

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

Accellera CDC Working Group Authors

Nomita
GOSWAMI
Aiyush
AGGARWAL
Rahul
PARASHAR



Ashish SONI
Jean-Christophe
BRIGNONE



life.augmented

Devender
KHARI

Jebin
MOHANDAS



Chetan CHOPPALI
SUDARSHAN



Suman
CHALANA



Bill
GASCOYNE



1

Farhad AHMED



Kranthi
PAMARTHI



Joachim
VOGES
John
JEBAKUMAR



Don MILLS



Jan
HAYEK



Presenters Introduction

Suman Chalana(Qualcomm)

- Suman is currently working as a Principal Engineer/manager at Qualcomm India Private Limited. She has 17+ Years of work experience in VLSI Front end flows e.g.: Static Verification(CDC, Lint, RDC) Automotive flow development, Synthesist, FV.
- Currently she is working as Static Verification Lead in CAD team, leading Static verification methodology / tool-flow deployment globally. Along with Static, she is also driving & managing the team for overall automotive Global CAD charter worldwide.

Jebin Vijai (Intel)

- Jebin has 20+ Years of experience in RTL Design and Verification Methodologies, working as Logic Design Methodology Engineer with expertise in Clock Domain Crossing, Reset Domain crossing, RTL Linting, Low Power, RTL Constraints Verification, Power Estimation and Formal
- Currently he is working on RTL Left shift solutions to improve the Quality and Turn around time

Ashish Soni (ST)

- Ashish Soni is working as a Principal Engineer at ST Microelectronics. With nearly 19+ years of experience, where he leads the Front-End methodology activities within the Automotive Group. He also leads ST's global static checks forums, focusing on Hierarchical CDC, RDC, Lint and RTL synthesis optimization solutions
- He specializes in Digital Design, CAD flows, and Methodology development.

Rahul Parashar (Agnisys)

- Rahul is currently working as an R&D Engineer-I in Agnisys Technology PVT. LTD. He has 2.5 years of experience in the Soc assembly and packaging, IP-Integration and IP_design
- Actively contributing to the CDC and Accellera Working Group.

Agenda

	Topic	Slide update/create	Presenter	Time
#0	Workshop introduction			5m
#1	CDC-RDC			15m
	CDC-RDC Basic Knowledge	Bill Gascoyne (Blue Pearl), Jan Hayek (Bosch Sensortec)	Suman Chalana (Qualcomm), Ashish Soni (ST)	3m
	Setup Constraints & Verification	Ping Yeung (Nvidia) Jebin Vijai (Intel)	Suman Chalana (Qualcomm), Ashish Soni (ST)	3m
	Structural CDC/RDC	Chetan Choppali Sudarshan (Marvell) Suman Chalana (Qualcomm)	Suman Chalana (Qualcomm)	3m
	CDC Assertion-Based Verification	Kranthi Pamarthi (Renesas Electronics), Don Mills (Arm)	Suman Chalana (Qualcomm)	3m
	CDC-RDC Hierarchical Flow	Jean-Christophe Brignone, Ashish Soni (STMicroelectronics)	Jebin Vijai (Intel)	3m
#2	Accellera CDC WG			25m
	Standard	Iredamola Olopade, Jebin Vijai (Intel)	Jebin Vijai (Intel)	3m
	Testing	Devender Khari (Agnisys)	Rahul Parashar (Agnisys)	3m
	Output	Joachim Voges (Infineon), Devender Khari (Agnisys)	Rahul Parashar (Agnisys)	3m
	Format	Kranthi Pamarthi (Renesas Electronics)	Rahul Parashar (Agnisys)	3m
	Assertion	Farhad Ahmed (Siemens EDA), Suman Chalana (Qualcomm)	Rahul Parashar (Agnisys)	3m
	Training	Jean-Christophe Brignone, Diana Kalel, Ashish Soni (ST)	Jebin Vijai (Intel), Ashish Soni (ST)	3m
	Call For Contribution	Jean-Christophe Brignone (ST), Anupam Bakshi (Agnisys)	Jebin Vijai (Intel), Ashish Soni (ST)	2m
	Q & A			5m

1.1 CDC-RDC Basic Knowledge:

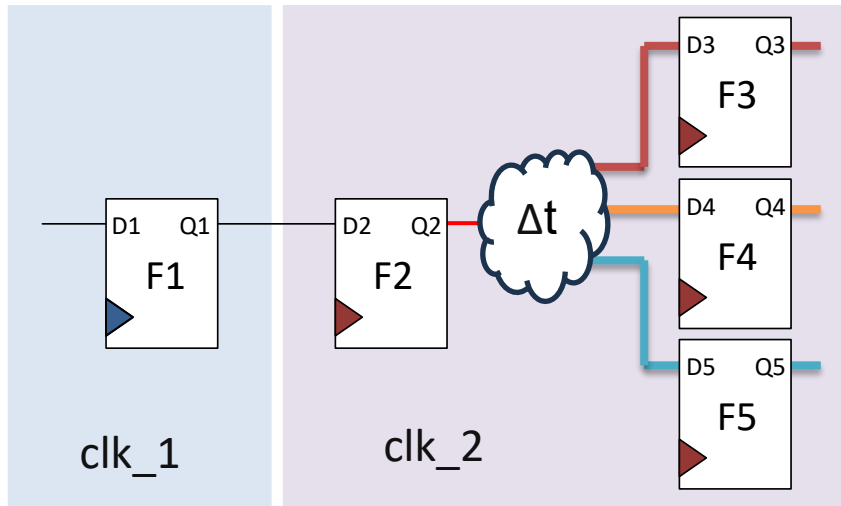
Coin Toss Analogy

- 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!

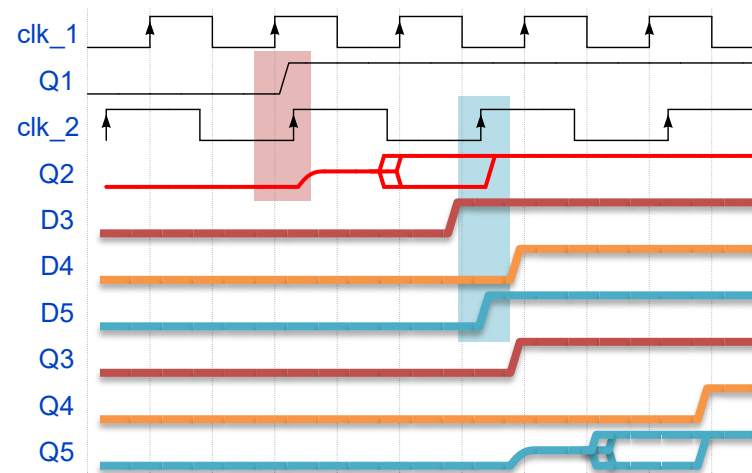


Why do CDCs need fixing?

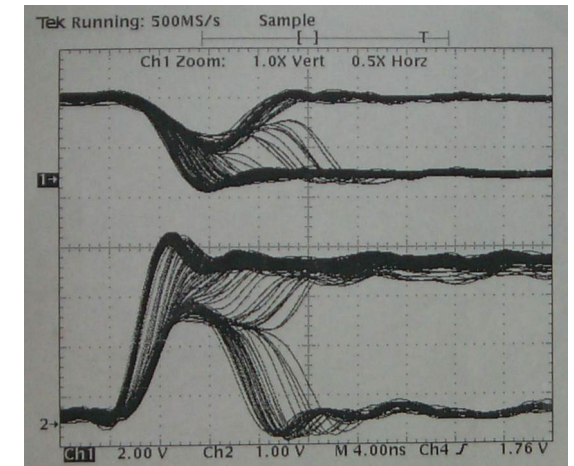
- Metastability
 - **Timing violations** on registers resulting in an indeterminate state lasting long enough to violate **timing assumptions** of the following circuits
... the coin decided too late / has not decided yet



(a) structure



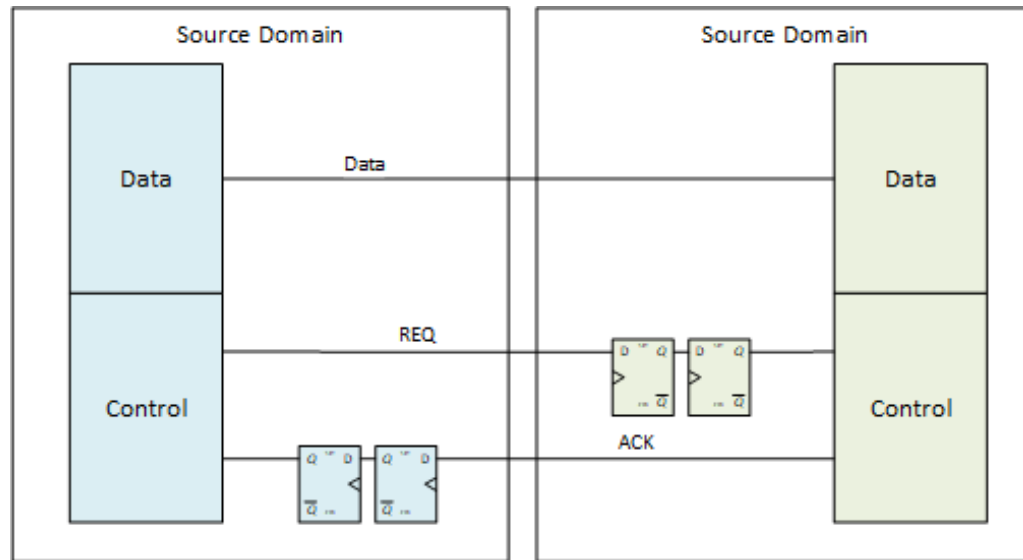
(b) waveform



(c) Physical representation

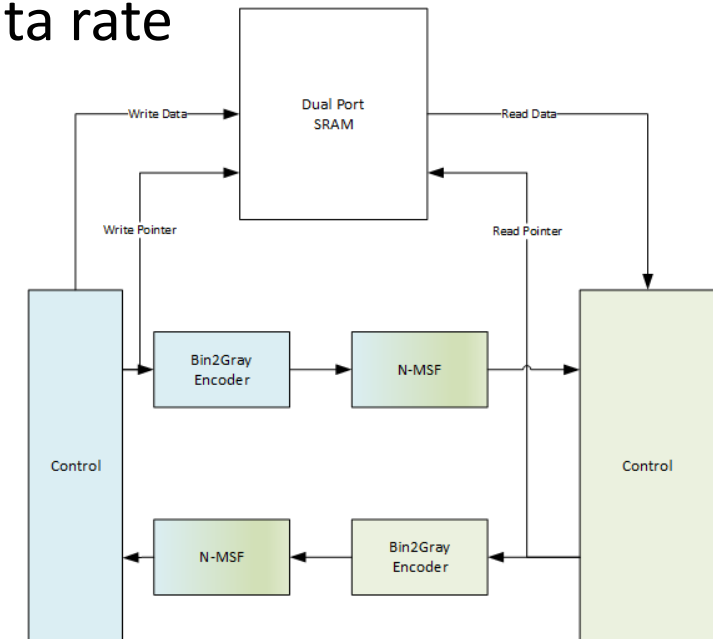
More Complex Synchronization

- Handshake Protocol



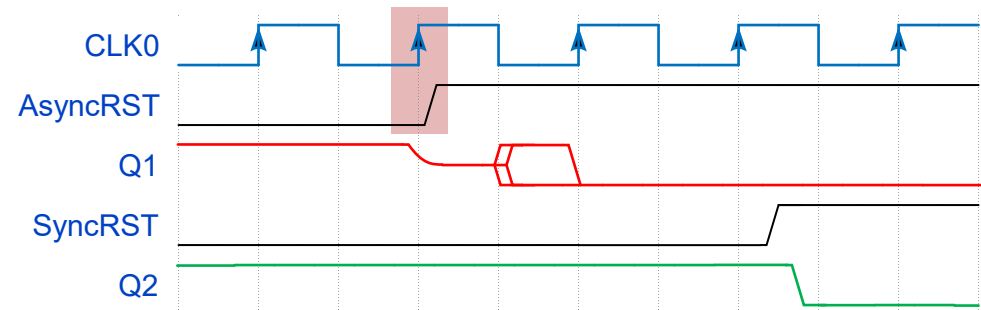
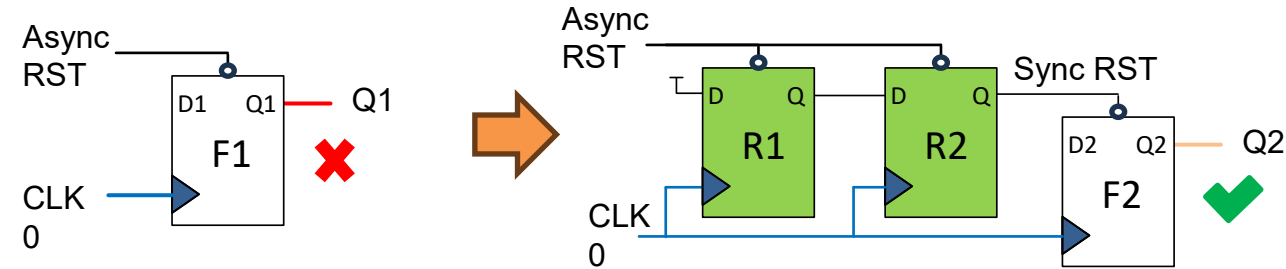
- FIFO

