uniserver

D3.2 1st Analysis of Processor Cores under Various Stress Conditions

Contract number	688540
Project website	http://www.uniserver2020.eu
Contractual deadline	Project Month 12 (M12): 31st January 2017
Actual Delivery Date	20/2/2017
Dissemination level	Confidential
Report Version	1.0
Main Authors	S. Das (ARM), Z. Hadjilambrou (UCY), Y. Sazeides (UCY)
Contributors	
Reviewers	Dimitris Gizopoulos (UoA), Georgios Karakonstantis (QUB), George Papadimitriou (UoA), Manolis Kaliorakis (UoA), Athanasios Chatzidimitriou (UoA)
Keywords	voltage noise, genetic algorithm, voltage margins

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.

© 2017. 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
0.1	First draft added layout, GA framework description, IPC, thermal and EM measurements
0.2	Added Juno voltage noise measurements
1.0	Fixed minor issues

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. The X-Gene3 platform is a 2.8GHz 32-core system that can benefit significantly from margin elimination techniques developed as a part of the UniServer Project. However, the X-Gene3 is due to be available in the later half of 2017. In the absence of X-Gene3, we have performed the voltage noise characterization using two alternative ARMv8 based platforms:

A) We have extended our analysis of the dual-core Cortex-A57 Platform (internally at ARM code-named as the Juno platform) and use the on-chip digital-storage oscilloscope (DSO) circuitry to develop an understanding and knowledge about the power-delivery network behaviour. The measurement and analysis work that has been ongoing since mid-2015 has been extended to analyse the behaviour of power-grid oscillations during scan-shifts. We have now applied this problem both to margin measurements during scan-shifts and to full-system simulation studies using industrial workloads. This work was submitted for publication in the IEEE Journal of Solid-State Circuits (JSSC) in November 2016 and has been accepted for publication (January 2017). We have also developed a comprehensive power-grid analysis platform that is currently being collated and prepared for a subsequent journal submission.

B) We continue to develop novel (but indirect) voltage noise measurement setup for current-generation X-Gene2 system that lacks capabilities for direct on-die voltage noise measurement. We have developed a frequency-domain measurement technique that analyzes Electromagnetic (EM) emanations from the chip to characterize the frequency of power-supply oscillations and their relative amplitudes. This setup is currently under development and we have promising preliminary results.

This deliverable presents the combined efforts for both approaches outlined above. We present the power-delivery simulation methodology developed on the Juno platform which elucidates our understanding of software interactions with the power-grid behavior. We present worst-case vector generation results using a genetic-algorithm (GA) framework. The GA framework successfully generates a stress tests that causes 2X more voltage noise than conventional applications. The stress test reveals that operational voltage can be reduced by at least 6% for dual core operation and by at least 10% for single core operation. We also present results from direct measurement of the power-grid during scan-shifts.

We developed a GA framework, similar to the one for the Juno board, and apply it to the Applied Micro X-Gene2 platform. We evaluate the GA effectiveness on the X-Gene2 chip running on a Tigershark validation board. The Tigershark board does not support fine-grain voltage measurements so generation of voltage noise stress test is not straightforward. Still, we show that the GA framework approach is applicable on this platform by using it to generate instructions per cycle (IPC) and thermal stress tests for the X-Gene2. These results reassure that the framework should be able to generate maximum voltage noise on the X-Gene3 , which will offer fine grain voltage visibility. Last, in the absence of fine grain voltage monitoring capabilities, we explore the potential of monitoring of CPU voltage noise indirectly through electromagnetic emanations with encouraging initial results.

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. We intend to target the Applied Micro X-Gene3 platform for these measurements. The X-Gene3 is a 32-core processor that provides means to directly measure supply voltage noise. However, the X-Gene3 board is due in the later half of 2017.

In the absence of the X-Gene3 board, the consortium has made progress towards characterizing voltage noise in two ways: a) We have extended our capabilities of measurement of on-chip voltage noise on the Juno platform [13] to develop a simulation and modelling methodology that helps our understanding of the voltage-noise problem and b) we have developed indirect voltage-noise measurement methods to help us estimate on-chip voltage noise using off-chip electromagnetic (EM) emanations measurements. We use the existing X-Gene2 platform to achieve this.

Central to this deliverable is the genetic algorithm (GA) framework for generation of voltage noise viruses. The framework successfully generates a voltage noise virus on Juno board [6]. Juno board offers a dual core Cortex-A57 CPU and fine grain on-chip voltage measurements [5]. The voltage noise virus exposed the pessimistic margins. Despite worst case voltage noise caused by the virus,the voltage can be reduced by 6% for dual core operation and by 10% for single core operation. When running conventional applications, voltage can be reduced even further as these application cause 2X less voltage noise compared to the virus.

This deliverable describes a simulation and modelling analysis that is informed by our capability to directly measure on-chip supply voltage noise. We use the scan-shifting scenario as a particular case of worst-case supply voltage noise and present results.

We further develop the genetic algorithm (GA) based automated code generation for worst-case supply voltage noise and apply it to the X-Gene2. On the X-Gene2 Tigershark board the GA framework is able to generate IPC (instructions per cycle) and thermal stress tests. Voltage noise virus generation is not straight forward since the Tigershark board does not support fine grain voltage measuremetns. Still though, we show that the GA framework approach is applicable on this platform and this is reassuring that the framework will be able to generate voltage noise virus on the X-Gene3 that will offer fine grain voltage visibility.

Despite the lack of fine grain voltage measurements on Tigershark board we are in the process of evaluating CPU Electromagnetic Emanation (EM) as a proxy for voltage noise. This deliverable describes our experimental setup and shows why EM approach seems promising for characterizing voltage noise.

The rest of the deliverable is organized as follows: Section 2 present the GA framework for automatic stress test generation. Sections 3 and 4 present voltage noise analysis on Juno board. Section 5 presents EM measurements on X-Gene2. Section 6 extends and applies the GA framework on the X-Gene2 to generate IPC and temperature viruses. Section 7 concludes.

2. Genetic Algorithm Framework for Stress Test Generation

This section presents the GA framework for automatic stress test generation.

2.1 Framework Description

A framework for stress test generation based on genetic algorithms has been developed. Stress tests are micro benchmarks that can be used to characterize the system margins. For instance, a voltage noise stress test generates maximum voltage noise and can be used for characterizing voltage margins. Some other examples of stress tests are power and IPC "viruses". A power virus can be used for characterizing a system's TDP (Thermal Design Power) and an IPC virus characterizes the maximum instruction throughput.

Genetic algorithms (GA) are suitable for stress test generation. Previous work used genetic algorithms to generate power and di/dt (voltage noise) viruses [2,3,4]. Fig. 1 shows the overview of the genetic algorithm framework. In general, genetic algorithms optimize toward the solution by applying bio inspired operators like crossover and mutation. Our Genetic framework's goal is to generate an ARM assembly instruction sequence that maximizes the metric of interest. The framework takes as inputs ARM instructions that will be used in the optimization procedure (other ISA like x86 or MIPS can be used as well). Some examples of metrics of interest that the framework can optimize for are the following: maximum temperature, power, IPC and voltage noise.

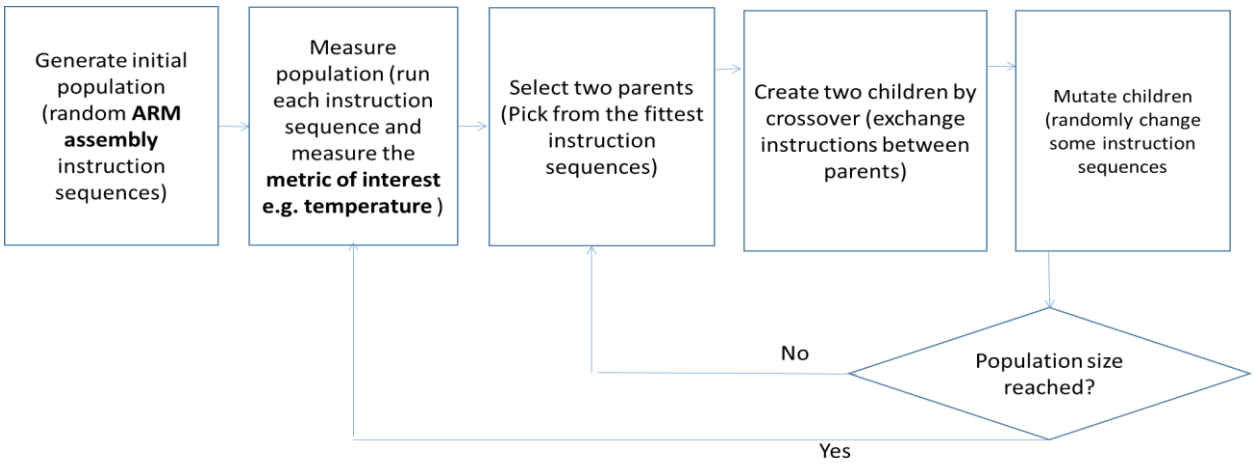


Figure 1. Genetic algorithm overview

Description of how the genetic algorithm works follows. Initially the algorithm generates a random population of instruction sequences. Each instruction sequence represents an individual of the population. Each individual is compiled to binary. Then the binary is executed on the machine and its fitness is recorded. For instance, if the metric of interest is maximum temperature, then, the temperature during the execution is measured and recorded. The individual's that causes the highest temperature is consider the fittest. After all individuals of a population are measured, the algorithm proceeds on generating the next population. Until the population size is reached the algorithm performs the follow steps: 1) Select two parents (the fitter an individual is the higher chance to be selected as parents), 2) create two children by exchanging instruction among two parents (crossover), 3) Mutate the children by randomly changing some children's instructions. The whole process of measuring and generating new populations continues until we are happy with the results (e.g. produced an individual that reaches the thermal limit), or the algorithm has converged (meaning it has produced a population of identical individuals), or for many generations the metric of interest does not improve.

