EECS 222A: System-on-Chip Description and 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

- Homework Assignment 3
  - Discussion
- System Validation using SCE
  - Simulation, Debugging, Tracing
- Homework Assignment 4
  - Creating behaviors from C code
  - Tasks

Homework Assignment 3

- Example Project
  - Elevator Control System (ECS)
    - Distributed embedded system
    - Set of communicating Elevator Control Units (ECU)
- Project Documentation:
  - D. Castellanos, R. Dömer:
    "System-Level Modeling and Simulation of an Elevator Control System",
    CECS Technical Report 07-04, June 2007.
  - http://www.cecs.uci.edu/~doemer/publications/CECS_TR_07_04.pdf

Homework Assignment 3

- Decomposition of ECS into multiple ECUs
  - Floor panel
    - panel at each floor and each shaft with up/down controls
  - Floor display
    - display of current floor and direction at each floor
  - Floor door
    - Control unit to open/close doors at each floor
  - Car panel
    - panel in each car with request controls
  - Car display
    - display of current floor and direction in each car
  - Car door
    - Control unit to open/close doors in each car
  - Main control unit
    - central control unit to control the entire ECS
  - Motor control unit
    - control unit for the motor atop each shaft

Homework Assignment 3

- Task
  - Develop a simple system specification model
    - such as one ECU in the ECS
    - the system should consist of
      - Test bench behavior Main
      - Stimulus behavior
      - Design Under Test (DUT)
      - Monitor behavior
    - in SpecC language
  - Simulate your model

Homework Assignment 3

- Deliverables
  - 1-page specification of your system
    - Illustration figure / schematic view of DUT with ports
    - Brief (!) description of functionality (in English)
  - Executable source code (in SpecC)
  - Successful simulation run
- Due
  - Week 6 (this week)

System Validation using SCE

- Simulation
  - **`scc DesignName –sc2out –vv –ww ./DesignName`**
  - Header file **`sim.sh`**
    - Access to simulation time
      - macros **`PICO_SEC`**, **`NANO_SEC`**, **`MICRO_SEC`**, **`MILLI_SEC`**, **`SEC`**
      - typedef **`sim_time`**, **`sim_delta`**, **`sim_time_string`**
      - function **`now()`**, **`delta()`**
      - conversion functions **`time2str()`**, **`str2time()`**
    - Handling of bit vectors
      - conversion functions **`bit2str()`**, **`ubit2str()`**, **`str2bit()`**, **`str2ubit()`**
    - Handling of long-long values
      - conversion functions **`ll2str()`**, **`ull2str()`**, **`str2ll()`**, **`str2ull()`**

System Validation using SCE

- Debugging
  - **`scc DesignName –sc2out –vv –ww –g -G`**
    `gdb ./DesignName`
    `ddd ./DesignName`
  - Header file **`sim.sh`**
    - Access to simulation engine state
      - functions **`ready_queue()`**, **`running_queue()`**, etc.
      - functions **`_print_ready_queue()`**, **`_print_running_queue()`**, etc.
      - function **`_print_process_states()`**
      - function **`_print_simulator_state()`**
    - Access to current instance
      - functions **`active_class()`**, **`active_instance()`**
      - functions **`current_class()`**, **`current_instance()`**
      - functions **`print_active_path()`**, **`print_current_path()`**
      - ...

System Validation using SCE

- Tracing
  - **`scc DesignName –sc2out –vv –ww -Tvcds`**
    `./DesignName`
    `gtkwave DesignName.vcd`
  - Trace instructions in file **`DesignName.do`**
  - Trace log in file **`DesignName.vcd`**
  - Waveform display **`gtkwave`**
    - available as **`/opt/gtkwave/bin/gtkwave`**
- Documentation:
  - E. Johnson, A. Gerstlauer, R. Dömer: *"Efficient Debugging and Tracing of System Level Designs"*, CECS Technical Report 06-08, May 2006.
  - http://www.cecs.uci.edu/~doemer/publications/CECS_TR_06_08.pdf

Homework Assignment 4

- Task
  - Creating behaviors in C code
    - initial system consists of
      - Test bench behavior Main
      - Stimulus behavior
      - Design Under Test (DUT)
      - Monitor behavior
    - we want to create more hierarchy in DUT
  - See posted detailed instructions!

Homework Assignment 4

- Deliverables
  - 1-paragraph description about the two tasks
    - How far did you get?
    - What were the problems?
    - How did you solve it?
  - Report the time stamps
    - How long did it take to do the tasks?
- Due
  - Week 7 (next week)