

# Phase Modulation Side Channels: Jittery JTAG for On-Chip Voltage Measurements

Colin O’Flynn<sup>1</sup>

Dalhousie University, Halifax, Canada, [colin@oflynn.com](mailto:colin@oflynn.com)

## Abstract.

Measuring the fluctuations of the clock phase of a target was identified as a leakage source on early electromagnetic side-channel investigations. Despite this, only recently was directly measuring the clock phase (or jitter) of digital signals from a target connected to being a source of exploitable leakage. As the phase of a clock output will be related to signal propagation delay through the target, and this propagation delay is related to voltage, this means that most digital devices perform an unintended phase modulation (PM) of their internal voltage onto clock output phases.

This paper first demonstrates an unprofiled CPA attack against a Cortex-M microcontroller using the phase of a clock output, observing the signal on both optically isolated and capacitively isolated paths. The unprofiled attack takes only 2–4× more traces than an attack using a classic shunt-resistor measurement.

It is then demonstrated how the JTAG bypass mode can be used to force a clock through a digital device. This forced clock signal can then be used as a highly effective oscilloscope that is located on the target device. As the attack does not require modifications to the device (such as capacitor removal or heat spreader removal) it is difficult to detect using existing countermeasures. The example attack over JTAG uses an unprofiled CPA attack, requiring only about 5× more traces than an ideal shunt-resistor based measurement. In addition, a version of this attack using a fault correlation analysis attack is also demonstrated.

Countermeasures are discussed, and a simple resampling countermeasure is tested. All tools both offensive and defensive presented in the paper have been released under open-source licenses.

**Keywords:** power analysis · phase modulation · remote power analysis · JTAG

## 1 Introduction

The introduction of differential power analysis [KJJ99] began a rich history of attacking embedded devices by using physical measurements of power, either directly with a shunt resistor or via a proxy measurement. Proxies have included for example EM measurements [GMO01, AARR03], long-range radio measurements [CPM<sup>+</sup>18], optical emissions [SNK<sup>+</sup>12], I/O pin leakage [SPK<sup>+</sup>10], remote power analysis using shared resources between targets [ZS18, SGMT18a], and even acoustic sound recorded from the decoupling capacitors on a device [GST14]. One of the recent proxies has been the jitter of a clock signal coming from a target, called JitSCA [SMTG23].

The implications of the JitSCA work is that a digital device processing a clock signal inherently imparts small amounts of jitter related to the power consumption, and measuring a ‘jitter trace’ is a direct proxy for a ‘power trace’. Because the timing information will be encoded on communication interfaces, this opens up methods of attacking devices where classic shunt-based or EM-based power analysis seemed physically impossible.

This paper will use the idea of data-leaking delays when specifically referring to signal delays which depend on data being processed *elsewhere*. That is the delay does not depend on the value of the data being transferred in the signal, but instead the data being transferred is used to generate clock edges which have a measurable phase shift (or jitter) from a reference clock.

To avoid recreating new domain-specific acronyms, we will generally refer to the delay information as a *Phase Modulation*(PM) encoding or our leakage. To maintain the link to well-known radio definitions, the signal which carries this PM will be the *carrier*. The author stresses that the objective of this naming is not to “invent” a new leakage, but to use a more generic name for a physical phenomenon that has been exploited in several prior papers under different names. The author also notes that the mention of *phase modulation* as a leakage source was identified in early papers on side-channel emissions, including at least from CHES 2002 for example which calls it *angle modulation* (referring to the vector representation typical in radio receivers) [AARR03].

## 1.1 Threat Model

Phase Modulation (PM) leakage represents an increased threat compared to classic shunt or EM measurements, as it means purely digital interfaces that carry only timing information can be used to leak side-channel power information. In this work, we will first demonstrate how the leakage can transfer across two types of isolators, and then demonstrate how the ubiquitous JTAG interface can be used to acquire “power traces”. These traces do not require advanced analysis methods, a classic unprofiled CPA attack on the traces recorded from the JTAG port successfully recovers the encryption key in 8900 traces (compared to 1800 for a shunt resistor with the same platform).

As JTAG is an almost universal interface on digital systems, the demonstration of the capability of JTAG to serve as a sensor for recording power traces is particularly powerful in practical scenarios. Work to detect an attacker bringing an EM probe near the device [MFT<sup>+</sup>14, HiHM<sup>+</sup>14], or detect an attacker modifying the impedance of the power supply network [GKDG20], should not be triggered if only the JTAG interface is being used (as this interface would be used during regular debug or test operations).

Many other high-speed digital interfaces also are likely to provide useful interface. Many embedded devices boot from a QSPI or eMMC device, and an attacker who has access to those pins can use the clock signals to generate reliable side-channel measurements. Externally accessible interfaces, such as a microSD card, are of course especially interesting as they allow one to “reach inside” the device. While these protocols may not have constant clocks, an attacker can take advantage of the protocols to force more frequent transactions. The host (target) will poll the microSD card until a busy bit is clear, and an attacker can keep this bit set for long stretches of time to force higher than normal bus activity.

Industry frameworks for ranking attacks, such as found in the Joint Interpretation Library (JIL) or Common Criteria references, makes an important distinction between attacks which require considerable technique expertise to apply them, and those that can be “commoditized” [SOG24]. From a practical standpoint, these PM-based measurements have the advantage of being highly repeatable and reliable. While the *identification* phase of an attack requires a high level of skill (and possibly equipment), the *exploitation* phase can have very low cost and skill. The entire measurement can be done with digital devices, such as a FPGA. This differs from many classic attack measurement techniques: for example using an EM probe is also non-invasive, but the requirement of carefully positioning the probe means there is still a high level of exploitation equipment cost and expertise required. An attack which requires only plugging a device into the JTAG port, and possibly adding some wires for communication or triggering, is in-line with the user experience for existing commoditized attack devices such as game console ‘mod chips’ or automotive ECU ‘tuning tools’.

## 1.2 Contributions

This paper has two major parts: first, we’ll work to validate and extend the work of JitSCA [SMTG23], which was the first paper to take the work of using TDCs for on-chip power measurements [SGMT18a] and instead measure the data leaking delays through an external interface.

Section 3 will first extend the [SMTG23] work in several important ways: (1) demonstrate the use of RF mixers for phase measurement, with a successful unprofiled CPA attack taking only 2775 traces over a 10m optical link, (2) validating the leakage with low-cost Cortex-M microcontrollers, (3) validate the leakage across several isolator technologies, and finally (4) link several prior results to this leakage.

The second set of contributions is focused around the application of this leakage. Knowing that signals processed by a device have an added phase contribution, we then focus on the common JTAG port. The JTAG port has been well-studied previously, but the author is not aware of any use of the port for obtaining side-channel power measurements. This paper will introduce for the first time the following attacks on a system where an attacker is *only accessing the digital JTAG port* (no analog probes): in Section 4 a successful unprofiled CPA attack taking 8900 traces; and in Section 5 an unprofiled fault sensitivity analysis attack taking 41500 traces.

In addition, a detailed analysis of the capabilities of the JTAG ports on several devices is covered in Section 4.2. This will demonstrate that typical devices have much higher speed JTAG speeds capable than datasheet specifications suggest. It will also demonstrate how some commercial devices have the ability to fully disable the JTAG port, forming an effective countermeasure to this attack.

Finally, we have released under open-source licenses extensions to existing open-source tools, as well as new designs of tools for measurement, as well as released all the code used in this paper. The objective is to allow other researchers to immediately recreate and extend the results in this paper. These are available at <https://github.com/colinoflynn/phase-modulation-sca>.

These tools are useful beyond side-channel measurements as well. The “remote voltage sensor” one can build with JTAG can be used to simply validate power supply stability *within* the target device for example, or using the sensor as part of a verification that a design has not been tampered with.

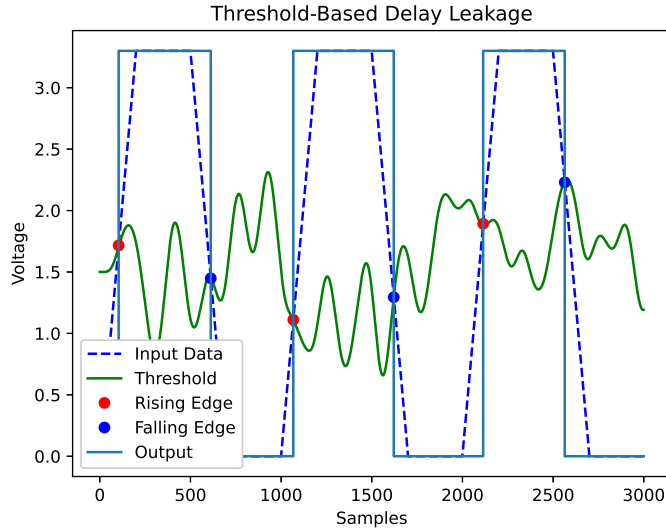
## 2 Background

We have attempted to connect the leakage described in this paper to many previous results, to demonstrate that there is already a large body of work we can draw on, and should temper our view that this is a “new” leakage source. In fact the core idea of phase modulation leakage is described in some of the earliest work on EM leakage [AARR03]. Much of the more recent relevant work falls into the category of remote power analysis (RPA), including use of attacking across FPGA tenants [SGMT18a], across chips on a PCB [SGMT18b], and RF measurement [CPM<sup>+</sup>18]. This connection to previous work also includes countermeasures that can be directly applied to this leakage source.

We’ll briefly discuss *why* digital devices have this voltage-sensitive delay, before going into more detail on the relevant existing work across several related domains.

### 2.1 Sources of Delay

Fundamentally, the delay through a digital device depends on several factors. The most commonly considered are temperature and voltage. Digital devices will use different constructions of internal logic, but for our purpose we can consider a simple inverter chain which has two MOSFETs used to build a buffer. For all of these constructions, there will



**Figure 1:** Changes in threshold levels will also result in output jitter.

be some *threshold voltage* and *rise and fall times*, which are all affected by changes in voltage [GSH<sup>+</sup>15].

A purely internal buffer may depend on a core voltage supply, but an I/O buffer may also include the effect of additional voltage rails. The signal at the input of a buffer will have some non-zero rise and fall time, which means that the time when the input data passes the threshold voltage for the buffer will slightly vary with changes in the VCC.

This is shown in Figure 1. The slower rising and falling edge (dashed line) gets converted to the square wave output (solid line), and the resulting square wave has differing  $t_{on}$  and  $t_{off}$ . The threshold (green line) on each of the edges is a ratio of the VCC of the device, which is varying due to the effect of varying power consumption and the fact the power distribution network internal and external to an IC cannot have zero impedance, so there is always some voltage drop with changing power consumption.

This specific example is brought up to emphasize that the jitter is not only from *internal logic*, but even the act of taking a signal into and out of a digital device will add jitter that depends on the power supply.

## 2.2 Prior Work

While the core of this paper is on leakage resulting from voltage-dependant delays as demonstrated in [SMTG23], previous work on remote power analysis [ZS18, SGMT18a], and physically connected I/O pin leakage [SPK<sup>+</sup>10] is all closely connected. We will summarize the important prior work in those specific areas, before also summarizing some of the prior work looking at attacks over the JTAG interface.

### 2.2.1 TDCs and On-Chip Measurement

Time to Digital Converters (TDCs) convert a measurement of the time between two signals (such as delay, phase, or jitter measurements) into a digital value [RAB10]. Typically they can be implemented with ring oscillators or delay lines (typically carry chains). TDCs are well suited to FPGA implementation, with a large body of work discussing implementation in various devices and with resolutions in off-the-shelf FPGA platforms down to 1.8 pS [RAB10, Sve20].

The use of TDCs specifically for side-channel attacks was first presented in [SGMT18a], with other work suggesting their use for defensively detecting attacks such as voltage glitching [ZSZF13], along with related remote analysis work using ring oscillators instead [ZS18]. The work on TDCs for side-channel measurement underpins our own work here: the usage of TDCs for side-channel measurements is possible because voltage fluctuations on a device lead to changes in propagation delay. The same general structure can be used for two different purposes: in [SGMT18a] the objective is to measure the changes in propagation delay within the TDC, so measurements are done relative to a constant reference clock or trigger signal. In our work the goal is to measure changes in the clock or trigger signal, which would require keeping the propagation delay within the TDC constant, as in [SMTG23].

Prior work has also looked at usage of TDCs to measure voltage fluctuations on-board but not on-chip, for example [SGMT18b] demonstrated a successful attack where a TDC runs in a separate FPGA that shares a power supply with the target. In these cases the TDC is still used as a sensor for propagation delay changes due to voltage, and not to measure a changing clock phase.

Simple TDCs sample at a single clock edge, but delay information is encoded on both edges of the clock. A dual-edge TDC used for side channel is presented in [DWR<sup>+</sup>23], which is an open-source design. A very complete summary of TDC designs for side-channel attacks is available in [Lyt24], with various constructions of TDCs in FPGAs discussed in [MCA19].

### 2.2.2 Direct Jitter-based SCA

Despite the broad body of work on TDCs and their usefulness for side-channel analysis, it was more recently in [SMTG23] that the idea of measuring the phase shift (or jitter) *off-chip* was presented. It is [SMTG23] that is the most direct link to the work presented in this paper.

As will be demonstrated in this paper, while [SMTG23] was the first time that the external measurement of jitter was *explicitly* called out as the source of leakage, many other papers appear to have been measuring the effects. Changes in the clock (jitter) will become a change in phase and thus frequency of a measured signal, in particular this has previously been part of papers including those which measured RF leakage remotely [CPM<sup>+</sup>18], and also some of the earliest EM work explicitly mentioned phase modulation as a leakage source [AARR03]. A chapter on unintended side channels in [AZ23] also includes a discussion of phase modulation, similar to [CPM<sup>+</sup>18] looking at frequency shifts rather than a full phase modulation decoder.

### 2.2.3 Fault Sensitivity Analysis and Correlation Analysis

The ability of fault attacks to be connected to data leakage is well-known, having been initially presented in [LSG<sup>+</sup>10, MMP<sup>+</sup>11]. More specific results linking fault sensitivity to a power trace have been presented, in particular [LED<sup>+</sup>13] directly demonstrates the link between fault sensitivity traces and power analysis traces. More recently a very complete summary of these attacks was presented in [SMC21]. The work in [SMC21] also takes the usage of fault sensitivity analysis and demonstrates how non-profiled attacks such as correlation power analysis can be applied on the fault results. We will use a similar target and attack to that used in [SMC21] in Section 5.

Other domain-specific leakage such as for example using RowHammer to measure power [CTH<sup>+</sup>22] can be seen as another application that is resulting from a data leaking delay.

#### 2.2.4 Connected Leakage Measurements

Side-channel measurements which are based on physical connections to the target device, but without using “normal” side-channel probes (such as EM probes or shunt resistors) have been covered in literature previously. The consideration that side-channel leakage may not require an explicit measurement probe goes back to some of the earliest work in side-channel power analysis [Sha00], with the most relevant work for us in [SPK<sup>+</sup>10] which demonstrates a similar attack on AES where fluctuations in I/O pin voltages results in an exploitable measurement.

Closely related includes attacks against asymmetric encryption [GPT14], and attacks against AES where the measurement is done from the primary side of a switch-mode power supply [SLT16]. This demonstrates that leakage has often been measured at very small levels that ultimately results in successful attacks.

#### 2.2.5 RF Leakage

A final area of related leakage measurement work is that using longer-range electromagnetic or radio techniques. This work is relevant as it often uses a radio decoding technique, rather than a classic oscilloscope-only (baseband) sampling technique. For example in [GPPT15] the frequency shift is explicitly shown as part of the reception technique. More recently in [CPM<sup>+</sup>18] a remote attack on a software AES implementation as shown, which will closely match the sort of targets discussed in this work.

In [CPM<sup>+</sup>18] the leakage source is also explained as a leakage signal being modulated in both frequency and amplitude onto a carrier. In [CPM<sup>+</sup>18] this carrier is from a subsystem designed to transmit RF, but many of the points raised by [CPM<sup>+</sup>18] will follow for our system where the carrier is a normal digital clock signal.

