Catching the Flu: Emerging Threats from a Third Party Power Management Unit

Rajesh JayashankaraShridevi Chidhambaranathan Rajamanikkam
Koushik Chakraborty Sanghamitra Roy

USU BRIDGE LAB, Electrical and Computer Engineering, Utah State University
{rajesh.js, chidham}@aggiemail.usu.edu {koushik.chakraborty, sanghamitra.roy}@usu.edu

ABSTRACT

Power management units (PMU) have come into the spotlight with energy efficiency becoming a first order constraint in MPSoC designs. To cater to the exponential rise in power events, and to meet the demands of tight power and energy budgets, PMUs are evolving to more complex and intelligent designs. In an era defined by energy efficient computing, a malicious circuit embedded in a third party PMU can adversely affect the operation of the entire MPSoC.

This work presents and evaluates two covert security threats, *P-VIRUS* and *DROWSY*, designed using a malicious third party PMU. Further, we propose a non-invasive IP risk assessment module to monitor the trustworthiness of the deployed PMU. Our techniques help flag malicious activities in the PMU with less than 1% area and power overheads.

1. INTRODUCTION

The phenomenal growth and integration of transistor devices has reshaped our notion of reliable and trustworthy hardware. Several emerging trends in hardware integration and user computing practices are exposing new challenges for secure hardware design. First, high performance Multiprocessor System-on-Chips (MPSoCs) are emerging as an alternative to many general purpose and embedded processors. Second, these MPSoCs promote unprecedented integration of Third Party Intellectual Property (3PIP) components, for reducing cost and design complexity. Third, MPSoC integrators are reluctant to re-verify the procured 3PIP, due to sky-rocketing verification costs and aggressive time-to-market schedules [3]. Fourth, the 3PIP vendors are averse to expose their internal design RTL to the MPSoC integrator to protect their competitive advantages, which creates a stiff barrier to many existing hardware trust assurance techniques relying on gate-level introspection [19,21]. In this ecosystem, an MPSoC integrator cannot ensure full trustworthiness of every 3PIP.

In this context, security assurance of the 3PIP power management unit (PMU) has profound implications on future MPSoC designs. PMUs offer a host of services ranging from dynamic control of power rails, voltage scaling, to managing

power states. Traditional simple PMUs for low cost power conversion are evolving into feature rich, hardware controlled and intelligent power management IPs (PMIPs). PMIPs are expected to increase energy efficiency and provide flexible power management in high-end MPSoCs [17]. In fact, the global PMU semiconductor market has seen the fastest growth among all analog designs. PMUs accounted for nearly 50% of the total analog IC revenue in 2012 [9], and are projected to grow at over 7.87% during 2013-2018 [18]. These emerging trends can create a potent economic threat in the presence of a malicious PMU (M-PMU).

In this evolving PMU market, a key research question is *how can we assure security and trustworthiness of third party PMUs?* The evolution of this industry has created a dynamic, volatile and competitive landscape, resulting in many new suppliers entering the niche market, raising the possibility of 3PIP M-PMUs. A trojan embedded in a 3PIP M-PMU can cause sporadic errors, reliability issues, or even catastrophic failures. These attacks can lead to data corruption, denial of service, or degrade the system performance and energy efficiency. For example, a malicious increase in the supply voltage can cause a surge in the peak power and chip temperature, leading to thermal throttling or functional errors due to chip failure. In this work, we explore two specific attack models and security assurance techniques in the presence of a M-PMU. Our schemes forego reliance on popular trojan detection techniques that are likely to be ineffective for 3PIP trojans [19,21].

Our contributions in this paper are as follows:

- We trace the determinant of the cataclysmic security loophole in the PMU, and evaluate two covert attacks from the PMU: *P-VIRUS* and *DROWSY* (Section 3).
- We propose an *IP risk assessment module* (IPRAM) that can be implemented by the MPSoC integrator and realizes our security assurance techniques: a *P-VIRUS Monitor* (PM) and a *Wakeup Alarm* (WA) (Section 4).
- We present a thorough evaluation of the IPRAM. Our results show that PM and WA can effectively detect malicious PMU attacks, while incurring low overheads (less than 1%) in area and power (Section 6).

