AccuCore STA DSPF Backannotation Timing Verification Design Flow

Abstract

This application note highlights when and why DSPF backannotation is needed during timing verification, and details the SILVACO design flow process steps necessary to utilize this enhanced method of timing analysis. The tools to be discussed include EXPERT, HIPEX-NET, HIPEX-RC, ClarityRLC, AccuCore/AccuCore STA and SILOS along with data formats Graphical Design System II (GDSII) , Detailed Standard Parasitic Format (DSPF) , Standard Delay Format (SDF), Liberty .lib Non-Linear-Delay-Model (NLDM) and Verilog Register Transfer Level (RTL).

Background

Verification has surpassed actual design in the effort required to tapeout a modern IC design. Timing verification is a critical and at times difficult and lengthy step in the tapeout process of a circuit. Traditionally, a Verilog gate-level netlist would be simulated in an event-driven logic simulator such as SILOS with a set of vectors from a user test-bench together with a set of verilog gate-level timing models to validate that both function and timing requirements were met.

As process technology migrated from sub-micron (i.e. >=0.5um), where gate capacitance >> wire capacitance, down to Deep Sub-Micron (DSM) (i.e. <0.5um), where gate capacitance ~= wire capacitance; wire loading delay effects could no longer be ignored or compensated for by simple timing margin pro-rating. Wireload estimate models and SDF backannotation was used to provide additional net based timing effect details during Verilog regression simulation.

As process technology continued to shrink and designs grew proportionally larger, gate-level timing regression simulation run-times increased significantly. Gate-level schematic design also became a significant bottleneck, but thanks to RTL logic synthesis and Liberty .lib NLDM timing models together with statistical wireload models, circuit design became quick and simple again. Timing verification based on the current vector based simulation method continued to suffer however.

Vectorless STA methods utilizing verilog gate-level post-synthesis netlists and the same liberty .lib NLDM timing models as the synthesis tools eased the task of timing verification and timing model and delay calculation correlation effects. SDF backannotation initially continued to be used to provide the necessary additional net based timing effect details. However, the shortcomings of the STA+SDF method soon began to limit design Quality of Results (QoR) and did not provide the necessary capabilities to account for the non-linear Ceffective effects of Very Deep Sub-Micron (VDSM) process technologies (i.e. <=90nm). VDSM required detailed RC tree parasitic information including actual coupling capacitance network structure to accurately account for long wire and sub-node net delay and noise/Signal Integrity (SI) effects.

The Design Flow

DSPF is a common industry format for exchanging post-route wire resistance (R) and capacitance (C) parasitic details of a circuit's actual physical layout from GDSII format data. Static Timing Analysis (STA) tools like AccuCore STA can import DSPF data and backannotate these values into a flat gate-level verilog netlist to account for actual wireloading effect details on the circuit's timing, and noise/SI behavior. The verilog netlist alone only provides the basic LVS connectivity between the gates of the design. A Liberty .lib format gate-level timing library provides the slope and load input to output timing and functional behavior for the gates in the design. Initial or preliminary pre-route timing estimates are possible with only the gate-level netlist and timing models, ignoring the wire loading and wire delay effects.

EXPERT with embedded HIPEX-RC or HIPEX-RC standalone batch mode can be used to extract detailed RC data in DSPF syntax from GDSII data which can then be imported by AccuCore STA and combined with the pre-route verilog netlist and Liberty .lib NLDM cell timing models to perform fast vectorless post-route cell-based

STA critical path identification and verification. Optional RC network reduction can be performed either during the HIPEX-RC extraction process OR as a separate post extraction process utilizing CLARITYRLC.

ClarityRLC permits the generation of additional statistical process variation skew corner netlists concurrently with state-of-the-art RLC passive network reduction utilizing user controllable separate R, L & C reduction and spreading parameters while minimizing network timing and SI effect reduction errors. This permits, from a single pass of the HIPEX-RC extraction process, quick access to multiple netlist corners and compression ratios for run-time vs. accuracy trade-offs on large designs without later loss of access to the full netlist for sign-off and audit purposes.

