Voltage Margins Identification on Commercial x86-64 Multicore Microprocessors

George Papadimitriou Manolis Kaliorakis Athanasios Chatzidimitriou
Charalampos Magdalinos Dimitris Gizopoulos
Computer Architecture Lab, Dept. of Informatics & Telecommunications, University of Athens, Greece
{georgepap, manoliskal, achatz, c.magdalinos, dgizop}@di.uoa.gr

***Abstract*—In this paper, we explore the pessimistic voltage guardbands of two multicore x86-64 microprocessor chips that belong to different microarchitectures (one ultra-low power and one high-performance microprocessor), when programs are executed on individual cores of the CPU chips. We also examine the energy and temperature gains as positive effects of lowering the voltage in both chips while preserving the functional correctness of programs. The behavior of the cores was examined executing 8 different workloads from the SPEC CPU2006 suite. Our differential experimental study is performed on two state-of-the-art x86-64 microprocessors: an ultra-low power Intel Core i5-4200U and a high-performance Intel Core i7-3970X. Based on the results, the cores on each microprocessor chip behave differently for different workloads when undervolted, and the voltage guardbands are more than 15% below the nominal voltage levels. We show that the energy efficiency can be increased by a maximum of 20% and the reduction of temperature can be up to 25%.**

Keywords—voltage scaling, characterization, energy efficiency, multicore x86-64 CPUs, error detection and correction

I. Introduction

As transistor dimensions get smaller and smaller, the level of process variation among them increases, causing severe differences on operating voltage of the chips, which is typically addressed by the introduction of high (conservative) guardbands. Taking into consideration the growing effects of the Dennard's scaling break, the effects of process variation are of even greater importance since it means that as the transistors are getting smaller the era of dark silicon is getting closer than ever. The trend of adding more cores on the same microprocessor chip also augments the effects of process variation. Therefore, Dennard's scaling failure in conjunction with process variation leads to excessive power consumption to prevent reliability issues in each of the cores. However, the effects of the process variation are different among the cores of the same multicore chips not only due to the abnormalities of the manufacturing process, but also due to the characteristics of the application running on different cores (workload-dependence). This means that, for different programs, it is possible to notice that process variation exhibits different results for the same core. For instance, when running a specific benchmark, a core might exhibit a certain behavior and when trying with another benchmark the core's behavior might differ significantly [1].

The reliable operation of a chip below nominal voltage conditions may be possible due to the process variation. Process variation is the variation in the attributes of transistors that occurs due to the manufacturing process and is very common for microprocessor chips (especially for those that were designed with a lithography process smaller than 65nm). As microprocessor chips are downscaled, the effects of process variation are increased. According to studies concerning a number of current commercial microprocessor chips, process variation is a very common phenomenon among them and the variation may reach up to 20% of power leakage between two cores of the same chip and a 7 celcius degrees difference in temperature [2]. These observations made microprocessor designers change their approach and adapt their designs so as to face this problem. In order to do so, microprocessor designers introduced guardbands on microprocessor chips which set their voltage level higher, alleviating the effects of process variation and ensuring that the processor operates reliably. Since the microprocessor operates on a higher voltage than it is necessary, the power consumption and temperature levels are increased. The reliability of the chip must be preserved at all costs, however, guardbands introduce higher energy consumption than normal since some of the cores are forced to operate in higher voltages than they need due to the process variation.

Bacha et al. [3] [4] have proposed a method used for exploiting the reliable margins of microprocessor cores which achieves on-line voltage speculation by monitoring the ECC error rate of the microprocessor. It is used to dynamically lower the voltage and to find the lowest voltage level where the system works reliably. The result is to lower the power consumption effectively while maintaining the frequency (and thus performance). This can be done with additional hardware monitors that foresee system instability and adjust the frequency accordingly but it can also be done through a software-based solution. It is a different method than the Dynamic Voltage Frequency Scaling (DVFS), which already exists in the x86 microprocessors [5]. DVFS is a method where the clock frequency of the microprocessor chip is reduced, which makes it possible for the voltage supply to be reduced as well, in locked steps. In that way, lower power consumption is achieved without compromising the reliable operation of the microprocessor. Voltage speculation differs from DVFS since (1) it only reduces voltage and not the frequency of the microprocessor chip, and (2) it takes into account both static and dynamic variation.

