Methods for Robust and Energy-Efficient Microprocessor Architectures

George Papadimitriou[1]

National and Kapodistrian University of Athens
Department of Informatics and Telecommunications
georgepap@di.uoa.gr

Abstract. This work presents novel methods for ensuring the correctness of the microprocessor during the post-silicon validation phase and for improving the energy efficiency requirements of modern microprocessors. These methods can be applied during the prototyping phase of the microprocessors or after their release to the market. More specifically, in the first part of the thesis, we present and describe two different ISA-independent software-based post-silicon validation methods, which contribute to formalization and modeling as well as the acceleration of the post-silicon validation process and expose difficult-to-find bugs in the address translation mechanisms (ATM) of modern microprocessors. In the second part of the thesis we present a detailed system-level voltage scaling characterization study for two state-of-the-art ARMv8-based multicore CPUs. We present an extensive characterization study which identifies the pessimistic voltage guardbands (the increased voltage margins set by the manufacturer) of each individual microprocessor core and analyze any abnormal behavior that may occur in off-nominal voltage conditions. We then introduce the development of dedicated programs (diagnostic micro-viruses) that aim to accelerate the time-consuming voltage margins characterization studies by stressing the fundamental hardware components. Finally, we present a comprehensive exploration which aims (1) to identify the best performance per watt operation points, (2) to reveal how and why the different core allocation options affect the energy consumption, and (3) to enhance the default Linux scheduler to take task allocation decisions for balanced performance and energy efficiency.

Keywords: Dependability, Correctness, Post-Silicon Validation, Design Bugs, Address Translation, Energy Efficiency, Voltage Margins.

1 Introduction

Reducing supply voltage is one of the most efficient techniques to reduce the dynamic power consumption of the microprocessor, because dynamic power is quadratic in voltage. However, supply voltage scaling increases subthreshold leakage currents, increases leakage power, and also poses numerous circuit design challenges. Process

[1] *Dissertation Advisor: Dimitris Gizopoulos, Professor.*

variations and temperature parameters (dynamic variations), caused by different workload interactions are also major factors that affect microprocessor's energy efficiency. Furthermore, during microprocessor chip fabrication, process variations can affect transistor dimensions (length, width, oxide thickness etc. [1]) which have direct impact on the threshold voltage of a MOS device [2]. As technology scales further down, the percentage of these variations compared to the overall transistor size increases and raises major concerns for designers, who aim to improve energy efficiency. This variation is classified as static variation and remains constant after fabrication. Both static and dynamic variations lead microprocessor architects to apply conservative guardbands (operating voltage and frequency settings), as shown in Fig. 1a to avoid timing failures and guarantee correct operation, even in the worst-case conditions excited by unknown workloads, environmental conditions, and aging [3]. The guardband results in faster circuit operation under typical workloads than required at the target frequency, resulting in additional cycle time, as shown in Fig. 1b. In case of a timing emergency caused by voltage droops, the extra margin prevents timing violations and failures by tolerating circuit slowdown. While static guardbanding ensures robust execution, it tends to be severely overestimated as timing emergencies rarely occur, making it less energy-efficient [4].

As operating frequency and integration density increase, the total chip power dissipation also increases. This is evident from the fact that due to the demand for increased functionality on a single chip, more and more transistors are being packed on a single die and hence, the switching frequency increases in every technology generation. However, by developing aggressive and sophisticated mechanisms to boost performance and enhance the energy efficiency in conjunction with the decrease of the size of transistors, microprocessors have become extremely complex systems, making the microprocessor verification and manufacturing testing a major challenge for the semiconductor industry. Microprocessors integrate various types of cores and functional units and are highly adaptive for dynamically optimizing the peak performance, power efficiency, and idle power consumption. Today microprocessors are complex, heterogeneous machines that contain different cores for different types of workloads. This complexity and heterogeneity bring forward the difficult task of design verification and validation. Even for most established microprocessor vendors, the task of verifying a modern microprocessor and ensuring correct operation is increasingly challenging [5]. Always trying to

