NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATION

MASTER of SCIENCES GRADUATE PROGRAM

M.Sc. THESIS

Voltage-Margin Advancements among Ultra-low Power Multicore CPU Generations

Nikolaos D. Vazatis

Supervisor: **Dimitris Gizopoulos,** Professor

ATHENS

OCTOBER 2019

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Πρόοδος Περιθωρίων Τάσης Λειτουργίας ανάμεσα σε Γενιές Πολυπύρηνων Επεξεργαστών Εξαιρετικά Χαμηλής Ισχύος

Νικόλαος Δ. Βαζάτης

Επιβλέπων: **Δημήτρης Γκιζόπουλος,** Καθηγητής

ΑΘΗΝΑ

ΟΚΤΩΒΡΙΟΣ 2019

M.Sc. THESIS

Voltage-Margin Advancements among Ultra-low Power Multicore CPU Generations

Nikolaos D. Vazatis
S.N.: M1404

SUPERVISOR: **Dimitris Gizopoulos,** Professor

October 2019

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Πρόοδος Περιθωρίων Τάσης Λειτουργίας ανάμεσα σε Γενιές Πολυπύρηνων Επεξεργαστών Εξαιρετικά Χαμηλής Ισχύος

Νικόλαος Δ. Βαζάτης

Α.Μ.: Μ1404

ΕΠΙΒΛΕΠΩΝ: **Δημήτρης Γκιζόπουλος,** Καθηγητής

Οκτώβριος 2019

ABSTRACT

The main goal of this master thesis is to fully characterize the behavior of Intel's 6th generation Skylake microprocessor, during off-nominal voltage conditions. This characterization is conducted in stages through a CPU undervolting procedure, firstly by gradually reducing the voltage, while maintaining the maximum available frequency (2.3 GHz), and then by reducing the frequency at half (1.2 GHz). The latter, serves the purpose of exposing even lower voltage margins, in which the system operates in normal behavior while sacrificing speed performance. However, in both cases the minimum safe voltage margin is 100 mV below the nominal voltage. Furthermore, we extensively analyze and report any type of error and system crash occurrence.

Afterwards, we present our Skylake characterization results along with those found in a previous study conducted for Intel's 4th generation Haswell microprocessor and we present the voltage-margin advancements between the two ultra-low power CPU generations. During the characterization, we also collected temperature and power measurements. The results are demonstrated in detail through *core-to-core*, *chip-to-chip* and *benchmark-to-benchmark* variations. Our study shows that during the voltage reduction, unsafe operation regions are not formed due to lack of corrected errors occurrences. The maximum voltage reduction that can be achieved is 11.24% with exceptional power consuming gains of up to 41% for specific configurations. However, regarding temperature efficiency there were observed both gains and losses.

SUBJECT AREA: Computers Architecture

KEYWORDS: Energy efficiency, voltage and frequency scaling, power, temperature, error detection and correction, ultra-low power CPUs

ΠΕΡΙΛΗΨΗ

Ο κύριος σκοπός αυτής της μεταπτυχιακής εργασίας είναι ο πλήρης χαρακτηρισμός της συμπεριφοράς της 6ης γενιάς μικροεπεξεργαστή Skylake της Intel, με μια αξιολόγηση 10 δοκιμαστικών προγραμμάτων. Αυτή η αξιολόγηση τελείται μέσω μιας διαδικασίας σταδιακής μείωσης της τάσης του επεξεργαστή, πρώτα διατηρώντας τη μέγιστη συχνότητα στα 2.3 GHz και στη συνέχεια μειώνοντας την συχνότητα στο μισό (1.2 GHz). Η τελευταία περίπτωση μελετήθηκε με σκοπό την αποκάλυψη ακόμα χαμηλότερων ορίων τάσης, στα οποία το σύστημα λειτουργεί με κανονική συμπεριφορά με κόστος την χαμηλότερη απόδοση σε ταχύτητα. Ωστόσο, και στις δύο περιπτώσεις το ελάχιστο ασφαλές περιθώριο τάσης είναι 100 mV κάτω από την ονομαστική. Επιπλέον, αναλύουμε και αναφέρουμε εκτενώς κάθε περίπτωση οποιουδήποτε τύπου σφάλμα και κατάρρευση συστήματος.

Στη συνέχεια, παρουσιάζουμε τα αποτελέσματα από τον χαρακτηρισμό της μηχανής Skylake, μαζί με αυτά που βρέθηκαν σε προηγούμενη μελέτη για τον 4ης γενιάς μικροεπεξεργαστή Haswell της Intel και παρουσιάζουμε την πρόοδο των περιθωρίων τάσης, ανάμεσα στις δύο γενιές επεξεργαστών εξαιρετικά χαμηλής ισχύος. Κατά την διαδικασία χαρακτηρισμού, συλλέξαμε επίσης μετρήσεις θερμοκρασίας και ισχύος. Τα αποτελέσματα παρουσιάζονται αναλυτικά μέσω μεταβολών από *πυρήνα-σε-πυρήνα*, *κύκλωμα-σε-κύκλωμα* και *δοκιμασία-σε-δοκιμασία*. Η μελέτη μας, δείχνει ότι κατά τη διάρκεια μείωσης της τάσης δεν σχηματίζονται περιοχές μη ασφαλούς λειτουργίας λόγω μη εμφάνισης διορθωμένων λαθών. Η μέγιστη μείωση τάσης που μπορεί να επιτευχθεί είναι 11.24% με εξαιρετικά κέρδη στην κατανάλωση ισχύος μέχρι και 41% για συγκεκριμένες παραμετροποιήσεις. Ωστόσο, σχετικά με την απόδοση της θερμοκρασίας παρατηρήθηκαν τόσο κέρδη όσο και απώλειες.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Αρχιτεκτονική Υπολογιστών

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Απόδοση ενέργειας, κλιμάκωση τάσης και συχνότητας, ισχύς, θερμοκρασία, αναγνώριση και διόρθωση σφαλμάτων, επεξεργαστές εξαιρετικά χαμηλής ισχύος

For my parents,
Dimitrios N. Vazatis and Aspasia K. Vazati

To the memory of my godfather,
Evangelos Z. Zikos (1939-2019)

ACKNOWLEDGEMENTS

I would first like to thank my thesis advisor Professor Dimitris Gizopoulos of Informatics and Telecommunications department at the National and Kapodistrian University of Athens. I am grateful for the opportunity he gave me to explore such an interesting and emerging computer engineering research topic as low-power computing.

I would also like to thank the PhD candidates George Papadimitriou and Athanasios Chatzidimitriou who were involved in the validation of this research thesis. Their passionate participation and vast expertise knowledge were valuable in the occasion of any problem might had arisen during the conduction of this thesis.

CONTENTS

LIST OF FIGURES

LIST OF IMAGES

LIST OF TABLES

PREFACE

This thesis was conducted as part of the Master of Sciences graduate program for the Informatics and Telecommunications department at the National and Kapodistrian University of Athens.

1. INTRODUCTION

We live in an era of ever-increasing demands in powerful computing devices. This is achieved by increasing the number of chip transistors with an increasing pace following Moore's law. As everything in real life, while technology advances, new and more complex computational problems arise that need to be solved. Yet the power resources to overcome these improvement challenges are neither efficiently spent nor are infinite. On one extreme, embedded devices with a variety of applications in digital consumer market and automotive, medical or aerospace industries, need to extend battery duration. On the other extreme, large cloud computing centers are already such power hungry, that cooling requirements costs need to be considered and optimally minimized. Green computing, is an emerging area, which handles these issues. Highly sophisticated and advanced low power design techniques, are continuously developed and applied in this direction. Projects such as Aurora exascale supercomputer, it was initially estimated that would require around 200 MW of electric power, but now it is expected to consume just 45-60 MW [1]. An overview of the most important categories in power management techniques is shown in Image 1, while a detailed demonstration of selected techniques at the architectural level can be found in [2].

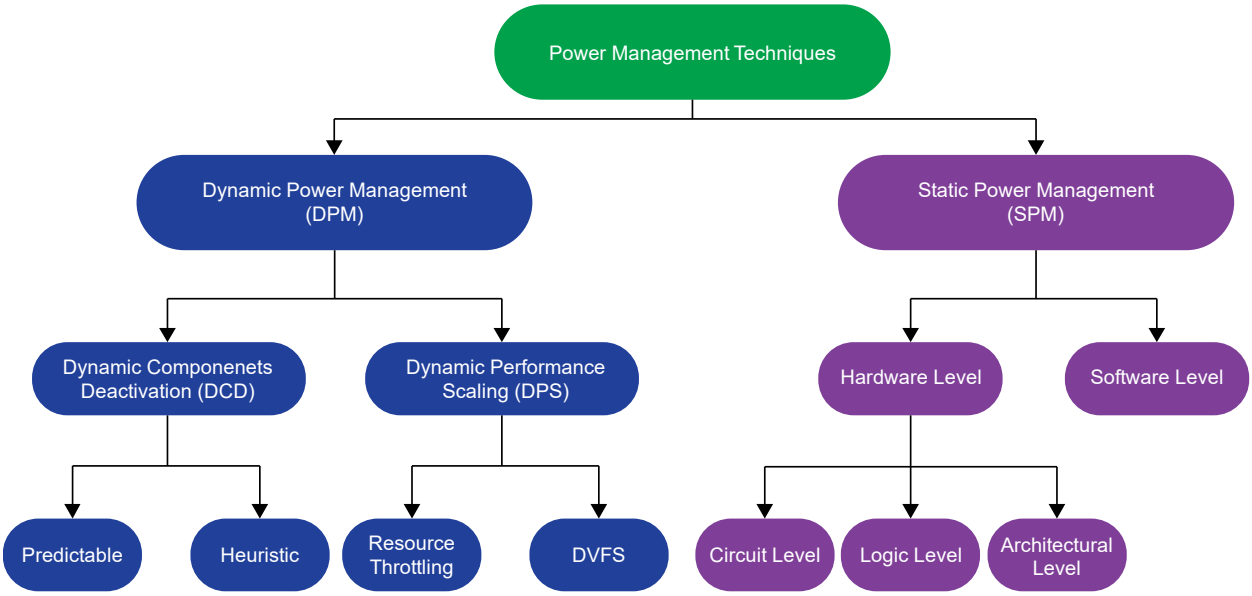


Image 1: High-level power management techniques [3]

For power efficiency, we need to operate the microprocessor at lower voltages. Unfortunately, the pessimistic voltage margins applied by the microprocessor vendors, are not optimized for maximum energy savings due to technology limitations, such as process variability during microprocessor manufacturing and circuit aging [4]. Nevertheless, there are limits on how low these voltages can be, because if they get exceeded, errors occur in the system, compromising its reliability. In this case, the potential for an unexpected fatal crash also increases.

One of the techniques that scales the operational voltages for power efficiency is Dynamic Voltage Frequency Scaling (DVFS). It corelates directly with the dynamic power consumed by a microprocessor, which is given by the equation, $P = aCV^2f$, where a is a constant representing CPU's activity factor, C is the load capacitance, which is fixed for a specific microprocessor, V is the supply voltage and f is the operating frequency. A variant of DVFS is Dynamic Voltage Scaling (DVS), which scales the voltage, while

frequency remains intact. In such case, if we manage to reduce voltage, then power will dissipate quadratically.

However, there are works such as [5], [6] that predict the diminishing gains from the application of DVFS technique, as the process technology progresses through the upcoming years. While in the last years, leakage power is indeed the primary design concern and clock gating remains the most popular power reduction technique, as of 2016 DVFS still held a great share among other power reduction methods (ANNEX, Image 13[1], p. 99). Compared to the previous survey of 2013, it even showed an increase in usage.

In general, there are studies where DVFS and its variant DVS have been applied to integrated circuits, such as FPGAs in [7] and [8] respectively. Similar undervolting studies have been done for GPUs [9], [10], for network-on-chips (NoCs) [11] and for data centers [12]. Nevertheless, along with reduced microprocessor voltages which translates to less power consumption, we must fully characterize the microprocessor to ensure a reliable operation without errors. Past works such as [13] and [14], have conducted such studies for the characterization of multicore CPUs. During the characterization they gradually lowered the voltage and they reported 20% and 19.4% in energy savings respectively. Regarding FPGAs, the first study on fault characterization has been conducted in [15].

This thesis, focuses on the characterization of the Intel Skylake i5-6200U microprocessor, in order to understand how and at what magnitude the safe voltage-margins variability is affected, in new ultra-low power CPU generations. We present our results along with the Intel Haswell i5-4200U microprocessor, which has been characterized in a previous work [16]. Specifically, for ultra-low power processors there are other studies too, such as [17], which however demonstrates power reduction techniques at the circuit level.

During the years, microprocessor vendors have designed and synthesized highly sophisticated technologies, that compensate the power consumption for the increased performance needs, such as Enhanced Intel SpeedStep Technology (EIST), Intel Turbo Boost Technology 2.0 and Intel Speed Shift Technology (SST). To the best of our knowledge, there is no other characterization study previously undertaken, on an SST enabled system with EIST and Turbo Boost disabled. There is a non-characterization study however [18], that discusses the validity of the Running Average Power Limit (RAPL) register counters, in the case where EIST is disabled.

The rest of this thesis is organized as follows. In Chapter 2, we present the underlying architectures that the microprocessor and the software use to report and resolve various types of errors. We also demonstrate, the main microarchitecture differences between the Skylake and Haswell processors, along with details of Intel technologies that the study relies on. In Chapter 3, we present the experimental framework setup and the methodology we used to conduct our study. In Chapter 4, we initially compare the results we acquired during the study of i5-6200U at full and half clock frequencies and then we present them along with the characterization results obtained from i5-4200U study. Finally, this thesis is concluded with Chapter 5.

[1] Bar ranges without values have been enlarged for visual purposes (their actual values are close or a lot lower than 10%).

2. BACKGROUND AND LITERATURE REVIEW

2.1 Machine Check Architecture (MCA)

Machine Check Architecture [19], is a system's mechanism developed by Intel, in order operating system to get informed of hardware (machine) errors. Such errors, are system bus errors, error-correcting code (ECC) errors, parity errors, cache errors and translation lookaside buffer (TLB) errors. Their detection and report are done through machine-check registers, which consist of a set of control, status and error-reporting model-specific registers (MSRs) shown in Table 1. Global control MSRs are used to set up machine checking, while Error-Reporting Bank Registers are used for recording detected errors. However, MCA cannot detect errors such as Silent Data Corruption (SDC) errors.

Table 1: Machine-check MSRs (64-bit)

Global Control MSRs	Description
IA32_MCG_CAP	Read-only register. Contains feature bits with the banks number of error reporting MSRs
IA32_MCG_STATUS	Describes current processor's state after a machine-check exception (#MC)
IA32_MCG_CTL	On its existence, it controls the reporting of machine-check exceptions
IA32_MCG_EXT_CTL	On its existence, allows the processor to signal some #MC to only a single logical processor
Error-Reporting Bank Registers	**Description**
IA32_MC*i*_CTL	Controls signaling of #MC for errors produced by a specific hardware unit
IA32_MC*i*_STATUS	Contains information related to a machine-check error if its VAL flag (bit 63) is set
IA32_MC*i*_ADDR	Contains the address of the code or data memory location that produced the machine-check error if the ADDRV flag (bits 35:0) in the IA32_MCi_STATUS register is set
IA32_MC*i*_MISC	Contains additional information describing the machine-check error if the MISCV flag (bit 59) in the IA32_MCi_STATUS register is set
IA32_MC*i*_CTL2	On its existence, provides the programming interface to use corrected MC error signaling capability

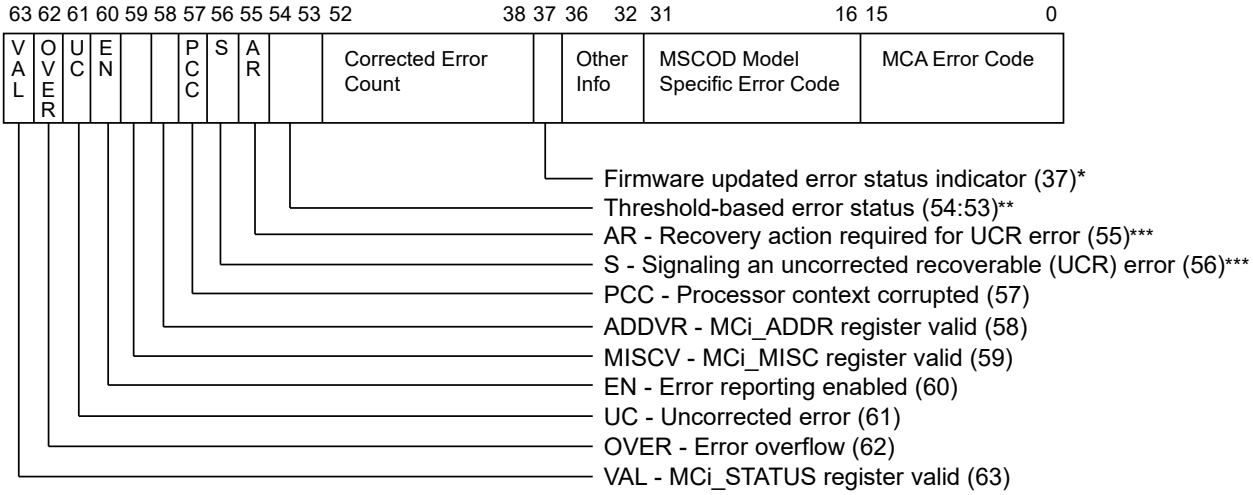
In the presence of MCA, the microprocessor uses a signaling mechanism in the case an uncorrected machine-check error is detected, that generates an abort class exception, which is called machine-check exception (#MC). After that machine-check exception, the microprocessor isn't usually allowed to be restarted reliably.

Regarding cache error reporting, in the past the determining factor of cache status was the number of correction events that occurred in a cache. Starting with Intel Core Duo processors, a new mechanism called "threshold-based error status" was introduced. In

this mechanism the cache status is determined by the number of cache lines (ECC blocks) that inflict repeated corrections. A green status, indicates that the cache lines that inflict repeated corrections do not exceed the pre-defined by Intel threshold. A yellow status, indicates that this threshold was exceeded and the issue must be addressed. This cache status shouldn't be considered as critical as an uncorrected error but rather as a warning for an upcoming uncorrected error, in the case the threshold is reached. However, an uncorrected error can happen even without the appearance of a yellow status.

Machine-check architecture received an architectural enhancement with the 45 nm process technology. When corrected machine-check errors occur, the microprocessor can report information about them and send an interrupt, known as corrected machine-check error interrupt (CMCI). Prior to CMCI, the threshold-based error reporting, allowed the software to request the status of hardware corrected MC errors only by periodic polling of the registers' banks.

Machine-check architecture and CMCI aware processors may have the ability of software recovery from specific uncorrected recoverable (UCR) machine check errors. That allows the continuation of the execution process. UCR errors are detected and signaled by the microprocessor uncorrected errors, which have not corrupted the microprocessor functionality. The MSR used for reporting UCR errors and existing corrected or uncorrected errors is IA32_MCi_STATUS (Image 2).



* When IA32_MCG_CAP[25] (MCG_EMC_P) is set, bit 37 is not part of "Other Information".

** When IA32_MCG_CAP[11] (MCG_TES_P) is not set, these bits are model-specific (part of "Other Information").

*** When IA32_MCG_CAP[11] or IA32_MCG_CAP[24] are not set, these bits are reserved, or model-specific (part of "Other Information").

Image 2: IA32_MCi_Status MSR

An overview of the errors detectable by MCA, is shown on the next page in Table 2.

Table 2: Classification of errors handled by MCA

Type of Errors		Description	Action
Corrected Errors (CE)		Errors detected and corrected by hardware	Execution is unaffected
Uncorrected Errors (UE)		Errors detected but not corrected by hardware	Execution cannot continue
Uncorrected Recoverable Errors (UCR)	Uncorrected no action required (UCNA)	Valid processor state. Some data are corrupted but not consumed	Execution may continue
	Software recoverable action optional (SRAO)	Valid processor state. Some data are corrupted, but not consumed	System software may issue a recovery action
	Software recoverable action required (SRAR)	Conditional microprocessor state. Some data are corrupted and consumed	If MISCV and ADDRV flags in IA32_MCi_STATUS MSR are not set, system shutdown is recommended. Else system software may issue a recovery action

When one of the above errors is detected, the microprocessor writes a 16-bit error code to the MCA error code field of one of the IA32_MCi_STATUS registers. There are two types of MCA error codes, compound and simple error codes. Compound error codes describe errors related to the TLBs, memory, caches, bus and interconnect logic, and internal timer. A brief description of them can be seen in Table 3 and Table 4.

Table 3: Compound error code encoding of IA32_MCi_Status [15:0]

Type	Form
Generic Cache Hierarchy	000F 0000 0000 11LL
TLB Errors	000F 0000 0001 TTLL
Memory Controller Errors	000F 0000 1MMM CCCC
Cache Hierarchy Errors	000F 0001 RRRR TTLL
Extended Memory Errors	000F 0010 1MMM CCCC
Bus and Interconnect Errors	000F 1PPT RRRR IILL

F - Correction report filtering
LL - Memory hierarchy level
TT - Transaction type
MMM and CCCC - Memory transaction type and channel
RRRR - Type of action associated with the error
PP, T and II - Participation, Timeout and Memory I/O

} Encoding bits

Table 4: Simple error code encoding of IA32_MCi_Status [15:0]

Error Code	Binary Encoding	Meaning
No Error	0000 0000 0000 0000	No error has been reported to this bank of error-reporting registers
Unclassified	0000 0000 0000 0001	This error has not been classified into the MCA error classes
Microcode ROM Parity Error	0000 0000 0000 0010	Parity error in internal microcode ROM
External Error	0000 0000 0000 0011	The BINIT# from another processor caused this processor to enter machine check
FRC Error	0000 0000 0000 0100	FRC (functional redundancy check) master/slave error
Internal Parity Error	0000 0000 0000 0101	Internal parity error
SMM Handler Code Access Violation	0000 0000 0000 0110	An attempt was made by the SMM Handler to execute outside the ranges specified by SMRR
Internal Timer Error	0000 0100 0000 0000	Internal timer error
I/O Error	0000 1110 0000 1011	Generic I/O error
Internal Unclassified	0000 01xx xxxx xxxx	Internal unclassified errors

2.2 Windows Error Hardware Architecture (WHEA)

WHEA is a machine check error handler for Windows OS, introduced in Vista version (2006). It is synthesized by a stack of components shown in Image 3.

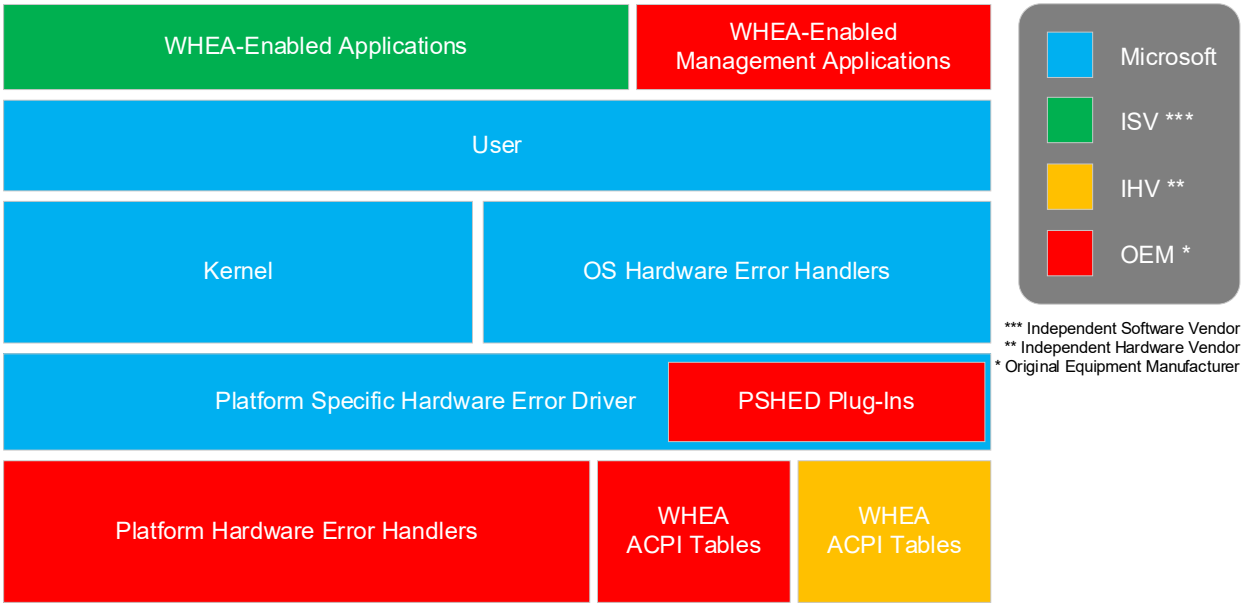


Image 3: WHEA components [20]

A core element in WHEA is the hardware error source, which is basically any hardware unit with the ability to inform the operating system about the existence of errors. One such unit is #MC that we discussed about, earlier in section 2.1. For this reason, a list of all

platform-specific hardware error sources is maintained by the operating system. Upon operating system's start, WHEA identifies the existing hardware error sources in the platform. Newer than Windows Vista versions, have access to such records through Advanced Configuration and Power Interface (ACPI) tables, shown in Table 5, firmware interactions, and other platform-specific mechanisms.

Table 5: ACPI tables used by WHEA

Name	Description
Error Record Serialization Table (ERST)	During boot the OS gets information so it can interact with the platform's error record serialization hardware
BOOT Error Record Table (BERT)	During boot the firmware notifies the OS that the system either crashed or was shutdown unexpectedly
Hardware Error Source Table (HEST)	The firmware gives to the OS the appropriate information for system's hardware error sources
Error Injection Table (EINJ)	Generic interface mechanism through which the OS can inject hardware errors to the platform

For each hardware error source, a low-level hardware error handler (LLHEH) is assigned. LLHEH is the first code that executes after a hardware error occurs and acts as an operating system agent by acknowledging the presence of a hardware error and collecting any related information to this. The data are then formed to packets that are sent to the kernel. Both kernel and LLHEH layers interface with platform-specific hardware error driver (PSHED) layer where platform-specific information resides. Depending the platform, this layer's default capabilities can be further enhanced by vendor made PSHED plug-ins.

When a hardware error occurs, WHEA issues the creation of an error record that describes its nature and an Event Tracing for Windows (ETW) hardware error event is raised. Then PSHED or PSHED plug-ins in their presence, add further details to the error record. Finally, the kernel stores the error record in system's event log. All error records follow a WHEA_ERROR_RECORD format, which is compliant with the Unified Extensible Firmware Interface (UEFI) as shown on the next page in Image 4.

In the case, the hardware error is an uncorrected hardware error (either fatal or nonfatal), then a WHEA_UNCORRECTABLE_ERROR bug check with a value of 0x00000124 is raised, that uses the data provided by WHEA. This bug check has four parameters. Parameter 1, refers to the type of error source that reported the error. In Parameter 2, the address of the WHEA_ERROR_RECORD is stored that provides information about the error. The rest parameters, are reserved for other purposes.

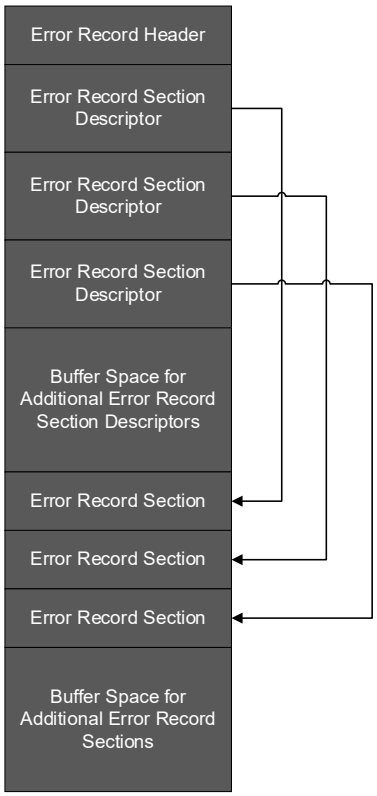


Image 4: Format of error records used by WHEA [21]

2.3 Microarchitecture of Haswell and Skylake Microprocessors

The Haswell (2013) and Skylake (2015) Intel processors which are under the scope of this thesis, have both entered the market at the Tock beat of Intel's Tick-Tock model. This means that the processors are using the same manufacturing process technology as their Tick predecessors (22 nm and 14 nm respectively). Concerning the microarchitecture, Intel designed Haswell based on Ivy Bridge (2011) and Skylake based on Broadwell (2014), receiving numerous enhancements and innovations. To keep the context compact, herewith we present only the most important differences between Skylake and Haswell and therefore we do not explain any microarchitecture details such as pipeline functionality.

2.3.1 Haswell

In Haswell, a great emphasis was given on power-performance with the deployment of newly introduced techniques. Firstly, the idle power was improved by 20%. This was achieved by extending C-states (more on this on section 2.4), with the deeper low power C6 and C7 sleep states. Also, the introduction of the new active idle-power state S0ix, offers significant improvements concerning battery life.

However, one of the most revolutionary features presented in Intel's 4th generation microprocessor, is the Fully Integrated Voltage Regulator (FIVR), which got much attention. Prior implementations, had issues that prevented the broad development of FIVR. Its purpose, was to reduce motherboard's voltage regulator complexity and provide more efficient power to the microprocessor. Previously, there were five separate input voltages that powered on the microprocessor, namely V_{CORE}, V_{GPU}, V_{CCIO}, V_{CCSA} and V_{CCPLL}, which were reduced to one, as shown in Image 5 on the next page (the System

Agent is the Memory Controller Hub formerly known as Northbridge, while PLL stands for Phase Locked Loop).

Fully Integrated Voltage Regulator (FIVR)

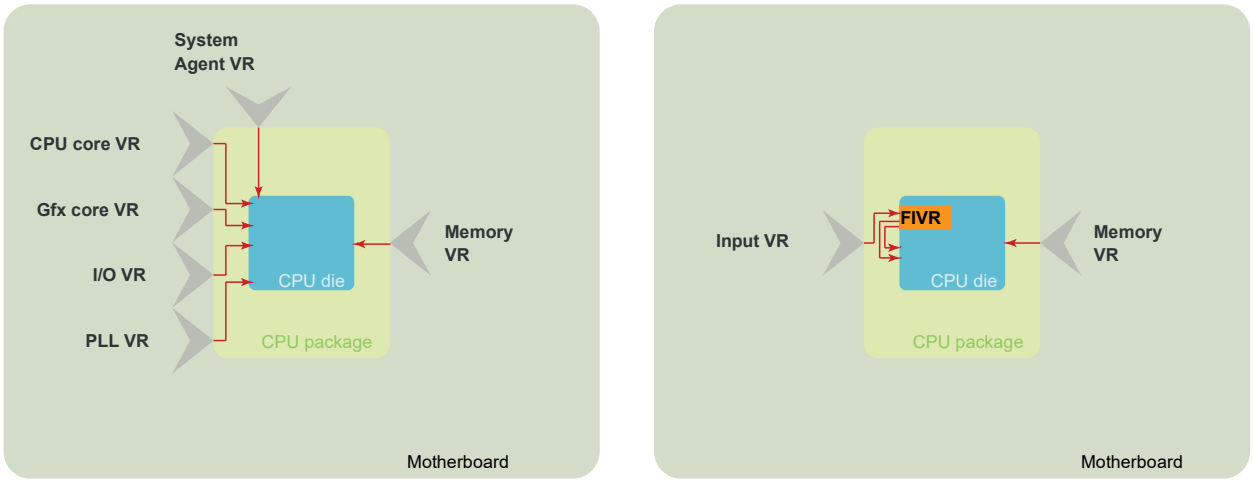


Image 5: Fully Integrated Voltage Regulator (FIVR) [22]

FIVR are synchronous 140 MHz multi-phase (up to 16 phases) buck regulators, which are housed into the die itself [23]. They have features such as up to 80 MHz unity gain bandwidth, non-magnetic package trace inductors and on-die MIM capacitors. There are two stages in Haswell's powering scheme. At first stage, a motherboard voltage regulator feeds the die with approximately 1.8 V, converted initially from a 12-20 V supplied by either the power supply unit or battery. The second conversion stage, is composed of between 8 and 31 FIVRs depending the product. This scheme offers high configurability in terms of I/O voltage, that minimizes the overall power consumption of the die.

