

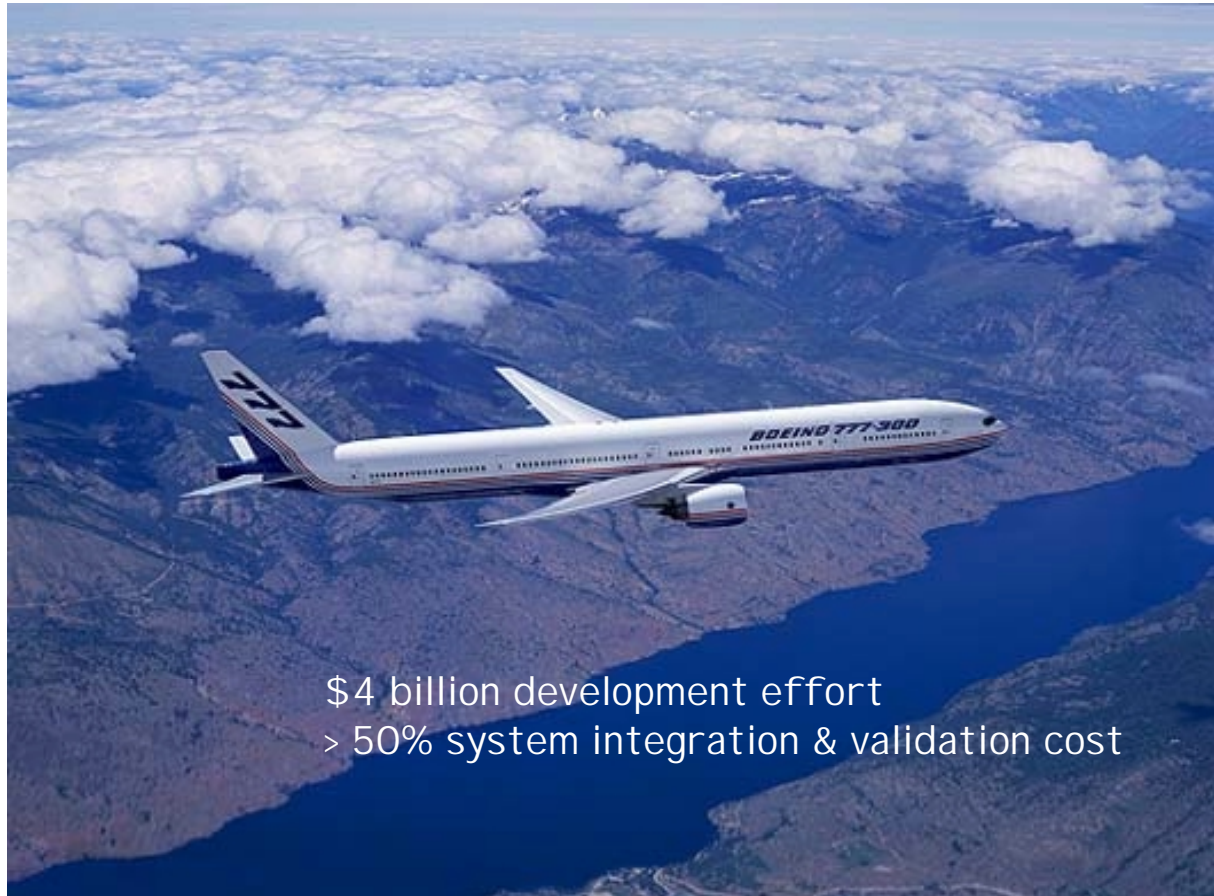
Embedded Software Engineering

3 Unit Course, Spring 2002
EECS Department, UC Berkeley

Christoph Kirsch

www.eecs.berkeley.edu/~fresco/giotto/course-2002

It's significant



\$4 billion development effort
> 50% system integration & validation cost

It's tricky

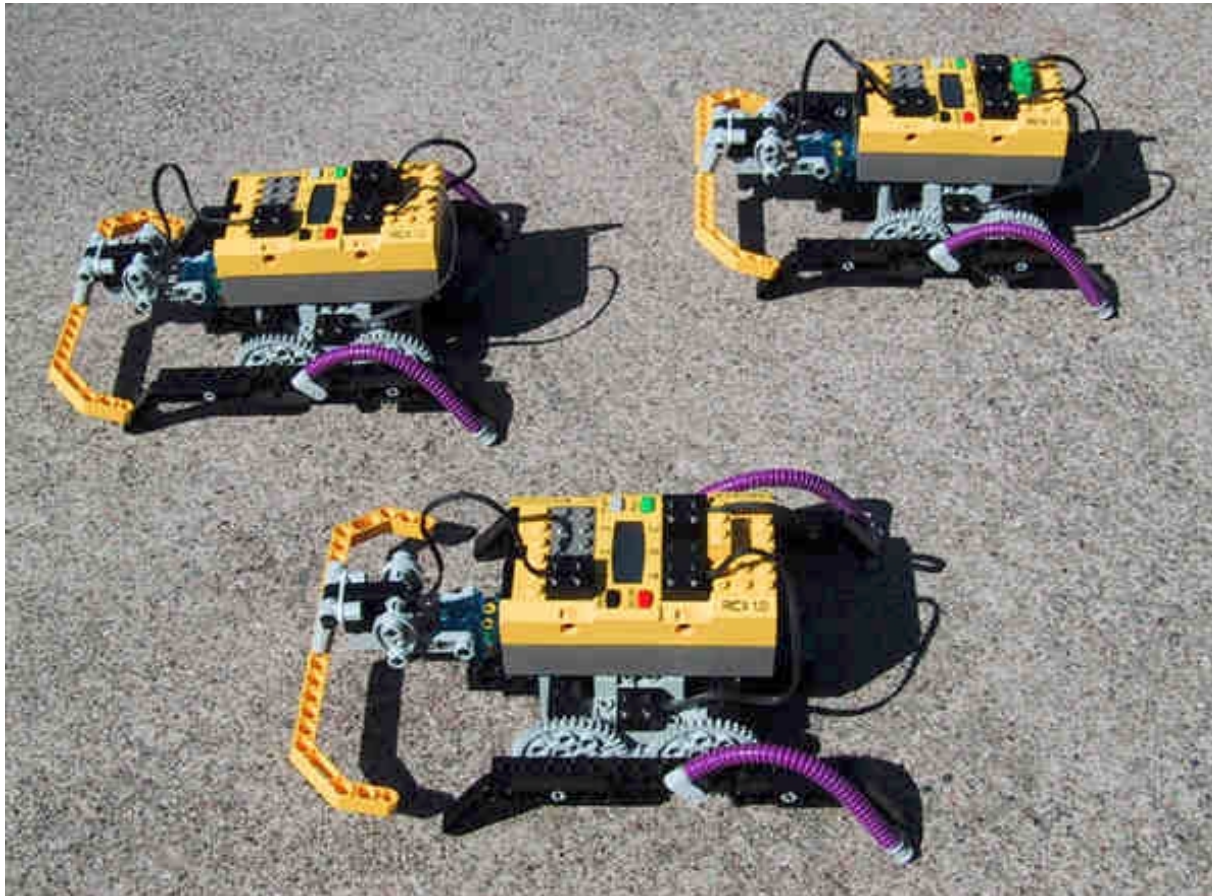


It's risky

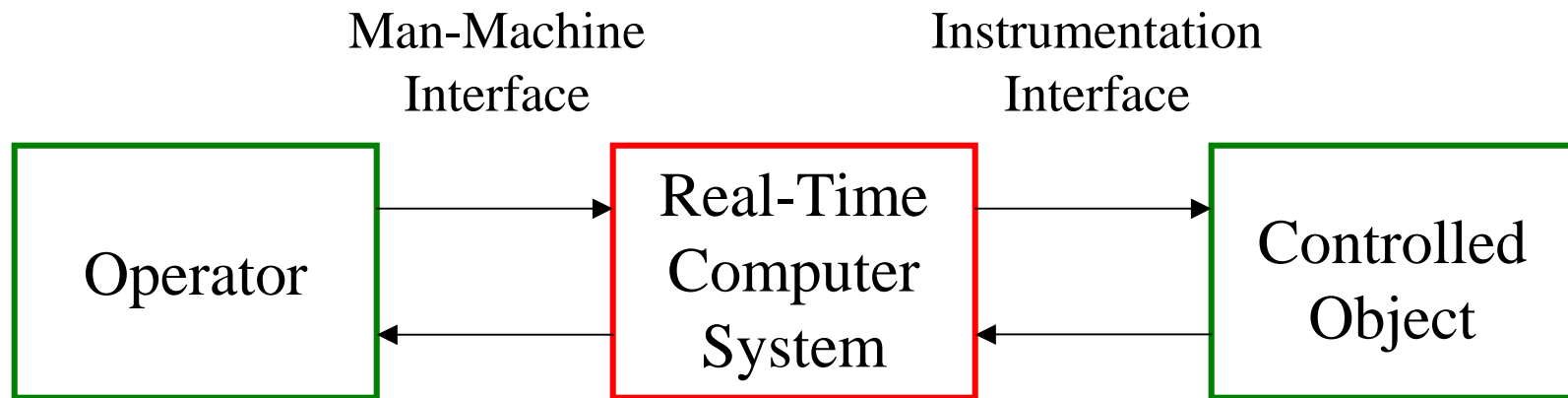


French Guyana, June 4, 1996
\$800 million embedded software failure

It's fun



Problem

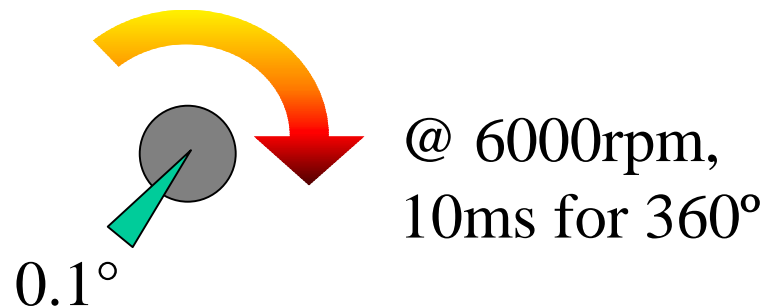


Kopetz97

Methodologies for the implementation of
embedded **real-time** applications

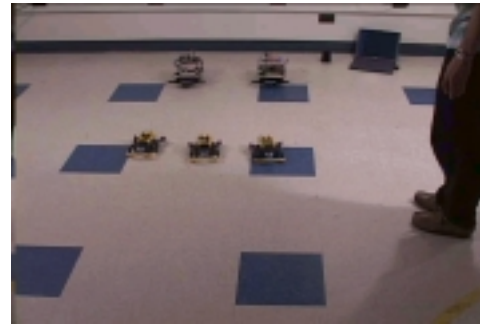
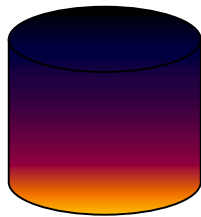
- Methodology: **tool-supported**, **abstract**, **compositional**
- Implementation: **compositional**, **scalable**, **dependable**

Engine Controller



- Temporal accuracy of $3\mu\text{sec}$
- Up to 100 concurrent software tasks
- Hard real-time: no missed deadlines

Video Streaming



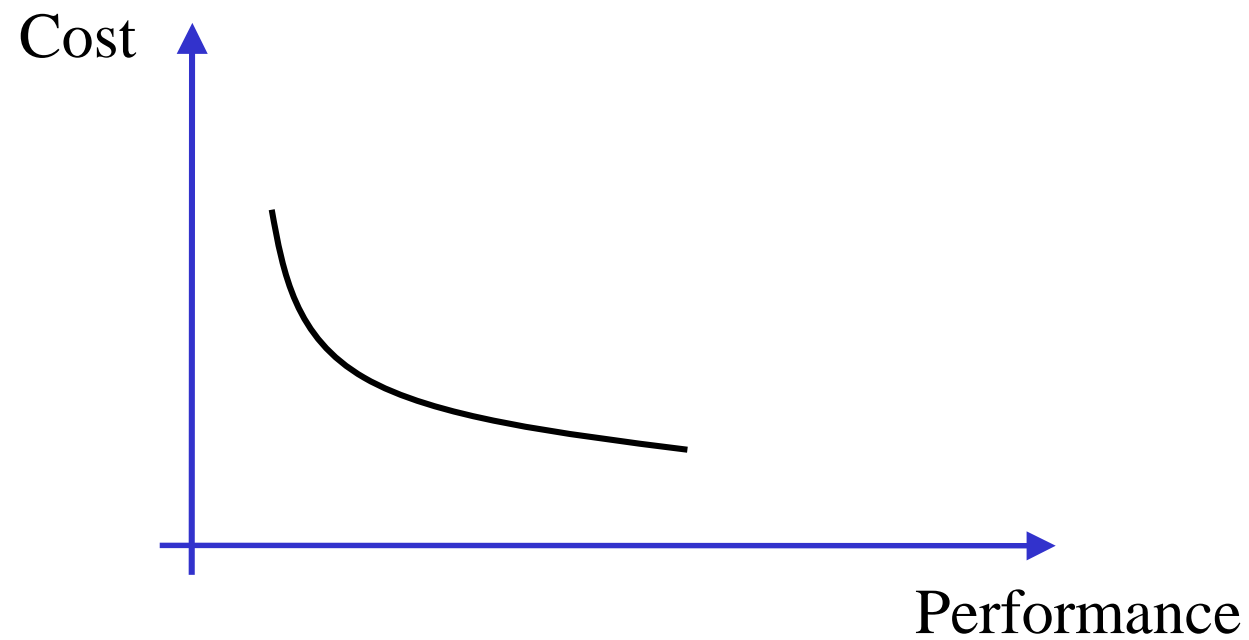
- 25 frames/sec
- Dynamic resource allocation
- Soft real-time: degraded QoS

Real-Time Systems

Characteristics	Hard	Soft
Response time	Hard-required	Soft-desired
Peak-load performance	Predictable	Degraded
Control of pace	Environment	Computer
Redundancy	Active	Checkpoint
Error detection	Autonomous	User assisted

