RISC-V SoC 환경에서 RoCC를 이용한 캐시 부채널 공격

Cache Side-Channel Attacks Exploiting RoCC Interface on RISC-V SoC Platform

황예원o 서태원 구건재

고려대학교 컴퓨터학과

KOREA UNIVERSITY

Computer System Architecture Lab

Outline

RISC-V Architecture

- RISC-type open-source ISA
- RISC-V ISA can be extended by including modular instruction sets
- Gaining popularity: attracting interest from industry and academia

RISC-V Architecture

- **Open Source:** RISC-V provides access to its design specifications, **allowing full customization**
- **Flexibility:** The simple, modular instruction set allows users to add custom instructions, optimizing the processor **for specific domains**
- **Simplicity:** RISC-V can simplify chip design flows, **reduce complexity**, and optimize hardware-software interactions.

RISC-V Architecture

- **Open Source:** RISC-V provides access to its design specifications, **allowing full customization**

- **Flexi** s users to rocessor **for specific domains**

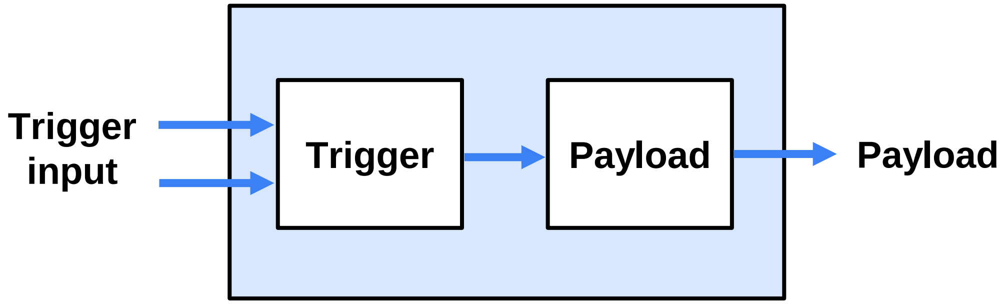
We need to focus on the security issues of the *customizable open-source ISA*

- **Simplicity:** RISC-V can simplify chip design flows, **reduce complexity**, and optimize hardware-software interactions.

고려대학교 KOREA UNIVERSITY

Hardware Trojan

- Malicious modifications of the circuitry in an integrated circuit
- A trigger determines when the hardware Trojan activates
- A payload determines what happens when the hardware Trojan is triggered



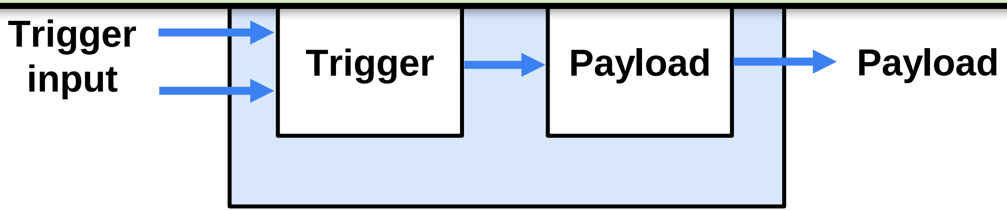
Hardware Trojan

Hardware Trojan

- Malicious modifications of the circuitry in an integrated circuit
- A trigger determines when the hardware Trojan activates
- A ed

▪ Unverified or untrusted third-party IPs may include hardware Trojans.

