Robust power management in the IBM z13

The power management strategy adopted for the IBM z13™ processor chip (referred to as the CP or Central Processor chip) is guided by three basic principles: (a) controlling the peak power consumption by setting a realistic limit on the so-called thermal design power or thermal design point (TDP) driven by customer workloads and maximum-power stress microbenchmarks; (b) reduction of the voltage margin by using a novel dynamic guard-banding technique; and (c) the creation of a rich new set of fine-grained, time-synchronized sensors that track performance, power, temperature, and power management behavior for a running machine. A prime requirement of the power management architecture is that the efficient control mechanisms be designed in such a manner that the high standards of IBM z Systems™ application performance and reliability be maintained without any compromise. In this paper, we describe the key features constituting the z13 CP robust power management architecture and design that meet the stipulated objectives.

Introduction

Power delivery and dissipation limits constitute a major constraint in achieving the market-driven performance targets of next-generation server and mainframe systems [1]. While technology scaling allows increased device and core count at near-historical growth rates, operational frequency and single-thread performance growth has slowed considerably because of the power “wall.” The IBM z13™ system is no exception in this regard, and as such, a careful design of the power management support is essential to the product’s success in the marketplace.

Several basic goals exist with respect to the z13 power management (PM) architecture. For example, one major goal is the reduction of power over-allocation by power-capping [2]. Very high-power workloads that are outside the realm of known or expected customer applications need to be detected, and an appropriate throttling action has to be triggered to make sure that the chip stays within a stipulated power budget. One way to set a high-power threshold is to define a limit on sustained instructions completed per cycle (IPC) measured over a cycle window that is chosen carefully in consideration of the thermal time constant of the system. This high IPC (and hence activity) “sensor” is easy to construct via digital activity counters that do not require post-silicon calibration. Hence, in the z13 design, such a IPC-driven activity sensor can be used to serve the needs of the power management function. In the most conservative setting typical of reliable IBM mainframes, the actual power budget limit, which is often referred to as the thermal design power (TDP), must be determined via systematic generation of synthetic maximum-power loops that are above the power levels of all (real) customer workloads, but significantly below the theoretical worst-case power obtained by adding the highest-utilization power values of individual units or macros. The choice of the TDP limit must also ensure that it complies with the stipulated power delivery limit. In other words, the TDP must be less than the voltage regulator design point (RDP), which defines the ultimate limit. This approach ensures a realistic peak power limit while preserving the robust performance and functionality requirements.

Another key objective is to reduce the voltage margin guard band required to handle inductive noise events. Such events are caused by large temporal gradients in load current (di/dt). Depending on the magnitude of the net inductance (L) driven by the power supply and on-chip wires, the noise (L · di/dt) on the supply voltage rail is an effect that must suitably be provisioned for in setting the...
operational voltage point. Instead of a very conservative static guard band added to the supply voltage to protect against voltage droop (noise), the objective here is to use a lower voltage (to save power), but with reliance on specially crafted sensors to detect unacceptably large droops. The response to such voltage droops is via dynamic “activity throttle” actions in order to contain the droop level to safe limits. Again, this approach ensures energy-efficient resilience and robust performance. The adopted strategy is unique to IBM z Systems®, but draws upon the experiences gleaned from a similar exercise pursued in the context of the IBM Power Systems [3–5].

A third major goal of this design is to create high visibility into performance, power, temperature and power management behavior in a fine-grained, time-synchronized manner. Prior to the z13, there was a limited set of power management related data, on coarse-grain time scales, and in an asynchronous manner. These data could help interpret the power and thermal profile of the prior z Systems in only a limited manner, especially in terms of being able to relate that to the performance data being collected elsewhere. The new systems approach on the z13 collects data from just under 300 sensors that are precisely synchronized to cover all these categories via a single point of collection and control. A microcontroller component contained in the z13 CP chip is dedicated for this purpose and operates on a fine-grained time scale of 16 ms per sensor update.

The remainder of this paper is organized as follows. In the next section, we provide a general overview of the z13 power management architecture. It includes subsections on key component subsystems within the overall functional specification of the power management facility. For example, sub-sections on alternative methods for digital sensing of power and droop events (e.g., the so-called power-proxy apparatus and critical path monitors (CPM)) are included within this power management overview. The section “Power-management characterization through automated microbenchmark generation” is devoted to pre-silicon and post-silicon characterization, tuning, and stress-test methodology via automated microbenchmark generation. Individual subsections within this section are devoted to key methods within the MicroProbe automation toset that are relevant to noise stress generation and the training (or calibration) of power-proxy digital sensors architected to sense the level of active compute resource utilization at short and long time scales (respectively). These are described in more detail later on as part of the core-level power management functions.

z13 power management architecture is implemented as a two-level hierarchy: core-level and chip-level. Most of the functionalities implemented (at either level) are dedicated to detect situations where there is excessive power consumption or reduced voltage margin. The performance throttling mechanism (PTM), implemented at core-level, is the only actuator mechanism. This actuator is triggered locally by mechanisms implemented at the core level. Globally, the per-core PTM actuators can be triggered by control functions implemented at the chip level. The chip is instrumented with many sensors for platform characterization, providing accurate and fine-grained measurement of the chip activity. The remainder of the section provides more details of the power management architecture.

