DesignCon 2003
System-on-Chip and ASIC Design Conference

A Combined Hardware-Software Approach for Low-Power SoCs: Applying Adaptive Voltage Scaling and Intelligent Energy Management Software

Krisztián Flautner
David Flynn

ARM Limited

Mark Rives

National Semiconductor Corporation

Abstract

Increased functionality and performance demands are challenging System-on-Chip (SoC) designers to seek better methods for optimizing available battery power in portable applications. Key areas of exploration include dynamic voltage scaling and improved software algorithms for the control of power modes. Dynamic voltage scaling can be improved by adaptively monitoring hardware performance to minimize the applied supply voltage for any given clock frequency. While adaptive voltage scaling optimizes power use based on temporal environmental conditions, Intelligent Energy Management (IEM) algorithms optimize power consumption based on the dynamic workload of the processor. IEM software and hardware monitor the execution and communication characteristics of workloads and predictively set the performance of the processor to the level that minimizes energy use, while still meeting application deadlines. The combined use of adaptive voltage scaling and IEM provides the optimum trade-off between performance and battery life for portable devices.

Authors' Biography

Krisztián Flautner, Principal Research Engineer, ARM, holds a Ph.D. degree in Computer Science and Engineering from the University of Michigan. His thesis explored the relevance of multithreading for interactive desktop workloads and described the implementation of an automatic power-management algorithm for processors supporting dynamic voltage scaling. Dr. Flautner's research interests are focused on simple ideas that enable high-performance low-power computers to support advanced software environments. In the research group at ARM Limited, he is currently working on the next generation ARM architecture.

David Flynn has been with ARM for 11 years and is a Fellow in the Research and Development group based in Cambridge, UK, specializing in System-on-Chip IP deployment and methodology. He is the original architect behind ARM's synthesizable CPU family and the AMBA on-chip interconnect standard. His current research focus is low-energy system-level design. He holds a number of patents in on-chip bus, low power and embedded processing sub-system design (8 US, 21 worldwide) and has a BSc (1st) in Computer Science from Hatfield Polytechnic, UK.

Mark Rives, Principal Applications Engineer, National Semiconductor Corporation, joined National in 1995 where he is responsible for supporting advanced development projects in the Portable Power Group. His previous experience includes the design and support of National's direct IF-sampling Diversity Receiver Chipset in addition to a wide range of both system level and IC design projects from pro audio gear to pacemakers. He received his B.S. in Electrical Engineering from Mississippi State University in 1987.

Introduction

Low power consumption is arguably the most important feature of embedded processors, which significantly impacts the cost and physical size of the end device. Even though the processor may not be the most power-hungry component of a system, it is essential to manage processor power in order to reduce overall system power consumption. Better processor power efficiency can increase the available power budget for features such as color screens and backlights, which are growing in popularity on portable devices.

Historically, low power consumption in embedded processors has been achieved through simple designs, limited use of speculation, and employing a number of low-power sleep modes that reduce idle-mode power consumption. Embedded processors are now performing more sophisticated tasks, which require ever-higher performance levels. As a result, new processor designs are more dependent on sophisticated architectural techniques (such as prediction and speculation) to achieve high performance. Unfortunately, such techniques can also significantly increase the processor's power consumption.

Process technology trends are also complicating the power story. Until recently, CMOS transistors consumed negligible amounts of power under static conditions. However, as process geometries shrink to provide increasing speed and density, their static (leakage) power consumption has also increased. Current estimates suggest that static power accounts for about 15%-20% of the total power on chips implemented in 0.13µm high-speed processes. Moreover, as process technology moves below 0.1µm, static power consumption is set to increase exponentially, and will soon dominate the total power consumed by the processor.

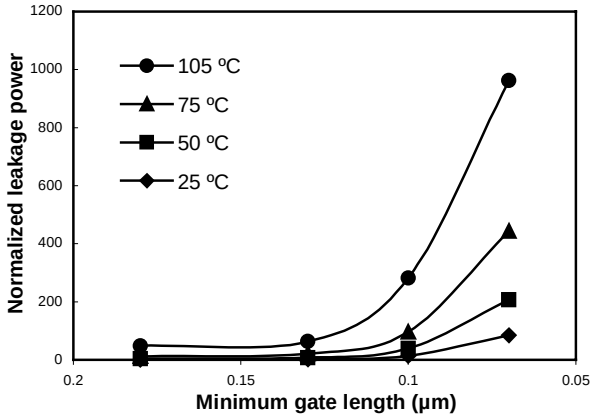

Figure 1. Normalized leakage power through an inverter

The circuit simulation parameters including threshold voltage were obtained from the Berkeley Predictive Spice Models [1]. The leakage power numbers were obtained by HSPICE simulations.

Figure 1 shows projections for leakage power increase in future process technologies. There is a strong correlation between the operating temperature and the amount of leakage power. However, regardless of the temperature, all lines exhibit exponential trends. In embedded processors, where the majority of transistors are usually dedicated to

memory structures (such as caches), leakage is a particularly important problem to attack, since the static power consumption of these structures can dominate overall power consumption.

