Chypnosis: Undervolting-based Static Side-channel Attacks

Kyle Mitard
Worcester Polytechnic Institute
krmitard@wpi.edu

Saleh Khalaj Monfared
Worcester Polytechnic Institute
skmonfared@wpi.edu

Fatemeh Khojasteh Dana
Worcester Polytechnic Institute
fdana@wpi.edu

Robert Dumitru
Ruhr University Bochum &
The University of Adelaide
robert.dumitru@adelaide.edu.au

Yuval Yarom
Ruhr University Bochum
yuval.yarom@rub.de

Shahin Tajik
Worcester Polytechnic Institute
stajik@wpi.edu

Abstract*—Static side-channel analysis attacks, which rely on a stopped clock to extract sensitive information, pose a growing threat to embedded systems' security. To protect against such attacks, several proposed defenses aim to detect unexpected variations in the clock signal and clear sensitive states. In this work, we present *Chypnosis*, an undervolting attack technique that indirectly stops the target circuit clock, while retaining stored data. Crucially, Chypnosis also blocks the state clearing stage of prior defenses, allowing recovery of secret information even in their presence. However, basic undervolting is not sufficient in the presence of voltage sensors designed to handle fault injection via voltage tampering. To overcome such defenses, we observe that rapidly dropping the supply voltage can disable the response mechanism of voltage sensor systems. We implement Chypnosis on various FPGAs, demonstrating the successful bypass of their sensors, both in the form of soft and hard intellectual property (IP) cores. To highlight the real-world applicability of Chypnosis, we show that the alert handler of the OpenTitan root-of-trust, responsible for providing hardware responses to threats, can be bypassed. Furthermore, we demonstrate that by combining Chypnosis with static side-channel analysis techniques, namely laser logic state imaging (LLSI) and impedance analysis (IA), we can extract sensitive information from a side-channel protected cryptographic module used in OpenTitan, even in the presence of established clock and voltage sensors. Finally, we propose and implement an improvement to an established FPGA-compatible clock detection countermeasure, and we validate its resilience against Chypnosis.

1. Introduction

Physical side-channel attacks (SCA) can undermine the security of cryptographic implementations on integrated circuits (ICs). These attacks typically exploit the inevitable influence of data transitions during computation on current consumption or voltage drop on a chip. Dynamic side-channel attacks, such as power [23] and electromagnetic analysis [2], can exploit such data transitions and recover the secret from the chip. Recently, however, static physical SCA attacks have been gaining attention, in which adversaries can extract static data stored in memories, such as Flip-Flops (FFs). Examples of such static attacks include static power analysis [35], Laser Logic State Imaging (LLSI) [24], Impedance Analysis (IA) [31], and Thermal Laser Stimulation (TLS) [25].

Such static attacks require some level of tampering with the clock and voltage of the target chip. First, the attacker must freeze the circuit's state by halting its clock, because the time required for recovering the static data stored in registers is significantly longer than the clock period [24, 31]. Second, in some of these static attacks, known as backscatter attacks (e.g., LLSI and IA), the adversary must modulate the voltage supplying the chip to produce a detectable modulated reflection during laser or microwave stimulation. In these attacks, the adversary stimulates the chip using external signals (e.g., near-infrared laser beams for LLSI or microwave radiation for IA) and measures the modulated reflections to infer the internal circuit state or memory contents. Due to their active nature, static backscatter attacks often achieve a higher Signal-to-Noise Ratio (SNR) and, in some cases, can extract secrets with a single trace, rendering data randomization techniques such as masking ineffective [24, 31].

Consequently, detecting clock or voltage tampering and then responding by wiping the sensitive data is a promising countermeasure against these static attacks. Various clock and voltage sensors, both as soft intellectual property (IP) cores [13, 14] and hard IP cores [27, 48], have been developed for this purpose. Moreover, specialized sensors targeting specific threat vectors, such as voltage glitching [49, 62] and laser probing [33, 53, 60], can also serve as effective countermeasures against backscatter attacks. Additionally, the assumption that an attacker can access and manipulate the system clock is often unrealistic in real-world scenarios. Many secure ICs rely on internal clock sources [27]

This article has been accepted for publication in *IEEE Symposium on Security and Privacy 2026*.

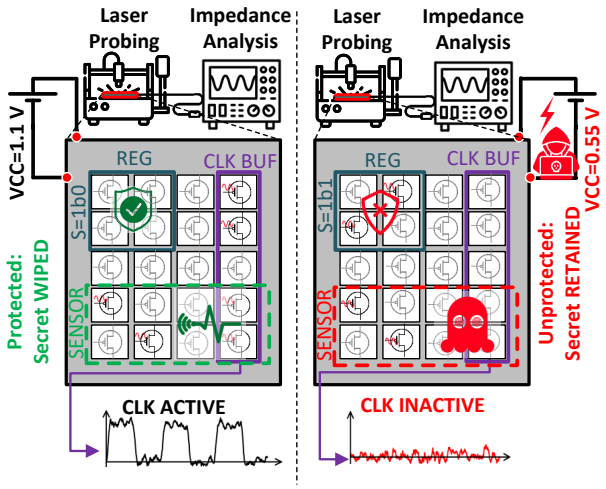


Figure 1: High-level overview of Chypnosis Attacks

for cryptographic operations, making external clock control extremely difficult for an adversary.

Driven by these observations, in this paper, we ask the following questions: *(1) Is it possible to halt the system's clock without tampering with its source? (2) Can we perform static side-channel attacks without triggering clock and voltage sensors?*

1.1. Our Contribution

In this work, we positively answer both questions. We introduce *Chypnosis* attacks, in which an adversary bypasses clock and voltage sensors, places a chip in hibernation while retaining its data. In this condition, the adversary can perform a static side-channel attack to recover the retained data. Our attack exploits the observation that rapidly lowering the supply voltage below nominal thresholds induces a brownout condition, where logic components (e.g., state machines and clock buffers) cease switching, but volatile memory elements, such as SRAM and flip-flops, continue to retain data. Figure 1 presents an abstract overview of our attack.

We perform Chypnosis on both SRAM-based and Flash-based FPGAs fabricated using 28 nm processes. First, we conduct extensive experiments to determine the voltage thresholds required to induce hibernation at various clock frequencies. Next, we demonstrate that entering this brownout condition effectively halts the clock and suppresses switching activity without requiring direct control over the clock source. We confirm this behavior through electrical and optical measurements. We further show that a switching freeze caused by a rapid voltage drop disables the response circuitry of clock, voltage, and brownout detection (BOD) sensors on FPGAs, irrespective of being implemented as soft IPs [13, 14] or hard IPs [27]. Hence, the sensitive data is not wiped and remains on the chip.

To demonstrate Chypnosis's effectiveness in practical scenarios, we successfully attack the FPGA implementation of OpenTitan Earl Grey and bypass its *alert handler* [42], which is responsible for providing hardware responses to threats. Finally, we demonstrate that even in the brownout state, it is possible to delicately modulate the supply voltage without crashing or waking the system, thereby successfully performing LLSI and IA attacks. We demonstrate the extraction of secret data from a side-channel-protected symmetric cryptographic module, used in OpenTitan, in a single trace. We conclude by proposing a circuit-based FPGA-compatible countermeasure, which successfully mitigates Chypnosis.

2. Technical Background

2.1. Conventional Static SCA Countermeasures

2.1.1. Detection-based Countermeasures.

Clock Sensors. The primary requirement for launching a static SCA attack is to stop the system clock. Hence, if we have a sensor that detects that the clock has been halted for a while, it can trigger a response, such as wiping sensitive data, before an adversary can recover it. Some secure IC families are equipped with internal clock sensors capable of detecting anomalies in clock behavior [27]. However, the specific design details of these sensors are proprietary. FPGAs also contain internal circuitry, such as phase-locked loops (PLLs), that can detect irregularities in the clock waveform; however, they must be explicitly configured by the user to function as a security sensor. Commercial solutions, such as the AMD/Xilinx Security Monitor IP core [57], offer built-in clock monitoring capabilities; however, these features are typically available only to specific customers. Several attempts have been made in the literature to design clock freeze detection sensors in the form of soft IPs for FPGAs. For instance, Farheen et al.[14] proposed using internal clock oscillators to monitor the integrity of external clocks. More recently, Dumitru et al.[13] introduced two sensor variants that detect clock freezing without relying on any internal clock sources.

Voltage Sensors. Similar to clock sensors, many secure ICs are equipped with voltage sensors to detect voltage tampering [27]. On FPGAs, analog-to-digital converters (ADCs), such as the XADC available in AMD/Xilinx FPGAs [58], provide built-in voltage sensing capabilities. These sensors can monitor both internal and external voltages, converting analog signals into digital values that can be processed by user-defined digital logic on the FPGA. In addition to built-in voltage sensors, FPGA users can also configure their own delay-based ADCs, such as ring oscillators (ROs) [61] and time-to-digital converters (TDCs) [62], on the FPGA.

Laser Sensors. For specific backscatter-based attacks, such as LLSI, sensors capable of directly detecting incident laser beams have also been investigated. Similarly to voltage tampering detection, ADCs have demonstrated sensitivity to localized temperature increases caused by laser irradiation. As a result, ROs [53] and TDCs [33] have been utilized to detect laser probing attacks on FPGAs.

2.1.2. Response-based Countermeasures. An often overlooked aspect of countermeasures is the system's response after such powerful attacks are detected. Although the conventional assumption is that sensitive data can simply be zeroized upon detection, this may not be suitable in many real-world scenarios. First, zeroization itself can cause dynamic side-channel leakage [13], such as through power side-channels that reveal Hamming weights. To mitigate this, operations such as masked clear are employed, in which sensitive register contents are overwritten with random values [13]. Second, sometimes continued system operation is desired, which could become impossible after zeroization. In such cases, schemes such as Moving Target Defense (MTD) can be employed to mitigate the threat without interrupting the system's operation. For instance, randomizing the placement and routing of a circuit on an FPGA via partial reconfiguration has been shown to be effective against LLSI and IA attacks [32, 33].

2.2. Brownout Condition

When the source voltage of a transistor exceeds a certain threshold, the transistor effectively functions like a switch, responding to changes in the gate voltage. Reducing the supply voltage, commonly referred to as voltage scaling, has been widely used to improve the energy efficiency of microprocessors. The lower bound for voltage scaling is typically constrained to around half of the nominal operating voltage [17]. However, it has been demonstrated that standard CMOS logic gates can operate effectively even below the threshold voltage. Based on these observations, prototype designs have shown that by carefully replacing analog-like components with standard digital switching elements, it is possible to push voltage scaling into the subthreshold region and extend the traditional limits of low-voltage operation [52, 59]. On commercial FPGAs, however, subthreshold operation is hard to achieve due to the high energy consumption of conventional FPGA interconnects. While multiple research proposals have been proposed to enhance the performance of subthreshold FPGAs by optimizing interconnect drivers and operating them in the near-threshold voltage region [16, 47, 50], they have yet to be realized on commercial FPGAs.

As the supply voltage drops in FPGAs, a brownout condition occurs where the transistors cannot drive the capacitive loads at the gates of other transistors. Consequently, clock buffers and PLL circuits will also cease to function, and the clock signal distribution will halt. Meanwhile, memory cells, such as SRAM cells and flip-flops (FFs), may fully retain their data, as their Data Retention Voltage (DRV) [8, 20] is typically lower than the logic operating threshold. For instance, in the case of SRAM-based FPGAs, a brownout condition causes the stoppage of all switching activities within configuration logic blocks (CLBs) and switch boxes. At the same time, the SRAMs and flip-flops (FFs) retain the FPGA configuration and user data. If the voltage drops further, memory cells will lose their content, and thus, the FPGA will crash and require reconfiguration. Therefore, a narrow subregion within the subthreshold operating range, above the data retention voltage (DRV), exists where the FPGA enters a "hibernation" state. Furthermore, an FPGA can wake up from hibernation by raising the voltage back above the threshold voltage and resume operation as usual.

