Scaling, Power, and the Future of CMOS

Mark Horowitz, Elad Alon, Dinesh Patil, Stanford University
Samuel Naffziger, Rajesh Kumar, Intel
Kerry Bernstein, IBM

I. Introduction

In 1974 Robert Dennard wrote a paper [1] that explored different methods of scaling MOS devices, and pointed out that if voltages scaled with lithographic dimensions, one achieved the benefits we all now assume with scaling: faster, lower energy, and cheaper gates. The lower energy per switching event exactly matched the increased energy by having more gates and having them switch faster, so in theory the power per unit area would stay constant. While we have not followed these scaling rules completely, for the past 30 years we could count on technology rescuing design projects from missing their performance or power targets.

Unfortunately, no exponential can last forever and recently scaling has diverged from the ideal relationships that Dennard proposed many years ago. The fundamental problem, which Dennard noted in his paper, is that all device voltages can't scale; in particular, since kT/q does not scale and leakage currents are set by the transistor's threshold voltage, there is a limit to how low one can make a transistor's V_{th}. With V_{th} fixed, changing V_{dd} simply trades off energy and performance. The net result is that from the 130mn technology forward, V_{dd} has been scaling slowly, if at all.

This poor future power scaling, combined with previously applied aggressive performance scaling techniques, has made power the number one problem in modern chip design. Designers can no longer focus on creating the highest performance chips because it is nearly guaranteed that the highest performance circuit they can create will dissipate too much power. Instead, designers must now focus on power efficiency in order to achieve high performance while staying under their power constraints.

This paper briefly reviews the forces that caused the power problem, the solutions that were applied, and what the solutions tell us about the problem. As systems became more power constrained, optimizing the power became more critical; viewing power reduction from an optimization perspective provides valuable insights. Section III describes these insights in more detail, including why V_{dd} and V_{th} have stopped scaling. Section IV describes some of the low power techniques that have been used in the past in the context of the optimization framework. This framework also makes it easy to see the impact of variability, which is discussed in more detail in Section V along with the adaptive mechanisms that have been proposed and deployed to minimize the energy cost. Section VI describes possible strategies for dealing with the slowdown in gate energy scaling, and the final section concludes by discussing the implications of these strategies for device designers.

II. Scaling, kT/q, and the Problem

While CMOS technology was invented in 1963, it took the first power crisis in the 1980s to cause VLSI chips to switch from nMOS, which during the late 1970s was the dominant VLSI technology. During this period V_{dd} was fixed to 5V, and was not scaling with technology to maintain system compatibility. For control and speed reasons, this meant that the depletion thresholds for the nMOS loads did not scale rapidly, so the current per minimum gate scaled only slowly. The net result was that the power of the chips started growing with the complexity, and chips rapidly went from a Watt to multiple Watts, with the final nMOS VLSI chips dissipating over 10W [2]. While the peak currents in CMOS were as large as nMOS, since they were transients that lasted roughly 1/20 of a clock cycle, a CMOS processor ran at roughly 10x lower power than a similar nMOS chip.

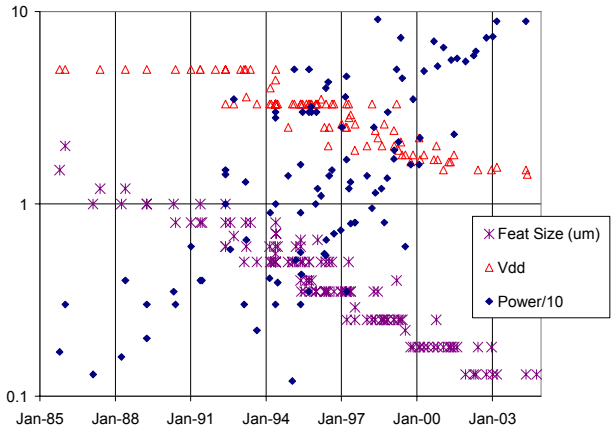

Fig 1. Microprocessor V_{dd}, Power/10, and feature size versus year. From 1994 to today V_{dd} has roughly tracked feature size.

Fig 1 uses microprocessor data to track CMOS technology scaling since the mid-1980s to today. It plots technology feature size, V_{dd}, and power versus time. Through four generations of technology, from the 2µm generation in the early 1980s to the 0.5µm generation in the mid-1990s, the power savings from switching to CMOS was large enough that V_{dd} did not need to scale and was kept constant at 5V. To mitigate high fields and reduce power, V_{dd} started to scale with 0.5µm technology, and has continued to scale at roughly V_{dd} = feature size * 10V/µm until the 130nm technology[1].

Power continued to increase during this time, even though we were roughly following the ideal scaling relationship. Part of this increase in power was due to increases in area, but power density increased by 30x during this period as well. The

[1] In high-performance microprocessor technologies, the supply voltage scaling slowed down even earlier, at the 180nm node.

