

# Fault-Tolerance Through-Silicon-Via for 3-D Integrated Circuits

---

**Khanh N. Dang**

Vietnam National University Hanoi, Vietnam

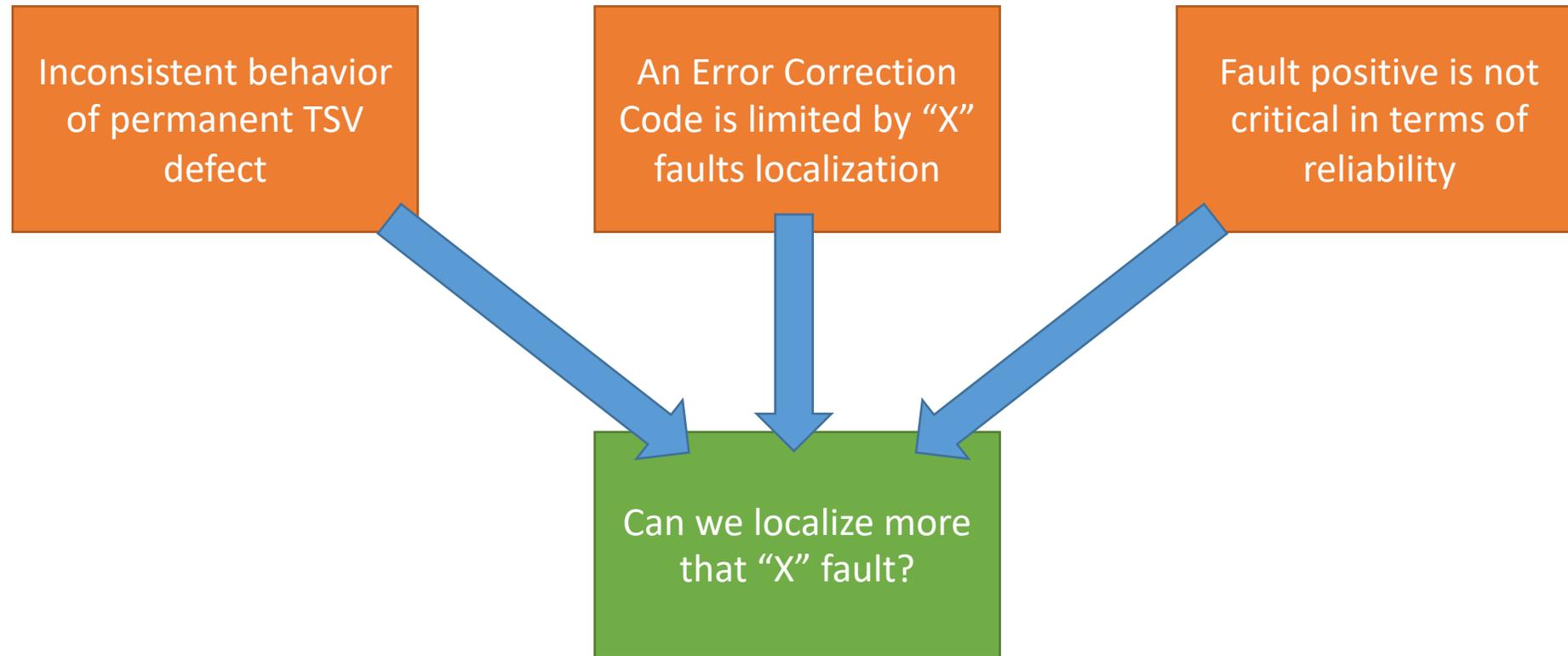
Email: [khanh.n.dang@vnu.edu.vn](mailto:khanh.n.dang@vnu.edu.vn)

# Acknowledgment

---

- This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2018.312.

# Summary



- Fault positive: a device is healthy but it was determined as faulty

› Introduction

› Proposed architecture

› Evaluation

› Conclusion

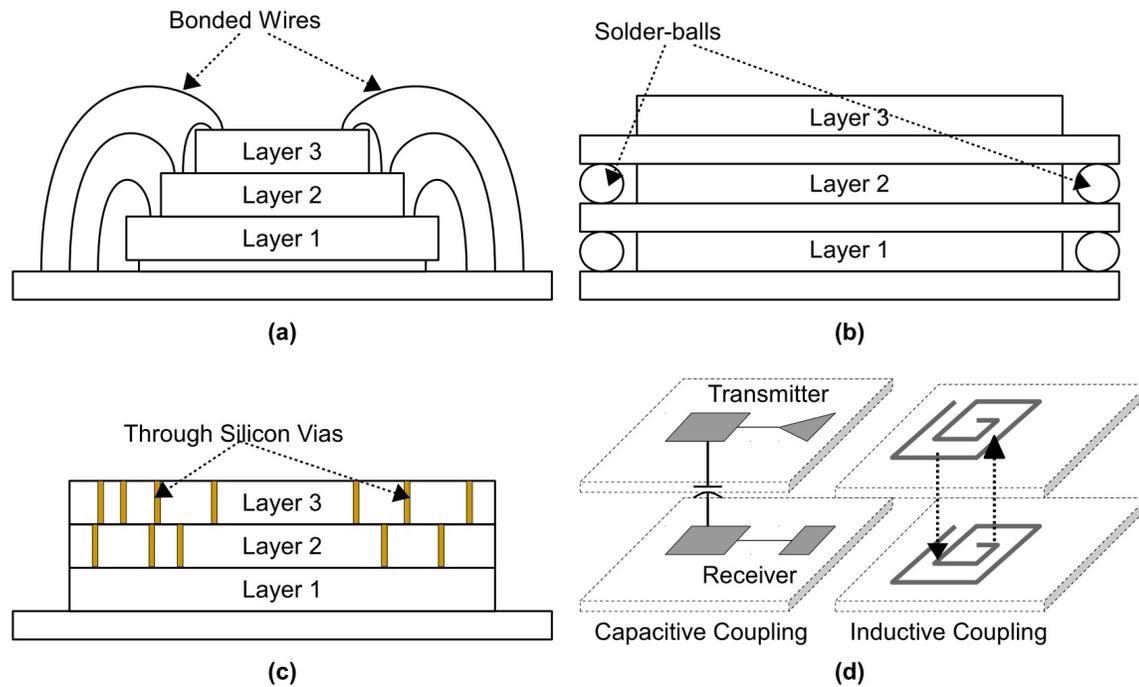
› Introduction

› Proposed architecture

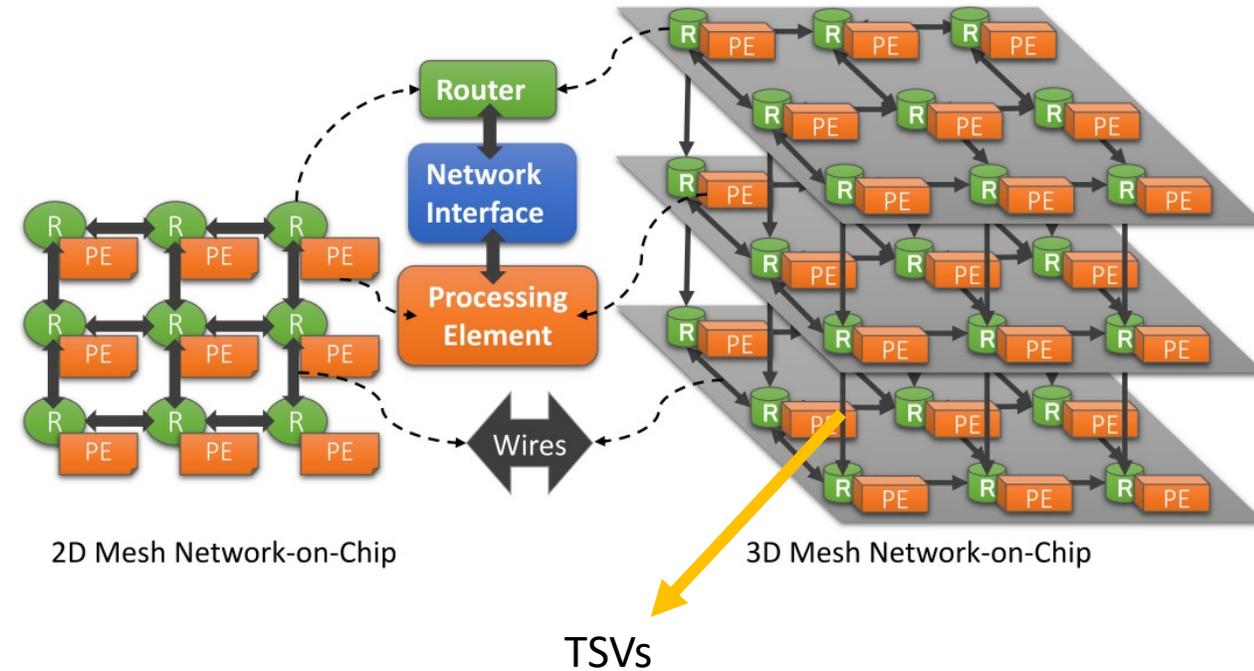
› Evaluation

› Conclusion

# Toward the 3D-ICs



3D Integration technologies: (a) Wire bonding; (b) Solder balls; (c) Through Silicon Vias (TSVs); (d) Wireless stacking.



3D Mesh Network-on-Chip to utilize TSVs as vertical links

To keep up with the increase of integration density, moving to the third dimensions could be an promising solution

One of the most mature near-future technology is TSV (Through Silicon Via)

# TSV Reliability Issue

Despite of several advantages, TSVs also have several reliability issues:

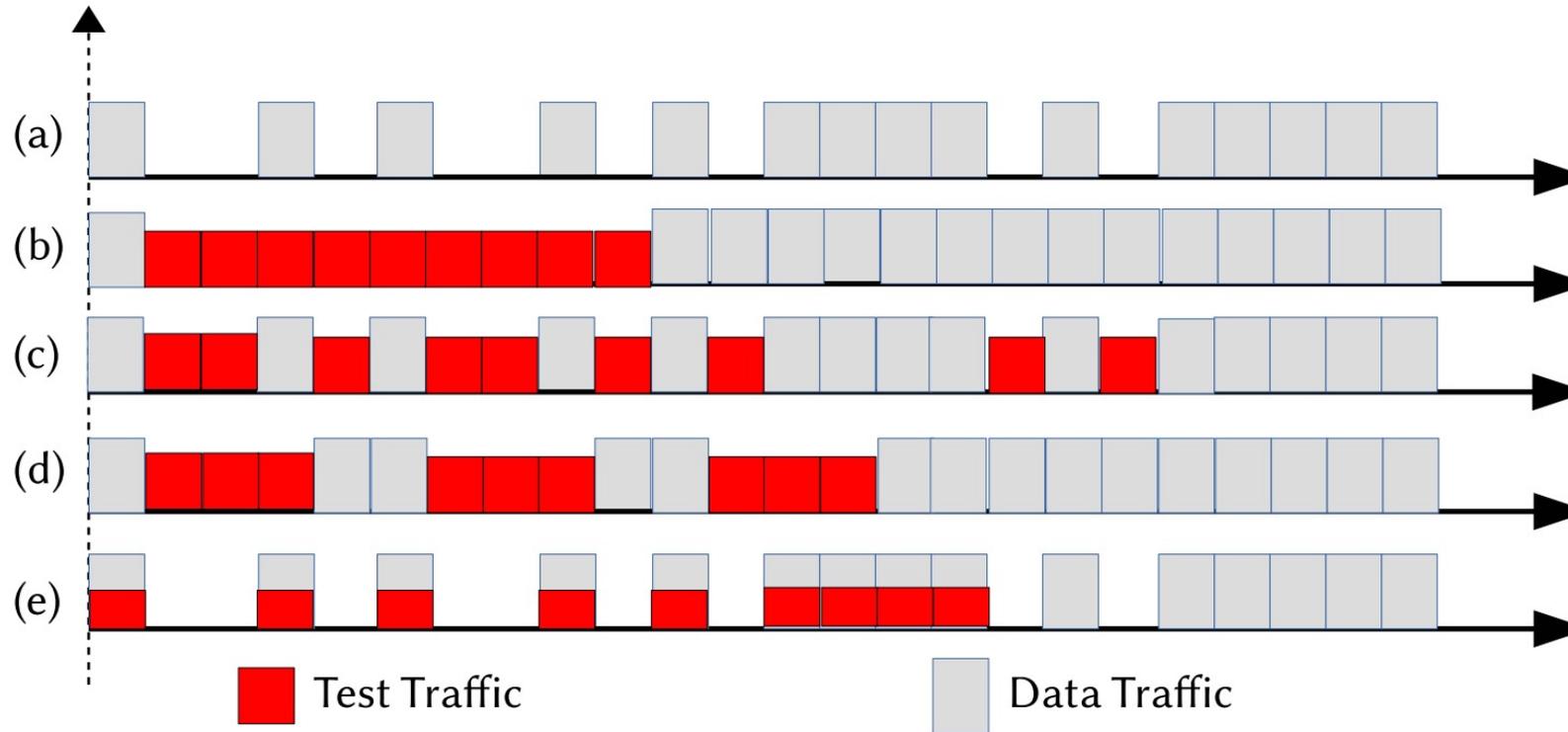
- The yield rate is low due to imperfect manufacturing processes.
- Cross-talk issues:
  - Because TSVs run in parallel, cross-talk is a common issue
- Vulnerable to thermal stress:
  - Difference in thermal expansion coefficients of materials.
  - The difference in layers' temperature could lead to crack/misalignment on TSV.
- Higher operating temperature:
  - Thermal dissipation become more difficult in 3D-Ics
    - Higher layers receive heat from bottom layers
    - TSV as heatsink or micro flood cooling is still immature

# To solve the reliability issue

---

- Testing:
  - Built-in-Self-Test, external testing, online testing:
    - For on-line, periodical BIST could be used.
  - Error Correction/Detection Code
    - For example: SECDED, OLSC, ...
- Recovery:
  - Hardware tolerance: Redundancies, Re-mapping
  - Information redundancies: coding, re-transmission
  - Algorithm based: fault-tolerant routing

# Online testing strategies



- The sequence of data and test traffic under different strategies: (a) application traffic; (b) block test; (c) free time test traffic injection; (d) split free time test; (e) **on-communication test**.

# Error Correction Code localization limitation

- In coding theory, a code  $C$  is said to be  $k$  error correcting (fault localization) if the minimum Hamming distance between two code word is at least  $2k+1$ .
- Using ECCs for fault detection and localization is possible, but limited by number of faults.
  - Extended Hamming code: 1 fault correction/localization, 2 fault detections
  - Orthogonal Latin Square code:  $m \times m$  data bit with  $2mt$  parity bit ( $t < m$ ) could correct  $t$  faults.
- If we can improve ECCs by increasing localization ability, we could obtain:
  - Non-blocking test
  - No performance degradation
  - Fast execution time.

# Contributions

In this paper, we propose a novel method named On-communication TSV Test (OCTT) which is specially designed for correcting and detecting faults in TSV-based links:

- A comprehensive set of algorithm including: *Statistical Detection* and *Isolation-end-Check*.
- *Statistical Detection* take the outputs of ECC and mark the suspicious positions
- *Isolation-end-Check* reconfirms the suspicious positions as healthy or faulty.

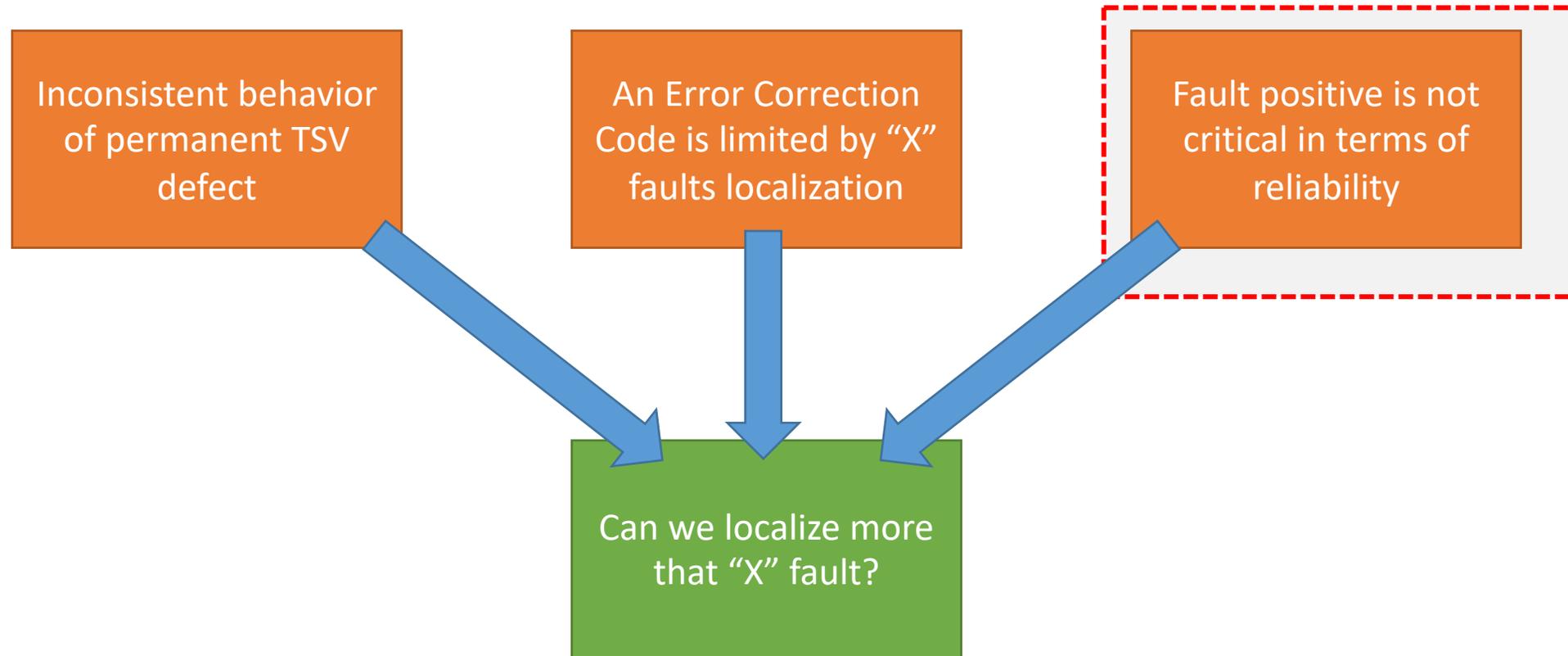
› Introduction

› Proposed architecture

› Evaluation

› Conclusion

# Proposed method



- Fault positive: a device is healthy but it was determined as faulty

# Testing accuracy

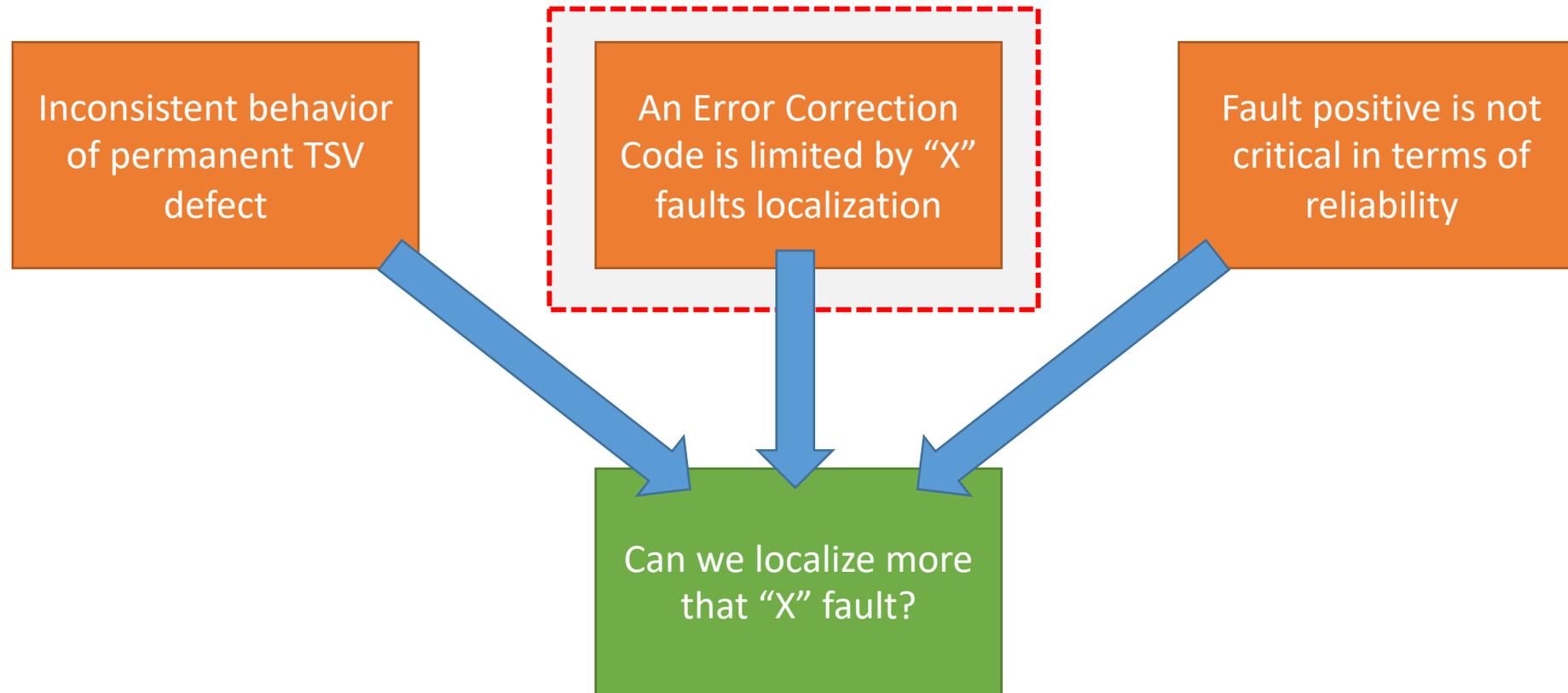
- In testing, there are four major cases.
  - False negative is the critical case: the device is faulty but the testing process determine it as healthy.
  - The false positive case is less critical.
- For ECCs, since it works **in parallel with the communication** (no interruption and no performance degradation):
  - The communication still perform as usual.
  - False positive is non critical.

