# Adding Temporal Redundancy to Delay Insensitive Codes to Mitigate Single Event Effects

Julian Pontes (FACIN-PUCRS)

Pascal Vivet (CEA-LETI)

Ney Calazans (FACIN-PUCRS)

FACIN-PUCRS(Brazil) & LETI-CEA (France)

# Motivation

- Advanced Tech Nodes Constraints
  - Signal Integrity and Process Variation
    - Solved at design time
  - Soft Errors
    - Not treated in standard flow
- Soft Errors in Asynchronous
  - Timing Deviations
    - Almost immune except for forks
  - A bit flip in control may stall handshake

# Our Objective

"Take advantage of m-of-n DI Codes to add temporal redundancy, allowing to detect and (eventually) correct soft errors"

# Outline

- Related Work
- SEE in QDI Pipelines Analysis
- TRDIC Proposal
- SEE Validation - Flow and Environment
- Results
- Conclusions and Ongoing Work

# Related Work

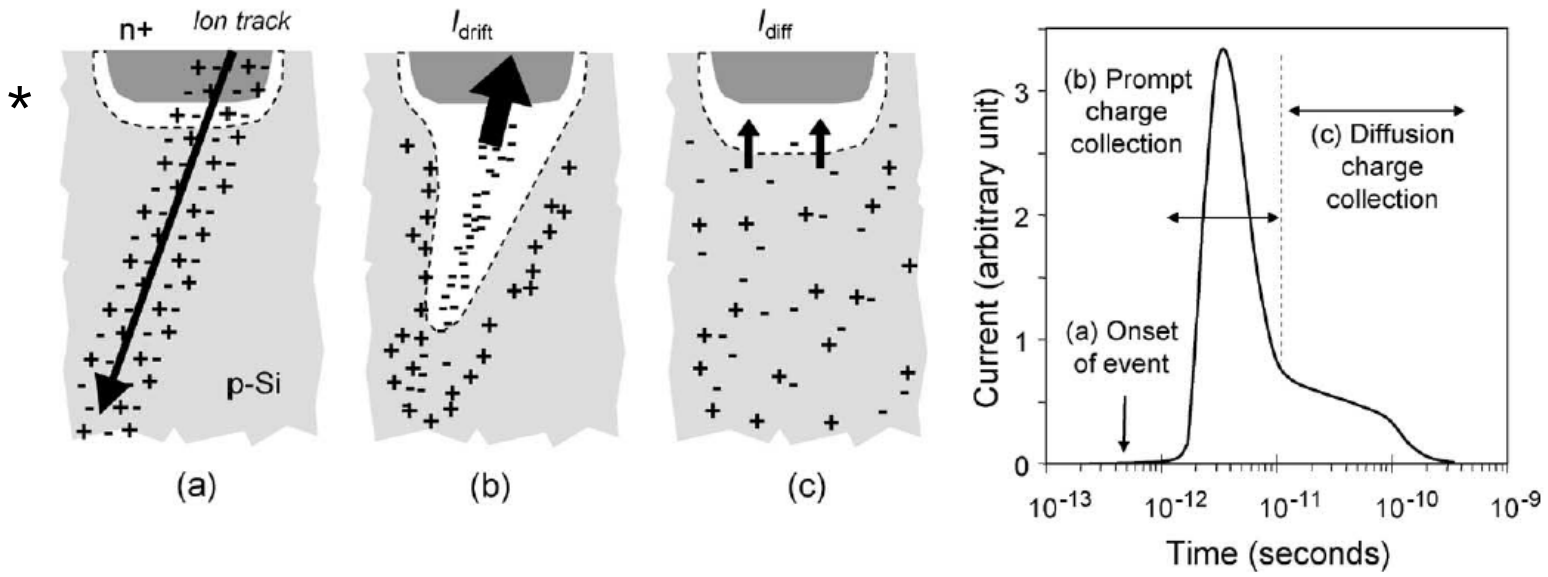
## Asynchronous Design Hardening Techniques

- Asynchronous x Synchronous (Asyncs are more robust!)
  - Bastos et al. (Microeletronics Reliability-2010)
  - Rahbaran and Steininger (IEEE Trans. on Dep. & Sec. Comp.-2009)
- Standard-cell level (Resizing to improve roibustness)
  - Bastos et al. (IOLTS-2010)
- Logic-level redundancy
  - Jang and Martin (ASYNC-2005) (Double-check, spatial redundancy)
  - Monet, Renaudin and Leveugle (IOLTS-05 ) (High area overhead or improved filtering capability)
- Pipeline level (Various design techniques against glitches)
  - Bainbridge and Salisbury (ASYNC-2009) (no error correction, though)
- New Delay Insensitive Codes (Hard to DI, due to validity det)
  - Agyekum and Nowick (DATE-2011)

# Outline

- Related Work
- SEE in QDI Pipelines Analysis
- TRDIC Proposal
- SEE Validation - Flow and Environment
- Results
- Conclusions and Ongoing Work

# SEE Physical Impact




*

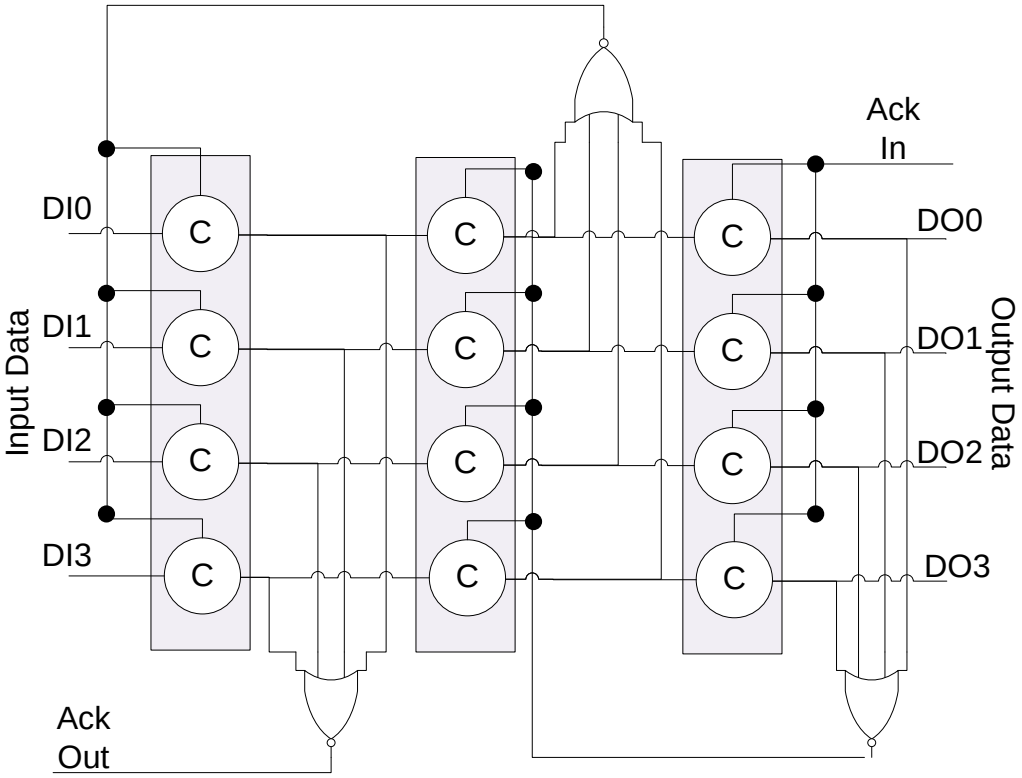
$$I(t) = I_0(e^{-t/\tau\alpha} - e^{-t/\tau\beta})$$

