

Saint - Petersburg State University of Aerospace Instrumentation, Russia;  
Nokia Research Center and Nokia Devices , Finland.

# SystemC Modelling of the Embedded Networks

**Valentin Olenev, Yuriy Sheynin, Elena Suvorova;  
Sergey Balandin, Michel Gillet.**



Nokia Research Center



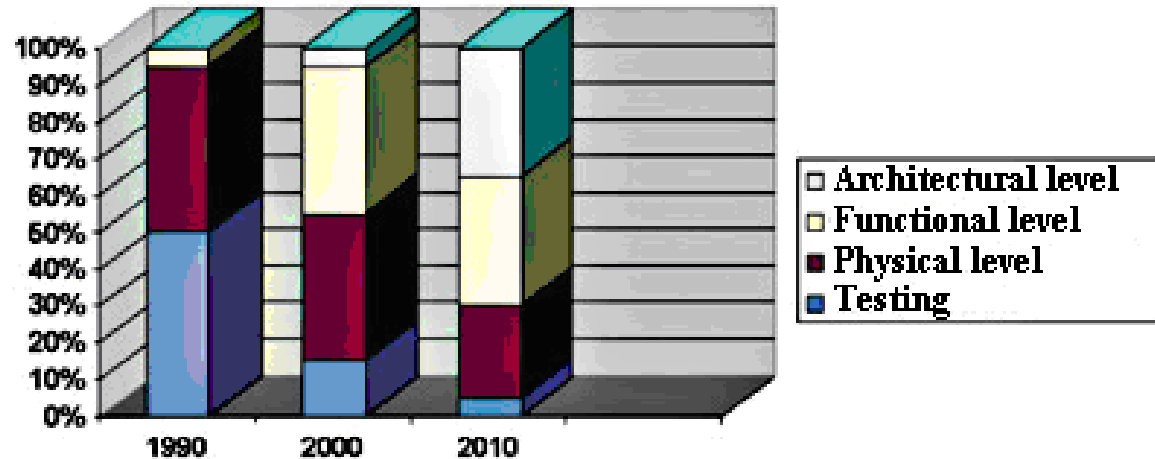
# Embedded systems

***The embedded system*** is a special kind of computer system where the processing logic is built into the device which it operates.

The most standard set of requirements to such systems are:

- Small power consumption;
- Small physical sizes;
- Absence of active cooling;
- The processor, system logic and other, are often combined on one chip.

