Fully Hierarchical CDC Analysis Using Comprehensive CDC Meta Database

Youngchan Lee, Youngsik Kim, Soenil Brian Choi
Samsung Electronics Co., Ltd., Seoul, Korea
(yc0325.lee@samsung.com, ys31.kim@samsung.com, seonilb.choi@samsung.com)

Vikas Sachdeva
Real Intent Inc., Sunnyvale, CA 94089, U.S.A
(vikas@realintent.com)

I. Backgrounds and Challenges

All SOC designs today have many subcomponents with complex multi-clock interactions. These independently developed components come together to enable a rich feature set for the SOC. Accompanying this abundance of features, there is a proliferation of internal and external protocols and aggressive power requirements that lead to an explosion in the number of asynchronous clocks in SOC's. Ensuring that this complex SOC works according to specifications, demands that design and verification teams spend an increasing amount of time on verifying the correctness of asynchronous boundaries in the chip.

Traditionally, flat CDC analysis has been used to verify CDC correctness of the SOCs. In flat CDC verification full design is elaborated and analyzed using SOC level constraints. CDC violations which were already reviewed in the IP are analyzed again. Also to sign-off a constant communication is needed between SOC engineer and IP engineer. In recent years hierarchical approaches for CDC verification have also been explored as a way to partition the problem into manageable chunks and enable more predictable schedules. In hierarchical CDC verification, each IP or block is verified and its CDC information is saved. As higher-level subsystems or SOCs are assembled, the CDC information from the lower levels is used at the top level for CDC verification. Hierarchical approaches thus far have relied on saving boundary models for IPs and reusing those models for SOC top-level analysis. Boundary models are good for identifying block vs. SOC assumption mismatches, but they provide only limited coverage for complex CDC interfaces and re-convergences when distributed across IPs. Since distributed interfaces are a norm in today's SOCs, traditional hierarchical approaches end up providing only limited coverage from the CDC point of view.

II. TRADITIONAL CDC ANALYSIS FLOW

A straightforward way to do CDC verification on an SOC is by doing a flat run. A flat CDC verification run is relatively easy to setup and covers are CDC issues. Ensuring correct setup is a critical part of CDC verification and can be a significant effort. The simplicity of setup is a major advantage of flat CDC verification. How there are also some disadvantages with this flow. One is that full-SOC RTL is only available late in the design cycle, at a time when iterations are costly. Moreover for huge SOCs the huge volume of CDC data can make debug overwhelming and inefficient.

Flat CDC verification usually works well for small SOCs, but for big SOCs a robust hierarchical methodology is much more effective. Many verification and implementation tools such as synthesis, static timing analysis and functional verification support hierarchical methodologies that promote IP-reuse principles and divide-and-conquer paradigm for breaking down the volume of data.

