Hardware Description Languages SystemC and TLM overview

Sandro Penolazzi
Dept. of Electronic Systems, School of ICT, KTH
E-mail: sandrop@kth.se

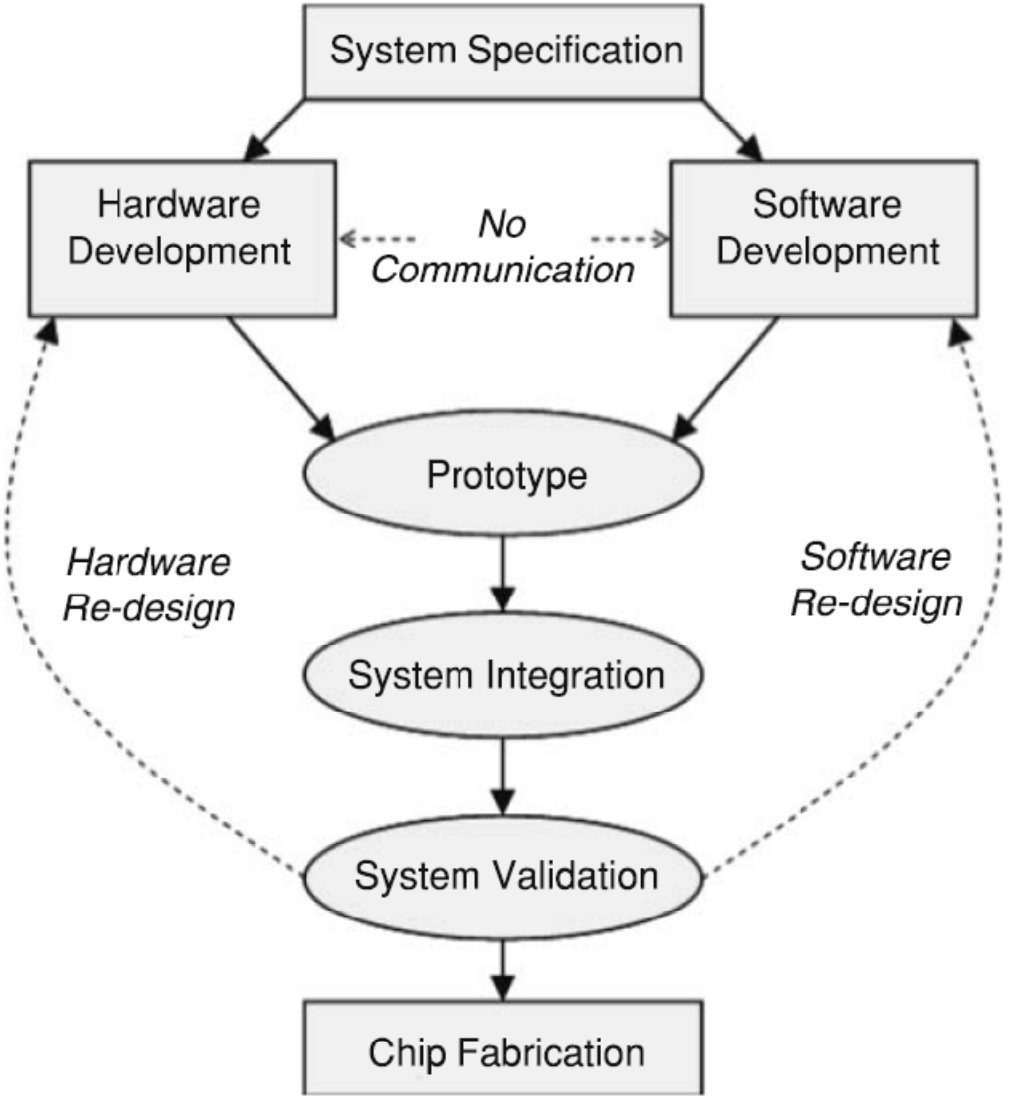
Outline

- Introduction to System-on-Chip concept
- Problems of classic approach to hardware design
- Solution: increasing abstraction level
  - SystemC
  - Transaction-Level Modeling
- Summary

The System-on-Chip concept

- Integration of individual components on a single chip to form an entire system
  - Possible thanks to great manufacturing / technology advances
  - Based on usage of fundamental building blocks, i.e. *Intellectual Properties* (IPs)
  - Oriented to hardware reutilization (templates)--> reduce design time

