CICC '93 Preprint (IEEE Custom Integrated Circuits Conference, San Diego, May 9–12)
comments to Lyon@Apple.com or (408) 974-1569

COST, POWER, AND PARALLELISM IN SPEECH SIGNAL PROCESSING

Richard F. Lyon

Apple Computer
One Infinite Loop
Cupertino, CA 95014

ABSTRACT

The cost of speech signal processing and other computationally intensive functions is increasingly influenced by power consumption as products are made smaller and more portable. That is, in portable products, weight and battery life are bigger issues than silicon area and total computational capability. A recent emphasis on the power problem within the VLSI signal processing community has led to an understanding of how parallelism can significantly reduce the cost of a system by greatly reducing clock speed, supply voltage, and power consumption, even though at the expense of silicon area and other measures of efficiency. Several different kinds and degrees of parallelism, including massive analog parallelism, should be considered in planning to reduce the total cost of speech signal processors. In this tutorial paper, recent and older ideas are reviewed with respect to their potential applicability to modern products, as well as with respect to their difficulties.

1. INTRODUCTION AND MOTIVATION

Speech processing and other signal processing systems have enough regularity and inherent parallelism that they are amenable to a wide range of optimization techniques. The VLSI academic and industrial communities have traditionally focussed much more emphasis on optimization of silicon area and clock speed than of power consumption, though with portable products becoming popular this trend is changing quickly. Analytical optimization techniques sometimes lead to general rules of thumb; for example, designers know to break up amplifiers and drivers into stages with gains not much above *e* to optimize delay [1]. But when the rules of thumb come from the optimization of a single variable, they will be at odds with the optimization of a more relevant joint cost function. Since system-level cost functions have been difficult to define at the chip design stage, and since good joint optimization tools have been difficult to get and to use, chip designers may still be applying rules of thumb that lead to very power-inefficient, and hence costly, product designs.

Historically, we have seen how applying huge numbers of transistors to a problem can reduce the cost of a system (as in the use of a DSP chip instead of a few inductors and capacitors to implement a filter). In the age of portable products, we should think in terms of applying huge numbers of transistors to reduce the size and weight of the battery. The examples surveyed in this tutorial support the idea of using parallelism of slow processing units as a power-saving alternative to the single fast central processor that characterizes many of today's desktop and portable products.

1.1. Power as a Primary Cost Factor

In today's desktop computers, the silicon chips and associated components tend to cost more than the power supply and cooling systems. While this may also be true in portable computing products, the weight and limited operating time of the portable power source implies another kind of cost to the user. The option of increasing the energy density by using more exotic batteries is available, but at a very high cost. In new generation RISC-based desktop systems, getting rid of the heat is an increasingly costly problem. Overall, it seems that electrical energy is very cheap, but storing and transporting it, and getting rid of its waste heat, are increasingly expensive in more compact and higher-performance products. Mead has argued that computer system cost has stayed proportional to power consumption over many generations of machines, and that we should expect this trend to continue [2].

In spite of this situation, we do not have good institutionalized ways of estimating and trading off the true costs associated with the power consumption of our computational subsystems. In this tutorial, we emphasize power as the primary cost factor that needs more explicit

consideration at all levels of design, in hopes of raising the level of attention paid to power by chip and product designers.

To make things worse, other kinds of hard-to-evalulate costs are often involved in system optimization. For example, Narkiewicz and Burleson [3] present techniques that "allow tradeoffs between VLSI costs, performance and precision," while Orailoglu and Karri [4] "systematically explore the three-dimensional design space spanned by cost, performance, and fault-tolerance constraints." Apparently performance, precision, and fault-tolerance are important dimensions that, like power, haven't yet been integrated into "system cost functions."

1.2. Portable Speech Processors

Portable sound and speech products tend to have severe real-time computational requirements relative to other portable products, and hence a bigger problem with battery weight and running time. We need to develop a bag of tricks to attack this general problem at all levels.

An important property of speech processors is that they have a bursty load. Voice coders, recognizers, synthesizers, recorders, etc. only need to process speech when speech is present. An important source of power efficiency in such systems is the ability to shut down or slow down parts of the system when their capabilities are not needed. A less obvious power saving technique is to move some of the work from the fast real-time mode into a slower and more efficient background mode, for example in a voice-mail compressor. In section 2 we explain how a slow mode not only reduces the power level, but can also reduce the total energy consumption.

1.3. "Are DSP Chips Obsolete?"

In a recent paper, Stewart, Payne, and Levergood [5] posed the question of what a specialized DSP chip is good for in the age of fast RISC chips. Using the DEC Alpha architecture [6] as an example, they argue that it is generally preferred to let a fast RISC chip do the kinds of computational jobs, such as speech processing, that have been largely relegated to DSP chips during the last decade. Unfortunately, they totally ignore power as a cost factor, and propose a solution that is about an order of magnitude more energy hungry than the DSP alternative. We discuss how optimizing a chip for speed leads to its inefficient use of energy, again using the DEC Alpha RISC chip as an example.

