an industry-standard static timing analyzer.

2.1.3 Interconnect Parasitics

The optimization is applied to a pre-existing microprocessor database. The interconnect capacitances are assumed to scale according to the new process technology. Interconnect resistances are assumed to be insignificant for this optimization.

2.1.4 Power

Block-level switching (P_A) and leakage (P_L) powers are:

$$P_A = a[c_T (W_{hV_t} + W_{lV_t}) + C_I]V^2 f; \quad P_L = (\lambda W_{hV_t} + W_{lV_t}) I_{off} \beta V$$

$$P_{total} = P_A + P_L$$

where P_{total} is total power, a is activity factor, c_T is capacitance per unit width, $W_{hVt}[W_{lVt}]$ is high[low]-Vt device width, V is supply voltage, f is target frequency, C_I is total interconnect capacitance, λ is h_{Vt} to l_{Vt} leakage ratio, I_{off} is l_{Vt} leakage current per unit width, β is an empirical factor which accounts for impacts of within-die L variations, stack effect, noise and fraction of "*off*" transistors on leakage power. Short-circuit power is modeled simply as an increase in a, since it is typically less than 10% of the switching power. Its dependencies on Vt and signal slopes are ignored.

2.2 Assumptions

The clock skew is assumed fixed and ideal clock waveforms are provided at the clock inputs. Furthermore, since migration of the design database to next generation process technology is in an early phase, we assume that very stringent delay requirements for few critical paths in some circuit blocks can be relaxed.

2.3 Optimization Algorithm

The optimization algorithm is based on the well-known Lagrangian Relaxation technique [1]. The problem formulation:

Minimize $\alpha(P_{total}) + WNS + WS$

Subject to $\forall_{u \to v} A_u + d_{u \to v} \le A; \quad \forall_{pi} A_{pi} = A^T_{pi} + adj_{pi};$

$$\max_{u \to v}(s_{u \to v} - s_{limit}) = WS; \quad \max_{po}(A_{po} - r_{po}) = WNS$$

where α is power-delay tradeoff factor, *WNS* is worst negative slack, *WS* is worst slope, *pi* is primary input, *po* is primary output, A is arrival time, r is required time, d is delay, A^T is arrival time constraint, *adj* is adjustment based on changing load capacitance. Applying LR, the objective becomes

$$i_{STATS}(\mu, \sigma, w) = \alpha[W_{avg} + \lambda W_{avglV_t}] + \Sigma_{u \to v} \mu_{u \to v} d_{u \to v} + \Sigma_{po} \mu_{po}(-r_{po}) +$$

$$\Sigma_{pi} \mu_{pi}(a^T_{pi} + adj_{pi}) + \Sigma_{u \to v} \sigma_{u \to v}(s_{u \to v} - s_{limit})$$

where the multipliers μ and σ signify delay and slope criticality.

Figure 1 shows the algorithmic flowchart. The inner loop traverses the netlist in reverse topological order to numerically solve for width and Vt variables. As Vt is not a continuous variable, there is a slight quality degradation. Inner loop complexity is O(n). Timing analysis and multiplier update are performed during the outer loop. Lower bound is tracked during the outer loop to monitor convergence.

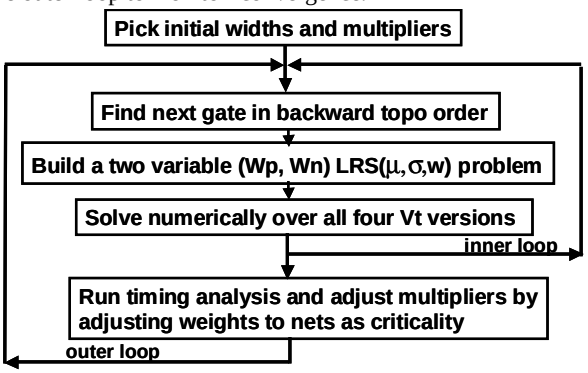

Figure 1: iSTATS algorithm flowchart

2.4 Implementation and Results

iSTATS is implemented as a modular and extensible C-code. The experiments are executed on HP-UX B.10.20. Various static CMOS circuit blocks in an existing microprocessor are selected as test cases. The design optimizations are performed in a 0.13μm dual Vt process for 2.2 Ghz target clock frequency.

