NCTU-EE IC LAB - FALL 2019

Lab10 Exercise

Verification: CTS (From Lab09) Coverage and Assertion

Data Preparation

1. Extract test data from TA's directory:

   % tar xvf ~iclabta01/Lab10.tar

2. The extracted LAB directory contains:

   a. EXERCISE/

Description

you need to write the verification pattern for the CTS (from Lab09). You need to complete following things:

1. PATTERN.sv (Lab10/ EXERCISE /00_TESTBED/PATTERN.sv)
   Generate pattern data.
   Send pattern data to payment.sv and make sure that it will achieve coverage goals.
   You also need to do the reset first.
2. CHECKER.sv (Lab10/ EXERCISE/00_TESTBED/CHECKER.sv)
   Write your covergroups and assertions here.

Specifications

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

1. Create a covergroup including coverpoint **inf.D.d_pw[0].**

   There are four bins which range 0001~0011, 0100~0111, 1000~1011, 1100~1111

   in binary, respectively.

   Each bin should be hit at least 10 time.

   (sample the value at posedge clk when passwd_valid is high)

2. Create a covergroup including coverpoint **inf.D.d_id[0]** with auto_bin_max = 256.

   (means that you need to divide the ID 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)

3. Create a covergroup including coverpoint **inf.D.d_act[0]**

   There are five actions for **inf.D.d_act[0]:** trnsf, deposit, check_blc, change_pw, exit.

   Create the transition bins from one action to itself or others (except from exit to exit)

such as: trnsf to trnsf, trnsf to deposit, trnsf to check_blc and so on.

There are total 24 transition bins. Each transition bin should be hit at least 10 times.

(sample the value at posedge clk when act_valid is high)

4. Create a covergroup including coverpoint **inf.err_msg**

   Every case of **inf.err_msg** except No_Err should occur at least 10 times.

   (sample the value at negedge clk when inf.out_valid is high)

5. Create a covergroup including coverpoints **inf.complete** and **inf.out_balance**

   The bins of **inf.complete** need be 0 and 1 hit at least 100 times.

   The bins of **inf.out_balance** should range 1~17695, 17696~29492, 29493~36045, 36046~47842, 47843~65536, respectively. And each needs to be hit at least 20 times.

   Create a cross bin for the above **inf.complete** and **inf.out_balance**.

   Each combination should occur at least 20 times.

   (sample the value at negedge clk when inf.out_valid is high)

   Ex: (0 or 1) x (1~17695, 17696~29492, 29493~36045, 36046~47842, 47843~65536)

Note: When you send the pattern to the payment.sv, you need to follow the input specs from the Lab09.

For example, all outputs should be zero after reset.

You should write some assertion in your CHECKER.sv to check.

If you violate the following assertion, this part you would be fail.

And if you violate the specs and assertions didn't discover but TA discover during demo, you will also fail.

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

***You should hand in the code of this part.**

1. All outputs should be zero after reset.
2. If action is completed, inf.err_msg must be 4'b0.
3. When aciotn is trnsf, the gap length between id_valid and amnt_valid is at least 1 cycle.

Notice that your assertion warning messages should be "Assertion X is violated", where X is number.

Note

1. **Grading Policy**
   - Coverage 70%
     - Spec 1- 10%
     - Spec 2- 10%
     - Spec 3- 10%
     - Spec 4- 10%,
     - Spec 5- 10%
   - Simulation time 20%
   - Assertion 30%
     - Assertion 1- 10%
     - Assertion 2- 10%
     - Assertion 3- 10%

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

   The second demo will be 30% off. The highest second demo score will be 70.

2. **Please upload the following file on newE3 platform before 12:00 at noon on 12/9:**
   - PATTERN_iclabxx.sv, CHECKER_iclabxx.sv

   If you use .txt file to read your pattern, please also upload that file.

   Or you will be failed at demo.

   And if you have modified the Usertype_PKG.sv, please also upload that file too.

