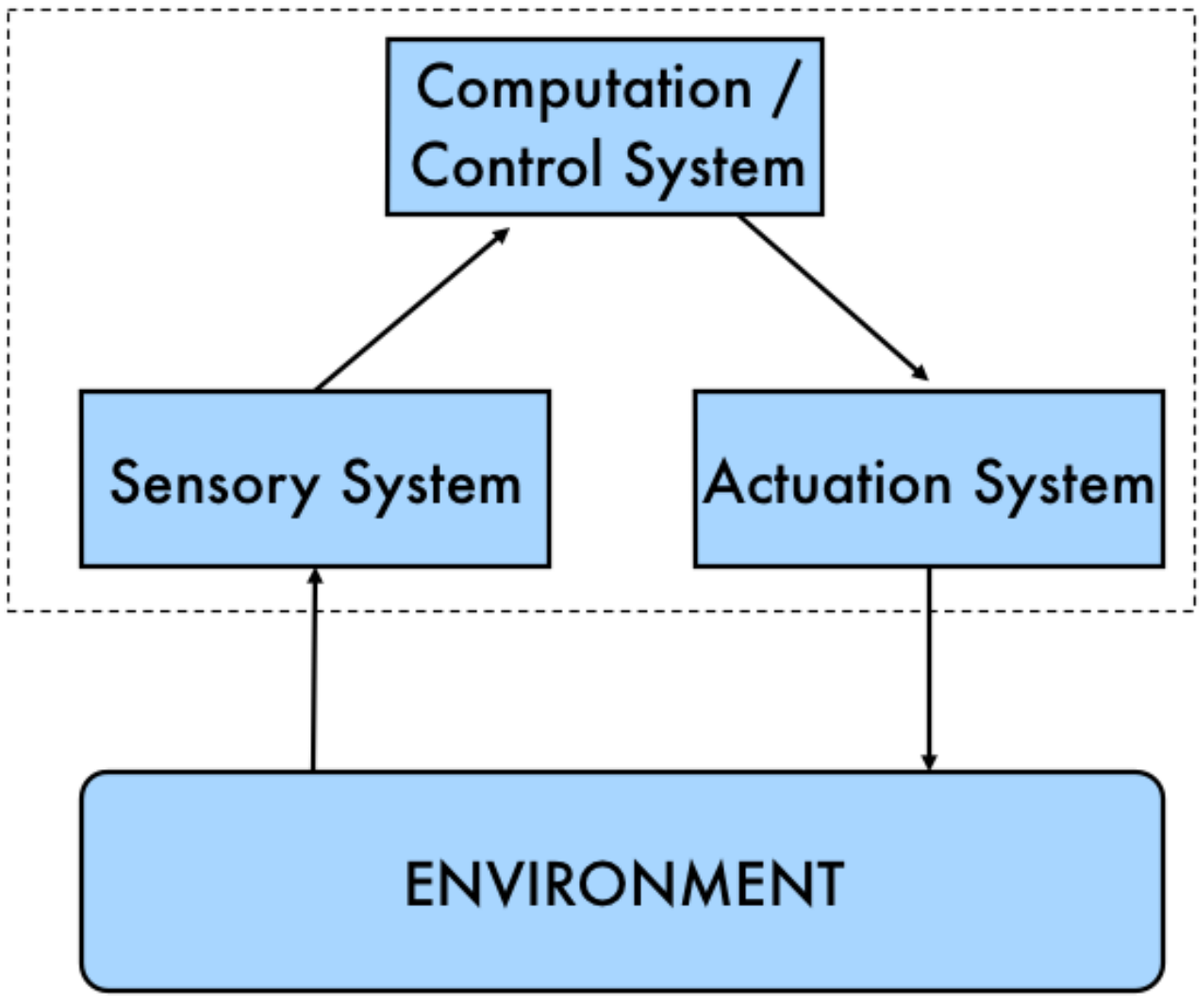
Spring 2022
ECE 568: Embedded Systems
Lecture #3: Timing Constraints

Vijay Raghunathan
vr@purdue.edu

Recap: Timing in Embedded Sys.

- Computation is in response to external events
- Interaction with environment causes problems
  - Indeterminacy in execution
    - e.g., waiting for events from multiple sources
  - Physical environment is delay intolerant
    - can't put it on wait with an hour glass icon!
- Handling timing constraints crucial to the design of embedded systems



What is an *"Event"*?

