Insight Analyzer
v5.21
Release Notes

5.21-13339
2022 October 14

Contents

Overview

This version of Insight Analyzer is a refinement release based on the features established in v5.20. This release includes bug fixes as well as additional functionality based on already established features.

While the increment in numbering is relatively small (v5.21 vs v5.20), the scope of updates in this release that actually impact usability and quality are significant. Users of any prior version of v5.x series are advised to adopt this version without hesitation.

As an overview, the updates in this version include the following highlights:

- **GUI**: Extended capabilities in the Power & Boundary editing workspace.
- **Power states**: Increased flexibility in making power state definitions, and handling the overlap between different state related checks.
- **Float and Contention checks**: Numerous extensions to support additional scenarios and increased scope of checking.
- **Quality and field input**: Recent software updates have been based on feedback from a diversity of field cases, increasing quality of both the checking of circuits and the reporting of information.

Updates & Changes

Circuit Loading

Handling of Backslash '\'

Starting in this version, backslash characters are removed from cell names when parsing all spice-like netlist formats. Previously, this was done only for Spectre formats.

Enhanced Logging

The following additions are made in the main process log:

- Load time summary, added after all loading and definitions are completed.
- Details on addVfiltMark merging. The process of merging multiple net tags through resistor shorts (links) can expose conflicts in user's marking or definitions. This process outputs details to the log.
- Additional info on problem instances during netlist parsing, such as when missing a cell definition or when all pins aren't connected.

Net Properties

Applications that depend on net properties (or "attributes", cdb net addAttribute) may have experienced slow run times in prior versions. This version uses significantly faster and more efficient storage of net properties, for user-created applications that may have this dependency (such applications are generally not common and do not impact most users).

Capacity

Memory footprint allowance has been increased, for cases where the number of individual declarations in a circuit netlist file(s) have gone over many billions. Examples include cases where the size of a netlist file is in excess of 20 gigabytes.

Extracted Parasitics

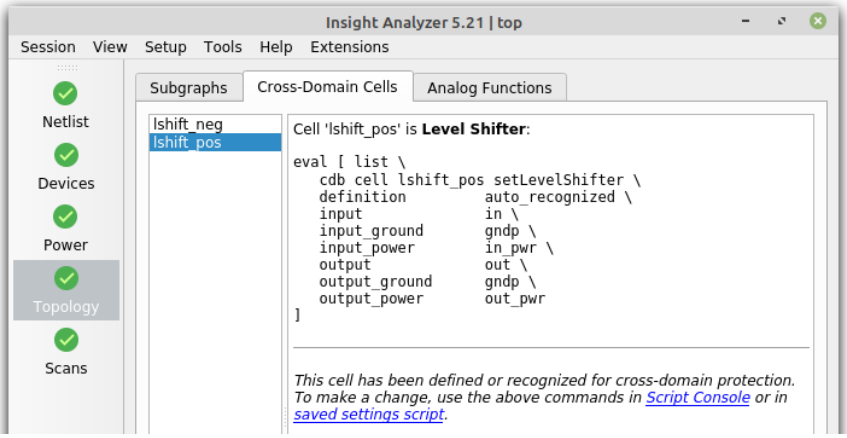
Loading extracted parasitics has been made faster, with optimized internal storage of large SPF / DSPF / SPEF data sets.

Recognition

Isolation & Level Shifters

The recognition of isolation cells and level shifters has been further enhanced, following input from users with new cases, many of which are novel topologies.

In addition, isolation and level shifter cells are now outlined clearly in the GUI, in a format that can be directly copy / pasted into steps scripts with any modifications that might be needed.



New format for level shifter and isolation cells, as Tcl code, in the Topology GUI.

Hysteresis

A new subgraph type has been added to support hysteresis receivers. There are a few essential variations of this pattern recognized. These apply to the Analog Float checks, and possibly other applications.

These hysteresis patterns appear in the Topology workspace, under Subgraphs. See also the script command `cdb inst isHysteresisFeedback`.

Tie-high / Tie-low

In addition to specific circuit patterns of tie-high and tie-low, this version also uses constant value to spread virtual ties through the system. This helps to adapt to new virtual tie topologies that have not been explicitly defined. Related to this are the impacts in Power Connections check, Domain Crossings check, and script commands `inst isMissingTieHighLow` and `net isTieHighLow -effective`.

