Fraunhofer Institut
Integrierte Schaltunge

RTL Design-Flow with SystemC in an Industrial Design Project

Modeling a GPS Receiver Using SystemC

Bernhard Niemann
Martin Speitel
Fraunhofer Institute for Integrated Circuits

www.iis.fhg.de
www.iis.fhg.de/kursbuch/kurse/systemc.html

Introduction

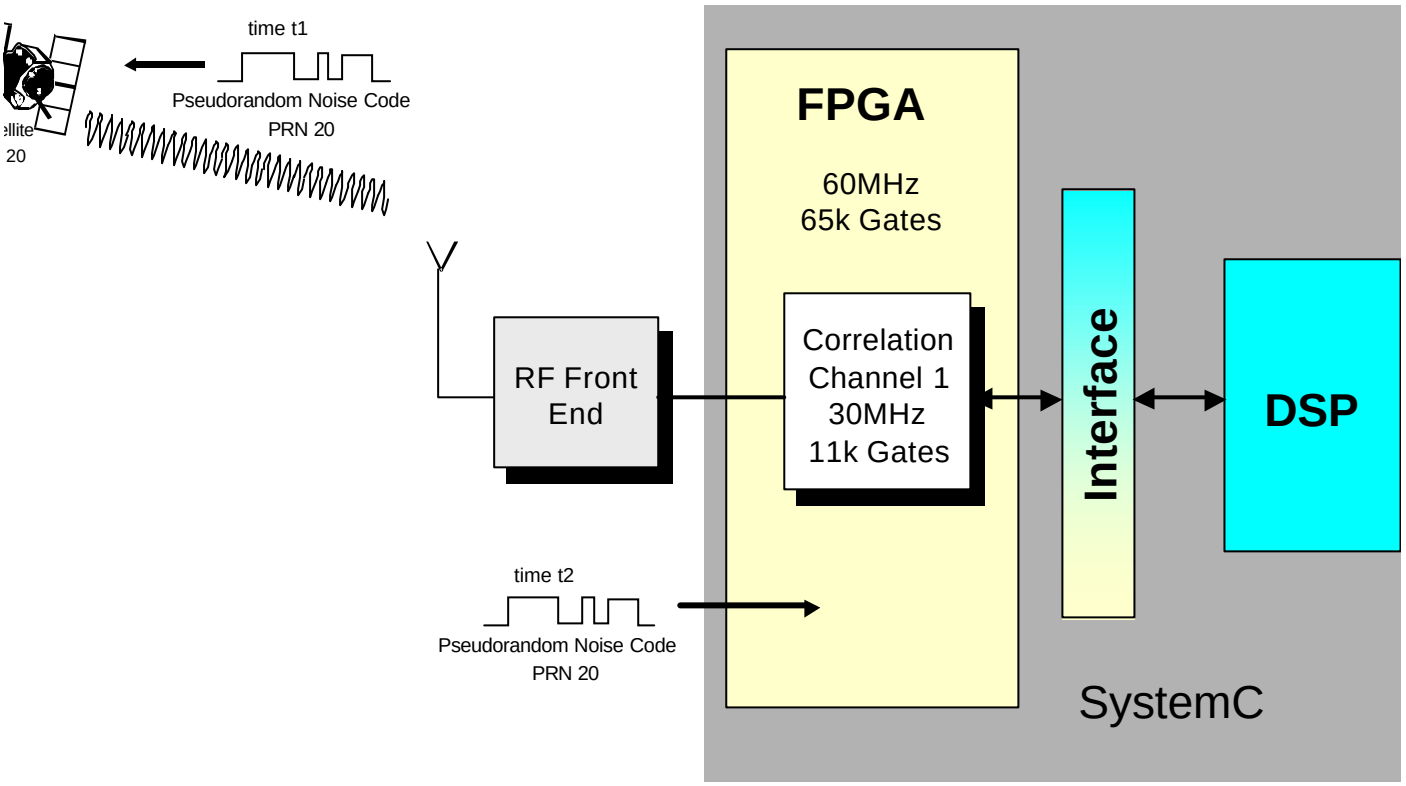
- Introduction
- System Overview
- SystemC Design Flow
  - Overview
  - Simulation
  - Synthesis (beta evaluation)
- Experience
- Conclusion

