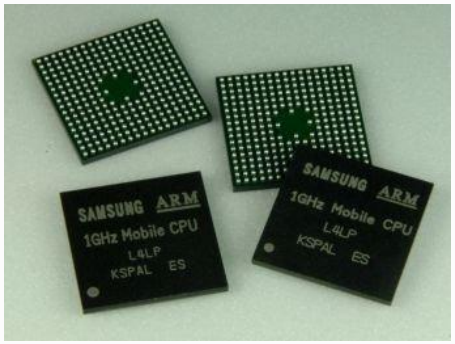
Adaptive Robustness Tuning for High Performance Domino Logic

Bharan Giridhar[1], David Fick[1],
Matthew Fojtik[1], Sudhir Satpathy[1], David Bull[2],
Dennis Sylvester[1] and David Blaauw[1]

[1]University of Michigan, USA
[2]ARM, United Kingdom

Motivation

- Targeted use of Domino logic remains the designer's choice to implement speed critical paths in power constrained designs



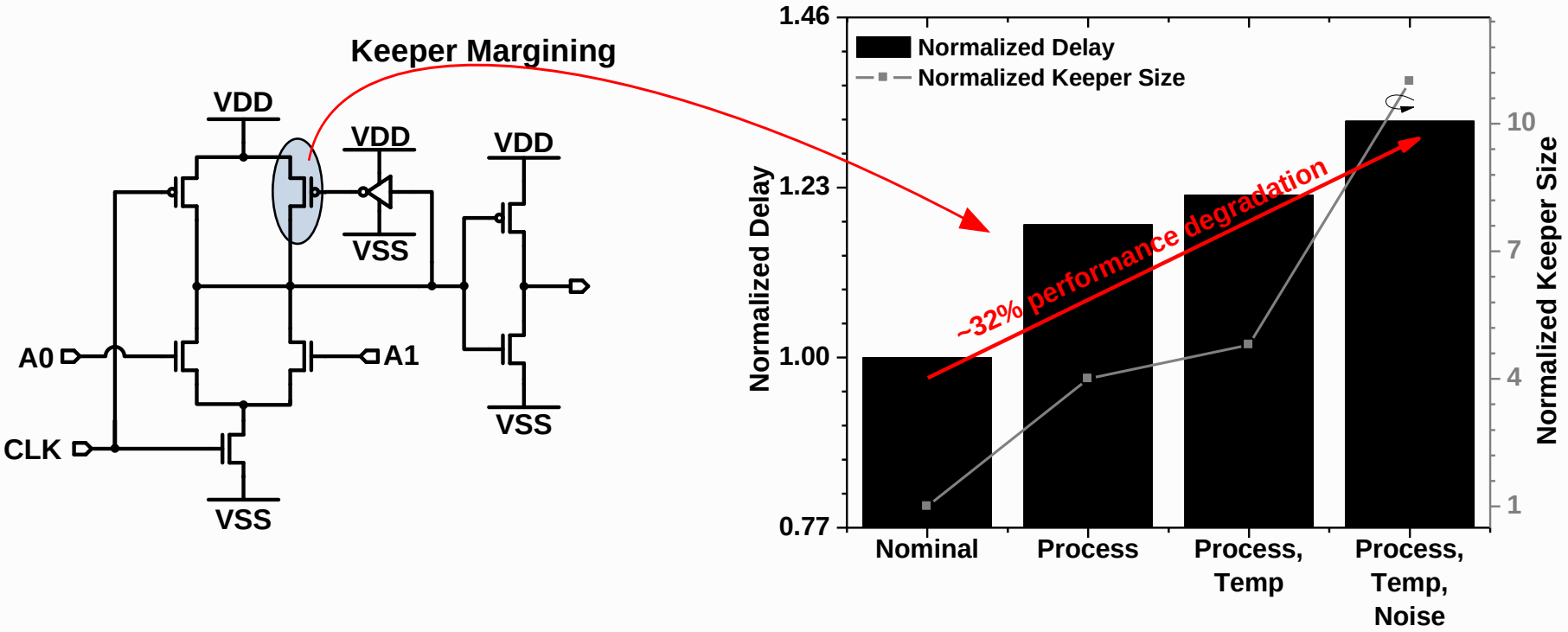
Samsung Hummingbird, 2009
http://www.samsung.com

AMD Bulldozer
Golden et al., ISSCC 2011

- With increasing process variation, domino logic is getting less beneficial
  - Increasing margins for leakage, charge sharing and noise

Motivation

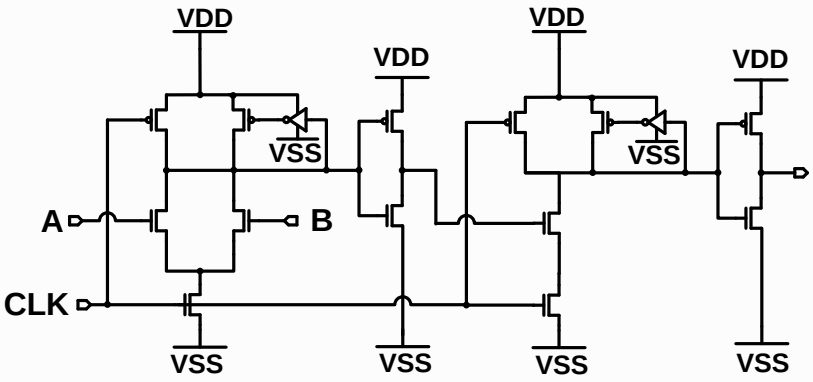
- Margining the keeper for robustness under worst-case PVT can degrade performance by ~32%



- ART Domino tracks PVT and trades extra robustness and timing margins for performance

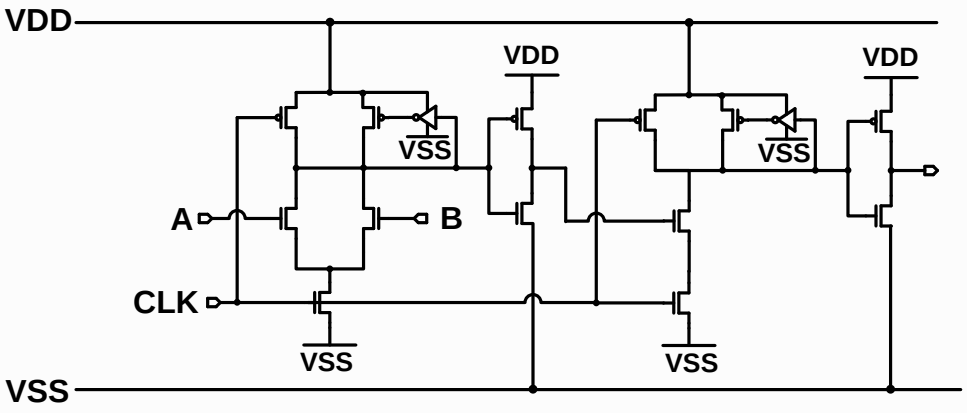
ART Domino Gate

- Fast, speculative evaluation followed by fully margined (safe) evaluation to detect errors



ART Domino Gate

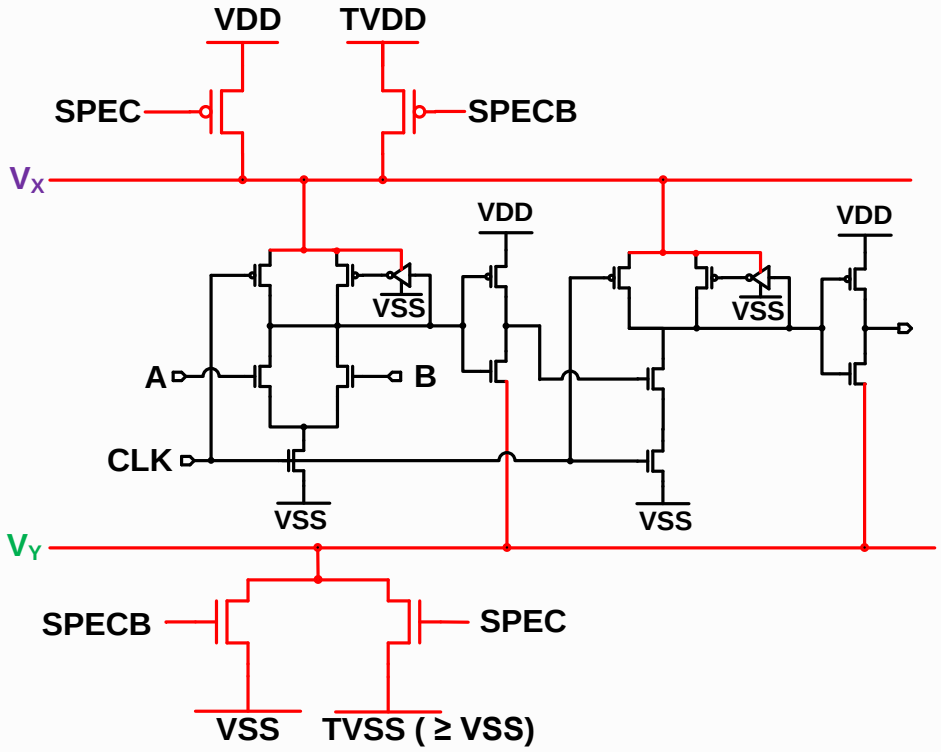
- Fast, speculative evaluation followed by fully margined (safe) evaluation to detect errors

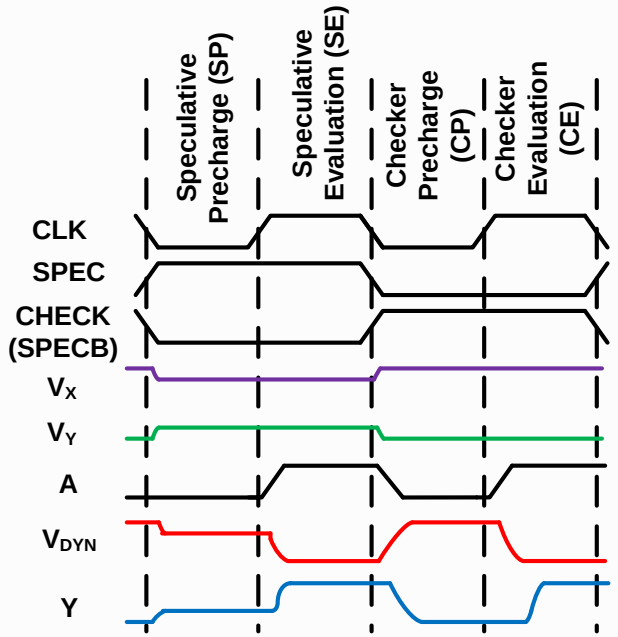


ART Domino Gate

- Fast, speculative evaluation followed by fully margined (safe) evaluation to detect errors

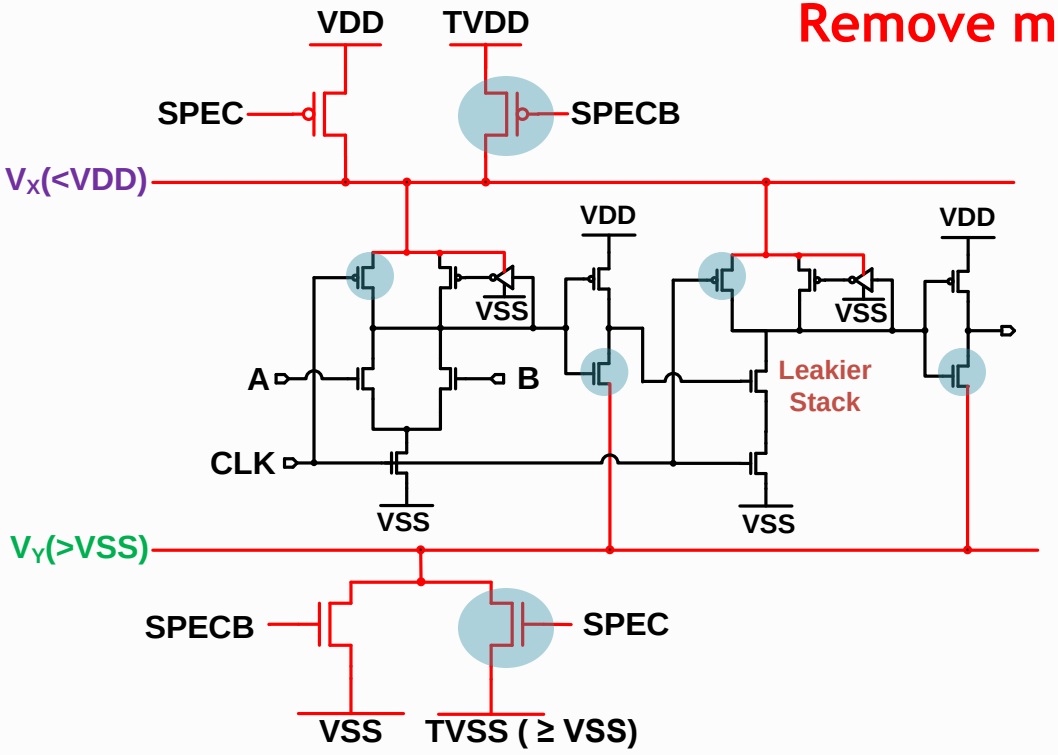
Headers/footers shared across gates to minimize overhead



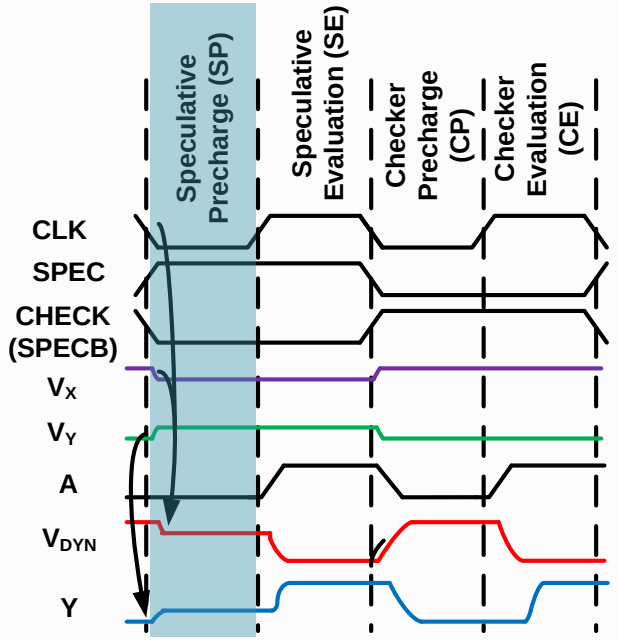


ART Domino Gate

- V_X < VDD and V_Y > VSS speed critical transitions at both nodes by reducing voltage swings
- V_Y > VSS speeds the following gate by trading its noise margin for speed