#### 2.2.6 JTAG Attacks

The final type of prior work to be discussed is not a side-channel attack at all, but to discuss the general usage of the JTAG port for attacking embedded systems. Almost all embedded devices (microcontrollers, FPGAs, ASICs) include a JTAG interface, as it is used for factory test, debugging, programming, and circuit board testing. Due to its ubiquitous presence the JTAG interface is an obvious attack vector and has been well studied as a threat to embedded systems [RK10, MGB16]. Work on defenses typically involves detecting unusual commands or usage, or detecting the device suddenly entering an unexpected state [RK10, MGB16, RTBT19].

Work on scan-based attacks exists, which use the scan chain to either directly access registers of interest, or use the scan chain for loading test patterns and observing the results [LTPP07]. These attacks may be used for a form of side-channel attacks, but ultimately are using the JTAG port with valid timing and commands. These scan-based attacks are exploiting poor security segmentation or design, rather than it being a fundamental flaw across any device with a JTAG port.

Despite this long history of JTAG security threat analysis, we have not found any prior discussion of even the *possibility* of the JTAG port being used as a side-channel power measurement source. This demonstrates that new and powerful side-channel measurement techniques may still be lurking in our embedded systems.

### 2.3 Side-Channel Result Format

We will primarily be using two metrics for discussing side-channel leakage: Test-Vector Leakage Assessment (TVLA) which uses a Welch’s t-test result [GJJR11], and a standard correlation power analysis (CPA) attack using a Hamming weight (HW) leakage model based on the S-Box output [BCO04].

Typically TVLA results are considered to show leakage if an absolute value beyond 4.5 occurs in the middle third of the AES operation [GJJR11]. Where we are testing results across different device clock frequencies and sample rates we slightly relax this to instead concentrate on any points within at least the first 0.1 – 1.5 rounds of AES, but with most sample rates including later rounds (but never the load or unload operations).

All of the TVLA results are taken with 20 000 encryptions, which are randomly split during capture into operations with a fixed or operations with a random plaintext, with keys and plaintexts defined in [GJJR11] for the fixed group. The scripts used for capturing these results are available in the companion repository for verification.

Because the t-test does not guarantee *exploitable* leakage, we also perform a standard CPA attack [BCO04].

Additional metrics, such as Partial Guessing Entropy (PGE) vs. traces can be generated from the datasets and scripts that have been made available with this paper.

### 3 Phase Modulation (PM) Leakage Validation

In this section, we will first validate the work of [SMTG23] and demonstrate that a clock signal coming from a microcontroller has encoded within the jitter of the clock signal the same information that would leak over a power trace. We’ll first introduce a simple method of measuring this using a mixer device, and validate the leakage can be seen through two different types of isolator technologies used to remove all other paths that analog signals could be leaking.

This will form the basis for the attacks presented in Section 4 (power analysis using phase measurements of JTAG signals) and Section 5 (fault sensitivity analysis using JTAG).

#### 3.1 Mixer for Phase Measurement

Measuring the phase between two signals is a common problem in RF circuits. One well-known technique is to use a *mixer*, which is typically thought of a device that mixes two sine waves of frequency  $f_1$  and  $f_2$  to generate an output at  $f_1 + f_2$  and  $f_1 - f_2$ . This is used as part of upconverters or downconverters for example, by filtering away one of the undesired signals.

Using them when feeding the same input signal, but with different phases, can result in an output related to the phase between the two signals [Kur78]. In this work, we will use the Mini-Circuits ADE-1+ mixer to compare different signals. Compared to using an “absolute” delay measurement circuit, this has the advantage that we can measure phase changes relative to a reference signal.

We will refer to the signals to our mixer as the Local Oscillator (LO) and Radio Frequency (RF) signals. The LO port is fed our reference signal, and the RF port is fed the signal we wish to measure the phase of.

This technique means a low-jitter source clock is not needed, since we are primarily measuring *added* jitter, and not absolute jitter. In particular, when we discuss measuring PM through the JTAG interface in Section 4 the mixer is an ideal solution for our setup as allows us to change between shunt-based and PM-based measurements with only a minor addition.

For side-channel analysis, the idea of using phase modulation (or ‘angle modulation’, referring to the use of the vector angle) was also explored in early results [AARR03]. These previous results identified the phase leakage, but typically appeared to primarily be using this to break countermeasures or implementations, and did not fully explore the connection between shunt-based power traces and phase modulation.

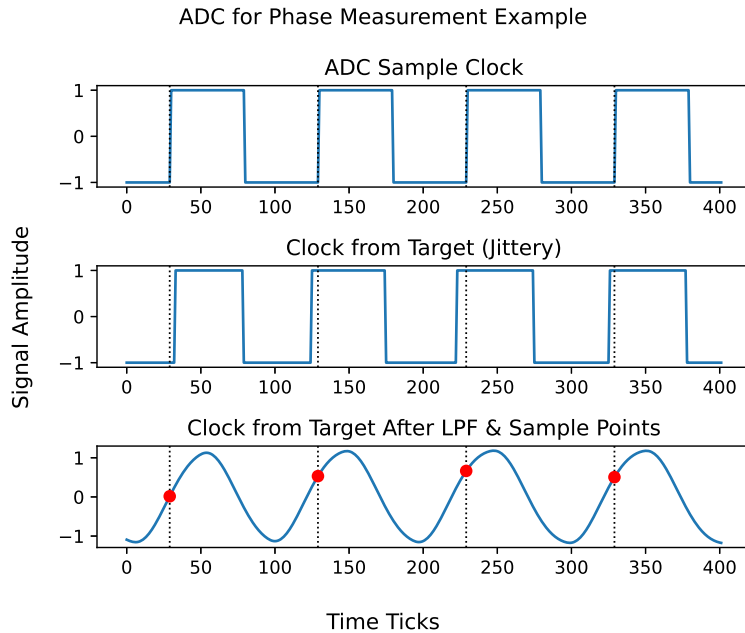
The focus on mixer-based (instead of TDC-based) solutions in this paper also makes the link between phase modulation decoders [AARR03] and the JitSCA [SMTG23] concrete.

Future work exploring EM or RF measurements can make use of mixers to more directly sample signals which have been processed by other blocks such as amplifiers, filters, or down-converters.

### 3.1.1 Sampling-Based Mixers

While the majority of this work uses a physical mixer device, it should be noted that the synchronous sampling technique itself can be used as a form of a mixer. This works because the ChipWhisperer-Husky slightly cleans up the clock coming from the device, and if this cleaned-up clock is used for sampling the original source clock, small variations in the source clock will become larger variations in voltage due to the shifting sample points relative to the jittery clock. This is shown in Figure 2.

To perform this sampling, only a voltage divider is required to connect the clock input to the ADC input. The voltage divider is required to reduce the logic level signal to a small-scale signal suitable for the ChipWhisperer analog input, and the resistance of the voltage divider with the input capacitance of the ChipWhisperer results in a low pass filter (LPF).



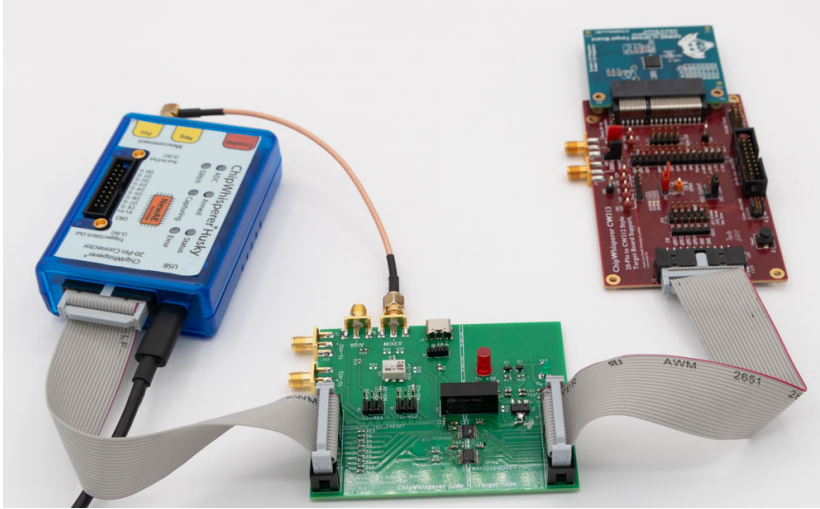
**Figure 2:** The ADC sample clock is a cleaned up version of the jittery target clock. Sampling the jittery clock after a low pass filter (LPF) results in samples related to the phase difference between the signals.

## 3.2 Hardware and Test Setup

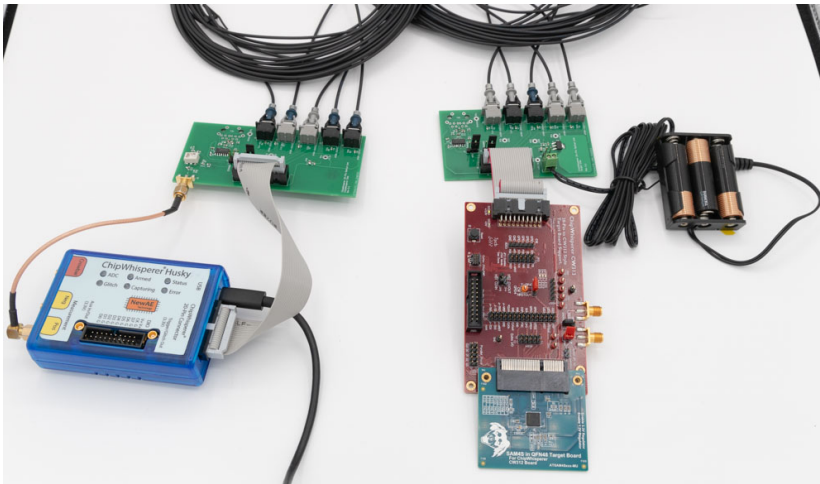
To validate the basic assumptions of this data-dependant delay, we use two different isolator technologies to isolate the normal ChipWhisperer target board from the ChipWhisperer capture platform. The two different implementations tested are shown in Figure 3 and Figure 4. This uses two different technologies of isolation barriers to separate the target from the capture oscilloscope.

The first board in Figure 3 uses MAX22164FAEE galvanic isolators, which are rated up to 200 MHz. Similar isolators were also used in [SMTG23]. The MAX22164FAEE use





**Figure 3:** IC-based galvanic isolator board.



**Figure 4:** Optical galvanic isolator board (normally separated by  $>1\text{m}$ , shown nearby for photo only).

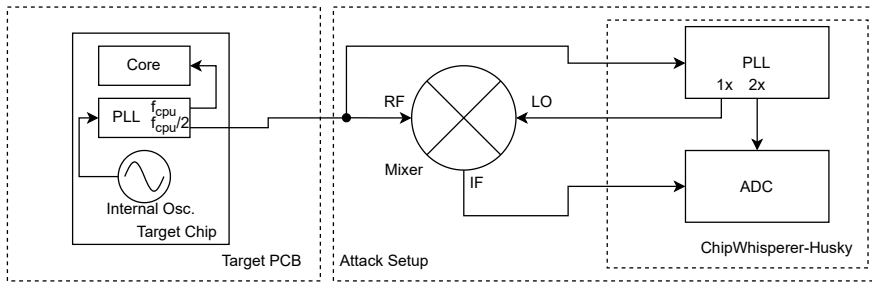
capacitive coupling to achieve galvanic isolation. Due to their close proximity, there could still be some signal coupling across. To avoid this possibility, we also setup an optical isolator test board shown in Figure 4, and power the target from batteries. Tests with optical isolators were included in [SPK<sup>+</sup>10], but using discrete optical components which transfer analog signals (and also coupled some leakage). Instead we used Broadcom optical transceiver devices (AFBR-1624Z and AFBR-2624Z), which include signal conditioning and have purely logic-level interfaces. More details are available in Appendix B, as well as on the associated git repository.

Both of these boards also have the ADE-1+ mixer mounted, allowing the boards to be used to directly for phase measurement purposes. Since the phase measurement is measured with the ADC on the ChipWhisperer-Husky, we can first perform a relative baseline capture to understand the classic shunt-based power analysis leakage results.

For all of the tests in this work, an Arm (ATSAM4S2A) device is used. Most of these results use an internal oscillator, which a PLL in the ATSAM4S2A is used to set the CPU core frequency. As will be discussed, the device is used at 15 MHz or 120 MHz core

**Table 1:** For comparison, results of shunt-based attacks on the ATSAM4S2A target, where  $|TVLA|$  is maximum absolute t-test result after 10K traces, and CPA is number of traces required for complete key recovery. Device is running from internal oscillator on all examples.

CPU Freq	Asynchronous Sampling				Synchronous Sampling			
	120 MS/s		60 MS/s		120 MS/S		60 MS/s	
	$ TVLA $	CPA	$ TVLA $	CPA	$ TVLA $	CPA	$ TVLA $	CPA
120 MHz	5.0	28375	4.2	66750	43.9	750	28.8	1350
15 MHz	5.9	19900	5.0	22800	24.0	3250	25.8	3600



**Figure 5:** Baseline validation is done by measuring the jitter relative to a cleaned-up clock. This requires only a connection to the clock output from the target device.

frequency.

The sampling is done with the ChipWhisperer [OC14], specifically the ChipWhisperer-Husky. Note that the ChipWhisperer is normally using a synchronous sampling, which provides much better results for the same sampling rate [OC12, OC15]. Appendix A provides some calibration information on the target, a summary of this is presented in Table 1.

The results of the baseline power measurement use the result format discussed in Section 2.3. In general we will use the synchronous sampling setup as it provided the best results, although throughout the paper results of both asynchronous and synchronous will be presented to reflect a variety of attack scenarios.

Note as well that the internal oscillator of the device drifts slightly, so the power traces become less aligned further from the trigger event. This hurts the asynchronous results more than if the device had an on-board crystal oscillator. See [OC15] and Appendix A for more detail.

### 3.3 Clock Source

In order to measure the jitter in the output clock with a mixer, we need a source clock to measure our jitter against.

For this work, the target is configured to output a clock at half the CPU frequency out a pin. This pin is fed into the ChipWhisperer-Husky, where the PLL chip (TI part CDCI6214) is used to clean up the jitter, and the results are fed out a pin of the ChipWhisperer-Husky. A mixer then combines these two clock sources, and the output of this mixer is fed into the ADC input of the ChipWhisperer-Husky. This setup is shown in Figure 5. This figure does not show the isolation barrier which exists between the target PCB and attack setup blocks.

The mixer output is sampled at  $2\times$  the input clock frequency, using another PLL

output pin. This maintains a phase synchronization to the target.

This setup is the most realistic scenario as it is applicable even with limited physical access. This also more closely matches typical radio receiver design, where a local clock is locked to the carrier.

Other variations of the setup include the case where a low-jitter input clock is available from the target as well. This would be the case where a target device uses a crystal as a frequency reference, and generates a higher frequency from a PLL. In that case the ChipWhisperer-Husky could use this same frequency reference to generate a phase-locked sampling clock, and then sample the jittery output signal (e.g., from a communications interface or similar).

As yet another variation, when we attack the JTAG interface, we will instead feed our own reference clock into the device. As long as we have both a reference clock with lower jitter & a jittery clock from the target, the attack should succeed.

### 3.4 Mixer Leakage Results

The output clock from the target is a 60 MHz clock, which is derived from the internal MCU clock of 120 MHz divided by two. The ChipWhisperer-Husky takes this 60 MHz clock and multiplies it by two using the PLL block (which also cleans up the jitter), which forms the low-jitter reference clock (and also the sampling clock).

This allows us to sample the output of the mixer at both rising and falling edges of the 60 MHz output clock from the target. Both edges contain phase information, so we want to sample at two points per clock cycle.

An example trace as measured at the mixer output is shown in Figure 3.4. The top shows the mixer output for a single measurement (left) and an average of 10000 traces (right). Both closely match a reference wave (bottom) captured using a shunt resistor.

Both the ic-based and optical isolator based boards result in similar power traces visually. Looking at the results of both the TVLA and CPA attacks in Table 2 shows that the attacks are surprisingly effective. These results can be directly compared to the shunt based reference which took 750 traces (120 MS/s sampling, 120 MHz CPU, synchronous capture cells in Table 1). It took only about four times more traces to recover the full key when measuring the optically isolated clock signal, compared to a direct shunt measurement on a filtered power supply from the CW313 baseboard.