- Increased bandwidth
- Throttling
- Handles intermittent peaks of incoming data rate



Asynchronous Reset Release

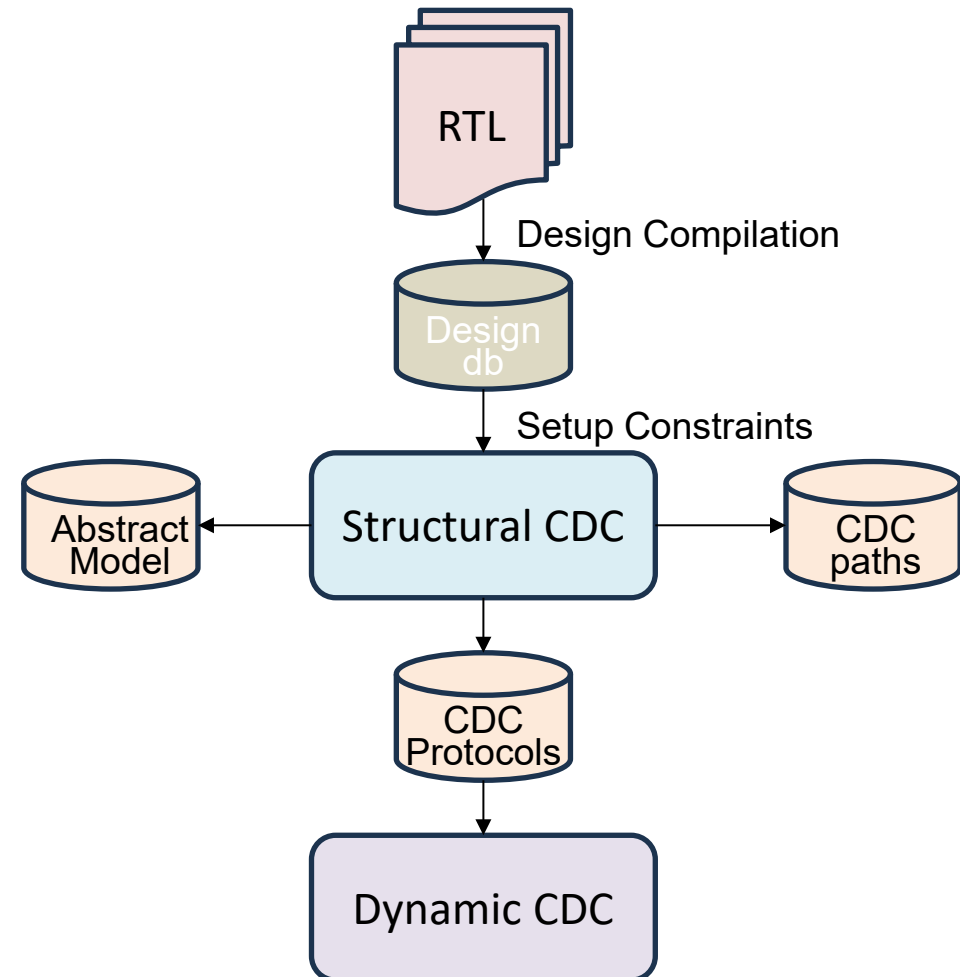
- In addition to setup- and hold-, DFF models also have **recovery-time**
 - Time between asynchronous Set/Reset release and clock when data and output are different
 - Violating recovery time is no different than violating setup/hold
- Possible to synchronize asynchronous reset on release edge only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



1.2 CDC SETUP & CONSTRAINTS

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



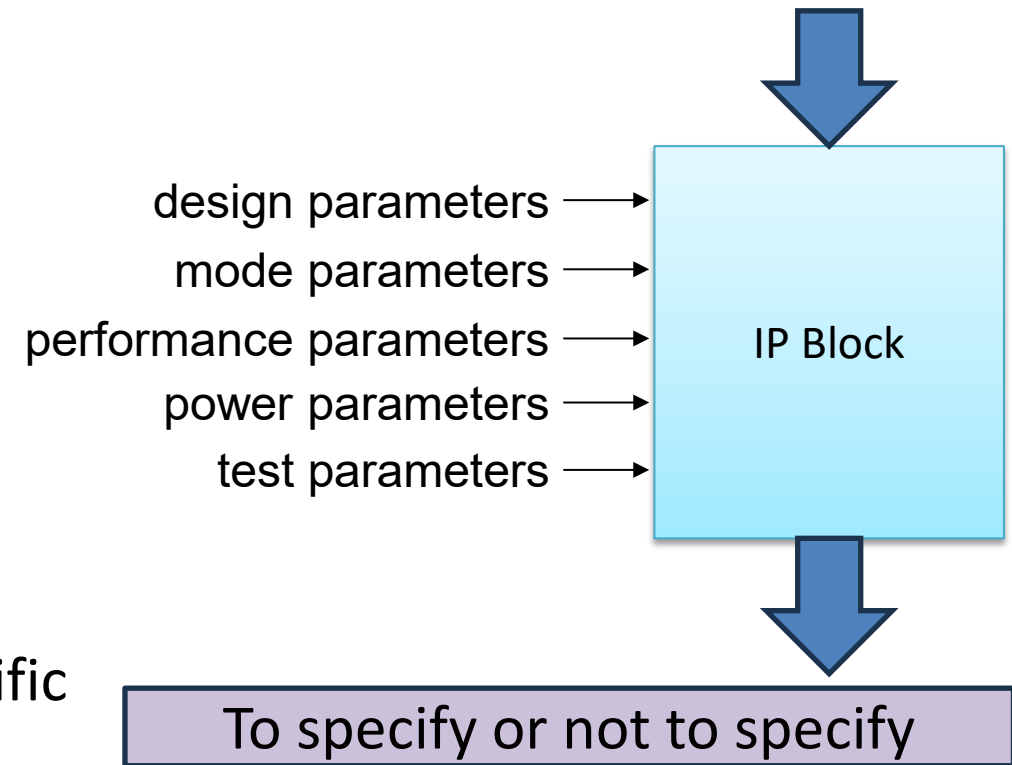
Reusing timing
constraints is risky

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

Clock groups for timing analysis \neq Clock groups for CDC analysis
Signal paths waived for timing analysis \neq Signal paths waived for CDC analysis