**Power-management hardware-firmware microarchitecture**

Figure 1 depicts the hardware-centric data flow between the processor cores and the chip-level power-management logic. It covers the computation of triggers for performance throttling and the critical path monitor (CPM) connections to the PTM. The CPM is a circuit-level sensor [8, 9] that effectively provides a gauge of the amount of timing margin available at any given time within workload execution. Typically, the largest loss of circuit timing margin occurs during a voltage droop event. Other sources of consideration in such loss include: localized thermal hot spots, circuit aging over time, critical path timing profile changes due to dynamic voltage-frequency scaling (if applicable), and so on. In the z13 design, the other effects mentioned are either too small or non-existent by virtue of particular design choices.

If the voltage detected droop exceeds a pre-determined threshold level, dynamic actuation of the droop mitigation control logic is asserted. In principle, this could be achieved via adjusting the clock frequency via digital phase-locked loop (DPLL) controls as in [3, 4] or, as is chosen for initial implementation in the z13, via instruction throttling which is similar to the mechanism used in [5]. The short-term activity proxy (STAP) and long-term activity proxy (LTAP) are activity counter based digital sensors architected to sense the level of active compute resource utilization at short and long time scales (respectively). These are described in more detail later on as part of the core-level power management functions.

The power management in the z13 CP is done using hardware-firmware (HW-FW) co-programming. In HW, the pervasive region within the CP/SC (where SC refers to the System Controller) chipset [6] contains a highly optimized microcontroller called the Self-Boot Engine (SBE), so named because the design was first used in the POWER8® microprocessor to assist or replace the “boot” function provided by the flexible service processor (FSP)
in Power Systems [10]. The SBE has its own instruction set architecture (ISA) that supports (load/store) instructions that translate to SCOM (scan communications) registers’ read/write commands, ALU instructions [e.g., ADD, SUB, AND, OR, XOR, Rotate (left/right)], branch instructions, and others. Note that the SCOM unit serves as an interface to read and write on-chip processor registers. Off-chip data and control information is read into the processor via SCOM registers. In addition, SBE uses a local, on-chip SRAM memory (also present in the pervasive region) called PIBMEM as its code/data memory. SBE acts as the centralized power-management controller for the chip. In addition to SBE, there is a dedicated core-level power-management block in each core and a chip-level power-management block in the “uncore” or nest region that perform specialized power-management functions like power-proxy measurements, CBCA (cycle-by-cycle activity) detection, core-throttling and various modulation and control utilities associated with the CPM-sensed data (referred to as “CPM Filter and Thresholding” in Figure 1). Note that on a multi-core processor chip, the units comprising non-local (or shared) caches, interconnect elements, memory controllers and other non-processor-core functions are collectively referred to as the “nest” in IBM terminology. The rest of the industry typically calls this the “uncore.” There are also temperature sensors, skitters, and CPMs that are placed at various points in the chip, which can be monitored by the SBE.

(The skitter [11], which is a precursor to the CPM, is a special circuit macro that serves as a simplified circuit delay sensor that is also useful in detecting voltage noise.)

Even though the SBE microcontroller off-loads the support element (SE) and the FSP quite significantly, there is firmware within the FSP that initially activates the SBE, reads hundreds of time-synchronized sensors from SBE’s memory (PIBMEM), and performs error handling and other high-level functions.

Core-level power management functions
In this subsection, we provide an overview of the core-level power management functions within the overall z13 power management architecture.

PTM function
The performance throttling mechanism (PTM) is responsible for reducing system performance in order to mitigate unwanted, resilience-threatening situations (e.g., excessive power consumption or voltage droops). Whenever the PTM mechanism is triggered, the execution pipeline is throttled, reducing the power consumption and the voltage droop. After the throttling event, a ramp-up function starts “un-throttling” the pipeline until the normal execution rate is re-established. The mechanism is fully configurable, providing parameters to control the degree of the initial throttling, the duration, and the ramp rate. This mechanism also supports sustained throttling events.

