

High Performance DSPs – What's Hot and What's Not?

Bryan Ackland
Lucent Technologies
Holmdel NJ 07733
+1 (732) 949-7248
bda@lucent.com

Chris Nicol
Lucent Technologies
Holmdel NJ 07733
+1 (732) 949-3024
chrisn@lucent.com

1. ABSTRACT

This paper compares low power techniques current in the research literature with those used in commercial DSP design and explores why some techniques have not yet had commercial impact. It also examines the low power needs of future DSP applications.

1.1 Keywords

DSP, low power, architecture, circuit design.

2. INTRODUCTION

When the general purpose programmable DSP was first introduced in 1980, it offered a 50:1 multiply accumulate (MAC) performance advantage over the microprocessor as shown in Figure 1. Over the intervening years, the microprocessor has seen huge gains in performance through a combination of architectural innovation and process improvement. By comparison, the DSP has seen only modest performance improvement so that today's microprocessors outperform DSPs, even in MAC dominated applications.

And yet DSPs generate over \$3 billion of revenue for the semiconductor industry each year. This is because the DSP has been able to redefine itself as a low cost, low power signal processing engine whose MOP/mm² and MOP/mW ratings are today an order of magnitude better than its high power microprocessor competitors as shown in Figure 2. Low cost and low power, combined with the flexibility and time-to-market that come from a programmable solution, have made the DSP the implementation of choice in a number of large consumer markets including voice band modems, cellular terminals, speakerphone/answering machines and automotive.

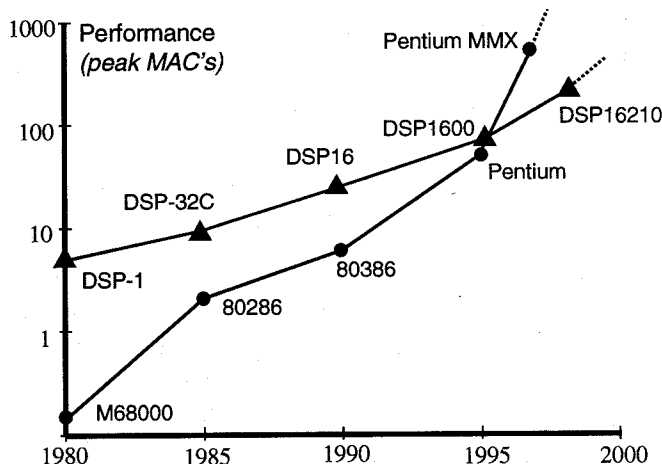


Figure 1. Performance of DSPs vs. Microprocessors

DSPs have achieved this area and power efficiency through a combination of architectural and circuit level design trade-offs. Yet, of the many low power circuit and methodology improvements proposed in the literature over the past decade, only a few have made it into mainstream DSP design. This paper reviews techniques used in today's low power DSPs and attempts to understand why some promising approaches have not yet had significant commercial impact. It also looks forward to next generation DSP applications and how designers might tackle the seemingly conflicting constraints of performance, power, programmability and cost.

3. LOW POWER DESIGN TECHNIQUES

We review some promising power reduction techniques in the literature and relate them to DSP applications.

3.1 Architectural and Algorithmic

When implementing a system on a programmable DSP, the algorithmic optimizations that result in a reduction in power consumption are made primarily in software. In [1], a study of embedded processor code demonstrates that keeping data on-chip (in memory banks and register files) minimizes the power consumption. Fortunately, optimizing compilers seek to do the same (to save cycles) and are therefore likely to produce a near-optimal result. Significant power savings (20% of chip power) are possible by having the compiler (or the user) control the manner in which operands are fed to the MAC [2].

