

Introduction to JasperGold SuperLint

Presenter: Jui-Huang Tsai
2023.11.28



Outline

- Lint / Auto-Formal Categories
- Basic Lint / Auto-Formal Execution Steps
- How to use
- Summary
- Reference



Lint Categories (1/3)

- NAMING
 - Checks naming conventions on RTL elements, for example modules, instances functions, etc. based on a rules file defined by user
- FILEFORMAT
 - Checks file formatting to make files portable and reusable, for example, file name different from module name
- CODINGSTYLE
 - Checks to ensure there are no semantic and functional issues in the code, for example, unconnected ports, unused and unassigned variables, or undriven signals
- SIM_SYNTH
 - Checks for scenarios that can cause mismatch between pre and post synthesis simulation results, for example, incomplete sensitivity list



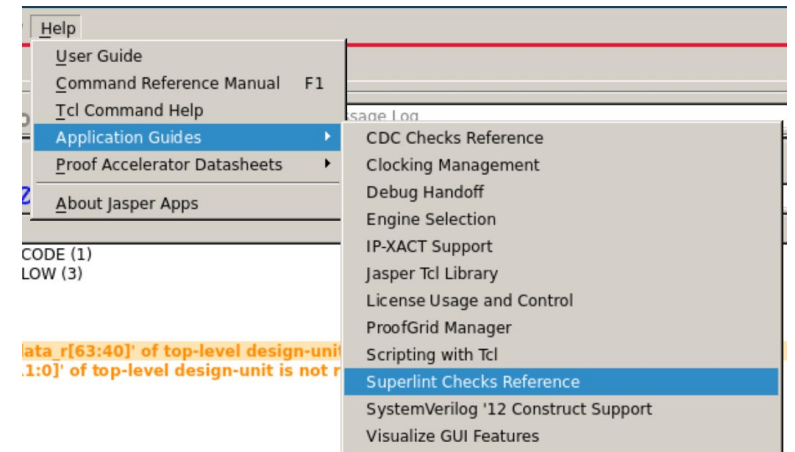
LINT Categories (2/3)

- SYNTHESIS
 - Checks for constructs that are not synthesizable, for example, the Initial statement to initialize the signal in Verilog code can result in unpredictable synthesis behavior
- STRUCTURAL
 - Checks for design structures that can cause functional or downstream tool issues, for example, Flip-Flops missing resets
- RACES
 - Checks for simulation race condition scenarios. Race conditions occur when two events (with event order dependent code) take place at the same simulation time. For example, read and write on the same register in different always blocks



LINT Categories (3/3)

- CONNECTIVITY
 - Checks for the connectivity path provided with the `config_rtlDs –connectivity` command. For example, mandatory connections between two nodes in the design
- BLACKBOX
 - Checks for blackbox related scenarios while elaborating the design. For example, explicit blackboxing of module/instance due to the `-bbox_m/-bbox_i` options passed to the `elaborate` command
- For more information, please refer to “Superlint Checks Reference”



Auto-Formal Categories (1/2)

- AUTO_FORMAL_OVERFLOW
 - Overflow in assignments, expressions, divide by zero scenarios and shift operations
- AUTO_FORMAL_BUS
 - Bus contention and floating
- AUTO_FORMAL_CASE
 - Priority/Unique and default case reachability
- AUTO_FORMAL_COMBO_LOOP
 - Detection of combo loops
- AUTO_FORMAL_DEAD_CODE
 - RTL code block reachability



Auto-Formal Categories (2/2)

- AUTO_FORMAL_FSM_DEADLOCK_LIVELOCK
 - FSM deadlock (there is no outgoing path) and FSM livelock (there is no path back to initial state)
- AUTO_FORMAL_FSM_REACHABILITY
 - Unreachable FSM transitions
- AUTO_FORMAL_OUT_OF_BOUND_INDEXING
 - Index out of range
- AUTO_FORMAL_SIGNALS
 - Stuck-at, signal deadlock, signal not toggled
- AUTO_FORMAL_X_ASSIGNMENT
 - A reachable X-assignment was found



Basic Lint / Auto-Formal Execution Steps

1. Configure Rules
2. Analyze and Elaborate RTL
3. Design Configuration (Auto Formal Only)
4. Run Checks
5. Analysis and Debug
6. Create Waivers
7. Generate Reports



Configure Rules (1/2)

- Rules file contains the list of all LINT rules
 - `lint_rule.def`
- Customizations
 - Category names
 - Rules included in each category
 - Enable or disable a category
 - Severity
 - Short message
 - Rule name
 - Specific parameter for a rule, for example, pattern for naming convention rules.