2. RELATED WORK

Recent works in hardware security range from protection against malicious circuit modifications in outsourced foundries to protection of IPs against side-channel attacks [19]. However, these mechanisms are ineffective to protect against malicious 3PIPs in an MPSoC. The concept of task duplication proposed by Chen et al. [12] is an important step in secur-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

DAC '16, June 05-09, 2016, Austin, TX, USA

Copyright 2016 ACM ISBN 978-1-4503-4236-0/16/06 \$15.00.

DOI: <http://dx.doi.org/10.1145/2897937.2897994>.

ing 3PIPs, but it is unable to protect against a M-PMU. The concept of security in power management is unfamiliar and is principally limited to traditional verification techniques. Existing literature involves power management robustness analysis [4], power grid verification [15], and thermal monitoring [13], to ensure functionality and reliability. *The central focus of this work is exposing the security vulnerability in power management—an uncharted territory.*

3. THREAT EVALUATION

In this section, we identify the loopholes in current and forthcoming power management solutions, and evaluate the impact of a M-PMU attack on MPSoC architectures. We outline the life cycle of a M-PMU trojan (Section 3.1), discuss trojan activation mechanisms (Section 3.2), and present two attacks: (a) PMU-Voltage driven Immunity Reduction and Unhealth Syndrome, *P-VIRUS* (Section 3.3) and (b) *DROWSY* (Section 3.4). We then discuss the potency and design footprints of these attacks (Section 3.6).

3.1 Life Cycle of a 3PIP M-PMU Trojan

We outline the sequence of phases to realize a generic hardware trojan in the 3PIP PMU.

- *Trojan Insertion*: A few key engineers on the PMU 3PIP team can insert the malicious circuit during the design or test phase [21]. We have seen many cases of disgruntled employees involved in unprofessional practices. The Volkswagen scandal exposed the presence of corporate sabotage [2]. Recently, a key player in the processor sector was accused of using its position to ward off competition [5]. In such scenarios, an unsuspecting IP vendor will end up supplying a trojan embedded IP to its clients.
- *Trojan Activation*: The attacker can employ a variety of subtle activation mechanisms that can either be triggered by the *internal* or the *external* environment [19]. Another class of trojan triggers involves the coalition of software-hardware, where an application running on the hardware can send a sequence of cryptic messages/requests to activate a dormant trojan in the 3PIP PMU [11].
- *Trojan Operation*: Once activated, the trojan can inflict a plethora of attacks on the on-chip components, adversely affecting the overall system behavior. In this work, we demonstrate two specific attacks that can cause a substantial degradation in energy efficiency and performance.

3.2 M-PMU Activation

The activation phase is particularly important to a trojan design. We envisage a software-hardware coalition based sequential trigger that takes advantage of the voltage change requests made by the on-chip components. Analogous to the idea presented by Waksman et al. (*sequence cheat code*) [20], we use a sequence of power event requests such as specific voltage identification (VID) patterns to trigger the underlying trojan. An unsuspecting software can be designed, to send a particular sequence of voltage change requests in the form of VID signals to the PMU. The embedded trojan in the 3PIP PMU monitors the staggered incoming requests for the complex sequence. In modern processors, VID signals are typically 6 to 8 bits and a sequence of these requests can potentially have an enormous state space resulting in astronomical test times. The complexity of trojan activation during testing is investigated in Section 6.1.

Req. V_{CC}	VID _{TF}	VID _{FEV}	V_{CC_F}
1.65V	74	7C - AC	1.73V - 2.02V
1.8V	87	8B - BB	1.88V - 2.02V

Table 1: VID manipulation example. Req. V_{CC} is the voltage requested by the core by sending VID_{TF} (Hex). The LFSR generates random values between 001 to 111 and manipulates the VID_{TF} to a value in the range of VID_{FEV} (Hex). V_{CC_F} is the range of infected voltage supplied to the core.

3.3 P-VIRUS

Our first attack model, *P-VIRUS*, targets the energy efficiency and performance of the MPSoC. The embedded trojan in the PMU, inconspicuously manipulates the supply voltage request made by a processor for a set frequency, leading to improper voltage-frequency (VF) assignments.

