System power management support in the IBM POWER6 microprocessor

M. S. Floyd
S. Ghiasi
T. W. Keller
K. Rajamani
F. L. Rawson
J. C. Rubio
M. S. Ware

The IBM POWER6™ microprocessor chip supports advanced, dynamic power management solutions for managing not just the chip but the entire server. The design facilitates a programmable power management solution for greater flexibility and integration into system- and data-center-wide management solutions. The design of the POWER6 microprocessor provides real-time access to detailed and accurate information on power, temperature, and performance. Together, the sensing, actuation, and management support available in the POWER6 processor, known as the EnergyScale™ architecture, enables higher performance, greater energy efficiency, and new power management capabilities such as power and thermal capping and power savings with explicit performance control. This paper provides an overview of the innovative design of the POWER6 processor that enables these advanced, dynamic system power management solutions.

Introduction

With rising energy costs and technology trends predicting increased power densities and variability in power consumption, power management considerations now significantly affect the reliable operation and performance of server systems. Meanwhile, users have diverse requirements for a highly reliable, trouble-free computing infrastructure, low energy consumption, and high levels of performance, in which the dominant requirement can change over time. Consequently, server systems must provide greater flexibility in order to adapt to varying hardware components, environmental conditions, workloads, and customer requirements for energy savings and performance.

Energy efficiency of server-class computers is becoming increasingly important to customers, partly because of the recent higher cost of energy and worldwide energy shortages. As they update their data centers, customers are also discovering that existing power delivery and air conditioning systems are inadequate for the newer, more-power-hungry systems, leaving them with the unpleasant choice of either limiting computing performance or spending unacceptably large amounts of money to upgrade or build new facilities in order to handle the increasing power and thermal loads. This dilemma is driven by ever-increasing computer power densities, caused by steadily shrinking components using faster technologies and clock speeds. Now, the principal design constraint to system performance is delivering power to and dissipating heat from various system components, especially microprocessor and memory subsystem chips. Since servers have fixed power and cooling budgets (e.g., a 100-W processor socket in a rack-mounted system), the frequency and/or throughput of the chips must be fixed to a point well below their capability, sacrificing sizable amounts of performance, even when a non-peak workload is running or the thermal environment is favorable.

Fixing the operational parameters of the server to static, pre-runtime values in order to meet fixed power supply and cooling budgets has become an increasingly losing proposition, since chips, in turn, are forced to operate at significantly below their runtime capabilities because of a cascade of effects. With technology scaling, there is increasing variability in the power and performance characteristics of "identical" chips

0018-8646/07/$5.00 © 2007 IBM

manufactured under identical processes [1]. Using static, pre-runtime operational parameters demands more provisioning of power and cooling in the system in order to meet economically viable chip manufacturing yields.

The same is true for variability in the thermal operating environment. Some data centers run with a cooler ambient temperature than others. In addition, some data centers are at sea level, where the air is dense and more effective at dissipating heat. Some servers may have an acoustic limit required by the customer, which reduces the amount of available airflow in some cases. While some alternative solutions such as water cooling can reduce these thermal concerns, there is a reluctance to use these more expensive alternatives. Additionally, there is often significant runtime variability in the power consumption and temperature because of natural fluctuations in system utilization and type of workloads being run. The now-common usage of special circuit technologies such as clock gating for improved power efficiency makes the power consumption and, therefore, dissipation even more sensitive to workload and utilization. Realistically, servers rarely (if ever) simultaneously run with worst-case workloads at 100% utilization under extremely hot environmental conditions. Therefore, fixing operational parameters to a static, pre-runtime value forces overly conservative (lower power/performance) choices to be determined by such a worst-case scenario, penalizing the performance of almost all systems, workloads, and environments.

There are multiple design requirements for modern servers, including reliable operation, high performance, and lower energy consumption. As described earlier, using statically determined operational parameter limits to ensure reliable operation across a wide range of process, workload, and environmental variations leaves performance adjustable. Higher performance and lower energy consumption are often conflicting requirements for a given design. However, there is usually flexibility when only one of these requirements dominates, e.g., performance requirements may be relaxed for batch workloads or during non-peak hours, or a high-volume, lower-end system design point might tolerate lower performance in exchange for higher energy savings from the same processor chip than a higher-end system design. Building adaptability into the server is the key to avoiding conservative design points in order to accommodate variability and to take further advantage of flexibility in power and performance requirements. A design in which operational parameters are dictated by runtime component, workload, and environmental conditions as well as by the customer's current power versus performance requirement is less conservative and more readily adjusted to the needs of the customer at any given time.

The IBM POWER6* processor is designed exactly with this goal in mind. By providing a high degree of adaptability that supports advanced, dynamic power management techniques, the POWER6 processor EnergyScale* implementation enables higher performance and greater energy efficiency by providing more options to the system designer to dynamically tune it to the exact requirements of the server. The design focuses on enabling feedback-driven control of power and associated performance for robust adaptability to a wide range of conditions and requirements. Explicit focus was placed on developing each of the key elements for such an infrastructure: sensors, actuators, and communications for control. As a result, POWER6 microprocessor-based systems provide an array of capabilities both for monitoring power consumption and environmental and workload characteristics and for controlling a variety of mechanisms in order to realize the desired power/performance trade-offs (i.e., the highest power reduction for a given performance loss). Dedicated communication channels from an external controller and built-in on-chip networks support real-time access to both sensors and actuators for flexible programmable implementations of power management solutions. Together, they enable a wide range of policy-guided power management solutions and completely new power management capabilities such as explicit power capping and an ability to dynamically fine-tune the energy/performance trade-offs.

Figure 1 shows a conceptual diagram of the power management elements in a POWER6 microprocessor-based system. This figure shows some of the sensors (distributed thermal sense resistors on the chip [i.e., thermistors], activity counters in the processor cores and memory hierarchy elements, off-chip circuitry for current and voltage sensing), actuators (on-chip ones in the cores and memory controller and off-chip ones through the programmable oscillator and voltage regulator modules [VRMs]), and communications and control elements in the form of a communications network that links the on-chip elements via I2C (inter-integrated circuit) to the off-chip control element, the *thermal and power management device* [TPMD]).

In this paper, first we discuss features on the processor chip that enable dynamic system power/performance optimization, including sensors, actuators, and operating modes, many of which are unique to the POWER6 microprocessor. Next, we present the support for communications and control for effective system and data-center management enabled by the POWER6 processor. Subsequently, we discuss how some power management capabilities are enabled by the POWER6 processor design, and we briefly describe the interaction between these features, OS (operating system) software, and the hypervisor. Finally, we explain how the power

policy is managed by the system infrastructure and controlled by the customer.

On-chip support for power/performance optimization

Computational performance and processor power consumption are interrelated and must be optimized together in order to achieve new levels of performance at reasonable levels of power consumption. Traditional techniques that increase the performance of the processor also dramatically increase both its power consumption and the density of the heat that it produces. Permanently increasing the processor frequency, for example, requires increasing the processor supply voltage, leading to permanent superlinear increases in power consumption. More desirable are techniques that improve performance without imposing permanent penalties in power by adapting to workload and the environment.

The POWER6 processor supports dynamic power/performance optimizations by providing a variety of control mechanisms, actuators, and sensors via on-chip hardware. The actuators provide the ability to change both the operating and the idle mode of the system components and to regulate the activity in a component to lower its power consumption and thermal dissipation. The power and thermal management solutions for systems using the POWER6 processor adopt a measurement-guided strategy to dynamic power management in order to address the variability in power and thermal characteristics due to process, workload, and environmental factors. The key to any measurement-guided solution is the sensors, which provide real-time measurement feedback.