Kopetz97

Microcontroller Market



Mechatronics

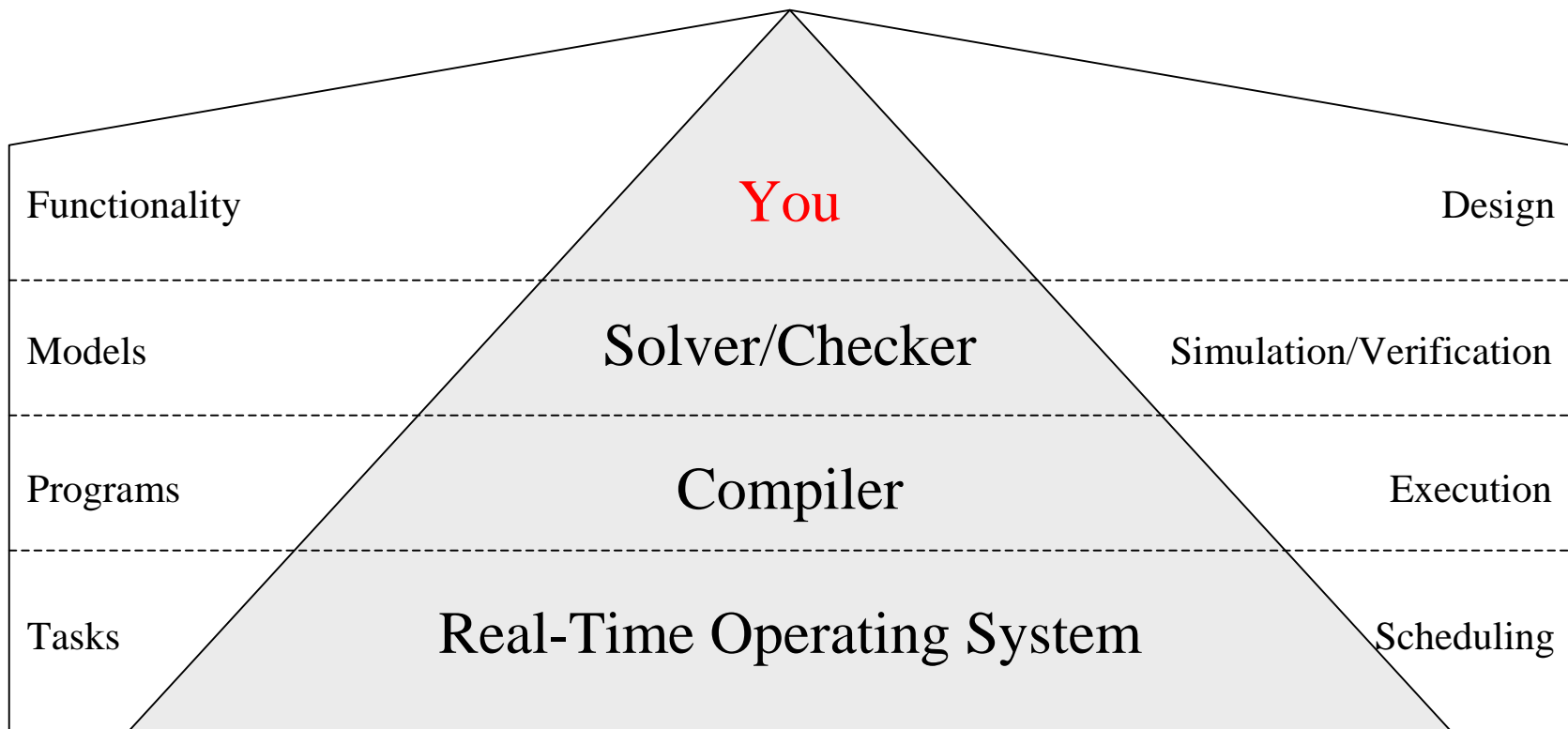


Fly-by-wire

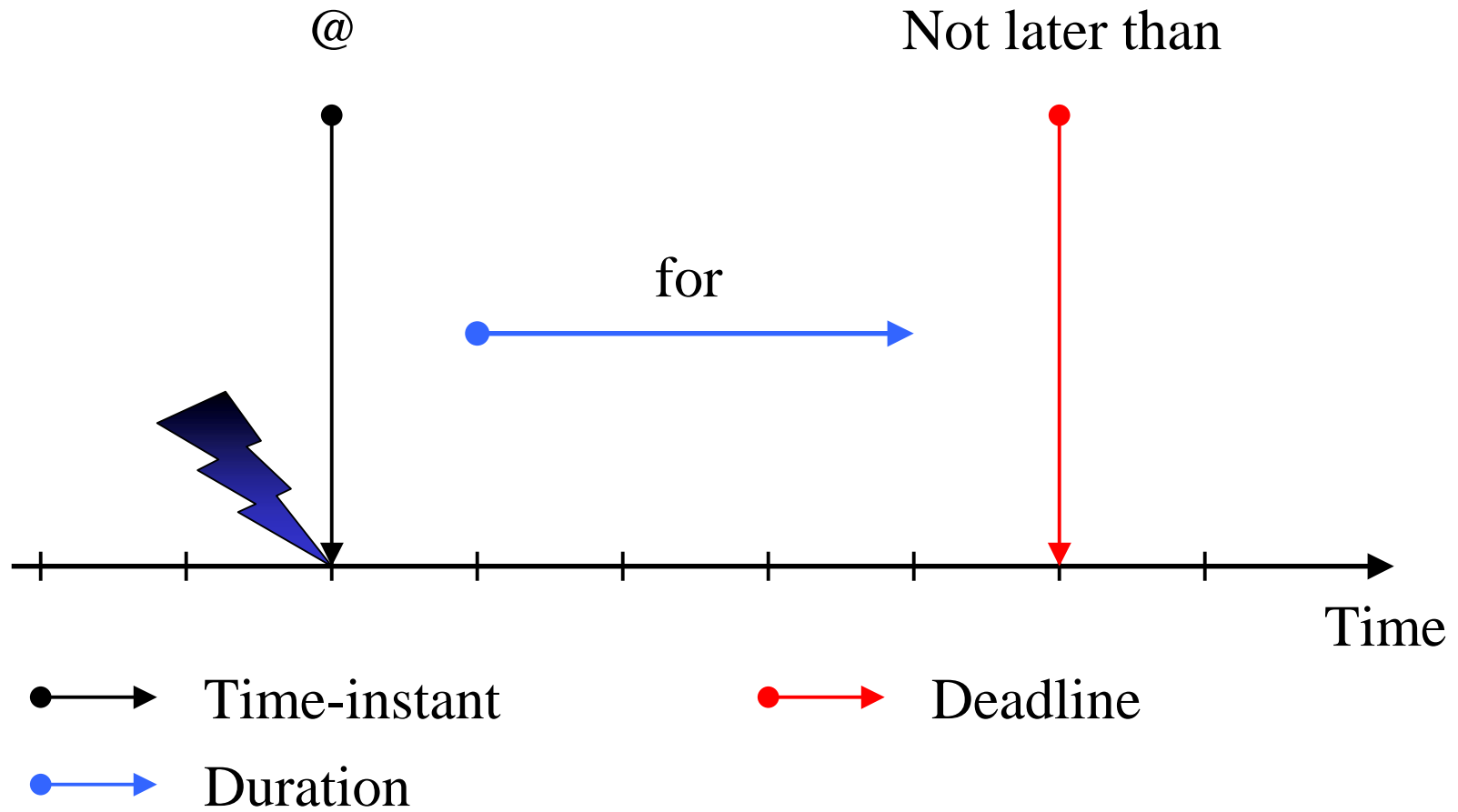


Drive-by-wire

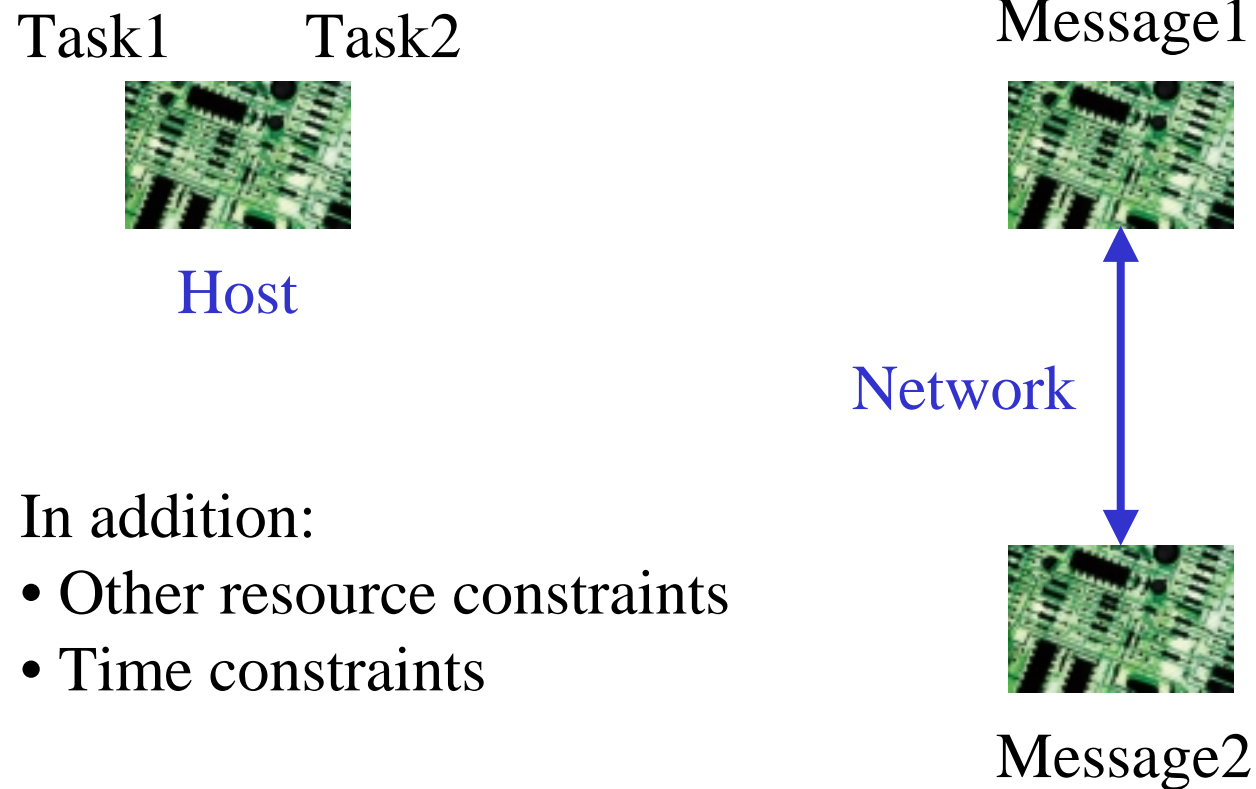
Embedded Software Engineering



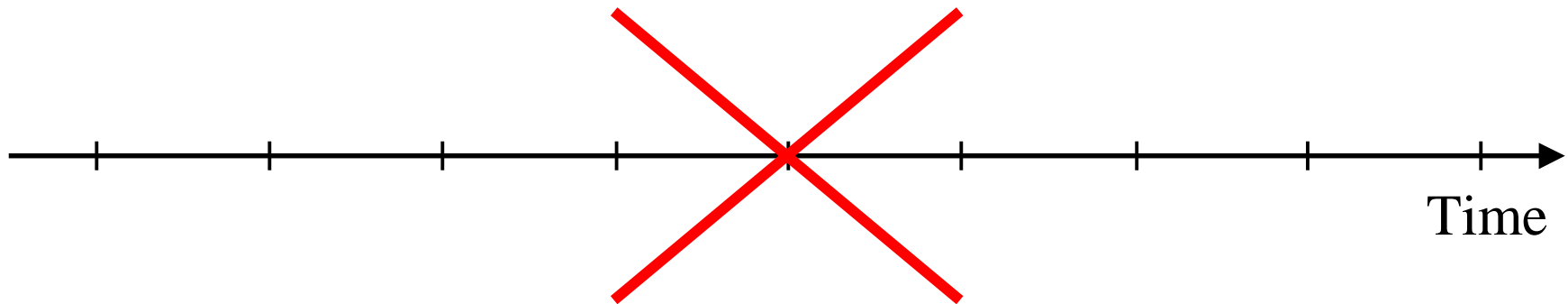
Real-Time



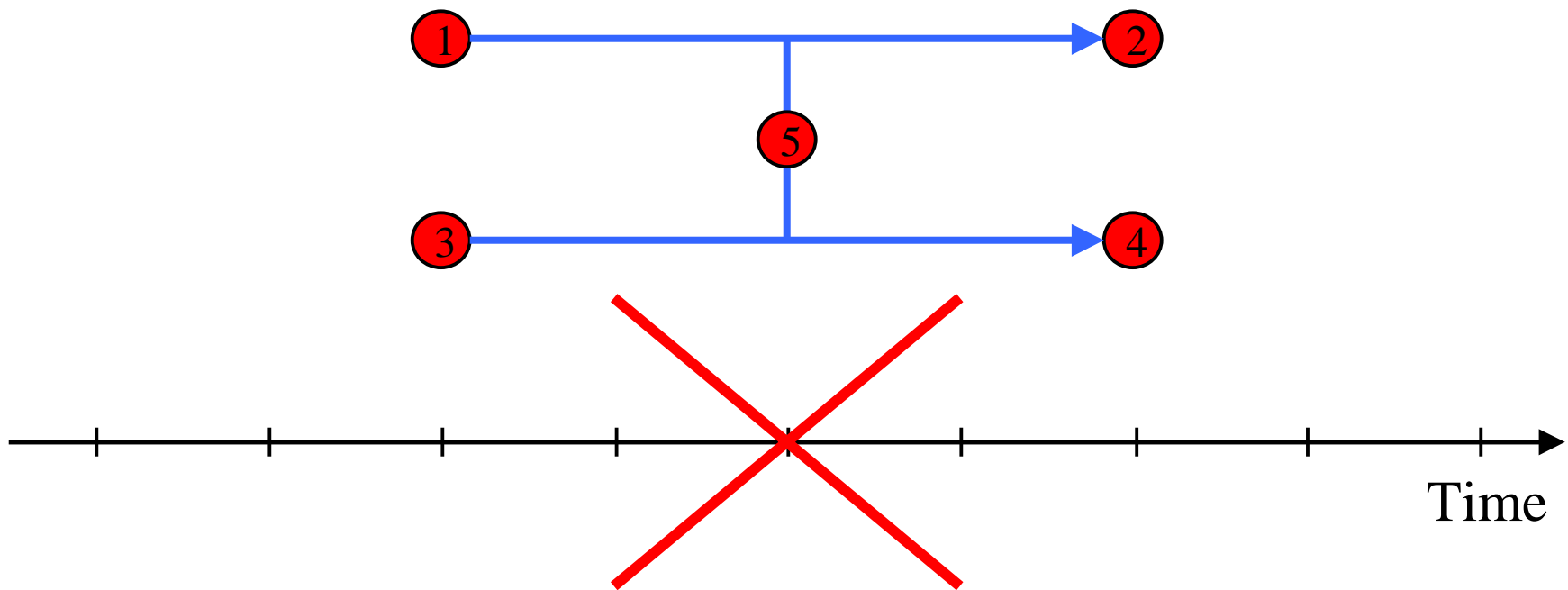
Concurrency



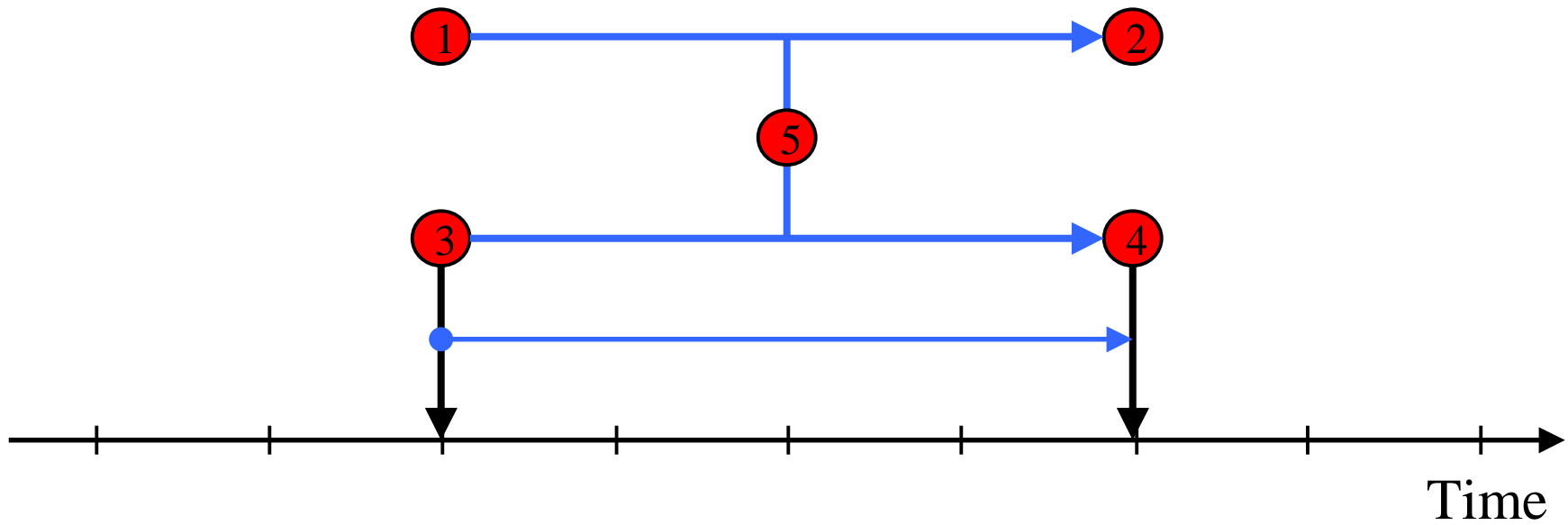
Sequential Programming



Multiprogramming



Real-Time Programming



Embedded Software



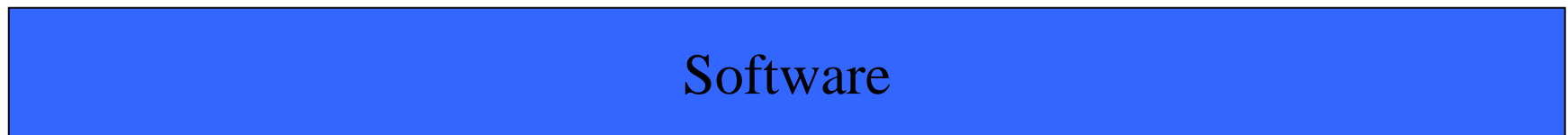
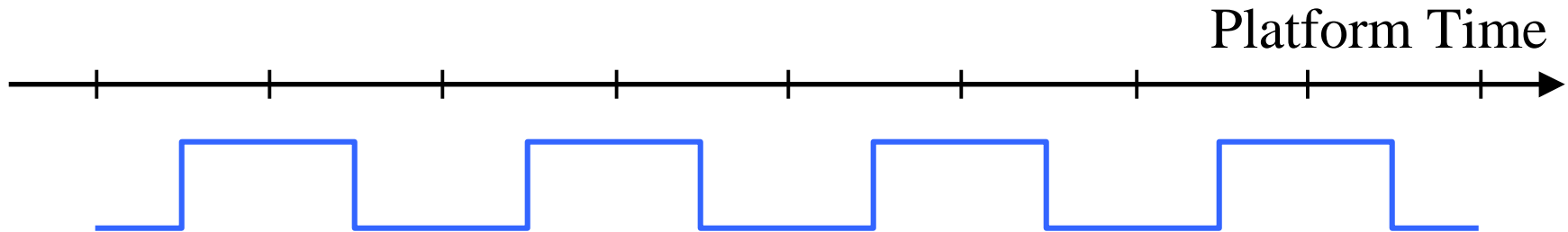
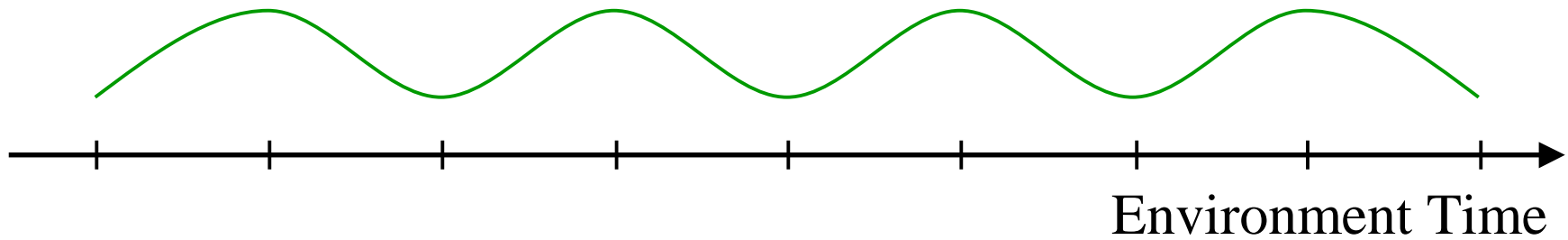
Environment

Environment Processes

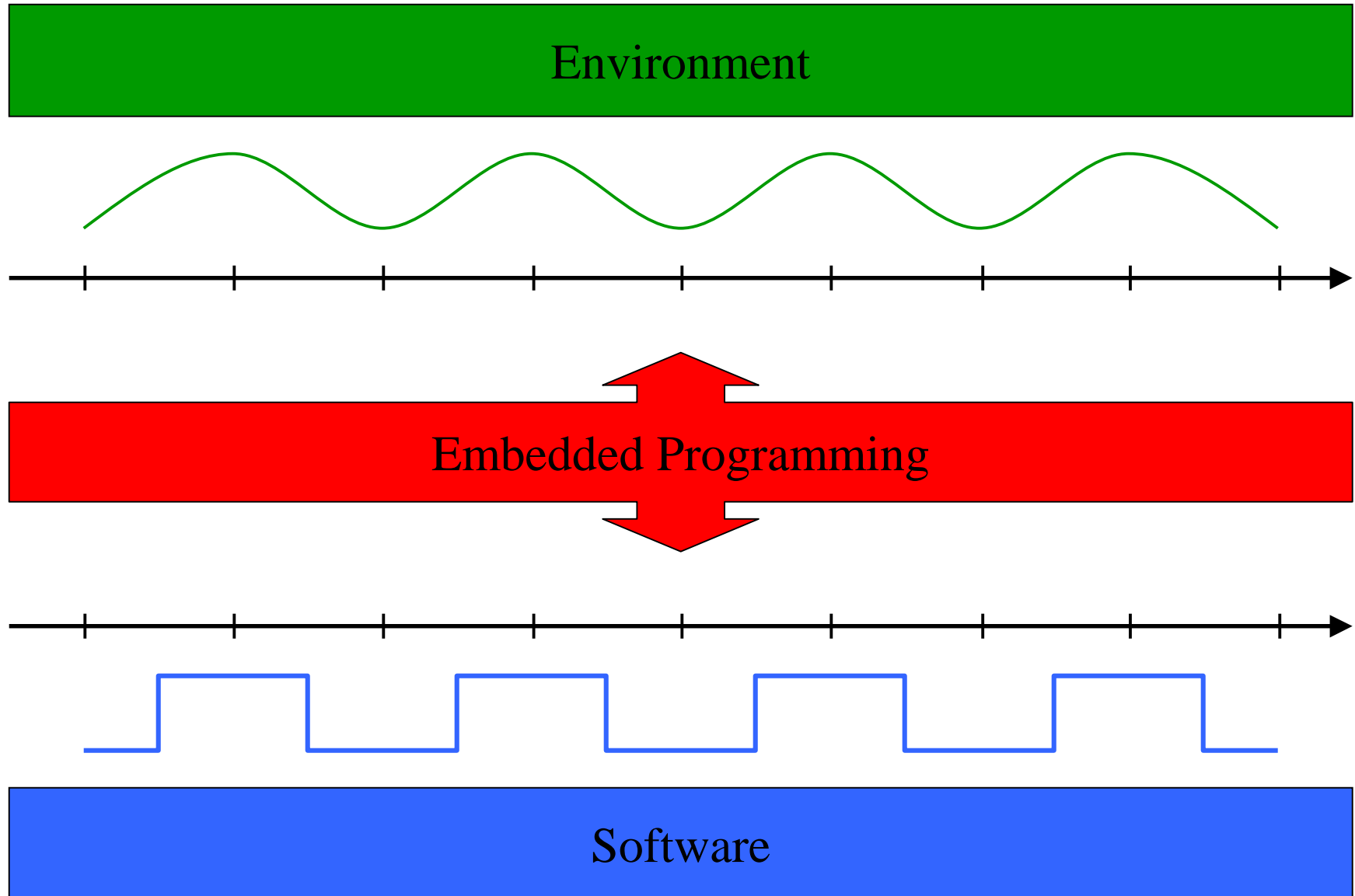
Software Processes

Software

Environment vs. Platform Time



The Art of Embedded Programming

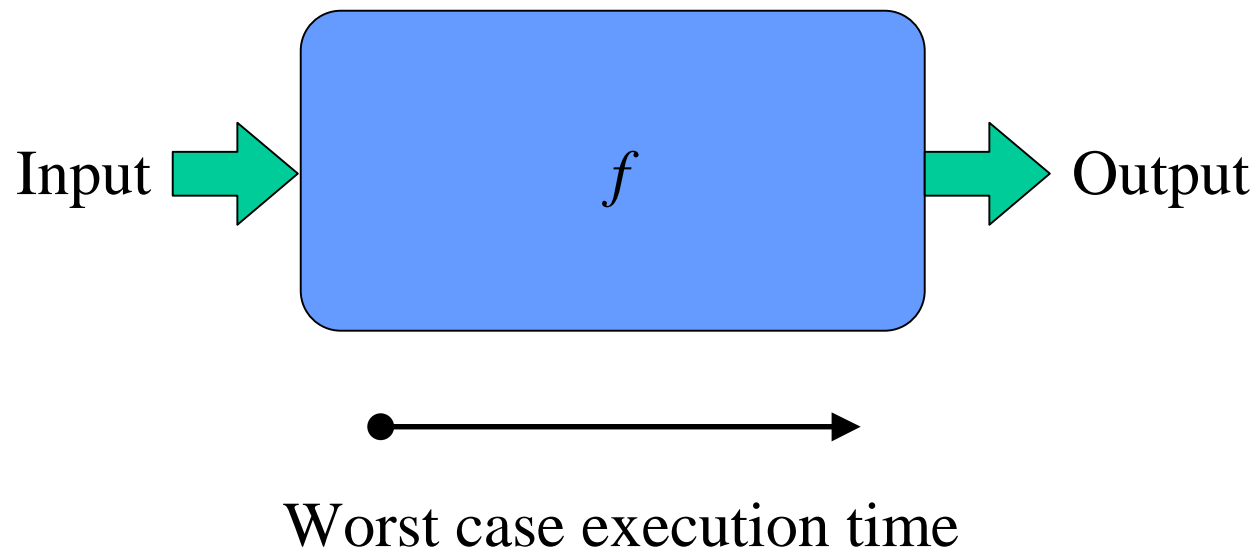


Embedded Programming

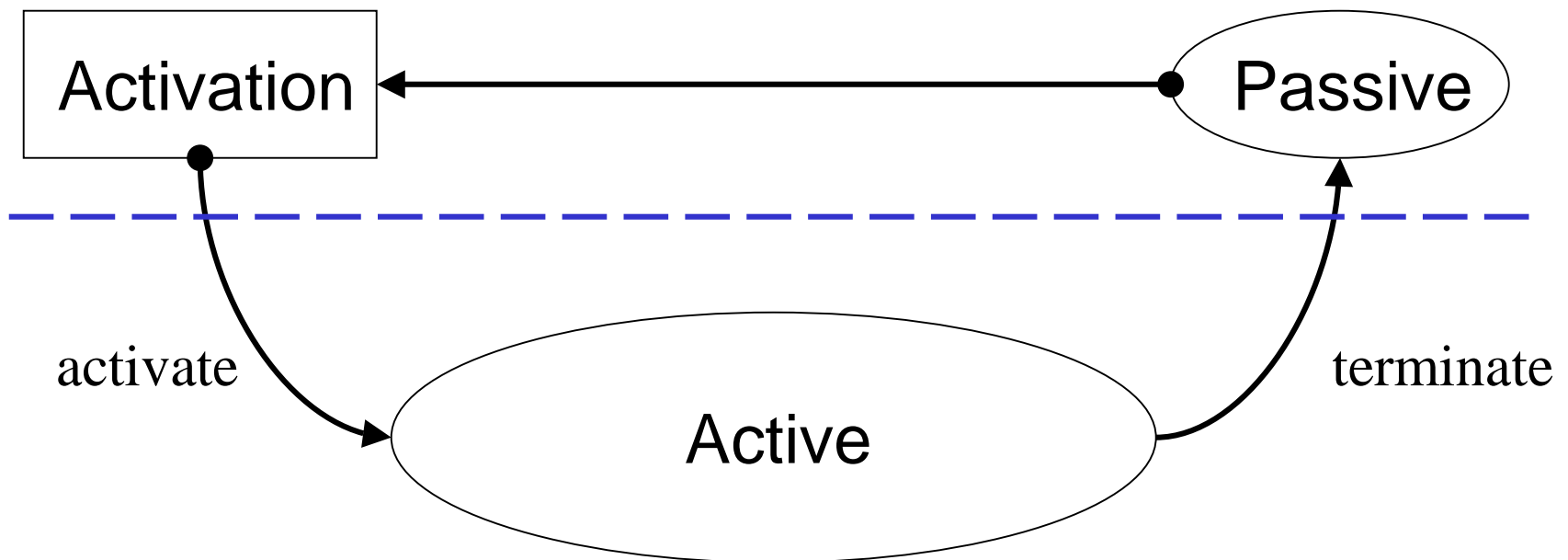
...requires the **integration** of:

1. **Real-time operating system concepts**
2. Embedded programming languages
3. Embedded compilers
4. SE, modeling, and simulation techniques
5. Formal methods