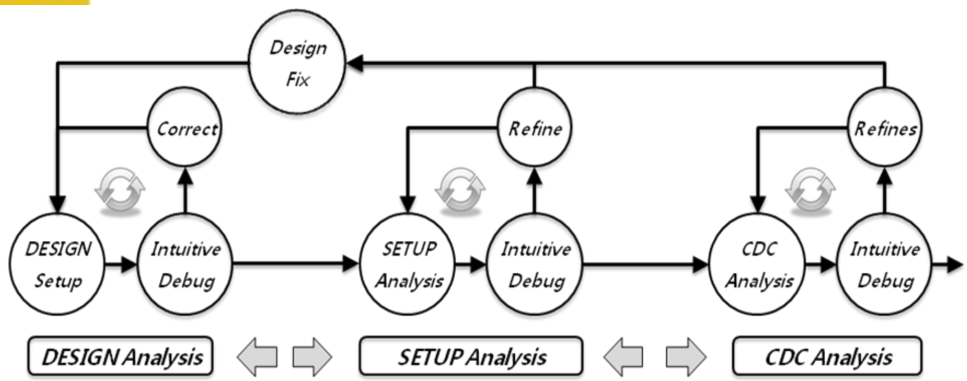


FIGURE I. FLAT CDC SIGN-OFF FLOW

III. SUGGESTED NEW HIERARCHICAL APPROACH

This paper introduces an innovative systematic approach to enable accurate hierarchical CDC verification. The approach uses IP-reuse principles and a divide-and-conquer paradigm to remove redundancy in the CDC verification effort. Rather than using boundary model (constraint-based), the proposed approach uses a new comprehensive CDC Meta-database. This Meta-database stores comprehensive CDC-relevant information such as CDC components, CDC environment, CDC assumptions, synchronizers, clock trees, boundary connectivity and CDC configurations. Said Meta-database ensures that the CDC report at the IP level, combined with the hierarchical CDC report at the SOC level, provides flat-equivalent CDC sign-off coverage. Since all CDC-relevant information is stored in the database, it allows consistency checks between IP and SOC to catch CDC-intent mismatches. The approach also saves communication effort between IP providers and consumers as CDC-relevant information and assumptions are saved in the Meta-database and is available for SOC designer use. In our flow, the Meta-database was saved as a deliverable in our IP-management system so information could be reused for multiple SOCs at the same time, leading to significant effort-reduction and productivity gains for our SOC and IP teams.

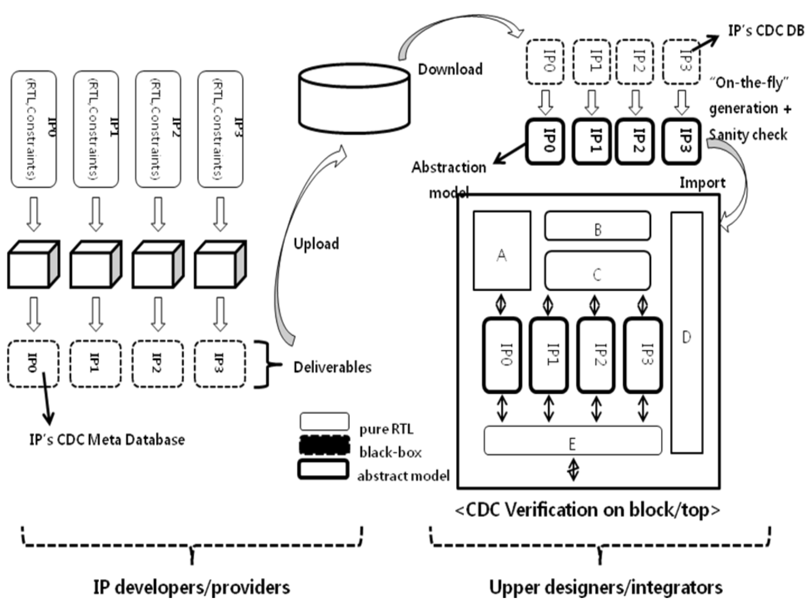


FIGURE II. CONCEPT OF THE THE NEW HIERARCHICAL CDC ANALYSIS

In the new hierarchical bottom-up CDC verification flow, each IP or block is verified and its CDC information is saved. As higher-level subsystems or SOCs are assembled, the CDC information from the lower levels is used at the top level for CDC verification. In a bottom-up flow, the constraints for each block are developed separately by each block-level designer. Also, SOC constraints may be developed independently from block-level constraints. This can lead to inconsistencies among these constraints. So, in a bottom-up verification flow it is a must to verify the coherency of these constraints.

The first step in the flow is to do flat block-level CDC runs and sign-off on those CDC results. Once those have been signed-off, a CDC Meta-Database, to be used at a higher level, is written out. At the next higher level, the first step is to check hierarchical consistency. These consistency checks indicate whether the Meta-Database needs to be regenerated or not. Once hierarchical inconsistency has been resolved, the CDC violations can be analyzed. Once these CDC violations have been analyzed, CDC sign-off at the next higher level finishes.

