Cyber-Physical Systems (CSC.T431)

Introduction to CPS, Synchronous Model (1)

Instructor: Takuo Watanabe (Department of Computer Science)

Agenda

1. Course Introduction
2. Introduction to Cyber-Physical Systems
3. Synchronous Model (1)

Course Support & Material

- Slides: OCW-i
- Course Web: https://titech-cps.github.io
- Course Slack: titech-cps.slack.com

About This Class

(course syllabus)

- Overview
  - A cyber-physical system (CPS) is a collection of computational entities that communicate with one other and interact with the physical world via sensors and actuators. Such systems are omnipresent today, from automobiles to smart cities. In this course, basic concepts, theory, and issues of cyber-physical systems are examined.
- Objective
  - The course aims to ensure that students not only grasp the concepts, but also obtain basic skills to <u>formally model and verify</u> cyber-physical systems.

Outcomes

(course syllabus)

By the end of the course students should be able to:

1. explain the basic concepts of cyber-physical systems, reactive systems, synchronous/asynchronous computation models, dynamical systems, timed model, and hybrid systems.
2. have an understanding of safety/liveness properties of synchronous/asynchronous computation models.
3. have an understanding of the properties of dynamical systems, timed model, and hybrid systems.

Schedule

(Changes will be shown on the course web)

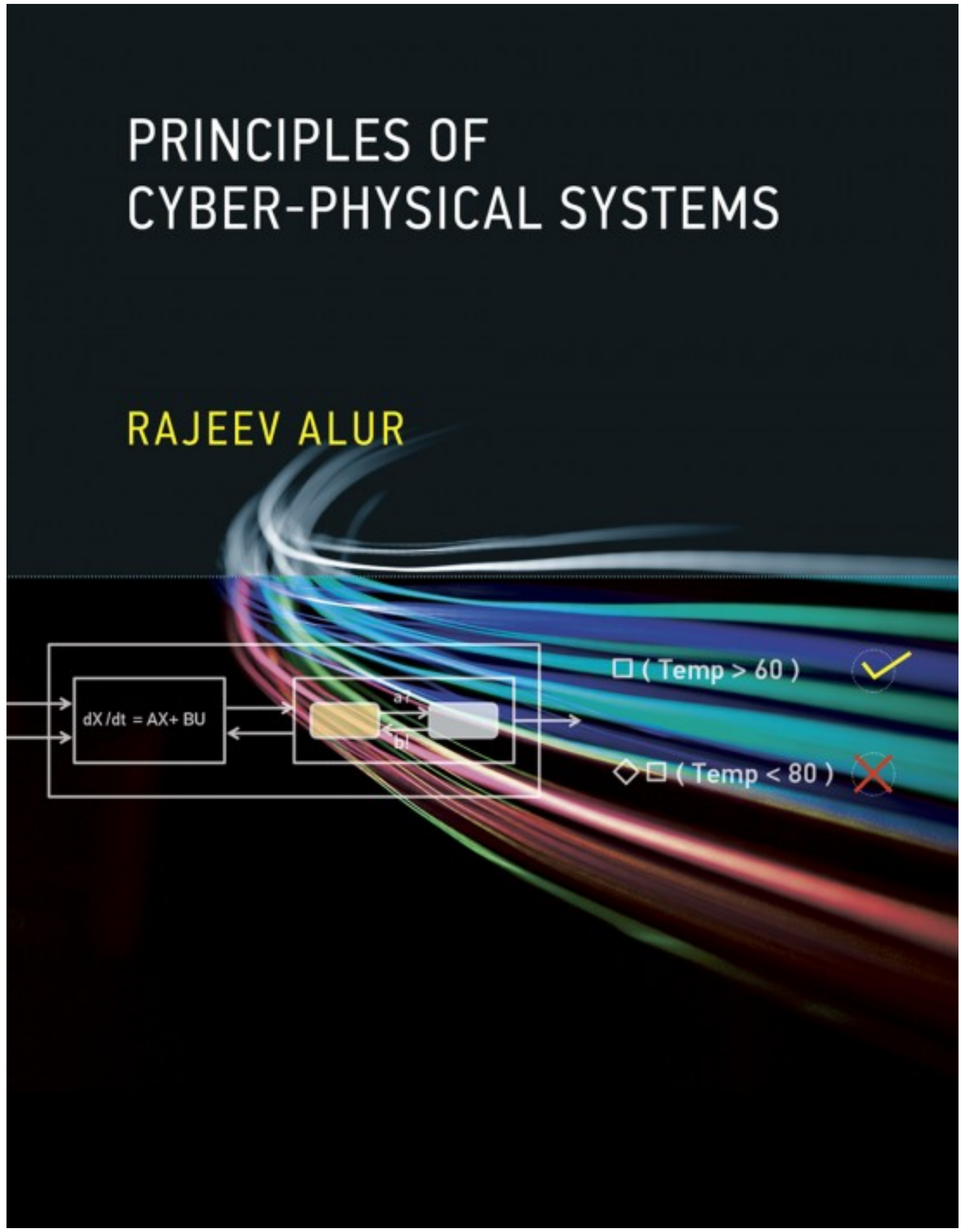
1. Course Introduction, Concepts of Cyber-Physical Systems
2. Synchronous Model (1): Concepts and Formalization
3. Synchronous Model (1): Properties
4. Safety Requirements and Verification
5. Asynchronous Model (1): Concepts and Formalization
6. Asynchronous Model (2): Properties
7. Liveness Requirements and Verification
8. Dynamical Systems (1): Concepts and Formalization
9. Dynamical Systems (2): Properties and Verification
10. Timed Models (1): Concepts and Formalization
11. Timed Model (2): Properties
12. Hybrid Systems (1): Concepts and Formalization
13. Hybrid Systems (2): Properties
14. Programming Models for CPS

