Noxim
the NoC Simulator

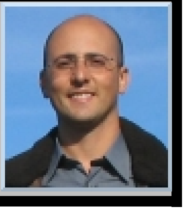
User Guide

http://www.noxim.org/

Maurizio Palesi, PhD

Email: mpalesi@diit.unict.it

Home: http://www.diit.unict.it/users/mpalesi/



Davide Patti, PhD

Email: dpatti@diit.unict.it

Home: http://www.diit.unict.it/users/dpatti/

Fabrizio Fazzino

Email: fabrizio@fazzino.it

Home: http://www.fazzino.it/

Introduction

Welcome to Noxim, the Network-on-Chip Simulator developed at the University of Catania (Italy) by the team of Computer Architecture shown on the cover of this manual.

The Noxim simulator is developed using SystemC, a system description language based on C++, and it can be downloaded from SourceForge under GPL license terms.

Noxim has a command line interface for defining several parameters of a NoC. In particular the user can customize the network size, buffer size, packet size distribution, routing algorithm, selection strategy, packet injection rate, traffic time distribution, traffic pattern, hot-spot traffic distribution.

The simulator allows NoC evaluation in terms of throughput, delay and power consumption. This information is delivered to the user both in terms of average and per-communication results.

In detail, the user is allowed to collect different evaluation metrics including the total number of received packets/flits, global average throughput, max/min global delay, total energy consumption, per-communications delay/throughput/energy etc.

The Noxim simulator is shipped along with Noxim Explorer, a tool useful during the design space exploration phase. Infact, Noxim Explorer executes many simulations using Noxim in order to explore the design space, and modifying the configuration parameters for each simulation. Noxim Explorer will create new configuration parameters for you, or complete the exploration according to the information read from a script (known as exploration script or space file).

Installation

This chapter will show you how to install Noxim on your computer.

Supported Platforms

Noxim is written using the C++ language and the SystemC library, so it is easily portable to any platform for which SystemC is supported: please refer to their (SystemC) documentation to know which they are. Just to let you know, we usually work under GNU/Linux (mainly Ubuntu) but SystemC is known to run under other platforms including Apple Mac OS X and Sun Solaris with GCC and Microsoft Windows with Visual C++ (but Cygwin with GCC is also known to work).

This document will then detail the steps required to build Noxim from the sources, including the only prerequisite, i.e. the SystemC installation.

Prerequisite: SystemC installation

- To compile SystemC you will obviously need a C++ compiler; if you still don't have it, on Debian/Ubuntu platforms you may install all the required tools with the following command:

```
sudo apt-get install build-essential
```

- Download SystemC (currently at version 2.2.0) from **http://www.systemc.org/downloads/standards/** (a free registration is required).
- Unpack it; please note that some versions have a wrong file extension. For instance you may have to use the following commands to

untar it:

```
    mv systemc-2.2.0.tgz systemc-2.2.0.tar
    tar xvf systemc-2.2.0.tar
```

- Enter the newly created directory and refer to the file INSTALL which details all the steps required for building. Basically they are:

```
    mkdir objdir
    cd objdir
    export CXX=g++
    ../configure
    make
    make install
    cd ..
    rm -rf objdir
```

- With modern versions of GCC you may find the "make" command above to fail with the following error message:

```
     ../../../../src/sysc/utils/sc_utils_ids.cpp: In function ‘int sc_core::initialize()’:
     ../../../../src/sysc/utils/sc_utils_ids.cpp:110: error: ‘getenv’ is not a member of ‘std’
     ../../../../src/sysc/utils/sc_utils_ids.cpp:111: error: ‘strcmp’ was not declared in this scope
    make[3]: *** [sc_utils_ids.o] Error 1
    make[3]: Leaving directory `/opt/systemc-2.2.0/objdir/src/sysc/utils'
    make[2]: *** [all-recursive] Error 1
    make[2]: Leaving directory `/opt/systemc-2.2.0/objdir/src/sysc'
    make[1]: *** [all-recursive] Error 1
    make[1]: Leaving directory `/opt/systemc-2.2.0/objdir/src'
    make: *** [all-recursive] Error 1
```

If this is your case (for instance it happens while compiling SystemC 2.2.0 with GCC 4.4), then please note that this is not a bug of the compiler but a bug in the SystemC sources, because they have forgot a couple of include clauses.