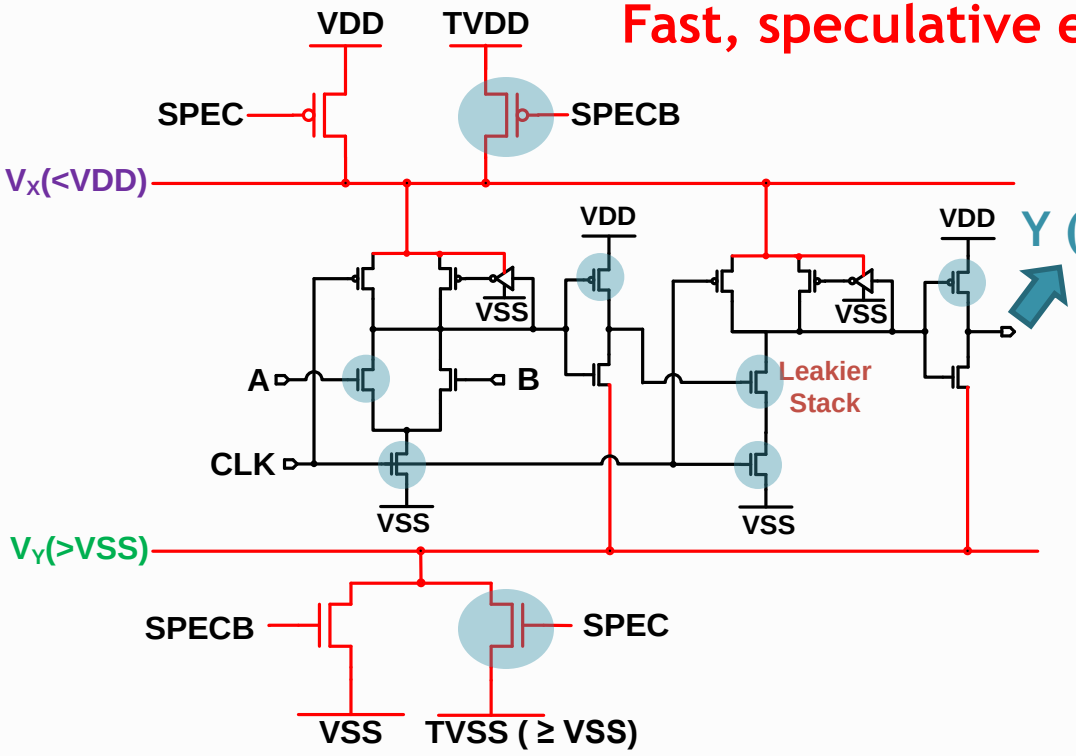


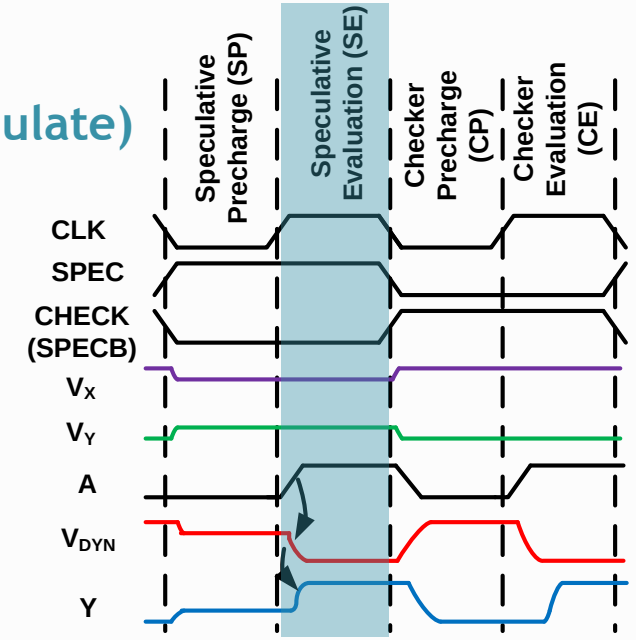
Remove margins



ART Domino Gate

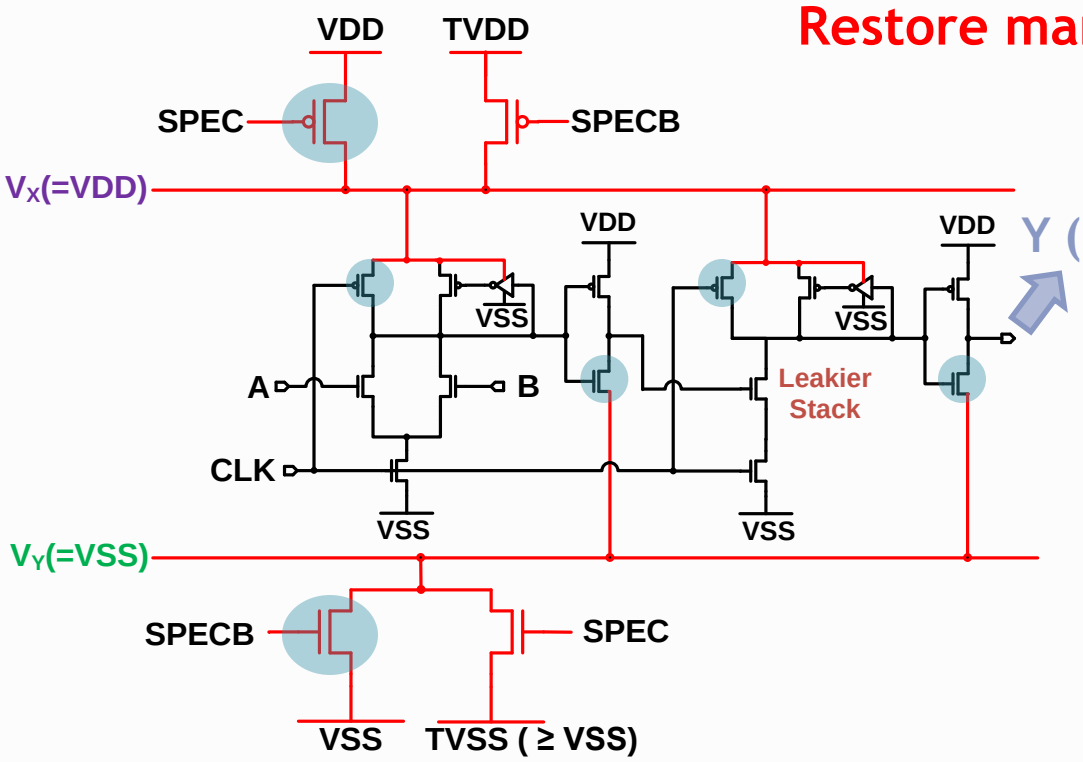
- V_X < VDD and V_Y > VSS speed critical transitions at both nodes by reducing voltage swings
- V_Y > VSS speeds the following gate by trading its noise margin for speed

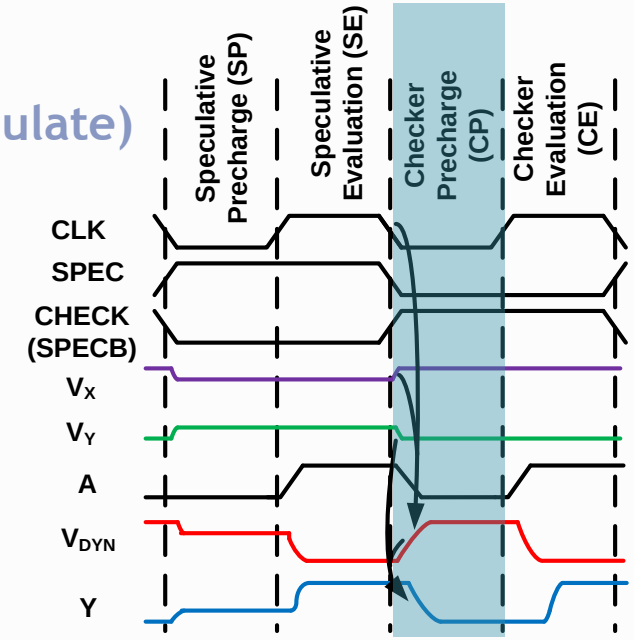




ART Domino Gate

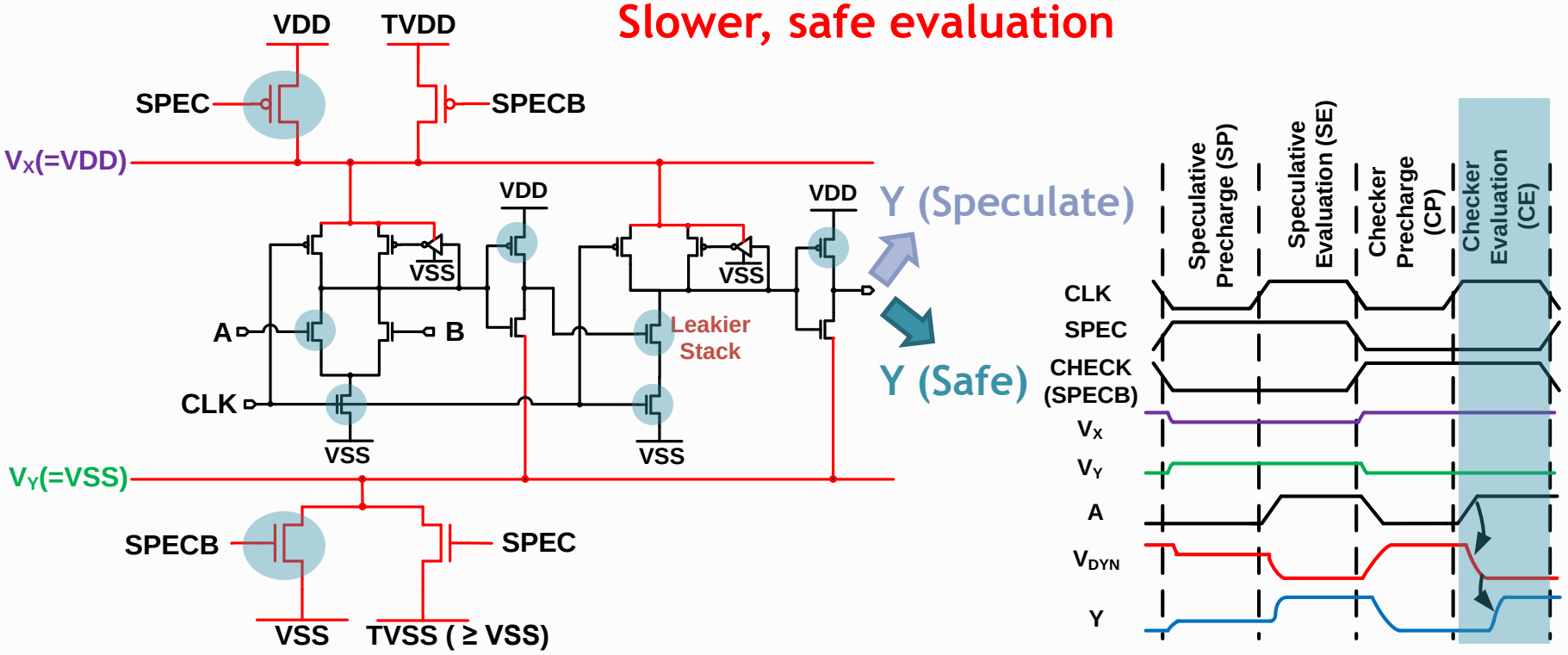
- Slower, safe evaluation performed in the background
  - No impact on computation latency





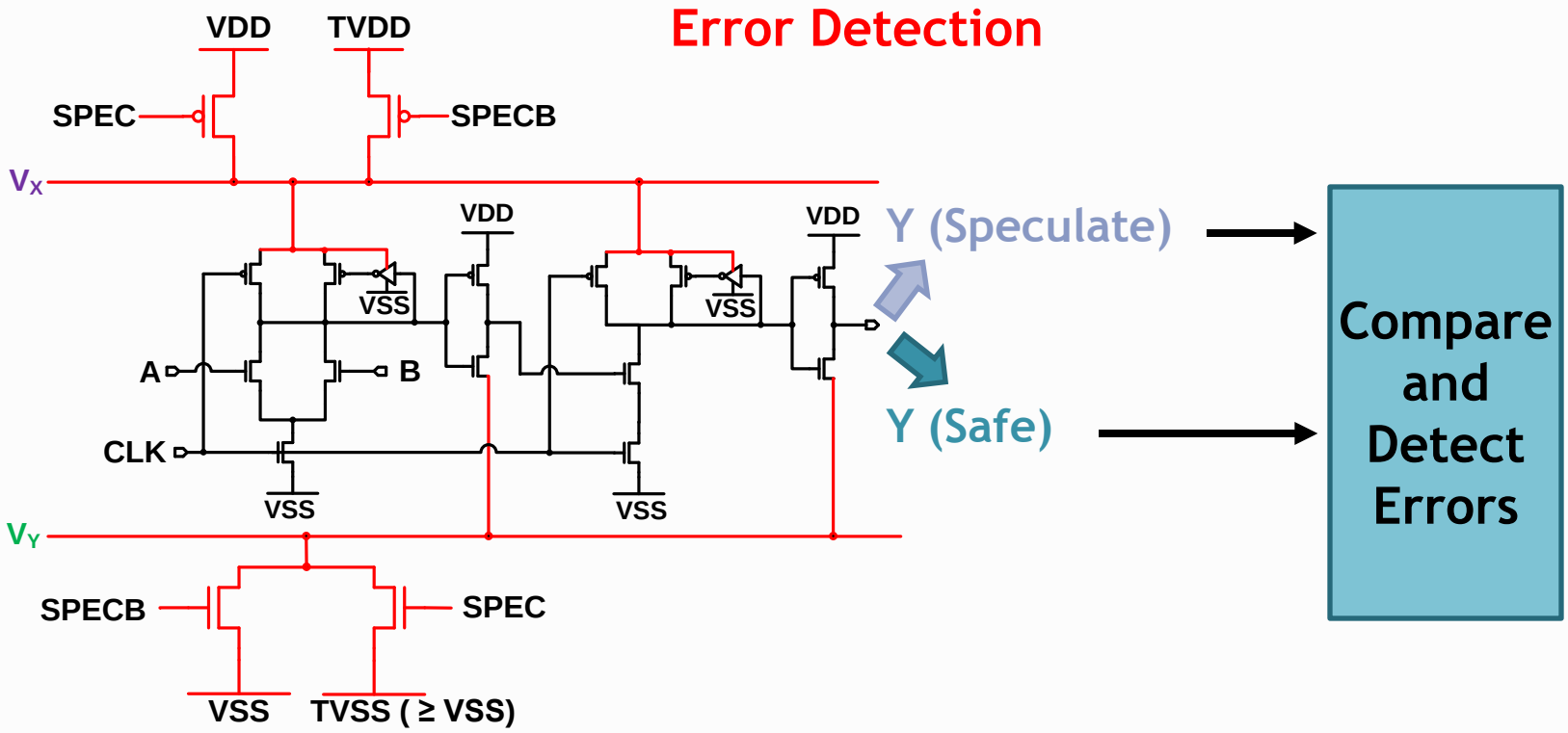
ART Domino Gate

- Slower, safe evaluation performed in the background
  - No impact on computation latency

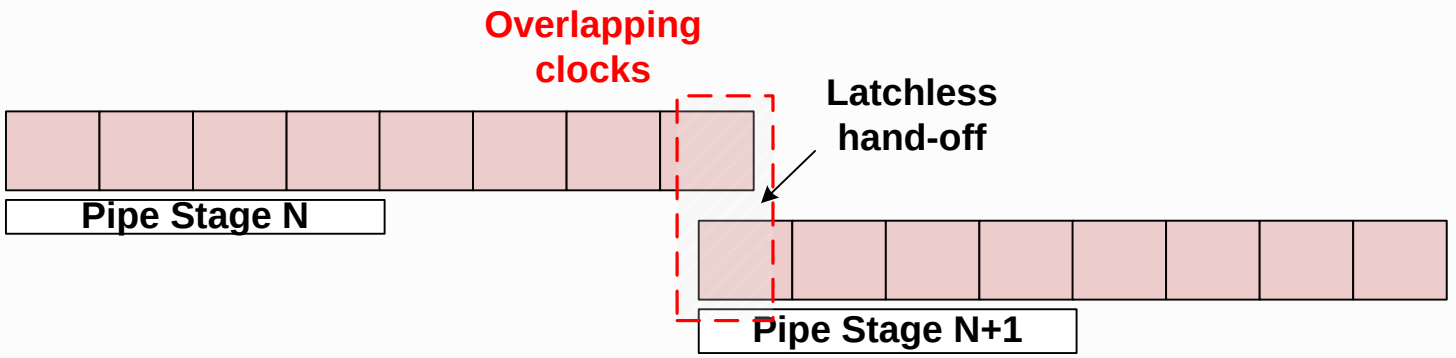


ART Domino Gate

- In case of errors, the errant computation is flushed from the pipeline
  - The result of safe evaluation is propagated, guaranteeing forward progress

