HW/SW Entwurfssprachen am Beispiel System-C

Florian Zaruba, Thomas Weber

November 2, 2014

1 HW/SW Design Languages

tba

2 What is SystemC

tba

2.1 History

SystemC were introduced in 1999 as SystemC 1.0. At this time it provided some basic features, e.g., Signals, Simulation of a kernel and fixed point arithmetic. It was also possible to break down designs it into modules in order to reuse code.

SystemC 2.0 was started in the late 2000. It extends the previous version with events as primitive behavior triggers as well as channels, interfaces and ports. It was also more powerful in modeling at transaction level. Furthermore its complete library were new coded to upgrade it to an independent System Level Description Language (SLDL).

2.2 Benefits

tba

2.3 Drawbacks

tba

2.4 SysC vs. C

tba

2.5 SysC vs. VHDL

3 Automatic Partitioning

3.1 What is Partitioning?

Partitioning means the separation of hardware and software parts with focus on hardware/software co-design. Traditionally the hardware part of an embedded system is written in VHDL *(more present in Europe)* or Verilog *(more present in the USA)* while the software part is written in *assembly*, *C* or *C++*. The common design-flow (depicted in Figure 1) shows that partitioning splits the design in hardware and software. The software part is the shorter one of this two paths because it needs only an compiler to translate it to machine-code. Therefore, the hardware part is the much longer way because of the hardware design-flow which is necessary for every chip-design. If the hardware and software parts

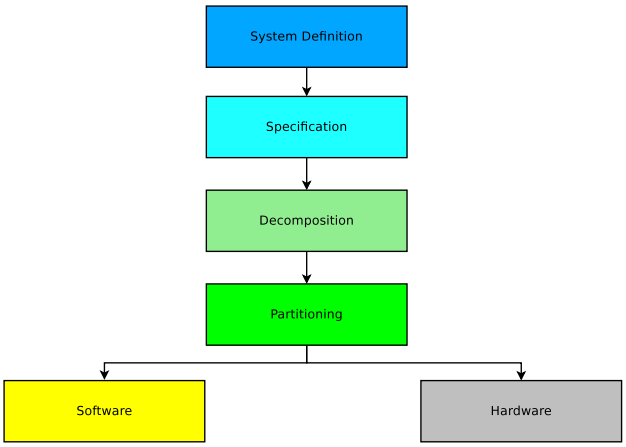

Figure 1: Design flow *(c.f. Hardware Modelling VO)*.

should be modified because of, e.g., performance reasons, these two parts must be modified or completely redesigned in order to fit the specifications. This leads to an much higher and probably expensive procedure for the hardware part, thus its design-flow is much longer and every change in the *VHDL/Verilog* code leads to a new compilation, technology-mapping and place and route.

4 Tools

4.1 LegUP

LegUP[1] is an open source high level synthesis tool for fpga based systems from the University of Toronto. It takes an standart C program as input and im-

[1] Its current version is 3.0

plements the hardware fitting part to hardware and the software fitting part to software. LegUP uses an MIPS soft core and uses a standard bus interface for communication between the hardware parts and the processor. **Automatic Partitioning? - Yes in a way.**

4.2 Bambu

Bambu[2] is a free framework for the high-level synthesis of complex applications.

4.3 Modelsim

Well known tool for simulation. It also supports SystemC simulation.

5 Creators

5.1 Accellera

Accellera Systems Initiative is a non-profit organization composed of an broad range of members which aims to create system-level standards for the worldwide electronics industry. One of the IEEE[3] standards ratified by accellera was IEEE 1666 SystemC.

6 Users

6.1 ARM

Figure 2: ARM Ltd.

ARM (Advanced RISC Machines) Ltd. has an active community[4] and there are many informations about tools and IP cores available. SystemC is used by many chip designers and they use the systemC-IP-cores from ARM Ltd. to simulate their designs.

6.2 AMD

6.3 Intel

[2]It can be downloaded at http://panda.dei.polimi.it
[3]This standards organization also ratified VHDL, Verilog and SystemVerilog
[4]http://community.arm.com