JackHammer: Efficient Rowhammer on Heterogeneous FPGA-CPU Platforms

Zane Weissman [1] **Thore Tiemann** [2] Daniel Moghimi [1] Evan Custodio [3] Thomas Eisenbarth [2] Berk Sunar [1]

[1]Worcester Polytechnic Institute, MA, USA

[2]University of Lübeck, Lübeck, Germany

[3]Intel Corporation, Hudson, MA, USA

CHES 2020, Sep. 14 – 18 2020

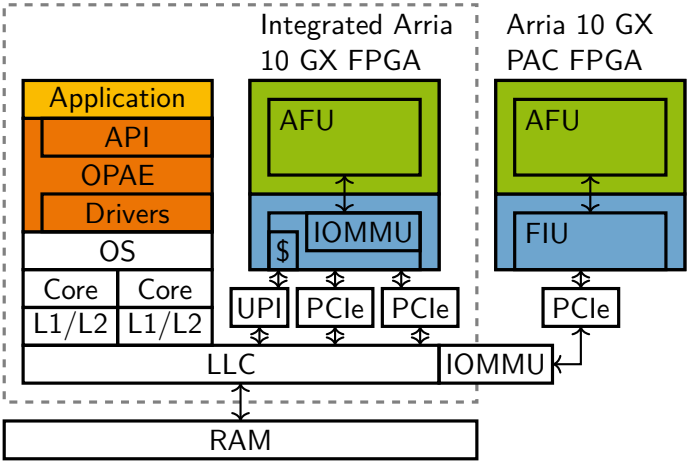
Motivation

Microsoft
Azure

Motivation

Microsoft
Azure

Motivation


Integrated Arria
10 GX FPGA
Arria 10 GX
PAC FPGA
Application
API
OPAE
Drivers
OS
Core
Core
L1/L2
L1/L2
AFU
$
IOMMU
AFU
FIU
UPI
PCIe
PCIe
PCIe
LLC
IOMMU
RAM

Motivation

Important Considerations

- Address spaces: physical, virtual, I/O virtual

Motivation

Important Considerations

- Address spaces: physical, virtual, I/O virtual
- Pages (4 KB) and hugepages (2 MB)

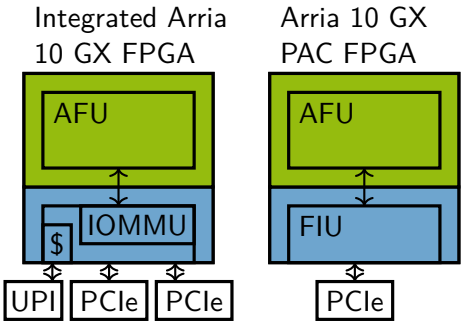
Motivation

Important Considerations

- Address spaces: physical, virtual, I/O virtual
- Pages (4 KB) and hugepages (2 MB)
- Which caches are/aren't modified by CPU/FPGA reads/writes/flushes

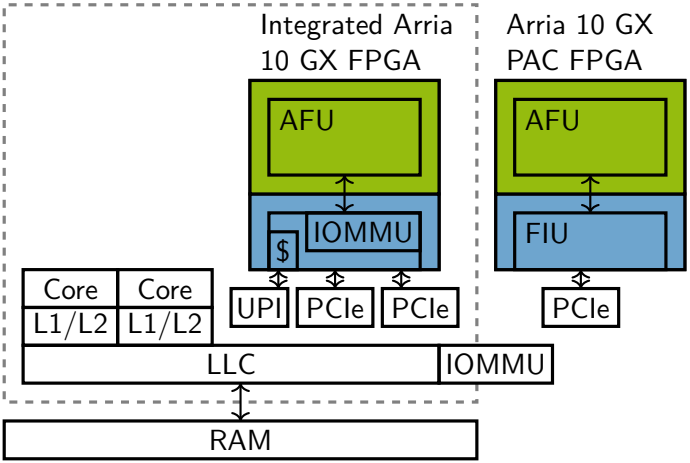
Background

Intel Acceleration Stack



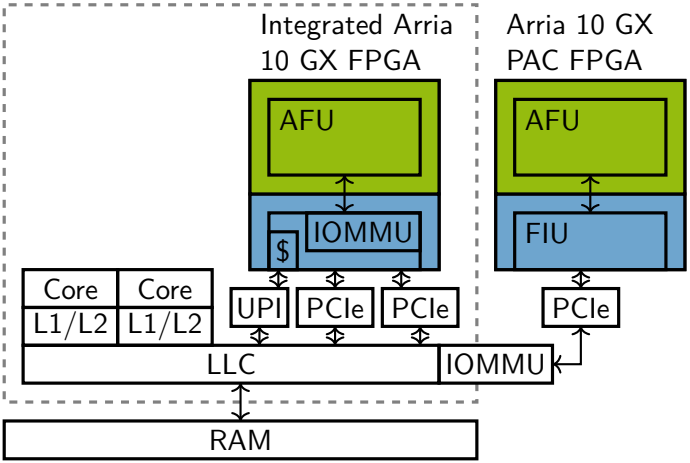
Background

Intel Acceleration Stack



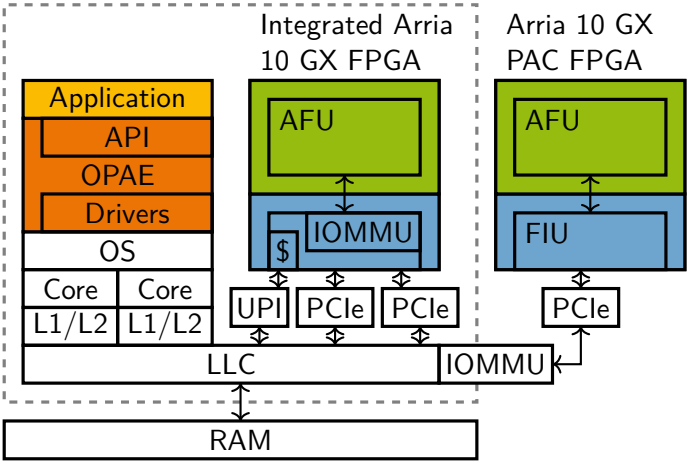
Background

Intel Acceleration Stack



Background

Intel Acceleration Stack



Background

Core Cache Interface Port

- MMIO
  - Device Feature Header
- DMA
  - Communication channels

Background

Core Cache Interface Port

- MMIO
  - Device Feature Header
- DMA
  - Communication channels
  - Physical addressing of (huge)pages

