ENABLING A CONTROL SYSTEM APPROACH TO SIDE-CHANNEL AND FAULT ATTACKS

by

Matthew Carter

A Thesis
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Master of Science
Computer Engineering

Committee:

______________________________ Dr Jens-Peter Kaps, Thesis Director

______________________________ Dr. Kris Gaj, Committee Member

______________________________ Dr. Craig Lorie, Committee Member

______________________________ Dr. Monson Hayes, Department Chair

______________________________ Dr. Kenneth S. Ball, Dean, Volgenau School of Engineering

Date:__________________________ Fall Semester 2018
George Mason University
Fairfax, VA

Enabling a Control System Approach to Side-Channel and Fault Attacks

A Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at George Mason University

by

Matthew Carter
Bachelor of Science
University of Virginia School of Engineering, 2010

Director: Jens-Peter Kaps, Associate Professor
Department of Electrical and Computer Engineering

Spring Semester 2018
George Mason University
Fairfax, VA

DEDICATION

This is dedicated to my wife Heba for her abundant patience and support.

ACKNOWLEDGEMENTS

I would like to thank my family, friends, and colleagues, without whom, this work would have never been possible.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS

Analog to Digital Converter	ADC
System on a Chip	SOC
Complementary Metal-Oxide Semiconductor	CMOS
Simple Power Analysis	SPA
Differential Power Analysis	DPA
True Random Number Generator	TRNG
Flexible Open-Source Workbench for Side-Channel Analysis	FOBOS
Test Vector Leakage Assessment	TVLA
Integrated Circuits	IC
Device Under Test	DUT
Processing System	PS
Programmable Logic	PL

ABSTRACT

ENABLING A CONTROL SYSTEM APPROACH TO SIDE-CHANNEL AND FAULT ATTACKS

Matthew Carter, M.S.

George Mason University, 2018

Thesis Director: Dr. Jens-Peter Kaps

As the number of embedded devices continues to grow, attacks that require physical access to the device become more plausible. Two sub-classifications of these attacks, Side-Channel Attacks (SCA) and Fault attacks, necessitate the attacker to be familiar with the target implementation. Side-Channel Attacks exploit information leaked by the target device to discover secret cryptographic keys. Fault attacks act upon the system to induce error in device operation that may result in information leakage or improper execution. The error produced by the attack is dependent on the method used to inject the fault. This paper discusses some of the advances in SCAs and Fault Attacks and proposes a control system approach to these classes of attacks. The result of the research is a System on a Chip (SOC) for measuring power consumption, analyzing results, and refining measurement as a feedback loop.

INTRODUCTION

Traditional cryptanalysis attacks the algorithmic strength of a cryptographic system. A cipher that is resistant to attacks that attempt to derive the secret key from ciphertext and plaintext may still be vulnerable to Side-Channel and Fault Attacks. These types of attacks exploit aspects of the cryptographic system.

Side-Channel Attacks

A Side-Channel Attack involves passively capturing information leakage during the cryptographic transformation to defeat its security. In either case, the attacker requires physical access to the device to perform the attack. Paul Kocher demonstrated in [16] that devices expose secret key data in the form of timing variation when transforming data through operations such as RSA signature generation. In the case of RSA implemented with binary exponentiation, the time between the start of signature generation to completion depends on the private key data. The following is a functional description of binary exponentiation.

```
Where $d$ is the exponent and $M$ is the variable raised
to the exponent, i.e. $f(M) = M^{d}$


$S = 1$

$P = M$

for each bit $i$ in $d$:

    if ($i$ == 1)

        $S = S \cdot P \bmod N$

    $P = P \cdot P \mathit{mod} N$
```

It is clear to see, based on the functional description that, for each iteration a multiplication will only occur if the current bit in the exponent is a 1. This will impact the timing of exponentiation based on the value of the exponent. In the case of RSA signature generation, the exponent is the secret key value.

The attacker measures the time during the cryptographic operation and discovers the RSA private key. This technique was appropriately referred to as a timing attack. Timing cryptanalysis was the first of a new assortment of attacks that focused not on the algorithmic strength of a cipher, but on the aspects of that cipher's operation.
Based upon similar principles as the timing attack, leakage of secret key data may occur in other forms such as power consumption, electromagnetic radiation,
and cache contents.

Integrated circuits, such as microprocessors, microcontrollers, and SRAM, consist of complementary metal-oxide semiconductor (CMOS) gates. The dynamic power consumption of CMOS gates contributes to the externally observable power consumption. Figure 1 is a representation of a CMOS inverter gate with the load capacitance C_L and voltage of the power supply V_{DD}. On the rising-edge of the input to the gate (0 → 1), the power supply charges the capacitance. Equation 1 defines the relationship between change of bit state and dynamic power consumption [33].

$$P_{dyn} = C_L V_{DD}^2 P_{0\to 1} f$$

Equation 1: Dynamic Power Consumption of CMOS Gates

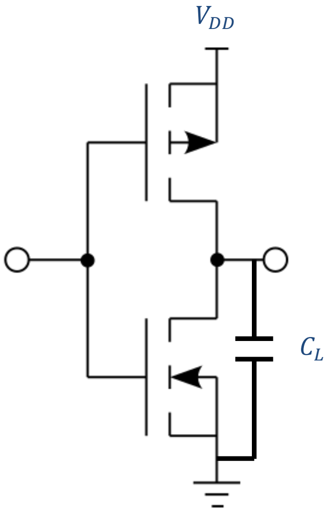


Figure 1: CMOS Inverter Gate

The relationship between the inputs to CMOS gates and power consumption leaks information about the values of secret data processed by CMOS devices. After measuring this leakage during selected operations of a device, an attacker analyzes the measurements to discover the secret data.

Fault Attacks

Fault attacks alter operation of a cryptographic device by physically acting upon it to insert a fault. The resulting fault may cause a variety of results depending on the implementation, including incorrect ciphertext generation, output of the plaintext, and others.

As described in [28], a clock glitch fault attack is when an attacker manipulates the clock to introduce an error in the device's execution. The technique involves inserting a rising edge in the clock prior to the completion of the device's clock cycle, shown in Figure 2. The most common desired result of a clock glitch attack against a microprocessor is for the device to skip the instruction fetched prior to the glitch.

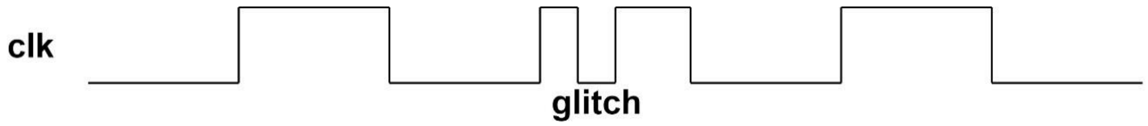


Figure 2: Clock signal with the addition of a glitch

Feedback Control Systems

Control systems manage the behavior of a device or system based on predefined control logic. The two subclasses are open-loop and closed-loop control systems. Open-loop control systems do not process feedback from the managed device or system. On the other hand, closed-loop control systems process feedback from the managed device or system to compare the output(s) to the desired output(s). Figure 3 is the model for closed-loop control systems. The use of feedback loops to automate control systems was first applied to an industrial process in 1769 to control the speed of a steam engine. Additional background and applications of control systems is available in [6]. The model for closed-loop control systems remains consistent between applications, however, the implementation of the control logic will vary based on the application. Some of the potential controller implementations include PID, Fuzzy Logic, and Machine Learning. The most successful implementation type is dependent on the system that receives the controller’s input and generates the output.

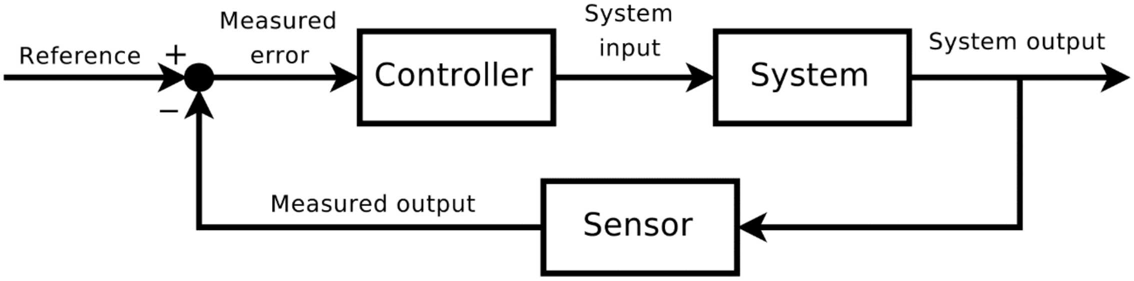


Figure 3: Closed-Loop Feedback Control System

PREVIOUS WORK

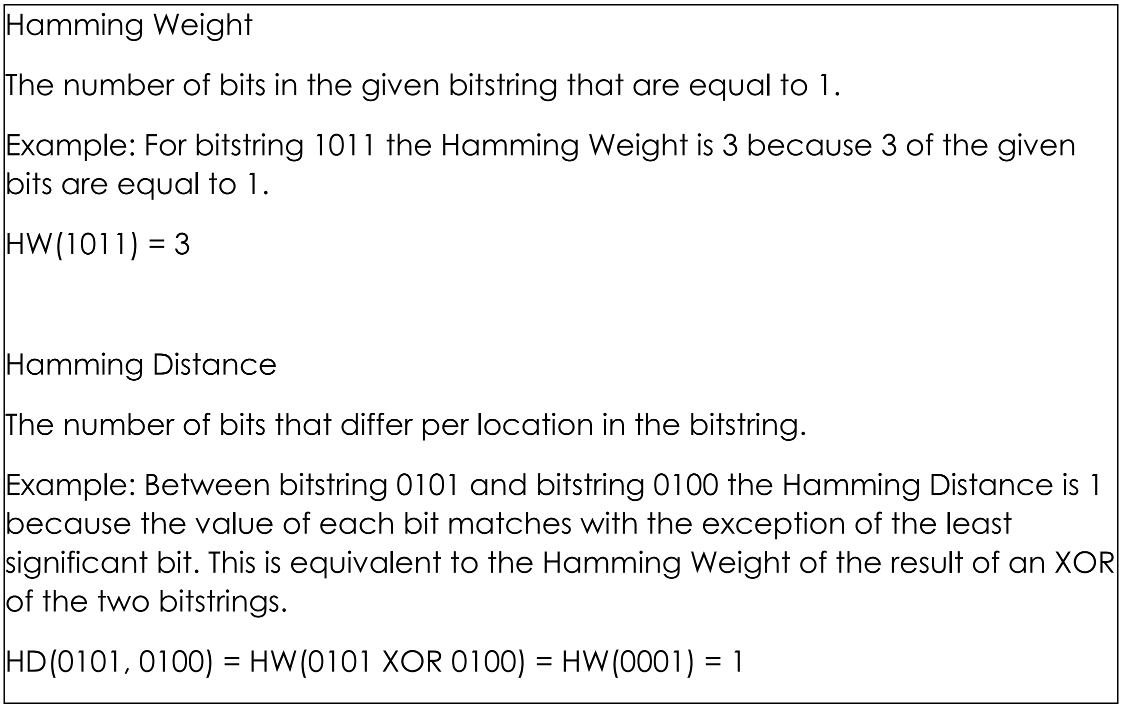
Side-Channel Attacks

Power Analysis

The first attacks using device power consumption to reveal secret cryptographic keys was reported in [14]. The first type of attack described in the report, simple power analysis (SPA), exploits the relationship between power consumption and the operation performed. A cryptographic algorithm leaks information about the secret key data if the operations vary based on its value.

