Teaching Embedded Systems at Berkeley

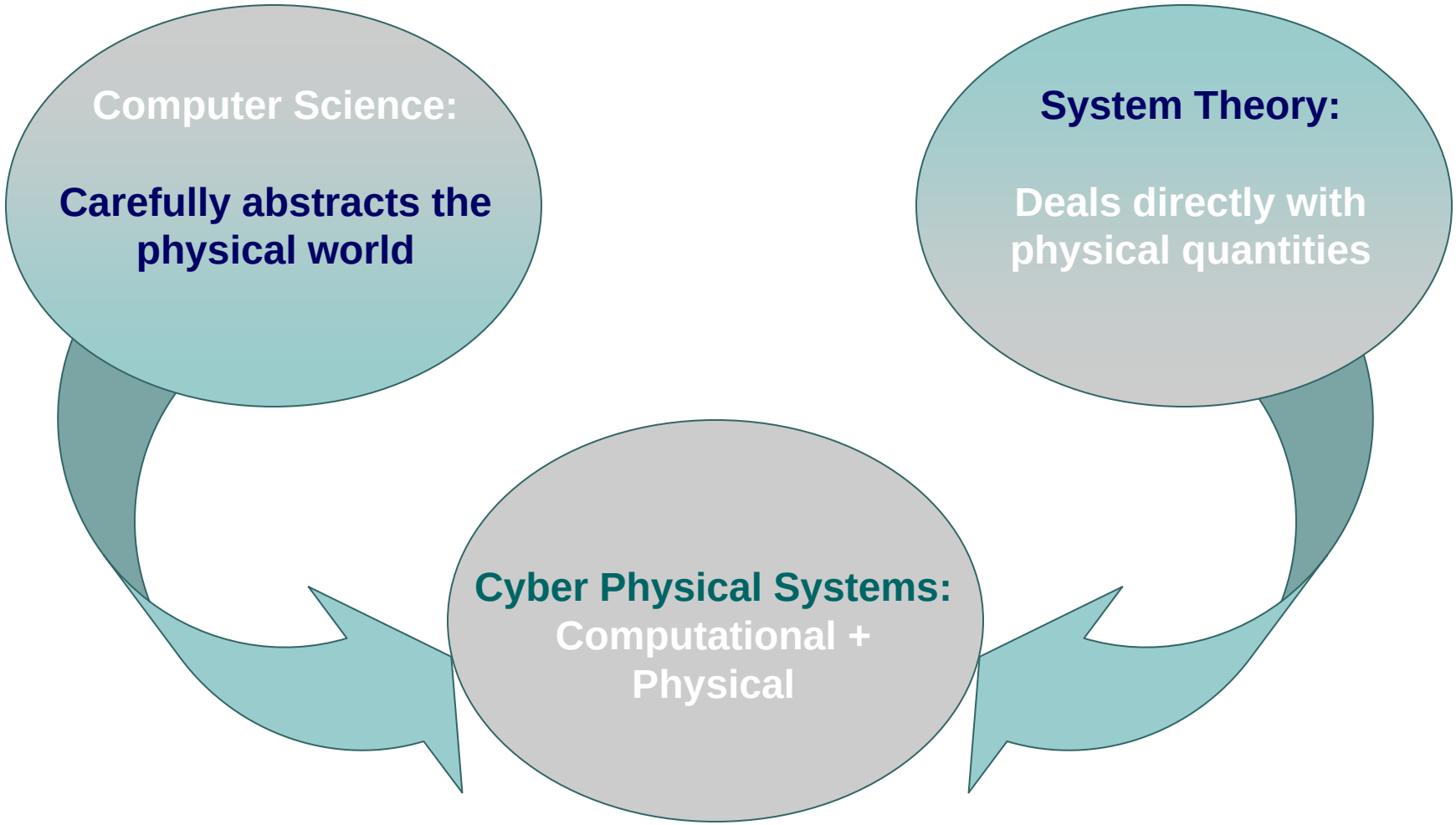
Edward A. Lee

Robert S. Pepper Distinguished Professor
UC Berkeley

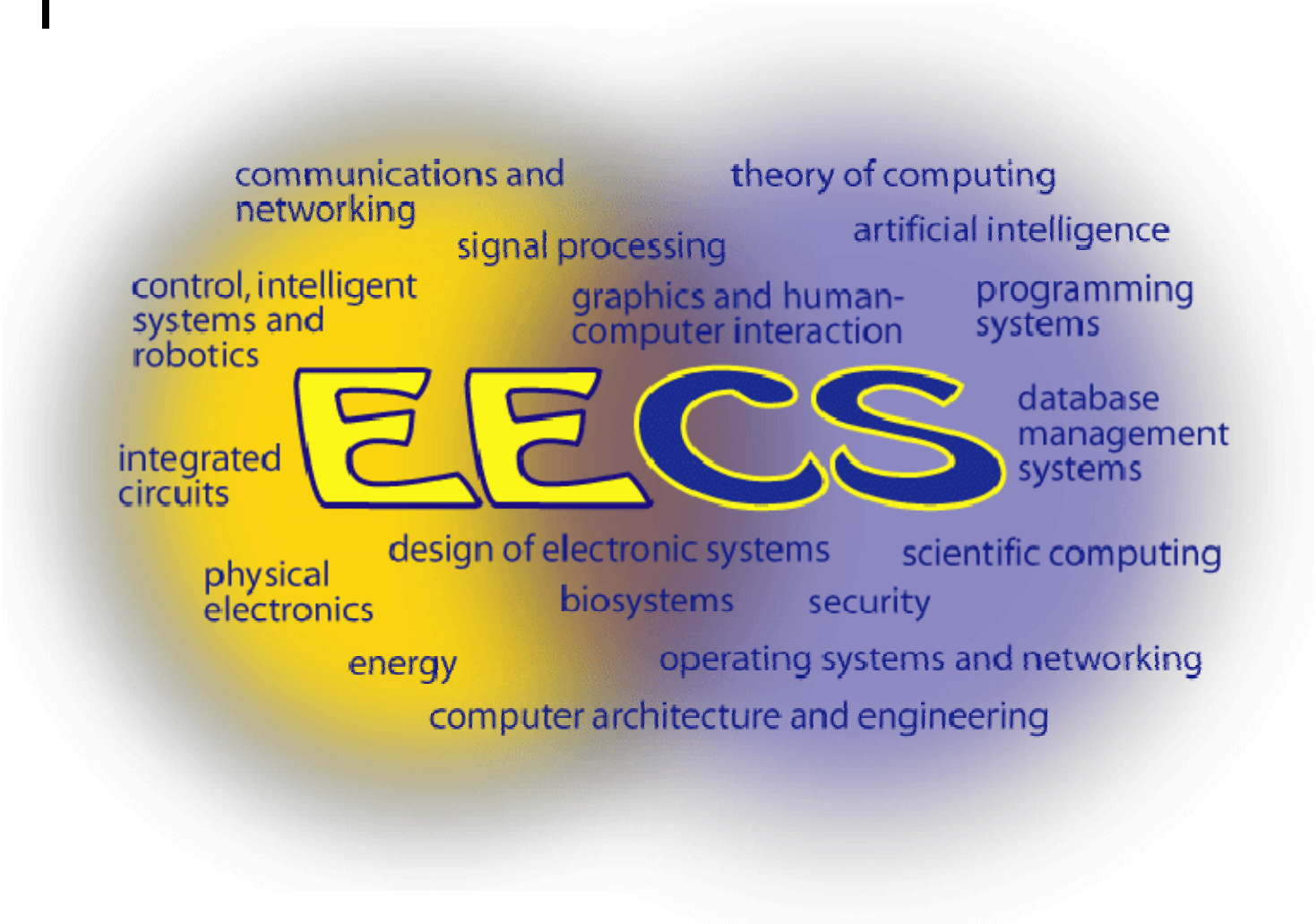
Panel at CPS Forum
CPS Week

San Francisco, CA
April 15, 2009

CPS is Multidisciplinary



An Advantage:
A Unified EECS Department



New Undergraduate Course (Spring 2008) "Introduction to Embedded Systems"

This course is intended to introduce students to the design and analysis of computational systems that interact with physical processes. Applications of such systems include medical devices and systems, consumer electronics, toys and games, assisted living, traffic control and safety, automotive systems, process control, energy management and conservation, environmental control, aircraft control systems, communications systems, instrumentation, critical infrastructure control (electric power, water resources, and communications systems for example), robotics and distributed robotics (telepresence, telemedicine), defense systems, manufacturing, and smart structures.

