Embedded Real-Time Systems

Reinhard von Hanxleden

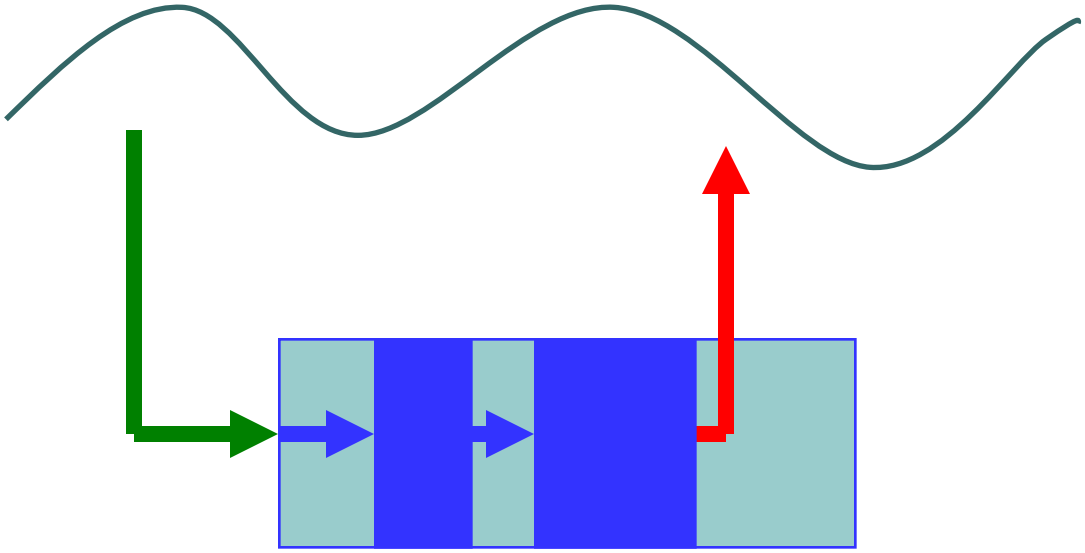
Christian-Albrechts-Universität zu Kiel

Lecture 6c: Time-Triggered Models

The 5-Minute Review Session

1. What is a *firing function*?
2. What is a *constructive* state machine? How does constructiveness relate to well-formedness?
3. How do actors communicate in a *dataflow* model?
4. What is *synchronous* dataflow? *Homogeneous* dataflow? *Dynamic* dataflow?
5. What is a necessary condition for a *bounded memory infinite execution* of a dataflow model? Is it sufficient?