The operands provided as inputs to a given operation also influence power consumption as shown by the second type of attack, differential power analysis (DPA). DPA combines knowledge of a cryptographic algorithm's implementation and statistical analysis to reveal secret key data. This attack uses knowledge of the target's implementation to define a selection function D for which all inputs are known except for the secret key value. The output of the selection function is the value of the target bit. Based on a key guess, the attacker aligns the traces for each encryption and computes a differential trace value Δd. As the number of traces increase, the Δd value will approach either 0 if the guess is incorrect or the influence of that bit on power consumption if the guess is correct. Additional information on DPA and its ability to break cryptographic systems is available in [15] and [17].

Correlation Power Analysis (CPA), first described in [2], also exploits the relationship between the value of operands and power consumption to reveal secret keys in cryptosystems. CPA requires defining a power model for the expected power consumption during an operation in the cipher implementation. This typically incorporates the concepts of Hamming Weight or Hamming Distance. The definitions of the two metrics are as follows:



Hamming Weight

The number of bits in the given bitstring that are equal to 1.

Example: For bitstring 1011 the Hamming Weight is 3 because 3 of the given bits are equal to 1.

HW(1011) = 3

Hamming Distance

The number of bits that differ per location in the bitstring.

Example: Between bitstring 0101 and bitstring 0100 the Hamming Distance is 1 because the value of each bit matches with the exception of the least significant bit. This is equivalent to the Hamming Weight of the result of an XOR of the two bitstrings.

HD(0101, 0100) = HW(0101 XOR 0100) = HW(0001) = 1

The Hamming Distance leakage model as it applies to correlation power analysis (CPA) is further illustrated by [3].

Power Analysis is not limited to a specific type of device. If an attacker designs an appropriate power model, this technique applies to microcontrollers, FPGAs, ASICS, and others. Further evidence of its suitability with a wide range of devices is shown in [9], [31], [37], [19], [21], and [13].

Combining leakage models to achieve results better than each independently is another SCA technique as shown in [29]. This technique of combining leakage models, referred to as second order DPA, is explained in detail in [12].

Alignment of Measurements

Alignment of traces impact the results of Side-Channel Analysis. If the measurements interpreted as leakage are not aligned to the device's operation, successful extraction of key bits is impossible. Techniques exist to improve the alignment. Knowledge of the path delays in a circuit enable an attacker to improve DPA results by realigning power consumption traces as shown in [23].

Some countermeasures designed to defeat Side-Channel Attacks attempt to make it difficult to identify measurements that correlate to cipher operation. [38] coins the term “elastic alignment” as a method to defeat the aforementioned countermeasures. Alignment of measurements becomes much simpler when analysis of captured traces contributes to the configuration of the capture mechanism. This relationship is easily realized through a feedback loop based on control theory.

Fault Attacks

Several different techniques exist for breaking device security by inducing a fault. Classification of types of fault attacks and descriptions for each type of attack is in [35]. This is expanded on by [28], but also discusses various methods available to perform fault injection including manipulating the clock and power supply. A targeted fault attack in [1] is against an RSA implementation that incorporates the Chinese Remainder Theorem.

There are several methods to introduce error in the operation of a cryptographic transformation. The research of [18] demonstrates power and clock glitch fault injection on a true random number generator (TRNG). Furthermore, the report includes the designs for thermo and underpower attacks.

Clock Glitch

Instruction skip attacks modify the execution of processors by altering the clock. The reported results in [20] demonstrate the effectiveness of this type of fault attack by implementing a system for performing clock glitches and using it to attack a device executing AES.

One consideration when inserting a clock glitch is at what point in time to introduce the rising edge. This is especially true when targeting a pipelined architecture as shown in [39]. The report identifies the clock cycle to insert a clock glitch based on the stages of the RISC pipeline.

Voltage Fault

Voltage faults may be subcategorized into two different methods of injecting the fault. Reducing the voltage to induce propagation delays results in timing errors in a device. At the time of [27], the RSA implementation in OpenSSL incorporated the fixed window exponentiation algorithm. The report illustrates an attack on the algorithm using power regulation. Alternatively, significantly increasing the voltage for a short duration may cause bit errors in device operation, known as a voltage glitch. Experimental injection of faults through voltage glitches and the resulting errors may be seen in [25].

SCA and Fault Attack Tools

Following the work [14] in the area of Side-Channel Attacks, researchers demonstrated similar attacks against vulnerable devices. Commonly, Side-Channel and Fault Attacks are executed through hardware and software components built for the individual research effort. Implementation time for these components required to perform SCA is costly despite the availability of published attacks. For this reason, reproduction and extension of attacks is more challenging when researchers built a custom attack system. To avoid such a delay, open-source and commercial off-the-shelf (COTS) tools may be incorporated to provide a foundation for additional contributions. Tools for SCA may be subdivided into categories such as test device development board, software for analysis of leakage, fully capable attack platforms, and others. All of these, however, reduce the duplication of effort in the research area.

Implementation Evaluation

Evaluation of prototype implementations of fabric-based products for susceptibility to SCA requires a cryptographic development board. The Side Channel Attack Evaluation Board (SASEBO) is intended to fulfill this need. The most recent versions of the evaluation board are the SAKURA boards. The development board includes HDL for cryptographic algorithms commonly used in security applications. The development board provided a test device for correlation-based electromagnetic analysis (CEMA) in [10].

Open-source Software

Side Channel Attacks require hardware to interact and measure device leakage, however, software packages for processing measurements are independent of the collection system. SideChannelMarvels is a collection of SCA-related open-source software projects available on GitHub. One such project, [4], contains software for performing correlation power analysis on previously collected power traces. A similar open-source project, OpenSCA Toolbox [26], consists of matlab source code developed for DPA. Open-source software solutions such as these are ideal for a researcher with a system for collecting power traces, but without a means of analyzing the measurements.

Systems for Capture and Analysis

The Flexible, Open-source Board for Side-channel Analysis (FOBOS) is an open-source platform for performing Side-Channel Attacks. The FOBOS system executes a CPA attack against an FPGA running AES as shown in [34] through the cooperation of components such as a control FPGA, an oscilloscope, and analysis software. All parts of the system are COTS products.

Recent development on the FOBOS system added the feature: Test Vector Leakage Assessment (TVLA). TVLA calculates t-values using Welche's t-test on collected measurements. The t-values indicate if two sets of collected samples for different plaintext inputs may be differentiated. The result of which indicates if leakage of information is occurring through the measured medium.

Chip Whisperer is open-source and includes support for CPA like FOBOS, but is also capable of performing other Side-Channel Attacks and fault injection. An introduction to the tool as well as a list of attack tutorials Chip Whisperer is capable of performing is available at [8]. Several of the components in the system are custom made, but detailed descriptions and templates are available. Several examples of practical attacks built upon the foundation of the Chip Whisperer platform is available in [25].

The DPA Workstation is a proprietary system for Side-Channel Analysis. It combines an Oscilloscope for capturing traces and a PCI interface for measurement offloading. The high-speed PCI interface improves the capture process by removing the bottleneck caused by traditional offloading interfaces such as USB. The entire list of features and deliverables is available at [7].

A proprietary system with a similar architecture to the DPA Workstation is the Riscure Inspector. The system includes hardware components for the physical interaction with the target device for each type of attack. Software support intended for execution on a PC provides the interface for the user with the hardware components. An overview of the system may be found in [46].

Traditionally, an SCA attack system collects measurements with an oscilloscope, however, [25] demonstrated that by synchronizing measurements to the target device's clock, correlation between measurements and leakage is possible with one trace per target device cycle. To facilitate measurement collection on an FPGA, the author developed the open-source tool OpenADC. Removal of DPA and other SCA methods' fundamental dependence on a high-speed collection system such as an oscilloscope, reduces the hardware overhead cost substantially.

As shown in [24], the consistency of power measurements significantly improves as the capture interval is synchronized with the target device clock. The OpenADC captures power traces at a phase offset between device clock cycle. This configurable offset is a significant factor contributing to the effectiveness of SCA when traces are captured with OpenADC.

Feedback Control Systems

PID Algorithm

The PID acronym stands for proportional, integral, and derivative. Proportional is the feedback's error relative desired output, integral is the error accumulated via measurements during control iterations, and derivative is the error relative to the error measured during the last control iteration. Each of the calculated variables contribute to the control response. [22] describes the design of PID control system for non-linear feedback.

Fuzzy Logic

The concept of fuzzy logic was first presented in [40]. It defines fuzzy sets that are characterized by membership functions. The result of the membership function is a grade between 0 and 1, defining the membership degree in a particular subset for a given input. This process is referred to as “fuzzification”. The fact that inclusion in a set is scaled rather than absolute discriminates fuzzy set theory from traditional set theory. The inclusion within a subset of a fuzzy set is determined as follows:

A is a fuzzy set such that $A = \{ X, Y, Z \}$

M_X is the membership degree of i in subset X

$M_X = f_X(i)$ and $0 <= M_X <= 1$

Classical set theory defines set operators such as AND, OR, and NOT. These same operators are redefined in fuzzy logic to operate on the membership degrees produced by the membership functions. The fuzzy logic controller has predefined rules which form a decision matrix that maps the membership degrees of the input to membership degrees of the output fuzzy set. The final step, referred to as “defuzzification”, translates the membership degrees of the output fuzzy set to the control action. Further introduction to these concepts and more concerning Fuzzy Logic is at [5].