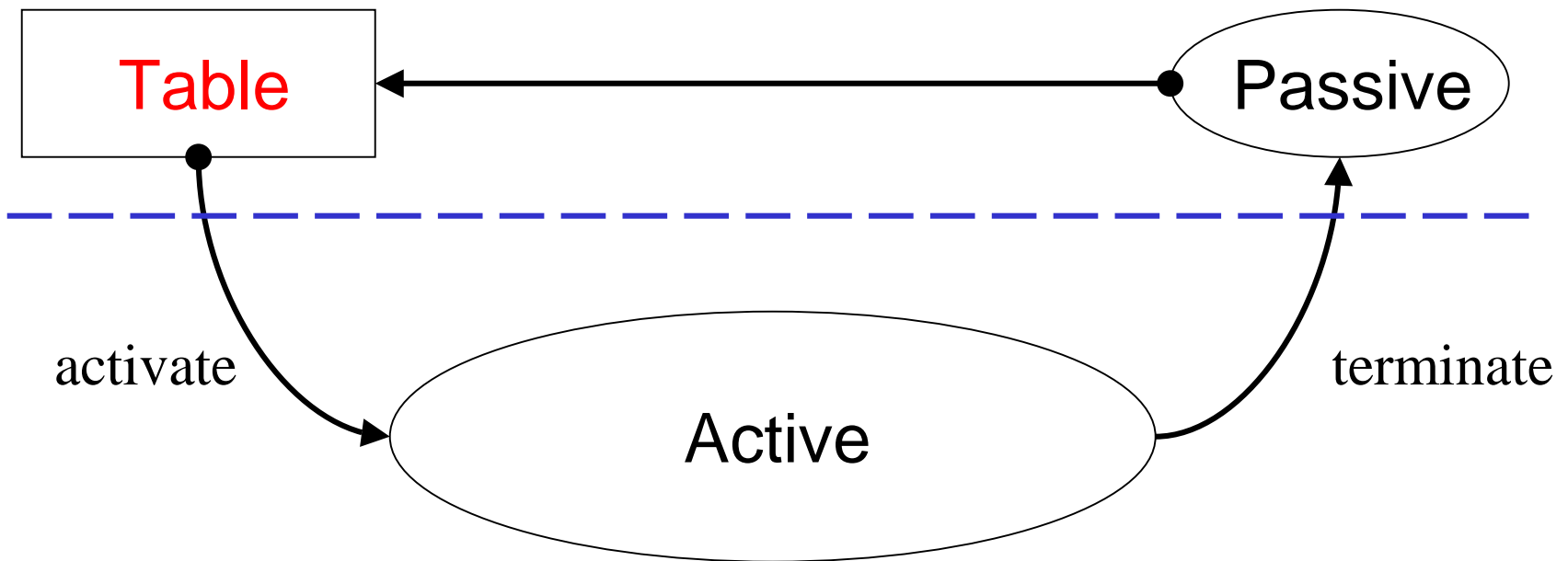
Real-Time Task



Real-Time Scheduling

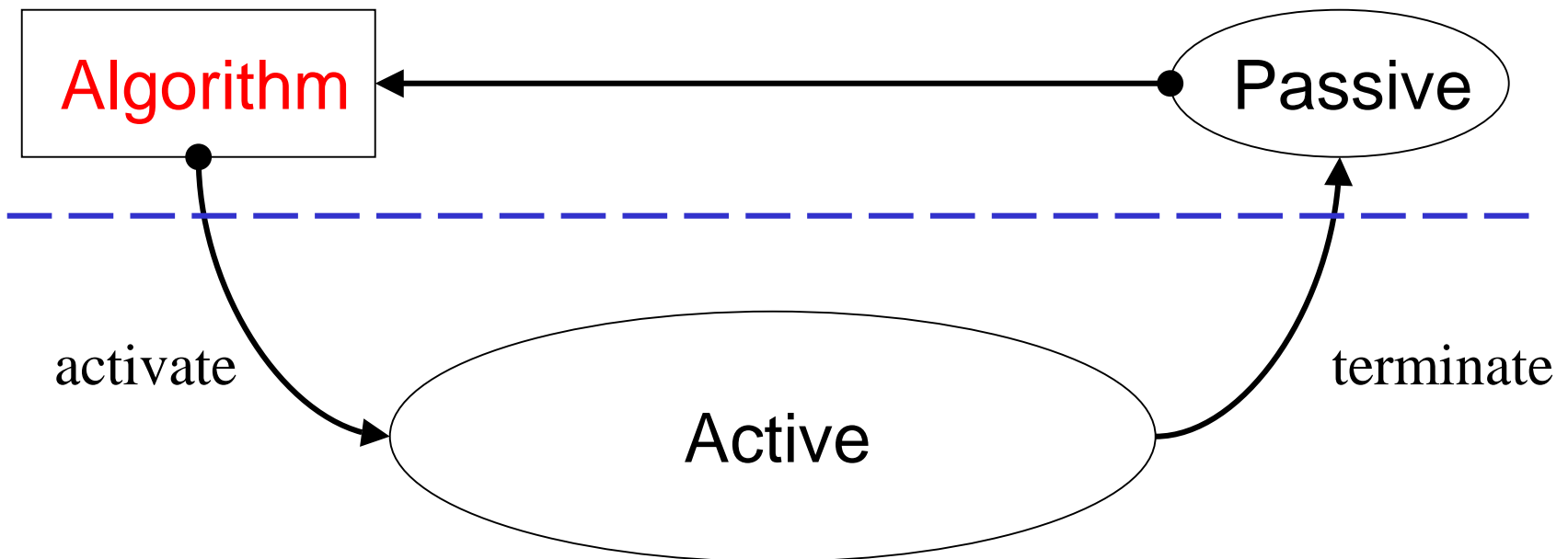


Off-Line Scheduling



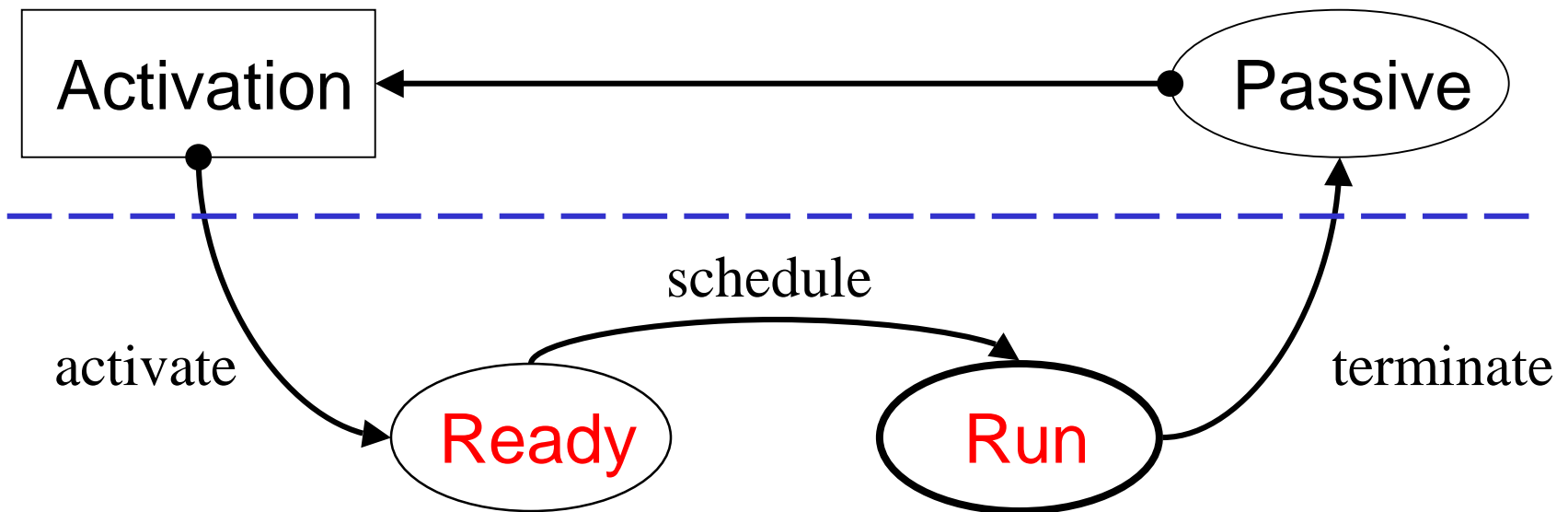
Static System

On-Line Scheduling

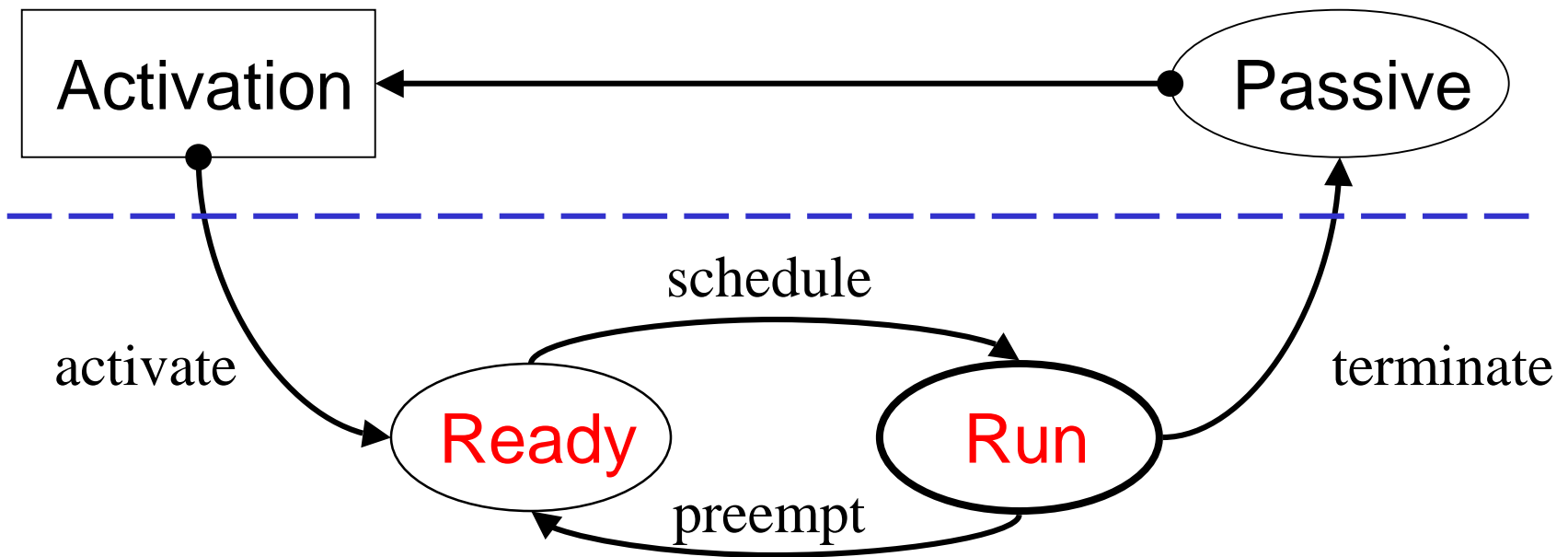


Dynamic System

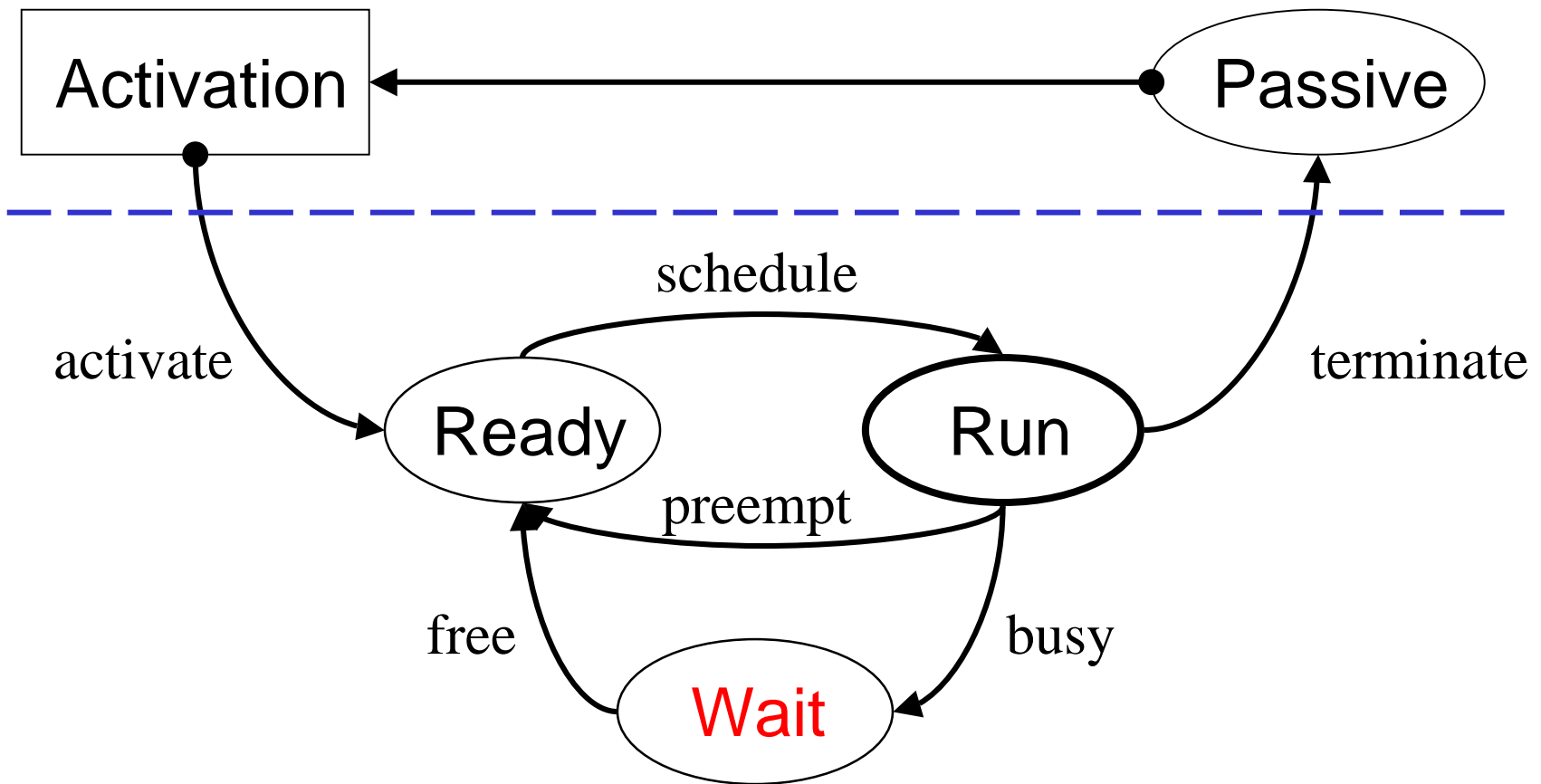
Non-Preemptive Scheduling



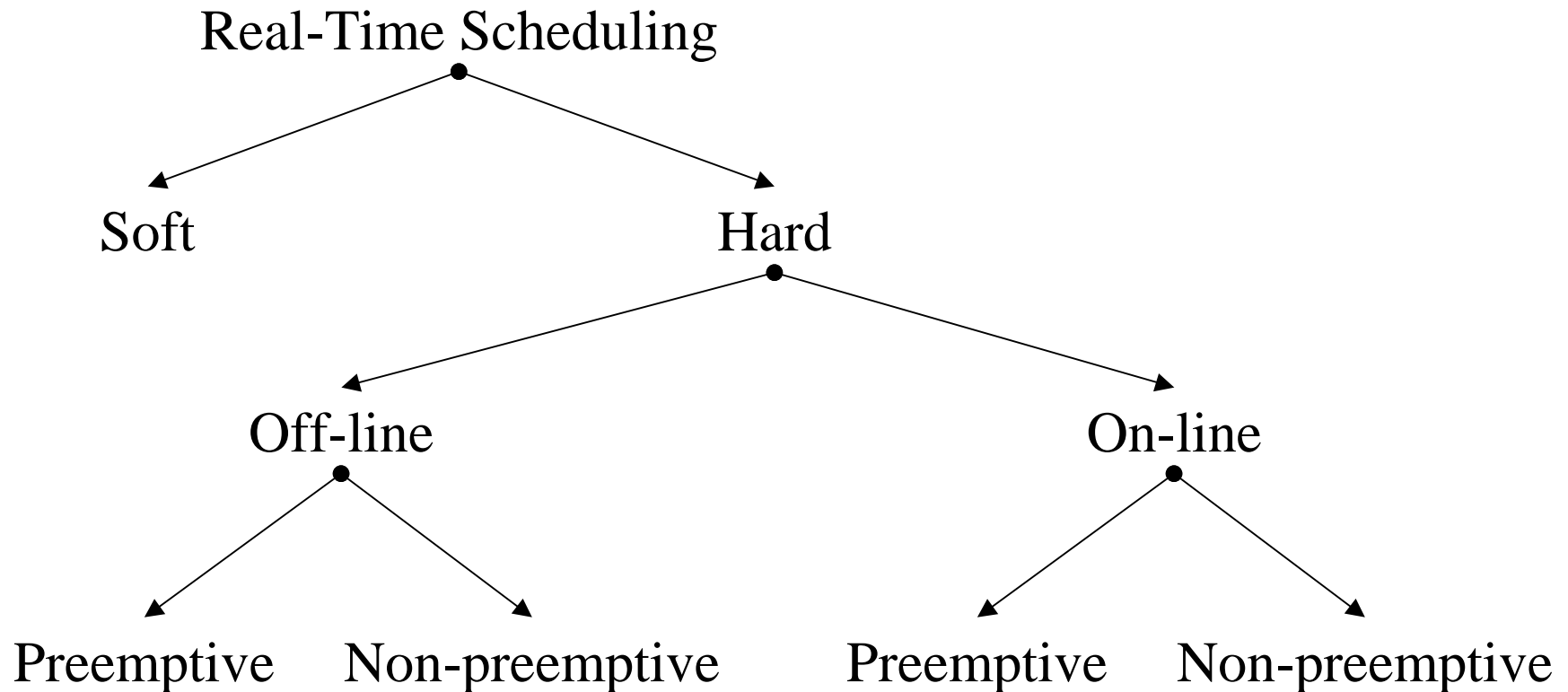
Preemptive Scheduling



Shared Resources



Scheduling Problem

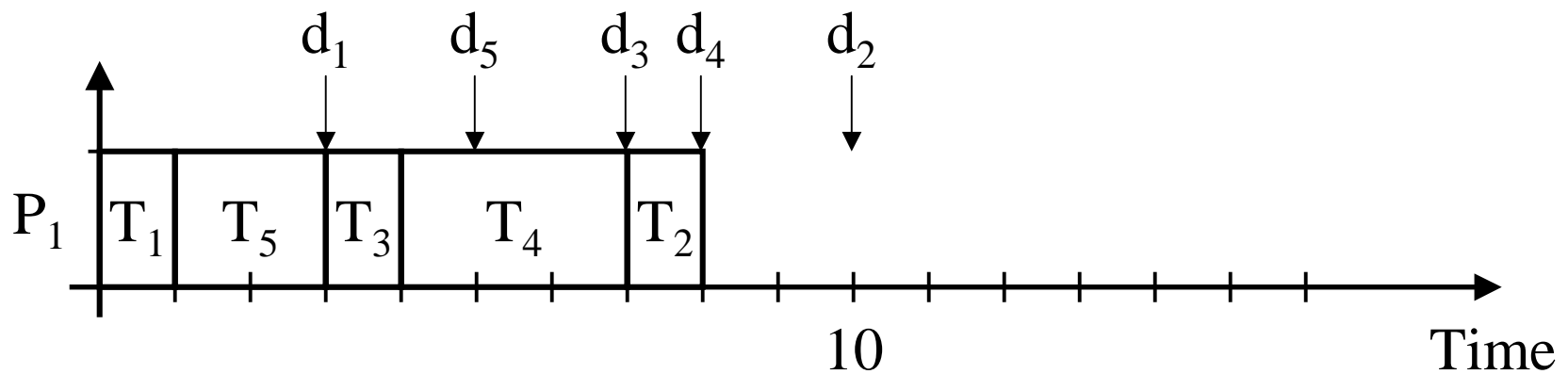


Earliest Due Date

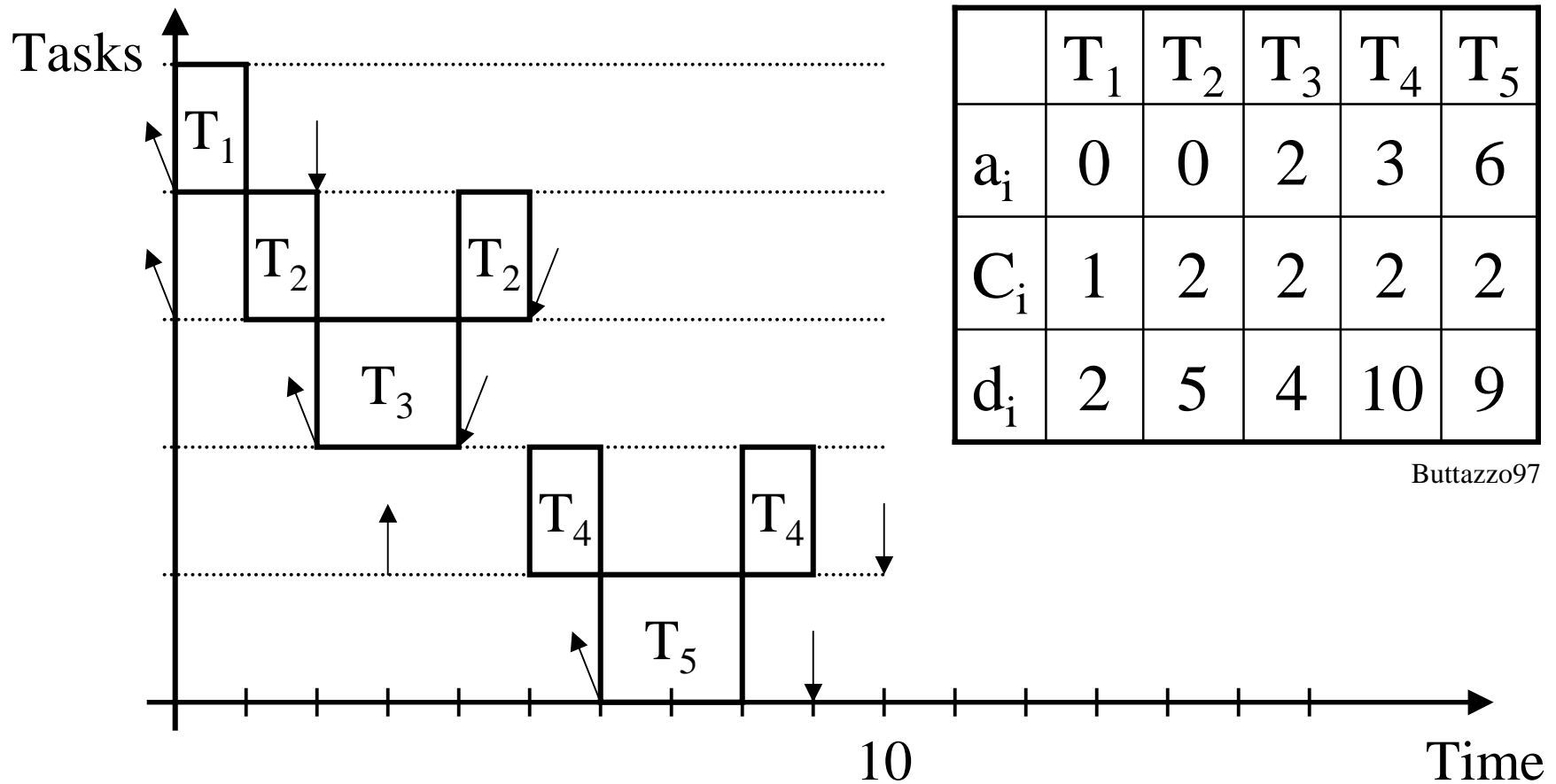
	T_1	T_2	T_3	T_4	T_5
C_i	1	1	1	3	2
d_i	3	10	7	8	5

