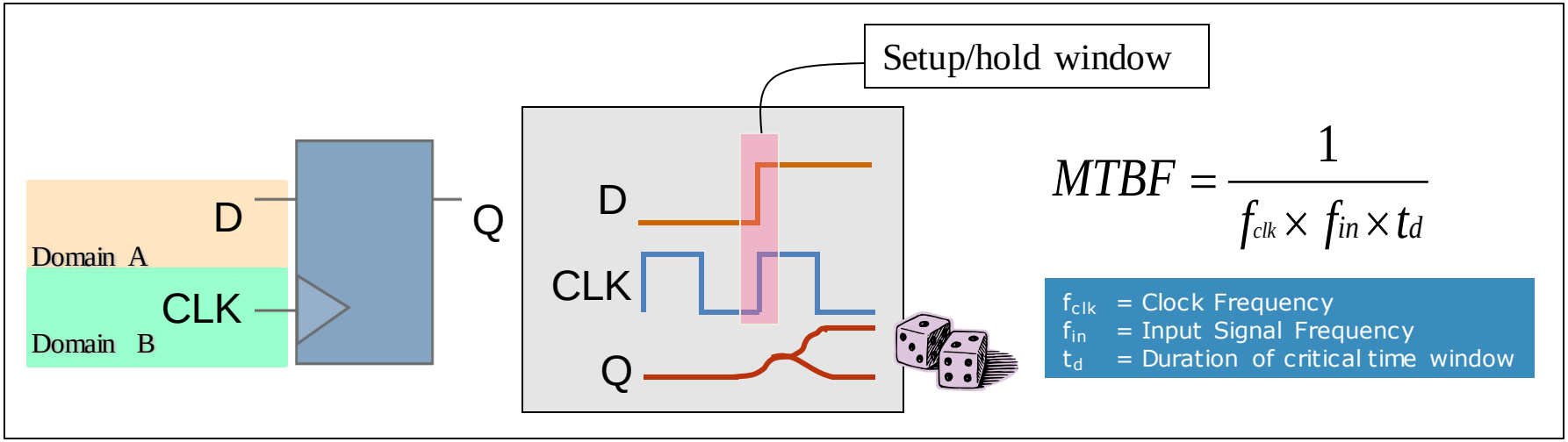
A Pragmatic Approach to Metastability-Aware Simulation

Joseph Bulone, Kalray

Roger Sabbagh, Mentor Graphics

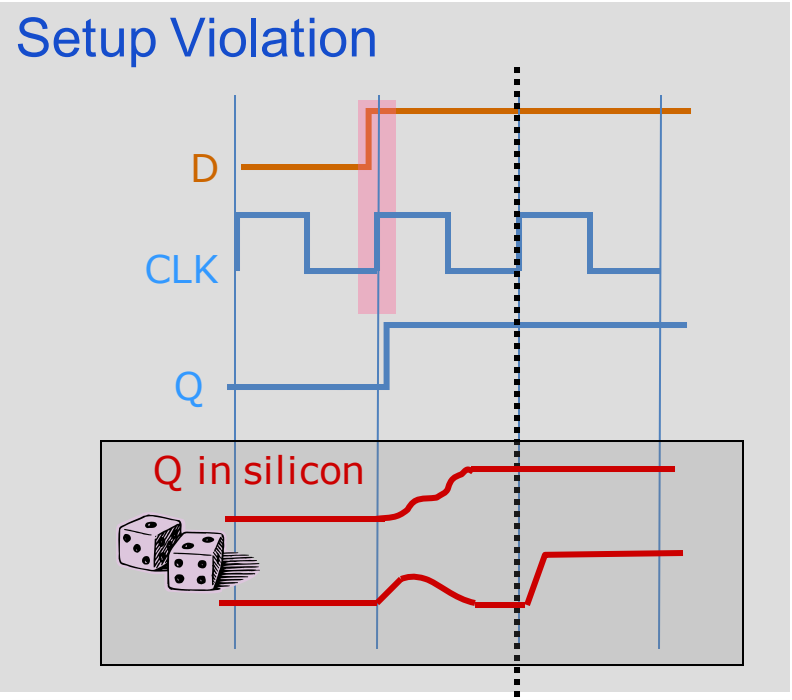
Metastability Effects

- Storage elements can enter a metastable state
- When timing constraints are violated

