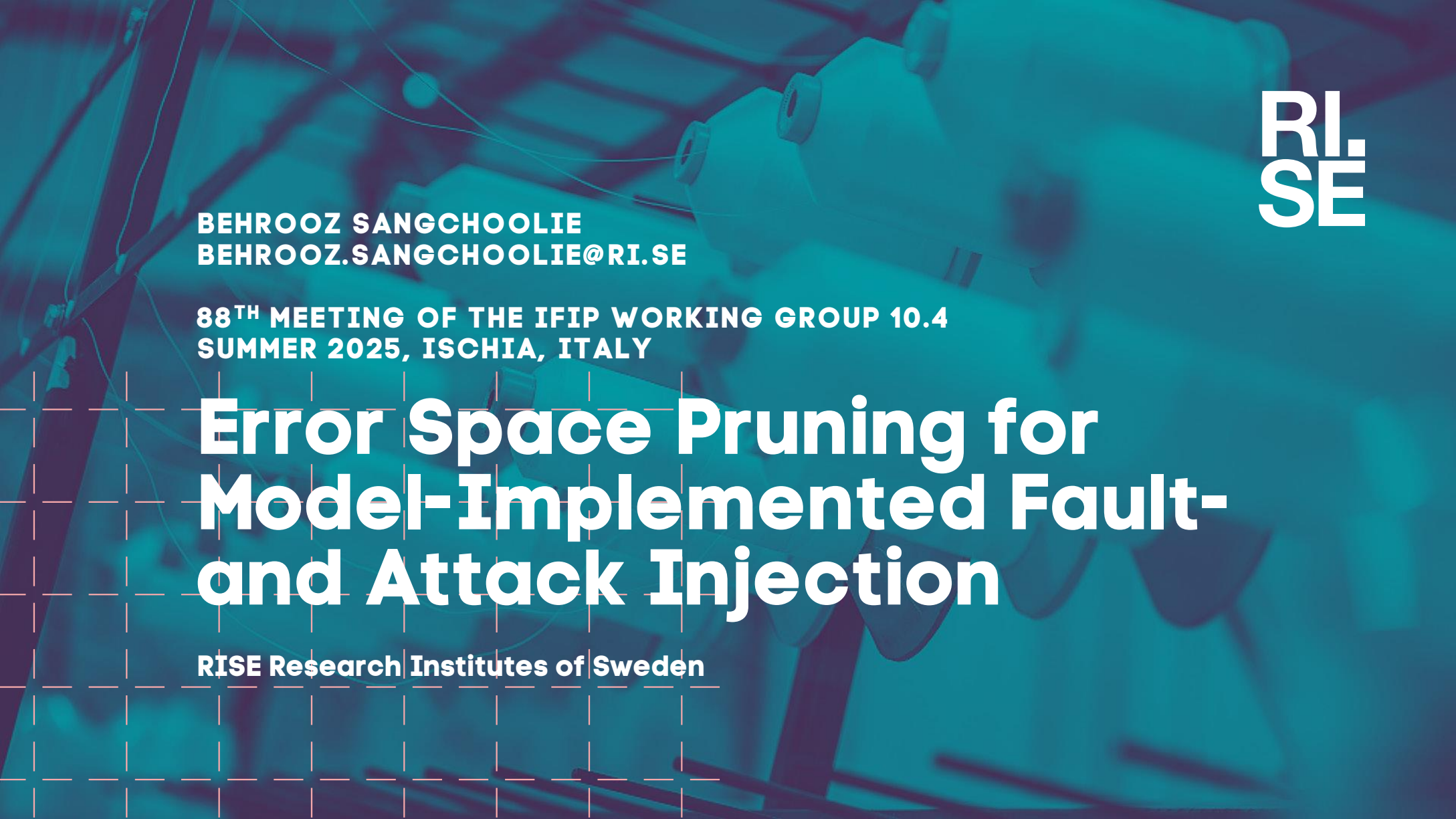


BEHROOZ SANGCHOOLIE
BEHROOZ.SANGCHOOLIE@RI.SE

88TH MEETING OF THE IFIP WORKING GROUP 10.4
SUMMER 2025, ISCHIA, ITALY

Error Space Pruning for Model-Implemented Fault- and Attack Injection

RISE Research Institutes of Sweden

Problem Description

- Increasing need for **resilience** towards faults and cybersecurity attacks
- Resilience may be achieved by layers of **mechanisms** for **detecting** and **handling** attacks
- In order to **test** these mechanisms, experimental verification techniques such as **fault- and attack injection** can be employed
- Due to the complexity of systems, the **efficiency** of fault- and attack injection in terms of the time and effort needed to explore the fault- or attack space (= **error space**) important
- Several techniques for **reducing** the error space to improve efficiency proposed[1-4]