1.4. Previous Surveys

There are several previous excellent surveys of power-saving techniques and the size-cost-speed-voltage-power trends and limits of CMOS technology (mostly from an academic perspective). An early study by Swanson and Meindl (Stanford) [7] explores CMOS logic structures at very low supply voltages. Mead and Conway (Caltech and Xerox) [8] provide in-depth chapters on the "Physics of Computational Systems", including technology scaling and energetics of very low voltage CMOS, as well as an introduction to the kinds of "Highly Concurrent Systems" architectures that can take advantage of low-speed low-power operation (with contributions from Kung and Leiserson of CMU and Browning and Rem of Caltech). Vittoz (Swiss CSEM and EPFL) describes "Micropower" techniques for portable electronic devices [9]. Mead (Caltech) [2] emphasizes power costs and the power efficiency of special-purpose and analog signal processing implementations in "Neuromorphic" systems. Burr, Williamson, and Peterson (Stanford) [10] survey low-power issues in modern high-performance signal processing systems, emphasizing current projects at Stanford. Brodersen, Chandrakasan, and Sheng (Berkeley) [11] provide an excellent tutorial with specific examples of choices at many levels that can improve power efficiency in signal processors.

In the present tutorial, rather than introducing new ideas, we attempt to assemble ideas to support an attitude adjustment of design engineers, emphasizing the conclusions of Mead and Conway [8] over a decade ago, that "In real systems, the cost of power, cooling, and electrical bypassing often exceeds the cost of the chips themselves. Hence any discussion of the cost of computation must include the energy cost of individual steps of the computation process." and "Perhaps the greatest challenge that VLSI presents to computer science is that of developing a theory of computation that accommodates a more general model of the costs involved in computing."

2. POWER CONSUMPTION BASICS

Power consumption in modern CMOS circuits is usually dominated by dynamic power related to the charging and discharging of circuit nodes between the two power supply voltage levels that represent logic 1 and logic 0. Dynamic power can be expressed as:

$$P = fCV^2$$

where f is the clock frequency, C is the effective total capacitance being switched at the rate of one transition per clock cycle, and V is the supply voltage.

The effective total capacitance is computed as a weighted sum of all the node capacitances in the circuit, with each node's weight equal to its number of logic transitions per clock cycle, averaged over the conditions of interest. Clock nodes have two transitions per cycle, so they count double, while static logic nodes may average one-half transition per cycle or less; precharged logic nodes are typically somewhere in between.

2.1. Reducing Power

It might almost go without saying that power can be reduced in three ways:

- Reduce Frequency
- Reduce Capacitance
- Reduce Voltage

Importantly, the voltage factor applies twice, so any reduction in voltage is worth twice as much power savings as a proportionate reduction in frequency or capacitance. But the supply voltage may also be the most difficult parameter for the designer to change freely, due to system and compatibility constraints. Typically reducing the supply voltage will also require reducing the clock frequency, so other changes will be needed to maintain performance.

Reductions in effective switched capacitance can come from two main sources: physical-level and circuit-level optimizations that reduce node capacitance per logic gate, and logical/architectural reorganizations that reduce the number of nodes or gates being switched. Avoiding clock and data transitions in portions of the processor that are not needed for a particular part of the computation is one technique that pays off well. Making low-level "sizing" optimization criteria favor power rather than speed will result in smaller transistors, smaller cell layouts, and shorter wires, all contributing to lower capacitance.

Reductions in clock frequency are generally acceptable only if there is some other change that allows the system computational requirements to be met. For example, the average clock frequency can be reduced if the clock frequency is made variable and the computational requirements are variable. Or the clock frequency can be cut by a factor of N if N copies of the processor are used and can efficiently share the computational load. In this case, the silicon area and capacitance increase by a factor of N, but the lower clock rate will allow a lower supply voltage and a great power savings. This kind of trading of frequency and voltage against area and parallelism is a large under-exploited source of power savings.

CMOS circuits also dissipate power during switching due to the fact that the transistor charging a node doesn't quite turn off before the other transistor tries to discharge the node. This "crossover" or "short-circuit" power is again proportional to the effective switching frequency and to a "strength times slowness" factor that scales pretty much like the capacitance; but this power is also a very expansive function of supply voltage (between quadratic and exponential). If the supply voltage is reduced to around twice the typical transistor threshold, the crossover power involves only subthreshold conduction, and is largely negligible (as long as threshold magnitudes are large compared to kT/q); but for 3-V and 5-V chips it may be a significant power as well as a significant contributor to power supply noise spikes. DC power is also significant if threshold voltages are reduced to the point where the turned-off transistor has a substantial leakage. Burr and Peterson [12] analyze the contribution of crossover power and DC power at very low threshold and supply voltages, where they are important considerations in establishing limits to ultra low power operation.

2.2. Circuit Speed versus Voltage

Operating a system at a supply voltage higher than the clock frequency requires is a big waste of power. The system designer should think about scaling supply voltage and system clock frequency together for best efficiency. Using self-timed circuits or voltage-dependent clock oscillators, it is possible to make the system speed adapt to conditions such as dropping battery voltage, changing temperature, etc. For example, the technique of Von Kaenel *et al*. [13] can be used to drop the supply voltage to match the logic speed, or can be turned around to adjust the logic speed to match the supply voltage.

The designer needs to know what to expect in the speed versus voltage tradeoff even when the system handles it automatically. The expected scaling between voltage and speed depends on what voltage range is considered; figure 1 characterizes some possibilities.

