A Day in the Life of a Crossfire User

A Fractal whitepaper

Introduction

This whitepaper illustrates how an IP-designer can be assisted by the Fractal Crossfire tool during a typical working day. Crossfire is the tool of choice for IP- and library-designers as well as users to ensure that all views, models and databases are consistent, complete and viable. The following is a quick illustration of these criteria by means of an example:

- Consistent: The condition of a timing arc in Liberty should match the functional model in Verilog
- Complete: A terminal shape with a label should be present in LEF and layout views for all schematic pins
- Viable: Power arcs should demonstrate consistently increasing power with increasing output load, operating temperature and supply voltage

By reviewing typical daily use, we demonstrate how the versatility and ease of use of Crossfire leads to both a higher quality design at the end of a day as well as a boost in productivity. As Crossfire detects IP quality issues before they become a problem, less time is spent chasing IP consistency issues introduced by suppliers or in-house design-groups.

Our viewpoint is that of a fly on the wall in a large fabless design house that produces mixed signal designs for the mobile market. Our host is a seasoned designer called Jim (username tkirk) who combines certain boldness to IP design with a firm footing in design-quality. Their current project, the Enigma II design, is a next generation of a previous successful design that is now ported to a smaller process node and enhanced with additional interfaces that were previously off-chip.

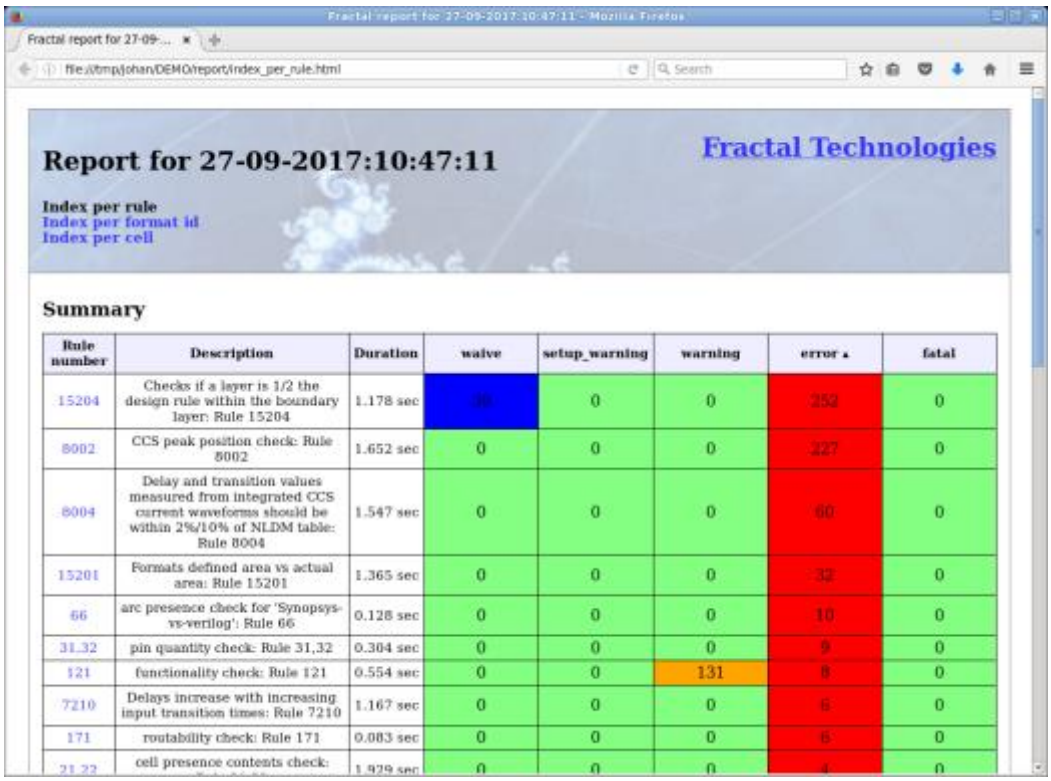
Arrive in the office: check the nightly regression

After settling down with a large cappuccino in his cubicle Jim's first action is to check up on the nightly Crossfire regression runs. The design of the Enigma I had been plagued by several last-minute hiccups that almost caused the team to miss their deadline. Luckily everything worked out fine after a couple of late night debug sessions pouring over LVS and power verification reports. The worst of it all was to notice that all these issues could have been found and fixed much earlier in the design process. Noticing that some pin-label in an OASIS view is missing or that some rarely used corner-case .lib file specifies an incorrect related power pin would not have required the whole Enigma to be assembled before this could be detected.