The benefits from this design alteration are numerous, including 50% or more battery life improvements for mobile products, increased available power (translates to overall best performance increase), decreased power required for a given level of performance and decreased platform cost and size. However, FIVR is sensitive to the layout of the die and the package, which includes the inductors, due to the high switching frequency used. Thus, every die/package combination is individually tested and optimized.

Nevertheless, for unknown reasons[2] Intel decided to drop the FIVR design from Skylake onwards and return the voltage regulation back to the motherboard. For further details on Haswell, someone can refer to [24].

2.3.2 Skylake

Intel's 4th generation microprocessor, comes with architectural changes in cache memory and several power-optimization enhancements. Concerning the memory subsystem, the design modification occurred on the embedded Dynamic Random-Access Memory (eDRAM), is of high importance. In previous generations such as Broadwell, eDRAM was used as an L4 cache. Since Skylake, eDRAM exists as a memory-side cache (Image 6, p. 36). This offers the capability, of being visible for data requests by every device or

[2] Unofficial speculations talk about added costs in microprocessors' manufacturing procedure

microprocessor core. In the case of a miss, the requested address value is allocated in the eDRAM but in the case of a hit the requester will receive the available cached data. This change also makes eDRAM not architectural, namely it can cache any data including those that are not cached in memory while there is no need to be flushed for coherency maintenance purposes.

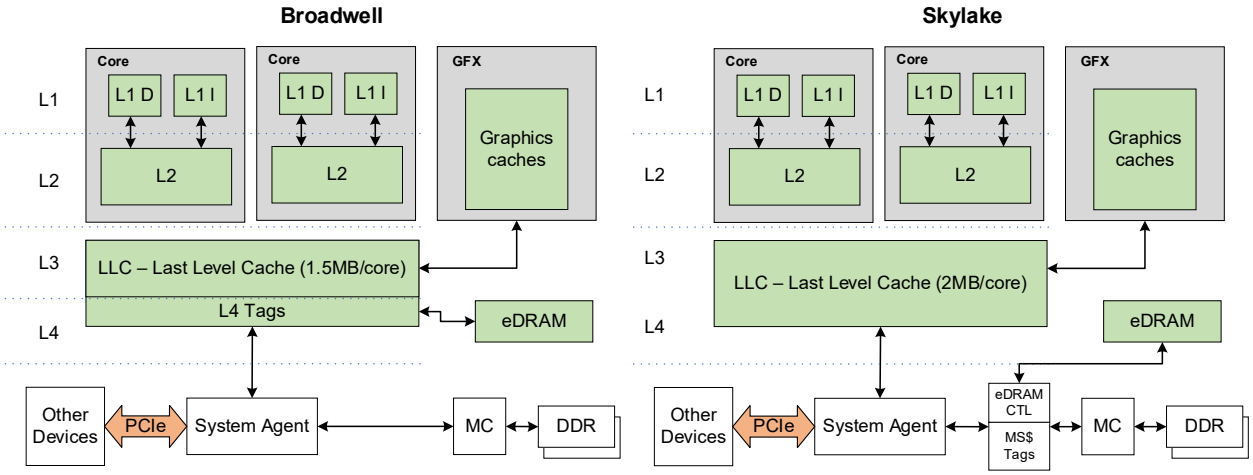


Image 6: Broadwell vs Skylake eDRAM architecture [25]

In the power optimizations compartment, the new Intel Speed Shift Technology, which introduces the concept of hardware-controlled performance states (P-states), is a distinguished addition. Therefore, it will be analytical discussed in section 2.6. The legacy Enhanced Intel SpeedStep Technology (discussed in section 2.4), also obtained an expansion in domains such as System Agent, DDR and eDRAM. This enhances the power efficiency depending bandwidth usage by raising the frequency of CPU and or GPU. Other power optimizations include the power gating of Intel AVX2 (Advanced Vector Extension 2.0) hardware, which turns off when it is not used to eliminate power leakage. Also, underused resources are downscaled and improved power profiles depending the application were introduced including, idle power reduction and C1 state power reduction through improved dynamic capacitance C_{dyn}. These power management innovations provide an overall better Performance/Watt for the core operation.

Finally, further details regarding Skylake can be found in [26].

2.3.3 Summary

In Table 6 are summarized, the most important architectural and microarchitectural advancements of Skylake compared to Haswell. Also, in Image 14 and Image 15 (ANNEX, pp. 100-101) are depicted the front end, the execution engine and the memory subsystem of a Haswell and Skylake single core, respectively. On both images the key differences are marked too.

Table 6: Haswell and Skylake generations in numbers

	Parameter	Haswell		Skylake	
Core	Out-of-order Window	192		224	
	In-flight Loads	72		72	
	In-flight Stores	42		56	
	Scheduler Entries	60		97	
	Integer Register File	168		180	
	FP Register file	168		168	
	Allocation Queue	56		64/thread	
Caches	L0 µOP Cache	1536 µOPs 8-way *		1536 µOPs 8-way	
	L1 Instruction Cache	32KB, 8-way		32KB, 8-way	
	L1 Data Cache	32KB, 8-way	*Fastest load-to-use* 4 cycles *Load bandwidth* 64 Bytes/cycle *Store bandwidth* 32 Bytes/cycle	32KB, 8-way	*Fastest load-to-use* 4 cycles ** *Load bandwidth* 64 Bytes/cycle *Store bandwidth* 32 Bytes/cycle
	L2 Unified Cache	256KB, 8-way	*Fastest load-to-use* 11 cycles *Bandwidth to L1* 64 Bytes/cycle	256KB, 4-way	*Fastest load-to-use* 12 cycles *Bandwidth to L1* 64 Bytes/cycle
	L3 Cache	1.5-3MB/core		Up to 2MB/core Up to 16-way	
	L4 Cache	128MB ***		Not applicable	
	Side Cache	Not applicable		64MB eDRAM *** 128MB	
TLBs	ITLB	*Page translations* 4KB	*Entries* 128, 4-way Dynamic partition	ITLB	*Page translations* 4KB	*Entries* 128, 4-way Dynamic partition
		2MB/4MB	8, full assoc.		2MB/4MB	8, full assoc.
	DTLB	4KB	64, 4-way Fixed partition	DTLB	4KB	64, 4-way Fixed partition
		2MB/4MB	32, 4-way		2MB/4MB	32, 4-way
		1GB	4, 4-way		1GB	4, 4-way
	STLB	4KB + 2MB	1024, 8-way shared	STLB	4KB + 2MB	1536, 12-way Fixed partition
					1GB	16, 4-way Fixed partition

* Caches are associative, have a write-back policy and use 64-byte lines ** 5 cycles for complex addresses *** Only on the Iris Pro GPUs

2.4 Enhanced Intel SpeedStep Technology (EIST)

Enhanced Intel SpeedStep Technology (EIST), is a power and thermal management technology [27]. It was introduced with Pentium M in 2005 by Intel. This allows the operating system to modify the microprocessor's performance and power consumption dynamically. The underlying mechanism has total control over the frequency and voltage scaling. EIST achieves a 96% reduction in microprocessor core unavailability time compared to previous versions (10 μs versus 250 μs). Each frequency-voltage operating point (pair), is called a P-state (Performance-state). P0 is the highest power consuming state, which also corresponds to the high frequency mode (HFM). The rest of P-states (P1, P2, ... Pn-1, Pn) are defined in lower performance increments, with Pn state corresponding to the low frequency mode (LFM). These P-states, are defined in Advanced Configuration and Power Interface (ACPI). Depending on the workload, EIST is responsible for the transition from one P-state to another by keeping up with stored values in specific model-specific registers (MSRs). In the case that the microprocessor becomes idle, it enters in one of the C-states (idle power saving states), listed in Table 7. P-states are substates of C0, since CPU must be active in order to execute code.

Table 7: Processor C-states (Skylake U/Y microprocessor lines) [28]

State	Description
C0	Active mode (includes P0-Pn states)
C1	Auto halt
C1E	Auto halt, low frequency, low voltage
C2	Temporary state before C3, memory path open
C3	L1/L2 cashes flush, clocks off
C6	Save core states before shutdown and BCLK off
C7	C6 + LLC may be flushed
C8	C7 + LLC must be flushed
C9	C8 + most Uncore voltages/IA/GT/SA at 0V, V_{CCIO} unchanged
C10	C9 + all VRs at PS4 or LPM, 24MHz clock off

2.5 Intel Turbo Boost Technology 2.0

Intel has defined a power metric, known as Thermal Design Power (TDP), which basically is the maximum worst-case power that microprocessor can draw when it performs work. In the scope of this thesis, both ultra-low power Skylake and Haswell microprocessors, sponsor a maximum TDP of 15 W. TDP multiplied by time gives the heat amount a microprocessor can generate. If this amount gets past, it can be fatal for the microprocessor. However, there are times where the current workload does not fully utilize all the cores and thus the microprocessor operates way safely and far from its maximum potential. Then, given the opportunity Turbo Boost kicks in and temporarily overclocks the base CPU clock frequency. If the TDP threshold gets exceeded, then Turbo Boost will again lower the frequency speed to prevent overheating problems.

2.6 Intel Speed Shift Technology (SST)

Intel Speed Shift Technology (SST[3]), was introduced with Skylake in 2015. In legacy systems (see EIST), the OS was responsible for the P-states transition depending on CPU present utilization. However, that approach had two drawbacks [27]. Firstly, the evaluation of microprocessor needs, couldn't be any less than tens of milliseconds because that would be too aggressive and wouldn't offer any benefit otherwise. Secondly, the high level of OS in the computer abstraction hierarchy, does not allow close observation of workload behavior. With the SST, the CPU performs the actual P-state control autonomously (Image 7) while OS plays a supportive role providing suggestions on energy/performance preferences for a minimum quality of service (QoS).

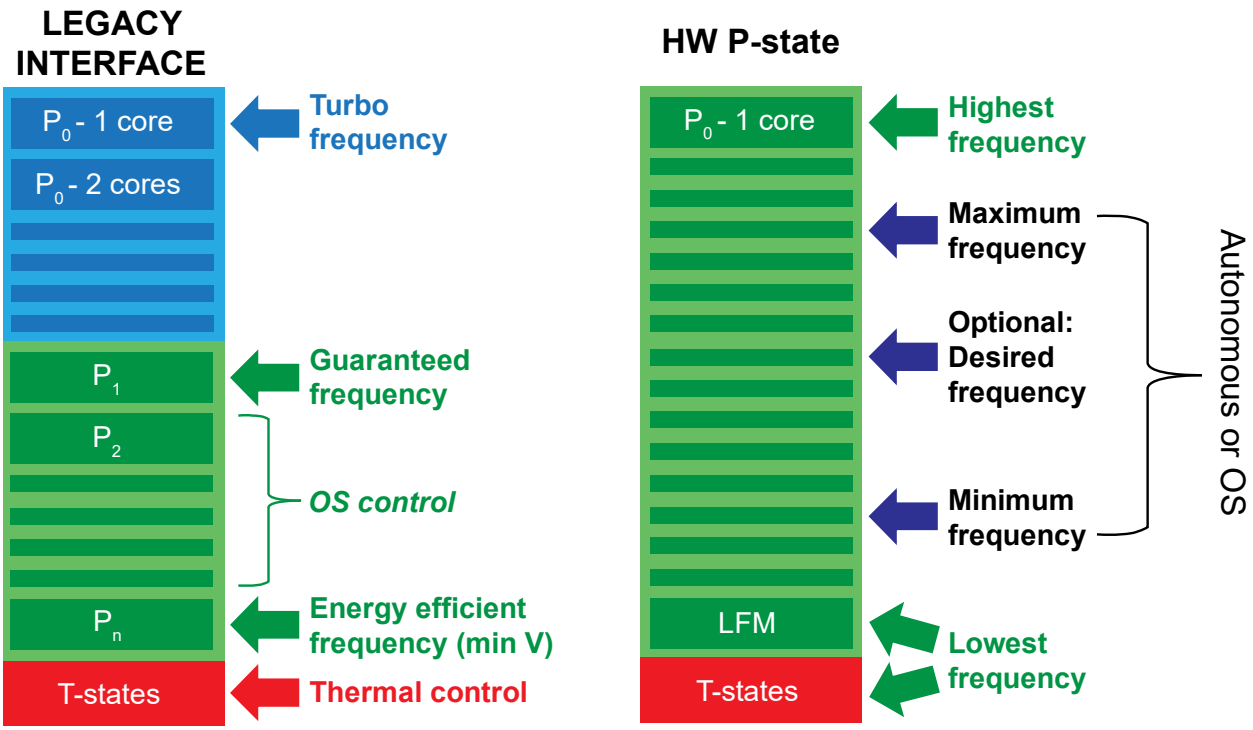


Image 7: Legacy versus Intel Speed Shift Technology control [29]

Therefore SST, provides higher performance, responsiveness and efficiency at power constrained form factors through autonomous algorithms.

2.7 Intel Hyper-Threading (HT) Technology

Traditional multiprocessing allowed the execution of concurrent threads on different processors, accelerating the program performance. However, there existed long idle times where CPU remained unutilized. Thus, Intel developed Hyper-Threading (HT) Technology[4], which makes a single physical microprocessor appear as two logical processors to the operating system. That, allowed the system to operate near peak bandwidth. HT is a processor simultaneous multithreading technology (SMT), where the

[3] Contrary to the other Intel's technologies presented here, SST is not an officially coined abbreviation.

[4] Intel Hyper-Threading (HT) Technology was firstly introduced in Xeon server processors in 2002 and in Pentium 4 processors later in the same year.

architectural state (Image 8) is duplicated for the two logical processors and the system appears to have four logical processors [30].

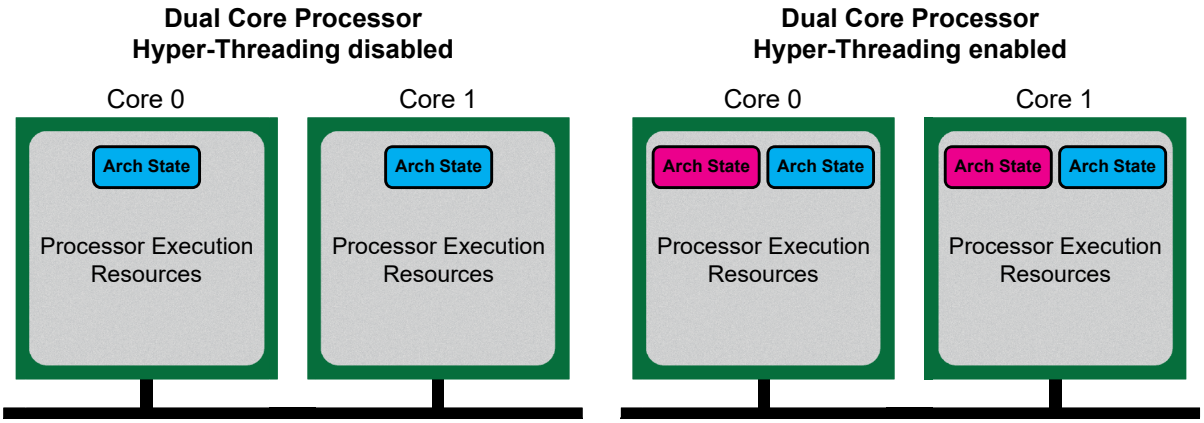


Image 8: Visualization of Hyper-Threading Technology

The architectural state consists of various microprocessor registers such as control, general purpose and model-specific registers (MSRs). While logical processors have their own advanced programmable interrupt controller (APIC), they share physical execution resources such as branch predictors, control unit, caches and memory bus. At the operating systems level, logical processors are managed as physical processors, where runnable tasks or threads are scheduled. For the best performance to be assured, two optimizations must be implemented by the operating system along with HT. Firstly, when only one logical microprocessor is active and uses all the microprocessor execution resources, the operating system should be able to stop the microprocessor execution allowing the microprocessor to go into lower power mode. Secondly, the operating system should be able to schedule threads to logical processors on different physical processors before scheduling multiple threads to the same physical microprocessor. In any occasion might arise, this optimization allows software threads to use different physical execution resources.

3. EXPERIMENTAL FRAMEWORK SETUP

In the first section of this chapter, we describe the hardware setup of our system on which we run our characterization experiments. On the second section the software setup of the framework follows, along with the methodology we used to perform the experiments. As we have already discussed, the focus of this study is to present the advances concerning the safe voltage margins between two different ultra-low power microprocessor generations i.e. Skylake and Haswell that have been introduced in section 2.3. Microprocessor vendors set pessimistic voltage margins (i.e., higher voltage than actually required for the microprocessor's operation), and thus, the energy efficiency is limited. The main idea is to expose the safe voltage points in which microprocessor can reliably operate. If the microprocessor operates below the safe voltage points it crashes unexpectedly. Our purpose is to find the lowest safe operation voltage V_{min} and compare the two generations safe voltage margins. Due to the correlation between power, voltage, and frequency we also conducted an evaluation of the new generation microprocessor (Skylake) for the half frequency and we further documented the V_{min} along with temperature and power measurements. Herewith, we also present the differences of our Skylake-based framework with the one used in a previous Haswell characterization study [16].

3.1 Hardware Setup

The system we used to perform the experiments includes an Intel i5-6200U Skylake microprocessor and 8 GB RAM. It belongs in the same ultra-low power microprocessor family as the older Intel i5-4200U Haswell microprocessor on which the previous study was conducted. The Skylake microprocessor consists of two physical cores. However, through Intel Hyper-Threading Technology each core can support two threads. In contrast to the previous study of Haswell microprocessor, the Intel Turbo Boost Technology 2.0 on the Skylake system was disabled through Enhanced Intel SpeedStep Technology (EIST) in system's firmware preferences. Thus, the system was set to operate at its base frequency of 2.3 GHz instead of its maximum potential of 2.8 GHz in the case that Turbo Boost was enabled. Another difference that exists in our Skylake system setup is the support of the new Intel Speed Shift Technology that we have already introduced in section 2.6. Finally, the starting nominal voltage for this system was defined at V_{S_nom} = 0.890 V while for the Haswell one was V_{H_nom} = 0.844 V.

3.2 Software Setup and Methodology

Both Skylake and Haswell systems used Windows 10 as operating system. In order to conduct the experiments, we used a set of benchmarks and software utilities that form the base of our framework and helped in the direction of results collection.

On the following page in Image 9, the three-phase experimental framework setup is presented. In Phase I we make the appropriate initialization of the system in terms of software settings adjustments. In order to conduct an extended study and expose any process variability between Skylake cores, we set each time the microprocessor's affinity accordingly. This means that we constrain on purpose the execution of the processes on specific cores. For example, in the case we want to characterize thread 0 of Core 1 we set all system's processes to run on either thread 2 or thread 3 of Core 2 except the benchmark process, which we set to be run on thread 0 of Core 1. This has two benefits. First, we always know that the thread under characterization won't be used by other

processes. Secondly, in case of hardware errors, we can distinguish on which thread they occurred and identify if they were caused by the benchmark process or any other process.

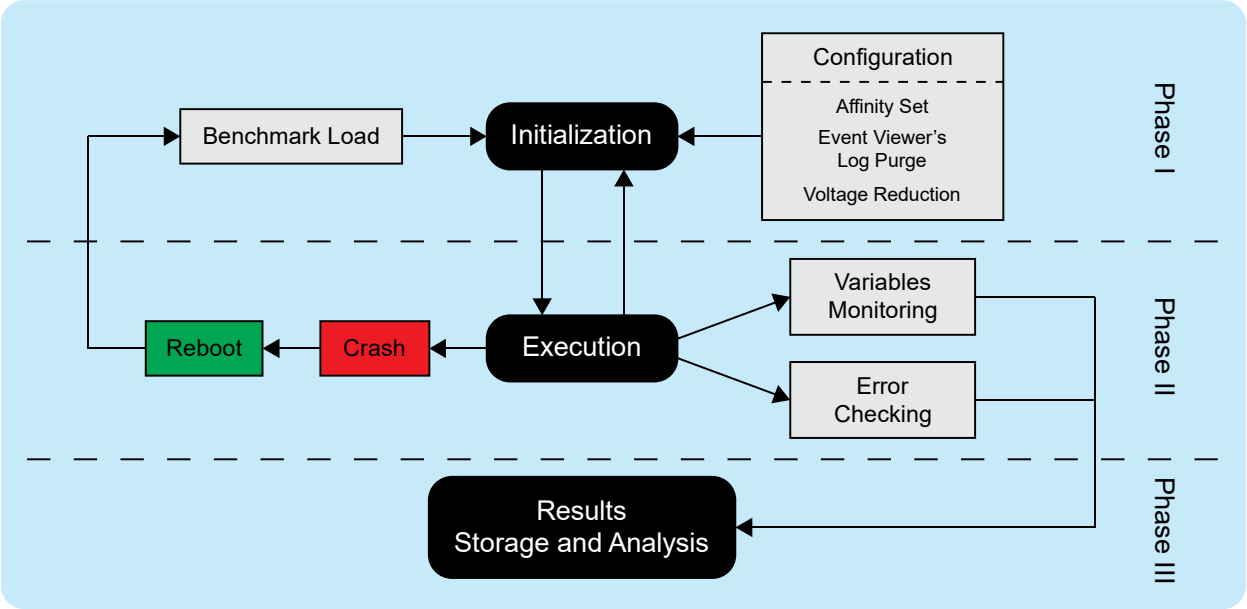


Image 9: Experimental framework setup

In order to set the microprocessor affinity, we used the utility Process Lasso [31], which sets and monitors the affinity in a more convenient way than setting it from Windows PowerShell [32].

Another important utility we use is Event Viewer [33]. It is integrated in Windows operating system and it basically logs system events. After each benchmark execution we visit Event Viewer and check if there were occurred any hardware errors, reported by WHEA as previously discussed in section 2.2. Whenever a hardware error occurs the Event Viewer is cleared from all logs to avoid any confusions with later benchmark executions. If things go wrong, it also helps to keep an eye on time and check the timestamps of benchmark execution and hardware error occurrences.

The third and most important element in the configuration setup is the voltage offset from the nominal voltage of 0.890 V. This after all is the main variable we track for the characterization of the microprocessor. For the voltage reduction we use Intel Extreme Tuning Utility (XTU) [34]. Although, there exists a smaller step size option of 1 mV, in the previous Haswell study it is noted that the actual granularity that the system provides is 5 mV. For this reason, we reduce the voltage of the microprocessor by 5 mV steps too. Apart from the voltage reduction usage of this utility, we used it in Phase II of our experimental framework for variables monitoring such as temperature and power. However, during our study period we experienced an abnormal behavior in the operation of this utility. Sometimes, after the system recovered from a crash due to a high voltage reduction, the monitors of Intel XTU showed zeroed values. Surprisingly, the workaround to this problem was to force a new system crash.

We should note down, that later in our study we found the HWiNFO [35] utility, which offers a vast amount of monitoring variables along with WHEA error reporting. Thus, for the rest of the study, we also used this tool on the side.

Since our study evaluated the microprocessor under full and half frequency levels, the configuration setup of Phase I, included also the frequency handling of the

microprocessor. This was done from the power options of the operation system. In the case of maximum base frequency, we set the performance to 100% and in the case of half speed performance we set it to 50%. In both cases a stable frequency was provided throughout the benchmarks' execution.

The final step in the Initialization phase of the experimental framework (Phase I), is the benchmark loading in order to be executed on command prompt. We used a selection of 10 representatives (shown in Table 8) from SPEC CPU2006 benchmark suite [36]. It is extended by two more benchmarks (*dealII*[5] and *bwaves*) than the previous Haswell study.

Table 8: Benchmark representatives from SPEC CPU2006 suite

Benchmark	Language	Type	Application Area
bzip2	C	Integer	Compression
mcf	C	Integer	Combinatorial Optimization
milc	C	Floating Point	Physics / Quantum Chromodynamics
namd	C++	Floating Point	Biology / Molecular Dynamics
hmmer	C	Integer	Search Gene Sequence
h264ref	C	Integer	Video Compression
gobmk	C	Integer	Artificial Intelligence: Go
zeusmp	Fortran	Floating Point	Physics / CFD
dealII	C++	Floating Point	Finite Element Analysis
bwaves	Fortran	Floating Point	Fluid Dynamics

In the Execution phase (Phase II) of each benchmark the actual monitoring of temperature and power variables was performed. Phase I and Phase II alternate between each other after each benchmark execution. If the benchmark execution completes without a system crash, we check for any other type of errors (in Table 9) we use the same error classification schema as in [8] and then we increase the voltage offset for a new benchmark run. This is done until a system crash occurs.

Table 9: Error classification

Error	Description
SDC (Silent Data Corruption)	The benchmark was successfully completed but there was a mismatch between the program output and the correct (golden) output
AC (Application Crash)	The application process was not terminated properly (the exit value was different than zero)
CE (Corrected Error)	Errors were detected and corrected by hardware
UE (Uncorrected Error)	Errors were detected but not corrected by hardware (this type of error always led to a system crash)
SC (System Crash)	The system was unresponsive (frozen screen etc.)

[5] Benchmark *dealII* is read as deal2 and should not be mistakenly read as dea3, because of the selected typesetting font.

Then, as already depicted in Image 9, the system reboots either automatically or manually (by the operating system's recovery mechanisms) and a new set of benchmark executions in incremental voltage offsets is performed until a new crash happens. This benchmark execution cycle is done three times in total, for each of the six defined affinities (as shown in Table 10). In the All Cores no HT affinity configuration, Hyper-Threading is disabled.

Table 10: Affinity configurations of microprocessor

	Active Cores			
Affinity Configuration	**Core 0**	**Core 1**	**Core 2**	**Core 3**
Core 0	X			
Core 1		X		
Core 2			X	
Core 3				X
All Cores	X	X	X	X
All Cores no HT	X		X	

We assume that a core is fully characterized only after three distinct system crashes at the same voltage offset. This part is handled differently in the previous study, where only one crash among any of the three benchmarks runs at any voltage offset, was enough to fully characterize an affinity configuration, as show on the next page in Image 10. In our setup, we mark the first occurrence of crash among the three benchmark runs of any specific voltage offset that this happens. Then we further investigate the behavior of the chip around the voltage offset area that the system crashed, for the same affinity configuration. This is done for the following scenarios:

1. If the crash occurs for the first time before the third benchmark run at a voltage offset i.e. either at 2nd or 1st trial, we complete the 3-benchmark characterization runs for that specific voltage offset after the system reboot, by trying to execute the benchmark either 1 or 2 more times respectively. If all three tries lead to system crashes, we conclude the characterization for that affinity configuration.
2. If the crash occurs on the third benchmark execution, we mark this voltage offset as fully characterized, but after the system reboots, we again try to reach the next lower voltage offset and document a total of three crashes for that or even lower voltage offsets.

In order to do this, we deplete the voltage pool set vertically in larger than 5 mV steps until we reach the voltage offset 5 mV before the suspicious voltage offset that the system crashed initially. This is done without any benchmark executions between the intermediate voltage offsets. To avoid large voltage oscillations that happen after any larger than 5 mV increase in the voltage offset, we let the system to rest for a few seconds at every voltage offset increase. In this way we ensure that the voltage is stabilized around shorter deviations. Then, depending on which of the above two scenarios we follow, we reach the appropriate voltage offset. In our opinion, this exhaustive approach offers an even more fine-grained characterization of the microprocessor as we will explain later in section 4.1.

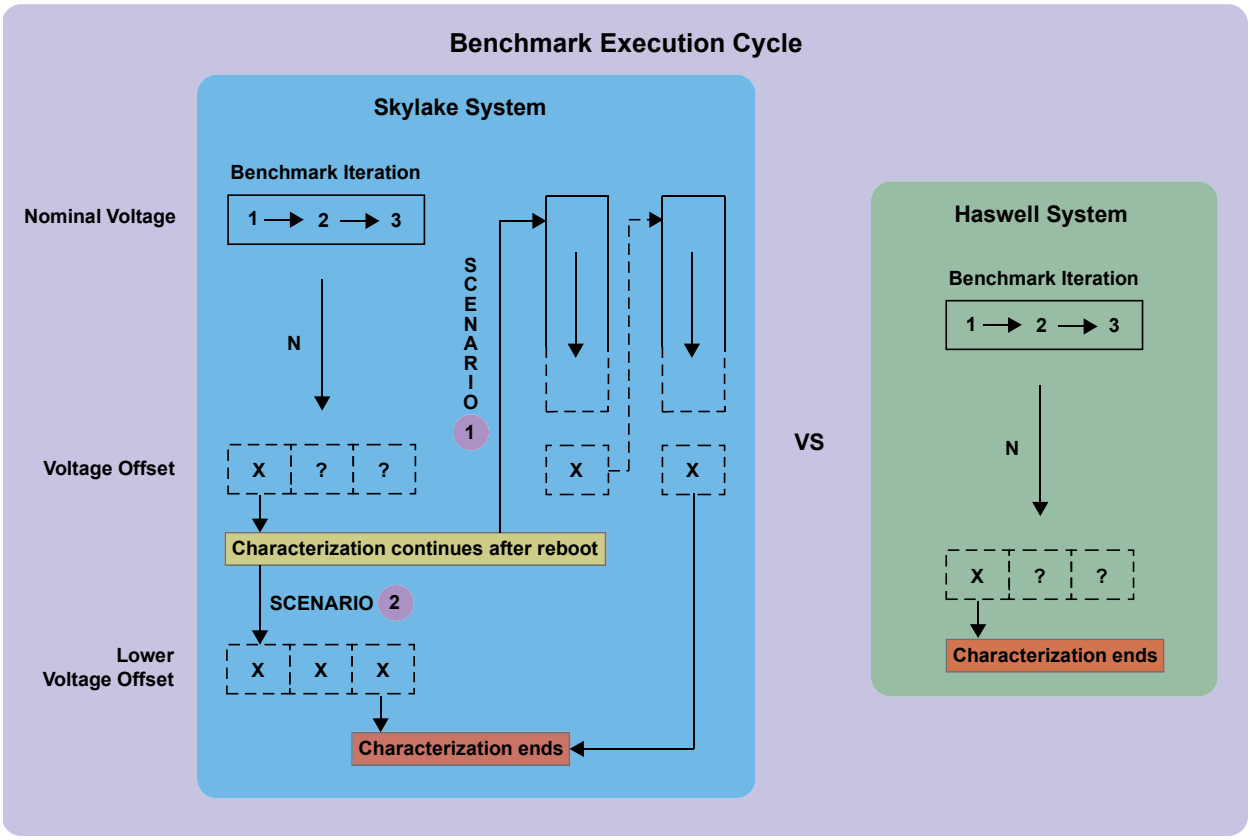


Image 10: Skylake vs Haswell benchmark execution cycle (X indicates a crash)

In the final phase (Phase III), the measurements and observations that were collected during the benchmark executions were documented and further analyzed to get the characterization results that we will show in the next section of this study.