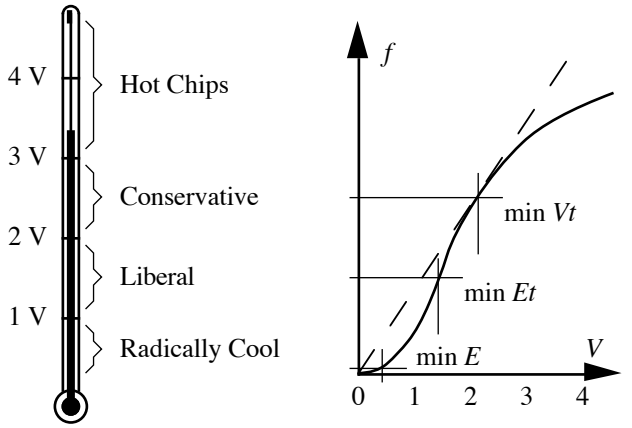

Figure 1. A crude characterization of power supply voltage ranges available to the chip and system designer, and an approximate speed-voltage curve (solid) shown with a linear approximation (dashed).

The speed of logic circuits has traditionally been approximated as linear with supply voltage, as in the dashed line in figure 1. This "rule of thumb" was reasonable for supply voltages large compared to a threshold voltage, in older technologies that did not show short-channel effects and velocity saturation within the typical range of supply voltages. According to this approximation, each 1% increase in speed costs a 1% increase in voltage, a 2% increase in switching energy, and a 3% increase in power.

Many designers have internalized a second-order correction derived from the standard quadratic MOS transistor model—speed varies as supply voltage minus threshold, and hence is more than proportional to supply voltage:

$$\frac{\Delta f}{f} > \frac{\Delta V}{V} \quad \text{"old scaling"}$$

This relation tells designers that reducing the supply voltage has a relatively large speed penalty—a rule that is no longer true at typical standard voltages.

A relevant cost function that can be optimized is the voltage-time product Vt, or equivalently Et^2, where switching energy E is proportional to V^2, and t is the logic delay, inversely proportion to f. According to "old scaling", this cost function is optimized by letting V increase toward infinity. In modern fine-line processes, velocity saturation is a dominant effect in the 3-V to 5-V region, so circuit speed actually varies less than linearly with supply voltage:

$$\frac{\Delta f}{f} < \frac{\Delta V}{V} \quad \text{"new scaling"}$$

The actual optimum of the Et^2 criterion will occur at the voltage that defines the boundary between the old scaling, where velocity saturation was negligible, and the new scaling, where it dominates the threshold effect. This critical voltage is decreasing as the technology scales, and is already below 3 Volts, according to the speed derating curve of the Philips HLL (High-speed Low-power Low-voltage) CMOS logic family [14]. Many designers have not yet internalized this new reality, so they continue to pay *more than* a 3% power increase for each 1% speed increase by keeping the supply voltage around 3 to 5 Volts.

Other sensible criteria that may be analytically optimized, as reviewed by Burr and Peterson [12], include the energy-time product Et and the switching energy E. Optimizing E puts zero weight on speed, which is probably not useful in current products, but is useful in deriving lower bounds to sensible supply voltages. At the optimum, where supply voltage is approximately equal to threshold voltage, power will be proportional to speed, but only very low speeds will be achievable. As the cost of transistors continues to fall, making up for the speed loss with increased parallelism will push efficient designs toward this limit.

Optimizing Et means that each 1% increase in speed is worth a 1% increase in switching energy and a 2% increase in power. Burr and Peterson [12] show that this criterion is optimized when the supply voltage is equal to three times the threshold voltage, ignoring velocity saturation and subthreshold effects. If velocity saturation is already significant at this voltage, the actual optimum voltage will be even lower.

If clock frequency varies linearly with voltage, power varies as the cube. In modern technologies, the power will be less than cubic at traditional high voltages, and more than cubic at sufficiently low voltages. But the large power effect at low voltages comes with a corresponding clock speed penalty. The switching energy or energy per clock cycle is still proportional to V^2. Therefore, if a system can be built with a slow low-voltage "background" mode and a faster "foreground" or "real-time" mode, computations that can be deferred to the slower mode can be done much more energy efficiently.

3. DIGITAL PARALLELISM OPTIONS

At the recent IEEE Workshop on VLSI Signal Processing, there was considerable excitement around the topic of low-power techniques. In addition to the tutorial of Brodersen *et al.* [11] mentioned above, there were several contributed papers on low-power techniques [15, 16], as well as a number of papers on architecture alternatives that provide a range of examples that the techniques could be applied to or evaluated on. For this tutorial, I draw digital parallelism examples mainly from that workshop and its predecessors, and from my own work. Varying styles and degrees of parallelism distinguish the architectural options.

One useful tool for coarse architecture-level evaluation of alternatives is the "power factor approximation" (PFA) method of Chau and Powell [17, 18], which provides a method of estimating the energy cost of a system based on the number and size of multiplication, storage, and communication operations, etc. Berkelaar and Theeuwen [19] provide a tool that directly manipulates designs to explore the area-power-delay space. In both of these approaches, however, since they do not consider supply voltage among the parameters under the control of the designer, some of their conclusions do not apply when supply voltage is jointly optimized with architecture. Alternatively, Chandrakasan *et al.* [15] specifically consider architecture and algorithm transformations jointly with voltage optimization to minimize power at a specified throughput, and find that the optimum supply voltage on real-world examples is often about 1.5 Volts, somewhat below the Et optimum.