- Collection Time Constant of the Junction
- Time Constant for Initially Establishing the Ion Track

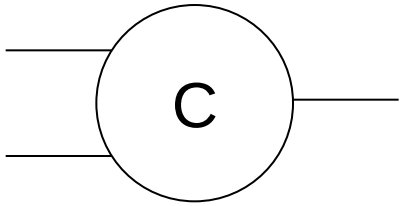
* - IBM experiments in soft fails in computer electronics(1978-1994) – 1996

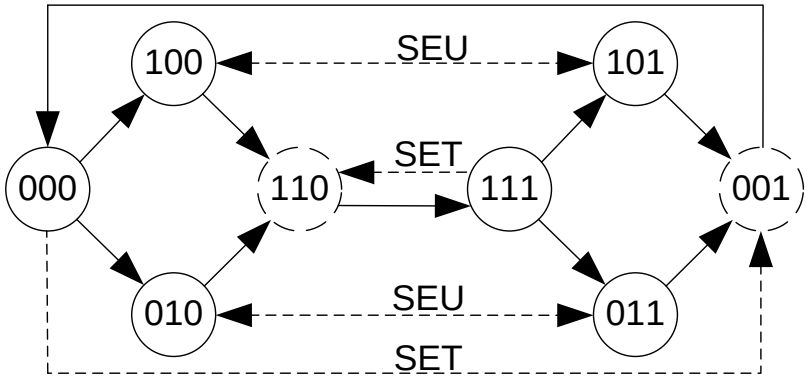
# SEE in QDI Pipelines

- QDI logic is almost immune to delay variations
  - Except for isochronic forks
- Bit flip may cause
  - Stall in handshake protocol
  - Erroneous or invalid data
- Final effect depends on
  - Victim cell →
    - Mostly C-elements
  - The 4-phase protocol step affected
- To understand → deeper look into C-element behavior



# SEE in C-elements

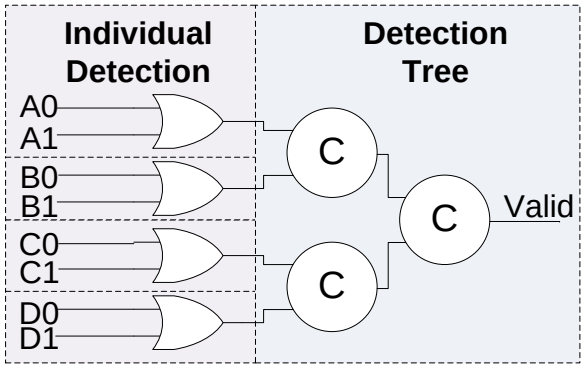


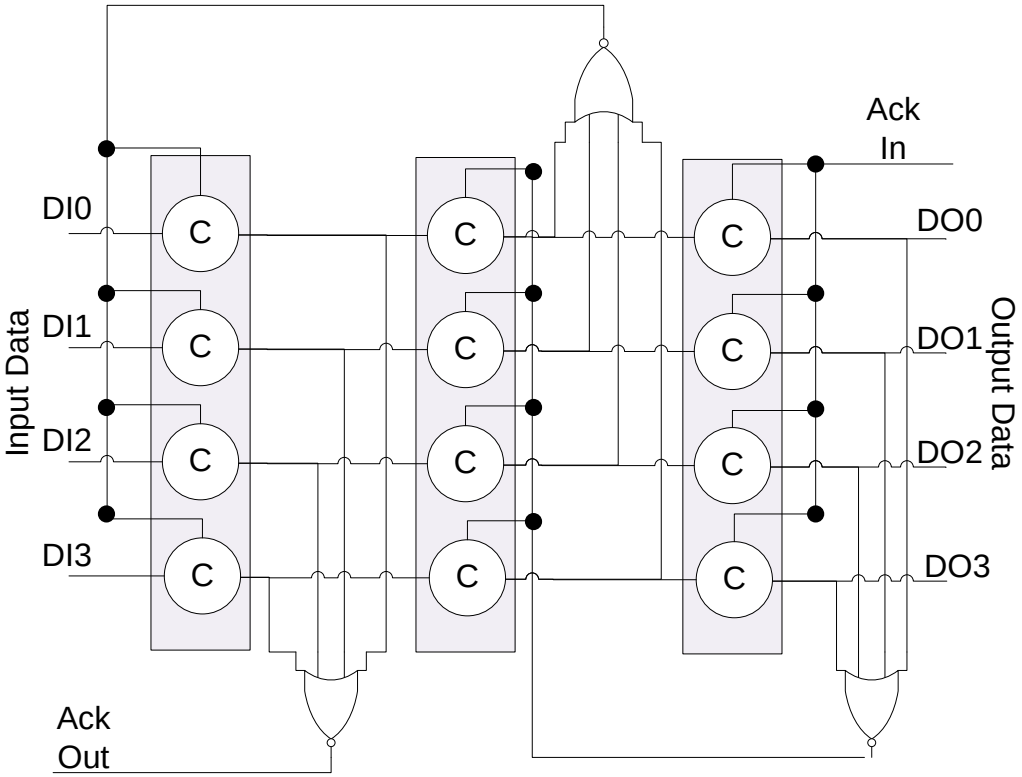

| | Charge to cause SEE (normalized to state 111) | | | | | |
|---|---|---|---|---|---|---|
| States → | 000 | 010 | 011 | 100 | 101 | 111 |
| Charge → | 0.720 | 0.088 | 0.120 | 0.097 | 0.100 | 1.000 |

- C-element driving a capacitance of 8.1fF
- Single Event Transients
  - States 000 and 111 are driving nodes
- Single Event Upsets
  - Floating Nodes (the rest) – much less charge required