We will complete the presentation of the experimental framework by remarking that it comes with a limitation. Everything must be done manually, from the loading and execution of the benchmarks to measurements observation and documentation. During the 4 months period that the experiments were conducted, it is estimated that there were collected about 10.000 sets of voltage, temperature and power measurements along with the observation for any type of error. A more automated way to gather the measurements under Windows OS would be less cumbersome. We conclude this section with Table 11 on the following page, where the main configuration differences of Skylake and Haswell systems are summarized.

Table 11: Configuration overview of i5-6200U and i5-4200U systems

Parameter		System	
		i5-6200U	**i5-4200U**
Launch Date		Q3 ‘15	Q3 ‘13
ISA / uArch		x86 / Skylake	x86 / Haswell
Core / Thread Counts		2 / 4	2 / 4
Clock Rate	**Base**	2.3 GHz	1.6 GHz
	Turbo	2.8 GHz	2.6 GHz
Lithography		14 nm	22 nm
L1 Instruction Cache		2 x 32 KB 8-way set associative	
L1 Data Cache		2 x 32 KB 8-way set associative	
L2 Cache		2 x 256 KB 4-way set associative	8-way set associative
L3 Cache		3 MB 12-way set associative	
Max TDP		15 W	
Nominal Voltage		0.890 V	0.844 V
RAM		8 GB 1600 MHz DDR3	Size not available
Operating System		Windows 10	
SPEC CPU2006 Representatives		10	8
Characterization Cycle Completion		3 crashes	1 crash
Enhanced Intel SpeedStep Technology		Disabled	Enabled
Intel Turbo Boost Technology 2.0		Disabled	Enabled
Intel Speed Shift Technology Support		Yes	No
Intel Hyper-Threading Support		Yes	Yes

4. CHARACTERIZATION RESULTS

In this chapter we present the characterization results of our study and we prove that there is a marginal advancement in safe voltage margins between ultra-low power microprocessor generations. Concerning the Haswell generation, we used the results that were demonstrated in a previous study [16]. Specifically, we are going to present in full detail, the characterization results for i5-6200U at both 2.3 GHz (full speed) and 1.2 GHz (half speed). Here, 1.2 GHz is the half frequency value of 2.3 GHz as reported by Windows Task Manager instead of the accurate 1.15 GHz value, as someone might have expected. For both these configurations, we gradually reduced the voltage of the microprocessor by 5 mV steps. Then a *core-to-core* and *benchmark-to-benchmark* variation follow for these two different setups. The same is done for i5-6200U at full (base) speed and i5-4200U at 2.6 GHz (full turbo speed). Along this procedure we observed the system for any abnormal behavior (Silent Data Corruption, Error Correction Code errors, etc.). The two microprocessor generations are then characterized in terms of *total voltage reduction percentage* and *core resilience* metrics. A full evaluation was also done regarding temperature and power metrics. This was done extensively in *core-to-core*, *chip-to-chip* and *benchmark-to-benchmark* variations.

4.1 Visualization of Results

In order to better understand how the results were documented for each benchmark run, on the next page in Table 12 we present an exemplary overview regarding the characterization of i5-6200U at 1.2 GHz for dealII benchmark. On the first run of any benchmark there is not issued any undervolting. This is the nominal level of voltage, which for the purpose of this Skylake study, has been defined to be V_{S_nom} = 0.890 V for all benchmarks both at full and half speed modes. The temperature and power values we monitor for this first benchmark execution are also used as nominal values for the corresponding system efficiency metrics that we will discuss later in this chapter.

By following the procedure as discussed in section 3.2 until a crash occurs, we found that after the system reboot there existed occasions where we succeeded to reach an even lower voltage offset and run benchmarks successfully. In this case the three system crashes occurred at later voltage offsets. We hypothesize that this gap range between first and final three crash voltage offsets happens because of system fatigue decompression This means that as we continuously run benchmarks voltage offset after voltage offset, the stiffness of the chip increases due to accumulated workload. When a crash finally occurs, the system reboots and the chip returns to a lower stiffness state (workload decompression). This allows to pass through higher voltage offsets than the one where a crash happened previously.

An example of this happened in the case of All Cores no HT affinity. In column 2 (i.e. second benchmark run) of voltage offset 105 mV a crash occurred for the first time. When the system rebooted, we did reach not only the same voltage offset and run the dealII benchmark one more time, but also reach a lower voltage offset (110 mV) without crash among the three benchmark runs that followed. In our opinion this offers an even more detailed characterization of the chip.

We remind that we conclude the characterization of a core only if we mark three crashes on the same voltage offset. In most of the experiments we got three consecutive crashes for the same voltage offset even after a system reboot. A representation of this can be seen in the columns of Core 3 and All Cores affinities of Table 12. For Core 3 the three crashes happen 120 mV below nominal voltage while for All Cores this happens one voltage offset earlier i.e. at 115 mV.

Table 12: Exemplary overview of errors per run and lowest safe voltage offset of i5-6200U @ 1.2 GHz characterization for dealll benchmark

dealll characterization at 50% speed																							
	Core 0			Core 1			Core 2			Core 3			All Cores			All Cores no HT		
Undervolting (mV)	1st	2nd	3rd	1st	2nd	3rd	1st	2nd	3rd	1st	2nd	3rd	1st	2nd	3rd	1st	2nd	3rd
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
75	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
85	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
95	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
105	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	UE	0
110	0	0	0	0	0	0	UE	X	UE	0	0	0	0	0	0	0	0	0
115	0	0	0	UE	0	UE	-	-	-	UE	UE	UE	UE	UE	UE	UE	UE	UE
120	UE	UE	CE	UE	UE	UE	-	-	-	UE	UE	UE	-	-	-	-	-	-
125	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
130	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

The symbol UE indicates that for that voltage an Uncorrected Error was reported by WHEA (section 2.2). Opposed to other studies such as [8], in our study an Uncorrected Error was always fatal, which led to a Blue Screen of Death (BSOD) [37] and an immediate reboot of the operating system. However, there were also occasions such as the one depicted in the second benchmark run column of Core 2 characterization, where a system crash occurred without a UE. This is symbolized with the letter X. Concerning the Corrected Errors they were documented as CE. Such an error can be seen on the third benchmark run of Core 0 characterization (voltage offset 120 mV). Finally, all the zero values in Table 12 indicate that no error or any other abnormal behavior occurred during the benchmark execution.

Every benchmark series of execution presented in this study comes from an analysis akin to the previous one, in order to extract the safe V_{min} of each affinity for 10 representative benchmarks of SPEC CPU2006 suite (Table 8 on page 43) and then of the whole microprocessor. In the previous paradigm for *dealII* benchmark the lowest safe voltage of i5-6200U @ 1.2 GHz is V_{min} = 0.790 V.

4.2 Intel Core i5-6200U Skylake Microprocessor Full Speed Study

In this section we will start the presentation of the first part of the results i.e. the characterization of Skylake microprocessor at its maximum base frequency of 2.3 GHz.

In the case a CE error is not shown in the diagrams, it means that after the CE a UE followed, which led to system crash. We assume that a system crash is of higher importance and thus we show for that benchmark run (which gave at first a CE), only the crash, depicted in black color.

In all diagrams of both full and half speed studies, we adopt a four-color scheme that represent the following regions:

- Safe region (blue): During the benchmark runs no error or any other abnormal behavior was observed.
- Unsafe region (grey): This region includes voltage ranges in which we observed any type of error except a system crash. However, in our study those regions do not exist at all. In the case we observed any non-crash error this happened for single benchmark runs and therefore no unsafe regions were formed.
- Crash region (black): This region defines the beginning of system crashes.
- Safe after crash zones (yellow): These regions emerged from the procedure we extensively analyzed in Chapter 3. They are basically voltage ranges of normal operation between system crashes.

In i5-6200U at 2.3 GHz study, safe after crash zones occurred for *milc* (Figure 3, Core 0/2/3), *namd* (Figure 4, Core 0), *hmmer* (Figure 5, Core 0/2), *gobmk* (Figure 7, Core 1), *dealII* (Figure 8, Core 2/3), *zeusmp* (Figure 9, Core 0/1/2) and *bwaves* (Figure 10, Core 0). The largest range between the first and the last three crashes occurred for *bwaves* (110 mV and 130 mV respectively). We assume, that these zones are stochastic and are depended on system's present conditions. However, we cannot ignore their existence given the high number of occurrences, we observed such behavior.

4.2.1 Bzip2

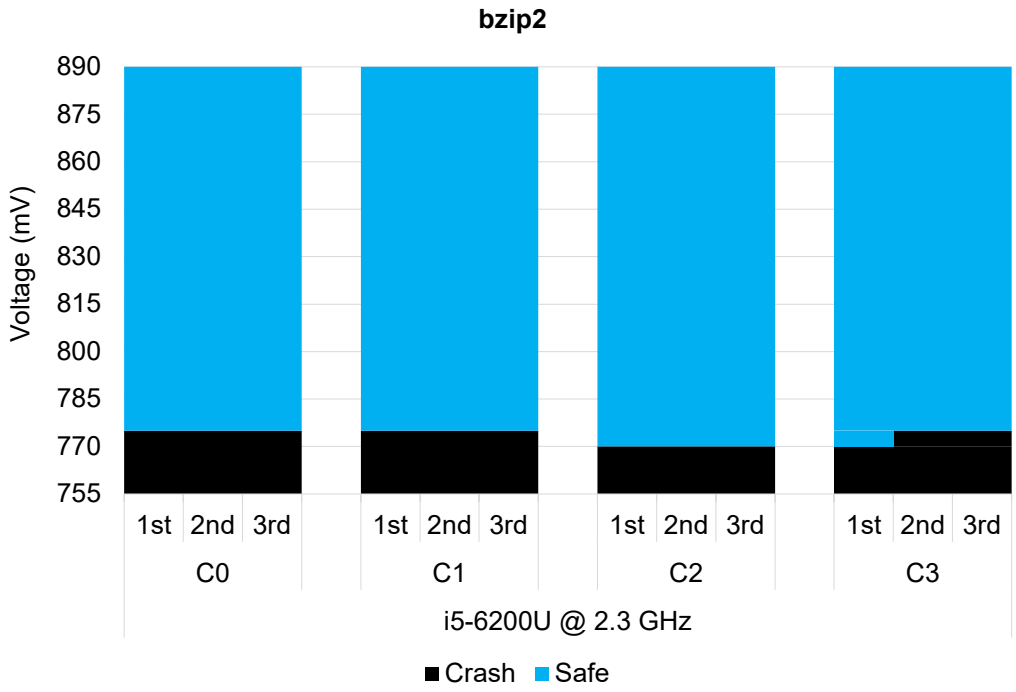


Figure 1: i5-6200U @ 2.3 GHz characterization for bzip2 benchmark

4.2.2 Mcf

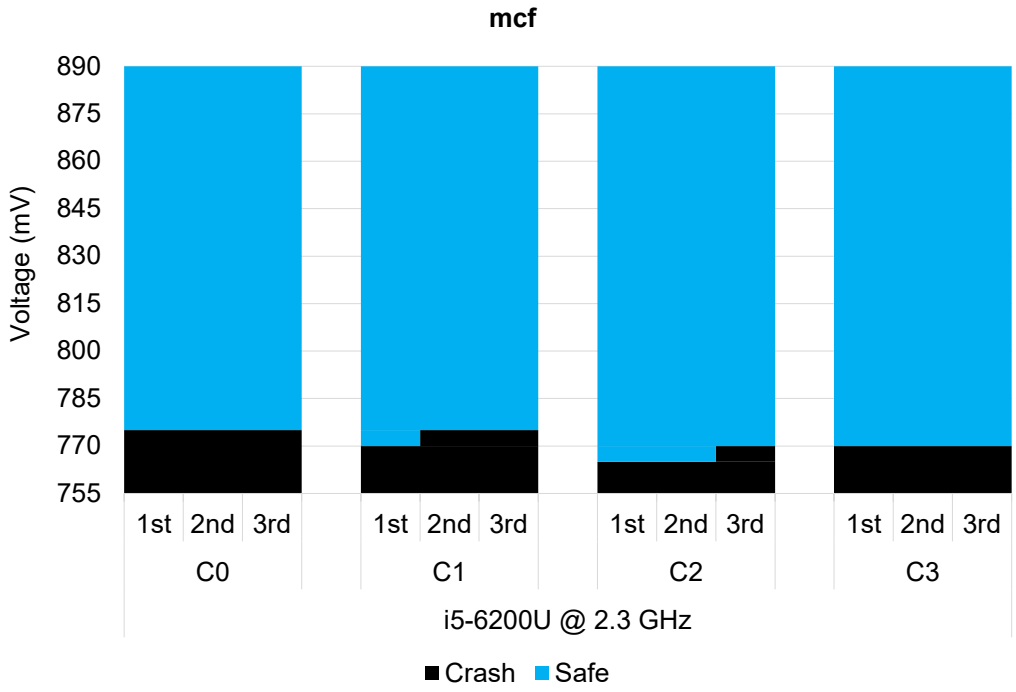


Figure 2: i5-6200U @ 2.3 GHz characterization for mcf benchmark

4.2.3 Milc

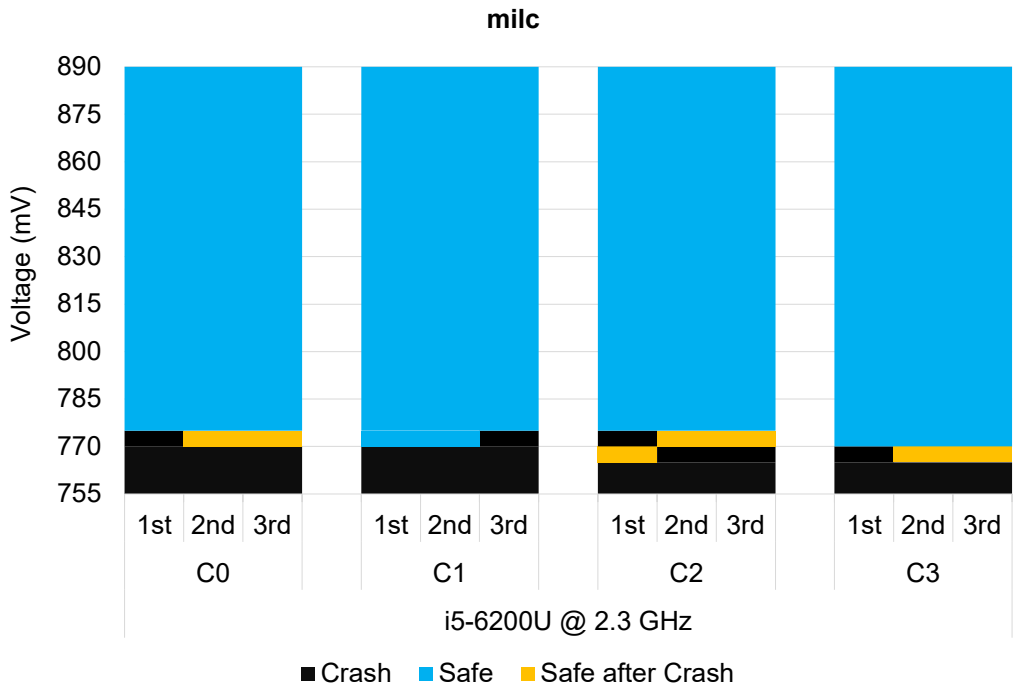


Figure 3: i5-6200U @ 2.3 GHz characterization for milc benchmark

4.2.4 Namd

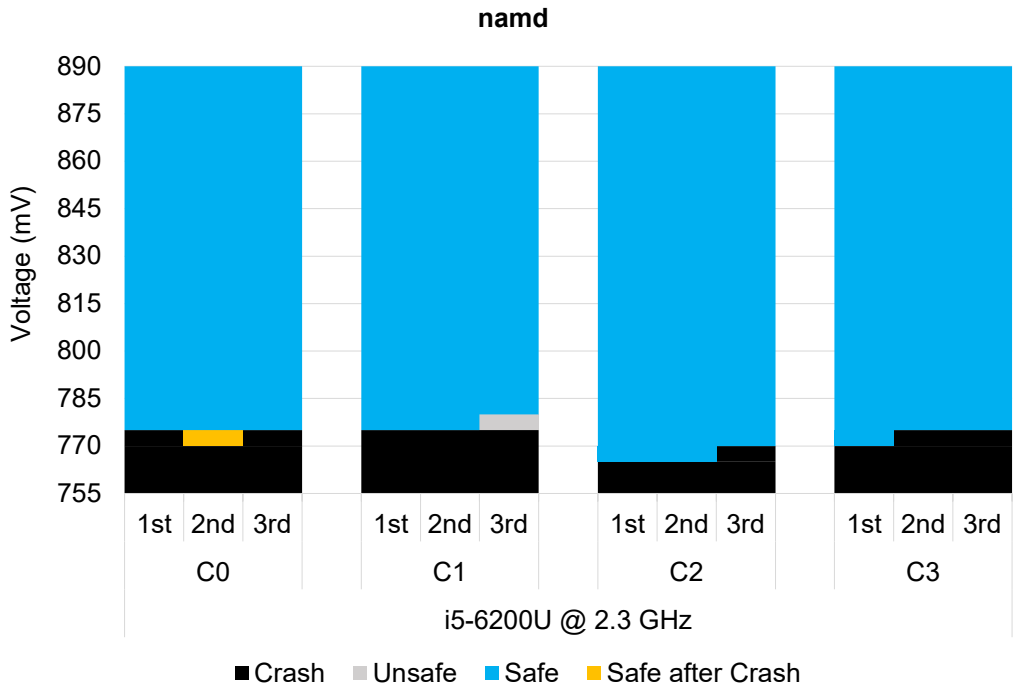


Figure 4: i5-6200U @ 2.3 GHz characterization for namd benchmark

4.2.5 Hmmer

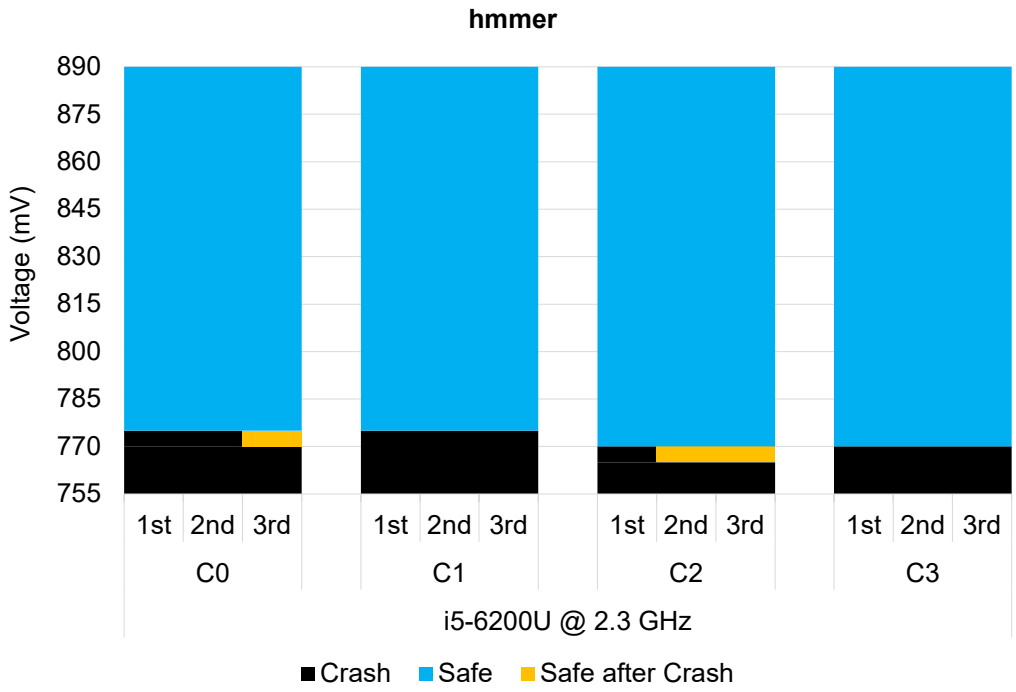


Figure 5: i5-6200U @ 2.3 GHz characterization for hmmer benchmark

4.2.6 H264ref

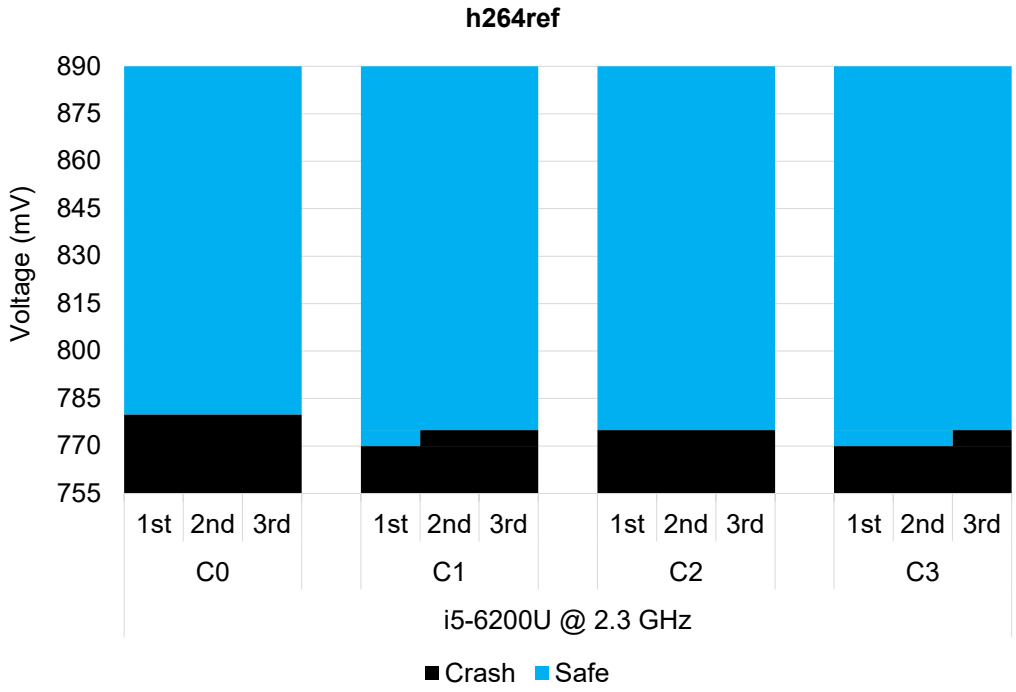


Figure 6: i5-6200U @ 2.3 GHz characterization for h264ref benchmark

4.2.7 Gobmk

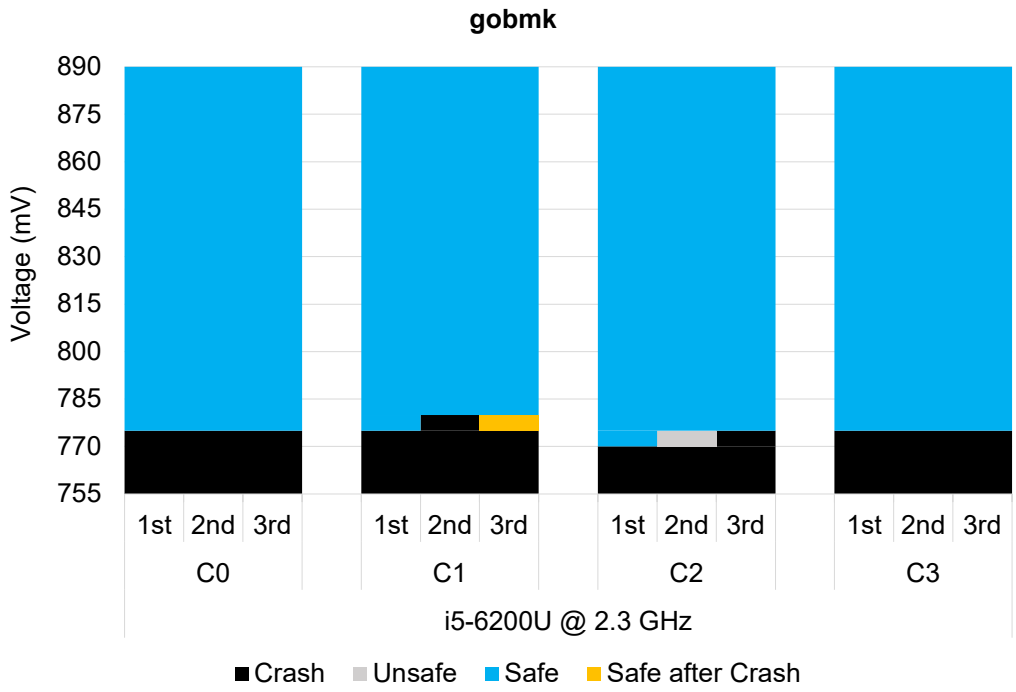


Figure 7: i5-6200U @ 2.3 GHz characterization for gobmk benchmark

4.2.8 Dealll

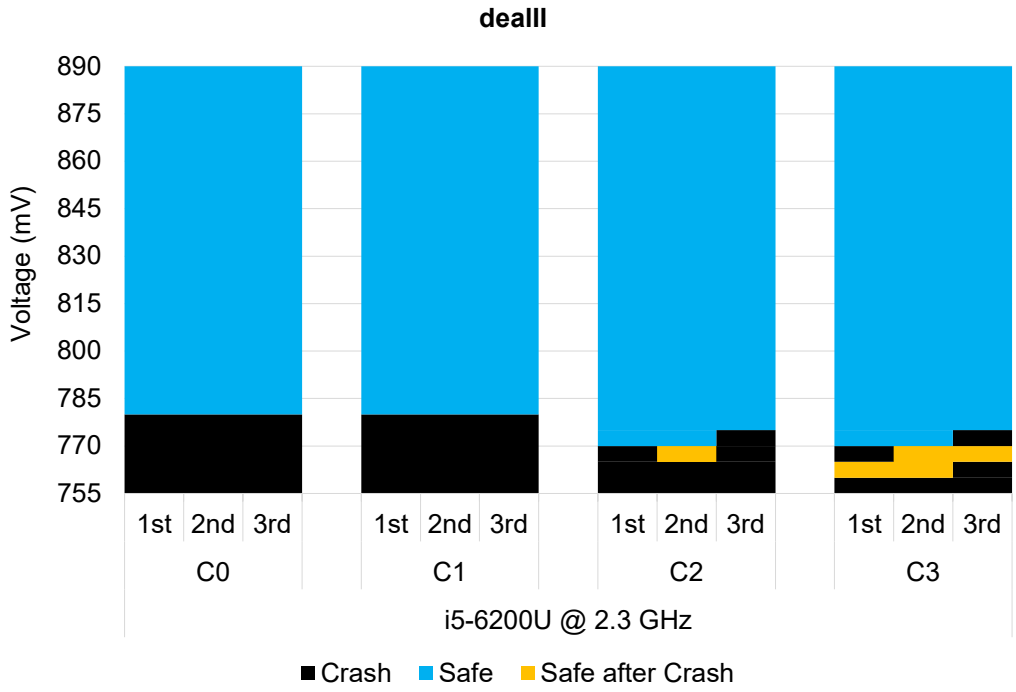


Figure 8: i5-6200U @ 2.3 GHz characterization for dealII benchmark

4.2.9 Zeusmp

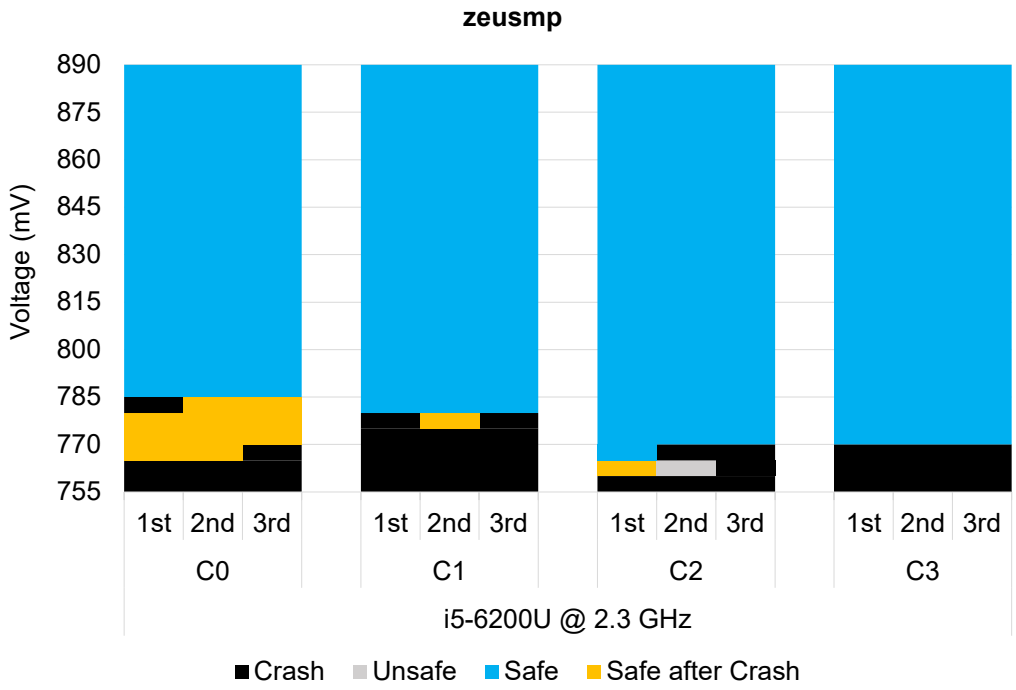


Figure 9: i5-6200U @ 2.3 GHz characterization for zeusmp benchmark

4.2.10 Bwaves

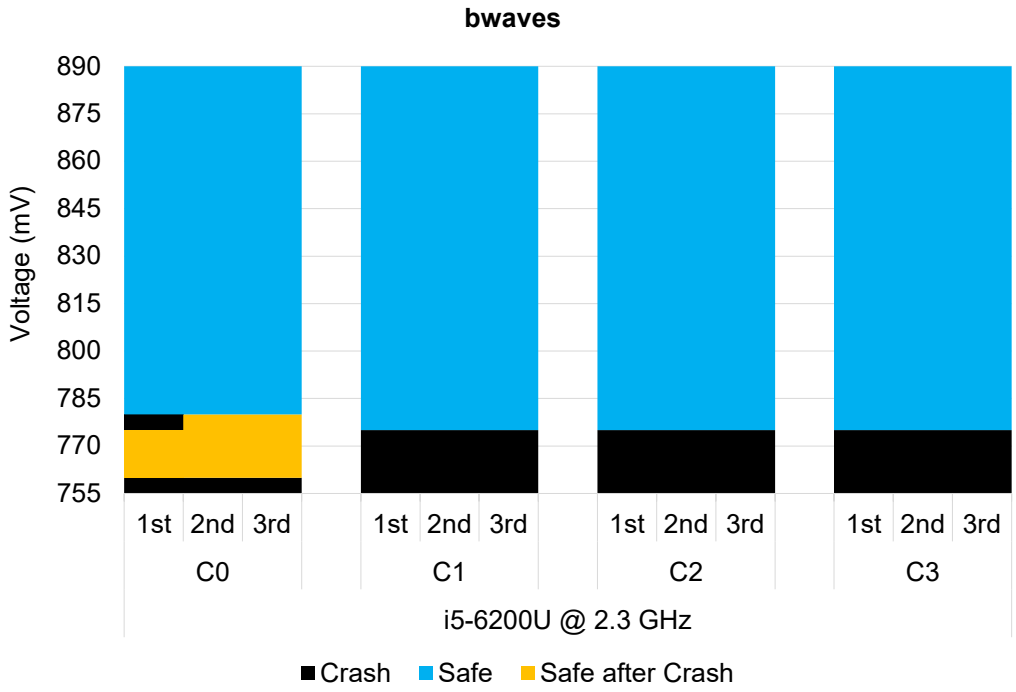


Figure 10: i5-6200U @ 2.3 GHz characterization for bwaves benchmark

4.3 Intel Core i5-6200U Skylake Microprocessor Half Speed Study

Herewith, we present the characterization results of Skylake microprocessor at half the base frequency i.e. 1.2 GHz.

