A Physical-Location-Aware Fault Redistribution for Maximum IR-Drop Reduction

Fu-Wei Chen, Shih-Liang Chen, Yung-Sheng Lin
and TingTing Hwang

National Tsing Hua University
Department of Computer Science

Outline

- Introduction
- Motivation
- Design Flow
- Experimental Results
- Conclusion

Outline

- **Introduction**
- Motivation
- Design Flow
- Experimental Results
- Conclusion

Introduction

- IR-drop effect increases up to 16% during at-speed test as compared normal mode [N. Ahmed, ICCAD’06]
  - Test vector designed to generate switching
  - Test vector compression
- IR-drop → Long path delay → Delay fault

Previous Work

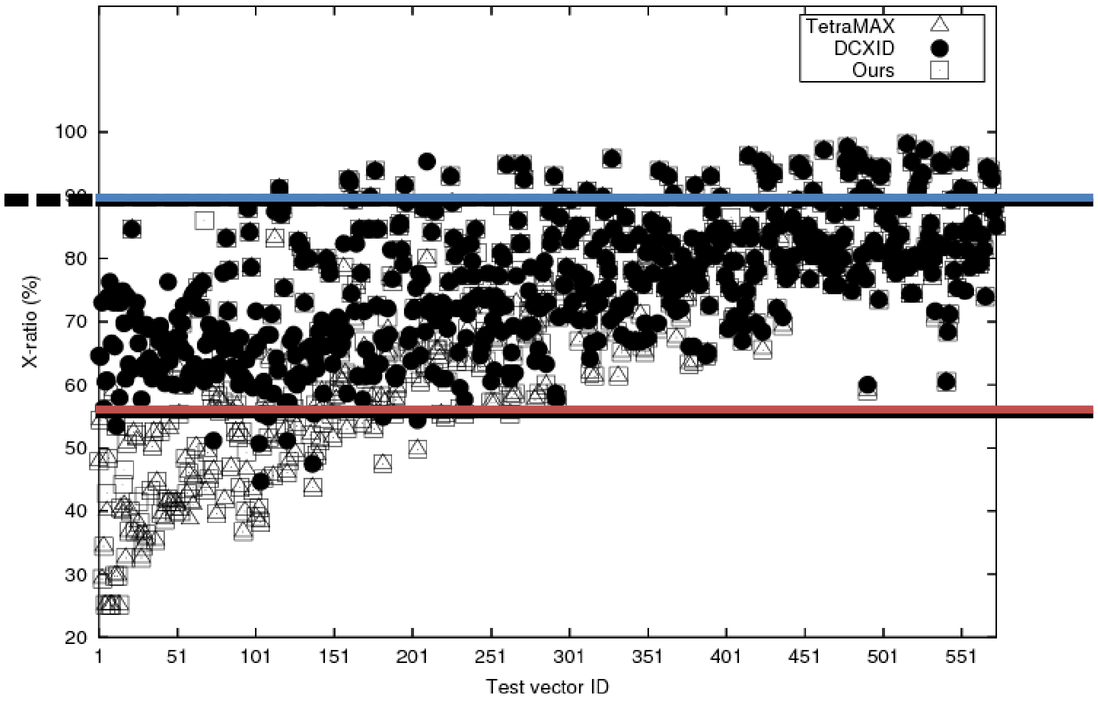
- X-filling to reduce IR-drop [4,11,14,15]
  - Reduce IR-drop effect in at-speed testing
  - Depend on the number and characteristic of X-bit distribution
  - Sensitization paths propagate through the same hot region

- X-identification to redistribute X-bit [10]
  - Redistribute X-bits evenly in test vector
  - Reduce IR-drop effect after X-filling

Outline

- Introduction
- **Motivation**
- Design Flow
- Experimental Results
- Conclusion