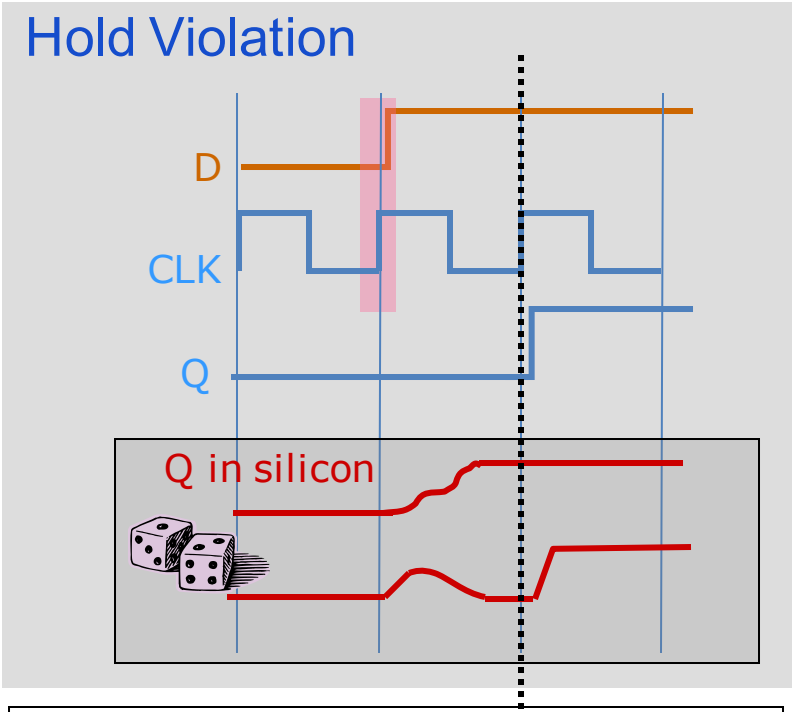


- Cycle-based effect: unpredictable propagation time

Single-Cycle Delay and Bleed-Through

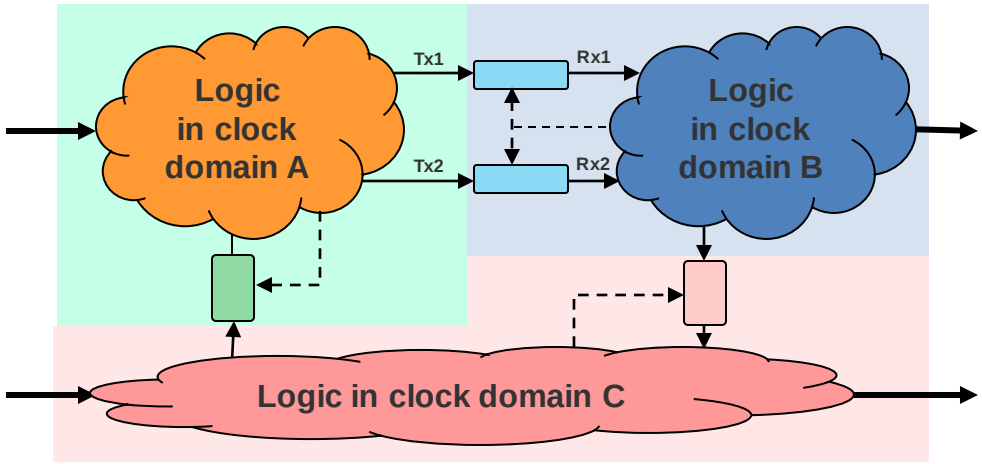


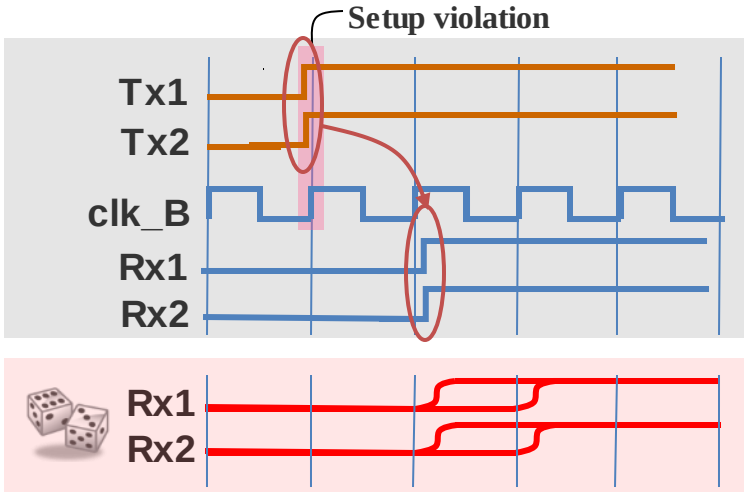
Simulation captures a ‘1’ while silicon produces either a ‘1’ or ‘0’.
Effect: single-cycle delay