# Designing tendencies at various levels of abstraction

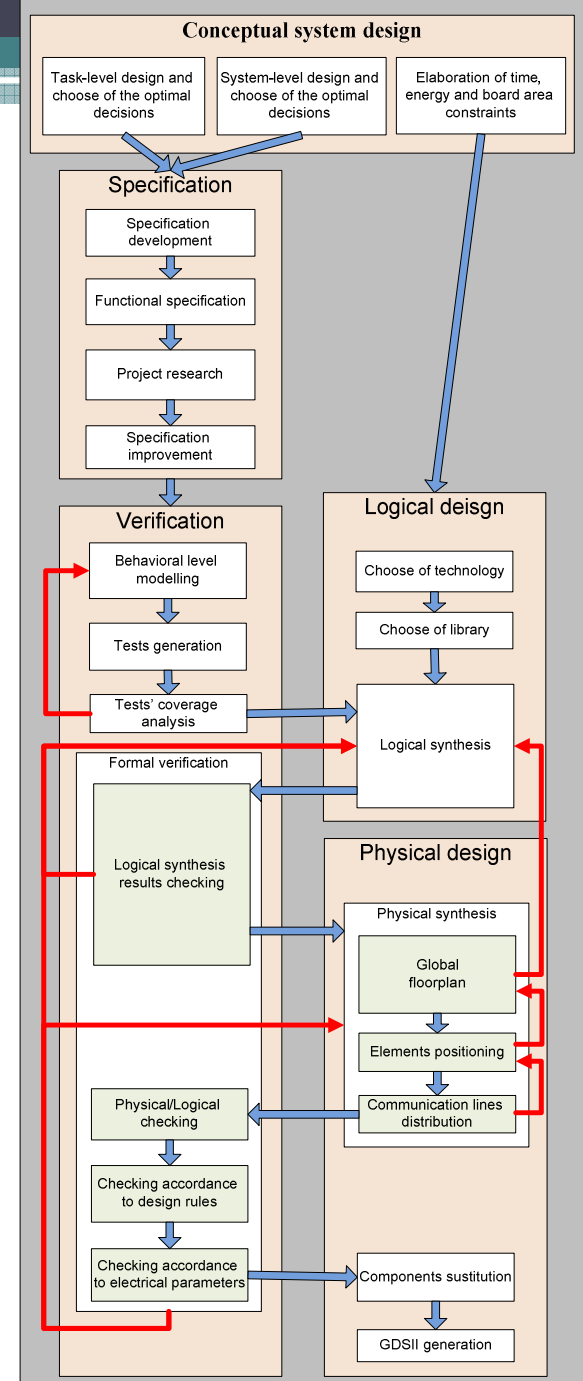


Systems design has a number of difficulties caused by:

- increasing complexity of projects
- increasing requirements to products reliability
- power consumption
- necessity of the shortest terms of the design.

# System general design flow:

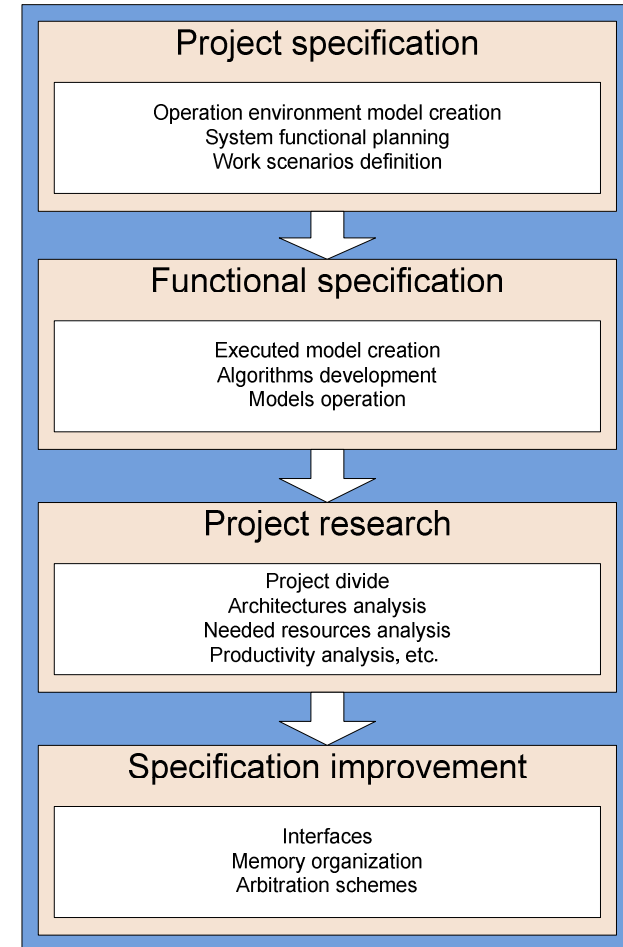
- **Conceptual system design;**
  - directions choice
  - analysis of the system
  - development of system specification drafts
- **Specification;**
  - final version of specification is maintained
  - the model in a high level language is implemented (usually in C/C ++, SystemC, SDL);
- **Logical design;**
  - converting of the specification to the register transfer level
    - the result is a specification in Verilog/VHDL
  - converting of the specification to the gate level;
- **Project verification;**
  - verification of the project and design decisions on conformity to the initial specification and to other requirements;
- **Physical design;**
  - begins from a physical synthesis
  - finishes on the final specification in the GDSII format.



# Modelling for the specification development process

At the stage of project specification modelling is applicable for the following steps:

- Creation of the system functional model;
- Description of the system in terms of the algorithms and functions which it should implement;
- System modeling in its operational environment with real data and signals;
- Specification of the system architecture in terms of necessary resources and hardware-software realization of functional model.