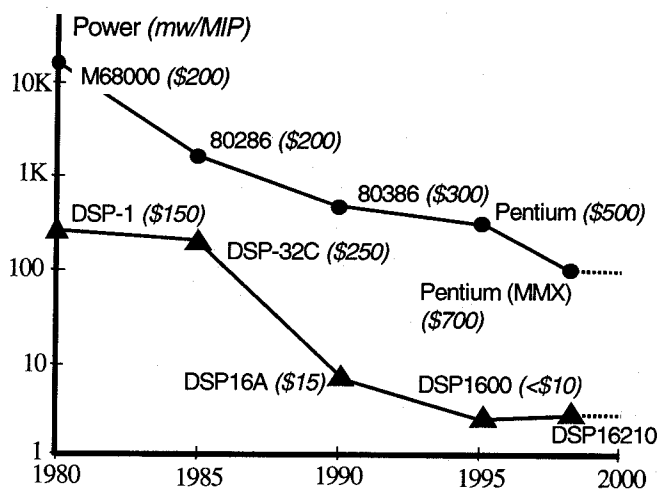


Figure 2. Power & Cost of DSPs vs. Microprocessors

At the architectural level, programmable DSPs usually contain special addressing modes whose purpose is to save execution cycles. Examples are circular buffer addressing modes that enable a smooth-running FIR filter to be implemented in a very tight loop. Clever use of these modes can have a huge impact on performance and power. Adaptive coefficient scaling [3] can also be applied to programmable DSPs to reduce power in adaptive filters.

3.2 Arithmetic Modules

Multipliers and adders (MACs) are the basic building blocks of DSPs. Historically array multipliers have been used in DSPs because the layout maps to the bit-sliced design of a full custom data path. However, Wallace tree multipliers are arguably more energy efficient because of the parallel summation of the partial products (PPs). Results in [4], show that the power savings due to reduced switching more than compensate for the additional interconnect capacitance of the tree, especially for longer word widths (16-32 bits). The tree architecture is also known to be faster and therefore more suitable for high performance DSP chips. In [5], the Booth re-coding of FIR filter multipliers is optimized for typical filter responses. The same techniques can be used to minimize power in FIR filters on programmable DSPs providing the filter coefficients are applied to the Booth re-coded input.

Delayed Booth re-coding, to eliminate spurious switching in array multipliers is addressed in [6]. Approaches for tree multipliers include the leap-frog multiplier [7], tri-state driver insertion into fast paths to minimize the propagation of glitches [8], the re-synchronization of PPs using a delayed clock [9] and the balancing of the signal paths from both inputs to the PPs [10].

A comparison of adder designs from a power perspective can be found in [11]. It turns out that the slowest architecture (the ripple carry) is also the most efficient. The design of a fast, power-efficient adder for DSPs is

still an open challenge. The Binary Look-ahead Carry adder [12], used in the NEC V830 [13], is a high performance adder that uses conventional complimentary static logic is therefore robust at low voltages.

3.3 SRAM

The survey in [14] gives an excellent overview of techniques used to minimize the power consumption in SRAMs. Notable is the dividing of the RAM into banks, using bit-line isolation sense-amps, hierarchical word line decoding, low swing data busses, address driven clock and control signal activation. Instruction buffers [15] reduce the need to activate SRAMs for every instruction fetch.

3.4 Logic Families

In the general purpose (micro)processor (GPP), where performance is the driving goal, dynamic logic families are normally chosen because they are considered the fastest. For DSPs however, minimizing power is more important and the high activity ratio of dynamic logic families more than offsets the reduced input capacitance – especially considering the correlation found in signal data streams.

CPL and DPL [16] are examples of logic families designed for high performance, low power operation. Logic styles like CPL, that generate threshold voltage drops at any point in the circuit, are dangerous for low voltage operation - even when level restoration logic is used. DPL is claimed to be fast and energy efficient [17]. The problems with DPL arise only when layout issues are considered. Our own experiments concluded that the differential nature of DPL and the limited ability to exploit source-drain sharing in the layout resulted in DPL being some what less efficient than complimentary CMOS. This conclusion is supported by the results in [18]. So, uninterestingly enough, static complimentary CMOS remains the low power logic of choice.

