Multiple-clock Domain FPGA Designs: Challenges & Solutions

Reuven Dobkin, CTO

SEFUW, ESTEC, 2018

vSync Circuits

- EDA & IP Company

- Mission:
  - Develop and provide our customers with
    Integration and ***Verification*** solutions
    for Multiple Clock Domain Designs

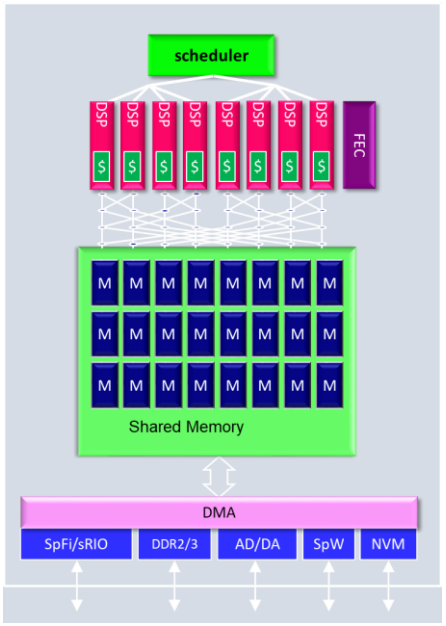
Outline

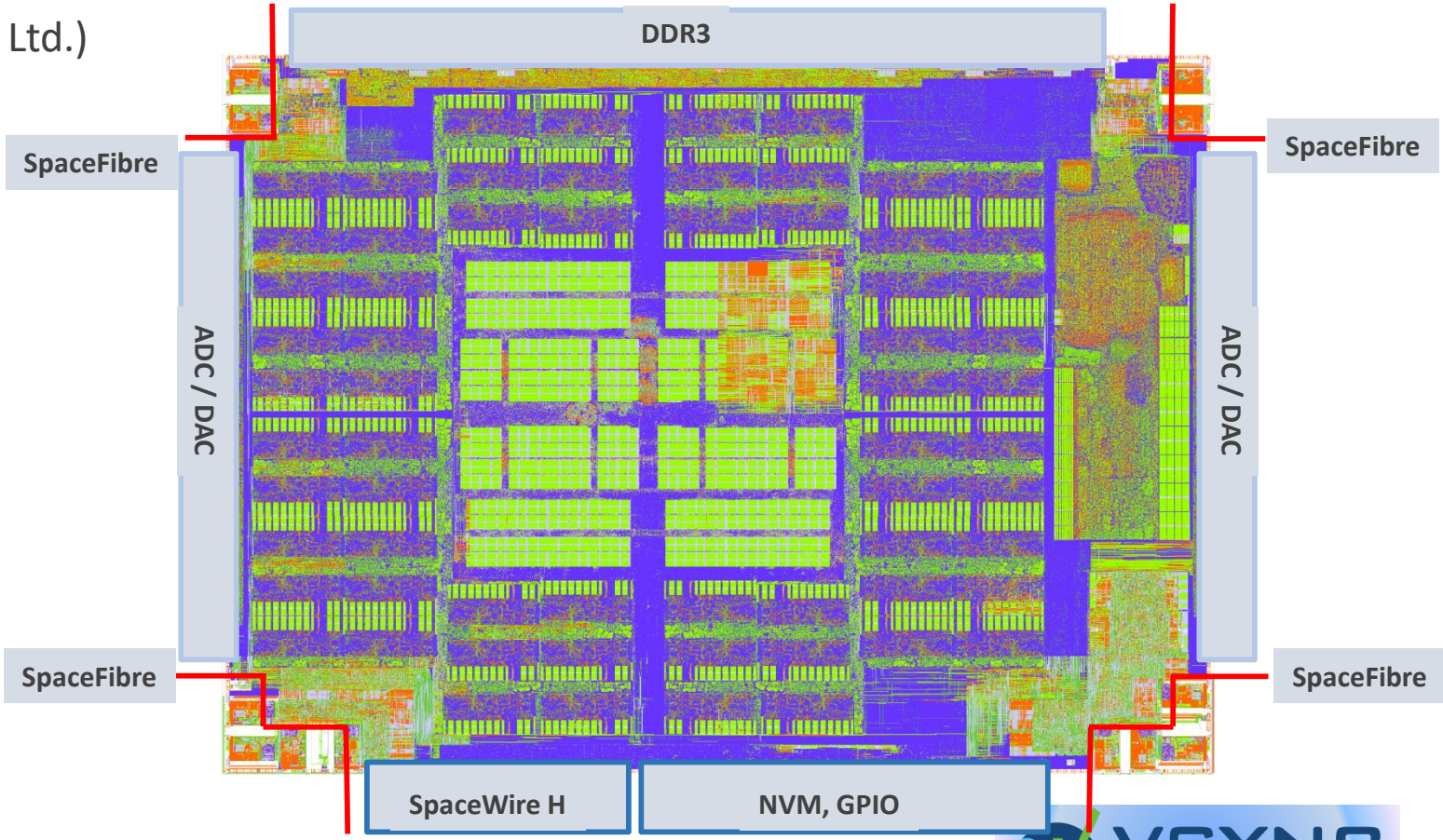
- Multiple-Clock Domain (MCD) Designs
  - A few anecdotes
  - Tips & Tricks
- 3 MCD Challenges
- A note on Vincent (on whom?)