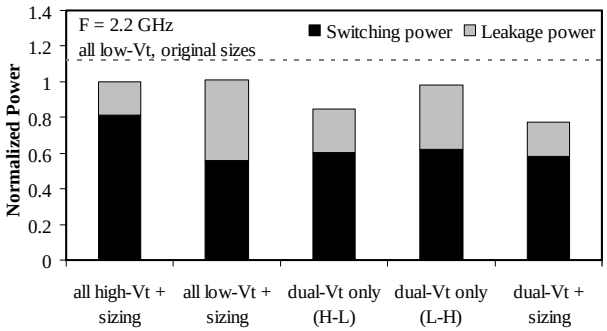

Figure 2: Switching and leakage power tradeoffs

Figure 2 shows P_{total}, with P_A and P_L components, for different design optimizations. The power is averaged over a number of circuit blocks for the following design options:

1. start with all high Vt, perform sizing optimization
2. start with all low Vt, perform sizing optimization
3. start with all high Vt, perform optimal low Vt allocation
4. start with all low Vt, perform optimal high Vt allocation
5. simultaneously optimize Vt allocation and sizing.

The total power achieved by simultaneous Vt allocation and sizing is at least 10% smaller than other design options that which use dual-Vt allocation without sizing, and is 20% smaller than either of the single-Vt design options where only sizing is used. Among "dual-Vt only" options, option 3 is better, indicating that high-to-low Vt conversion is preferred.

Figure 3 shows histograms of average device width and low-Vt usage in the different circuit blocks for options 3 and 5. Clearly, low-Vt usage and average width vary by large amounts from block to block since they are dependent on the inherent logic topology of the block and its timing requirement. This holds true for both design options. Therefore, design optimizations and guidelines for Vt allocation and sizing must be derived by comprehending the wide assortment of circuit block types in a complex μP design.

