ADVERSARIAL VOLTAGE TRANSIENTS IN MULTI-TENANT FPGA ENVIRONMENTS

by

Andrew J. Gerber

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Computer Engineering

Approved:

Koushik Chakraborty, Ph.D. Major Professor	Sanghamitra Roy, Ph.D. Committee Member
Chris Winstead, Ph.D. Committee Member	Jonathan Phillips, Ph.D. Committee Member

D. Richard Cutler, Ph.D.
Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2025

ABSTRACT

Adversarial Voltage Transients in Multi-Tenant FPGA Environments

by

Andrew J. Gerber, Master of Science

Utah State University, 2025

Major Professor: Koushik Chakraborty, Ph.D.
Department: Electrical and Computer Engineering

This research investigates the impact of adversarial voltage transients in multi-tenant FPGA environments. By exploiting the shared power delivery system, a malicious design can induce voltage drops, causing timing errors in a victim AI application. The study utilizes the Xilinx DPU IP to run YOLOv3 while employing power plundering circuits built using multiple single inverter ring oscillators to induce voltage transients. A successful denial of service (DoS) attack is demonstrated, resulting in a total system crash that necessitates power cycling the board. Although the attack did not modify the outputs of the model as intended, the design process and results analysis provided critical insights into current and future countermeasures. Most significantly, it explores the tamper monitoring function integrated into all AMD Xilinx Zynq UltraScale+ devices. To the best of the researcher's knowledge, this security feature has not yet been discussed in the field of multi-tenant FPGA adversarial voltage transient attacks.

(30 pages)

PUBLIC ABSTRACT

Adversarial Voltage Transients in Multi-Tenant FPGA Environments

Andrew J. Gerber

A Field Programmable Gate Array (FPGA) is a special type of computer chip that can be reconfigured to implement a nearly unlimited number of functions. Recent trends have led some companies to offer cloud-based FPGA solutions. Researchers are exploring how to properly secure multi-tenant environments where the designs from two or more customers are placed on the same FPGA with logical and spatial isolation, though multi-tenancy is not yet commercially available in cloud FPGAs. The digital circuits within an FPGA require a clock to synchronize timing within the design. The maximum speed that this clock can run at is determined by the amount of logic in a single clock cycle, quality of the silicon chip, supply voltage, and several other factors. Furthermore, if the voltage is reduced without reducing the clock speed, the device will encounter *timing errors* and produce erroneous outputs.

This work attempts to cause a voltage-induced timing error in a multi-tenant FPGA environment where one design is a victim AI application and the other is a malicious *power plundering* design intended to cause short disruptions in the supply voltage called *voltage transients*. This is done by dramatically increasing the power usage of the malicious design over a very short period of time. Although the power supply is capable of delivering this much power, it takes time for it to adjust to the increased demand. This brief period, where the power supply has not caught up with the demand, causes a *voltage drop* which can lead to timing errors if severe enough. The opposite, a *voltage spike*, occurs when power usage dramatically decreases.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisors, Dr. Chakraborty and Dr. Roy, for their invaluable guidance and support throughout the writing of this work. Their expertise, encouragement, and insightful feedback have been instrumental in shaping this work. I am truly grateful for their patience and dedication, which have greatly contributed to my academic and personal growth.

Andrew J. Gerber

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACRONYMS

ASIC	Application Specific Integrated Circuits
CNN	Convolutional Neural Networks
CSU	Configuration and Security Unit (Zynq UltraScale+ devices)
DECAP	Decoupling Capacitor
DNN	Deep Neural Network
DoS	Denial of Service
DRC	Design Rule Check
FaaS	FPGA as a Service
FPGA	Field Programmable Gate Array
LLM	Large Language Model
LUT	Lookup Table
ML	Machine Learning
OVP	Over-Voltage Protection
PDS	Power Delivery System
PG	Power Grid
PL	Programmable Logic (Zynq devices)
PMBus	Power Management Bus
PMIC	Power Management Integrated Circuit
PMU	Platform Management Unit (Zynq UltraScale+ devices)
PPC	Power Plundering Cell
PS	Processor System (Zynq devices)
RO	Ring Oscillator
TDC	Time to Digital Converter
TPU	Tensor Processing Unit
UVP	Under-Voltage Protection

CHAPTER 1

INTRODUCTION

Artificial Intelligence (AI) is a broad term that has been used with both vague and specific meanings. For simplicity, this work will use the term AI to refer to the broad collection of Machine Learning (ML), Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), Large Language Models (LLM), and all other such models and algorithms. While different models have different architectures and designs, the core calculations come down to matrix operations. This work focuses on disrupting these calculations by inducing timing errors via adversarial voltage transients during calculations or model loading.

