Hardware/Software Codesign

HW/SW Entwurfssprachen am Beispiel System-C

Reinhold Gschweicher - 0828055

Vienna University of Technology

November 4, 2012

SystemC

SystemC is an open C++ class library used for hardware system design and validation. The SystemC class libraries add hardware attributes to the C++ language. [5]

It was first defined by the Open SystemC Initiative (OSCI) in 2000

SystemC ist standardised by IEEE 1666-2011 standard [3]

available free at `http://www.accellera.org/`


accellera
SYSTEMS INITIATIVE


AMD


ARM


cadence


freescale


intel


magillem


Mentor Graphics


NXP


QUALCOMM

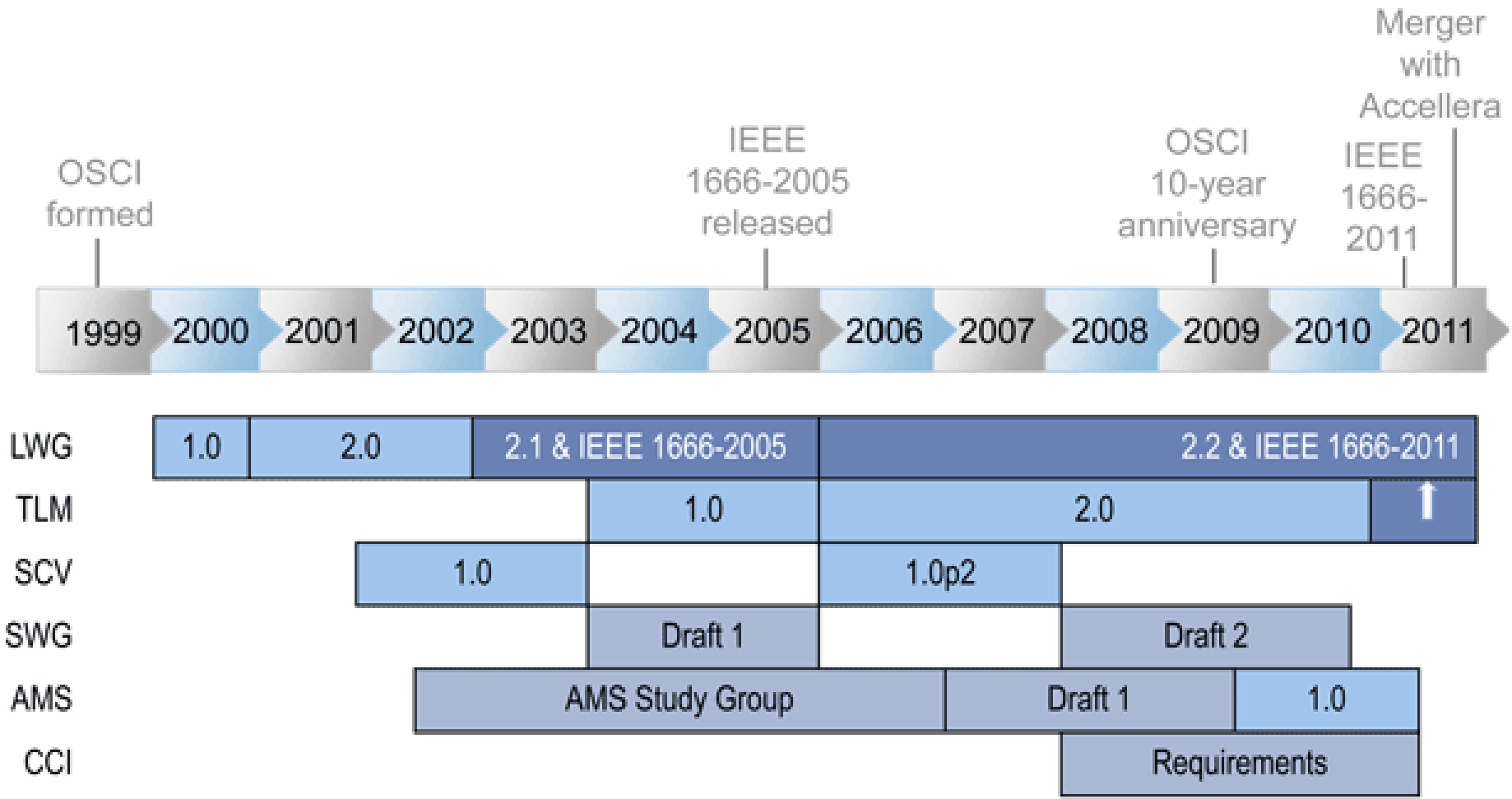

RENESAS
Renesas Mobile Corporation


SpringSoft


SYNOPSYS


TEXAS
INSTRUMENTS

History


OSCI formed
IEEE 1666-2005 released
OSCI 10-year anniversary
IEEE 1666-2011
Merger with Accellera
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
LWG
1.0
2.0
2.1 & IEEE 1666-2005
2.2 & IEEE 1666-2011
TLM
1.0
2.0
SCV
1.0
1.0p2
SWG
Draft 1
Draft 2
AMS
AMS Study Group
Draft 1
1.0
CCI
Requirements

