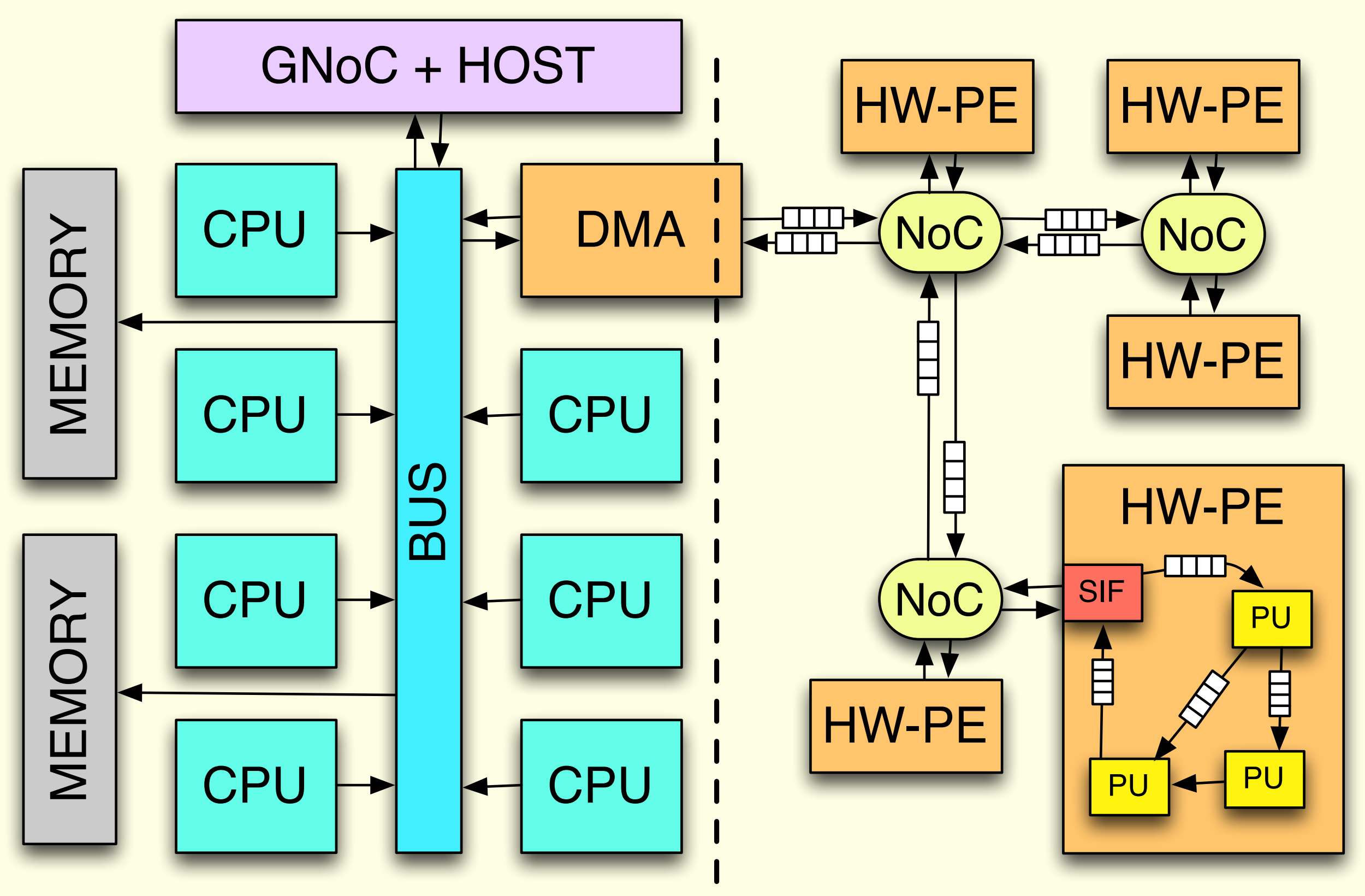
Fast and Accurate TLM Simulations using Temporal Decoupling for FIFO-based Communications