Since the primary power consumers in server systems are the processors and the memory, the actuators and sensors for the POWER6 processor are specifically designed to address the control and monitoring needs of both of these components.

Actuators

The POWER6 microprocessor provides several mechanisms to control, or actuate, the behavior of the chip in order to manage the power consumption of the chip and the system. These actuators provide control of the state and activity of both the POWER6 processor chip and, through the on-chip memory controllers, the DRAM (dynamic random access memory) subsystem.

Processor pipeline throttling

The POWER6 processor provides the ability to throttle the processor core pipeline in order to manage power. Pipeline throttling explicitly saves a moderate amount of active latch and array data switching power by limiting the rate at which instructions can be dispatched through the pipe to the execution engines. More importantly, it provides the existing clock hardware and power gating mechanisms with an opportunity to engage. The disadvantage to this method is that it is performance/power inefficient, since the corresponding power savings is approximately linear with performance loss, and with a shallow slope. However, this ability is invaluable since a rapid-response mechanism is required to best enforce a power or thermal limit. Pipeline throttling is also important as a per-core actuator, which is important for more localized management. The POWER6 microprocessor chip provides three primary processor throttling mechanisms, described below.

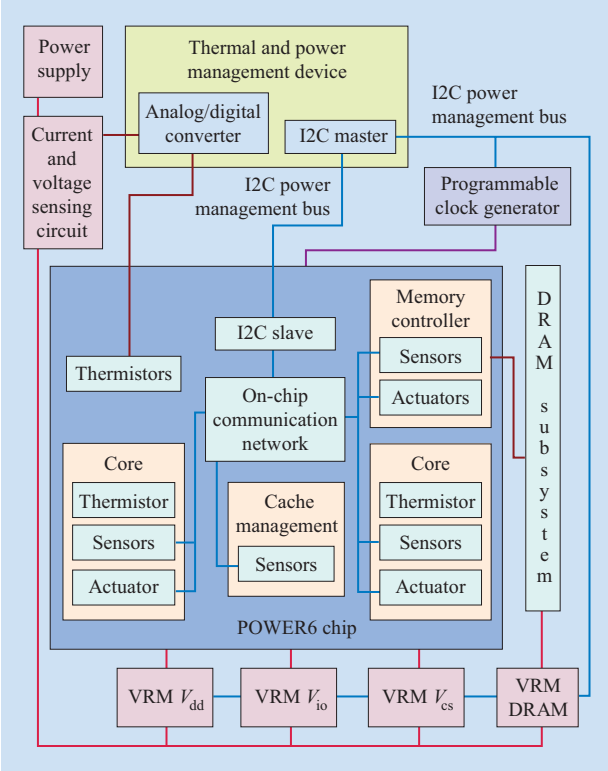

Figure 1

Conceptual topology of a POWER6 chip-based system showing the location of sensors, actuators, and power management logic. (The sensors and actuators are described throughout this paper.)

Emergency throttle

Heat is generated when the circuits on the chip dissipate power. Throttling the processor cores is the quickest way to address undesirable temperature levels from this heat. The cores are the primary consumers of power on the processor chip and, in many cases, in the server. A reduction of their activity also leads to a reduction in power consumption by the memory subsystem, another

significant source of power consumption in the system. When the hardware detects that a critical temperature has been reached, the emergency mechanism causes the hardware to automatically throttle the dispatch rate of the threads running on the processor cores in order to avoid a thermal runaway. Once the temperature has dropped below the warning level, the processor core gradually increases the dispatch rate back to the "normal" rate.

Run/hold throttle

The POWER6 microprocessor implements a run/hold processor throttling capability to provide the control firmware with an immediate actuation method to control power and temperature. This is accomplished by alternately stopping and releasing the dispatch of instructions. A 16-bit hold value (N) and a 16-bit run value (M) may be set dynamically during runtime in order to control the degree of throttling. For example, running for M cycles followed by holding for N cycles results in an active cycle fraction of $M/(M+N)$. While it may seem that only the run:hold ratio is important, larger values of M and N are better than smaller ones, because of many microarchitectural effects such as pipeline depth, re-fetch latency, various levels of cache latency, and multiprocessor activity (e.g., spin lock acquisition). Run/hold throttle is most useful as a method to manage power or enforce a thermal limit since it tends to provide a proportional reduction in power in response to dynamic changes in M and N.

Run/hold throttling requires that special consideration be taken when configuring the run and hold values of M and N, since the impact of this mechanism is workload dependent. In the POWER6 processor implementation, run/hold throttling is cycle based, not instruction based. It attempts to stop dispatch even if a group of instructions are not available for dispatch. Additionally, the threads may not always have something to dispatch on all run cycles. Therefore, the effectiveness and impact of run/hold throttling on performance can vary based on the workload that is running at the time, although in general it tends to average out, especially when using larger values. The disadvantage to using larger values is that it can create bursts of activity on the core, which may not be ideal in a symmetric multiprocessing (SMP) environment. On the POWER6 microprocessor, properly tuned values of M and N have been determined to produce the desired result.

In the POWER6 processor, the fetch rate can selectively be throttled instead of the dispatch rate, but we have empirically determined that controlling the dispatch rate provides more-predictable performance/power trade-offs across a wider set of workloads.

Instruction throttle

A new pipeline throttling capability was added to the POWER6 processor that provides a performance-centric pipeline throttle through an explicit limit on the instructions per cycle (IPC). The limit is specified as the maximum number of instructions that can be executed in a specific window of time given in processor cycles. Whereas run/hold throttling reduces activity for all workloads, IPC throttling can be configured to penalize only those workloads that exhibit more than the targeted level of activity.

The IPC throttling cap is for the entire core, not each thread. IPC throttling can be used to limit either the dispatch or the completion rate of instructions, and like run/hold throttling, it can be activated at the fetch or dispatch stage of the pipeline. The same 16-bit run/hold counters are reused for this mode. Over an execution window, defined as a programmable number of cycles, Y, only up to a maximum programmable number of instructions, Z, on either thread are permitted to complete. In this manner, the core is penalized only when it starts to exceed a certain performance threshold; otherwise, throttling never engages. The performance and resulting power reductions caused by IPC throttling are dependent on workload as well as the values of Y and Z. For given values of Y and Z, a workload with IPC greater than (Z/Y) will see its performance reduced while another with a lower IPC will see no impact on performance/power. As a selective throttling mode, IPC throttling can be used to avoid throttling when it is undesirable and also to enable more complex power management solutions that can trade off explicit levels of activity and power of a processor core for increased activity in another component or subsystem sharing the same power–cooling domain.

Multiple voltage domains

The voltage supplied to complementary metal-oxide semiconductor (CMOS) circuits has a large impact on their power consumption, because of the effects of switching and transistor leakage. Voltage to the circuits should be kept as low as possible in order to reduce the chip power consumption. However, operating voltage is a critical determinant of the performance of CMOS circuits. Some circuit designs, such as arrays and off-chip interface circuitry, typically require higher voltage levels than other circuits on the chip in order to attain the required performance. Some circuits can operate over a range of voltage levels, whereas others require a fixed voltage in order to properly operate. Additionally, because of their design and the nature of their function, some regions of the chip consume more power than other regions. Furthermore, power consumption in some regions of the chip usually varies with workload more

than in others. Increasingly, variations from manufacturing in newer silicon technologies often make the power consumed by these different types of regions on each chip different.

