Spyglass CDC verification

Speaker:Victor

Advisor: Lih-Yih Chiou

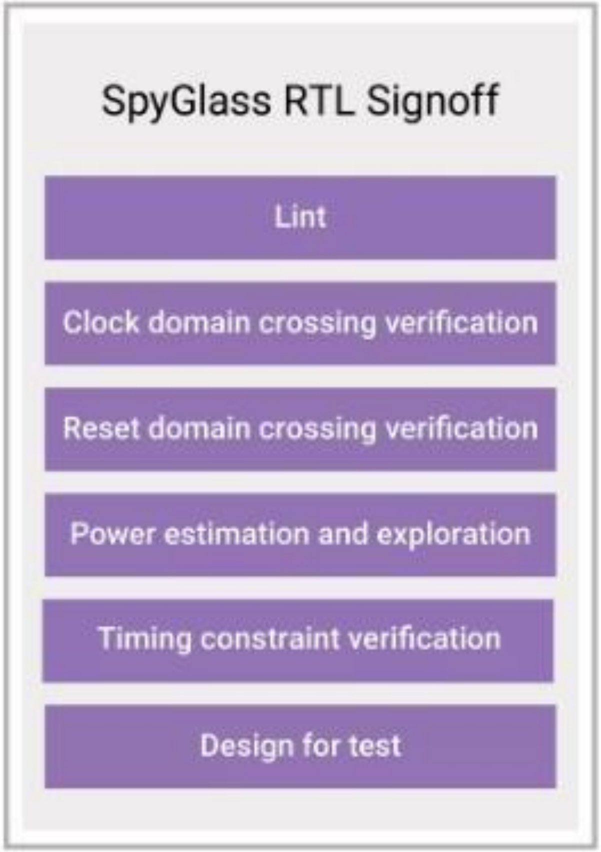
Date: 2024/11/6

Outline

- Introduction
- Import design
- CDC verification

Introduction

- Spyglass is the early design analysis tool enabled efficient verification and optimization of soc designs
- With this tool, designers can gain insight into their designs early in the RTL process.
  - Lint, CDC, LP, DFT, Constraint