Power Saving Opportunities

A way to bridge the gap between high performance and low power is to allow the processor to run at different performance levels depending on the current workload. An MPEG video player, for example, requires about an order of magnitude higher performance than an MP3 audio player. Even greater savings can be achieved by reducing the processor's supply voltage as the clock frequency is reduced. Dynamic Voltage Scaling (DVS) exploits the fact that the peak frequency of a processor implemented in CMOS is proportional to the supply voltage, while the amount of dynamic energy required for a given workload is proportional to the square of the processor's supply voltage [2]. Reducing the supply voltage while slowing the processor's clock frequency yields a quadratic reduction in energy consumption, at the cost of increased run time.

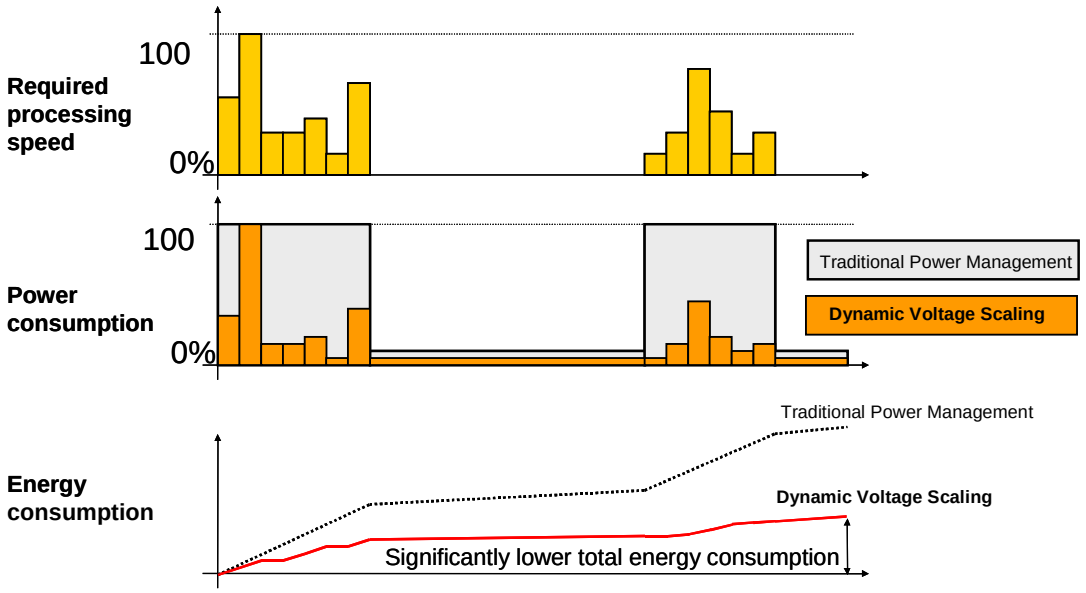

Figure 2. Traditional Power Management vs. Dynamic Voltage Scaling

Often, the processor is running too fast. For example, it is pointless from a quality-of-service perspective to decode the 30 frames of a video in half a second, when the software is only required to display those frames during a one second interval. Completing a task before its deadline is an inefficient use of energy [3]. The key to taking advantage of this trade-off is the use of performance-setting algorithms that aim to reduce the processor's performance level (clock frequency) only when it is not critical to meet the application's deadlines. Figure 2 illustrates a significantly lower total energy consumption using dynamic voltage scaling compared with traditional gated-clock power

management, for the same workload. Note that with DVS, the lower supply voltage reduces static power even when the clock is gated off.

Static leakage power can also be substantially reduced if the processor does not always have to operate at its peak performance level. One technique for accomplishing this is adaptive reverse body biasing (ABB). Combined with dynamic voltage scaling, this can yield substantial reductions in both leakage and dynamic power consumption [4]. The key enabler for controlling both DVS and ABB is knowledge about how fast a given workload needs to run. This information can be provided by performance-setting algorithms that take various operating system and optional application-specific information into account to provide an estimate for the necessary performance level of the processor.

Just as performance-setting algorithms optimize power consumption based on workload variations, significant power efficiency can also be gained if the processor does not have to operate under worst-case assumptions but can tune its operating parameters to temporal environmental conditions [5]. Processors are designed to operate reliably over a wide range of temperature levels and variations of the silicon substrate. Increased voltage levels must be used to assure the large safe-operating range at the cost of reduced power efficiency. By monitoring the margin between expected and actual operating conditions, the voltage level of the processor can be reduced without sacrificing operational stability. This closed-loop monitoring of system margin will be referred to as adaptive voltage scaling (AVS).

While DVS, AVS, and ABB are effective ways of managing the processor's power consumption, integrating these ideas into SoC designs has proven to be a significant challenge. The key issue is that not all parts of the SoC can be scaled in equal measure. Consequently, multiple voltage and frequency domains with asynchronous interfaces are required. Moreover, these extra parameters complicate testing and validation processes and require special support from synthesis tools.

System-on-Chip Implementation

Figure 3 depicts the system architecture required to implement the systemic power reduction schemes in a SoC design combining Intelligent Energy Management (IEM) with Adaptive Power Control (APC). The IEM interfaces with the CPU though the AMBA Peripheral Bus allowing it to be easily added to any ARM-based SoC design.