[1] *B. Sangchoolie, F. Ayatolahi, R. Johansson, and J. Karlsson. "A Comparison of Inject-on-Read and Inject-on-Write in ISA-Level Fault Injection", EDCC 2015.*

[2] *A. C. Bagbaba, M. Jenihhin, J. Raik, and C. Sauer. "Efficient Fault Injection based on Dynamic HDL Slicing Technique". In: CoRR abs/2002.00787, 2020.*

[3] *B. Sangchoolie, K. Pattabiraman, and J. Karlsson. "An Empirical Study of the Impact of Single and Multiple Bit-Flip Errors in Programs". In: IEEE Transactions on Dependable and Secure Computing 19.3, 2022.*

[4] *I. Tuzov, D. de Andres, and J.-C. Ruiz. "Reversing FPGA architectures for speeding up fault injection: does it pay?", EDCC 2022.*

RI. SE

Background – error space reduction

- ***Inject-on-read***[1,2]: Faults and attacks injected into resource immediately before resource is **read (used)**
- ***Inject-on-write***[3,4]: Faults and attacks injected into resource immediately after **written (created or updated)**
- ***Fault list collapsing***[5,6]: Collapse faults determined to be equivalent into equivalence classes
- ***Code-slicing***[7]: Determine source code statements targeted for injection in order to affect a certain criterion (target system output)

[1] *R. Barbosa, J. Vinter, P. Folkesson, and J. Karlsson. "Assembly-Level Pre-injection Analysis for Improving Fault Injection Efficiency", EDCC 2005.*

[2] *G. Munkby and S. Schupp. "Improving Fault Injection of Soft Errors Using Program Dependencies". In: Testing: Academic Industrial Conference - Practice and Research Techniques. 2008.*

[3] *J. Grinschgl, A. Krieg, C. Steger, R. Weiss, H. Bock, and J. Haid. "Efficient fault emulation using automatic pre-injection memory access analysis". In: 2012 IEEE International SOC Conference. 2012.*

[4] *B. Sangchoolie, F. Ayatolahi, R. Johansson, and J. Karlsson. "A Comparison of Inject-on-Read and Inject-on-Write in ISA-Level Fault Injection", EDCC 2015.*

[5] *L. Berrojo, I. Gonzalez, F. Corno, M. Reorda, G. Squillero, L. Entrena, and C. Lopez. "New techniques for speeding-up fault-injection campaigns". In: Proc. 2002 Design, Automation and Test in Europe Conference and Exhibition. 2002.*

[6] *D. Smith, B. Johnson, and J. Profeta. "System dependability evaluation via a fault list generation algorithm". In: IEEE Transactions on Computers 45.8, 1996.*

[7] *A. C. Bagbaba, M. Jenihhin, J. Raik, and C. Sauer. "Efficient Fault Injection based on Dynamic HDL Slicing Technique". In: CoRR abs/2002.00787, 2020.*

RI. SE

Background – error space reduction

- ***Post-injection analysis***[1-4]: Analyse results from fault/attack injection experiments to determine what experiments to perform next
- ***Error space pruning***[5-8]: Prune faults and attacks with known outcome or equivalent to other faults and attacks determined through:
  - ***Static*** (pre-injection) analyses
  - ***Dynamic*** (post-injection) analyses

[1] *E. W. Czeck and D. P. Siewiorek. "Observations on the Effects of Fault Manifestation as a Function of Workload". In: IEEE Transactions on Computers, 41.5, 1992.*

[2] *J. Aidemark, P. Folkesson, and J. Karlsson. "Path-based error coverage prediction". In: Proceedings Seventh International On-Line Testing Workshop. 2001.*

[3] *P. Folkesson and J. Karlsson. "Considering Workload Input Variations in Error Coverage Estimation". EDCC 1999.*

[4] *B. Sangchoolie, K. Pattabiraman, and J. Karlsson. "One Bit is (Not) Enough: An Empirical Study of the Impact of Single and Multiple Bit- Flip Errors". In: 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). 2017.*

[5] *S. K. S. Hari, S. V. Adve, H. Naeimi, and P. Ramachandran. "Relyzer: Exploiting Application-Level Fault Equivalence to Analyze Application Resiliency to Transient Faults". In: Proc. of the 17th International Conference on Architectural Support for Programming Languages and Operating Systems, 2012.*