0-7803-9269-8/05/$20.00 (c) 2005 IEEE

principle causes of this increase in power were the performance optimizations (such as improved circuit design, better sizing optimization, and deeper pipelines) that were applied to microprocessor chips. Fig 2 plots the clock cycle time normalized to the delay of an inverter, and shows that the frequency scaled much faster than the basic gate speed. Frequency increased by about 2x per generation, which caused the power density to exponentially rise.

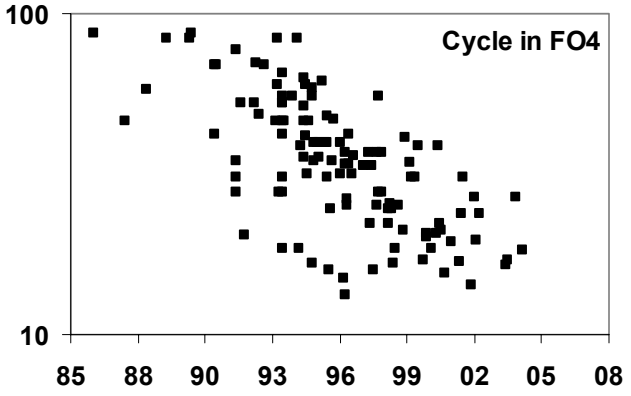

Fig 2. Plot of processor cycle time measured in the estimated delay of a Fanout of 4 inverter in that technology.

Fortunately for power issues, both the increase in die size and the practice of "super" frequency scaling have recently stopped. Because the thermal voltage does not scale, we have unfortunately hit the point where we can no longer continue to reduce V_{th}. V_{th} is critical because for most modern devices the sub-threshold leakage still dominates the total leakage, and this current is exponentially related to V_{th}. Reductions in V_{th} have made leakage power large enough that it needs to be considered in the power budget – this means that to minimize power, V_{th} is set as the result of an optimization, and not set by technology scaling.

III. Optimization Perspective

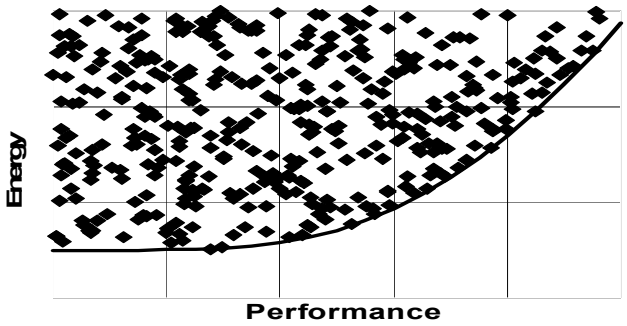

Fig 3. The Pareto optimal curve is the boundary of the space of all possible solutions in the Energy-Performance plane

Imagine that one tried all the different ways to build a unit (e.g. an adder) using all possible transistor sizes, circuit methods, and supply and threshold voltages. Fig 3 shows the result of plotting all of these solutions on a graph with performance on one axis and the energy consumed for a single operation on the other. The optimal design point depends on the application constraints, e.g. max. power or min. performance requirements, but will always lie on the lower right edge of the feasible set that forms the Pareto optimal points. The qualitative shape of this curve is always the same, and follows from the law of diminishing returns. Moving between low energy points causes large shifts in performance for small energy changes, while high performance points require large amounts of energy for small performance improvements. Fig 4 estimates the energy-performance trade-offs using published microprocessor data.

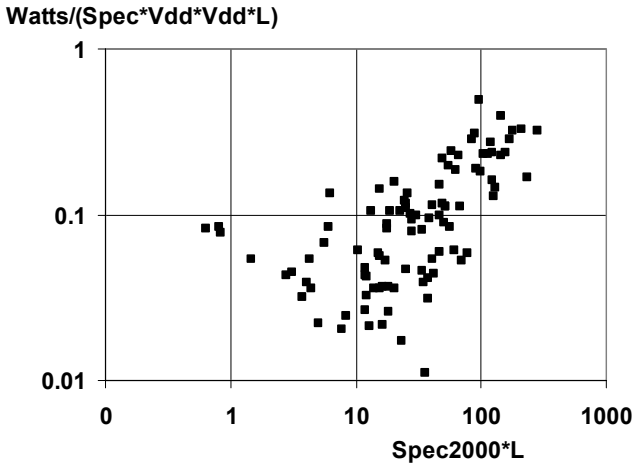

Fig 4. Energy consumed per operation for CMOS processors built during the past 20 years – the data has been normalized to remove direct technology effects. These commercial processors differ by over 10x in the energy needed to execute an operation.

While a complete optimizer does not exist, tools that optimize a subset of the parameters exist. The best tools today handle V_{dd}, V_{th} and transistor sizing for a given circuit topology [3],[4]. The result of the tool is a sized circuit, and the optimal values of V_{dd} and V_{th} to use for the circuit.[2] In this framework, V_{dd} and V_{th} are not scaling parameters, but rather they are set by the results of the power optimization.

