



# Standardizing CDC and RDC abstract models

Accellera CDC Working Group

Authors

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

Chetan  
CHOPPALI SUDARSHAN

Farhad  
AHMED

Iredamola  
OLOPADE

Sean  
O'DONOHUE

Bill  
GASCOYNE

Kranthi  
PAMARTHI

Anupam  
BAKSHI





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



SIEMENS



SYNOPSYS



RENESAS



# Presenter Introduction

- Anupam Bakshi
  - leads Agnisys for register automation & iSpec.ai for automating assertion generation

# Agenda (DVCON Japan)

	Topic	Slide contributors
#1	CDC-RDC (35min)	
	CDC-RDC Basic Knowledge	Bill Gascoyne, Blue Pearl; Jan Hayek, Robert Bosch; Chetan Choppali Sudarshan, Marvell; Ashish Soni, ST Micro; Jebin Mohandas, Intel
	CDC Verification on RTL	Ping Yeung, Nvidia
	CDC Assertion-Based Verification	Kranthi Pamarthi, Renesas Electronics
	CDC-RDC Hierarchical Flow	Farhad Ahmed, Siemens EDA
#2	Accellera CDC Working Group (10min)	
	Standard	Iredamola Olopade, Intel
	Output	Sean O'Donohue, Synopsys
	Format	Devender Khari, Agnisisys
	Assertion	Kranthi Pamarthi, Renesas Electronics
	Testing	Farhad Ahmed, Siemens EDA
	Training	JC Brignone, Diana Kalel, ST Micro





# 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

# 1. 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

## 2. 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!





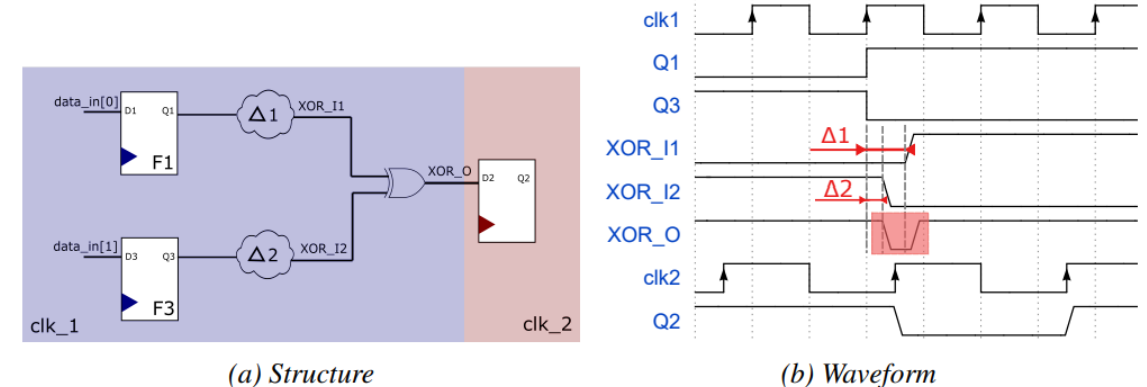
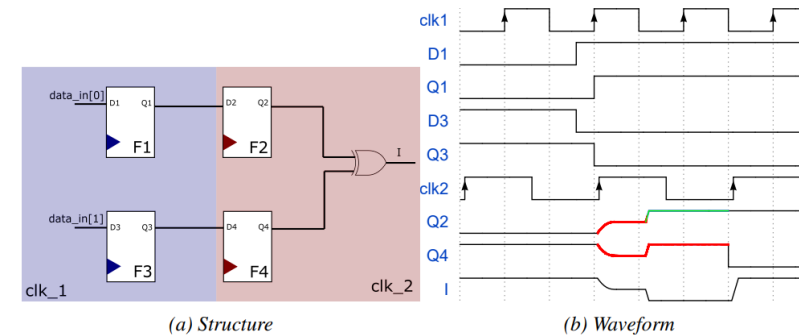
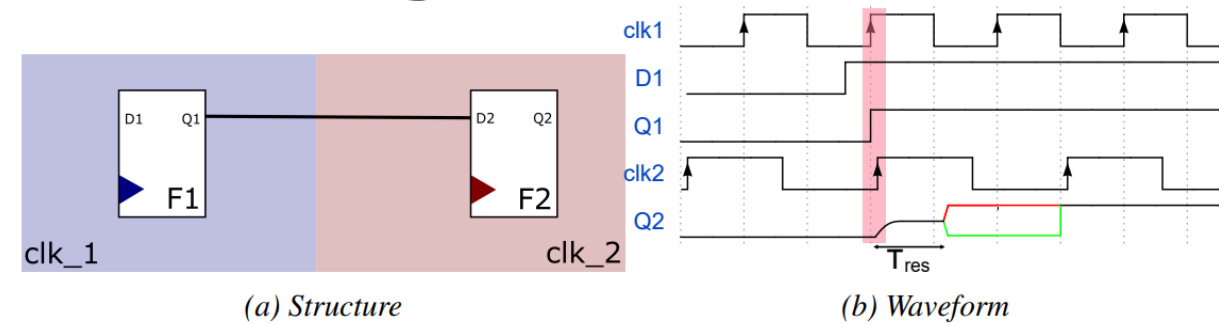
# 3. Why do CDCs need fixing?

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

The coin on its edge

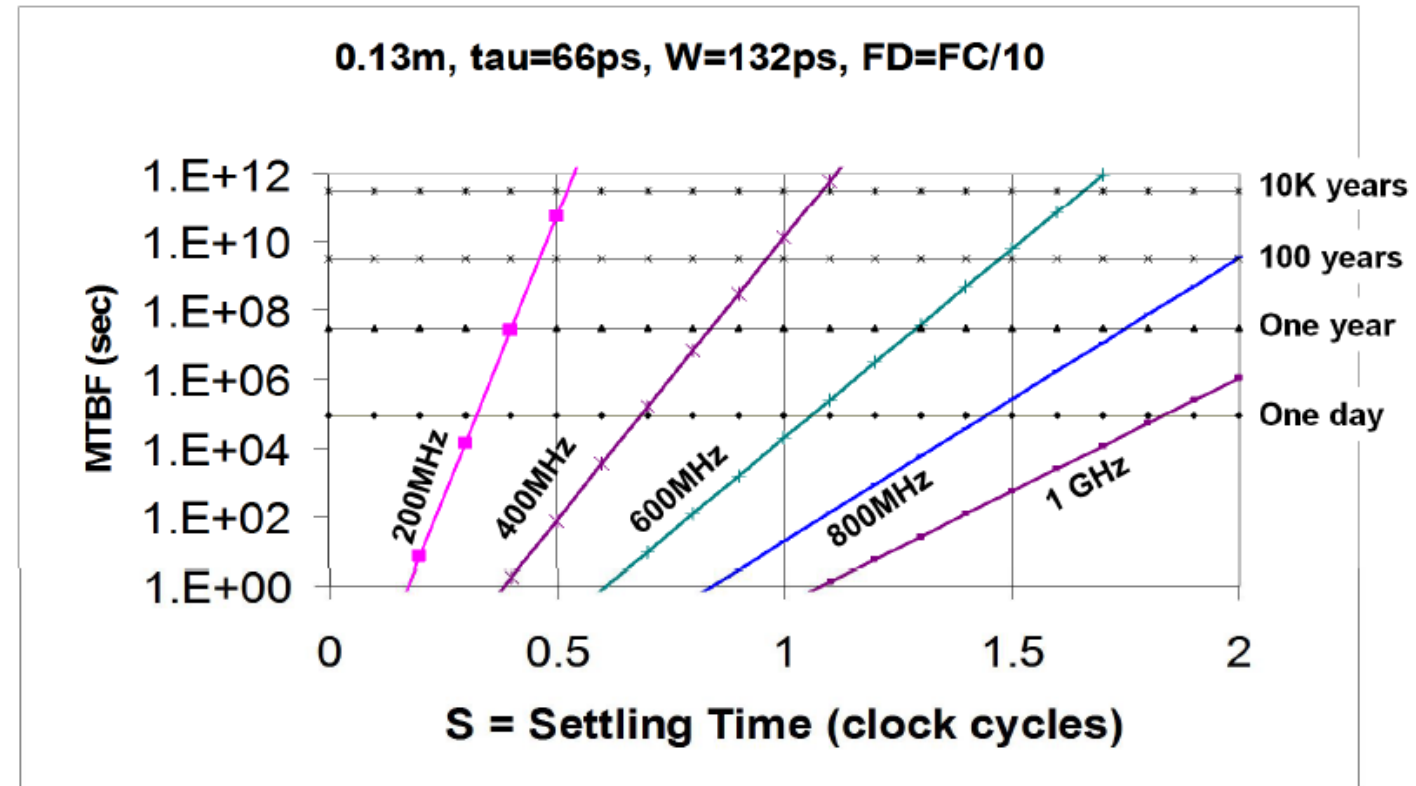
- Loss of Data Coherency
  - The indeterminate state settles to a random value

