

ECPS 203

Embedded Systems Modeling and Design

Lecture 6

Rainer Dömer

doemer@uci.edu

Center for Embedded and Cyber-physical Systems
University of California, Irvine



Lecture 6: Overview

- Introduction to IEEE SystemC
 - Overview
 - Resources
- SystemC: From the Ground Up (Part 1)
 - Introduction to SystemC
 - Core concepts and syntax
- Project Discussion
 - Assignment 2
 - Assignment 3

SystemC Overview

- SystemC System-Level Description Language
 - C++ class library, layered software architecture
 - Discrete event simulation
 - Hierarchy of *modules* connected by *ports*
 - Communication via *interfaces* and *channels*
 - IEEE Standard 1666-2011
- Abstraction Levels, Modeling Methodology
 - Untimed model
 - Transaction-level model
 - Bus-functional model
 - Cycle-accurate model

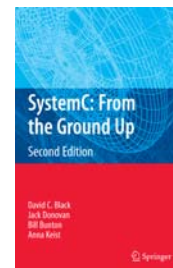
ECPS203: Embedded Systems Modeling and Design, Lecture 6

(c) 2017 R. Doemer

3

SystemC Overview

- Online Resources on ECPS 203 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, SystemC 2.1
 - Various resources about SystemC history
 - SystemC 2.3.1
 - Current version (installed on EECS servers)
 - SystemC TLM-2.0
 - Introduction, whitepaper, and requirements
 - Quick reference
 - SystemC quick-reference card



ECPS203: Embedded Systems Modeling and Design, Lecture 6

(c) 2017 R. Doemer

4

IEEE SystemC Language

- SystemC: From the Ground Up (Part 1)
 - **DAC15_systemC_Training.pdf**, slides 1 through 33 by David Black, Doulos
 - SystemC training day at Design Automation Conference 2015
 - *“The Definitive Guide to SystemC: The SystemC Language”*
 - Introduction to SystemC
 - Overview and background
 - Central concepts, SystemC World
 - Core Concepts and Syntax
 - Data
 - Modules and connectivity

ECPS203: Embedded Systems Modeling and Design, Lecture 6

(c) 2017 R. Doemer

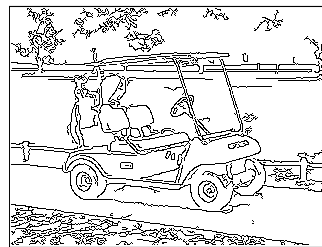
5

ECPS 203 Project

- Application Example: Canny Edge Detector
 - Embedded system model for image processing:
Automatic edge detection in a digital camera



golfcart.pgm



golfcart.pgm_s_0.60_l_0.30_h_0.80.pgm

- Application source and documentation:
 - http://marathon.csee.usf.edu/edge/edge_detection.html
 - http://en.wikipedia.org/wiki/Canny_edge_detector

ECPS203: Embedded Systems Modeling and Design, Lecture 6

(c) 2017 R. Doemer

6

Project Assignment 2

- Task: Clean C++ model with static memory allocation
 - Prepare the C++ source code for modeling in SystemC
 - Configure parameters for specific application
 - Apply static memory allocation
- Steps
 1. Fix the off-by-one bug in the `non_max_supp` function
 2. Clean-up the code for compilation without warnings
 3. Fix configuration parameters to compile-time constants
 4. Remove or replace dynamic memory allocation
- Deliverables
 - Source code and text file: `canny.cpp`, `canny.txt`
- Due
 - Wednesday, next week: October 18, 2017, 6pm

ECPS203: Embedded Systems Modeling and Design, Lecture 6

(c) 2017 R. Doemer

7

Project Assignment 3

- Task: Introduction to SystemC
 - Capture and simulate the introductory example by Doulos
- Steps
 1. Structural model is shown on slide 25
 2. Source file structure is shown on slide 32
 3. Capture the partial source code provided on slides 21-36
 4. Fill in the omitted source code for the monitor module
 - For test cases 1*6, 2*6, ..., 7*6, monitor and validate the output
 5. Simulate the model with Accellera SystemC library
- Deliverables
 - Source files, `Makefile`, `README` in `Multiplier.zip`
- Due
 - Wednesday, next week: October 25, 2017, 6pm

ECPS203: Embedded Systems Modeling and Design, Lecture 6

(c) 2017 R. Doemer

8