- An indicator that some *phenomenon of interest* has occurred
  - Phenomenon can be something in the physical world or in the cyber world (related to hardware or software)

- In our context, an event is usually assumed to be instantaneous; time when the event occurs is referred to as the *event occurrence time*
  - If it is not, we will usually be interested in the start point and/or end point of the event

- Periodic events: Occur at regular intervals
  - The event separation interval is called the *time period* of the periodic event
- Sporadic events: Can occur at any time (with some restriction on how fast events can occur)

- The event can represent either a stimulus (i.e., input) to the system from its environment, or is an externally observable response (i.e., output) that the system makes

Timing Constraints

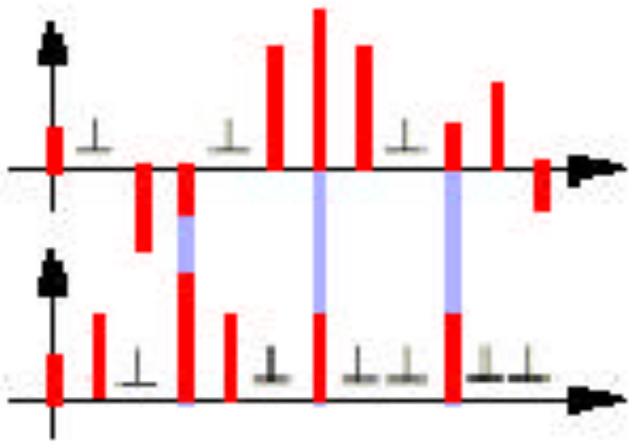
- Timing constraints impose certain temporal restrictions on a system or its environment/users

- Key questions central to embedded systems and timing constraints:
  - What kind of timing constraints are commonly encountered?
  - How do we arrange computation to take place such that it satisfies the timing constraints?

