EEL 4783: HDL in Digital System Design

Lecture: SystemC Language and Its Usage

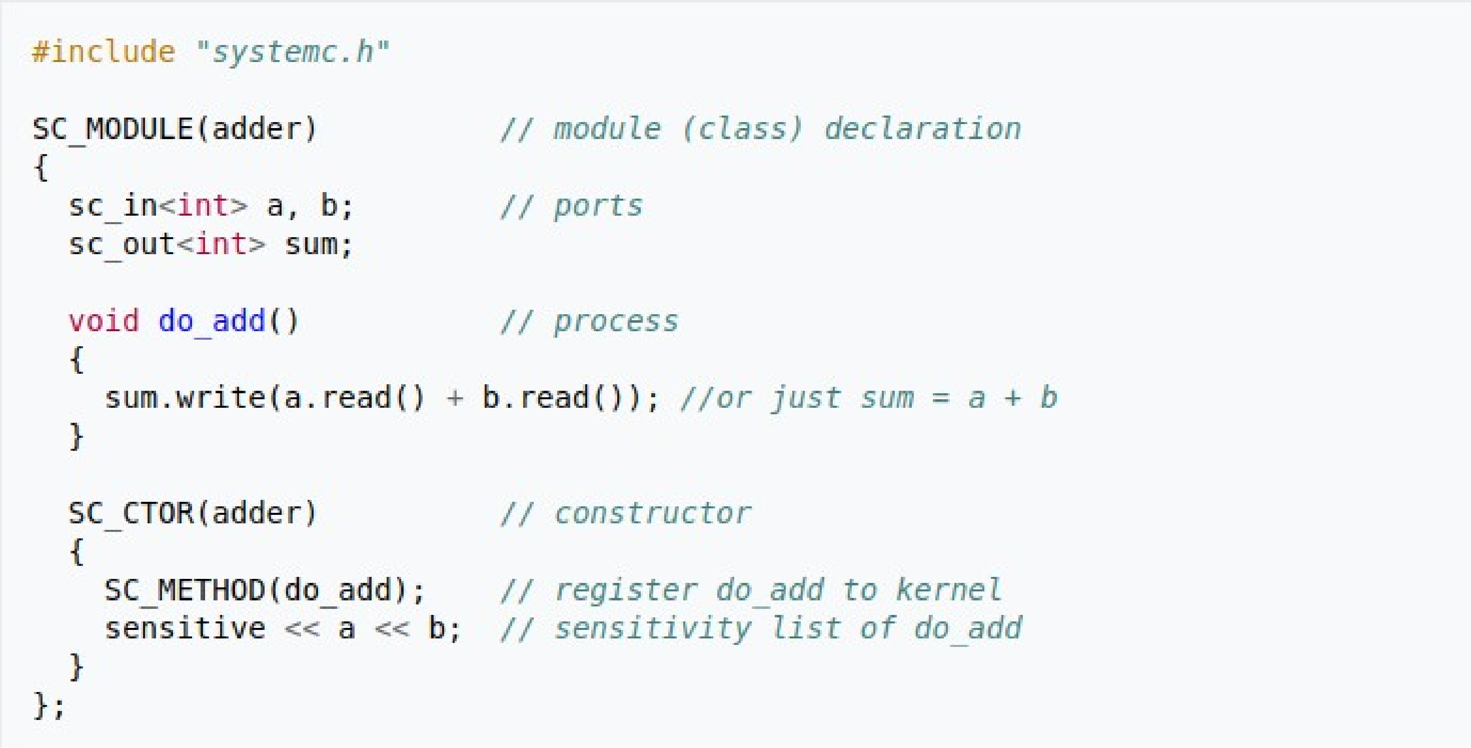
Part 1

Prof. Mingjie Lin

UCF

Stands For Opportunity

SystemC Code Example



```
#include "systemc.h"

SC_MODULE(adder)          // module (class) declaration
{
  sc_in<int> a, b;        // ports
  sc_out<int> sum;

  void do_add()           // process
  {
    sum.write(a.read() + b.read()); //or just sum = a + b
  }

  SC_CTOR(adder)          // constructor
  {
    SC_METHOD(do_add);    // register do_add to kernel
    sensitive << a << b;  // sensitivity list of do_add
  }
};
```

