

FERMAT | *Hiren Patel and Sandeep Shukla* | June 2005

{hiren, shukla}@vt.edu

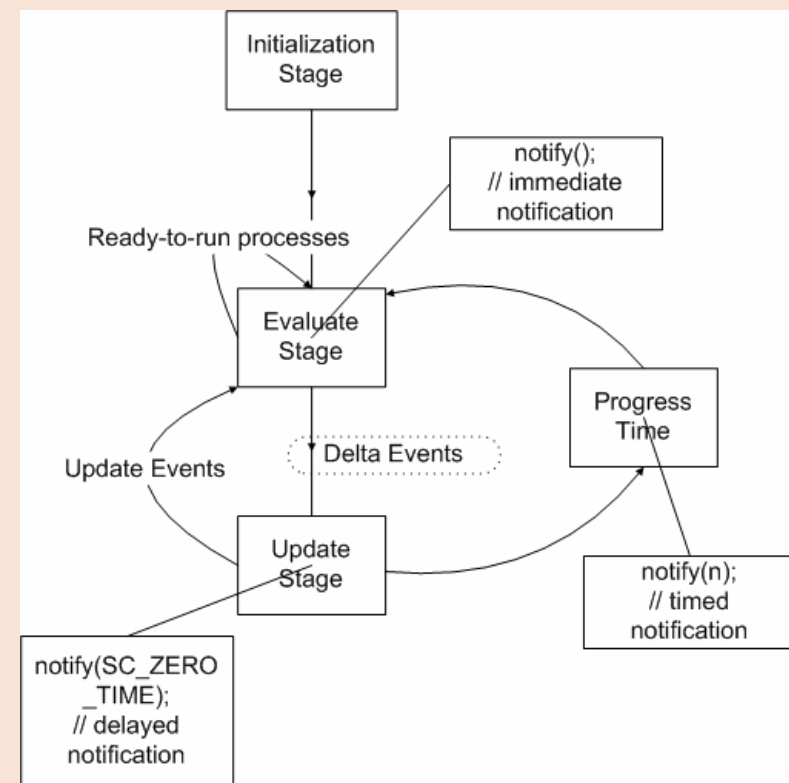
Towards Heterogeneous Simulation Kernels for System Level Models

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

- ```
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();
 }
};
```

