Characterizing Clock Glitching Attacks on CVA6 PMP Configuration Flow

Kévin Quénéhervé, William Pensec, Tanguy Philippe, Vianney Lapotre

Characterizing Clock Glitching Attacks on CVA6 PMP Configuration Flow

Kévin QUÉNÉHERVÉ, William PENSEC, Philippe TANGUY, Vianney LAPÔTRE
Université Bretagne Sud, UMR6285, Lab-STICC, Lorient, France, firstname.lastname@univ-ubs.fr

Context

Fault-Injection Attacks (FIA) [1] pose significant threats to the security and reliability of embedded systems. Nashimoto et al. [2] have illustrated the possibility to modify the Physical Memory Protection (PMP) configuration registers on a RISC-V processor through FIA using clock glitching. However, their study did not delve into the consequences of faults on these PMP configuration registers. Thus, in this study, we investigate the effects of clock glitching on the PMP configuration flow of a CVA6 RISC-V core.

CVA6 PMP

Physical Memory Protection (PMP) [3] allows to protect 16 maximum memory regions by assigning specific access permissions. Each memory region is governed by 2 Control Status Registers (CSRs):

- `pmpcfg` controls access permissions (read, write, execute) and addressing mode.
- `pmpaddr` stores region size and base address based on addressing mode.

In the CVA6 core, PMP configuration flow through pipeline stages is depicted in the Figure 1. Registers maintaining the PMP configuration are located in the *CSR Write* module of the *Commit*.

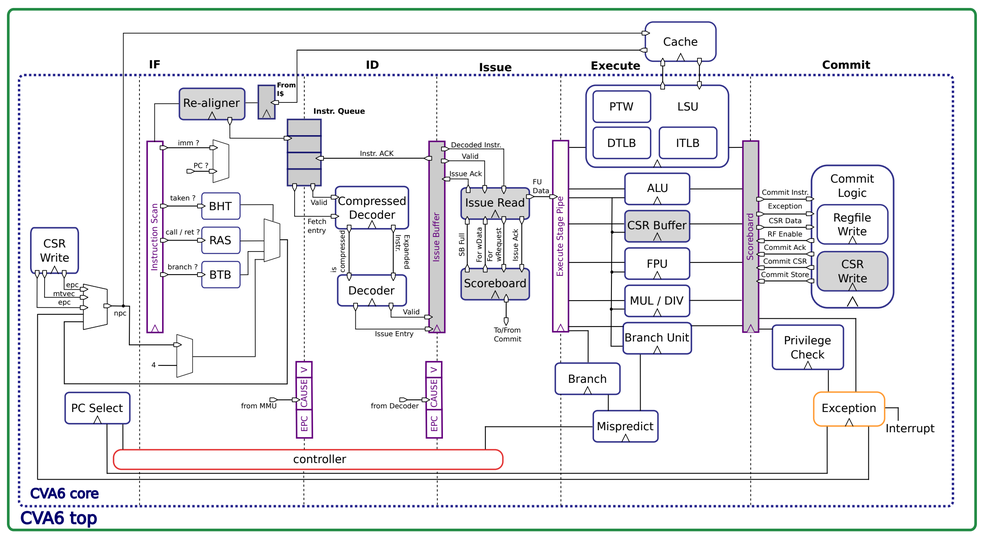

Figure 1: CVA6 architecture

Experimental setup

Our experimental setup relies on the Chipwhisperer Lite to inject faults through clock glitching targeting the Arty A7-100T FPGA board.
Each fault injection campaign explore a set of parameters presented in Figure 2. There are 1,970,001 injections per campaign.
Figure 3 illustrates the target pseudo code including instruction for PMP configuration.

