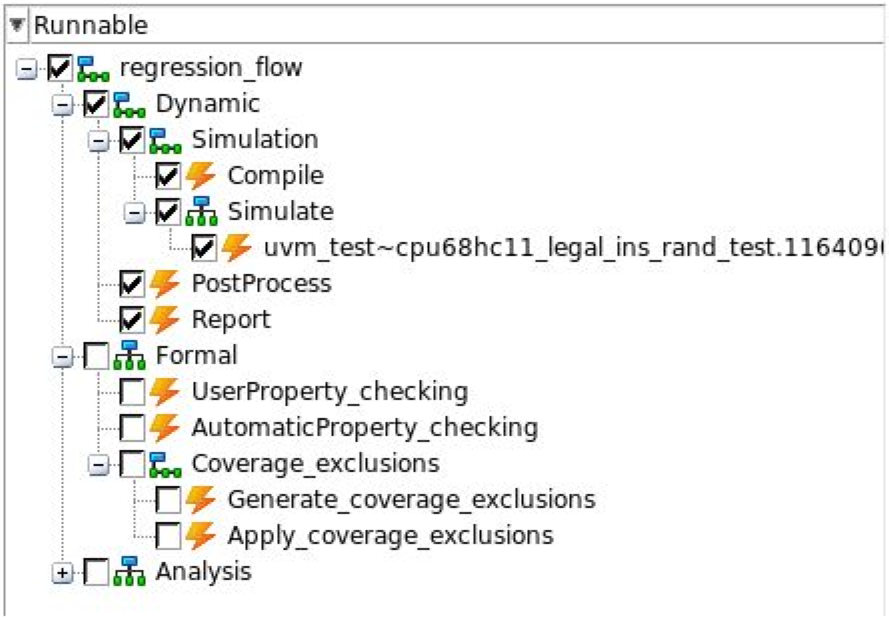
VRM App Layer User Manual

Table of contents

Introduction

this document intends to document the use model of the “generic” RMDB that I developed to mimic what should/could be an app layer provided to our end users as part of the Questa VRM tool and facilitate its adoption. Eventually it aims at serving projects with minimum customization/parameterization.
As of today the flow is as depicted below:



```
▾ Runnable
⊟ ☑ regression_flow
   ⊟ ☑ Dynamic
      ⊟ ☑ Simulation
            ☑ Compile
         ⊟ ☑ Simulate
               ☑ uvm_test~cpu68hc11_legal_ins_rand_test.116409
         ☑ PostProcess
         ☑ Report
   ⊟ ☐ Formal
         ☐ UserProperty_checking
         ☐ AutomaticProperty_checking
      ⊟ ☐ Coverage_exclusions
            ☐ Generate_coverage_exclusions
            ☐ Apply_coverage_exclusions
   ⊞ ☐ Analysis
```

there are 3 independent child runnables of the top runnable “regression_flow” :

1. Dynamic
   a. take care of running all regression steps
2. Formal
   a. take care of running formal tools
3. Analysis
   a. placeholder for runnables other than simulation such as profiling or specific post processing that we don’t want to achieve after each regression run

Known limitations

the current app layer has the following limitations:

- doesn’t take care of compile scripts
  - a template Makefile is provided which is generic enough to be reused but is not part of the delivery
  - to be implemented by the end user using makefile or any other ways of his preference, he will just need to have a top makefile implementing the target all calling his own scripts

- only support Questa tool suite
- only support UVM tests
    - support for non UVM tests may be added later
    - shall not need huge work/redesign of RMDB
- doesn't implement the high level layer/UI, that shall be done by engineering with a wizard and/or ui

Prerequisites

there are certain prerequisites for the app layer to work properly.
TODO put here the prerequisites on environment, directory structure, etc …

Quickstart guide

this is a quickstart guide to get a regression flow up and running quickly with minimum customization but as well minimum effort. You can refer to the other sections of the user guide to customize further the regression environment and add additional features if required.