Fuzzy logic clearly has an application for control systems, however, it has also been evaluated in the Side-Channel Analysis domain. [32] combines electromagnetic and power measurements with Fuzzy Logic to determine the secret key bits of a cryptosystem. The findings of this report suggest that fuzzy logic produces better results than traditional Side-Channel Analysis methods when processing noisy measurements.

Machine Learning

Machine learning is a broad field with applications to many different domains. Iterative learning control (ILC) algorithm, classified as a machine learning algorithm, is commonly implemented in control systems for repetitive processes such as robotics. [36] presents a model for ILC based control of the output of probability density functions (PDF) in non-Gaussian stochastic systems.

Machine Learning has already been applied to break security of hardware devices. Pattern analysis algorithms in machine learning are referred to as kernel methods. A subclass of kernel methods is Support Vector Machines (SVM), tailored to solve pattern recognition problems. An initial study on the suitability of the Least Squares Support Vector Machine (LS-SVM) learning algorithm for deriving secret key data from power measurements is shown in [11]. In the report, the results of the LS-SVM approach are comparable to the results of a template attack.

MOTIVATION

To address the security implications of side-channel and fault attacks, researchers developed countermeasures in hardware and software. This, in turn, became an iterative process with new attacks emerging followed by countermeasures to thwart them. Despite the availability of countermeasures, developers may choose not to implement them for any number of reasons including cost, time, or ignorance of their existence. New manufacturing techniques and architectures safeguard devices from traditional side-channel and fault attacks regardless of a lack of countermeasures.

The evolution of integrated circuits (IC) in devices presents new challenges for side-channel and fault attacks. CMOS technology, used in the construction of ICs, has reduced in physical size as a direct result of advances in manufacturing. Recent ICs consist of several processing units that may even vary in frequency and supply voltage. The addition of processing units inhibits the ability of the attacker to isolate the effect of secret data on the power consumption. Noise added to measurements by parallel execution reduces effectiveness of side-channel attacks. Identifying valid points of execution to inject faults becomes far more challenging. To address these and other barriers raised by technology trends in ICs, an attack platform capable of performing more complex side-channel and fault attacks based on principles of control theory is required.

Control theory is a branch of mathematics and engineering that studies the control of dynamical systems. Systems based on control theory administer control actions based on the state of process variables to achieve desired outputs and stability from the controlled system. The same relationship between controller and dynamical system may be applied to the attack platform and target platform. Refining the measurement configuration based upon the latest output provided by the feedback loop will produce the most targeted result possible with the measurement system. This new model for physical attacks builds not only on the advances in side-channel platforms, but also on control theory.

To implement a physical attack platform capable of using control theory principles to improve results, latency between capture and analysis must be minimized. Other platforms for side-channel analysis physically and logically disassociate components for capture and analysis. Typically, a PC performs the analysis on results collected by another component. The transfer of samples from the capture component to the PC creates a bottleneck if significant hardware resources are not incorporated to overcome it. To avoid such requirements and simplify the design, capture and analysis will be performed by the same component.

Systems for evaluation of target platforms' susceptibility to side-channel and fault attacks allow researchers to accelerate their experiments. Even with a fully capable system, the existing functionality may be insufficient to attack the target hardware. Increasing the number of probes and fault injection hardware increases demands on the attack platform. For this reason, a flexible, open-source design must provide the foundation for further improvements. The design of a low-cost, effective platform capable of concerted side-channel and fault attacks will aid future development of attacks and countermeasures.

RESEARCH APPROACH

A singular component capable of capture and analysis requires capture hardware and the ability to execute software for analysis. The Xilinx Zynq product family integrates a ARM core in FPGA fabric. Furthermore, the PYNQ project includes python libraries for functionality specific to Zynq products such as programming an overlay into the FPGA from the ARM core. The ARM core is executing a Linux Operating System and is capable of executing analysis software. More information concerning Xilinx's open-source project PYNQ is available at [30].

The first step to producing an attack platform on the PYNQ board is to port existing FOBOS functionality for control, data acquisition, and analysis to the PYNQ architecture. Integrating the existing software and HDL for FOBOS will reduce effort in supporting side-channel features implemented previously. Verification of success will be based on reproduction of earlier experiments carried out by FOBOS. Following the FOBOS port, improving the capture mechanism on the new attack platform will be next. The hardware for capture will initially rely on OpenADC, the open-source capture tool mentioned previously. The PYNQ board will directly interface with OpenADC to configure operating parameters and retrieve samples. This design will be efficient due to the parallel processing capabilities and through reduction of traces provided by OpenADC as compared to an Oscilloscope. The fewer the traces, the less cycles required to process them. The new capture method success ratio and time efficiency will be compared to results of ported FOBOS functionality with the use of an oscilloscope.

Control theory will ensure the attacks are effective despite a reduced sample size. For side-channel attacks such as power analysis, configuration of the capture hardware will be the control action. Implementing similar functionality as FOBOS's TVLA would produce t-values that may be interpreted as perceived leakage. These t-values are the process variables and the desired set point is when t-values are most prominent. Evaluation of success is based on whether this new platform is capable of aligning capture traces at the appropriate phase offset without direction. The final closed-loop feedback control system including the device under test (DUT) appears in Figure 4.

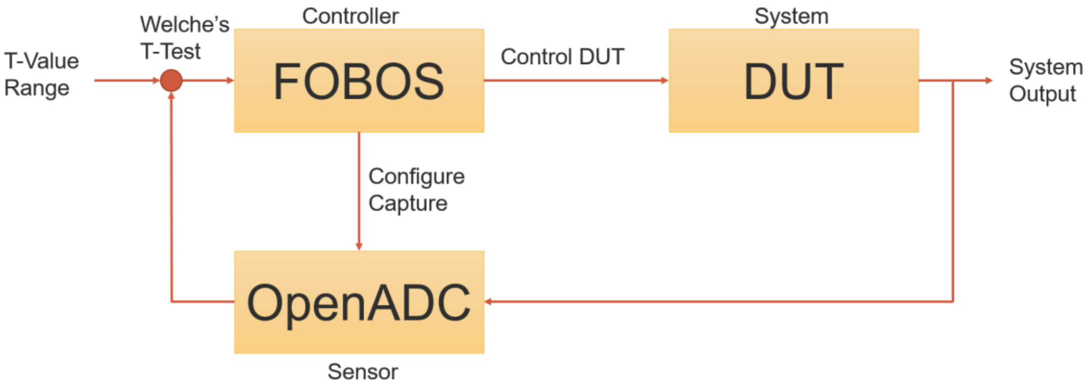


Figure 4: FOBOS Control System

A similar approach may be applied to fault attacks. Alignment of fault attacks based on control logic is just one approach. After the completion of side-channel attacks incorporating control theory, the attack capabilities will continue to expand.

PYNQ: PYTHON PRODUCTIVITY FOR ZYNQ

The Xilinx Zynq product family combines an ARM processor with FPGA fabric as a System on a Chip (SOC). Xilinx's open-source project PYNQ built a library written in the Python programming language that is compatible with Zynq boards to allow for simple integration of software with hardware. The ARM processor runs the Linux operating system. The Python software executes in this Linux environment and uses the PYNQ library to interact with kernel module responsible for communication with the hardware.

The FPGA fabric is referred to as the Programmable Logic (PL) and the ARM processor system is referred to as the Processing System (PS). [41 Figure 5] provides a high-level overview of the PS and PL components.

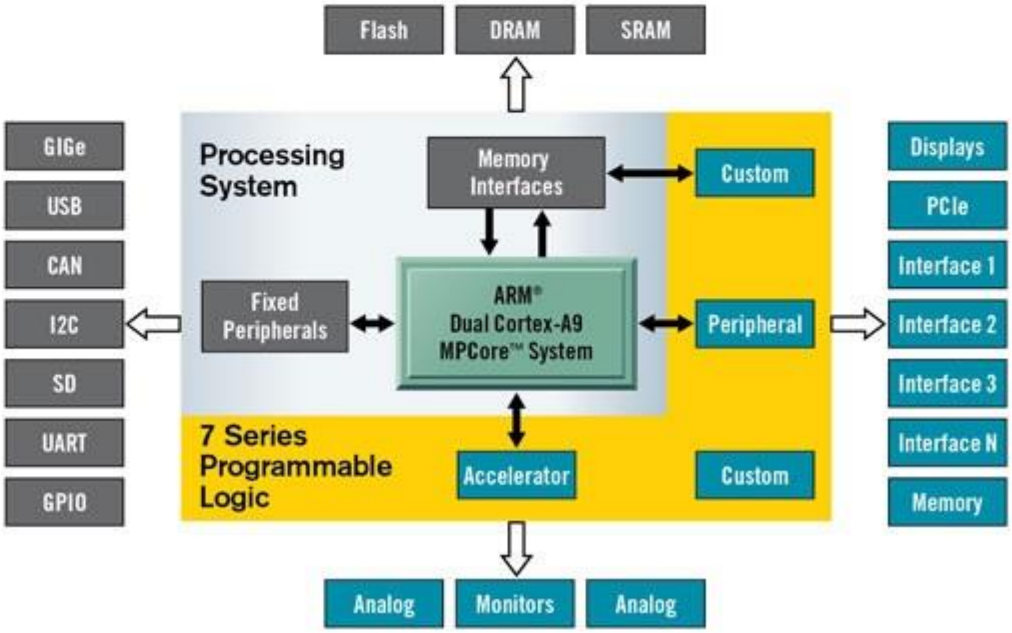


Figure 5: PYNQ-Z1 block diagram

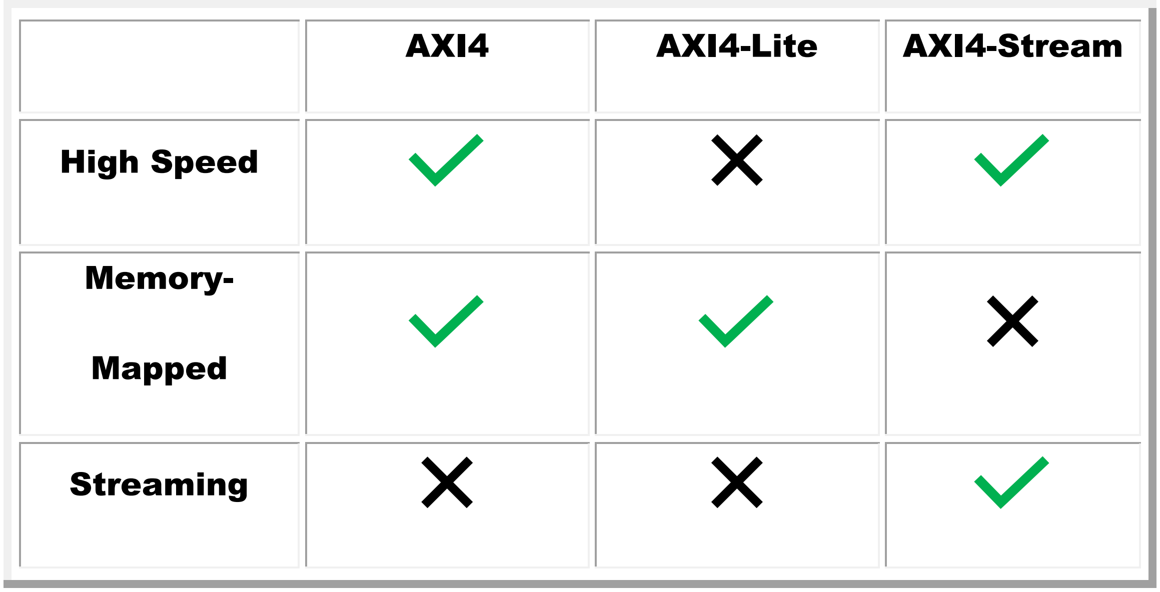
The Advanced eXtensible Interface (AXI) protocol provides the bidirectional communication channel between the PS and the PL. The PYNQ-Z1 board by Digilent was the target platform for all development and testing, however, all HDL and Python code developed as part of this research effort is likely compatible with the PYNQ-Z2.