Claude Helmstetter[1,4] <cl@ude.fr>, Jérôme Cornet[2], Bruno Galilée[2], Matthieu Moy[3], Pascal Vivet[1]

[1]CEA - LETI - Minatec Campus, [2]STMicrolelectronics - Grenoble, [3]Grenoble-INP/Verimag, [4]CNRS/Verimag

Introduction

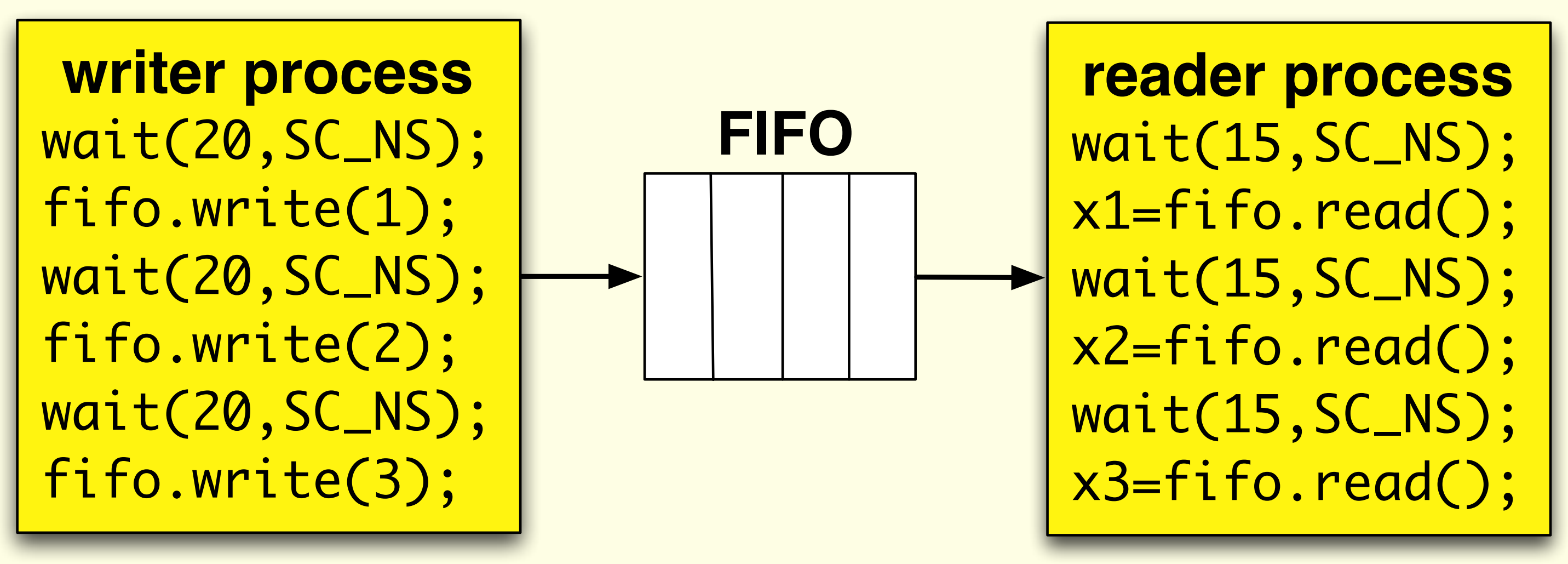
context: P2012/STHORM heterogeneous SoC



Memory-based transactions
→ use **temporal decoupling**
with respect to IEEE 1666-2011