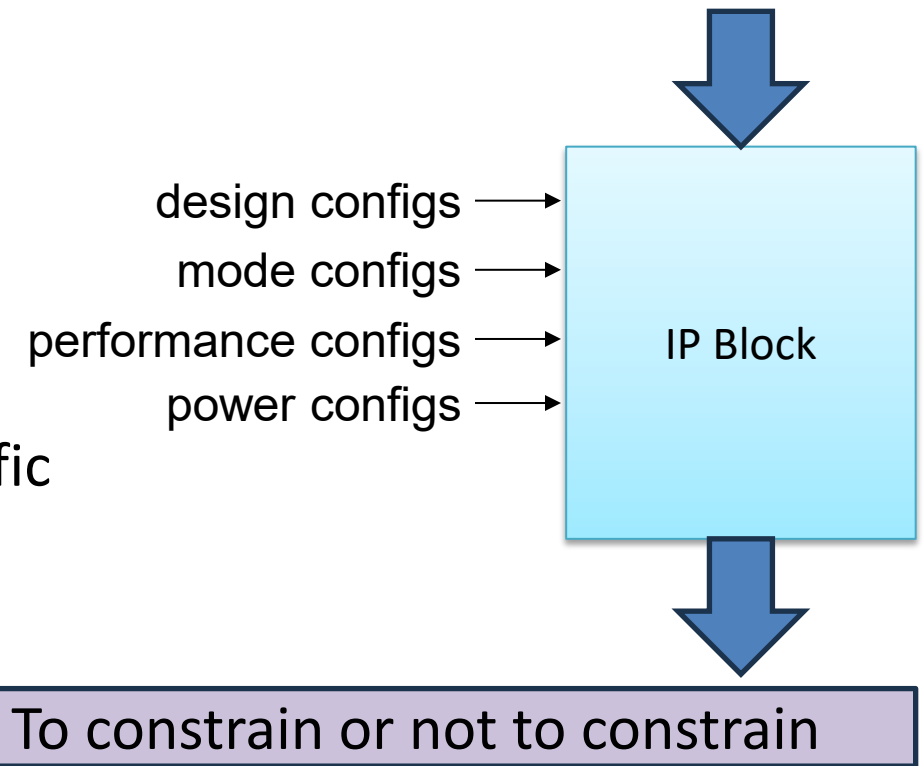
Challenge #1: Design Parameters

- 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



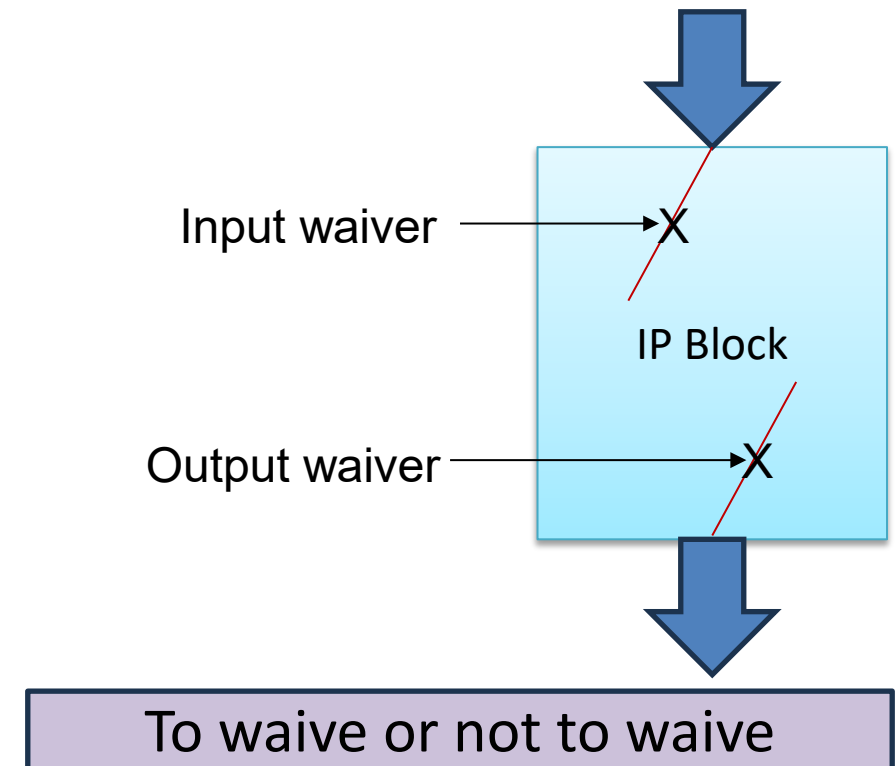
Challenge #2: Configuration Signals

- 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



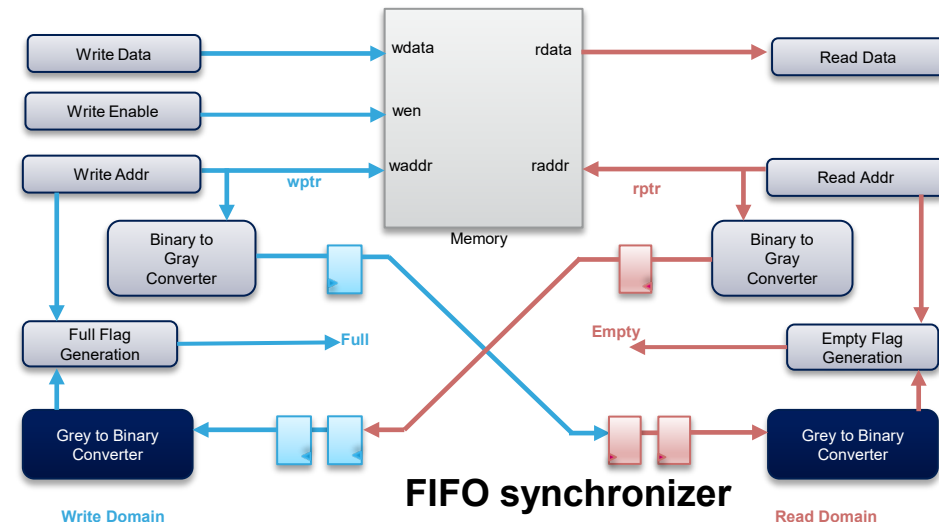
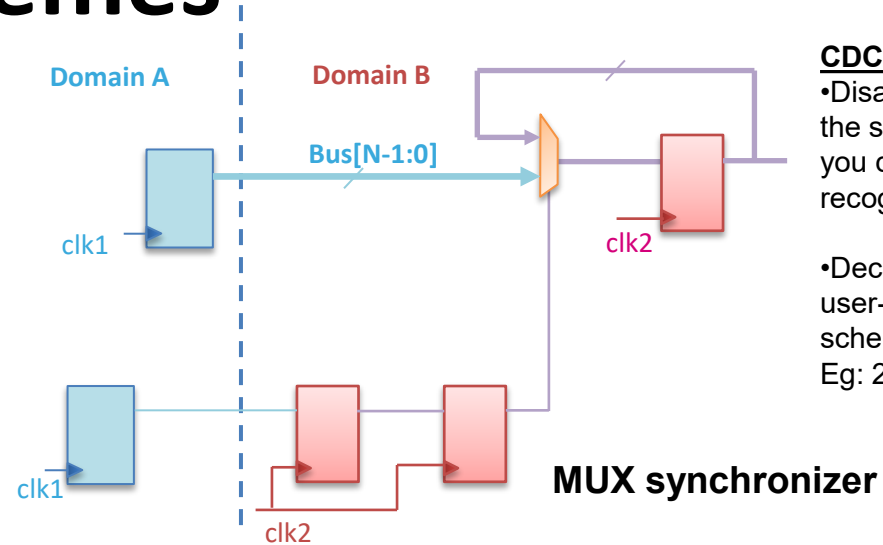
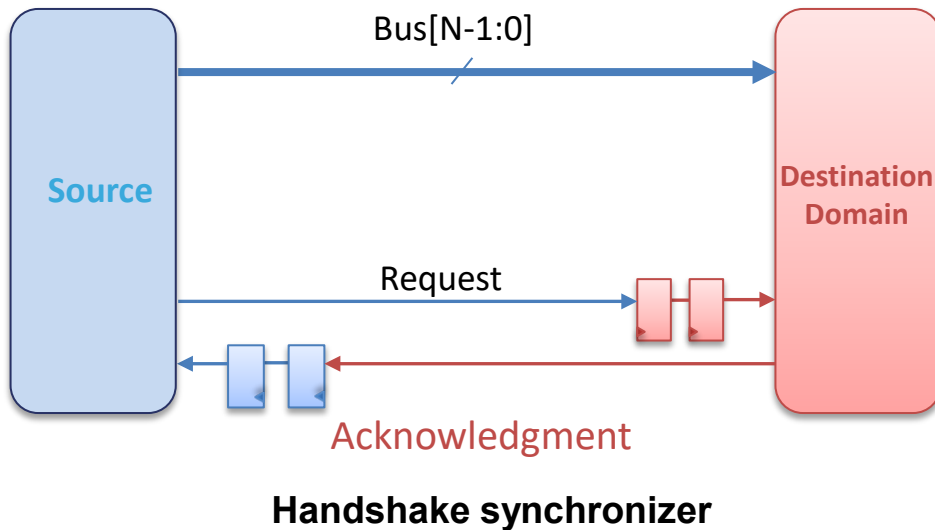
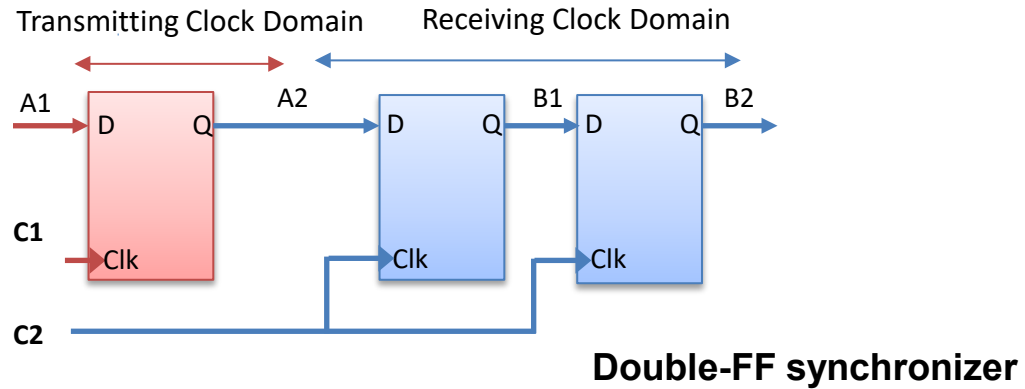
Challenge #3: CDC Waivers

- 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 must not be waived
- The Abstract Model will become waiver-specific



1.3 Structural CDC/RDC

Structural CDC - Commonly Used Synchronization Schemes



CDC Component:

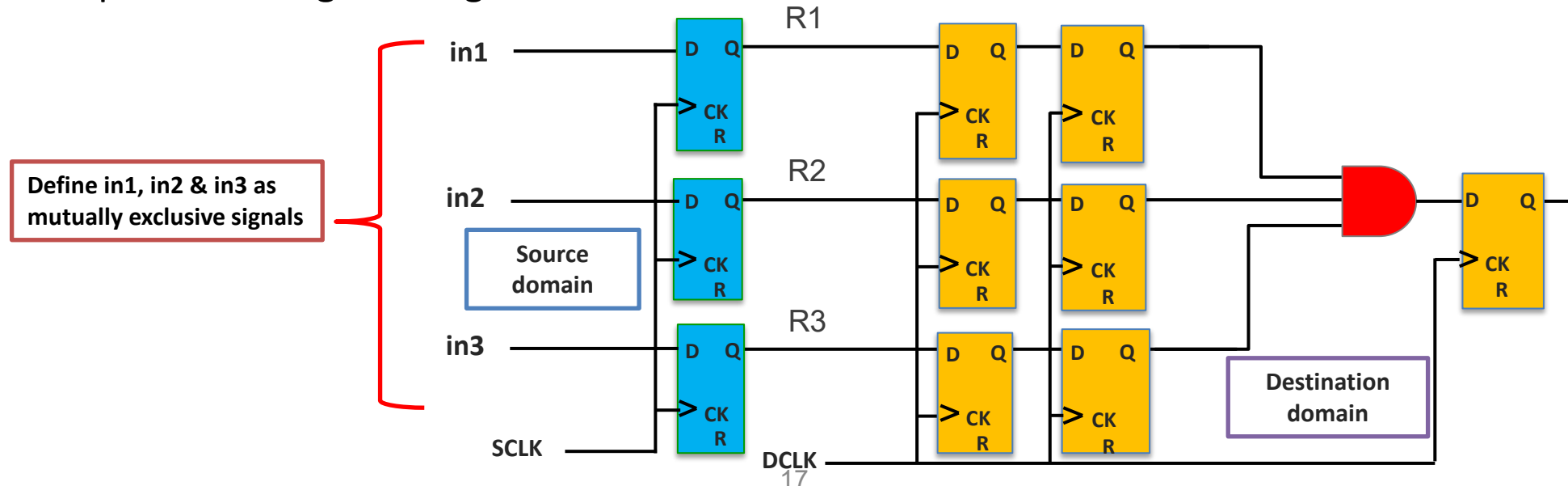
- Disable automatic detection of the specific synchronizer type that you don't want the tool to recognize automatically.

- Declare your own scheme as user-defined synchronizer (before scheme detection)
Eg: 2DFF cdccomponent

- 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)

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

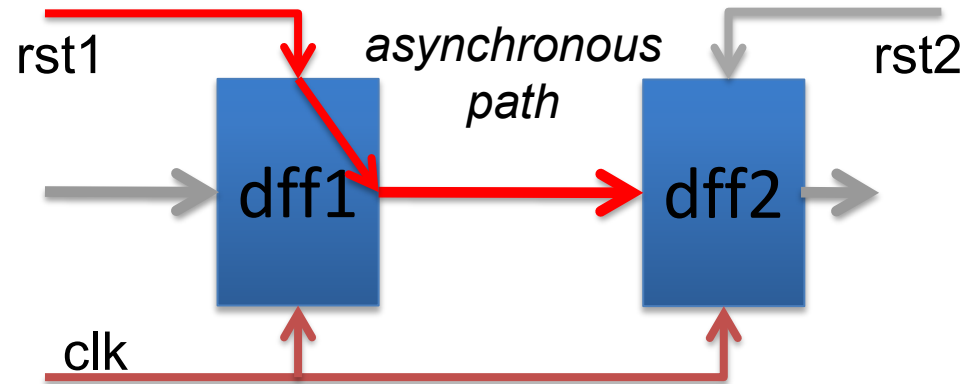


Structural CDC

- 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

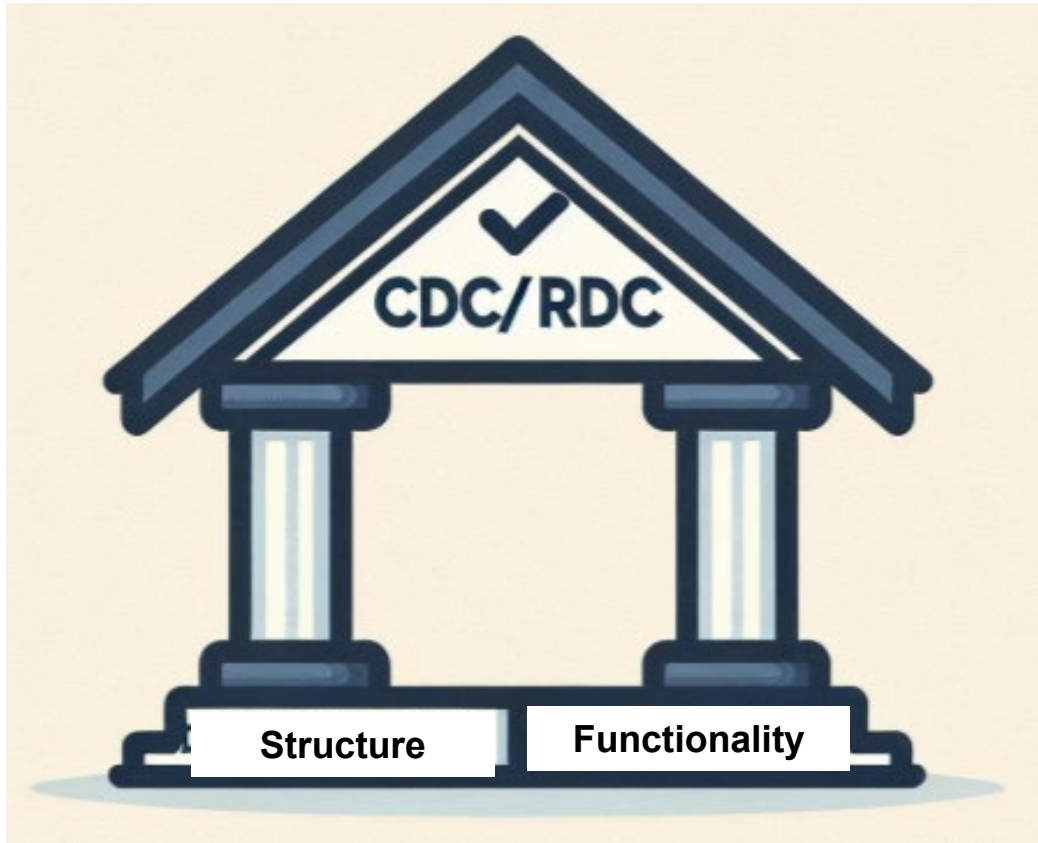
Reset-Domain Crossing

- 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

Structural Verification vs Assertion



- **Structural CDC/RDC Verification** analyze the design structure. It doesn't analyze the design functionality. As long as the structure is valid, the verification pass.
- **Assertions** covers design functionality. They can be used in Dynamic Simulation and/or Formal Verification.