As the adoption of AI grows, and the size of the underlying models increases, the training and use of AI has become computationally very expensive. Using a general-purpose CPU is infeasible for such computations. More specialized hardware is required to run meaningful models in a reasonable amount of time.

A popular choice for running AI applications is using General-Purpose Graphics Processing Units (GPGPU), more commonly referred to as simply a GPU. This hardware, originally designed to process image arrays (i.e. matrices), excels at performing large numbers of matrix operations across large datasets. This is exactly what AI applications require.

Other, more specialized hardware used for AI are custom Application Specific Integrated Circuits (ASIC) designed from the ground up for running AI applications. One example of such a product is Google's Tensor Processing Unit (TPU). Such products can provide best-in-class performance but are very expensive. The cost of designing and manufacturing large, complex chips has increased exponentially with advancing technology nodes. This cost provides a significant barrier to entry for all but the largest of companies.

Field Programmable Gate Arrays (FPGA) provide a middle ground between GPUs and AI ASICs. They can provide higher performance-per-watt compared to GPUs, without the need for the substantial financial investment and extensive engineering expertise required

for large custom ASICs. Both FPGAs and ASICs can implement the same digital designs. Although the performance of a design implemented on an FPGA is lower than that of an ASIC, the development cycle for FPGAs is significantly less complex.

1.1 Multi-Tenant Cloud FPGA Solutions

Some cloud providers have started offering single-tenant FPGA solutions to customers for running complex workloads such as AI applications. A non-exhaustive list of cloud providers offering FPGA as a service (FaaS) includes Amazon Web Services, Microsoft Azure, Google Cloud Platform, IBM Cloud, and Alibaba Cloud. Because virtualization and containerization has been essential to the scaling of cloud services, researchers are investigating how to securely enable cloud FPGA solutions to share a single hardware platform between multiple clients, multi-tenancy. FPGA vendors have already added support for partial dynamic reconfiguration which allows partitions within the design to be reprogrammed while the rest of the design continues operating. This feature can be used to co-locate designs from multiple customers on a single FPGA.

As with software applications running alongside other untrusted applications, there are potential dangers inherent with multi-tenant FPGAs. Side-channel attacks have become increasingly common in software running on general-purpose CPUs where it is common for untrusted programs to run side-by-side. This is far less common in the world of digital hardware design. While it is true that IP designs often come from multiple sources, they can be chosen selectively and be thoroughly vetted. However, in the case of multi-tenant FPGAs, an entirely untrusted design could be running alongside and using shared resources. From this, multi-tenant FPGA attacks have emerged as a new area of research.

1.2 Hardware Threat Model

Multi-tenant FPGA side channel attacks take advantage of the co-location of the victim and attacker designs. All user partitions in a multi-tenant FPGA share the same Power Delivery System (PDS), which can be exploited to cause harm to the victim. Figure 1.1 illustrates this hardware threat model.

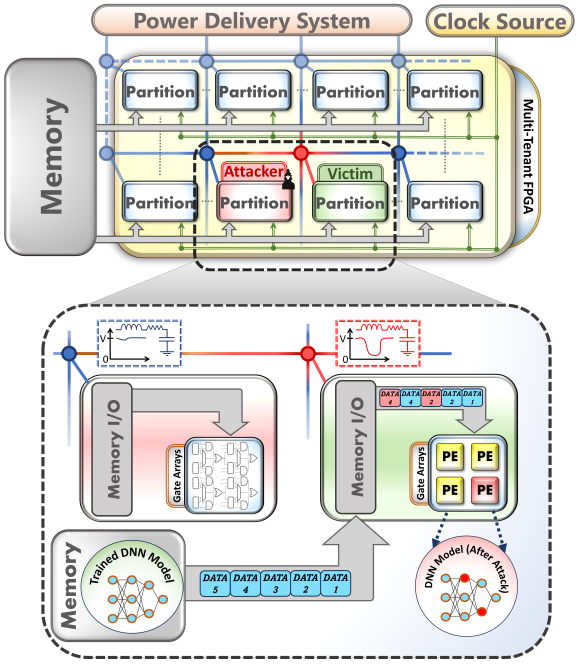


Fig. 1.1: Hardware threat model

The attacker can cause voltage transients in the PDS which then affects the victim design. For details of how the voltage transients are induced, see Section 3.2.1. These voltage transients, specifically the voltage droop, cause an increase in propagation delay. Because the clock source is driven by a different, unaffected voltage rail, the design can begin encountering timing errors, depending on the severity of the voltage droop. An AI model running on an FPGA accelerator can be compromised by strategically timed voltage transients. One method of achieving this is to use voltage transients to cause data duplication while weight matrices are streamed from RAM into the accelerator [1].

