

Institute of Computer Engineering - Chair of Adaptive Dynamic Systems

FT-CGRA: Compiler-Transparent Fault Tolerance for CGRAs

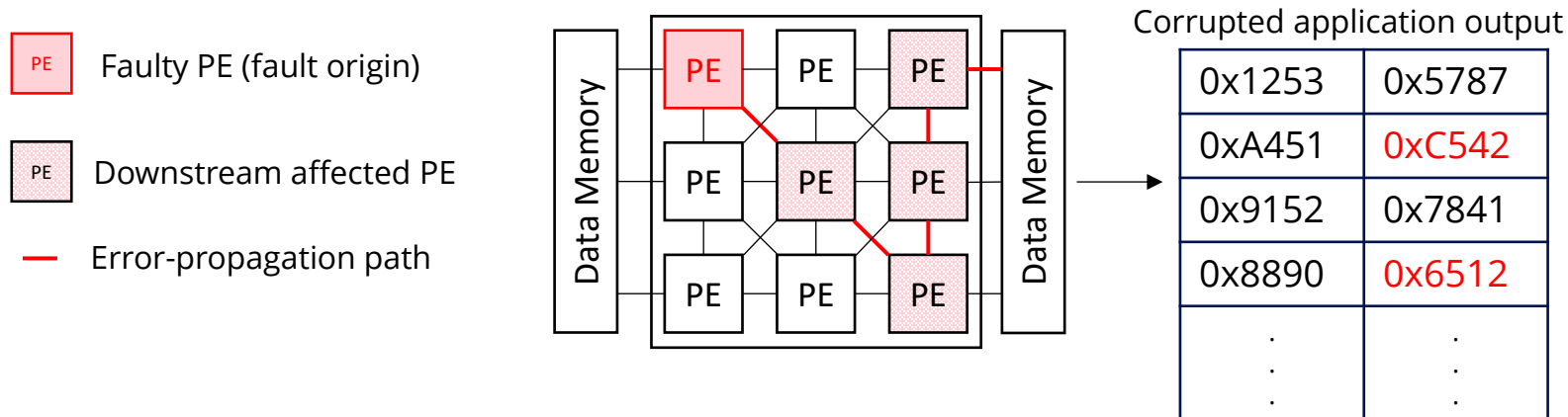
37th IEEE International Conference on Application-specific Systems, Architectures and Processors

September 2-4 2026, Imperial College London, London, UK

Ensieh Aliagha, Vijaya Shangamesh Manoharan Amutha, Diana Göhringer



- Technology scaling increases vulnerability to transient and permanent faults [1]
- A faulty PE or interconnect can corrupt the final output [2]
- The regular CGRA structure enables efficient spatial redundancy and local recovery



[1] S. M. Jafri et al., "Private reliability environments for efficient fault-tolerance in CGRAs", in Springer Design Automation for Embedded Systems 2014.

[2] G. Lee et al., 2017. "Fault Recovery Time Analysis for Coarse-Grained Reconfigurable Architectures", in ACM TECS 2018.



Related Work

Motivation and related work

FT-CGRA

Evaluation

Summary

Redundancy and masking

Examples:

- Selective TMR [2]
 - SoftwareTMR [3]
 - PE-level TMR with live FPGA fault injection [4]
 - M2STaR: spatial/temporal redundancy [1]
-
- **High transient fault coverage**
 - **Replication and voting overhead**
 - **Voter vulnerability in conventional TMR**

Detection and coding

Examples:

- PRES: low-cost PE-level error detection [5]
 - Parity and ECC (Memories) [3]
-
- **Low-cost and efficient detection**
 - **Recovery mechanism still required**

Remapping and reconfiguration

Examples:

- Hot-swapping (spare-based) [7]
 - Spatial-temporal remapping [6]
 - M2STaR: voter/branch reconfiguration with precompiled patterns [1]
-
- **Bypass permanently faulty resources**
 - **Restore functionality without full replication**
 - **Remapping/transformation latency**

Research gap

Prior works often rely on **replication**, **voting**, or runtime **reconfiguration**, with **high overhead** or **limited scalability**

Proposed FT-CGRA

- **Lightweight PE-level error detection (mod3/parity)**
- **Fast local spare handover**
- **Support both transient and permanent PE faults with lower cost**

[1] X. Kong et al., "M2STaR: A Multimode Spatio-Temporal Redundancy Design for Fault-Tolerant Coarse-Grained Reconfigurable Architectures," in IEEE TCAD 2023.

[2] T. Imagawa et al., "A cost-effective selective TMR for heterogeneous coarse-grained reconfigurable architectures based on DFG-level vulnerability analysis," in IEEE DATE 2013.