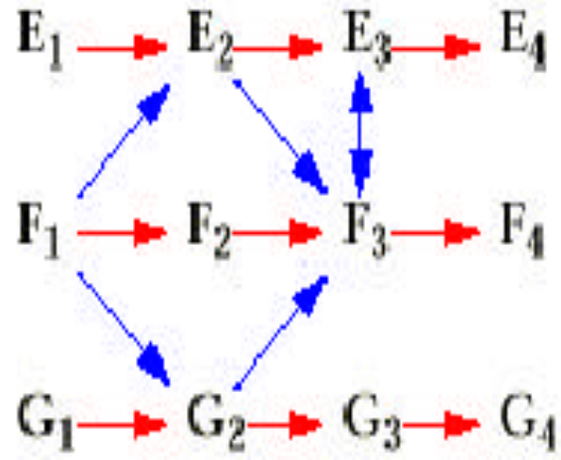
Many Notions of Time

continuous time

discrete time



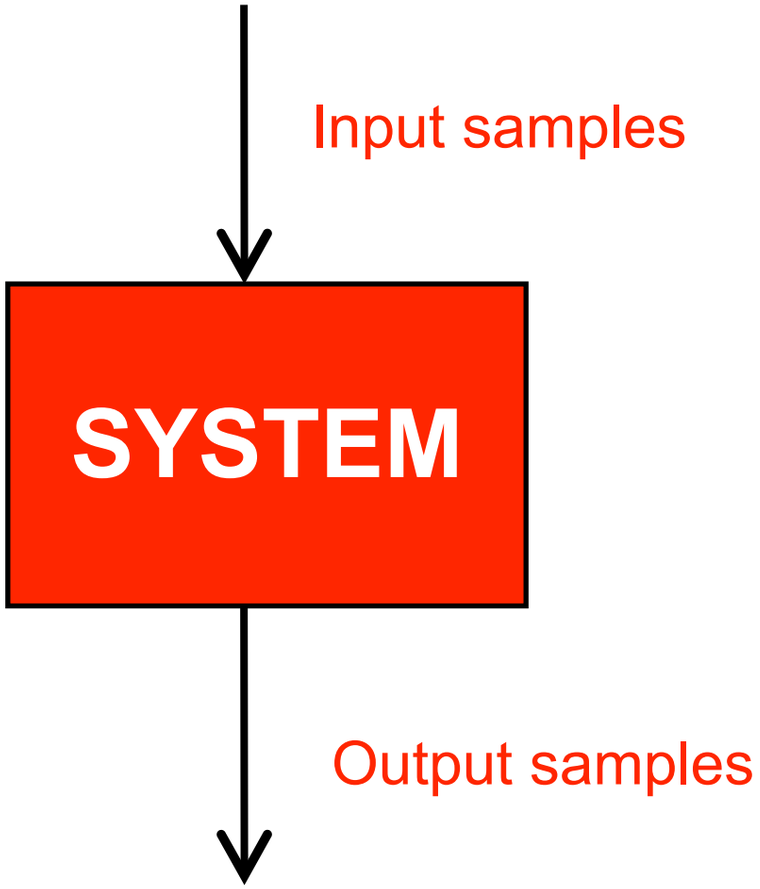
synchronous/reactive

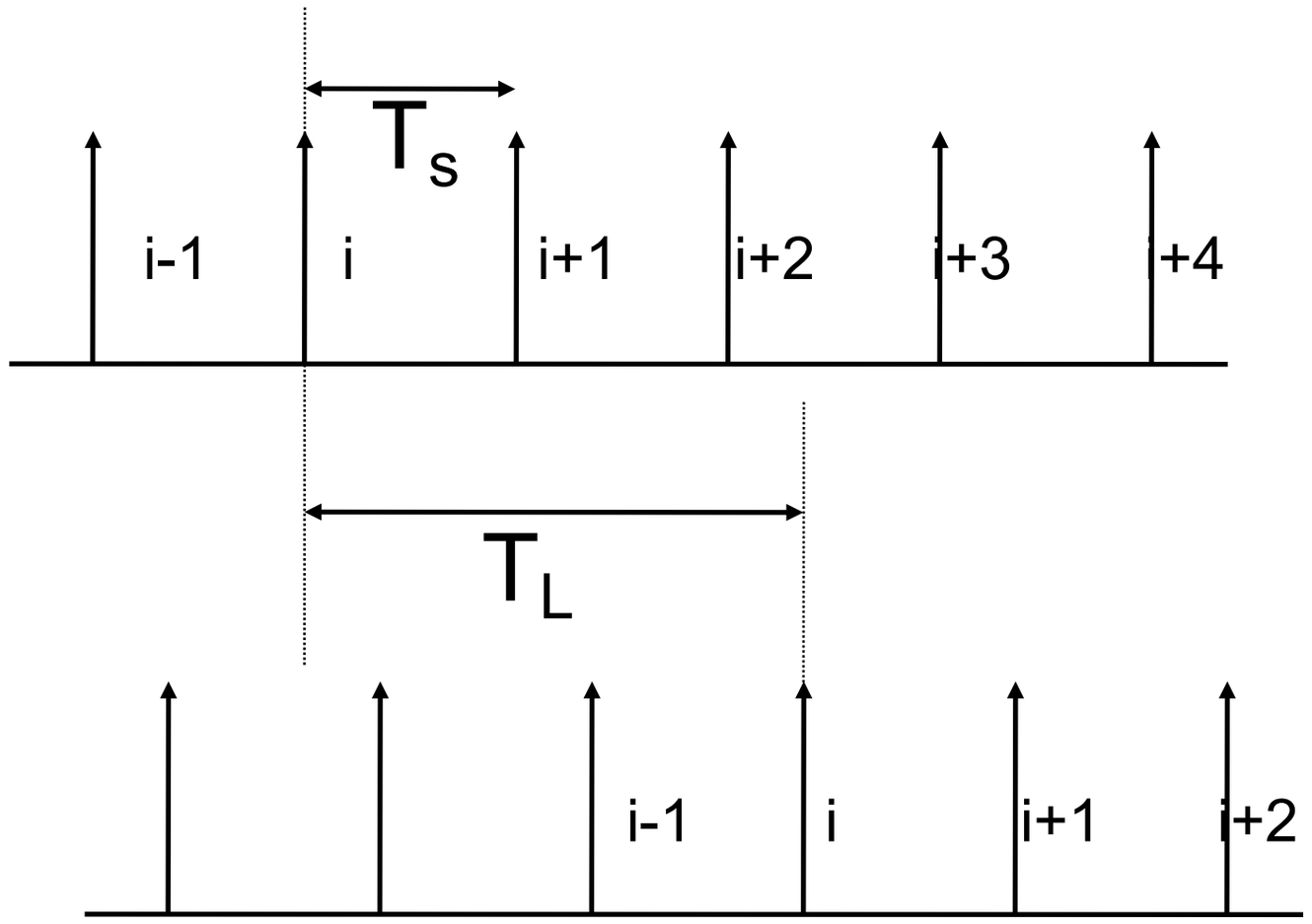


partially-ordered discrete events

Timing Constraints in DSP Systems

- Independent timing constraints: T_S and T_L





Guaranteeing Latency/Throughput

- Satisfying throughput (rate) constraints is (relatively) not very complicated
  - Constant event rate case is easy
  - Bursty rate is harder, but buffering helps and a large enough buffer is sufficient
    - as long as service rate is equal to the average throughput requirement
  - Can you think of a simple technique to improve throughput?