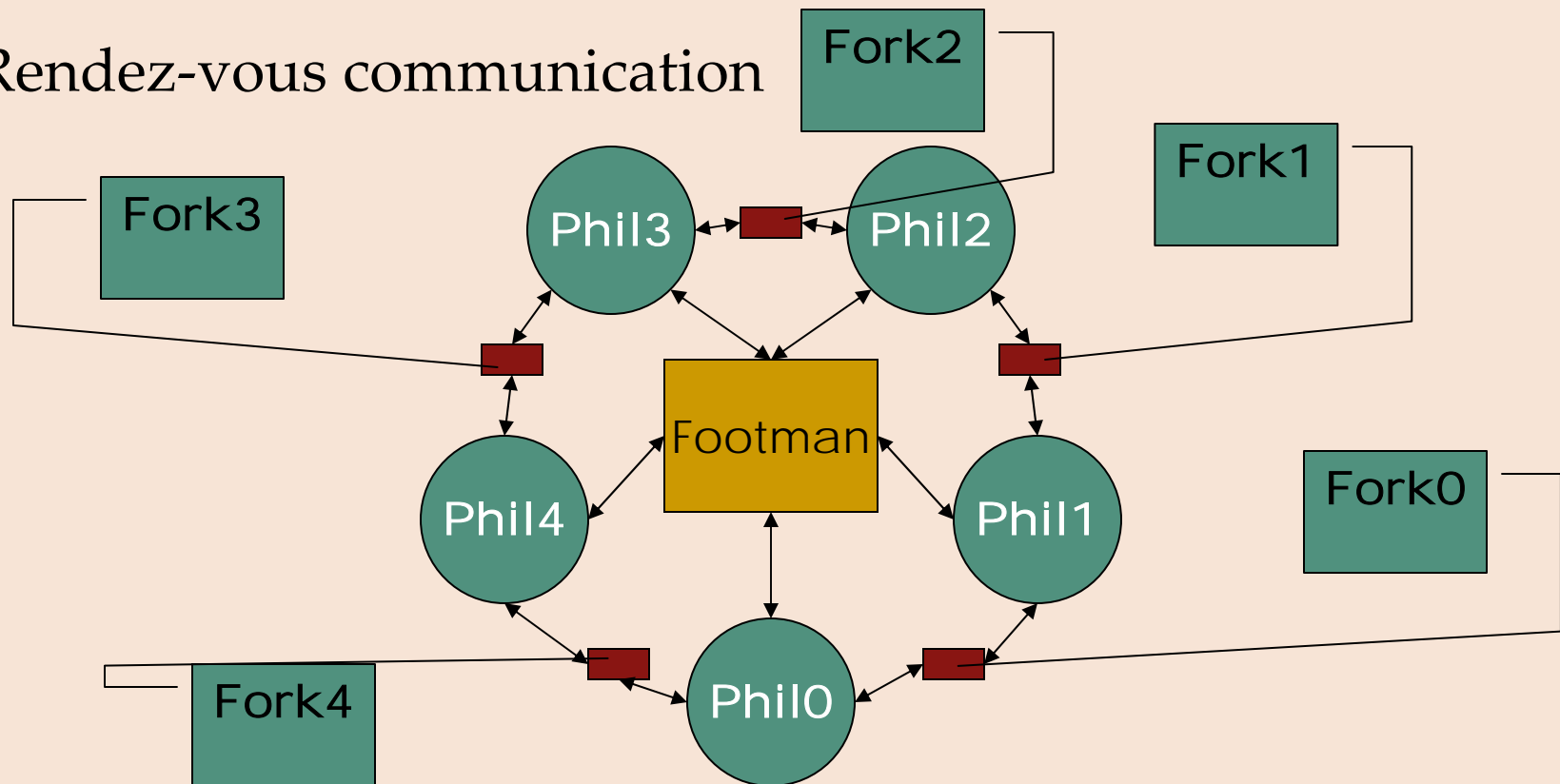
# An Example MoC Extension: Synchronous Data Flow in SystemC

