PicoEMP: A Low-Cost EMFI Platform Compared to BBI and Voltage Fault Injection using TDC & External VCC Measurements

Colin O'Flynn
Electrical & Computer Engineering Department
Dalhousie University
Halifax, Canada
colin@oflynn.com

***Abstract*—Electromagnetic Fault Injection (EMFI) has been demonstrated to be useful for both academic and industrial research. Due to the dangerous voltages involved, most work is done with commercial tools. This paper introduces a safety-focused low-cost and open-source design that can be built for less than $50 using only off-the-shelf parts.**

The paper also introduces an iCE40 based Time-to-Digital Converter (TDC), which is used to visualize the glitch inserted by the EMFI tool. This demonstrates the internal voltage perturbations between voltage, body biasing injection (BBI), and EMFI all result in similar waveforms. In addition, a link between an easy-to-measure external voltage measurement and the internal measurement is made. Attacks are also made on a hardware AES engine, and a soft-core RISC-V processor, all running on the same iCE40 FPGA.

The platform is used to demonstrate several aspects of fault injection, including that the spatial positioning of the EMFI probe can impact the glitch strength, and that the same physical device may require widely different glitch parameters when running different designs.

I. Introduction

Fault injection is a method of introducing computational errors (faults) into digital devices. The danger of faults in cryptographic algorithms was quickly realized [1], which has led to a productive series of attacks in recent years.

Open-source [2] and commercial tools for fault injection have become available to support this effort. Of the injection methods, electromagnetic fault injection has the advantage of requiring a limited amount of device preparation. This has made EMFI a popular method of fault injection in both academia & industry. A more detailed overview of EMFI is given in [3].

Examples of EMFI on practical targets include attacks on Android smart phones [4], standard desktop and server computer processors [5], and automotive platforms [6], [7].

Practical work on EMFI including answering the question of triggering such injections, demonstrated against AES [8]. As will be shown in this work, the parallels between voltage glitching and EMFI mean other high-level demonstrations such as voltage glitching against Linux [9] or on microcontroller bootloaders [5] should be applicable with EMFI.

Fault attacks on cryptographic algorithms are known [10], including against RSA [1], AES [11], ECDSA [12], and Dilithim [13]. Academic work in fault injection often uses EMFI as an injection method since it requires minimal (or no) modifications to the target, which can be seen in work such as [13], [14], [15],

Of course the defender is interested in countermeasures [16] and detectors [17], [18] to protect systems from such faults.

In such a rich body of previous work, this work focuses on several practical improvements from the current literature.

1) it introduces a low-cost and open-source EMFI platform that focuses on operator safety foremost using a fully isolated architecture,
2) it introduces an open-source Time-to-Digital Converter (TDC) reference platform that can run on the popular iCE40 FPGA,
3) it compares the resulting measurements of different EMFI platforms using the reference platform, and
4) it compares (internal) TDC-based measurements to external measurements of the VCC pin.

A GIT repository is available at https://github.com/newaetech/chipshouter-picoemp with the related open-source information, and all data from this paper is availabe at https://github.com/colinoflynn/picoemp-tdc-paper.

The reference platform allows comparison of different fault injection mechanisms measured using both on-die measurements (with a TDC-based ADC) [19], as well as external measurements. Compared to previous work [20], the external measurement has advantages in (a) black-box attacks where an attacker cannot add TDC cores, and (b) microcontrollers and other devices which cannot implement a TDC.

This work also ties together several previous papers which have looked at different fault injection mechanisms on separate devices. Based on the previous work we expected to see similar internal power rail bounces between voltage [19] and EMFI [20], but a study using the *same* device was missing which helps quantify this. This paper concretely shows that the injected waveform between at least three major fault injection methods (EMFI, BBI, and voltage glitching) are similar.

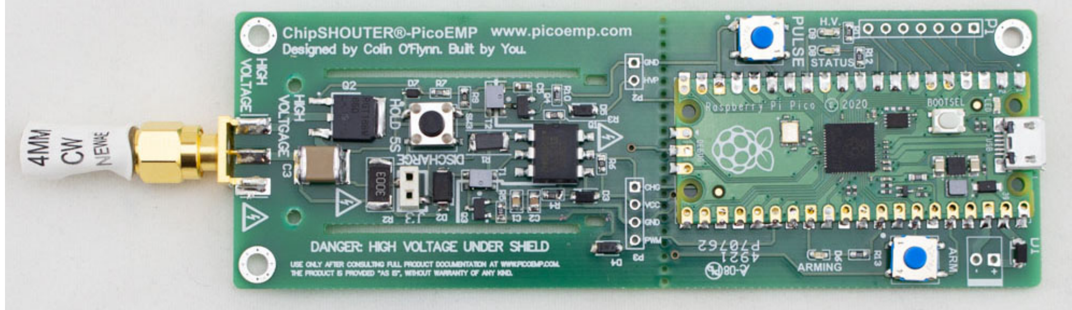

Figure 1: The PicoEMP showing both high-voltage (on left) and low-voltage (on right) circuitry..

Having circuitry on a device that can "self-inject" glitches can be useful for testing or calibration [21]. Practically, this circuitry will be limited to influencing internal voltage rails, so having a link between voltage fault injection and other methods (EMFI & BBI) provides valuable information for the use of such circuitry.

Finally, more practical glitch effects are demonstrated using two different practical targets: a RISC-V soft-core running on the same iCE40 microcontroller, and a hardware AES core implemented on the iCE40. This allows a more direct link between the on-die voltage measurements and the observed glitch effects, which can be compared to previous findings which showed that TDC measurements could be used to calibrate the effectiveness of an EMFI glitch [20].

This allows a link between more experimental glitch results and purely measurement-based results. This is valuable for comparing future platforms, as well as for building glitch detectors (which may be tested by comparing them to TDC measurements on the same device). The external measurement technique compared in this paper is also helpful for validating glitch detectors in production devices, where the TDC is not implemented in the final production device.

A. Choice of Target

This work targets the iCE40 FPGA as an example target. As a target for this work it has several advantages: it is low-cost and available on many development boards including iCEBreaker, iCEStick, etc. Second, it has a complete open source toolchain that allows rapid synthesis runs [22] (a feature that will be used in this work to tune the TDC). Finally — unlike larger FPGAs which are normally used only in more expensive industrial controllers and equipment, the iCE40 is found in many consumer goods. This includes the iPhone [23], Samsung Galaxy S5 [24], and Square Terminal credit card reader [25]. The wide availability of low-cost & real-life targets makes it an interesting match for a low-cost EMFI tool, as an entire EMFI test suite could be built for less than $100, and becomes suitable for undergraduate, high-school, or hobbyist environments.

B. Related Work - EMFI

As discussed previously, EMFI is a powerful tool that is used well in both industrial and academic settings. These previously mentioned works often have a fault model, where experimentation to understand realistic fault models has been done in [26], [27]. Validating these fault models has been done by comparing EMFI across hardware and software implementations of the same algorithm (such as hardware & software AES implementations in [28]).

The root cause of the faults appears to be that EM fault injection causes timing errors [29], which can also be seen to target data transfers with high accuracy [30]. All of these attacks, of course, require an EMFI tool.

The design of EMFI tools is covered in the literature from at least 2017 [26], [31], [32], [33], [34], [35]. As will be explained in Section II the PicoEMP described herein differs as it offers a high degree of inherent safety matching commercial tool offerings.

A related attack to EMFI is Body Biasing Injection (BBI), which uses a physical contact to inject a high voltage pulse into the backside of the die [36]. This is described as a type of localized electromagnetic fault injection in previous literature. The coupling mechanism is different, but has similar effects which depend on spatial positioning of the probe.

C. Related Work - TDC

Use of a Time to Digital Converter (TDC) as a clever method to visualize the internal power rails of a device during a glitch operation was demonstrated in [37], which was the basis for a more advanced analysis showing how both positive and negative voltage glitchs all result in similar ringing as measured internally [19].