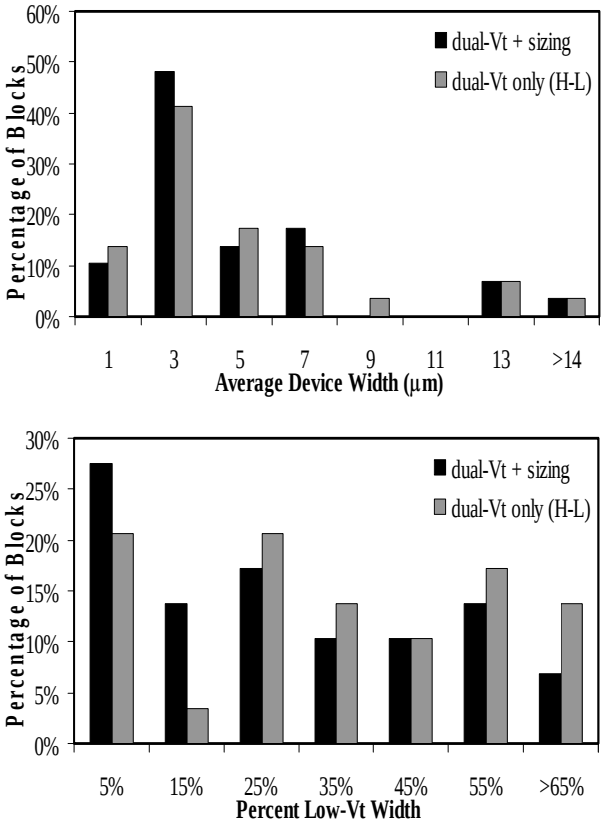

Figure 3: iSTATS results for different circuit blocks

3. HEURISTIC ITERATIVE OPTIMIZER

In IC design community, Pathmill is commonly used for static timing analysis. AMPS, based on pathmill, is widely used for sizing a circuit. There are several heuristic dual-Vt allocation algorithms reported in [4,5,10]. We developed an algorithm for priority-based transistor-level allocation of dual-Vt (TA-DVT). Both AMPS and TA-DVT are used in an iterative flow to achieve optimization by conjoint sizing and dual-Vt allocation. Parasitics for delay and power estimation are extracted from the original 0.18μm microprocessor design and scaled to 0.13μm process technology. After sizing and Vt allocation, we use the original interconnect parasitics, but re-estimate the transistor-related parasitics. Power estimations are the same as that described in the section 2 and include gate oxide leakage power. Some transistors are marked as "don't touch" during Vt allocation, and forced to be high-Vt. For example, many transistors in clock generation circuits and dynamic circuits are excluded for Vt change. We identify these "don't touch" transistors based on our design methodology for noise. Additionally, some transistors need to have the same Vt based on the design requirements. For instance, multiple array instances of the same circuit template may need to have the same Vt due to design and layout considerations.

3.1 Flows and Algorithms

We consider 4 flows that use AMPS for sizing and TA-DVT for Vt allocation. They are shown in Fig. 4(a). In flow 1, only sizing is done using AMPS. In flow 2, only dual-Vt allocation is performed. In flow 3, we do a heuristic iterative optimization with both sizing and dual-Vt allocation to minimize total power. The heuristics proposed in flow 3 introduce an intermediate performance target for each iteration, as illustrated in Fig. 4(b). AMPS is used to meet only the intermediate performance target and TA-DVT is then used to meet the final target. We use five different intermediate performance targets, relaxed by 0%, 7%, 15%, 22% and 30% from the final target. The design with the lowest total active power is then selected. We find that the best intermediate target depends strongly on the circuit block type. A second AMPS run is used after TA-DVT, as some paths may be faster than the target due to low-Vt allocations, creating opportunities for further downsizing. We also implement a flow 3x in which all transistors are converted to low-Vt first. Then we use AMPS which mostly downsizes the devices. TA-DVT is then used for selective high-Vt allocation.

3.2 TA-DVT Algorithm

The TA-DVT flow is illustrated in Figure 5. We first estimate parasitics of the initial circuit. The circuit can be all high-Vt, all low-Vt or dual-Vt to begin with. Pathmill is then used to identify critical paths. We extract slack S_p of all critical paths reported by Pathmill, and delay, d_{ip}, of all transistors in these paths. Note that the delay of a transistor is path dependent, i.e., the delay of the same transistor i can be different in path p and path q, or $d_{ip} \neq d_{iq}$. We also need to know the delay improvement Δd from high-Vt to low-Vt for all candidate transistors. If the initial circuit includes all high-Vt transistors, then we perform another Pathmill run with all low-Vt, and vice versa. For common paths found in both Pathmill runs, we extract the path-dependent delay improvement, Δd_{ip}, for transistor i in path p. When the corresponding path in the second pathmill run is not found for path q, the average transistor delay improvement $\Delta d_{i,}$, which is the difference between average high-Vt delay and low-Vt delay in the two Pathmill runs for transistor i, is used for path q. We find that the percentage delay change $\Delta d_i/d_{i,}$, varies significantly among transistors because the transistor delay is strongly dependent on the input signal slope, stack effect, the fraction of interconnect capacitance in the load, etc. However, the variation in delay change for a transistor among different paths running through it is small.