For these reasons, the POWER6 processor supports multiple voltage domains on the chip. The circuits are grouped into four classes: analog clock generation circuits, analog off-chip interface circuits, arrays (on-chip SRAMs), and logic, each with its own voltage. By default, all logic, including most latches, operates at the V_{dd} voltage. The arrays are provided with a separate higher voltage (V_{cs}) than the standard V_{dd} logic voltage in order to support higher-performance SRAMs and increase manufacturing yield while allowing the other logic devices to be set to the lower V_{dd} value. The analog circuits used in the off-chip interface devices require a higher and invariant voltage, which is named V_{io}, for between-chip signaling. Similarly, the analog clock generation portion of the phase-locked loop (PLL) device, which is responsible for generating the internal clocks for the chip, requires an even higher invariant voltage (named AV_{dd}) in order to generate a clean output (i.e., one with minimal noise and jitter). These independent domains allow circuit-type-specific tuning of voltage to minimize power consumption while achieving higher performance. Other voltage-sensitive circuits, such as the frequency dividers and phase detectors in the PLL, are designed to run off the invariant V_{io} voltage to allow more-rapid and larger changes in V_{dd}. This enables dynamic voltage and frequency scaling (DVFS), as described in the next section, to vary the frequency of the chip without being limited by such sensitive circuits.

Processor dynamic voltage and frequency scaling

The effects of pipeline throttling, dynamic frequency scaling (DFS), and DVFS are shown in **Figure 2**.

By far the most efficient power-savings lever for semiconductor circuits, short of powering them off, is a reduction in their supplied voltage (and the accompanying reduction in frequency). CMOS circuit switching power has a quadratic dependence on voltage (power $\propto$ voltage^2) and a linear dependence on frequency (power $\propto$ frequency). In addition, static transistor current leakage, which is also dependent on voltage, has become a significant component of circuit power consumption in newer silicon technologies. Additionally, higher voltages and frequencies tend to raise the operating temperature of the silicon, causing a further compounding effect on power consumption since leakage is also a function of temperature.

These factors can be described using the following equation:

$$P = (((C_s \times V_{dd}) + (I_{tr} \times t_{tr})) \times f \times SF + I_{dc}) \times V_{dd} \approx C_s \times V_{dd}^{(2.x)} \times f,$$

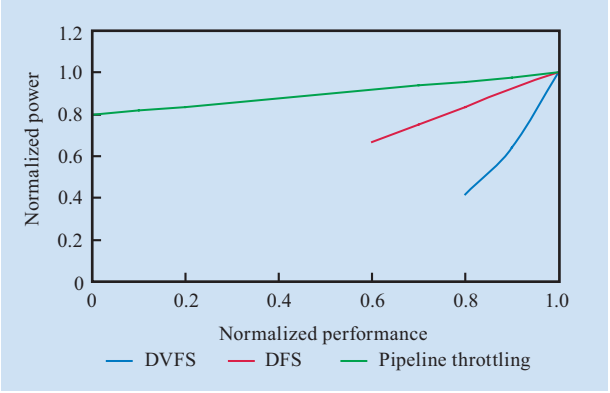

Figure 2

Comparison of different power management actuators on the POWER6 chip.

where x is in the range of 4 to 6; C_s is the switching capacitance; SF is the switching factor, or average number of times during a clock cycle that the gates logically change state before resolving to a stable value; I_{dc} is the transistor leakage current; I_{tr} and t_{tr} are the transistor switching current and transition time, respectively; V_{dd} is voltage; and f is frequency of operation.

Because f is related to V_{dd}, for simplicity, we assume a linear relationship,

$$P \sim C_s \times V_{dd}^{(3.x)}.$$

This higher-order (more than cubic) relationship has been readily observed in our hardware laboratories on chips manufactured using the more recently offered IBM 90-nm and 65-nm silicon transistor technologies.

Therefore, changes in frequency and, more importantly, voltage can have a significant impact on power consumption. The POWER6 processor supports dynamic changes in both voltage and frequency in order to provide the ability to tune it to a desired power/performance point during runtime. Of all the POWER6 processor actuators, DVFS typically provides the best power/performance trade-off (i.e., the highest power reduction for a given performance loss). The circuits in the POWER6 chip are designed specifically to operate across wide voltage and frequency ranges. As previously stated, some voltage domains are invariant (i.e., not slewed[1] during runtime), and some circuits run at a higher voltage level than others. Circuits requiring more-stringent specifications are placed on higher and/or invariant voltage domains so the range of scaling that can be realized will not be limited. Voltage-level-translation

[1] *Slew* is when a signal is changed at any point in a circuit at a maximum rate.

circuits, which are required when crossing between circuits of different voltage levels, are designed to handle these larger ranges of operation. The separate array domain voltage, V_{cs}, is slewed in concert with logic voltage, V_{dd}, in order to maximize the scaling range.

Additionally, the POWER6 processor has configurable parameters to allow itself as well as other chips in the system to functionally handle changes caused by frequency and voltage scaling. Sufficient "elasticity" has been designed into the high-speed Elastic Interface [2–4] and the "target-time" settings have been selected to handle the entire dynamic range of scaling. If the system interface timing at nominal frequency is at the limit of its operation at the optimal configuration, then scaling may require insertion of a small latency penalty; this causes a larger "target-time" elasticity to be selected, but this is usually not the case. To support scaling of the frequency, a configurable clock generation circuitry that requires special PLL range and tuned settings is designed in the arrays. A limitation of the DVFS implementation on the POWER6 processor is the rate of actuation of both voltage and frequency. The voltage is supplied by VRMs, which are components external to the processor that have slow control interfaces and long settling times. An external programmable oscillator provides the reference clock source to the PLL clock generation macros in the processor chip. Processor chips in previous and current IBM POWER* processor-based systems share a common clock frequency source. For multiprocessor performance, frequency changes must be coordinated across the system, which is a time-consuming process that involves the external reference clock oscillator.

Situations requiring a quicker response may require temporarily engaging the pipeline throttling actuator until the voltage and frequency can be modified. Additionally, since both cores on a chip share the same voltage and frequency source, individual core power reduction cannot take advantage of DVFS but must rely exclusively on pipeline throttling, which is a much less power/performance efficient lever than DFS or DVFS, as illustrated in Figure 2.

Processor idle modes

There are times when the system is idle (i.e., it runs without work for a particular processor to do). Historically, when the OS or hypervisor did not have a software thread to run on a processor, it entered the *idle loop*, a small sequence of instructions that repeatedly execute, while waiting for work to arrive. However, constantly executing instructions when there is no work unnecessarily wastes power. For this reason, the POWER6 processor supports the *nap* mode. Each hardware thread on a given processor core can issue an instruction that puts it into nap mode. When both hardware threads for that core are in nap mode, the whole processor core then enters the nap state. In the nap state, the processor eliminates almost all of the switching power in the core by stopping the internal clocks and restricting operation of its functional units. As a result, use of the nap mode has been observed on hardware to provide between 10% and 20% power savings over running the idle loop. The two POWER6 cores enter and exit nap mode independently of each other.

