EM Fault Model Characterization on SoCs

From different architectures to the same fault model

Thomas Trouchkine[1], Guillaume Bouffard[1,2], Jessy Clédière[3]

September 17, 2021

[1]National Cybersecurity Agency of France (ANSSI), Paris, France
[2]DIENS, École Normale Supérieure, CNRS, PSL University, Paris, France
[3]CEA, LETI, MINATEC Campus, Grenoble, France

Introduction - Handling sensitive operations

Sensitive operations

Introduction - Handling sensitive operations

Sensitive operations

Payment

Healthcare

Identification

Introduction - Handling sensitive operations

Sensitive operations

Payment

Healthcare

Identification

Historically

- handled by smartcards
- security designed devices
- high level security evaluation

Introduction - Handling sensitive operations

Sensitive operations

Payment Healthcare Identification

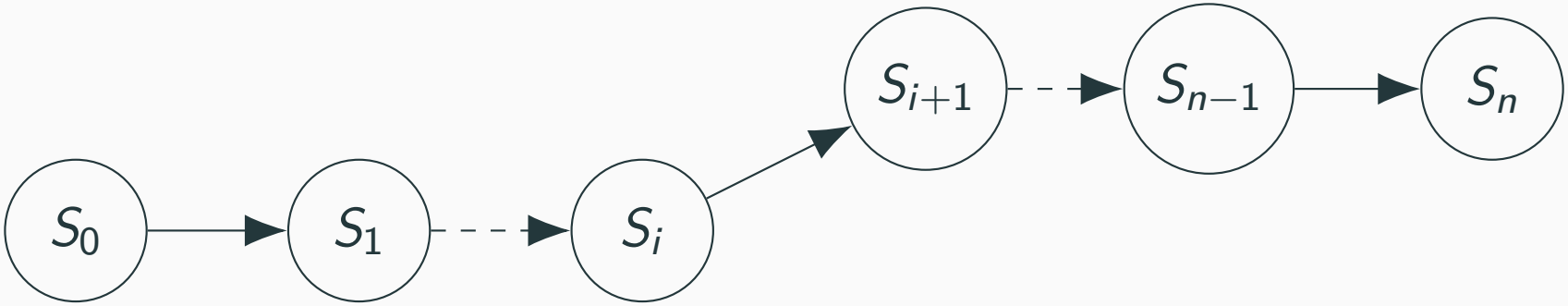
Historically

- handled by smartcards
- security designed devices
- high level security evaluation

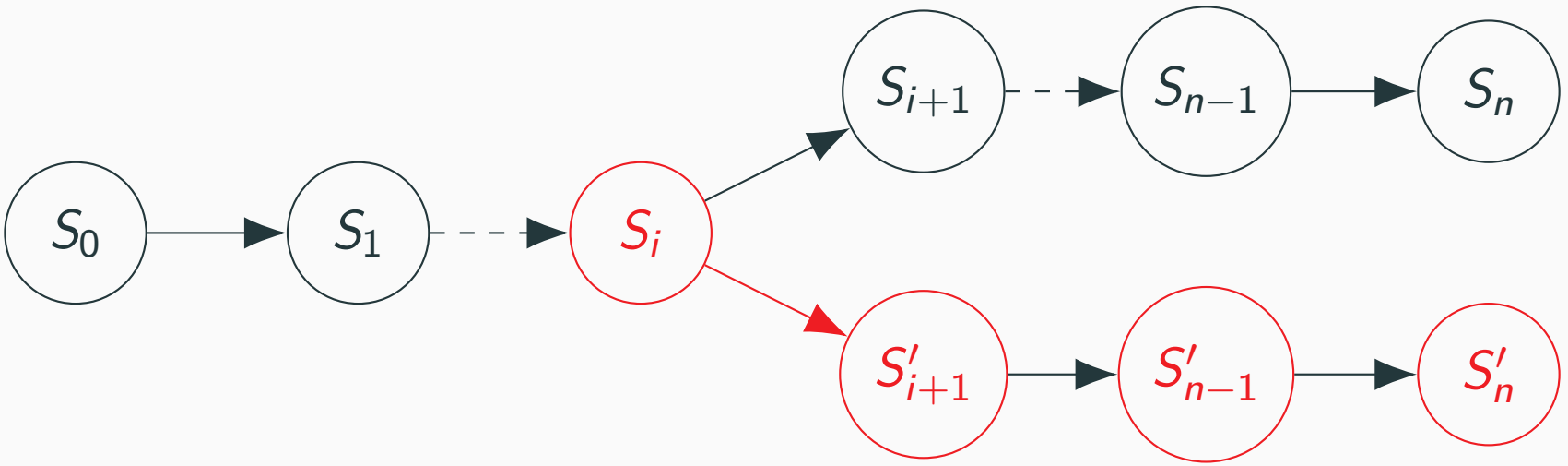
Nowadays

- handled by smartphones or laptops
- performance designed devices
- security added recently
- no security evaluation

Introduction - Perturbation attacks



Introduction - Perturbation attacks



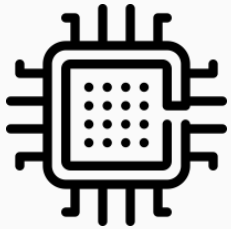
Introduction - Perturbation attacks

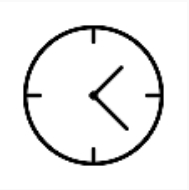
Electromagnetic waves

Temperature

Voltage

Light

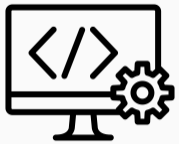




Clock

X-ray

Body biasing



Software

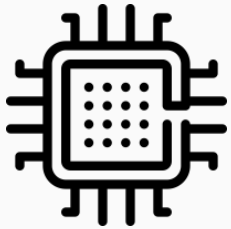
Introduction - Perturbation attacks

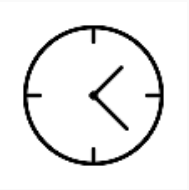
Electromagnetic waves [OGM15; DLM19]