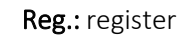
[3] G. Lee et al., 2017. "Fault Recovery Time Analysis for Coarse-Grained Reconfigurable Architectures", in ACM TECS 2018.

[4] J. M. Kühn et al., "Testing reliability techniques for SoCs with fault tolerant CGRA by using live FPGA fault injection," in IEEE FPT 2013.

[5] S. M. Jafri et al., "Private reliability environments for efficient fault-tolerance in CGRAs ", in Springer Design Automation for Embedded Systems 2014.

[6] S. Eisenhardt et al., "Spatial and Temporal Data Path Remapping for Fault-Tolerant Coarse-Grained Reconfigurable Architectures," in IEEE DFTS 2011.

[7] Z. E. Rákossy et al., "Hot-swapping architecture with back-biased testing for mitigation of permanent faults in functional unit array," in IEEE DATE 2013.





- Lightweight in-PE error detection
 - **Parity checking:** addition and subtraction
 - **Mod3 residue checking:** multiplication
 - Detects errors caused by transient and permanent faults
- Fault classification through re-execution
 - Distinguishes transient from permanent faults using lightweight re-execution
 - **Successful retry** → Transient fault
 - **Repeated error** → Permanent fault
- Compiler-transparent recovery
 - **Transient fault:** retry the operation
 - **Permanent fault:** hand over execution to a neighboring spare PE
 - Neighbor aliasing avoids remapping and recompilation



Why One Spare per 2×2 PE Array?

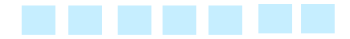
Motivation and related work



FT-CGRA



Evaluation

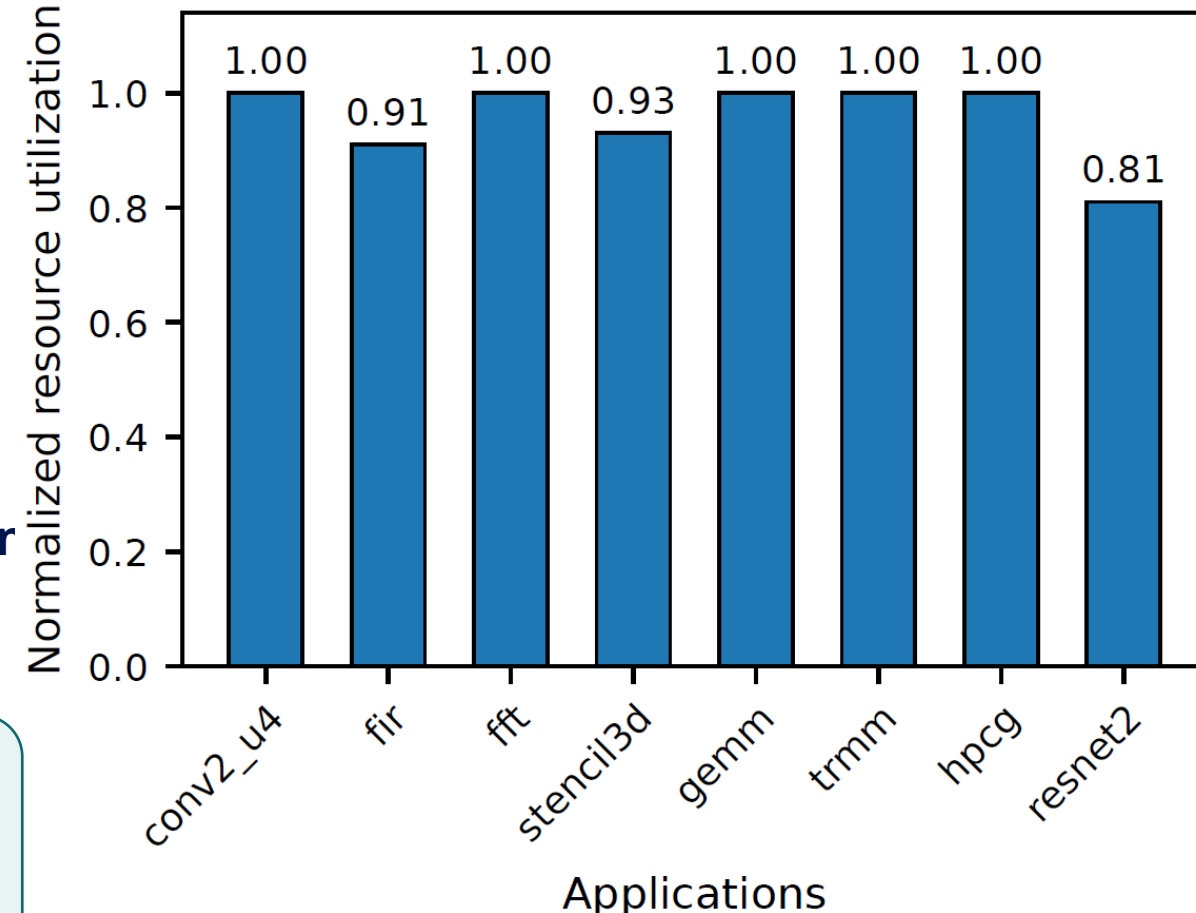


Summary



- **Baseline:** 4×4 CGRA without spare PEs
- **FT-CGRA:** 4×4 CGRA with 4 spares in total (One per 2×2 PE array)
- Utilization normalized to the baseline CGRA
- **Five of eight applications retain full utilization**
- Worst-case normalized utilization: **0.81**

