

Assessing Triple Modular Redundancy for Wide-Link, Low-Latency NoC Routers

Reliability and Physical Design Challenges

IEEE DFTS 2026

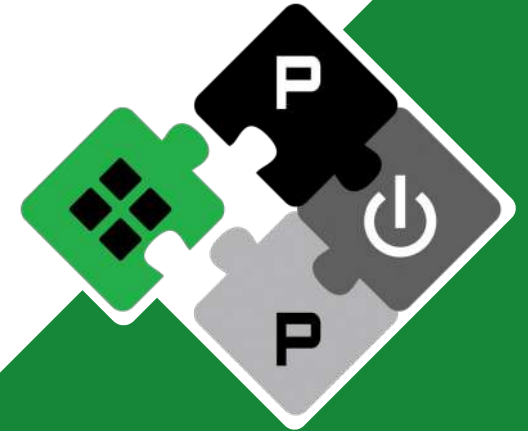
Chen Wu chenwu@iis.ee.ethz.ch

Michael Rogenmoser, Luca Benini, Angelo Garofalo

ETH Zürich · University of Bologna

PULP Platform

Open Source Hardware, the way it should be!



pulp-platform.org

@pulp_platform

company/pulp-

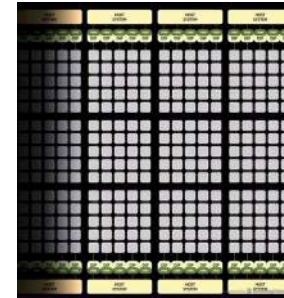
youtube.com/pulp_platform



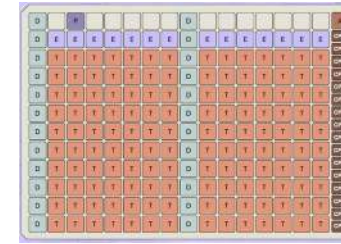
Physical-AI tiles are heading into harsh environments



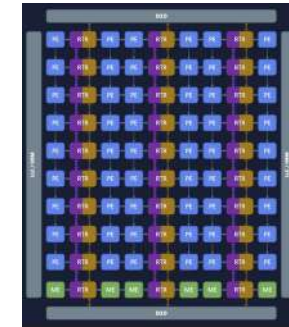
- **Tile-based many-PE accelerators are the template for Physical AI**
 - compute + local memory + NoC router + NI per tile, replicated in a 2D mesh
- **Deployed in radiation-heavy environments**
 - space, avionics, safety-critical systems
 - single-event upsets and transients are unavoidable; continuous correct operation is required
- **The NoC is the backbone and a single point of failure**
 - one uncorrected fault can corrupt data, deadlock communication, or hang the entire system



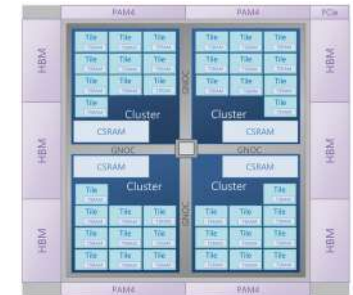
Tesla Dojo (2022)



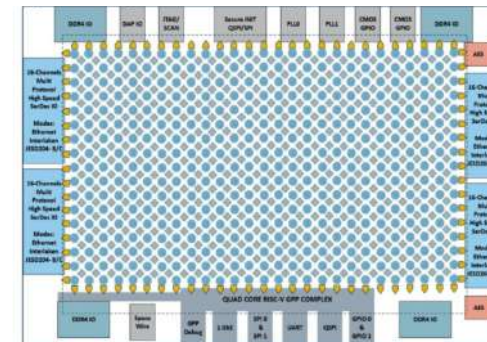
Tenstorrent Blackhole (2024)



Meta MTIA 400 (2026)



Microsoft Maia 200 (2026)



HyperX Midnight SoC (2023)



Why existing reliable-NoC work does not transfer



Prior fault-tolerant NoCs

- **Built for permanent faults^[1]**
 - aging and manufacturing defects, not radiation-induced transients
- **Radiation-aware work still has gaps^[2]**
 - Combinational logic exposed to SETs
 - No self-recovery, faults accumulate
- **Small fault-injection campaigns**
- **No full backend evaluation**

NoC routers have changed

- **Ultra-wide physical links (e.g. 512-bit)**
 - bulk data transfers at high throughput
- **Low latency: no deep pipelines**
- **High operating frequencies (1 GHz and beyond)**
- **Advanced technology nodes**
 - abundant routing resources

Open question:

what does TMR cost, and what does it buy, on a wide-link, low-latency router?

[1] "Vicis: A reliable network for unreliable silicon," DAC2009

[2] "Reliability and performance evaluation of a fault- tolerant MPSoC interconnection architecture ." DFT2025



What we did



1

Three TMR granularities on the FlooNoC router

coarse, state-only and full TMR, inserted at RTL with the open-source TMRG tool

2

RTL-to-GDSII in TSMC 7 nm for all routers

frequency sweep, area, and the physical-design constraints that keep the redundancy alive

3

Fault injection at RTL and on the post-layout netlist

SEU, SET and pairs of SEUs; more than one million sampled faults per campaign

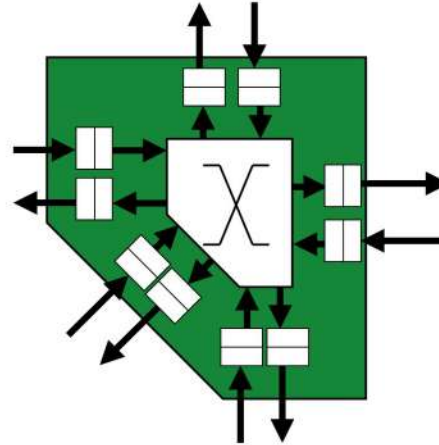
4

System-level evaluation in a Physical-AI compute tile

Snitch cluster + TMR router: area, power on GEMM, timing targeting at 1 GHz



Target: FlooNoC^[1] router inside a Snitch^[2] compute tile



FlooNoC router

Open source, AXI4 compatible

Low latency: 2 cycles per hop

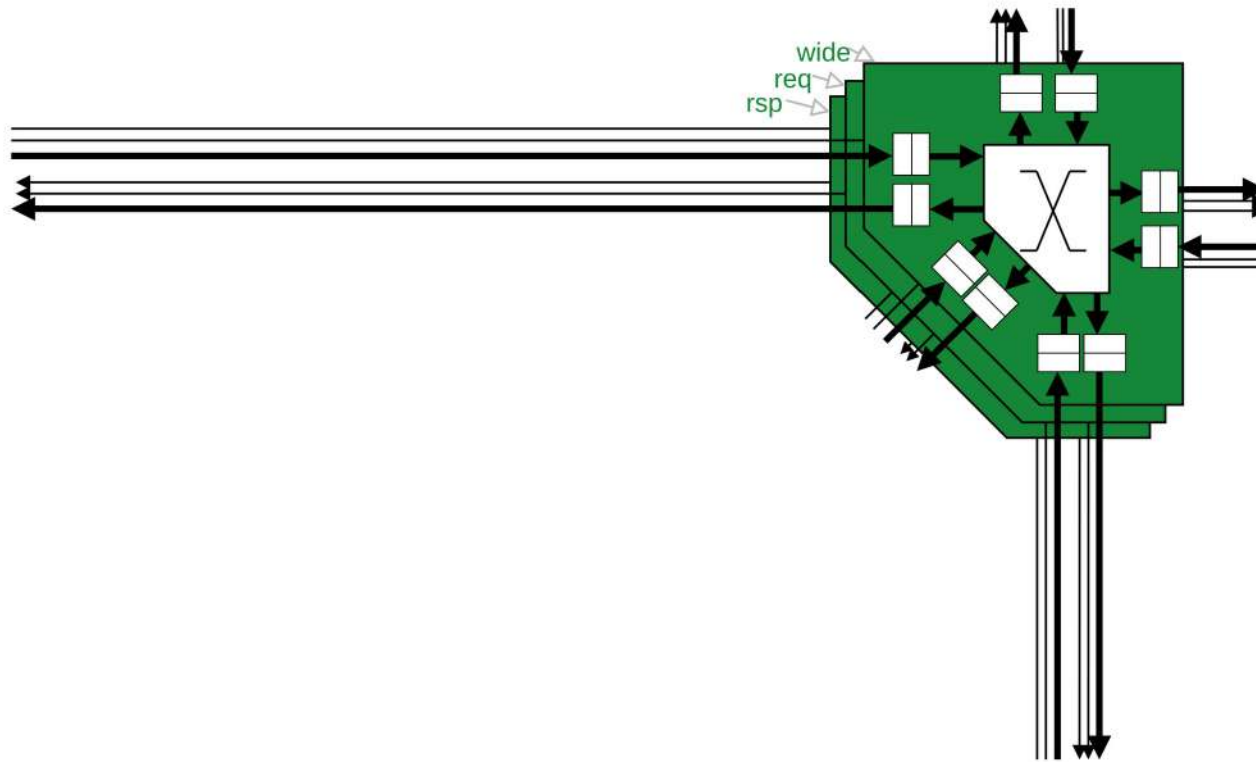
- no pipeline stage between input and output buffers
- 5 ports: N, E, S, W + local

[1] T. Fischer et al., "FlooNoC: A 645-Gb/s/link 0.15-pJ/B/hop open-source NoC with wide physical links and end-to-end AXI4 parallel multistream support," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 33, no. 4, pp. 1094–1107, 2025.

[2] Z. Florian, et al. "Snitch: A tiny pseudo dual-issue processor for area and energy efficient execution of floating-point intensive workloads." IEEE Transactions on Computers. pp.1845-1860, 2020.



Target: FlooNoC^[1] router inside a Snitch^[2] compute tile



3 physical links, 3 router networks

wide: 512-bit, DMA bursts

req, rsp: 64-bit narrow

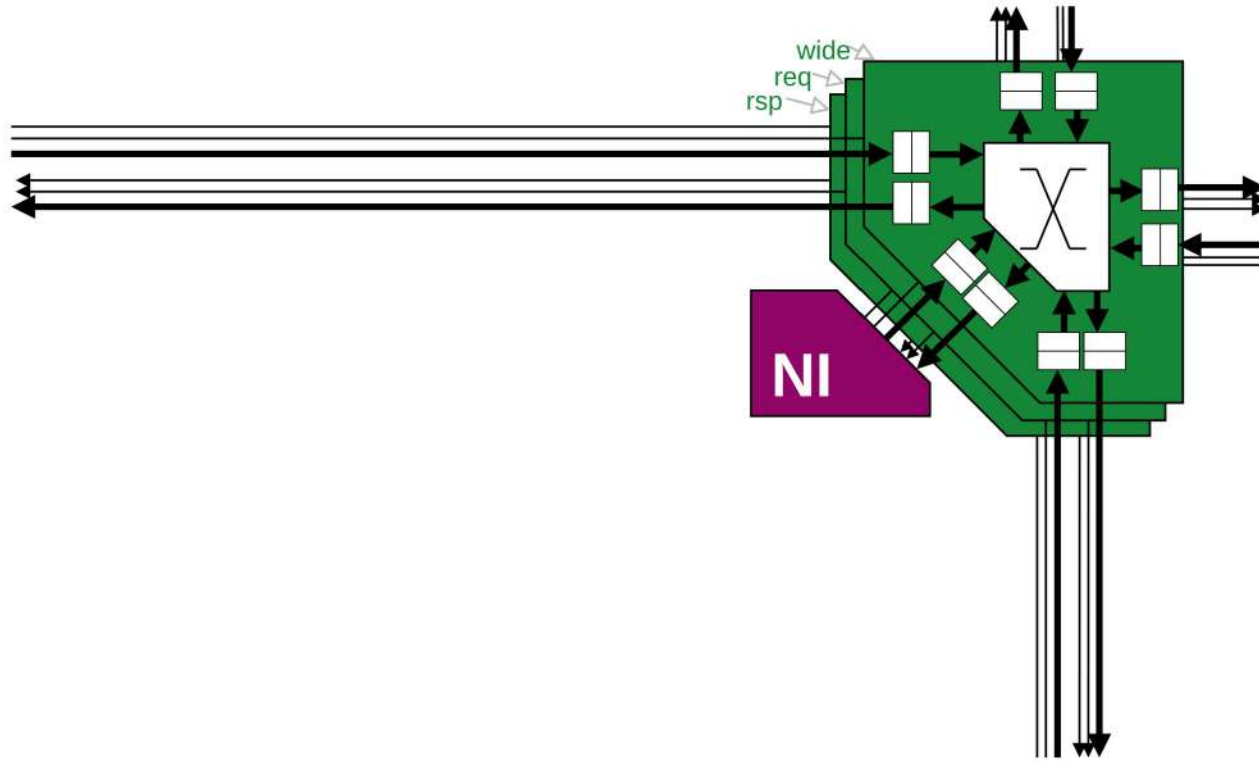
- each link has its own router network
- bursts never block narrow requests and responses

[1] T. Fischer et al., "FlooNoC: A 645-Gb/s/link 0.15-pJ/B/hop open-source NoC with wide physical links and end-to-end AXI4 parallel multistream support," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 33, no. 4, pp. 1094–1107, 2025.