4.3.1 Bzip2

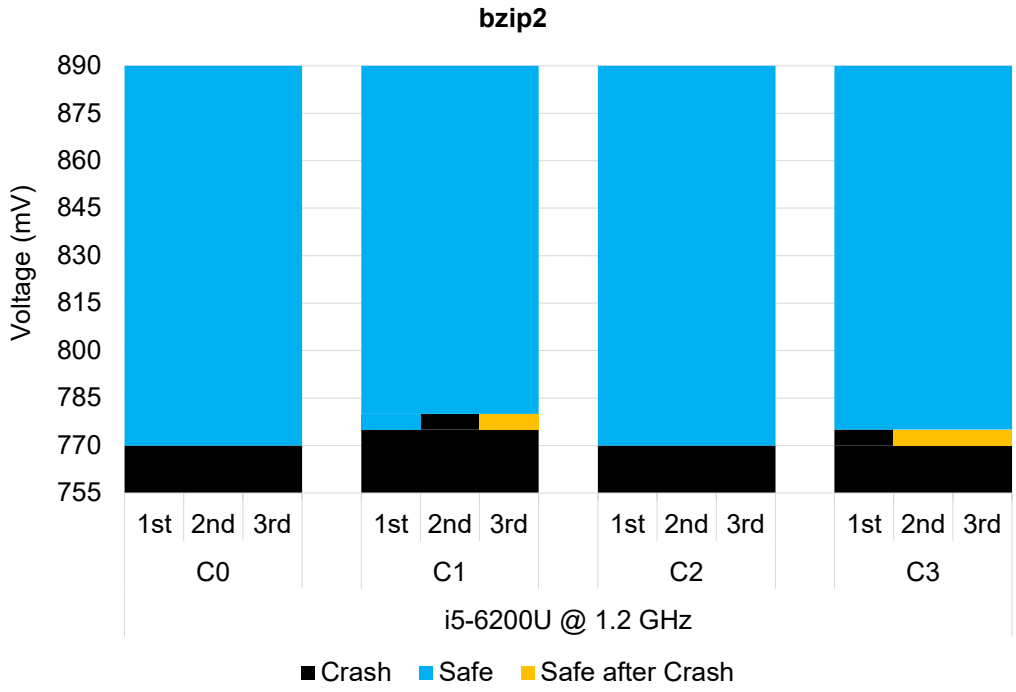


Figure 11: i5-6200U @ 1.2 GHz characterization for bzip2 benchmark

4.3.2 Mcf

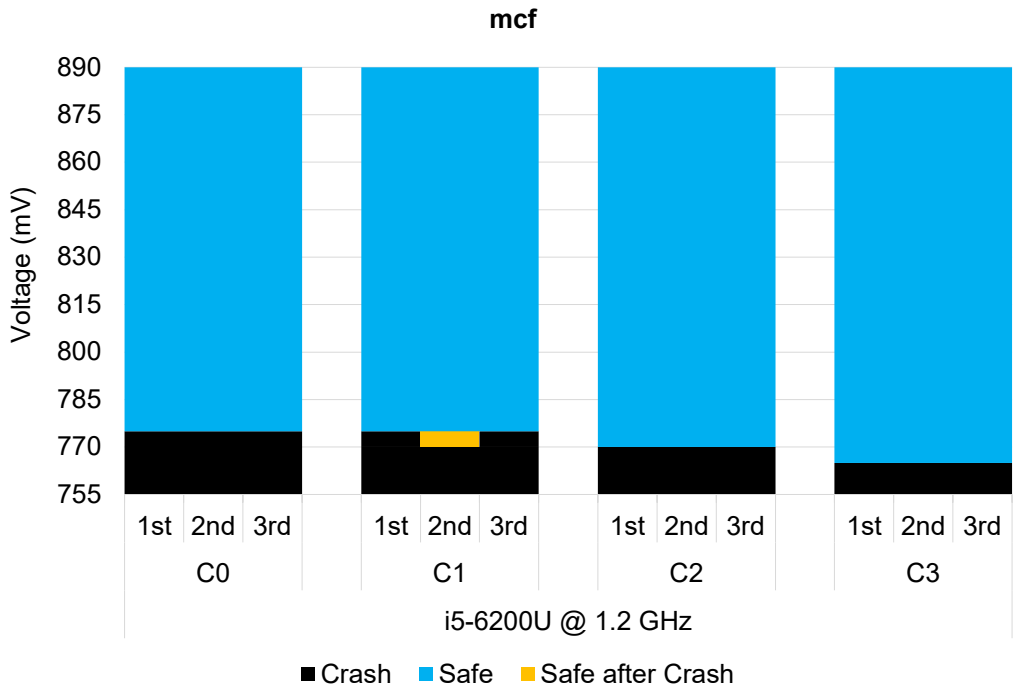


Figure 12: i5-6200U @ 1.2 GHz characterization for mcf benchmark

4.3.3 Milc

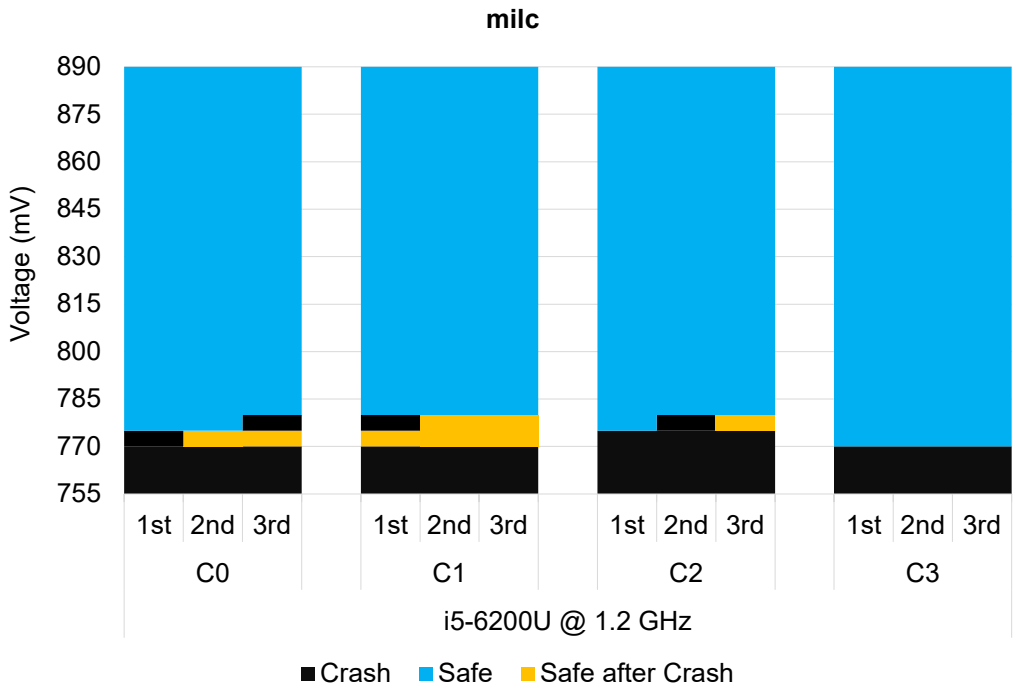


Figure 13: i5-6200U @ 1.2 GHz characterization for milc benchmark

4.3.4 Namd

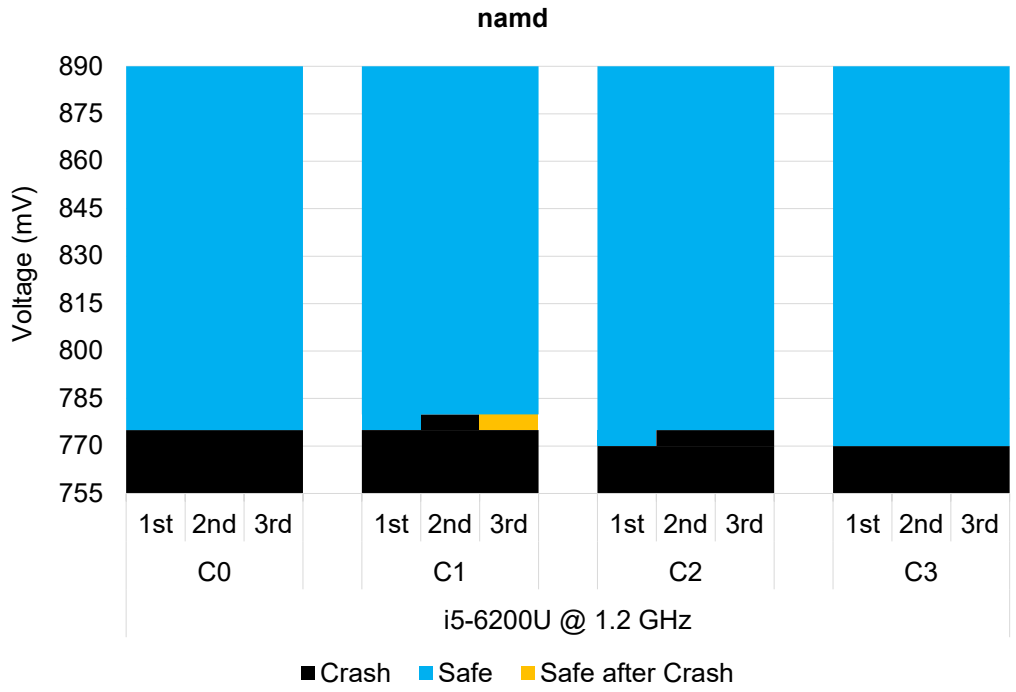


Figure 14: i5-6200U @ 1.2 GHz characterization for namd benchmark

4.3.5 Hmmer

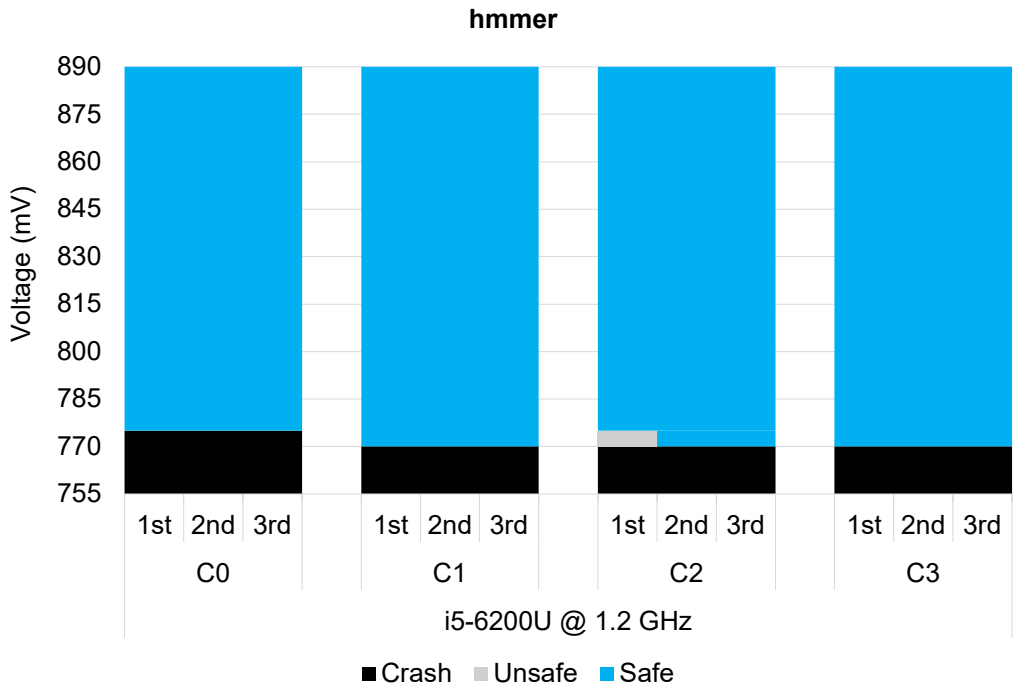


Figure 15: i5-6200U @ 1.2 GHz characterization for hmmer benchmark

4.3.6 H264ref

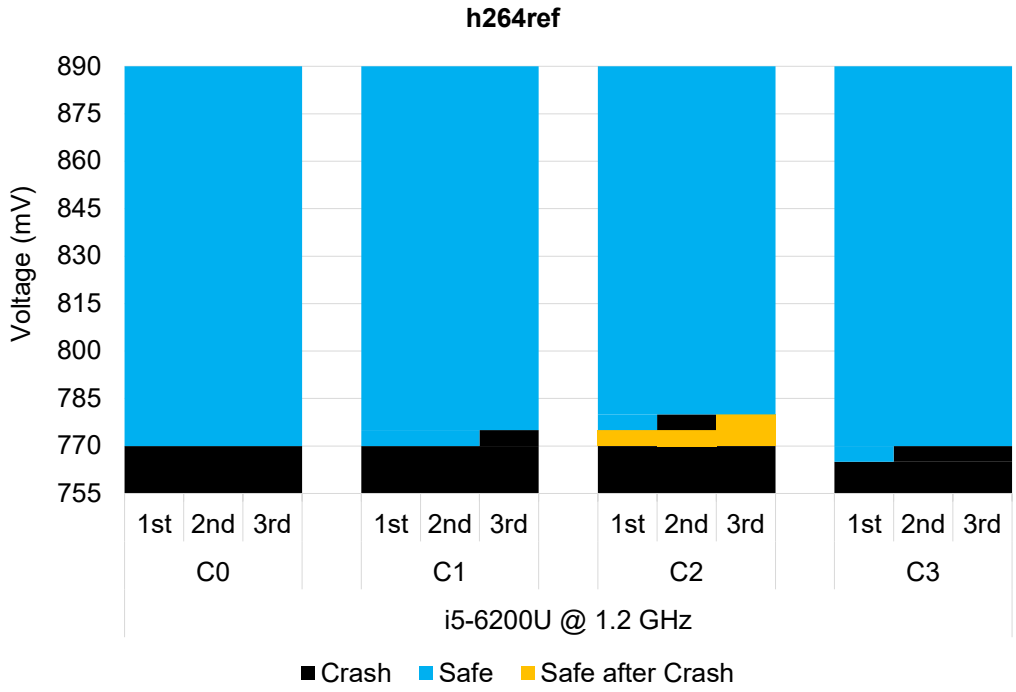


Figure 16: i5-6200U @ 1.2 GHz characterization for h264ref benchmark

4.3.7 Gobmk

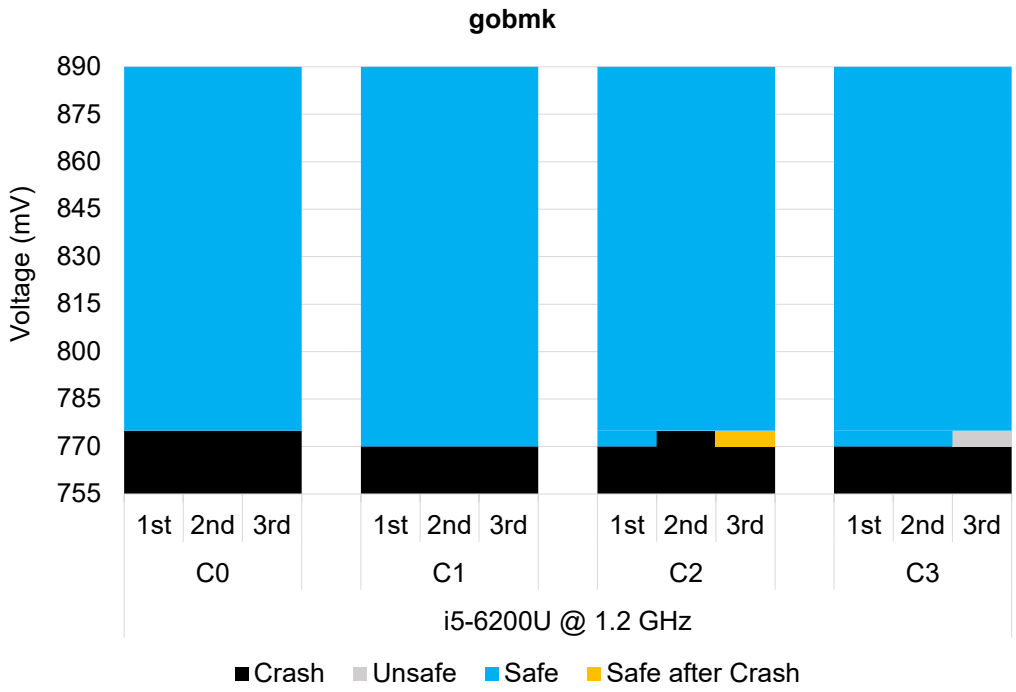


Figure 17: i5-6200U @ 1.2 GHz characterization for gobmk benchmark

4.3.8 Dealll

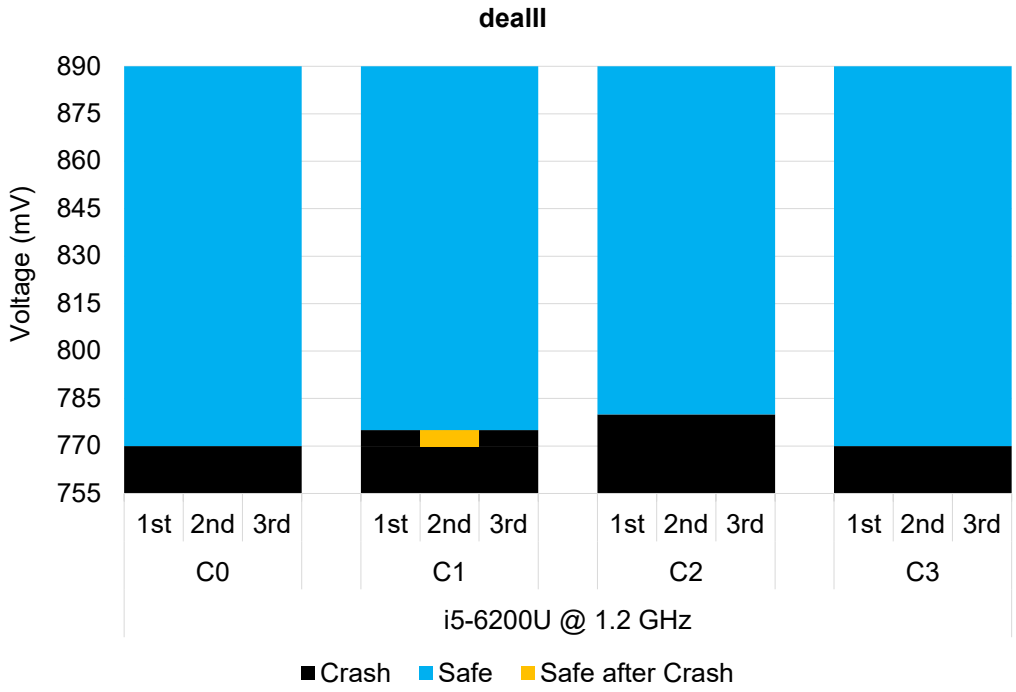


Figure 18: i5-6200U @ 1.2 GHz characterization for dealII benchmark

4.3.9 Zeusmp

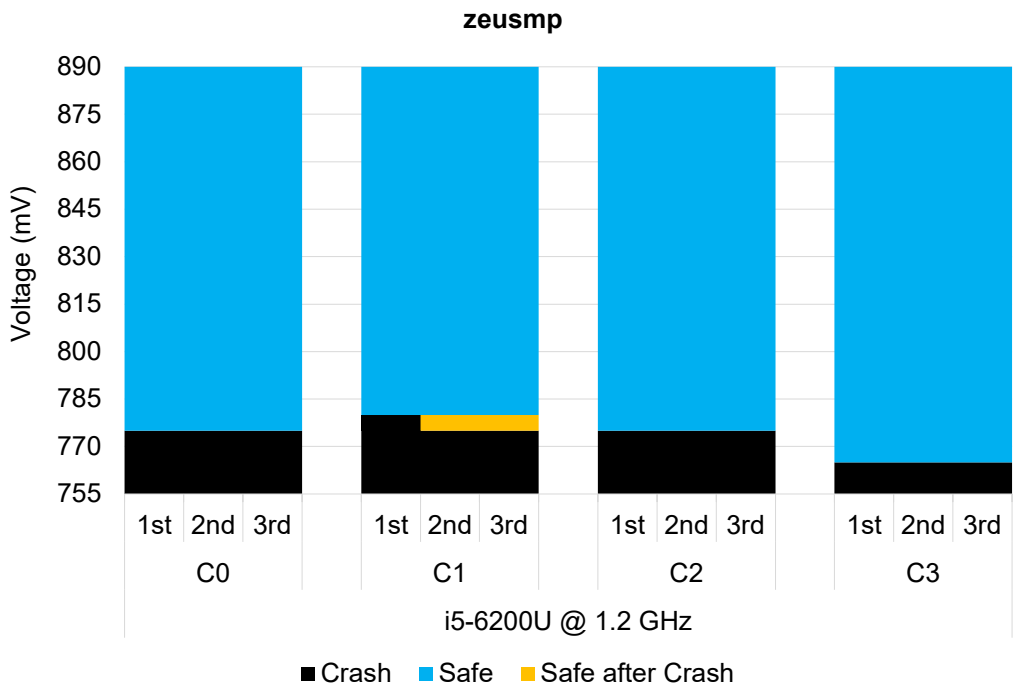


Figure 19: i5-6200U @ 1.2 GHz characterization for zeusmp benchmark

4.3.10 Bwaves

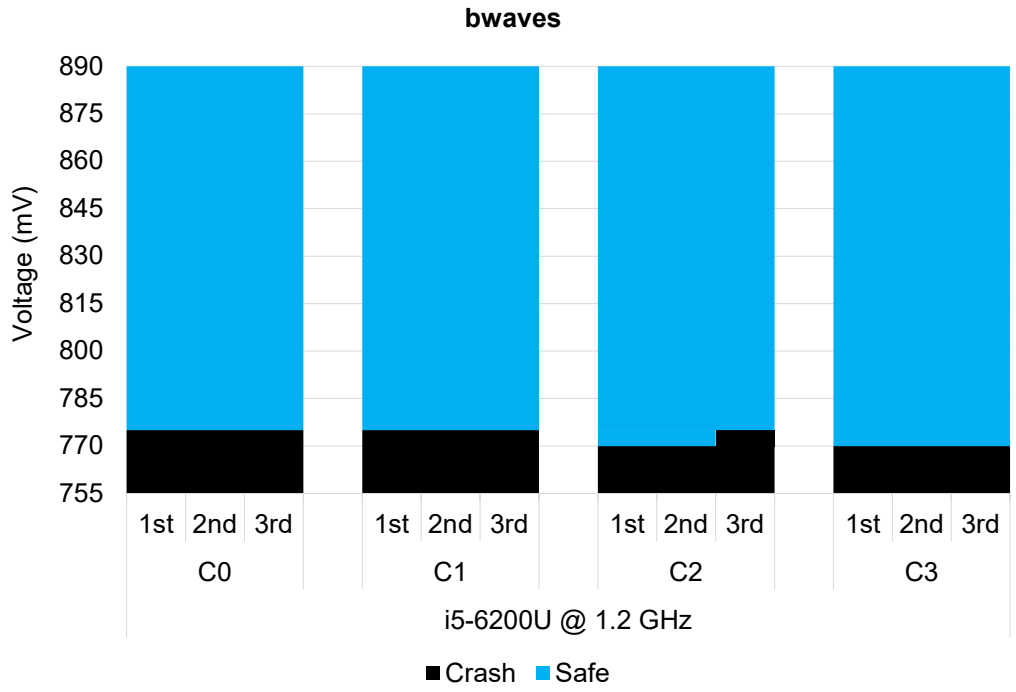


Figure 20: i5-6200U @ 1.2 GHz characterization for bwaves benchmark

4.4 Intel Core i5-6200U Skylake Microprocessor Full Speed vs Half Speed Study

In this section we compare the independent studies conducted in full and half speed modes on i5-6200U microprocessor. We remind that in our study a System Crash is of higher importance than a Corrected Error. This led in diagrams for which CE errors are not shown. Therefore, in Table 13, we display all CE errors for both setups independently.

Table 13: Cache hierarchy errors (corrected hardware errors) at full and half speed of i5-6200U

Frequency	Affinity	Benchmark	Execution Cycle	Voltage Offset (mV)
i5-6200U @ 2.3 GHz	Core 2	namd	3rd	120
	Core 2	gobmk	2nd	115
	Core 2	zeusmp	2nd	125
Frequency	**Affinity**	**Benchmark**	**Execution Cycle**	**Voltage Offset (mV)**
i5-6200U @ 1.2 GHz	All Cores no HT	bzip2	1st	110
	All Cores	milc	1st	110
	Core 2	hmmer	1st	115
	Core 3	gobmk	3rd	115
	Core 0	dealII	3rd	120
	All Cores no HT	zeusmp	1st	115

Below, in Table 14 and on page 61 in Table 15, we summarize the lowest safe voltage offset for the two studies in every affinity and benchmark combination.

Table 14: Lowest safe voltage offsets (mV) of i5-6200U @ 2.3 GHz

	i5-6200U Full Speed Affinity Configuration						
Benchmark	**Core 0**	**Core 1**	**Core 2**	**Core 3**	**All Cores**	**All Cores no HT**	**MIN**
bzip2	110	110	115	110	105	105	105
mcf	110	110	115	115	110	105	105
milc	110	110	110	115	105	100	100
namd	110	110	115	110	100	105	100
hmmer	110	110	115	115	105	105	105
h264ref	105	110	110	110	105	105	105
gobmk	110	105	110	110	105	105	105
dealII	105	105	110	110	100	105	100
zeusmp	100	105	115	115	105	105	100
bwaves	105	110	110	110	100	105	100
			Total minimum voltage offset				100

It comes that in both cases the lowest safe voltage offset is 100 mV. This means that the system can perform without any abnormal behavior at V_{S_min} = 0.790 V (coarse-grained estimation) instead of the nominal V_{S_nom} = 0.890 V. This, ensures a less power consuming operation of the system as we will demonstrate in section 4.7.

Table 15: Lowest safe voltage offsets (mV) of i5-6200U @ 1.2 GHz

	i5-6200U Half Speed Affinity Configuration						
Benchmark	**Core 0**	**Core 1**	**Core 2**	**Core 3**	**All Cores**	**All Cores no HT**	**MIN**
bzip2	115	105	115	110	110	110	105
mcf	110	110	115	120	110	110	110
milc	105	105	105	115	105	100	100
namd	110	105	110	115	110	105	105
hmmer	110	115	115	115	105	110	105
h264ref	115	110	105	115	110	105	105
gobmk	110	115	110	115	110	105	105
dealII	115	110	105	115	110	100	100
zeusmp	110	105	110	120	100	105	100
bwaves	110	110	110	115	105	100	100
				Total minimum voltage offset			100

Finally, the timing differences in benchmark executions at nominal voltage in all configurations for both setups, are shown in Table 16 and Table 17. The benchmark with the fastest execution both for 2.3 GHz and 1.2 GHz was *mcf* except the occasions of Core 1 and All Cores respectively, for which *bwaves* was the fastest one. Concerning the slowest benchmark, *dealII* had the worst execution times for both frequencies.

Table 16: Benchmark execution timings (s) of i5-6200U @ 2.3 GHz

					i5-6200U Full Speed Affinity Configuration	
Benchmark	**Core 0**	**Core 1**	**Core 2**	**Core 3**	**All Cores**	**All Cores no HT**
bzip2	32.2	32.1	37.1	31.7	27.8	29.8
mcf	20.1	19.8	24.7	19.7	16.9	17.8
milc	31.5	32.1	34.4	31.8	29.0	30.0
namd	40.0	39.8	40.3	39.0	36.5	38.1
hmmer	23.2	23.0	20.8	23.9	21.2	21.4
h264ref	43.0	42.2	46.1	42.4	40.6	41.8
gobmk	52.3	51.9	48.2	51.2	46.7	49.2
dealII	57.2	60.2	52.7	56.7	50.7	53.6
zeusmp	32.7	33.0	25.3	32.8	21.2	22.9
bwaves	31.4	18.8	26.9	22.0	22.7	23.7

Table 17: Benchmark execution timings (s) of i5-6200U @ 1.2 GHz

	i5-6200U Half Speed Affinity Configuration					
Benchmark	**Core 0**	**Core 1**	**Core 2**	**Core 3**	**All Cores**	**All Cores no HT**
bzip2	72.2	59.6	61.7	77.4	48.8	49.8
mcf	45.9	38.9	45.6	51.8	33.8	34.1
milc	57.1	46.0	56.0	63.6	46.8	46.6
namd	76.5	64.8	74.6	81.7	66.4	67.4
hmmer	54.0	55.1	52.4	60.0	42.2	42.8
h264ref	82.4	80.0	78.3	88.3	69.7	76.2
gobmk	103.8	84.9	128.9	109.3	80.0	88.7
dealII	115.1	103.8	143.7	129.2	88.6	97.0
zeusmp	55.0	52.8	65.3	64.4	37.5	55.7
bwaves	50.1	45.1	48.8	55.0	27.0	41.9

We continue the comparison in terms of *core-to-core and benchmark-to-benchmark* variations (fine-grained estimations). In all representations that follow on, the lowest average safe voltages (V_{safe}) correspond to the last safe voltage offset before any crash occurs, while the lowest average crash voltages (V_{crash}) correspond to the first voltage offset at which we documented three crashes. In the case of V_{crash}, this procedure gives slightly higher crash tolerance results. The difference is that, if we calculate the average crash voltage at the voltage offset where the system crashed for the first time, we might get stricter boundaries between safe and crash regions. This happens due to the absence of unsafe regions, as we have already explained in the beginning of section 4.2. In order to make the differences in the values of the variables more distinct, we chose a more qualitative representation, by calculating the lowest average V_{crash} at the three system crashes voltage offset of every affinity configuration.