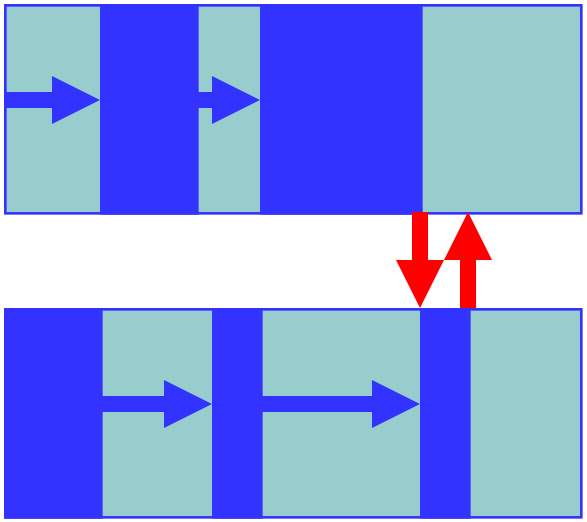
Standard Practice



make output available
as soon as ready

Slide adapted from Tom Henzinger

Standard Practice

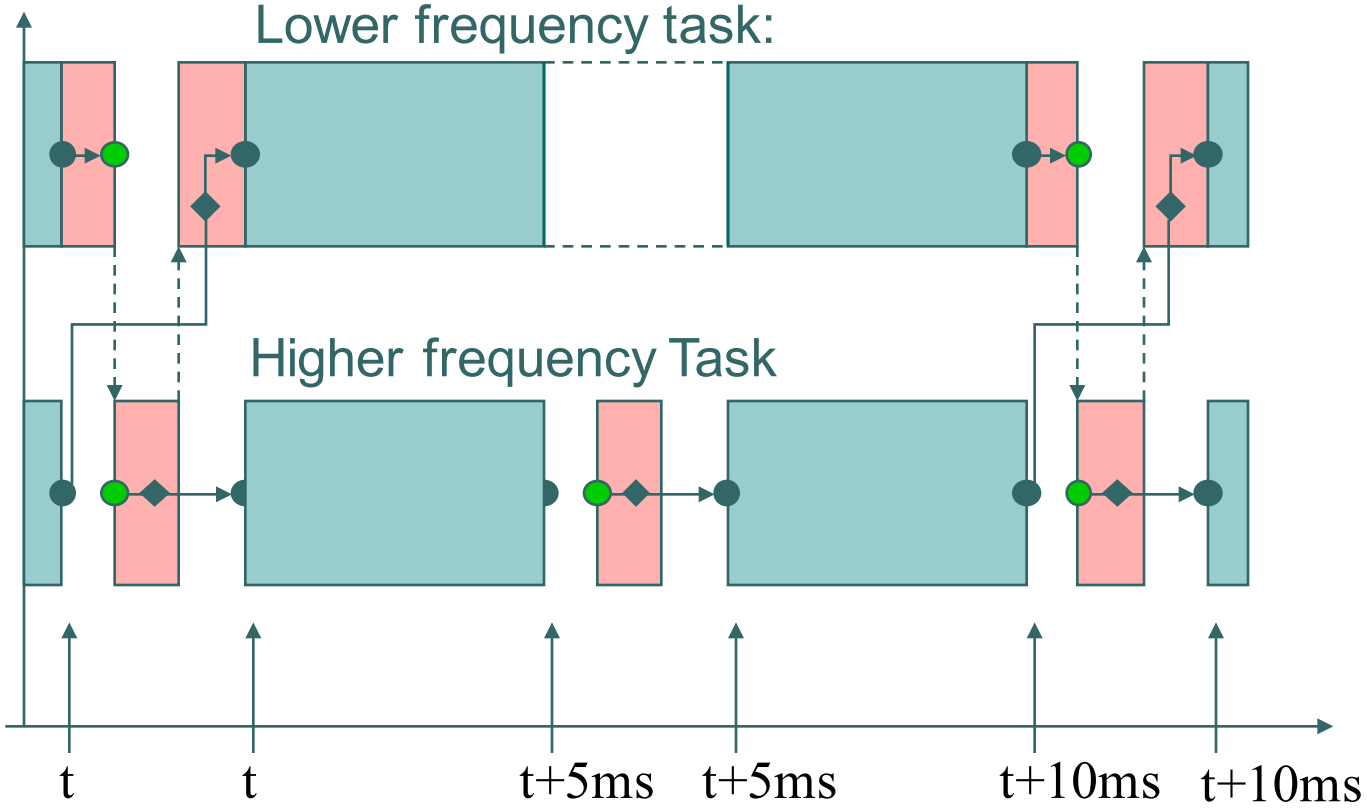


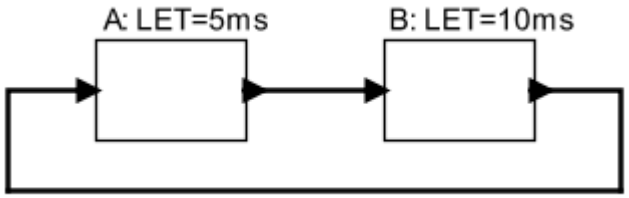
data race

Slide adapted from Tom Henzinger