Specifically for EMFI, in [20] the authors use an array of TDC circuits on an Artix-A100 microcontroller to visualize not only the waveform of the EMFI in time, but also spatially by implementing arrays of TDCs. In addition a linear feedback shift register (LFSR) is used to gauge the effectiveness of a fault on a simple digital circuit. We build off this by using more complete examples in the FPGA to demonstrate the "real-life" glitch effectiveness (as well

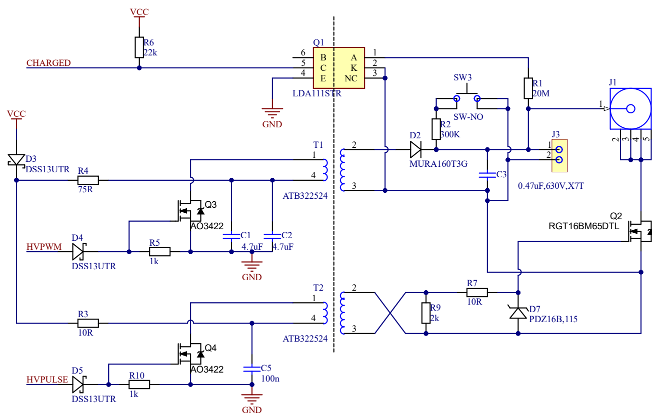

Figure 2: The PicoEMP high-voltage architecture fully isolates the high-voltage supply.

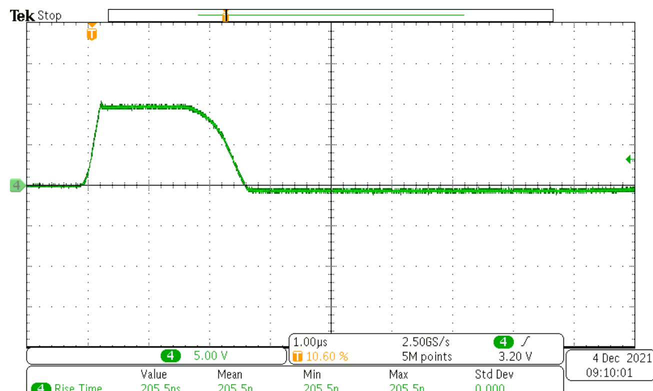

Figure 3: The PicoEMP gate drive waveform measured at Q2.

as releasing the full design files for recreating the TDC). In addition, the previous work have not compared voltage glitching & EMFI on the same platform to understand whether TDC measurements could be considered "source-agnostic". That is, if one has a TDC sensor, is it sufficient to understand the effectiveness of the glitch regardless of whether the source of the glitch was EMFI or voltage glitching.

II. PicoEMP Design

While PicoEMP is a low-cost design (it can be built for under $50), it also targets a design which can be built easily and repeatably. It uses all off-the-shelf parts, and almost all parts are surface mount (allowing assembly at larger scale). PicoEMP is designed to make EMFI a commodity attack surface, challenging the narrative that EMFI is a higher-end attack than voltage or clock glitching [3].

EMFI tools typically use either a direct-drive or coupled architecture. With direct-drive the capacitor bank is directly switched onto the output, which can be done using a low-side drive (more dangerous but cheaper) or high-side drive (safer but more complex to design) [38]. Some of the earliest open-source EMFI tools used a low-side drive and commented on the danger level this entails in the presentation [31]. Commercial tools use a high-side drive, such as seen in the ChipSHOUTER schematic [1]. Other open source designs that use a coupled architecture allow a simple drive design but with a much higher safety margin, with the downside of requiring more analysis of the probe matching to the coupling mechanism [35].

PicoEMP differs from these previous designs by fully isolating the high-voltage supply from the user trigger inputs. It does this by using a transformer to couple the high-voltage supply along with the gate drive waveform. The use of gate drive transformers (GDT) to isolate the MOSFET is a well-known solution with MOSFETs and IGBTs [39].

The PicoEMP architecture is unique, as rather than using off-the-shelf GDT, a transformer designed for high-voltage photo-flash chargers in small cameras and phones, the ATB322524, is used. The schematic of the PicoEMP high-voltage circuit shows that transformers T1 and T2 are the same part number, which simplifies construction, as no custom-wound parts are needed. Transformer T2 provides both isolation and increases the voltage at the logic level to the 10V level required by the gate.

Typically, MOSFETs or IGBTs are very sensitive to too much voltage on the gate, which causes "punch-through" and immediately destroys the device. To prevent this, diode D7 and resistor R7 limit the maximum voltage (with help from the gate capacitance that slightly slows the rising edge). This forms an extremely effective gate drive circuitry, as can be seen in Figure 3, which is the gate drive voltage during a pulse. The rise time at the gate is relatively slow (200 nS) compared to commercial tooling, but as will be seen, it still creates effective glitches.

The control side of the PicoEMP is provided by a Raspberry Pi Pico (not shown on the schematic, but visible in Figure 1). To drive the high-voltage circuitry, a waveform of 2.5 KHz with a duty cycle of 1.46% is used. This charges capacitor C3 to 310 V. Feedback about the capacitor voltage is provided with optoisolator Q1. This generates a voltage that is related to the charge voltage, and is primarily used to provide a "charged" signal when the voltage on C3 exceeds approximately 220 V. The transfer function for the optoisolator will vary substantially between devices, so providing accurate charge voltage requires a per-unit calibration to map the optoisolator output to the capacitor voltage. A header (J3) is provided to allow advanced users to easily perform this calibration if they wish.

High-voltage isolation was validated with a Sentry 20 Plus

[1] https://github.com/newaetech/ChipSHOUTER/blob/master/documentation/NPCA-CW520-ChipSHOUTER-07_Schematic.pdf

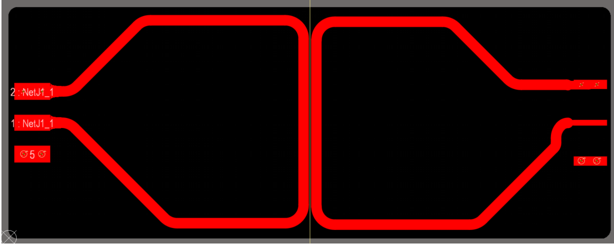

Figure 4: A simple planar calibration board, has a SMA to connect to the EMFI tool on the left, and a BNC to connect to the oscilloscope on the right.

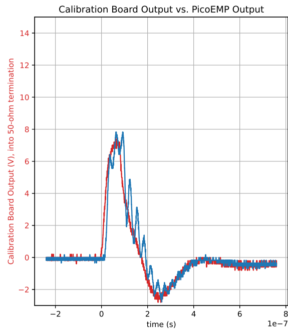

Figure 5: PicoEMP Cal (input & output)

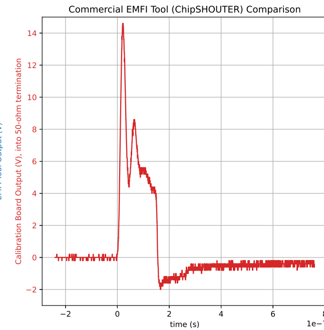

Figure 6: ChipSHOUTER Cal (output only)

hipot tester, which applied 1 kV for 60 seconds between the control side & SMA output connector. The tester reported no detectable current (0.0 mA) during the test, indicating the isolation can safely keep the high voltage from reaching the control logic.

The physical construction uses an off-the-shelf plastic enclosure (Hammond 1551BTRD) as a shield over the entire high-voltage circuitry. The only exposed high-voltage area is the SMA connector, which has limited user safety risk due to the narrow pulses generated by the PicoEMP. The use of the GDT architecture means that the output is incapable of being turned "on" continuously, as the GDT can only pass narrow pulses. This further improves the safety margin, since even if the probe is removed the shock hazard is minimized.