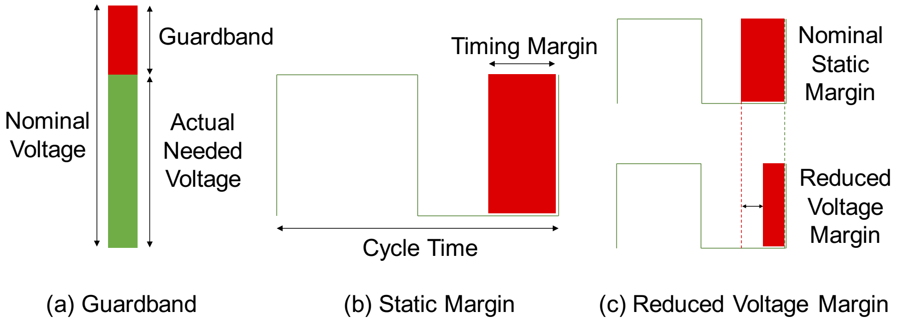

Fig. 1. Voltage guardband ensures reliability by inserting an extra timing margin. Reduced voltage margins improve total system efficiency without affecting the reliability.

deliver higher performance but also energy efficiency to end-users, manufacturers are forced to gradually design more complex circuits and employ very large verification teams to eliminate all design bugs (if possible) in a timely manner.

2 Post-Silicon Validation of the Address Translation Mechanisms

Post-silicon validation aims to ensure that the first "real" thing (not models of the design), i.e., the prototype chips, conform to microprocessor design specifications. It allows detection of rare functional design bugs, but also electrical bugs that manifest themselves only under certain operating conditions, such as thermal effects or process variations. Due to the very high program execution throughput of post-silicon validation (at the speed of the actual prototype chip), the design verification teams attempt to execute as many test programs as possible (such as automatically generated random and directed random test programs). This way, they can obtain the most extensive possible validation coverage (as many potential validation scenarios as possible are adapted) before massive chip production to detect any anomalous behavior.

With the galloping adoption of virtualization today (and the complexity its support adds to ISAs and microarchitectures), the performance and correctness of the address translation mechanisms (ATMs) get more critical. The ATM of a modern microprocessor includes complex hardware structures and can be a predominant source of severe escaped bugs which are, however, very hard to detect as recent reports have shown. Unlike other hardware components (functional blocks, registers, memory subsystem, peripheral control units, etc.) the output of the ATM of a microprocessor (i.e., the physical address) is not observable to any program or architectural visible locations (e.g., architectural register). Because of the "hidden" operation of a microprocessor's address translation subsystem, observability and long bug detection latencies are critical obstacles for its post-silicon validation and debug. Moreover, the address translation process involves several stages and several different hardware structures (i.e., Translation Lookaside Buffers - TLBs), aiming to improve the total microprocessor's performance, and bugs in each of these blocks may lead to wrong address translations, and thus, to unpredictable system behaviors.

To this end, in this thesis we present two contributions on the post-silicon validation of the address translation mechanisms of modern microprocessors.

2.1 Accelerating Post-Silicon Validation

Special purpose verification languages support automatic stimulus generation to enable better specification and design coverage. These frameworks allow design engineers to express pseudo random test program generation along with complex event scenarios using generic *Test Templates*. A Test Template defines any desired verification scenario, and based on the microprocessor's architecture, the corresponding post-silicon validation programs are generated.

Fig. 2 shows the proposed framework for post-silicon validation of the ATM hardware ([6] [7]). As we explain in the next subsection, our methodology does not require previously generated golden responses by a simulator (a major requirement to facilitate the execution of very large numbers of post-silicon validation programs).

The Program Generation Engine takes as inputs:

- a detailed model of the ATM paths (existing framework),
- a Test Template, which specifies the overall silicon validation plan and describes the test scenarios (existing framework),
- the Page Table information (proposed enhancement), and
- a corresponding memory image (proposed enhancement).

The above scenarios are translated by the Program Generation Engine into complete post-silicon validation programs. The final self-checking validation programs completely stress and validate the ATM hardware of the microprocessor. Subsequently, the validation programs are loaded into the prototype chip (also referred to as the device under validation – DUV) and the validation process begins. In the next subsections we describe the enhancements required in the traditional flow.

Before the execution of the validation programs, the physical memory locations that the test program will validate (of course these should be different than the ones holding the validation program itself or the page table) are written with data values equal to their actual physical address (i.e., every physical address A in the range being validated contains value A).