→ Can leak sensitive information or cause malfunctions

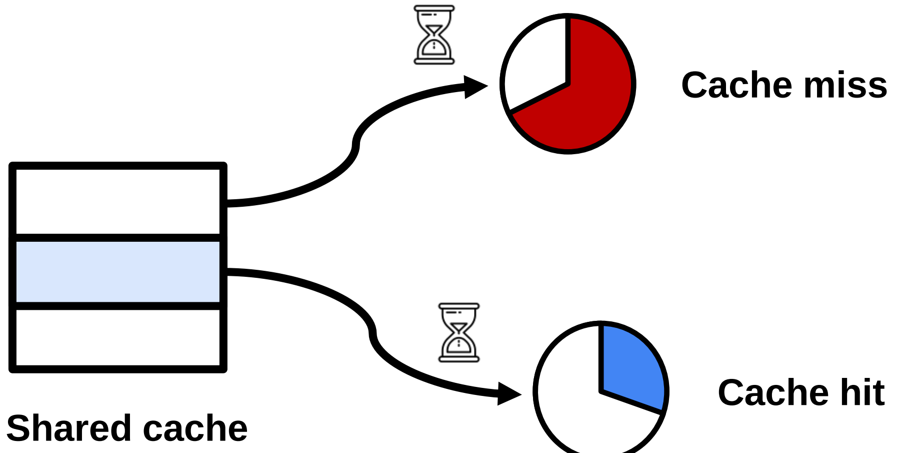


Hardware Trojan

Cache Side-Channel Attack

- Methods that infer program activities by examining cache access patterns
- Exploits the difference between hit and miss access times in the cache
- Can leak information by exploiting hardware vulnerabilities

→ Impossible to fully prevent with software-only solutions



Cache Side-Channel Attack

- Methods that infer program activities by examining cache access patterns
- Exploits the difference between hit and miss access times in the cache
- Can leak information by exploiting hardware vulnerabilities

  → Impossible to fully prevent with software-only solutions

Common methods

- Flush+Reload
- Prime+Probe
- Evict+Time
- Flush+Flush

⋮

Exploits page sharing mechanisms in the last level cache (LLC)

고려대학교 KOREA UNIVERSITY

Three Steps of Flush+Reload Attack

1. Initialization

   Flush cache line using clflush instruction

2. Waiting

   Wait until victim execute

3. Recovering

   Retrieve information by measuring time

Three Steps of Flush+Reload Attack

1. **Initialization**

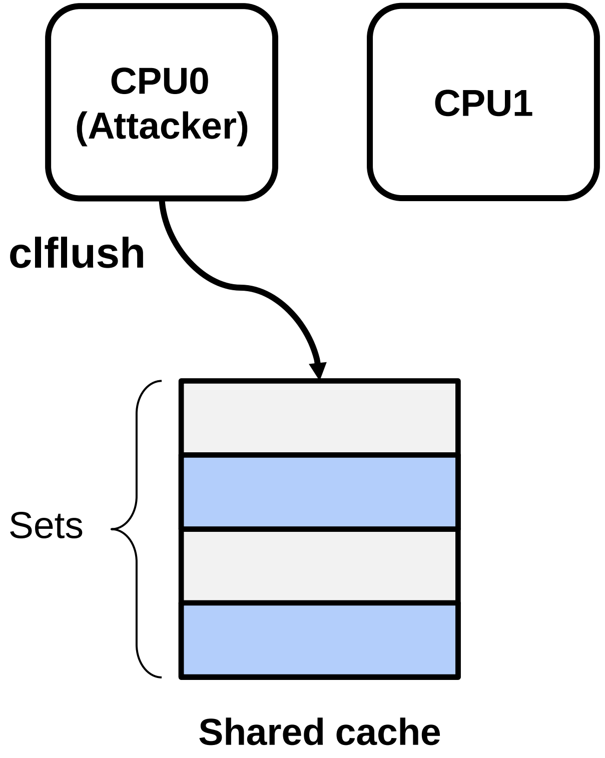
   Flush cache line using clflush instruction