For low-Vt assignment, priority of each candidate transistor i is computed based on the following weight function:

$$\mathbf{W_i} = \Sigma_{\forall\ p \text{ covered by } i \text{ with } Sip < 0} \mathbf{MIN}(\Delta d_{ip}, |S_{ip}|) / w_i$$

where w_i is the width of transistor i and S_{ip} is the slack of transistor i in path p. Only paths with negative slack, covered by the transistor, are considered in the summation. Transistors with smaller width and higher impact on negative slack due to Vt change have higher weights. The candidate transistors are stored in a queue implemented as a binary heap: the transistor with the highest weight is placed at the top of the heap. In each round of low-Vt allocation, the top transistor in the heap is selected, converted to low-Vt, and removed from the heap. The slacks of all paths affected by the new low-Vt transistor are updated, along with their weights. Transistors that do not belong to critical paths are removed from the heap and the heap is updated with a new transistor on the top. The low-Vt allocation process continues until the heap is empty. The low-to-high Vt allocation works in a similar way as shown in Fig. 5. However, a different weight function is used.

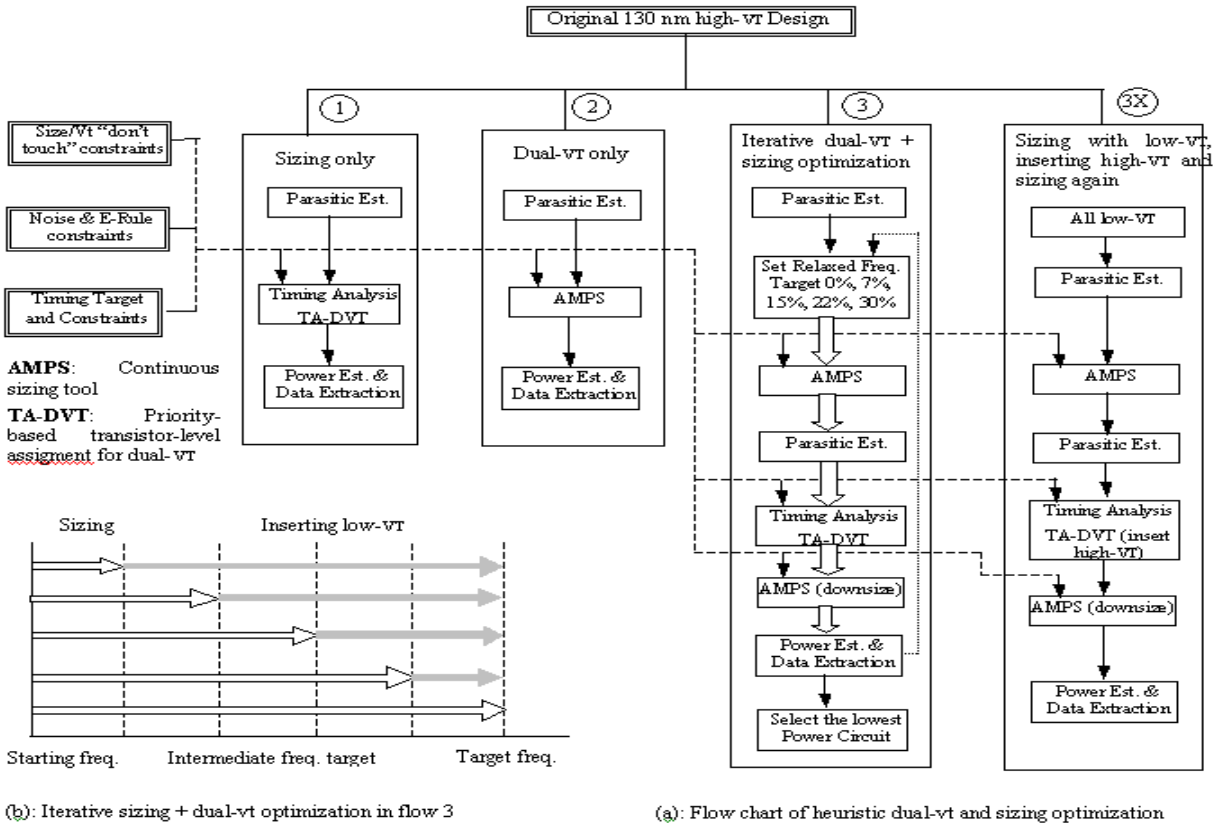

Figure 4: Flowcharts of heuristic iterative optimization (using AMPS and TA-DVT) and different design options