V_{dd}	nMOS V_{th}	Sensitivity $(\partial E/\partial V_{dd})/(\partial Perf./\partial V_{dd})$
550mV	321mV	0.031
700mV	189mV	0.194
850mV	183mV	0.7633
1V	182mV	1.8835

Table 1. Optimal V_{dd}, V_{th}, and sensitivity for a 90nm inverter at 80°C with 20% activity factor driving a fixed capacitive load.

One can estimate V_{dd} and V_{th} by remembering that at each of the optimal points, the marginal cost in energy for a change in delay is the same for all of the parameters that the optimizer is free to control[3]. Moreover, since we know the basic relationship between V_{dd}, energy, and delay for a simple inverter, and the energy and delay of all CMOS gates have similar V_{dd} dependence, we can estimate the trade-offs for an entire design by using the inverter data.

[2] If the technology provides multiple transistor types, the tools can even select the correct V_{th}'s for the application, and which V_{th} each transistor should use.
[3] Or that parameter value is constrained to a user-defined max. or min. value. For example, V_{dd} might be constrained to V_{ddMax} by reliability considerations.

Since the marginal energy cost for a change in performance should be the same for both V_{dd} and V_{th}, for each V_{dd} there is a unique value of V_{th} which minimizes the energy. As V_{dd} increases, increasing the amount of energy you are willing to spend on improving performance, the optimal V_{th} will decrease, increasing the leakage current so that the marginal energy costs for each unit of performance remain in balance. As Nose and Sakurai have shown [5], the resulting optimization sets the leakage energy to be about 30% of the active power.[4] Thus the large rise in leakage current that accompanies new high-performance technology is intentional – it is done to reduce the total power the chip dissipates.

IV. Low Power Circuits and Architecture

This same view on equalizing the marginal delay cost for a reduction in energy holds for low-power circuits and architectures, although it is rarely discussed that way. Many papers simply discuss energy savings without discussing the performance costs. A technique with moderate performance cost might be well-suited for a low-speed machine with a large marginal delay cost per unit energy, but would actually make the power higher if it was applied to a fast machine with a small marginal delay cost for energy reduction.

The best techniques have negative performance cost to reduce energy – they improve both performance and energy. These techniques generally involve problem reformulation or algorithmic changes that allow the desired task to be accomplished with less computation than before. While they are by their nature application specific, these techniques can change the power required for a task by orders of magnitude [6], more than any other method. These changes are generally made at the architectural level, but sometimes implementation decisions are critical too. Adding specialized hardware reduces the overhead work a more general hardware block would need to do, and thus can improve both energy and performance. Since these ideas require domain specific insight, no tools to support this activity exist.

The next set of low-power techniques are those that nominally have zero performance cost – these techniques remove energy that is simply being wasted by the system. Before power became a critical problem designers were rarely concerned whether a unit was doing useful work, they were only concerned about functionality and performance. At the circuit level these techniques generally are tied to clock gating to prevent units from transitioning when they are not producing useful outputs. The larger power reductions come from applying this idea at the system level. Subsystems often support different execution states, from powered off, to ready-to-run. Modern PCs use an interface called ACPI to allow the software to deactivate unused units so that they don't dissipate power [7]. A digital cell phone's power advantage over analog phones comes mostly from an architecture that was borrowed from pagers in which the phone is actually off most of the time.

[4] But the minima is flat – from 20% to 100% there is <10% energy change.

The dual of reducing energy with no performance cost are techniques that improve performance with no energy cost. Parallelism is the most commonly used example of this approach [8]. For applications with data parallelism, it is possible to use two functional units each running at half rate, rather than using a single unit running at full rate. Since the energy per operation is lower as you decrease performance, this parallel solution will dissipate less power than the original solution. Often there is no need to explicitly build parallel units because pipelining can achieve a similar effect.

In reality the energy cost of parallelism is not zero, since there is some cost in distributing operands and collecting the results, or in the pipeline flops, but these costs are generally modest. The efficiency of parallelism is often limited by the application – it must have enough work to do that partially filled blocks don't occur that often, since these increase the average energy cost.

Other "low-power" techniques are really methods to reduce energy by increasing the delay of the circuit, or techniques that give the low-level optimizer more degrees of freedom. The former include using power gating to reduce leakage and low swing interconnects, while the latter include dual threshold technologies [9], or allowing gates to connect to either of two different power supplies [10]. As previously mentioned, techniques with modest delay costs might be advantageous for a low-performance design, but may not be in a high-performance system since these systems operate at a point where the allowable marginal delay cost is very small.

Most of the remaining low power techniques are really methods of dealing with application, environmental or fabrication uncertainty, so before we describe them we first need to discuss the energy cost of variability.

V. Impact of Variability on Energy

So far we have examined the optimization problem as if we knew what the desired performance requirement was, and we also had the relationship between our control variables (V_{dd}, V_{th}, etc.) and performance. Neither of these assumptions is true in a real system. If we build a fixed system for an application with variable computation rates, its performance must exceed the requirements of the application, and its power must always be smaller than what the system can support. Since we have shown that higher levels of performance require higher energy per operation, this solution will, on average, waste energy.