- 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

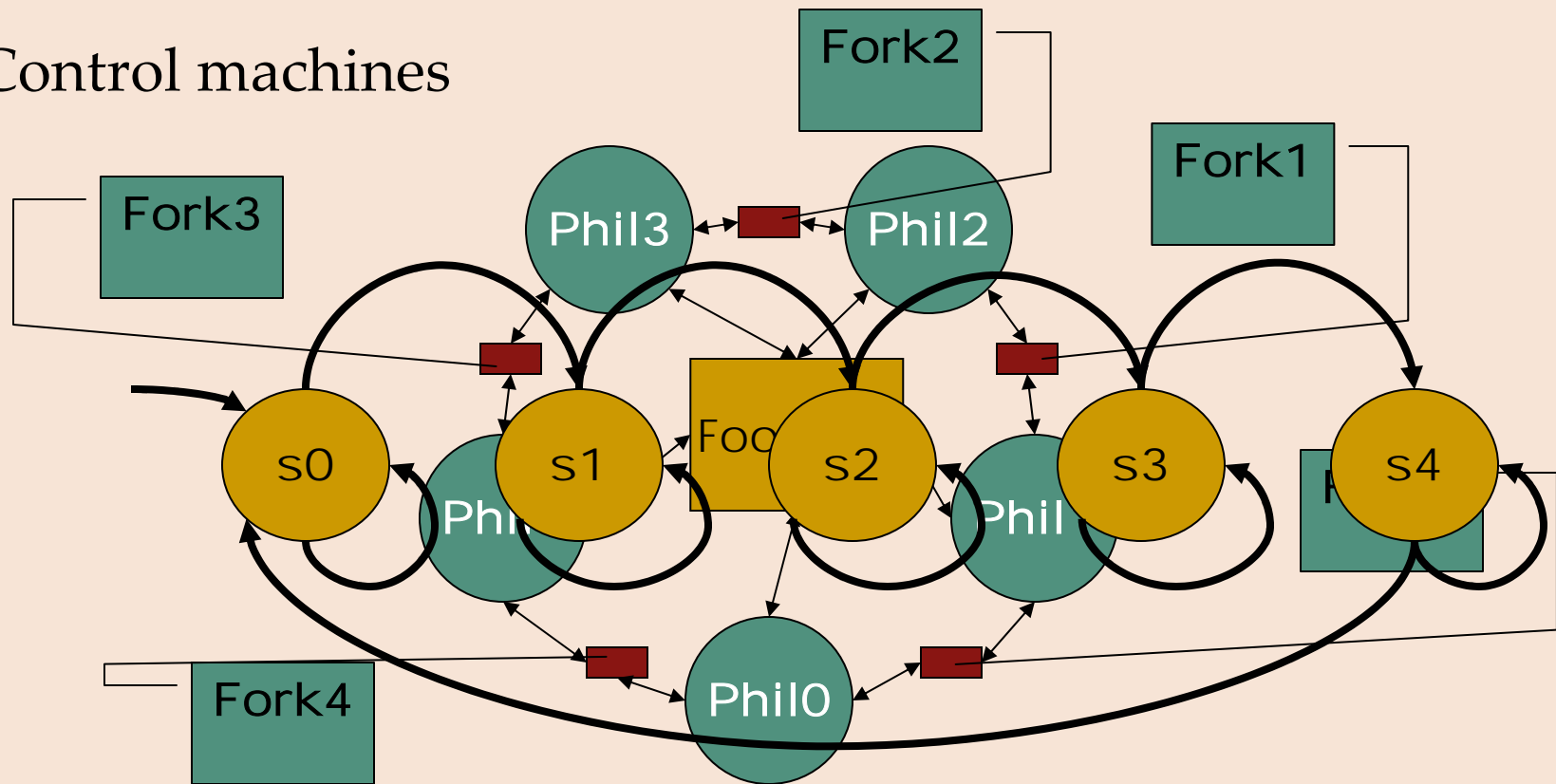
## Communicating Sequential Processes

- Rendez-vous communication

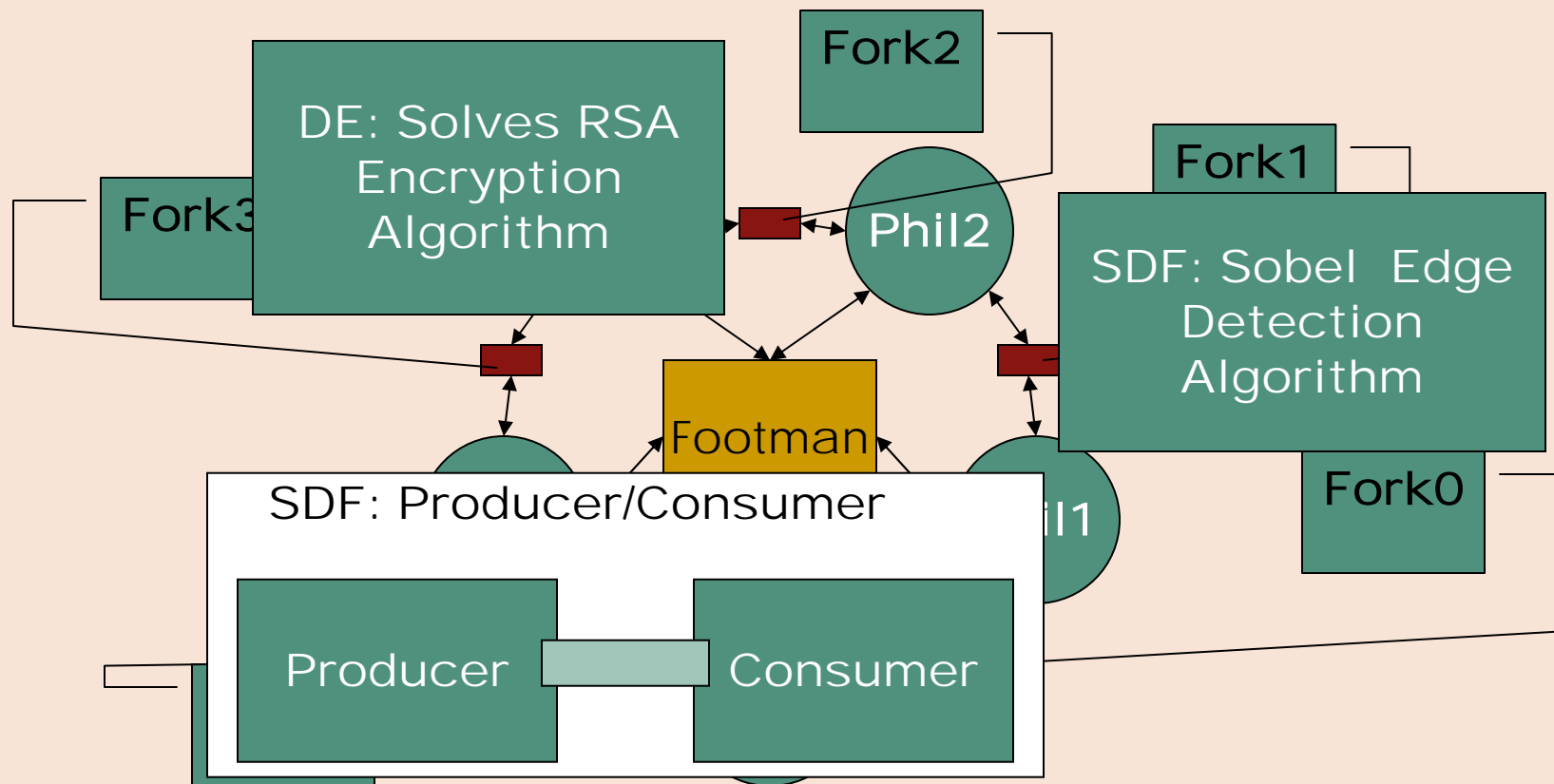