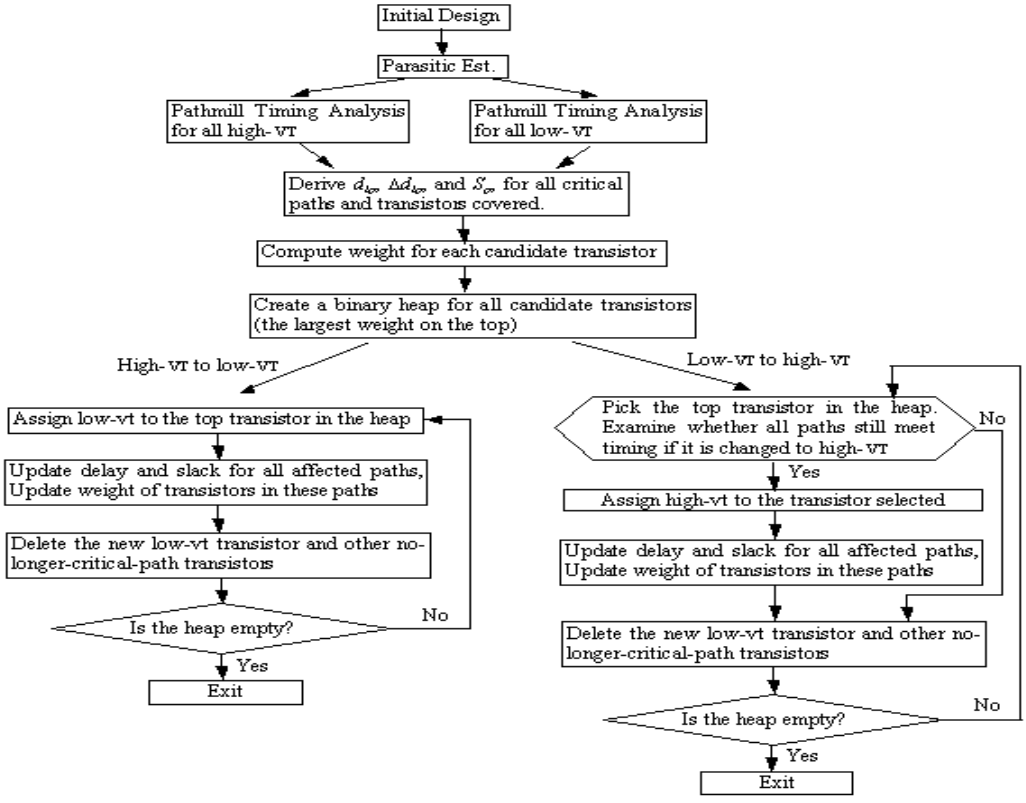

Figure 5: Flowchart of priority-based transistor-level allocation of dual-Vt (TA-DVT)

3.3 Implementation and Results

In Fig. 6, we use a circuit block as an example to illustrate the dynamics of flow 3. The lowest total power is obtained when the intermediate target is relaxed by 15%. As expected, both low-Vt percentage and leakage power increase as the intermediate target is relaxed further, since more low-Vt transistors are needed to meet the final performance target. Fig. 7 summarizes total power of 100 circuit blocks for different frequency targets for all techniques. The "iterative sizing + dual-Vt allocation" produces the best solution. However, "sizing with all lowVt + low-to-high dual-Vt" provides approximately the same power. "Sizing only" and "dual-Vt only" are comparable in power, but they cannot meet the high performance targets achievable by the other techniques. Fig. 8 shows histograms of percentage low-Vt device width and average transistor width for all circuit blocks at 2GHz clock frequency for "dual-Vt only" and "sizing + dual-Vt". The dual-Vt optimization varies significantly across circuits. The number of circuit blocks with large low-Vt percentage is substantially smaller in "dual-Vt + sizing" than in "dual-Vt only". An activity factor of 0.1 is used in the dual-Vt + sizing optimizations. However, the optimization is not very sensitive to activity factor assumption, as shown in Fig. 9.