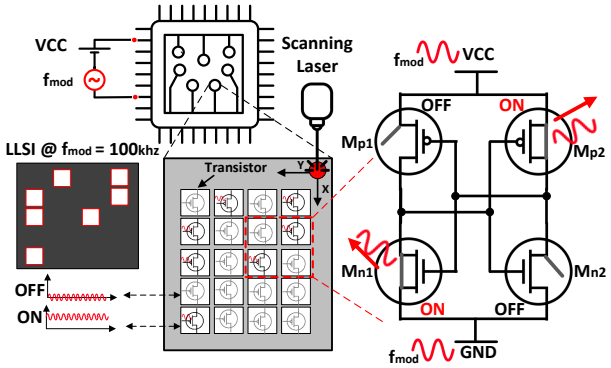


Figure 2: High-level overview of LLSI attack

2.3. Laser Logic State Imaging (LLSI)

Laser Logic State Imaging (LLSI) [41] is a static side-channel attack method. LLSI is a subset of optical probing attacks, in which a near-infrared laser is focused on the transistors from the backside of an IC, and its reflection becomes modulated by the gate or drain of a transistor during switching activity. This modulated reflection can be analyzed in two main ways. In the first, known as Laser Voltage Probing (LVP), the attacker repeatedly samples the reflection at a single point to reconstruct a waveform of the processed data. In the second, known as Laser Voltage Imaging (LVI), the laser is scanned across a region of interest while a spectrum analyzer filters out modulations at a specific frequency. In a typical LVI setup, the objective is to identify transistors switching at a given frequency and generate a 2D activity map highlighting regions operating at that frequency. Combined with LVP, these techniques can reveal internal signal activity, provided that the signals are not static [11, 54].

LLSI builds upon the LVI technique by enabling the probing or imaging of static signals. By modulating the power supply at a known frequency, as illustrated in Figure 2, the voltage at the transistors that are in ON states will also be modulated, generating a measurable LVI signal. In contrast, transistors in the OFF state do not produce a significant signal. As a result, the logic state of a memory cell can be inferred based on the observed LVI activity. Figure 2 presents a simplified example of an LLSI image of an SRAM cell composed of two cross-coupled inverters. LLSI has been successfully used to extract data from registers in FPGAs [24, 25] and SRAM cells in microcontrollers [10, 22].

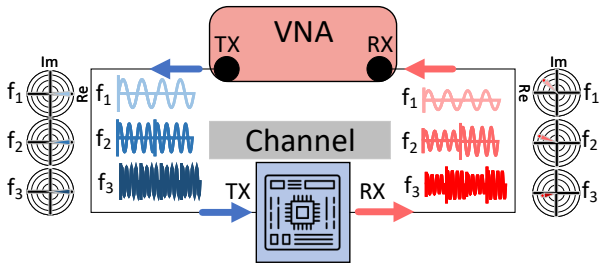


Figure 3: High-level illustration of impedance attack

2.4. Impedance Analysis (IA)

Impedance Analysis (IA) [31] is another static side-channel attack that can recover secret data by measuring changes in the impedance of an IC's power delivery network (PDN). The key idea is that the temporary contents of registers and their corresponding wiring influence the IC's PDN impedance, which leads to changes in how electrical signals with various frequencies reflect or transmit through the IC's PDN. To measure such reflections and transmissions, the attacker uses a Vector Network Analyzer (VNA) to inject high-frequency sine waves into the IC's power rails and then captures the so-called scattering parameters. Different regions of an IC respond differently at various frequencies [36], and thus, by sweeping across a range of frequencies, the attacker can essentially probe multiple localized areas of the chip simultaneously without needing physical access to specific wires or locations.

Figure 3 illustrates a high-level overview of an impedance attack. The attack process is conceptually similar to channel estimation in wireless communications, where reference radio frequency (RF) signals are transmitted through a channel and the received signals are analyzed. In the case of impedance attacks, the "channel" corresponds to the PDN of the target system, while the transmitter and receiver are the ports of a VNA. RF waves are injected into the PDN at specific frequencies and amplitudes, and the responses are captured at the receiver with amplitude and phase modulations introduced by the circuit's internal state. By analyzing these modulated parameters, an attacker can extract secret information. Prior work has demonstrated the effectiveness of this approach for attacking both protected and unprotected cryptographic implementations [31], as well as for reverse engineering purposes [6].

3. Threat Model

In our threat model, we assume that the adversary has physical access to the target device. We consider profiling (template) attack scenarios in which the adversary can profile a training device before launching an attack on the actual target. During the attack phase, we assume that all detection- and response-based countermeasures described in Section 2.1 are active. We further assume that the security sensors' internals may be unknown to the attacker. The adversary does not have access to the system's clock source; however, to read the contents of registers at a specific clock cycle, she must halt the clock. We also assume the adversary can access and manipulate the IC's core voltage supply rails and remove the decoupling capacitors on the printed circuit board (PCB). Such tampering attempts require some knowledge of the PCB schematic, which can be obtained through documentation, visual analysis, or multimeter testing. Moreover, the adversary can capture snapshots of the hardware state using techniques such as LLSI or IA and subsequently recover the values stored in registers. For IA, the attacker must have access to VNAs and function generators, which are standard RF instruments. For LLSI, failure-analysis equipment can typically be rented for a few hundred dollars per hour rather than purchased outright. For secret extraction, we consider a template attack threat model: (i) LLSI requires localization of target registers, achievable through reverse-engineering or prior design knowledge; (ii) IA requires knowledge of the cryptographic algorithm, masking scheme, and key size. We also assume the adversary has some knowledge of the system's clock frequency to estimate the brownout voltage thresholds.

To understand how an adversary might benefit from such an attack in the real world, we can consider the following examples. One example would be the decryption core on FPGAs or microcontrollers/microprocessors, which is programmed with a cryptographic key. Such decryption engines can be used, for example, to decrypt the device's bitstream or firmware. By extracting the secret key, the adversary can clone, reverse-engineer, and tamper with the design contained in the bitstream or firmware. Moreover, if the same key is used for multiple ICs in the field, the attacker can compromise the security of other ICs that use the same key.

4. Experimental Setup

4.1. Devices under Test (DUT)

To test undervolting effects on the clock and voltage sensors, we used two chip families: AMD/Xilinx 7 Series FPGAs and Microchip PolarFire SoC FPGAs, both manufactured using 28 nm technology. While the latter has dedicated hard IP (ASIC) clock and voltage sensors [27], the former has only a hard IP voltage sensor [48], and the clock sensor should be configured as a soft IP (e.g., [13]).

4.1.1. AMD/Xilinx FPGAs. We used NewAE CW305 boards [38], which have an AMD Artix-7 FPGA (part number XC7A100T-FTG256). We also used a ChipWhisperer CW310 Bergen Board [39], which has an AMD Kintex-7 flip-chip FPGA (part number XC7K410T-FBG676). We selected these boards primarily because they both provide direct access to the FPGA's PDN. Furthermore, the OpenTitan design is compatible with the CW310 Bergen Board. We focused on the V_{CCINT} power rail, as it directly powers the FPGA's core logic and registers. Moreover, these boards do not contain any decoupling capacitors on the core voltage PDN, making a rapid voltage drop feasible.

4.1.2. Microchip FPGA SoCs. We used a Microchip PolarFire Discovery Kit board, which contains a Microchip PolarFire SoC FPGA (part number MPFS095T-1FCSG325E) [29]. We focused on the V_{DD} power rail, as it directly powers the SoC FPGA's core logic and registers. For a faster voltage drop, we removed decoupling capacitors C152, C162, C161, C144, C140, C134, and C169, which together account for 491.1 μF, leaving 0.5308 μF of decoupling capacitance on V_{DD}. To gain control over V_{DD}, we removed inductor L10 and supplied V_{DD} through the test point adjacent to the inductor. We soldered a BNC to a jumper lead cable to the V_{DD} test point (+) and the ground pad of C144.

4.2. Optical Setup

To perform optical probing and photon emission analysis, we used a Hamamatsu PHEMOS-X FA microscope [19]. The system is equipped with a 1.3 µm High-power Incoherent Light source (HIL) for optical probing. It supports objective lenses with 5x/0.14 NA, 20x/0.4 NA, and 50/0.76 NA magnifications, along with additional scanner zoom up to 8x.

In LVI/LLSI mode, the laser is scanned across the device's surface using galvanometric mirrors. The reflected light is separated via semi-transparent mirrors and directed to a photodetector. Its output is passed through a bandpass filter tuned to the frequency of interest. The measured signal amplitude at each scan location is mapped to its corresponding spatial position, forming a grayscale-encoded 2D image.

Photon emission is another backside failure analysis technique that captures weak light emissions from the chip's transistors with long exposure times (typically 5 s or more). When current flows through a P-N junction, it can emit a small number of photons due to energy level transitions. For this experiment, we used the PHEMOS-X InGaAs camera operating at –70°C with a 20x lens to observe photon emission from a 15-stage ring oscillator implemented on the CW310. This technique was employed to monitor dynamic internal signals, which we expect to become static upon entering hibernation.

4.3. Side-channel Attack Setups

We used the CW305 and CW310 boards for all side-channel attacks. Figure 4 depicts the high-level diagram of our experimental setup for IA and LLSI, where the measurements are carried out with an external controller.

4.3.1. LLSI Attack Setup. For LLSI, we used the same high-level procedure as shown in Figure 4. We desoldered the bridge between TP20 and TP19 on the CW310, which cuts off V_{CCINT} from the onboard voltage regulator. In its place, we connect one channel of a BK Precision 9130 power supply to the SMA connector at J3 (VCCINT_SHUNTLO). This is because the onboard voltage regulator cannot supply a DC voltage that is low enough to hibernate the FPGA.

The AC modulation comes from a Tektronix AFG3021 single-channel function generator, capacitively coupled

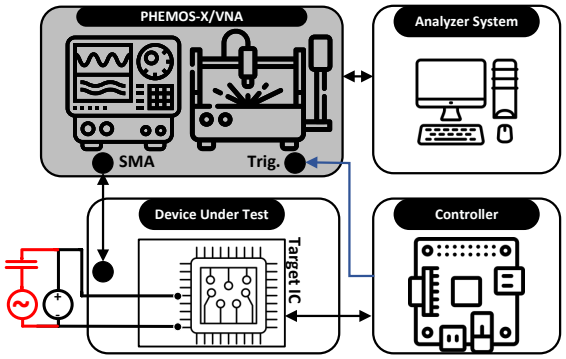


Figure 4: LLSI and IA Setup. Blue parts are used only in IA, red only in LLSI, and the rest are common to both.

through a 10 μF electrolytic capacitor soldered to J23 pin 2 on the same side of the shunt as the DC supply. We use a 100 kHz sinusoidal modulation with an amplitude of roughly 25 mV, as measured where the capacitor is connected to V_{CCINT}. Through trial and error, we found that to be the highest amplitude that would not crash the DUT by bringing the FPGA out of hibernation and into cutoff. We achieve this amplitude at the input by setting the function generator output to 1 V_{pp} to compensate for impedance mismatch.

We programmed the DUT with registers clocked at 10 MHz with an on-chip MMCM. We can independently set each register to a constant 1, a constant 0, or flip with every clock cycle via a USB serial port through an Arduino (i.e., the controller) connected to the PMOD connectors.

4.3.2. IA Attack Setup. To control the state of the target FPGA on the CW305 board during IA, we used a NewAE CW-Lite board [40]. It facilitates serial communication with the DUT and acts as an intermediate controller for transferring plaintext and receiving ciphertext from the target IC during profiling. The V_{CCINT} voltage on the CW305 can be controlled through USB and an onboard programmable voltage regulator using Python APIs. We used a Keysight ENA Network Analyzer E5080A [21] for our measurements, which supports RF/microwave scattering analysis across frequencies ranging from 9 kHz to 6 GHz. To ensure reliable signal transmission, we used Minicircuit CBL-2FT-SMNM+ shielded characterization cables [30].

After loading a bitstream into the target FPGA, arbitrary input data (e.g., plaintexts and keys) are generated by the Analyzer System and sent to the Controller, as shown in Figure 4. The controller transmits this data to the target IC via a serial interface. At a designated timestamp, the system drops the voltage to pause the clock; then, the controller triggers the VNA to capture a measurement. The VNA then collects the trace and returns it to the Analyzer System.

The VNA measurement parameters in our analysis are determined experimentally. The intermediate frequency (IF) bandwidth is set to 500 Hz, and the averaging factor is configured as $N_{\text{Avg}} = 400$ to reduce the measurement noise. We configure the VNA to perform a single-port measurement of S_{11}, the reflection scattering parameter, which provides a

linear estimation of the impedance profile. Although both amplitude and phase values of the scattering parameters are recorded, we use only the phase component in our impedance analysis due to its noise resiliency [36]. All measurements are performed differentially using a reference program, and the VNA output power is set to 10 dBm. Upon completion of the computation on the target IC, the controller receives the ciphertexts and forwards them back to the Analyzer System over a serial connection for verification.