The processor nap state is interruptible, so the OS or hypervisor can re-awaken a napping core either by issuing the appropriate type of interrupt or by configuring the core to wake up on the basis of external (I/O) or timer (decrementer) interrupts. The latency of returning from nap is considerably greater than the latency of exiting the idle loop but is small enough that nap can be used in many cases to put idle processor cores into a lower power state. When a napping core receives the wake-up interrupt, the OS or hypervisor then runs the proper scheduling code to dispatch the waiting software thread.

Another use for nap is to reduce the power consumed by cores present in the system that are not licensed for use. Many IBM systems are sold with more processor chips and cores than are licensed, with the intent that the purchaser can reduce the initial cost of acquisition and later easily upgrade the system as conditions warrant. This capability is called *capacity upgrade on demand* (CUoD) and is a standard feature of the IBM System i* and System p* product lines. Prior to the introduction of the POWER6 processor, all unlicensed cores consumed full power and ran an idle loop, waiting to be licensed. In POWER6 processor-based systems, all unlicensed cores are kept in nap state to reduce the power consumption of the machine.

Memory controller dynamic modes

The memory subsystem consumes a significant fraction of the power in high-end servers [4]. These servers typically provide multiple ranks of DRAM chips to provide both high-capacity and high-bandwidth main stores. However, workloads cannot keep all of the DRAM chips constantly busy. In these server systems, a large portion of the DRAM power is consumed by idle chips. DRAM manufacturers provide a lower-power idle mode, power down, to decrease the DRAM idle power by deactivating the clock-enable control signal to the DRAM chips. This idle mode provides approximately a 90% reduction in device power in most DRAM designs.

The POWER6 processor memory controller, located on the processor chip, exploits the DRAM power-down mode, enabling significant savings in the memory subsystem power. The controller implements a queue-driven policy for power-down exploitation in which the built-in request queues are monitored on a rank basis; whenever devices of a rank are idle and there are no pending requests to that particular rank, the DRAM

chips for that rank are put into power-down. The DRAMs are removed from power-down as soon as a request to that rank is queued in the controller or when the rank must be refreshed. Traditionally, IBM server systems have not used the power-down mode because of potential performance loss caused by the overhead of entering and exiting the mode. With the queue-driven policy implemented in the POWER6 processor controller, applications of the power-down mode rarely see loss of performance due to this overhead, yet the system can obtain a significant reduction in DRAM power consumption using this mechanism.

Memory controller throttling

The on-chip POWER6 processor memory controller also supports activity regulation in the memory subsystem to address *active* mode power consumption in the DRAM devices. Since DRAM active power is proportional to the rate of memory requests being serviced, power savings result from a direct trade-off with memory subsystem performance (bandwidth). In the POWER6 processor memory system, the memory controller communicates with the DRAM devices through buffer chips. The communication between the memory controller and the DRAM buffer chip is in the form of *command* and *data packets* sent over a serial path in multiple clock cycle windows called *frames*. Three different modes of throttling are supported in the POWER6 processor controller: by explicitly controlling the ratio of active to idle frames, by controlling the number of read or write requests in a 128-frame interval, and by limiting the number of concurrent requests (from 1 to 16) a memory controller can support. The throttling modes provide fast and flexible control of the DRAM subsystem power for system thermal management. In addition, the modes enable more-sophisticated power/performance optimization strategies in which DRAM subsystem power/performance may be traded off with processor power/performance [5].

Sensors

A key focus of the POWER6 processor design was to make all pertinent measures for power, performance, and thermal management decisions readily available. The POWER6 processor provides real-time access to on-chip temperature, processor core activity, and off-chip cache and memory activity. On-chip operational stability information is also provided to support more-aggressive mechanisms for dynamically tuning voltage and frequency settings.

Temperature sensors

The POWER6 processor provides two types of on-chip temperature monitoring sensors. An on-chip thermal sensor (OCTS) is implemented as a thermistor (wire resistor), instead of the frequently used thermal diode. The OCTS resistance is a linear function of its temperature and is determined by measuring the current that results from applying an external voltage across its terminals. The OCTS is calibrated during manufacturing tests by measuring its resistance at known temperatures. These calibration points are used to build a linear relationship between resistance and temperature over the operating range of the POWER6 processor. This information is recorded as part of the processor vital product data (VPD) and stored in an external EEPROM (electrically erasable programmable read-only memory) associated with each processor chip. One OCTS is located in each processor core, and one is located near the center of the processor chip. The disadvantages of the OCTS are that it requires off-chip module pins, external voltage, and an analog/digital (A/D) converter.

The second type of thermal sensor provided is an on-chip digital thermal sensor (DTS) that can be read without an off-chip A/D converter. It is based on a temperature-sensitive ring oscillator (TSRO) design. Since the speed of the ring oscillator changes in response to temperature, a count of the number of oscillations over a given period can be translated into a temperature. Unlike the more traditional performance screen ring oscillator (PSRO), the TSRO is designed to be voltage invariant since on-chip voltages fluctuate naturally during runtime and may change due to DVFS. DTS calibration is performed during manufacturing tests using a method similar to the calibration of the OCTSs, in which a relationship is constructed between the counts and the known temperatures.

Figure 3 shows the placement of the DTSs and thermistors (i.e., OCTS) on the POWER6 processor floorplan. The POWER6 microprocessor chip contains dual processor cores running at speeds in excess of 4.5 GHz. The area between and around the processor cores runs at half the core clock rate and contains an 8-MB level 2 (L2) cache, level 3 (L3) directory to support a 32-MB off-chip L3 cache, dual DRAM memory controllers, an IBM GX-protocol I/O controller, and interconnection fabric circuitry to support up to 128-way SMP system. The chip is manufactured using the IBM 65-nm silicon technology, containing 790 million transistors within a total area of 341 mm^2. On the POWER6 processor, eight DTSs are located in each core and eight are outside the cores, for a total of 24 sensors per processor chip. The DTSs are intentionally placed in order to monitor likely areas of high activity and high power density, i.e., *hot spots*, in addition to forming a rough grid across the chip to facilitate creation of a power map of the chip.

The DTSs are part of a thermal monitoring system that collects the TSRO counts and makes them available via

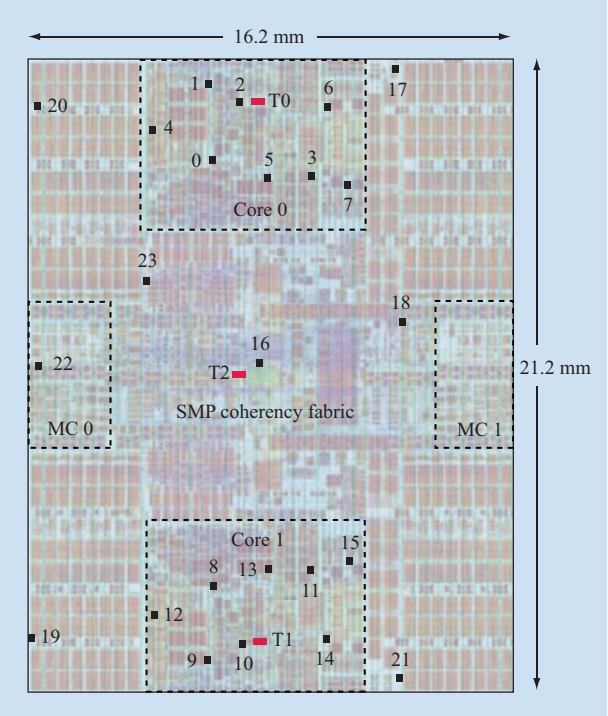

Figure 3

Die photograph of the POWER6 microprocessor chip showing the location of the digital thermal sensors (DTSs 0–23) and thermistors (on-chip thermal sensors T0–T2). (MC: memory controller.)

special registers. This system is also included in the emergency thermal management system discussed earlier.