In this paper, we analyze the reduced voltage margins of the cores of two commercial microprocessor chips (Intel Sandy Bridge-E and Haswell), which can operate reliably and at the same time maintain their frequency at the highest levels. When the cores operate in conservative guardbands, the power consumption is higher than needed, while when the cores operate beyond nominal voltage conditions but with the same frequency, they do not compromise their performance. A great gain of this is the simultaneous decrease of microprocessor chip's energy consumption and temperature, which can prolong its operating life. We performed the experimental evaluation on two state-of-the-art x86-64 microprocessors, (a ultra-low power Intel Core i5-4200U and a high-performance Intel Core i7-3970X) to study: (1) the crash points for each individual core for benchmarks shown in Table III for voltage margins below nominal conditions, (2) the core-to-core variation of the crash points among the cores for the same benchmark, (3) the temperature and power gains due to undervolting, and (4) the amount of manifested cache ECC errors while we reduce the voltage and keep their frequency values unaffected at the highest values.

By lowering the voltage margins of each microprocessor chip we also monitor the ECC errors from any protected hardware unit in the microprocessors. This is done by using the existing protection mechanisms that comply with the Machine Check Architecture (MCA). In that way, not only there is no need for the additional hardware protection mechanism, but a future study might prove it feasible to lower voltage levels at a safe point by simply watching out for increases in errors reported by the already installed mechanisms. This study shows a remarkable up to 20% power consumption reduction by a 15.02% voltage level reduction. This study also focuses on examining the behavior of different microprocessor chips concerning the hardware errors derived from MCA caused by reducing the voltage levels of the microprocessor. By examining the results we can notice that there are significant differences among the cores of the same chip. We present that the safe points of operation vary when running different benchmarks on a specific core. More specifically, while running the same benchmark on each core of a chip we can observe that the error rate (expressed in ECC errors per minute of execution) differs from one core to the other. Finally, the voltage level at which one core becomes unreliable differs when running different benchmarks.

The paper is constructed as follows: in Section II we describe the related work from literature. Section III presents the microarchitectural details for the microprocessors of this study, information about the Machine Check Architecture, the method for ECC error loging and reporting. It also describes the experimental methodology used in this study. Section IV presents the experimental results of our study for both ultra-low power and high-performance microprocessors, concerning the analysis of pessimistic voltage margins, the power consumption, as well as temperature measurements. Finally, Section V concludes the paper.

II. Related Work

During the last years, the improvement of microprocessors' energy efficiency by reducing their supply voltage is a primary concern of many scientific studies. For example, in [6] [7] and [8] 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 [9] and [10] 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 either in the hardware or in the software level were presented to mitigate the effects of voltage noise [11] [12] [13] [14] [15] or to recover from them after their occurrence [16].

Apart from these studies that are mainly concentrated on the core and the voltage droops, [3] and [4] 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. The authors in these studies observe how the ECC rates increase along with the supply voltage reduction. Moreover, the authors in [17] [18] and [19] propose several microarchitectural approaches to ensure the correct operation of caches in ultra-low voltage conditions. Finally, the characterization studies of commercial chips in off-nominal voltage conditions are limited [3] [4] [20] [21] [22].

III. Experimental Setup

A. Microprocessors Description

Table I summarizes the most important architectural and microarchitectural parameters for both ultra-low power Intel Core i5-4200U and high-performance Intel Core i7-3970X microprocessors that are used in our study.

TABLE I. MICROPROCESSORS' CHARACTERISTICS.