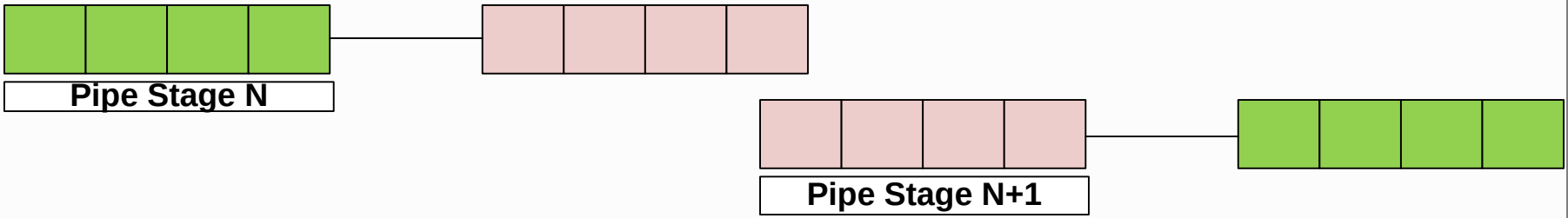


ART Pipeline

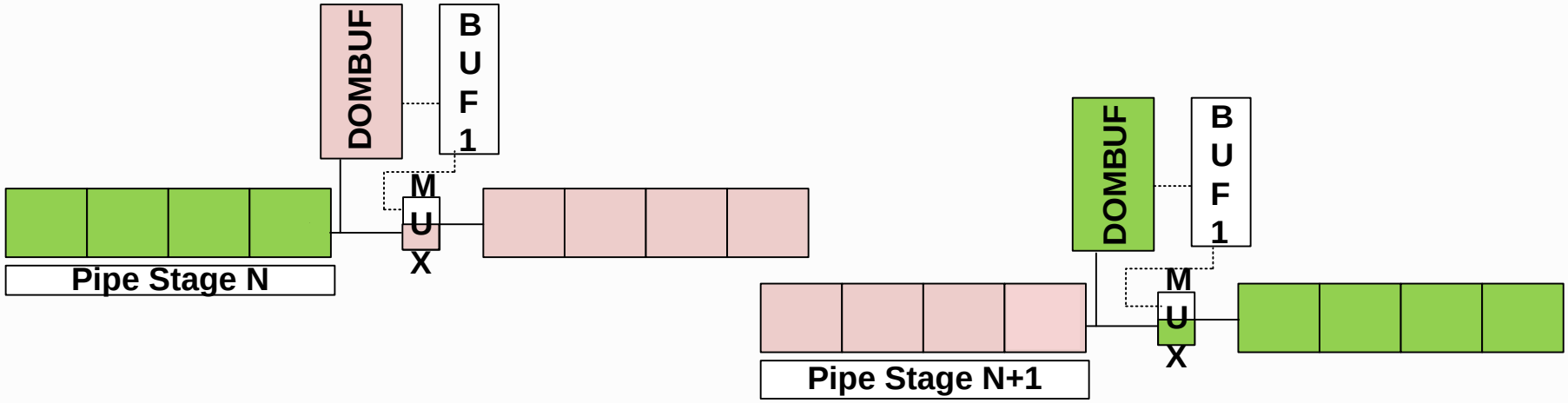


Conventional Domino Pipeline

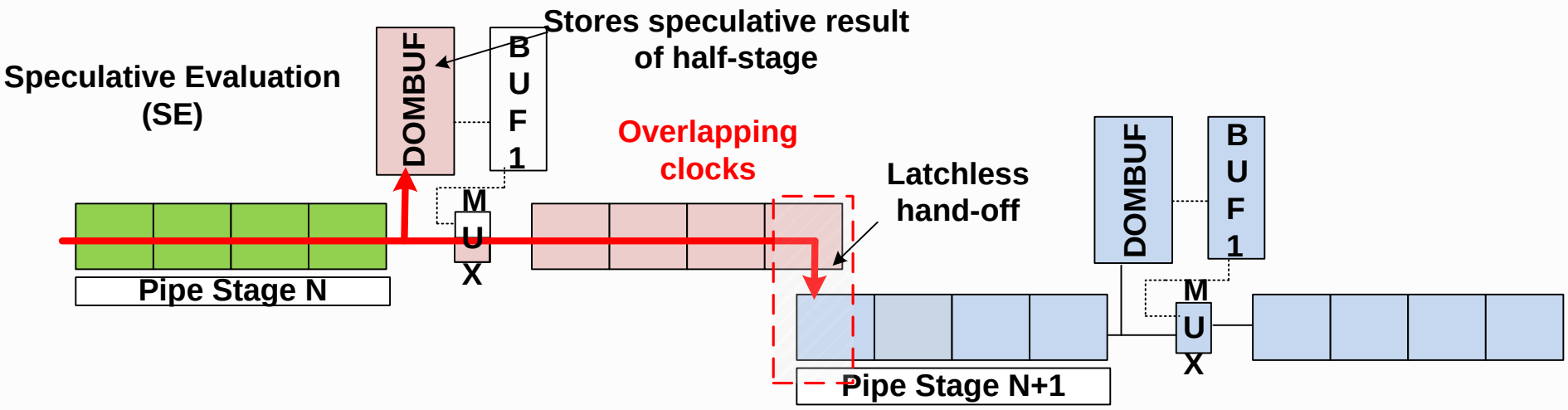
ART Pipeline



ART Pipeline

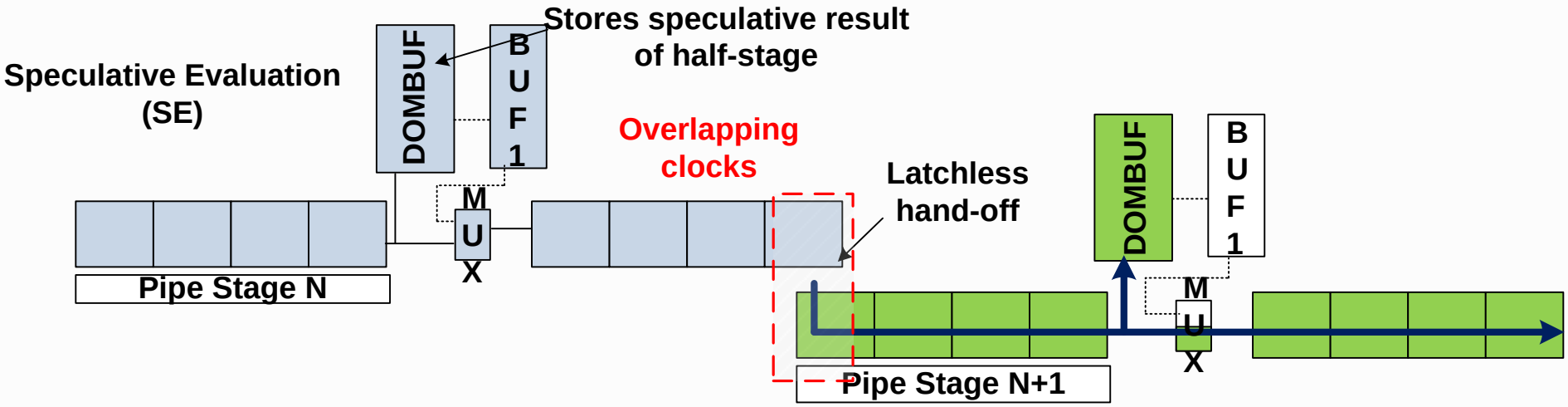


ART Pipeline



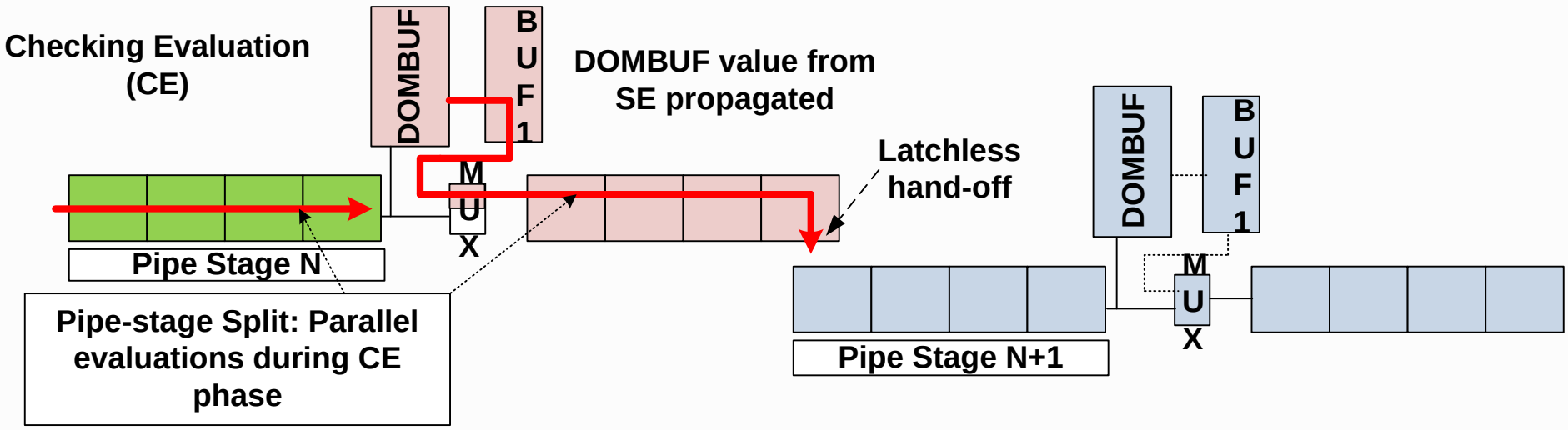
- During SE, DOMBUF snoops on the value propagated forward through the mux
- Overlapping clocks eliminate latches between pipe stages, provide skew tolerance

ART Pipeline



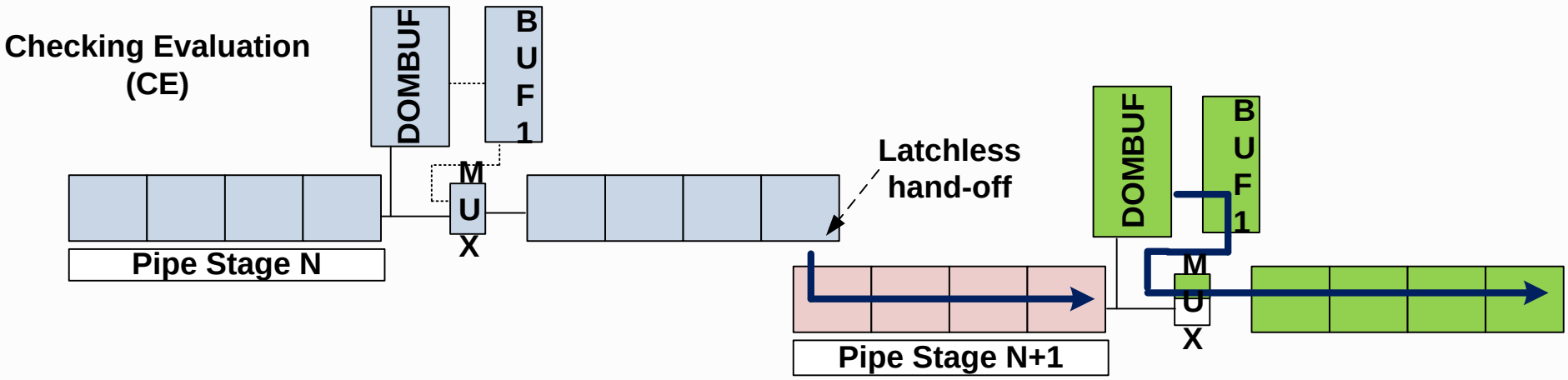
- During SE, DOMBUF snoops on the value propagated forward through the mux
- Overlapping clocks eliminate latches between pipe stages, provide skew tolerance

ART Pipeline



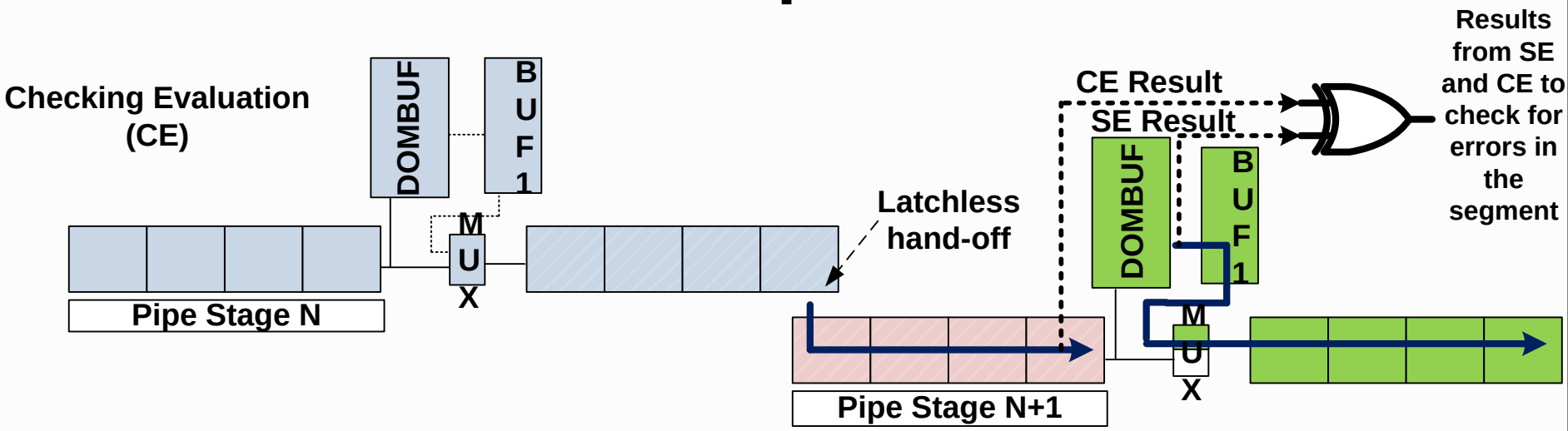
- To allow the slower safe evaluation to complete, each pipe stage is split in half during CE
- Value stored on DOMBUF propagated forward cutting the stage depth by half

ART Pipeline



- Both halves of each pipe stage perform safe evaluations simultaneously

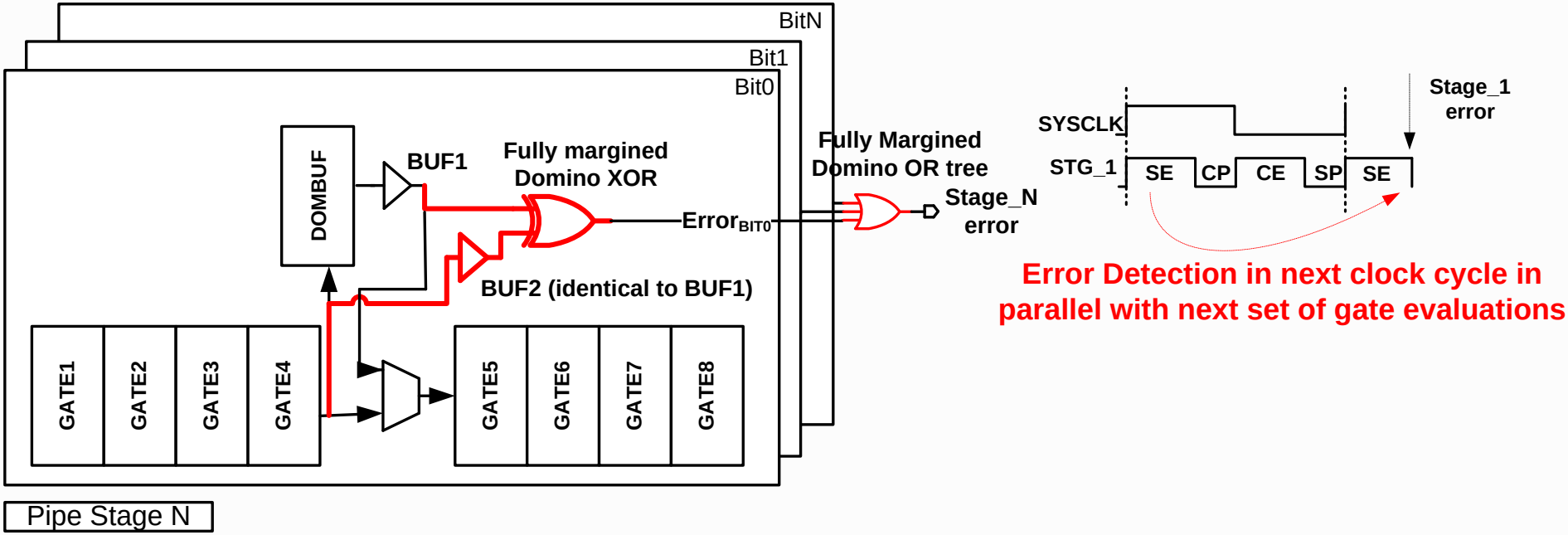
ART Pipeline



- Both halves of each pipe stage perform safe evaluations simultaneously
- Error detector at each DOMBUF checks the segment till the preceding DOMBUF for errors