AXI Protocol

The AXI protocols are part of a specification known as the Advanced Microcontroller Bus Architecture (AMBA). AXI4 is the version of the protocol adopted and supported by Xilinx. This includes three types of interfaces including the standard AXI4, AXI-Lite, and AXI4-Stream.

Table 1 is a high-level comparison of the three interfaces. Additional information regarding the AXI protocols is available at [43].

Table 1: Comparison of AXI4 Interfaces



	AXI4	AXI4-Lite	AXI4-Stream
High Speed	✓	✗	✓
Memory-Mapped	✓	✓	✗
Streaming	✗	✗	✓

The protocol varies between the three interfaces, but all three designate a master-slave relationship between two interfaces that transfer data in increments of the desired bus data width. AXI-Stream interfaces transfer blocks of data that are some multiple of the bus data width. The AXI4 and AXI4-Lite use memory-mapped communication to write and read memory in the PL. The combination of the AXI4 protocol, Linux kernel drivers, and PYNQ software forms the pathway for writing configuration registers and reading status registers of IP cores programmed in the PL from software executing on the PS. [42 Figure 6] shows the physical interfaces and associated protocols of the PS-PL interconnect. These interfaces include four general purpose (GP), four high performance (HP), and one Accelerator Coherency Port (ACP). The GP ports are AXI4, the HP ports are AXI4-Stream, and the ACP port acts as a HP port with cache coherency.

The PYNQ board architecture supports software applications that are accelerated via hardware support. This is ideally suited for a closed-loop feedback system comprised of a hardware sensor and software for analysis.

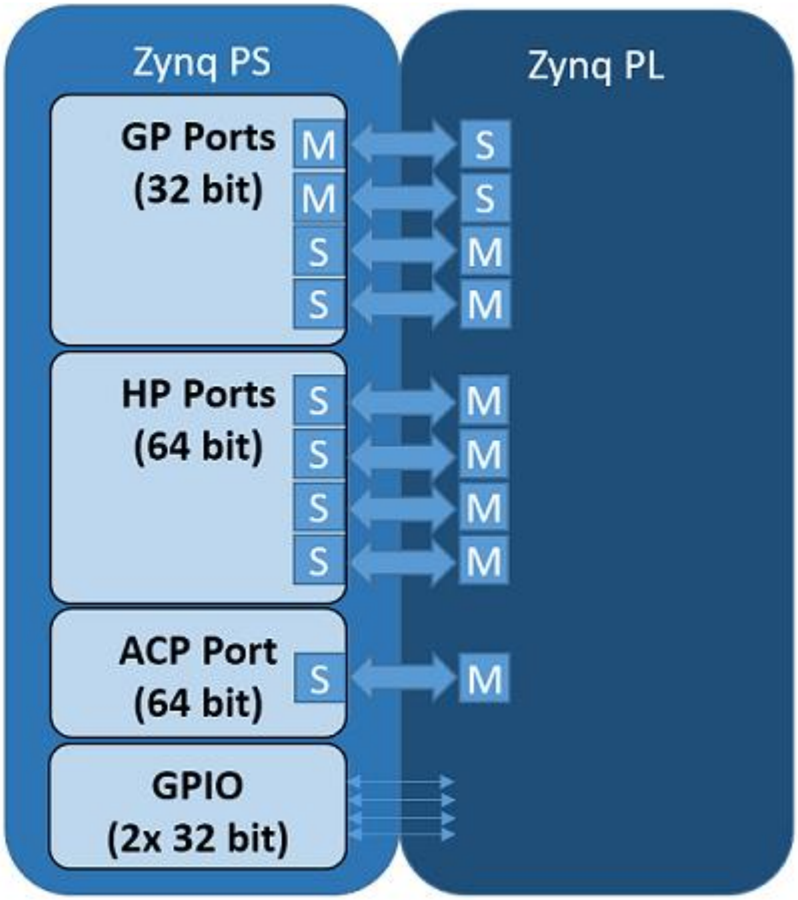


Figure 6: PS to PL Interconnect

FOBOS V3

This research effort builds upon earlier work contributed towards the FOBOS system. By incorporating Python and HDL code developed towards earlier versions of FOBOS, the development time will be significantly reduced. The most recent release of FOBOS was version 2. Any software and IP cores produced during this effort will be part of FOBOS version 3. The final block diagram for the FOBOS hardware components will be as shown in Figure 7.

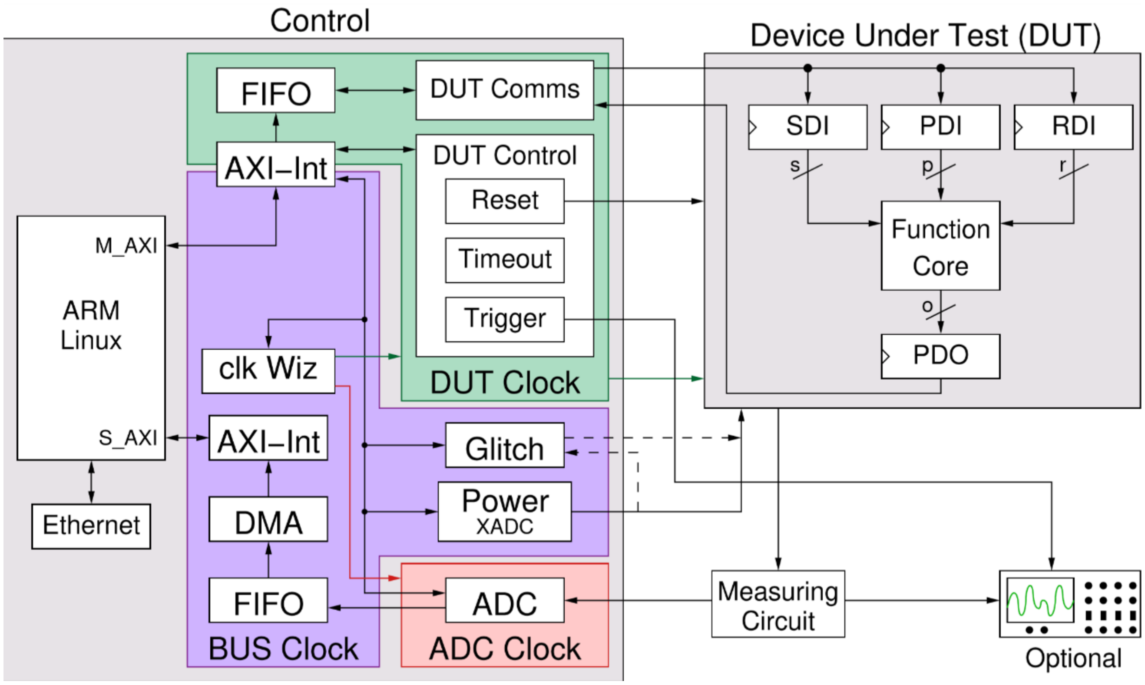


Figure 7: FOBOS v3 Block Diagram

The Device Under Test (DUT) will be the target of the side-channel or fault attack. In Figure 7, the control device represents the PYNQ board. The PYNQ board will be responsible for driving the DUT operations. Additionally, the control device will capture power measurements and perform power glitches. This paper discusses development and testing of the hardware and software components required to capture power traces.

OpenADC

As discussed previously, the OpenADC is an open-source hardware device with an analog to digital converter (ADC) capable of measuring power consumption. At a maximum frequency of 105 MHz, an oscilloscope is not required to capture power traces of target devices with a clock rate equal to or less than 105 MHz. The board outputs ADC samples at the clock rate provided as an input. The technical specification for the OpenADC is available at [44]. The OpenADC is represented by the ADC component in the block diagram of Figure 7. The OpenADC board is shown in Figure 8.

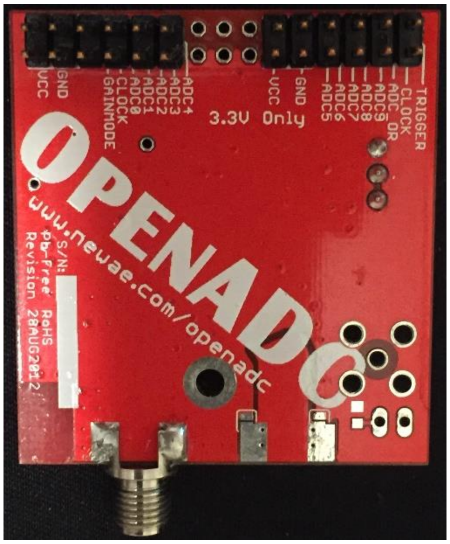


Figure 8: OpenADC Capture Components

The OpenADC has peripheral module (Pmod) headers compatible with Digilent boards such as the PYNQ-Z1. The two Pmod connectors of the OpenADC and PYNQ-Z1 are the communication mechanism for capturing measurements. Integration of FOBOS capture hardware with OpenADC provides the platform with immediate ADC values for analysis.

FOBOS ACQUISITION

Four high-level constructs form the system for acquisition of power samples in FOBOS v3: the OpenADC interface, data pipeline, clock generator, and software drivers. The OpenADC interface is an IP core programmed in the PL that controls and converts OpenADC outputs to power measurements. Additionally, this IP core passes the measurements to the data pipeline. The data pipeline stores measurements in a first-in, first-out (FIFO) buffer and provides direct memory access (DMA) to the PS.