3.1. Bit-Serial vs. Bit-Parallel

An obvious level of parallelism familiar in most digital computing machines is bit-level parallelism. Microprocessor chips have been advancing exponentially in this dimension, though at a relatively slow rate (4-bit to 64-bit machines in 20 years is a doubling every 5 years, which is quite slow compared to the rate of improvement of other metrics in this field). There is little reason to expect or want an increase in the size of arithmetic operands beyond 32, 64, or 80 bits, depending on the

application, and therefore little opportunity for increased parallelism of this sort.

At the other end of the spectrum, bit-serial methods are attractive because they reduce interconnect complexity and make it easier to employ other levels of parallelism. Lyon [20, 21, 22] has shown both dedicated (functionally parallel) and programmable (SIMD) bit-serial architectures for signal processing, and Wawrzynek and Mead have shown a configurable architecture [23, 24, 25]. The shallow logic depths and simple interconnect topologies of these designs were exploited for speed and area efficiency, but can also contribute to energy efficiency in configurations that meet the throughput needs at a lower clock rate and supply voltage. Summerfield and Lyon [26] showed that bit-serial dedicated signal processors built with standard cell automatic place and route tools can be much more locally connected, and hence more area and energy efficient, than typical bit-parallel designs.

When higher levels of parallelism are difficult to use, as in general-purpose von Neuman machines, parallelism at the bit level is essential to achieving high performance. Because of this fact, bit-parallel techniques have come to dominate the computing industry, including computer engineering education. Most architecture examples in the literature start from an implicit assumption of bit-level parallelism, but could in most cases be adapted to bit-serial approaches.

3.2. Programmable, Configurable, etc.

Programmability is the key to the cost effectiveness and proliferation of microprocessors, but comes at a high energy cost relative to more specialized or dedicated architectures. In portable products that process speech and other signals, some flexibility and re-programmability may still be valuable, but power should take on a much larger weight in the cost function. An attractive intermediate between dedicated fixed-function architectures and programmable fetch-execute architectures is the family of "configurable" architectures.

The configurable bit-serial architecture of Wawrzynek and Mead [23, 24, 25], mentioned above, implemented a variety of music synthesis algorithms about three orders of magnitude faster than a programmable microprocessor of similar technology, with much better energy efficiency.

Yeung and Rabaey's configurable "nanoprocessor" based architecture [27] has been benchmarked at about a billion operations per second per chip, with 64 nanoprocessors per chip, in a variety of signal processing domains including speech recognition. Like Wawrzynek and Mead's, this architecture uses static scheduling of a signal flow graph to attain an effective compromise between generality and efficiency. But because each nanoprocessor includes a small control store, it is even more flexible.

Dedicated hardware for predetermined regular algorithms is the most efficient approach when it is applicable. Systolic arrays represent a maximally parallel approach to dedicated hardware for large signal processing problems, and are a natural candidate for low-voltage operation. Good reviews of systolic and array techniques are given by Rao and Kailath [28] and by Kung [29].

An example of the design of a VLSI digital audio equalizer is given by Slump *et al*. [30]. Two VLSI designs were completed and compared. Both are essentially dedicated chips, but with internal control programmability. The first try, based on a single multiplier, turned out to be inefficient due to the difficulty of controlling it in a way that would take advantage of all the regularities and efficiencies known at the algorithm level. The second version, using four smaller multipliers on one chip, did the job with less than half the silicon and at a lower clock rate, due largely to more flexible programming. The details are complex enough that we do not try to draw a simple conclusion, except to note that more multipliers does not necessarily mean higher cost. Architectural exploration is still needed to arrive at good designs.

3.3. RISC, VLIW, SIMD, MIMD, etc.

Within the realm of programmable computing machines, parallelism options abound. Multi-processor machines with single or multiple instruction streams (SIMD or MIMD) have been widely explored. SIMD machines have reduced control overhead, but MIMD machines are more general. Within single-processor machines, multiple function units may be organized using very-long-instruction-word (VLIW) or superscalar techniques. VLIW is more efficient, given a good compiler, but superscalar is compatible with pre-existing instruction-set architectures, so is a more popular technique—for example, the Alpha is a dual-issue machine, or superscalar up to two instructions per cycle. Explicit vector instructions are another possibility, either controlling a vector pipeline or a SIMD vector datapath.

All of these techniques have traditionally been explored for their speed potential. But because efficient use of parallelism allows a given throughput at a lower clock rate and therefore a lower supply voltage, the same techniques are promising for power reduction. The most effective alternative for sufficiently regular problem classes, as found in speech signal processing, will be those that reduce control complexity and overhead, such as explicitly vector oriented alternatives.

4. PHYSICAL LEVEL OPTIMIZATION

Many choices are available in converting a logic design to circuits and layouts. We review a few possibilities here in terms of their impact on speed, power, and area costs.

4.1. The Speed/Power Penalty

Circuits optimized for speed use much more power than more moderate designs, when measured at a fixed clock rate and supply voltage where both will operate correctly. Under such conditions, the more moderate design is preferred. But the dominant "Hot Chips" approach optimizes for speed, ignoring other system cost factors. In this subsection, we discuss several physical-level choices, and how optimizing them for speed increases power consumption and system cost.

Transistor sizing is a key optimization tool. For very high speed designs, it is generally necessary to use large transistors, so that stray and wire capacitance is reduced relative to active gate capacitance. But at the high-speed end of the range, a large increase in size, and thus total capacitance, is needed to get a modest speed improvement. That is, the energy of a computation increases much more rapidly than the speed as transistors are made larger, with diminishing returns at higher speed. Larger transistors also lead to larger cell layouts and longer interconnect wires, to compound the problem. Tools are available for optimizing transistor sizes for speed, but usually not for other criteria.