3.3.1 P-VIRUS - Attack Environment

Figure 1a, illustrates the environment of a *P-VIRUS* attack. The processor requests the PMU for a specific voltage level based on the decision made by a dynamic voltage frequency scaling (DVFS) control algorithm. For example, the Intel processors use a Serial Voltage IDentification (SVID) interface to communicate the VID requests to the PMU, which in turn decodes the VID and supplies the corresponding voltage to the the processor through the power rails. Specifically, in Intel's i7-4650 processor line, the 8-bit VID signal (00h-FFh) can request for a voltage ranging between 0 to 3.04V, though the specified processor operating voltage range is between 1.64V to 1.85V [10].

3.3.2 P-VIRUS - Attack Variants

We envisage three variants in *P-VIRUS*, based on the manipulation of the voltage request and the resulting behavior.

- *FRACTURE* ($V_{supply} < V_{request}$): The voltage supplied by the PMU is less than the requested voltage. The processor circuit cannot meet the timing constraints for the particular frequency due to the higher delay caused by the lower supply voltage, leading to functional errors.
- *FEVER* ($V_{supply} > V_{request}$): The supply voltage is higher than the requested voltage. The higher supply voltage will result in a loss of energy efficiency, and cause a rise in the on-chip temperature, similar to the onset of a fever. Such a subtle manipulation of the supply voltage can also keep the attack stealthy. To mitigate the elevated temperature, the system may lower the operating frequency, thereby degrading the processor performance.
- *STROKE* ($V_{supply} \gg V_{request}$): The supplied voltage is substantially higher than the requested voltage. The excess heat dissipation due to a sudden increase in the temperature can lead to a catastrophic failure and chip burnout, mimicking a stroke attack.

We model the *FEVER* variant to show that a modest voltage manipulation can significantly affect the system.

3.3.3 P-VIRUS - Implementation

Figure 1b shows the implementation of the *P-VIRUS*. The VID decoder block deciphers the incoming VID signal and the trojan manipulates this VID signal. To keep the attack stealthy, the degree of attack is constrained by altering only 3 bits (4, 5 & 6) of the VID in a random fashion using a linear feedback shift register (LFSR). The infected VID (VID_{FEV}) request is forwarded to the control circuit that manages requests from many on-chip components and instructs the reg-

Figure 1: Conceptual block diagram of a P-VIRUS attack. The PMU corrupts the voltage request sending a higher operating voltage to the processor. Figure 1b illustrates the manipulation of the VID signal to inflict a P-VIRUS attack.

Figure 2: Figure 2a shows the trends in the power management granularity. Figure 2b shows the DROWSY implementation. DROWSY is a random focused attack, where the blocks under attack vary at different epochs to keep the attack stealthy.

ulator to scale the voltage. The regulator outputs the scaled voltage to the power rail corresponding to the block that requested the voltage change. The *P-VIRUS* infected supply voltage value (V_{CCF}) corresponding to the VID_{FEV} is constrained within the $Max. Core V_{CC}^1$. The limitation and randomness of voltage manipulation obscures the attack, while ensuring that the attack is significant enough to cause a considerable degradation in system behavior.

Table 1 gives an example of the VID manipulation for an incoming request of 1.65V. A 3-bit LFSR generates a random number between 001 to 111, to inflict an inconsistent attack and circumvent the generation of identifiable patterns in system behavior. On adding it to VID_{TF} , we get a range of possible VID_{FEV} (Column 3). Note that the V_{CCF} is limited to a maximum of 2.02V ($Max. Core V_{CC}$) though the VID_{FEV} corresponds to a voltage range of 1.73V to 2.21V.

3.4 DROWSY

In our second attack, DROWSY, the M-PMU tampers with the sleep and wake up requests of the on-chip components. The attack mimics the effects of drowsiness, affecting the availability of on-chip resources.

3.4.1 DROWSY - Attack Environment

Figure 2a illustrates the emerging trend in the granularity of control for dynamic power management (DPM). Compared to the conventional independent power domains, the emerging architectures have control grains that are an order of magnitude smaller [17]. In densely integrated modern MPSoCs, the number of independently controllable blocks rise to hundreds, thereby increasing the complexity and latency involved for run-time monitoring of individual blocks. DPM techniques involve selectively turning off the unused components, based on the speculated idle time and the history of execution patterns. In conventional architectures, the

¹Max. Core V_{CC} is the voltage beyond which permanent damage is likely.

operating system (OS) is responsible for the idle state management of blocks. To achieve the fine grain control in emerging architectures, a layer of hardware control in the form of PMIP is introduced, in addition to the OS control.