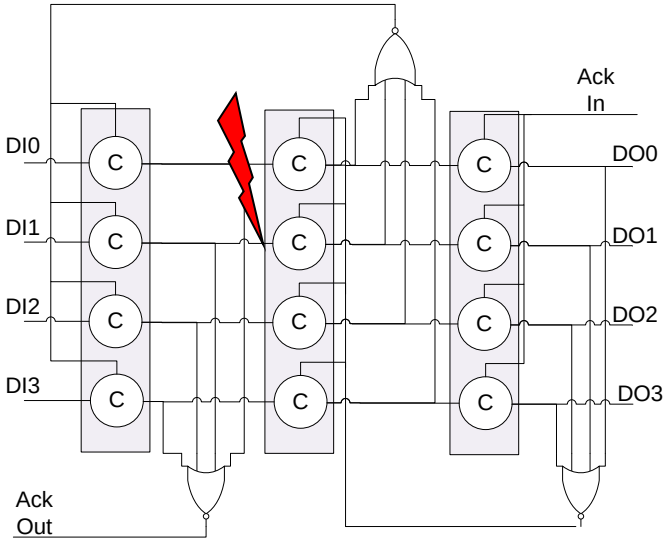
# SEE in QDI Pipelines

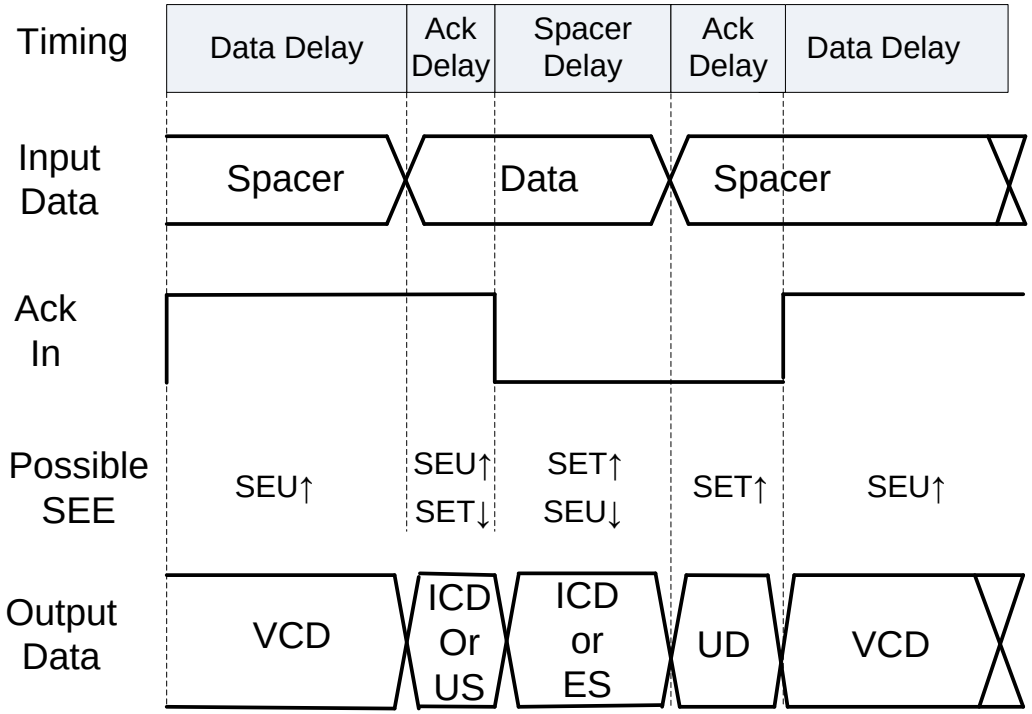



- Detection based on C-element trees are almost immune to soft errors
  - The last C-element in the tree is dangerous
- Protocol SEE and timing analysis consider
  - data link errors only



# 1-of-n Pipeline



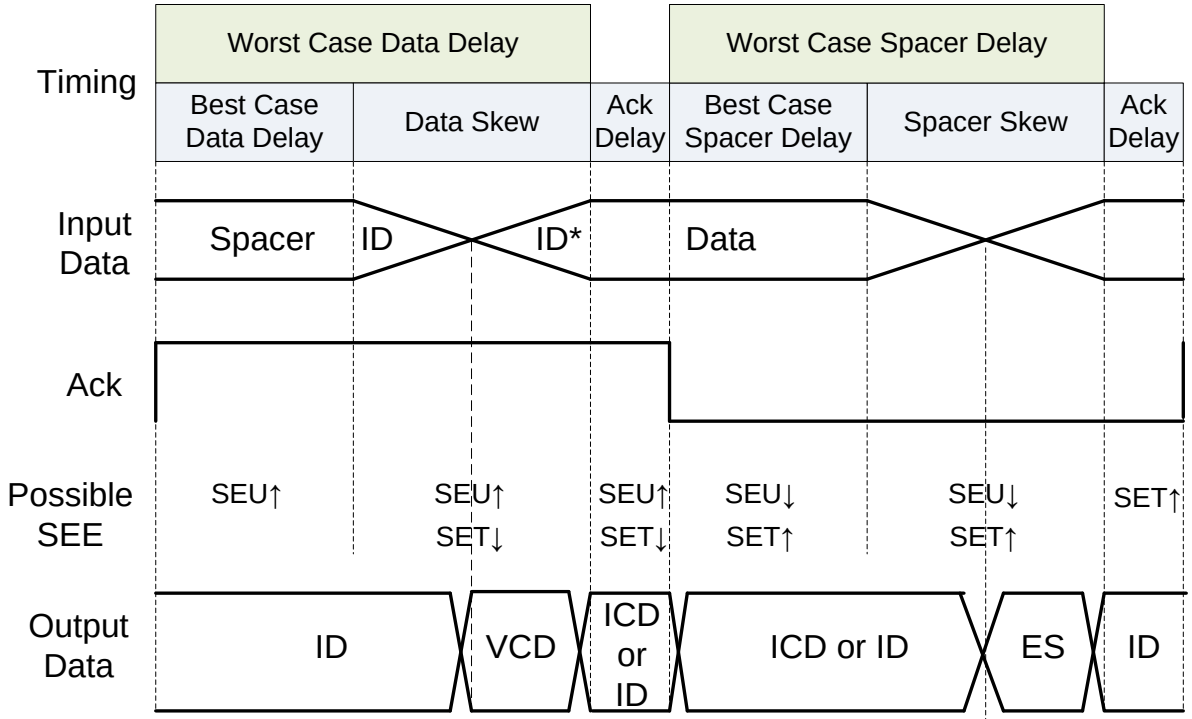




VCD = Valid Corrupted Data
ICD = Invalid Corrupted Data
US = Unexpected Spacer
ES = Early Spacer
UD = Unexpected Data

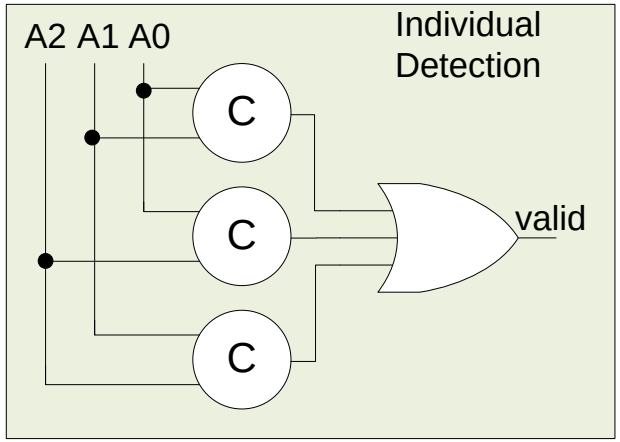
Data link always in an excited state

- 1-bit distance between data and spacer

# m-of-n QDI Pipeline (m>1)




ID = Invalid Data
VCD = Valid Corrupted Data
ICD = Invalid Corrupted Data
ES = Early Spacer




- Encoding has SEE filtering properties
- Detection is more complex → 2-of-3 example besides
- Higher code density (for 1<m<(n-1))

# SEE QDI Timing Analysis

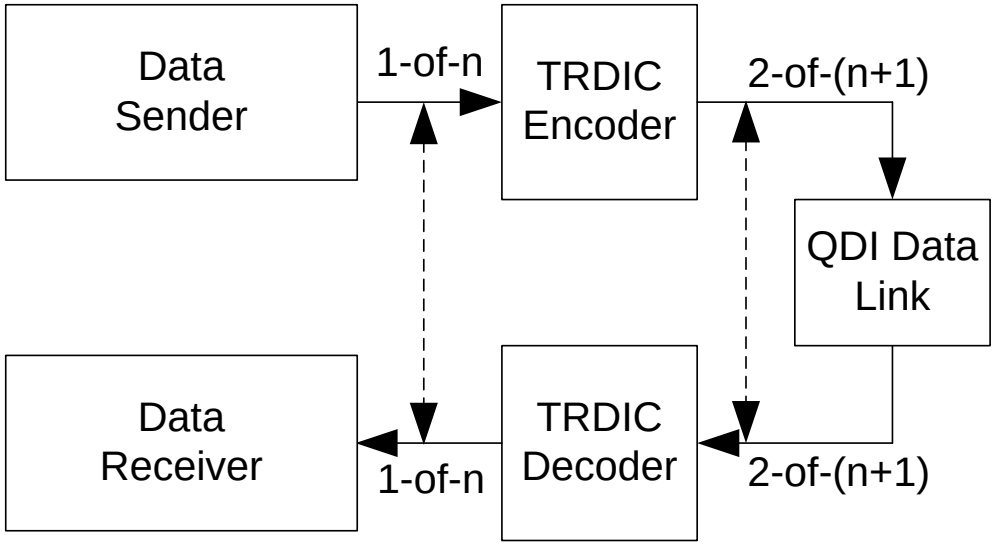
- Effect depends on the window where SEE happens
- Adding timing constraints may eliminate possibility of Valid Corrupted Data (VCD), verifiable by STA
- Stall probability depends on sender-receiver performance relationship