**Table 2:** Strong leakage is seen with both the IC-based and optical isolator. Target is running at 120 MHz, with a 60 MHz clock output.

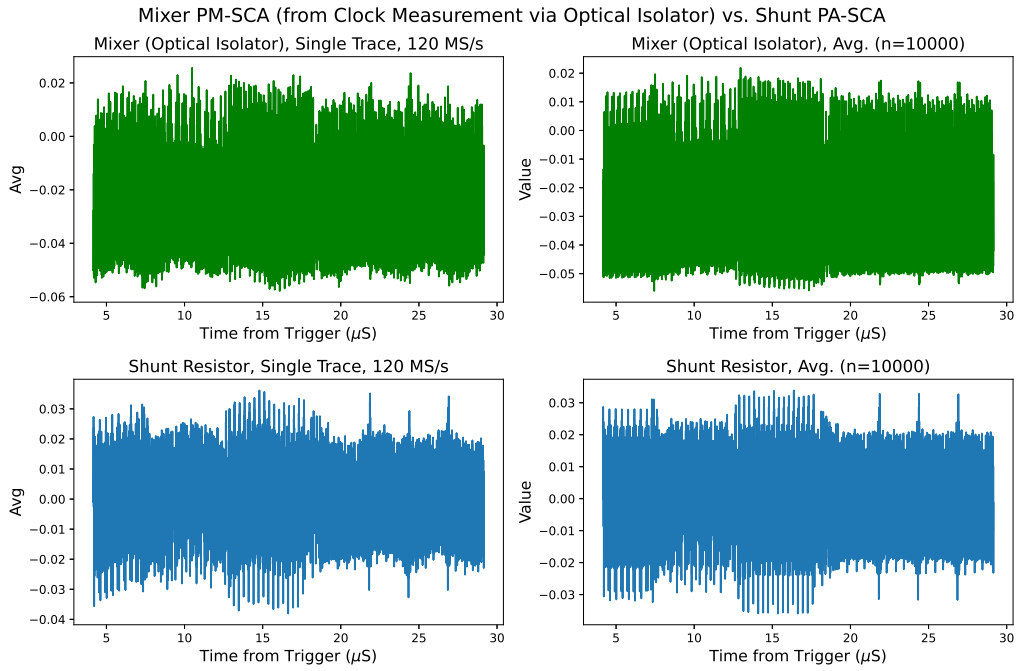
Measurement	Isolator	PLL Filter	TVLA	CPA
Mixer	IC-Based	Default	21.7	3900
Mixer	IC-Based	Lowered	20.7	3400
Mixer	Optical	Lowered	24.1	2775
Voltage Divider	IC-Based	Lowered	4.7	25775

The results of the CPA attack output are shown visually in Figure 7, which shows a very “classic” looking CPA peaks for each of the sixteen bytes in our software based AES implementation.

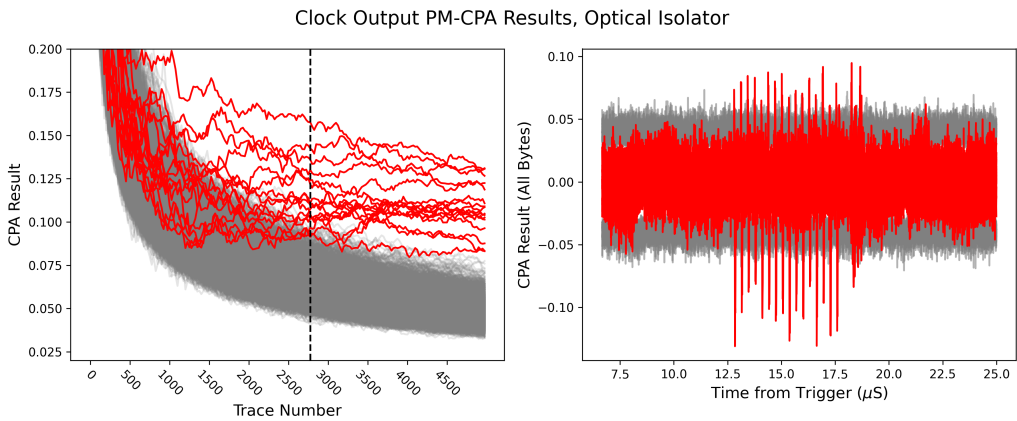
It can also be seen that the voltage divider output in Table 2 (which *does not* use the mixer IC, but relies on the mixing effect of the ADC described in Section 3.1.1) was much less effective, but still exceeds the 4.5 TVLA threshold suggesting exploitable leakage.

### 3.5 Direct Sampling Results

While better resolution will be achieved by using purpose-built phase measurement tooling, a sufficiently fast sampling rate can be used to directly sample the output clock. As a



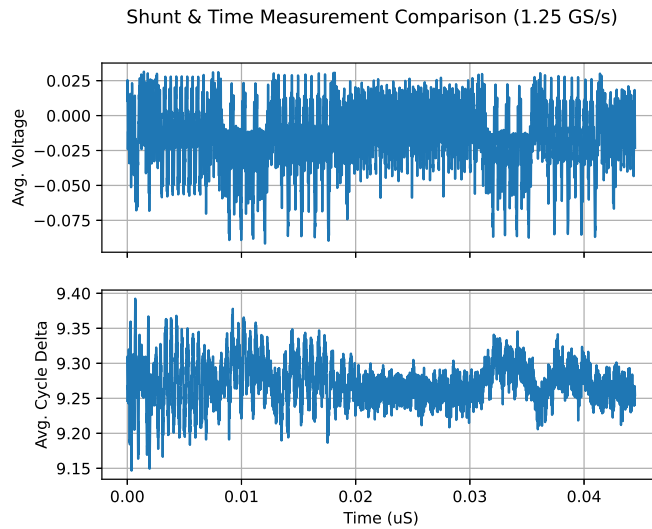
**Figure 6:** The waveform at the output of the mixer through optical isolators (top, green) looks very similar to a waveform recorded across a shunt resistor (bottom, blue).



**Figure 7:** The results of a standard CPA attack on the mixer output look similar to a CPA attack on a power trace, here shown as both output convergence (left) and peaks over time for all 16 bytes. Dashed line on the left at 2775 indicates where full key recovery occurs.

short demonstration of this, a PicoScope oscilloscope which is sampling at 1.25 GS/s will be used.

To do this, the output clock of the target device is connected to the oscilloscope input in AC-coupled mode. The time between each zero-crossing is then recorded as a “new” power trace. The effective sample rate of this “new” power traces is much lower, in Figure 8 the shunt measurement has been scaled to match the (lower) effective sample rate.



**Figure 8:** Comparing a shunt measurement (top) with a delay trace created from measuring the time between zero crossings of the clock output (bottom). The direct measurement used for this figure results in lower resolution compared to the techniques from Section 3.4, but it demonstrates that even a low-end oscilloscope can be used *as-is* for detecting this phase leakage.

This direct sampling *did not* result in a TVLA result indicating a useful leakage, but as can be seen from Figure 8 the waveform has a clear appearance that follows the shunt measurement. Due to the internal oscillator drift the delay measurement quickly becomes less aligned with this measurement technique. No useful results were obtained in an unprofiled CPA attack (agreeing with the t-test results), but the clear appearance of identifiable structure suggests this technique could be useful for future work.

### 3.6 I/O Pin Leakage Evaluation and Baseline

As discussed in [SPK<sup>+</sup>10], measurements of the I/O voltages can lead to simple measurement of the power supply. In [SPK<sup>+</sup>10], this is done for three settings: a GND pin, a I/O pin set high, and a I/O pin set low. Of the devices tested in [SPK<sup>+</sup>10], only one of them (an FPGA) has a different core voltage than an I/O voltage. That particular device shows the best leakage from the ground rail, some leakage when the I/O pin is set low, and no leakage when the pin is set high.

We expect on our device to see similar leakage, as our ATSAM4S2A also has separate core and I/O voltage rails. As seen in Table 3, our device shows the best leakage from the I/O pin being set low. When measured directly (without an isolator) the TVLA result of 5.1 suggests an exploitable leakage.

Adding the isolators (only the best IO-Low results shown in the last two rows) results in the TVLA peaks being below 4.5 (the level of 4.5 being a typical threshold above which a device is considered leaking [GJJR11]). The IC-based isolators appear to do a better

**Table 3:** Leakage on a static header pin, measured on (1) a GND pin, (2) the I/O pin set high, and (3) the I/O pin set low. This is repeated for both isolator technologies to compare the effect of adding the isolator.  $|TVLA|$  measured at 10k traces.

Measurement	Isolator	$ TVLA $
GND	None	3.8
IO-High	None	3.7
IO-Low	None	5.1
IO-Low	IC-Based	2.0
IO-Low	Optical	3.5

job of removing this small amount of leakage, which could be an artifact of the IC-based isolators being slightly more electrically noisy. In all cases the leakage is much lower than that measured from the clock signal phase, and CPA attacks up to 100K traces are not successful.

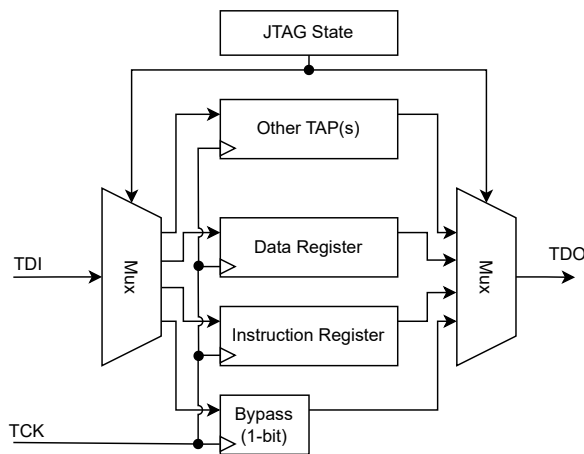
## 4 Power Analysis Over JTAG

Most digital devices implement a Joint Test Action Group (JTAG) interface, as it’s used for in-circuit test (using the Boundary Scan mode), as well as used for programming and debug. The fundamental basis of these ports are that they use a *synchronous serial* interface, with a data input (Test Data In, or TDI), data output (Test Data Out, or TDO), and a clock (Test Clock, or TCK).

A fundamental idea of JTAG is the idea of a *scan chain*, where multiple devices are connected together into one chain. For this to function, every device supports a *bypass* mode, which connects the TDI input to the TDO output. This connection is done with a single register as shown in Figure 9.

The bypass register has a mux that can be used to select other modes as well, so there is some combinational logic on both the input and output of the register.

Bypass mode is typically supported whether the JTAG port is used in boundary scan or debug mode. Because bypass mode is expected to be available to avoid breaking the “scan chain”, bypass mode is often available even when certain debug functionality have been disabled for security reasons. Some devices *do* have an explicit ability to disable bypass mode, as will be discussed in Section 4.2, which blocks this attack.



**Figure 9:** JTAG includes multiple registers, including a single-bit bypass register.

As bypass mode is passing a digital signal across the same fabric as might be running sensitive cryptographic operations, bypass mode can be used as a *highly effective* oscilloscope for performing side-channel power analysis measurements based on signal delay.

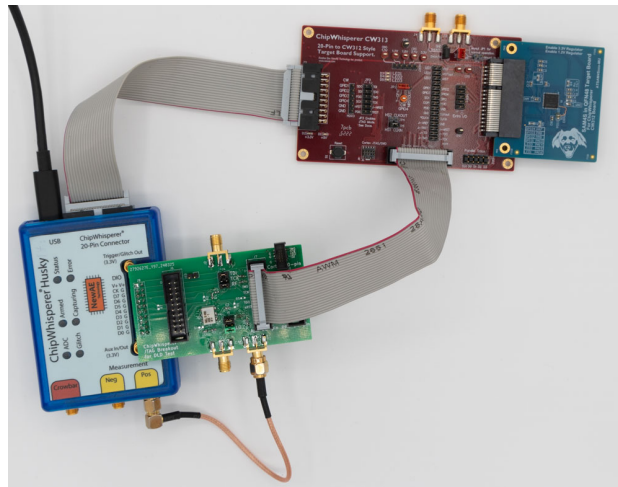
We will begin by demonstrating a power analysis attack on an AES implementation running in the ATSAM4S2A target used earlier. We will then measure the sampling rate that is possible across several devices, including the small microcontroller used here, MCUs, and FPGAs. We will then discuss additional measurement methods including using fault correlation analysis with JTAG, and testing simple countermeasures.

## 4.1 Software AES Attack

In order to perform the attack, the "UserIO" pins are used on the ChipWhisperer-Husky. These can perform bit-banged I/O sequences, and thus can be used to put the JTAG core into bypass mode. If using external equipment, this would mean holding the TMS pin low, and floating the TCK and TDI pins, then feeding appropriate square waves into the TCK and TDI pins.

To sample the delay through the JTAG bypass register, the TDI pin is fed a clock that is half the frequency of the TCK pin, with an appropriate phase offset such that the TDI pin is sampled in alternating high and low states by the internal JTAG logic.

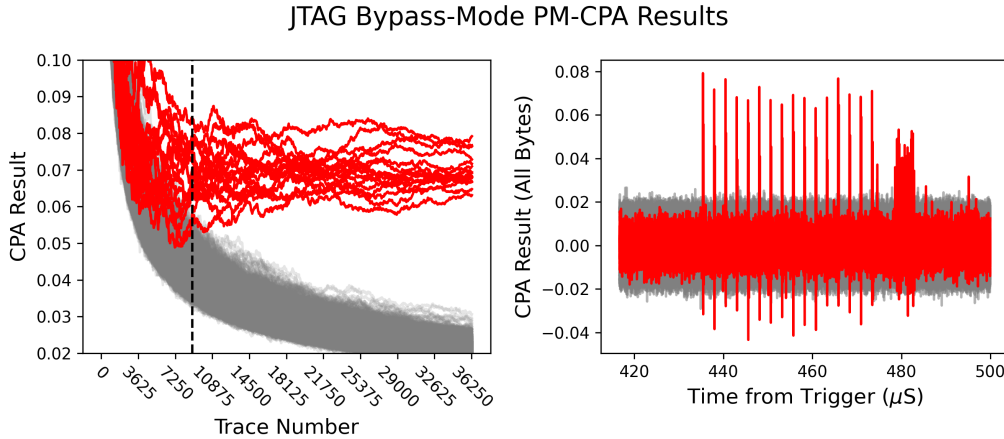
As before, a phase measurement can be taken by comparing the input signal at TDI to the output signal at TDO. Again a mixer is used to perform the phase measurement between TDI and TDO.



**Figure 10:** This board can be used for JTAG attacks using mixer-based phase measurements (Section 4.1) and fault-based measurements (Section 5.1).

### 4.1.1 Hardware Setup

For this attack, the physical setup is shown in Figure 10. Note that there is *no* galvanic isolation in this setup. This was done as the attack will compare results with the JTAG clock turned off later, to determine if there is any coupling of the core voltage onto the I/O voltage as described in [SPK<sup>+</sup>10], and explored in Section 3.6. If an attacker could perform an attack by measuring the static I/O voltage in this setup, it would be an unfair comparison to use a galvanic isolator. The objective is to show that the JTAG measurement is a *more serious* threat model than measuring power at accessible I/O headers as in [SPK<sup>+</sup>10].



**Figure 11:** CPA results of PM-CPA, using the JTAG scan chain as the delay measurement source.

To perform the JTAG measurement, the ChipWhisperer-Husky FPGA code was extended to allow feeding the required clocks onto the TCK and TDI pins. This means no external hardware (such as signal generators) are required. These modifications have been made open-source as part of the repository associated with this paper.

The same ADE-1+ mixer is attached to the TDI and TDO lines, which carry the signal going to and coming back from the target. More details of this are provided in Appendix B, and in the open-source repository.

If using asynchronous measurements, the TDI and TDO pins is not synchronized to the device clock, and it may be either high or low at the beginning of the measurement (depending when the trigger event occurs).

