SpecC Environment Setup

Andreas Gerstlauer

1. Access and Setup

Below are instructions on how to access and setup your SpecC environment on the department's instructional servers:

(a) The SpecC tool set is installed only on the Linux machines in the ECE LRC. Information about how to access the LRC machines is available at:
http://www.ece.utexas.edu/it/remote-linux
You can work on any of the regular 64-bit Linux servers.

(b) Once you have logged into one of the ECE-LRC linux machines, you need to load the SpecC module via:
% module load sce

(c) Once the module is loaded you are ready to run the SpecC reference compiler 'scc' at your command line. You can get help on the command line options of the SpecC compiler and any of the 'sir_xxx' tools by using 'man <*command*>'. The installation directory of the SpecC tool set can be accessed using the environment variable '$SPECC'.

2. HelloWorld Example

The below procedure walks you through a simple “Hello World” SpecC example:

(a) Make sure that your SpecC environment is setup so that the SpecC reference compiler 'scc' can be run at command line and the environment variable 'SPECC' is set to the installation directory.

(b) First create a work directory for the SpecC files.
(~) % mkdir specc_work
Change the current working directory to the newly created directory.
(~) % cd specc_work/
Copy the SpecC source file for the HelloWorld program from the examples directory into the newly created directory
(~/specc_work) % cp $SPECC/examples/simple/HelloWorld.sc .

(c) Compile the SpecC specification using the SpecC reference compiler.
(~/specc_work) % man scc
(~/specc_work) % scc HelloWorld -sc2out –vv -ww
If the sources do not have any errors, ‘scc’ generates a binary that has the same name as the design name along with intermediate SpecC and C++-source files:
(~/specc_work) % ls
HelloWorld HelloWorld.cc HelloWorld.h HelloWorld.o HelloWorld.sc HelloWorld.si

(d) Run the “HelloWorld” example using the newly created binary to see its output:
(~/specc_work) % ./HelloWorld
Hello World!

(e) Another possible flow is by creating intermediate SpecC Internal Representation (SIR) files instead of compiling the entire design in one go. In this flow, the ‘.sir’ files for all the modules/files in the design are pre-compiled and then the final binary is created in a separate step:
(~/specc_work) % scc HelloWorld -sc2sir –vv
*(~/specc_work) % ls *.sir*
HelloWorld.sir
(~/specc_work) % scc HelloWorld -sir2out -vv
(~/specc_work) % ./HelloWorld
Hello World!

(f) SpecC command line tools (‘sir_xxx’) can then be used to manipulate ‘.sir’ files:
(~/specc_work) % man sir_list
(~/specc_work) % sir_list –t HelloWorld.sir
behavior Main
(~/specc_work) % man sir_tree
(~/specc_work) % sir_tree –bltc HelloWorld.sir
B i l behavior Main