Critical path monitors

The POWER6 processor introduces an additional sensor type to monitor the operational status of the circuits in the chip. It does this by modeling the critical timing paths in various regions of the processor chip with special monitors. Critical path monitors (CPMs) are placed next to the DTSs throughout the chip in order to be as close as possible to potential thermal hot spots and areas of high current draw. These regions are traditionally the first circuits to fail, due to voltage supply droop (caused by di/dt) and the slowing down of transistors subjected to localized heating. These regions also tend to contain the limiting critical speed paths of the chip because high power density is associated with regions of wire congestion and closely packed circuits.

Different delay paths can be synthesized with a CPM that can be tuned to match the response of circuitry in the monitored region under different voltage, frequency, and temperature conditions. A clock edge is launched through these delay paths, and its progress is recorded by a sequence of latches. The number of latches the signal is able to traverse before the end of the clock cycle gives us an indication of the performance of critical paths in that region. Each delay path has its own characteristic response curve. Each CPM is configurable, allowing the most representative delay path to be selected. Each CPM also has a programmable amount of insertion delay before the synthesizing circuits are encountered by the clock edge, allowing the CPMs to be used over a wide range of voltages and frequencies.

A CPM monitors circuit timing and, indirectly, the impacts of voltage and temperature variation in its region of the chip. If the voltage is inadequate for the current operating parameters, this insufficient voltage is reflected by a reduction in the number of delay elements through which the launched edge passes. Likewise, the less-dramatic impacts of temperature and frequency are detected. The goal of the CPM is to anticipate a problem before it occurs in its region and to indicate when there is margin available to increase performance. Additional details about the CPM are provided in Drake et al. [6].

Since the CPMs provide real-time feedback on the voltage–frequency relationship for the current workload and operating environment, more-intelligent adaptive power management solutions may be enabled. Since the circuits are being actively monitored during runtime, traditional guard bands may be lessened since we no longer have to statically apply margin to handle the worst-case scenario (workload, silicon variation, and temperature). The DVFS algorithms may be able to use CPM measurements, in conjunction with DTS readings, to steer the dynamic choice of frequency and voltages. These measurements may also be used to inform power/performance trade-off algorithms on how best to remain in a safe operating region based on current chip conditions, which constantly vary due to workload and environment changes.

Processor core and memory activity counters

The POWER6 microprocessor uses dedicated counters to track various activities that are important to power/performance optimization. These counters re-use signals already available to the performance monitoring unit (PMU) but are implemented independently from the traditional user-programmable performance counters and, thus, are always available for power management techniques to use.

For each processor core, a *finished* (*completed*) instruction count register, a *dispatched* instruction count register, and a processor *cycle* count register are available. The three counts are cycle synchronized and are obtained in a single access. They can be used to obtain the finished instruction throughput (i.e., IPC) and, when used to obtain the dispatch-to-finish rate, the extent of speculative activity in the core. Since both threads share

the same core resource, the counts are combined per thread into a single per-core register.

For the memory hierarchy, there are two boot-time programmable counters per core that can be set to obtain a weighted sum of the counts of the different memory hierarchy-level accesses possible in POWER6 processor-based systems. The POWER6 processor tags reload data that is returned to the core after an L1 cache miss with a field indicating from which level of the memory hierarchy the data originated. Depending on the weights programmed, one can use each counter to infer one of various measures of the workload memory hierarchy usage. These measures include the number of accesses to specific cache levels, estimates of cycles spent at specific levels in the memory hierarchy over a given duration, and the extent of compute-memory dependence of the workload. The latter allows the frequency dependence of the performance for the workload to be determined. These two counters are cycle synchronized so that a single reading provides the values of the counters plus a cycle count that can serve either as a time stamp for the memory access counts or as an independent counter in order to deduce the operating frequency of the chip.

Additional counters are available in each of the two on-chip memory controllers to provide more information about DRAM activity. These provide time-stamped counts of the number of reads, writes, DRAM activations, and power-ups (DRAM requests when in power-down mode). They provide a detailed view of the memory subsystem activity, which can be correlated to memory subsystem power and can be used to assess the potential impacts of memory power-down and/or throttling usage.

In summary, these counters enable a number of power/performance optimizations via real-time workload analysis, estimating the impact of power management actions on workloads [7, 8] and even estimating system subcomponent power in the absence of direct measurements [9, 10].

Power sensors

Because of the difficulty in designing and realizing accurate power measurements using on-chip mechanisms, no attempt was made to design on-chip power sensors. High-precision external circuits are used in POWER6 processor-based servers in order to measure the power consumption of select server subsystems and components, including the POWER6 chip, as well as total system power consumption in order to enable enforcement of an absolute power cap. In addition, individual VRMs provide reasonably accurate voltage and current measurements for both the POWER6 chip and memory (DRAMs).

Off-chip communication and control

POWER6 processor-based systems are designed with support for multiple processors. During normal operation, each of these processors may run different workloads, each with different power consumption. Thus, power management actions performed for one processor may directly oppose those needed by a different processor. Such a scenario requires the design of a robust centralized management entity.

System architects considered the option of performing these management actions as part of the OS or hypervisor, as an in-band approach. In this option, software running in the POWER6 processors would monitor the workload and environmental conditions using the sensors described earlier and dynamically control the actuators to manage system power and temperature. This solution would seemingly result in a highly flexible system without the need for much additional hardware. However, these highly critical actions have hard real-time constraints that are difficult to enforce when running as in-band services. Another problem is that highly active systems would require closer monitoring of the power and thermals, resulting in the power management code competing with the rest of the applications running in the processors, potentially hurting overall performance. In addition to these drawbacks, an in-band solution also carries a large firmware-development cost to restructure the OSs and hypervisor to operate in hard real-time on a single dedicated thread.

Thus, we chose to use a dedicated external microcontroller—the TPMD. The POWER6 processor was designed to allow this external device to manage the power and temperature of the system, using an out-of-band approach, as illustrated in Figure 1.

On-chip sensor and actuator network

The POWER6 processor supports a mechanism in which selected internal registers are available during runtime via an on-chip network [11]. This network allows out-of-band software to access these registers dynamically during runtime. Registers associated with all of the on-chip sensors and actuators described previously are connected to this network. This eliminates the need for out-of-band software from having to transfer to an I/O device or memory to read or configure these registers indirectly. The same network can also facilitate in-band power management solutions that enable registers in one processor to be accessible for code executing on another processor. Additional care was taken to optimize the most frequently performed operations, including providing *packed access* to sensors with multiple sensors being accessed through a single read operation on the network. This ensures that sufficient hard real-time access

bandwidth is available for an out-of-band power management solution.

Dedicated processor power management bus

The TPMD uses a multidrop serial link to communicate with the POWER6 processor. This bus runs over the I2C bus electrical and addressing protocol [12], which is connected to the on-chip network, as described in a previous section. The payload in the I2C packet holds the address (24 bits) and data (64 bits) of the internal register. In addition, each POWER6 chip has a unique I2C address, which allows multiple processors to share a power management bus. To ensure necessary bandwidth on the bus and maintain fast access times (i.e., of the order of a few milliseconds), configurations with a large number of processors separate these processors onto different buses.