SystemC Introduction

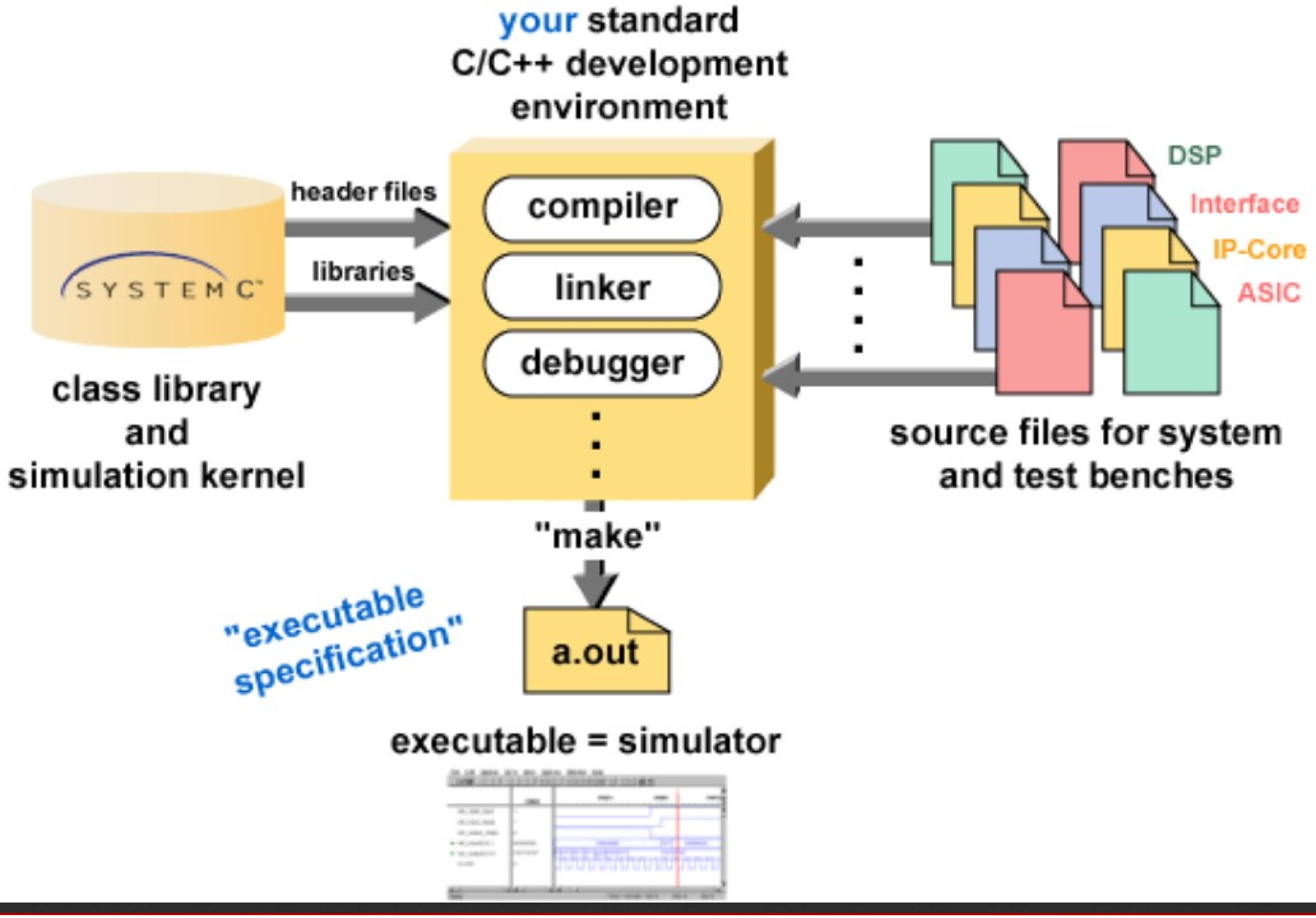
- Why not leverage experience of C/C++ developers for H/W & System Level Design?
- But C/C++ have no
  - – notion of time
    - No event sequencing
  - – Concurrency
    - But H/W is inherently concurrent
  - – H/W Data Types
    - No ‘Z’value for tri-state buses

SystemC is ...

- C++ Class Library use for
  - Cycle-Accurate model for Software Algorithm
  - Hardware Architecture
  - Interface of SoC (System-on-Chip)
  - System-level designs
  - Executable Specification

- www.systemc.org
  - All you can use information

SystemC Environment



SystemC History

- SystemC 1.0
  - Provide VHDL like capabilities
  - Simulation kernel
  - Fixed point arithmetic data types
  - Signals (communication channels)
  - Modules
    - Break down designs into smaller parts

SystemC History

- SystemC 2.0
  - Complete library rewrite to upgrade into true SLDL
  - Events as primitive behavior triggers
  - Channels, Interfaces and Ports
  - Much more powerful modeling for Transaction Level
- Future SystemC3.0
  - Modeling of Oss
  - Support of embedded S/W models

