A Comparison of Two SystemC/TLM Semantics for Formal Verification

Olivier Ponsini and Claude Helmstetter

INRIA / VASY

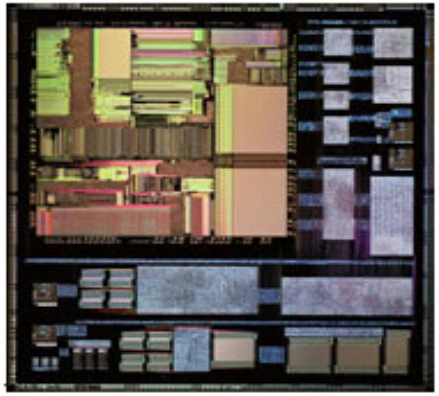
http://www.inrialpes.fr/vasy

Outline

- Transaction level modeling in SystemC
- Verifying SystemC/TLM
- The Lotos/CADP framework for semantics comparison
- Conclusion

Electronic design models

Physical chip

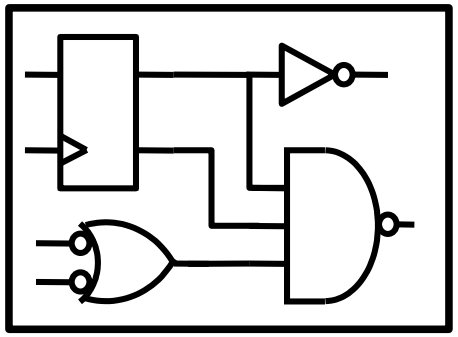


- Accurate
- Fast

but:

- Too late
- Hard to debug

Hardware description (RTL)



- Accurate
- Easy to debug

but:

- Too late
- Very slow

Abstract model (TLM)

```
...
D=read_block(A,S)
D=transform(D,P);
write_block(D,A);
...
```

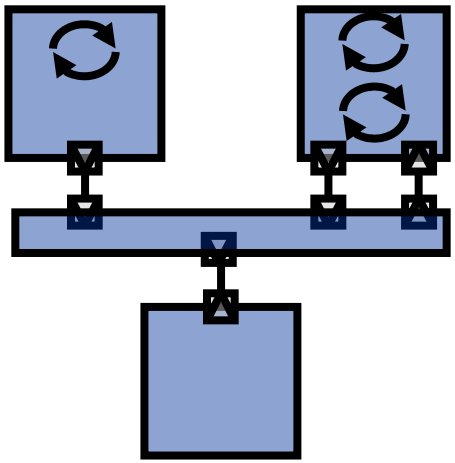
- Fast
- Early available

but:

- Less accurate
- No synthesis

Untimed transaction level models

- Embedded software programmer's view
  - Architecture: modules
  - Behavior: processes
  - Communication:
    - Transactions (inter-modules)
    - Synchronizations (inter-processes)



- Untimed TLM model of computation
  - Concurrent execution of independent processes
  - System synchronization for causal dependencies
- Reference model
  - Functional verification
  - Embedded software development
  - Co-simulation

SystemC

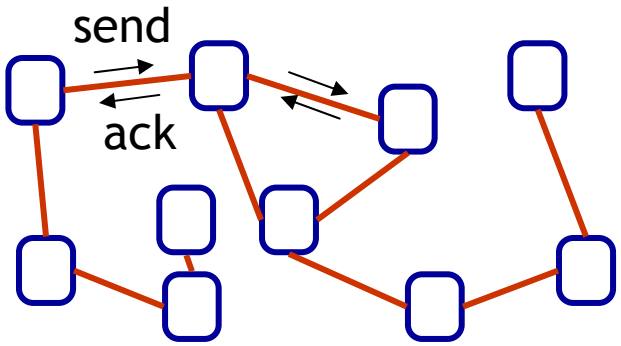
- A C++ library
- Heterogeneous (hard/soft) system modeling
  - *Module* and *port* classes to describe architectures
  - *Threads* and *events* to describe behaviors
- System simulation
  - A global nonpreemptive scheduler
  - A simulated time

Outline

- Transaction level modeling in SystemC
- Verifying SystemC/TLM
- The Lotos/CADP framework for semantics comparison
- Conclusion