This requires a simple preprocessing step of shifting the captured waveform by a sample to ensure all the captures start at the same state of the TDO output. There is a slight loss of data here, so better results are achieved by synchronizing the clock to the target device, which is possible with the ChipWhisperer-Husky.

#### 4.1.2 Results

All measurements are taken at 30 MS/s, with the target running at 15 MHz. As will be explained in Section 4.2, this is related to the specific target board being less stable when driven at high JTAG frequencies.

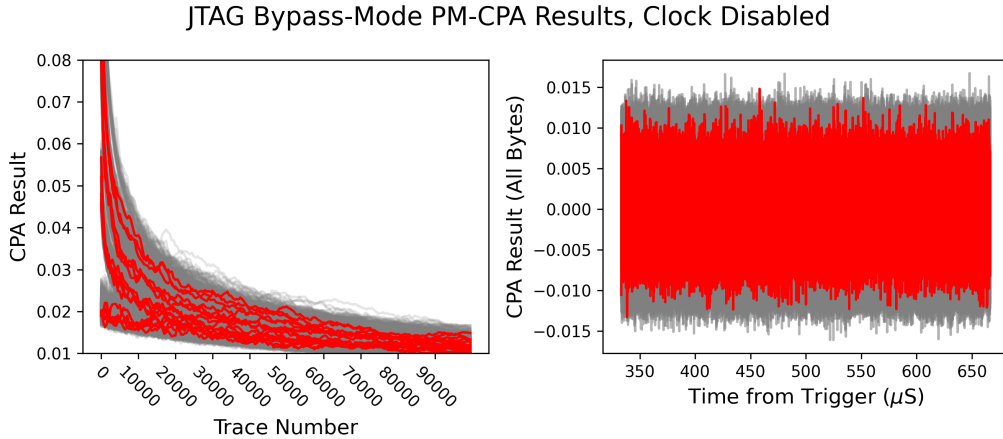
To provide a comparison for this same sample rate, two baseline measurements will be performed here using a classic voltage input ADC, such as used with shunt measurements or EM measurements: (1) a measurement across a shunt resistor, (2) and a measurement at the mixer output with the TDO pin connected to the bypass register but no signal driven into TDI.

The objective of this is to first define what a classic shunt-resistor based CPA attack results are, and then the second measurement is confirming that there is no useful leakage occurring from the I/O pin itself.

A comparison of the attack measurement and the two baselines are given in Table 4.

The CPA results themselves for the JTAG based PM attack are given in Figure 11. Using the exact same setup the toggling TDI is disabled (held constant), which drives TDO low. This becomes the CPA attack for a constant I/O pin value. The attempted CPA attack results for the I/O pin leakage is given in Figure 12. With the TDO pin driven





**Figure 12:** The same setup as Figure 11 but the input to TDI is stopped to force the TDO to a constant value.

**Table 4:** Results of JTAG delay measurements using the board from Figure 10.  $|TVLA|$  is the t-test result after 10K tests, and CPA is the number of traces required for full key recovery. All results at 30 MS/s, with target running at 15 MHz.

Measurement Basis	$ TVLA $	CPA
Shunt on VCC	24.0	1800
JTAG: Mixer, Toggling TDO	11.7	8900
JTAG: Mixer, Constant TDO	4.0	–

low none of the key bytes were recovered after 100K traces, and as can be seen no useful leakage was visible.

This confirms that the JTAG delay measurement is the source of the strong leakage, not the static I/O pin leakage as was demonstrated in [SPK<sup>+</sup>10]. Heavy averaging of the I/O pin signal shows *some* leakage (see Figure 18 in a later section for this visually represented). The objective of this is to confirm that the JTAG based phase measurement technique results in a significantly stronger signal than the I/O pin leakage itself.

## 4.2 JTAG Functionality Across Devices

The implementation of the JTAG port varies with devices. In this section we consider two important variations: the maximum operating frequency, and the ability to disable the JTAG port.

The first, the maximum operating frequency, defines the maximum frequency we can use to acquire phase measurement information. We will consider this the TCK frequency, where on each edge of the TCK line we shift out a new bit (and thus new phase shift information). The datasheet includes a maximum *specified* frequency, this is for the entire JTAG block, and we only need the bypass mode to operate. We expect the maximum TCK frequency where bypass mode works to be higher than the specified maximum TCK frequency for using the entire block.

Table 5 summarizes the maximum frequency we observed a stable square wave being fed through the TDI to TDO pin. Note this table presents only a measurement of a functioning JTAG scan chain itself. For some boards the high JTAG scan chain frequency interrupted regular device operation. The SAM4S2A target board used here for example does not

have decoupling capacitors (to improve the shunt-based results). Driving the JTAG TCK frequency beyond 40 MHz caused the device to not reliably boot, even though the JTAG scan chain itself was still operating. We anticipate this may be caused by the board being setup for the most reliable shunt-based power measurements, at the expense of stability (lack of decoupling capacitors). In Section 5 the decoupling capacitors are mounted for increasing the JTAG frequency to allow more effective fault sensitivity measurements.

The last column of Table 5 summarizes if devices have an “always-on” JTAG port which allows bypass mode regardless of debug security settings. Devices marked with a ‘Y’ in this column indicate the JTAG bypass mode is always available. Some devices, such as the STM32 for example, have an ability to disable the JTAG port. While this disabled port can sometimes be re-enabled by fault attacks, doing so is outside the threat model of this paper (since an attacker with such access could perform an effective side-channel attack without the JTAG port at all).

More details of the specific capabilities of each devices is given in Appendix C. Devices with limitations are marked ‘N’ in the table: the ATSAM4S2A for example has an always-on JTAG port, but it requires setting a specific I/O pin to force the JTAG port into boundary scan mode. As this is again outside of the threat model presented in Section 1.1, we have marked it as ‘N’.

**Table 5:** Maximum measured JTAG TCK rate at which a stable square wave can be clocked through on various devices. Occasional bit errors may be present.

Device	Type	Spec MHz		Actual MHz	JTAG
		Core $F_{max}$	TCK $F_{max}$	TCK $F_{max}$	Always On
SAM4S2A	Arm MCU	120	–	210	N
STM32F303	Arm MCU	72	–	173	N
STM32G474	Arm MCU	170	–	208	N
MPC5676R	PPC MCU	180	10	200	Y
MPC5777C	PPC MCU	300	10	186	Y
MK24FN1M0VLL	Arm MCU	120	25	181	Y
XC7A35T	FPGA	N/A	66	105	Y
XCS6LX75	FPGA	N/A	66	125	Y

### 4.3 Multiple TAPs on one JTAG Port

Many devices have multiple internal test access port (TAP) modules connected to one physical JTAG port. The MPC5676R from Table 5 has eight separate JTAG TAPs, each one supporting a bypass instruction.

This means that on some devices it’s possible to “move” the sensor around on the die. The results of running the same 10K T-Test leakage evaluation, using the JTAG mixer board of Figure 10, for the various TAPs is presented in Table 6 (see Appendix A for more information on the setup). In the TVLA results of Table 6 the CPU is running at 16 MHz, the TCK frequency is 32 MHz, and the sampling is at 32 MS/s

Table 6 also includes the maximum TCK frequency that allowed for *zero* bit errors seen on the TDO output for at least 16376 consecutive samples (the size of the logic analyzer buffer in the ChipWhisperer-Husky). This is a more strict condition than used for the maximum frequency of Table 6. It again demonstrates different characteristics of the TAP interfaces, even though all are using the same physical JTAG port.

**Table 6:** Testing TAPs within the MPC5676R. TVLA results based on 16 MHz core clock, 32 MS/s, software AES.  $TCK_{F_{max}}$  based on 16376 non-erroneous TDO  $\rightarrow$  TDI shifts.

TAP Name	Instruction	TVLA	$TCK_{F_{max}}$
Default	11111	4.55	60
NPC	10000	3.79	164
ONCE	10001	4.18	160
eTPU	10010	4.04	164
NXDM	10011	4.61	96
NXFR	10100	3.38	160
eTPU2	10110	4.15	92
NXDM_B	10111	4.28	88
ONCE_1	11001	3.44	164

## 5 JTAG Input and Internal Delay Measurement

Instead of measuring the *output* delay using external equipment, we can also observe the phase modulation data by measuring an internal state that will depend on the propagation delay of an input or internal signal path.

This is more powerful since it can be much more difficult to protect with countermeasures (to be discussed in Section 6). The general idea of this is known as fault correlation analysis [LSG<sup>+</sup>10, MMP<sup>+</sup>11, SMC21] (see Section 2.2.3 for more background).

To perform the fault correlation analysis, the same setup as in Figure 10 is used, except only a standard 20-pin cable is connected, no SMA cable to connect the mixer output is needed. The ADC is not used at all in this setup.

The ChipWhisperer-Husky includes a logic analyzer (LA) functionality on the digital input/output pins, which can be used to monitor the TCK/TDI/TDO pins we are using to communicate with the device. This LA functionality can be setup to run from the same clock we generate on TCK, and thus we can sample the output TDO pin from the device to detect faults on our high-speed TDI/TDO signal path that was previously used for PM measurements. The LA cannot directly measure phase information as the sample rate is too low, but we *can* use it to measure the digital output state.

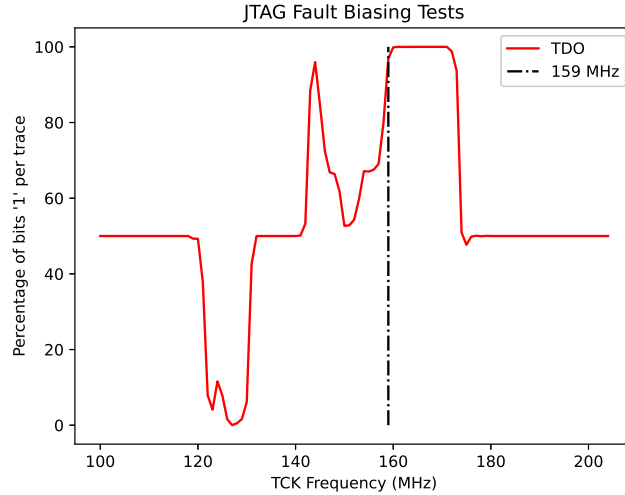
A similar analysis could be done from any output signal, we simply need a way of *biasing* the device such that it is producing faulty output data. Here the JTAG scan chain has the advantage that we can bias the JTAG communications path to a faulty state by adjusting the timing of the TCK and TDI pins to violate the setup and hold times. The JTAG communication core is independent from the microcontroller core, and thus we are not introducing faults into the actual operations. This also means fault detectors will not detect our fault biasing setup.

In general, using a synchronous communications interface and biasing it by adjusting the timing of input data to cause faulty data to be latched in will be effective for this technique. Using the JTAG communications interface again has the significant advantage of not impacting normal operation of the device.

### 5.1 Fault Correlation Analysis and 1-bit TDC

The classic design of the Time to Digital Converter (TDC) uses an array of flip-flops, with the same signal fed through a chain of delay elements. Each tap in the delay element is fed to a flip-flop. With appropriate delay setup, the circuit is configured to sample how far an input travelled through the delay chain, converting the input pulse phase to a digital signal [MCA19].

A figure showing the JTAG scan chain when used in bypass mode was shown in Figure 9.



**Figure 13:** Depending on the phase we expect to see either 100%, 50%, or 0% of 1-bits. Anything else is a faulty area.

Of importance, note that the input mux is a combinational logic circuit that will be subject to variation of propagation delay. In Section 4 we were measuring the *output* delay, but we are here effectively measuring the *input* delay. By biasing our input clock edge to be such that we see roughly an equal number of faults (incorrectly sampled bits), changes in the voltage will result in faults becoming less likely or more likely.

The disadvantage of this is we have only a single bit to sample this data. The effect of reducing the bit-depth of ADC measurements in power analysis measurements was explored in [OD19], and thus we expect the leakage to be present still, just requiring more traces. The case of a single-bit TDC specifically was demonstrated in [JUP24], again confirming that a 1-bit TDC is effective for side-channel attacks.

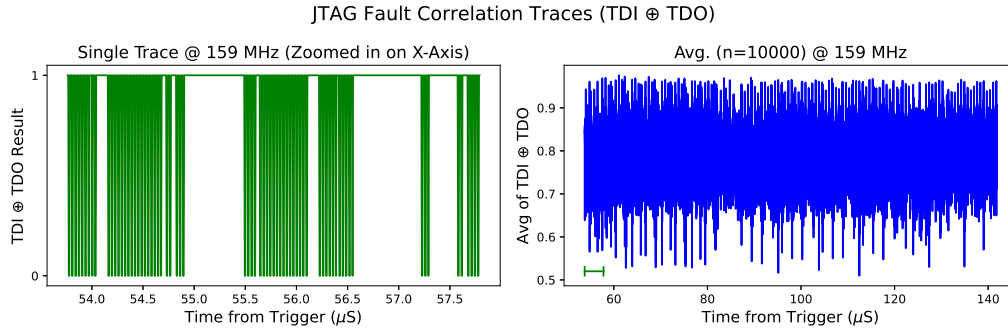
For this work, the JTAG bypass scan-chain measurement technique could be considered both a 1-bit TDC or a fault result. If considering it a fault result, we are measuring faulty data being shifted in/around the scan chain. As the majority of similar work has focused on fault correlation analysis (especially around the question of low-bit measurements), we will refer to this as a fault correlation analysis problem.

Compared to more classic fault correlation analysis, this scan-chain fault analysis has the advantage that we can sample at very high frequencies. We are able to receive a data point at each clock edge, and not simply receive a data point on the overall success or failure of the operation, as in [SMC21]. We must emphasize that direct comparisons to classic fault correlation analysis are not entirely “fair”, as the scan-chain fault analysis has more discrete time sampling information (closer to that of a TDC).

## 5.2 Determining Fault Parameters

To perform the fault correlation analysis, we use the same ChipWhisperer setup, which can feed a clock into TCK, and a half-rate clock into TDI. If we needed to run this at a specific frequency we could use a dynamic phase shift to adjust the phase until we observed the expected faults on the output to bias our measurement setup.

A simpler method is to sweep the frequency setting, as the delays on the TCK and TDI pins will also vary with frequency due to our design. As seen from Figure 13, there are several frequencies resulting in a number of faults on the bypass register.



**Figure 14:** Fault correlation at 159 MHz, showing single and average traces. Single trace is from a smaller area as marked on the lower left of the average trace.

Note that the ChipWhisperer-Husky can also generate clock glitch waveforms, and thus we could instead send a glitchy clock on TCK (or glitchy data on TDI). We have chosen to instead concentrate on a simple square wave pattern, as we expect the square wave pattern to be more easily repeatable on other (non-ChipWhisperer) platforms. This also demonstrates that any system which can be biased into a natural faulty state can be used for fault correlation analysis.

We would emphasize that simply looking for a “natural faulty state” could be that faults are occurring on the input of the *measurement device*. In this case we would be measuring the *output* phase differences, since small variations in the phase will still cause the likelihood of a ‘1’ being sampled vs a ‘0’ being sampled. While this will also work, the measurements will be much more sensitive to noise from power fluctuations on the *measurement device*. Using the phase adjustment blocks in the FPGA can be used to validate the measurement device input violating setup and hold time is not the source of these faults.

### 5.3 AES Attack Results

The same 15 MHz CPU setting for a software AES attack will be performed as in Section 4.1. The only difference from previous examples is the decoupling capacitors have been mounted on the microcontroller to avoid issues with usage of the JTAG port at higher frequencies.

Based on Figure 13, the ChipWhisperer-Husky was configured for a 159 MS/s sampling rate (again, this is the logic analyzer sampling rate, the ADC is not used). This will output a 159 MHz signal on TCK, with a 79.5 MHz signal on the TDI pins. Two examples of a captured “trace” are given in Figure 14, one is a single trace, and one is an average of a larger trace set for the T-Test group to show how the traces average out to a power-trace visually.

Each sample point is the result of  $\text{TDI} \oplus \text{TDO}$  at the sample time. If there are no faults we expect TDI and TDO to match on each clock edge (or, depending on timing, may have an offset of one cycle as discussed).