# Heterogeneous Extensions Finite State Machine

- Control machines



# Heterogeneous Extensions DE, FSM, SDF & CSP



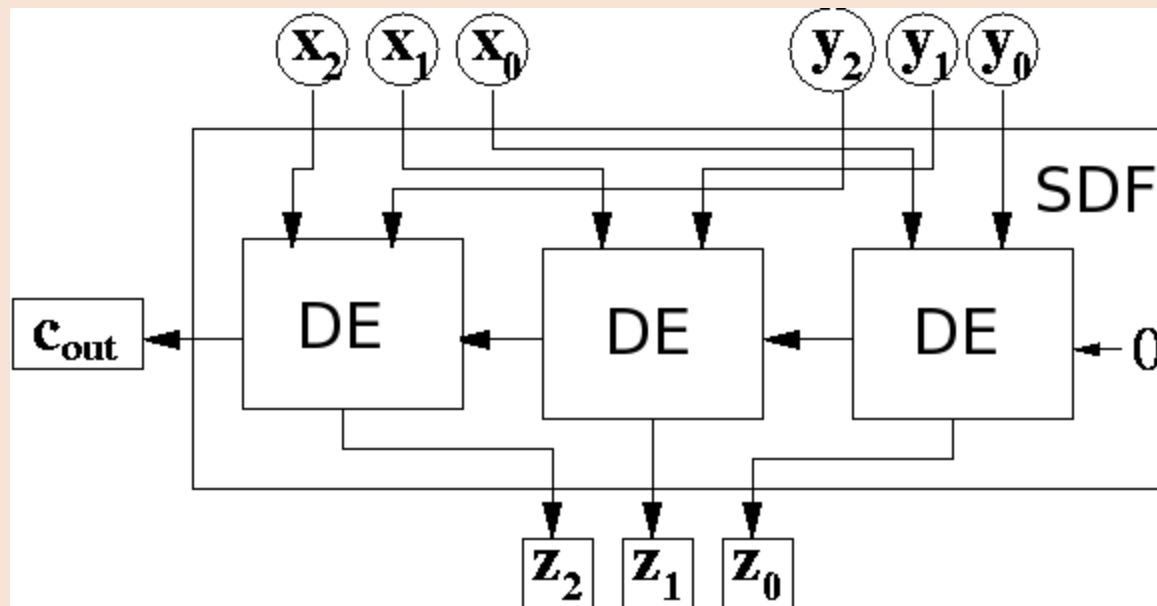
# Simulation Efficiency

## 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

