Chair of Electronic Design Automation
Department of Computer Engineering
School of Computation, Information and Technology
Technical University of Munich

Techniques and Tools for Fast Fault Injection Simulations of RISC-V Processors at RTL

Johannes Geier, Daniel Mueller-Gritschneder, and Ulf Schlichtmann

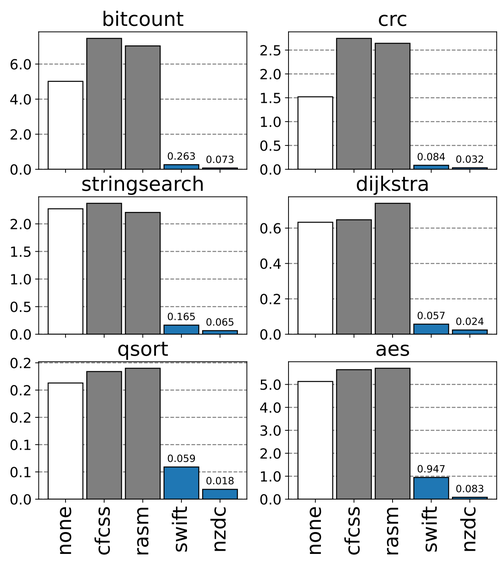
Motivation: Evaluating Soft Error Resilience

Functional Safety (FuSa)

- Fault Tolerance / Fault Detection
- Fault Model: **Random**
- Example: Cosmic and Package Radiation, EMF
- Statistic Evaluation (ISO-26262)
- Metrics: Silent Data Corruption (SDC), Detectable Unrecoverable Error (DUE)

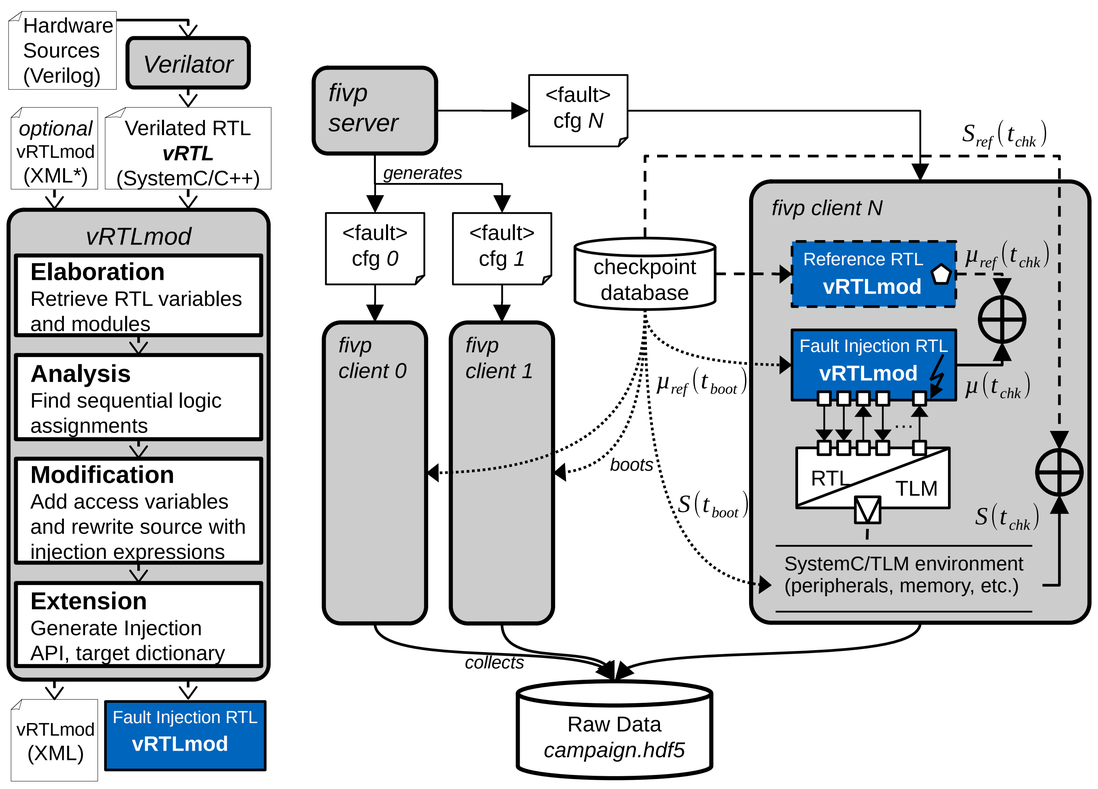
Security

- Fault Detection
- Fault Model: **Targeted**
- Example: Fault Injection Attack, DFA
- Verification against Attack Vectors
- Metrics: e.g., Attack Feasibility



[3] SDC Rates MiBench w/ Software Implemented Hardware Fault Tolerance

vRTLmod: Verilator RTL Fault Injection Modification



vRTLmod Flow [1]

Usage of vRTLmod Output in Virtual Platform without *masking* [1,2]. With *masking checks* (this work).

