REHAD: Using Low-Frequency Reconfigurable Hardware for Cache Side-Channel Attacks Detection

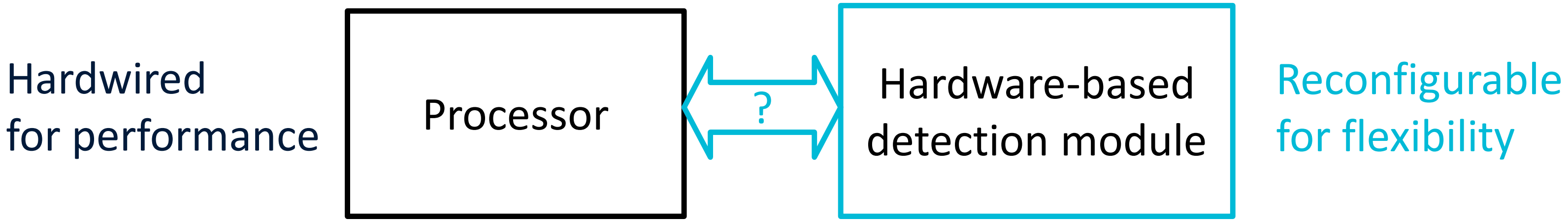
Yuxiao MAO
Vincent MIGLIORE
Vincent NICOMETTE

SILM 2020

Motivation

- Lots of low-level attacks in recent year
  - Cache side-channel attacks, Spectre and Meltdown …
- Solutions proposed so far
  - Software (**Pros:** flexible. **Cons:** high overhead, difficulty of getting low-level information)
  - Dedicated hardware (**Pros:** fine tuned mitigation, efficient. **Cons:** impossible to adapt to new attacks)

➢ Our solution: REHAD (REconfigurable Hardware for Attacks Detection)