Four local spares enable fault recovery with **limited utilization loss** for most applications



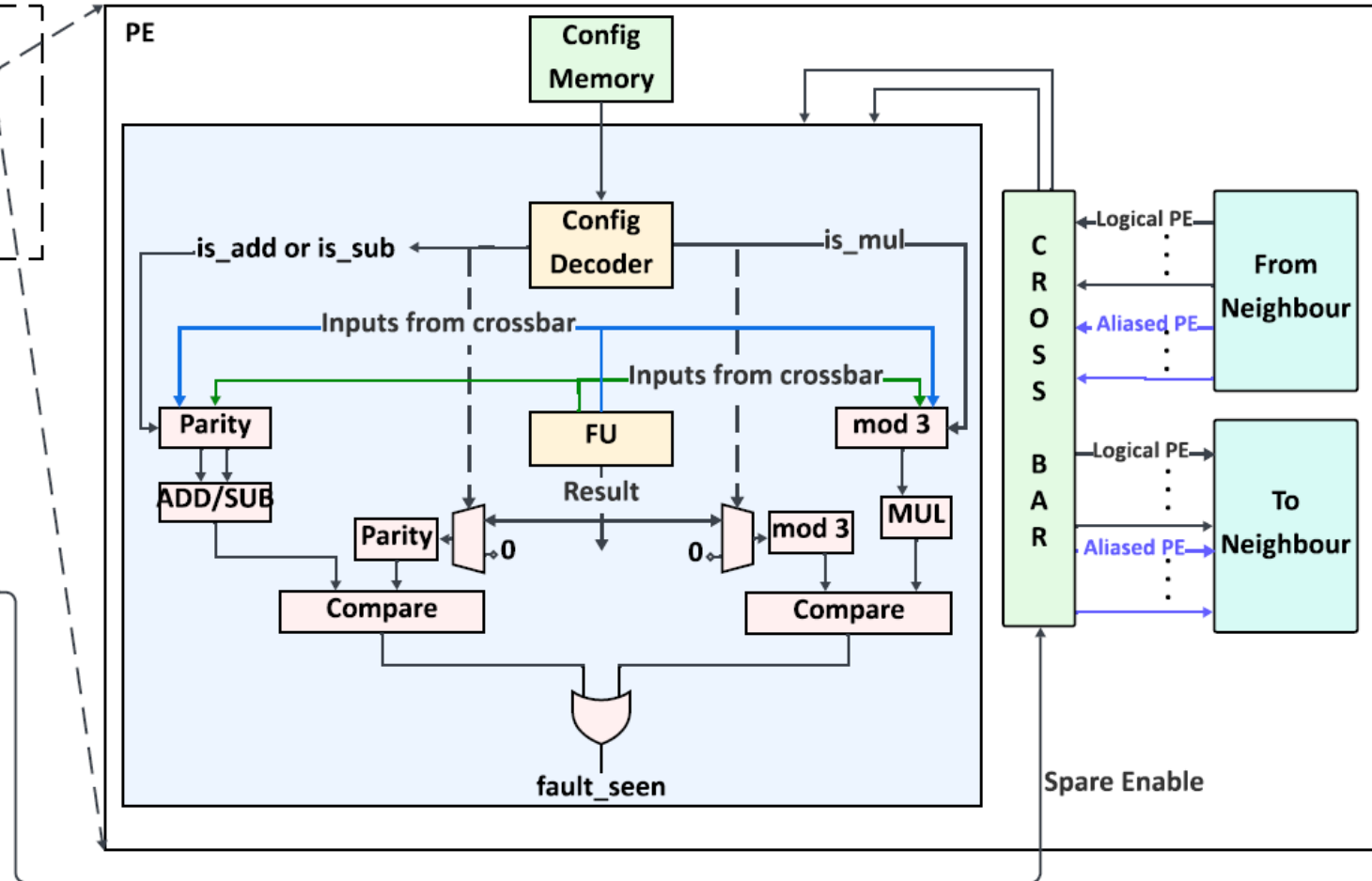
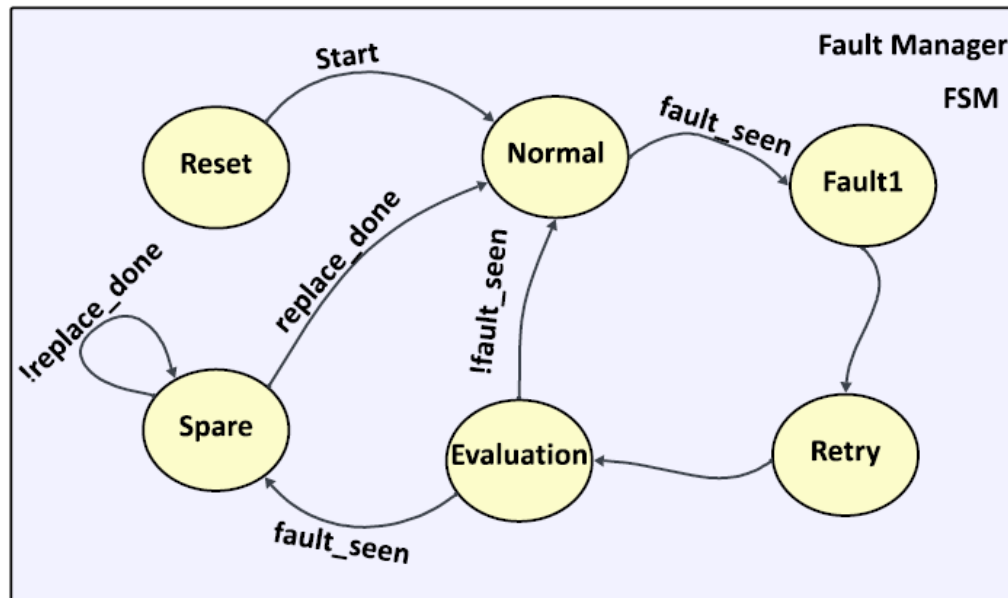
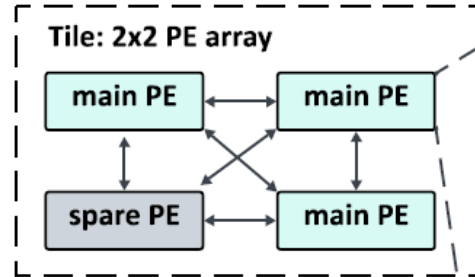
FT-CGRA architecture