Input: Cycle-accurate SystemC/C++ models of Verilog RTL
Ouput: Fault Injectable vRTL (vRTLmod)

- Source-Code transformation on vRTL (LLVM/Clang Frontend Tool)
- Automatic insertion of injection expressions in source code
- Small overhead (ca. 10%) compared to plain vRTL
- Integrate as Module in Transaction-level Virtual Platform SoC

Checkpoint Restore Boot (CRB)

Setup:

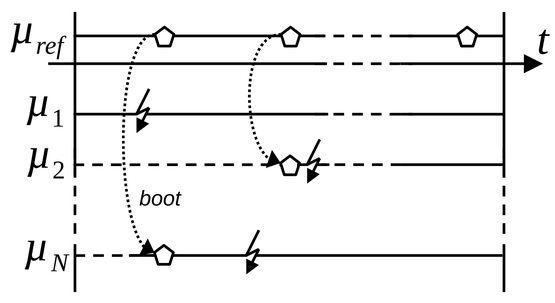
- Pre-recorded Checkpoints ⬠ during fault-free reference simulation

Warm-up:

- Select ⬠ as close as possible before fault injection point ↯

Cooldown:

- Execute simulation until the end



State-of-the-art and widely used technique to accelerate FI simulations.
→ Around 50% save for uniformly distributed experiments

Checkpoint Restore Boot with Masking (CMSK)

Setup:

- Same as **CRB**. Pre-recorded checkpoints ⬠ from reference simulation
- Add a dummy Reference CPU core

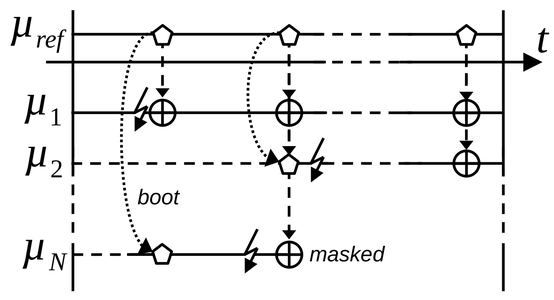
Warm-up:

- Select ⬠ as close as possible before fault injection point ↯

Cooldown:

- After injection during cooldown, load ⬠ into dummy Reference CPU core
- Perform masking check with a bit-wise comparison of sequential logic: $\mu(t) \bigoplus \mu_{ref}(t) \stackrel{?}{=} \emptyset$

→ Additional 15-25% save for uniformly sampled experiments



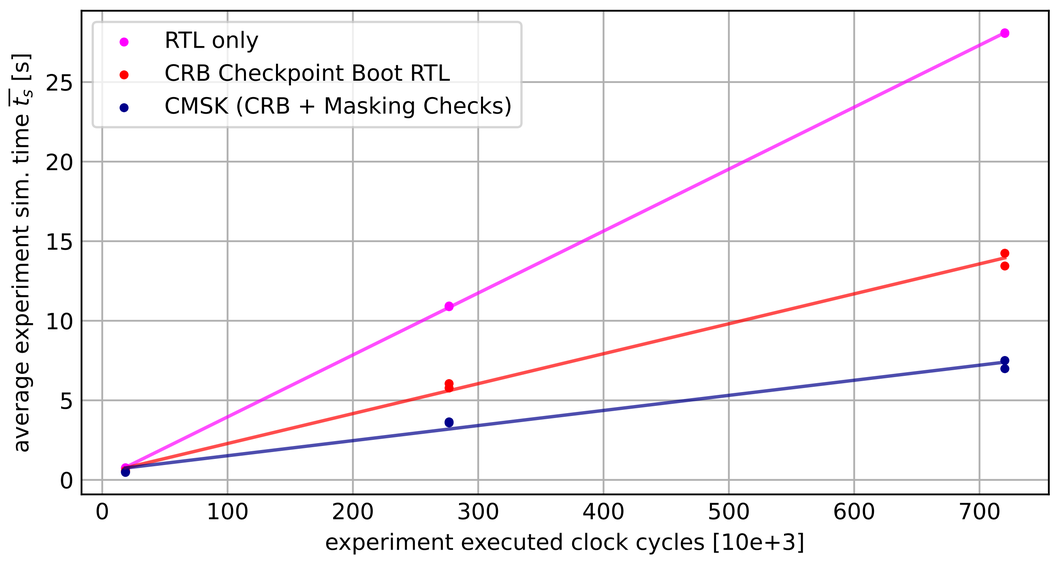
Experimental Results

Experimental Setup:

- Simple SoC with **cv32e40p**[a] RISC-V *rv32imac* core as fault injection target
- Running Embench™ IoT benchmark programs [b]
- Average simulation time $\overline{t_s}$ and classification of experiments conducted on differently configured RTL FI. Checkpoint interval of 10,000 clock cycles. 11,000 equal experiments per benchmark and configuration combination.

Fault Classification:

MSK Fault masked and detected with **CMSK** feature
DUE Detectable Unrecoverable Error (exceptions, bus faults, timeouts)
SDC Silent Data Corruption
pLAT Potentially Latent, i.e., not an SDC, MSK, or DUE



benchmark	CC[K]	config	$\overline{t_s}$	save[%]	MSK	[%]	DUE	[%]	SDC	[%]	pLAT	[%]
aha-mont64	18.5	RTL	0.75	-	0	0.00	199	1.81	567	5.15	10,234	93.0
aha-mont64	18.5	CRB	0.56	25.3	0	0.00	199	1.81	567	5.15	10,234	93.0
aha-mont64	18.5	CMSK	0.48	36.0	2,821	25.65	199	1.81	567	5.15	7,413	67.4
huffbench	276	RTL	10.9	-	0	0.00	478	4.35	223	2.03	10,299	93.6
huffbench	276	CRB	5.91	45.8	0	0.00	478	4.35	223	2.03	10,299	93.6
huffbench	276	CMSK	3.62	66.8	4,875	44.32	478	4.35	223	2.03	5,424	49.3
picojpeg	720	RTL	28.1	-	0	0.00	248	2.25	71	0.65	10,681	97.1
picojpeg	720	CRB	13.8	50.9	0	0.00	248	2.25	71	0.65	10,681	97.1
picojpeg	720	CMSK	7.25	74.2	5,592	50.84	248	2.25	71	0.65	5,089	46.3

[a] M. Gautschi et al., "Near-Threshold RISC-V Core With DSP Extensions for Scalable IoT Endpoint Devices," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 10, pp. 2700-2713, Oct. 2017, doi: 10.1109/TVLSI.2017.2654506.
[b] Embench™. 2024. url: https://github.com/embench.

Publications

[1] J. Geier and D. Mueller-Gritschneder. 2023. "vRTLmod : An LLVM based Open-source Tool to Enable Fault Injection in Verilator RTL Simulations." In *20th ACM International Conference on Computing Frontiers (CF '23)*. Association for Computing Machinery, (May 2023). DOI: 10.1145/3587135.3591435
[2] S.Pircher, J. Geier, J. Danner, D. Mueller-Gritschnedern and A. Wachter-Zeh. 2023. "Key-recovery fault injection attack on the classic McEliece KEM. In Code-Based Cryptography". In *Code-Based Cryptography*. Jean-Christophe Deneuville, editor. Springer Nature Switzerland, Cham, 37–61. ISBN: 978-3-031-29689-5
[3] U. Sharif, D. Mueller-Gritschneder, U. Schlichtmann. Compas: Compiler-assisted Software-implemented Hardware Fault Tolerance for RISC-V. In *2022 11th Mediterranean Conference on Embedded Computing (MECO)*, 1-4- DOI: 10.1109/MECO55406.2022.9797144.

[1,2] Research was partly funded by the Bavarian State Ministry for Economic Affairs, Regional Development and Energy as part of the Bavarian funding program for research and development "Information and Communication Technology" through the project MITHRIL, grant number IUK623/002.

This work has been developed in the project MANNHEIM-FlexKI funded by the German Federal Ministry of Education and Research (BMBF) under contract no.01IS22086L.

SPONSORED BY THE Federal Ministry of Education and Research

Contact: johannes.geier@tum.de

Open Source: https://github.com/tum-ei-eda/vrtlmod