Last Year's Conclusions

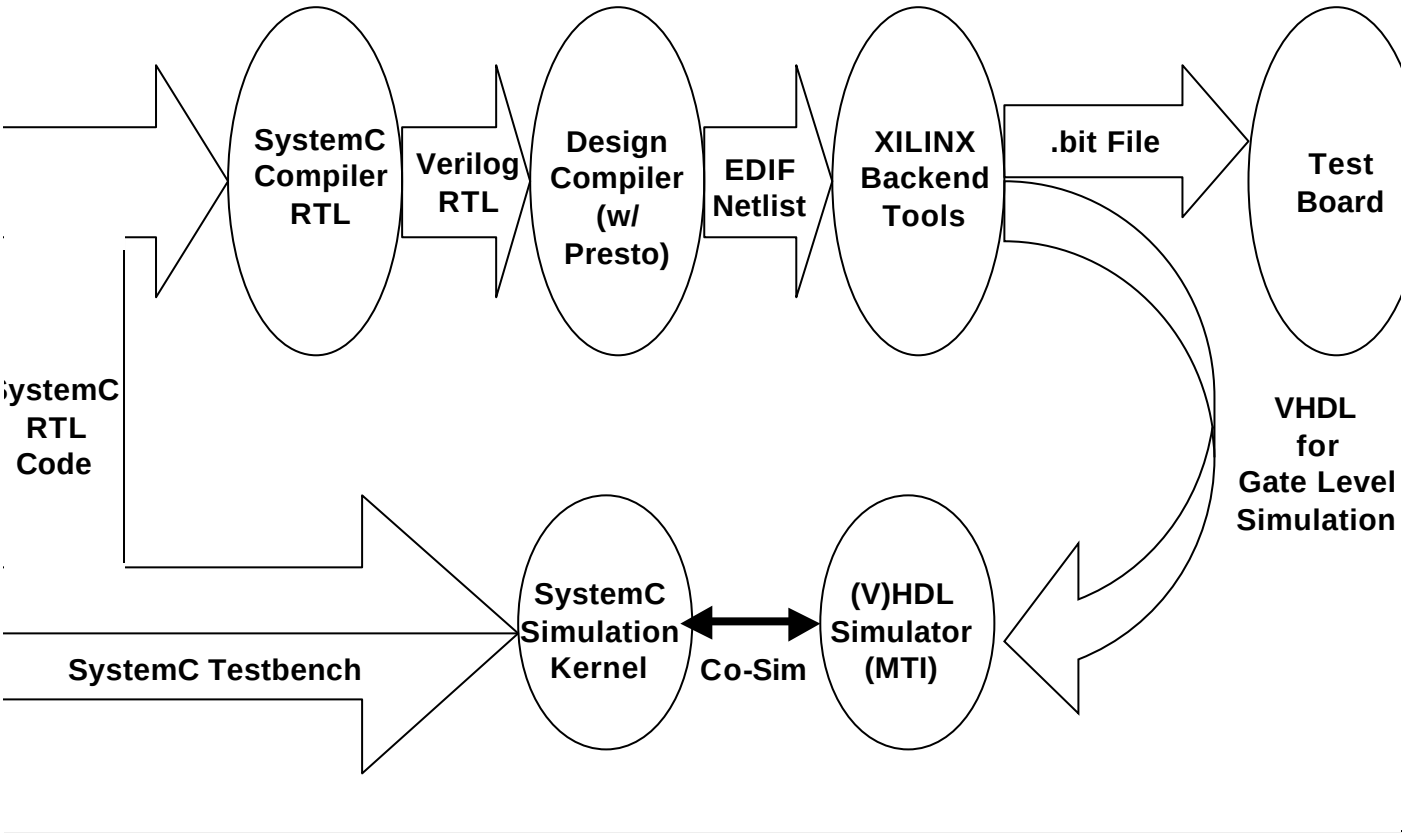
Fraunhofer Institut Integrierte Schaltungen

Pros	• Easy to bring existing HW and SW modules into SystemC • HW/SW Codesign without any commercial tools • Easy debugging • Easy design exploration
Cons	• Until now no converters (V)HDL <-> SystemC • Some problems with SystemC data types
Project will go on	• SystemC Compiler will be available

System Overview - Architecture



Design Flow - Overview



Simulation - Overview

Simulation has been used to

- test/develop DSP software (SystemC built-in simulation kernel)
- verify SystemC RTL model against VHDL RTL model (Co-Simulation SystemC/VHDL)
- verify synthesis results (convert mapped FPGA design back to VHDL and use SystemC VHDL Co-Simulation)

SystemC VHDL Co-Simulation has been carried out using

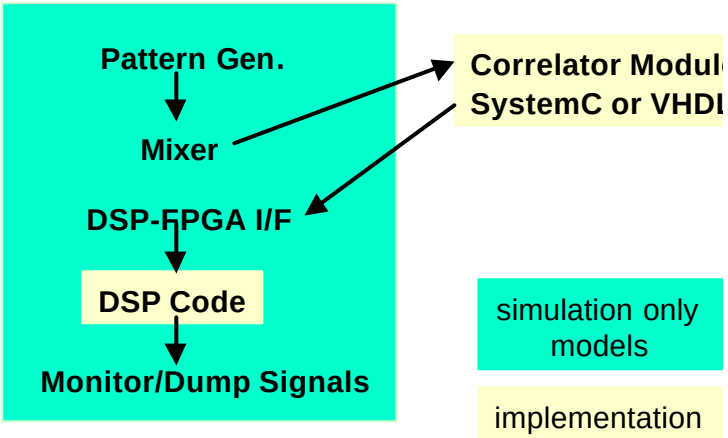
- SystemC test bench as the master
- Modelsim VHDL simulator as the slave