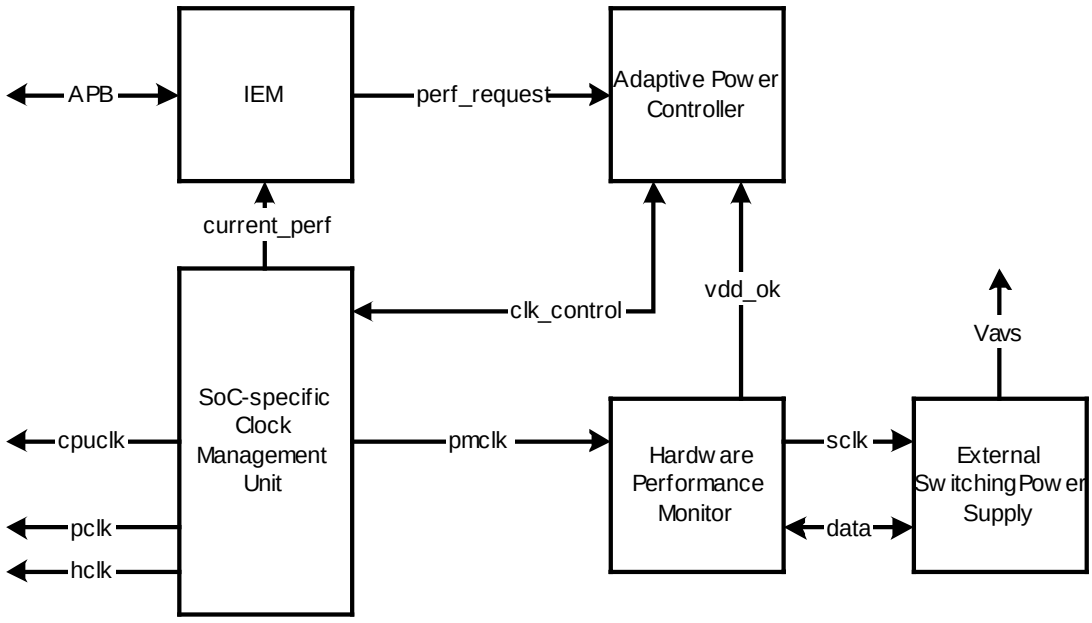

Figure 3. IEM + AVS Architecture

The IEM software and hardware monitor the system workload to generate a performance request. The APC can then set the correct operating voltage in either open-loop or closed-loop mode without processor intervention. The APC will transparently provide the fastest possible response while assuring that the processor will always receive the minimum safe operating voltage for any given clock frequency. The APC would also coordinate all clock switching including the verification of stable supply voltage. The IEM provides a uniform software interface to simplify implementation and reuse. The APC provides an open-standard interface to the external power supply.

ARM Limited and National Semiconductor Corporation have agreed to work together to offer synthesizable intellectual property (IP) to implement the IEM and AVS functionality for SoC designers. Work is underway to assure support from both design tool vendors and operating system vendors to allow SoC designers to implement this power saving technology transparently. The following sections provide more detail on the key components of the solution.

Intelligent Energy Management

Completing a task before its deadline, and then idling, is significantly less energy efficient than running the task more slowly so that the deadline is met exactly. The goal is to reduce the performance level of the processor without allowing applications to miss their deadlines. The central issue is how the right level of performance can be predicted for the application.

The Intelligent Energy Management (IEM) framework provides a hardware and software mechanism for achieving these goals: it standardizes the interface for setting the processor's performance level, specifies counters for measuring the amount of work that is being accomplished, and includes operating system and application-level algorithms for predicting future behavior.

The IEM software layer has the ability to combine the results of multiple algorithms and arrive at a single global decision. The policy stack illustrated in Figure 4 supports multiple independent performance-setting policies in a unified manner. The primary reason for having multiple policies is to allow the specialization of performance-setting algorithms to specific situations, instead of having to make a single algorithm perform well under all conditions. The policy stack keeps track of commands and performance-level requests from each policy and uses this information to combine them into a single global performance-level decision when needed.

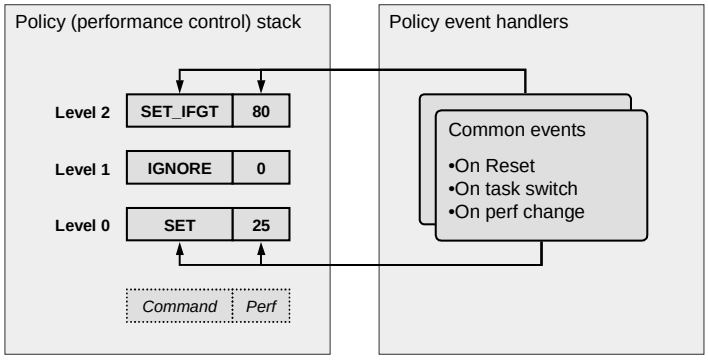

Figure 4. Performance policy stack

The different policies are not aware of their positions in the hierarchy and can base their performance decisions on any event in the system. When a policy requests a performance level, it submits a command along with its desired performance to the policy stack. The command specifies how the requested performance should be combined with requests from lower levels on the stack: it can specify to ignore (*IGNORE*) the request at the current level, to force (*SET*) a performance level without regard to any requests from below, or set a performance level only if the request is greater than anything below

(*SET_IFGT*). When a new performance level request arrives, then the commands on the stack are evaluated bottom-up to compute the new global performance level. In Figure 4, the evaluation would yield the following: at level 0 the global prediction is set to 25, at level 1 it remains at 25, and level 2 changes the prediction to 80.

Using this system, performance requests can be submitted any time and a new result computed without explicitly having to invoke all the performance-setting policies. While policies can be triggered by any event in the system and they may submit a new performance request at any time, there are sets of common events of interest to all. On these events, instead of recomputing the global performance level each time a policy modifies its request, the performance level is computed only once after all interested policies' event handlers have been invoked. Currently the set of common events are: reset, task switch, task create, and performance change. The performance change event is a notification which is sent to each policy and does not usually cause any changes to the performance requests on the stack.