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



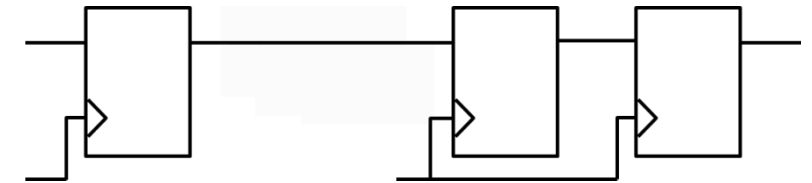
# 4. MTBF Mean Time Between Failures

- given metastability at  $t = 0$ ,  
probability of metastability at  $t > 0 \rightarrow e^{-t/\tau}$
- failure: still metastable at next clock edge
  - failure =  $p(\text{enter m.s.}) \times p(\text{still m.s. after } T_s)$   
 $= T_W/T_C \times e^{-T_s/\tau}$
  - rate(failure)  
 $= \text{rate}(\text{enter m.s.}) \times p(\text{still m.s. after } T_s)$   
 $= T_W \times f_C \times f_D \times e^{-T_s/\tau}$
- MTBF =  $1 / \text{rate}(\text{failure})$   $MTBF = \frac{e^{T_s/\tau}}{T_W f_C f_D}$



# 4. Synchronization

## 4.1 Multi-flop synchronizers



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

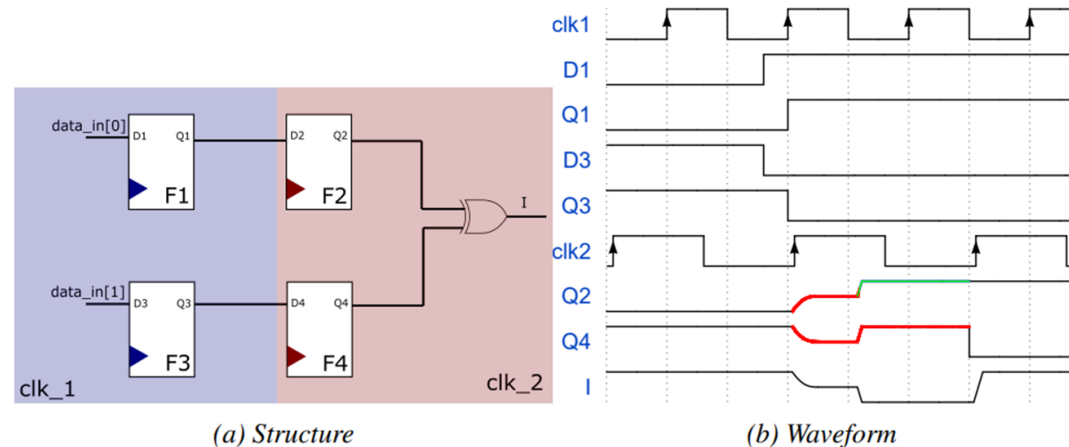
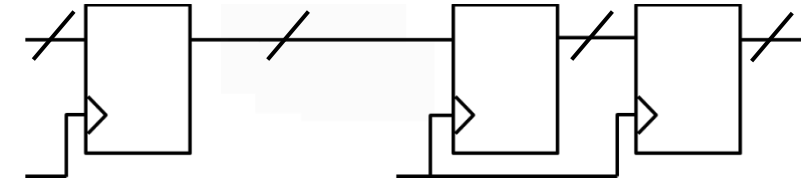
$$MTBF(t_r) = \frac{e^{\frac{t_r}{\tau}}}{T_0 * F_c * F_d} * \frac{e^{\frac{T_r}{\tau}}}{T_0 * F_c}$$

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

# 4. Synchronization

## 4.1 Multi-flop synchronizers

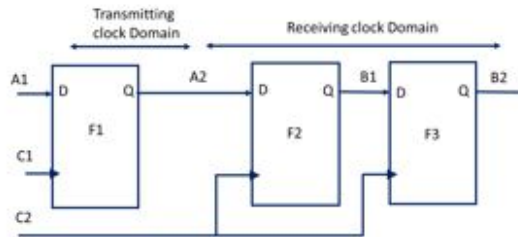
- With multiple bits, metastability is still addressed but data coherency is a problem!
  - If multiple bits change on the same cycle, the result of each bit is random
  - This synchronization works only if the data is “arav” (only one bit changes)



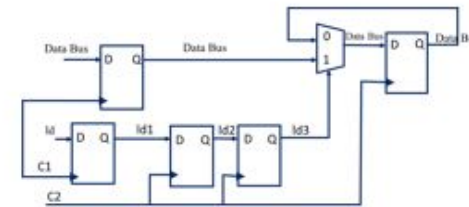


## 4. Structural CDC

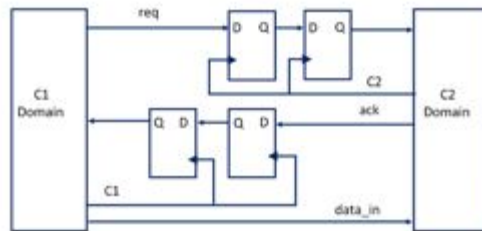
# Commonly Used Synchronization Schemes



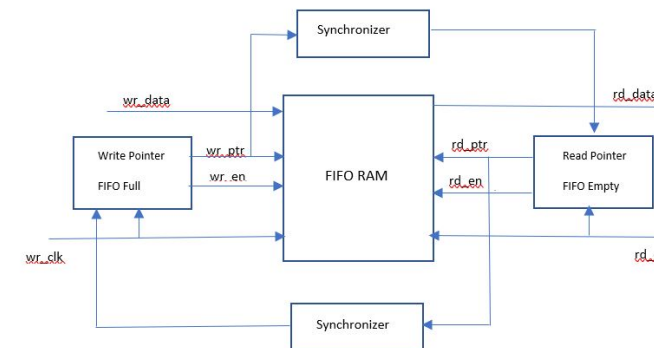
## Double-FF synchronizer



MUX synchronizer



## Handshake synchronizer



## FIFO synchronizer

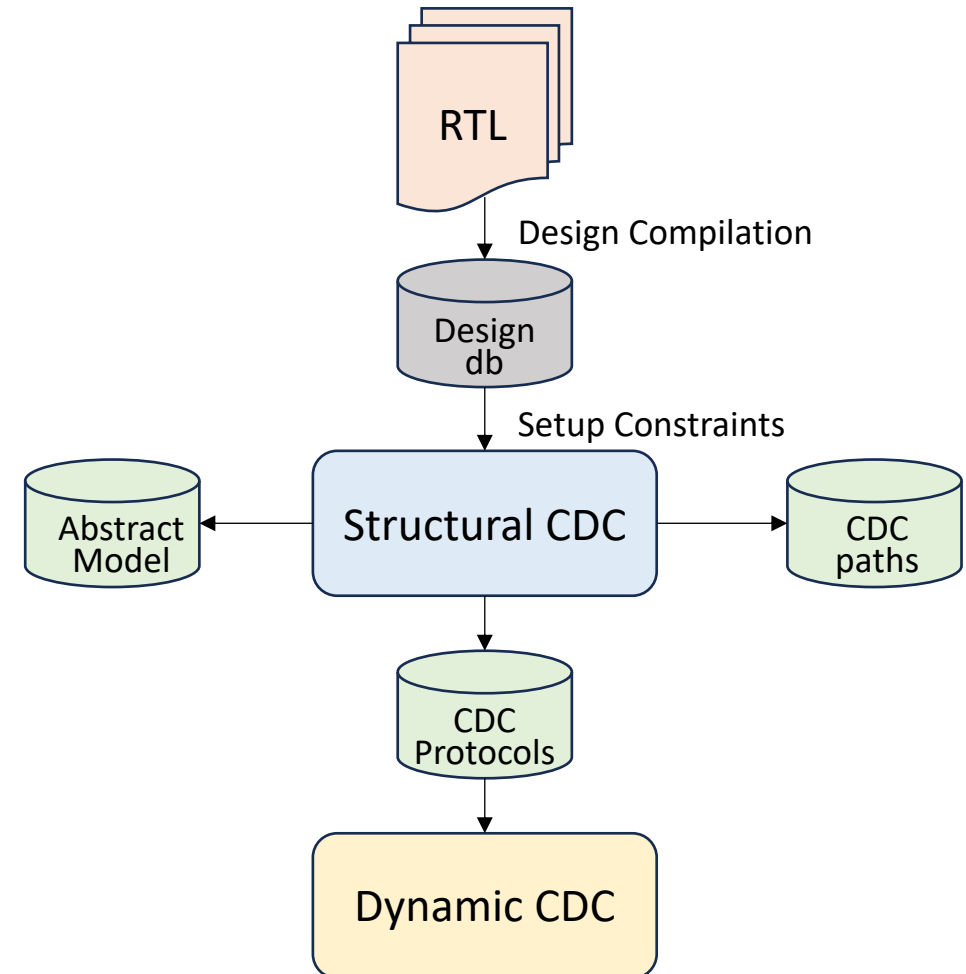


# CDC verification on RTL

- Setup generation
- CDC setup check
- CDC structural checks
- CDC assertions-based verification
- Hierarchical CDC structural verification

# 1. CDC Verification flow

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



## 2. 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  $\neq$  Clock groups for CDC analysis  
Signal paths waived for time analysis  $\neq$  Signal paths waived for CDC analysis

