NCTU-EE IC LAB - Spring 2022

Lab09 Exercise

Verification: Pokemon Simulation Game (From Lab08)

Data Preparation

1. Extract test data from TA's directory:

```
% tar xvf ~iclabta01/Lab09.tar
```

2. The extracted LAB directory contains:

    Exercise/
    Practice/

Description

You need to write the verification pattern for the PSG (from Lab08). You need to complete following things:

1. PATTERN.sv (Lab09/Exercise /00_TESTBED/PATTERN.sv)
   Generate pattern data.
   Send pattern data to pokemon.sv and bridge.sv and make sure that it will achieve coverage goals.
   You also need to check the correctness of the output signals of the design.
2. CHECKER.sv (Lab09/Exercise /00_TESTBED/CHECKER.sv)
   Write your cover groups and assertions here.

Specifications

Coverage: (TA's DESIGN + TA's CHECKER + Your PATTERN)

至少要 120 次

1. Create a covergroup including coverpoint inf.out_info[31:28] and inf.out_info[27:24] (Player Pokemon info when action is not Attack; Defender Pokemon info when action is Attack).The bins of inf.out_info[31:28] needs to be No_stage, Lowest, Middle and Highest, respectively. The bins of inf.out_info[27:24] needs to be No_type, Grass, Fire, Water, Electric, Normal, respectively. Each bin should be hit at least 20 times. (sample the value at negedge clk when inf.out_valid is high)(Note: We will use exactly word like "Lowest" instead of number "4'd1")

   1 2 3 4 => 20

   1 2 3 4 5 1 => 120.

2. Create a covergroup including coverpoint inf.D.d_id[0] (means 0~7 bits of input signal D when typing your ID) with auto_bin_max = 256. (means that you need to divide the inf.D.d_id[0] signal into 256 bins averagely). And each bin has to be hit at least 1 time. (sample the value at posedge clk when id_valid is high)

   [illegible]每個選手都要

3. Create a covergroup including coverpoint inf.D.d_act[0] (means 0~3 bits of input signal D when typing your action). There are six actions for inf.D.d_act[0]: Buy, Sell, Deposit, Check, Use_item, Attack. Create the transition bins from one action to itself or others. such as: Buy to Buy, Buy to Sell, Buy to Deposit, Buy to Check, Buy to Use_item, Buy to Attack and so on. There are total 36

   所有 action 的組合

transition bins. Each transition bin should be hit at least 10 times. (sample the value at posedge clk when act_valid is high).

400次 要成功200次 error 200次

4. Create a covergroup including coverpoints inf.complete. The bins of inf.complete need to be 0 and 1, and each bin should be hit at least 200 times. (sample the value at negedge clk when inf.out_valid is high)

每個error出現20次

5. Create a covergroup including coverpoint inf.err_msg. Every case of inf.err_msg except No_Err should occur at least 20 times. (sample the value at negedge clk when inf.out_valid is high)

Notice:

1. When you send the pattern to the pokemon.sv, you need to follow the specs from the Lab08. For example, all input valid signals won't overlap with each other. You can write some assertions in your CHECKER.sv to check. If you violate the specs and your assertions didn't discover but TA discover during demo, you will fail.
2. After passing the last pattern, your PATTERN should finish immediately. (Still remember not to violate any specs.)
3. During demo, TA will also use wrong design to test if your pattern can check the correctness of the output signals of the design. When the answer is wrong, you should stop the program immediately and display "Wrong Answer" on the terminal.

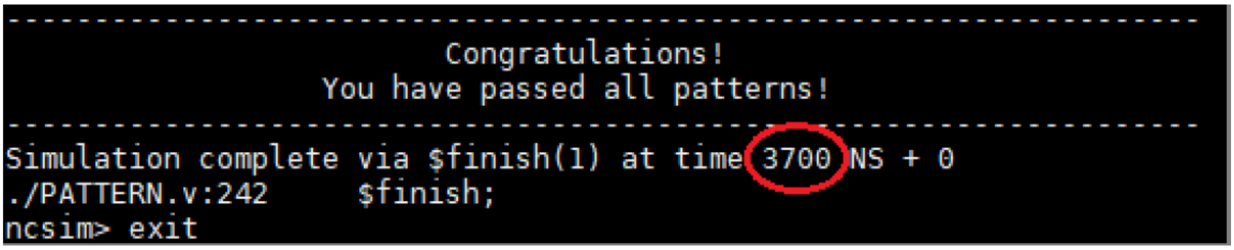
Assertion: (TA's DESIGN + TA's PATTERN+ Your CHECKER)