CMOS & Dynamic

Updates have been made in the detection and marking of CMOS and dynamic logic. These apply to uncommon cases involving passgates, quasi-parallel stacks, intermediate nodes with power rail names, and others.

One recent example is that of a NAND-2 structure that shares a common Nfet footer switch with an adjacent dynamic logic stack. Both the CMOS NAND and the dynamic output are allowed to co-exist as distinct recognized parts.

Subgraph Generator

To assist in making new subgraph templates, Insight Analyzer is able to convert an example netlist into the proper format of a subgraph template, automatically.

You access the subgraph template generator from menu Tools > Additions > Generate Subgraph Templates. The result is a template file that can later be used to guide circuit pattern recognition, based on the example circuit you have already provided as the seed for the generator.

For more information, please see User Manual appendix J, “Template Based Circuit Recognition”.

GUI

Device Editing

There is now better handling of the selection under preferences for “show base device names”. When changing this preference setting between the two options (base device or wrapper names), the GUI devices table is updated accordingly.

Power & State Editing

Zero vs Off

This version now supports two ways of representing a power supply that is being turned off:

- Using the word “off”: Most appropriate for checking and describing violations where a power domain is off.
- Using “0” (zero) in the power state table: Sometimes assumed by a user when making entries in the GUI.

In either case, Insight Analyzer adapts to the definition style and renders the outcomes appropriate for each check. For more information, please see the User Manual section on “Power Definitions”.

Display of Replicated Rails

Where a power state table defines rows for replicated power rails (power rails that are contained in replicated cells below the circuit top level), the following should be noted:

- In GUI: The power state table is displayed in terms of cell + net, and does not show replications. A single value may be displayed in a state column, as representative of all replications for that state.
- In API & checking: The power state table internally has all detail of the replications. Saved

settings will show the complete table with all rows for replicated rails.

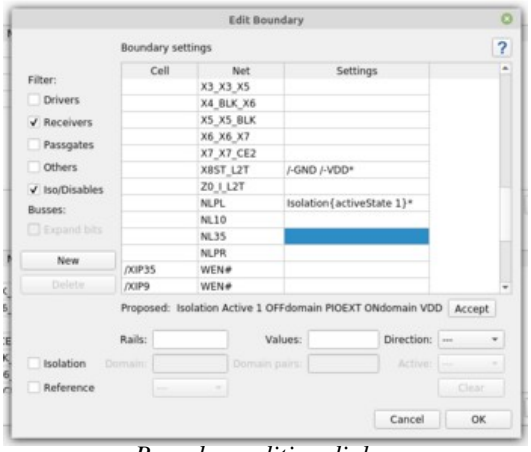
- If you do have replicated rails, definitions should be made through script (UPF or saved settings), and GUI should not be used to make edits in the table.

Boundary Editing

The editing functions for toplevel IO have been significantly updated in this version of Insight Analyzer.

Filter Controls

To help with circuits having high pin count at top level, the boundary table can be filtered by IO type. The filter controls are based only on topology of the actual IO nets (not based on “pin info” statements or user definition).



Boundary editing dialog

- Drivers: A driver output is connected to the IO.
- Receivers: A proper receiver is connected to the IO net.
- Passgates: The IO leads to transmission gates.
- Others: IO nets that do not fit into any of the above.
- Iso / Disables: These are IO nets that propagate into the circuit where isolation, level shifters, or analog disable circuits are controlled.
- Expand Bus Bits: An option to expand or collapse bits of a grouped bus.

Accept Proposed Definition

For isolation controls, which sometimes require complex definitions, a button is provided to automatically fill in the required entry fields. When an isolation control is selected in the table, the “Accept” button is enabled.

Report View

The displayed row numbers of a report will now respond appropriately to the “Collapse” control. When a report is collapsed or expanded, the row numbers are re-calculated where lines are grouped together.

Logging

Timestamps

The format of writing time stamp at start of each line may be restored by command line option -logTimestampEachLine, provided at startup.

Script Errors

In some scenarios, a running user script would have an error (such as variable not defined), and the error would not be apparent in the main log. This version now includes the output of `set ::errorInfo` in the main log, showing a detailed stack trace as a result.

Checks & Applications

Analog Floats

The Analog Float checks have been substantially updated over prior versions. Please see updated page “Analog Floats” in the Scan Reference documentation.

Field Input

The Float checks have been updated to cover numerous additional scenarios that lie somewhat beyond the scope supported in prior versions. Some of these are outlined here.

