Abstract

CDC analysis has evolved as an inevitable stage in RTL quality signoff in the last two decades. Over this period, the designs have grown exponentially to SOC's having 2 trillion+ transistors and chiplet's having 7+ SOC's. Today CDC verification has become a multifaceted effort across the chips designed for clients, servers, mobile, automotives, memory, AI/ML, FPGA etc. with focus on cleaning up of thousands of clocks and constraints, integrating the SVA's for constraints in validation environment to check for correctness, looking for power domain and DFT logic induced crossings, finally signing off with netlist CDC to unearth any glitches and missing crossings during synthesis. As the design sizes increased in every generation the EDA tools could not handle running flat and the only way of handling design complexity was through **hierarchical CDC analysis** consuming abstracts. Also, hierarchical analysis helps to enable the analysis in parallel with teams across the globe. Even with all these significant progress in capabilities of EDA tools the major bottleneck in CDC analysis of complex SOC's and Chiplets is consuming abstracts generated by different vendor tools. **Different vendor tool abstracts** are seen because of multiple IP vendors , even in house teams might deliver abstracts generated with different vendors tools.

The **Accellera CDC Working-Group** aims to define a **standard** CDC IP-XACT model to be portable and reusable regardless of the involved verification tool.

As moving from monolithic designs to IP/SOC with IPs sourced from a small/select providers to sourcing IPs globally (to create differentiated products), the quality must be maintained as driving faster time-to-market. In areas where the standards (SystemVerilog, OVM/UVM, LP/UPF) are present, the integration is able to meet the above (quality, speed). However, in areas where standards (in this case, CDC) are not available, most options trade-off either quality, or time-to-market, or both :-(Creating a standard for inter-operable collateral addresses this gap.

This **tutorial** aims to remind the definitions of **CDC-RDC Basic Concepts and constraints**, as well as the description of the **reference verification flow**, and addressing the **goals, scope & deliverables of the Accellera CDC Working Group** in order to elaborate a specification of the standard abstract model.

Presenters Bio

- **Joachim VOGES**
  - holds a diploma in electrical engineering from the University of Kaiserslautern (Germany).
  - After more than 25 years of experience in the semiconductor industry, he works as a principal engineer for the structural verification and RTL sign-off of automotive microcontrollers at Infineon Technologies AG in Munich.
  - He is a member of the Accellera working group "CDC", mainly contributing to the subgroup for output collaterals.

Presenters Bio

- **Jean-Christophe BRIGNONE**
  - JC works currently as a Senior Member of the Technical Staff for the CDC-RDC Verification of RF & MPU SoC at STMicroelectronics in Grenoble (France), leading the Company level CDC-RDC Verification Methodology Working Group for tool-flow deployment
  - He is a member of the Accellera Working Group "CDC", leading the “Training” subgroup & representing his Company as voter
  - He has authored over 18 papers on static checks and Hierarchical flows, presented at industrial & scientific international conferences and won Best Paper awards
  - Reaching more than 26 years of experience in the semiconductor industry as digital designer, he specializes in static & formal verification for 12 years
  - He holds a Master degree in Microelectronics engineering from the University of Toulouse (France)

2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

MUNICH, GERMANY
OCTOBER 15-16, 2024

Breakthrough in CDC-RDC Verification Defining a Standard for Interoperable Abstract Model

Accellera CDC Working Group Authors

Jean-Christophe BRIGNONE
Ashish SONI
ST life.augmented

Joachim VOGES
Infineon
Jan HAYEK
BOSCH

Devender KHARI
AGNISYS
Jebin MOHANDAS
intel

Suman CHALANA
Q

Manish BHATI
Farhad AHMED
SIEMENS

Chetan CHOPPALI
SUDARSHAN
MARVELL

Bill GASCOYNE
Blue Pearl Software

Kranthi PAMARTHI
RENESAS
accellera SYSTEMS INITIATIVE

Bridging the Gap: Standardizing CDC and RDC Closure with Interoperable Abstract Models

Accellera CDC Working Group

Authors

Devender KHARI

Ashish SONI

Jean-Christophe BRIGNONE

Jebin MOHANDAS

Suman CHALANA

Manish BHATI

Farhad AHMED

Joachim VOGES

Jan HAYEK

Chetan CHOPPALI SUDARSHAN

Bill GASCOYNE

Kranthi PAMARTHI

2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION JAPAN

Standardizing CDC and RDC abstract models

Accellera CDC Working Group

Anupam BAKSHI

Ping YEUNG

Chetan CHOPPALI SUDARSHAN

Farhad AHMED

Iredamola OLOPADE

Sean O'DONOHUE

Bill GASCOYNE

Kranthi PAMARTHI

2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

SAN JOSE, CA, USA
MARCH 4-7, 2024

Hierarchical CDC and RDC closure with standard abstract models

Accellera CDC Working Group

Ping YEUNG — NVIDIA
Chetan CHOPPALI SUDARSHAN — MARVELL
Farhad AHMED — SIEMENS
Iredamola OLOPADE — intel
Sean O'DONOHUE — SYNOPSYS
Bill GASCOYNE — Blue Pearl Software
Kranthi PAMARTHI — RENESAS
Anupam BAKSHI — AGNISYS, SYSTEM DEVELOPMENT WITH CERTAINTY

2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE 10 YEAR ANNIVERSARY

Making the impossible possible: CDC and RDC closure with abstracts from different tools

Accellera CDC Working Group

Diana KALEL
Jean-Christophe BRIGNONE
Farhad AHMED
Eldad FALIK
Jebin Mohandes
Jerome AVEZOU
Bill GASCOYNE
Kranthi PAMARTHI







Presenter Introduction

- Companies
  - Jan
  - Jean-Christophe
  - Joachim
- EDA Vendors
  - TBC

Agenda

	Topic	Slide update/create	Presenter	Time
#0	**Accellera updates**		TBC	5m
#1	**CDC-RDC**			55m
	CDC-RDC Basic Knowledge	Bill Gascoyne (Blue Pearl), Jan Hayek (Bosch)	Jan Hayek (Bosch)	10m
	Setup Constraints & Verification	Ping Yeung (Nvidia) Jebin Vijai (Intel)	Abdelouahab Ayari (Siemens)	10m
	Structural CDC/RDC	Chetan Choppali Sudarshan (Marvell) Suman Chalana (Qualcomm)	Abdelouahab Ayari (Siemens)	10m
	CDC Assertion-Based Verification	Kranthi Pamarthi (Renesas Electronics) Manish Bhati (Siemens EDA)	Joachim Voges (Infineon)	10m
	CDC-RDC Hierarchical Flow	Jean-Christophe Brignone, Ashish Soni (ST)	Jean-Christophe Brignone (ST)	10m
	Q & A			10m
#2	**Accellera CDC**			20m
	Standard	Iredamola Olopade, Jebin Vijai (Intel)	Jean-Christophe Brignone (ST)	3m
	Format	Devender Khari (Agnisys)	Joachim Voges (Infineon)	3m
	Output	Joachim Voges (Infineon), Devender Khari (Agnisys)	Joachim Voges (Infineon)	3m
	Assertion	Kranthi Pamarthi (Renesas Electronics), Manish Bhati (Siemens EDA)	Jan Hayek (Bosch)	3m
	Testing	Farhad Ahmed (Siemens EDA), Suman Chalana (Qualcomm)	Jan Hayek (Bosch)	3m
	Training	Jean-Christophe Brignone, Diana Kalel, Ashish Soni (ST)	Jean-Christophe Brignone (ST)	3m

1.1 CDC-RDC Basic Knowledge:

- Synchronous vs. asynchronous clocks
- Problems related to Clock Domain Crossing (CDC)
- CDC Synchronization
- Problems related to Reset Domain Crossing (RDC)
- RDC Synchronization

Synchronous vs. Asynchronous

- Synchronous clocks
  - Same source
  - Have an easily-established timing relationship
  - Static Timing Analysis works
- Asynchronous clocks
  - From different sources
  - Timing relationship unknown or difficult to establish
  - Static Timing Analysis doesn’t work
- Multi-clock designs, NOT clockless

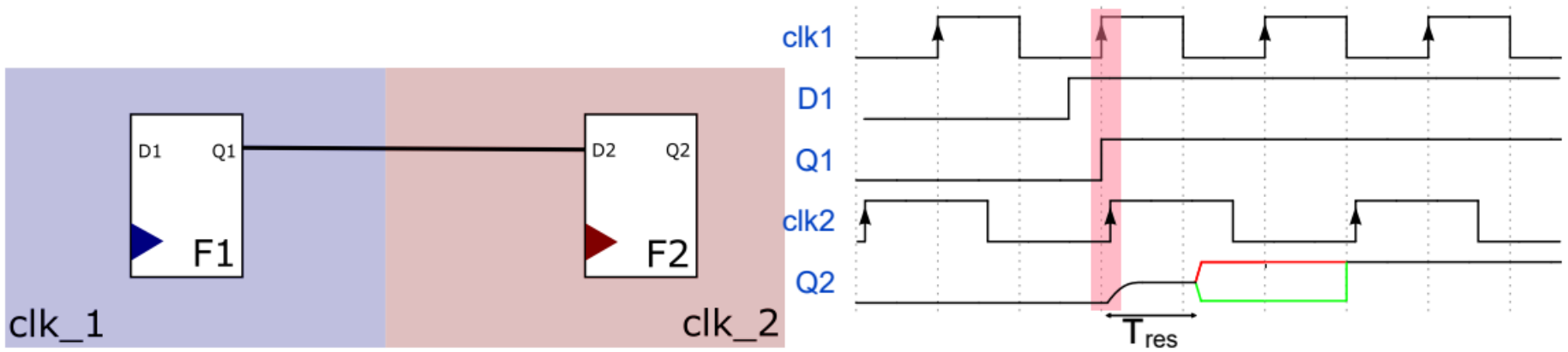
Coin Toss Analogy

- Think of a setup/hold violation result as the toss of a coin
  - Heads or Tails, but also very rarely it might just stay on its edge (metastability) before falling one way or the other
- Fixing metastability and fixing data coherency are independent
- For one bit, fixing metastability is enough
  - Coherency doesn't matter, since either heads or tails is fine
- For multiple bits, must fix metastability AND data coherency
  - Requires all heads or all tails from multiple coins
  - A losing bet!

Why do CDCs need fixing?

- Metastability
  - Timing violations on registers resulting in an indeterminate state lasting more than one clock cycle

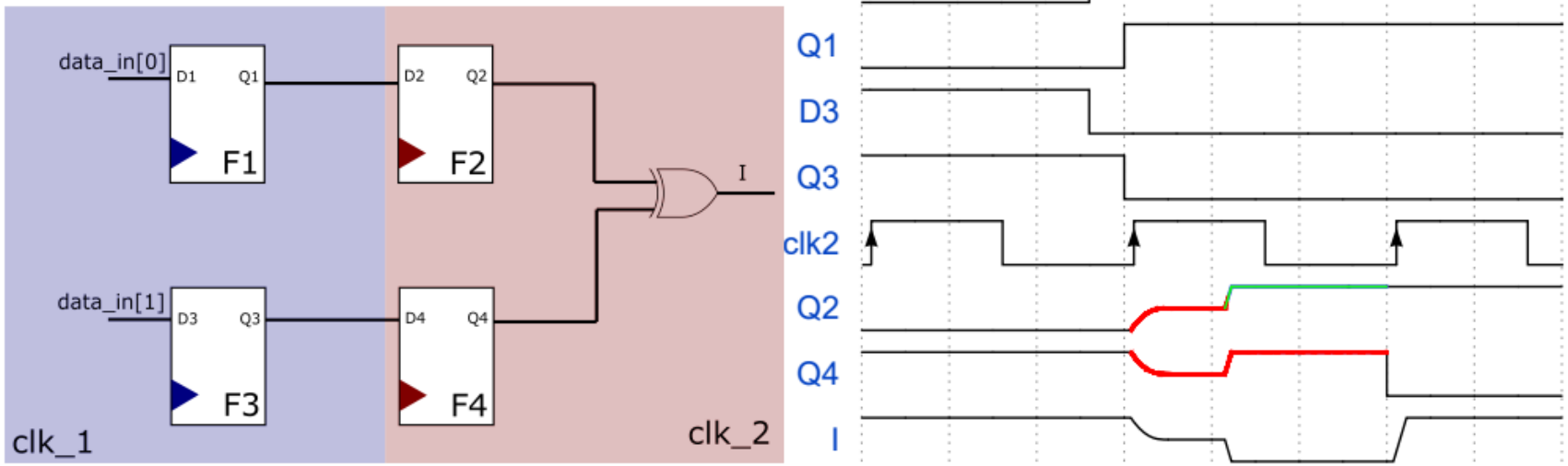
    The coin on its edge



(a) Structure *(b) Waveform*

Why do CDCs need fixing?

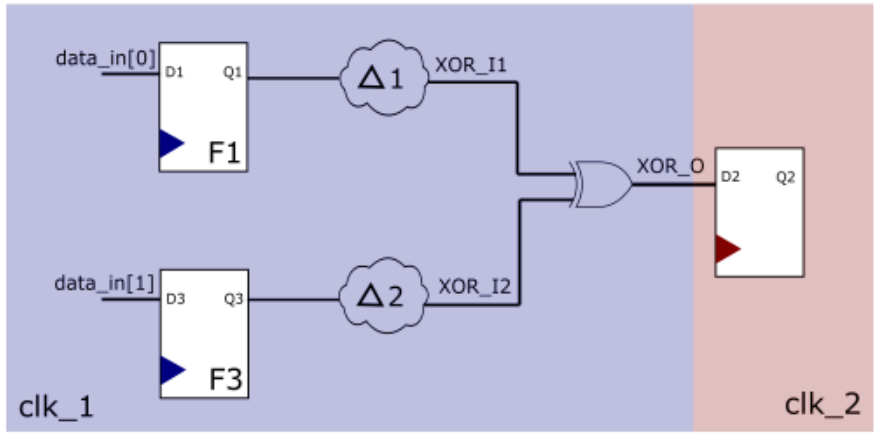
- Loss of Data Coherency
  - The indeterminate state settles to a random value



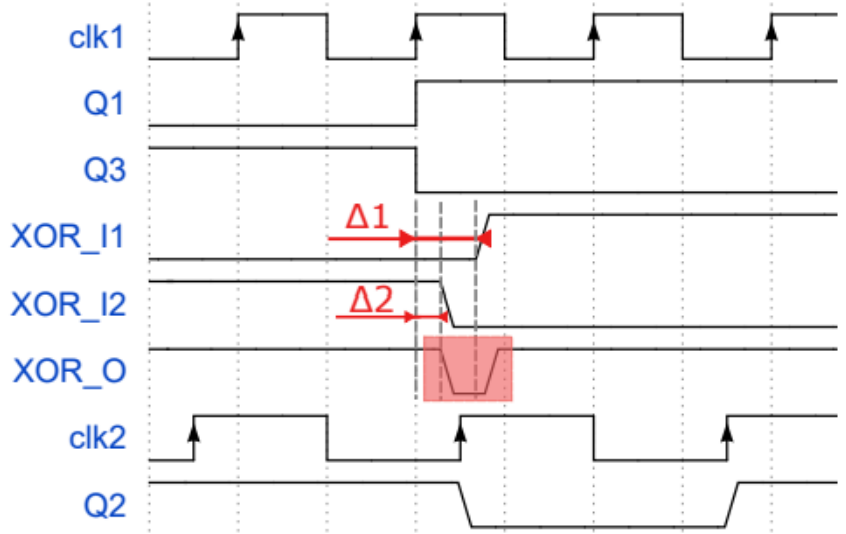
(a) Structure *(b) Waveform*

Why do CDCs need fixing?

