EECS 222: Embedded System Modeling Lecture 10

Rainer Dömer

doemer@uci.edu

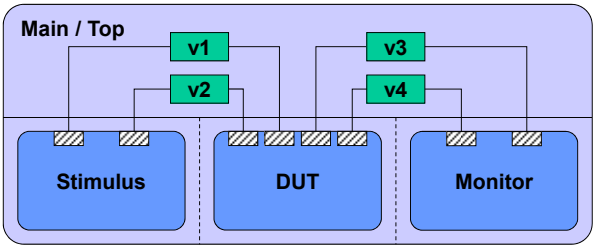
The Henry Samueli School of Engineering
Electrical Engineering and Computer Science
University of California, Irvine

Lecture 10: Overview

- Test Bench Modeling
  - Stimulus
  - DUT
  - Monitor
- Project Assignment 5
  - Video stream processing
  - Structural test bench model of the Canny Edge Detector
  - ➢ Discussion
  - ➢ Model development on the whiteboard

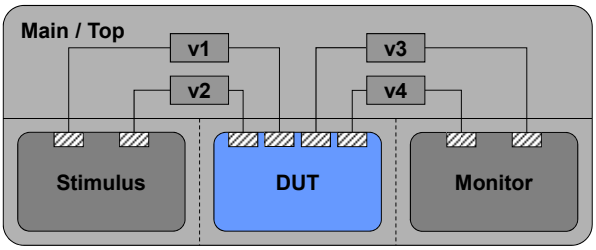
Test Bench Modeling

- Typical Test Bench
  - Top-level: **`Main`** behavior, or **`Top`** module
  - Stimulus: provides test vectors
  - Design Under Test (DUT): represents the target SoC
  - Monitor: observes and checks outputs
    - ➢ plus high-level connectivity via ports



Test Bench Modeling

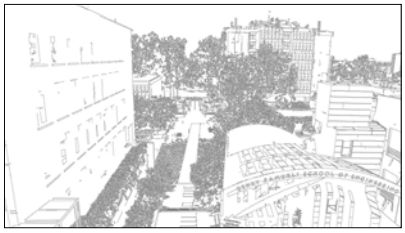
- Test Bench
  - Main / Top, Stimulus, Monitor
  - *Simulation only, no synthesis (no modeling restrictions)*
- DUT
  - Design Under Test
  - *Simulation and synthesis! (restricted by modeling guidelines!)*



EECS 222 Project

- Application Example: Canny Edge Detector
  - Embedded system model for image processing: Automatic Edge Detection in a Digital Video Camera

EngPlaza001.bmp



EngPlaza001_edges.pgm

  - Video taken by a drone hovering over UCI Engineering Plaza
    - Available on the server: **`~eecs222/public/video/`**
    - High resolution, 2704 by 1520 pixes
    - Video length 9 seconds, using 20 extracted frames for test bench model

Project Assignment 5

- Task: Structural Test Bench Model
  - Convert the application to process a stream of video frames
  - Choose either SpecC or SystemC for modeling and simulation
  - Add test bench structure to the model from Assignment 4
- Steps
  1. Convert the application to process a stream of video frames
  2. Create test bench structure: Stimulus, Platform, Monitor
  3. Create platform structure: DataIn, DUT, DataOut
- Deliverables
  - **`Canny.sc`** or **`Canny.cpp`** (choose one!)
  - **`Canny.txt`**
- Due
  - Next week: February 13, 2019, 6pm

Project Assignment 5

- Task: Structural Test Bench Model
  - Expected instance tree

```
Main / Top
|------ Stimulus stimulus
|------ Platform platform
|       |------ DataIn din
|       |------ DUT canny
|       \------ DataOut dout
\------ Monitor monitor
```

  - Communication via standard channels
    - SystemC: **`sc_fifo<IMAGE>`** based on class **`IMAGE`**
    - SpecC: **`c_img_queue`** based on typedef **`img`**
  - ➢ Pay attention to stack sizes!

Project Assignment 5

- Structural Test Bench for the Canny Edge Detector
  - Discussion on whiteboard: Top-level structure, platform for DUT