

# Pushing ASIC Performance in a Power Envelope

Ruchir Puri, Leon Stok, John Cohn\*  
David Kung, David Pan  
IBM Research, Yorktown Hts, NY  
\*IBM Microelectronics, Essex Jn, VT

{ruchir,leonstok,johncohn,kung,dpan}@us.ibm.com

Dennis Sylvester, Ashish Srivastava  
Sarvesh Kulkarni  
EECS, University of Michigan  
Ann Arbor, MI

{dmcs,ansrivas,shkulkar}@umich.edu

## ABSTRACT

Power dissipation is becoming the most challenging design constraint in nanometer technologies. Among various design implementation schemes, standard cell ASICs offer the best power efficiency for high-performance applications. The flexibility of ASICs allow for the use of multiple voltages and multiple thresholds to match the performance of critical regions to their timing constraints, and minimize the power everywhere else. We explore the trade-off between multiple supply voltages and multiple threshold voltages in the optimization of dynamic and static power. The use of multiple supply voltages presents some unique physical and electrical challenges. Level shifters need to be introduced between the various voltage regions. Several level shifter implementations will be shown. The physical layout needs to be designed to ensure the efficient delivery of the correct voltage to various voltage regions. More flexibility can be gained by using appropriate level shifters. We will discuss optimization techniques such as clock skew scheduling which can be effectively used to push performance in a power neutral way.

## Categories and Subject Descriptors

B.7 [Integrated Circuits]: Design Styles—Design Aides

## General Terms

Algorithms, Performance, Design

## Keywords

ASIC, Design Optimization, High-Performance, Low-Power

## 1. INTRODUCTION

Power efficiency is becoming an increasingly important design metric in deep sub-micron designs. One of the major advantages of ASICs compared to other implementation methods is the power advantage. A dedicated ASIC will have a significantly better power-performance product than a general purpose processor or regular fabrics such as FPGAs. For designs that push the envelope of power and performance, ASIC technology remains to be the only choice. However, the cost pressures in nanometer technologies are forcing designers to push the limits of design technology in

order to fully exploit increasingly complex and expensive technology capabilities. In this paper, we discuss technology, circuit, layout and optimization techniques to improve the power delay product. We focus on the issue of pushing ASIC performance in a power envelope by exploiting the use of multiple supply voltages (Vdd) and multiple device thresholds (Vth). In section 2, we discuss the trade-off between multiple Vdd and multiple Vth options to optimize power. In section 3, we present novel design techniques to physically implement fine-grained generic voltage islands for multiple-Vdd implementations. In the context of multi-Vdd implementation, we also present some novel level conversion circuits which can be used to implement very flexible voltage island schemes. In section 4, we discuss optimization techniques such as clock skew scheduling which can be effectively used to push performance in a power neutral way. Finally, we present a design case study to show the relative impact of some design techniques in a low-power ASIC methodology.

## 2. POWER - PERFORMANCE TRADE - OFF IN MULTI-VDD / VTH TECHNOLOGIES

This section explores the trade-off between multiple supply voltages and multiple threshold voltages in the optimization of dynamic and static power. From a dynamic power perspective, supply voltage reduction is the most effective technique to limit power. However, the delay increase with reducing Vdd degrades the throughput of the circuit. Similarly, to reduce static power an increase in Vth provides exponential improvements, again at the expense of speed. To counter the loss in performance, dual Vdd [1, 2] and dual Vth [3, 4, 5] techniques have been proposed. These approaches assign gates on critical paths to operate at the higher Vdd or lower Vth and non-critical portions of the circuit operate at lower Vdd or higher Vth, reducing the total power consumption without degrading performance (held fixed as a constraint). These techniques have been successfully implemented but most of the existing work focuses on one of these techniques in isolation as opposed to jointly.

Previous work [6] estimates the optimal Vdd and Vth values to be used in multi-voltage systems to minimize either dynamic or static power respectively. They confirm earlier work [7] claiming that, in a dual Vdd system the optimal lower Vdd is 60-70% of the original Vdd. In general, [7, 8] have found optimized multi-Vdd systems to provide dynamic power reductions of roughly 40-45%. In [9], it is shown that intelligently reducing Vth in multi-Vdd systems can offset the traditional delay penalties at low-Vdd with lessened static power consequences (due to both the reduced Vdd and Ioff levels). In order to explore the achievable design envelope in a joint multiple Vdd and Vth environ-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2003, June 2-6, 2003, Anaheim, California, USA.

