cadence

Introduction to JasperGold SuperLint

Presenter: Jui-Huang Tsai

2023.11.28

cadence

Outline

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

cadence

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

cadence

LINT Categories (2/3)

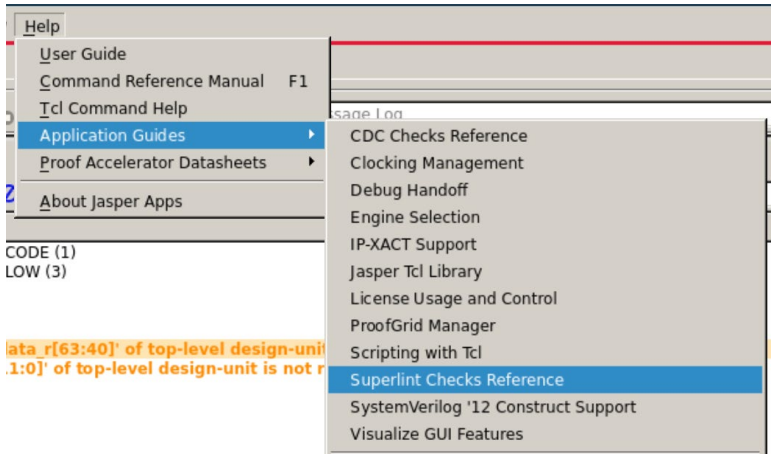
- SYNTHESIS
  - Checks for constructs that are not synthesizable, for example, the Initial statement to initialize the signal in Verilog code can result in unpredictable synthesis behavior
- STRUCTURAL
  - Checks for design structures that can cause functional or downstream tool issues, for example, 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

cadence

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”



cadence

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

cadence

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

cadence

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

cadence

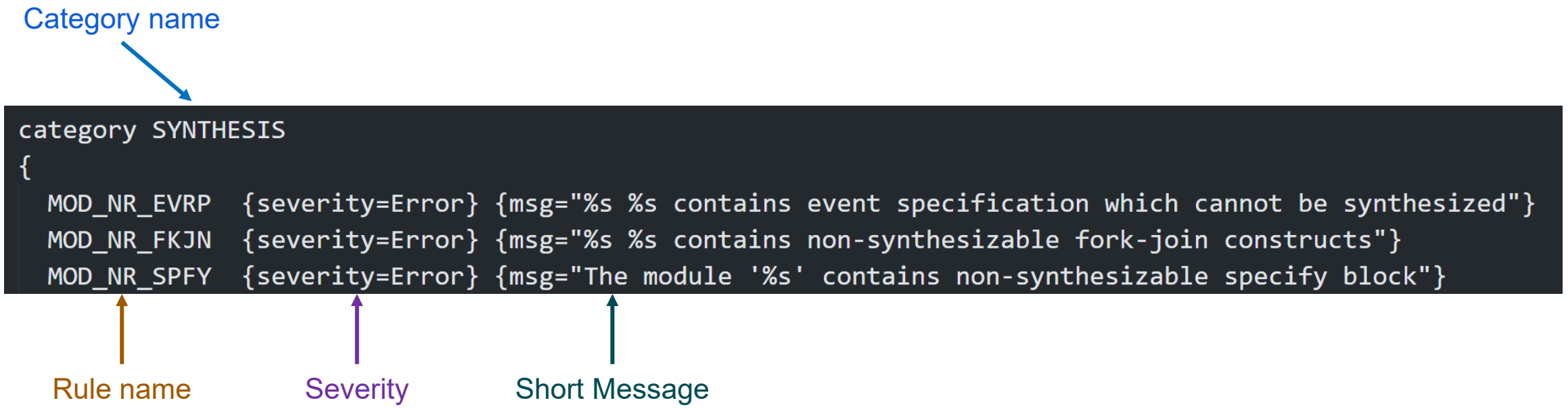
Configure Rules (1/2)

- 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.

cadence

Configure Rules (2/2)

