Fast Functional SystemC Simulator using Static Scheduling

Richard Buchmann, Alain Greiner

Universite Pierre et Marie Curie, ASIM/LIP6
4 place Jussieu, 75252 Paris
France
richard.buchmann@lip6.fr, alain.greiner@lip6.fr

March 20, 2006

Abstract

We present a new SystemC simulator more than 10x faster than Standard SystemC package. SystemCASS use signal dependancy analysis and static scheduling.

Method

We follow few steps :

1. **Port dependancies** analysis for each hardware component instance
2. Netlist elaboration
3. **Static scheduling** generation
4. **Cycle accurate** simulation

Two ways to get port dependancy informations are available : using automatic compiler/analyser at compile time; or either writing dependancy rules by hand.

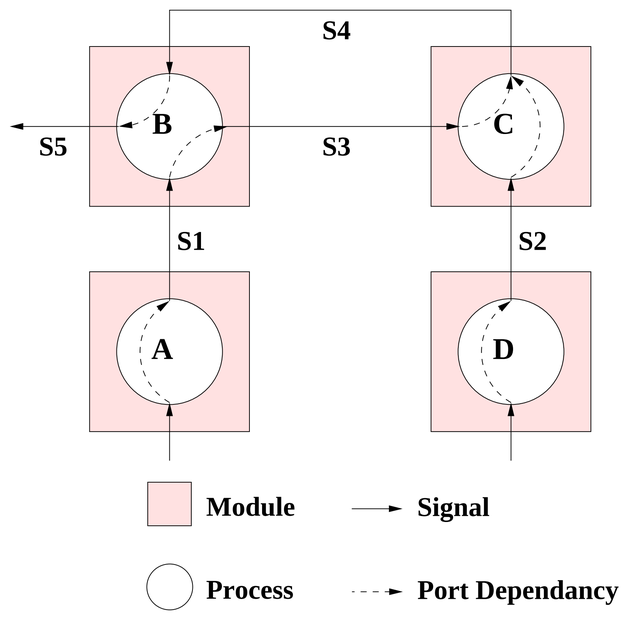

Figure 1: Mouchard's Toplevel Example

The automatic analyser parses the source code; build the data support for each variable; and writes port dependancy rules to an external source file.