- Glitches
  - Multiple synchronized paths reconverge to cause unexpected momentary transitions



(a) Structure



(b) Waveform

MTBF Mean Time Between Failures

- given metastability at t = 0,

  probability of metastability at t > 0 -> $e^{-t/\tau}$

- failure: still metastable at next clock edge

  - failure = p(enter m.s) x p(still m.s after T_S)

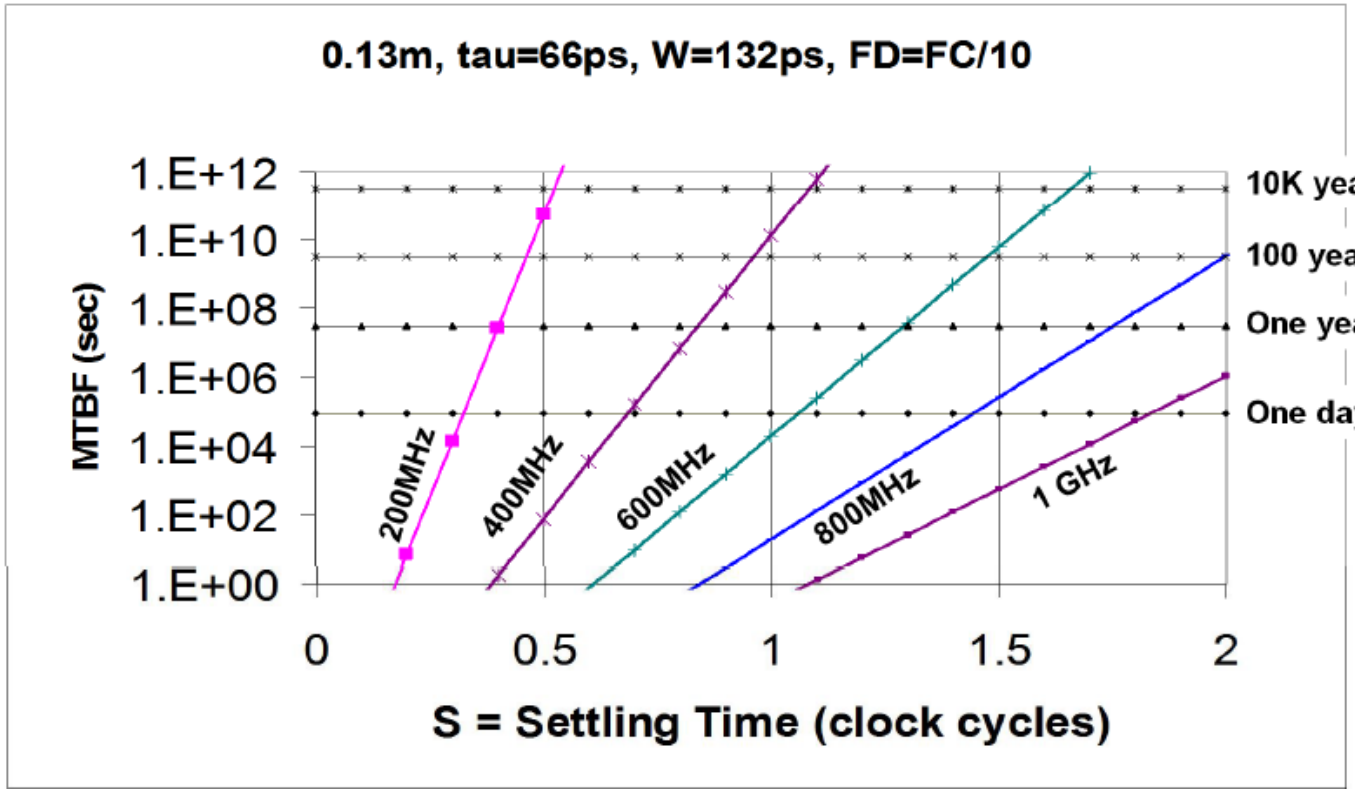
    = T_W/T_C x $e^{-Ts/\tau}$

  - rate(failure)

    = rate(enter m.s) x p(still m.s after T_S)

    = T_W x f_C x f_D x $e^{-Ts/\tau}$

- MTBF = 1 / rate(failure) $MTBF = \frac{e^{T_s/\tau}}{T_W f_C f_D}$



Synchronization

4.1 Multi-flop synchronizers

- Synchronizing one bit with two DFFs changes odds of metastability on the 2nd flop from ~1/p to ~1/p^2
  - The probability of a metastability event in a 2-flop metastability resolver

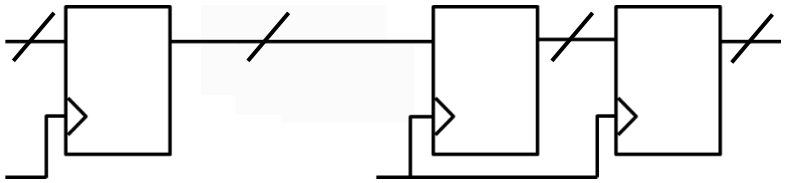
$$MTBF(t_r) = \frac{e^{\frac{t_r}{\tau}}}{T_0 * F_c * F_d} * \frac{e^{\frac{T_r}{\tau}}}{T_0 * F_c}$$

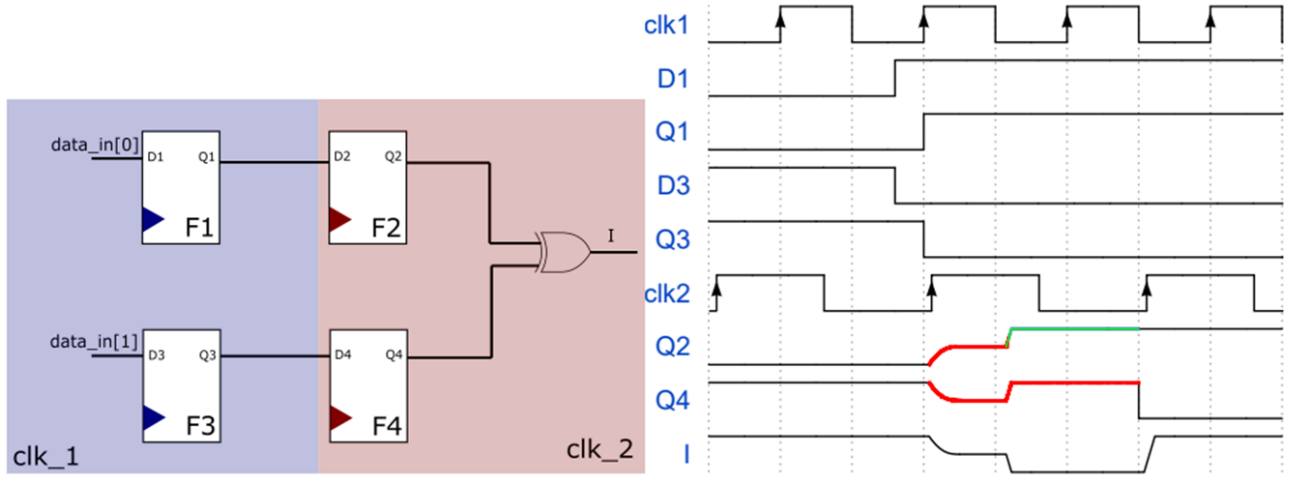
  - For a typical .25um ASIC technology, T_0=9.6nS,τ=0.31nS, and for T_r=2.3nS,F_c=100Mhz and F_d=1Mhz, the MTBF=20.1 days.
  - When using a 2-flop synchronizer, the MTBF at the output of the 2nd flop will be 9.57*10^10years.
  - Add a 3rd DFF for ~1/p^3
  - One bit matches cycle *n* or cycle *n+1* by coincidence

Synchronization

4.1 Multi-flop synchronizers

- With multiple bits, metastability is still addressed but data coherency is a problem!
  - If multiple bits change on the same cycle, the result of each bit is random
  - This synchronization works only if the data is “gray” (only one bit changes)



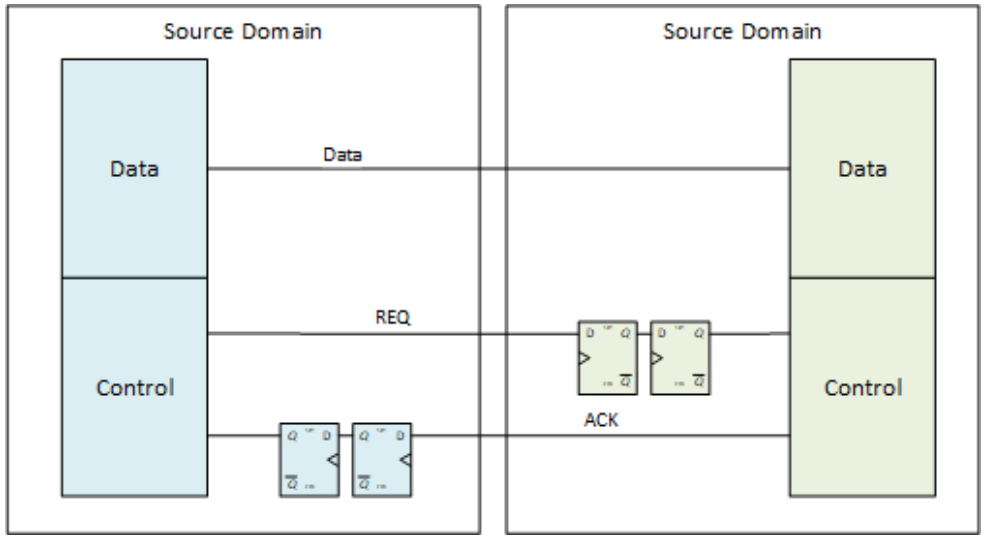


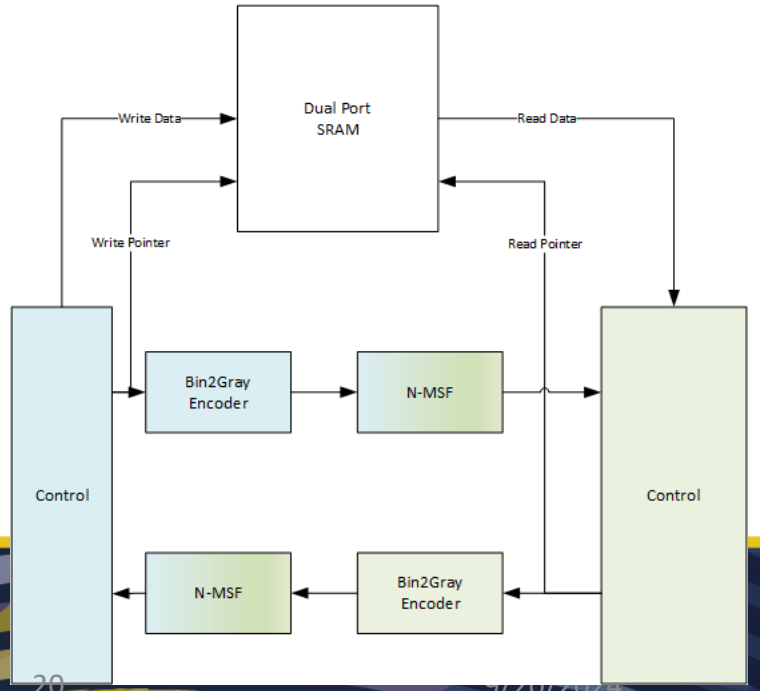
(a) Structure *(b) Waveform*

Synchronization

4.2 Protocol-based synchronizers

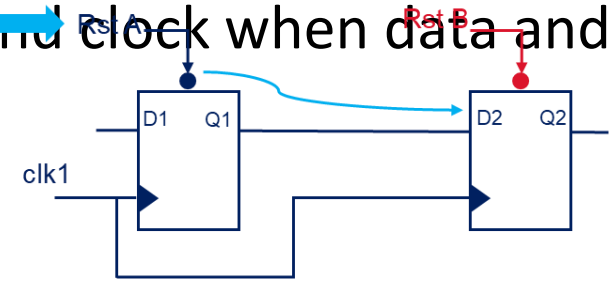
- Simple Qualifier
- Handshake protocol
- FIFO
  - Increased bandwidth
  - Throttling
  - Handles intermittent peaks of incoming data rate

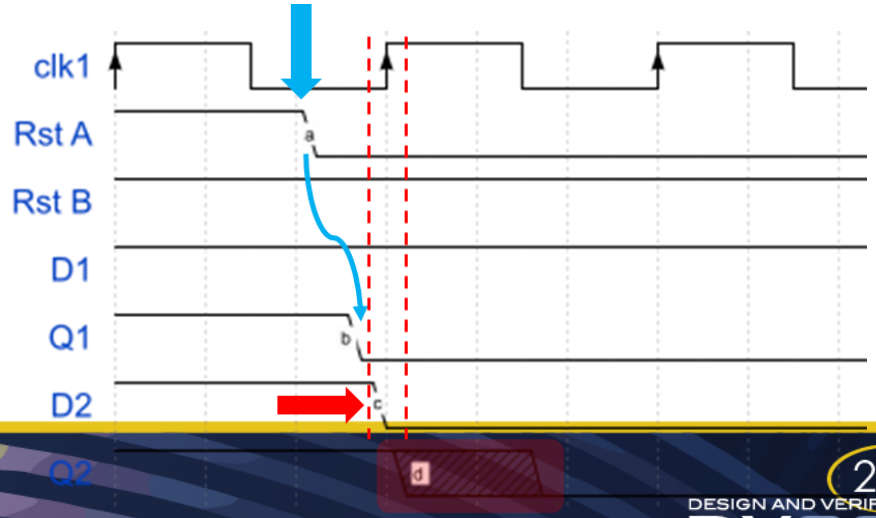




Asynchronous Reset Release

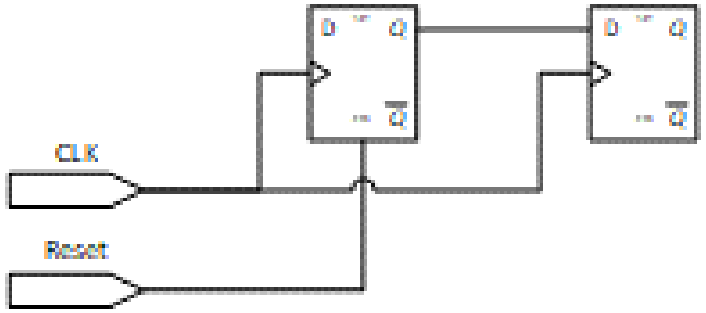
- In addition to setup and hold, DFF models also have **recovery time**
  - Time between asynchronous Set/Reset release and clock when data and output are different
  - Violating recovery time is no different than violating setup/hold
- Possible to synchronize asynchronous reset on release edge only
  - Static analysis is sufficient to make this determination

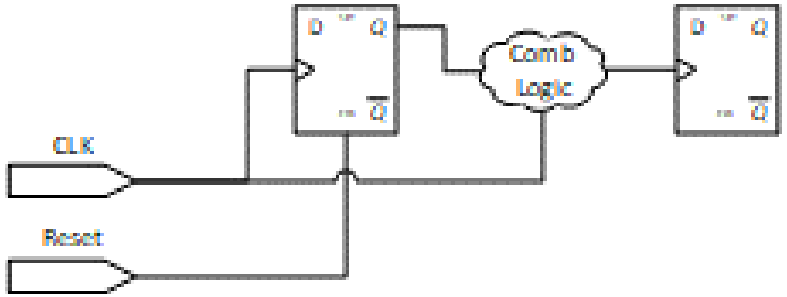




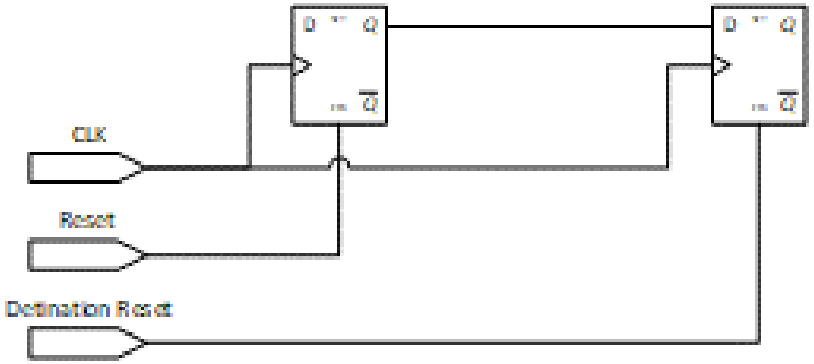
The Reset assertion RDC problem

- Paths passing from CLR to Q are usually not timing closed

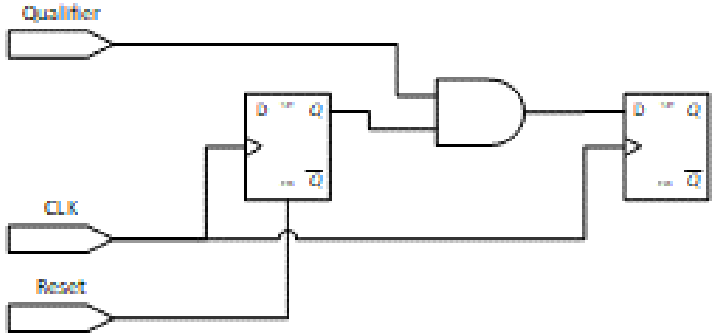




- Using reset ordering



- Using a CDC Control
  - CDC Control must be synchronous with target domain

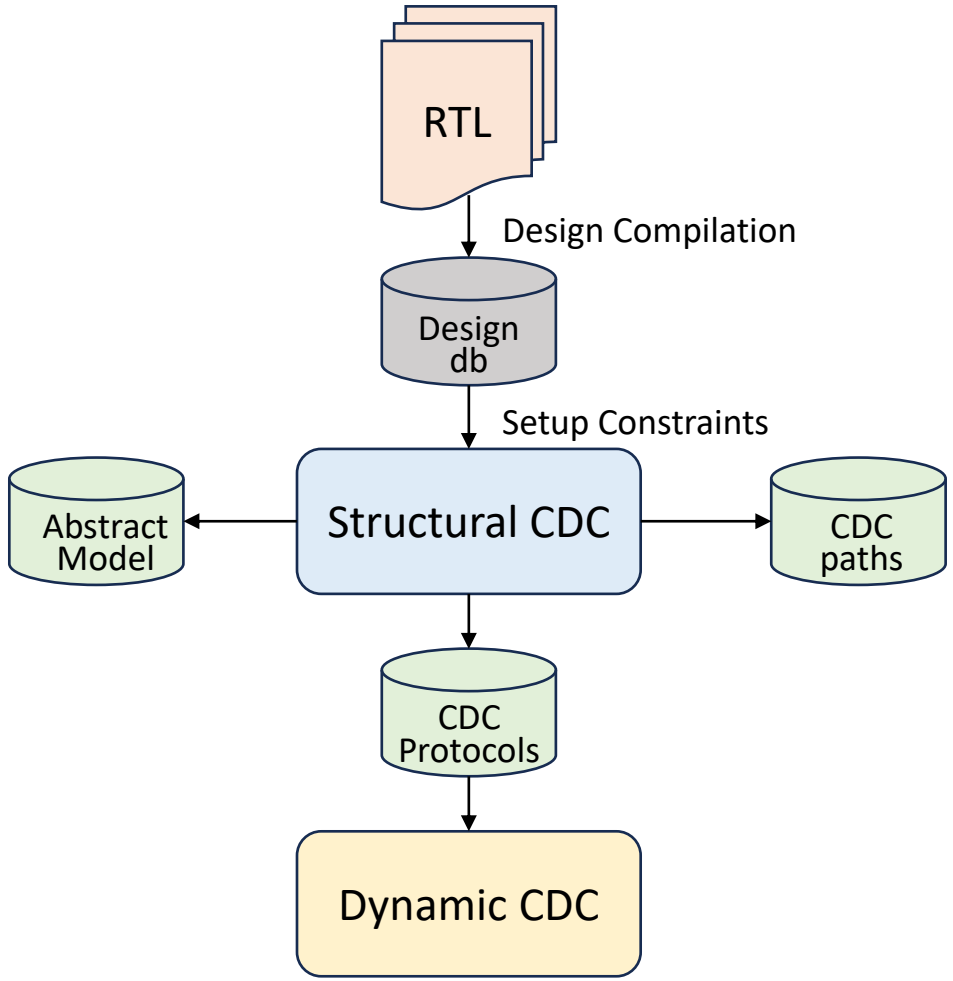


1.2 CDC Setup & Constraints

- CDC Verification flow
- Setup constraints
- Challenges

CDC Verification flow

- Design Compilation
  - Parameters, defines
  - SV packages, SV configuration, SV interfaces
- Setup Constraints
  - Clock, reset, and IO signals
  - Configuration: stable, constant inputs
- Structural CDC Check
  - CDC schemes validation and debugging
- Abstract Model Generation
- Dynamic/Formal CDC Verification
  - CDC constraints and protocols