Thermal and power management device

To handle the requirements of POWER6 processor-based systems and provide safe operating conditions, these systems use a combined hardware–firmware approach to power management. The sensors and actuators described earlier constitute the hardware. They are reliable and flexible, but except for protecting the system from catastrophic failure, they are not entirely autonomous. The task of deciding which actuators to use is done by the code running on the TPMD.

The functions of the TPMD are to gather information from the sensors, decide which actuators to use and to what extent, and to send control information to the selected actuators. In one system implementation (see Figure 1), the TPMD is a readily available microcontroller that includes its own fast I2C bus interfaces to communicate with the POWER6 processor, VRMs, programmable clock chip, and other devices; A/D converters to access external analog sensors; a prioritized interrupt unit; as well as its own internal Flash ROM (read-only memory) and internal RAM (random access memory).

The TPMD runs the power management firmware component [13], which is based on a thread-based, real-time, embedded operating system. Actions (e.g., read DTS sensor, determine safe frequency setting) are implemented as real-time threads with explicit priorities and deadlines. A view of all of the sensors in the system is maintained in the internal RAM, which allows power management algorithms to consider system-wide optimizations.

The TPMD firmware is also integrated with the rest of the firmware and software that runs in the system. This integration allows upper levels of the system management stack, like the IBM PowerExecutive* management software [14], to access information about the system or to control parameters of the power management algorithms that run in the TPMD.

Power/performance optimization modes

The capabilities of the EnergyScale architecture, which was introduced in the POWER6 processor and the systems designed around it, enable a variety of power management solutions. In this section, we briefly describe two specific solutions that can be enabled: power/thermal capping and performance-sensitive power savings.

Power/thermal capping

Power/thermal capping refers to the ability to limit the power consumption and dissipation at a component, a subsystem, or the full system level. This capping provides the power and/or thermal constraints as specified by the customer.

Thermal capping solutions can be implemented using the thermal sensors for feedback. As described earlier, OCTS (coarse-grain, analog) and DTS (fine-grain, digital) sensor measurements are available. As discussed earlier, two methods of changing on-chip temperatures are available: per-core pipeline throttling and the system-wide frequency scaling (with corresponding scaling of chip domain voltages). These methods can be used independently of or in conjunction with each other. In addition to the on-chip sensors, system designers can provide other thermal sensors on planar components or near blowers and fans to feed system-specific thermal capping solutions.

Power capping solutions can be similarly implemented using off-chip, per-domain current and voltage information from suitable VRMs. These relatively slow sensors can be supplemented by per-subsystem measurement circuitry providing accurate and fast access to voltage and current information for any instrumented domain. Domains monitored for power consumption would typically include total system power consumption. If external power measurements of domains are not provided by system designers, the instruction dispatch and finish rate of the two processor cores can be used to estimate a rough (and relative) power consumption value. Since change in processor activity is the likely precursor to changes in power and temperature, these counters can also be analyzed to proactively predict and, therefore, prevent a serious power or thermal situation that might require much more extreme measures to correct after the fact.

We note that when addressing full system power consumption or off-chip subsystem power and thermal capping solutions, the memory subsystem throttling controls on the POWER6 chip can also be engaged to enable appropriate capping.

Performance-sensitive power savings

Performance-sensitive power savings refers to power management solutions that optimize for power savings while recognizing specific performance requirements. Power savings has been an important goal, first in embedded systems, later in laptop and desktop systems, and now in high-end server systems. Server systems typically have greater performance requirements and constraints, and they have been able to exploit system idleness for power savings only through nap mode usage and lower-level active voltage–frequency states with techniques such as demand-based switching. In both cases, power savings is directly dependent on system inactivity, with no savings obtainable when the system is fully active.

With the incorporation of activity monitors in the form of counters dedicated to power management (see the section on processor core and memory activity counters, earlier in this paper), the POWER6 processor enables explicit power/performance trade-offs. Performance-feedback-based power management solutions can be implemented using relative instruction throughput information as a measure of relative performance impact of power management actions [7]. Additionally, one can assess workload characteristics, such as how dependent the performance of the workload is on frequency [7, 8], based on the extent of memory hierarchy activity. The counters provide explicit means to analyze executing workload characteristics and provide feedback on any power management-induced activity that changes in real time. Coupling this feedback with actuation can (1) enable explicit, user-specified performance trade-offs for higher power savings when desired and (2) avoid conservative decisions on power management when workload characteristics or user specifications permit more aggressive actions.

Additionally, the distributed activity monitors in the two cores and memory subsystem provide subsystem-specific activity information. This enables sophisticated power management solutions in which the power and performance of different subsystems can be traded off against each other [5].

System software enablement

In this section, we discuss other changes in the processor design that allow system software to take into account the impact of power management on system behavior, which is necessary to allow the features of the EnergyScale architecture to be enabled.

Accounting and utilization

To track time, POWER architecture-based processors have a special-purpose register, the timebase, which increments at a fixed, uniform rate for all of the processor cores in the SMP system. The timebase is not affected by processor frequency scaling and always increments at the same rate. Early implementations of POWER architecture-based systems used the elapsed timebase ticks to determine the utilization of processor resources by a particular program for accounting purposes.

The IBM POWER5* processor introduced simultaneous multithreading (SMT), which is a feature in which two threads share the resources of a processor core. With SMT, a per-thread processor utilization of resources register (PURR) was introduced to accurately track the processor usage in timebase ticks by each of the two hardware threads. Processor usage by a program mapped to a particular hardware thread can be inferred from the corresponding PURR counts.

With the introduction of dynamic power management actions such as pipeline throttling and DVFS, additional facilities are introduced to allow software to appropriately account for throttled or scaled-down availability of the processor resources. The POWER6 processor implements scaled PURRs (SPURR) that provide a scaled count per hardware thread that factors in the impact of throttling and DVFS, and whose count is mathematically equivalent to

$$SPURR = PURR * (f_{\text{effective}}/f_{\text{nominal}}) * (1 - \text{cycles}_{\text{throttled}}/\text{cycles}_{\text{total}}),$$

where $f_{\text{effective}}$ is the effective frequency during the count-accumulation interval, f_{nominal} is the nominal frequency of the processor that is adhered to in the absence of power management actions, $\text{cycles}_{\text{throttled}}$ is the number of processor cycles that the pipeline was halted during the count-accumulation interval, and $\text{cycles}_{\text{total}}$ is the total number of processor cycles during that interval.

On POWER6 processor-based systems, accounting calculations can factor in the extent of below-nominal-frequency usage of a processor by a program because of power management actions by using the SPURR, if desired. Conversely, in system designs in which the effective frequency can be increased above the nominal under favorable conditions, the SPURR can be used to track the extent of above-nominal-frequency usage by programs that demand and use a higher frequency.