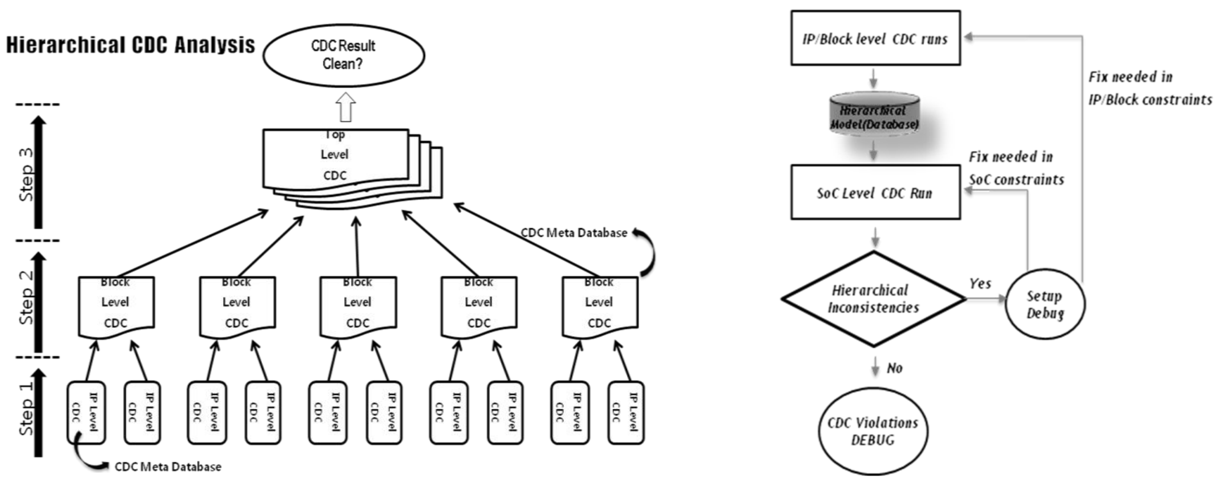


FIGURE III. SIGNOFF USING NEW HIERARCHICAL CDC ANALYSIS

IV. CDC META-DATABASE

The accuracy of new hierarchical flow depends on the amount of information which is saved in CDC Meta-Database. If the amount of information that is saved is limited for example only boundary clock information the sign-off is not as comprehensive as flat CDC Sign-off. If the information saved is too comprehensive then the performance and capacity gains may not be as expected from hierarchical flow. So the challenge for the hierarchical CDC methodology is to save but sufficient information to provide flat level accuracy but still concise enough to provide performance and capacity gains. The comprehensive CDC Meta-Database contains following kind of information:

- **Port Connectivity**
  Information about how input and output ports of the blocks, clock domains on those blocks and how these ports are connected to block internal flops.

- **Environment**
  Information about all kinds of CDC constraints applied at the block level.

- **CDC Configuration**
  Information about CDC configuration used to sign-off at block level, for examples what synchronizer depth were used. This information can be used to ascertain quality of IP Meta-Database which is delivered to upper-level IP consumers implicitly.

- **Synchronizer Connectivity**

Information about synchronizer connectivity in the block. This is important to do advanced CDC checks like Reconvergence, Glitches and CDC interfaces.

- **IP Assumptions**
  Information about CDC assumptions that were used by block designer to sign-off the block. This is not to be confused with waivers. CDC assumptions are well qualified properties which can be verified. For example two signals which cannot toggle together can be saved as an IP assumption.

- **Clock Tree Connectivity**
  Connectivity of the clock tree at the block level. This is important to handle internal clocks and clock multiplexers in the design.

- **CDC Attributes**
  CDC attributes on the internal logic of the design for example whether a signal can only toggle during reset phase or can also toggle during non-reset phase.

- **Design Information**
  The parameters used to elaborate the design and other such design information.

- **Schematic database**
  The schematic database of full block level design.

Traditional CDC hierarchical models mostly save only port connectivity. In recent years some new models have also been developed which advance the information saved to more than just port connectivity. The above mentioned model is unique as it saves comprehensive CDC information to provide flat level CDC accuracy with performance and capacity gains. Figure IV, represents difference between a traditional boundary based hierarchical model and the new CDC Meta-Database which this paper provides. While in boundary based model on boundary connectivity is saved and rest of the logic is black boxed, CDC Meta-Database is similar to a binary gray-box where all CDC and clock relevant logic is visible to upper level analysis.

