Unveiling the Invisible Threads: Dynamic Information Flow Tracking and the Intriguing World of Fault Injection Attacks

William PENSEC, Vianney LAPÔTRE, Guy GOGNIAT
Lab-STICC, UMR 6285, Université Bretagne Sud, Lorient, France
firstname.lastname@univ-ubs.fr

Information Flow Tracking in a RISC-V processor

Different types of IFT [1, 2]:

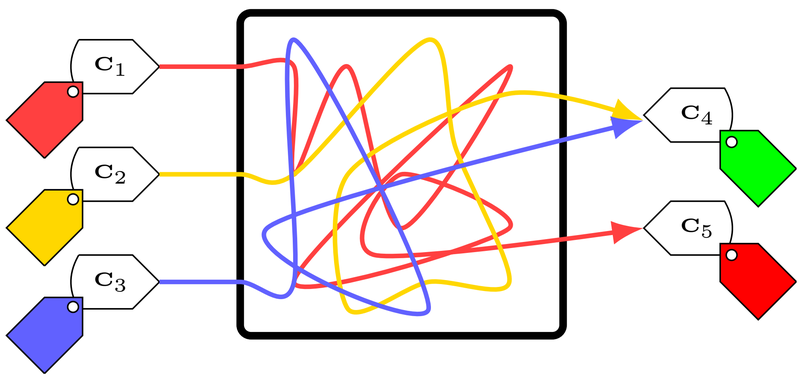
- Static or dynamic
- Software, hardware (in-core, off-core [3] (dedicated CPU, co-processor)) or mixed