Configure Rules (2/2)

- You can edit the file (lint_rule.def) to adjust lint and auto-formal configuration

Category name

```
category SYNTHESIS
{
  MOD_NR_EVRP {severity=Error} {msg="%s %s contains event specification which cannot be synthesized"}
  MOD_NR_FKJN {severity=Error} {msg="%s %s contains non-synthesizable fork-join constructs"}
  MOD_NR_SPFY {severity=Error} {msg="The module '%s' contains non-synthesizable specify block"}
```

Rule name

Severity

Short Message



Analyze and Elaborate RTL

- `% analyze -sv -f 01_RTL/design.f`
- `% elaborate -top smc_veneer ${DESIGN}`



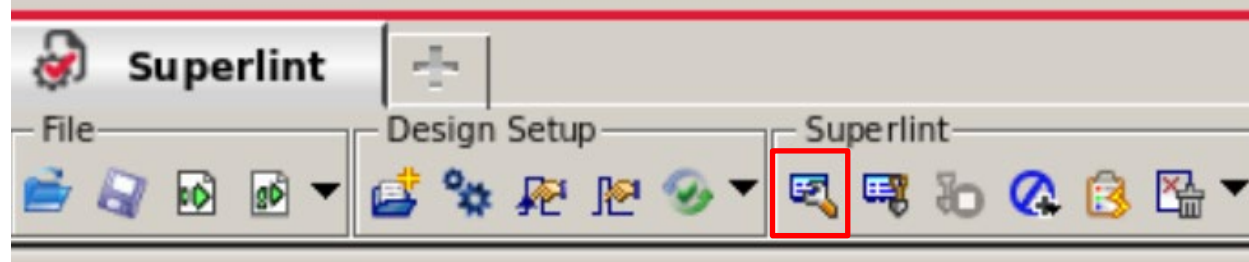
Design Configuration (Auto Formal Only)

- Clock and reset definition
 - `clock –infer |<specify_clock/s> : Clock Definition`
 - `reset <specify_reset_signals> : Reset Definition`
- Signal configuration required to configure the environment
 - `check_superlint –signal (-constant {{signal_name value}+}...`
 - `assume –env {...}`



Run Checks

- `% check_superlint -extract`
- GUI



Analysis and Debug

Violation Tree

The screenshot displays the Cadence Superlint interface with three main panels:

- Violation Messages View (Left):** A tree structure showing violations categorized by domain (LINT 131) and tag (MOD_NO_IPRG 2). It lists various errors such as "Input port 'inf.C data r[63:40]' of top-level design-unit is not registered" and "Expression in design-unit 'BEV' has both signed '360' and unsigned 'bev_bal.black tea'".
- Analysis Browser (Right):** Displays the source code of the design unit 'EXP_NR_MXSU_130'. The code is a Verilog-like snippet showing signal declarations and assignments. A specific line (780) is highlighted in yellow, corresponding to a violation in the Violation Tree.
- Waiver List (Bottom Right):** A table listing waivers for the analysis. The table has columns for Comment, Id, Source, Upr, Tag, and Category. A single waiver is listed with Comment 'test1', Id '1', Source 'EXP_NR_MXSU', and Category '*'.

Comment	Id	Source	Upr	Tag	Category
test1	1	EXP_NR_MXSU			*

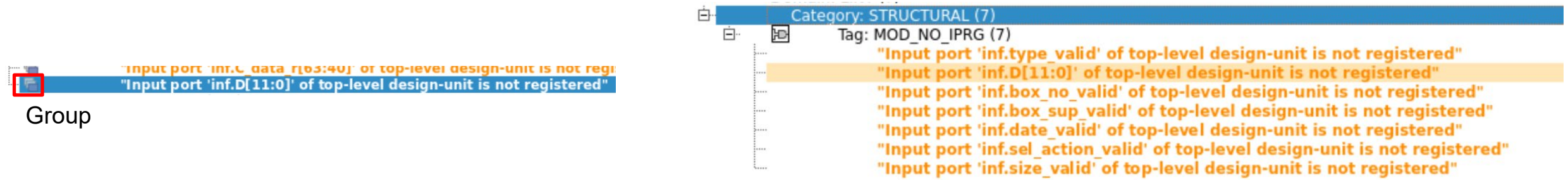
The bottom panel shows the console output, including instructions for running the analysis and the results of the extraction process.