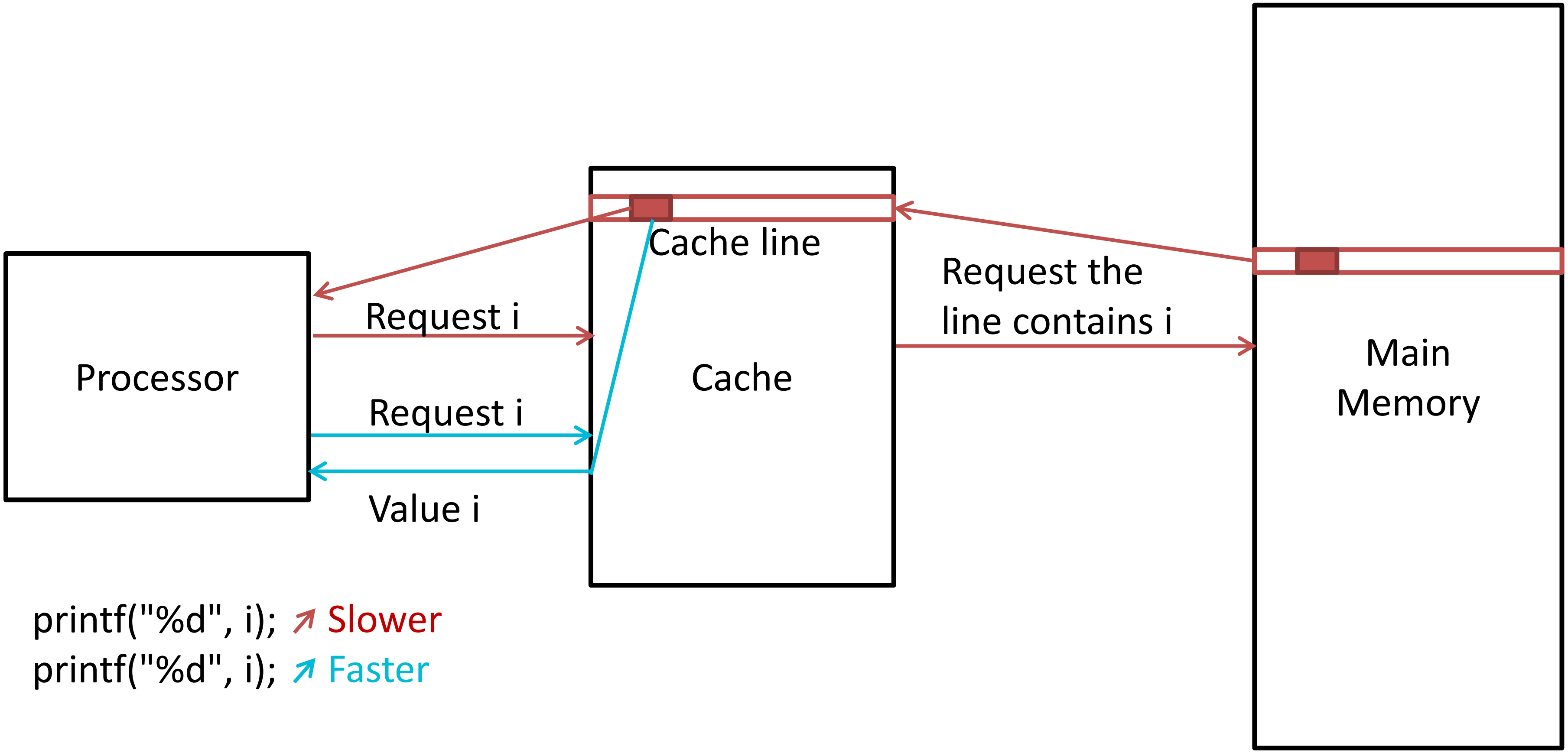


- Challenges
  - Frequency gap between the processor core and reconfigurable hardware
  - Covering as many attacks as possible
  - Amount and type of information exchanged between the processor and the detection module

Outline

- Motivation
- Cache side-channel attacks
- REHAD architecture
- Implementation
- Conclusion and future work