As an example, consider a system with no variations except that the input comes in bursts, and the machine is active only 1% of the time. If techniques such as power gating (also known as sleep transistors [14]) are not used, the optimal V_{th} will make the leakage power 30% of the **average** active power, or 100x lower than in the case when the unit is busy all the time. This will increase V_{th} by roughly 160mV, and force V_{dd} to rise by a similar percentage to maintain the desired performance. The increase in V_{dd} makes the energy per operation higher, so the low duty cycle is translating to loss in power. If the threshold increases by 50%, then V_{dd}

will increase by roughly 40%, roughly doubling the energy of each operation.[5]

Unlike the deterministic optimization problem that was described in the previous section, fabrication variations change the problem into the optimization of a probabilistic circuit. The inability to set all device parameters to exactly their desired value has an energy cost. To understand this cost and what can be done to reduce it, we first need to look at the types of variability that occur in modern chips. The uncertainty in transistor parameters can be broken into three large groups by looking at how the errors are correlated. Die to Die (D2D) variations have large correlation distances and affect all transistors on a die in the same way. Within Die (WID) variations are correlated only over small distance, affecting a group of transistors on the die. Random variations (Ran) are uncorrelated changes that affect each transistor – this last group depends on the area of the device [11]. The correlated variations are often systematic in nature, and can often be traced to design differences in parameters such as local density or device orientation.

With uncertainty, the first question is what to use as the objective function for the optimization. Generally, one wants to optimize the energy and performance specifications so that some fraction of the parts will meet these targets. For example, if we wanted to sell 80% of the parts, the performance specification would be the performance of the part that is slower than 90% of the distribution, and the energy spec would be the energy of the part that is higher than 90% of the distribution. Thus in the face of uncertainty, the optimizer must use this lower performance and higher power as the metrics for the part, even though they can't exist on the same die. This cost is easy to see for D2D variations since all transistors will be changed by the same amount, so the underlying optimization problem remains the same. Fig 5 shows how the optimal energy-performance curve degrades as the uncertainty in V_{th} increases.

While the optimization problem gets more complex with Ran and WID variations, since one must consider variations during construction of the delay paths to be optimized, some tools for this task are starting to emerge [12],[13]. The effect of V_{th} variation on leakage current is also critical, but it is easier to calculate. For leakage, we are interested in calculating the average leakage current of each transistor, and for exponential functions, this can be much larger than predicted by simply using the average V_{th}. Even though averaging all of the device's threshold voltages together may result in the desired V_{th}, the leakage of the devices with lower thresholds will be exponentially larger than that of the devices with high threshold. This means that the total leakage will be dominated by the devices with lower thresholds, and hence the average leakage per device will be significantly higher than the leakage of a single device with the average V_{th}.

[5] The cost is even higher if we compare it to the case where the machine only needs to handle 1% of the peak rate, but this input was evenly distributed. In this case the required performance has decreased which dramatically reduces the required energy. In fact Chandrakasan [33] proposed adding FIFO buffers between functional elements to smooth out computation bursts to gain energy efficiency.

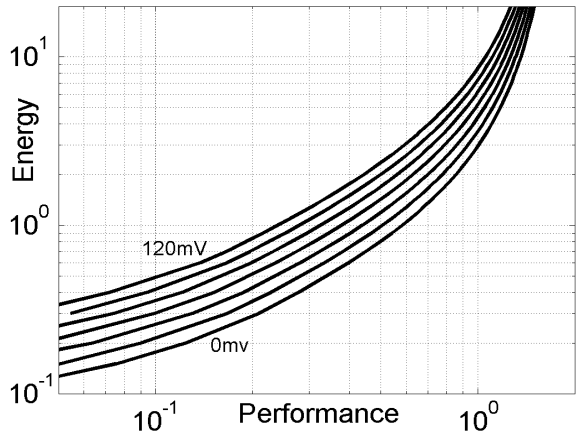

Fig 5. Optimal energy-performance curves for inverters; power is calculated for the circuits with $-\Delta V_{th}$ shift, and delay from those with $+\Delta V_{th}$ shift. The cost is much higher if the optimizer does not consider the variation when setting the parameters.

The cost of these variations strongly depends on which mechanisms are used to reduce the amount of margining that is required. When no compensation at all is used the costs of all the types of variations are similar to the D2D cost shown in Fig 5. One of the first techniques systems used was to realize that some of the power limits were actually set by worst-case cooling concerns. By putting thermal feedback in the system, the manufacturer can use higher power parts, and in rare case where a high power part has cooling trouble the system simply thermally throttles the part.

One can reduce the energy costs by allowing the system to change the value of V_{dd} and possibly even V_{th} to better optimize the power for the desired performance given the actual fabrication parameters. While V_{dd} control is not that difficult, since today many chips already have a dedicated power regulator associated with them, V_{th} control is more problematic. In modern technologies it is difficult to build devices that provide any ability to control V_{th} post fabrication. Back-gate control of V_{th} for short-channel devices is less than 100mV, and modulates leakage power by less than 5x, even if one is willing to both reverse and forward bias the substrate [14],[15]. Even this restricted control will be of some help, since it will enable the chip to operate closer to the true optimal point.