Buttazzo97

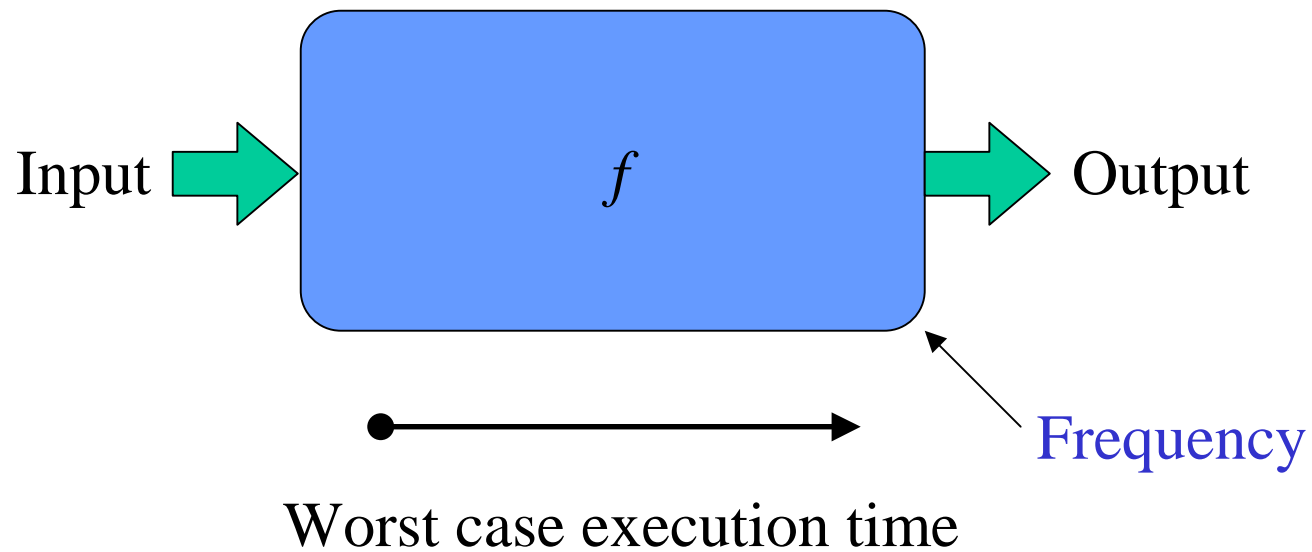
Processors



Earliest Deadline First

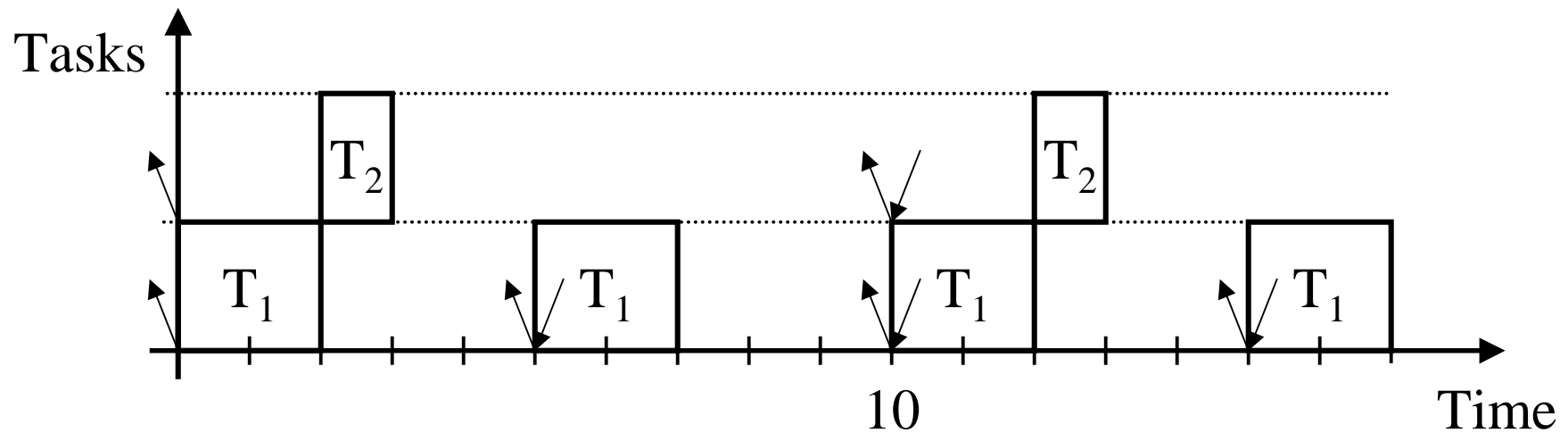


Real-Time Periodic Task

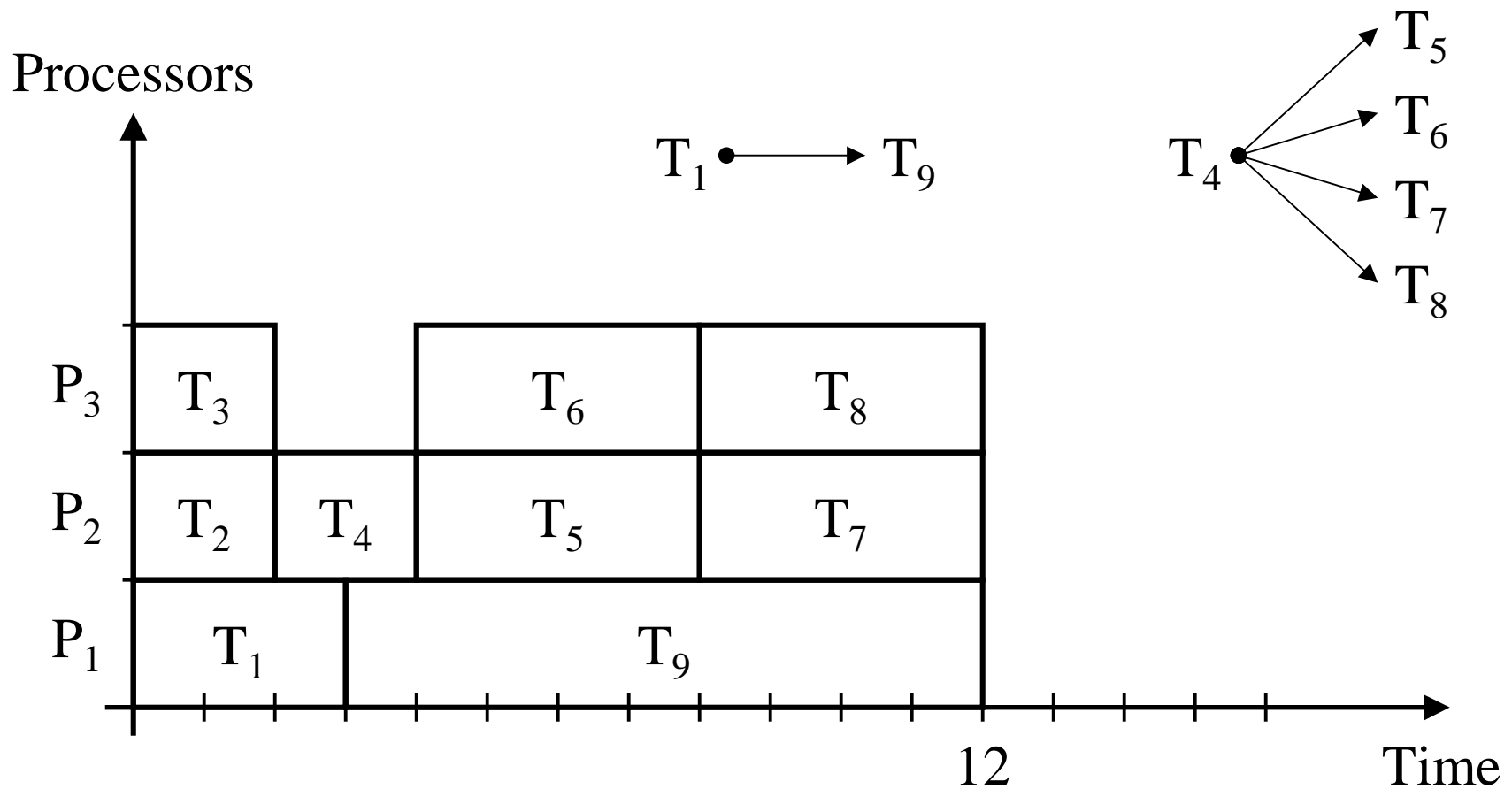


Rate Monotonic Analysis

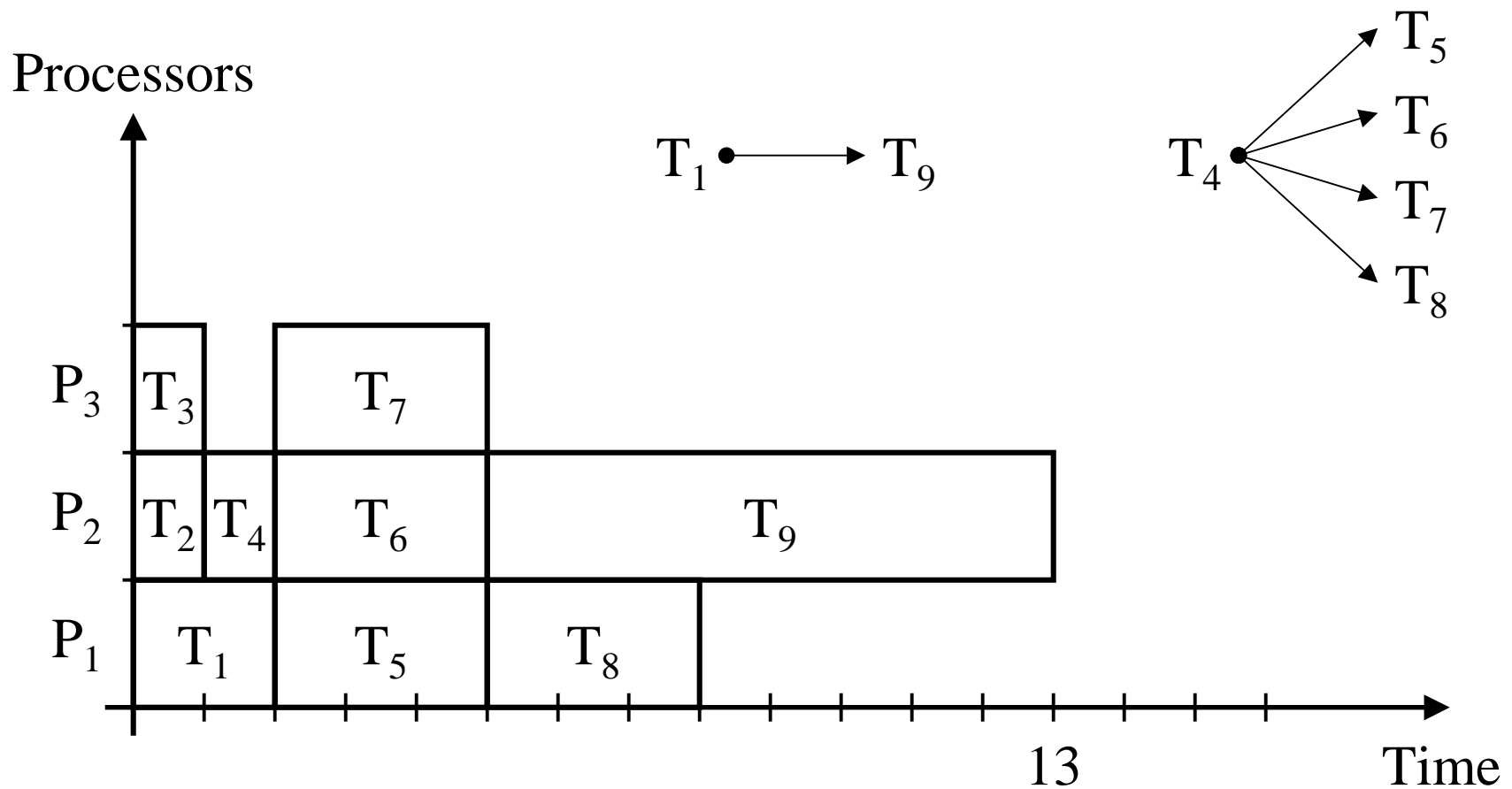
	T_1	T_2
C_i	2	1
p_i	5	10



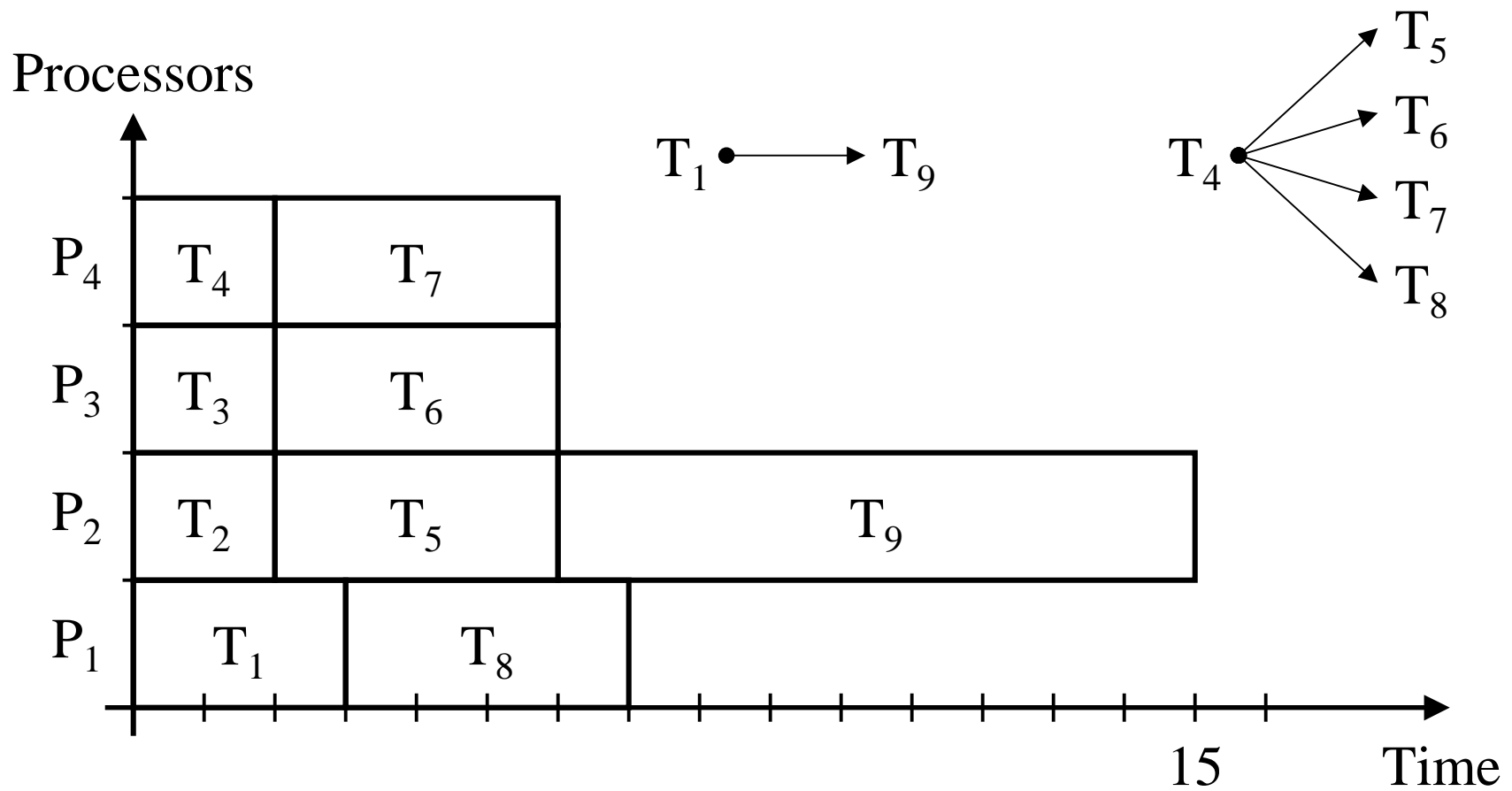
Scheduling Anomalies



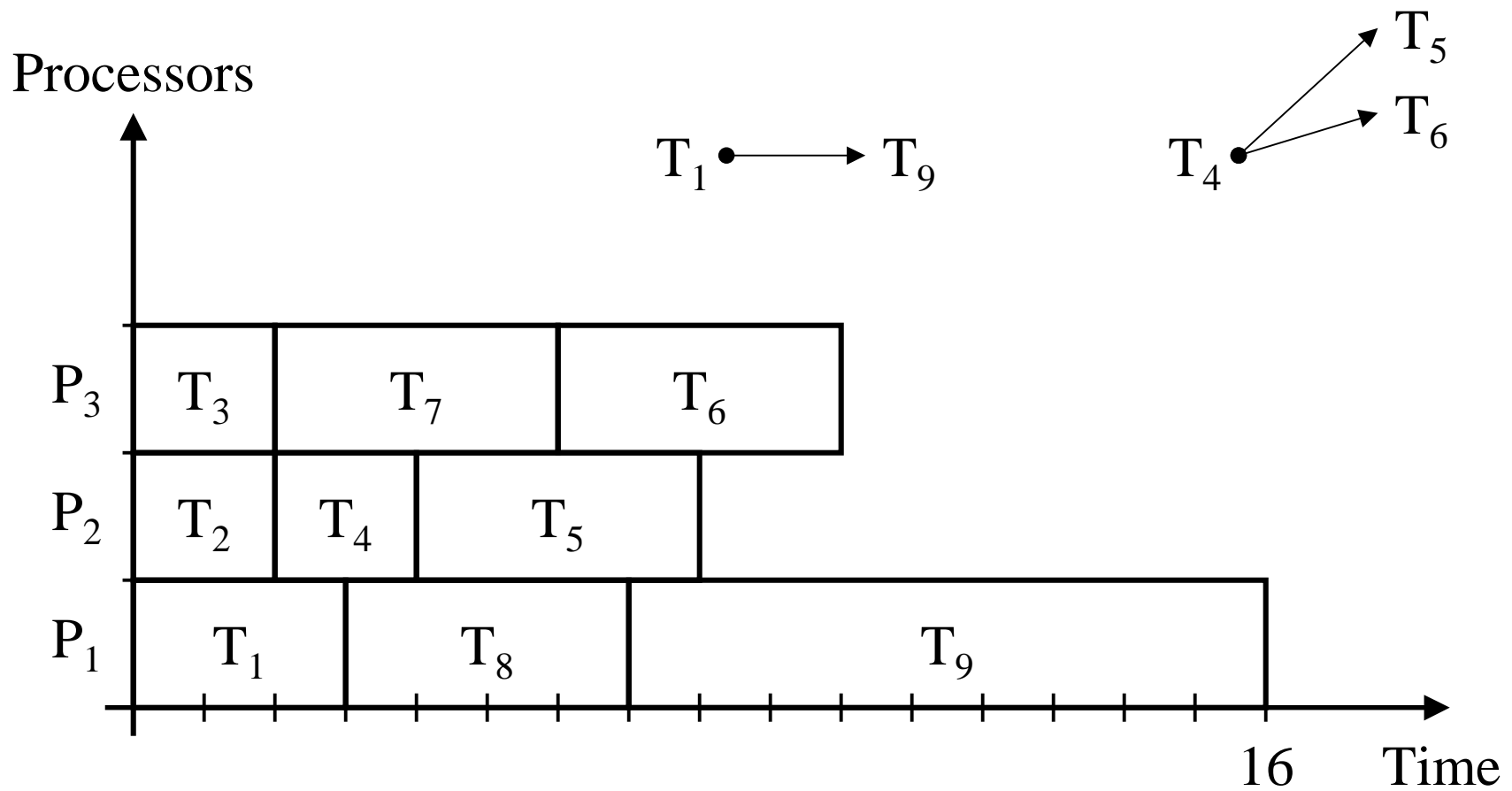
Shorter Computation Times



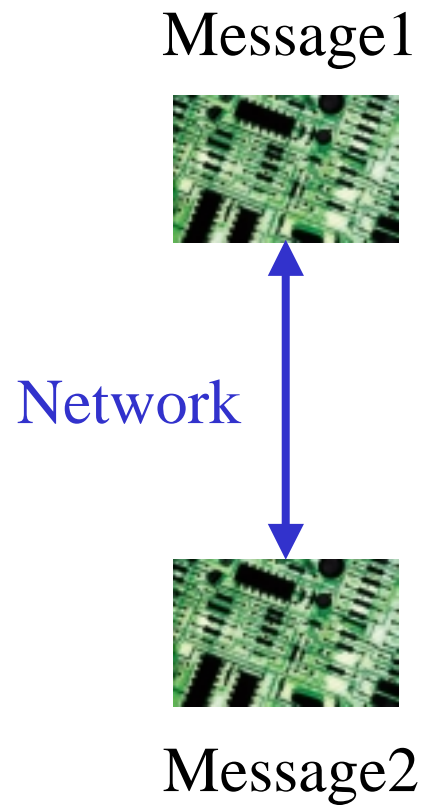
More Processors



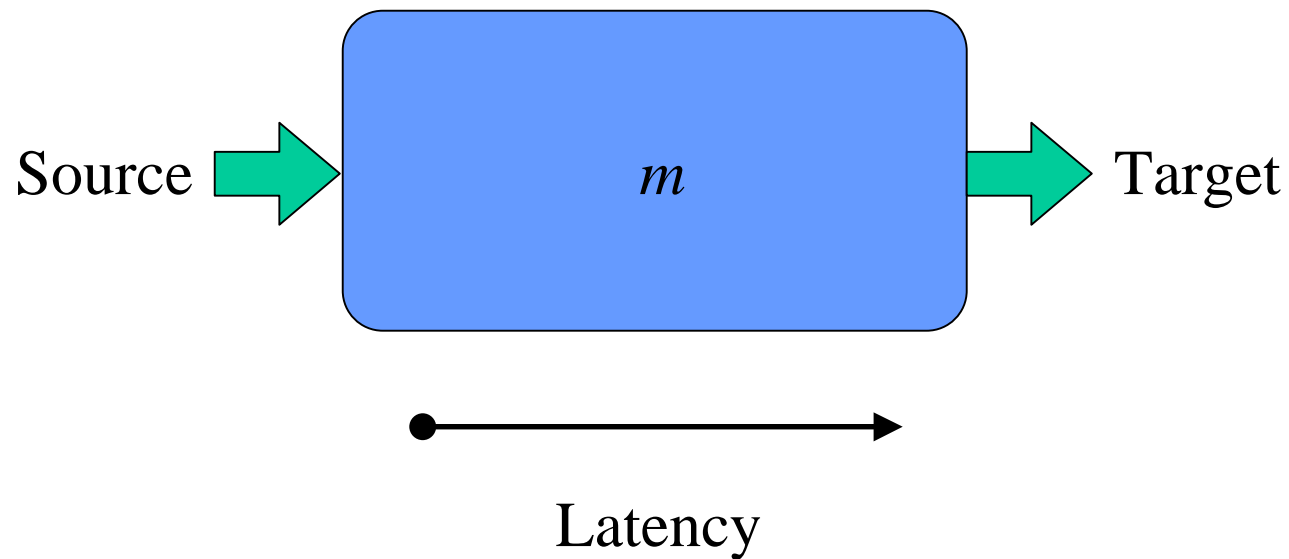
Weaker Precedence



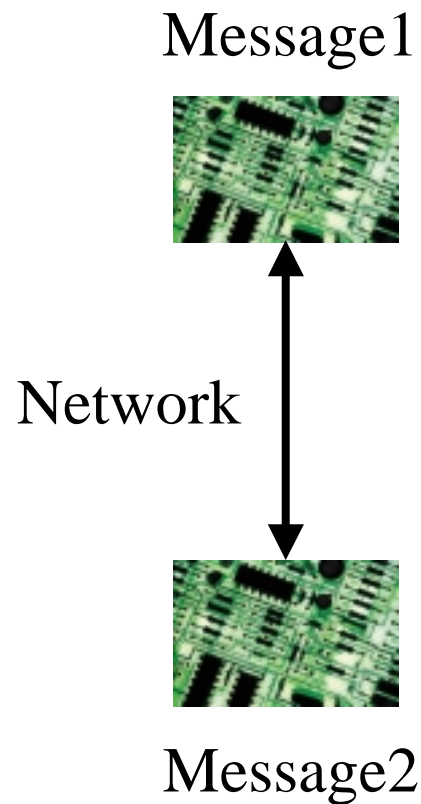
Real-Time Communication



Real-Time Message

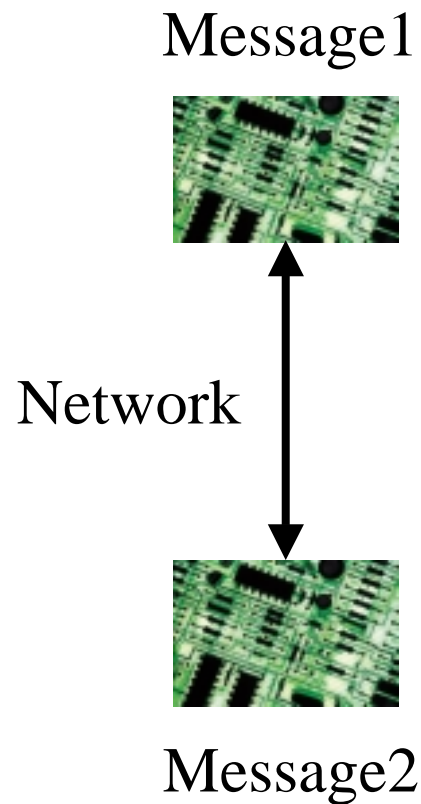


Explicit Flow Control



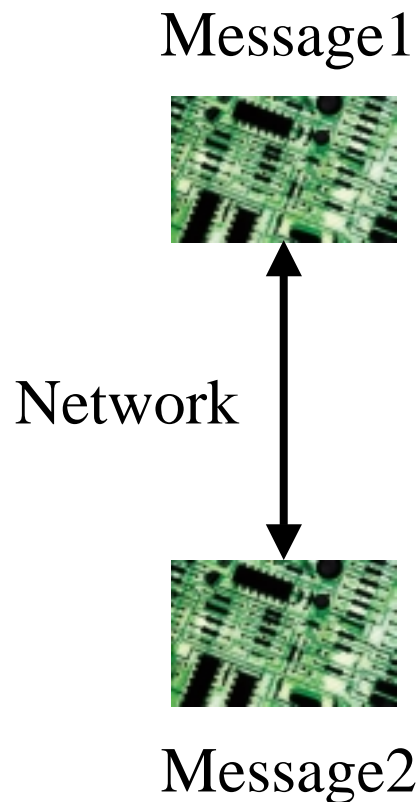
- Send time not known a priori
- Sender can detect errors

Implicit Flow Control



- Send time is known a priori
- Receiver can detect errors

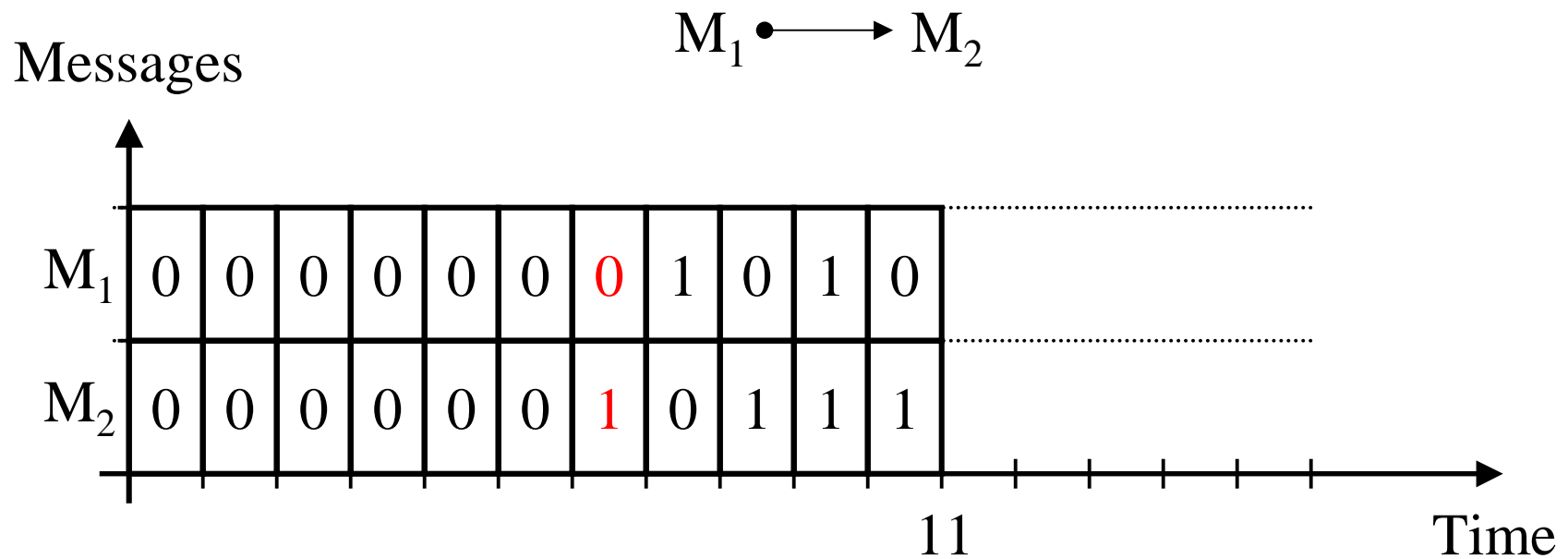
Explicit Flow Control: Priority



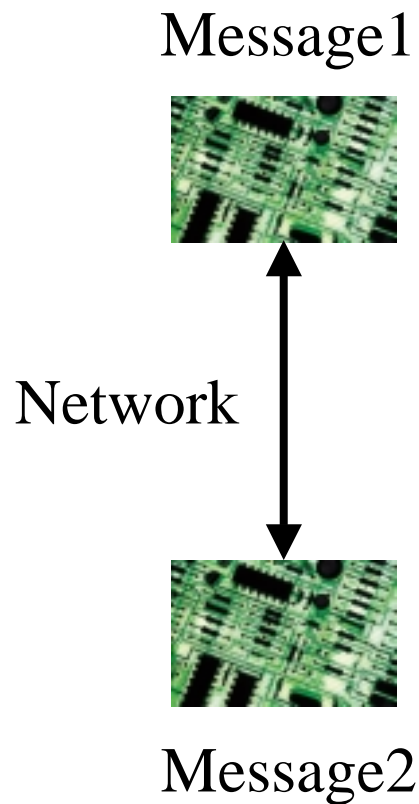
Medium-Access Protocols:

- CSMA/CD - LON, Echelon 1990
- CSMA/CA - **CAN**, Bosch 1990
- FTDMA - Byteflight, BMW 2000
- TDMA - TTP, Kopetz 1993

Control Area Network



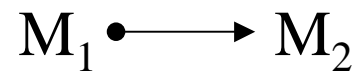
Implicit Flow Control: Time



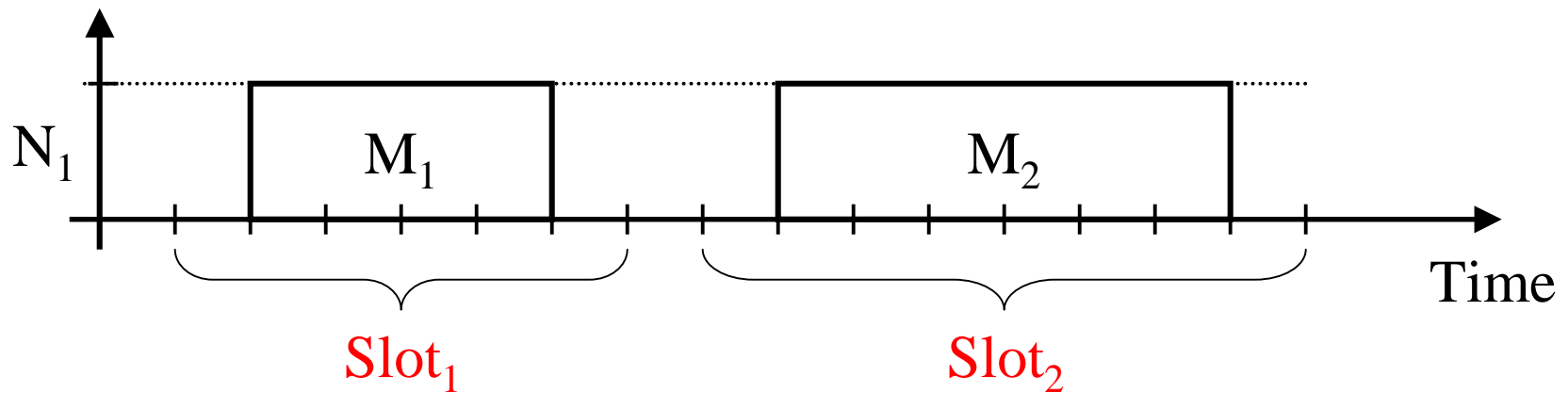
Medium-Access Protocols:

- FTDMA - Byteflight, BMW 2000
- TDMA - **TTP**, Kopetz 1993

Time-Triggered Protocol



Network



Literature

- RT scheduling:
 - Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications. G. Buttazzo. Kluwer, 1997.
- RT communication:
 - Real-Time Systems – Design Principles for Distributed Embedded Applications. H. Kopetz. Kluwer, 1997.
 - Byteflight, CAN papers.

Embedded Programming

...requires the **integration** of:

1. Real-time operating system concepts
2. **Embedded programming languages**
3. Embedded compilers
4. SE, modeling, and simulation techniques
5. Formal methods