These issues had let the team to evaluate the use of Fractal Crossfire in the short window of time between the completion of Enigma I and the arrival of the specs for the follow-up version from marketing. It was found that Crossfire could spot these anomalies, and many more, on IP blocks before they were integrated and used. As a result, for Enigma II nightly

regression Crossfire runs are now deployed that verify every individual IP block as a standalone entity.

The Crossfire QA dashboard this morning was showing a new issue with a USB PHY that had popped up with the latest revision.



Rule number	Description	Duration	waive	setup_warning	warning	error ▲	fatal
15204	Checks if a layer is 1/2 the design rule within the boundary layer: Rule 15204	1.178 sec	0	0	0	252	0
8002	CCS peak position check: Rule 8002	1.652 sec	0	0	0	227	0
8004	Delay and transition values measured from integrated CCS current waveforms should be within 2%/10% of NLDM table: Rule 8004	1.547 sec	0	0	0	60	0
15201	Formats defined area vs actual area: Rule 15201	1.365 sec	0	0	0	32	0
66	arc presence check for 'Synopsys-vs-verilog': Rule 66	0.128 sec	0	0	0	10	0
31,32	pin quantity check: Rule 31,32	0.304 sec	0	0	0	9	0
121	functionality check: Rule 121	0.554 sec	0	0	131	8	0
7210	Delays increase with increasing input transition times: Rule 7210	1.167 sec	0	0	0	6	0
171	routability check: Rule 171	0.083 sec	0	0	0	6	0
21,22	cell presence contents check:	1.929 sec	0	0	0	4	0

Figure 1, Crossfire HTML dashboard

Sure enough, not only was the USB PHY flagged as containing an inconsistency (different power arcs between Verilog and Liberty), also the nightly synthesis run of Enigma II had failed. Instead of softly cursing and foreseeing a morning wasted on debugging the issue, Jim was able to rely on Crossfire's built-in issue visualization, the Diagnose tool, to quickly narrow down on the root cause. By double clicking on the issue flagged in the report, the two Verilog and Liberty files are instantly opened at the right location allowing to easily see a delta between the two models. In this case all that's needed on Jim's behalf is a screenshot and a message to the USB team prompting them to sort it out. As all relevant filenames and line-numbers are indicated the message is simple: USB-team: do solve!

Crossfire is a big time-saver here as both teams immediately understand what the issue is and can use Crossfire diagnostics to extract the root cause. Gone are the days where long discussions were needed to convince the other party that an issue was present at all, what was causing it and what consequences this might have.

Morning design work

Having dealt with the routine tasks and the usual flood of emails, our hero has plenty of time left for some actual design work. The ambition today is to complete the Liberty power model

of the custom-designed controller block. Yesterday's SPICE runs had proved the validity of the intended circuit, and while the layout-designers are now busy to do their custom design magic on it, it's up to Jim to complete the necessary additional Liberty modeling. Keeping track of 7 different power domains and close to 100 power terminals is a daunting task. There's a high risk of omissions or typos, however strict naming schemes one tries to adhere to.

Luckily these are the issues that are easy to resolve with Crossfire. When feeding it the respective SPICE and .lib files, Crossfire can crosscheck the correctness of the power-arcs and power pin-attributes. Again usability is crucial here: a failed check like "Incorrect related power pin VDD_S_67 for terminal RCN_6" is not helpful unless it is accompanied with a detailed illustration of what is actually wrong. By double clicking the message, Crossfire automatically starts the SpiceVisionPro viewer that traces the signal from terminal RCN_6 through the SPICE hierarchy to a different power terminal.