There are significant benefits in using multiple performance-setting policies, each optimized for a particular situation, instead of a single one that needs to be optimal under all circumstances. Figure 5 provides some qualitative insight into the characteristics of the IEM algorithm vs. LongRun, a conventional algorithm implemented in the Crusoe processor's firmware. The biggest difference between the two algorithms is that while LongRun keeps on ramping the performance level up and down in fast succession, the IEM algorithm stays close to a target performance level.

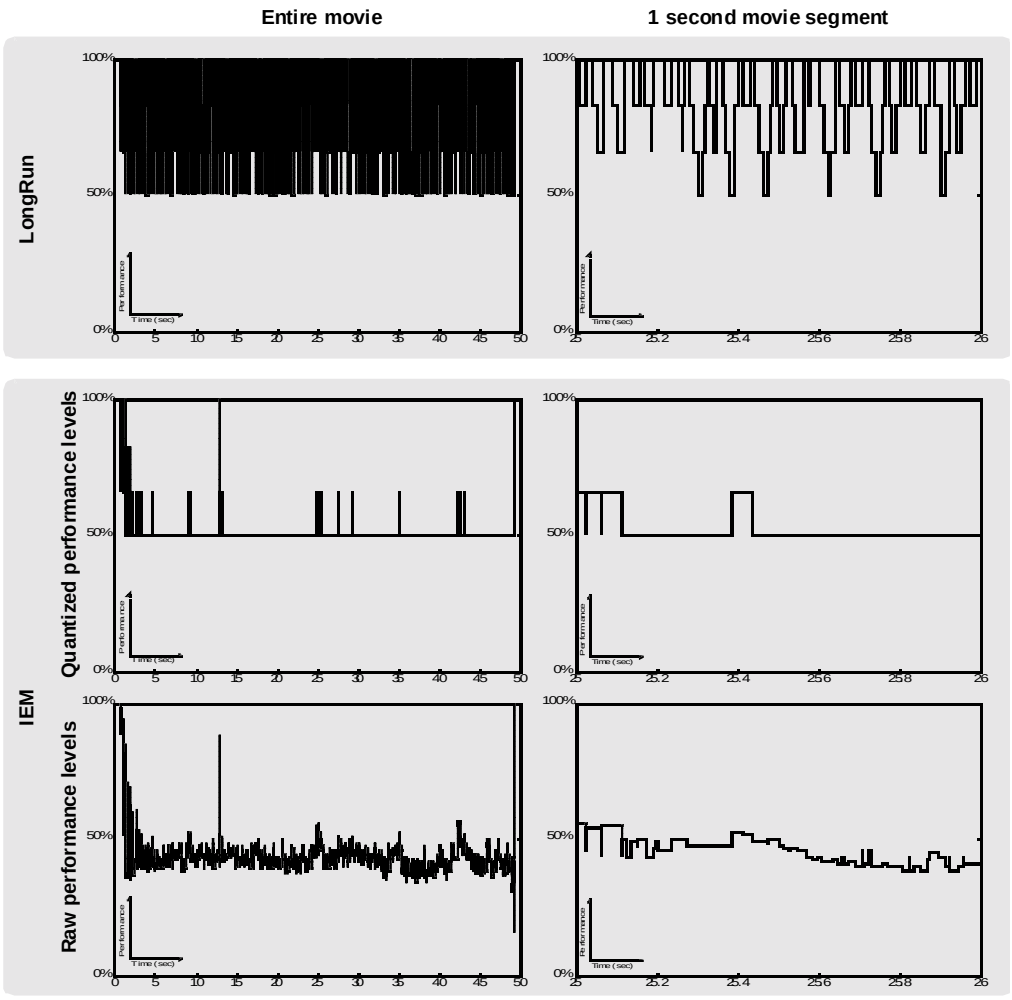

Figure 5. Performance-setting during MPEG video playback of Red's Nightmare

To achieve the most effective energy reduction with minimal intrusion, application monitoring and performance-setting decisions need operating system involvement.