SpyGlass RTL Signoff

Lint

Clock domain crossing verification

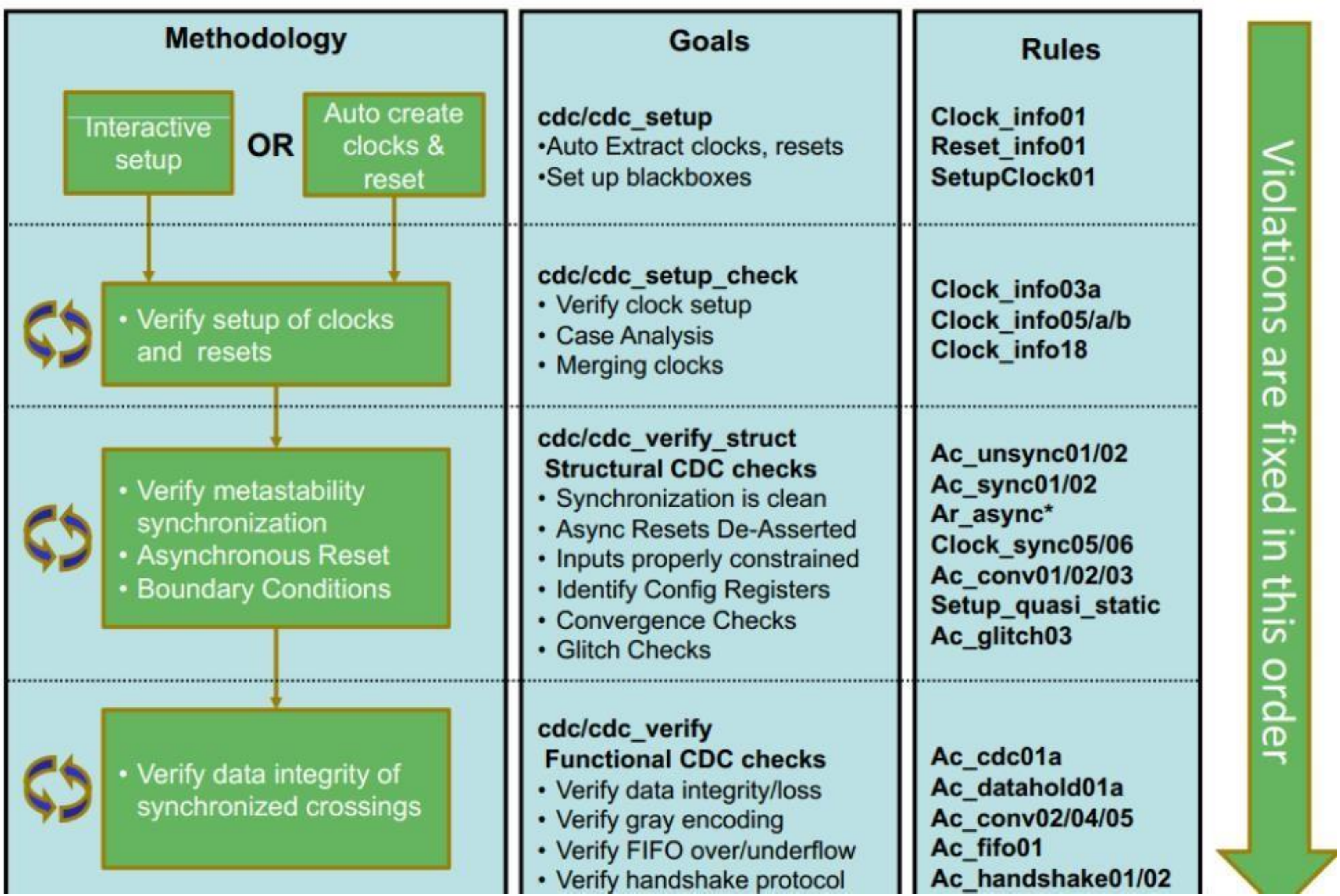
Reset domain crossing verification

Power estimation and exploration

Timing constraint verification

Design for test

Spyglass CDC verification flow

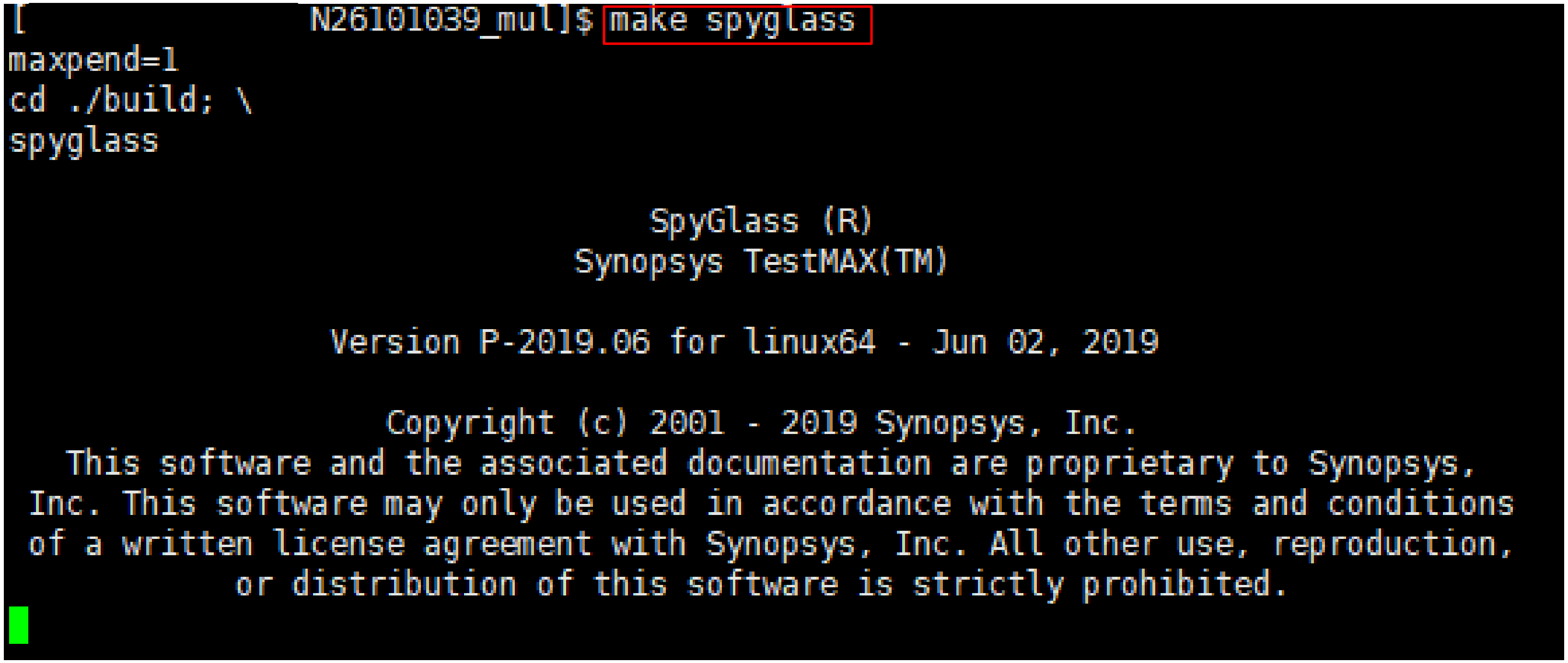


Methodology	Goals	Rules
Interactive setup OR Auto create clocks & reset	**cdc/cdc_setup** •Auto Extract clocks, resets •Set up blackboxes	**Clock_info01** **Reset_info01** **SetupClock01**
• Verify setup of clocks and resets	**cdc/cdc_setup_check** • Verify clock setup • Case Analysis • Merging clocks	**Clock_info03a** **Clock_info05/a/b** **Clock_info18**
• Verify metastability synchronization • Asynchronous Reset • Boundary Conditions	**cdc/cdc_verify_struct** **Structural CDC checks** • Synchronization is clean • Async Resets De-Asserted • Inputs properly constrained • Identify Config Registers • Convergence Checks • Glitch Checks	**Ac_unsync01/02** **Ac_sync01/02** **Ar_async*** **Clock_sync05/06** **Ac_conv01/02/03** **Setup_quasi_static** **Ac_glitch03**