Temperature

Voltage

Light

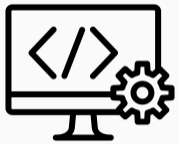




Clock

X-ray

Body biasing



Software

Characterization - Targets

BCM2837

(Raspberry Pi 3 B)

Intel Core i3-6100T

(Custom motherboard)

Case study - Characterization Method

Test program

```
orr r5, r5;
/*
 * Arbitrary number
 * of repetitions
 */
orr r5, r5;
```

Case study - Characterization Method

Test program

```
orr r5, r5;
/*
 * Arbitrary number
 * of repetitions
 */
orr r5, r5;
```

Initial values

Register	Initial values
r0	0xfffe0001
r1	0xfffd0002
r2	0xfffb0004
r3	0xfff70008
r4	0xffef0010

Case study - Characterization Method

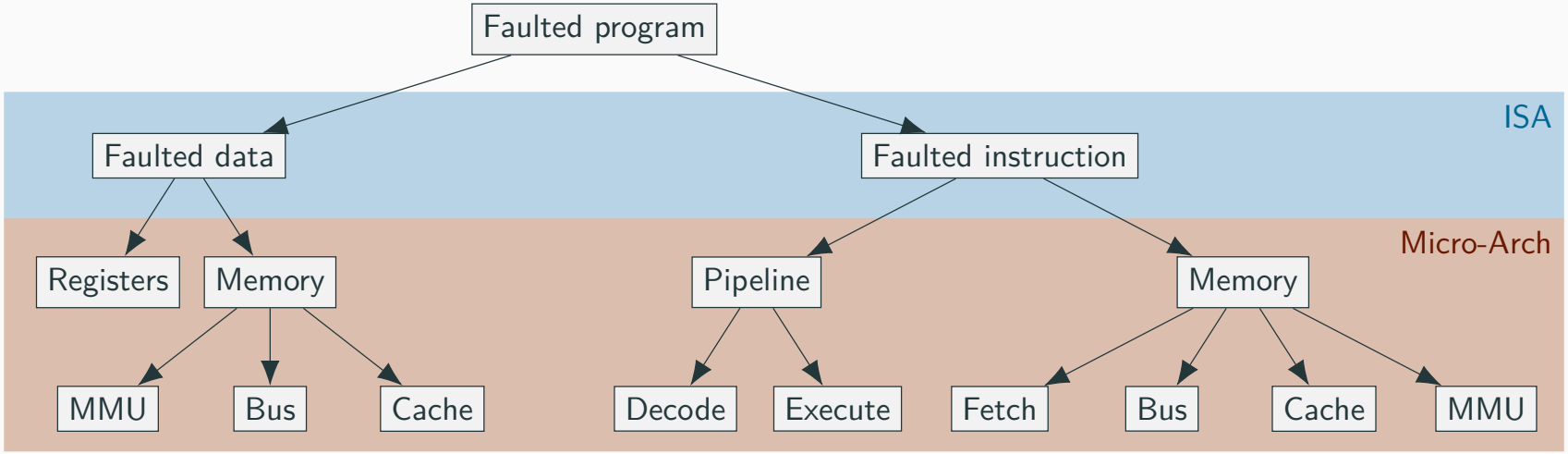
Test program

```
orr r5, r5;
/*
 * Arbitrary number
 * of repetitions
 */
orr r5, r5;
```