**Valid CDC/RDC =
Correct Structure +
Correct Functionality**

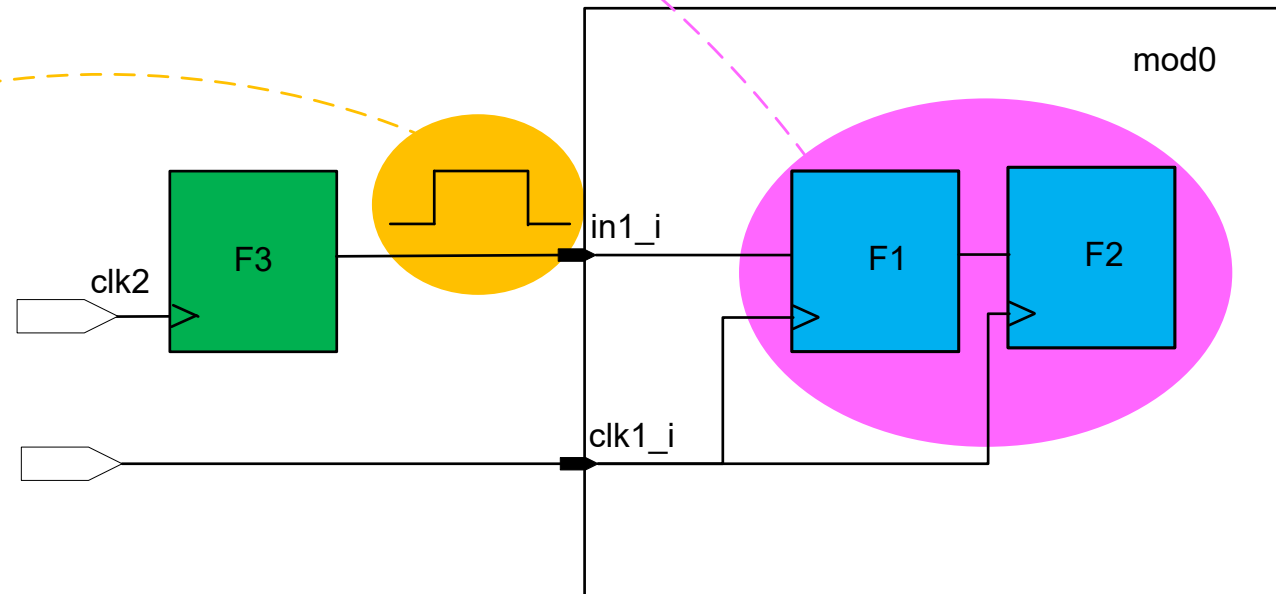
A Simple CDC Example

Correct structure:

There is a 2-DFF synchronizer between clock domains.

Correct functionality:

Pulse width of in1_i must be wide enough to be captured by clk1_i.



Functionality check is Covered by Assertion to check:
the minimum pulse width of the input data w.r.t. destination
clock clk1_i

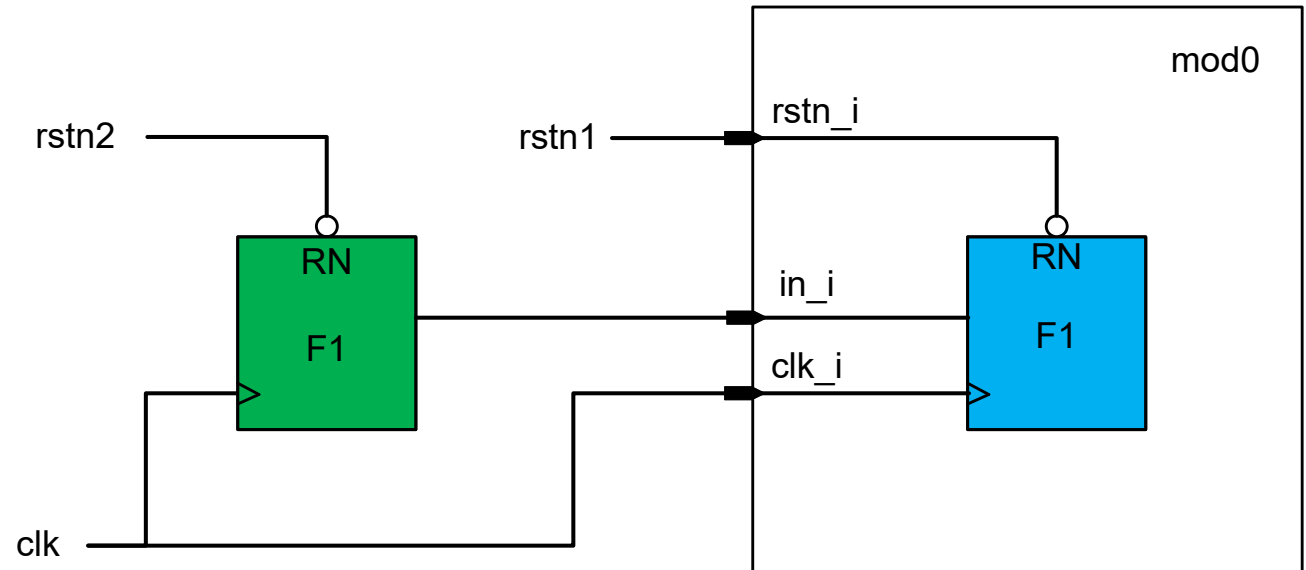
A Simple RDC Example

Structure:

Mod0 doesn't provide any RDC handling structure.

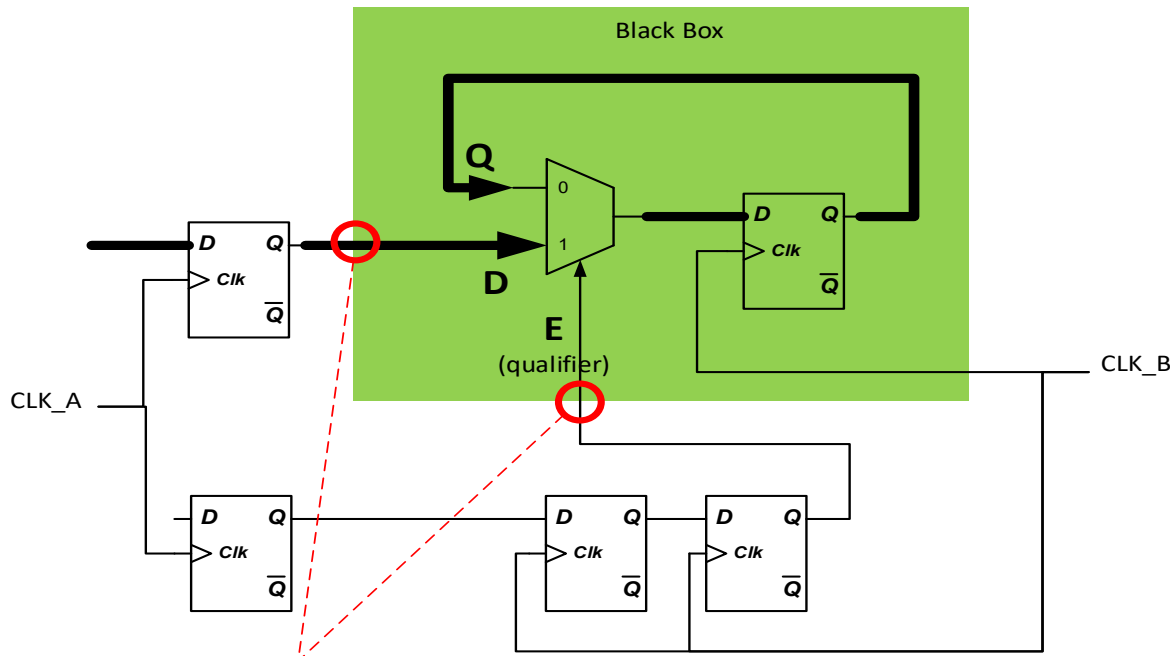
Correct Functionality:

rstn1 must be already 0 when rstn2 becomes 0.



Black Box Complexity

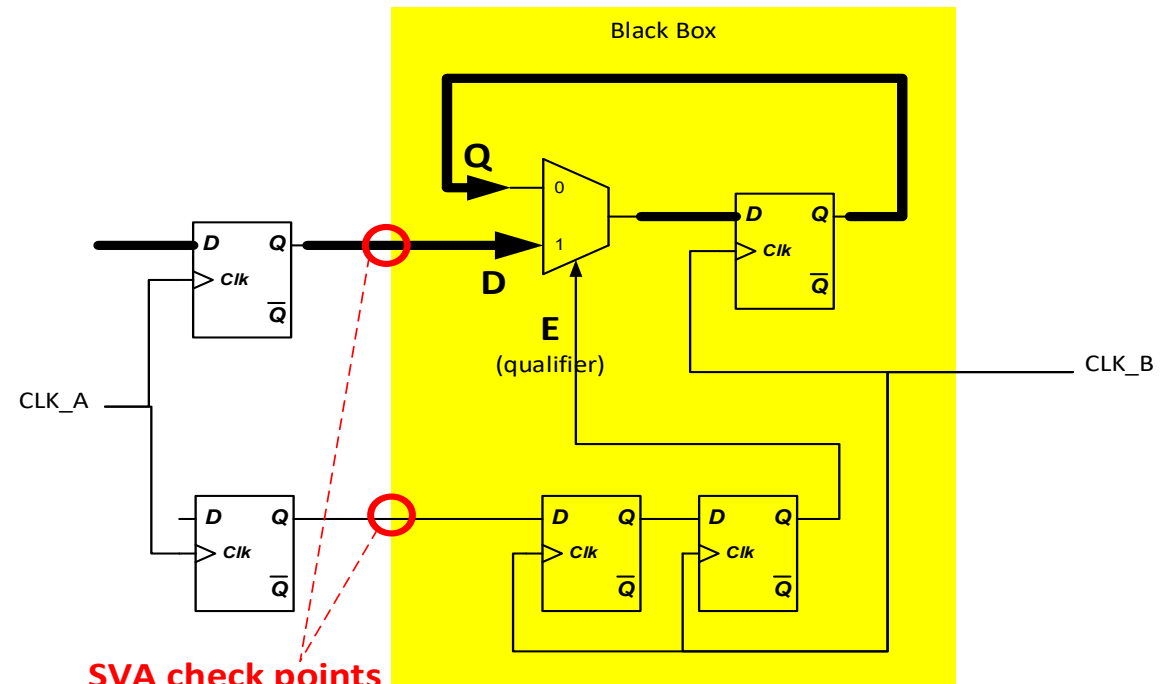
The assertion is applied at the black box boundary, leading to complexity in some cases as noted below in blue text.