		True status	
		Faulty	Healthy
Testing result	Faulty	True negative	False positive
	Healthy	False negative	True positive

Correct decision

Incorrect decision

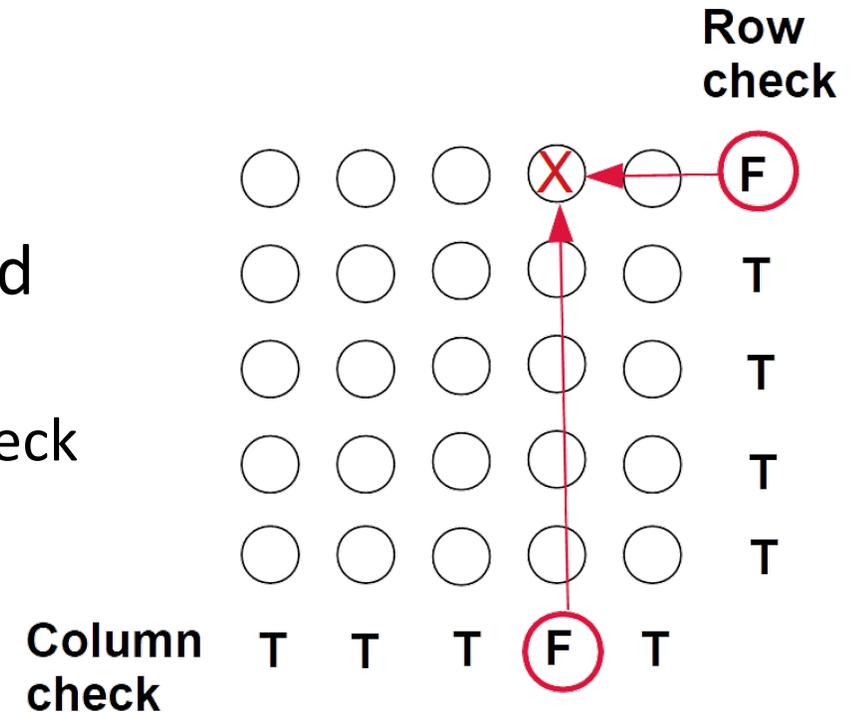
# Proposed method



- Fault positive: a device is healthy but it was determined as faulty

# Baseline Parity Product Code

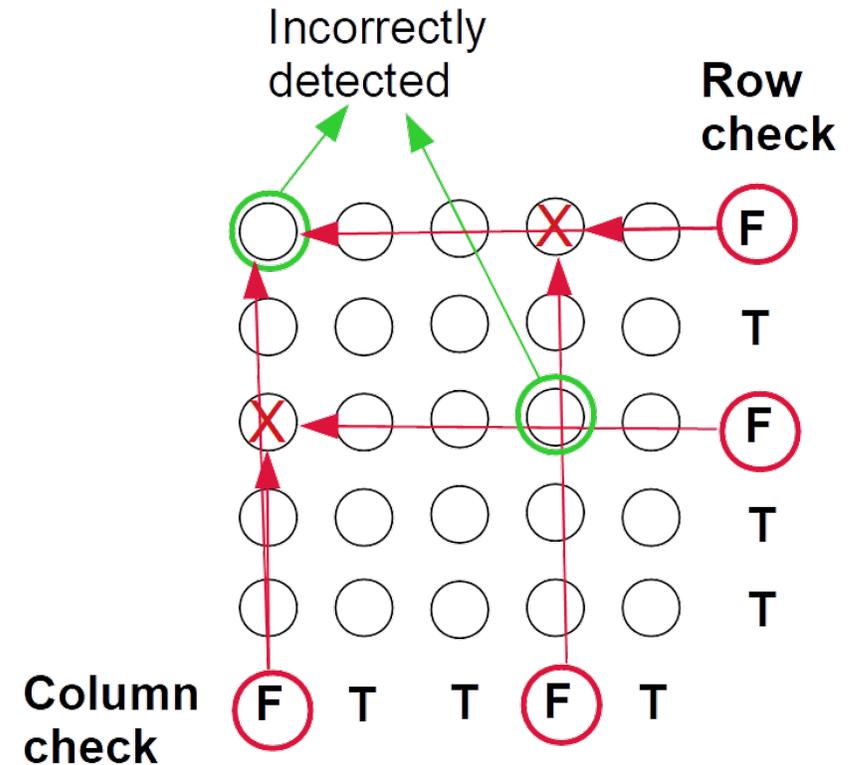
- A TSV-based connection are organized in a group  $M \times N$  data bit which is encoded into  $(M+1)(N+1)$  code-word bit.
- Parity bits are:
  - Row parity bits
  - Column parity bits
  - All bit parity
- When a flit of  $(M+1)$  rows  $(N+1)$  columns are received
  - Check row parity and column parity
  - Faulty position has faulty row check and faulty column check
- Parity product code (PPC) can:
  - Localize/correct one fault
  - Detect two faults



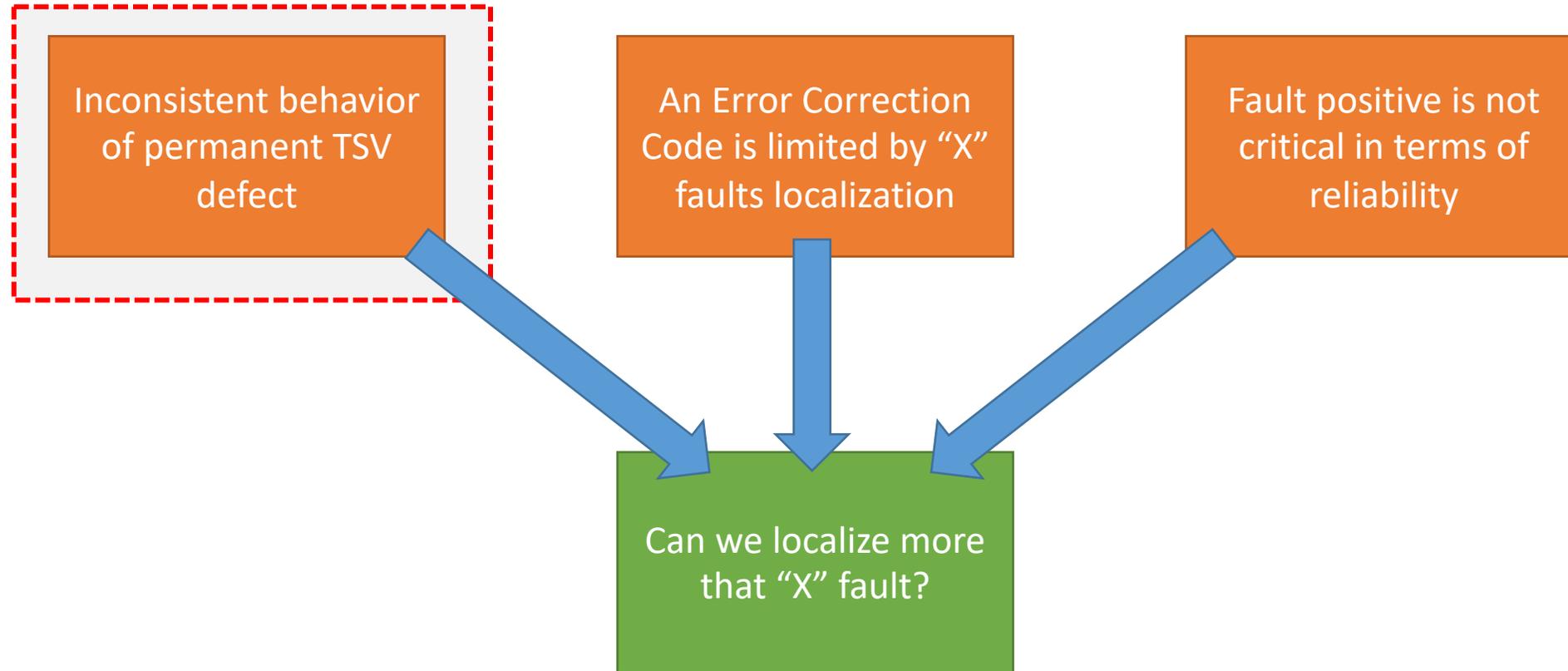
# Fault detection: 2+ faults

- If there is two or more faults, there is a chance of detection:
  - If there is 2+ faulty row check, or 2+ faulty column check, there are multiple faulty bit
- If we consider the localization:
  - **Red X**: true negative
  - **Green circle**: false positive
- Can we remove the false positive case?

		True status	
		Faulty	Healthy
Testing result	Faulty	True negative	False positive
	Healthy	False negative	True positive