Parameters	Configuration	
	i5-4200U (Haswell)	**i7-3970X (Sandy Bridge-E)**
ISA / uArch	x86 / Haswell	x86 / Sandy Bridge-E
Core/Thread Counts	2/4	6/12
Frequency	1.6 GHz (2.6 GHz)	3.5 GHz (4.0 GHz)
L1 Instr. Cache	32 KB	
L1 Data Cache	32 KB	
L2 Cache	256 KB	
L3 Cache	3 MB	15 MB
Technology	22 nm	32 nm
Launch Date	Q3 '13	Q4 '12
Nominal Voltage	0.844 V	1.365 V
Max TDP	15 W	150 W

B. Machine Check Architecture

MCA [23] is the term used to describe Intel's mechanism for detecting, reporting and correcting hardware (machine) errors. These errors can be placed in one of the following three categories: *corrected* hardware errors (CE) and *uncorrected* hardware errors (UE) and *uncorrected recoverable* errors (UCR). The main difference of UCR and UE errors is that after

a UE error the microprocessor cannot be reliably restarted. On the other hand, UCR and CE is that the former can be recovered through the execution of software, while the latter is corrected by the hardware itself. UCR errors can be placed into one of three sub-categories as presented in Table II. This classification depends on whether specific action is required to correct the error. If no action is required then the error is either classified as an *uncorrected no action required* error (UCNA) or as an *uncorrected software recoverable action optional* error (SRAO). If an action is required, then the error is classified as a *software recovery action required* error (SRAR). Errors in the UCNA category are reported as CE by the hardware. This category consists of errors that occur in data that are not consumed and therefore, no action for recovery is required. Errors in the SRAO category also occur in a piece of data that are not consumed by the system. However, if a UCNA error is reported by microprocessor, then specific recovery action can be taken by system software. On the other hand, if the error is not reported, no action is required. The reporting of the errors can be restricted to one logical core or it can be delivered to all the available cores. The MCA is used to report errors such as system bus, ECC, parity, cache, and TLB errors.

TABLE II: MCA ERRORS CLASSIFICATION.

Types of MCA		Description
CE (Corrected Error)		An error corrected by Hardware
UE (Uncorrected Error)		Hardware could not correct an error. Microprocessor context is corrupted and cannot continue to operate a system.
UCR (Uncorrected Recoverable Error)	SRAR (Software Recoverable Action Required)	The error is detected and microprocessor already consumed the data. Shutdown of the system is recommended if the error is not documented. Else specific action needs to be taken by system software.
	SRAO (Software Recoverable Action Optional)	Some data in memory are corrupted but the data have not been consumed and system can perform a recovery action if the error is documented.
	UCNA (Uncorrected No Action Required)	Some data in the system are corrupted but the data have not been consumed and system may continue to operate.

C. Error Logging and Reporting

In this study, we rely on *Windows Hardware Error Architecture* (WHEA) [24] to monitor any hardware event that occurs on lower voltage levels than the nominal. WHEA is introduced since Windows Vista and it is Windows' proposal for taking advantage of MCA providing better, precise and standardized error reports. More specifically, WHEA provides a standard record format for error data and so it is easier to discover the source of the error. Another great feature is that it is now possible to recover from UCR errors that cannot be corrected solely by hardware. Furthermore, it is extensible, which means that as hardware provides more sophisticated error reporting mechanisms, it will be possible for these mechanisms to be accommodated into WHEA and further augment its capabilities.

D. Experimental Methodology

This paper focuses specifically on exploring the voltage guardbands and process variation on two multicore x86-64 microprocessor chips running the Microsoft Windows 10 OS. A subset of the SPEC CPU2006 [25] benchmark suite was used on each microprocessor chip to measure essential metrics, such as Machine Check Architecture (MCA) errors, current voltage level, temperature and power consumption. The purpose of these runs is to examine whether different cores on the same chip exhibit different behaviors when being undervolted depending on the benchmark that runs on it each time. In this way, we examine the process variation of each core of the two chips and we also performed an extensive comparison of the behavior of the two chips based on the undervolting endurance of each core of the two microprocessor chips when executing a specific benchmark.