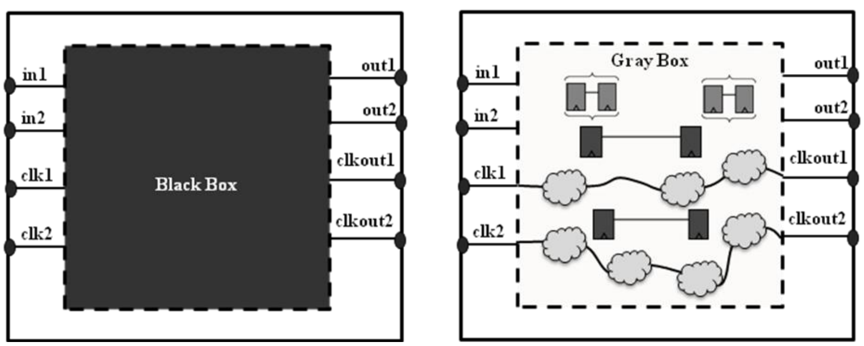


FIGURE IV. BOUNDARY MODEL VS. NEW CDC META-DATABASE

Since the CDC Meta-Database is comprehensive model it provides one more unique advantage not available in traditional CDC hierarchical models. Typically an IP can be instantiated with multiple different parameter values at SOC level. In above model designers just need to create meta model with default parameterization values and they can automatically create meta models for all other instantiations whatever parameter values they have.

The comprehensive CDC Meta-Database also provides following advantages over traditional models.

- **Significantly reduce volume of the data to be reviewed at top level**

  Using CDC Meta-Database a CDC tool can identify whether a crossing has already been reviewed by IP designer or not. Using that information CDC tool reports the violation at the top level. This reduces the number of violations that are reported at top level. Also it avoids duplication of reviews and same violation is not reviewed multiple times by IP designer and SOC designer. Consider the design in following figure V.

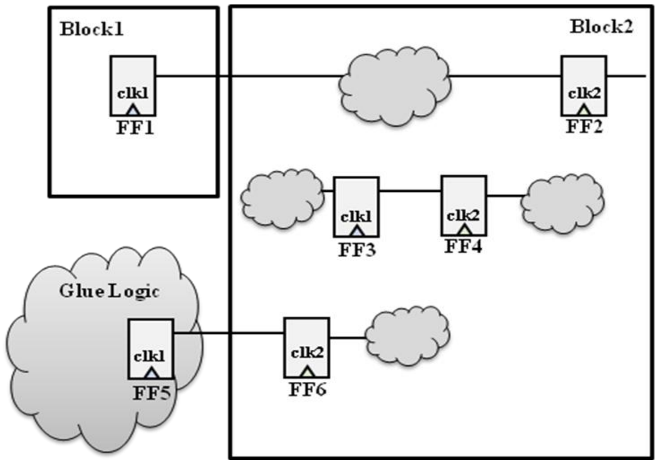


FIGURE V. BLOCK VS TOP CDC VIOLATIONS

Here CDC violations from FF3 to FF4 are within the scope of Block2 - these are the violations that should be reviewed at the Block2 level and should not be reported at the SOC level. Other violations, i.e. FF1 to FF2 and FF5 to FF6, should be reviewed at the SOC level.

- **Accurate reporting of inter-block CDC violations**

  At the SOC level there can be CDC violations which involve two or more lower-level blocks. These problems can only be identified while running CDC analysis at the SOC level. Consider the design shown in Figure VI, there are two synchronizers in this design (F5-F6 and F7-F8).