[2] Z. Florian, et al. "Snitch: A tiny pseudo dual-issue processor for area and energy efficient execution of floating-point intensive workloads." IEEE Transactions on Computers. pp.1845-1860, 2020.



Target: FlooNoC^[1] router inside a Snitch^[2] compute tile



Network interface (NI)

Maps AXI4 transactions onto flits

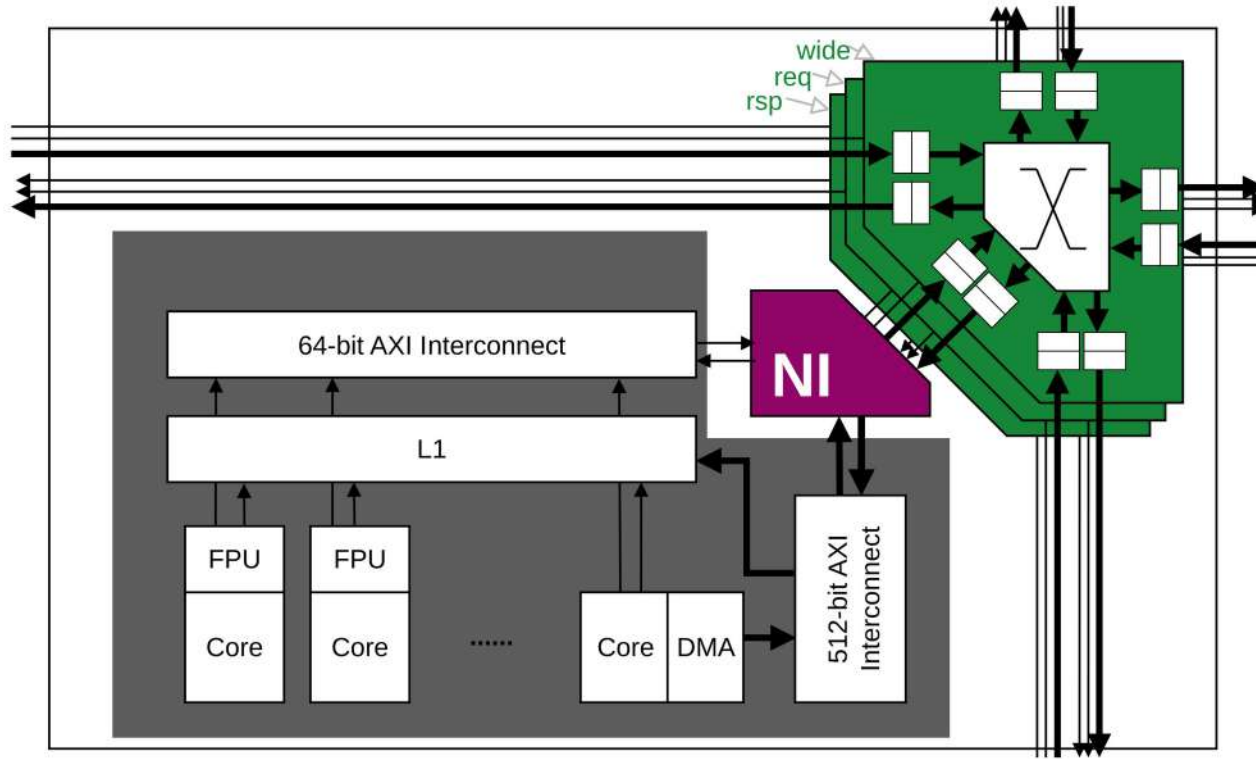
- bridges the tile's 64-bit and 512-bit AXI buses to the three links

[1] T. Fischer et al., "FlooNoC: A 645-Gb/s/link 0.15-pJ/B/hop open-source NoC with wide physical links and end-to-end AXI4 parallel multistream support," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 33, no. 4, pp. 1094–1107, 2025.

[2] Z. Florian, et al. "Snitch: A tiny pseudo dual-issue processor for area and energy efficient execution of floating-point intensive workloads." IEEE Transactions on Computers. pp.1845-1860, 2020.



Target: FlooNoC^[1] router inside a Snitch^[2] compute tile



Snitch cluster tile

8 RISC-V cores + FP units

128 KB L1 scratchpad, DMA core

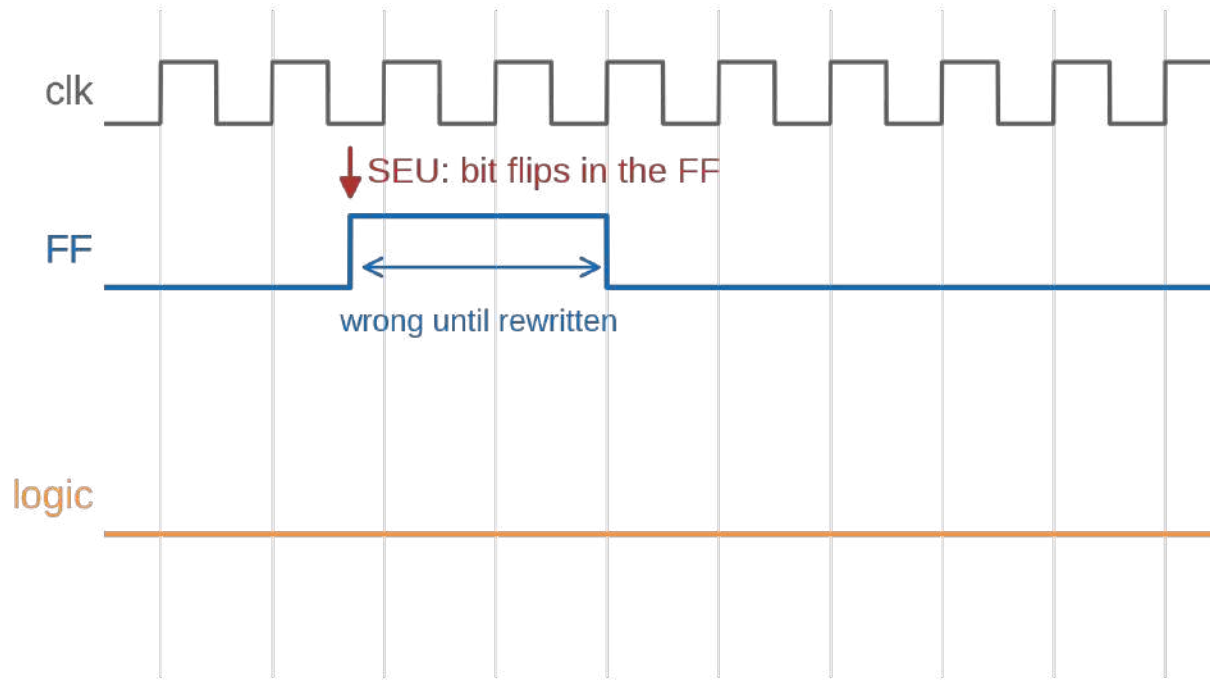
- 1 GHz target, worst-case corner

[1] T. Fischer et al., "FlooNoC: A 645-Gb/s/link 0.15-pJ/B/hop open-source NoC with wide physical links and end-to-end AXI4 parallel multistream support," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 33, no. 4, pp. 1094–1107, 2025.

[2] Z. Florian, et al. "Snitch: A tiny pseudo dual-issue processor for area and energy efficient execution of floating-point intensive workloads." IEEE Transactions on Computers. pp.1845-1860, 2020.



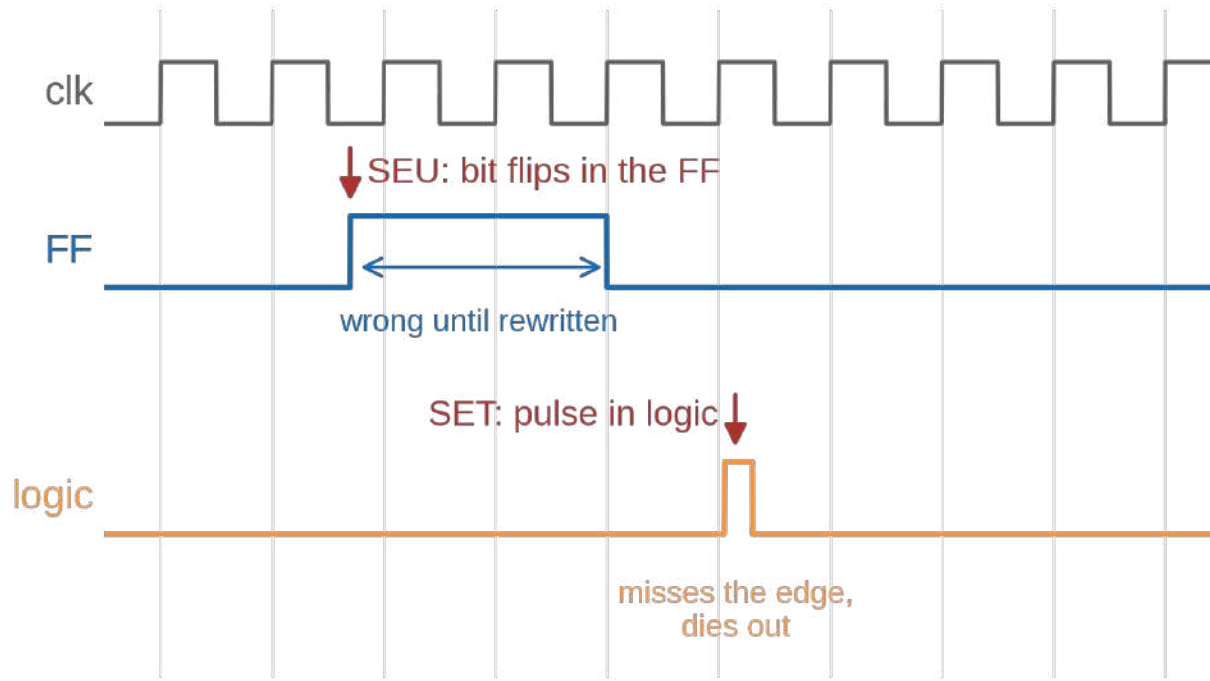
Fault model: single-event upsets



- **SEU**: bit flip in a flip-flop or SRAM cell
 - persists until the element is overwritten
 - uncorrected upsets can accumulate