Motivation: Even Distribution of X-Bit Is Not Enough



TetraMAX

DCXID

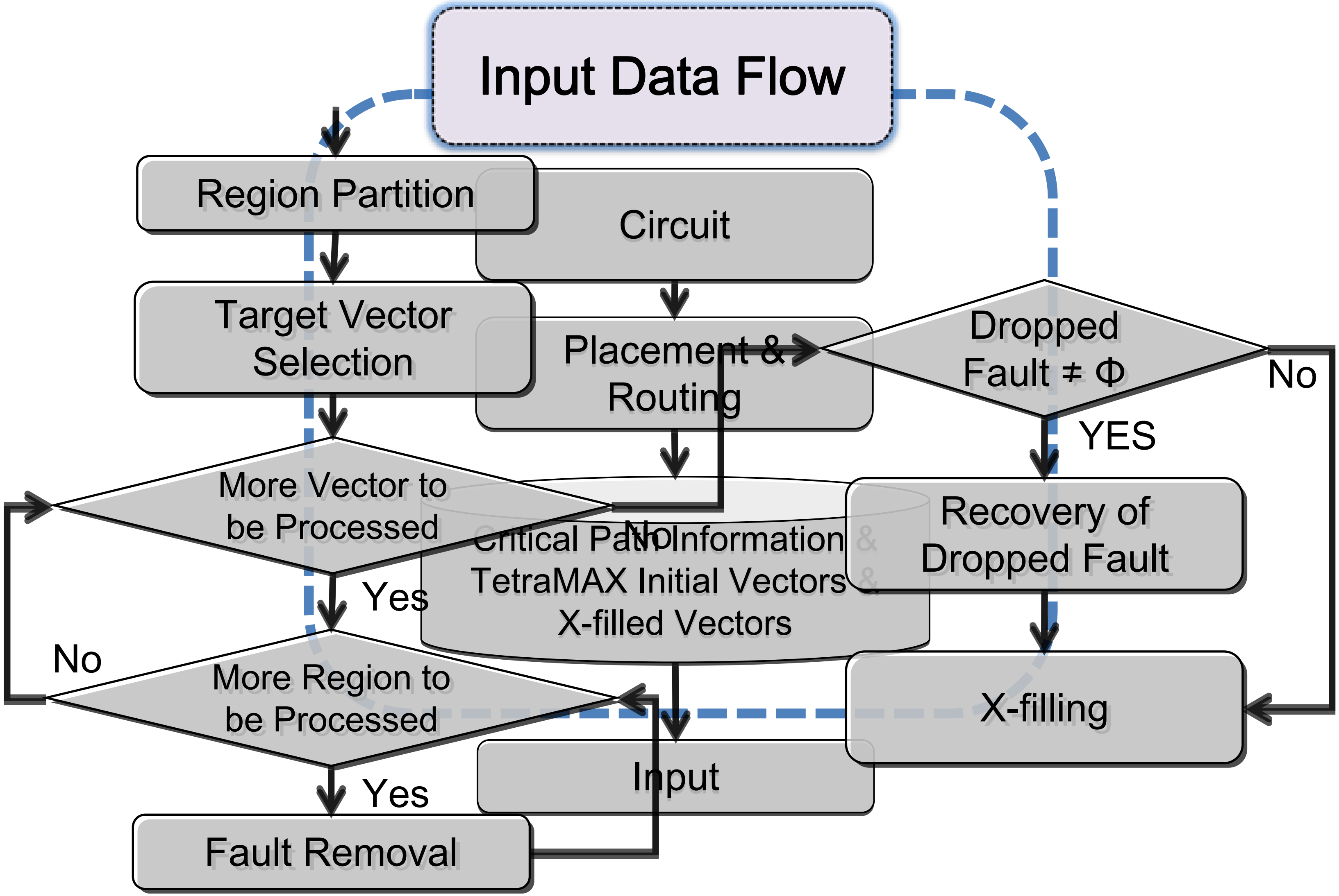
Ours

DCXID: Set X-bits evenly in all test vectors
Ours: Set X-bits in hot region

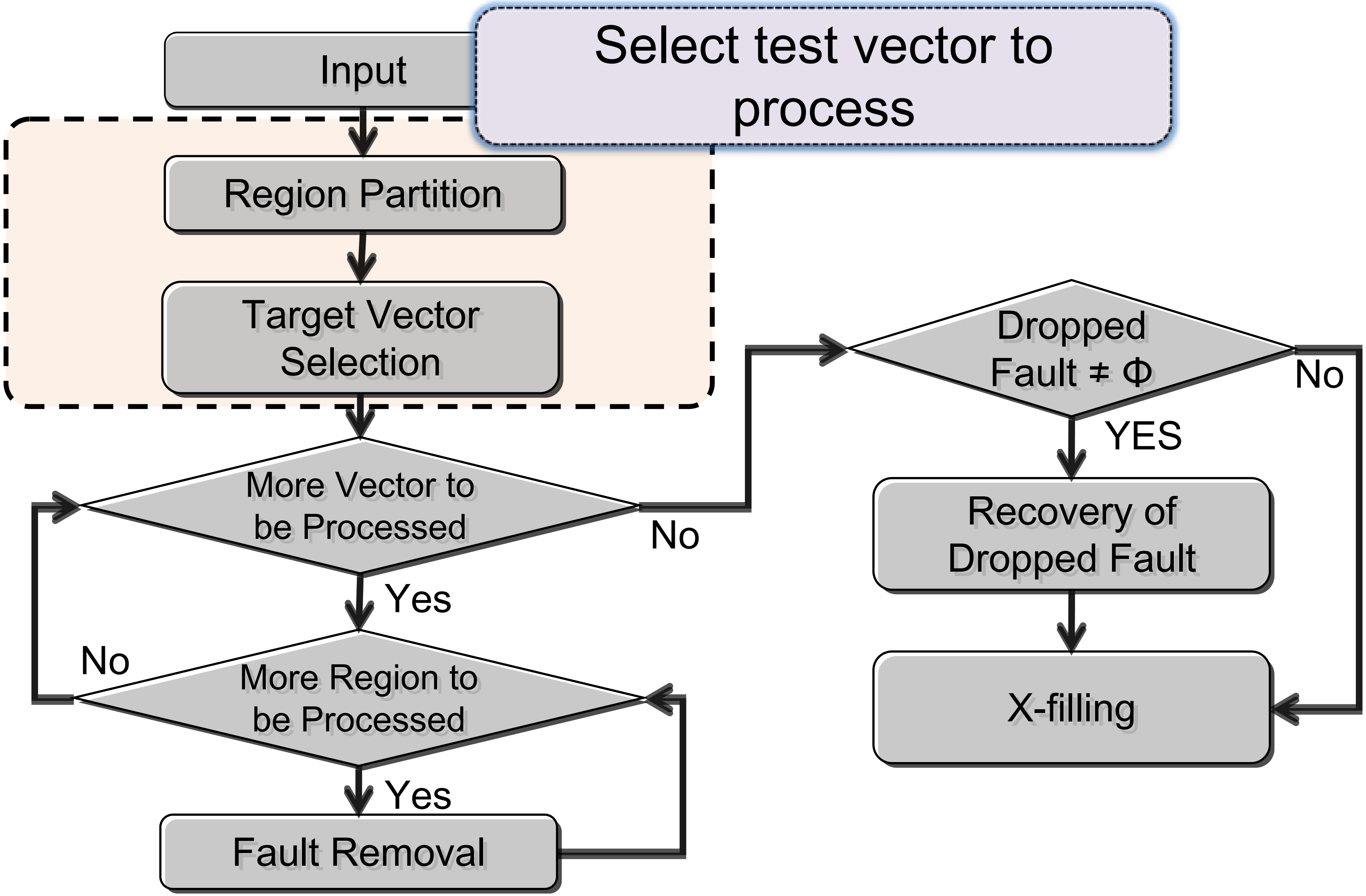
Outline

- Introduction
- Motivation
- **Design Flow**
- Experimental Results
- Conclusion

Design Flow

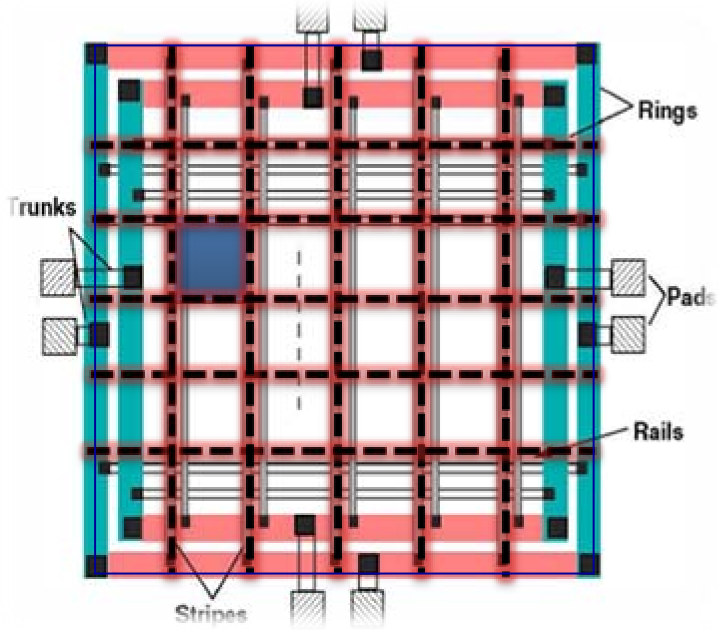

Input Data Flow
Region Partition
Circuit
Target Vector Selection
Placement & Routing
Dropped Fault ≠ Φ
No
YES
More Vector to be Processed
No
Critical Path Information & TetraMAX Initial Vectors & X-filled Vectors
Recovery of Dropped Fault
Yes
No
More Region to be Processed
X-filling
Input
Yes
Fault Removal