AccuCore STA can also handle basic noise delay effects by user-controlled ladder network conversion of coupling capacitance netlist details. Identified pruned critical paths can then optionally be exported for path-based SPICE simulation including side-input control and all stimulus and .measure statements necessary for SPICE sign-off in critical maximum accuracy applications when reference to a SPICE netlist and models is also provided.

NOTE: As an alternate option, AccuCore can also be directed to completely SPICE re-characterize a design from a transistor level RC SPICE or DSPF netlist and SPICE models. AccuCore is architected to permit independent and reversible analysis dependent coupled network ladder conversion and in-circuit active driver-load context RC reduction for maximum compression with bounded timing impact.

SILVACO Design Flow DSPF Extraction and Backannotation steps from GDSII

DSPF Extraction Steps from EXPERT GUI

1) Create/copy the <process>.r_cmd technology file in/to the design directory
2) Create/copy the <process>.c_cmd technology file in/to the design directory
3) Create/copy the <design>_lvs.net schematic LVS netlist in/to the design directory
4) Run expert 4.8.15.R (or greater) in read/write mode and load the .gds or .eld design file
5) Navigate to (**Verification->Extraction->Setup**) Layout Parameter Extraction Setup
6) Select the **Layout** tab and edit run options
7) Select the **Node Names** tab and edit run options
8) Select the **ERC** tab and edit run options
9) Select the **Cell Explosion** tab and edit run options
10) Select the **Netlisting** tab and edit run options
11) Select the **Technology** tab and edit run options
12) Select the **Parasitic Extraction** tab and edit run options
13) Select the **CRC** tab and edit run options
14) Select the **Backannotation and LVS** tab and edit run options
15) Execute (**Verification->Extraction->Hipex-Net->Run**) to create HDB database results
16) (optionally) Execute (**Verification->Extraction->Hipex-Net->View Summary**) and check for run-time errors
17) (optionally) Execute (**Verification->Extraction->Hipex-Net->View Netlist**) and review
18) Execute (**Verification->Extraction->Hipex-R->Run**) to create RDB database results
19) (optionally) Execute (**Verification->Extraction->Hipex-R->View Summary**) and check for run-time errors
20) (optionally) Execute (**Verification->Extraction->Hipex-R->View Netlist**) and review
21) Execute (**Verification->Extraction->Hipex-C->Run**) to create CDB database results
22) (optionally) Execute (**Verification->Extraction->Hipex-C->View Summary**) and check for run-time errors
23) (optionally) Execute (**Verification->Extraction->Hipex-C->View Netlist**) and review
24) Execute (**Verification->Extraction->Hipex-RC->Run**) to merge RDB and CDB database results
25) Execute (**Verification->Extraction->Netlister->DSPF Netlist**) to create DSPF netlist results
26) (optionally) Execute (**Verification->Extraction->Hipex-RC->View DSPF Netlist**) and review
27) (optionally) Execute (**Verification->Extraction->ClarityRLC->Run**) to create RC reduced DSPF

NOTE: Any run options not discussed in detail below can commonly be left at their default settings. Some settings are dependent upon the structure and form of the <design>_lvs.net file and the GDSII. Only settings required for the example illustrated are discussed and representative of the recommended method. Other methods and settings may also be possible and appropriate for alternate design flows. Additional notes have been added to highlight areas where deviations from the common assumptions may result in design flow difficulties.

Layout tab run option explanations

Top Cell – defines the top level physical hierarchical database cell to be extracted which should be selected from the pull-down menu list

NOTE: The design is assumed to consist of only the top level cell and leaf level gate cell primitives. If your design contains an alternate structure it is recommended that a copy be altered to match that assumed.

Node Names tab run option explanations

Global node names (Ground, Power) defines the top level global power and ground name respectively that are assumed to match the LVS netlist.

NOTE: The design is assumed to have a single global power and a single global ground. If more than one of each/either exists, it is recommended that the design be extracted hierarchically bottom-up as separate voltage domains and then analyzed individually if reasonably feasible. While it may be possible and acceptable to extract the whole design in a single pass when it contains multiple global power domains, care should be taken to ensure that downstream analysis tools can accept and analyze netlists of that type.

ERC tab run option explanations

NOTE: Open node (Rename opens) may cause LVS difficulties for some tools, this is however the recommended default for the SILVACO design flow.