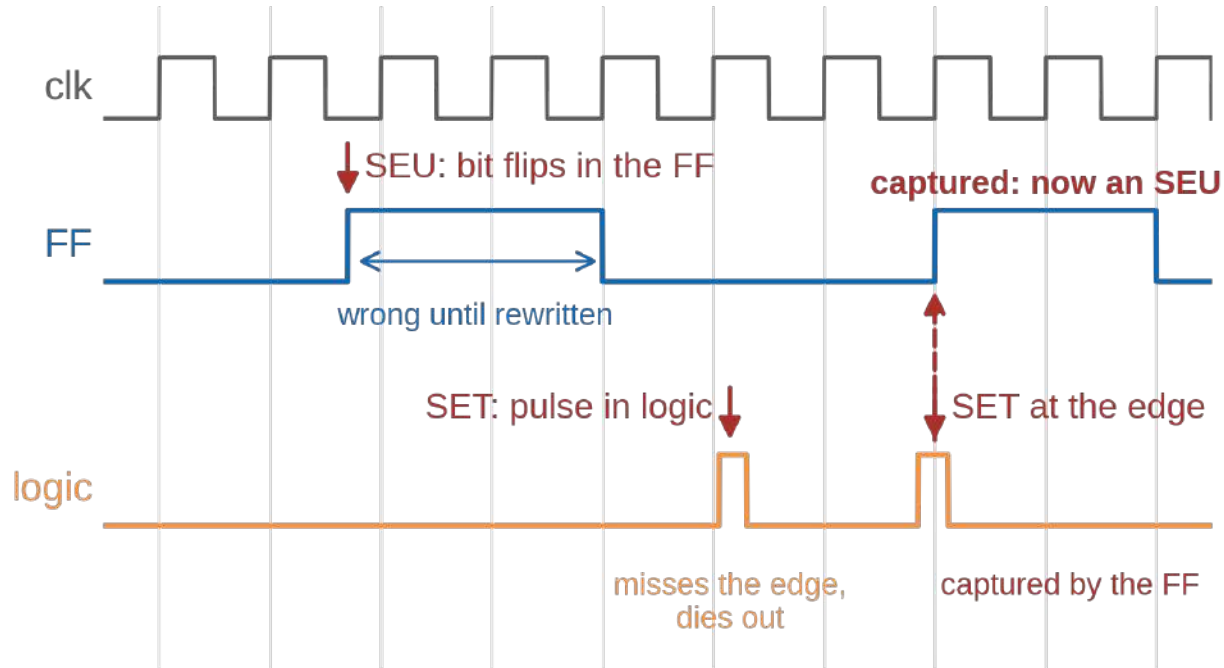
Fault model: single-event transients



- **SEU**: bit flip in a flip-flop or SRAM cell
 - persists until the element is overwritten
 - uncorrected upsets can accumulate
- **SET**: transient pulse in logic
 - most pulses die out before the next clock edge



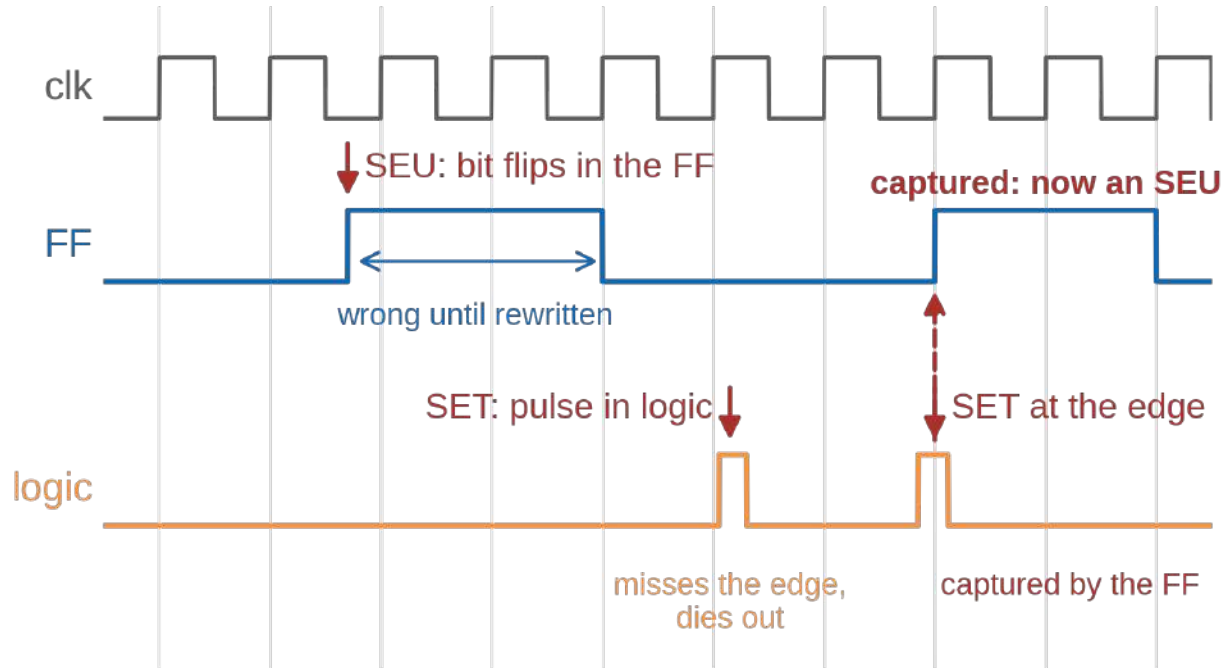
Fault model: a captured transient becomes an upset



- **SEU**: bit flip in a flip-flop or SRAM cell
 - persists until the element is overwritten
 - uncorrected upsets can accumulate
- **SET**: transient pulse in combinational logic
 - most pulses die out before the next clock edge
 - a pulse captured at the clock edge becomes an SEU
 - more likely at high frequency



Fault model: multiple upsets



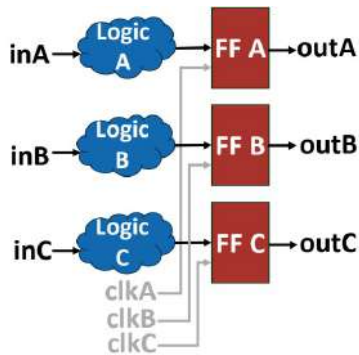
- **SEU:** bit flip in a flip-flop or SRAM cell
 - **persists** until the element is overwritten
 - uncorrected upsets can accumulate
- **SET:** transient pulse in combinational logic
 - most pulses die out before the next clock edge
 - a pulse **captured** at the clock edge becomes an SEU
 - more likely at high frequency
- **Multiple upsets**
 - **same clock cycle:** one strike flips several neighbouring FFs
 - **random cycles:** independent strikes accumulate over time



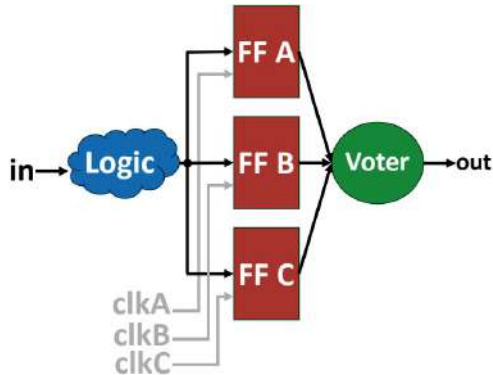
Three TMR granularities on the same router



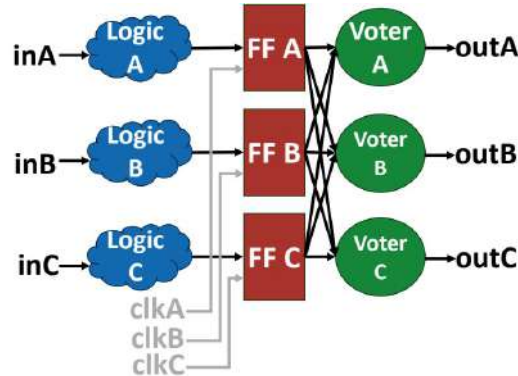
Coarse TMR



State-only TMR



Full TMR



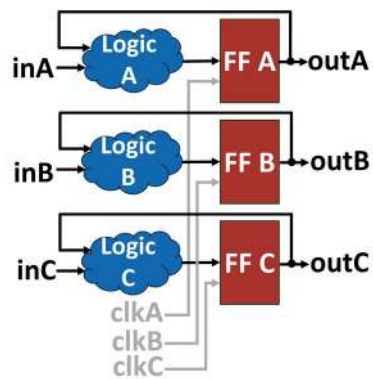
	Coarse	State-only	Full
Triplicated	whole router	flip-flops only	flip-flops + logic
Voting	at destination NI	per FF, inside	per FF + at NI



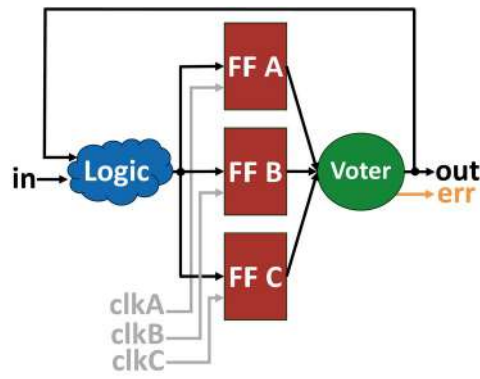
Three TMR granularities on the same router



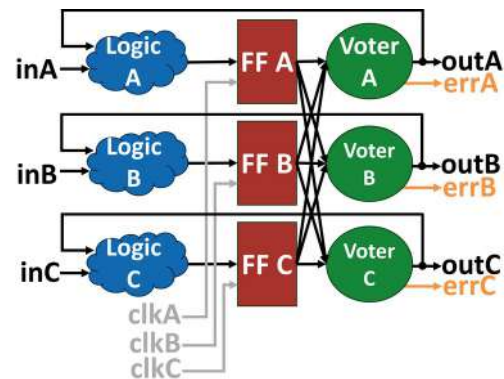
Coarse TMR



State-only TMR



Full TMR



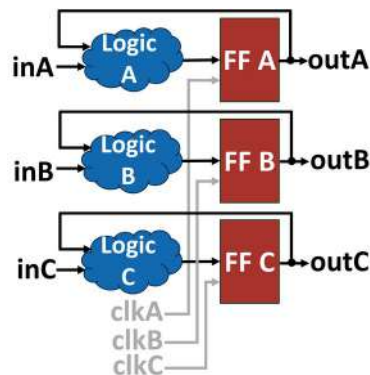
	Coarse	State-only	Full
Triplicated	whole router	flip-flops only	flip-flops + logic
Voting	at destination NI	per FF, inside	per FF + at NI
Self-repair	none	every cycle	every cycle



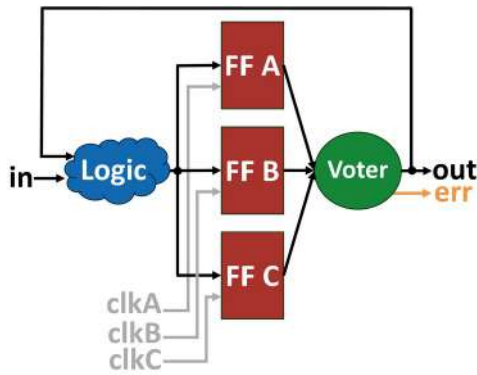
Three TMR granularities on the same router