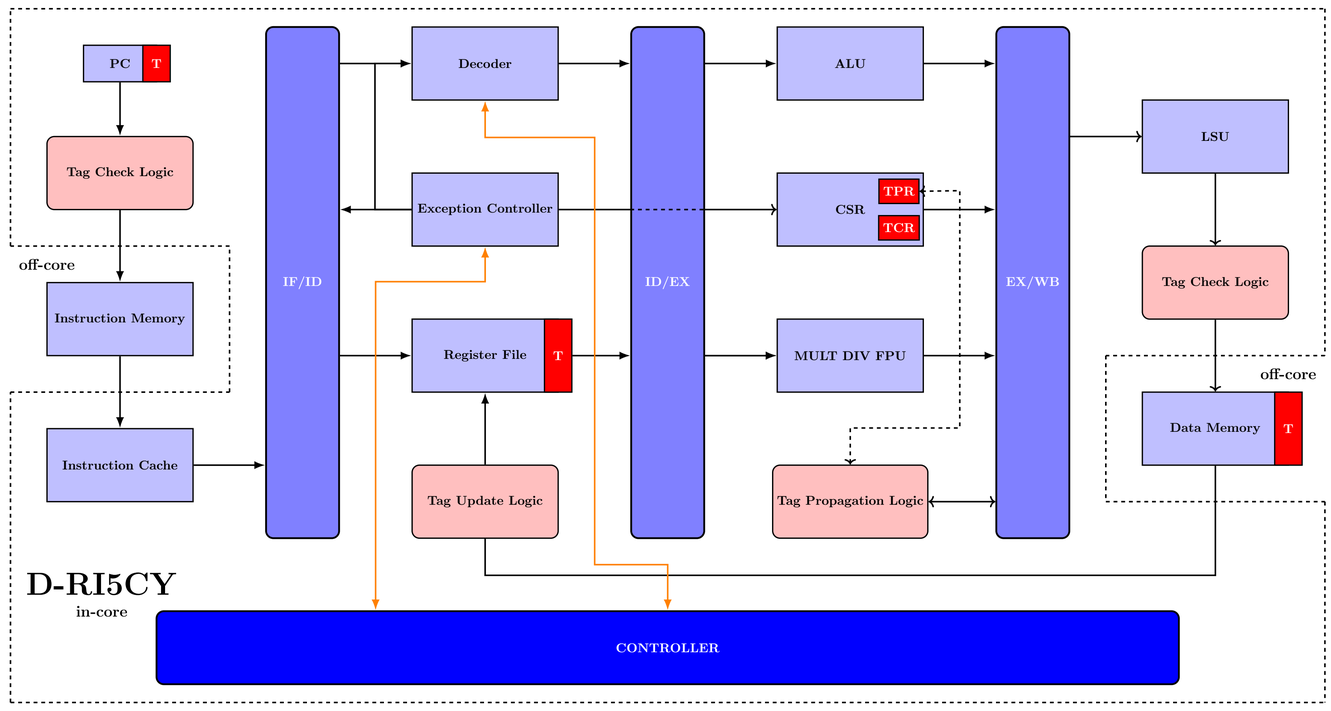
Three steps

- Tag initialization
- Tag propagation
- Tag verification

Levels of IFT

- Application level
- OS level
- Architectural level



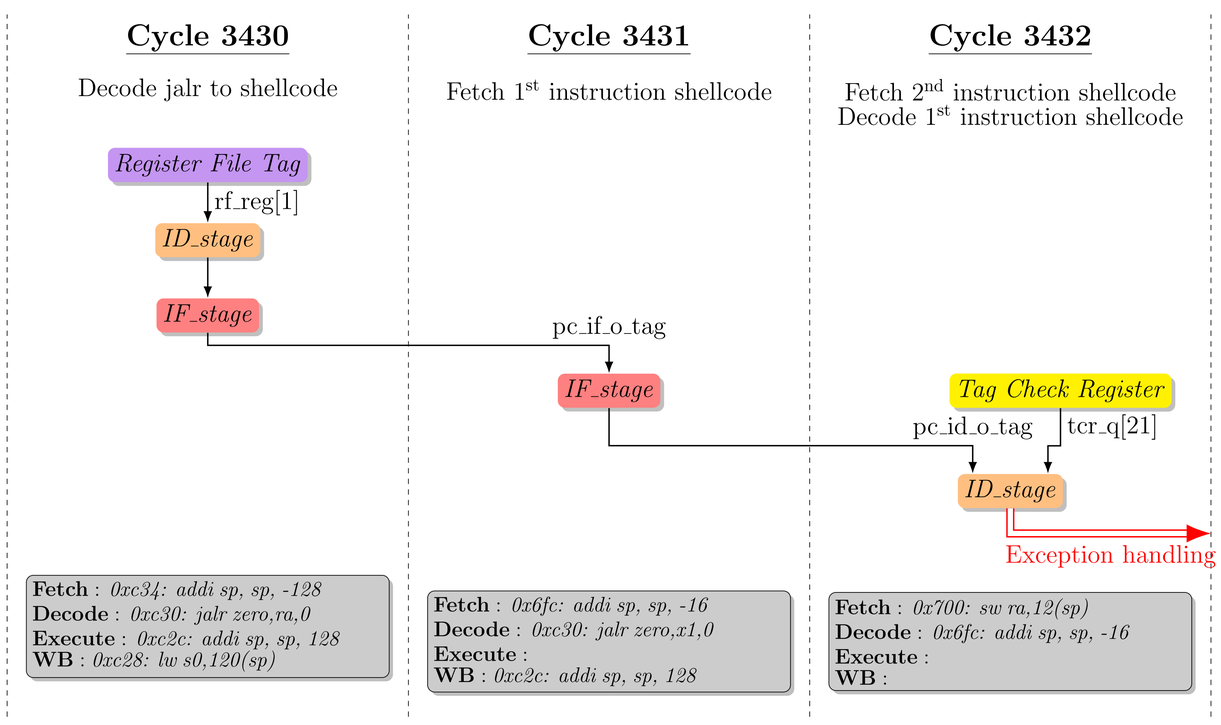


D-RI5CY [4] has been developed by researchers from Columbia University, and University of Turin.