• Verify data integrity of synchronized crossings	**cdc/cdc_verify** **Functional CDC checks** • Verify data integrity/loss • Verify gray encoding • Verify FIFO over/underflow • Verify handshake protocol	**Ac_cdc01a** **Ac_datahold01a** **Ac_conv02/04/05** **Ac_fifo01** **Ac_handshake01/02**

Violations are fixed in this order

Open GUI

- [spyglass/build] $ spyglass
- [spyglass] $ make spyglass
  - In this exercise & homework

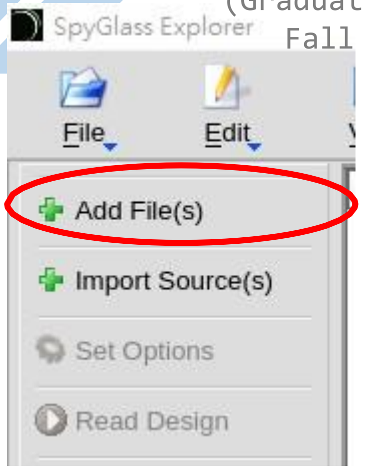


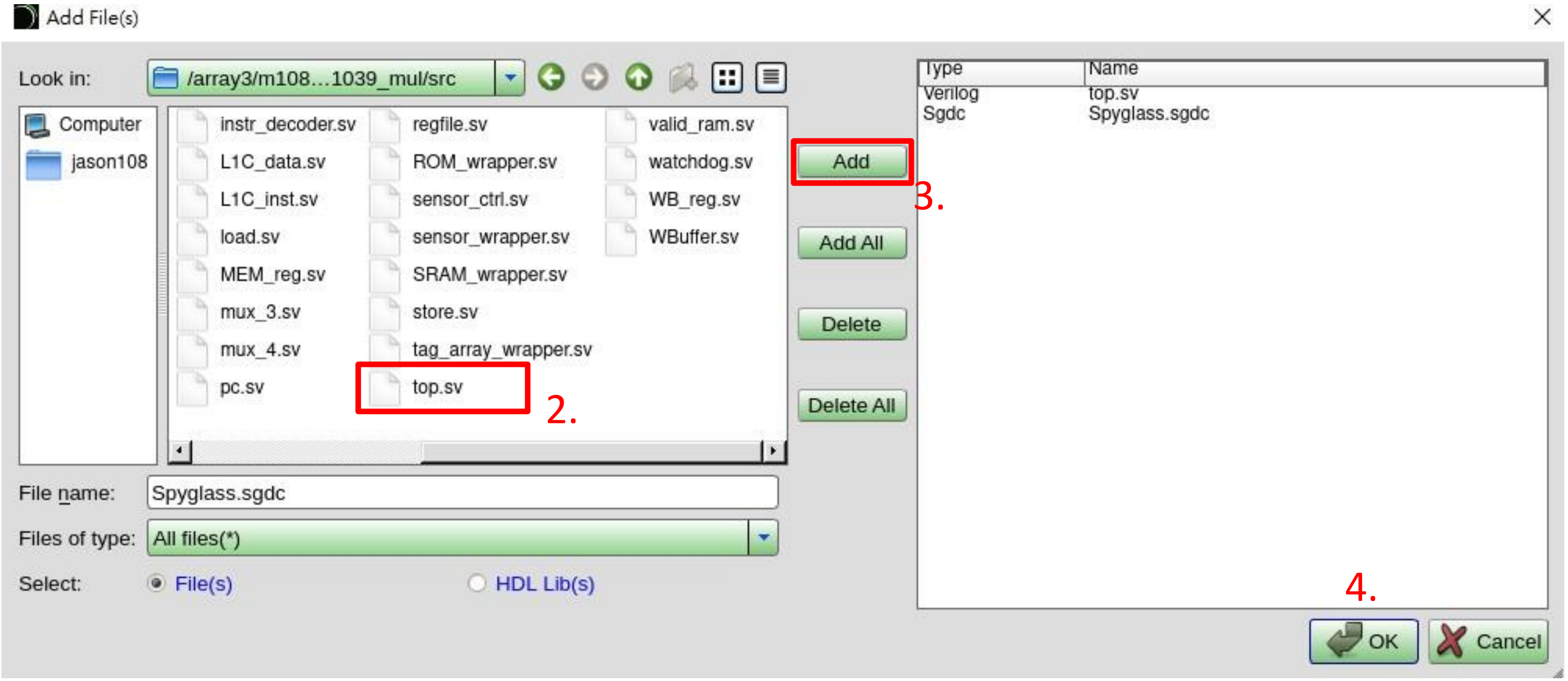
```
[[redacted] N26101039_mul]$ make spyglass
maxpend=1
cd ./build; \
spyglass

                              SpyGlass (R)
                          Synopsys TestMAX(TM)

                 Version P-2019.06 for linux64 - Jun 02, 2019

                    Copyright (c) 2001 - 2019 Synopsys, Inc.
   This software and the associated documentation are proprietary to Synopsys,
 Inc. This software may only be used in accordance with the terms and conditions
 of a written license agreement with Synopsys, Inc. All other use, reproduction,
            or distribution of this software is strictly prohibited.
```

