# Parallel Simulation of SystemC Loosely-Timed Transaction Level Models

## Master of Science Thesis

Konstantinos Sotiropoulos
( kisp@kth.se )

KTH Royal Institute of Technology

Intel Sweden AB

| | |
|---|---|
| Supervisor: | Björn Runåker (Intel Sweden AB) |
| Examiner: | Prof. Ingo Sander (KTH) |
| Academic advisor: | PhD student George Ungureanu (KTH) |

# Motivation

- This project stems from the work of Björn Runåker:
  speeding up the simulation of 5G radio base stations.
- A **coarse-grained** approach was adopted:
  multiple instantiations of SystemC's simulation engine.
- But motivated a **finer-grained** approach:
  parallelism within a single instance?

# Problem Statement

- The verdict is categorical:
  SystemC's Reference Simulation Environment must be **bypassed**.

- Transaction Level Modeling in SystemC:
  **breaks the separation of concerns** between execution and communication.

- **Address the question:**
  can we transform a SystemC TLM 2.0 LT model into a parallel application?

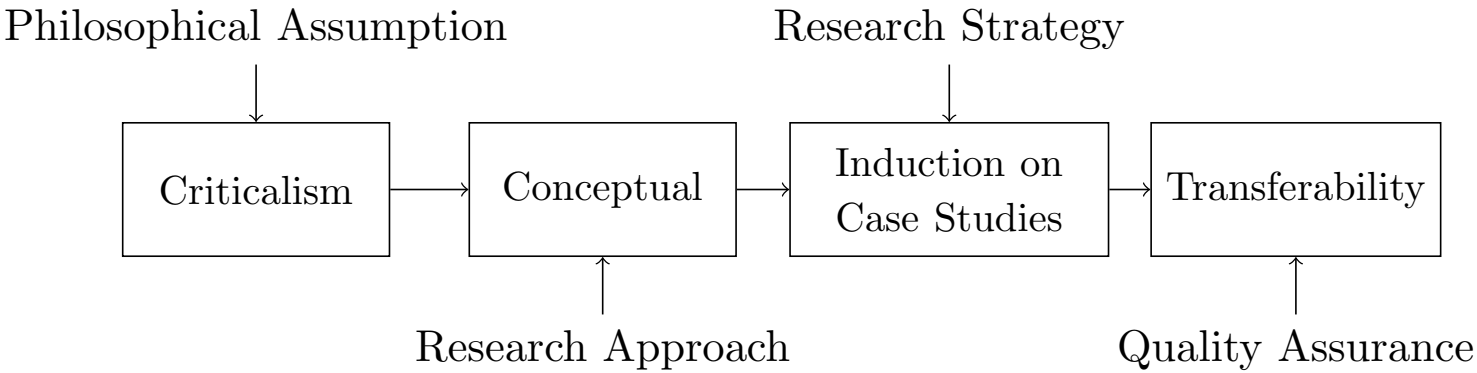
# Purpose

- SystemC TLM 2.0 used to construct **Virtual Platforms**: enabling hardware/software co-simulation.

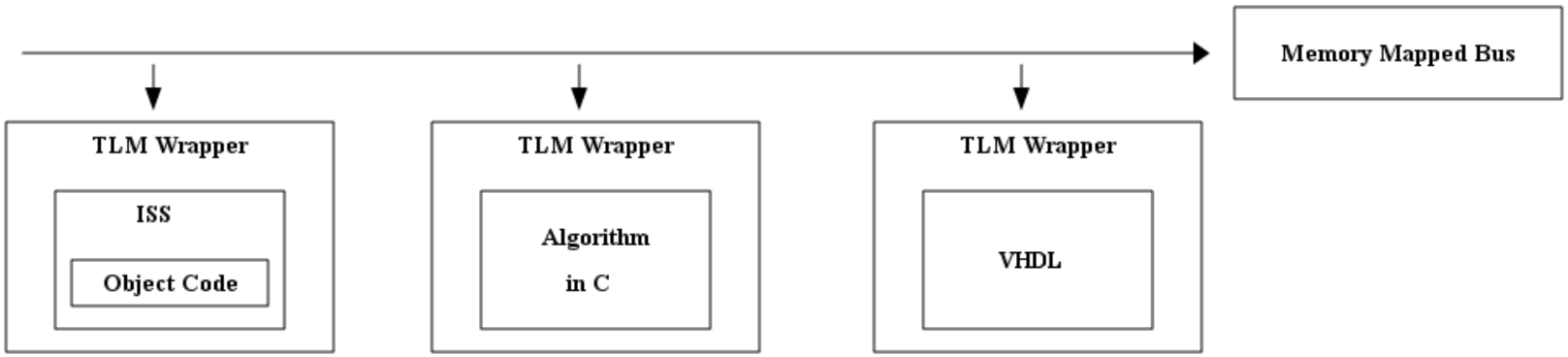
- From SystemC Evolution Day 2016:

  *"SystemC must embrace true parallelism otherwise it will go down the same path as the dinosaurs"*

# Qualitative Research Methodology

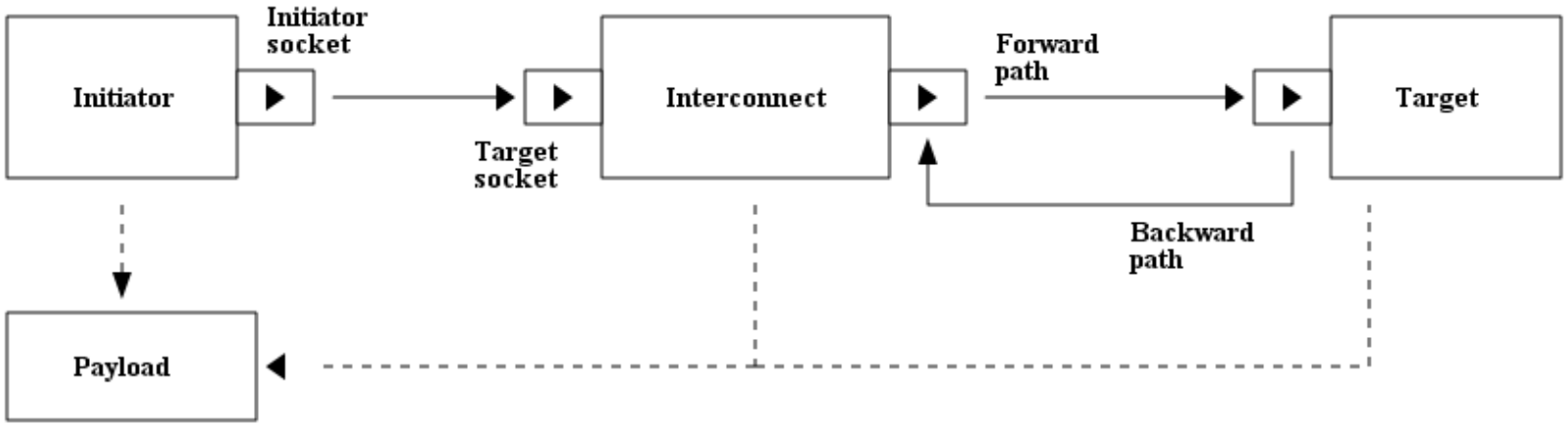


# The Role of SystemC TLM 2.0



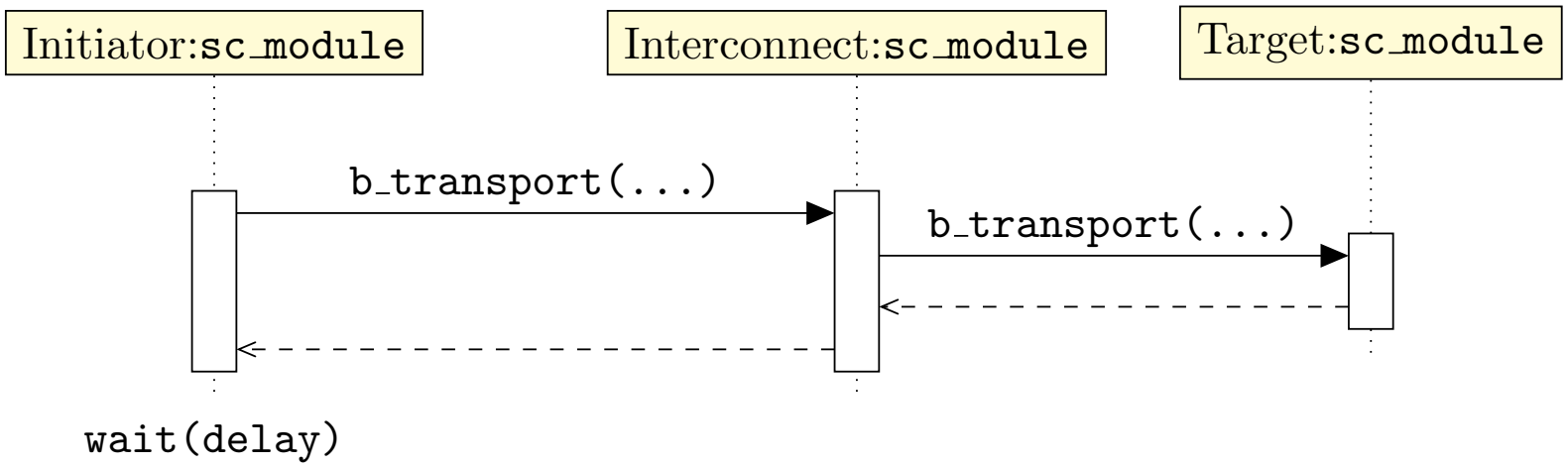

- SystemC TLM 2.0: a communication-centric API meant for **gluing** IP components

# An example TLM




- An **initiator** initiates a transaction which is routed through **interconnect** components and eventually reaches the **target** component.

