Mini spec for VRM Application Layer

Introduction

following a meeting in April 2014, the idea is to define together with Darron/Gabriel a way of providing an application layer for quicker adoption/deployment of VRM. the general idea is to provide together with VRM a set of RMDB files that will define a flow that should enable to have most of our prospects to get up and running.

Use model

That section describes the use model of what the application layer shall be. so basically steps the user will need to proceed to use the application layer and get up and running.
we should document here what the flow will be.

this is what the flow will be like:

1. **create a test file** capturing the list of testnames to be ran
   a. create vsim -f file (user specificied)
   b. create run.do file (default + user specified)
2. **create a global configuration file** that documents the options
   a. need to define here what we want to put visible to the end user
3. **invoke vrun with test file/configuration file as arguments** + a specific option for application layer
4. **iterate through 1-2-3** or
5. edit generated RMDB file and **start working on your own**

shall we do something around runtime configuration? (timeout/rerun strategy etc …)

Requirements

the way of implementing htis application layer is to provide out of the box at least the following functionalities:

1. **compile design and testbench**
2. **run simulation**
3. **generate post simulation report**
4. **implement a way of capturing dependencies?**
   a. using VRM file to pass the different configuration?

each of these functionalities will be spec out and detailed in the following sections. We want to be directive and not to generic/open, for people who want high customization they should go on their own and are not the target for this application layer.
Though we should have some configurability to ensure that the minimum extension can be done without writing your own RMDB

Compile Design & Testbench

After a little bit of thinking, I would tend to have that runnable fully implemented by the user as this is to specific tobe generic. we can still provide an example makefile but productizing it seem difficult and I believe low value … we should focus on simualtion and post processing.

~~the generic runnable can be derived from the VRM FAQ. we will use the compile runnable from FAQ example reference in section References at FAQ RMDB used for Compile Design & Testbench .~~
~~Compiling implementation within the RMDB as low value, more than implementation the only added value is to have the dependency, i.e. dont run regression if compilation failed, and control of the task, i.e. select the compile runnable. Thus we are going to use a makefile to implement the compilation itself and an RMDB base just to launch the Make command. In this way we can be flexible and adapt to existing customer Makefiles for instance which is quite common across customers.~~
~~so the application layer for compiling will be a set of 2 deliverables:~~
~~* include Makefile to compile and optimize the design~~
~~* RMDB include file to launch the Makefile command~~

~~References~~

~~FAQ RMDB used for Compile Design & Testbench~~

~~**SimpleMake**~~

~~I have a very simple make file and I've attempted to covert it to an rmdb so that I can use VRM for the same purpose. The rmdb only has two runnables, one to build and one to simulate. VRM errors out with the build runnable so never gets to simulation, what am I doin wrong ?~~
~~Answer~~

~~**UsingMake**~~

~~I currently have a complete environment compiling and simulating using make with a makefile. Can VRM call my make file and re-use some of the investment I've already made in scripting ?~~
~~Answer~~
~~RMDB~~

Run Simulation

the generic runnable can be derived from the VRM FAQ. Here we shouldn’t rely on Makefiles but rather have a full RMDB implementation and use the test centric approach. it will be in charge of extracting the testnames from a user input and create the list of tests runnable.

it will have to be configurable to specify things such as how many test in parralel, timeout, etc.
The use model is that user shall provide:

- testfile list input file
- option list input file
    - need to define the format

Generate Simulation Report

TBD

Implementation and use model

we need to define where the app stop in terms of features and where the end user will have to edit the RMDB.

features to put in RMDB:

- launch simulation from testfile
- merging coverage
- ranking coverage
- reporting send email on reg completion
- others to be extracted from already existing FAQ?
    - could be implemented by using RMDB include?