3.4.1 Mixed-Voltage logic families,

A new breed of logic families is emerging for future low power systems. Quad-rail logic [19] keeps the logic fast (at full voltage swing) while reducing power by limiting the swing on inter-module interconnects. Another approach [20] partitions the logic so that critical paths are operated at VDDH; others at VDDL with level converters at VDDH/VDDL interfaces. Critical issues are noise immunity, yield, CAD tool support and the efficient generation and distribution of multiple power rails.

3.5 Flip-flops

Pipelining is used extensively throughout DSP data paths to improve performance and facilitate voltage scaling. As a result, flip-flops (and their associated clock network) contribute up to 30-40% of the total power consumption. The study presented in [21] focuses on reducing the power-delay product. Simple techniques such as weak static feedback are shown to be less efficient because of

the extra energy needed to “flip” the latch. The circuit in [22] was used in a one volt, low power DSP and contains only 2 clock devices per latch. The edge-triggered latch used in the *StrongARM* processor [23] contains only 3 clock FETs – it functions with low-swing clock signals for a further reduction in clock network power [24].

3.6 CAD Tools

Synthesis tools that are capable of minimizing the power consumption of a circuit are still in early stages of development. Estimating a circuit’s power consumption via the probability of a transition and propagating the probability through the circuit eliminates large amounts of simulation but requires the user to enter the probability of transitions at the inputs.

Most power-critical circuits today are designed with the aid of *PowerMill* which adds power reporting capabilities to a timing (circuit) simulator. The *HEAT* tool [25] gives *SPICE*-accurate power estimates without simulation. This tool characterizes cells using *SPICE* and extrapolates the power consumption based on the fan-out of each cell.

3.7 Technology and Voltage Scaling

The concept of dynamic voltage scaling is possibly the most revolutionary low power technique to date. In [26] the level of an input data FIFO is used to control the supply voltage of an asynchronous DSP using an on-chip DC-DC converter. This work has been extended to synchronous DSPs [27].

Two problems with operating a circuit at reduced voltage are reduced speed and increased static current. The static current, normally ignored in CMOS design, becomes increasingly significant at low supply voltage. Multi-threshold technologies [28] enable logic to use low V_{th} transistors for speed, while high V_{th} transistors are used to “cut-off” the power supply when the chip is placed into sleep mode. Another approach is to adjust the threshold voltage dynamically via the bulk bias voltage [29].

Voltage scaling enables the concept of “voltage scheduling” [30]. With the knowledge of the load requirements of a system in advance, the operating system can schedule tasks and processor supply voltage in order to minimize power while still meeting task deadlines.

4. COMMERCIAL DSP DESIGN

Lucent has developed over the years a number of low power DSP cores for a wide variety of applications. We examined the techniques used to deliver this low power performance for three representative designs:

- 16210 is a *high performance* dual-MAC 16-bit DSP for baseband processing in wireless base-stations and modem pool applications
- 1609 is a *low cost* 16-bit DSP for digital cordless phones, answering machines, speakerphones and other consumer communication applications

- CPP (Communications Protocol Processor) is a *low power* 16/32 bit RISC core used as a microcontroller in cellular terminal applications

Each has achieved an impressive mix of performance, power and cost as summarized in Figure 3. The design techniques used by these designers, however, include only a small subset of the research topics listed in Section 3. In fact, much of the power reduction achieved comes from a few straightforward techniques that could simply be considered good design practice.

DSP	MIPS	MAC/S	CMOS	Active	Sleep
16210	100	200	0.35 μ	325mW	5.4mW
1609	80	80	0.3 μ	265mW	500 μ W
CPP	50	-	0.35 μ	93 mW	75 μ W

Figure 3. Low Power DSP Specifications at 3.3V

4.1 Silicon Process and Physical Layout

At the risk of stating the obvious, the best strategy is to have access to a high speed, low voltage process with low intrinsic parasitics. Each of the design teams used full custom layout in their datapaths. The primary reason was not area or performance, but power reduction through careful control of circuit topology, transistor size and local parasitics.