Performance analysis

Building on support from earlier generations of the processor family, the POWER6 processor provides performance counters that allow programmers to make modifications to the system in order to more fully understand specific program behaviors. The POWER6 processor provides several new selectable events to allow these programmers to determine the effect of throttling and frequency scaling on the system and processor core performance:

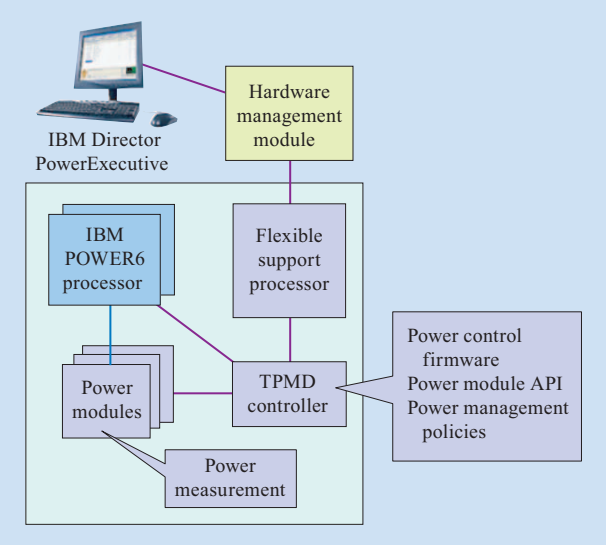

Figure 4

High level view of POWER6 system power management infrastructure. (API, application-programming interface; TPMD, thermal and power management device.)

- Frequency is being slewed down below nominal by TPMD.
- Frequency has been slewed up above nominal by TPMD.
- Processor is being throttled by the TPMD.
- Emergency processor throttling (e.g., due to a thermal event) is active.

Note that these can be used in conjunction with the SPURR to get a complete picture of the effect of power management policy on the behavior of the system under differing workloads and operating environments.

Power policy management

The POWER6 chip supports numerous power management mechanisms, but most of the control and policy logic controlling the usage of these mechanisms is not resident or executing on the processor chip. This is to increase the timeliness and accuracy of the control, to provide systems-level power management on multiprocessor systems, to enable system-specific power management solutions, and in general, to provide greater flexibility in the range of power management solutions that can be deployed. These controls are directed by the customer to enforce system or data-center-wide policies. This infrastructure is depicted in **Figure 4**.

The system power management firmware running on the POWER6 processor hypervisor, TPMD component, and system service processor interacts with higher-level system monitoring/management layers such as the IBM PowerExecutive [14]. This operates by relaying lower-level system power/thermal information upward to the customer and sending data-center and system policy directives downward into the hardware components. Policy directives can dictate specific operating modes for the power management system such as the following:

- *Trending and data collection*—In this mode, the data that is collected from sensors on the POWER6 processor (plus the other, off-chip sensors) is forwarded to a management server that records it and creates trending and other displays for user evaluation.
- *Oversubscription protection*—In systems with dual or redundant supplies, additional performance can be obtained using the combined supply capabilities of all supplies. However, if one of the supplies fails, the power management would immediately switch to normal or reduced levels of operation to avoid oversubscribing the functioning power subsystem. This can also allow less-expensive servers to be built for a higher (common-case) performance requirement while maintaining the reliability, availability, and serviceability (RAS) redundancy feature expected of IBM servers.
- *Power/thermal capping*—In this mode, explicit system- and subsystem-level power and thermal caps can be set and enforced for increased flexibility of data-center operations, emergency management, increased reliability, or increased system lifetime.
- *Power savings*—Different options for saving power can be dictated including specific trade-offs in performance (see the section on performance-sensitive power savings) and choice of power-savings strategy.
- *Maximum performance*—Nominal operating points for a system are typically dictated by worst-case workload and operating conditions for given system supply and cooling constraints. A power management solution utilizing all of the POWER6 processor sensors can potentially support a superior performance mode using above-nominal operating points, depending on additional margins allowed by current workload and environment conditions (assuming that the hardware has been tested and has been qualified at the selected operating condition).

One can easily envision more-sophisticated policies supporting combinations of the above operating modes. In all cases, external logic outside the POWER6 processor can select the mechanisms to engage and dictate the range of different settings used by the power management logic

by using the POWER6 processor facilities and adhering to its specified ranges of operation as determined during manufacturing testing. Ultimately, the customer must select the desired power/performance trade-off policy via the management console. This policy is then communicated to the system firmware, which can then engage the appropriate mechanisms to implement that request. In the future, we anticipate that higher levels of automation will be implemented into the power management control system that can more dynamically adapt to changes in system operation for the customer.

Concluding remarks

In this paper, we have described the features implemented by the POWER6 processor to support power management and power/performance optimization. The POWER6 chip is the first in the POWER architecture family that provides advanced power management features, known as EnergyScale technology. These features provide the basic mechanisms that support the dynamic runtime control and reduction of power consumption as well as measurement sensors to ensure that the processor operates within a safe operating range and at the optimal performance level. In order to offer the most efficient solution, e.g., providing the best performance for the lowest power, the POWER6 chip has special facilities that an external microcontroller can use to track and intelligently control the behavior of the processor in real time. Since power management can cause the processor to operate at a speed that is different from its nominal value, the POWER6 chip includes unique features that allow accounting, monitoring, and performance management software to track the changes in processor operations. By having the ability to dynamically adapt to changing conditions in real time, POWER6 processor-based systems operate more efficiently and can more easily meet the diverse requirements imposed by an increasingly power-constrained world.

References

1. K. Bernstein, D. J. Frank, A. E. Gattiker, W. Haensch, B. L. Ji, S. R. Nassif, E. J. Nowak, D. J. Pearson, and N. J. Rohrer, "High-Performance CMOS Variability in the 65-nm Regime and Beyond," *IBM J. Res. & Dev.* **50**, No. 4/5, 433–449 (2006).
2. D. M. Dreps, F. D. Ferraiolo, and K. C. Gower, "Elastic Interface Apparatus and Method Therefor," U.S. Patent No. 6,334,163, December 25, 2001.
3. F. Ferraiolo, E. Cordero, D. Dreps, M. Floyd, K. Gower, and B. McCredie, "POWER4 Synchronous Wave-Pipelined Interface," *Hot Chips 11: A Symposium on High-Performance Chips*, August 15–17, 1999; see *http://www.hotchips.org/archives/hc11/2_Mon/hc99.s2.2.Ferriaolo.pdf*.
4. C. Lefurgy, K. Rajamani, F. Rawson, W. Felter, M. Kistler, and T. W. Keller, "Energy Management for Commercial Servers," *IEEE Computer*, December 2003, pp. 30–48.
5. W. Felter, K. Rajamani, T. Keller, and C. Rusu, "A Performance-Conserving Approach for Reducing Peak Power Consumption in Server Systems," *Proceedings of the 19th ACM International Conference on Supercomputing*, Cambridge, MA, June 20–22, 2005, pp. 293–302.
6. A. Drake, R. Senger, H. Deogun, G. Carpenter, S. Ghiasi, T. Nguyen, N. James, M. Floyd, and V. Pokala, "A Distributed Critical-Path Timing Monitor for a 65nm High-Performance Microprocessor," *Proceedings of the International Solid-State Circuits Conference*, February 11–15, 2007, pp. 398–399.
7. K. Rajamani, H. Hanson, J. Rubio, S. Ghiasi, and F. Rawson, "Application-aware Power Management," *Proceedings of the 2006 IEEE International Symposium on Workload Characterization*, San Jose, CA, October 2006, pp. 39–48.
8. R. Kotla, S. Ghiasi, T. Keller, and F. Rawson, "Scheduling Processor Voltage and Frequency in Server and Cluster Systems," *Proceedings of the 19th International Parallel and Distributed Processing Symposium*, April 4–8, 2005.
9. K. Rajamani, H. Hanson, J. C. Rubio, S. Ghiasi, and F. L. Rawson, "Online Power and Performance Estimation for Dynamic Power Management," *Research Report RC-24007*, IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598, July 2006; see *http://domino.research.ibm.com/library/cyberdig.nsf/398c93678b87a12d8525656200797aca/8a07e78644cb39d9852571b200687e3b?OpenDocument*.
10. C. Isci, G. Contreras, and M. Martonosi, "Hardware Performance Counters for Detailed Runtime Power and Thermal Estimations: Experiences and Proposals," *Hardware Performance Monitor Design and Functionality Workshop*, February 2005.
11. M. S. Floyd, L. S. Leitner, and K. F. Reick, "Method and Apparatus for a High-Speed Serial Communications Bus Protocol with Positive Acknowledgement," U.S. Patent No. 6,529,979, March 4, 2003.
12. Phillips Semiconductors, "The I^2C-Bus Specification," Version 2.1, white paper; see *http://www.nxp.com/acrobat_download/literature/9398/39340011.pdf*.
13. H.-Y. McCreary, M. A. Broyles, M. S. Floyd, A. J. Geissler, S. P. Hartman, F. L. Rawson, T. J. Rosedahl, J. C. Rubio, and M. S. Ware, "EnergyScale for IBM POWER6 Microprocessor-Based Systems," *IBM J. Res. & Dev.* **51**, No. 6, 775–786 (2007, this issue).
14. P. K. Popa, "Managing Server Energy Consumption Using IBM PowerExecutive," white paper (May 2006); see *ftp://ftp.software.ibm.com/common/ssi/rep_wh/n/XSW02410USEN/XSW02410USEN.PDF*.

