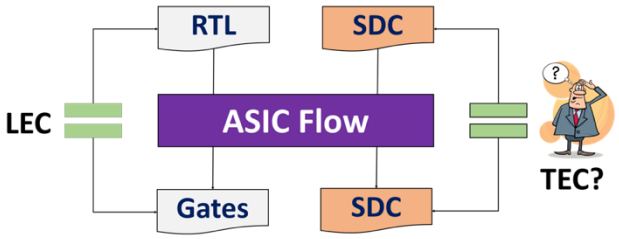
TEC – Timing Equivalence Checking; a modern design flow necessity

Chip design essentially has two components; functionality and timing. Both are critically important and dependent on each other. As the design matures, designers perform LEC (Logical Equivalence Checking) to verify the functional correctness of their design throughout the design cycle. However, the timing side of the equation is completely ignored as, up until now, there was no such thing as "timing equivalency checking" or TEC.



What is TEC? – In today's ASIC flow, running LEC alone to check for functional equivalence is not sufficient, rather both the functionality and the timing equivalency, with SDC annotated to the design should be checked. Thus, TEC is an essential task that bridges the gap between functionality and timing.

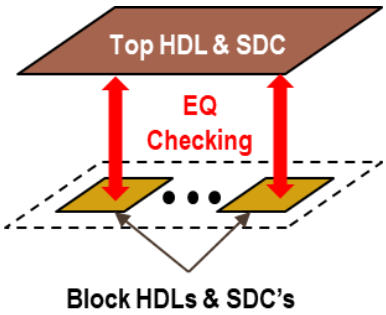
Since there is little available options performing TEC, much of the work is done manually through manual inspection and based on experience and design knowledge. Excellicon realizing the need for a comprehensive and systematic approach to validate equivalency of the timing constraints have developed a series of very complex and comprehensive features to ensure proper propagation of timing constraints while checking the equivalency of the constraints in its Constraints Certifier; ConCert, platform.

Excellicon Solution

1. Top level against the hierarchical Block level (T2B)
2. Top level against the Top level (T2T)
    a. 2D2S – Two top level designs and their corresponding SDC's
    b. 2D1S – Two top level designs but single SDC
    c. 1D2S – Single top level design, but two SDC's
    d. M2M – Merged mode SDC against Modal SDC
    e. F2H – Flat SDC against Hierarchical SDC
    f. F2Box – Flat SDC against Hierarchical SDC with Blackboxes
    g. S2ETM – SDC against the ETM model

Top to Block (T2B) Mode

In the T2B mode, the TEC is critical in checking for the consistency of timing between different levels of hierarchy. As an example, many design flows use previous version of the timing constraints for the next revision of the chip. Many IPs also come with their own timing constraints developed independent of the design at some point during the IP design life-cycle, perhaps modified multiple times. Once the constraints are assigned to various blocks, then the big question is that if there is equivalency between various layers of hierarchy or consistency across the blocks and the top level.



A simple case may be that the clocks are designed to be synchronous at the top level, while at the IP level, they are meant to be asynchronous. Assumed constant values set by the *set_case_analysis* command at the IP level may conflict with what is defined at the top level.

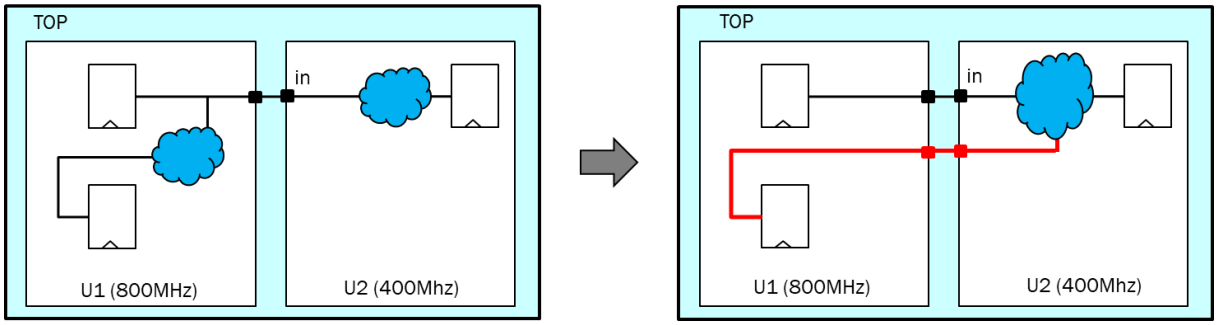
Yet another typical case is when design's timing constraints are promoted or demoted based on design flow requirements, various issues may arise leading to timing inconsistencies which may result in many timing closure iterations. Additionally, many times the design is repartitioned and there is a need for a systematic validation of all associated timing constraints with respect to underlying functional code.

For most comprehensive and most efficient TEC, there is a need for an independent audit of the design timing constraints at various stages of design. Without performing detailed TEC many cycles are wasted trying to pare down the issues caused due to lack of equivalency. It will be even impossible to tapeout a chip with certainty with respect to timing.

Top to Top (T2T) Mode

The T2T TEC is used for checking for impact of timing as the design undergoes various changes throughout the design cycle, starting from RTL, then to synthesized gates; to routed design and finally to ECO's (if any). It is essential to validate that the original timing intent has not changed during this design cycle.

Optimization engines for Synthesis or P&R employ many techniques to meet the timing. These include logical restructuring, cloning, de-cloning, pin-swapping, additional buffering etc. The resulting design, although functionally equivalent to the original, may contain new paths that may be timed differently (or not be timed at all) as originally intended. Similarly, ECO's performed on the design structure may also result in the same issue if the original SDC is applied on the ECO'd design, without checking for timing equivalence.