For D2D variations[6] one can think of using a technique that adapts V_{dd} to ensure that the part runs at the desired speed, and/or adapts V_{th} using the small control available to bring the leakage current in range [16],[17]. In this case the cost of the variation is that the resulting circuit is not running at a point on the Pareto optimal curve. Rather, the fabrication moved the design off that point, and we have to use V_{dd} and V_{th} to correct for the variation. As shown in Fig 6, for small variations in device parameters, the cost of variation with this control is small. Notice that in this adaptive scheme, the feedback does not need to be dynamic. In many cases each

[6] Or more generally for variations where the correlation distances are large enough to build an adaptive supply to correct for the variations.

part is tested, and the optimal setting for V_{dd} and possibly V_{th} are programmed on the part to be used during system start-up.

WID variations are harder to handle, since it is impossible to actively correct for these variations. The only choice here is to margin the circuit to guarantee that it will meet spec even when the gates are slow. This cost will be similar to the uncorrected energy overhead shown in Fig 5.

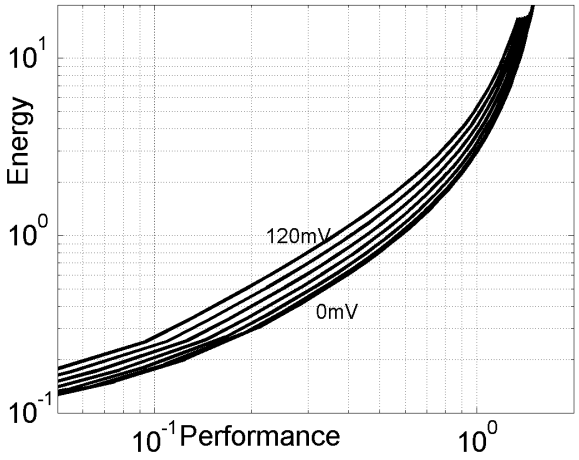

Fig 6. Cost of D2D ΔV_{th} if V_{dd} is allowed to adapt to each part. For small ΔV_{th} the cost is very small – the 20mV ΔV_{th} curve is almost on top of the 0mV curve. For larger changes adapting V_{dd} become less effective, but still reduces the overall cost by about 2x.

The ability to adjust V_{dd} and V_{th} also allows the hardware to adjust to an application's computing requirements. In mobile and embedded devices, the operating system can provide directives to the hardware to inform it about the current performance demands on the system, and hence the chip's supply voltage, frequency, and perhaps even threshold voltage [18],[19],[20] would be adjusted to keep power consumption at the minimum level. In systems that adapt V_{dd} and V_{th}, an important issue is how to determine the correct value of V_{dd}. Most current systems that make use of these techniques have a small set of distinct operating modes that are defined a priori, for example in early laptops where the frequency/power of the processor was switched between two settings based only on whether the machine is operating off of the battery or an external power source. This is done to allow the chips to be tested before being shipped to customers. More aggressive systems use on-chip delay matching circuits to control V_{dd} [21], but these matching circuits must be margined to ensure that they are always longer than the real paths. Austin and Blaauw in Razor [22] have shown it is possible to use the actual hardware to check timing. They then built a system with an error recovery mechanism that allowed them to reduce the supply so that the critical paths on occasion were too slow. The additional voltage scaling decreased the effective energy per op by 54%.

VI. Looking Forward

While there are many complex issues about future designs that are hard to foresee, simple math says that if scaling continues and dies don't shrink in size then the average energy per gate must continue to decline by about 2x per generation to keep power constant. Since the gates are shrinking in size, we can assume that 1.4 of this 2x will come from the lower capacitance associated with the devices. There are three basic approaches to deal with the other factor of 1.4. One possibility is that the average V_{dd} will continue to scale, but more slowly (20% per generation) than before. Another option is that the supply stays constant, but the average activity factor of the gates falls so the total energy remains constant. A third option is that dies simply shrink to maintain the required power levels.

Historically, one method of improving hardware performance is to exploit more parallelism, and add more functional units. If the power supplies continue to scale down in voltage (although slowly), we can add functional units while staying inside of our power budget. The side-effect will be that the basic gates will be operating at a point where the marginal energy cost for performance is smaller than it was before (see Table 1)[7]. Thus you can build many active gates, but all of the functions that you add must have very low marginal energy cost compared to the added performance they are supplying. The domain where this type of solution makes the most sense is in applications that have abundant levels of parallelism, where the utilization of the functional units is high, so the energy overhead is small.

