Crossfire Power Domain Modeling Checks and the Need for Incoming Inspection of Semiconductor IP

A Fractal whitepaper

Introduction

Success of most advanced SoC designs is to a large extent determined by battery life and thus power consumption of the design. Using an advanced manufacturing technology helps in that supply voltage and leakage current is lowered, but most crucial is to build the design such that only those devices required for the task at hand are allowed to draw current.

IP designers manage this by introducing power domains within which certain functionality is grouped and that can be activated or suspended by advanced power-control logic. The respective IP models in Liberty or SPICE must accurately reflect the power-domain design to enable final SoC power budget verification.

Correctness of this final power budget verification relies on correct inputs: the power modeling in Liberty and the implementation in the SPICE netlist. Qualifying these Liberty and SPICE models for power and timing characterization is covered by the 1100 family of checks.
This paper will describe these checks in more detail and illustrate their use and significance by reviewing several often-heard misconceptions regarding IP qualification. Phrases like “final LVS will catch it” or “if the tools read it, it’s OK” will be shown as providing only false comfort. Instead we’ll show that the only real comfort comes from rigorous power domain modeling qualification using Crossfire.

Crossfire 1100 family of Liberty vs SPICE checks

Crossfire is the industry standard tooling for validating Hard and Soft IP packages before they are integrated in SoC designs. Crossfire supports all common design formats and databases and has 100’s of built-in checks that validate the internal consistency of the various models across all characterized process corners.

The Crossfire 1100 family of checks applies to Liberty delay and power models, that describe conditions, related power-pins and power-down functions. These checks all have in common that they can only be validated by rigorous inspection and evaluation of the corresponding SPICE netlist. Only the SPICE netlist can ascertain that a condition is indeed correctly modeled or identify the power pin related to a certain IP terminal. Debugging errors reported by the 1100 checks is made easier by the integration of the SpiceVisionPRO schematic viewer by Concept Engineering that will highlight those parts of the SPICE netlist that fail to match the Liberty description, as is illustrated in the figure below.

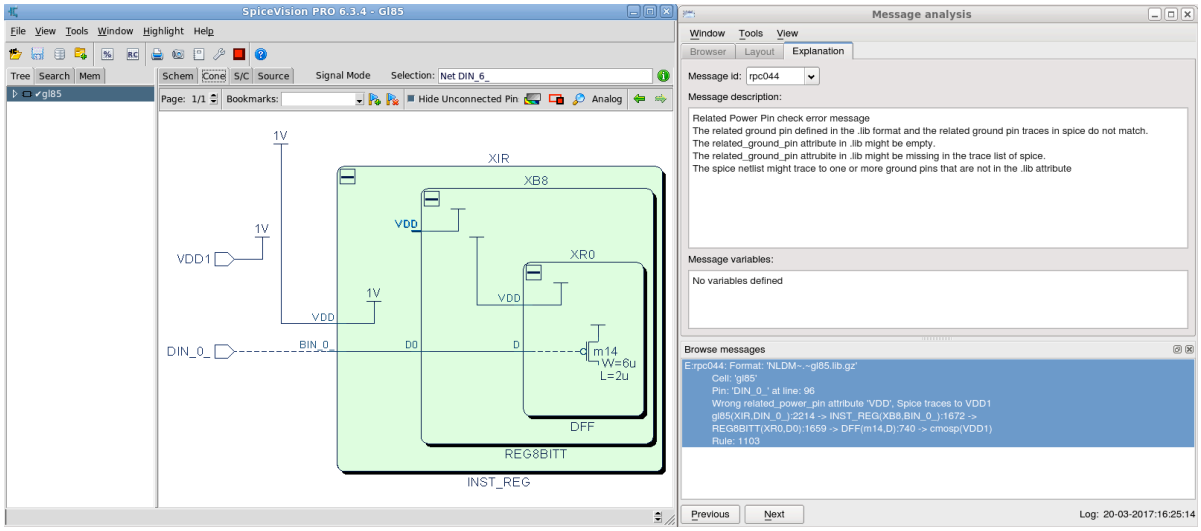

Figure 1, Crossfire visualization of failed related power-pin check

SpiceVision® PRO takes SPICE netlists and SPICE models and generates clean, easy-to-read transistor-level schematics, circuit fragments, and design documentation to speed up circuit design, circuit debugging and circuit optimization at the transistor-level.