Verifying SystemC/TLM models

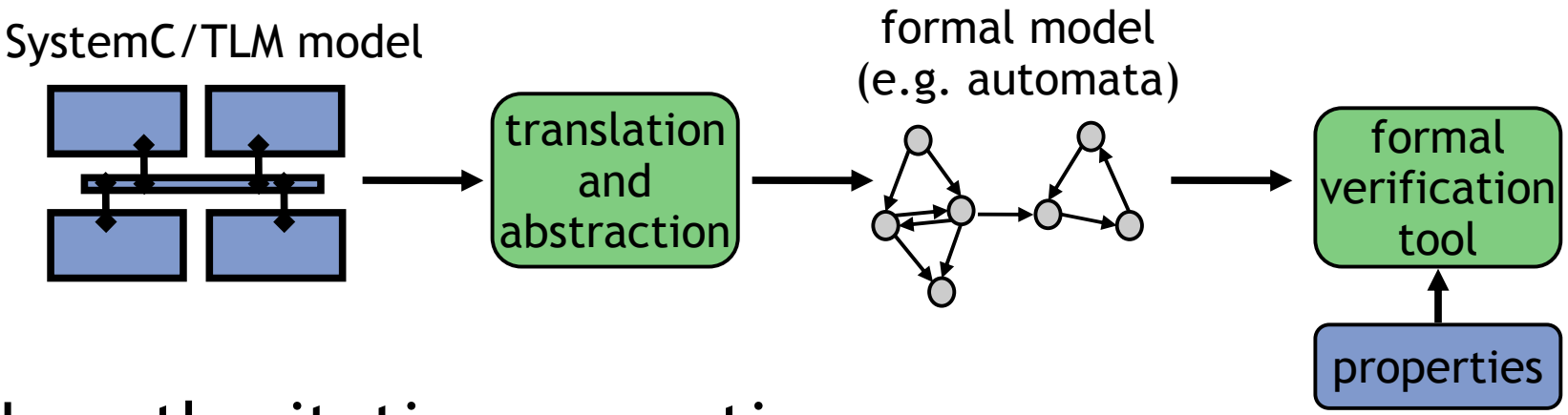
- Main TLM modeling challenge: find all the synchronizations needed between processes
  - Not the software sequential algorithms
  - But the interactions between components



- Verification needs to explore
  - Data space *and*
  - Processes interleaving space

Semantics of SystemC/TLM

- Usual approach for formal verification



- No authoritative semantics
  - SystemC simulation semantics
  - Concurrent TLM semantics

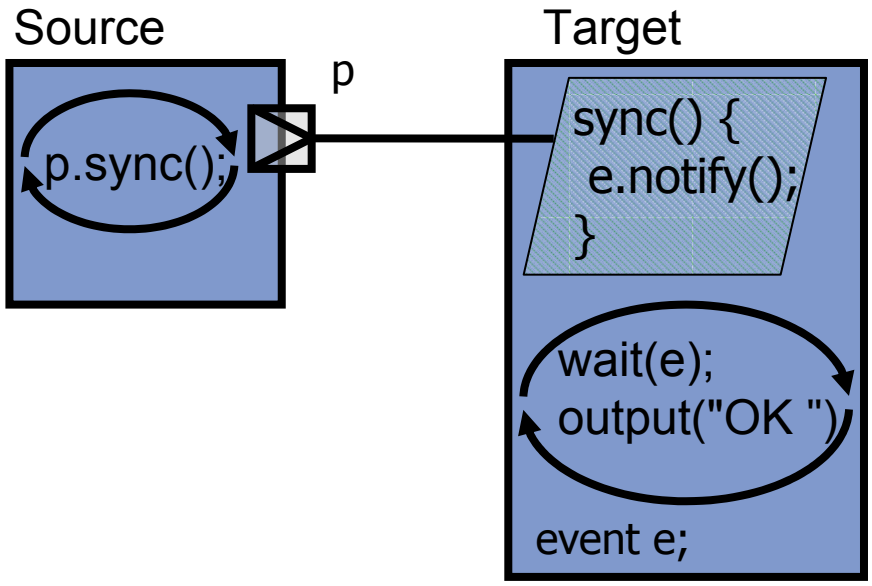
Outline

- Transaction level modeling in SystemC
- Verifying SystemC/TLM
- The Lotos/CADP framework for semantics comparison
- Conclusion