1. All outputs signals (including pokemon.sv and bridge.sv) should be zero after reset.
2. If action is completed, err_msg should be 4'b0.
3. If action is not completed, out_info should be 64'b0.
4. The gap between each input valid is at least 1 cycle and at most 5 cycles.
5. All input valid signals won't overlap with each other.
6. Out_valid can only be high for exactly one cycle.
7. Next operation will be valid 2-10 cycles after out_valid fall.
8. Latency should be less than 1200 cycles for each operation.

Notice that once the spec is violated, you should stop the program immediately and show the assertion message on the terminal. Your assertion warning messages should be "Assertion X is violated", where X is the number of assertions. You can directly copy the messages provided by TA in student.txt. The definition of cycle and latency is the same as Lab08.

Note

1. **Grading Policy**
   - **Coverage + assertion:** 70%
   - **Simulation time of coverage:** 30%



```
------------------------------------------------------------------
                    Congratulations!
               You have passed all patterns!
------------------------------------------------------------------
Simulation complete via $finish(1) at time 3700 NS + 0
./PATTERN.v:242     $finish;
ncsim> exit
```

-

You need to pass all specs and will get the simulation time score.

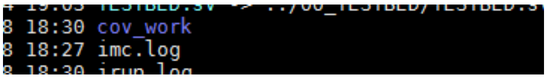
The second demo will be 30% off.

2. **Please upload the following file on newE3 platform before 23:59 on 5/8:**
   - PATTERN_iclabXXX.sv (ex: PATTERN_iclab099.sv)
   - CHECKER_iclabXXX.sv (ex: CHECKER_iclab099.sv)
   - dram_iclabXXX.dat (ex: dram_iclab099.dat)

     Remember using dram.dat when you read your data in pattern. Or you will be failed at demo. And if you have modified the Usertype_PKG.sv, please also upload that file too.
   - Usertype_PKG_iclabXXX.sv (ex: Usertype_PKG_iclab099.sv) (optional)
3. Since the purpose of this Lab is to use SystemVerilog to do verification. You should generate pattern in the PATTERN.sv directly instead of using read file method.
4. Don't use any wire/reg/submodule/parameter name called *error*, *congratulation*, *latch* or *fail* otherwise you will fail the lab. Note: * means any char in front of or behind the word. e.g: error_note is forbidden.
5. Please do not display any other information on the CHECKER.sv and PATTERN.sv except for the information written in the student.txt.

Using Cadence IMC (Incisive Metrics Center)