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



Analyze and Elaborate RTL

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

cadence

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 {…}

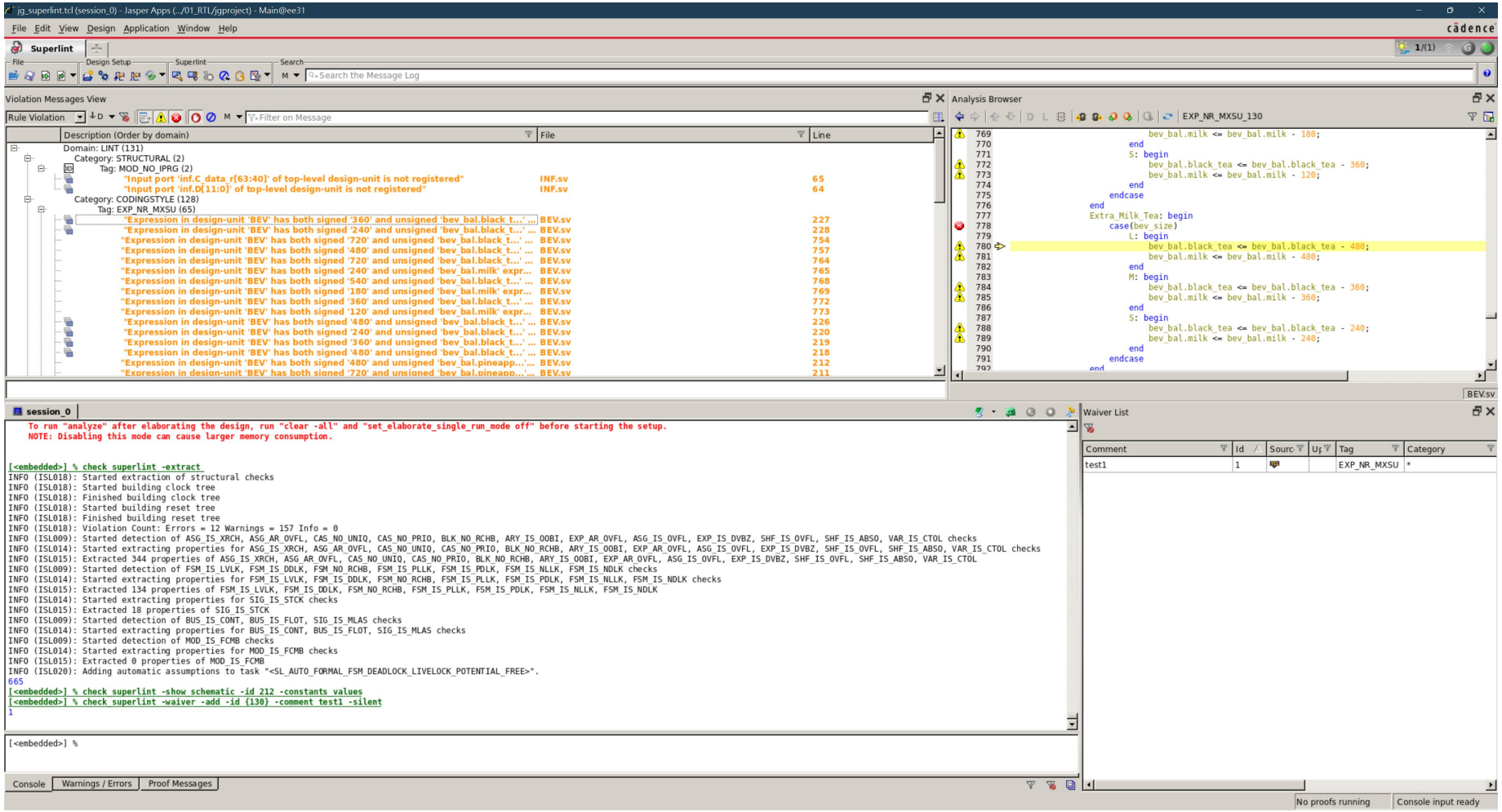
cadence

Run Checks