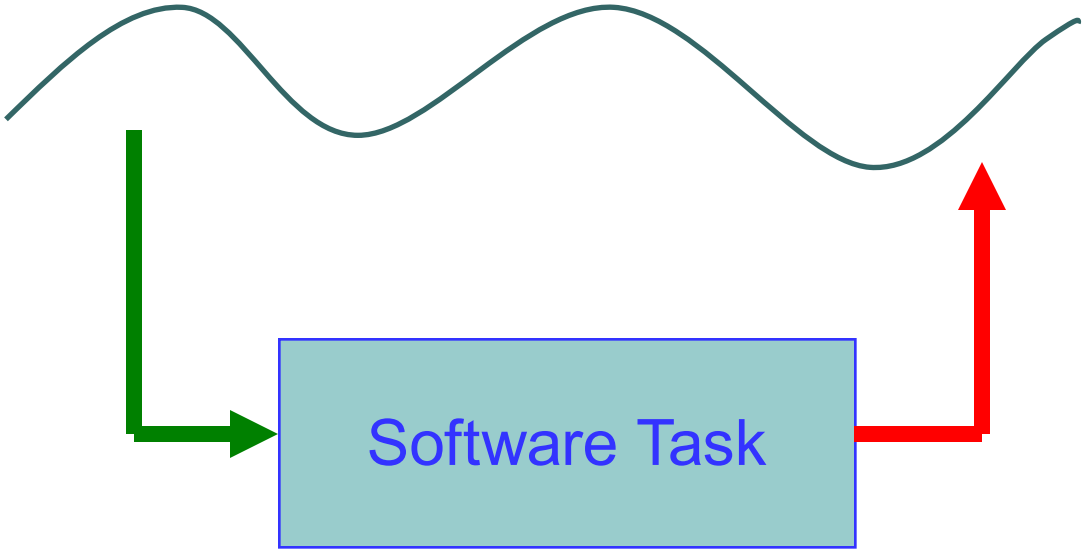
Example: Time-Triggered Models with LET

In some time-triggered models (e.g. Giotto, TDL), each actor has a **logical execution time** (LET). Its actual execution time always appears to have taken the time of the LET.





