uniserver

D3.5 2nd Analysis of Processor Cores under Various Stress Conditions

Contract number	688540
Project website	http://www.uniserver2020.eu
Contractual deadline	Project Month 24 (M24): 31st January 2018
Actual Delivery Date	
Dissemination level	Confidential
Report Version	1.0
Main Authors	S. Das (ARM), Z. Hadjilambrou (UCY), Y. Sazeides (UCY)
Contributors	
Reviewers	Denis Guilhot (WSE), Arnau Prat (SPA)
Keywords	voltage noise, genetic algorithm, voltage margins, electromagnetic emanations, EM

Notice: The research leading to these results has received funding from the European Community's Horizon 2020 Programme for Research and Technical development under grant agreement no. 688540.

Disclaimer
This deliverable has been prepared by the responsible Work Package of the Project in accordance with the Consortium Agreement and the Grant Agreement Nr 688540. It solely reflects the opinion of the parties to such agreements on a collective basis in the context of the project and to the extent foreseen in such agreements.

© 2018. UniServer Consortium Partners. All rights reserved

Acknowledgements
The work presented in this document has been conducted in the context of the EU Horizon 2020. UniServer is a 36-month project that started on February 1st, 2016 and is funded by the European Commission. The partners in the project are:

The Queen's University of Belfast (QUB)
The University of Cyprus (UCY)
The University of Athens (UoA)
Applied Micro Circuits Corporation Deutschland Gmbh (APM)
ARM Holdings UK (ARM)
IBM Ireland Limited (IBM)
University of Thessaly (UTH)
WorldSensing (WSE)
Meritorious Audit Limited (MER)
Sparsity (SPA)

More information
Public UniServer reports and other information pertaining to the project are available through the UniServer public Web site under http://www.uniserver2020.eu.

Change Log

Version	Description of change

Table of Contents

Index of Figures

Index of Tables

Executive Summary

Voltage noise forces CPU designers to set pessimistic voltage and frequency margins. These margins limit the performance and energy efficiency of CPUs. The focus of this deliverable is a voltage noise characterization of low-power ARM enterprise server platforms that will broaden our understanding on power-delivery network behaviour. Central to this work is the ability to monitor voltage noise. Note that the current main Uniserver evaluation platform, the X-Gene2, does not support direct measurement of the CPU voltage measurements, which impedes our efforts to characterize CPU voltage noise. To address this limitation, we have proceeded with the following work directions:

A) We extended our analysis using the dual-core Cortex-A57 / Cortex-A72 Platform (internally at ARM and code-named as the Juno platform), which is equipped with an on-chip digital-sampling oscilloscope (DSO) circuitry that supports direct measurement of the power-delivery network. Particularly, we have developed a Linux driver of the on-chip DSO that is part of the platform. Using this driver, we have developed a system-level power-network characterisation framework that allows for applying system-level benchmark scenarios and collecting power-delivery network results. The analysis of the results can reveal hardware/software interactions.
B) We established a novel cross-platform approach for characterizing CPU Power Delivery Network voltage noise that leverages electromagnetic emanations. Compared to common voltage noise approaches i.e. on-chip monitoring, circuits external sense pins, this approach is not intrusive to the system as it does not require physical access on the CPU. We validate the efficacy of the EM approach through Vmin and direct voltage noise measurements on the Cortex-A72. And we show the generality of the approach in total on 4 CPUs (Cortex-A72, Cortex-A53, X-Gene2 and AMD Athlon II X645 x86 CPU).

This deliverable presents the combined efforts for both approaches outlined above. Utilizing the system level characterization framework, we analyzed the impact of the OS on the power-delivery network and we observed that voltage droop emergencies can be caused by system calls, daemons and system counters, which can be similar in magnitude to that of stress tests. Utilizing the EM methodology, we generate voltage noise viruses and determine the PDN 1st order resonant frequency on multiple CPUs of different uArch, ISA, frequency and technology nodes. The generated voltage noise stress tests have 20-75mV higher Vmin than conventional benchmarks, and, cause 30-100mV larger droops. Using the viruses' Vmin as nominal voltage indicator we can eliminate 37.5mV-150mV voltage margin depending on platform.

For future work, we plan to continue voltage margin characterization efforts on X-Gene hardware and other modern CPUs and GPUs.

1. Introduction

Voltage noise can occur due to different micro architectural events that cause sudden variation in CPU activity and power demands [1,2,6,7]. A large voltage drop can cause timing errors. In order to ensure reliable operation, CPU designers add up to 20% voltage margin for a given CPU frequency [1]. This pessimistic voltage and frequency margins limit the performance and energy efficiency of CPUs. A fundamental pillar of the UniServer Project is the analysis and characterization of worst-case supply voltage noise in high-end server-class processors. However, the current Uniserver project experimental platform, which is X-Gene2 CPU, does not offer any voltage noise visibility means.
Therefore, the consortium has made progress towards characterizing voltage noise in two ways:

a) We have developed an analytical system-level test framework for the formal description and application of tests that target the characterisation of interactions between the system and the power delivery network for the Juno platform. The framework is feasible due to the existence of an on-chip digital sampling oscilloscope (DSO) on this board that allows for a direct measurement of the power-delivery network.
b) We have developed indirect voltage-noise measurement methods to help us estimate on-chip voltage noise using off-chip electromagnetic (EM) emanations measurements [58]. We use the Juno voltage noise monitor to validate the methodology and we show the generality of the approach on 3 more CPU (Cortex-A53, X-Gene2 and AMD Athlon II X645 x86 CPU). We provide strong evidence of the general

applicability of the approach as we evaluate CPUs of different uArch, ISA, power requirements, market segments and technology nodes.

Central to this deliverable is the genetic algorithm (GA) framework for generation of voltage noise viruses. The framework successfully generates EM voltage noise viruses that, depending on platform, cause 20-100mV higher voltage droop, and have Vmin 20-75mV higher than conventional benchmarks. The viruses' Vmin guide us to eliminate 37.5-150mV of voltage margin depending on the platform. Furthermore, we propose a fast EM approach for detecting the 1st order PDN resonant frequency. The resonant frequency of the evaluated platforms is found around 66-150MHz.

Beside the power-network voltage noise characterisation, while running synthetic benchmarks or benchmarks that target single core or multi-core systems (Table 1), the system-level power-network characterization framework allows the execution of Operating System (OS) tests, such as the Linux Test Project (LTP) tests shown in Table I. The analysis of these tests has revealed many reproducible scenarios of system hardware/software interactions that induce voltage droops that sometimes can be higher than that of stress tests.

The rest of the deliverable is organized as follows: Section 2 presents the PDN fundamental. Section 3 presents the framework for system voltage noise analysis on Juno board. Section 4 presents the non-intrusive cross-platform EM methodology for PDN characterization. Section 5 concludes.

Table 1. Benchmarks details

synthetic benchmarks	
noise virus (resonant, resonant2)	maximizes voltage noise
power virus (maxpower)	maximizes power consumption
workload benchmarks	
SPEC06	single-core focused
NAS	Parallel multi-core focused
system benchmarks	
Linux Test Project (LTP)	kernel focused

2. Power Delivery Network (PDN) Fundamentals

Figure 1 (a) shows a simplistic representation of the PDN of a die-package-PCB system [6][43]. The current demand due to on-chip switching transistors is modelled as a lumped current source, I_{LOAD}. Explicit decoupling capacitors (henceforth, referred to as decaps) and non-switching, but powered-on, transistors act as localized charge reservoirs that provide the high-frequency component of the demand current, $I_{LOAD.}$ The on-chip power-grid resistance is modelled as a lumped resistor, R_{DIE}, connected in series with C_{DIE}. The total die current (I_{DIE}) is sourced through the inductive power-line traces of the package and the PCB, represented by a series R-L (resistor, inductor) equivalent circuit. The discrete decaps on the PCB and package are represented by an ideal capacitance (C_{PKG}, C_{PCB}) in series with its effective series inductance (ESL) and effective series resistance (ESR). Figure 1(b) shows the input impedance of the distributed RLC network as seen from the die. The impedance spectrum shows multiple resonance peaks due to multiple LC-tank circuits. The highest impedance peak, referred to as the first-order resonance peak is attributed to the die-capacitance (C_{DIE}) interacting with its counterpart inductance (LPKG). The first-order resonance also occurs at the highest frequency (50MHz-200MHz) compared to the second- (~1-10MHz) and third-order (~10KHz) resonances that are due to downstream capacitor networks.

The resonance frequencies also manifest in the time-domain when the PDN is excited by a step-current excitation (Figure 1 (c)). Power-supply oscillations of larger magnitudes can also be set off within the supply network due to sustained program activity with alternating periods of high-current and low-current consuming instructions within a loop [2][16]. When the frequency of the time-varying current aligns closely with the 1st-order resonance frequency, voltage oscillations are maximized in amplitude. High voltage oscillations can lead to bit-flips in SRAM storage arrays, timing errors in logic paths [1][2][7][16] and reliability issues due to

gate-oxide stress [7][8]. Such periodic events often result in system/application crashes and/or incorrect execution output through silent data corruptions (SDCs)[2][46].

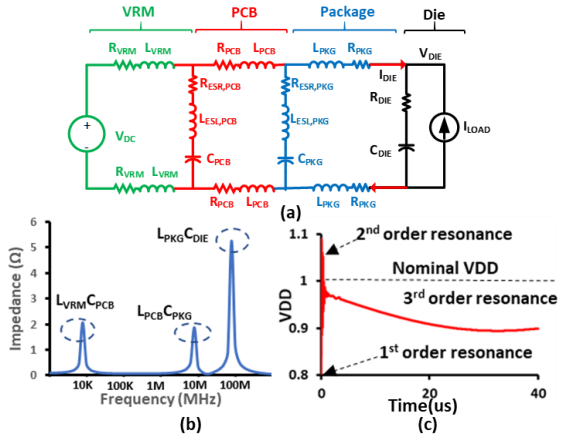

Figure 1. (a) A simplified model of the PDN [43]. The impedance as seen by the die has multiple resonance frequencies, shown in the frequency-domain response in (b) and time-domain response to a step-current excitation in (c)

However, it has been shown that micro-architectural events such as branch mispredictions [6] can cause high voltage droops similar in magnitude to droops induced by stress tests. As power noise is affected by infrequent combination of system and micro-architectural events, it is crucial to understand the role of the Operating System (OS). To achieve this goal a direct measurement approach is required that can be used for analysing the impact of system behaviour on the power noise. For this reason, we have implemented a Linux driver for the On-Chip digital sampling oscilloscope (DSO) of the dual-core Cortex-A57 / Cortex-A72 Platform (internally at ARM code-named as the Juno platform). The architecture of the driver is described in the following.

3. System-Level Interactions on Power-Delivery

3.1. The Digital Sampling Oscilloscope

A high-bandwidth on-chip digital sampling oscilloscope (OC-DSO) snoops the supply rails of the A57 cluster [5]. The OC-DSO (Figure 2) runs continuously in real-time, logging data and capturing waveforms on trigger events. Event counter and tide-mark registers track the size and frequency of voltage transients. For voltage transients of interest, threshold and gradient triggers can initiate waveform capture of up to 2K points into the internal SRAM trace buffer. A decimation block allows flexible bandwidth/sample rate to allow measurement of low frequency transients.