[6] *F. Ayatolahi, B. Sangchoolie, R. Johansson, and J. Karlsson. "A Study of the Impact of Single Bit-Flip and Double Bit-Flip Errors on Program Execution". In: Computer Safety, Reliability, and Security., 2013.*

[7] *I. Tuzov, D. de Andres, and J.-C. Ruiz. "Reversing FPGA architectures for speeding up fault injection: does it pay?", EDCC 2022.*

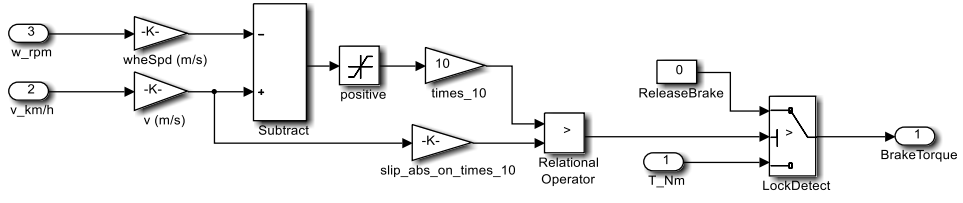
[8] *B. Sangchoolie, K. Pattabiraman, and J. Karlsson. "An Empirical Study of the Impact of Single and Multiple Bit-Flip Errors in Programs". In: IEEE Transactions on Dependable and Secure Computing 19.3, 2022.*

RI. SE

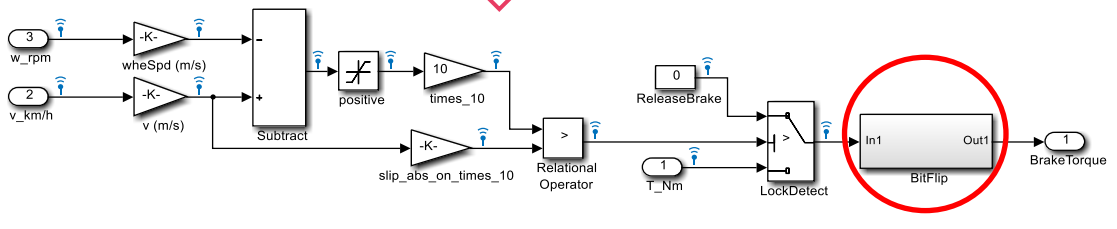
Model-implemented fault/attack injection

- Useful for early dependability evaluation of software developed as models
- Injections performed using fault and attack injection blocks inserted into target system model