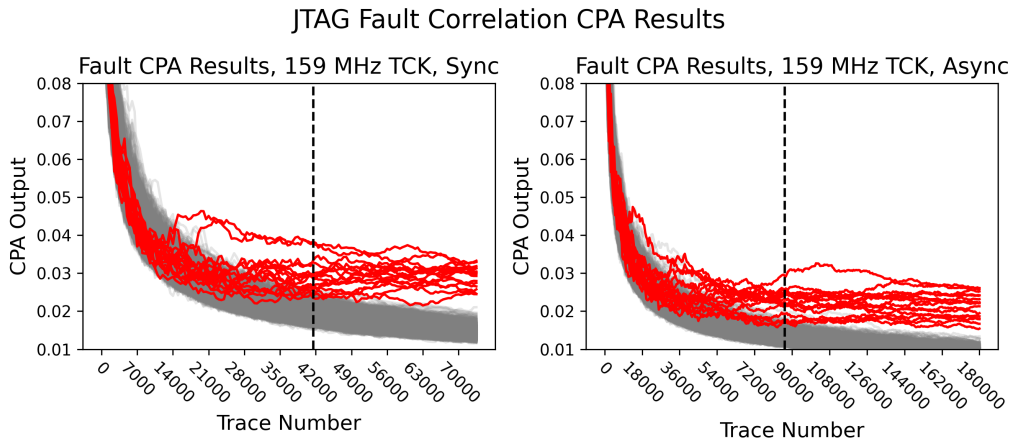
The sampled value is fed in as a power trace to the CPA attack, which has the standard Hamming weight (HW) leakage model [BCO04] used for all other results in this paper.

Table 7 details the TVLA and CPA results for our fault sensitivity attack. Note the TVLA result here is similar for both asynchronous and synchronous, but the CPA result does show improved results with synchronous sampling. Compared to the baseline measurements in Table 1 the difference between synchronous and asynchronous measurements is reduced. This may be related to the very simplistic application of the classic CPA leakage model being applied to our fault trace.

**Table 7:** Results of fault correlation analysis attacks on the ATSAM4S2A target launched over the JTAG interface, where  $|TVLA|$  is maximum absolute t-test result after 10K traces, and CPA is number of traces required for complete key recovery. Device is running from internal oscillator on all examples.

CPU Freq	Asynchronous Sampling 159 MS/s		Synchronous Sampling 159 MS/s	
	$ TVLA $	CPA	$ TVLA $	CPA
15 MHz	5.0	86500	4.7	41500

We can also compare these results to the fault correlation attack in [SMC21], where an unprofiled CPA attack takes at best 714K traces, compared to 87K traces in our work. The much lower trace count in our work is primarily related to our ability to achieve very high sample rates for our fault correlation analysis, and not using a single point per trace like classic fault correlation analysis.



**Figure 15:** CPA attack results for both asynchronous (JTAG-port only) and synchronous (CPU clock connection required). Red traces are correct key bytes. Dotted line shows when full key recovery is reached.

## 6 Countermeasures

Countermeasures which are effective against side-channel power analysis at an algorithmic level are expected to be similarly effective against jitter measurements. This section discusses specific engineering-level countermeasures which can be applied to existing or new systems to hide the jitter from an attacker.

Some existing engineering style countermeasures against power analysis may be effective against PM leakage. For example random clock switching is frequently employed [GM11]. Provided an attacker cannot lock to this clock using techniques in [OC15] this could be effective against jitter measurements. An attacker using the ‘direct sampling’ technique in Section 3.5 may be able to also perform the attack by simply post-processing the captured jittery traces.

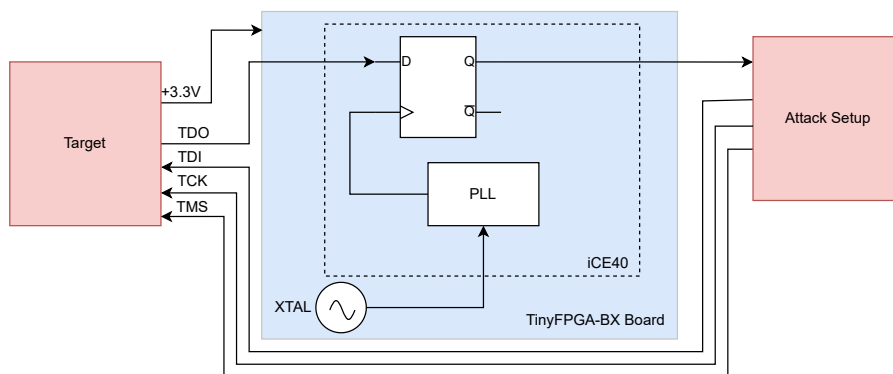
Notably, a number of existing devices also have an option to fully disable the JTAG port (such that even boundary scan does not work). This was discussed in Section 4.2, and if a device has the JTAG port fully disabled it provides the *most* effective countermeasure.

## 6.1 Communications Protocols Repeaters and Re-timers

On high-speed communications protocols such as Ethernet, USB, SATA, PCIe, and HDMI, it is very common to use a *repeater* in certain situations. These repeaters may simply be *redrivers*, which are similar to an analog amplifier, in that they take the input signal and re-transmit it, trying to add as little additional jitter as possible. Such re-drivers may reduce the leakage, but fundamentally will not remove it.

Instead such interfaces should pass through some form of *retimer*. A retimer has some level of decoding of the input protocol, such the transmitted protocol is actually reconstructed based on a *new local oscillator at the retimer*. Examples of devices which include retimer functionality include Ethernet hubs & USB hubs.

If using a high security device which includes a USB interface, adding a USB hub IC would prevent jitter measurements from being measured externally to the device. Some interfaces that may be easily accessible, such as SD cards, do not typically have retimer ICs available. For these devices, it may be sufficient to build a dumber resampler described next.



**Figure 16:** A simple resampler provides protection against direct measurement, but still has some risk of fault injection correlation attacks.

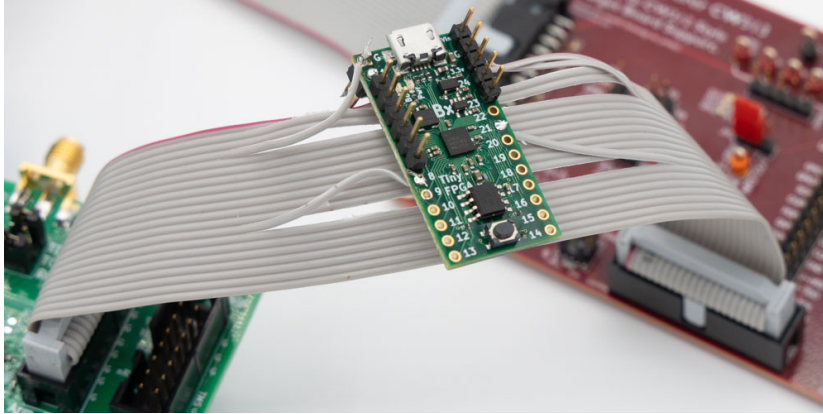
## 6.2 Digital I/O Resampling

For straightforward digital I/O pins, such as those driving a SPI or JTAG connection, a digital I/O resampler is sufficient to remove the jitter. Note this differs from the retimer as the resampler is *not* retiming the signal to a protocol clock. Instead it's simply using a local oscillator to sample the digital I/O pin state and forward it onward. A simple implementation that will be tested here is shown in Figure 16.

For this to be successful, the resampler must run sufficiently fast to avoid introducing timing errors. On many protocols such as UART, JTAG, and SPI this can be done reasonably well. For example many SPI, UART, and JTAG protocols are typically running at 115 kHz to 20 MHz. As a practical demonstration, the resampler from Figure 16 was implemented in a low-cost iCE40 FPGA. This FPGA can run the resampler at 180 MHz, which is suitable for use on these embedded systems.

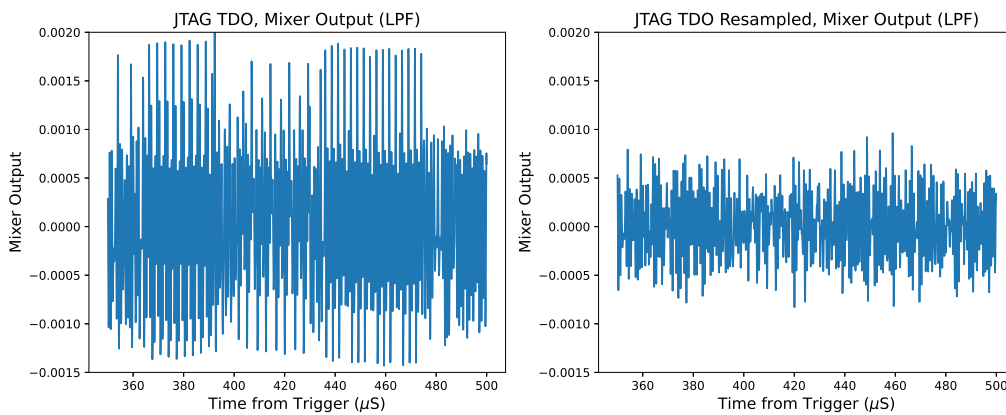
To test the resampler, we used the JTAG attack against AES from Section 4.1. We then inserted a countermeasure prototype into the JTAG cable, as seen in Figure 17. This countermeasure resamples the TDO pin before the data returns to the JTAG delay measurement board (that board is shown in Figure 10, with more details in Appendix B).

Repeating the experiment of Section 4.1 did not result in a successful attack in 100K traces. The T-Test score were also improved, being reduced from 11.7 to 4.1. Inspecting the mixer output traces in Figure 18 *does* still show some visible structure. Remember



**Figure 17:** A simple countermeasure is implemented in a low-cost iCE40 IC to resample the JTAG TDO pin.

this example is not galvanically isolated, so this leakage is likely I/O pin leakage or power supply coupling [SPK<sup>+</sup>10].



**Figure 18:** The countermeasure on the TDO pin significantly reduces the leakage (TVLA results go from 11.8 to 4.1), but there is still some structure seen under heavy filtering. Both of these are average power traces ( $n = 10000$ ) with an additional low-pass filter.

The simple countermeasure will not protect against fault sensitivity analysis as shown in Section 5. For complete protection, resampling both the input and output is required, as the fault sensitivity analysis still provides a relatively strong signal, and is *less* impacted by the resampler on the output. There is still some impact as the resampler reduces the maximum data throughput, but does not prevent the sampling on the input.

Input resampling must be done on both the data and clock lines. Resampling just the data line could still allow an attacker to adjust the waveform on the clock line to find when errors occur, although with less control.

### 6.3 JTAG Activity Blocking

Previous work has tried to detect abuse of the JTAG protocol, such as an attacker fuzzing the interface or transitioning between invalid states [RK10, MGB16, RTBT19]. As



demonstrated by this paper however, the JTAG scan chain can be used for side-channel measurements without anything beyond the bypass operation.

During sensitive operations (such as encryption or decryption), it may be required to prevent any JTAG activity, and hold the pins at a constant level. Practically this is difficult as it would make devices non-compliant with the JTAG specifications. Implementing it as a user fuse or option may allow more sensitive users to partially disable the JTAG port, and accept the non-standard JTAG implementation.

As previously mentioned, existing commercial devices often have a JTAG disable mode which disables the JTAG port (such as the STM32 series with RDP2 level security). Other devices (such as the SAM4S) keep the microcontroller core in reset while performing boundary-scan JTAG operations, which is also an effective countermeasure against this attack.

## 7 Conclusions

Despite exploration of phase modulation (PM) leakage being presented as part of the early work on DPA & EM analysis [AARR03], little has been done to exploit this phase modulation directly until the work of JitSCA [SMTG23]. Indirectly related work on remote power analysis [ZS18, SGMT18a] has included development of techniques and open-source TDCs that can be used for measurements, and more work is needed to connect the work in these subfields.

This paper started with a background on many of these different subfields that all appear to be related to the problem of phase modulation leakage. To extend the work of [SMTG23] the “jitter SCA” (or PM-SCA here) was validated using a new-to-SCA technique (but a historically well-known technique in other fields) of using a frequency mixer, and the leakage is also validated that it can be exploited even if the signal is received through off-the-shelf optical transceivers.

This fundamental leakage property is then used to attack the JTAG scan chain, by turning the scan chain into an effective oscilloscope for side-channel measurements. This requires no analog connections to the target board, and *at most* requires adding wires to measure other digital signals (such as I/O pins, triggers, or clocks). This makes an attack which can be easily replicated by a non-expert user, and one which makes very little assumptions about access to the device. The attack does not require modifications such as removal of decoupling capacitors, heatsinks, heat spreaders, or separation of Package-on-Package (PoP) devices that is typically needed for shunt or EM-based measurements.

All resources associated with the paper are available at <https://github.com/colinoflynn/phase-modulation-sca>.

### 7.1 Phase-Modulation (PM) Measurements

The leakage resulting from this PM measurement is no worse than leakage that would be observed from shunt-based or EM side channels. As seen from the comparison of shunt-based and a best-case PM-based (Section 3.4), the PM-based measurement requires only two times as many traces as the shunt-based measurement *in the best case*. The *best case* means a high-frequency clock being output from the target device *that is processing sensitive data*.

Clocks which are simply on-board the system are not at risk. For example, in PCI there is commonly a PCI reference clock. This reference clock is generated by an external clock device and fed *into* the processor. Clocks such as the DDR clocks, which are generally generated on-board the SoC, would be at risk. However DDR clocks are more difficult to probe, and on most embedded devices would require the same level of physical access as doing an EM probe based attack would have.

Greater consideration should be given to clocks which are sent out an available user interface, or where physical protection against near-field EM attacks is present but the PM-side channel could be monitored without having to bypass the physical protection. Special attention should be paid to clocks running during early-stage bootloaders, for example clocks used by eMMC, QSPI, and SD cards.

Eliminating the data-leaking delay is best done by resynchronizing clocks to a new clock domain. Blocks which add jitter can be helpful, and are often already present in the form of spread-spectrum clocking options.

## 7.2 JTAG Attacks

As this paper demonstrates, a side-channel power analysis attack can be mounted via the JTAG interface using the phase modulation measurement technique. This is particularly dangerous as many boards expose the JTAG interface, even when other measurement techniques may be difficult (e.g., microcontroller behind shield or inside epoxy). If working with a system, simple tests to validate the maximum JTAG frequency in bypass mode can be illuminating. Countermeasures against this can include various levels of modification to the board and design, to either block JTAG during sensitive operations, or to resynchronize signals to another clock.

Previous work on fault correlation analysis has also been replicated in this work, which demonstrates the link between fault correlation analysis and phase modulation leakage. The same data-leaking delay is the root cause of both leakages, just we consider different ways of measuring this delay.

## 7.3 Future Works

The idea of phase modulation being especially useful for electromagnetic attacks was discovered in [AARR03]. Despite this, there appears to have been little recent work testing the idea around EM and RF attacks. More recent RF attacks do demonstrate that frequency-shifts are observed carrying leaking data [CPM<sup>+</sup>18], but this attacks a strong RF carrier. Using the emissions based on the clock frequency of digital devices, but decoding them with a PM receiver (which includes the required carrier phase tracking to decode the phase data) may allow extending the PM side channel attacks to longer distances.

Work to build more advanced measurement platforms and attacks is also likely to be fruitful. There is a broad body of work on TDCs for example, and a TDC-based hardware solution may be more effective on some devices than the mixer-based measurement used in our work. Our usage of a mixer-based solution allowed leveraging the existing architecture of the ChipWhisperer-Husky, but future work to *replace* the ADC block with a TDC block could result in an even more powerful tool. We also focused on the mixer-based solution as it allowed very direct comparisons between shunt-based and phase-based measurements, but future work which focuses only on PM-SCA will be less concerned with the direct comparison.

Further work linking other measurement results to the PM side channel may also lead additional results. Attacks which are successful in the frequency domain for example may be in part due to the phase modulation working as a frequency mixer to change the leakage [MG10]. As demonstrated in this paper, understanding the fundamental leakage source often leads to applications of it in new and unexpected ways, such as the simple attack on the ubiquitous JTAG interface.