Background

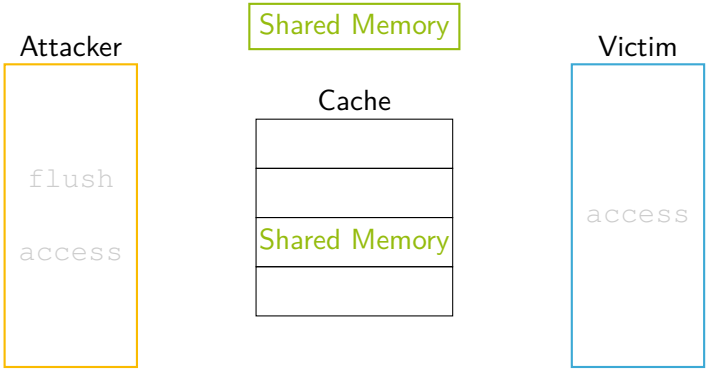
Core Cache Interface Port

- MMIO
  - Device Feature Header
- DMA
  - Communication channels
  - Physical addressing of (huge)pages
  - Caching hints

```
RdLine_I  WrLine_I
RdLine_S  WrLine_M
          WrPush_I
```

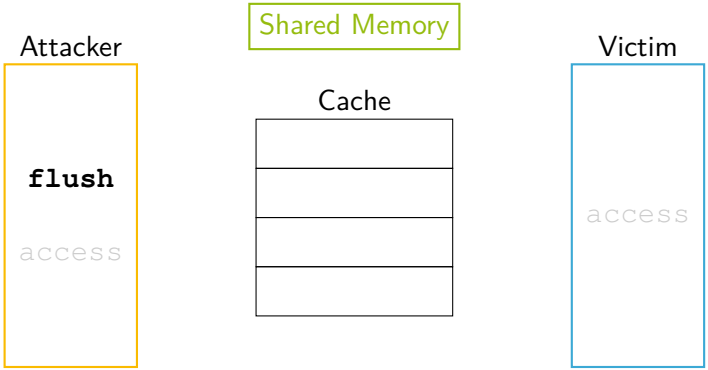
Cache Attacks

Background – Flush+Reload



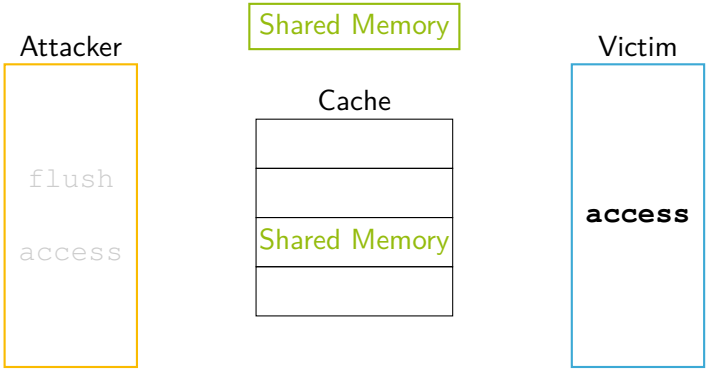
Cache Attacks

Background – Flush+Reload



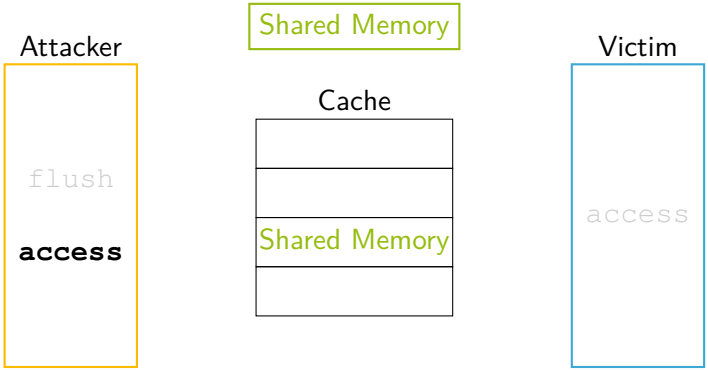
Cache Attacks

Background – Flush+Reload



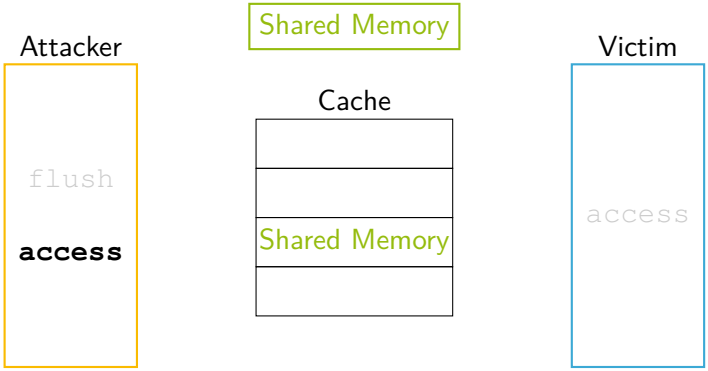
Cache Attacks

Background – Flush+Reload



Cache Attacks

Background – Flush+Reload

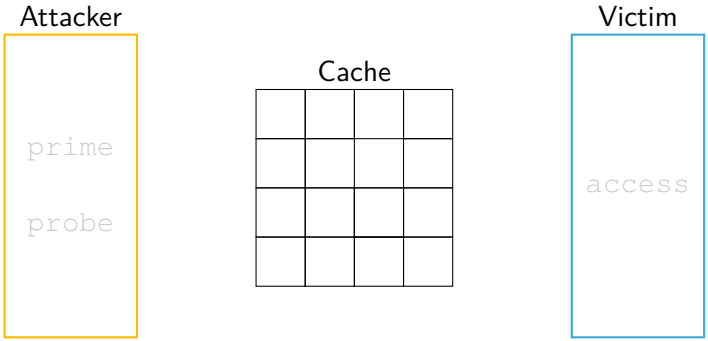


fast
Victim accessed

slow