4.4.1 Core-to-Core Variation

On the next page in Figure 21 a *core-to-core* variation of i5-6200U characterization studies at 2.3 GHz and 1.2 GHz frequencies is demonstrated.

The analysis showed that for i5-6200U at 2.3 GHz, the lowest average V_{Sf_safe} among all benchmarks is 777.5 V observed in *mcf* and *hmmer* and the lowest average V_{Sf_crash} is 767.5 V observed in *milc* and *zeusmp*. For the i5-6200U at 1.2 GHz, the lowest average V_{Sh_safe} = 776.3 V and the lowest average V_{Sh_crash} is 768.8 V in *mcf* and *h264ref* benchmarks respectively.

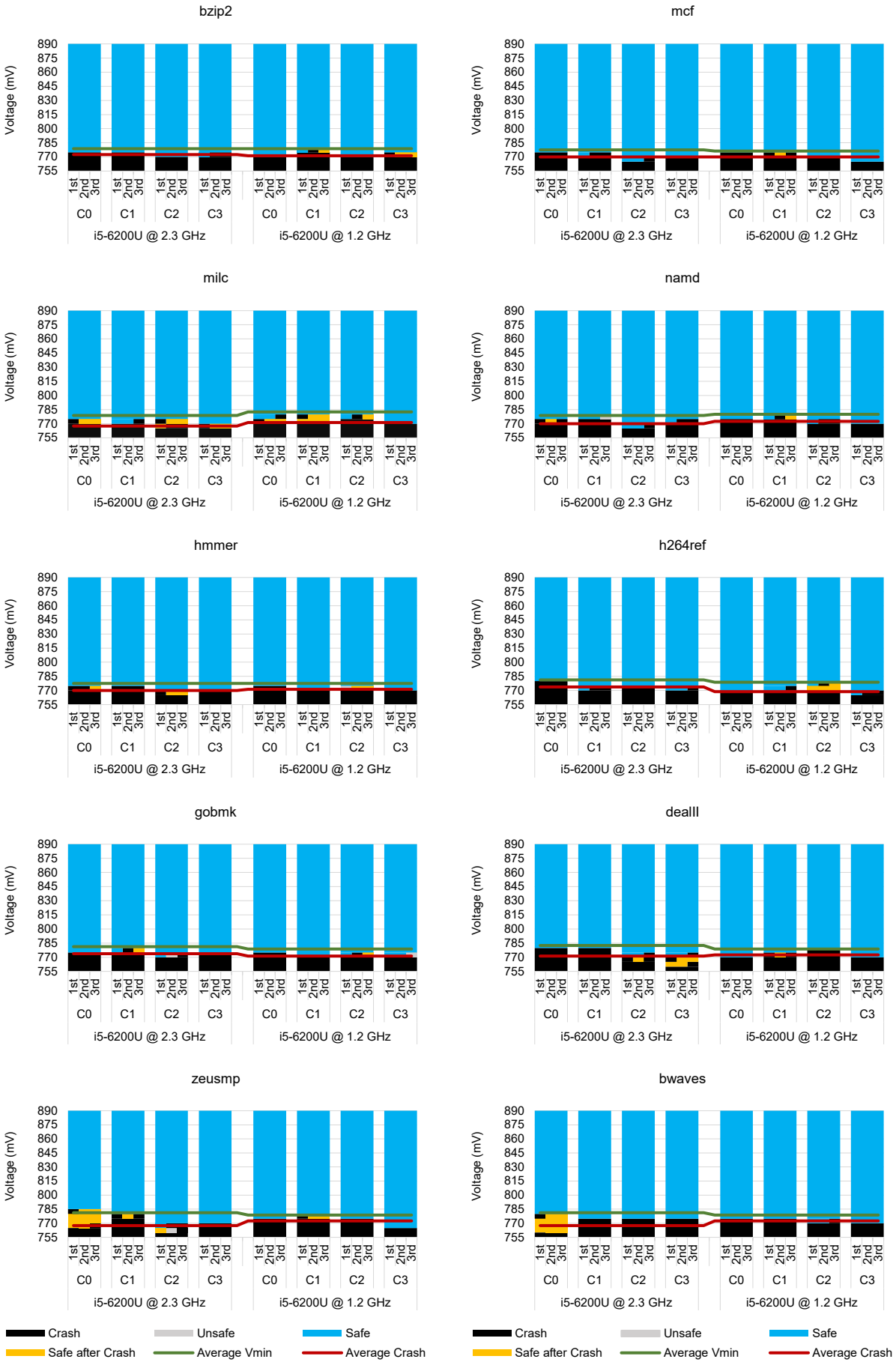


Figure 21: Skylake characterization results for 10 SPEC CPU2006 benchmarks of the i5-6200U chip at full and half speed modes in core-to-core variation

4.4.2 Benchmark-to-Benchmark Variation

In the same sense as in section 4.4.1, on page 65 (Figure 22) and on page 66 (Figure 23), a *benchmark-to-benchmark* variation of i5-6200U characterization studies at 2.3 GHz and 1.2 GHz frequencies is demonstrated, regarding the lowest average V_{safe} and lowest average V_{crash} values.

The analysis showed that for i5-6200U at 2.3 GHz, the lowest average V_{Sf_safe} among all cores is 777.5 V and the lowest average V_{Sf_crash} is 767.5 V both observed in Core 2. For the i5-6200U at 1.2 GHz, the lowest average V_{Sh_safe} = 775.0 V and the lowest average V_{Sh_crash} is 768.5 V both observed in Core 3.

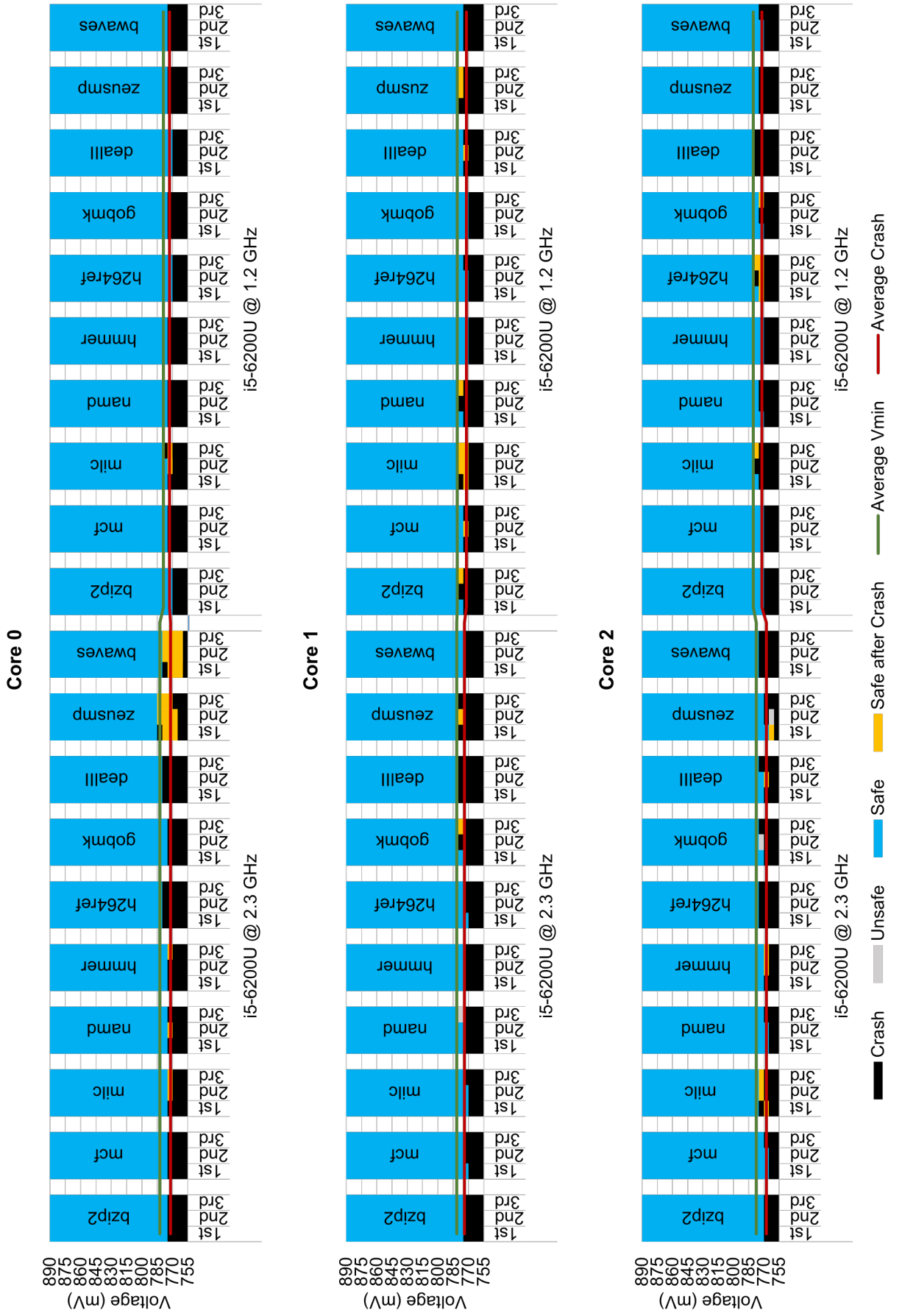


Figure 22: Skylake characterization results for 10 SPEC CPU2006 benchmarks on i5-6200U chip at full and half speed modes in benchmark-to-benchmark variation (Core 0, Core 1, Core 2)

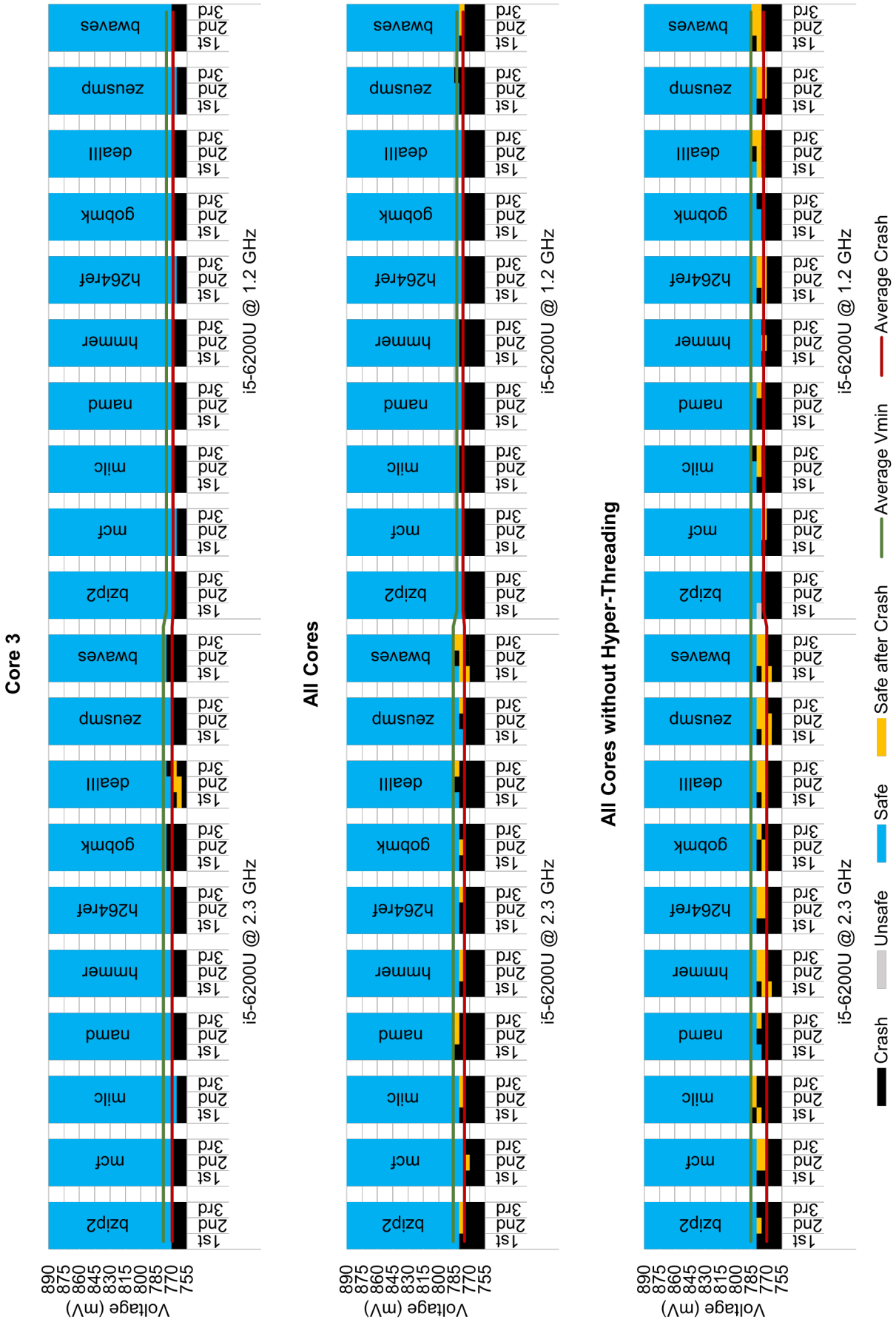


Figure 23: Skylake characterization results for 10 SPEC CPU2006 benchmarks on i5-6200U chip at full and half speed modes in benchmark-to-benchmark variation (Core 3, All Cores)

4.5 Intel Core i5-6200U Skylake vs Intel Core i5-4200U Haswell Microprocessors Study

In this section we present the independent studies conducted on i5-6200U and i5-4200U microprocessors in *core-to-core* and *benchmark-to-benchmark* variations (fine-grained estimations). In both variations it is illustrated the difference in unsafe regions. In the case of i5-4200U, corrected hardware errors were found in almost any affinity case and thus the unsafe regions are more apparent. Since that study didn't include *dealII* and *bwaves* benchmarks, we excluded them for both variations from i5-6200U too. Another point we must clarify is that the voltages of the two microprocessors are not defined on the same scale. The characterization voltage range of i5-6200U is between 0.890 V and 0.755 V while for i5-4200U it is between 0.844 V and 0.744 V. Thus, the y-axis on Figure 24-Figure 26, represents the voltage reduction from the nominal voltage of each microprocessor.

Below, in Table 18 we summarize the lowest safe voltage offset for the i5-4200U study in every affinity and benchmark combination as it was presented in [16]. There were no data reported for the *mcf* benchmark in All Cores affinity because there was not enough RAM. In the case of i5-42000 the lowest safe operation voltage offset was found to be 80 mV. We remind that the nominal voltage of i5-4200U system was defined at V_{H_nom} = 0.844 V. This means that the system can perform without any abnormal behavior up to V_{H_min} = 0.764 V (coarse-grained estimation).

Table 18: Lowest safe voltage offsets (mV) of i5-4200U @ 2.6 GHz

	i5-4200U Affinity Configuration						
Benchmark	**Core 0**	**Core 1**	**Core 2**	**Core 3**	**All Cores**	**All Cores no HT**	**MIN**
bzip2	85	85	85	85	85	80	80
mcf	85	90	85	85	-	85	85
milc	85	85	85	85	80	80	80
namd	90	90	85	90	90	85	85
hmmer	90	90	90	90	85	80	80
h264ref	90	85	90	85	85	85	85
gobmk	85	85	90	90	85	85	85
zeusmp	85	90	85	90	85	85	85
				Total minimum voltage offset			80

In all *core-to-core and benchmark-to-benchmark* variations that follow on, the calculation of the lowest average V_{crash} in the case of i5-6200 is done as explained in the last paragraph of section 4.4, while for i5-4200 is calculated at the first crash voltage offset of every affinity configuration since there wasn't followed such procedure in that study.

4.5.1 Core-to-Core Variation

On the next page in Figure 24, a *core-to-core* variation between i5-6200U and i5-4200U characterization studies is demonstrated, regarding the lowest average V_{safe} and lowest average V_{crash} values. The comparison shows that, for i5-6200U the lowest average V_{S_safe} among all benchmarks occurs for 12.6% voltage reduction at 777.5 V observed in *mcf* and *hmmer* and the lowest average V_{S_crash} occurs for 13.8% voltage reduction at

767.5 V observed in *milc* and *zeusmp*. For the i5-4200U, the lowest average V_{H_safe} occurs for 9.9% voltage reduction at 760.3 V (*hmmer*, *zeusmp*) and the lowest average V_{H_crash} occurs for 11.3% voltage reduction at 749.0 V (*hmmer*).

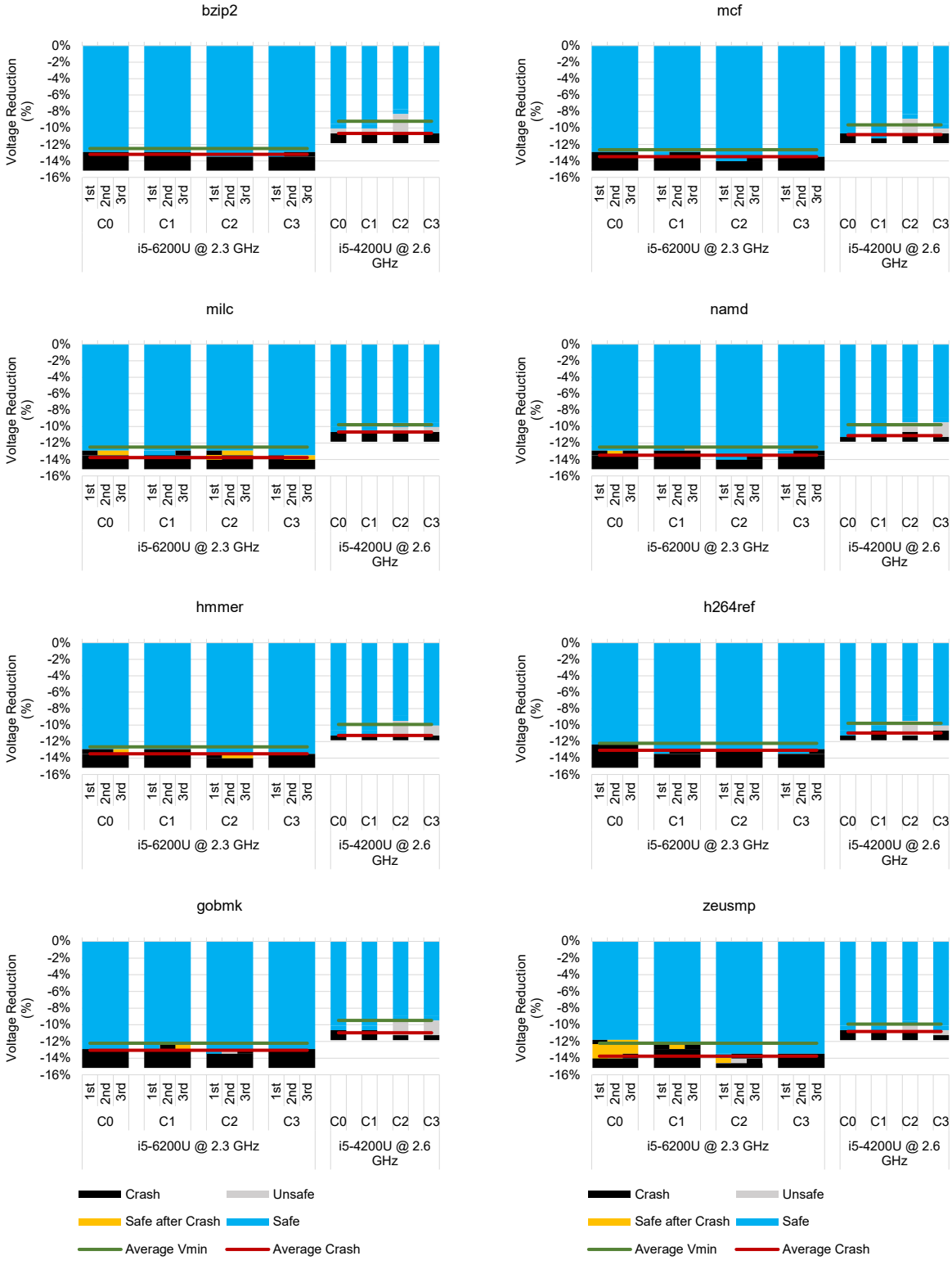


Figure 24: Skylake and Haswell characterization results for 8 SPEC CPU2006 benchmarks on i5-6200U and i5-4200U chips in core-to-core variation

4.5.2 Benchmark-to-Benchmark Variation

The *benchmark-to-benchmark* variation between i5-6200U and i5-4200U is depicted in Figure 25 and Figure 26. The analysis showed that for i5-6200U, the lowest average V_{S_safe} among all cores occurs for 12.3% voltage reduction at 776.9 V and the lowest average V_{S_crash} occurs for 13.8% voltage reduction at 766.9 V both observed in Core 2. For the i5-4200U, the lowest average V_{H_safe} = 757.8 V (10.2% of voltage reduction) observed both in Core 0 and Core 1 while the lowest average V_{H_crash} occurs for 10.9% voltage reduction at 751.5 V observed in Core 3.

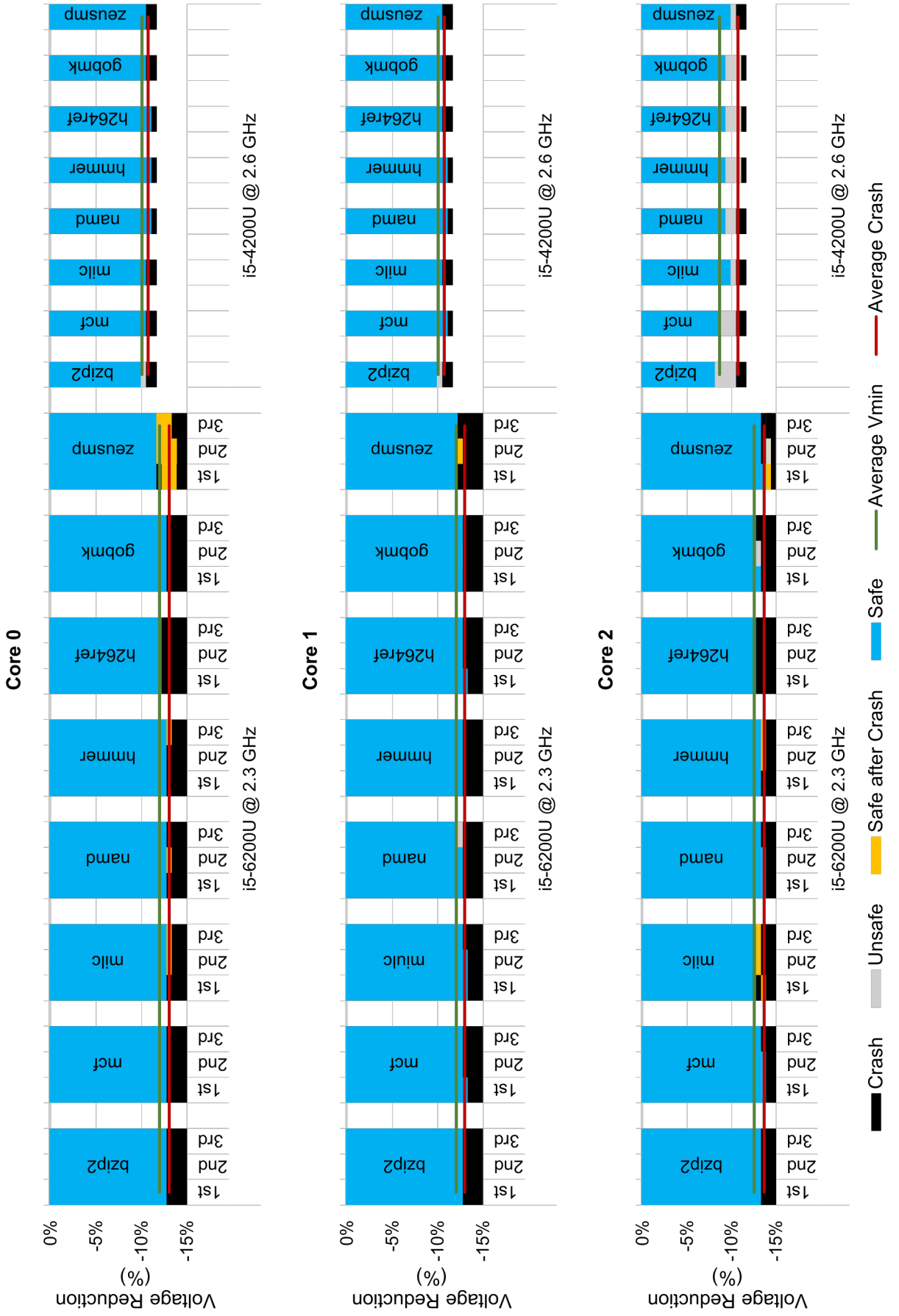


Figure 25: Skylake and Haswell characterization results for 8 SPEC CPU2006 benchmarks on i5-6200U and i5-4200U chips in benchmark-to-benchmark variation (Core 0, Core 1, Core 2)

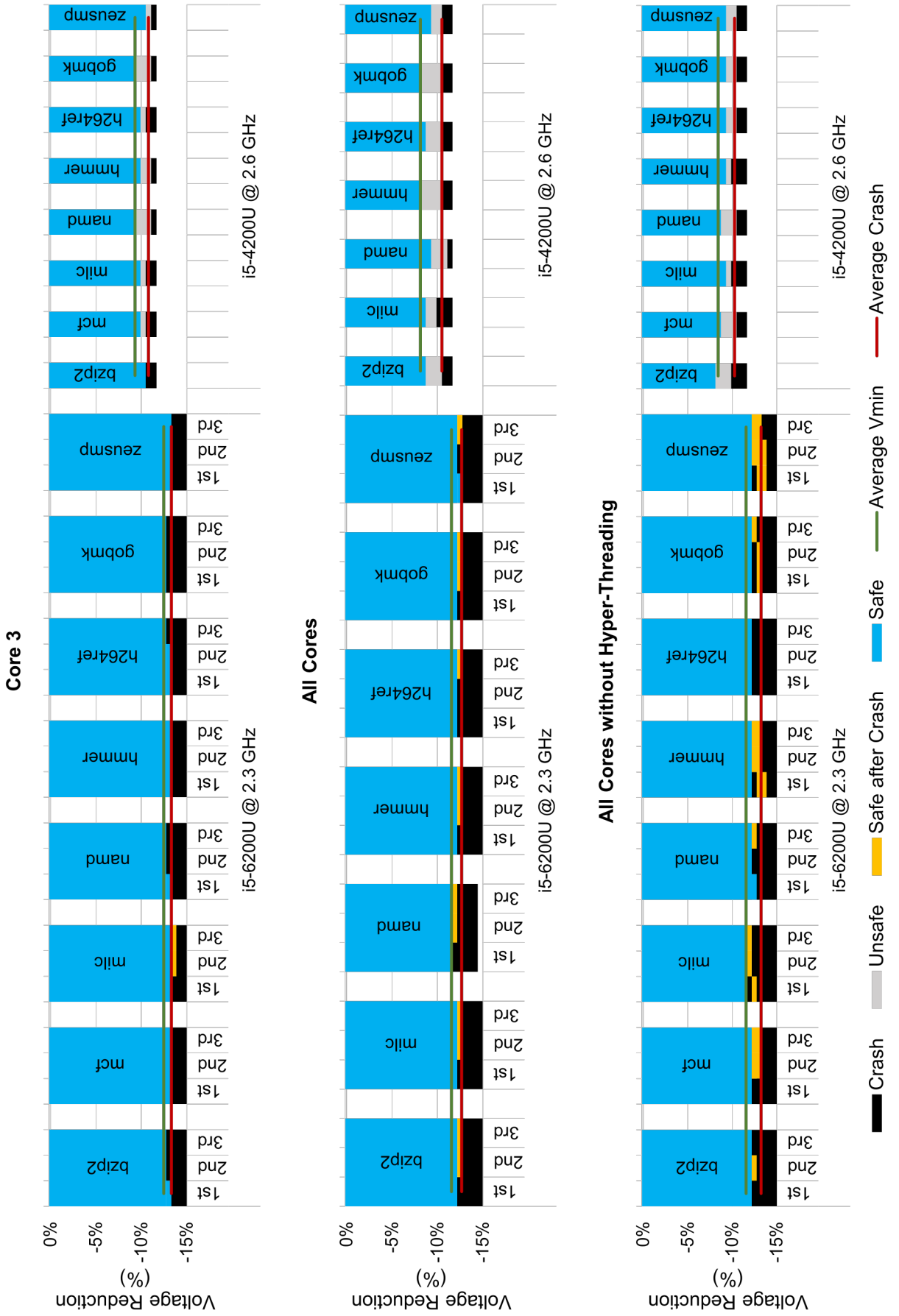


Figure 26: Skylake and Haswell characterization results for 8 SPEC CPU2006 benchmarks on i5-6200U and i5-4200U chips in benchmark-to-benchmark variation (Core 3, All Cores)

4.6 Voltage Reduction and Core Resilience Metrics

In this section we analyze the results, voltage reduction wise and we introduce the core *resilience* metric. We define as *resilience* of a Core, the sum of benchmarks in which a Core managed to withstand the largest voltage reduction before it crashed, compared to the other Cores of the same CPU. The evaluation is done in *benchmark-to-benchmark* and *core-to-core* variations.

Examining the *benchmark-to-benchmark* representation of voltage reduction results (depicted on Figure 21 on the following page) we devised Table 19, in which i5-6200U at 1.2 GHz achieved the maximum voltage reduction in-between the most affinity configurations. This means that as the CPU is getting undervolted, we can get an even more stable and low-power operation at the cost of CPU speed performance. The voltage-margin improvement of Skylake against Haswell is obvious in this case too. However, this is a 1 versus 1, direct comparison and should not be confused with the overall safe voltage reduction, which was at the same levels for both Skylake's configurations, as we already saw.