Motivation and related work

FT-CGRA

Evaluation

Summary



Analytical Reliability Comparison

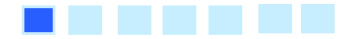
Motivation and related work



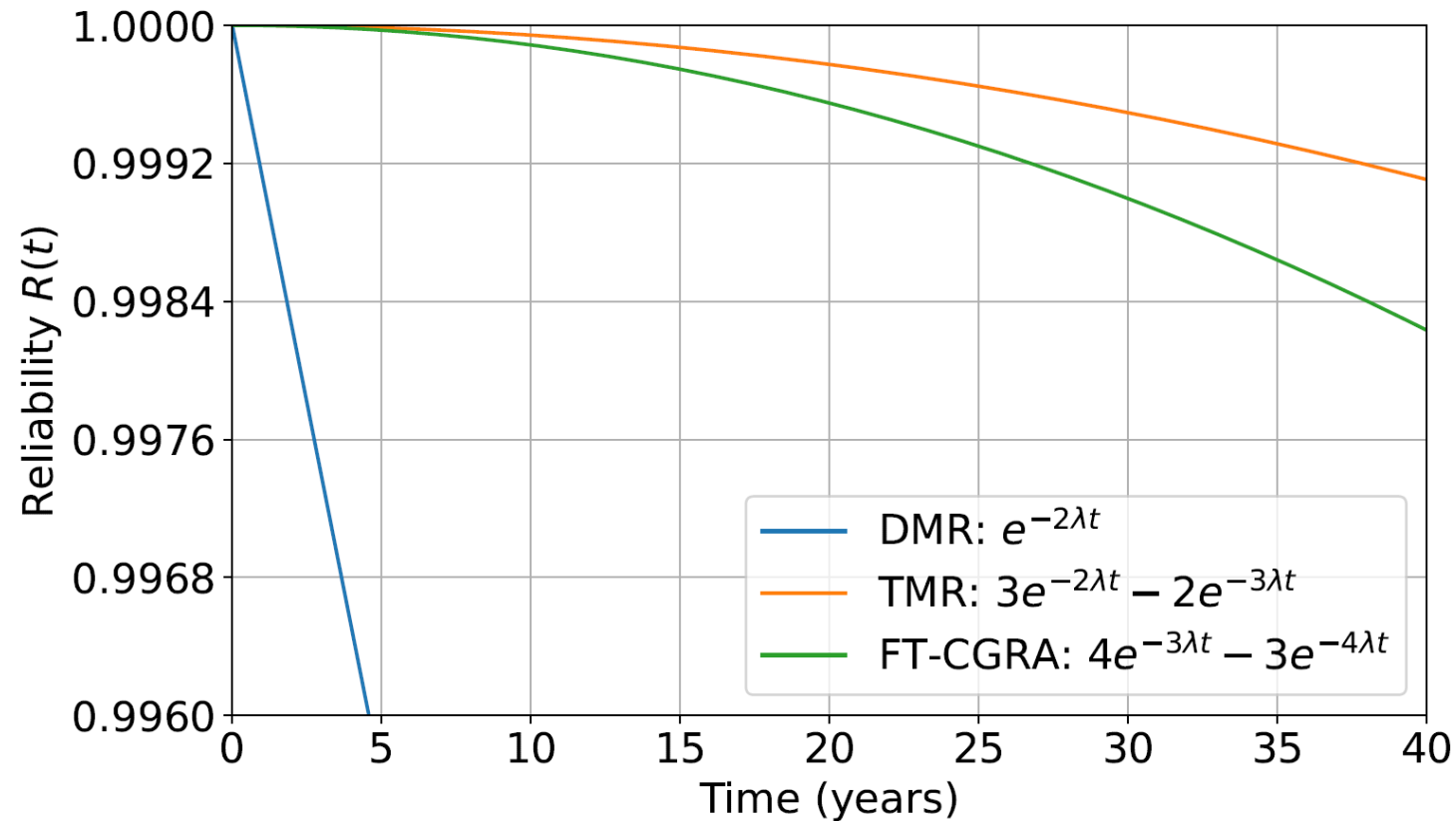
FT-CGRA



Evaluation



Summary



Assumption: $\lambda = 50 \text{ FIT} = 50 \times 10^{-9} h^{-1}$, $1 \text{ FIT} = \frac{1 \text{ failure}}{10^9 \text{ device-hours}}$



Markov Reliability Evaluation

Motivation and related work



FT-CGRA



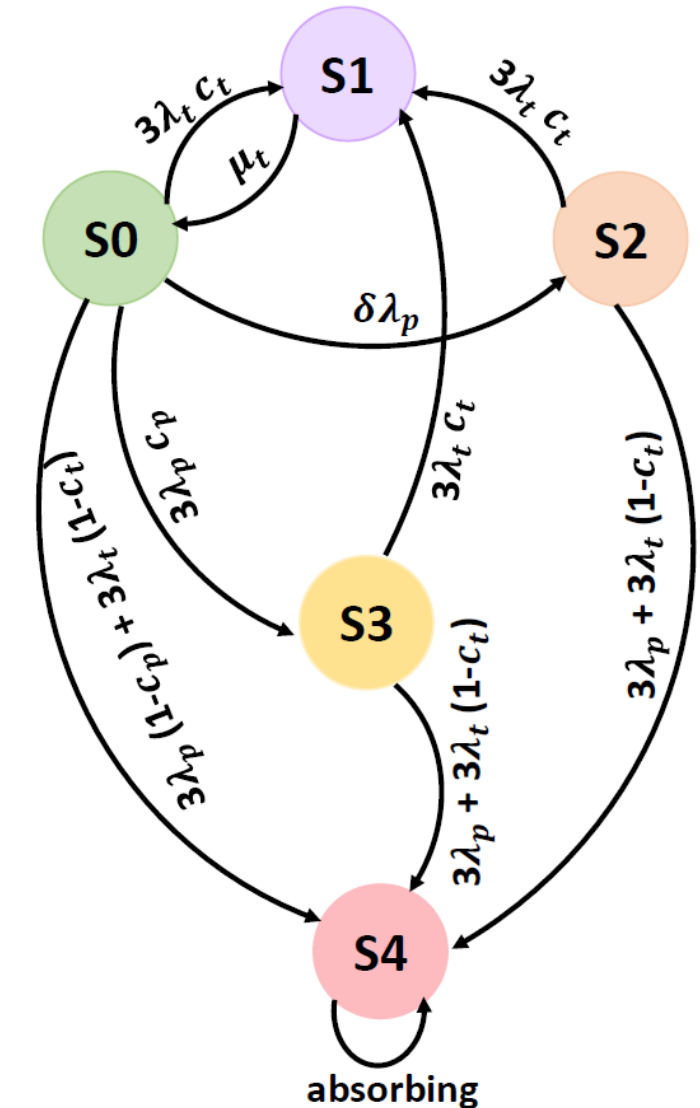
Evaluation



Summary



- **S0**: Healthy, spare available
- **S1**: Transient fault recovery is in-progress
- **S2**: Active PEs healthy, spare unavailable
- **S3**: Operational after one permanent PE fault; spare activated
- **S4**: Failed