Victim did not access

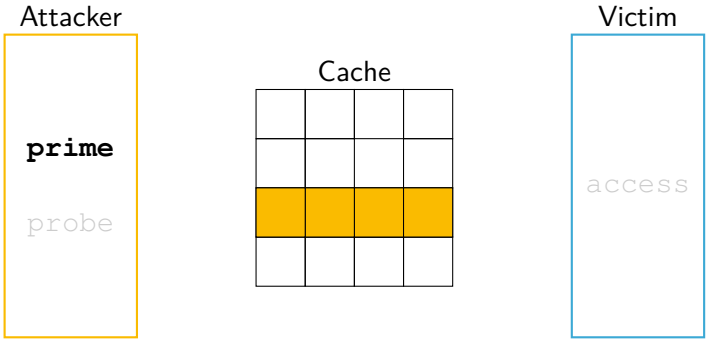
Cache Attacks

Background – Prime+Probe



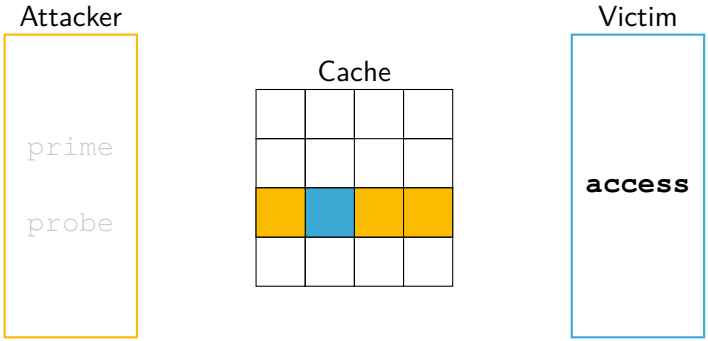
Cache Attacks

Background – Prime+Probe



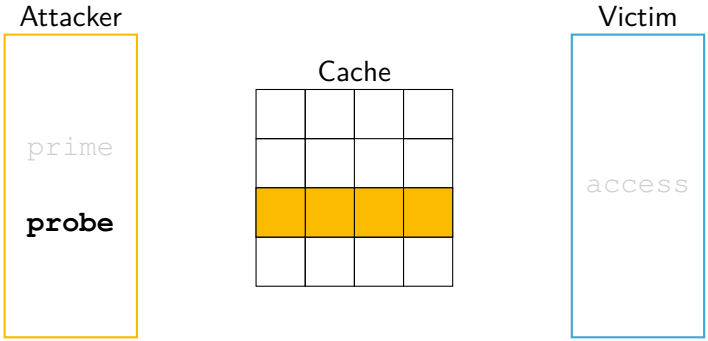
Cache Attacks

Background – Prime+Probe



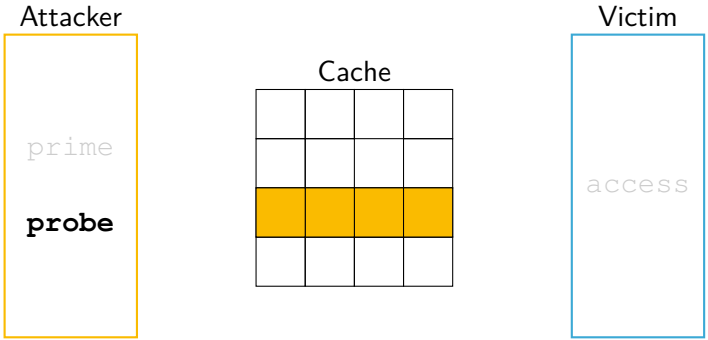
Cache Attacks

Background – Prime+Probe



Cache Attacks

Background – Prime+Probe

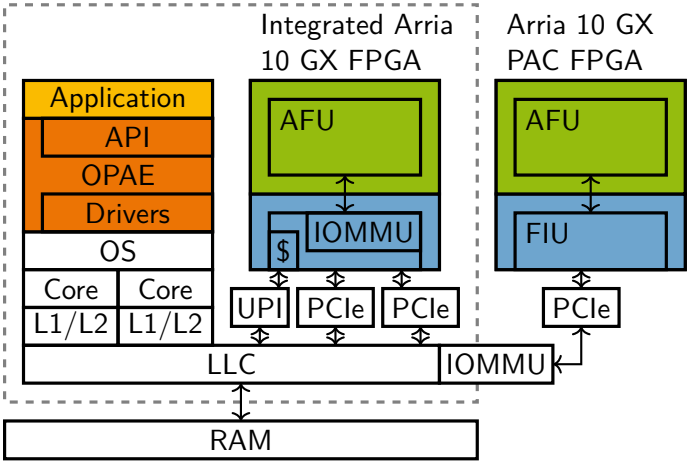


fast
Victim did not accessed

slow
Victim accessed

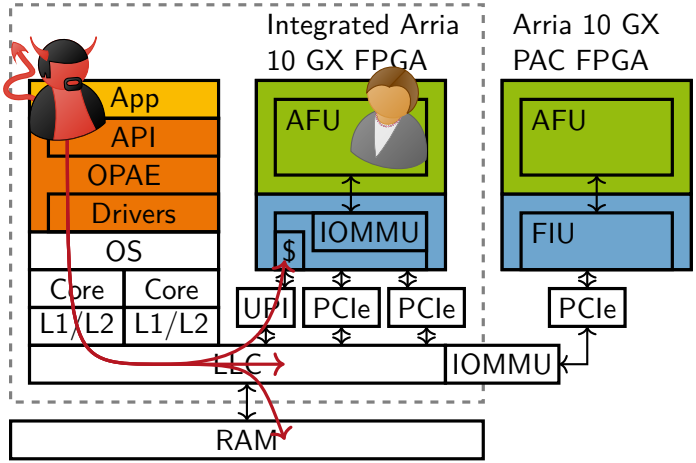
Cache Attacks

Attack Vectors – CPU



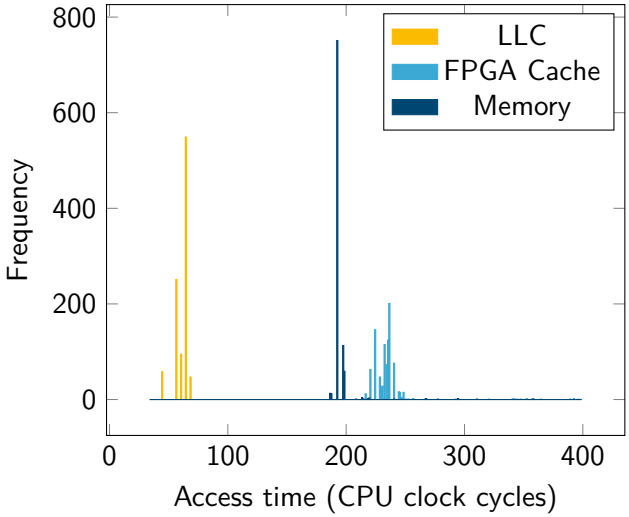
Cache Attacks

Attack Vectors – CPU



Cache Attacks

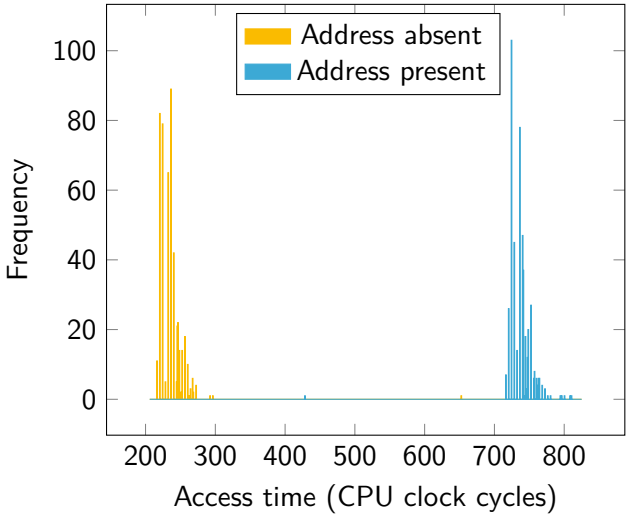
Attack Vectors – CPU