The ultimate limit of this type of scaling leads one to reduce V_{dd} below V_{th} and operating transistors in subthreshold. This approach has been explored in papers looking at minimum energy solutions [23],[24]. Interestingly, in the subthreshold region the marginal energy cost of changing V_{th} is zero, since the on-to-off current ratio is completely set by the value of V_{dd}. Changing V_{th} changes both the leakage current and the cycle time (set by the on current), so their product (leakage energy) is constant. Analogous to the minimum delay solution, these machines operate where the marginal cost in delay for lowering the energy is infinite, so the micro-architecture of these designs should contain only the minimum hardware needed to perform the function, including using minimum-sized transistors[8]. The performance cost to enter into the subthreshold region can be very large. These machines run 2-3 orders of magnitude slower than an energy efficient machine running at a low V_{dd} (but $>V_{th}$), and have energy/operation that are 2 times lower.

While these large, parallel solutions were demonstrated running at very low voltages in the early 1990s [8], they have not been widely adopted in industry. One of the issues we have not considered is cost. As the level of parallelism increases, the marginal improvement in energy for doubling the parallelism decreases – in fact, as we just described, at the limit of subthreshold operation an infinite number of units could be added without any further decrease in the energy per operation. This means that a more cost-effective approach may be to raise the supply voltage slightly to double the

[7] Notice that when we scale V_{dd} down, V_{th} will increase in magnitude since we must decrease the energy cost of the leakage (dynamic power per gate is decreasing), and the cycle time in nanoseconds is increasing.

[8] In this operating condition some V_{th} control is critical, since one needs to set the pMOS to nMOS current ratio. This ratio is one important factor that sets the minimum V_{dd} where the logic will function.

performance of the functional units, and halve the number of functional units in order to reduce the die size. While this means that the part may not be operating strictly on the optimal tradeoff curve, the marginal energy costs (and hence the energy penalty) in this regime of operation are very small, so this small added energy might be worth creating a more economical solution.

Another technique that has been used to improve performance is to create specialized hardware tailored to a specific application or application domain. Of course to cover a broad application space, the chip will now need a number of different specialized functional units. This solution fits nicely with the other method to reduce the average power per gate, reducing the average activity factor of each gate. Since in a system with many specialized units the number of concurrent functional units that are simultaneously active at any one time is limited, the average activity is low. The activity factor of a unit while running would not need to decrease. This approach is already being used in cellphone systems, where all the digital functions have been integrated on one chip, and many functions have their own dedicated processing hardware. It is possible that microprocessors might move in this direction as well, by building heterogeneous multiprocessors. A chip might contain a fast, power hungry conventional processor as well as a simpler version of this processor connected to a vector execution unit, or an array of parallel processors. The conventional processor would be used for normal sequential applications, but applications with data parallelism could leverage the parallel execution engine and get significantly higher performance.

If specialization is not possible, simple integration can lead to performance and energy efficiencies. These efficiencies mean that even in the absence of an explicit power-saving approach, the die will not need to shrink in area by 1.4x in each generation. For example, current state-of-the-art high-speed off-chip interfaces consume roughly 20-30mW/Gb/s per channel [25], while on-chip interfaces have already been demonstrated that require roughly an order of magnitude lower energy [26],[27]. Processor I/O bandwidths are approaching 100GB/s and ~10-20W [28], and hence removing these I/O pins would provide extra power for additional logic gates. Since this integration also improves latency, it is likely that these integration projects will continue to deliver reduced system cost and power, as well as improved performance. Previous examples of this type of scaling leading to significant improvements can already be found in the inclusion of the floating point unit on the Intel 486 [29] and the memory controller on the AMD Opteron [30] processors.

In all of these situations, it is clear that power control for chips will become more sophisticated, and more critical for a number of reasons. First, as chips become more power constrained, they will be forced to operate closer to the real performance limits of the applications. The cost of margining the parts for worst case operation will simply be too high, and in fact some commercial parts are already making use of these ideas. As previously mentioned, all laptop processors use some kind of V_{dd} / frequency control to change the energy efficiency of the processors depending on whether they are running from the wall or on batteries, and Transmeta used information about the operating system load as well as parametric test data to help the processor adapt its supply voltages and frequency [31].

The next-generation Itanium II has a sophisticated power supply control system that keeps the power dissipation of the part constant. When it is running sequential code that leaves most of the functional units idle (i.e. does not hit the highest power dissipation), it raises its supply voltage and clock frequency so that it can run this code faster. In more parallel sections of the code, it lowers the supply and frequency to maintain the same power dissipation [32].

Even if the designers are not aggressive in removing the energy overhead of margining, to avoid the leakage power of the idle units designers will be forced to break up their chip into different power domains, and then activate only the necessary domains. Furthermore, particularly in the heterogeneous machines, the actual value of the supply voltage (and correspondingly the threshold voltage) applied to a functional unit when it is enabled should be different than the value used when activating other types of units in order to keep every active unit on its own Pareto curve.

VII. Conclusions

Power has always been a concern with scaling, and raising power levels of nMOS VLSI chips in the 1980s caused the industry to switch to CMOS. Since power became an issue in CMOS design in the 1990s many approaches have been used to try to reduce the growing power of VLSI systems. The two approaches that were most successful were the energy efficiency of technology scaling, and system level optimization to reduce the required computation. Work at the circuit and microachitecture levels had a smaller effect. The key point to remember about reducing chip power is that power and performance are integrally connected. Lowering power by reducing performance is easy, but the trick is to reduce energy without affecting the circuit's performance. Unfortunately, many of the magic bullets for decreasing energy without affecting performance have already been found and exploited. While there are no quick fixes, power growth must be addressed by application specific system level optimization, increasing use of specialized functional units and parallelism, and more adaptive control.