- Satisfying latency constraints is much harder
  - A system with service rate equal to the average throughput requirement is insufficient
  - Need proper scheduling, preemption, etc.
  - Will talk about this in more detail later…

More General Timing Constraints

- Two categories of timing constraints
  - Performance (Response/Output) constraints: set limits on response time of the system
  - Behavioral (Stimulus/Input) constraints: make demand on the rate at which the external environment supplies stimuli to the system

- Further classification: three types of temporal restrictions (not mutually exclusive)
  - Maximum: No more than "t" amount of time may elapse between the occurrence of one event and the occurrence of another
  - Minimum: No less than "t" amount of time may elapse between two events
  - Durational: An event must occur for "t" amount of time

- As defined before, event refers to either a stimulus/input to the system from its environment, or is an externally observable response/output that the system makes

Maximum Timing Constraints

- A. S-S combination: a maximum time is allowed between the occurrences of two stimuli
  - e.g., 2nd digit must be dialed no later than 20s after the 1st digit

- B. S-R combination: a maximum time is allowed between the arrival of a stimulus and the system's response
  - e.g., the caller shall receive a dial tone no later than 2s after lifting the phone receiver

- C. R-S combination: a maximum time is allowed between a system's response and the next stimulus from the environment
  - e.g., after receiving the dial tone, the caller shall dial the first digit within 30s

- D. R-R combination: a maximum time is allowed between two system's responses
  - e.g., after a connection is made, the caller will receive a ringback tone no more than 0.5s after the callee has received a ring tone

Observation

- A and C are behavioral requirements
  - Constraints on the environment/user
  - If the environment/user fails to generate the required stimulus within the prescribed time, the system will take a specific course of action
  - e.g., start a timer at the occurrence of the first stimulus (case A) or the system response (case C)
- B and D are system performance requirements

Minimum Timing Constraints

- S-S combination: a minimum time is required between the occurrences of two stimuli
  - e.g., a min of 0.5s must elapse between the dialing of one digit and dialing of the next
- S-R combination: a minimum time is required between the arrival of a stimulus and the system's response
  - e.g., after the caller has dialed 0, the system shall wait 15s before dialing (so that user may complete operator-assisted call himself)
- R-S combination: a minimum time is required between a system response and the next stimulus from the environment
  - e.g., where system can be busy processing requests from several ports, and can't accept new inputs at this port for a certain time
- R-R combination: a minimum time must pass between two responses by the system
  - e.g., user may need a certain time to act upon the first response

More Complex Timing Constraints

- Timing constraints on a sequence of responses
  - e.g., caller should dial 7 digits in 30s or less after lifting the receiver

- Durational: express duration of a stimulus or response
  - e.g., to get back to the operator, press the button for at least 15s (but not more than 30s); to get the dial tone press the button for more than 30s

- Two responses r1 and r2 should be heard within 60s after a stimulus s1, and r2 should be delayed at least by 15s after r1 and should occur within 30s after r1. Also, r1 & r2 should last for 3s and 4s respectively.

Real-time Tasks

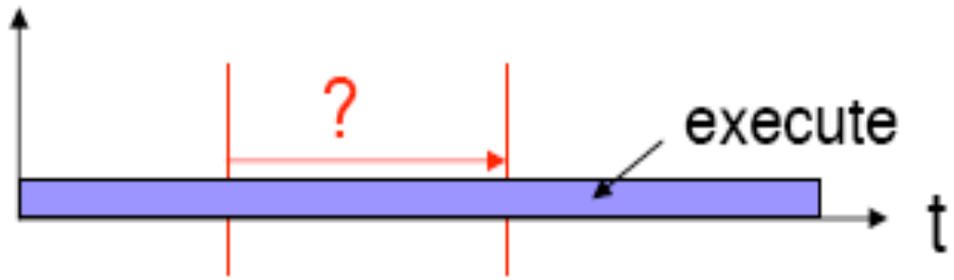
- Key distinction between a real-time task and a non-real-time task is the existence of a deadline
  - A correct result produced after deadline is wrong ("when" is as important as "what")
  - System must be able to consume next input before a deadline

- Hard Real-time vs. Soft Real-time
  - Depends on the consequence of a missed deadline
  - Soft: meeting deadline is desirable, but occasionally missing a deadline does not seriously jeopardize system behavior
  - Hard: missing deadline has catastrophic consequences for the system