Implementations of genetic algorithms can vary a lot as different implementations of parent selection, crossover and mutation operators can be used. Also, different values for parameters as population size, and mutation rate can be used. Table I shows the specific of our implementations. We find that this implementation specifics parameters work well for our purpose but still is an interesting future work to find whether other implementation can improve furthermore our results.

We will discuss some parameters. First, each individual is described by a loop body of 50 ARM instructions. We find that 50 instructions are enough to produce temperature, IPC, voltage noise and power

viruses. These instructions include arithmetic short and long latency, floating point, memory and branch operations. Regarding genetic operators, Figure 2 shows how we implemented the genetic algorithm operators. For parent selection, we used tournament selection. We implemented tournament selection by selecting randomly 5 individuals and promoting the fittest individual for crossover. For crossover, the one point crossover technique is used. An example of one point crossover is shown in Figure 2. In this example the individual's size is 4 instructions. A crossover point is randomly selected. In the example the crossover point is the second instruction. Instructions after crossover point are inherited from parent1 to child2 and from parent2 to child1. Whereas, instructions before crossover points are inherited from parent1 to child1 and from parent2 to child 2. Last, for mutation rate we chose a low value particularly 2%. Mutation rate dictates that each instruction of a child has a 2% chance to convert to a different instruction. For our purposes, we find that low mutation rates work the best because higher mutation rate can ruin the effectiveness of fit instruction sequences.

Table i. GA parameters

Population Size	50
Individual Size (in ARM instructions)	50
Parent Selection Operator	Tournament Selection
Tournament Size	5
Crossover operator	One point crossover
Mutation rate	2%

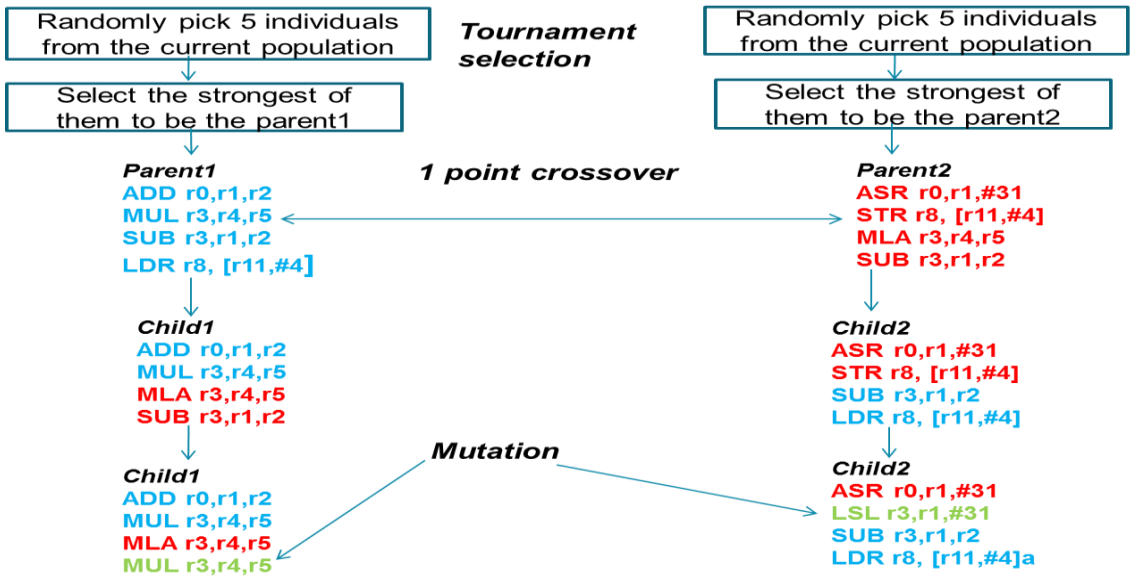


Figure 2. Genetic operators

3. Juno Voltage Noise Simulation Analysis

The X-Gene2 platform does not provide for direct measurement of on-chip supply voltage noise. There are a pair of “voltage-sense” pins that are available on package which are routed directly to the power management IC (PMIC). Our simulation analysis (shown in Section 3.1) shows that the signals at the PMIC are essentially low-bandwidth signals (in the KHz) range. Therefore, high-frequency on-chip voltage noise is heavily filtered and cannot be observed at these observation points.

In contrast, the X-Gene3 board addresses these complications by providing for voltage sense pins directly on the package. Instead of routing them to the PMIC, direct measurement of these voltage sense pins will enable us to read the on-chip noise directly on an external bench-top oscilloscope. In addition, the X-Gene3 chip also integrates on-chip voltage threshold-crossing detectors (for both voltage overshoots and undershoots) that can be queried for direct reading of on-chip voltage noise. The X-Gene3 board will be available in the later half of 2017 and is not available to the consortium currently.

We had to develop alternative mechanisms for on-chip voltage noise measurement and analysis in light of the existing limitations for voltage-sensing in the X-Gene2. We address this by a) developing and evaluating our methods on an alternative ARM-based platform (the Juno platform) and b) by trying out EM measurements as a different approach for voltage noise characterization.

We selectected the ARM Juno [13] platform for several compelling reasons. Firstly, the Juno platform integrates a high-performance (excess of 1.2GHz) dual-core ARM Cortex-A57 cluster. Micro-architecturally, the ARM Cortex-A57 core implements the same instruction set architecture (ISA) as the X-Gene2 and the X-Gene3. Hence, the application software written for the Cortex-A57 can be directly executed (binary compatible) with the X-Gene2 and the X-Gene3 platform.

Secondly, the Juno platform is a dual-core Cortex-A57 cluster. System-architecturally, this is very similar to a PMD (Processor Module) on the X-Gene processor series. Thus, the software interactions that we can reveal in our Juno analysis should also hold true when we are ready to directly measure the X-Gene3 platform. We particularly highlight the impact of multi-core execution and power-gating on supply voltage noise.

The third compelling reason for the choice of the Juno platform is the availability of an on-chip Digital Storage Oscilloscope (OC-DSO) [5,11] that was designed and integrated by UniServer partners (ARM) for direct snooping of on-chip voltage noise on the Cortex-A57 cluster. Our knowledge of this voltage monitoring peripheral enables us to develop a comprehensive analysis framework that we can translate to the X-Gene3 platform. The know-how gained from voltage noise analysis on Juno will be used on X-Gene3 chip that will support fine grain voltage measurements. Furthermore, we can directly use the OC-DSO to evaluate the effectiveness of our non-invasive noise measurement technique that we describe in section 4.

In this section, we describe our analysis setup for the Juno platform.

3.1 Juno PDN Simulation Framework

We designed and developed a simulation methodology to analyse the impact of supply noise. Our simulation
methodology enables us to execute industrial workloads which we then compare against direct measurement results to validate the model.

Fig. 3 illustrates the power delivery network model [6,12]. The hybrid model incorporates a combination of lumped elements and distributed network models to efficiently model system behavior across a wide frequency range. Accurate power delivery network (PDN) behavior at low-frequency requires a closed-loop, small-signal model of the VRM, with a voltage-feedback sense-point and an appropriate feedback-compensation model. Instead, we represent the VRM by an open-loop, small-signal lumped circuit model. Such an approach retains PDN accuracy at the mid- and high-frequency ranges (100kHz – 100MHz) without increasing overall simulation and modeling complexity.