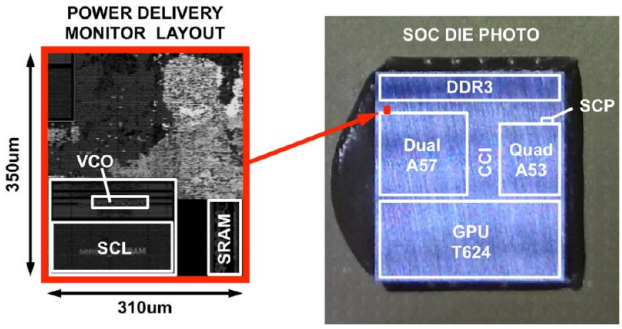

Figure 2. On-Chip DSO Based Power Delivery Monitor samples on-die voltage on the A57 cluster: Support for waveform capture of upto 2K points [1] enables correlation of simulation analysis

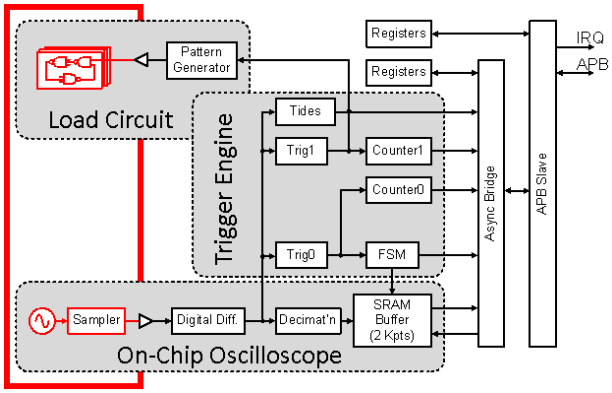

Figure 3. On-Chip DSO design

The trigger type instructs a scope whether to wait for a trigger condition or not and how to handle the arrival of such trigger. The trigger control subsystem of the utilized oscilloscope consists of blocks that are designed to provide flexibility on triggering the sensor. Auto-triggers, event counters and time stamp blocks also allow points of interest to be monitored, timed and counted. The utilized on-chip digital sampling oscilloscope has the following trigger types:

- Auto-Trigger: the auto-trigger is included to allow the device to automatically search for points of interest. Two auto-triggers are included, one threshold based and the other based on max transient. Both are flexible and have various settings for rising/falling/either transitions, programmable thresholds etc. The auto-trigger is also routed through the event counter, so that the second or third event etc. can be captured (essentially trigger inhibit).
- Event-counters: there are two event counters that can be used to count auto-trigger events. It utilizes a comparator with programmable threshold so that they can also generate triggers for capture.
- Time-stamp: the time-stamp units are essentially PCLK counters that can be used to determine at what count value a trigger or event counter occurred. They are flexible and can be stopped by signals from various sources.

3.2. DSO Driver Architecture Implementation Details

In the following, we present the software architecture, shown in Figure 4, of the developed driver for the on-chip digital sampling oscilloscope (OC-DSO) of the core Cortex-A57 / Cortex-A72 Platform. The A57 dual-core cluster, shown in Figure 1(a), consists of two ARM A57 cores and one on-chip digital sampling oscilloscope sharing the same power network.

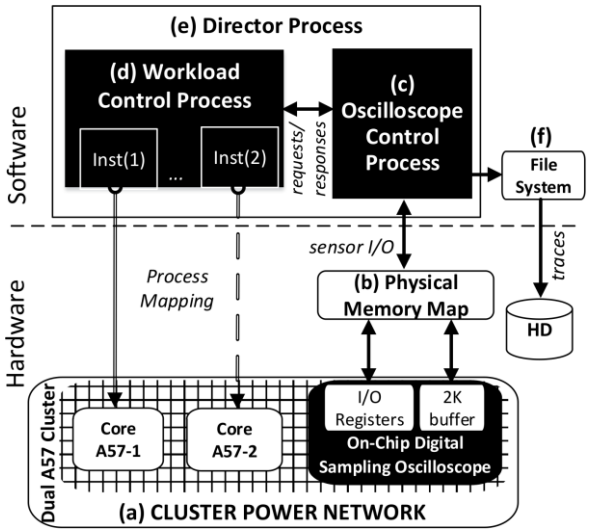

Figure 4. (a) A57 dual-core cluster with its power network and the on-chip digital sampling oscilloscope; (b) physical memory map of the oscilloscope I/O registers and the buffer of 2K traces; (c) process that drives the oscilloscope; (d) process for controlling the workload; (e) director process that is used for software-based triggering mode; (f) file system for storing time-domain traces

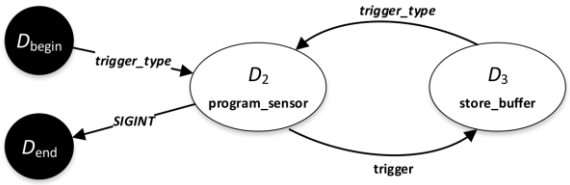

Figure 5. DSO Control Process

The I/O registers of the oscilloscope and its 2K measurements buffer are mapped on physical memory as shown in Figure 4(b). A control process, named hereafter Oscilloscope Control Process (OCP), is responsible for controlling the triggering types of the oscilloscope and the collection of the time-domain voltage measurements by accessing the buffer through the memory map. The collected measurements are then stored at the file system, shown in Figure 4(f). To control the execution of benchmarks on the cluster, the proposed architecture has a Workload Control Process (WCP), shown in Figure 4(d) that controls the process mapping on the cluster. The whole architecture is controlled by the Director Process, or simply denoted as director, is also shown in Figure 4(e). The director process can control both the workload and the oscilloscope triggering modes enabling an additional software-based triggering mode. These processes are discussed in detail in the following.

The Oscilloscope Control Process (OCP) is responsible for configuring the operating hardware-based triggering types of the oscilloscope and the collection of the time-domain voltage measurements. The FSM of this process is shown in Figure 5. Initially, the trigger type signal alerts the OCP to program the sensor with the corresponding hardware-based triggering type and its configuration. The hardware-based triggering types, namely the MANUAL, the THRESHOLD and the PCLK have already been discussed in Paragraph 3.1. The OCP is also responsible for collecting the time-domain voltage measurements after triggering by accessing the 2K size oscilloscope buffer. The collected measurements are stored in the file system, shown in Figure 4(f).

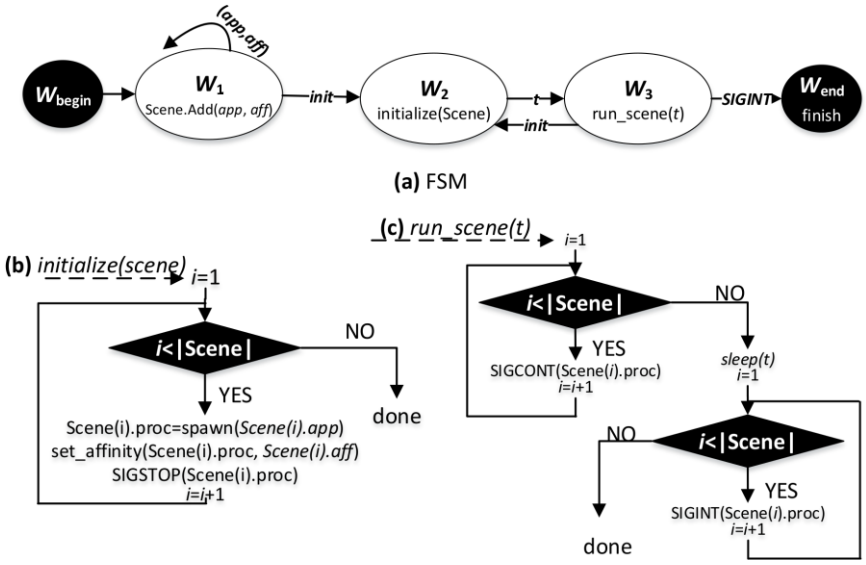

Figure 6. Workload Control Process

Micro-architectural and system events often cause abrupt changes in current demand, leading to inductive transients that stress timing guardbands. Manually creating workloads that can trigger worst-case resonances in the system is difficult due to the complexity of the underlying micro-architecture, especially in out-of-order cores, such as the ARM A57. We circumvent this issue by automatically generating worst-case workloads using a genetic-algorithm based framework [2] that is agnostic to the processor micro-architecture. Details on the genetic algorithm are presented in D3.2 2nd Analysis of Processor Cores under Various Stress Conditions.

To control the execution of benchmarks on the cluster, the proposed architecture utilizes the Workload Control Process (WCP), shown in Figure 4(d), that controls the registration, the initialization and the execution of benchmarks on the dual-core cluster. The FSM of the WCP is shown in Figure 6(a). Initially, the WCP gets into the W_1 state, in which it accepts pairs of the application *app* and an affinity configuration *aff*. Each accepted pair is included to the current scene, which is a vector of pairs. Then upon the assertion of *init* the WCP initializes the application by preparing their execution and setting their affinity as instructed by the

scene pairs. The flow diagram of the WCP initialization is shown in Figure 6(b). Note that the initialization ends by setting the application in sleep state by sending them a SIGSTOP signal. Then, upon the assertion of the *t* signal, the WCP wakes-up the applications. The flow diagram of the WCP run_scene is shown in Figure 6(c). A SIGCONT signal is send to every application contained in the scene to continue their operation. After time of *t* is elapsed the applications are instructed to terminate by sending them a SIGINT signal.

The director process, or simply denoted as director, has the joint overall high-level supervision of the WCP and the OCP and provides an API to the user for initializing these processes. The API is available in Shell Script and the Computer Language C. The user can register applications, initialize workload scenarios and control the triggering modes selection of the oscilloscope by using the API of the director.

3.3. Longer Time-domain Traces

Algorithm 1 Software Trigger Algorithm

Inputs: t, d, w: time, duration, sliding window
app[N], aff[N]: applications and affinities for scene creation

Output: $V(t)$: time-domain votalge

```
WCP=WorkloadControlProcess(start)
for(i = 1 : N) do
    WCP(app[i],aff[i])
done
windowsNo=d/w
for(j = 1 : windowsNo) do
    ToI=t + w · i
    {//also stores V(ToI : ToI + w) traces at medium}
    OCP=OscilloscopeControlProcess(trigger_type=pclk, conf=ToI)
    WCP(init)&
    trap WCP(ToI+20ms)
done
WCP(SIGINT)
OCP(SIGINT)
```

Algorithm I: Software-based triggering algorithm

The director has a build-in algorithm, shown in Algorithm I, which consists of an additional triggering mode, the software-based triggering mode. This mode enables the replication of operating conditions by carefully orchestrating the WCP and the OCP, in contrast to previous hardware-based triggering modes that enable only the collection of time-domain traces of few microseconds, . At the same time, it controls a sliding window of PCLK based hardware triggering by configuring the OCP. This way, it allows for the collection of longer time-domain traces, at the order of hundreds of milliseconds, and enables a system level analysis of the root architectural events that impact the system power integrity.