- `% check_superlint -extract`
- GUI

Analysis and Debug

Violation Tree

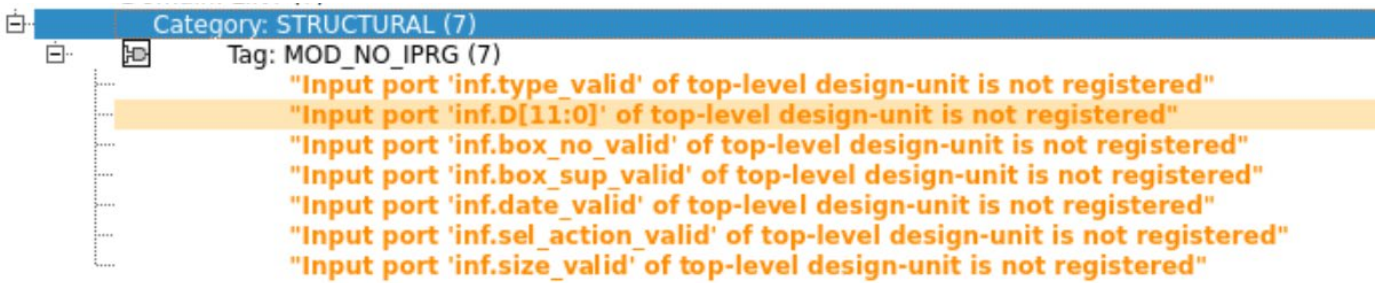


Source Code

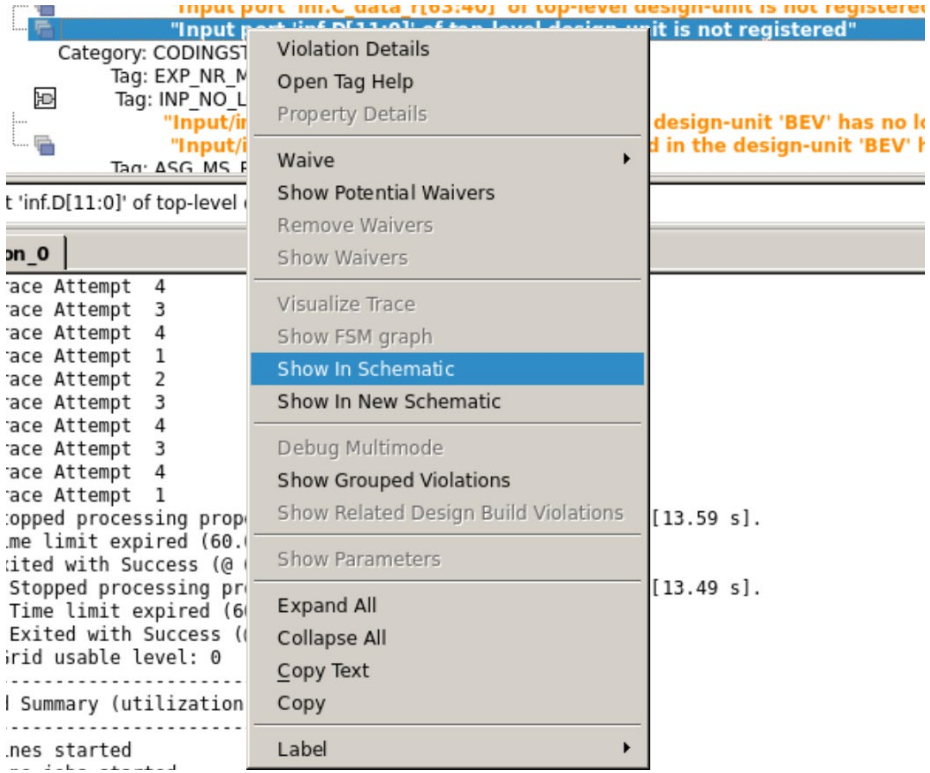
Waiver List

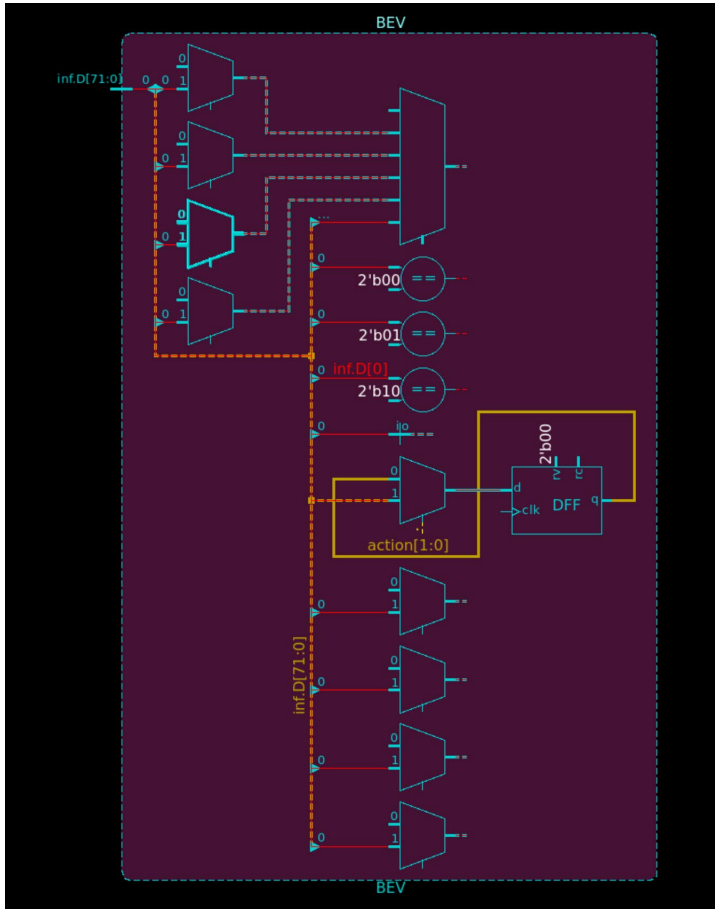
Analysis and Debug – Grouping Feature (Lint)