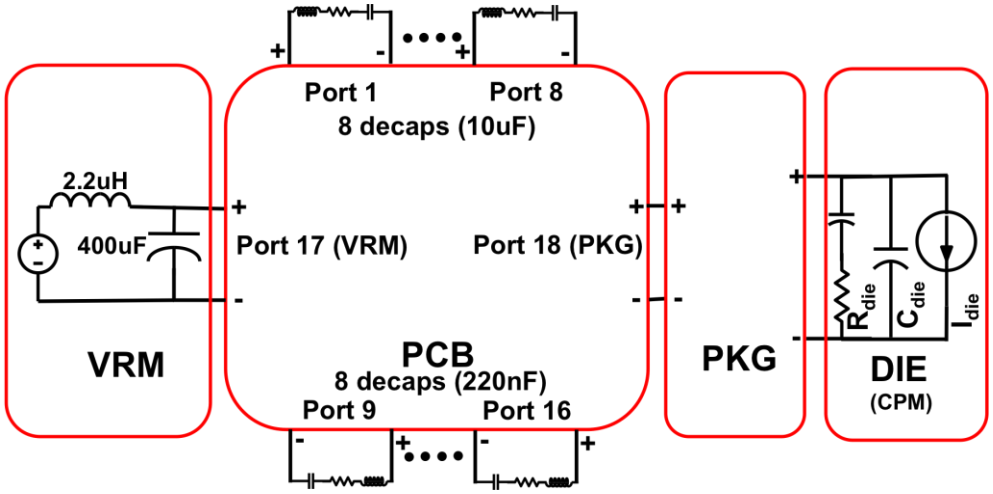


Figure 3. PDN simulation model consisting of lumped and distributed parameters

We extract a lumped chip power model (CPM) for the A57 compute cluster using Apache Redhawk [14]. The lumped model of the die consists of a current source that represents switching transistors. Non-switching transistors act as local decoupling capacitors that provide instantaneous current demands. The power grid resistance (Rdie) is also modeled in the power model for the die. Resonance interactions between the die and the rest of the PDN network occur at frequencies in the range around 100MHz, where a lumped circuit model is sufficient for accurate modeling of the die.

3.2 PDN simulation results

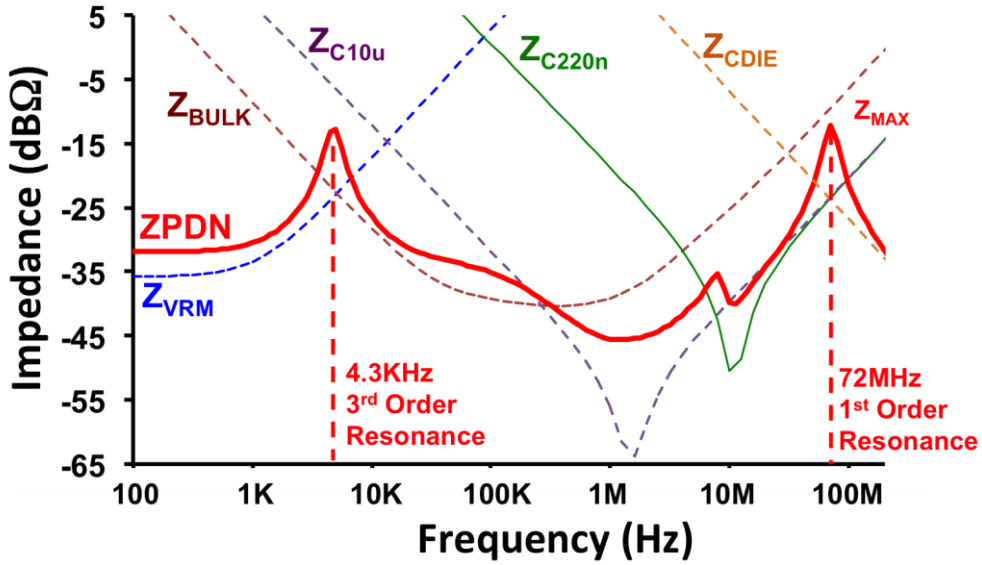


Figure 4. PDN Frequency Domain Simulation Results

Fig. 4 shows the input impedance of the PDN as a function of frequency. The impedance is represented in the db-Ohm scale (1ohm = 0dB) in order to highlight key attributes of the PDN that may not be seen on the linear scale. The dB scale helps highlight the *relative* difference in magnitude i.e. every 6dB reduction is equivalent to a magnitude difference of 50%.

The VRM (voltage regulator module), PCB (printed circuit board) decoupling capacitors and the die-capacitance form three parallel impedance branches of the overall PDN. At low-frequencies (<1KHz), the path of least impedance is through the VRM. The series inductor (2.2uH) in the switching regulator circuit dominates overall VRM impedance (Z_{VRM}). The inductor and the bulk-capacitor (400uF) at the VRM output form a LC-tank circuit that resonates at 4.5KHz. This represents the *third-order* resonance frequency of the system PDN.

The impedance of the VRM bulk-capacitor (Z_{BULK}) is ultimately limited by its ESL, which causes inductive behavior beyond the self-resonance frequency of 750KHz. The PCB decap network consists of 8 capacitors

of value 10uF each and an additional 8 capacitors of value 220nF. The set of 10uF capacitors have a self-resonance frequency of 1MHz. Beyond this, their frequency response is dominated by the parasitic inductance loop formed by the series connection of the decap ESL and the PCB trace inductance connecting the capacitor to the die bumps. Looking in from the decap pads as a single lumped port, this parasitic loop inductance was measured to be 143pH (in simulation) for the set of 8 10uF capacitors.

An interesting observation is that the system PDN impedance (Z_{PDN}) does not have a prominent 2nd-order resonance peak. This occurs due to the 10uF decaps that begin to interact capacitively before Z_{BULK} exhibits inductive behavior, thus neutralizing the 2nd-order resonance peak.

The key conclusions to draw from the above frequency-domain analysis are:

a) The first-order resonance is determined by the die-capacitance interacting with the parasitic inductance of the PCB decaps. The 16 PCB decaps act as a lumped circuit element that present an combined inductance of 72pH.

b) The PCB decaps act capacitively at frequency below 10MHz to reduce the PDN impedance. Beyond this frequency, decaps act effectively as inductors.

The parasitic inductance of the PCB decaps is dominated by their connection to the PCB. The ESL of individual capacitors is ~400pH, whereas the PCB connections from the decaps to the package add an effective inductance of 1.2nH. Thus, the ESL of each capacitor element affects the overall inductance weakly, which is dominated as such by PCB parasitics.

4. Measurement Results on the Juno PDN

Fig. 5 shows our on-chip measurement framework. A high-bandwidth on-chip digital storage oscilloscope (OC-DSO) snoops the supply rails of the Cortex-A57 cluster. The OC-DSO 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 adaptation of the bandwidth/sample rate to allow measurement of low frequency transients.

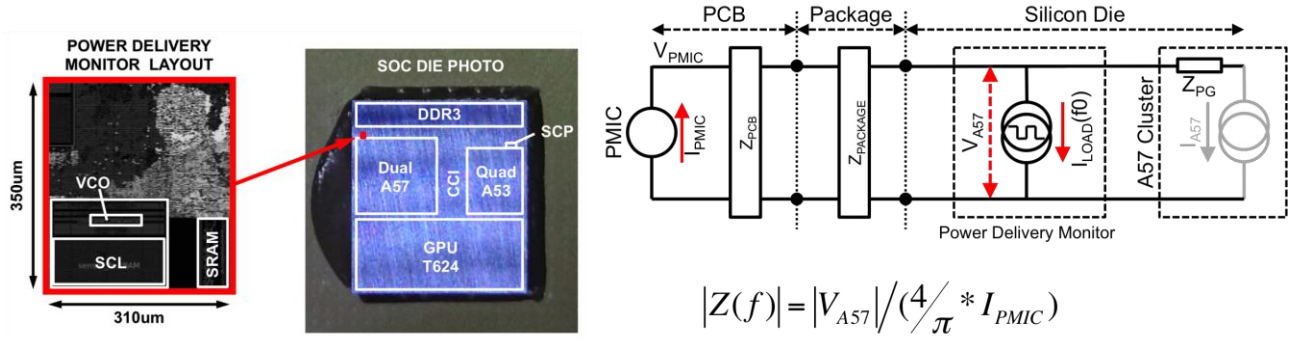


Figure 5. PDN Measurement Setup – The On-Chip Digital Storage Oscilloscope (OC-DSO) design and integration into Juno predates Project UniServer and has been provided here for completeness.

The OC-DSO is augmented with a stimulus-generation circuitry to enable on-chip measurement of the PDN input impedance. The stimulus consists of multiple chains of ring-oscillators (RO) that present a near-constant current load to the A57 PDN depending upon the number of ROs that are enabled. Using the Synthetic Current Load (SCL) block, we create a square-wave current excitation on the PDN by enabling and disabling the RO-load at a specific frequency. We measure the maximum voltage droop induced by this periodic excitation and obtain the input impedance spectrum by sweeping the square-wave frequency across a range.