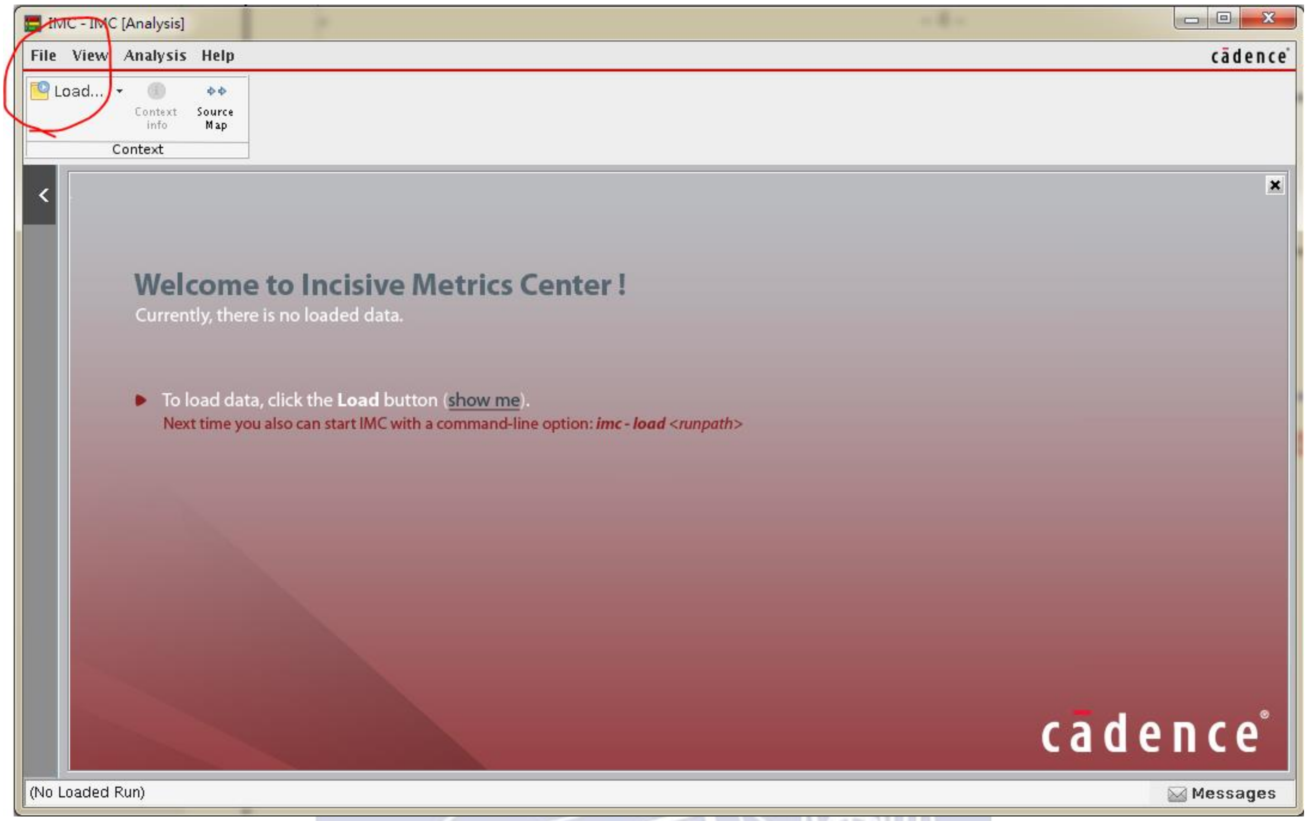
1. % irun TESTBED.sv -define RTL -debug -coverage U –covoverwrite (./02_run_cov)



```
4 19:05 TESTBED.sv -> ../00_TESTBED/TESTBED.s
8 18:30 cov_work
8 18:27 imc.log
8 18:30 irun.log
```

2. % imc &

3. Load the coverage database (default path: /01_RTL/cov_work/scope/test)



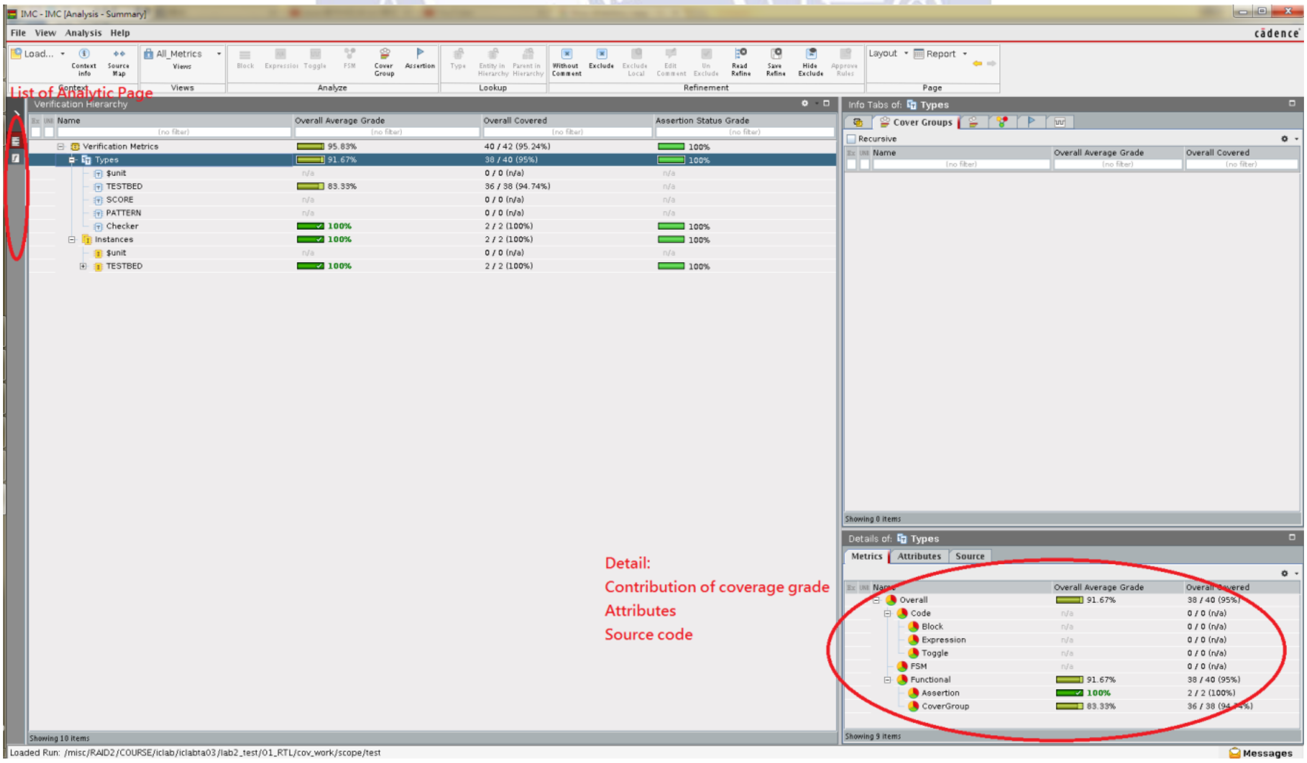
Welcome to Incisive Metrics Center !