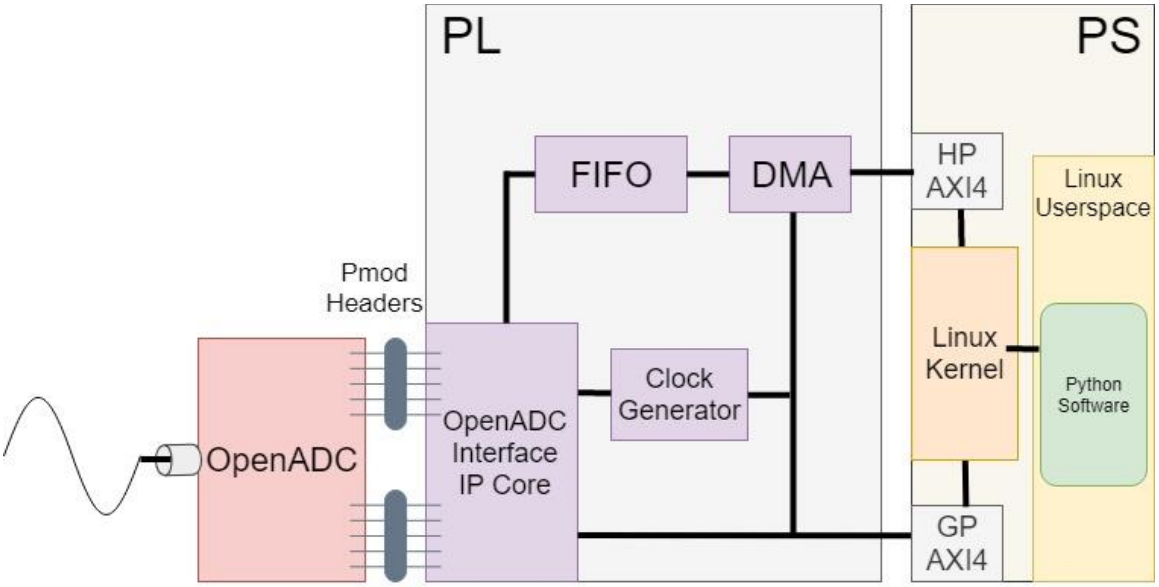


Figure 9: FOBOS v3 Acquisition Components

Finally, software drivers configure the IP cores and read from the data pipeline via DMA software. Figure 9 represents a block diagram of the FOBOS v3 acquisition components.

OpenADC Interface

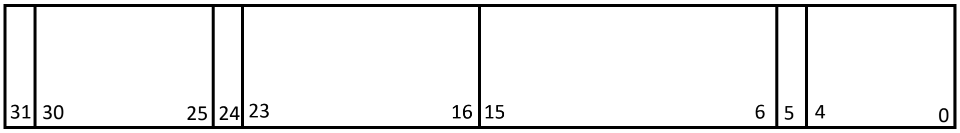
The OpenADC Interface is an IP core written in VHDL that controls and converts OpenADC outputs to measurement values in the FIFO. The module has two AXI4 interfaces. The AXI4 Slave interface is capable of writing and reading configuration registers. The AXI4-Stream Master interface transfers values to the FIFO that the OpenADC Interface received from the OpenADC.

The number of values transferred contiguously is referred to as the block size. The block size is dependent on the configuration of the IP core, however, it will always be a multiple of four. This restriction exists due to the fidelity of the ADC, which is 10 bits. The OpenADC interface concatenates the 10 bits of the measured value with additional 6 bits for convenience. Four of these values are concatenated and transferred to the FIFO in sizes of 64 bits. The OpenADC Interface buffers the values until it receives four from the OpenADC, allowing the transmit to the FIFO to complete before the next 64-bit value is ready. For convenience of the software, the block size configuration value is represented as the number of 64-bit values transferred contiguously.

There are two supported modes of capture for the OpenADC Interface: acquisition and downsample. Both capture modes are initiated by an enable flag, which is either provided in the configuration register or as an input to the IP core. Acquisition mode captures every ADC value until the number of blocks specified in the configuration register. Downsample mode captures ADC values based on the divisor value and the mean configuration bit until the number of blocks in the configuration.

The OpenADC Interface stores configuration and status in system registers. The system registers are capture mode, capture count, divisor, and status. The following figure displays these four registers and their fields.

Capture Mode



Start	# bits	Field
31	1	Start Capture
25	6	Least Significant Bits of Measurement
24	1	ADC Hi/Lo
16	8	PWM Duty Cycle for OpenADC Gain
6	10	Block Size
5	1	Capture Mode
4	1	Reset Capture
0	4	Reserved

Capture Count

Start	# bits	Field
31	1	Reserved
0	31	Number of Blocks to Capture

Divisor Register

Start	# bits	Field
31	1	Calculate Mean
0	31	Divisor Value

Status Register

Start	# bits	Field
31	1	Capture Complete
0	31	Number of Blocks Captured

Figure 10: OpenADC Interface System Registers

Data Pipeline

A FIFO and DMA controller are the hardware components that compose the data pipeline. The FIFO stores received values and the DMA controller empties values from it. The AXI4-Stream interface separates data via a signal called TLAST. The OpenADC Interface asserts the TLAST signal at the configured block size. The DMA controller has access to the PS's physical memory. When requested, the DMA controller retrieves data from the FIFO and copies that to a contiguous memory space specified by the PS software. Each read from the FIFO will be segmented by the pre-defined block size. This pipeline allows the software to asynchronously request the data captured by the OpenADC Interface into a prepared memory buffer.

Clock Generator

The Clock Generator outputs a clock signal based on the register values during initialization or later configuration. The OpenADC and the corresponding Interface IP core use this as the frequency to capture values. Software on the PS configures the clock signal outputted by the Clock Generator based on the target clock. The configurable options are the clock rate and the phase shift of the generated clock signal.

Software Support

Interaction with the PL from the PS involves multiple layers of software. Linux kernel modules facilitate communication with the PL through AXI4 hardware interfaces. Userspace libraries use system calls as the communication mechanism to the kernel modules. Lastly, userspace processes include these libraries with layered functionality that achieves a specific desired cooperation between the hardware and software.

During development of IP cores with AXI4 interfaces, the IP cores are assigned a physical address space. The kernel accesses this physical address space through the AXI4 ports in the PS. If the IP core defines registers, each register corresponds to a memory address in that block. The kernel translates reads and writes of virtual addresses that map to these physical addresses to reads and writes of their respective physical addresses via AXI4. AXI4 not only enables the PS to access PL memory, but also the PL to access PS memory. In this alternate case, the kernel provides physical memory locations to be read from or written to via AXI4 by the DMA controller.

The userspace libraries included in the PYNQ open-source project provide an abstraction for other software when accessing IP address space. This class, named memory mapped input/output (MMIO), maps the physical memory space of an IP core through the mmap system call with the /dev/mem character device. When software makes a call the MMIO read or write function, the library translates that read or write to an offset in the memory space returned by the mmap system call. The XLNK class in the PYNQ library allocates contiguous memory that is accessible to the PL. The DMA class then writes the physical address of this buffer to the DMA controller’s registers before initiating a transfer to or from that buffer.

To support seamless integration of the IP cores and python software, the author developed an openADC and clock generator python drivers. Additionally, the class fobosAcquisition uses these drivers to configure and check the status of these IP cores.

Verification of Acquisition Success

After completion of the hardware and software modules for acquisition, verification of the successful capture of all power traces began. The testing could be achieved by adding another capture tool and comparing the results, but the author preferred a minor change to the PL that produced verifiable results in the software. This required the addition of a binary counter to the PL incrementing at the frequency of the target clock. The OpenADC Interface has an optional port that takes the counter's output as an input. This counter value replaces the output of the OpenADC, allowing the python software to verify the output received from DMA. If each value received is one value greater than the previous value, then the acquisition system successfully handled all data entries.

The binary counter IP core verifies the clock generator, data pipeline, and Python software, however, it does not verify accuracy of OpenADC samples with the acquisition system. To test acquisition with the OpenADC, the Analog Discovery tool by Digilent produced a repeatable signal at the same clock frequency as the clock generator. This tool is capable of measuring, recording, and generating digital and analog signals. More information about the Analog Discovery is available at [45]. The test setup for waveform generation is shown in Figure 11.

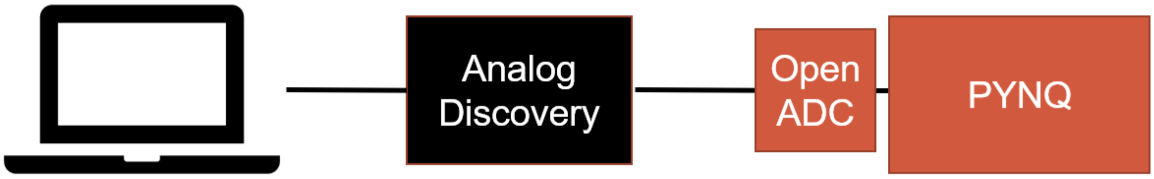


Figure 11: Test Setup with Waveform Generation

The OpenADC has a female SMA connector for input to the ADC. The Analog Discovery has an adapter board for BNC cables. A BNC to SMA cable provided the input to the OpenADC from the Analog Discovery.

Figure 12 is the matplotlib graphs of a 100 KHz sine wave with an amplitude of 100 mV generated by the Analog Discovery and captured by FOBOS v3 Acquisition System. The OpenADC gain mode was set to high with a gain value of zero. At lower gain values, some anomalous ADC values distort the graph. The top graph includes these traces. A median filter applied to the captured ADC values removed these values. The plot after application of the median filter is the bottom of the two. Precision of the ADC values is unimportant for verification of the gain and clock control, therefore, all additional traces are passed through a median filter prior to plotting.

