Modern Hardware Margins: CPUs, GPUs, FPGAs

Recent System-Level Studies

Dimitris Gizopoulos
George Papadimitriou
Athanasios Chatzidimitriou
University of Athens
{dgizop, georgepap, achatz}@di.uoa.gr

Vijay Janapa Reddi
Harvard University
vj@eecs.harvard.edu

Jingwen Leng
Shanghai Jiao Tong University
leng-jw@sjtu.edu.cn

Behzad Salami
Osman S. Unsal
Adrian Cristal Kestelman
Barcelona Supercomputing Center
{behzad.salami, osman.unsal, adrian.cristal}@bsc.es

***Abstract*—Modern large-scale computing systems (data centers, supercomputers, cloud and edge setups and high-end cyber-physical systems) employ heterogeneous architectures that consist of multicore CPUs, general-purpose many-core GPUs, and programmable FPGAs. The effective utilization of these architectures poses several challenges, among which a primary one is power consumption. Voltage reduction is one of the most efficient methods to reduce power consumption of a chip. With the galloping adoption of hardware accelerators (i.e., GPUs and FPGAs) in large datacenters and other large-scale computing infrastructures, a comprehensive evaluation of the safe voltage reduction levels for each different chip can be employed for efficient reduction of the total power. We present a survey of recent studies in voltage margins reduction at the system level for modern CPUs, GPUs and FPGAs. The pessimistic voltage guardbands inserted by the silicon vendors can be exploited in all devices for significant power savings. Voltage reduction can reach 12% in multicore CPUs, 20% in manycore GPUs and 39% in FPGAs.**

Keywords—voltage margins, power consumption, energy efficiency, multicore CPU, many-core GPU, FPGA, accelerators.

I. Introduction

Process variations can affect the dimensions of the transistors (length, oxide thickness, etc.) due to the modern fabrication process, and thus, can impact the threshold voltage of a MOS device [1]. Such *static* variations remain constant after the release of the chip to the market. On top of that, transistor aging and *dynamic* variation in supply voltage and temperature, caused by different workload interactions can also affect the correct operation of a chip. Accounting the different types of variations, silicon vendors offset the best-case supply voltage with a fixed guardband to ensure reliability under worst-case conditions, as shown in Fig. 1a. The guardband results in faster circuit operation than required at the target frequency for typical workloads, which results in additional (thus wasted) cycle time, as shown in Fig. 1b. In case of a timing emergency caused by voltage droops, the extra margin prevents timing violations and failures by tolerating circuit slowdown. While static, worst-case guardbanding ensures robust execution for virtually all circumstances, it severely affects power and energy efficiency of the average case [2].

Supply voltage reduction (Fig. 1c) is one of the most efficient techniques to reduce the power consumption of the chip, because dynamic power is quadratic in voltage. Several system-level approaches have been proposed to predict and effectively utilize the safe operation limits (i.e., V_{min}) of the microprocessors. For example, the authors in [3] [4] propose an approach to predict the large voltage noise droops. Along the same lines the authors of [5] [6] proposed a firmware-based approach to predict the lowest safe voltage operation by observing corrected errors manifested on caches of an Intel Itanium processor. The energy gain in these studies comes from the variations of the V_{min} when the same workload runs on different cores (core-to-core variation) or different workloads run on the same core (workload-to-workload variation).

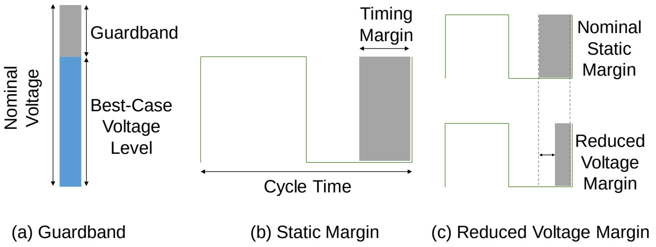


Fig. 1. Voltage guardband ensures reliability by effectively inserting extra timing margin. Reduced voltage margins can improve total system efficiency if they don’t affect reliable operation.

Similar to multicore CPUs, many-core GPU architectures also require a large voltage guardband for reliable operation under all types of variations. However, their massive nature as well as their distinctive microarchitectural features render the traditional CPU-centric analysis framework and solutions unsuitable for GPUs. As such, prior work focuses on manycore-centric modeling, analysis, and smoothing techniques [23] [24] [25].

Furthermore, FPGAs are getting increasingly popular as acceleration platforms thanks to their massively parallel architecture and the capability of stream-fashion computation and data marshaling. Due to the development of High-Level Synthesis (HLS) tools in recent years, it is expected that FPGAs will be exploited in 1/3 of data centers by 2020 [15]. However, their power consumption is still a key concern, especially when compared against equivalent ASIC designs.

In this paper, we summarize recent system-level analysis and evaluation of safe voltage margins in multicore CPUs, manycore GPUs, and FPGAs. We aim to present consolidated results and observations for heterogenous architectures and summarize the emerging trends in hardware margins and energy-efficiency.