Textbook

Principles of Cyber-Physical Systems
Rajeev Alur (Professor, U. Pennsylvania)
MIT Press, 2015, 464pp., 17x19in
ISBN: 9780262029117

https://mitpress.mit.edu/books/principles-cyber-physical-systems

https://www.cis.upenn.edu/~alur/pcps.html



Prerequisites

- Theoretical Computer Science
  - Automata Theory
  - Mathematical Logic
    - Propositional Logic, Predicate Logic, Proofs, Models
- System Programming / System Software
  - Concepts of Operating Systems / Concurrent Programming
    - Processes, Threads, Synchronization, Inter-Process Communication, etc.
- Programming Languages
  - Concepts of Languages Mechanisms
    - Variables, Types, Scope, Expressions, Control Structures, Procedures and Functions, etc.

Grading

Two assignments (50%+50%)

- #1 : middle October
- #2 : end of 3Q

Details will be announced later. Please wait for a while.

On-Line Services

- Lecture: Zoom
- Slides, Announcements: OCW-i
- Course Web: https://titech-cps.github.io
- Q&A, Discussions, Announcements: Slack (titech-cps.slack.com)

Introduction to Cyber-Physical Systems

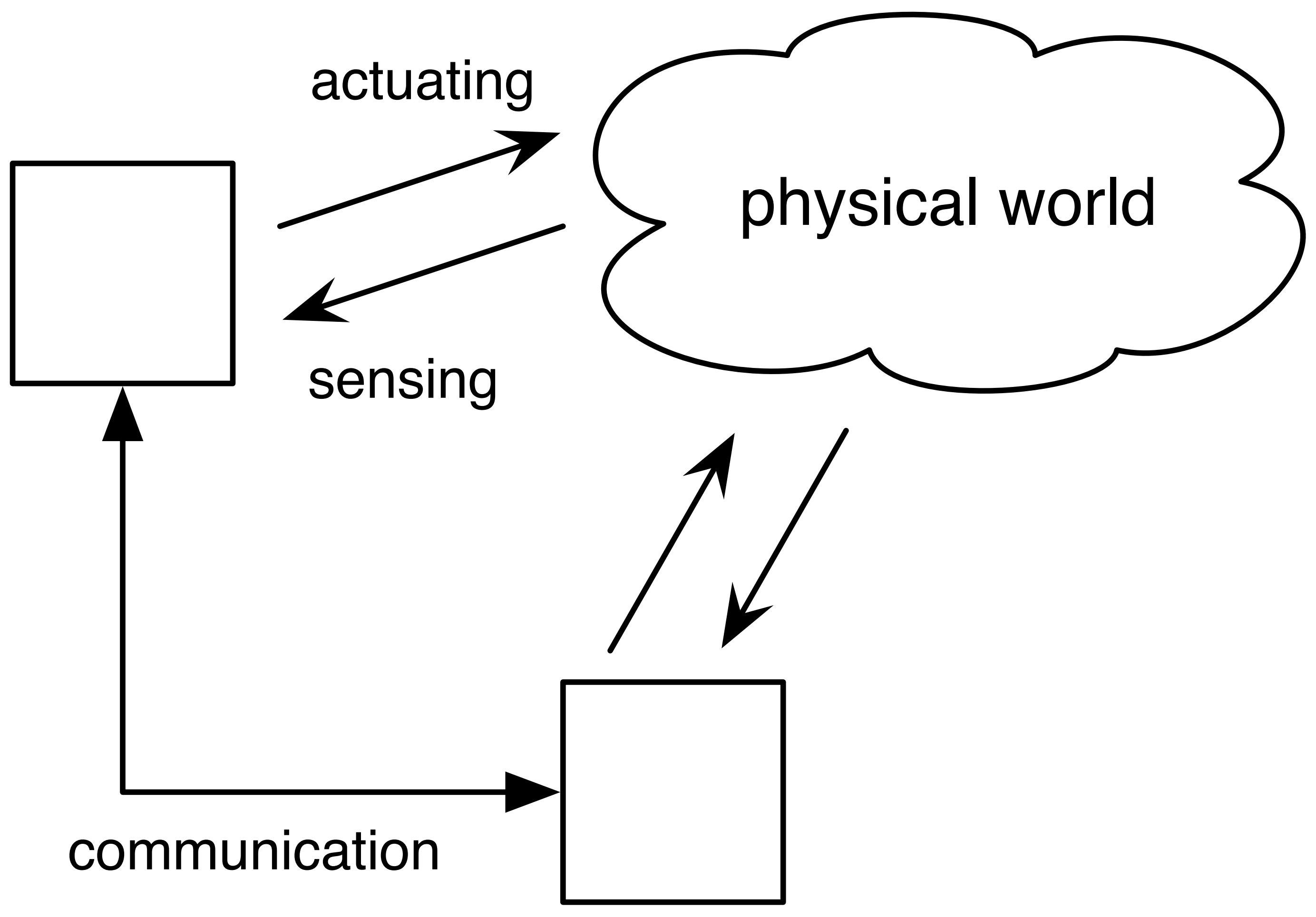
What is a Cyber-Physical System? (1)

- Traditional Computers
  - General-purpose standalone computing devices
    - Desktop/Laptop PCs, Mainframes, Supercomputers

- Embedded Systems
  - Special-purpose systems with integrated computing devices (microcontrollers, etc.)
    - Consumer digital devices (mobile phones, cameras, electronic instruments, etc.)
    - Home/Office appliances (TV sets, STBs, game consoles, microwave ovens, etc.)
    - Transportation (Cars, Trains, Airplanes, Spacecrafts, etc.), Weapons (Missiles, etc.)

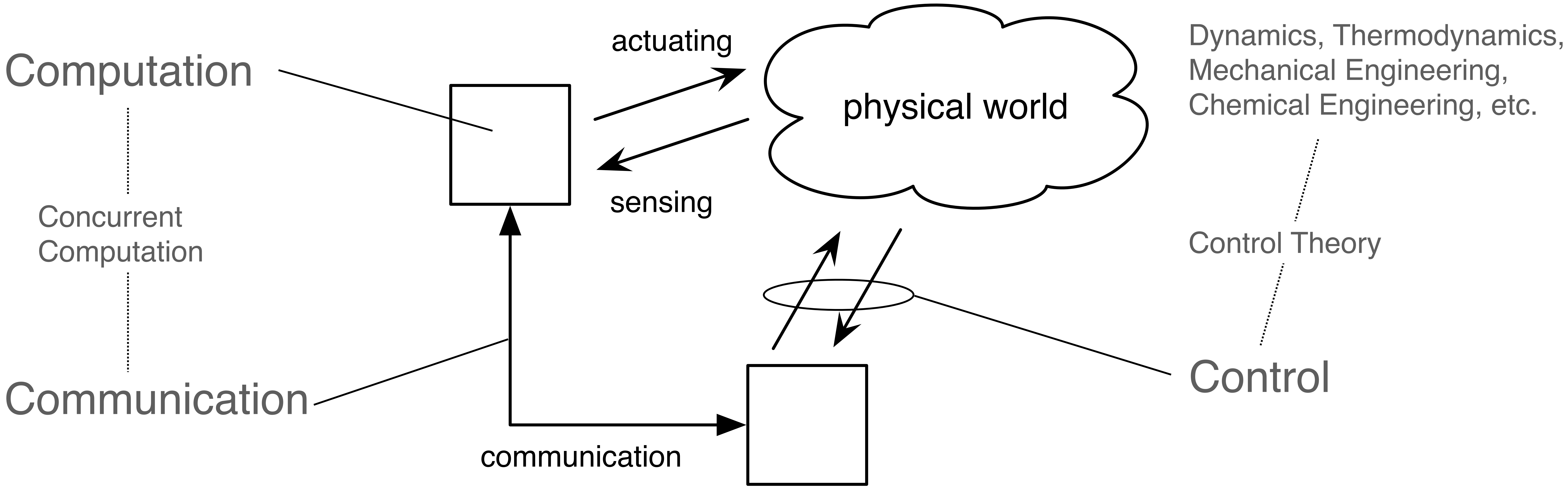
What is a Cyber-Physical System? (2)