The transformers T1 & T2 are physically small, which allows them to easily fit under the shield. This is one of the primary limitations of PicoEMP: it's slow recharge rate due to this small transformer. The capacitor storing the high voltage is only 0.47 μF to avoid overloading the small transformer during charge. Despite these limitations, PicoEMP produces useful results.

To compare the pulse output capability, a simple calibration board is used. This board uses a standard 2-layer PCB, with the top layer shown in Figure 4. The design files for this board are available as part of the PicoEMP repository. The waveform measured at the output of this calibration board with a 50 ohm terminator on the oscilloscope, and also at

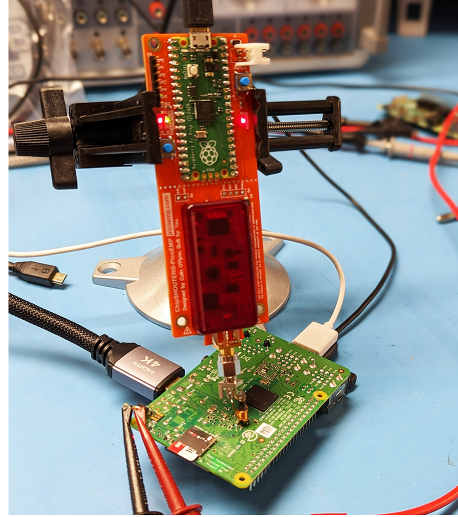

Figure 7: Using PicoEMP on a Raspberry Pi target.

the output of the PicoEMP using a high voltage differential probe, and these results are shown in Figure 5. The scaling between the two axes is 11.5$\times$. Figure 6 includes the output of the calibration board for a comparable glitch coming from a commercially available EMFI tool (ChipSHOUTER). From this output, we can see that the EMFI strength is comparable to that coming from larger commercial tools. But the waveform alone is not sufficient to prove this, so we will now perform several demonstrations of this tool in practical scenarios.

A. *Testing Against Raspberry Pi Model 3B+*

The remainder of this paper will concentrate on the evaluation of this tool on an iCE40 FPGA, and comparison to other techniques. As a short prelude to that, a demonstration of the PicoEMP on a "practical" target will also be included to demonstrate that the fault effect is sufficient to impact a real-life target. The real-life target is a Raspberry Pi Model 3B+, which is running at 1400 MHz CPU speed with default configurations. The same board was targeted by EMFI in [40]. The target of the fault here is the `pycryptodome` library version `3.1`, which is vulnerable to the well-known fault during the RSA signing operation [1]. Later versions of the library include a check to avoid returning a faulty signature.

The PicoEMP is pointed at the backside of the Raspberry Pi Model 3B+ during the signing operation as in Figure 7. This generates a faulty signature which allows recovery of the RSA private key. The files to run this demonstration are linked from the PicoEMP repository. Note that this is running Python as a regular process in the operating system, and relies primarily on the fact that the fault is *most likely* injected into the RSA signature operation. This is not guaranteed, as shown in Table I, which demonstrates that

Table I: Results of RSA Fault Attack on Raspberry Pi 3B+

Result	Count	Percentage
No Impact	33	30 %
System Hang	1	0.9 %
Application Crash	45	41 %
RSA Fault (invalid)	4	3.7 %
RSA Fault (success)	26	24 %

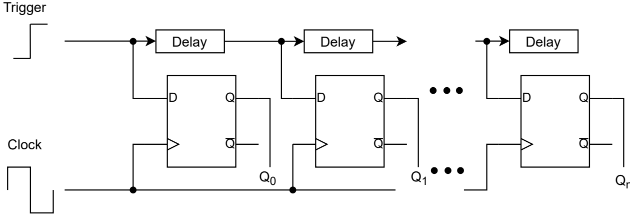

Figure 8: A simple Time to Digital Converter (TDC) uses a low-skew clock routing fabric to sample all flip-flops at the same time, with a delayed signal at each flip-flop input.

crashes of the operating system or Python interpreter are also possible.

Application crashes are most commonly manifested as a `segmentation error`, but various other errors, including `illegal instruction`, and `Memory Allocation Failed` indicate that more interesting OS fault attacks may be possible.

III. iCE40 Based TDC

The use of Time-to-Digital Converter (TDC) in FPGAs is well known, with various resources including online tutorials [41]. The TDC uses a combination of a delay element and flip-flops, as shown in Figure 8. The clock for the flip-flops transits on a low-skew clock routing fabric, while the data path includes delay elements.

To use the TDC for voltage measurements, we take advantage of the fact that the delay through the delay elements will vary with voltage. Because the clock is in a lower-skew clock routing fabric, it does not see the same level of variation with changing voltage. In this case, the trigger and clock are normally the same signal, but we may need to add some additional delay at the start of the trigger signal.

The delay element is most commonly a carry chain due to the optimized propagation time in the FPGA for this element, but any sort of element can be used as the delay. To compare two such options in the iCE40UP5K, the average propagation time of both a carry-chain based element (using the `SB_CARRY` primitive) and a LUT based element (using the `SB_LUT4` primitive) were measured. This measurement was done by routing a 25 MHz clock through the elements and comparing the phase shift as 1,2,3, and 4 delay elements are inserted. The first element is always located at a fixed point, and the input and output locations are fixed, as well.

Table II: iCE40 Delay Element Measurements

Using SB_CARRY			Using SB_LUT4		
V_{int}	$\overline{delay}$	σ_{delay}	V_{int}	$\overline{delay}$	σ_{delay}
1.1 V	0.52 nS	0.21 nS	1.1 V	2.09 nS	0.82 nS
1.2 V	0.36 nS	0.16 nS	1.2 V	1.44 nS	0.53 nS
1.3 V	0.30 nS	0.12 nS	1.3 V	1.12 nS	0.42 nS

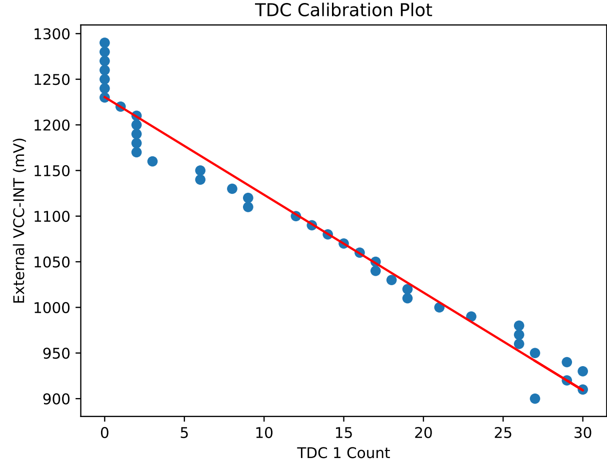

Figure 9: The TDC calibration values, measured points in blue, calibration line used in red.

Table II shows the results of this experiment. Note that the reported σ_{delay} is due to variations in the average per-element delay as the number of elements differs. This is an artifact of slightly changing routing paths required to change the number of delay elements, as the measurement itself has much lower jitter, the measurement taken with a 10 GS/s oscilloscope, providing 100 pS resolution. It can be seen from Table II that the delay varies with voltage, and that the `SB_CARRY` provides the lowest delay. Typically, the carry chain has the additional advantage that the FPGA routing between carry chain elements is more compact, which is also seen in the lower σ_{delay} for the `SB_CARRY` as well.

Using the IceStorm toolchain with the apio flow, changing the number of "calibration delay" elements in the TDC is done by recreating the TDC with the required configuration. This means that there is virtually unlimited flexibility in the TDC design parameters, since the actual TDCs are re-implemented & reprogrammed into the FPGA in a matter of seconds. There is no additional logic required to change TDC settings (such as a limited number of steps in a mux). To calibrate the output of the TDC with known voltages, a DAC80501 (available in an off-the-shelf development board) is used to set the core voltage of the FPGA. This allows calibration of the TDC on-the-fly for accurate voltage measurements.