Concurrency

Parallel Composition

Task1 || Task2

Control

I/O Decomposition

Task1 ↔ Task2

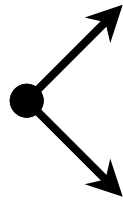
Data

Control Operators

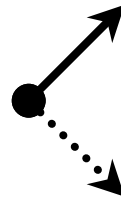
Sequential



Parallel



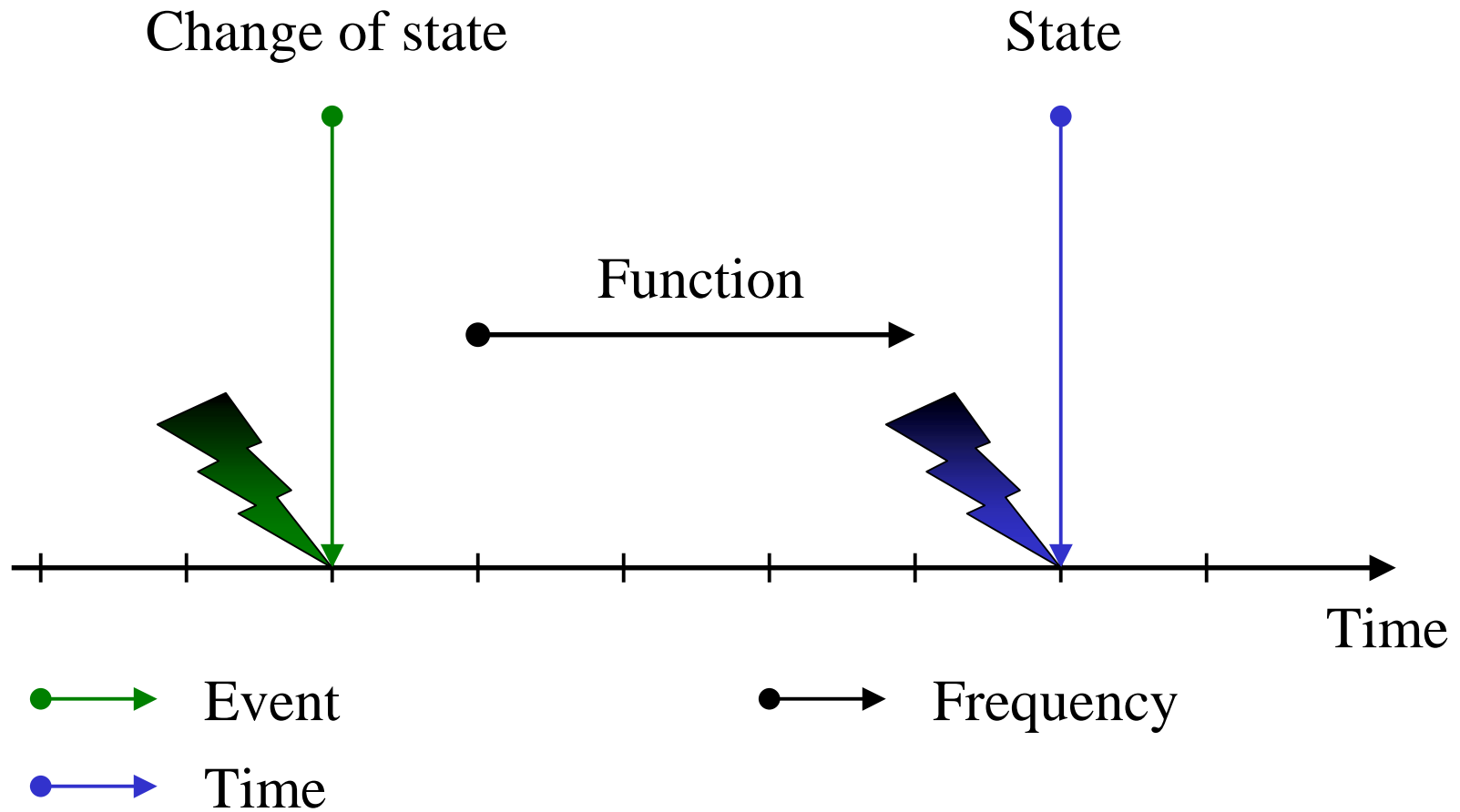
Choice



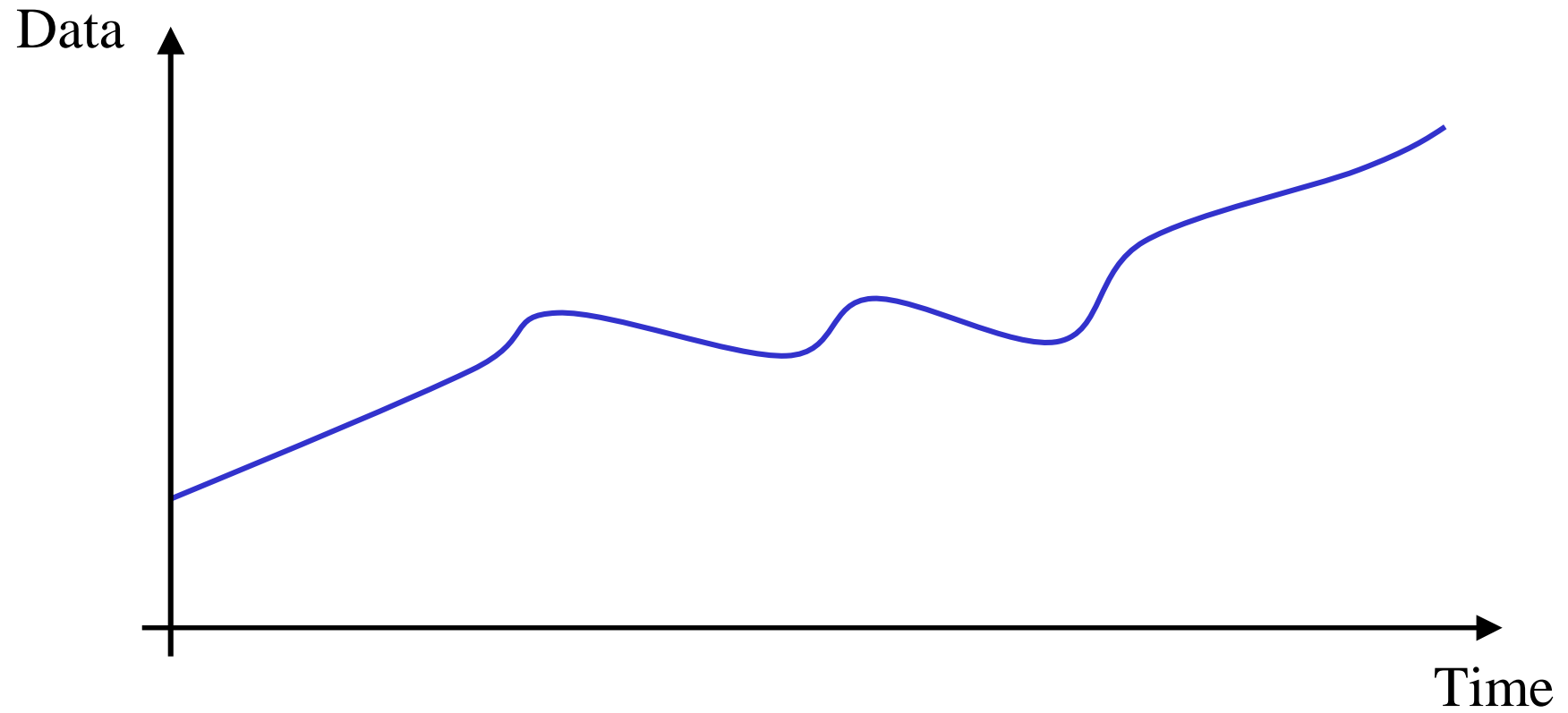
Loop



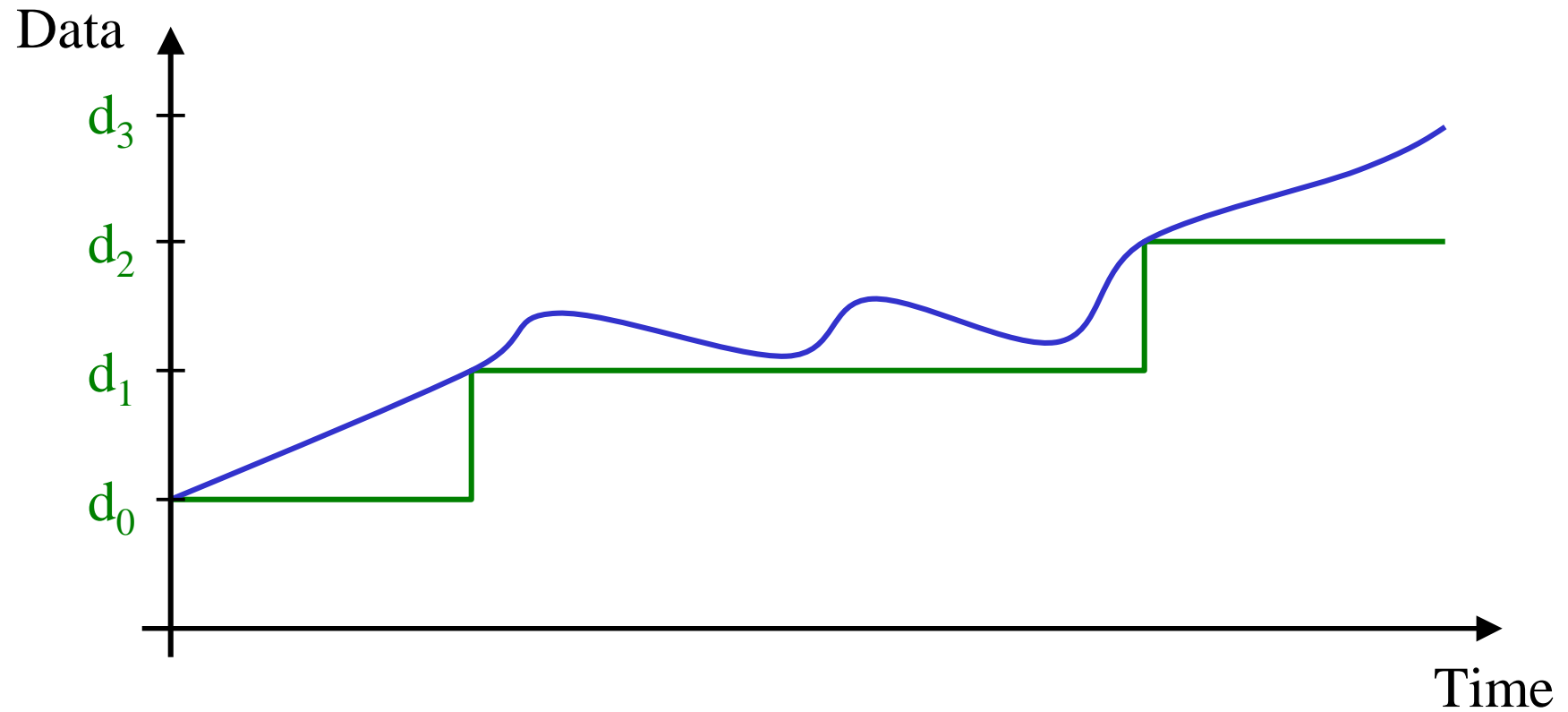
Real-Time



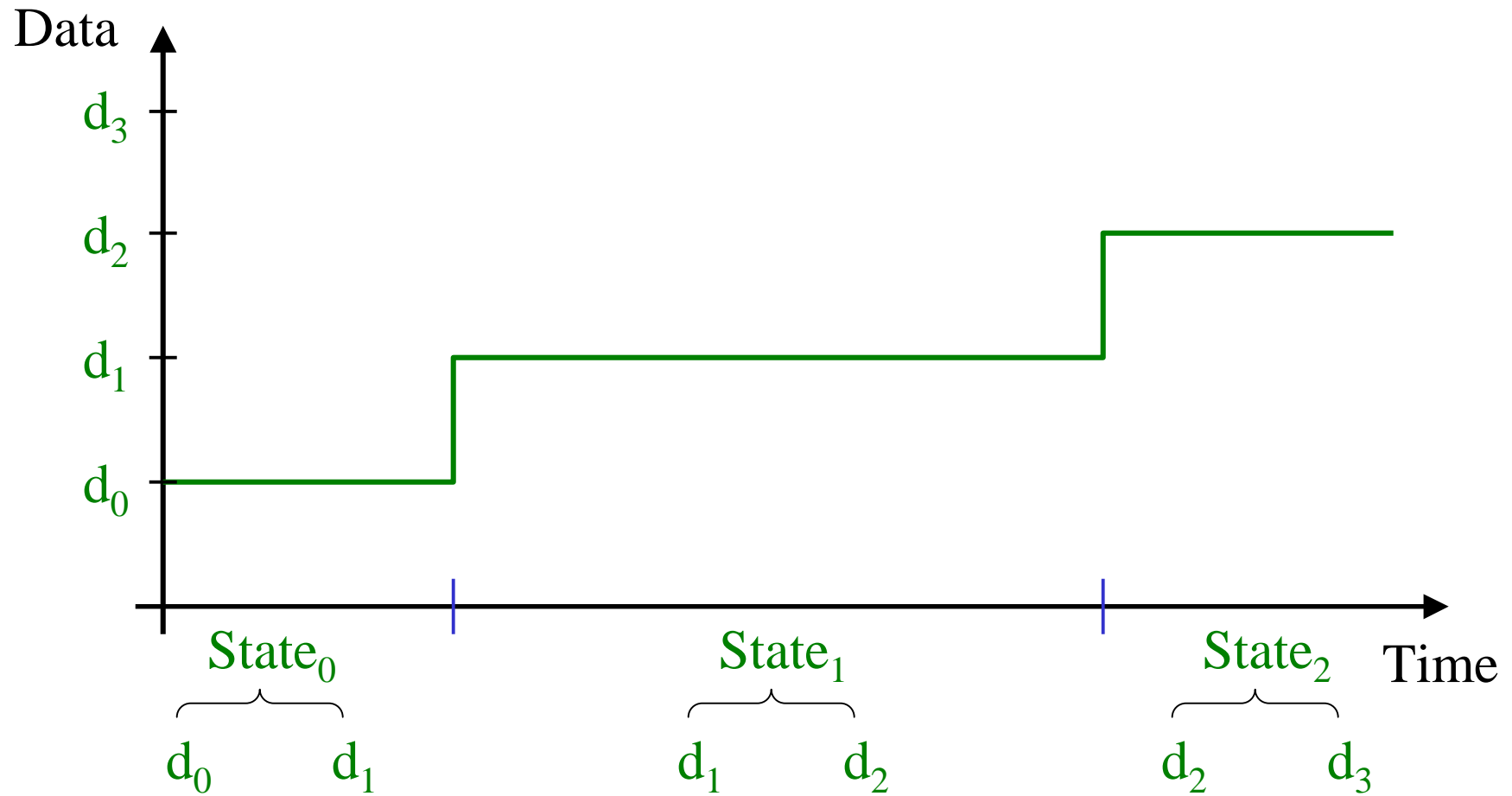
Real World



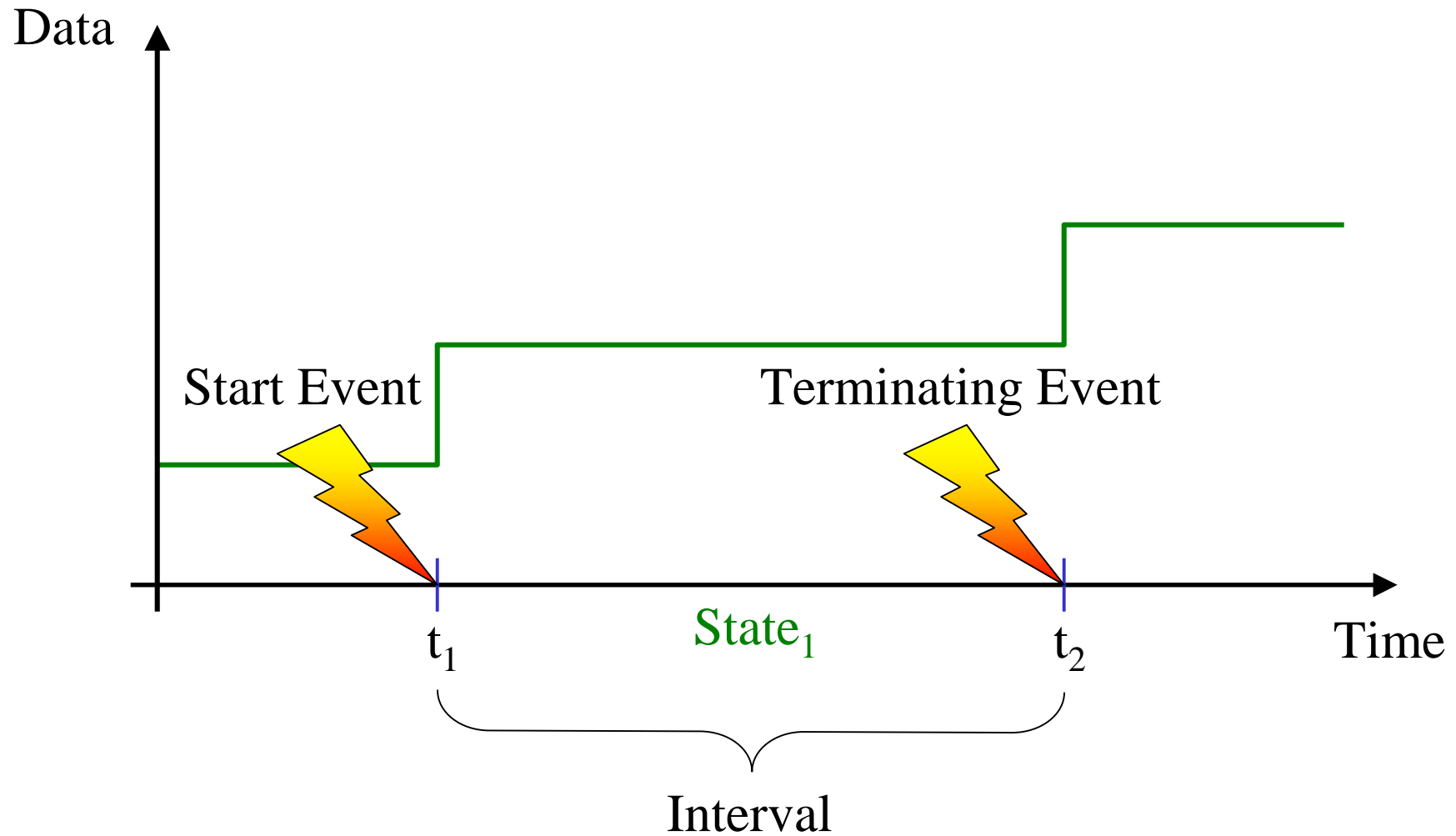
Discrete Data



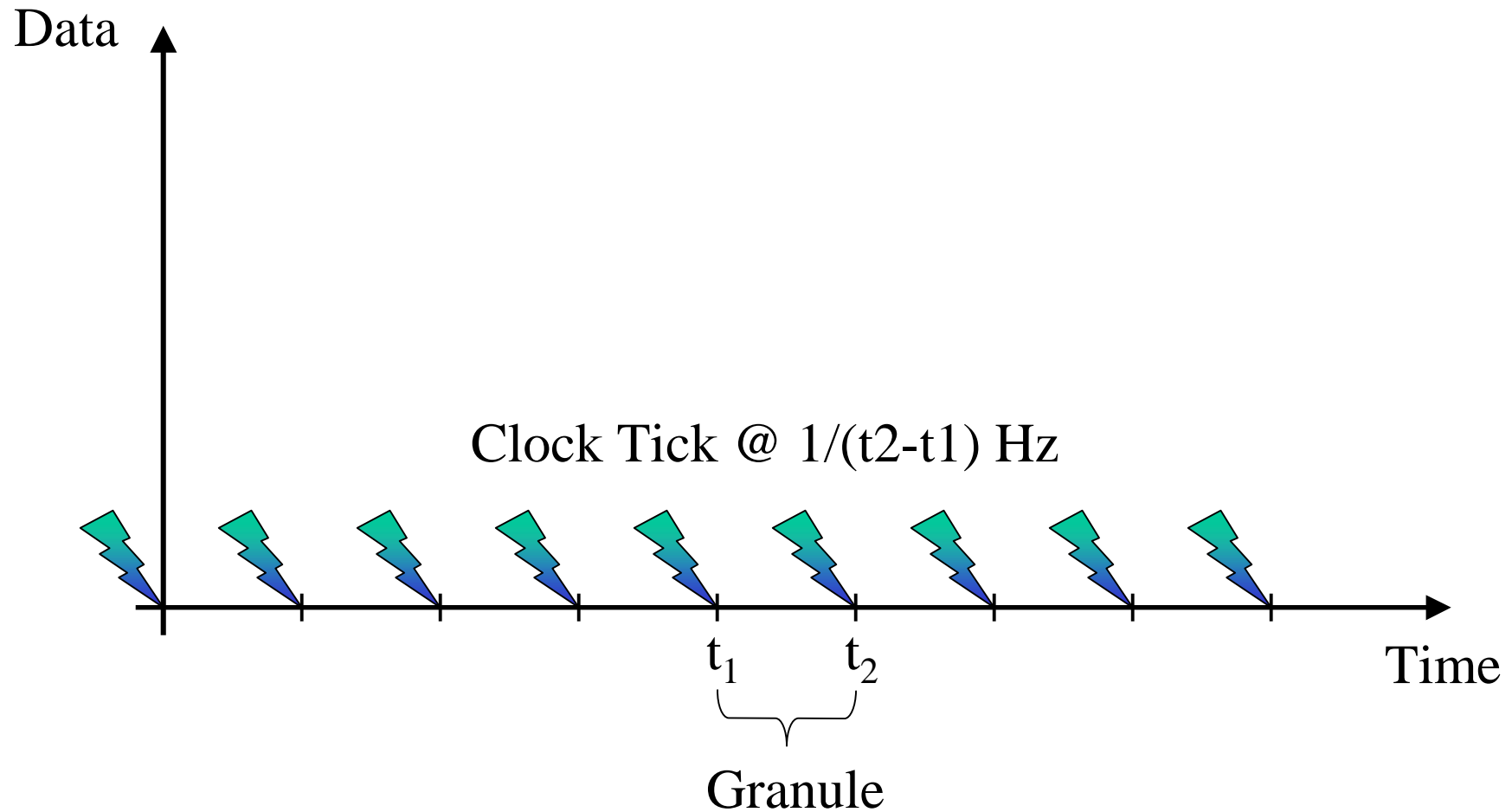
State



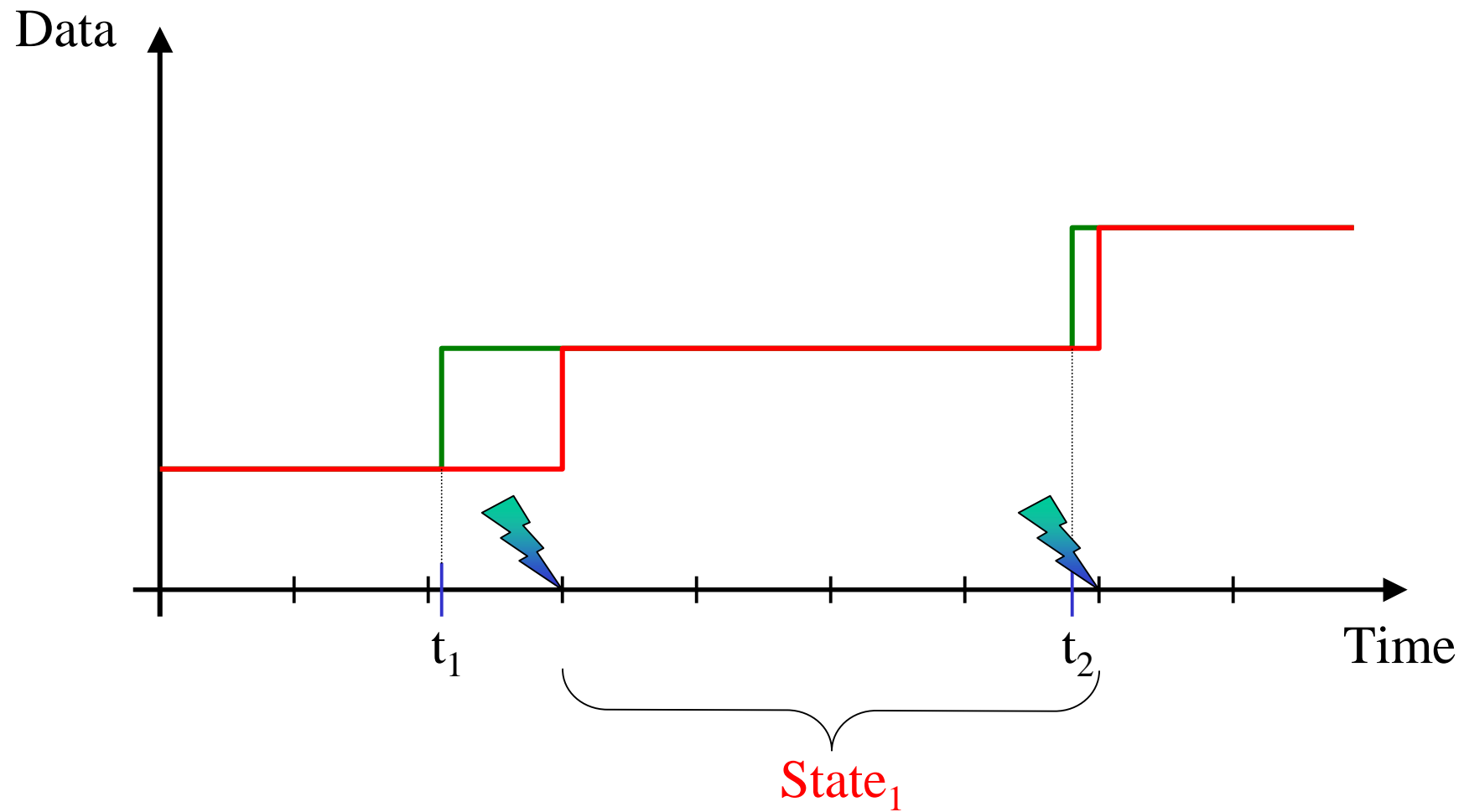
Continuous Time



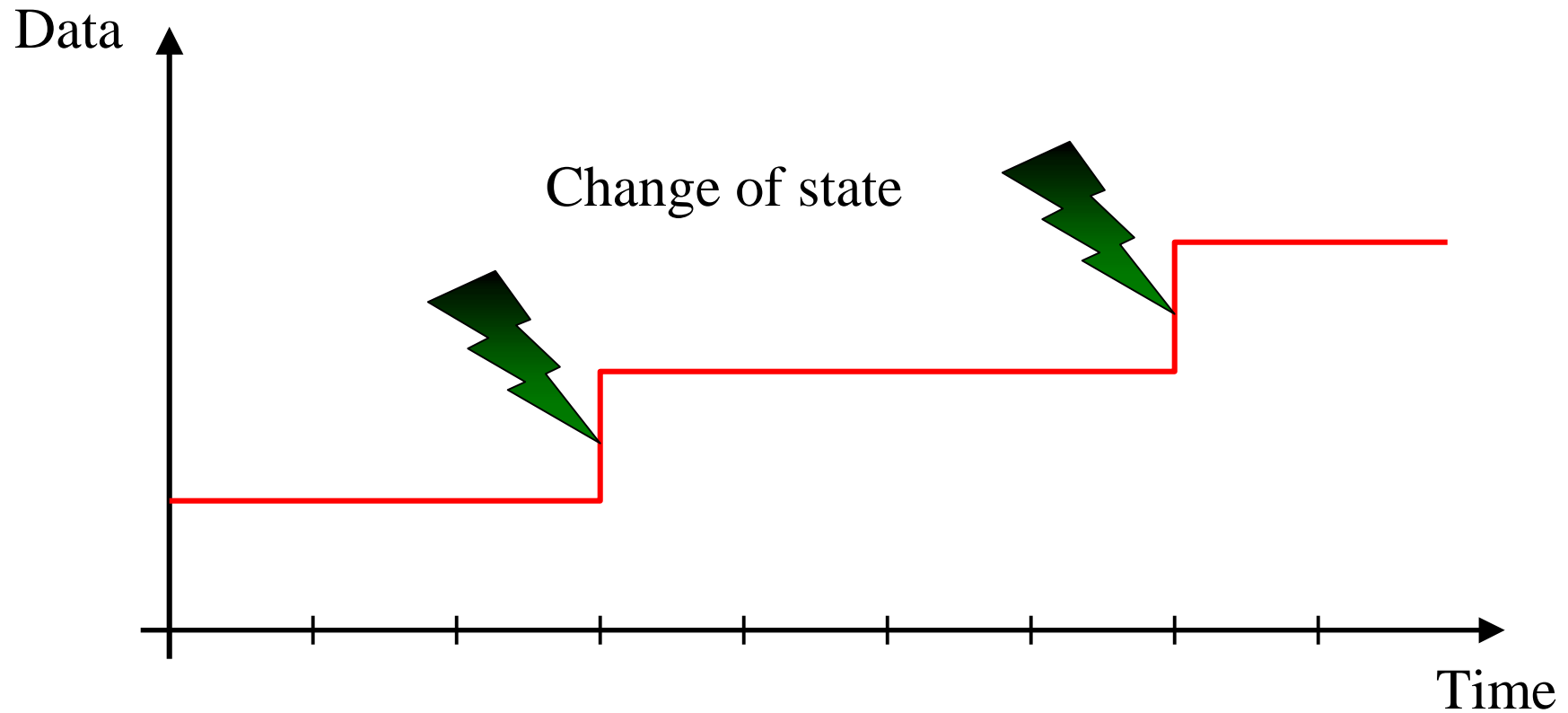
Digital Clock



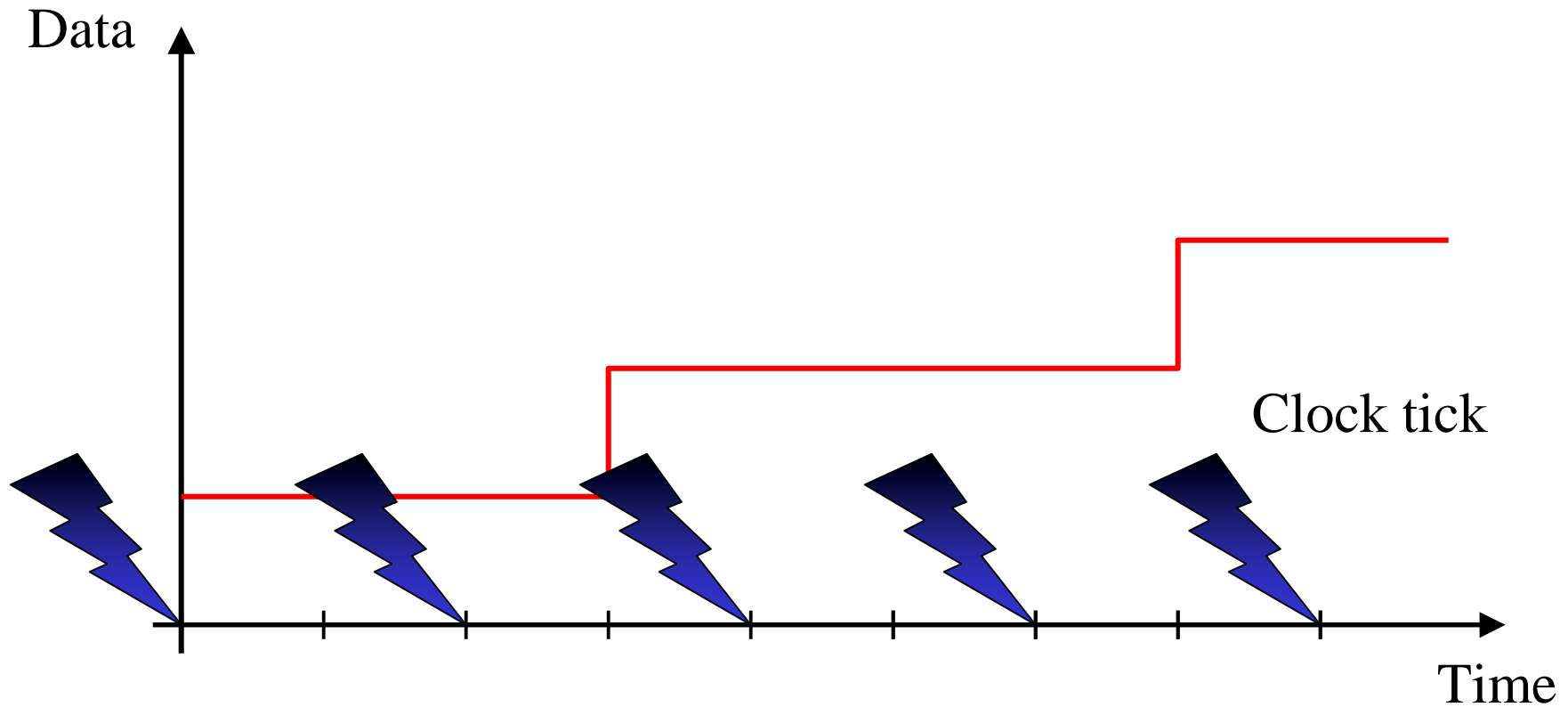
Discrete Time



Event-Triggered (ET) System



Time-Triggered (TT) System

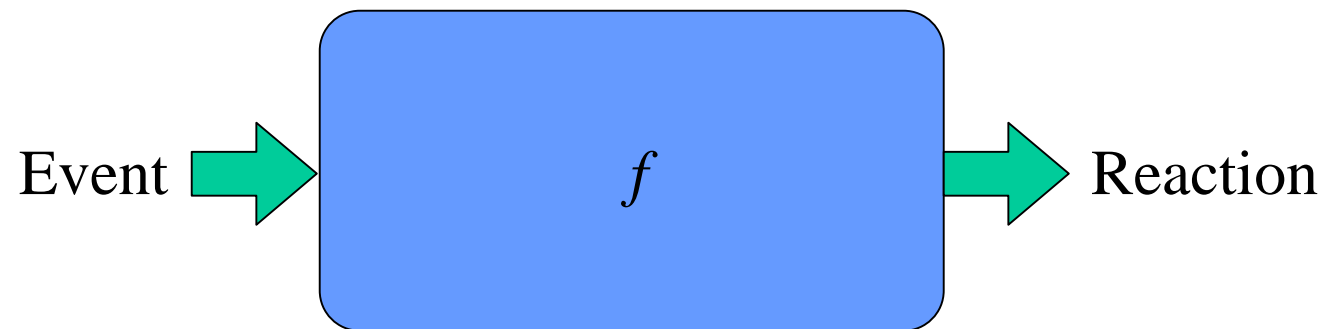


Esterel - Giotto

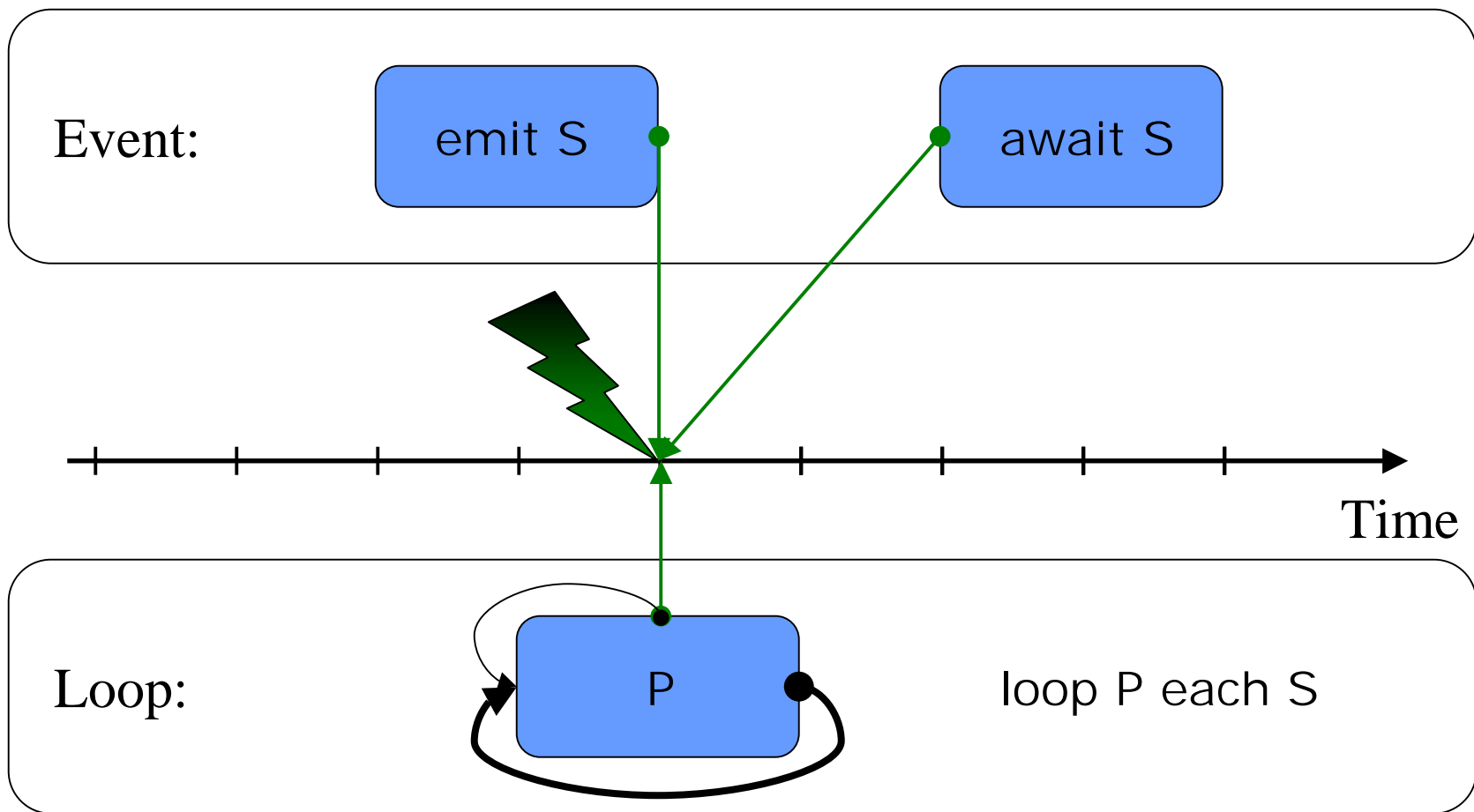
- Esterel:
 - Synchronous reactive language
 - Event-triggered semantics

- Giotto:
 - Time-triggered semantics
 - Distributed platforms

Event - Reaction

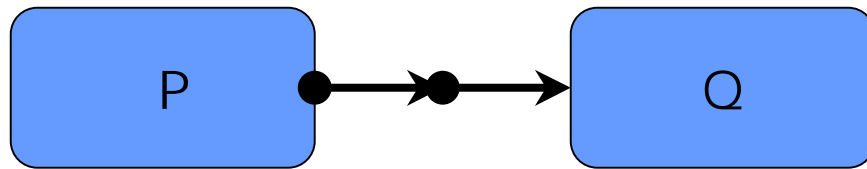


Esterel: Event



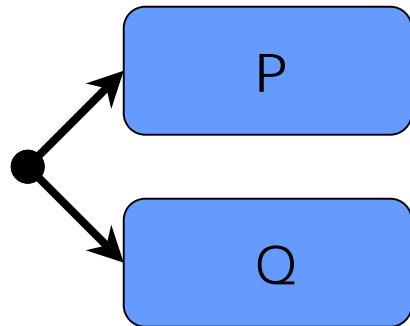
Esterel: Operators

Sequential:



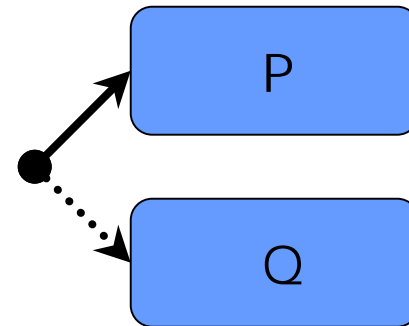
$P ; Q$

Parallel:



$P \parallel Q$

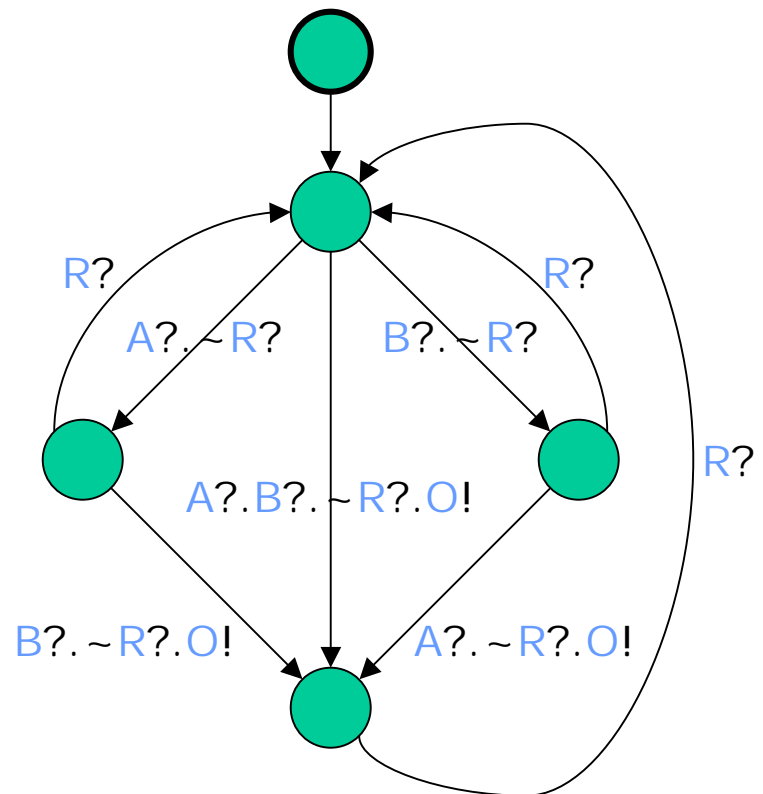
Choice:



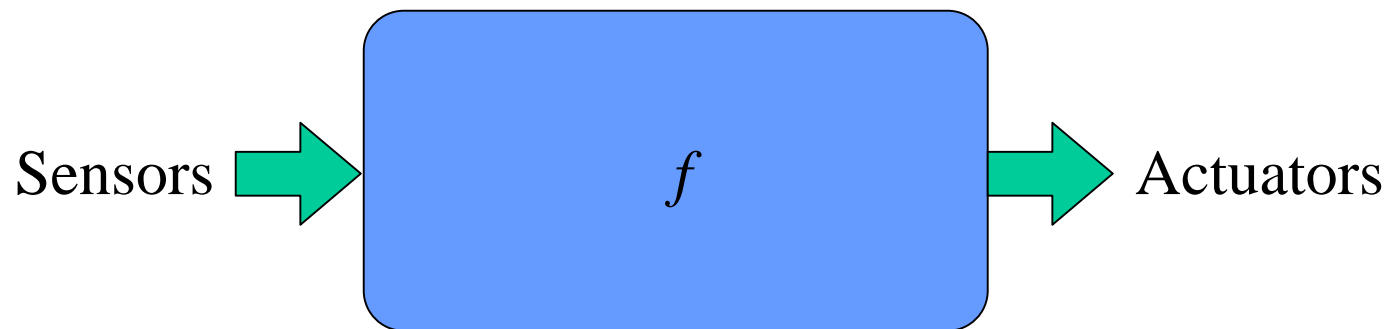
present S then P else Q

Esterel: Controller

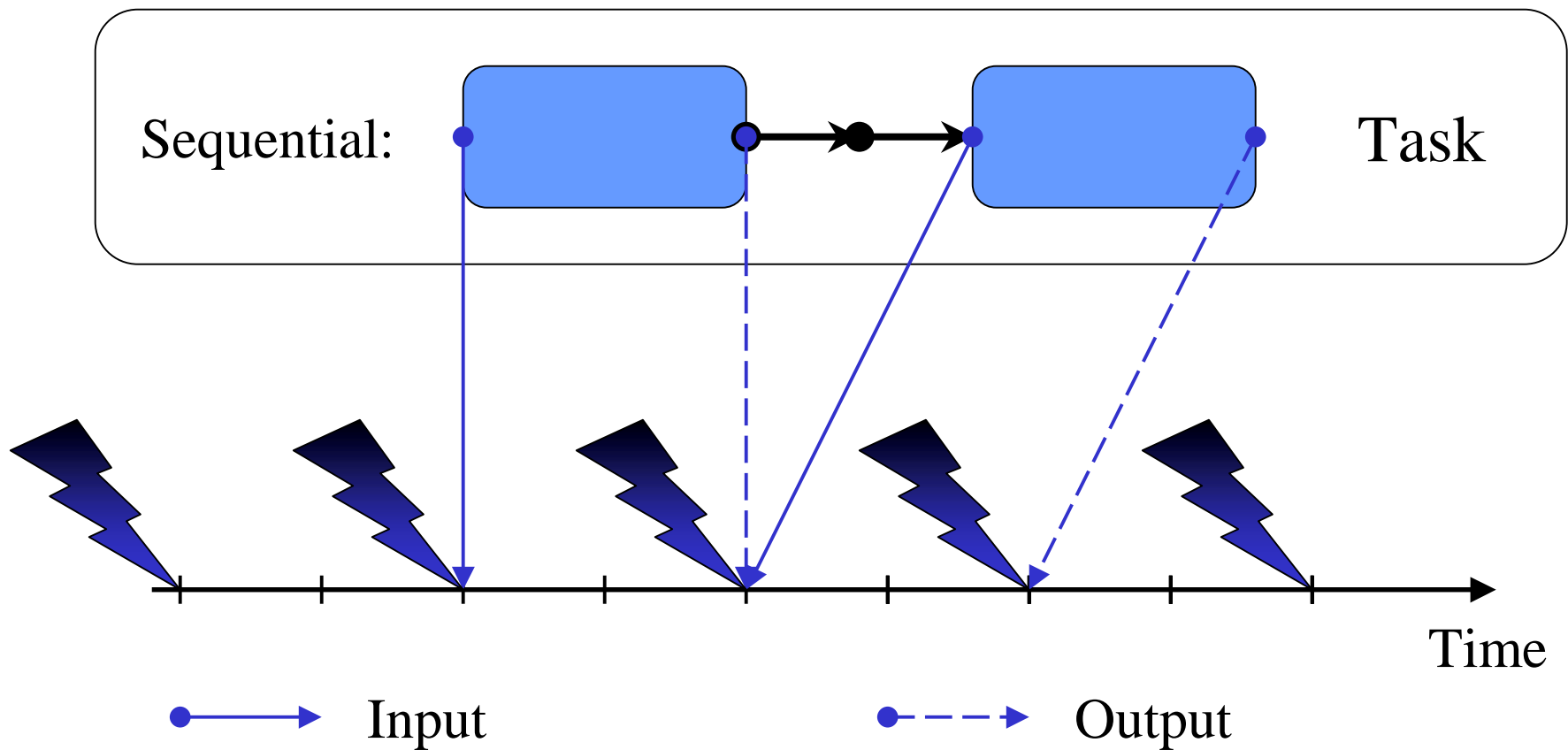
```
module normal:  
  input A, B, R;  
  output O;  
  loop  
    [ await A || await B ];  
    emit O  
  each R  
end module
```



Sensor - Control Law - Actuator

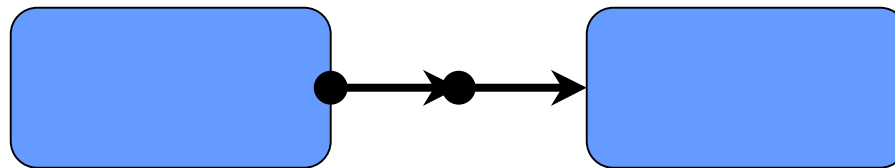


Giotto: Time



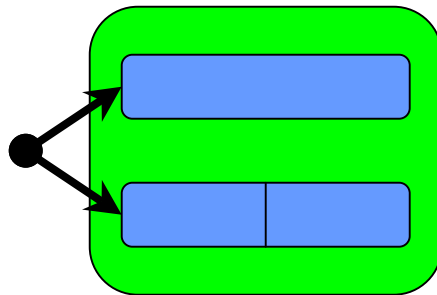
Giotto: Operators

Sequential:



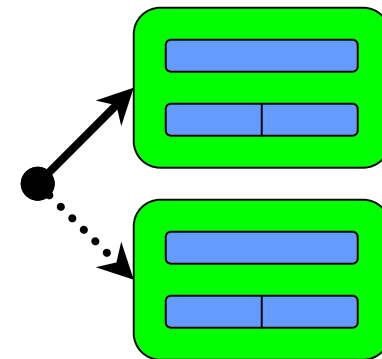
Task

Parallel:



Mode

Choice:

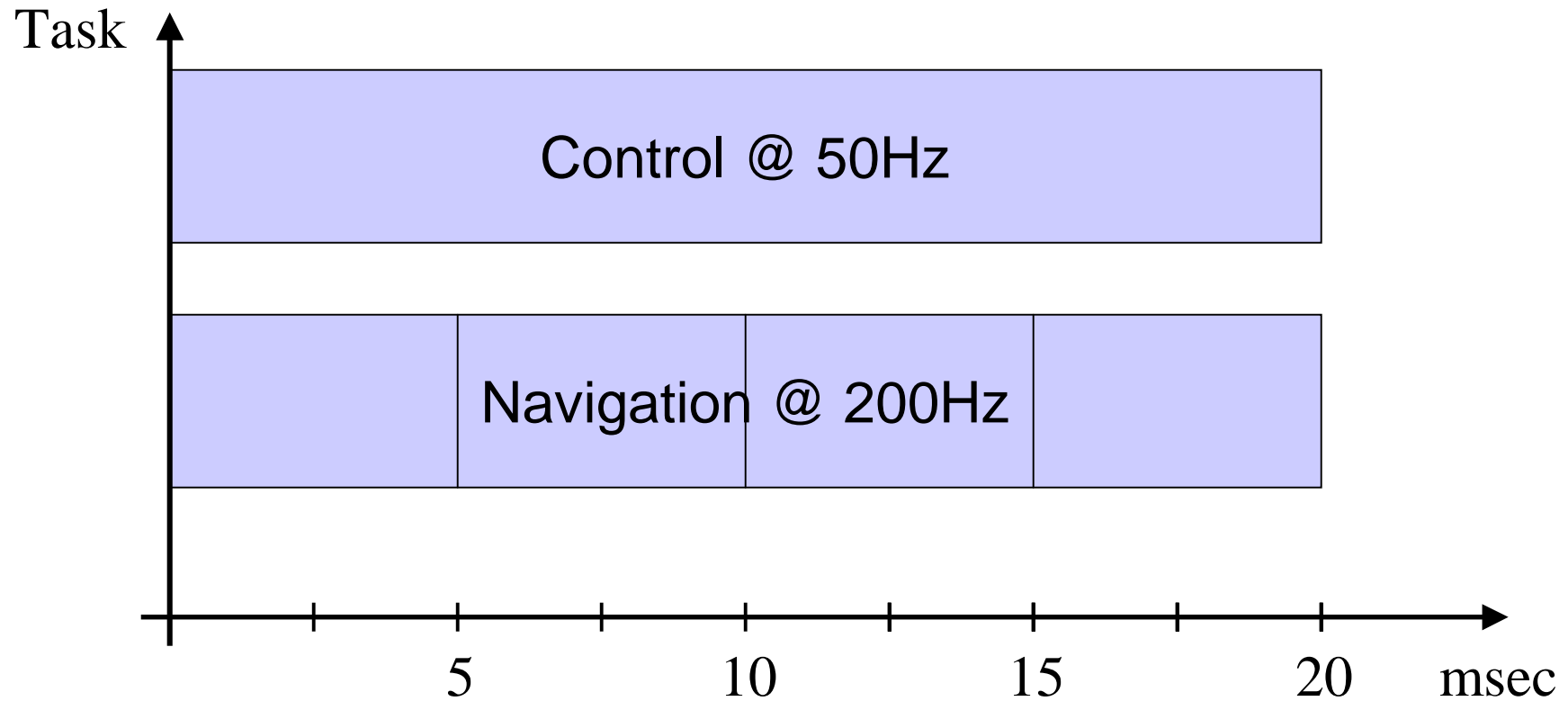


Program

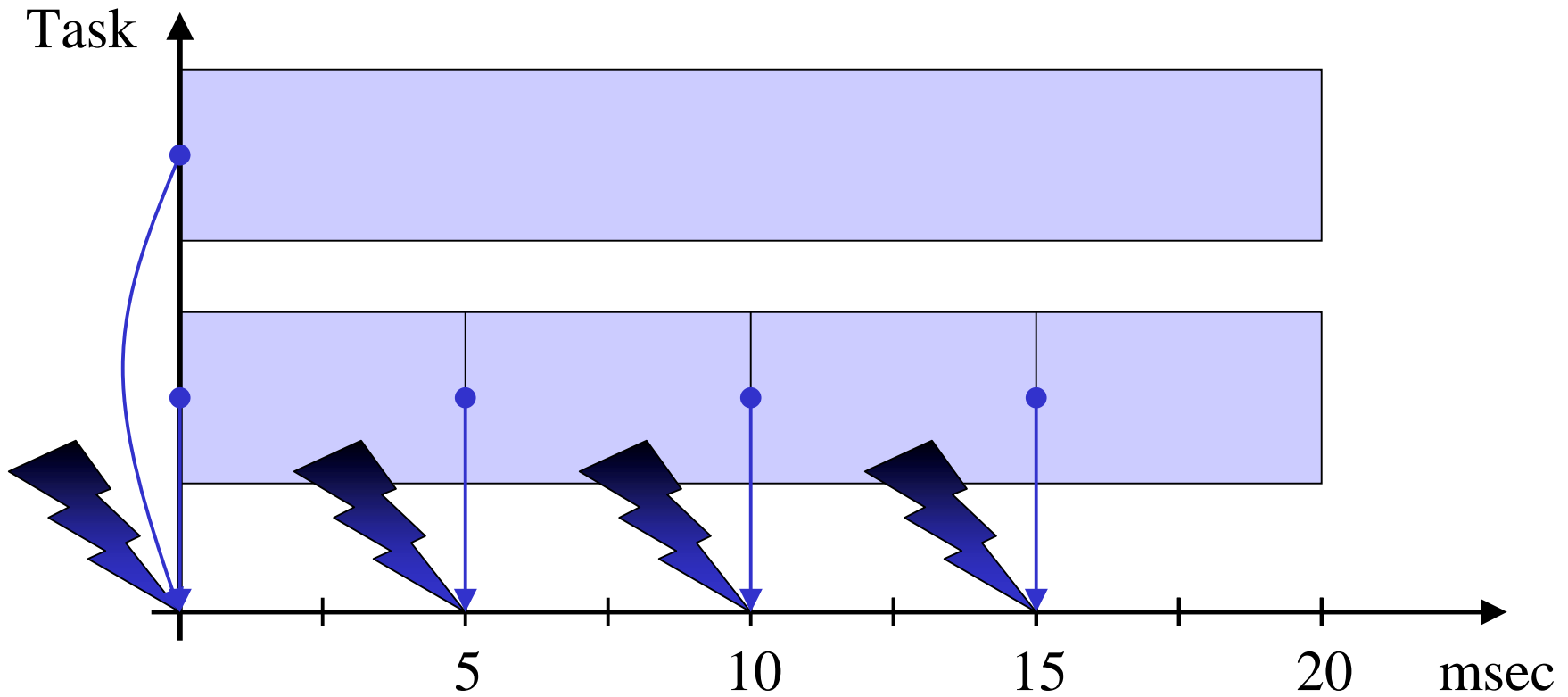
Giotto: Helicopter Control

```
mode normal ( ) period 20ms
{
    taskfreq 1 do servo = Control ( position ) ;
    taskfreq 4 do position = Navigation ( GPS, position ) ;
}
```

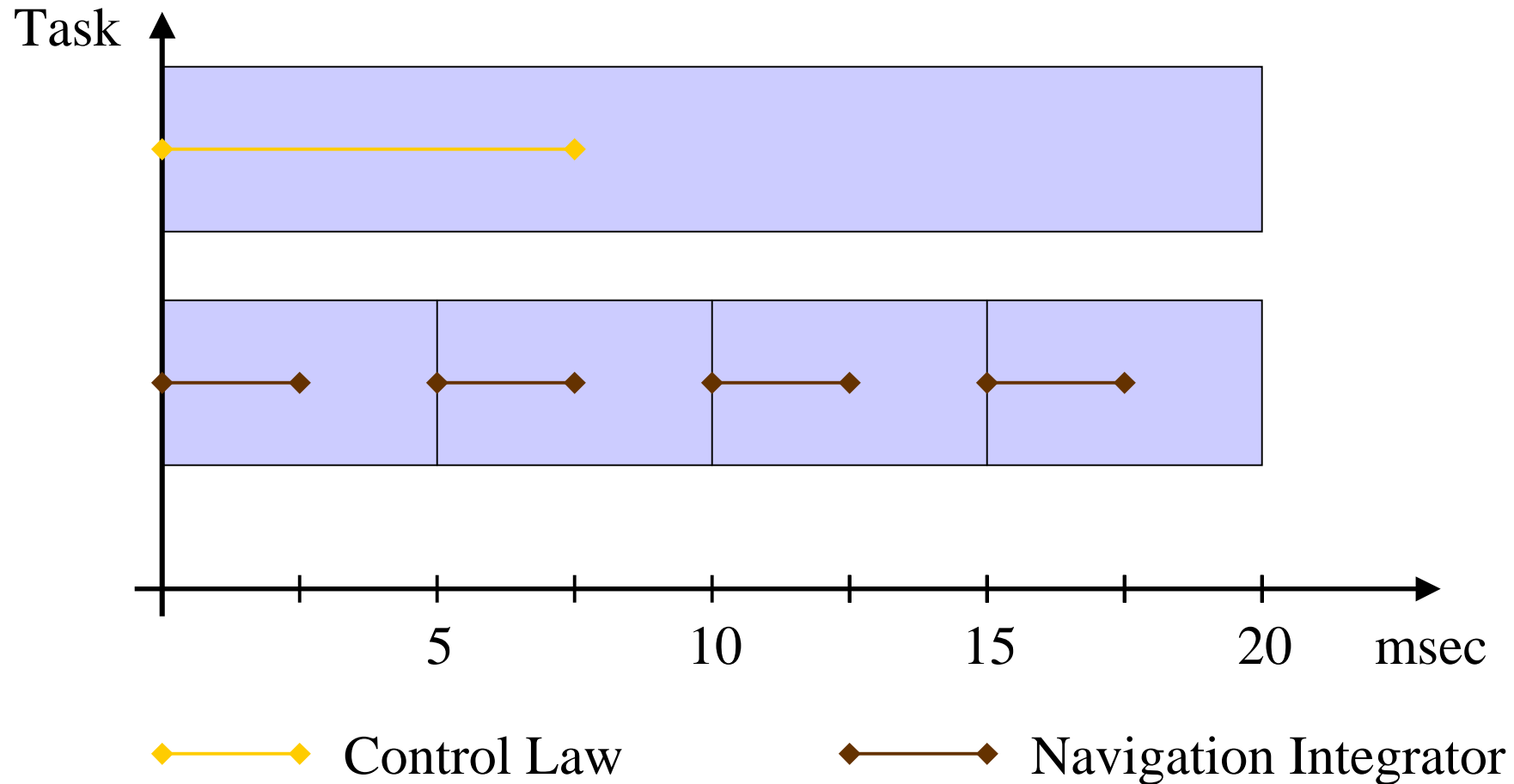
Semantics



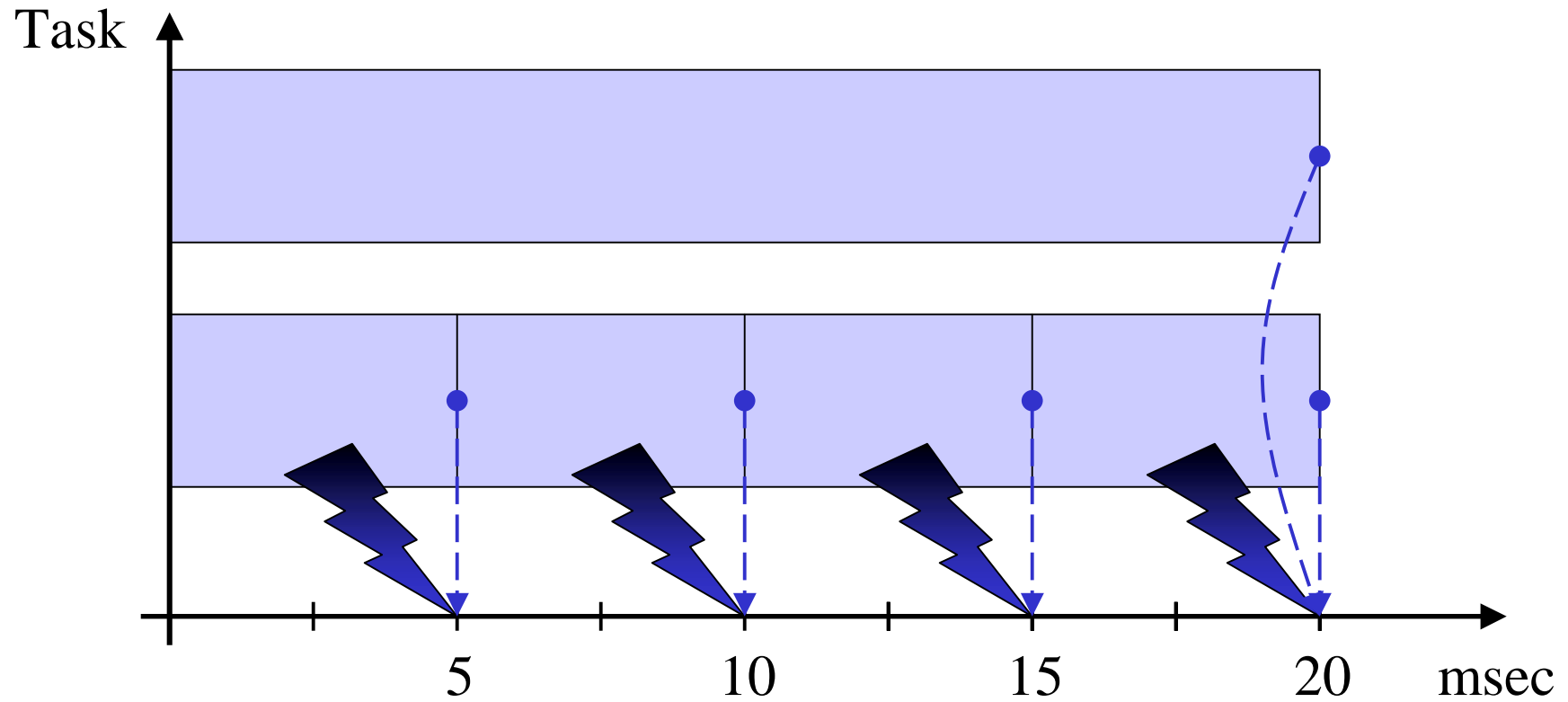
Input



Computation



Output



Literature

- Esterel:
 - The Foundations of Esterel. G. Berry. In Proof, Language and Interaction: Essays in Honour of Robin Milner. G. Plotkin, C. Stirling and M. Tofte, editors. MIT Press, 2000.
 - Synchronous programming of reactive systems. N. Halbwachs. Kluwer, 1993.
- Giotto:
 - Embedded Control Systems Development with Giotto. B. Horowitz, T. Henzinger, C. Kirsch. 2001.
 - Giotto: A Time-Triggered Language for Embedded Programming. B. Horowitz, T. Henzinger, C. Kirsch. 2001.

Embedded Programming

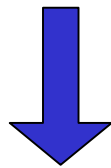
...requires the **integration** of:

1. Real-time operating system concepts
2. Embedded programming languages
3. **Embedded compilers**
4. SE, modeling, and simulation techniques
5. Formal methods