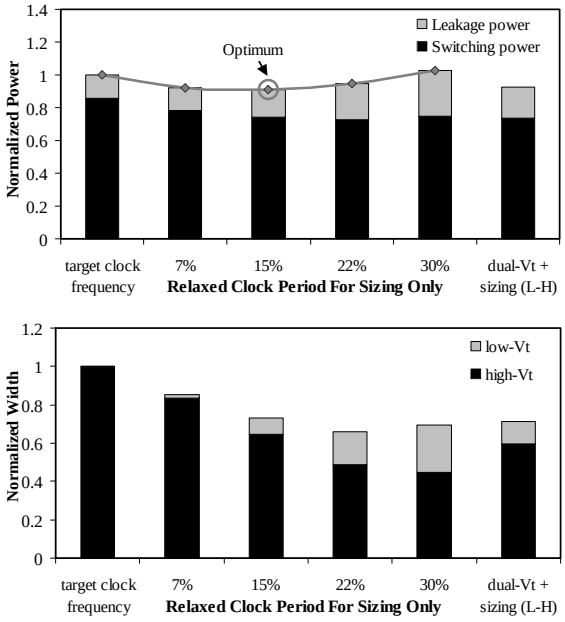

Figure 6: Power & width dynamics of heuristic optimization

4. RESULTS AND DISCUSSION

Run time of the iSTATS deterministic global optimizer is significantly shorter than the iterative optimizer across a large sample of circuit blocks. For example, for a block containing 10000 transistors, iSTATS is 5x faster than the heuristic flow. However, the AMPS + TA-DVT optimization is easier to implement since it uses pre-existing tools for timing, sizing, and Vt allocation. No library characterization is needed. In addition, transistor-level Vt allocation and continuous sizing are possible to provide the best spatial granularity for the optimization.