Group

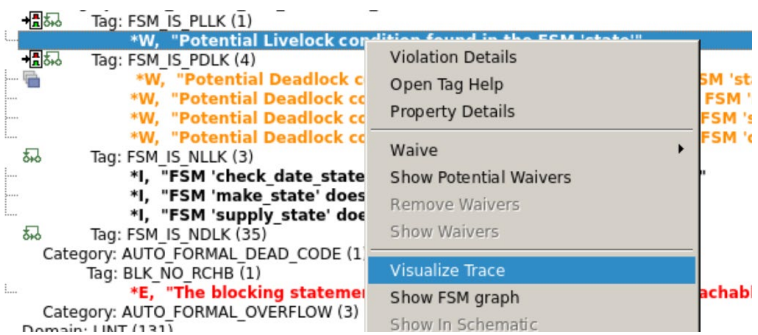


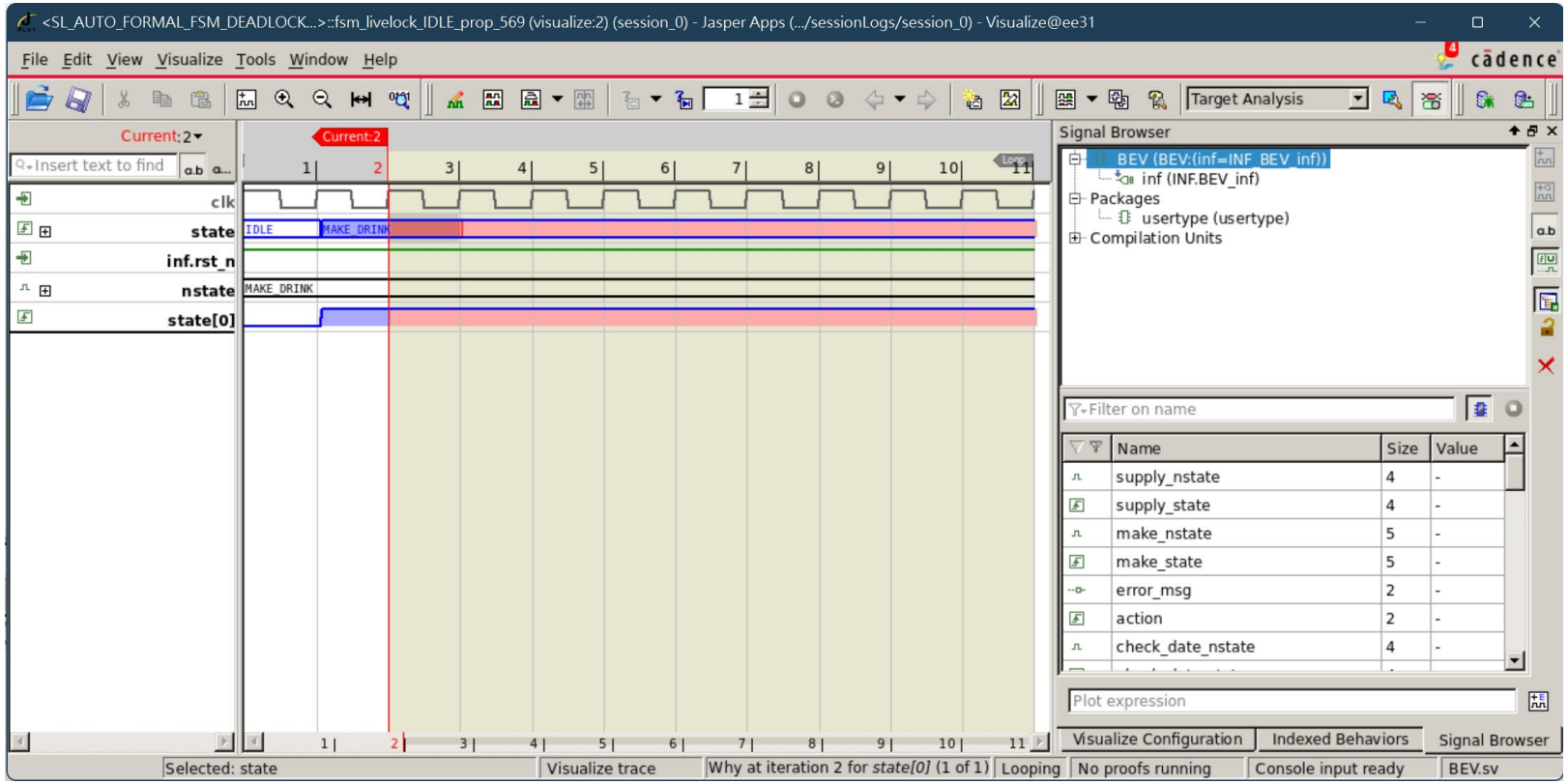
Analysis and Debug - Using Schematic (Lint)