5. Circuit Operation During Hibernation

5.1. Hibernation Voltage Characterization

5.1.1. Undervolting Voltage-Frequency Scan. To characterize the resilience of FPGA targets under voltage stress, we developed a systematic *Hibernation Scan* methodology that explores the functional limits of an FPGA's internal logic as supply voltage decreases across a range of clock frequencies. This method reveals the threshold conditions under which the FPGA ceases to reliably perform core operations such as register assignment and clock-driven state transitions. The procedure is executed through a software controller that communicates with an on-chip UART-based hardware test module implemented in RTL within the target FPGA DUT. Crucially, this UART module is on a separate voltage rail to the FPGA internal logic, so it remains functional during our undervolting experiments.

Hardware Instrumentation. The FPGA on the CW305 board is configured with a test logic block that performs two key functions during a timed evaluation window of undervolting. First, a known `register assignment`, e.g., `reg_out <= 8'h88`, that allows for deterministic verification of data latching under voltage stress is executed. Then, a `clock counter` that increments on every rising clock edge, validates whether the internal clock network and FFs remain operational.

Initialization Procedure. On the host side, the undervolting scan is orchestrated via a high-level script. The scan proceeds by sweeping a grid of `(frequency, voltage)` pairs. The PLL is configured to the desired frequency f at the supply voltage v.

Furthermore, we deploy a Debug Register Reset routine at RTL level that resets all registers to a known baseline value. Before each iteration of the sweep, two time parameters are set. t_d as the FPGA initialization delay before evaluation (the delay incurred due to UART communication for send commands), and t_t as the duration in which the FPGA is undervolted.

Undervolting Phase. When the test is initialized, the supply voltage is simultaneously lowered to v via the programmable on-board power supply. The chip is allowed to run in the undervolted state for a minimum of $t = 0.8$ seconds, ensuring that the on-chip execution occurs during the undervolted state ($t > t_d + t_t$). After the undervolting phase, the voltage is restored to a nominal level (typically 1.0 V), and the contents of the test registers are read back.

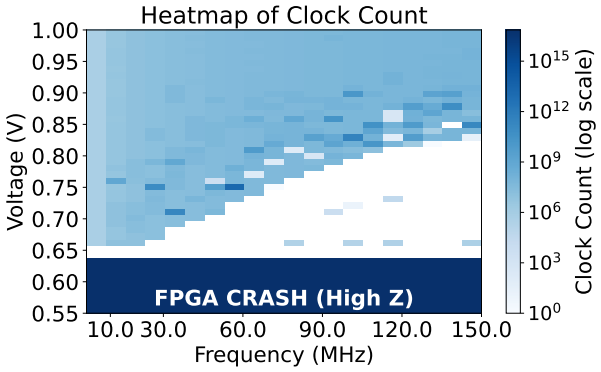


Figure 5: Heatmap of the `clock_count` during undervaluing with different working clock frequencies. White spots highlight the hibernation voltage.

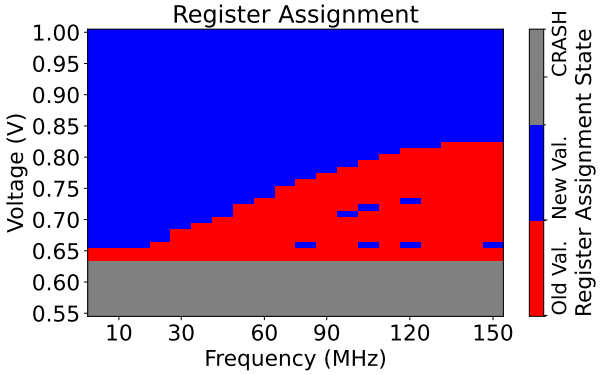


Figure 6: Heatmap of the `reg_assign` during undervaluing with different working clock frequencies.

The register assigned value (`reg_assign`) and the clock counter state (`clock_count`) from FPGA are extracted via UART. Moreover, we verify system stability by performing the hardware sanity test. If the FPGA does not return to a clean baseline state, it is assumed to have entered a permanent fault state (crashed). In this case we reprogram the FPGA bitstream to recover from the crash. For each `(f, v)` pair, the results are logged, including the observed values of `reg_assign`, `clock_count`, and a boolean `crash` flag indicating system failure. The details of the undervolting scan is described in Appendix A.

5.1.2. Clock Count Reliability Under Undervolting. Figure 5 illustrates the behavior of the internal clock counter across a sweep of operating frequencies and supply voltages. The heatmap encodes the `clock_count` output using a logarithmic scale, where darker shades represent higher accumulated counts during the evaluation phase, and lighter shades indicate degraded or failed clock operation. White regions correspond to clock counts below the minimum detectable threshold and are considered *Hibernated* states.

At nominal voltage levels (above 0.85 V), the clock operates reliably across the entire frequency range up to 150 MHz, as evidenced by the uniform high count values (light blue regions). As the voltage decreases, however, a

clear degradation pattern emerges: higher frequencies are the first to experience failure, while lower frequencies maintain functionality down to lower voltage thresholds. The transition boundary from dark to light regions represents the onset of clock instability and is referred to as the *hibernation voltage*. This is the minimum voltage at which the clock counter can still increment meaningfully at a given frequency. Below this boundary, the white regions indicate complete clock failure—either due to the PLL losing lock, internal flip-flops ceasing to toggle, or propagation delays exceeding the clock period. The stable dark blue strip at the bottom of the plot (around 0.63 V and below) is a result of a complete FPGA crash where it cannot be recovered and captured data are all high-impedance (i.e., `0xff`).

5.1.3. Register Assignment Reliability. Figure 6 presents a discrete heatmap characterizing the behavior of the FPGA's register assignment under varying voltage and frequency conditions. Each cell represents the observed value of a target register after the undervolting test phase, with three possible states: a correct new assignment (blue), an old or default value indicating a failure to assign (red), or a system crash (gray).

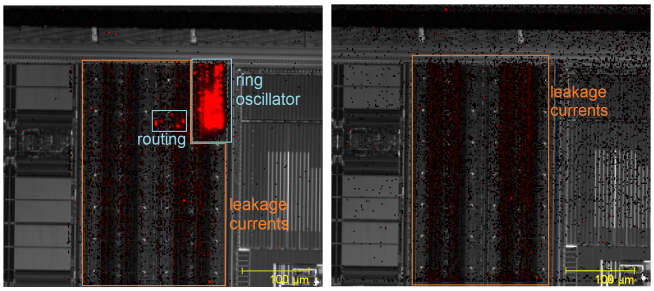
At voltages above 0.85 V, the register reliably latches the expected value (`0x88`) across all operating frequencies, indicating stable sequential logic and reliable data propagation. However, as the voltage drops below 0.70 V, an increasing number of cells transition to red, particularly at higher frequencies. This transition indicates a failure in the FPGA's ability to commit new values to registers—likely due to setup/hold time violations, degraded signal swing, or metastability induced by reduced supply voltage. Below approximately 0.60 V, the majority of register operations either result in incorrect values or trigger system crashes. These regions highlight the lower bound of operational safety for secure register logic.

The security implications of such failures are significant. Many FPGA-based protection mechanisms, including cryptographic randomization, register obfuscation, or randomized key preloading, rely on the ability to overwrite internal state deterministically. If undervolting prevents register assignments from executing correctly, it opens the possibility of residual sensitive data being left behind and making the system vulnerable to static SCA such as IA and LLSI.

5.2. Verifying Hibernation with Photon Emission

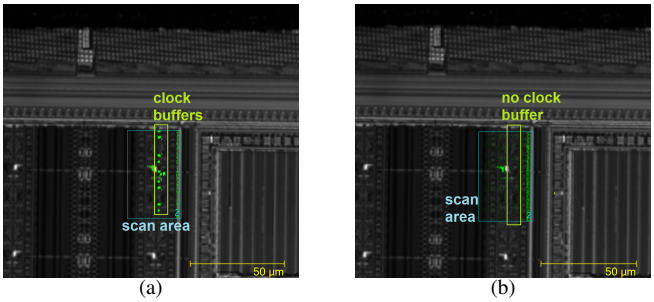
Relying on the FPGA IOs to verify that circuit switching is disabled may not be reliable, as an undervolted DUT may not be able to drive the IO buffers due to them being a large capacitive load. Hence, we can perform photon emission analysis to measure the activity to verify disabled circuits without relying on the chip itself. Using photon emission, we can observe dynamic internal signals, which should become static when entering hibernation.

As a test circuit, we chose a ring oscillator because it has a high emission rate due to its fast switching gates, and it is also the building block of various clock and voltage



(a) (b)

Figure 7: Photon emission images of a ring oscillator on a Kintex-7 at (a) nominal (V_{CCINT}=1000 mV) and (b) hibernation (V_{CCINT}=555 mV) voltages



(a) (b)

Figure 8: LVI images of some Kintex-7 clock buffers at (a) nominal (V_{CCINT}=1000 mV) and (b) hibernation V_{CCINT}=555 mV) voltages

sensors [14]. We can see the photon emission of the ring oscillator, as shown in Figure 7a. After lowering V_{CCINT} to hibernation levels, we find that the ring oscillator disappears, leaving only the leakage currents shown in Figure 7b. The presence of leakage currents during hibernation suggests that photon emission still occurs under these conditions due to the retained FPGA configuration. Therefore, the disappearance of the ring oscillator in the emission images suggests that the undervolting operation entirely disables its switching activity.

5.3. LVI of Clock Buffers

Another method to verify that internal clocks are disabled during hibernation is through LVI. We implemented four registers on the CW310, clocked at 10 MHz by an on-chip MMCM. Using the high-power incoherent light HIL and a 50× objective lens on the PHEMOS-X microscope, we performed an LVI scan at 10 MHz to locate the clock buffers. Under normal operating conditions, we observed several bright spots corresponding to active clock buffers, as shown in Figure 8a. However, when V_{CCINT} is lowered to hibernation levels, these bright spots disappear, as illustrated in Figure 8b. This disappearance indicates that the clock buffers are no longer switching at 10 MHz, confirming that the registers are effectively disabled. This effect may be attributed to the MMCM either losing its ability to drive the clock network or being entirely disabled under hibernation.

5.4. Verifying Hibernation of Flash-Based FPGAs

Although the flash-based PolarFire's configuration is non-volatile, unlike the SRAM-based FPGAs, the data stored in its registers is volatile. To characterize the hibernation range of the PolarFire, we run a counter, manually lower V_{DD}, and then restore it to its nominal value. If the FPGA does not crash, the counter value should remain unchanged regardless of the duration of the hibernation, and the device should resume operation from that exact point once power is restored. Based on these experiments, we observe that the PolarFire SoC FPGA supports hibernation across a wide voltage range, approximately from 0.9 V down to 0.2 V.

6. Defeating Sensors

6.1. Defeating Clock Sensors (Soft IPs)

Devices run on clocks that are either externally sourced, such as from crystal oscillators, or generated internally by oscillator circuitry. Several countermeasures based on clock sensors have been designed to protect systems from clock tampering. This is vital as clocks can be a single point of failure for security mechanisms that depend on reliable state modification like cryptographic randomization, obfuscated registers, and randomized key loading. Data retention under stopped clock conditions also exposes sensitive values to static side-channel analysis. Here, we focus on a recently proposed countermeasure [13] specifically designed to mitigate this threat. The function of this countermeasure is two-fold: it monitors the system clock such that it can trigger an alarm upon detecting a stop condition, then, in response to this alarm, it performs some actions to clear sensitive data from the circuit. This has only been evaluated on target circuits operating with a nominal supply voltage. We investigate its ability to perform both its detection and clearing functions when undervolted by Chypnosis. We briefly explain both variants of the countermeasure presented in [13], to contextualize our findings.

6.1.1. PLL-based Clock Sensor. The first variant uses a Phase-Locked Loop (PLL), see Figure 9a. This is a standard clock management control circuit used to generate clock signals derived from an externally sourced reference clock. PLLs will typically include an output status signal to indicate whether their generated clocks are synchronized, or "locked", to the reference. A stopped input clock will be detected rapidly by a PLL and signaled via deassertion of this signal, so [13] proposes using this as an alarm.