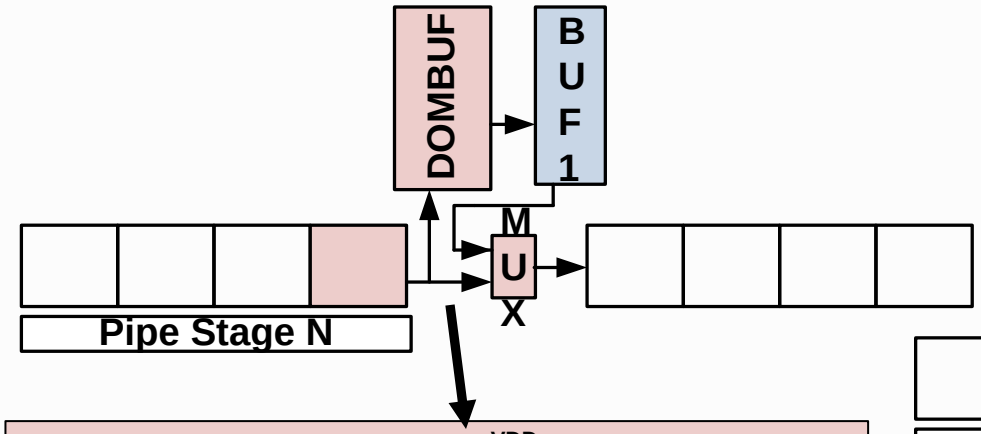
ART Pipeline

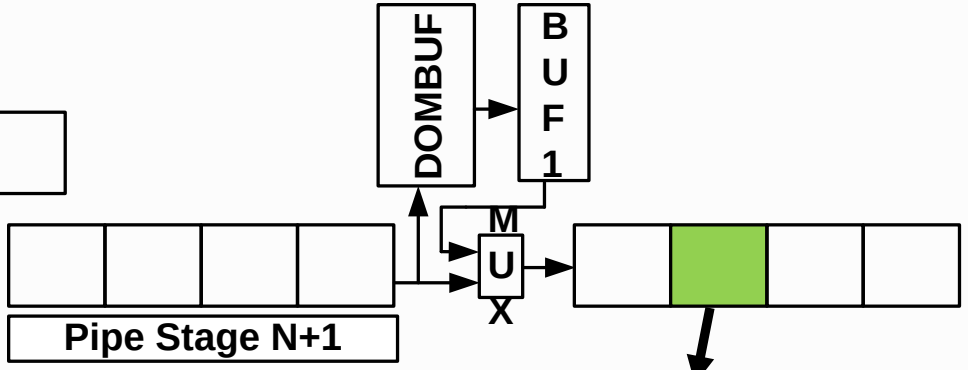
Error Detection

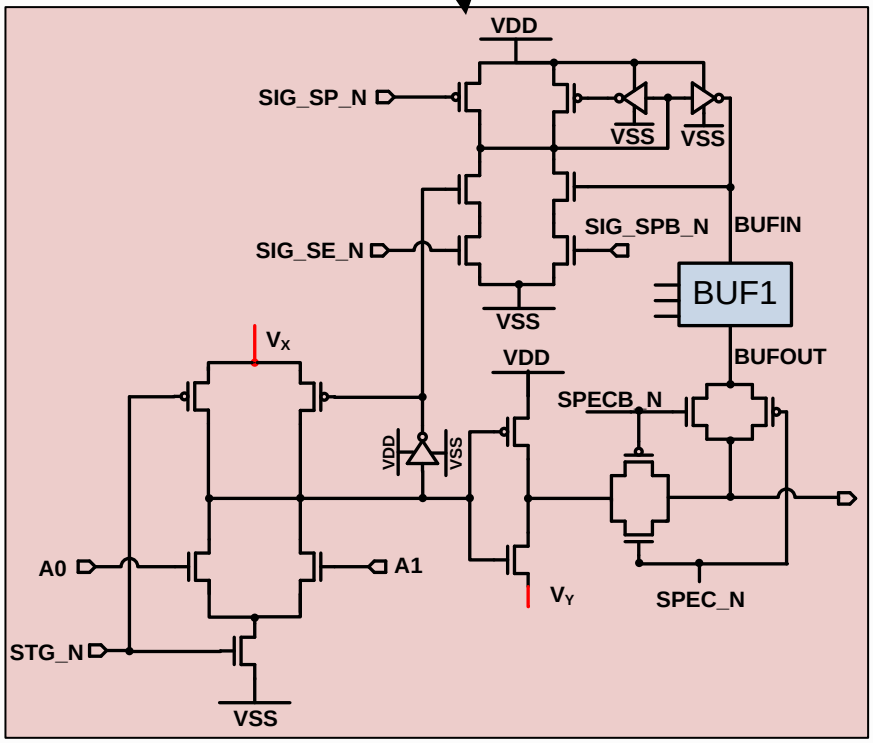


- CP: Precharge error logic (in red)
- CE: Copy Gate4 TO BUF2 and DOMBUF to BUF1
- SP+SE: Evaluate Domino XOR, Domino OR tree

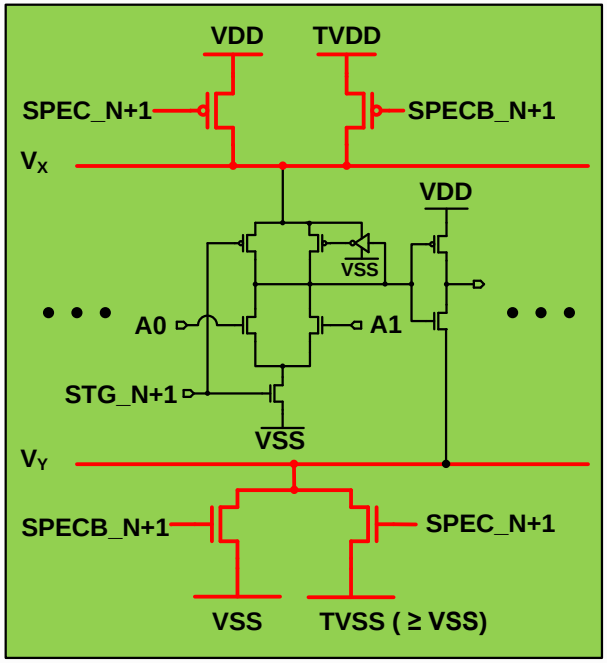
ART Pipeline



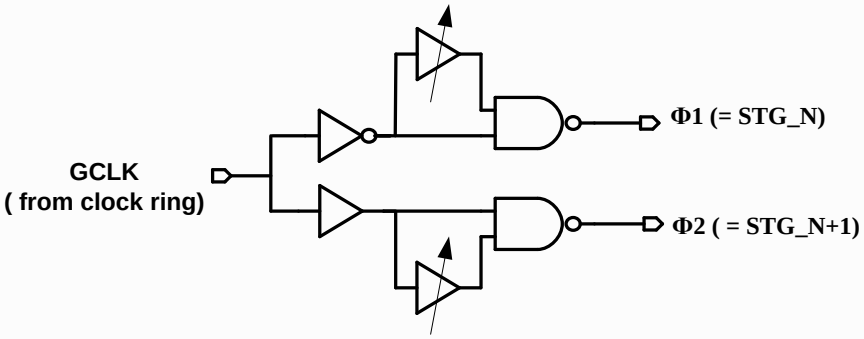


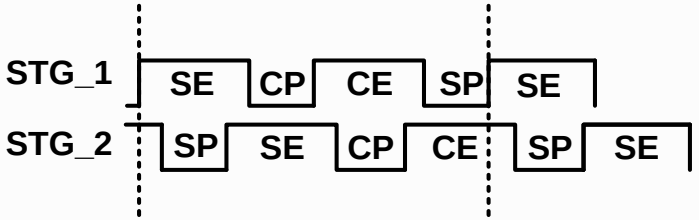


DOMBUF, BUF1(2) are fully margined

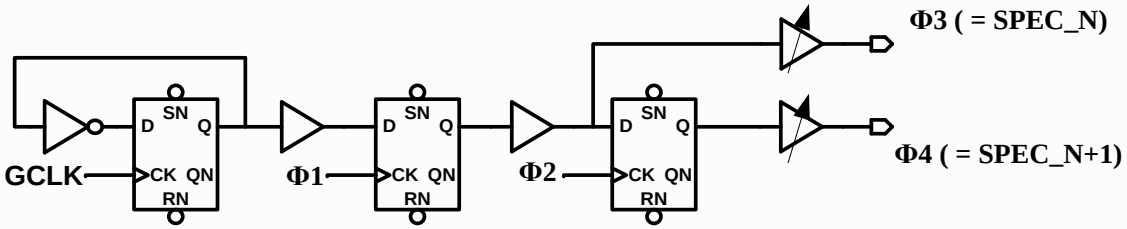


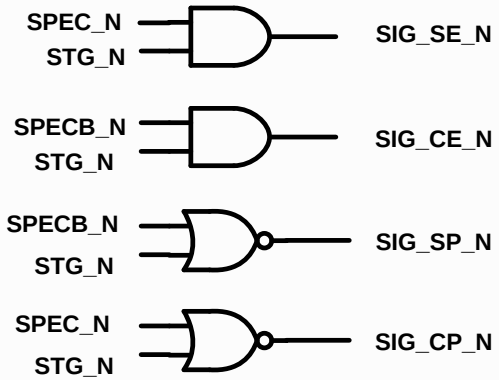
ART Clock Generation





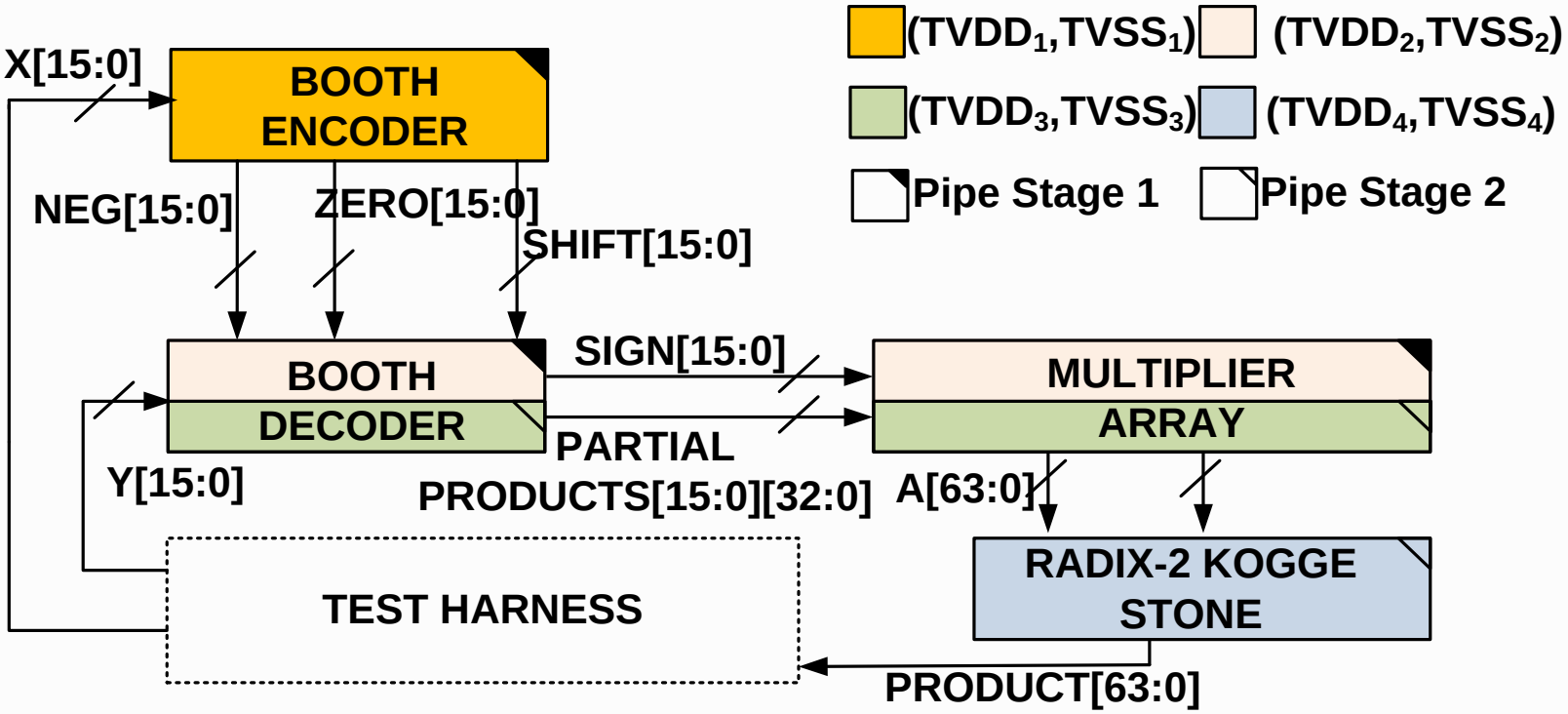
Domino gates clocked by the global clock generator