Currently, there is no loaded data.

- To load data, click the **Load** button (show me).

Next time you also can start IMC with a command-line option: ***imc - load** <runpath>*

Panel



Name	Overall Average Grade	Overall Covered	Assertion Status Grade
(no filter)	(no filter)	(no filter)	(no filter)
Verification Metrics	95.83%	40 / 42 (95.24%)	100%
Types	91.67%	38 / 40 (95%)	100%
$unit	n/a	0 / 0 (n/a)	n/a
TESTBED	83.33%	36 / 38 (94.74%)	n/a
SCORE	n/a	0 / 0 (n/a)	n/a
PATTERN	n/a	0 / 0 (n/a)	n/a
Checker	100%	2 / 2 (100%)	100%
Instances	100%	2 / 2 (100%)	100%
$unit	n/a	0 / 0 (n/a)	n/a
TESTBED	100%	2 / 2 (100%)	100%

Showing 10 items

Name	Overall Average Grade	Overall Covered
(no filter)	(no filter)	(no filter)

Showing 0 items

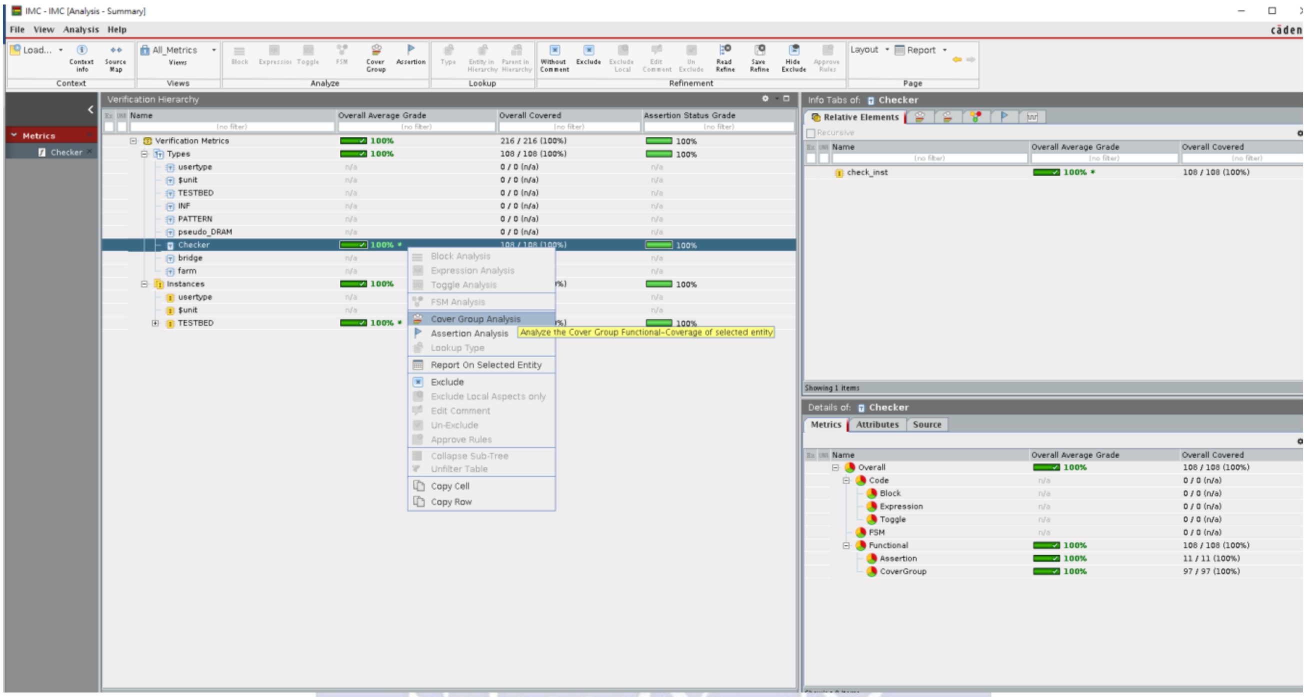
Name	Overall Average Grade	Overall Covered
Overall	91.67%	38 / 40 (95%)
Code	n/a	0 / 0 (n/a)
Block	n/a	0 / 0 (n/a)
Expression	n/a	0 / 0 (n/a)
Toggle	n/a	0 / 0 (n/a)
FSM	n/a	0 / 0 (n/a)
Functional	91.67%	38 / 40 (95%)
Assertion	100%	2 / 2 (100%)
CoverGroup	83.33%	36 / 38 (94.74%)