Semi-custom ASIC designs based on gate arrays or standard cells are particularly susceptible to the problem of excessive capacitance, both because transistors are generously oversized for speed and because wire routing capacitance is often far from optimal. Good gate array and standard cell families optimized for energy efficiency would be quite useful. Dedicating extra wiring channel space and feedthrough options could help reduce long wires, and thereby trade increased area for improved efficiency and speed. Using smaller transistors and better optimized macrocells could recoup the area cost while further improving the energy efficiency at the cost of speed.

In optimizing for speed, it is important to minimize clock skew. Clock gating logic that would be useful for reducing activity in unused portions of a circuit could potentially save lots of power, but would increase clock skew and therefore adversely impact speed. In the Alpha chip, since speed was paramount, it was decided to drive the clock signal ungated to almost everywhere that it was needed, irrespective of the energy wasted.

Some circuit forms that are designed for high speed, such as "true single phase clock" (TSPC) dynamic logic [31, 32], require fast clock transitions to operate correctly. Therefore, large clock driver transistors are needed, and a large capacitive load is presented to the clock pre-driver stage. Almost half the power consumed by the Alpha is in clock distribution, and since it uses TSPC and is optimized for speed, almost half of that in the pre-drivers.

The combination of non-gated clocks and the TSPC logic form has allowed the Alpha to run at 200 MHz and claim the "Hot Chip" prize of its day, but at a tremendous cost in energy efficiency. The packaging, cooling, and high-speed interconnect issues in any product that uses the Alpha near its rated speed will dominate the system cost. Products that use the Alpha well below its rated speed, to save cost, will still suffer a substantial penalty from the fact that the chip was optimized for speed, even if they run at a reduced voltage.

In general, an improvement in circuit speed can be converted to an improvement in power by reducing the supply voltage to bring the speed back down. However, as argued above without proof, this speed-optimization approach still results in a design way off the optimum for other more realistic cost functions.

4.2. The Area/Power Symbiosis

In many cases, optimizing a circuit for area, rather than for speed, will lead to reduced computation energy with only a modest speed penalty. Smaller transistors lead to smaller cells and shorter wires, for less total capacitance, as mentioned above. Circuit choices that minimize the number of transistors may also reduce the effective capacitance being switched.

A popular technique in nMOS for saving transistors and area is the use of pass-transistor logic [8]. Simple single-ended n-type pass-transistor logic has not been used much in CMOS because the typical CMOS inverter characteristic is not compatible with the reduced logic-high level at the output of a pass transistor: it is difficult to achieve a low logic threshold and a low DC current. An "Ultra Low Power" technique proposed by Lowy and Tiemann [16] solves this problem by a simple increase in the magnitude of p-type thresholds (and optionally also a reduction in n-type thresholds), thereby achieving an excellent combination of the advantages of nMOS and CMOS. Their technique is particularly useful when the circuit must operate at conventional supply voltages and be compatible with conventional CMOS circuits. It is unclear whether it works well or has compelling advantages at very low supply voltages.

Another pass-transistor logic form termed "complementary pass-transistor logic" (CPL) has been described by Yano et al. [33]. They used cross-coupled pullups to restore logic levels after complementary pass-transistor paths, rather than modifying inverter thresholds, so more transistors are needed than in simple pass-transistor logic. According to Brodersen *et al.* [11], this circuit form, if used with reduced n-type transistor thresholds, is a full order of magnitude more energy efficient, in an adder application, than differential cascode voltage switch logic (DCVSL), with other logic forms studied falling in between (Lowy and Tiemann's new logic form was not included, but may be even better).

Another dual-rail version of pass-transistor logic, termed complementary set-reset logic (CSRL), has been proposed by Wawrzynek and Mead [23]; the pass-transistor paths need not be complementary, but instead represent set and reset conditions for registers. In many applications CSRL circuits are almost as compact as nMOS-like circuits, but simpler and more robust, requiring no process modifications. CSRL is not known for its high speed, but Mead's group and others use this versatile form extensively in conjunction with micropower analog designs [34].

In some pass-transistor circuit forms, pass transistors may be used to implement dynamic storage. In that case, as with any dynamic logic form, leakage will determine a lower bound on clock speed and hence on power, unless other methods are employed to "staticize" the design. In CSRL, shift registers are inherently fully static, but dynamic storage is used in registers that neither set nor reset.

5. "MICROPOWER" TECHNIQUES

Reducing power consumption by more than a few orders of magnitude will require more extensive changes in our thinking. We briefly comment on both digital and analog "micropower" approaches.

5.1. Low Switching Energy

The analyses of Burr and Peterson [35] show that 3-dimensional MCM packaging of high-performance signal processing systems will require operation near the switching energy optimum, with transistor threshold voltages aggressively reduced into the 100–200 mV range, due to power and cooling costs. This approach currently appears radical, but it is well motivated by technology trends and projected needs in real applications. More moderate threshold reductions will work well with most of the other power-saving methods discussed here.

In a new paper, Younis and Knight [36] show how to implement "Charge Recovery Logic" in CMOS to recover an increasing fraction of switching energy at decreasing clock rates. This approach appears even more radical, but looks like a good way to achieve a flexible speed-power tradeoff, with power porportional to speed squared, without reducing the power supply voltage.

