CVA6 RISC-V PMP Vulnerabilities against FIA

Kévin Quénéhervé[1], Philippe Tanguy[1], Rachid Dafali[2] and Vianney Lapôtre[1]

[1] Université Bretagne Sud UMR 6285, Lab-STICC Lorient, France
[2] DGA MI Bruz, France

Abstract

Fault Injection Attacks (FIA) present a considerable threats to the security and reliability of embedded systems. FIAs can compromise an embedded processor by altering its clock signal, power supply or by using electromagnetic pulses. This study focuses on analyzing the impact of FIA on the Physical Memory Protection (PMP) configuration flow within a CVA6 RISC-V core. We conducted fault injection campaigns on an FPGA implementation using an ARTY A7-100T board to characterize the resulting fault effects. To achieve this, we utilized clock glitches as the primary method of fault injection. Our experimental findings reveal that FIAs can induce various effects on PMP configuration registers. By categorizing these effects according to the injection parameters, we demonstrate that specific effects can be reliably achieved under varying injection conditions, often with a high probability of success for an attacker.

Introduction

The Physical Memory Protection (PMP) mechanism, although optional, is widely supported due to its pivotal role in system security and its integration in TEEs. Despite its importance, PMP is vulnerable to physical attacks, such as Fault Injection Attacks (FIA) [1], which include techniques like voltage pulses, electromagnetic pulses, and clock glitching. Studies, such as those by Nashimoto et al. [2], have demonstrated the feasibility of modifying PMP configuration registers on RISC-V processors through clock glitching. While most systems integrate the clock, making direct access challenging for attackers, clock glitching remains a relevant model for understanding fault vulnerabilities and developing countermeasures. The main contributions of this paper are as follows: we investigate the impact of clock glitching on the PMP configuration flow of the RISC-V CVA6 core [3]. Additionally, we classify the effects observed on PMP configuration registers based on fault types and injection parameters.

Experimental setup

The experimental setup, based on [4], uses a system-on-chip (SoC) built with the LiteX framework, featuring a CVA6 core [3], RAM, GPIOs and UART. The SoC is implemented on a Digilent Arty A7-100T FPGA board [5]. Fault injection is conducted with the ChipWhisperer Lite [6], generating a 25 MHz clock signal for glitches. Clock injection parameters are refined to better target the PMP instructions, with *External Offset* ranging from 0 to 100 and *Repeat* set to 1.

In this experiment, we assume an attacker attempts to bypass the PMP mechanism to access sensitive data or execute arbitrary code. The evaluation uses software running in M-MODE to configure a protected PMP region in NAPOT mode, as in [2]. CSR instructions set `pmpcfg0` to `0x99` and `pmpaddr0` to `0x800018F`. Prior to this configuration, all PMP registers are initialized to `0x76`. This setup enforces read-only permissions on a 128-byte memory region starting at `0x20000600` and locks the PMP configuration, preventing further modifications. Fault injection targets the PMP configuration instructions, where `pmpaddr0` spans 2 to 5 lines and `pmpcfg0` is configured in a single instruction (line 6).

```
1 TRIGGER high
2 @ret = (&@base) >> 2
3 @ret &= ~(size >> 3)
4 @ret |= ((size >> 3) - 1)
5 csrw pmpaddr0, @ret
6 csrs pmpcfg0, (0x99)
7 modify the protect value
```

Listing 1: *Target Pseudo-Code*

Results analysis

Using clock glitches as the primary fault injection method, experiments conducted on an ARTY A7-100T FPGA board resulted in 2,126 modifications to PMP configuration registers out of 836,381 injections, with 1,877 allowing unauthorized memory writes to protected regions. The fault effects were classified into three groups:

- Group G1 includes 51 faults that cause complex effects, such as storing random or faulty values in multiple PMP configuration registers, or correct values in unintentional registers.
- Group G2 includes 1,708 faults impacting a single PMP register, either `pmpcfg0` or `pmpaddr0`. Most faults occurred on `pmpaddr0`, with multiple bit-flips being the most common effect, followed by single bit-flips and register resets with 1,668 fault.