1.3 Power Delivery System

This research used an FPGA from the world's leading vendor, AMD Xilinx, though products from other vendors will be similar in many ways. AMD Xilinx FPGAs utilize multiple voltage rails for different parts of the chip. A few of the primary rails found in Xilinx Zynq products power the programmable logic (VCCINT), processor system (VCCINTPS), I/O, and Block RAM. A voltage transient in one of these domains should not affect the

others, as the power networks on the chip are isolated. The voltage of each rail is typically different than the others but supply power to different parts of the chip, though some do share a common voltage level. The VCCINT and VCCINTPS domains share a common nominal voltage but are fully isolated on chip. As per the AMD Xilinx design guides, these two rails can be powered by a single supply and be connected at the board level, as is the case for the ZCU104. This design will allow voltage transients from the VCCINT domain to affect the VCCINTPS domain.

The PDS of the board is responsible for providing a clean voltage supply to the components. A clean voltage is constant with low ripple, no matter the power demands of the circuit. Many PDS use a switching regulator to step down a higher voltage to the lower voltage required by advanced process technologies. A simple switching regulator circuit is shown in Figure 1.2. The digital controller senses the voltage at the output and adjusts the pulse width modulation controlling the switching MOSFET. The inductor decouples the two voltages and provides a supply and sink to maintain a consistent output current while the MOSFET switches on and off. The digital controller used dictates a finite switching frequency. Within a single period of a switching frequency, the regulator is unable to adjust its output to meet changing current demands from the circuit. For example, a sudden increase in device current results in a voltage droop until the next on-period of the MOSFET. The switching frequency of the ZCU104 VCCINT supply was measured to be approximately 800 KHz.

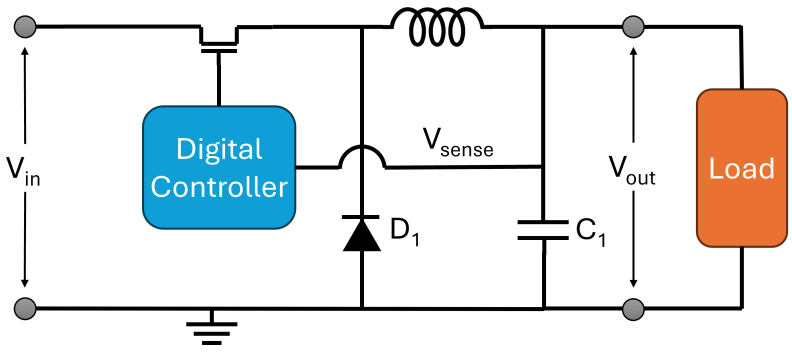


Fig. 1.2: Digital switching regulator circuit

To smooth the output of the voltage regulator, decoupling capacitors are placed between the regulator output and ground to decouple the current through the inductor and the current demands of the load thus maintaining a constant voltage. The size of these capacitors determines the tolerable current increase within a switching cycle.

Once the supply voltage is delivered to the input pins of the chip, power distribution is handled by a Power Grid (PG) constructed using the interconnect metal layers. A primary concern of PG design is to keep the voltage drop caused by high current through the resistive metal wires, called IR drop, within tolerable levels for both static (i.e. constant) and dynamic (i.e. transient) power. In-silicon Decoupling Capacitors (DECAP) are used in addition to the discrete capacitors on the PCB. These DECAP cells are placed throughout the chip and help to prevent localized voltage transients.

When a large, sudden increase in power consumption occurs, it is the responsibility of this hierarchy of decoupling capacitors to maintain the required voltage levels. Even if the power supply is capable of delivering that much constant power, decoupling capacitors on the PCB and within the chip are still necessary to maintain voltage levels during these sudden changes.

1.3.1 ZCU104 PDS and Voltage Rails

The AMD Xilinx Zynq UltraScale+ MPSoC ZCU104 Evaluation Kit used in this project's experiment combines several of the chip's voltage rails into a single rail on the board. This ties together these otherwise separate power domains and allows voltage transients to affect all connected domains. The primary internal power domains, PS low-power, PS high-power, and PL all share a common voltage rail at the board level. This allows adversarial voltage transients from the PL domain to affect the processor system. Table 1.1 shows a summary of all the voltage rails of the device.

ZCU104 Voltage Rails			
Chip Rail	**Board Rail**	**Voltage**	**Description**
VCC_PSINTLP	VCCINT	0.85 V	PS low-power
VCC_PSINTFP			PS full-power
VCC_PSINTFP_DDR			PS DDR
VCCINT			PL internal
VCCBRAM			PL Block RAM
PS_MGTRAVCC	MGTRAVCC	0.85 V	PS-GTR
VCCINT_VCU	VCCINT_VCU	0.90 V	VCU internal
VCC_PSPLL	MGT1V2	1.20 V	PS PLL
VCC_PSBATT	VCC_PSBATT	1.2-1.5 V	PS RTC and
VCC_PSAUX	VCC1V8	1.80 V	PS auxiliary
VCCAUX			PL auxiliary
VCCO_PSIO[0:3]			PS I/O
VCC_PSDDR_PLL	VCCPSDDRPLL*		PS DDR PLL
VCC_PSADC	PS_SYSMON_AVCC*		PS SYSMON ADC
VCCADC	FPGA_SYSMON_AVCC*		PL SYSMON ADC
PS_MGTRAVTT	MGT1V8	1.80 V	PS-GTR termination