---

**Figure 1**
The z13 CP chip power management high-level data flow. (CPM: Critical Path Monitor; CBCA: Cycle-by-Cycle Activity; STAP: Short-term Activity Proxy; LTAP: Long-term Activity Proxy; ISU: Instruction Sequencing Unit; IFU: Instruction Fetch Unit; LSU: Load-Store Unit; VSU: Vector/Scalar Floating Point Unit; Max: multiplexer.)
That is, while still in the ramp-up phase, a new throttling event can be triggered and this would restart the whole process. The ramp function is necessary to prevent the PTM logic from inducing a voltage droop event when the throttling is released.

**Throttle metering**

In order to validate the PTM function, a throttle metering capability is implemented. This measures the degree to which PTM is invoked. This is crucial to be able to properly calibrate the power management functions and to understand the impact of throttling on performance measurements. Since the PTM function could affect the system’s performance, it is designed to be triggered only in extremely unlikely scenarios. This strategy ensures minimal impact to real system performance.

**Activity proxies**

Activity proxies constitute a well-known technique to estimate or track the power consumption profile of processor chips [12–16]. In the z13 design, they consist of a set of 16 activity counters. These counter values are multiplied by weights (and then summed up) to provide an estimation of the core (and then the chip) power consumption. One important step in defining activity proxies is the selection of the input activities to be monitored. The activity counters are selected with the aim of covering the different types of activities that can be generated in the core. The counters cover various functional units (fixed point unit, floating point unit, load-store unit, etc.), instruction cache, data cache, branch target buffer, completed instructions, dispatched instructions, etc. Another factor that affects the temporal sampling granularity (and hence the number of power estimations per second), as well as the estimation accuracy, is the number of activity counters that are used as input. As the number of activity counters increases, the accuracy of power estimate improves. However, it takes longer to generate an estimation. The z13 CP chip improves on prior activity proxy designs by implementing three different activity proxies. This enables a wider timing window selection to support power management functions at different timescales. Additionally, the z13 provides hardware to sum across the core activity counters to produce a chip-level activity count. We describe the activity proxies in the rest of this sub-section.

The long-term activity proxy (LTAP) uses all 16 activity counters as input. Therefore, it provides a more accurate estimation at the expense of slow estimation speed. This proxy is used to provide power characterization support via the firmware interface. The final product selection mode is to use the shortest time scale of 32 $\mu$s for the LTAP computation. The time period of 32 $\mu$s is determined by the firmware interface that is needed to manage power at the chip-level, spanning multiple cores with low-speed serial communication links. The result is a relatively slower time-constant gauge of the power at the chip-level. The 16-counter hardware apparatus to estimate per-core power is natively much faster, of course. The sampling period for doing the weighted sum accumulation is determined by the number of bits in each counter. This counter-width is chosen in such a manner that the event activity counts monitored are large enough on average to provide statistical significance in computing the weighted sum. As such, LTAP provides the highest achievable level of accuracy in the chip-level power capping function implemented within the SBE. This is a practical implementation choice, because it is used in conjunction with the CPMs that detect voltage droop. If there is a rapid increase in power that happens on a time scale that is smaller than 32 $\mu$s, then before the LTAP can engage, the regulator voltage output starts to drop due to the current draw. This engages the CPM throttling until the longer-time-scale LTAP could catch up and start to throttle for any sustained capping requirements.

The short-term activity proxy (STAP) is similar to the LTAP, but it uses only 8 counters, trading off some accuracy to improve the estimation speed. The appropriate set of counters has been selected using techniques successfully used in previous IBM chips [14]. The relatively fast and accurate estimations make it a potential choice in producing higher speed power estimations, if or where needed. STAP might also serve a role in future chips (for rapid current change slope detection) where CPMs are not present, but the accuracy of the STAP might need to be refined in that case.

The cycle-by-cycle activity (CBCA) proxy is a very fast proxy, capable of making estimates at a very fast rate ($<10$ ns), when considering the hardware sampling interval. Firmware overheads to use this for core- and chip-level throttling are minimal, because the CBCA-driven PTM throttle is in effect a hardware-autonomous control loop within the larger firmware-driven power management control loop. The CBCA uses 4 activity counters as input, and therefore, its lower accuracy makes it unsuitable for performing chip-level power estimations. However, it is ideal for triggering the PTM function whenever a steep change in activity is detected at the chip-level. A fast triggering mechanism is crucial to reduce the voltage droop produced due to changes of activity at the chip-level.