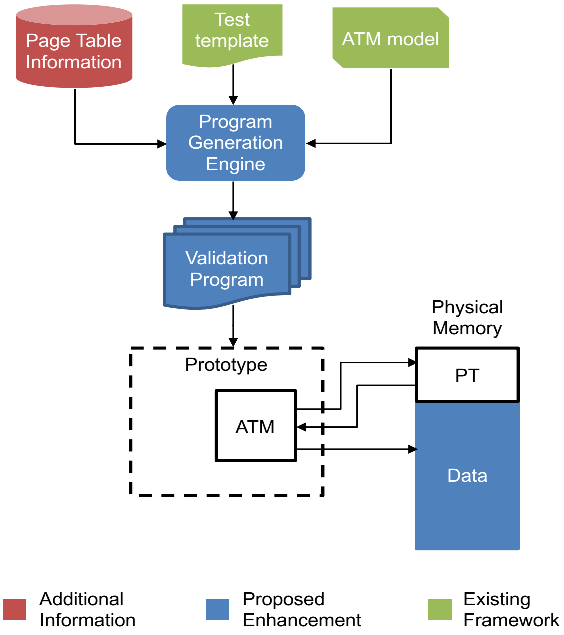

Fig. 2. The proposed ATM silicon validation framework.

While storing the desired data in memory, the microprocessor operates in real-addressing mode during this phase, so that the ATM is disabled and does not impede the proper memory initialization of the desired data (otherwise a bug in ATM could threaten the initial state of the physical memory). This is a key for the proposed methodology, because we must ensure that the initial data in memory must be correct before the validation process begins. Additionally, in most of the cases, the DRAM in a commercial server is ECC protected, so we are sure that the stored data in memory are not jeopardized. For example, if the physical address 0x123456000 belongs to the area being validated, the data value 0x123456XXX is stored in it. The last 12 bits can have any value since they represent the Offset. Therefore, when for example a virtual address VA=0xAABBCCXXX is mapped to the physical address PA=123456XXX, the fetched datum will be equal to 0123456XXX.

2.2 Unveiling Difficult Bugs in Address Translation Mechanisms

The second contribution of this thesis, which is presented in [8], further enhances the post-silicon validation phase of the ATM by presenting another novel self-checking validation method that unveils and detects rare bug scenarios in Address Translation Caching Arrays (ATCA). ATCAs are among the most important structures for microprocessor functionality and performance and escaped bugs in these arrays can lead to unpredictable system behaviors in the field. Using a comprehensive experimental study, we first present and analyze rare bug scenarios and demonstrate the reason why they are difficult to detect. Our goal is to unveil and detect difficult bugs in ATCAs. Even if bugs manifest themselves by executing traditional validation tests, detecting them is unlikely due to the high possibility of masking during the execution of a traditional validation test. Another reason is that there are bugs in ATCAs that may affect only the performance of the microprocessor and not its functionality. For these reasons, we propose a novel method that guarantees detection of such difficult bugs in the silicon prototype. Our validation method is self-checking (like the previous one) and does not require any hardware instrumentation. We demonstrate that, unlike traditional end-of-test checking techniques, this method effectively detects all the injected bugs we created, by using common use-cases of a real hardware prototype.

As shown in Fig. 3, our proposed methodology detected all 2559 bugs injected into the Gem5 simulator (bug coverage 100%). On the other hand, we compared the proposed method to the end-of-test checking techniques, which detected only 255 injected bugs (bug coverage 9.97%). This difference is due to the limitations of these techniques in the validation flow, given that they check if the output of the DUV is equivalent to a golden reference output. In most of the cases (2304 out of 2559 bugs), although a bug is excited, it does not affect the output.