## Thanks

Thanks for Jean-Pierre Thibault for discussion and assistance with the ChipWhisperer-Husky FPGA code. This paper would not be possible without the contributors to the open-source ChipWhisperer design, especially the open-source FPGA code developed by Jean-Pierre Thibault, and the firmware developed by Alex Dewar. Thanks also to the reviewers who have substantially improved this paper with important feedback and improved the presentation.

## References

- [AARR03] Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. The EM side-channel(s). In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *CHES 2002*, volume 2523 of *LNCS*, pages 29–45. Springer, Heidelberg, August 2003.
- [AZ23] Milos Prvulovic Alenka Zajić. *Understanding Analog Side Channels Using Cryptography Algorithms*. Springer Cham, 2023.
- [BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Marc Joye and Jean-Jacques Quisquater, editors, *CHES 2004*, volume 3156 of *LNCS*, pages 16–29. Springer, Heidelberg, August 2004.
- [CPM<sup>+</sup>18] Giovanni Camurati, Sebastian Poeplau, Marius Muench, Tom Hayes, and Aurélien Francillon. Screaming channels: When electromagnetic side channels meet radio transceivers. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 163–177. ACM Press, October 2018.
- [CTH<sup>+</sup>22] Yaakov Cohen, Kevin Sam Tharayil, Arie Haenel, Daniel Genkin, Angelos D. Keromytis, Yossi Oren, and Yuval Yarom. HammerScope: Observing DRAM power consumption using Rowhammer. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 547–561. ACM Press, November 2022.
- [DWR<sup>+</sup>23] Colin Drewes, Olivia Weng, Keegan Ryan, Bill Hunter, Christopher McCarty, Ryan Kastner, and Dustin Richmond. Turn on, tune in, listen up: Maximizing side-channel recovery in time-to-digital converters. In *Proceedings of the 2023 ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA ’23*, page 111–122, New York, NY, USA, 2023. Association for Computing Machinery.
- [GJJR11] Gilbert Goodwill, Benjamin Jun, Josh Jaffe, and Pankaj Rohatgi. A testing methodology for side channel resistance validation. In *NIST Non-Invasive Attack Testing Workshop*. National Institute of Standards and Technology (NIST), 2011.
- [GKDG20] Navyata Gattu, Mohammad Nasim Imtiaz Khan, Asmit De, and Swaroop Ghosh. Power side channel attack analysis and detection. In *Proceedings of the 39th International Conference on Computer-Aided Design, ICCAD ’20*, New York, NY, USA, 2020. Association for Computing Machinery.
- [GM11] Tim Güneysu and Amir Moradi. Generic side-channel countermeasures for reconfigurable devices. In Bart Preneel and Tsuyoshi Takagi, editors,

- CHES 2011*, volume 6917 of *LNCS*, pages 33–48. Springer, Heidelberg, September / October 2011.
- [GMO01] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *CHES 2001*, volume 2162 of *LNCS*, pages 251–261. Springer, Heidelberg, May 2001.
- [GPPT15] Daniel Genkin, Lev Pachmanov, Itamar Pipman, and Eran Tromer. Stealing keys from PCs using a radio: Cheap electromagnetic attacks on windowed exponentiation. In Tim Güneysu and Helena Handschuh, editors, *CHES 2015*, volume 9293 of *LNCS*, pages 207–228. Springer, Heidelberg, September 2015.
- [GPT14] Daniel Genkin, Itamar Pipman, and Eran Tromer. Get your hands off my laptop: Physical side-channel key-extraction attacks on PCs. In Lejla Batina and Matthew Robshaw, editors, *CHES 2014*, volume 8731 of *LNCS*, pages 242–260. Springer, Heidelberg, September 2014.
- [GSH<sup>+</sup>15] Xu Gao, Chunchun Sui, Sameer Hemmady, Joey Rivera, Susumu Joe Yakura, David Pommerenke, Abhishek Patnaik, and Daryl G. Beetner. Modeling static delay variations in push-pull cmos digital logic circuits due to electrical disturbances in the power supply. *IEEE Transactions on Electromagnetic Compatibility*, 57(5):1179–1187, 2015.
- [GST14] Daniel Genkin, Adi Shamir, and Eran Tromer. RSA key extraction via low-bandwidth acoustic cryptanalysis. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 444–461. Springer, Heidelberg, August 2014.
- [HiHM<sup>+</sup>14] Naofumi Homma, Yu ichi Hayashi, Noriyuki Miura, Daisuke Fujimoto, Daichi Tanaka, Makoto Nagata, and Takafumi Aoki. EM attack is non-invasive? - design methodology and validity verification of EM attack sensor. In Lejla Batina and Matthew Robshaw, editors, *CHES 2014*, volume 8731 of *LNCS*, pages 1–16. Springer, Heidelberg, September 2014.
- [JUP24] Darshana Jayasinghe, Brian Udugama, and Sri Parameswaran. llutsensor: Detecting FPGA voltage fluctuations using lookup tables. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2024(1):51–86, 2024.
- [KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 388–397. Springer, Heidelberg, August 1999.
- [Kur78] Stephan Kurtz. Mixers as phase detectors. *WJ Tech-Note*, 1978. [https://www.rfcafe.com/references/articles/wj-tech-notes/Mixers\\_phase\\_detectors.pdf](https://www.rfcafe.com/references/articles/wj-tech-notes/Mixers_phase_detectors.pdf).
- [LED<sup>+</sup>13] Yang Li, Sho Endo, Nicolas Debande, Naofumi Homma, Takafumi Aoki, Thanh-Ha Le, Jean-Luc Danger, Kazuo Ohta, and Kazuo Sakiyama. Exploring the relations between fault sensitivity and power consumption. In Emmanuel Prouff, editor, *COSADE 2013*, volume 7864 of *LNCS*, pages 137–153. Springer, Heidelberg, March 2013.
- [LSG<sup>+</sup>10] Yang Li, Kazuo Sakiyama, Shigeto Gomisawa, Toshinori Fukunaga, Junko Takahashi, and Kazuo Ohta. Fault sensitivity analysis. In Stefan Mangard and François-Xavier Standaert, editors, *CHES 2010*, volume 6225 of *LNCS*, pages 320–334. Springer, Heidelberg, August 2010.

- [LTPP07] Jeremy Lee, Mohammad Tehranipoor, Chintan Patel, and Jim Plusquellic. Securing designs against scan-based side-channel attacks. *IEEE Transactions on Dependable and Secure Computing*, 4(4):325–336, 2007.
- [Lyt24] Daniil Lytikov. Investigating time-digital-converters for hardware security in fpgas. Master’s thesis, Oregon State University, 2024.
- [MCA19] Rui Machado, Jorge Cabral, and Filipe Serra Alves. Recent developments and challenges in fpga-based time-to-digital converters. *IEEE Transactions on Instrumentation and Measurement*, 68(11):4205–4221, 2019.
- [MFT<sup>+</sup>14] Noriyuki Miura, Daisuke Fujimoto, Daichi Tanaka, Yu-ichi Hayashi, Naofumi Homma, Takafumi Aoki, and Makoto Nagata. A local em-analysis attack resistant cryptographic engine with fully-digital oscillator-based tamper-access sensor. In *2014 Symposium on VLSI Circuits Digest of Technical Papers*, pages 1–2, 2014.
- [MG10] Edgar Mateos and Catherine H. Gebotys. A new correlation frequency analysis of the side channel. In *Proceedings of the 5th Workshop on Embedded Systems Security, WESS ’10*, New York, NY, USA, 2010. Association for Computing Machinery.
- [MGB16] F. Majeric, B. Gonzalvo, and L. Bossuet. Jtag combined attack - another approach for fault injection. In *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5, 2016.
- [MMP<sup>+</sup>11] Amir Moradi, Oliver Mischke, Christof Paar, Yang Li, Kazuo Ohta, and Kazuo Sakiyama. On the power of fault sensitivity analysis and collision side-channel attacks in a combined setting. In Bart Preneel and Tsuyoshi Takagi, editors, *CHES 2011*, volume 6917 of *LNCS*, pages 292–311. Springer, Heidelberg, September / October 2011.
- [OC12] Colin O’Flynn and Zhizhang Chen. A case study of side-channel analysis using decoupling capacitor power measurement with the openadc. In Joaquín García-Alfaro, Frédéric Cuppens, Nora Cuppens-Boulahia, Ali Miri, and Nadia Tawbi, editors, *Foundations and Practice of Security - 5th International Symposium, FPS 2012, Montreal, QC, Canada, October 25-26, 2012, Revised Selected Papers*, volume 7743 of *Lecture Notes in Computer Science*, pages 341–356. Springer, 2012.
- [OC14] Colin O’Flynn and Zhizhang (David) Chen. ChipWhisperer: An open-source platform for hardware embedded security research. In Emmanuel Prouff, editor, *COSADE 2014*, volume 8622 of *LNCS*, pages 243–260. Springer, Heidelberg, April 2014.
- [OC15] Colin O’Flynn and Zhizhang Chen. Synchronous sampling and clock recovery of internal oscillators for side channel analysis and fault injection. *Journal of Cryptographic Engineering*, 5(1):53–69, April 2015.
- [OD19] Colin O’Flynn and Alex Dewar. On-device power analysis across hardware security domains. *IACR TCHES*, 2019(4):126–153, 2019. <https://tches.iacr.org/index.php/TCHES/article/view/8347>.
- [RAB10] Gordon W. Roberts and Mohammad Ali-Bakhshian. A brief introduction to time-to-digital and digital-to-time converters. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 57(3):153–157, 2010.

- [RK10] Kurt Rosenfeld and Ramesh Karri. Attacks and defenses for jtag. *IEEE Design and Test of Computers*, 27(1):36–47, 2010.
- [RTBT19] Xuanle Ren, Francisco Pimentel Torres, R. D. Blanton, and Vítor Grade Tavares. Ic protection against jtag-based attacks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 38(1):149–162, 2019.
- [SGMT18a] Falk Schellenberg, Dennis R.E. Gnad, Amir Moradi, and Mehdi B. Tahoori. An inside job: Remote power analysis attacks on fpgas. In *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1111–1116, 2018.
- [SGMT18b] Falk Schellenberg, Dennis R.E. Gnad, Amir Moradi, and Mehdi B. Tahoori. Remote inter-chip power analysis side-channel attacks at board-level. In *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–7, 2018.
- [Sha00] Adi Shamir. Protecting smart cards from passive power analysis with detached power supplies. In Çetin Kaya Koç and Christof Paar, editors, *CHES 2000*, volume 1965 of *LNCS*, pages 71–77. Springer, Heidelberg, August 2000.
- [SLT16] Sami Saab, Andrew Leiserson, and Michael Tunstall. Key extraction from the primary side of a switched-mode power supply. In *2016 IEEE Asian Hardware-Oriented Security and Trust (AsianHOST)*, pages 1–7, 2016.
- [SMC21] Albert Spruyt, Alyssa Milburn, and Łukasz Chmielewski. Fault injection as an oscilloscope: Fault correlation analysis. *IACR TCHES*, 2021(1):192–216, 2021. <https://tches.iacr.org/index.php/TCHES/article/view/8732>.
- [SMTG23] Kai Schoos, Sergej Meschkov, Mehdi B. Tahoori, and Dennis R. E. Gnad. JitSCA: Jitter-based side-channel analysis in picoscale resolution. *IACR TCHES*, 2023(3):294–320, 2023.
- [SNK<sup>+</sup>12] Alexander Schlösser, Dmitry Nedospasov, Juliane Krämer, Susanna Orlic, and Jean-Pierre Seifert. Simple photonic emission analysis of AES - photonic side channel analysis for the rest of us. In Emmanuel Prouff and Patrick Schaumont, editors, *CHES 2012*, volume 7428 of *LNCS*, pages 41–57. Springer, Heidelberg, September 2012.
- [SOG24] SOGIS. Application of attack potential to smartcards and similar devices. Technical report, Joint Interpretation Library, 2024.
- [SPK<sup>+</sup>10] Jörn-Marc Schmidt, Thomas Plos, Mario Kirschbaum, Michael Hutter, Marcel Medwed, and Christoph Herbst. Side-channel leakage across borders. In Dieter Gollmann, Jean-Louis Lanet, and Julien Iguchi-Cartigny, editors, *Smart Card Research and Advanced Application*, pages 36–48, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
- [Sve20] Engström Sven. A 1.8 ps time-to-digital converter (tdc) implemented in a 20 nm field-programmable gate array (fpga) using a ones-counter encoding scheme with embedded bin-width calibrations and temperature correction. Master’s thesis, Linköping University, Computer Engineering, 2020.
- [ZS18] Mark Zhao and G. Edward Suh. FPGA-based remote power side-channel attacks. In *2018 IEEE Symposium on Security and Privacy*, pages 229–244. IEEE Computer Society Press, May 2018.

- 
- [ZSZF13] Kenneth M. Zick, Meeta Srivastav, Wei Zhang, and Matthew French. Sensing nanosecond-scale voltage attacks and natural transients in fpgas. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, FPGA '13, page 101–104, New York, NY, USA, 2013. Association for Computing Machinery.

## Appendix A: Additional Results

### 7.4 Details of AES Target and Calibration

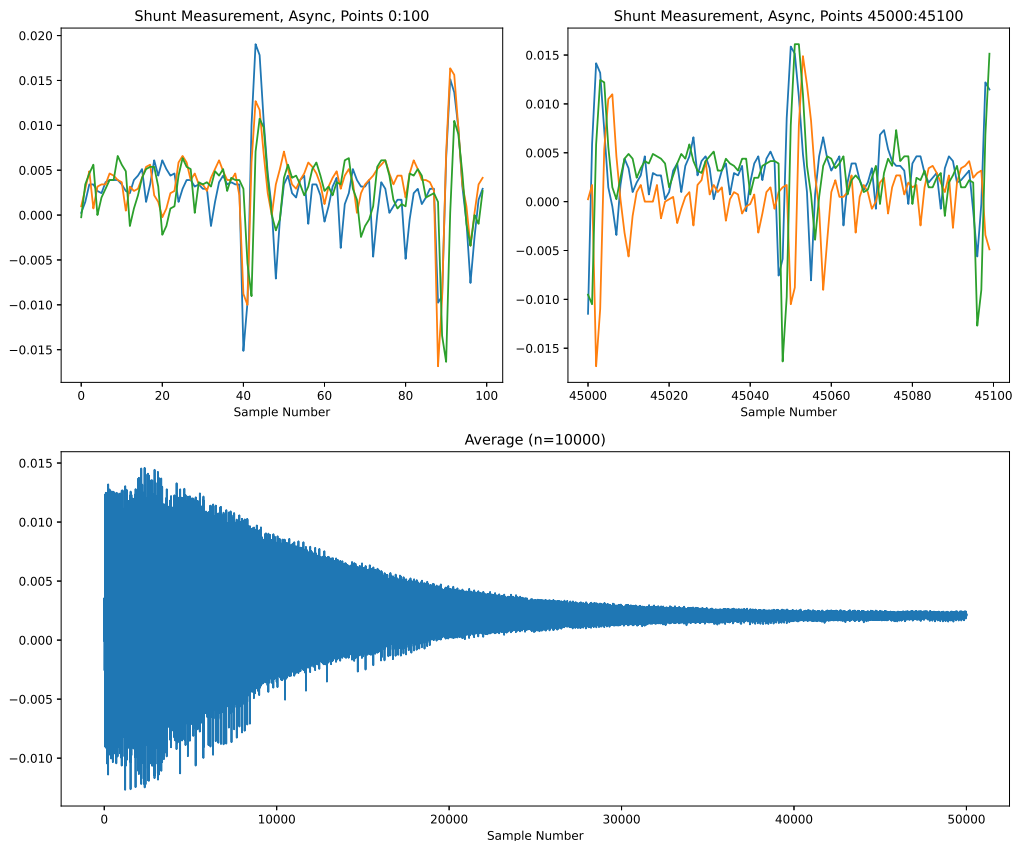
The majority of the results in this paper attack a software AES implemented on the ATSAM4S2A microcontroller. For these examples, the default AES build in ChipWhisperer is used. The specific firmware source and binaries are available in the associated repository for this paper.