3.4.2 DROWSY - Attack Variants

We envisage several incarnations in the DROWSY attack:

- *Delayed Sleep:* The PMU delays the sleep signal, allowing the components to be active even when unused, thereby degrading the energy efficiency of the system.
- *Delayed Wake up:* The delay in wake up can result in resource unavailability. Latency sensitive applications will suffer from performance degradation.
- *Abrupt Sleep:* Abrupt transition of blocks to sleep states results in loss of data and performance degradation.

In this paper, we demonstrate the *delayed wake up* scenario.

3.4.3 DROWSY implementation

Figure 2b shows the operation of the DROWSY attack. When the on-chip resources are idle, they are put to sleep to save power. The control circuit manages the requests coming from the PMIP or the OS and puts the corresponding block to the sleep state. When the OS requests for the resource to wake-up and resume operation, the PMU delays the response to this request, adversely affecting the application performance. During an epoch, one or many blocks may send the wake-up request to the control circuit in the PMU. A *block select* module randomly chooses a subset of blocks to inflict a sluggish response. The response to these blocks are delayed by a random, but bounded time, using the delay block fed by the LFSR. The randomness in attack, the constraint in delay threshold, and the incoherent focus on a subset of blocks, obscure the DROWSY attack.

3.5 Evaluation Methodology

We evaluate the effects of the trojan using Sniper6.0 [6]. We

Figure 3: Effect of the P-VIRUS trojan on the peak power, energy and performance of applications. Peak power increases as a result of increased operating voltage. Energy increases due to the increased power and degradation in application performance. Performance degrades due to the thermal throttling to ensure chip safety.

Figure 4: Performance degradation due to DROWSY.

model the Intel i7-4650U processor and inject the trojan in it. To model the effect of *P-VIRUS*, we tamper with the voltages of processor performance-states (p-states), while running the Splash2 benchmarks. A widely-used *on-demand* DVFS algorithm is used for p-state assignment. The frequency is throttled down when the dynamic power (thermal load) rises beyond the set Thermal Design Power (TDP), to shun assignments that can lead to a chip burn-out.

To model the effect of *DROWSY*, we repurpose the cost associated with sleep periods and add an extra latency whenever a block resumes from a sleep state.

3.6 Potency and Design Footprint

We evaluate the potential of the threat in terms of increase in energy consumption, peak power and degradation in performance (Section 3.6.1). We also present the design footprint to realize these trojans in the PMU (Section 3.6.2).

3.6.1 Potency

P-VIRUS: Figure 3a shows the comparison of peak power between the *P-VIRUS* free and *P-VIRUS* infected executions. On an average, the peak power increases by nearly 19% with *ocean.cont* incurring the highest increase in peak (23%). The rise in peak power is limited by the thermal design power (TDP), beyond which the operating frequency of the processor is throttled down to prevent a thermal failure.

Figure 3b shows the impact of *P-VIRUS* on the energy consumed. On an average, the energy consumed increases by 18%. We observe an anomaly for the *ocean.cont* benchmark, which incurs a meager 5% increase in energy. Figure 3c, reveals that *ocean.cont* does not incur any performance degradation. For this benchmark, the processor is able to operate at a higher voltage without breaching the set TDP. The loss in energy efficiency is purely due to the increased power. For the other benchmarks, Figure 3c shows the degradation in processor performance due to thermal throttling (up to 26%). Overall, the trojan degrades the system behavior in all the three measured metrics.

DROWSY: Figure 4 shows the performance degradation due to *DROWSY*. The sluggish responses to wake-up requests increase the number of idle cycles, resulting in a maximum performance degradation of 59% (average across benchmarks = 34%). The blocks in sleep or those under transition cannot be accessed, thereby preventing any functional errors.

3.6.2 Design Footprint

To evaluate the footprint of *P-VIRUS* and *DROWSY*, we augment the PMU RTL of the OpenSPARC T2 processor [14]. We implement the logic discussed in Sections 3.3 and 3.4, and synthesize with the TSMC 45nm library using the Synopsys Design Compiler. *P-VIRUS* incurs area and power overheads of 1.39% and 1.06%, respectively. *DROWSY* incurs an overhead of 1.84% in area and 1.92% in power.