We use Intel's Extreme Tuning Utility (XTU) [26] for Windows 10 to reduce the voltage in both microprocessors, as well as to estimate the power consumption and temperature. This paper verifies that there is a significant gain in power efficiency by reducing the voltage, while at the same time there is no loss in reliability and performance. By taking into consideration the problems discussed above, one can understand the importance of exploiting the undervolting capabilities introduced by the use of guardbands, which are used to address the problem of process variation. Lower power consumption also means lower temperatures, which are known to prolong the life expectancy of microprocessor chips.

In our experiments, we also consider the number of correctable hardware errors occurring per minute, while executing a benchmark at a lowered voltage; expressed as *error rate*. By examining if the error rate is higher at lower voltage levels, it is possible to understand how stable the system is and whether it is possible to lower the voltage even more before any uncorrectable errors or data corruption occur.

TABLE III. CHARACTERISTICS OF BENCHMARKS USED IN THIS STUDY.

Benchmark	Suite	General Category
bzip2	int	Compression of files
mcf	int	Combinatorial Optimization/Vehicle scheduling in Public mass transportation
milc	FP	Physics/Quantum Chromodynamics
namd	FP	Scientific, Structural Biology, Classical Molecular Dynamics Simulation
hmmer	int	Gene sequence database search
h264ref	int	Video compression
gobmk	int	Artificial Intelligence - game playing
zeusmp	FP	Physics/Magnetohydrodynamics

The nominal voltage for the microprocessors under test is 0.844 V and 1.365 V and the frequency of operation is 2.6 GHz and 4.0 GHz, for the Intel Core i5-4200U (Haswell) and the Intel Core i7-3970X (Sandy Bridge-E), respectively. For each benchmark, we performed three consecutive runs to ensure the stability of results.

IV. Characterization Results

For our differential study between the Sandy Bridge-E and the Haswell microprocessors, we executed 3 consecutive runs for each benchmark of Table III reducing the voltage from the nominal VDD with a step of 5 mV. Our experiments are based on two experimental setups, reducing the voltage while running the benchmarks in each individual core of the two chips. During our experiments, we collected all the results needed to make an extensive comparison between the two chips, concerning:

- the voltage reduction according to the observed crash points below the nominal V_{DD},
- the ECC errors that were manifested from caches,
- the thermal reduction achieved when the chips operate in the lowest off-nominal voltage condition,
- the power gains achieved when the chips operate in the lowest off-nominal voltage condition.

Finally, with our results we can observe the core-to-core variation of executing the same benchmark in each core of the two chips while reducing the voltage up to the lowest off-nominal voltage value. In the following subsections, we illustrate in detail all the aforementioned results of our comprehensive study.

A. Voltage Reduction Evaluation according to the Observed Crash Points

In the first part of our differential study, we compare the two microprocessors (the Sandy Bridge-E and the Haswell) concerning their Crash points running the benchmarks in each individual core of the two chips. With the term Crash point, we define the voltage value in which a microprocessor stops to be functional. All the corrected errors that do not affect the correct operation of the microprocessors appear before the Crash points.

In Fig. 1, we illustrate the percentage of voltage reduction from the nominal voltage for each of the 8 benchmarks of our study that run on each of the 6 cores of the Sandy Bridge-E, while Fig. 2 presents the same percentage of reduction for the 2 cores of the Haswell microprocessor. In general, the Sandy Bridge-E has higher percentage of reduction with a range from 12.09% to 15.02%, while the same percentage for the Haswell ranges from 9.95% to 10.55%. Finally, Fig. 3 illustrates the variation of voltage reduction among the 6 cores of the Sandy Bridge-E. The highest variation is observed in Core 5 (from 12.09% to 15.02% voltage reduction).

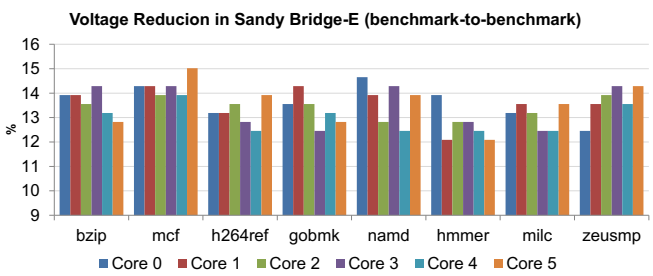

Fig. 1. Voltage reduction percentage in Sandy Bridge-E illustrating benchmark-to-benchmark variation.