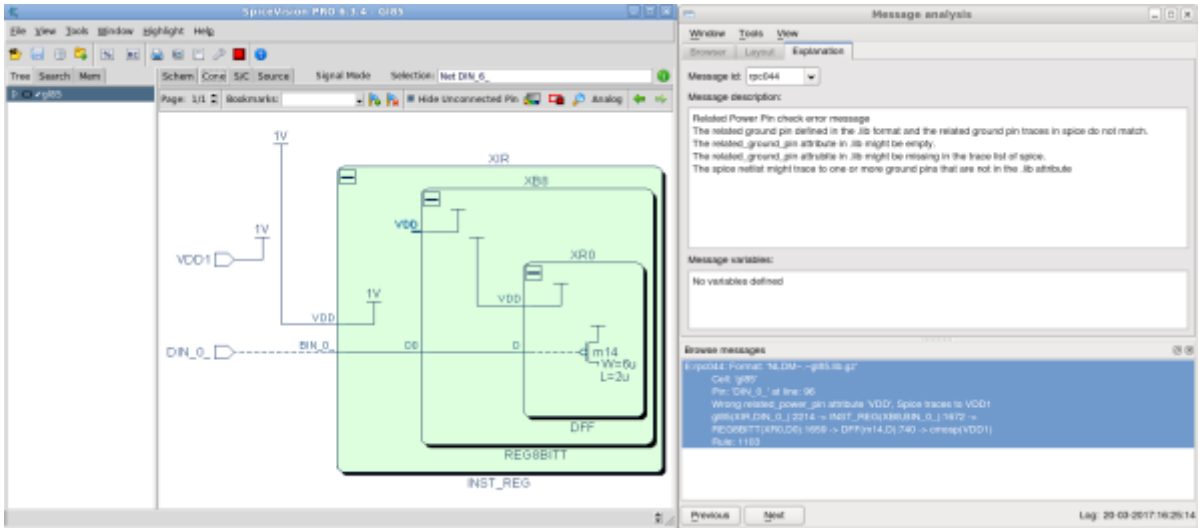


Figure 2, Crossfire SpiceVision Pro error diagnostics

"Thank you Crossfire" Jim silently whispers as this issue was actually a schematic mistake, not caught by the SPICE simulations as the necessary input condition was not yet added to the input vectors. Jim implements the schematic fix and quickly sends of a new revision to the layout-team for them to implement.

After lunch: a new foundry library has arrived

As the manufacturing process for Enigma II is maturing, their preferred foundry has shipped a new version of the IO library and cell-library to be used for synthesis. Part of the Enigma I learning was to always run sanity checks on new IP before inserting it in the synthesis flow. This incoming inspection on foundation IP is one of the highlights of Crossfire. Not only is Crossfire able to run sophisticated checks on the views that have been received, it can also perform these on the terabytes of data that are required these days to represent the 100's of process corners. To achieve that, Crossfire relies on a dedicated computational architecture deploying data servers that provide the various verification tasks with the necessary data. That way Crossfire makes optimal use of the available HPC computing infrastructure, and

does not need to rely on sampling for giving a thumbs-up/thumbs-down verdict on the library but instead is able to perform an exhaustive verification.

A consequence of this thoroughness is that the volume of issues is also huge once an error has been detected. If Crossfire had left it at merely flagging the issues it would be hardly useful as no human can deal with 1,000s of errors. Jim could of course pick one, dive into it, implement some fix and count on his luck for this fix to resolve also the remaining violations of the same rule. Today's library shipment contained the perfect example: for incoming inspection an upper limit on terminal capacitances was imposed, yet the new shipment violates it on several occasions, yet by a slight margin. Is this a cause for concern or should it be accepted?

This is where the Crossfire error-fingerprint visualization helps out. This feature can visualize all components contributing to a type of failure, like cell-class, arc-type, etc. in a single image.

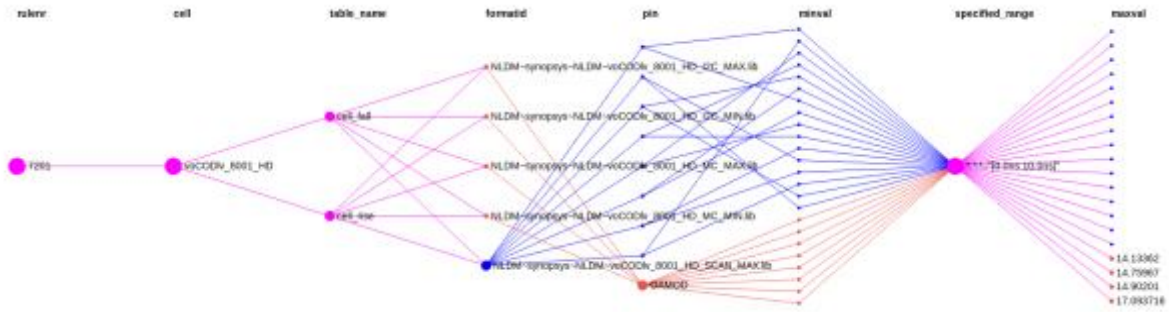


Figure 3, Example Crossfire error fingerprint

By looking at the image it is instantly clear that a couple of cells have increased capacitances and only by a small margin. By increasing the limit Jim is able to approve the new release. Gone are the days where boldness was required to accept such a release hoping nothing fatal would emerge during final timing verification.