4. M-PMU DETECTION

In this section, we explore a novel *Intellectual Property Risk Assessment Module* (IPRAM), *designed by the MPSoC integrator and placed at the interface of critical 3PIP blocks*. The IPRAM assumes no support from the 3PIP PMU vendor and effectively detects malicious activities in PMUs, across various vendors. The IPRAM consists of two low complexity blocks: (a) *P-VIRUS* Monitor (Section 4.2) and (b) Wakeup Alarm (Section 4.3).

4.1 IPRAM OVERVIEW

Figure 5a shows the block diagram of the proposed IPRAM. The outgoing power management requests from the 3PIP blocks are in parallel sent to the IPRAM and the corresponding response from the PMU is forwarded to it too. Since the IPRAM is placed in parallel to the communication between a 3PIP core and the PMU, it does not add to the latency of the power management state changes. The novel blocks in the IPRAM match the requests to responses and identify the presence of a malicious activity in the PMU's response.

4.2 P-VIRUS Monitor

P-VIRUS Monitor (PM) is based on the insight that *the supply voltage (V_{cc}) of a digital circuit profoundly influences its delay*. By characterizing the delay of a known circuit, we can estimate the imposed supply voltage on the block.

Figure 5a presents the interface of *PM*, modeled as a delay estimation block (*DEB*). Figure 5b shows a detailed operation of the *DEB*. The control unit power gates the *DEB* when not in use, to curtail the power consumption overhead. A series of cascaded delay buffers (DB) in the form of a tunable replica circuit are used to estimate the parasitic delay of the

Figure 5: Conceptual overview of the IPRAM. The IPRAM is designed in-house by the MPSoC integrator and placed at the interface of a 3PIP block to be monitored. Figures 5b and 5c show the detailed designs for the detection of P-VIRUS and DROWSY attacks.

circuit [7]. The cascaded buffers are sampled at equal epochs to capture the state transition at different stages and thereby estimate the delay for the imposed V_{cc} . The estimated delay is correlated to the values stored in the look up table to obtain the VID corresponding to the delay. The look up table is filled, based on the post-silicon characterization. The identified VID is then matched to the VID request in the comparator unit, by ignoring the 2 least significant bits to grant a margin for error. If the values do not match, the DEB flags an anomalous behavior.

4.3 Wake Up Alarm

Wake Up Alarm (WA) is a finite state machine (FSM) to observe the arrival of requests and responses of the sleep state transitions. Delayed and unprompted state changes triggered by the malicious PMU can be detected.

WA is modeled as a response audit block (RAB) in Figure 5a, where the request sent to the PMU is also sent in parallel to the RAB. Figure 5c shows a detailed diagram of the RAB. MPSoC blocks usually have multiple sleep states, varying in state transition times and power saving capabilities. The control circuit in the RAB feeds a multiplexer to select the appropriate delay associated with the sleep-to-wake transition. This delay is imposed on the request and forwarded to the capture circuit, that consists of a FSM with 3 states, *default* (D), *wait* (W) and *anomaly detected* (T). The FSM is in the D state until it detects a signal from the 3PIP. The state transitions are discussed below.

- $D \rightarrow W$: On receiving a wake-up request from the 3PIP, the state changes from D to W.
- $W \rightarrow D$: If the PMU response is observed within the vendor specified response time, it triggers a transition from W to D. The 3PIP resumes its operation.
- $W \rightarrow T$: If the PMU response is delayed, the delay block triggers a transition from W to T, flagging the anomalous behavior—a *delayed wake up* attack.
- $T \rightarrow D$: Once the attack is flagged, the signals and monitor blocks are reset, and the state transitions to the default (D) state to wait for the next request.
- $D \rightarrow T$: An unsolicited shutdown/wake-up signal from the PMU, is captured and flagged—an *abrupt* attack.

5. METHODOLOGY

In this section, we discuss the methodology used to evaluate the efficacy and overhead of the IPRAM.