The TDC architecture is shown in Figure 10. The TDC output is written to an embedded block ram (EBR), allowing for a time-series sampling of the TDC output. To keep the

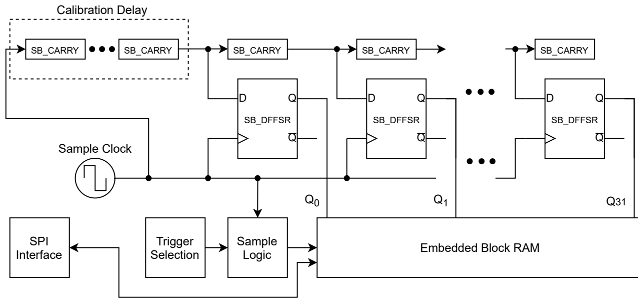

Figure 10: The TDC writes the output to a block ram, which is read using SPI.

logic as simple as possible, in order to reduce the chance of glitching the EBR core itself, a single embedded block ram is used. This limits the TDC to 128 samples.

The main downside of the iCE40 is the relatively slow fabric. The iCE40UP5K device used on the board specifies a maximum expected frequency for a 16-bit counter (similar in design to our TDC) of 100 MHz, and maximum speed of the block ram of 150 MHz. In our experimentation reliable TDC waveforms were only achieved up to around 60 MHz, the lower frequency here related to the fact that we need to account for a lower operating voltage during the glitch insertion measurement. For the rest of the experiments, the TDC is running with a 25 MHz input clock, that is internally increased with a PLL to 46 MHz.

In [42] an iCE40 based TDC is reported running at 96 MHz, but this is not for fault injection so has a stable power supply. It is also a slightly different device (iCE40HX-8k vs. our iCE40UP5K).

The clock for the TDC is running continuously, and a trigger is used to start writing the TDC sequence to the block ram. The trigger input is part of the dynamically generated Verilog code. In this work, three modes are used: the trigger runs immediately (used for calibration with a constant voltage level), the TDC triggers on an external signal (used when driven from an external glitch trigger), and finally, the TDC triggers based on a change of one of the TDC outputs (used where no external trigger is present).

IV. Measurement Setup

The ChipWhisperer-Husky platform was used to test the glitch results. This platform provides an iCE40 target (NAE-CW312T-ICE40) that is instrumented for power analysis and glitch attacks. This has a iCE40UP5K-UWG30, which is in the iCE40 UltraPlus family, in wafer level chip scale packaging (WLCSP). The WLCSP allows simple use of BBI, which was another motivation for the selection of this device [43]. The design files for the target board are available[2].

[2]https://github.com/newaetech/chipwhisperer-target-cw308t/tree/main/CW312T_ICE40UP

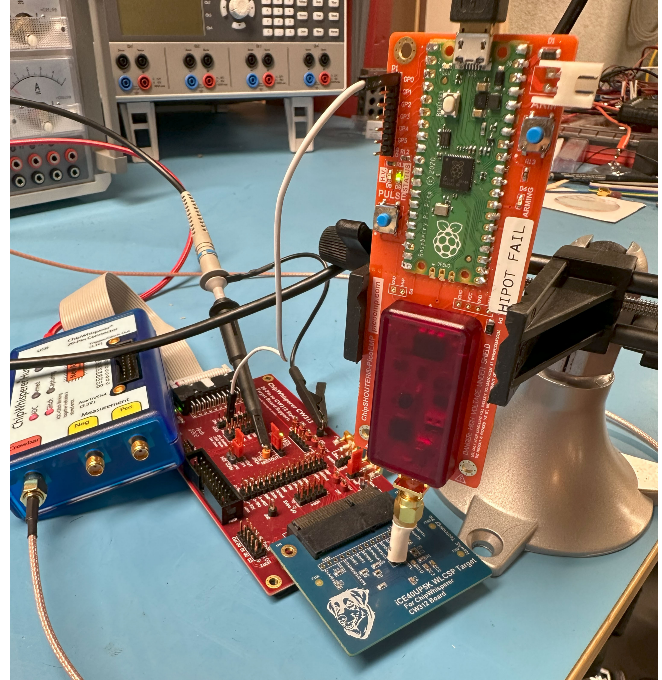

Figure 11: The PicoEMP being used to inject a glitch into the iCE40. A PicoScope 6403D (off-screen) also measures the VCC power rail from connector J3.

The ChipWhisperer-Husky is used to communicate with the target device, along with providing glitch triggering. In addition, the ChipWhisperer-Husky includes a voltage glitch generator based on a crowbar circuit [44]

To measure the glitches, an external oscilloscope (PicoScope 6403D) is connected to one of the SMA connectors on the target board (CW312). A photo showing one of the setups is in Figure 11.

To provide glitch results, three different types of targets are loaded into the iCE40 FPGA: The TDC sensor described earlier, an AES hardware accelerator core, and a soft-core RISC-V processor. The later two are briefly described here.

A. AES Hardware Faults

Fault attacks on AES allow recovery of the secret key when the attacker can obtain a faulty ciphertext. Various attacks are available targeting different areas of the AES algorithm [45], [46]. For this work, we consider the case where a faulty output ciphertext is produced, with the core continuing to operate normally on the next operation, as a successful fault.

The AES core targeted is a straightforward implementation with one round processed per clock cycle, provided by Google Vault. This core is part of an existing ChipWhisperer AES demonstration (not designed by the authors). When implemented in the iCE40 FPGA, it takes 4316 of 5280 LUTs (81%), with a maximum clock of 22 MHz. In practice,

when running on the iCE40 target board, reliable results were only obtained up to 15 MHz (at 20 MHz faulty results were often obtained even without fault injection). This is likely due to the shunt resistor of the iCE40 target board that reduces the core voltage. As mentioned above, the iCE40 fabric is relatively slow, and the large area consumption of the AES core constrains the place & route algorithm.

Communication with the core is handled via a serial interface with the ChipWhisperer-Husky, and the core provides a trigger output used to time the glitches.

B. Software Faults

Fault attacks on software are done in a RISC-V soft-core implemented in the iCE40. This soft-core is the open-source NEORV32 core [47]. The core is configured with the 'small' configuration (`rv32i_Zicsr`), with 64 kB of code space ("ROM") & 64 kB of data space ("RAM"). All memory is actually implemented in the embedded block ram of the iCE40.

Software fault attacks are performed on a double loop calibration code. The glitches can also be tested against other code compiled for the core (such as MBED-TLS RSA or AES) using the provided framework, but in this work the calibration code is used.

The NEORV32 core is clocked at 25 MHz to match the same input clock rate of the TDC, which means the clock and glitch setup in the ChipWhisperer-Husky will be identical between the TDC and software glitch. Communication with the NEORV32 is handled over a serial interface with the ChipWhisperer-Husky, and the firmware provides a trigger output used for timing the glitches.

V. Body Biasing Injection (BBI) Setup

The open source BBI probe described in [43] is used as a comparative tool. To prepare the iCE40UP5K in WLCSP for BBI attacks, the optically protective cover of the WLCSP device was removed by mechanical means (scraping). As in [43] the resistance was measured from the exposed back die to the GND pin. On the iCE40UP5K in WLCSP it was found to be around 2.5 $M\Omega$, significantly more than the 100-400 $k\Omega$ range reported for the STM32F415OG device in [43].

No faults (not even resets) could be inserted using the BBI probe in this configuration, which was assumed to be related to the very high resistance that causes limited current flow. As a further test, the internal TDC & external VCC measurement method was used to check for any disturbance and no disturbance of the power supply was observed.