To fix it, add the following includes at the top of file **../src/sysc/utils/sc_utils_ids.cpp** :

```
    #include <cstdlib>
    #include <cstring>
```

You may even modify that file without using any text editor, just use this shell command (yes, Fabrizio still loves UNIX shell!!!):

```
sed -i '1 i #include <cstdlib>\n#include <cstring>' ../src/sysc/utils/sc_utils_ids.cpp
```

Then restart from the "make" step in the list above.

Once you have installed SystemC correctly, you may then jump to the next step.

Build SystemC

If SystemC is installed correctly, then you just have to compile Noxim.

1) Extract the source files and go to the "bin" directory.

2) In that directory edit the file Makefile.defs (NOT Makefile.deps!) to modify the "SYSTEMC" environment variable according to your SystemC installation path.

3) Just run "make". You may ignore warning messages (if any), so if you don't get any error you are ready to run Noxim for the first time using the command:

```
    ./noxim
```

If everything works fine, it is now safe for you to copy or move this executable elsewhere; if you are a maniac of cleaning please note that "make clean" will also delete the executable... so move it before cleaning!

That's all, folks!

User Manual

This short guide will explain some of the most commonly used parameters and options that can be passed to noxim on the command line.

The synopsis of the command is:

```
    noxim [ options ]
```

where

```
    options Command-line options.
```

The Noxim simulator basically launches a NoC simulation.
You can execute the command to access to the list of options:

```
    ./noxim -help
```

The output provided by this command should look like this:

```
             SystemC 2.2.0 --- Mar 11 2010 11:07:33
        Copyright (c) 1996-2006 by all Contributors
                    ALL RIGHTS RESERVED


         Noxim - the NoC Simulator
                (C) University of Catania


Usage: ./noxim [options]
```

```
where [options] is one or more of the following ones:
        -help           Show this help and exit
         -verbose N       Verbosity level (1=low, 2=medium, 3=high,
default off)
             -trace  FILENAME  Trace  signals  to  a  VCD  file  named
'FILENAME.vcd' (default off)
        -dimx N                 Set  the  mesh  X  dimension  to  the
specified integer value (default 4)
        -dimy N                 Set  the  mesh  Y  dimension  to  the
specified integer value (default 4)
        -buffer N  Set  the  buffer  depth  of  each  channel  of  the
router to the specified integer value [flits] (default 4)
        -size Nmin Nmax Set the minimum and maximum packet size to
the specified integer values [flits] (default min=2, max=10)
          -routing TYPE    Set the routing algorithm to TYPE where
TYPE is one of the following (default 0):
                 xy              XY routing algorithm
            westfirst       West-First routing algorithm
            northlast       North-Last routing algorithm
            negativefirst   Negative-First routing algorithm
            oddeven     Odd-Even routing algorithm
            dyad T      DyAD routing algorithm with threshold T
            fullyadaptive   Fully-Adaptive routing algorithm
            table FILENAME   Routing  Table  Based  routing  algorithm
with table in the specified file
       -sel TYPE   Set the selection strategy to TYPE where TYPE is
one of the following (default 0):
            random      Random selection strategy
            bufferlevel Buffer-Level Based selection strategy
            nop         Neighbors-on-Path selection strategy
       -pir R TYPE      Set   the   packet   injection   rate   to   the
specified   real   value   [0..1]   (default   0.01)   and   the   time
```

```
distribution of traffic to TYPE where TYPE is one of the following:
            poisson               Memory-less Poisson distribution
(default)
            burst R         Burst distribution with given real
burstness
            pareto on off r  Self-similar Pareto distribution with
given real parameters (alfa-on alfa-off r)
            custom R         Custom distribution with given real
probability of retransmission
       -traffic TYPE         Set the spatial distribution of traffic
to TYPE where TYPE is one of the following (default 0'):
            random     Random traffic distribution
            transpose1  Transpose matrix 1 traffic distribution
            transpose2  Transpose matrix 2 traffic distribution
            bitreversal Bit-reversal traffic distribution
            butterfly   Butterfly traffic distribution
            shuffle     Shuffle traffic distribution
            table FILENAME   Traffic    Table    Based    traffic
distribution with table in the specified file
          -hs ID P           Add node ID to hotspot nodes, with
percentage P (0..1) (Only for 'random' traffic)
       -warmup N        Start to collect statistics after N cycles
(default 1000)
       -seed N                Set the seed of the random generator
(default time())
       -detailed        Show detailed statistics
       -volume N        Stop the simulation when either the maximum
number of cycles has been reached or N flits have been delivered
       -sim N                Run for the specified simulation time
[cycles] (default 10000)

If you find this program useful please don't forget to mention in
```

```
your paper Maurizio Palesi <mpalesi@diit.unict.it>
If you find this program useless please feel free to complain with
Davide Patti <dpatti@diit.unict.it>
And if you want to send money please feel free to PayPal to
Fabrizio Fazzino <fabrizio@fazzino.it>
```