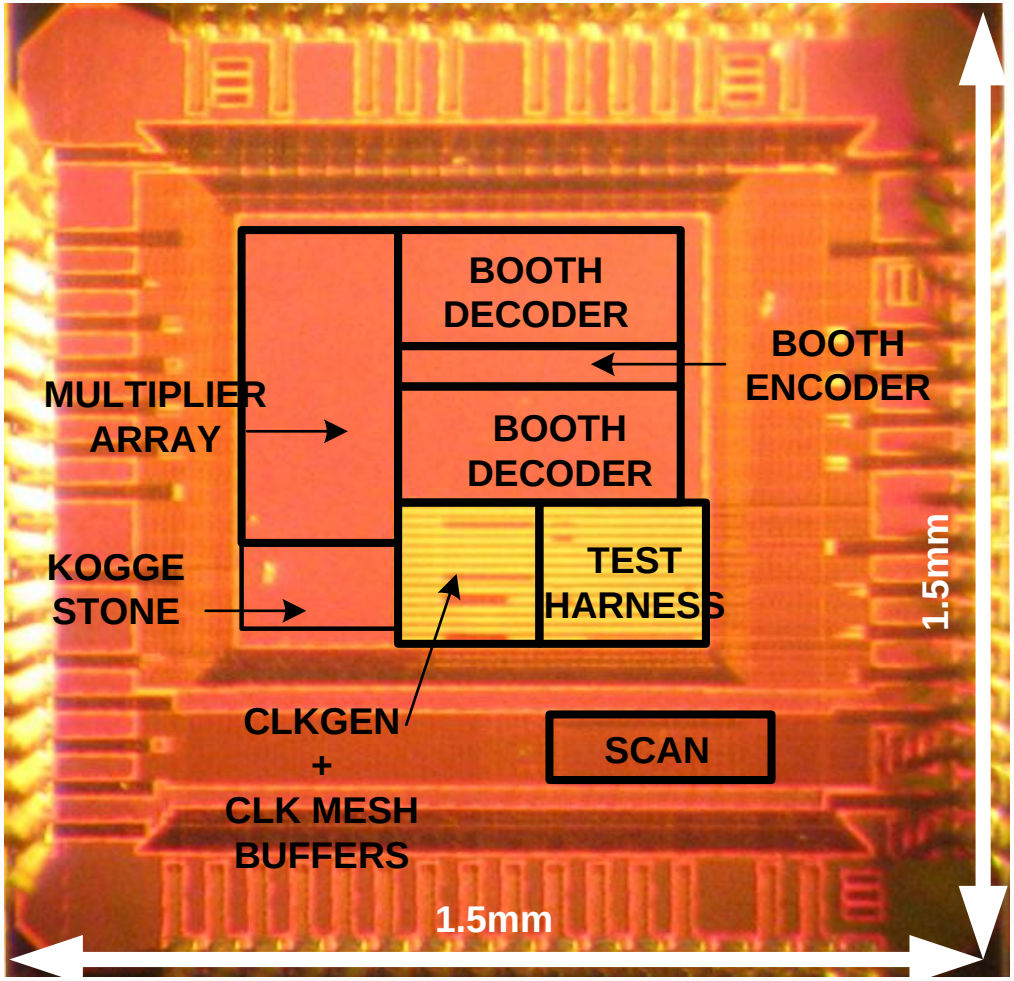
Power gates and other locally derived signals clocked using locally generated clocks

System Implementation



- 32 × 32b multiplier in 65nm CMOS
- 4 TVDD/TVSS voltage domains, 2 pipeline stages

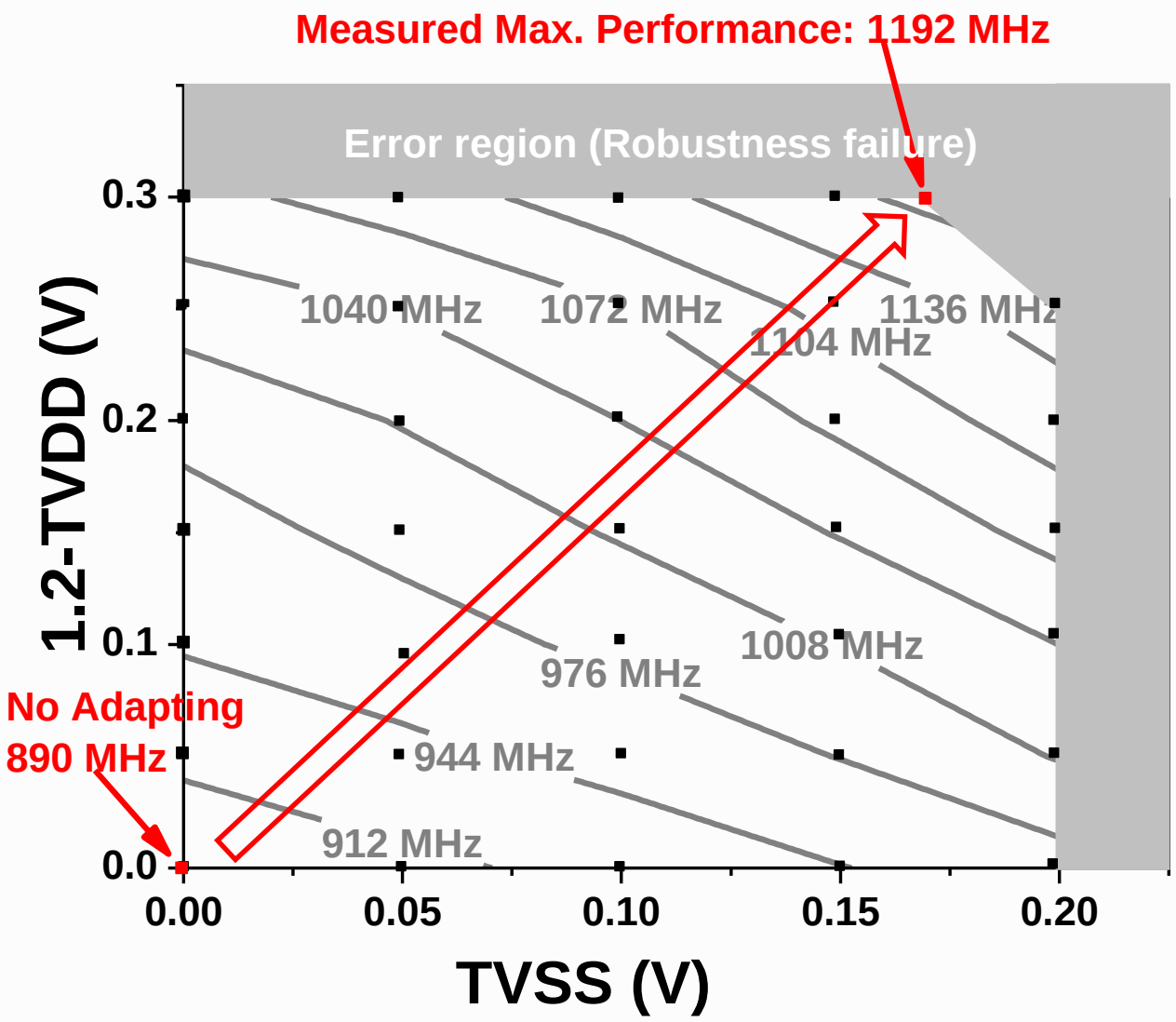
Die Micrograph



65nm CMOS process

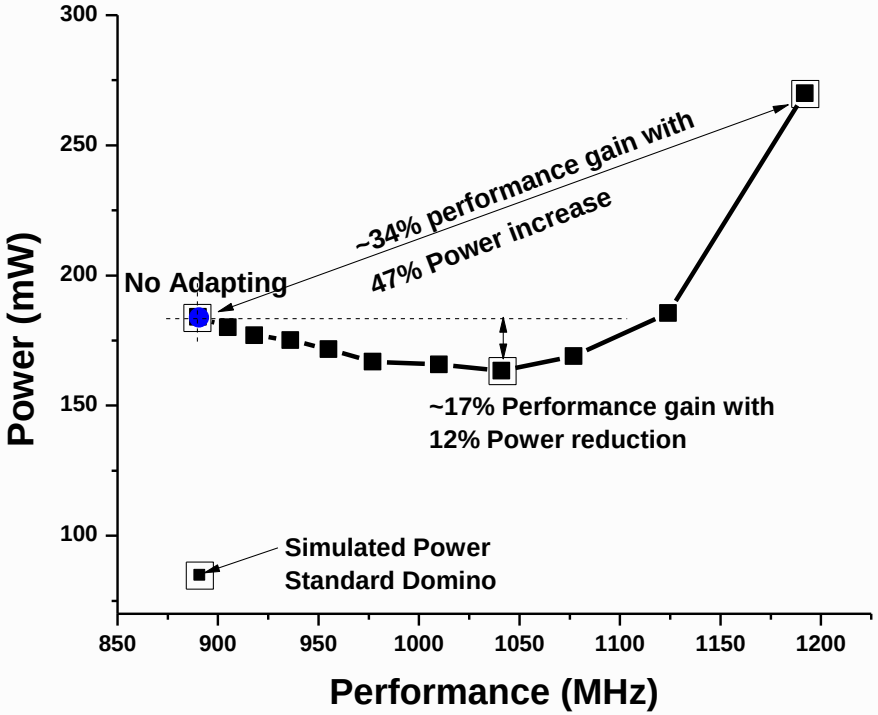
Die Size: 1496µm X 1496µm

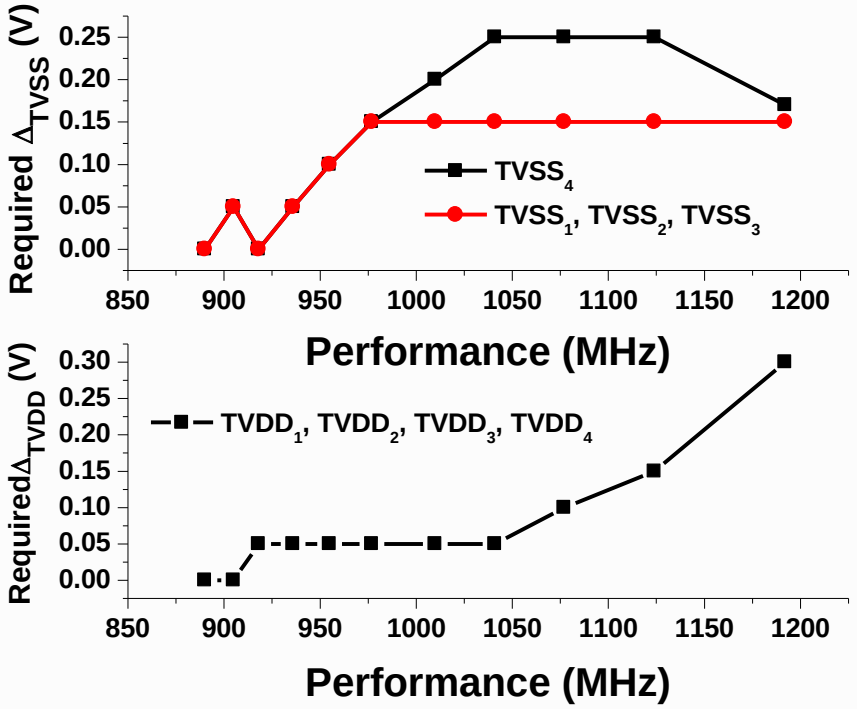
Measured Results



Performance with ART up by 34% by eliminating robustness margins at nominal PVT

Measured Results



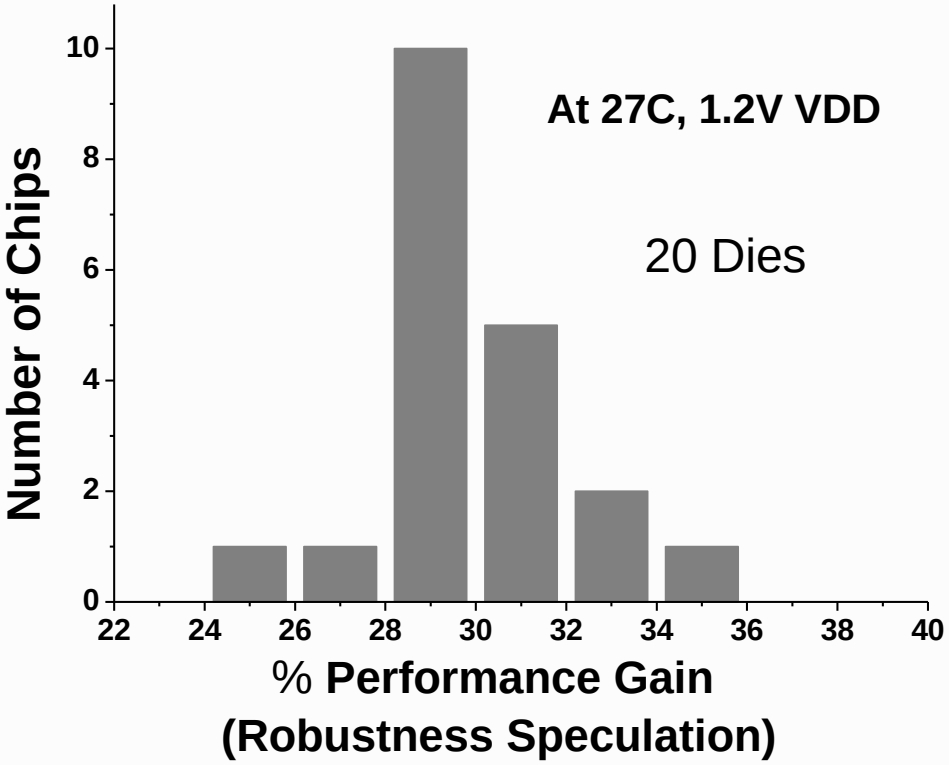


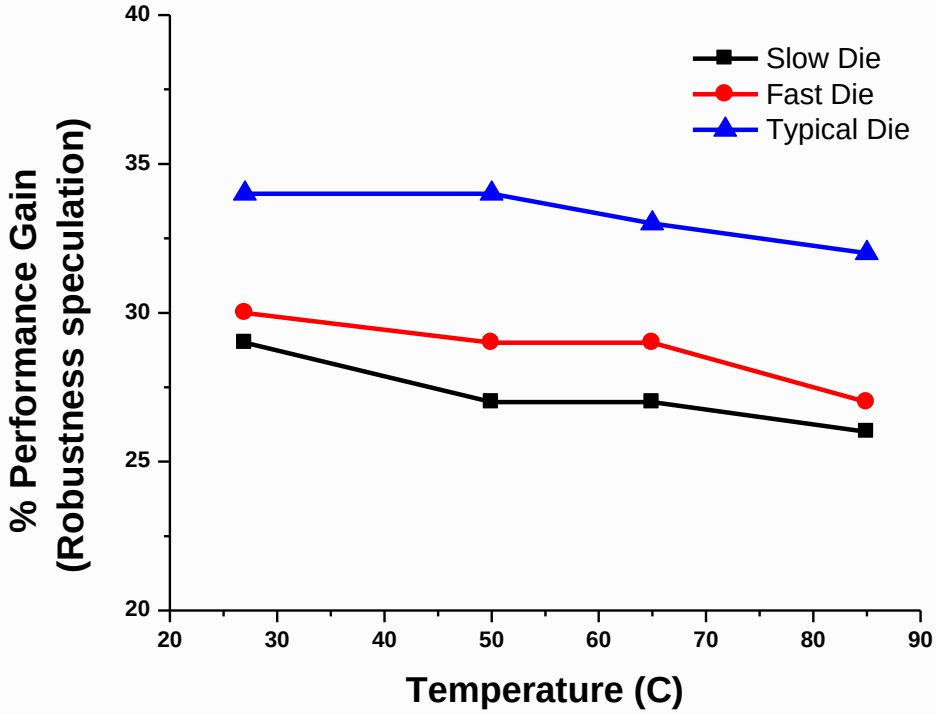
Minimum Power at each frequency

ΔTVDD/ΔTVSS at each measured power-frequency point

- Method applicable to performance constrained logic where performance cannot be obtained by any other means
- Power increases sharply at higher frequencies due to increased short circuit current on the output inverter