The LET (Logical Execution Time) Programming Model



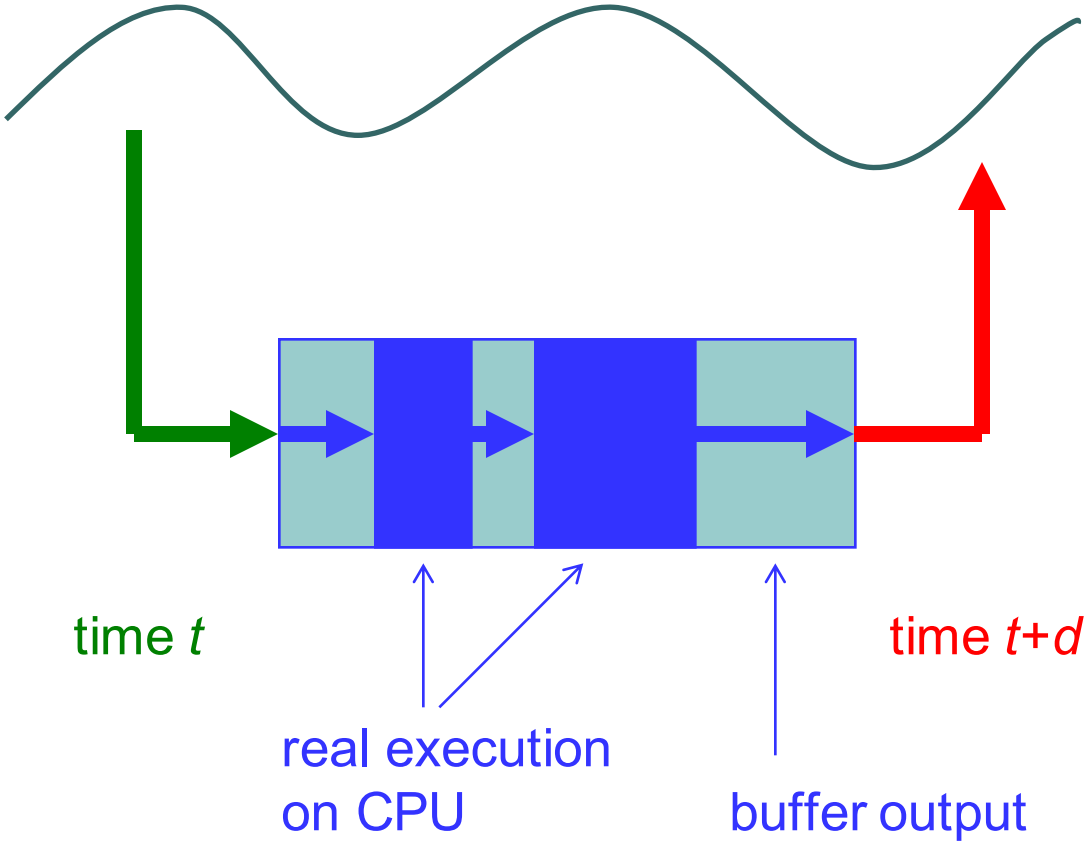
read sensor
input at time *t*

write actuator
output at time *t*+*d*,
for specified *d*

Examples: Giotto, TDL,

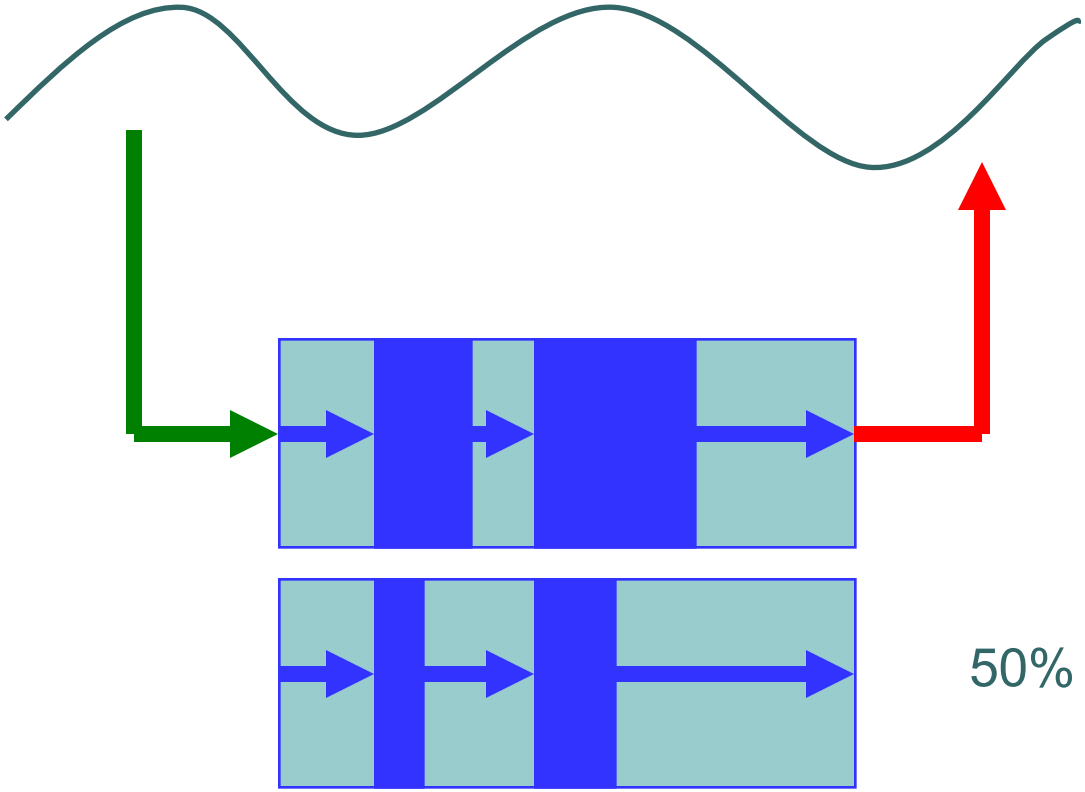
Slide from Tom Henzinger

The LET (Logical Execution Time) Programming Model



Slide from Tom Henzinger

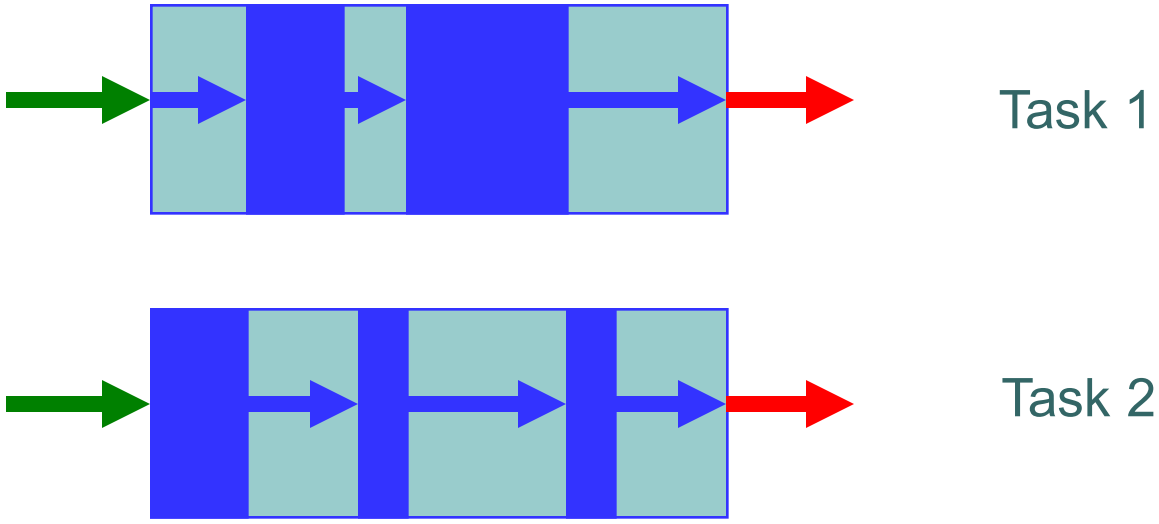
Portability



50% CPU speedup

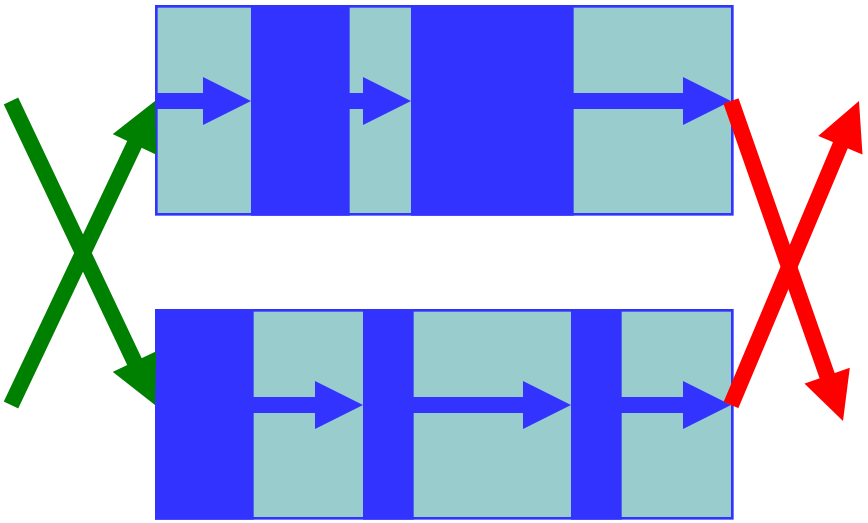
Slide from Tom Henzinger

Composability



Slide from Tom Henzinger

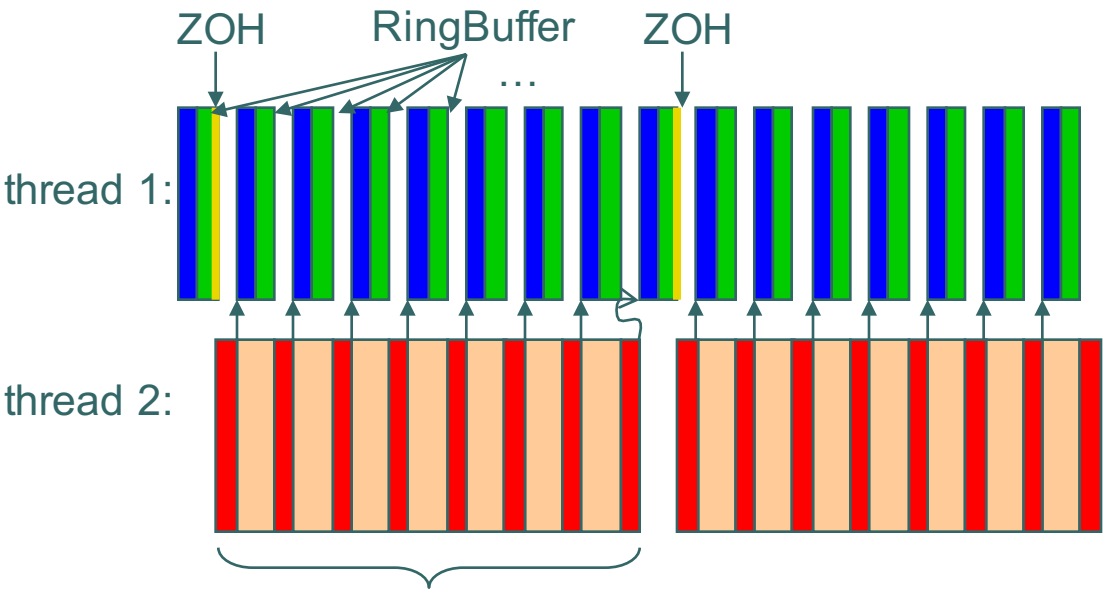
Determinism