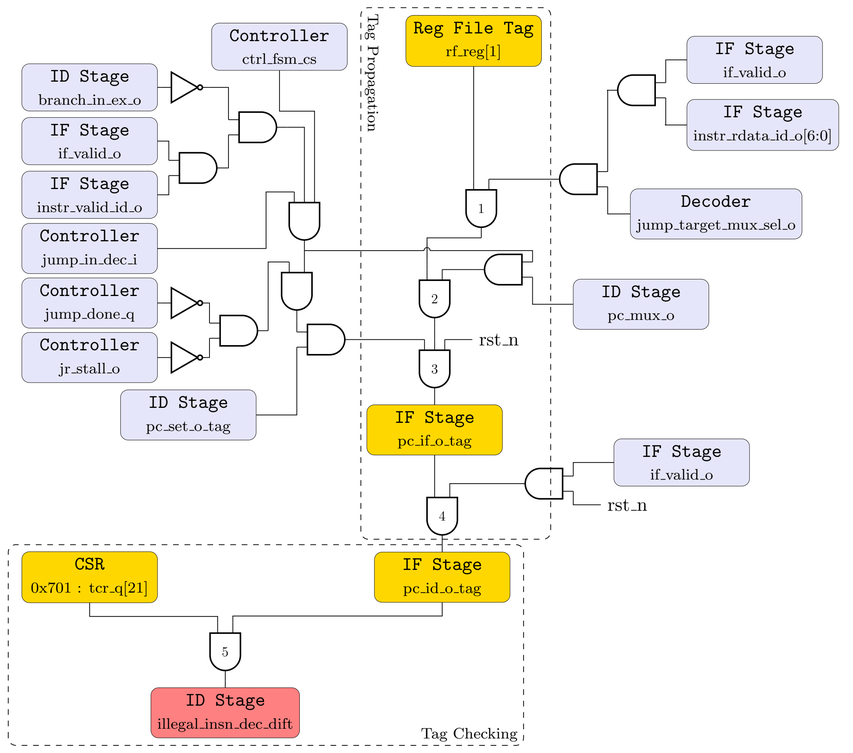
Physical Attacks against DIFT

We consider an attacker able to:

- perform physical attacks to defeat the DIFT mechanism and realize a software attack,
- inject faults in registers associated to the DIFT-related components: set to 0, set to 1 and bit-flips.



Tag propagation in a buffer overflow attack



Logic description of the exception driving in a buffer overflow attack

Results

We used CABA fault injection simulations to evaluate the sensitivity of DIFT. A total of 4212 simulations have been performed. About 2.21% lead to successful attacks.

55 registers DIFT-related. 13 critical registers highlighted with injection campaign. About 34.41% are due to set to 0 fault type, 11.83% are due to set to 1 fault type and 53.76% are due to a bitflip. About 2.59% of the simulated injections delay the DIFT exception.

	Crash	NSTR	Delay	Success	Total
Buffer overflow	0	1362	20	22 (1.57%)	1404
Format string	0	1743	77	52 (2.78%)	1872
Compare/Compute	0	905	12	19 (2.03%)	936

	Cycle 3428			Cycle 3429			Cycle 3430			Cycle 3431			Cycle 3432		
	set0	set1	bitflip	set0	set1	bitflip	set0	set1	bitflip	set0	set1	bitflip	set0	set1	bitflip
pc_if_o_tag										✓		✓			
rf_reg[1]							✓		✓						
tcr_q	✓			✓			✓			✓			✓		
tcr_q[21]			✓			✓			✓			✓			✓
tpr_q	✓	✓		✓	✓										
tpr_q[12]			✓			✓									
tpr_q[15]			✓			✓									

Buffer overflow: success per register, fault type and simulation time

Perspectives

- Implement and evaluate countermeasures as simple parity and Hamming code (work in progress) taking into account constraints (performance, area, consumption) to protect critical computation related to DIFT.
- Extend the study to the entire D-RI5CY core and take into account a more complex threat model (multi-faults models).
- Perform a fault injection campaign targeting a FPGA implementation.

Bibliography

[1] W. Hu *et al.*, "Hardware information flow tracking," *ACM Computing Surveys*, 2021. DOI: 10.1145/3447867.

[2] K. Chen *et al.*, "Dynamic information flow tracking: Taxonomy, challenges, and opportunities," *Micromachines*, 2021. DOI: 10.3390/mi12080898.

[3] H. Kannan *et al.*, "Decoupling dynamic information flow tracking with a dedicated coprocessor," in *International Conference on Dependable Systems & Networks*, IEEE, 2009. DOI: 10.1109/DSN.2009.5270347.

[4] C. Palmiero *et al.*, "Design and implementation of a dynamic information flow tracking architecture to secure a RISC-V core for IoT applications," in *High Performance Extreme Computing*, 2018. DOI: 10.1109/HPEC.2018.8547578.