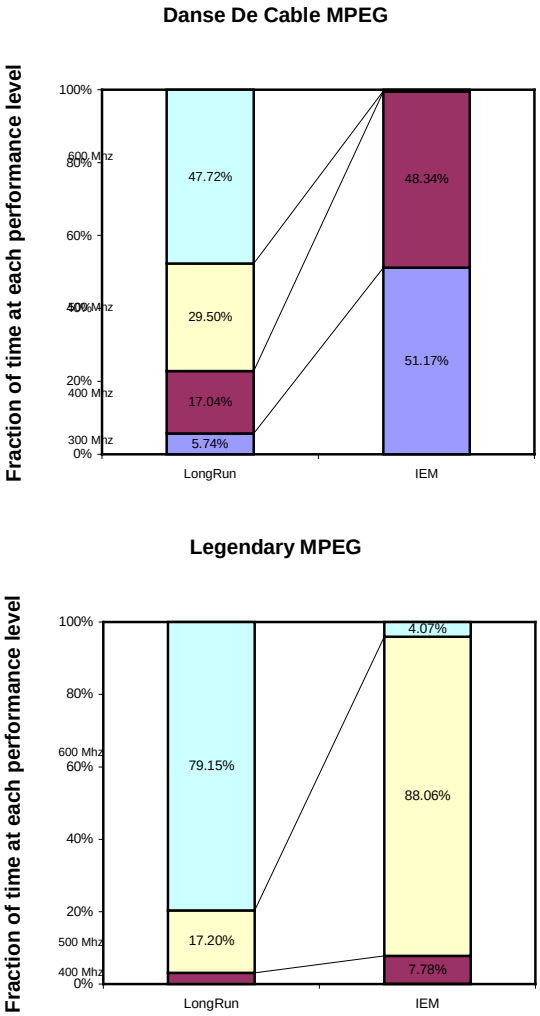

Figure 6. MPEG video playback LongRun vs. IEM

Figure 6 illustrates the fraction of time spent at each of the processor's four performance levels (300, 400, 500, and 600 MHz) using the Crusoe's built-in LongRun power manager, contrasted with IEM during playbacks of two MPEG movies. The data for both algorithms were collected on the same hardware. However, during the IEM measurements, the built-in LongRun power manager was disabled. While the playback quality of the different runs was identical, it can be seen that IEM spends significantly more time below peak performance than LongRun. During the first movie, IEM switches mostly between two performance levels. The machine's minimum 300 MHz and 400 MHz clock frequencies are sufficient for the first movie, while during the second, it settles on the processor's third performance level at 500 MHz. LongRun, on the other hand, chooses the machine's peak performance setting for the dominant portion of execution time during both movies.

Voltage Scaling Methods and Benefits

Currently, proprietary dynamic voltage scaling (DVS) solutions offer improved performance by reducing the supply voltage as the clock frequency is reduced. Open-loop DVS, as shown in Figure 7, allows the processor to set the supply voltage based on a table of frequency/voltage pairs. This table must be determined by characterization to assure sufficient margin for all operating conditions and process corners.

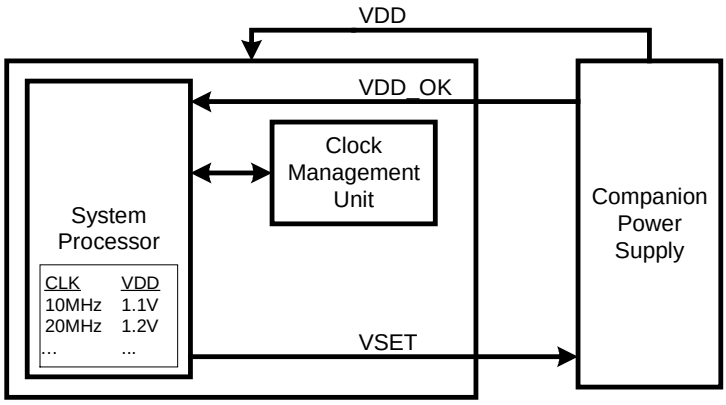

Figure 7. Proprietary Open-Loop DVS

In operation, the processor must determine the desired operating frequency, request a new voltage, wait for the voltage to stabilize, and then switch itself to the new frequency. The switch may be made immediately when changing from a higher frequency to a lower frequency. When switching from a lower frequency to a higher frequency, the power supply voltage must be high enough to support the new frequency prior to changing the clock. Power supply stability can be assured either by a time delay or by an analog measurement. Use of a time delay is risky, since there will always be a desire to implement the minimum possible delay for enhanced processor response time.

Open-loop operation can be simplified by creating an Adaptive Power Controller (APC) module to off-load the voltage scaling and clock management from the processor. The APC approach supports a common software API allowing the DVS function to be easily accessed by applications or the operating system. Providing a standard interface to the external power supply also simplifies system design and facilitates second-source options for the power supply component. An architecture using an APC is shown in Figure 8.

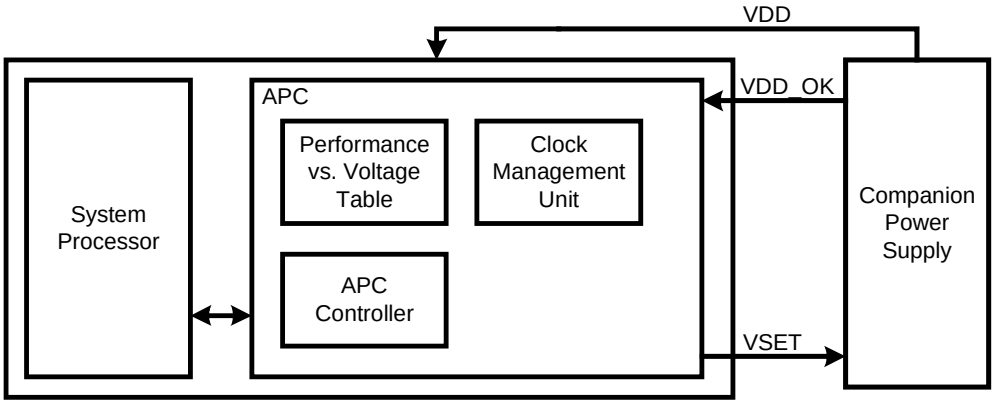

Figure 8. Vendor Independent Open-Loop DVS

Closed-loop or Adaptive Voltage Scaling (AVS) is a new approach, which offers improved performance and ease of implementation compared to open-loop DVS. In the closed-loop system shown in Figure 9, the voltage is set automatically by monitoring the system's performance margin and adjusting the supply voltage adaptively.