By running an IP block though the 1100 family of checks, designers are assured that this IP won't have any power or delay modeling surprises. Skipping the validation of the Liberty IP models on the other hand poses considerable risks. Liberty models are crucial for both design-verification for timing, power and noise and serve as input for various synthesis tools. Consequently, modeling errors lead to either wrong logic being synthesized or to unreliable design-verification runs, either because of false errors or, worse, false approvals.

Yet even though the need for IP qualification before deployment is obvious, a number of myths persist in the design community regarding the need for IP validation. In the next sections the most commonly encountered myths are exposed as wishful thinking and misconceptions.

Misconception 1: "If the design-tools read it, the IP must be OK"

If the "design-tools read it" the IP modeling files are syntactically correct at best and have not been damaged during compression and file transfer. From this assessment one cannot assume any correctness or completeness on the semantics of the IP models. To see why, imagine being an EDA tool developer that will read Liberty files as input. These tools have a highly complex job to do like synthesis or timing/power verification. Tackling this is the aim of the tool, not making sure that the input Liberty file is in fact consistent with say a SPICE or Verilog model which isn't even provided as input to the tool. Tool developers must rightly assume that whatever input a user feeds into their tools is the user's responsibility and is valid and consistent.

It's thus easy to see that missing arcs, pins or conditions will go unnoticed when read by a synthesis or verification tool, leading to an inefficient design at best. A timing or power verification tool couldn't care less about a missing delay or power arc or a condition that doesn't match with the function of the cell. Instead it assumes the Liberty files model the cell

correctly and will check power-domains and delays accordingly. This leads to either false violations or undetected violations. False violations are truly the best of all evils: after extensive debugging designers will typically find the root-cause and can fix the IP Liberty modeling. Undetected violations or false positives simply go unnoticed as also final LVS will not detect those as we'll discuss later.

Failing a formal verification of these Liberty functions as performed by Crossfire Liberty modeling issues can only be detected by exhaustive transistor level simulation of the entire design, something that's hardly practical for a billion-transistor design. Consequently, the first prototypes may violate the power budget and the problem needs to be resolved through silicon debug.

In conclusion: EDA tools accepting input models without complaints are a necessary condition for bug-free design, but it's far from a sufficient condition.

Misconception 2: "IP inconsistencies are trivial"

Let's for a moment assume that many IP quality issues indeed are trivial. But when 4 TeraBytes of IP shipment has been received, making sure that it is clear even of trivial issues becomes a daunting task. This requires a dedicated QA tool like Crossfire that can handle the large data volumes in an efficient way to filter out all the quality issues, including the trivial ones. Crossfire was engineered from the ground up to deal with large data volumes and to make efficient use of the available high-performance computing infrastructure.

This triviality of quality issues is of course another myth that can come in handy to justify shortcuts when pressure on designers is high. In reality, IP quality issues can be far from trivial. Consider a Hard-IP block, like a physical interface, which may have 100 different power pins all of which have modeled power-down/select functions. Any error in this Liberty modeling, which is after all manually created, can be a power budget risk that is hard to fix. The checks for related power pins and power-down functions are known as Crossfire rules 1102, 1103, 1104 and 1108. These checks involve tracing in the (hierarchical) SPICE netlist the power connections to the transistors connected to a particular cell terminal. When a mismatch is found, i.e. a different power-pin is connected in SPICE as is specified in Liberty, the user needs to be presented with the means to traverse the SPICE hierarchy to make informed decision on where to make corrections. Crossfire does this by configuring SpiceVisionPRO by Concept Engineering to display a net-trace through the SPICE hierarchy, allowing users to follow the signal through the various blocks and to descend deeper into sub-blocks when needed. The figure below illustrates a top-level view of such a trace for a Hard-IP block of only moderate complexity.