Simulation - Test Environment

Fraunhofer Institut Integrierte Schaltunge

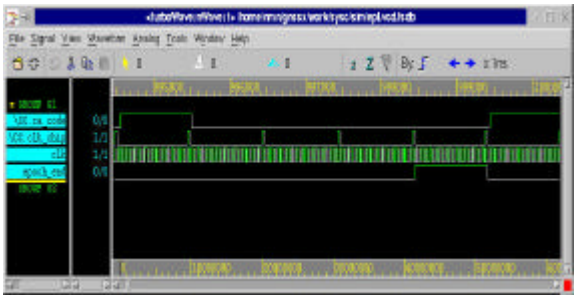
Objectives for test bench

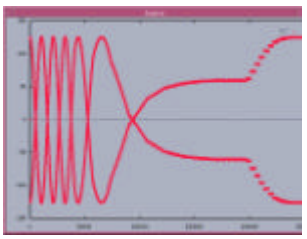
- simulation speed
- implementation effort
- integration in system development flow



Results of simulation

- waveform dump
- several ASCII files suitable for plotting with gnuplot
- simulation log on stdout





Simulation - Results

Fraunhofer Institut Integrierte Schaltungen

Simulation was carried out on a SUN Ultra 80

- 2 CPUs with 450 Mhz
- 3.0 GB RAM, 6.4 GB Virtual

Simulation time was measured using

- `/usr/bin/time` command
- a simulation time of approx. 60 C/A cycles

Simulation Engine	VHDL RTL	SystemC RTL	SystemC Funct.	Time
SystemC Kernel	N/A	Correlator	Mixer, DSP FPGA I/F, (DSP Code)	03:10 user 00:02 sys
SystemC Kernel, Modelsim VHDL Simulation	Correlator	N/A	Mixer, DSP FPGA I/F, (DSP Code)	03:50 user 02:13 sys
Modelsim VHDL Simulation	Correlator, Mixer, DSP FPGA I/F	N/A	N/A	21:02 user 00:00 sys

Synthesis (beta) - Overview

The SystemC Compiler RTL flow

- setup
- translate SystemC to Verilog
- read/analyze the generated Verilog and write .db file
- normal compile flow using dc_shell

C++ pre-processor used for reading SystemC files

- we have used g++
- also possible: SUN CC

```
/* setup */
hdlin_enable_presto = "true"
hdlin_unsigned_integers = "false"

/* translate SystemC to Verilog */
compile_systemc -cpp cpp -cpp_options "-C \
 -I../src/common" -rtl -rtl_format verilog \
 ../src/core_correlator/correlator.cc

/*analyze SystemC Compiler generated Verilog*/
analyze -f verilog {"rtl_work/correlator.v"}

/* write .db file */
elaborate correlator
write -format db -hier -output "./db/" +
   "correlator_sysc_elab.db"
quit
```

Presto is the new (V)HDL Compiler of Synopsys. It uses new (advanced) mapping algorithms and supports more language constructs. The reason it has to be used with the SystemC synthesis flow is, that SystemC Compiler writes out Verilog 2000. To be able to read the generated Verilog 2000 file into design compiler, you need to enable Presto.

For the beta version of SystemC Compiler the recommended flow is to translate SystemC to Verilog and generate the .db file from the translated Verilog source code. Support for directly writing a .db file from SystemC should be included in the production release of SystemC Compiler.

Synthesis (beta) - Results

Synthesis results with SystemC differ from VHDL results, because

- Presto is used to analyze Verilog files generated by SystemC Compiler
- no Presto used for VHDL, because mapping to FPGA was not possible
- Presto mapping to GTECH differs from normal VHDL Compiler mapping

For some example blocks, VHDL was rewritten to match mapping c Presto (e.g. operator sharing)

➢No differences in the behavior could be detected

Input to Synthesis	Remarks	Gate Count	max. Frequency
SystemC / Presto	SystemC code is identical to original VHDL code	8.5k	37MHz
VHDL (original) / no Presto	the original VHDL code of the correlator module	11k	50MHz
VHDL (modified) / no Presto	some parts modified to match SystemC (Presto) mapping results	10k	44MHz

Experience - Tools

Fraunhofer Institut Integrierte Schaltungen

HDL SystemC CoSim

- offers tight integration of SystemC in existing HDL flow
- no major problems encountered
- the `ifgen` command crashes for module names with more than 12 characters (beta version)

ystemC Compiler (beta)

- translation of SystemC to Verilog is straightforward
- Verilog output is useful for learning and improving SystemC modeling style
- SystemC Modeling Guide was already available for beta version, however with quite some typos

Conclusion

ystemC can be used or real-world designs

- there is tool support for the complete SystemC flow from system specification to synthesis
- Easy to do HW/SW co-design
- Easy design exploration

till some problems

- some problems with SystemC data types
- some tool bugs/limitations with tools (e.g. currently no FC2 support, will be included in later versions)
- some confusion with SystemC versions (v1.0, v1.1beta, v1.0.1, v1.2beta)

se for HW/SW co-esign

- training (language/methodology) necessary
- consider using SystemC for projects that
  - do not have too tough project plans
  - require HW/SW co-design and co-simulation