Original model

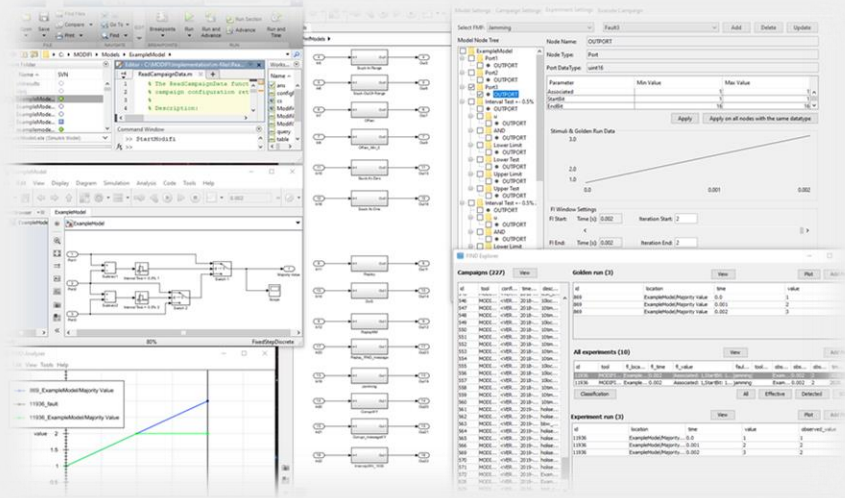


Model with fault/attack injection support



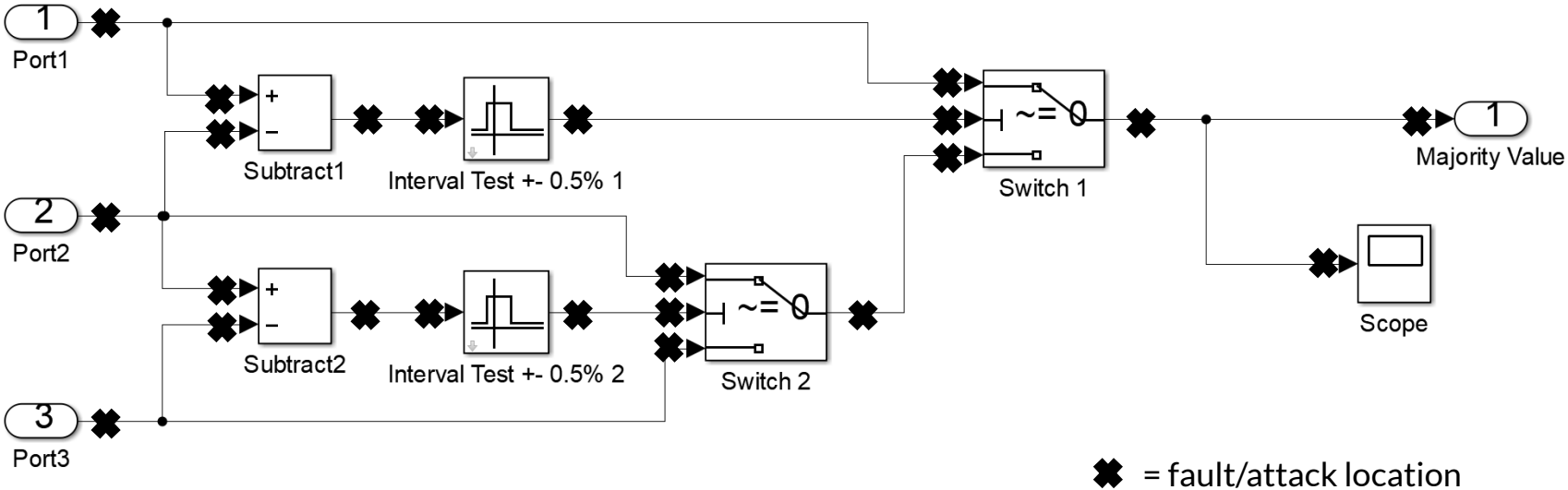
MODIFI tool

- MODIFI[1] is a fault/attack injection tool for Simulink models
  - Provides a large number of fault/attack models, e.g., *bit-flip faults*, *stuck-at faults*, *sensor faults*, *replay attacks*, *jamming attacks*, …
  - Includes support for analyzing and visualizing fault/attack injection results
  - Simulink models are commonly used in the automotive and avionic domains.



[1] *R. Svenningsson, J. Vinter, H. Eriksson, and M. Törngren. "MODIFI: A MODel-Implemented Fault Injection Tool". In: Computer Safety, Reliability, and Security, 2010.*

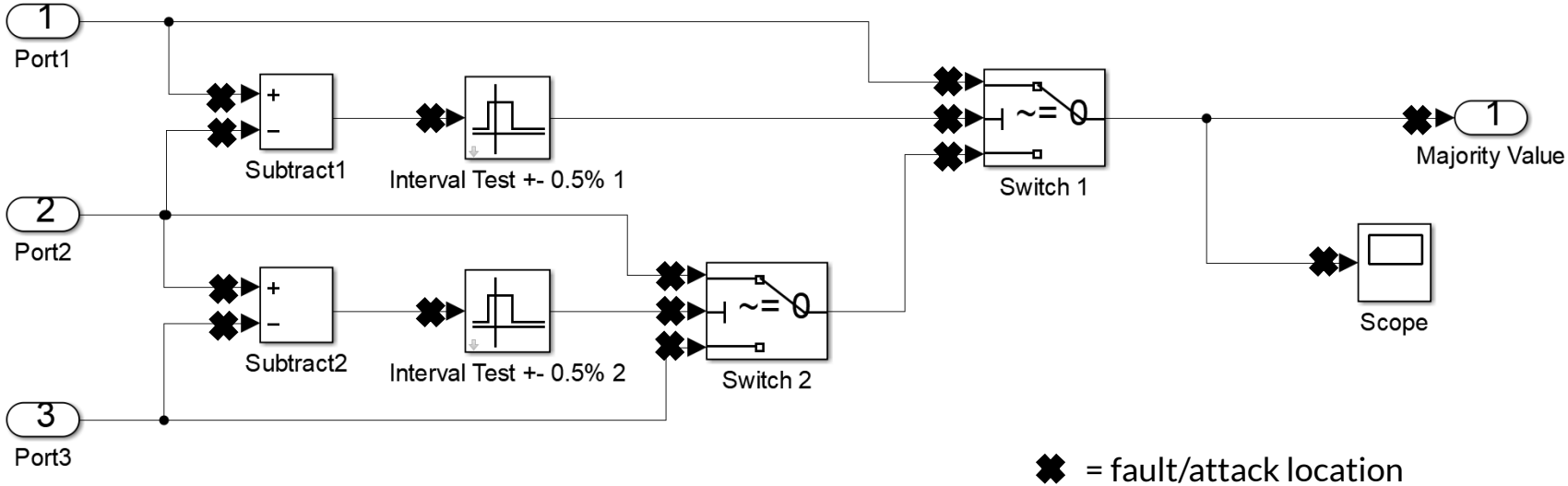
Pre-injection in MODIFI:
No pre-injection analysis