**Critical path monitors (CPM)**

We use Critical Path Monitors (CPMs) in the z13 to measure available timing margin. These CPMs are refinements of the CPMs implemented in previous IBM chips [8, 9]. CPMs allow for measuring the timing margin and therefore are very sensitive to voltage droops. CPMs...
have been used in the past for actively reducing the voltage guard-band to save energy [3, 4] and to protect the chip from voltage droop events [5]. In the z13 architecture, they are used in a similar fashion, but instead of triggering voltage/frequency adjustments, the PTM function is triggered whenever a certain (configurable) threshold is met. Adjusting the voltage would be a relatively high-latency operation, since off-chip voltage regulation is used; adjusting the frequency can be quite fast, but it requires careful control of clock skew and jitter during ramp-down or ramp-up operations. As such, design verification under variable frequency operational modes can become quite complex in products that require very high reliability. The PTM knob was selected for voltage droop management in the z13.

**Chip-level power and thermal management functions**

In this subsection, we describe the chip-level power management logic (CLPM) that is implemented in z13 with a focus on the implementation macros outside the core within the pervasive infrastructure.

**Chip-level CBCA-based throttle-event generation**

Individual CBCA values from the cores are summed up at chip-level, and the current sum is subtracted from the average of the previous 8 CBCA values. The resulting difference is compared with a threshold. If the computed delta is above the threshold a PTM instruction-throttle-event is generated.

**Chip-level activity proxy**

In order to monitor and manage the overall power consumption of all cores in the chip, a chip-level activity proxy is implemented. For this chip-level activity proxy, two different modes exist: the short-term mode and the long-term mode.

In the short-term mode, every time a core STAP value is updated, this value is sent via a serial link to the chip-level, where all of the cores’ STAP values are added together to form a chip-level STAP (CLSTAP). The short-term mode is useful for performing power-capping by hardware at the chip-level; thus, it enables effective management of chip power budgets.

In the long-term mode, every time a core LTAP value is updated, this value is sent via a serial link to the chip-level, where all of the cores’ LTAP values are added together to form a chip-level LTAP (CLLTAP). The long-term mode is useful in generating a chip-level activity number that might allow firmware to find hot spots in a system that would trigger selective throttling or power shifting between chips/nodes.

The hardware implements a multiplexor (MUX) that selects either the long-term or the short-term activity proxy in the core logic. As described before, the LTAP is chosen for high-accuracy power-capping decisions at a longer time scale; whereas, in principle, the STAP could be selected for smaller time-scale decisions where higher speed (but less accurate) power estimations are needed. Any update in the selected value at the output of the MUX is transferred serially to the chip-level logic. The serial link always transfers the maximum number of bits. The chip-level logic forms a 32 bit number and uses an adder tree for this to reduce latency and hardware complexity. An overview of this structure is depicted in Figure 1.

**Chip-level power capping**

The power capping function ensures that the chip continues to function even if the chip tries to exceed its power budget. The z13 CP chip has the option to use either STAP or LTAP as the basis for capping. The benefit of using activity proxies instead of true power measurement for power capping, is that the performance is deterministic and repeatable for the same workload across different chips and different environmental conditions. This supports the z13 goal of very high reliability in that the performance for a workload under the power fault conditions can be bounded and does not vary across systems or chips. One way to dynamically adjust the power capping limits in a running system is to use the SBE to adapt the power capping threshold real-time based on chip internal information, which is accessible by the SBE micro. Alternately, the SBE can actuate a sustained throttle level determined periodically by firmware rather than using the automated hardware throttle mechanisms. This allows the option of using a firmware-based feedback controller for power capping similar to those used in prior IBM systems [17]. The chip power estimation within a time period is computed as follows:

\[
\text{ChipPowerEstimation} = \sum_{i=1}^{#\text{cores}} \left( \frac{\sum_{j=1}^{#\text{events}} C_{ij} \times w_j}{R} + L - f(T_i) \right)
\]

where \( C_{ij} \) is the count of event \( j \) in core \( i \) within the interval, and \( w_j \) is the corresponding weight of the event. \( R \) is a scalar factor to convert the core activity proxy value to units of power consumption. \( L \) is the worst-case leakage power (as obtained from the chip’s vital product data (VPD) repository), and \( f(T_i) \) is the additional power reduction due to operating core \( i \) at throttle level \( T_i \). The weighted sum of core events is implemented in hardware, while firmware must account for the factors of \( R, L, \) and \( T_i \). Note that the worst-case leakage value \( L \) takes into account the maximum temperature bounds, and is valid for the nominal supply voltage level of the chip. The factor \( R \) incorporates the effect of the particular operational voltage and frequency settings of the chip.
Since STAP and LTAP track only switching power, the power estimation includes margin for leakage power, which is modeled using worst-case manufacturing assumptions. Once throttling is invoked, the power estimation is further adjusted to account for some power reduction due to lower clock-power during the “throttled” time duration. Specifically, the reduction in the \( C_{ij} \) activity values resulting from throttling does not account for the additional leakage power reduction caused by the drop in temperature. The \( f(T_i) \) adjustment accounts for this correction.