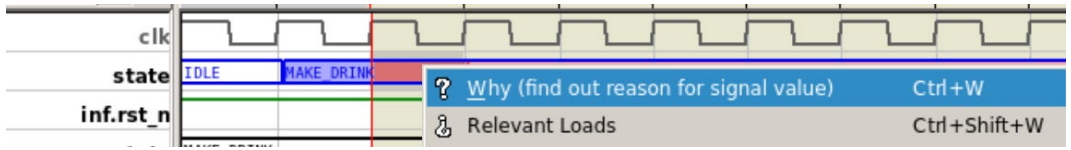


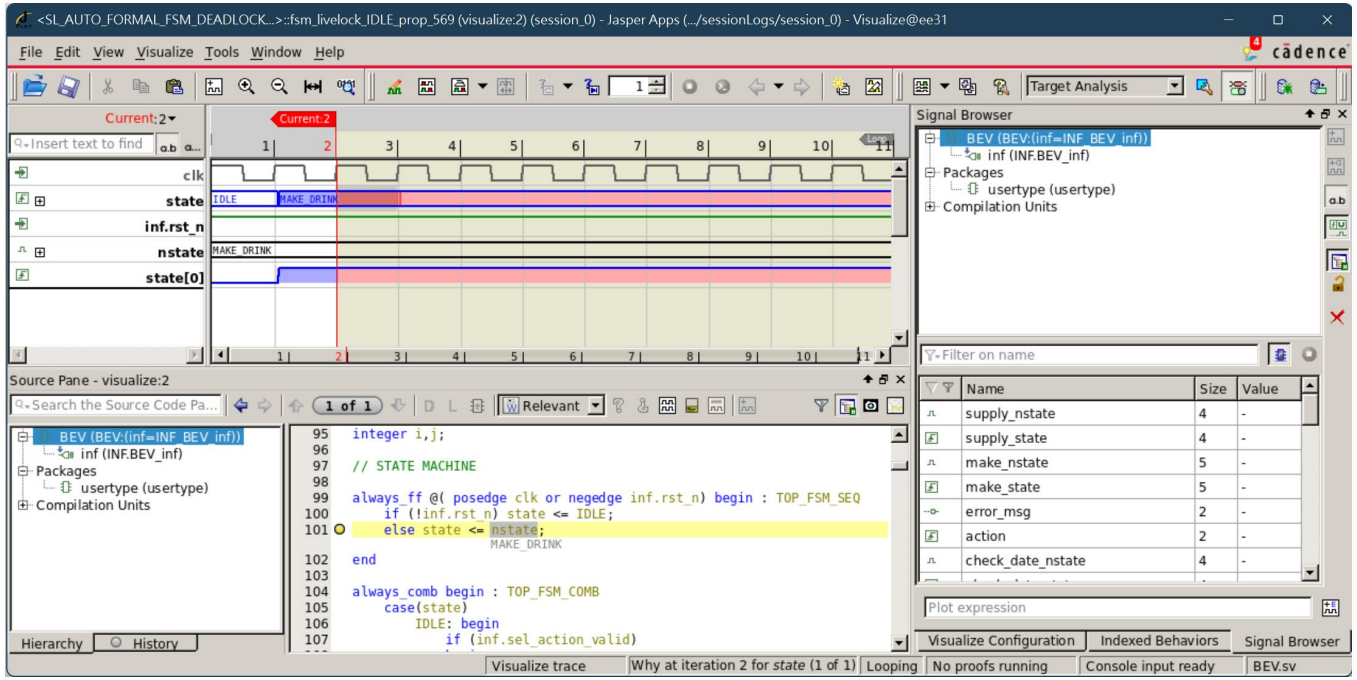
Analysis and Debug – Visualize Waveform (Auto-Formal) (1/2)