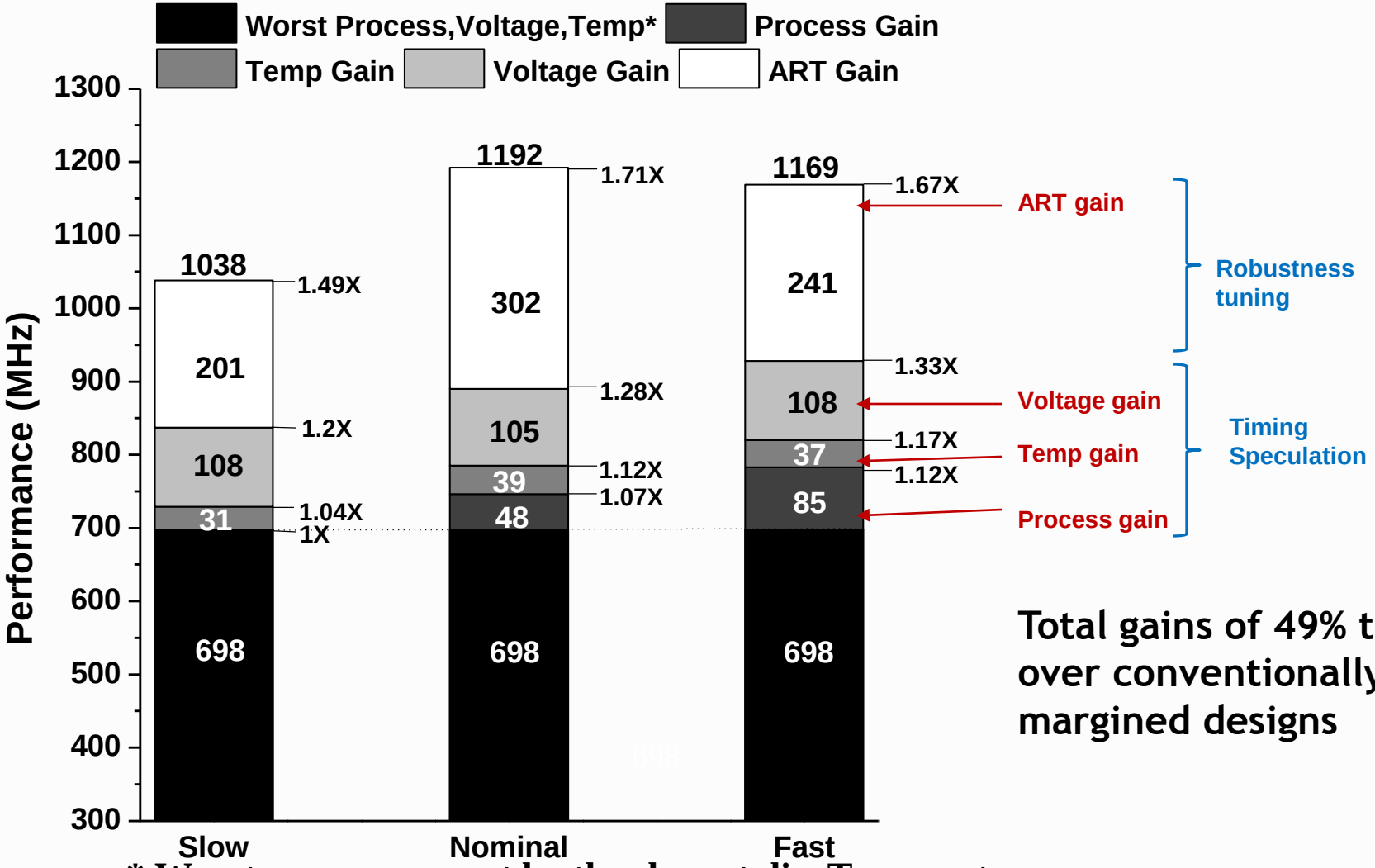
ART Performance Gain





- Measured average % gain due to robustness tuning across 20 dies is ~28%
- % Gain decreases at higher temperatures as gates become less robust

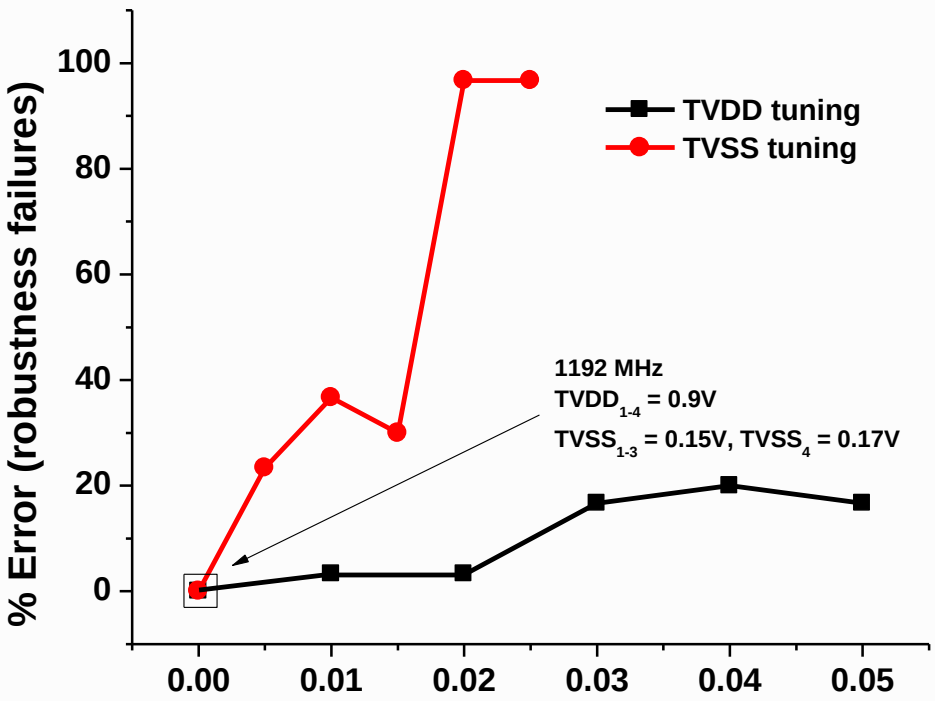
Overall Performance Gain

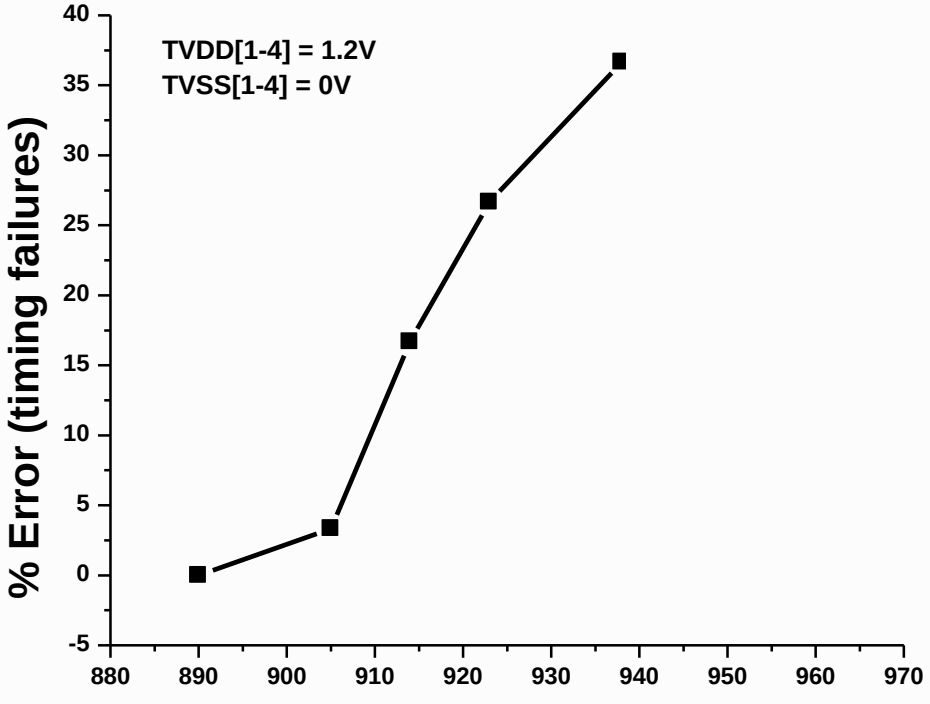


Total gains of 49% to 71% over conventionally margined designs

* Worst process was set by the slowest die. Temperature was set to 85C and Supply was degraded by 10% to 1.08V

Error Rate





Δ_{TVDD}(V), Δ_{TVSS}(V)* * Δ_{TVDD} = 0.9V-TVDD, Δ_{TVSS} = TVSS-0.17V

- Error rate is more sensitive to TVSS than TVDD
  - TVSS tuning also affects robustness of following gate

Conclusion

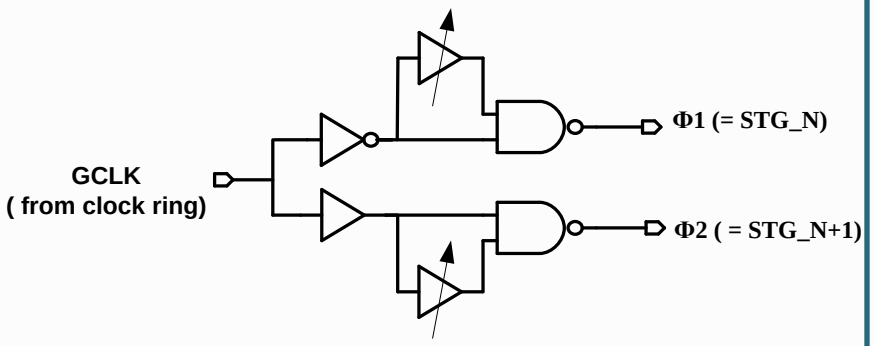
- A new high speed design style called ART Domino was presented
- ART Domino provides overall gain of up to 1.71x over conventional domino
  - 3.2x faster than static CMOS
- Design overhead is amortized over pipeline stage depth

Thank You

Questions?