# Modelling for systems implementation validation

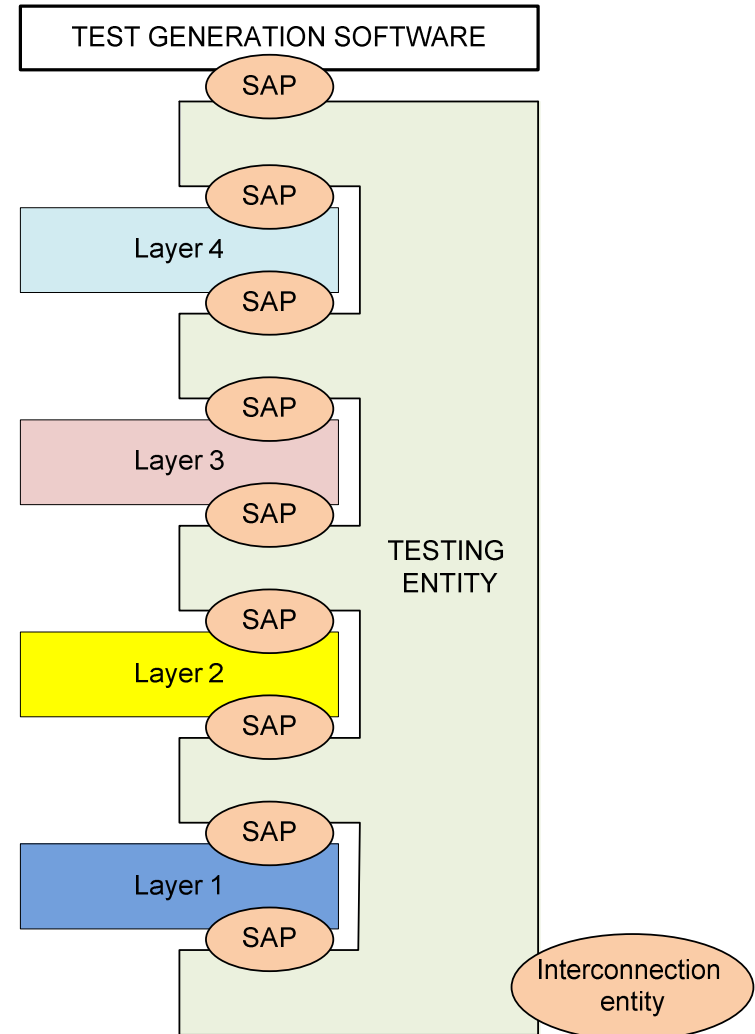
Main requirements for structure of functional design and verification tools are:

- analysis of the architecture, productivity and other system parameters;
- hardware-software design of systems, possibility of joint development and verification of hardware and embedded software;
- system design with processor blocks, use of processors' models in hardware and software development;
- uniform design environment, from the system level to the register transfer level and the gate level with C, C ++, SystemC languages support and hardware description languages such as VHDL and Verilog;
- libraries and high level constructs for functional blocks and communication channels, connectivity tables included;
- tools for the projects data control and documentation.

# Modeling for device testing

Such a Tester allows:

- to validate the standard specification itself, e.g. in the course of the standard development and evolution;
- to validate the model itself for correspondence to the specification;
- to test prototypes or boards in production;
- to verify products for conformance with the standard.



# Requirements to the design language

- It should be based on a uniform notation for a possibility of simultaneous work by both programmers and hardware designers;
- It should support special primitives for verification at different abstraction levels, should be applicable to use at all levels;
- It should support software reuse, reuse some its structures, model components;
- It should support detail of implementation or synthesis up to the RTL;
- It should be able to integrate both hardware and software components models;
- It should have processor simulators to execute compiled for them object codes;
- It should support integration of different computing models;
- It should allow co-simulation and interaction of models at different abstraction levels: functional, architectural, transactions level, PHY level;
- It should support modeling of communications at different abstraction levels and integration of these models into efficient system model.