✖ = fault/attack location
⇒ 23 locations

RI. SE

Pre-injection in MODIFI: *Inject-on-read*

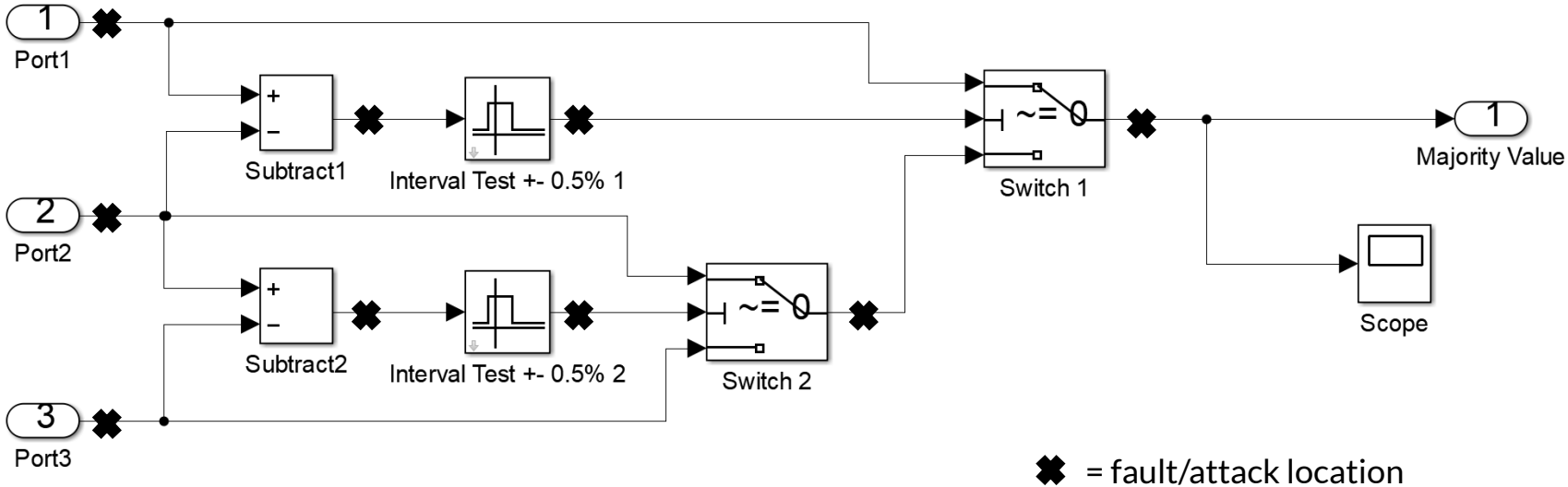


✖ = fault/attack location
⇒ 14 locations
⇒ 23-14 = 9 locations reduction
⇒ 39 % reduction in error space

Folkesson P, Sangchoolie B, Kleberger P: On the evaluation of three pre-injection analysis techniques suitable for model-implemented fault- and attack injection.In: 27th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2022). 2022.