Design Flow



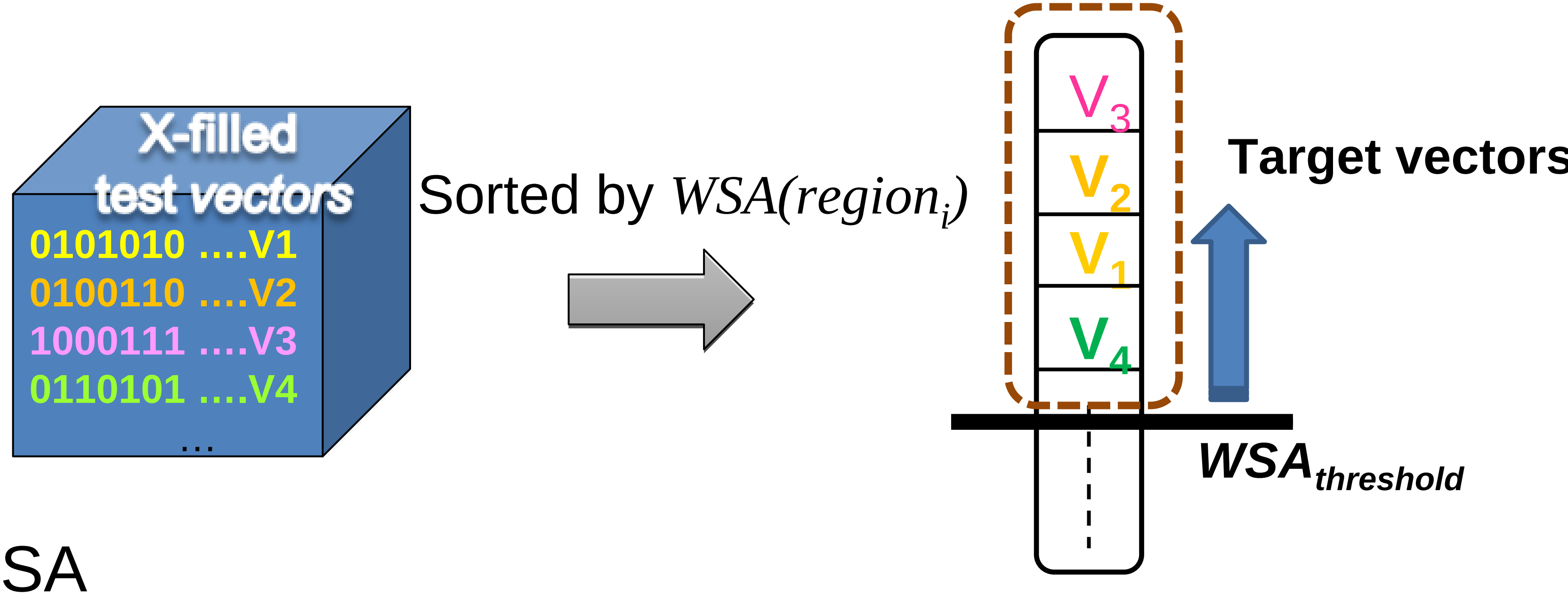
Select test vector to process

Region Partition



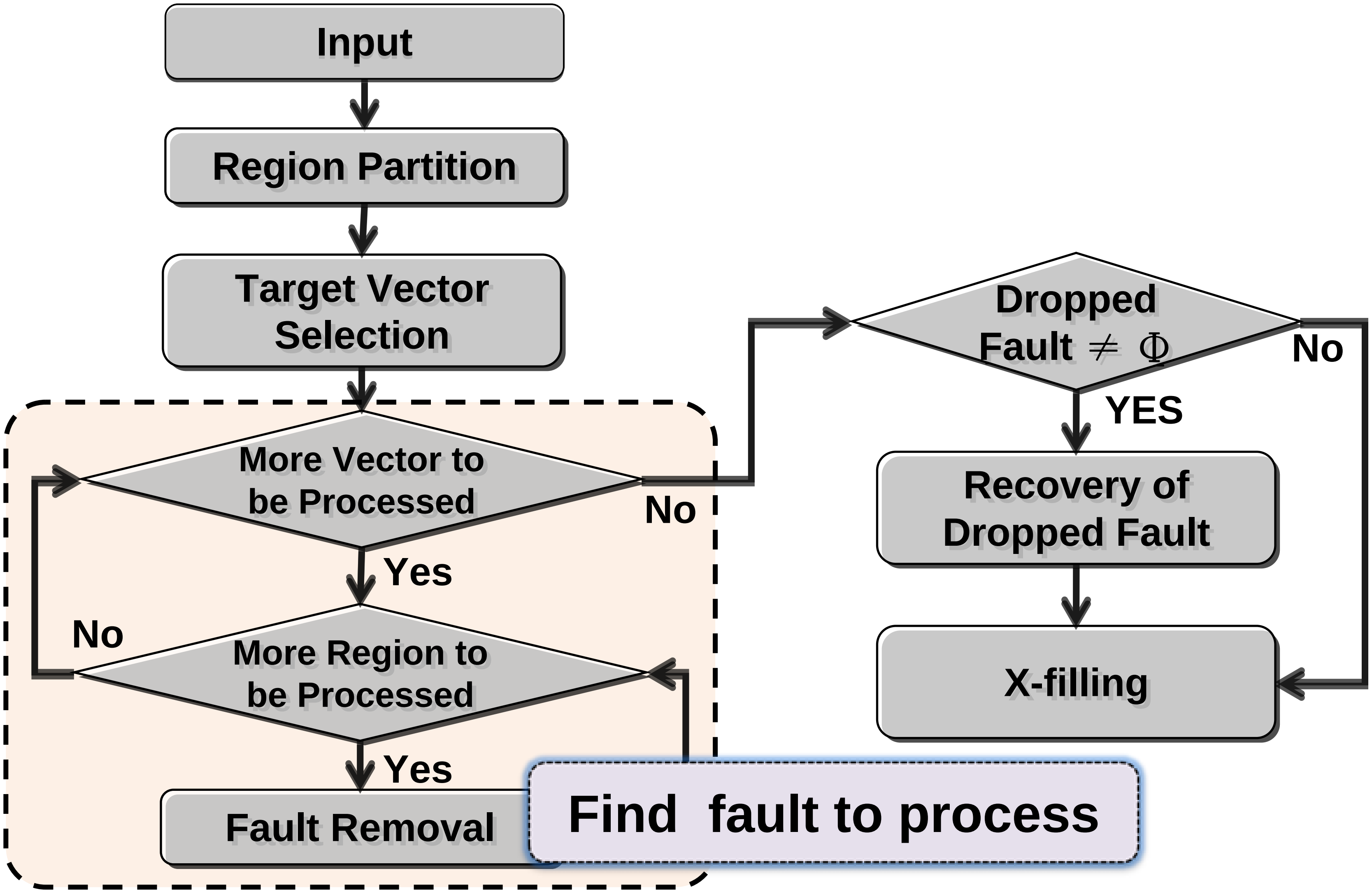
- All components assigned to region
- Each region surrounded by stripes and rails
- The power rings placed around the core chip.
- Four VDD and VSS pads inserted to the respective rings