Concurrency

Parallel Composition

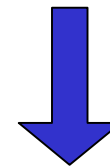
Task1 || Task2



Task1 ; Task2
Task2 ; Task1

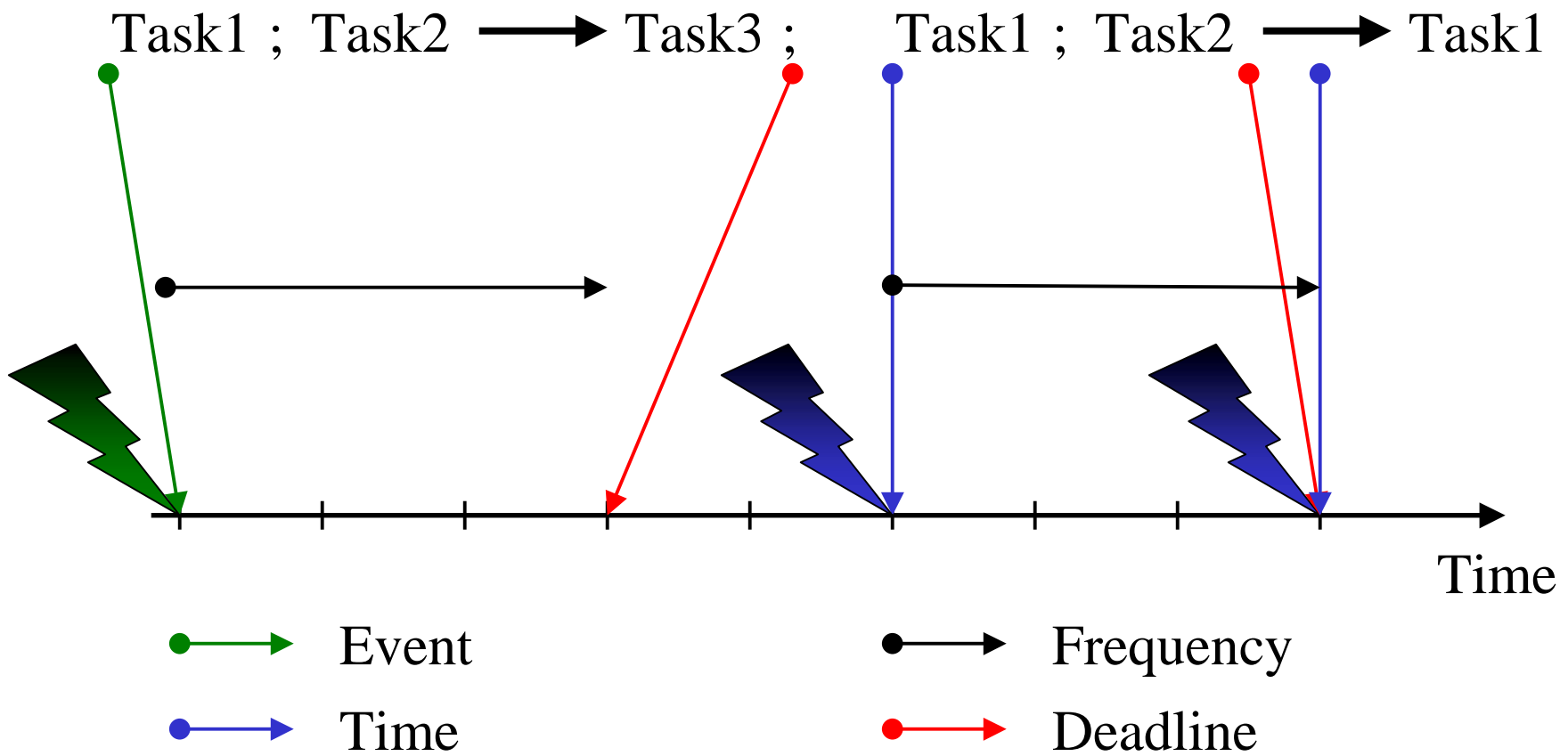
I/O Decomposition

Task1 ↔ Task2

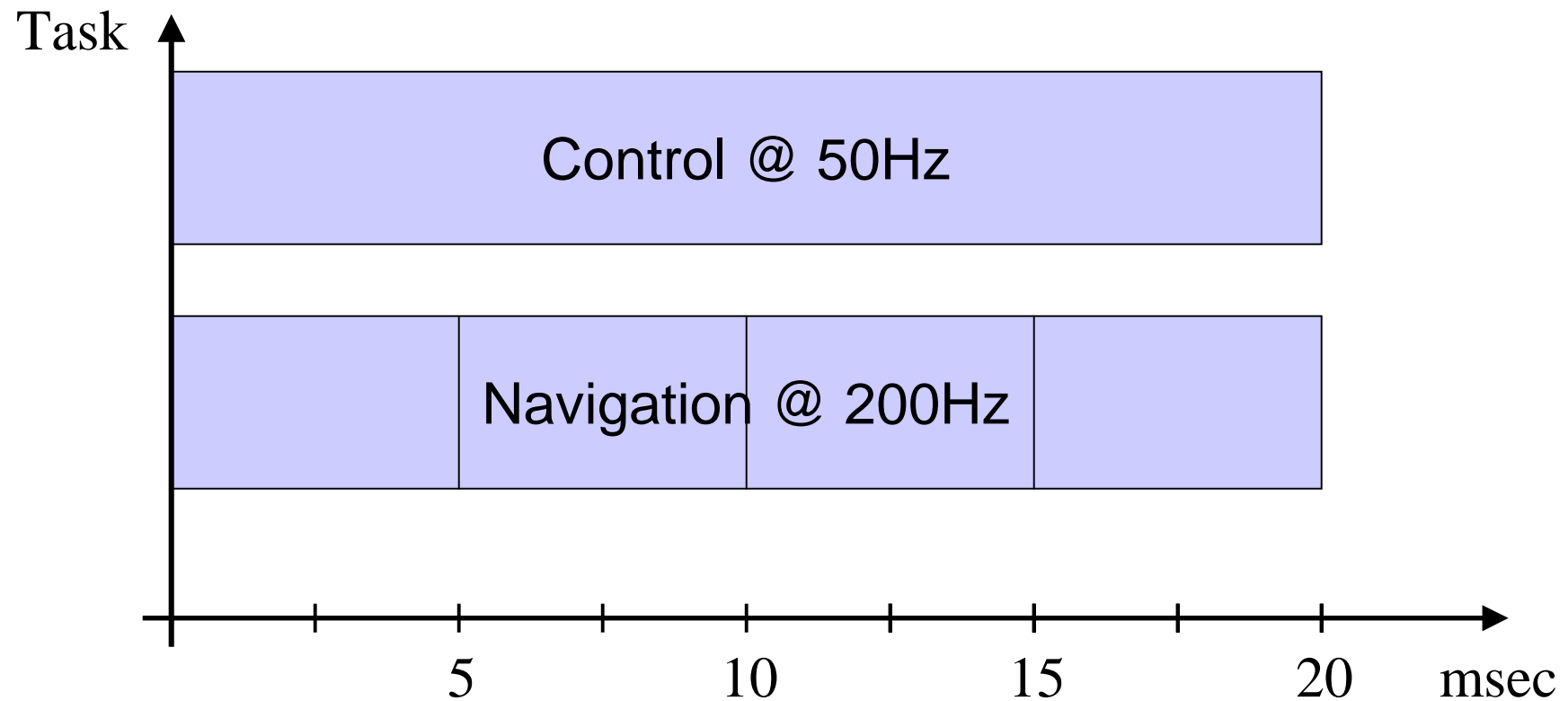


Task1 → Task2
Task2 → Task1

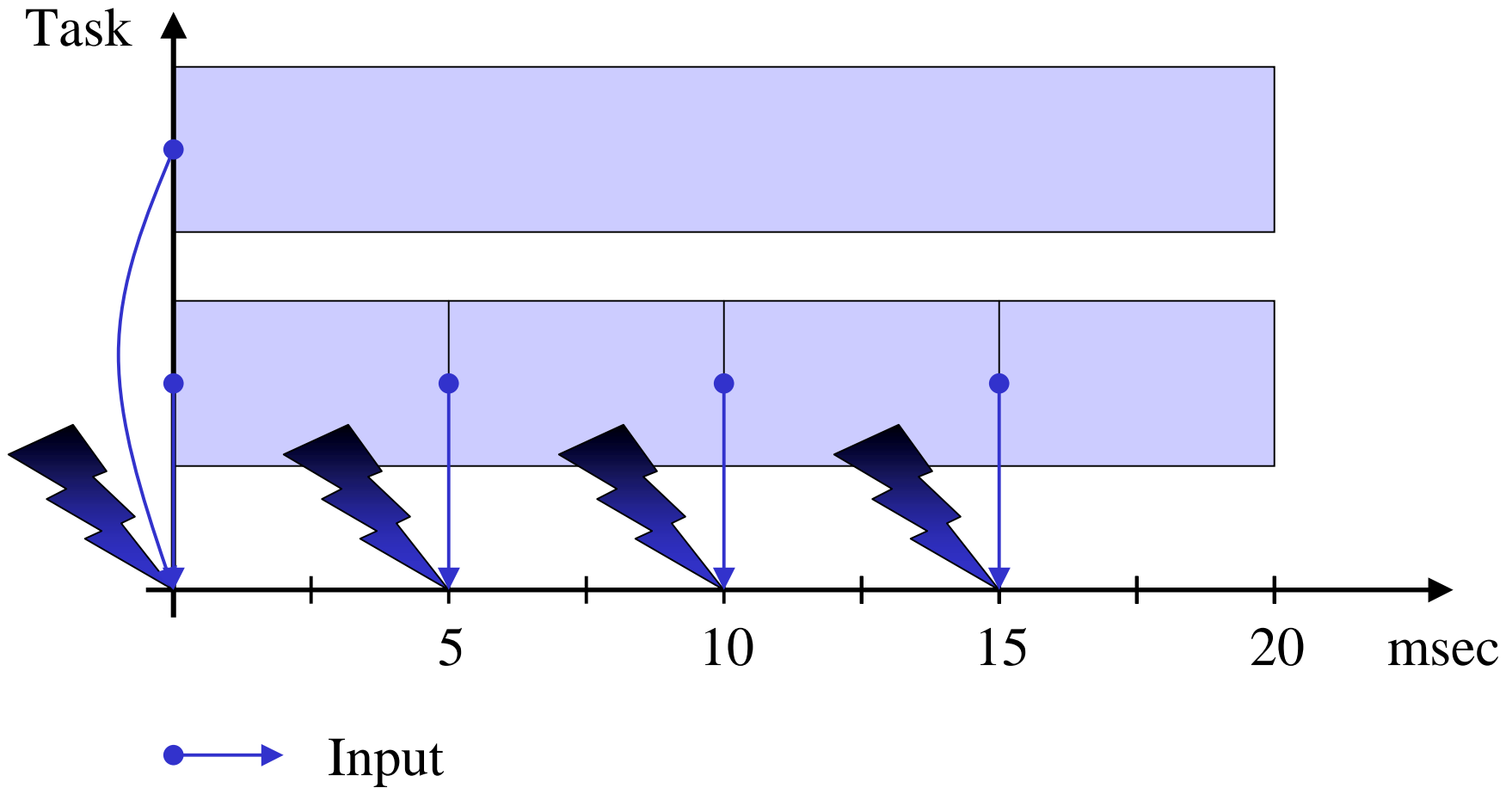
Real-Time



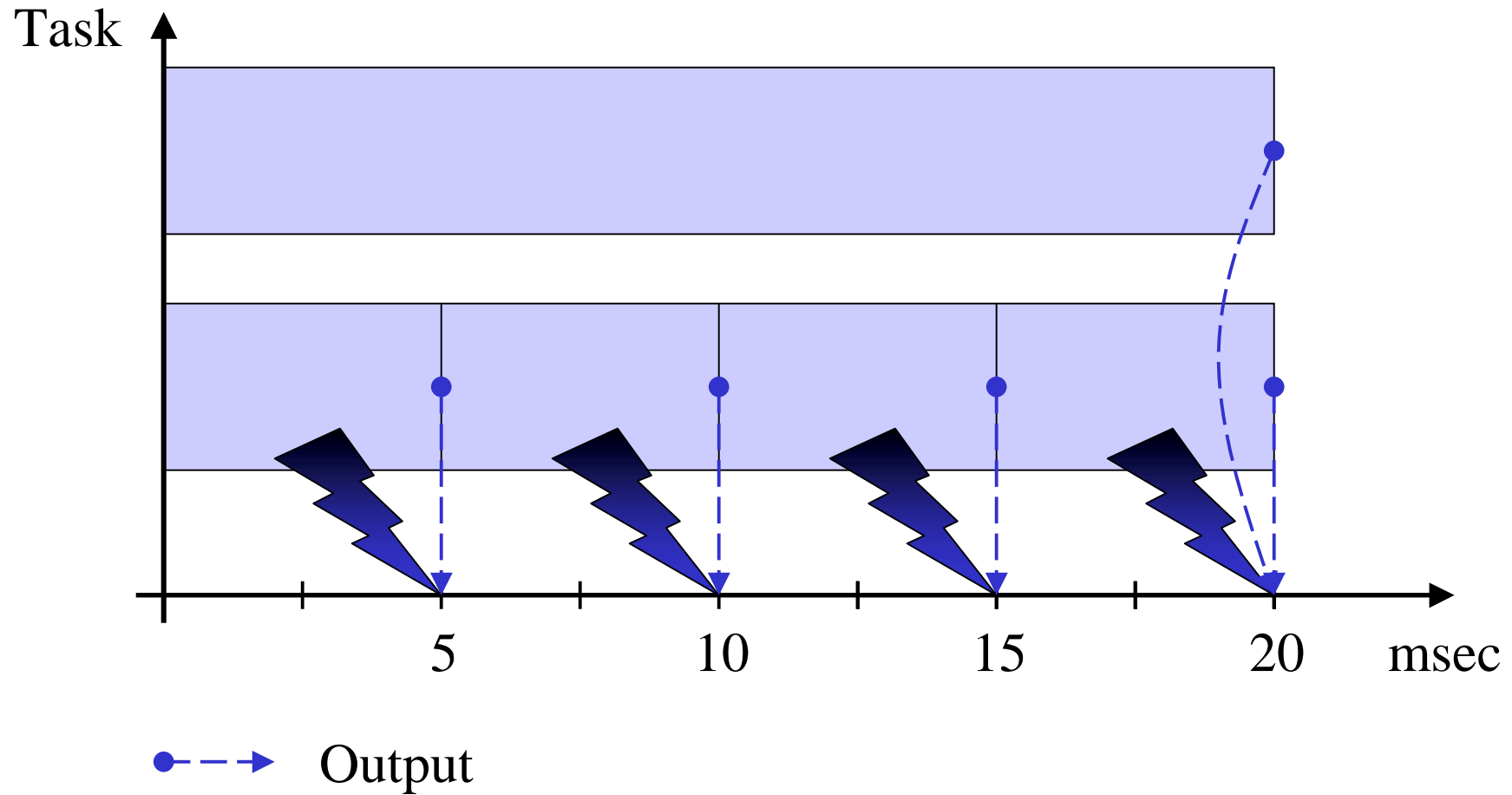
Helicopter Control



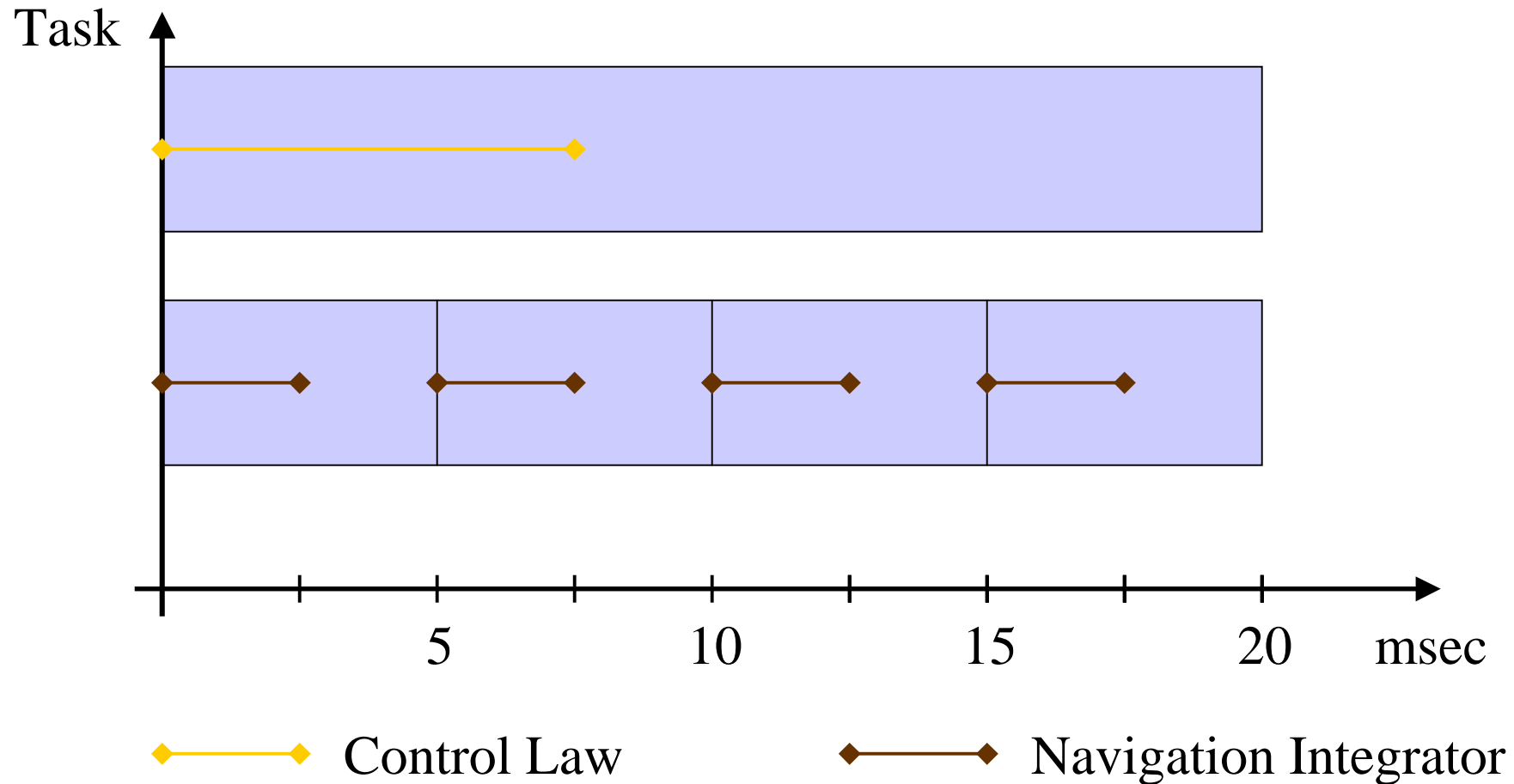
Read



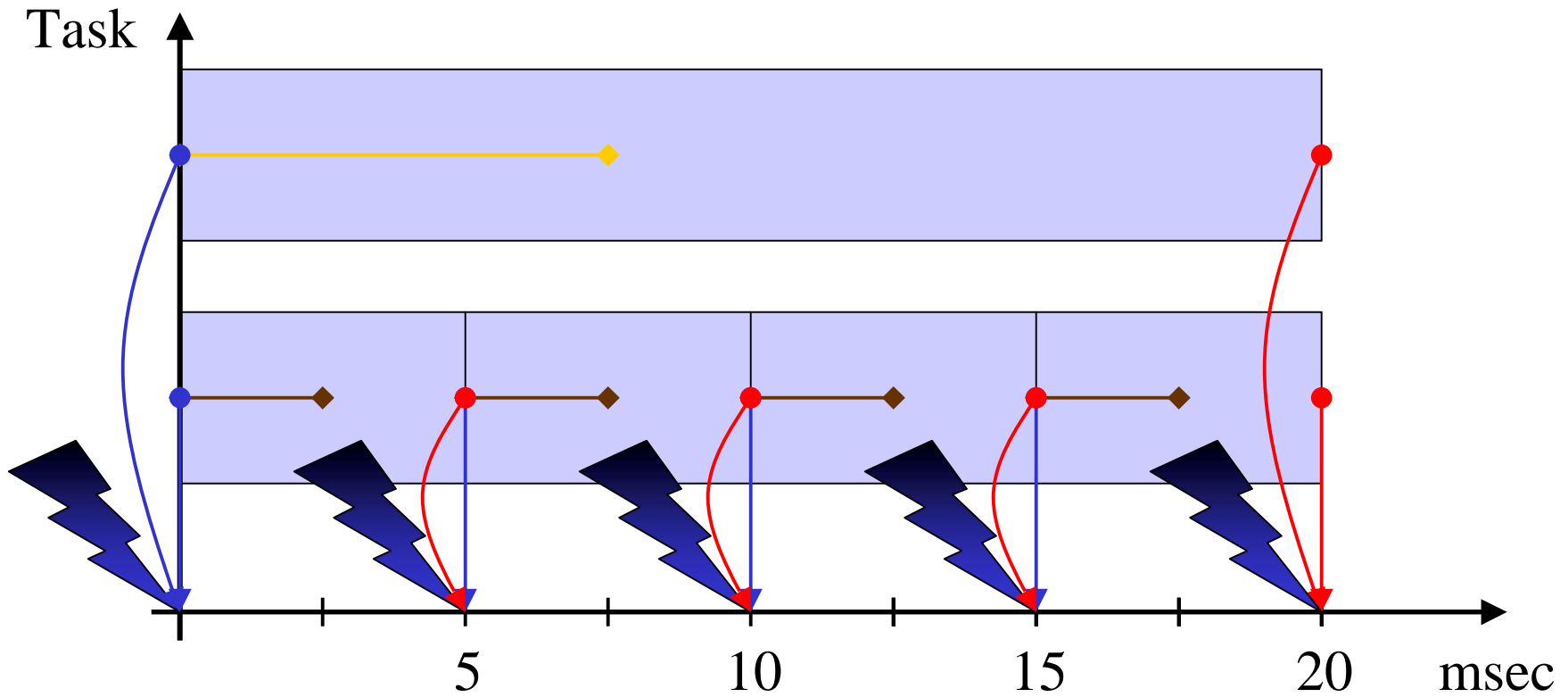
Write



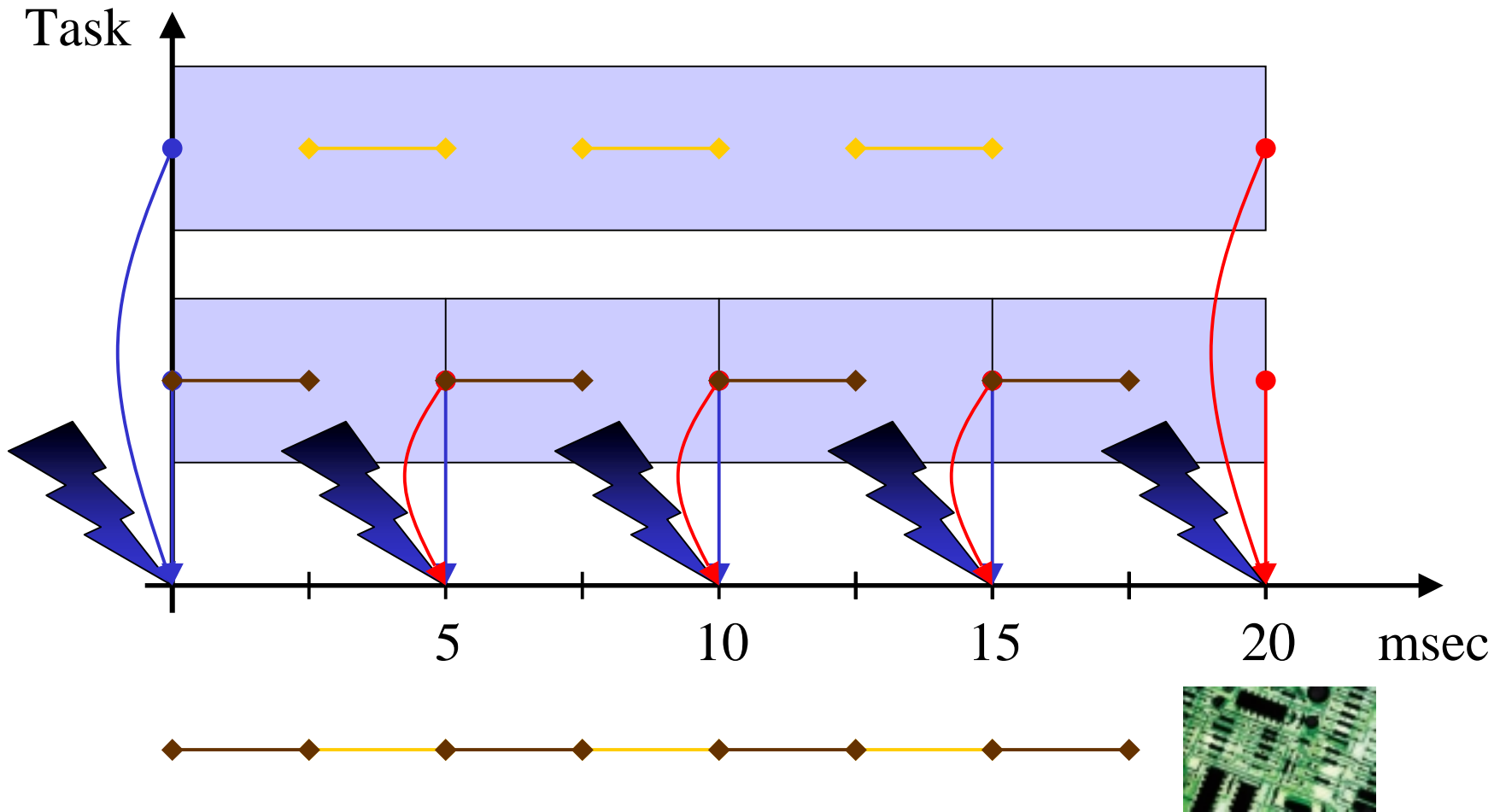
Worst Case Execution Time



Deadline



Code



Literature

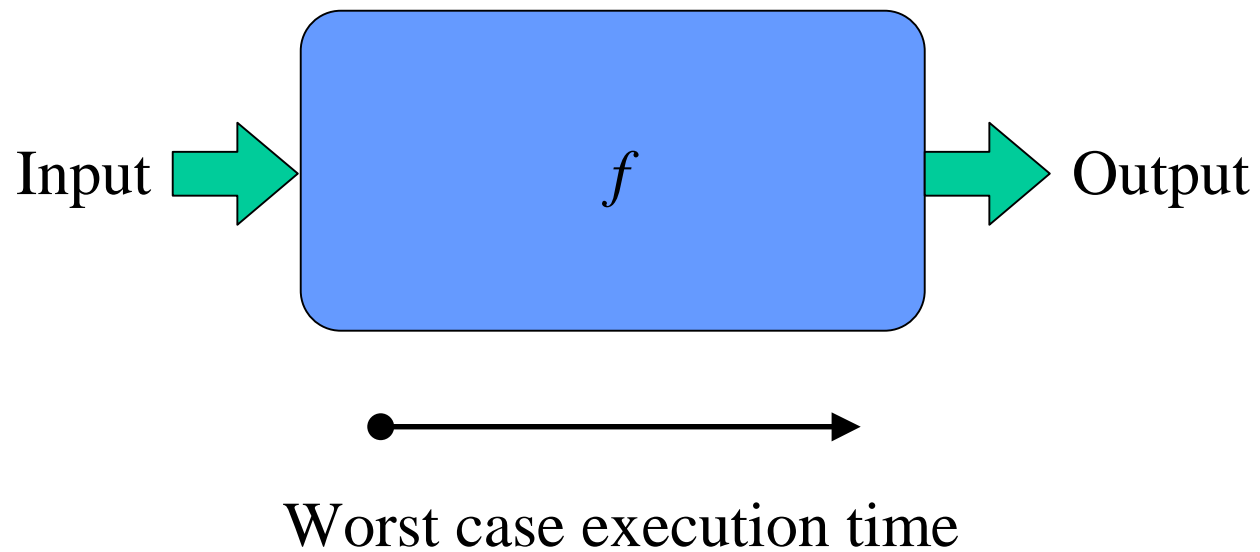
- Some compiler books:
 - Compiler Construction, N. Wirth, Addison-Wesley, 1996.
 - Compilers, Principles, Techniques, and Tools. A.V. Aho, R. Sethi, J.D. Ullman. Addison-Wesley, 1985.
 - Compiler Design. R. Wilhelm, D. Maurer. Addison-Wesley, 1995.

Embedded Programming

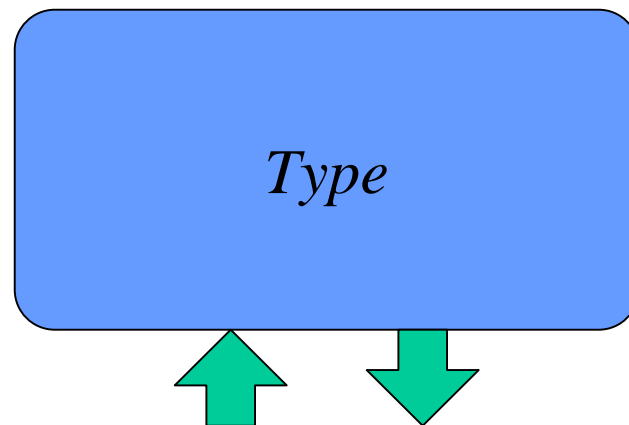
...requires the **integration** of:

1. Real-time operating system concepts
2. Embedded programming languages
3. Embedded compilers
4. **SE, modeling, and simulation techniques**
5. Formal methods