The on-chip measurement setup enables high-bandwidth probing of the internal supply rails of the A57. Accuracy of the time-domain measurement is limited by the minimum resolution of the internal ADC and the internal noise generated by the clocked transistors. Therefore, the minimum voltage droop that can be measured using this technique is in the range between 5mV-10mV. The limitations on the maximum current draw of the SCL block and the accuracy of the internal ADC limit the minimum impedance magnitudes (~50mOhm) that can be measured using this technique.

4.1 Worst-Case Resonant Code Generation

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 that is agnostic to the processor micro-architecture.

We used the genetic algorithm setup outlined in Section 2 to develop resonant workloads for the Juno platform. An initial seed population of instructions is generated from the packaged vectors delivered with the A57 processor IP. The algorithm uses voltage-noise measurements, from the on-chip oscilloscope circuitry as the optimization objective function. Selection criteria, based on the droop magnitude, prune the workload population by selecting the ones causing the maximum voltage droops. These serve as “parents” which are then paired and mutated to create the next generation of workloads. The process is iterated until the increase in voltage droop across succeeding generations saturates. The fittest instruction in the final iteration is chosen as the worst-case workload (the “GA_Max_Droop” workload). This approach allows flexibility since different measurement variables (such as average current consumption) can also be chosen as the optimization objectives.

Genetic-algorithm based approaches often converge on local minima and may not generate the global minima. Therefore, we evaluate the output of the algorithm in terms of its efficacy in exciting the worst-case resonance. Fig. 6 shows the simulated current waveform (I_VDD) for two successive loops of the “GA_Max_Droop” workload. The current waveform appears discontinuous due to an ideal PDN being used,

with infinite current bandwidth. Each loop is 67ns in duration when the processor is operated at the clock frequency of 1.27GHz. This is close to the maximum frequency of operation for the compute cluster, as measured in silicon.

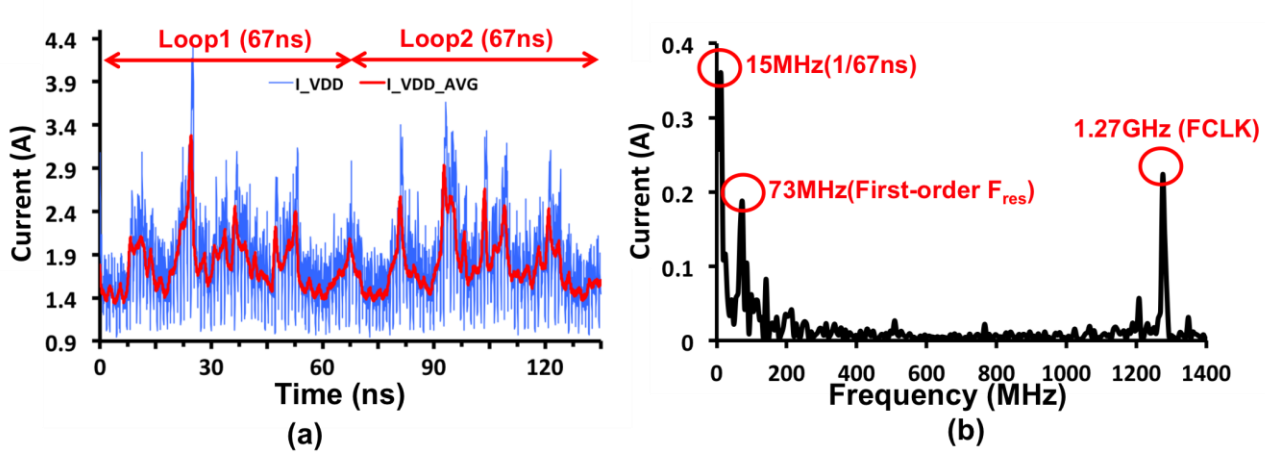


Figure 6. Simulated current waveform (I_VDD) for two successive loops of the "GA_Max_Droop" workload

Fig. 7 shows how the "gaDIDT" workload (the GA generated voltage noise virus) compares to other workloads in terms of voltage noise and Vmin (minimum operational voltage for a given frequency). Measurements from single and dual core runs are shown. On dual core runs both voltage noise and vmin are higher. The voltage noise virus generates 2X more voltage droop than conventional workloads. Also, the vmin measurements suggest that the operational voltage for 1.1GHz frequency can be reduced by at least 6% for dual core operation and by 10% for single core operation.

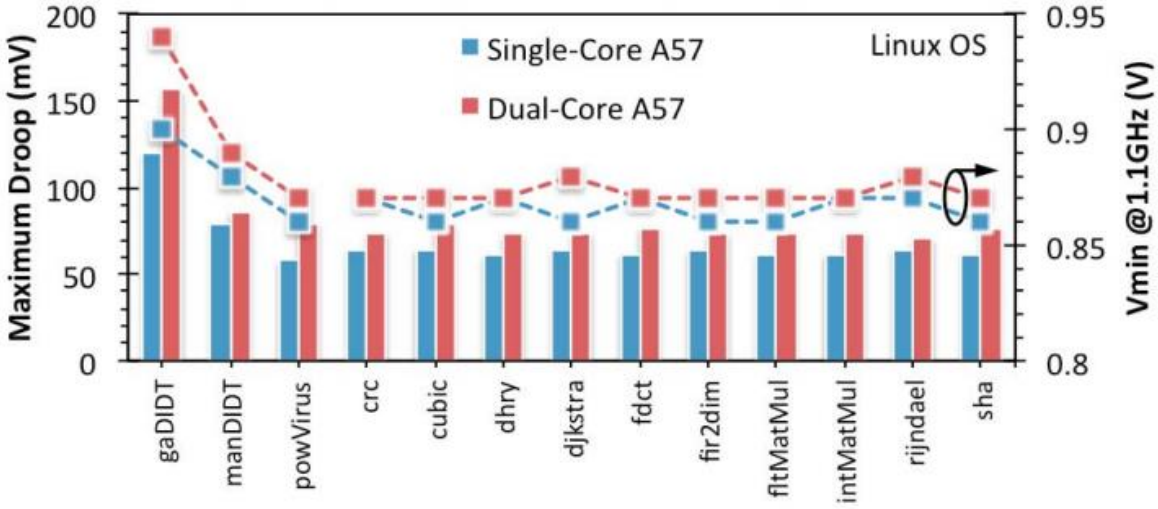


Figure 7. Vmin and maximum voltage noise for various workloads

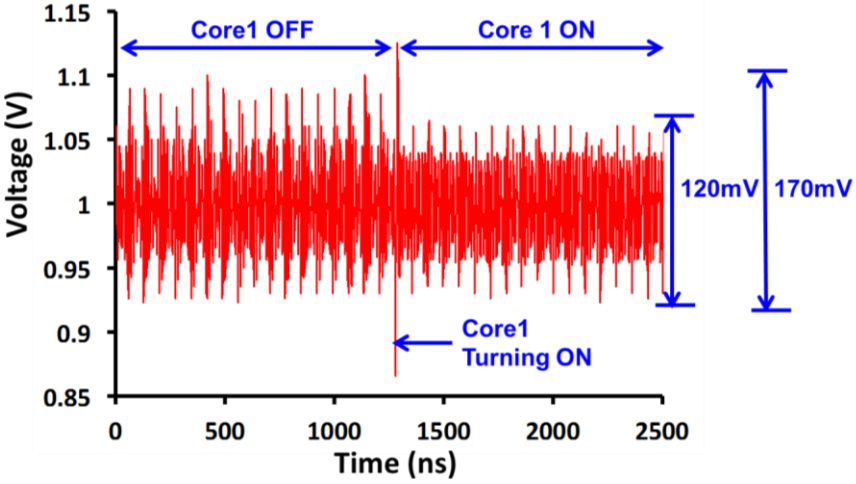


Figure 8. Dependence of system resonance frequency as a function of number of cores in operation. Core 0 is executing resonant code. Core 1 is power-gated initially and then turned on.