# Communication in LT TLM

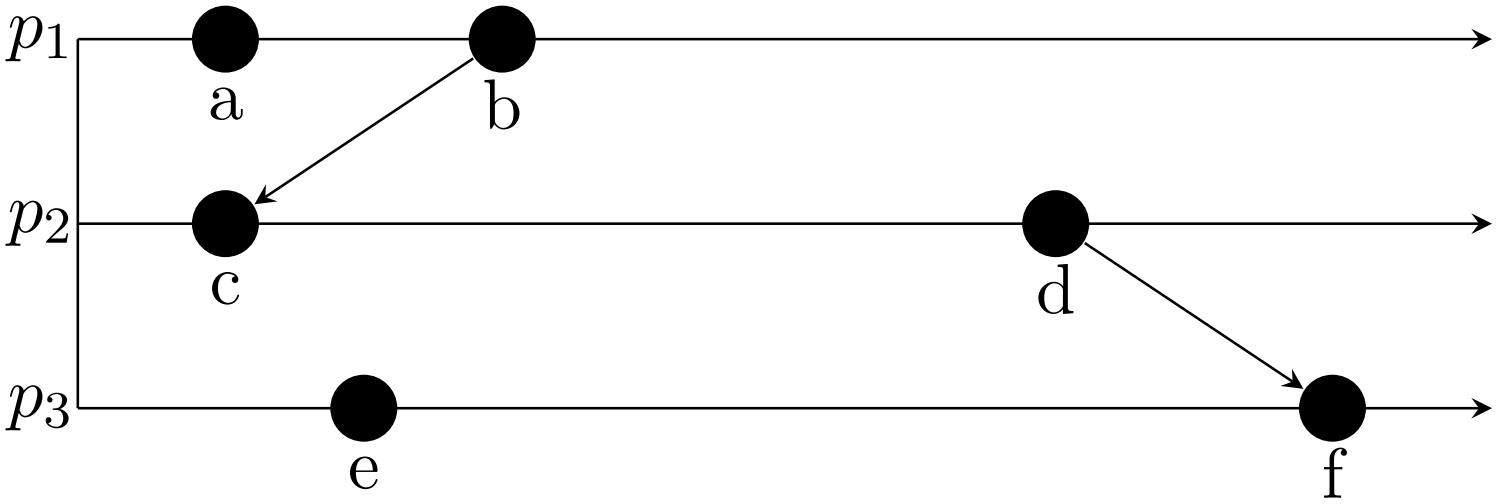



- Looks like a series of of **remote function calls**
  No Channels!

# The DE Model of Computation

- Provides the **operational semantics** of:
  Electronic System-Level Design Languages.
- A model is a system of:
  **processes** that **execute** and **communicate**
- Logic Time vs Real Time:
  logic time is also relativistic.