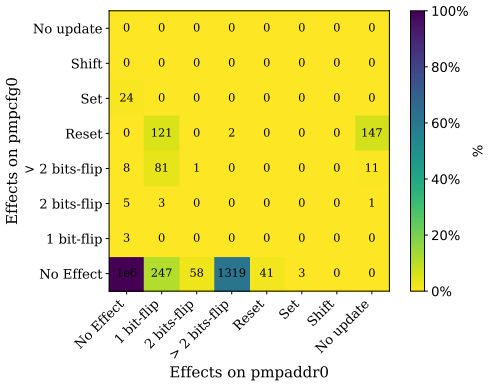

Figure 1: *Fault impact on Groups* `G2` *&* `G3`

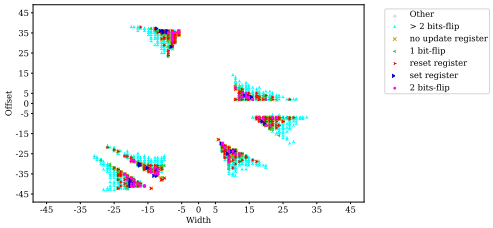

Figure 2: *Fault effects on the both PMP registers in regards with injection parameters between Width and Offset*

The first column and last row of Figure 1 show the number of fault injections causing each specific fault effect for this group. This is attributed to the multi-instruction configuration required for `pmpaddr`*0* compared to the single-instruction configuration for `pmpcfg`*0* with only 40 faults.

- Group G3 includes 367 faults affecting both `pmpcfg`*0* and `pmpaddr`*0* without impacting other registers. These couple of effects are built around bit-flip, register reset or no update value in the configured register. Figure 1 shows that a fault injection can cause to couple of effects in the two configured registers.

It is worth noting that 452 injections over 2,126 cause to single bit-flips in a least one register. However, multiple bits-flip is the most common effect on `pmpaddr`*0* among all various types of effects.

Figure 2 shows the types of fault effects in relation with *Width* and *Offset* clock injection parameters, revealing six sensitive zones where all fault effects are observable (one symbol per faulty register). Subzones within these zones correspond to specific effects. Both figures 3 and 4 illustrate types of fault effects localization for a single PMP register based on *Width* and *External Offset* parameters. The *External Offset*, ranging from 0 to 100 clock cycles, defines the delay between a trigger and a glitch. This analysis demonstrates that carefully tuning parameters like *Width*, *Offset*, and *External Offset* can reliably induce specific fault effects, highlighting PMP vulnerabilities to FIA and offering insights into their exploitation.

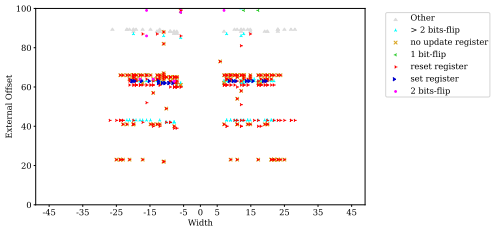

Figure 3: *Fault effects on* `pmpcfg`*0 in regards with injection parameters between Width and External Offset*

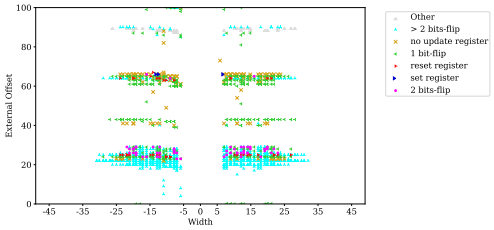

Figure 4: *Fault effects on* `pmpaddr`*0 in regards with injection parameters between Width and External Offset*

Future Works

We recognize that further studies are needed to address multiple bit-flips, shifts, set/reset register, and complex fault effects (group `G1`). Register no update could potentially be explained by the *instruction skip* fault model, while complex fault effects may result from faulty control signals or register indexes. However, additional experiments are necessary to validate these hypotheses. Future work will also focus on analyzing clock injection parameters in relation to instructions within the pipeline stages. In addition, it would be essential to compare the effects of different fault types with other FI methods, such as EM and voltage analysis.

References

[1] Karaklajić, Duško and Schmidt, Jörn-Marc and Verbauwhede, Ingrid. "Hardware Designer's Guide to Fault Attacks". In: *IEEE Transactions on Very Large Scale Integration Systems* (2013). DOI: `10.1109/TVLSI.2012.2231707`.

[2] Shoei Nashimoto et al. "Bypassing Isolated Execution on RISC-V using Side-Channel-Assisted Fault-Injection and Its Countermeasure". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)* (2021). DOI: `10.46586/tches.v2022.i1.28-68`.

[3] OpenHW Group. *cva6*. openhwgroup. 2019. URL: `https://github.com/litex-hub/pythondata-cpu-cva6`.

[4] Kévin Quénéhervé et al. "Exploring Fault Injection Attacks on CVA6 PMP Configuration Flow". In: *2024 27th Euromicro Conference on Digital System Design (DSD)*. 2024. DOI: `10.1109/DSD64264.2024.00015`.

[5] *Arty A7*. Digilent. 2019. URL: `https://digilent.com/reference/programmable-logic/arty-a7/reference-manual?redirect=1`.

[6] *Chipwhisperer Lite*. NewAETech. 2019. URL: `https://rtfm.newae.com/Capture/ChipWhisperer-Lite/`.