- A cyber-physical system (CPS) is a collection of computing devices
  - communicating with each other, and
  - interacting with the physical world via sensors and actuators.
- Generalization of Embedded Systems
  - "Networked Embedded Systems"
- Example
  - Autonomous Mobile Robots
  - Wireless Sensor-Actor Networks (WSANs)
  - Public Infrastructure (Smart Building, Transportation, Telecommunication, Plants, etc.)



Challenges

Systematic Integration of Computation, Communication & Control



Reactive Systems

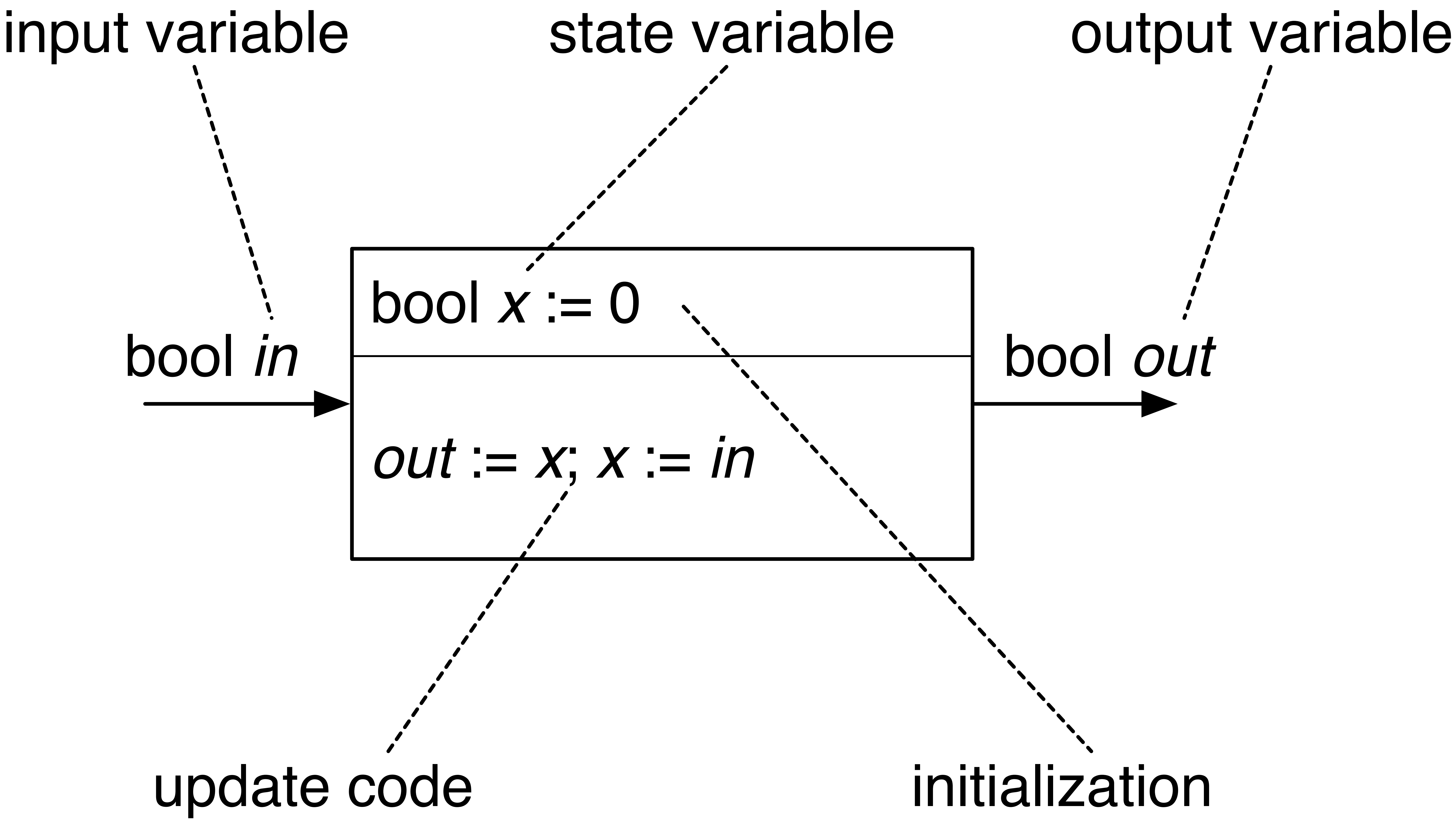
- Transformational System
  - A computing device produces an output (and stops) when supplied with an input.
  - Classical model of computation: Turing Machine, Recursive Function
  - Ex. Compiler
- Reactive System
  - A computational system that interacts with its environment.
    - It is not supposed to stop but should be continuously ready for interactions.
    - Timeliness is often important for desired reaction.
  - Ex. Embedded Systems, GUIs