2. Waiting

   Wait until victim execute

3. Recovering

   Retrieve information by measuring time



Three Steps of Flush+Reload Attack

1. Initialization

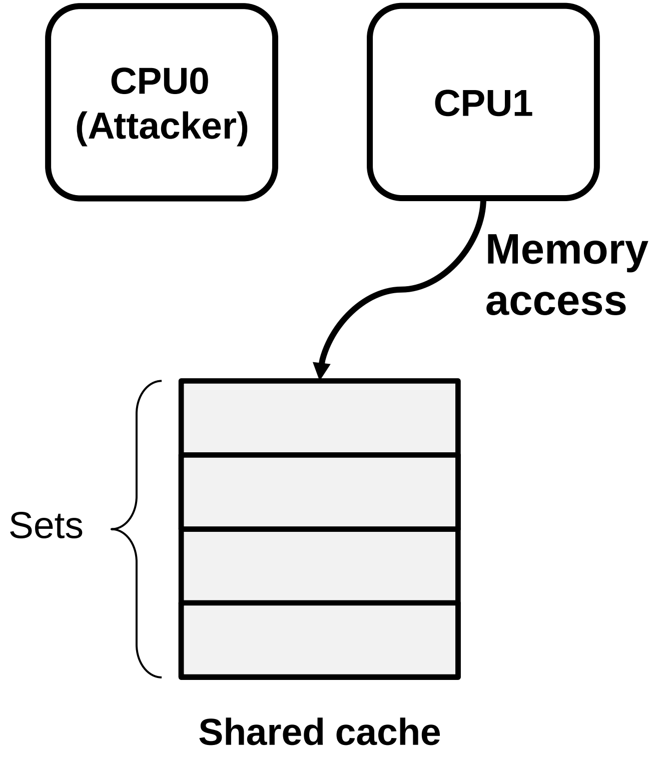
   Flush cache line using clflush instruction

2. **Waiting**

   Wait until victim execute

3. Recovering

   Retrieve information by measuring time



KOREA UNIVERSITY 고려대학교

Three Steps of Flush+Reload Attack

1. Initialization

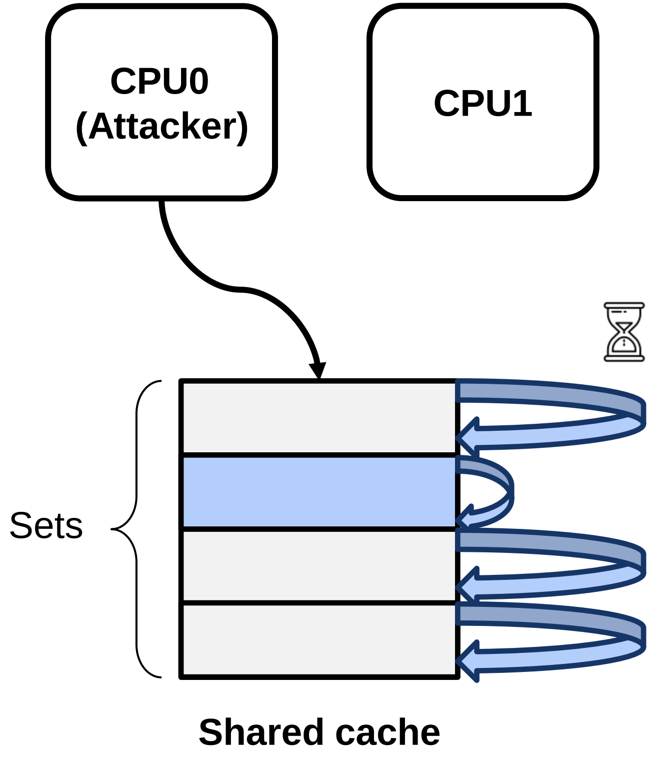
   Flush cache line using clflush instruction