Source Code

Waiver List



Analysis and Debug – Grouping Feature (Lint)



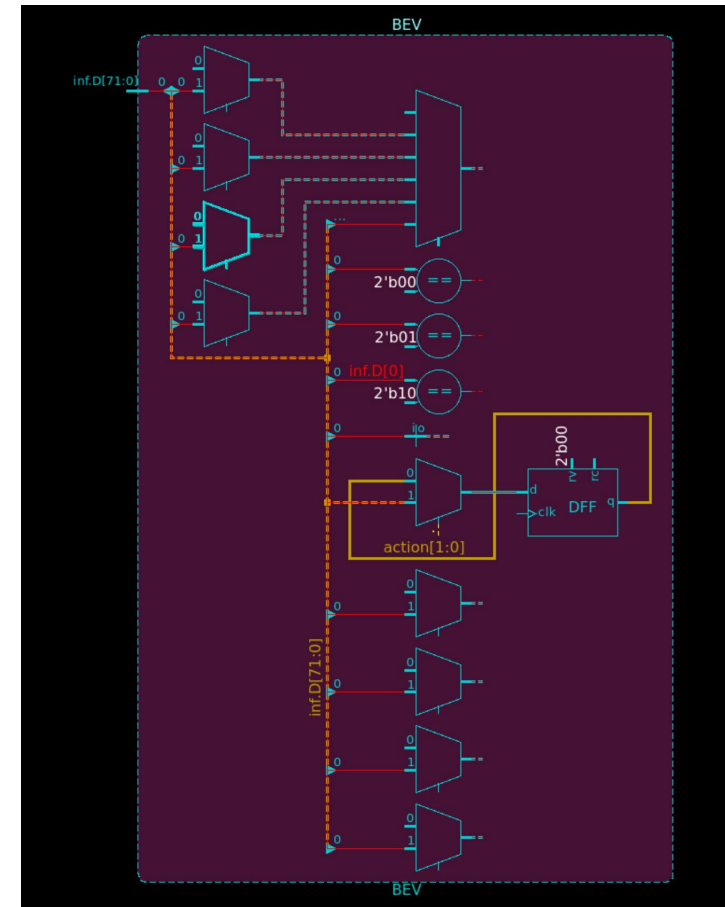
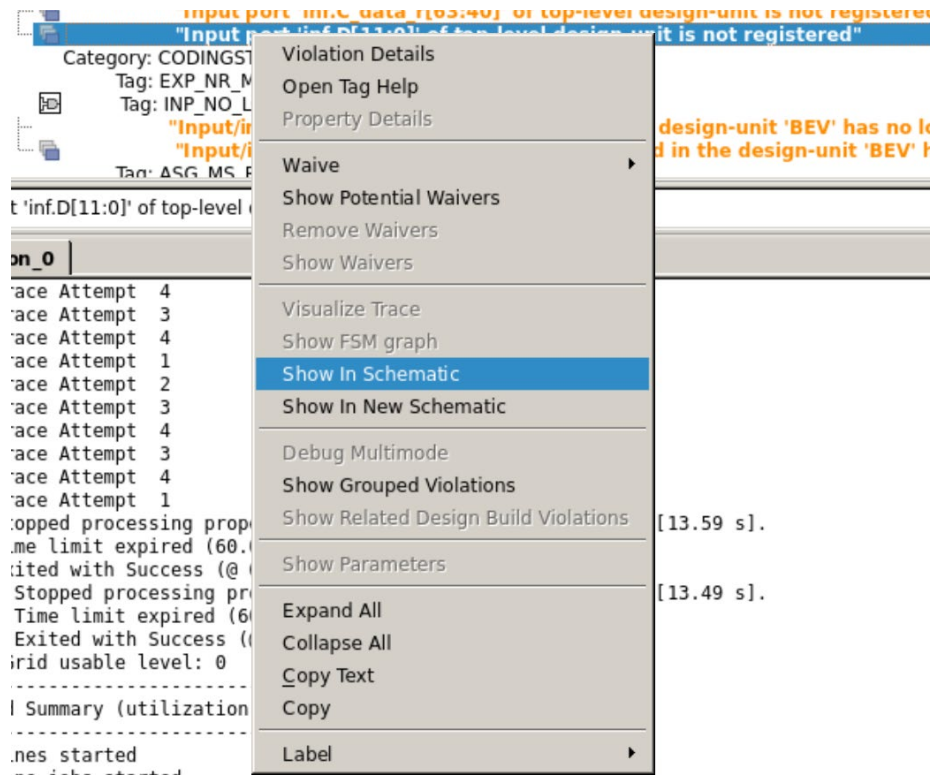
The image shows a screenshot of a linting tool interface. On the left, a small icon of a person is highlighted with a red box, and the word "Group" is written below it. The main area displays a list of linting errors. The first error is highlighted in blue and reads: "Input port 'inf.C_data_r[63:40]' of top-level design-unit is not registered". The second error is highlighted in orange and reads: "Input port 'inf.D[11:0]' of top-level design-unit is not registered". To the right, a detailed view of the selected error is shown, listing seven input ports that are not registered: 'inf.type_valid', 'inf.D[11:0]', 'inf.box_no_valid', 'inf.box_sup_valid', 'inf.date_valid', 'inf.sel_action_valid', and 'inf.size_valid'. The interface includes a tree view on the left with icons for file, folder, and error, and a header bar with the text "Category: STRUCTURAL (7)" and "Tag: MOD_NO_IPRG (7)".