To evaluate the PLL-based sensor, we port the openly available implementation [1] to our CW310 board. This is designed for 7-Series AMD/Xilinx FPGAs and involves a Mixed-Mode Clock Manager (MMCM) with a `LOCKED` status signal from its internal PLL. The system then uses an alarm based on the inverse of this signal, $\overline{\texttt{LOCKED}}$, to switch the input of sensitive registers to instead come from an RNG. When input has been switched, subsequently arriving clock pulses from the PLL output latch the random values to clear sensitive data, this is called a *masked clear*. We first confirm the detection and clearing of our ported implementation to work correctly at the nominal supply voltage.

From preliminary Chypnosis experiments with a 10 MHz clock and setting V_{CCINT} to a 0.555 V hibernation voltage, we find that the protection mechanism fails. For system introspection to find the cause, we add an SR latch that takes the alarm as input, and a register that takes the alarm as a latch enable signal, both are shown in orange in Figure 9a. When we perform Chypnosis again, we find the SR latched value to be high, indicating successful detection. We also find the register output (`V_output`) retained its default value, suggesting that the mechanism to latch random data into the register failed. We further confirm the clock had indeed stopped using photo emission and laser probing of `sys_clk` as in Section 5.2. These results demonstrate that while the PLL-based protection can detect a stopped clock, it can still be bypassed by undervolting due to its failure to perform the masked clear (data overwrite) response actions.

6.1.2. Asynchronous Clock Sensor. For systems where PLLs are unsuitable, such as in low-power designs or systems with clock gating, [13] proposes a different clock sensor system based on signal propagation delay, see Figure 9b. The clock sensor is based on a tapped delay chain that takes the clock as the input, and effectively samples it at various time offsets defined by the delays between taps. All of these taps are fed into a combinatorial element that asserts an alarm signal if all inputs are equal, which is the case if the clock is stopped. The system works similarly to the PLL solution in that it uses the alarm, `stop_detect`, to select randomness as the input to sensitive registers. However, this design does not have a transitioning clock source to use for latching the randomness, thus the system also needs circuitry for generating an active clock edge. This is based on a secondary delay chain that takes the alarm signal, since this provides a low-to-high transition upon an alarm. The clock source going into the target, `sys_clk`, is multiplexed between the original `clk` and `delayed_edge`.

We port the asynchronous delay-based sensor countermeasure variant [1] to our CW310 board, and again confirm that the countermeasure works correctly at nominal supply voltage. Initial Chypnosis experiments, as carried out against the PLL-based system, similarly show the protection mechanism of this variant to fail. We modify the circuit with a similarly placed SR latch and register (shown in orange in Figure 9b) for introspection. Then, when performing Chypnosis again, we observe the same result as for the PLL system, which is that the detection mechanism functions correctly to trigger an alarm, but subsequent latching for the masked clear does not complete. We include a timing diagram of internal signals in Appendix D. We again confirm the clock had stopped using photon emission and laser probing of `sys_clk` as in Section 5.2. Thereby, we can also bypass this countermeasure variant with undervolting.

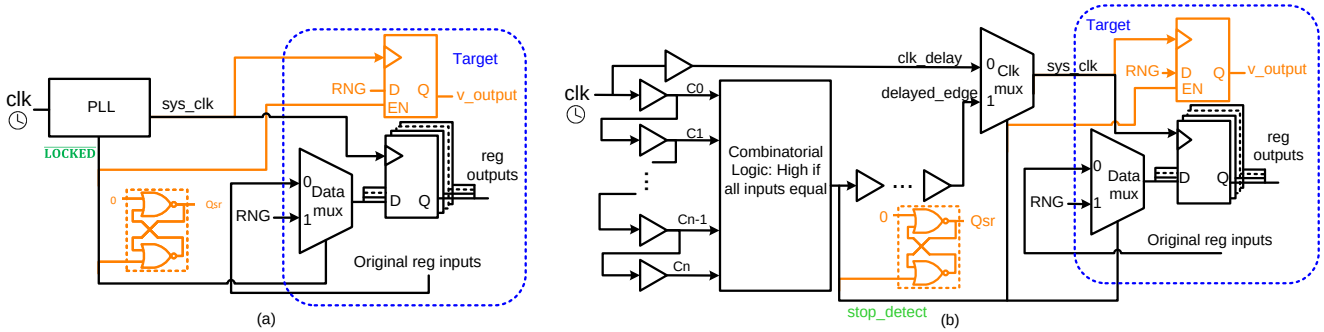


Figure 9: Borrowed Time countermeasures: (a) PLL-based clock sensor, (b) Delay chain-based sensor. The SR latch and the register (in orange) detect if the alarm goes high and if register latching occurs in an undervolting-induced clock halt.

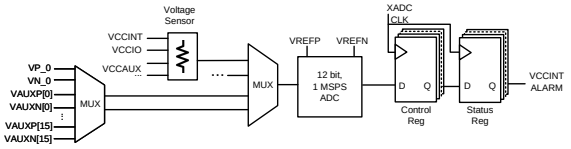


Figure 10: Block diagram of the XADC on 7 Series FPGAs

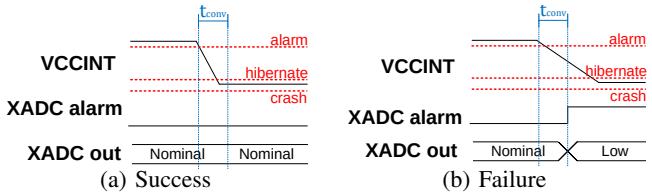


Figure 11: Graph of VCC_{INT} and associated XADC readings for an attempt to defeat the XADC

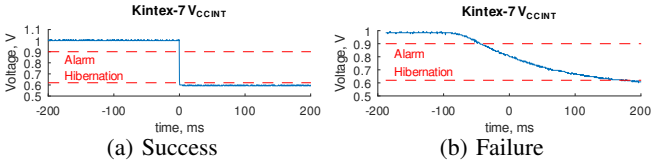


Figure 12: Oscilloscope waveforms for V_{CCINT} when attempting to defeat the XADC alarm signal

6.2. Defeating Xilinx Voltage Sensor (Hard IP)

AMD/Xilinx 7 Series FPGAs are equipped with an on-chip voltage sensor for V_{CCINT}, which is connected to the XADC, a 12-bit, 1 Mega Samples Per Second (MSPS) analog-to-digital converter [58]. Through a multiplexer, the XADC also shares its functionality with a temperature sensor, other voltage rail sensors (e.g., V_{CCAUX} and V_{CCIO}), and various analog I/Os, as illustrated in Figure 10.

The XADC features a built-in voltage alarm that triggers when the voltage falls outside a user-defined range, from V_{alarmL} to V_{alarmH}. However, these alarms rely on the digital output of the XADC. If the V_{CCINT} voltage is dropped rapidly enough, such that the time between V_{HIB} and V_{alarmL} is shorter than the XADC's conversion time (T_{conv}), the alarm will not be triggered. If successful, the last XADC reading before hibernation will show the normal operating voltage, as shown in Figure 11a.

When using a BK Precision 9130 triple-output power supply to power the ChipWhisperer CW310 through SMA connector J3, we were able to reduce V_{CCINT} to hibernation levels in approximately 400 ms. However, this drop was too slow to bypass detection, and the XADC alarm signal was triggered, as shown in Figure 12b. In contrast, by using a Tektronix AFG 3021 function generator as the power source, we achieved a much faster voltage drop to V_{HIB} within 80 μs—sufficient to avoid triggering the alarm signal (see Figure 12a). Although the XADC's sample rate of 1 MSPS suggests a conversion time (T_{conv}) of 1 μs, the actual conversion time may be longer due to pipelining in the ADC architecture [5]. Pipelining allows higher sample rates but introduces additional latency. While the XADC documentation does not explicitly mention its latency or pipelined design, our experimental results suggest that pipelining is employed. We also note that the XADC is not outright disabled during undervolting, as it is powered by V_{CCAUX} (1.8 V) rather than V_{CCINT}. Instead, it is more likely that the associated control registers become disabled, effectively rendering the XADC non-functional during hibernation.

6.3. Defeating Microchip AT Module (Hard IP)

Microchip PolarFire SoC FPGAs are equipped with a dedicated anti-tamper (AT) module [29] which consists of, among other things, on-chip voltage and clock sensors. These sensors can then zeroize the device, erasing the programmed bitstream and all data. Unlike the XADC on Xilinx devices, these sensors have fixed low and high voltage alarm thresholds, which assert the `VOLT_DETECT_1P0_LOW` and `VOLT_DETECT_1P0_HIGH` flags, respectively. In addition, the system's controller clock slows from 80 MHz to 20 MHz when it detects a V_{DD} brownout, and the tamper macro asserts the `SLOW_CLOCK` flag, which effectively serves as a second low-voltage alarm signal. The AT module monitors various tamper flags, which can be set to zeroize the FPGA in one of two ways. There is a watchdog timer that waits 1000 clock cycles after being enabled before zeroizing the device. For more critical tamper events, an additional input is available to zeroize the device immediately. To ensure the fastest possible zeroization, we connect

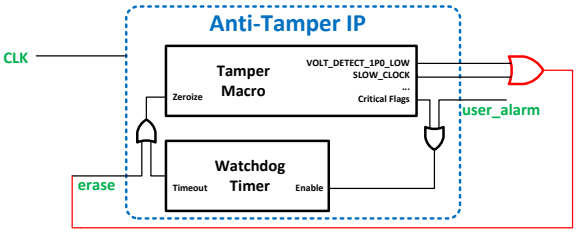


Figure 13: A simplified block diagram of the AT module on Microchip PolarFire SoC FPGAs

the VOLT_DETECT_1P0_LOW and SLOW_CLOCK flags to the latter input (see Figure 13).

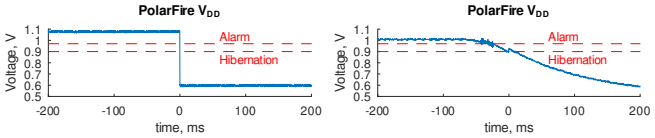
Using a BK Precision 9130 triple-output power supply to power the modified Microchip Kit, we were able to reduce V_{DD} to V_{HIB} in approximately 61.6,ms (see Figure 14b). However, this voltage drop was too slow to bypass detection, resulting in both the `SLOW_CLOCK` and `VOLT_DETECT_1P0_LOW` flags being triggered. When configured to do so, these flags can initiate zeroization. In contrast, using a Tektronix AFG 3021 function generator as the power source enabled a much faster voltage drop (approximately 430 ns to reach V_{HIB}) which was fast enough to avoid triggering either flag (see Figure 14a). Interestingly, even when the flags are configured to trigger zeroization, it does not occur in this rapid-drop scenario. Additionally, if we disable zeroization and set these flags to write to a register, no data is written in the rapid-drop scenario.

6.4. Voltage Falling Time vs Clock Frequency

We explore precisely how undervolting fall time influences attack success. Unfortunately, our equipment and setups do not support customizable voltage falling times. However, we recall that we expect rapid undervolting over a short time frame succeeds in bypassing voltage sensor detection because this time frame is shorter than, and thereby falls entirely within, the sensor sampling and response period. Therefore, we posit that the fall time is crucial in relation to the sensor's sampling frequency. Thus, we investigate the success of our approach in sensor bypass when altering the target device's clock frequency relative to a fixed voltage fall time. Note that a sensor's sampling rate may not be equal to the device clock rate, but rather is likely derived from the system clock rate.

Figure 15 shows the success rate of undervolting sensor bypass attempts against the Microchip anti-tamper module when setting the FPGA to various clock frequencies. We find that for our fixed fall time of 430 ns, clock and voltage sensor bypass success rates are higher for lower clock frequencies, which matches our expectations. Note that the fall time can be further decreased by deploying high-bandwidth instruments and high-frequency connections to the PCB.

In the case of the AMD/Xilinx sensors, we find the clock frequency to have no impact on bypass success, which is 100% for the same range of tested frequencies. This is even the case using the setup with the slower 80 μs falling time.