4.2 Logic and Registers

Complimentary CMOS is used throughout these designs - dynamic logic is relegated a very few speed-critical circuits. Static flip-flops are always used in pipelines so that they can be readily stalled. Flip-flops with weak static feedback are used to minimize clock loading.

NMOS only transmission gates are used to build multiplexers. These are followed by buffers which have high threshold pFETs to prevent static current. This increases the wafer cost by less than 5%.

4.3 Memories and Busses

DSPs often rely on single cycle SRAM to guarantee real-time performance. This precludes the use of off-chip memory (for power reasons) and cache hierarchies (because of latency). But even on the same die, large fast SRAMs can be very power hungry and so designers use a number of simple techniques to reduce dissipation.

Hierarchical memory sub-banking along with clocks gated by the output of the address decoder limit activity to a small portion of the overall memory space. Address and data busses are similarly structured to reduce the switched capacitance per access. Bit-line isolation sense amps limit the voltage swing on bit lines to the minimum necessary to ensure a reliable sense operation.

Bus keepers are routinely used to prevent drifting bus lines from drawing DC current through input buffers.

Unidirectional busses are often used to trade area for power by limiting bus transitions and also the overhead of enabling bus drivers.

In one important application, designers noted that there were many more zeros than ones stored in an on-chip data memory. A single bit line memory design with inverted data reduced overall power dissipation by 10%.

4.4 Power Control

Clock gating is used extensively to limit data transitions and clock dissipation to those portions of the processor and those peripherals that are active. The registers in a multiplier, for example, will only be clocked when a multiply instruction is issued. In CPP, external memory wait states cause the processor to shut down with all clocks disabled until the memory request has been satisfied. A power control register allows for explicit programmer control over a number of different "sleep" states. PLLs generate clock frequencies according to the state of the machine. In very low power stand-by applications a *wait for interrupt* instruction shuts down all clock activity, using combinational logic to test the interrupt and re-start the machine.

4.5 Instruction Set Architecture

In high-speed designs, RISC architectures have been successful because of cycle time reduction. In low power applications, the trend is towards more complex instructions to improve code density and minimize the number of instruction and data fetches. Duplicate register files minimize the power overhead of context switches. In the CPP, a variable cycle multiplier eliminates the unnecessary calculation of sign extension bits when dealing with small operands. In the left-to-right multiplier [15] used in the 16210 the carry rippling moves from most-significant PPs to the least significant PPs. For FIR filters this can result in power savings because most of the activity is in the least significant PPs.

4.6 CAD Tools

Relatively primitive power analysis tools were used in these designs. At the architectural level, Cmodel simulations were used to estimate transition counts on critical signals. Circuit simulators such as *PowerMill* gave designers more detailed feedback on the effectiveness of their designs. No power driven synthesis was used.

In each case, the wish-list of power reduction techniques discussed at the beginning of the project was much longer than the set actually used. In the end, time to market and design cost considerations focused attention on techniques that had the most impact with the least disruption to the team's design style.

5. WHAT DIDN'T MAKE IT & WHY?

A number of low power techniques which have featured prominently in the research literature, have not found their way into commercial DSP design. In looking at why these

techniques have not yet had significant commercial impact, at least three reasons emerge.

5.1 Incomplete Solutions

The year was 1993 and a hot topic at Bell Labs was adiabatic logic [31]. Business units, attracted to the promises of 10-times reduction in power, supported a research effort to explore these new techniques. However it soon became clear that adiabatic logic suffered a number of very real implementation problems. Because of the latching property of each gate and associated synchronization problems, it was near impossible to integrate into existing product design flows. In addition it required a clock generator that could deliver high powered multi-phase clock waveforms to a data-dependent clock-load with greater than 95% efficiency. While it remains an interesting problem that continues to be investigated in a number of research laboratories, adiabatic logic will continue to be just an "interesting problem" until a complete solution comprising logic, latches, clock and design methodology can be found.