Import design

Import design

- Read design file(.v/.sv)
  - read_file –type verilog “top module file”
  - read_file -type sgdc “sgdc file”





Look in: /array3/m108...1039_mul/src

Type	Name
Verilog	top.sv
Sgdc	Spyglass.sgdc

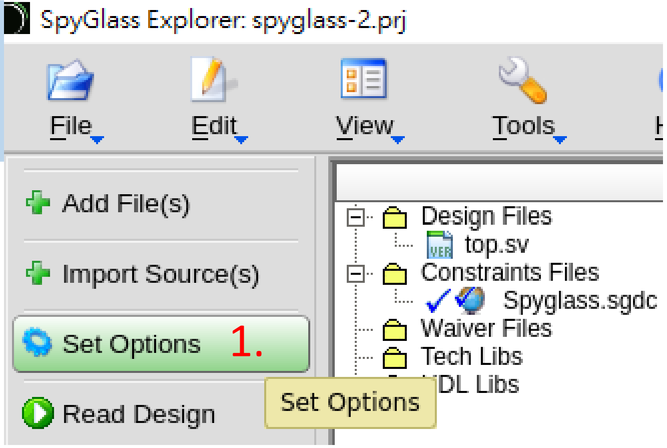
File name: Spyglass.sgdc

Files of type: All files(*)

Select: ● File(s) ○ HDL Lib(s)

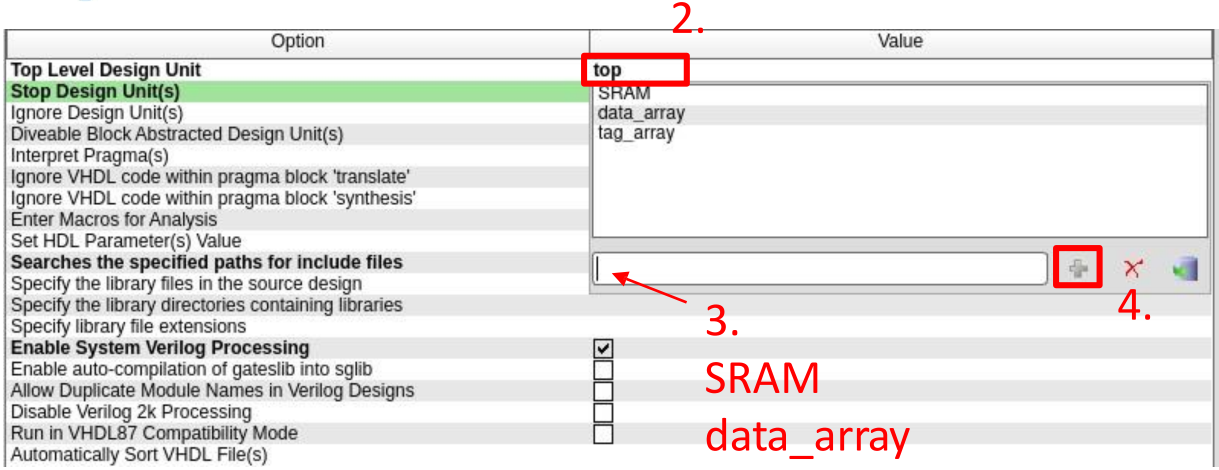
Option settings

- Read design & constraint file
  - set_option top “top module name”
  - set_option stop “blackbox module name”
  - set_option incdir “path”
  - set_option enableSV yes