- Flush instruction
- Reload timing
  $\Rightarrow$ **Flush+Reload**

Cache Attacks

Attack Vectors – CPU

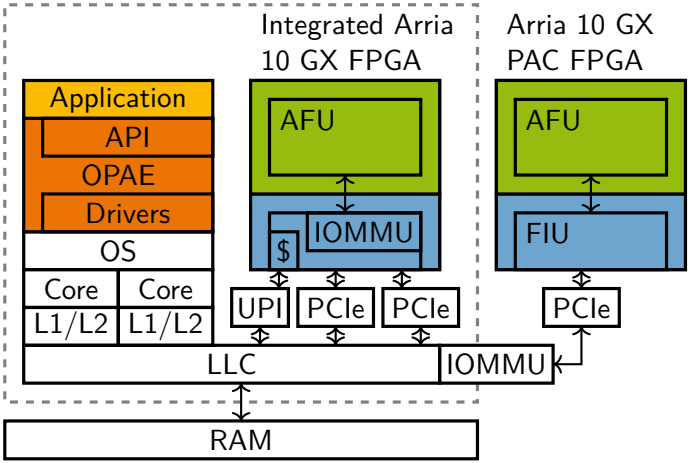


- Flush instruction
- Reload timing
  $\Rightarrow$ **Flush+Reload**

- Flush instruction
- Flush timing
  $\Rightarrow$ **Flush+Flush**

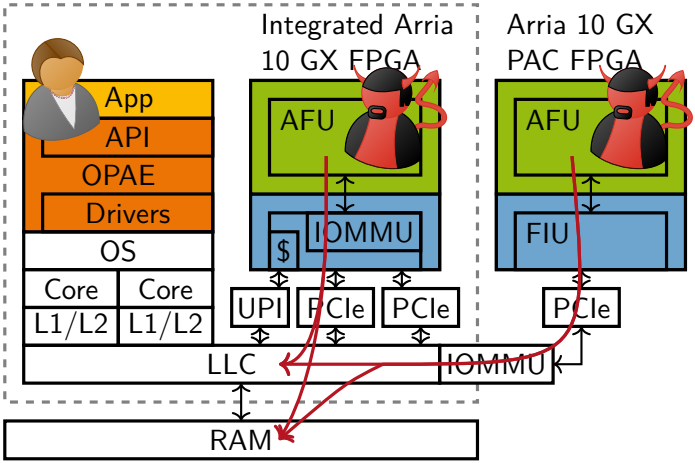
Cache Attacks

Attack Vectors – FPGA/PCIe



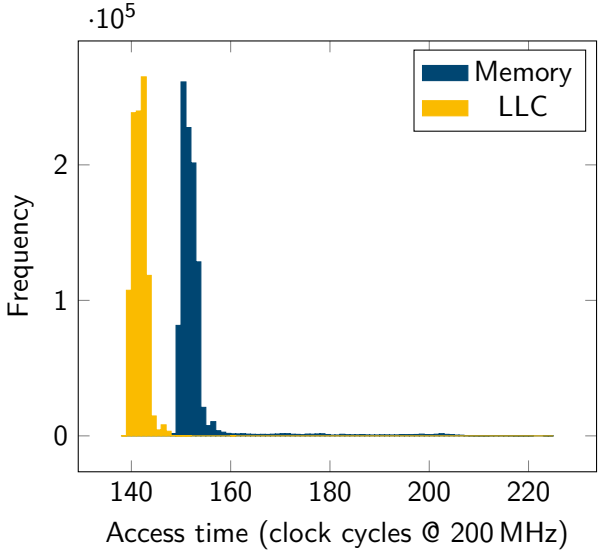
Cache Attacks

Attack Vectors – FPGA/PCIe



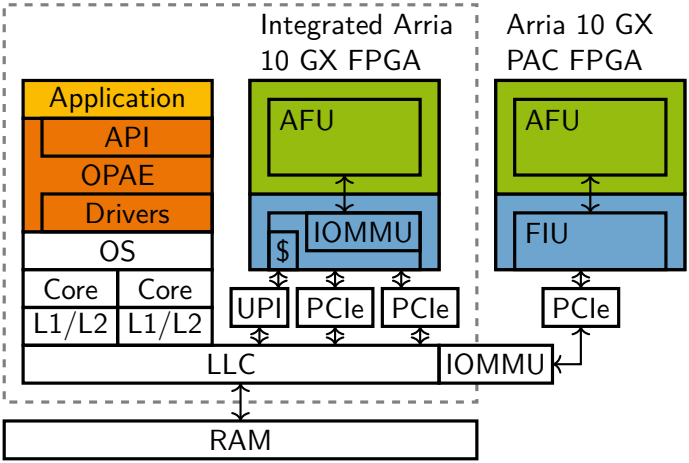
Cache Attacks

Attack Vectors – FPGA/PCIe



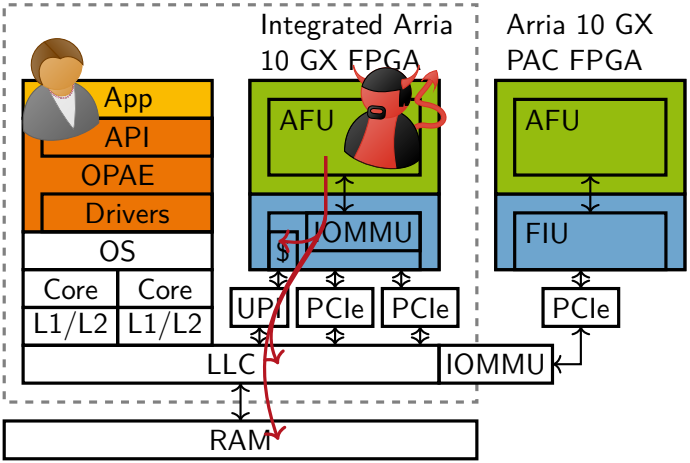
Cache Attacks

Attack Vectors – FPGA/UPI



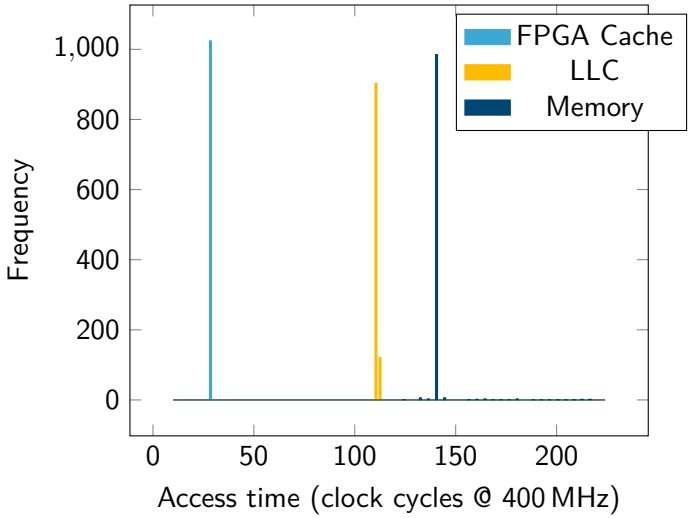
Cache Attacks

Attack Vectors – FPGA/UPI