RI. SE

Pre-injection in MODIFI: *Inject-on-write*

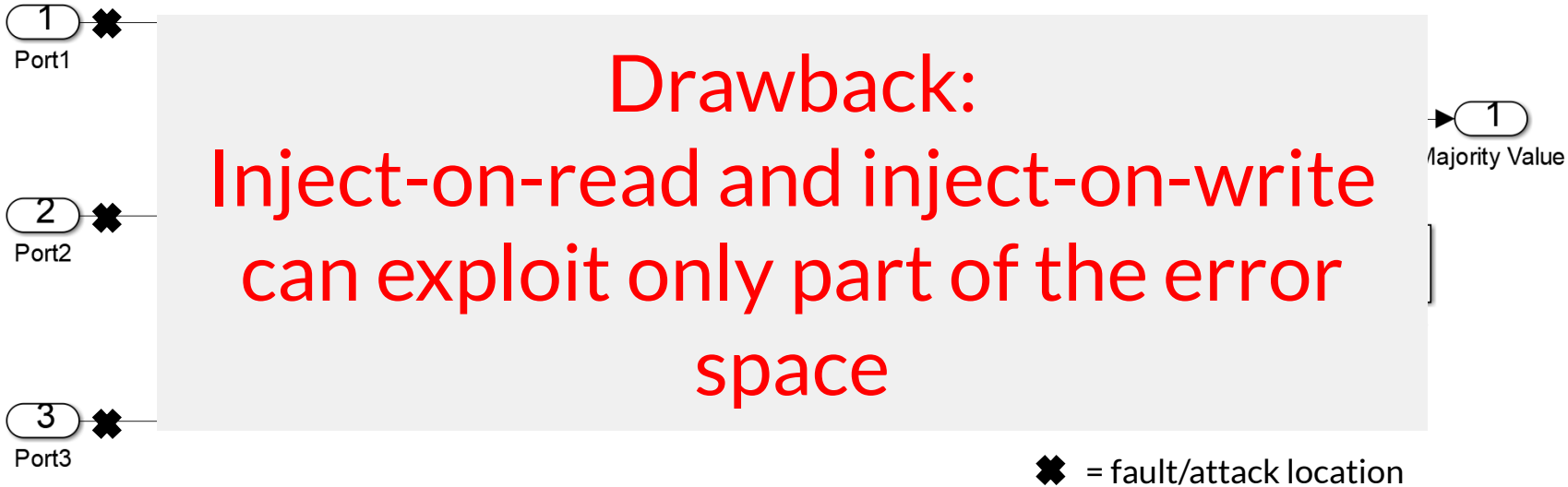


✖ = fault/attack location
⇒ 9 locations
⇒ 23-9 = 14 locations reduction
⇒ 61 % reduction in error space

Folkesson P, Sangchoolie B, Kleberger P: On the evaluation of three pre-injection analysis techniques suitable for model-implemented fault- and attack injection.In: 27th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2022). 2022.

RI. SE

Pre-injection in MODIFI: *Inject-on-write*



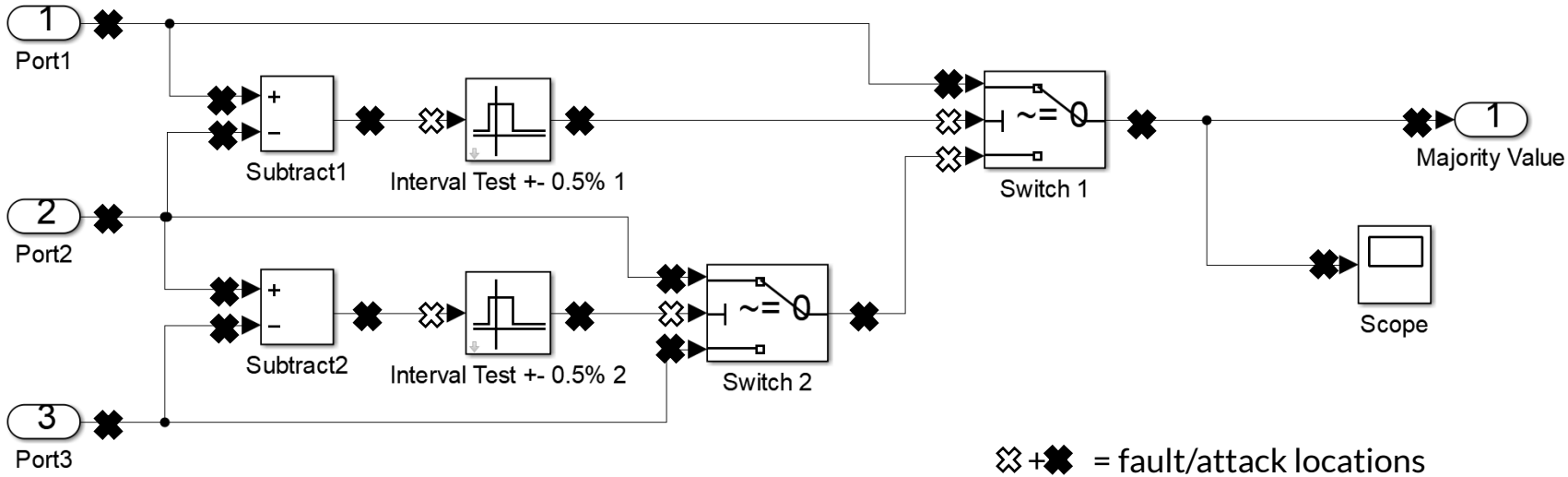
Drawback:
Inject-on-read and inject-on-write can exploit only part of the error space

✖ = fault/attack location
⇒ 9 locations
⇒ 23-9 = 14 locations reduction
⇒ 61 % reduction in error space

Folkesson P, Sangchoolie B, Kleberger P: On the evaluation of three pre-injection analysis techniques suitable for model-implemented fault- and attack injection.In: 27th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2022). 2022.