VSYNC circuits

Multiple Clock Domains? A “space” example

- Rad-Hard 64-core DSP-CPU (Ramon Chips Ltd.)
- CDC Sign-off: by vSync CDC Platform
- Fully Functional after fabrication





REF: R. Ginosar, et al., “RC64: High Performance Rad-Hard Manycore”, ESA DSP Day 2016

Recall: How to?

Make it wrong

- Avoiding Synchronization
- One Flop Synchronizer
- Sneaky Path
- Greedy Path
- Flakey Protocol
- Async Clear
- DFT Leak
- Power optimization Leak
- Pulse Synchronizer
- Slow-to-Fast Synchronizer
- Parallel Synchronizer
- Reconvergence path
- Conservative Synchronizer
- Glitching control path
- Bad Constraining

REF: R. Ginosar, Fourteen Ways to Fool Your Synchronizer, ASYNC 2003

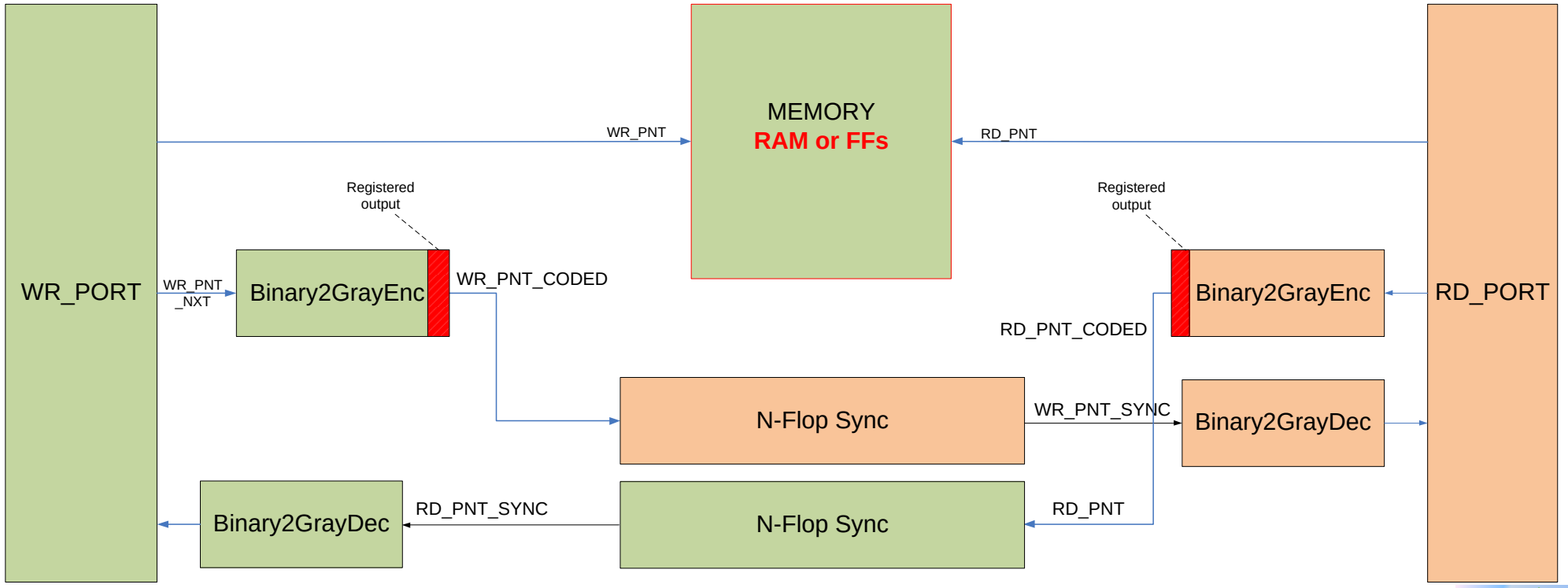
Make it correct

- Universal synchronizer
- Handshake event driven synchronizer
- FIFO gray-code based synchronizer
- Mesochronous synchronizer
- Periodic Synchronizer
- Predictive synchronizer
- Adaptive Synchronizer
- Local-delay latching synchronizer
- Asynchronous reset synchronizer
- Glitch-free gator
- Glitch-free clock switch
- Quasi-static synchronizer
- Fast h/s synchronizer with MTBF trade-off