SpyGlass Explorer: spyglass-2.prj

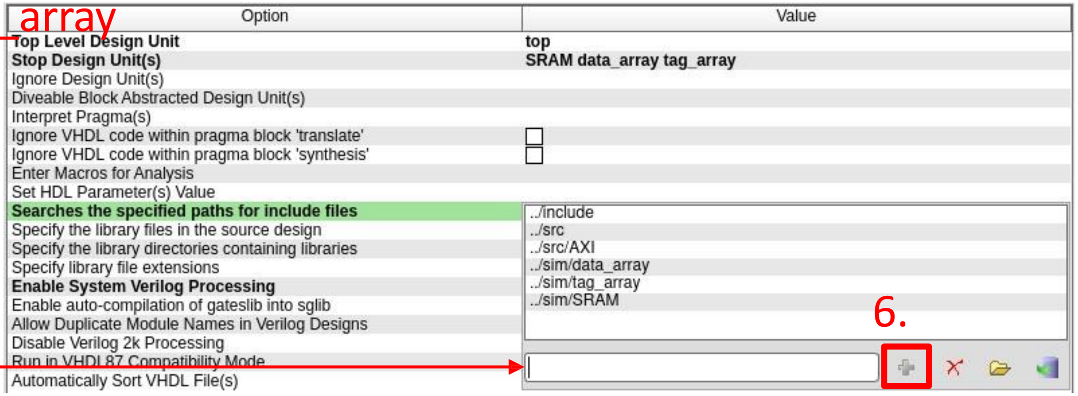
2.



Option	Value
Top Level Design Unit	top
Stop Design Unit(s)	SRAM data_array tag_array
Ignore Design Unit(s)	
Diveable Block Abstracted Design Unit(s)	
Interpret Pragma(s)	
Ignore VHDL code within pragma block 'translate'	
Ignore VHDL code within pragma block 'synthesis'	
Enter Macros for Analysis	
Set HDL Parameter(s) Value	
Searches the specified paths for include files	
Specify the library files in the source design	
Specify the library directories containing libraries	
Specify library file extensions	
Enable System Verilog Processing	☑
Enable auto-compilation of gateslib into sglib	□
Allow Duplicate Module Names in Verilog Designs	□
Disable Verilog 2k Processing	□
Run in VHDL87 Compatibility Mode	□
Automatically Sort VHDL File(s)	

3.
SRAM
data_array
tag_array

4.



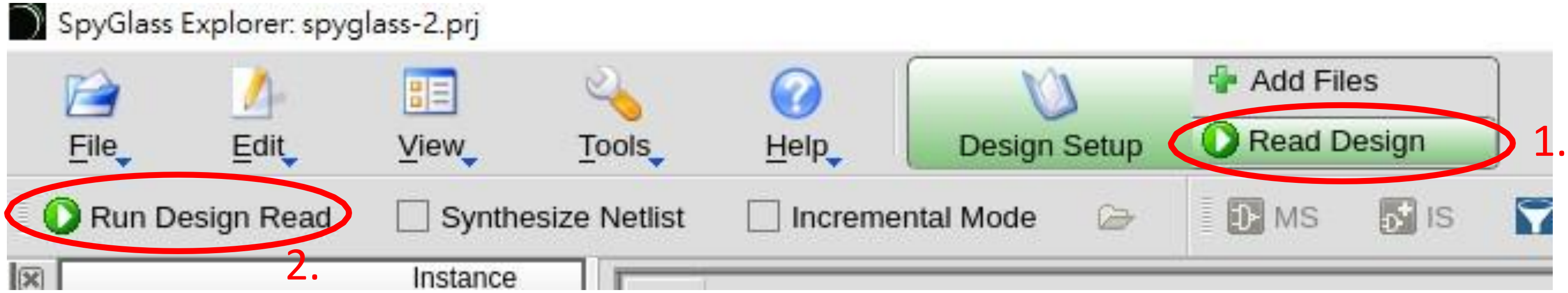
Option	Value
Top Level Design Unit	**top**
Stop Design Unit(s)	**SRAM data_array tag_array**
Ignore Design Unit(s)	
Diveable Block Abstracted Design Unit(s)	
Interpret Pragma(s)	
Ignore VHDL code within pragma block 'translate'	□
Ignore VHDL code within pragma block 'synthesis'	□
Enter Macros for Analysis	
Set HDL Parameter(s) Value	
Searches the specified paths for include files	../include ../src ../src/AXI ../sim/data_array ../sim/tag_array ../sim/SRAM
Specify the library files in the source design	
Specify the library directories containing libraries	
Specify library file extensions	
Enable System Verilog Processing	
Enable auto-compilation of gateslib into sglib	
Allow Duplicate Module Names in Verilog Designs	
Disable Verilog 2k Processing	
Run in VHDL87 Compatibility Mode	
Automatically Sort VHDL File(s)	

5.
Your include file path

6.