RI. SE

Pre-injection in MODIFI:
Error space pruning of signals

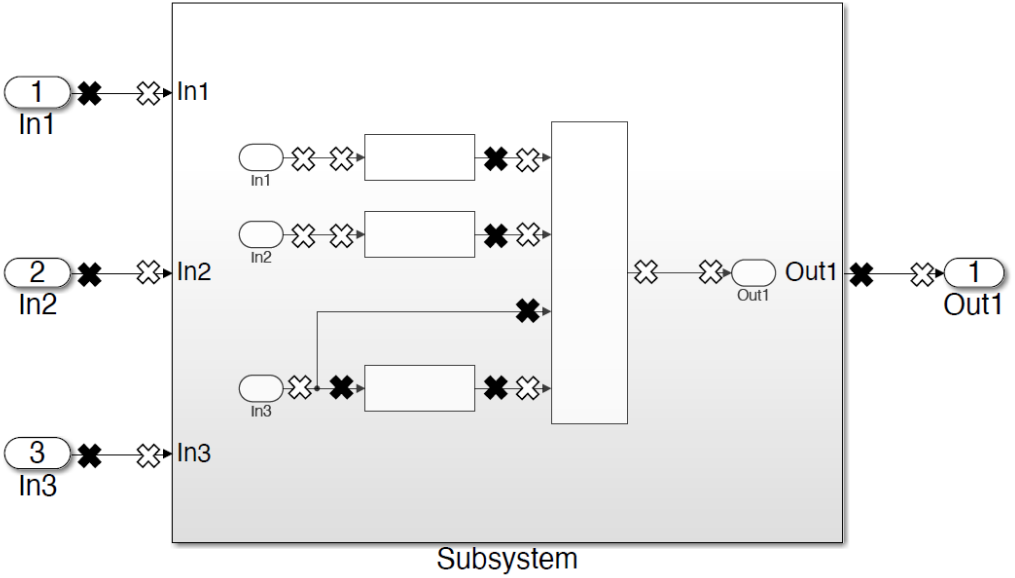


✖+✖ = fault/attack locations
⇒ 23 locations
✖ = pruned locations
⇒ 5 locations pruned
⇒ 22 % reduction in error space

Folkesson P, Sangchoolie B, Kleberger P: On the evaluation of three pre-injection analysis techniques suitable for model-implemented fault- and attack injection.In: 27th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2022). 2022.

