EECS 222: Embedded System Modeling Lecture 6

Rainer Dömer

doemer@uci.edu

The Henry Samueli School of Engineering
Electrical Engineering and Computer Science
University of California, Irvine

Lecture 6: Overview

- SystemC System Description Language
  - SystemC Overview
  - Resources
- Introduction to the SystemC Language (Part 1)
  - Presentation by Stuart Swan, Cadence
- Homework Assignment 3
  - Producer-consumer example in SystemC

SystemC Overview

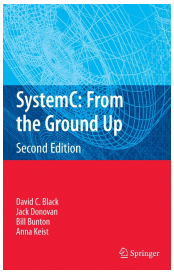
- Goals
  - Common C++ Modeling Platform
    - System level modeling
    - Register Transfer Level (RTL) modeling
  - Seamless Co-Design of Hardware and Software
  - Intellectual Property (IP) Reuse
  - Free licensing, Open Source
  - Standard, de-facto and official
- Accellera Systems Initiative
  - Formerly Open SystemC Initiative (OSCI)
  - Standardization body and consortium of leading companies
    - Synopsys, Cadence, CoWare, Frontier, ...
    - Intel, AMD, Qualcomm, Infineon, NEC, …
  - Open community

SystemC Overview

- System-Level Description Language
  - C++ class library, layered software architecture
  - Hierarchy of *modules* connected by *ports*
  - Communication via *interfaces* and *channels*
  - Discrete Event Simulation
- Abstraction Levels, Modeling Methodology
  - Untimed Model
  - Transaction-level Model
  - Bus-functional Model
  - Cycle-accurate Model

SystemC Overview

- Online Resources (EECS 222 course website)
  - Accellera Systems Initiative, SystemC Standardization Body
  - SystemC Standard Language Reference Manual
    - IEEE 1666-2011 (free download)
  - ➢ *SystemC: From the Ground Up (2nd edition)*
    - ➢ Text book (free download from UCI network)
  - SystemC 2.0:
    - Introduction, functional specification, user's guide
  - SystemC 2.1:
    - Overview and features
  - SystemC 2.3.1: (current version, installed on servers)
    - New features 2011
  - SystemC TLM-2.0:
    - Introduction, whitepaper, and requirements



Introduction to SystemC

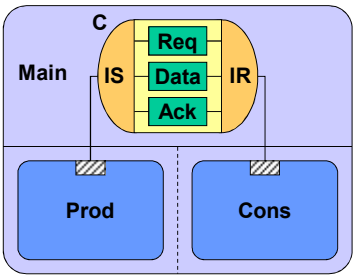
- Presentation by Stuart Swan, Cadence, 2002
  - Goals and Requirements
  - History and Organization
  - Versions, Contents, Coverage
  - Language Architecture
  - Modeling, Models of Computation, Examples
  - Communication Refinement
  - Outlook
- Example:
  - **`simple_fifo.cpp`**

Homework Assignment 3

- Task: Introduction to SystemC Language and Simulation
- Steps
  - SystemC library installed at **`/opt/pkg/systemc-2.3.1/`**
  - Study and simulate the **`simple_fifo`** reference example
  - Build and simulate a Producer-Consumer example
    - Producer **`Prod`** should send string **`"Apples and Oranges"`** character by character to the consumer **`Cons`**
    - Translate the SpecC model of Assignment 2 to SystemC
      - Reference model **`~eecs222/public/ProdCons.sc`**
    - Use the same structure and functionality
    - Use the same protocol channel with **`Ack`**, **`Req`**, and **`Data`**
      - Hint: Use delta notifications, e.g. **`Ack.notify(SC_ZERO_TIME)`**
- Deliverables
  - Source and log file: **`ProdCons.cpp`**, **`ProdCons.log`**
- Due
  - April 24, 2017, 12pm (noon!)

Homework Assignment 2 Solution

- Start with Sender-Receiver Example
  - Add behavior **`Main`** with structural connectivity
  - Sender **`S`** becomes **`Prod`**, with loop over string
  - Receiver **`R`** becomes **`Cons`**, with loop
  - Type **`float`** becomes **`char`**
  - Compile and simulate
  - Create log file and submit




```
behavior S(IS Port)
{
 float X;
 void main(void)
 { ...
   Port.Send(X);
   ...
 }
};
```

```
behavior R(IR Port)
{
 float Y;
 void main(void)
 {...
  Y=Port.Receive();
  ...
 }
};
```

```
interface IS
{
 void Send(float);
};
interface IR
{
 float Receive(void);
};
```

```
channel C
    implements IS, IR
{
 event Req;
 float Data;
 event Ack;

 void Send(float X)
 { Data = X;
   notify Req;
   wait Ack;
 }
 float Receive(void)
 { float Y;
   wait Req;
   Y = Data;
   notify Ack;
   return Y;
 }
};
```