* derived from VCC1V8

Table 1.1: ZCU104 voltage rails

CHAPTER 2

RELATED WORK

2.1 FPGA Adversarial Voltage Transient Attacks

Previous research has explored adversarial voltage transient attacks in multi-tenant cloud FPGAs running AI applications. Rakin et al. have developed both a novel "FPGA hardware fault injection scheme" and a search algorithm "to identify the most vulnerable DNN weight package indices" [1]. It utilizes a power plundering circuit to induce transient voltage drops, resulting in setup timing failures [2]. Temporal targeting of these attacks is achieved by leveraging a time-to-digital converter (TDC) to monitor voltage fluctuations caused by the DNN. Similar experiments have been conducted by many other researchers spanning multiple FPGA vendors, device generations [3], attack circuit designs [4], and victim applications [5] [6].

Matas et al. have demonstrated the effectiveness of a unique power plundering design using XOR trees and high fan-out nets rather than the typical Ring Oscillator (RO) used by other designs [4]. This design was verified to be Design Rule Check (DRC) clean and harder to detect than RO based designs.

2.2 Real World Attack Tactics

Apruzzese et al. conducted an analysis of adversarial Machine Learning (ML) attacks and highlighted the gap between attacks in research and practical applications [7]. The authors argue that "abundant real-world evidence suggests that actual attackers use simple tactics," rather than sophisticated gradient-based attacks. Through three real-world case studies, the paper demonstrates how practical insights can be overlooked in research. It also provides a comprehensive analysis of recent adversarial ML papers, identifying trends and blind spots. The authors propose guidelines to bridge the gap between research and

practice, emphasizing precise threat modeling, cost-driven assessments, industry-academia collaboration, and reproducible research.

CHAPTER 3

METHODOLOGY

This research attempts to perform a similar attack as was presented in the Deep-Dup paper. That is, a voltage transient induced timing error in an AI accelerator caused by a power plundering circuit in a multi-tenant FPGA environment. The hardware used for this experiment was the AMD Xilinx Zynq UltraScale+ MPSoC ZCU104 Evaluation Kit running the pre-built PYNQ image.

3.1 AI Application

The Xilinx Deep Learning Processor Unit (DPU) is a publicly available IP designed for use in Xilinx Zynq 7000 SoC and Zynq UltraScale+ MPSoC devices. It is a programmable CNN accelerator implemented in the PL. It supports a range of operations such as convolution, deconvolution, max pooling, ReLU and Leaky ReLU activations, concatenation, element-wise operations, dilation, reorganization, fully connected layers, and batch normalization. The DPU's configurable architecture allows for different core counts and AXI interfaces, making it highly adaptable to various deep learning tasks.

The hardware design used in this project is a fork of the Xilinx PYNQ-DPU GitHub repository [8] . The changes made to the design include reducing the number of DPU instances from two to one, as well as adding the Power Plundering Module described in Section 3.2. The models tested were YOLOv3, a real-time object detection model, and MNIST, a benchmark dataset of handwritten digits. They were run using the prebuilt model files and example Jupyter Notebook workflow.

3.2 Power Plundering Module

The Power Plundering Module, shown in Figure 3.1, includes a memory-mapped control interface and uses the same 75 MHz clock as the AXI4-Lite subordinate control port of the

DPU. It contains 25 clusters with 1024 power plundering cells per cluster, a total of 25,600 cells, as well as an activation pattern generator. This number of PPCs uses 5% of the ZCU104's total available logic cells. The activation pattern and number of PPC clusters to activate can be adjusted via the control registers.

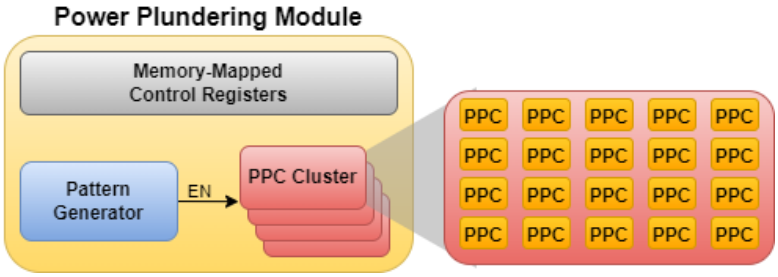


Fig. 3.1: Power plundering module

3.2.1 Malicious Circuit Design

The purpose of the malicious circuit is to consume as much power as possible. When the circuit is enabled or disabled a significant increase or decrease in power demand occurs, respectively. This sudden change in power demand causes voltage transients to occur. The details of the mechanisms behind this are discussed in Section 1.3. Most designs use some form of ring oscillator, though there is at least one other design using XOR trees [4].