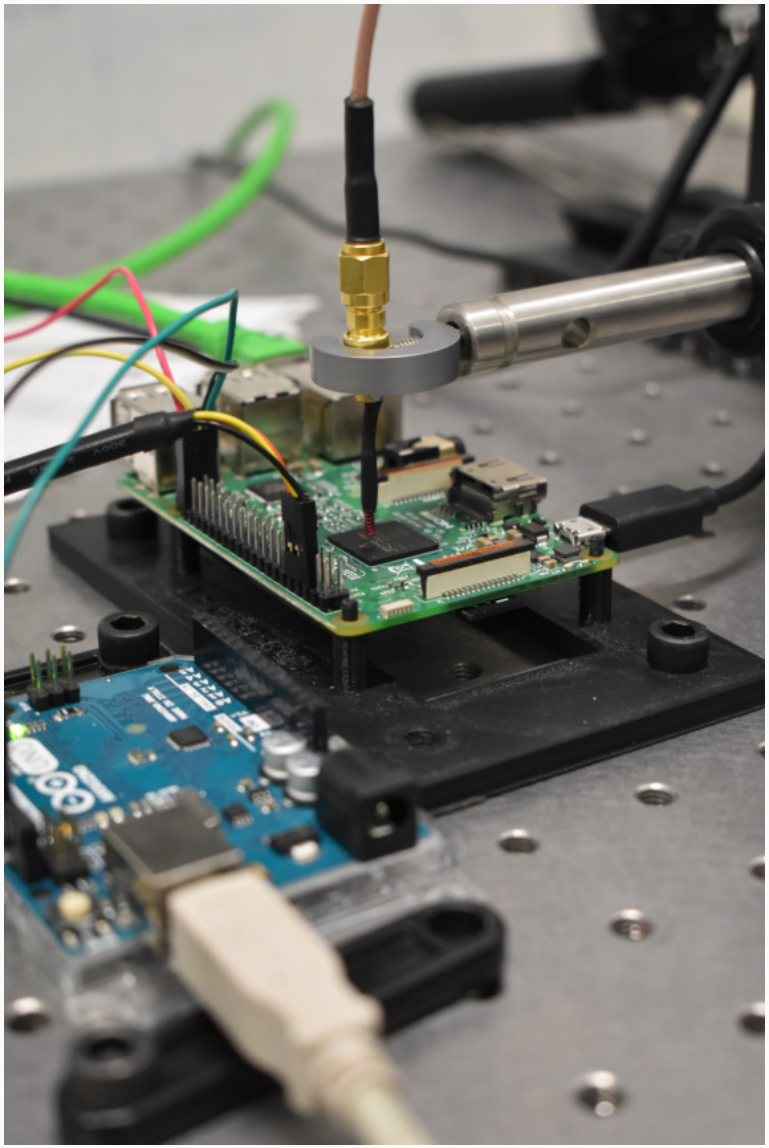
Initial values

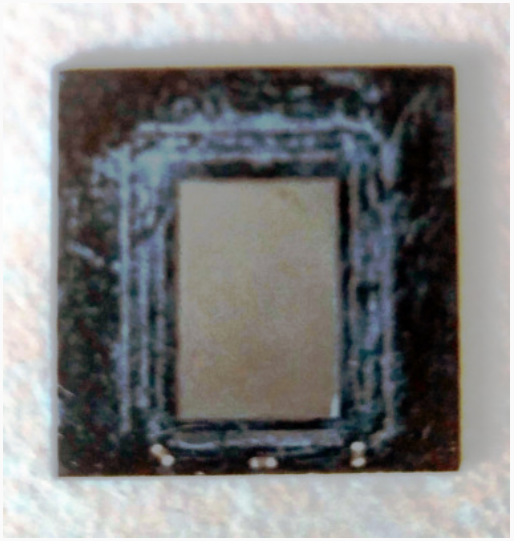
Register	Initial values
r0	0xfffe0001
r1	0xfffd0002
r2	0xfffb0004
r3	0xfff70008
r4	0xffef0010

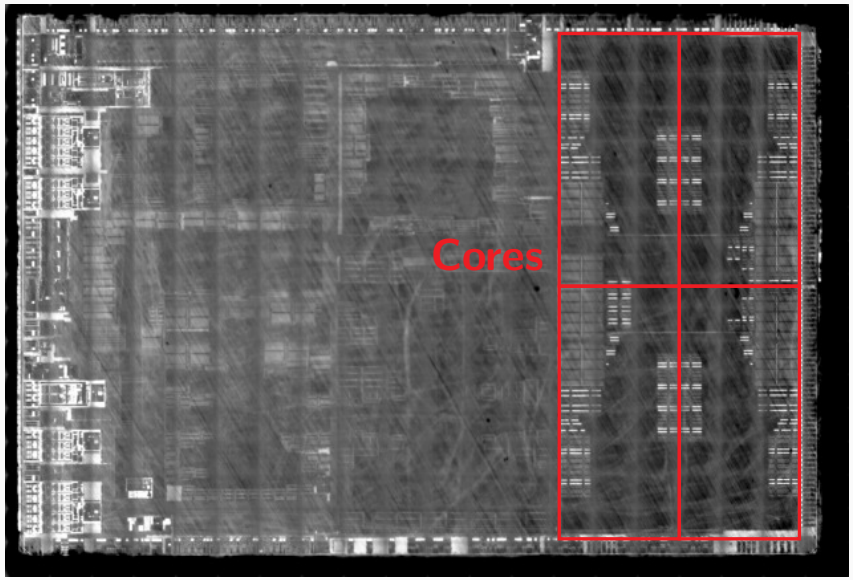
Top down approach



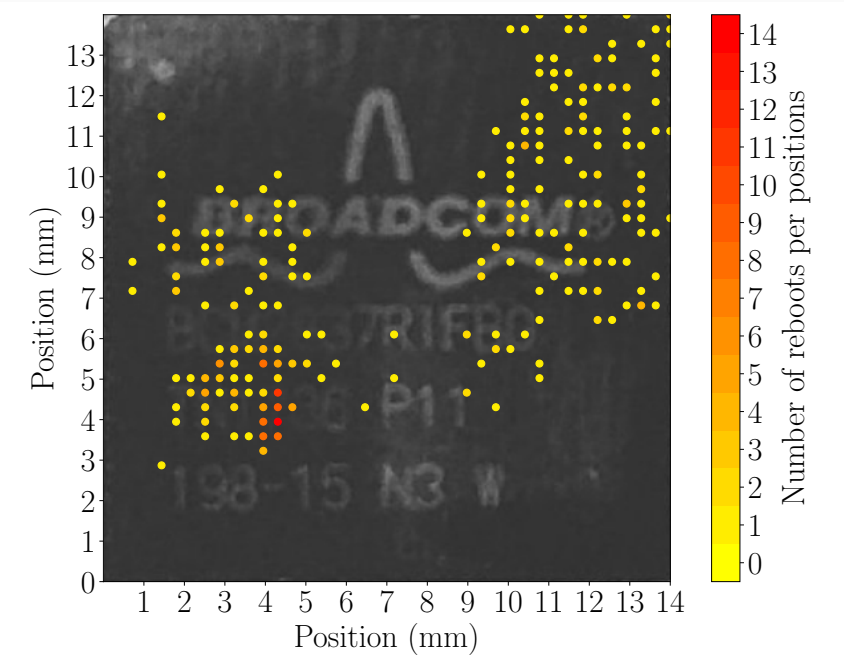
Characterization - BCM2837 (Raspberry Pi 3)