Table 19: CPU results based on maximum voltage reduction for every affinity configuration in benchmark-to-benchmark variation

Affinity Configuration	i5-6200U @ 2.3 GHz	i5-6200U @ 1.2 GHz	i5-4200U @ 2.6 GHz
Core 0	5	**9**	0
Core 1	**7**	**7**	0
Core 2	**10**	5	0
Core 3	3	**10**	0
All Cores	4	**9**	0
All Cores no HT	7	**8**	0
Score	**2**	**5**	**-**

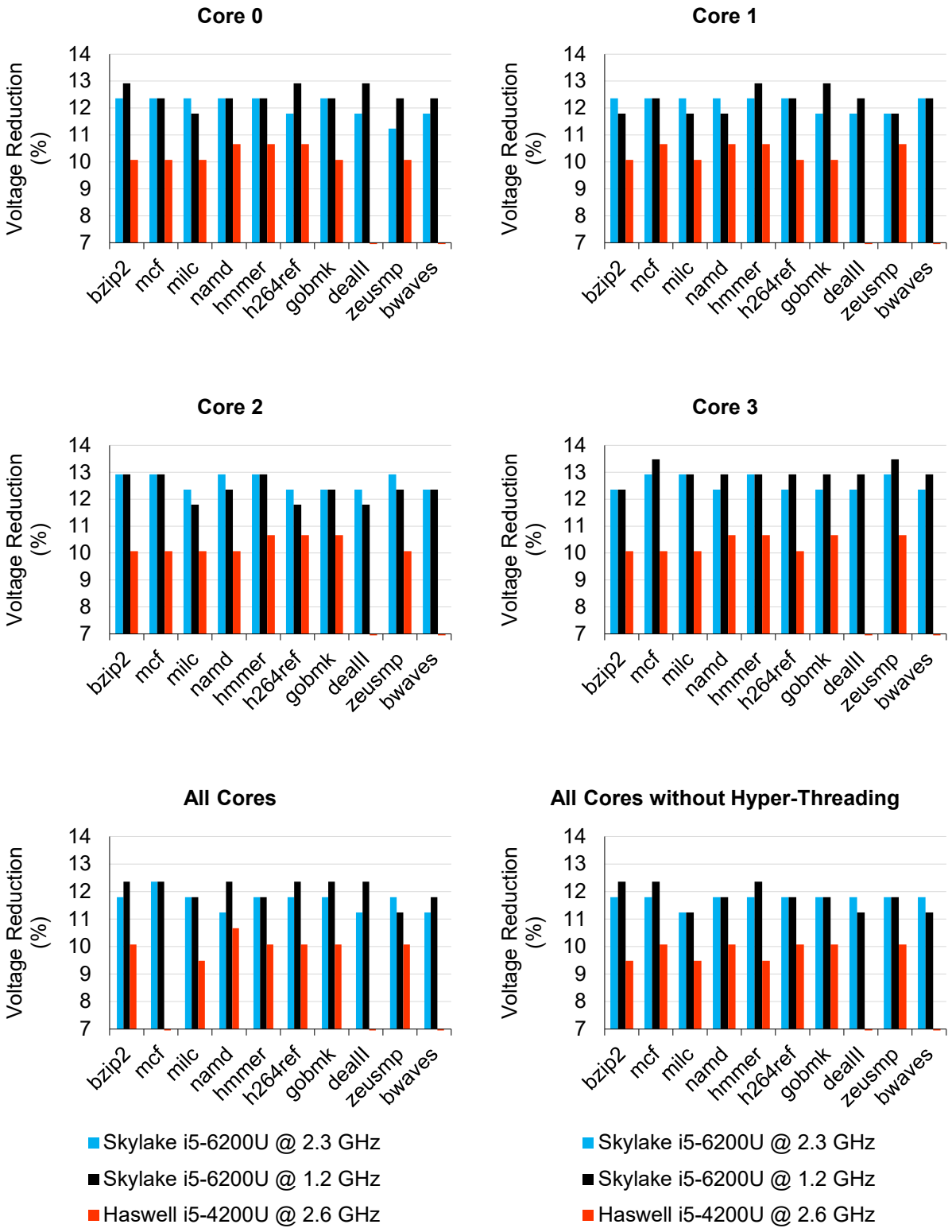


Figure 27: Voltage reduction percentage for 10 SPEC CPU2006 benchmarks among Skylake and Haswell chips in benchmark-to-benchmark variation

In Table 20, Table 21 and Table 22, the variable *X* indicates the Core with the highest voltage reduction while the variable *M* indicates that the Core has an intermediate voltage reduction between the highest and the lowest extremes that were observed for the other Cores. The empty cells indicate the Cores that had the lowest voltage reduction for each benchmark. The resilience mapping for all three configuration setups, derived from the diagrams in Figure 28 and Figure 29 (*core-to-core* variations) on the following pages.

In the i5-6200U at full speed trials, Core 2 was the most resilient followed closely behind by Core 3 (9 versus 8 respectively). Conversely, in the i5-6200U at half speed trials, Core 3 outperformed Core 2 and the other Cores. For this reason, we identify Core 3 as the Core that manages to be the most resilient during the benchmarks undervolting runs on i5-6200U.

On the other hand, the resilience metric is more balanced in the case of i5-4200U CPU. Thus, there is no clear winner and we come to conclusion that Haswell cores can be equally resilient in most benchmarks undervolting runs independently.

Table 20: Resilience mapping of i5-6200U @ 2.3 GHz

	i5-6200U Full Speed Resilience			
Benchmark	**Core 0**	**Core 1**	**Core 2**	**Core 3**
bzip2			X	
mcf			X	X
milc				X
namd			X	
hmmer			X	X
h264ref		X	X	X
gobmk	X		X	X
dealII			X	X
zeusmp		M	X	X
bwaves		X	X	X
Score	**1**	**2**	**9**	**8**

Table 21: Resilience mapping of i5-6200U @ 1.2 GHz

	i5-6200U Half Speed Resilience			
Benchmark	**Core 0**	**Core 1**	**Core 2**	**Core 3**
bzip2	X		X	M
mcf			M	X
milc				X
namd	M		M	X
hmmer		X	X	X
h264ref	X	M		X
gobmk		X		X
dealII	X	M		X
zeusmp	M		M	X
bwaves				X
Score	**3**	**2**	**2**	**9**

Table 22: Resilience mapping of i5-4200U @ 2.6 GHz

	i5-4200U Resilience			
Benchmark	**Core 0**	**Core 1**	**Core 2**	**Core 3**
bzip2	X	X	X	X
mcf		X		
milc	X	X	X	X
namd	X	X		X
hmmer	X	X	X	X
h264ref	X		X	
gobmk			X	X
zeusmp		X		X
Score	**5**	**6**	**5**	**6**

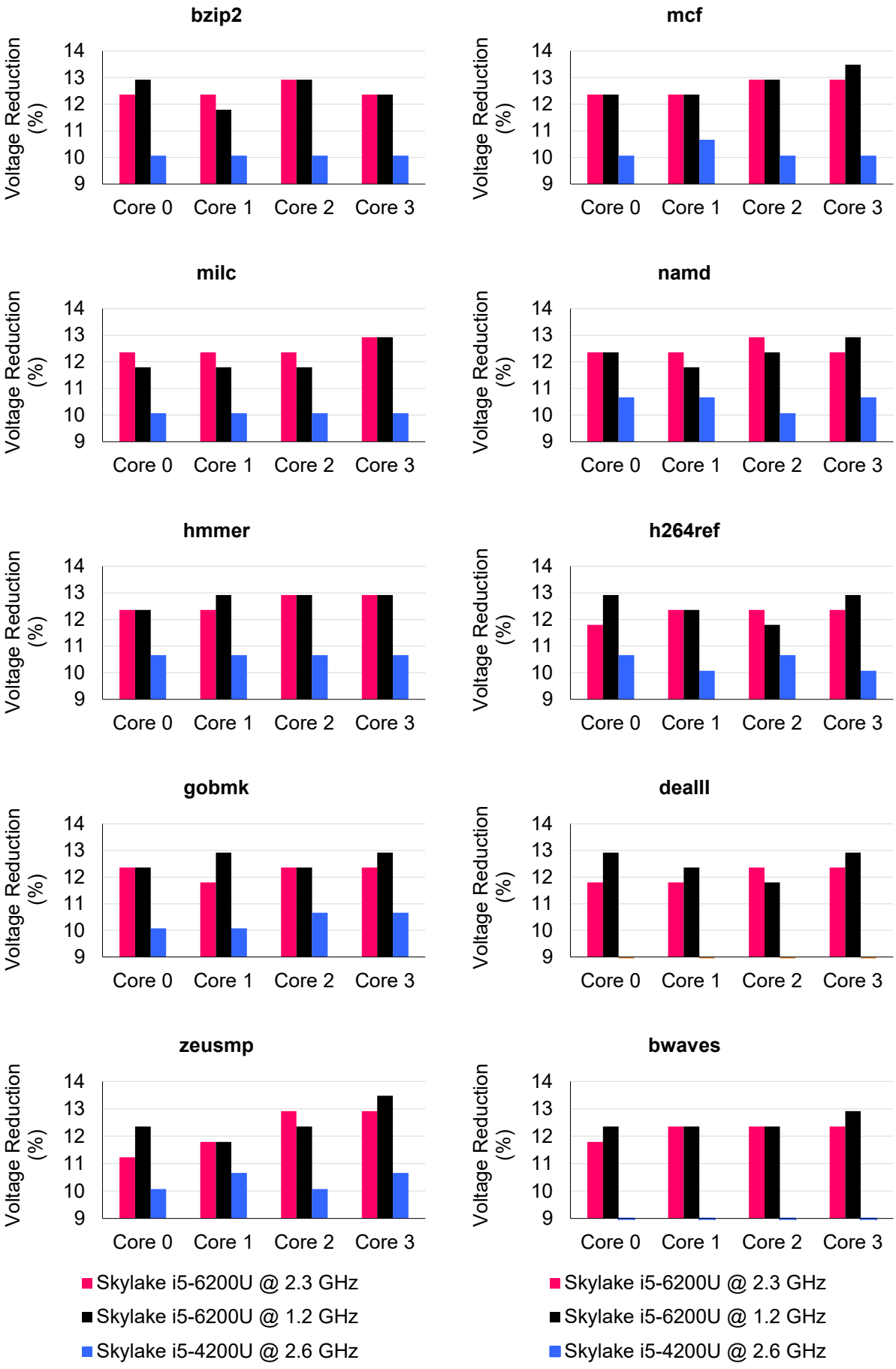


Figure 28: Voltage reduction percentage for 10 SPEC CPU2006 benchmarks among Skylake and Haswell chips in core-to-core variation

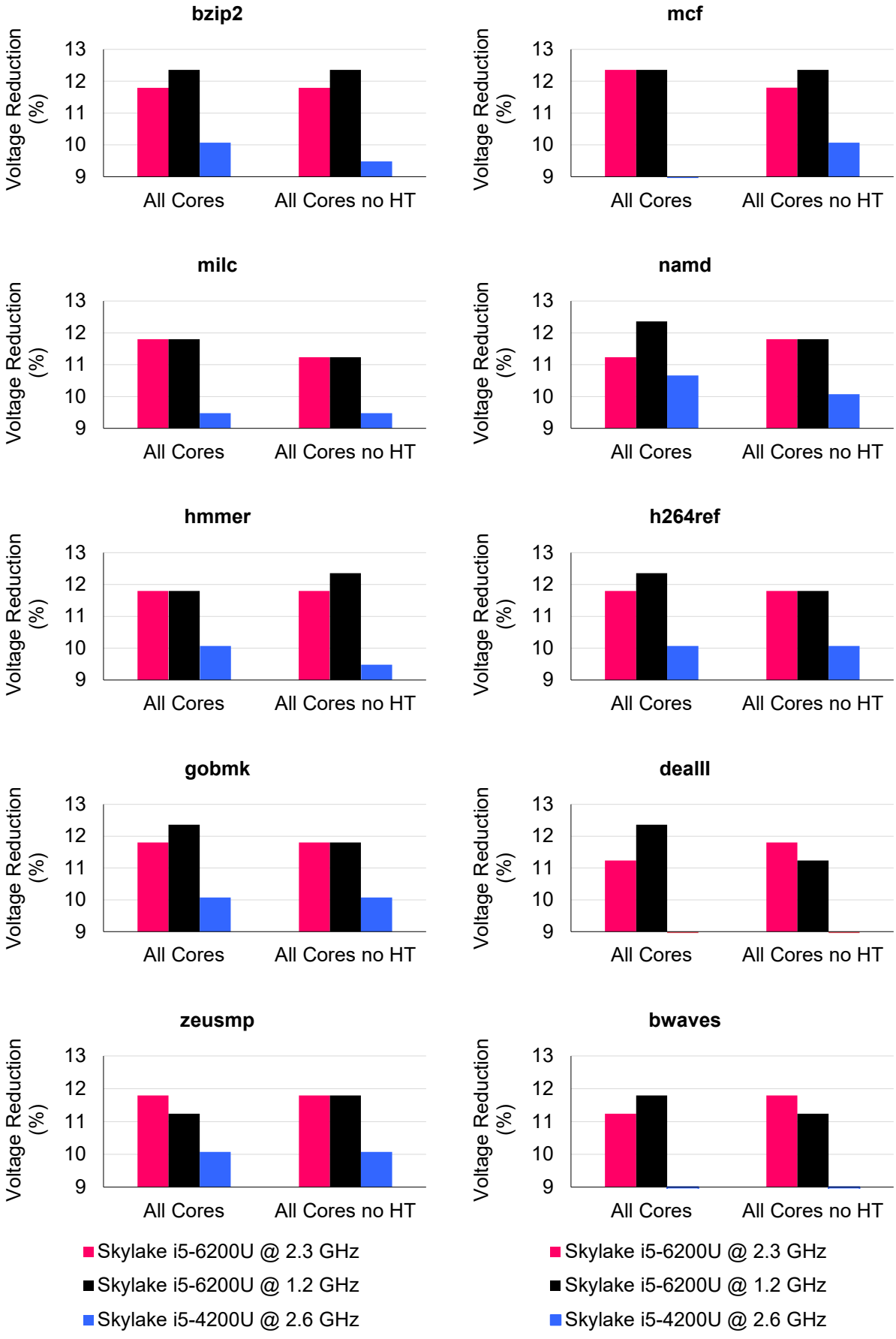


Figure 29: Voltage reduction percentage for 10 SPEC CPU2006 benchmarks among Skylake and Haswell chips in All Core and All Cores without Hyper-Threading variations

4.7 Temperature and Power Metrics

Hereby, we illustrate in an exhaustive way the package temperature and package power results we acquired from the characterization study. As shown in Image 11, we documented the highest temperature and power values independently, within a benchmark execution timeframe. This means that the highest values could have been observed at different times or even at the same time during a benchmark execution. In the first sections (4.7.1 and 4.7.2) an evaluation in terms of absolute measurement values follows both for temperature and power and in the later sections in terms of efficiency. The comparisons are performed between the Skylake and Haswell setups in *frequency-to-frequency*, *core-to-core* and *benchmark-to-benchmark* variations.

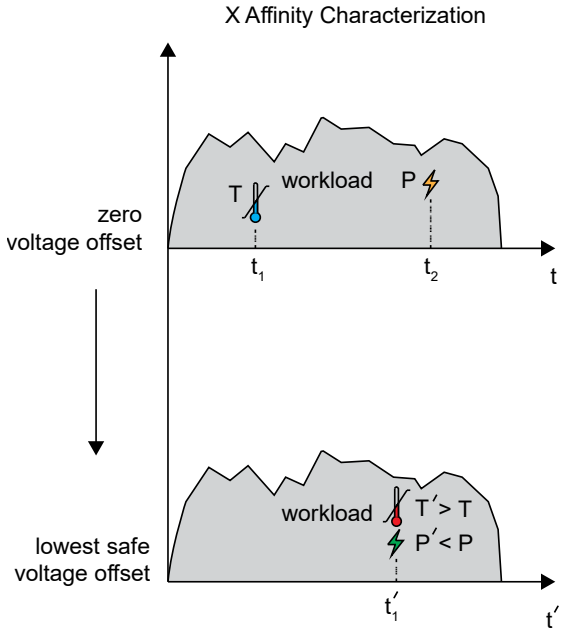


Image 11: Temperature and power occurrences within benchmark execution timeframes

4.7.1 Absolute Total Average Temperature

On the next page in Figure 30, a *frequency-to-frequency* representation of the Absolute Total Average Temperature (ATAT) is shown, derived from the characterization study of i5-6200U chip for two different frequency speeds (2.3 GHz and 1.2 GHz). We note that *frequency-to-frequency* is basically a *chip-to-chip* variation. The *ATAT* variable, is the average temperature from all voltage offsets (vertical sum) of the average absolute temperatures derived from the three benchmark executions (horizontal sum) at each voltage offset.

It is apparent, that in the half speed frequency (1.2 GHz) the temperature is lower in every core affinity for all benchmarks. In the case of 2.3 GHz the highest *ATAT* (50.8 °C) occurred for *gobmk* benchmark during Core 2 characterization while this happened for *zeusmp* benchmark again during Core 2 characterization with a temperature value of 48.6 °C. Among the 4 threads at full speed, Core 2 affinity has the hottest temperature profile (48.8 °C) for all benchmarks, while the coolest profile (48.0 °C) appears in the case of Core 3. In the case of half speed, we observe the hottest profile (44.5 °C) in Core 2 and the coolest one (42.4 °C) in Core 0.

Regarding the *core-to-core* variation (Figure 31 and Figure 32) we can see clearly that the undervolted chip is operating at cooler temperatures in half speed frequency as expected. This happens for all benchmarks and affinity configurations.

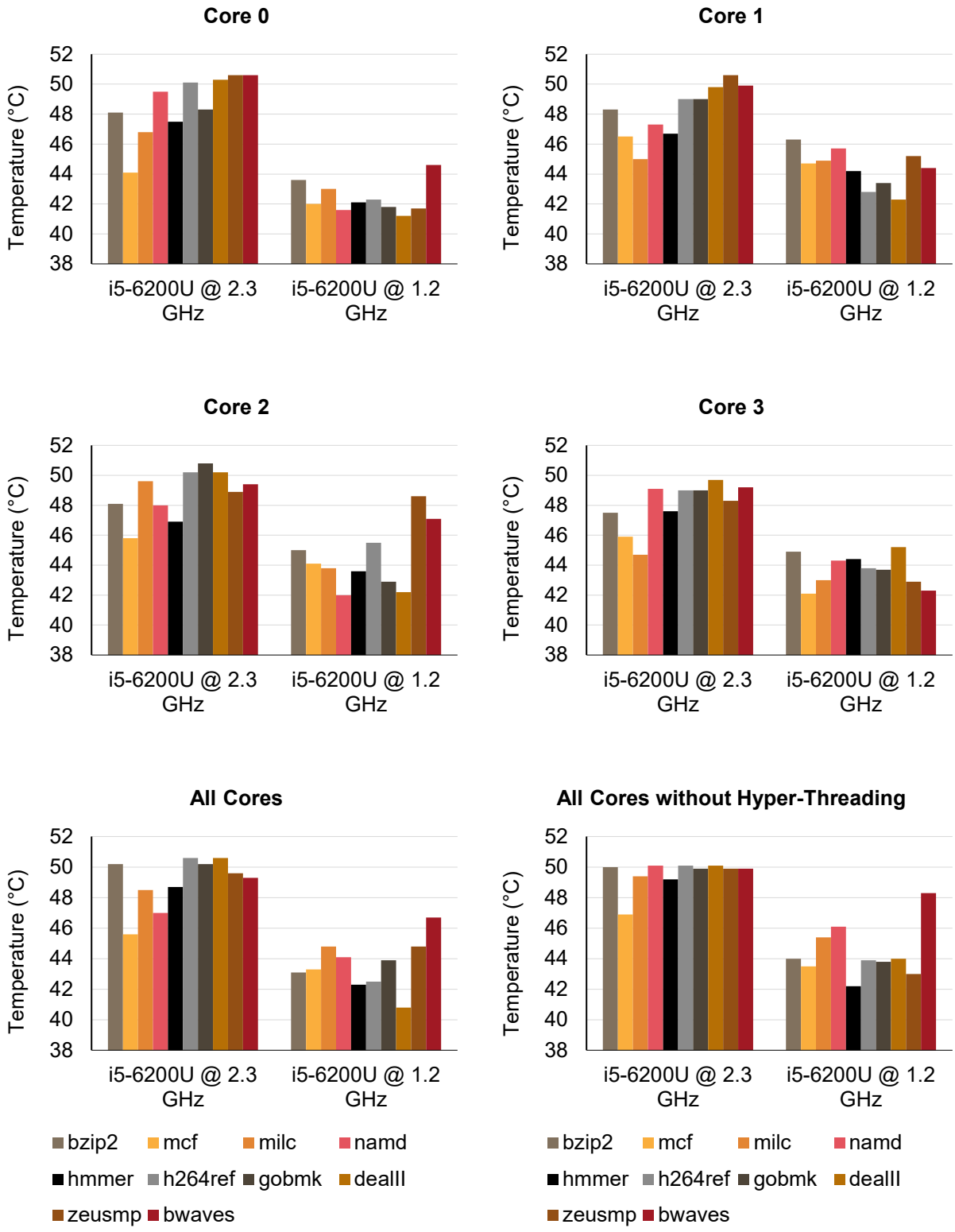


Figure 30: Absolute Total Average Temperature for 10 SPEC CPU2006 benchmarks of the Skylake chip on all affinity configurations in frequency-to-frequency variation

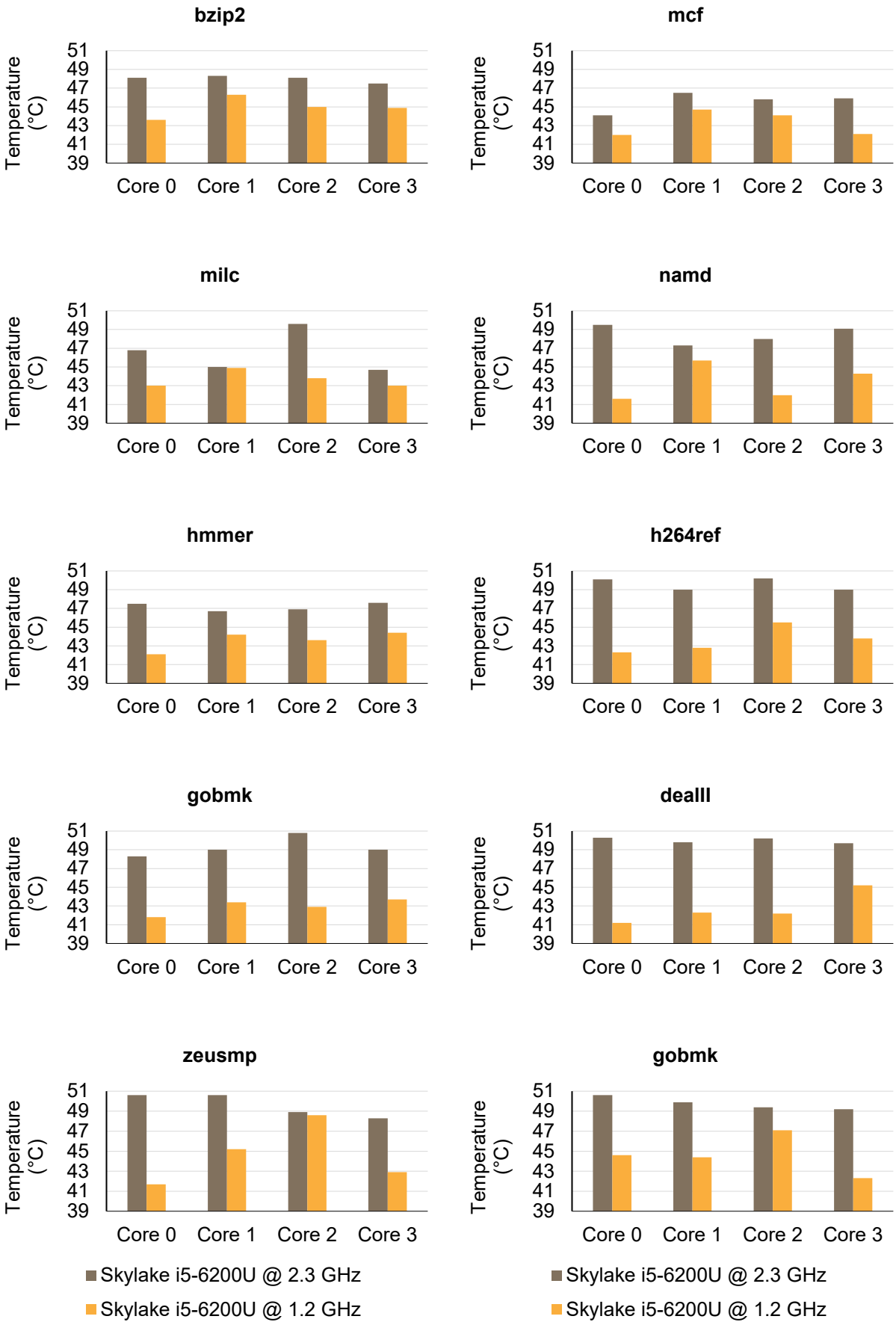


Figure 31: Absolute Total Average Temperature for 10 SPEC CPU2006 benchmarks of the Skylake chip at full and half speed modes in core-to-core variation

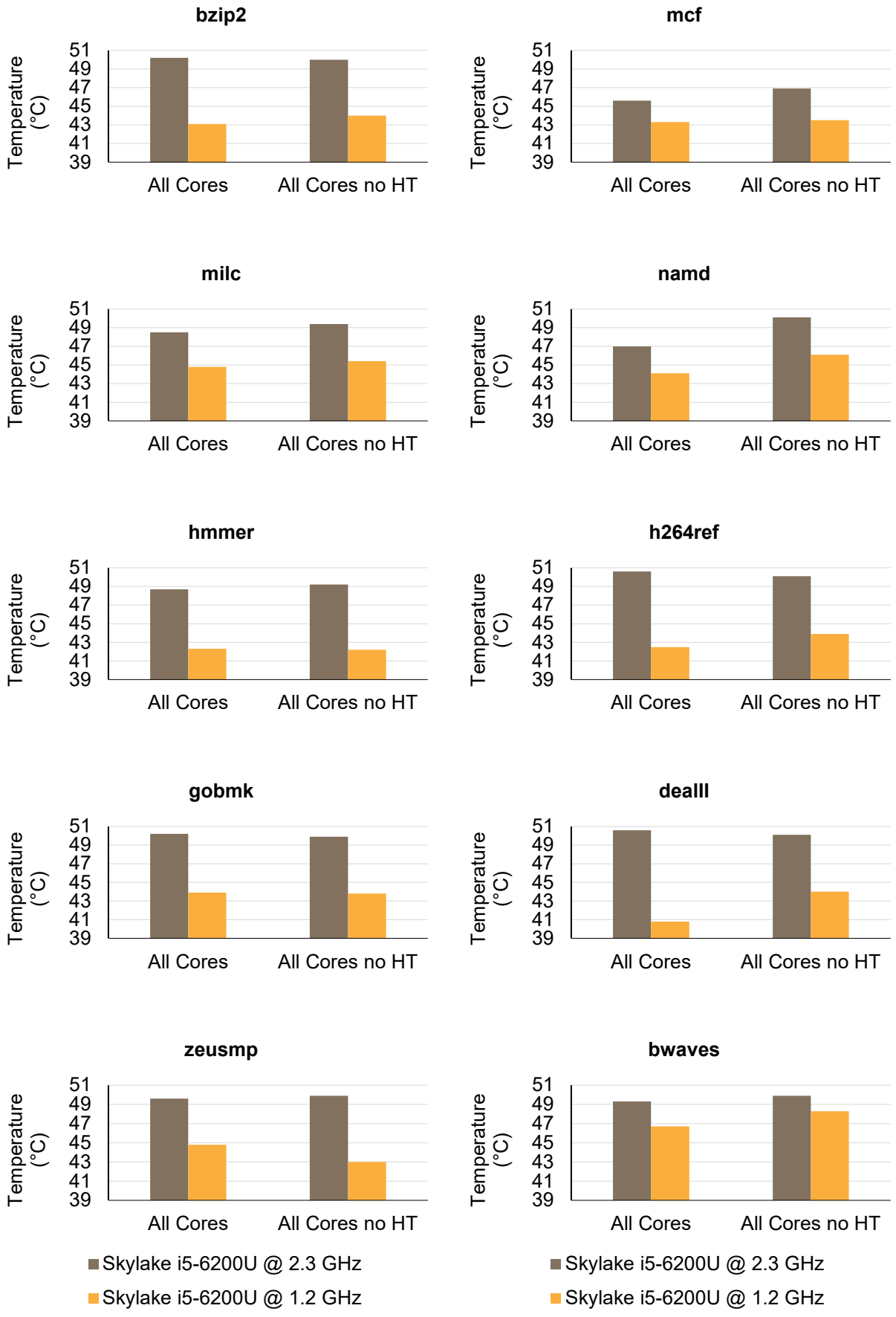


Figure 32: Absolute Total Average Temperature for 10 SPEC CPU2006 benchmarks of the Skylake chip at full and half speed modes in All Core and All Cores without Hyper-Threading variations

4.7.2 Absolute Total Average Power

In a similar manner as with Absolute Total Average Temperature (subsection 4.7.1), we define *ATAP* as the Absolute Total Average Power. For i5-6200U at 2.3 GHz, *bwaves* is the most power consuming benchmark in all affinity configurations. This is true in the case of 1.2 GHz setup too, except in Core 2 where *namd* is the most power consuming benchmark (Figure 33).

The power comparison in *core-to-core* variation is shown in Figure 34 and Figure 35.