2. Waiting

   Wait until victim execute

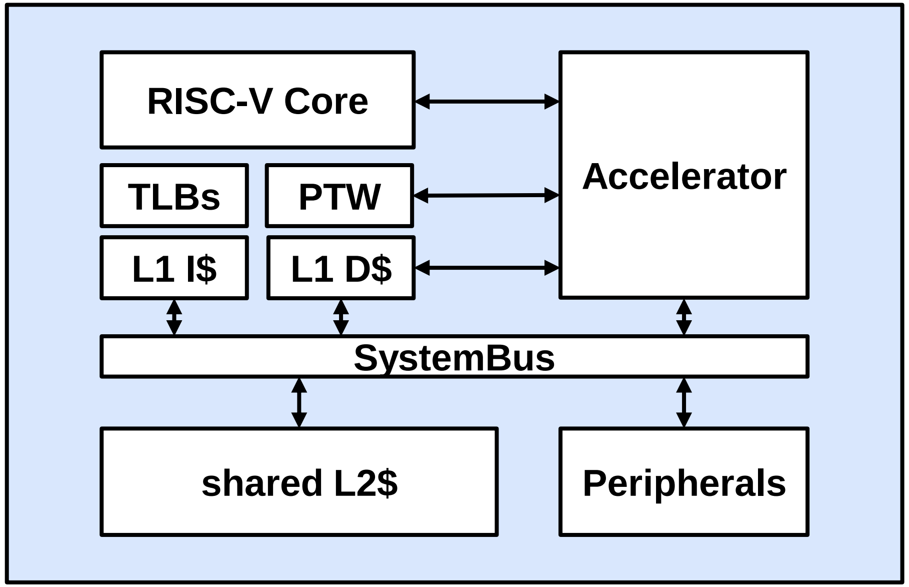
3. **Recovering**

   Retrieve information by measuring time



RISC-V SoC Platform

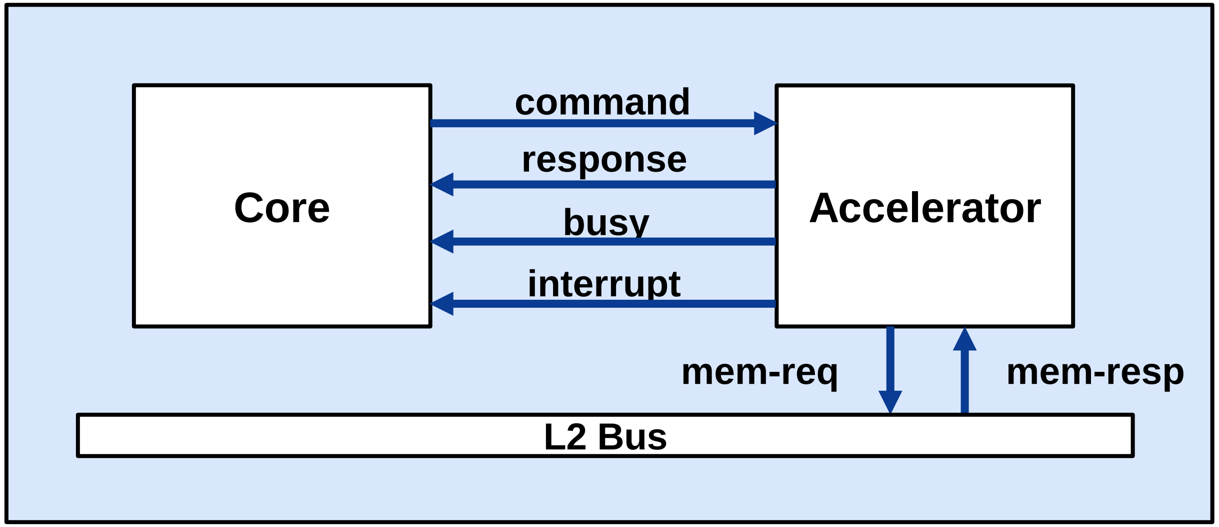
- **RISC-V SoCs can include accelerators**
- **Accelerators can directly access L2 cache via RoCC interface**



KOREA UNIVERSITY 고려대학교

RoCC Interface

- Rocket Custom Coprocessor (RoCC) interface is the decoupled interface between RISC-V core and co-processors.
- Controls the accelerator using custom opcodes