# The DE Manifold




- **Execution:**

$$b = f(a) \implies a \propto b \implies a \sqsubset b$$

- **Communication:**

$$c = g(b) \implies b \propto c \implies b \sqsubset c$$

# SystemC's DES

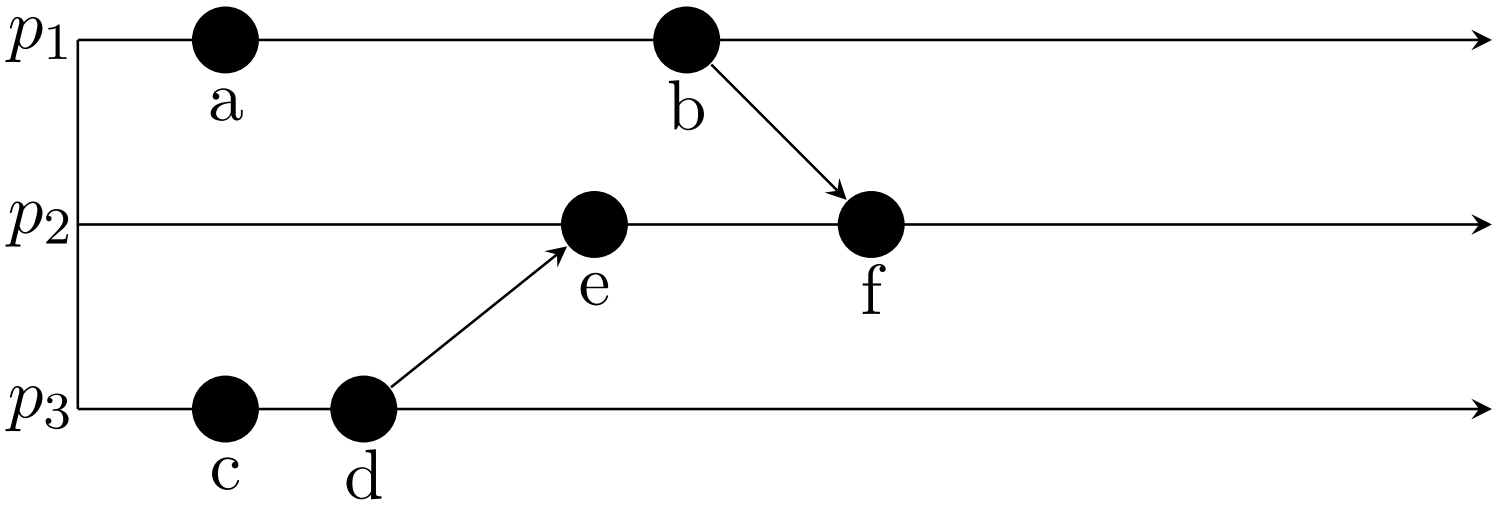
- A realization of the DE MoC:
  is a **Discrete Event Simulator (DES)**.
- SystemC's DES:
  uses **coroutines** to emulate space dimensionality.
- The **kernel** has a global perspective on logic time:
  every process implicitly communicates with the kernel.

# Parallel DES

- A Parallel DES preserves spatial decomposition:
  processes keep their own perspective of logic time.

- Communication is **Synchronization**:
  with communication processes synchronize their time perspectives.

# Causality Hazard




- Event $e$ might occur earlier in real time than $f$.
- Event $e$ may causally affect event $f$.
- How can $p_2$ determine when it is safe to advance its logic time perspective?