Target Vector Selection



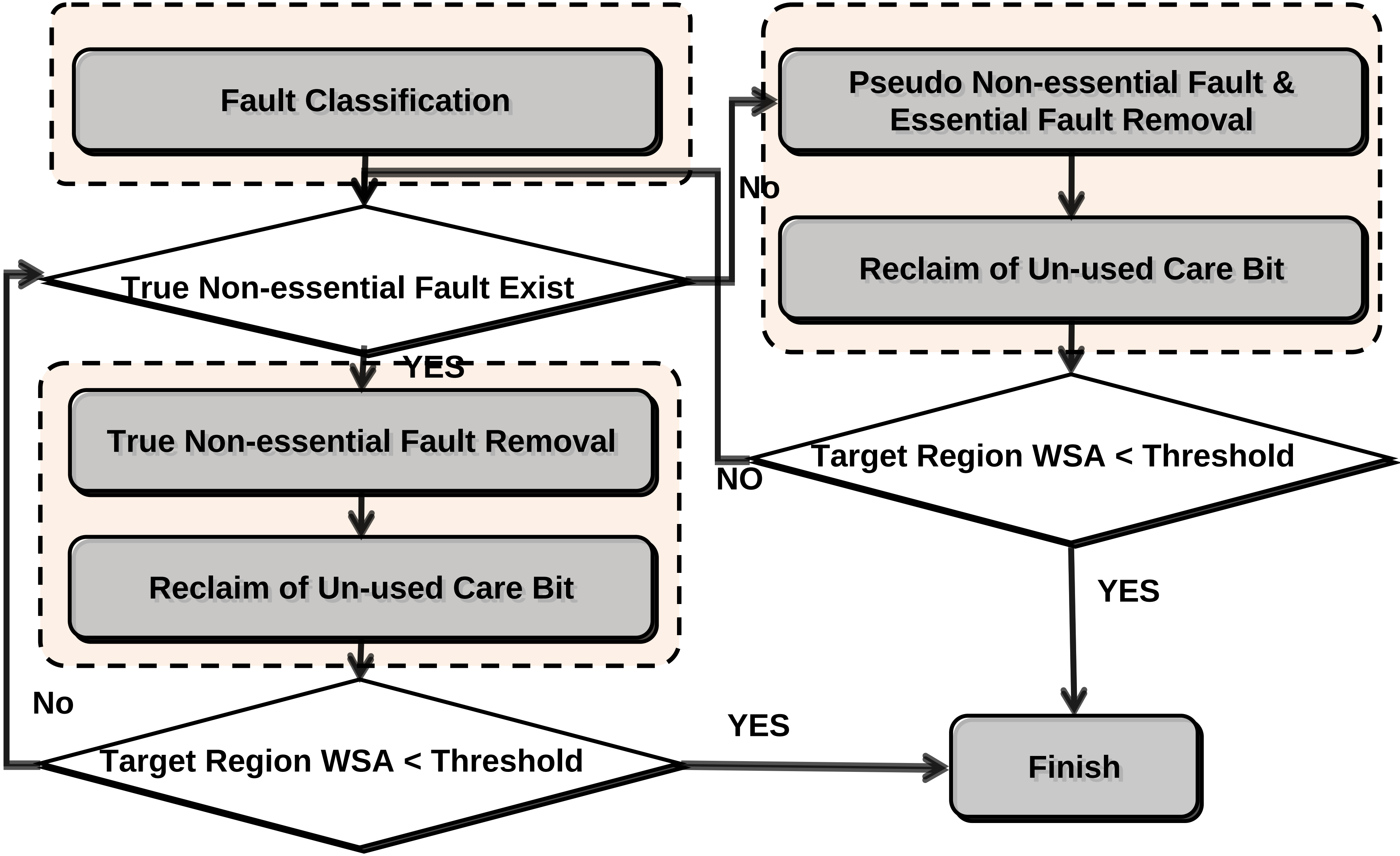
- WSA
  - $WSA(region_i)$ represents **IR-drop effect** of each region
  - $WSA(region_i) \propto$ Switching activity impact
  - $WSA(region_i)$ of target region is **equal or larger than** $\boldsymbol{WSA_{threshold}}$