PM: To evaluate the efficacy of PM, we model the cascaded buffers present in the DEB, in HSPICE and obtain the delay characteristics for different V_{CC} values. To account for process variation, we include a Monte Carlo analysis by creating

SL	Area	Power	TP	Time (years)
8	0.38%	0.17%	5.42×10^{-20}	2.92×10^8
16	0.87%	0.40%	2.94×10^{-39}	5.40×10^{27}
64	3.40%	2.10%	7.46×10^{-155}	2.13×10^{143}

Table 2: Area and power overhead, and state space exploration time for the proposed trigger. TP represents trigger probability.

a Gaussian distribution of three parameters: threshold voltage, effective channel length and transistor width [16].

WA: Since WA identifies DROWSY via a FSM, based on the incoming signals, we evaluate WA by implementing a Verilog RTL model. We use Xilinx ISIM to perform exhaustive tests and observe the behavior for different scenarios.

We implement the blocks of IPRAM in Verilog RTL and synthesize the RTL with the TSMC 45nm library using Synopsys Design Compiler to find the design overheads.

6. EXPERIMENTAL RESULTS

In this section, we evaluate the complexity of trojan activation during the pre-deployment test (Section 6.1). We then present the results for PM and WA based on two metrics: efficacy (Section 6.2) and design overhead (Section 6.3).

6.1 Limitations of Post Silicon Testing

The need for IPRAM arises from the limitations of post-silicon testing, as demonstrated in Table 2. Our evaluation reveals the magnitude of complexity involved in *guaranteed* trojan activation during testing [8]. While testing a PMU applying test patterns (VID), and observing the resulting voltage levels, one must wait for voltage stabilization before running subsequent tests. Typical voltage level stabilization times lie between 0.1 to a few milliseconds [1]. With these considerations, we present the data for an 8-bit VID and *sequence lengths* (SL) of 8, 16 and 64 signals. For the SL of 8, we see a meager area (0.38%) and power (0.17%) overhead, while the state space exploration time is enormous. The data shows the extremely low probability of activating the M-PMU during the post-silicon tests, thereby emphasizing the need for the proposed security assurance techniques.

6.2 Solution Efficacy

PM: Figure 6 illustrates two representative cases of PM. In Figure 6a, we see the worst case scenario for detection, where the difference between V_{CC_F} (1.73V) and V_{CC} (1.65V) is a mere 0.08V. This is the lowest increment in voltage that the FEVER can impose (Section 3.3). The overlap in the distribution implies that these delay values can occur both in the presence and absence of a P-VIRUS attack leading to an inaccuracy in trojan detection (false detection). Figure 6c represents a typical case, where P-VIRUS increases the voltage

Figure 6: Results from the PM. 2 representative cases of the delay characteristics for P-VIRUS detection. Figure 6a, shows the worst case when the difference between the infected voltage (V_{CC_F}) and the requested value (V_{CC}) is minimum (0.08V). Figure 6c shows the typical case ($V_{CC_F} - V_{CC} = 0.1V$). Figures 6b and 6d show the FP and FN analysis for various threshold values.

Figure 7: Important scenarios of WA. Figure 7a illustrates the correct behavior. Figure 7b shows a delayed response attack detection. Figure 7c detects unsolicited response.

by 0.1V. For attacks with a voltage increment greater than 0.1V, the distribution curve is distinct (no overlap), implying that there exists an optimum delay threshold with zero false detections.

Threshold Selection: Delay threshold indicates the minimum bound on the delay that can occur for a chosen voltage. If a voltage higher than the requested value is supplied, the obtained delay value will ideally be below the threshold. Smart threshold selection helps reduce the number of false positives (FP) and false negatives (FN). Figure 6b, shows the FP and FN analysis of the worst case. With an optimum threshold of 1.22ns, the FP is a mere 1% and FN is 4%, giving a false detection rate (FDR) of 5%. As we move away from the optimum threshold, the FDR increases rapidly. In the typical case, the tolerance for error in optimum threshold selection increases (Figure 6d).

WA: Figure 7 shows the three important cases of DROWSY detection. The states 00, 01 and 11 indicate the states D , W and T , respectively. Figure 7b illustrates an attack scenario. The 3PIP core sends a wake-up request, but the response is delayed by the PMU. WA flags the trojan, once the vendor specified time expires without a PMU response.