5.2 Inadequate Commercial CAD Support

Some of the techniques that featured prominently on our designers wish list but never made it to product involved reducing power in non-critical paths by logic restructuring, transistor sizing or through the use of multiple supply voltages. Each of these techniques has been shown to provide significant power reduction in the research literature. But with semiconductor houses relying less on in-house CAD tools and more on commercial tools suites, these techniques will not be adopted until they have mainstream vendor support.

5.3 Insufficient Benefit Solutions

When it became clear that asynchronous logic was never going to outperform synchronous logic, it was re-targeted at the low power community. While there are examples of asynchronous circuits that operate at very low power compared to their synchronous counterparts, designers have opted for the simpler technique of using synchronous logic with gated clocks. Commercial CAD tools do not support the synthesis and verification of asynchronous circuits and manual design is fraught with danger. Asynchronous design is important for those relatively small portions of a circuit that cross clock domains. But, for the bulk of the DSP design process, the benefits do not justify the disruption to the design flow.

A question we asked of our own designers is: "*How much power do I need to save, for you to change the way you do things and adopt a new design flow?*" Their answer: a reduction in power of 4-5 times. Designers do not want to change the way they "think" about their designs. For smaller power gains they will happily change their scripts to target new libraries, adopt new tools and are certainly open to new algorithmic and architectural techniques,

providing these do not significantly alter the existing design methodology.

Another issue is the *level of confidence that a design will work correctly*. Often, product managers who are under extreme time-to-market pressure abandon new power reduction techniques. A fully functional one Watt chip in a customer's prototype board (on schedule) is preferable to a partially functional 200mW chip on an E-beam prober six months later. By the time the chip reaches production, the difference in power may well have been made up moving to a new technology.

6. NEXT GENERATION DSPs

The expanding and converging fields of computing and digital communications are creating new demands for high performance, programmable signal processing engines including:

- *Embedded Applications* such as cable modems, set-top box, digital audio broadcast and smart phones.
- *PC based Applications* such as 3-D graphics, DSL modems and real-time video communication
- *Infrastructure Applications* such as modem pools, cable modem head-end, and wireless basestation.

The performance requirements of these applications are significantly beyond the capabilities of today's DSP. Minor enhancements to the architecture in combination with process improvement will not bridge the gap in time. What is needed is a class of architectures that provide:

- Very high levels of DSP integer (and in some cases floating point) performance ranging from hundreds of MOPS to tens of GOPs.
- Large memory and I/O bandwidth
- Support for complex, multithreaded, real-time synchronous applications
- A programmer friendly, compiler driven programming environment with support for parallel programming and debugging.
- Extensibility to meet a wide range of cost/performance/power constraints.

How can these seemingly conflicting constraints be resolved? High performance implies high power. Programmability is also expensive in terms of power dissipation (at least an order of magnitude over dedicated hardware solutions). Architectures that are good compiler targets (e.g. RISC, VLIW) tend to be inefficient in their use of memory - further increasing power consumption.

7. HIGH PERFORMANCE & LOW POWER

A key differentiating factor for many of these new applications is that power reduction becomes important as a mean of reducing cost by enabling a cheaper package and reducing power supply and cooling requirements rather than as a means of increasing battery life. Peak

currents and worst case power dissipation become as important as average power dissipation.

A circuit is designed to achieve a required level of performance under worst case conditions at a minimum voltage. However, power is usually quoted at the maximum voltage. Increasing the supply voltage above the minimum level does nothing other than increase power dissipation. This suggests some form of dynamic voltage scaling, allowing chips to be operated at less than nominal voltage. Alternatively, voltage scaling allows chips to be designed for nominal conditions, thereby relaxing performance constraints.

Parallel architectures will become increasingly important in developing next generation DSPs. Parallel execution is the only way to achieve the performance levels demanded by these new applications. Fortunately, parallel architectures also provide a means of reducing overall power dissipation by trading chip area (parallel data paths) for reduced clock frequency and supply voltage.