The backside of the WLCSP was lightly wet sanded with 600-grit sandpaper, and the resistance was measured again with a slightly thinner substrate. The resistance was now measured in the 500 $k\Omega$ range. The glitch insertion attempts also generated perturbations visible on both the TDC & external VCC measurement points. At the same time, testing with a simple software double loop resulted in successful glitches (as well as resets of the device). This early work suggested that using measurements of the internal power supply with either the TDC or external VCC power pin method provides very valuable feedback during fault injection setup.

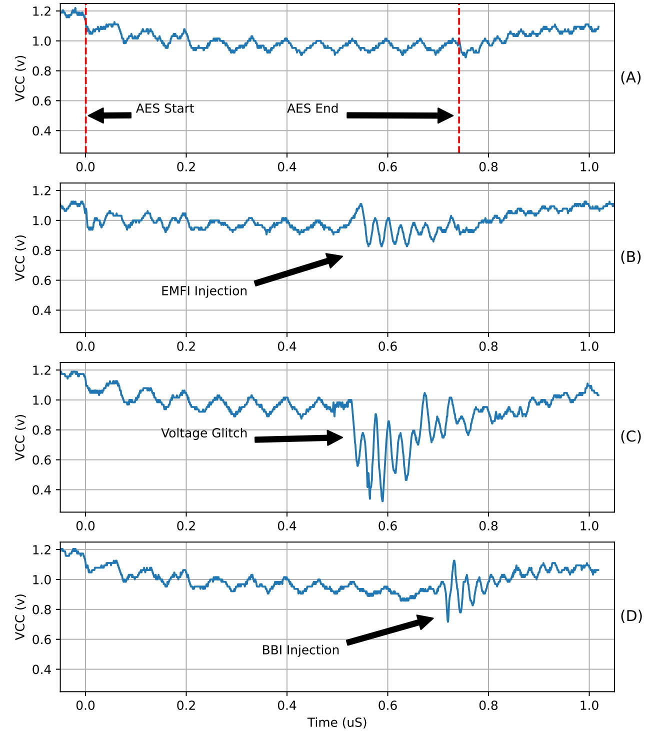

Figure 12: Measurements of the VCC-INT power rail using external oscilloscope during hardware AES operations.

VI. Results and Discussion

The objective of this work was to gain a better understanding of how different types of fault injection affect a target device, to understand the connection between these methods. In addition, it introduces a low-cost EMFI tool, and the tool must be characterized compared to other methods and tools. To this end, the following results allow us to:

- Understand the effectiveness of a low-cost EMFI tool (PicoEMP).
- Understand how the internal (TDC) measurements correspond to the external (VCC) measurements.
- Compare measurements across multiple types of fault injection methods.
- Compare effective & not effective glitches from the EMFI tool.

To start, a hardware AES core that is implemented on the reference platform (iCE40) is attacked with voltage, EMFI, and BBI fault injection. A similar test is then performed on

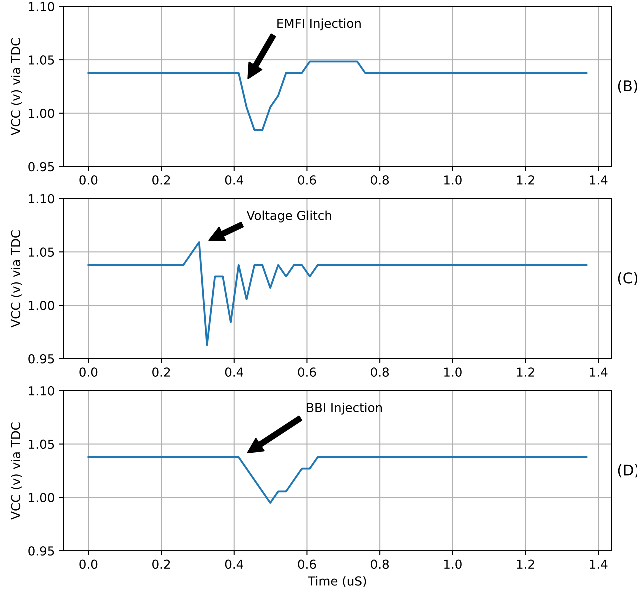

Figure 13: Measurements of the VCC-INT power rail using TDC during hardware AES operations.

a soft core RISC-V microcontroller (NEORV32), demonstrating the difference in glitch parameters required on the same physical device but running different circuitry. This soft-core platform is also used to explore if it is possible to visualize effective and ineffective glitches using external VCC measurement techniques.

A. Hardware AES Core

Faults on the hardware AES core described earlier were inserted with all three techniques (voltage, EMFI, BBI).

Figure 12 shows the measurement of the external power rail during fault insertions. In each case, the faults were adjusted to provide faulty operation without appearing to permanently impact the core. Because the core is implemented in an FPGA, faults which are too powerful are likely to cause configuration bit-flips.

The Figure 12-A shows the power rail *without* a fault inserted as a reference, a small variation can be seen due to the shunt present in the target board. It can be seen that each of the fault insertion methods have differing levels of disturbance visible on the power rail, however, they all produce visible waveforms on the VCC power rail.

For the voltage fault injection Figure 12-C, the measurement and injection are done with different connections on the CW312 base board (which provides two SMA connectors).

The corresponding measurements from the internal TDC measurements are shown in Figure 13. The waveforms show some similarity to the externally measured ones, but the ringing appears to be less pronounced.

The settings for each of the fault injection probes used to generate Figure 12 and Figure 13 are given in Table III.

Table III: Glitching Hardware AES Implementation

Type	Pulse Width	Probe Type	Voltage
EMFI (PicoEMP)	112 nS	1mm ferrite coil	V_{charge}=310V
Voltage ([44])	66.7 nS	SMA (CW312)	Crowbar on VCC (1.2V)
BBI ([43])	267 nS	Spring needle	V_{input}=20.0 V (see [43])

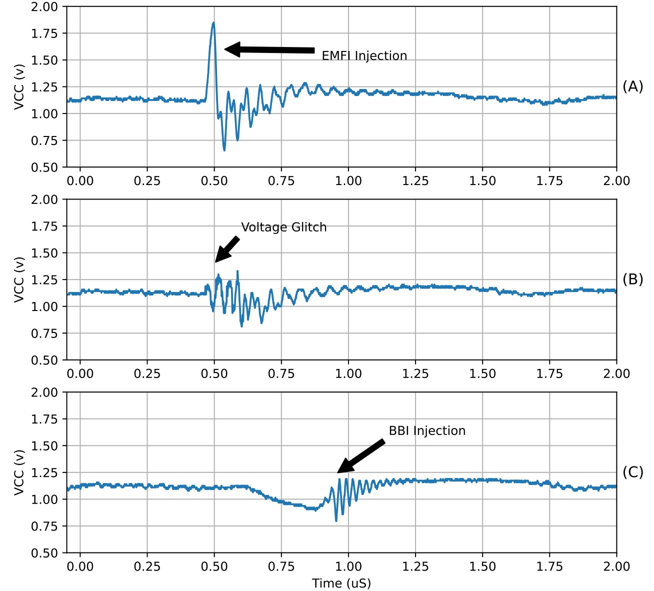

Figure 14: Measurements of the VCC-INT power rail using external oscilloscope during RISC-V soft-core operation.

B. Software Glitches

The glitches on the NEORV32 microcontroller core are again adjusted to provide faulty operation, but without causing a reset or FPGA configuration errors. The fault measurement for each of the injection methods is shown in Figure 14 using the external oscilloscope and Figure 15 using the internal TDC.

Successful glitches required longer pulse widths, as can be seen in the inserted glitch waveforms. Compared to the AES engine, the glitch waveforms for all methods are more pronounced. The reported timing closure for the NEORV32 core (22 MHz) is close to the AES engine, but we found that the NEORV32 core seemed more stable even in overclocking compared to the AES engine (we run the NEORV32 core at 25 MHz in these tests). The more stable design may explain why more powerful faults are required compared to the hardware AES core.