The malicious, voltage transient inducing circuit design used in this experiment is based on the Power Striker Cell designed by Luo et al. [5], hereafter referred to as a Power Plundering Cell (PPC). The PPC is illustrated in Figure 3.2. The design utilizes the two LUT5 elements available in the hardware primitive LUT6_2 to implement two `NAND` gates, combining the feedback signal with an enable. Each LUT5 feeds back into itself after passing through a latch. Inserting the latch avoids combinatorial loop errors and warnings during implementation and bitstream validation. The 64-bit value used to configure the LUT6_2 is 0xFF00_0000_F0F0_0000.

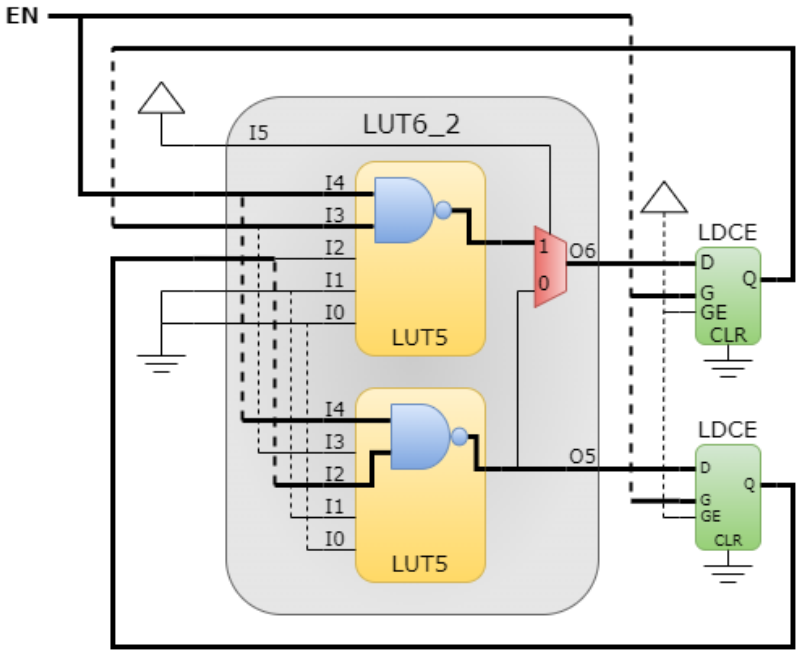


Fig. 3.2: Power plundering cell

3.2.2 Activation Pattern Generator

This attack implementation leverages a simple power plunder activation pattern generator with runtime configurable on-time and off-time intervals. This approach offers several significant strengths over model specific targeted attacks:

1. **No Prior Knowledge Required**: The primary advantage is that it does not necessitate any prior knowledge of the victim. This makes the implementation suitable for **strictly white-box** environment and highly adaptable to various scenarios without needing extensive reconnaissance or analysis.

2. **Domain Versatility**: The victim can be any hardware or software component sharing the same power rail as the PPCs. On the ZCU104, this includes software running on the PS in addition to logic implemented in the PL. Again, this makes the design flexible and capable of targeting a wide range of potential victims, including software applications, without needing specific adjustments.

3. **Simplified Design**: By not requiring detailed information about the victim, the design of the attack is greatly simplified. This can lead to faster and cheaper development, which is crucial under tight project timelines or limited budget.

4. **Real-World Representation**: This method may be more representative of real-world attacks, where attackers often do not have detailed knowledge of their targets or choose not to invest the time and money to develop a model-specific targeting method. This makes the implementation more realistic and applicable to practical scenarios.

However, it is important to note that this approach does have a downside. The results tend to be probabilistic rather than deterministic, which means there is an element of uncertainty in the outcomes. Unlike targeted methods that yield more predictable results, this approach relies on probability, which can introduce variability in the effectiveness of the attack and intended damages.

3.3 Monitoring and Measuring

Monitoring of the supply voltage rails was performed using the `get_rails()` method provided by the `pynq` python package. While this is very simple and convenient, it has a limited polling frequency. The 100 ms sample period used is far too slow to capture voltage transients. Reliably capturing these voltage transients would require sampling frequencies in excess of 100 MS/s.

An external oscilloscope could meet these requirements, but testing at the ZCU104's VCCINT probe point proved ineffective for several reasons: (1) the probe pad is positioned directly at the output of the PDS inductor were the transient voltage spikes due to the switching frequency of the voltage regulator are strongest, and (2) the PPC induced voltage transients attempting to be measured are highly decoupled due to the capacitance distributed between the supply and device. Thus, with the oscilloscope used for testing, it proved to be infeasible to measure the PPC transients over the noise of the switching regulator.