# Proposed method



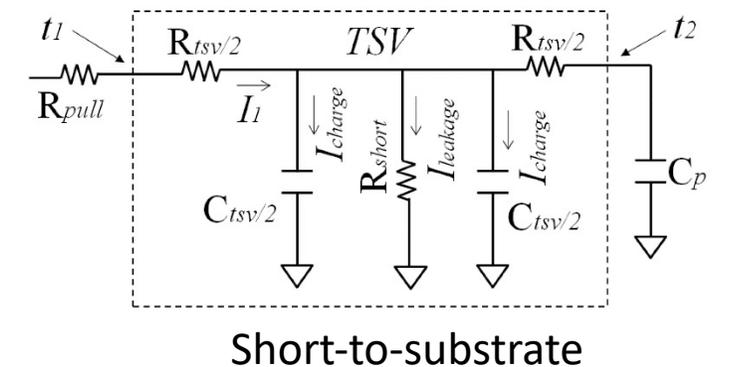
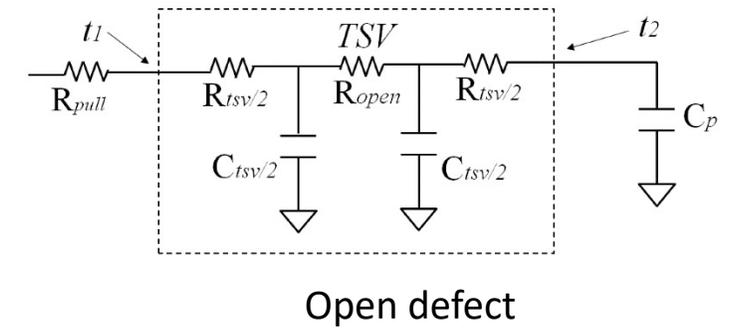
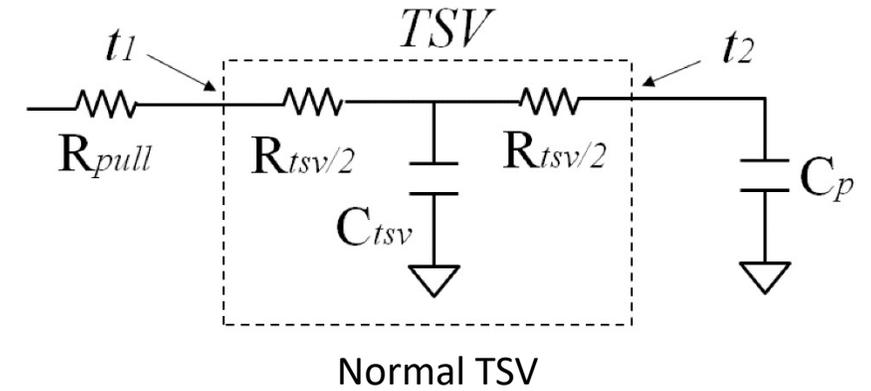
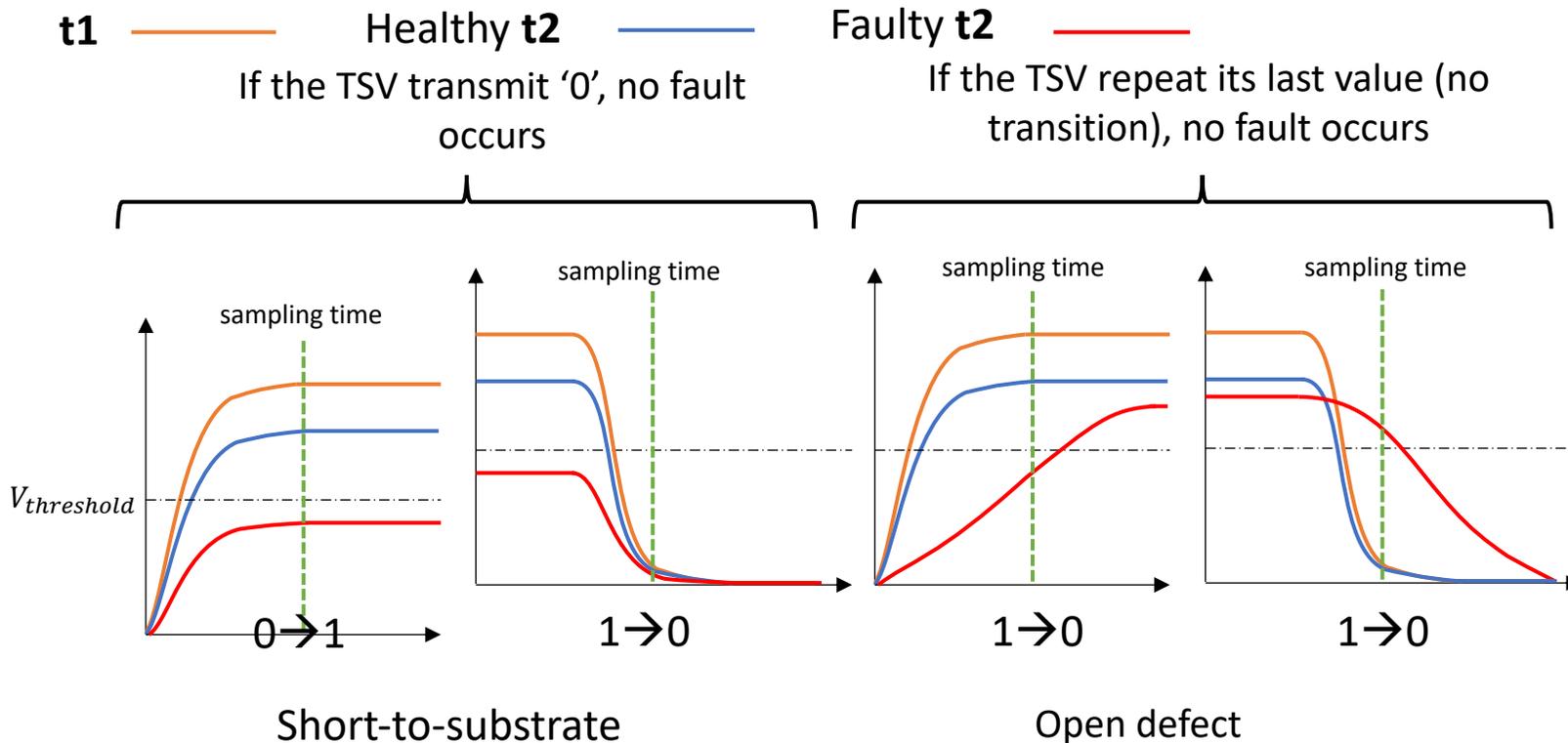
- Fault positive: a device is healthy but it was determined as faulty

# Fault model

The TSV defects are modeled as:

- Short-to-substrate: the value of TSV is stuck at '0'.
- Open: slow down the transition of TSV, delay the value of TSV by one clock cycle.

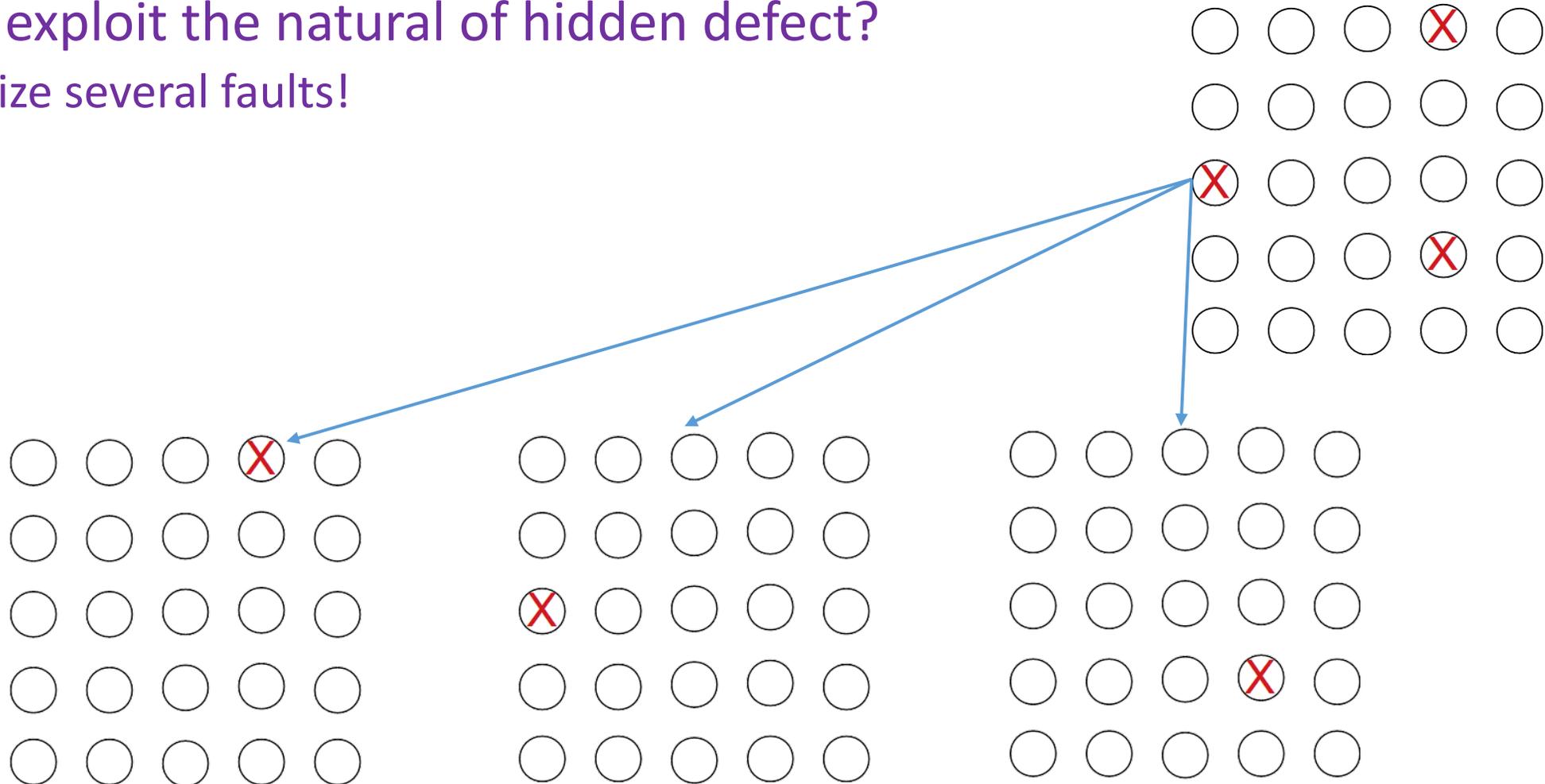
➔ Hidden fault could happen during operation



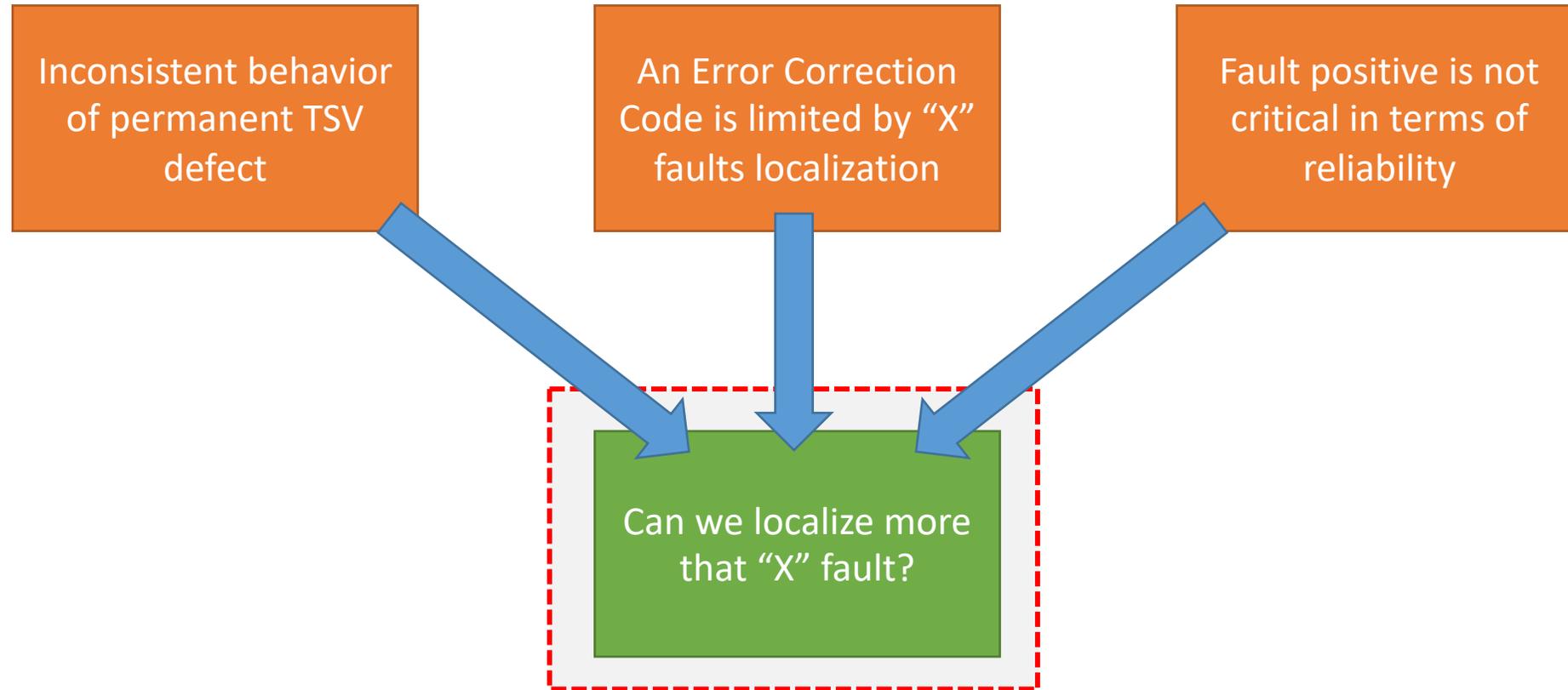
Y. Zhao et al., "Online Fault Tolerance Technique for TSV-Based 3-DIC," IEEE Trans. VLSI Syst., vol. 23, no. 8, pp. 1567–1571, 2015.