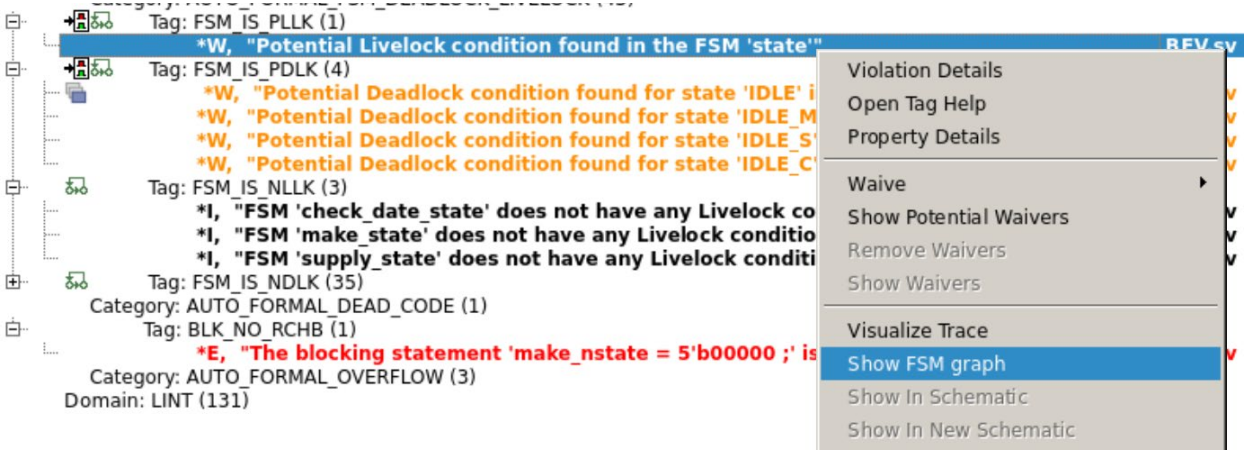


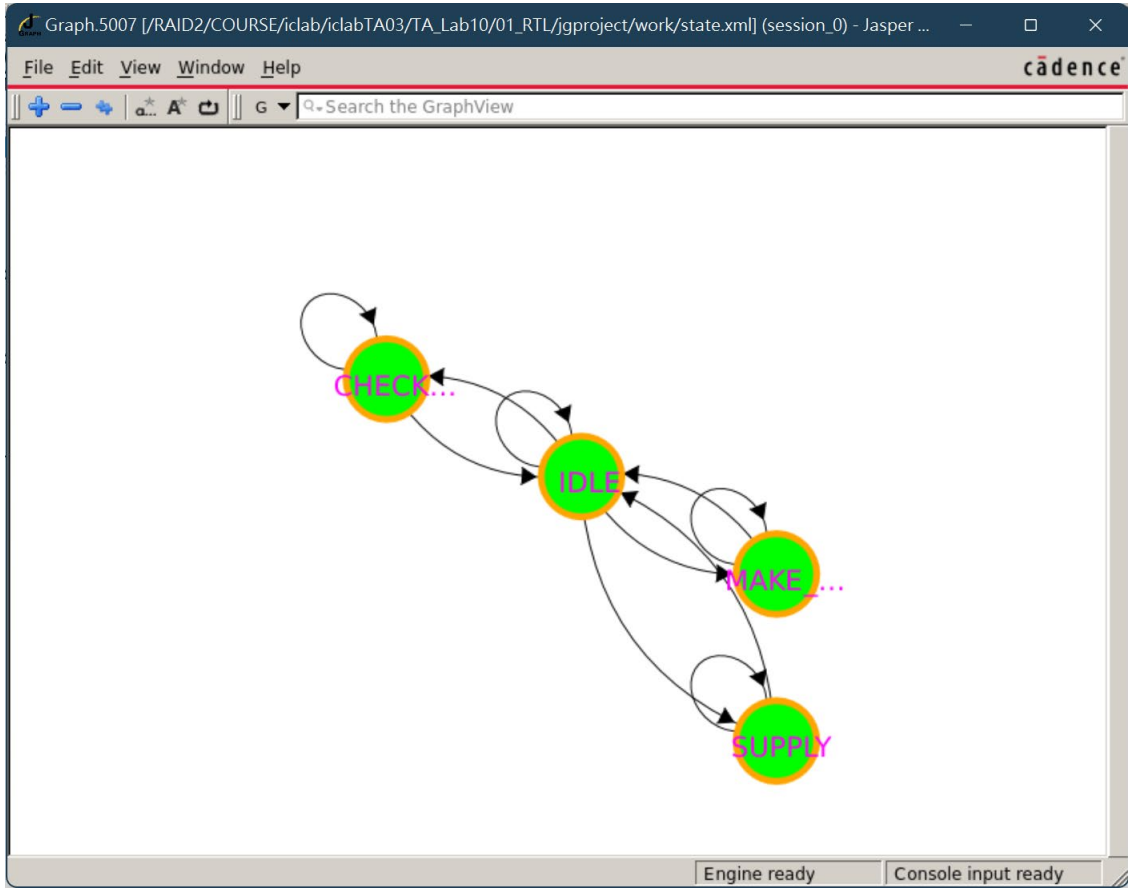
Analysis and Debug – Visualize Waveform (Auto-Formal) (2/2)