Copyright 2003 ACM 1-58113-688-9/03/0006 ...\$5.00.

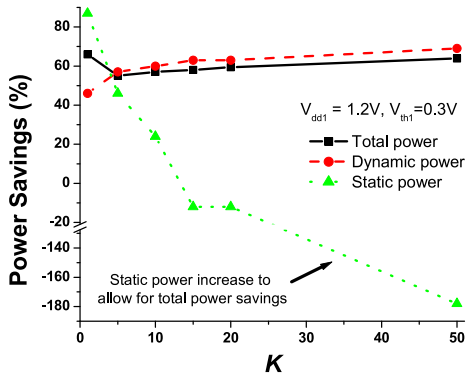


Figure 1: Breakdown of total power savings into static and dynamic components.

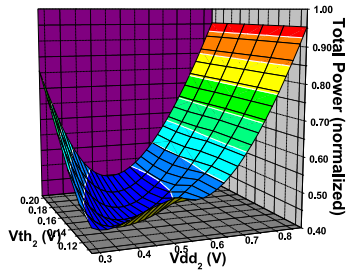


Figure 2: Power reduction as a function of second Vdd and Vth.

ment, we formulate a linear programming problem to minimize power by assigning capacitance (representing gates) to a combination of supply and threshold voltages (assuming a known initial path delay distribution) [10]. Our general framework is similar to [6], but enables several key enhancements: 1) we minimize total power consumption, defined as the sum of static and dynamic components, 2) we simultaneously optimize both Vdd and Vth to achieve this goal, and 3) we consider DIBL, which strongly limits the achievable power reduction in a multi-Vdd, single Vth environment.

Our results indicate that the total power reduction achievable in modern and future integrated circuits is on the order of 60-65% using the dual Vdd/Vth technique (figures 1 and 3). A key factor when optimizing a multi-Vdd/Vth system is a parameter K which is the ratio of dynamic to static power in the original single Vdd/Vth design. Larger K values push the optimization towards lower Vdd and Vth to address the dominant dynamic power. An important finding is that the optimal second Vdd in multi-Vth systems is approximately 50% of the higher supply voltage which is contrasted with 0.6-0.7\*Vdd1 for single Vth designs as previously found. An implication of this finding is that level converter structures must be capable of converting over a larger relative range. This seems feasible provided the level converters themselves leverage multiple threshold voltages.

Continued aggressive channel length scaling (without commensurate supply voltage reductions) and new device structures such as strained-Si channels point to increasingly velocity saturated devices that are ideal for voltage scaling (figure 2). The inclusion of level conversion delay penalties demonstrates the trade-off between allocating available slack to level conversion and achievable power reductions. Typically, 1-2 asynchronous level conversions per path are tolerable in designs with larger logic depths (30+ FO4 de-

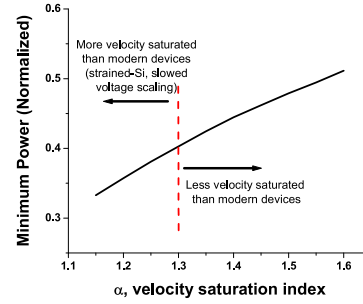


Figure 3: Future devices may be more velocity saturated, yielding lower power consumption.

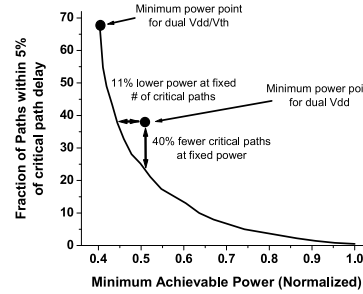


Figure 4: Dual-Vdd/Vth provides better power-criticality trade-off than dual-Vdd for same power.

lays) with <15% power penalty. Additionally, we note the relationship between power savings and critical path density; this is important since a rapidly increasing number of critical paths combined with rising process variability increases design times and emphasizes a need for improved statistical timing analysis tools. Dual Vdd/Vth offers better control of the slack-power trade-off compared to dual Vdd only as shown in figure 4. In future designs that are both power and variability-constrained, the design space of figure 4 may become crucial.