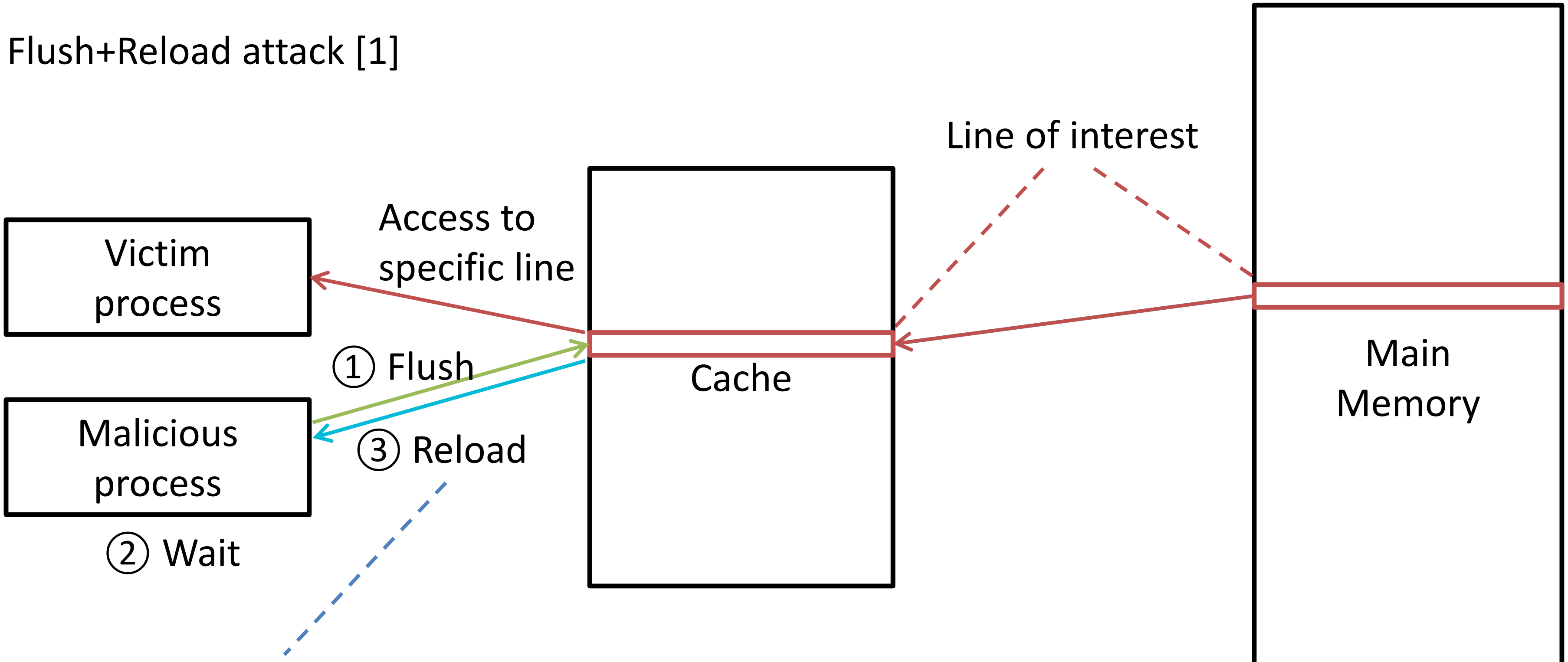
Cache



printf("%d", i); ↗ Slower
printf("%d", i); ↗ Faster

Cache side-channel attacks

Flush+Reload attack [1]



Measure time of [Reload]

If Victim accessed [line] => Fast

If not => Slow

[1] Y. Yarom and K. Falkner, "FLUSH+RELOAD: a High Resolution, Low Noise, L3 Cache Side-Channel Attack," in Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, Aug. 2014, pp. 719–732.