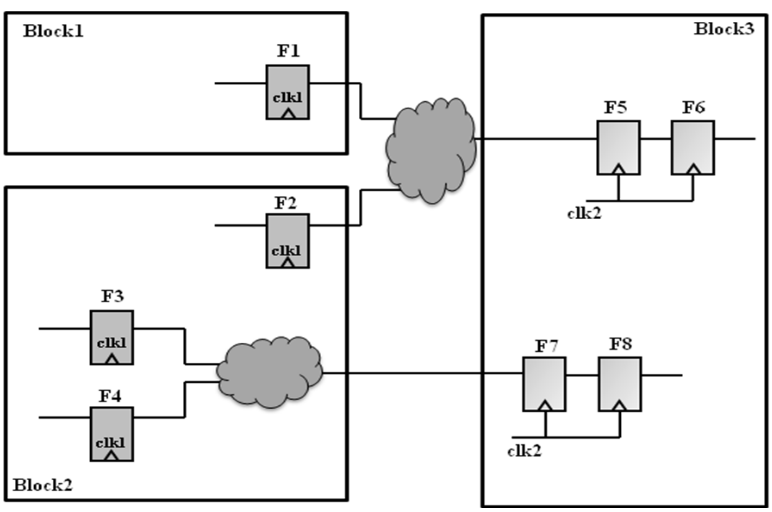


FIGURE VI. INTER BLOCK CDC VIOLATIONS

The first stage of each synchronizer, i.e. `F5` and `F7`, is being driven by combinational logic. Since we know that combinational logic driving the first stage of a synchronizer has the possibility of a glitch hazard being captured asynchronously by the receiving domain, both of the above are CDC violations. Designers doing

CDC sign-off at the block level may or may not be aware of how their block is integrated at the SOC level. As a result these violations only become visible when SOC is integrated.

The implication is that when CDC verification is done at the block level, the hierarchical model needs to save information about the ports in order to report CDC violations at the SOC level. Since CDC Meta-Database saves comprehensive information (all information required for flat-compatible CDC analysis) these kinds of CDC violations can be identified at top level. For example, consider the design in Figure VII. In this design, multiple synchronized signals (`syncout1, syncout2, syncout3`) are converging resulting in the hazard of loss of correlation effects from uncertainty. If a primitive TCL-based boundary model that just saves limited information about the ports is used, the CDC violation will not be reported at the SOC level. Even it is reported, the result from constraint-based boundary model will be much abstracted or rough report, giving designers poor debuggability.

- **Advanced CDC Sign-off coverage**

  The comprehensive CDC Meta-Database also allows CDC tool to do advanced sign-off checks like reconvergence and interfaces. For example, consider the design in Figure VII. In this design, multiple synchronized signals (syncout1, syncout2, syncout3) are converging creating the possibility of loss of correlation effects from these signals. If a comprehensive CDC Meta-Database is not used, the CDC violation cannot be reported at the SOC level in hierarchical flow.