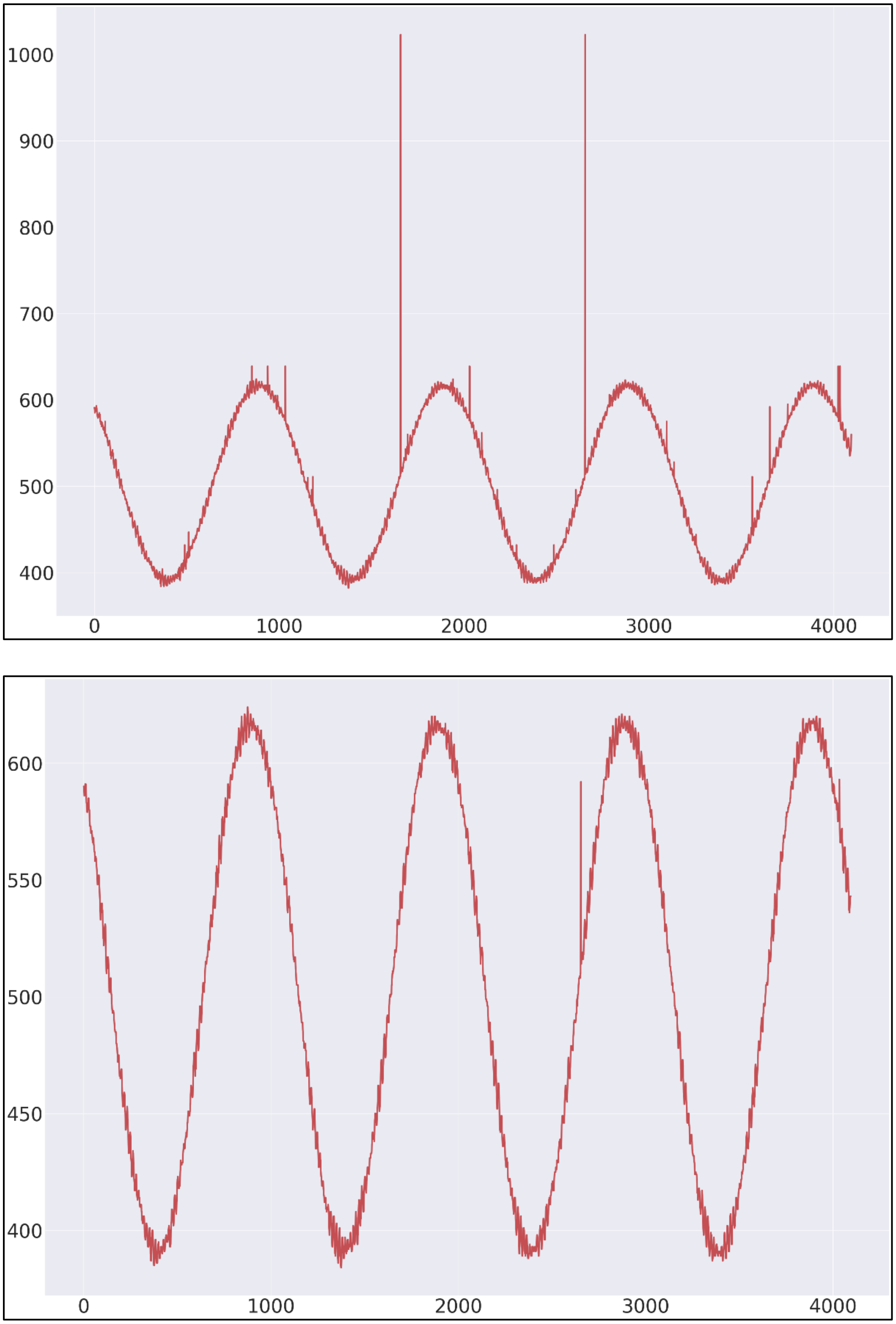


Figure 12: Matplotlib Graph of Sine Wave

After obtaining a reference value, the next step was to vary the gain and clock for the given signal. The OpenADC filters a PWM signal produced by the OpenADC Interface. The OpenADC converts the PWM signal to a gain value between 0 and 1 volts. The duty cycle of the PWM signal ranges from 0 to 30.3%. Increasing the duty cycle results in an increase in the gain. For low gain mode, the OpenADC applies a gain between -4.5 dB to +43.5 dB. In high gain mode, the OpenADC applies a gain between +7.5 dB to +55.5 dB. In Figure 13, the red line is a plot of the 100 KHz sine wave at a sample rate of 50 MHz and a gain of +7.5 dB. The blue line is at a sample rate of 100 MHz and a gain of +7.5 dB. The green line is at a sample rate of 50 MHz and a gain of +12 dB. The same sine wave signal adjusts in amplitude and period due to the configuration options set for the OpenADC.

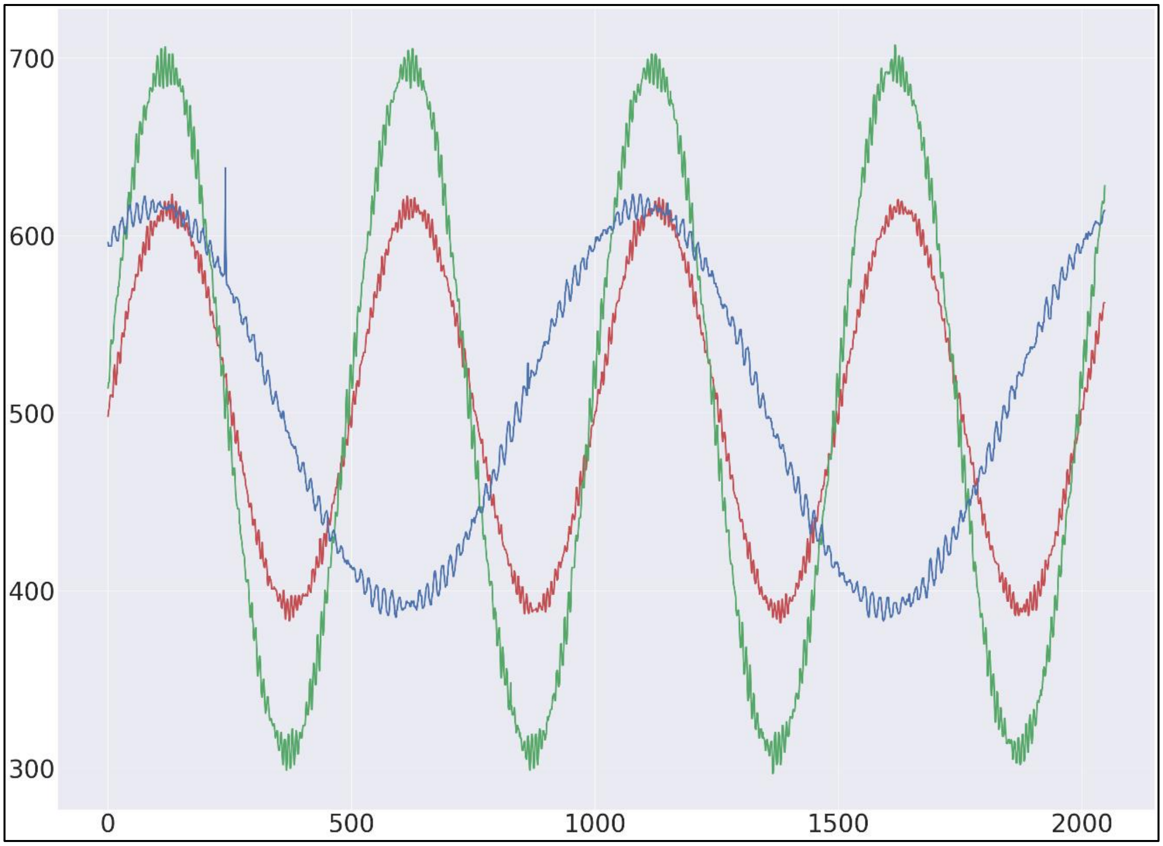


Figure 13: Plots with Different Gains and Rates

As mentioned previously, the two modes of capture are acquisition and downsample. When capturing a fixed number of traces smaller than the FIFO buffer size, all traces reached the Python software without any loss. Two primary factors impacted the success of acquisition of all traces when continuously capturing or when the fixed size was larger than the FIFO buffer. These two factors were the target clock rate and what the python software did with the traces after receiving them from the DMA. At higher target clock frequencies, the latency of the DMA for a block exceeded the latency of the input to the FIFO, and consequently the FIFO fills and traces are lost. At lower clock rates, adding delay in Python between DMA receives had a similar effect. By pre-allocating memory in Python and incrementing the receive pointer by a block size each DMA receive, successful capture of all values was possible with higher clock frequencies and capture sizes larger than the FIFO buffer. Support for even higher clock frequencies are probable if the DMA software transitions to a programming language with faster execution such as C, however, limiting samples to the maximum size of a FIFO buffer is sufficient for the intended purpose.

To add greater flexibility to the OpenADC Interface, the author added a fourth system register called divisor. When the field for calculate mean is set, this register specifies how many samples the OpenADC Interface will calculate the mean over before forwarding the mean to the FIFO. Otherwise, samples are discarded based on the divisor value. This setting allows the OpenADC to capture at a higher clock frequency than the rate of transfer of the DMA. During default operation of the OpenADC Interface the value of the divisor system register is zero and all samples will be forwarded to the FIFO.

Acquisition Results

After completion of the hardware and software components that are part of FOBOS v3 acquisition, the next step was to evaluate the effectiveness of capture using the various configuration options. The first test determined the maximum transfer rate of DMA for given block sizes. Using an acceptable block size determined from the first test, the second test consisted of verification that all counter values are captured for a given clock rate and number of samples.

Maximum Transfer Rate

This test attempted to determine the maximum transfer rate of the DMA for Python code based on the block size. The Python “timeit” module provided the timing mechanism. Results were collected for ten iterations for each block size. Each iteration transferred 2^{14} Samples (32 KB) from OpenADC to Python. All software not directly related to DMA transfer completed prior to timing measurement start. The total bytes read are smaller than the FIFO buffer size, avoiding a stoppage due to a shortage in resources. Table 2 includes the results of this test.