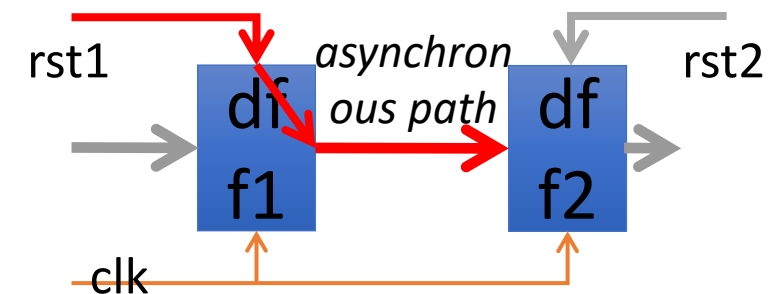
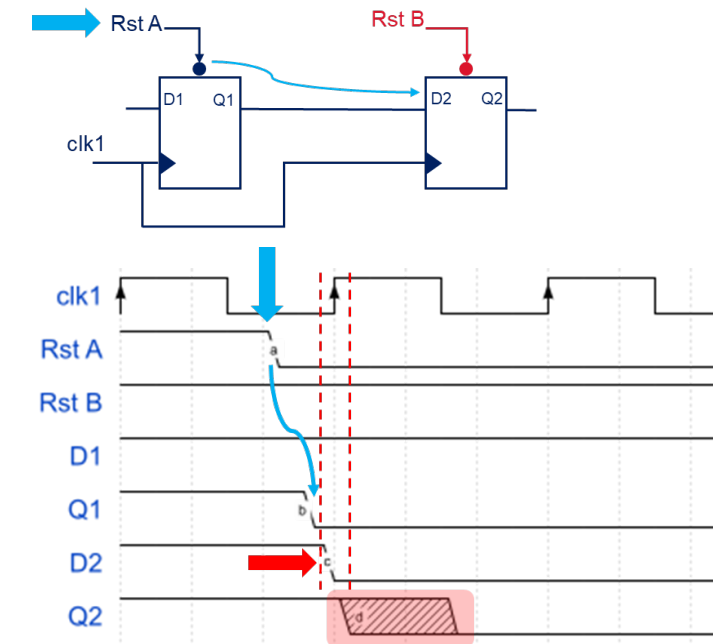


# 3. Dynamic CDC Verification

- Dynamic verification is to ensure
  - structural CDC check is done with the proper constraints and assumptions
  - the identified CDC paths follow the protocols 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

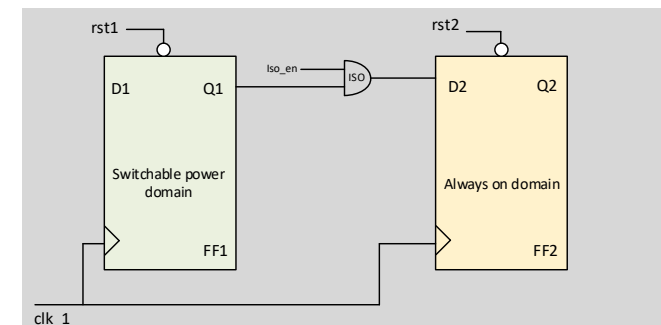
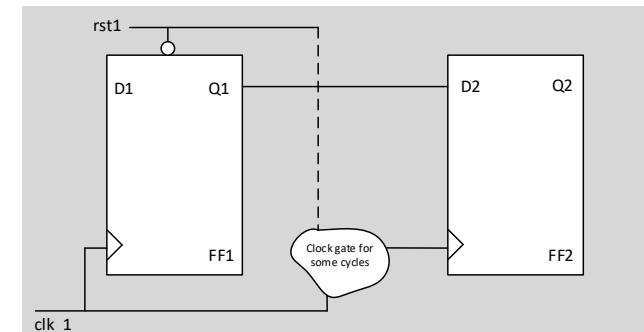
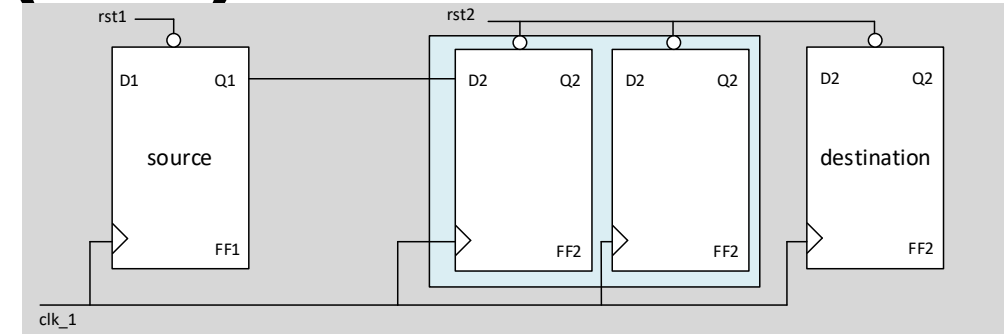
# 4. Reset Analysis

- DFF models also have **recovery time**
  - Time between asynchronous Set/Reset release and clock when data and output are different
  - Static analysis is sufficient to determine synchronization point
- 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



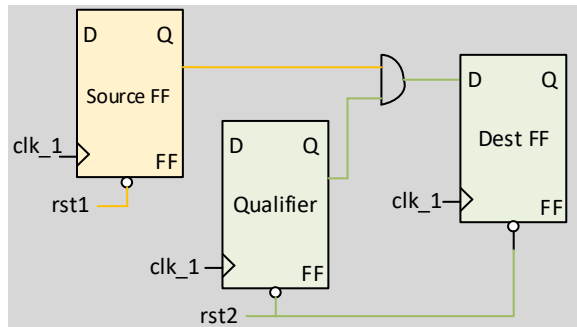
# 5. Resolving RDC issues (1/2)

- Adding synchronizer in the destination domain
- Reset ordering of different resets in the design
- Reset-less paths: Clock gate the downstream flops for some cycles when rst1 asserts to avoid glitch capture
- Multipower domain designs : reset ordering, isolation en asserted

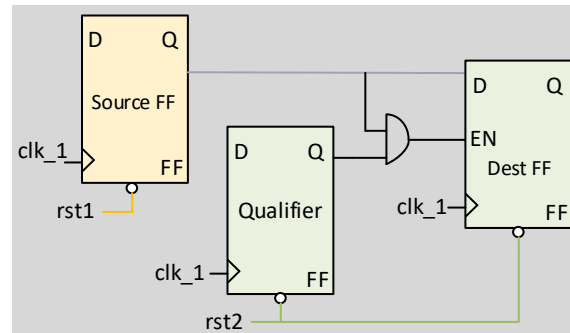


# 5. Resolving RDC issues (2/2)

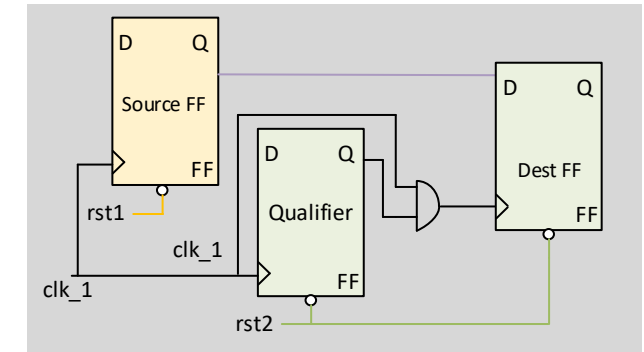
- Using Qualifiers
  - Data Qualifier
  - Enable Qualifier
  - Clock Qualifier



Data Qualifier



Enable Qualifier



Clock Qualifier



# Structural CDC/RDC - Limitations

## 1- Constraints based static checks

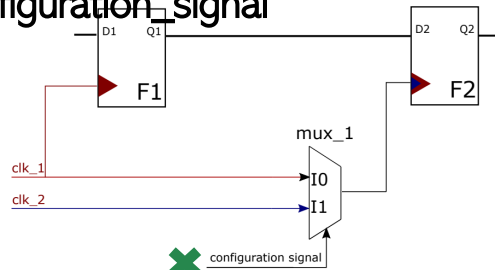
- Affect the results of the structural checks
- Are taken blindly for the structural verification
  - e.g., a CDC can be bypassed if the crossing signal is pseudo-static

## 2- Rules based static checks

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

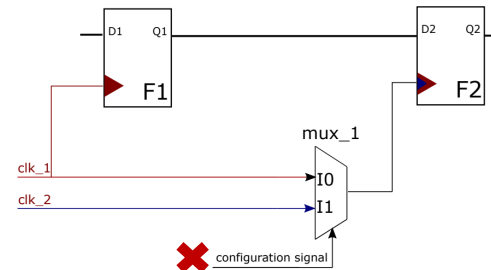
✗ set\_case\_analysis 0

configuration\_signal



Synchronous

✗ set\_case\_analysis 1 configuration\_signal



Asynchronous



# CDC Assertion based Verification

- Constraint based static checks
- Rule based static checks



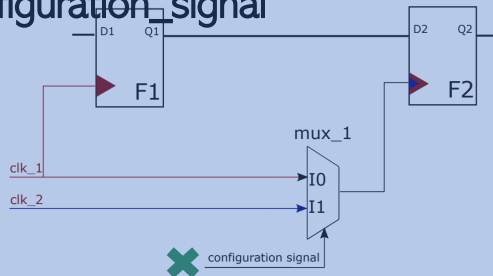
# Overcoming Limitations – Assertions Based Verification