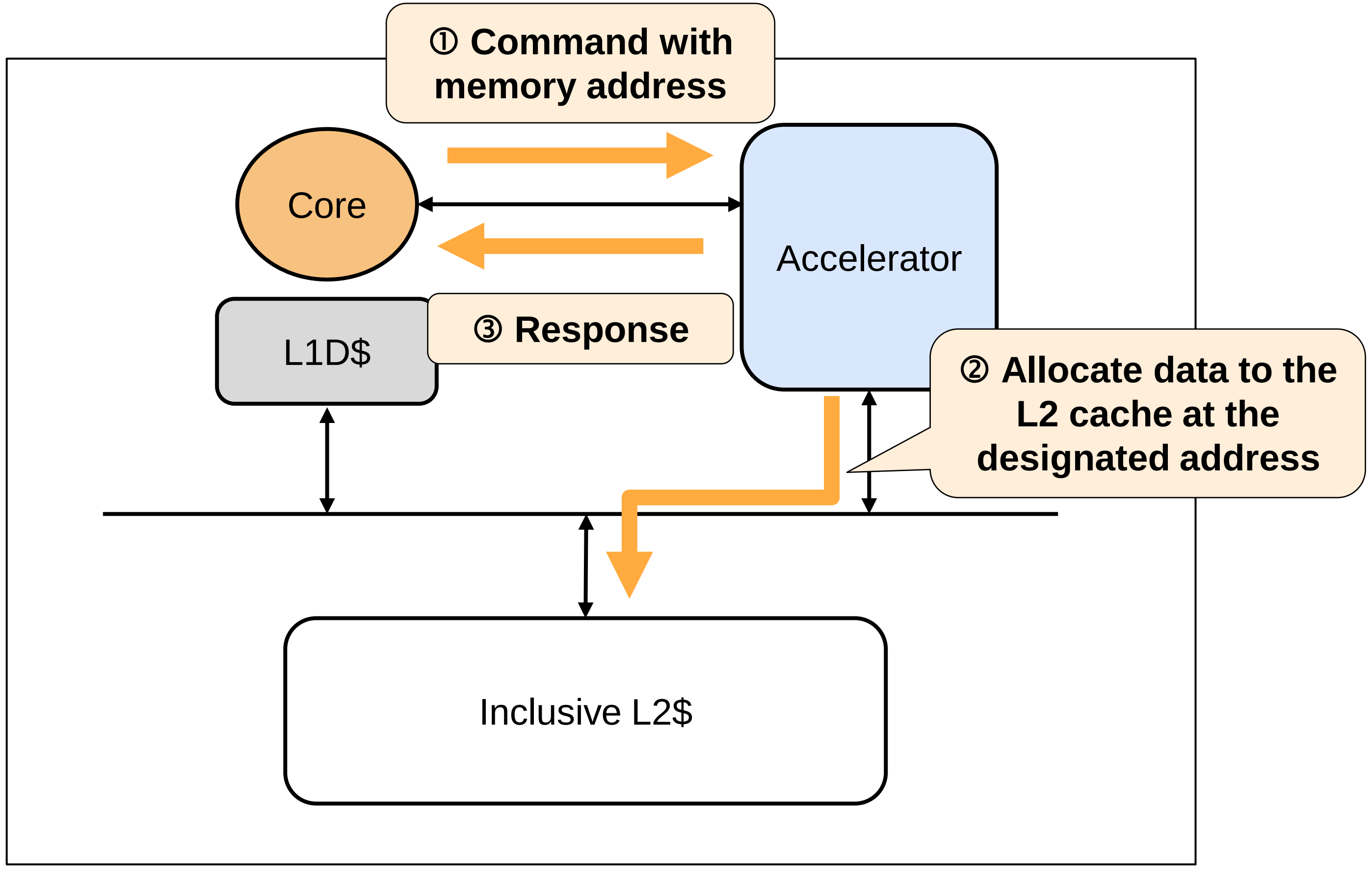
RoCC Interface

- RoCC commands are easily customized.
- An attacker can easily take control of the accelerator by using the opcodes assigned to RoCC interface

```
#define ROCC_INSTRUCTION_0_R_R (opcode, rs1, rs2, func7){
    asm volatile(
    ".insn r " STR(CAT(CUSTOM_, opcode)) ", "
    STR(0x3) ", " STR(func7) ", x0, %0, %1”
    :
    : "r"(rs1), "r"(rs2));
}
```