Simulation captures a ‘0’ while silicon produces either a ‘1’ or ‘0’.
Effect: bleed-through

CDC Reconvergence



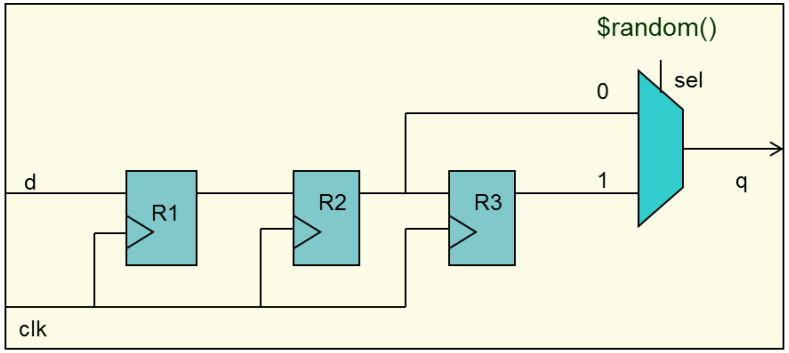


Timing relationships between signals look OK in simulation…
but may be skewed in silicon

If the logic in domain B depends on such timing relationships, it will lead to a **functional bug**

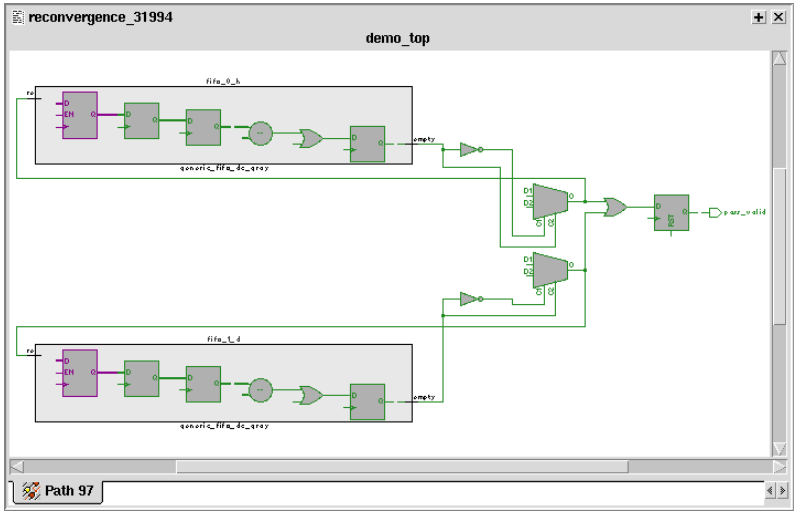
Classical Methods

- Modified synchronizer



- Incomplete
  - Bleed-through not modeled
  - Misses paths with other synchronizers
    - E.g. 3rd party IP