Group

Category: STRUCTURAL (7)
Tag: MOD_NO_IPRG (7)

- "Input port 'inf.C_data_r[63:40]' of top-level design-unit is not registered"
- "Input port 'inf.D[11:0]' of top-level design-unit is not registered"
- "Input port 'inf.type_valid' of top-level design-unit is not registered"
- "Input port 'inf.D[11:0]' of top-level design-unit is not registered"
- "Input port 'inf.box_no_valid' of top-level design-unit is not registered"
- "Input port 'inf.box_sup_valid' of top-level design-unit is not registered"
- "Input port 'inf.date_valid' of top-level design-unit is not registered"
- "Input port 'inf.sel_action_valid' of top-level design-unit is not registered"
- "Input port 'inf.size_valid' of top-level design-unit is not registered"

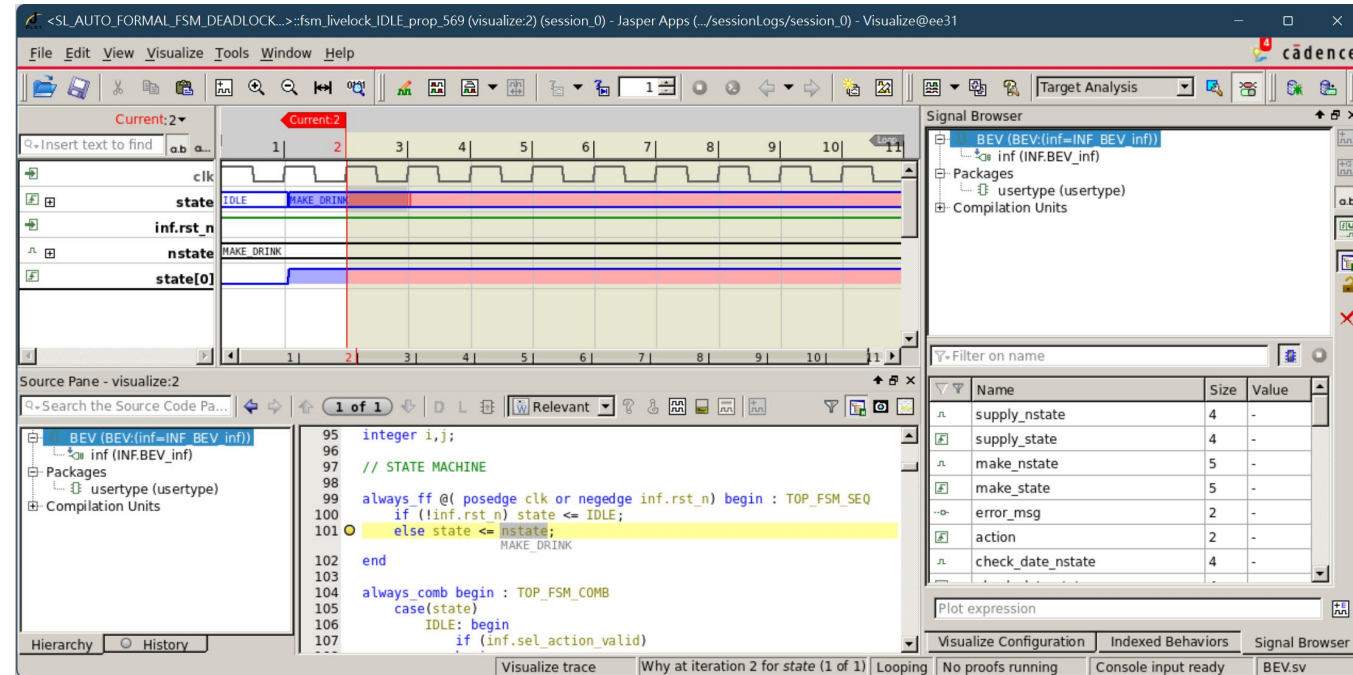
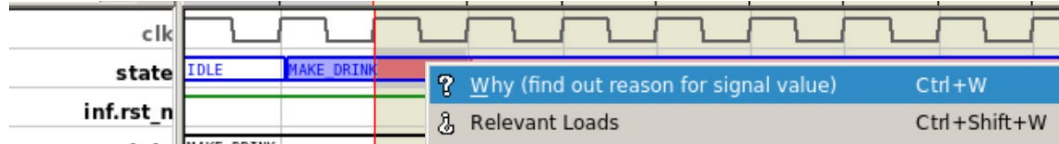
Analysis and Debug - Using Schematic (Lint)