## 1- Constraints based static checks

- Affect the results of the structural checks
- Are taken blindly for the structural verification

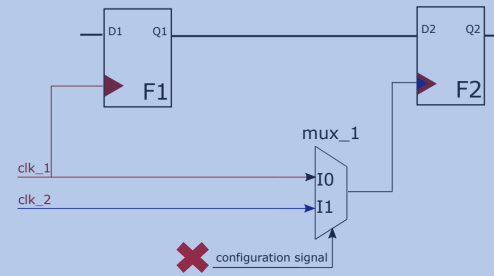
e.g. a CDC can be bypassed if the crossing signal is pseudo-static

✗ set\_base\_analysis configuration\_signal



Synchronous

✗ set\_base\_analysis configuration\_signal



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

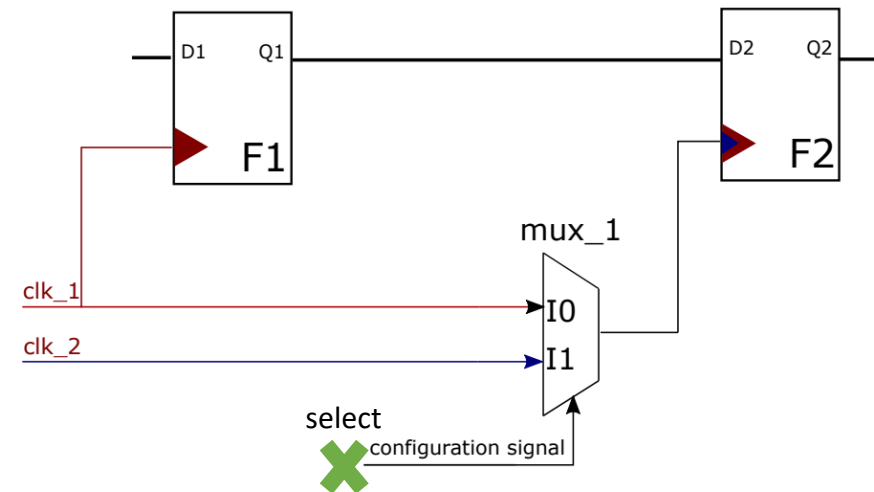
## 1- Constraints based static checks

- 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

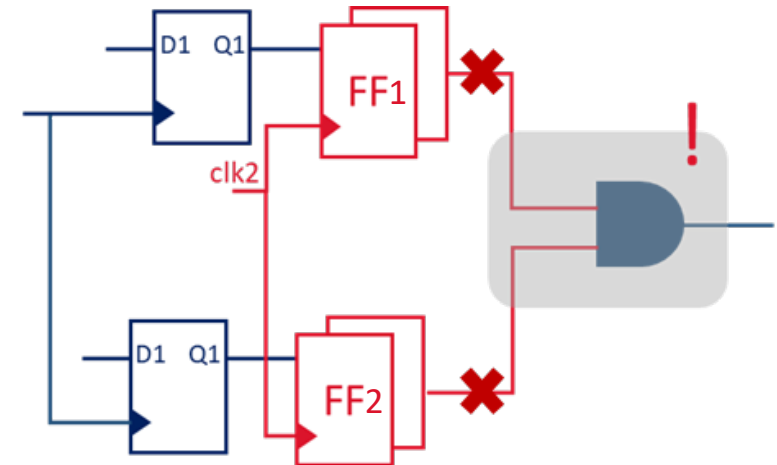
- Constraints to be double checked with the functional verification

- Mutually exclusive

```
define_exclusive –signals [set of signals names]
```



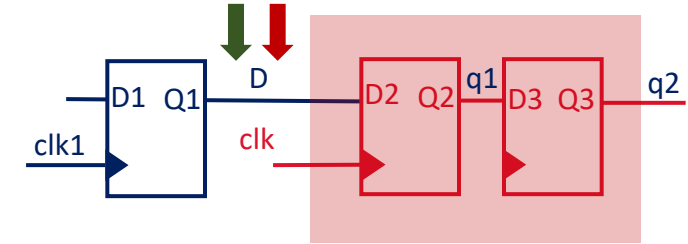
```
property mutex (data, clk);  
  @(posedge clk)  
    $onehot0(data ^ $past(data));  
endproperty
```



# Overcoming Limitations – Assertions Based Verification

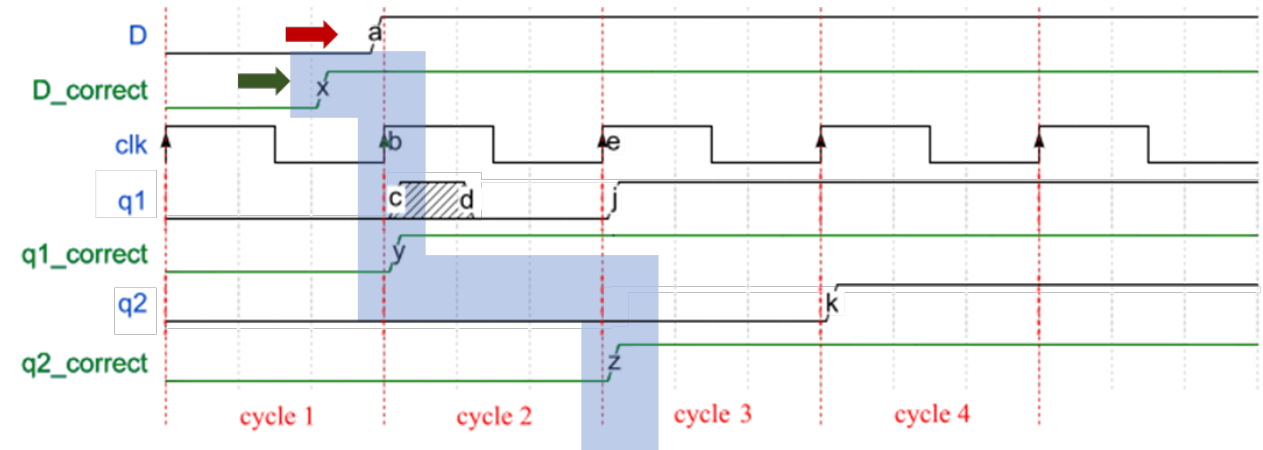
## 2- Rules based static checks

- 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

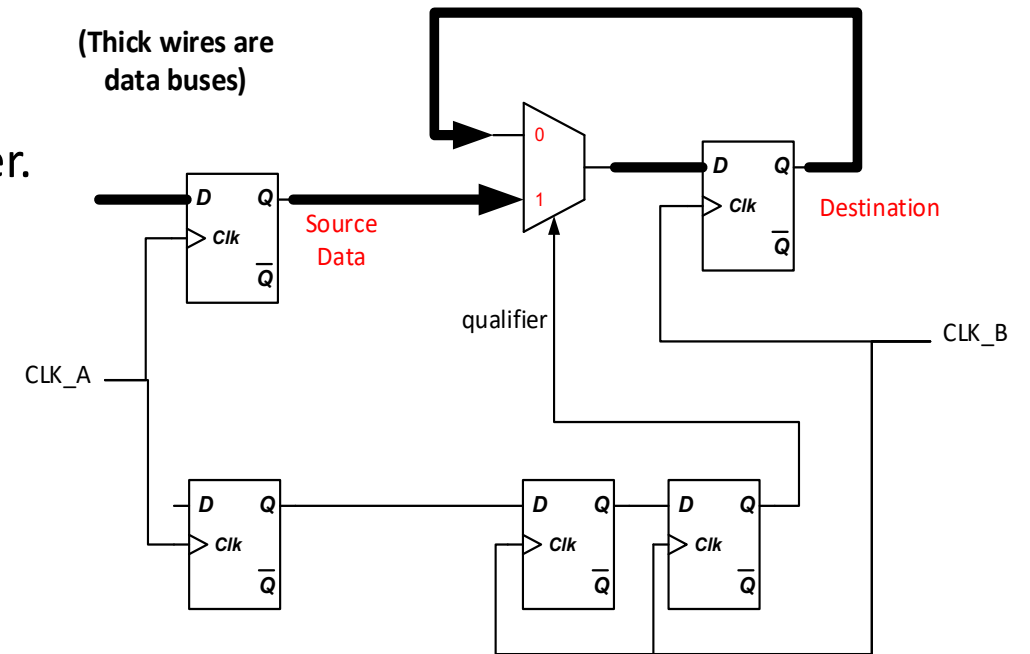
## 2- Rules based static checks

- 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.

