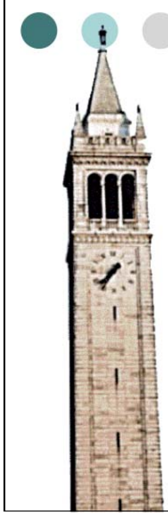


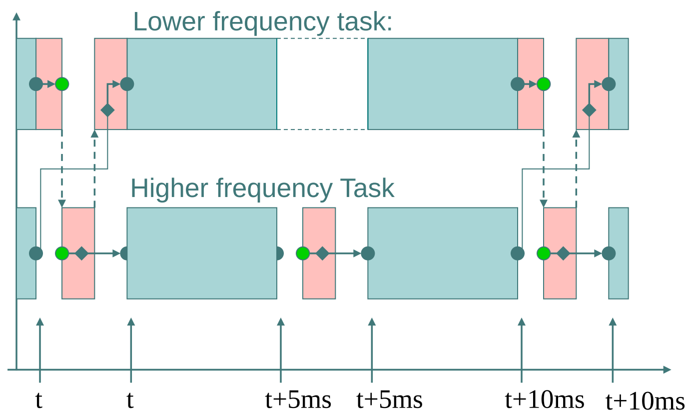
Introduction to Embedded Systems

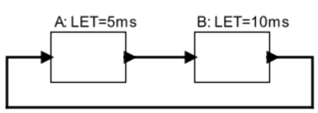
Edward A. Lee & Sanjit A. Seshia
UC Berkeley
EECS 149
Spring 2010

Lecture 20: Time-Triggered Models

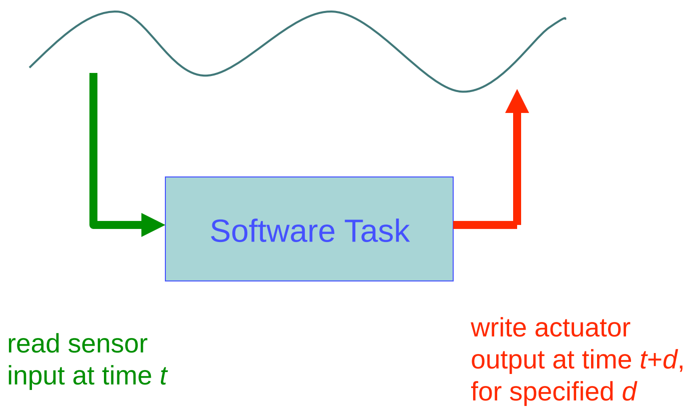
Example: Time-Triggered Models with LET

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

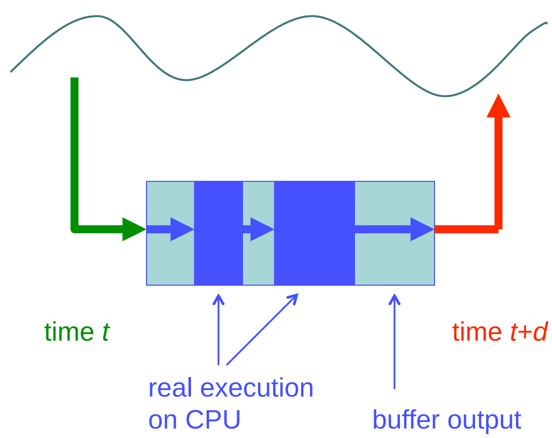


Examples: Giotto, TDL,

Slide from Tom Henzinger

EECS 149, UC Berkeley: 3

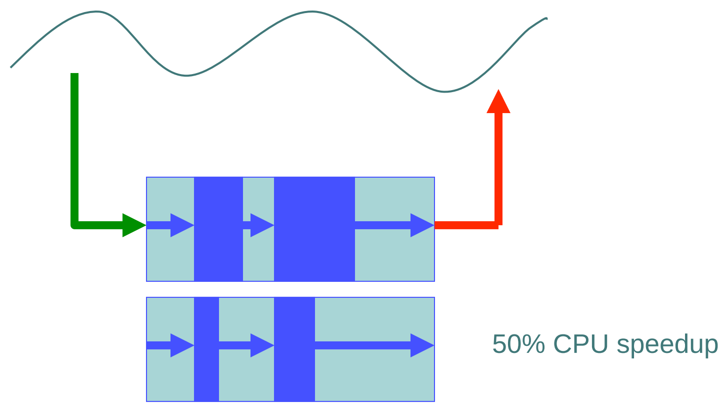
The LET (Logical Execution Time) Programming Model