FIFO-based communications
→ use **temporal decoupling**
as presented in this work

issue: too many context switches in FIFO model

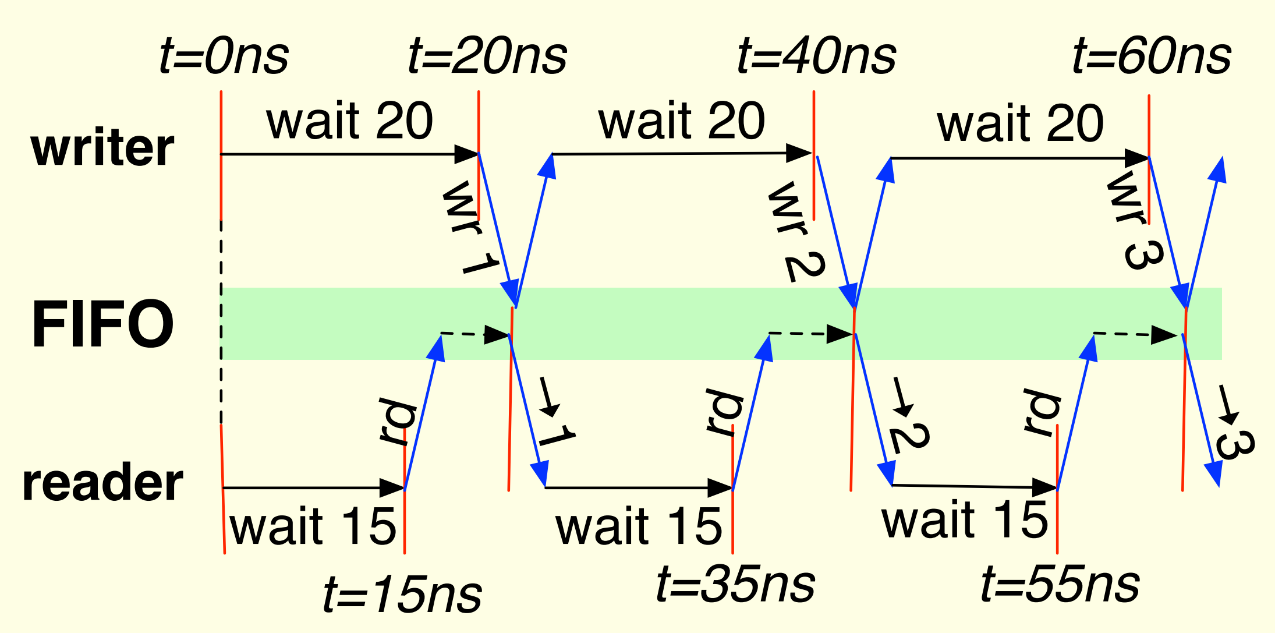


writer process

```
wait(20,SC_NS);
fifo.write(1);
wait(20,SC_NS);
fifo.write(2);
wait(20,SC_NS);
fifo.write(3);
```