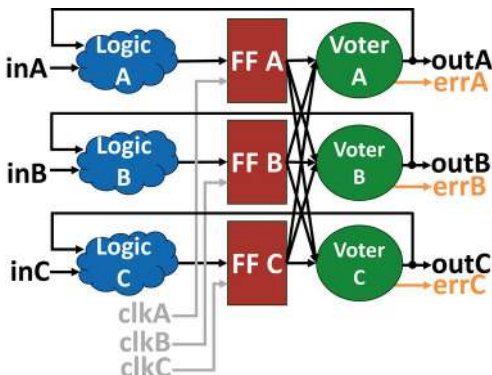
Coarse TMR



State-only TMR



Full TMR



	Coarse	State-only	Full
Triplicated	whole router	flip-flops only	flip-flops + logic
Voting	at destination NI	per FF, inside	per FF + at NI
Self-repair	none	every cycle	every cycle
Covers	SEU + SET, single faults only	SEU, no SET	SEU + SET



Implementation



- **Automatic insertion with TMRG^[1]**
 - open-source, RTL-level insertion
- **Independent clock and reset per replica**
 - three clk / rst ports per router: a glitch on one cannot hit all replicas
- **Aggregated error output**
 - all voter mismatches ORed into one bit: the system counts corrected upsets
- **Border voters (coarse, full) sit in the NI**
 - $\approx 1/6$ of the router area, not evaluated here

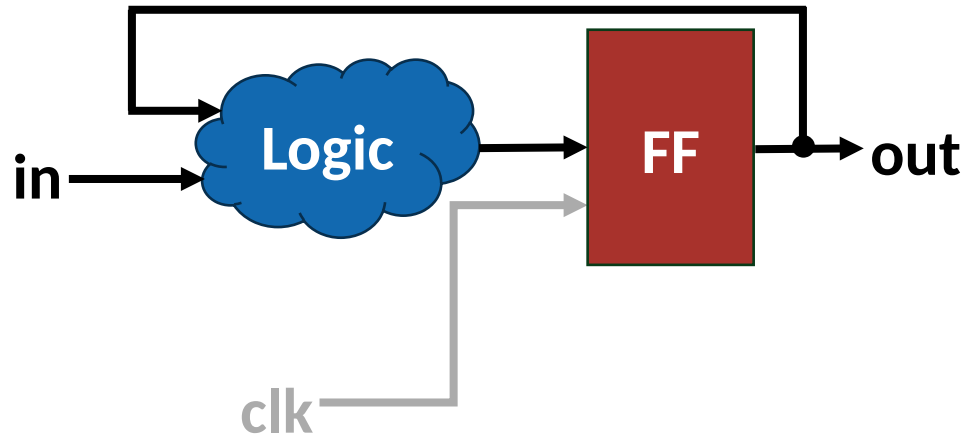
[1] S. Kulis, “Single event effects mitigation with TMRG tool,” Journal of Instrumentation, vol. 12, no. 01, p. C01082, 2017.



Implementation – Feedback path



- TMR with per-FF voting
 - Uncorrected states: stay and even **propagate**

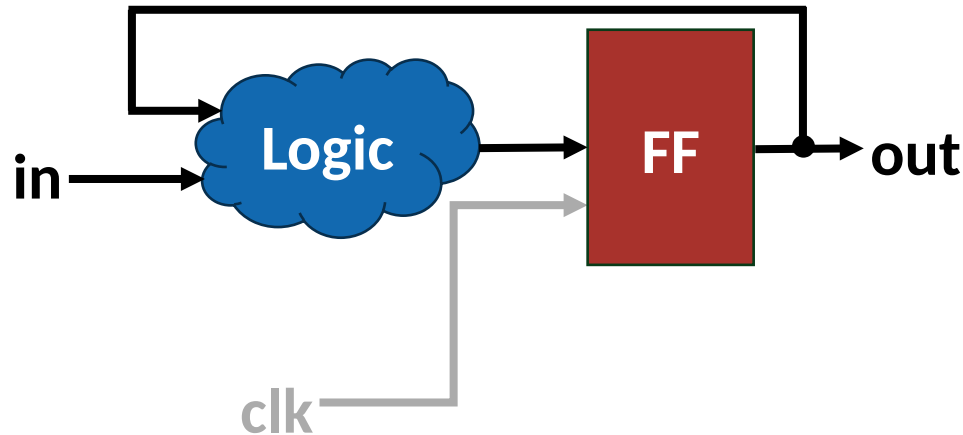


Implementation – Feedback path

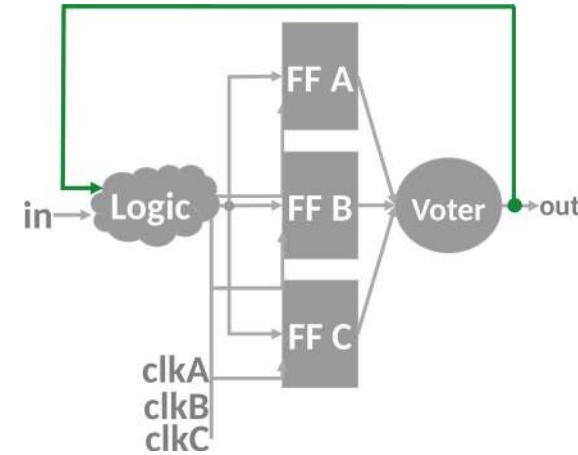


- **TMR with per-FF voting**

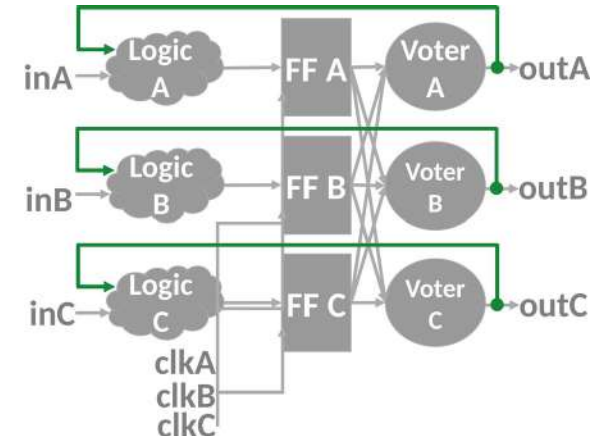
- Uncorrected states: stay and even **propagate**
- To correct: voted value re-enters the logic through a **feedback path** and internal states can self-recover



State-only TMR green = voter output fed back to the logic



Full TMR same feedback, once per replica



Implementation – Feedback path



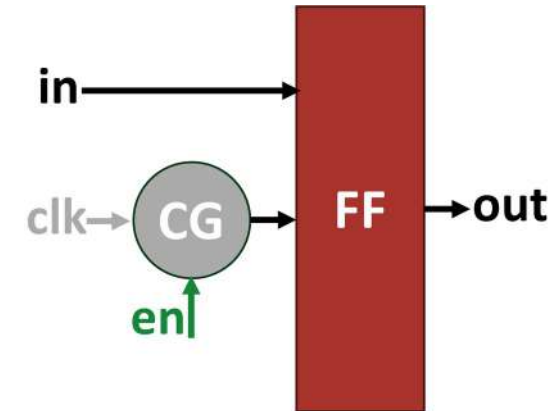
- **TMR with per-FF voting**

- Uncorrected states: stay and even **propagate**
- To correct: voted value re-enters the logic through a **feedback path** and internal states can self-recover

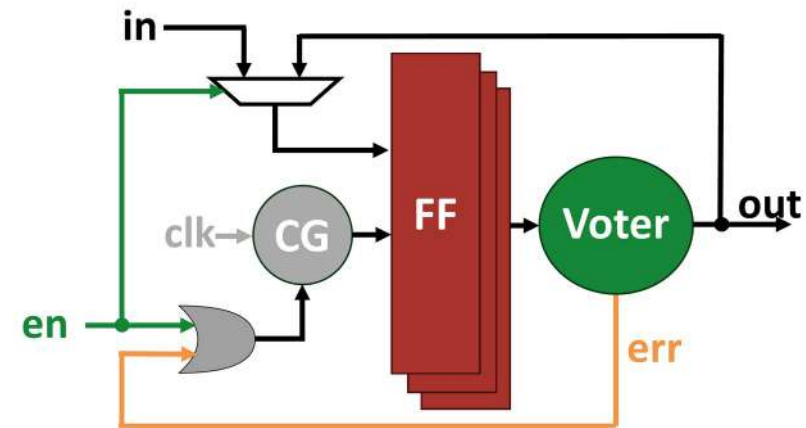
- **Clock-gated FFs**

- When gated: unable to recover
- **Special feedback path** (with **error** signal): A gated, corrupted FF is ensured to recover in time to avoid accumulation

Baseline: a gated FF just holds its value



TMR: the voter's error re-enables the gated clock



Physical design: setup and constraints



Setup

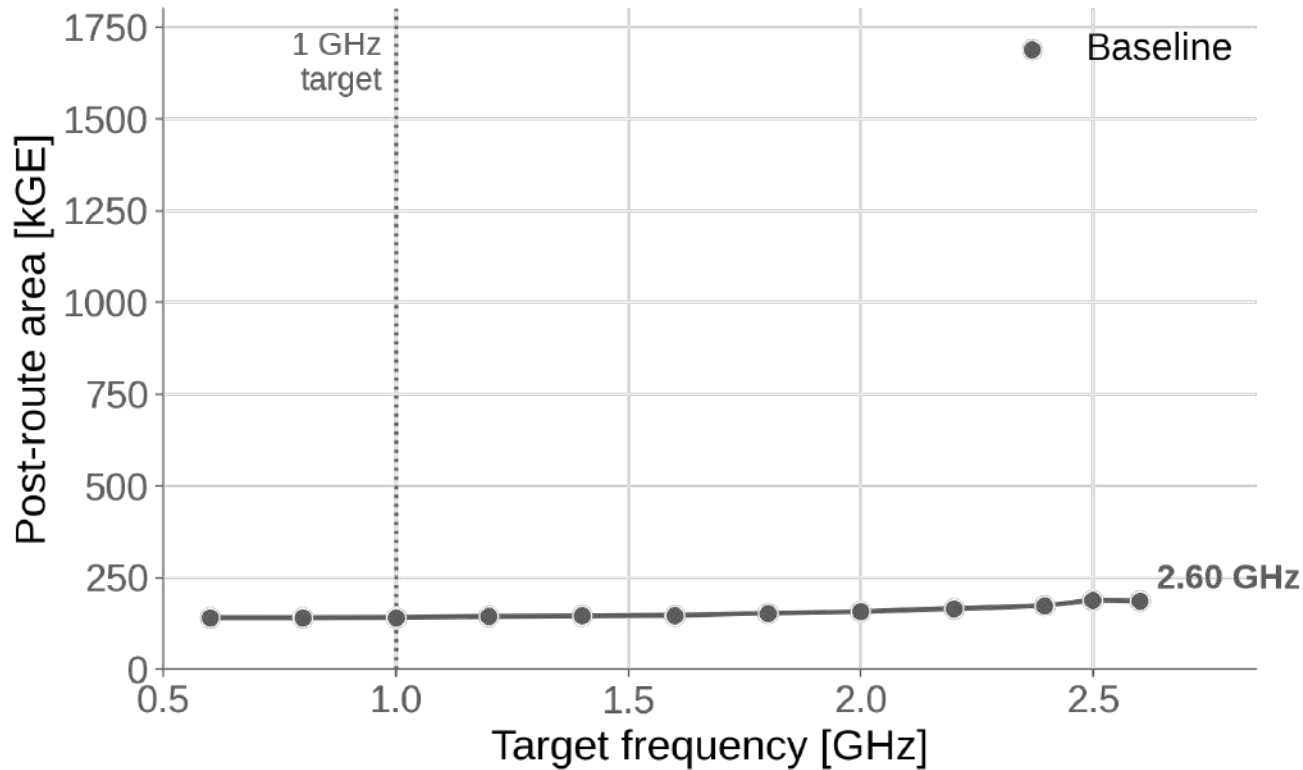
- TSMC 7 nm FinFET, RTL-to-GDSII
- Synopsys Fusion Compiler 2024.09
- Worst case: SS, 125 °C, 0.675 V
- Target-frequency sweep per design
 - die sized as in the snitch cluster tile