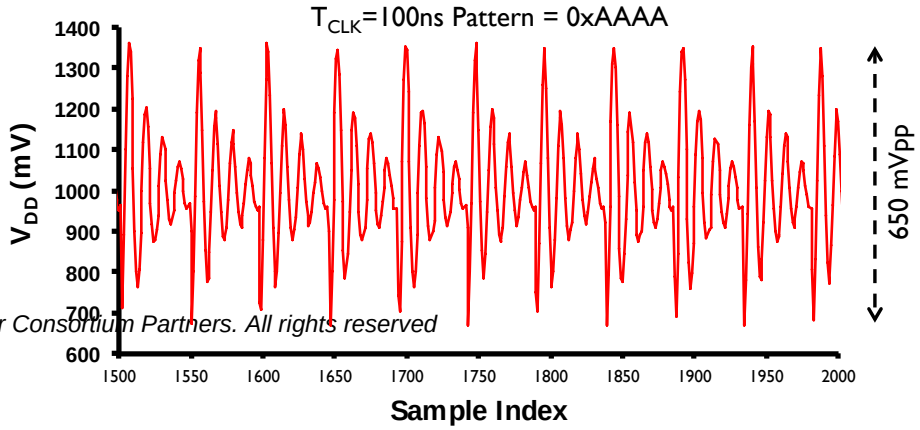
4.2 Multi-Core Interactions on System Resonant Frequencies

Fig. 8 illustrates the impact of power-gating on the system resonance frequency. In the dual-core configuration, one of the cores (Core 0) is executing resonant code. Initially, the counterpart core (Core 1) is kept powered off and is then turned on mid-execution. The system suffers a strong voltage undershoot due to current transients when Core 1 is suddenly triggered. The magnitude of the system oscillations subsequently achieve steady-state amplitudes that reflect the presence of additional capacitance in the power-delivery network. The magnitude of oscillations in both phases of oscillations is also labelled in the figure. The system suffers higher amplitude of oscillation due to reduced capacitance in the first phase of execution (when Core 1 is power-gated). The frequency of oscillation is also higher (83MHz) compared to the second-phase of oscillation where the system frequency is lower at 72MHz.

This phenomenon illustrates the difficulty in designing adaptive circuits that can respond in time to supply-voltage oscillations. The system suffers more current consumption when multiple cores are turned on. On the other hand, power-gating reduces total system capacitance thereby making the system susceptible to greater AC noise. This trade-off needs to be an important consideration when designing adaptive circuits to cope with voltage noise.

4.3 Voltage Noise During Scan-Shift

High toggle-rate in flip-flops during scan-shift increases the activity rate in combinational logic. This significantly increases processor I_{DD} compared to functional tests, leading to large V_{DD} swings, potentially causing test-pattern mismatches during vector replay on the Automatic Test Equipment (ATE). Low toggle-rate test-vectors reduce I_{DD} during scan-shift, at the expense of increased test-time.



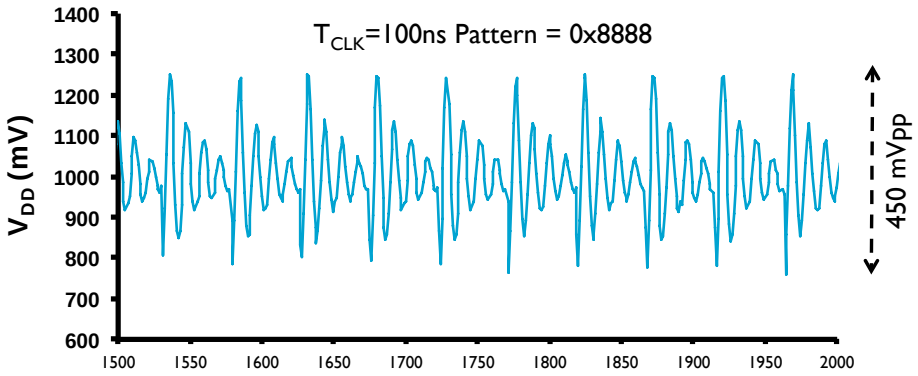


Figure 9. Noise Measurements during scan-shift

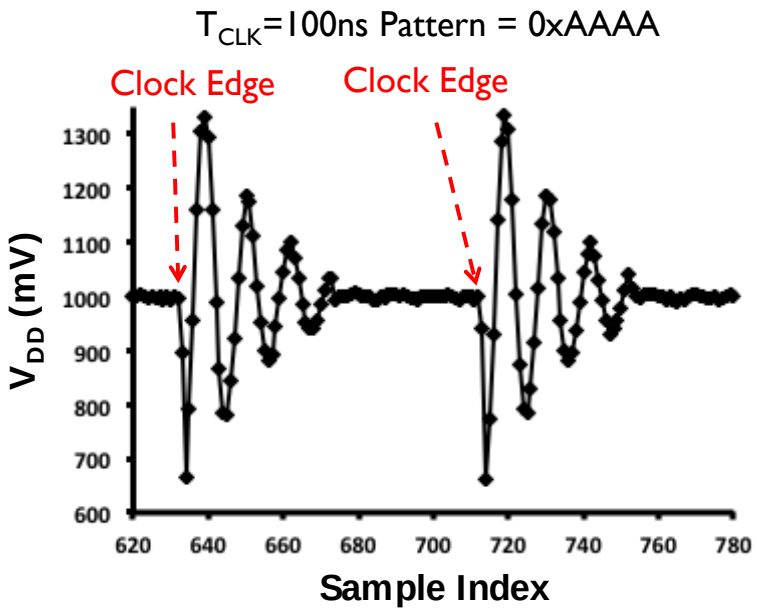
Fig. 9 shows V_{DD} oscillations, as measured using the PDM, during scan-shift. The shift pattern 0xAAAA results in an extreme power supply noise condition, since all flip-flops inside the scan-chain toggle in every cycle. We compare this against the power-supply noise observed using a relaxed pattern (0x8888) that suppresses half of the transitions during the shift operation. The increase in step-current magnitude at the rising edge of the scan-clock results in larger magnitude of V_{DD} undershoots and overshoots for the worst-case toggle pattern. In both cases, the magnitude of the peak-to-peak swing is significant (650 mVpp for the worst-case pattern and 513mVpp for the relaxed pattern) and is in excess of V_{DD} noise observed during functional test.

Fig. 10 shows the frequency-dependence of the scan-clock on the shift-induced power-supply oscillations. The rising edge of the clock represents the initiation of the current step, where all flip-flops toggle as a new bit is shifted in. The resultant switching activity in the combinational logic generates high peak currents.

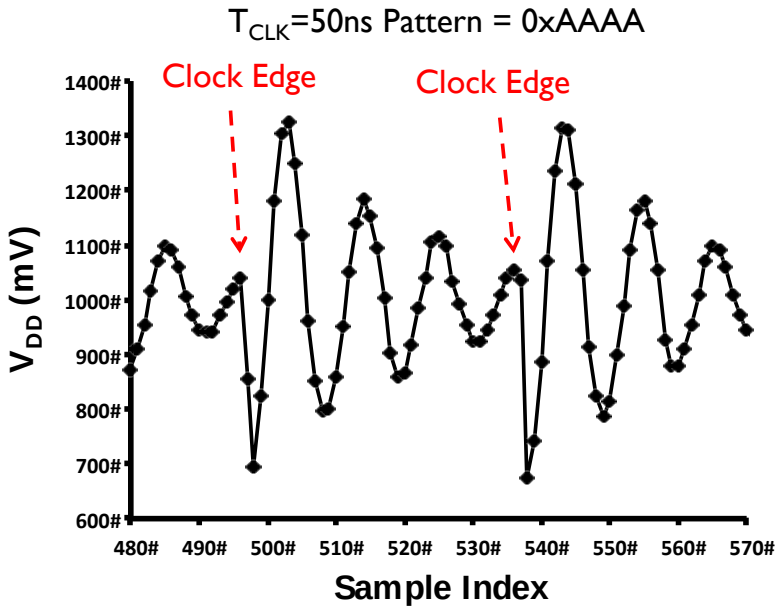
The scan-clock frequency in Fig. 10(a) is 10MHz (cycle time of 100ns) during which all oscillations eventually attenuate. Another step-current excitation is generated at the rising edge of the next shift cycle. The falling clock edge does not cause a sufficiently large excitation since no combinational logic toggles (only the clock-network). The frequency of power-supply oscillations matches the previously measured first resonance (Fig. 5).

Fig. 10(b) shows the supply oscillations at a shift-frequency of 20MHz (50ns cycle time). Now, the power-supply oscillations initiated at the rising clock edge do not have sufficient time to attenuate before the next rising edge. Consequently, the supply network experiences the effects of two current steps at the subsequent rising clock-edge. The first is the attenuated but time-shifted effect of oscillations initiated at the first rising edge of the clock that superimpose with new oscillations initiated at the second rising-edge of the clock.

This level of visibility into supply voltage noise during scan test is expected to allow rapid optimization of the conflicting goals of maximizing scan test speed and correlation with functional tests.



(a)



(b)

Figure 10. Zoomed in version of the scan-clock showing the presence of the rising clock-edge at the initiation of the ringing.