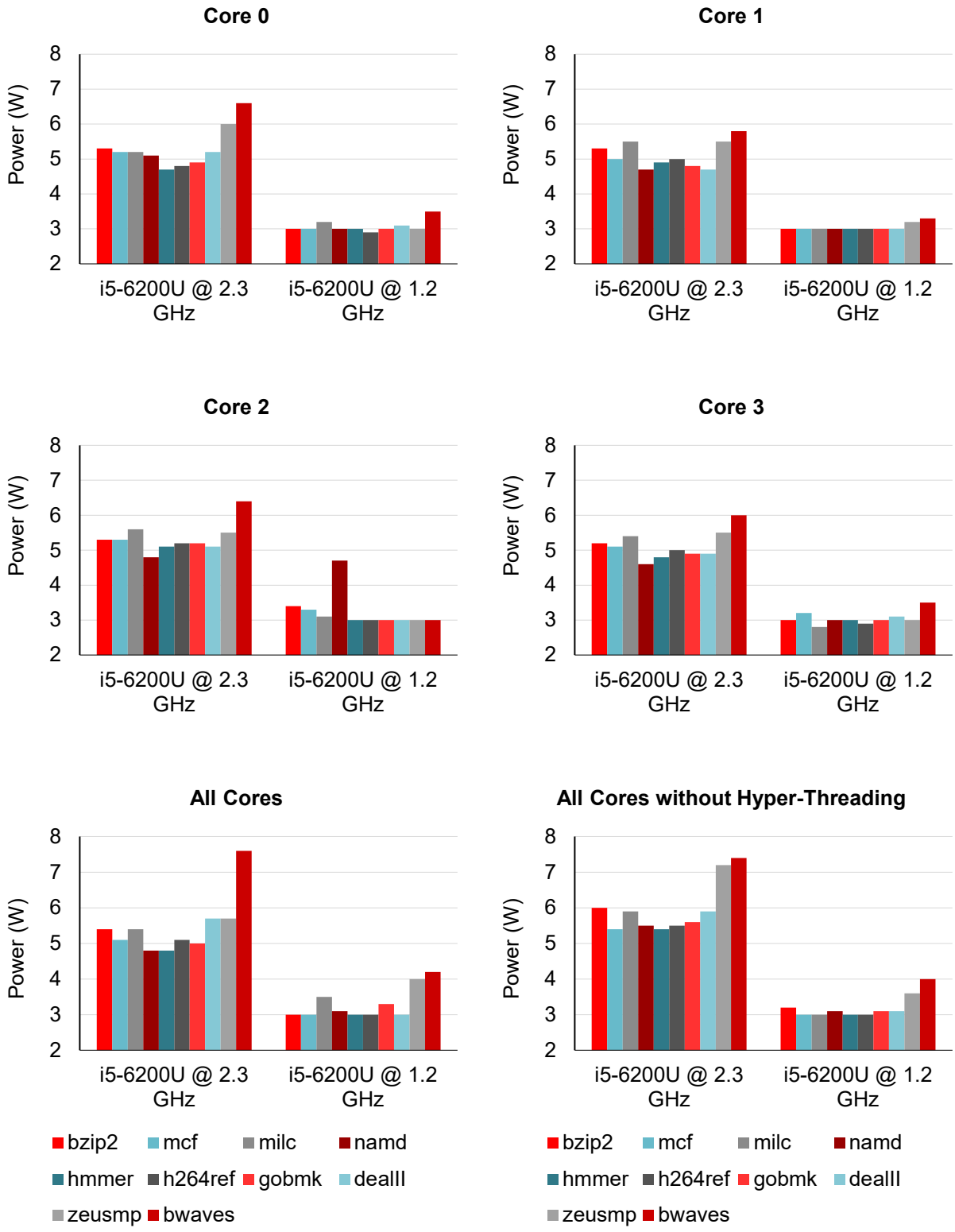


Figure 33: Absolute Total Average Power for 10 SPEC CPU2006 benchmarks of the Skylake chip on all affinity configurations in frequency-to-frequency variation

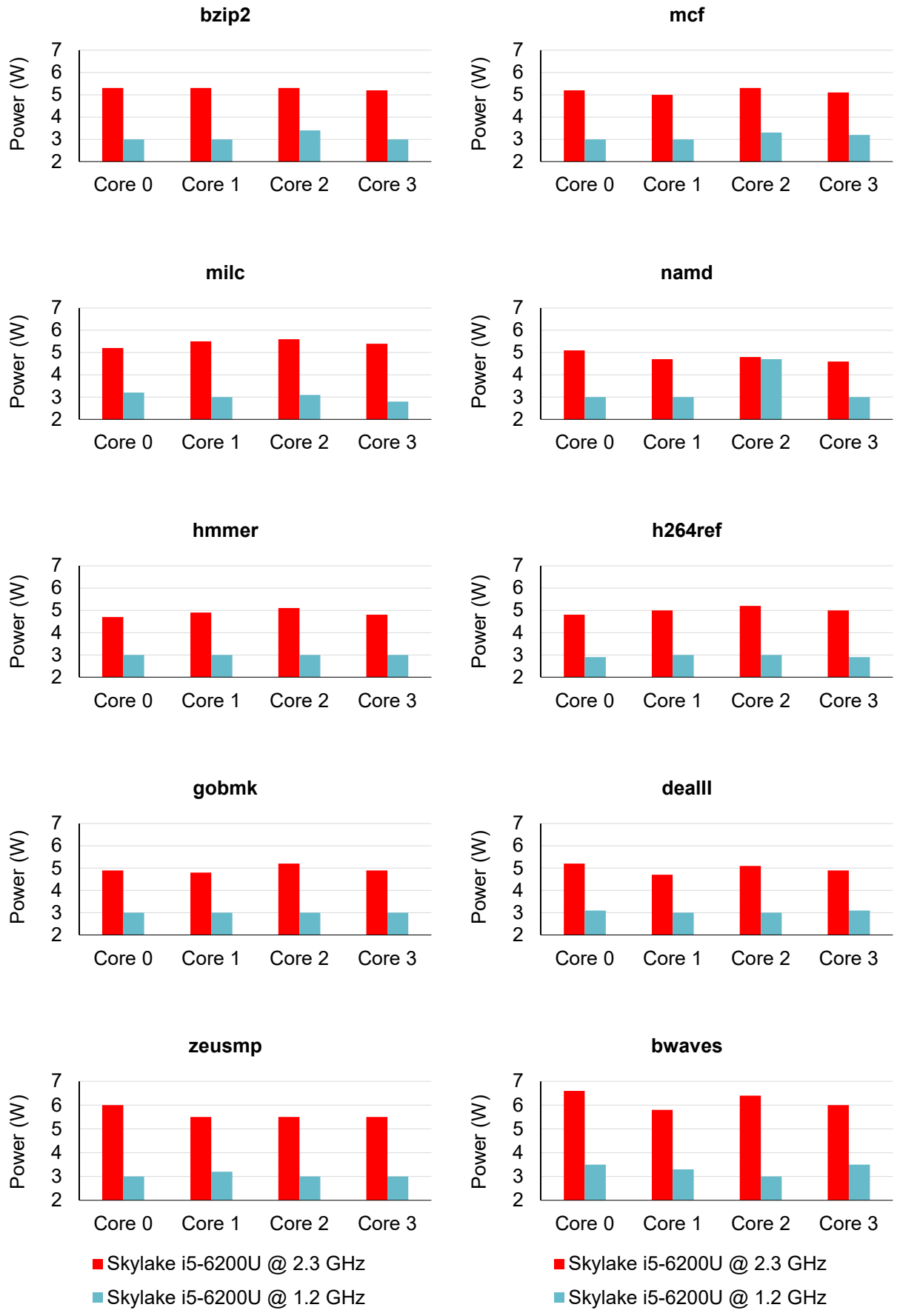


Figure 34: Absolute Total Average Power for 10 SPEC CPU2006 benchmarks of the Skylake chip at full and half speed modes in core-to-core variation

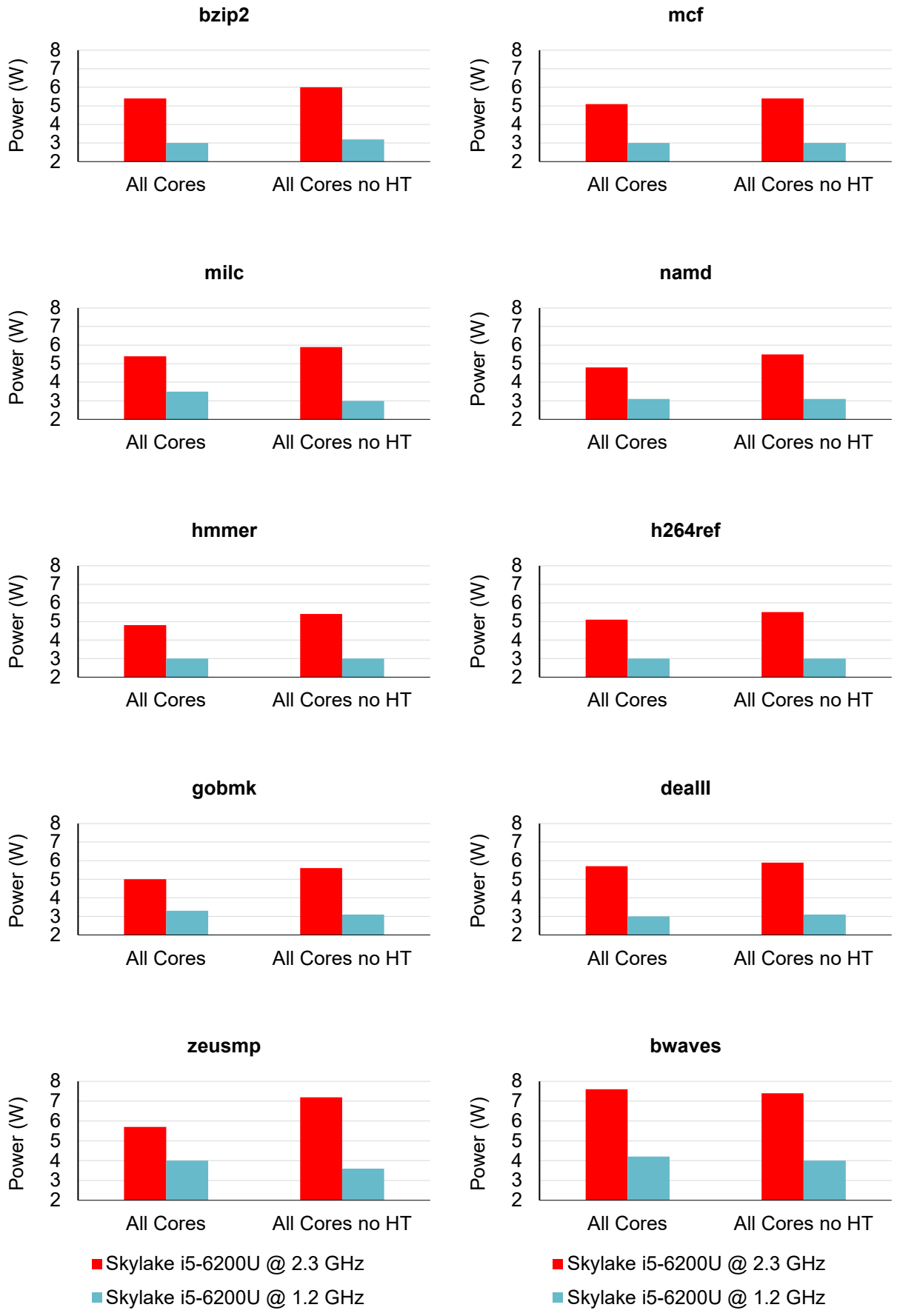


Figure 35: Absolute Total Average Power for 10 SPEC CPU2006 benchmarks of the Skylake chip at full and half speed modes in All Core and All Cores without Hyper-Threading variations

4.7.3 Temperature Efficiency

Another temperature metric we used, is temperature efficiency i.e. the percentage variation of temperature at the lowest safe voltage with reference to the temperature we measured at the nominal voltage. In Figure 36 and Figure 37 we show a comparison among Skylake and Haswell configurations for all benchmarks and affinities. We remind that *dealII* and *bwaves* benchmarks weren't included in the Haswell study. For the same chip also, there weren't any data for All Cores affinity in the case of *mcf* benchmark.

We notice that the Haswell chip achieves temperature gains in every occasion. This means that as the chip was undervolted, the measured temperature at the lowest safe voltage offset was lower than the nominal one, even by the small amount of 1%. This happened for *mcf* (Core 0/2), *milc* (Core 1) and *namd* (Core 3) benchmarks. The highest temperature gain was 8% and was observed in the cases of *namd* (Core 0), *zeusmp* (Core 3) and *bzip2* (All Cores).

On the contrary, we did not notice such behavior in Skylake. There were occasions for which the temperature efficiency was negative at the lowest safe voltage. This is depicted in diagrams with a value of -1 (red bar). Note that -1 is only a symbolism to indicate the existence of the temperature efficiency loss and not a real value. This behavior is also more apparent in All Cores and All Cores without Hyper-Threading affinities.

It has already been described in section 3.2, that every experiment is 4-dimensionally constrained by voltage offset, benchmark type, execution time and affinity configuration. In our opinion, the observation of higher temperatures than the nominal ones at the lowest safe voltage offsets, is caused by either that Enhanced Intel SpeedStep Technology was disabled, by the existence of Intel Speed Shift Technology, or both. Concerning the first speculation, we explained in section 2.4, that through EIST the operating system is responsible for the P-states alternation. Therefore, with the EIST disabled we assume that during the benchmark execution the chip operation cannot compensate the load in such efficient way and the temperature rises. This can be further backed up by the observation of increased fan operation speeds during the last voltage offsets before a crash finally occurred. Concerning the SST speculation, we found references such as [38] and [39], where consumers experienced temperature spikes during the system operation on systems that support SST. During our observation of both temperature and power measurements we always documented the highest values of these variables. Since SST is running autonomously, we speculate that the loss in temperature efficiency is caused due to random temperature spikes.

Finally, the highest temperature gain in Skylake setup at 2.3 GHz was 11% for *mcf* (Core 1) and at 1.2 GHz was 10% for *bwaves* (Core 0). In general, in *core-to-core* variation we observed higher temperature gain values (except the loss cases) for the Skylake chip versus the Haswell one. This isn't true for All Cores and All Cores without Hyper-Threading variations where Haswell was almost always slightly ahead.

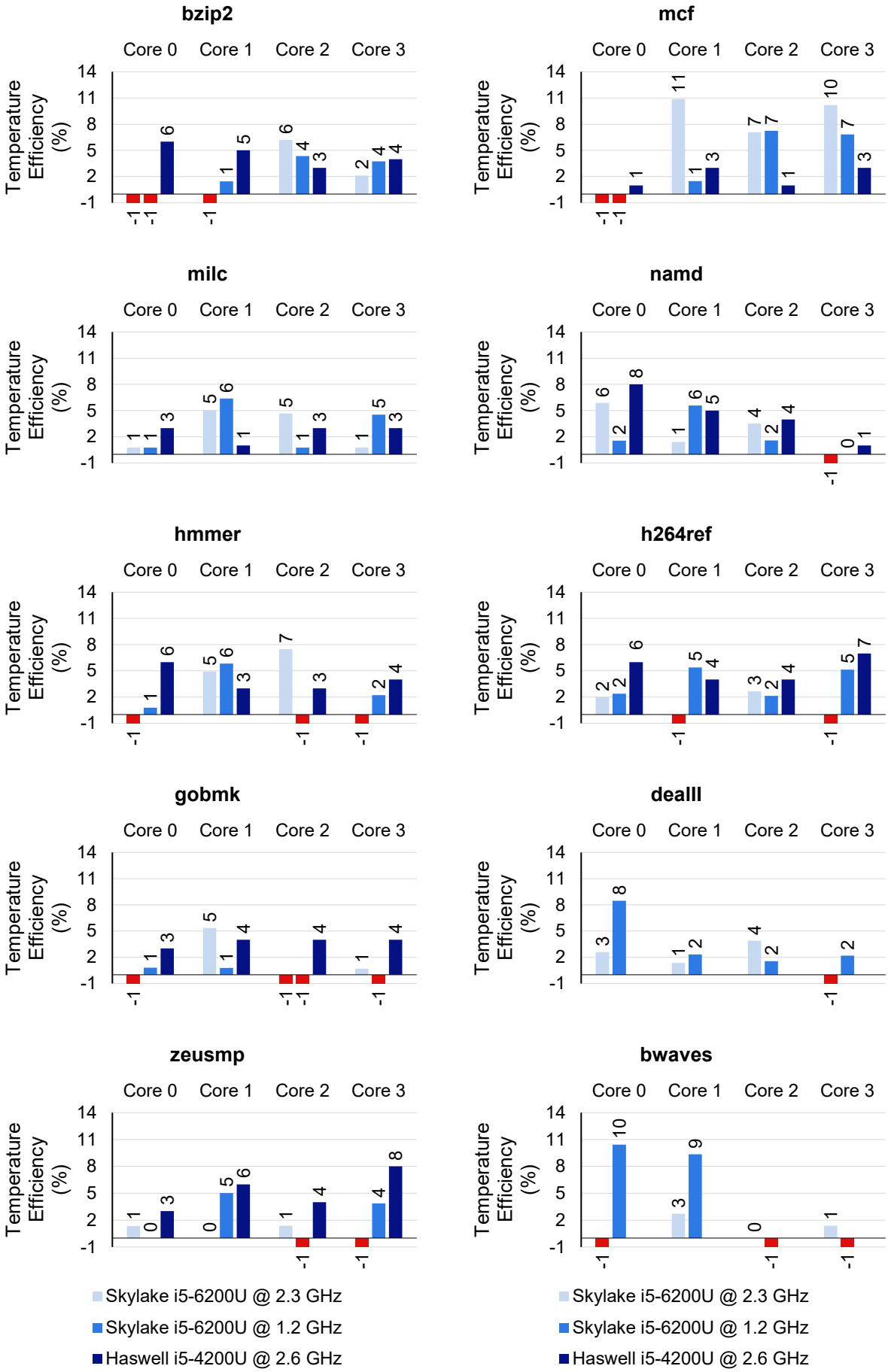


Figure 36: Temperature efficiency percentage for 10 SPEC CPU2006 benchmarks among Skylake and Haswell chips in core-to-core variation

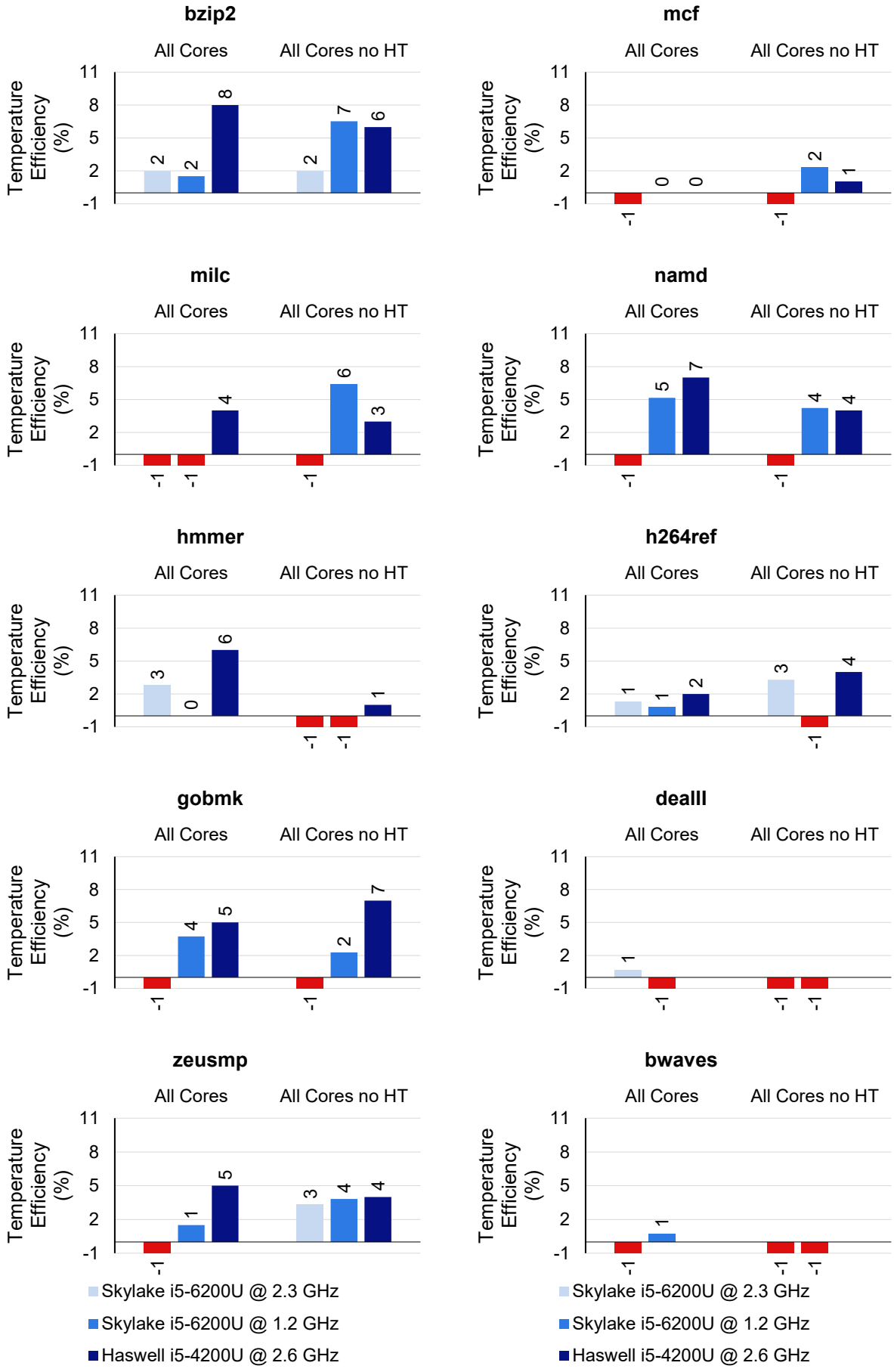


Figure 37: Temperature efficiency percentage for 10 SPEC CPU2006 benchmarks among Skylake and Haswell chips in All Core and All Cores without Hyper-Threading variations

4.7.4 Power Efficiency

We conclude the presentation of results of this chapter with the presentation of power efficiency metric. Power efficiency is calculated as the percentage variation between the power value at the lowest safe voltage offset and the power at nominal voltage. The analysis of *core-to-core* variations in Figure 38 and Figure 39, showed that contrary to the temperature efficiency metric, Skylake outperformed Haswell in terms of power reduction.

More specifically, the highest power reduction observed in the case of Skylake at 2.3 GHz with a value of 41% for *namd* benchmark in Core 0. In the case of Skylake at 1.2 GHz, in most of the configurations the power efficiency was nearly zero. Since, the microprocessor operated almost always in the stable and already low power of 3 W (as we can see in subsection 4.7.2), it seems that even we reduce the frequency to 50% we cannot achieve lower power consumption for every affinity. However, the highest power reduction observed is 33% for *milc* and *h264ref* benchmarks both in Core 3. Finally, the Haswell in most cases achieved a consistent power reduction of 20% while the highest one was 22% for *zeusmp* benchmark in All Cores no HT affinity configuration.

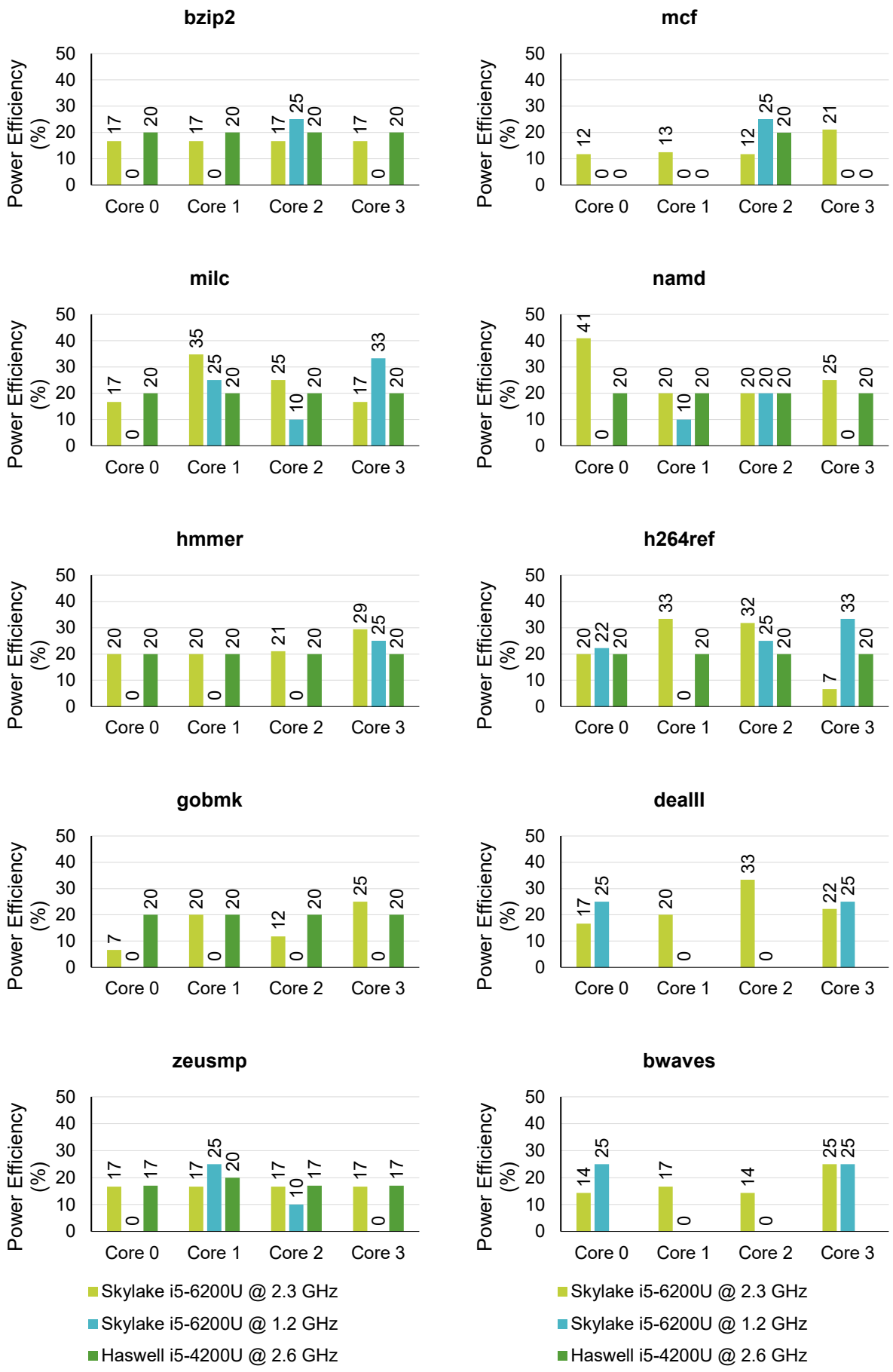


Figure 38: Power efficiency percentage for 10 SPEC CPU2006 benchmarks among Skylake and Haswell chips in core-to-core variation

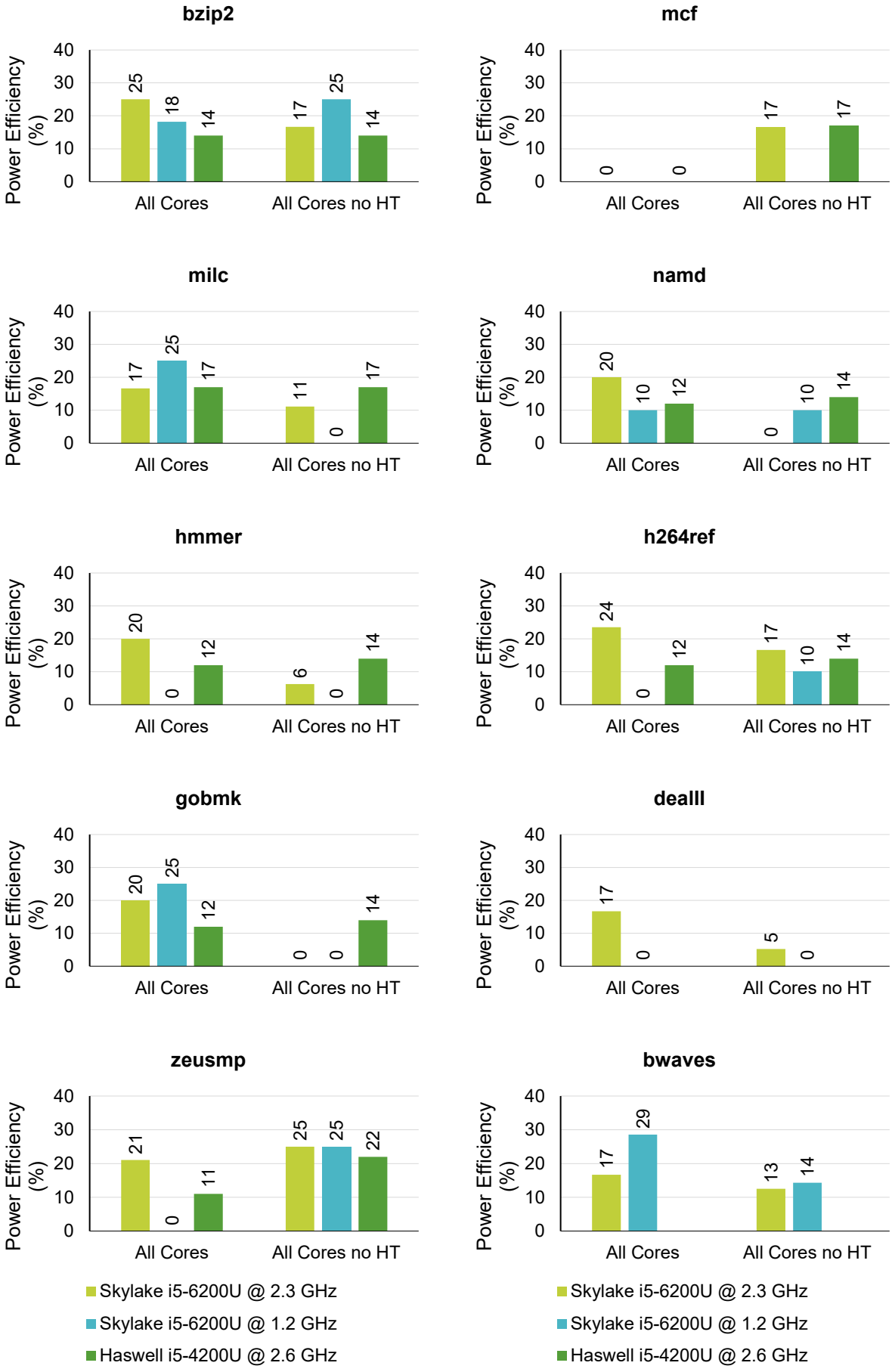


Figure 39: Power efficiency percentage for 10 SPEC CPU2006 benchmarks among Skylake and Haswell chips in All Core and All Cores without Hyper-Threading variations

5. CONCLUSIONS

In this final chapter we summarize the contributions of this thesis and suggest directions for future work.

The main research question was what are the advancements in voltage margins among different ultra-low power microprocessors. We demonstrated in an exhaustive way that the new Skylake generation can be undervolted 11.24% i.e. 100 mV below the nominal voltage (890 mV) while for Haswell this happens at 9.48% i.e. 80 mV below the nominal voltage (844 mV). This indicates a marginal increase in voltage reduction of 1.76 percentile units for Skylake microprocessor. Our study was further extended by evaluating Skylake's behavior under half the base frequency (1.2 GHz), which didn't uncover any lower safe voltage margins. Regarding errors other than system crashes, Skylake system produced such errors at single benchmark runs. In general, we didn't observe unsafe regions, contrary to the previous Haswell study. We also introduced, an extended research around the crash areas and we exposed a microprocessor metastable state where safe operation regions extend even beyond the first crash occurrence. The temperature gains in Haswell were consistent, while in Skylake we even observed losses in temperature efficiency at the lowest safe voltage offset due to either that Enhanced Intel SpeedStep Technology was turned off, by the support of the newly introduced Intel Speed Shift Technology, or both. However, Skylake outperformed Haswell in terms of power efficiency, as we monitored higher gains almost at every affinity configuration.

As future work, we propose the development of a uniform and automated framework under Windows operating system. This will contribute in the direction of simpler and customized data acquisition. Both full and half speed studies of i5-6200U showed that as soon as a critically low voltage is reached, a system crash occurs even before the start of the 3-benchmark cycle. Therefore, we also encourage the characterization of the microprocessor through a different benchmark execution scheme such as the one shown in Image 12 versus the one we used in our study (horizontal scheme).