Memory & High Fan-in

Memory architecture poses particular challenges to the high-impedance checks, which look for a “hole” in a wide array of input states. New techniques have been added to recognize fundamental parts of address and control, and break up the problem solving into specific parts. This has increased the feasibility of convergence (closing the arc of control) on large cases such as those found in SRAM arrays.

Dead CMOS

When searching for upstream contributors of state, a CMOS output stage that depends on a power supply that is off (based on power state table, as appropriate) is generally considered to be issuing a logic low (rather than wildcard), for most realistic outcomes.

Switches Above Defined Rails

Prior versions of Insight Analyzer would consider a rail definition as “solid” when applied to conditional float checks. If a receiver or gate input was powered by a defined rail, it was considered as a victim for potential high impedance gate node. Any condition of high impedance on that gate would be reported.

In this version, during the Analog Float checks, Insight Analyzer explores for power switches that may lie above defined rails. If a defined rail is switched (A) by the same controls that cause high impedance (B), these two factors are reconciled and the float case can be eliminated if appropriate. The defined rail is off (A, receiver is off) while the gate node is high impedance (B).

Stack Trace

The stack trace in Details section has been trimmed slightly, with the intent of showing fewer nodes in cases where the trace becomes complex (many input variables).

Contention

The Contention checks have been updated to cover numerous additional scenarios that lie somewhat beyond the scope supported in prior versions. Some of these are outlined here.

Stack U-Turn

Contention cases that depend on a U-turn path within a driver stack are now supported. These are cases where a driver may share a portion of a common stack with other drivers, and current can flow backwards through a transistor from one driver on the shared node to another driver on the same shared node. In such cases, the typical stack of PMOS transistors to power, for example, may be only partially enabled. Current is not coming directly from the power rail, but is coming from another driver on the shared common node within the pullup stack.

Suspected Contention

The Contention check has been adjusted to report suspected cases that could not be proven. Following is an outline of relevant points:

- The Contention check, and the Analog Floats check, both issue a violation named “Unsafe Cell”. This violation indicates that the given cell (usually a dedicated block such as precharge, latch, or other lower level function) was proven to have an internal problem node by investigating the cell in isolation from the higher circuit. This is a warning that the cell could become a problem if not connected correctly.
- Beginning with this version, the Contention check can now issue a violation named “Suspected Contention” in cases where a proof (positive or negative) could not be reached within the scope of the whole circuit (convergence limitations). The suspected contention is reported in terms of the circuit parts that are known to cause contention, while ignoring other parts that may have been too complex for a timely solution (such as SRAM column address decoding, FPGA passgate arrays, etc).

Stack Trace

The stack trace in Details section has been trimmed slightly, with the intent of showing fewer nodes in cases where the trace becomes complex.

Domain Crossings

The Domain Crossings check has been updated from continued feedback of field cases. Please see updated documentation section “Domain Crossings” in the Scan Reference.

Report Wording

Often, a violation related to domain crossing can be complicated, as it is composed of a number of diverse factors (isolation control signal, control propagation, level shifter type, rail on/off states, origin of the floating data signal, etc). To help users better understand these violations, there have been tweaks to the wording of these violations. The wording changes are in the details section of a report line item.

Auto-Accept Definitions

If an isolation control is left undefined, the impact can be numerous violations in the Domain Crossings check. Many level shifters are found to have floating inputs, and their controls are coming from a toplevel input that was not defined in setup.

In this version, a new option is provided that will minimize these unwanted violations as follows:

1. There is a first encounter of an isolation stage depends on a control that is not defined.
2. A violation is issued for that isolation stage with undefined control, as normally done. An additional note is provided, showing that the control is assumed.
3. The isolation control is traced up-stream to the source. That control source is remembered, along with the active high or low state of the first encountered case (above).
4. For any subsequent encounters of isolation stages that depend on the same control source, with the same active state, violations are suppressed.

The option is enabled in Domain Crossings plugin flags as -autoAcceptIsoDefinitions.

Novel Isolation Topology

Normally, isolation of an incoming signal that is off is provided by one of the following traditional methods:

1. Isolation level shifter, dedicated to the purpose.
2. NAND or NOR, from standard logic library.
3. NMOS clamp on ground (usually safest), or PMOS clamp to positive rail.