3.4. Voltage noise statistics

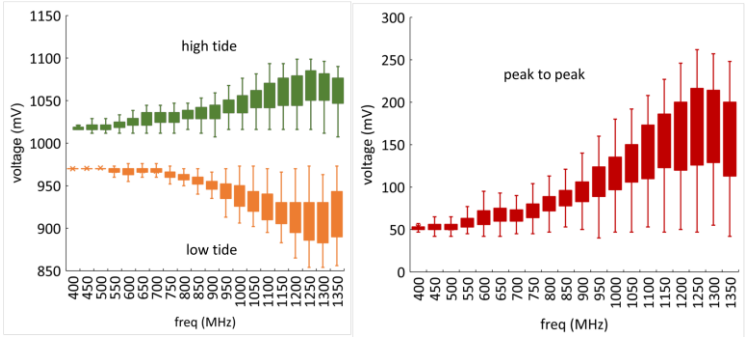

Figure 7. Low/High tides and peak to peak voltage swing of the resonant benchmark when sweeping operating frequency

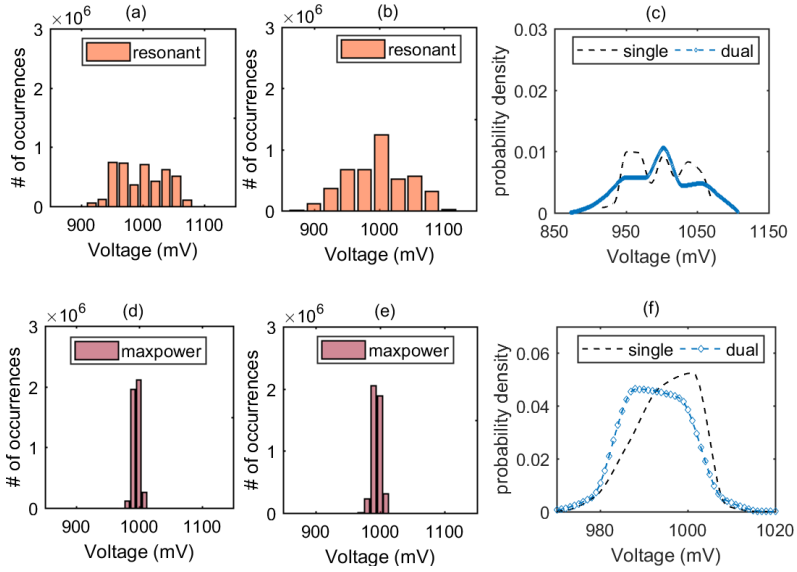

Figure 8. Voltage density of synthetic benchmarks and probability density functions

The following statistical measurements are collected using data provided by the sensor:
Low/high-tides: that is the lowest and the highest voltage measurements observed. Tides can be reset by the OCP and are collected during any triggering types. In Figure 7 the low-tide Vmin and the high-tide Vmax for two synthetic benchmarks are shown, when they are executed for t=60 secs. The *resonant* benchmark is a voltage noise virus and the *maxpower* is a power consumption virus. Interestingly, we observe that the tides drop lower after the execution of the benchmarks. This observation is general for all benchmarks and power noise viruses. There is a very high droop that follows the ending of execution of most of the examined benchmarks, which as we will show later, is related to the operation of context switch that takes place.

Voltage probability density, mean voltage *m* and standard deviation *s*: the voltage variability of the power rail is evaluated by collecting many buffers and applying statistical clustering on the observed voltage values. This is conducted by clustering the voltages to bins on the *x*-axis. As an example, in Figure 8 the voltage variability for the *resonant* and the *maxpower* benchmarks are shown, which has been evaluated using buffers collected after running the benchmarks for 1 hour and manual trigger configuration for the oscilloscope. Note that just few thousands of buffers are enough to get a similar result. The probability density of the bins is evaluated as the ratio of the number of occurrences of each bin to the total number of voltage samples. A linear fit on the probability density is shown. We observe that the probability density of the *maxpower* benchmark is very close to a normal distribution, while the probability density of the resonant is not, as it exhibits three hill peak points. Nevertheless, based on this experiment the standard variation s of the voltage is also evaluated.

We have collected these statistics for all benchmarks. As an example, Figure 9 depicts the voltage densities of bwaves benchmark from the SPEC suite and the *sp* benchmark with input data set C from the NAS suite. In Figure 9c, the low and high tides are also marked together with the observer voltage probability density (line) as well as the variability modelled by a normal distribution using a mean voltage *m* and variability s. As expected, we observe that the variability is lower compared to the resonant benchmark, and that there is a dc swift of the noise when two cores are used. The dual-core case exhibits higher dc voltage shift, because it consumes more power. We also observe that both the single and dual-core cases exhibit similar voltage noise swings, which is attributed to the fact that it is almost impossible to synchronize the execution of the two benchmarks on the cores at the system level.

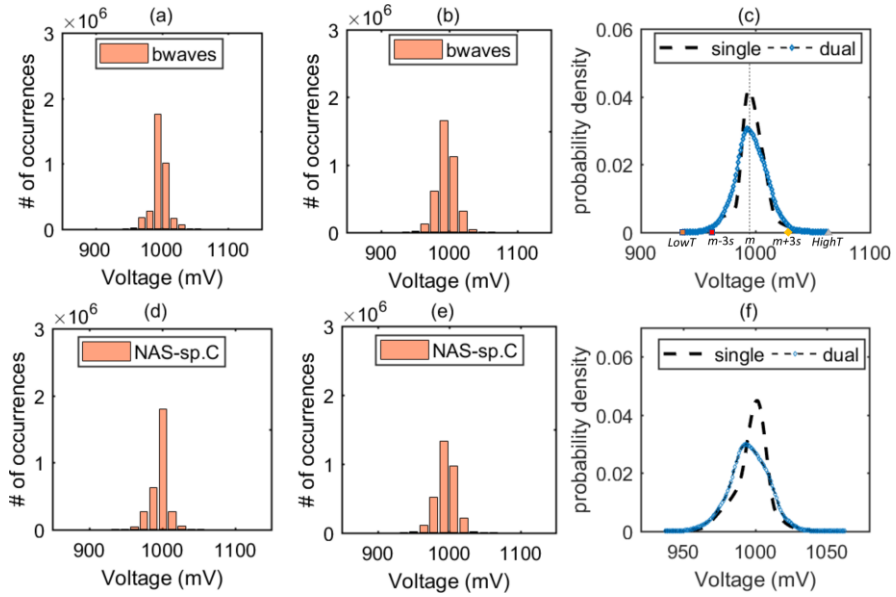

Figure 9. Voltage probability density of 1 SPEC and 1 NAS benchmark and probability density functions: (a)(d) single-core execution; (b)(e) dual-core execution; (c)(f) probability densities; (a)(b)(c) bwaves benchmark from SPEC suite; (d)(e)(f) sp benchmark with input set C from NAS suite

The voltage variability statistics for all the examined benchmarks (synthetic, SPEC and NAS) are presented in Figure 10 for single-core (Figure 10a) and dual-core cases (Figure 10). Note that the voltage noise of the synthetic viruses (resonant2, resonant – the difference of these is that resonant2 targets to maximize the single core noise and resonant the dual) exhibit much higher noise variability than any other benchmark.

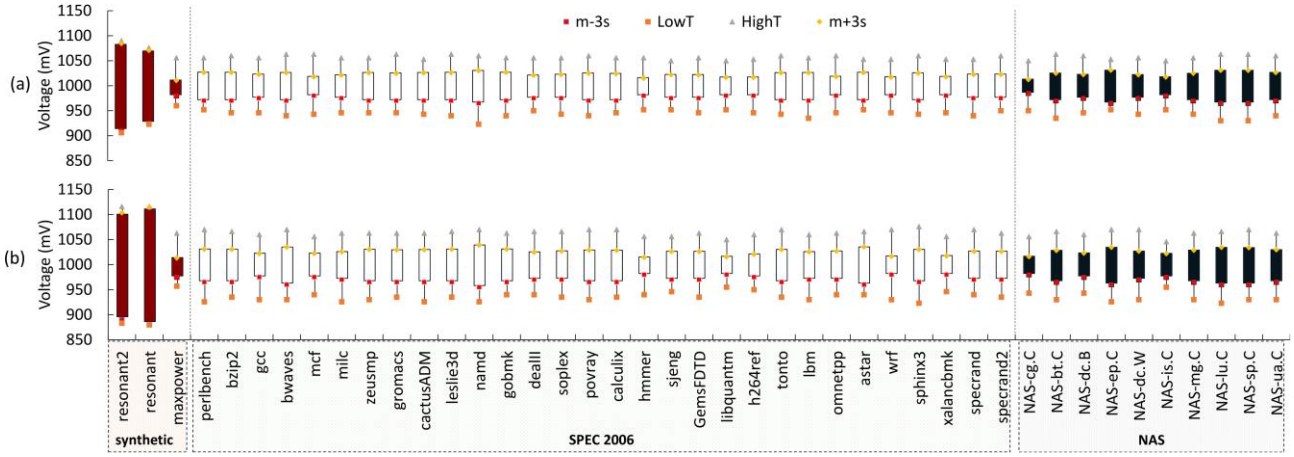

Figure 10. Voltage variability statistics for synthetic, SPEC06 and NAS benchmarks: (a) single-core; (b) dual-core

The voltage variability is also collected for Linux system calls using the Linux-Test-Project (LTP) infrastructure and are presented in Figure 11. Interestingly, we observe that there are system-calls, such as

fchown, llseek, clone etc., that and occasionally exhibit voltage swings similar or even higher than that of power noise viruses. We are investigating the reason for this phenomenon. A spectral analysis of these droops has revealed max energy at 350Mhz, which matches to that of on-chip PLLs. We also observe that there are system calls, such as those related to privileged scheduling, virtual memory and process control, that exhibit systematically high voltage droops.