# Outline

- Related Work
- SEE in QDI Pipelines Analysis
- TRDIC Proposal
- SEE Validation - Flow and Environment
- Results
- Conclusions and Ongoing Work

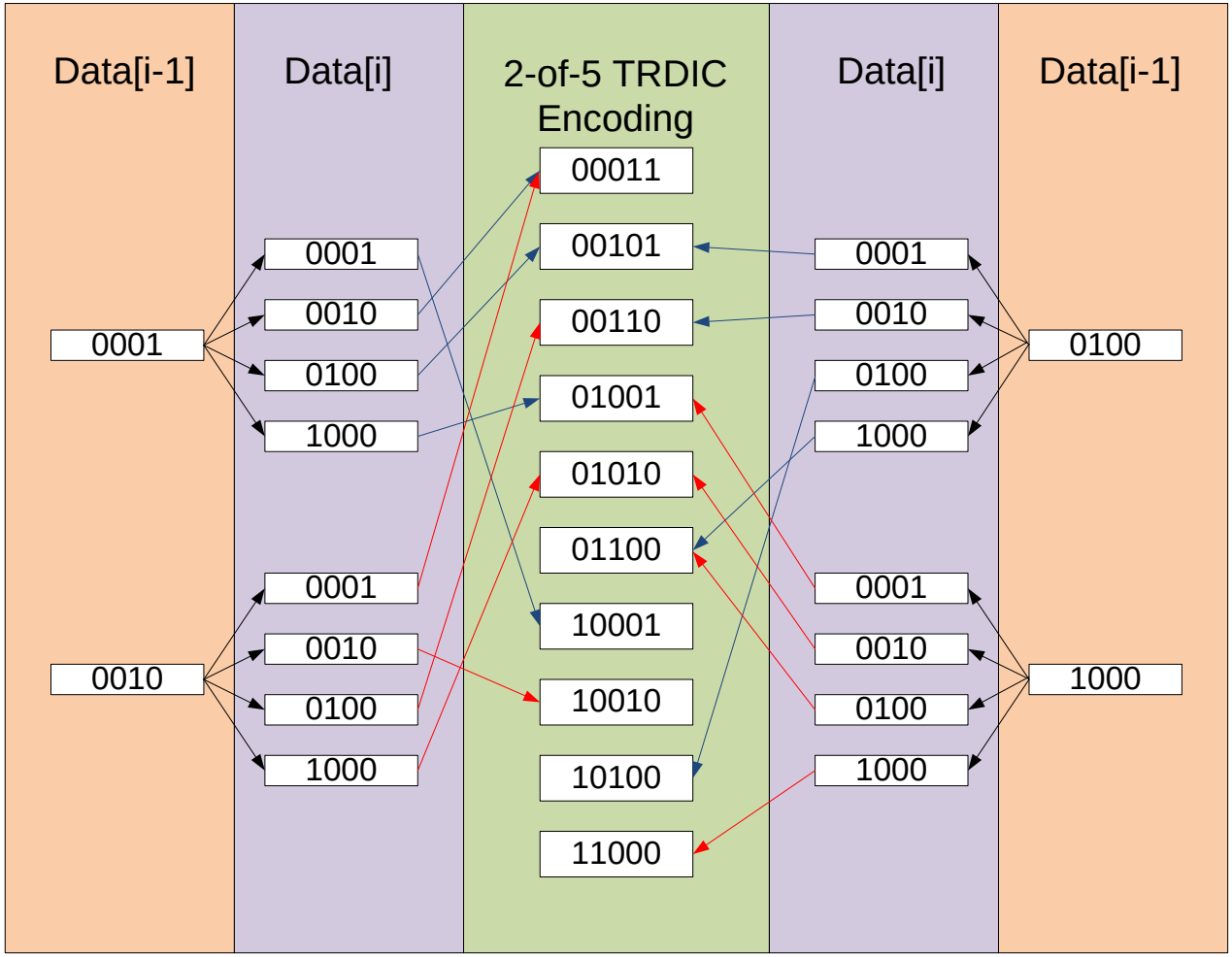
# TRDIC: Temporal Redundancy in DI Codes

- Principle
  - Convert*1-of-n* code into *2-of-(n+1) code*
    - A more robust code
  - Encode current data with previous data
    - It is **as if** we sent every datum twice
  - Double check & correction at the receiver side



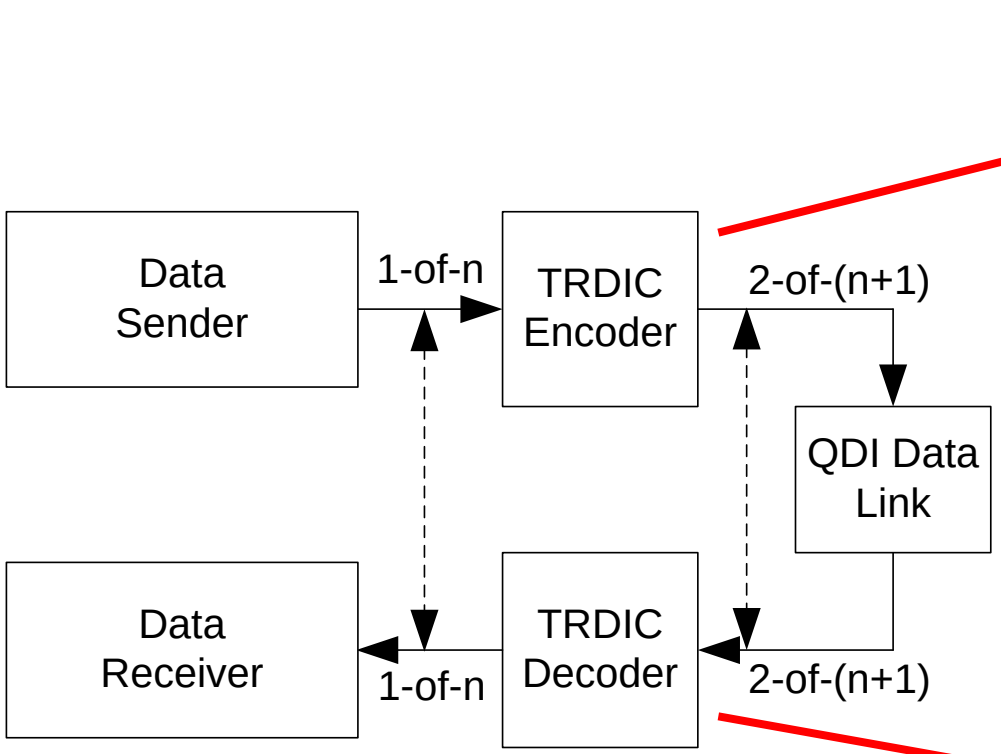

- Advantages
  - Increase SEE robustness by adding redundancy
  - Preserve performance by keeping token throughput
  - Good for intrachip communication architectures