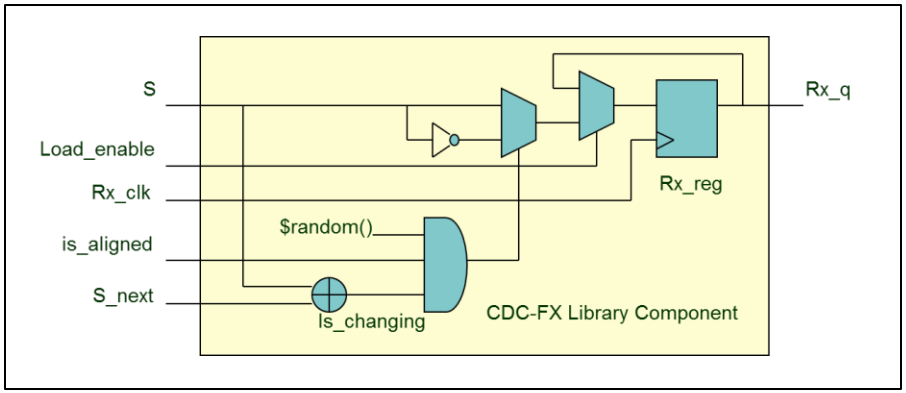
- CDC formal analysis



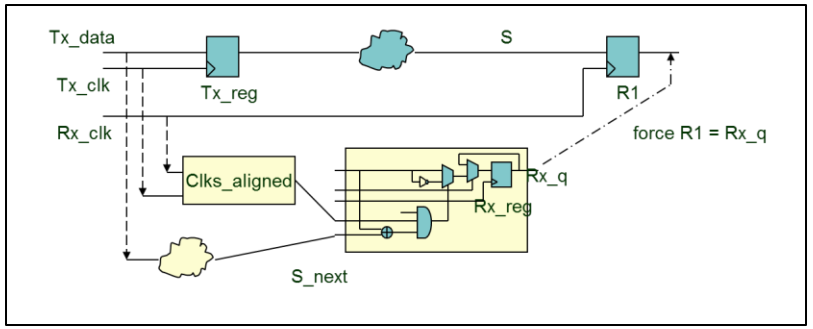
  - Must manually review
  - May miss some deep endpoints

Questa CDC-FX Method

- Metastability effects model



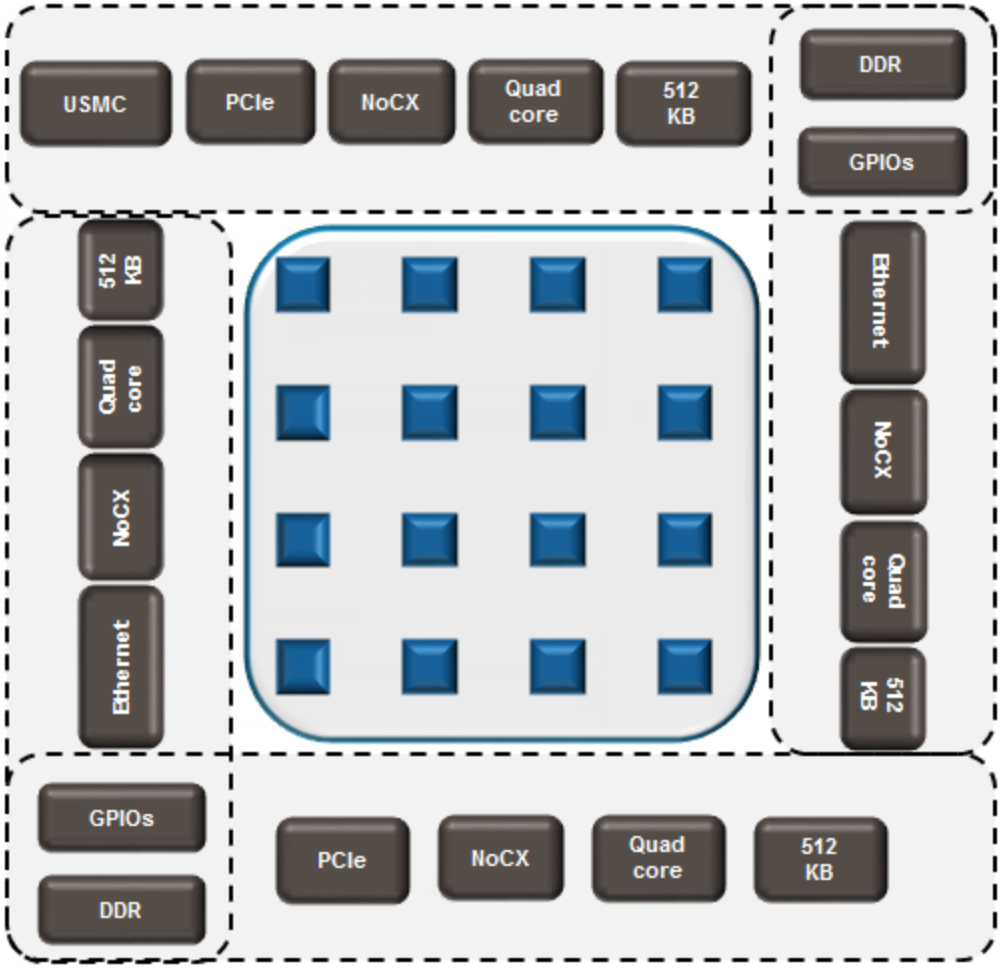
- Bind to each CDC signal



- Models bleed-through and single-cycle delay
- Independent random control for each bit
- Built-in coverage points