Setup Constraints

- The set of constraints used to guide CDC verification

- Clocks
- Resets
- Configuration signals
- Black boxes
- Primary inputs/outputs

Don’t rely blindly on timing constraints

Reuse timing constraints is risky

- Pseudo-static signals
- Exclusive signals
- Gray coded buses
- Custom synchronizers
- False path

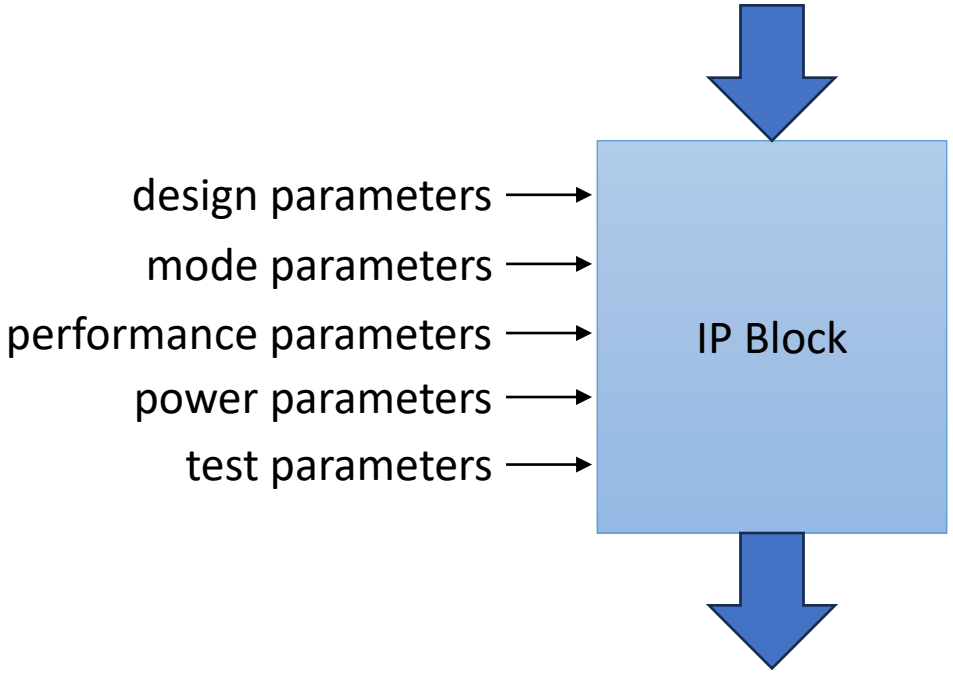
Clock groups for timing analysis =/= Clock groups for CDC analysis
Signal paths waived for time analysis =/= Signal paths waived for CDC analysis

Challenge #1: Design Parameters

- Blocks/IPs have many parameters
  - Some used for design, performance, optimization
  - Some used for integration, mode
  - Some used for DFT, DFP, DFM, etc

- Most parameters will affect CDC results
- Some parameters may not affect CDC results
  - Data_width, Addr_width
  - FIFO_depth, RAM_size

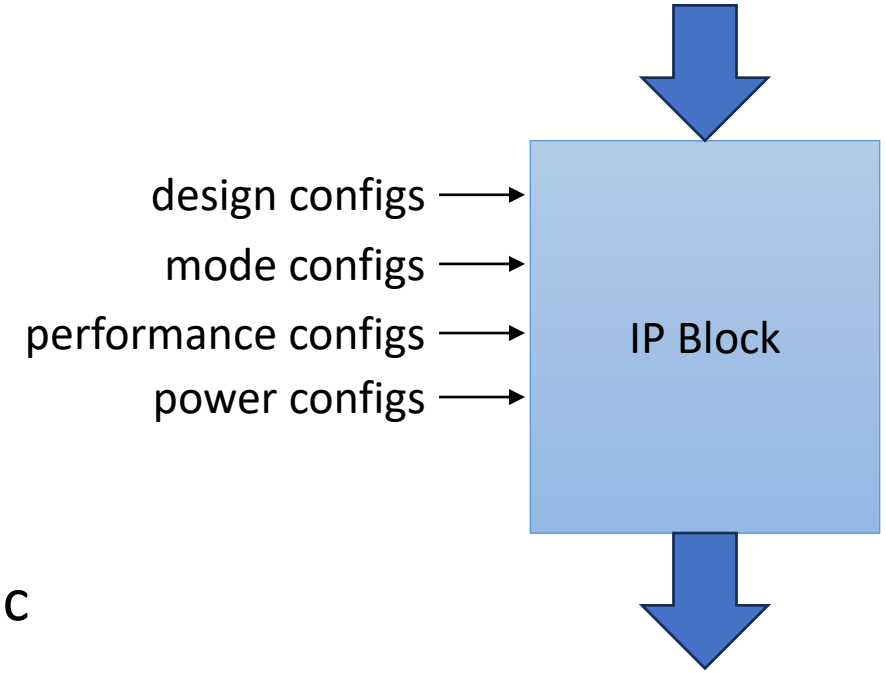
- The abstract model will become parameter-specific



To specify or not to specify

Challenge #2: Configuration Signals

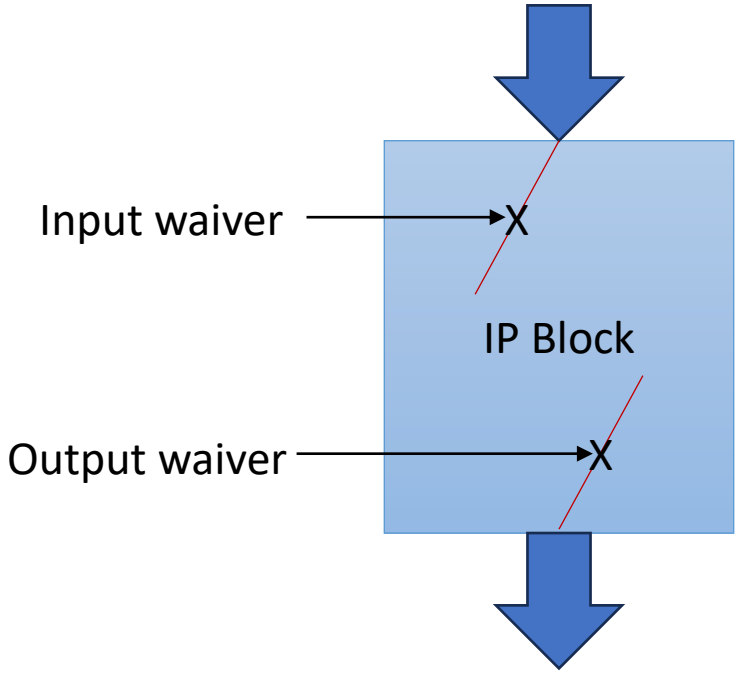
- Blocks/IPs have many configuration signals
  - Some used for design, performance, optimization
  - Some used for integration, mode
  - Some used for DFT, DFP, DFM, etc
- Most configuration signals will affect CDC results
  - Clock select, gating signals
- The abstract model will become configuration-specific



To constraint or not to constraint

Challenge #3: CDC Waivers

- Blocks/IPs have many CDC violations
  - Some are on the input signals
  - Some are on the output signals
- Some of the input violations can be waived
  - Pseudo-static input signals
  - Output signals
- Some of the input violations should not be waived
- The abstract model will become waiver-specific

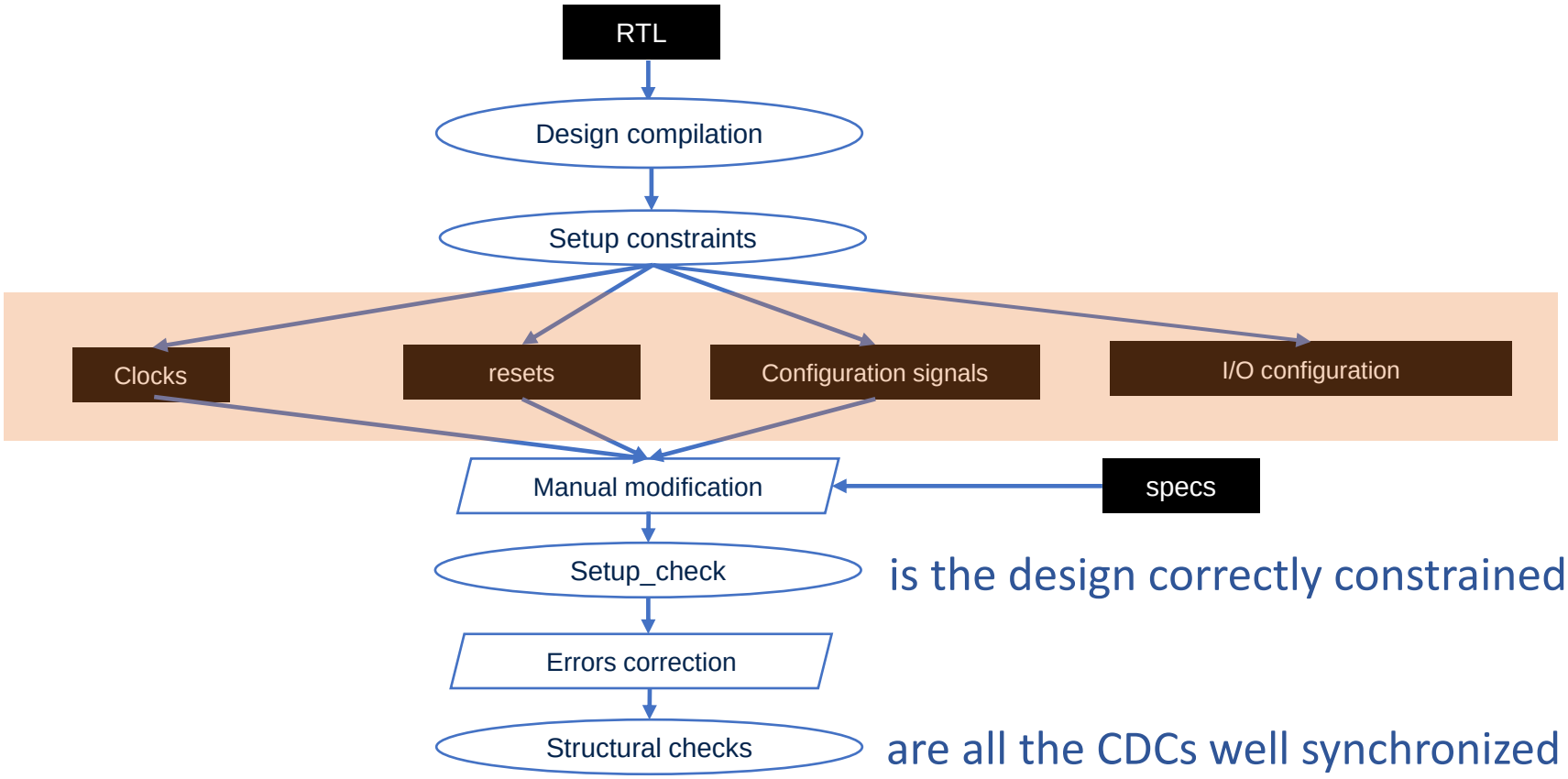


To waive or not to waive

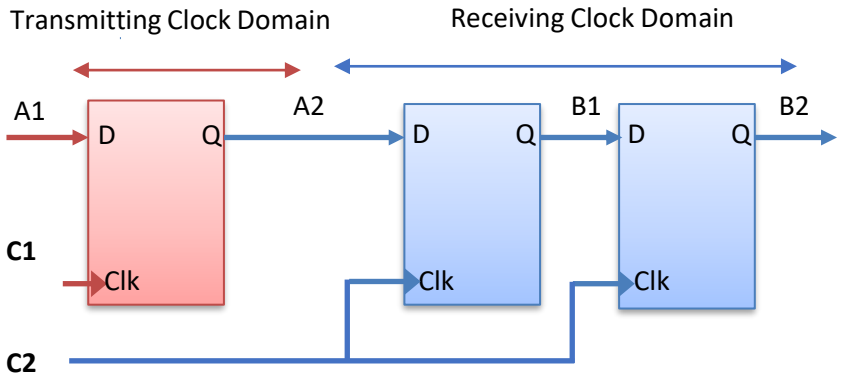
1.3 Structural CDC/RDC

- Structural CDC
- User defined synchronization modules
- CDC constraints
- Reset Domain Crossings

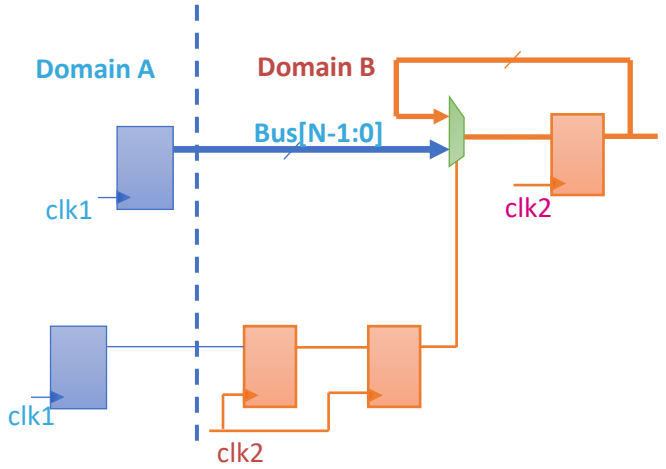
Structural CDC Analysis



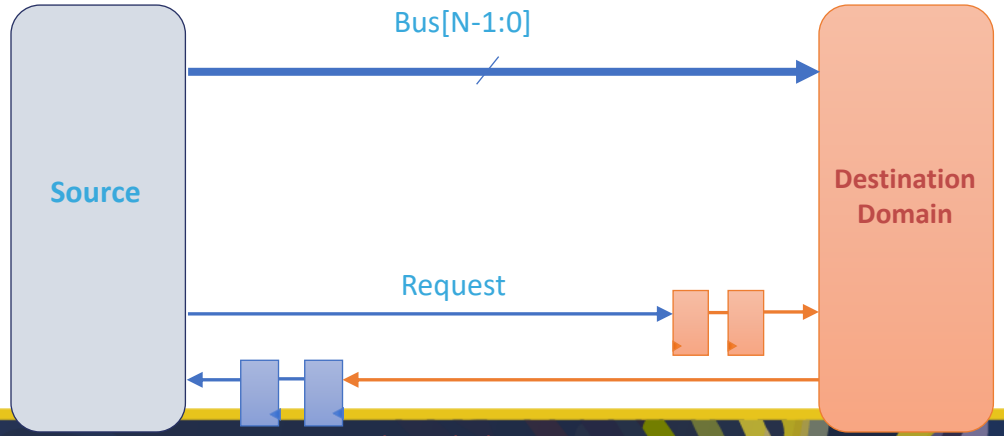
Structural CDC - Commonly Used Synchronization Schemes