5. Electromagnetic Emanations Measurement on X-Gene2

In the absence of fine grain voltage monitoring we examined other ways for characterizing voltage virus. A prominent solution seems to be extracting voltage noise information from EM emanations. Analyzing EM emanations emitted from the a computing system has been used in related work for revealing cryptographic key information [10] as well as execution profiling [9].

5.1 Experimental setup and methodology for generating spectrum spikes

Fig. 11 shows our experimental setup. The setup consists of five main components, the tigershark board, an antenna for receiving the EM emanations, a stand to keep the antenna position stable, a spectrum analyzer for capturing the frequency spectrum and a PC workstation that communicates with the analyser for automating the measurements procedure. Note that we had to turn the board upside down in order to get the antenna closer to the CPU and get the best possible emanation readings.

The motivation behind using EM readings for voltage noise analysis is the following. A loop of instructions that executes continuously has a specific period which is determined by the loop iteration time. If inside the loop multiple activities exist that draw different power and produce different EM fields around the CPU, then the loop should cause a visible oscillation on the spectrum analyzer at frequency equal to 1/loop_period. For instance in the Fig. 11 we show a spike at 126MHz which was intentionally caused by a loop of instructions running on the CPU.

We follow the procedure for generating visible spikes on the spectrum analyzer on various frequencies. We find that EOR (bitwise xor operation) and MUL (multiplication) instructions when executed on the X-Gene2 draw significantly different power requirements (approximately 3 Watts difference). So we construct a loop that consists of a sequence of EORs followed by a sequence of MULs. All instructions are executed sequentially due to read after write dependencies that we force among instructions. We change the loop period by changing the number of EOR and MULS. Table II shows how we sweep the EOR and MULS to achieve different loop periods/frequencies. Note that EOR execute in one CPU cycle whereas MUL need 5 cycles, therefore MULs are always more than EORs.

Table ii. Sweep of MULS and EOR instructions to achieve a spike at specific frequency on the spectrum analyzer

LoopFrequency(MHz)	#of MULS	#of EORS
10.00	24	120
20.00	12	60
30.00	8	40
40.00	6	30
50.00	4	28
60.00	4	20
68.57	3	20
80.00	3	15
88.89	2	17
100.00	2	14
109.09	2	12
120.00	2	10
126.32	1	14
133.33	1	13
150.00	1	11
160.00	1	10
160.00	1	10
171.43	1	9
184.62	1	8
200.00	1	7

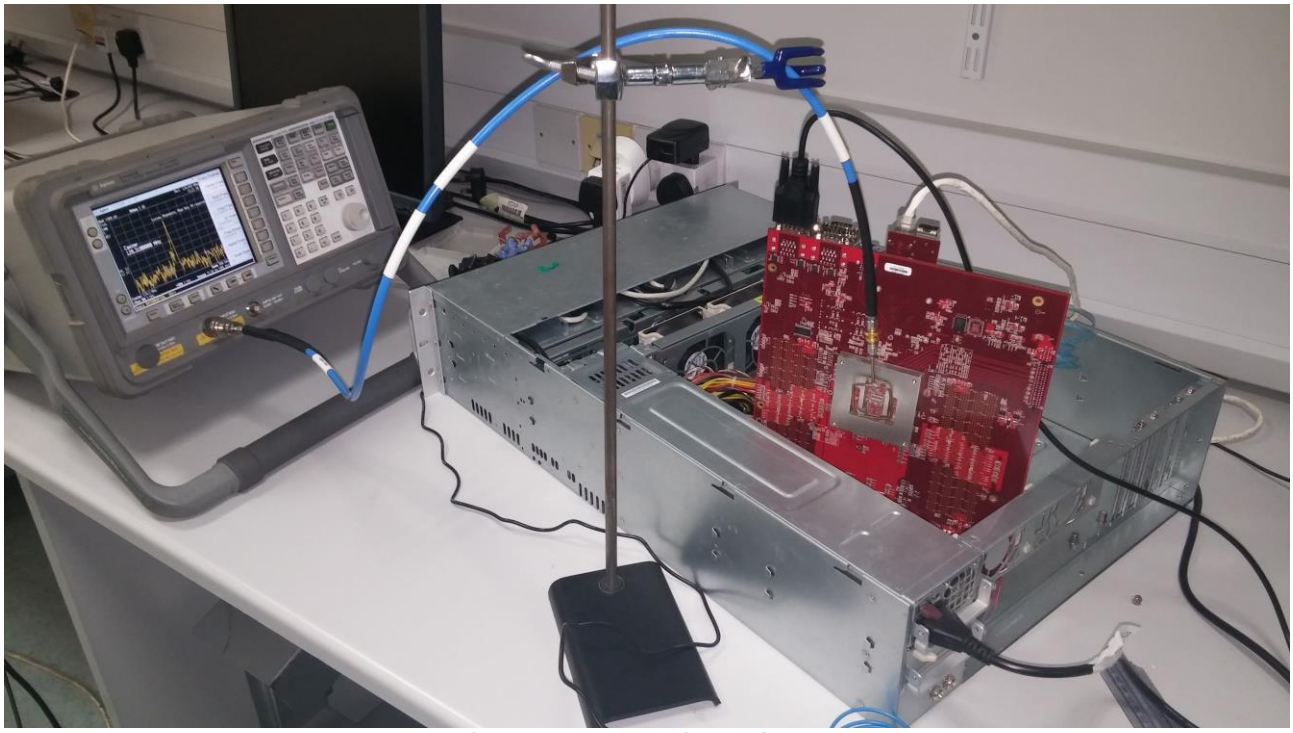


Figure 11. EM experimental setup

We are interested in the spectrum that the power delivery network resonant frequency is found. Which is usually between 1-200MHz [2]. Finding the resonant frequency is imperative for maximum voltage noise generation. A virus that switch from high to low power activity with a frequency equal to PDN resonant causes maximum voltage noise [2,6]. The alternation between EOR and MUL instructions actually tries to mimic this variation in activity. The loop frequency that produces the highest voltage noise should indicate the resonant frequency. Similar procedure is followed in [2] for identifying resonant frequency. Note that due to the difference in execution cycles among MUL and EOR in some cases 50/50 duty cycle cannot be achieved. Therefore, instead of 90MHz we actually achieve a frequency 88.89MHz or instead of 110MHz 109.09MHz.

We use the maximum amplitude recorded in the spectrum analyzer for identifying the resonant. Fig. 12 shows the maximum amplitude in pW recorded on the spectrum analyzer versus the loop frequency. The graph may imply that the resonant is located at 126MHz. The shape of the graph is very similar to what you expect from a resonant as it has a rather monotonic increase before the peak and a decrease after the peak. These findings are preliminary as the transmister receiver gain factors are not considered yet in the analysis.

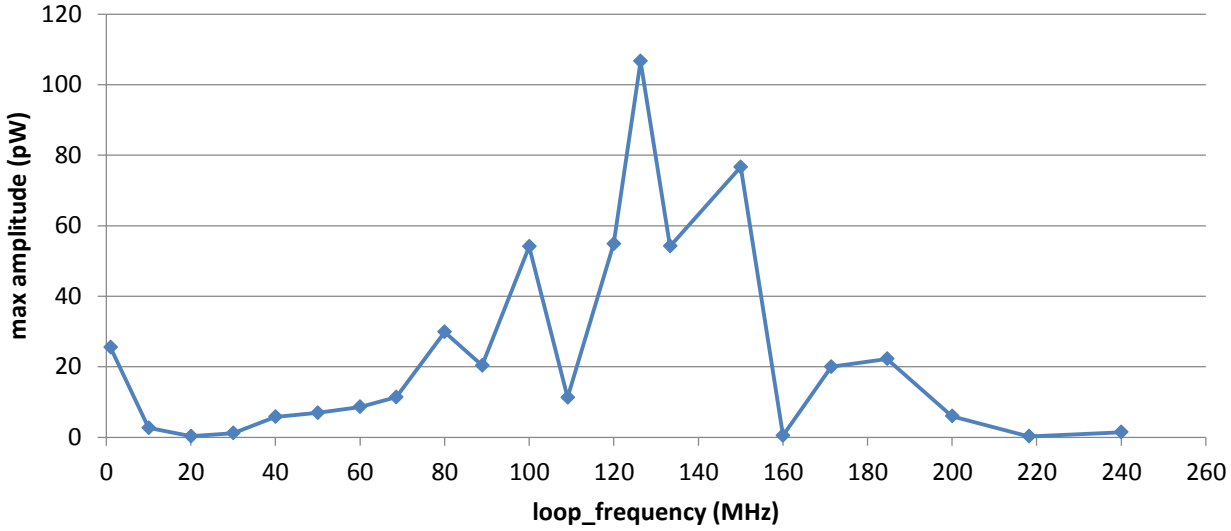


Figure 12. Signal amplitude vs different instruction loop frequencies

5.2 Examining the sources of EM signals

In this Section we report some other experimental findings based on EM emanations. We observe that changing the CPU voltage while keeping stable workload and CPU frequency doesn't impact the recorded amplitude. Fig. 13 shows that the average amplitude is rather constant and does not monotonically increase versus the CPU voltage. On the other side, the average CPU power consumption recorded using the HEI (described in Deliverable 4.1) increases with increasing CPU frequency. Also, Fig. 14 shows the amplitude in pW and the average power of the CPU for the various instructions loops introduced in Table II. It is clearly shown that the average power and the amplitudes recorded by spectrum analyzer do not follow the same trend. For instance 126MHz has the highest amplitude whereas the highest average power is found at 133MHz.