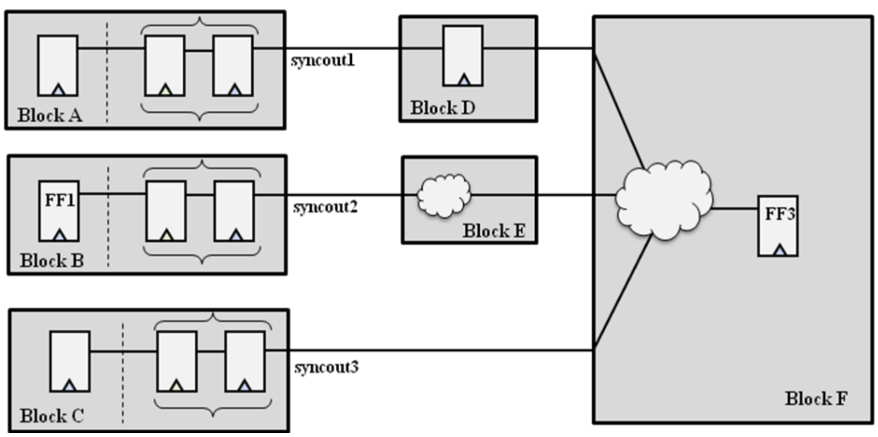


FIGURE VII. RECONVERGENCE CHECK IN HIERARCHICAL FLOW

- **IP Assumptions**

  Typically CDC sign-off is done using some architectural and design specific assumptions available in designers head only. Traditionally these assumptions are converted into waivers and then those waivers are used to signoff the reports. After block has been signed-off SOC level designer either incorporates those waivers in top level analysis directly or he constantly interacts with IP designer to understand those assumptions. If an IP is instantiated in multiple SOCs, IP designer has to interact with all the SOC designers even though IP CDC assumptions are the same for each SOC. This reduces productivity of both IP team and SOC team. The new CDC Meta-Database takes a completely new approach here and saves IP assumptions in the model. These assumptions are then first checked at SOC level and then based upon that reduce the number of violation automatically.

V. Methodology Changes For New Hierarchical Flow

In this section, we describe things that we have done or things that should be taken into consideration to adopt the new hierarchical flow. The followings are the changes while we needed to change on our previous SOC CDC verification environment.

IP Deliverables
Earlier for flat CDC sign-off flow, IP provider was required to deliver RTL, waivers and constraints used for sign-off. In the new hierarchical flow the IP delivery was simplified and IP provider only needed to provide RTL and CDC meta-database (i.e. IP.db). The waivers and constraints are included in the Meta-Database implicitly and used for upper-level sanity check. Now IP providers don't need to care about the deliverables except for RTL with the far more simplified IP delivery system.

Strengthened IP Sign-off Quality
In flat analysis based system, IP owners put less focus on IP boundary constraining and assumptions because the flat analysis had taken case of most of the checks. With this new hierarchical environment, the IP sign-off rules system has been strengthened even more. Several new checking rules were added for the boundary checking

Strategies Selecting IPs for Abstraction
Not all IP is needed to be abstracted for IP-to-Block hierarchical flow. Too many IPs abstracted result in another headache of managing them. Also, the abstraction of some IPs, for example bus bridge IPs, can give confusion the CDC tool and results in other kind of false-negatives or human effort for constraining. It is necessary to choose proper IPs for hierarchical flow to maximize the effect of the new methodology.

On-the-fly Sanity Check for Parameterized IPs
To handle all the parameterized cases, IP level designer only needed to submit CDC Meta-Database for default parameter values. At higher level if IP was instantiated using non default parameter values a CDC Meta-Database with instantiated parameterized was automatically generated using CDC Meta-Database of default parameter values. If the new generated CDC Meta-Database did not meet IP sign-off quality metrics the IP designers were appropriately intimated to clean the results and meet sign-off quality metrics.

IP Assumptions
IP designers were motivated to add IP assumptions in their CDC runs. These assumptions later increased productivity by saving time involved in multiple interactions between SOC designer and IP designer. They also contributed to the reduction on the number of less important CDC warnings.

VI. Results

To take advantage of the effectiveness and efficiency of our new approach, we employed it on the verification of our premium-sized SOC designs together with 2-level hierarchical flow, one is for IP-to-BLK abstraction and the other is BLK-to-TOP abstraction. Normally, physical design blocks include 1 or 2 big mega-IPs and many small IPs and we targeted the big mega-IPs for abstraction to maximize the effect of hierarchical philosophy. For the BLK-to-TOP hierarchical flow, all the blocks (physical design units) were abstracted for our analysis.