Classic design flow for SoC



Source: F. Ghenassia. *Transaction-Level Modeling with SystemC, 2005*

Classic design flow bottlenecks

Classic design flow becomes unsuitable because of 3 main reasons:

1. Explosive complexity
   - In hardware: more and more complex components
   - In software: more and more complex applications
2. Time-to-market pressure
   - Classic design flow takes too long, as one has to wait until a prototype is ready
3. Increasing cost
   - Larger workforce
   - Technological fabrication process becomes more costly
   - EDA tools become also more costly

Looking for new solutions

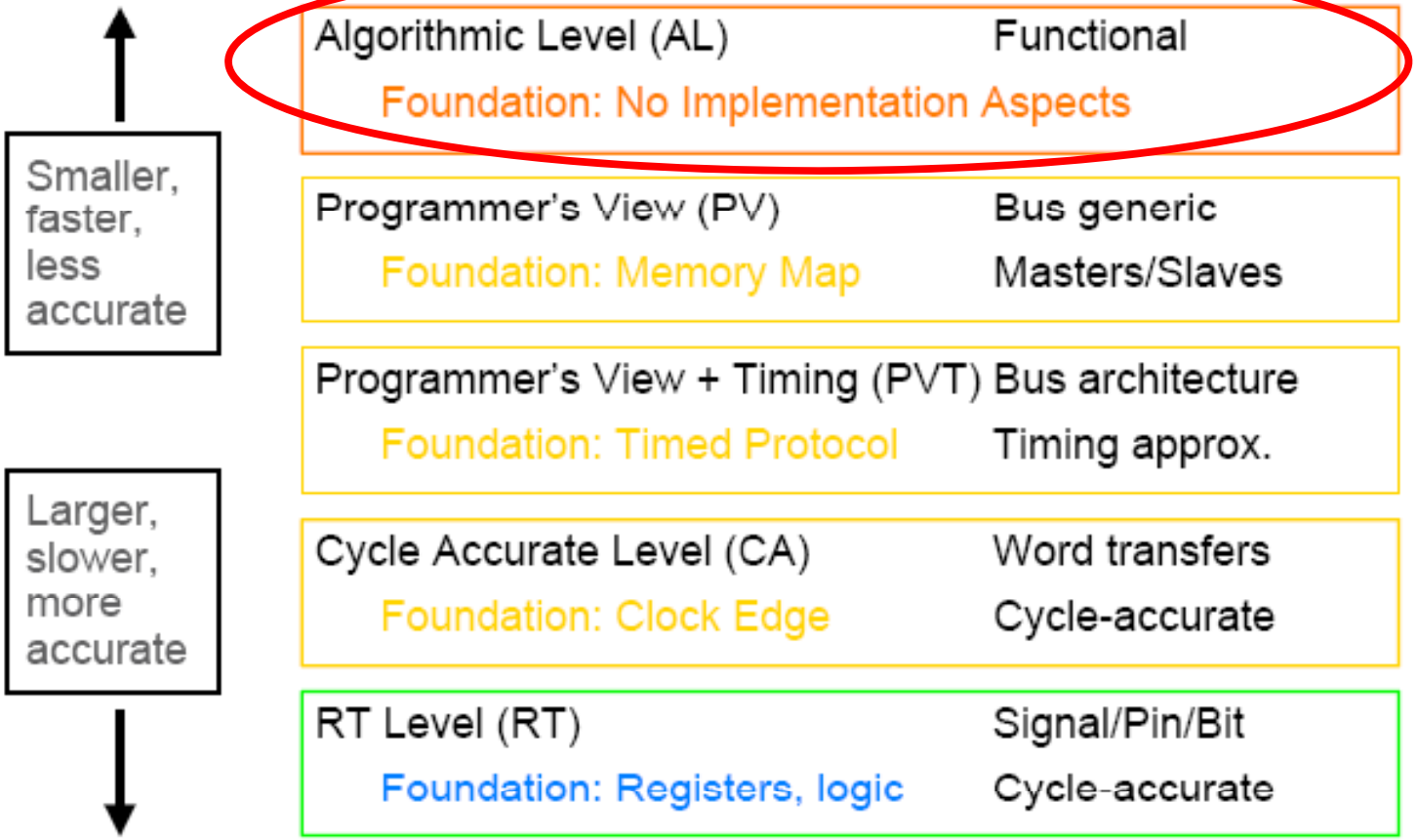
New solutions should be oriented to:

1. Maintaining and promoting the IP reuse concept, yet consider drawbacks
2. **Raising design abstraction above Register-Transfer Level --> System-Level Design**
   - shorter time to market
   - good potential to perform architecture analysis and functional verification
   - Find the right trade-off between speed (pure algorithmic model) and accuracy (Register-Transfer Level)

From RTL to TLM

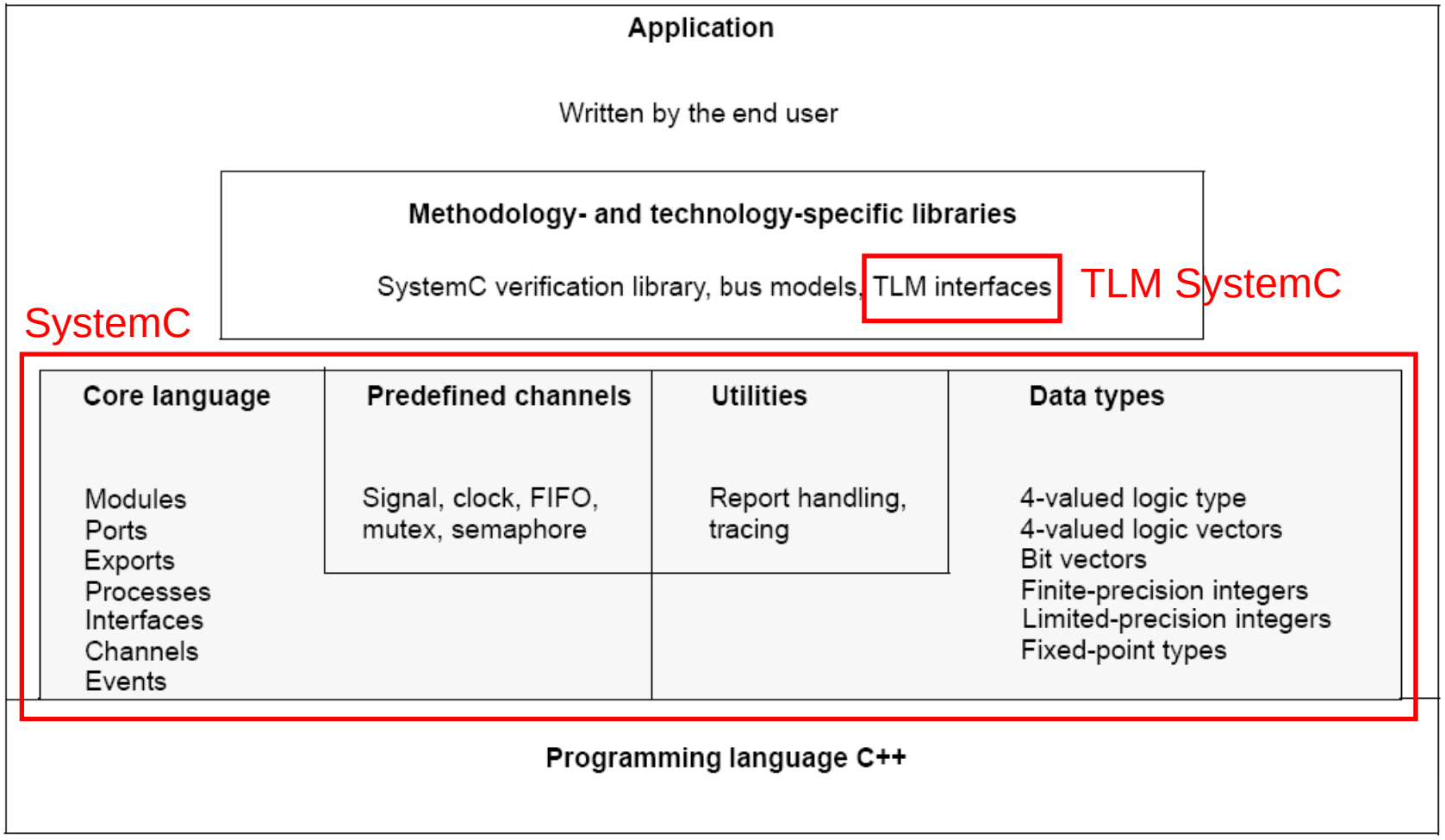
- In 1999, Coware and Synopsys propose standardization of a C++ set of classes for hardware modeling --> it is the beginning of SystemC
- The first SystemC version was basically targeting an RTL implementation, then it became more abstract
  - Channels are introduced
  - For system-level synchronization, events have slowly replaced clocks
- Some features of SystemC are enhanced, especially those related to implementing the communication aspects --> TLM becomes also a standard