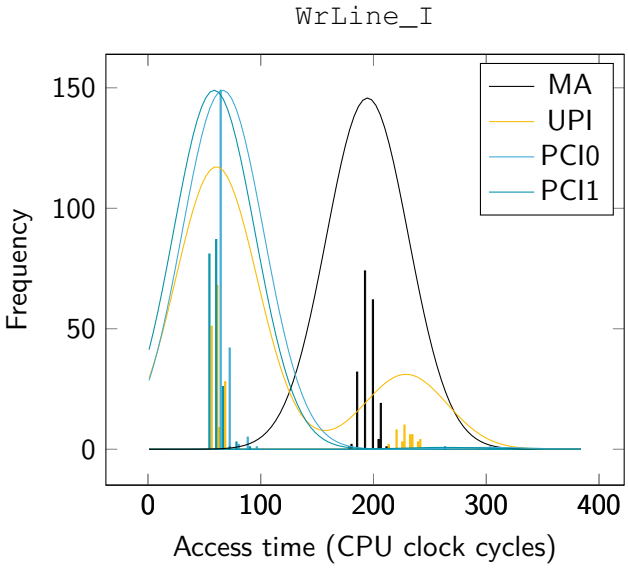
Cache Attacks

Attack Vectors – FPGA/UPI



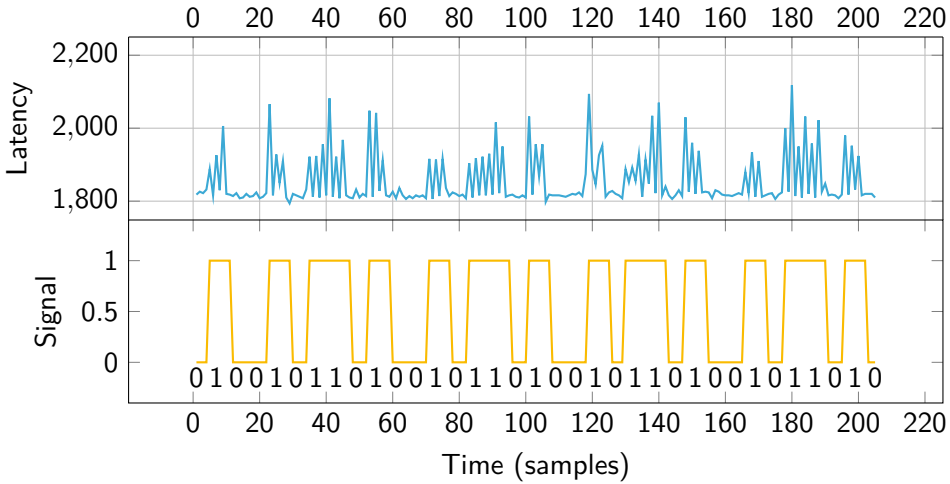
Cache Attacks

Attack Vectors – Caching Hints



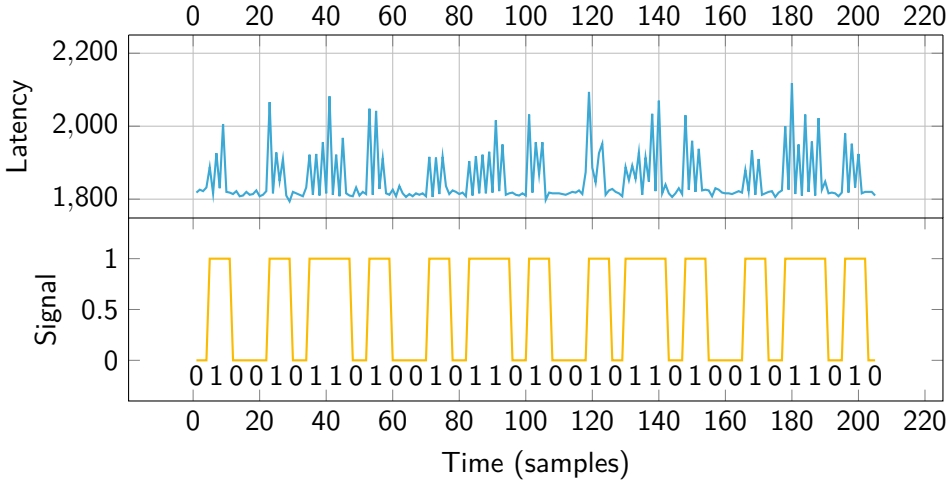
Cache Attacks

Covert Channel



Cache Attacks

Covert Channel



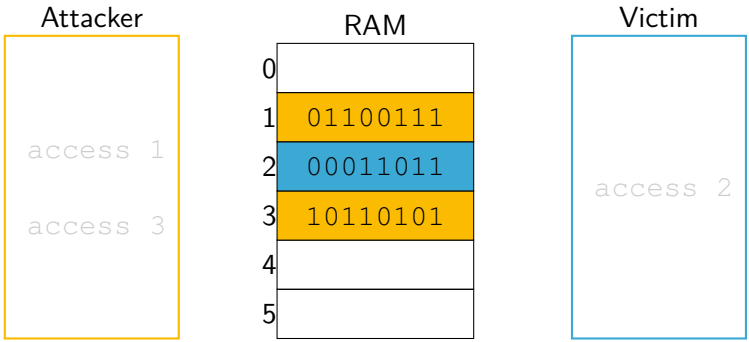
Throughput: 94.98 kBit/s

JackHammer

Background – Rowhammer

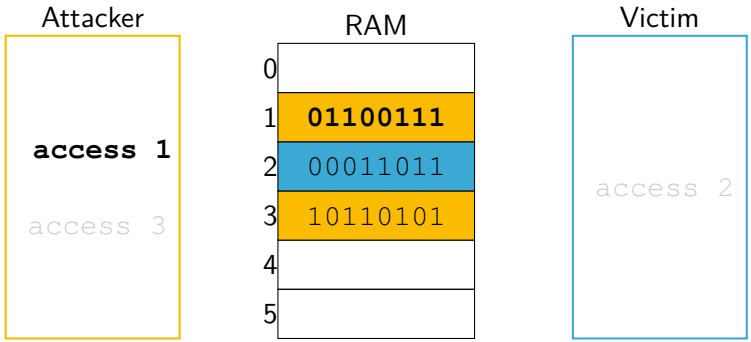
JackHammer

Background – Rowhammer



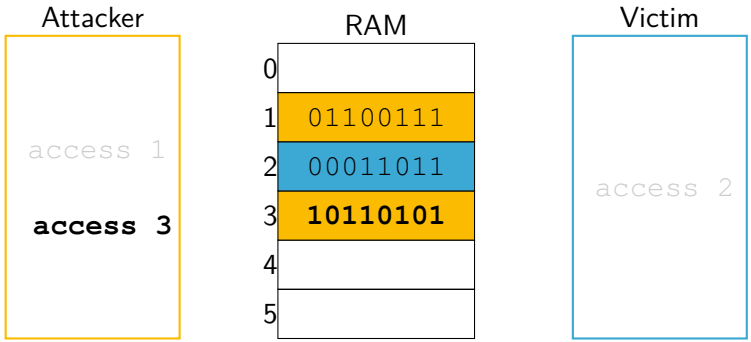
JackHammer

Background – Rowhammer



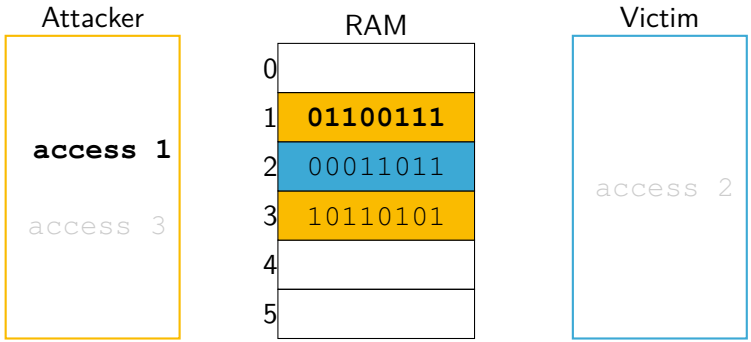
JackHammer

Background – Rowhammer



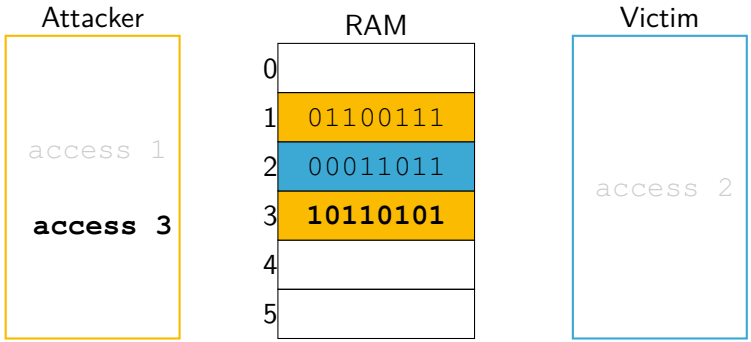
JackHammer

Background – Rowhammer