**Thermal control**

For thermal control, the pervasive-logic-centric microcontroller (SBE) reads the two hottest digital thermal sensors (DTSs) from each core in 16-ms loop intervals. It computes the average of the two, compares against a threshold value, and throttles that core in case the temperature exceeds the threshold for the next 16 ms. The SBE releases the throttle if the temperature falls below the threshold. This sequence continues every 16 ms.

**Power management microcontroller (SBE)**

In this subsection, we provide details of the SBE microcontroller operational semantics. The Self-Boot Engine is a highly specialized microcontroller with its own ISA (instruction set architecture) catering towards on-chip communication and initialization operations. The SBE in the z13 continuously runs a program out of its dedicated on-chip SRAM memory (PIBMEM), collecting data from 290 sensors in the CP chip, as well as 30 on the SC chip, in a 16 ms interval loop. For each sensor, the SBE records the instantaneous (average, maximum, minimum), accumulator, tick-counter, and squared accumulator value fields in its memory. This data is read by the system firmware once every hour, processed, and used for real time logging for running machines. The tick-counter keeps track of how many 16-ms sensor values were added to the accumulator and squared accumulators. This permits IBM to compute the hourly average and three sigma values from these sensor data structures. All 290 sensors are gathered with time synchronicity within 16 ms or less of one another. All the hourly maximum and minimum values have a time stamp that is accurate to within 64 ms for when the event happened during the hour. In addition, there is a specialized “snapshot” mechanism in the firmware that allows for the capture of a group of 88 critical sensors in their current state (out of the entire 290) for a given 16-ms interval. There are two modes of group capture. In the IBM lab environment, one can capture this group of sensors every 16 ms and create a temporal plot of the time-synchronized behavior of all 88 sensors by streaming the data out through the system control structure. The second mode is “application” mode in which the “snapshots” are taken based on 5 predetermined classes of sensors hitting maximum-value conditions in the 1 hour interval. Every hour, these 5 snapshots are returned with all the normal sensor data. However, the snapshots are time-stamped and can be directly correlated with a critical moment in time during that hour when a class of sensors peaked, allowing one to see what the other 88 sensors in the system’s current instantaneous state were.

**Power-management characterization through automated microbenchmark generation**

In this section, we provide details of the methods used to test the functionality and operational “corners” of the various features within the power management architecture. The methodology is centered around the use of an automated microbenchmark generation toolset called MicroProbe [12].

**Overview**

Calibration and characterization of power-management mechanisms in a pre-silicon setting is typically inadequate. This is because it is difficult to model power and inductive noise behavior very precisely. Therefore, a systematic, direct-measurement-based calibration and characterization of power management mechanisms in a post-silicon setting (during the processor test and “bring-up” process) is required to firm up the calibration settings, voltage levels, and package characteristics that ensure robust power-management functionality.

A key part of such a post-silicon processor testing process is the use of specially crafted microbenchmarks to calibrate and test the power-management functionalities. These corner-case workloads are sometimes referred to as stressmarks in the research literature. The stressmarks maximize the voltage noise and power consumption or generate different microarchitecture activities to train or test activity proxies. Manual, expert-driven generation (and fine-tuning) of such stressmarks can be a tedious and error-prone process. To overcome these issues, we used the MicroProbe [12] framework to systematically generate the stressmarks required during the z13 post-silicon test/bring-up process. Automation in stressmark generation has been attempted by other research groups, but as discussed in [12], MicroProbe has significant new features, tested in real machine contexts, that have not been available before. The recent work of Kim et al. [18] is an example of inductive noise stress test generation that was used in an AMD processor context. However, MicroProbe uses a novel thread alignment technique that achieves superior results in a mainframe processor execution environment.

**Noise stressmark generation**

In order to reduce the voltage margin, the cores need to be able to detect voltage droop and trigger capping in case
the margin is insufficient under extreme situations. Therefore, the CBCA and CPMs must be tested on a wide set of cases, including the worst-case ones. Therefore, a complete noise characterization is needed. We used the methodology tested on the IBM zEnterprise EC12 (zEC12) for the noise stressmark generation and characterization [13]. This methodology permits the control of different parameters affecting the voltage noise, providing a complete understanding of the voltage noise and the noise mitigation mechanisms (CBCA/CPM).