Most of the results use the ATSAM4S2A microcontroller plugged into the "CW313" baseboard (the default board for the ChipWhisperer-Husky). The only non-default configuration is the shunt resistor "override" jumper is mounted that shorts out the shunt resistor. Note that the shunt resistor path still has noticeable inductance, so a reasonably strong signal is still available for classic DPA/CPA style power analysis attacks.

The attack scripts and settings for the CPA attack are also included in the open-source repository.

#### 7.4.1 Clock Setup

Most of the results use the internal oscillator with a PLL enabled. This internal oscillator is not as stable as a crystal, and has the problem that longer asynchronous captures naturally become desynchronized. This is partially why results of even the baseline captures seem to take more traces than you would expect for a simple microcontroller.



**Figure 19:** The slight drift in the internal oscillator reduces attack effectiveness.

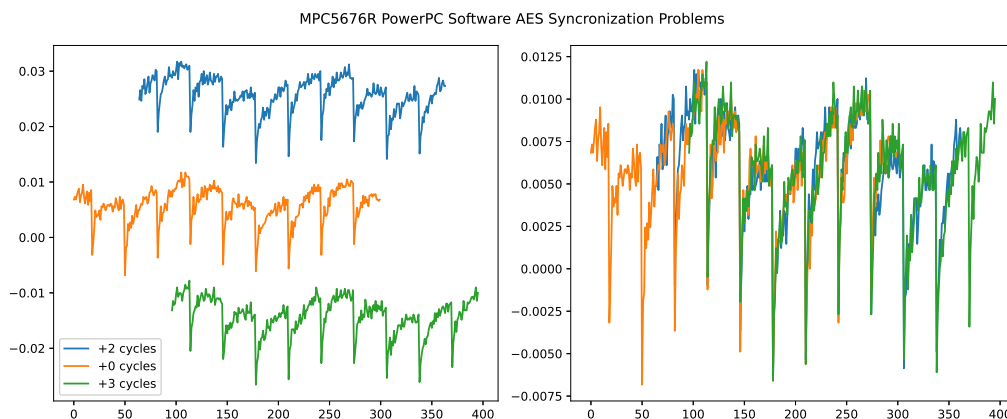


To visualize this, Figure 19 shows the start and end of three power traces on the top left and top right respectively. There are 50K points (samples) in each power trace, and you can see the samples towards the end (points 45000–45100) of the power trace look less aligned than the start (points 0–100). This is made more obvious when looking at the average of all 10 000 power traces in the dataset. The later sample numbers (points) have a reduced average due to the increasing misalignment as we get further from the trigger that occurred at sample 0.

## 7.5 Details of the MPC5676R Multi-TAP Attack

The NXP MPC5676R was used for the multi-TAP attack, as this is a more complex PowerPC based device that includes multiple JTAG TAPs. In addition, the different TAPs and required instructions are well defined in the datasheet from NXP. The microcontroller used in the rest of this paper (the ATSAM4S2A) contains at least two main TAPs (Cortex-M debug TAP & boundary scan TAP), but switching between them is done with an I/O pin, and not using a JTAG instruction.

For this reason the MPC5676R was a better demonstration for the multi-TAP attack. The downside of the MPC5676R is the relatively low TVLA scores. These TVLA scores are partially due to the fact the MPC5676R itself adds some jitter due to timing variations. This can be seen in Figure 20, where it’s seen that the correctly lined up power waveforms (on the right) require adding different cycle offsets. Note that the cycles are lined up exactly (e.g., it’s not sampling jitter) as these are all synchronously sampled. Instead something is non-constant within the MPC5676R execution. This wasn’t explored further and was left for future work.



**Figure 20:** The MPC5676R has non-constant execution cycles.

Running a standard CPA attack on a shunt measurement of the AES implementation requires >5K traces even with optimal synchronization and shunt settings.

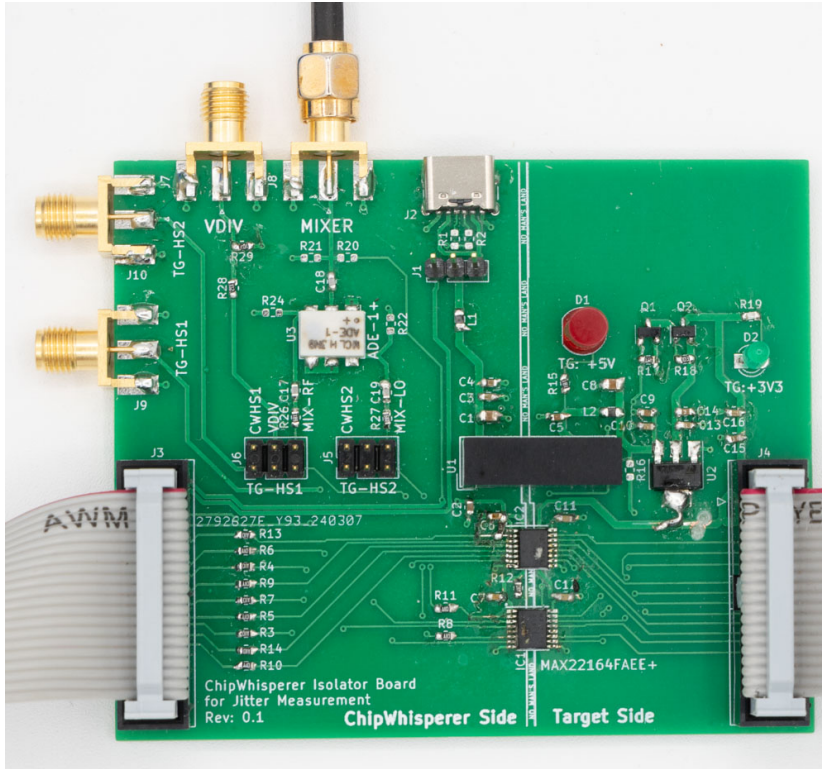
The code used for switching TAPs is included in the open-source repository, and users can test the same sort of attack on other devices which they find have well-defined TAP instructions for switching modes.

## Appendix B: Hardware Details

Although the entire details of hardware used in this work is available in the companion repository (<https://github.com/colinoflynn/phase-modulation-sca>), copies of

schematic and more detailed photos of the PCBs are included in this Appendix for quick reference.

## 7.6 IC-Based Isolator Board



**Figure 21:** IC-based galvanic isolator board.

The schematic is given in Figures 22, 23, 24. Full source and PCB files are available in the repository as well.

Note this board requires a 5V supply. The ChipWhisperer-Husky can be modified with a jumper wire to provide 5V on the 20-pin header, or an external USB connector can be used with the board. The isolator includes a switch so that the isolated power will follow the ChipWhisperer target power pin, so you can power cycle the target from your attack scripts.

To use the ChipWhisperer PLL to clean up a source clock (for captures of Section 3.4):

- Set jumpers on J6 to route TG-HS1 to CWHS1 and MIXRF
- Set ChipWhisperer-Husky to use extclk as PLL source, adjust PLL parameters as needed, and output clock on HS2
- Set jumpers on J5 to route CWHS2 to MIX-LO.

## 7.7 Optical Isolator Board

The optical isolator board uses two isolator modules for transmit and receive:

- AFBR-2624Z

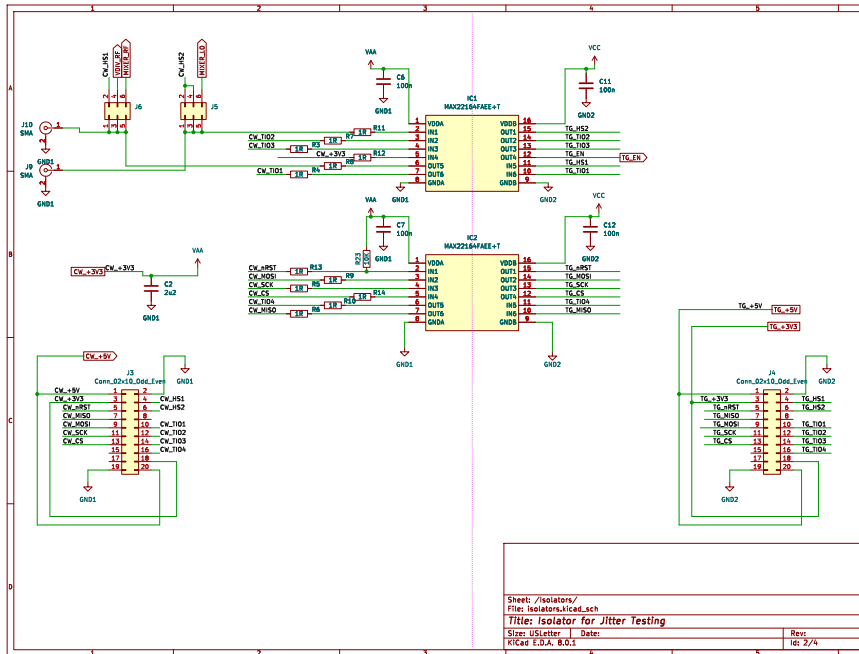


Figure 22: IC-Based Isolator, Headers and Isolator ICs

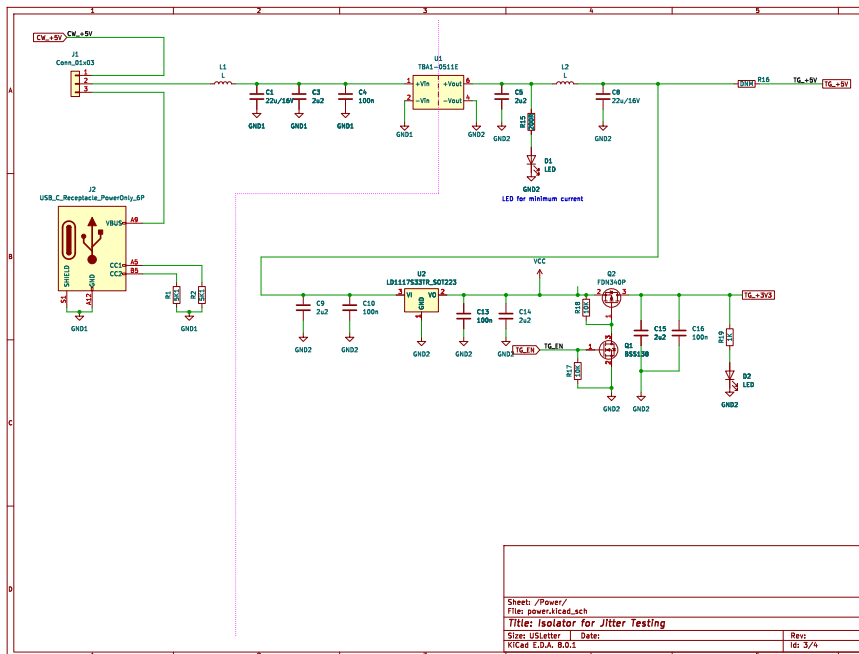
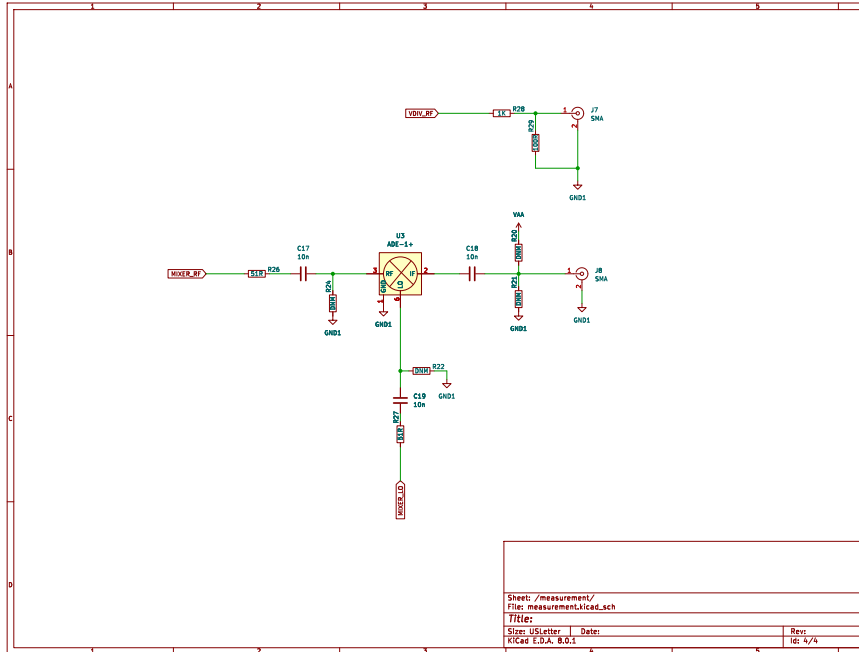


Figure 23: IC-Based Isolator, Power

- AFBR-1624Z



**Figure 24:** IC-Based Isolator, Mixer

These devices are rated up to 50 MBaud, but in practice we found they worked successfully with our 60 MHz clock signal as well. The schematic also has a provision to mounted a higher-speed device (AFBR-59F2Z) which is rated up to 250 MBaud. This device has the disadvantage that it has an automatic sleep mode when a constant signal was detected, which would mean we *couldn't* use it for replicating the 'I/O Pin leakage through an isolator' test from [SPK<sup>+</sup>10], as this required setting the pin to a fixed state.

The higher-speed devices were not mounted on our test boards as the lower-speed ones had sufficient performance.

The optical interfaces are connected with one of these fiber optic cables:

- HFBR-RNS001Z (1m)
- HFBR-RNS002Z (2m)
- HFBR-RNS010Z (10m)

### 7.7.1 ChipWhisperer Side

The ChipWhisperer side includes the mixer along with the optical isolators, and is shown in Figure 25. The schematic is in Figures 26, 27.

To replicate the phase measurement from Section 3.4:

- Set jumpers on J6 to route 50MB HS1 optical input to both CWHS1 and MIXLO.
- Set ChipWhisperer-Husky to use extclk as PLL source, adjust PLL parameters as needed, and output clock on HS2
- Set jumpers on J6 to route CWHS2 to MIXRF.

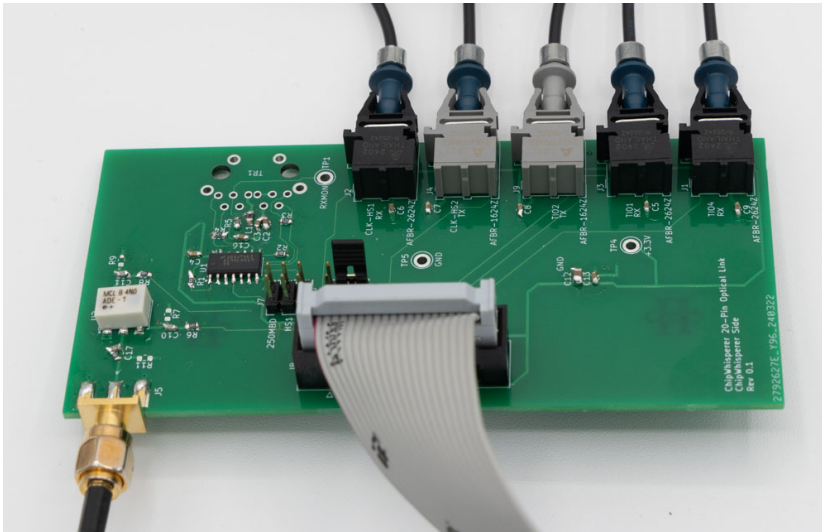


Figure 25: Optical isolator board, ChipWhisperer side.