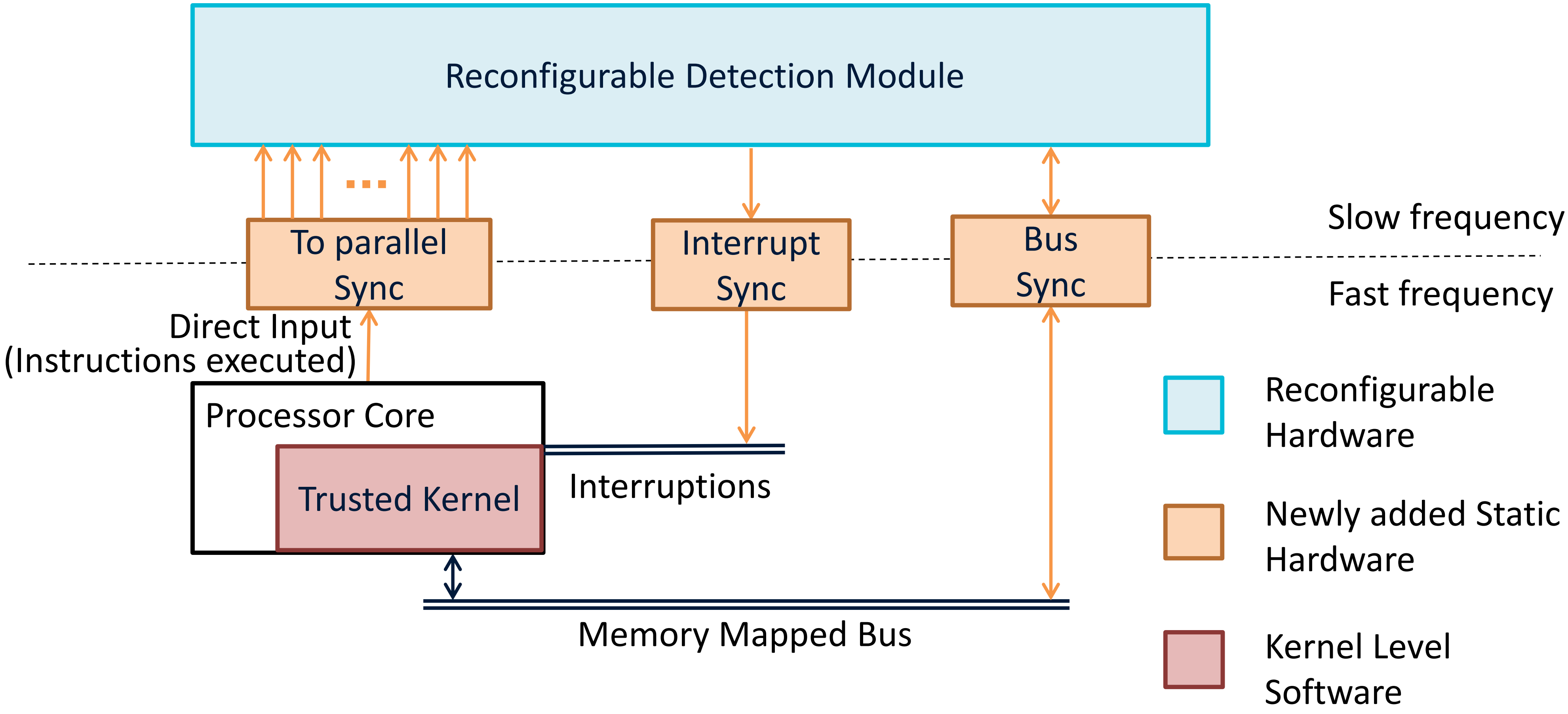
Cache side-channel defenses

Common protection methods against cache side-channel attacks

		Software	Hardware	
			Hardwired	Highly Reconfigurable
Prevention		• Constant time programming • Enhancing time and space isolation • Limiting timer utilization	• Redesign shared hardware architecture • Clock modification	
Detection	Static	• Binary file analyzing		
	Dynamic	• Periodically monitoring using Hardware Performance Counters	• Shared hardware events monitoring	• REHAD

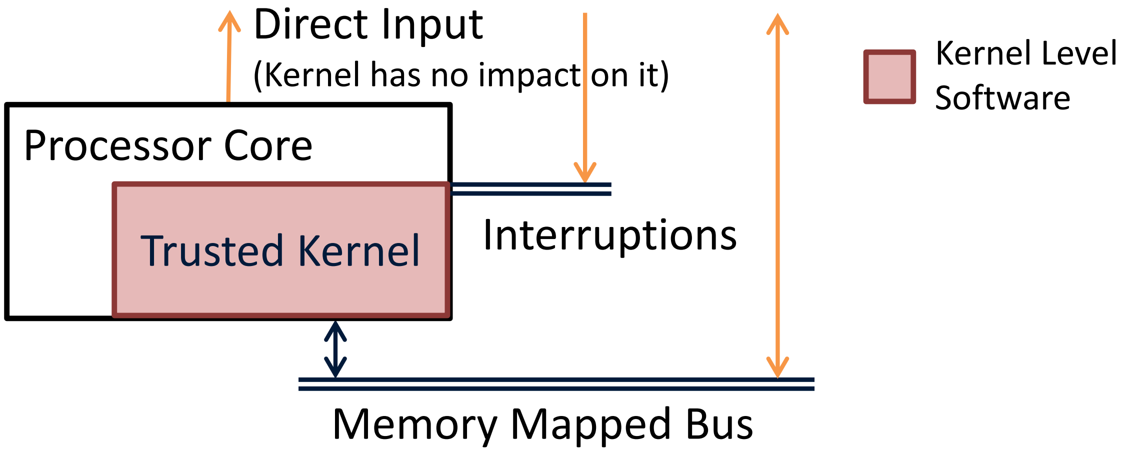
Exists for other attacks such as ROP or malware detection, but without considering the frequency gap