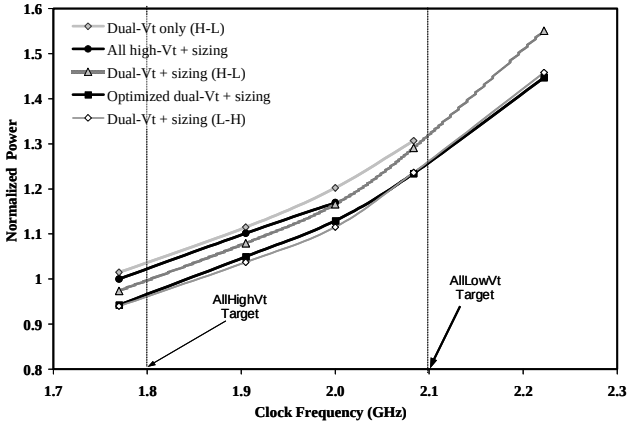

Figure 7: Total power versus frequency for 5 techniques

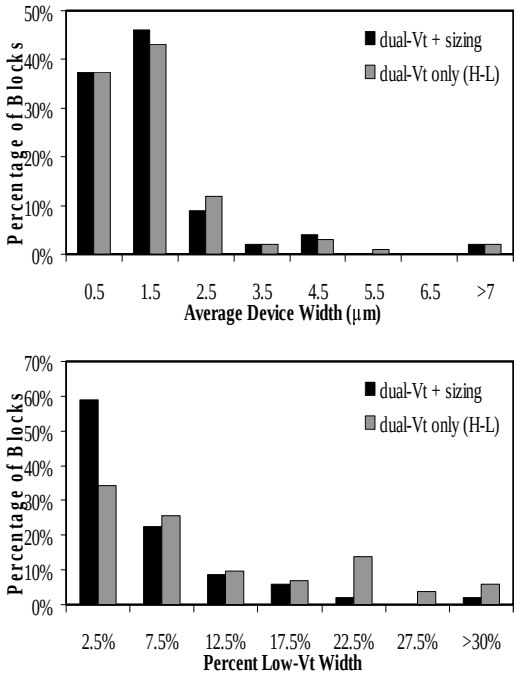

Figure 8: Histograms of low-Vt usage and device width

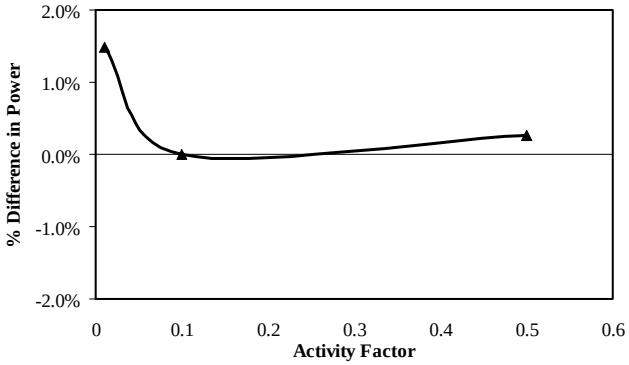

Figure 9: Sensitivity of power sensitivity to activity factor for dual-Vt+sizing optimization

Figs. 10 and 11 show results of the optimization across a range of target clock frequencies and activity factors for 100 functional unit blocks containing 500-50000 transistors per block. With all high-Vt devices and original sizes, typical achievable clock frequency would be 1.8GHz. If all devices are converted to low-Vt, 2.1GHz frequency is achievable without any resizing. Target 2GHz clock frequency can be achieved by sizing only with all high-Vt. The total power for 0.1 activity factor is 5% smaller than a design with dual-Vt only.

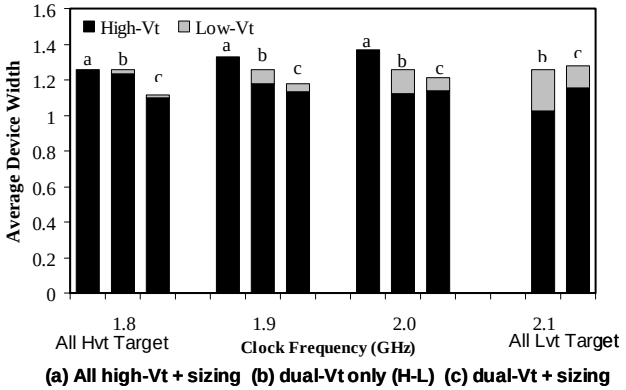

Figure 10: Average device width vs. frequency for various design options

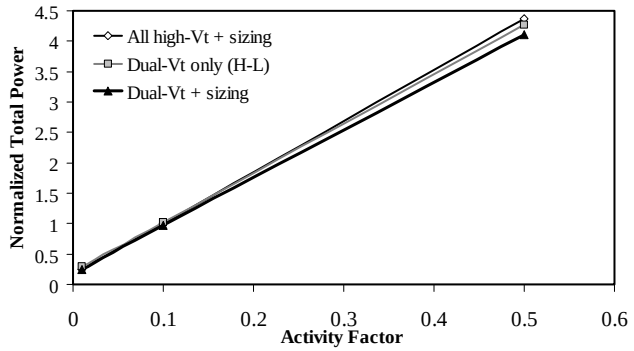

Figure 11: Power vs. activity for various design options at 2.0Ghz clock frequency