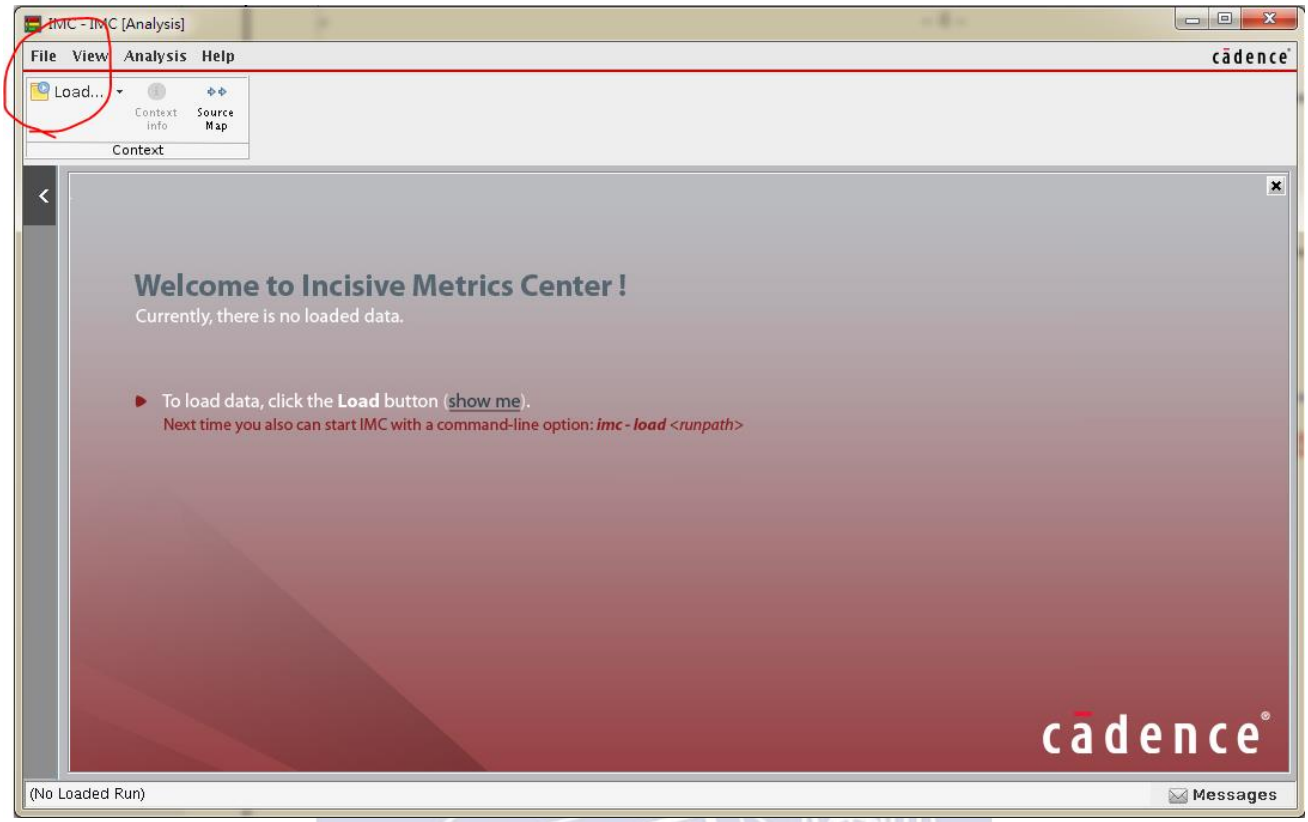
   (Usertype_PKG_iclabXX.sv)

Using Cadence IMC (Incisive Metrics Center)