History

- 1999 Open SystemC Initiative formed
- 2000 V1.0 released
- 2001 V2.0 released
- 2005 V2.1 released
- 2005 IEEE 1666-2005 Standard for SystemC approved
- 2007 V2.2 released
- 2011 IEEE 1666-2011 Standard for SystemC approved
- 2011 merger OSCI with accellera
- 2012 V2.3 released

Components

Features [1]

- Modules
  principle structural building blocks
- Methods and Threads
  - SC_METHOD
  - SC_THREAD
  - SC_CTHREAD
- Signal
- Ports
- Channels
- Interfaces
- Events
  simple synchronization between threads

Asynchronous NOT

```
SC_MODULE(a_not)
{
  // ports
  sc_in<bool> input;
  sc_out<bool> output;
  // process
  void do_not()
  {
    output.write(!input);
  }
  // constructor
  SC_CTOR(a_not)
  {
    SC_METHOD(do_not);
    sensitive << input << output;
  }
};
```

Synchronous NOT

```
SC_MODULE(sync_not)
{
  sc_in<bool> clock;
  sc_in<bool> input;
  sc_out<bool> output;

  SC_CTOR(sync_not)
  {
    SC_CTHREAD(do_not, clock.pos());
    reset_signal_is(reset, true);
  }
  void do_not()
  {
    if(reset)
    {
      output.write(false);
    }
    while(true)
    {
      wait(1);
      output.write(!input);
    }
  }
};
```

(Dis)Advantages [5]