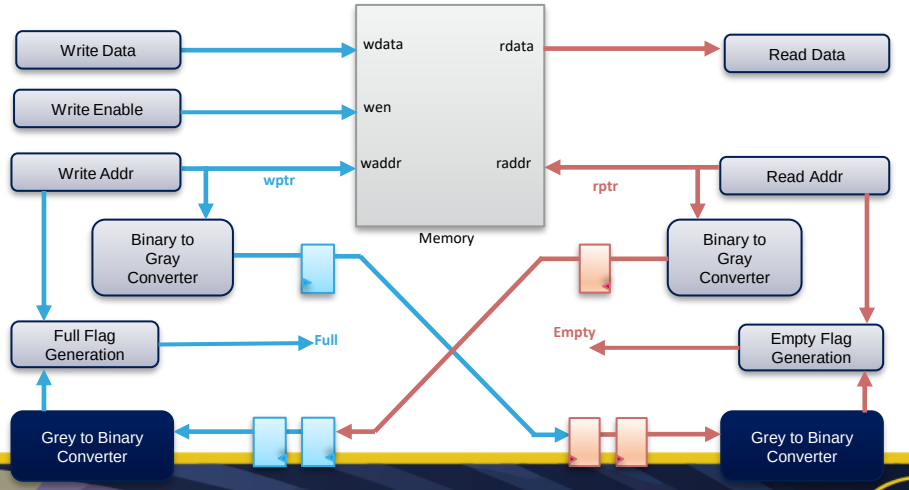
Double-FF synchronizer



MUX synchronizer



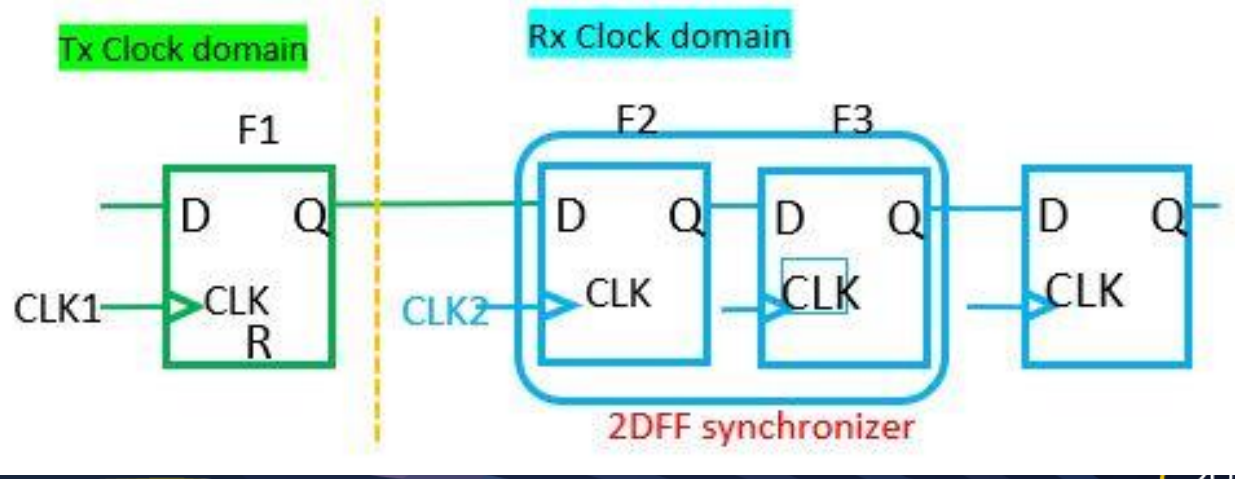
Handshake synchronizer



FIFO synchronizer

Structural CDC – User defined sync modules

- Double FF Synchronizer
  - Most design houses prefer to use their own CDC components
    - Disable automatic detection of the specific synchronizer type that you don't want the tool to recognize automatically
    - Declare your own scheme as user-defined synchronizer (before scheme detection)
  - Example: Use my own 2DFF only
    - Disable auto-detection of 2DFF
    - Declare your own module as 2DFF



Structural CDC

- Various Signal Configurations possible for structural CDC Analysis
  - Constant
  - Static
  - Mutually exclusive / Gray code
  - Externally synchronized
  - CDC False paths
    - *Not recommended (avoid using it to mask real CDCs)*
- Purpose
  - Define signal behavior that can help to reduce CDC analysis noise
    - Exclude certain paths which may not have any standard synchronizer but safe for CDC
    - Helps to speed up CDC analysis

Structural CDC

- CDC Constraints – Constant Declaration
  - It can be applied on a port or on an internal signal
  - A constant signal does not change in a given mode and hence does not cause a CDC issue

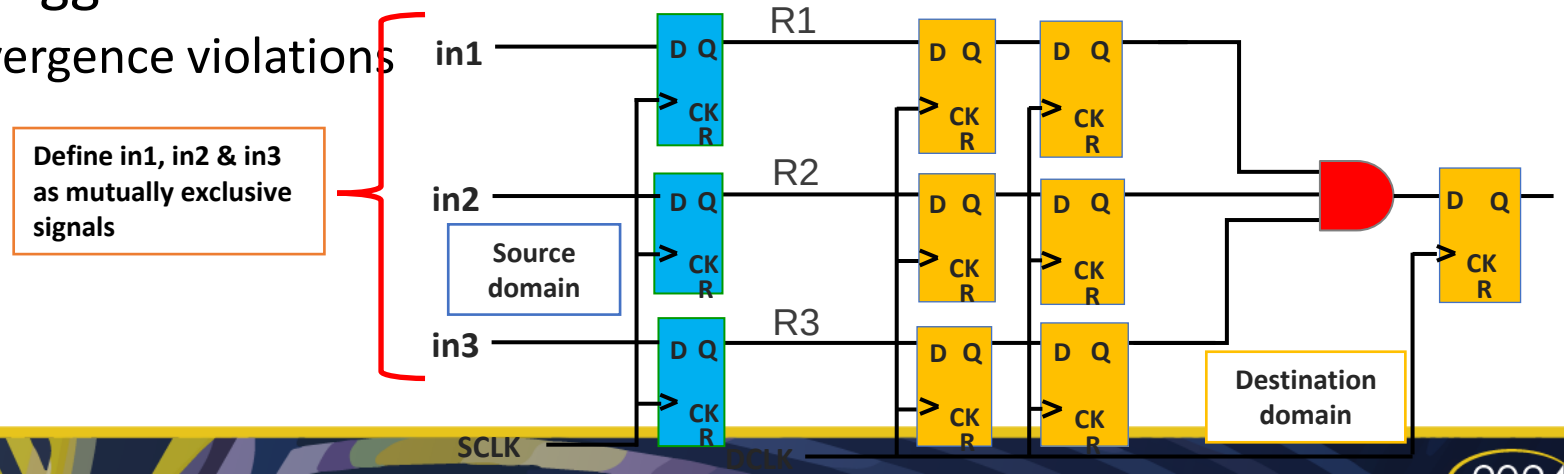
- Purpose
  - Define signal behavior that can help to reduce CDC analysis noise
    - Exclude certain paths which may not have any standard synchronizer but safe for CDC
    - Helps to speed up CDC analysis

Structural CDC

- CDC Constraints – Static Declaration
  - Any signal that does not change while the destination is active
  - Same as quasi-static or pseudo-static
  - A static signal does not cause CDC issues because
    - The receiver clock is not active
    - The receiver is under reset

Structural CDC

- CDC Constraints – Gray Coded Declaration
  - A bus can be specified as gray coded – Only one bit can toggle at a time
- CDC Constraints – Mutually Exclusive Toggle Declaration
  - A set of independent signals that can toggle only one at a time can be defined as mutually exclusive toggle
    - Helps in avoiding convergence violations

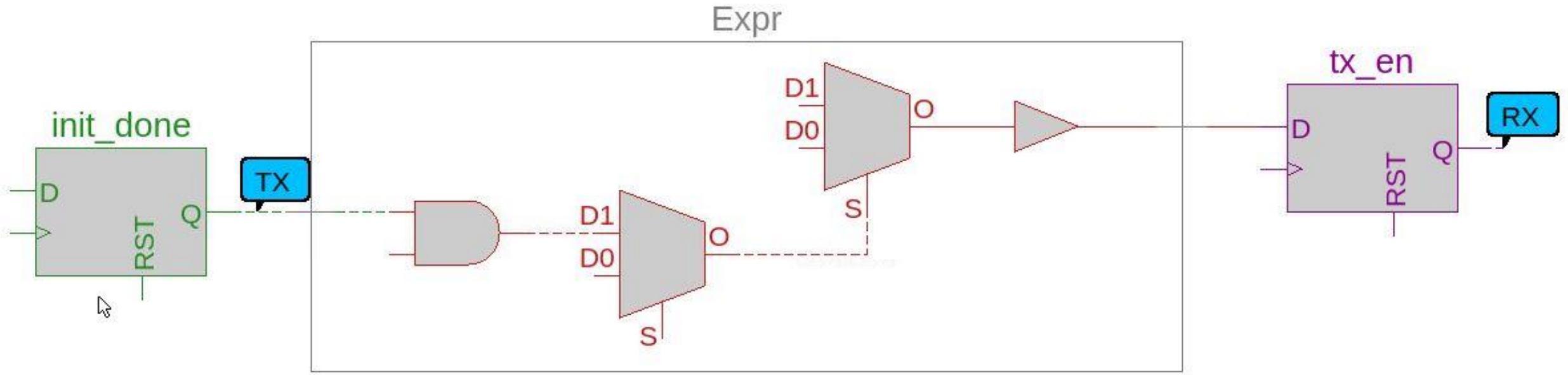


Structural CDC

- CDC Constraints - Externally Synchronized
  - A block level input/output port can be declared as *externally synchronized*
    - Represents the output of a control synchronizer (2DFF/Edge/Pulse)
    - Can be used as the control path for complex synchronizers (MUX Synchronizer, Glitch Protector)
    - Helps in auto-detection of the above composite synchronization scheme types
- CDC Constraints - CDC False Path Declaration
  - CDC Checks can be disabled on certain paths by user-defined constraints
  - User can set a constraint to let the tool automatically identify a functionally false path and hence reports the path as a safe path

Structural CDC

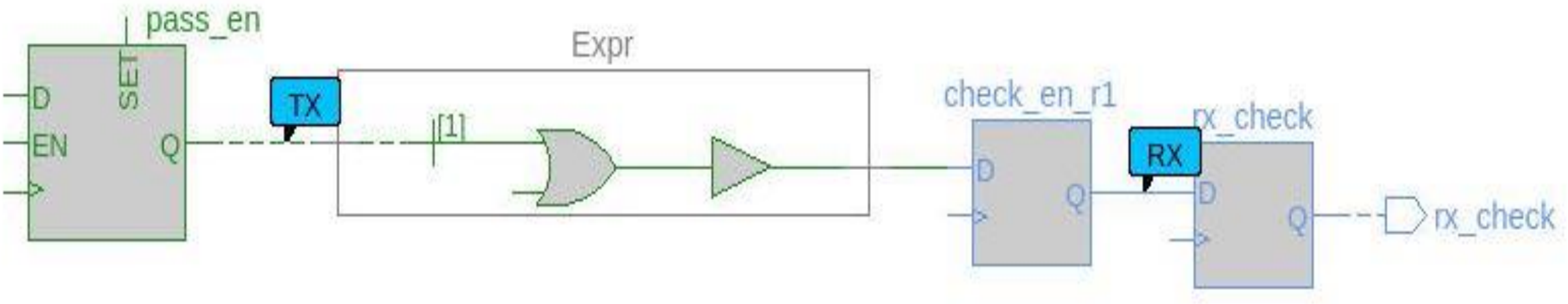
- Missing synchronizer on CDC path




```
Status   : Uninspected
Severity : Violation
Check    : Single-bit signal does not have proper synchronizer
```

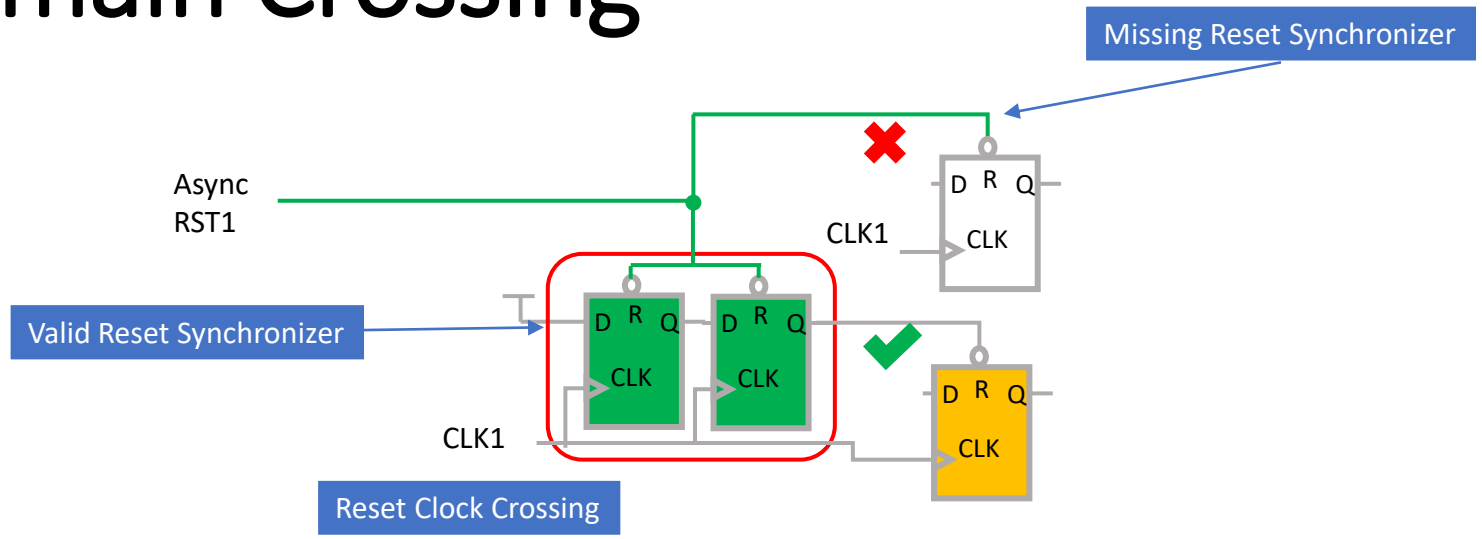
Structural CDC

- Combo-logic before synchronizer on CDC path




```
Status          : Uninspected
Severity        : Violation
Check           : Combinational logic before synchronizer
```

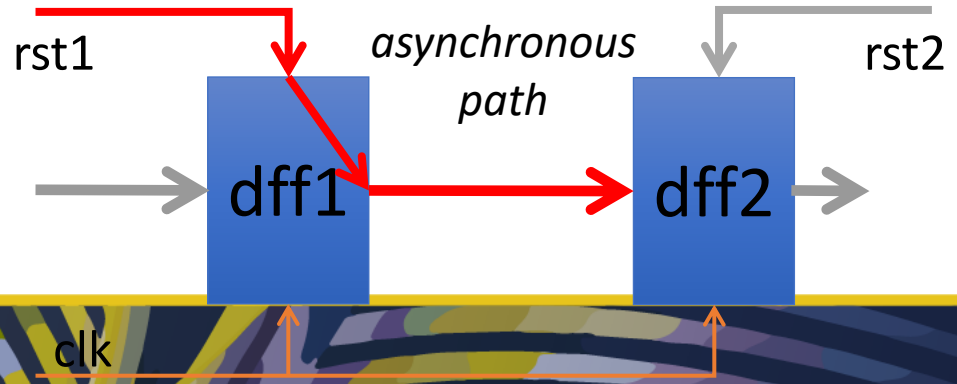
Reset-Domain Crossing



- Reset signal crossing from one clock domain to another
  - The asynchronous de-assertion of the reset signal at the destination flop can cause the signal to become metastable
  - Reset signals are required to be synchronized to destination domains for synchronous de-assertions

Reset-Domain Crossing

- Asynchronous reset domains causes meta-stability
  - Contain registers whose resets are asserted asynchronously
  - Originate in one asynchronous reset domain
  - Sampled by register(s) in a different reset domain
  - Reset ordering of different resets in the design



1.4 CDC Assertions

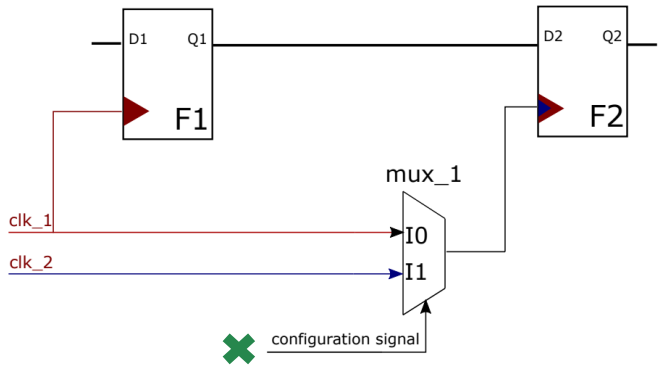
- Assertion Based Verification
- Overcoming Limitations

Structural CDC/RDC - Limitations

1- Constraints based static checks

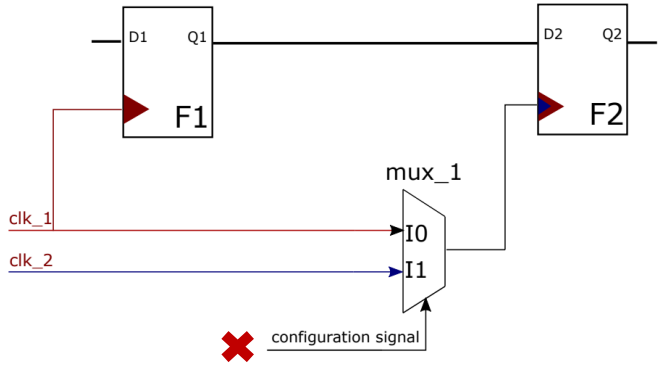
- Affect the results of the structural checks
- Are taken blindly for the structural verification
  - e.g., a CDC can be bypassed if the crossing signal is pseudo-static

✖ set_case_analysis 0 configuration_signal



Synchronous

✖ set_case_analysis 1 configuration_signal



Asynchronous

2- Rules based static checks

- Not possible to have rules for all architectures
- False positives / negatives
- Cannot verify correctness of design

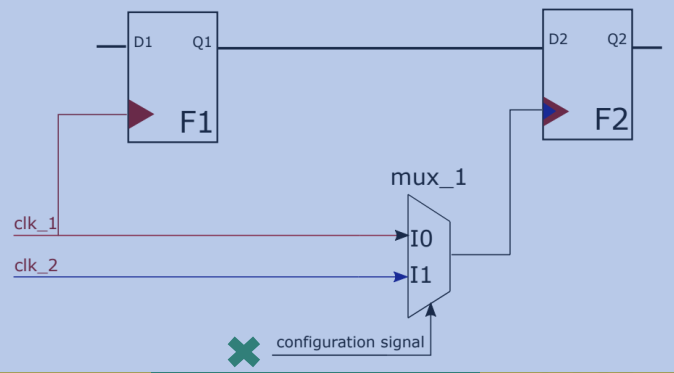
Overcoming Limitations-Assertions based Verification