(a) Success (b) Failure

Figure 14: Oscilloscope waveforms for V_{DD} when attempting to defeat the PolarFire anti-tamper module

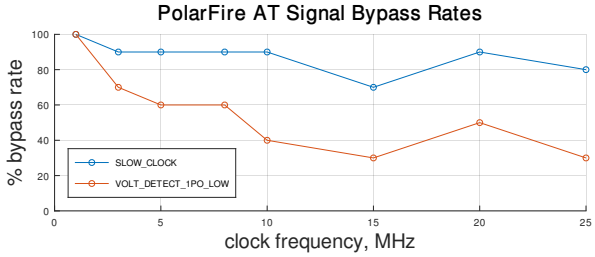


Figure 15: The success rate of bypassing the sensors using undervolting at various clock frequencies used by the anti-tamper module of Microchip FPGA

Since this is well below the sample rate used for the XADC, as discussed in Section 6.2, we expect the sensor's possible pipelined architecture to be responsible.

6.5. Defeating OpenTitan's Alert Handler

OpenTitan is an open-source silicon root of trust (RoT) project that provides a secure, transparent, and verifiable foundation for computing systems. It implements side-channel protected cryptographic functions, secure boot, device identity, and runtime attestation, and is designed for high-assurance environments. Earl Grey is the first full-chip implementation of the OpenTitan open-source RoT, designed as a low-power 32-bit RISC-V microprocessor. It features dedicated, secure cryptographic accelerators (e.g., HMAC and CSRNG), secure memory (flash, SRAM, OTP), and other hardened countermeasures, such as memory scrambling and enhanced physical memory protection [44]. Earl Grey is equipped with various defenses, including detection-based countermeasures against side-channel and fault attacks. Figure 16 illustrates the high-level diagram of OpenTitan's Earl Grey analog sensing and response mechanism. The *Analog Sensors Top* (AST) [43] module of OpenTitan provides an interface to analog/digital elements that monitor key environmental parameters such as voltage, clock integrity, and temperature. These sensors detect anomalies, such as voltage or clock glitches, that may indicate tampering attempts. Moreover, the *Sensor Controller* receives analog alert signals from the AST and forwards them to the alert handler, classifying each as recoverable or fatal. It also supports wake-up signaling, status readback [45]. Although multiple analog sensors can be deployed and connected to OpenTitan, we focus on AMD/Xilinx's XADC voltage sensor (See Section 6.2). As shown, the Earl Grey facilitates differential signaling for its sensors to increase

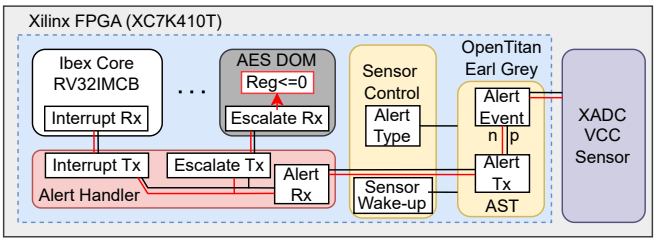


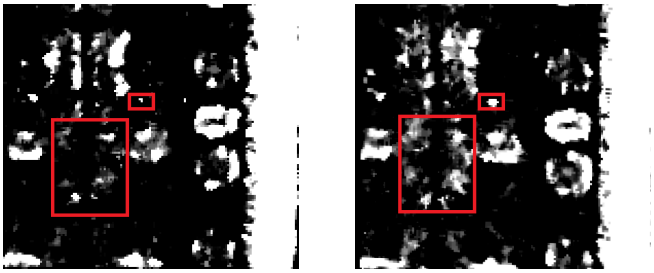
Figure 16: Architectures of OpenTitan Earl Gray Analog Sensors and Alert Handling Mechanism

reliability and protect against fault attacks [43]. Alert signals are then carried to the *Alert Handler* and are classified, and initial interrupts to the processor are raised accordingly. If the processor fails to respond in time, the alert handler escalates these alerts through programmable hardware actions such as chip reset, key erasure, or privilege lowering to contain potential threats. Differential alert signaling and programmable escalation protocols across four severity levels are provided in this module [42]. Although the escalation and response mechanism is handled by software by default, hardware/software transactions create *hundreds of microseconds* of latency that can be bypassed easily by our rapid undervolting. In this case, we utilize a fast-track response in hardware via OpenTitan's alert handler, which can be enabled by setting the accumulation trigger to 1 and configuring escalation to begin at phase 0, allowing critical alerts to trigger a response in as few as *4 clock cycles* when fully synchronous.

In this experiment, we follow the same procedure for deploying the XADC as described in Section 6.2. With a similar principle, a falling time of approximately $80\,\mu s$ is fast enough to surpass the differential signaling of the sensor, regardless of the underlying mechanism provided by OpenTitan. Our experiments confirm that the key register (masked values) assignments indeed do not occur. Nevertheless, the minimum latency introduced by the OpenTitan in such a scenario can be estimated to be $4 \times \frac{1s}{24Mhz} = 166.66ns$, where the default peripheral domain clock in Earl Grey runs at $24Mhz$. In the case where the sensor latency is negligible, an undervolting attack with a falling edge faster than $166.66ns$ (which is feasible with the same experimental attacker setup) disables differential alert propagation in OpenTitan and renders the response circuitry ineffective.

7. Side-channel Results

Having shown that undervolting can circumvent on-chip sensors to induce the conditions necessary for static SCA, namely an idle clock and continued data retention, we now investigate the viability of static SCA data extraction techniques against undervolted targets. Specifically, we use LLSI to perform direct register value readout, and we use IA to extract masked key shares from a cryptographic circuit.



(a) (b)

Figure 17: LLSI images of a register in 0 (left) and 1 (right) states on a hibernated FPGA, major differences boxed

7.1. Hibernated LLSI Attack

For LLSI to be feasible in undervolted conditions we must be able to clearly distinguish between a register in the 0 and 1 state from scans. We prepare our target FPGA with a register, and to discern its physical location we configure it to toggle between 0 and 1 at a certain frequency. This enables us to first localize it with LVI. Then, for optimal LLSI scans we used high laser power (90%) with a sufficiently large laser wavelength $\lambda = 1300\,\text{nm}$ to avoid inducing bit flips, ensuring we inject no unintended faults.

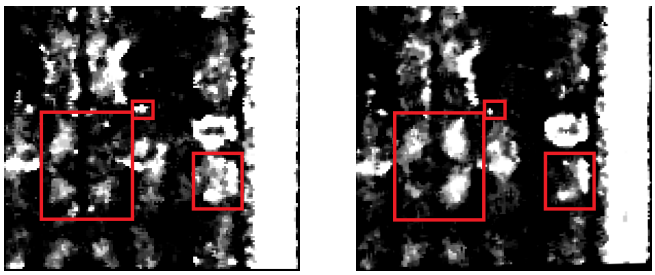
We apply similar image processing techniques to our scans as those used in photon emission analysis [26]. Specifically, we apply a median filter to remove salt-and-pepper noise, followed by a bilateral filter to smooth the image while preserving edges. The processed images, shown in Figures 17a and 17b, clearly distinguish between the two logic states. For reference, we also compare this to LLSI images of the same register on the same target in normal (non-hibernated) conditions, see Figure 18. The contrast between 0 and 1 states is more distinct in nominal conditions. However, there is still sufficient distinguishability to directly extract data from hibernated targets using LLSI.

7.2. Hibernated Impedance Attack

The practicality of IA against undervolted targets similarly depends on the distinguishability of individual bits. We perform a template impedance analysis attack [31], which first requires profiling to identify the specific frequency bands where the responses depend on data, localized to specific target registers.

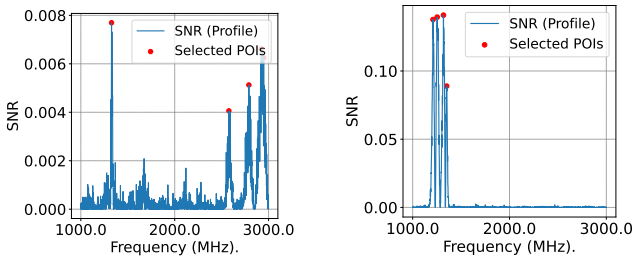
Our target is an AES protected with 3-share Domain-Oriented Masking (DOM) [15], used in OpenTitan. The attack scenario targets masked key bytes loaded into the internal key registers. We also equip the target with the voltage sensor in Section 6.2 and clock sensor protection from Section 6.1 for these registers.

In this scenario, we attack when the AES key's first byte shares and the corresponding input byte shares are loaded into the target. This makes for 24 distinct bit-level profiling tasks (8×3 shares). We perform undervolting at a hibernation voltage of *0.64 V* for both profiling and attack phase to disable the sensor. For profiling, we collect



(a) (b)

Figure 18: LLSI images of a register in 0 (left) and 1 (right) states on a non-hibernated FPGA, major differences boxed



(a) (b)

Figure 19: SNR curve for `Bit=1`(a) and `Bit=4`(b) of the key share across different frequencies. POIs shown in red.

$N_p = 20{,}000$ traces, each contributing to profiling all target bits. We separate N_p traces into two classes of $k = 0$ and $k = 1$, where k is the target bit to be profiled. For a given bit we define the SNR at a frequency index f as:

$$\text{SNR}(f) = \frac{(\mu_0(f) - \mu_1(f))^2}{\frac{1}{2}(\sigma_0^2(f) + \sigma_1^2(f)) + \epsilon}$$

where $\mu_0(f)$ and $\mu_1(f)$ are the mean values of class $(k = 0)$ and class $(k = 1)$ traces at frequency f. σ^2 represents the variance, and ϵ is a small constant added for numerical stability (e.g., $\epsilon = 10^{-8}$). We extract the Points of Interest (POIs) in the frequency domain by picking those that maximize the SNR. The details of POI selection and parameterization is described in Appendix C.

Figures 19a and 19b illustrate the observed bit-level leakage across the frequency spectrum for the `KeyBit=1`, `KeyBit=4` from the first share byte. These results are visualized using the SNR, emphasizing that distinct POIs across frequencies for individual bits enable template attacks to isolate and extract bit-specific information effectively. On the other hand, a similar analysis can be done using the phase Difference of Mean (DM) metric. For instance Figure 20, illustrates the $DM = \mu_0(\phi_f) - \mu_1(\phi_f)$ for $k \in \{0, 5, 6\}$. zoomed in a specific frequency window, illustrating the distinct bit-level impedance leakage.

Following the profiling phase, we conduct a single-trace attack against the instance with unknown key shares. To mitigate noise and improve robustness, we perform VNA-enabled averaging using $N_{\text{avg}} = 400$ repeated acquisitions for the same attack trace. The impedance template attack

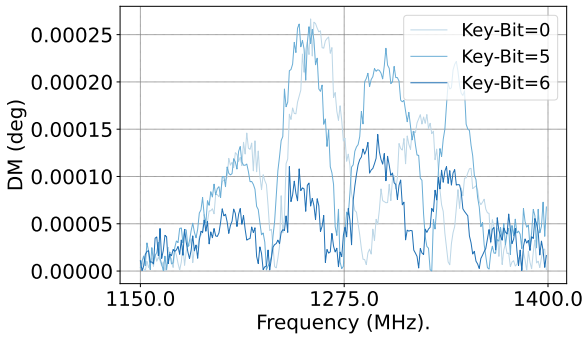


Figure 20: Phase Difference of Means impedance leakage for `Bit=0`, `Bit=5`, and `Bit=6` in one share byte over a zoomed in frequency window.

successfully recovers all individual bits from each share when enough iteration of averaging is performed. Naturally, the recovered bits from the three shares can then be combined to reconstruct the full first byte of the AES master key. For a detail key bit extraction analysis see Appendix B.

8. Undervolting-Resilient Countermeasure

This section proposes an improved design of the clock sensor systems evaluated in Section 6 that equips them to handle our undervolting attacks. We implement the updated design in both the AMD/Xilinx and Microchip PolarFire SoC FPGAs and find them to protect against our attacks.

To recap, both original clock sensor variants assert an alarm signal upon detecting a stopped clock, but undervolting hinders their subsequent response actions for performing a masked clear. Specifically, the mechanism for synchronous latching, to overwrite sensitive values with randomness, is affected and cannot complete. To remedy the problem, we propose modifying the masked clear mechanism. We observe that when using flip-flops that support asynchronous resets, such resets complete correctly, even in a brownout state. This robustness stems from these reset paths typically being built to remain dependable even when the device operates at, or marginally beyond, its process, voltage, and temperature limits. However, a naive use of asynchronous resets tied directly to the alarm signal would zeroize registers. As Dumitru et al. [13] point out, this would directly leak their Hamming weight dynamically. A potential fix is also to use flip-flop primitives, which instead asynchronously *preset* to 1. We verify that such flip-flops are supported and that the preset mechanism also works in brownout conditions. However, whether a flip-flop resets or presets is typically fixed to the primitive type, i.e., the same flip-flop cannot be dynamically configured to work as either type. Hence, directly using presets instead would still result in leakage.