SVA check points

Less complexity in this scenario:

1. Signal path delay due to physical implementation



SVA check points

Complexity in this scenario:

1. We might not know the synchronizer's structure since it is in the black box:
 - a. Number of DFFs
 - b. Active clock edges of each DFF
2. Signal path delay due to physical implementation

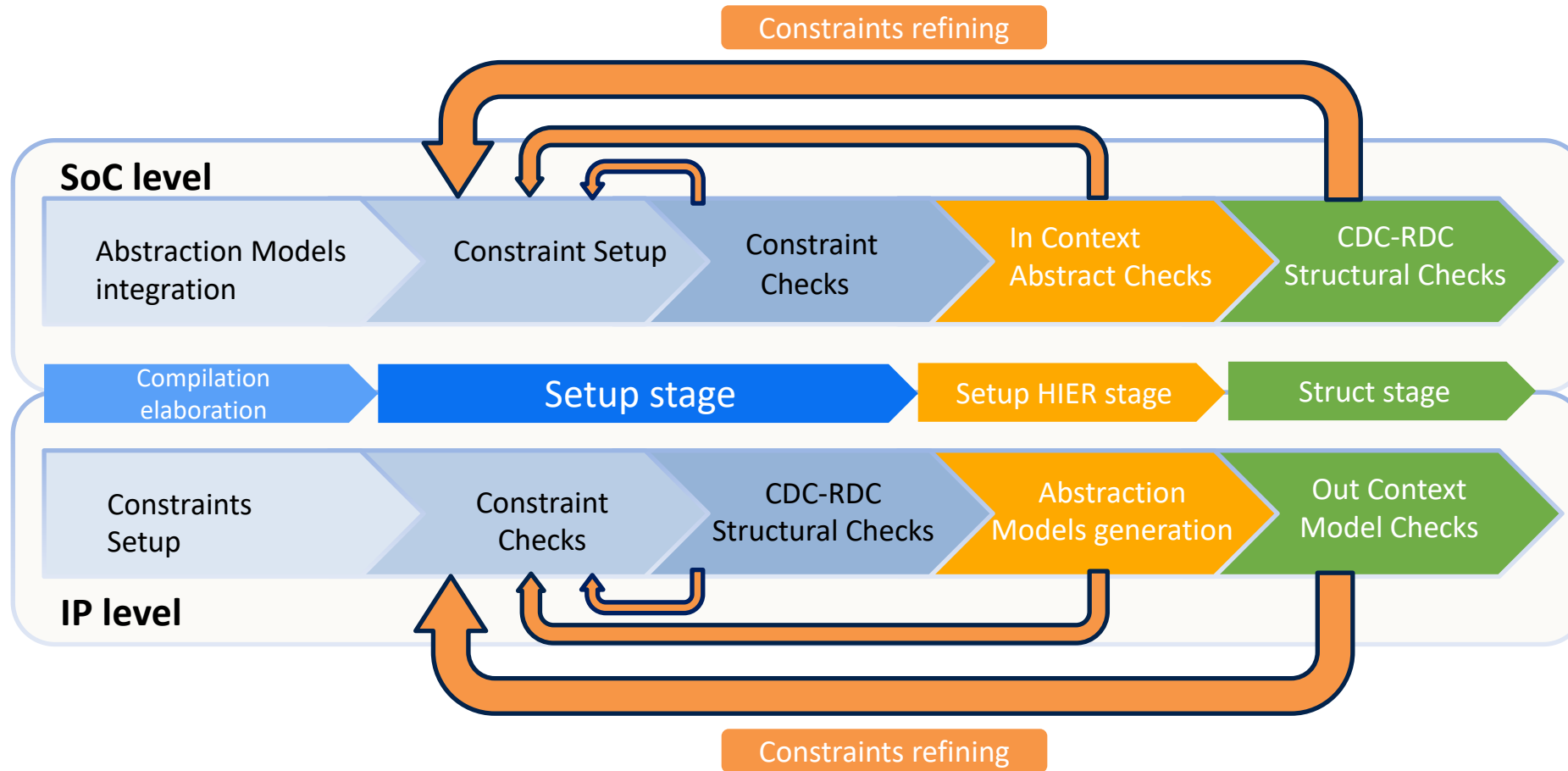
1.5 Hierarchical CDC/RDC

CDC-RDC Hierarchical Flow

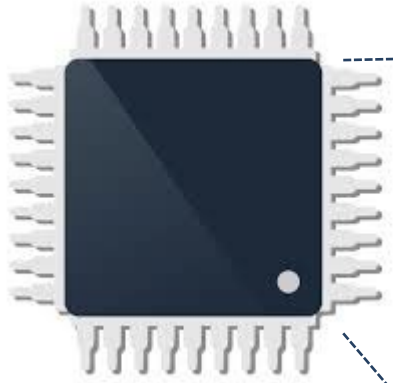
Why ?

- 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
 - Re-utilization of sub-blocks clean up effort at SoC level by using CDC-RDC models
 - Reduces SoC runtime & the total of violations to be analyzed at SoC level which leads to increase in productivity
- Challenges:
 - Dependency to sub-blocks provider to deliver CDC-RDC model
 - Compatibility of CDC-RDC models in case of multiple EDA tools usage

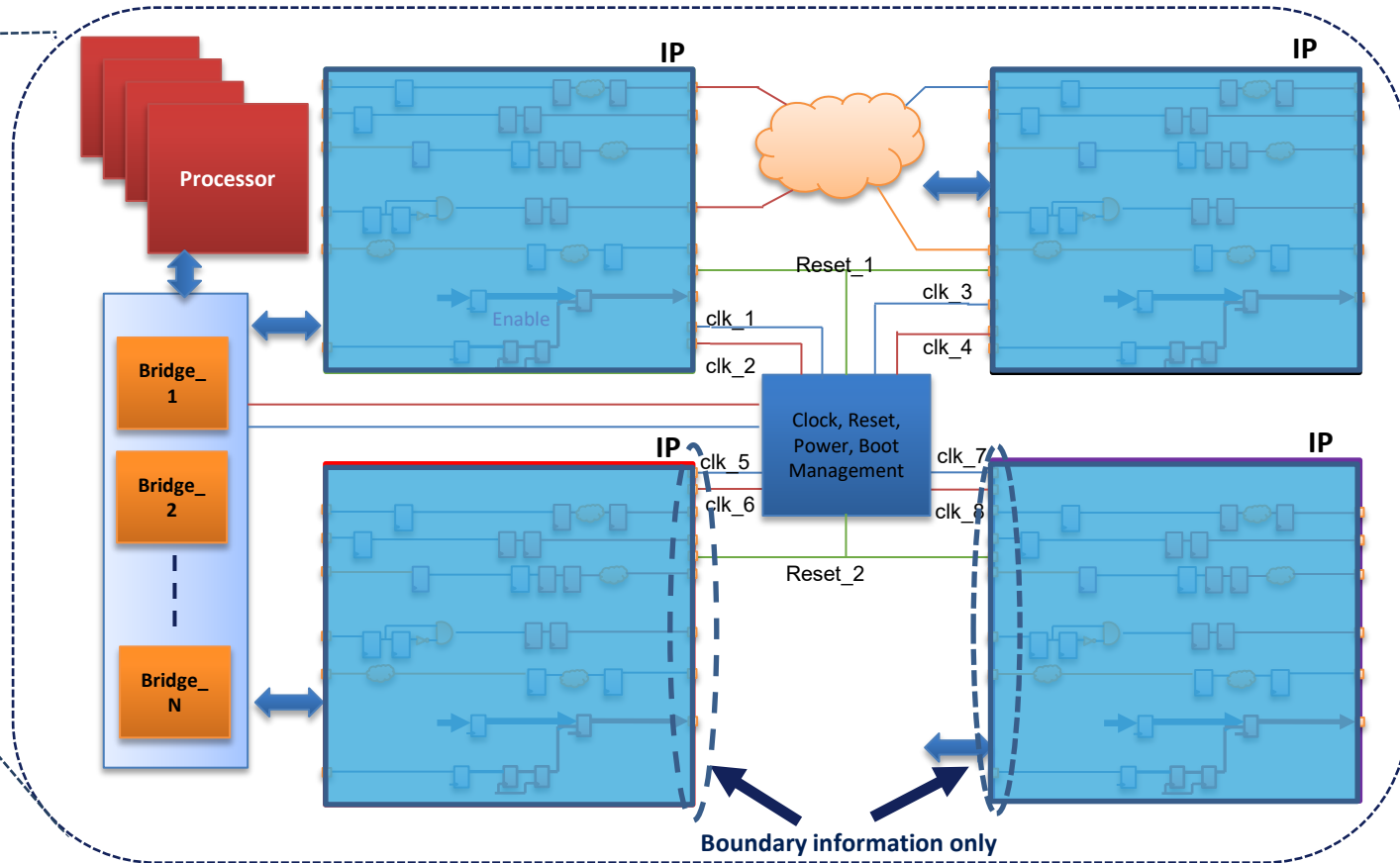
CDC-RDC Hierarchical Flow



CDC-RDC Hierarchical Flow



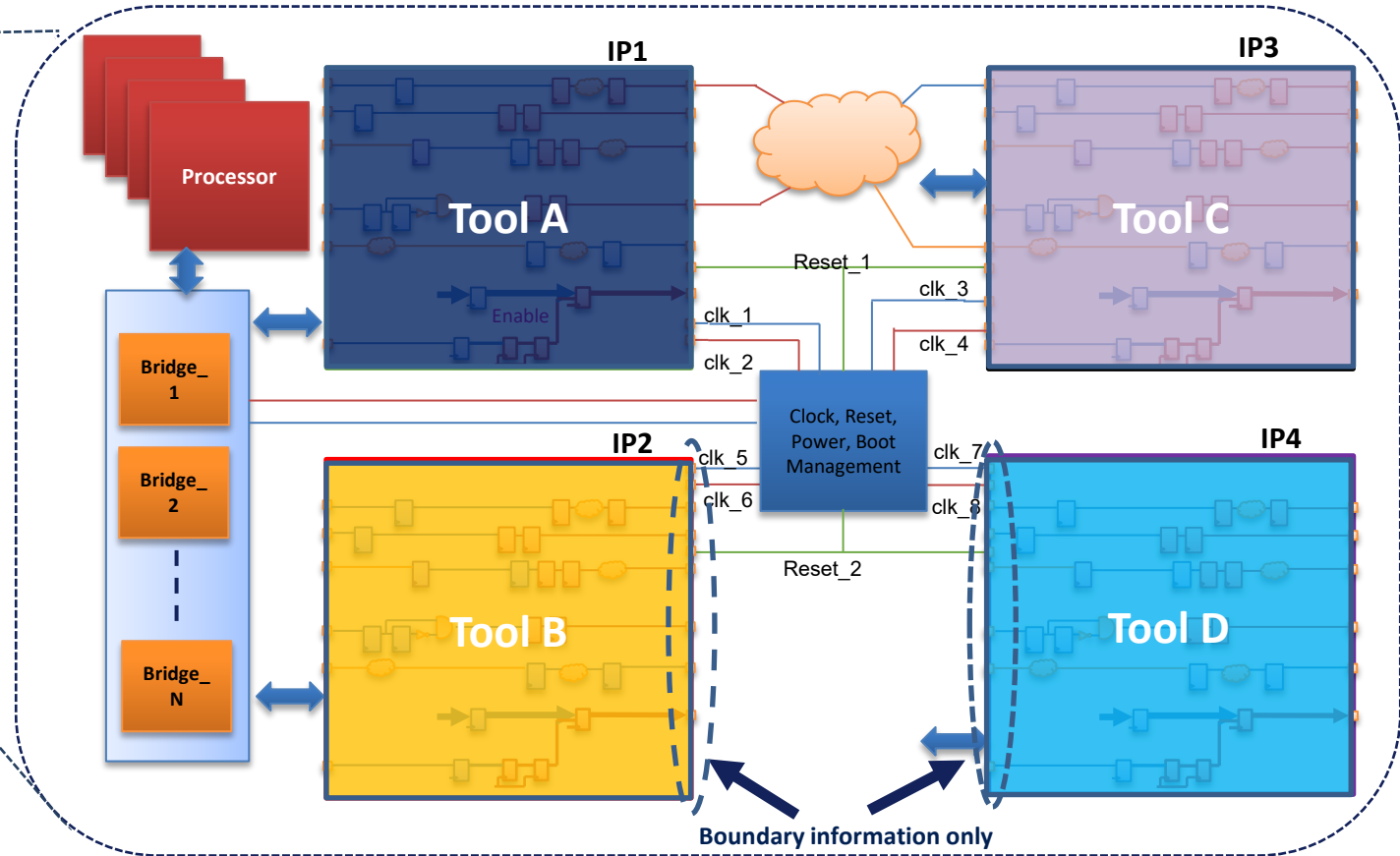
- Abstraction Models enable all Boundary related CDC-RDC info required at SoC integration Checks
- Much better approach compared to Black Box