1- Constraints based static checks

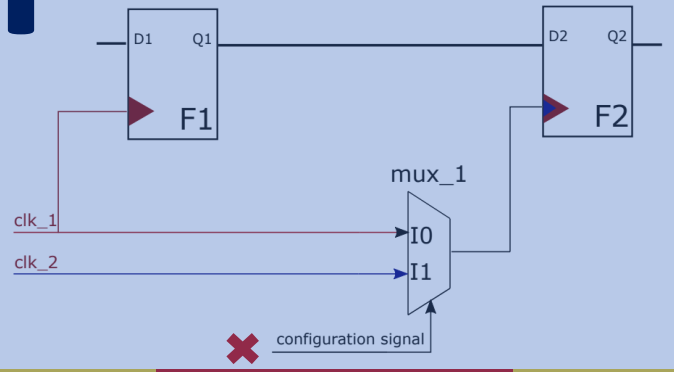
- Affect the results of the structural checks
- Are taken blindly for the structural verification
  - e.g., a CDC can be bypassed if the crossing signal is pseudo-static

Assumptions

set_case_analysis 0 configuration_signal set_case_analysis 1 configuration_signal



Synchronous



Asynchronous

2- Rules based static checks

- Not possible to have rules for all architectures
- False positives / negatives
- Cannot verify correctness of design

Assertions

Overcoming Limitations-Assertions based Verification

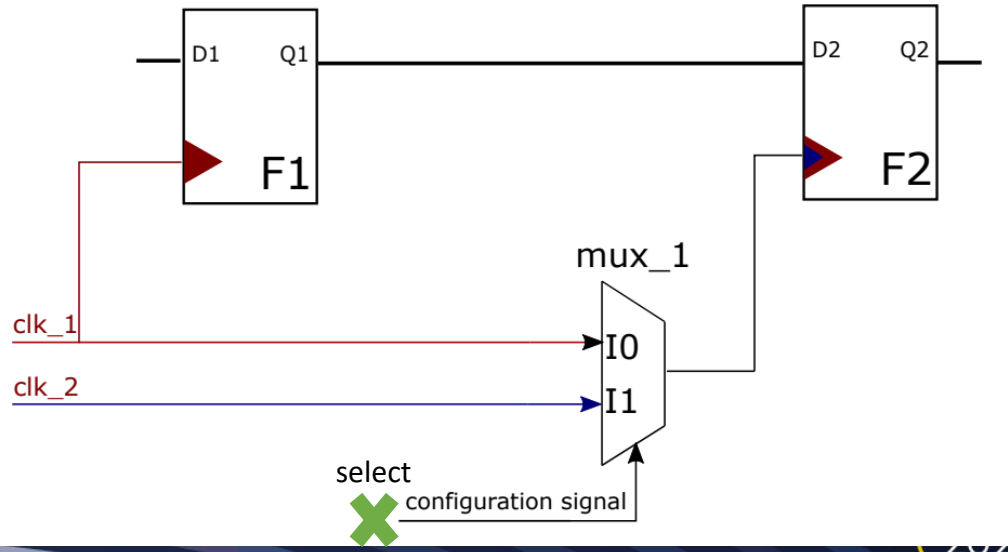
1- Constraints based static checks

- Constraints to be double checked with the functional verification

- Configuration signals

define_constant –value [0/1] –signal [signal name]

```
always@*
  begin
    assert_cdc_constant_prop : assert (select === value)
```



Overcoming Limitations-Assertions based Verification

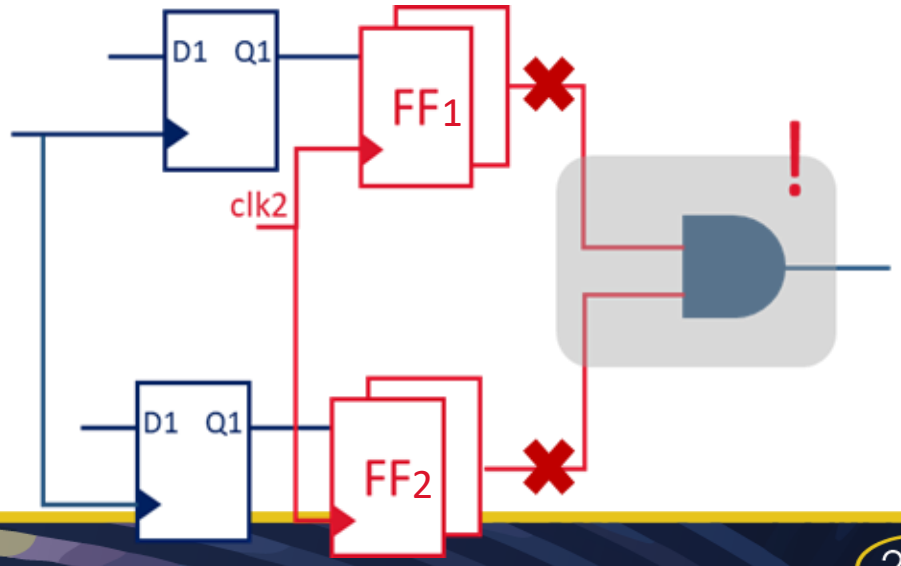
1- Constraints based static checks

- Constraints to be double checked with the functional verification

- Mutually exclusive

define_exclusive –signals [set of signals names]

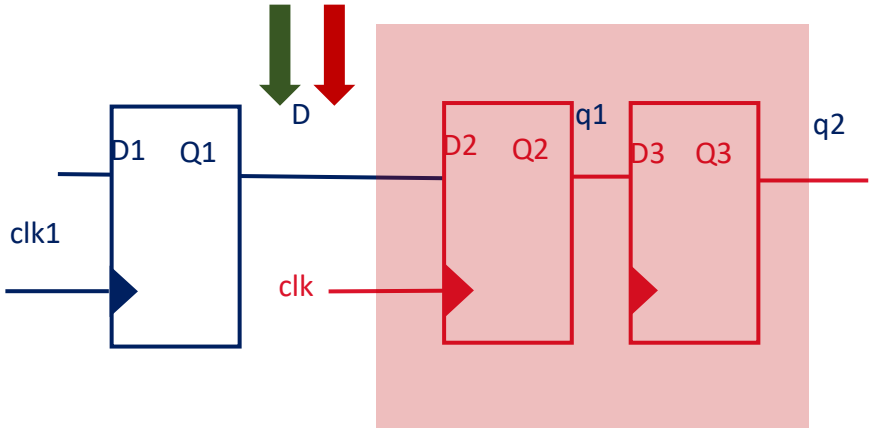
```
property mutex (data, clk);
   @(posedge clk)
    $onehot0(data ^ $past(data));
 endproperty
```



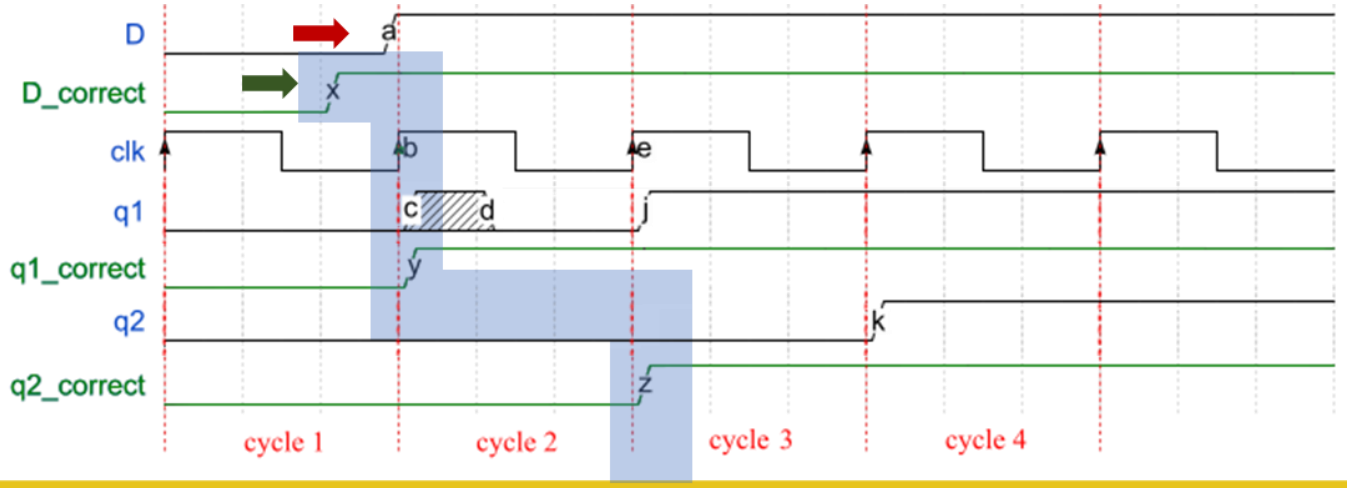
Overcoming Limitations-Assertions based Verification

2- Rules based static checks

- Fundamentally target to verify **design intent**
- CDC paths are **not covered by STA** :
  - Make sure source data is stable while crossing.




```
property cdc_data_stable (D, NUM_CYCLES);
    @(posedge clock)
     ##1 $changed(D) |-> $stable(D)[*(NUM_CYCLES-1)];
endproperty
```

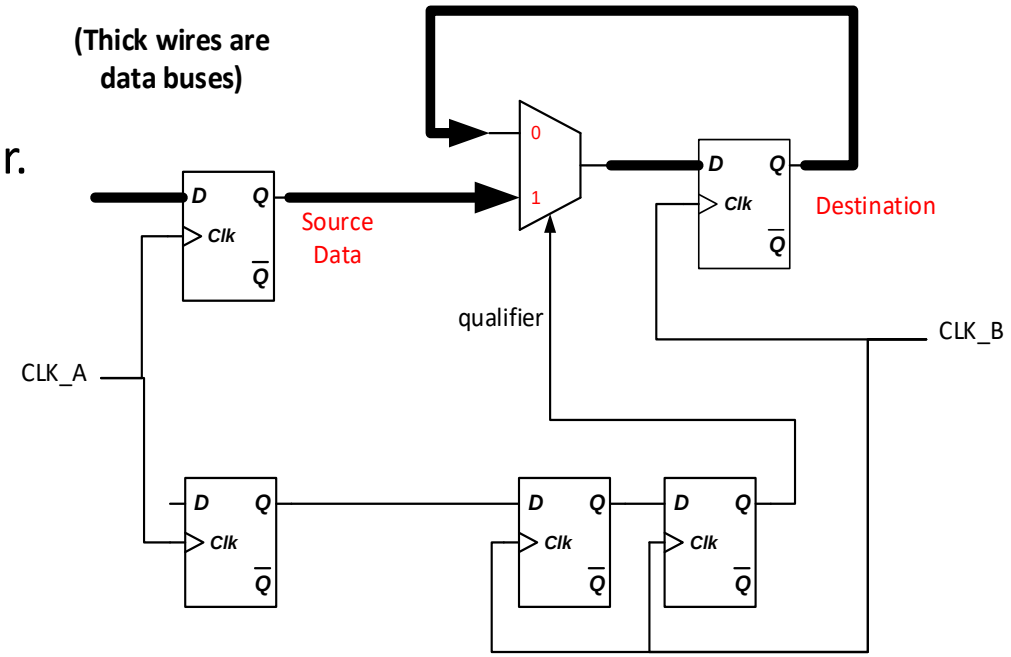


- NUM_CYCLES is based on synchronization latency

Overcoming Limitations-Assertions based Verification

2- Rules based static checks

- Fundamentally target to verify **design intent**
- CDC paths are **not covered by STA** :
  - Make sure source data is stable for several cycles.
  - Enabler : Make sure source data is stable wrt to its enabler.

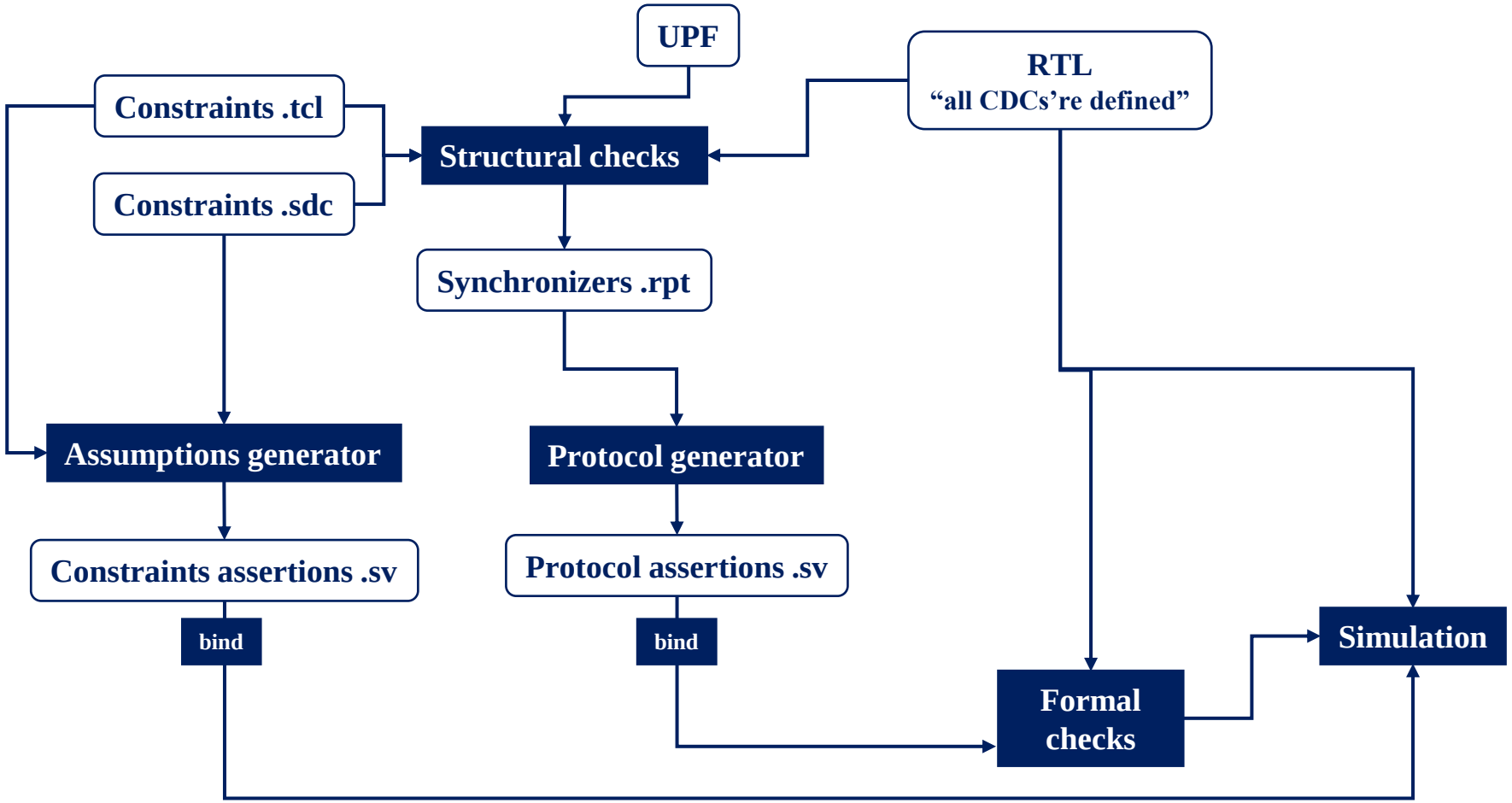


Source data must be static when CDC Control is enabling.
Source data can toggle when CDC Control is disabling.
CDC Control must be a known value

```
property qual_data_stable (SRC_DATA,QUAL,SETUP_ROOM);
   ##1 $changed(SRC_DATA) |-> (QUAL === 1'b0)[*SETUP_ROOM];
endproperty
```

- SETUP_ROOM = Synchronization Latency + Implementation Headroom

ABV - Assertions based Verification



Dynamic CDC Verification

- Dynamic verification is to ensure
  - structural CDC check is done with the proper constraints and assumptions
  - the identified CDC paths follow the protcols defined by the CDC schemes

- CDC constraint properties
  - Assertions are generated based on the setup constraints
  - Ideally, should be done concurrently with structural CDC check
  - Violations can potentially invalidate the complete structural CDC
- CDC protocol properties
  - Assertions are generated based on the CDC paths
  - Violations can potentially invalidate the CDC paths