Setup your regression

step 1 Setup your compile flow

The end user has the choice of using VRM for the compile flow or do the compile on his own and untick the compile runnable. It is up to the user to use the provided Makefile template or use his own compile flow.
if one choose to use the Makefile then Makefile parameters shall be overriden at make invocation or template Makefile may be edited and modified to set the variables at the appropriate values for the project.
See section Makefile parameter section for more details.

step 2 Setup your list of tests

Tests to be run by the regression must be captured into a file, called a test file. The test file follows a csv like file format. It follows a determined format as defined below:

	testname	**repeat count**	**seed**
level	***compulsory***	***compulsory***	***optional*** (set to ‘random’ by default)
description	name of your test	# of iterations	list of seeds space separated

The test file is basically a list of tests with their associated options (seed, # of repeat). 2 examples of test file are shown below.
The 1st one define a set of directed and random tests used fir running tests to reach coverage. All tests leave the seeds empty and thus set it as random.

```
# File Syntax is
# <testname> <repeat_count> <1st seed>...<nth seed>
# If not enough seeds then random is used to pad seeds.

#directed tests
ace_rw_generic_test 1
ace_rw_generic_reordering_test 1
ace_rw_phase_test 1
ace_rw_txn_test 1
ace_rw_txn_len_size_incr_test 1
ace_generic_test 1
```

```
ace_txn_test 1

#random tests
ace_rw_txn_system_random_test 1
ace_rw_txn_nonshareable_random_test 1
ace_rw_txn_innershareable_random_test 1
ace_rw_txn_outershareable_random_test 1
ace_rw_txn_random_test 16
```

the 2nd one shows a test file of a regression running only the contributing tests, it has been generated automatically after analyzing all tests contribution and rebuilding the test file from there. not that each test now has a defined seed number and is only run once.

```
ace_rw_txn_random_test 1 1857278929
ace_rw_txn_random_test 1 1356686004
ace_rw_txn_random_test 1 1987789029
ace_rw_txn_random_test 1 950649920
ace_rw_txn_random_test 1 44670287
ace_rw_txn_random_test 1 206765227
ace_rw_txn_random_test 1 1183696954
ace_rw_txn_random_test 1 1077398618
ace_rw_txn_random_test 1 1973792859
ace_rw_txn_random_test 1 54993495
ace_rw_txn_random_test 1 59206403
ace_rw_txn_random_test 1 830722182
ace_rw_txn_random_test 1 500472126
ace_rw_txn_random_test 1 1069218584
ace_rw_txn_random_test 1 1632976657
ace_rw_txn_random_test 1 632504634
ace_rw_txn_random_test 1 1860941171
ace_rw_txn_random_test 1 1972545130
ace_rw_txn_random_test 1 2074607036
ace_rw_txn_random_test 1 344078625
ace_rw_txn_random_test 1 581068967
ace_rw_txn_random_test 1 651866545
ace_rw_txn_random_test 1 873574746
ace_rw_txn_random_test 1 886426089
```

step 3 Setup your regression run

Last step is to setup the minimal set of parameters required by Questa VRM to run properly. The list of required parameters are below:

description	name	default
filename of test file containing the list of tests	testfile	none
path to the test file	testfilePath	none
path to Questa library ini file	MODELSIMINI	modelsim.ini in run

		directory
path to snapshot to simulate	SNAPHOT	none
command run prior vsim invocation	vsimPrecommand	none
vsim invocation options	vsimoptions	-uvmcontrol=all -msgmode both -classdebug -mvchome $QUESTA_MVC_HOME -do "run.do" -coverage

If you use the Makefile and/or want to trend code lines and DUT versions changes, there are additional parameters required below:

description	name	default
index for DUT version release ***(must be an integer)***	PRJ_DUT_VERSION	none
path to the root directory to DUT sources	PRJ_DUT_SRC_ROOT	none
path to the root directory to TB sources	PRJ_TB_SRC_ROOT	none

That section only show required parameters, optional parameters are discussed in more details into section “VRM configuration parameters”.
To set the parameters via the GUI refer to chapter “Adding New Configurations to the Project File” and “Edit VRM Configurations” of Questa VRM documentation.
To set the parameters via the command line refer to chapter “Override Parameter Values from Command Line”of Questa VRM documentation.

Run your regression

refer here to Questa VRM command line and GUI to launch the regression with the parameters setup in the previous section.
will be done later as that doesn’t add value

Look at regression results

TODO document here the outputs generated by the regression and where they are located
the regrssion run generate a number of outputs that can be analyzed at the end or in the course of the regression
TODO put the exact reference and filename of the files depicted below

Ranking report

after all tests are ran, a ranking process is launched and provide the following outputs:

- list of contributing tests
- list of non contributing test
- optimized regression list which allow to rerun a regression with only contributing tests

Coverage report

after all tests are ran a coverage report in HTML format is generated and strored under <regression dir>/report/coverage. refer to Questa user manual on “coverage report” for further details.