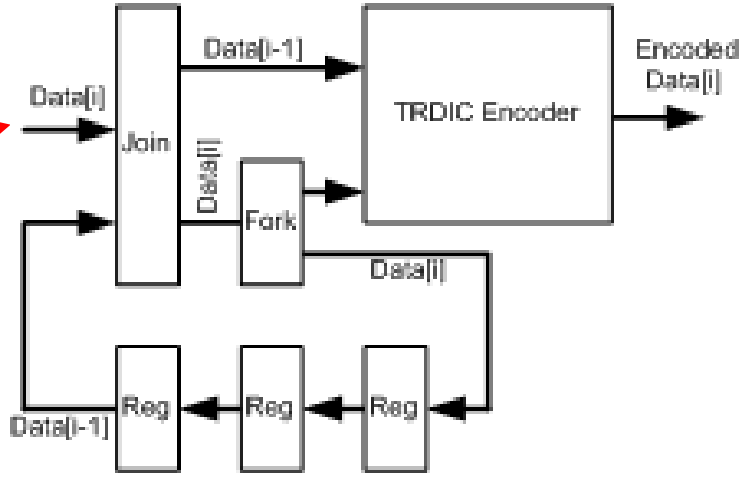
# TRDIC: Encoding Method

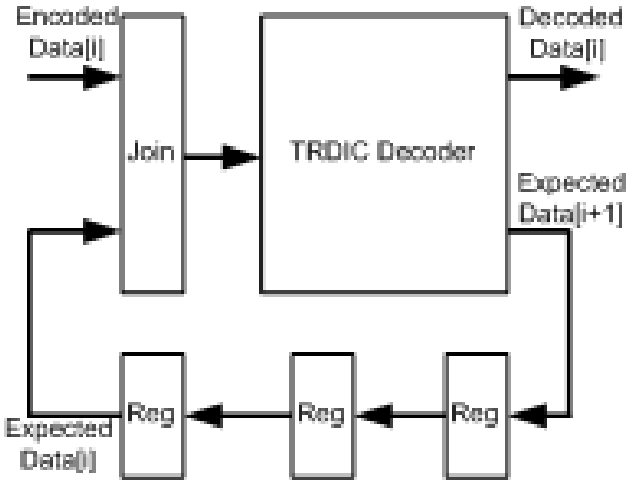



- Conversion done simply by ORing of consecutive codewords
- MSB of TRDIC indicates if consecutive codewords are equal (1) or not (0)