Keeping the redundancy alive

- No merging of replica registers
 - no shared multibit cells, clock or reset between replicas
- Voters preserved
 - boundary optimisation and ungrouping disabled for voter modules
- Minimum Manhattan distance between replicas
 - placement constraint against multi-bit upsets



Area vs. frequency: TMR cost grows with granularity



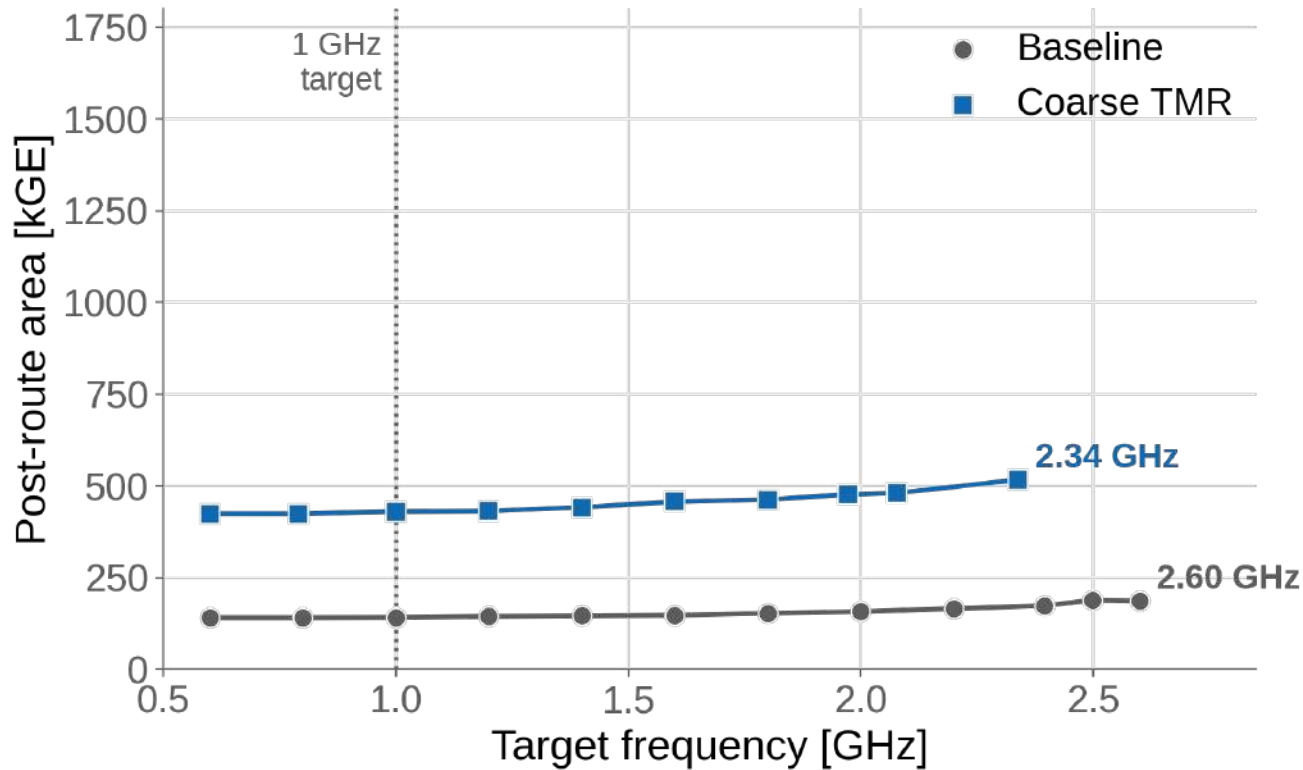
Baseline

141 kGE at 1 GHz

fmax 2.60 GHz



Area vs. frequency: TMR cost grows with granularity



Coarse TMR

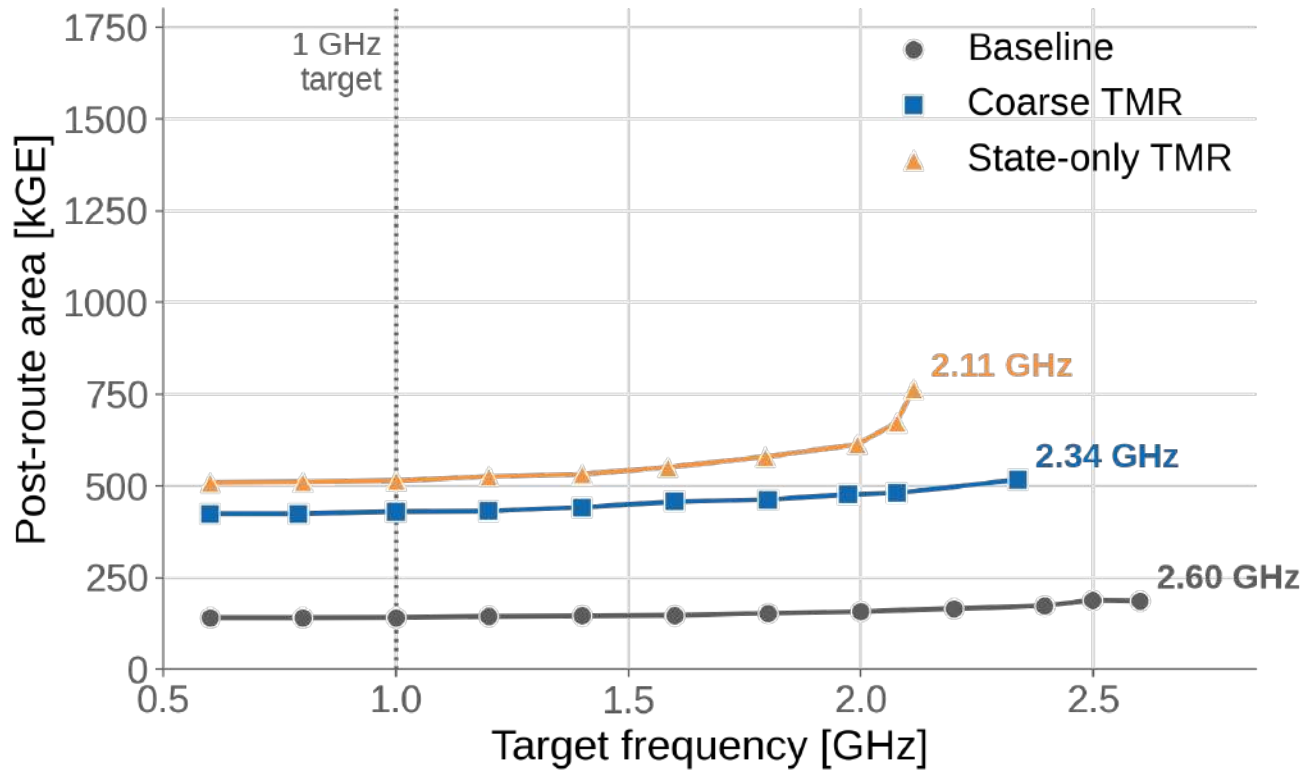
3.05× area at 1 GHz

fmax 2.34 GHz

- pure triplication
- no voters inside



Area vs. frequency: TMR cost grows with granularity



State-only TMR

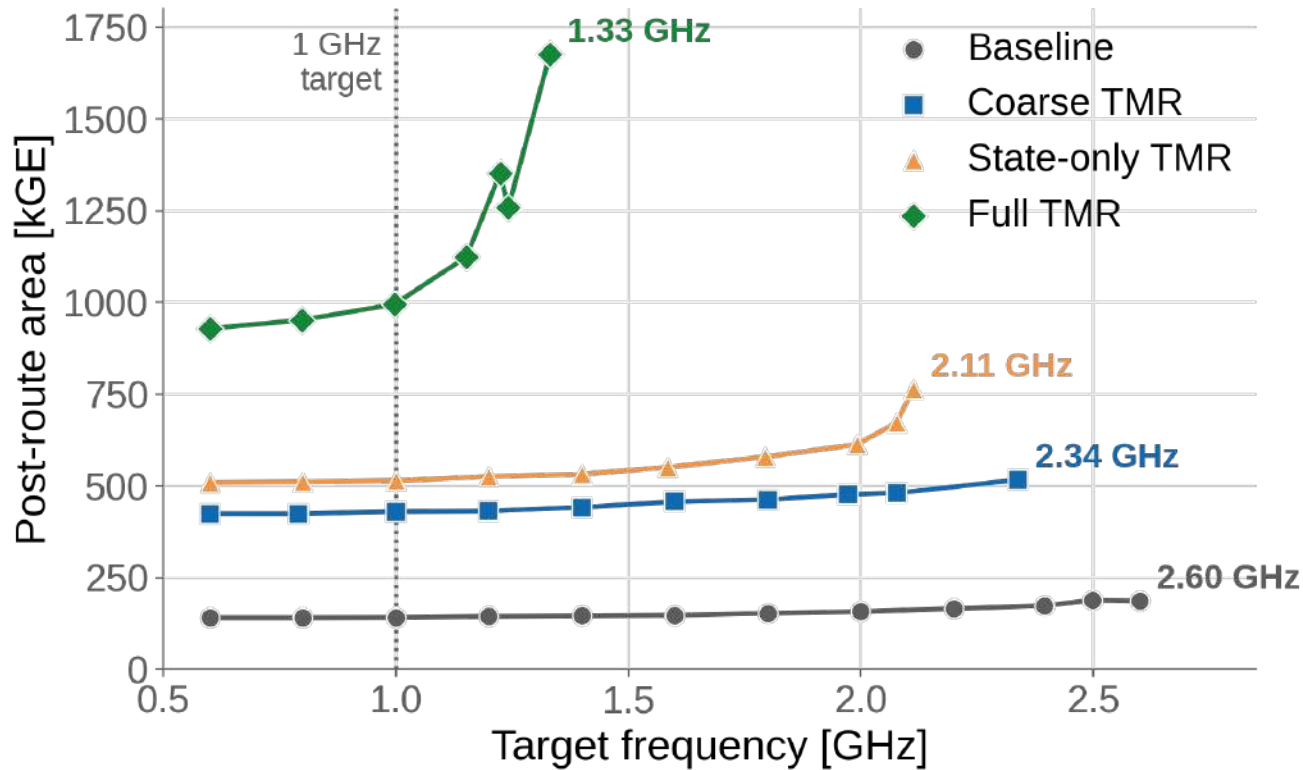
3.64× area at 1 GHz

f_{max} 2.11 GHz

- internal voters + clock-gate feedback on the critical path



Area vs. frequency: TMR cost grows with granularity



Full TMR

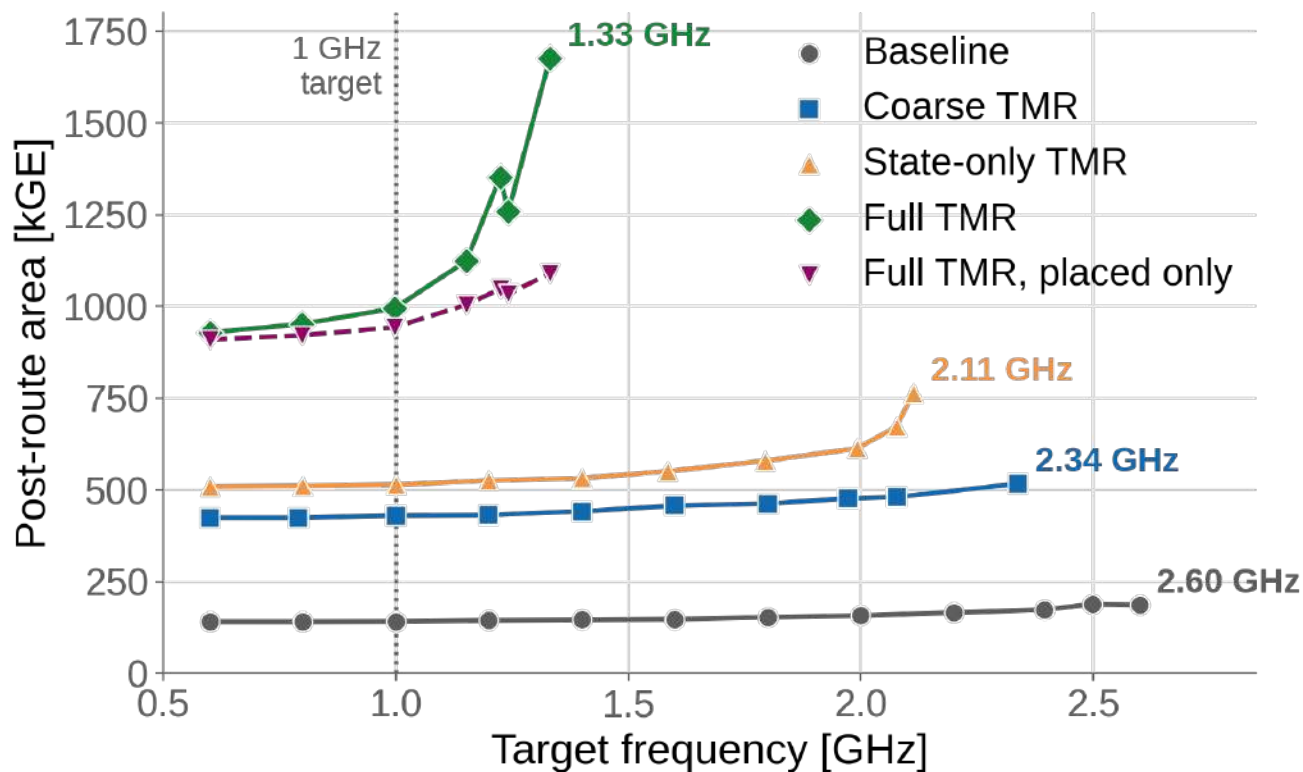
7.04× area at 1 GHz

fmax 1.33 GHz, about half of baseline

- area explodes when pushed toward fmax



Area vs. frequency: TMR cost grows with granularity