In looking at this future world, one wonders if the push for ever shorter channel length devices will continue to bring strong returns. Already the return in gate speed is modest, and with supplies not scaling, the energy savings come from the parasitic and wire capacitance scaling. Short gates force very thin gate oxides and gate leakage, a problem that has become a power issue. Even today most applications would benefit from other types of devices, like a very small, but very low leakage device. These devices would be used for memories and other circuits that have very low average activity ratios. The transition probability is low enough that

the optimal V_{dd} for these structures can be larger than other gates, and they don't need very short effective channel lengths – they just need to be physically small. Another interesting new device optimization issue is the relationship between intrinsic device speed and variability. Both slower devices and uncertainty in devices cost energy, and hence the most energy efficient devices may no longer be those with the shortest effective channel length. In addition, as variability increases, the minimum operating voltage gets pushed up due to stability issues, which reduce the energy/delay range of the V_{dd} knob. Any process improvements that increase the range of V_{dd} and V_{th} control will enable better energy efficiency. Finally, devices and efficient energy-storage elements that allow one to build efficient power conversion on-die would decrease the cost of the power control schemes that will be needed in the future.

VIII. Acknowledgments

M. Horowitz, E. Alon, and D. Patil would like to thank MARCO for funding support.

IX. References

[1] R.H. Dennard, F.H. Gaensslen, V.L. Rideout, E. Bassous, and A.R. LeBlanc, "Design of Ion-Implanted MOSFET's with Very Small Physical Dimensions," *IEEE Journal of Solid-State Circuits*, Oct. 1974.

[2] M. Forsyth, W.S. Jaffe, D. Tanksalvala, J. Wheeler, and J. Yetter, "A 32-bit VLSI CPU with 15-MIPS Peak Performance**,"** *IEEE Journal of Solid-State Circuits*, Oct. 1987.

[3] A.R. Conn, I.M. Elfadel, W.W. Molzen Jr., P.R. O'Brien, P.N. Strenski, C. Visweswariah, C.B. Whan, "Gradient-Based Optimization of Custom Circuits Using a Static-Timing Formulation," *Design Automation Conference*, June 1999.

[4] S. Boyd, S.J. Kim, D. Patil, and M. Horowitz, "Digital Circuit Sizing via Geometric Programming," to appear in *Operations Research*, 2005.

[5] K. Nose and T. Sakurai, "Optimization of V_{DD} and V_{TH} for Low-Power and High-Speed Applications," *Design Automation Conference*, Jan. 2000.

[6] N. Zhang and R. Brodersen, "The cost of flexibility in systems on a chip design for signal processing applications," http://bwrc.eecs.berkeley.edu/Classes/EE225C/Papers/arch_design.doc, 2002.

[7] "Advanced Configuration and Power Interface Specification," Hewlett-Packard Corp., Intel Corp., Microsoft Corp., Phoenix Technologies Ltd., and Toshiba Corp., http://www.acpi.info/DOWNLOADS/ACPIspec30.pdf, Sept. 2004.

[8] A.P. Chandrakasan, S. Sheng, and R.W. Broderson, "Low-Power CMOS Digital Design", *IEEE Journal of Solid-State Circuits*, April 1992.

[9] L. Wei, Z. Chen; K. Roy, Y. Ye, V. De, "Mixed-V_{th} (MVT) CMOS Circuit Design Methodology for Low Power Applications," *Design Automation Conference*, June 1999.

[10] Y. Shimazaki, R. Zlatanovici, and B. Nikolic, **"**A Shared-Well Dual-Supply-Voltage 64-bit ALU," *IEEE Journal of Solid-State Circuits*, March 2004.

[11] M.J.M. Pelgrom, A.C.J. Duinmaijer, and A.P.G. Welbers, "Matching Properties of MOS Transistors," *IEEE Journal of Solid-State Circuits*, Oct 1989.

[12] D. Patil, S. Yun, S.-J. Kim, A. Cheung, S. Boyd, and M. Horowitz, "A New Method for Design of Robust Digital Circuits," *International Symposium on Quality Electronic Design*, March 2005.

[13] X. Bai, C. Visweswariah, P.N. Strenski, and D.J. Hathaway, "Uncertainty-Aware Circuit Optimization," *Design Automation Conference*, June 2002.

[14] J.W. Tschanz, S.G. Narendra, Y. Ye, B. A. Bloechel, S. Borkar, and V. De, "Dynamic Sleep Transistor and Body Bias for Active Leakage Power Control of Microprocessors", *IEEE Journal of Solid-State Circuits*, Nov. 2003.