Showing 9 items

Detail:
Contribution of coverage grade
Attributes
Source code

Loaded Run: /misc/RAID2/COURSE/iclab/iclabta03/lab2_test/01_RTL/cov_work/scope/test

4. Covergroup analysis



IMC - IMC [Analysis - Summary]

File View Analysis Help
Load... Context info Source Map
All_Metrics Views
Block Expression Toggle FSM Cover Group Assertion
Type Entity in Hierarchy Parent in Hierarchy
Without Comment Exclude Exclude Local Edit Comment Un Exclude Read Refine Save Refine Hide Exclude Approve Rules
Layout Report
Context Views Analyze Lookup Refinement Page
Metrics
Checker

Verification Hierarchy

Name	Overall Average Grade	Overall Covered	Assertion Status Grade
Verification Metrics	100%	216 / 216 (100%)	100%
Types	100%	108 / 108 (100%)	100%
usertype	n/a	0 / 0 (n/a)	n/a
$unit	n/a	0 / 0 (n/a)	n/a
TESTBED	n/a	0 / 0 (n/a)	n/a
INF	n/a	0 / 0 (n/a)	n/a
PATTERN	n/a	0 / 0 (n/a)	n/a
pseudo_DRAM	n/a	0 / 0 (n/a)	n/a
Checker	100% *	108 / 108 (100%)	100%
bridge	n/a		n/a
farm	n/a		n/a
Instances	100%	%)	100%
usertype	n/a		n/a
$unit	n/a		n/a
TESTBED	100% *	%)	100%

Analyze the Cover Group Functional-Coverage of selected entity

Info Tabs of: Checker

Relative Elements

Name	Overall Average Grade	Overall Covered
check_inst	100% *	108 / 108 (100%)