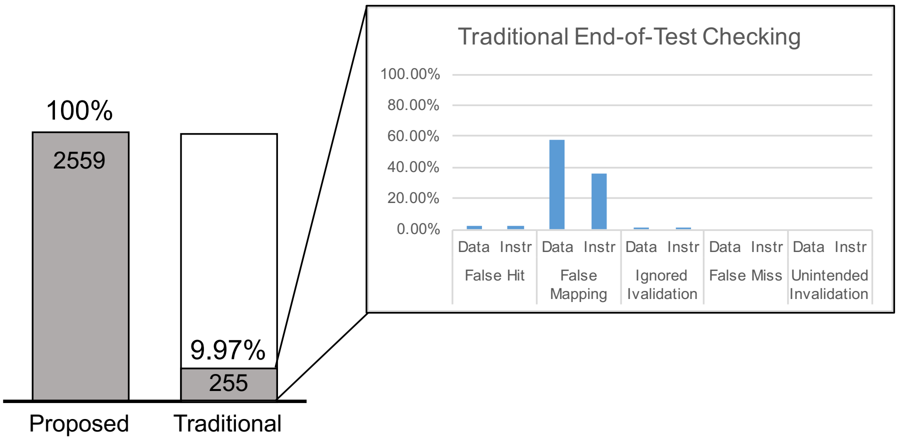

Fig. 3. Bug coverage for the proposed method vs. traditional end-of-test checking techniques.

3 Energy Efficiency for Multicore CPUs

3.1 Measuring Voltage Guardbands of Server-Grade ARMv8 CPU Cores

The third contribution of this thesis is a system-level voltage scaling characterization study for ARMv8-based multicore CPUs manufactured in 28nm ([9] [10]). The primary targets of this study are

1. to identify the pessimistic voltage guardbands (the increased voltage margins set by the manufacturer) of each individual microprocessor core by exposing their safe V_{min}, as shown in Fig. 4. Safe V_{min} is defined as the minimal working voltage of the microprocessor for any workload or operating condition at a specific clock frequency, and

2. to characterize and analyze any abnormal behavior that occur in voltage levels below the safe V_{min}.

The study's backbone is a fully automated system-level framework built around Applied Micro's (APM) X-Gene 2 micro-server. The automated infrastructure aims to increase the throughput of massive undervolting campaigns that require multiple benchmarks execution at several voltage supply levels of all individual cores. The characterization process requires minimal human intervention and records all possible abnormalities due to undervolting: silent data corruptions (SDC, e.g., program output mismatches without any hardware error notification), corrected errors, uncorrected (but detected) errors (provided by Linux EDAC driver), as well as application and system crashes.

Towards the formalization of the any potential behavior in undervolting conditions we also present a simple consolidated function; the Severity function. Severity function

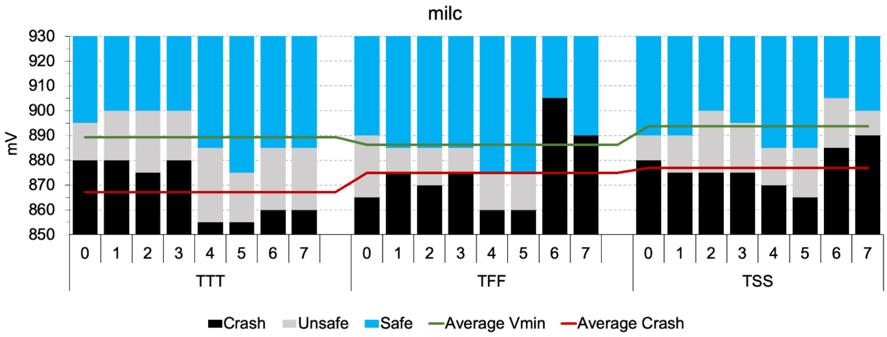

Fig. 4. X-Gene 2 characterization results for *milc* on three different chips (TTT, TFF, TSS). Blue represents the Safe region; grey represents the Unsafe region; and black represents the Crash region.

aggregates the effects of reduced voltage operation in the cores of a multicore CPU by assigning values to the different abnormal observations. The lower the voltage level, the higher the value of the severity function. The severity function assists an undervolting classification of the cores of a CPU chip for a given benchmark: different core, benchmark and voltage values lead to different severity patterns, some with an abrupt increase to the severity (e.g., the benchmark keeps executing correctly until a voltage level at which the system crashes), while others have a "smooth" severity increase while voltage is reduced (the system remains responsive throughout a range of voltage values but it generates ECC errors or produces SDCs). The fine-grained analysis of the behavior of the machine using the severity function can assist energy efficiency decisions for task-to-core allocation by the system software.