Real-Time Task

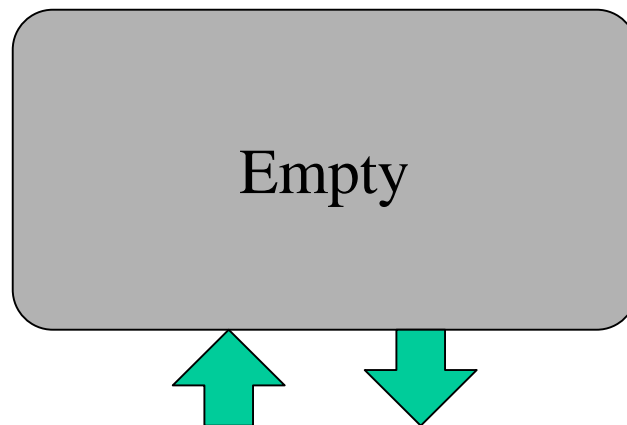


Abstract Data Type



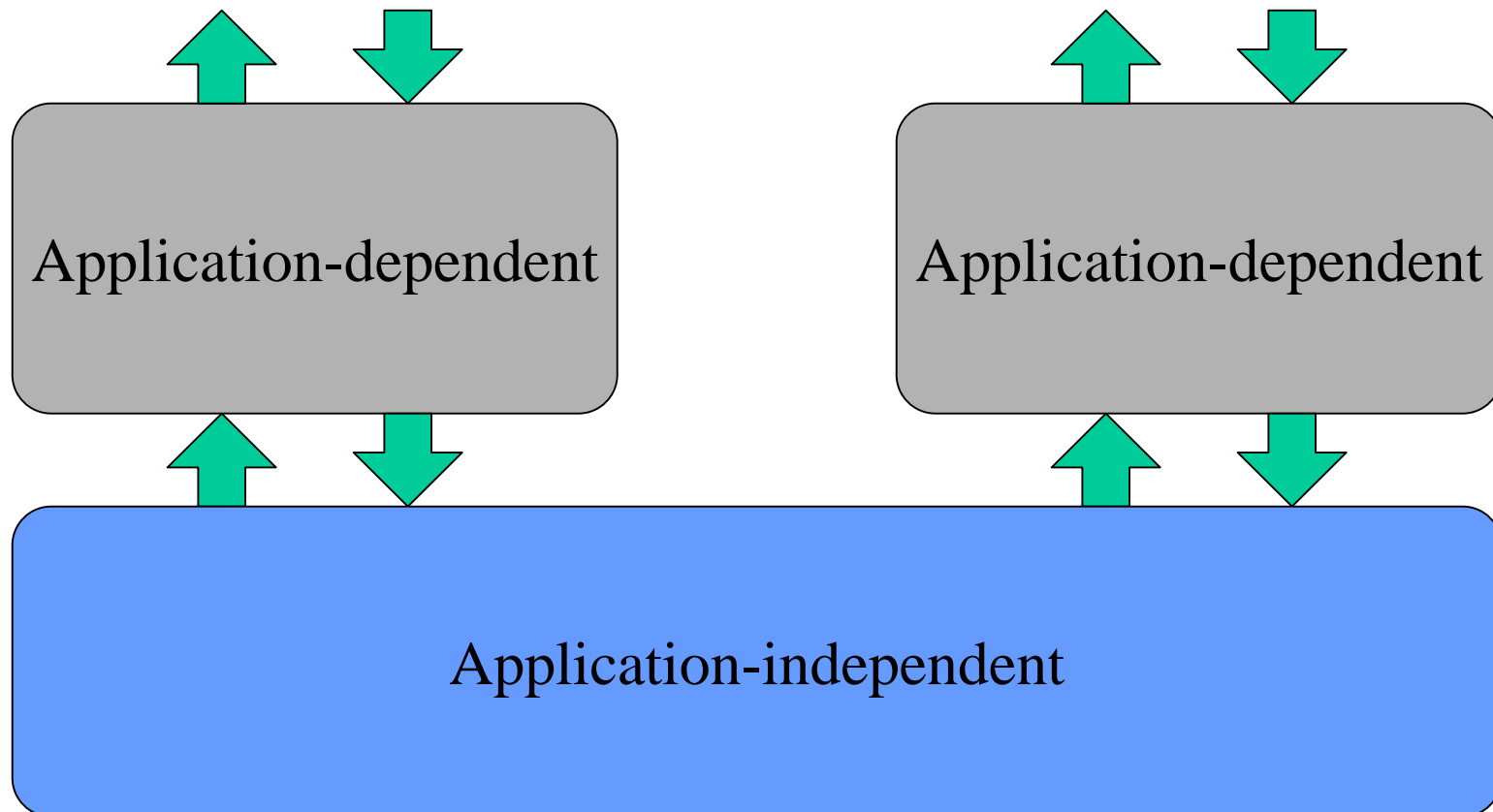
Interface: Set of methods

Abstract Interface

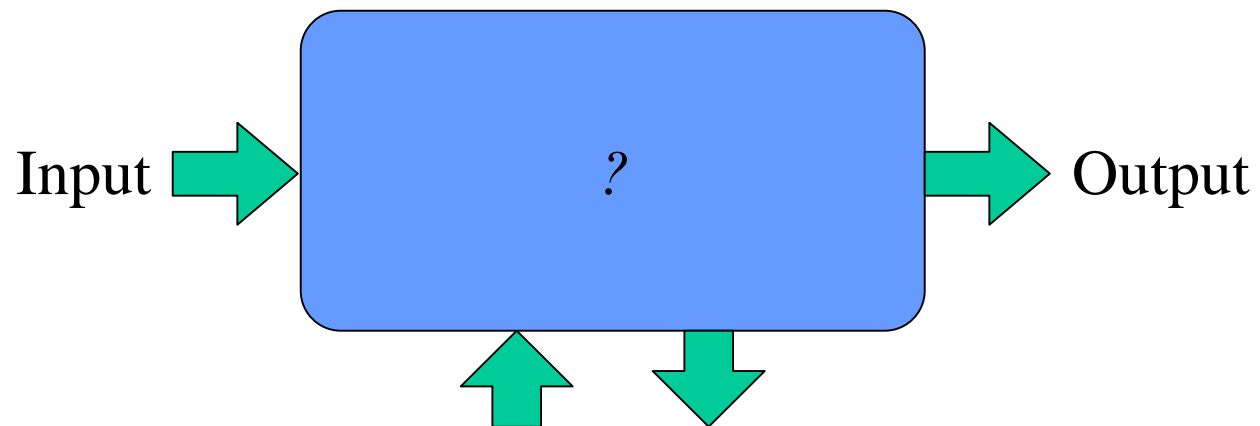


Interface: Set of methods

Framework



Type vs. Task



Interface: Set of methods

Literature

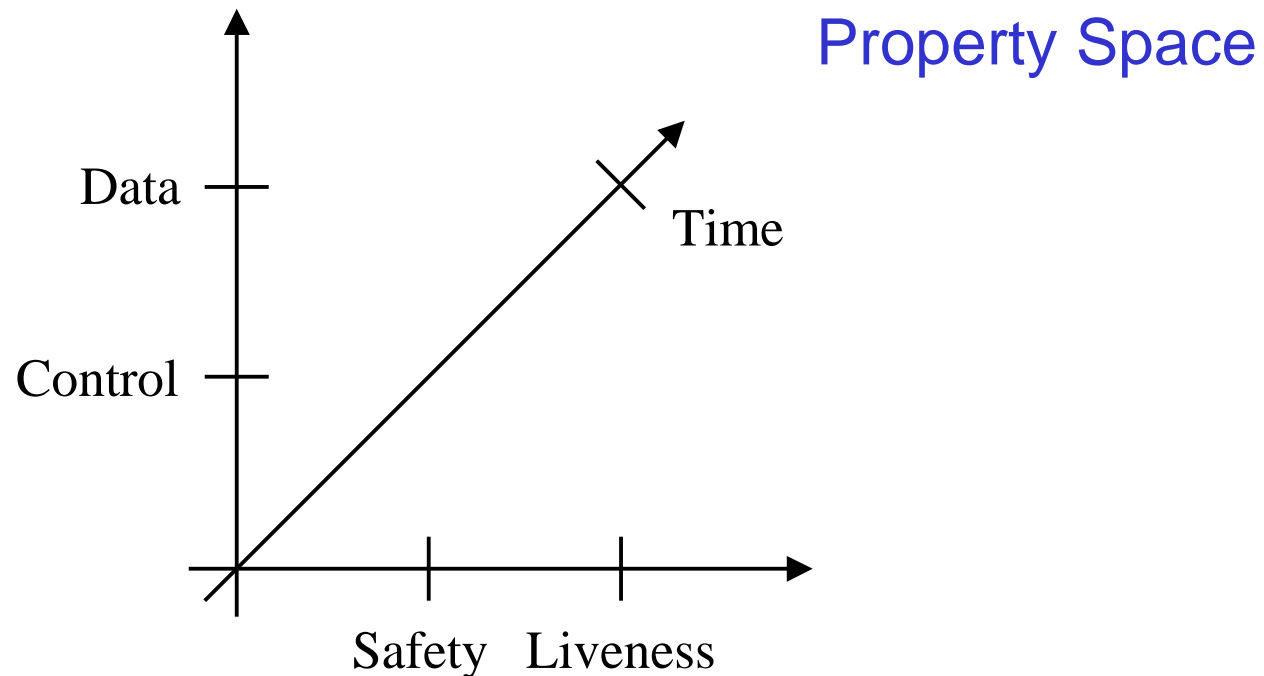
- Patterns & Frameworks:
 - Design Patterns: Elements of Reusable Object Oriented Software. E. Gamma, J. Vlissides, R. Johnson, R. Helm. Addison Wesley, 1994.
 - Design Patterns for Object-Oriented Software Development. W. Pree, E. Gamma. Addison Wesley, 1995.

Embedded Programming

...requires the **integration** of:

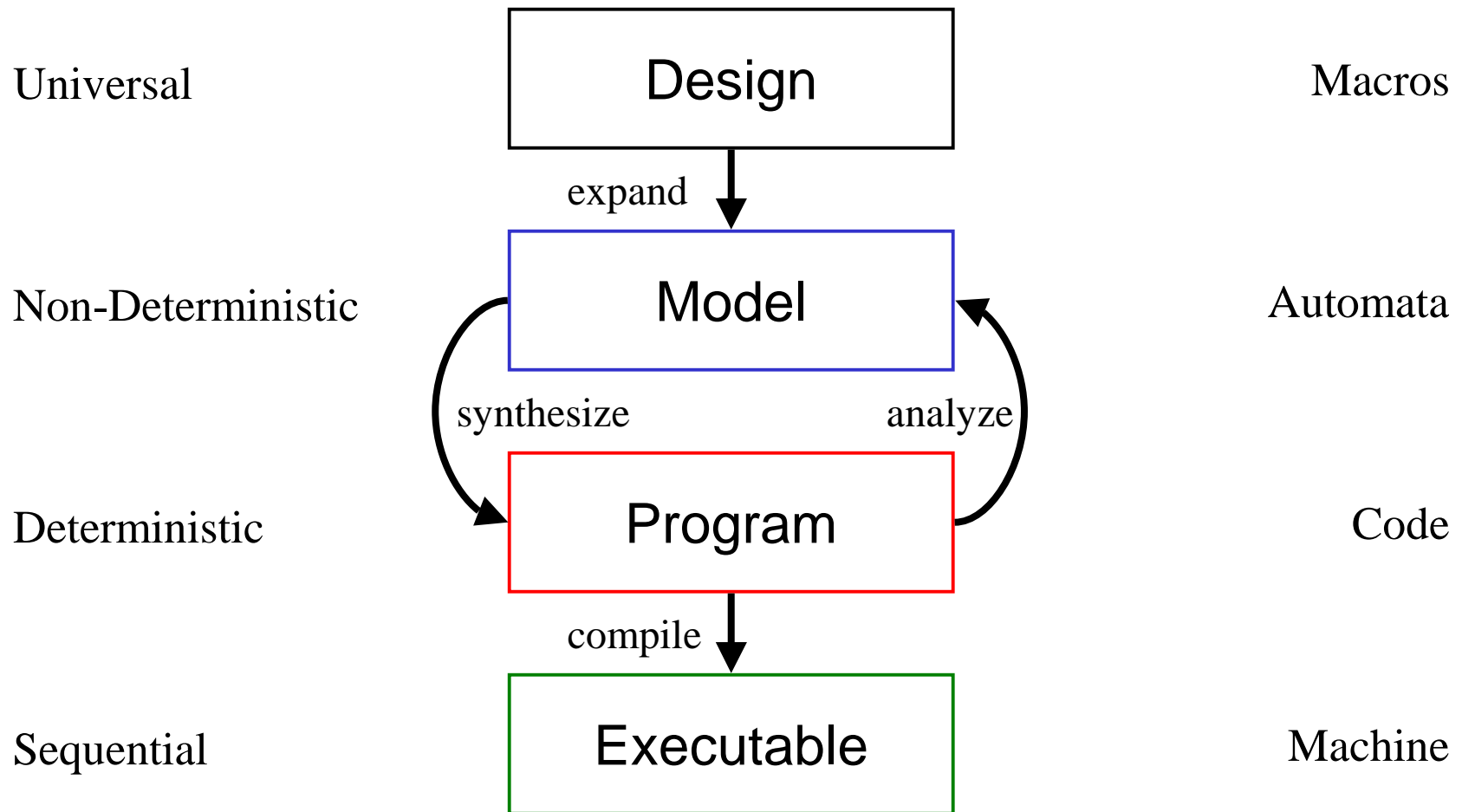
1. Real-time operating system concepts
2. Embedded programming languages
3. Embedded compilers
4. SE, modeling, and simulation techniques
5. **Formal methods**

Formal Verification



- Safety: Wrong things never happen!
- Liveness: Something useful will happen eventually!

Language Hierarchy

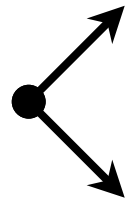


Non-Determinism

Sequential



Parallel

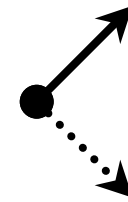


A

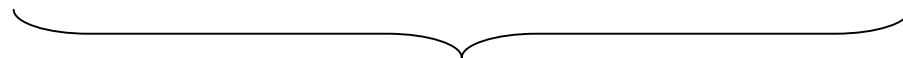
Choice



Non-Determinism



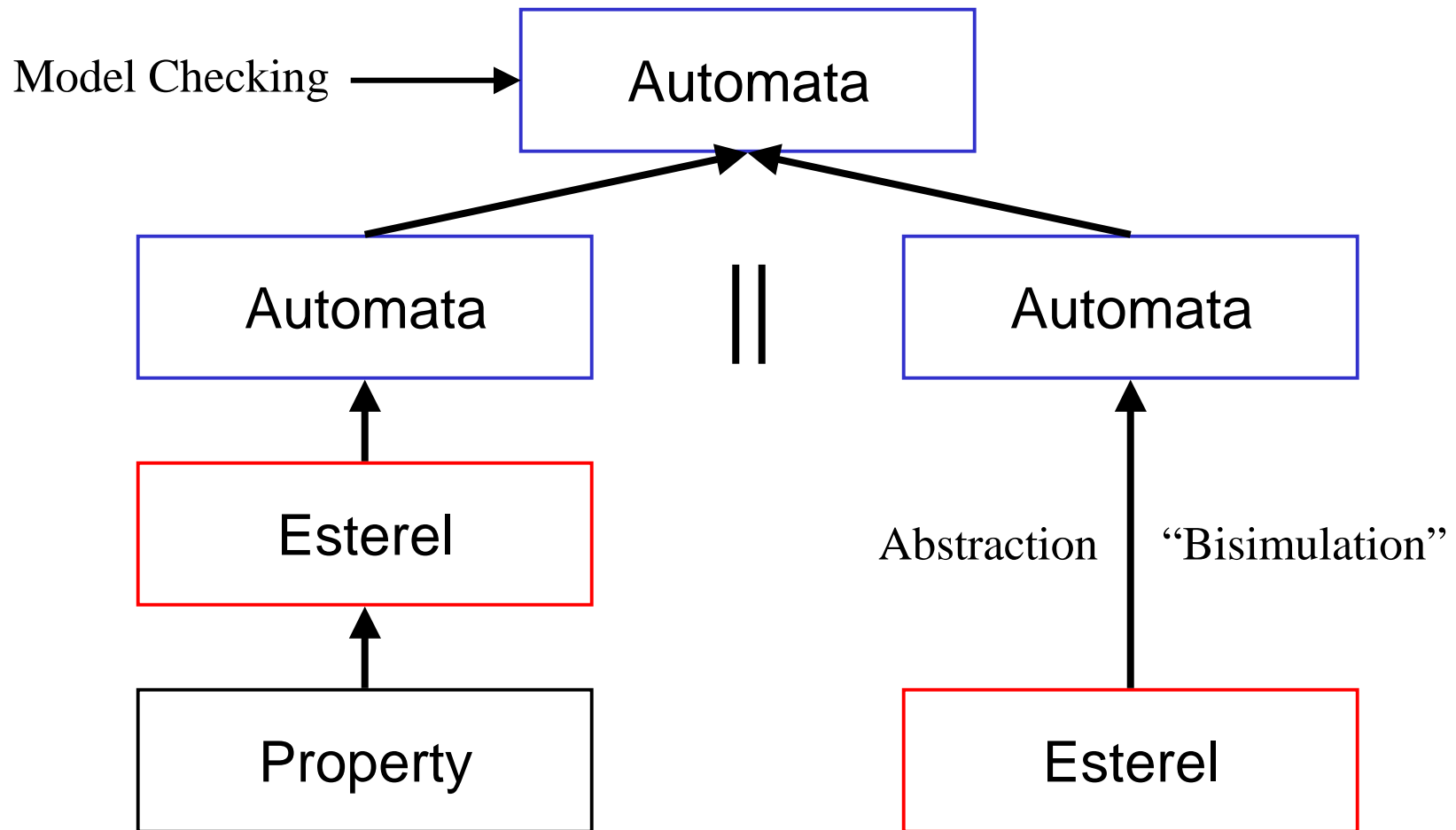
E



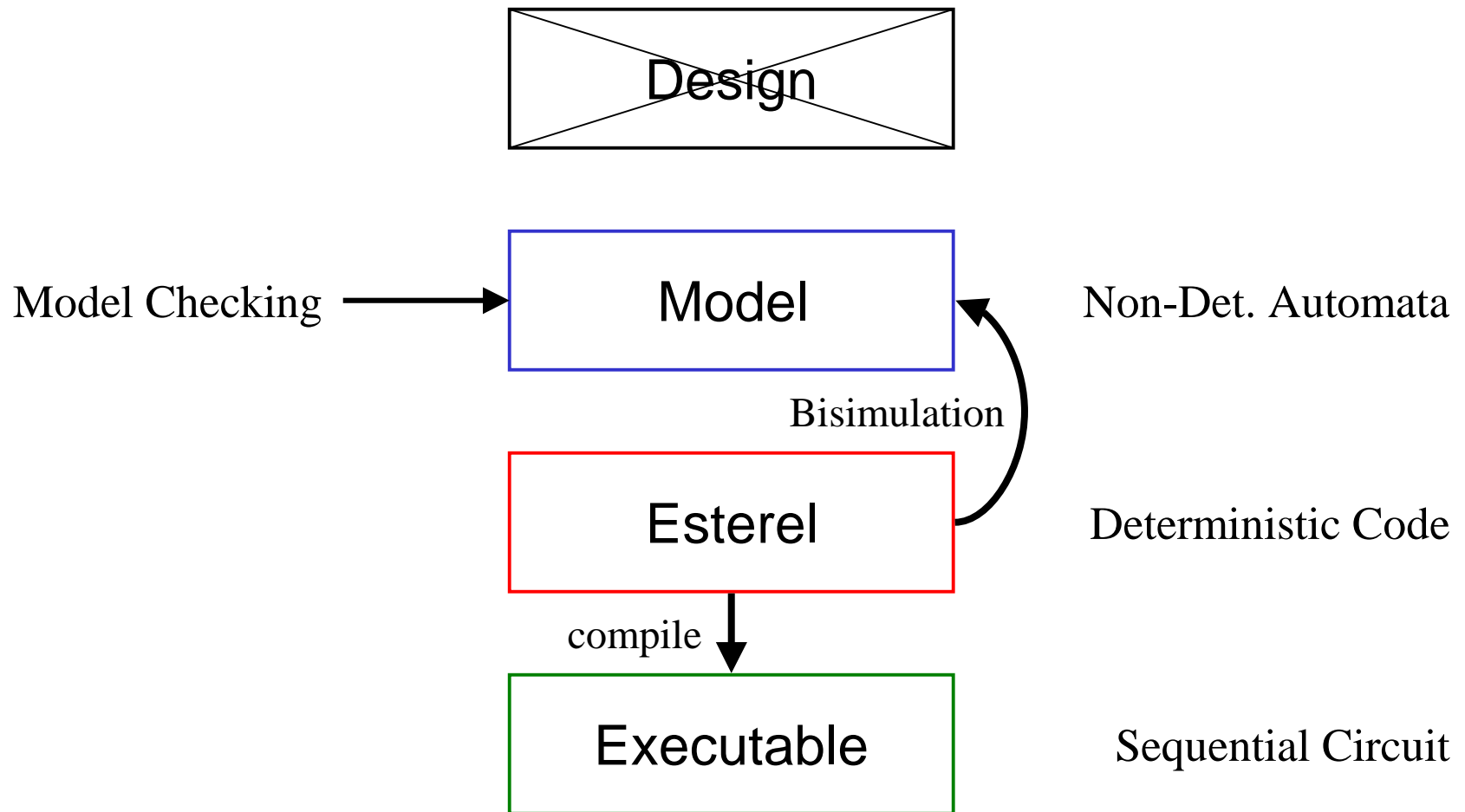
Programming Operators

Modeling Operator

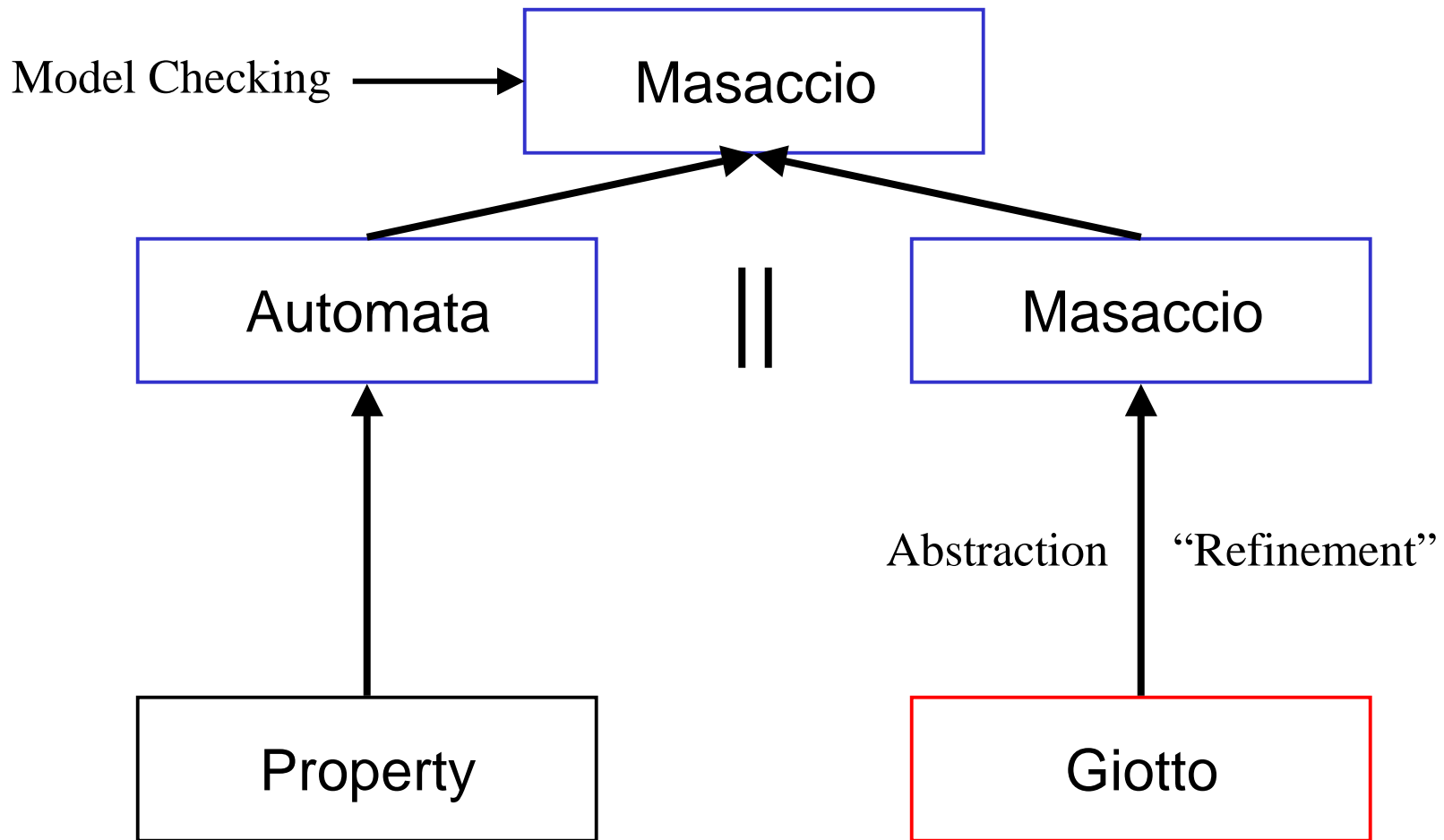
Esterel: Verification



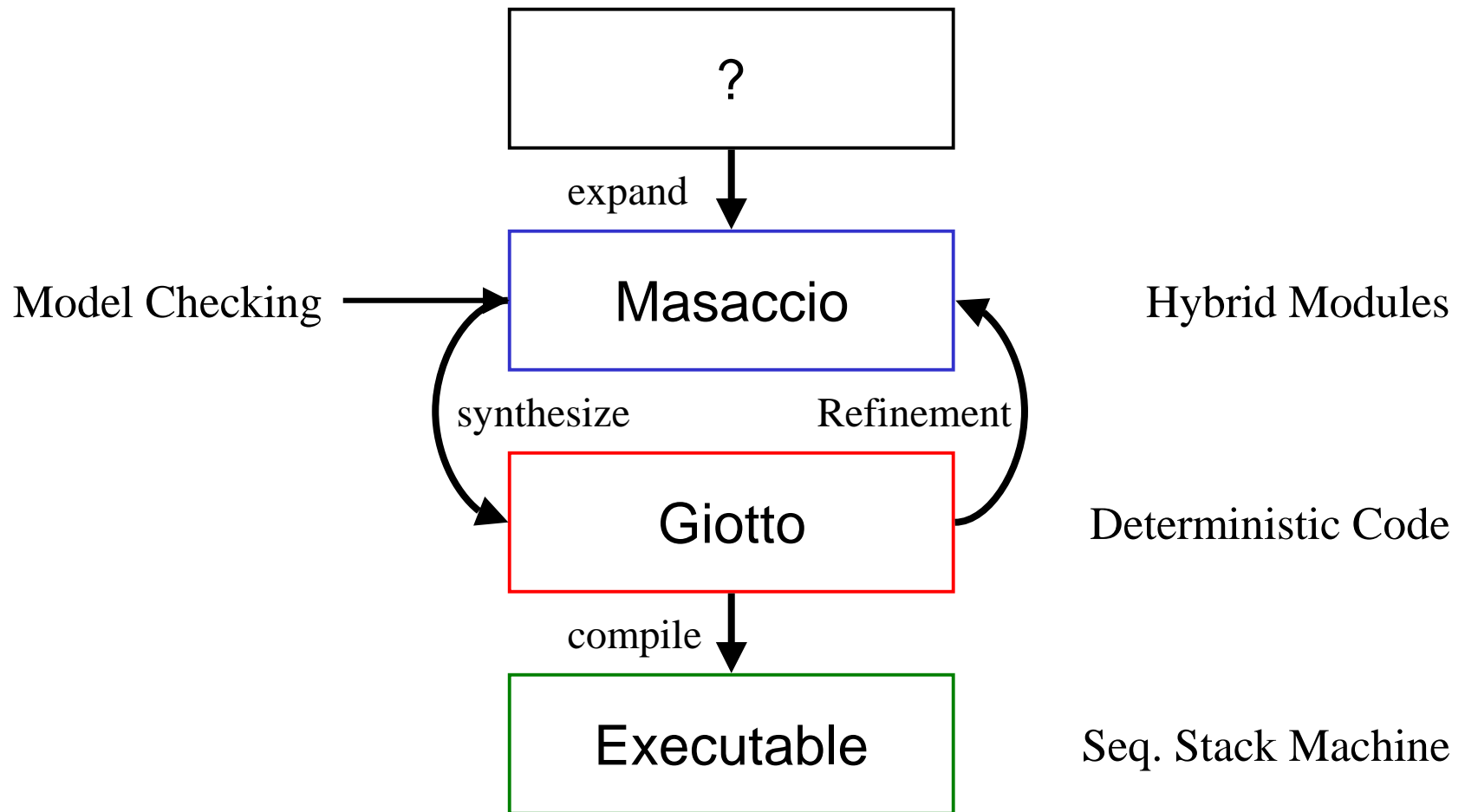
Esterel: Hierarchy



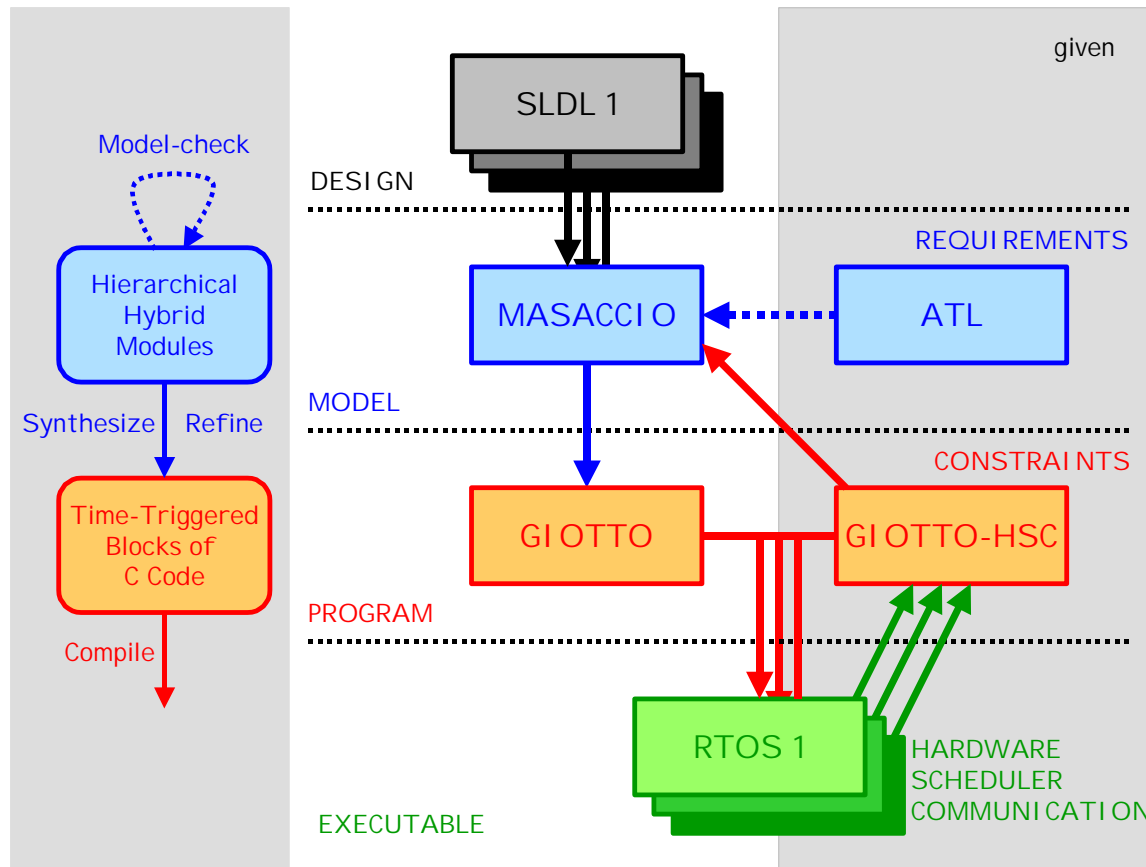
Giotto: Verification



Giotto: Hierarchy



Giotto: Hierarchy



Literature

- Esterel:
 - Papers @ www.esterel.org
- Giotto:
 - T.A. Henzinger. Masaccio: A Formal Model for Embedded Components. LNCS 1872, Springer, 2000, pp. 549-563.

End