CHAPTER 4

RESULTS

This section presents experimental results and a thorough analysis of the methodology and system security.The methods used did not cause timing errors in the victim soft-IP Xilinx DPU, but a successful Denial of Service (DoS) attack is demonstrated. More specifically, the adversarial voltage transients either: (1) failed to affect any observable change in the output or (2) induced a total system crash, both software and hardware, to the point that the device was no longer accessible via JTAG and required power cycling the board. Section 4.4 conducts a thorough analysis of the nature and mechanisms behind the total system crashes.

4.1 Attack Effectiveness

The results of the attempted attack when a DoS did not occur produced bit-for-bit parity in the outputs compared to those under normal conditions. Despite the attack not causing any disruption or alteration in the system's performance, other than complete DoS, this outcome is significant as it confirms the robustness and reliability of the DPU under such conditions. The experiment demonstrated that the system's security measures are effective, ensuring that the outputs remain consistent and unaltered, thereby maintaining the integrity of the AI model. This outcome can be characterized as instructive and educational for several reasons:

1. **Validation of DPU Robustness**: The absence of any impact on the AI model's performance underscores the robustness of the system against this type of attack. This finding is significant as it demonstrates the effectiveness of the current security measures in place, ensuring the integrity and reliability of the AI model.

2. **Methodological Insights**: Although the attack did not achieve the intended disruption, it provided critical insights into the attack methodology. Understanding the

reasons behind the failure is essential for refining the techniques and developing more sophisticated approaches in future attempts. This iterative process is fundamental to advancing the field of hardware security.

3. **Point of Reference for Future Work**: This experiment establishes a reference point for future testing. The knowledge that the current setup can withstand this specific type of attack allows researchers to explore other potential vulnerabilities or to enhance their strategies to achieve different outcomes. It serves as a benchmark for evaluating the effectiveness of various security measures.

4.2 Power Plundering Strength and Patterns

The runtime configurable activation pattern and power plundering strength make testing various combinations much easier. Using a binary search pattern, plundering strengths of greatest interest were identified. It was observed that enabling 12k PPCs maintained a stable system while 13k inevitably lead to system crashes. The patterns and plunder strengths used are shown in Table 4.1.

Power Plundering Strength and Patterns					
Strength	**On-Cycles : Off-Cycles**				
	1:100	3:100	5:10	3:5	100:900
12k	clean	clean	clean	clean	clean
13k	crash	-	-	-	crash

Table 4.1: Power plundering strength and patterns

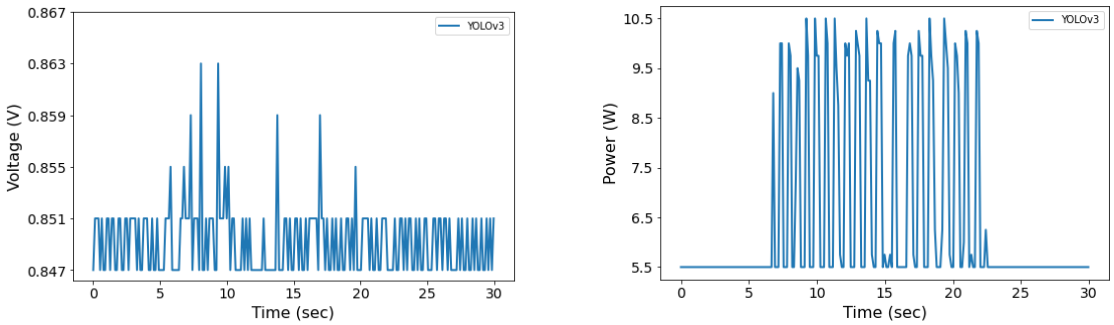
Not all patterns were tested at the 13k strength due to the tedious process of manually restarting the board after a system crash. The tests that were performed show that neither short on-time, nor long on-time paired with long off-time kept the board from crashing. It is also important to note that the board does not immediately crash but continues running for several seconds. However, the outputs from AI application during this brief period were all verified to be bit-for-bit clean.

Early experiments with permanently enabling the power plundering circuitry showed that strengths of up to 20k could be enabled without crashing the board. Strengths of 25k and above invariably lead to crashes after a single activation.

These results show that a single, high-magnitude voltage transient can be induced with a low probability of causing a system crash. It seems that, starting at 13k, increasing power plundering strength increases the probability that each individual transient will trigger a system crash. Once a critical threshold is reached, 25k in this case, this probability approaches 100%.