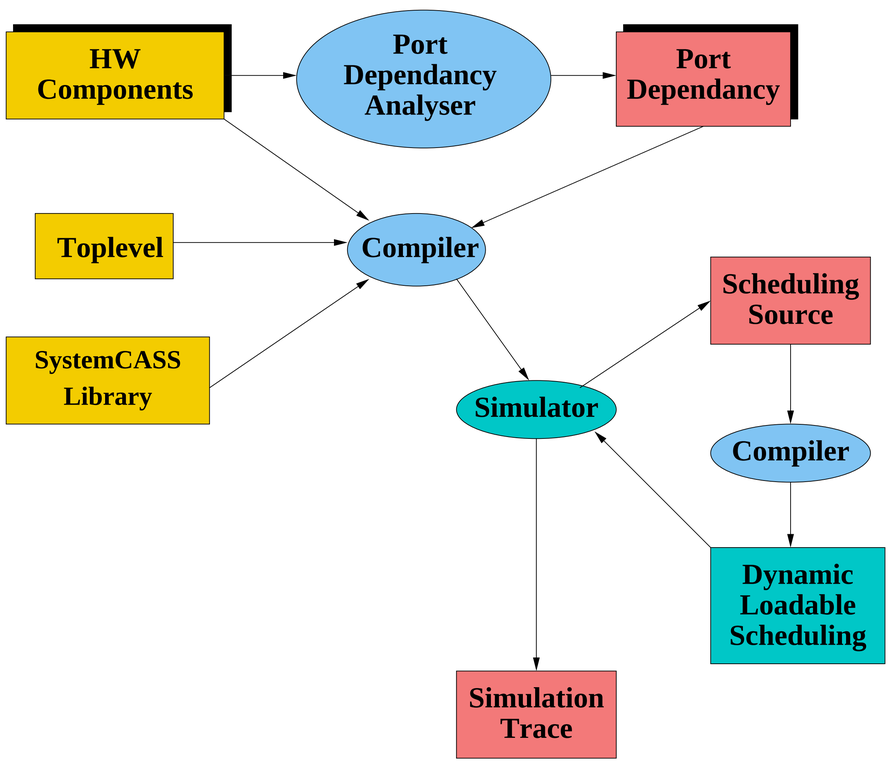

Figure 2: SystemCASS Workflow

This table shows some differences between standard SystemC and our implementation.

Simulator Library	SystemC 2.0.1	SystemCASS
Simulator Type	Discrete Event	Cycle Based
Language Subset	All	Core Subset
Scheduling	Dynamic	Static
Communication Interface	Costly	Lightweight
Sensitivity List	From user	No need
Port Dependancy	No need	Automatic

Figure 3: Simulator Overview

Platforms

NCPU is a real gigabit ethernet application using multiprocessor architecture.
This platform use **SoCLIB** IP Cores. (http://soclib.lip6.fr)

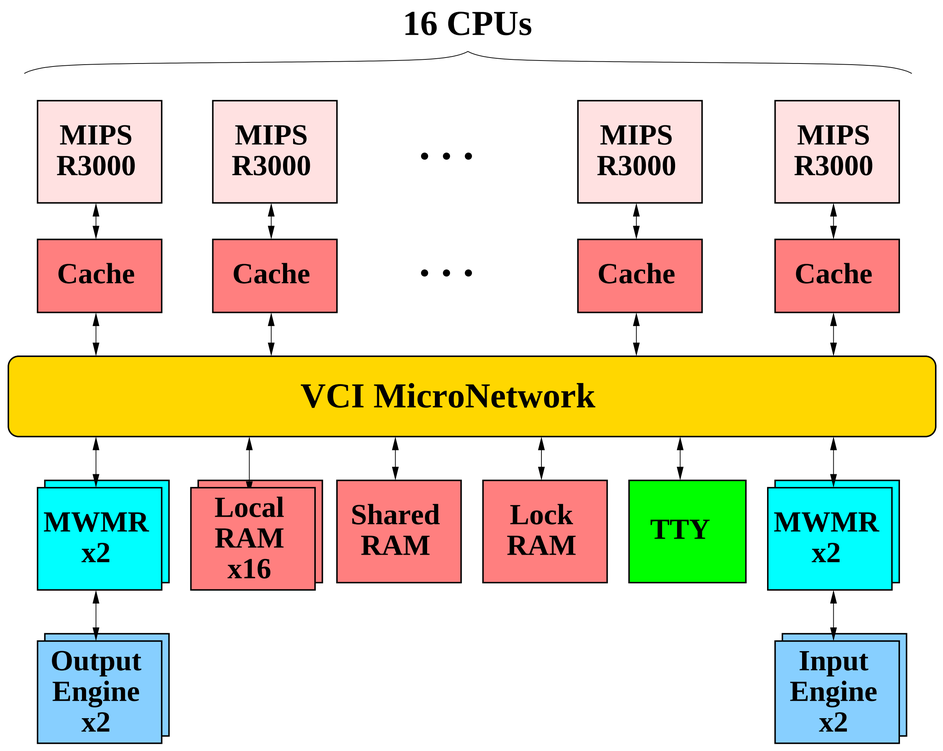

Figure 4: NCPU : Gigabit Ethernet Application

Results

We execute the same models using two different engines.

Simulator Library	SystemC 2.0.1	SystemCASS
NCPU Platform (16x CPU)	1 725 cycles/s	23 000 cycles/s
NCPU Platform (24x CPU)	1 042 cycles/s	16 949 cycles/s

Figure 5: Performance overview

Conclusions

SystemCASS :

- is more than **10x faster** than SystemC standard package
- gets data dependancy automatically.