6.3 Implementation Overhead

PM and WA incurs area overheads of 0.97% and 0.14%, respectively. The power overheads are negligible, as blocks in the IPRAM are power gated when not in use.

7. CONCLUSION

This work outlines a novel security threat stemming from a malicious third party PMU that can adversely impact the MPSoC's system behavior. To detect anomalous M-PMU behavior, we propose a low complexity IPRAM. Our modules incur marginal overheads and have a FDR of 5%, while detecting anomalous behaviors inflicted by the M-PMU.

Acknowledgments

This work was supported in part by National Science Foundation grants (CNS-1117425, CAREER-1253024, CCF-1318826, CNS-1421022, CNS-1421068). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

8. REFERENCES

- [1] Single-Phase Wide VIN Range DC/DC Controller for Intel IMVP-6/IMVP-6.5 CPUs. <http://cds.linear.com/docs/en/datasheet/3816f.pdf>.
- [2] AUERBACH, D. Volkswagen's Villains. http://www.slate.com/articles/technology/bitwise/2015/10/volkswagen_s_emissions_scandal_has_a_villain_and_it_s_the_not_the_people.html.
- [3] BAILY, B. AND OTHERS Scalable Verification: Evolution or Revolution?: Panel Discussion on Test and Verification. Session in DAC, 2015.
- [4] BEMBARON, F. AND OTHERS Low power verification methodology using upf. *Proceedings of DVCon* (2009), 228–233.
- [5] BRILL, J. The intersection of consumer protection and competition in the new world of privacy. *Competition Pol'y Int'l* 7 (2011), 7–313.
- [6] CARLSON, T. E. AND OTHERS Sniper: exploring the level of abstraction for scalable and accurate parallel multi-core simulation. In *Proc. of SC* (2011).
- [7] CHAKRABORTY, A. AND OTHERS Dynamic thermal clock skew compensation using tunable delay buffers. In *ISLPED* (2006), pp. 162–167.
- [8] CHAKRABORTY, R. S. AND OTHERS MERO: A Statistical Approach for Hardware Trojan Detection. In *CHES* (2009), pp. 396–410.
- [9] GARTNER INC. *Competitive Landscape: Power Management IC and Power Semiconductor Vendors*. Gartner Research, 2012.
- [10] INTEL. Intel Core i7-4650U Processor. <http://ark.intel.com/products/75114/> Intel-Core-i7-4650U-Processor-4M-Cache-up-to-3-30-GHz.
- [11] KING, S. T. AND OTHERS Designing and Implementing Malicious Hardware. vol. 8, pp. 1–8.
- [12] LIU, C. AND OTHERS Shielding heterogeneous MPSoCs from untrustworthy 3PIPs through security-driven task scheduling. In *Proc. of DFT* (2013), pp. 101–106.
- [13] MESA-MARTINEZ, F. J. AND OTHERS Power model validation through thermal measurements. In *ISCA* (2007), pp. 302–311.
- [14] NAWATHE, U. G. AND OTHERS Implementation of an 8-core, 64-thread, power-efficient sparc server on a chip. *J. of Solid-State Circ.* 43, 1 (2008), 6–20.
- [15] NIZAM, M. AND OTHERS Power grid voltage integrity verification. In *ISLPED* (2005), pp. 239–244.
- [16] SARANGI, S. AND OTHERS VARIUS: A Model of Process Variation and Resulting Timing Errors for Microarchitects. *IEEE Trans. on Semiconductor Manufacturing* 21 (2008), 3–13.
- [17] SONICS INC. ICE-Grain Power Architecture for Mainstream SoC Designs-the Semiconductor IP Industry's First Power Management Solution, 2015.
- [18] TECHNAVIO. *Global Power Management Semiconductor IC Market 2014-2018*. TechNavio, 2014.
- [19] TEHRANIPOOR, M., AND WANG, C. *Introduction to Hardware Security and Trust*. Springer, 2011.
- [20] WAKSMAN, A., AND SETHUMADHAVAN, S. Silencing Hardware Backdoors. In *Proc. of Security and Privacy* (2011), pp. 49–63.
- [21] WAKSMAN, A. AND OTHERS FANCI: identification of stealthy malicious logic using boolean functional analysis. In *Proc. of CCS* (2013), pp. 697–708.