A uniform framework for comparison

- Lotos
  - Process algebra
  - Formal semantics
    - Asynchronous concurrent processes
    - Synchronization and communication by *rendezvous*
- CADP
  - μ-calculus model-checking
  - Equivalence checking
  - Compositional verification

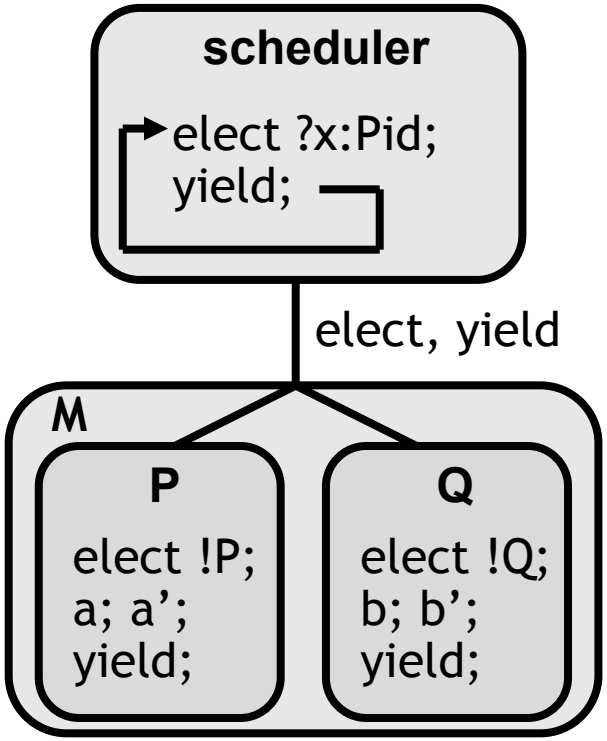
Motivating example 1

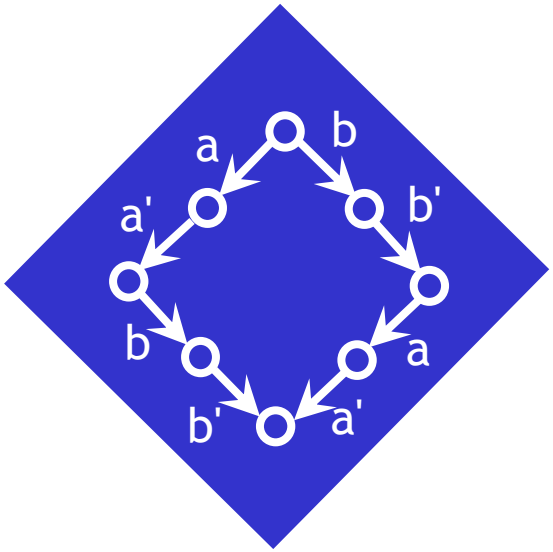


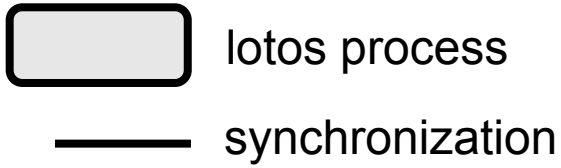
- Incomplete synchronization between Source and Target processes: possible system deadlock