Design Flow

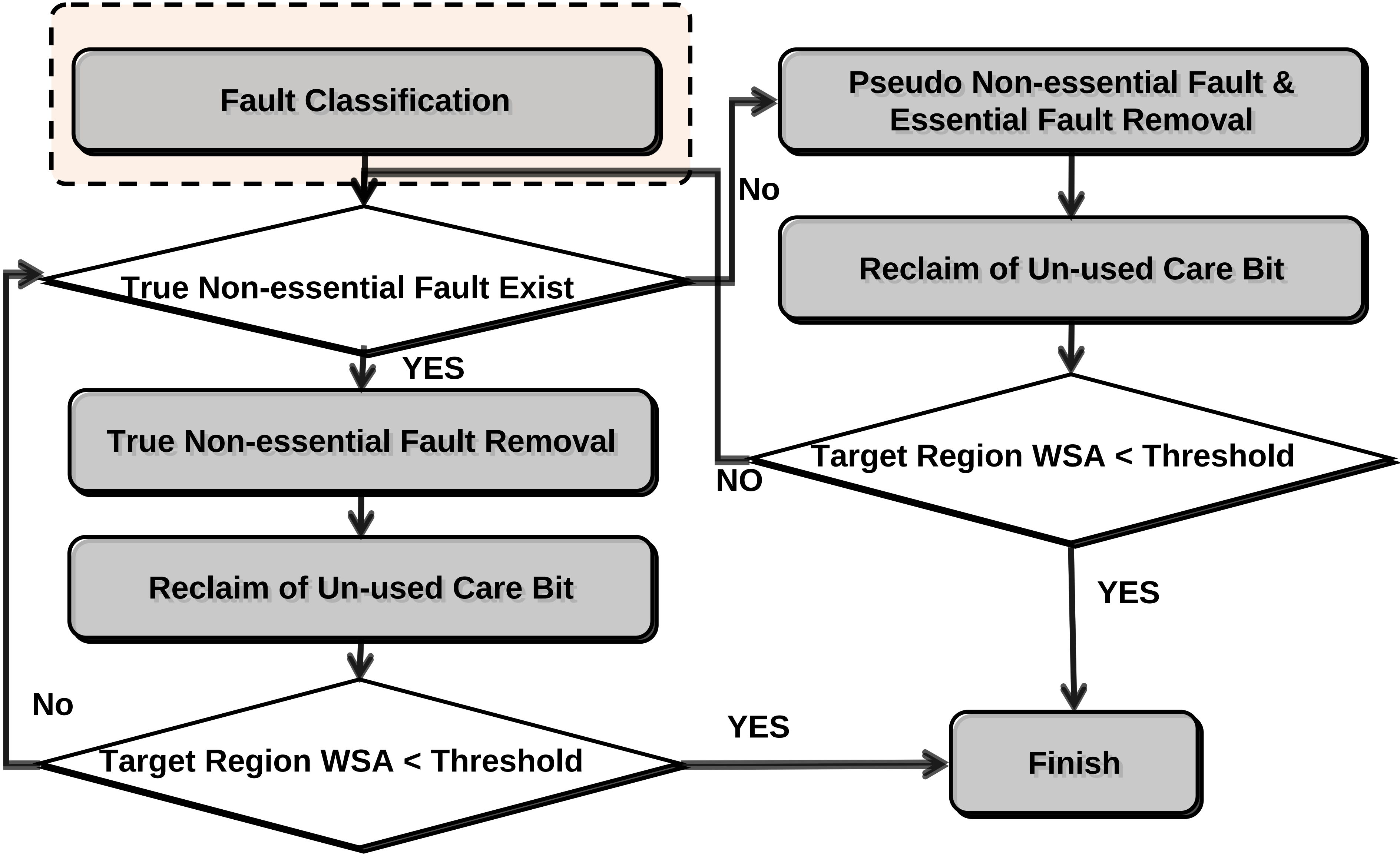


Find fault to process

Fault Removal Flow

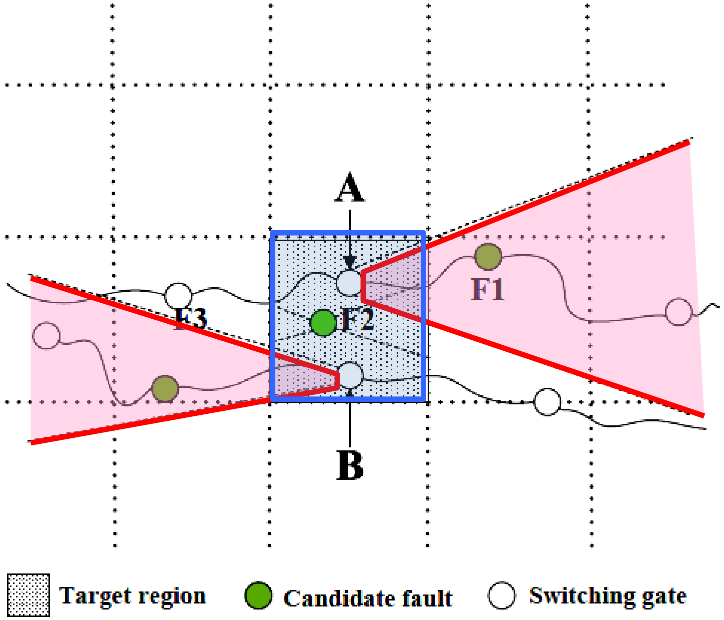


Fault Removal Flow



Fault Removal

- Candidate fault
  - **Be in** target region or **propagate through** target region (based on physical-location information)
  - Impact of IR-drop
    - *WSA* in target region
  - Fault coverage



Fault Classification

- Candidate fault classification
  - Essential fault
    - Covered by only one test vector
  - Non-essential fault
    - Covered by more than one test vector

Fault coverage issue

Fault Classification

- Candidate fault classification
  - Essential fault
    - Covered by only one test vector
  - Non-essential fault
    - Covered by more than one test vector
    - **True** non-essential fault
      - No other **essential faults** are on the propagation path
    - **Pseudo** non-essential fault
      - Essential fault is on the propagation path