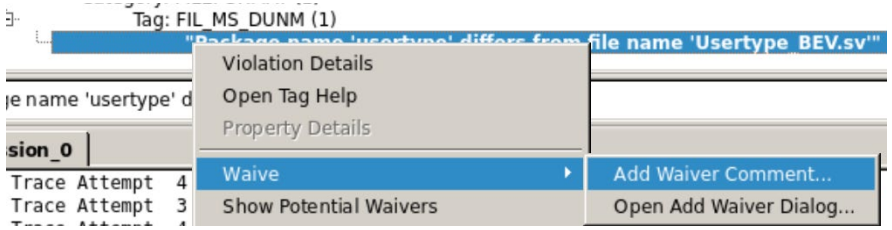
Analysis and Debug – Using FSM Graph

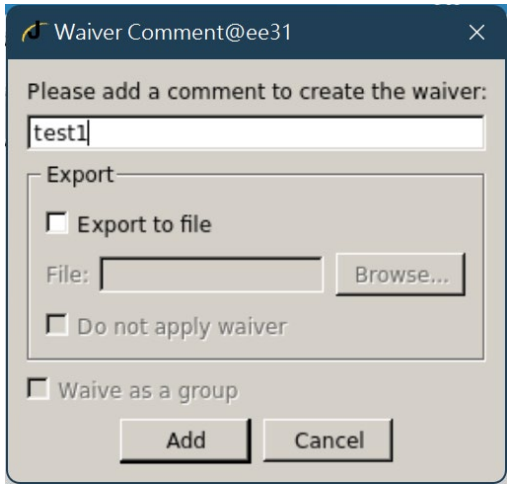




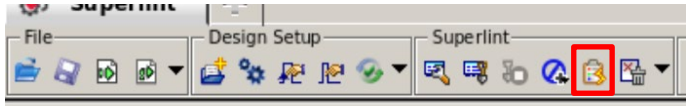
Create Waivers

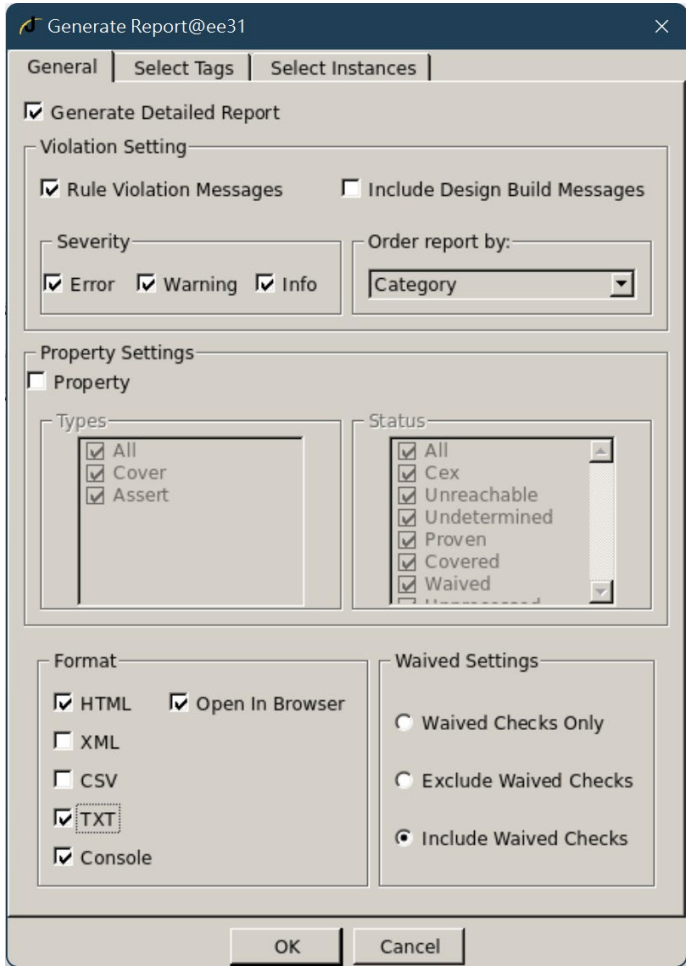
- Some Warning/Error may can be ignored
- Will be removed from the violation messages view tree

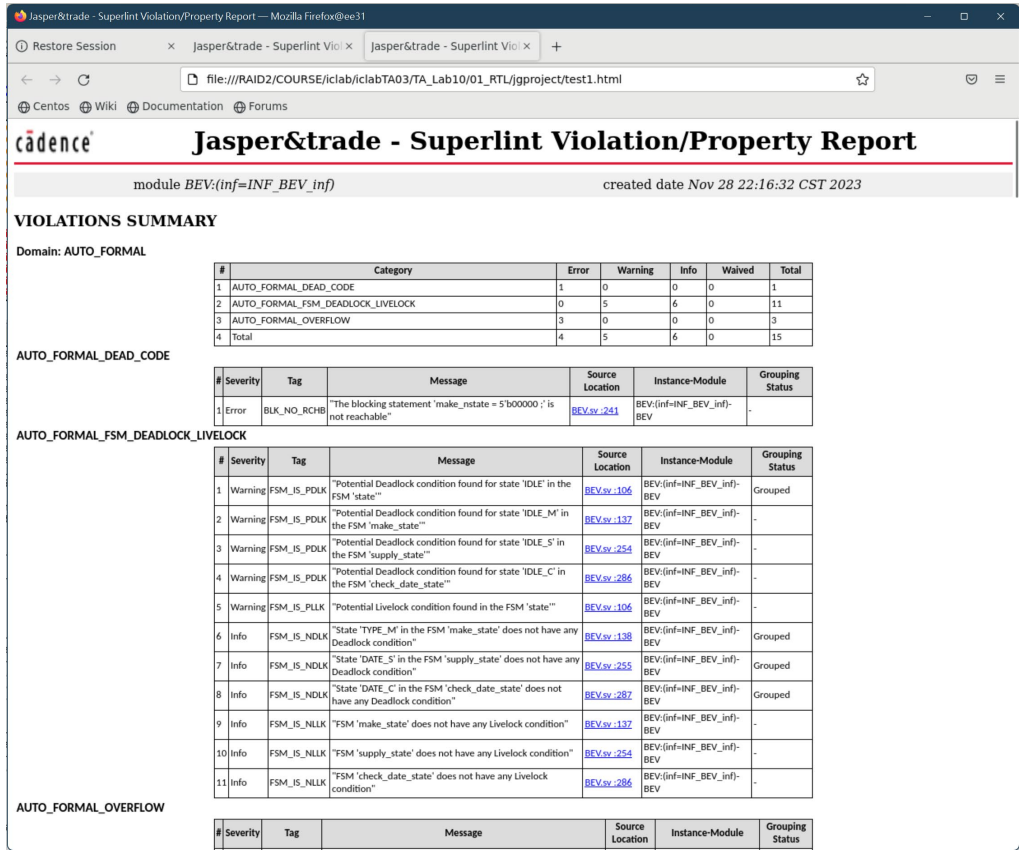




Generate Report







#	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

#	Severity	Tag	Message	Source Location	Instance-Module	Grouping Status
1	Error	BLK_NO_RCHB	"The blocking statement 'make_nstate = 5'b00000 ;' is not reachable"	BEV.sv :241	BEV:(inf=INF_BEV_inf)-BEV	-

#	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.sv :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.sv :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.sv :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.sv :286	BEV:(inf=INF_BEV_inf)-BEV	-
5	Warning	FSM_IS_PLLK	"Potential Livelock condition found in the FSM 'state'"	BEV.sv :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.sv :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.sv :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.sv :287	BEV:(inf=INF_BEV_inf)-BEV	Grouped
9	Info	FSM_IS_NLLK	"FSM 'make_state' does not have any Livelock condition"	BEV.sv :137	BEV:(inf=INF_BEV_inf)-BEV	-
10	Info	FSM_IS_NLLK	"FSM 'supply_state' does not have any Livelock condition"	BEV.sv :254	BEV:(inf=INF_BEV_inf)-BEV	-
11	Info	FSM_IS_NLLK	"FSM 'check_date_state' does not have any Livelock condition"	BEV.sv :286	BEV:(inf=INF_BEV_inf)-BEV	-