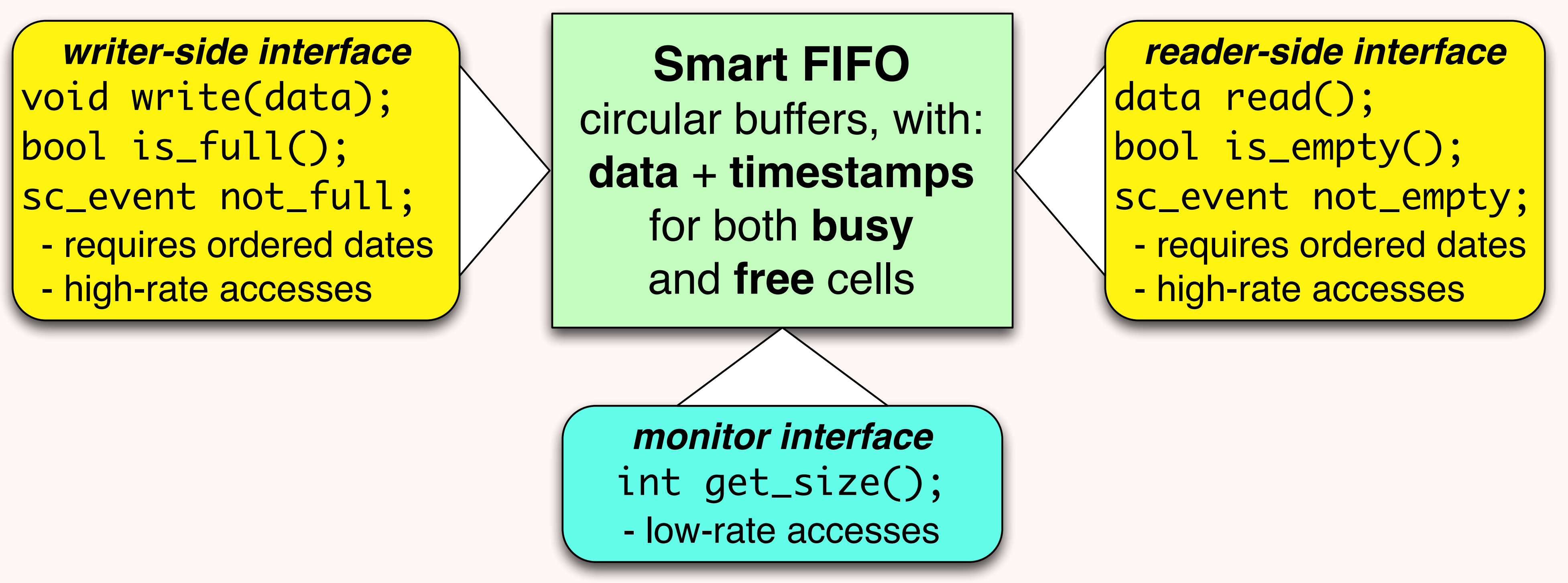
reader process

```
wait(15,SC_NS);
x1=fifo.read();
wait(15,SC_NS);
x2=fifo.read();
wait(15,SC_NS);
x3=fifo.read();
```



The Smart FIFO

main idea: Develop a FIFO model that use timestamps to set local dates and limit context switches



writer-side interface

```
void write(data);
bool is_full();
sc_event not_full;
```

- requires ordered dates
- high-rate accesses

Smart FIFO
circular buffers, with:
data + timestamps
for both **busy**
and **free** cells

reader-side interface

```
data read();
bool is_empty();
sc_event not_empty;
```

- requires ordered dates
- high-rate accesses

monitor interface

```
int get_size();
```

- low-rate accesses

other approaches:

- tlm_fifo (from OSCI TLM 1.0): no timestamp ⇒ wrong behavior if used with temporal decoupling
- sc_event_queue (SystemC): timestamps, but no size control
- loose timing accuracy: some stream protocols are faster but introduce more or less timing errors

Algorithm of the `write` method

1. if all cells are busy, synchronize the writer process and wait until a cell is available (1 context switch)
2. if the first free cell freeing date is in the future, then increase the writer process local time up to this date
3. update the cell: fill the data and set the insertion date; advance the first free cell index
4. wake up a blocked reader process, if any.