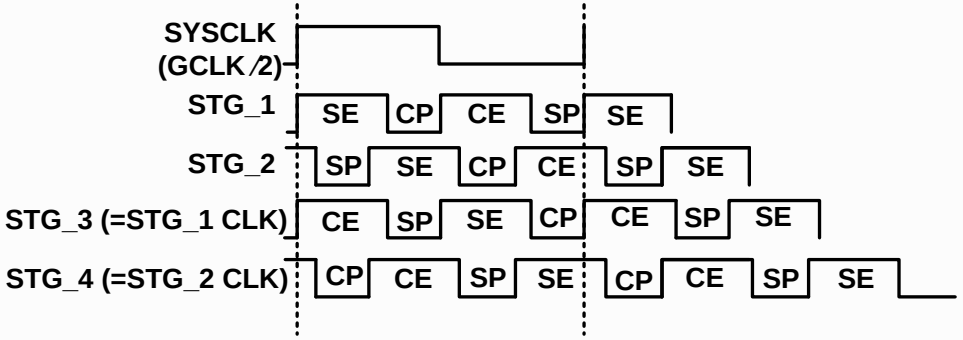
BACKUP SLIDES

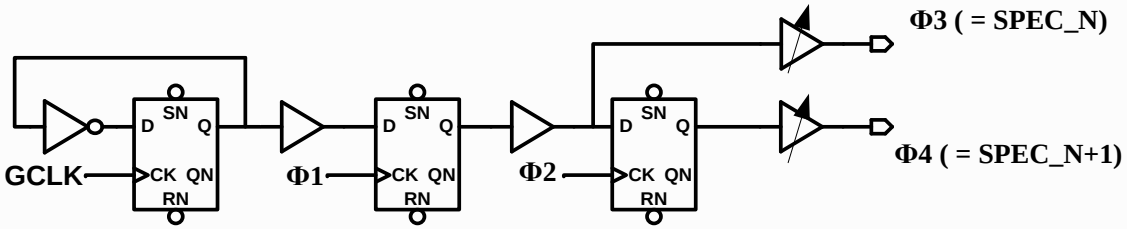
ART Clock Generation

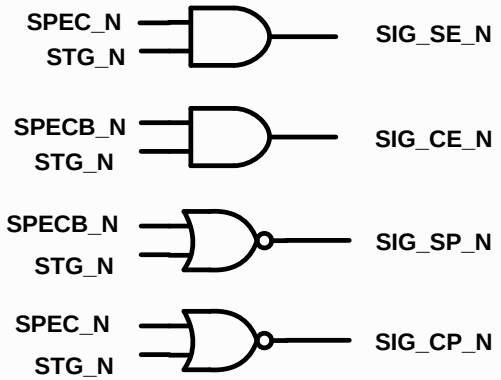


Globally generated clocks with relaxed skew constraints

Example
4-Stage Pipeline Operation Phases

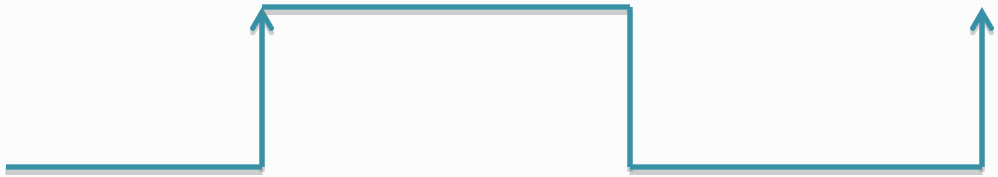




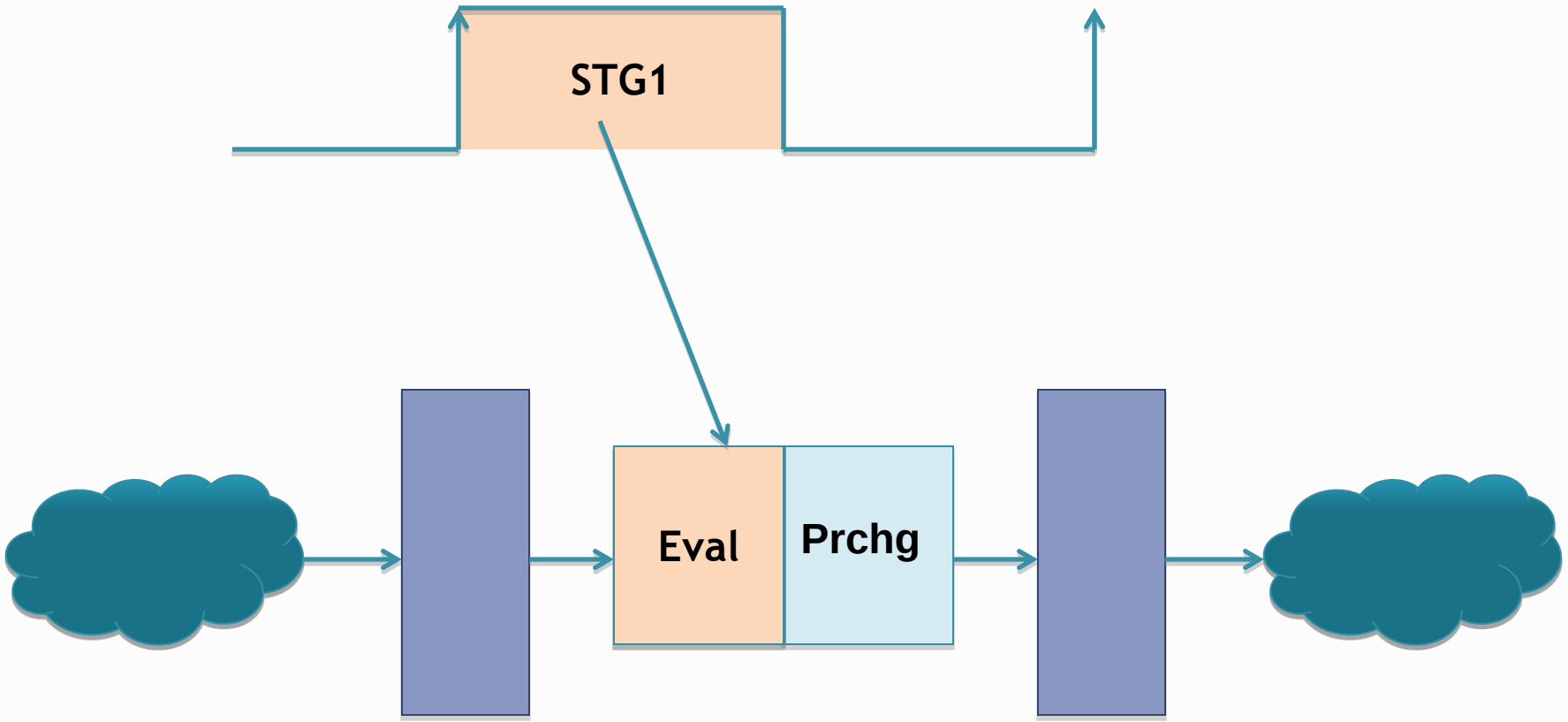


Locally generated clocking signals with stricter skew constraints

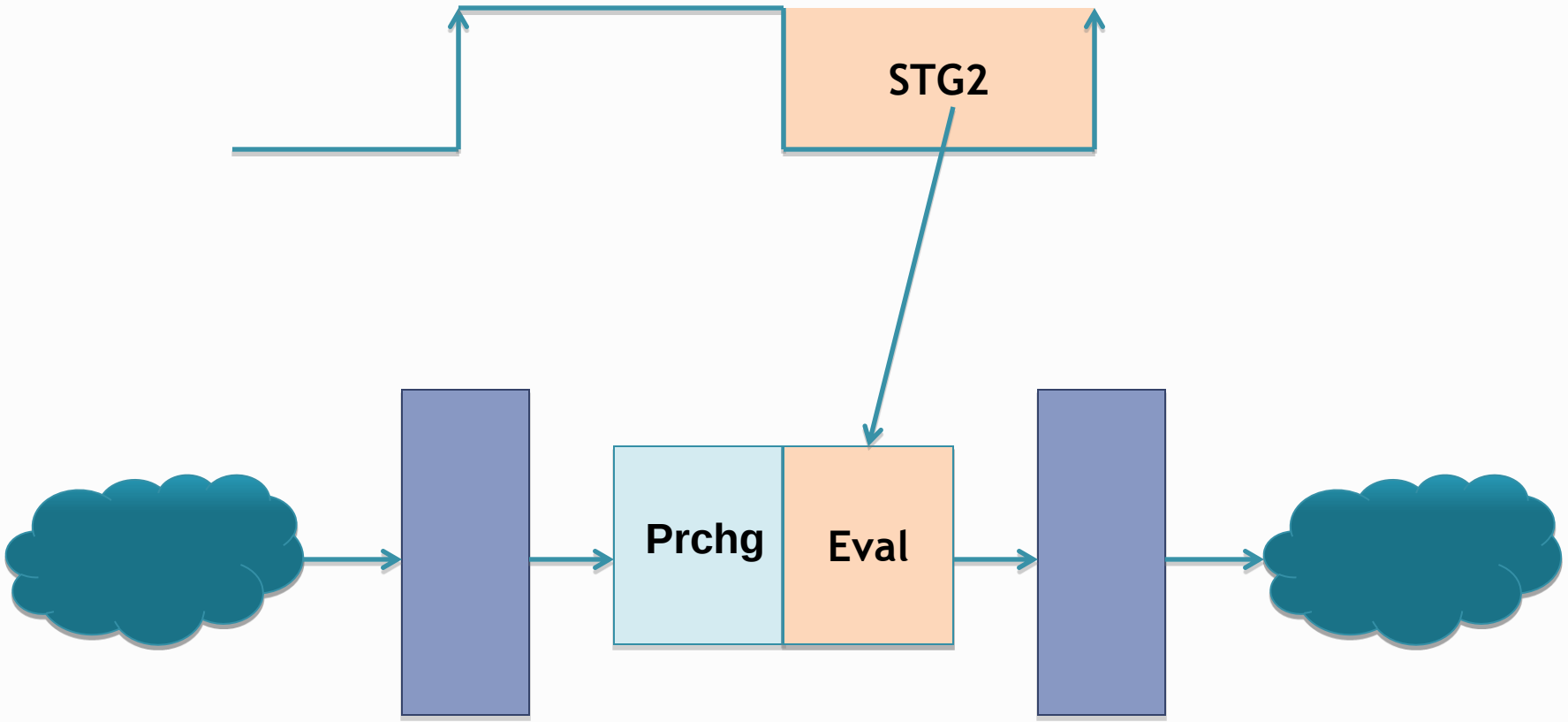
System Implementation
Conventional Domino



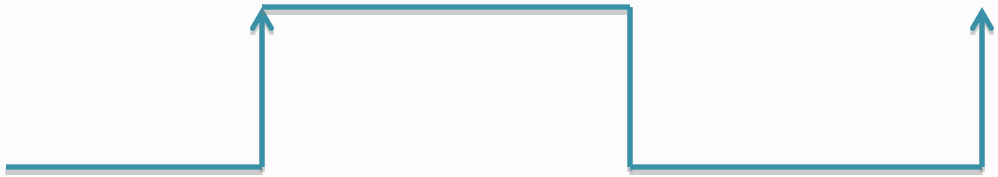
System Implementation
Conventional Domino



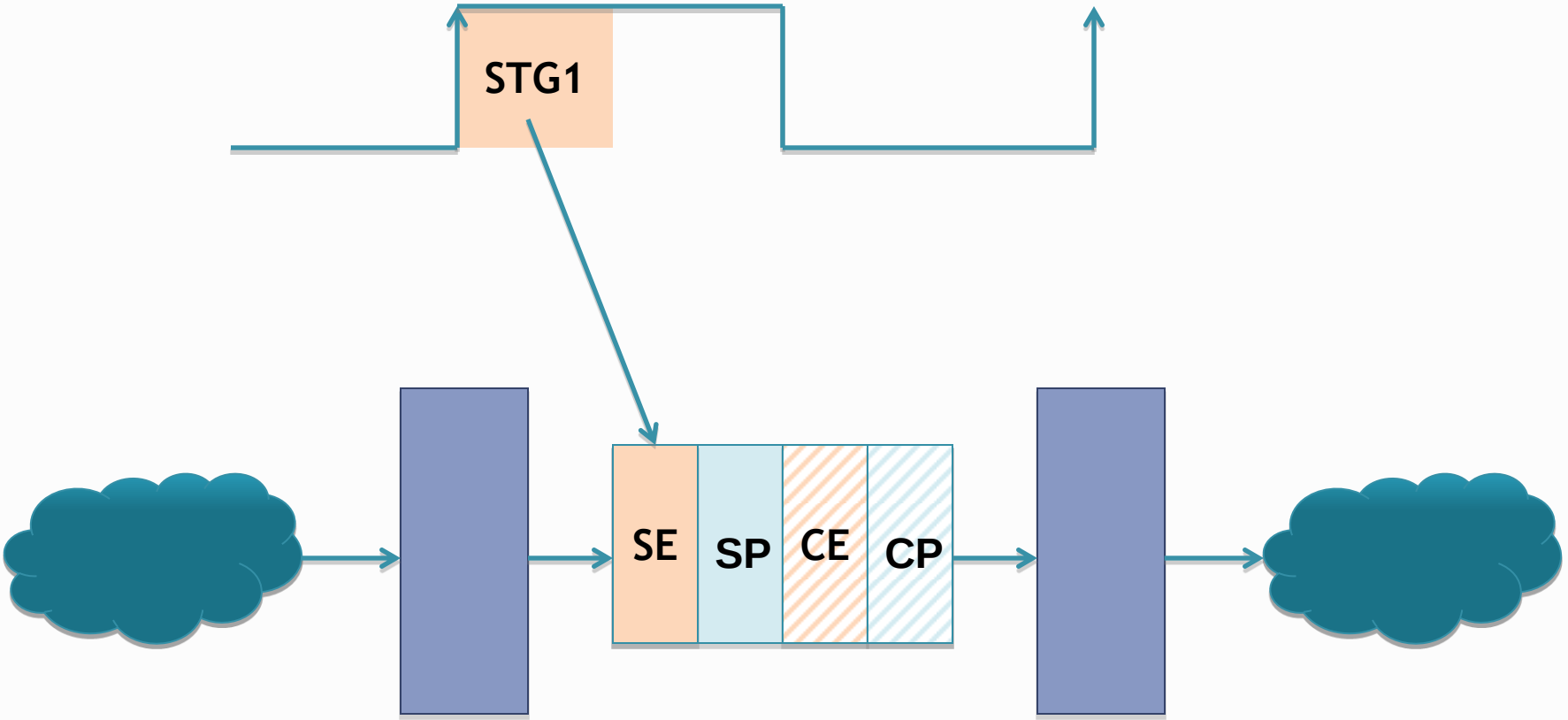
System Implementation
Conventional Domino