Cell Explosion tab run option explanations

The top level physical cell and ALL leaf level gate physical cells should be configured as HCELL. ALL contact or via physical cells should be configured as SMASH. ALL intermediate hierarchy physical cells should be configured as EXPLODE.

NOTE: Where possible it is preferable to pre-configure a copy of the layout to match the previously mentioned preferred physical structure (i.e. top + leaf only) with no additional GDSII SREF in the design. Pre-smashing and pre-merging many small objects and arrays **MAY** improve RC extraction performance in large designs, if multiple extracts are anticipated, thereby reducing the total number of database objects to be processed. This is especially true is situations where many data overlaps/enclosures exist.

Netlisting tab run option explanations

Coupling mode – enables the extraction of coupling capacitance IF the c_cmd deck is programmed to do so.

Generate SPICE file – enables the creation of a SPICE format netlist file.

Generate DSPF file – enables the creation of a DSPF format netlist file.

Generate SPEF file – enables the creation of a SPEF format netlist file.

NOTE: Parasitic capacitor netlist (Coupling threshold) default is 0pF (i.e. ALL node to node coupling values permitted by the extraction rule set deck). In some case this may be highly excessive and increase run-time unnecessarily. For some designs setting this parameter to a suitable value to pre-filter the netlist during C extraction may be appropriate and beneficial. It is generally recommended to use the default (disabled) at the expense of run-time. Set this option if the design AND/OR the analysis mode, make inclusion of small coupling capacitance values unnecessary or unwarranted. Filtered coupling capacitance C values are converted to ladder equivalents to ground for each node of the coupling capacitance.

NOTE: The three Generate * file options (SPICE,DSPF,SPEF) do NOT normally significantly increase the overall run-time. It is therefore recommended that all three be enabled to avoid the potential for the need to later re-run RC extraction if a needed file type is missing. Disable unneeded output formats (SPICE,SPEF) ONLY if you are sure that the additional formats are unnecessary and will only fill up disk space.

Technology tab run option explanations

Parasitic capacitance technology (Use external LISA script) – defines an external file containing the capacitance extraction rules for the process in SILVACO EXPERT/HIPEX-RC LISA format.

NOTE: older version files of this format may require import and save conversion to function correctly in the current version of software.

Parasitic resistance technology (Use external LISA script) – defines and external file containing the resistance extraction rules for the process in SILVACO EXPERT/HIPEX-RC LISA format.

NOTE: older version files of this format may require import and save conversion to function correctly in the current version of software.

NOTE: It is assumed that a proper EXPERT technology file has been previously imported and saved into the .eld file for at least the connectivity and device info. Derived layer info along with Parasitic capacitance and Parasitic resistance info may also source from the EXPERT technology file imported into the .eld database file. If so, no additional external files for this respective data are required. It is assumed that at least the Parasitic R (r_cmd file) and Parasitic C (c_cmd file) come from an external source.

Parasitic Extraction tab run option explanations

Net Collections, Ignored nets (Net name) – normally the global power and ground nets should be added to this net group. Add any additional nets that should NOT be included in the RC extraction process. Settings in the

Net Collections group ONLY affect RC extraction; LVS netlist extraction is NOT affected and is handled by HIPEX-NET during netlist extraction processing.

NOTE: unless performing selective or incremental RC extraction for a detailed analysis of specific critical nets ONLY the Net Collections options as defined above should be utilized. Alternate design flow methods may necessitate altering some or all of these settings as appropriate.

Capacitance Extraction

NOTE: the GDSII is assumed to include the entire design of interest. If the physical database is only partial or contains additional "cropped" perimeter objects Include dangle nets may be relevant to net loading on some nets.

RC Reduction – enables concurrent RC reduction during the extraction process of the original RC output netlist.

NOTE: RC Reduction is an option in this tab however it is recommended that it NOT normally be used for the main initial RC extraction unless it is known prior that no impact to quality of results will occur for critical path analysis. Separate and independent RC reduction and skew corner generation can and should be utilized from the CRC tab thereby preserving the option to later return to the original unreduced RC netlist without the need to re-extract.

Specify number of processors (Parallel Mode) (Max=4) – Enables the speeding up of RC extraction my means of parallel processing. NO accuracy will loss result from the use of this option.

