SystemC Environment Setup

Mahesh Prabhu

1. Access and Setup

Below are the instructions for accessing and setting up your environment for running SystemC on the Linux Machines in the department:

(a) The SystemC 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 do NOT need X server for running SystemC, so you can work remotely by using clients like "putty".

(b) You can also install SystemC locally on your machines. See http://www.systemc.org/downloads/standards/
for the latest SystemC release.

(c) Once you have logged in, set the 'SYSTEMC' environment variable to the directory of the SystemC installation located at:
/usr/local/packages/systemc-2.2.0
Depending on your shell (default in LRC is the C shell *csh* but you can find out what you are running through the 'SHELL' environment variable, i.e. by typing *echo $SHELL*), setting the 'SYSTEMC' variable is done as follows for [t]csh:
% setenv SYSTEMC /usr/local/packages/systemc-2.2.0
or, for [ba]sh:
% export SYSTEMC=/usr/local/packages/systemc-2.2.0

(d) Once the environment variable is set, you can access the SystemC installation by referring to the 'SYSTEMC' variable. The standard GNU C++ compiler 'g++' is then used to compile SystemC code and link it against the SystemC libraries. Note that if your are compiling on a 64-bit LRC machine, you will have to supply the '-m32' flag (to match the LRC SystemC installation, which is 32-bit):
% g++ -m32 -I$SYSTEMC/include –L$SYSTEMC/lib-linux –lsystemc –lm <source>

(e) It is recommended that you create a 'Makefile' for compiling your SystemC sources first to object (.o) files and then linking everything together into a final simulation executable. An example of a corresponding 'Makefile' that you can use as a starting point is available at:
/home/projects/gerstl/systemc-2.2.0/examples/Makefile.defs
Adjust this template to your needs by at least setting the 'MODULE' macro to the name of your main (top-level) source file (and potentially extending the 'OBJS' macro to list all the files that are part of your design).

2. Simple FIFO Example

The below procedure walks you through a simple FIFO SystemC example:

(a) Make sure that your SystemC environment is setup so that the environment variable 'SYSTEMC' is set to the installation directory. The example that we are going to use is the simple FIFO producer/consumer example that is part of the standard SystemC installation.

(b) First create a work directory for the SystemC files.

(~) % mkdir sysc_work

Change the current working directory to the newly created directory.

(~) % cd sysc_work/

Copy the SystemC source files for the simple_fifo program from the examples directory into the newly created directory

(~/sysc_work) % cp /home/projects/gerstl/systemc-2.2.0/examples/simple_fifo/ .*
(~/sysc_work)% ls
Makefile simple_fifo.cpp

(c) You can use the provided 'Makefile' to compile the example:

(~/sysc_work)% make
(~/sysc_work)% ls
Makefile simple_fifo simple_fifo.cpp simple_fifo.o*

(d) Run the generated binary to simulate the example design:

(~/sysc_work)% ./simple_fifo