Now we'll take a closer look at each option.

-help

The -help option allow you to know the possible options accepted by Noxim (the same list that you can see above).

-verbose N

With the -verbose option you can monitor the verbosity level of the output generated by Noxim. There are four levels. By default verbosity output is off: in this case you'll get only the main statistics produced by Noxim (total received packets, total received flits, global average delay, global average throughput, throughput, max delay, total energy).

When the verbosity level is set to low, in addition to the output generated when verbosity is off, the configuration parameters are reported and you can see the work done by each element of the NoC system (i.e. processing elements and routers).

Currently the "medium" verbosity level has no difference with the previous one.

When the verbosity level is set to high, in addition to the output produced when the verbosity is low (medium), you can see a detailed information about flit for each activity performed by each NoC element.

-trace FILENAME

You use the -trace option to trace all the SystemC signals used in the NoC simulator (clock, reset, req_to_east, etc.) to a VCD file named ’FILENAME.vcd’. The default value is off.

A tool commonly used to visualize VCD trace files is GTKWave: if you are using Linux you should find a package with this name (lowercase) for your distro.

-dimx N / -dimy N

The options -dimx and -dimy are used to set topology information, i.e. the width and height of the matrix representing the mesh of the NoC.

-buffer N

The option -buffer is used to define the buffer size of each channel of the router. This size is expressed in flits. Please read the above instructions for more details.

-routing TYPE

The -routing option enable you to specify one of the routing algorithms listed above.

-sel TYPE

With the -sel option you can choose the selection function. The default is random. The selection function is used to choose the output port to which every flit has to be sent. Rather than using a pseudo-random algorithm, you may also opt for a buffer-level strategy. Here the favourites output ports are those which are connected to biggest number of free channels. A free channel is one with the greater number of free slots in the destination FIFO buffer.

-pir R TYPE

With the -pir option you can set the Packet Injection Rate (PIR) to the specified real value.

-traffic TYPE

The -traffic option is used to manage the time distribution of the traffic. In particular you can decide if the traffic generation is patterned as poisson, burst, pseudo-pareto or custom distribution. For random traffic you can define some nodes as hot spot nodes. This is accomplished with the following -hs option.

-hs ID P

The -hs option is used to specify the Hot-Spot nodes. Along with the node identificator you must specify the hot spot percentage.

-warmup N

With the -warmup option you can set the start time after which the simulator starts to collect statistics.

-seed N

The -seed option is used to set the seed of the random number generator used by the simulator. By default it will use the standard time() function.

-detailed

The -detailed option provide per-communications statistics. In particular, for each destination node are collected the aggregated average delay and throughput. Then the statistics for each communication having that node as a destination node are reported using a table.

```
-volume N
```

The -volume option is used to stop the simulation either when the maximum number of cycles has been reached or when N flits have been delivered.

```
-sim N
```

The -sim option is used to specify the number of clock cycles that have to be simulated. The default value is 10000 (ten thousands) cycles.

Examples

If you want to simulate a 8x8 NoC, you must execute this command:

```
        ./noxim -dimx 8 -dimy 8
```

To make a more accurate simulation you must specify the simulation time (in terms of clock cycles) and the warm-up session. This is accomplished using the -sim and -warmup options.

Then to carry out a 8x8 NoC with a simulation time of 40000 cycles and warm-up session of 5000 cycles you have to run the following command:

```
        ./noxim -sim 40000 -warmup 5000 -dimx 8 -dimy 8
```

If you want change the FIFO buffer size you can set the -buffer option. For example, if you wish a buffer size of two flits you must run this command:

```
./noxim -sim 40000 -warmup 5000 -dimx 8 -dimy 8 -buffer 2
```

These were the basic Noxim capabilities.