Next: deal with second source fab

The success of Enigma I had caught their company a bit by surprise, so much that total revenue in the end was limited by amount of silicon the foundry could supply. This time they wanted to be prepared for success so had engaged with a second source foundry. For the design-team this meant that they would synthesize the entire design in shadow-mode for the second source foundry. This of course was not feasible with the same effort the team invested in the primary process.

To cope with this Jim had organized that IP quality inspection would be performed by the second source foundry before shipment. Crossfire provided them with the means to do this in the form of the Transport format. Transport is a language to specify the quality requirements the design team expects IP to meet before they start designing with it. Outcome of the Crossfire evaluation after Enigma I was a set of checks and limits that captured their design style and which were shipped as a Transport file to the second source foundry. They now run

Crossfire for every shipment they want to make and provide the team with an embedded Crossfire report and check-results database.

To Jim this was feeling quite natural and also restored some balance sometimes missing in the relation between foundry and design-teams. Foundries provide designers with DRC decks that need to be respected, Transport is way for a design-team to communicate quality rules to library and IP suppliers.

Today's mailbox revealed a new memory IP shipment by their second source foundry, neatly accompanied with a Crossfire check database as agreed. When browsing through the checks for the new memory shipment it was clear that many improvements had indeed been made since last time. But also a couple of new check violations had appeared: two of the setup times for the wordline inputs of the memory were not behaving according to the trend for several extreme process corners. Interestingly, the memory characterization guys had waived this. "It's a bug Jim, but not like we know it", he thinks to himself as he looks at the waive details.

A waive in the Crossfire world is like a temporary approval of a rule violation. The benefit is that a quality-run appears clean, where exceptions are coded as waives for known accepted issues with certain cells. When applying a waive, users are expected to enter an explanation as well as a time limit. Also in this case the foundry characterization team had done their homework, explaining that the setup time trend mismatch was something they needed to further investigate but did not want to withhold the memory release. Expected lift-date for the waive would be within a month time, and before that another revision was expected.

From a previous employer Jim had dealt with similar qualification schemes performed by suppliers. But without the appropriate tools like the Transport specification format and the waiving mechanism, he had seen suppliers changing the limits of the checks themselves just to be able to allow them to deliver "clean" IP. Not so with Crossfire and Transport where it's the end-user that remains in control and can accept or reject what a supplier is providing. Just like a foundry can accept or reject DRC rule violations in a GDS.

On the way home: call with Fractal

Jim had been noticing that their increasing reliance on Crossfire was giving rise to less and less fires flaring up just before he planned to end his working day. Also today was no exception, which meant leaving his cubicle on a relatively civilized time. The downside of that was heavy traffic but it would still get him home in time to join his kids for their soccer training.

From the car there was plenty of time to do his monthly call with the Fractal support team. During these calls tool usage issues, new checks or modifications could be discussed and the Fractal team had a chance to update him on planned new releases. On the agenda today was the latest twist on the litho-friendly rules for their custom layout blocks. The refinement they needed in Crossfire was apparently already planned for the "the next release". It was clear that also other companies using Crossfire were dealing with the same issues, Crossfire was well on its way to become the de-facto standard in IP qualification. Not because of some standardization board, but simply because of the amount of users.

Conclusion

This working day has shown various examples of IP quality issues that can be detected by Crossfire, although it's only small snapshot from the 200+ checks and close to 50 formats supported by the tool. Deploying Crossfire as early in the design flow as possible ensures issues are found long before they can become a real danger to the design schedule. This by itself is already a big time-saver as the sooner a bug is found, the less time needs to be spent on fixing it. Next to that, the ease-of-use of Crossfire illustrated with the dashboards, issue diagnostics and fingerprint visualization, simplifies the debugging and classification of issues, thereby freeing up valuable time for designers to work on their core-business: designing IP.