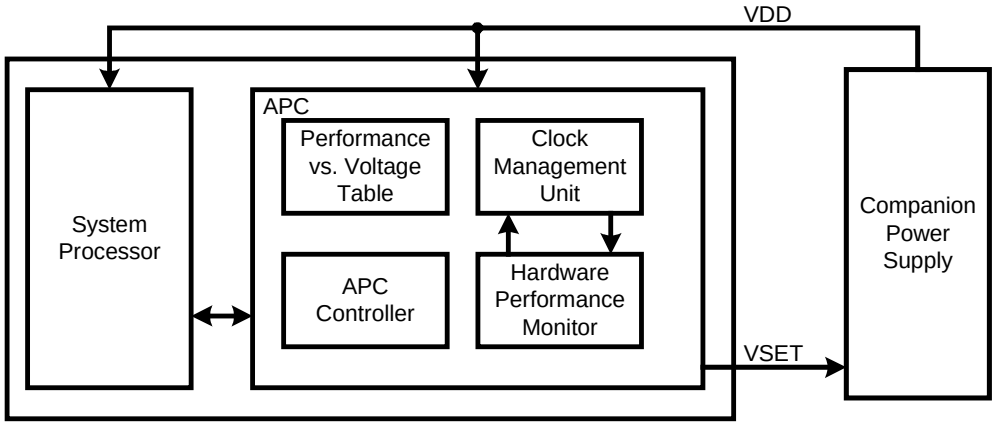

Figure 9. Closed-Loop AVS

Since the system is closed-loop in nature, a much finer degree of control over the voltage is possible when compared to the discrete table values in an open-loop system. Response time of the AVS system can be much faster, since it is limited only by the external power supply. The performance measuring circuitry can be used to verify the power supply stability to offer the fastest possible switching from one clock frequency to the next.

Closed-loop operation also offers improved power savings since the operating voltage margin may be reduced due to the continuous voltage updates. Any temperature effects are inherently compensated by the necessary change in supply voltage. This allows the AVS-equipped SoC to be operated at a lower voltage at room temperature since the voltage will be increased automatically as the temperature increases.

For example, in a 1.8V system with +/-5% tolerance, the system must operate at 85C and 1.71V. For a 200mA load, this equates to 342mW. Even though the system will operate at a lower voltage at 25C (say 1.5V), normally at least 1.71V must be provided to assure 85C operation. With closed-loop AVS, the system can be run on 1.5V at 25C with no problems since the AVS technology will increase the voltage if necessary. This allows a 25C power of 300mW, a saving of 42mW or 14%, even at the maximum clock frequency.

Figure 10 shows measured data for a closed-loop AVS system running at 32MHz, 16MHz, 8MHz, and 4MHz. The plot shows voltage vs. time where the highest voltage is associated with 32MHz operation. The three traces show how the closed-loop AVS voltage is automatically adjusted with changing temperature.

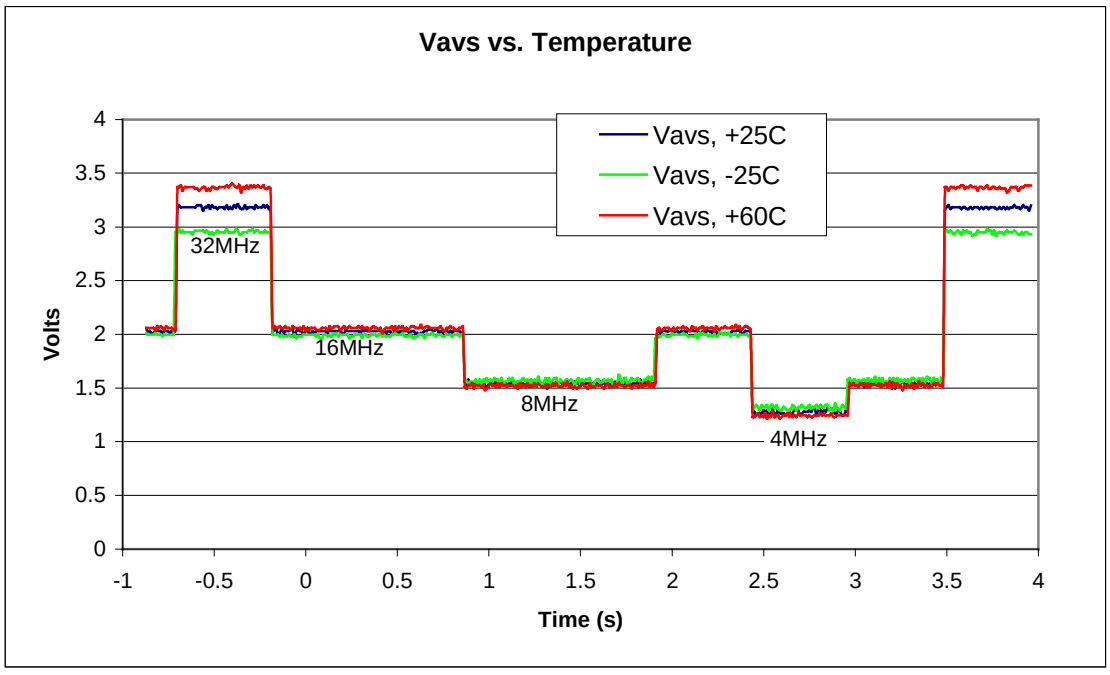

Figure 10. AVS Voltage vs. Temperature

Figure 11 compares the power used by a system with a fixed 3.3V supply with the power used with AVS. The effect of reduced margin at 25C can be seen in reduced power, even at the highest clock frequency.

