By Rick Eram, Sales and Operations VP, Excellicon, Laguna Hills, CA

Migrating from Manual and Validation Based Timing Constraints Development to Automated Synthesis Model Extracted from Design

It wasn't that long ago when engineers were laying gates by hand and creating chips by assembling layers of gates next to each other. Many of us think of it as a very distant past, and can't even imagine doing such a thing or even think of tackling a design using such primitive approach.

In reality this same manual methodology is what is exactly happening in the world of timing constraints today, essential step to closing timing when performing Static Timing Analysis. The advance of technology nodes has led to very dense and extremely high gate counts, which in turn is making the development of timing constraints increasingly crucial and critical during implementation stage of design. Additionally timing driven placement and routing tools require high quality timing constraints files to come up with an optimum and routable placement.

The methods used today for creating timing constraints for the most part have not changed much since the advance of automated and timing driven Placement and Routing tools. As we crossed 180nm, 130nm and then 90nm, the timing constraints became an integral part of timing closure process specially when the tools required solid timing constraints files to address SI effects, lower leakages, process corner effects, improved power consumption, and higher speeds. Surprisingly the engineers have continued to create and manage the timing constraints files by hand. The process that is based on manipulating individual timing constraints files (which we call "file based") has been the norm. There has been little change in creating better methodologies or practices on the timing side of the design closure as compared to the functional aspect of design closure and validation, which has experienced significant, advances in techniques and methodologies.

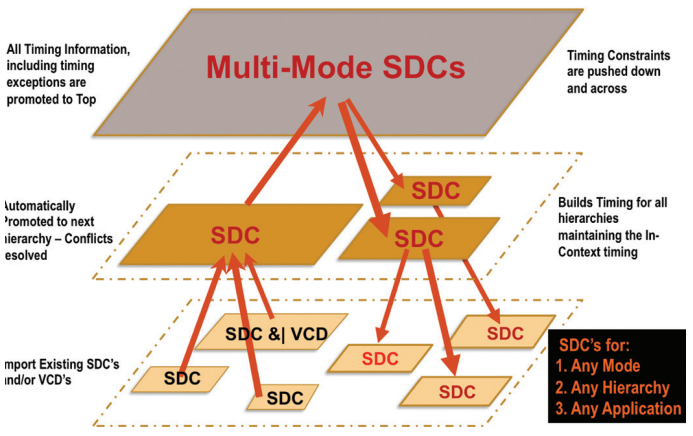
The EDA industry response to date has been a patch work of incremental tools to address the issues faced by designer in validating and manipulating hundreds of timing constraints files as designs have grown in size, complexity and layers of hierarchy. Many CAD teams have even resorted to development of internal solutions, stringing a set of complex tools, scripts, and spreadsheets together to create a methodology to address the pain. While the pain was growing the designers justifiably demanded more validation and verification, and EDA industry responded with various forms of linting and checkers offering medicine to suppress only the symptoms as opposed to offering a fundamental cure to the disease. We all quickly discovered that more validation meant also higher risk of errors, as it helped to enable incomplete and manual promotion and demotion of timing constraints as well as manual validation of various modes. Interestingly none of the available EDA tools made a significant impact in the design flows and methodologies, and the issues faced by designers kept growing while the available tools filled only a small gap in the flow. This is evident by lack of EDA tool adoption in the market place where there has been no leader in the timing constraints space.

The need for synthesis of timing constraints is becoming very clear as we race towards 10nm and even 7nm. It is simply impossible to manage 3 or 4 layers of hierarchy with hundreds or even thousands of clocks and hundreds of IP's, along with multiple modes of operations a modern SOC embodies. This is all before even trying to account for impact of tens of process corners into the final timing closure as required by advanced technology nodes.

AUTOMATED SYNTHESIS APPROACH

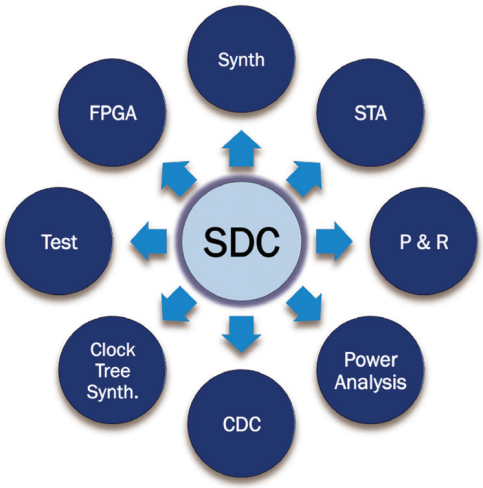
In order to manage timing constraints the timing data needs to be all available in an easily accessible database which can then be manipulated automatically as opposed to the traditional "file based" approach of managing and editing hundreds of timing constraints files by hand, to then be forced into verifying each one by one.