Analysis and Debug – Visualize Waveform (Auto-Formal) (1/2)

The screenshot displays the Cadence Jasper tool interface for analyzing formal verification results. On the left, a list of violations is shown, including 'Potential Livelock condition found in the FSM state', 'Potential Deadlock condition found in the FSM state', and 'The blocking statement'. A context menu is open over the violations, offering options like 'Violation Details', 'Open Tag Help', 'Property Details', 'Waive', 'Show Potential Waivers', 'Remove Waivers', 'Show Waivers', 'Visualize Trace', 'Show FSM graph', and 'Show In Schematic'. The 'Visualize Trace' option is highlighted. The central pane shows a waveform visualization for the 'state' signal, with a red vertical line indicating the current time step (Current: 2). The waveform shows the state transitions over time, with 'IDLE' and 'MAKE DRINK' states visible. The right pane shows the 'Signal Browser' with a list of signals and their values. The signals listed are: supply_nstate (4), supply_state (4), make_nstate (5), make_state (5), error_msg (2), action (2), and check_date_nstate (4). The bottom status bar indicates 'Selected: state', 'Visualize trace', 'Why at iteration 2 for state[0] (1 of 1)', 'Looping', 'No proofs running', 'Console input ready', and 'BEV.sv'.

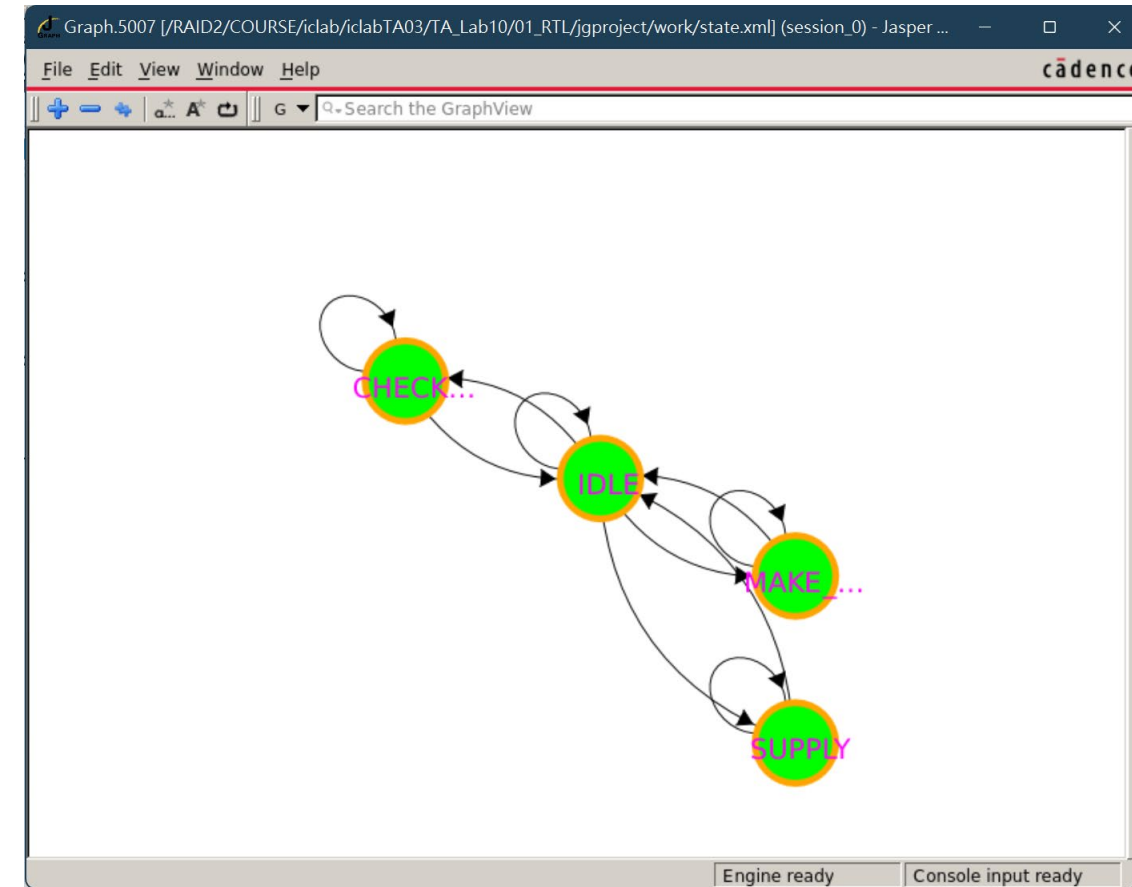
Analysis and Debug – Visualize Waveform (Auto-Formal) (2/2)