3.2 Fast System-Level Voltage Margins Characterization

Characterization campaigns (like the previous one) with many different benchmarks and for many different microprocessor chips are very time-consuming. The accurate identification of the voltage under-scaling limits in a real multicore system requires massive execution of a large number of real workloads in all the cores of the chip (and all different chips of a system), for different voltage and frequency values. For instance, to identify the V_{min} of each one of the eight cores of the Applied Micro's (APM) X-Gene 2 microprocessor, we used the SPEC CPU2006 benchmarks and repeated each experiment 10 times starting from the nominal voltage value (980mV) until their crash voltage value (~880mV). These experiments required about 2 months for a complete characterization for all the cores of one microprocessor chip.

To accelerate the characterization process, we introduce the development of dedicated programs (diagnostic micro-viruses), which are presented in [11] and is the fourth contribution of this thesis. The micro-viruses aim to stress the most fundamental hardware components of the microprocessor aiming to provide quickly the safe V_{min} (Fig. 5). With our diagnostic micro-viruses, we effectively stress (individually or simultaneously) all the main components of the chip:

a. the caches (the L1 data and instruction caches, the unified L2 caches and the last level L3 cache of the chips) and

b. the two main functional components of the pipeline (the ALU and the FPU).

These diagnostic micro-viruses are executed in very short time (~3 days for the entire massive characterization campaign for each individual core of one microprocessor chip) compared to normal benchmarks, such as those of the SPEC CPU2006 suite, which need 2 months for a detailed characterization of the same microprocessor chip. The micro-viruses' purpose is to reveal the variation of the safe voltage margins across cores of the multicore chip and also to contribute to diagnosis by exposing and classifying the abnormal behavior of each CPU unit (silent data corruptions, bit-cell errors, and timing failures).

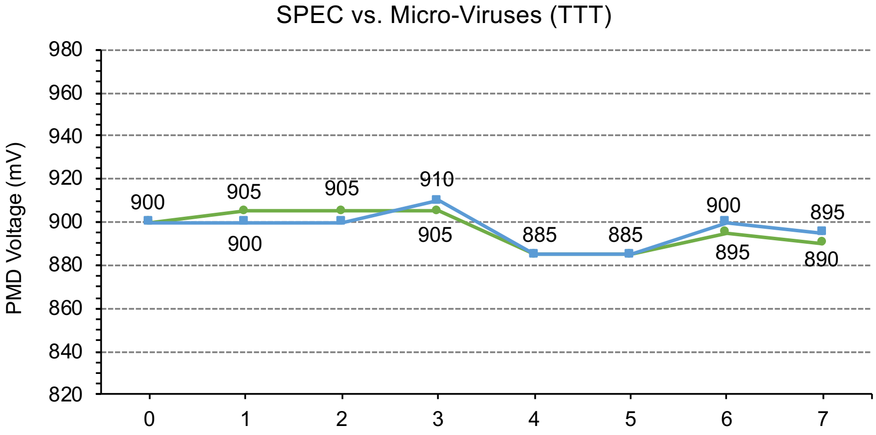

Fig. 5. Maximum Vmin among 12 SPEC CPU2006 benchmarks and the proposed micro-viruses for TTT chip.

3.3 Balancing Energy and Performance on Multicore ARMv8 CPUs

The fifth contribution of this thesis, which presented in [12], is to complete the voltage margins characterization for multicore executions and for different clock frequencies. In that part of the thesis, we also present a new software-based scheme for server-grade machines, which provides large energy savings while maintaining high performance levels. Particularly, in this part of the thesis:

- We expose the pessimistic voltage guardbands (Fig. 6) of two state-of-the-art ARMv8 microprocessors (manufactured in 28nm and 16nm – the X-Gene 2 and X-Gene 3, respectively) to identify the safe V_{min} points of the CPU chips in multicore executions. We show that as the number of active threads increases, core-to-core and workload-to-workload variability has a minimal impact on V_{min}.
- We present measurements on the correlation of the safe V_{min} to the voltage droop magnitude (Fig. 7), and show that in multicore executions the emergency voltage