Excellicon's approach to addressing the management and creation of timing constraints is based on a database of timing constraints, which enables correct-by-construction of constraints hence reducing the validation cycles. The use of database essentially eliminates the problems of constraints promotion or demotion as the data is made available for any



layer of hierarchy. Once the tool reads the design in, the entire clocking and control structure of the design is deterministically abstracted for examination by the designer. Even existing timing constraints files and/or VCD's can be used as seed input to provide additional information to the process.

In order to simplify the initial inspection of the design; a painful process using traditional validation based methodologies, the user will simply be presented with all the clocks, clock relations, and mode information extracted formally and directly from the actual design (RTL and/or Gates) at the initial tool execution, eliminating mismatches between design and timing constraints completely. Through use of Clock-map, SDC-Map, and Clock/Mode analyzer, designer completely eliminates the need to draw clocking structure manually to decipher the operations and structure of the design, which inevitably are all the information needed for any down stream tool such as synthesis, vector-less power analysis, or clock domain crossing analysis.



SDC is de-facto standard as seed input to many tools & applications

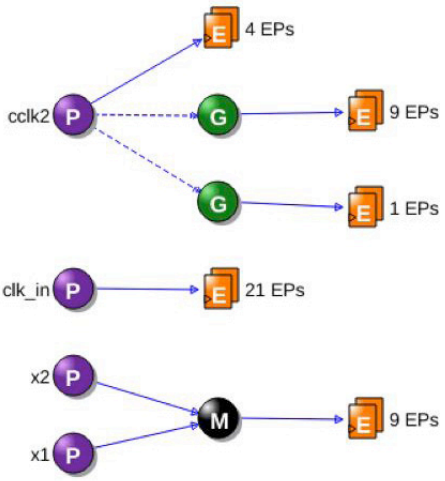
Above mentioned approach enables Excellicon tools to be able to provide many capabilities when it comes to promotion, demotion, or even budgeting. In fact Excellicon provides three techniques for each having visibility over the entire design at all times, as opposed to the old "file based approach".

Use of a new tool such as Excellicon may seem as a disruptive approach to the existing methodologies, however once the risks, magnitude of validation work, necessary manual manipulation involved, and the productivity gains are taken into account, there is simply no tradeoff off. Promotions or demotion can be performed in seconds, and the modes can be created as easily as just adding a new mode to the initial clock/mode analyzer setup.

VERIFICATION AND DESIGN REVIEW PRODUCTIVITY

Once the all the necessary constraints files are generated and made available to the back-end designers, the designers may need to check for equivalency or check for SI impact, which is why Excellicon provides a comprehensive verification product line for validation of existing timing constraints files. In fact in some cases the designer may even choose to use older versions of constraints files or ones delivered by IP vendors in which case they can very easily validate the constraints through many visual capabilities and extensive reports. The constraints can be overlaid on top of what is extracted from the design to analyze coverage and examine SDC files for any missing constraints.

Back-end designers can also perform very complex analysis for timing budget tradeoffs. There is not a comparison to visually inspect timing budgets and review the design issues.



CONCLUSION

The SOC complexity made possible by advanced nodes has had a significant impact on importance and complexity of timing constraints files. As a result manual authoring, manipulation, and validation of such files is becoming a huge challenge to the designers. Design managers are also realizing the impact and cost of bad or incomplete timing constraints on schedules through out the entire design flow, as the issues are no longer isolated to any specific stage of the design. Synthesis of timing constraints is essential in today's complex SOC's to streamline the creation and validation process prior to performing timing analysis. Manager must also ensure that the timing knowledge available in the early design stages is not thrown away and they are properly captured and propagated forward to the implementation stage, to the back-end designers so they do not end up iterating back and forth to find information for validation of constraints such as timing exceptions, or budgets from front-end designers.

Using an End-to-End methodology ensures capture and propagation of timing information during every stage of design while eliminating many manual steps in the process and reducing several weeks of work to hours.