R. Dobkin and R.Ginosar, Fast Universal Synchronizers, PATMOS, Sep. 2008
D. Verbitsky, R. Dobkin, S. Beer and R. Ginosar, StarSync: An Extendable Standard-cell Mesochronous Synchronizer, Integration—the VLSI Journal, 2014
R. Dobkin, R. Ginosar and C. Sotiriou, High Rate Data Synchronization in GALS SoCs, IEEE Trans. on VLSI, 14(10):1063-1074, Oct. 2006
R. Dobkin and R.Ginosar, Two phase synchronization with sub-cycle latency, INTEGRATION, the VLSI journal, 2009.

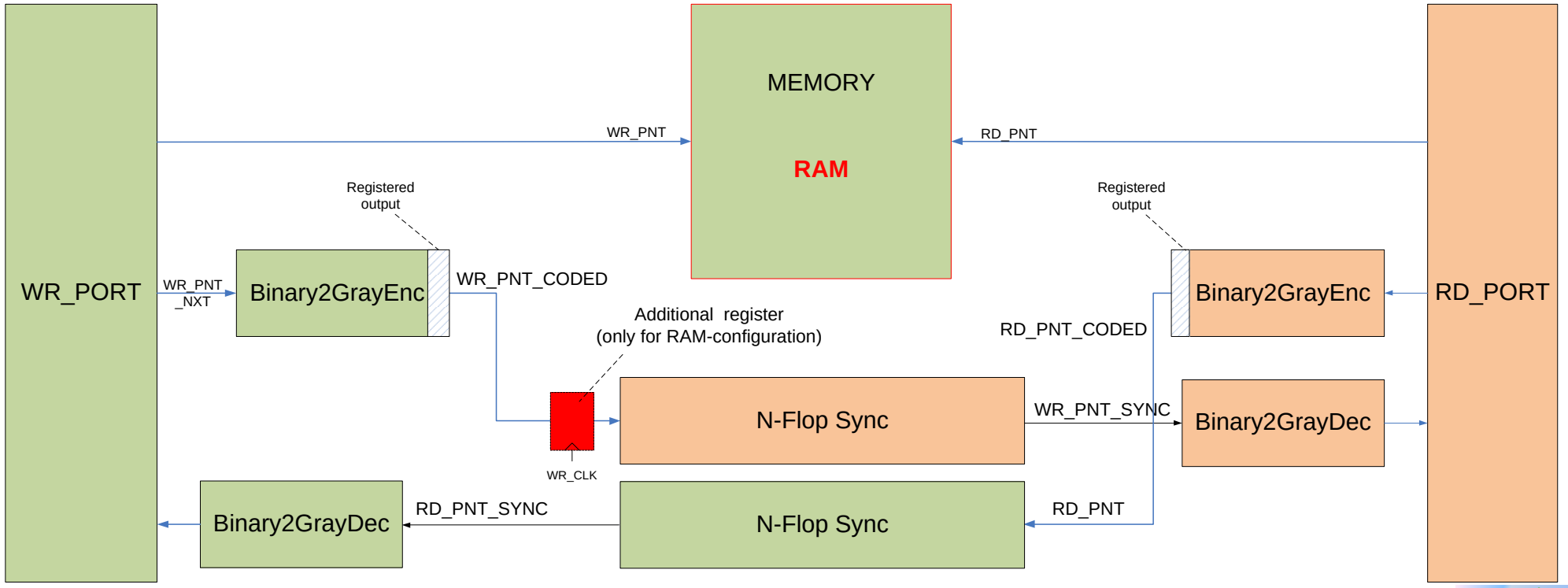
A few notes on FIFO design (1)