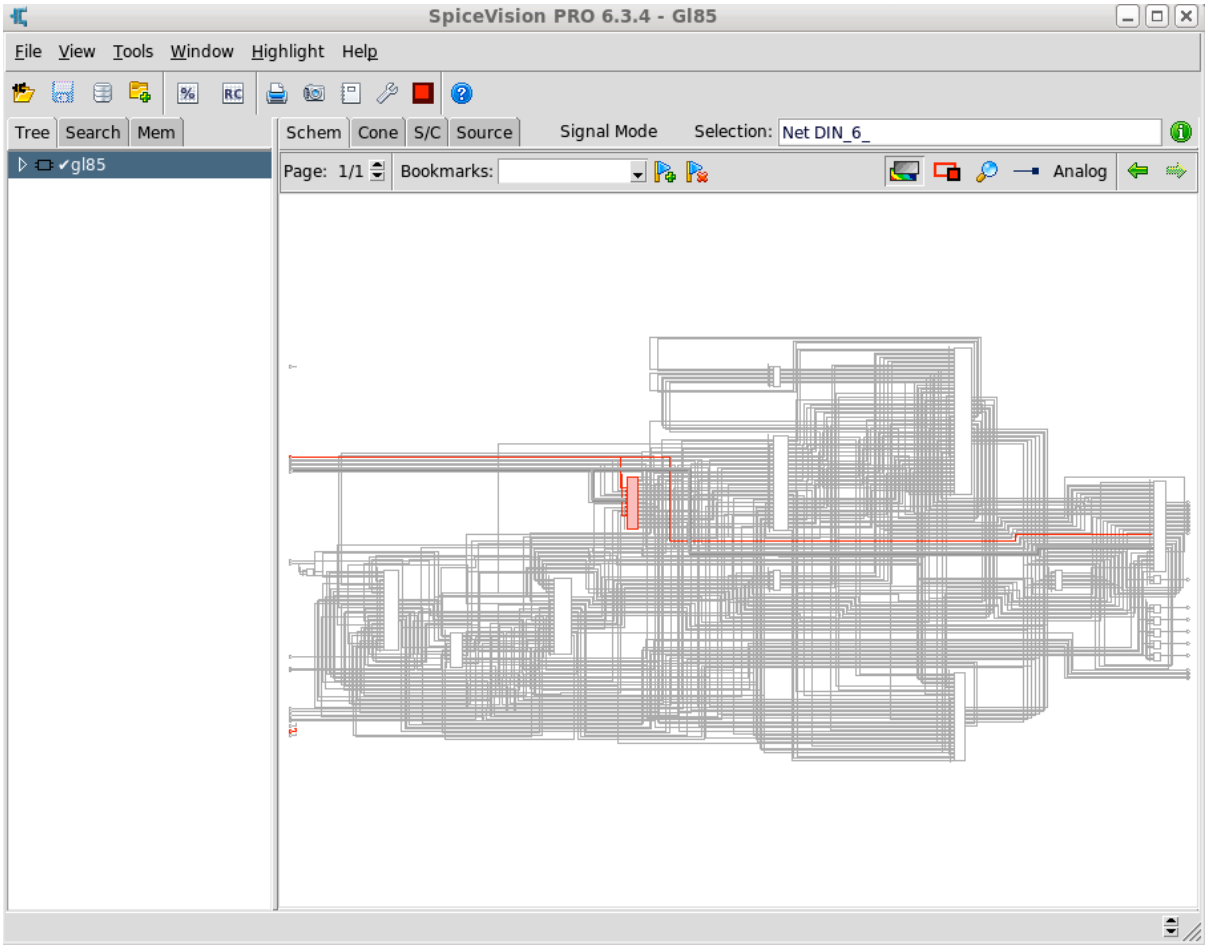

Figure 2, Trace of a power-connection signal in SPICE

As this example illustrates, such power-pin errors are far from trivial. The data-volumes (10's of GB of SPICE netlist for example) and the complexity of the designs by far exceed the capabilities of home-grown QA scripts and mandate the use of a sophisticated verification tool like Crossfire. Crossfire not only effectively detects these issues, it is also capable of guiding the user to the root-cause in the SPICE netlist.

Misconception 3: "There's always final LVS"

Indeed final design verification is named final for a good reason. Anything still wrong with the design needs to be detected and repaired before the design can be taped out. Although very reliable, there are good reasons that designers dread finding themselves in the situation of having to investigate final LVS and other design-verification issues. Obvious reason is the sheer size of the design. By definition final design verification considers the entire design, so any issue that is detected, even though it's local to an IP block, is reported in the context of the entire design. This makes debugging hard as designers first need to localize the problem, i.e. identify which IP blocks are involved, and then can dive into finding the root-cause.