For the multicore CPUs part of this paper, we focus on two recent state-of-the-art ARMv8-compliant microprocessors. By experimenting on these recent multicore microprocessor chips, we present a number of observations which can potentially improve energy-efficiency in future designs. We report up to 18.4% power reduction in single-core executions and up to 17.6% in multicore CPU executions [8]-[14]; the multicore

CPUs part of this survey was conducted in the context of the UniServer project [9] [14]. For the manycore GPUs part of this survey, we focus on a wide range of four NVIDIA GPUs spanning two generations (Fermi and Kepler). We report the V_{min} points of different programs and we show as large as 20% voltage guardbands on these GPUs; the energy efficiency improvements through voltage reduction can be as large as 25% [24]-[28]. We also report the experimental evaluation of aggressive undervolting in FPGAs. By experimenting on real FPGA fabrics, we show the significant effectiveness of this technique to reduce their power consumption by on average 90%, first, by eliminating the voltage guardband which is measured by on average 39% and also by trading-off the power-reliability in further lower voltage levels [16]-[20]. The research on FPGA's undervolting is being conducted under LEGaTO project [21] [22]. Table I below presents the consolidated information about the voltage reduction and power savings potential for all the platforms of this study.

TABLE I. MINIMUM/MAXIMUM VOLTAGE AND POWER REDUCTION FOR THE CPUS, GPUS, FPGAS OF THIS STUDY.

Platform	ISA / Family	Process Technology	V_{dd} (mV)	Voltage Reduction	Power Reduction
X-Gene 2	ARMv8	28nm	980	6.1% - 11.7%	11.6% - 18.4%
X-Gene 3	ARMv8	16nm	870	4.6% - 11.5%	10.9% - 17.6%
GPU	Kepler	28nm	1090	9.2% - 18.3%	8% - 25%
	Fermi	40nm	1090	11.6% - 20.3%	7% - 22%
FPGA	VC707	28nm	1000	41.2%	89.2%
	ZC702	28nm	1000	42.8%	89.8%
	KC705	28nm	1000	38.5% - 42.8%	87.2% - 90.1%

II. EXCEEDING GUARDBANDS IN ARMV8 MULTICORE CPUS

A. *Applied Micro's X-Gene 2 and X-Gene 3 Specifications*

Applied Micro's (now Ampere Computing) X-Gene 2 and X-Gene 3 multicore CPUs consist of 8 and 32 64-bit ARMv8-compliant cores, respectively. Both CPUs offer high-end processing performance. X-Gene 3 microprocessor has a main power domain that includes the CPU cores, the L1, L2 and L3 cache memories, and the memory controllers, which is called PCP (Processor ComPlex) power domain. X-Gene 2 has a similar structure; the difference is that it has 8 cores instead of 32, and the L3 cache, which is 8MB instead of 32MB, is located in a different power domain. The operating voltage of the main power domain can change from 980mV downwards in X-Gene 2, and from 870mV downwards in X-Gene 3. While all the CPU cores operate at the same voltage, each pair of cores (PMD – Processor MoDule) can operate at different frequency. Frequency ranges from 300MHz to 2.4GHz in X-Gene 2, and from 375MHz to 3GHz in X- Gene 3 (at 1/8 steps of the maximum clock frequency in both microprocessors).

B. *Exposing V_{min} Values in Single-Core Executions*

We experimentally obtain the V_{min} values of 10 SPEC CPU2006 [7] benchmarks on the three X-Gene 2 chips (TTT, TFF, TSS) [8] [9] [10] [11] [12] [13], running the entire time-consuming undervolting experiments multiple times for each benchmark. This part of the study focuses on a quantitative analysis of the V_{min} for diverse microprocessor chips of the same architecture in order to expose the potential guardbands of each chip, as well as to quantify how the program behavior affects the guardband and to measure the core-to-core and chip-to-chip variation. V_{min} is defined as the minimal working voltage of the microprocessor for any workload or operating condition at a specific clock frequency.

For a significant number of benchmarks, we can see variations between different programs and different chips. Fig. 2 presents the most robust core for each chip, and for these programs the V_{min} varies from 885mV to 865mV for TTT, from 885mV to 860mV for TFF, and from 900mV to 870mV for TSS. Considering that the nominal voltage for the X-Gene 2 is 980mV, there is a significant reduction of voltage without affecting the correct execution of programs (single-core runs), which is at least 9.7% for the TTT and TFF, and 8.2% for the TSS. The corresponding power (and corresponding energy) savings are 18.4% for the TTT and TFF chip, and 15.7% for the TSS chip. We also notice that the workload-to-workload variation (~3%) remains the same across the three chips of the same architecture; however, there is significant variation among the chips. This means that there is a program dependency of V_{min} behavior in all chips.

C. *Exposing V_{min} Values in Multi-Core Executions*

Through massive characterization experiments running 25 multi-threaded benchmarks, we obtained the multicore V_{min} values on the two different technology ARMv8-compliant microprocessors: X-Gene 2 and X-Gene 3 (28nm and 16nm, respectively). Fig. 3 shows the V_{min} characterization results for the 25 benchmarks on X-Gene 2 with 8-thread executions of the benchmarks for the three different frequencies: 2.4GHz, 1.2GHz, 0.9GHz, and X-Gene 3 with 32-thread executions for 3GHz and 1.5GHz, respectively [14]. Fig. 3 shows, that for the same number of threads and at the same frequency, the V_{min} for all 25 benchmarks is virtually the same. There are some cases, where a benchmark has a little lower V_{min}, only 10mV or ~1% of the nominal voltage.