In this version, Insight Analyzer now considers a new category of isolation stage:

4. Novel topology: Isolation is provided by a receiving stage that does not fit into one of the above categories. During the Domain Crossings check, if a floating input and isolation control arrive at a receiving stage that is not one of the common isolation types (above), a fallback method analyzes the receiver in terms of transistor-by-transistor behavior. This handles cases of, for example, a 6-transistor CMOS structure that combines switching and clamping to achieve the appropriate isolation function.

Tie-high / Tie-low

Enhancements have been made in the handling of constant tie-high and tie-low sources, as sources for valid isolation control. A tie-high or tie-low circuit that is not recognized will now have the desired impact, in spite of any topology recognition issue.

A tie-low source that depends on a supply rail that is off (according to power state table, as appropriate) is now considered a risk factor for any receiver on the virtual tie-low output.

In a stuck-low case, the “invisible” power rail, powering the tie-low output, will be displayed as part of the corresponding violation.

Rail Names in Waivers

This is an early access feature, not normally enabled. Please see the section on Waivers.

Power Connections

Multi-Factor Reporting

Prior versions would not report transistor Vg issues if the same transistor had already triggered a Vd / Vs violation. This version will report both types for the same transistor, if appropriate.

In particular, this can lead to cases such as the following: (1) A Vs/d over-volt violation is reported, followed by (B) new additional violation that the same transistor is being under-driven at its gate (missing level shifter).

Bumping SOA Violations

An adjustment has been made in the handling of MOSFET safe operating area violations (over SOA) related to gate voltages. In prior versions, a gate-source or gate-drain SOA violation could effectively block a gate-bulk SOA violation on the same MOSFET. In this version, if the gate-bulk violation is determined to be worse than the gate-source or gate-drain violation, the MOSFET is reported on the basis of the gate-bulk violation, likely with a higher priority value listed.

Stack Height

There have been small updates related to stacks with branching and shared nodes. Some complicated cases of stack branching were resulting in unrealistic transistor counts in the Stack Height report.

Passgate XOR

Strictly speaking, an XOR circuit based on transmission gate can be a violation of stack height. The counted total of transistors from power rail to output can be in excess of the typical threshold for failure (often 2 or 3). For this reason, a stack height violation that includes the transmission gate of an XOR structure will be noted in the report as “XOR”.

Waivers

Fanout Value Tolerance

In prior versions, reports that include numeric values that are used as part of the waiver key, the key is generated to one decimal place of precision. This allows waivers to match violations that may have slight variation in these values.

Starting in this version, the Fanout plugin will allow waiver matching to within a user defined percentage tolerance.

The following new user preferences are used to control this behavior:

Preference Name	Default	Description
waiverValueToleranceEnable	0	Enables waiver matching to tolerance. When set to false, there is no change in behavior from prior versions.
waiverValueToleranceDefault	10	Specifies the percent tolerance for numeric values to be considered a match for all reports not otherwise set by other user preferences. This setting is ignored in this release, but is reserved for future releases.
waiverValueToleranceForFanout	10	Specifies the percent tolerance for Fanout values to be considered a match. This setting is ignored unless unless waiverValueToleranceEnable is 1.
waiverValuePrecision	3	The precision to save numeric fields in the waiver key. This setting is ignored unless unless waiverValueToleranceEnable is 1.

As an example, consider a Fanout violation with a value of 1.333.

Prior to this version, a waiver would use a key value of 1.3 such that any (otherwise identical) violation with a value in the range 1.300 -1.399 would match.

In this version, the new behavior can be enabled with the following command:

```
project preference waiverValueToleranceEnable 1
```

A waiver for this violation will now use a key value of 1.333 and any (otherwise identical) violation with a value of 1.333 ±10% will match. The precision of the saved key value and the tolerance can be set in a similar manner.

API

Insight Analyzer script commands have been expanded in this version. There is a mix of extended commands and new commands, as outlined below.