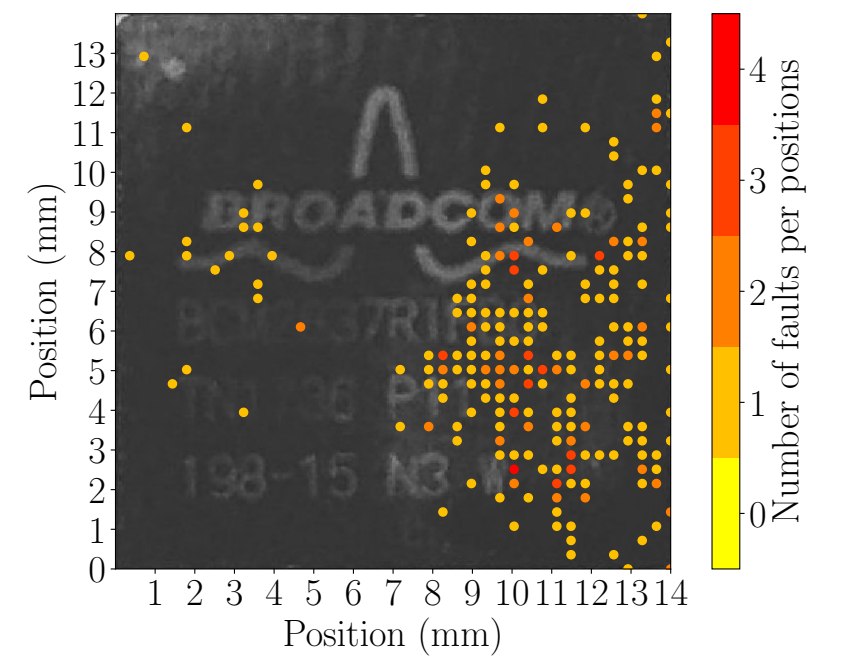




Characterization - BCM2837 (Raspberry Pi 3)