SystemC simulation semantics in Lotos

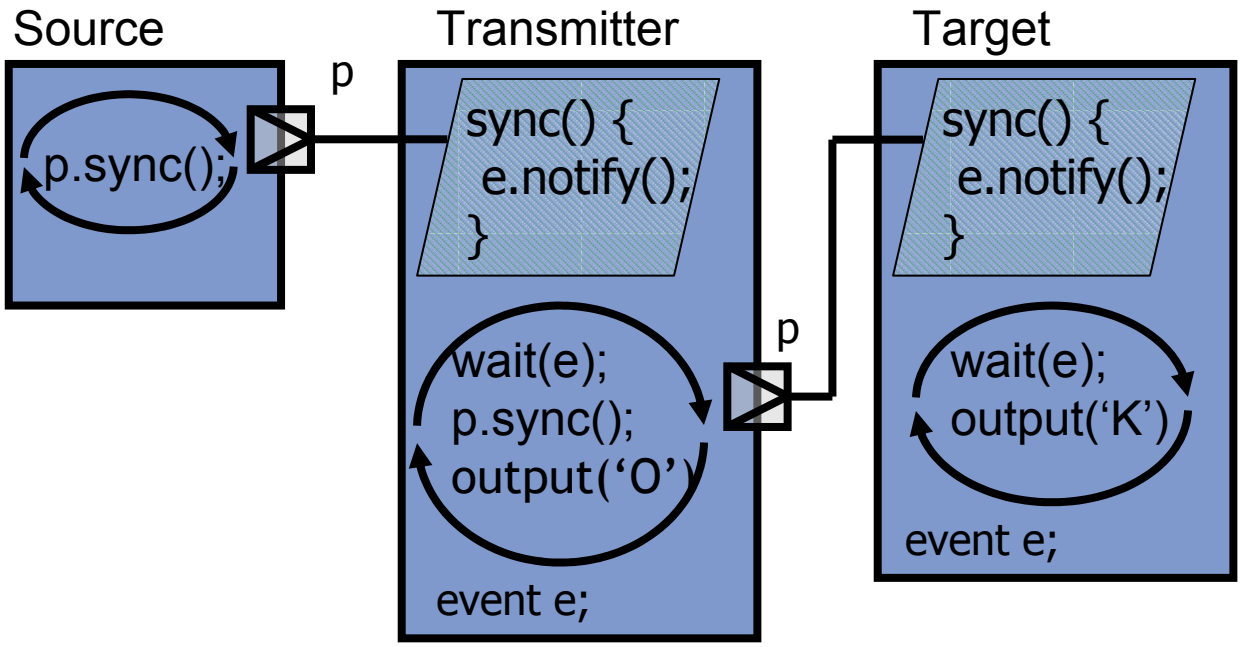
```
SC_MODULE(M) {
  SC_HAS_PROCESS(M);
  SC_CTOR(M) {
    SC_THREAD(P);
    SC_THREAD(Q);
  }
  void P() { a; a' }
  void Q() { b; b' }
};
```







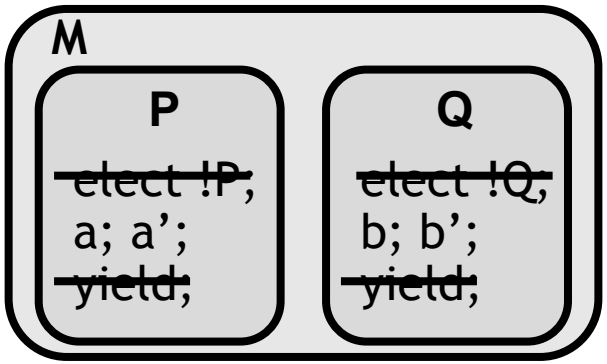
Motivating example 2

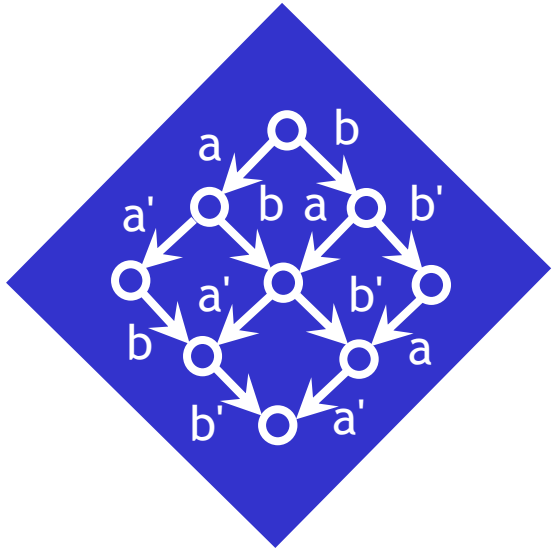


- In a concurrent implementation, "OK" and "KO" are both possible outputs
- With SystemC simulation semantics, the possibly erroneous "KO" output is missed