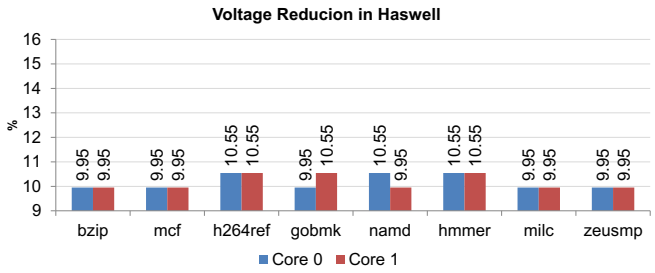

Fig. 2. Voltage reduction percentage in Haswell.

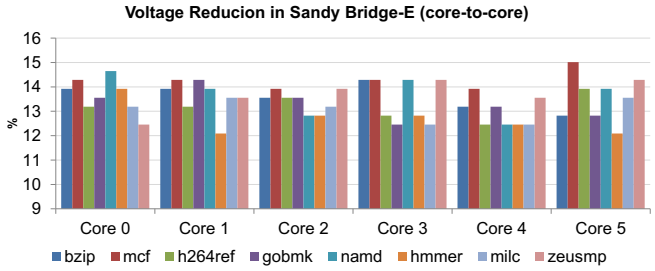

Fig. 3. Core-to-Core variation percentage in Sandy Bridge-E.

B. ECC Errors Evaluation

Next, we present the population of the ECC errors that were observed per each step of voltage reduction (5mV) in the two chips of our differential study. For the Sandy Bridge-E microprocessor no errors were observed before the system crash for all the benchmarks that are used in this study running on the six cores of the processor. On the other hand, Fig. 4 and Fig. 5 represent the population of the ECC errors that were observed in the Core 0 and Core 1 respectively of the Haswell microprocessor.

In Core 0 only one ECC error was manifested during the execution of bzip2 benchmark with a voltage reduction of 85mV from the nominal. On the contrary, Core 1 of the same chip reports more errors for all the executed benchmarks of our study (Fig. 5). We can observe that bzip2 is the most sensitive benchmark (prone to errors) as it starts to report errors after a voltage reduction of 70mV and also reports the highest observed population of errors (11 errors with a voltage reduction of 85mV). In general, from the same figure we observe that there is a smooth increase in the population of

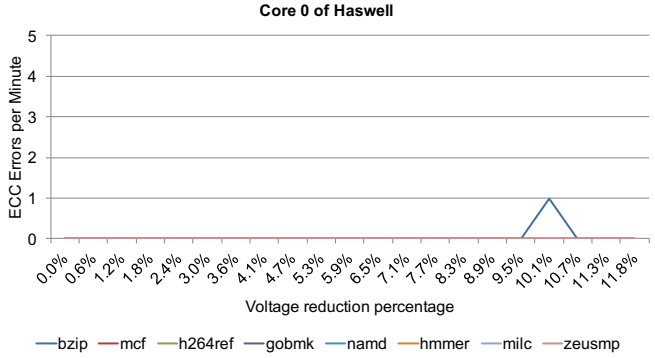

Fig. 4. ECC errors per minute in Core 0 (Haswell).