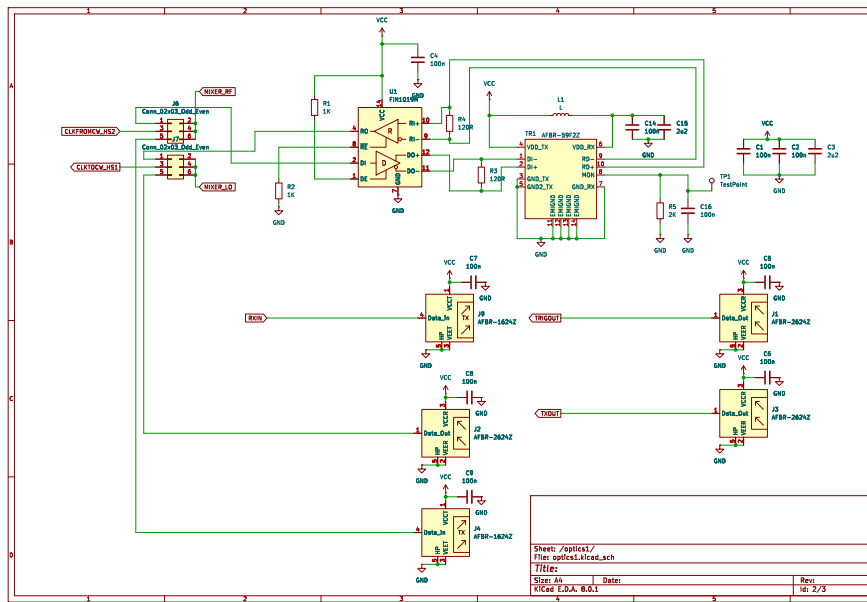


Figure 26: Optical Isolator, ChipWhisperer Side, Optics

### 7.7.2 Target Side

The target side (Figure 28) is relatively simple. Unlike the IC-based isolator there is no provision to automatically turn power on & off to save an additional optical cable. The target is powered by a 5V input, here we just used a 3xAA battery holder which was sufficient to power the SAM4S2A target for long captures. We did not use the faster optical transceiver modules. The schematic is in Figures 29, 30.

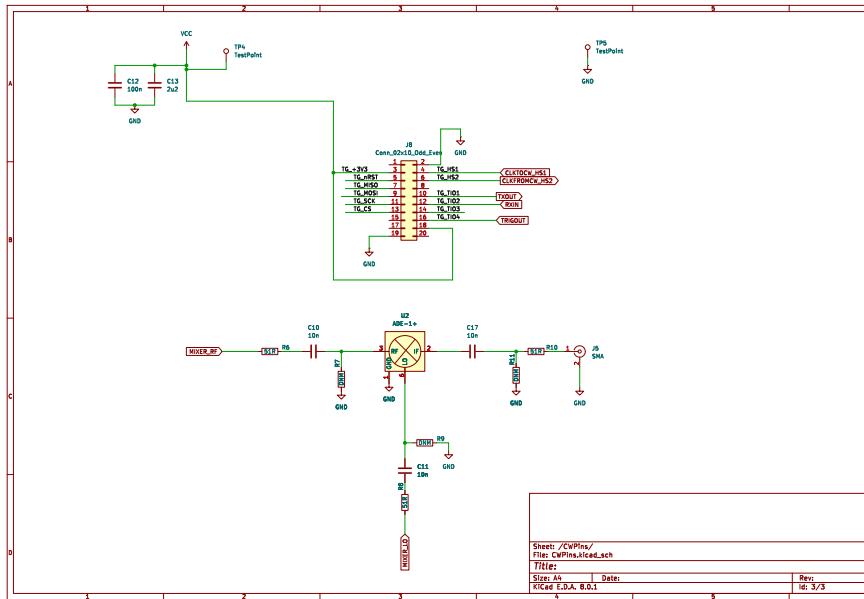


Figure 27: Optical Isolator, ChipWhisperer Side, Mixer & CW

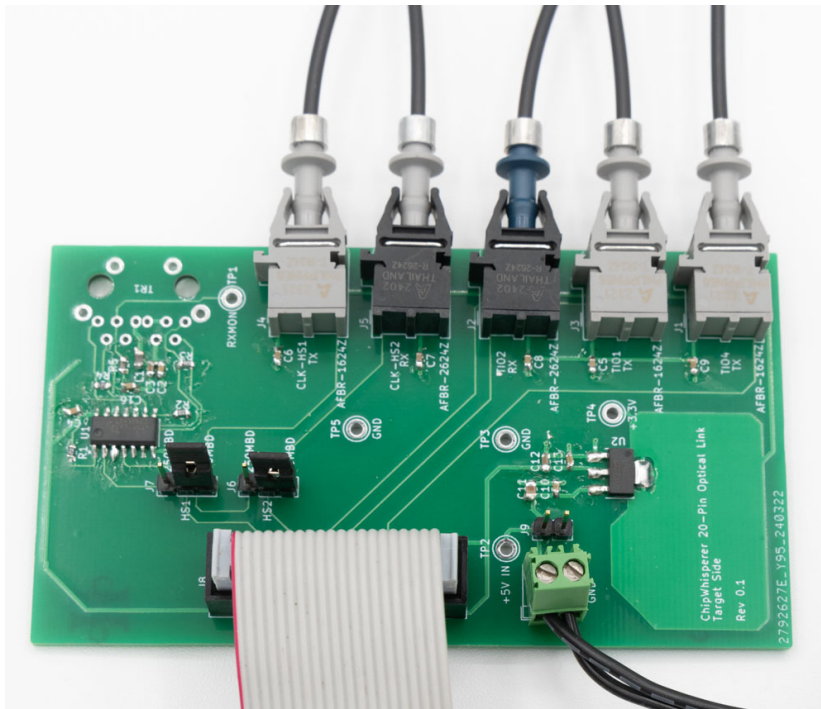


Figure 28: Optical isolator board, Target side.

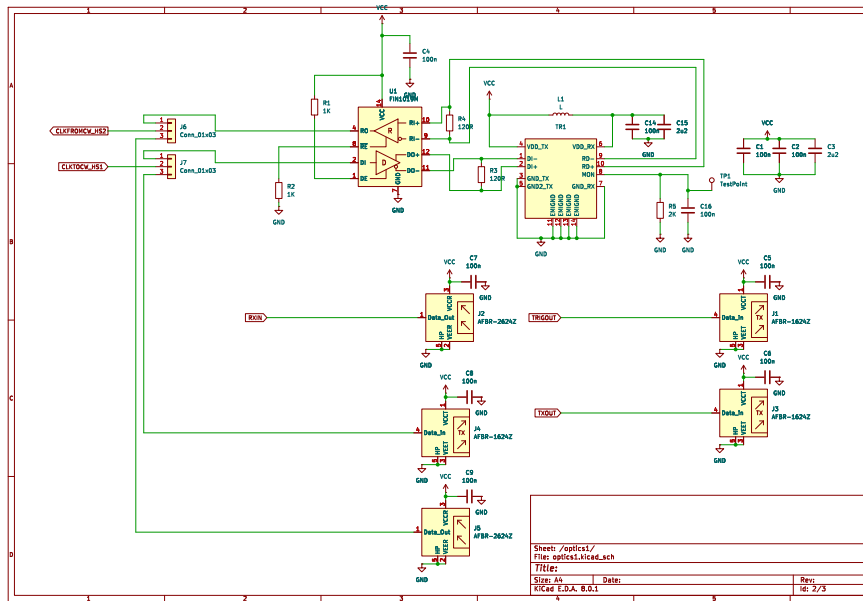


Figure 29: Optical Isolator, Target Side, Optics

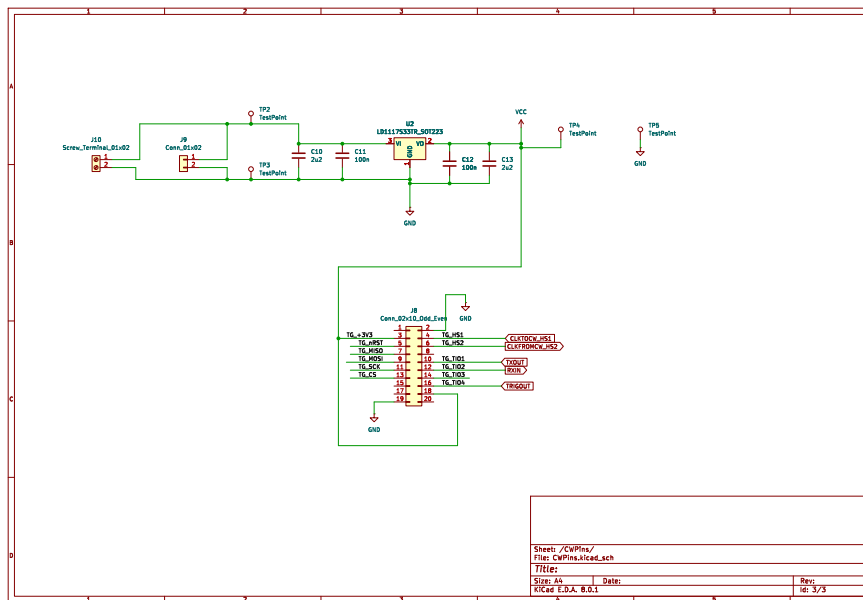


Figure 30: Optical Isolator, Target Side, Power and Connector

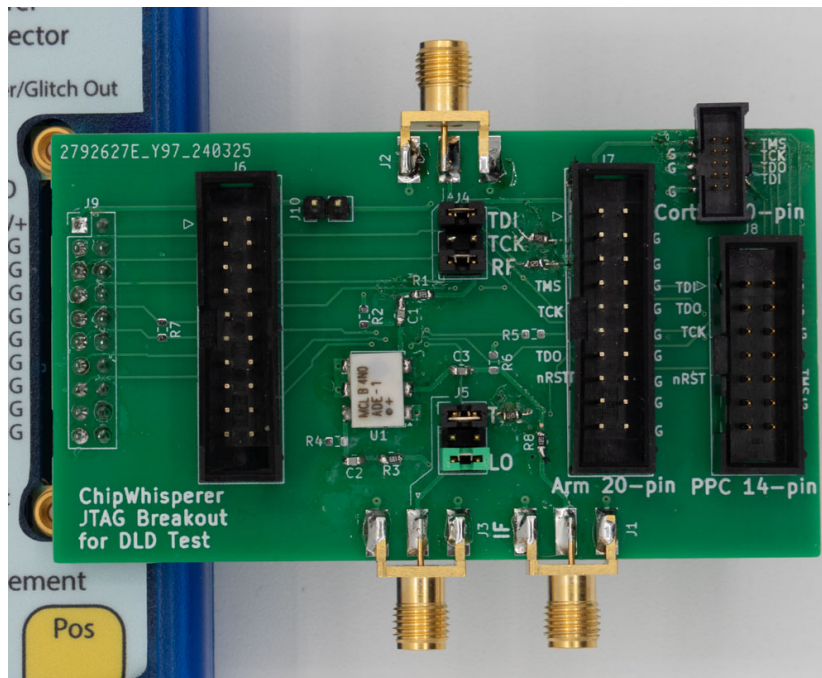
## 7.8 JTAG Attack Board

The JTAG attack board (see in Figure 31) can be used for several purposes:

- Connect the mixer LO & RF inputs to TDO/TDI pins for JTAG bypass based clock phase measurement.
- Connect the mixer LO & RF inputs to SMA pins for other usage of a mixer (e.g., mixer breakout).
- Convert the 20-pin header on the ChipWhisperer to 3 different JTAG standards.
- Adds series termination in the JTAG signals for better signal integrity.

This board was used for both the JTAG PM-side channel attacks (Section 4.1) as well as for the fault analysis work (Section 5). While the board isn't needed for the fault analysis work, we use it still to provide (a) slightly better signal integrity allowing faster JTAG operation, and (b) conversion to other JTAG headers that are common on different development boards.

The schematic is given in Figure 32.



**Figure 31:** JTAG Phase Measurement Board



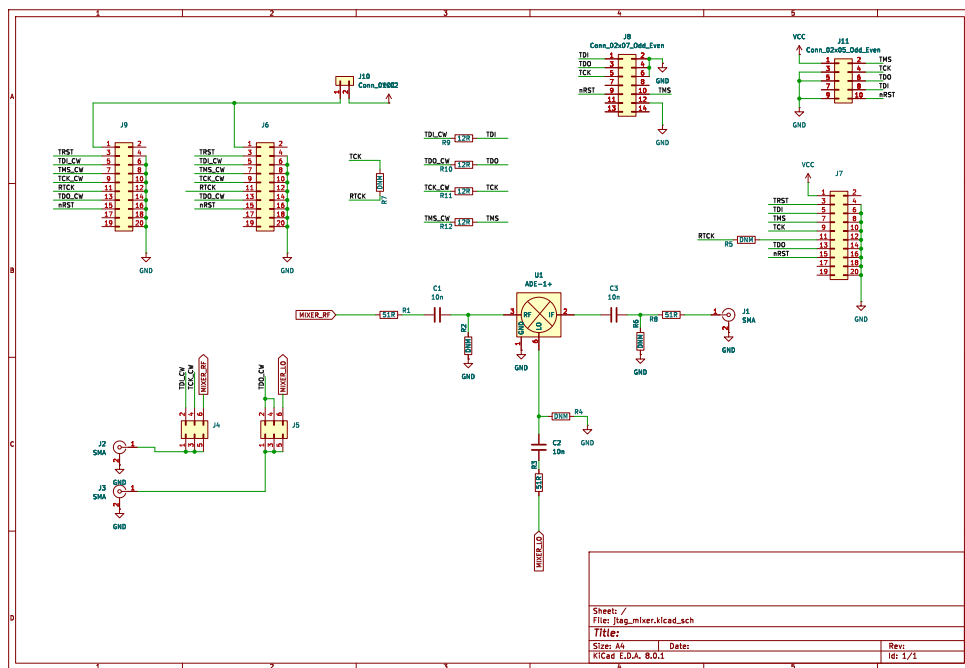


Figure 32: JTAG Phase Measurement Board schematic.

## Appendix C: JTAG Capabilities

The following provides additional details to the “JTAG Always On” capability column in Table 5

### 7.9 ATSAM4S2A

The ATSAM4S2A has a “security bit” that provides debug security. With the security bit set, the JTAG is disabled (no bypass instruction accepted) if the JTAGSEL pin is low. The JTAGSEL pin being low selects the DEBUG/Arm TAP, which would be the normal use (e.g., the JTAGSEL pin is tied low on most products).

If the JTAGSEL pin is high this selects the boundary-scan TAP, which still functions (and also has a BYPASS register/mode) even if the debug security bit is set. In this mode however the processor is held in reset.

### 7.10 STM32{F,G}

The STM32 series have several level of debug security. The most common is three levels of security: RDP0 (lowest), RDP1, and RDP2 (highest).

Bypass mode is disabled in RDP2. The reference manual specifies that the boundary scan TAP & debug TAP are in series, so when the debug TAP is disabled it disables the entire chain.

While glitch attacks are known against RDP2 to transition to RDP1 (which has JTAG enabled, but prevents code read-back), this requires physical access to the device that would also imply a side-channel attack could be done with a shunt measurement or an EM probe.

### 7.11 MPC5676R

The MPC5676R has a “censored” mode to provide debug security.

The NXP MPC5676R reference manual, section 6.5.7, specifies that: **When a device is in a censored state, the debug port (JTAG/Nexus) is disabled and only JTAG BSDL commands can be used. Access to the Nexus/JTAG clients on a censored device requires inputting the proper password into the JTAG Censorship Control Register during reset.**

We have confirmed that the bypass command is still functional. Note that some of the taps from Table 6 would be inaccessible in censored mode, but at minimum the standard BSDL bypass tap is always available.

### 7.12 MPC5777C

The MPC5777C is a more complex device than the MPC5676R, but the core censorship logic is only around the debug TAP (and not boundary scan TAP).

With a censored MPC5777C device the JTAG port was still functional for bypass mode.

### 7.13 MK24F

The K24P144M120SF5RM reference manual in section 8.3.3 specifies that: **When flash security is active the JTAG port cannot access the memory resources of the MCU. Boundary scan chain operations work, but debugging capabilities are disabled so that the debug port cannot read flash contents.**

We confirmed that the boundary scan (bypass mode) still works when "locked" (the flash security is the only code security feature available). Note the JTAG port is normally used to unlock the device so this would be expected to work.

### **7.14 Artix 7 and Spartan 6**

The Artix 7 contains options to prevent readback or reconfiguration, along with options to force usage of an encrypted bitstreams. Even after all security eFuses are enabled the JTAG port still works (as expected bitstream programming would not work without the correct key, but bypass mode still worked).

The older Spartan 6 device is similar to the Artix 7, and also does not offer a JTAG disable mode. Security features are focused only on disabling loading (or readback) of bitstreams, the JTAG port always remains accessible.