Analysis and Debug – Using FSM Graph

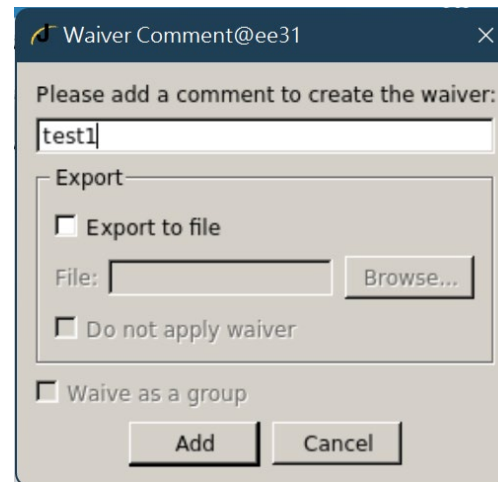
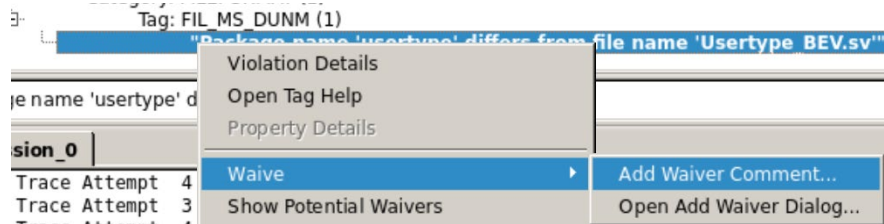
The screenshot shows a list of violations in the Jasper tool. The violations are categorized by tag and include messages about potential livelock and deadlock conditions. A context menu is open over the list, showing options like 'Violation Details', 'Open Tag Help', 'Property Details', 'Waive', 'Show Potential Waivers', 'Remove Waivers', 'Show Waivers', 'Visualize Trace', 'Show FSM graph', 'Show In Schematic', and 'Show In New Schematic'. The 'Show FSM graph' option is highlighted.

- Tag: FSM_IS_PLLK (1)
 - *W, "Potential Livelock condition found in the FSM 'state'"
- Tag: FSM_IS_PDLK (4)
 - *W, "Potential Deadlock condition found for state 'IDLE' i"
 - *W, "Potential Deadlock condition found for state 'IDLE M"
 - *W, "Potential Deadlock condition found for state 'IDLE S"
 - *W, "Potential Deadlock condition found for state 'IDLE C"
- Tag: FSM_IS_NLLK (3)
 - *I, "FSM 'check_date_state' does not have any Livelock co"
 - *I, "FSM 'make_state' does not have any Livelock conditio"
 - *I, "FSM 'supply_state' does not have any Livelock conditi"
- Tag: FSM_IS_NDLK (35)
 - Category: AUTO_FORMAL_DEAD_CODE (1)
 - Tag: BLK_NO_RCHB (1)
 - *E, "The blocking statement 'make_nstate = 5'b00000 ;' is"
 - Category: AUTO_FORMAL_OVERFLOW (3)
 - Domain: LINT (131)



Create Waivers

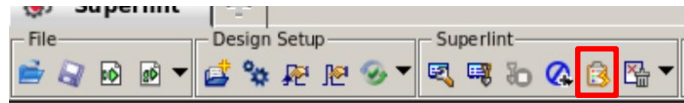
- Some Warning/Error may can be ignored
- Will be removed from the violation messages view tree



A screenshot of the 'Waiver List' table. The table has a title bar 'Waiver List' and a close button. The table has columns: Comment, Id, Source, Uf, Tag, and Category. The table contains one row with the following data: Comment: test1, Id: 1, Source: (empty), Uf: (empty), Tag: FIL_MS_DUNM, Category: *.

