



SystemC 2.0 Specification and Benefits

Thorsten Grötter
Synopsys, Inc.

Motivation

- **SystemC 1.0**
HW modeling (RTL and behavioral)
- **SystemC 2.0**
extend scope to System-Level Modeling

- **System-Level Modeling**
 - functional models
 - transaction-level platform models
 - high-level architecture models

How do we achieve this?

- We use a flexible and powerful *Model of Computation* (MoC).
- A MoC is characterized by
 - the model of time employed,
 - the rules for process activation, and
 - the supported means of communication.

MoC: Model of Time

- SystemC 1.0
 - Relative floating-point model of time (double)
- SystemC 2.0
 - Absolute (64 bit) unsigned integer model of time
- Why?
 - Avoid finite precision effects, e.g. underflow
 - Use absolute model of time: define time units (IP exchange)

MoC: Rules for Process Activation

- **SystemC 1.0**
 - Static sensitivity
 - ◆ Processes are made sensitive to a fixed set of signals during elaboration
- **SystemC 2.0**
 - Static sensitivity
 - Dynamic sensitivity
 - ◆ The sensitivity (activation condition) of a process can be altered during simulation (after elaboration)
 - ◆ Main features: events and extended wait() method

Waiting

```
wait(); // as in SystemC 1.0
wait(event); // wait for event
wait(e1 | e2 | e3); // wait for first event
wait(e1 & e2 & e3); // wait for all events
wait(200, SC_NS); // wait for 200ns

// wait with timeout
wait(200, SC_NS, e1 | e2);
wait(200, SC_NS, e1 & e2);
```

MoC: Communication

- SystemC 1.0
 - Fixed set of communication channels (sc_signal, ...) and ports (sc_in, sc_out, ...).
 - SystemC 2.0
 - user-defined
 - ◆ interfaces
 - ◆ channels
 - ◆ ports
 - richer set of predefined channels (HW signals, FIFO, semaphore, mutex, ...)
- } Define your own bus, message queue, ... etc.

Interfaces and Channels

- An interface is a set of methods implemented by a channel.

```
struct write_if : public sc_interface
{
    virtual void write(char) = 0;
    virtual void reset() = 0;
};
```

```
struct read_if : public sc_interface
{
    virtual void read(char &) = 0;
    virtual int num_available() = 0;
};
```

- A channel can implement multiple interfaces.

Ports

■ Ports ...

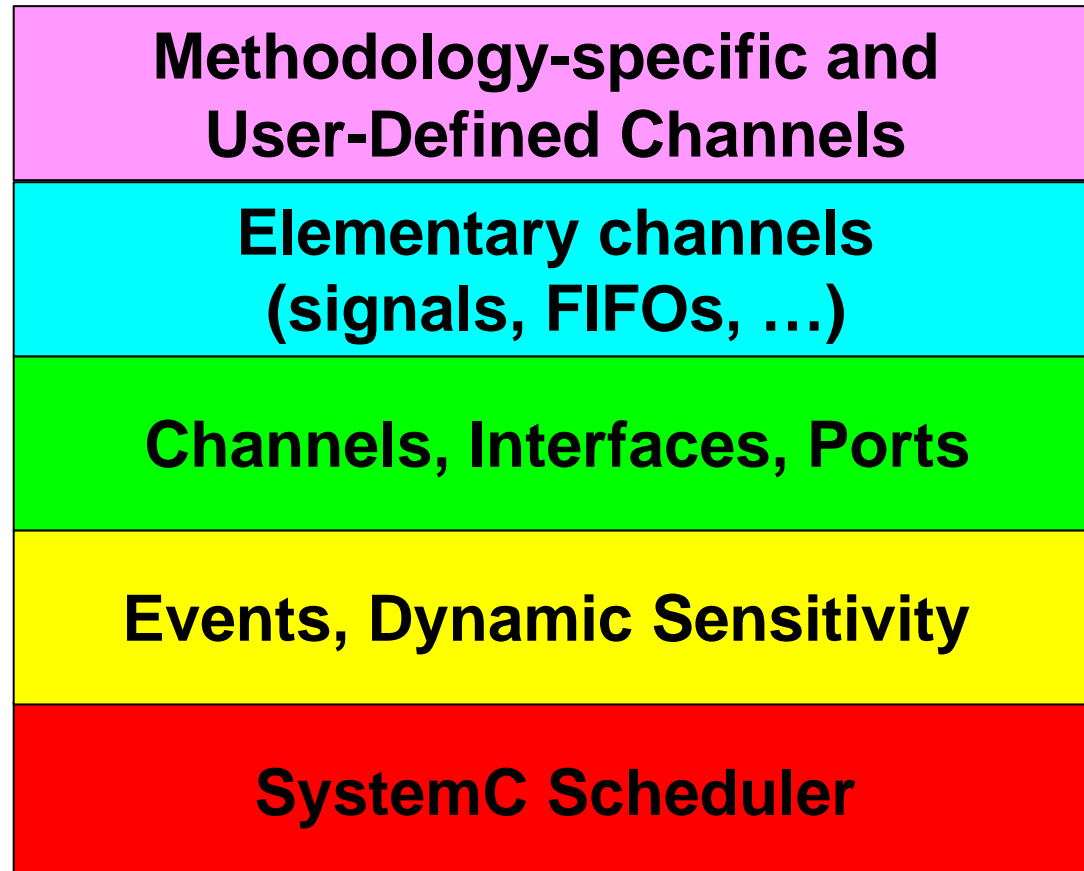
- connect modules and channels
- specify the required interface (e.g. `sc_port<IF>`)
- give modules (processes) access to interface methods

```
sc_port<write_if> p;  
  
void some_process() {  
    ...  
    p->reset();  
    p->write('X');  
    ...  
}
```