To understand this phenomenon, we study the voltage droop magnitude of the microprocessors for all the different frequency and core allocation configurations, by leveraging the embedded oscilloscope in the X-Gene 3 microprocessor. Fig. 4 presents two different ranges of voltage droop magnitude when the microprocessor operates at 3GHz: (a) the [55mV, 65mV) in which we present the configurations of all programs that produce voltage droops more than or equal to 55mV and less than 65mV, and (b) the [45mV, 55mV) in which we present the configurations of all programs that produce voltage droops more than or equal to 45mV and less than 55mV.

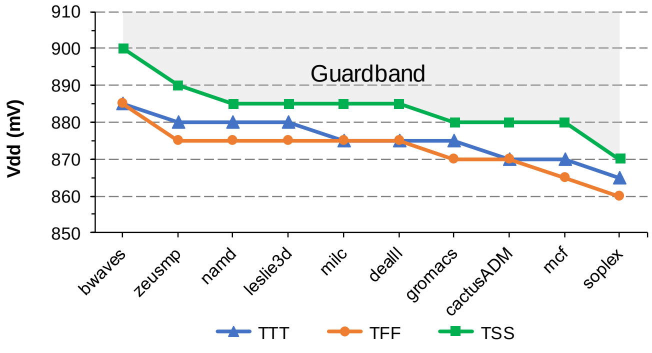


Fig. 2. V_{min} single-core results at 2.4 GHz for 10 SPEC CPU2006 programs on 3 different X-Gene 2 chips (TTT, TFF, TSS).

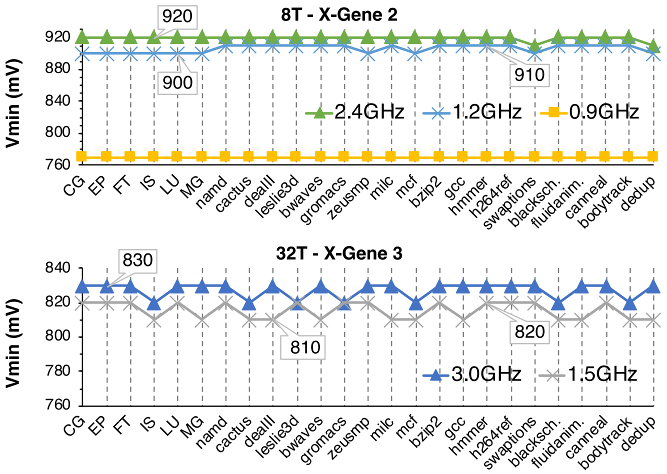


Fig. 3. V_{min} results for multicore runs. The graph presents the X-Gene 2 V_{min} points (top) with 8 threads in 2.4 GHz, 1.2 GHz, and 0.9 GHz clock frequencies, and the X-Gene 3 V_{min} points (bottom) with 32 threads in 3 GHz and 1.5 GHz clock frequencies.

As we can see on the left graph of Fig. 4, the configurations with 32 threads and 16 spreaded threads (one thread is running on each PMD), which means that all 16 PMDs of the microprocessor running at 3GHz frequency produce voltage droop magnitude between 55mV and 65mV. However, the configuration of 16 clustered threads (two threads running on a PMD) has almost zero droops in the range of [55mV, 65mV) for all programs. On the right graph of Fig. 4, the configurations with 16 clustered threads and 8 spreaded threads produce voltage droop magnitude in the range of [45mV, 55mV). Thus, the emergency voltage droops are massive and lead to virtually workload-independent V_{min}.

Although the workload variability marginally affects the V_{min} in multicore executions, core allocation and clock frequency are the major contributors to the V_{min}. The reason is that the frequency and different core allocations are the main factors that can affect the emergency voltage droop magnitude. In particular, the largest amount of voltage reduction (12%) is a result of clock division in a specific clock frequency, while just one step further frequency reduction (due to clock skipping) delivers 3% further voltage reduction. Moreover, assigning the running threads in different cores, we can achieve up to 3% more voltage reduction.

Combining all observations for single-core and multicore characterization, we obtain an optimal scheme of the microprocessors when running real workloads, which can achieve on average 25.2% energy savings on X-Gene 2, and 22.3% energy savings on X-Gene 3, with a minimal performance penalty of 3.2% on X-Gene 2 and 2.5% on X-Gene 3 compared to the default voltage and frequency microprocessor's conditions.

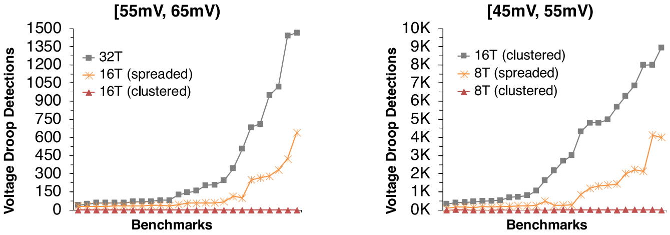


Fig. 4. Voltage droop detections for each program in 2 different voltage droop magnitudes.