Trend report

After each regression run a trending report in HTML format is generated and strored under <regression dir>/report/trend. Refer to Questa user manual on “trend report” for further details.

merged coverage of current regression

at the end of the regression run, the merged coverage of the regression is available under <regression dir> and is saved as well under <regression dir>/logs with a timestamp suffix. the 1st merged coverage can be used to check the coverage of the specific regression, do analysis query (which test contributed to what, etc …) while the latter is saved to make sure that one can go back and do these queries even after a regression clean that delete all datas under <regression dir> except for the logs directory contents.

merged coverage of all regressions

at the end of regression run, the coverage result of the regression is merged with teh previosu regressions result in <regression dir>/logs. that enables one to have the merged coverage of all regressions run from the beginning of the project till present.

trend coverage of all regressions

at the end of regression run, the trendable coverage result of the regression is merged with the trend coverage file to track the regression trend.

Coverage exclusions

Summary of regression results

TODO put here as a table all the files above with names and location

description	name	location
merged coverage	<reg name>_merge.ucdb	<regression dir>

Advanced Customization

Makefile parameters

TODO will do that at the end as this is not the added value of the app layer

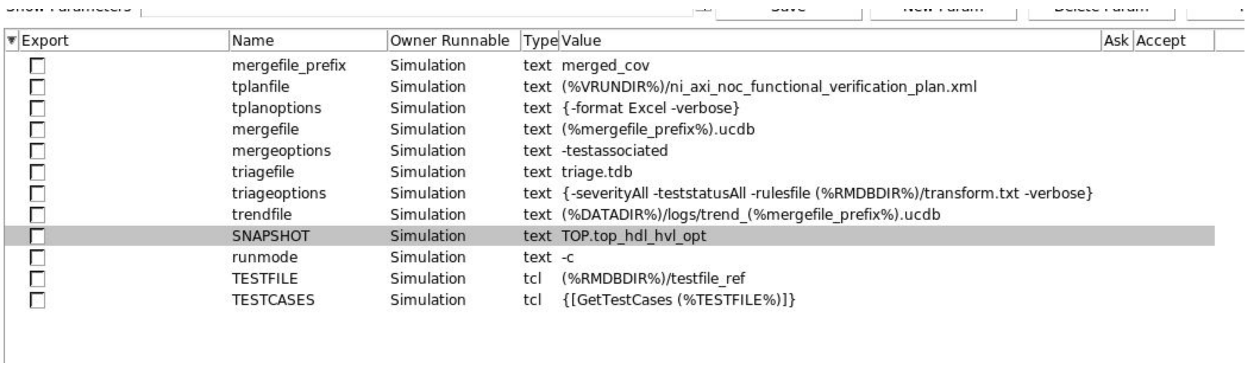
Testfile parameters

TODO put here the details of testfile format

VRM configuration parameters

TODO describe all configurable parameters of RMDB

TODO remove snapshots below when done



Export	Name	Owner Runnable	Type	Value	Ask	Accept
□	mergefile_prefix	Simulation	text	merged_cov		
□	tplanfile	Simulation	text	(%VRUNDIR%)/ni_axi_noc_functional_verification_plan.xml		
□	tplanoptions	Simulation	text	{-format Excel -verbose}		
□	mergefile	Simulation	text	(%mergefile_prefix%).ucdb		
□	mergeoptions	Simulation	text	-testassociated		
□	triagefile	Simulation	text	triage.tdb		
□	triageoptions	Simulation	text	{-severityAll -teststatusAll -rulesfile (%RMDBDIR%)/transform.txt -verbose}		
□	trendfile	Simulation	text	(%DATADIR%)/logs/trend_(%mergefile_prefix%).ucdb		
□	SNAPSHOT	Simulation	text	TOP.top_hdl_hvl_opt		
□	runmode	Simulation	text	-c		
□	TESTFILE	Simulation	tcl	(%RMDBDIR%)/testfile_ref		
□	TESTCASES	Simulation	tcl	{[GetTestCases (%TESTFILE%)]}		

Coverage exclusions

Profiling

Optional Steps

apply coverage exclusions