Received March 28, 2007; accepted for publication August 20, 2007; Internet publication October 23, 2007

Michael S. Floyd *IBM Systems and Technology Group, 11400 Burnet Road, Austin, Texas 78758 (mfloyd@us.ibm.com).*
Mr. Floyd is a Senior Engineer specializing in RAS and power-efficient design of server microprocessors and systems. He received a bachelor's degree in computer engineering from the Georgia Institute of Technology in 1995 and a master's degree in electrical engineering from Stanford University in 2000. His 12 years of experience with IBM include bring up, testing, and debugging of the Power PC 620* and POWER4* microprocessors, in addition to holding RAS design, lead, and microarchitecture definition roles for the POWER4, POWER5, and POWER6 microprocessors and support chips. After leading the POWER6 microprocessor RAS design, he worked with the IBM Research Division to develop the POWER6 system power management implementation. Mr. Floyd, who is currently the POWER7* Adaptive Power Management Design Leader/Architect, has been issued 27 patents and has 47 more pending. He has been named an IBM Master Inventor and has co-authored four published papers.

Soraya Ghiasi *IBM Research Division, Austin Research Laboratory, 11501 Burnet Road, Austin, Texas 78758 (sghiasi@us.ibm.com).* Dr. Ghiasi is a Research Staff Member in the power-aware systems department at the IBM Austin Research Laboratory. She is interested in holistic design, which leverages all levels of the development stack to enable thermal, power, and performance management of processors, computer systems, and data centers. She has more than a dozen refereed publications ranging from data ingest systems to telerobotics to power and performance modeling. Dr. Ghiasi received her B.A. degree in physics in 1996 and her Ph.D. in computer science from the University of Colorado in 2004.

Tom W. Keller *IBM Research Division, Austin Research Laboratory, 11501 Burnet Road, Austin, Texas 78758 (tkeller@us.ibm.com).* Dr. Keller is an IBM Distinguished Engineer working in the area of data-center and systems power. When in the power-aware systems department in IBM Research, he began the first system-level power management efforts for IBM products. Before joining IBM in 1989 as the Technical Lead in the AIX* performance group, he led the Department of Energy's performance evaluation of the first Cray computer at the Los Alamos Scientific Laboratory and prototyped a parallel database machine at MCC, where his team created the longstanding TPC-C benchmark. He has served as Associate Director of the University of Texas Computation Center and Chair of ACM Sigmetrics. He is the author of more than 40 refereed publications and numerous patents and is a member of the IEEE and ACM. Dr. Keller received a Ph.D. degree in computer sciences and a B.S. degree in physics with honors from the University of Texas at Austin.

Karthick Rajamani *IBM Research Division, Austin Research Laboratory, 11501 Burnet Road, Austin, Texas 78758 (karthick@us.ibm.com).* Dr. Rajamani joined IBM Research as a Research Staff Member in the power-aware systems department in 2001. His research interests are in the design and development of computer systems and technologies with the focus on power, performance, and management. He has more than ten refereed publications and several patent applications on power management. He received his B.Tech. degree in electronics and communications engineering from the Indian Institute of Technology, Madras, and his Ph.D. degree in electrical and computer engineering from Rice University.

Freeman L. Rawson *IBM Research Division, Austin Research Laboratory, 11501 Burnet Road, Austin, Texas 78758 (frawson@us.ibm.com).* Mr. Rawson is a Senior Technical Staff Member in the power-aware systems department in the Austin Research Laboratory. Prior to coming to the Research Division, he spent 23 years in IBM development working on a wide variety of systems-related projects. His research interests include applied artificial intelligence, machine learning, systems management, operating systems, and systems architecture. He received a B.S. degree in mathematics from Michigan State University and a Ph.D. degree in philosophy from Stanford University. He holds 20 patents, has 20 more patent applications pending, and has published more than 20 refereed technical papers. He is a member of the IEEE Computer Society, the ACM, and AAAI.

Juan C. Rubio *IBM Research Division, Austin Research Laboratory, 11501 Burnet Road, Austin, Texas 78758 (rubioj@us.ibm.com).* Dr. Rubio is a Research Staff Member at the IBM Austin Research Laboratory. He received a B.S. degree in electrical engineering from Universidad Santa Maria La Antigua, Panama, in 1997 and a Ph.D. degree in computer engineering from the University of Texas at Austin in 2004. His dissertation explored a hierarchical architecture to improve data access in large enterprise workloads. At IBM, he has studied architectural techniques to monitor, model, and manage power and temperature in computer servers.

Malcolm S. Ware *IBM Research Division, Austin Research Laboratory, 11501 Burnet Road, Austin, Texas 78758 (mware@us.ibm.com).* Mr. Ware received a B.S. degree in electrical engineering from Purdue University in 1983 and an M.S. degree in computer architecture and communications from North Carolina State University in 1986. He spent his first 10 years with IBM at the Research Triangle Park (RTP) facility developing speech and image coding algorithms, music synthesizers, and low-speed modems for the Mwave DSP. In 1993 he went on international assignment for 5 years to the IBM Zurich Research Laboratory (ZRL) in Switzerland, and he worked with IBM Fellow Gottfried Ungerboeck on high-speed modems including V.34 and V.90 for the Mwave products shipped in Lenovo ThinkPad** laptop computers. After returning to RTP for 2 years to examine broadband and network processing opportunities, he spent 16 months on assignment again at ZRL developing prototypes of ADSL (Asymmetric Digital Subscriber Line), Symmetric High-speed Digital Subscriber Line (SHDSL), and Very high-speed Digital Subscriber Line (VDSL) broadband transceivers. From 2002 to 2003, Mr. Ware studied wireless transmission systems at RTP and then power modeling for embedded systems. In 2003 he joined IBM Austin Research Laboratory to pursue power-aware systems including the IBM server-class systems for both System x* and System p, with a primary focus on closed-loop control systems to manage power, thermals, and performance dynamically. He was promoted to Senior Technical Staff Member in 2006. He currently holds 39 U.S. patents.