Slide from Tom Henzinger

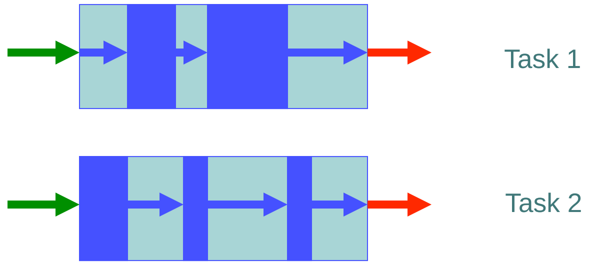
EECS 149, UC Berkeley: 4

Portability



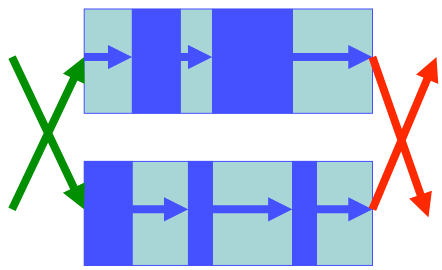
Slide from Tom Henzinger

Composability



Slide from Tom Henzinger

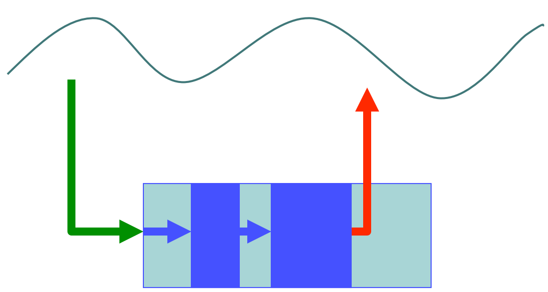
Determinism



Timing predictability: minimal jitter
Function predictability: no race conditions

Slide from Tom Henzinger

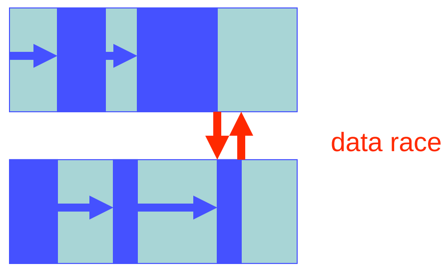
Contrast LET with Standard Practice



make output available
as soon as ready

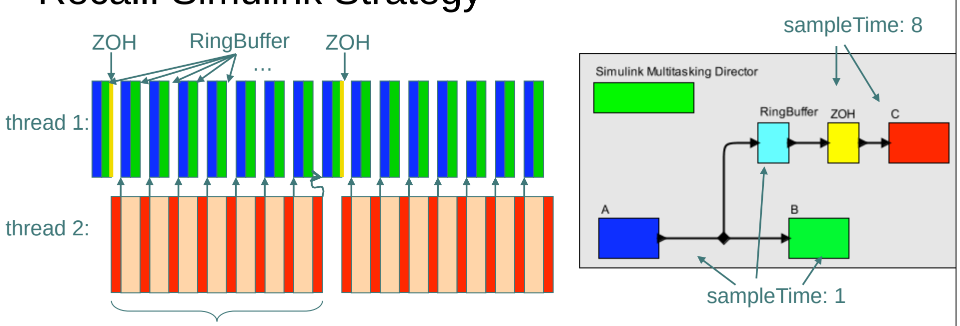
Slide from Tom Henzinger

Contrast LET with Standard Practice



Slide from Tom Henzinger

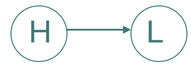
Recall: Simulink Strategy



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.
See Reading paper for details.

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.