III. Many Core GPUs

A. Opportunity for Guardband Optimization

The margin between the nominal voltage and V_{min} reflects the optimization potential. Measurement results in Fig. 5a show that GPUs require as large as 20% voltage guardbands to tolerate worst-case conditions. Fig. 5b shows that measured energy efficiency improvements of reducing the voltage guardband can be as large as 25% [26] [28].

The minimum energy saving is only 8%, which suggests the guardband optimization potential is program-dependent. Voltage guardband is often impacted by different variation types, including process, voltage, and thermal (PVT) variations. The work [26] uses the method of exclusion to rule out other types of variations and identify the voltage noise as the major consumer of voltage guardband and also the root cause of program dependent guardband behavior.

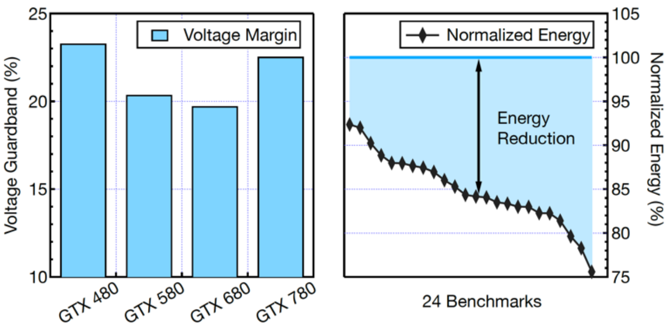
B. Modeling

To understand voltage guardbands in more detail, we need a modeling framework. *GPUVolt* [30] simulates the voltage noise behavior by calculating the time domain response of the power (voltage) delivery model under current input profiles of each core (Fig. 6). We use *GPUWattch* [27], a cycle-level GPU power simulator, to approximate the current variation profile of each GPU core under a certain supply voltage level. *GPUWattch* takes the microarchitectural activity statistics from *GPGPU-Sim* [31], a cycle-level simulator, and calculates the power consumption of each microarchitectural component.

GPUVolt's power delivery model consists of three parts (Fig. 6): the printed circuit board (PCB), the package, and the on-die power delivery network (PDN). We use a lumped model for the PCB and package circuit and a distributed model for the on-die PDN. The distributed model can reflect both intra-SM as well as inter-SM voltage noise interference. We also propose a TPD based PDN scaling methodology since there is no public information on its actual PDN design. The validation results show *GPUVolt* has a Pearson's correlation of 0.9 with hardware based V_{min} measurement.

C. Root Cause

We characterize and analyze the root cause for the large



(a) Voltage guardband.

(b) Energy reduction.

Fig. 5. Opportunity of exploiting the voltage guardband. (a) Measured voltage guardband on four commercial graphics cards. (b) Measured energy reduction on a GTX 480.

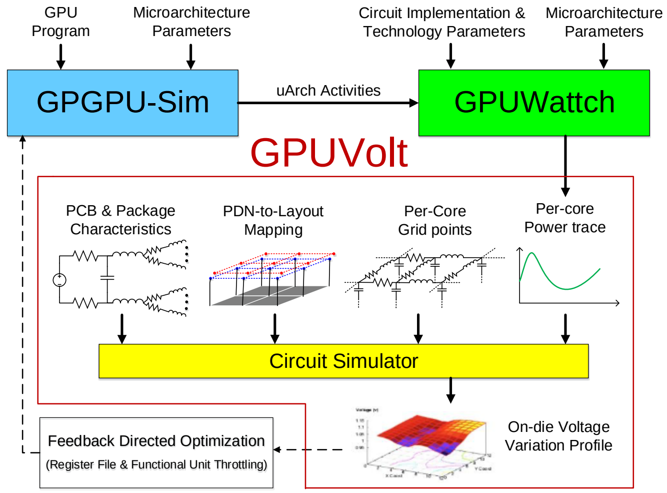

GPU Program
Microarchitecture Parameters
Circuit Implementation & Technology Parameters
Microarchitecture Parameters
GPGPU-Sim
uArch Activities
GPUWattch
GPUVolt
PCB & Package Characteristics
PDN-to-Layout Mapping
Per-Core Grid points
Per-core Power trace
Circuit Simulator
Feedback Directed Optimization
(Register File & Functional Unit Throttling)
On-die Voltage Variation Profile

Fig. 6. Overview of the integrated performance, power, and voltage modelling framework.

voltage droops in the manycore architecture using *GPUVolt*. We perform the analysis at the single-core level to study the impact of individual microarchitectural component and then enlarge the analysis the scope to the entire processor level to study the inter-core voltage interference.

Single-Core: We leverage the linear property of the voltage model to quantify each component's contribution to a single SM's voltage noise. Fig. 7 shows the contribution of the major components. The register file is the single most dominant source of voltage droops, which is closely tied to the GPU's unique characteristics. Modern GPUs require a large register file to hold the architectural states of thousands of concurrent threads (multiple times of the L1 cache size). Consequently, the GPU's RF access rate and power consumption are much higher compared to the CPU [27] [32].

Many Core: To understand the voltage noise characteristics in manycore GPUs, we propose a conceptual framework [29]. In short, it examines voltage noise in the temporal (i.e., time varying) and spatial (i.e., core versus chip-wide) dimensions. Using this framework, we determined that there are two main types of GPU voltage noise: the fast-occurring first-order droops that are localized to a small cluster of neighboring cores, and the slow-occurring chip-wide second-order droops, shown in Fig. 8.