4.3 Voltage and Power

Voltage and power monitoring was performed during execution of the YOLOv3 model on the DPU as well as with various activation patterns of the power plundering module at 12k strength. The data presented shows these two separately to highlight the individual power profiles of each process. The voltage and power for YOLOv3, shown in Figure 4.1, are not particularly useful at a sampling frequency of only 10 S/s. Using a sampling frequency above 100 KS/s would provide enough detail to start to map the power profiles of each layer of the model and potentially provide a reference for targeting specific periods of execution.



(a) Voltage (b) Power

Fig. 4.1: VCCINT voltage rail while running YOLOv3

The voltage and power measurements for the power plundering module, shown in Figures 4.2 and 4.3, have some useful information, even at this relatively low sampling fre-

quency. The 100:100 pattern is the most interesting with its consistent high voltage spikes and low voltage droops. Though these may not be the true maximum and minimum values, they provide insight into the waveform of the voltage transients. It is hypothesized that the difference in min and max values for the patterns are likely due to the limited sample frequency. The spikes and droops from some of the patterns may just be missed due to the probability of timing alignment.

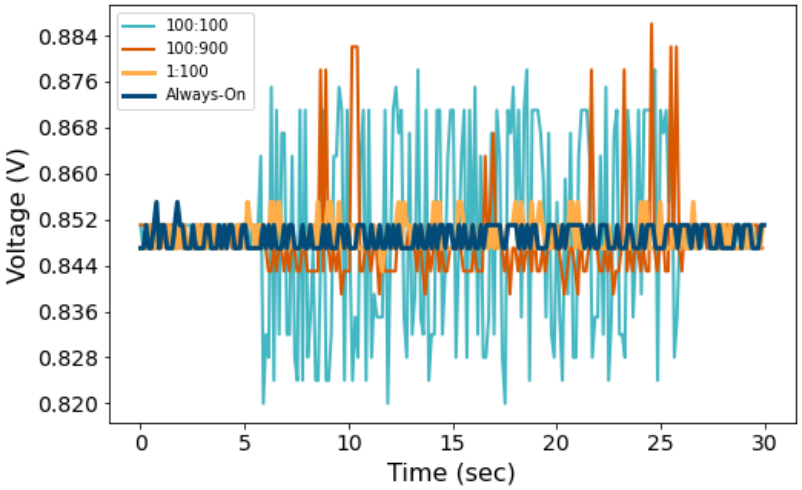


Fig. 4.2: VCCINT voltage running 12k power plundering

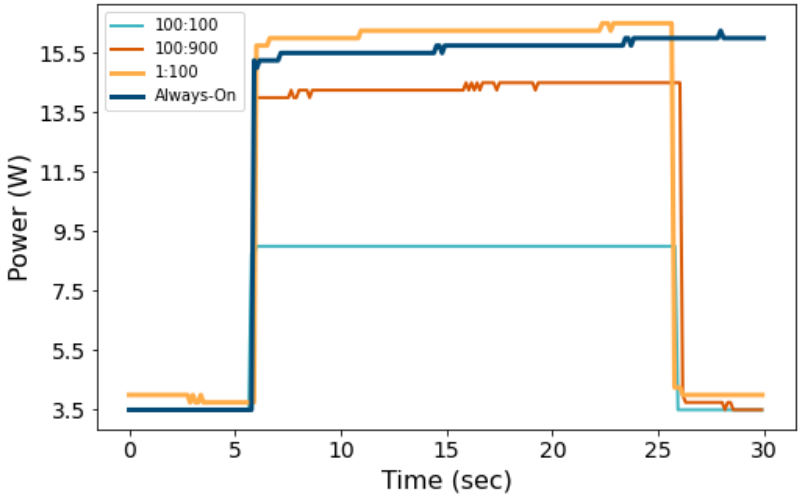


Fig. 4.3: VCCINT power running 12k power plundering

4.4 Analysis of Voltage Transient Induced System Crashes

Symptoms of a total system crash were identified as:

- PS system unresponsive via ethernet
- System is inaccessible via JTAG
- VCCINT and VCCINT_VCU power good LEDs off
- All other power LEDs remain on

Further investigation into the system crashes revealed that the VCCINT and VCCINT_VCU rails had been shut down. The corresponding power good LEDs were not illuminated, and their failed state was verified with a multimeter. All other power and status LEDs remained in a nominal state. A summary of the chip and board voltage rails is found in Table 1.1.

The VCCINT PDS consists of an IRPS5401 PMIC controlling an external MOSFET power stage TDA21240 buck-converter. The PMIC features both over and under voltage protection (OVP and UVP) which can be adjusted via PMBus. Responses to these events are configurable, but both support triggering a latched shutdown mode where the low-side MOSFET is held in the on-state. Though not conclusively proven, it is likely that the OVP or UVP of the PMIC is the cause of the system crashes. Future experiments could definitively verify this by accessing the PMIC via an external PMBus controller.