- Eliminate glitching into Sync: Gray Encoder output must be sampled



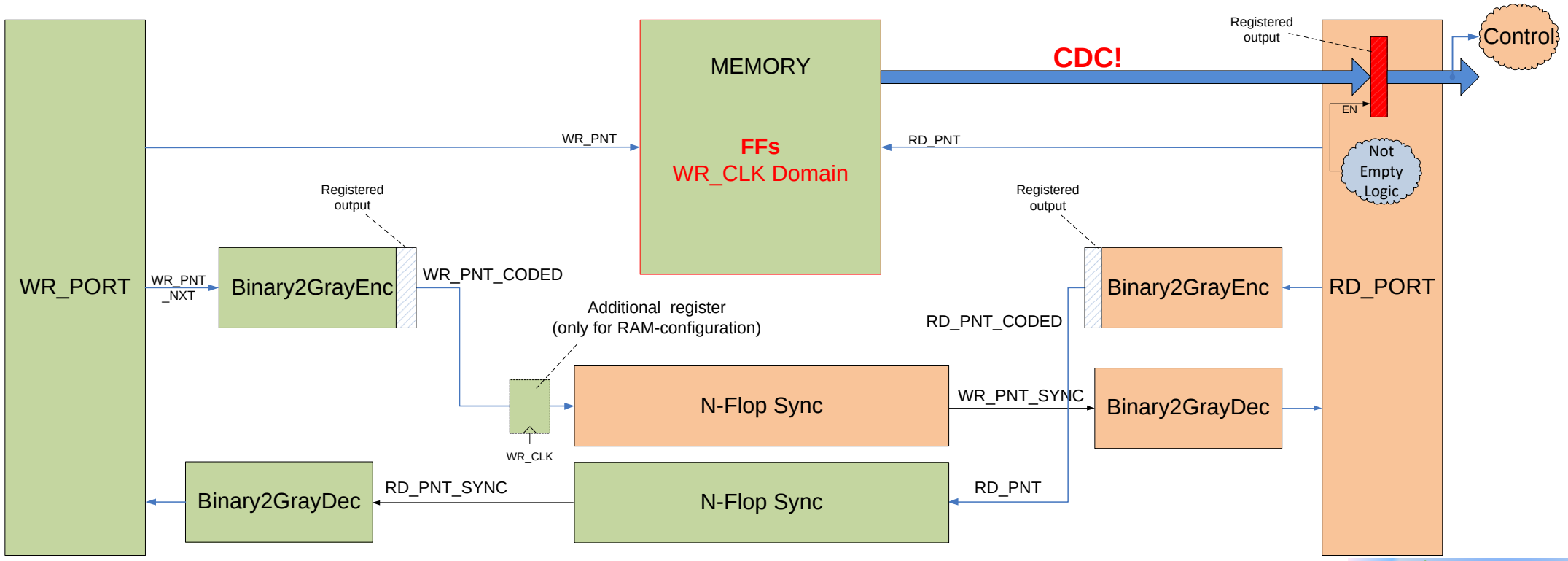
A few notes on FIFO design (2)

- RAM write could be “tricky”: Check out the RAM specification for write latency



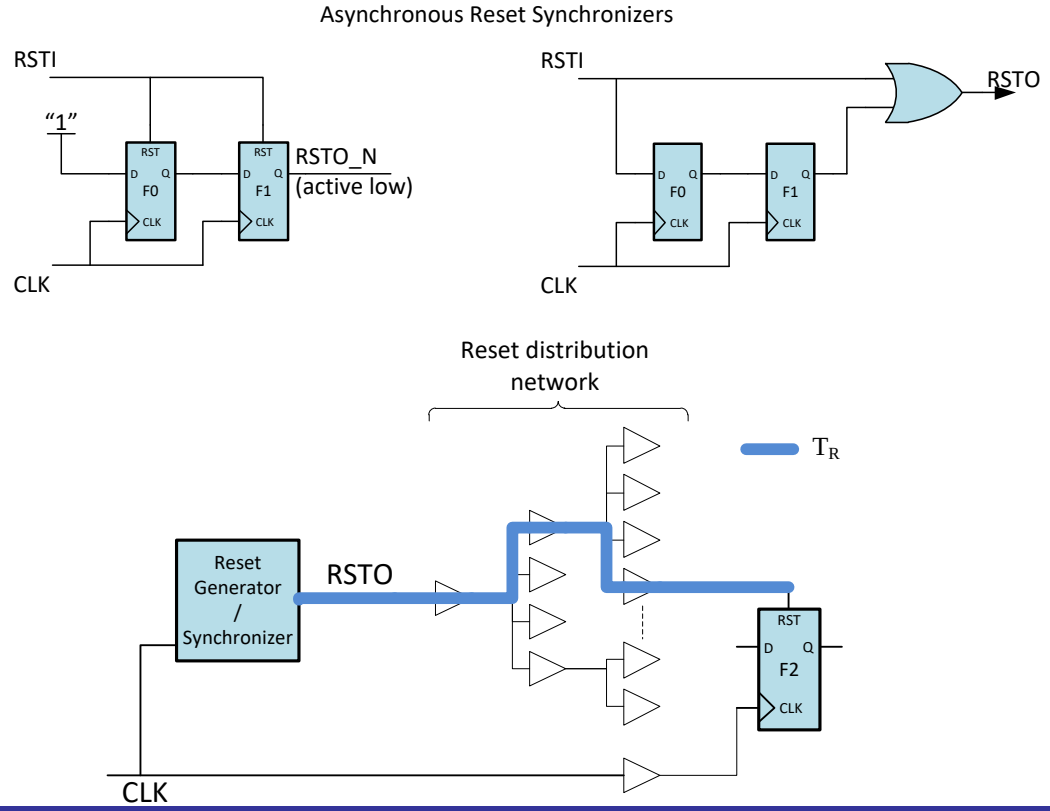
A few notes on FIFO design (3)