5.2. Massive Analog Parallelism

The massively parallel "neural" micropower analog signal processing approach of Mead [37] has been applied to speech signal processing by Lyon and Mead and their colleagues [38, 39, 40, 41, 42]. Compared to the custom digital bit-serial approach of Summerfield and Lyon [26], the analog approach uses about an order of magnitude less silicon area and several orders of magnitude less power, but delivers much less precision.

Mead and Faggin [2, 43] have argued that for low-precision sensory front ends, where data variability and noise are inherently large compared to circuit component variability and noise, analog approaches show a significant advantage.

In terms of silicon area for state storage, the crossover between analog and digital approaches can be estimated based on the number of bits of precision needed. For some number of bits, N, the area of N digital memory cells will roughly match the area of an analog state storage capacitor and related circuits big enough for N bits of precision. We estimate that N is around 8 bits today. The additional local circuitry and power needed to process the stored state will almost always favor analog, since digital representations require lots of switching of lots of bits, while micropower analog approaches only incrementally change stored state voltages. Analog approaches may be preferred in terms of power even up to 12 or more bits of precision.

Analog techniques that use high-performance operational amplifiers are not as energy efficient as the micropower techniques, but may sometimes still be competitive with digital approaches.

6. LOW-POWER CHALLENGES

Implementing the ideas presented in this tutorial presents the chip designer and system designer with a number of challenges, some of which will best be met by the consensus of the industry through new standards and conventions.

Power supply voltage is a tool of great leverage for power reduction, but is not something that is easy to change continuously, or separately on different subsystems. Particularly in battery operated products, the power supply voltage must be planned to meet voltages attainable from one or several electrochemical cells, of a type selected for a variety of complex reasons. If the product can be operated without a voltage regulator, there will be greater savings, but then a more variable supply voltage must be tolerated, possibly along with a variable clock speed. If multiple chips are needed, the supply voltage must be selected to be within the intersection of their operating voltage regions—nearly impossible with today's chips if you want to operate at an efficient low voltage—or parts of the system need to operate at different voltages. If part of the system is operated at a lower voltage by dropping the supply through a dissipative series regulator, then the factor of V^2 savings that was expected at the chip level will not be realized at the system level—only one factor of V will apply.

Specialization is another powerful tool for power and silicon area savings. A general-purpose von Neuman machine has a computational overhead of several orders of magnitude over a more specialized or dedicated signal processing architecture, while a programmable DSP chip

falls somewhere between. But specialization can be costly by limiting the capabilities of a system, or by requiring additional design time. Efficient configurable or programmable architectures with a reasonable degree of specialization for signal processing applications are needed, but there is not yet a consensus as to which architectures to pursue beyond the single-chip DSP. Specializations that rely on parallelism with low control overhead, as in SIMD signal processors, offer a combination of advantages.

Analog micro-power techniques are tremendously power efficient relative to computing with digital representations of signals, but the kinds of accuracy, repeatability, and testability that we have come to expect from digital approaches are difficult or impossible to attain. This does not mean we are pessimistic on the outlook for massively parallel analog subsystems. Rather, we need to solve some problems and identify appropriate applications, in order to find good commercial success possibilities such as the one Synaptics found for optical character reading. Faggin and Mead [43] discuss this challenge.

Dynamically controlling the power-speed tradeoff is another important challenge. Making an existing portable computer architecture smart enough to judge the user's priorities may be quite difficult. Giving the user control over the tradeoff may be easier. Just as users of videotape have accepted the job of choosing 2-hour, 4-hour, or 6-hour recording mode, the user of a portable product could choose 2-hour, 4-hour, or 6-hour mode. A user stuck on an airplane would appreciate the option. If the product is a voice recorder, the tradeoff may influence sound quality; for other applications, it may influence responsiveness or other factors. The user will learn to make the tradeoff.

Can an existing chip spec'd at 200 MHz at 3.3 Volts and 30 Watts be operated an order of magnitude slower with two orders of magnitude power reduction, as expected around the *Et* optimum? Such scaling appears to be realized in Philips HLL CMOS [14]. Will the Alpha run at 20 MHz with a 1.1 Volt power supply? We suspect not. Typically somewhere in a chip will be unusual circuits with an extra threshold drop, which will kill performance rapidly as supply voltage is reduced. Chip designers need to learn that such circuits, which are motivated by optimizing performance at a particular supply voltage, take away a powerful power saving tool from the system designer.

7. CONCLUSIONS

There are a number of techniques available for reducing power consumption and system cost in speech signal processing products, and in other computing systems, as demanded especially by high-performance portable applications. Reducing the power supply voltage is the most important technique available for reducing the switching energy of logic functions. Using smaller transistors also reduces switching energy. Both of these techniques result in slower operation, so performance may need to be made up by parallelism. For parallelism to be effective, it needs to computationally efficient and power efficient, but also flexible enough for the range of tasks that it may be employed to implement. The extra silicon usage of this "low and slow" approach [44] will be a small price to pay for system-level savings.

The key impediment to taking advantage of cost-saving and power-saving techniques is the difficulty in changing to a new way of thinking about performance tradeoffs as the economics and technology of microelectronics continue to evolve. System designers and chip designers need to work together to accelerate progress.