Comment	Id	Source	Uf	Tag	Category
test1	1			FIL_MS_DUNM	*

Generate Report



Generate Report@ee31

General | Select Tags | Select Instances

☒ Generate Detailed Report

Violation Setting

☒ Rule Violation Messages ☐ Include Design Build Messages

Severity: ☒ Error ☒ Warning ☒ Info

Order report by:

Property Settings

☐ Property

Types

- ☒ All
- ☒ Cover
- ☒ Assert

Status

- ☒ All
- ☒ Cex
- ☒ Unreachable
- ☒ Undetermined
- ☒ Proven
- ☒ Covered
- ☒ Waived

Format

☒ HTML ☒ Open In Browser

☐ XML

☐ CSV

☒ TXT

☒ Console

Waived Settings

☐ Waived Checks Only

☐ Exclude Waived Checks

☒ Include Waived Checks

OK Cancel

Jasper&trade - Superlint Violation/Property Report

module BEV:(inf=INF_BEV_inf) created date Nov 28 22:16:32 CST 2023

VIOLATIONS SUMMARY

Domain: AUTO_FORMAL

#	Category	Error	Warning	Info	Waived	Total
1	AUTO_FORMAL_DEAD_CODE	1	0	0	0	1
2	AUTO_FORMAL_FSM_DEADLOCK_LIVELOCK	0	5	6	0	11
3	AUTO_FORMAL_OVERFLOW	3	0	0	0	3
4	Total	4	5	6	0	15

AUTO_FORMAL_DEAD_CODE

#	Severity	Tag	Message	Source Location	Instance-Module	Grouping Status
1	Error	BULK_NO_RCHB	"The blocking statement 'make_nstate = 5'b000000' is not reachable"	BEV.vv:241	BEV:(inf=INF_BEV_inf)-BEV	-

AUTO_FORMAL_FSM_DEADLOCK_LIVELOCK

#	Severity	Tag	Message	Source Location	Instance-Module	Grouping Status
1	Warning	FSM_IS_PDLK	"Potential Deadlock condition found for state 'IDLE' in the FSM 'state'"	BEV.vv:106	BEV:(inf=INF_BEV_inf)-BEV	Grouped
2	Warning	FSM_IS_PDLK	"Potential Deadlock condition found for state 'IDLE_M' in the FSM 'make_state'"	BEV.vv:137	BEV:(inf=INF_BEV_inf)-BEV	-
3	Warning	FSM_IS_PDLK	"Potential Deadlock condition found for state 'IDLE_S' in the FSM 'supply_state'"	BEV.vv:254	BEV:(inf=INF_BEV_inf)-BEV	-
4	Warning	FSM_IS_PDLK	"Potential Deadlock condition found for state 'IDLE_C' in the FSM 'check_date_state'"	BEV.vv:286	BEV:(inf=INF_BEV_inf)-BEV	-
5	Warning	FSM_IS_PLK	"Potential Livelock condition found in the FSM 'state'"	BEV.vv:106	BEV:(inf=INF_BEV_inf)-BEV	-
6	Info	FSM_IS_NDLK	"State 'TYPE_M' in the FSM 'make_state' does not have any Deadlock condition"	BEV.vv:138	BEV:(inf=INF_BEV_inf)-BEV	Grouped
7	Info	FSM_IS_NDLK	"State 'DATE_S' in the FSM 'supply_state' does not have any Deadlock condition"	BEV.vv:255	BEV:(inf=INF_BEV_inf)-BEV	Grouped
8	Info	FSM_IS_NDLK	"State 'DATE_C' in the FSM 'check_date_state' does not have any Deadlock condition"	BEV.vv:287	BEV:(inf=INF_BEV_inf)-BEV	Grouped
9	Info	FSM_IS_NLKL	"FSM 'make_state' does not have any Livelock condition"	BEV.vv:137	BEV:(inf=INF_BEV_inf)-BEV	-
10	Info	FSM_IS_NLKL	"FSM 'supply_state' does not have any Livelock condition"	BEV.vv:254	BEV:(inf=INF_BEV_inf)-BEV	-
11	Info	FSM_IS_NLKL	"FSM 'check_date_state' does not have any Livelock condition"	BEV.vv:286	BEV:(inf=INF_BEV_inf)-BEV	-

AUTO_FORMAL_OVERFLOW

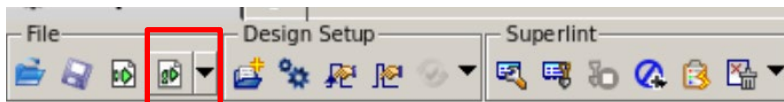
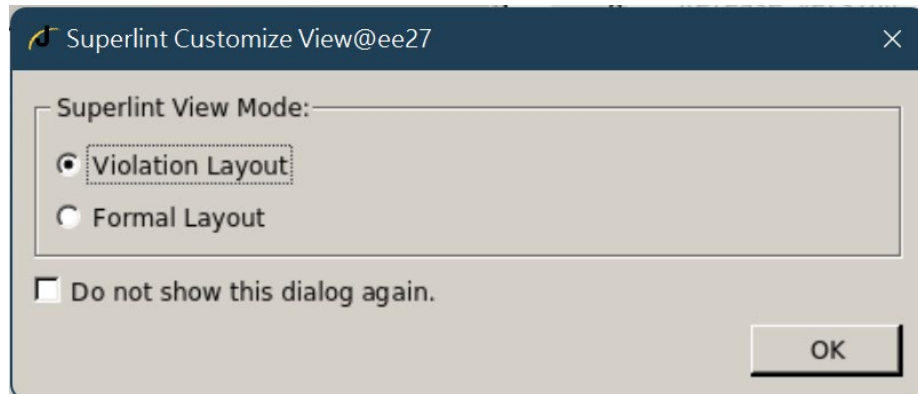
#	Severity	Tag	Message	Source Location	Instance-Module	Grouping Status
---	----------	-----	---------	-----------------	-----------------	-----------------