The noise stressmarks are built by concatenating high-power instruction sequences with low-power instruction sequences. This generates transient fluctuations of voltage in the power delivery network. The frequency of changes between the high/low instruction sequences—which can be controlled via the sequence length—affects the overall voltage droop generated, achieving its maximum at the power delivery network resonance frequency. Additionally, the transitions between high-power to low-power are synchronized across the cores in order to maximize the voltage droop. There are a few alternative schemes that were tried in making the inter-core synchronization as exact as practically feasible. One method that was used with success in the zEC12 experiments [13] utilizes a particular instruction at specially crafted frequency that uses the system clock as a reference to affect the synchronization. In the z13, other techniques tied to spin-lock synchronization mechanisms in a hardware cache coherence protocol context were also exploited to achieve this objective.

Maximum and minimum power instruction sequences were derived via a systematic approach [12, 13]. First, every instruction of the ISA is profiled in order to generate an energy-centric instruction ranking. These ranks define which instructions are more power-hungry and which ones consume low power. The maximum-power instruction sequence is derived after (automatically) exploring a wide set of combinations of the most power-consuming instructions for each functional unit. Once the sequence is defined, memory and branch activity is added to further maximize the power consumption. Not only has this systematic, automated approach been proved to outperform previous manual efforts, but it also enables the definition of the maximum-power stressmark in the early stages of the testing process. The manual approach is inadequate because it cannot provide the search coverage afforded by automated generation of many thousands of candidate maximum-power instruction sequences. The systematic, automated search proves to be much faster, and it generally also yields a higher-power sequence than that achieved manually by any expert within the design team. Minimum power instruction sequence definition is also driven by the energy-centric instruction ranking methodology. It is derived using the last instruction within the ranking table. Once maximum and minimum instruction sequences are defined, noise stressmarks are generated by concatenating them, as mentioned before.

**Activity proxy training**

CBCA and short/long-term activity proxies must be calibrated in order to provide accurate power consumption estimations. The calibration methodology uses statistical regression techniques and genetic algorithm searches to find the optimal set of events (that are to be monitored via activity counters) and their weights for a given training set [14]. A proper training set definition is crucial to find a solution with large coverage (i.e., accurate estimates under a diverse range of execution environments). We used MicroProbe to implement a systematic method to generate rich training sets for power proxies [12, 15]. We generated low-to-high activity stressmarks for each functional unit as well as for each possible combination of functional units. In addition, several stressmarks covering memory activities were also automatically generated using MicroProbe. Overall, the training set was composed of more than ten thousand stressmarks with unique microarchitecture activity, covering all types of activities.

**Power-management results and validation**

This section covers some representative results of power management features and characterization efforts for the z13.

**Figure 2** compares the chip power values of a microbenchmark generated by the MicroProbe methodology against two benchmarks (hmmer and astar) selected from the CINT2006 benchmark suite that is part of the larger SPEC CPU2006 suite. These two are chosen for illustration because they are amongst the highest power workloads within the SPEC CPU2006 suite. The figure also includes the case of a microbenchmark that is generated through a manual approach. Chip power values are normalized to the MicroProbe-generated worst-case benchmark. Overall, the MicroProbe methodology generates a benchmark that can be up to 18% higher (“HMNER ST” in Figure 2) power than a worst-case real workload. The methodology enables us to set accurate maximum power targets, ensuring a more robust product. It also enables rapid bring-up through automated generation of microbenchmarks in contrast with manual generation. Although not shown in the figure, MicroProbe also generates a noise stressmark that has 70% higher noise compared to a typical functional stressmark.

**Figure 3** shows experimental results that demonstrate the ability of the z13 LTAP to track the chip $V_{dd}$ power. In this experiment, we ran several steady workload exercisers that stressed a z13 chip in different ways and measured the $V_{dd}$ power rail, chip temperature, and LTAP activity counters. Since LTAP does not track leakage power, we subtracted it from the power rail measurement...
to train the LTAP. Our leakage model was formed by measuring the power while the chip was idle and correcting it for the average temperature while running each workload. We used a genetic algorithm to train the LTAP coefficients. This algorithm is described in our prior work [14]. The genetic algorithm optimizes the coefficients to find the best fit for reducing the average error in power across all workloads.

The results are shown in Figure 3. Each workload is represented by a point on the plot. The x-axis shows the chip supply voltage ($V_{dd}$) rail power measured for the workload, scaled to the workload with the highest power consumption. The y-axis shows the absolute error of the leakage power estimated for the workload summed with the additional power estimated by the LTAP. The error includes both the error in the leakage power model and the LTAP error. The average absolute error across all workloads is 1%. The highest power consumption workloads show errors of up to 2%, which is similar to the expected error on hardware-based measurement in servers [14]. Although only two of the workload exercisers were in the “highest power” category in this particular experiment, our general experience with the very highest power workloads shows that the error margin for those is similarly small (i.e., < 2%). Power capping requires high accuracy measurement near the upper end of the power consumption range, since capping is not required for system protection at low power consumption rates. Therefore, we believe the power proxy is appropriate as a system indicator to be used for the power capping capability in the z13.