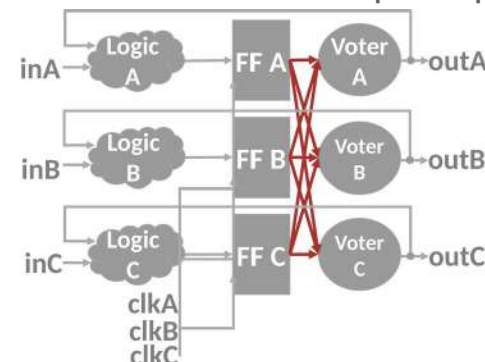
Where the full-TMR area goes

routing adds up to 50 % area near f_{max}

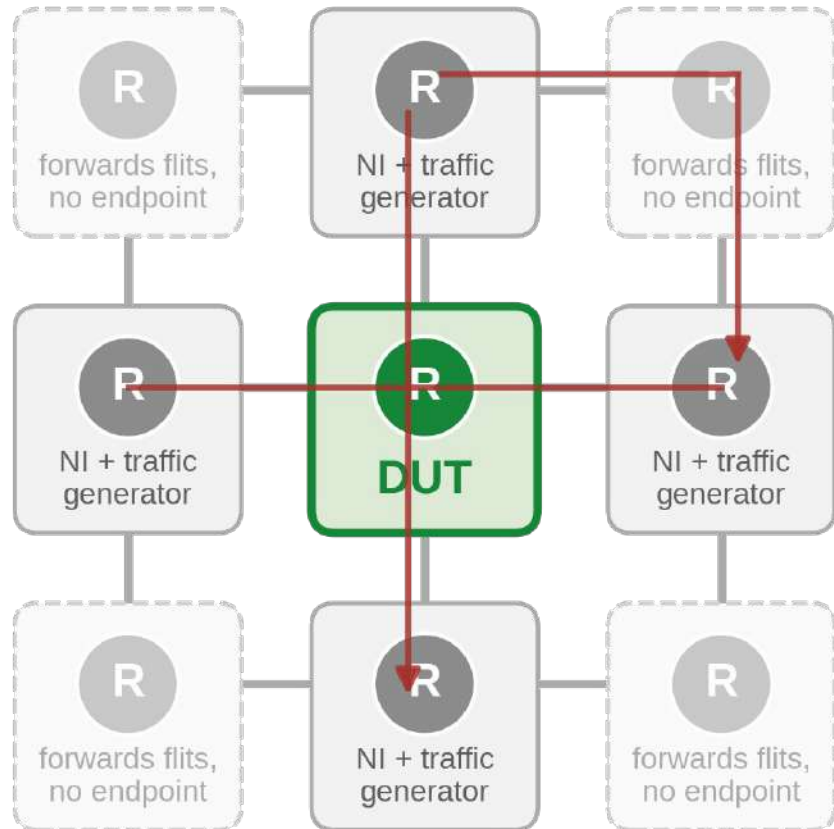
purple dashed = placed only,
green solid = routed

- a voter after every FF, and FFs are > 70 % of the router
- local congestion from per-FF voters: routing closes timing with buffers

Full TMR: 9 FF $\rightarrow$ voter wires per triplet (red)



Fault injection with VC Z01X: RTL and 7 nm netlist



400 random transactions between the 4 endpoints and the DUT

Tool

Synopsys VC Z01X 2025.06: concurrent fault simulation on RTL (SEU) and on the placed-and-routed netlist (SET)

Statistical sampling

99 % confidence, 0.1 % interval → about 1.6 million faults per campaign

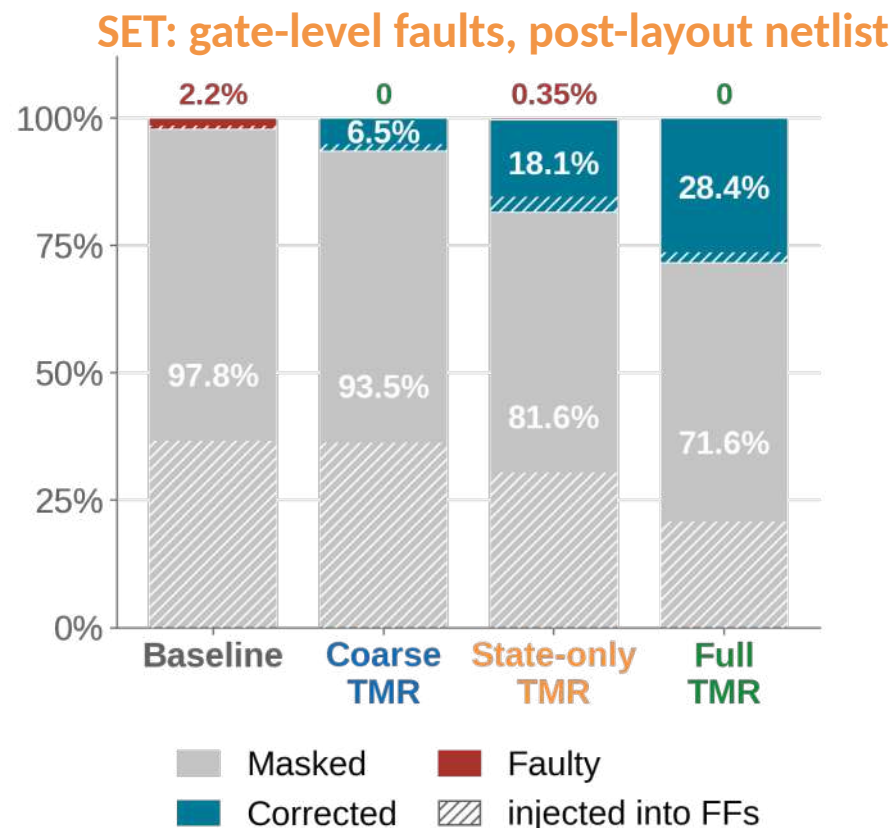
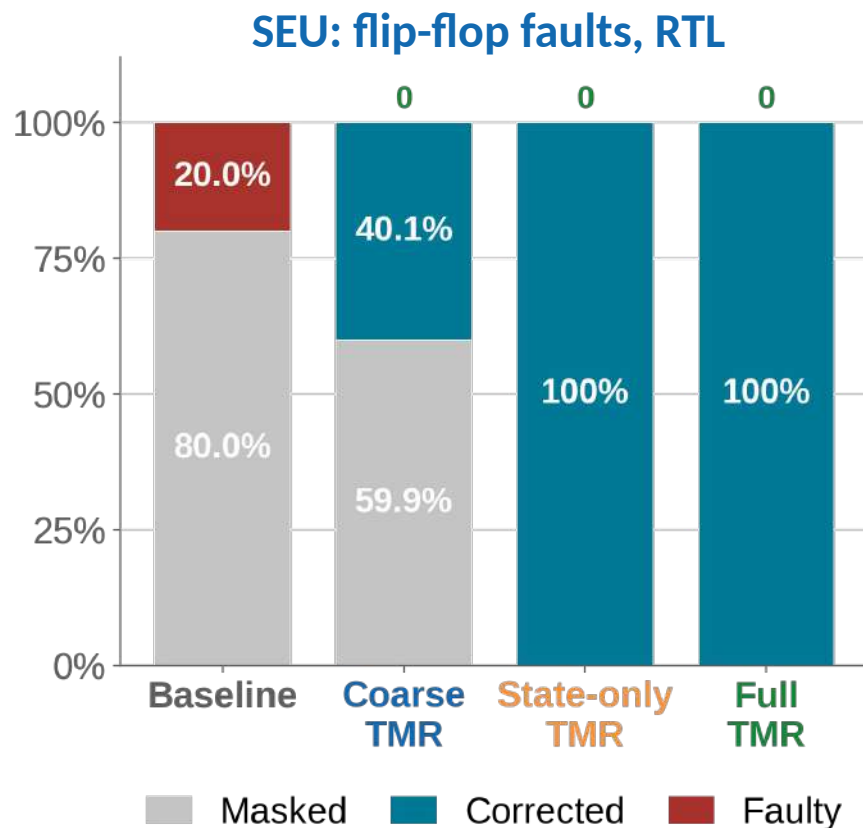
Outcome

Masked • Corrected • Faulty

Faulty = corrupted or misrouted flit, or a flit that never leaves the router.



Single faults: state-only TMR is vulnerable to SETs



Baseline 20 % faulty, every TMR variant 0

State-only: 0.35 % escape. A SET in shared logic is latched by all three replicas.