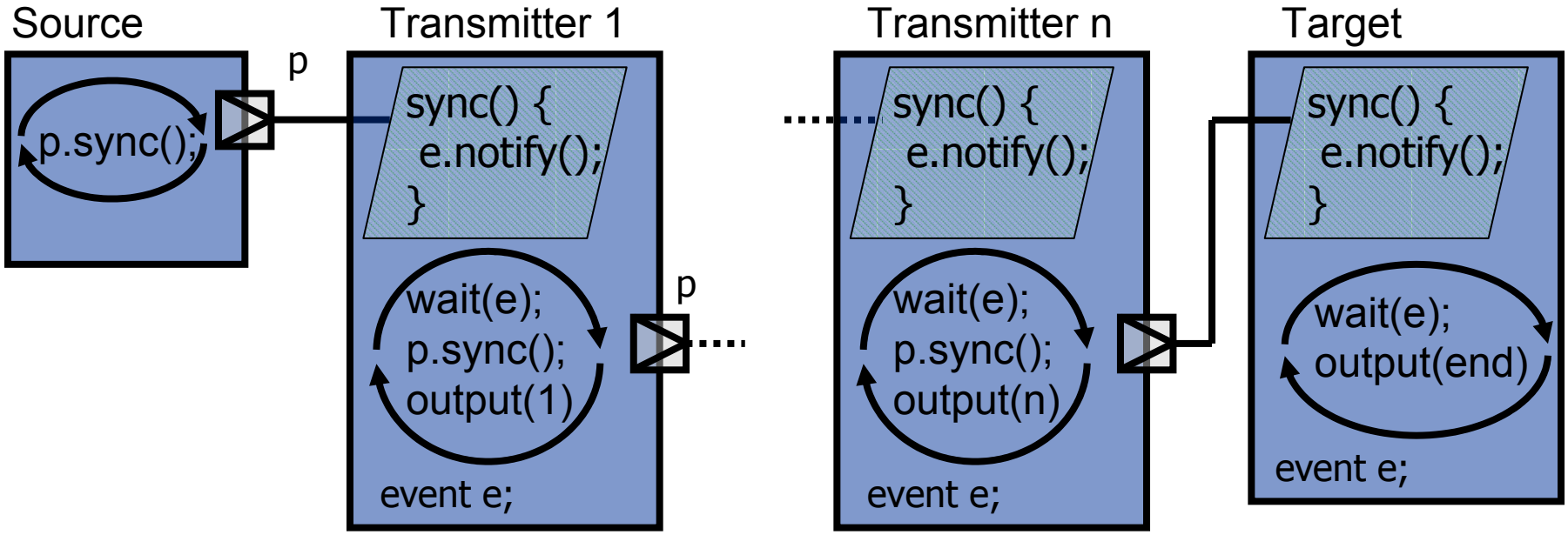
Concurrent TLM semantics in Lotos

```
SC_MODULE(M) {
  SC_HAS_PROCESS(M);
  SC_CTOR(M) {
    SC_THREAD(P);
    SC_THREAD(Q);
  }
  void P() { a; a' }
  void Q() { b; b' }
};
```





Comparison benchmark



Comparison of the two semantics

- Qualitative
  - Concurrent TLM $\supset_{branching}$ SystemC simulation
  - Concurrent TLM $\not\subset_{branching}$ SystemC simulation
  - The concurrent TLM semantics generalizes the SystemC simulation semantics
- Quantitative
  - SystemC simulation generates bigger LTSs although they are strictly included in concurrent TLM models
  - Once minimized, interesting behaviors are indeed less numerous in SystemC simulation models

Conclusion

- Existing SystemC/TLM formal semantics are difficult to compare
- We used Lotos/CADP as a uniform framework to
  - Show concurrent TLM semantics generalizes SystemC simulation semantics
  - Show concurrent TLM semantics scales as well as SystemC simulation semantics
  - Evaluate the performance impact of several transaction encoding variants

Perspectives

- On-going work on an industrial case-study (approx. 26 000 lines of code)
- Automating the translation
- Inverse translation: Lotos into SystemC/TLM

A Comparison of Two SystemC/TLM Semantics for Formal Verification

Olivier Ponsini and Claude Helmstetter

INRIA / VASY

http://www.inrialpes.fr/vasy