----Oana Botez & Petronela Ezaru-----

Regression script

The run_regress.pl script was made so developers and testers can verify a hole directory of tests and see the results in HTML pages.

The regression script is written in Perl and can be run with different options:

run_regress.pl -hdl <name> [-www] [-results_dir <path2>] [-no_exec] [-errors_only] [-dir_filter <directory_name>] [-h] [-version] [-v] [-val] [-no_post_compile] [-no_post_compile_cpp]

Note: The variable WORK is the path to your home folder (must be defined in .bash_profile)

ex: WORK=HOME/svn or WORK=HOME/fpl/cslc (no final slash)

- -hdl <name> -> only tests from hdl named <name> are chosen. <name> can be one of the following:

  verilog : run the Verilog verilog tests
  csl : run the CSL csl tests (golden test suite).
  csl_v2 : run the CSL csl_v2 tests (CSL version 2 tests)
  csl_new_bug : run the CSL csl_new_bug tests (tests that should be verified and promoted to csl golden)
  csl_test_gen : run the CSL csl_test_gen tests (Perl script generators)
  csl_design_gen : run the CSL csl_design_gen tests (designGen.cpp)
  csl_cc_gen : run the CSL csl_cc_gen tests (Perl script corner case generators)
  csl_ar_gen : run the CSL csl_ar_gen tests (Perl script corner case generators)

- -dir_filter <filter> -> only run the regression script on the specified directory
- -errors_only -> only display the failed tests in the HTML output
- -major_errors_only -> only display the failed tests with MAJOR ERRORS in the HTML output
- -www -> pops up firefox with the html results
- -no_post_compile -> Do not compile the generated code
- -no_post_compile_cpp -> Do not compile the CPP and CSIM CPP generated code
- -no_post_compile_systemc -> Do not compile the SystemC CPP generated code
- -results_dir <path2> -> saves the current results files to <path2>
- -no_exec -> do not run cslc; the regression generates the HTML report using the last version of the results files
- -h -> display help info
- -version -> prints version
- -v -> verbosity on
- -val -> run valgrind to detect memory leaks

E.g. : ./run_regress.pl -hdl csl

Note: this runs the golden regression (where all the tests are passing)

The run_regress.pl script contains subroutines that do the following:

- processing the external arguments (options) and if an invalid argument is used an error is shown
- testing if the hdl directory(<name>) is valid
- calling the regression for a specific directory
- creating an array with all the tests from the specified directory
- compiling the entire test suite and creating the compiler log file, which contains the result of the compilation
- checking if any error has occurred (warning, minor error, major error)

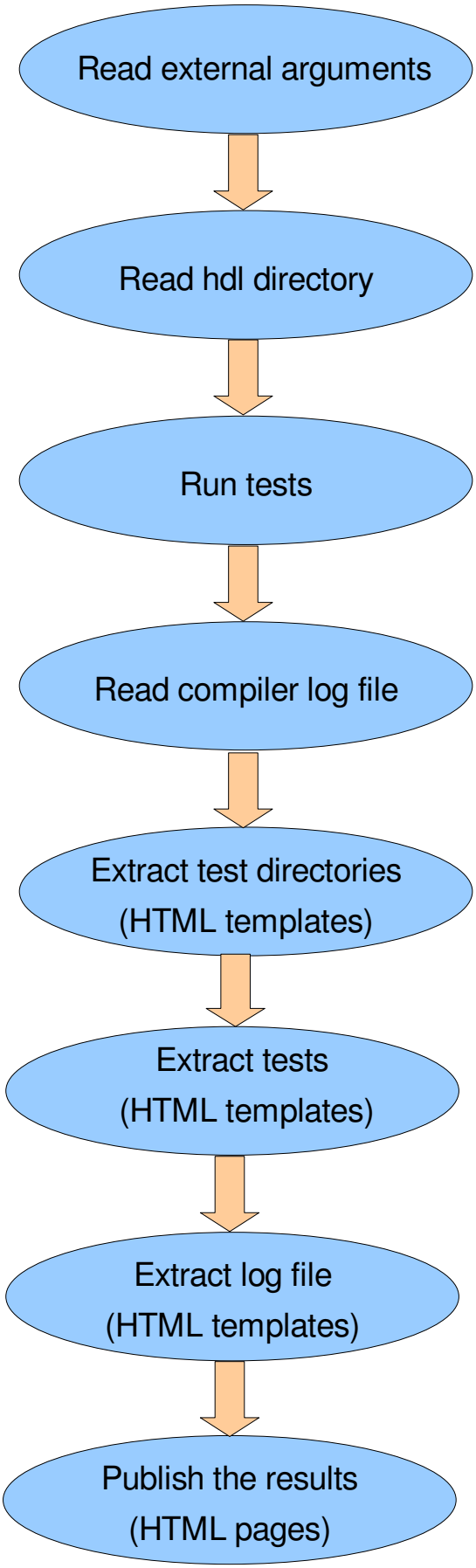
The regression script is using HTML templates :summary.tmpl and test_list_page.tmpl. In these files there are template variables used to extract the tests directory, tests names, log files, stages, status stages, overall file status,etc. from the regression script.

The summary.tmpl creates the main HTML page which contains information about the number of tests, number of tests that pass and fail, the percentage of tests that pass , etc. . This page also has a table and the header of it contains the hdl directory, test directories, all the names of the stages and other information about the tests from the test directories. When you click one of these directories another HTML page opens which is created by the test_list_page.tmpl. This page contains all tests' names (maximum 50 tests per page) , all the names of the stages, overall file status.

The results are added in a directory created in /test/report/ with the name results_<date>_<time>.

E.g.: results_2008.09.16_11_18_23

When the regression script has finished running there is a message in the command line that shows the user the overall status of the tests ('passed' or 'failed'). In order to get the 'passed' message all the tests have to pass.


Read external arguments
Read hdl directory
Run tests
Read compiler log file
Extract test directories
(HTML templates)
Extract tests
(HTML templates)
Extract log file
(HTML templates)
Publish the results
(HTML pages)