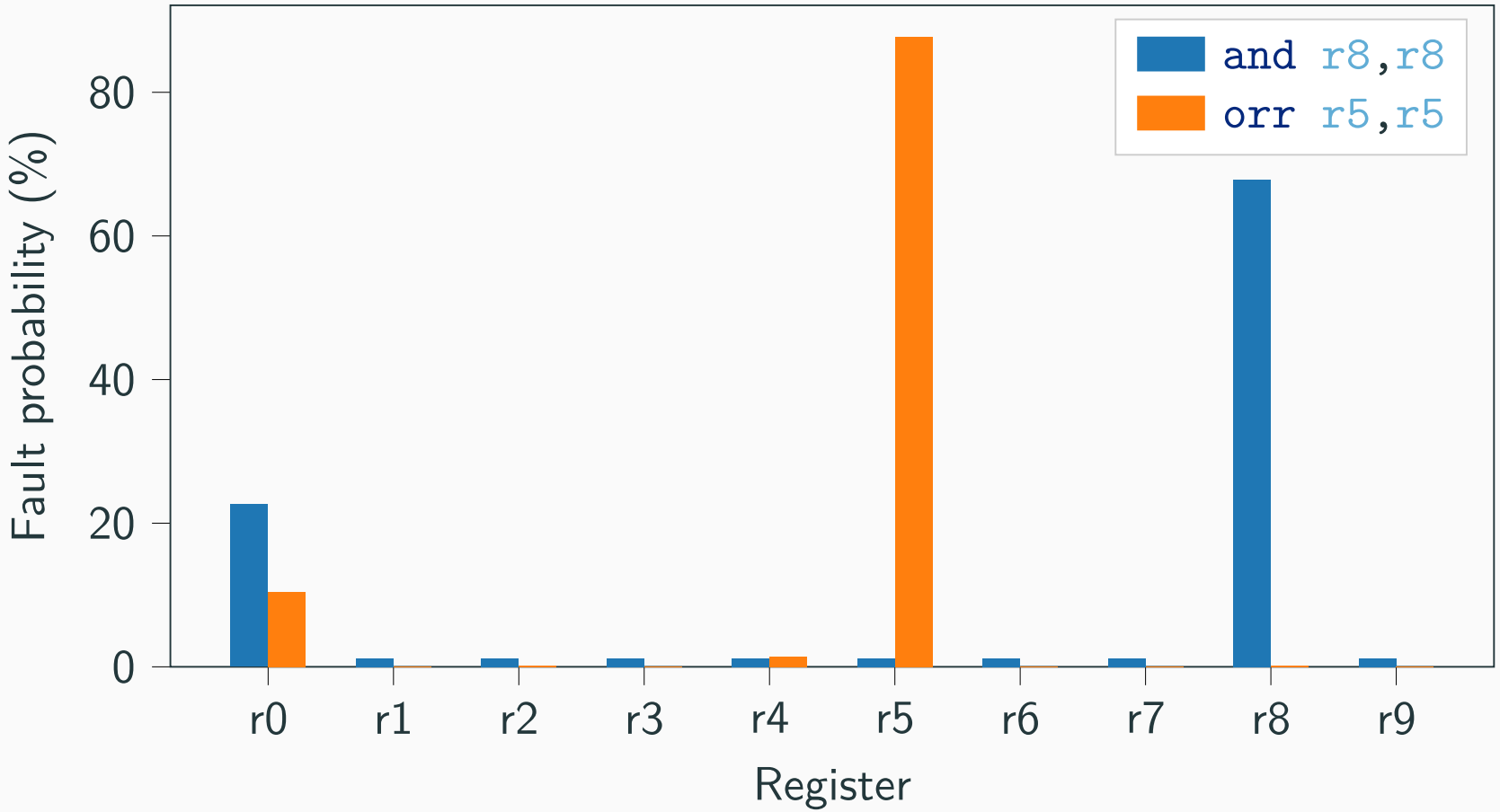
Spots leading to reboots



Spots leading to faults

Characterization - BCM2837

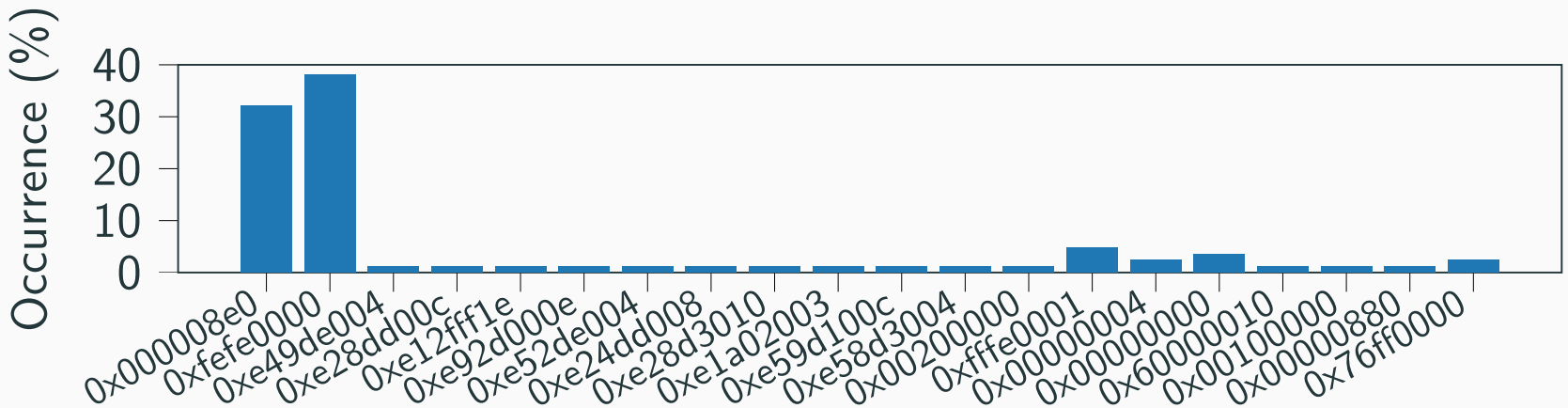
Faulted register distribution regarding the executed instruction