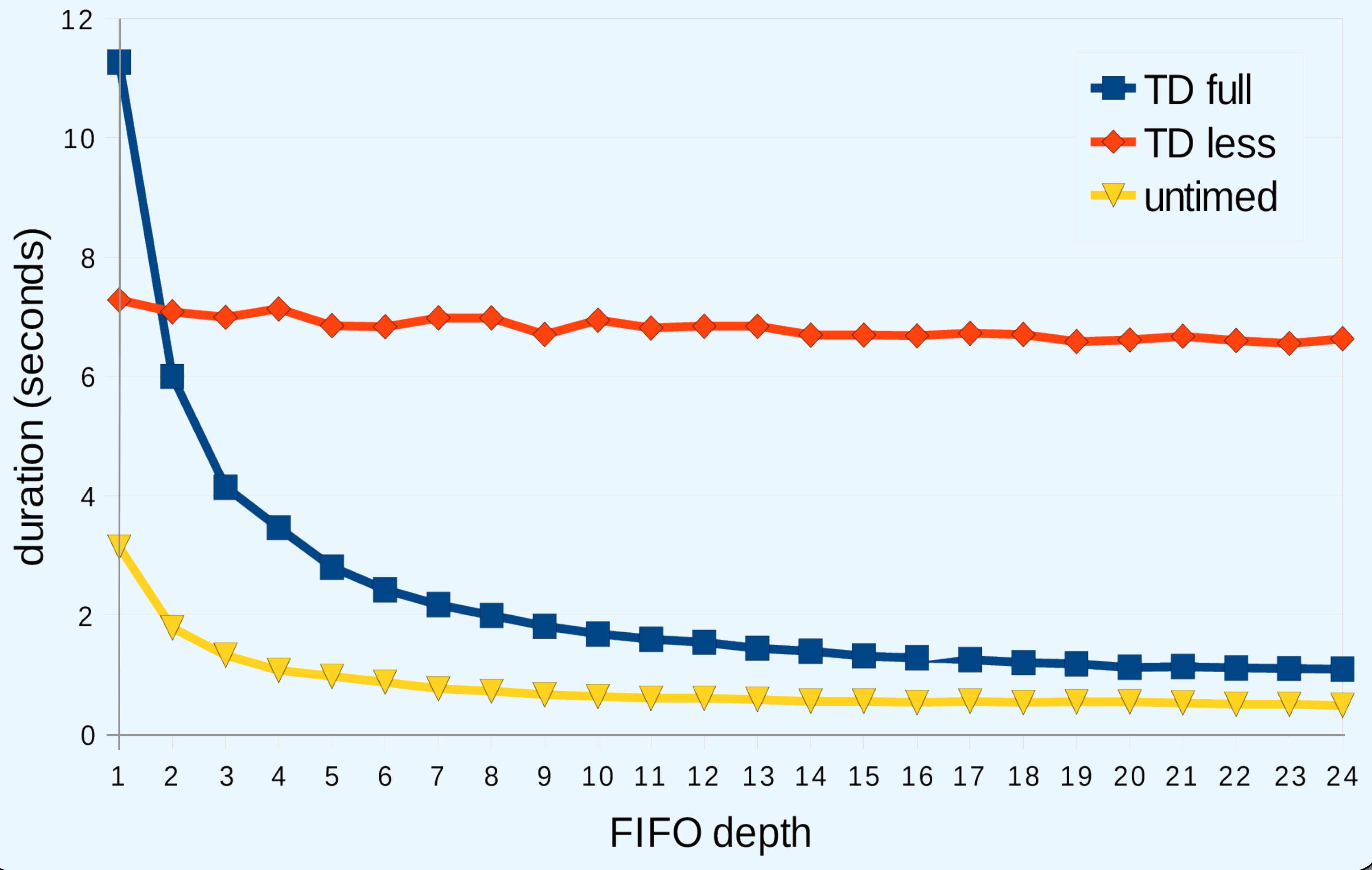
Algorithm of the `is_empty` method

returns true if and only if:
1. either all cells are (internally) free
2. or the insertion date of the first busy cell is in the future.

Algorithm of the `get_size` method

not so simple... see the paper.

Simulation durations



Conclusion

Using the Smart FIFO:
- As **few context switches** as there are in an untimed model
- Up to **6 times faster** than a basic FIFO
- **Timing** perfectly **preserved** (excepting delta-cylces and scheduling)
- No need of a time quantum

Case study: P2012/STHORM TLM model
- Successful and seamless integration
- Behavior and timing preserved
- Simulation speed: **+ 42.3 %**

Demo available on the laptop