We identify the sensitivity to the activity alignment as the reason why a particular droop type is present/absent. The first-order droop occurs very fast and requires almost perfect alignment for multiple cores to resonate (i.e., global droop). In contrast, the second-order droop occurs much slower and there exists loosely aligned core activity owing to GPU's single-program multiple-data execution model (which we call implicit synchronization) that can cause global droops. These events that can lead to implicit synchronization include I-/D-cache miss and thread block launch.

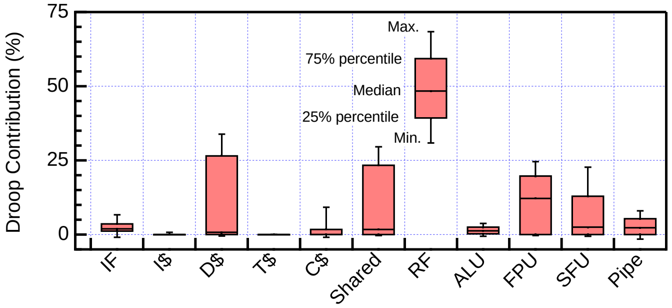


Fig. 7. Component contribution to any voltage droop greater than 3% at the single-SM level.

D. *Smoothing & Optimization*

To smooth GPU voltage noise, we introduce *hierarchical voltage smoothing* [29], where each level specifically targets one type of voltage droop. For the first-order droop, we train a prediction model (off-line trained) to predict the local first-order droop using the root-cause analysis data based on register file and dispatch unit activity. The models provide enough response time for smoothing to work. For the second-order droop, caused by the implicit synchronization, the smoothing mechanism leverages existing hardware communication mechanisms and delays execution to disrupt the current and future synchronization pattern.

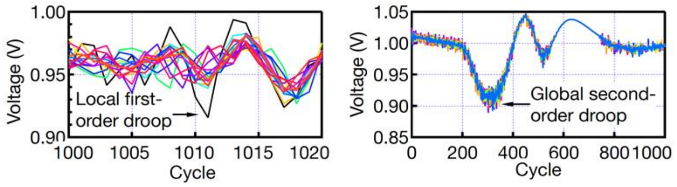
The hierarchical mechanism reduces the worst-case droop by 31%, which enables a smaller voltage guardband for energy-efficiency improvements. We observe an average 7.8% savings. Besides the minimum voltage reduction, we also propose V_{min} prediction based adaptation technique [26] coupled with asymmetric resilience fail-safe mechanism [33] to maximize the energy saving potential (20%).

IV. Supply Voltage Reduction in FPGAs

A. *Experimental Results*

Experiments are performed on representative FPGAs from Xilinx, a main vendor, including VC707 (performance-oriented Virtex), two identical samples of KC705 (A & B, power-oriented Kintex), and ZC702 (CPU-based Zynq). Among various FPGA components, a major part of experiments is initially performed on on-chip memories or Block RAMs (BRAMs), thanks to their importance in the architecture of state-of-the-art applications like FPGA-based DNNs as well as the capability of their voltage rail to be independently regulated. BRAMs are a set of small blocks of SRAMs, distributed over the chip, and in a programmable-fashion can be chained to build larger memories. All evaluated platforms are fabricated with 28nm technology and their nominal/default BRAM's voltage level (VCCBRAM) is 1V.

As shown in Fig. 9, undervolting VCCBRAM below the nominal level, the performance or reliability of the BRAMs are not affected until a certain level, *i.e.*, minimum safe voltage or V_{min}. This region is the GUARDBAND, which is mainly considered by vendors to ensure the worst-case environmental and process scenarios. In the GUARDBAND voltage region, data can be safely retrieved without compromising reliability.



(a) First-order droop.

(b) Second-order droop.

Fig. 8. Two major voltage droop types in GPUs.

Further undervolting, although the FPGA is still accessible, the content of some BRAMs experience faults or bit-flips. We call it as the CRITICAL region. Finally, further undervolting, the DONE pin is unset at V_{crash} and the FPGA does not respond for any request in the CRASH region. As seen, there is a slight difference of mentioned voltage margins among platforms even for identical samples of KC705; however, those three voltage regions are recognizable for all.

As shown in Fig. 9 (for VC707), the power is continuously reduced through undervolting in both GUARDBAND and CRITICAL voltage regions; however, within the CRITICAL region, some of the memories are infected. The fault rate exponentially increases by further undervolting within the CRITICAL region and arrives to 652 faults/Mbit at V_{crash}. In the same line, we observe that the fault rate exponentially increases up to 153, 254, and 60 faults/Mbit at V_{crash} for ZC702, KC705-A, and KC705-B, respectively.

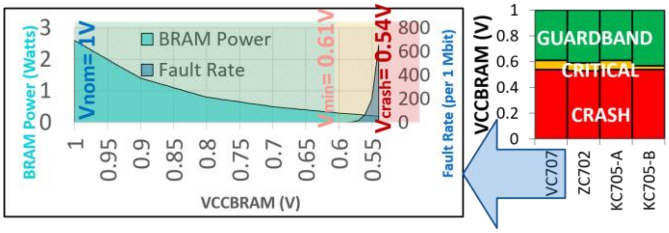


Fig. 9. Voltage behavior and power/reliability trade-off behavior of FPGAs (@ambient temperature).