Figure 4 shows actual oscilloscope measurements with and without CBCA mechanism enabled on a specially crafted microbenchmark that stresses the noise behavior on the chip. For this particular microbenchmark, all cores go through from a worst-case power consumption phase to an almost idle power consumption phase, thereby creating a high power noise in the system. The CBCA mechanism captures the transition in power behavior and reacts accordingly to reduce noise. The y-axis shows the voltage measurements that are normalized to the average voltage value seen across the run. As Figure 4 shows, the CBCA mechanism achieves considerable noise reduction. As stated before, the absolute accuracy in power estimation is not a required goal for the CBCA sensor; rather, the speed with which a power gradient is sensed is more important. As a result of power estimation inaccuracies, the CBCA-enabled voltage droop may, on rare occasions, be slightly greater than the CBCA-disabled voltage droop, but overall, the worst-case voltage droop is reduced. This means that the worst-case minimum voltage (due to noise) experienced at the circuit-level is higher than before; hence the frequency guard band can be made smaller. This
naturally translates to higher frequency (and performance) for the given nominal voltage.

Figure 5 shows the maximum and average temperature values from the 5 digital thermal sensors (DTS) implemented on each core for different throttling levels. The data is normalized to the maximum temperature seen and the workload run is the MicroProbe-generated maximum power stressmark. The throttling mechanism reduces the temperature significantly (> 40%) when high throttling levels are used.

Figure 6 shows the improvements in $V_{\text{min}}$ (minimum operating voltage) across different chips with CPM enabled. The data is normalized to the voltage data from the first chip (chip 1) where the CPM mechanism is disabled. The workload is a specialized functional test to stress different circuit paths. This workload was a manually-crafted high resource utilization stressmark; it is not the worst-case $di/dt$ inductive noise stressmark generated by MicroProbe. Depending on the chip that is being characterized, $V_{\text{min}}$ improvements range from 4.7% to 6.3% with CPM mechanism enabled. It should be noted that in Figure 6, the frequency of operation is considered to be fixed, and $V_{\text{min}}$ in this context refers to the minimum operating voltage that can be sustained without circuit
timing failure. Without CPM, a very conservative guard band has to be added to the voltage corresponding to the target frequency. Hence, the minimum permissible operating voltage is higher. With CPM enabled, the principle of dynamic guard banding [4] is invoked—so the minimum permissible voltage point can be pushed down to a lower value. In effect, this is a case of under-volting. In the rare case where the circuit voltage is sensed to approach the new (lower) $V_{\text{min}}$, PTM is actuated to circumvent the potential voltage emergency.

Related work and discussion
The z13 power management architecture is an innovative, new feature in z Systems. Many of the core ideas related to power estimation and dynamic guard banding have evolved from prior IBM designs within the Power Systems family [3, 4], as we already stated. However, the particular combination of three different activity proxies to account for decisions at various time scales has not been attempted before.

The concept of using digital counter-based sensors to estimate power has been used in non-IBM processor products as well. Initial attempts at monitoring on-chip power using analog circuitry, as incorporated in Intel’s Montecito (Itanium) design [19], reportedly met with calibration and accuracy issues. Subsequent Intel processor designs (Nehalem [20] and Sandy Bridge [21]) as well as AMD’s Kabini chip [22] all used digital power metering methods for on-chip power management.

When it comes to the z13 power management architecture, a key difference is that previous Power System products, as well as other non-IBM offerings have used complex digital power proxy implementations that use a relatively large number of activity counters for precision. In the z13, the emphasis was on using lower-complexity activity proxies to detect power change deltas. In other words, the focus was not so much on accurate power estimation on a per-core basis; rather, the objective was to set a tighter bound on maximum permissible power consumption, and to reduce the noise-related guard band. The overriding concern was that system resilience should not be compromised in the attempt to lower the power cost for the targeted product performance.

The mainframe product space has historically been focused on high performance and high system-level utilization through virtualization of hardware CPU resources. However, in recent product cycles, z Systems have encountered the power wall in the form of current delivery limits, even if dissipation limits have been adequately met (or even extended) through advanced liquid cooling and/or proprietary packaging technology where needed. The growth in processor core count, spurred by significantly smaller single-thread performance growth has further exacerbated the power delivery problem. At the same time, due to workload diversity across multiple cores and improved degrees of active power reduction via clock-gating, the maximum power swing and attendant inductive noise levels have been on the rise.