A major *theme of this course will be on the interplay of practical design with formal models of systems*, including both software components and physical dynamics. A major emphasis will be on building high confidence systems with real-time and concurrent behaviors.

- *Cyber-Physical Systems*
- *Model-Based Design*
- *Sensors and Actuators*
- *Interfacing to Sensors and Actuators*
- *Actors, Dataflow*
- *Modeling Modal Behavior*
- *Concurrency: Threads and Interrupts*
- *Hybrid Systems*
- *Simulation*
- *Specification; Temporal Logic*
- *Reachability Analysis*
- *Controller Synthesis*
- *Control Design for FSMs and ODEs*
- *Real-Time Operating Systems (RTOS)*
- *Scheduling: Rate-Monotonic and EDF*
- *Concurrency Models*
- *Execution Time Analysis*
- *Localization and Mapping*
- *Real-Time Networking*
- *Distributed Embedded Systems*

A Theme in our CPS Courses: Model-based design

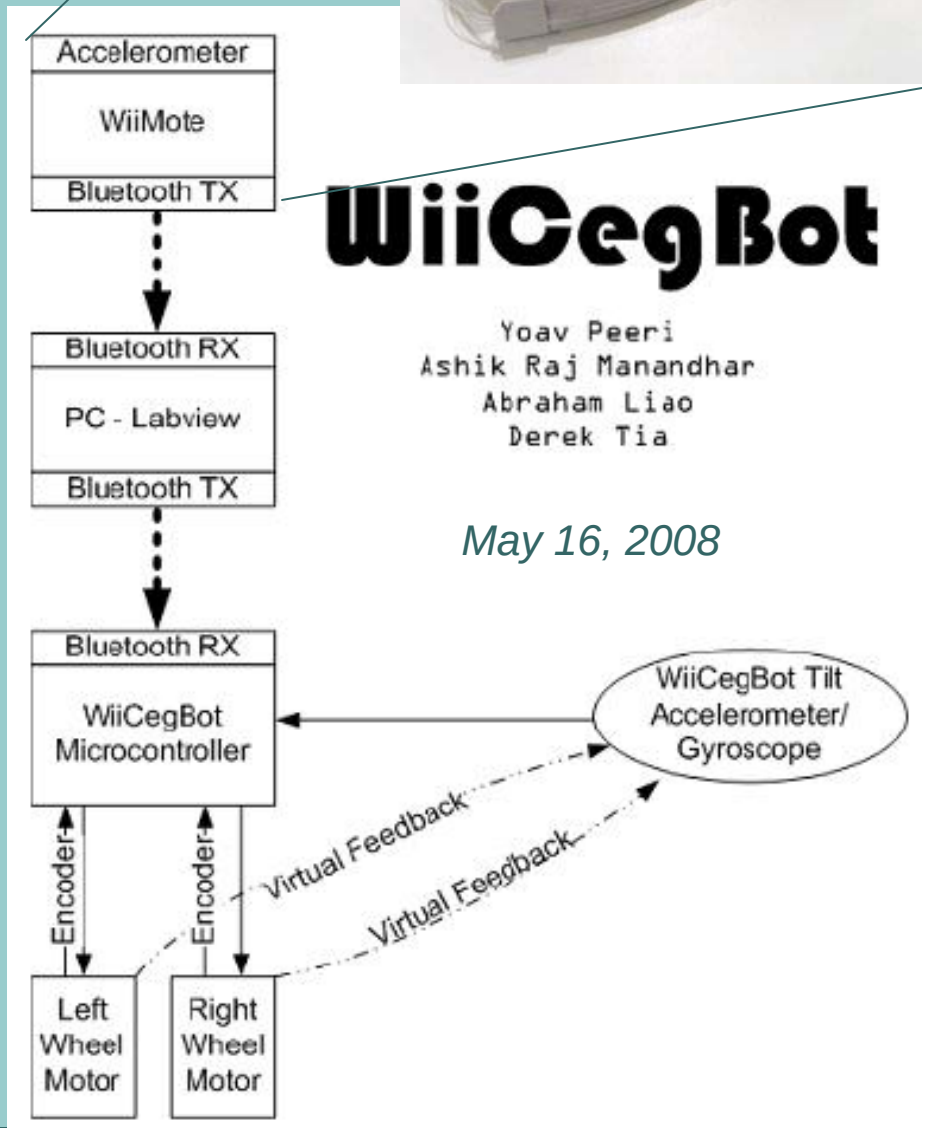
Models are abstractions of systems:

- structural (OO design)
- ontological (type systems)
- imperative logic (“procedural epistemology”)
- functional logic
- actor-oriented (including dataflow models)

All of these have their place…

Example Class Project

One of the five project teams developed a balancing robot inspired by the Segway. They used a Nintendo Wiimote as a controller communicating with a PC running LabVIEW, communicating with a Lego Mindstorm NXT, which they programmed in C.



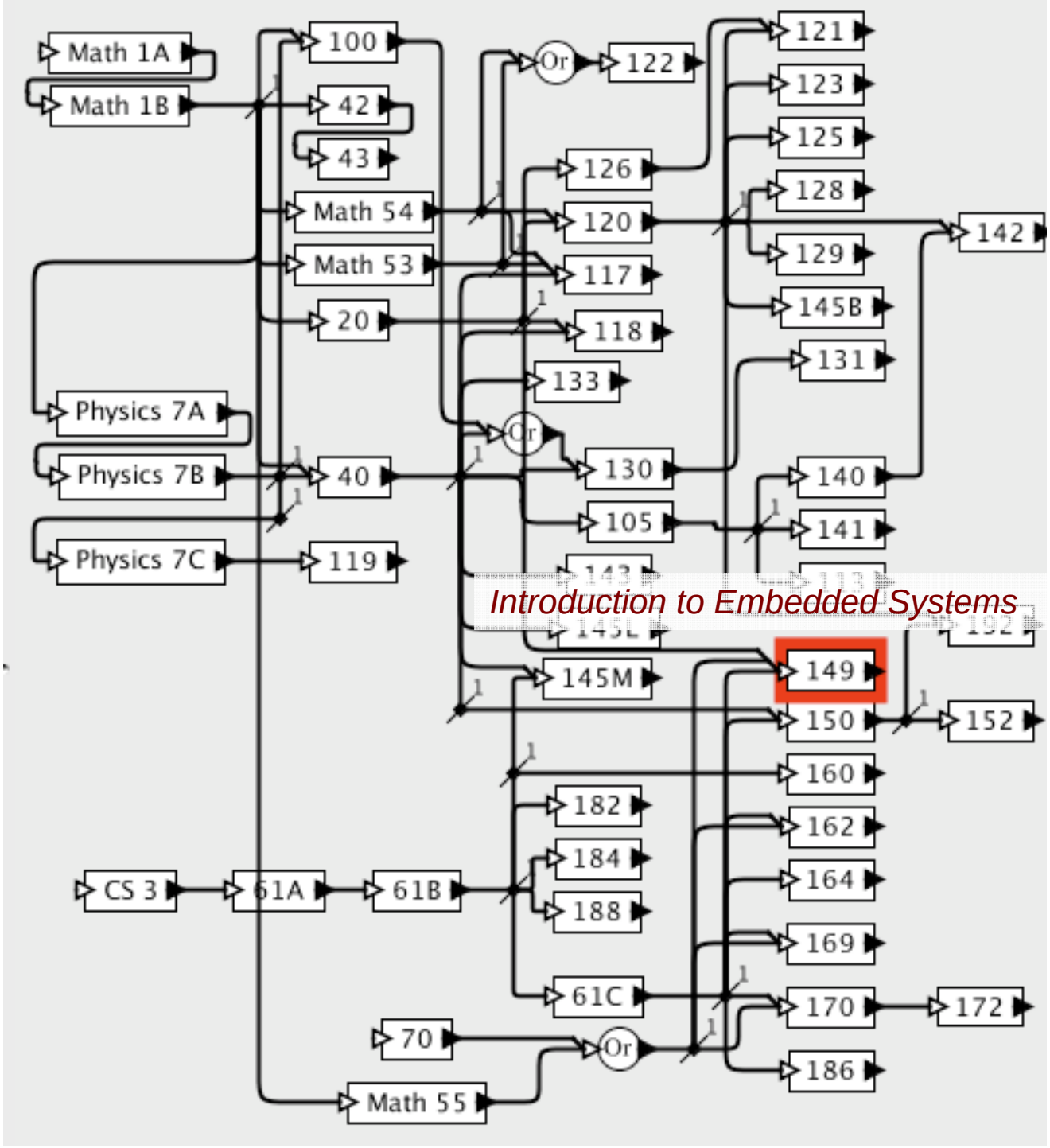
WiiCegBot

Yoav Peeri
Ashik Raj Manandhar
Abraham Liao
Derek Tia

May 16, 2008