For designs that do not demand ultra low power, designers can avoid the physical design issues associated with the use of multiple supply voltages on a chip by aggressive scaling of a single Vdd combined with multiple device threshold voltages (as illustrated by the case study in section 5). For instance, the use of 1.2V as Vdd for 130nm technologies is commonplace and assumed in the above discussion. However, the use of a single 0.9V supply with a small subset of gates using an ultra-low Vth to maintain speed may yield lower overall power. To investigate this possibility, we use the same design space exploration tool as above to look at the efficacy of single Vdd/multi-Vth design. Again, we normalize power to the single Vdd, single Vth design point. In table 1 we see that the potential improvements from a single Vdd/multi-Vth system can be quite substantial especially when K is large. For a reasonable K value of 10, a single Vdd system can provide 65-77% of the gains that dual Vdd/Vth shows depending on the number of threshold voltages used (2 or 3). Furthermore, the numbers for dual Vdd/Vth do not include level conversion penalties so can be considered as best-case power reductions. Contrary to the dual Vdd case, the inclusion of a third Vth when a single optimized (flexible) supply voltage is used provides appreciable gains beyond the dual-Vth system. Since each extra mask step for an additional Vth level increases the wafer

**Table 1: Comparison among single Vdd and dual Vdd techniques. Initial design point has Vdd=1.2V and Vth=0.3V.**

K	Minimum achievable power (normalized to single Vdd/Vth)			Vdd used in single Vdd/	
	Dual Vdd/ dual-Vth	Single Vdd/		dual -Vth	triple -Vth
		dual-Vth	triple-Vth		
1	0.34	0.54	0.48	1.20	1.10
5	0.45	0.67	0.62	0.93	0.87
10	0.43	0.63	0.56	0.89	0.81
15	0.42	0.61	0.52	0.89	0.75
20	0.41	0.58	0.49	0.83	0.75
50	0.36	0.50	0.41	0.77	0.69

fabrication cost by 3%, use of multiple supply voltages by itself remains a very attractive choice for power-reduction. In the following section, we discuss the electrical and physical design issues of multiple Vdd implementations.

### 3. DESIGN ISSUES IN MULTI- VDD ASICS

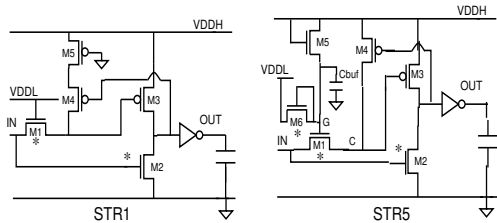
Design of ASICs with multiple supply voltages presents some unique electrical and physical design challenges. In this section, we present some novel solutions to these challenges.

#### 3.1 Circuit Design Issues

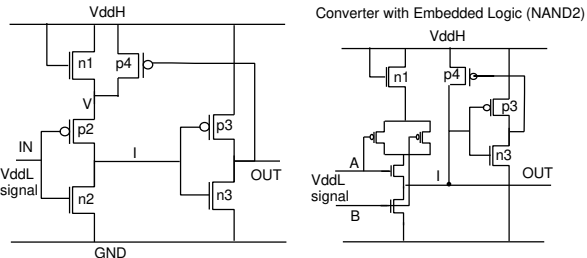
Electrically, to avoid excessive static power consumption between the low and high voltage regions, level converters (LC) need to be inserted. Minimizing the overhead of level converter insertion while meeting interfacing constraints presents a significant challenge. In this section, we describe some novel level converter circuits which not only provide efficient delay and power characteristics but also enable very flexible physical design of multi-Vdd schemes.

We have developed several versions of the low-energy asynchronous pass-gate (PG) based level converter from [8]. Figure 5 shows the new level converters. Transistors with (\*) indicate low-Vth devices. The first, STR1, relies on a known high-performance dynamic logic technique of splitting the keeper into two devices to minimize the capacitive load on the actual gate. STR5 employs the threshold drop of M5 to create a higher gate voltage for the pass-transistor and effectively speed it up. Transistor M6 is added to ensure that the gate voltage of M1 does not exceed VDDL + VTH which would yield reverse leakage current into VDDL. In comparison to the DCVS LC, STR 5 is up to 17% faster at the optimal delay point or consumes up to 50% less energy at fixed delay. STR1 has a simpler design and enables 30-40% lower energy than DCVS and 15-30% lower energy than the PG structure. Furthermore, we investigated the use of STR1 for embedded logic functionality and found that it is 4% faster with 55% lower energy than a 2-in NAND DCVS gate when the low Vdd is 0.8V (VddH=1.2V).