To avoid introducing dynamic leakage upon an asynchronous clear, we can randomize the overall number of transitions by constructing unbiased registers based on a combination of these flip-flop primitives. Our scheme involves randomly selecting which type of register primitive

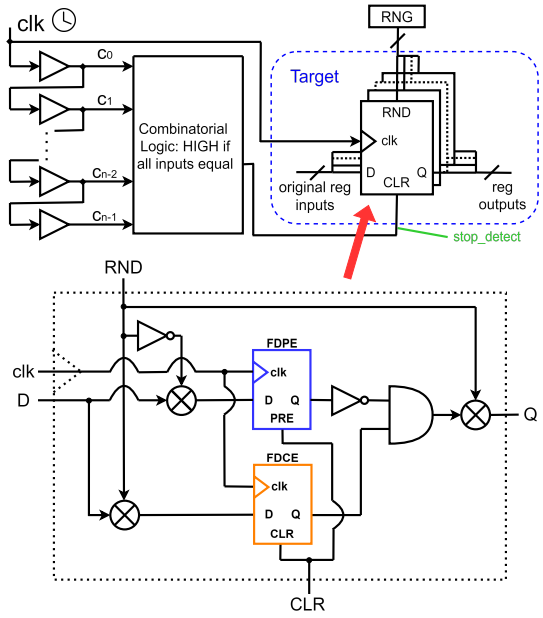


Figure 21: Complementary Registers within asynchronous delay-based Borrowed Time countermeasure with asynchronous clearing, equipped to handle undervolting attacks.

is used to store each bit of a sensitive value. Since the primitives are fixed components that require instantiation at build time, for the storage of a single sensitive bit, we implement one flip-flop of each kind and use randomness to dynamically select between them at runtime. Moreover, we also want to ensure that among the random number of transitions upon a clear, there is an equal number of $0 \rightarrow 1$ as $1 \rightarrow 0$ transitions so as to avoid information leakage due to imbalances. To that aim we propose a *complementary register* design that enables performing masked clears using the asynchronous reset/preset mechanisms.

Design and Implementations.[1] We implement the revised clock sensor systems into both the Xilinx/AMD and Microchip PolarFire SoC FPGA platforms attacked (with sensor bypasses) in Section 6 and Section 7.

Figure 21 depicts the revised asynchronous delay-based clock sensor system that we implement in the 7-Series AMD/Xilinx FPGAs [56] incorporating complementary registers. To the outside, they work as a regular single-cycle, rising-edge-triggered registers. Internally, they comprise two flip-flops with different properties: one which can be asynchronously reset to 0, and another which can be asynchronously preset to 1. Respectively, we implement these using FDCE (orange) and FDPE (blue) primitives. Both registers store the sensitive data in some form. For each bit of sensitive data, one flip-flop stores the bit and the other stores its complement. Which of the two flip-flops stores the data itself, i.e., the polarity of the complementary register, is determined by a random selector bit. The same "clear" signal sourced from the clock sensor alarm is connected to both the asynchronous reset and preset lines.

The sources of randomness used with our countermeasure are also likely to be affected by undervolting. However, this does not impact our countermeasure since we "pre-load" the complementary registers with randomness generated in previous clock cycles during normal circuit operation.

We extend the overhead analysis from the original work [13], finding the cost of our revised design to be similar and modest overall. As in [13], Table 1 reports power and utilization overheads with respect to a lightweight AES circuit [1] (Base), the circuit equipped with the original Borrowed Time clock sensor design (BT_{orig}), and our updated version with complementary registers (BT_{CR}). The randomness requirements are unchanged. With the updated design we double up the number of flip-flops for sensitive values and use additional LUTs for each complementary register's input and output logic (shown as logic gates in Figure 21). Conversely, LUTs used in the edge generation and data multiplexing circuitry from BT_{orig} are no longer required. While complementary registers impose no overhead in terms of extra clock cycles, if they are built into a design's critical path, the extra LUTs may affect the circuit's maximum possible clock frequency. This is the case in our evaluated AES circuit, where the critical path is slightly longer than the original Borrowed Time countermeasure. BT_{orig} instead only added a data multiplexing stage at the register inputs. The updated design also cuts down in other areas not captured by the table as we no longer require the same clock multiplexing and buffering circuitry. We further note that our protection (along with its overhead), when incorporated within a target with masking our protection, need only apply to one share of sensitive values.

	Power [mW]	c.p. [ns]	f_{max} [MHz]	LUTs	Regs
Base	116	5.080	196.9	1387	535
BT_{orig}	138	6.009	166.4	4079	1163
BT_{CR}	143	6.628	150.9	4075	1294

TABLE 1: Clock sensor countermeasure overheads on lightweight base AES core: power, critical path (c.p.) with corresponding max frequency, LUT and Register utilization.

In the Microchip PolarFire SoC FPGA we implement the PLL-based clock sensor with similar architecture to Figure 21, simply swapping out the detector circuit for a PLL contained in a PolarFire Clock Conditioning Circuitry (PF_CCC) module and its output `PLL_LOCK` in place of `stop_detect`. In place of FDPE and FDCE we use equivalent PolarFire flip-flop macros [28], DFN1E1C0 and DFN1E1P0, which asynchronously clear and preset, respectively. Both primitives have active low reset/preset, so we tie `PLL_LOCK` to them directly without negating it.

We validate the full improved countermeasure designs on both target FPGA families in the context of our undervolting attacks and find them to protect against the attacks.

[1]The source code for our design artifacts is available at https://github.com/0xADE1A1DE/Borrowed-Time

9. Discussion

9.1. Applicability to Static Power SCA

The main conditions necessary for static power SCA are sensitive data retention in the target during an idle clocking period, which we show to be simultaneously achievable with Chypnosis. We expect the data extraction methods of these attacks to still be possible with undervolting, however it is likely to affect their performance. Previous works [9, 34] point out that increased supply voltage amplifies leakage, but also that control over this parameter is not strictly necessary and only serves to reduce the overall number of traces needed. Moreover, Chypnosis does not preclude controlling any of the other parameters that maximize attack performance such as high temperature, intra-trace averaging, and post-processing. Importantly, Chypnosis relaxes the requisite adversary capabilities to not need any form of controllable clock, and to work in the presence of clock and voltage sensors. We leave investigation of undervolting static power SCA attacks to future work. We reaffirm that the revised clock sensor countermeasure we propose in Section 8 offers resilience to these attacks.

9.2. Comparison with Voltage Glitching Attacks

The undervolting used in our proposed attack might initially appear similar to conventional voltage glitching attacks [7, 55]. In voltage glitching, the adversary induces a transient voltage drop to cause timing violations in sequential logic, potentially triggering unauthorized state transitions. Such faults could, in some cases, bypass the response mechanisms of the countermeasures discussed in this paper. However, unlike glitching attacks, where the voltage returns to its nominal level and the system resumes normal operation, our approach involves a permanent voltage drop and clock halt, which is crucial for static SCA. Moreover, in glitching scenarios, various countermeasures such as Error Correction Codes (ECC) [51] can be employed to protect against transient faults. In contrast, the persistent nature of the voltage drop in our attack disables on-chip fault detection and response mechanisms, rendering them ineffective.

9.3. Comparison with Data Remanence Attacks

At first glance, one might assume that our proposed attack is similar to data remanence [4, 46], Cold Boot [18] or Pentimento [12] attacks, in which the adversary exploits charge retention or bias temperature instability effects in underlying transistors to recover data previously stored in memory. However, our attack differs in several key ways. First, it does not rely on temperature effects for data recovery. Additionally, those attacks generally assume that the adversary can run their firmware or bitstream on the chip after the sensitive data has been wiped, exploiting analog features of the memory (e.g., SRAM metastability or flip-flop propagation delay) to recover the contents. In contrast, our approach directly measures memory content that remains retained during a brownout condition, with the assumption that the adversary cannot take control of the chip by executing code or reading back any data at a later time.

9.4. Applicability to ASICs

Two questions may arise regarding the applicability of our attack to ASICs: (1) Can this technique inhibit the overwriting of secrets by voltage and clock sensors implemented in ASICs? (2) Can it also stop ASIC clocks? For the first question, it is important to note that the voltage and clock sensors in the Microchip SoC, as well as the voltage sensor in AMD/Xilinx FPGAs, are implemented as ASICs (hard IP). Our results clearly demonstrate that these sensors can be bypassed. Regarding the second question, prior work has already demonstrated that brownout conditions exist for SRAMs on ASICs and microcontrollers [37]. However, FPGAs have inherently larger capacitances than ASICs, which may make them more vulnerable to this attack compared to ASICs. As a result, while a detailed comparison between FPGAs and ASICs is beyond the scope of this work, our findings strongly indicate that the core mechanisms exploited in our attack are not exclusive to FPGAs and warrant further investigation in future work.

9.5. End-to-end Key Extraction

In Section 7.1 and Section 7.2 we demonstrated register bit extraction and a masked AES key-byte extraction on an undervolted FPGA using impedance and LLSI attacks, respectively. To perform complete key recovery, an attacker can simply repeat the same procedure for all key bytes over a longer period. Since undervolting halts the clock while retaining data indefinitely, a longer measurement time for extraction has no inhibitory consequences for the approach.

10. Conclusion

In this work, we introduced *Chypnosis*, a novel undervolting attack that exploits the vulnerability of chips during brownout conditions. By inducing rapid voltage drops, we demonstrated that it is possible to halt all internal clock sources and freeze the circuit's state without triggering conventional clock, voltage, or brownout detection (BOD) sensors, and consequently, the erasure of sensitive data. This enables adversaries to extract the retained secrets in flip-flops and other non-volatile memories using static side-channels, such as LLSI and IA. Our extensive experiments on SRAM-based and Flash-based FPGAs validated our claims. We also showed that our attack can bypass the OpenTitan RoT's alert handler, demonstrating its real-world impact. To mitigate the threats posed by Chypnosis, we proposed a revised clock sensor countermeasure design, which we demonstrated can protect all evaluated systems, even in brownout conditions.

11. Responsible Disclosure

Following the discovery of the vulnerability, we responsibly disclosed it to AMD, Microchip, and OpenTitan on June 7, 2025, upon completing the initial version of the manuscript. All parties have responded to initial contact and have remained in contact since the paper report was sent out immediately following their initial responses. All parties acknowledged receiving the report. Since the report's acknowledgments, we have held meetings with representatives from all relevant parties to discuss plans for addressing the issues and the embargo timelines. In each of the meetings, we informed the parties that the findings would be embargoed until at least September 4. AMD publicly acknowledged the vulnerability on September 18 [3].

Acknowledgments

This effort was sponsored by NSF Grants CNS-2150123 and CNS-2338069; Draper Scholars Program; Research and Development (R&D) grant from the Massachusetts Technology Collaborative; an ARC Discovery Project number DP210102670; and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972.

References

[1] 0xADE1A1DE, "Borrowed Time: An in-chip countermeasure against static side-channel analysis attacks," https://github.com/0xADE1A1DE/Borrowed-Time, 2025, accessed: 2025-04-03.

[2] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The EM Side-Channel(s)," in *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, ser. Lecture Notes in Computer Science, B. S. K. Jr., Ç. K. Koç, and C. Paar, Eds., vol. 2523. Springer, 2002, pp. 29–45. [Online]. Available: https://doi.org/10.1007/3-540-36400-5_4

[3] AMD Product Security, "Undervoltage-based static side-channel attacks ("chypnosis") on fpgas," AMD, Security Brief AMD-SB-8018, 2025, potential Impact: Loss of Confidentiality; Severity: Medium. [Online]. Available: https://www.amd.com/en/resources/product-security/bulletin/amd-sb-8018.html

[4] N. A. Anagnostopoulos, T. Arul, M. Rosenstihl, A. Schaller, S. Gabmeyer, and S. Katzenbeisser, "Low-temperature Data Remanence Attacks Against Intrinsic SRAM PUFs," in *21st Euromicro Conference on Digital System Design, DSD 2018, Prague, Czech Republic, August 29-31, 2018*, M. Novotný, N. Konofaos, and A. Skavhaug, Eds. IEEE Computer Society, 2018, pp. 581–585. [Online]. Available: https://doi.org/10.1109/DSD.2018.00102

[5] Analog Devices Inc., "Understanding Pipelined ADCs," https://www.analog.com/en/resources/technical-articles/understanding-pipelined-adcs.html, 2001, [Accessed 2025-04-09].