Hierarchical Model Standardization



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



2.1 Accellera CDC Working Group

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 SOC's 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 a Language Reference Manual (**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



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

147 members from 23 companies (as of Dec 03 '24)

5 active sub-groups:

Output-Collateral, Format, Assertions, Testing, Training

V0.5 available for public review in April 2025

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

Agnisys	Aldec	AMD	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

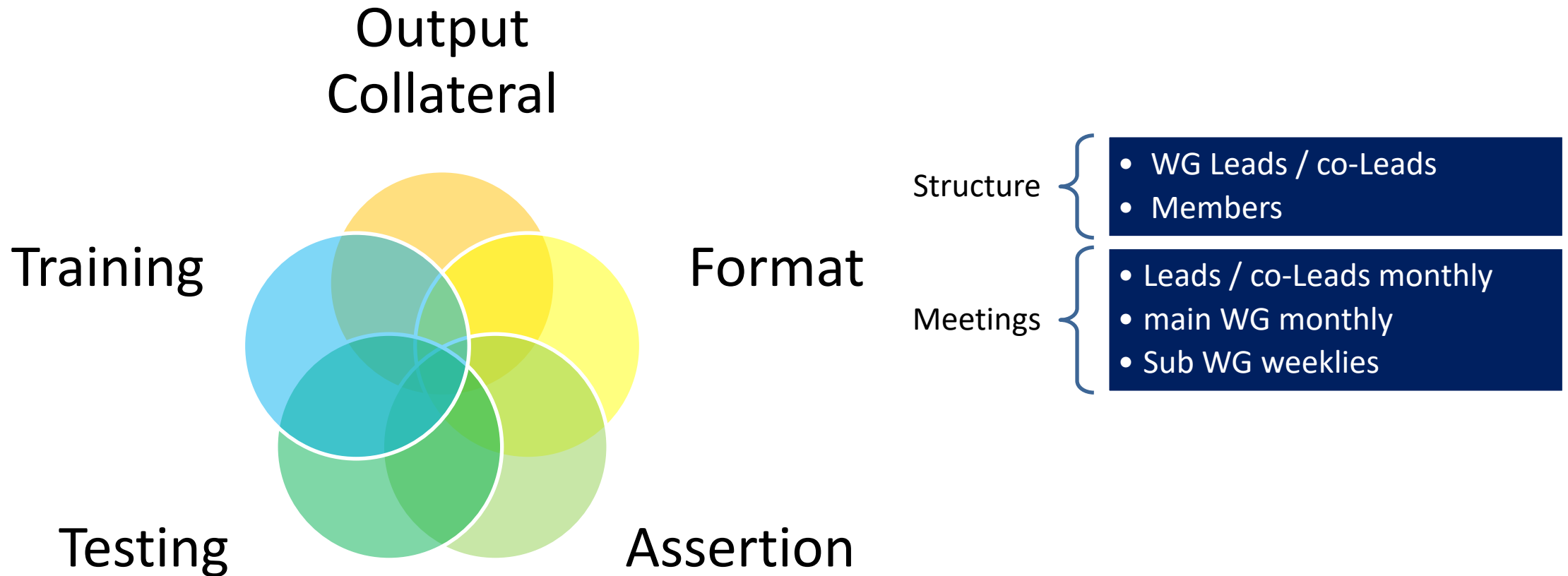
Human readable, and
machine parseable

Multi-mode/
param/instance
compliant

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

Constraints/
Assumptions can be
verified with SVAs

The Five Sub Working Groups



2.2 Testing Subgroup

Testing Subgroup Mission

- **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.

Testing Subgroup Mission

- **Work Details**

1. Review small examples provided in the latest LRM
2. Create small RTL test cases based on the examples
 - By the Subgroup members
3. Analyze the RTL test cases using various EDA tools
4. Make the test cases available for EDA vendors as first level QA testing

2.3 Output Collateral Subgroup

Output Collateral Subgroup

Address most
industry
standard
interfaces

Identify limitations
and extensions for
the attributes

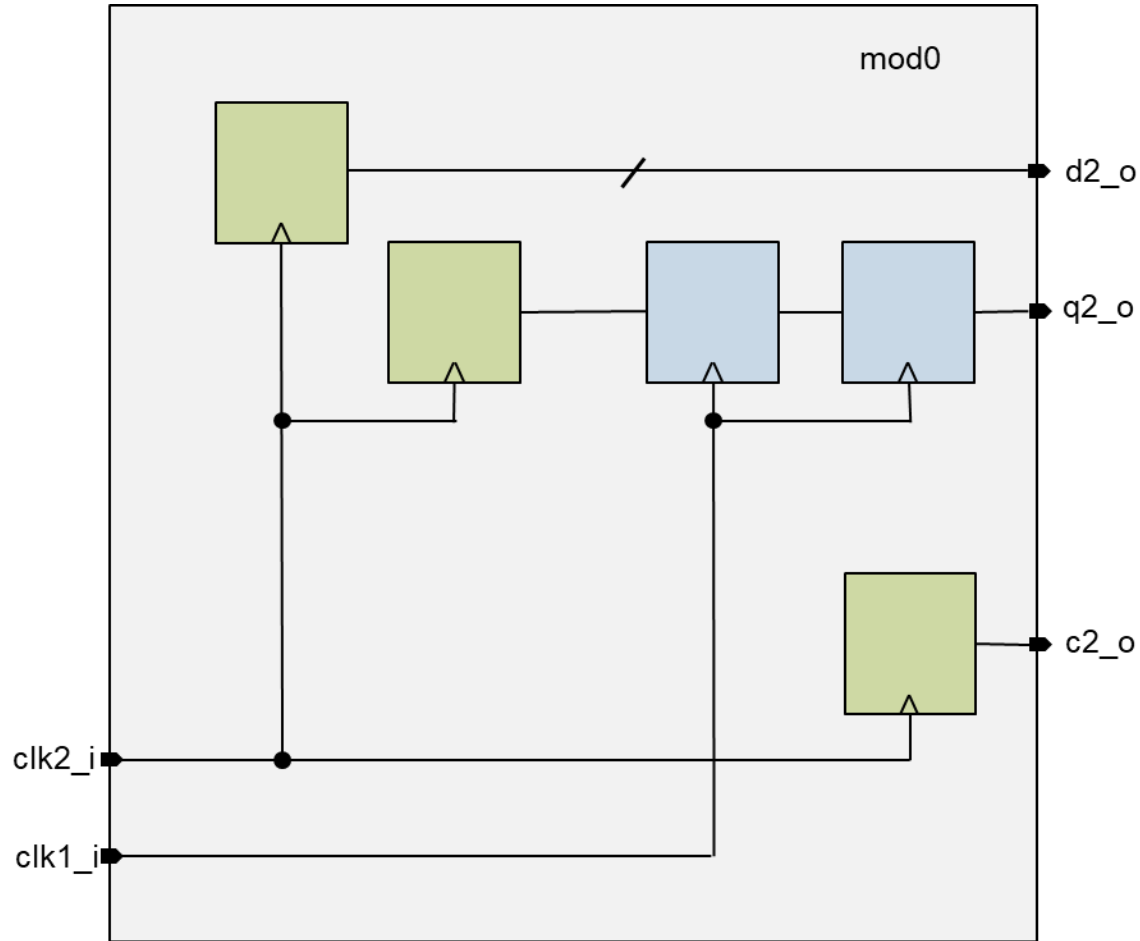
Attributes for
CDC/RDC block
model



Attributes Table in LRM v0.5

- Attributes split into domains
- Indicated by the color

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_data_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_data_from_reset	; separated list	{reset-names}	Optional
port	rdc_data_to_reset	; separated list	{reset-names}	Optional
port	rdc_data_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

Port Attribute Modelling



 clk1_i clock domain
 clk2_i clock domain

example in Tcl-format for output interface

```
cdc_set_module mod0
```

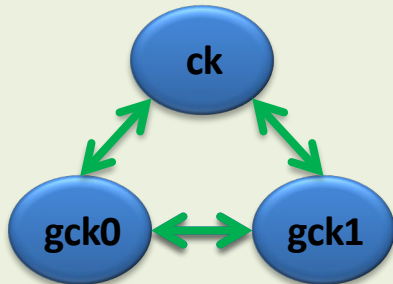
```
cdc_set_port d2_o
    -type          data
    -direction     output
    -associated_from_clocks clk2_i
    -cdc_control    q2_o
```

```
cdc_set_port q2_o
    -type          cdc_control
    -direction     output
    -cdc_data_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
```

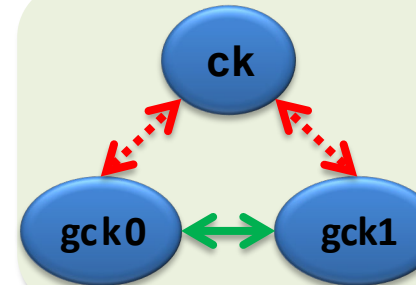
cdc_set_clock_group (Tcl format)

1 domain



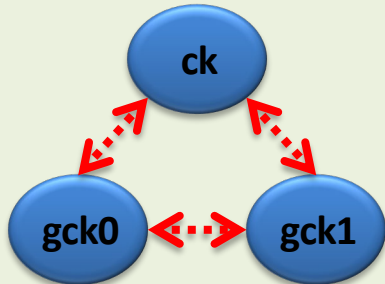
```
cdc_set_clock_group -name common_domain -clocks {ck gck0 gck1}
```

2 domains



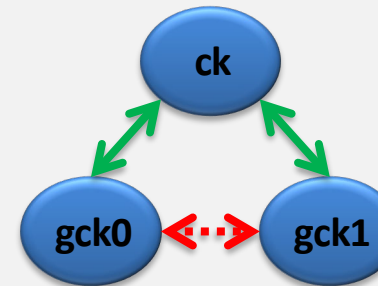
```
cdc_set_clock_group -name small_domain -clocks {ck}  
cdc_set_clock_group -name large_domain -clocks {gck0 gck1}
```

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



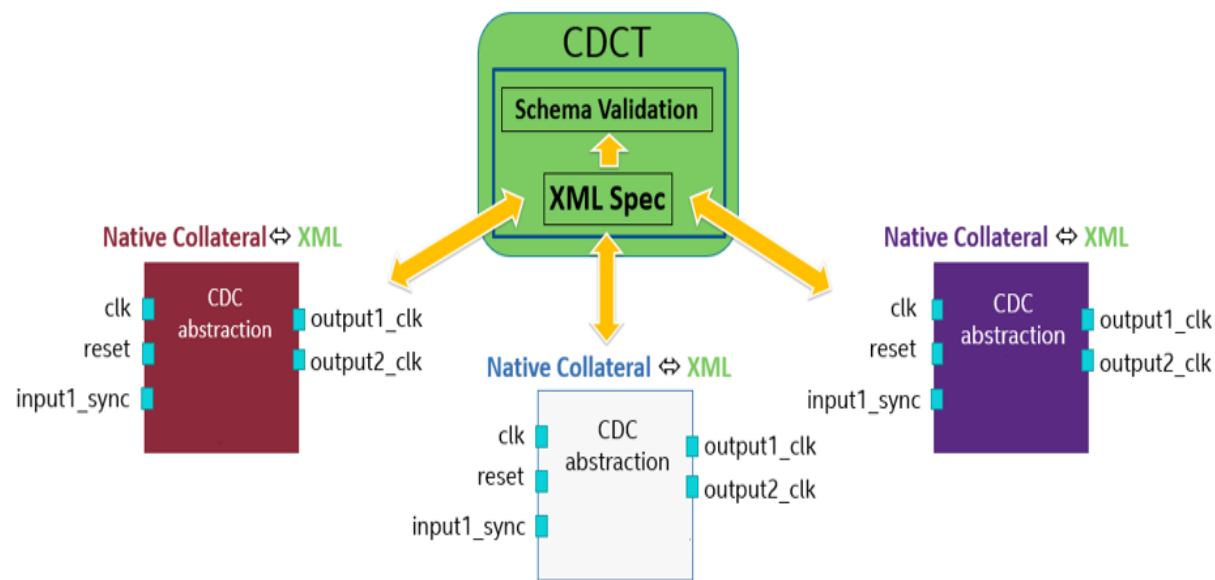
```
cdc_set_clock_group -name left_branch -clocks {ck gck0}  
cdc_set_clock_group -name right_branch -clocks {ck gck1}
```

compatibility sets:
common members
allowed

2.4 Format Subgroup

Format Subgroup Mission

- **Goal:**
 - Determine exact format for domain specific language to capture the required attributes/data from input/output collaterals
- **Methodology:**
 - Evaluated different options like Excel, IP-XACT, TCL, JSON, etc.
 - Experimented with populating the formats to ascertain the ability to meet the requirement
 - Determined pros and cons for each option of format
 - The format should support easy translation to the formats of different EDA tools and vice-versa
- **Outcome:** CDC Workgroup voted for a combination of IP-XACT and TCL as the chosen format for CDC