- All paths covered
- Independent of synchronizer type

Application to the MPPA-256

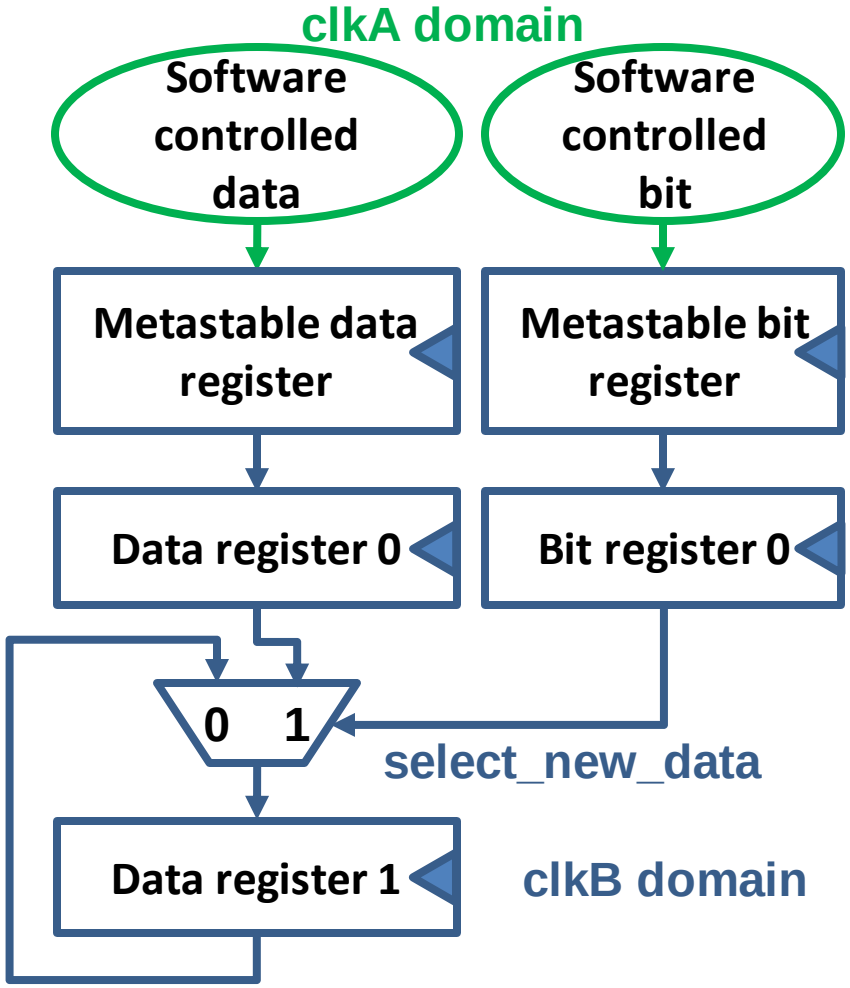


- 160 Mgates
- 28 nm
- 3rd party IPs as
  - DDR controller
  - PCIe controller
  - Ether. controller
  - Flash controller

The static CDC results are a user issue

- Extracted CDC scheme classification
  - Well-known and formally proved as correct
  - Conditionally proved thus formal investigation
    - OK
    - KO
    - No conclusion
  - Missing synchronizer
  - Not extracted e.g. too deep reconvergence
- Small in-house IPs or synchronizers
  - Can be controlled and managed
- Complex 3rd party IP
  - Not manageable: too many (> 10 000)

CDC correctness scheme can be stimuli/constraint dependent



- Correctness meaning ?
  - Data register 1 interpretation ?
- How to ensure correctness ?
  - Input stimuli constraints
  - clkB skew and clkA frequency
- No general formal model including metastability effects and constraints