1.5 Hierarchical CDC/RDC

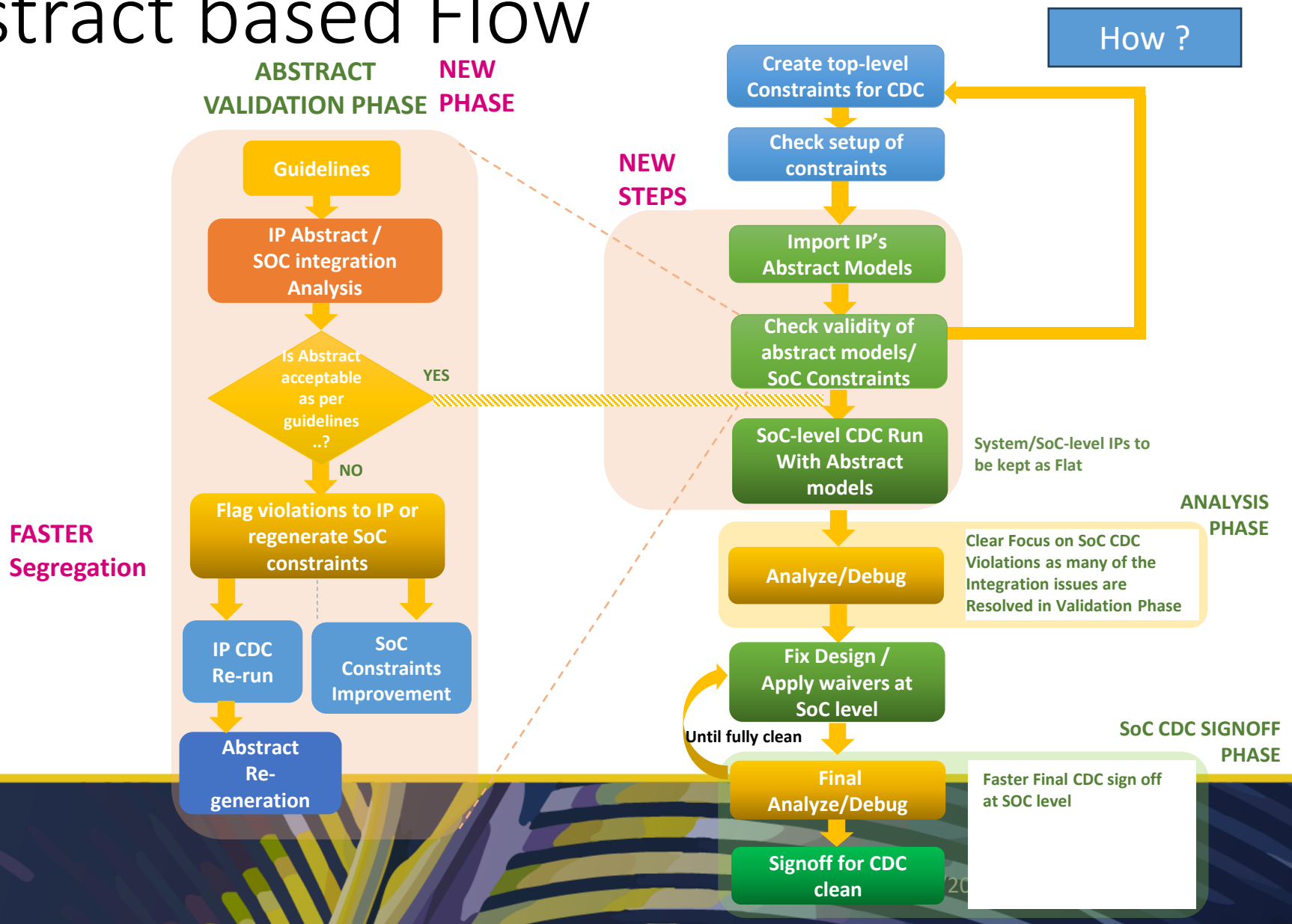
- Hierarchical Flow

CDC-RDC Hierarchical Flow

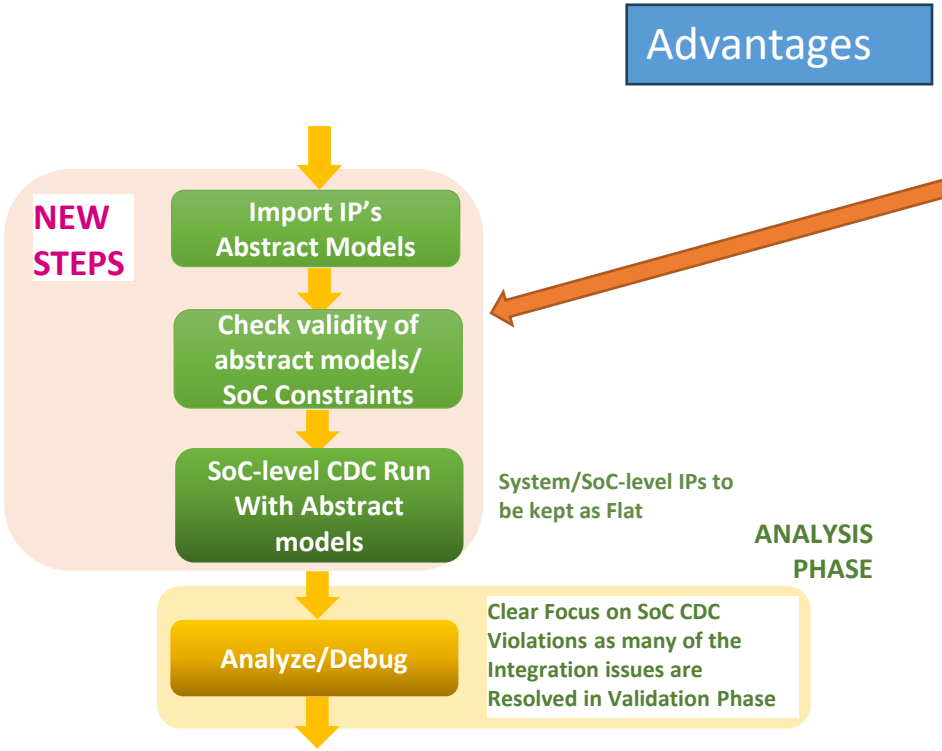
Why ?

- CDC-RDC Verification at SoC level
  - Flat RTL analysis only possible for small SoC and/or SoC simple clock-reset strategy
- Hierarchical strategy means first identify sub-blocks to be analyzed separately then modeled
  - Each CDC-RDC model being integrated at SoC level
- Allows parallelization of sub-blocks analysis and noiseless analysis at SoC level
- Challenges:
  - Dependency to sub-blocks provider to deliver CDC-RDC model
  - Compatibility of CDC-RDC models in case of multiple EDA tools usage

IP Abstract based Flow



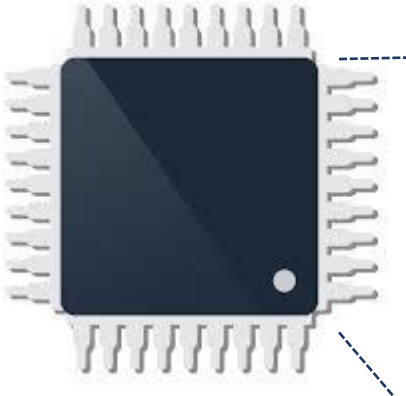
Usage of IP Abstracts



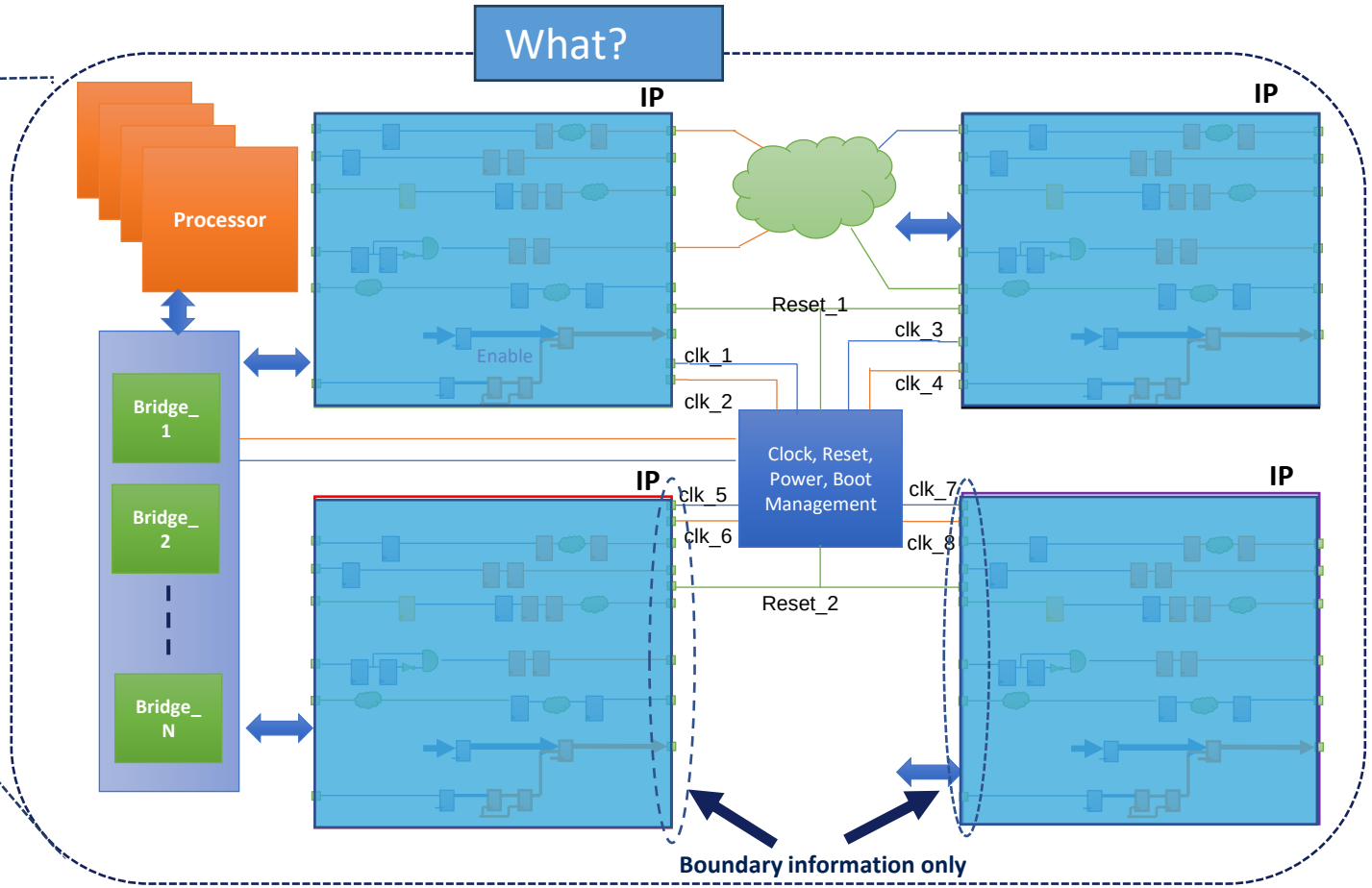
Advantages

- **Focus on Improving the Quality** of CDC checks
- **Re-utilization of IP CDC** clean up effort at SoC level by using **Abstracts**
- **Segregating IP CDC issues** from that of **SoC**, thereby **reducing** the violations to be analyzed at SoC level.
- **Re-aligns the Focus** of debug effort on the actual **SoC integration issues** and not on IP issues.
- **Reduces SoC runtime** which leads to increase in productivity

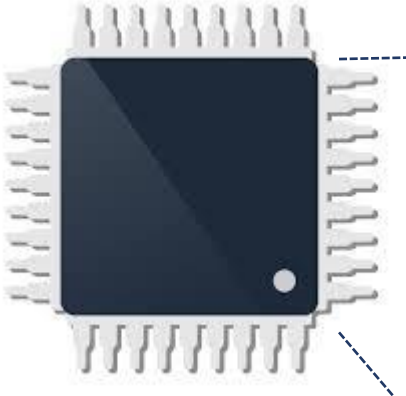
CDC-RDC Hierarchical Flow



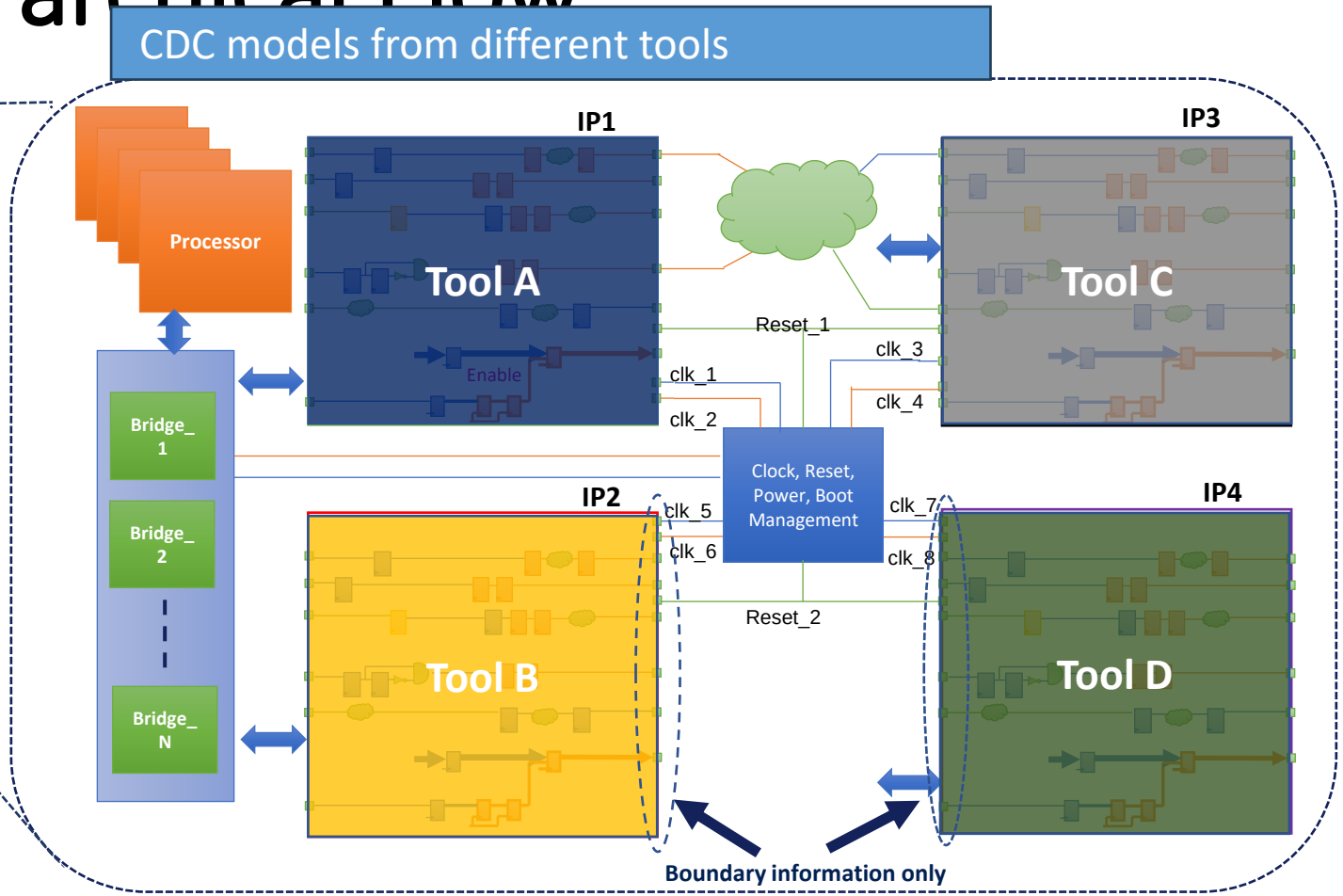
- Abstraction Models enable all Boundary related CDC-RDC info required at SoC integration Checks
- Much better approach compared to Black Box



CDC-RDC Hierarchical Flow



- CDC models are currently lacking standardization.
- CDC models from different tools are not compatible.



2.1 Accellera CDC Working Group

- Presentation
- The five sub working groups
- Call for contribution

Accellera CDC WG initiative

- The WG was formed in Jan. 2023 to explore the need for the creation of a standard to converge CDC collateral integration from different tools/vendors for ease (time-to-market) and quality (bug-free silicon).

- Fundamentally, what is being proposed is a common CDC interface standard that:
  - Every vendor/tool can translate their native format to/from (maintaining their IP)
  - Every IP can run their tool-of-choice to verify and produce collateral, and generate the standard format for SOCs that use a different tool
  - Every SOC can quickly (time-to-market) and safely (quality) integrate either native collateral, or translate from the standard collateral into their tool-of-choice

- The Accellera CDC WG goal, as approved by the Accellera board is as follows:
  - Produce an LRM for publication
  - Enable all EDA vendors in developing tools that meet this specification in generated collateral
  - Enable IP companies to generate collateral using various vendor/tools
  - Enable SOC companies to consume generate collaterals from different vendor/tools into their tool-of-choice

What was the problem?

As we move from monolithic designs … to IP/SOC with IPs sourced from a small/select providers … to sourcing IPs globally (to create differentiated products) …

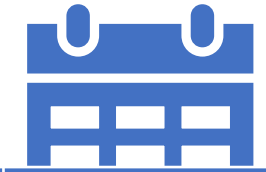
We must maintain quality as we drive faster time-to-market

In areas where we have standards (SystemVerilog, OVM/UVM, LP/UPF), the integration is able to meet the above (quality, speed)

But in areas where we don't have standards (in this case, CDC), most options trade-off either quality, or time-to-market, or both :-(

Creating a standard for inter-operable collateral addresses this gap

Accellera CDC WG initiative



Ver	Focus	Timeline
v0.1	CDC	Oct 2023
v0.3	RDC & Assertions	July 2024
v0.5	Complexities & Extensions	Dec 2024
v1.0	Final LRM release	Mar 2025

Pre-WG launched Sep ‘22 to evaluate need. WG launched Jan ’23

134 members from 24 companies (as of Sept 06 ‘24)

5 active sub-groups: Output-Collateral, Format, Assertions, Testing, Training

Agnisys	Aldec	AMD	AMS	Analog Devices	ARM	Arteris	Blue Pearl Software
Cadence	Huawei	Infineon	Intel	Marvel	Microchip Technologies	Microsoft	NVIDIA
NXP	Qualcomm	Renesas	Robert Bosch	Siemens EDA	ST Micro	Synopsys	Verilab

Accellera CDC WG Scope

Tool-agnostic interoperable collateral

Supporting hierarchical CDC/RDC/Glitch structural analysis

Human readable, and machine parseable

LP/UPF compliant

Multi-mode/param/instance comppliant

Covering majority of common interface protocols (e.g. AMBA, UCIe, etc.)