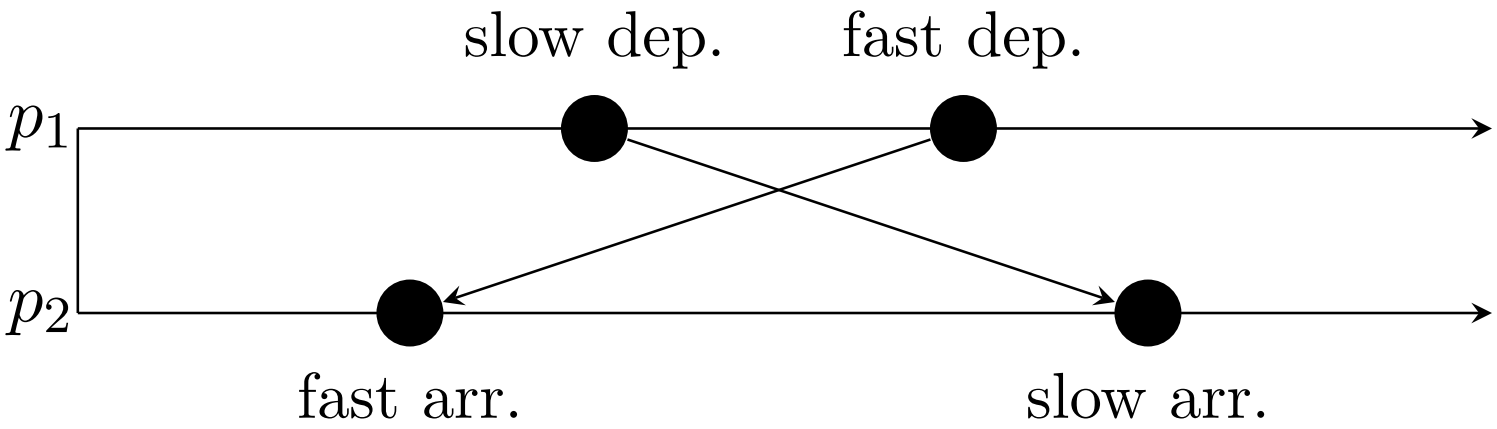
# The CMB Synchronization Algorithm

- The **Chandy/Misra/Bryant** synchronization algorithm:
  at the heart of the proposed PDES.
- **Block** a process until:
  it gathers knowledge about other processes' perspective of time.
- Implement in MPI:
  unlimited **scalability**.

# Proposed PDES Evaluation

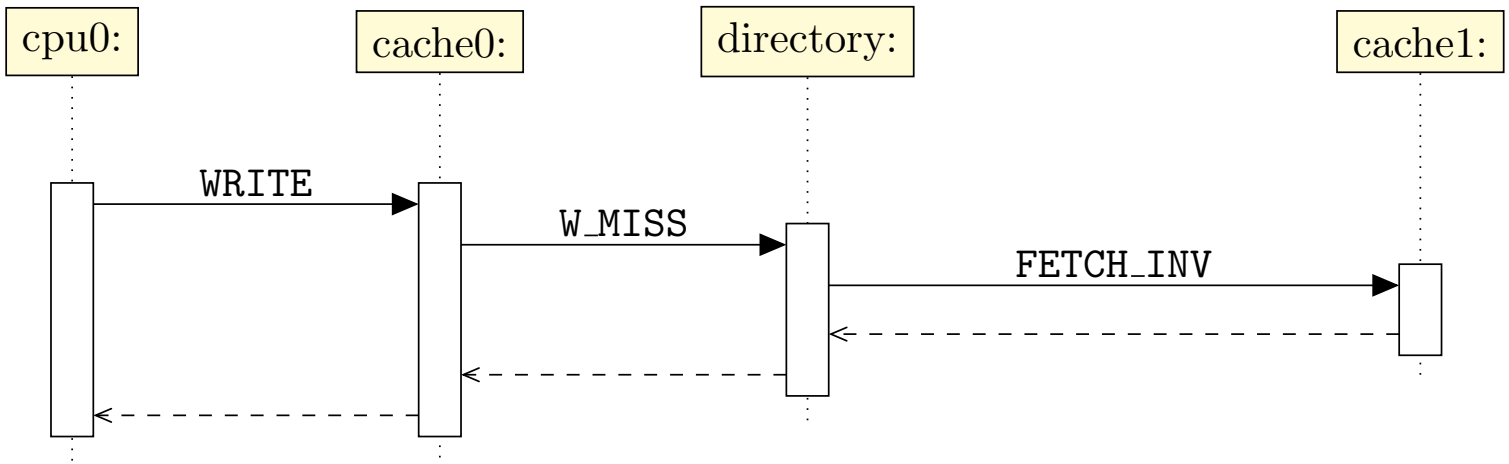
- Construct **two case** studies:
  each with a sequential and parallel version.
- **Case Study 1:** airtraffic simulation
  create a global log of departures/arrivals.
- **Case Study 2:** cache-coherent multiprocessor:
  software is emulated by a memory trace.

# Non-Monotonic Communication in Case Study 1




- Different airplane velocities

# Non-monotonic Communication in Case Study 2




```
wait(delay)
```

- A write-miss on an exclusive block

# Verdict

- **Recoding** is deemed feasible:
  but far from being fully automated.

# Contributions

- A novel presentation of the DE MoC.
- An updated CMB implementation with new features introduced in MPI 3.0.

# Future Work

- An alpha version of a **recoding infrastructure**:
  SystemC LT TLM → a parallel MPI application.

# Questions?

Thank you for your time!

`github.com/kromancer/Thesis`