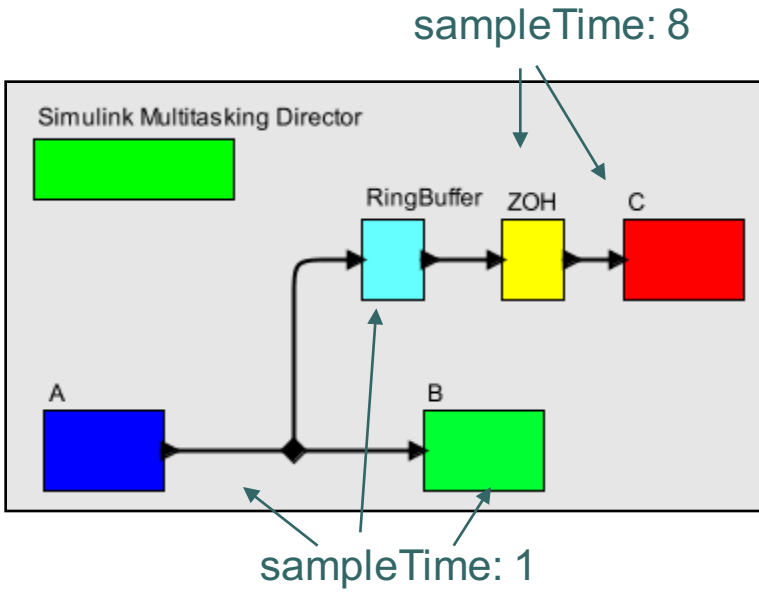


Timing predictability: minimal jitter
Function predictability: no race conditions

Slide from Tom Henzinger

Recall: Simulink Strategy

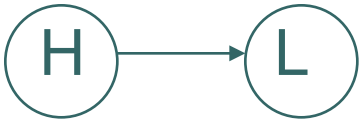




Problem: in a naïve implementation, ZOH would copy the entire buffer. Copying large amounts of data can take a long time.

Solution: alternate buffers where A writes to, using clever (and careful) pointer managing.

The high-to-low-priority protocol [Caspi et al, 2008]



- L keeps a double buffer: `B[0,1]`
- Two (boolean) pointers: `current, next`
- H writes to: `B[next]`
- L reads from: `B[current]`

- When L arrives: `current := next`
- When H arrives:
```
if (current = next) then
    next := not next
```

- Initially: `current=next=0, B[0]= B[1]= default`

No copying of buffers. Works for arbitrary arrival patterns..

Reading

P. Caspi, N. Scaife, C. Sofronis and S.Tripakis. *Semantics-Preserving Multitask Implementation of Synchronous Programs*. In *ACM Trans. Embedded Computing Systems*. ACM, 2008.

To Go Further

Thomas A. Henzinger, Benjamin Horowitz, and Christoph M. Kirsch

Giotto: A Time-triggered Language for Embedded Programming

Proceedings of the IEEE 91:84-99, 2003