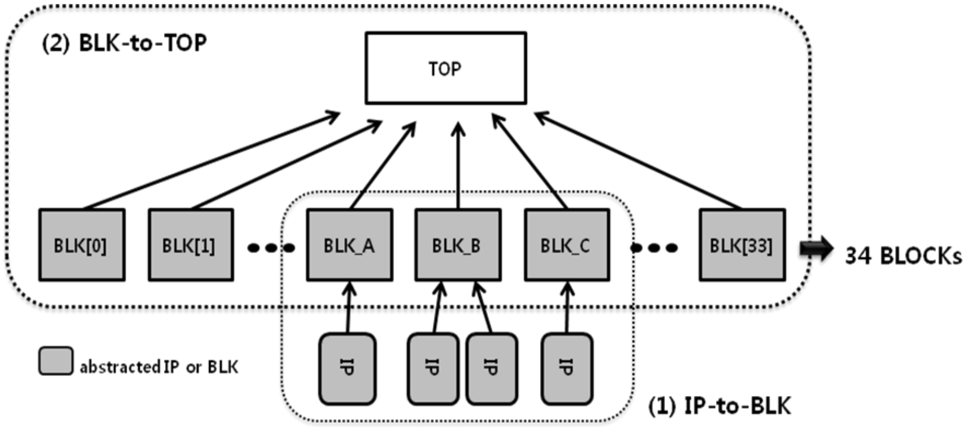


Figure VIII. Scope of Results

Table I. shows the results of 3 big blocks among all blocks. Several rule policies were newly defined to qualify the inconsistency between IPs and blocks. This inconsistency checking ensures there is no missing CDC path during abstraction and encapsulation of IP's intra-block crossings. Also the new checking policy helps IP consumers identify the design intents and attributes like quasi-static paths, constant or mutually-exclusive paths, glitch-safe attributes and so on. By cleaning the setup issues and inconsistencies first, we could get even cleaner results and this help block owners focus on the most real problems. From Table I. and Figure VII, we could reduce the number of reports for review dramatically as well as the hardware resources (run-time and memory). In all cases more than 90% of all paths were suppressed by encapsulating intra-IP paths and introducing the boundary checks.

TABLE I. FLAT ANALYSIS VS. HIERARCHICAL ANALYSIS ON 3 WORST BLOCKS

CDC rule categories / measured performance	Block-A		Block-B		Block-C	
	Flat analysis	Hierarchical analysis	Flat analysis	Hierarchical analysis	Flat analysis	Hierarchical analysis
Rough size [Gate-count]	23M	-	40M	-	16M	-
Number of Mega-IPs abstracted	-	2	-	2	-	1
Setup rule check	86	53	166	144	44	36
*** Boundary consistency check	N/A	243	N/A	8	N/A	8
Glitch check	110	39	248	131	71	29
Synchronization check	23,445	152	11,627	2817	7,356	23
Reset check	348	142	6239	5091	282	69
Run-time[min] (each IP runtime included)	26	5 (18)	32	16 (28)	11	2 (7)
Memory consumption [GB]	9.5	4.9	17.9	8.3	5.2	3.9
*** Estimated human effort	100%	27%	100%	34%	100%	28%

*** Boundary check rule(available only on hierarchical analysis)
*** Block : Physical implementation unit(SOC top consists of 30 ~ 40 blocks typically.)
*** Tool used : Meridian CDC(Real Intent Inc.), with verbose options fully enabled to count all paths
*** Estimated human effort : Calculated based on designers' feedback for review

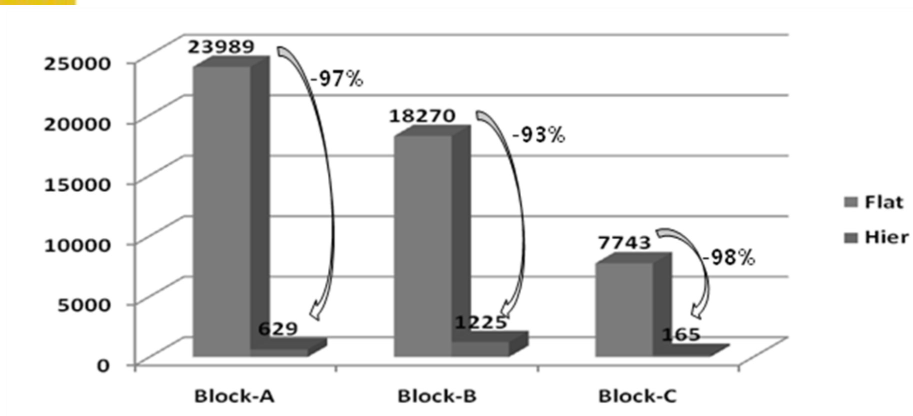


Figure IX. Comparison Of The Number Of Report Flat analysis vs. Hierarchical analysis on 3 worst blocks

The similar approach was used to verify top design and the same merits were acquired. Furthermore, we could relieve the overhead of machine resources (run-time and memory) enormously, which had been one of the biggest challenges on SOC verification.

