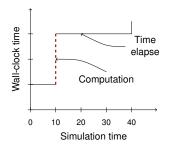
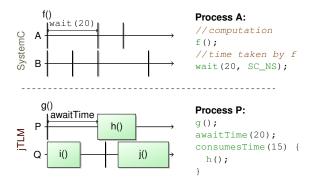
Outline Modeling of Time in Discrete-Event Simulation of Systems-on-Chip Transaction Level Modeling and jTLM Time and Duration in jTLM Giovanni Funchal^{1,2} and Matthieu Moy¹ 3 Applications ¹ Verimag (Grenoble INP) Grenoble, France Implementation ²STMicroelectronics Grenoble, France Conclusion Work partially supported by HELP ANR project MEMOCODE, July 2011 Matthieu Moy (Verimag) Modeling of Time/jTLM MEMOCODE, July 2011 < 1 / 21 > Modeling of Time/jTLM MEMOCODE, July 2011 < 2 / 21 > jTLM jTLM Modern Systems-on-a-Chip Transaction-Level Modeling Software • (Fast) simulation essential in the design-flow ► To write/debug software ← ► To validate architectural choices As reference for hardware verification • Transaction-Level Modeling (TLM): High level of abstraction Hardware Suitable for Industry Standard = SystemC/TLM Modeling of Time/jTLM MEMOCODE, July 2011 < 4 / 21 > Modeling of Time/jTLM MEMOCODE, July 2011 < 5 / 21 > jTLM jTLM SystemC/TLM vs. "TLM Abstraction Level" jTLM: goals and peculiarities SystemC **TLM** jTLM • jTLM's goal: define "TLM" independently of SystemC Cycle Not cooperative (true parallelism) Parallelism this talk accurate Not C++ (Java)No δ-cycle Clocks Function Interesting features RTL calls Small and simple code (≈ 500 LOC) Nice experimentation platform Coroutine semantics Not meant for production Gale lev | δ-cycle Matthieu Moy (Verimag) Modeling of Time/jTLM MEMOCODE, July 2011 < 6 / 21 > Matthieu Moy (Verimag) Modeling of Time/jTLM MEMOCODE, July 2011 < 7 / 21 > Simulation Time Vs Wall-Clock Time Time in SystemC and jTLM





Modeling of Time/jTLM

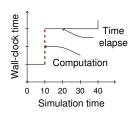
MEMOCODE, July 2011 < 10 / 21 >

Matthieu Moy (Verimag)

jTLM

Time à la SystemC: awaitTime (T)

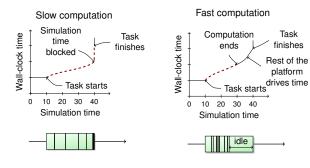
- By default, time does not pass ⇒ instantaneous tasks
- awaitTime(T): let other processes execute for T time units



f(); // instantaneous awaitTime(20);

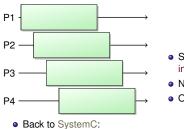
Matthieu Moy (Verimag) Modeling of Time/jTLM MEMOCODE, July 2011 < 11 / 21 > Duration

Execution of consumesTime (T)



Modeling of Time/iTLM MEMOCODE, July 2011 < 13 / 21 >

Parallelization



¡TLM's Semantics

- Simultaneous tasks run in parallel
- Non-simultaneous tasks don't

MEMOCODE, July 2011 < 19 / 21 >

- Overlapping tasks do

Matthieu Moy (Verimag)

- ▶ Parallelizing within δ -cycle = great if you have clocks Simulation time is the bottleneck with quantitative/fuzzy time

Matthieu Moy (Verimag) Modeling of Time/jTLM MEMOCODE, July 2011 < 16 / 21 >

Time Queue and consumesTime (T) Current Curren Current instant instant instant e(30) P, Q, Rime Elapse Time Elapse Q R Process P: Process Q: Process R: ▶f(); ▶i(); ►consumesTime(50){ ▶awaitTime(90); g() ▶consumesTime(30){ ►h(); k();

Modeling of Time/jTLM

Task with Known Duration: consumesTime (T)

Semantics:

jTLM

- Start and end dates known
- Actions contained in task spread in between
- Advantages:
 - Model closer to actual system
 - Less bugs hidden
 - Better parallelization

```
consumesTime(15)
    f1();
    f2();
    f3();
consumesTime(10) {
    g();
```

MEMOCODE, July 2011 < 15 / 21 >

Matthieu Moy (Verimag) Modeling of Time/jTLM MEMOCODE, July 2011 < 12 / 21 >

Exposing Bugs

Example bug: mis-placed synchronization:

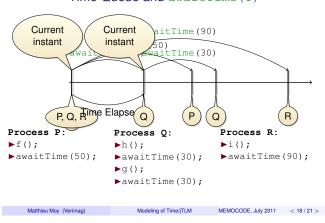
```
flag = true;
                               while(!flag)
         awaitTime(5);
                                   awaitTime(1);
                            writeIMG();
                               awaitTime(10);
         awaitTime(10):
                               readIMG():
\Rightarrow bug never seen in simulation
```

consumesTime(15) { while (!flag) awaitTime(1); flag = true; writeIMG(); awaitTime(10); readIMG();

⇒ strictly more behaviors, including the buggy one

Modeling of Time/jTLM





Perspectives

Summary

jTLM

- Tasks with duration
- Exhibit more behaviors/bugs
- Better parallelization
- Skipped from the talk (cf. paper)
 - Tasks with a priori unknown duration
 - ▶ jTLM's cooperative mode
- Perspectives
 - Adapt the ideas to SystemC (ongoing, not so hard)
 Run-time Verification to explore schedules (science-fiction)

 - Open-Source Release?

Thank you! → Questions?