Even though using low-Vt devices prevents upsizing of transistors resulting in smaller switching power and reduced average device width than the "sizing only" option, leakage power is significantly larger. As a result, total power is increased. Conjoint dual-Vt and sizing achieves a more optimal balance between low-Vt usage and transistor upsizing needed to achieve the target 2GHz clock frequency. As a result, total power and average device width are 5-10% smaller than the previous two design options. A design with only high-Vt devices fails to achieve 2.1GHz clock frequency target by sizing only, whereas designs containing dual-Vt devices are able to meet this aggressive performance target. The total power is 5-10% smaller when both sizing and dual-Vt are optimized jointly, when compared to optimal dual-Vt allocation only with no transistor resizing. Leakage power is significantly smaller for the design with joint dual-Vt and sizing, since it reduces low-Vt usage from 40% to 15% to achieve proper balance between switching and leakage power components that minimizes total power. The average device widths are approximately the same for both. At higher values of activity factors, leakage power has less impact on total power. As a result, techniques that favor more low-Vt usage become more beneficial. Downsizing of transistors allowed by more low-Vt usage reduces switching power, and thus total power, significantly without incurring major impacts of additional leakage from the low-Vt devices. Thus, benefits of joint dual-Vt and sizing over "dual-Vt only" or "sizing only" are more pronounced when activity is higher.

5. CONCLUSIONS

Various design automation solutions for single Vt design migration to a dual-Vt process technology have been presented. The results are generated using a Lagrangian Relaxation based tool, iSTATS and a heuristic iterative optimization flow. Dual-Vt allocation and sizing reduces total power by at least 10% compared with pure Vt allocation methods and at least 25% compared with pure sizing methods.

6. REFERENCES

[1] Chen, C.P., Chu, C., and Wong, D.F., Fast and exact simultaneous gate and wire sizing by LR. IEEE TCAD, Vol. 18, No. 7, July 1999, 1014-1025.

[2] Ishihara, T., and Asada, K., **A** system level memory power optimization technique using multiple supply and threshold voltages**.** IEEE/ACM DAC 2001, 456-461.

[3] Ko, U., Pua, A., Hill, A., and Srivastava, P., Hybrid dual-threshold design techniques for high-performance processors with low-power features. ISLPED 1997, 307-311.

[4] Pant, P., Roy, R., and Chatterjee, A., Dual-Threshold Voltage Assignment with Transistor Sizing for Low Power CMOS Circuits. TVLSI, Vol. 9, No. 2, 4, 2001, 390-394.

[5] Sirichotiyakul, S., et al, Stand-by power minimization through simultaneous threshold voltage selection and circuit sizing. DAC 1999, 436-441.

[6] Sundararajan, V., and Parhi, K. K., Low power synthesis of dual threshold voltage CMOS VLSI circuits. IEEE ISLPED 1999**,** 139-144.

[7] Tripathi, N., Bhosle, A., Samanta, D., and Pal, A., Optimal assignment of high threshold voltage for synthesizing dual threshold CMOS circuits. VLSI Design, India, 2001, 227-232.

[8] Wei, L., et al, Design and Optimization of Dual-Threshold Circuits for Low-Voltage Low-Power Applications. IEEE TVLSI, Vol. 7, No. 1, March 1999, 16-23.

[9] Wei, L., Chen, Z., and Roy, K., Ye, Y., De, V., Mixed-Vth (MVT) CMOS Circuit Design Methodology for Low Power Applications. ACM/IEEE DAC 1999, 430-435.

[10] Wei, L., Roy, K., and Koh, C. K., Power Minimization by Simultaneous Dual-Vth Assignment and Gate-sizing. IEEE CICC 2000, 413-416.

[11] Wong, Q., and Vrudhula, S.B.K., Static power optimization of deep submicron CMOS circuits for dual V/sub T/ technology. IEEE/ACM ICCAD 1998, 490-496.

[12] Wong, Q., and Vrudhula, S.B.K., An investigation of power delay trade-offs for dual V/sub t/ CMOS circuits. ICCD 1999, 556-562.