Level converters presented above require both a high as well as low power supply for level conversion. This limits the physical placement of such level converters to the boundary of high and low voltage designs which restricts the physical design flexibility. To address this, we developed a novel asynchronous level-converter, which requires only one supply (VddH) to convert the incoming low voltage signal to the higher voltage making its placement much more flexible [17] in the entire high voltage regions. In addition to the single supply advantage, this converter exhibits a significantly improved power dissipation compared to the traditional DCVS



**Figure 5: Dual-supply level converters**



**Figure 6: Single-supply level converters**

converter. Figure 6 shows the new single supply converter. We utilize the threshold drop across the nfet n1 to provide a virtual low-supply voltage to the input inverter (p2,n2). It was discussed in section 2 that the optimal low-supply voltage in a dual-supply design is generally 40% below the high supply. However, due to saturation of Vth in sub-100nm technologies, supply voltage cannot be scaled much below 1V. This is because in order to maintain good CMOS performance characteristics, it is desirable to have the ratio of Vth/Vdd below 0.3 [18]. Thus typically, the low supply in sub-100nm designs will be limited to 25-30% below the high-supply voltage. Figure 7 shows that when compared to traditional DCVS converter (in 130nm Cu11 technology with nominal Vdd=1.5V), the new converter achieves up to 5% better delay and consumes 50% less total power and approximately 30% less leakage power, in nominal operating range of low-voltage supply. The biggest advantage of this converter is its flexible placement which enables efficient physical design of fine-grained voltage islands as discussed in the following section.

#### 3.2 Physical Design Issues

Most of the previous work [11] in multi-Vdd designs has mainly focused on Clustered Voltage Scaling by Usami et al. [1]. Unfortunately, this methodology enforces a rigid circuit row based layout of high and low voltage cells. This can be overly restrictive as it may require significant perturbation in location of timing critical cells thereby degrading performance. In this section, we present some physical implementation schemes based on voltage islands which allow more flexibility in their layout.

##### 3.2.1 Macro based Voltage Islands

Recently, a new voltage island methodology to enable multiple supply voltages in SoCs was introduced [12] which allows various functional units of the ASIC/SoC to operate at different voltages. This voltage island methodology can be used in variety of designs. For example, in an SoC that integrates a processor core with memory and control logic, performance critical processor core requires highest voltage to maximize its performance. However, the on chip memory and control logic may not require the highest voltage

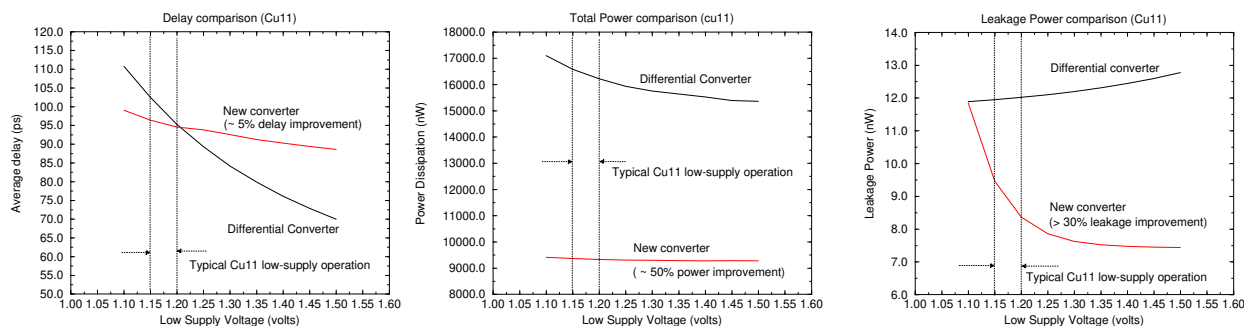


Figure 7: Comparison of DCVS converter with single-supply converter w.r.t change in low supply voltage.