B. Environmental Temperature

We extend the experimental study to evaluate the impact of the environmental temperature on the reliability of FPGA. Toward this goal, we place the FPGA boards inside a heat chamber where the environmental temperature can be regulated. As can be seen in Fig. 10, with heating up, the fault rate constantly reduces; for instance, by more than 3X for 30◦C higher temperature (for VC707). Also, the changes on the fault rate over the voltage are significantly different among platforms evaluated; for instance, as can be seen, a relatively 156% more fault rate at 50◦C is reduced to 11.6% less fault rate at 80◦C, for VC707 vs. KC705-A. The significant variation on the fault rate of different platforms and the impact of the temperature can be the consequence of the architectural and technological difference, process variation, or aging effects among them.

V. Related Work

During the last years, the goal for improving chips' energy efficiency, while reducing their power supply voltage is a main concern of many scientific studies that investigate the chips' operation limits in nominal and off-nominal conditions. For example, in [35] [36] [37] the authors propose methods to maximize voltage droops in single core and multicore chips in order to investigate their worst-case behavior due to the generated voltage noise effects. In order to eliminate the effects of voltage noise, studies such as [3] and [4] focus on the prediction of critical parts of benchmarks, in which large voltage noise glitches are likely to occur, leading to system malfunctions. In the same context, several studies were presented to mitigate the effects of voltage noise [38] [39] [40] [41] [42] or to recover from them after their occurrence [43].

Apart from these studies that are mainly concentrated on the core and the voltage droops, [5] and [6] focus on the observation of the errors manifested on caches of a commercial Intel Itanium processor during the execution of benchmarks in voltage conditions in off-nominal values. There are also numerous characterization studies of commercial chips in off-nominal voltage conditions for CPUs, GPUs and FPGAs [5] [6] [26] [34] [44] [45] [46].

VI. Conclusion

In this paper, we summarize system-level evaluations of the voltage margins of recent multicore CPUs, manycore GPUs, and FPGAs. We first presented the results on two recent state-of-the-art ARMv8-compliant microprocessors chips. By leveraging the pessimistic voltage guardbands, we can achieve up to 18.4% power reduction in single-core executions and up to 17.6% in multicore executions. For the manycore GPUs case, we report a comprehensive measurement using four NVIDIA GPUs (Fermi and Kepler architectures). We showed that manycore GPUs require as large as 20% voltage guardbands, while the energy efficiency improvements of reducing this guardband can be as large as 25%. We also report experimental evaluation of aggressive undervolting in FPGAs, and showed a power consumption reduction of 90% on average, (on average 39% voltage reduction).

Acknowledgment

The research leading to results in the CPU part was funded by the EU H2020 Programme under the UniServer project (http://www.uniserver2020.eu), grant agreement n◦ 688540. Also, the research leading to results in FPGA part was funded by the EU H2020 Programme under the LEGaTO project (https://legato-project.eu), grant agreement n◦ 780681.

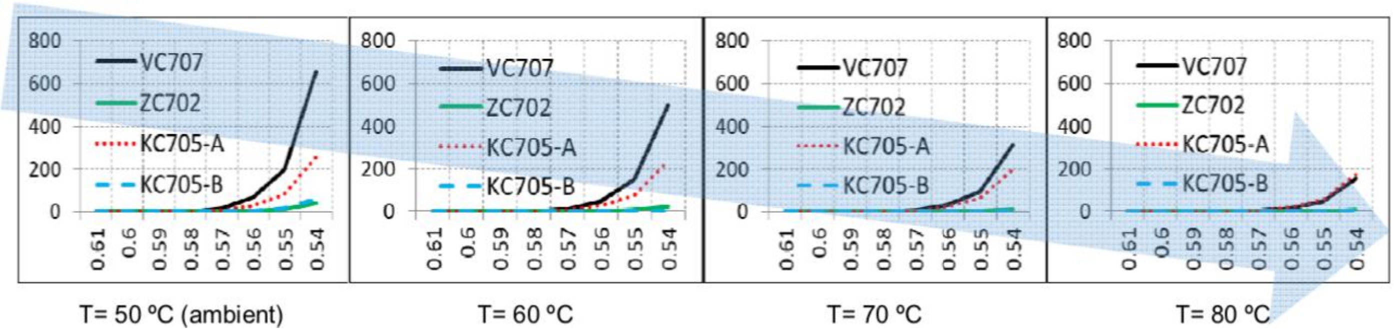


Fig. 10. Reliability of low-voltage BRAMs by experimenting with the environmental temperature changes (x-axis: VCCBRAM, y-axis: fault rate).

REFERENCES

[1] W. Schemmert and G. Zimmer, "Threshold-voltage sensitivity of ion-implanted m.o.s. transistors due to process variations," Electronics Letters, vol. 10, no. 9, p. 151, 1974.

[2] Y. Zu, C. R. Lefurgy, J. Leng, M. Halpern, M. S. Floyd, and V. J. Reddi, "Adaptive guardband scheduling to improve system-level efficiency of the POWER7+," in IEEE/ACM International Symposium on Microarchitecture (MICRO), 2015.