# TRDIC Converters



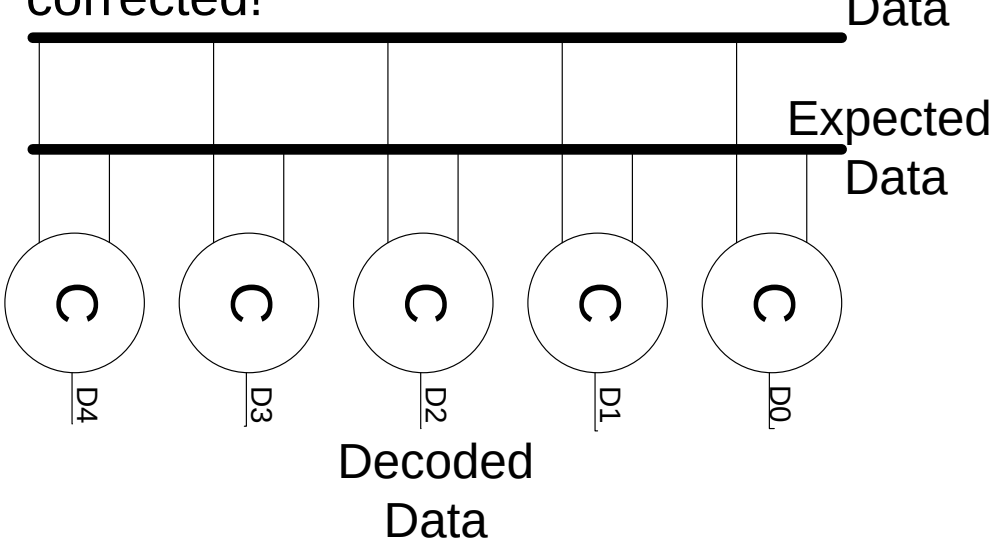



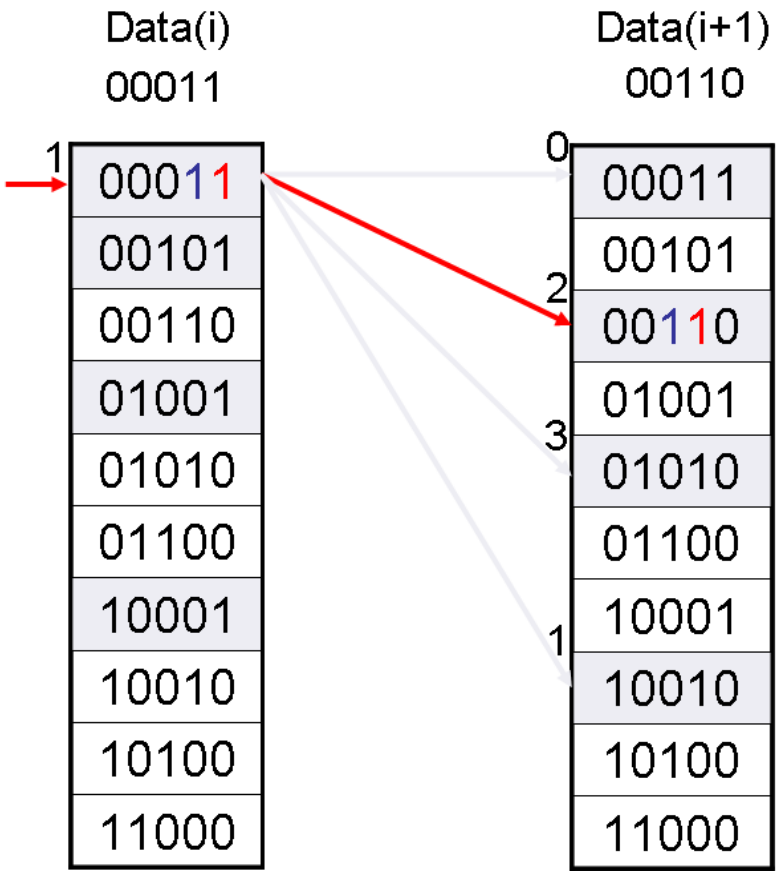



# TRDIC Double-Check Decoder

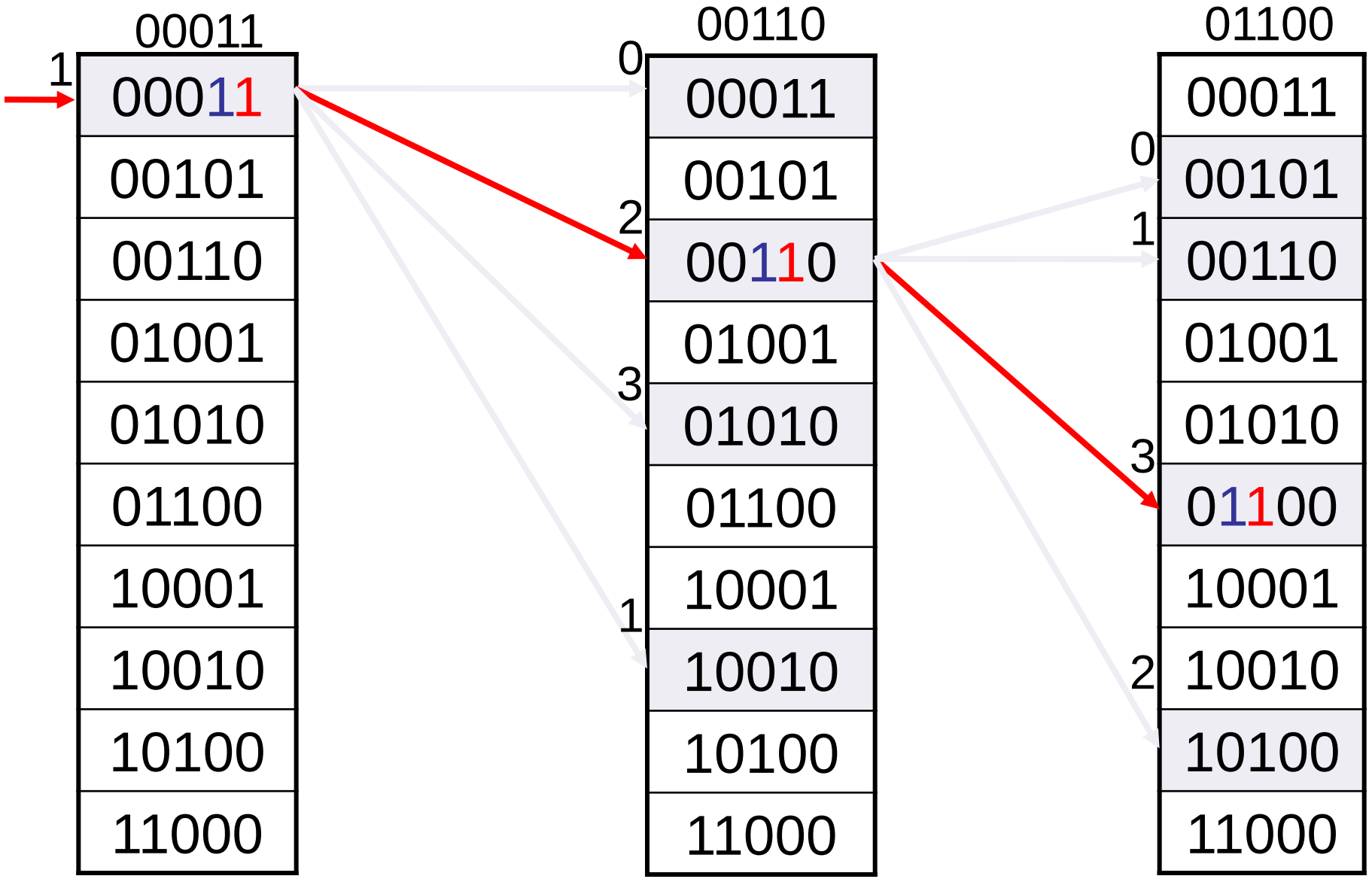
- Can solve just Invalid Corrupted Data (ICD) errors (2-stage trellis)
  - More common situation in 2-of-n codes
- A more complex trellis-based decoder increases error detection and correction capabilities

- Assume 0001 followed by 0010
- Encoder outputs 10001 (assumed) and next 00011
- Decoder obtains 0001. Next data must contain 00010. If not, error detected or corrected!



# TRDIC 3-stage Trellis Decoding

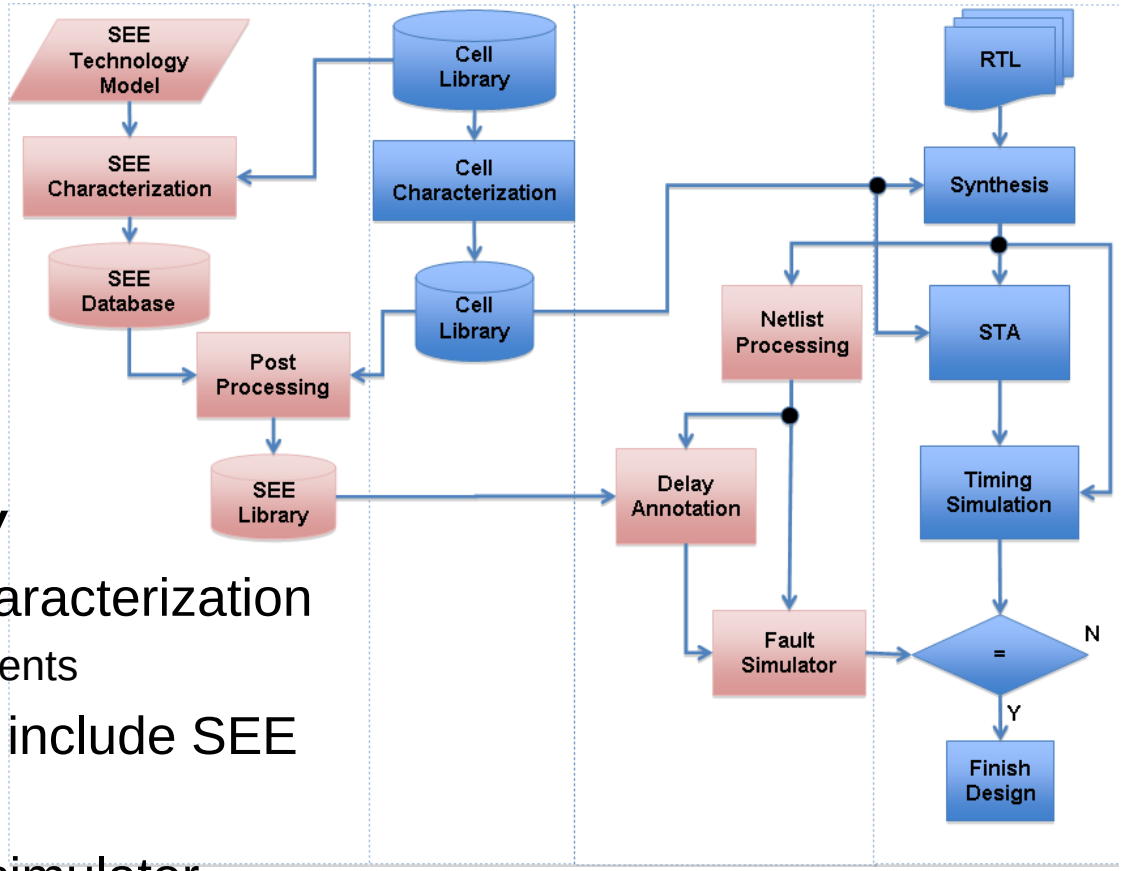


# Outline

- Related Work
- SEE in QDI Pipelines Analysis
- TRDIC Proposal
- SEE Validation - Flow and Environment
- Results
- Conclusions and Ongoing Work

# SEE Validation Flow

[Pontes, Vivet , Calazans, DATE'12]