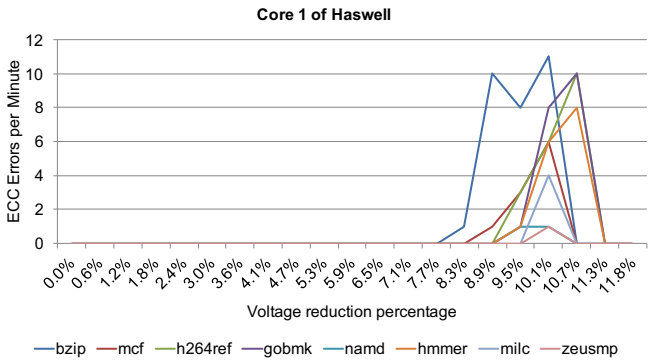

Fig. 5. ECC errors per minute in Core 1 (Haswell).

errors starting from 70mV of voltage reduction and ending (having its peak) at 90mV-95mV of voltage reduction. Similar observations were also reported in [3] and [4]. In all our cases, we observed System Crashes for voltage reductions greater than 90mV.

C. Power Gains Evaluation

To complete the whole picture in our differential study, we illustrate the results of all the cores of the two microprocessors, when they operate in the lowest voltage value above the Crash point. Fig. 6 and Fig. 7 illustrate the power gains of all our experimental configurations for the Sandy Bridge-E and the Haswell microprocessor respectively.

In general, the power gains of the Haswell processor are higher than that of the Sandy Bridge-E. The highest power gains for the Sandy Bridge-E processor were 20% and were observed running the bzip2 and the h264ref benchmarks in Core 0, while the lowest power gains were 12% when we ran the milc benchmark in Core 4. For the Haswell processor, we observed 20% power gains in all experiments except for the zeusmp benchmark for which we measured 17% in both cores.

Table IV summarizes the findings of our differential study between the Sandy Bridge-E and the Haswell microprocessors using 8 benchmarks (bzip2, mcf, namd, milc, hmmer, h264ref, gobmk, zeusmp) with diverse behaviors from the SPEC CPU2006 benchmark suite [25].

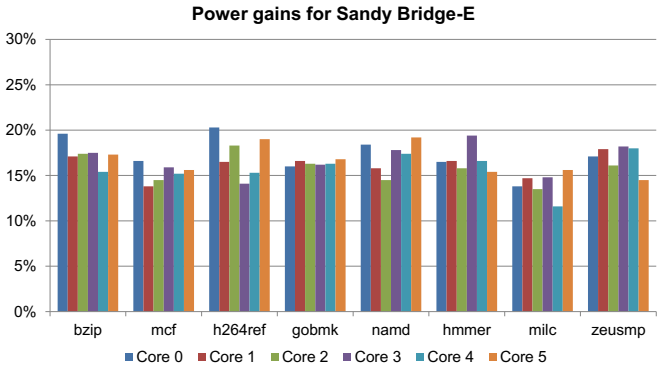

Fig. 6. Power gains percentage (Sandy Bridge-E).

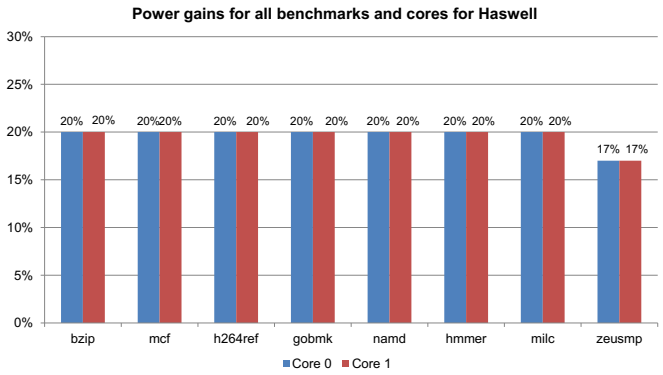

Fig. 7. Power gains percentage (Haswell).

TABLE IV. RESULTS OF ALL THE CORES OF THE TWO INTEL MICROPROCESSORS.

	Intel Core i7-3970X		Intel Core i5-4200U	
	min	***max***	***min***	***max***
crash points below nominal V_{DD}	12.09%	15.02%	9.95%	10.55%
ECC Error Rate	0		0	11
temp. reduction	16%	25%	1%	8%
power gains	12%	20%	17%	20%

D. Thermal Reduction Evaluation

In this subsection, we illustrate the temperature gains for all the cores and all the benchmarks of both microprocessors, when they operate in the lowest voltage value before the Crash point. Fig. 8 and Fig. 9 illustrate the temperature gains of all our configurations for the Sandy Bridge-E and the Haswell microprocessor respectively.