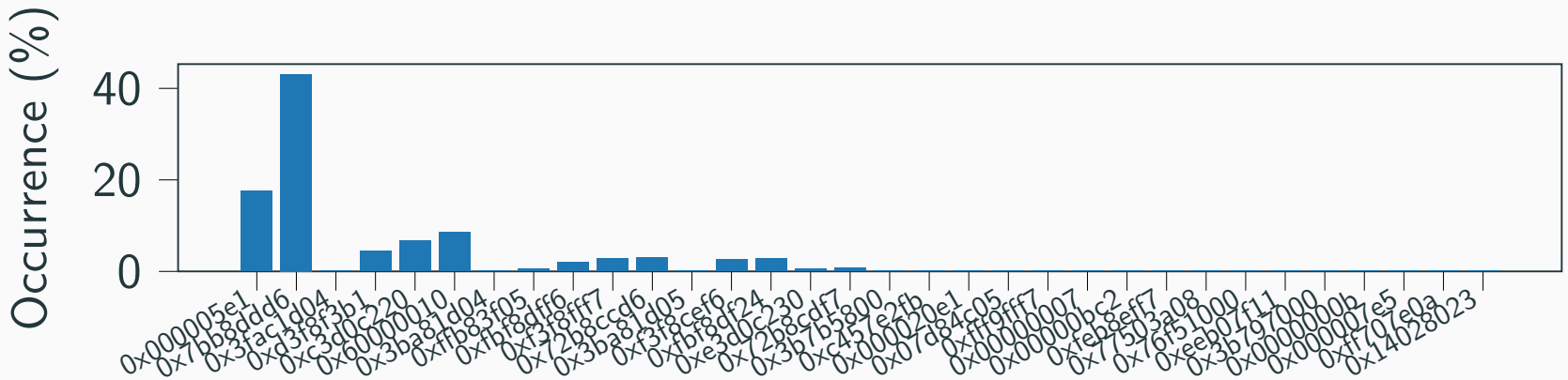
Characterization - BCM2837

Faulted value distribution regarding the executed instruction

`and r8,r8`



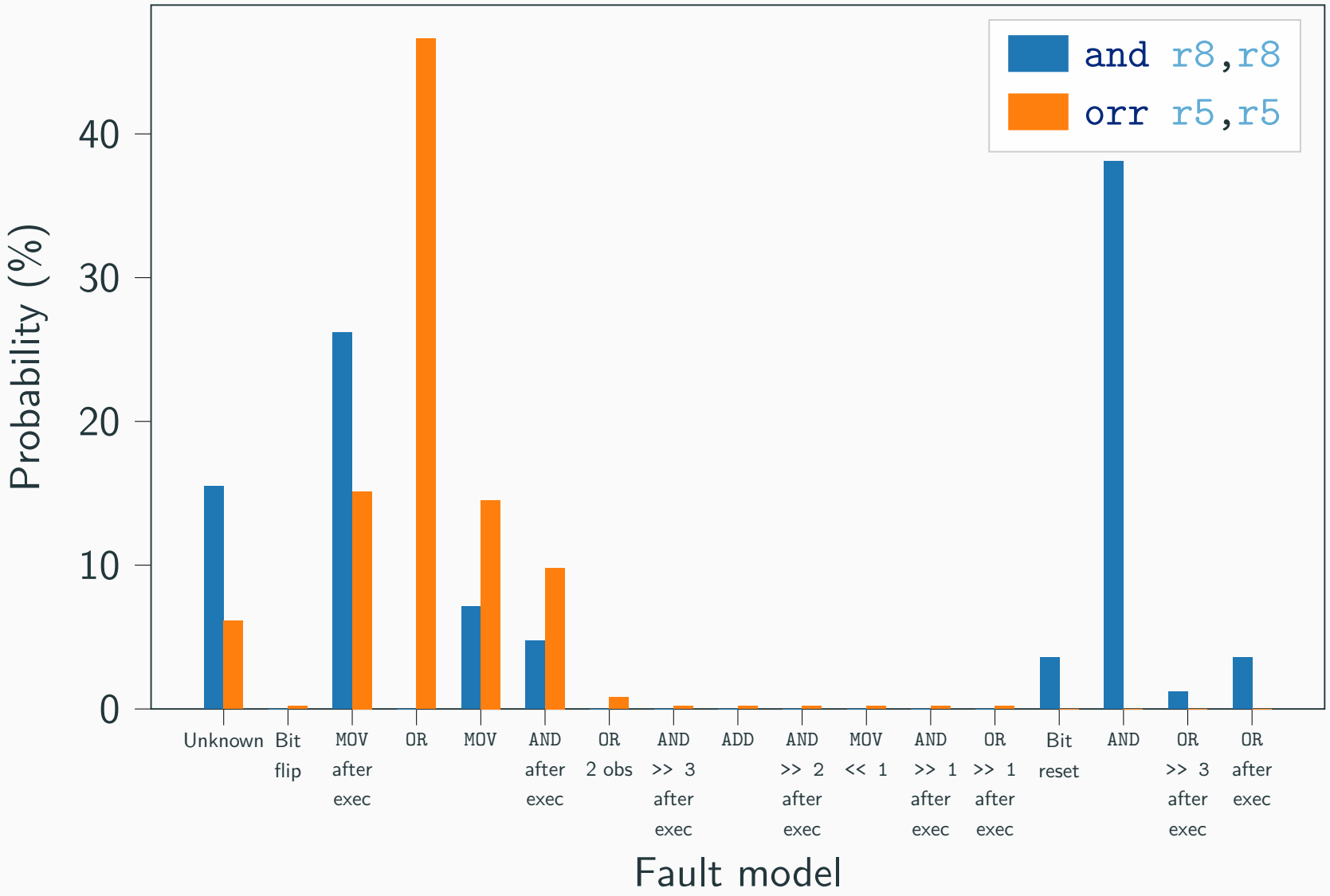
`orr r5,r5`



Faulted values

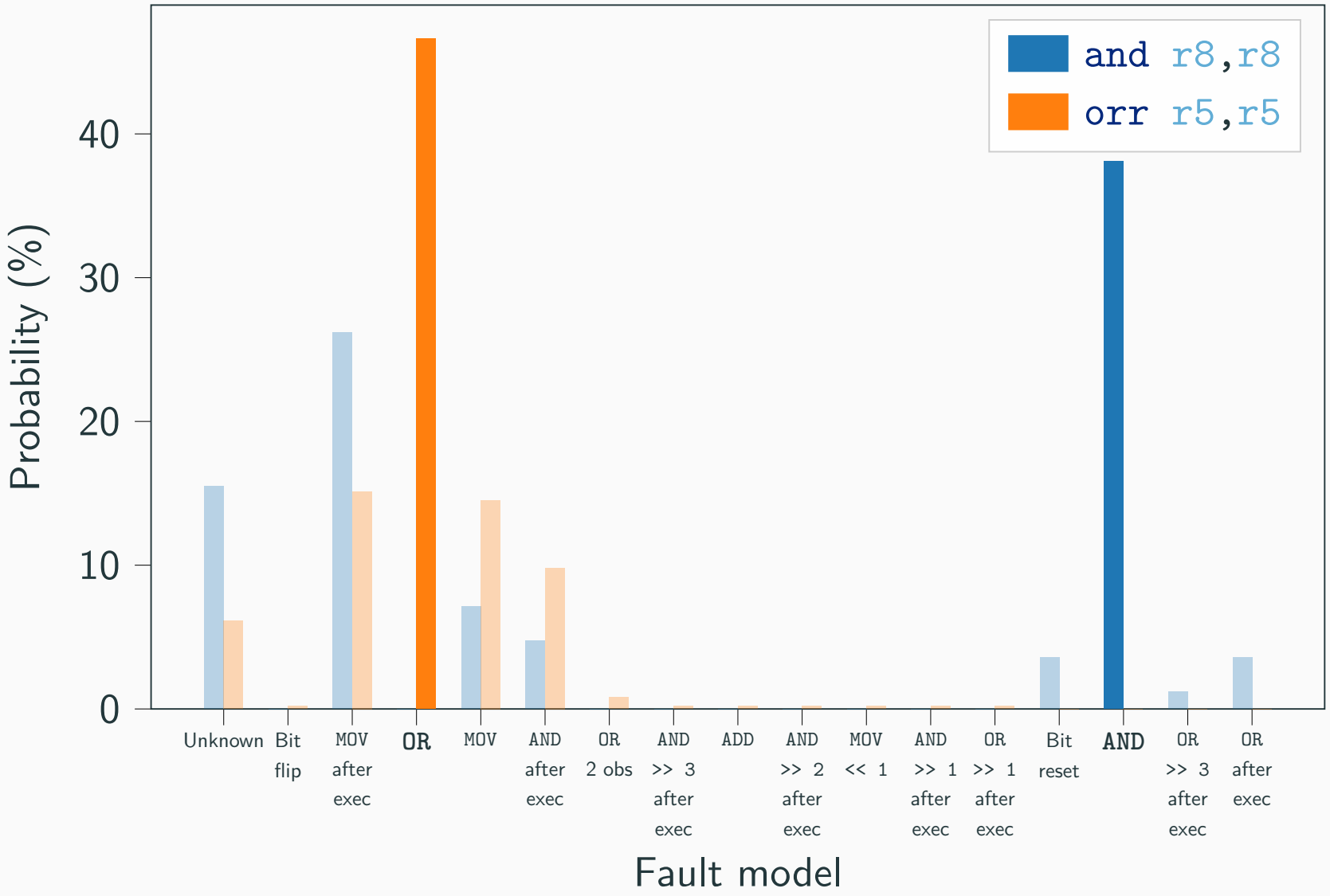
Characterization - BCM2837

Fault model distribution regarding the executed instruction



Characterization - BCM2837

Fault model distribution regarding the executed instruction



Characterization - BCM2837

Instruction matching the `OR` fault model for the `orr r5,r5` instruction

Faulted instruction	Occurrence (%)
`orr r5,r1`	92.54 %
`orr r5,r0`	6.14 %
`orr r5,r7`	1.32 %