Table 2: Acquisition Timing by Block Size

Block Size	Mean Latency	Latency Per Byte
100	7.092 ms	216.445 ns
200	3.587 ms	109.446 ns
300	2.351 ms	71.732 ns
400	1.843 ms	56.242 ns
500	1.486 ms	45.353 ns
600	1.162 ms	35.448 ns
700	.983 ms	30.000 ns
800	.989 ms	30.184 ns
900	.793 ms	24.201 ns
1000	.802 ms	24.485 ns

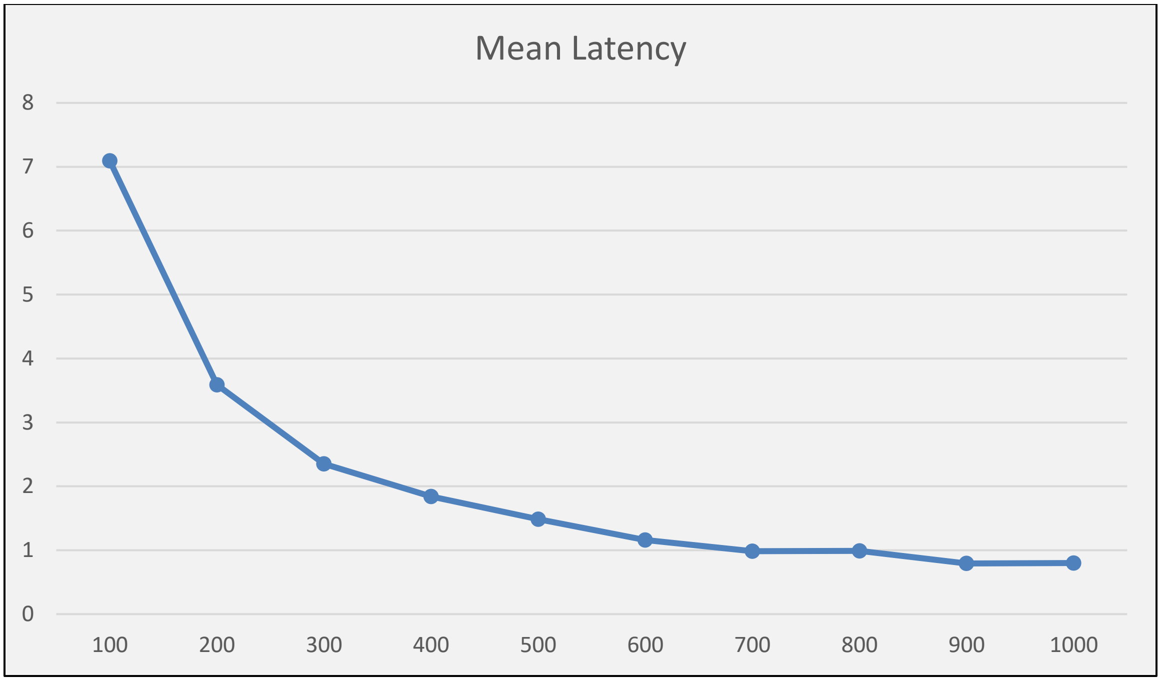


Figure 14: Mean Transfer Time of 32KB by Block Size

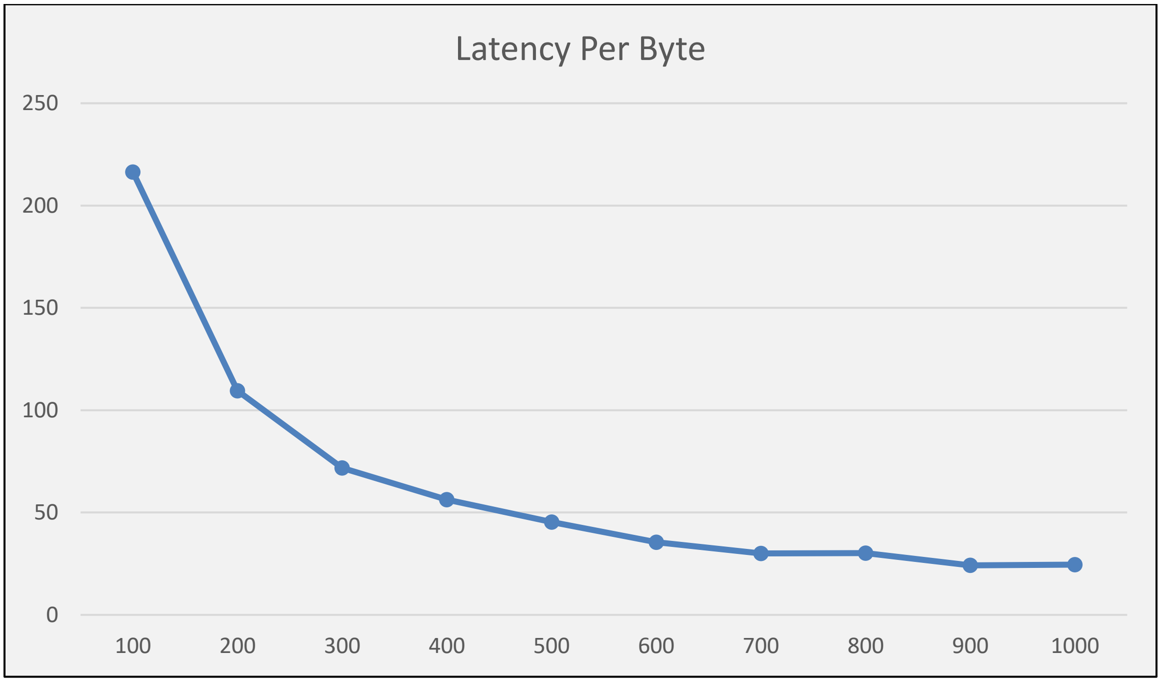


Figure 15: Transfer Time Per Byte by Block Size

$$Mean\ Latency = \frac{\sum_{i=1}^{10} timeit_i}{10}$$

$$Latency\ Per\ Byte = \frac{Mean\ Latency}{32\ KB}$$

Equation 2: Mean Latency and Latency Per Byte

Increasing the block size yields a greater rate of transfer of DMA from PL to PS memory, however, there is a diminishing rate of return. The relationships between both total transfer time and transfer time per byte to block size are plotted in Figure x and Figure y, respectively. Equation 2 are the equations used to derive these results. These results prompted limiting the block size to a maximum of 1024, represented by 10 bits in the Capture Mode register.

Performance evaluation of the Python software and data pipeline is dependent on comparison with the data rate of the OpenADC. For the highest frequency target clock rate of 105 MHz, the data rate of the OpenADC is 4.762 ns per byte. This value was calculated using Equation 3. Based on this result, the transfer rate into Python is unable to prevent the loss of samples for capture sizes larger than the FIFO buffer. The size of the FIFO buffer is 256 KB to minimize the impact of this limitation. This buffer size corresponds to 128 kilosamples, which appropriately bounds the output of the control loop.

$$Data\ Rate\ (seconds\ per\ byte) = \frac{1}{Frequency \times Bytes\ per\ Sample}$$

Equation 3: Data Rate of OpenADC

CONCLUSION AND REMAINING WORK

As the number of independent components executing on a given hardware device increases, attacks that associate device power consumption to leakage of secret data are more challenging. Refinement of measurements based on intermediate results will enable more successful side-channel analysis. To achieve this, the research approach married the statistic processing of side-channel analysis with the control logic applied to feedback-loops.

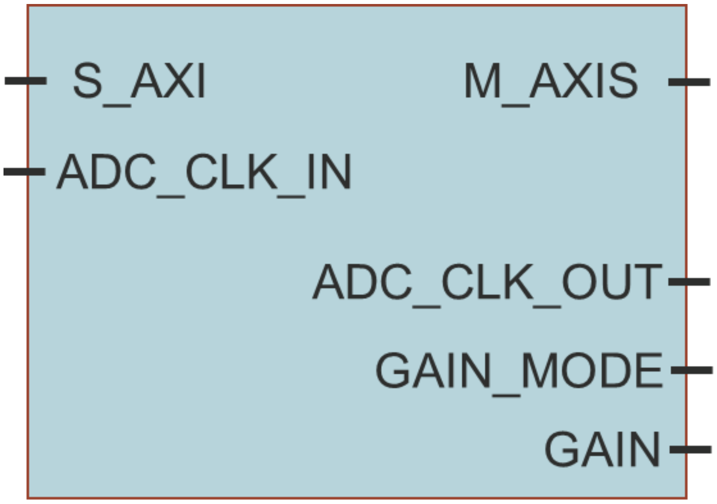
The aim of this research was to develop a platform capable of implementing a closed-loop feedback system for side-channel and fault attacks. This involves controlling the target of the attack, measuring feedback, analyzing feedback relative to a reference value, and altering system to reach desired results. To support measurement acquisition at higher data rates, the sensor component of the system was implemented in hardware. The analysis and control of the system is in software to ensure a flexible implementation.

One variant of control systems possible on this platform is configuration of the sensor to exceed the desired t-value range based on measured power consumption. Despite being unable to test with the integration of all FOBOS v3 components, each functional component was independently verified. Similar control systems may be created based on the fundamental concepts demonstrated in this research.

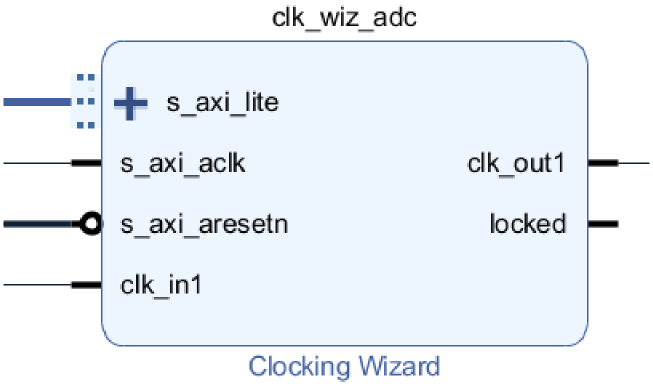
The acquisition component of FOBOS v3 is complete. The next task is integration of the acquisition component with the trigger and DUT control. This includes any modifications to support coordination of the entire attack flow. After integration is complete, the control system logic put forth in this paper may be instrumented on the fully functional platform.

APPENDIX A: FOBOS ACQUSITION COMPONENTS

OpenADC Interface:



Clocking Wizard:



Software Support:

```
# Program PL
overlay = Overlay("./fobosAcq.bit")

# Instantiate drivers and FobosAcquisition Class
dmaRecv = overlay.axi_dma_from_openadc_to_ps
openadcInterface = overlay.openadc_interface_v1_0_0
adcClk = overlay.clk_wiz_adc
fobosAcq = FobosAcquisition(openadcInterface, dmaRecv,
adcClk)

# Capture 4096 Traces
traces = fobosAcq.captureFixedTraces(4096)
if len(traces) != 4096:
    print("failed to capture appropriate number of
traces %d vs. %d" % (len(traces), 4096) )

# Verify that no loss occurred when testing with binary
counter
fobosAcq.verifyContinuous(traces)
fobosAcq.plotTraces(traces)
```

REFERENCES

[1] H. Bar-El et al. “The Sorcerer’s Apprentice Guide to Fault Attacks”. In: Proceedings of the IEEE94.2 (2006), pp. 370–382.issn:0018-9219. DOI: 10.1109/JPROC.2005.862424.

[2] E. Brier, C. Clavier, and F. Olivier. “Correlation power analysis with a leakage model”. In: Berlin, Germany, 2004, pp. 16 –29.