NOTE: This parameter should be set not greater than the number of actual CPU cores in the hardware being run on. Do NOT count threads or virtual CPUs on hardware with hyperthreading enabled. See the SILVACO general recommendation regarding disabling hyperthreading on hardware primarily for running SILVACO software.

CRC tab run option explanations

This tab controls the optional running of ClarityRLC RC reduction as a post-RC-netlist generation process.

Input/Output, Format – select DSPF

Reduction (Coupling mode) – enables the preservation of coupling capacitance structures in the output RC netlist. If this option is not selected the RC output netlist will contain ladder-converted equivalent values if coupling capacitances are present in the input RC netlist.

Thresholds – Defines cut-off values for R and C from which to reject in the final output netlist.

NOTE: separate R and C thresholds for either all cells and each cell individually can be defined under this dialog. The most common use of this feature is to set different values for the top level separate from the leaf cell level if desired.

NOTE:If multiple process skew corners are desired in addition to RC reduction, it is necessary to run ClarityRLC in stand-alone batch mode separate from the EXPERT GUI.

Backannotation and LVS tab **run option explanations**

Backannotation and LVS (Backannotate) – Enables the use of an LVS backannotation SPICE netlist to output the RC extracted netlist including the schematic net and instance names.

Schematic file – Defines the LVS schematic backannotation SPICE netlist from which to reference for net and instance names for the extracted RC netlist.

NOTE: Failure to set and properly define a suitable matching structure (and net and instance naming) LVS SPICE netlist will result in the subsequent failure of the downstream DSPF backannotation during the STA process.

AccuCore STA DSPF Backannotation Steps

1) Create/copy the <design>_sta.tcl STA run script in/to the design directory
2) Create/copy the <design>_sta.cfg STA configuration file in/to the design directory
3) Run accucore <design>_sta.tcl |& tee sta_log
4) Review sta_log file for errors and warnings
5) Review *.rpt timing reports (debug, checks and paths)

AccuCore STA .cfg file configuration run option explanations

inputs – specifies top level input ports (except clocks)

outputs – specifies top level output ports

inouts – specifies top level bidirectional ports

clocks – specifies top level input clock ports

powers – specifies top level global power ports

grounds – specifies top level global ground ports

in_snps_lib_name – specifies the .lib Liberty cell timing library file(s) (space separated list)

in_vlog_netlist_name – specifies the .v verilog netlist file(s) (space separated list)

in_spf_name – specifies the .dspf DSPF RC parasitics netlist file

input_time – specifies the input port to clock timing an slope relationship

NOTE: clocks may be internal nets in addition to top level ports.

output_time – specifies clock to output port timing relationship (setup/hold)

clock_time – specifies clock slope and timing characteristics

NOTE: clocks may be internal nets in addition to top level ports.

NOTE: additional configuration options exist in AccuCore STA. Only the most common commands are identified above.

AccuCore STA .tcl file analysis run option explanations

sta_read_cfg – specifies the STA .cfg configuration file

sta_report_file – specifies a .rpt report file to output analysis results to. Common report groups are (debug, checks & paths)

sta_verify_netlist – controls reporting netlist debug information

print_clock_waveforms -all_nets – controls reporting clock debug information

report_warnings -all – controls reporting STA process debug information

verify_checks – specifies details of the STA timing constraint analysis to be run

report_checks – controls reporting of STA timing constraint analysis results

find_paths – specifies details of the STA critical path timing analysis to be run

report_paths – controls reporting of STA critical path timing analysis results

NOTE: additional analysis options exist in AccuCore STA. Only the most common commands are identified above.

For additional information regarding the run options described above and other available please refer to the Guardian Verification User’s Manual Chapter 3.

For additional information regarding basic use of EXPERT please refer to the Expert Layout Editor User’s Manual.

For additional information regarding HIPEX-NET and HIPEX-RC please refer to the HIPEX User’s Manual Volume I & II.

For additional information regarding CLARITYRLC please refer to the *ClarityRLC User’s Manual.*

For additional information regarding AccuCore STA please refer to the *AccuCore STA User’s Manual* and the *AccuCore STA Command Reference Manual*.