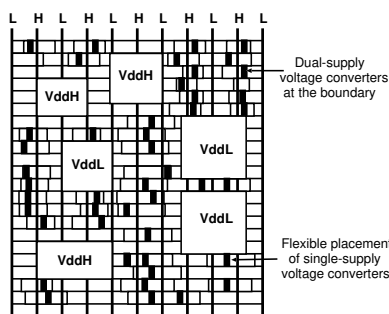


Figure 8: Generic voltage island layout style.

operation and can be operated at a reduced voltage to save significant active power without compromising system performance. In addition, voltage flexibility at unit level allows pre-designed standard components from other applications to be reused in a new SoC application. Voltage islands can also facilitate power savings in battery powered applications which are more sensitive to standby power. Traditionally, designers use power gating [13] to limit leakage current in quiescent states. The use of voltage islands at functional unit level in an SoC provides an effective physical design approach to gate the power supply of the entire macro in order to completely power it off.

### 3.2.2 Fine-Grained Generic Voltage Islands

The macro-based voltage island methodology is targeted towards an entire macro or functional unit to be assigned to a different voltage. In the case of designs that are highly performance critical as well as severely power constrained, it is useful to have a finer grained control over the supply voltages for ASICs or even within a macro/core in an SoC. We propose a flexible physical design approach which allows generic voltage islands and enables a fine grained implementation of the dual-supply voltage assignment in a placement driven synthesis framework [14]. A generic voltage island (GVI) structure with power grid is shown in figure 8, where we can assign different voltages at both macro and cell levels, and it has more freedom in terms of layout style by allowing multiple voltage islands within the same row. A generic design flow is built on top of an IBM's placement driven synthesis (PDS) design closure tool [15]. PDS integrates logic synthesis, placement, buffering, gate sizing, and multiple threshold voltages optimization [16]. The overall flow of generic voltage island is as follows. First, PDS timing closure is run with the entire circuit timed at VddH. For deep submicron circuits, interconnect delay dominates the gate delay. Thus we need a rough placement information to identify critical versus non-critical cells. Once PDS reaches

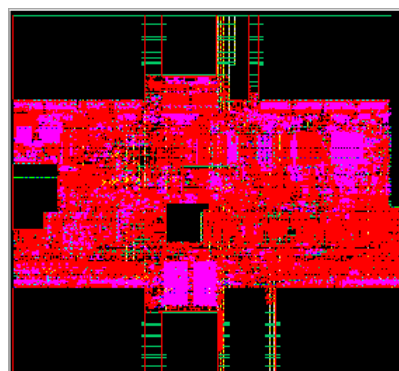


Figure 9: A processor with generic voltage islands.

a later stage of optimization, e.g., global placement is determined and timing is more or less closed, we can perform the generic voltage island generation, by assigning non-critical cells to a lower supply voltage. To minimize the physical design overhead, we consider two kinds of adjacencies during VddL macro/cell selection. One is the logic adjacency, i.e., the low voltage cells are as contiguous as possible in signal paths to minimize the number of level shifters. The other is the physical adjacency, i.e., low voltage cells are physically close to each other, so that it is easy to form voltage islands. Since GVI is implemented within the framework of PDS, we can employ various optimization engines during voltage assignment, e.g., to trade-off gate sizing with voltage assignment. After voltage assignment, low and high voltage cells are clustered to form the fine grained generic voltage islands. The clustering step requires the knowledge of power grid topology which is co-designed with this placement in order to enable a flexible placement of fine grained voltage islands. We first define the power grid patterns to facilitate the placement movement. They are computed based on the cell locations that are assigned to high and low voltage cells. Then we will move cells locally (while trying to maintain the original cell order) to form VddL and VddH islands. Traditionally, a dual-supply DCVS level converter is used to interface signals across VddL and VddH voltage islands. Since DCVS LC requires both VddL and VddH supplies, their placement is limited to the boundary of low and high voltage islands where both the supplies are easily available. To remove this placement restriction on level converter, we utilize the single supply voltage level-converter (figure 6). Since this converter requires only VddH supply, it can be placed anywhere in the VddH voltage islands, thereby enabling much more flexible placement. This results in significantly smaller physical design overhead for LC insertion as the converters can be inserted in uncongested regions.