RoCC macro code

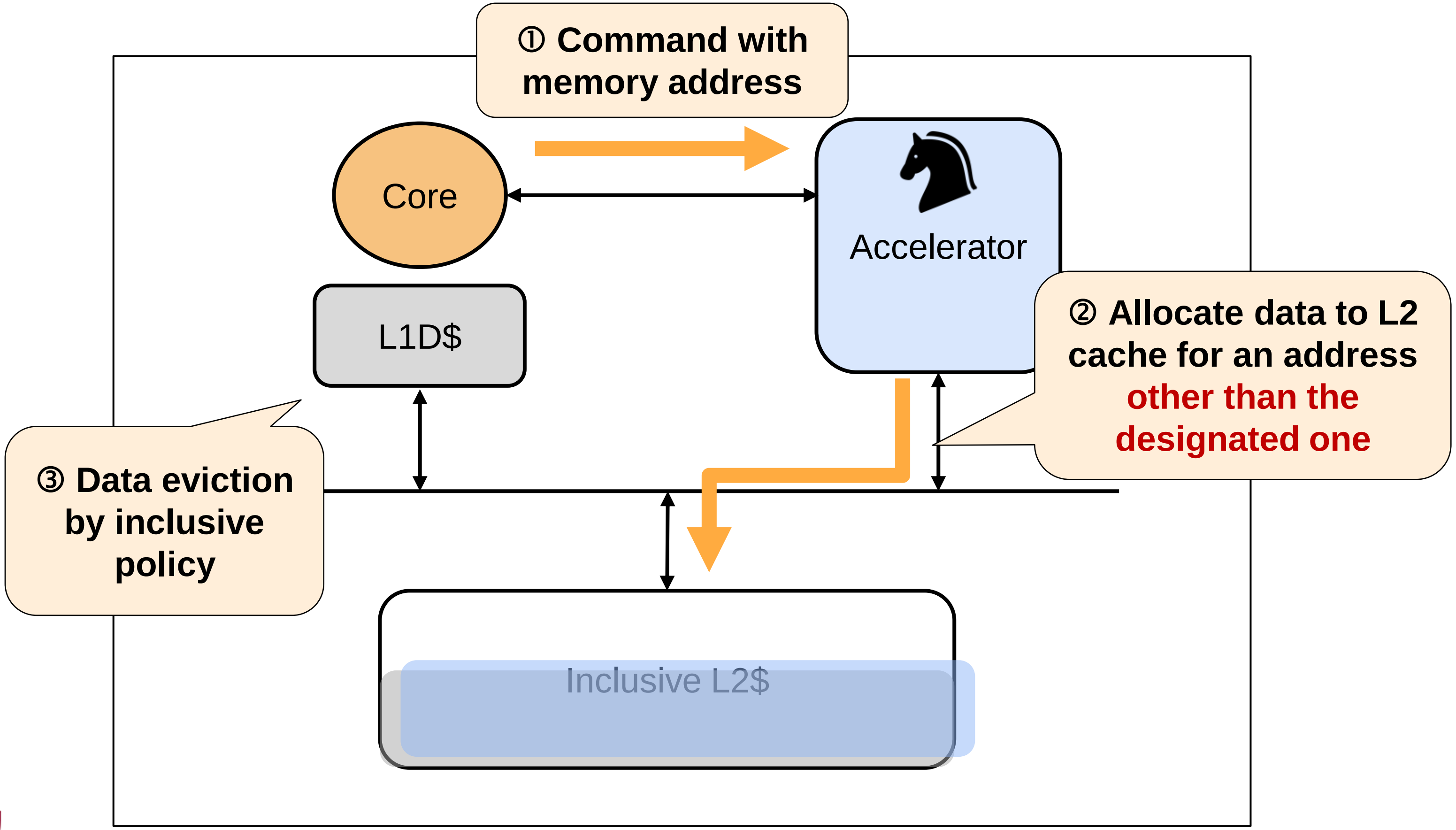
Basic Command Processing on RISC-V SoC



① Command with memory address
Core
Accelerator
L1D$
③ Response
② Allocate data to the L2 cache at the designated address
Inclusive L2$

고려대학교 KOREA UNIVERSITY

Cache Side-Channel Attacks on RISC-V SoC


① Command with memory address
Core
Accelerator
L1D$
② Allocate data to L2 cache for an address other than the designated one
③ Data eviction by inclusive policy
Inclusive L2$

Configuration

Chipyard infrastructure

Core	RISC-V Rocket core (In-order core architecture)
RISC-V ISA	RV64GC
L1 cache	16KB (4-way, 64B cache line)
L2 cache	512KB (8-way, 64B cache line)
Accelerator	Gemmini with hardware Trojan
Scratchpad capacity	256 KB