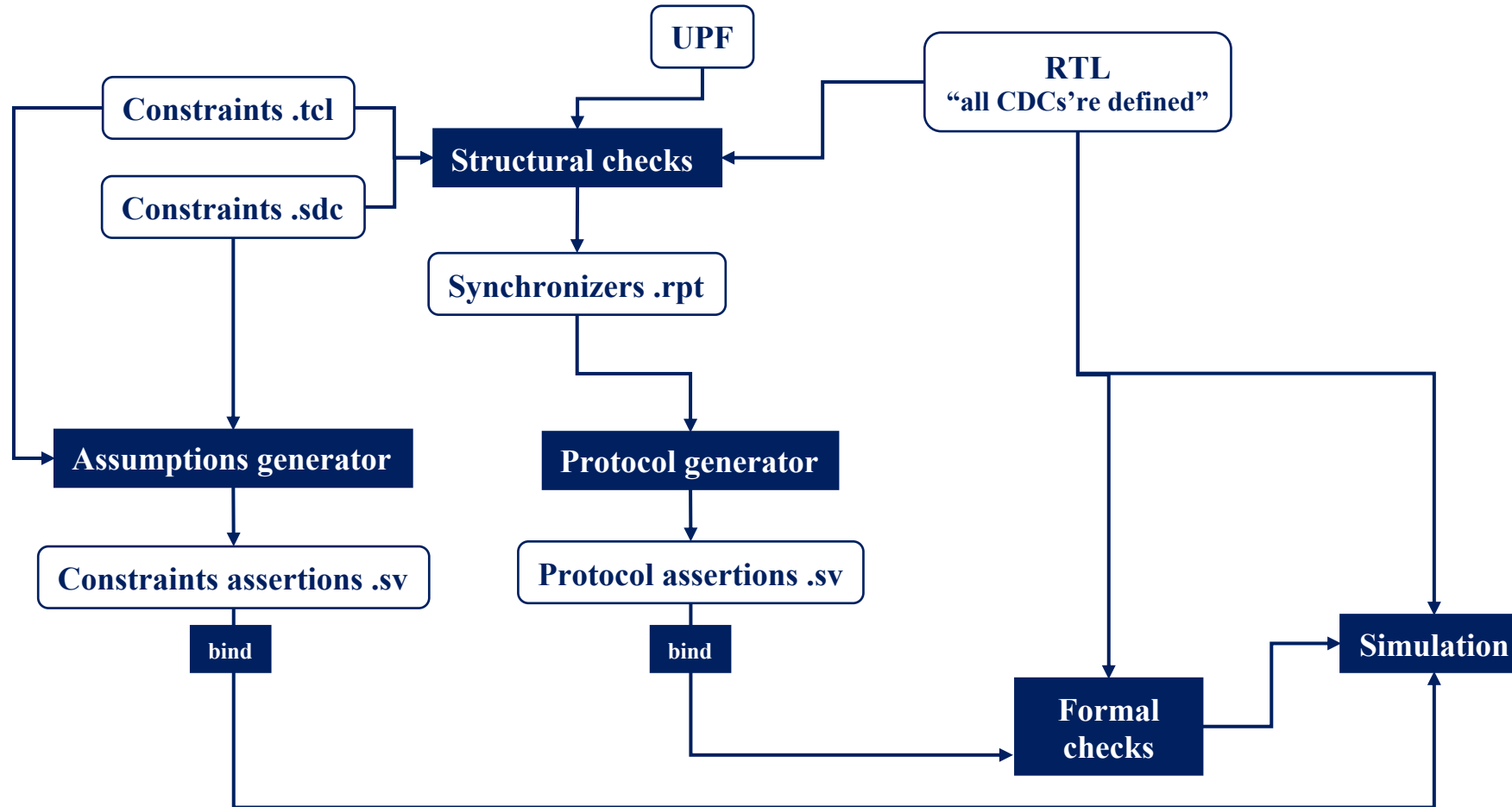
Source data must be static when CDC Control is enabling.  
 Source data can toggle when CDC Control is disabling.  
 CDC Control must be a known value

```
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

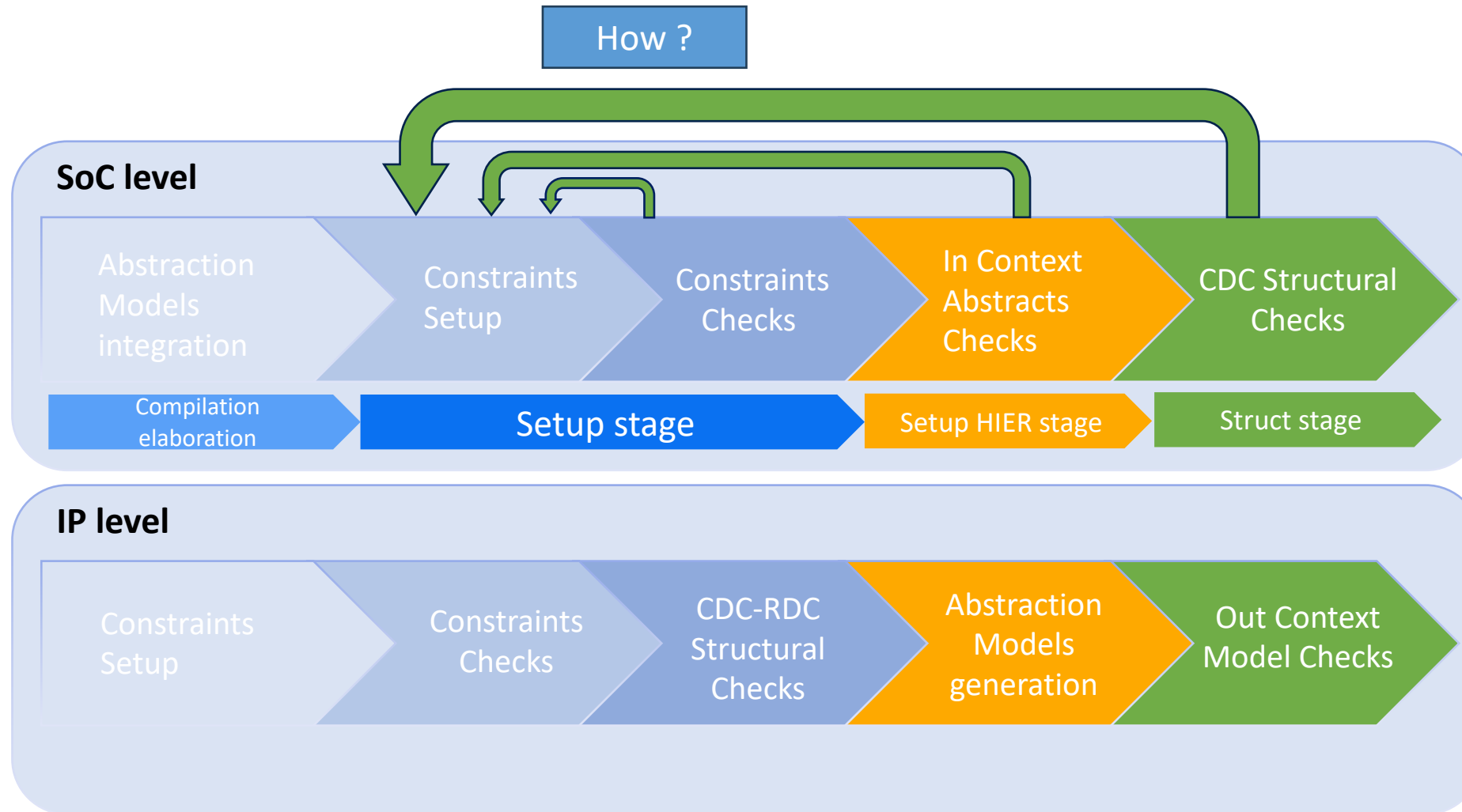




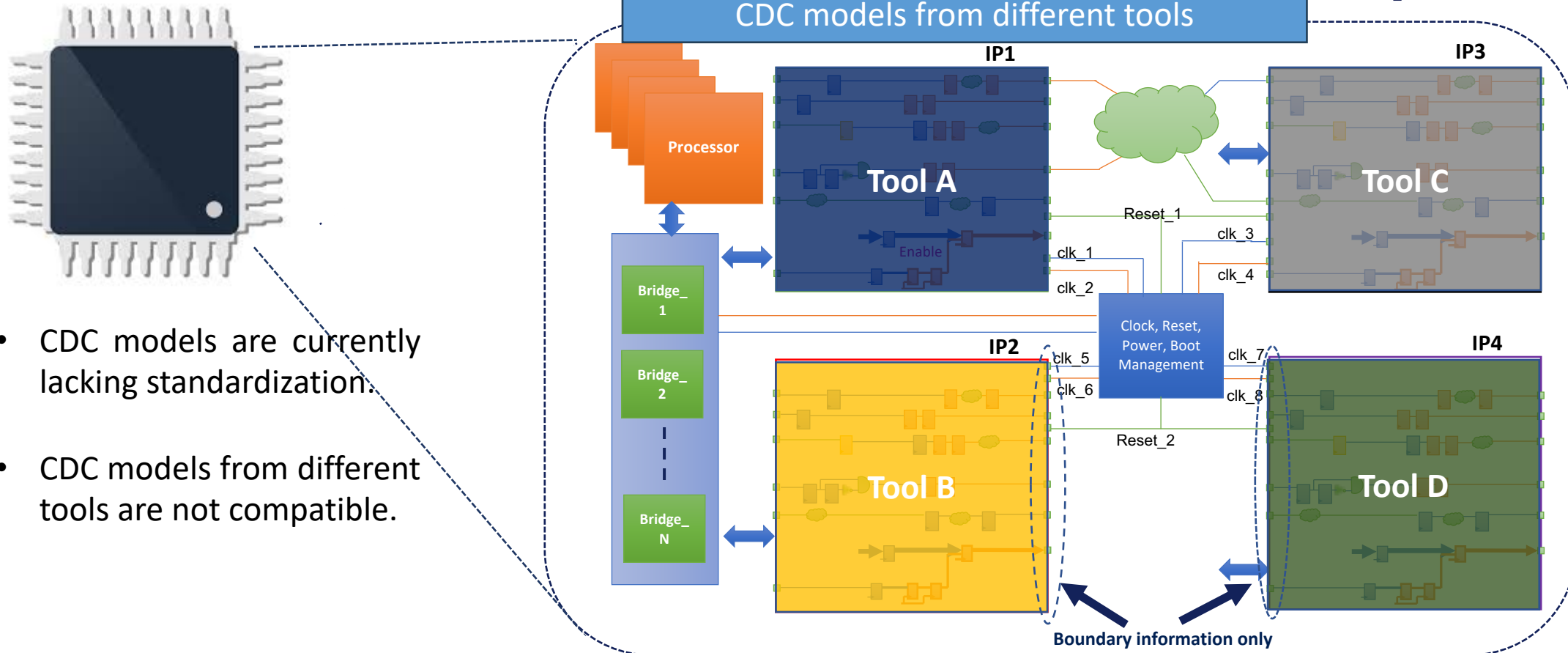
# CDC-RDC Hierarchical Flow

- Why?
- How?
- CDC Models from different tools

# CDC-RDC Hierarchical Flow (1/2)



# CDC-RDC Hierarchical Flow (2/2)





# Accellera CDC Working Group

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





# 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  
**124 members from 24 companies (as of Feb 19 '24)**  
5 active sub-groups: Output-Collateral, Format, Assertions,  
Testing, Training

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

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

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

Constraints/Assumptions  
can be verified with SVAs

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



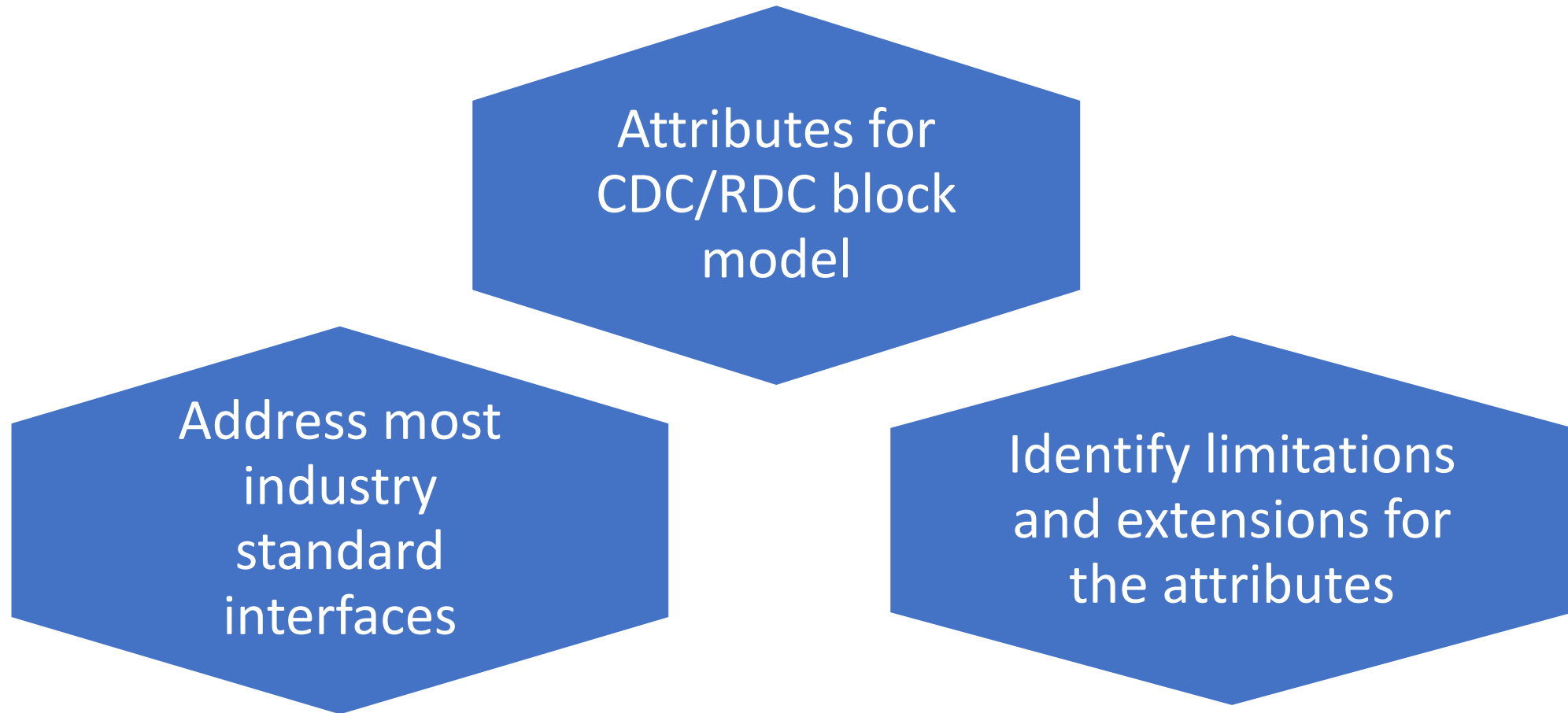
# Accellera CDC Working Group

- Output Collateral Subgroup



# Accellera CDC WG initiative

Output-Collateral



# Summary of Changes To Attribute Table

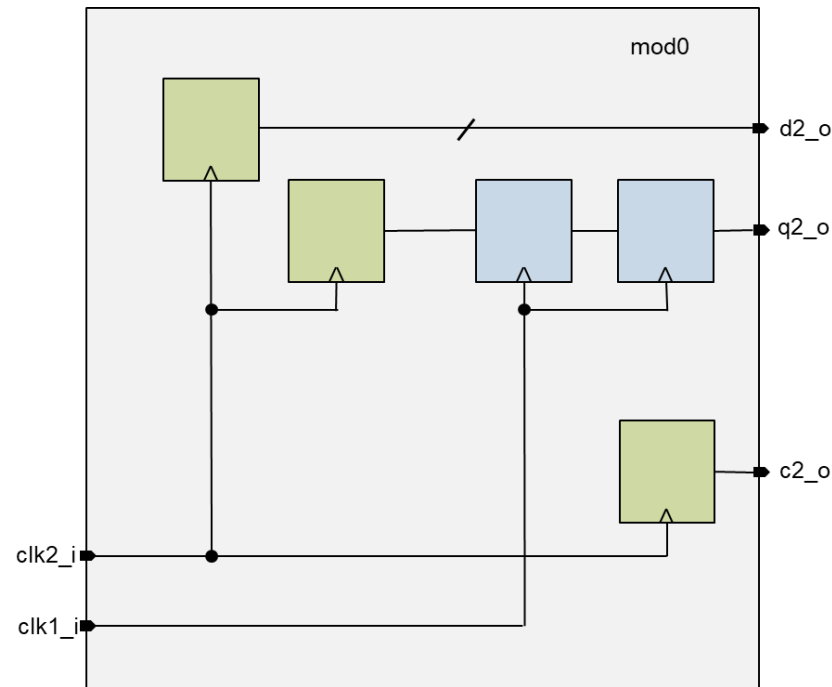
- Parameter
- Port Attribute Modelling
- Removal of keywords that are not EDA-Tool-Agnostic
- Removal of clock\_group and reset\_group
- Set\_cdc\_clock\_group
- Adding more examples to describe usage of reset, polarity, ignore, cdc\_static and constant
- Deferring usage of “gray\_code” and clock\_period to LRM 0.2

Attribute	Switch	data_type	values	Mandatory
module	name	string	{module name}	Yes
parameter	name	string		Yes
parameter	value	range-list	{ values }	optional
parameter	type	string	{ int, string, boolean }	optional
parameter	ignore	string	{ignore}	optional
port	name	string	{ name }	Yes
port	direction	defined set	{ input, output, inout, <b>internal</b> }	Yes
port	type	defined set	{ data, clock, async reset, qualifier, <b>supply</b> , <b>analog</b> }	Yes
port	logic	defined set	{ combo, buffer, inverter, <b>internal</b> , glitch-free-combo, internal-sync }	optional