[6] M. S. Awal and M. T. Rahman, "Impedance leakage vulnerability and its utilization in reverse-engineering embedded software," *arXiv preprint arXiv:2310.03175*, 2023.

[7] R. Buhren, H. N. Jacob, T. Krachenfels, and J. Seifert, "One Glitch to Rule Them All: Fault Injection Attacks Against AMD's Secure Encrypted Virtualization," in *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*, Y. Kim, J. Kim, G. Vigna, and E. Shi, Eds. ACM, 2021, pp. 2875–2889. [Online]. Available: https://doi.org/10.1145/3460120.3484779

[8] B. H. Calhoun and A. P. Chandrakasan, "Static noise margin variation for sub-threshold SRAM in 65-nm CMOS," *IEEE Journal of solid-state circuits*, vol. 41, no. 7, pp. 1673–1679, 2006.

[9] G. Cassiers, L. Masure, C. Momin, T. Moos, and F. Standaert, "Prime-field Masking in Hardware and its Soundness against Low-noise SCA Attacks," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2023, no. 2, pp. 482–518, 2023. [Online]. Available: https://doi.org/10.46586/tches.v2023.i2.482-518

[10] S. Chef, C. Chua, J. Tay, and C. Gan, "Quantitative study of photoelectric laser stimulation for logic state imaging in embedded SRAM," in *International Symposium for Testing and Failure Analysis*, vol. 84215. ASM International, 2021, pp. 154–162.

[11] S. Chef, C. T. Chua, J. Y. Tay, J. Cheah, and C. L. Gan, "Embedded-EEPROM descrambling via laser-based techniques - A case study on AVR MCU," in *Workshop on Fault Detection and Tolerance in Cryptography, FDTC 2022, Virtual Event / Italy, September 16, 2022*. IEEE, 2022, pp. 1–8. [Online]. Available: https://doi.org/10.1109/FDTC57191.2022.00010

[12] C. Drewes, O. Weng, A. Meza, A. Althoff, D. Kohlbrenner, R. Kastner, and D. Richmond, "Pentimento: Data Remanence in Cloud FPGAs," in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2, ASPLOS 2024, La Jolla, CA, USA, 27 April 2024- 1 May 2024*, R. Gupta, N. B. Abu-Ghazaleh, M. Musuvathi, and D. Tsafrir, Eds. ACM, 2024, pp. 862–878. [Online]. Available: https://doi.org/10.1145/3620665.3640355

[13] R. Dumitru, T. Moos, A. Wabnitz, and Y. Yarom, "On Borrowed Time - Preventing Static Side-channel Analysis," in *32nd Annual Network and Distributed System Security Symposium, NDSS 2025, San Diego, California, USA, February 24-28, 2025*. The Internet Society, 2025. [Online]. Available: https://www.ndss-symposium.org/ndss-paper/on-borrowed-time-preventing-static-side-channel-analysis/

[14] T. Farheen, S. Roy, S. Tajik, and D. Forte, "A Twofold Clock and Voltage-based Detection Method for Laser Logic State Imaging Attack,"

IEEE Trans. Very Large Scale Integr. Syst., vol. 31, no. 1, pp. 65–78, 2023. [Online]. Available: https://doi.org/10.1109/TVLSI.2022.3214724

[15] H. Groß, S. Mangard, and T. Korak, "Domain-oriented Masking: Compact Masked Hardware Implementations with Arbitrary Protection Order," *IACR Cryptol. ePrint Arch.*, p. 486, 2016. [Online]. Available: http://eprint.iacr.org/2016/486

[16] P. J. Grossmann, M. E. Leeser, and M. Onabajo, "Minimum energy analysis and experimental verification of a latch-based subthreshold FPGA," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 59, no. 12, pp. 942–946, 2012.

[17] S. I. Haider and L. Nazhandali, "Utilizing subthreshold technology for the creation of secure circuits," in *International Symposium on Circuits and Systems (ISCAS 2008), 18-21 May 2008, Sheraton Seattle Hotel, Seattle, Washington, USA*. IEEE, 2008, pp. 3182–3185. [Online]. Available: https://doi.org/10.1109/ISCAS.2008.4542134

[18] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten, "Lest we remember: cold-boot attacks on encryption keys," *Commun. ACM*, vol. 52, no. 5, pp. 91–98, 2009. [Online]. Available: https://doi.org/10.1145/1506409.1506429

[19] Hamamatsu Photonics K.K., "PHEMOS-X Emission Microscope C15765-01," https://www.hamamatsu.com/us/en/product/semiconductor-manufacturing-support-systems/failure-analysis-system/C15765-01.html, accessed: 2025-06-05.

[20] D. E. Holcomb, A. Rahmati, M. Salajegheh, W. P. Burleson, and K. Fu, "DRV-Fingerprinting: Using Data Retention Voltage of SRAM Cells for Chip Identification," in *Radio Frequency Identification. Security and Privacy Issues - 8th International Workshop, RFIDSec 2012, Nijmegen, The Netherlands, July 2-3, 2012, Revised Selected Papers*, ser. Lecture Notes in Computer Science, J. Hoepman and I. Verbauwhede, Eds., vol. 7739. Springer, 2012, pp. 165–179. [Online]. Available: https://doi.org/10.1007/978-3-642-36140-1_12

[21] Keysight, *Keysight Documentations*, 2025, accessed: 2025-03-31. [Online]. Available: https://www.keysight.com/us/en/product/E5080A/e5080a-ena-vector-network-analyzer.html

[22] T. Kiyan, H. Lohrke, and C. Boit, "Comparative assessment of optical techniques for semi-invasive SRAM data read-out on an MSP430 microcontroller," in *International Symposium for Testing and Failure Analysis*, vol. 81009. ASM International, 2018, pp. 266–271.

[23] P. C. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, ser. Lecture Notes in Computer Science, M. J. Wiener, Ed., vol. 1666. Springer, 1999, pp. 388–397. [Online]. Available: https://doi.org/10.1007/3-540-48405-1_25

[24] T. Krachenfels, F. Ganji, A. Moradi, S. Tajik, and J. Seifert, "Real-world Snapshots vs. Theory: Questioning the t-Probing Security Model," in *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*. IEEE, 2021, pp. 1955–1971. [Online]. Available: https://doi.org/10.1109/SP40001.2021.00029

[25] T. Krachenfels, T. Kiyan, S. Tajik, and J. Seifert, "Automatic Extraction of Secrets from the Transistor Jungle using Laser-assisted Side-channel Attacks," in *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, M. D. Bailey and R. Greenstadt, Eds. USENIX Association, 2021, pp. 627–644. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/krachenfels

[26] D. M. Mehta, M. Hashemi, D. Forte, S. Tajik, and F. Ganji, "1/0 Shades of UC: Photonic Side-Channel Analysis of Universal Circuits," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2024, no. 3, pp. 574–602, Jul. 2024. [Online]. Available: https://tches.iacr.org/index.php/TCHES/article/view/11688

[27] Microchip, "PolarFire Family FPGA Security User Guide," 2025.

[28] ——, *PolarFire FPGA and PolarFire SoC FPGA Macro Library User Guide*, 2023, [Accessed 2025-06-05]. [Online]. Available: https://ww1.microchip.com/downloads/aemdocuments/documents/fpga/core-docs/libero/2023_1/tool/pf_mlg.pdf

[29] Microchip Technology Inc., "PolarFire SoC FPGA MPFS095T-1FCSG325E," https://www.microchipdirect.com/product/MPFS095T-1FCSG325E, accessed: 2025-06-05.

[30] MiniCircuits, *MiniCircuits Datasheets*, 2025, accessed: 2025-03-31. [Online]. Available: https://www.mouser.com/datasheet/2/1030/CBL_2FT_SMNM_2b-2303455.pdf

[31] S. K. Monfared, T. Mosavirik, and S. Tajik, "LeakyOhm: Secret Bits Extraction using Impedance Analysis," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS 2023, Copenhagen, Denmark, November 26-30, 2023*, W. Meng, C. D. Jensen, C. Cremers, and E. Kirda, Eds. ACM, 2023, pp. 1675–1689. [Online]. Available: https://doi.org/10.1145/3576915.3623092

[32] S. K. Monfared, D. Forte, and S. Tajik, "RandOhm: Mitigating Impedance Side-channel Attacks using Randomized Circuit Configurations," in *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design, ICCAD 2024, Newark Liberty International Airport Marriott, NJ, USA, October 27-31, 2024*, J. Xiong and R. Wille, Eds. ACM, 2024, pp. 221:1–221:9. [Online]. Available: https://doi.org/10.1145/3676536.3676687

[33] S. K. Monfared, K. Mitard, A. Cannon, D. Forte, and S. Tajik, "LaserEscape: Detecting and Mitigating

Optical Probing Attacks," in *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design, ICCAD 2024, Newark Liberty International Airport Marriott, NJ, USA, October 27-31, 2024*, J. Xiong and R. Wille, Eds. ACM, 2024, pp. 224:1–224:10. [Online]. Available: https://doi.org/10.1145/3676536.3676822

[34] T. Moos, "Static Power SCA of Sub-100 nm CMOS ASICs and the Insecurity of Masking Schemes in Low-noise Environments," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2019, no. 3, pp. 202–232, 2019. [Online]. Available: https://doi.org/10.13154/tches.v2019.i3.202-232

[35] A. Moradi, "Side-channel Leakage through Static Power - Should We Care about in Practice?" in *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, ser. Lecture Notes in Computer Science, L. Batina and M. Robshaw, Eds., vol. 8731. Springer, 2014, pp. 562–579. [Online]. Available: https://doi.org/10.1007/978-3-662-44709-3_31

[36] T. Mosavirik, S. K. Monfared, M. Saadat-Safa, and S. Tajik, "Silicon Echoes: Non-invasive Trojan and Tamper Detection using Frequency-selective Impedance Analysis," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2023, no. 4, pp. 238–261, 2023. [Online]. Available: https://doi.org/10.46586/tches.v2023.i4.238-261

[37] D. Nedospasov, J. Seifert, C. Helfmeier, and C. Boit, "Invasive PUF Analysis," in *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography, Los Alamitos, CA, USA, August 20, 2013*, W. Fischer and J. Schmidt, Eds. IEEE Computer Society, 2013, pp. 30–38. [Online]. Available: https://doi.org/10.1109/FDTC.2013.19

[38] NewAE, "CW305 Artix FPGA Target," https://rtfm.newae.com/Targets/CW305ArtixFPGA, 2025, [Online; accessed 2025-03-31].

[39] ——, "BCW310 Bergen Board," https://rtfm.newae.com/Targets/CW310BergenBoard, 2025, [Accessed 2025-03-31].

[40] NewAE, *NewAE Hardware Product*, 2025, accessed: 2025-03-31. [Online]. Available: https://rtfm.newae.com/Capture/ChipWhisperer-Lite

[41] B. Niu, G. M. E. Khoo, Y.-C. S. Chen, F. Chapman, D. Bockelman, and T. Tong, "Laser logic state imaging (llsi)," in *International Symposium for Testing and Failure Analysis*, vol. 30927. ASM International, 2014, pp. 65–72.

[42] OpenTitan Project, "OpenTitan Documentation," https://opentitan.org/book/hw/top_earlgrey/ip_autogen/alert_handler/index.html#alert-handler-technical-specification, 2025.

[43] ——, "Analog Sensor Top Technical Specification," https://opentitan.org/book/hw/top_earlgrey/ip/ast/index.html, 2025.

[44] ——, "OpenTitan Earl Grey," https://opentitan.org/book/hw/top_earlgrey/doc/datasheet.html, 2025.

[45] ——, "Sensor Control Technical Specification," https://opentitan.org/book/hw/top_earlgrey/ip/sensor_ctrl/index.html, 2025.