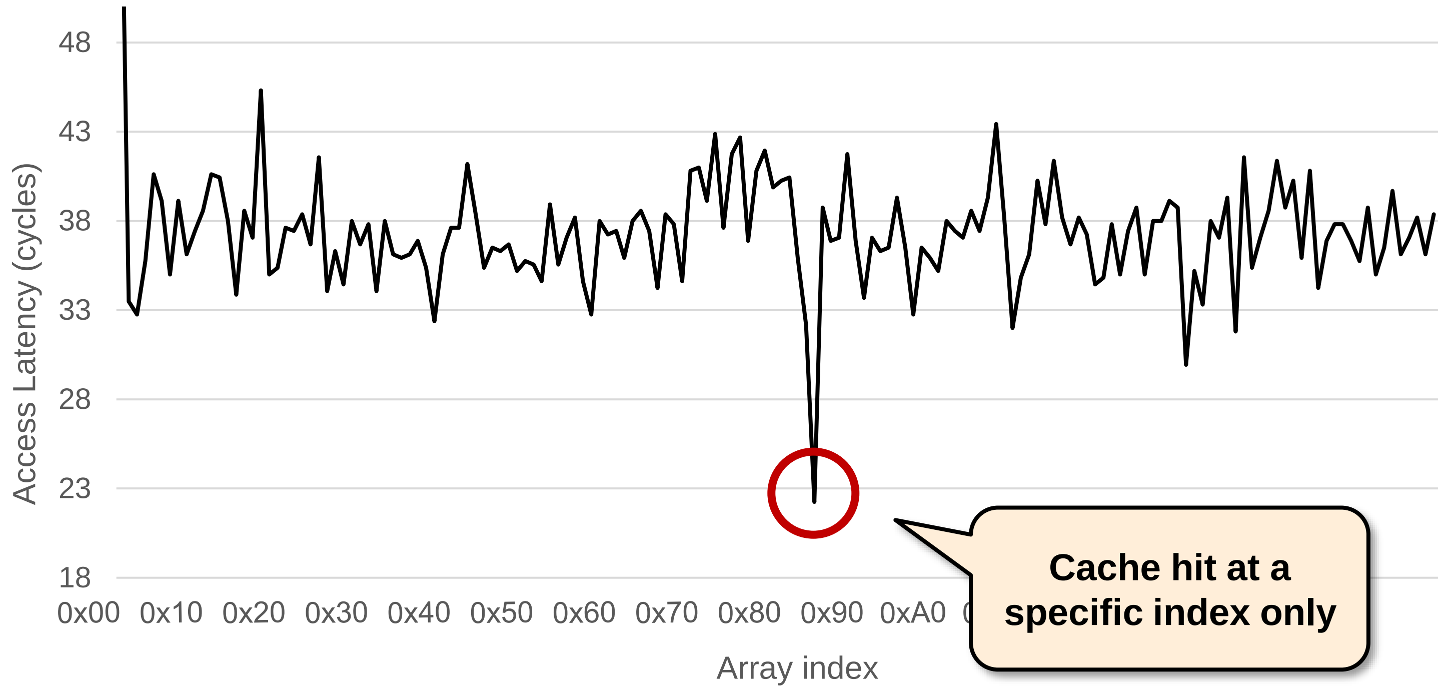
Memory Access Timing Measurement

```
For i from 0 to L1SIZE-1
    addr=&array[i*LINE];
    time1=rdcycle();
    junk = *addr;
    time2=rdcycle()-time1;
End
```