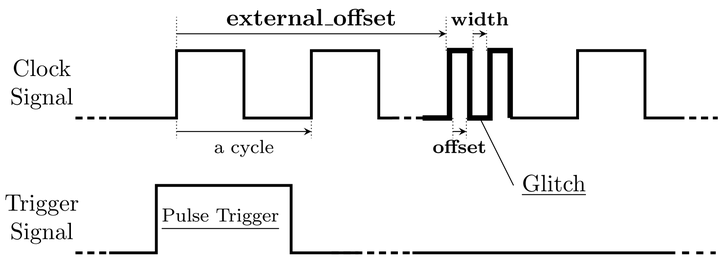

Figure 2: Clock Glitch principles parameters

```
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;
```

Figure 3: Target pseudo code

Effects of FIA on PMP configuration

Results show that among performed injections, 4,267 resulted in modifying the PMP configuration to allow write operations into protected memory.s Among these:

- 1,407 affected `pmpcfg0` & `pmpaddr0` exclusively detailed in the table 1.
- 1,587 affected either `pmpcfg0` or `pmpaddr0` exclusively detailed in the table 2.
- The remaining 1,273 injections resulted in *complex* effects (`G1`).

Figure 3 highlights the correlation between observed fault effects and the injection parameters *Width* and *Offset*.

- Six sensitive zones are identified where all fault effects can occur
- Within each sensitive zone, specific sub-zones corresponding to parameters leading to a particular effect can be delineated.
- Notably, the single bit-flip effect predominantly occurs at the boundaries of these zones.

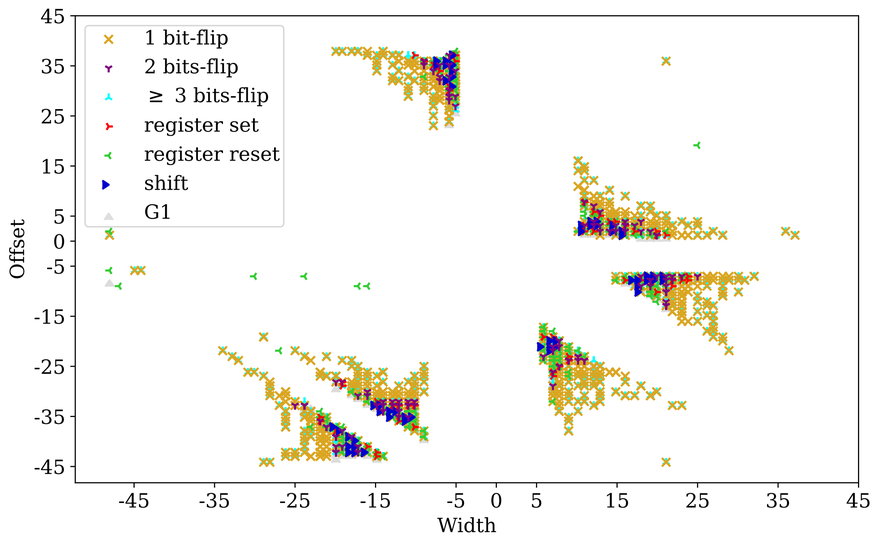

Figure 4: Fault effects about injection parameters

Table 1: `pmpcfg0` & `pmpaddr0`

	shift / 1 bit-flip	1 bit-flip / 1 bit-flip	2 bits-flip / 1 bit-flip	1 bit-flip / 2 bits-flip	≥ 3 bits-flip / 1 bit-flip	1 bit-flip / ≥ 3 bits-flip	2 bits-flip / 2 bits-flip	≥ 3 bits-flip / ≥ 3 bits-flip	reset registers
pmpcfg0 / *pmpaddr0*	87	46	22	34	608	22	2	37	549

Table 2: `pmpcfg0` or `pmpaddr0`

	1 bit-flip	2 bits-flip	≥ 3 bits-flip	register set	register reset
pmpcfg0	23	9	3	17	0
pmpaddr0	571	67	508	44	345

Conclusion & perspectives

This study shows clock glitch-based FIA on the CVA6 PMP configuration, identifying exploitable effects. Consequently, attackers could manipulate injection parameters to achieve desired effects with a high probability. In future works, experiments will be conducted to comprehend the impacts of faults and develop customized countermeasures for each pipeline stage. Additionally, we plan to extend the approach to other critical processor execution flows.

Bibliography

[1] H. Bar-El *et al.*, "The sorcerer's apprentice guide to fault attacks," *Proceedings of the IEEE*, 2006.
[2] S. Nashimoto *et al.*, "Bypassing Isolated Execution on RISC-V using Side-Channel-Assisted Fault-Injection and Its Countermeasure," *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)*, 2021.
[3] "Volume 2, Privileged Specification version 20211203," RISC-V. (), [Online]. Available: `https://riscv.org/technical/specifications/`.