Example of Non-essential Fault Classification

- Construct **fault table** for each test vector

	bit_1	bit_2	bit_3	bit_4
$fault_1$	1	0	1	0
$fault_2$	1	1	1	0
$fault_3$	0	1	0	1

care-bits$_1$:{ 1, 3}

care-bits$_2$:{ 1, 2, 3}

care-bits$_3$:{ 2, 4}

essential fault : $fault_1$

non-essential fault : $fault_2$ and $fault_3$

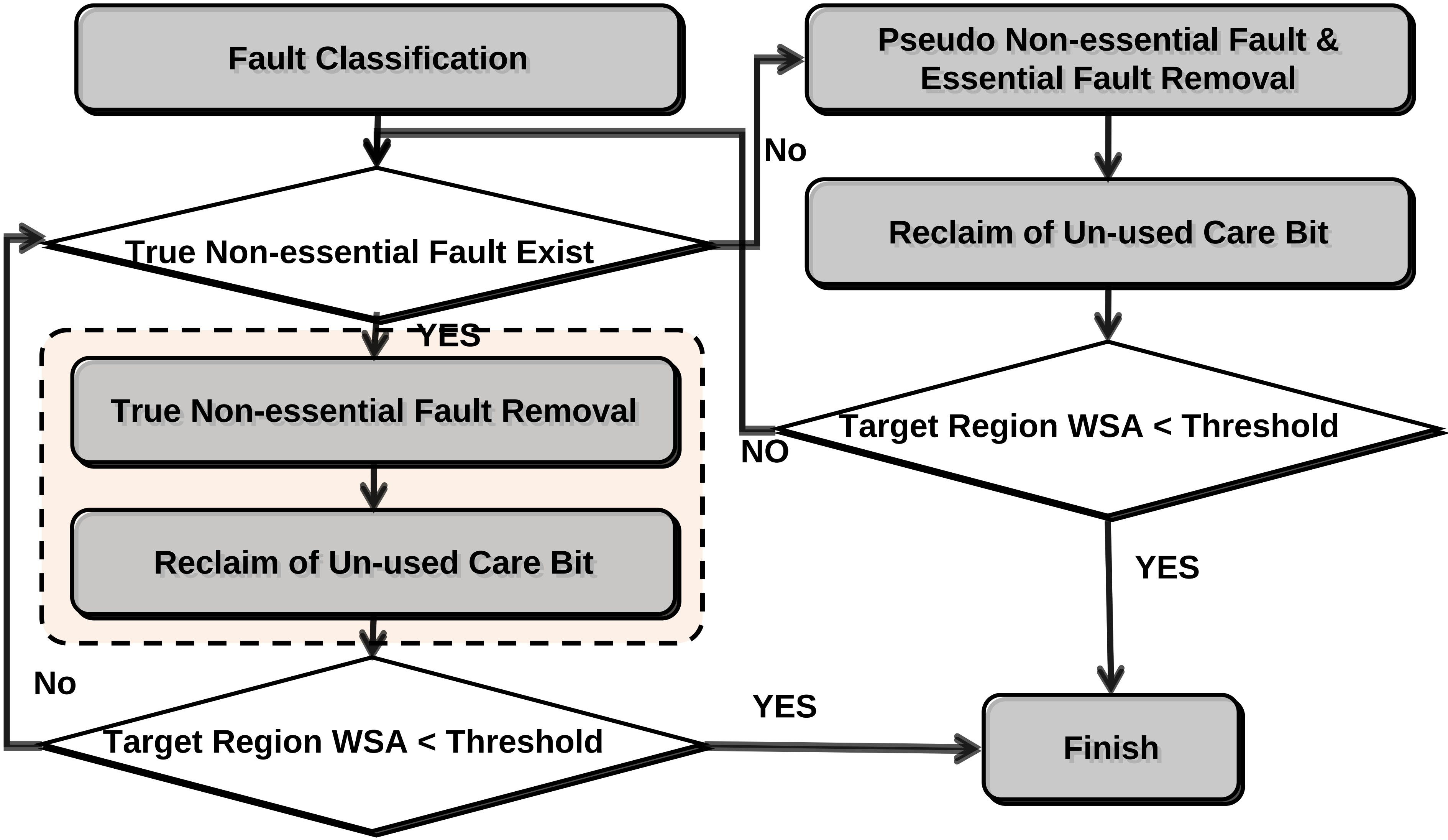
$$care\text{-}bit_3 \cap care\text{-}bit_1 = \Phi$$

True non-essential fault

$$care\text{-}bit_2 \cap care\text{-}bit_1 = \{1\} \neq \Phi$$

Pseudo non-essential fault

Fault Removal Flow



Fault Classification

True Non-essential Fault Exist

YES

True Non-essential Fault Removal

Reclaim of Un-used Care Bit

No

Target Region WSA < Threshold

YES

No

Pseudo Non-essential Fault & Essential Fault Removal

Reclaim of Un-used Care Bit

Target Region WSA < Threshold

NO

YES

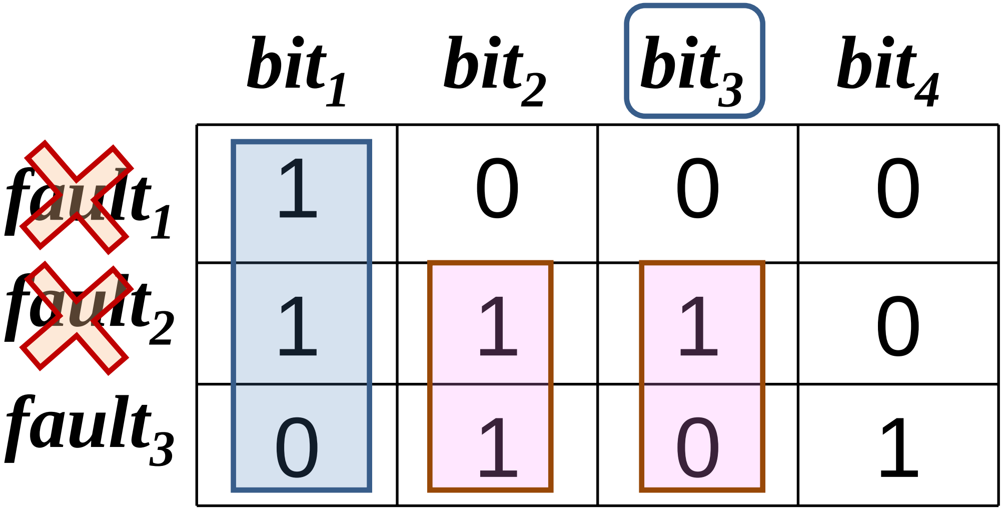
Finish

True Non-essential Fault Removal

- Remove true non-essential fault to reduce *WSA*
  - *Removal_gain(target fault) = R_WSA* x *#detectable_vector*
    - *R_WSA*: the reduction of *WSA* in region after removal *target fault*
    - *#detectable_vector*: the number of test vector that covers this *target fault*

  - Remove *target fault* with **maximum** ***Removal_gain*** until no true non-essential fault or *WSA* is less than threshold