At the same time the tape-out deadline is really close, but without final design verification that cannot happen. So on top of having to deal with tough problems on a full-chip scale, the pressure to get them resolved is also immense. In conclusion: final design verification should always pass as it's a nightmare to debug and a high risk to missing deadlines.

Another aspect to keep in mind is that design verification will pass designs that are correctly hooked up but still contain wrong timing or power domain modeling. Not all verification can feasibly be done by running LVS and SPICE simulations. Instead, timing and power verification relies on correct models being supplied in the Liberty files. If these are incorrect, for example when cell-terminals are linked to the wrong power pins, the verification may consider parts of the design to be switched-off and hence pass the power budget constraints. In reality, logic connected to these pins could still be drawing on the power budget because of a Liberty modeling error. Situations like these pass all final design verification, except for impractical exhaustive SPICE simulation, but in the end may considerably delay market introduction of a new device.

It's these situations that the Crossfire 1100 family of checks intends to catch while no harm has yet been done. The incoming inspection of an IP block using Crossfire ensures the design team that the Liberty actually matches the logic and netlist of the design. During incoming inspection, repair of issues can still be accommodated by the design schedule. Final timing and power verification becomes much more reliable and only integration issues remain as potential final LVS/SPICE verification hurdles.

To conclude, final design verification by LVS and SPICE simulation can theoretically catch all issues but is both impractical and a high risk to the design schedule. Instead, incoming IP inspection using Crossfire allows final verification to almost always pass.

Misconception 4: "IP coming from the foundry is by definition clean"

Let's cut this one short: Foundry IP should be clean but if it isn't, it's still your problem. Many large fabless companies have already realized this and are using the Fractal Transport formalism to exchange their IP quality requirements with their suppliers, including foundries. Where a foundry would provide a design-team with the design-rules to which the final mask layout has to obey, designers provide their suppliers including foundries with a Transport quality rule-deck to ensure that any IP that is shipped to them can be reliably used. Together with the IP shipment, the design-team will receive a Crossfire report based on the Transport checks that they provided. This report highlights what has passed and what issues may need to be waived and fixed in future releases.

It's the Transport formalism that allows design teams to rely on clean IP that is shipped to them by their foundry or any other IP supplier, as it has been verified with Crossfire.

Misconception 5: "We can deal with it ourselves"

Recognizing and dealing with IP quality issues is a 180 degree change of mindset as compared to the previous sections. Dealing with quality issues in whatever way is an indispensable first step on the road to total QA control. Many design aspects can only be covered by internal solutions, in particular those that relate to internal design-tools and design-styles. These need to be comprehensively covered by accompanying custom QA rules.

As far as standardized design tools, formats or databases are concerned, productivity gains can be made by incorporating an industry standard QA tool like Crossfire. Crossfire is the de-facto standard in QA solutions which funnels all the QA requirements from the entire design industry into a golden tool. Crossfire comes pre-packed with 100's of pre-defined, standard, checks as every user of Liberty or a layout database will have the same consistency requirements. These checks and the infrastructure behind it to process these checks in large IP shipments with acceptable runtimes can be re-used by internal QA teams. With Crossfire they can provide robust solutions for industry standard checks and at the same time use the Crossfire data management framework to efficiently implement custom checks.

Conclusions

SoC designs in advanced process nodes can only meet their power budgets by introducing a variety of power domains to ensure that only those devices needed to perform a certain function are in fact active. These power domains must be modeled correctly in the corresponding Liberty files so designers can rely on the reports of the power budget verification tools. These Liberty power models require extensive validation before IP blocks can be reliably integrated into a complete design.

A laissez-faire attitude towards IP QA qualification is no longer acceptable and puts design schedules at risk. IP received from suppliers is never clean, not even from foundries. Easy re-assurances such as successful parsing of the IP by the design-tools are myths. IP quality issues are far from trivial and if they go unnoticed can cause large delays during final verification. A professional, industry standard QA solution like Crossfire is needed to for a quality controlled hand-off between IP suppliers and designers.