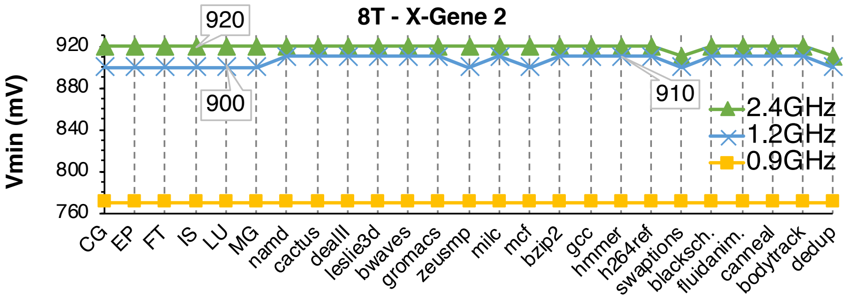

Fig. 6. The Vmin characterization results. This graph presents the X-Gene 2 safe Vmin points for all benchmarks with 8 threads in 2.4 GHz, 1.2 GHz, and 0.9 GHz.

droop events occur regardless of the workload. However, for executions in a single or very few cores, core-to-core and workload-to-workload variability exist to a larger extend.

- We perform an extensive study to identify and analyze the tradeoffs between energy and performance at different voltage and frequency combinations (Fig. 8), as well as at different thread scaling and core allocation configurations. Our analysis reveals that depending on the course-grain characteristics of a program and the number of active threads, there is an optimal combination of voltage, frequency and core allocation for better energy efficiency.

- We also developed a simple online monitoring daemon which monitors the running processes on the system and guides the Linux scheduler to take the appropriate decisions regarding: (a) the core(s) to which a new process should be assigned, and (b) when one or more running processes should be migrated to other cores. At the same time, the daemon dynamically adjusts the V/F settings according to the optimal policies.

- Finally, we evaluate the optimal energy efficient scheme by running the monitoring daemon in a realistic scenario of a server's operation, which (a) randomly selects the issued programs, (b) dynamically migrates the running processes on the system, and (c) dynamically adjusts the voltage and frequency settings for

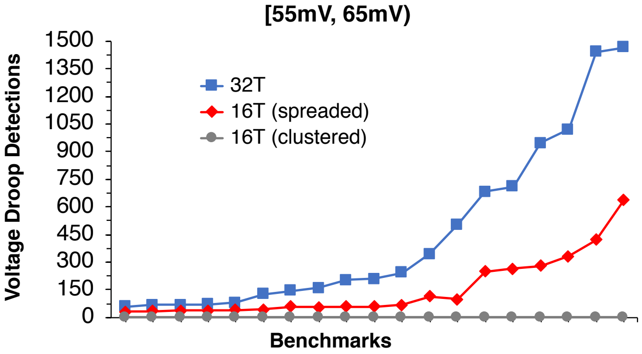

Fig. 7. Voltage droop detections for each program in range between 55mV and 65mV voltage droop magnitude.

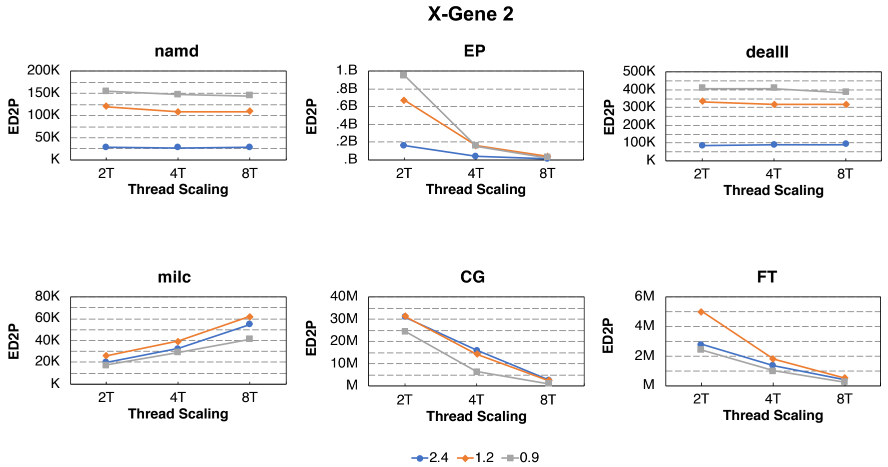

Fig. 8. Energy Delay Squared Product (ED2P) for 8, 4, and 2 threading options for 3 different frequencies (2.4GHz, 1.2GHz, 0.9GHz) of X-Gene 2.

lower energy consumption. We report several comparisons among different configurations to present a detailed evaluation of the optimal scheme, and show that it can achieve on average 25.2% energy savings on X-Gene 2, and 22.3% energy savings on X-Gene 3, with a minimal performance penalty of 3.2% on X-Gene 2 and 2.5% on X-Gene 3 compared to the default voltage and frequency microprocessor's conditions.