We have applied this generic voltage island approach to an IBM processor core in 130nm Cu11 technology with approximately 50k cell instances with  $V_{ddH} = 1.5V$  and  $V_{ddL} = 1.2V$ . Figure 9 shows the layout of this processor designed using generic voltage islands which resulted in 8% total power savings without any delay or area penalty.

#### 4. CLOCK SKEW SCHEDULING

In this section, we will discuss a technique known as clock skew scheduling, which in our experience with several industrial ASICs can be very useful in improving cycle time without much impact on power.

Typically high-performance circuit designers treat latches in their designs as sacred. Enormous amount of design effort is spent to implement a low skew clock tree. Traditionally, this has meant that clock signals cannot be intentionally skewed in order to balance non-critical and critical paths associated with a given latch in order to speed up critical paths. In our experience, in an automated flow such as the one for ASICs, clock skew scheduling can be effectively used to circumvent power-performance trade-off. It improves the power-performance product by improving the timing characteristics of a design with little or no impact to the logic circuits. The clock arrival time at a latch determines the capture and launch times of the data signal, therefore it controls the early and late mode slacks of the paths coming into and going out of the latch. A later clock arrival at a latch will improve (worsen) the late (early) mode slack of incoming paths and worsen (improve) the late (early) mode slack of the outgoing paths relative to the respective original slacks. An earlier clock arrival at a latch will have precisely the opposite effect. Instead of targeting zero clock skew, it is of interest to entertain the following optimization problem: Given a placed and routed design, find a set of clock arrival times at each latch such that the clock cycle time of the design and the number of late mode critical paths are minimized. This problem can be formulated mathematically and solved optimally using a linear programming [19], a binary search [20] or a parametric shortest path algorithm [21]. After obtaining such an “optimal” clock skew schedule, we have to build a clock tree which realizes the schedule. In the following we will describe our clock skew scheduling methodology and present some experimental results.

In the physical synthesis flow at IBM, timing closure is performed by PDS which resulted in a placed and optimized design. To extract additional cycle time improvement, clock skew scheduling is invoked as a function inside our internal timing tool, Einstimer. The user can specify the maximum skew allowed (e.g., 10 % of cycle time) and a skew schedule is produced at each latch which balances out the slacks of the paths to reduce clock period subject to the constraint that the early mode slacks below a threshold are not degraded. Instead of using exact algorithms to find the optimal schedule, we use a greedy heuristic. The clock arrival time at each latch is changed incrementally to improve the slack on one side of the latch at the expense of degrading the slack on the other side of the latch. At each step checks are performed to ensure that the amount of skew at the latch is within the specified range and that the critical early mode slacks are not degraded. This process is iterated until no improvement on the late mode slack can be made at any latch. The output at completion is a file containing the clock skew schedule.

The next stage is to build a clock tree according to this

**Table 2: Clock Skew Scheduling**

Design	Cycle Time Reduction	Critical Path Reduction
A	6.7%	456 → 19
B	2.3%	3309 → 417
C	32%	1819 → 1

schedule. Clock Designer, our clock tree generation tool, first builds a zero skew clock tree. It then clusters latches with similar skews and locations to be driven by the same clock splitter. Additional clock splitters will be inserted and some splitters will be cloned to achieve the required delay. The number of additional splitters required depends on how closely the clock skew schedule is followed. In our experiments, the allowed deviation is 25ps for the IBM’s SA27 0.18 $\mu$ m technology. The total cell area of splitters added is very small compared to the total logic area of the designs so the power impact is negligible. The results are shown in table 2 where we get up to 32% improvement in cycle time and a dramatic reduction in the number of critical timing end points.