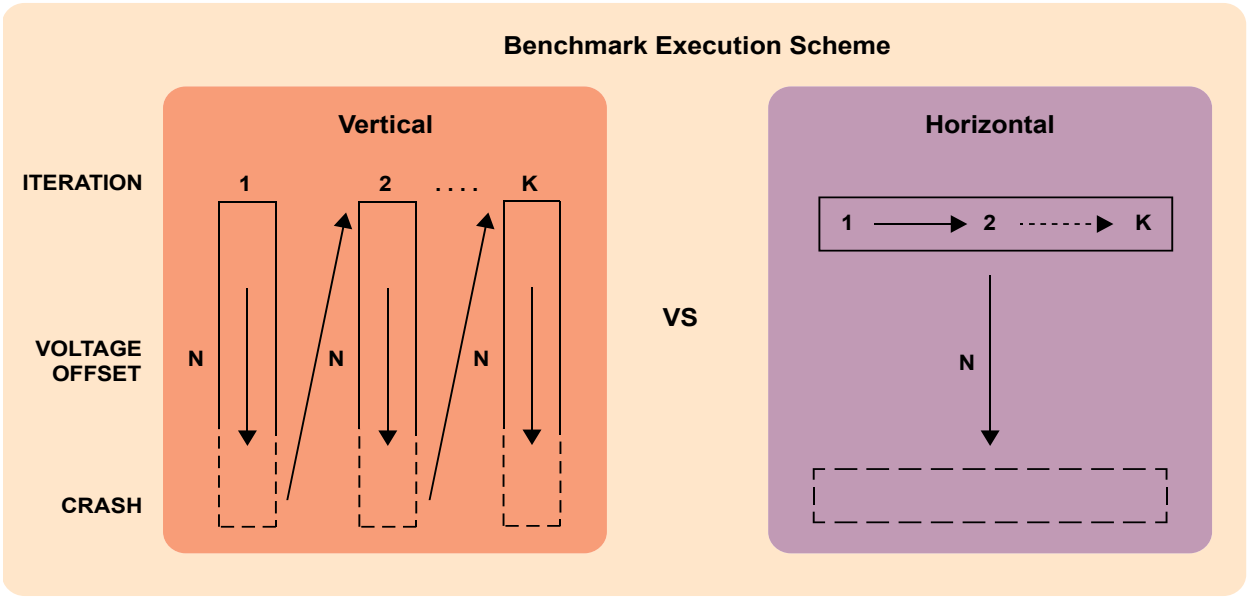


Image 12: Vertical versus horizontal benchmark execution scheme

The execution of benchmarks, in independent vertical voltage pool sets will produce faster, an initial rough estimation of safe operation voltage margins. This can be proved

with the following example. Let's assume that a microprocessor undervoltage tolerance terminates *N* levels of voltage offsets below the nominal voltage. Then, if we execute the benchmark *K* times at every voltage offset, the time that will pass until we have a first system crash evaluation will be $N \cdot K$ times more, than the case where the benchmark would run only once every voltage offset (*N* times).

Finally, the prediction of microprocessor safe voltage reduction guard bands could be done following the current scientific trends by applying machine learning algorithms. For this to come, research community must first have a large data pool of characterized microprocessors. This can be realized by following backwards existing microprocessor lineages. Then, by looking specific parameters based on manufacturing technology and microprocessor architecture, in the future we might be in position to optimally reduce the voltage of the forthcoming generations of microprocessors beforehand.

ABBREVIATIONS - ACRONYMS

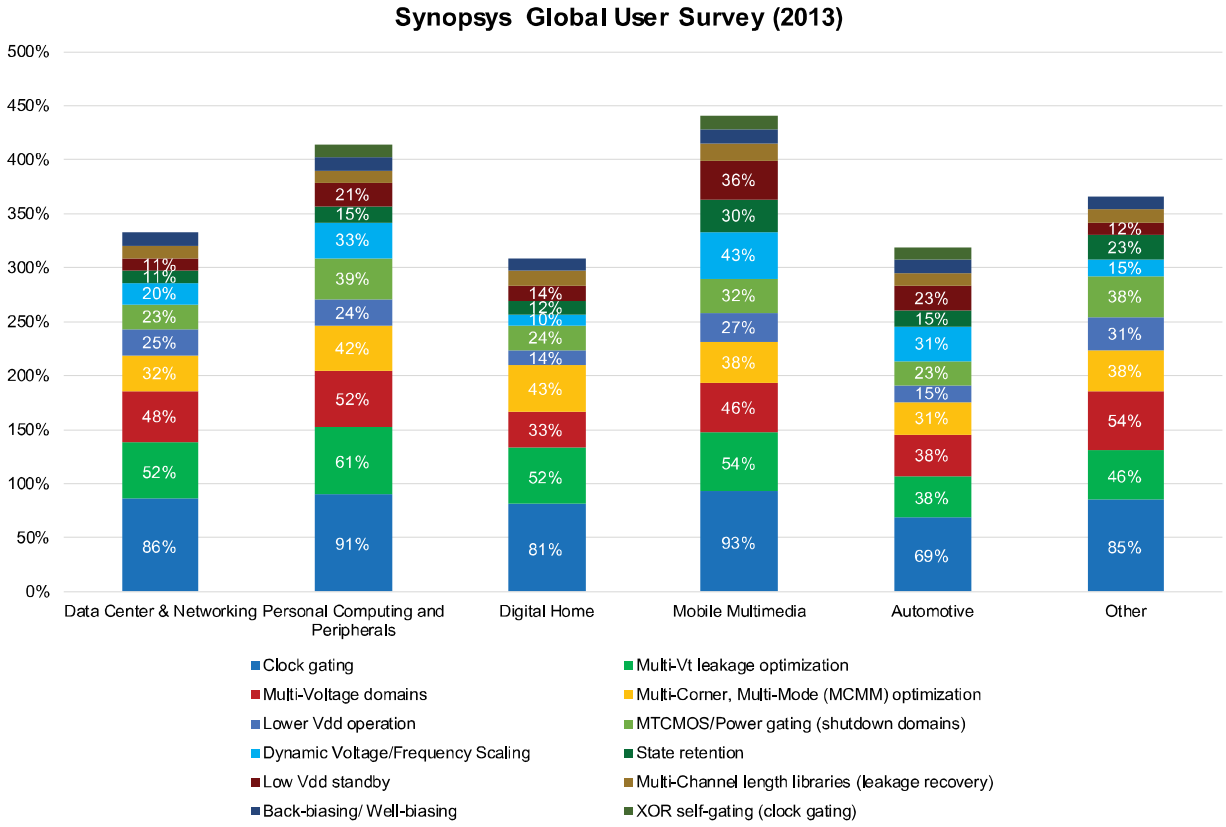
#MC	Machine-Check Exception
AC	Application Crash
ACPI	Advanced Configuration and Power Interface
ATAP	Absolute Total Average Power
ATAT	Absolute Total Average Temperature
BERT	BOOT Error Record Table
BSOD	Blue Screen of Death
CE	Corrected Error
CMCI	Corrected Machine-Check Error Interrupt
CPU	Central Processing Unit
DDR3	Third-generation Double Data Rate SDRAM memory technology
DRAM	Dynamic Random-Access Memory
DTLB	Data TLB
DVFS	Dynamic Voltage/Frequency Scaling
DVS	Dynamic Voltage Scaling
ECC	Error-Correcting Code
eDRAM	Embedded DRAM
EINJ	Error Injection Table
EIST	Enhanced Intel SpeedStep Technology
ERST	Error Record Serialization Table
ETW	Event Tracing for Windows
FIVR	Fully Integrated Voltage Regulator
FP	Floating Point
FPGA	Field Programmable Gate Array
FRC	Functional Redundancy Check
GPU	Graphics Processing Unit
HEST	Hardware Error Source Table
I/O	Input/Output
Intel AVX2	Intel Advanced Vector Extension 2.0
Intel HT	Intel Hyper-Threading
Intel SST	Intel Speed Shift Technology

ITLB	Instruction TLB
LLHEH	Low Level Hardware Error Handler
LLC	Last Level Cache
MC	Memory Controller
MCA	Machine-Check Architecture
MSR	Model-Specific Register
MIM	Metal-Insulator-Metal
NoC	Network-on-Chip
OS	Operating System
PSHED	Platform-Specific Hardware Error Driver
PLL	Phase Locked Loop
QoS	Quality of Service
RAPL	Running Average Power Limit
ROM	Read-Only Memory
SA	System Agent
SC	System Crash
SDC	Silent Data Corruption
SDRAM	Synchronous DRAM
SMM	System Management Mode
SRAO	Software recoverable action optional
SRAR	Software recoverable action required
STLB	Second-level TLB
TDP	Thermal Design Power
TLB	Translation Lookaside Buffer
uArch	Micro-Architecture
UC	Uncorrected
UCNA	Uncorrected no action required
UCR	Uncorrected Recoverable
UEFI	Unified Extensible Firmware Interface
WHEA	Windows Hardware Error Architecture
µOp	Micro-Operation

°C	**Celsius**
MHz	**MegaHertz**
GHz	**GigaHertz**
KB	**KiloByte**
MB	**MegaByte**
W	**Watt**
MW	**MegaWatt**
V	**Volt**
mV	**milliVolt**
nm	**nanometer**
mm^2	**square millimeter**
s	**second**

Metric Units

ANNEX



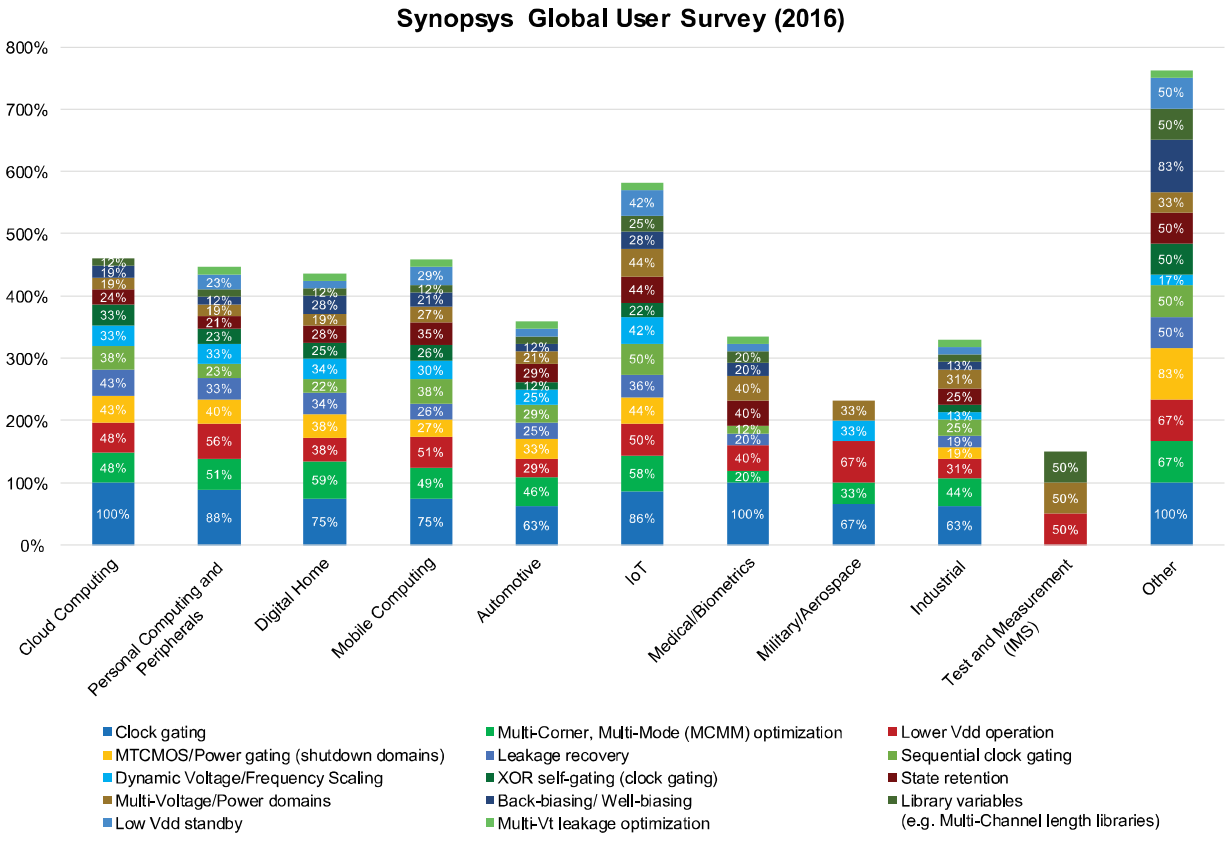


Image 13: Low power techniques usage across application markets (DVFS is shown in cyan color) [40], [41]

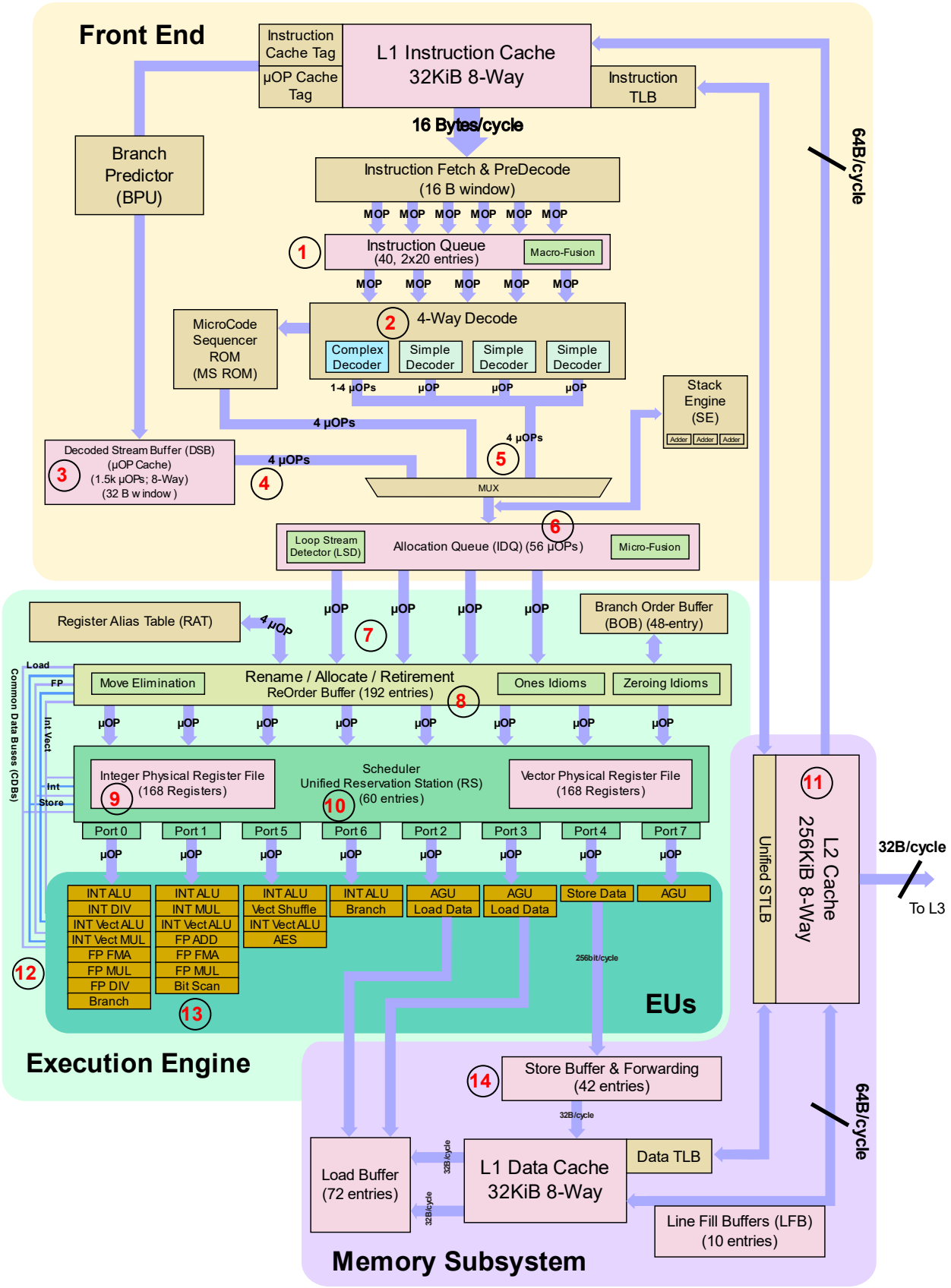


Image 14: Haswell single core block diagram [42]

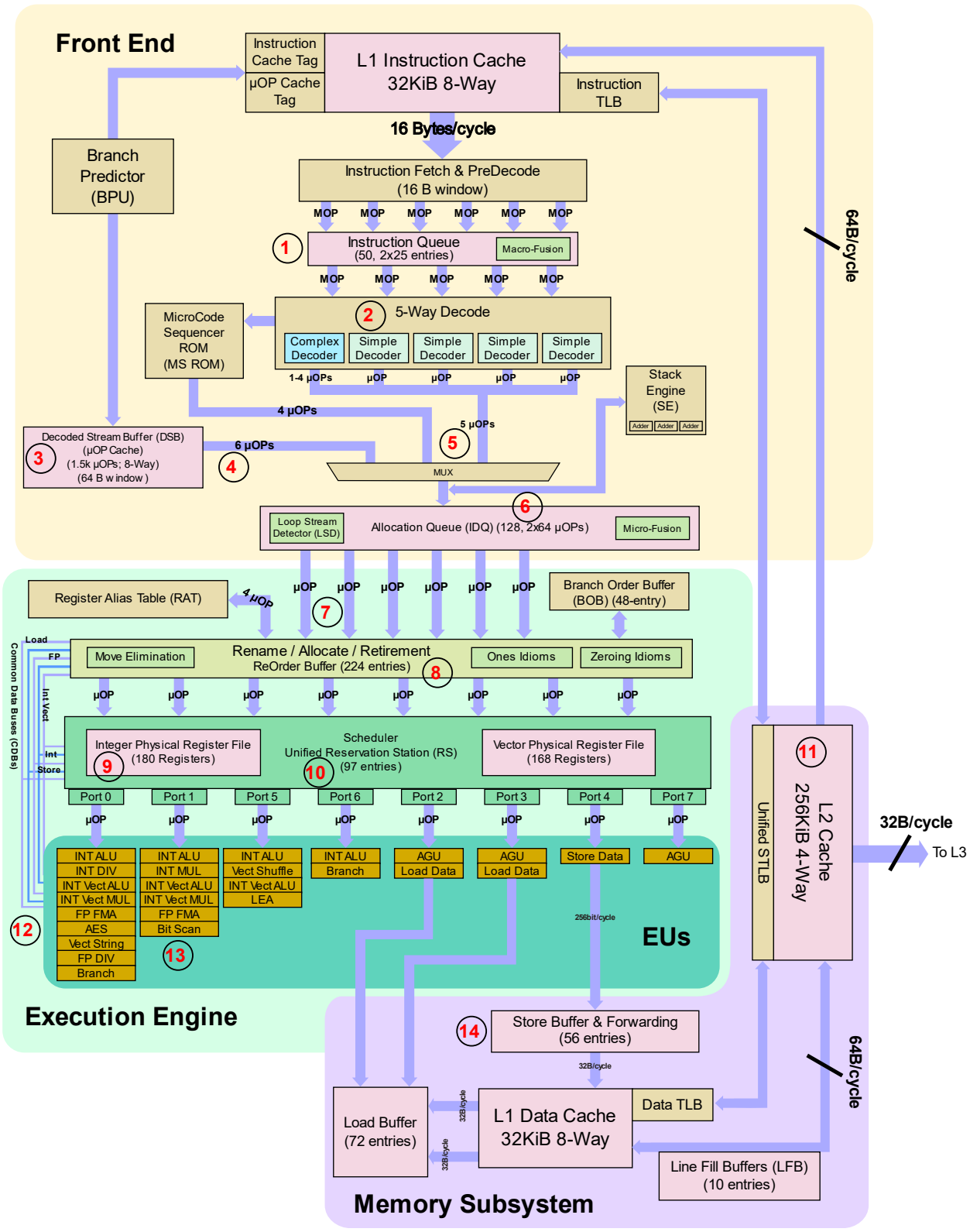


Image 15: Skylake single core block diagram [43]

REFERENCES

[1] D. Kramer, "DOE steps further toward exascale computing," 11 04 2018. [Online]. Available: https://physicstoday.scitation.org/do/10.1063/PT.6.2.20180411a/full/. [Accessed 07 08 2019].

[2] I. Ratković, N. Bežanić, O. S. Ünsal, A. Cristal and V. Milutinović, "Chapter One - An Overview of Architecture-Level Power- and Energy-Efficient Design Techniques," in *Advances in Computers*, Elsevier, 2015; doi:https://doi.org/10.1016/bs.adcom.2015.04.001, pp. 1-57.

[3] J. Haj-Yahya, A. Mendelson, Y. B. Asher and A. Chattopadhyay, Energy Efficient High Performance Processors: Recent Approaches for Designing Green High Performance Computing, Singapore, Singapore: Springer, 2018.

[4] A. Bacha and R. Teodorescu, "Dynamic Reduction of Voltage Margins by Leveraging On-chip ECC in Itanium II Processors," in *Proceedings of the 40th Annual International Symposium on Computer Architecture*, New York, NY, USA, 2013; doi:10.1145/2485922.2485948.

[5] E. Le Sueur and H. Gernot, "Dynamic voltage and frequency scaling: the laws of diminishing returns," in *Proceedings of the International Conference on Power-Aware Computing and Systems (HotPower)*, Vancouver, British Columbia, Canada, 2010.

[6] K. Koukos, D. Black-Schaffer, V. Spiliopoulos and S. Kaxiras, "Towards more efficient execution: a decoupled access-execute approach," in *Proceedings of the 27th international ACM conference on International conference on supercomputing (ICS)*, Eugene, Oregon, USA, 2013; doi:10.1145/2464996.2465012.

[7] P. Mantovani, E. G. Cota, K. Tien, C. Pilato, G. D. Guglielmo, K. Shepard and L. P. Carloni, "An FPGA-Based Infrastructure for Fine-Grained DVFS Analysis in High-Performance Embedded Systems," in *Proceedings of the 2016 53rd Annual Design Automation Conference (DAC)*, Austin, Texas, 2016; doi:10.1145/2897937.2897984.

[8] C. Chow, L. Tsui, P. Leong and W. Luk, "Dynamic Voltage Scaling for Commercial FPGAs," in *Proceedings of the 2005 IEEE International Conference on Field-Programmable Technology (FPT)*, Singapore, 2005; doi:10.1109/FPT.2005.1568543.

[9] J. Leng, A. Buyuktosunoglu, R. Bertr, P. Bose and V. J. Reddi , "Safe limits on voltage reduction efficiency in GPUs: A direct measurement approach," in *Proceedings of the 2015 48th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Waikiki, Hawai, USA, 2015; doi:10.1145/2830772.2830811.

[10] X. Mei, L. S. Yung, K. Zhao and X. Chu, "A Measurement Study of GPU DVFS on Energy Conservation," in *Proceedings of the Workshop on Power-Aware Computing and Systems (HotPower)*, Farmington, Pennsylvania, 2013; doi:10.1145/2525526.2525852.

[11] S. Usman, S. U. Khan and S. Khan , "A comparative study of voltage/frequency scaling in NoC," in *Proceedings of the IEEE International Conference on Electro-Information Technology (EIT)*, Rapid City, South Dakota, USA, 2013; doi:10.1109/EIT.2013.6632716 .

[12] S. Wang, Z. Qian, J. Yuan and I. You, "A DVFS Based Energy-Efficient Tasks Scheduling in a Data Center," *IEEE Access,* vol. 5, pp. 13090-13102, 2017; doi:10.1109/ACCESS.2017.2724598.

[13] G. Papadimitriou, M. Kaliorakis, A. Chatzidimitriou, C. Magdalinos and D. Gizopoulos, "Voltage margins identification on commercial x86-64 multicore microprocessors," in *Proceedings of the 2017 IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS)*, Thessaloniki, Greece, 2017; doi:10.1109/IOLTS.2017.8046198.

[14] 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 the 2017 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Cambridge, Massachusetts, USA, 2017; doi:10.1145/3123939.3124537.

[15] B. Salami, O. Unsal and A. Cristal, "Fault Characterization Through FPGA Undervolting," in *Proceedings of the 2018 28th International Conference on Field Programmable Logic and Applications (FPL)*, Dublin, Ireland, 2018; doi:10.1109/FPL.2018.00023.

[16] C. Magdalinos, "Exposing the Voltage Design Margins of Modern x86 Microprocessors", BSc thesis, Dept. of Informatics and Telecommunication, National and Kapodistrian University of Athens, Athens, 2016.

[17] Y. Lee, M. Seok, S. Hanson, D. Blaauw and D. Sylvester , "Standby power reduction techniques for ultra-low power processors," in *Proceedings of the 34th European Solid-State Circuits Conference (ESSCIRC)*, Edinburgh, UK, 2008; doi:10.1109/ESSCIRC.2008.4681823.

[18] M. Travers, "CPU Power Consumption Experiments and Results Analysis of Intel i7-4820K," tech. report, μSystems Research Group, School of Electrical and Electronic Engineering, Newcastle University, 2015.

[19] "Intel® 64 and IA-32 Architectures Software Developer's Manual, vol. 3B," Order Number: 253669-070US, Intel Corp., May 2019, pp. 15.1-15.36.

[20] G. McIntyre, "Interpreting a WHEA error for a MCA fault – Ntdebugging Blog," 28 01 2011. [Online]. Available: https://blogs.msdn.microsoft.com/ntdebugging/2011/01/28/interpreting-a-whea-error-for-a-mca-fault/. [Accessed 07 08 2019].

[21] M. Jacobs, "Windows Hardware Error Architecture Overview - Windows drivers | Microsoft Docs," 20 04 2017. [Online]. Available: https://docs.microsoft.com/en-us/windows-hardware/drivers/whea/windows-hardware-error-architecture-overview. [Accessed 07 08 2019].

[22] ScotXW, "File:Fully Integrated Voltage Regulator.svg - Wikimedia Commons," 21 07 2017. [Online]. Available: https://commons.wikimedia.org/wiki/File:Fully_Integrated_Voltage_Regulator.svg. [Accessed 07 08 2019].

[23] E. A. Burton, G. Schrom, F. Paillet, J. Douglas, W. J. Lambert, K. Radhakrishnan and M. . J. Hill, "FIVR — Fully integrated voltage regulators on 4th generation Intel® Core™ SoCs," in *Proceedings of the 2014 IEEE Applied Power Electronics Conference and Exposition (APEC)*, Fort Worth, Texas, USA, 2014; doi:10.1109/APEC.2014.6803344.

[24] P. Hammarlund, A. J. Martinez, A. A. Bajwa, D. L. Hill, E. Hallnor, H. Jiang, M. Dixon, M. Derr, M. Hunsaker, R. Kumar, R. . B. Osborne, R. Rajwar, R. Singhal, R. D'Sa, R. Chappell, S. Kaushik, S. Chennupaty, S. Jourdan, S. Gunther, T. Piazza and T. Burton, "Haswell: The Fourth-Generation Intel Core Processor," *IEEE Micro,* vol. 34, no. 2, 2014, pp. 6-20; doi:10.1109/MM.2014.10.

[25] J. Mandelblat, "Technology Insight: Intel's Next Generation Microarchitecture Code Name Skylake," in *Intel Developer Forum (IDF15)*, 2015.

[26] J. Doweck, W.-f. Kao, A. Kuan-yu Lu, J. Mandelblat, A. Rahatekar, L. Rappoport, E. Rotem, A. Yasin and A. Yoaz, "Inside 6th-Generation Intel Core: New Microarchitecture Code-Named Skylake," *IEEE Micro,* vol. 37, no. 2, 2017, pp. 52-62; doi:10.1109/MM.2017.38.

[27] "Enhanced Intel® SpeedStep® Technology for the Intel® Pentium® M Processor," white paper, Intel Corp., 2004.

[28] "6th Generation Intel® Processor Families for U/Y-Platforms, Datasheet, vol. 1 of 2," Order Number: 332990-008EN, Intel Corp., August 2018.

[29] E. Rotem, "Intel® Architecture, Code Name Skylake Deep Dive: A New Architecture to Manage Power Performance and Energy Efficiency," in *Intel Developer Forum (IDF15)*, 2015.

[30] D. T. Marr, F. Binns, D. L. Hill, G. Hinton, D. A. Koufaty, J. A. Miller and M. Upton, "Hyper-Threading Technology Architecture and Microarchitecture," *Intel Technology Journal,* Q1 2002.

[31] J. Collake, "Process Lasso v9.0.0.440," Bitsum LLC, Talbott, Tennessee, USA, 2002-2019.

[32] S. Choi and S. Wheeler, "PowerShell Scripting | Microsoft Docs," 27 08 2018. [Online]. Available: https://docs.microsoft.com/en-us/powershell/scripting/overview?view=powershell-6. [Accessed 07 08 2019].

[33] "Event Viewer | Microsoft Docs," 22 02 2013. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc766042(v=ws.11). [Accessed 07 08 2019].

[34] "Intel® Extreme Tuning Utility (Intel® XTU) v6.4.1.19," Intel Corp., Santa Clara, California, USA, 2011-2019.

[35] M. Malík - REALiX, "HWiNFO v5.74," Malacky, Slovakia, 1995-2019.

[36] J. L. Henning, "SPEC CPU2006 benchmark descriptions," *ACM SIGARCH Computer Architecture News,* vol. 34, no. 4, 2006, pp. 1-17; doi:10.1145/1186736.1186737.

[37] D. Marshall and E. Graff, "Blue Screen Data - Windows drivers | Microsoft Docs," 23 05 2017. [Online]. Available: https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger/blue-screen-data. [Accessed 07 08 2019].

[38] CCapt, "thermal sensor issue i77700k - Intel® Community Forum," 05 05 2017. [Online]. Available: https://forums.intel.com/s/question/0D70P0000068ZPkSAM/thermal-sensor-issue-i77700k?language=en_US&tstart=0#472356. [Accessed 07 08 2019].

[39] raidmaxGuy, "7700k jumpy core temps? - Overclock.net - An Overclocking Community," 15 06 2017. [Online]. Available: https://www.overclock.net/forum/8-intel-general/1632379-7700k-jumpy-core-temps.html. [Accessed 07 08 2019].

[40] M. A. White, "Semiconductor Engineering - Power Reduction Techniques: Are they all the same for established planar, FD-SOI and finFET transistors?," 07 08 2014. [Online]. Available: https://semiengineering.com/power-reduction-techniques/. [Accessed 07 08 2019].

[41] B. Bailey, "Semiconductor Engineering - Power Optimization Strategies Widen: Different markets are heading in different directions, raising questions about whether the chip industry can effectively respond to all of those demands," 10 05 2018. [Online]. Available: https://semiengineering.com/power-optimization-strategies-widen/. [Accessed 07 08 2019].

[42] WikiChip, "Haswell - Microarchitectures - Intel - WikiChip," [Online]. Available: https://en.wikichip.org/wiki/intel/microarchitectures/haswell_(client). [Accessed 07 08 2019].

[43] WikiChip, "Skylake (client) - Microarchitectures - Intel - WikiChip," [Online]. Available: https://en.wikichip.org/wiki/intel/microarchitectures/skylake_(client). [Accessed 07 08 2019].