C. EMFI Pulse Width Effect

One of the parameters when working with EMFI is the width of the EMFI pulse. Because the EMFI coupling mechanism is related to the change of the magnetic field,

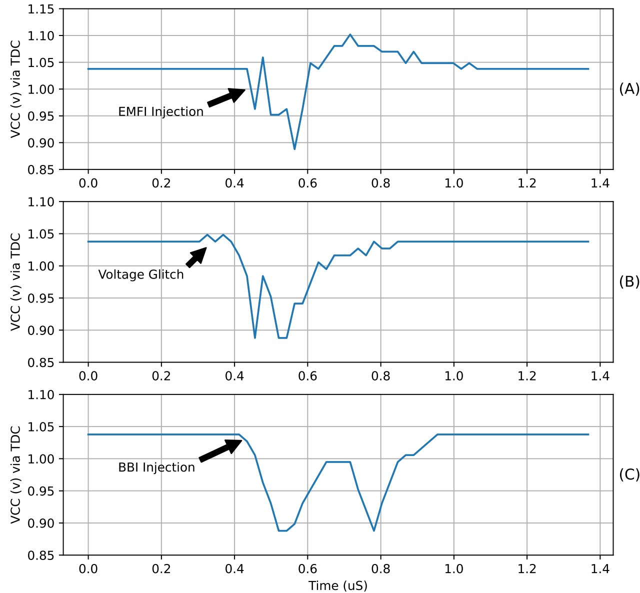

Figure 15: Measurements of the VCC-INT power rail using TDC during RISC-V soft-core operation.

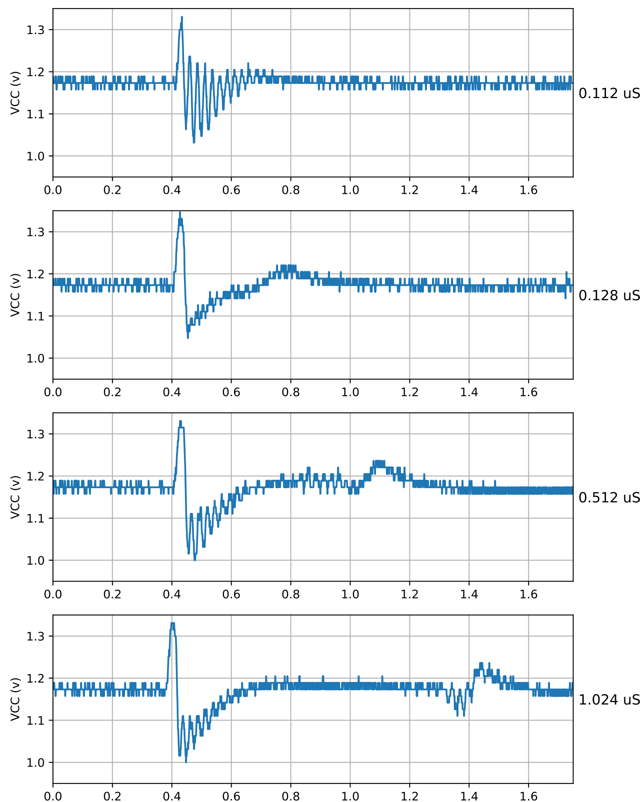

Figure 16: Comparison of EMFI Pulse Width, measured using oscilloscope on VCC externally.

we expect that a wider or narrower pulse does not result in a wider or narrower injected pulse in the target, but instead changes the delay between the *two* injected pulses: one for the rising edge, and one for the falling edge. This is explored with a detailed simulation in [48].

This can be confirmed using both external and internal VCC measurements, as shown in Figure 16 and Fig 17 respectively. Looking first at Fig 16, when there is a 1.024 μS delay (the bottom-most figure), the two edges are clearly visible at 0.4 μS and 1.4 μS. As the pulse width is shortened, the second rising edge moves towards the first edge. The narrowest pulse that had a visible effect on the power rail was the 0.112 μS width. It is notable that this very short width had additional ringing present, which suggests that changing the pulse width may have effects beyond just the change of the pulse location.

Looking at the form of the pulse generated by the PicoEMP tool in Figure 5, it is seen that there is a much sharper rising edge on the pulse (thus a faster rate of change of the magnetic field). This is also seen in the coupled waveforms, as the inserted pulse from the rising edge is much larger than that from the falling edge.

The internal measurements have a similar form, although noticeably the sharp rising edge present in the external measurements is missing. In addition, the ringing from the very short pulse was not visible in these TDC measurements. Due to the limited buffer size, the TDC does not capture the full waveform for the 1.024 μS delay, but the rising edge can be seen just at the end of the data capture.

D. *Effective & Ineffective EMFI Settings*

EMFI has many parameters to adjust, including both the spatial position and settings on the glitch generator itself. If a measurement of the external VCC power rail could provide useful information, this could potentially help speed up EMFI evaluations. For this work, the settings of the EMFI glitch generator (PicoEMP) were fixed, that is, the same charge voltage, pulse width, and delay from trigger were always used.

In the following example, the spatial position of the EMFI probe was varied with the objective of finding examples of an unaffected target, a successful glitch, and a reset of the target. As the target in this case is the soft-core NEORV32 processor, we expect it to be sensitive to “too powerful” glitches that corrupt the RAM-based configuration logic.

A record of the EMFI glitches as measured using an oscilloscope attached to the VCC rail is given in Figure 18. The middle subfigure of Figure 18, which has a “Normal” result, is also interesting, as the polarity of the inserted glitch appears opposite to the other inserted glitches. This was a result of only spatial changes in position, not a result of changing the probe type or injected waveform polarity.

This polarity inversion is another validation of the injection model and simulation from [48]. If we assume that the glitches are primarily coupled to loops formed by power

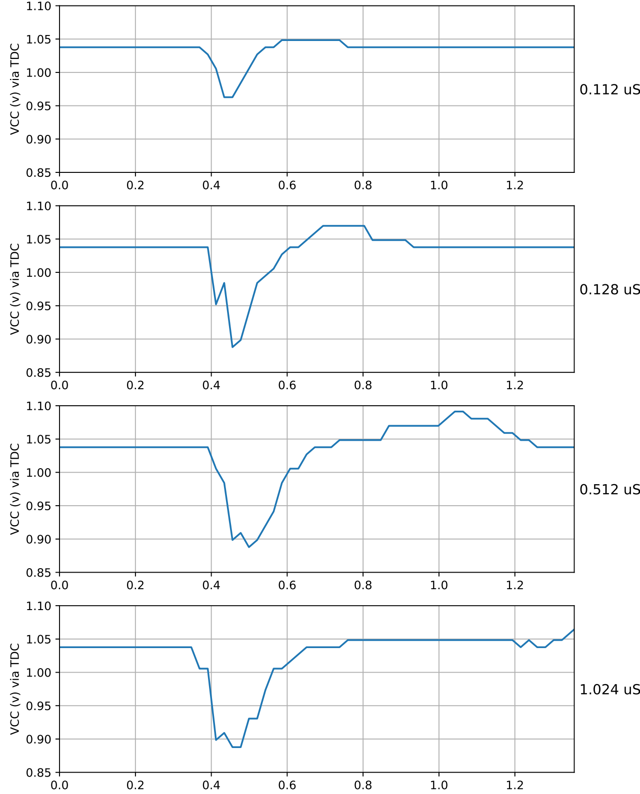

Figure 17: Comparison of EMFI Pulse Width, measured using TDC on VCC internally.