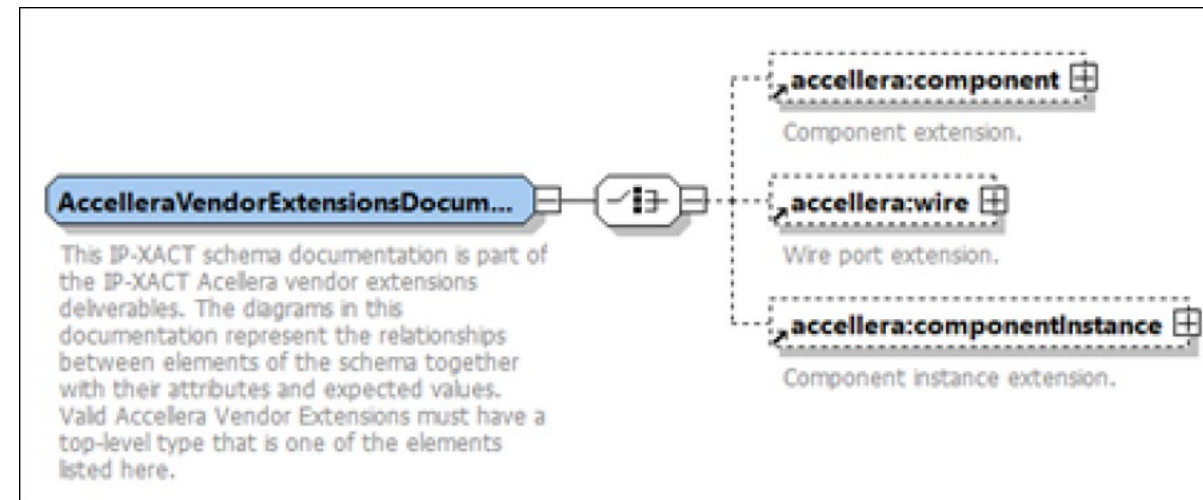


Format features

- Describing IP can be easily done by IP-XACT
 - IP-XACT is industry standard for IP definition and packaging
 - Good for standardizing the structure of CDC specification
 - Use models of IP and Product companies
 - to verify and produce collateral
 - to generate the standard format for SoCs that use a different tool
- 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
 - 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

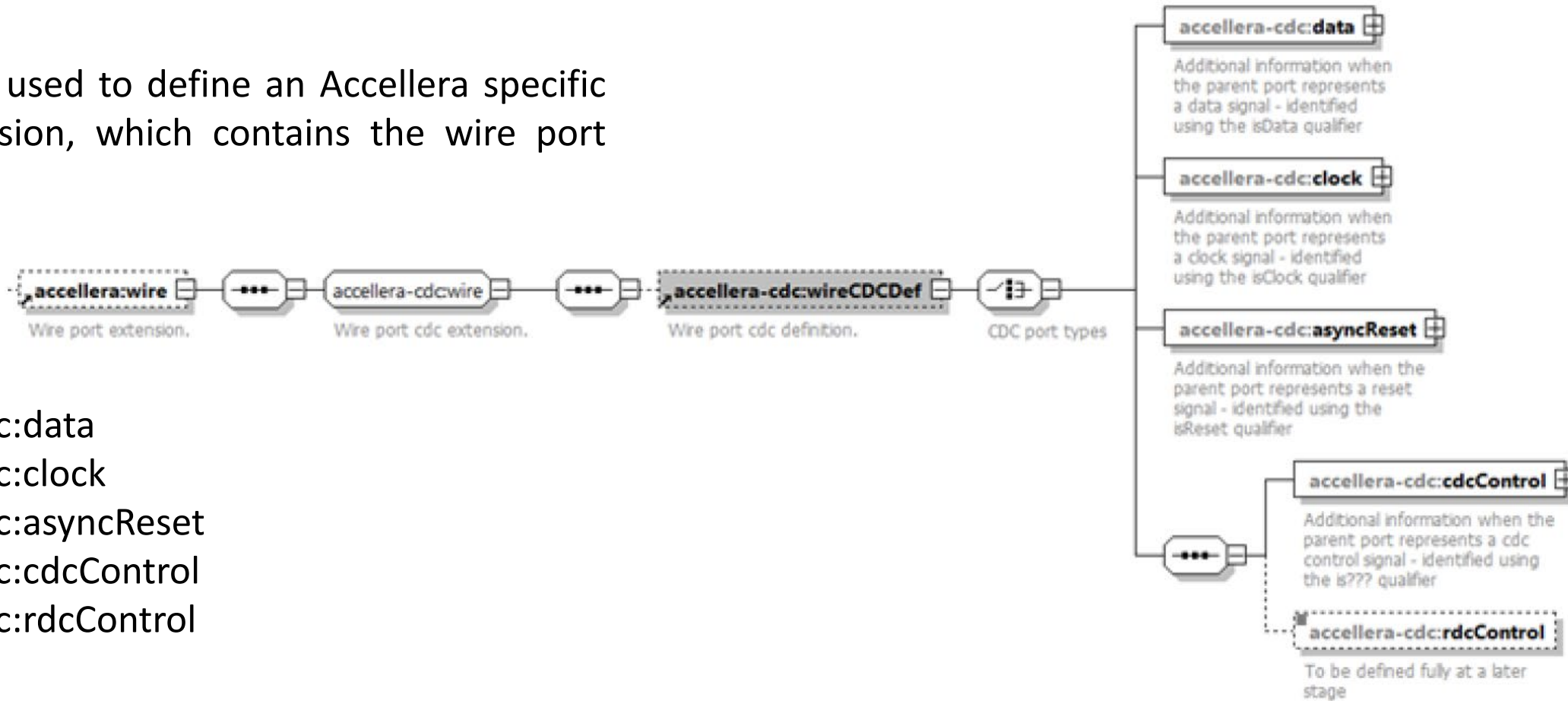
IP-XACT Format for CDC (1/4)

- The IP-XACT standard already captures some attributes related to clock and reset
- IP-XACT allows extending the standard using vendor extensions
- Vendor extensions for CDC are defined as Accellera Vendor extensions at different elements in the design hierarchy
- Top level elements for the Accellera vendor extensions for CDC are:
 - `accellera:component`
 - `accellera:wire`
 - `accellera:componentInstance`



IP-XACT Format for CDC (2/4)

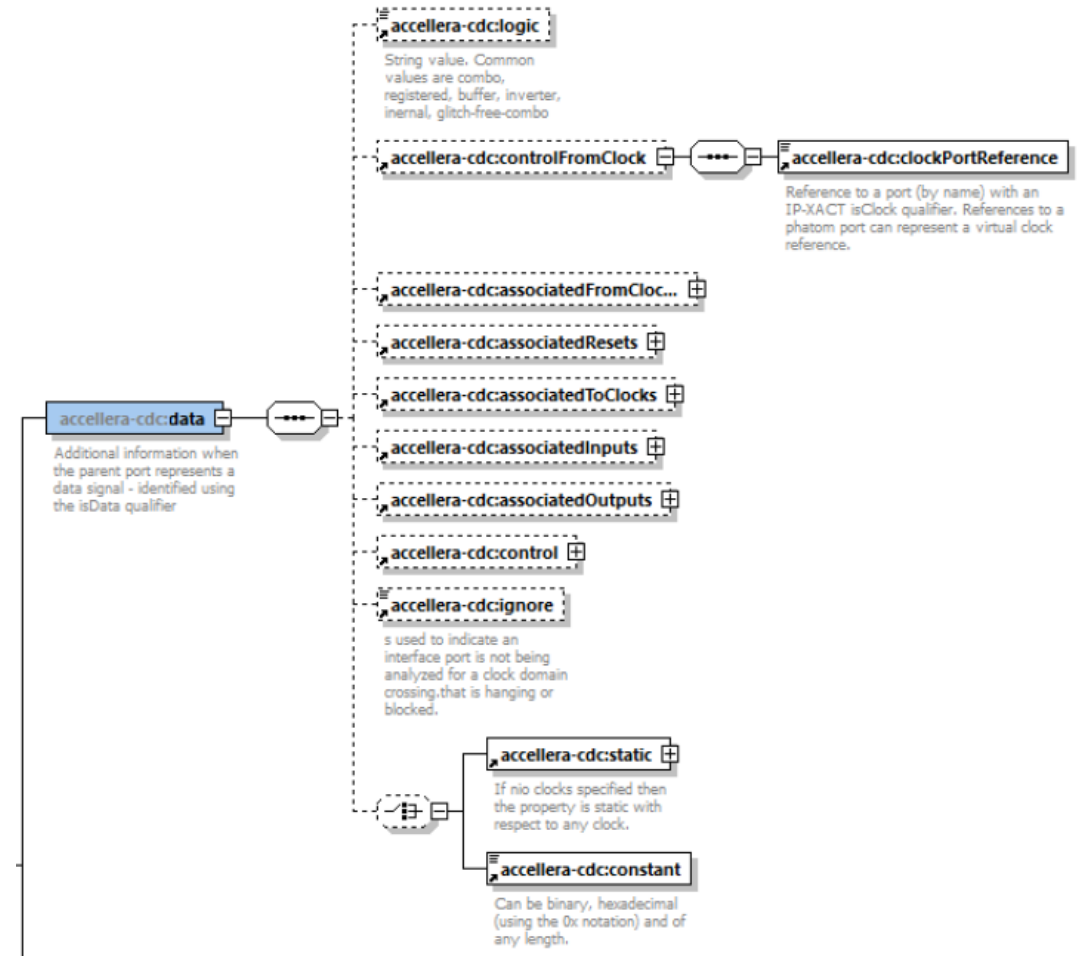
- accellera:wire is used to define an Accellera specific wire port extension, which contains the wire port CDC definition



- accellera-cdc:data
- accellera-cdc:clock
- accellera-cdc:asyncReset
- accellera-cdc:cdcControl
- accellera-cdc:rdcControl

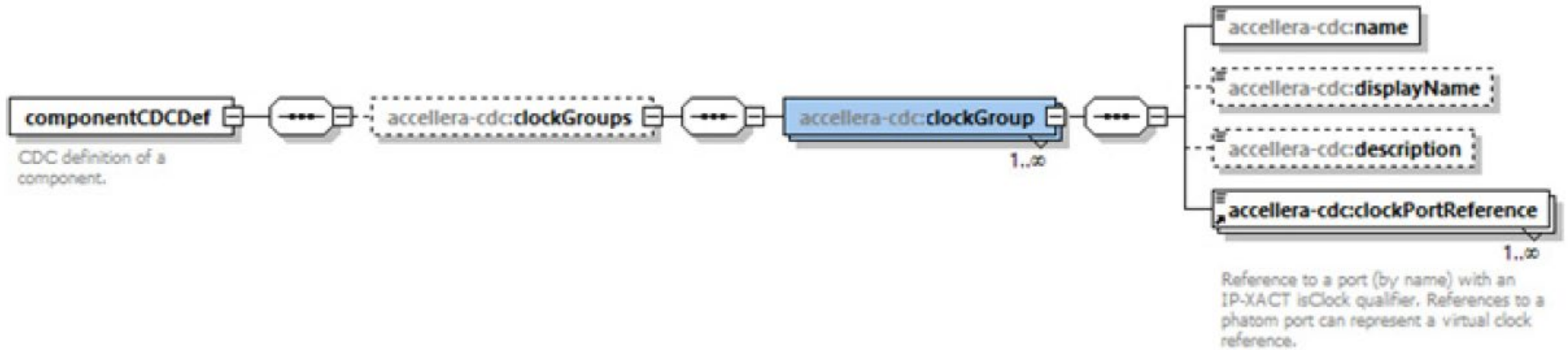
IP-XACT Format for CDC (3/4)

- When a port represents a data signal, all attributes required for CDC are captured in this extension
- Similarly, the schema is defined for asyncReset, cdcControl, and rdcControl



IP-XACT Format for CDC (4/4)

- The extension `accelera-cdc:componentCDCDef` is defined to capture the clock groups of a component using the `accelera-cdc:clockGroup` extensions
- It contains references to clock ports or phantom ports (virtual clock) in addition to name, displayName, and description.