# Hidden defects

- Because of the hidden defects, we could observe the inconsistency of the ECC correction patterns
- Can we exploit the natural of hidden defect?
  - Localize several faults!



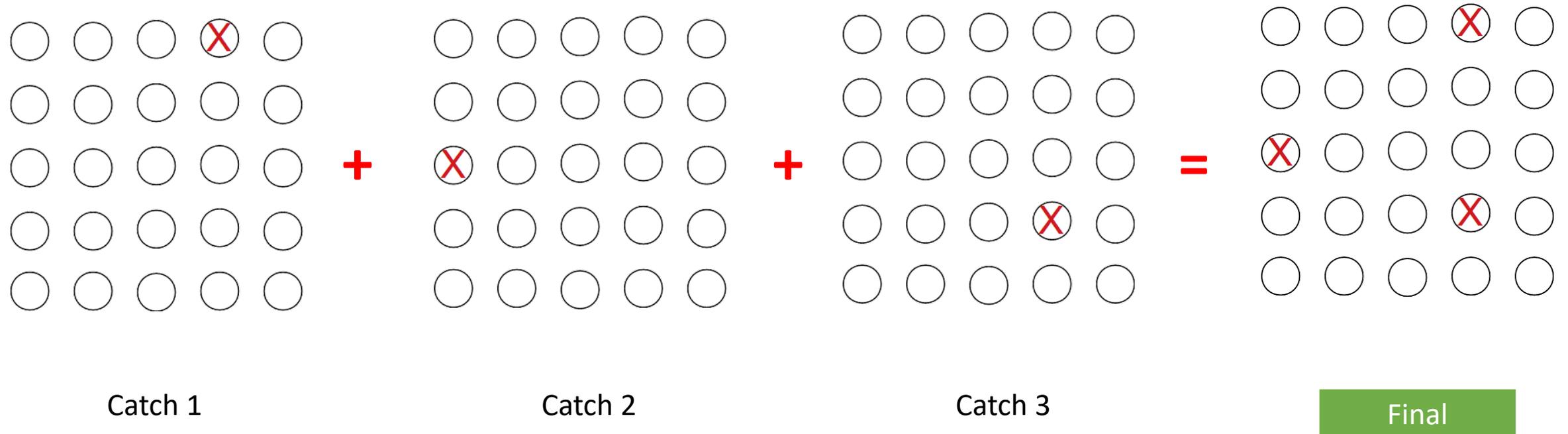
# Proposed method



- Fault positive: a device is healthy but it was determined as faulty

# Statistical detection: cautious localization

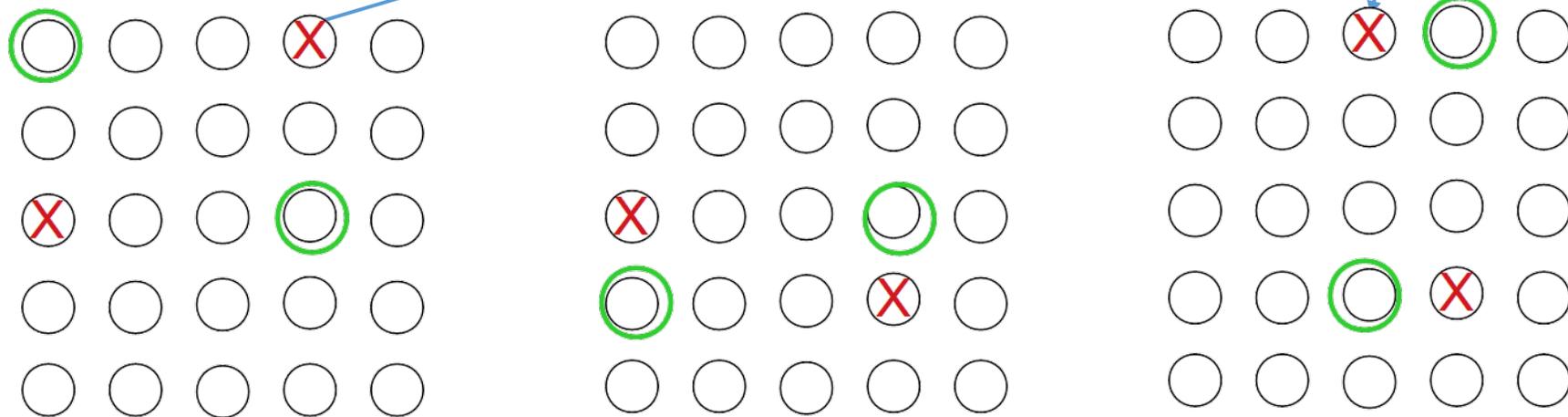
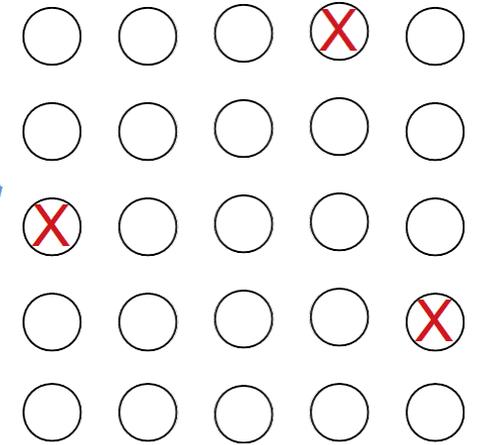
- Perform ECC in T cycles (period) to capture as much as possible:
- If there is only one fault occurrence, we could accumulate the result!



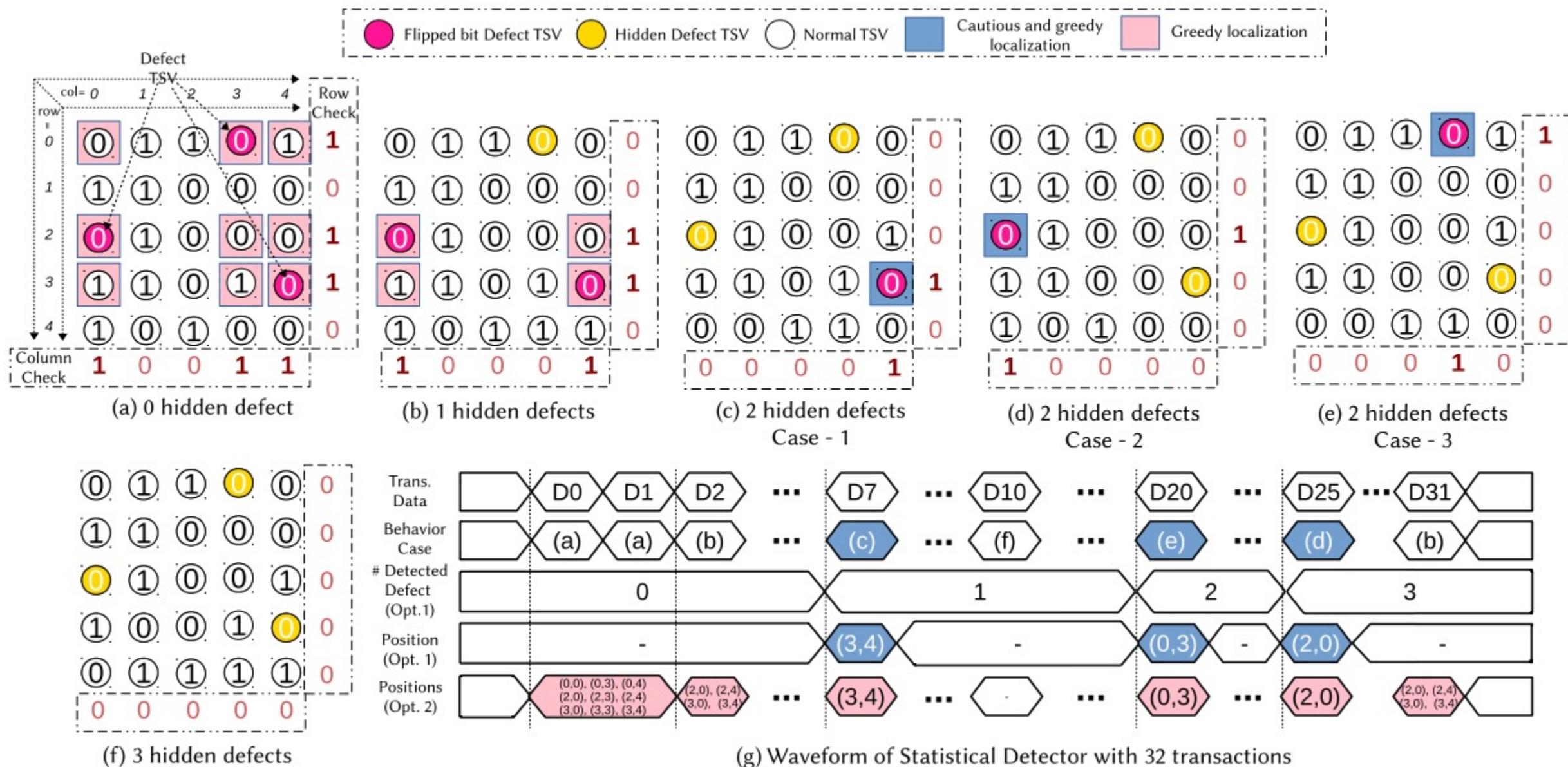
- By cautiously capture single fault localization, a single error correction coding technique could localize more than one fault.

# Statistical Detection (continue)

- Perform PPC in several cycle to capture as much as possible:
- With two or more defect occurrences, there are false positive cases
  - If we are cautious, just skip this case and wait until on fault left
- However, if there is multiple faults, we “cannot” wait:
  - Let be greedy!