- Typical systems have a mix of both types of tasks

Support for Time in Languages

- Measure elapsed time

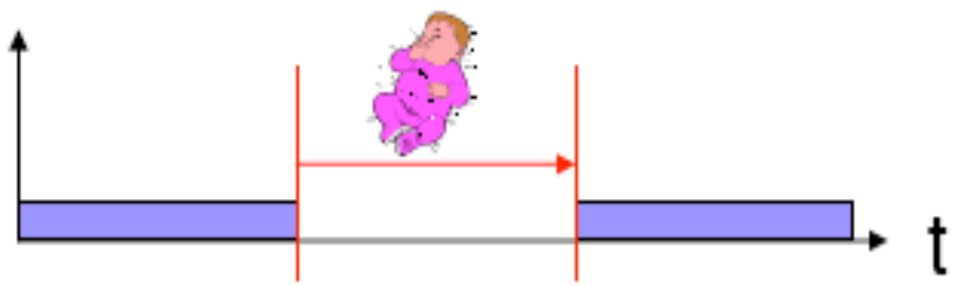


```
clock_t begin = clock();

/* here, do your time-consuming job */

clock_t end = clock();
double time_spent = (double)(end - begin) / CLOCKS_PER_SEC;
```

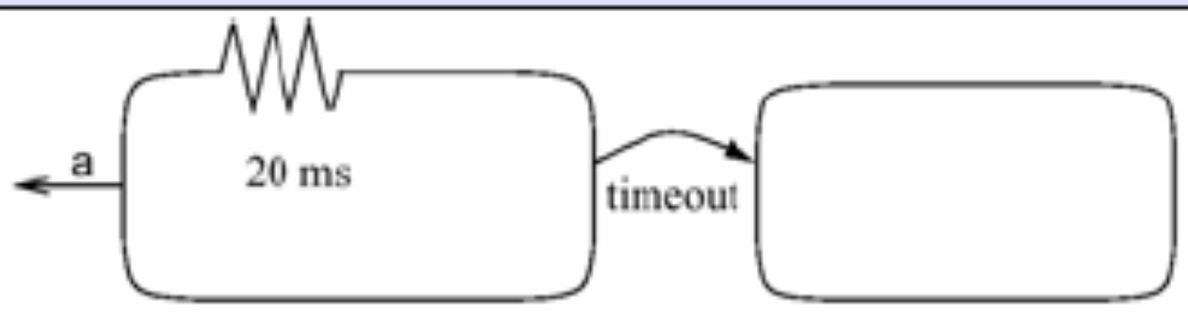
- Delaying processes



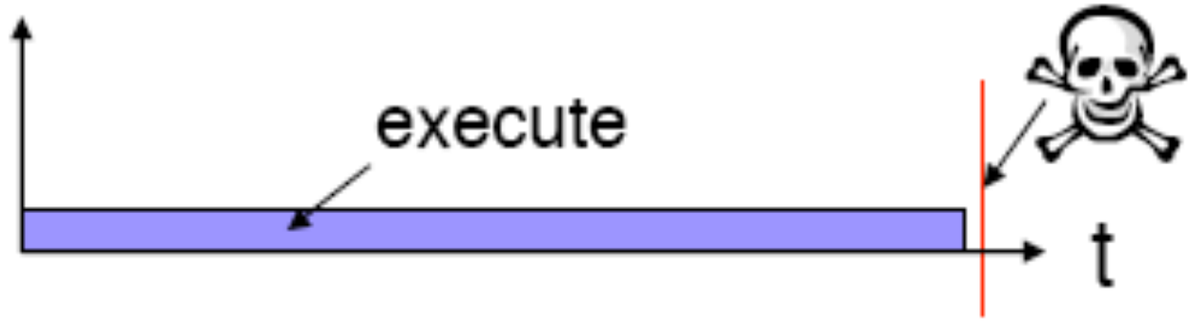
[Simunic, UCSD]

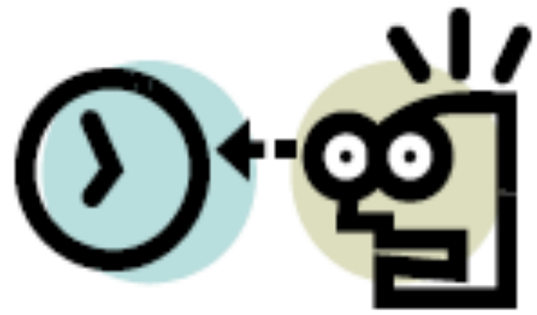
Support for Time in Languages

Timeouts



Deadlines





[Simunic, UCSD]

Summary

- Time is an essential differentiating factor between embedded systems and conventional computing systems
- Timing constraints can be specified in various ways
- Support for timing is lacking in programming languages