REHAD architecture

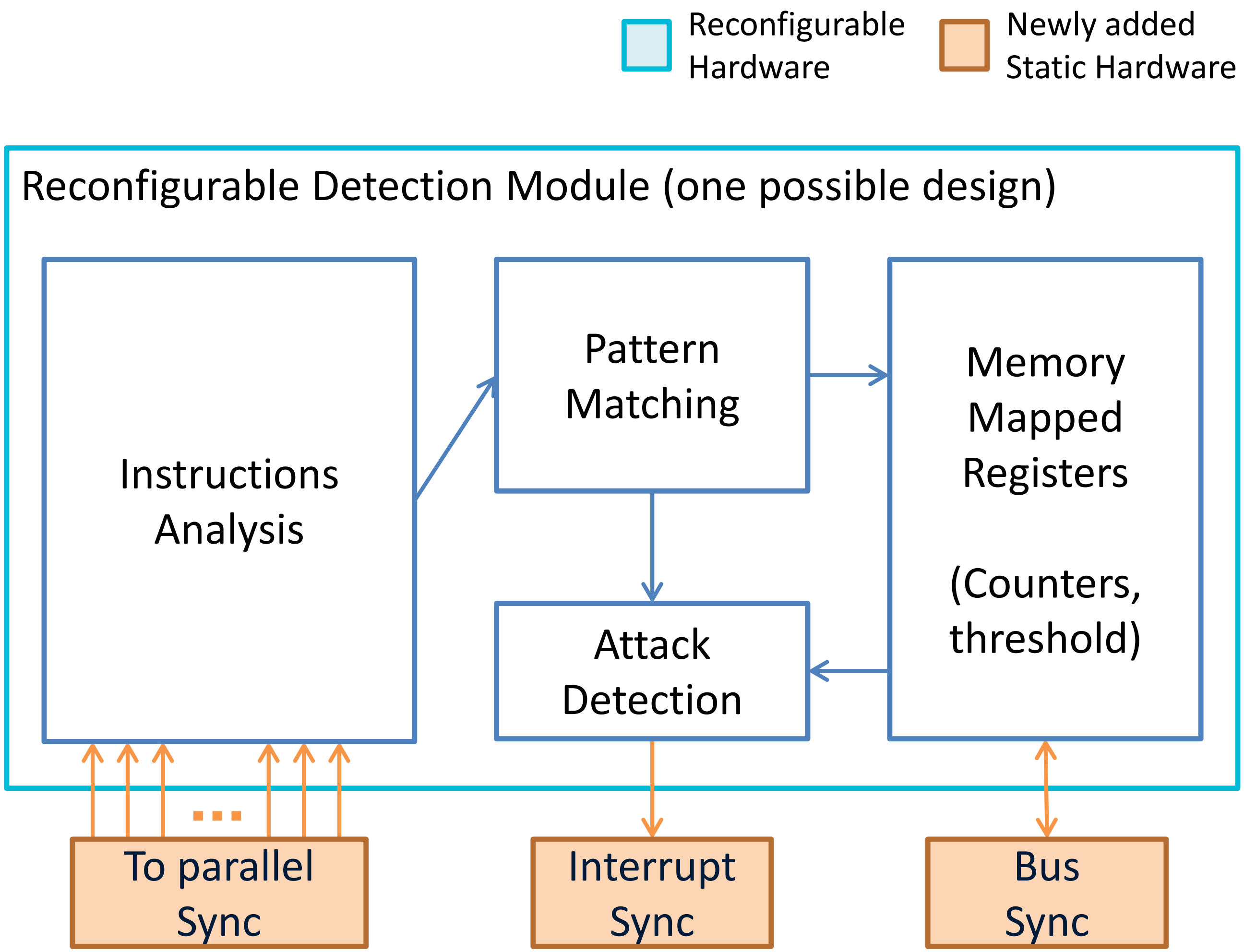


REHAD: Trusted Kernel

- Configures
  - Detection mode
  - Detection threshold
- Provides information
  - Process number
  - Hardware Performance Counters
- Handles interruptions
  - Gets context information
  - Decides activation of protection mechanism
- Reconfigures
  - In order to detect new attacks, like a patch for software