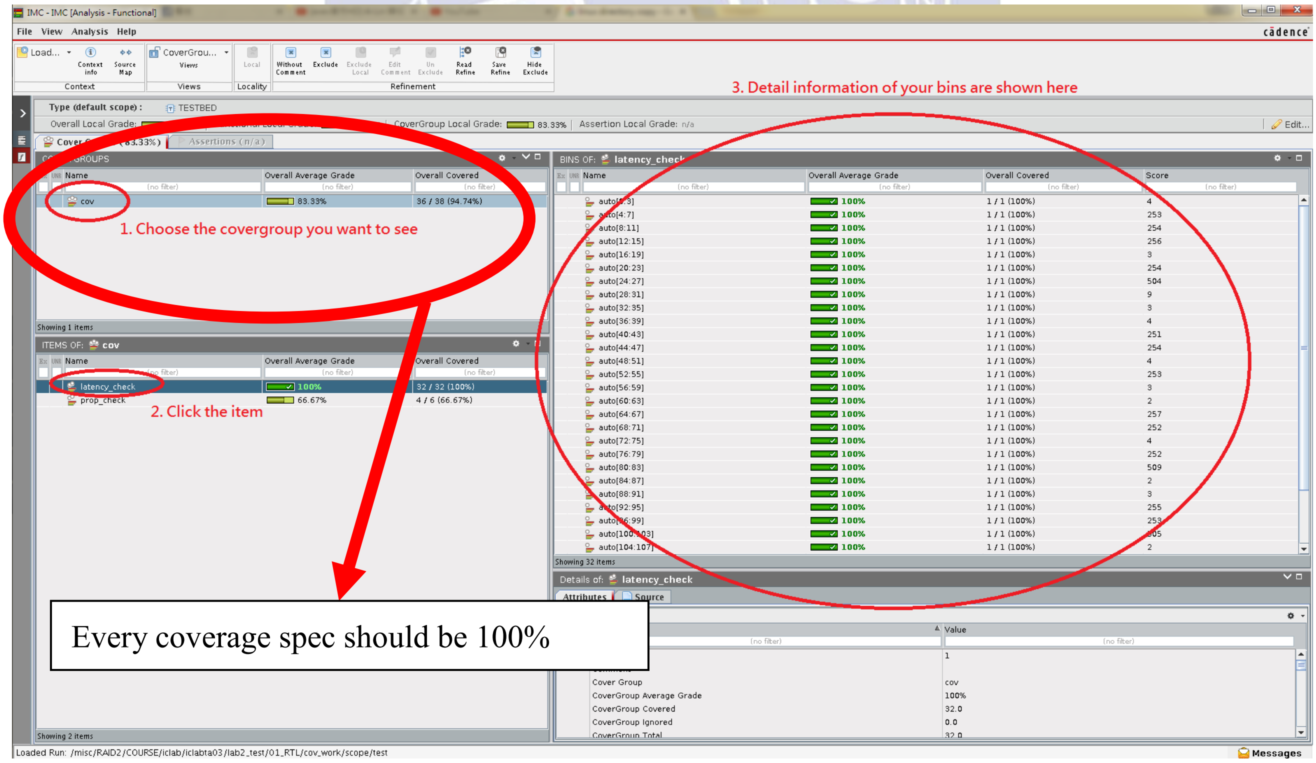
Showing 1 items

Details of: Checker

Metrics | Attributes | Source

Name	Overall Average Grade	Overall Covered
Overall	100%	108 / 108 (100%)
Code	n/a	0 / 0 (n/a)
Block	n/a	0 / 0 (n/a)
Expression	n/a	0 / 0 (n/a)
Toggle	n/a	0 / 0 (n/a)
FSM	n/a	0 / 0 (n/a)
Functional	100%	108 / 108 (100%)
Assertion	100%	11 / 11 (100%)
CoverGroup	100%	97 / 97 (100%)

5. Detail of items inside covergroup



IMC - IMC [Analysis - Functional]

File View Analysis Help
Load... Context info Source Map
CoverGrou... Views
Local
Without Comment Exclude Exclude Local Edit Comment Un Exclude Read Refine Save Refine Hide Exclude
Context Views Locality Refinement

Type (default scope) : TESTBED

Overall Local Grade: | CoverGroup Local Grade: 83.33% | Assertion Local Grade: n/a | Edit...

Cover (83.33%) | Assertions (n/a)

GROUPS

Name	Overall Average Grade	Overall Covered
cov	83.33%	36 / 38 (94.74%)

1. Choose the covergroup you want to see

Showing 1 items

ITEMS OF: cov

Name	Overall Average Grade	Overall Covered
latency_check	100%	32 / 32 (100%)
prop_check	66.67%	4 / 6 (66.67%)