[3] V. J. Reddi, M. S. Gupta, G. Holloway, G.-Y. Wei, M. D. Smith, and D. Brooks, "Voltage emergency prediction: Using signatures to reduce operating margins," in IEEE International Symposium on High-Performance Computer Architecture (HPCA), 2009.

[4] M. S. Gupta, V. J. Reddi, G. Holloway, Gu-Yeon Wei, and D. M. Brooks, "An event-guided approach to reducing voltage noise in processors," in IEEE/ACM Design, Automation & Test in Europe, 2009.

[5] A. Bacha and R. Teodorescu, "Dynamic reduction of voltage margins by leveraging on-chip ECC in Itanium II processors," in ACM/IEEE International Symposium on Computer Architecture (ISCA), 2013.

[6] A. Bacha and R. Teodorescu, "Using ECC Feedback to Guide Voltage Speculation in Low-Voltage Processors," in IEEE/ACM International Symposium on Microarchitecture (MICRO), 2014.

[7] J. L. Henning, "SPEC CPU2006 benchmark descriptions," ACM SIGARCH Computer Arch. News, vol. 34, no. 4, pp. 1–17, Sep. 2006.

[8] G. Papadimitriou, M. Kaliorakis, A. Chatzidimitriou, D. Gizopoulos, P. Lawthers, and S. Das, "Harnessing voltage margins for energy efficiency in multicore CPUs," in Proceedings of IEEE/ACM International Symposium on Microarchitecture (MICRO-50), 2017.

[9] G. Papadimitriou, M. Kaliorakis, A. Chatzidimitriou, D. Gizopoulos, G. Favor, K. Sankaran and S. Das, "A system-level voltage/frequency scaling characterization framework for multicore CPUs," in 13th IEEE Workshop on Silicon Errors in Logic - System Effects (SELSE), 2017.

[10] G. Papadimitriou, A. Chatzidimitriou, M. Kaliorakis, Y. Vastakis, and D. Gizopoulos, "Micro-Viruses for Fast System-Level Voltage Margins Characterization in Multicore CPUs," in IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), 2018.

[11] M. Kaliorakis, A. Chatzidimitriou, G. Papadimitriou, and D. Gizopoulos, "Statistical Analysis of Multicore CPUs Operation in Scaled Voltage Conditions," in IEEE Computer Architecture Letters (IEEE CAL), vol: 17, no: 2, pp. 109-112, July 2018.

[12] K. Tovletoglou, *et al.*, "Measuring and Exploiting Guardbands of Server-Grade ARMv8 CPU Cores and DRAMs", IEEE/IFIP International Conference on Dependable Systems and Networks Workshops, 2018.

[13] G. Karakonstantis, *et al.*, "An Energy-Efficient and Error-Resilient Server Ecosystem Exceeding Conservative Scaling Limits", ACM/IEEE Design, Automation, and Test in Europe (DATE 2018), Dresden, Germany, 2018.

[14] G. Papadimitriou, A. Chatzidimitriou, and D. Gizopoulos, "Adaptive Voltage/Frequency Scaling and Core Allocation for Balanced Energy and Performance on Multicore CPUs," in IEEE International Symposium on High-Performance Computer Architecture, 2019.

[15] M. Feldman, "Good Times for FPGA Enthusiasts.", in Top500, 2016. https://www.top500.org/news/good-times-for-fpga-enthusiasts/

[16] B. Salami, O. Unsal, and A. Cristal, "Comprehensive Evaluation of Supply Voltage Underscaling in FPGA on-Chip Memories," in IEEE/ACM International Symposium on Microarchitecture, 2018.

[17] B. Salami, O. Unsal, and A. Cristal, "Fault Characterization Through FPGA Undervolting", in International Conference on Field Programmable Logic and Applications (FPL), 2018.

[18] B. Salami, O. Unsal, and A. Cristal, "Evaluating Built-In ECC of FPGA On-Chip Memories for the Mitigation of Undervolting Faults", in Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 2019.

[19] B. Salami, O. Unsal, and A. Cristal, "A Demo of FPGA Aggressive Voltage Downscaling: Power and Reliability Tradeoffs", in International Conference on Field Programmable Logic and Applications, 2018.

[20] B. Salami, "Aggressive undervolting of FPGAs: power & reliability trade-offs," Ph.D. Disserattion, Universitat Politècnica de Catalunya (UPC), 2018.

[21] A. Cristal, et. al, "LEGaTO: towards energy-efficient, secure, fault-tolerant toolset for heterogeneous computing" in 15th ACM International Conference on Computing Frontiers (CF), 2018.

[22] A. Cristal, et. al, "LEGaTO: first steps towards energy-efficient toolset for heterogeneous computing", in 8th International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS), 2018.

[23] A. Zou, *et al.*, "Ivory: Early-stage design space exploration tool for integrated voltage regulators," in ACM/EDAC/IEEE Design Automation Conference (DAC), 2017.

[24] A. Zou, *et al.*, "Voltage-stacked gpus: A control theory driven cross-layer solution for practical voltage stacking in gpus," in IEEE/ACM International Symposium on Microarchitecture, Fukuoka, 2018.

[25] A. Zou, *et al.*, "Efficient and reliable power delivery in voltage-stacked manycore system with hybrid charge-recycling regulators," in ACM/EDAC/IEEE Design Automation Conference, (DAC), 2018.