port	qualifier_from_clock	; separated list	{ clock-names }	optional
port	associated_clocks	; separated list	{ clock-names }	optional
port	associated_resets	; separated list	{ reset-names }	optional
port	associated_inputs	; separated list	{ ports }	optional
port	associated_outputs	; separated list	{ ports }	optional
port	<b>clock_group</b>	string	{clock group}	Yes
port	<b>reset_group</b>	string	{reset group}	Yes
port	qualifiers	; separated list	{ associated-ports }	optional
port	polarity	defined set	{ high, low, both }	Yes
port	ignore	string	{ blocked, hanging }	optional
port	quasi_static	; separated list	{ <b>be any</b> , <clocks to which it is quasi_static> }	optional
port	set_case_analysis	; separated list	{ <b>binary</b> , <b>hex</b> , and of any length}	optional
port	gray_coded	boolean	{ true, false:default }	optional
port	clock_period	string	{clock period}	optional
tool	name	string	{EDA 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 who generate the collateral}	optional
design	description	string		optional
set_cdc_clock_group	async	string	–group {logical_ck1, logical_ck2}    –group {logical_ck1} –group {logical_ck2}	mandatory
set_cdc_clock_group	sync	string	–group {logical_ck1, logical_ck2}    –group {logical_ck1} –group {logical_ck2}	mandatory



# Port Attribute Modelling - output interface example

Output-Collateral



port -name d2\_o -type data -direction output -associated\_from\_clock clk2\_i -cdc\_control q2\_o

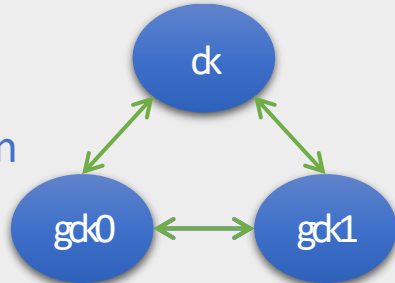
port -name q2\_o -type cdc\_control -direction output -cdc\_control\_from\_clock clk2\_i  
-associated\_from\_clock clk1\_i -associated\_output d2\_o

port -name c2\_o -type data -direction output - associated\_from\_clock clk2\_i

# set\_cdc\_clock\_group

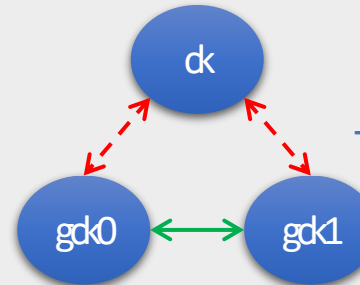
Output-Collateral

One domain



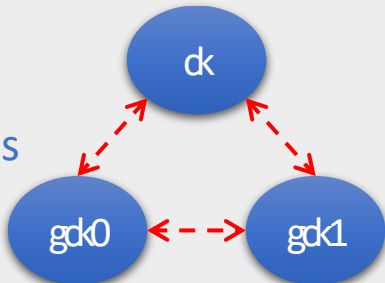
```
set_cdc_clock_group -name 1dom {ck gck0 gck1}
```

Two domains



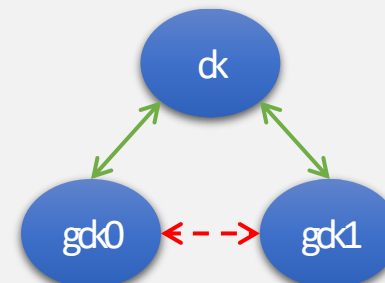
```
set_cdc_clock_group -name 2domA {ck}  
set_cdc_clock_group -name 2domB {gck0 gck1}
```

Three domains