2. Click the item

3. Detail information of your bins are shown here

BINS OF: latency_check

Name	Overall Average Grade	Overall Covered	Score
auto[0:3]	100%	1 / 1 (100%)	4
auto[4:7]	100%	1 / 1 (100%)	253
auto[8:11]	100%	1 / 1 (100%)	254
auto[12:15]	100%	1 / 1 (100%)	256
auto[16:19]	100%	1 / 1 (100%)	3
auto[20:23]	100%	1 / 1 (100%)	254
auto[24:27]	100%	1 / 1 (100%)	504
auto[28:31]	100%	1 / 1 (100%)	9
auto[32:35]	100%	1 / 1 (100%)	3
auto[36:39]	100%	1 / 1 (100%)	4
auto[40:43]	100%	1 / 1 (100%)	251
auto[44:47]	100%	1 / 1 (100%)	254
auto[48:51]	100%	1 / 1 (100%)	4
auto[52:55]	100%	1 / 1 (100%)	253
auto[56:59]	100%	1 / 1 (100%)	3
auto[60:63]	100%	1 / 1 (100%)	2
auto[64:67]	100%	1 / 1 (100%)	257
auto[68:71]	100%	1 / 1 (100%)	252
auto[72:75]	100%	1 / 1 (100%)	4
auto[76:79]	100%	1 / 1 (100%)	252
auto[80:83]	100%	1 / 1 (100%)	509
auto[84:87]	100%	1 / 1 (100%)	2
auto[88:91]	100%	1 / 1 (100%)	3
auto[92:95]	100%	1 / 1 (100%)	255
auto[96:99]	100%	1 / 1 (100%)	253
auto[100:103]	100%	1 / 1 (100%)	505
auto[104:107]	100%	1 / 1 (100%)	2

Showing 32 items

Details of: latency_check

Attributes | Source

	Value
	1
Cover Group	cov
CoverGroup Average Grade	100%
CoverGroup Covered	32.0
CoverGroup Ignored	0.0
CoverGroup Total	32.0

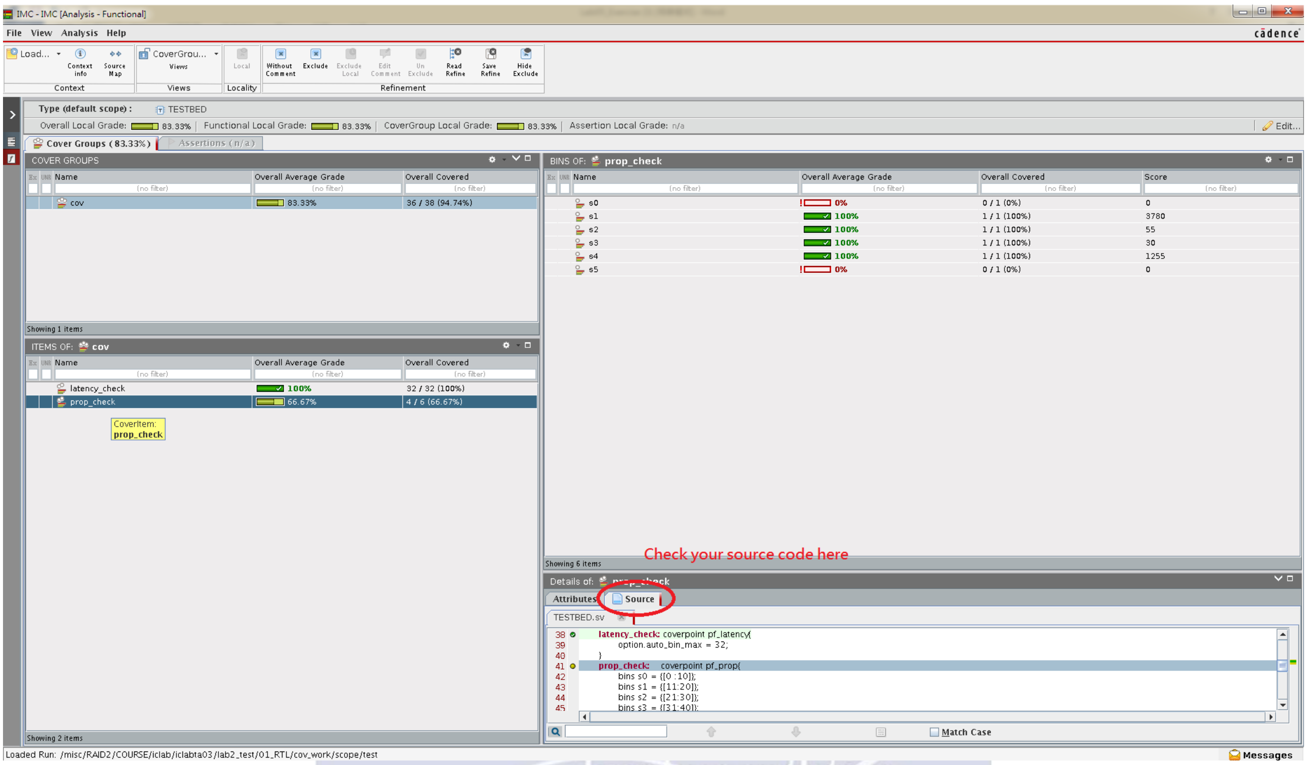
Showing 2 items

Loaded Run: /misc/RAID2/COURSE/iclab/iclabta03/lab2_test/01_RTL/cov_work/scope/test