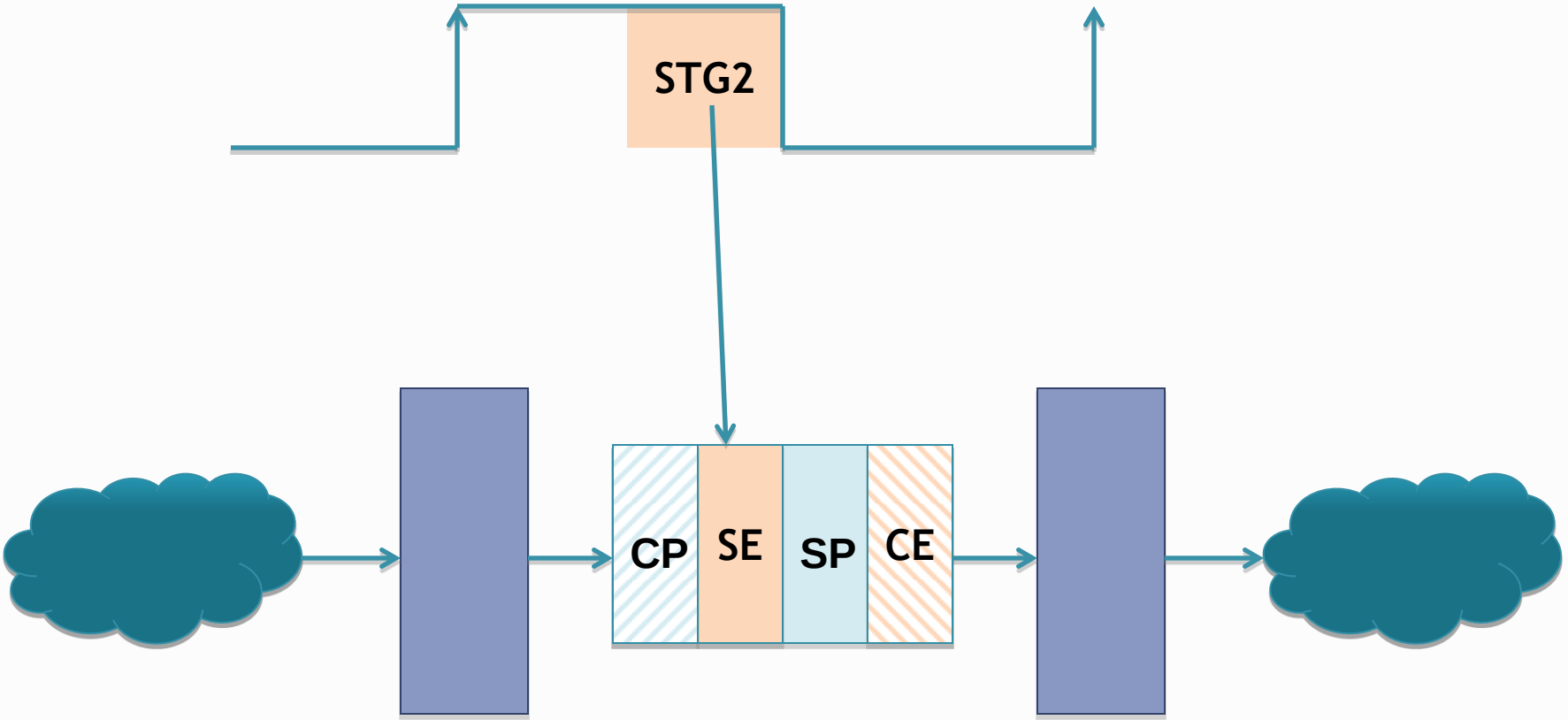
System Implementation
ART Domino



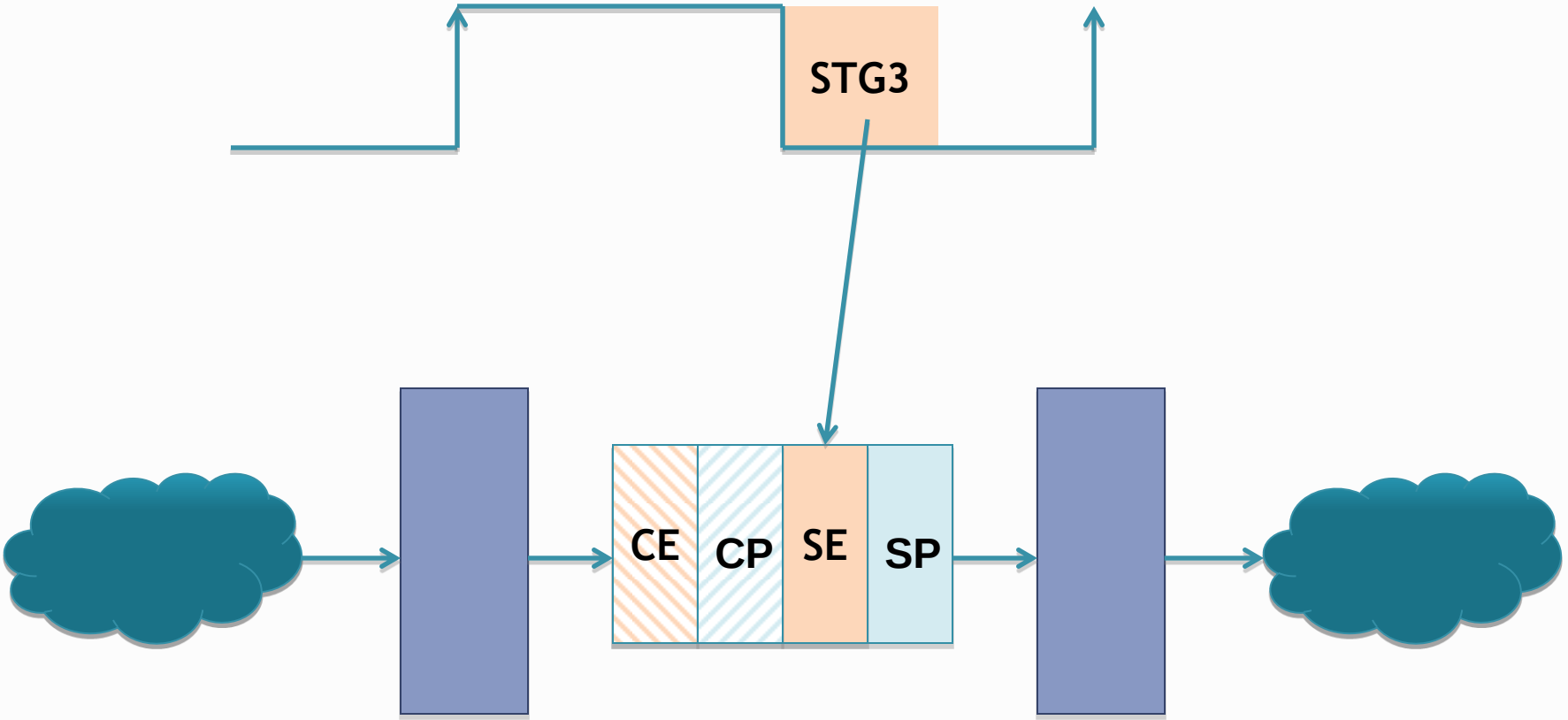
System Implementation
ART Domino



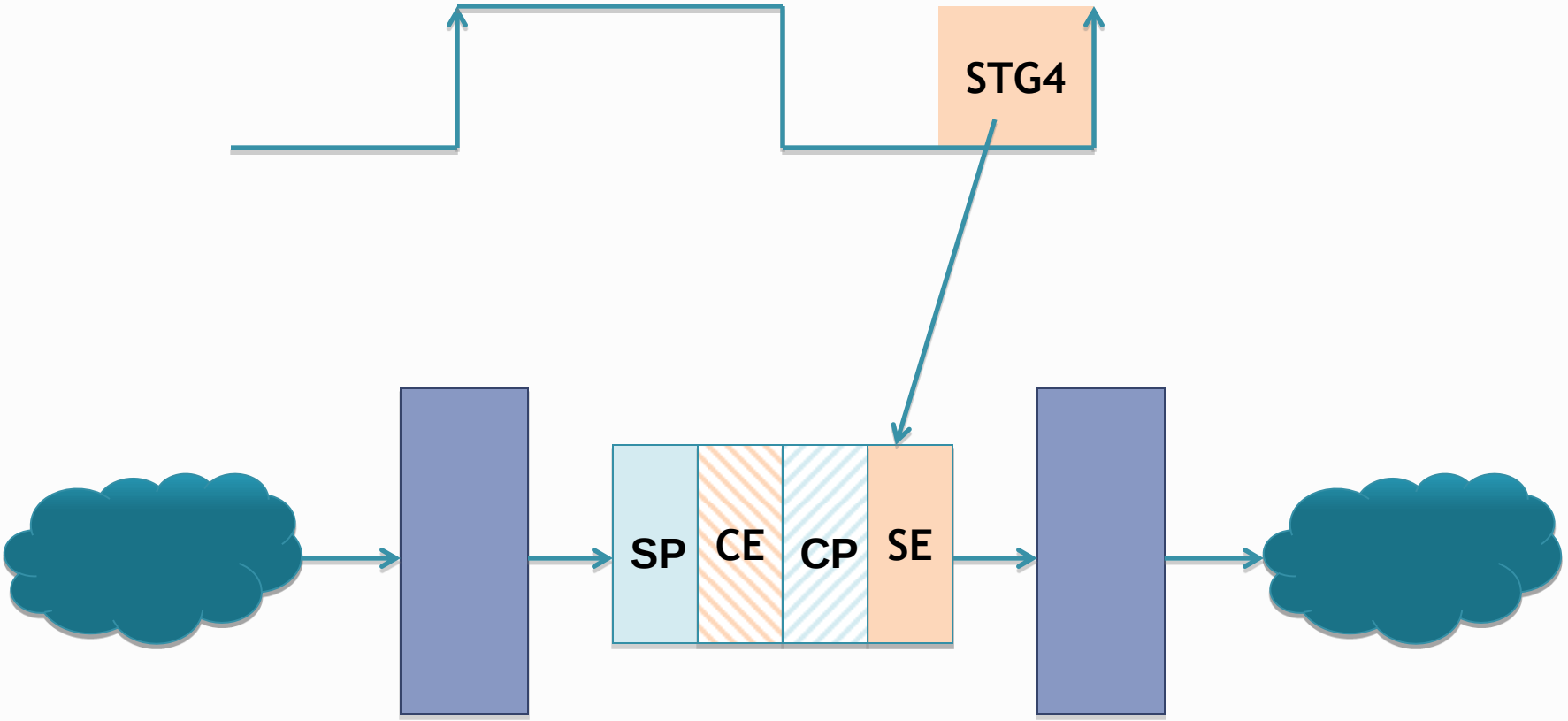
System Implementation
ART Domino



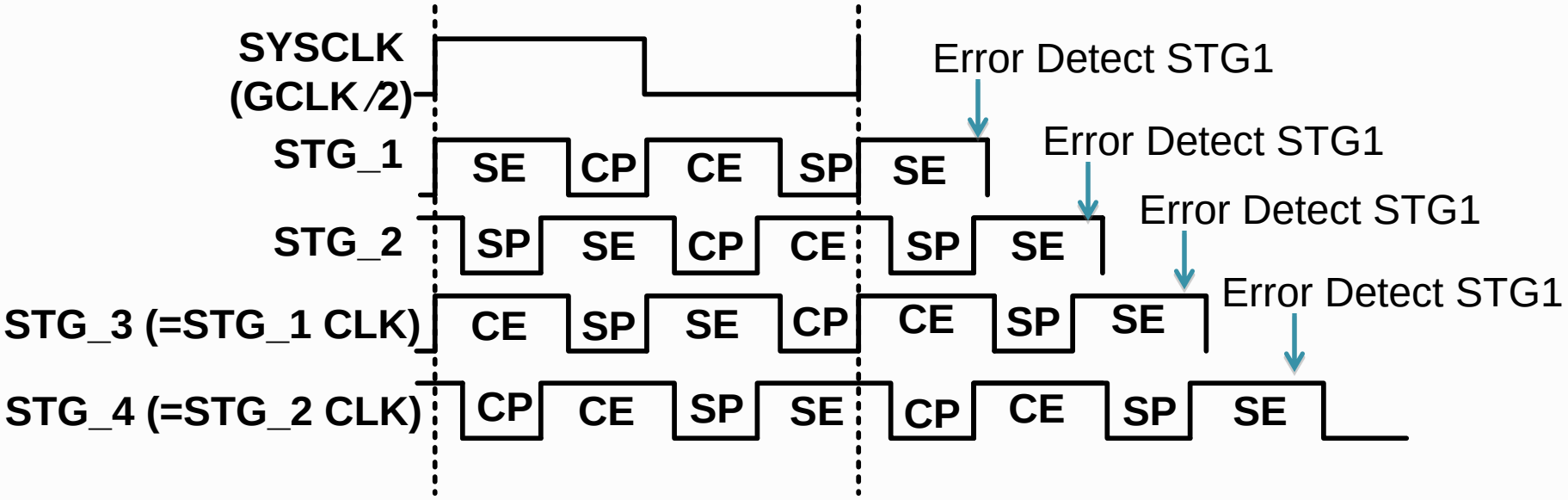
System Implementation
ART Domino



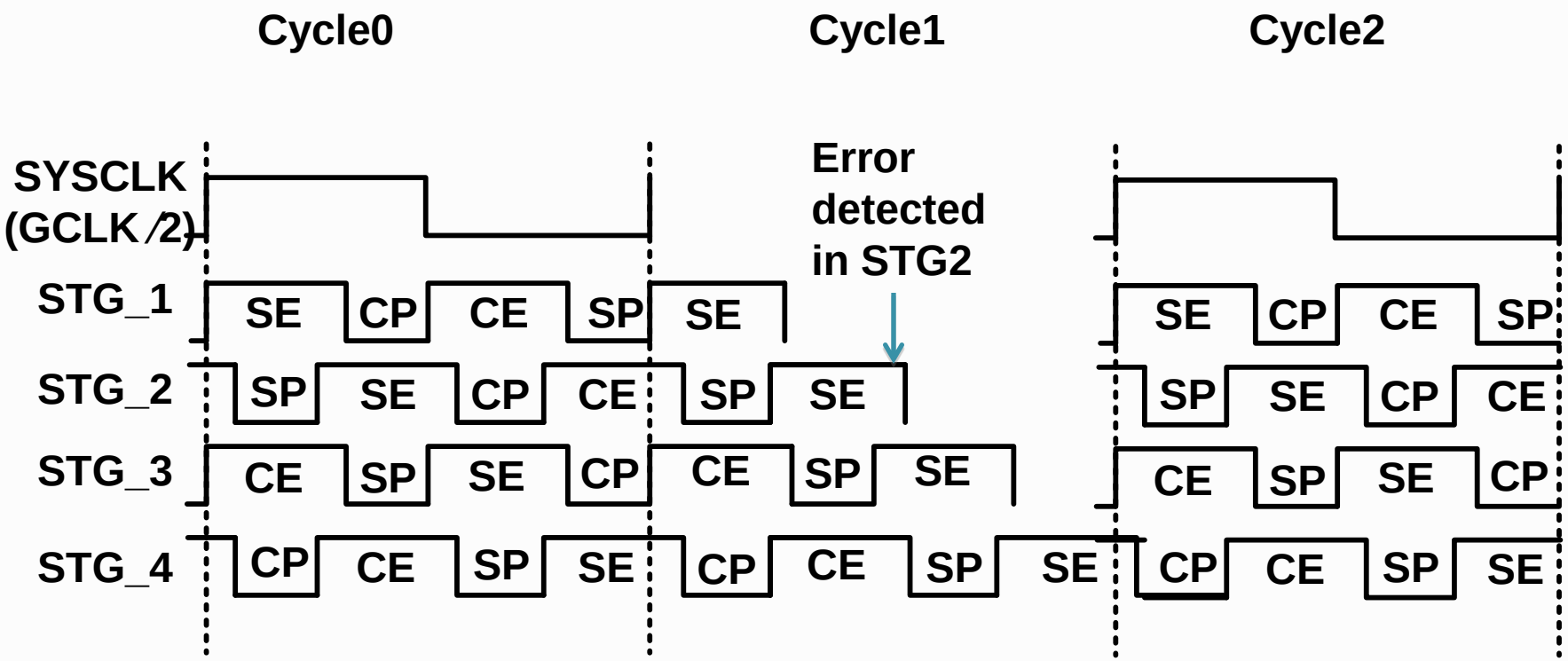
System Implementation
ART Domino



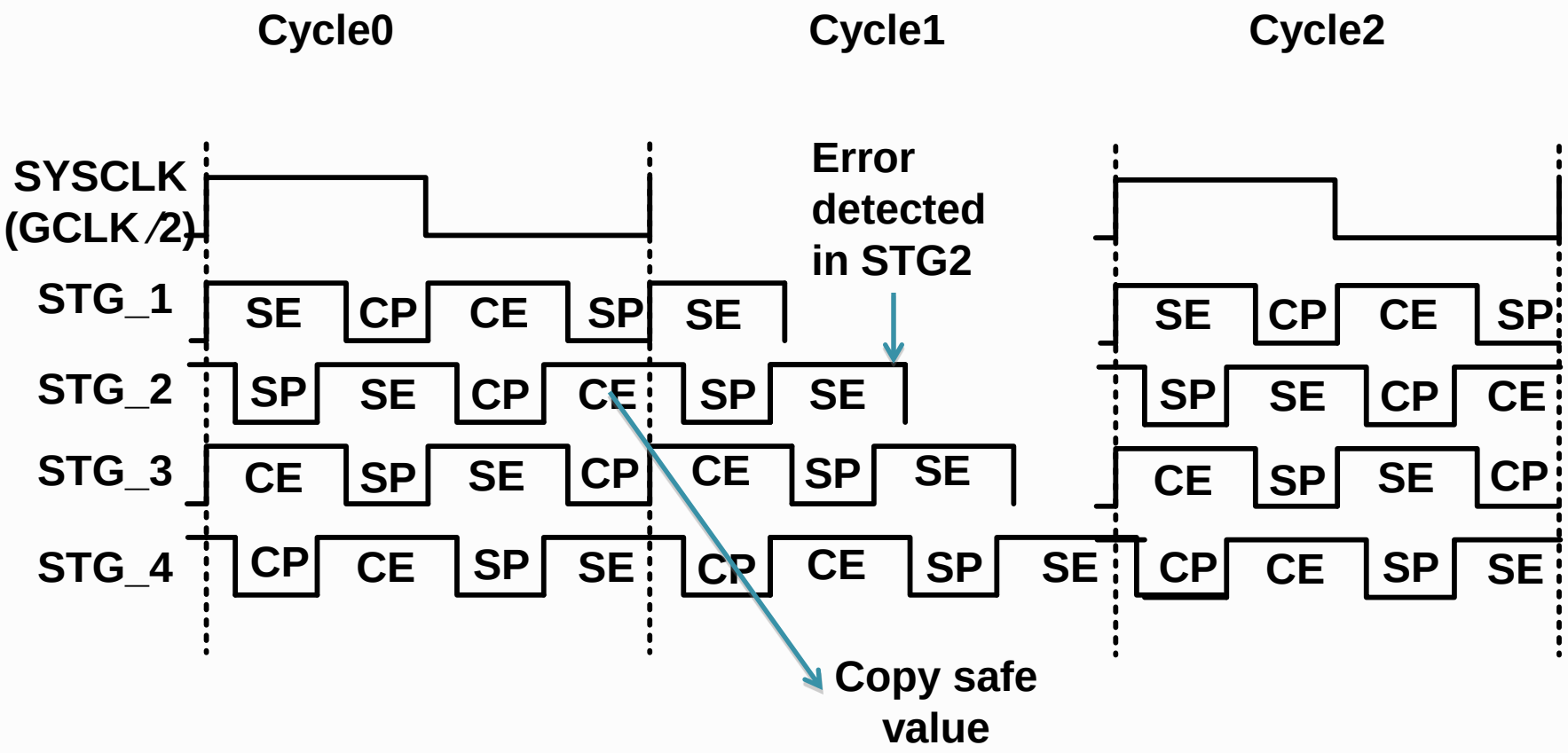
System Implementation
ART Domino



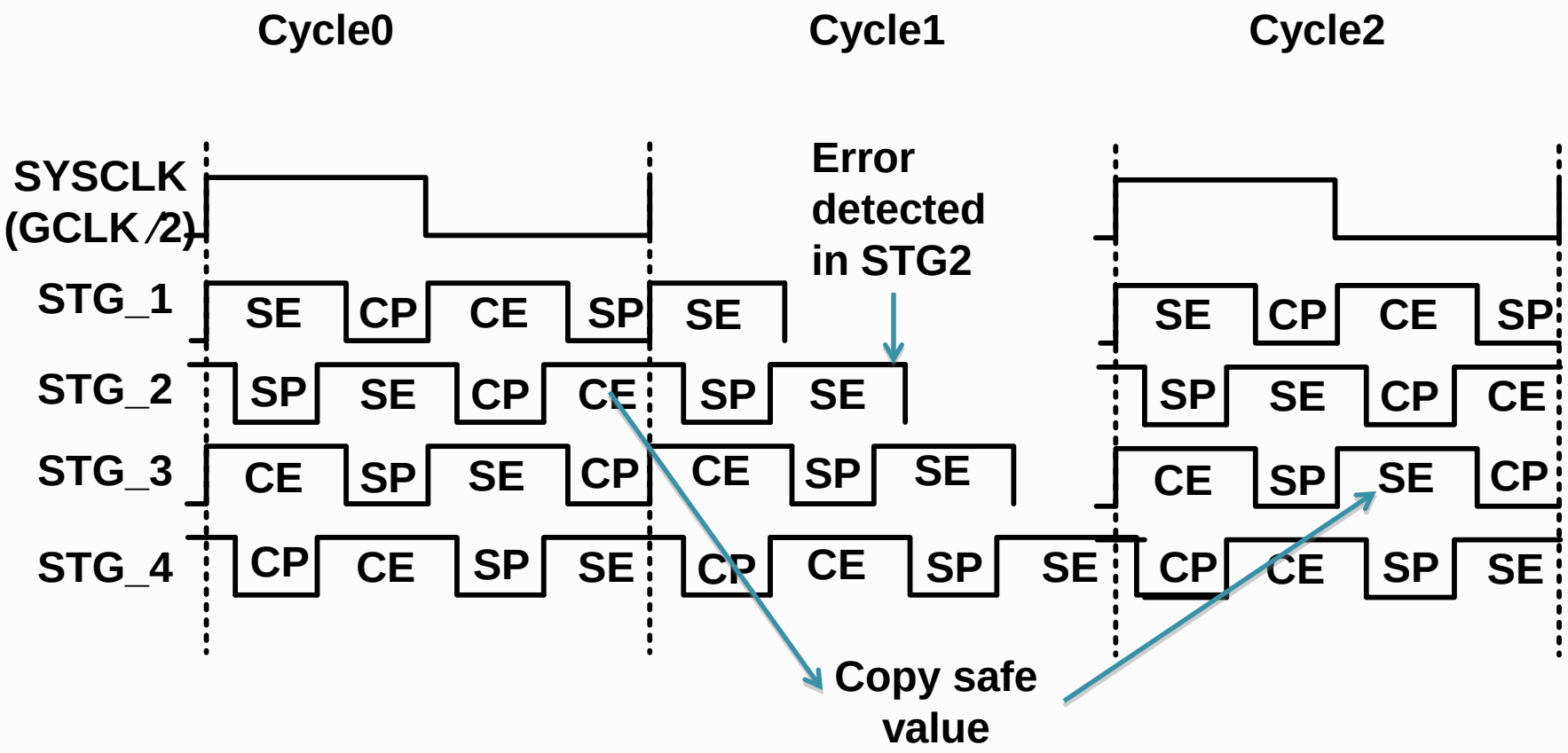
Architectural Replay



Architectural Replay



Architectural Replay



Error Detection Scenarios