Recognition

cdb inst isHysteresisFeedback
Hysteresis driver, such as comparator output stage, commonly include a MOSFET that is gated by the output.

cdb inst isGatedDiode, isGatedDiodeControl
Updates in the classification of MOSFETs wired as diodes with controls.

cdb inst isCrossCoupled
Identifies cross coupled structures that may appear in level shifters etc.

cdb inst isColumnMux
Uses topology to recognize SRAM column multiplexer function, such as MOSFET acting as passgate on bit line.

cdb inst isAnalogDisableSwitch
Indicates that the given MOSFET instance has the function of cutting off current in an analog bias stack.

cdb inst isAnalogDisableClamp
Indicates that the given MOSFET instance acts as a disable clamp during periods of low power mode.

cdb net isAnalogDisable
Indicates the control of an analog power-down state.

cdb inst isSafeBiasVictim
Determines whether the given MOSFET instance is protected as an SOA victim in an analog bias network.

cdb inst isMissingTieHighLow
Check for a MOSFET gate directly tied to power/ground rail, according to foundry rules. These rules take into account some additional factors, such as the placement of the victim device within a stack.

Scanning

cscan instances -avoidProfileRevisit
Scan with option to skip block cells that repeat same power / ground profile at IO connections.

cscan allowRevisitCurrentProfile

Command to re-admit a subsequent repeat of same power / ground profile at IO connections.

Multi-Threading

cstore (all sub commands)

The behavior of these commands has been updated, to emulate the equivalent Tcl commands as closely as possible.

Nets

cdb net isSegment, isSegmentMaster, isSegmentSlave, getSegments, getSegmentMaster

Commands for working with a set of nets that have been merged together by resistors.

cdb net isPossibleLogicSignal

A weak indication of “logic” function, for applications that may need to capture all such cases.

cdb net isTieHighLow -effective

Option to consider “effective” tie functions that are given by a stuck state and not recognition.

cdb net isParallelBusBit, getParallelBusBits

Uses fuzzy heuristics to determine that a net is part of a bus within the same container cell.

cdb net getAddressAndControl

Attempt to find address and control inputs that determine state on the given net.

cdb net findUpstreamRails

Explore up stream to find the rail(s) that directly feed the state of the given net.

Marker Tags

While not directly related to transistor level circuit checking, support for net marker tags has been expanded in this version of Insight Analyzer. The following commands are provided only for applications that have this particular need (not common).

cdb net addVfiltMark -replace, -tightenMinMax

Options added to the “add mark” command.

cdb net getVfiltMark -allSegments

Option added to the “get mark” command.

cdb net clearVfiltMarks

This command now propagates to all linked ancestors by default.

Instances

cdb inst getName -cellInst

Option to return the name with parent cell, not the hierarchical path.

cdb inst getHookupSignature
Generates a profile string related to the connections on IO of the given instance.

Other

cdb getPowerDefRails -replications
Option to return all replications of power rails, if any.

cdb inst checkIsolation -ignoreTopInputs
Option to disregard a toplevel IO as cause for domain isolation problem, when using this API to return a result (not related to running the Domain Crossings plugin).

Upcoming Features

The next major release will be Insight Analyzer v5.30. The upcoming version will have particular updates that have already been implemented, but not normally enabled, in the current v5.21.

Users who need access to these features can enable them as described in the following sections.

Test Framework

In v5.30, a test and QA framework is provided through the GUI. This may be enabled in v5.21 as follows:

- From Linux: setenv INSIGHT_EVAL testFramework
- From Tcl script: runenv setHiddenFeatures testFramework

Power Connections

In v5.30, the Power Connections check uses a number of new techniques to reduce pessimistic false positives in analog circuits. These updates may be enabled in v5.21 as follows:

- From Linux: setenv INSIGHT_EVAL v53
- From Tcl script: runenv setHiddenFeatures v53

Waivers in Domain Crossings

In v5.30, an update is available for Domain Crossings waiver keys. Starting in this version, it is possible to enable this update, using an environment variable prior to invoking the tool:

- From Linux: setenv INSIGHT_HOOKS floats_includeRailsInHashKey=1

Summary of Differences

When compared to Insight Analyzer v5.20, the following differences are notable:

Saved Settings

A script written from Session > Project > Save will now have additional sections that had not been saved in prior versions. Additionally, the choice of a default project name and the declared working directory have been adjusted.

Reports

Insight R Numbers

The details section of most report lines, from most checks, will have Insight Reference Numbers embedded. Example: The violation “Float, LS internal" is accompanied by “R253” as a reference to documentation and examples.

Domain Crossings

The details within Domain Crossings report will often have wording updates over prior versions.

Floats & Contention

The details in reports from Analog Floats and Contention checks will show stack traces that are different from older versions.

Contention

New violation type “Suspected contention” has been added to reports.

Waiver Files

The waivers for Domain Crossings may now have rail names included in the key strings (waiver database files).