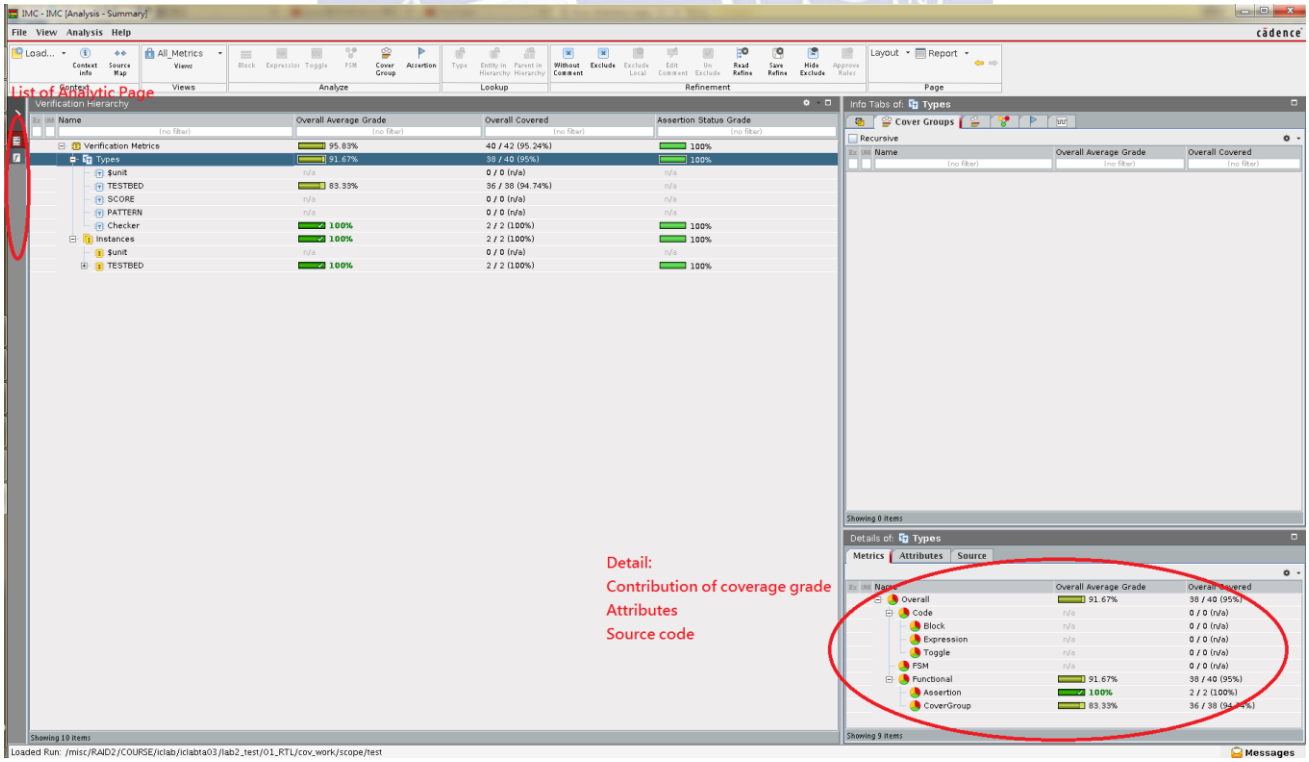
1. % irun TESTBED.sv -define RTL -debug -coverage U –covoverwrite (./02_run_cov)

2. % imc &

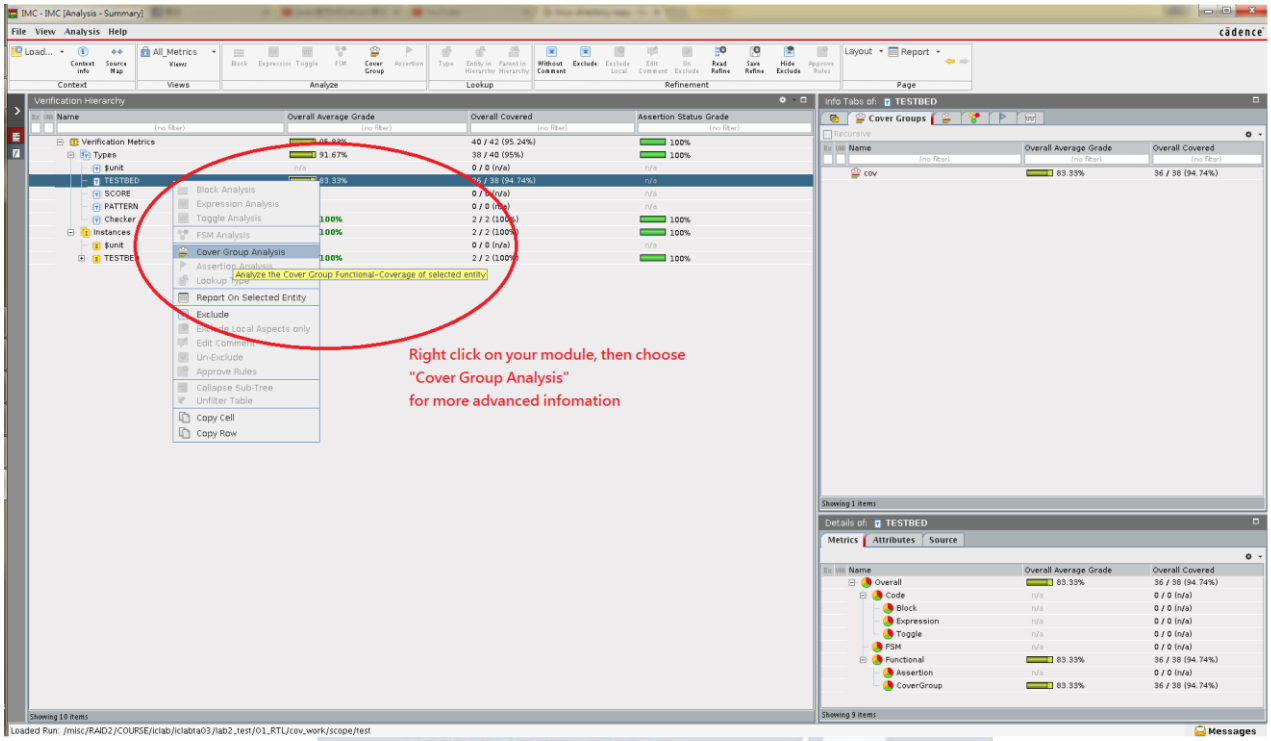
3. Load the coverage database (default path: /01_RTL/cov_work/scope/test)