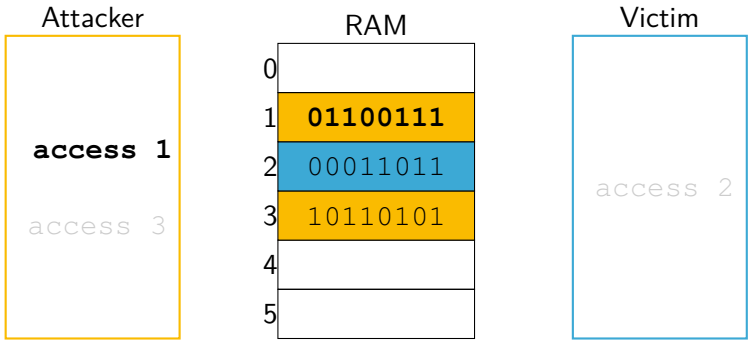
JackHammer

Background – Rowhammer



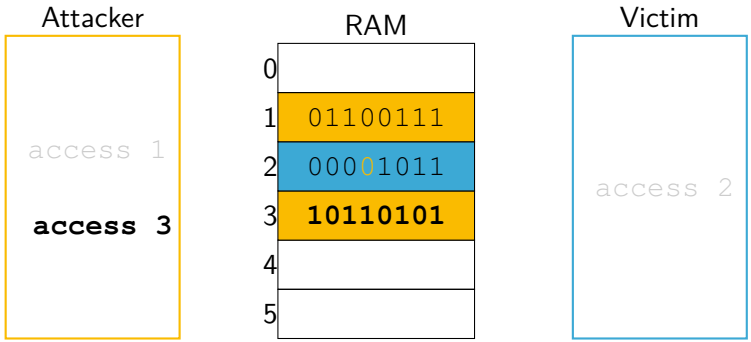
JackHammer

Background – Rowhammer



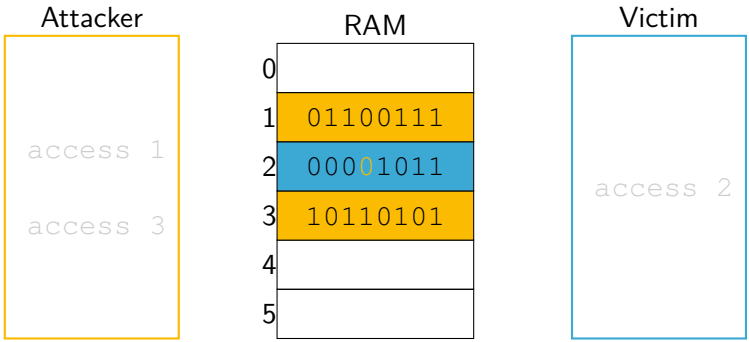
JackHammer

Background – Rowhammer



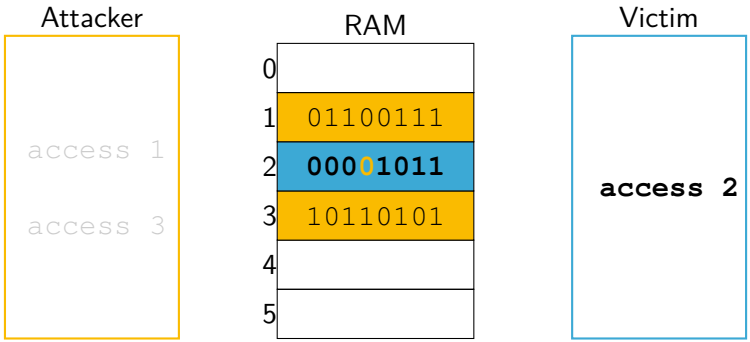
JackHammer

Background – Rowhammer



JackHammer

Background – Rowhammer



JackHammer

Background – Rowhammer

JackHammer

Background – Rowhammer

- **Aggressor rows** accessed by the attacker must be near **victim rows**

JackHammer

Background – Rowhammer

- **Aggressor rows** accessed by the attacker must be near **victim rows**
- Rows mapped by XORing bits of the physical address on most modern CPUs (desktop, server, mobile) - see “DRAMA” by Pessl et al.