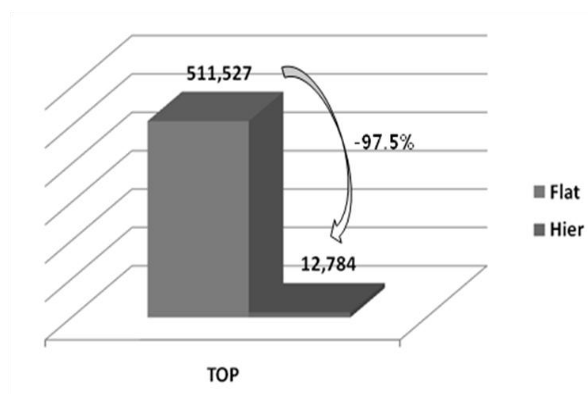
On the stage of block analysis, hardware resources (run-time and memory) are not so important because the turn-around time is relatively small. But, At initial stage of top CDC analysis, the turn-around time of a job is much more important because 8 hours for setup analysis or about 30 hours for full structural analysis would not be reasonable also even more impractical if we consider the tight project schedule and continuously increasing size of SOC design. By way of the new hierarchical approach we could renew the result twice a day and now the machine limitation is not the bottle-neck anymore but human resource for the check and review became a new bottle-neck. This is really a big enhancement if we think it takes about 30 hours to get a new result with flat analysis.

TABLE II. FLAT ANALYSIS VS. HIERARCHICAL ANALYSIS ON SOC TOP

Items for Measurements (Design info., Rule check counts, Performance)	TOP	
	Flat analysis	Hierarchical analysis
Rough area[Gate count]	800M	-
Number of Blocks Abstraced	-	34
Setup rule check	3,223	333
Boundary consistency check ***	N/A	4,220
Glitch check	10,945	947
Synchronization check	474,637	7,209
Reset check	22,722	75
Run-time [min]	1,796	214
Memory consumption [GB]	242	89
Estimated human effort	100%	34%

*** Boundary check rule(available only on hierarchical analysis)
*** Block : Physical implementation unit(All 34 blocks abstracted for hierarchical analysis)
*** Tool used : Meridian CDC(Real Intent Inc.), with verbose options fully enabled to count all paths
*** Estimated human effort : Calculated based on designers' feedback for review

FIGURE X. COMPARISON OF THE REPORT COUNT(TOP)



VII. CONCLUSION

The verification of CDC on 300M ~ 400M gate count SOC design had been a demanding and time-consuming job and the endlessly increasing size of SOC is even more worsening the verification environment. In this paper, we introduced a new hierarchical CDC flow that imports a comprehensive meta-database for complete and rigorous CDC analysis of hundred-millions SOC designs. The meta-database described in this paper includes most of the information required for CDC verification, so it can be used for bottom-up CDC analysis perfectly instead of flat RTL. By using this CDC database, a flat-equivalent CDC sign-off coverage with a little performance trade-off that would not have been possible if we used constraint-based boundary model. We also explained what we had to change on our CDC flow and environment to adopt this new hierarchical flow successfully. Finally, we showed our verification results from our real project together with specific data and numbers and they show we could be able to acquire a big reduction on human effort as well as mechanical performance gain.

VIII. REFERENCES

[1] Clifford E. Cummings, Synthesis and Scripting Techniques for Designing Multi-Asynchronous Clock Designs, SNUG 2001

[2] Clifford E. Cummings, Clock Domain Crossing (CDC) Design & Verification Techniques Using System Verilog, SNUG 2008

[3] Meridian CDC Reference Manual P14.1, Bottom-Up Hierarchical CDC Verification Flow, Real Intent, Inc., 2018

[4] Meridian CDC Reference Manual P14.1, Rule Reference, Real Intent, Inc., 2018

[5] Millions to thousands issues through knowledge based SoC CDC Verification, ISOCC 2015