- Register Read side data output + enable (*who knows how it is used further…*)



Asynchronous Reset CDC

- Reset must meet setup/hold constrains



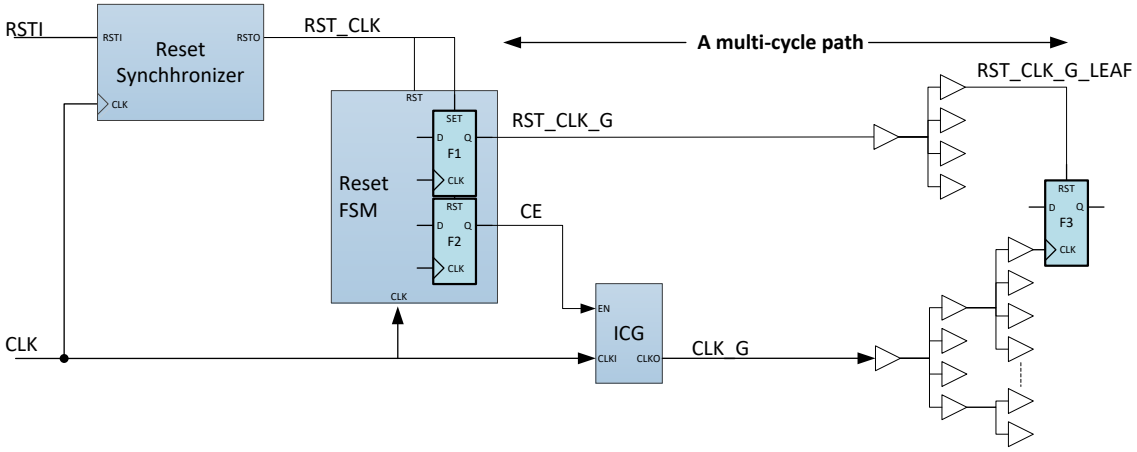
Fast clocks

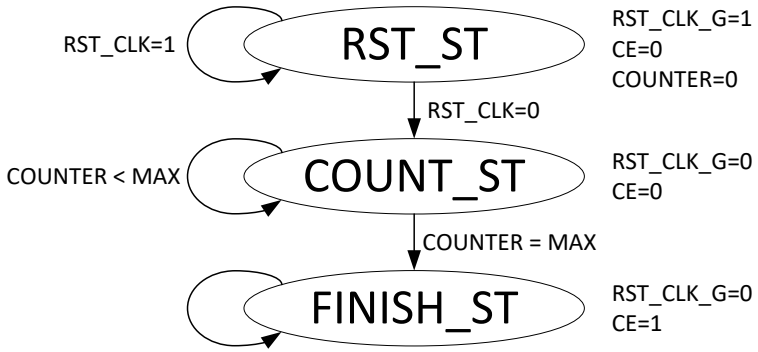
$$T_{CLK} \geq T_R + T_{SU}$$

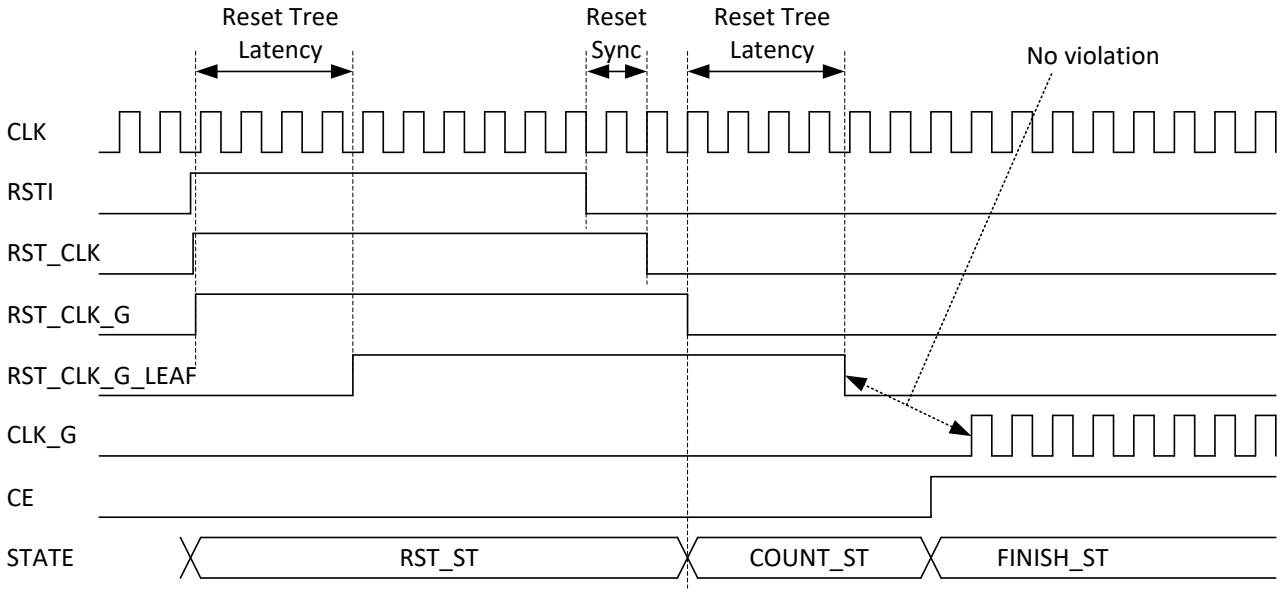
Large designs

REF: R. Dobkin, Asynchronous Reset Synchronization and Distribution, embedded.com, 2017

An elegant solution: The ***Clock-Gated*** Async Reset Synchronizer







Another Good Approach: *Power Up Initialization in FPGA*

- FPGA allows programming memory default state on power up
- Benefits:
  - A **significant** reduction of FPGA global resources utilization
  - Elimination of the related timing issues of the asynchronous reset removal
- Not applicable when:
  - Reset is functional during application run
  - Reset value depends on an external value
- Technically done by replacing asynchronous resets with signal defaults:

  VHDL: *signal my_signal : std_logic := '0';*

  Verilog: *reg my_flop = 1'b0;*

- **NOT** applicable for ASIC!

VSYNC circuits

Multiple Clock Design
Challenges

- Setup: Clock & Reset requirements...
- Complexity: design and verification
- Integration: (Black-box) Third-party IPs

Getting Requirements…

G.I.G.O.

- Clocks setup – Who is Who?
  - Clock relationships
  - Clock sources
    - Async, meso, periodic, etc.