Levels of abstraction



Model at a few levels that target the "pain" and risk in your D&V flow

SystemC Language Architecture



Transaction-Level Modeling (TLM)

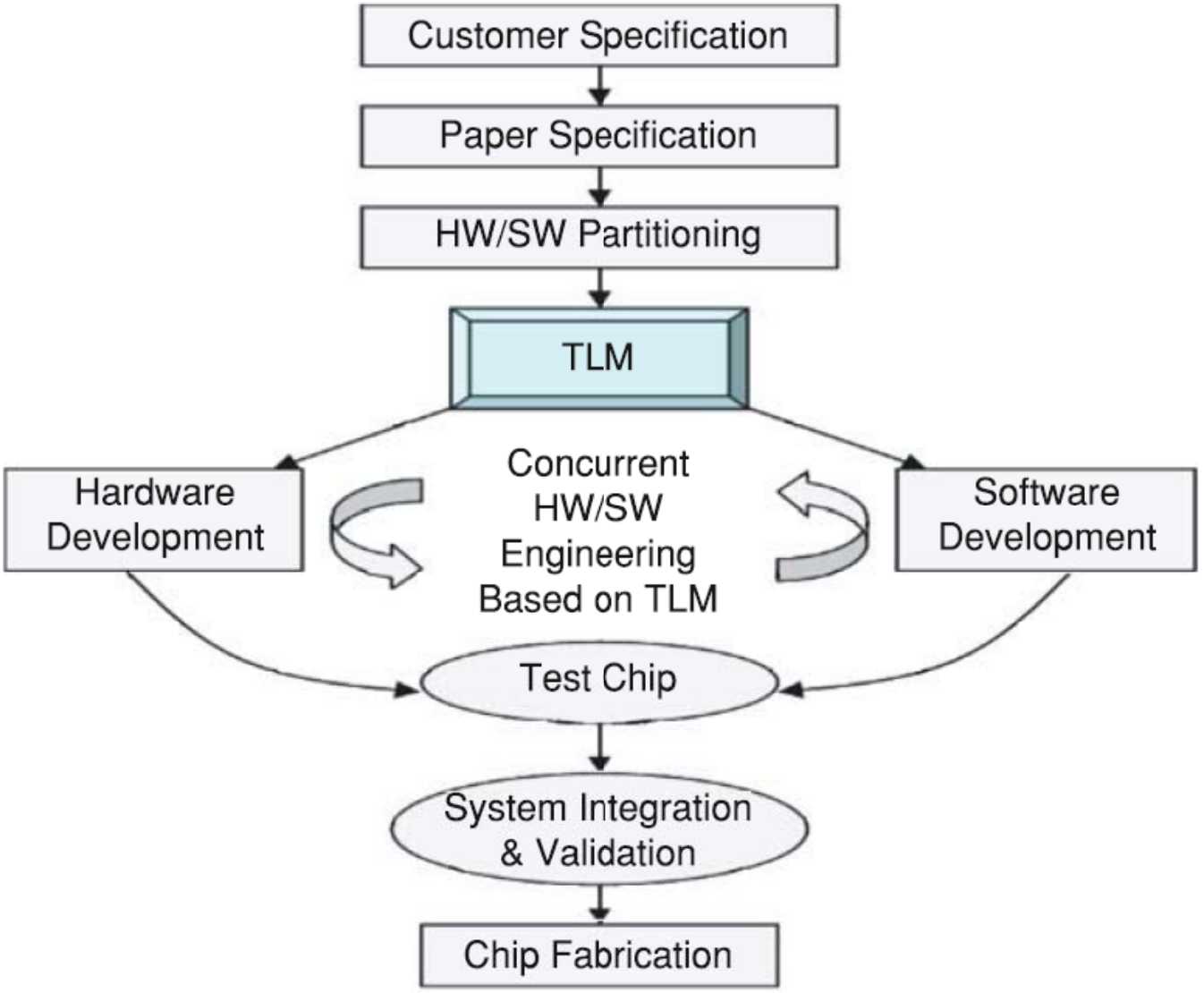
- Transaction: the data transfer (i.e. communication) or synchronization between two modules at an instant (i.e. SoC event)
- Founded on high-level programming languages such as SystemC
- TLM highlights the concept of separating communication from computation within a system
- TLM allows early software development and early architecture analysis thanks to an adequate trade-off between speed and accuracy