#### 5. CASE STUDY

We applied some low-power techniques to a fixed-function real-time distributed DSP [22], in order to access their relative impact on the power-performance tradeoff. The original chip was about 2.3M gates. To implement this design, we selected IBM’s Cu11 130nm technology, and focused on the critical seven-macro subset consisting of Hilbert Transform, FIR filter (3), and FFT (3) macros. The subset requires about 240K logic gates and 42 KB of register array about 20% of the full design. The target frequency of the design was 177MHz. The goal of this study was to quantify the relative benefit of: electrical optimizations using fine grained libraries, voltage scaling and multiple thresholds; and major arithmetic and logic optimizations. Therefore, the design was first synthesized using IBM’s synthesis tool Boole-Dozer [23], and placed using Cplace (IBM’s placement tool). This gave us a baseline data point for the purpose of impact comparison of various design techniques. Detailed parasitics were extracted for steiner tree approximations of the routing. In the power calculations we looked at the clock tree power, the flip-flop data and clock power, the random logic cell power and the register array power. The power numbers can be found in the second column of table 3. Most of the power is used in the random logic. Using IBM behavioral synthesis tool Hiasynth, we applied several arithmetic optimization techniques such as balancing adder trees and using carry-save adder implementations. In addition we replaced the common adders with bitstack components built from regular Cu11 standard cells, that have a very compact implementation resulting in shorter wiring. In addition we applied placement driven optimizations in PDS during the placement. This resulted in an overall power savings of 46%, and a relative power performance gain of 2.8

**Table 3: Power Calculation (mW)**

	Baseline	ArithOpt	FG.Lib	VoltScale (1.1V)
Clock	18	14.8	12.5	10.5
FF-Data	14.1	14	11.7	9.8
FF-Clk	6	6	6	5
Logic	109.1	38.6	31.9	26.8
Array	12	12	12	10
Leakage	0.5	0.26	0.24	0.24
Total	159.7	85.66	74.34	62.34

We now look at the relative contributions of each of the

**Table 4: Power Performance (MHz/mW)**

	Base -line	Arith Opt	FG Lib	1.1V	1.0V	0.9V
Power	159.7	85.66	74.34	62.34	51	46
Performance	94	145	177	177	177	141
Pwr Savings %		46.36	53.45	60.96	68.07	71.20
Rel Pwr Perf	1	2.88	4.05	4.82	5.90	5.21

electrical optimizations. Applying a fine grained library [24], with significantly more gate sizes results in an additional 7% of power savings. Gain-based logic synthesis [25] can take advantage of this library and size the gates more precisely such that no nets are overdriven. Despite that the number of flip flops is not changed, less capacitive load gets reflected back to the flip flops and their data power is somewhat reduced. The placed area is smaller which leads to a smaller clock tree and less power consumption. An additional 7% is saved in power as can be seen in table 4, column four. The optimizations also have a positive effect on the performance of the design and the relative power-performance increases by a factor of 4 compared to the baseline design.

All results above were measured at 1.2V. PDS allows for optimization using voltage scaling. Lowering the voltage to 1.1V, has a positive effect on all components of the power as can be seen in column 5, where an additional 7.5% of power is saved. PDS was able to apply sufficient optimization to keep the performance of the design at the targeted 177Mhz. To study the effect of voltage scaling further, we reduced the voltage to 1.0V and further to 0.9V (the minimum voltage allowed in the Cu11 technology). By applying multi-Vth optimizations in PDS [16], we were able to keep the performance at 177 Mhz and reduced the power to 51mW. The total power savings more than offsets the increase in leakage power (leakage increases from 0.24mW to 2.3mW). Scaling the voltage to the minimum allowable supply voltage of 0.9V reduces the performance (to 141 Mhz) which cannot be recovered through multi-Vth optimizations. Since the re-sizing of low-Vth gates limited their use everywhere, the number of low-Vth gates did not increase much further. The total power reduced to 46mW. As can be seen from table 4, aggressive voltage scaling with multi-Vth optimizations to 1.0V Vdd provides the best power-performance trade-off.

## 6. CONCLUSIONS

In this paper, we explored the trade-off between multiple supply voltages and multiple threshold voltages in the optimization of dynamic and static power which can result in 60-65% power savings. Novel solutions to the unique physical and electrical challenges presented by multiple voltage schemes were proposed. We described a new single supply level converter that does not restrict the physical design. A power performance improvement of 5.9 was obtained by applying some of these optimization techniques to a hardware DSP test case. In this, electrical optimizations such as voltage scaling, multi-threshold optimization and the use of finer grained libraries enabled 2X improvement and the remaining 2.9X was enabled by high level arithmetic optimizations.

## 7. ACKNOWLEDGMENTS

The work on generic voltage islands was done in collaboration with Tony Correale, Doug Lamb and Dave Wallach. We thank Bill Scarpero, Bill Migatz, Paul Campbell for help with clock scheduling experiments and Subhrajit Bhattacharya, Lakshmi Reddy for help with PDS experiments.