Lastly, the analysis reveals that the number of active cores impacts the amplitudes. Fig. 15 shows that as the number of active cores increases the amplitude increases as well. The figure shows the amplitude recorded when running an EOR/MUL targeting 126.31MHz loop on 1,4 and 8 cores. Voltage noise behaves similarly as shown in previous work [1,2,5].

We are still in the process of correlating the voltage noise with the EM readings. Nonetheless, the experiments in this Section indicate that a correlation is very likely to exist. This is work in progress and the future findings are expected to be reported in D3.5.

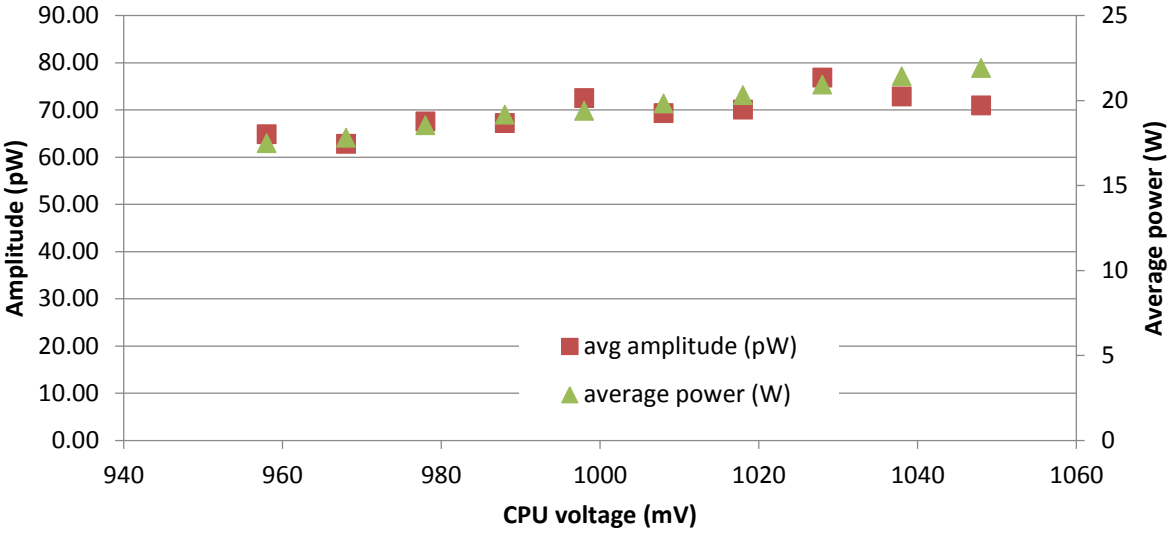


Figure 13. EM signal amplitude and average CPU power vs CPU voltage

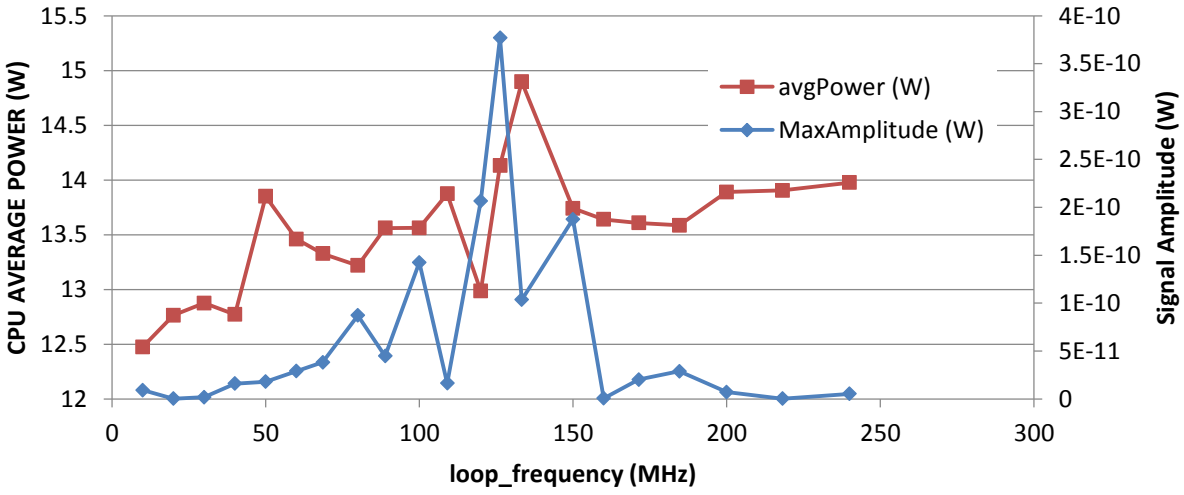


Figure 14. Average CPU power and signal amplitude versus loop frequency

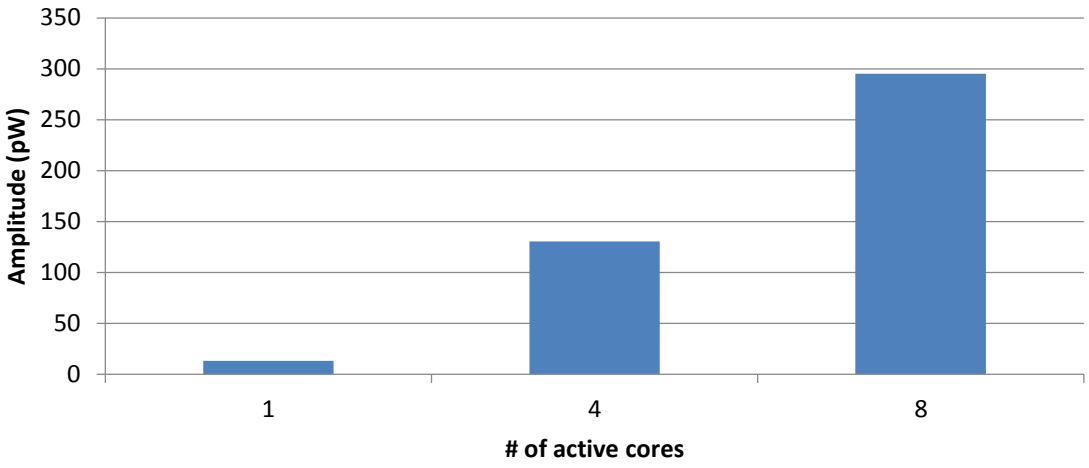


Figure 15. Em signal amplitude vs # of active cores

6 IPC and Thermal Virus Generation on the X-Gene2

The GA has proven to successfully generate voltage noise virus on Juno board. Unfortunately, the XGene-2 Tigershark board does not offer fine grain voltage noise visibility. Therefore, we cannot use the GA to generate a voltage noise virus for the Tigershark board. Still though, we use the GA to generate IPC and thermal viruses on the Tigershark board. This confirms that the GA framework can be used on the Xgene chips. We expect that we will be able to use the framework for generating maximum voltage noise on the X-Gene3 which will offer fine grain voltage visibility.

Fig. 16 shows how the single thread IPC virus compares to other benchmarks (SPEC,PARSEC and cloudsuite). The figure clearly shows that the IPC virus achieves but far the highest IPC. Fig. 17 shows how the temperature virus compares to other benchmarks. The virus achieves the highest temperature. Note that the temperature virus raises the temperature higher than the IPC virus. Also the temperature reported is average temperature during the virus execution. At some points the temperature surpassed the 100 Celsius which is considered by a lot of CPUs as the critical threshold [8].

Fig. 18 shows the main differences among the maxIPC and maxTemperature viruses. The maxTemperature virus has less IPC but consumes higher power compared to maxIPC virus. This is the result of maxTemperature having more memory instructions than integer operations. Memory instructions involve the memory hierrarch but they usually have higher latency than integer operations, hence, more power is drawn but less instructions per cycle are executed. We also noticed that temperature virus favored long latency instructions over short latency instructions, for instance FSQRT over FMOV and MUL over MOV.