Read design

- Compile and analyze your design
  - current_goal Design_Read -alltop
  - link_design –force
  - Fatel, Error violations must be resolved in this state
- Every time you modify your design, you need to run “Read Design”

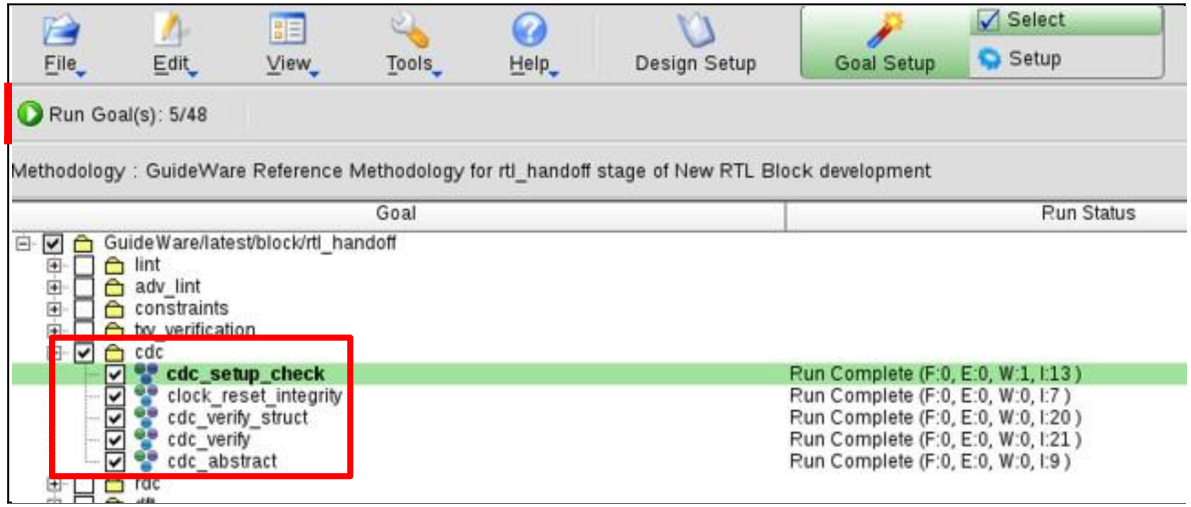


- Or you can source the tcl file at spyglass shell at this homework
  - “source ../script/Spyglass_CDC.tcl”

CDC verification

Start CDC check with spyglass

- Goal setup
  - Analyze a specific task “CDC”
- Run Goal
  - It will start to analyze your goal



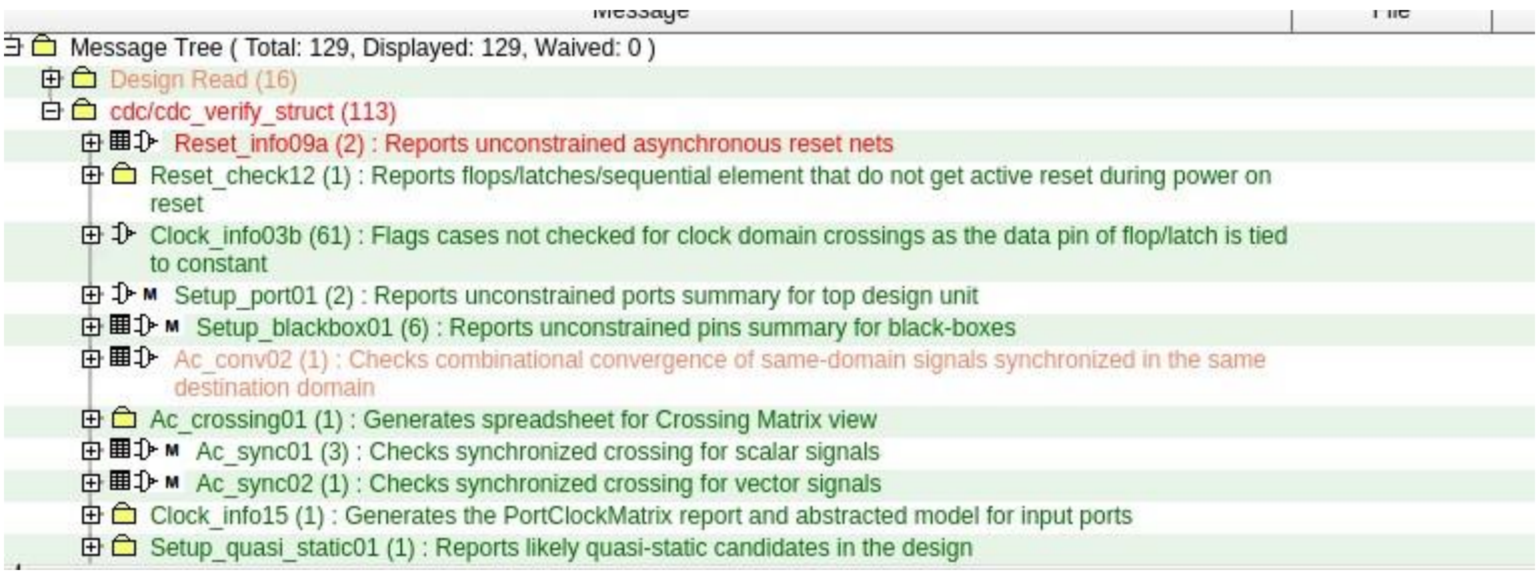
Run Goal(s): 5/48

Methodology : GuideWare Reference Methodology for rtl_handoff stage of New RTL Block development