Overview of Topics

- Formal Models
  - Synchronous Model (all components execute in lock-step in a sequence of rounds)
  - Asynchronous Model (different activities execute at independent speeds)
  - Dynamical Systems (continuous-time models for capturing the evolution of phys. world)
  - Timed Model (concrete bounds on timing delays)
  - Hybrid Systems (integration of discrete interaction and continuity)
- Specification & Analysis
  - Safety (nothing bad ever happens)
  - Liveness (something good eventually happens)

Synchronous Model (1)

Ex. Delay

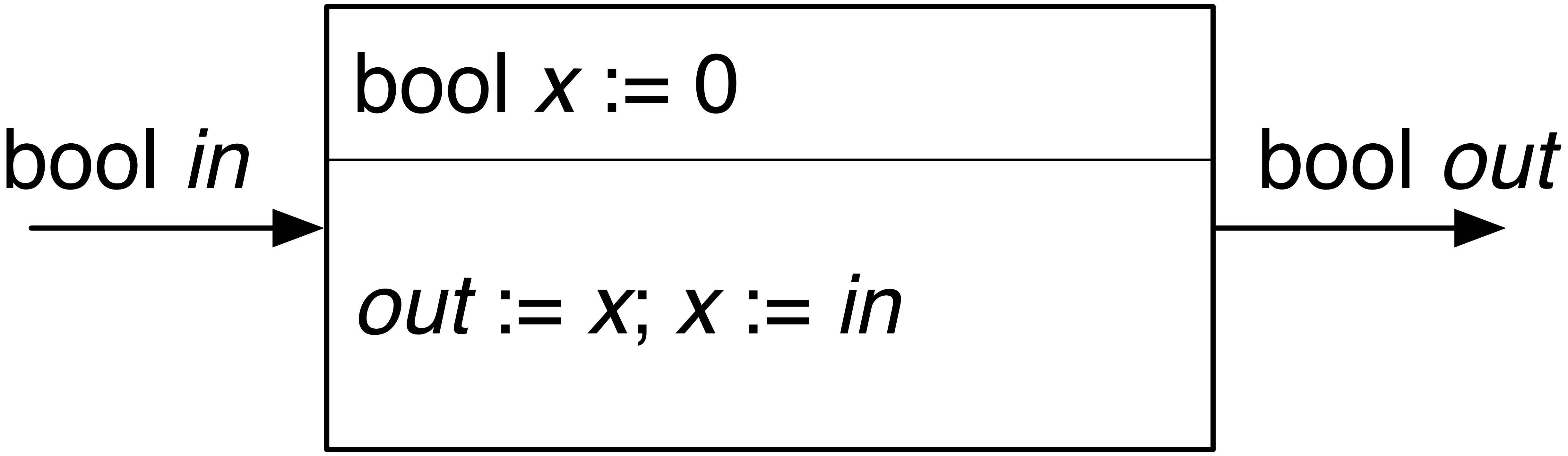


input variable
state variable
output variable
bool x := 0
bool in
bool out
out := x; x := in
update code
initialization