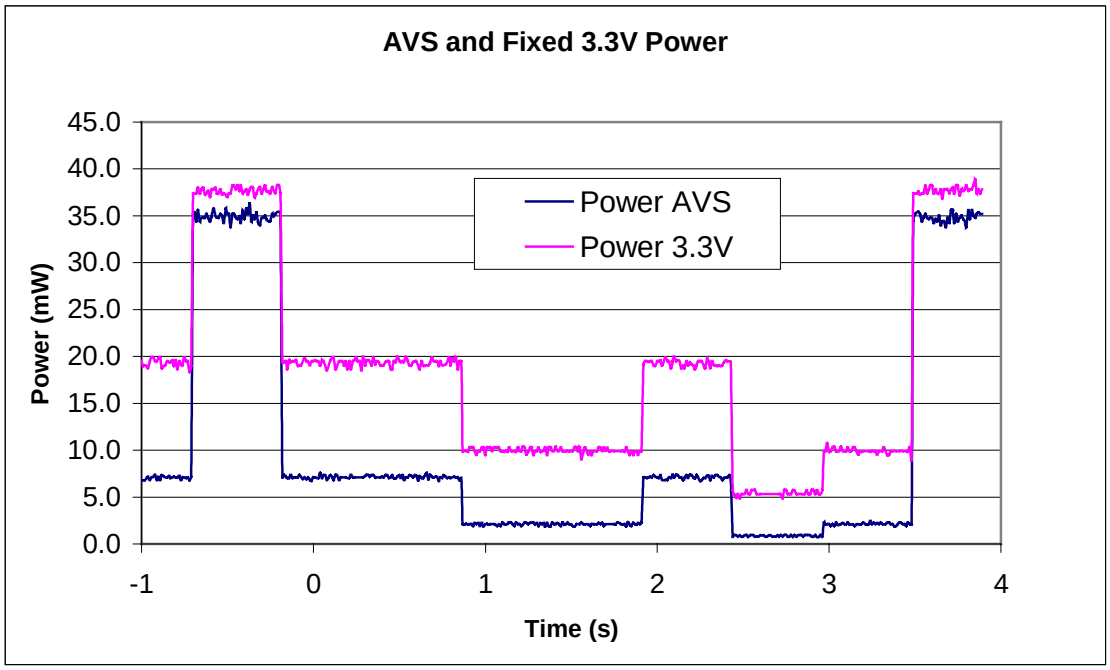

Figure 11. Power Reduction with AVS

Figure 12 shows the percentage of power used by the AVS system relative to that used by the fixed 3.3V system. As the clock frequency is decreased, the power savings provided by AVS can increase to 80% or more when using a switching regulator for the power supply.

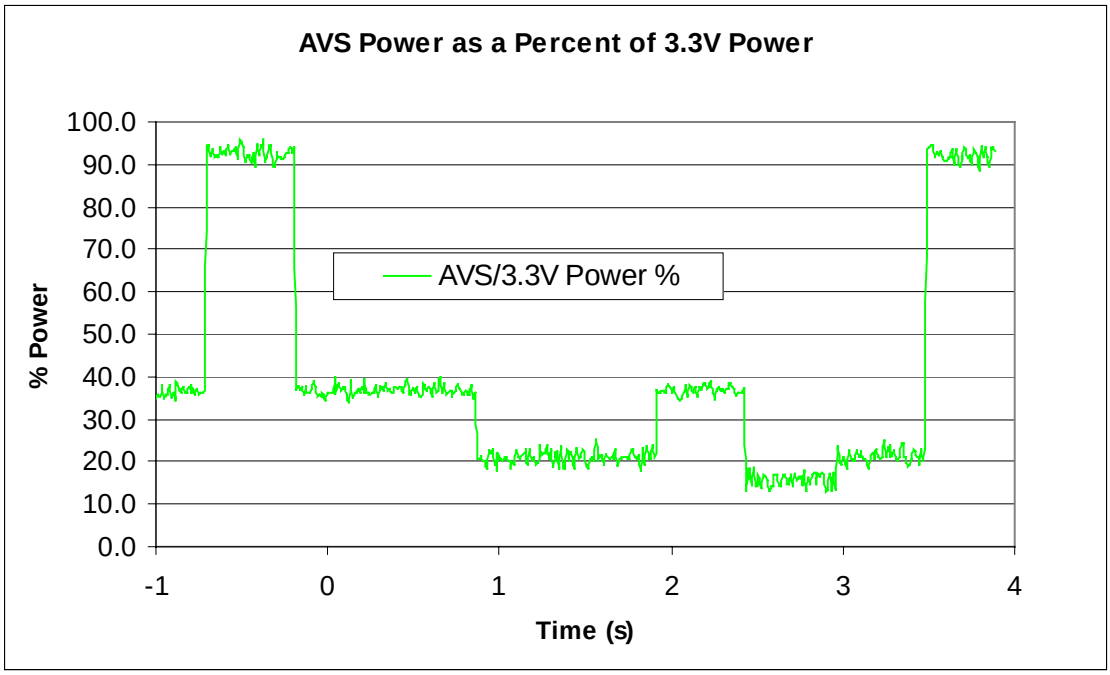

Figure 12. Relative Power Usage of AVS vs. a 3.3V Fixed Supply

SoC Design Flow Issues

Until now, dynamic voltage scaling has only been commercially exploited in stand-alone CPU integrated circuits. To support voltage scaling of processing sub-systems within a system-on-chip design requires enhancements to both EDA tools and design methodology. Key issues include:

- Multiple physical power domains
- Synchronous clock relationships across boundaries
- Standard-cell library and RAM compiler design views
- Static timing verification
- Manufacturing test

Multiple power domains require careful handling at interfaces where some form of analog level-shifting is required between different voltages. Also, many EDA tools treat voltage rails as special global resources which are implicitly connected, which makes separating voltage domains a manually intensive design step.