How to use

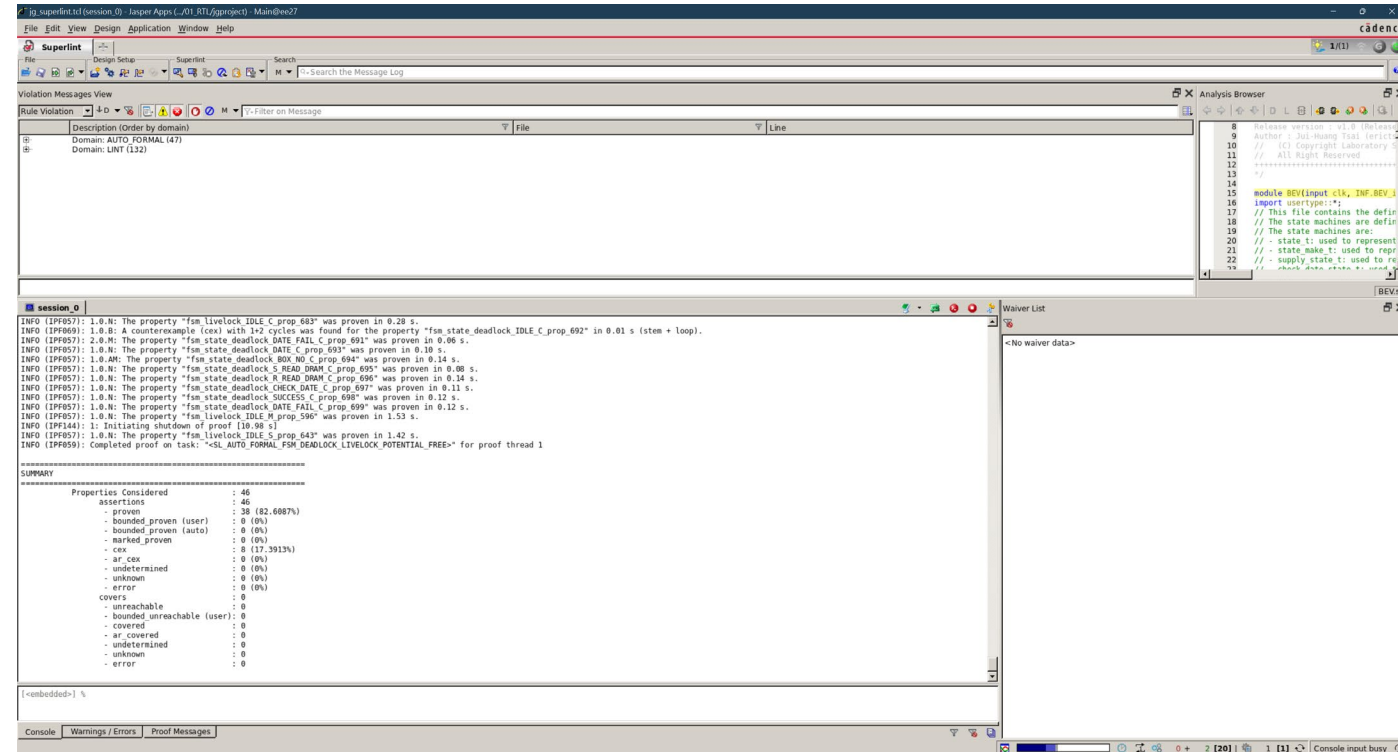
- The file we need
 - `jg_superlint.tcl`
 - `lint_rule.def`
 - `design.f`
 - Put all the design file name into this file
- `./06_superlint`



Startup



This bottom can reload whole design again



Reference

- rak_jasper_superlint_auto_formal_lab_instructions
- rak_jasper_superlint_auto_formal_overview
- rak_jasper_superlint_lint_lab_instructions
- rak_jasper_superlint_lint
- 2023 Autumn ICLAB Lab09/Lab10