TCL Format for CDC

- TCL Format for CDC specification is a set of API commands
- One cdc file per block/module
 - users specify the CDC attributes for ports of that module
 - users can also specify clock groups
- EDA tools should process the CDC specification in TCL format and generate corresponding IP-XACT collateral
- Similarly, there could be a requirement for generating TCL specification of CDC from IP-XACT

API Commands

- **cdc_set_module** : indicates the block/module name for which cdc specification is being provided in a file
- **cdc_set_port** : sets attributes of a port
- **cdc_set_clock_group** : set clock groups of synchronous clocks
- **cdc_set_param** : defines parameters within the scope of a module

2.5 Assertion Subgroup

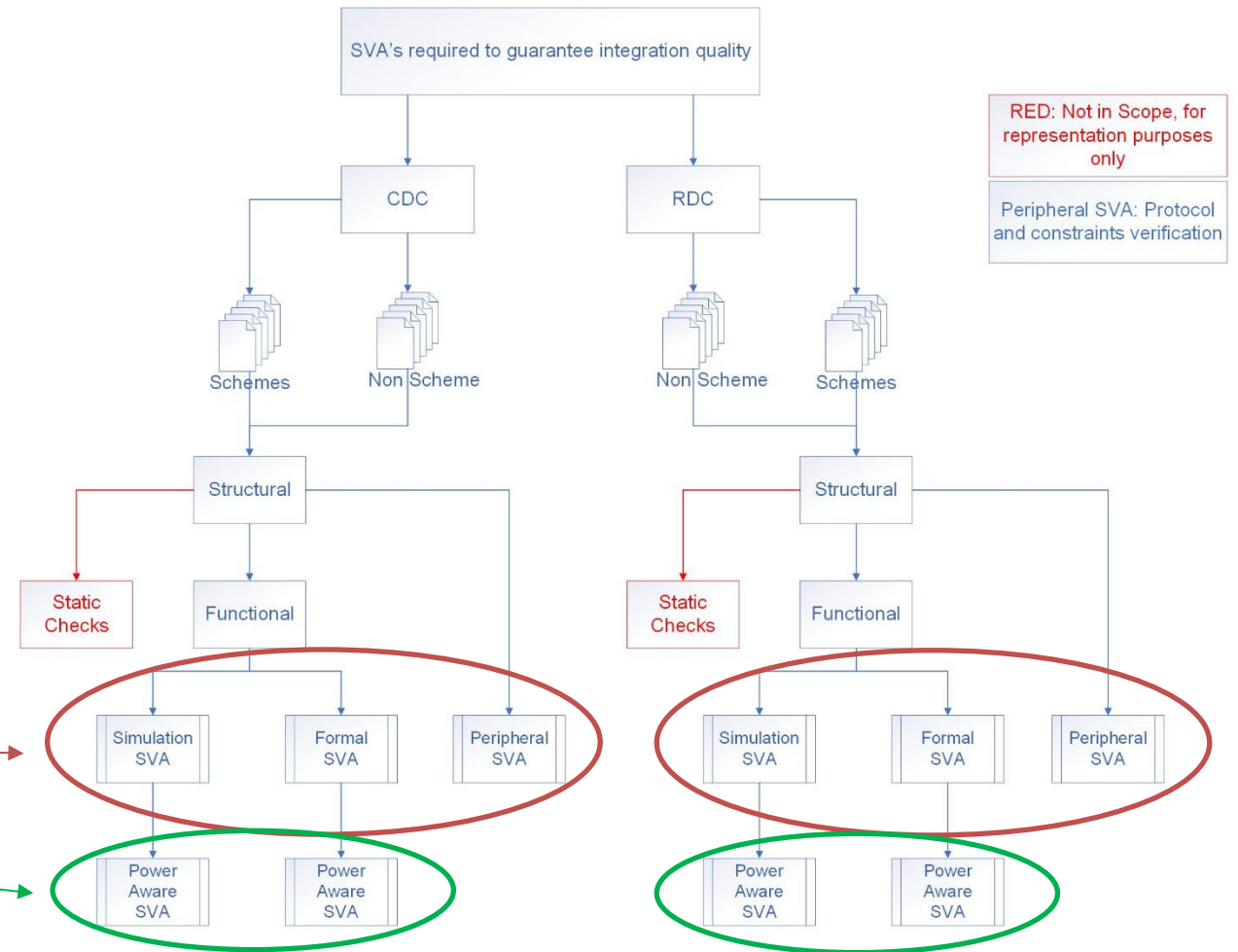
Assertion Subgroup Mission

- **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

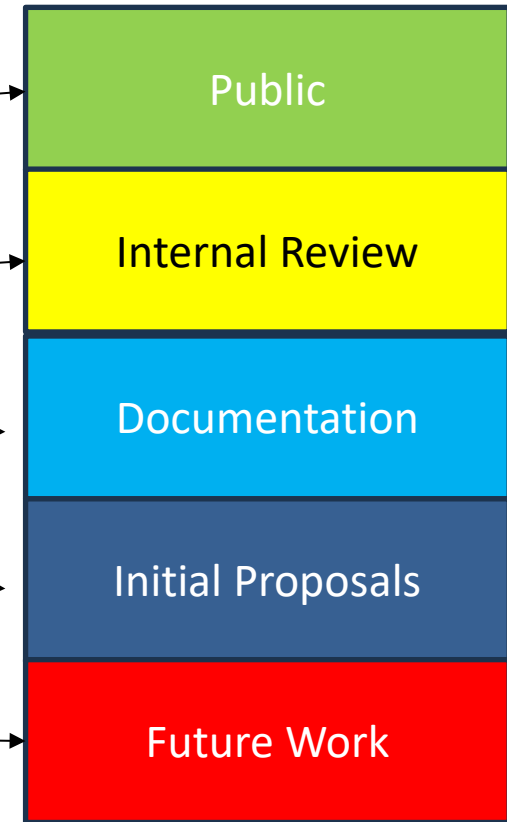
- CDC architectures are studied for possible verification strategies.
- Guidance to produce re-useable SVA for both Formal and Dynamic Verification.
- Guidance extracted from collateral.
- Guidance shall follow SVA LRM.
- Guidance shall be tool/vendor independent.
- Current Work
- Future Work



Assertion Subgroup

- LRM Clause 8 : SVA Requirements for black box CDC integrity verification

- 8.1 : Overview
- 8.2 : Sampling Edge Requirement
- 8.3 : Implementation Headroom for a Crossing
- 8.4 : Verification Clock for a Crossing
- 8.5 : Verification of Parameters
- 8.6 : Verification of Constant Signals
- 8.7 : Verification of Open Loop Crossing
- 8.8 : Verification of Closed Loop Crossing – Handshake
- 8.9 : Verification of Mutually Exclusive Buses
- 8.10 : Asynchronous FIFO
- 8.11 : Reset Domain Crossing



2.6 Training Subgroup

Training Subgroup Mission

- **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 :
 - CDC-RDC basic knowledge
 - 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**
 - Tool dependency issue
 - Necessity to create an inter operational CDC-RDC model
4. Inter operational CDC-RDC model **integration manual**

Conferences

- Accellera CDC WG work promotion through conferences

- Past/current conferences

- DVCON Europe 2024
- IP-SoC Design Reuse conference Europe 2024
- DVCON US 2025
- DVCON China 2025
- DAC 2025
- DVCON Japan 2025
- DVCON Taiwan 2025
- DVCON India 2025

- Targeted conferences (To Be Confirmed)

- DVCON Europe 2025
- IP-SoC Design Reuse conference Europe 2025
- DVCON US / China / Japan / India / Europe / Taiwan 2026
- DATE 2026
- VLSI 2026
- DAC 2026

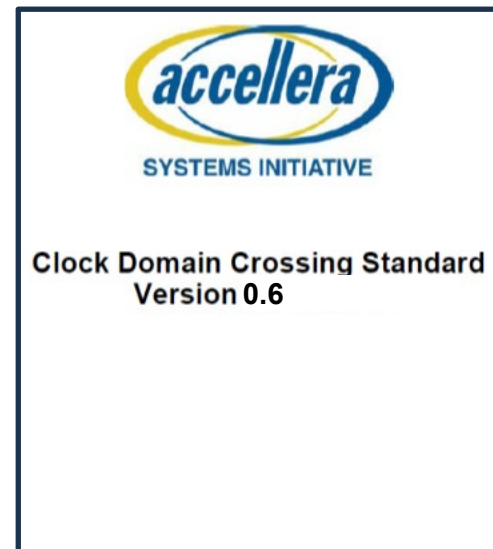


2.7 Call For Contribution

Language Reference Manual (LRM)

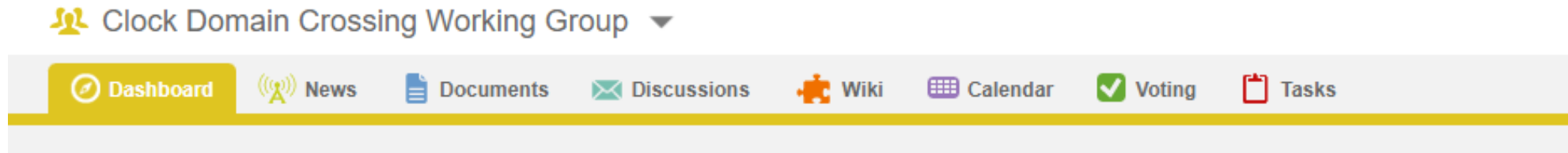
Version 0.5 of the CDC LRM has been released for public review on 14 April 2025.

- [CDC 0.5 Public Review Draft 2025.04.14](#) Rev 0.1
- Download the CDC Draft Standard 0.5: <https://bit.ly/3tdwMpe>
- Forum: <https://bit.ly/42vYdb4>



Call for Contribution !

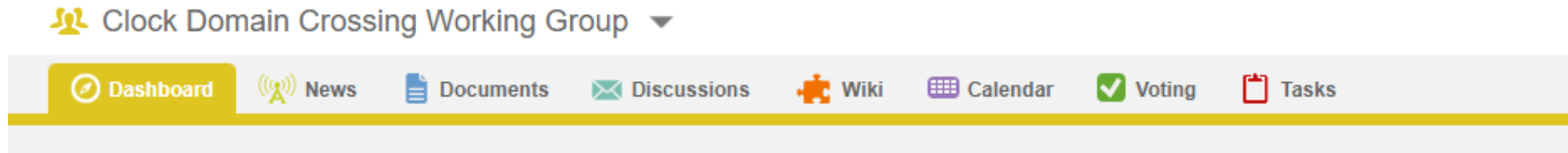
Accellera CDC Working Group



<https://workspace.accelera.org/wg/CDC>

Call for Contribution !

Accellera CDC Working Group



<https://workspace.accelera.org/wg/CDC>

**Non-Accellera members can join and
provide feedback on the standard:**

<https://www.accelera.org/community>

Questions?