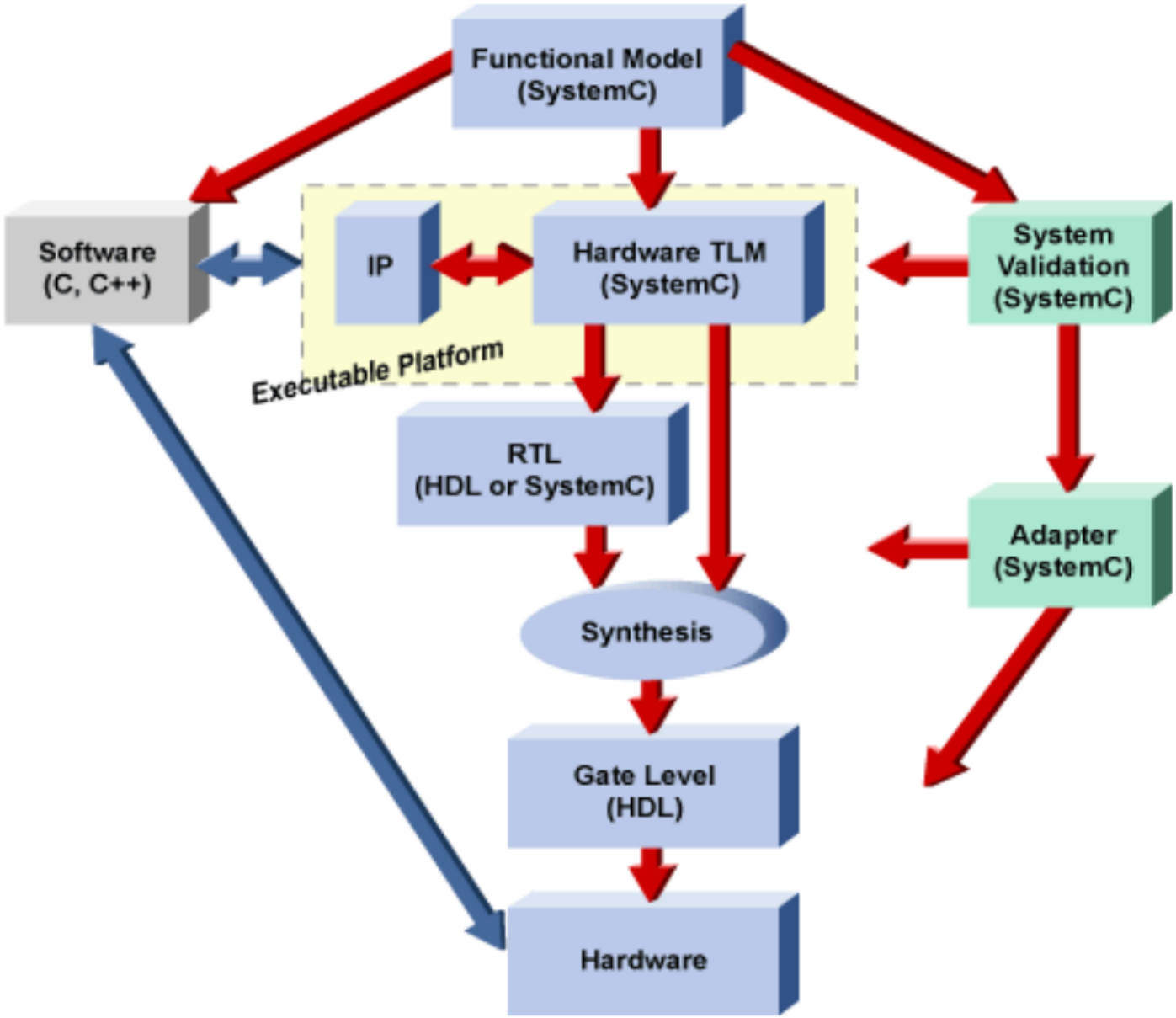
Advantages

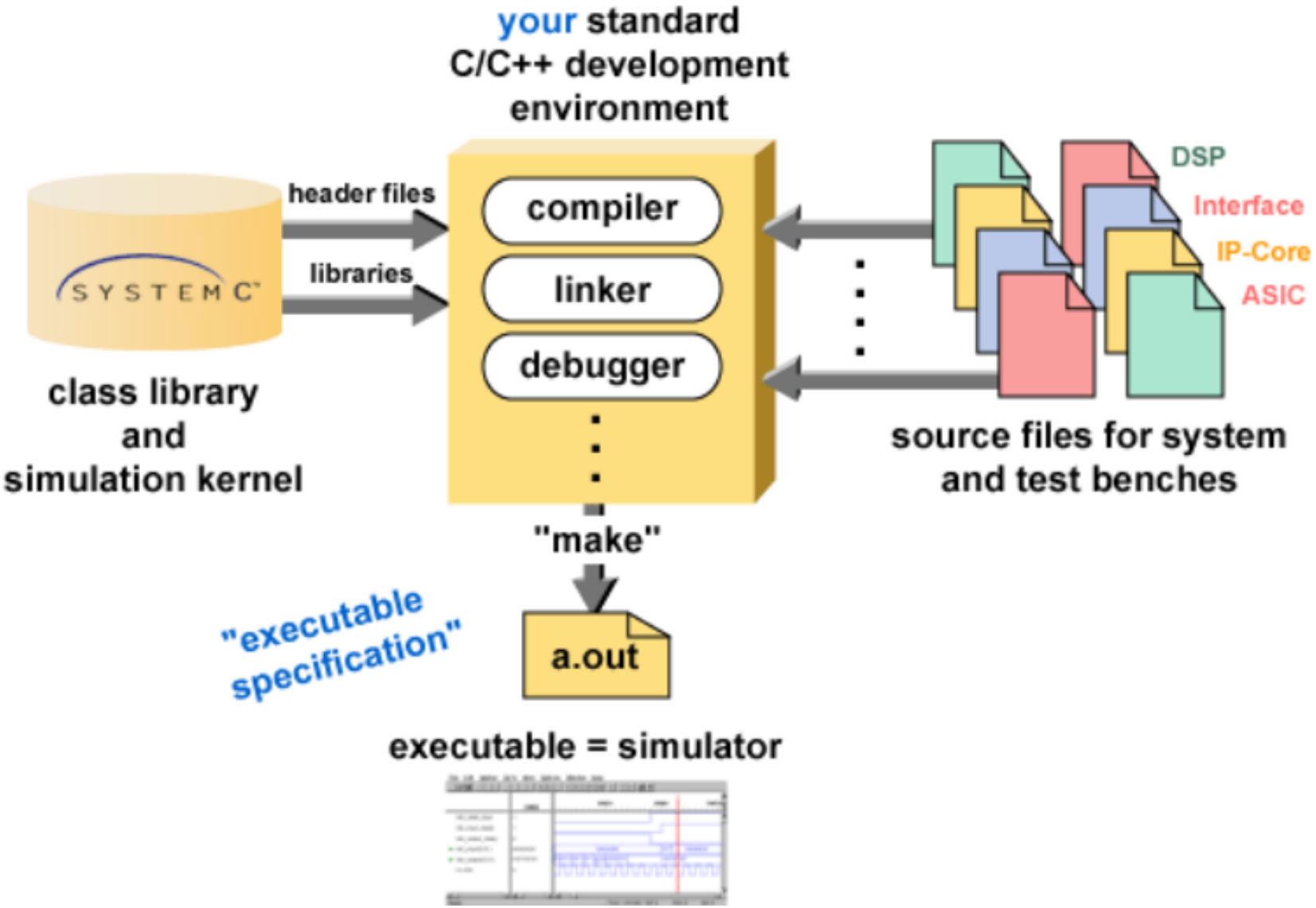
- open source
- describe a system at several abstractions levels
  - Functional Model
  - Hardware TLM
  - System Validation
  - RTL
- formal verification