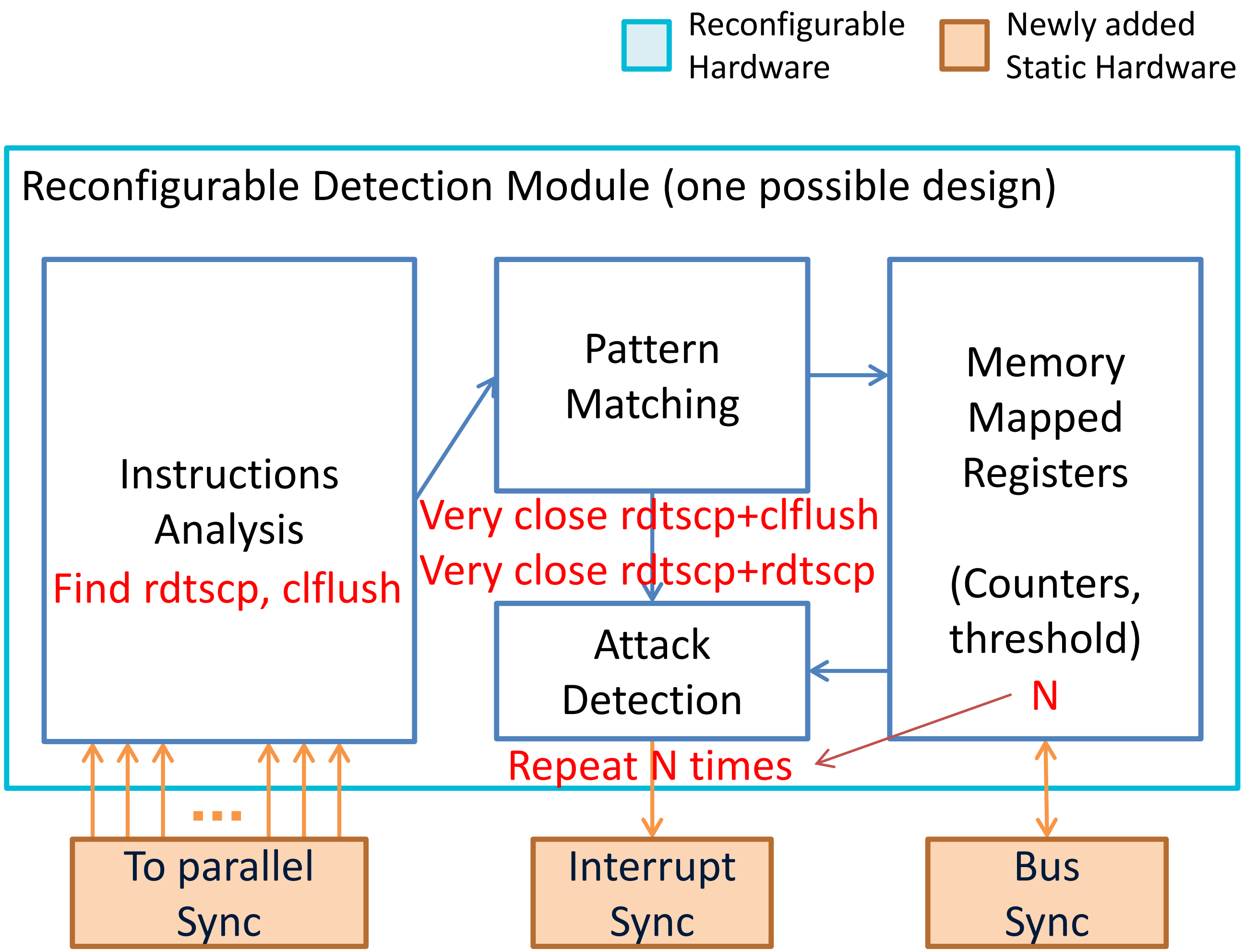
REHAD: Detection Module



Implementation: Detect Flush+Reload

```
mfence
rdtscp
mov %eax, %esi
mov (%ebx), %eax
rdtscp
sub %esi, %eax
clflush (%ebx)
```

Flush+Reload attack on x86 from Mastik Toolkit [2]



[2] Y. Yarom, "Mastik: A Micro-Architectural Side-Channel Toolkit," 2016. https://cs.adelaide.edu.au/yval/Mastik/