λ_p Permanent failure rate of one active PE

λ_t Transient fault rate of one active PE

δ Dormancy factor of the spare PE

c_p Permanent fault coverage

c_t Transient fault coverage

μ_t Transient fault recovery rate



Comparison to M2STaR

Motivation and related work



FT-CGRA



Evaluation



Summary



- M2STaR [1]: multimode spatio-temporal redundancy for fault tolerant CGRAs
- Combines redundancy with voter and branch reconfiguration
- Uses precompiled patterns for fast runtime configuration transformation
- Reports up to 93.84% fault coverage and $1.31 \times$ MTTF versus spatial TMR

c_p	$\delta = 0$	$\delta = 0.01$	$\delta = 0.05$	$\delta = 0.1$
0.9	0.582	0.580	0.576	0.571
0.95	0.596	0.595	0.591	0.585
0.98	0.604	0.603	0.600	0.595
1	0.611	0.610	0.605	0.601

Sensitivity of normalized FT-CGRA mean time to failure (MTTF) relative to M2STaR best solution
M2STaR best solution = 1.0; higher is better

[1] X. Kong et al., "M2STaR: A Multimode Spatio-Temporal Redundancy Design for Fault-Tolerant Coarse-Grained Reconfigurable Architectures," in IEEE TCAD 2023.



