

SpecC Environment Setup

Mahesh Prabhu

1. Access and Setup

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

- (a) The SpecC tool set is installed only on the Linux machines in the ECE LRC. All the Linux machines in LRC available for access are listed at <http://www.ece.utexas.edu/it/linux/#bit-linux-machines>. You can work on any one of the 32-bit machines. The machines themselves are physically located in ENS507.
- (b) The Linux machines can be remotely accessed from the LRC windows machines using "putty" or "X-win32" which are installed on all the windows machines. You can open gui items only through "X-win32".
- (c) The SpecC tool set is NOT available for installation on your computer. You would have to access the ECE LRC Linux machines remotely from your machine. See remote access instructions at <http://www.ece.utexas.edu/it/linux/#logging-in-remotely>.

Once you have logged into one of the ECE LRC linux machines, you need to source "setup.csh" or "setup.sh" located at
/home/projects/courses/fall_08/ee382v-17295/sce-20080601/bin/
depending on whether your shell is a c-shell or bash-shell. The default shell at the LRC is the c-shell. You can identify the shell type by using the 'SHELL' environment variable:

```
% echo $SHELL  
/bin/tcsh
```

Then we can source the setup file as follows:

```
% source /home/projects/courses/fall_08/ee382v-17295/sce-20080601/bin/setup.csh
```

- (d) Once the setup file is sourced 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 by using 'man scc'. The installation directory of the SpecC tool set can be accessed using the environment variable 'SPEC_C'.

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. The example that we are going to use is a simple hello world program in SpecC.

- (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) % scc HelloWorld -sc2out -vv
```

The “-vv” option increases the verbosity level so that you can observe what the compiler is doing. If the SpecC file does not have any errors then scc would generate a binary file which has the same name as the design name along with the c-source files and the SpecC intermediate file:

```
(~/specc_work) % ls
```

```
HelloWorld HelloWorld.cc HelloWorld.h HelloWorld.o HelloWorld.sc
```

```
HelloWorld.si
```

- (d) Run the HelloWorld example using the newly created binary and you should get the below output from the testbench:

```
(~/specc_work) % ./HelloWorld
```

```
Hello World!
```

- (e) Another possible flow is by creating intermediate ‘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 is created and then the final binary is created in a separate step. For the HelloWorld example we can do the following:

```
(~/specc_work) % scc HelloWorld -sc2sir -vv
```

```
(~/specc_work) % ls *.sir
```

```
HelloWorld.sir
```

```
(~/specc_work) % scc HelloWorld -sir2out -vv
```

```
(~/specc_work) % ./HelloWorld
```

```
Hello World!
```