[26] J. Leng, A. Buyuktosunoglu, R. Bertran, P. Bose, and V. J. Reddi, "Safe limits on voltage reduction efficiency in gpus: A direct measurement approach," in IEEE/ACM International Symposium on Microarchitecture (MICRO), 2015.

[27] J. Leng, T. H. Hetherington, A. ElTantawy, S. Z. Gilani, N. S. Kim, T. M. Aamodt, and V. J. Reddi, "GPUWattch: enabling energy optimizations in GPGPUs," in ACM/IEEE International Symposium on Computer Architecture (ISCA), 2013.

[28] J. Leng, Y. Zu, and V. J. Reddi, "Energy Efficiency Benefits of Reducing the Voltage Guardband on the Kepler GPU Architecture," in Workshop on Silicon Errors in Logic - System Effects (SELSE), 2014.

[29] J. Leng, Y. Zu, and V. J. Reddi, "GPU voltage noise: Characterization and hierarchical smoothing of spatial and temporal voltage noise interference in GPU architectures," in IEEE International Symposium on High Performance Computer Architecture (HPCA), 2015.

[30] J. Leng, Y. Zu, M. Rhu, M. S. Gupta, and V. J. Reddi, "GPUVolt: modeling and characterizing voltage noise in GPU architectures," in International Symposium on Low Power Electronics and Design (ISLPED), 2014.

[31] A. Bakhoda, G. L. Yuan, W. W. L. Fung, H. Wong, and T. M. Aamodt, "Analyzing CUDA workloads using a detailed GPU simulator," in IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), 2009.

[32] M. Gebhart, D. R. Johnson, D. Tarjan, S. W. Keckler, W. J. Dally, E. Lindholm, and K. Skadron, "Energy-efficient mechanisms for managing thread context in throughput processors," in ACM/IEEE International Symposium on Computer Architecture (ISCA), 2011.

[33] J. Leng, A. Buyuktosunoglu, R. Bertran, P. Bose, and V. J. Reddi, "Asymmetric Resilience for Accelerator-Rich Systems," Computer Architecture Letters, 2019.

[34] C. R. Lefurgy, A. J. Drake, M. S. Floyd, M. S. Allen-Ware, B. Brock, J.A. Tierno, J. B. Carter, "Active management of timing guardband to save energy in POWER7", in International Symposium on Microarchitecture (MICRO), 2009.

[35] M. Ketkar, and E. Chiprout, "A microarchitecture-based framework for pre- and post-silicon power delivery analysis", in International Symposium on Microarchitecture (MICRO), 2009.

[36] Y. Kim, and L. K. John, "Automated di/dt stressmark generation for microprocessor power delivery networks", in International Symposium on Low Power Electronics and Design (ISPLED), 2011.

[37] Y. Kim, L. K. John, S. Pant, S. Manne, M. Schulte, W. L. Bircher, and M. S. S. Govindan, "AUDIT: Stress Testing the Automatic Way", in International Symposium on Microarchitecture (MICRO), 2012.

[38] V. J. Reddi, S. Kanev, W. Kim, S. Campanoni, M. D. Smith, G.-Y. Wei, and D. Brooks, "Voltage smoothing: Characterizing and mitigating voltage noise in production processors via software-guided thread scheduling", in International Symposium on Microarchitecture (MICRO), 2010.

[39] M. S. Gupta, K. K. Rangan, M. D. Smith, G.-Y. Wei, and D. Brooks, "Towards a software approach to mitigate voltage emergencies", in International Symposium on Low Power Electronics and Design (ISPLED), 2007.

[40] R. Joseph, D. Brooks, and M. Martonosi, "Control techniques to eliminate voltage emergencies in high performance processors", in IEEE International Conference on High-Performance Computer Architecture (HPCA), 2003.

[41] T. N. Miller, R. Thomas, X. Pan, and R. Teodorescu, "VRSync: Characterizing and eliminating synchronization-induced voltage emergencies in many-core processors", in International Symposium on Computer Architecture (ISCA), 2012.

[42] M. D. Powel, and T. N. Vijaykumar, "Pipeline muffling and a priori current ramping: architectural techniques to reduce high-frequency inductive noise", in International Symposium on Low Power Electronics and Design (ISPLED), 2003.

[43] M. S. Gupta, K. K. Rangan, M. D. Smith, G.-Y. Wei, and D. Brooks, "DeCoR: A Delayed Commit and Rollback mechanism for handling inductive noise in processors", in International Conference on High-Performance Computer Architecture (HPCA), 2008.

[44] I. Ahmed, Sh. Zhao, J. Meijers, O. Trescases, and V. Betz, "Automatic BRAM Testing for Robust Dynamic Voltage Scaling for FPGAs", 28th International Conference on Field Programmable Logic and Applications (FPL), 2018.

[45] Linda L. Shen, Ibrahim Ahmed and Vaughn Betz, "Fast Voltage Transients on FPGAs: Impact and Mitigation Strategies", in 27th IEEE International Symposium On Field-Programmable Custom Computing Machines (FCCM), 2019.

[46] José L. Núñez-Yáñez, "Energy Proportional Neural Network Inference with Adaptive Voltage and Frequency Scaling". IEEE Trans. Computers 68(5): 676-687 (2019).