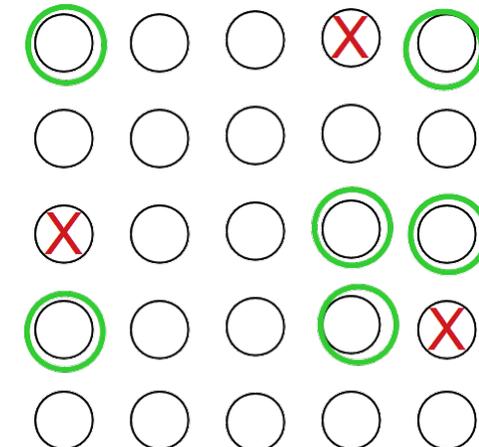
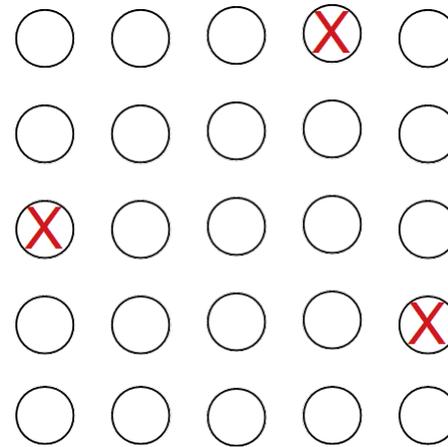


# Statistical detection: greedy localization



# Statistical detection: Output of greedy localization

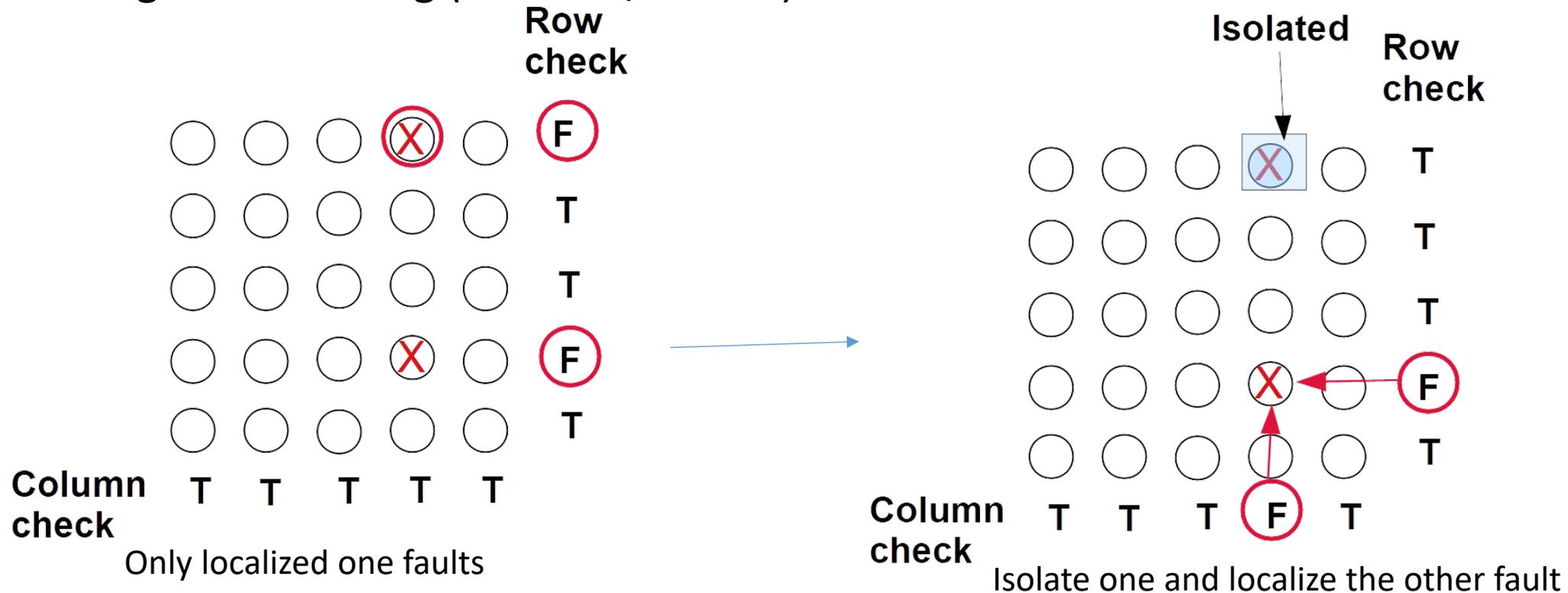
- It greedily captures as much as position as possible:
  - They are suspicious TSVs.
- Accuracy
  - **Red X**: true negative
  - **Green circle**: false positive
- Since the false positive TSV still transfer data:
  - It is not critical
- However:
  - Are there still hidden defects?
  - Is it possible to remove the false positive cases?



Output

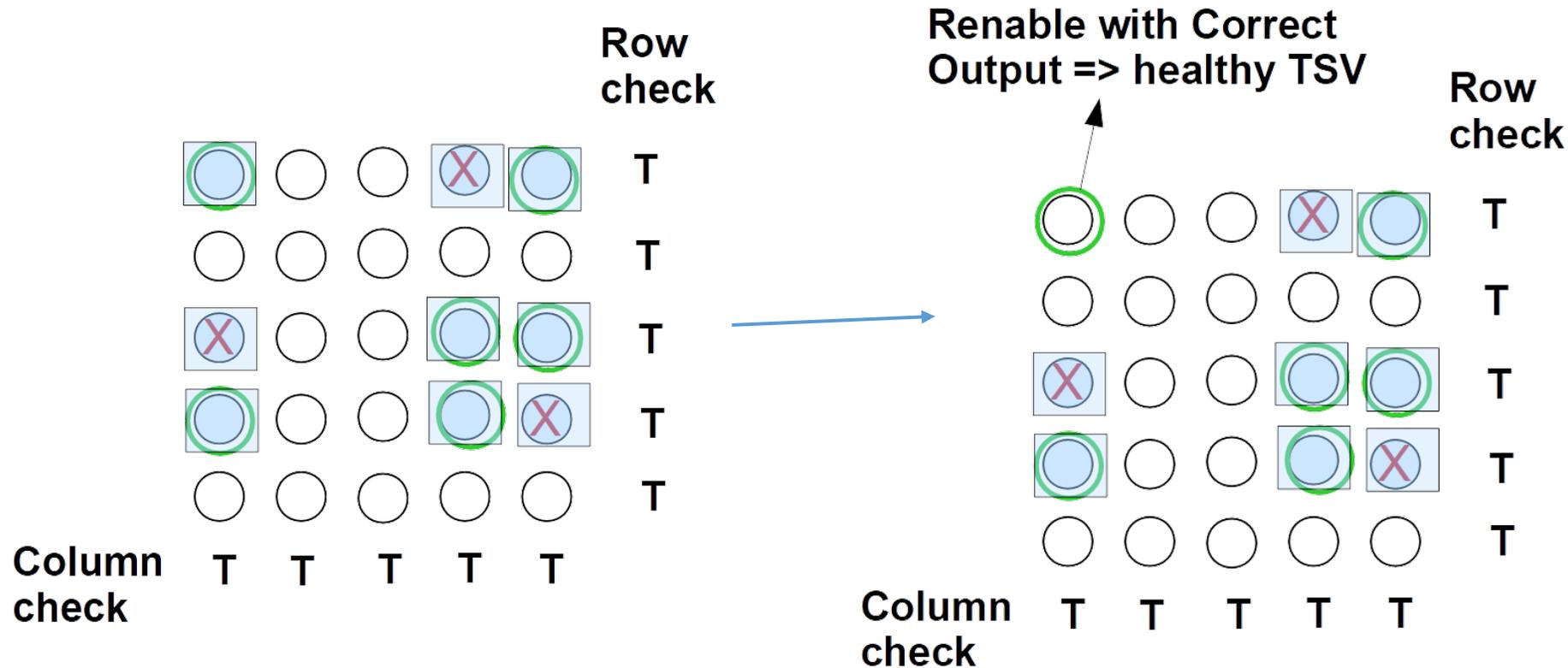
# Isolate-and-Check: keep isolating

- After the statistical detection, we have a set of suspicious positions.
- We “isolate” them:
  - They still transfer data as usual
  - The ECC (parity product code) do not take consideration of its value: remove it from encoding and decoding (use mux/demux).



# Isolation-and-Check: re-checking

- Re-assign (disable isolation) each suspicious TSV:
  - If the output of ECC shows the suspicious TSV healthy: false positive case
  - If the output of ECC shows the suspicious TSV faulty: true negative case



› Introduction

› Proposed architecture

› Evaluation

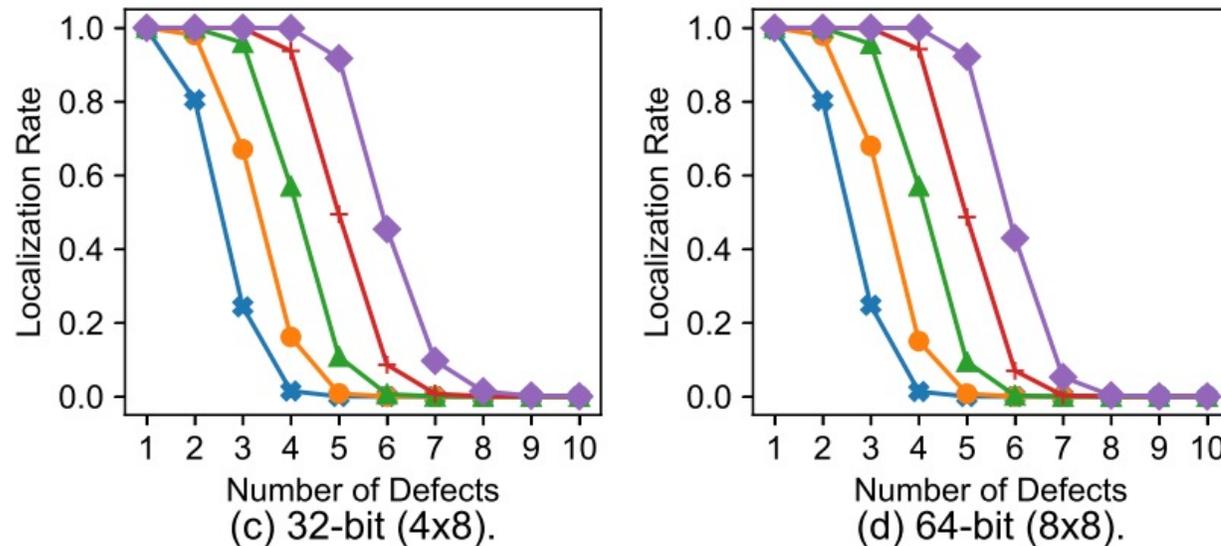
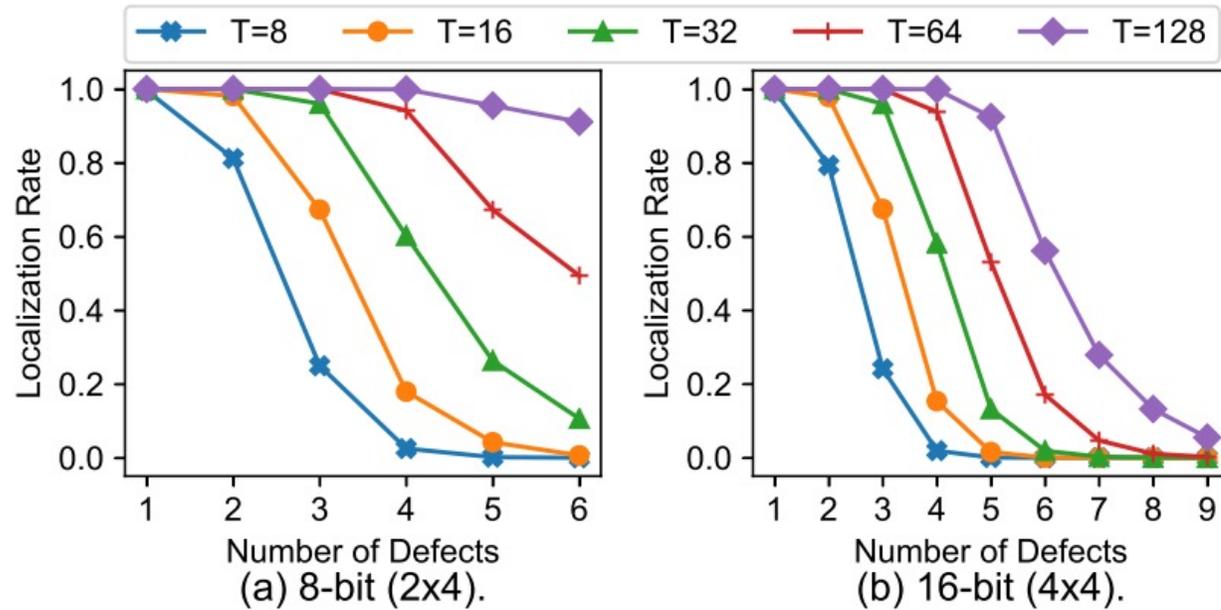
› Conclusion

# Evaluation methodology

- Design in Verilog HDL
- Design with:
  - NANGATE 45nm and NCSU FreePDK TSV
  - Synopsys Design Compiler
  - Cadence Innovus
- Localization performance:
  - Monte-Carlo simulation with 10,000 tests.
    - For small number of tests, brute-force test is used.