[46] Y. Oren, A. Sadeghi, and C. Wachsmann, "On the Effectiveness of the Remanence Decay Side-channel to Clone Memory-based PUFs," in *Cryptographic Hardware and Embedded Systems - CHES 2013 - 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings*, ser. Lecture Notes in Computer Science, G. Bertoni and J. Coron, Eds., vol. 8086. Springer, 2013, pp. 107–125. [Online]. Available: https://doi.org/10.1007/978-3-642-40349-1_7

[47] S. Pable and M. Hasan, "High speed interconnect through device optimization for subthreshold FPGA," *Microelectronics Journal*, vol. 42, no. 3, pp. 545–552, 2011.

[48] E. Peterson, "Developing tamper resistant designs with Xilinx Virtex-6 and 7 series FPGAs," *Application Note XAPP1084 (v1.4). Xilinx Corporation*, 2017.

[49] G. Provelengios, D. Holcomb, and R. Tessier, "Mitigating voltage attacks in multi-tenant FPGAs," *ACM transactions on reconfigurable technology and systems (TRETS)*, vol. 14, no. 2, pp. 1–24, 2021.

[50] H. Qi, O. A. Ayorinde, and B. H. Calhoun, "An energy-efficient near/sub-threshold FPGA interconnect architecture using dynamic voltage scaling and power-gating," in *2016 International Conference on Field-Programmable Technology, FPT 2016, Xi'an, China, December 7-9, 2016*, Y. Song, S. Wang, B. Nelson, J. Li, and Y. Peng, Eds. IEEE, 2016, pp. 20–27. [Online]. Available: https://doi.org/10.1109/FPT.2016.7929183

[51] C. Spensky, A. Machiry, N. Burow, H. Okhravi, R. Housley, Z. Gu, H. Jamjoom, C. Kruegel, and G. Vigna, "Glitching Demystified: Analyzing Control-flow-based Glitching Attacks and Defenses," in *51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2021, Taipei, Taiwan, June 21-24, 2021*. IEEE, 2021, pp. 400–412. [Online]. Available: https://doi.org/10.1109/DSN48987.2021.00051

[52] V. Sze, R. Blázquez, M. Bhardwaj, and A. P. Chandrakasan, "An Energy Efficient Sub-threshold Baseband Processor Architecture for Pulsed Ultra-wideband Communications," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing, ICASSP 2006, Toulouse, France, May 14-19, 2006*. IEEE, 2006, pp. 908–911. [Online]. Available: https://doi.org/10.1109/ICASSP.2006.1660802

[53] S. Tajik, J. Fietkau, H. Lohrke, J. Seifert, and C. Boit, "PUFMon: Security monitoring of FPGAs using physically unclonable functions," in *23rd IEEE International Symposium on On-Line Testing and Robust System Design, IOLTS 2017, Thessaloniki, Greece, July 3-5, 2017*. IEEE, 2017, pp. 186–191. [Online]. Available:

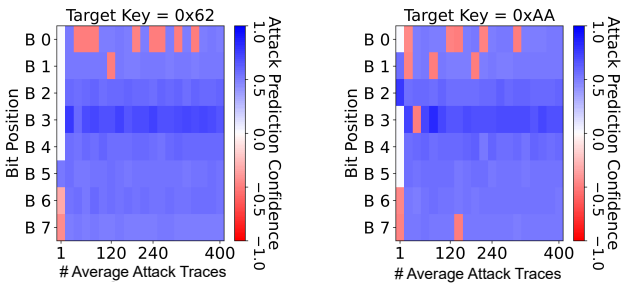
https://doi.org/10.1109/IOLTS.2017.8046216
[54] S. Tajik, H. Lohrke, J. Seifert, and C. Boit, "On the Power of Optical Contactless Probing: Attacking Bitstream Encryption of FPGAs," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, B. Thuraisingham, D. Evans, T. Malkin, and D. Xu, Eds. ACM, 2017, pp. 1661–1674. [Online]. Available: https://doi.org/10.1145/3133956.3134039
[55] A. Tang, S. Sethumadhavan, and S. J. Stolfo, "CLKSCREW: Exposing the Perils of Security-oblivious Energy Management," in *26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017*, E. Kirda and T. Ristenpart, Eds. USENIX Association, 2017, pp. 1057–1074. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/tang
[56] Xilinx, Inc., *Vivado Design Suite 7 Series FPGA and Zynq 7000 SoC Libraries Guide (UG953)*, November 2024, [Accessed 2025-05-27]. [Online]. Available: https://docs.amd.com/r/en-US/ug953-vivado-7series-libraries/
[57] ——, *Security Monitor IP Core*, 2019, [Accessed 2025-06-05]. [Online]. Available: https://www.xilinx.com/support/documents/product-briefs/security-monitor-ip-core-product-brief.pdf
[58] Xilinx Inc., *7 Series FPGAs and Zynq-7000 SoC XADC Dual 12-Bit 1 MSPS Analog-to-Digital Converter User Guide*, Jun. 2025, accessed: 2025-03-31. [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug480_7Series_XADC.pdf
[59] B. Zhai, S. Pant, L. Nazhandali, S. Hanson, J. Olson, A. Reeves, M. Minuth, R. Helfand, T. M. Austin, D. Sylvester, and D. T. Blaauw, "Energy-efficient Subthreshold Processor Design," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 17, no. 8, pp. 1127–1137, 2009. [Online]. Available: https://doi.org/10.1109/TVLSI.2008.2007564
[60] H. Zhang, L. Lin, Q. Fang, and M. Alioto, "Laser Voltage Probing Attack Detection With 100% Area/Time Coverage at Above/Below the Bandgap Wavelength and Fully-automated Design," *IEEE J. Solid State Circuits*, vol. 58, no. 10, pp. 2919–2930, 2023. [Online]. Available: https://doi.org/10.1109/JSSC.2023.3274596
[61] K. M. Zick and J. P. Hayes, "Low-cost sensing with ring oscillator arrays for healthier reconfigurable systems," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 5, no. 1, pp. 1–26, 2012.
[62] K. M. Zick, M. Srivastav, W. Zhang, and M. French, "Sensing nanosecond-scale voltage attacks and natural transients in FPGAs," in *The 2013 ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA '13, Monterey, CA, USA, February 11-13, 2013*, B. L. Hutchings and V. Betz, Eds. ACM, 2013, pp. 101–104. [Online]. Available: https://doi.org/10.1145/2435264.2435283

Appendix A. Hibernation Characterization Algorithm

Algorithm 1 outlines in detail the scanning procedure presented in Section 5.1 for discovering the combined voltage and clock frequency hibernation threshold of a device.

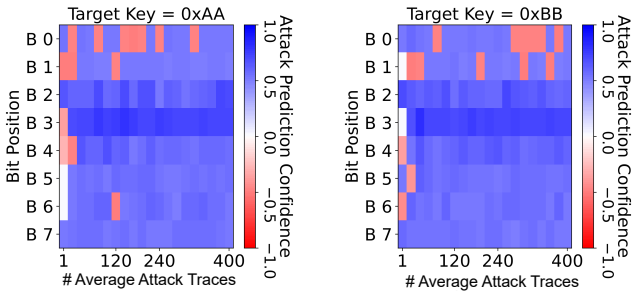
Algorithm 1 Undervolting Voltage-Frequency Scan

function HIBERNATION SCAN
  for $f \in \text{linspace}(f_{low}, f_{high}, f_{step})$ **do** ▷ Sweep frequency
    for $v \in \text{linspace}(V_{high}, V_{low}, -V_{step})$ **do** ▷ Sweep voltage
      Write (PLL_Freq) $\leftarrow f$
      DEBUG_REG_RESET() ▷ Reset all values
      / Set Initial Delay */*
      Write (REG_INIT_DELAY) $\leftarrow t_d$
      / Set Test Duration */*
      Write (REG_EVAL_TIME) $\leftarrow t_t$ ▷ Minimum 0.5s
      / Trigger Evaluation */*
      Write (REG_EXEC_TEST) $\leftarrow 0x01$
      / UNDERVOLTING */*
      Write (VAL_Voltage) $\leftarrow v$
      Wait $t = 0.8s$ ▷ Eval time $t > t_d + t_t$
      / Recovery phase */*
      Write (VAL_Voltage) $\leftarrow V_{high}$
      / Reading debug regs */*
      $R_{rec} \leftarrow$ Read(REG_DEBUG_VAL)
      $reg\_assign \leftarrow R_{rec}[0]$
      $clock\_count \leftarrow R_{rec}[1{:}10]$
      / Crash Recovery */*
      if DEBUG_REG_RESET() $\neq 0$ **then**
        $crash \leftarrow$ True
        REPROGRAM_FPGA()
      else
        $crash \leftarrow$ False
      Store $\{f, v, reg\_assign, clock\_count, crash\}$
function DEBUG_REG_RESET()
  Write (REG_DEBUG_RST) $\leftarrow 0x01$
  Wait 0.1 sec ▷ Wiping registers
  Write (REG_DEBUG_RST) $\leftarrow 0x00$
  $R \leftarrow$ Read(REG_DEBUG_VAL)
  if $R[0] \neq R_{Predefined}$ **then**
    return -1 ▷ Crash Detected
  else
    return 0



(a) (b)

Figure 22: Attack prediction confidence via LDA method for a target with `KeyByte=0x62`(a) and `KeyByte=0xAA`(b). Blue and Red spots indicate correct and wrong predictions.



(a) (b)

Figure 23: Attack prediction confidence via RF method for a target with `KeyByte=0xAA`(a) and `KeyByte=0xBB`(b). Blue and Red spots indicate correct and wrong predictions.

A.1. Failure Conditions and Interpretation

Our tests capture the following modes of failure:

- **Register Assignment Failure:** If the register assignment output differs from its expected value (e.g., `reg_assign` $\neq$ `0x88`), it indicates that flip-flops failed to latch the input due to timing violations introduced by low voltage.
- **Clocking Failure:** If the `clock_count` remains near zero, it implies that the clock network failed to propagate or that the counter logic ceased functioning.
- **Crash State:** If the debug registers fail to return to a known state after reset, this is treated as a system-wide logic failure. It often correlates with unstable supply levels that corrupt control paths or flip-flop states.

Appendix B. Impedance Attack Key Extraction Analysis

Section 7.2 describes the results of our Impedance Analysis attacks with Chypnosis. Here we show examples of the post-attack key bit extraction, based on template matching scores Figures 22 and 23 via LDA and RF methods, respectively. These figures present the confidence of the attack prediction. Blue colors represent the correct value, whereas red spots highlight the wrong prediction. In Figure 22a, a trace with `KeyByte=0x62` is captured and analyzed. As the number of averaging increases, the template model tends to make fewer errors predicting the right value for key bits. Furthermore, as shown in Figure 23b, the RF model performs poorly predicting some bits (e.g., `Bit=0`) with a small amount of averaging.

Appendix C. POI Selection in Impedance Attack Profiling

In our impedance analysis attack presented in Section 7.2, during profiling we select the frequencies (POIs) that carry the most side-channel information.

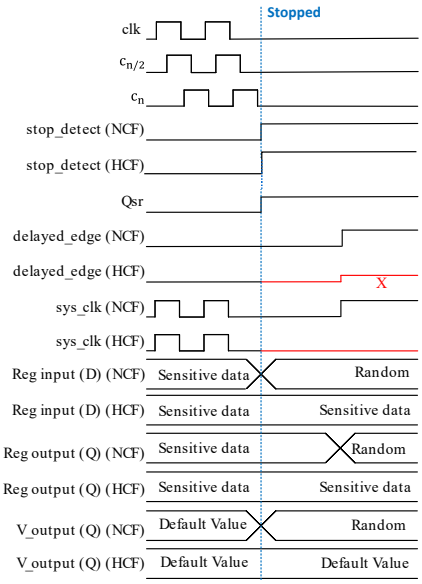


Figure 24: Timing diagram of the asynchronous circuit for clock halt detection and attempted register overwrite. Undervolting prevents the register from overwriting its content, causing it to retain the data. X indicates an unknown signal value during this process. NCF and HCF refer to normal clock freezing and hibernated clock freezing, respectively.

To select the top-k POIs, we use some empirically tuned parameters to constrain the candidate set $S(f) \subset SNR(f)$ with a *Minimum Height* where:

$$S(f) \geq \alpha \cdot \max(S)$$

with a relative height threshold $\alpha = 0.3$, and enforce a *Minimum Distance* so POIs are at least $d_{\min}$ samples apart:

$$|f_i - f_j| \geq d_{\min} \quad \forall i \neq j$$

Appendix D. Clock Sensor System Timing

The timing diagram of the clock sensor (Section 6.1) is shown under attack Figure 24.