Towards Heterogeneous Simulation Kernels for System Level Models
Main Contributions

- Introduce Heterogeneity in SystemC with Models of Computation (MoC) extensions
- Extend SystemC simulation framework with:
  - Synchronous Data Flow (SDF)
  - Finite State Machine (FSM)
  - Communicating Sequential Processes (CSP)
- Raise the Modeling Fidelity
- Step towards Behavioral Hierarchy with Heterogeneity.
SystemC’s Discrete-Event Kernel

- Evaluate-Update Paradigm
- Dynamic scheduling incurs unnecessary delta cycles
- Statically schedulable MoCs should avoid dynamic scheduling
An Example MoC Extension: Synchronous Data Flow in SystemC

- SDF models are:
  - Amenable to static scheduling
  - Require blocks to have predefined production and consumption rates
  - Construct repetition vector
  - Construct firing order
  - Executable schedule achieved with valid repetition vector and firing order
An Example MoC Extension: Synchronous Data Flow in SystemC

- Constructing an SDF model requires:
  - Encapsulating SDF specific processes (SC_SDF_METHOD) in a top-level SystemC process
  - Top-level constructs SDF graph with appropriate API
  - Every unique SDF model must be registered in its own sdf_graph instance
  - Top-level entry function must invoke sdf_trigger() function

```cpp
SC_MODULE( sdf_block ) {
    SDFPort< sc_uint< 8 > > sample_in; SDFPort< sc_uint< 8 > > sample_out; SC_SDF_METHOD( block_entry );
    SC_CTOR( sdf_block ) {
        // Register this block into sdf_graph
    }
    void block_entry() {
        // Entry specific code
    }
}

SC_MODULE( toplevel ) {
    sc_in_clk CLK;
    SC_THREAD( toplevel ) {
        sensitive << CLK.pos();
    }
    SC_CTOR( toplevel ) {
        // Instantiate SDF blocks and connect
        // the ports
    }
    void toplevel() {
        sdf_trigger();
    }
}
```
An Example MoC Extension: Synchronous Data Flow in SystemC

- During initialization all executable schedules are computed
- DE kernel continues executes without intervention until sdf_trigger() is invoked
- SDF kernel takes over and executes the SDF-specific blocks according to the computed schedule
Heterogeneous Extensions
Communicating Sequential Processes

- Rendez-vous communication
Heterogeneous Extensions
Finite State Machine

- Control machines
Heterogeneous Extensions
DE, FSM, SDF & CSP

- DE: Solves RSA Encryption Algorithm
- SDF: Sobel Edge Detection Algorithm
- Producer
- Consumer
- Phil1
- Phil2
- Phil0
- Fork0
- Fork1
- Fork2
- Fork3
Simulation Efficiency
A brief look

- Pure SDF models ~ 65% gains
- Pure FSM models ~ 10% degradation
- Pure CSP models ~ 1% gains
Behavioral Hierarchy with Heterogeneity

- Decompose design into small behaviors
- Behaviors expressed by different MoCs
Behavioral Hierarchy with Heterogeneity

- Semantics define interactions within MoC and across MoCs
- Hierarchical composition preserves behavioral hierarchy
Concluding Remarks

- Heterogeneous MoC-based extensions for System Level Models
- Extend SystemC kernel with MoC extensions for increased modeling fidelity
- Heterogeneity promotes behavioral hierarchy
Reference

- Website for SystemC-H: http://fermat.ece.vt.edu/systemc-h/
- Book