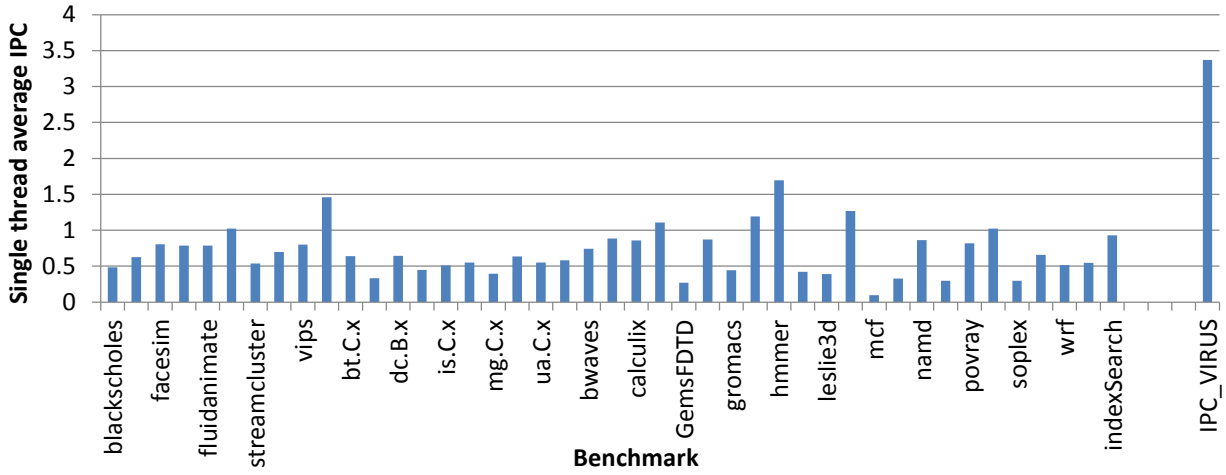


Figure 16. IPC of the virus vs conventional benchmarks

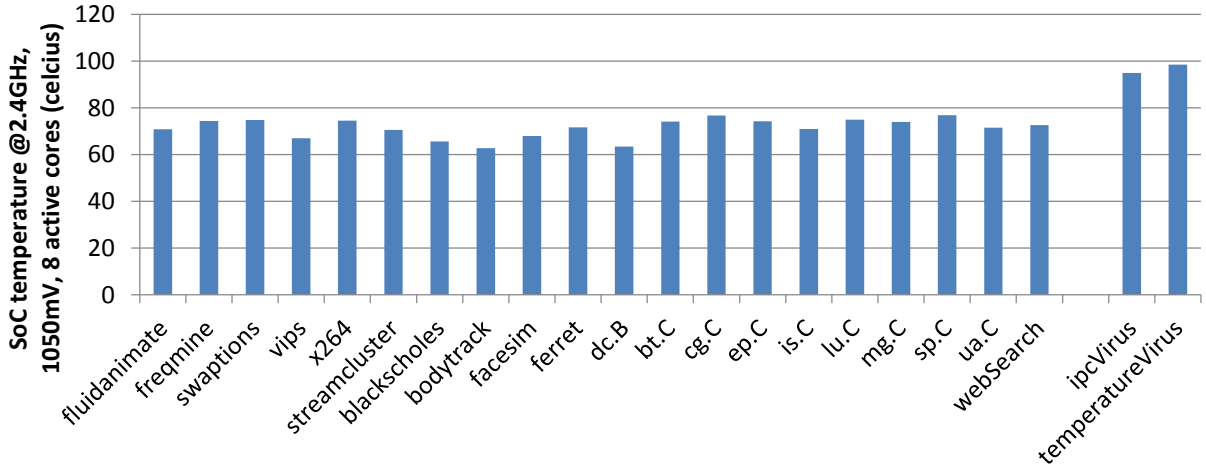


Figure 17. SoC temperature when running virus and conventional benchmarks

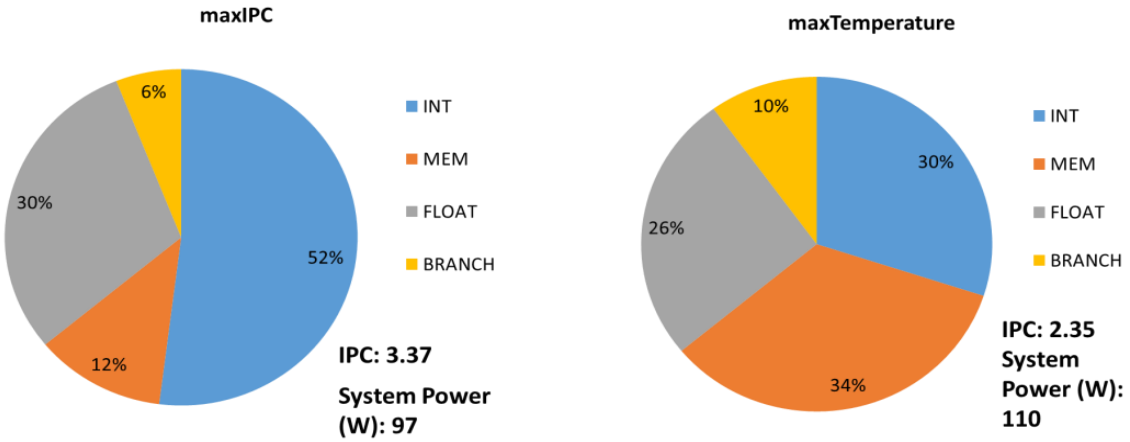


Figure 18. Comparison of instruction mix, IPC and power drawn of the viruses

7 Conclusions and Future Work

This deliverable attempts to stress the processor cores in order to expose the pessimistic voltage/frequency margins. A framework based on genetic algorithms is developed for generating stress tests. On a platform that supports on-chip voltage monitoring we show that at least a 10% voltage margin can be shaved for improved energy efficiency. The deliverable also describes a simulation and modelling analysis that is informed by our capability to directly measure on-chip supply voltage noise. We use the scan-shifting scenario as a particular case of worst-case supply voltage noise and present results. We applied the GA framework on the X-Gene2 board and successfully generated IPC and thermal viruses. The GA framework will also be applied on X-Gene3 when it becomes available for voltage noise characterization. In the absence of fine grain voltage noise monitoring capabilities, initial experimental analysis revealed that EM measurements seems a promising direction for voltage noise characterization.

Going forwards, we are planning on extending the usage of the Juno platform in order to qualify our EM measurement technique. Taking advantage of the OC-DSO, we can directly correlate the on-die noise with off-chip emanations. This can form the framework of a model that can be extended to any board, thereby laying the grounds of a generic characterization framework and will be used to drive the the noise characterization in the deliverable D3.5.

8 References

[1] Reddi, Vijay Janapa, et al. "Voltage noise in production processors." *IEEE micro* 31.1 (2011): 20-28.
[2] Kim, Youngtaek, et al. "AUDIT: Stress testing the automatic way." *Microarchitecture (MICRO), 2012 45th Annual IEEE/ACM International Symposium on*. 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., et al. "14.6 An all-digital power-delivery monitor for analysis of a 28nm dual-core ARM Cortex-A57 cluster." *Solid-State Circuits Conference-(ISSCC), 2015 IEEE International*. 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, et al. "Voltage noise: Why it's bad, and what to do about it." *5th IEEE Workshop on Silicon Errors in Logic-System Effects (SELSE), Palo Alto, CA*. 2009.
[8] http://www.intel.com/content/www/us/en/support/processors/000005597.html
[9] Sehatbakhsh, Nader, et al. "Spectral profiling: Observer-effect-free profiling by monitoring EM emanations." Microarchitecture (MICRO), 2016 49th Annual IEEE/ACM International Symposium on. IEEE, 2016.
[10] [Genkin, Daniel, et al. "Stealing keys from PCs using a radio: Cheap electromagnetic attacks on windowed exponentiation." International Workshop on Cryptographic Hardware and Embedded Systems. Springer Berlin Heidelberg, 2015.
[11] Paul Whatmough, Shidhartha Das, Zaccharias Hadjilambrou and David Bull. " Power Integrity Analysis of a 28nm Dual-Core ARM Cortex-A57 Cluster using an All-Digital Power Delivery Monitor" *Accepted to the Journal of Solid-State Circuits in January 2017.*
[12] Shidhartha Das, Paul Whatmough and David Bull. "System-Level Power Delivery Network Analysis for a Dual-Core Cortex-A57 Compute Cluster", Under Preparation for Submission to the Transactions on Circuits and Systems (TCAS-1)
[13] https://www.arm.com/files/pdf/Juno_ARM_Development_Platform_datasheet.pdf
[14] https://www.apache-da.com/products/- Redhawk and Sentinel-PSI

[END OF DOCUMENT]