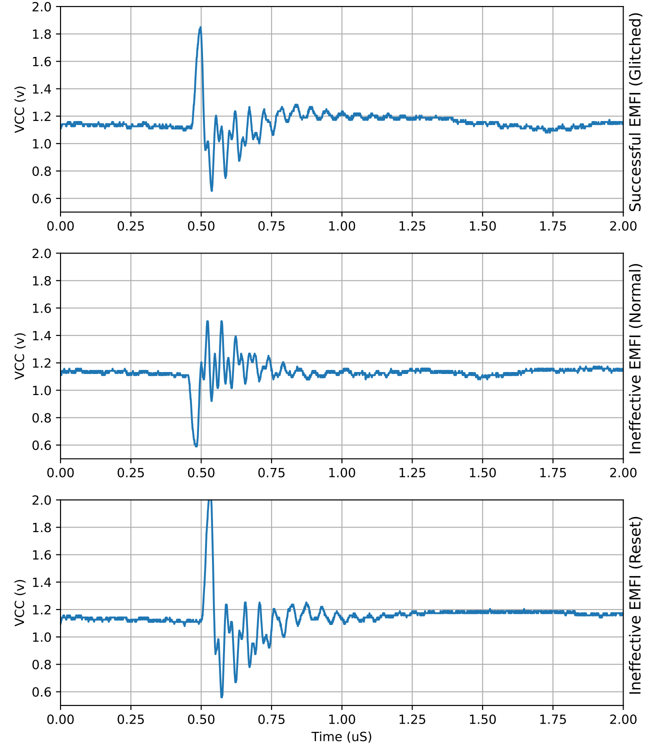

Figure 18: Comparison of EMFI pulses as measured on VCC-INT for effective & ineffective glitches.

rails, we could flip the polarity of the glitch by changing the position of the probe, which changes the effective "direction" of the loop relative to the VCC or GND rail. The polarity flip here demonstrates that this primary coupling mechanism exists as proposed.

On the question of effective or ineffective EMFI glitches, the result that device resets are generated by larger peak voltages is also shown in Figure 18, and the ineffective (normal) results were generated by the smallest peak values.

Using external measurements of the VCC rail can provide value insight into the actual coupled signal. The magnitude of this coupling varies with probe position, as again in this example, the only variation was the probe position and not the EMFI tool settings. In previous work, the scanning parameters are normally assumed to be XY location as one parameter, and glitch strength (normally in terms of EMFI voltage) as another. This suggests that there is a relationship between them, so by measuring the VCC rail it is possible to optimize the search space by determining if glitch strength should be increased (or reduced) at a given location.

This also explains another mechanism for a common result with EMFI: when scanning the surface, large sections of the chip are more sensitive and cause resets. This may simply be because of differences within the structure of the chip are causing larger coupled signals to the power rails (i.e., larger loops in the power rails).

VII. Conclusion

The use of a Time-to-Digital Converter (TDC) is very valuable for feedback on the fault injection process [19], [20]. This work has introduced a design of a TDC for the iCE40 device, and demonstrated how it can be used for measurement of voltage, BBI, and EMFI attacks. In practice many devices *won't* have a TDC present, so using the external measurement technique shown here may be very valuable.

For example, previous work measuring the results of BBI using different substrate types [49] demonstrated the need for such simple measurement techniques, showing that there are many areas of exploration still to be done.

The use of both the TDC & external measurement allow visualization of how glitches from voltage, EMFI, and BBI provide similar internal power rail perturbations. It also demonstrated how the polarity of the inserted glitch with EMFI can switch based only on the spatial positioning, without reversing the polarity of the injected pulse.

The low-cost EMFI tool introduced in this paper has also been validated in multiple scenarios, including a high-speed Arm processor (Raspberry Pi 3B+), a FPGA based AES hardware core, and a FPGA based RISC-V soft-core. This

demonstrates that even this low-cost EMFI tool can be used for a variety of academic and industrial work.

Future work can build on these open-source blocks to continue our understanding of how EMFI fault injection works and how to best protect our embedded systems against fault injection attacks.

All figures and data in this paper is available at https://github.com/colinoflynn/picoemp-tdc-paper.

ACKNOWLEDGMENT

As an open-source project, PicoEMP has additional contributed improvements not discussed in this paper. Thanks to those contributors, and special thanks is given to Thomas Roth for writing a C firmware version of the code, which *significantly* improves the simple MicroPython firmware described in this paper.

REFERENCES

[1] D. Boneh, R. A. DeMillo, and R. J. Lipton, "On the Importance of Checking Cryptographic Protocols for Faults," in *Advances in Cryptology — EUROCRYPT '97*, W. Fumy, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 37–51.

[2] D. Oswald, "Development of an integrated environment for side channel analysis and fault injection," Ph.D. dissertation, Ruhr-Universität Bochum, Bochum, Germany, 2009.

[3] A. Beckers, S. Guilley, P. Maurine, C. O'Flynn, and S. Picek, "(adversarial) electromagnetic disturbance in the industry," *IEEE Transactions on Computers*, vol. 72, no. 2, pp. 414–422, 2023.

[4] C. Fanjas, C. Gaine, D. Aboulkassimi, S. Pontié, and O. Potin, "Combined fault injection and real-time side-channel analysis for android secure-boot bypassing," in *Smart Card Research and Advanced Applications Conference CARDIS*. Berlin, Heidelberg: Springer-Verlag, 2023, p. 25–44.

[5] N. Kühnapfel, R. Buhren, H. N. Jacob, T. Krachenfels, C. Werling, and J.-P. Seifert, "Em-fault it yourself: Building a replicable emfi setup for desktop and server hardware," in *2022 IEEE Physical Assurance and Inspection of Electronics (PAINE)*, 2022, pp. 1–7.

[6] N. Wiersma and R. Pareja, "Safety != security: On the resilience of asil-d certified microcontrollers against fault injection attacks," in *2017 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, 2017, pp. 9–16.

[7] C. O'Flynn, "BAM bam!! on reliability of EMFI for in-situ automotive ECU attacks," *IACR Cryptol. ePrint Arch.*, p. 937, 2020. [Online]. Available: https://eprint.iacr.org/2020/937

[8] H. Lim, T.-H. Lee, S. Lim, J. Han, B.-Y. Sim, and D.-G. Han, "Fault injection method for hardware-implemented aes without artificial triggering," in *Proceedings of the 2020 ACM ICEA*, ser. ACM ICEA '20. New York, NY, USA: Association for Computing Machinery, 2021.

[9] N. Timmers and C. Mune, "Escalating privileges in linux using voltage fault injection," in *2017 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, 2017, pp. 1–8.

[10] I. Polian, "Fault attacks on cryptographic circuits," in *2019 17th IEEE International New Circuits and Systems Conference (NEWCAS)*, 2019, pp. 1–4.

[11] C. Giraud, "Dfa on aes," in *Advanced Encryption Standard – AES*, H. Dobbertin, V. Rijmen, and A. Sowa, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 27–41.

[12] A. Barenghi, G. M. Bertoni, L. Breveglieri, G. Pelosi, S. Sanfilippo, and R. Susella, "A fault-based secret key retrieval method for ecdsa: Analysis and countermeasure," *J. Emerg. Technol. Comput. Syst.*, vol. 13, no. 1, apr 2016.

[13] R. Singh, S. Islam, B. Sunar, and P. Schaumont, "An end-to-end analysis of emfi on bit-sliced post-quantum implementations," 2022.

[14] N. Bagheri, S. Sadeghi, P. Ravi, S. Bhasin, and H. Soleimany, "SIPFA: Statistical Ineffective Persistent Faults Analysis on Feistel Ciphers," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2022, no. 3, p. 367–390, Jun. 2022.

[15] P. Ravi, B. Yang, S. Bhasin, F. Zhang, and A. Chattopadhyay, "Fiddling the twiddle constants - fault injection analysis of the number theoretic transform," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2023, no. 2, p. 447–481, Mar. 2023.