The monitoring part of the daemon acts as a watchdog, which periodically monitors the utilized PMDs and the L3C accesses of each running process (except for the system processes). For each process, it counts the L3C accesses during 1M cycles (this actually varies from 300ms to 500ms in our systems; it depends on the IPC rate of each process) and if the L3C accesses are more than 3K, then it classifies this process as memory-intensive, otherwise it classifies it as CPU-intensive. Moreover, it classifies the processes according to the utilized PMDs in order to estimate the current V_{min}.

The second part of the daemon is the Placement. The daemon has been equipped with a fail-safe mechanism: either before the process(es) are invoked or before the frequency should be increased in one or more PMDs (i.e., a CPU-bound process), the daemon first increases the voltage to the next safe V_{min} level and then, if the voltage can be decreased according to utilized PMDs (this information is provided by the monitoring part), the daemon will set the voltage accordingly. By following this policy, there is a minimal increase of the total power consumption, however, this guarantees the reliable execution on a real system. The Placement part uses the classification performed by the Monitoring part to guide its decisions, and is invoked upon every process list or classification change.

The online monitoring daemon is minimally intrusive and has no impact on the safe V_{min}. Its performance overhead is also negligible, as it is only running periodically to read the performance counters and upon every process list change, to invoke the placement process, which has equal impact as a process migration of the Linux kernel. The daemon is invoked only after

a. either a new process is issued to the system or when a process finishes its execution (to check if a process migration is required), or

b. when a process changes its state (from CPU-intensive to the memory-intensive and vice versa).

In case (a), it reads the process mapping table and estimates the result, and in case (b), it periodically counts the L3C accesses. Note that, in case (b) the utilized PMDs cannot be changed. Utilized PMDs can only be changed when a new process is invoked, or when a process finishes its execution.

For the purposes of our experimental evaluation, we also developed a "workload generator" which creates a typical server workload from a "pool" of programs (which includes all the 29 SPEC CPU2006 and the 6 NPB benchmarks; in total 35 different programs). The generator can generate workloads of configurable duration by randomly selecting benchmarks from this pool and randomly defining the timeslot in which each benchmark will be invoked. To provide detailed results for a comprehensive analysis, we ran 4 different configurations for the same workload sequence:

Fig. 9 shows the average power for a 1-hour execution of randomly generated workload for the default microprocessor's settings (Baseline) and the Optimal scheme, for and X-Gene 3.

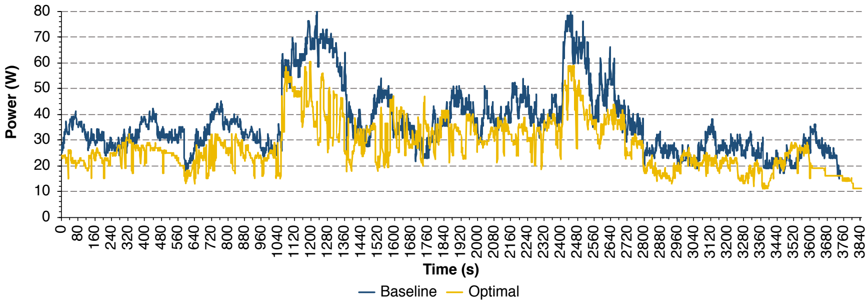

Fig. 9. Average power for the Baseline and the Optimal configurations in X-Gene 3 during 1-hour execution.

4 Conclusions

In this thesis, we presented two complementary methods to accelerate the post-silicon validation phase of modern microprocessors and to guarantee their energy efficiency. More specifically, we presented our contributions on post-silicon validation of the address translation mechanisms of modern microprocessors. We presented a comprehensive set of bug models, which correspond to the address translation mechanisms and classify the effects of both functional and electrical bugs in the hardware structures employed in address translation.