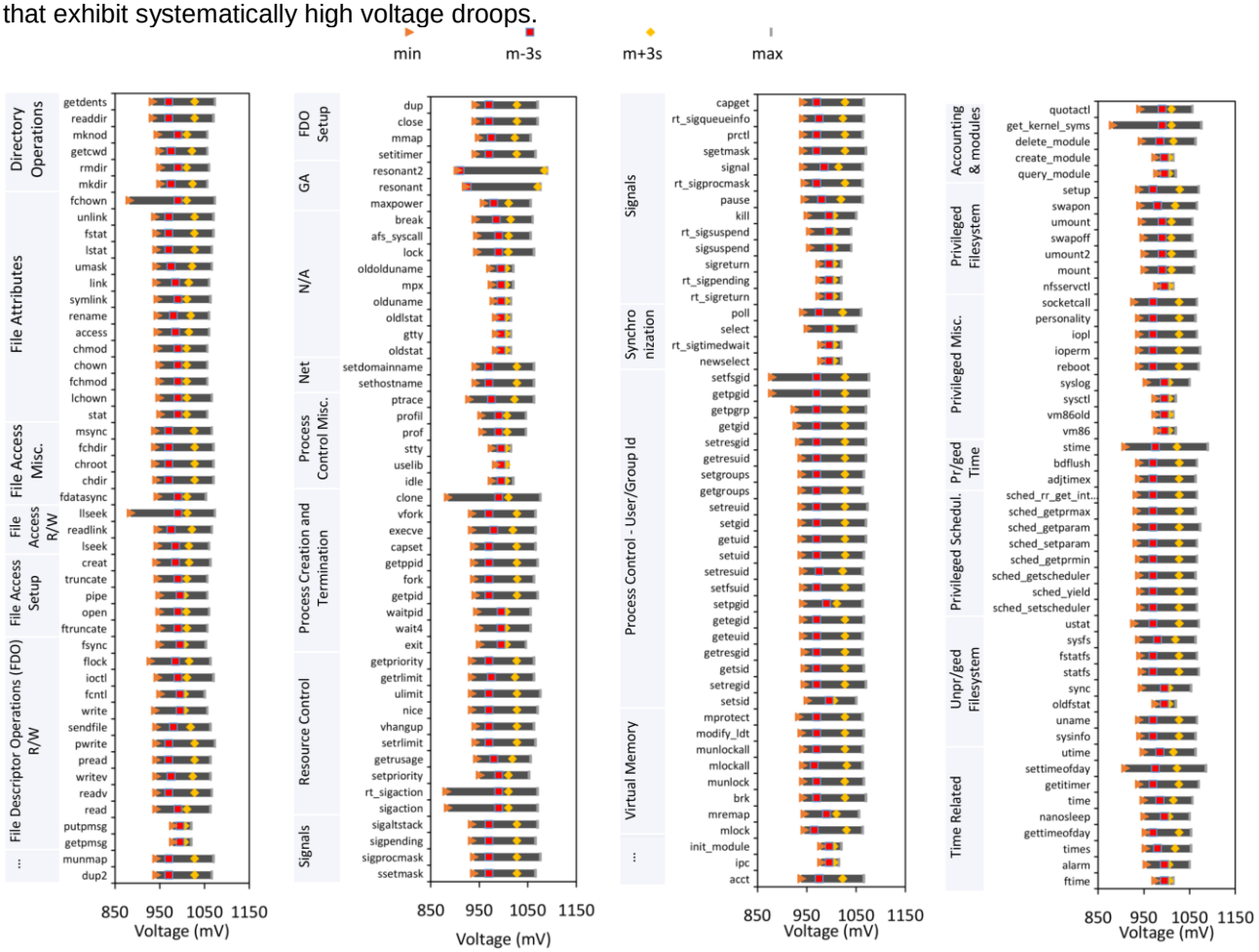

Figure 11. Voltage variability statistics for system calls

Another statistical value that shows potential and can be collected using the sensor is the time-between emergencies (TBE) rate is presented in Figure 12: the TBE rate is the distribution of the time elapsed between two successive voltage emergencies (triggers). A voltage emergency is defined as a trigger based on a threshold trigger type.

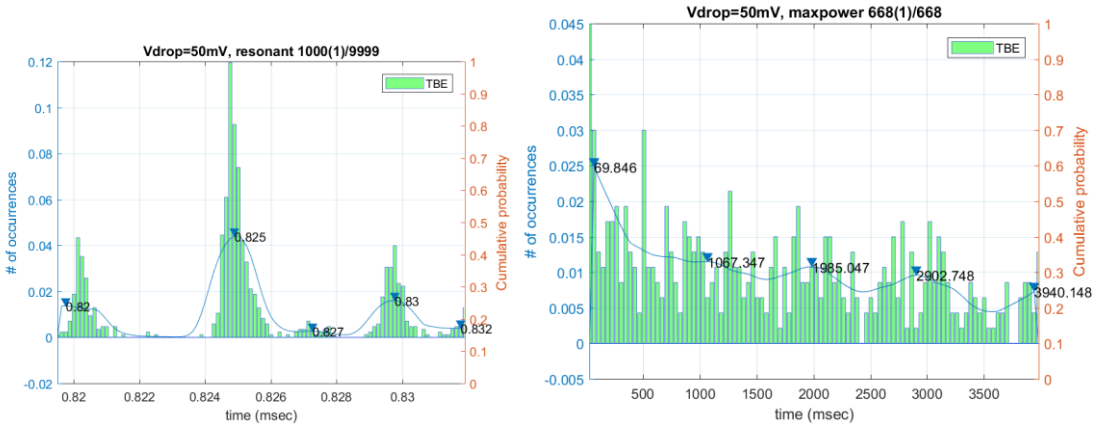

Figure 12. Time-between emergencies rate for resonant and maxpower benchmarks for trigger threshold voltage of 950mV, which corresponds to droops of 50mV

The OC-DSO is also equipped with counters that register the total number of triggers elapsed after their reset. The values of these counters can be used for computing the rate of triggers, which is simply called as droops rate hereafter. As the rate of droops also depends on the operating frequency, we use the formula Dr=C/(t*f), where Dr the droops rate, C the triggers counter value, t the elapsed time and f the operating frequency. Figure 13 depicts the droops rate for the resonant, maxpower and bwaves benchmarks. We observe that for an operating frequency of 1.1GHz the resonant benchmark exhibits the highest rate, which interestingly matches the operating frequency for the highest peak to peak voltage swing.

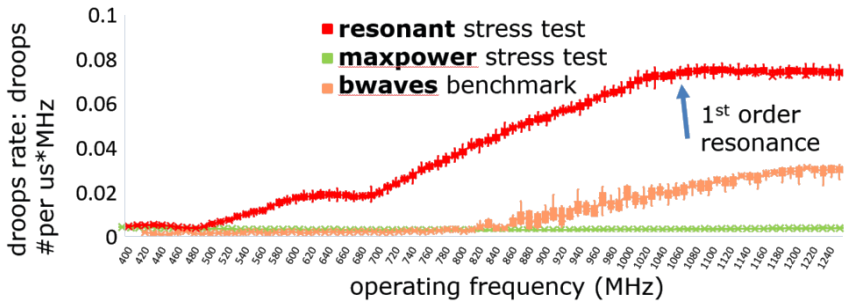

Figure 13. Trigger droops rate per MHz for resonant, maxpower and bwaves benchmarks

Next, we use the software-based trigger mode to collect a longer sequence of time-domain voltage series. In Figure 14, we present the time-domain for a context-switch, which is considered one of the most challenging tasks of a processor. We see that there are high droops, which we have correlated using events from the perf tool of Linux, with system calls. Figure 14 depicts the highest of those droops. Interestingly, the frequency domain of that droop also exhibits the highest energy at 350Mhz.

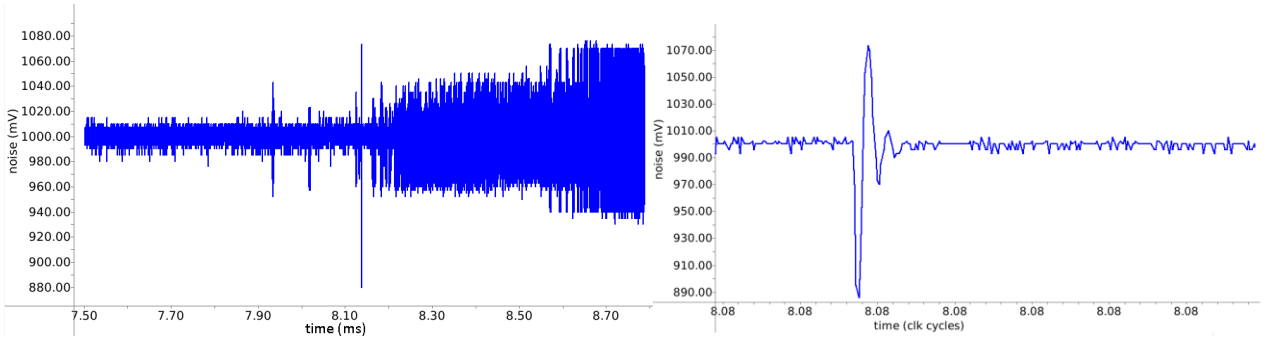

Figure 14. Software-based triggering on a context switch.

4. EM Methodology

The rest of the deliverable presents the EM-based methodology for power delivery network characterization. The EM discussion starts with an explanation of the theory that explains how EM emanations are linked with on-chip voltage noise.

4.1. Theoretical Link Between On-Chip Voltage Noise and EM Emanations

It is well-known that metallic conductors act as transmitting antennae that emanate EM radiation under oscillating voltage and current stimulation [17][20]. On-chip interconnections and transistors act as distributed radiating antennae due to time-varying current consumption induced through normal program execution. Simple periodic activity, such as that due to instruction loops, cause period variations in CPU power (i.e.

sequence of DIVs followed by ADDs) that manifest as visible spurs in the EM spectrum, at a frequency F equal to 1/T (where T is the loop period) [9].
Fundamental antenna theory (say, for a traditional Hertzian dipole) states that the component of the radiated power for the transmitting antenna, at a specific frequency, varies quadratically with the amplitude of the oscillating feed current [20] at the corresponding frequency and the so-called radiation resistance[1]. Periodic current load (I_{LOAD}), pulsing at the first-order resonance frequency, can trigger sustained oscillations of large magnitude in V_{DIE} and I_{DIE}.

We simulate the simplified PDN model with a persistently pulsing current excitation (I_{LOAD}) at 80MHz which matches the 1st-order resonance frequency (Figure 1 (b)). This sets of resonant oscillations in the PDN as illustrated by HSPICE [57] simulations in Figure 15. At resonance, both voltage and current oscillations maximize in amplitude. This, in turn, maximizes the radiated EM power from the on-chip distributed antennae, due to the quadratic dependence with oscillatory current amplitude. It is this relationship between radiated EM power and on-chip voltage noise that we leverage in this work to measure the first-order resonance frequency of the PDN. Simply put, measuring the frequency at which the amplitude of the emanated EM power is maximized directly reveals the 1st-order resonance frequency.
We validate the above theory that links CPU EM emanations with on-chip voltage noise using the ARM Juno [13]. This platform supports fine-grained, voltage-noise measurements in the time-domain. The measurements (described in Section 4.3.) confirm that a) emanated EM power is maximized at the first-order resonance frequency and b) maximization of EM power is strongly correlated with higher amplitude of power-supply oscillations.

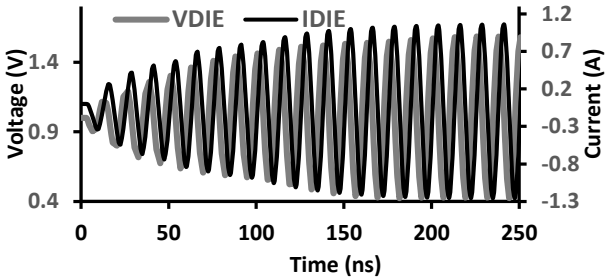

Figure 15. Simulated waveforms showing the die voltage (VDIE) and die current (IDIE) in the simplified PDN model in Fig. 1. ILOAD triggers the first-order resonance by pulsing at 80MHz. This causes both VDIE and IDIE to undergo large-magnitude oscillations, maximizing the radiated EM power.

4.2. EM Methodology Experimental Details and GA Framework

Table 2 shows an overview of the ARM and the AMD platforms used in this study. The ARM Juno [13] platform hosts a heterogeneous multiprocessing System-on-Chip (the so-called big.LITTLE configuration) consisting of separate clusters of the dual core Cortex-A72 and a quad core Cortex-A53 [42]. The platform integrates an on-chip power-supply monitor configurable as a digital storage oscilloscope (OC-DSO) [5] that is ideal for validating our proposed EM methodology. The OC-DSO provides fine-grained sampling of the voltage rails supplying the dual-core Cortex-A72 cluster. The Cortex-A53 cluster is in a separate voltage domain without explicit support for voltage-noise measurement. The Juno board runs a Debian OS with a 4.4.0-135-arm64 kernel. The DS-5 debugger [26] is used to access OC-DSO, sweep CPU frequency, change supply-voltages and power-gate both the Cortex-A72 and Cortex-A53 clusters, orchestrated through a system control processor (SCP) that enables this functionality [5].