Messages

Every coverage spec should be 100%

6. Check your source code


IMC - IMC [Analysis - Functional]
File View Analysis Help
cādence
Load... Context info Source Map
Context
CoverGrou... Views
Views
Local
Locality
Without Comment Exclude Exclude Local Edit Comment Un Exclude Read Refine Save Refine Hide Exclude
Refinement
Edit...
Attributes
Source
TESTBED.sv
Match Case
Messages

Type (default scope) : TESTBED

Overall Local Grade: 83.33% | Functional Local Grade: 83.33% | CoverGroup Local Grade: 83.33% | Assertion Local Grade: n/a

Cover Groups (83.33%) | Assertions (n/a)

COVER GROUPS

Ex	UNR	Name	Overall Average Grade	Overall Covered
		(no filter)	(no filter)	(no filter)
		cov	83.33%	36 / 38 (94.74%)

Showing 1 items

ITEMS OF: cov

Ex	UNR	Name	Overall Average Grade	Overall Covered
		(no filter)	(no filter)	(no filter)
		latency_check	✓ 100%	32 / 32 (100%)
		prop_check	66.67%	4 / 6 (66.67%)

CoverItem:
prop_check

Showing 2 items

BINS OF: prop_check

Ex	UNR	Name	Overall Average Grade	Overall Covered	Score
		(no filter)	(no filter)	(no filter)	(no filter)
		s0	! 0%	0 / 1 (0%)	0
		s1	✓ 100%	1 / 1 (100%)	3780
		s2	✓ 100%	1 / 1 (100%)	55
		s3	✓ 100%	1 / 1 (100%)	30
		s4	✓ 100%	1 / 1 (100%)	1255
		s5	! 0%	0 / 1 (0%)	0

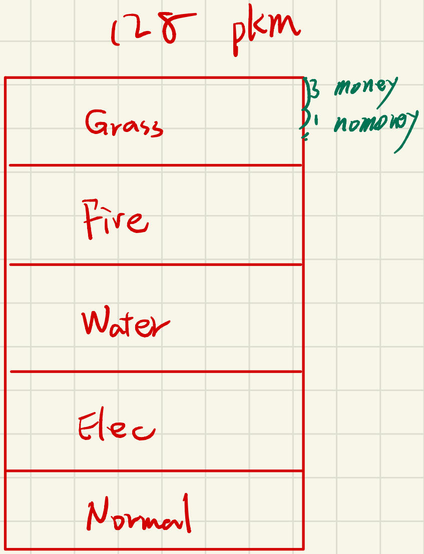
Showing 6 items

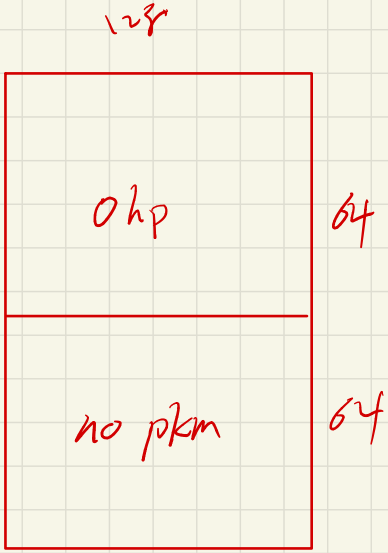
Check your source code here

Details of: prop_check

```
    latency_check: coverpoint pf_latency{
        option.auto_bin_max = 32;
    }
    prop_check:    coverpoint pf_prop{
        bins s0 = {[0 :10]};
        bins s1 = {[11:20]};
        bins s2 = {[21:30]};
        bins s3 = {[31:40]};
```

Loaded Run: /misc/RAID2/COURSE/iclab/iclabta03/lab2_test/01_RTL/cov_work/scope/test





11213141516

223242526

3343536

44546

556