Best-practice SoC design flows typically assume synchronous clocking relationships between sub-systems in order to allow top-level static timing-closure and analysis, automatic test structure insertion and test pattern generation. Ideally, multiple voltage domains should be treated as asynchronous because the tolerancing of buffered clocks across the top-level system becomes near-impossible where sub-systems can have variable voltage with respect to each other. Different sub-systems have inherently variable clock buffer latencies.

Cell libraries and memory compilers are normally characterized and modeled for a process and temperature range acrossa tightly toleranced (+/- 5% to +/- 10%) supply voltage. To ensure design integrity with voltage scaling, more comprehensive timing models are required. The design tools make this harder because in order to design for multiple performance levels, the target sub-system frequency must first be specified. Then, the power supply requirements must be determined to provide sufficient voltage to maintain operation, either statically or adaptively. However, from a design-flow perspective, it is necessary to work the other way around: start with a defined voltage and then calculate the achievable performance from the static timing analysis at this precise voltage. Characterizing RAMs at low voltage is complicated by the fact that sense amplifier performance degrades non-linearly with respect to logic gate speeds.

Verification of static timing and functional test are complicated with voltage scaling of parts of the SoC design. The EDA tools need to be guided.

Implications for Front-end Design

Front-end design tools typically read in RTL descriptions in Verilog or VHDL of the hardware design, for both simulation and synthesis. Such HDL descriptions have no concept of multiple power rails; a global view of power and ground are assumed.

Similarly, clocks and resets are treated as ideal signals in the HDL. These are later buffered as carefully balanced high-fanout buffer-tree networks.

The boundaries between voltage domains must be handled with detailed management of hierarchy and the instantiation of explicit voltage level-shifter cells between different voltage rails. The onus is on the designer to carefully abstract out the top-level management of clocks, resets, test scan chains and power management such that individual sub-systems can be synthesized and even hardened independently using standard ASIC design flows.

Implication for Back-end Design

The layout tools need to understand separate voltage rails and this may require manual intervention and careful inspection and review of conversion from the front-end logical design flow to the place and route implementation phase.

In the worst-case, the cell library may need to be replicated with special cell and power-rail naming schemes to ensure that optimization, setup and hold timing fixes applied to the post-routed top-level design do not accidentally stray over voltage domains or level-shifter boundaries.

Design verification needs to be extended beyond standard ASIC design flows to cover the extra complication of analog level-shifter integrity. This is especially relevant to power domains that can be powered off completely. These must not draw static currents from driven inputs, and need outputs clamped during power down and power up (i.e. operating outside valid logic state operation).

An ARM926EJ-S based design with independent voltage scaling of the cached CPU, which tackles all these design tool issues is scheduled for fabrication in February 2003.

Conclusions

The ARM Intelligent Energy Manager (IEM) provides continuous predictive monitoring of the CPU workload. It attempts to run the clock frequency at the lowest available value while still completing the work prior to its deadline. The correct performance level is set by predictive algorithms that are embedded in the operating system kernel to monitor all processes.

National Semiconductor Corporation's AVS technology accepts the IEM's performance request and sets the lowest possible operating voltage for any resulting clock frequency. Sufficient margin is always present to assure proper operation. Since National's hardware performance monitor is always adjusting the voltage for sufficient margin at any given clock frequency, the effects of process and temperature variation are inherently corrected. If the temperature rises, the margin will decrease and the voltage will be increased to compensate.

The combination of these two technologies will provide optimum power savings for embedded processors in portable systems.

The proposed dynamic voltage scaling system also forms the basis for techniques that address a chip's static (leakage) power consumption. Some of our initial investigations are described in [4].

References

[1] http://www-device.eecs.berkeley.edu

[2] T. Mudge. Power: A First Class Architectural Design Constraint. IEEE Computer, vol. 34, no. 4, April 2001.

[3] K. Govil, E. Chan, and H. Wasserman. Comparing Algorithms for Dynamic Speed-Setting of a Low-Power CPU. *Proceedings of the First International Conference on Mobile Computing and Networking*, November 1995.

[4] S. Martin, K. Flautner, D. Blaauw, and T. Mudge. Combined Dynamic Voltage Scaling and Adaptive Body Biasing for Optimal Power Consumption in Microprocessors under Dynamic Workloads. *Proceedings of the International Conference on Computer Aided Design (ICCAD 2002)*, San Jose, CA, November 2002.

[5] S. Dhar, D. Maksimovic, and B. Kranzen. Closed-loop Adaptive Voltage Scaling Controller for Standard-cell ASICs. *Proceedings of the 2002 International Symposium on Low-Power Electronics and Design (ISLPED 2002).* Monterey, CA, August 2002.