Reclaim Un-used Care Bit

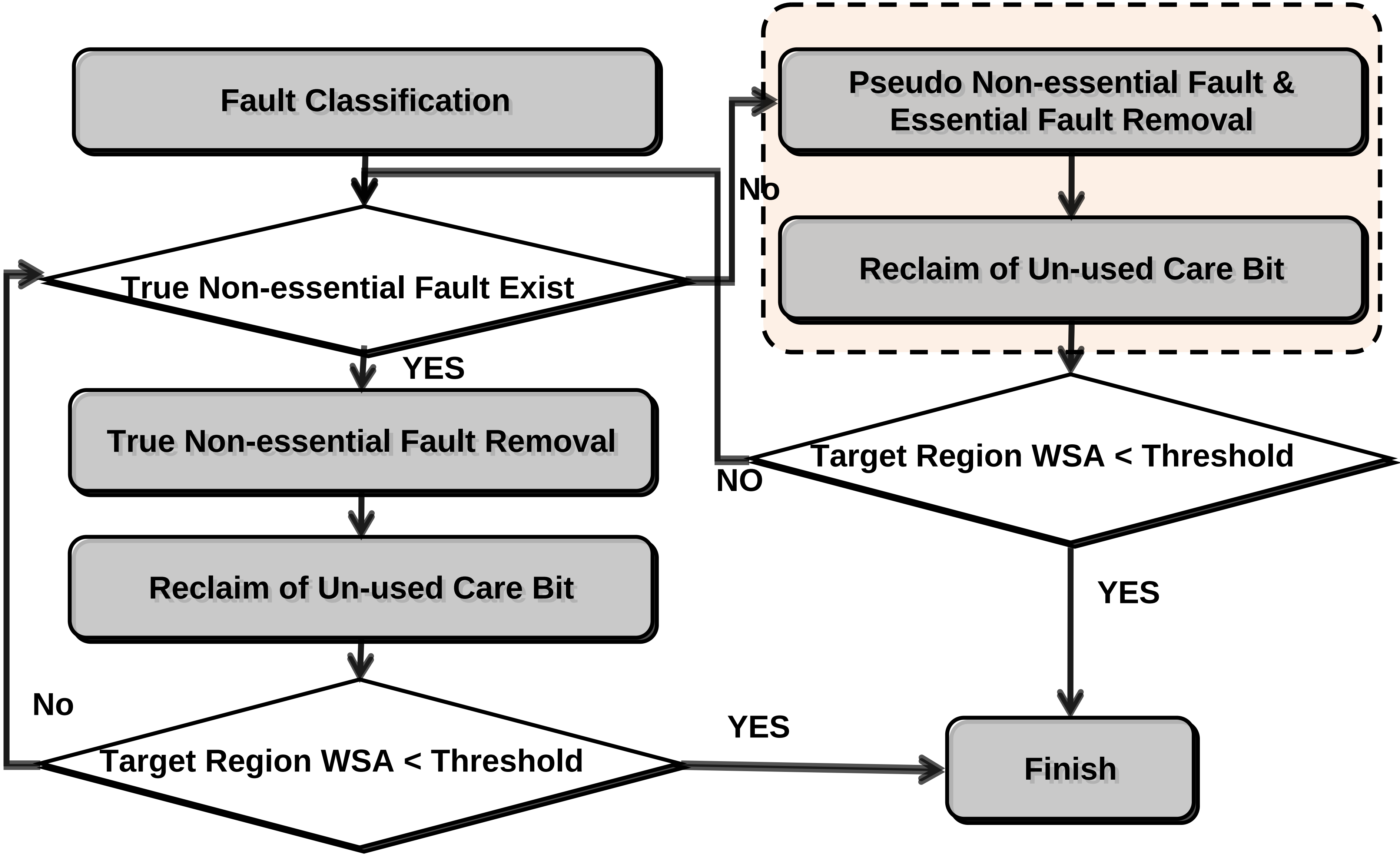


	bit_1	bit_2	bit_3	bit_4
~~$fault_1$~~	1	0	0	0
~~$fault_2$~~	1	1	1	0
$fault_3$	0	1	0	1