In general, reduction of voltage of the individual cores leads to higher temperature gains in Sandy Bridge-E than the Haswell microprocessor. The highest temperature gains in Sandy Bridge-E were observed running the bzip2 benchmark in Core 3 (25% temperature gains), while the lowest temperature gains for the same processor were observed running the hmmer benchmark in Core 5 (16% temperature gains). For the Haswell, the highest temperature gains were observed running the namd benchmark in Core 0 (8% temperature gains), while the lowest gains (1%) were observed running the mcf benchmark in Core 0 and Core 1 respectively.

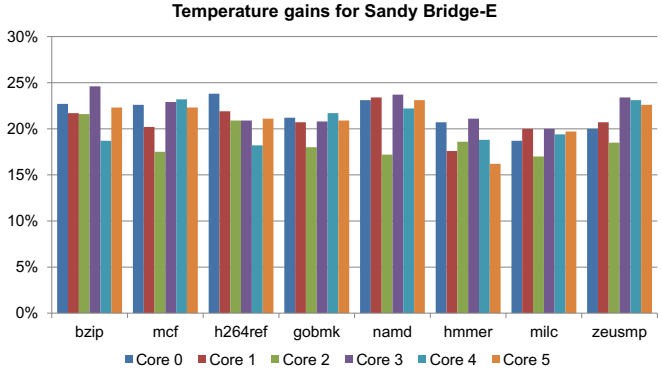

Fig. 8. Temperature gains percentage (Sandy Bridge-E).

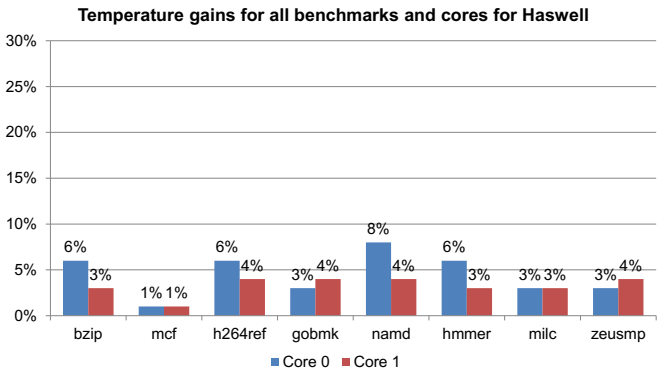

Fig. 9. Temperature gains percentage (Haswell).

V. Conclusion

The purpose of this study is to explore the limits of undervolting on two commercial x86-64 microprocessors chips with significant differences: the first one is an ultra-low power mobile microprocessor (Intel Core i5-4200U) whereas the second one is a high-performance microprocessor (Intel Core i7-3970X). Concerning the limits of undervolting, we observed that on the Intel Core i5-4200U a voltage reduction of 9.95% could be achieved without compromising the reliable execution of the programs we employed. On the other hand, on the Intel Core i7-3970X, depending on the nature of the workload and the core, a further reduction on voltage levels ranging from 12.09% to 15.02% was achieved. These voltage levels reductions translate into 17% to 20% reduction of power consumption on the Intel Core i5-4200U and 12% to 20% for the Intel Core i7-3970X. Regarding the temperatures, the gains on the i5-4200U are low (1% - 8% reduction) when compared to the results of i7-3970X (16% - 25% reduction). Therefore, it is evident that the microprocessor guardbands are too conservative and cause excessive power consumption in order to mitigate the effects of process variation, which in turn translates into higher temperatures.

Acknowledgment

This work is funded by the H2020 Framework Program of the European Union through the UniServer Project (Grant Agreement 688540) – http://www.uniserver2020.eu.

References

[1] B. Balaji, J. McCullough, R. K. Gupta, Y. Agarwal, "Accurate Characterization of the Variability in Power Consumption in Modern Mobile Processors", USENIX Conference on Power-Aware Computing and Systems (HotPower), 2012.