For the AMD setup, an Athlon II X4 645 CPU is used that is hosted on an ASUS M5A78L LE motherboard and Windows 8.1 OS. AMD Overdrive application [25] is used to change the voltage and the frequency of the CPU. This application also includes a stability test that is evaluated and compared against the GA generated dI/dt viruses. The motherboard integrates on-package Kelvin measurement pads that enable direct external monitoring of the on-chip voltage rails using differential probes connected to a bench-top oscilloscope.

[1] The radiating resistance of a conductor can be differentiated from its loss resistance, in that the former is a function of the geometry of the conductor and determines the magnitude and the directivity of the radiated power [20]. The loss resistance, in contrast, manifests as ohmic losses dissipated through the conductor.

X-Gene2 is the Uniserver main platform and to change the voltage and frequency of the CPU, we use the Hardware Exposure Interface (HEI) (Task 4.1).

Figure 16 shows both the ARM Juno and AMD desktop PC experimental setups. We use a square loop antenna (3 cm side length) as a receiver for the emanated EM radiation. We measure the frequency response of the antenna to monitor for self-resonance frequencies in the range of interest (50 MHz – 200 MHz). Figure 17 shows the single-port scattering parameter (S11) measurement of the antenna for a wide-frequency range [56]. The antenna has a relatively flat frequency response from DC until 1.2 GHz, with a self-resonance frequency at 2.95 GHz. Thus, we confirm that the antenna does not modulate the received signal in the frequency range where we expect the first-order resonance frequency of the PDN to lie (50 – 200 MHz). Furthermore, even though the antenna is not well matched in the frequency range of 50 – 200 MHz, it is still able to receive the emanated EM radiation in close proximity to the CPUs. The antenna is connected to a spectrum analyser through a low-loss coaxial cable to receive the emanated waves from the experimental platforms. The spectrum analysers Agilent E4402B (Juno setup, X-Gene2 setup) and Agilent N9332C (AMD setup) are used to measure the EM signals. The antenna is placed at a stable position 5-10cm close to the monitored CPUs. We record strong EM signals on either side of the PCB, but prefer the lower side due to proximity to the die.

Moreover, a crucial part of the EM methodology is harnessing the EM emanations for generating a dI/dt virus. In this work we use Genetic Algorithms (GA) to generate dI/dt viruses. The GA converges to dI/dt virus by performing an optimization towards maximizing the EM amplitude in the range where the PDN 1st order resonant frequency typically lies (~50-200MHz). More information about the GA framework are presented in D3.2 1st Analysis of Processor Cores under Various Stress Conditions.

Figure 16. Experimental setup for the ARM juno board (left), AMD desktop CPU (center) and X-Gene2 (right).

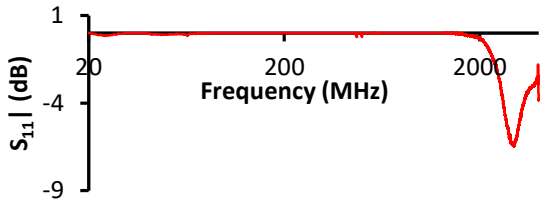

Figure 17. Measured |S11| for the square loop antenna indicating a self -resonance around 2.95 GHz.

Table 2. Experimental platform details.

MB	CPU	# of Cores	ISA	uArch	Highest Freq,Vol Point	Technology (nm)	OS	Voltage noise visibility
Juno Board R2	Cortex-A72	2	ARM V8	Out of Order	1.2GHz,1V	16	Debian	OC-DSO
Juno Board R2	Cortex-A53	4	ARM V8	In-Order	0.95GHz,1V	16	Debian	None
Asus M5A78L LE	Athlon II X4 645	4	x86-64	Out of Order	3.1GHz,1.4V	45	Windows 8.1	On-package pads
Tigershark Validation Board	X-Gene2	8	ARM V8	Out of Order	2.4GHz,0.98mV	22	Centos 7.2	None

4.3. EM Methodology Validation and Evaluation on Cortex-A72

4.3.1. Voltage Monitoring using the Juno OC-DSO

This section demonstrates the OC-DSO capability to capture voltage noise. OC-DSO monitors the supply rails of the Cortex-A72 cluster on the Juno platform. Figure 18 shows captured voltage waveforms for 3 different scenarios, namely a) operating-system idle b) during typical benchmark execution (SPEC2006 gcc) and c) during dI/dt virus execution (virus is obtained using GA search with feedback from OC-DSO). As expected, the virus execution causes by far the largest voltage noise. We next utilize the OC-DSO to correlate high voltage noise with high EM signal amplitude and validate the EM methodology.

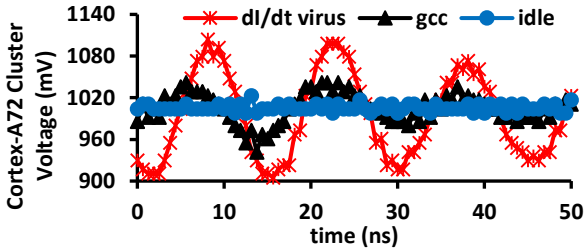

Figure 18. Voltage waveforms obtained by OC-DSO for 3 different workloads.

4.3.2. First-order Resonance Frequency Measurement

This section determines the Cortex-A72 cluster PDN 1st order resonant frequency using synthetic current load [16]. The results of this section serve as the golden reference for the Cortex-A72 PDN 1st order resonant frequency. The effectiveness of the EM approach in detecting the resonant frequency will be validated by comparing against the results of this section.

A description of the measurement procedure follows. The OC-DSO integrates a synthetic current load (SCL) block that can load the Cortex-A72 PDN with a known current stimulus. By measuring the response of the OC-DSO, it is possible to measure the first-order resonance frequency with a 1MHz frequency resolution. Using the methodology described in [16], we configure the SCL to load the PDN with a square-wave current excitation of known frequency. We sweep this fundamental frequency in the range between 10MHz and 130MHz (in steps of 1MHz) and record the maximum peak-to-peak voltage oscillation generated in response, at each frequency step. The basis of this characterization methodology is the observation that the voltage oscillation amplitude is maximized at the first-order resonance frequency. The results, plotted in Figure 19 reveal the first-order resonance frequency to be in the range between 66-72MHz (we observe a relatively flat frequency response around resonance) when both cores in the cluster are powered up (indicated by the label “C0C1” in the plot).

Moreover, Figure 19 illustrates the effect of power-gating an individual core in the cluster, on the first-order resonance frequency. The die capacitance of the power-gated cluster is disconnected from the PDN, leading to a net reduction of the total capacitance available. This is manifested by an increase in the first-order resonance frequency, as shown in the plot labelled “C0”. The resultant resonance frequency is in the range between 80-86MHz.

For the remainder of the paper, we consider a frequency in the range between 66-72MHz as the first-order resonance frequency for Cortex-A72 (the rest of experimental results are with both cores powered up). It is worth noting that the dominant frequency of the voltage waveform for the GA generated virus in Figure 18 is 66MHz. This underlines the effectiveness of GA to identify the resonance frequency

Note that the SCL block is a specific feature only available to the Cortex-A72 cores in the Juno platform and cannot be used to analyse other voltage domains.

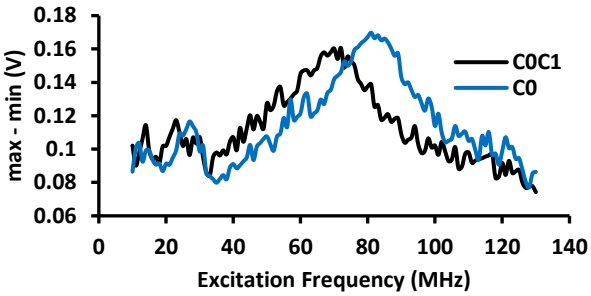

Figure 19. SCL stimulus reveals a resonant frequency in the range of 66-72MHz with two powered cores (C0C1) and 80-86MHz with one powered core (C0).

4.3.3. EM Emanations and Voltage-Noise Correlation

A GA search is performed on the Cortex-A72 with target to maximize EM amplitude (in the 1st order resonant frequency range of 50MHz-200MHz). Figure 20 shows how the EM amplitude and dominant frequency of the strongest individual (with "individual" we refer to a sequence of assembly instructions, each GA generation consists of 50 individuals) of each generation varies as the GA progresses. The figure also plots the maximum voltage droop caused by the strongest individual per generation (we obtain the droop using OC-DSO by re-running and measuring each individual after the GA search has finished). It is clearly seen that as the signal amplitude increases from generation to generation during the GA search the voltage droop increases as well. Therefore, it is safe to say that the GA search driven by EM signal amplitude essentially maximizes voltage noise. Furthermore, we observe that from the very first generations, the GA prefers individuals that have a dominant frequency in the range between 66-72MHz (the first-order resonance frequency range). Thus, we provide strong evidence that voltage noise and EM signal amplitude are both maximized at the resonant frequency.

Confirming the EM amplitude and voltage noise correlation further, we obtain the frequency-domain representation (using the Fast Fourier Transform (FFT) algorithm) of the voltage samples from the OC-DSO while executing the EM dI/dt virus. Figure 21 compares the spectrum analyser readings of the EM power captured by the receiver antenna with the FFT of OC-DSO voltage readings. The dominant frequency of both frequency-domain representations is exactly aligned at 67MHz, where we expect, according to Section 4.3.2, the first-order resonance frequency of the cluster to lie. Moreover, the two instruments agree on other less dominant spikes as well, such as the virus's base loop frequency (1/loop period) located at 16.66MHz.

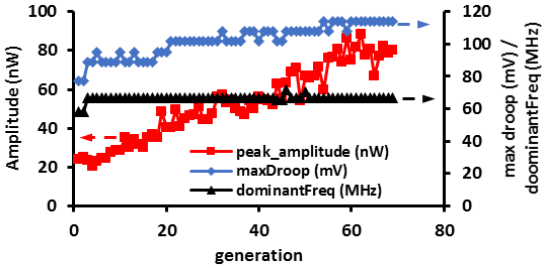

Figure 20. EM driven GA run on Cortex-A72. Peak amplitude (left axis) and maximum droop / dominant frequency (right axis) for the best individual of each GA generation.

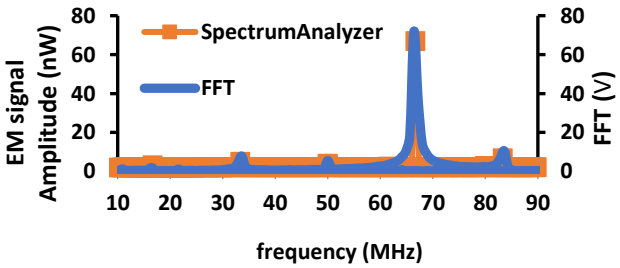

Figure 21. Comparison of spectrum analyzer readings (left axis) with FFT of OC-DSO voltage readings (right axis) during execution of EM dI/dt virus. The two measurements agree as they reveal spikes at the same frequencies.