We then presented a detailed system-level voltage scaling characterization study for single-core executions in ARMv8-based multicore CPUs manufactured in 28nm. Towards the formalization of the behavior in undervolting conditions we also presented a simple consolidated function; the Severity function, which aggregates the effects of reduced voltage operation. Aiming to accelerate the time-consuming characterization process, we also introduced the development of dedicated programs (diagnostic micro-viruses) that aim to stress the fundamental hardware components of APM's X-Gene 2 micro-server family and provide quickly the safe Vmin values for each core. The final contribution of this thesis in the area of energy-efficiency was a detailed system-level voltage scaling characterization study for multicore executions in two recent ARMv8-based multicore CPUs manufactured in 28nm and 16nm.

References

1. F. Salehuddin, I. Ahmad, F.A. Hamid, A. Zaharim, A. Maheran, A. Hamid, P. S. Menon, H. A. Elgomati, and B. Y. Majlis, "Optimization of process parameter variation in 45nm p-channel MOSFET using L18 Orthogonal Array," In Proceedings of IEEE International Conference on Semiconductor Electronic (ICSE '12), 2012.
2. W. Schemmert and G. Zimmer, "Threshold-voltage sensitivity of ion-implanted m.o.s. transistors due to process variations," Electronics Letters, vol. 10, no. 9, p. 151, 1974.
3. N. James, P. Restle, J. Friedrich, B. Huott, and B. McCredie, "Comparison of Split-Versus Connected-Core Supplies in the POWER6 Microprocessor," in 2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers, 2007.
4. Y. Zu, C. R. Lefurgy, J. Leng, M. Halpern, M. S. Floyd, and V. J. Reddi, "Adaptive guard-band scheduling to improve system-level efficiency of the POWER7+," in Proceedings of the 48th International Symposium on Microarchitecture - MICRO-48.
5. I. Wagner, An Effective Verification Solution for Modern Microprocessors. PhD thesis, University of Michigan, 2008.
6. G. Papadimitriou, A. Chatzidimitriou, D. Gizopoulos, and R. Morad, "ISA-independent post-silicon validation for the address translation mechanisms of modern microprocessors," in 2016 IEEE 22nd International Symposium on On-Line Testing and Robust System Design (IOLTS), 2016.
7. G. Papadimitriou, A. Chatzidimitriou, D. Gizopoulos, and R. Morad, "An Agile Post-Silicon Validation Methodology for the Address Translation Mechanisms of Modern Microprocessors," IEEE Transactions on Device and Materials Reliability, vol. 17, no. 1, Mar. 2017.
8. G. Papadimitriou, D. Gizopoulos, A. Chatzidimitriou, T. Kolan, A. Koyfman, R. Morad and V. Sokhin, "Unveiling difficult bugs in address translation caching arrays for effective post-silicon validation," International Conference on Computer Design (ICCD), 2016.
9. G. Papadimitriou, M. Kaliorakis, A. Chatzidimitriou, D. Gizopoulos, G. Favor, K. Sankaran and S. Das, "A system-level voltage/frequency scaling characterization framework for multicore CPUs," IEEE Workshop on Silicon Errors in Logic - System Effects, 2017.
10. G. Papadimitriou, M. Kaliorakis, A. Chatzidimitriou, D. Gizopoulos, P. Lawthers, and S. Das, "Harnessing voltage margins for energy efficiency in multicore CPUs," IEEE/ACM International Symposium on Microarchitecture - MICRO-50, 2017.
11. G. Papadimitriou, A. Chatzidimitriou, M. Kaliorakis, Y. Vastakis, and D. Gizopoulos, "Micro-Viruses for Fast System-Level Voltage Margins Characterization in Multicore CPUs," IEEE International Symposium on Performance Analysis of Systems and Software, 2018.
12. G. Papadimitriou, A. Chatzidimitriou, and D. Gizopoulos, "Adaptive Voltage/Frequency Scaling and Core Allocation for Balanced Energy and Performance on Multicore CPUs," in 2019 IEEE International Symposium on High Performance Computer Architecture, 2019.