    - Intra Black-box clock manipulations
  - Clock switching (e.g. SpaceWire)
- Reset setup – Who is Who?
  - Reset sequences
  - Power up / @ run

http://safetyandcommonsense.blogspot.co.il/2010/10/garbage-ingarbage-out.html

Clock Scheme
Impacts

- Synchronization solution choice
- CDC verification setup
  - Static
  - Dynamic
- Reliability verification (MTBF)
  - E.g. correlated / uncorrelated clocks
- Constraints generation for CDC
  - Sync types
  - Quasi-static CDC

Class	$\Delta\phi$	Δf	Synchronizer
Synchronous	0	0	None
Mesochronous	ϕ_c	**0**	**Phase compensation**
Multi-synchronous	drifts	0	Adaptive phase compensation
Plesiochronous	Varies	$f_d<\varepsilon$	Adaptive phase compensation
Periodic		$f_d>\varepsilon$	Predictive
Asynchronous			**Two-Flop**

REF: S. Beer, R. Ginosar, R. Dobkin and Y. Weizman, MTBF Estimation in Coherent Clock Domains, ASYNC'19, 2013.

REF: U.S. Patent 8631364, 2014

Clock & Reset Requirements
Solutions

Challenges

- Clocks setup – Who is Who?
  - Clock relationships
  - Clock sources
    - Async, meso, periodic, etc.
    - Intra Black-box clock manipulations
  - Clock switching
- Reset setup – Who is Who?
  - Reset sequences
  - Power up / @ run

Solutions (EDA)

- Auto Clock and Reset setup
  - Clock & Reset trees auto recognition
  - Vendor-IP recognition (e.g. PLLs)
  - SDC
  - Call for user intervention when needed