[15] A. Keshavarzi, S. Ma, S. Narendra, B. Bloechel, K. Mistry, T. Ghani, S. Borkar, and V. De, "Effectiveness of Reverse Body Bias for Leakage Control in Scaled Dual Vt CMOS ICs," *International Symposium on Low Power Electronic Design*, Aug. 2001.

[16] T. Chen and S. Naffziger, "Comparison of Adaptive Body Bias (ABB) and Adaptive Supply Voltage (ASV) for Improving Delay and Leakage under the Presence of Process Variation**",** *IEEE Trans. on Very Large Scale Integration Systems*, Oct. 2003.

[17] J.W. Tschanz, J.T. Kao, S.G. Narendra, R. Nair, D.A. Antoniadis, A.P. Chandrakasan, and V. De, "Adaptive Body Bias for Reducing Impacts of Die-to-Die and Within-Die Parameter Variations on Microprocessor Frequency and Leakage," *IEEE Journal of Solid-State Circuits*, Nov. 2002.

[18] T. Kuroda, K. Suzuki, S. Mita, T. Fujita, F. Yamane, F. Sano, A. Chiba, Y. Watanabe, K. Matsuda, T. Maeda, T. Sakurai, and T. Furuyama, "Variable Supply-Voltage Scheme for Low-Power High-Speed CMOS Digital Design," *IEEE Journal of Solid-State Circuits*, Mar. 1998.

[19] K.J. Nowka, G.D. Carpenter, E.W. MacDonald, H.C. Ngo, B.C. Brock, K.I. Ishii, T.Y. Nguyen, J.L. Burns, "A 32-bit PowerPC System-on-a-Chip With Support for Dynamic Voltage Scaling and Dynamic Frequency Scaling," *IEEE Journal of Solid-State Circuits*, Nov. 2002.

[20] S. Akui, K. Seno, M. Nakai, T. Meguro, T. Seki, T. Kondo, A. Hashiguchi, H. Kawahara, K. Kumano, and M. Shimura, "Dynamic Voltage and Frequency Management for a Low-Power Embedded Microprocessor," *IEEE International Solid-State Circuits* Conference, Feb. 2004.

[21] T. Fischer, F. Anderson, B. Patella, and S. Naffziger, "A 90nm Variable-Frequency Clock System for a Power-Managed Itanium®-Family Processor," *IEEE International Solid-State Circuits Conference*, Feb. 2005.

[22] S. Das, S. Pant, D. Roberts, S. Lee, D. Blaauw, T. Austin, T. Mudge, and K Flautner, "A Self-Tuning DVS Processor Using Delay-Error Detection and Correction," *IEEE Symposium on VLSI Circuits*, June 2005.

[23] B.H. Calhoun, A. Wang, and A. Chandrakasan, "Modeling and Sizing for Minimum Energy Operation in Subthreshold Circuits," *IEEE Journal of Solid-State* Circuits, Sept. 2005.

[24] L. Nazhandali, B. Zhai, J. Olson, A. Reeves, M. Minuth, R. Helfand, S. Pant, T. Austin, and D. Blaauw, **"**Energy Optimization of Subthreshold-Voltage Sensor Network Processors," *International Symposium on Computer Architecture*, June 2005.

[25] K. Chang, S. Pamarti, K. Kaviani, E. Alon, X. Shi, T.J. Chin, J. Shen, G. Yip, C. Madden, R. Schmitt, C. Yuan, F. Assaderaghi, and M. Horowitz, "Clocking and Circuit Design for a Parallel I/O on a First-Generation CELL Processor," *IEEE International Solid-State Circuits* Conference, Feb. 2005.

[26] R. Ho, K. Mai, and M. Horowitz, "Efficient On-Chip Global Interconnects**,"** *IEEE Symposium on VLSI Circuits*, June 2003.

[27] D. Schinkel, E. Mensink, E. Klumperink, E. van Tuijl, and B. Nauta, "A 3Gb/s/ch Transceiver for RC-Limited On-Chip Interconnects," *IEEE International Solid-State Circuits Conference*, Feb. 2005.

[28] D. Pham, *et. al.*, "The Design and Implementation of a First-Generation CELL Processor," *IEEE International Solid-State Circuits Conference*, Feb. 2005.

[29] B. Fu, A. Saini, and P.P. Gelsinger, "Performance and Microarchitecture of the i486 Processor," *IEEE International Conference on Computer Design: VLSI in Computers and Processors*, Oct. 1989.

[30] C.N. Keltcher, K.J. McGrath, A. Ahmed, and P. Conway, "The AMD Opteron Processor for Multiprocessor Servers," *IEEE Micro*, March 2003.

[31] "LongRun2 Technology," Transmeta Corp., http://www.transmeta.com/longrun2/ .

[32] C. Poirer, R. McGowen, C. Bostak, and S. Naffziger, "Power and Temperature Control on a 90nm Itanium®-Family Processor," *IEEE International Solid-State Circuits Conference*, Feb. 2005.

[33] V. Gutnik, and A.P. Chandrakasan, "Embedded Power Supply for Low-Power DSP," *IEEE Trans. on Very Large Scale Integration Systems*, Dec. 1997.