Constraints/Assumptions can be verified with SVAs

Can meet other needs (e.g. FPGA, Analog)

2.2 Output Collateral Subgroup

Output Collateral Subgroup

Output-Collateral

Attributes for CDC/RDC block model

Address most industry standard interfaces

Identify limitations and extensions for the attributes

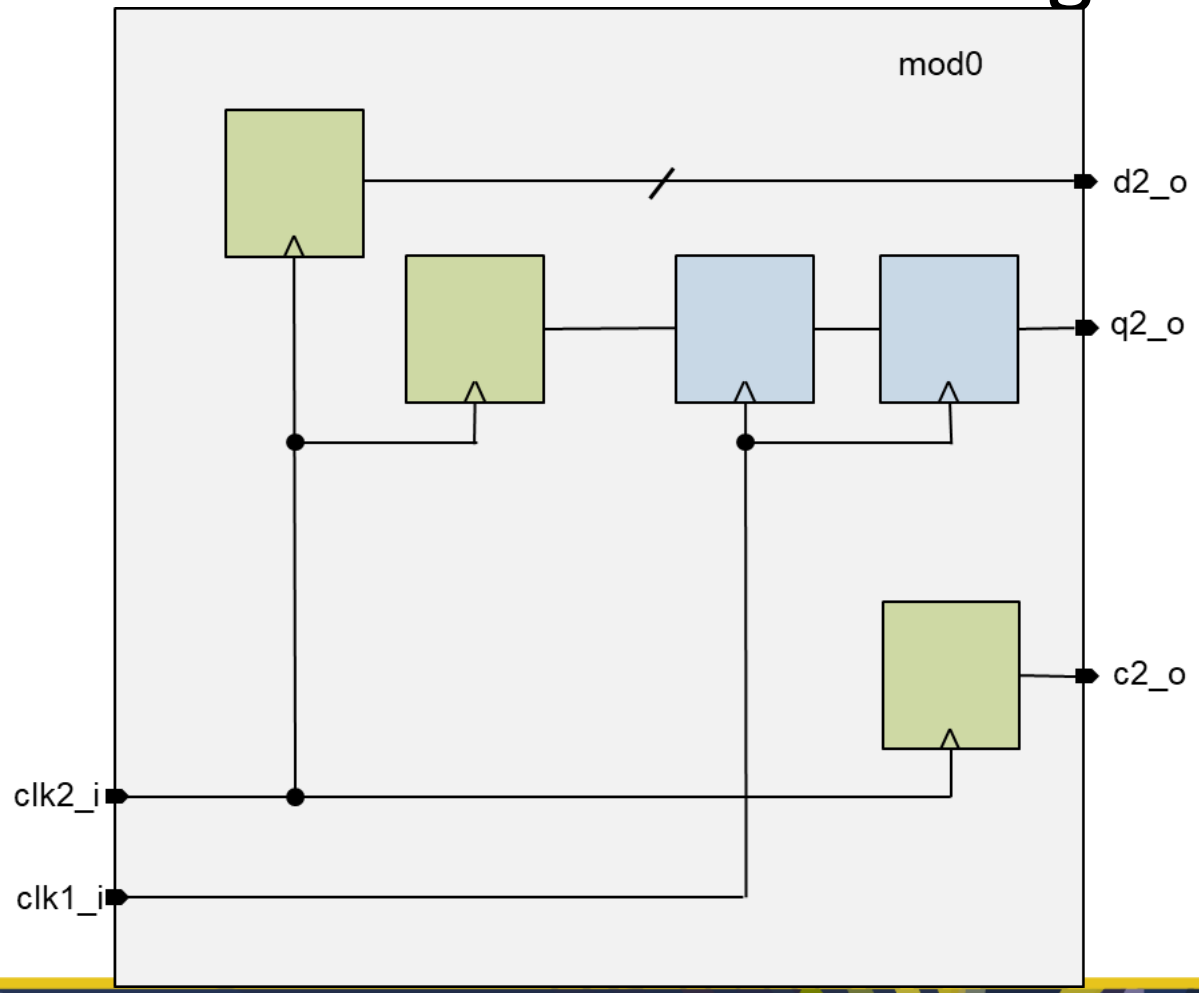
Attributes Table in LRM v0.3

Domain	Attribute	Type	Values	Mandatory
module	name	string	{module name}	Yes
parameter	name	string	{parameter name}	Yes
parameter	value	range-list	{values}	Optional
parameter	type	defined set	{string, Boolean, number (hex, decimal, oct, binary)}	Optional
parameter	ignore	Boolean	{true, false}	Optional
port	name	string	{port name}	Yes
port	direction	defined set	{input, output, inout}	Yes
port	type	defined set	{data, clock, virtual_clock, async reset, cdc_control, rdc_control, virtual_reset}	Yes
port	logic	defined set	{combo, buffer, inverter, glitch-free-combo, internal-sync}	Optional
port	cdc_control_from _clock	; separated list	{clock-names}	Optional
port	associated_from_clocks	; separated list	{clock-names}	Yes
port	associated_to_clocks	; separated list	{clock-names}	Optional
port	associated_inputs	; separated list	{ports}	Optional
port	associated_outputs	; separated list	{ports}	Optional
port	cdc_control	; separated list	{associated-ports}	Optional
port	polarity	defined set	{high, low, low_high}	Yes
port	ignore	defined set	{blocked, hanging}	Optional
port	cdc_static	; separated list	{clock-names}	Optional
port	constant	; separated list	{binary, hex, and of any length}	Optional
port	gray_coded	Boolean	{true, false:default}	Optional
port	clock_period	string	{clock period}	Optional
port	associated_from_reset	; separated list	{reset-names}	Optional
port	associated_to_reset	; separated list	{reset-names}	Optional
port	rdc_control_from_reset	; separated list	{reset-names}	Optional
port	rdc_control_to_reset	; separated list	{reset-names}	Optional
port	rdc_control_to_clock	; separated list	{clock-names}	Optional
port	rdc_clock_gate_location	defined set	{external or internal}	Optional
tool	name	string	{tool name}	Yes
tool	version	string	{tool Version}	Yes
design	version	string	{design milestone}	Optional
design	date	string	{collateral generation date}	Yes
design	username	string	{user/tool that generated the collateral}	Optional
design	description	string	{description}	Optional
set_cdc_clock_group	clocks	; separated list	{clock-names}	Yes
set_cdc_clock_group	name	string	{group-name}	Optional
set_reset_group	reset	; separated list	{clock-names}	Yes
set_reset_group	name	string	{group-name}	Optional

Output-Collateral

Port Attribute Modelling

Output-Collateral



Blue boxes are in clk1_i clock domain
Green boxes are in clk2_i clock domain

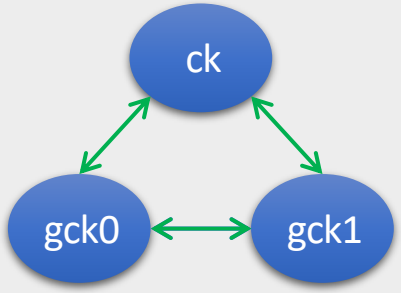
example in Tcl-format for output interface

```
cdc_set_module mod0

    cdc_set_port d2_o                    \
        -type                    data    \
        -direction               output  \
        -associated_from_clocks  clk2_i  \
        -cdc_control             q2_o

    cdc_set_port q2_o                            \
        -type                    cdc_control     \
        -direction               output          \
        -cdc_control_from_clock  clk2_i          \
        -associated_from_clock   clk1_i          \
        -associated_outputs      d2_o

    cdc_set_port c2_o                    \
        -type                    data    \
        -direction               output  \
        -associated_from_clock   clk2_i
```

cdc_set_clock_group (Tcl format)

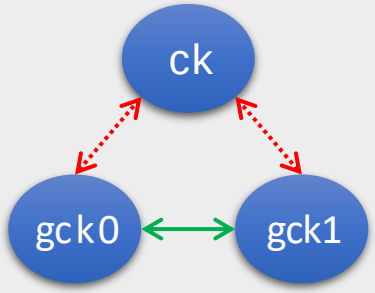
Output-Collateral

1 domain



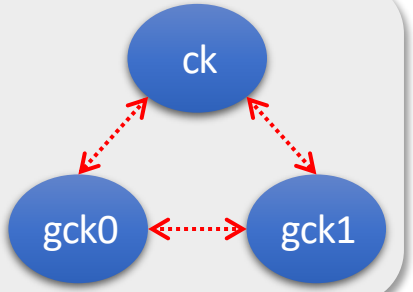

```
cdc_set_clock_group -name common_domain -clocks {ck gck0 gck1}
```

2 domains



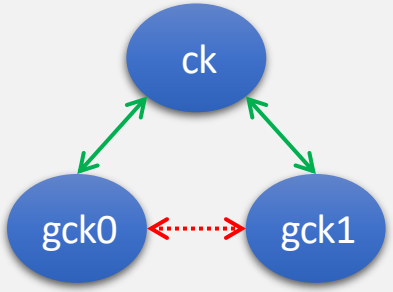

```
cdc_set_clock_group -name small_domain -clocks {ck}
cdc_set_clock_group -name large_domain -clocks {gck0 gck1}
```

3 domains




```
cdc_set_clock_group -name domain_C -clocks {ck}
cdc_set_clock_group -name domain_0 -clocks {gck0}
cdc_set_clock_group -name domain_1 -clocks {gck1}
```

2 clock branches



compatibility sets: common members allowed

```
cdc_set_clock_group -name left_branch  -clocks {ck gck0}
cdc_set_clock_group -name right_branch -clocks {ck gck1}
```

2.3 Format Subgroup

Format Subgroup Mission

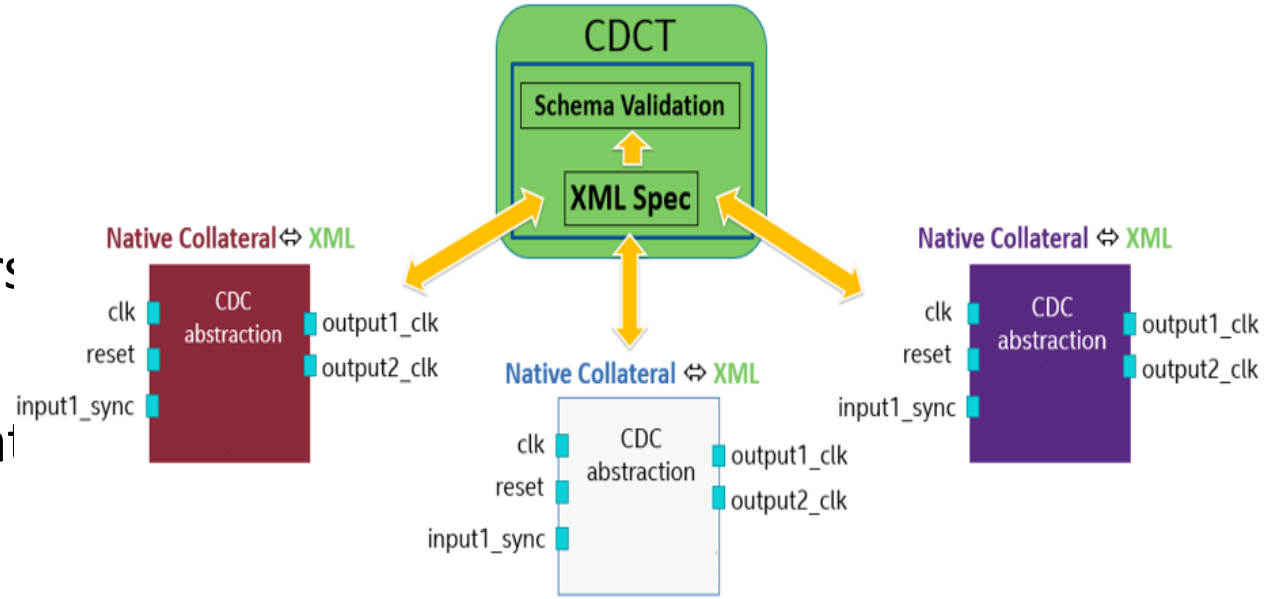
Format

- **Goal**
  1. Determine exact format for domain specific language that can be used to capture required attributes/data from input/output/verification collaterals.
  2. Ensure quality in terms of compliance to spec.
- **Methodology**
  1. List different options like IP-XACT, TCL, Excel, JSON, etc.
  2. Experiment with populating the formats to ascertain the ability to meet the requirements.
  3. Determine pros and cons for each option of format.
  4. Recommend a final format post CDC Workgroup approval

Feasibility Study

Format

- A limited feasibility study for CDC
  - conducted on a subsystem with multiple IPs connected by AMBA interfaces
  - across three different vendor tools
  - With limited support from the vendors
- Results:
  - 99.5% of what was identifiable in a flat run was also identifiable if the native abstraction collateral was replaced with an XML representation and translated across the vendor tools.



Primary Requirements

Format

- Describing IP
  - static or semi-static
  - IP-XACT is industry standard for IP definition and packaging
  - Use models of IP and Product companies
- Integration of IP
  - Dynamic environment requires programmability for CDC definition
  - Tcl is preferred and widely used in industry
  - Use models for Product and EDA companies

IP-XACT vs Tcl

Format

- IP-XACT
  - IP-XACT is perfect for static representation
  - Useful for IP Delivery and SoC Integration
  - Infrastructure required for converting existing proprietary formats to IP-XACT
- Tcl
  - Tcl handles dynamic and conditional CDC scenarios better
  - EDA companies currently supports proprietary formats that are Tcl like
  - Human readability issue

- CDC Workgroup voted to use combination of both Tcl and IP-XACT

Format Subgroup

Format

- EDA companies to provide transformers for Tcl to/from IP-XACT
  - Also provide translators to and from its native format from and to the standard format
- Standard is tool agnostic
- IP providers have option to choose tools
  - to verify and produce collateral
  - to generate the standard format for SoCs that use a different tool
- The format is released as part of LRM ver 0.3 in July
  - Tcl API commands capturing and handling clock domains and attributes
  - IP-XACT schema for CDC as Accellera vendor extensions to the IP-XACT standard

2.4 Assertion Subgroup

Assertion Subgroup Mission

Assertion

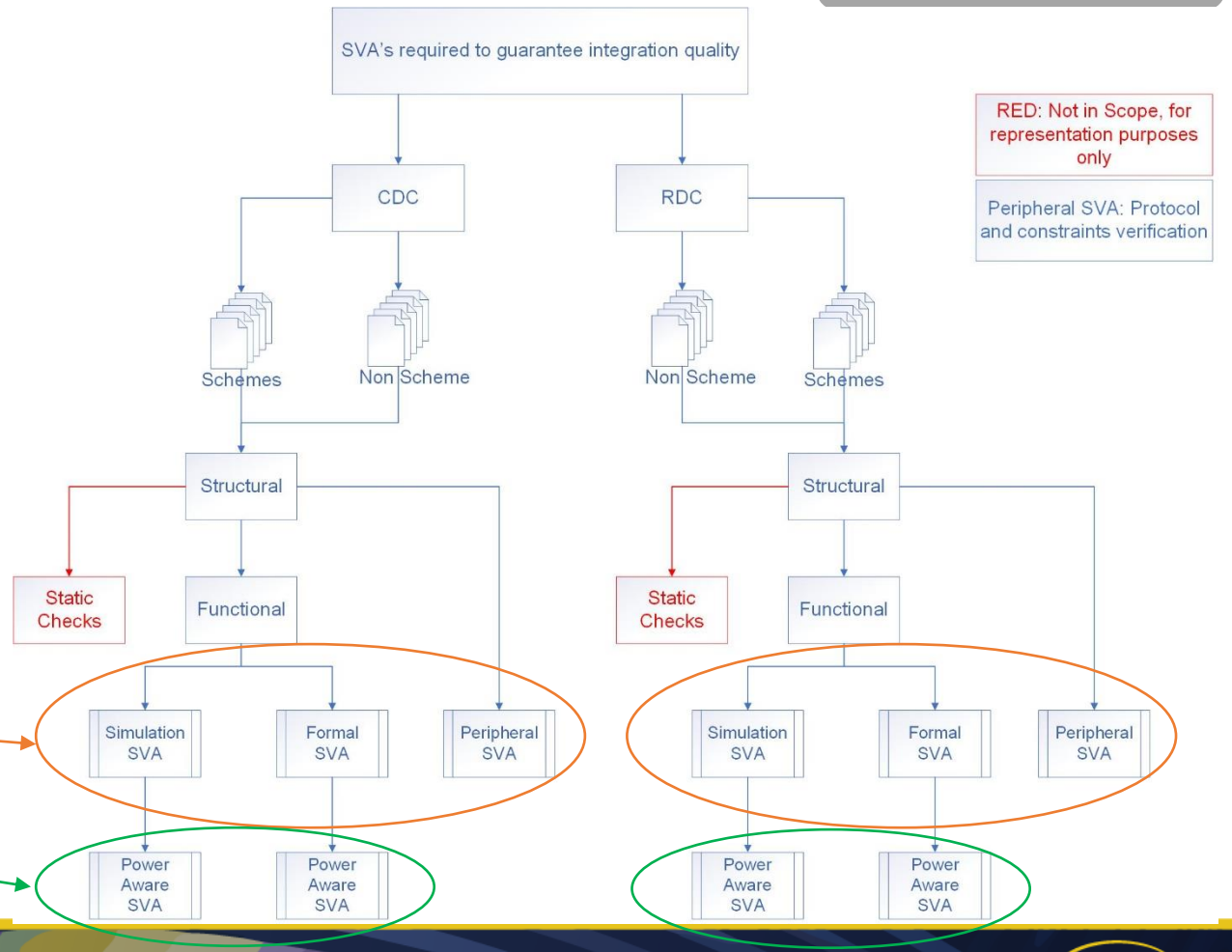
- **Goal**

  1. Produce Language Reference Manual (LRM) addendum for Assertions.
  2. Enable all EDA vendors in developing tools that meet specification for generating System Verilog Assertions (SVA) along with collateral.
  3. Enable Intellectual Property (IP) companies to generate SVA along with collateral using various vendors/tools.
  4. Enable System On Chip (SOC) companies to consume generated SVA from any vendor/tool into their tool of choice.