Goal	Run Status
☑ GuideWare/latest/block/rtl_handoff	
□ lint	
□ adv_lint	
□ constraints	
□ txv_verification	
☑ cdc	
☑ **cdc_setup_check**	Run Complete (F:0, E:0, W:1, I:13)
☑ clock_reset_integrity	Run Complete (F:0, E:0, W:0, I:7)
☑ cdc_verify_struct	Run Complete (F:0, E:0, W:0, I:20)
☑ cdc_verify	Run Complete (F:0, E:0, W:0, I:21)
☑ cdc_abstract	Run Complete (F:0, E:0, W:0, I:9)
□ rdc	

Violation severity

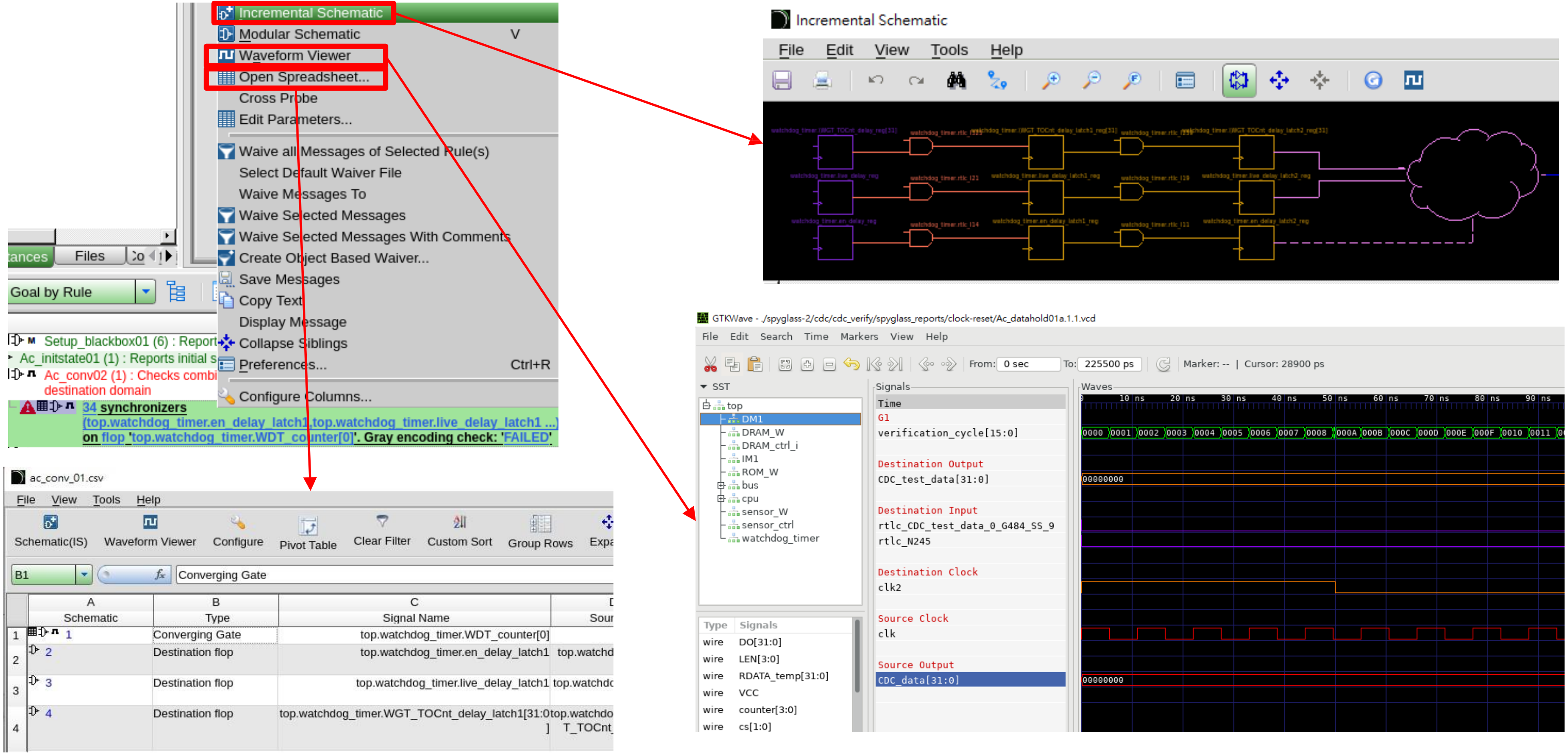
- Analyze results
  - Fatal
  - Error
  - Warning
  - Information
- Fatal, Error, warning violations must be resolved in this state



```
Message                                                                                      File
Message Tree ( Total: 129, Displayed: 129, Waived: 0 )
  Design Read (16)
  cdc/cdc_verify_struct (113)
    Reset_info09a (2) : Reports unconstrained asynchronous reset nets
    Reset_check12 (1) : Reports flops/latches/sequential element that do not get active reset during power on
    reset
    Clock_info03b (61) : Flags cases not checked for clock domain crossings as the data pin of flop/latch is tied
    to constant
    M Setup_port01 (2) : Reports unconstrained ports summary for top design unit
    M Setup_blackbox01 (6) : Reports unconstrained pins summary for black-boxes
    Ac_conv02 (1) : Checks combinational convergence of same-domain signals synchronized in the same
    destination domain
    Ac_crossing01 (1) : Generates spreadsheet for Crossing Matrix view
    M Ac_sync01 (3) : Checks synchronized crossing for scalar signals
    M Ac_sync02 (1) : Checks synchronized crossing for vector signals
    Clock_info15 (1) : Generates the PortClockMatrix report and abstracted model for input ports
    Setup_quasi_static01 (1) : Reports likely quasi-static candidates in the design
```