[3] Eric Brier, Christophe Clavier, and Francis Olivier. “Correlation power analysis with a leakage model”. English. In: vol. 3156. Cam-bridge, MA, United states, 2004, pp. 16 –29. URL: http://dx.doi.org/10.1007/978-3-540-28632-5_2.

[4] *Daredevil*. URL: https://github.com/SideChannelMarvels/Daredevil.

[5] Franck Dernoncourt. “Introduction to fuzzy logic”. In: (Jan. 2013).

[6] Richard Dorf and Robert Bishop. “Introduction to Control Systems”. In: Modern Control Systems. Ed. by Andrew Gilfillan. Upper Saddle River, NJ: Pearson Education, Inc., 2011, pp. 1–34. ISBN: 978-0-13-602458-3.

[7] *DPA Workstation Analysis Platform*. URL: https://www.rambus.com/security/dpa-countermeasures/dpa-workstation-platform/.

[8] *Getting Started*. URL: https://wiki.newae.com/Getting_Started.

[9] L. Guo et al. “A Chosen Plaintext Differential Power Analysis Attack on HMAC - SM3”. In: *2015 11th International Conference on Computational Intelligence and Security (CIS*). 2015, pp. 350–353. DOI: 10.1109/CIS.2015.91.

[10] Y. Hori et al. “SASEBO-GIII: A hardware security evaluation board equipped with a 28-nm FPGA”. In: *The 1st IEEE Global Conference on Consumer Electronics 2012*. 2012, pp. 657–660. DOI:10.1109/GCCE.2012.6379944.

[11] Gabriel Hospodar et al. “Machine learning in side-channel analysis: A first study”. English. In: *Journal of Cryptographic Engineering* 1.4 (2011), pp. 293 –302. ISSN: 21908508. URL: http://dx.doi.org/10.1007/s13389-011-0023-x.

[12] Marc Joye, Pascal Paillier, and Berry Schoenmakers. “On second-order differential power analysis”. English. In: vol. 3659. Edinburgh, United Kingdom, 2005, pp. 293 –308.

[13] N. Kamoun, L. Bossuet, and A. Ghazel. “Experimental implementation of DPA attacks on AES design with flash-based FPGA technology”. In: Piscataway, NJ, USA, 2009. URL: http://dx.doi.org/10.1109/SSD.2009.4956747.

[14] Paul Kocher, Joshua Jaffe, and Benjamin Jun. “Differential power analysis”. English. In: vol. 1666. Santa Barbara, CA, United states,1999, pp. 388 –397.

[15] Paul Kocher et al. “Introduction to differential power analysis”. In: *Journal of Cryptographic Engineering* 1.1 (2011), pp. 5–27. ISSN: 2190-8516. DOI: 10.1007/s13389-011-0006-y. URL: https://doi.org/10.1007/s13389-011-0006-y.

[16] Paul C. Kocher. “Timing attacks on implementations of diffie-hellman, RSA, DSS, and other systems”. English. In: vol. 1109. Santa Bar-bara, CA, United states, 1996, pp. 104 –113.

[17] Z. Martinasek, V. Clupek, and T. Krisztina. “General scheme of differential power analysis”. In: *2013 36th International Conference on Telecommunications and Signal Processing (TSP).* 2013, pp. 358–362. DOI: 10.1109/TSP.2013.6613952.

[18] H. Martín et al. “Fault Attacks on STRNGs: Impact of Glitches, Temperature, and Underpowering on Randomness”. In: *IEEE Trans-actions on Information Forensics and Security* 10.2 (2015), pp. 266–277.issn: 1556-6013.doi:10.1109/TIFS.2014.2374072.

[19] M. Masoomi, M. Masoumi, and M. Ahmadian. “A practical differential power analysis attack against an FPGA implementation of AES cryptosystem”. In: *2010 International Conference o Information Society.* 2010, pp. 308–312.

[20] M. Matsubayashi, A. Satoh, and J. Ishii. “Clock glitch generator on SAKURA-G for fault injection attack against a cryptographic circuit”. In: *2016 IEEE 5th Global Conference on Consumer Electronics*. 2016, pp. 1–4. DOI: 10.1109/GCCE.2016.7800490.

[21] L. Mazur and M. Novotný. “Differential power analysis on FPGA board: Boundaries of success”. In: *2017 6th Mediterranean Conference on Embedded Computing (MECO).* 2017, pp. 1–4. DOI: 10.1109/MECO.2017.7977168.

[22] I. Mizumoto et al. “Adaptive PID control system design for non-linear systems”. English. In: *International Journal of Modelling Identification and Control* 6.3 (200/), pp. 230 –8. ISSN: 1746-6172. URL: http://dx.doi.org/10.1504/IJMIC.2009.024463.

[23] G. D. Natale et al. “Power consumption traces realignment to improve differential power analysis”. In: *14th IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems*. 2011, pp. 201–206.doi:10.1109/DDECS.2011.5783080.

[24] C. O’Flynn and Zhizhang Chen. “A Case Study of Side-Channel Analysis Using Decoupling Capacitor Power Measurement with the OpenADC”. In: Berlin, Germany, 2013, pp. 341 –56.

[25] Colin O’Flynn. “A Framework for Embedded Hardware Security Analysis”. PhD thesis. Halifax, Nova Scotia: Dalhousie University, June 2017.

[26] *OpenSCA*. URL: http://opensca.sourceforge.net/index.html.

[27] A. Pellegrini, V. Bertacco, and T. Austin. “Fault-based attack of RSA authentication”. In: 2010 *Design, Automation Test in Europe Conference Exhibition (DATE 2010)*. 2010, pp. 855–860. DOI: 10.1109/DATE.2010.5456933.

[28] Roberta Piscitelli, Shivam Bhasin, and Francesco Regazzoni. “Fault Attacks, Injection Techniques and Tools for Simulation”. In: *Hardware Security and Trust: Design and Deployment of Integrated Circuits in a Threatened Environment*. Ed. by Nicolas Sklavos et al.Cham: Springer International Publishing, 2017, pp. 27–47. ISBN: 978-3-319-44318-8. DOI: 10.1007/978-3-319-44318-8_2. URL: https://doi.org/10.1007/978-3-319-44318-8_2.

[29] E. Prouff, M. Rivain, and R. Bevan. “Statistical Analysis of Second Order Differential Power Analysis”. In: *IEEE Transactions on Computers* 58.6 (2009), pp. 799–811. ISSN: 0018-9340.doi:10.1109/TC.2009.15.

[30] *PYNQ*: Python Productivity for Zynq. URL: http://www.pynq.io/.

[31] P. Rathnala, T. Wilmshurst, and A. Kharaz. “A practical approach to differential power analysis using PIC micrcontroller based embedded system”. *In: 2014 6th Computer Science and Electronic Engineering Conference (CEEC).* 2014, pp. 58–62. DOI: 10.1109/CEEC.2014.6958555.

[32] E. Saeedi and Yinan Kong. “Fuzzy analysis of side channel information”. English. In: Piscataway, NJ, USA, 2014, 5 pp. URL: http://dx.doi.org/10.1109/ICSPCS.2014.7021074.

[33] Francois-Xavier Standaert. “Introduction to Side-Channel Attacks”. In: *Secure Integrated Circuits and Systems*. Ed. by Ingrid M.R. Vebauwhede. Boston, MA:

Springer US, 2010, pp. 27–42. ISBN: 978-0-387-71829-3. DOI: 10.1007/978-0-387-71829-3_2. URL: https://doi.org/10.1007/978-0-387-71829-3_2.

[34] Rajesh Velegalati and Jens-Peter Kaps. *Towards a Flexible, Open-source BOard for Side-channel analysis (FOBOS).* Cryptographic architectures embedded in reconfigurable devices, CRYPTARCHI 2013. 2013.

[35] I. Verbauwhede, D. Karaklajic, and J. M. Schmidt. "The Fault Attack Jungle - A Classification Model to Guide You". In: *2011Workshop on Fault Diagnosis and Tolerance in Cryptography*. 2011, pp. 3–8. DOI: 10.1109/FDTC.2011.13.

[36] Aiping Wang, P. Afshar, and Hong Wang. "Complex stochastic systems modelling and control via iterative machine learning". English. In: *Neurocomputing* 71.13-15 (2008/08/), pp. 2685 –92. ISSN: 0925-2312. URL: http://dx.doi.org/10.1016/j.neucom.2007.06.018.

[37] Sixiang Wang et al. "Differential power analysis attack and countermeasures on MCrypton". In: *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC).* 2016, pp. 167–172.doi:10.1109/IMCEC.2016.7867194.

[38] Jasper G. J. van Woudenberg, Marc F. Witteman, and Bram Bakker. "Improving Differential Power Analysis by Elastic Alignment". In: *Topics in Cryptology – CT-RSA 2011*. Ed. by Aggelos Kiayias.Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 10–119.isbn: 978-3-642-19074-2.

[39] B. Yuce, N. F. Ghalaty, and P. Schaumont. "Improving Fault Attacks on Embedded Software Using RISC Pipeline Characterization". In: *2015 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. 2015, pp. 97–108. DOI: 10. 1109 / FDTC .2015.16.

[40] L.A. Zadeh. "Fuzzy sets". In: *Information and Control* 8.3 (1965), pp. 338 –353. ISSN: 0019-9958. DOI: https://doi.org/10.1016/S0019-9958(65)90241-X. URL: http://www.sciencedirect.com/science/article/pii/S001999586590241X.

[41] Xilinx, *PYNQ Block Diagram*. 2017. URL: https://pynq.readthedocs.io/en/v2.0/_images/zynq_block_diagram.jpg.

[42] Xilinx, *Zynq Interfaces*. 2018. URL: https://pynq.readthedocs.io/en/v2.3/_images/zynq_interfaces.png.

[43] ARM, *AMBA AXI and ACE Protocol Specification: AXI3, AXI4, and AXI4-Lite*. 2011.

[44] NewAE, *OpenADC Product Datasheet*. 2014. URL: http://www.newae.com/files/openadc-datasheet.pdf.

[45] Digilent, *Analog Discover Technical Reference Manual*. 2015. URL: https://reference.digilentinc.com/_media/analog_discovery:analog_discovery_rm.pdf.

[46] Riscure, *Riscure Inspector*. 2017. URL: https://www.riscure.com/uploads/2017/08/inspector_brochure.pdf

BIOGRAPHY

Matthew Carter graduated from Tabb High School, Yorktown, Virginia, in 2006. He received his Bachelor of Science in Computer Engineering from University of Virginia in 2010.