Assertion Subgroup

Assertion

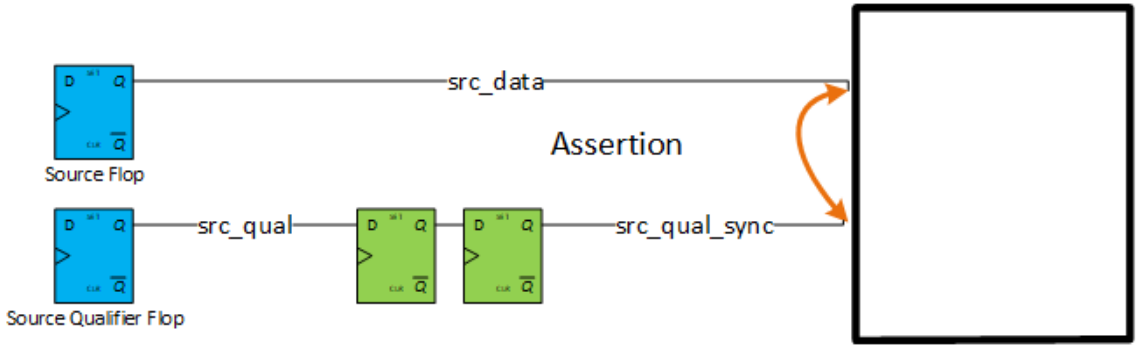
- CDC architectures are studied for possible verification strategies.
- Guidance to produce re-useable SVA for both Formal and Dynamic Verification.
- Guidance extracted from collateral.
- Guidance must follow SVA LRM.
- Guidance must be tool/vendor independent.
- Current Work
- Future Work

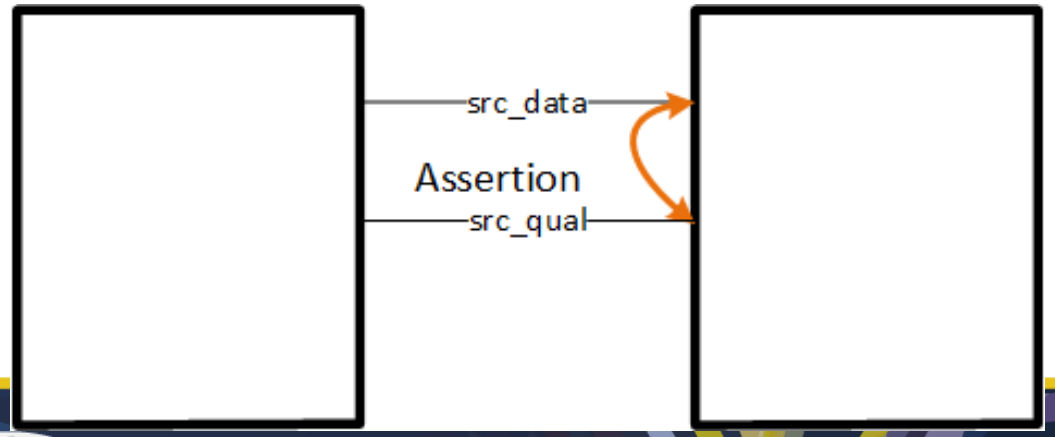


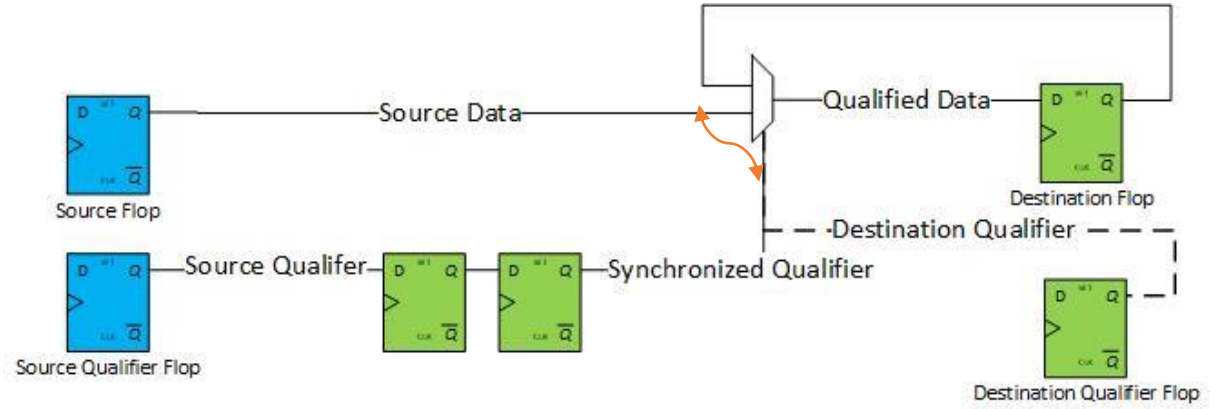
Assertion Subgroup

Assertion

- Possible Integration Scenarios under consider
  - Blackbox IP to Blackbox IP at SoC level.
  - SoC level glue logic to Blackbox IP.
  - Blackbox to SoC level glue logic.
  - Full Whitebox verification at IP level.







2.5 Testing Subgroup

Testing Subgroup Mission

Testing

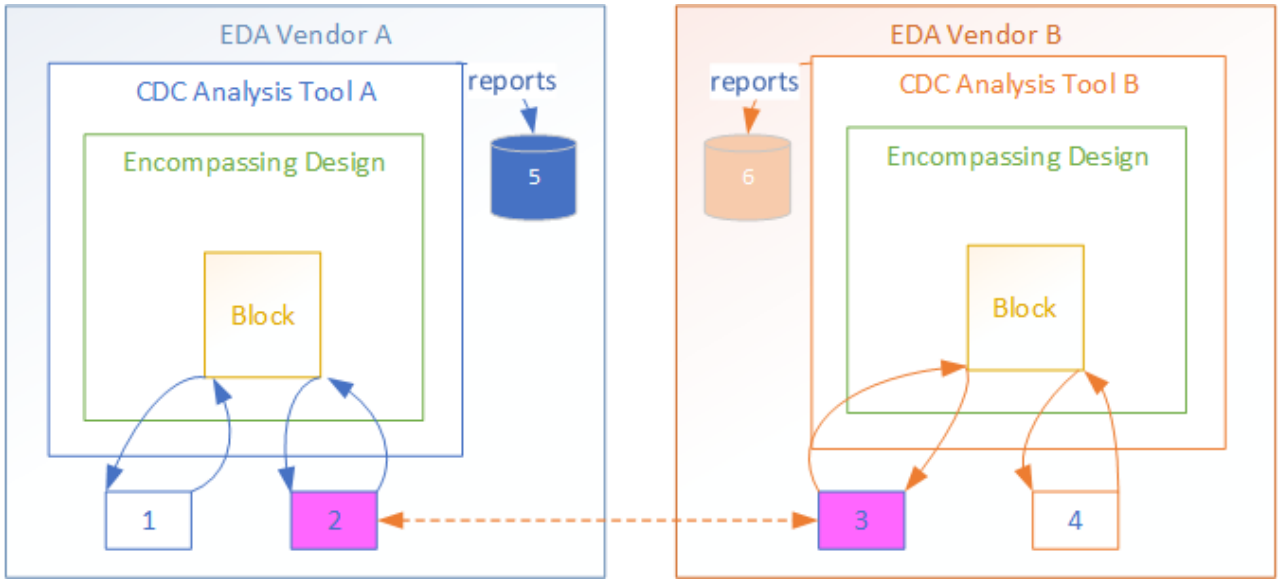
- **Goal**

  1. Evaluate the set of Accellera CDC attributes and protocols for completeness using multiple tools from multiple vendors.
  2. Demonstrate the use of the complete set of attributes and protocols as defined and formatted by other sub-groups.
  3. Provide RTL design examples within which the defined attributes and protocols can be further qualified and evaluated.

Methodology (#1)

Testing

- Testing by Tool Vendor A
  - Step#1 -Perform static flat CDC using Vendor A tool, creating the native block model (1) and Accellera abstract model (2)
  - Step#2 - Perform static hierarchical CDC using native block model (1) & using Accellera abstract model (2)
    - step#2.1 - Compare results flat vs hierarchical with native block model
    - step#2.2 - Compare results of hierarchical native vs Accellera abstract model(s)
  - Step#3 - Accellera to facilitate exchange of Accellera abstract model (2) with another tool for the same IP
    - step#3.1 - Perform static hierarchical CDC using Accellera abstract model by another tool vendor (3)
    - step#3.2 - Compare results of Accellera abstract model (2) and another tool vendor's Accellera abstract model (3)

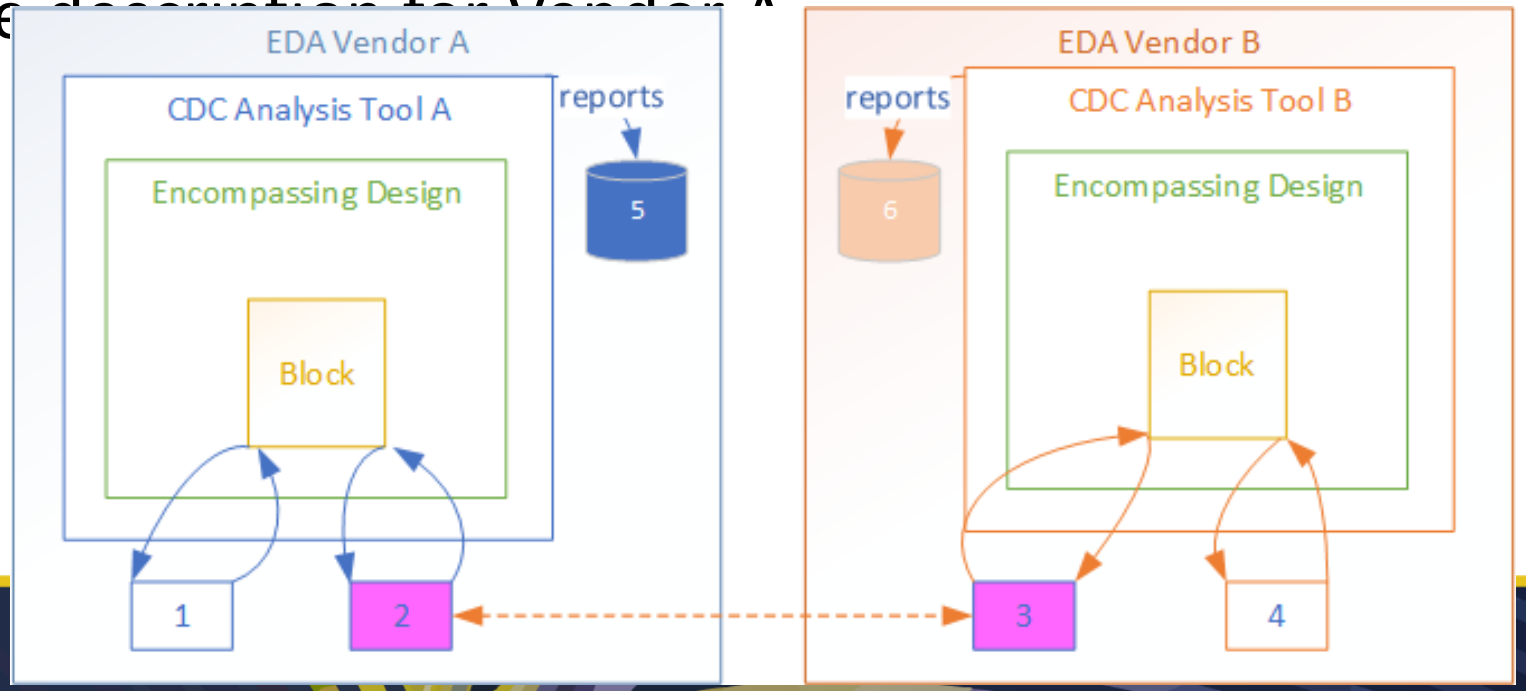


Methodology (#1) [cont…]

Testing

- Step#4 – Report fault model grading per step#2.2 and step#3.2. Fault grading information to be provided by Accellera CDC WG

- This process is of course symmetrical, and Vendor B performs tests to the above description for Vendor A



Methodology (#2)

Testing

- Testing by Non-tool Vendors
  - Participants with access to more than one required CDC tool can perform cross tool testing
  - Design IP per list of required interface protocol can be either an inhouse design if available or borrowed for the testing purpose (Accellera to facilitate).
  - EDA vendors to provide their tool support to participants

2.6 Training Subgroup

Training Subgroup Mission

Training

- **Goal**
  1. Raising awareness of the importance of defining a standard CDC-RDC model
  2. Provide generic documentation to let the CDC-RDC IP model user understand :
     - 1.1 CDC-RDC basic knowledge
     - 1.2 List of attributes & definition (related to IP CDC-RDC features/properties) as defined and agreed by the main CDC WG
  3. Presentation of the hierarchical flow
     - 2.1 tool dependency issue
     - 2.2 necessity to create an inter operational CDC-RDC model
  4. Inter operational CDC-RDC model integration manual

Conferences

Training

- Accellera CDC WG work prom o fere
  - Past/current conferences
    - DVCON Europe 2023
    - DVCON US 2024
    - DVCon Japan 2024
    - DVCON India 2024
    - DVCON Europe 2024
  - Targeted conferences (To Be Confirmed)
    - DVCON US / Japan / India / Europe / Taiwan / China 2025
    - DAC 2025
    - DATE 2025
    - VLSI 2025

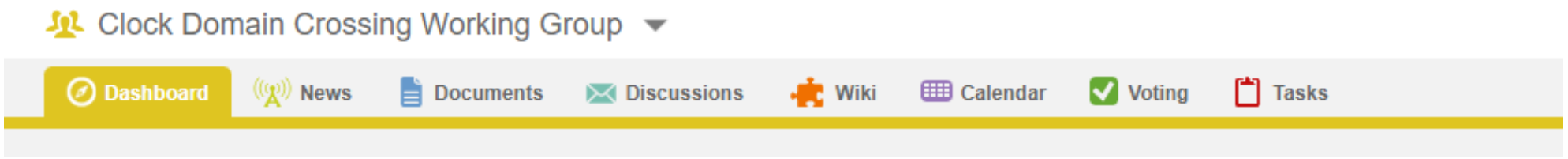
CDC LRM Draft 0.3 was open for public review till Sept 9, 2024

Clock Domain Crossing Standard Draft 0.3	Public review open through Sept. 9, 2024	2024-07-11

https://accellera.org/downloads/drafts-review

Call for Contribution !

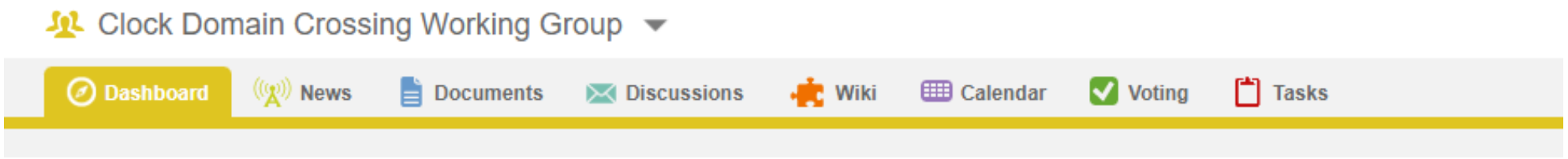
Accellera CDC Working Group



https://workspace.accellera.org/wg/CDC

Call for Contribution !

Accellera CDC Working Group



https://workspace.accellera.org/wg/CDC

Non-Accellera members can join and provide feedback on the standard:
https://www.accellera.org/community

Questions

- Finalize slide set with questions slide

Guidelines (1)

- Please keep the default font size for main lines at 28pt (or 26pt)
  - And use 24pt (or 22pt) font size for the sub bullets
- Use the default bullet style and color scheme supplied by this template
- Limited the number of bullets per page.
- Use keywords, not full sentences
- Please do not overlay Accellera or DVCon logo’s
- Check the page numbering

Guidelines (2)

- Your company name and/or logo are only allowed to appear on the title page.
- Minimize the use of product trademarks
- Page setup should follow on-screen-show (4:3)
- Do not use recurring text in headers and/or footers
- Do not use any sound effects
- Disable dynamic slide transitions
- Limit use of animations (not available in PDF export)

Guidelines (3)

- Use clip-art only if it helps to state the point more effectively (no generic clip-art)
- Use contrasting brightness levels, e.g., light-on-dark or dark-on-light. Keep the background color white
- Avoid red text or red lines
- Use the MS equation editor or MathType to embed formulas
- Embed pictures in vector format (e.g. Enhanced or Window Metafile format)

Questions

- Finalize slide set with questions slide