$fault_1$ is to be removed → $fault_2$ will be removed

Reclaim un-used care-bit bit_3 to X-bit

Fault Removal Flow



Fault Classification

True Non-essential Fault Exist

YES

True Non-essential Fault Removal

Reclaim of Un-used Care Bit

Target Region WSA < Threshold

No

YES

No

Pseudo Non-essential Fault & Essential Fault Removal

Reclaim of Un-used Care Bit

Target Region WSA < Threshold

NO

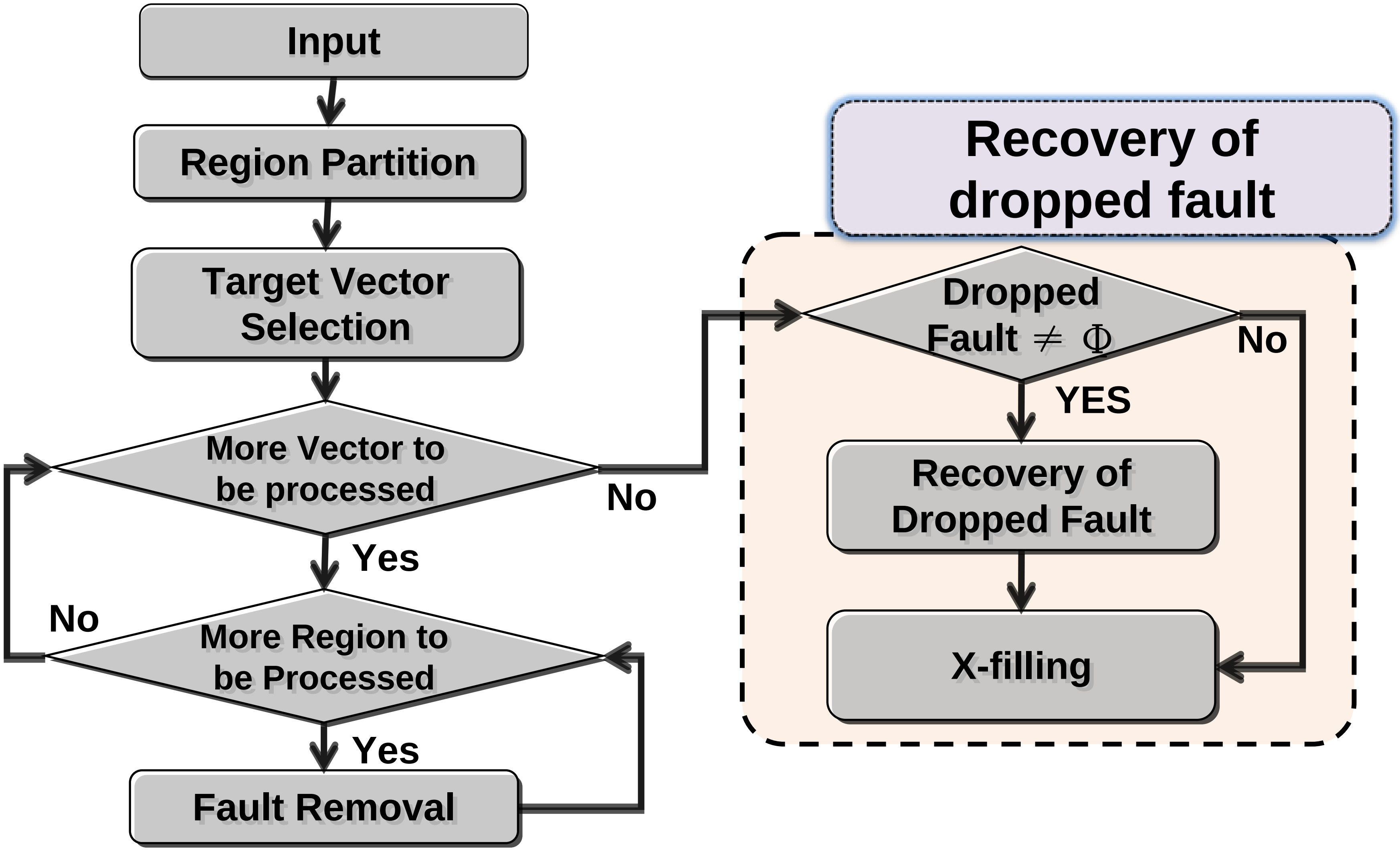
YES

Finish

Pseudo Non-essential Fault & Essential Fault Removal

- Reduction of WSA by setting care-bit to X
  - Gain function: $bit\_gain(cb) = \frac{R\_WSA}{\#\ fault}$
    - R_WSA : the reduction of WSA in region after setting the care-bit, cb, to X
    - $\#fault$: the number of removed fault if cb sets to X
  - Set one care-bit to X-bit with **maximum *bit_gain***

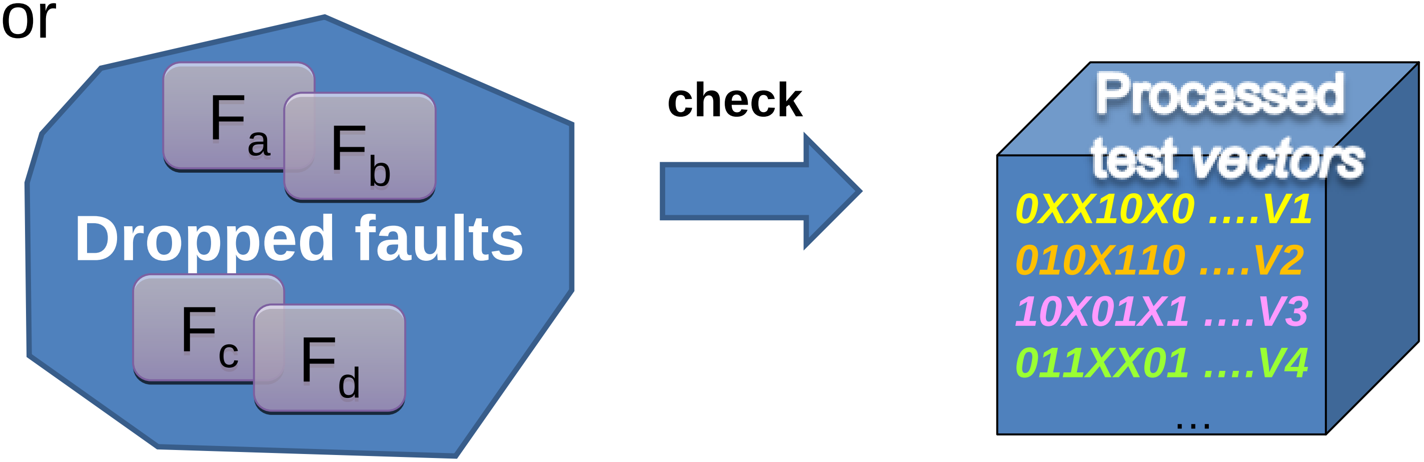
Design Flow



Recovery of dropped fault

Recovery of Dropped Fault

- After essential & pseudo non-essential fault removal is performed
  - Dropped fault → Fault coverage
- Check if any test vector can cover a dropped fault without violating IR-drop constraint
  - Care-bits of the dropped fault are consistent with test vector



Dropped faults

F_a F_b F_c F_d

check

Processed test vectors

0XX10X0 ….V1

010X110 ….V2

10X01X1 ….V3

011XX01 ….V4

…

Outline

- Introduction
- Motivation
- Design Flow
- **Experimental Results**
- Conclusion

Experimental Setup

- Benchmark: ITC’99
- Technology library: TSMC 90 nm
- Synthesis tool: Synopsys Design Compiler
- APR tool: SoC Encounter(SoCE)
- ATPG tool: TetraMAX
- IR-drop Simulator: RedHawk

Results of X-filling

Circuit Name	Random-filled		PB-based X-filling[4]			
	MAX.	Avg.	MAX.	(%)	Avg.	(%)
b14	5.66	0.798	5.66	0.00	0.6805	14.72
b15	3.17	0.12	2.87	9.30	0.1078	10.14
b17	4.32	0.1158	3.88	10.24	0.1039	10.24
b18	4.1	0.1151	3.67	11.82	0.1023	12.50
b19	3.81	0.1208	3.52	10.08	0.1048	15.27
b20	5.13	0.5132	4.98	3.01	0.4581	10.75
b21	7.81	0.4389	7.71	1.29	0.3907	10.99
b22	5.49	0.699	5.45	0.72	0.6157	11.92

- Improvements of MAX. *WSA* are **not significant (at most *3.01%*)**
- Select b14, b20, b21 and b22 as **difficult circuits**

Comparison of Maximum WSA for Difficult Circuits

Circuit Name	**MAX. *WSA***			**Improvement(%)**		**Overhead**
	TetraMAX	DCXID	Ours	DCXID	Ours	#DF(%)
b14	5.66	5.66	5.08	0.00	10.40	64(0.24)
b20	4.98	4.85	4.42	2.51	11.18	11(0.02)
b21	7.71	7.71	6.75	0.00	12.38	15(0.02)
b22	5.45	5.35	4.98	1.97	8.73	18(0.01)
				1.12	10.67	

- Comparison for maximum for each difficult circuit
- Maximum ***WSA*** reduction of our method is better than DCXID in difficult circuits
- #DF: the number of dropped faults

Comparison of Maximum IR-drop for Difficult Circuits

Circuit Name	MAX. IR-drop (*mV*)			Improvement (%)	
	TetraMAX	DCXID	Ours	DCXID	Ours
b14	314.3	312.7	284.3	0.5	9.56
b20	156.1	152.2	139.6	2.5	10.60
b21	110.6	109.0	97.4	1.45	11.89
b22	209.3	205.8	191.7	1.68	8.46
avg.				1.53	10.07

- Our method as compared with TetraMAX are 10.07% and 1.53%, respectively.

Outline

- Introduction
- Motivation
- Design Flow
- Experimental Results
- **Conclusion**

Conclusion

- A physical-location-aware X-identification method to redistribute faults
- An average of 8.54% more reduction of maximum IR-drop as compared to a previous work

Thank you