Debugging Violations

- Click “Incremental Schematic” to see simplified schematic
- Click “Open Spreadsheet” to see more detail reason
- Click “Waveform Viewer” to see failure signals



Setup_blackbox01 (6) : Report
Ac_initstate01 (1) : Reports initial s
Ac_conv02 (1) : Checks combi destination domain
34 synchronizers (top.watchdog_timer.en_delay_latch1,top.watchdog_timer.live_delay_latch1 ...) on flop 'top.watchdog_timer.WDT_counter[0]'. Gray encoding check: 'FAILED'

ac_conv_01.csv

B1 | Converging Gate

	A Schematic	B Type	C Signal Name	D Sour
1	1	Converging Gate	top.watchdog_timer.WDT_counter[0]	
2	2	Destination flop	top.watchdog_timer.en_delay_latch1	top.watchd
3	3	Destination flop	top.watchdog_timer.live_delay_latch1	top.watchdc
4	4	Destination flop	top.watchdog_timer.WGT_TOCnt_delay_latch1[31:0]	top.watchdo T_TOCnt_

GTKWave - ./spyglass-2/cdc/cdc_verify/spyglass_reports/clock-reset/Ac_datahold01a.1.1.vcd

From: 0 sec To: 225500 ps Marker: -- | Cursor: 28900 ps

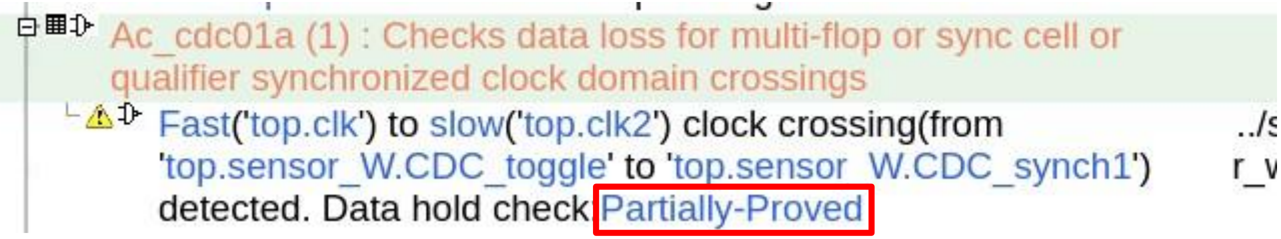
```
top
  DM1
    DRAM_W
    DRAM_ctrl_i
    IM1
    ROM_W
    bus
    cpu
    sensor_W
    sensor_ctrl
    watchdog_timer
```

Type	Signals
wire	DO[31:0]
wire	LEN[3:0]
wire	RDATA_temp[31:0]
wire	VCC
wire	counter[3:0]
wire	cs[1:0]

```
Time
G1
verification_cycle[15:0]        0000 0001 0002 0003 0004 0005 0006 0007 0008 000A 000B 000C 000D 000E 000F 0010 0011

Destination Output
CDC_test_data[31:0]             00000000

Destination Input
rtlc_CDC_test_data_0_G484_SS_9
rtlc_N245

Destination Clock
clk2

Source Clock
clk

Source Output
CDC_data[31:0]                  00000000
```

Partially-proved warning

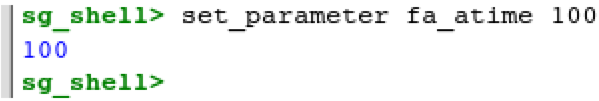
- Spyglass provides the number of cycles that have been explored during which no violation has been found.



Ac_cdc01a (1) : Checks data loss for multi-flop or sync cell or qualifier synchronized clock domain crossings

Fast('top.clk') to slow('top.clk2') clock crossing(from 'top.sensor_W.CDC_toggle' to 'top.sensor_W.CDC_synch1') detected. Data hold check Partially-Proved ../s r_v

- Set the fa_atime parameter to increase the amount of time that Spyglass spends on validating a single property.
  - “Set_parameter fa_time 100”



```
sg_shell> set_parameter fa_atime 100
100
sg_shell>
```