Ex. Delay

$$\mathsf{x} \xrightarrow{\mathsf{in}/\mathsf{out}} \mathsf{x}'$$

$$0 \xrightarrow{0/0} 0;\ 0 \xrightarrow{1/0} 1;\ 1 \xrightarrow{0/1} 0;\ 1 \xrightarrow{1/1} 0$$



round	**init**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
in		0	1	0	1	1	0	0	1
x		0	0	1	0	1	1	0	0
x'	0	0	1	0	1	1	0	0	1
out		0	0	1	0	1	1	0	0

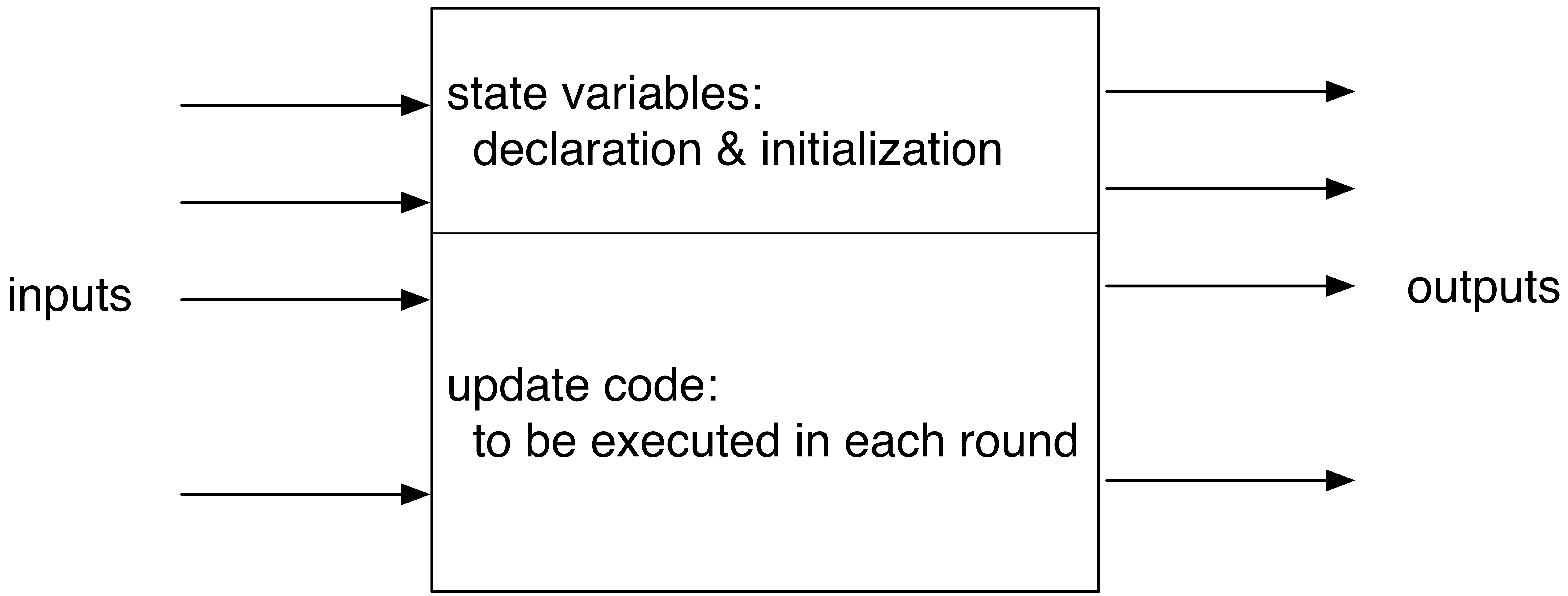
Synchrony Hypothesis

- We assume that the time needed to execute the update code is negligible compared to delay between successive input arrivals (or time between rounds).
- Logically, we can say that the execution of update code takes zero time, and the production of output and reception of inputs occurs at the same time.