Primitive and Hierarchical Channels

- Primitive channels
 - are atomic entities
 - have no visible internal structure
 - can use request-update scheme (HW signals)
- Hierarchical channels
 - are modules that implement interfaces
 - can have ports
 - can contain processes, modules, and channels
- Both implement interfaces

Architecture of SystemC 2.0

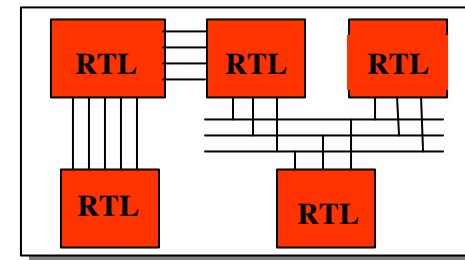
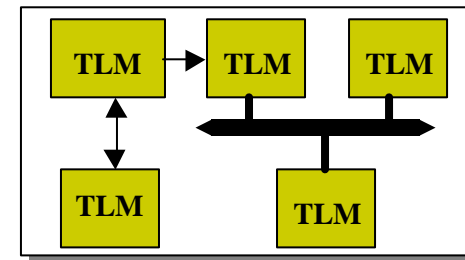
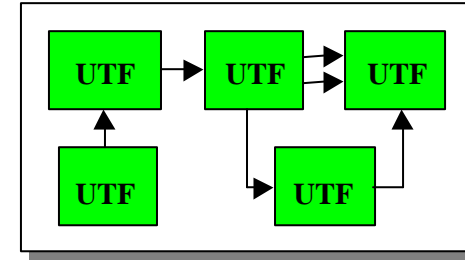


Model of Computation

- Very powerful and flexible
- Supports well known MoCs such as
 - discrete-event models
 - ◆ RTL / behavioral HW models
 - ◆ network modeling
 - ◆ transaction-level SoC platform modeling
 - Kahn process networks
 - ◆ static multi-rate data flow
 - ◆ dynamic data flow
 - Communicating Sequential Processes

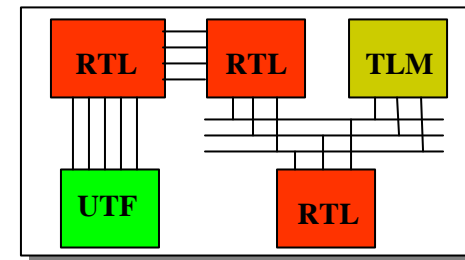
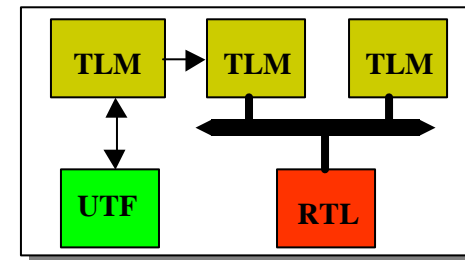
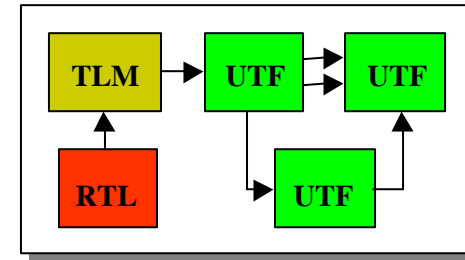
SystemC 2.0: A single language spanning multiple levels of abstraction

- (untimed) functional level
 - executable specification
- transaction level
 - platform design, HW/SW co-verification
- pin level
 - RTL/behavioral HW design and verification



Gradual refinement, early verification

- No need to refine executable specification in one giant step into RTL model
- Bus-cycle accurate transaction level models for fast platform simulation ($\gg 100$ k cycles/sec) early in the development process
- No need to glue together different simulators for co-simulation



Benefits of SystemC v2.0

- Enables, fast smooth system design
 - Communication can modeled and refined independent of function
- Supports virtually all system modeling needs
 - Flexible semantic foundation additions support most models of computation within one environment
 - Leverages all existing v1.0 and v1.1beta capabilities
- Broadly applicable, “best of breed” solution
 - Designed by 12 experts from six different EDA and System IC companies
 - Tuned for both EDA tool and IP use