4.3.4. V_{MIN} Tests On Cortex-A72

The virus generated in the previous section must ultimately limit the stability of the overall system due to the magnitude of voltage oscillations it generates. We quantify system-stability due to a workload by measuring the minimum operational voltage (V_{MIN}) at which the workload is executed correctly. Figure 22 compares the V_{MIN} of the EM virus against that of the SPEC2006 benchmarks and the V_{MIN} of a virus generated by the GA framework when optimizing for maximum voltage droop measured by the OC-DSO. All workloads are executed on both the Cortex-A72 cores, with each core running a separate instance of the workload. Each experiment is started at a high voltage and the voltage is progressively lowered in steps of 10mV until a system crash is observed. The workloads are run until completion and then the output is checked for SDC (by comparing the output against a golden reference obtained at nominal operating voltage of 1.0V). Figure 22 reports the highest voltage at which any deviation from the nominal execution is observed, either due to a SDC, an application crash or a system crash. We have observed (not shown in figure) that, typically, workloads suffer SDC or application crash approximately 10mV above the system crash. Both EM and OC-DSO viruses clearly cause higher voltage droop (in excess of 25mV compared to the "lbm", the SPEC benchmark with the highest voltage droop) and have higher V_{MIN} compared to the other workloads (20mV higher V_{MIN} compared to "lbm"). Both viruses (generated by targeting EM power or maximum voltage droop) stress the PDN in approximately similar manner.

This result also supports the claim that EM-driven GA is a feasible and reliable method for generating dI/dt viruses. For statistical confidence in our measurements, we perform 30 V_{MIN} tests for each virus and two V_{MIN} tests for each SPEC benchmark. SPEC benchmarks are executed with reference inputs, and, therefore, total V_{MIN} experimentation time is equal to about two days. Thereby, the SPEC benchmarks run for significant amount of time at voltages lower than the viruses' V_{MIN} without any failure.

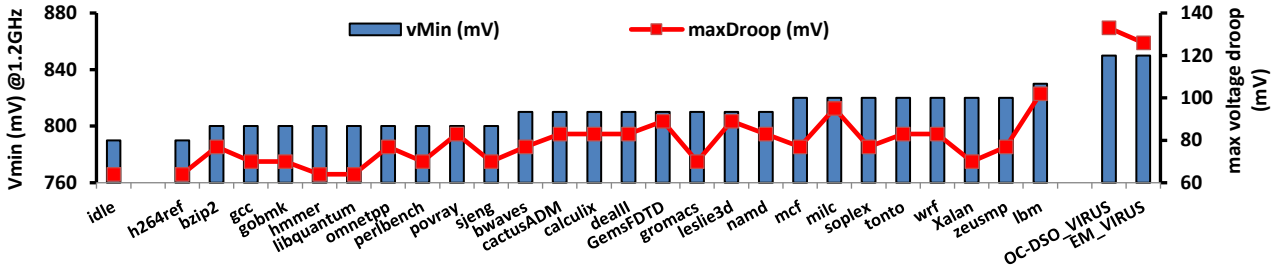

Figure 22. V_{MIN} (blue bar, left axis) and maximum voltage droop (red curve, right axis) of various workloads for dual core runs. Viruses (rightmost workloads) cause higher droop and have higher VMIN than typical benchmarks.

4.3.5. EM Methodology for Quickly finding PDN Resonant Frequency

As we show in the previous sub-sections, the GA framework is an effective approach for obtaining the resonant frequency and maximizing voltage-noise at that frequency, based only on external EM readings. However, the algorithm requires multiple generations for convergence and may require ~24 hours to terminate in our setup. In the case where the goal is only to determine the first-order resonance frequency (without the need for generating a virus), we propose an alternative EM based method that is extremely expedient, comparatively.

We first manually design an instruction loop composed of separate high and low current consuming sequences. We then sweep the CPU frequency which consequently modulates the loop frequency. Relative to the baseline, voltage noise is maximized when the frequency at which the loop-current fluctuates, due to alternating high and low-power consuming sequences, aligns with the resonant frequency. Thus, we expect that the highest voltage-noise may not necessarily correspond to the highest CPU frequency, rather, it is likely to manifest at a lower value where the fluctuating loop-current triggers resonant oscillation in the PDN [6].

In this specific case-study, we used a loop with the high current consuming sequence consisting of eight ADD instructions that are executed in 4 CPU cycles. In contrast, the low current consuming sequence consists of a single DIV instruction that takes 4 CPU cycles to execute. The difference in power consumption can be attributed to the fact that the core sustains an issue rate of two instructions per CPU cycle for the

single-cycle integer instructions whereas the multi-cycle DIV instruction achieves 0.25 instructions executed per CPU cycle.

The period of execution of the overall loop (with both the high-current and the low-current consuming portions) is 8ns at 1.2GHz CPU frequency. This corresponds to a loop frequency of 150MHz. We sweep CPU frequency from 1.2GHz down to 120MHz in steps of 20MHz (the frequency step is limited by the multiplier which defaults to 20). Thus, a CPU frequency of 1.2GHz corresponds to 150MHz loop frequency, 1.1GHz CPU frequency corresponds to 137.5MHz loop frequency, and so on. We record the EM signal amplitude at each frequency point. Figure 23 shows the results of the frequency sweep. The amplitude is maximized at 70-72MHz when both cores are powered up (labelled by "C0C1") and at 85MHz when just one core is powered up (labelled by "C0"). These results are within the 1st order resonant frequency ranges for C0C1 and C0 scenarios determined in Section 4.3.2.

This methodology determines the first-order resonance significantly faster than the GA-framework. In our setup, this requires tens of minutes to complete, as compared to about a whole day for the GA. Furthermore, the fast method helps mitigate against the risk of the GA converging to a local optimum. In such an event, it is possible to perturb the GA execution by increasing one of its key parameters, such as the mutation rate. In all our measurements, we achieved excellent agreement between the GA framework and the fast-direct frequency-sweep method without any additional tuning of GA.

These findings provide strong support for the claims in Section 4.1. about the relationship between CPU EM emanations and PDN voltage noise. We proceed next to establish the generality of the proposed methodology by applying to different CPU cores and different platforms. In the next sections, we apply the EM methodology on the Cortex-A53 cluster on the Juno platform as well as an AMD x86 CPU and X-Gene2.

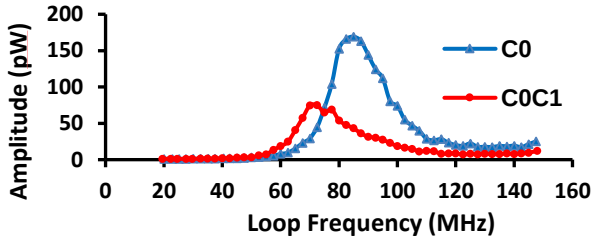

Figure 23. Resonant frequency exploration for Cortex-A72 with workload loop frequency modulated by CPU frequency.

4.4. EM Methodology Evaluation on Cortex-A53

We determine the first-order resonance frequency of the Cortex-A53 cluster using the fast methodology described in Section 4.3.5. We use a loop of 7 cycles that consists of 1 MUL instruction and 3 pairs of ADDs. A pair of ADDs is executed in parallel in 1 cycle whilst the MUL takes 4 cycles to execute. This translates to a loop frequency equal to 142MHz at a 1GHz CPU frequency. We sweep the CPU frequency from 1GHz down to 152MHz in steps of 16MHz (again the frequency step is limited by the frequency multiplier which defaults to 16 for the Cortex-A53).

The results of the sweep are shown in Figure 24. The quad-core cluster presents the highest die capacitance when all four cores are powered up. The first-order resonance frequency is inversely proportional to the square-root of the die capacitance [43]. The resonance frequency increases from 76.5MHz for all cores powered up (labelled as "C0C1C2C3") to 97MHz when just one core is powered up (labelled as "C0"). From the power-delivery perspective, power-gating individual cores in a cluster not only reduces the available useful capacitance that can mitigate high-magnitude voltage-droops, but also makes the frequency of voltage-noise oscillations higher. This has detrimental implications on voltage-noise mitigation mechanisms such as adaptive-clocking [21][29], that are extremely sensitive to response-latency. Thus, power-gating, whilst being beneficial from a leakage perspective, can affect power-delivery adversely.

Both the Cortex-A53 and Cortex-A72 clusters implement the same version of the ISA. Hence, we conduct a GA optimization run, with the same optimization parameters as in Section 4.3.3, but with the objective of obtaining a voltage-noise virus for the Cortex-A53 cluster. Figure 25 shows the inter-generational progression of the GA (left-axis showing received EM-power and the right-axis showing the dominant frequency of the strongest individual per generation). At the 107th generation the dominant frequency is 75MHz. This matches closely with what is observed in Figure 24 that shows the resonance frequency at 76MHz for four active cores. The agreement of the two independent approaches gives confidence that the resonant frequency is correctly identified by the GA.

Figure 26 shows the V_{MIN} of the EM virus (rightmost) compared to SPEC2006 benchmarks and idle (leftmost). The V_{MIN} is obtained with four active cores at a 950MHz CPU frequency using the V_{MIN} test methodology described in Section 4.3.4 but applied to the Cortex-A53. The V_{MIN} of the generated EM virus stands out (50mV higher) compared to the rest of the benchmarks.
It is important to note that the Cortex-A53 cluster does not have any support for voltage-noise measurement. Our methodology circumvents this shortcoming to obtain a) the first-order resonance frequency, and b) a virus that stresses voltage guardbands. This underlines the effectiveness and the generality of the proposed methodology.

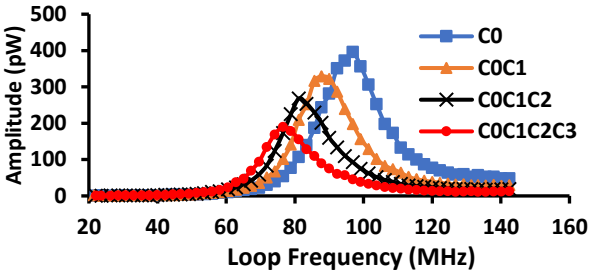

Figure 24. Resonant frequency exploration on Cortex-A53. For four powered cores (C0C1C2C3) the resonant frequency is 76.5MHz.

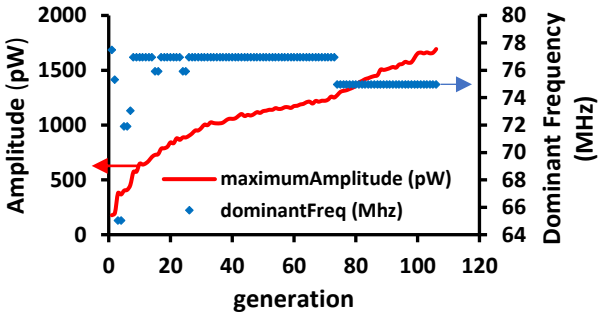

Figure 25. GA EM amplitude driven optimization for Cortex-A53.

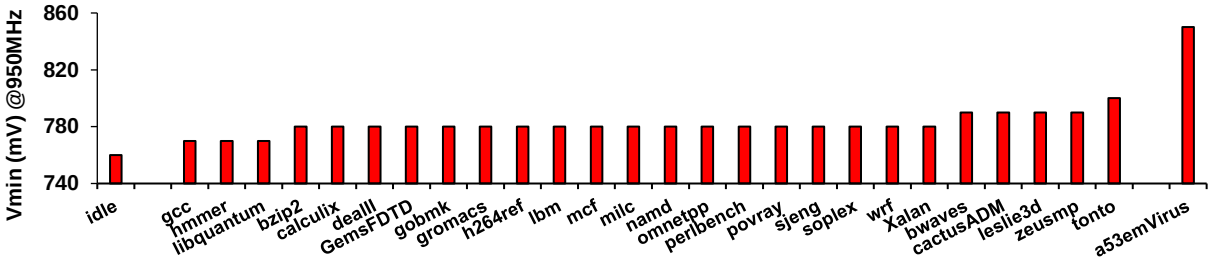

Figure 26. V_{MIN} measurements on Cortex-A53.