***An accurate SEE digital flow***

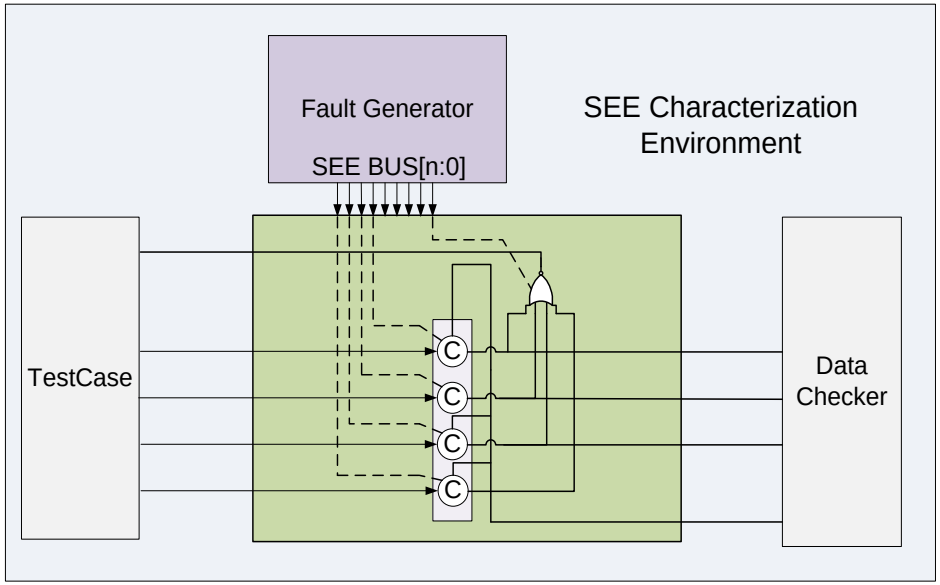
- Based on SEE Std. Cell characterization
  - For all cells, including C-elements
- Pipeline Timing Annotation include SEE glitches & delays
- Pipeline Attack using fault simulator
- Using std-tools & formats (Verilog netlist, SDF back-annotation, liberty .lib format)

# SEE Validation Environment

- Design Flow
  - Implementation using pseudo-synchronous technique (dummy rst clk)

[Thonnart, Beigné, Vivet ASYNC'12]

  - SEE Characterization & Simulation Environment
  - Attack on pipeline components



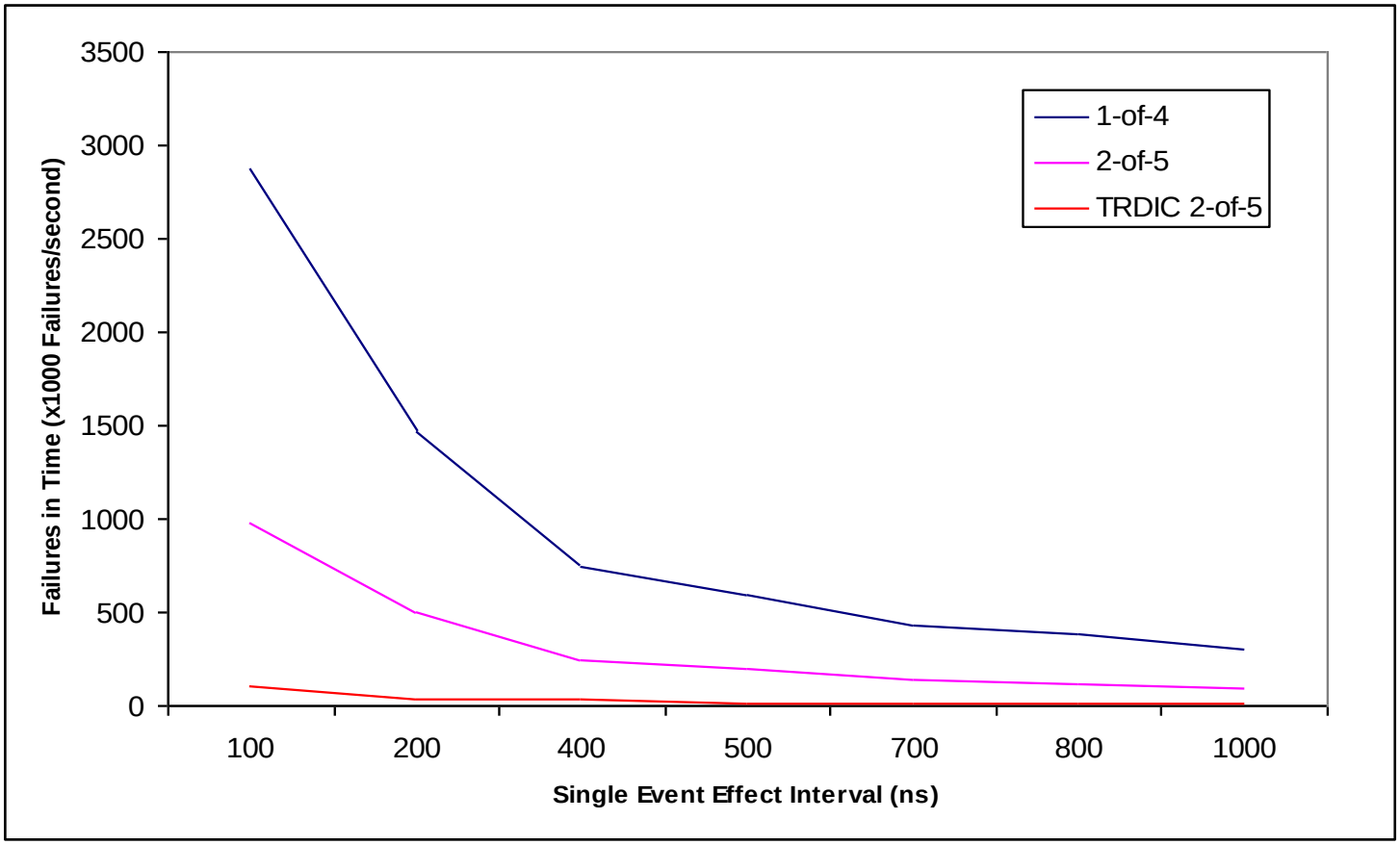

- Study of various QDI pipelines
  - 1-of-4
  - 2-of-5
  - 2-of-5 TRDIC (without encoder/decoder)
- Technology
  - STMicroelectronics, LP CMOS, 32nm

# Outline

- Related Work
- SEE in QDI Pipelines Analysis
- TRDIC Proposal
- SEE Validation - Flow and Environment
- Results
- Conclusions and Ongoing Work