[16] I. Polian and F. Regazzoni, "Counteracting malicious faults in cryptographic circuits," in *2017 22nd IEEE European Test Symposium (ETS)*, 2017, pp. 1–10.

[17] W. He, J. Breier, S. Bhasin, N. Miura, and M. Nagata, "An fpga-compatible pll-based sensor against fault injection attack," in *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2017, pp. 39–40.

[18] J. Breier, S. Bhasin, and W. He, "An electromagnetic fault injection sensor using hogge phase-detector," in *2017 18th International Symposium on Quality Electronic Design (ISQED)*, 2017, pp. 307–312.

[19] L. Zussa, J.-M. Dutertre, J. Clediere, and B. Robisson, "Analysis of the fault injection mechanism related to negative and positive power supply glitches using an on-chip voltmeter," in *2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, 2014, pp. 130–135.

[20] M. Paquette, B. Marquis, R. Bainbridge, and J. Chapman, "Visualizing electromagnetic fault injection with timing sensors," in *2021 IEEE Physical Assurance and Inspection of Electronics (PAINE)*, 2021, pp. 1–8.

[21] S. Endo, T. Sugawara, N. Homma, T. Aoki, and A. Satoh, "An on-chip glitchy-clock generator for testing fault injection attacks," *Journal of Cryptographic Engineering*, vol. 1, pp. 265–270, 2011.

[22] C. Wolf, "Yosys open synthesis suite," https://yosyshq.net/yosys/.

[23] iFixit. (2016) iPhone 7 Teardown. [Online]. Available: https://www.ifixit.com/Teardown/iPhone+7+Teardown/67382

[24] ——. (2014) Samsung Galaxy S5 Teardown. [Online]. Available: https://www.ifixit.com/Teardown/Samsung+Galaxy+S5+Teardown/24016

[25] C. O'Flynn. (2020) Square Terminal Teardown. [Online]. Available: https://colinoflynn.com/2020/04/square-terminal-teardown/

[26] M. Dumont, M. Lisart, and P. Maurine, "Electromagnetic Fault Injection : How Faults Occur," in *2019 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. IEEE Computer Society, 2019, pp. 9–16.

[27] A. Menu, J.-M. Dutertre, O. Potin, J.-B. Rigaud, and J.-L. Danger, "Experimental analysis of the electromagnetic instruction skip fault model," in *2020 15th Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*, 2020, pp. 1–7.

[28] A. Dehbaoui, J.-M. Dutertre, B. Robisson, and A. Tria, "Electromagnetic Transient Faults Injection on a Hardware and a Software Implementations of AES," in *2012 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. IEEE Computer Society, 2012, p. 7–15.

[29] M. Ghodrati, B. Yuce, S. Gujar, C. Deshpande, L. Nazhandali, and P. Schaumont, "Inducing local timing fault through em injection," in *Proceedings of the 55th Annual Design Automation Conference*, ser. DAC '18. New York, NY, USA: Association for Computing Machinery, 2018.

[30] A. Menu, S. Bhasin, J.-M. Dutertre, J.-B. Rigaud, and J.-L. Danger, "Precise spatio-temporal electromagnetic fault injections on data transfers," in *2019 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, 2019.

[31] A. Cui and R. Housley, "BADFET: Defeating modern secure boot using second-order pulsed electromagnetic fault injection," in *11th USENIX Workshop on Offensive Technologies (WOOT 17)*. Vancouver, BC: USENIX Association, 2017, p. 12.

[32] J. Balasch, D. Arumí, and S. Manich, "Design and validation of a platform for electromagnetic fault injection," in *2017 32nd Conference on Design of Circuits and Integrated Systems (DCIS)*, 2017, pp. 1–6.

[33] K. M. Abdellatif and O. Hériveaux, "SiliconToaster: A Cheap and Programmable EM Injector for Extracting Secrets," in *2020 Workshop on Fault Detection and Tolerance in Cryptography (FDTC)*. IEEE Computer Society, 2020, pp. 35–40.

[34] A. Beckers, M. Kinugawa, Y. Hayashi, J. Balasch, and I. Verbauwhede, "Design and evaluation of a spark gap based em-fault injection setup," in *"2020 IEEE International Symposium on Electromagnetic Compatibility & Signal/Power Integrity (EMCSI)"*, 2020, pp. 523–526.

[35] A. Beckers, M. Kinugawa, Y. Hayashi, D. Fujimoto, J. Balasch, B. Gierlichs, and I. Verbauwhede, "Design Considerations for EM Pulse Fault Injection," in *Smart Card Research and Advanced Applications Conference, CARDIS*, S. Belaïd and T. Güneysu, Eds. Springer, 2019, pp. 176–192.

[36] P. Maurine, K. Tobich, T. Ordas, and P. Y. Liardet, "Yet Another Fault Injection Technique : by Forward Body Biasing Injection," in *YACC'2012: Yet Another Conference on Cryptography*, Porquerolles Island, France, Sep. 2012. [Online]. Available: https://hal-lirmm.ccsd.cnrs.fr/lirmm-00762035

[37] K. M. Zick, M. Srivastav, W. Zhang, and M. French, "Sensing nanosecond-scale voltage attacks and natural transients in fpgas," in *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, ser. FPGA '13, New York, NY, USA, 2013, p. 101–104.

[38] J. van Woudenberg and C. O'Flynn, *The Hardware Hacking Handbook*. No Starch Press, 2021.

[39] Texas Instruments, "SLUA618A: Fundamentals of MOSFET and IGBT Gate Driver Circuits," https://www.ti.com/lit/ml/slua618a/slua618a.pdf, 2017.

[40] T. Trouchkine, G. Bouffard, and J. Clédière, "Em fault model characterization on socs: From different architectures to the same fault model," in *2021 Workshop on Fault Detection and Tolerance in Cryptography (FDTC)*, 2021, pp. 31–38.

[41] Harald Homulle and Edoardo Charbon. (2015) FPGA designs for reconfigurable converters. [Online]. Available: https://sps.ewi.tudelft.nl/fpga_tdc/TDC_basic.html

[42] D. R. E. Gnad, S. Rapp, J. Krautter, and M. B. Tahoori, "Checking for electrical level security threats in bitstreams for multi-tenant fpgas," in *2018 International Conference on Field-Programmable Technology (FPT)*, 2018, pp. 286–289.

[43] C. O'Flynn, "Low-cost body biasing injection (BBI) attacks on WLCSP devices," in *Smart Card Research and Advanced Applications Conference, CARDIS*, ser. Lecture Notes in Computer Science, P. Liardet and N. Mentens, Eds., vol. 12609. Springer, 2020, pp. 166–180.

[44] ——, "Fault injection using crowbars on embedded systems," Cryptology ePrint Archive, Paper 2016/810, 2016, https://eprint.iacr.org/2016/810.

[45] P. Dusart, G. Letourneux, and O. Vivolo, "Differential fault analysis on a.e.s." Cryptology ePrint Archive, Paper 2003/010, 2003, https://eprint.iacr.org/2003/010. [Online]. Available: https://eprint.iacr.org/2003/010

[46] A. Moradi, M. T. M. Shalmani, and M. Salmasizadeh, "A generalized method of differential fault attack against aes cryptosystem," in *Cryptographic Hardware and Embedded Systems (CHES): 8th International Workshop, Yokohama, Japan, October 10-13, 2006*. Springer, 2006, pp. 91–100.

[47] S. Nolting and ..., "The neorv32 risc-v processor," https://github.com/stnolting/neorv32, 2022.

[48] M. Dumont, M. Lisart, and P. Maurine, "Modeling and simulating electromagnetic fault injection," *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, vol. 40, no. 4, pp. 680–693, 2021.

[49] G. Chancel, J.-M. Gallière, and P. Maurine, " Body Biasing Injection: Impact of substrate types on the induced disturbances?" in *FDTC*, September 16 2022, Virtual workshop, IEEE.