4.5. Simultaneous Voltage Noise Monitoring of Multiple Voltage Domains

We next illustrate the capability of the EM-based methodology to monitor multiple voltage domains simultaneously. This is impossible with an on-chip or off-chip oscilloscope that has a direct physical probing on a single voltage domain. Whereas an antenna can detect voltage emergencies happening at the same time on both the Cortex-A72 and Cortex-A53. To demonstrate this capability, we run the Cortex-A72 and Cortex-A53 dI/dt viruses at the same time and capture the spectrum analyser readings as shown in Figure 27. The frequency-domain signatures of both viruses are clearly visible. This shows that the EM methodology offers an effective detection mechanism for voltage-noise oscillations occurring across multiple voltage domains, thereby underlining its applicability to heterogeneous System-on-Chips (SoCs).

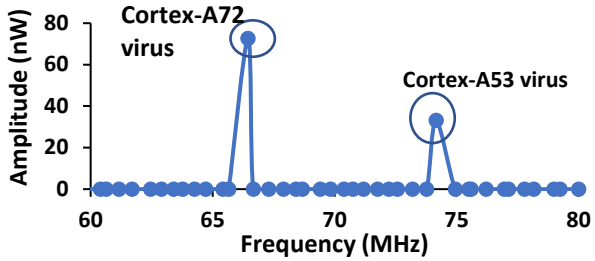

Figure 27. Simultaneous monitoring of voltage emergencies across multiple voltage domains through EM emanations.

4.6. EM Methodology Evaluation on AMD Athlon II X4 645

This section extends the evaluation from low-power mobile CPUs and the ARM ISA to high power x86-64 desktops (AMD Athlon II X4 645). The fast EM frequency sweep methodology for finding the resonant frequency (Section 3.3.5) is performed on the AMD CPU and the results are shown in Figure 28. The sweep reveals the first-order resonance frequency to be at 78MHz. An EM amplitude driven GA run shows excellent agreement converging to nearly the same resonant frequency (77MHz) as shown in Figure 29. The EM amplitude during the GA search follows the same trends as in the Juno board CPUs, Figures 8 and 13, it increases with each generation until it eventually converges.

For V_{MIN} comparison, the GA auto-generated EM virus is compared against common Windows (and Desktop CPU) workloads. The benchmark suite includes CPU intensive video rendering workloads such as Blender [33], Cinebench [34], scientific workloads such as Euler 3D [35] and all-around benchmark suites such as WEBXPRT [36] (mimics browser workloads) and GeekBench [37] (set of common workloads e.g. encryption, database queries etc.). Moreover, the EM virus is compared against the well-known Prime95 [38] stability test, AMD's own stability test application [25], and a GA virus generated through the voltage feedback from on-package Kelvin measurement pads (denoted as OscVirus). We monitor on-die voltage noise using a differential probe connected to an oscilloscope. The V_{MIN} and voltage noise results are shown in Figure 30. Unless noted otherwise, all measurements are with all four cores active.

The GA viruses (EMvirus, OscVirus) cause much higher voltage noise and have higher V_{MIN} compared to the rest of the workloads. The EM driven GA approach again is effective in generating voltage-noise viruses. The EM virus has a V_{MIN} of 1.3625V, 37.5mV below the nominal voltage at 3.1GHz. It is interesting to point that the EM based virus running on only two active cores is more severe than the AMD stability test and Prime95 on four active cores. To gain confidence in the V_{MIN} results we have run the AMD stability test and Prime95 for 24 hours at 1.287V and 1.28V respectively. They both pass the test whereas the EM virus causes immediate system-crash at 1.3V or even higher voltages.

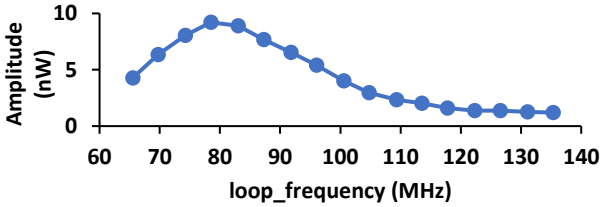

Figure 28. Loop frequency sweep on Athlon II X4 645 reveals a resonant frequency at 78.5MHz.

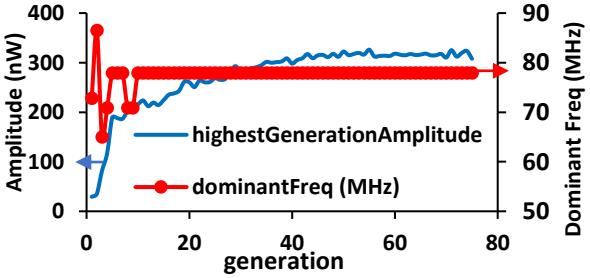

Figure 29. GA EM amplitude driven run on AMD CPU.

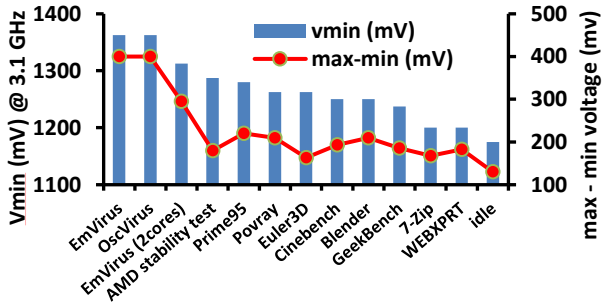

Figure 30. V_{MIN} and voltage noise measurements on the AMD CPU.

4.7. EM Methodology Evaluation on X-Gene2

This section presents the evaluation of the EM approach on X-Gene2. Figure 31 reveals the resonant frequency of X-Gene2 PDN. The results are obtained using the EM methodology for finding resonant frequency. The sweep reveals a resonant frequency at 150Mhz. We observe gaps between the data points because X-Gene2 supports only discrete CPU frequency values in steps of 300MHz. A GA optimization is able to maximize the EM amplitude at the 150MHz. The GA virus has comparable Vmin with SPEC workloads. As future work, we will continue the optimization of the GA stress tests for X-Gene2.

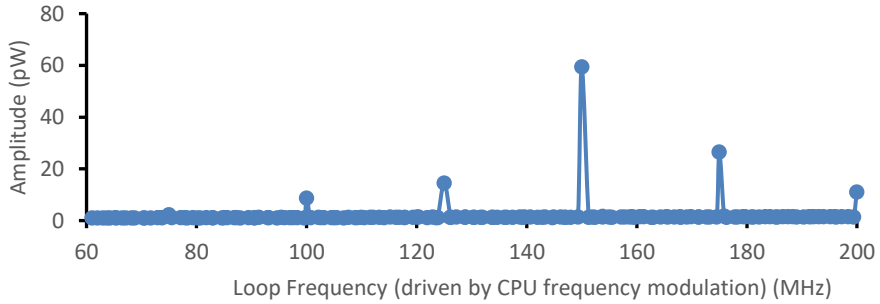

Figure 31. X-Gene2 frequency sweep reveals a PDN resonant frequency at 150MHz.

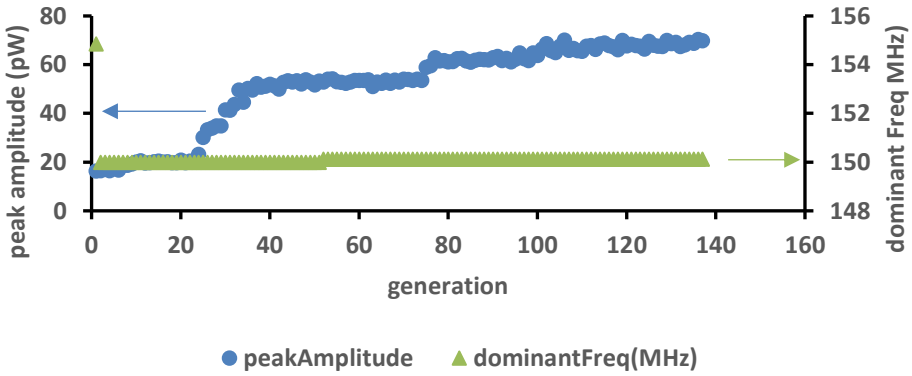

Figure 32. GA EM optimization on X-Gene2.

5. Conclusions and Future Work

This deliverable presents a system level voltage noise characterization framework. The analysis revealed that specific system calls cause high voltage noise. This is a new insight not shown in prior work.

Moreover, this work proposes a novel methodology for PDN characterization based on sensing modulations in CPU EM emanations. The proposed approach has the advantage of being non-intrusive to system-software and does not incur design-time complexities. The basic premise for this methodology is the presence of a correlation between the radiated EM power and on-chip voltage noise. The experimental

analysis clearly establishes this correlation. Additionally, we demonstrate the generality of the proposed approach by successfully applying it to different CPUs to obtain their PDN resonance frequency and generate voltage-noise viruses for them. Our experimental results show that PDN resonance frequency is typically located in the range between 50-150MHz and high-end CPUs have considerable voltage margins that can be shaved.

For future work, we plan to continue characterizing voltage noise and margins on X-Gene2/3 and other modern CPUs and GPUs.

6. References

[1] Reddi, Vijay Janapa, Svilen Kanev, Wonyoung Kim, Simone Campanoni, Michael D. Smith, Gu-Yeon Wei, and David Brooks. "Voltage noise in production processors." IEEE micro 31, no. 1 (2011): 20-28.

[2] Kim, Youngtaek, Lizy Kurian John, Sanjay Pant, Srilatha Manne, Michael Schulte, William Lloyd Bircher, and Madhu Saravana Sibi Govindan. "AUDIT: Stress testing the automatic way." In Microarchitecture (MICRO), 2012 45th Annual IEEE/ACM International Symposium on, pp. 212-223. IEEE, 2012.

[3] Polfliet, Stijn, Frederick Ryckbosch, and Lieven Eeckhout. "Automated full-system power characterization." IEEE Micro 31.3 (2011): 46-59.

[4] Ganesan, Karthik, and Lizy K. John. "MAximum Multicore POwer (MAMPO): an automatic multithreaded synthetic power virus generation framework for multicore systems." Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis. ACM, 2011.

[5] Whatmough, Paul N., Shidhartha Das, Zacharias Hadjilambrou, and David M. Bull. "14.6 An all-digital power-delivery monitor for analysis of a 28nm dual-core ARM Cortex-A57 cluster." In Solid-State Circuits Conference-(ISSCC), 2015 IEEE International, pp. 1-3. IEEE, 2015.

[6] Das, Shidhartha, Paul Whatmough, and David Bull. "Modeling and characterization of the system-level Power Delivery Network for a dual-core ARM Cortex-A57 cluster in 28nm CMOS." Low Power Electronics and Design (ISLPED), 2015 IEEE/ACM International Symposium on. IEEE, 2015.

[7] Reddi, Vijay Janapa, Meeta S. Gupta, Krishna K. Rangan, Simone Campanoni, Glenn Holloway, Michael D. Smith, Gu-Yeon Wei, and David Brooks. "Voltage noise: Why it's bad, and what to do about it." In 5th IEEE Workshop on Silicon Errors in Logic-System Effects (SELSE), Palo Alto, CA. 2009.