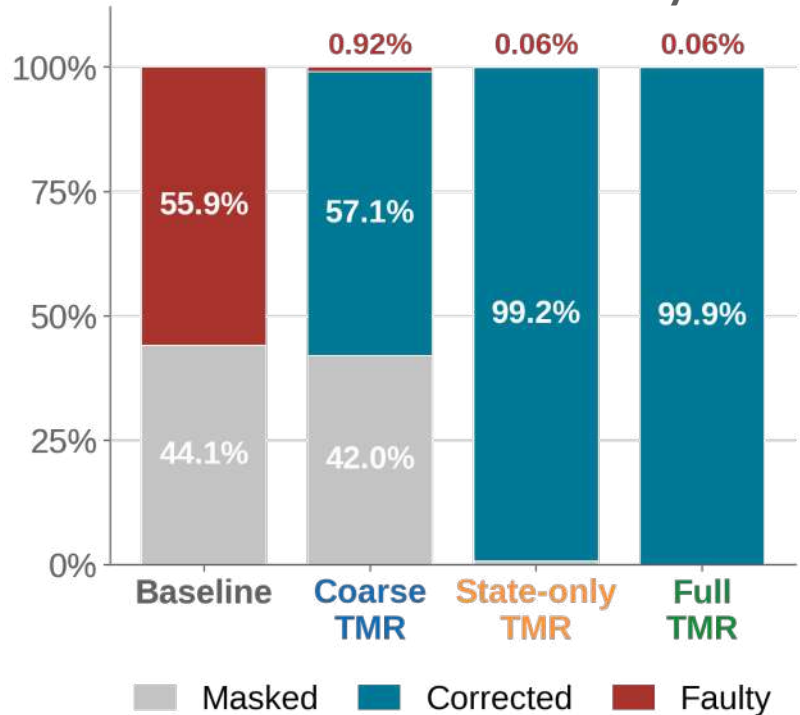


Two SEUs: coarse TMR breaks when faults accumulate

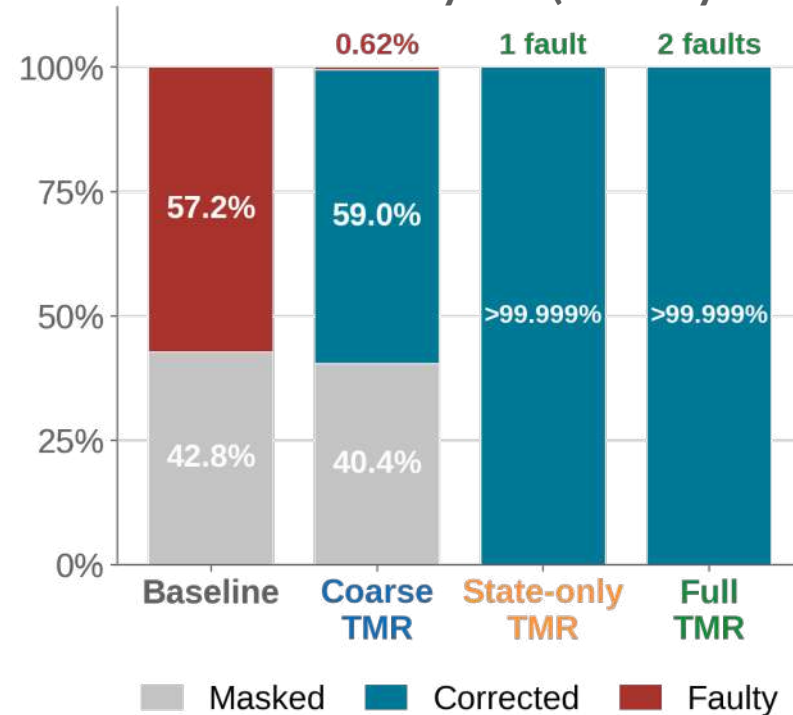


Two SEUs into control FFs only (buffer full, output valid). Data FFs (buffer entries) excluded: every new flit can overwrite them.

Two SEUs in the same cycle



Two SEUs at random cycles (1000-cycle window)



< 1 % for all: both faults hit two replicas of one FF (TMR's inherent limit)

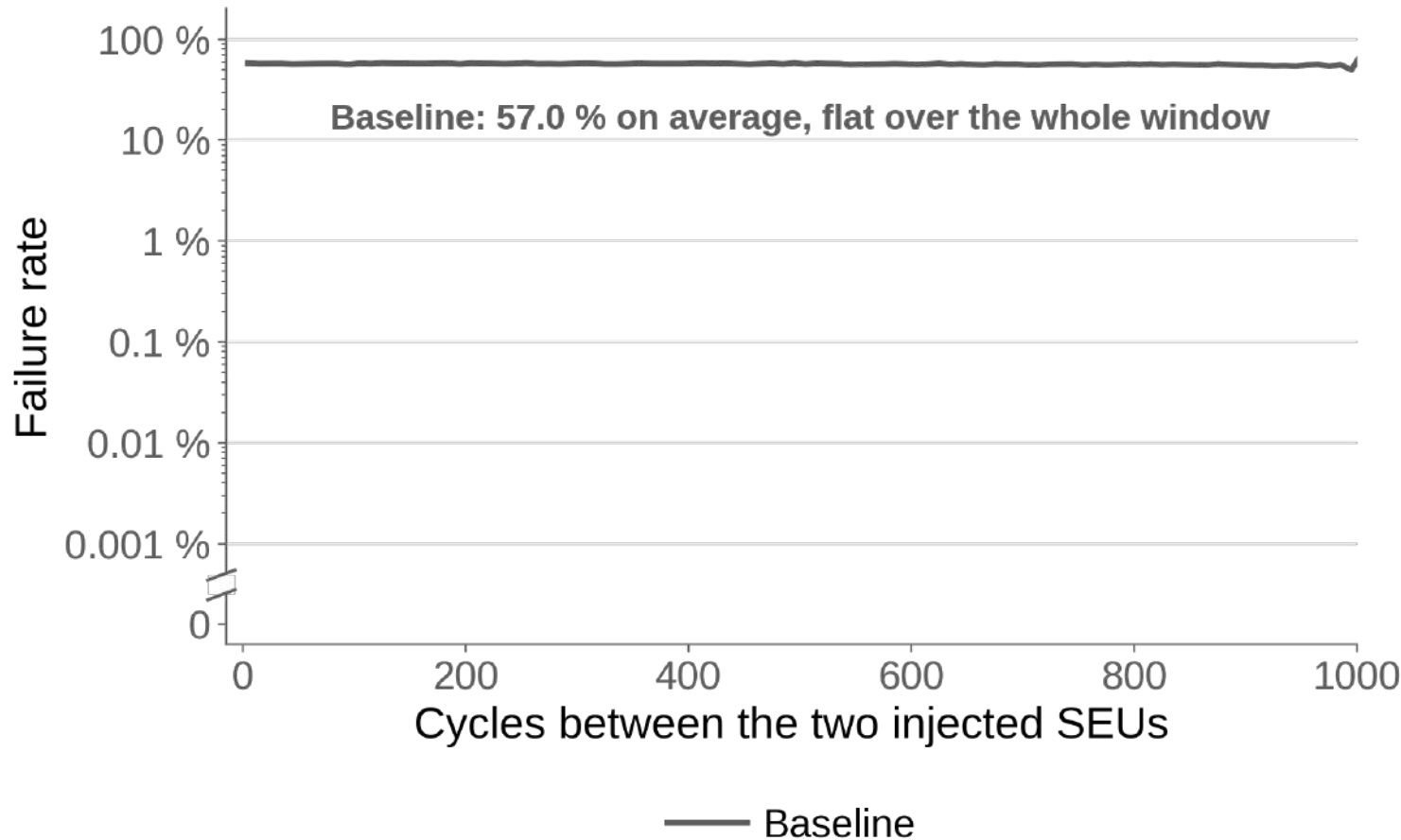
Coarse 0.6 % vs. 1 and 2 failures in 1.66 M for state-only and full. Faults are accumulated over time.



Coarse TMR cannot recover from accumulated faults



Two SEUs into control FFs only (buffer full, output valid). Data FFs (buffer entries) excluded: every new flit can overwrite them.



Baseline

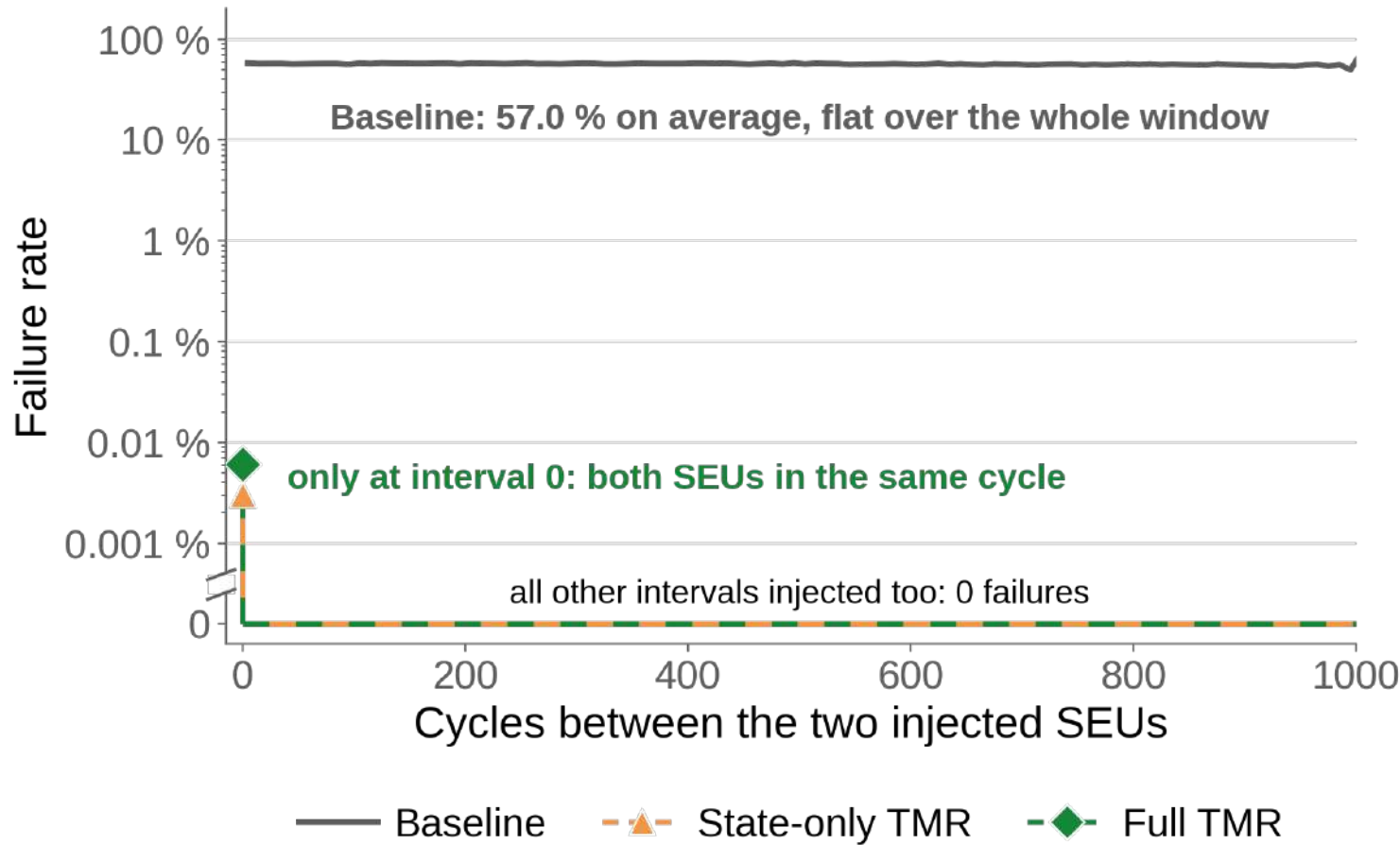
57 % failures, independent of the interval



Coarse TMR cannot recover from accumulated faults



Two SEUs into control FFs only (buffer full, output valid). Data FFs (buffer entries) excluded: every new flit can overwrite them.