## 8. REFERENCES

- [1] K. Usami and M. Horowitz, Clustered voltage scaling techniques for low-power Design, ISLPED 1995.
- [2] C. Chen, A. Srivastava, and M. Sarrafzadeh, On gate level power optimization using dual-supply voltages, IEEE Trans. on VLSI Systems, vol.9, p.616-629, Oct. 2001.
- [3] N. Rohrer, et al., A 480MHz RISC microprocessor in a 0.12 $\mu$ m Leff CMOS technology with copper interconnects, ISSCC 1998, p.240-241.
- [4] S. Sirichotiyakul, et al., Standby power minimization through simultaneous threshold voltage selection and circuit sizing, DAC 1999, p.436-441.
- [5] Q. Wang and S.Vrudhula, Algorithms for minimizing standby power in deep submicron, dual-Vt CMOS circuits, IEEE Transactions on CAD, vol.21, p.306-318, 2002.
- [6] M. Hamada, Y. Ootaguro, and T. Kuroda, Utilizing surplus timing for power reduction, CICC 2001, p.89-92.
- [7] K. Usami, et al., Automated Low-Power Technique Exploiting Multiple Supply Voltages Applied to a Media Processor, IEEE JSSC, Vol.33, No.3, 1998.
- [8] M. Hamada, et al., A top-down low power design technique using clustered voltage scaling with variable supply-voltage scheme, CICC 1998, p.495-498.
- [9] D. Sylvester and H. Kaul, Future performance challenges in nanometer design, DAC 2001, p.3-8.
- [10] A. Srivastava and D. Sylvester, Minimizing total power by simultaneous Vdd/Vth assignment, Proc. Asia-South Pacific DAC 2003, p.400-403.
- [11] C. Yeh, et al., Layout Techniques supporting the use of Dual Supply Voltages for Cell-based Designs, DAC 1999.
- [12] D. Lackey, et al., Managing Power and Performance for SOC Designs using voltage islands, ICCAD 2002.
- [13] S. Kosonocky, et al., Low Power Circuits and Technology for wireless digital Systems, IBM Journal of R&D, Vol. 47, No. 2/3, 2003.
- [14] A. Correale, D. Pan, D. Lamb, D. Wallach, D. Kung, R. Puri, Generic Voltage Island: CAD Flow and Design Experience, Austin Conference on Energy Efficient Design, March 2003 (IBM Research Report)
- [15] W. Donath, et al., Transformational Placement and Synthesis, DATE, 2000.
- [16] R. Puri, E. D'souza, L. Reddy, W. Scarpero, B. Wilson, Optimizing Power-Performance with Multi-Threshold Cu11 -Cu08 ASIC Libraries, Austin Conference on Energy Efficient Design, March 2003 (IBM Research Report).
- [17] R. Puri, D. Pan, D. Kung, A Flexible Design Approach for the Use of Dual Supply Voltages and Level Conversion for Low-Power ASIC Design, Austin Conference on Energy Efficient Design, March 2003 (IBM Research Report).
- [18] Y. Taur, CMOS Design near the limit of scaling, IBM Journal of R&D, Vol. 46, No. 2/3, 2002.
- [19] J. P. Fishburn, Clock Skew Optimization," IEEE Transactions on Computers C-39, pp 945-951, 1990.
- [20] T. G. Szymanski, Computing Optimal Clock Schedules," DAC 1992, p.399-404.
- [21] C. Albrecht, B. Korte, J. Schietke and J. Vygen, Cycle Time and Slack Optimization for VLSI-Chips, ICCAD 1999, p.232-238.
- [22] S. Bhattacharya, J. Cohn, R. Puri, L. Stok and D. Sunderland, Power reduction of Hardwired DSPs in standard ASIC methodology, Submitted to CICC, 2003.
- [23] L. Stok, et al., BooleDozer Logic Synthesis for ASICs, IBM Journal of R&D, Volume 40, no. 3/4, 1996.
- [24] F. Beeffink, P. Kudva, D. Kung, R. Puri, L. Stok, Combinatorial cell design for CMOS libraries, Integration: the VLSI Journal, V.29, p.67, 2000
- [25] P. Kudva, D. Kung, R. Puri, L. Stok, Gain based Synthesis, ICCAD Tutorial, 2000.