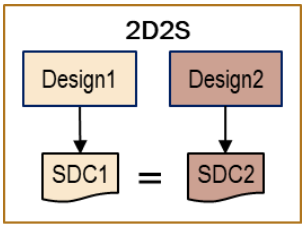


Consider the example shown in the above figure. After logic restructuring, the resulting logic is optimized by combining the logic from U1 and U2. Functionally, they are equivalent, however, from the timing point of view, the original SDC may not be valid, if applied on the resulting design. For example, if there is a timing exception placed on port "in", such as:

set_multicycle_path 2 –setup –through [get_pins U2/in]

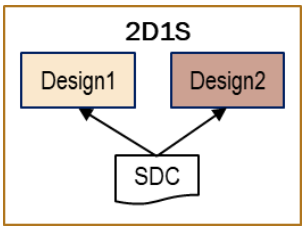
In this case, in the original design shown on the left side, only the U2 registers were multi-cycled, however, on the optimized design, the multicycle not only covers the U2 registers, but impacts the U1 registers also.

1. 2D2S – Two designs and corresponding SDC's



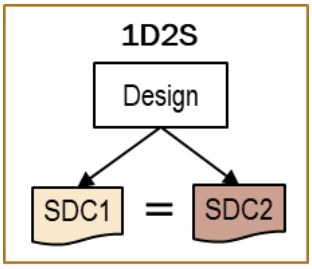
This method is applicable when there are two designs and their corresponding SDC's. For example, Design1 may be RTL and Design2 could be gates. This method can also be used for a situation when there are new revisions of the chip being designed and the designer needs to understand whether the timing constraints files are the equivalent between various design revisions.

2. 2D1S – Two designs, but same SDC



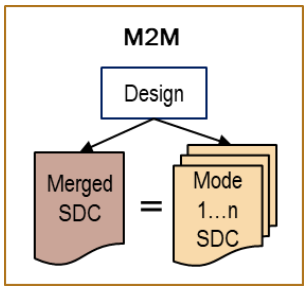
Similarly there are situations when there may be two different revisions of design and the same SDC is applied for both. For example, Design1 can be pre-ECO, while Design2 is post-ECO. Both designs do not need to be functionally equivalent.

3. 1D2S – Single design but different SDC's



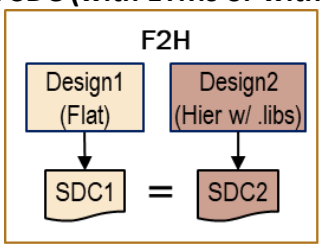
A case when there are two different constraints files for a single design. For example, the SDC is changed as the chip moves across the design flow and the designer wants to understand the differences from the original timing intent.

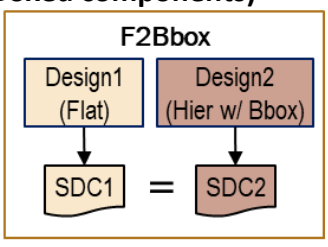
4. M2M – Merged Mode SDC against Individual Modes SDC.



Another area where the equivalence checking is particularly useful is when design team uses merged mode SDC to speed up the development process by merging several modes into one single SDC file, thus ensuring that the timing is met for modes of the chip. – Users can write the merged mode SDC manually or use the Excellicon capabilities to generate merged mode SDC. – In order to ensure confidence in the timing constraints of the resulting merged mode timing constraints, the designer can then perform equivalence checking against the individual modes of the design. A very unique check which allows designer much flexibility.

5. F2H – Flat SDC against Hierarchical SDC (with ETMs or with Black-Boxed components)

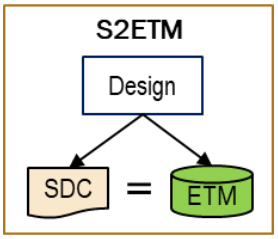




Often, chips are timed hierarchically and situations may arise when the hierarchical SDC may not correlate to the flat sign-off SDC. In such situations, TEC can be performed for flat SDC against hierarchical SDC,

with other blocks represented as ETM's or even black-boxes.

6. S2ETM – SDC against an ETM Model



Many times the ETM of an IP may not fully correspond to the SDC. Timing arcs may be missing or incomplete. Clock definition mismatch may occur which leads to incorrect timing analysis. The S2ETM equivalency checks for all such issues, thus ensuring that the ETM matches the associated SDC.

The Equivalency checking is part of the comprehensive ConCert verification platform that can be used independently or in conjunction with other Excellicon features such as timing constraints budgeting or constraints promotion.

Equivalence checking done properly as designs progress will result in significant reduction in unnecessary iterations, as well as reducing the overall time performing timing closure. Additionally, the analysis will reduce the risk of design failure and ensures highest quality timing constraints before tapeout.

Excellicon solution provides the fastest and most comprehensive approach to TEC independent of any timing engine for highest level of confidence in the analysis. Excellicon provides multi-CPU parallel execution option for most efficient execution of the analysis scalable as designs size grows.

About Excellicon

Excellicon is an innovative provider of end-to-end Timing Constraints Analysis and Debugging solutions for the automation of constraints authoring, completion, and validation from RTL to GDS with innovative analysis and debugging infrastructures. Excellicon products CONstraints MANager, CONstraints CERTifier, ConCert-BT (Budgeting Toolbox) and ConCert-ET (Exceptions Toolbox) address the needs of designers at every stage of SOC design and implementation in a unified environment. – Timing Closure; Done Once! Done Right!

For further information contact:
Rick Eram
www.excellicon.com