RI. SE

Pre-injection in MODIFI:
Error space pruning of signals and ports

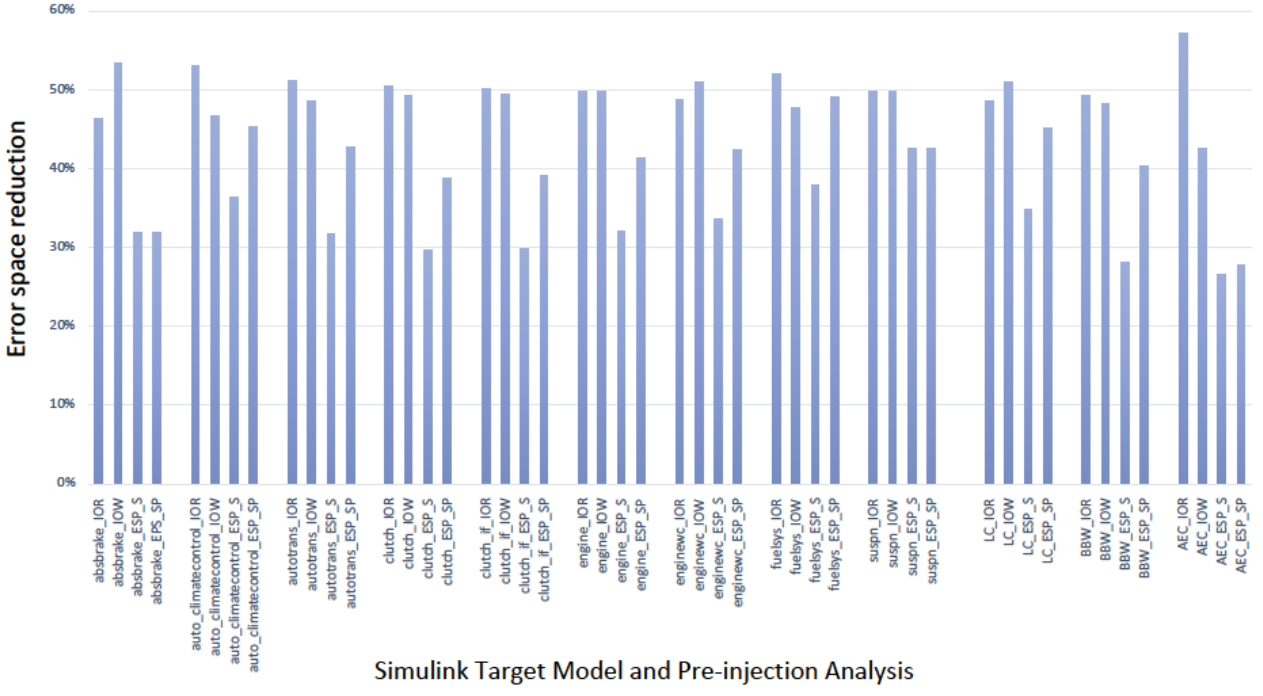


✖ = fault/attack locations
⇒ 23 locations
✖ = pruned locations
⇒ 14 locations pruned
⇒ 61 % reduction in error space

RI. SE

Experimental results: Error space reduction comparison

- Pre-injection analysis performed for 9 Mathworks automotive example models
- Comfort control model (**LC**), Brake-By-Wire model (**BBW**), Aero Engine Control Model (**AEC**)



Discussions

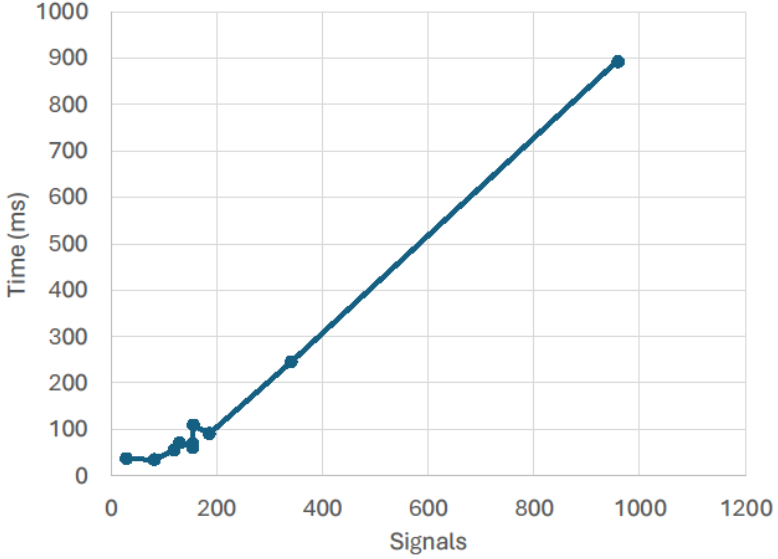
- The error space pruning techniques are built with the assumption of **“faults in equivalence classes lead to the same outcome”**.
  - Fault injection results show that the assumption holds.

RI.
SE

Discussions

- The error space pruning techniques are built with the assumption of **“faults in equivalence classes lead to the same outcome”**.
  - Fault injection results show that the assumption holds.
- The computational cost of the error space pruning techniques is in **order of seconds** and **scales linearly** with the number of injectable signals in the model.
  - Depending on the size of the model under test, each injection could take minutes.

  ➔ **Significant reduction in testing time.**

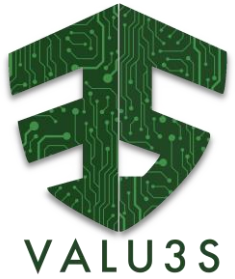


RI. SE

Acknowledgement

This project has received funding from the Chips Joint Undertaking (JU) under Grant Agreement No. 101095835 (project AGRARSENSE). The JU receives support from the European Union's Horizon 2020 research and innovation programme and Sweden, Spain, France, Ireland, Austria, the Netherlands, Italy, Poland, Germany, Norway, Finland, Latvia, Czechia, Türkiye.
Disclaimer: The Chips JU and the European Commission are not responsible for the content on this presentation or any use that may be made of the information it contains.

This project has received funding from Swedish VINNOVA FFI project (Diary number: 2018- 05013, 2019-03071)



This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 876852. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Austria, Czech Republic, Germany, Ireland, Italy, Portugal, Spain, Sweden, Turkey.
Disclaimer: The ECSEL JU and the European Commission are not responsible for the content on this presentation or any use that may be made of the information it contains.

RI.SE