The research literature reveals no shortage of parallel architectures including VLIW, RISC+SIMD, GPP+MMX and MIMD. While VLIW presents an attractive target for a DSP compiler, its power efficiency is compromised by poor code density and its scalability is limited. GPP+MMX and RISC+SIMD also provide limited scalability for most applications. MIMD provides task level parallelism (which is readily available in many of these applications) but fails to exploit the instruction level (data) parallelism that is found in many DSP tasks. MIMD in combination with VLIW or SIMD at the processor level would seem to provide the greatest potential for high order parallel execution [32]. It also, however, provides the greatest challenge to those developing DSP programming environments.

The combination of MIMD task parallel architectures and dynamic voltage scaling have led us to consider how one might use a real-time parallel operating system to dynamically manage power in a high-end DSP.

7.1 Voltage Scheduling on a MIMD DSP¹

An embedded RTOS supporting a dynamic scheduling algorithm (like Earliest-Deadline-First EDF) can use the task requirements to compute the required operating frequency of the processing elements (PEs). This in turn is used to determine the supply voltage. Ideally, the set of active tasks will be distributed evenly across the PEs because the clock frequency is set by the most heavily loaded PE. One way to overcome this is to have separate supply voltages for each PE. The communication network that connects the PEs must then support the communication between PEs operating at different frequencies and outputting signals at different levels. An

¹ This section describes ongoing research at Bell Labs by the Authors, K.J. Singh and A. Kalavade.

asynchronous bus with FIFOs and level shifters can support reliable communication on such a system.

8. CONCLUSIONS

Although commercial DSPs rely heavily on low power operation to define their place in the market, DSP designers have been slow to adopt low power design techniques as described in the research literature. Tomorrow's applications will require a new breed of high-performance yet power and cost-sensitive architectures. Next generation DSP designers will be looking to the research community to help squeeze out every milliamp. The ideas that will be quickly incorporated into these designs will be those that are readily supported by commercial CAD tools, that can be reliably verified before fabrication and that do not cause the designer to significantly "rethink" the design process.

MIMD-DSP can potentially provide the scalability for future DSP applications as well as being an attractive target for voltage scheduling to help maintain the mW/MIP and \$/MIP advantage that DSPs enjoy today. The challenge will be to provide the software development environment that will allow applications programmers to quickly generate and debug parallel, real-time code on these architectures.

9. ACKNOWLEDGEMENTS

The authors would like to thank Jill Bennet, John Fernando, Bill Griesbach, Trevor Little, Mark Luong, Brian Petryna, Bob Scavuzzo and Andrew Wang for their insights into commercial DSP design practice.

10. REFERENCES

- [1] V. Tiwari, S. Malik and A. Wolfe, "Power analysis of embedded software: A first step towards software power minimization", *TVLSI*, pp. 437-445, Dec. 1994.
- [2] H. Kojima, A. Shridhar, "Interlaced Accumulation Programming for Low Power DSP", *ISLPED*, 1996.
- [3] P. Larsson, C.J. Nicol, "Self-Adjusting Bit-Precision for Low-Power Digital Filters", *Symp. VLSI Circ.*, 1977.
- [4] P. Meier, R. Rutenbar, L. Carley, "Exploring Multiplier Arch. and Layout for Low Power", *CICC*, pp. 513-516, 1997.
- [5] C.J. Nicol, P. Larsson, "Low Power Multiplication for FIR Filtering", *ISLPED*, pp. 76-79, 1997.
- [6] T. Sakuta, W. Lee, P.T. Balsara, "Delay Balanced Multipliers for Low Power/Low Voltage DSP Core", *SLPE*, pp. 36-37, 1995.
- [7] S.S. Mehant-Shetti, C. Lemonds, P. Balsara, "Leap-Frog Multiplier", *ISLPED*, pp. 221-223, 1996.
- [8] J. Goodman, A. Chandrakasan, "A 1Mbs Energy /Security Scalable Encryption Processor using Adaptive Width and Supply", *ISSCC*, pp. 110-111, 1998.
- [9] E. Iwata *et-al.*, "A 2.2 GOPS Video DSP with 2-RISC MIMD, 6-PE SIMD Arch. for Real-Time MPEG2 Video Coding/Decoding", *ISSCC*, pp. 258-259, 1997.