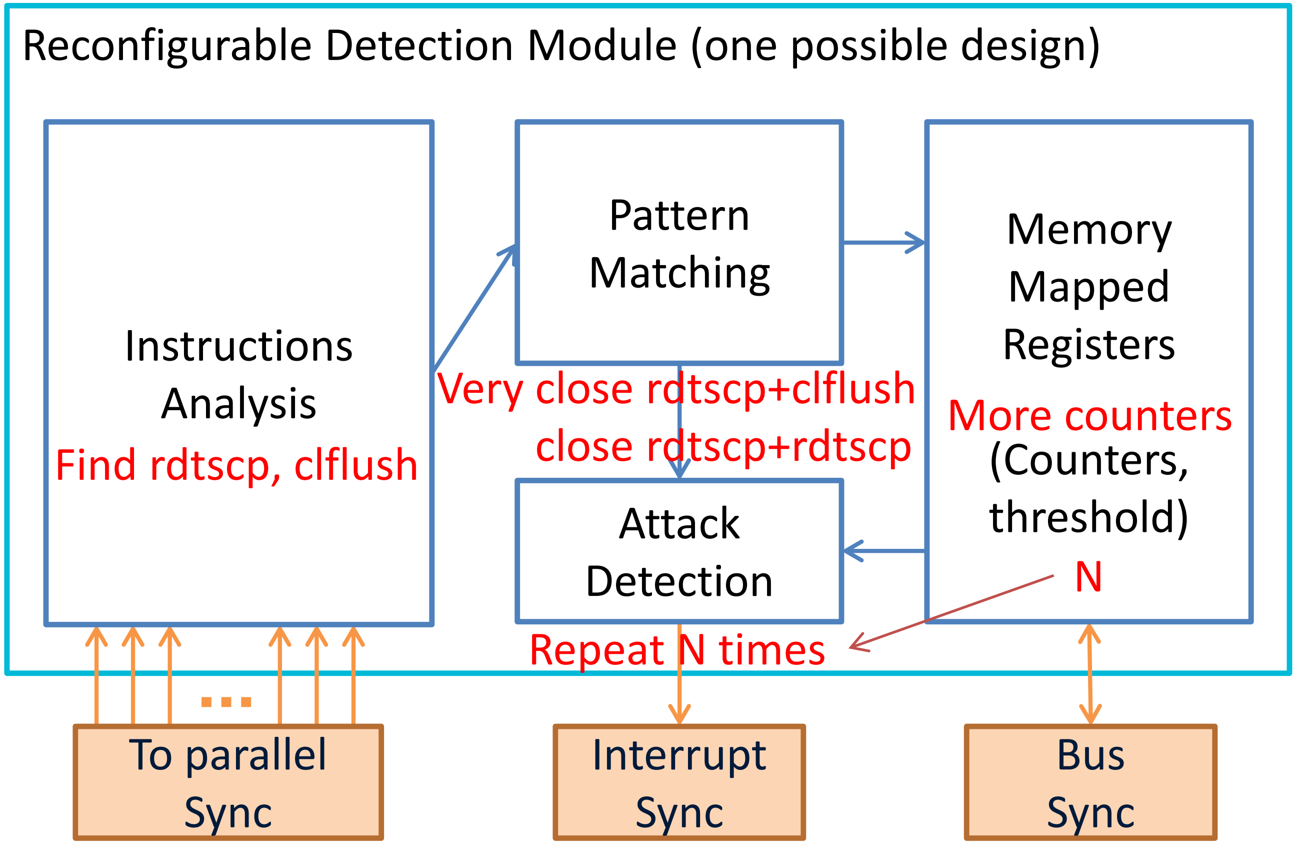
Implementation: Detect Prime+Probe

🟦 Reconfigurable Hardware

🟧 Newly added Static Hardware

```
rdtscp
mov %eax, %esi
mov (%rdi), %rax
mov (%rax), %rax
...
mov (%rax), %rax
mov (%rax), %rdi
rdtscp
sub %esi,%eax
```

Prime+Probe attack [3]
on x86 from Mastik Toolkit



Reconfigurable Detection Module (one possible design)

Instructions Analysis

Find rdtscp, clflush

Pattern Matching

Very close rdtscp+clflush

close rdtscp+rdtscp

Memory Mapped Registers

More counters

(Counters, threshold)

N

Attack Detection

Repeat N times

To parallel Sync

Interrupt Sync

Bus Sync

[3] D. A. Osvik, A. Shamir, and E. Tromer, “Cache Attacks and Countermeasures: The Case of AES,” in Topics in Cryptology CTRSA 2006, vol. 3860. Berlin, Heidelberg: Springer, Feb. 2006, pp. 1–20.

Implementation results

- Softcore Processor: ORCA [4]
  - ISA: RISC-V, RV32IM
  - Cache: L1 only (16 lines of 32 bytes)
- Hardware settings
  - Xilinx ML605 Evaluation Board (FPGA Virtex-6)
  - Frequency: 80 MHz (processor) / 5 MHz (detection module)
- Detection module
  - Configuration 1: Detect Flush+Reload attack
    - Resources usage: 208 LUTs, 65 FFs
  - Configuration 2: Detect Flush+Reload and Prime+Probe attacks
    - Resources usage: 215 LUTs, 70 FFs

[4] VectorBlox, “Orca,” 2019. https://github.com/VectorBlox/orca

Conclusion

- Low-frequency reconfigurable hardware for detection
  - Can be updated to fit new attacks and variants
- Hardware & software hybrid protection
- Instruction-based
  - Does not depend on specific shared hardware (e.g., cache)
  - Can be adapted to different processor cores, even different ISA
- No user program / compiler modification

- Drawbacks
  - Requires processor modification in order to output instructions
  - Requires additional resources on synchronization

Future Work

- Other softcore processor
  - Now on Rocket-Chip
- Multicore, multithread
- Other attacks
  - Microarchitectural Timing Attacks
  - Transient Execution Attacks (Spectre and Meltdown)
  - Return Oriented Programming
  - Malware signature

Thank you. Questions?

Yuxiao MAO	yuxiao.mao@laas.fr
Vincent MIGLIORE	vincent.migliore@laas.fr
Vincent NICOMETTE	vincent.nicomette@laas.fr

SILM 2020