Characterization - BCM2837

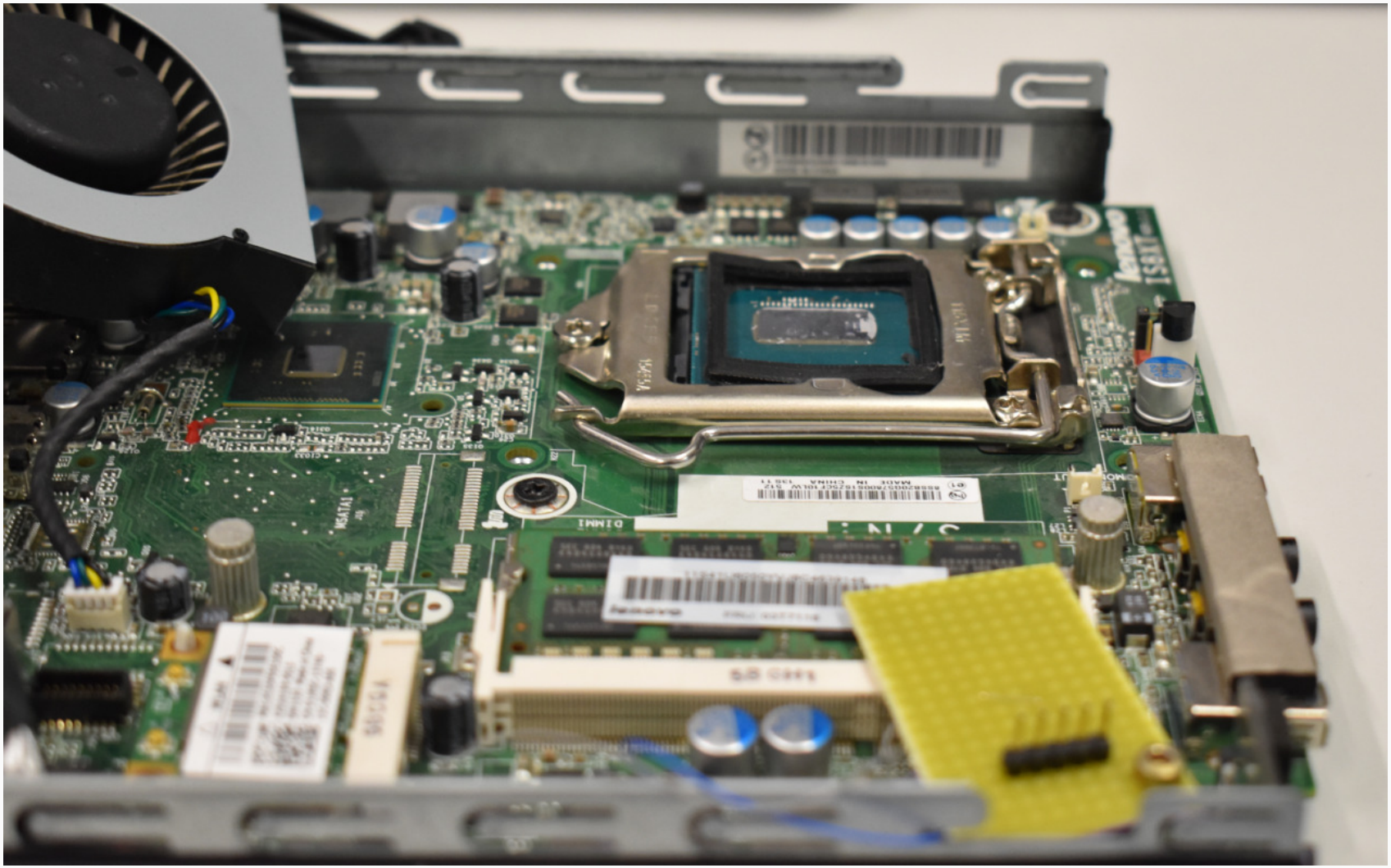
Instruction matching the `OR` fault model for the `orr r5,r5` instruction

Faulted instruction	Occurrence (%)
`orr r5,r1`	92.54 %
`orr r5,r0`	6.14 %
`orr r5,r7`	1.32 %

Instruction matching the `AND` fault model for the `and r8,r8` instruction

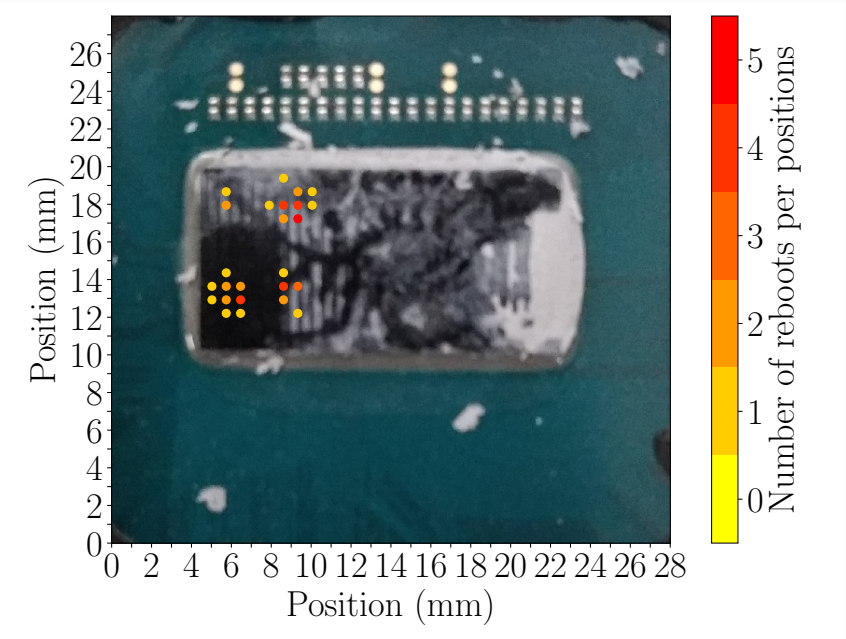
Faulted instruction	Occurrence (%)
`and r8,r0`	100 %