Disadvantages

- requires to learn new language (little impact, it's based on C/C++)
- big overhead when used as HDL

Methodology





[4]

http://embedded.eecs.berkeley.edu/research/hsc/class/ee249/lectures/l10-SystemC.pdf

Existing Tools

- gcc for simulation
- Modelsim (SystemC Simulation)
- Synopsys CoCentric SystemC Compiler (SystemC to Verilog)
- Xilinx Vivado Design Suite (C/C++, SystemC, MATLAB/Simulink to VHDL/Verilog)

Automatc Partitioning

does it work?
not yet

LegUp [2]

- An Open Source High-Level Synthesis Tool for FPGA-Based Processor/Accelerator Systems
- Standard C program to Verilog
- Partitions into FPGA-based MIPS soft processor and custom hardware
- visit `http://legup.eecg.utoronto.ca/` for more information

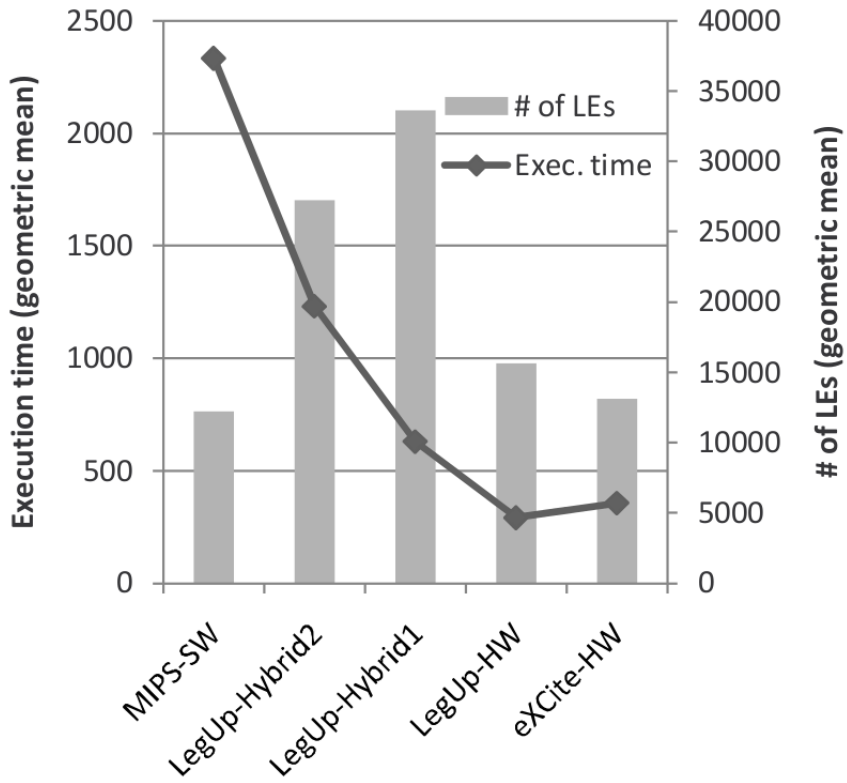


Fig. 9. Performance and area results (performance in μS).

References I


David C. Black, Jack Donovan, Bill Bunton, and Anna Keist.
SystemC: From the Ground Up, Second Edition.
Springer, 2nd edition, 2009.


Andrew Canis, Jongsok Choi, Mark Aldham, Victor Zhang, Ahmed Kammoona, Tomasz Czajkowski, Stephen D. Brown, , and Jason H. Anderson.
Legup: An open source high-level synthesis tool for fpga-based processor/accelerator systems.
2012.


IEEE.
Ieee standard for standard systemc language reference manual.
IEEE Std 1666-2011 (Revision of IEEE Std 1666-2005), pages 1 –638, 9 2012.

References II

John Moondanos.
Systemc tutorial.

I. Yarom and G. Glasser.
Systemc opportunities in chip design flow.
In *Electronics, Circuits and Systems, 2004. ICECS 2004. Proceedings of the 2004 11th IEEE International Conference on*, pages 507 – 510, dec. 2004.