[8] Alam, M., B. Weir, and A. Silverman. "A future of function or failure?[CMOS gate oxide scaling]." IEEE circuits and devices magazine 18, no. 2 (2002): 42-48.

[9] Callan, Robert, Nina Popovic, Alenka Zajić, and Milos Prvulovic. "A new approach for measuring electromagnetic side-channel energy available to the attacker in modern processor-memory systems." In Antennas and Propagation (EuCAP), 2015 9th European Conference on, pp. 1-5. IEEE, 2015.

[10] Sehatbakhsh, Nader, Alireza Nazari, Alenka Zajic, and Milos Prvulovic. "Spectral profiling: Observer-effect-free profiling by monitoring EM emanations." In Microarchitecture (MICRO), 2016 49th Annual IEEE/ACM International Symposium on, pp. 1-11. IEEE, 2016.

[11] Genkin, Daniel, Lev Pachmanov, Itamar Pipman, and Eran Tromer. "Stealing keys from PCs using a radio: Cheap electromagnetic attacks on windowed exponentiation." In International Workshop on Cryptographic Hardware and Embedded Systems, pp. 207-228. Springer Berlin Heidelberg, 2015.

[12] Mitchell, Melanie. An introduction to genetic algorithms. MIT press, 1998.

[13] ARM, http://infocenter.arm.com/help/topic/com.arm.doc.100114_0200_03_en/arm_versatile_express_juno_r2_development_platform_(v2m_juno_r2)_technical_reference_manual_100114_0200_03_en.pdf

[14] Nazari, Alireza, Nader Sehatbakhsh, Monjur Alam, Alenka Zajic, and Milos Prvulovic. "EDDIE: EM-Based Detection of Deviations in Program Execution." In Proceedings of the 44th Annual International Symposium on Computer Architecture, pp. 333-346. ACM, 2017..

[15] Callan, Robert, Alenka Zajic, and Milos Prvulovic. "A practical methodology for measuring the side-channel signal available to the attacker for instruction-level events." Microarchitecture (MICRO), 2014 47th Annual IEEE/ACM International Symposium on. IEEE, 2014.

[16] Whatmough, Paul N., Shidhartha Das, Zacharias Hadjilambrou, and David M. Bull. "Power Integrity Analysis of a 28 nm Dual-Core ARM Cortex-A57 Cluster Using an All-Digital Power Delivery Monitor." IEEE Journal of Solid-State Circuits 52, no. 6 (2017): 1643-1654.

[17] Stutzman, Warren L., and Gary A. Thiele. Antenna theory and design. John Wiley & Sons, 2012.

[18] Thomas, Renji, Naser Sedaghati, and Radu Teodorescu. "EmerGPU: Understanding and mitigating resonance-induced voltage noise in GPU architectures." Performance Analysis of Systems and Software (ISPASS), 2016 IEEE International Symposium on. IEEE, 2016.

[19] Leng, Jingwen, Yazhou Zu, and Vijay Janapa Reddi. "GPU voltage noise: Characterization and hierarchical smoothing of spatial and temporal voltage noise interference in GPU architectures." High Performance Computer Architecture (HPCA), 2015 IEEE 21st International Symposium on. IEEE, 2015.

[20] Jordan, Edward C., and K. G. Balmain. "EM Waves & Radiating Systems." (2006).

[21] Grenat, Aaron, Sanjay Pant, Ravinder Rachala, and Samuel Naffziger. "5.6 adaptive clocking system for improved power efficiency in a 28nm x86-64 microprocessor." In Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International, pp. 106-107. IEEE, 2014.

[22] Ravezzi, Luca, and Hamid Partovi. "Clock and synchronization networks for a 3 GHz 64 Bit ARMv8 8-core SoC." IEEE Journal of Solid-State Circuits 50.7 (2015): 1702-1710.

[23] Joshi, Ajay M., Lieven Eeckhout, Lizy K. John, and Ciji Isen. "Automated microprocessor stressmark generation." In High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium on, pp. 229-239. IEEE, 2008.

[24] National Instruments drivers, http://www.ni.com/downloads/drivers/

[25] AMD overdrive, https://www.amd.com/en/technologies/amd-overdrive

[26] DS-5 debugger, https://developer.arm.com/products/software-development-tools/ds-5-development-studio/ds-5-debugger/overview

[27] S. Das, “Razor: A Variation-Tolerant Design Methodology for Low-Power and Robust Computing”, Doctoral Dissertation, University of Michigan, 2009.

[28] Ernst, Dan, Shidhartha Das, Seokwoo Lee, David Blaauw, Todd Austin, Trevor Mudge, Nam Sung Kim, and Krisztián Flautner. "Razor: circuit-level correction of timing errors for low-power operation." IEEE Micro 24, no. 6 (2004): 10-20.

[29] Lefurgy, Charles R., Alan J. Drake, Michael S. Floyd, Malcolm S. Allen-Ware, Bishop Brock, Jose A. Tierno, and John B. Carter. "Active management of timing guardband to save energy in POWER7." In Microarchitecture (MICRO), 2011 44th Annual IEEE/ACM International Symposium on, pp. 1-11. IEEE, 2011.

[30] Reddi, Vijay Janapa, Meeta S. Gupta, Glenn Holloway, Gu-Yeon Wei, Michael D. Smith, and David Brooks. "Voltage emergency prediction: Using signatures to reduce operating margins." In High Performance Computer Architecture, 2009. HPCA 2009. IEEE 15th International Symposium on, pp. 18-29. IEEE, 2009.

[31] Bertran, Ramon, Alper Buyuktosunoglu, Pradip Bose, Timothy J. Slegel, Gerard Salem, Sean Carey, Richard F. Rizzolo, and Thomas Strach. "Voltage noise in multi-core processors: Empirical characterization and optimization opportunities." In Microarchitecture (MICRO), 2014 47th Annual IEEE/ACM International Symposium on, pp. 368-380. IEEE, 2014.

[32] E. Alon, V. Stojanovic, and M. A. Horowitz, “Circuits and techniques for high-resolution measurement of on-chip power supply noise,” J. SolidState Circuits, vol. 40, no. 4, pp. 820–828, Apr. 2005.

[33] Blender, https://www.blender.org/

[34] Cinebench, https://www.maxon.net/en/products/cinebench/

[35] Euler3d benchmark.www.caselab.okstate.edu/research/ euler3dbenchmark.html

[36] http://www.principledtechnologies.com/ benchmarkxprt/webxpr

[37] GeekBench, https://www.geekbench.com/

[38] Prime 95, https://www.mersenne.org/download/

[39] Kim, Youngtaek, and Lizy Kurian John. "Automated di/dt stressmark generation for microprocessor power delivery networks." In Proceedings of the 17th IEEE/ACM international symposium on Low-power electronics and design, pp. 253-258. IEEE Press, 2011.

[40] ARM V8 ISA, http://infocenter.arm.com/help/topic/com.arm.doc.den0024a/DEN0024A_v8_architecture_PG.pdf

[41] Oracle x86 assembly language reference manual, https://docs.oracle.com/cd/E18752_01/html/817-5477/docinfo.html

[42] big.LITTLE Whitepaper ARM, https://www.arm.com/files/pdf/big_LITTLE_Technology_the_Futue_of_Mobile.pdfbig.LITTLE Whitepaper ARM, https://www.arm.com/files/pdf/big_LITTLE_Technology_the_Futue_of_Mobile.pdf

[43] S. Pant, “Design and Analysis of Power Distribution Networks in VLSI Circuits”, Doctoral Dissertation, University of Michigan, 2007.

[44] Fluhr, E.J., Friedrich, J., Dreps, D., Zyuban, V., Still, G., Gonzalez, C., Hall, A., Hogenmiller, D., Malgioglio, F., Nett, R. and Paredes, J., 2014, February. 5.1 POWER8 TM: A 12-core server-class processor in 22nm SOI with 7.6 Tb/s off-chip bandwidth. In Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International (pp. 96-97). IEEE.

[45] Kurd, Nasser A., Subramani Bhamidipati, Christopher Mozak, Jeffrey L. Miller, Timothy M. Wilson, Mahadev Nemani, and Muntaquim Chowdhury. "Westmere: A family of 32nm IA processors." In Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International, pp. 96-97. IEEE, 2010.

[46] Papadimitriou, George, Manolis Kaliorakis, Athanasios Chatzidimitriou, Dimitris Gizopoulos, Peter Lawthers, and Shidhartha Das. "Harnessing voltage margins for energy efficiency in multicore CPUs." In Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture, pp. 503-516. ACM, 2017.

[47] Fischer, Tim, Jayen Desai, Bruce Doyle, Samuel Naffziger, and Ben Patella. "A 90-nm variable frequency clock system for a power-managed itanium architecture processor." IEEE Journal of Solid-State Circuits 41, no. 1 (2006): 218-228.

[48] NVIDIA, Tegra. "K1: A new era in mobile computing." Nvidia, Corp., White Paper (2014).

[49] Gwennap, Linley. "Qualcomm Krait 400 hits 2.3 GHz." Microprocessor report 27, no. 1 (2013): 1-6.

[50] Pyo, J., Shin, Y., Lee, H.J., Bae, S.I., Kim, M.S., Kim, K., Shin, K., Kwon, Y., Oh, H., Lim, J. and Lee, D.W., 2015, February. 23.1 20nm high-K metal-gate heterogeneous 64b quad-core CPUs and hexa-core GPU for high-performance and energy-efficient mobile application processor. In Solid-State Circuits Conference-(ISSCC), 2015 IEEE International (pp. 1-3). IEEE.

[51] Mair, H.T., Gammie, G., Wang, A., Lagerquist, R., Chung, C.J., Gururajarao, S., Kao, P., Rajagopalan, A., Saha, A., Jain, A. and Wang, E., 2016, January. 4.3 A 20nm 2.5 GHz ultra-low-power tri-cluster CPU subsystem with adaptive power allocation for optimal mobile SoC performance. In Solid-State Circuits Conference (ISSCC), 2016 IEEE International (pp. 76-77). IEEE.

[52] Pant, Pankaj, Joshua Zelman, Glenn Colon-Bonet, Jennifer Flint, and Steve Yurash. "Lessons from at-speed scan deployment on an Intel® Itanium® microprocessor." In Test Conference (ITC), 2010 IEEE International, pp. 1-8. IEEE, 2010.

[53] http://download.intel.com/design/mobile/datashts/31407804.pdf

[54] http://support.amd.com/TechDocs/31412.pdf

[55] DeHaven, Keith, and Joel Dietz. "Controlled collapse chip connection (C4)-an enabling technology." In Electronic Components and Technology Conference, 1994. Proceedings., 44th, pp. 1-6. IEEE, 1994.

[56] Bockelman, David E., and William R. Eisenstadt. "Combined differential and common-mode scattering parameters: Theory and simulation." IEEE transactions on microwave theory and techniques 43, no. 7 (1995): 1530-1539.

[57] HSPICE circuit simulation tool, https://www.synopsys.com/verification/ams-verification/circuit-simulation/hspice.html

[58] Hadjilambrou, Z., Das, S., Antoniades, M., & Sazeides, Y. (2017). Sensing CPU voltage noise through Electromagnetic Emanations. IEEE Computer Architecture Letters.

[END OF DOCUMENT]