- Cf. Synchronous Languages
  - Programming languages based on the synchrony hypothesis
  - e.g., Esterel, Lustre, Signal, Ceu

Synchronous Reactive Components

$$C = (I, O, S, [[Init]], [[React]])$$



Synchronous Reactive Components

$$C = (I, O, S, [[Init]], [[React]])$$

- A synchronous reactive component C is described by
  - a finite set I of typed input variables defining the set Q_I of *inputs*,
  - a finite set O of typed output variables defining the set Q_O of *outputs*,
  - a finite set S of typed state variables defining the set Q_S of *states*,
  - an initialization $Init$ defining the set $[[Init]] \subseteq Q_S$ of *initial states*, and
  - a reaction description $React$ defining the set $[[React]]$ of reactions of the form $s \xrightarrow{i/o} t$ where s, t are states, i is an input, and o is an output.

- Ex. (Delay) $I = \{\mathit{in}\}, O = \{\mathit{out}\}, S = \{x\}$

Valuation

- V : set of typed variables
- $q : V \to D$: a valuation over V
  - For all $v \in V$, $q(v)$ is a value of type τ_v (τ_v : type of v)
- Q_V : set of all valuations over V

$V = \{x, y\}$
$q : V \to N$
$q(x) = 1, q(y) = 2$
$\Downarrow$
$q = \{(x, 1), (y, 2)\}$

- Ex. (Delay) $I = \{\mathit{in}\}$, $O = \{\mathit{out}\}$, $S = \{x\}$
  - $Q_I = \{\{(\mathit{in}, 0)\}, \{(\mathit{in}, 1)\}\}$, $Q_O = \{\{(\mathit{out}, 0)\}, \{(\mathit{out}, 1)\}\}$, $Q_S = \{\{(x, 0)\}, \{(x, 1)\}\}$
  - $[[\mathit{Init}]] = \{\{(x, 0)\}\}$

Reaction

$$s_0 \xrightarrow{i_1/o_1} s_1 \xrightarrow{i_2/o_2} s_2 \xrightarrow{i_3/o_3} s_3 \cdots s_{k-1} \xrightarrow{i_k/o_k} s_k$$

- $s_j \in Q_S$: state ($0 \leq j \leq k$), $i_j \in Q_S$: input ($1 \leq j \leq k$), $o_j \in Q_S$: output ($1 \leq j \leq k$)
- $s_0 \in [[Init]]$: initial state
- $s_{j-1} \xrightarrow{i_j/o_j} s_j \in [[React]]$: reaction ($1 \leq j \leq k$)

Ex. Delay

$$
\begin{aligned}
I &= \{\mathsf{in}\} \\
O &= \{\mathsf{out}\} \\
S &= \{\mathsf{x}\} \\
Q_I &= \{\{(\mathsf{in}, 0)\}, \{(\mathsf{in}, 1)\}\} \\
Q_O &= \{\{(\mathsf{out}, 0)\}, \{(\mathsf{out}, 1)\}\} \\
Q_S &= \{\{(\mathsf{x}, 0)\}, \{(\mathsf{x}, 1)\}\} \\
[[Init]] &= \{\{(\mathsf{x}, 0)\}\} \\
[[React]] &= \left\{ \begin{array}{l}
\{(\mathsf{x}, 0)\} \xrightarrow{\{(\mathsf{in},0)\}/\{(\mathsf{out},0)\}} \{(\mathsf{x}, 0)\}, \\
\{(\mathsf{x}, 0)\} \xrightarrow{\{(\mathsf{in},1)\}/\{(\mathsf{out},0)\}} \{(\mathsf{x}, 1)\}, \\
\{(\mathsf{x}, 1)\} \xrightarrow{\{(\mathsf{in},0)\}/\{(\mathsf{out},1)\}} \{(\mathsf{x}, 0)\}, \\
\{(\mathsf{x}, 1)\} \xrightarrow{\{(\mathsf{in},1)\}/\{(\mathsf{out},1)\}} \{(\mathsf{x}, 1)\}
\end{array} \right\}
\end{aligned}
$$

$$
\Rightarrow \boxed{\left\{ \begin{array}{l}
0 \xrightarrow{0/0} 0, \\
0 \xrightarrow{1/0} 1, \\
1 \xrightarrow{0/1} 0, \\
1 \xrightarrow{1/1} 1
\end{array} \right\}}
$$

Summary

- Course Introduction
- Introduction to Cyber-Physical Systems
  - Definition of CPS, Reactive System, Overview of Topics
- Synchronous Model (1)
  - Delay Module
  - Formal Definition of Synchronous Reactive Modules