- [10] R. Fried, "Minimizing energy Dissipation in High Speed Multipliers", *ISLPED*, pp. 214-219, 1997.
- [11] T.K. Callaway, E.E. Swartzlander Jr., "Low Power Arithmetic Components", *Low Power Design Methodologies*, Kluwer Academic, Ch. 7, 1996.
- [12] R. Brent, H. Kung, "A Regular Layout for Parallel Adders", *IEEE Trans. Comp*, C-31 (3), pp 260-264, 1982.
- [13] K. Nadehara *et-al.*, "A Low-Power, 32-bit RISC Processor with Signal Processing Capability and its Multiply-Adder", *VLSI Signal Processing*, 1995.
- [14] K. Itoh, *et-al.*, "Trends in Low-Power RAM Circuit Technologies", *Proc. IEEE*, V83, N4, pp. 524-543, 1995.
- [15] M. Kamble, K.Ghose, "Analytical Energy Dissipation Models for Low Power Caches", *ISLPED*, 1997.
- [16] M. Suzuki *et-al.*, "A 1.5ns 32-b CMOS ALU in Double Pass-Transistor Logic", *JSSC*, V28, N11, 1993.
- [17] U. Ko *et-al.*, "Low-Power Design Techniques for High-Performance CMOS Adders", *TVLSI*, V3, N2, 1995.
- [18] R. Zimmermann and R. Gupta, "Low Power Logic Styles: CMOS vs. CPL", *ESSCIRC*, 1996.
- [19] R. Carley, I. Lys, "Quadrail: A Design Methodology for Ultra Low Power Integrated Circuits", *Proc. Napa Valley Workshop on Low Power IC design*, April 1994.
- [20] M. Igarashi *et-al.*, "A Low-Power Design Method Using Multiple Supply Voltages", *ISLPED*, 1997.
- [21] U. Ko, P.T. Balsara, "High Performance, Energy Efficient Master-Slave Flip-Flop Circuits", *SLPE*, 1995.
- [22] W. Lee *et-al.*, "A 1V DSP for Wireless Comms", *ISSCC (Paper 6.1 in Slide Supplement)*, 1997.
- [23] D. Dobberpuhl, "The Design of a High Performance Low Power Microprocessor", *ISLPED*, pp. 11-16, 1996.
- [24] H. Kawaguchi, T. Sakurai, "A Reduced Clock-Swing Flip-Flop (RCSFF) for 63% Clock Power Reduction", *Symp. VLSI Circ.*, pp. 97-98, 1997.
- [25] J. Satyanarayana, K. Parhi, "HEAT: Hierarchical Energy Analysis Tool", *DAC*, pp. 9-14, 1996.
- [26] L. Nielsen *et-al.*, "Low-Power Operation Using Self-Timed Circuits and Adaptive Voltage Scaling of the Supply Voltage", *IEEE TVLSI*, V2, N4, 1994.
- [27] R. Amirtharajah, A. Chandrakasan, "Self-Powered Low Power Signal Processing", *Symp. VLSI Circ.*, 1997.
- [28] S. Mutoh *et-al.*, "1-V Power Supply High-Speed Digital Circuit Technology with Multi-threshold Voltage CMOS", *JSSC*, V30, N8, pp. 847-854, Aug. 1995.
- [29] T. Sakurai *et-al.*, "Low-Power CMOS Design through Vth Control and Low-Swing Circuits", *ISLPED*, 1998.
- [30] M. Weiser *et-al.*, "Scheduling for Reduced CPU Energy", *USENIX 1st Symp. on Operating System Design and Implementation*, Monterey, pp. 13-23, Nov. 1994.
- [31] J. Denker, "Adiabatic Logic", *Proc. Napa Valley Workshop on Low Power IC Design*, April 1994.
- [32] H. Igura *et-al.*, "An 800MOPS 110mW 1.5V Parallel DSP for Mobile Multimedia Processing", *ISSCC*, 1998.