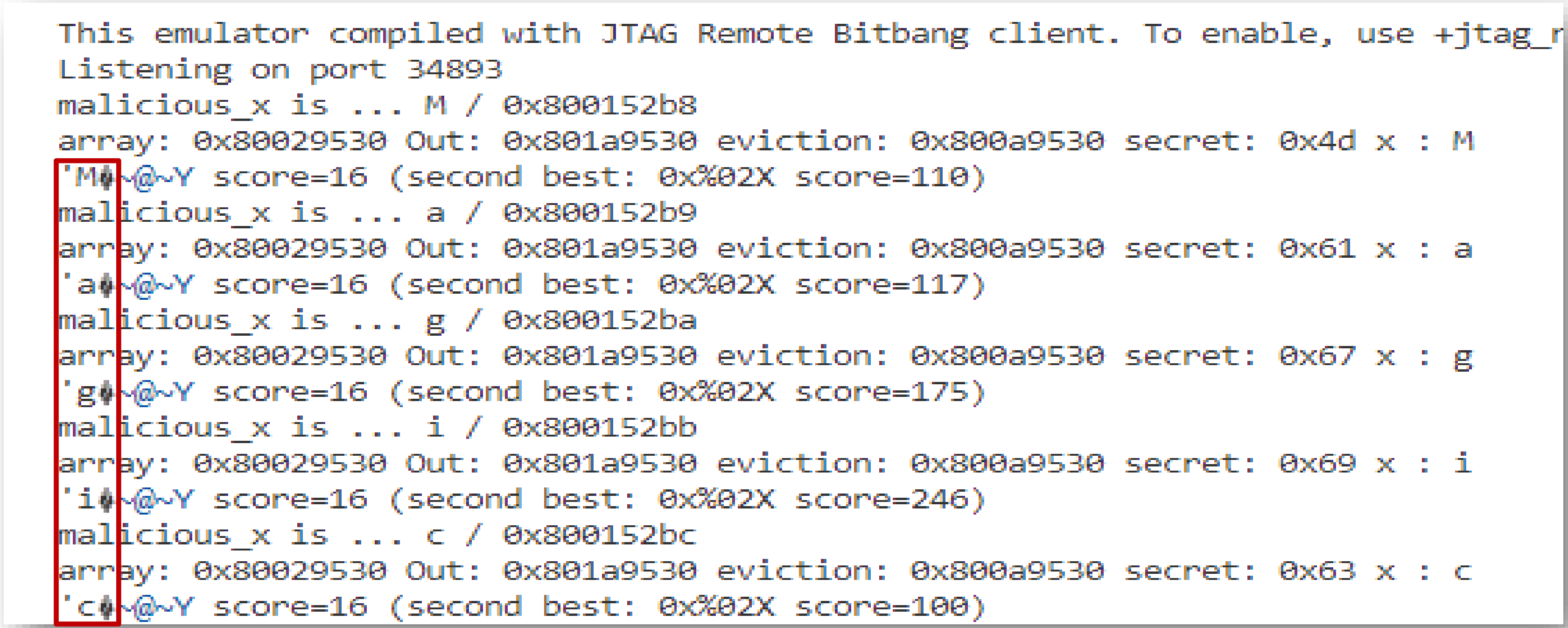
RISC-V CSR instruction

csrrs rd, cycle, x0

Attack Result



Attack Result



```
This emulator compiled with JTAG Remote Bitbang client. To enable, use +jtag_r
Listening on port 34893
malicious_x is ... M / 0x800152b8
array: 0x80029530 Out: 0x801a9530 eviction: 0x800a9530 secret: 0x4d x : M
'M�~@~Y score=16 (second best: 0x%02X score=110)
malicious_x is ... a / 0x800152b9
array: 0x80029530 Out: 0x801a9530 eviction: 0x800a9530 secret: 0x61 x : a
'a�~@~Y score=16 (second best: 0x%02X score=117)
malicious_x is ... g / 0x800152ba
array: 0x80029530 Out: 0x801a9530 eviction: 0x800a9530 secret: 0x67 x : g
'g�~@~Y score=16 (second best: 0x%02X score=175)
malicious_x is ... i / 0x800152bb
array: 0x80029530 Out: 0x801a9530 eviction: 0x800a9530 secret: 0x69 x : i
'i�~@~Y score=16 (second best: 0x%02X score=246)
malicious_x is ... c / 0x800152bc
array: 0x80029530 Out: 0x801a9530 eviction: 0x800a9530 secret: 0x63 x : c
'c�~@~Y score=16 (second best: 0x%02X score=100)
```

고려대학교
KOREA UNIVERSITY

Conclusion

Flush+Reload attack on RISC-V SoC Platform

- System-on-chips (SoCs) often use third-party IPs, posing a serious security threats by hardware Trojans.
- Attackers can exploit RoCC interface to perform cache side-channel attacks more efficiently.
- Architectural solutions are required to defend against the RoCC-based attacks

Thank you

Backup