In summary, we offer the following list of **top ten** [45] **standard ways to waste power and increase system cost:**

10. Do everything with a single central programmable processor, running fast enough for the worst-case computational demand.
9. When the processor is not needed to do real work, keep it running fast idle cycles.
8. Run the clock as fast as the chips allow, even when the application doesn't demand it.
7. Make all memory references across a central high-speed heavily-loaded system bus.
6. Optimize circuits for speed, using low fanouts, large transistors, and lots of pipelining.
5. Don't tolerate the additional clock skew of gated clocks to control which processing units are in use.
4. Use register circuits that require fast clock edges to operate correctly.
3. Use standard power supply voltages, regulated from a higher battery voltage.
2. Use digital signal representations wherever possible.

And finally, the number one way that you as a custom chip designer can waste power and increase cost in a portable speech processor:

1. To avoid being outsmarted by a clever system designer, make sure your chip won't operate correctly at a low voltage or a low clock rate.

REFERENCES

[1] A. M. Mohsen and C. A. Mead, "Delay Time Optimization for Driving and Sensing of Signals on High-Capacitance Paths of VLSI Systems," *IEEE J. Solid-State Circ.* **14**, pp. 462-470, 1979.

[2] C. Mead, "Neuromorphic Electronic Systems," *Proc. IEEE* **78**, pp. 1629–1636, 1990.

[3] J. Narkiewicz and W. P. Burleson, "VLSI Performance/Precision Tradeoffs of Approximate Rank-Order Filters," *VLSI Signal Processing, **V*** (K. Yao, R. Jain, and W. Przytula, eds.) pp. 185–194, IEEE 1992.

[4] A. Orailoglu and R. Karri "A Design Methodology for the High-Level Synthesis of Fault-Tolerant ASICs," *VLSI Signal Processing, **V*** (K. Yao, R. Jain, and W. Przytula, eds.) pp. 417–426, IEEE 1992.

[5] L. C. Stewart, A. C. Payne, and T. M. Levergood, "Are DSP Chips Obsolete?," *Intl. Conf. on Signal Processing Applications and Technology,* pp. 178–187, DSP Associates, Boston, 1992.

[6] D. W. Dobberpuhl *et xxii al*. "A 200-MHz 64-b Dual-Issue CMOS Microprocessor," *IEEE J. Solid-State Circ*. **27** pp. 1555–1567, 1992.

[7] R. M. Swanson and J. D. Meindl, "Ion-Implanted Complementary MOS Transistors in Low Voltage Circuits," *IEEE J. Solid-State Circ*. **7**, pp. 146–153, 1972.

[8] C. Mead and L. A. Conway, *Introduction to VLSI Systems,* Addison-Wesley, 1980.

[9] E. A. Vittoz, "Micropower Techniques," in *Design of MOS VLSI Circuits for Telecommunications* (Y. Tsividis and P. Antognetti, eds.), Prentice-Hall, 1985.

[10] J. Burr, P. R. Williamson, and A. Peterson, "Low Power Signal Processing Research at Stanford," *3rd NASA Symposium on VLSI Design,* pp. 11.1.1–11.1.12, Moscow, Idaho, Oct., 1991.

[11] R. Brodersen, A. Chandrakasan, and S. Sheng, "Low-Power Signal Processing Systems," *VLSI Signal Processing, **V*** (K. Yao, R. Jain, and W. Przytula, eds.) pp. 3–13, IEEE 1992.

[12] J. Burr and A. Peterson, "Ultra Low Power CMOS Technology," *3rd NASA Symposium on VLSI Design,* pp. 4.2.1–4.2.13, Moscow, Idaho, Oct., 1991.

[13] V. Von Kaenel, P. Macken, and M. G. R. Degrauwe, "A Voltage Reduction Technique for Battery-Operated Systems," *IEEE J. Solid-State Circ*. **25** pp. 1136–1140, 1990.

[14] Philips Semiconductors, "Fast, low-power HLL & LV-HCMOS logic families… …for systems with 1.2 V to 3.6 V supplies," undated product literature.

[15] A. Chandrakasan, M. Potkonjak, J. Rabaey, and R. Brodersen, "An Approach for Power Minimization Using Transformations," *VLSI Signal Processing, **V*** (K. Yao, R. Jain, and W. Przytula, eds.) pp. 41–50, IEEE 1992.

[16] M. Lowy and J. J. Tiemann, "Ultra-Low Power Digital CMOS Circuits," *VLSI Signal Processing, **V*** (K. Yao, R. Jain, and W. Przytula, eds.) pp. 31–40, IEEE 1992.

[17] S. R. Powell and P. M. Chau, "Estimating Power Dissipation of VLSI Signal Processing Chips: The PFA Technique," *VLSI Signal Processing, **IV*** (H. S. Moscovitz, K. Yao, and R. Jain, eds.), pp. 250–259, IEEE 1990.

[18] S. R. Powell and P. M. Chau, "A Model for Estimating Power Dissipation in a Class of DSP VLSI Chips," *IEEE Trans. Circ. and Sys.* **38** pp. 646–650, 1991.

[19] M. R. C. M. Berkelaar and J. F. M. Theeuwen, "Real Area-Power-Delay Trade-Off in the *Euclid* Logic Synthesis System," *Custom Integrated Circuits Conference,* pp. 14.3.1–14.3.4, IEEE, 1990.

[20] R. F. Lyon, "A Bit-Serial VLSI Architectural Methodology for Signal Processing," in *VLSI 81 Very Large Scale Integration* (J. P. Gray, ed.), Academic Press, 1981.