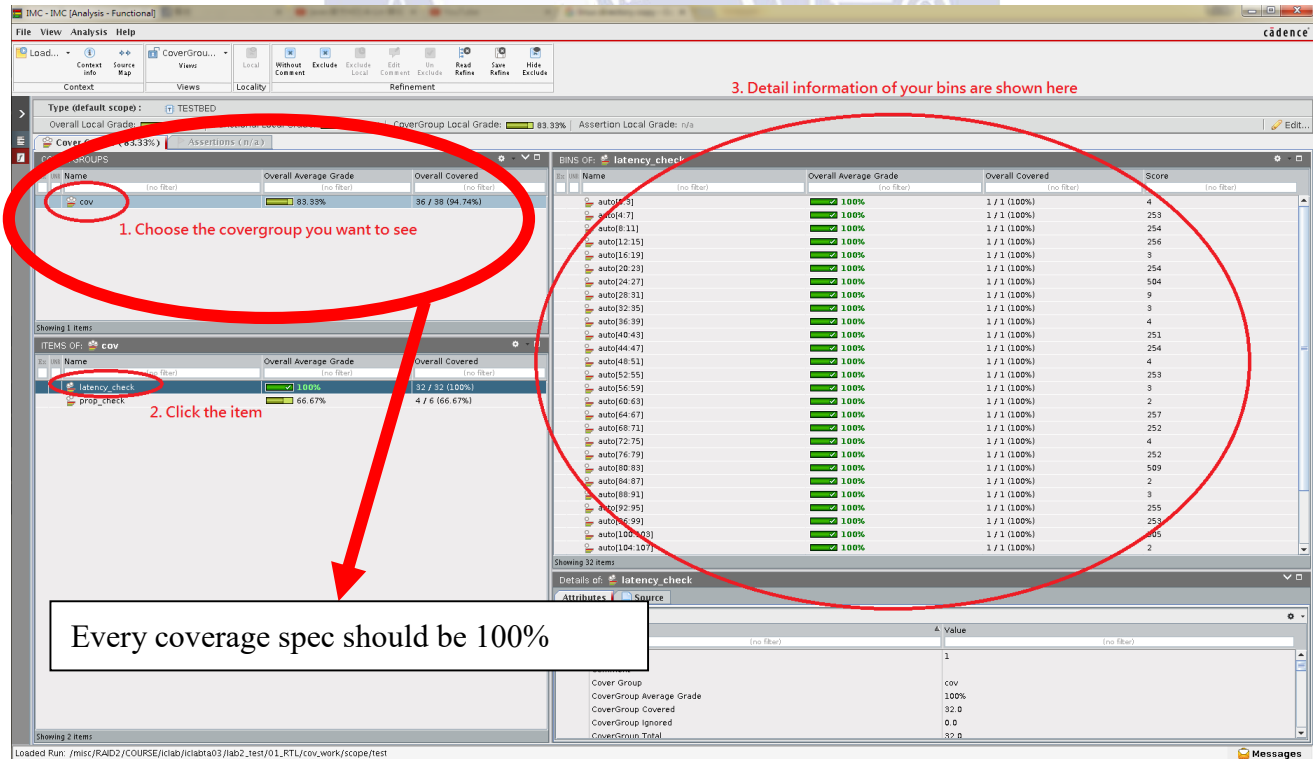
Panel



4. Covergroup analysis



5. Detail of items inside covergroup



6. Check your source code