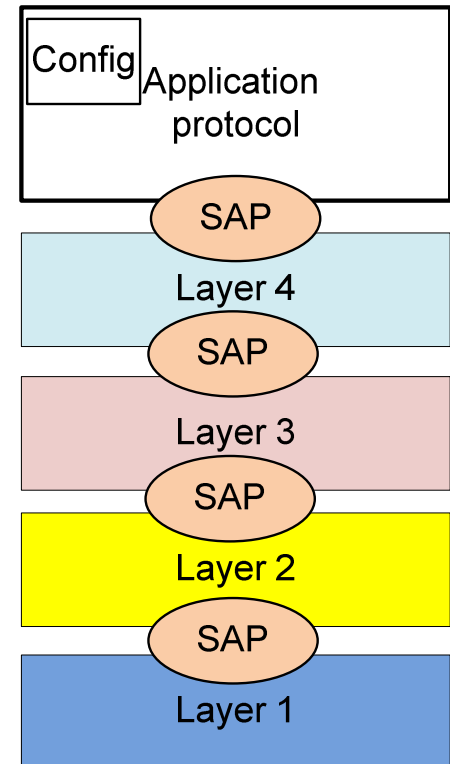
# SystemC advantages



- SystemC is based on C++, it is constructed on the basis of libraries extension and the simulation kernel, written in C++.
- SystemC uses such primitives as channels, interfaces and methods, it gives high flexibility in modeling that could be based on various computation models, provides possibility to integrate and use these models in parallel.
- SystemC supports models at all abstraction levels and uses, within the scope of OSCI (Open SystemC Initiative), standard semantics at transactions level and API; it guarantees possibility to work simultaneously with different abstraction level models.
- Most of developers, software and hardware experts are familiar with C/C++ and it point makes cooperation in HW/SW design easier.
- SystemC supports modern methodologies of verification by means of SystemC multilevel verification library (SCV - SystemC Verification library).
- SystemC has been created for support IP-blocks of based design, reuse of design and verification components.
- SystemC supports hardware modeling and detailing of a project to the RTL level.
- Modeling of on chip communications is naturally supported in SystemC.

# SystemC in system modeling

- In a system design flow main SystemC modeling areas are:
  - Functional modeling of system algorithms.
  - Modeling of system architectures at the transactions level.
  - Modeling of RTL and binding SystemC with implementation paths in the design flow.
- It is used not only to describe models, but also to develop a test environment for these models and to use it for testing of real boards (SystemC verification library).
- Testing of operation of several protocols, working one "over" another in a protocol stack.
  - By SystemC it is easy to describe structures, which are responsible for configuration of underlying model under needs of overlying, as well as to set various variants of configuration and functioning of the "upper" protocol.



# Verification with SystemC



- Self-diagnostic (introspection) and possibility to work with any type of data items.
- Transactions writing, that makes transactions-based verification (TBV) a supplementing part of transaction level modeling.
- A simple randomization based on the set of limitations. Simple randomization can create distribution of values based on complex sets of correct intervals, and to set empty intervals.
- Randomization allows to link probabilistic distribution with values or with a subset of values in defined interval. So complex distributions can be associated with test platforms.
- Complex casual scenarios on test platforms could be generated by setting complex conditions and combinations in specification of allowable values sets of testing variables.
- SCV also provides a simple database to store and investigate verification results, gives simple SystemC to HDL simulator linkage mechanism, compatible mechanism for errors processing and debugging, error detection mechanism

# Conclusions

- Modelling allows to accelerate and to simplify the design process for improvement of the standards quality and avoidance of specification errors.
- It helps to save the finance of the companies-developers.
- SystemC language occupies leading positions of modelling, as the most adapted, simple and clear language for off-site users.
- New SystemC standard updates also introduce ability to verification that expands the language applicability sphere.
- SystemC combination with other simulation languages is very perspective direction. (e.g., joint possibility to use SDL and C++).



**Thank You**