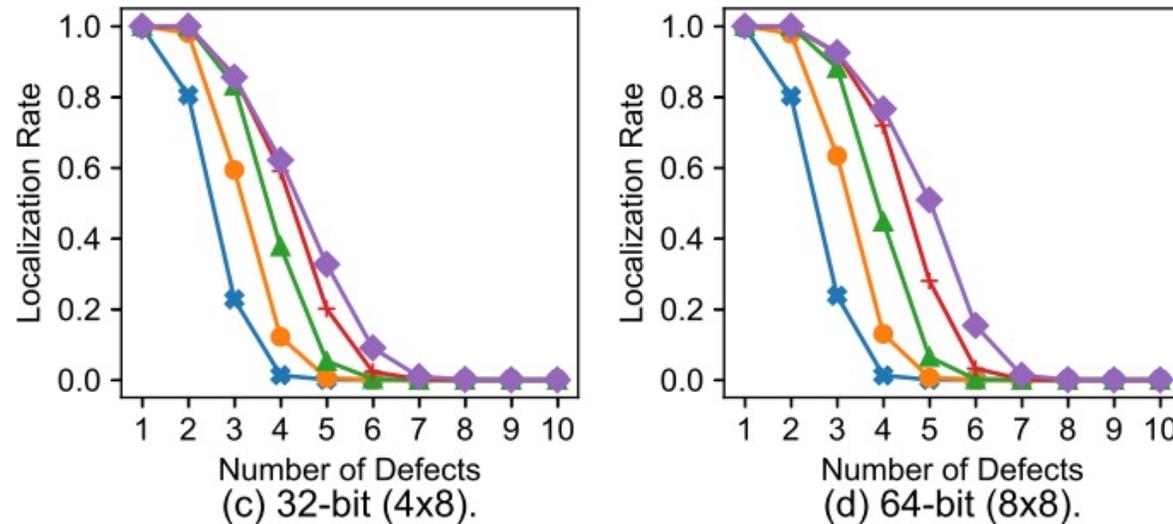
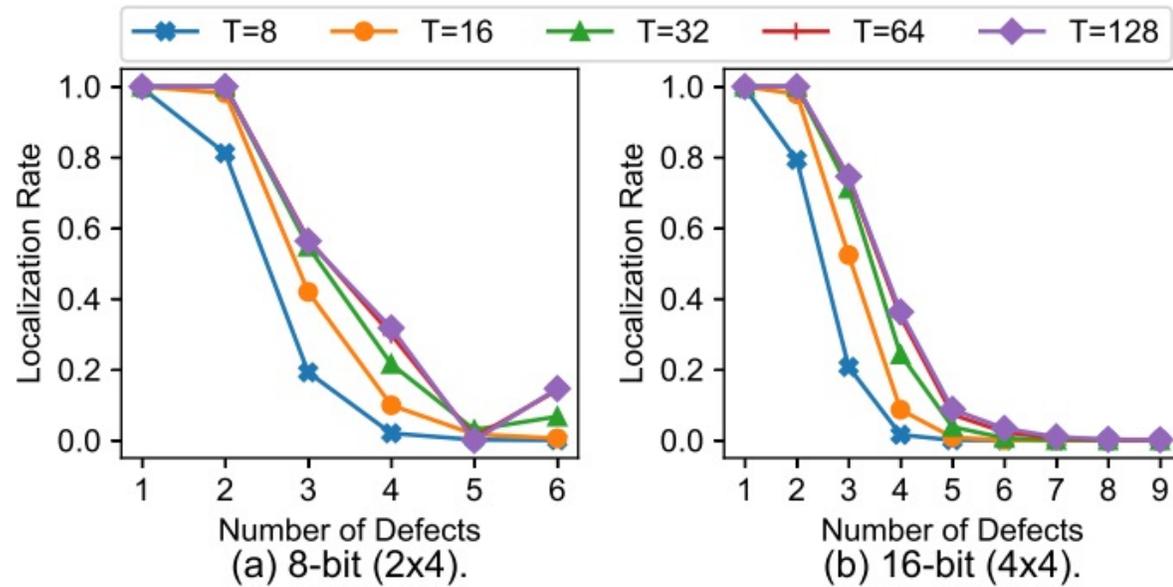
Width	8, 16, 32, and 64-bit
PPC	2x4, 4x4, 4x8, and 8x8
Statistical detection period (T)	8, 16, 32, 64 and 128 cycles
Number of defects	1-6 (2x4) 1-9 (others)
Detect type	Open/short

# Statistical detection: Localization rate w. false positive



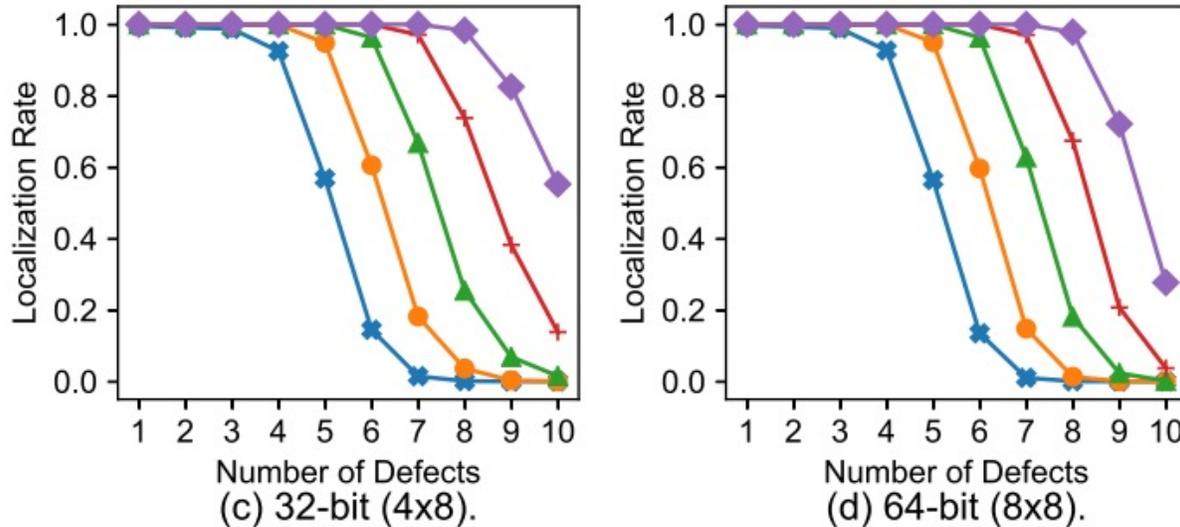
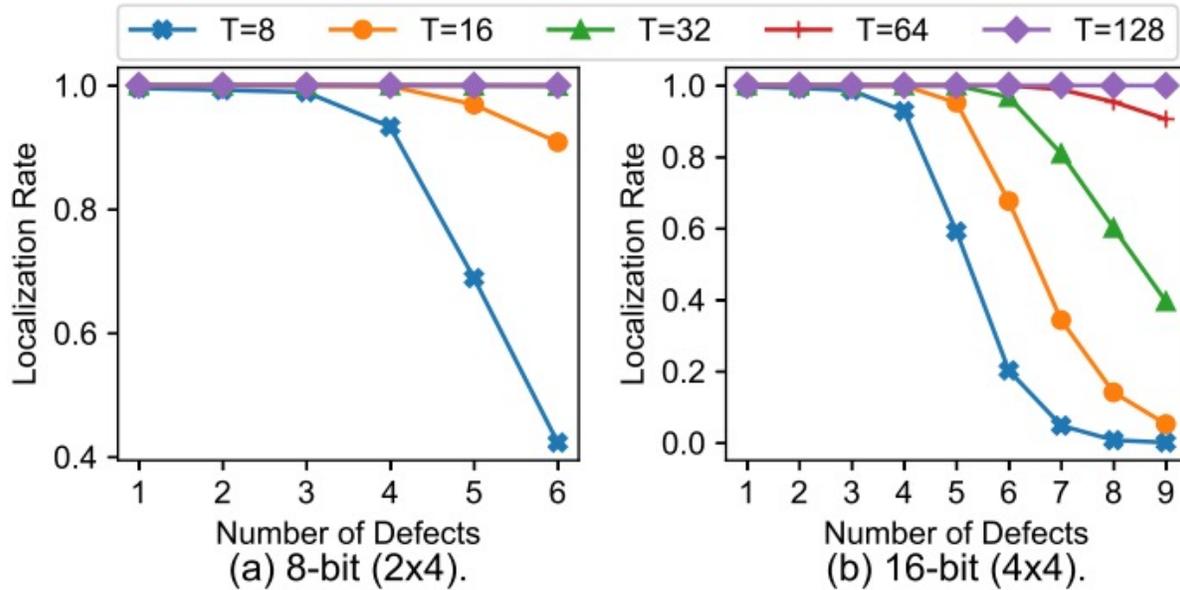
- Thank to the greedy localization, statistical detection could capture 4+ defects in most case:
- However, there are still false positive case:
  - TSV is healthy
  - Statistical detection marks is “suspicious:”