- Multi-modal analysis support
  - Clock-gating / switching

Multiple Clock Design
Challenges

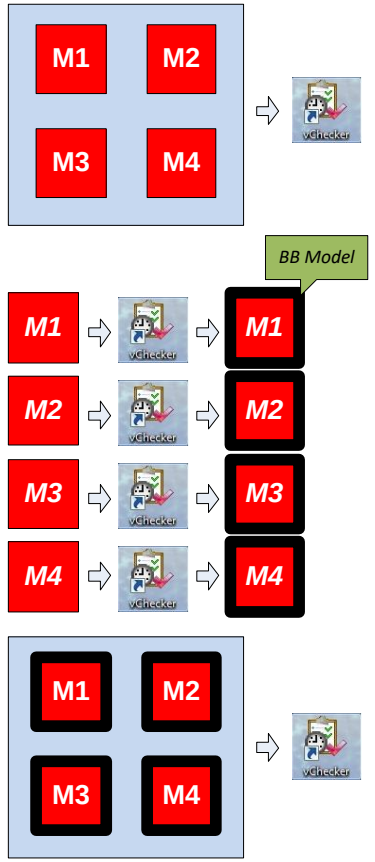
- Setup: Clock & Reset requirements...
- **Complexity: design and verification**
- Integration: (Black-box) Third-party IPs

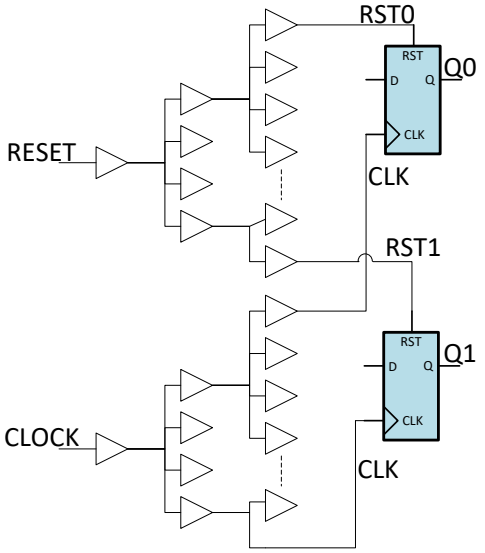
Complexity
Key issues

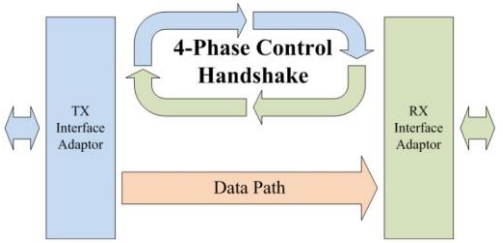
- Large designs
  - Long runtimes
- Many clocks
  - Multiple clock relations, large results set
- Many CDCs
  - A need to design multiple synchronizers
  - Multiple Quasi-Static CDCs
- Many third-party IPs
  - False alarms, undiscovered CDC issues
- Many operation modes
  - Long runtimes, enormously large results set

Complexity
Solutions

- Large designs
  - Parallel exploration, hierarchical analysis
- Many clocks
  - Automatic-clock recognition, SDC analysis
- Many CDCs
  - Pre-verified synchronizers (incl. Quasi-Static)
  - Verification support (m/s modeling + coverage)
- Many third-party IPs
  - Modeling and auto-recognition of the IPs*
- Many operation modes
  - Shared user data-base for the multiple modes







*REF: U.S. Patent 8661383, 2014

Multiple Clock Design
Challenges

- Setup: Clock & Reset requirements...
- Complexity: design and verification
- **Integration: (Black-box) Third-party IPs**

VSYNC circuits

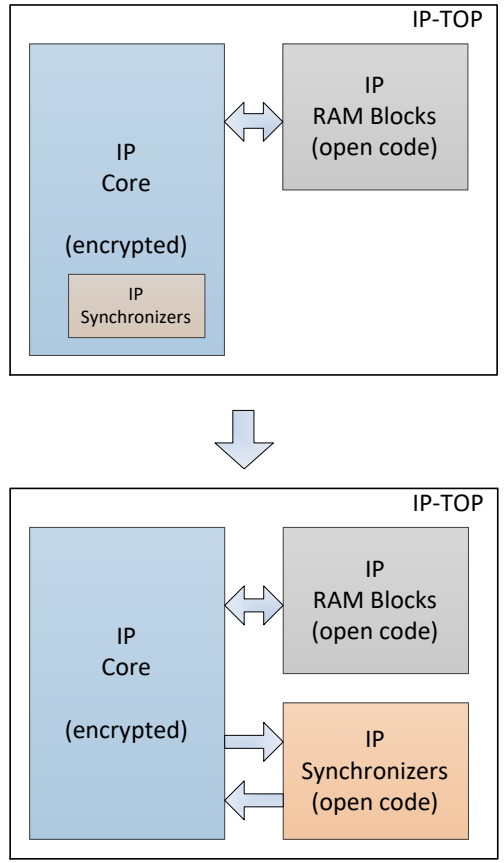
(Black-box) Third-party IPs
Key issues

- Third-party IP modules
  - Open-source
  - Encrypted
- May have internal synchronization schemes
  - May have internal synchronization bugs…
  - May cause CDC bugs, when incorrectly connected
    - Single / Multi-instance connections