Characterization - Intel Core i3-6100T



Characterization - Intel Core i3-6100T

```
or rbx,rbx
```

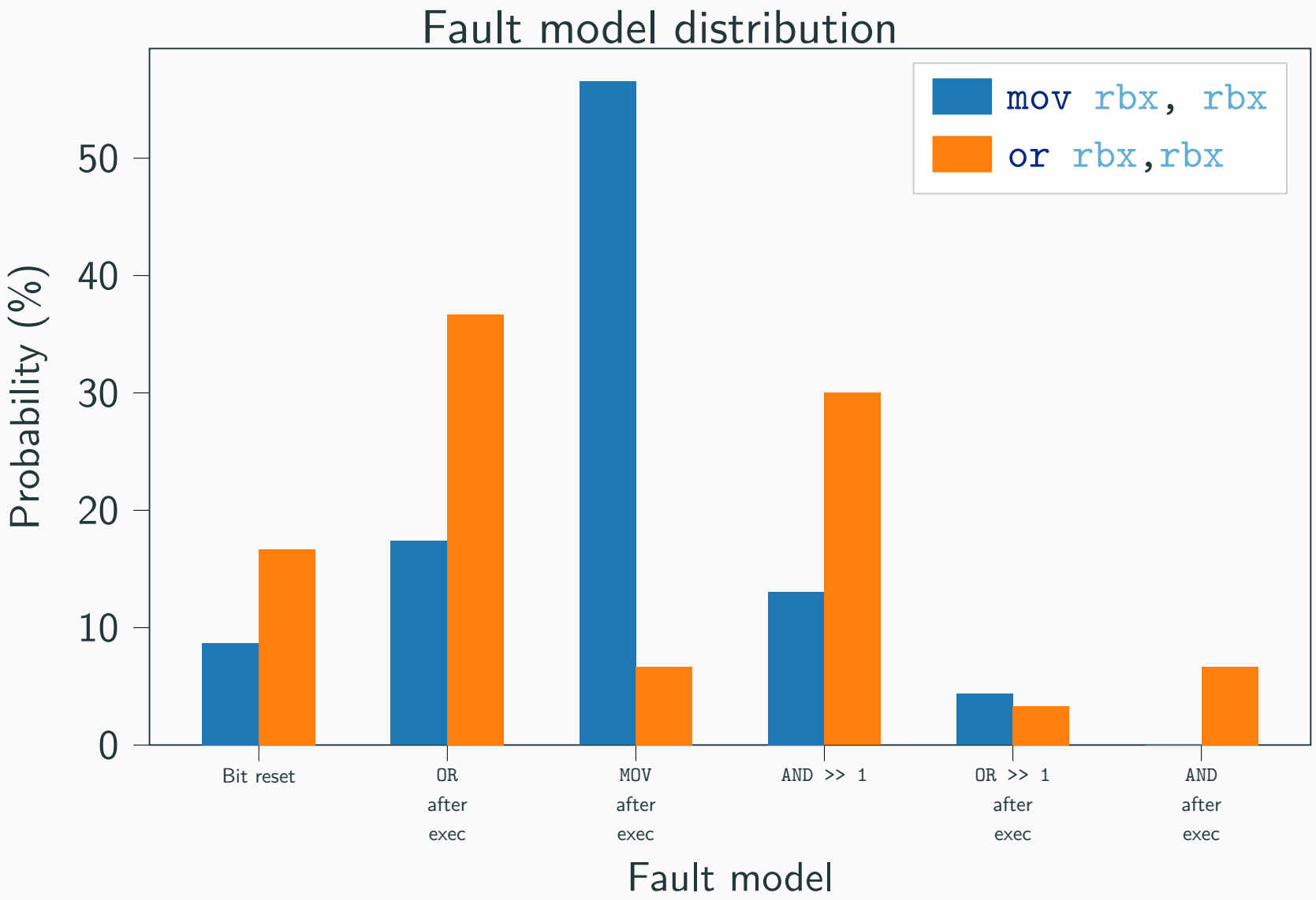


Spots leading to reboots

Faulted register:

- `rbx` in 100% of the cases

Characterization - Intel Core i3-6100T



Conclusion

- Different injection mediums have shown the similar fault models on different architecture (ARM, x86) and targets:
  - we suppose that there is an **underlying common mechanism** sensitive to perturbation
  - the **instruction cache** was identified as faulted on the BCM2837
  - EM fault injection is less efficient on flip chips
- These faults are suitable for an AES DFA

Questions ?

References

[DLM19] Mathieu Dumont, Mathieu Lisart, and Philippe Maurine. “Electromagnetic Fault Injection : How Faults Occur”. In: *2019 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2019, Atlanta, GA, USA, August 24, 2019*. IEEE, 2019, pp. 9–16. DOI: 10.1109/FDTC.2019.00010. URL: https://doi.org/10.1109/FDTC.2019.00010.

Bibliography ii

[OGM15] Sébastien Ordas, Ludovic Guillaume-Sage, and Philippe Maurine. "EM Injection: Fault Model and Locality". In: *2015 Workshop on Fault Diagnosis and Tolerance in Cryptography, FDTC 2015, Saint Malo, France, September 13, 2015*. Ed. by Naofumi Homma and Victor Lomné. IEEE Computer Society, 2015, pp. 3–13. DOI: 10.1109/FDTC.2015.9. URL: https://doi.org/10.1109/FDTC.2015.9.