# Statistical detection: Localization rate w/o false positive



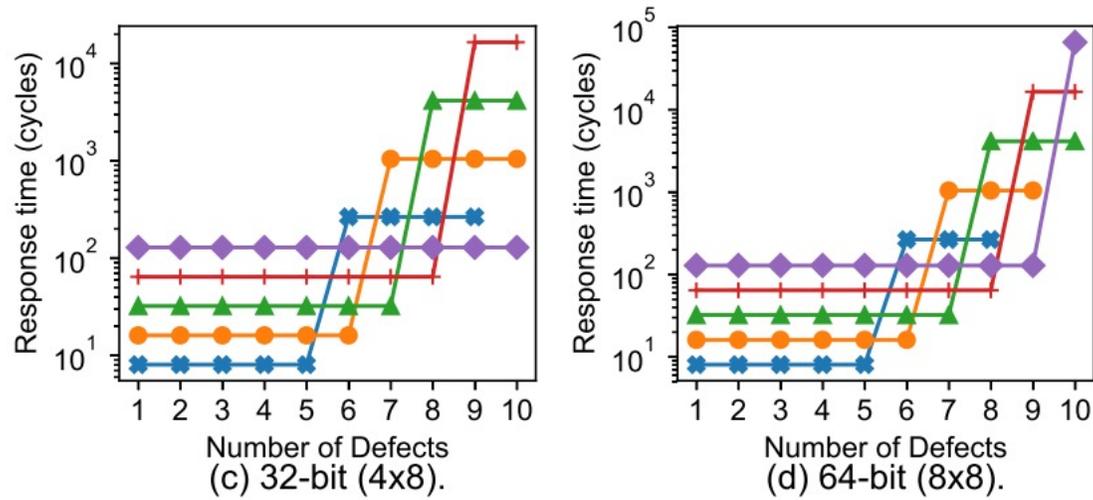
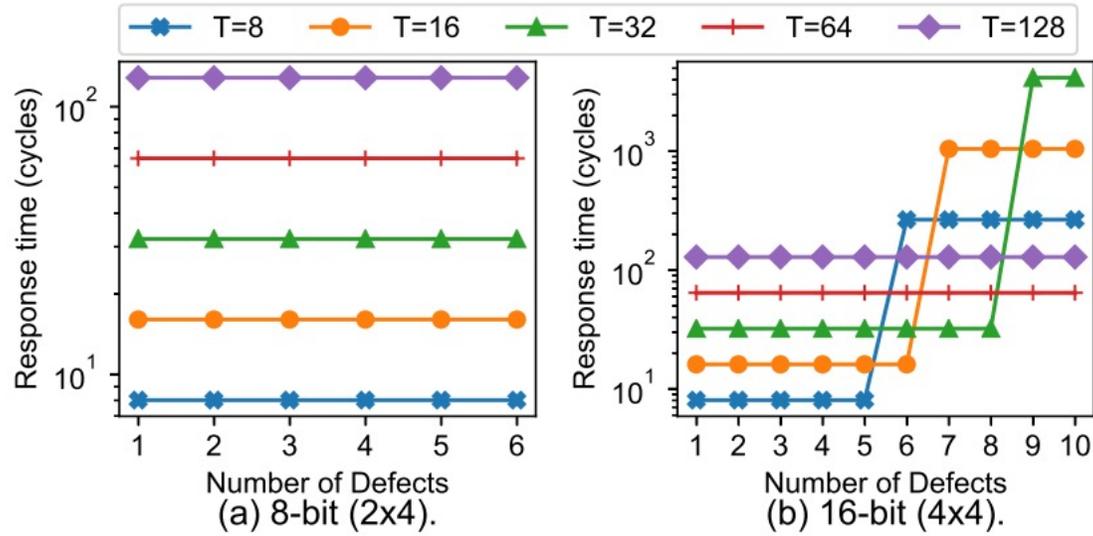
- If we consider false positive as incorrect test result
  - Statistical detection could localize two faults
  - PPC itself only localizes one fault
- This is 2x improvement from statistical detection

# Isolation-and-Check

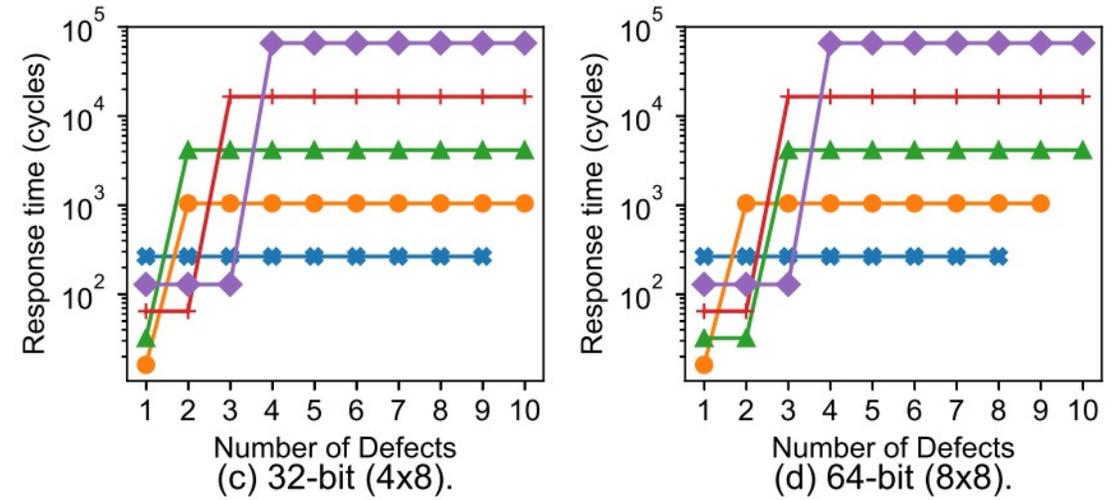
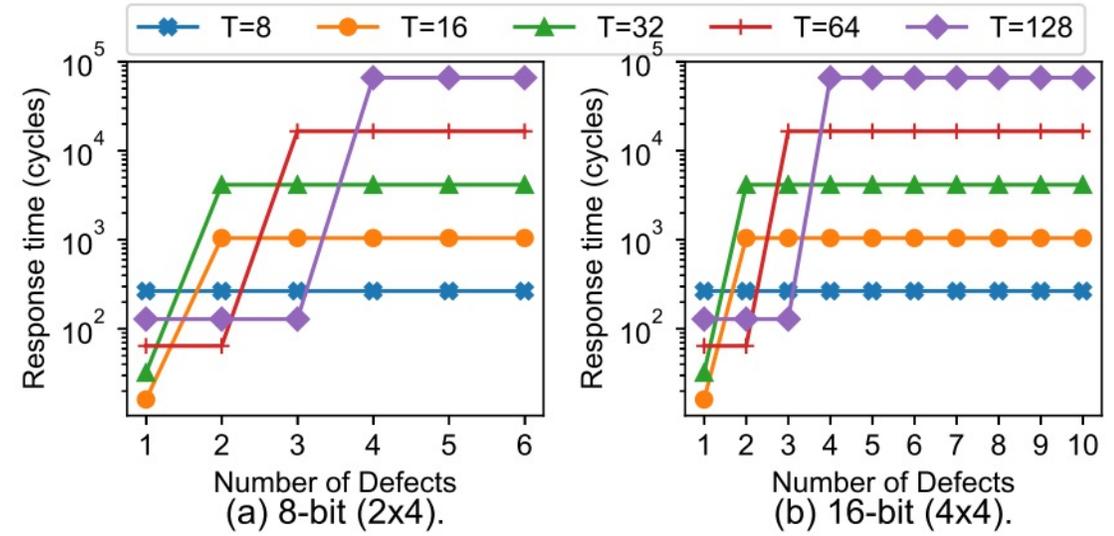


- By re-checking, OCTT remove the false positive cases.
- OCTT could localize up to 100% of 6 cases.
- With 8-9 case, the localization rate is still high (>90%).

# Execution Time



Best-case Execution Time



Worst-case Execution Time

# Hardware complexity

Scheme		Tech. (nm)	k (bit)	n (bit)	Area Cost ( $\mu m^2$ )		Latency (ns)		Power ( $\mu W$ )	
					Encoder	Decoder	Encoder	Decoder	Encoder	Decoder
Hamming [14]		45	32	39	94.1640	234.8780	0.55	1.12	30.0831	96.2898
SECDED [19]		45	32	40	111.7200	253.7640	0.60	1.44	36.9622	103.1422
SEC-DAEC [30] <sup>1</sup>		45	32	39	322	1902	0.53	1.33	-	-
TAEC [31] <sup>1</sup>		45	32	40	264	2628	0.45	1.32	-	-
PPC(4 × 8)		45	32	45	76.6080	187.2640	0.30	0.68	43.0272	129.4174
<b>OCTT</b>	Total	45	32	45	130.3400	2161.2500	0.39	0.72	48.639	$1.04 \times 10^3$
	PPC(4 × 8)	45	32	45	130.3400	327.4460	-	-	48.639	198.424
	Stat_det	45	32	45	-	751.1840	-	-	-	408.621
	Isol_Check	45	32	45	-	1016.3860	-	-	-	387.056

<sup>1</sup> We use the area optimization and lowest area cost design since our design is optimized for area cost.

- Our area cost is significantly higher since we have registers for storing suspicious position.
- However, thank to row/column check, the latency (max frequency) is still promising.

› Introduction

› Proposed architecture

› Evaluation

› Conclusion

# Conclusion

- This work presents a non-preemptive and online test for TSV:
  - **Statistical detection** to capture all possible suspicious TSVs
  - **Isolation and check** to capture more suspicious TSVs and remove false positive cases.
- The results show that **Statistical detection** improves 2x the localization rate of the baseline ECC while **Isolation and check** improves 6x.
- The hardware complexity of this work is reasonable since it could provide high localization rate.
- Future works:
  - Integrate into 3D-Network-on-Chip to study and area/power overhead.
  - Extract the execution time.

# Reference

- Y. Zhao et al., “Online Fault Tolerance Technique for TSV-Based 3-DIC,” IEEE Trans. VLSI Syst., vol. 23, no. 8, pp. 1567–1571, 2015.
- M. Cho et al., “Design method and test structure to characterize and repair TSV defect induced signal degradation in 3D system,” in Proc. Int. Conf. on Comput.-Aided Des., 2010, pp. 694–697.
- L. Jiang et al., “On effective through-silicon via repair for 3-D-stacked ICs,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 32, no. 4, pp. 559–571, 2013.
- J. Park et al., “Fresh: A new test result extraction scheme for fast tsv tests,” IEEE Trans Comput -Aided Design Integr Circuits Syst, vol. 36, no. 2, pp. 336–345, 2017.
- I. Jani et al., “Bists for post-bond test and electrical analysis of high density 3d interconnect defects,” in 2018 IEEE 23rd European Test Symposium (ETS). IEEE, 2018, pp. 1–6.
- L.-C. Li et al., “An efficient 3d-ic on-chip test framework to embed tsv testing in memory bist,” in Design Automation Conference (ASP-DAC), 2015 20th Asia and South Pacific. IEEE, 2015, pp. 520–525.
- C. Grecu et al., “Testing network-on-chip communication fabrics,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 26, no. 12, p. 2201, 2007.
- C. Liu et al., “Reuse-based test access and integrated test scheduling for network-on-chip,” in Proceedings of the conference on Design, automation and test in Europe: Proceedings. European Design and Automation Association, 2006, pp. 303–308.
- A. M. Amory et al., “A scalable test strategy for network-on-chip routers,” in IEEE International Conference on Test, 2005., Nov 2005, pp. 9 pp.–599.
- D. Xiang et al., “Multicast-based testing and thermal-aware test scheduling for 3d ics with a stacked network-on-chip,” IEEE Trans Comput, vol. 65, no. 9, pp. 2767–2779, Sep. 2016.
- J. Wang, M. Ebrahimi, L. Huang, X. Xie, Q. Li, G. Li, and A. Jantsch, “Efficient design-for-test approach for networks-on-chip,” IEEE Transactions on Computers, vol. 68, no. 2, pp. 198–213, 2019.

Thank you for your attention!