Transaction-Level Modeling (TLM)

- Components are modeled as modules
- Modules contain processes representing their behavior
- Modules exchange communication in form of transactions through channels
- Channels implement TLM interfaces
- Processes access these interfaces through modules ports
- Interfaces are the ones that allow the actual separation between communication and computation within a TLM system

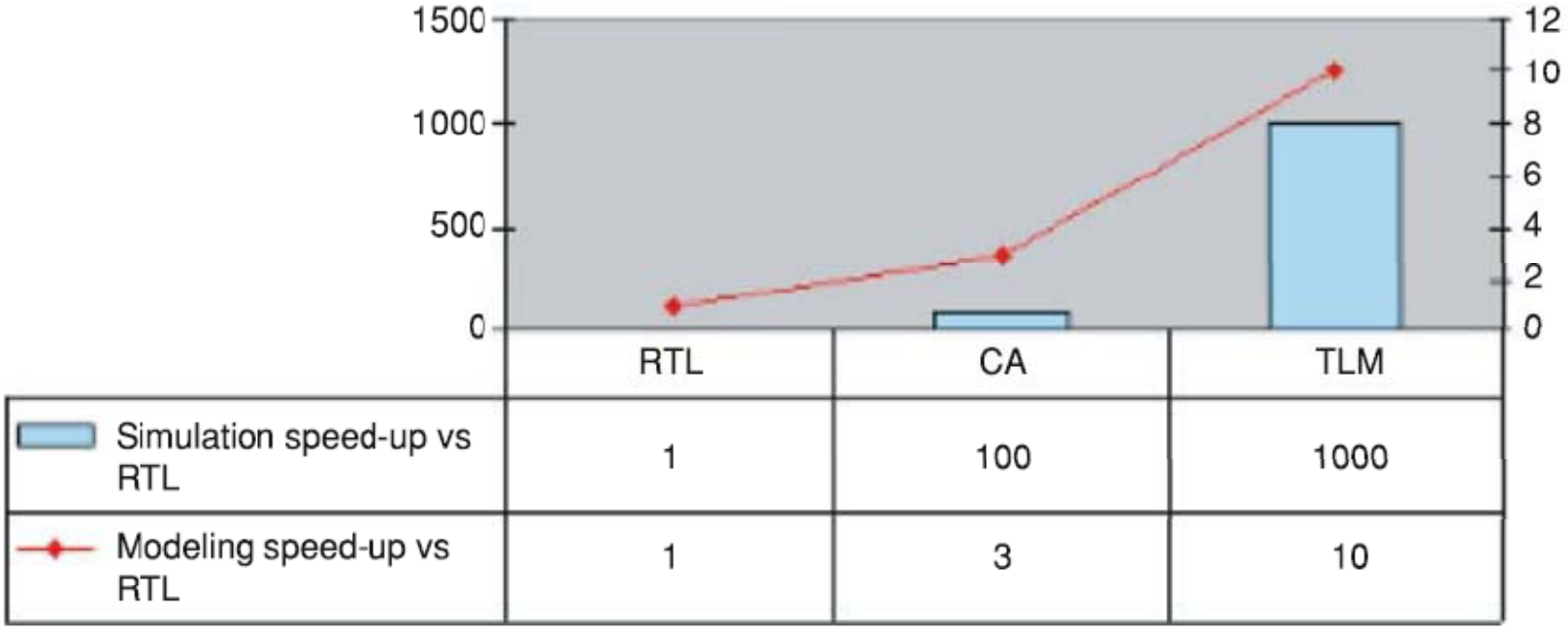
- **TLM defines a standard and a set of rules, relying on SystemC, that formalize how communication should be implemented**

The new SoC design flow



Source: F. Ghenassia. *Transaction-Level Modeling with SystemC, 2005*

Speed: RTL – CA – TLM



	RTL	CA	TLM
Simulation speed-up vs RTL	1	100	1000
Modeling speed-up vs RTL	1	3	10

Source: F. Ghenassia. *Transaction-Level Modeling with SystemC, 2005*

Summary

- Classic SoC design flow based on RTL has limitations
- SystemC/TLM is suitable for solving such limitations
  - Early software development
  - Architecture analysis
  - Functional verification
  - Consistency between work done in different teams

Further reading

- F. Ghenassia. *Transaction-Level Modeling with SystemC, 2005*
- D. C. Black and J. Donovan. *SystemC: From The Ground Up*, 2004

Thank you!