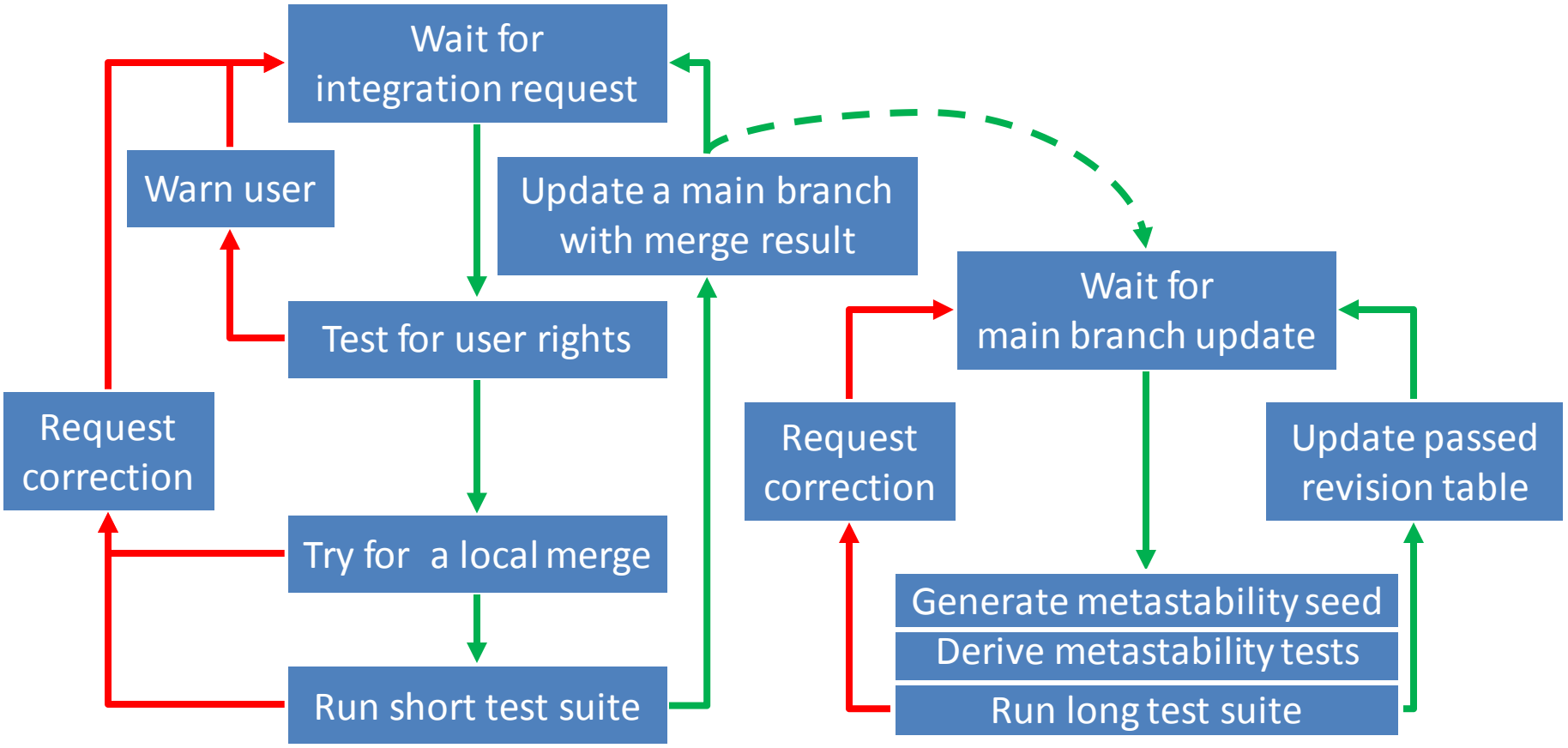
Some partial solutions

- Systematic waiving
  - Usually too risky
- Specific mode selection
  - Time consuming
    - Knowledge requirement
    - Number of modes
  - Mode description
  - Mode transitions
- Increase convergence depth threshold
  - Time consuming
    - Knowledge requirement
    - Manual analysis

The metastability-aware simulation

- Instrumentation thanks to static analysis results
  - Automated
- Seed choice
- Reception clock window parameter definition
  - Too large >= 100% of clock cycle
    - may lead to false errors
  - Too small < 100% of clock cycle
    - may lead to missed metastability potential events
  - Our choice = 99%

Metastability-aware simulation within continuous integration flow



Practical results

- Low impact onto overall simulation time
  - Setup/compilation time: 2x
  - Execution time: below parallel execution variability
- DDR controller bug observed on FPGA platform
  - Diagnostic in metastable-aware simulation
  - Reconvergent paths via 2 asynchronous FIFOs
- Flash controller was CDC behavior dependent onto software code

Conclusion and future

- Static approach is not enough especially with 3rd party IPs
- Dynamic results will depend on stimuli quality
- Expectations as user
  - CDC coverage metrics improvement
  - Qualification notion extension
  - Formal verification environment based on
    - Formal modeling of metastability effects
    - Formal constraints on the use model
    - Formal modeling of expected behavior taking into account metastability effects

Questions