In view of the prioritized focus on hardware reliability in z Systems, the new problem of managing power consumption had to be addressed in a manner that also met the challenge posed by the inductive noise-related voltage droops. These motivations resulted in the idea of introducing a new power management unit that would set realistic limits on maximum power consumption, while also reducing the effective voltage-related guard band to protect against noise. As such, the z13 power management architecture represents a unique blend of concepts in power conservation coupled with those that provide immunity against voltage noise.

Conclusion
In this paper, we presented a summary overview of the key features within the z13 power management architecture. As mentioned at the outset, the driving principle of this design has involved how to provide robust power management, without compromising real customer workload performance. The objectives were realized using an approach that avoids the worst-case voltage margins and overly conservative maximum-power assumptions employed by previous generations. Instead, the voltage guard band was lowered by relying on timely sensing and responding to voltage droop events to avoid circuit errors. This reduction in chip operating voltage by using dynamic throttling of instruction execution, and the underlying feedback control system, was stress-tested for robustness using an automated worst-case voltage noise generation methodology. Using automated
microbenchmark generation, we were able to more effectively tune our activity proxies and, by generating worst-case noisy workloads, to more effectively characterize and stress test the chip to ensure the remaining voltage margin is always sufficient for correct operation. Illustrative experimental results demonstrated the “efficient resilience” principle that was pursued effectively in this project where the chip, instead of being passively subjected to the workload running on it, is an active participant in ensuring optimal system performance.

Acknowledgments
The MicroProbe tool development work at IBM Research was partially sponsored by Defense Advanced Research Projects Agency (DARPA), Microsystems Technology Office (MTO), under contract no. HR0011-13-C-0022. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. This document is “Approved for Public Release: Distribution Unlimited.” Many individuals across IBM made this work possible. We extend particular thanks to Tilman Gloeckler, Mitesh Agrawal, Lee Eisen, Huajun Wen, and Karthick Rajamani.

References
Dr. Strach specializes in chip packaging, has been involved in post-silicon hardware test, characterization and power noise analysis, and associated simulation-based modeling. He is a member of the IBM zEnterprise EC12 and the IBM z13 processor products.

Preetham M. Lobo
IBM Systems, Bangalore, 560071 India
(preelobo@in.ibm.com). Mr. Lobo is a member of the pervasive and power-management logic design team for z System chips. He has worked on logic design in the IBM POWER7+ and POWER8 processors, and on power management logic design for the IBM z13. In addition, he was part of the pervasive “bring-up” team for the IBM zEnterprise EC12 and the IBM z13 processor products.

Ramon Bertran
IBM Research, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (rbertra@us.ibm.com). Dr. Bertran is a Research Staff Member in the Reliability- and Power-Aware Microarchitectures department. He has been involved in research and development work in support of IBM Power Systems and z Systems. His current focus is on power-aware and noise-aware computer architectures.

Gerard M. Salem
IBM Systems, Williston, VT 05495 USA (gsalem@us.ibm.com). Mr. Salem specializes in logic design and hardware debugging, and is involved in systems “bring-up” and integration as part of the overall system assurance and validation exercise prior to product readiness and customer shipment.

Malcolm Allen-Ware
IBM Research, Austin, TX 78758 USA (mware@us.ibm.com). Mr. Allen-Ware is a Distinguished Engineer who has been involved with power-management techniques across all of IBM systems including POWER® (Power System), mainframe (z System), and storage-related system offerings. In addition, he is now involved in Watson Software as a Service (SaaS) and POWER8 Cloud Infrastructure-as-a-Service (IaaS) based optimizations for SoftLayer® GTS Data Centers.

Richard Rizzolo
IBM Systems, Poughkeepsie, NY 12601 USA (rizzolo@us.ibm.com). Mr. Rizzolo is a Senior Technical Staff Member who has served as the z13 characterization lead. He specializes in test, characterization, and diagnostics as part of system assurance and validation prior to product shipment.

Sean M. Carey
IBM Systems, Poughkeepsie, NY 12601 USA (smcarey@us.ibm.com). Mr. Carey is part of the hardware design team for z System processor products and has also served to help as part of the “bring-up” and characterization team for the z13 product.

Thomas Strach
IBM Systems, 71032 Büblingen, Germany (starch@de.ibm.com). Dr. Strach specializes in chip packaging, power noise analysis, and associated simulation-based modeling. He has been involved in post-silicon hardware test, characterization and “bring-up” related work for the z13.

Alper Buyuktosunoglu
IBM Research, Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (alperb@us.ibm.com). Dr. Buyuktosunoglu is a Research Staff Member in the Reliability- and Power-Aware Microarchitectures department. His current focus is on microarchitecture definition and robust power management for Power Systems and z Systems.