**UG for polestar\_socenv testbench**

simulation environment for reference

Version 0.2

Jinfeng.Huang, 2018/01/17

UVM based testbench trunk/dv/polestar\_socenv refers to Sirius SoC env testbench, it developed as reference for IP owners/verifiers to create their own scenarios and testcases. This document would give a brief introduction to the directory architecture and on how to use the environment:

BTW, you can also has a quick skill by read doc/README file.

1. **Directory description**./bin/vcm.cfg : used to exclude or include coverage of some block; mask some uncovered

statement

./env: including all uvm components and objects

./assert : assertion related

./interface : interface related

./param\_def : define options and plusargs for `define or dummy some block

./register\_model: register model related

./stim: all soc level sequences, used by (virtual) sequencer

./uvc: all block uvm verification components, sometimes will be a block level env

./flist: link to filelist directory

./lib: c model files included

./model: link to model directory

./script: all needed perl scripts

./sim: simulation directory

./dump: dump file produce directory

./out:testcase running and log produce directory

./CovData\_$today: coverage produce directory

./out-$today: regression produce directory

./tb: link to testbench directory

./testcase: all testcases included (the rule for the testcase name is constrained, details in

file script/add\_test.pl)

./test\_ddr\_\*

./test\_spi\_\*

./test\_mipi\_\*

./...

1. **Running User guide**

Help:

>> vcs\_uvm.pl -h

Run a testcase:

Two methods:

First method(every testcase has own compile database)

>> vcs\_uvm.pl -sel -r

>> “select the index of testcase\_name you want to run”

Second method(every testcase use common compile database)

>> vcs\_uvm.pl -sel -c

>> vcs\_uvm.pl -sel -s

>> “select the index of testcase\_name you want to run”

ADD a testcase

>> vcs\_uvm.pl -add=”testcase\_name”

This will add an extended testcase in related dir and include it in test\_lib\_pkg.svh (sometimes you may need to modified it accordingly)

OPTIONS

The define options file is env/param\_def/myplusargs (remember to add the define options line by line or separated by blank); the block off options are defined in file env/param\_def/mydefine.v and env/param\_def/mydut.v which are inherited from Sirius’ env.

DO REGRESSION

>> vcs\_uvm.pl -b batch.f -round=N

batch.f is customized regression testcase list, N is the round for the regression;

It will mkdir a directory related with date and round. Take date=2018-01-03, and round=2 as example, it will mkdir directory out-2018-01-03 under directory sim/ and dump file will automatically shut off, under out-2018-01-03, each testcase will have directories for running name such as {testcase\_name}\_r0 for round 0 and {testcase\_name}\_r1 for round 1;

COVERAGE

When you need to open the coverage, please add the option “-cov” in the command to open it, the result will produced in directory sim/CovData\_$today

After the simulation, trans the database to html file by command “urg -dir \*\*\*.vdb”;

Note: the coverage type is added by “-cm <options>” (line/path/cond/tgl/fsm)

The configuration file is added by “-cm\_hier <config file>”

(+/-tree instance\_name [level number])

BUILD a testcase:

You should build the customized testcase under directory ./testcase/\*\*\*\_test/\*, you can refer to the exist testcases; then add the testcase to package file ./env/polestar\_test\_lib\_pkg.svh (a automatic script will be added later).

Compile and run options:

You can modified the define options file is in env/param\_def/myplusargs (remember to add the define options line by line or separated by blank), the block off options file are env/param\_def/mydefine.v and env/param\_def/mydut.v which are inherited from Sirius’s env;

DUMP:

You can modified your dump hierarchy in tb/dump.sv file;

The fsdb file is in sim/dump/{testcase\_name}\_00\*.fsdb

LOG file

Running a testcase is actually in directory sim/out/{testcase\_name}

Log file is produced in sim/out/{testcase\_name}/vcs\_sim.log

PASS or FAIL

Pass or fail will print in blue or red info separately.

LOAD code to filelist

You can load the filelist to verdi by typing “load” in directory sim/

Tags

Forward: “ctrl + ]”

Backward:”ctrl + t”