The VCCINT_VCU domain, which is not connected to VCCINT, was also affected. However, the VCU was unused in this design and thus should not be experiencing voltage transients. After examining the ZCU104 schematic, the cause was discovered to be that the enable signal of the voltage regulator for VCCINT_VCU is driven by the VCCINT_PGOOD signal. This is likely due to startup sequencing requirements.

Also worth noting, the PMIC directly outputs VCC1V8 on channel B and VCC1V2 on channels C and D, which were both unaffected by the system crash. However, the datasheet explicitly states that OVP/UVP events do not affect the other output channels unless global_fault_en is set.

4.4.1 Zynq UltraScale+ Tamper Monitoring

While exploring explanations for the total system crashes encountered, the tamper monitoring feature intrinsic to all Zynq UltraScale+ devices was discovered. This feature is managed by the CSU and PMU. Both units feature a fault-tolerant triple-redundant processor which should be capable of handling single-cycle voltage transient error correction. The PMU runs a static boot ROM at startup but can also be programmed to execute user code. The CSU only executes from ROM and is responsible for encryption key management and other key system security tasks.

After the boot process is complete, the primary purpose of the CSU is to monitor the system for tamper events. Voltage measurements from the PS SYSMON are monitored for over and under voltage events. The thresholds and response to these events can be configured via memory-mapped registers. The available responses include a *secure lockdown* mode which tri-states all MIOs, triggers the PMU to lockdown, and puts all blocks in reset, as well as a few other things.

Although it was determined that a tamper event was not the cause of the system crashes, this feature could easily be tuned to act as a countermeasure to adversarial voltage transient attacks. However, these systems are not present in Xilinx Virtex UltraScale+ devices, which power AWS EC2 F1 instances. Adding a system similar to the CSU in future products could provide methods for securing cloud-based multi-tenant FPGAs.

CHAPTER 5

CONCLUSION

The failed adversarial voltage transient attack of this project led to a wealth of information about the security of the hardware platform and Xilinx DPU. The existence of the fault-tolerant SCU and PMU, and accompanying tamper monitoring features, of the Zynq UltraScale+ devices opens new avenues of research for adversarial voltage transient counter measures. However, it is quite clear that multi-tenant FPGAs have many side-channels that need to be addressed before being considered a secure multi-tenant solution. Simply put, hardware co-tenancy is much more dangerous than the software co-tenancy that the modern cloud is built on.

Allowing only trusted, first-party designs to be programmed onto cloud-based multi-tenant FPGAs may be a viable alternative. In addition to the currently available dedicated FPGA instances, a vast library of multi-tenant safe hardware accelerator designs could be provided. If none of these designs fit the customer's needs, they would be required to use a custom design on a dedicated instance. This new product would allow a homogeneous collection of servers to provide a heterogeneous offering of accelerated computing.

REFERENCES

[1] A. S. Rakin, Y. Luo, X. Xu, and D. Fan, "{Deep-Dup}: An adversarial weight duplication attack framework to crush deep neural network in {Multi-Tenant}{FPGA}," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 1919–1936.

[2] M. Zhao and G. E. Suh, "Fpga-based remote power side-channel attacks," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 229–244.

[3] D. R. Gnad, F. Oboril, and M. B. Tahoori, "Voltage drop-based fault attacks on fpgas using valid bitstreams," in *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2017, pp. 1–7.

[4] K. Matas, T. M. La, K. D. Pham, and D. Koch, "Power-hammering through glitch amplification–attacks and mitigation," in *2020 IEEE 28th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. IEEE, 2020, pp. 65–69.

[5] Y. Luo, C. Gongye, Y. Fei, and X. Xu, "Deepstrike: Remotely-guided fault injection attacks on dnn accelerator in cloud-fpga," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 295–300.

[6] J. Krautter, D. R. Gnad, and M. B. Tahoori, "Fpgahammer: Remote voltage fault attacks on shared fpgas, suitable for dfa on aes," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 44–68, 2018.

[7] G. Apruzzese, H. S. Anderson, S. Dambra, D. Freeman, F. Pierazzi, and K. Roundy, ""real attackers don't compute gradients": bridging the gap between adversarial ml research and practice," in *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. IEEE, 2023, pp. 339–364.

[8] Xilinx, "dpu-pynq," https://github.com/Xilinx/DPU-PYNQ, 2022.

CURRICULUM VITAE

Andrew J. Gerber

Published Journal Articles

- Understanding Timing Error Characteristics from Overclocked Systolic Multiply-Accumulate Arrays in FPGAs, Andrew Chamberlin, Andrew Gerber, Mason Palmer, Tim Goodale, Noel Daniel Gundi , Koushik Chakraborty, and Sanghamitra Roy, *Journal of Low Power Electronics and Applications*, 2024