MTTF Improvement

Motivation and related work



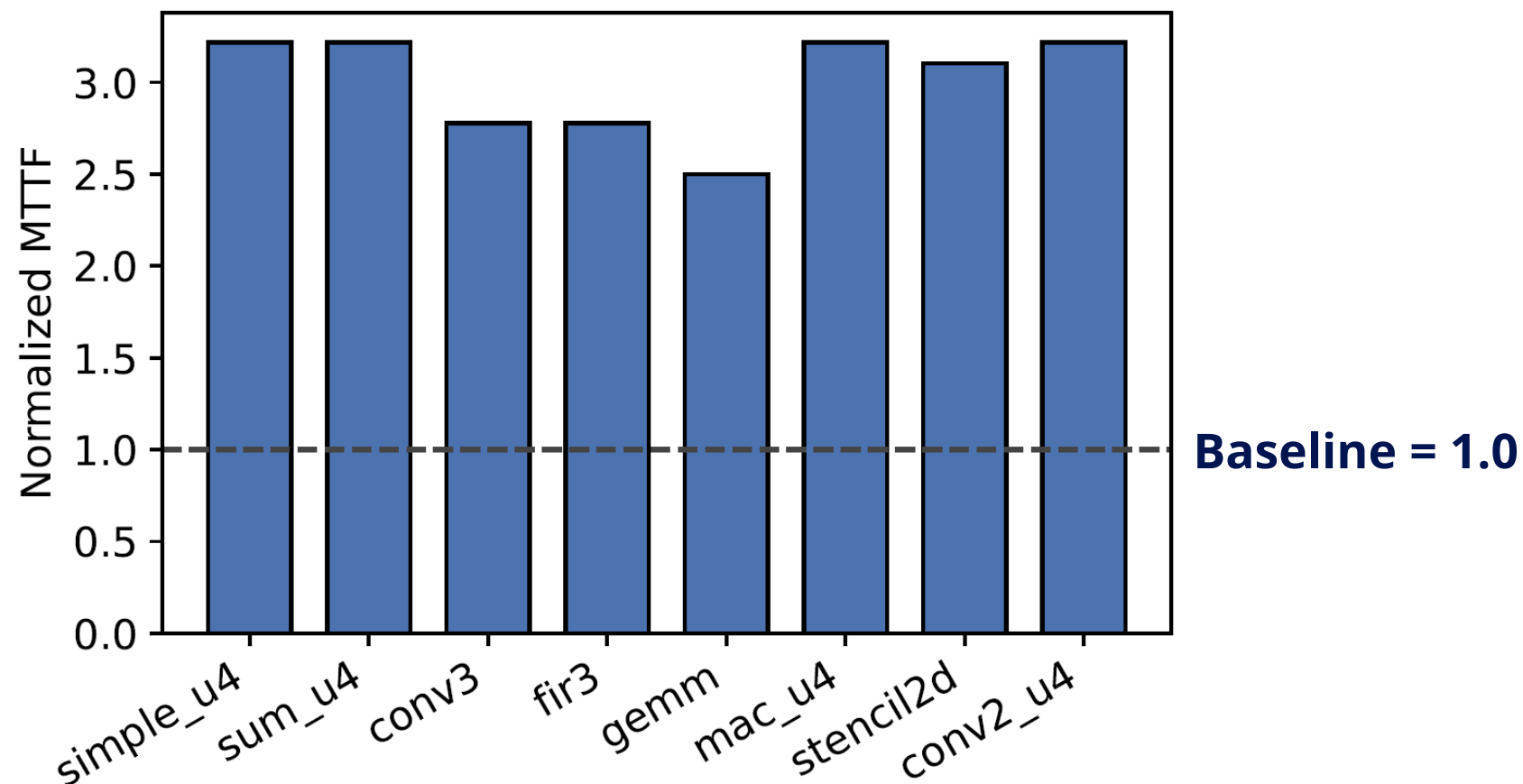
FT-CGRA



Evaluation



Summary



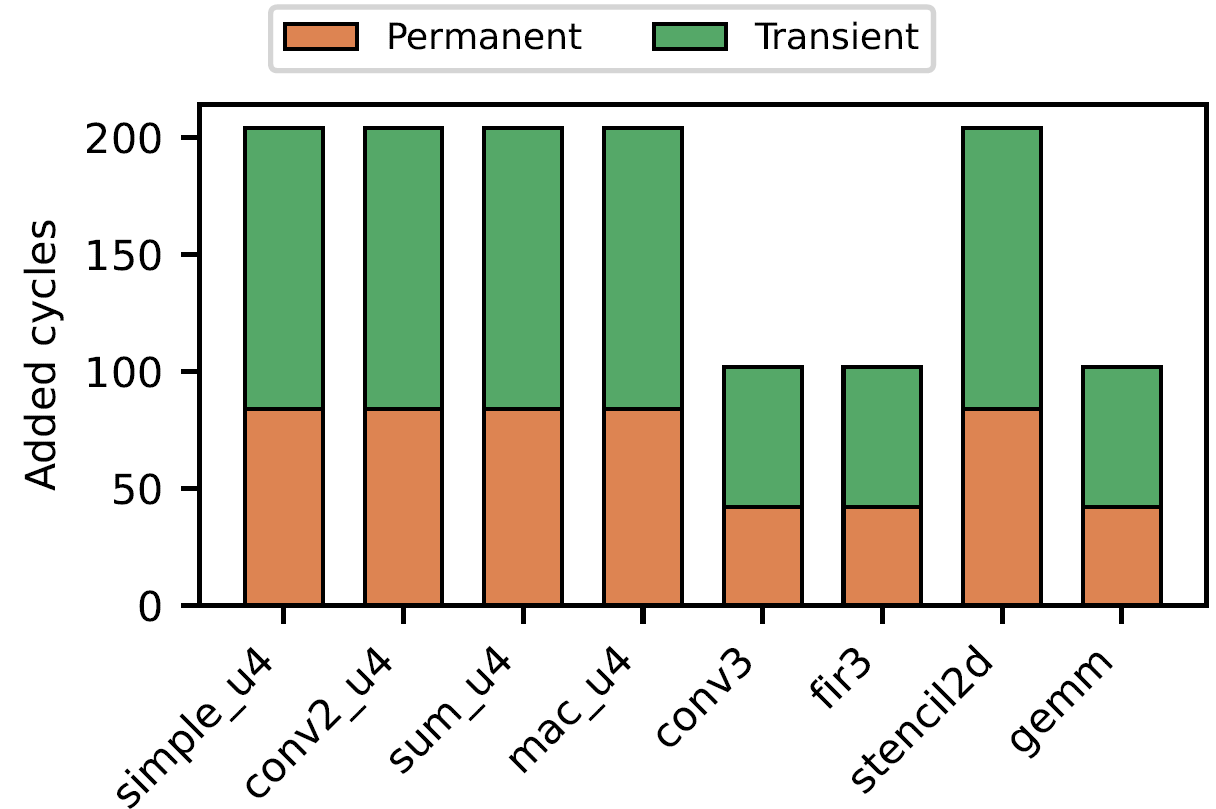
Normalized MTTF of FT-CGRA under the PE-only permanent fault model



Recovery Overhead

- **Execution length:** 100,000 cycles
- Injected permanent:transient fault ratio = 1:10