[21] R. F. Lyon, "FILTERS: An Integrated Digital Filter Subsystem" and "MSSP: A Bit-Serial Multiprocessor for Signal Processing," in *VLSI Signal Processing: A Bit-Serial Approach,* P. B. Denyer and D. Renshaw, Addison-Wesley, 1985.

[22] R. F. Lyon, "MSSP: A Bit-Serial Multiprocessor for Signal Processing," in *VLSI Signal Processing* (P. Cappello et al., eds.), IEEE Press, 1984.

[23] J. Wawrzynek and C. Mead, "A New Discipline for CMOS Design: An Architecture for Sound Synthesis," in *Chapel Hill Conference on Very Large Scale Integration,* H. Fuchs, ed., Computer Science Press, 1985.

[24] J. Wawrzynek and C. Mead, "A VLSI Architecture for Sound Synthesis," in *VLSI Signal Processing: A Bit-Serial Approach,* P. B. Denyer and D. Renshaw, Addison-Wesley, 1985.

[25] J. Wawrzynek and C. Mead, "A Reconfigurable Concurrent VLSI Architecture for Sound Synthesis," *VLSI Signal Processing, **II*** (S. Y Kung, R. E. Owen, and J. G. Nash, eds.), pp. 385–396, IEEE Press, 1986.

[26] C. D. Summerfield and R. F. Lyon, "ASIC Implementation of the Lyon Cochlea Model," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. V-673–676 1992.

[27] A. K. W Yeung and J. M. Rabaey, "A Data-Driven Architecture for Rapid Prototyping of High Throughput DSP Algorithms," *VLSI Signal Processing, **V*** (K. Yao, R. Jain, and W. Przytula, eds.) pp. 225–234, IEEE 1992.

[28] S. K. Rao and T. Kailath, "Regular Iterative Algorithms and Their Implementation on Processor Arrays," *Proc. IEEE* **76**, pp. 259–269, 1988.

[29] S. Y. Kung, *VLSI Array Processors*, Prentice-Hall, 1988.

[30] C. H. Slump, C. G. M. van Asma, J. K. P. Barels, and W. J. A. Brunink, “Design and Implementation of a Linear-Phase Equalizer in Digital Audio Signal Processing,” *VLSI Signal Processing, **V*** (K. Yao, R. Jain, and W. Przytula, eds.) pp. 297–306, IEEE 1992.

[31] Y. Ji-ren, I. Karlsson, and C. Svensson, “A True Single Phase Clock Dynamic CMOS Circuit Technique,” *IEEE J. Solid-State Circ.* **22**, pp. 899–901, 1987.

[32] J. Yuan and C. Svensson, “High-speed CMOS Circuit Technique,” *IEEE J. Solid-State Circ.* **24**, pp. 62–70, 1989.

[33] K. Yano, T. Yamanaka, T. Nishida, M. Saito, K. Shimohigashi, and A. Shimizu, “A 3.8-ns CMOS 16x16-b Multiplier Using Complementary Pass-Transistor Logic,” *IEEE J. Solid-State Circ.* **25**, pp. 388–395, 1990.

[34] C. A. Mead and T. Delbrück, "Scanners for Visualizing Activity of Analog VLSI Circuitry," *Analog Integrated Circuits and Signal Processing* **1**, pp. 93–106, 1991.

[35] J. Burr and A. Peterson, “Energy Considerations in Multichip Module-based Multiprocessors,” *IEEE Intl. Conf. on Computer Design,* pp. 593–600, Oct., 1991.

[36] S. Younis and T. Knight, “Practical Implementation of Charge Recovering Asymptotically Zero Power CMOS,” *1993 Symposium on Integrated Systems* (C. Ebeling and G. Borriello, eds.), Univ. of Washington, in press (and personal communication).

[37] C. Mead, *Analog VLSI and Neural Systems,* Addison-Wesley, 1989.

[38] R. F. Lyon and C. Mead, “An Analog Electronic Cochlea,” *IEEE Trans. ASSP* **36**, pp. 1119–1134, 1988.

[39] R. F. Lyon, “CCD Correlators for Auditory Models,” *25th Asilomar Conference on Signals, Systems and Computers,* IEEE Computer Society Press, 1991.

[40] C. A. Mead, X. Arreguit, and J. Lazzaro, “Analog VLSI Model of Binaural Hearing,” *IEEE Trans. Neural Networks* **2**, pp. 230–236, 1991.

[41] Lazzaro, J. and Mead, C., “Silicon models of auditory localization,” in *An Introduction to Neural and Electronic Networks* (S. F. Zornetzer, J. L. Davis, and C. Lau, eds.), Academic Press, 1990.

[42] L. Watts, D. Kerns, R. Lyon, and C. Mead, "Improved Implementation of the Silicon Cochlea," *IEEE J. Solid State Circ.* **27**, pp. 692–700, May 1992.

[43] F. Faggin, “VLSI Implementation of Neural Networks," in *An Introduction to Neural and Electronic Networks* (S. F. Zornetzer, J. L. Davis, and C. Lau, eds.), Academic Press, 1990.

[44] G. Gilder, *MICROCOSM*, pp. 145–148, Simon and Schuster, NY, 1989.

[45] D. Letterman *et al., The “Late Night with David Letterman” Book of Top Ten Lists,* Simon & Schuster, 1990.