[2] S. Mittal, "A Survey of Architectural Techniques for Managing Process Variation", ACM Computing Surveys (CSUR), Vol.: 48 (4), 2016.

[3] A. Bacha, and R. Teodorescu, "Dynamic reduction of voltage margins by leveraging on-chip ECC in Itanium II processors", in International Symposium on Computer Architecture (ISCA), 2013, pages 297–307.

[4] A. Bacha, and R. Teodorescu, "Using ECC feedback to guide voltage speculation in low-voltage processors", in International Symposium on Microarchitecture (MICRO), 2014, pages 306–318.

[5] E. Le Sueur, and G. Heiser, "Dynamic voltage and frequency scaling: the laws of diminishing returns", in International Conference on Power Aware Computing and Systems (HotPOWER), 2010.

[6] M. Ketkar, and E. Chiprout, "A microarchitecture-based framework for pre- and post-silicon power delivery analysis", in International Symposium on Microarchitecture (MICRO), 2009, pages 179–188.

[7] 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, pages 253-258.

[8] 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, pages 212–223.

[9] M. S. Gupta, V. J. Reddi, G. Holloway, G.-Y. Wai, and D. M. Brooks, "An event-guided approach to reducing voltage noise in processors", in Design, Automation & Test in Europe Conference (DATE), 2009, pages 160-165.

[10] V. J. Reddi, M. S. Gupta, G. H. Holloway, G.-Y. Wei, M. D. Smith, and D. M. Brooks, "Voltage emergency prediction: Using signatures to reduce operating margins", in International Conference on High-Performance Computer Architecture (HPCA), 2009, pages 18–29.

[11] 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, pages 77–88.

[12] 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, pages 123-128.

[13] R. Joseph, D. Brooks, and M. Martonosi, "Control techniques to eliminate voltage emergencies in high performance processors", in International Conference on High-Performance Computer Architecture (HPCA), 2003, pages 79–90.

[14] 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, pages 249–260.

[15] 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, pages 223-228.

[16] 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, pages 381–392.

[17] C. Wilkerson, H. Gao, A. R. Alameldeen, Z. Chishti, M. Khellah, and S.-L. Lu, "Trading off cache capacity for reliability to enable low voltage operation", in *International Symposium on Computer Architecture (ISCA)*, 2008, pages 203–214.

[18] Z. Chishti, A. R. Alameldeen, C. Wilkerson, W. Wu, and S.-L. Lu, "Improving cache lifetime reliability at ultra-low voltages", in International Symposium on Microarchitecture (MICRO), 2009, pages 89–99.

[19] H. Duwe, X. Jian, D. Petrisko, and R. Kumar, "Rescuing uncorrectable fault patterns in on-chip memories through error pattern transformation", in International Symposium on Computer Architecture (ISCA), 2016, pages 634–644.

[20] 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, pages 1–11.

[21] A. Bacha, and R. Teodorescu, "Authenticache: Harnessing cache ECC for system authentication", in International Symposium on Microarchitecture (MICRO), 2015, pages 128–140.

[22] J. Leng, A. Buyuktosunoglu, R. Bertran, P. Bose, V. J. Reddi, "Safe limits on voltage reduction efficiency in GPUs: a direct measurement approach", in International Symposium on Microarchitecture (MICRO), 2015, pages 294–307.

[23] Intel 64 and IA-32 Architectures Software Developer's Manual, Volume 3B. Intel Corp., Order No.: 325462-062US, March 2017, pp. 15–1.

[24] Windows Hardware Error Architecture (WHEA), https://msdn.microsoft.com/en-us/library/windows/hardware/ff559509(v=vs.85).aspx

[25] L. Henning, "SPEC CPU2006 benchmark descriptions," SIGARCH Comput. Archit. September 2006.

[26] Intel® Extreme Tuning Utility (Intel® XTU), https://downloadcenter.intel.com/download/24075/Intel-Extreme-Tuning-Utility-Intel-XTU-.