Third-party IP integration

- Open-code IPs:
  - Hard to dig in someone else's code
  - Thus, it is usually worthwhile to abstract out up to Gray/Silver BB*
  - CDC report shall be jointly reviewed with the vendor
    - …

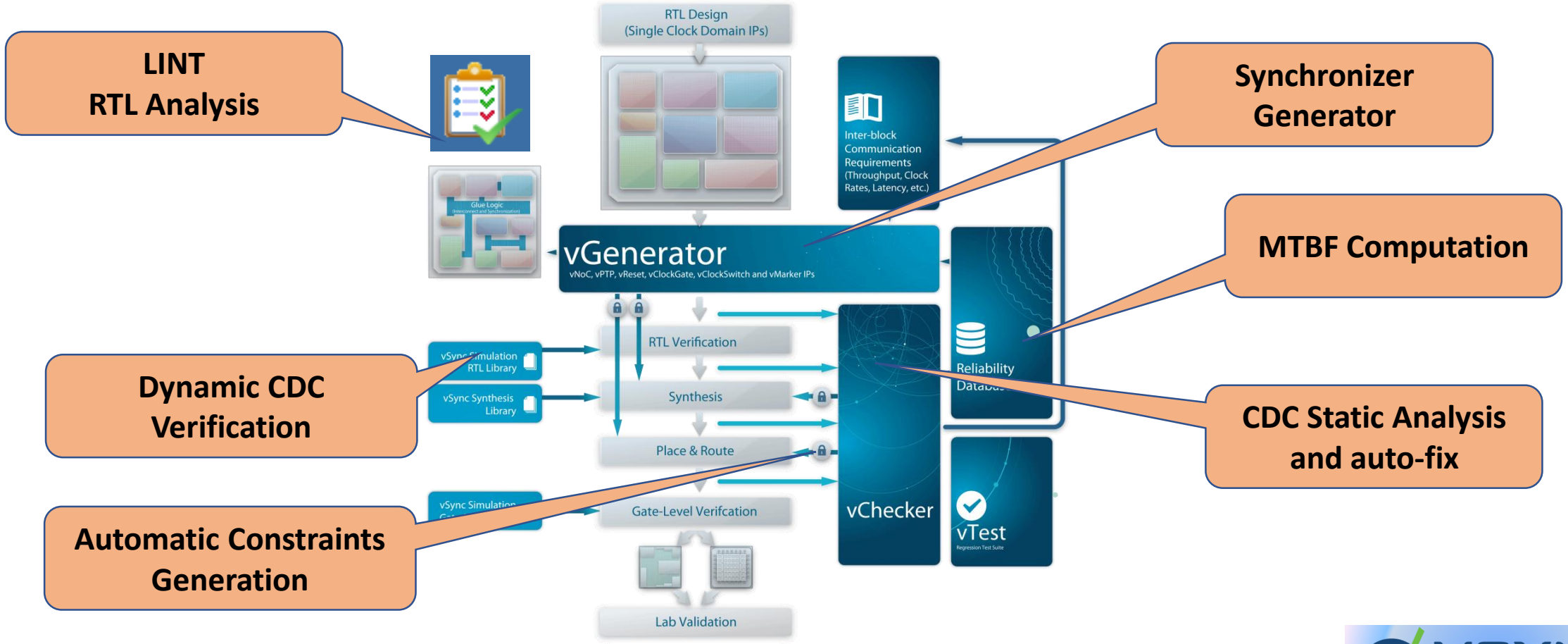
- Vendor IPs with internal synchronizers:
  - Ask IP vendor to extract all CDCs to a separate, open code hierarchy
  - Analyze/review this open-code with CDC verification tools
    - Static & Dynamic
  - Communicate with IP vendor on possible issues and waivers



*REF: U.S. Patent 8661383, 2014

VSYNC circuits

Dealing with BOTH design and verification *vSync Vincent CDC platform*



Thank you!

www.vsyncc.com

vSync Vincent Platform:
A Complete Solution for CDC