State-only and full TMR

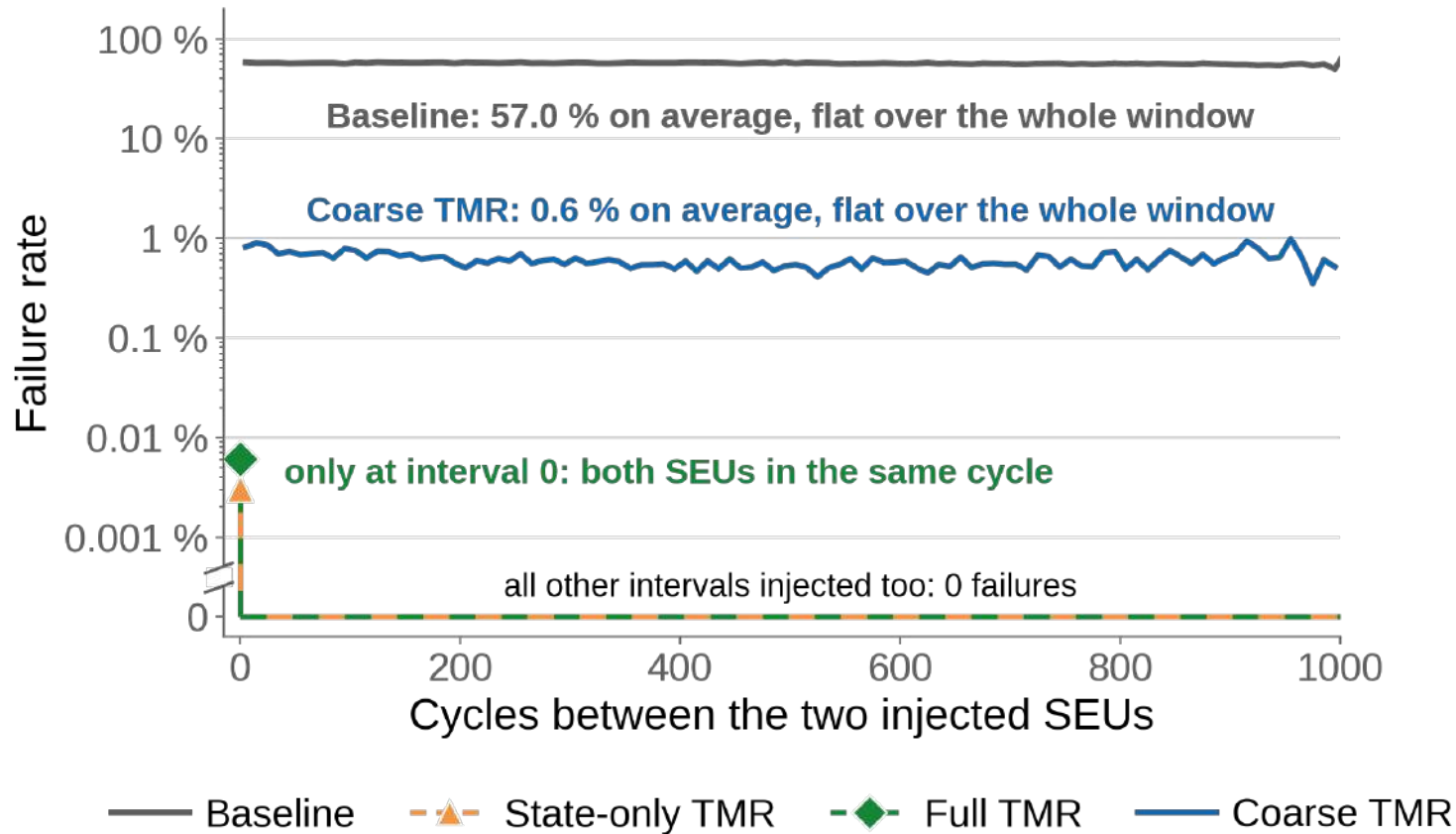
- failures only at interval 0: both SEUs in the **same cycle**, **same FF group**
- **No fault accumulation:** the first one is already corrected when the second arrives



Coarse TMR cannot recover from accumulated faults



Two SEUs into control FFs only (buffer full, output valid). Data FFs (buffer entries) excluded: every new flit overwrites them.



Coarse TMR

0.6 %, flat: the interval does not matter

- a corrupted control FF is not resynchronised
- the replica diverges; any later fault in another replica breaks the vote

Correcting every cycle inside the router is what stops faults from accumulating.



At tile level, even full TMR is affordable



+16.8 %

tile area with full TMR

+15.2 %

tile power, FP64 GEMM

1 GHz met

no timing loss, critical path in
the core

Router	Tile area (kGE)	Router area	Tile power (mW)	Router power
Baseline	5006	161	83.8	3.2
Coarse TMR	5286 (+5.6 %)	443 (2.75×)	93.3 (+11.3 %)	11.7 (3.69×)
State-only TMR	5337 (+6.6 %)	526 (3.26×)	87.0 (+3.8 %)	6.2 (1.96×)
Full TMR	5847 (+16.8 %)	965 (5.99×)	96.5 (+15.2 %)	11.6 (3.66×)

TSMC 7 nm, Fusion Compiler, SS 125 °C 0.675 V, 1 GHz. Power on an FP64 GEMM kernel.

Full TMR is the most costly scheme but the only reliable one, and at $\approx 3\%$ of the tile the router makes it affordable insurance.



Take-aways



The full TMR is the most robust solution

coarse TMR cannot recover from accumulated faults; state-only TMR lets SETs through

Full TMR is expensive, and routing is where it hurts

7.04× router area: per-FF voter wiring makes routing after placement far costlier than expected

Advanced nodes make it affordable

abundant routing resources: +16.8 % area, +15.2 % power, no timing impact at tile level



Chen Wu chenwu@iis.ee.ethz.ch

RTL of the three TMR routers + FI testbench:

github.com/imchenwu/FlooNoC/tree/tmr



scan for the code



Institut für Integrierte Systeme – ETH Zürich
Gloriastrasse 35
Zürich, Switzerland

DEI – Università di Bologna
Viale del Risorgimento 2
Bologna, Italy



References



- [1] T. Fischer et al., “FlooNoC: A 645-Gb/s/link 0.15-pJ/B/hop open-source NoC with wide physical links and end-to-end AXI4 parallel multistream support,” IEEE TVLSI, 2025.
- [2] F. Zaruba et al., “Snitch: A tiny pseudo dual-issue processor for area and energy efficient execution of floating-point intensive workloads,” IEEE Trans. Computers, 2021.
- [3] S. Kulis, “Single event effects mitigation with TMRG tool,” Journal of Instrumentation, 2017.
- [4] D. Fick et al., “Vicis: A reliable network for unreliable silicon,” DAC, 2009.
- [5] C. Feng et al., “Addressing transient and permanent faults in NoC with efficient fault-tolerant deflection router,” IEEE TVLSI, 2013.
- [6] T. H. Rausch et al., “Reliability and performance evaluation of a fault-tolerant MPSoC interconnection architecture,” IEEE DFT, 2025.
- [7] C. J. Elash et al., “Design and testing of a 32-bit radiation-tolerant RISC-V microcontroller at the 22-nm FD SOI node,” IEEE TNS, 2026.
- [8] S. Buchner et al., “Comparison of error rates in combinational and sequential logic,” IEEE TNS, 1997.
- [9] S. Lie, “Wafer-scale AI: GPU impossible performance,” Hot Chips 36, 2024.



Backup: scope of the evaluation and area weighting



- **Why the NI is outside the evaluation**

- border voters for coarse and full TMR live in the NI
- $NI \approx 1/6$ of the router area
- same TMR methodology applies; router is the critical block

- **Same-cycle, same-FF double hit**

- inherent to any TMR; suppressed by the enforced replica separation

Area weighting

A larger design intercepts proportionally more particles under a given flux.

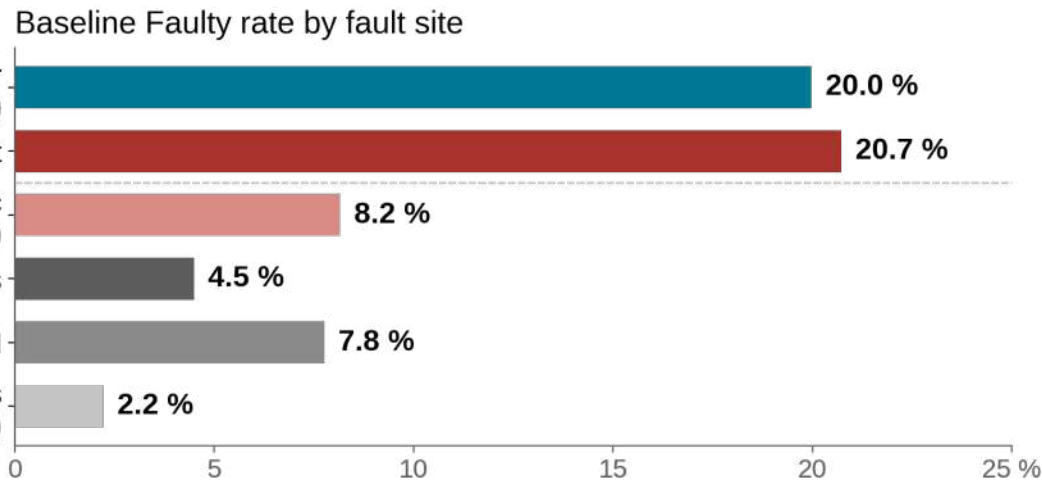
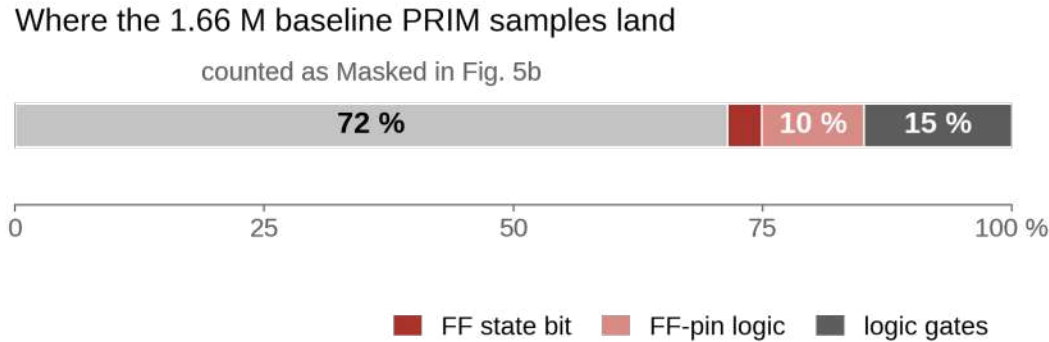
Full TMR corrects every single fault regardless of its larger cross-section, so weighting the per-fault outcomes by area does not change the ranking.

Testbench traffic

400 random transactions, uniform over narrow/wide read/write, all crossing the DUT; all ports and channels exercised.



Backup: why SETs hit the baseline less than SEUs



1. 72 % of the samples are unobservable

Each 7 nm FF cell model has 11 UDPs, only one holds the bit. The rest are untestable (UU) and count as Masked.

Testable faults only: **7.8 % Faulty**

2. A SET must be latched to matter

One-cycle pulse: needs a sensitised path into a register that loads at that edge.

An SEU in a buffer entry persists until the flit leaves.

Same site, same risk: SET on the FF bit 20.7 %, RTL SEU 20.0 %



Backup: why coarse TMR fails 15× more (Page 21)



The two SEUs land on ...	Share of pairs	Coarse TMR fails	State-only / Full fail
the same replica	33.3 %	0	0
two copies of the same FF	0.20 %	1,039	1,020
different FFs, two replicas, same link (req / rsp / wide)	22.1 %	13,820	0
different FFs, two replicas, different links	44.5 %	0	0
Faulty rate	1.62 M pairs	0.92 %	0.063 %

Coarse: one vote per link, at the NI

Any FF in replica A + any FF in replica B of the same link can corrupt the same output stream: $O(N^2)$ pairs.

e.g. input-buffer full flag (A) + output arbiter lock (B)

State-only / Full: one vote per FF, every cycle

Only two copies of the same FF defeat it: $O(N)$ pairs, inherent to any TMR.

Replica separation in placement makes this rare.

Random cycles (Fig. 6b), same mechanism: coarse 0.62 %, state-only 1 and full 2 failures out of 1.66 M.