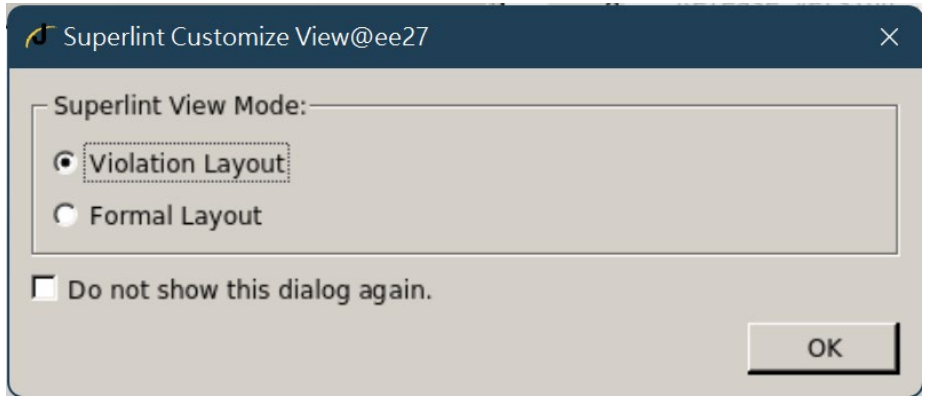
#	Severity	Tag	Message	Source Location	Instance-Module	Grouping Status

cadence

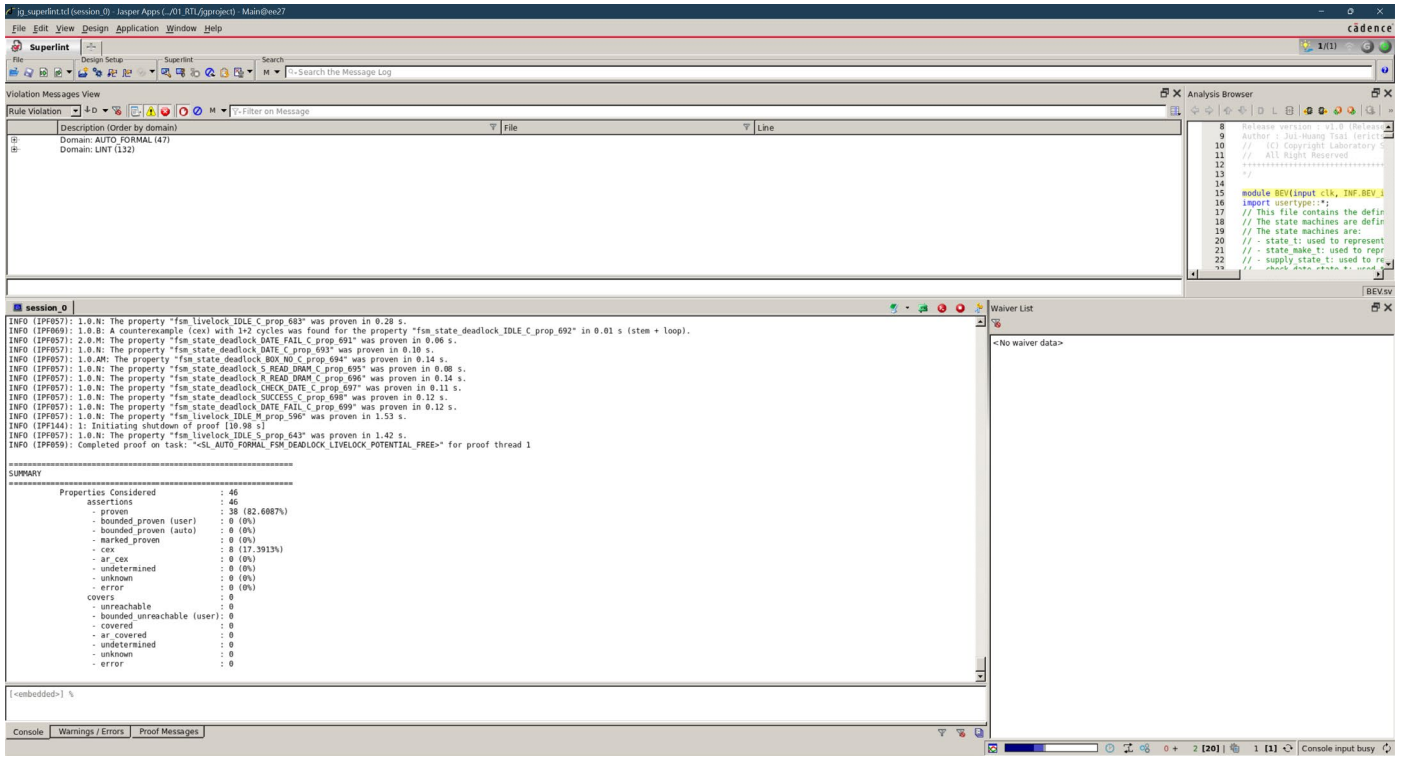
How to use

- 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



cadence

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