Precise Exceptions in Relaxed Architectures

Ben Simner[1] Alasdair Armstrong[1] Thomas Bauereiss[1] Brian Campbell[2]
Ohad Kammar[2] Jean Pichon-Pharabod[3] Peter Sewell[1]

...in collaboration with Arm

[1]University of Cambridge [2]University of Edinburgh [3]Aarhus University

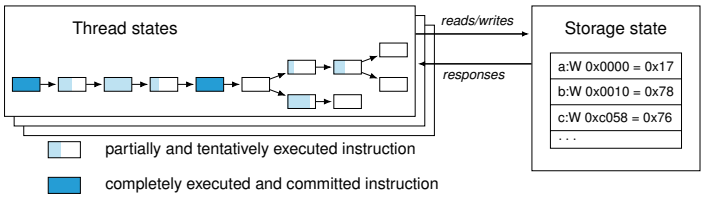
2025-06

Precision (à la Hennessy & Patterson):
exceptions appear to execute **between** instructions

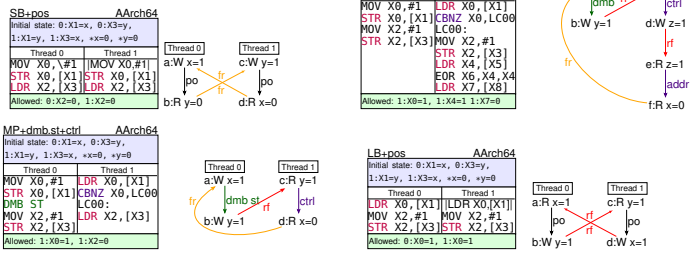
sequential definition
incompatible?
w/ relaxed architectures (e.g. Arm)

Relaxed memory — Observably out-of-order on Arm

Arm, RISC-V, and POWER allow observable out-of-order and speculative execution



[A tree of partially and completely executed fetch-decode-execute instances, on a single hardware thread, in real-hardware or operational-model execution.]

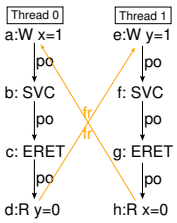


Precision: not just a fence?

- **out-of-order** execution **across exceptions**?
- **speculation**?
- store forwarding?
- context synchronisation?
- external aborts: **might-raise-exception**?
- interrupts? ...and the GIC?
- **how to specify all this**, w.r.t. the architectural intent, hardware behaviour, and system-software requirements?

Contributions

Testing hardware: Custom test harness with over 60 hand-written litmus tests.




```
rpi5 $ ./harness_kvm SB+svc-erets -n500k -q
Test SB+svc-erets Allowed
States 4
229760:>0:X2=1;1:X2=1;
95467:>0:X2=0;1:X2=1;
174771:>0:X2=1;1:X2=0;
2:>0:X2=0;1:X2=0;
Ok
Witnesses
Positive: 2 Negative: 499998
Observation SB+svc-erets Sometimes 2 499998
Time SB+svc-erets 214.449
```

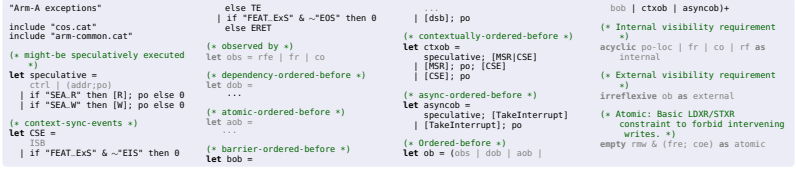
Name	m6g	m7g	m8g	odroid	m2	pi3	pi4	pi5
MP+dmb+ctrl-svc	0/16M	0/24M	0/12M	0/329M	0/360M	0/10M	0/230M	0/136M
MP+dmb+ctrlelr	0/16M	0/24M	0/12M	0/329M	0/360M	0/30M	0/318M	0/130M
MP+svc-eret+addr	U0/16M	U0/24M	U0/12M	149K/328M	U0/360M	376/9M	U0/228M	12/136M
MP.EL1+dmb+dataesrsvc	0/16M	0/24M	0/12M	0/16M	0/0	0/4M	0/14M	0/27M
S+dmb+svc	U0/16M	U0/24M	U0/12M	U0/328M	U0/360M	U0/41M	U0/222M	U0/101M
SB+dmb+eret	60/16M	120/24M	213/12M	262/328M	12K/360M	203K/41M	946K/222M	4K/100M
SB+dmb+rfisvc-addr	4/16M	235/24M	1K/12M	305K/328M	12/360M	1M/30M	7K/316M	197K/128M
MP+dmb+fault	0/16M	0/24M	0/12M	0/74M	0/0	0/2M	0/46M	0/80M

[Selected results. Observations/total runs. U =Allowed-but-unseen.]

A Model for Arm: OoO over exceptions.

Runnable in the Isla symbolic evaluator for Sail:

```
$ isla-axiomatic [...] -model exn.cat SB+svc-erets.litmus.toml
SB+svc-erets allowed (1 of 1) 2785ms ..................... ?
```



Acknowledgements We thank Richard Grisenthwaite (Arm EVP, Chief Architect, and Fellow), Martin Weidmann (Director of Product Management, Arm Architecture and Technology Group), and Will Deacon (Google) for detailed discussions about the Arm architecture. We thank Ben Laurie and Sarah de Haas (Google) for their support. We thank Jonathan Woodruff and others at the CL for their insightful discussions. This work was funded in part by Google. This work was funded in part by Arm. This work was funded in part by an AUFF starter grant (Pichon-Pharabod). This work was funded in part by two Amazon Research Awards (Pichon-Pharabod; Sewell and Simner). This work was funded in part by UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee for ERC-AdG-2022, EP/Y035976/1 SAFER. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 789108, ERC-AdG-2017 ELVER). This work is supported by ERC-2024-POC grant ELVER-CHECK, 101189371. Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. This work was supported in part by the Innovate UK project Digital Security by Design (DSbD) Technology Platform Prototype, 105694. The authors would like to thank the Isaac Newton Institute for Mathematical Sciences, Cambridge, for support and hospitality during the programme Big Specification, where work on this paper was undertaken. This work was supported by EPSRC grant EP/Z000580/1. This work was funded in part by a Royal Society University Research Fellowship. One of the authors has received funding from the UK Advanced Research and Innovation Agency (ARIA) as part of the project Qbs4Safety: Core Representation Underlying Safeguarded AI.