JackHammer

Background – Rowhammer

- **Aggressor rows** accessed by the attacker must be near **victim rows**
- Rows mapped by XORing bits of the physical address on most modern CPUs (desktop, server, mobile) - see “DRAMA” by Pessl et al.
- Attack probably relies on electromagnetic effects

JackHammer

Background – Rowhammer

- **Aggressor rows** accessed by the attacker must be near **victim rows**
- Rows mapped by XORing bits of the physical address on most modern CPUs (desktop, server, mobile) - see “DRAMA” by Pessl et al.
- Attack probably relies on electromagnetic effects
- Simplest defense: increase automatic DRAM row refresh rate

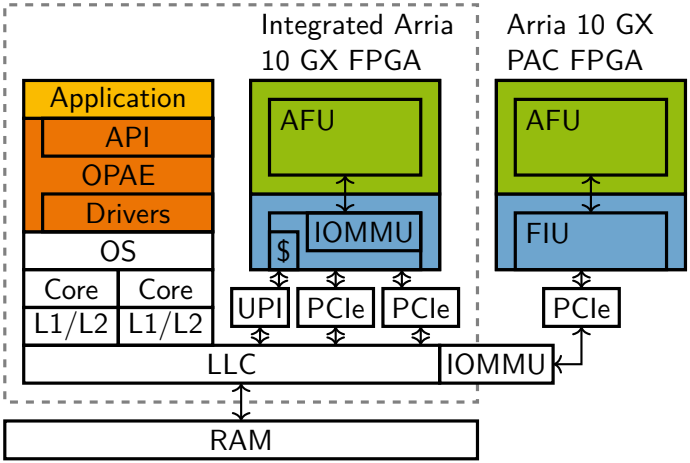
JackHammer

Background – Rowhammer

- **Aggressor rows** accessed by the attacker must be near **victim rows**
- Rows mapped by XORing bits of the physical address on most modern CPUs (desktop, server, mobile) - see “DRAMA” by Pessl et al.
- Attack probably relies on electromagnetic effects
- Simplest defense: increase automatic DRAM row refresh rate
- Shown to work on many DDR3, some DDR4, some ECC

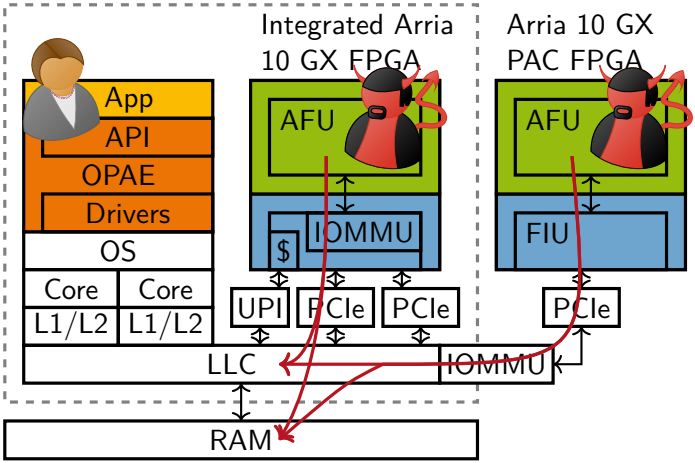
JackHammer

Background – Scenario



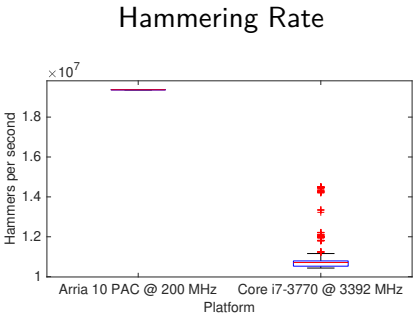
JackHammer

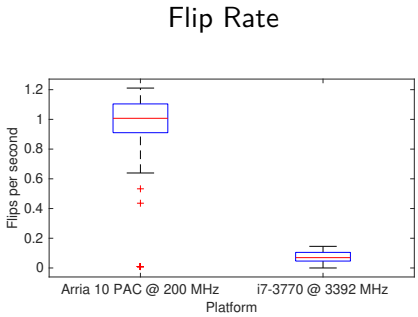
Background – Scenario



JackHammer

Performance

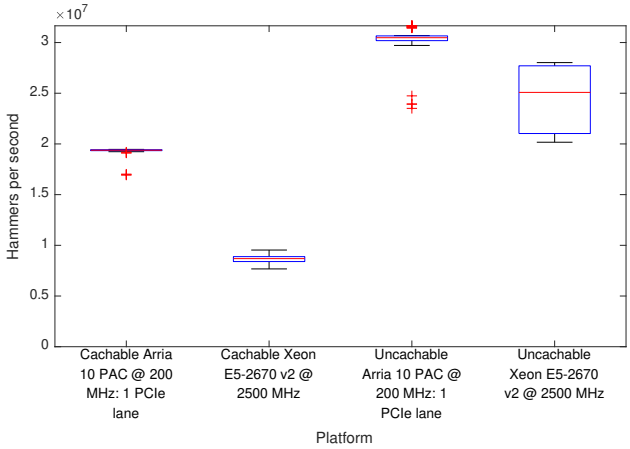




JackHammer

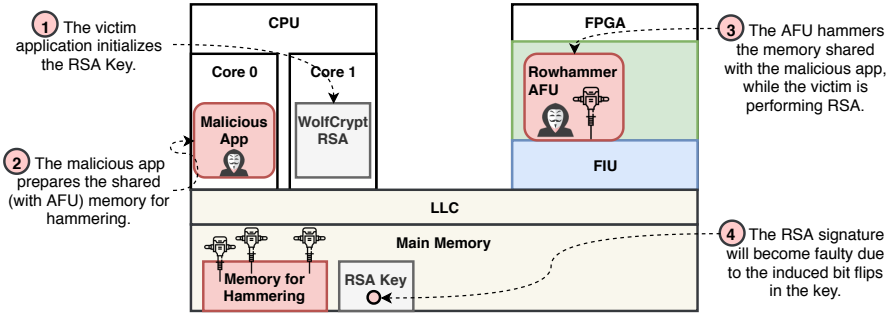
Caching and Rowhammer

Hammering rates with and without memory caching



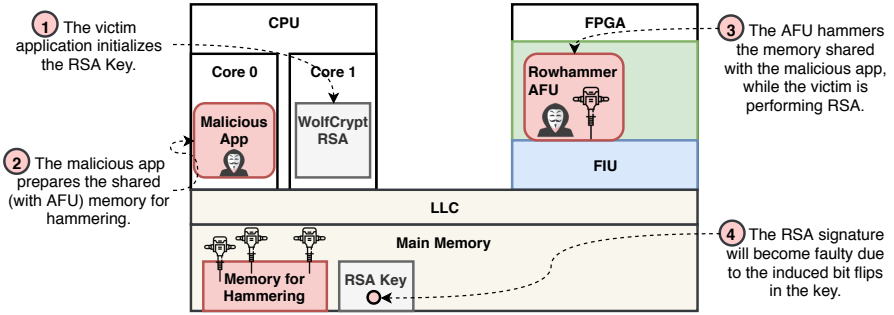
JackHammer

Fault Injection Attack (CVE-2019-19962)