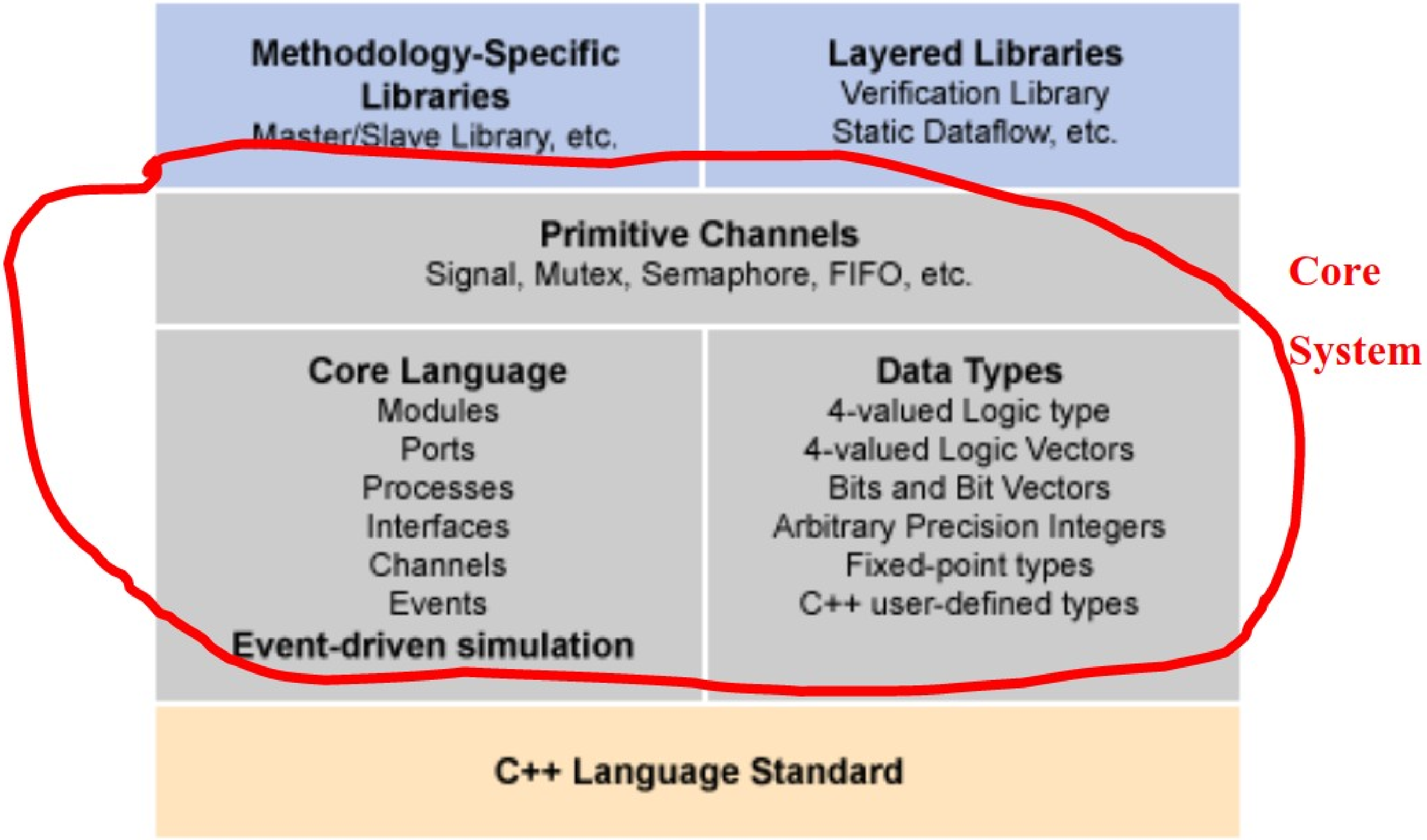
Objectives of SystemC 2.0

- Primary goal: Enable System-Level Modeling
  - Systems include hardware and software
  - Challenge:
    - Wide range of design models of computation
    - Wide range of design abstraction levels
    - Wide range of design methodologies

Objectives of SystemC 2.0

- Introduces a small but very general purpose modeling foundation => Core Language
- Support for other models of computation, methodologies, etc
  - They are built on top of the core language, hence are separate from it
    - Even SystemC1.0 Signalsare built on top of this core in SystemC2.0
    - Other library models are provided:
      - FIFO, Timers, ...

SystemC Language Architecture



Methodology-Specific Libraries
Master/Slave Library, etc.

Layered Libraries
Verification Library
Static Dataflow, etc.

Primitive Channels
Signal, Mutex, Semaphore, FIFO, etc.

Core Language
Modules
Ports
Processes
Interfaces
Channels
Events
Event-driven simulation

Data Types
4-valued Logic type
4-valued Logic Vectors
Bits and Bit Vectors
Arbitrary Precision Integers
Fixed-point types
C++ user-defined types

C++ Language Standard

Core System

Final issues

- Come by my office hours (right after class)
- Any questions or concerns?