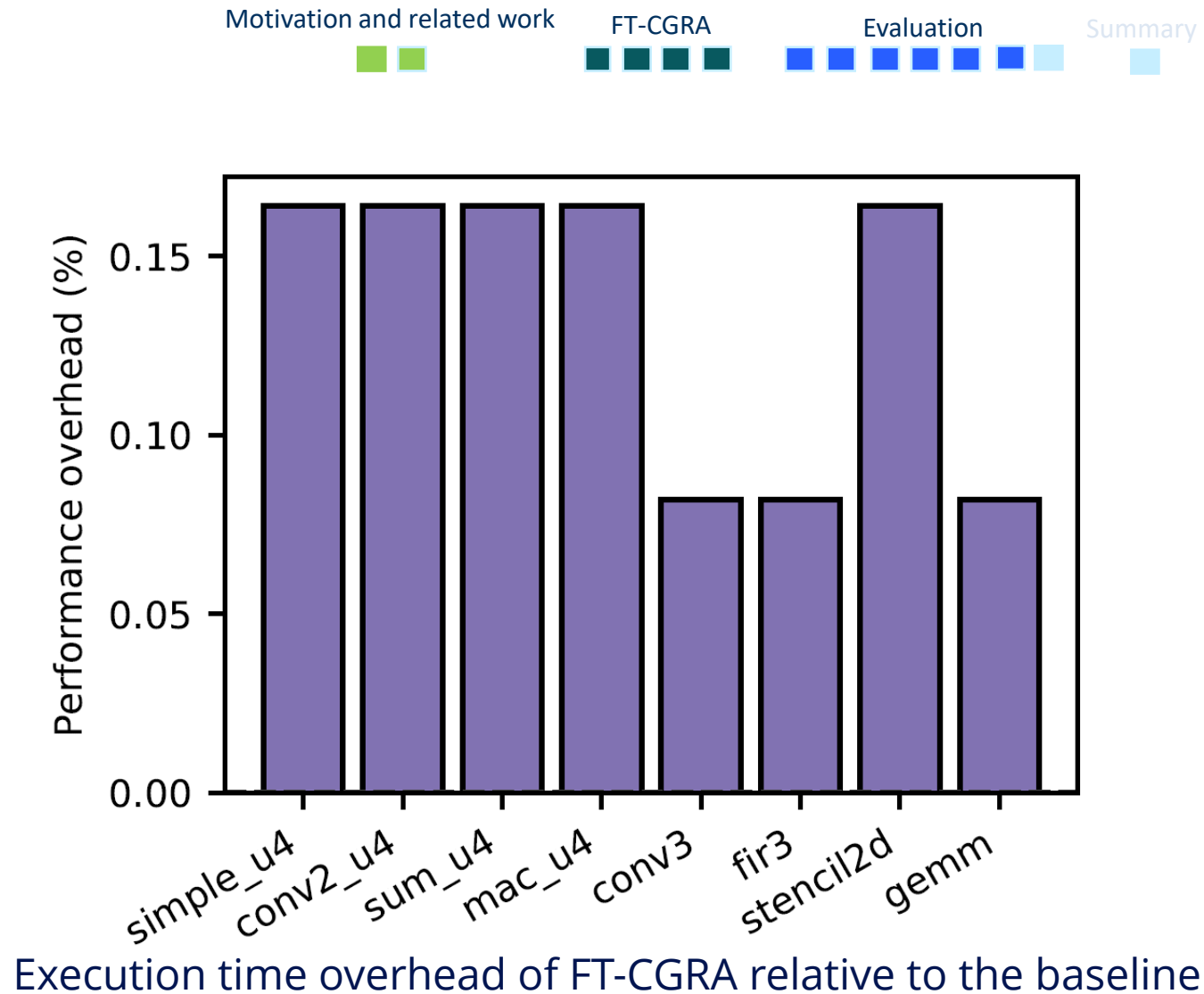
Recovery overhead remains **below**
~0.21% of total execution cycles



Additional execution cycles due to transient re-execution and permanent fault recovery

Performance Overhead

Execution time overhead remains **below**
0.17% across all benchmarks.



Power Consumption Overhead

Motivation and related work



FT-CGRA



Evaluation



Summary



- Power normalized to the baseline FU
- **FT-CGRA computation logic:** $1.22\times$ baseline $\rightarrow$ 22% overhead
- **FU with mod3 for all operations:** $1.74\times$ baseline $\rightarrow$ 74% overhead
- FT-CGRA reduces the power overhead from 74% to 22%
- **Fault Manager:** $0.20\times$ baseline FU power, reported separately

Metric	FT-CGRA	Base FU	FU + mod3	Fault manager
Normalized power	1.22	1.00	1.74	0.2





- **Lightweight fault tolerance for CGRAs**
 - In-PE detection using parity and mod3 residue checking
- **Unified transient and permanent fault handling**
 - Retry for transient faults; local spare handover for permanent faults
- **Compiler-transparent recovery**
 - Neighbor aliasing avoids remapping and recompilation
- **Low execution-time overhead:** $<0.2\%$ across all evaluated benchmarks
- **Significant reliability improvement:** $2.5\text{--}3.2\times$ the baseline MTTF under the PE-only permanent fault model

FT-CGRA improves reliability **through lightweight local detection and recovery**, without full replication or application remapping.



Thank You!

Questions & Discussion