JackHammer

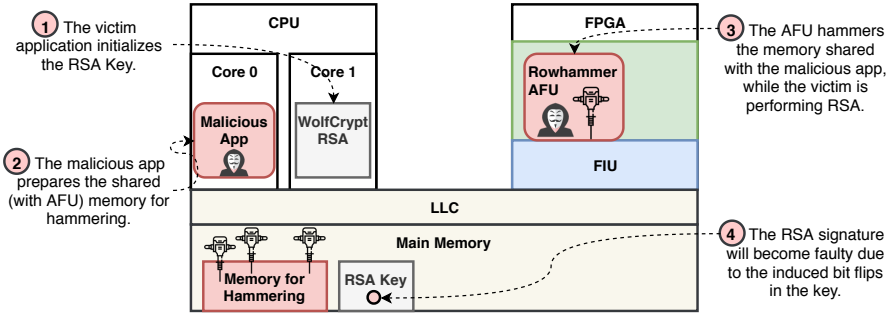
Fault Injection Attack (CVE-2019-19962)



- Best case: JackHammer causes a fault 25% faster than CPU Rowhammer

JackHammer

Fault Injection Attack (CVE-2019-19962)



- Best case: JackHammer causes a fault 25% faster than CPU Rowhammer
- With doubled DRAM row refresh rate: 185% faster than CPU

Conclusions

- Systematic verification of timing leakages

Conclusions

- Systematic verification of timing leakages
- Caching hint analysis

Conclusions

- Systematic verification of timing leakages
- Caching hint analysis
- Covert channel of 94.98 kBit/s

Conclusions

- Systematic verification of timing leakages
- Caching hint analysis
- Covert channel of 94.98 kBit/s
- Rowhammer performance acceleration by 25%

Conclusions

- Systematic verification of timing leakages
- Caching hint analysis
- Covert channel of 94.98 kBit/s
- Rowhammer performance acceleration by 25%
- CVE-2019-19962

Thanks for your attention!

✉ zweissman@wpi.edu t.tiemann@uni-luebeck.de

WPI

UNIVERSITÄT ZU LÜBECK
INSTITUTE FOR IT SECURITY

Background

FPGAs in the Cloud

- HAaaS vs. IaaS
- Major requirements[1]:
  - abstraction
  - sharing
  - compatibility
  - security
- Software security policies[2]

[1] Chen et al. '14

[2] Hategekimana et al. '18

Background

Physical Side-channels

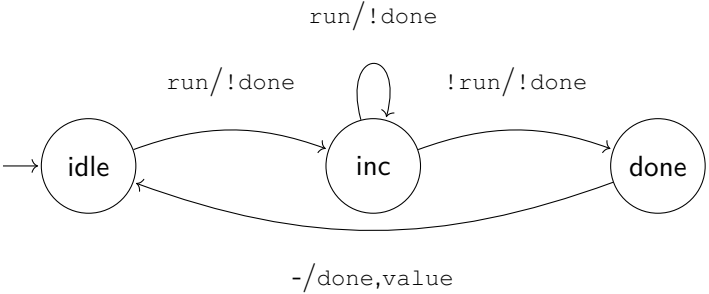
- Long wires CC and SC[3]
- Voltage fluctuation [4]
- Fault injection [5]
- SPA, DPA[6][7]

[3] Giechaskiel et al. '17 and '18
[4] Gnad et al. '17
[5] Krautter et al. '18
[6] Schellenberg et al. '18
[7] Zhao+Suh '18

Findings

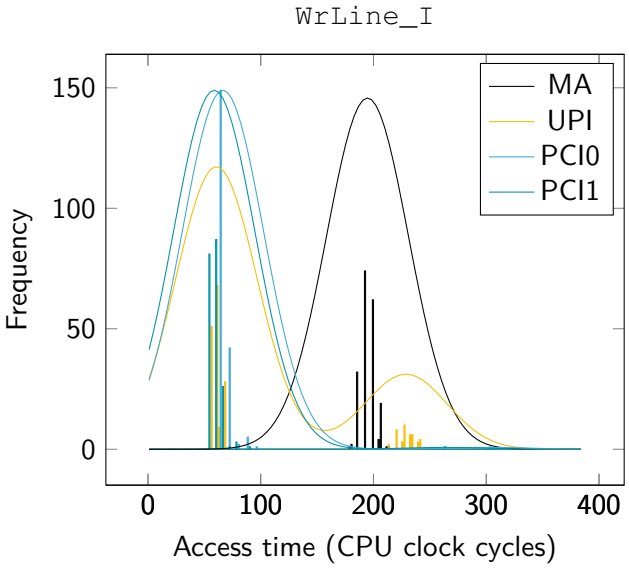
Hardware Timer

```
1 module hpc (
2         input               clk,
3         input               run,
4         output reg          done,
5         output reg [63:0] value
6 );
```



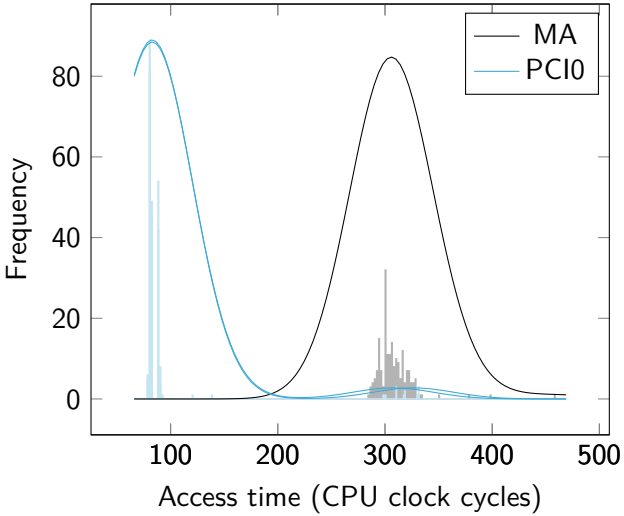
Findings

Write Caching Hints – Integrated Arria 10



Findings

Write Caching Hints – PAC



Future Work

- Attack PoC
- TLBleed
- CXL and CCIX
- Intra-FPGA cache attacks