```
set_cdc_clock_group -name 3domA {ck}  
set_cdc_clock_group -name 3domB {gck0}  
set_cdc_clock_group -name 3domC {gck1}
```

Domains?



```
set_cdc_clock_group -name trickyA {ck gck0}  
set_cdc_clock_group -name trickyB {ck gck1}
```



# Accellera CDC Working Group

- Format Subgroup



# Accellera CDC WG initiative

[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.

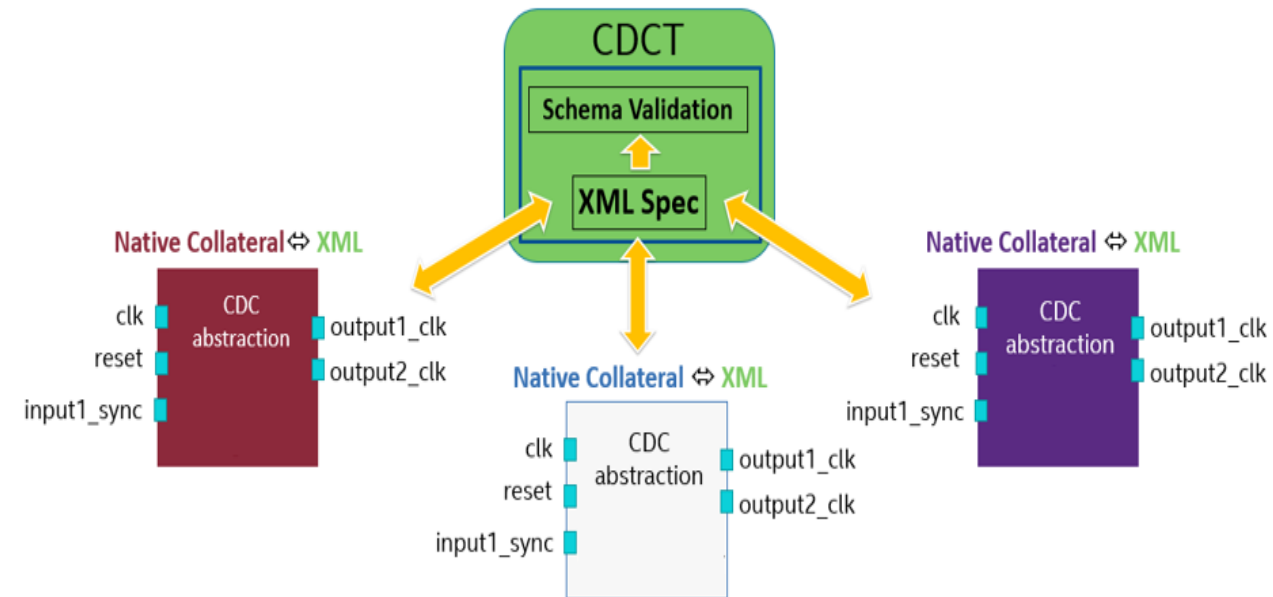
- **Methodology**

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

# Accellera CDC WG initiative

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.



# Accellera CDC WG initiative

[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



# Accellera CDC WG initiative

[Format](#)

- IP-XACT vs Tcl
  - 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 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

# Accellera CDC WG initiative

[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



# Accellera CDC Working Group

- Assertion Subgroup



# Accellera CDC WG initiative

Assertion

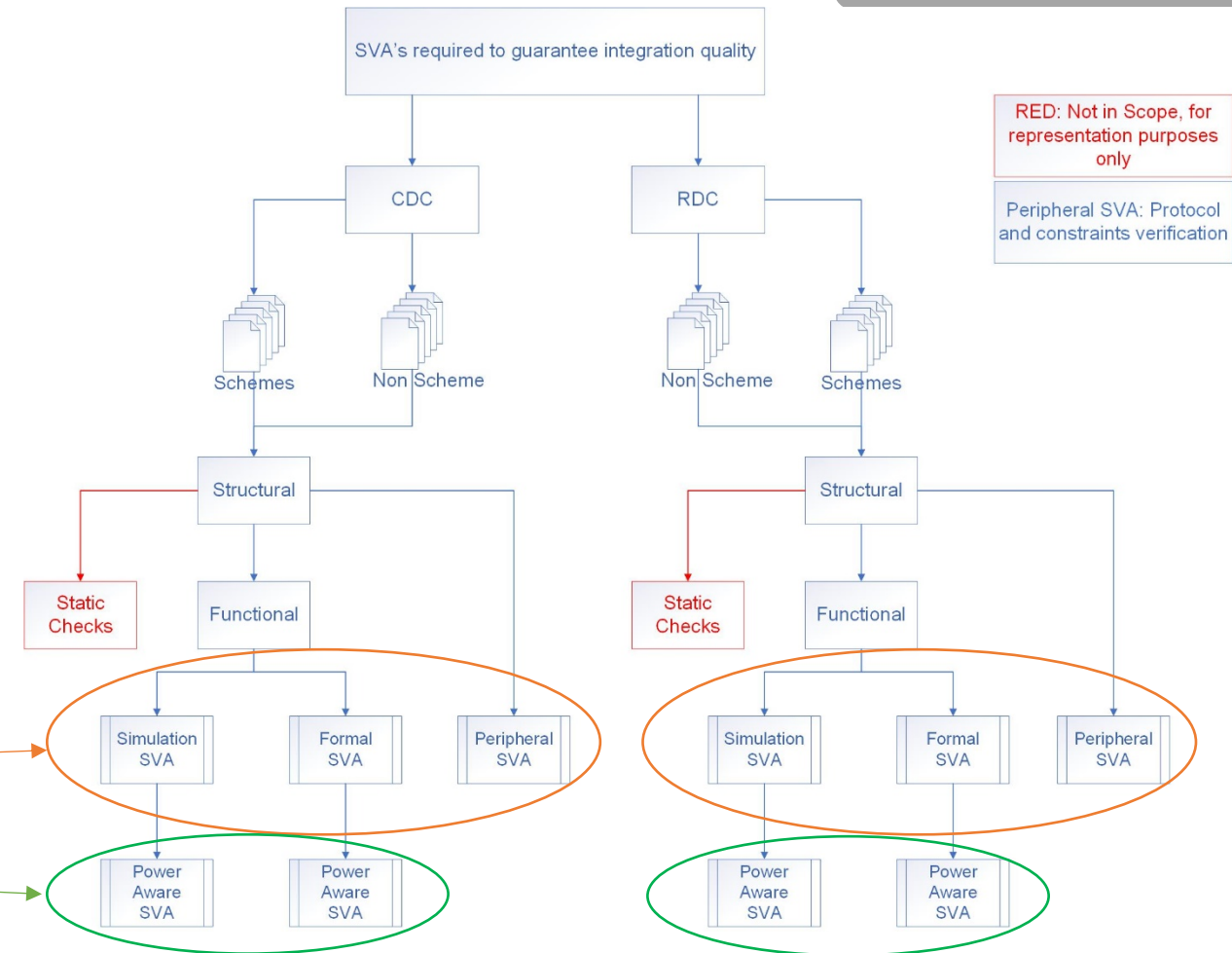
- **Goal**

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.

# Accellera CDC WG initiative

Assertion

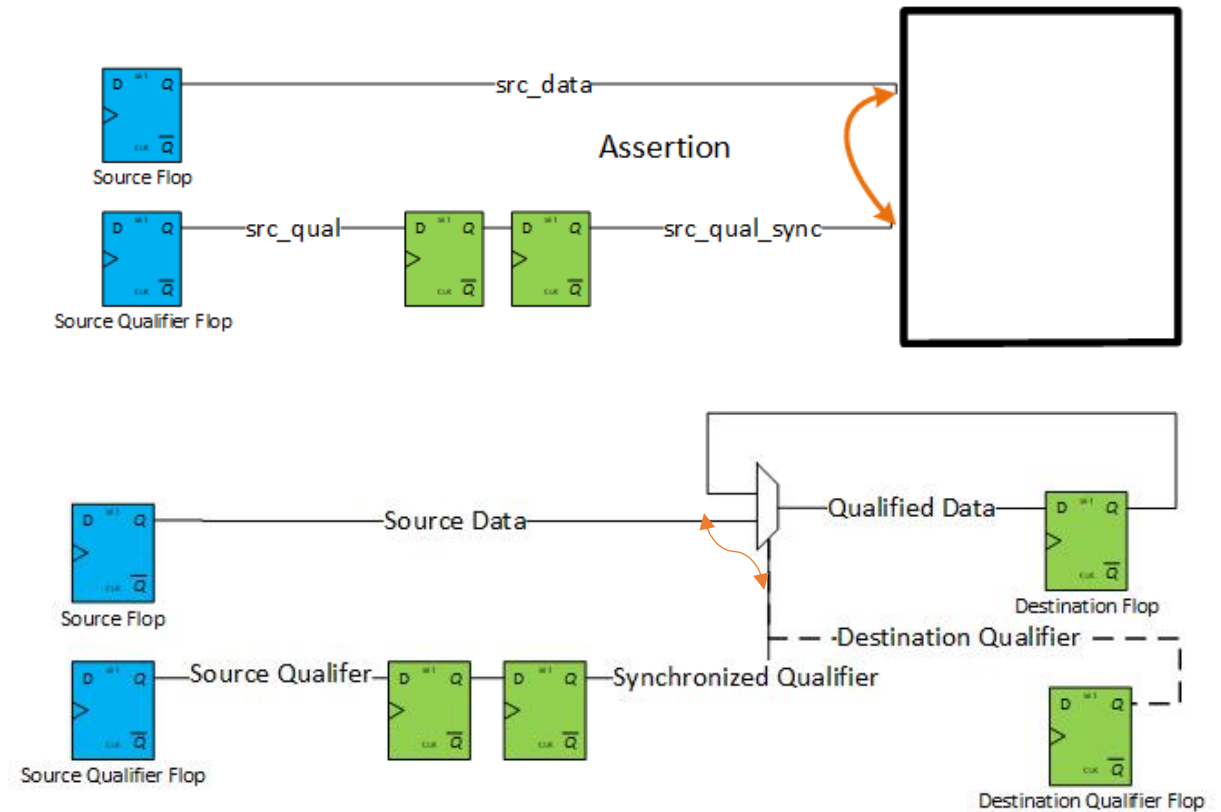
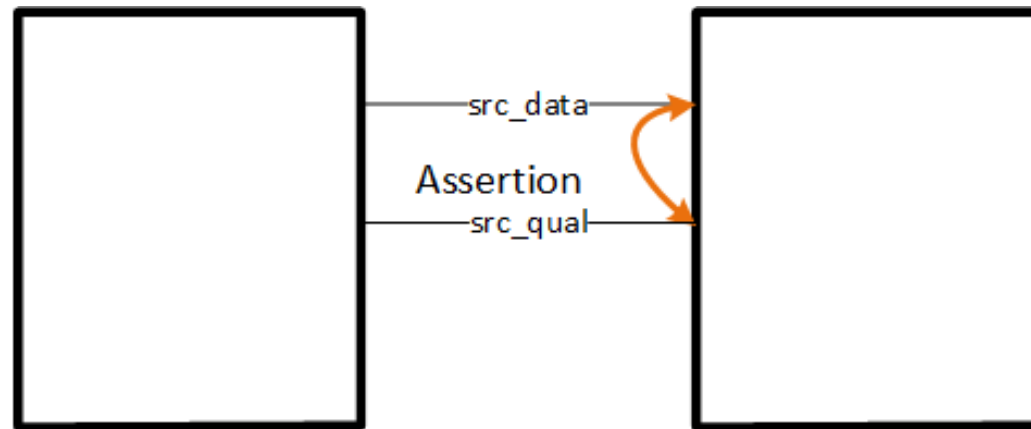
- Sub-working groups presentation
- CDC architectures are studied for possible verification strategies.
- Guidance to produce re-useable SVA for both Formal and Dynamic Verification.
- Guidance extracted from collateral.
- Guidance must follow SVA LRM.
- Guidance must be tool/vendor independent.
- Current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



# Accellera CDC WG initiative

Assertion

- Sub-working groups presentation
- Possible Integration Scenarios under consideration:
  - 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.





# Accellera CDC Working Group

- Testing Subgroup





# Accellera CDC WG initiative

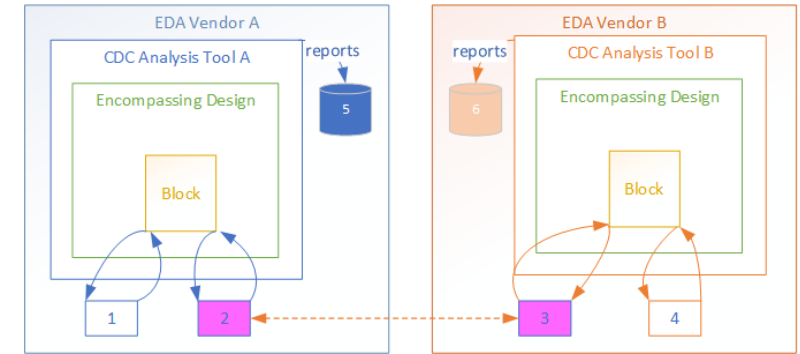
Testing

- Sub-working groups presentation

- **Goal**

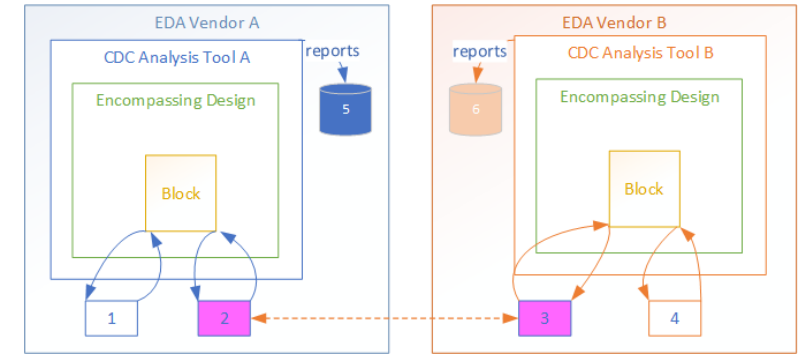
1. Evaluate the set of Accellera CDC attributes and protocols for completeness using multiple tools from multiple vendors.
2. Demonstrate the use of the complete set of attributes and protocols as defined and formatted by other sub-groups.
3. Provide RTL design examples within which the defined attributes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

# Methodology (#1)



- **Testing 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...]



- 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 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



# Accellera CDC Working Group

- Training Subgroup



# Accellera CDC WG initiative

Training

- Sub-working groups presentation

- **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

# Accellera CDC WG initiative

Training

- Sub-working groups presentation
- Accellera CDC WG work promotion through conferences
  - Past/current conferences
    - DVCON Europe 2023
    - DVCON US 2024
  - Targeted conferences (To Be Confirmed)
    - DVCON Japan / India / China / Taiwan / Europe 2024
    - DAC 2024
    - DATE 2025
    - VLSI 2025



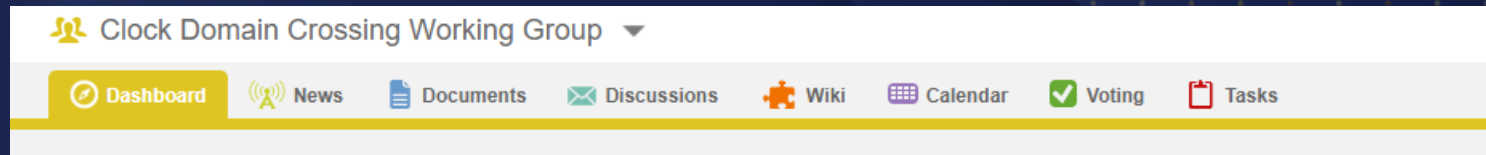


# CDC LRM Draft 0.3 open for public review till Sept 9, 2024

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

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

# Accellera CDC Working Group Call for Contribution !



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



# Questions ?