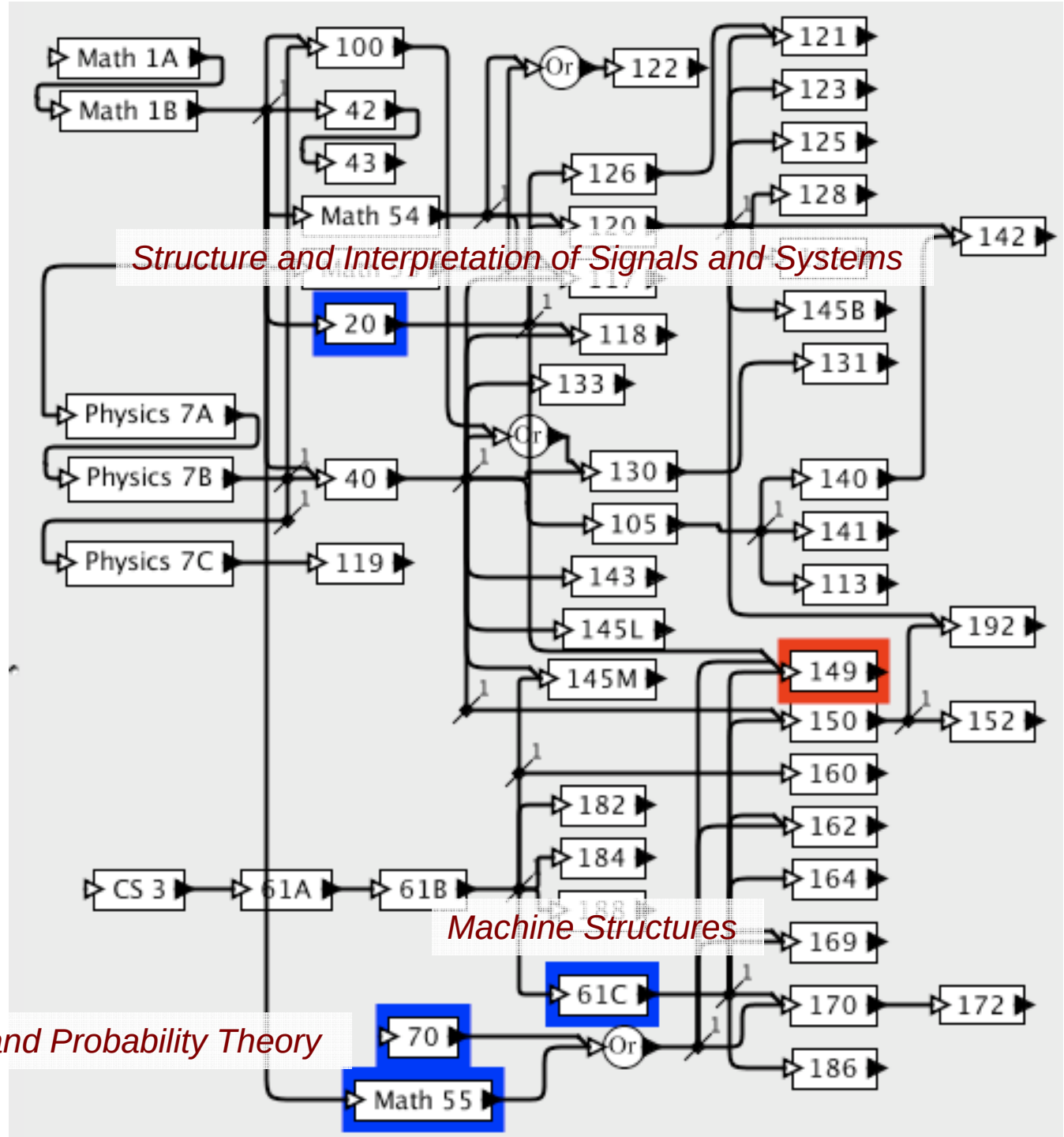
The Prerequisite Challenge

- Core course



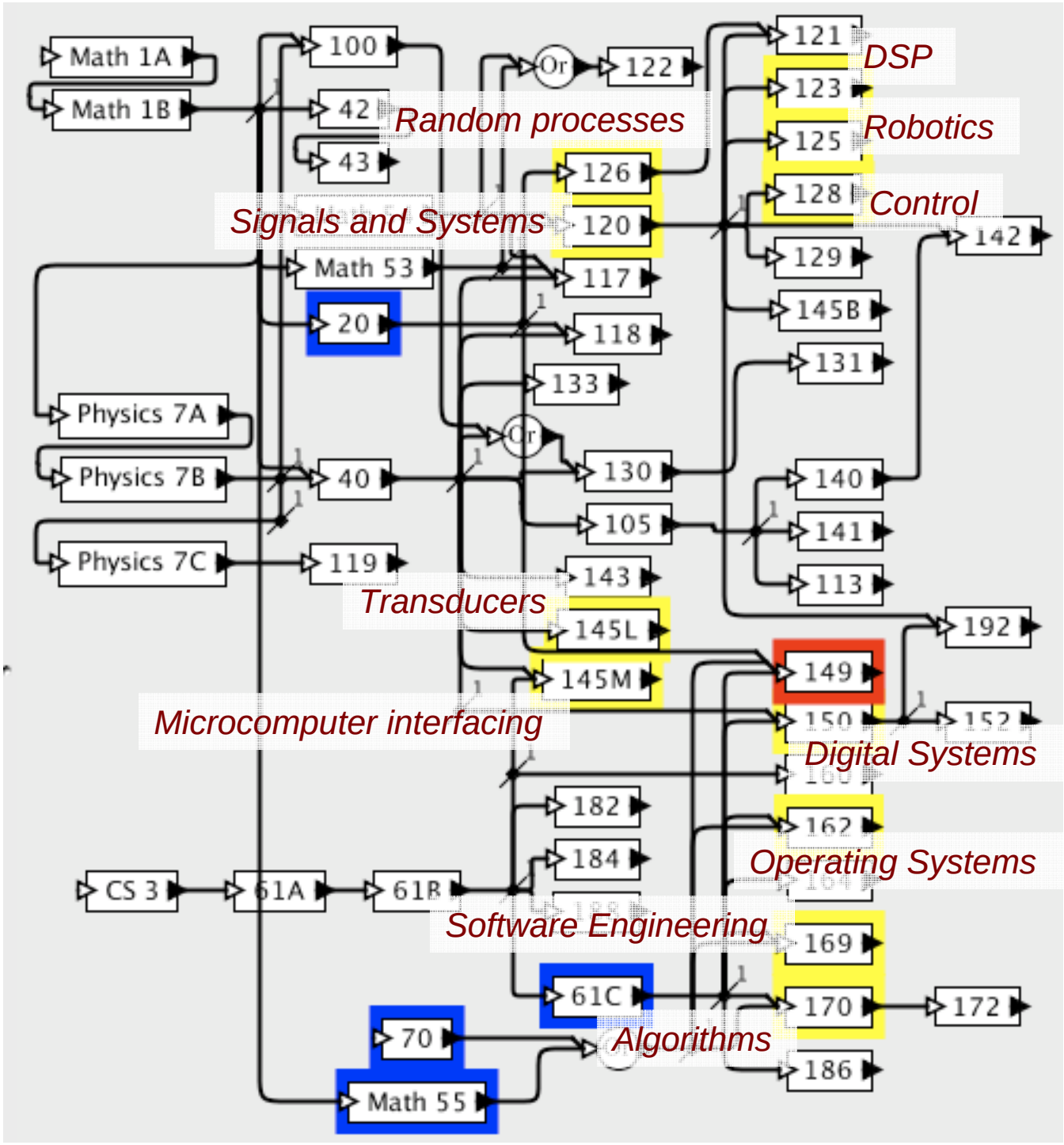
The Prerequisite Challenge

- Core course
- Prerequisites



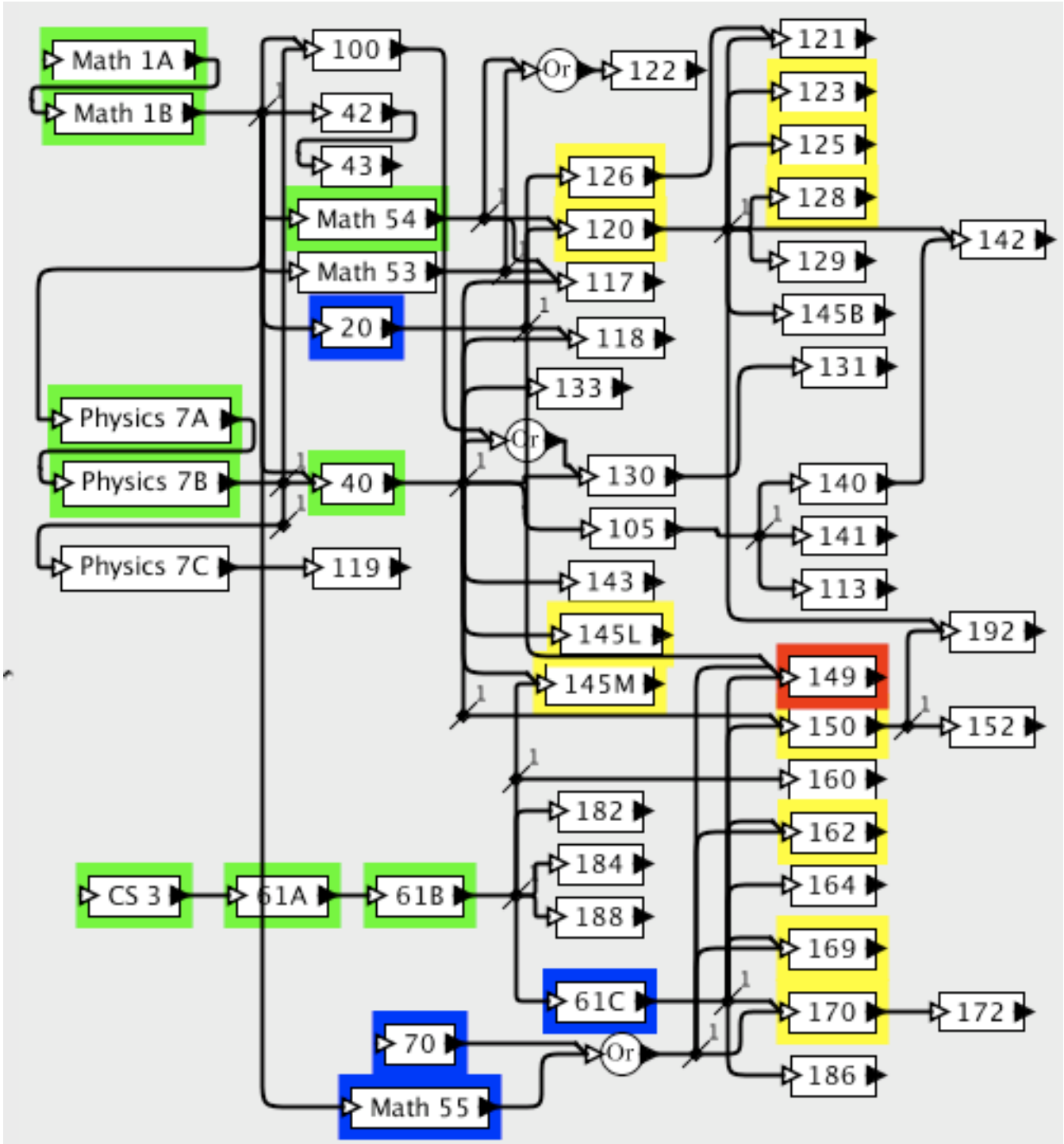
The Prerequisite Challenge

- Core course
- Prerequisites
- Defensible prerequisites



The Prerequisite Challenge

- Core course
- Prerequisites
- Defensible prerequisites
- Transitive closure



Conclusion

A *reductionist approach* (where we teach the foundations first) cannot work!

We have to be prepared to review topics as needed, and deal with a mix of backgrounds in the students.