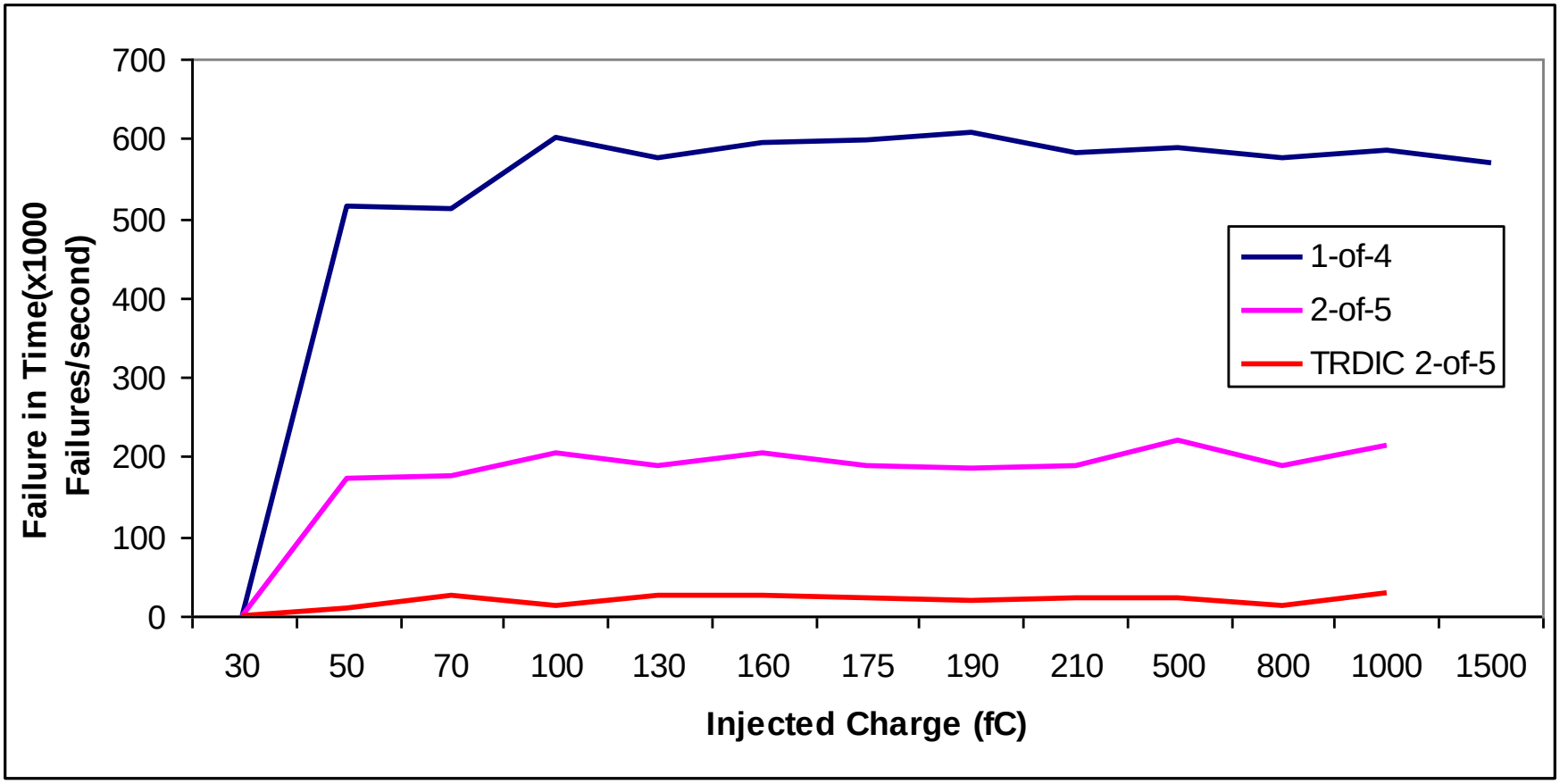
# SEE Fault Simulation Results (1/2)

- Failure x SEE Injection Rate
  - SEE Injection Charge = 175fC



# SEE Fault Simulation Results (2/2)

- Failure x SEE Injection Charge
  - SEE Rate = $5*10^6$ SEEs/second



# Results (16 stages, 32-bit WCHB pipeline)

## Area

| | Asynchronous Cells | Combinational Cells | Total |
|---|---|---|---|
| 1-of-4 | 1264/1919.7 | 482/1007.7 | 1746/2927.4 |
| 2-of-5 | 4080/6120.3 | 1280/2142.0 | 5226/8338.0 |

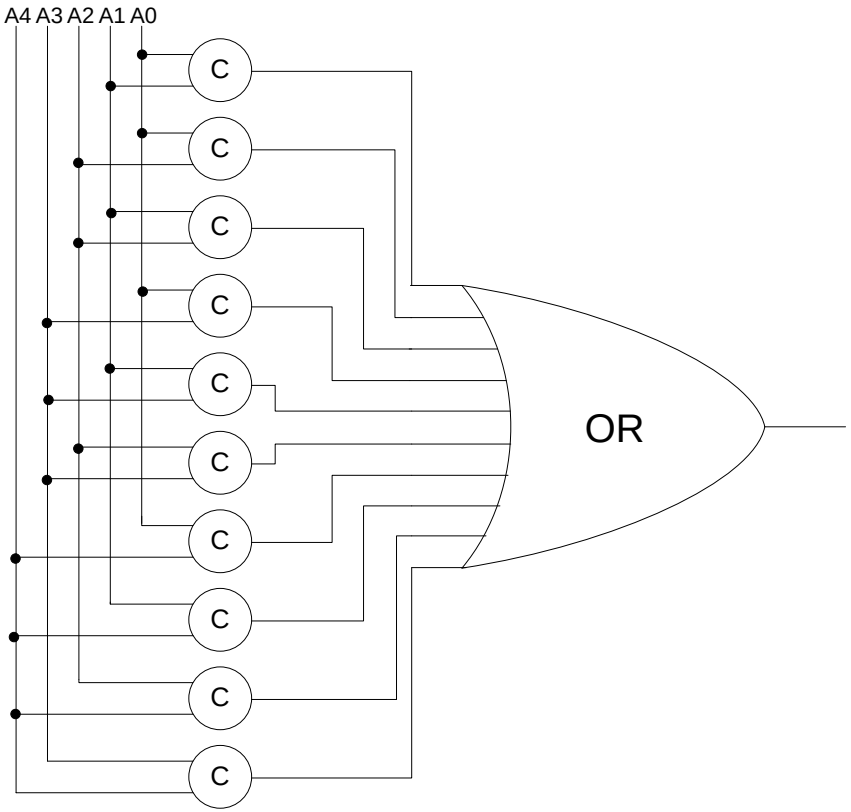
## Power

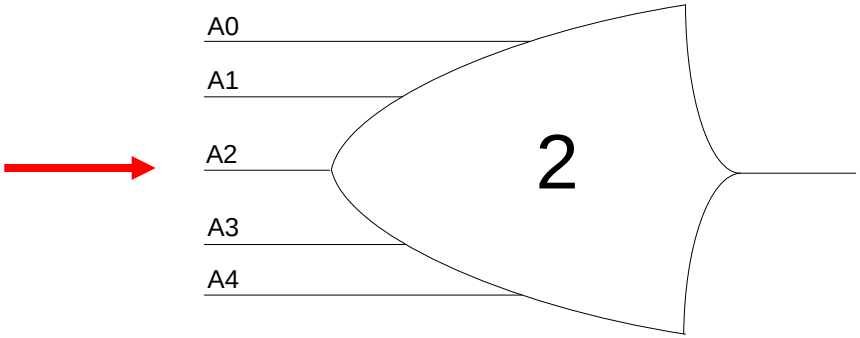
| | Leakage (µW) | Dynamic (µW) | Total (µW) |
|---|---|---|---|
| 1-of-4 | 134.7 | 2578.8 | 2713.5 |
| 2-of-5 | 317.4 | 5335.4 | 5652.8 |

## Performance

| Code | Maximum Throughput (Gbits/sec) | Latency (ns) |
|---|---|---|
| 1-of-4 | 40.80 | 1.2125 |
| 2-of-5 | 32.52 | 1.3685 |

# Completion Detection complexity




*What if we used complex gates? (like an NCL gate)*

*Completion detection for 2-of-5*

# Outline

- Related Work
- SEE in QDI Pipelines Analysis
- TRDIC Proposal
- SEE Validation - Flow and Environment
- Results
- Conclusions and Ongoing Work

# Conclusions and Ongoing Work

- TRDIC: Temporal Redundancy for DI Code
  - SEE filtering is provided by *2-of-n* encoding
  - Temporal Redundancy allows multi-bit correction
  - Well adapted for QDI pipeline & communication architecture
  - Fully evaluated with a SEE fault simulator
- Ongoing Work
  - Complete the design of TRDIC Encoder/Decoder
  - Evaluation of different TRDIC pipeline implementations
  - Integration of  TRDIC in Asynchronous Network-on-Chips
    - Hermes-A and Hermes-AA (PUCRS) [PATMOS'10], [SOCC'10]
    - ANoC (CEA-LETI) [ASYNC'05]
  - Impact in Area and Power
    - Design of specific cells for 2-of-5 completion detection

# Thanks to:

- CNPq (Brazilian Research Funding Agency)