EECS 222: Embedded System Modeling Lecture 13

Rainer Dömer

doemer@uci.edu

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

Lecture 13: Overview

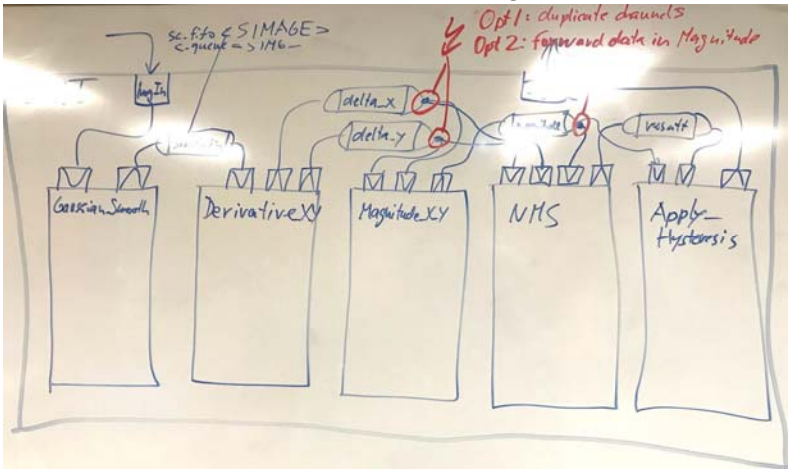
- Project Assignment 6
  - Refined structure of Gaussian Smooth
- System-on-Chip Environment (SCE)
  - Specification Modeling Guidelines
  - Design Example: GSM Vocoder
  - Interactive Demonstration

Project Assignment 6

- Task: Hierarchical DUT of the Canny Edge Detector
  - Refine the structural hierarchy of the DUT block
  - Refine the structural hierarchy of the Gaussian Smooth block
- Steps
  1. Refine the DUT structure
     - Gaussian Smooth, Derivative, …, Apply Hysteresis
  2. Refine the Gaussian Smooth structure
     - Receive Image, Gaussian Kernel, BlurX, BlurY
  3. Visualize the structural hierarchy of the model
- Deliverables
  - **`Canny.sc`** or **`Canny.cpp`** (choose one!)
  - **`Canny.txt`**
- Due: February 20, 2019, 6pm

Project Assignment 6

- Structural model of the DUT of the Canny Edge Detector
  - Discussion on whiteboard: Refined DUT structure



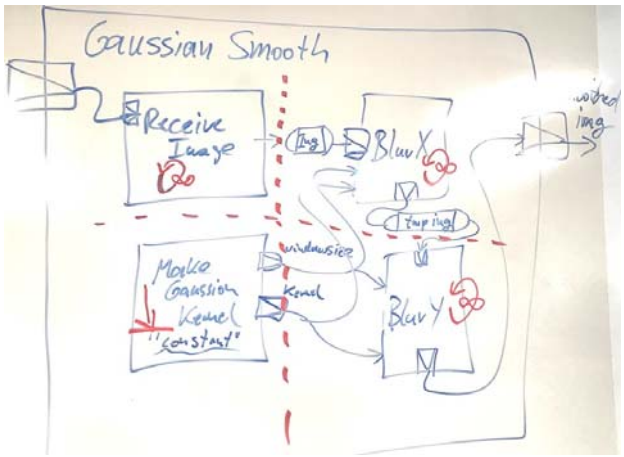
Project Assignment 6

- Step 2: Refined hierarchy of the Gaussian Smooth
  - Expected instance tree

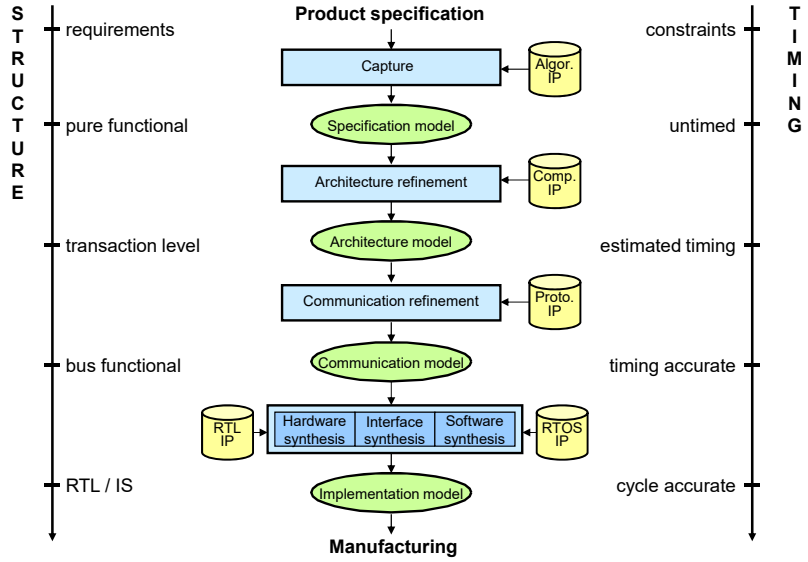
```
DUT canny
|------ Gaussian_Smooth gaussian_smooth
|       |------ Receive_Image receive
|       |------ Gaussian_Kernel gauss
|       |------ BlurX blurX
|       \------ BlurY blurY
|------ Derivative_X_Y derivative_x_y
|------ Magnitude_X_Y magnitude_x_y
|------ Non_Max_Supp non_max_supp
\------ Apply_Hysteresis apply_hysteresis
```


Project Assignment 6

- Structural model of the DUT of the Canny Edge Detector
  - Discussion on whiteboard: Refined Gaussian Smooth structure

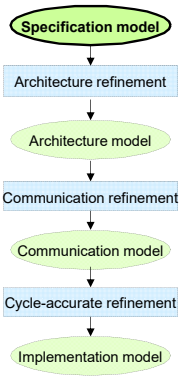


Top-Down Embedded Design Flow



Specification Model

- High-level, abstract model
  - Pure system functionality
  - Algorithmic behavior
  - No implementation details
- No implicit structure / architecture
  - Pure behavioral hierarchy
- Untimed
  - Execution in zero (logical) time
  - Causal ordering
  - Synchronization



(Source: A. Gerstlauer)

Specification Modeling Guidelines

- Example: Guidelines for SoC Environment (SCE)
  - Clean behavioral hierarchy
    - hierarchical behaviors:
      no code other than par, pipe, seq, fsm, and try-trap statements
    - leaf behaviors:
      Pure ANSI-C code (no SpecC constructs)
  - Clean communication
    - point-to-point communication via standard channels
    - ports of plain type or interface type, no pointers!
    - port maps to local variables or ports only
- Detailed rules for SoC Environment
  - CECS Technical Report:
    "*SCE Specification Model Reference Manual*"
    by A. Gerstlauer, R. Dömer, et al.
    - **`/opt/sce-20100908/doc/SpecRM.pdf`**

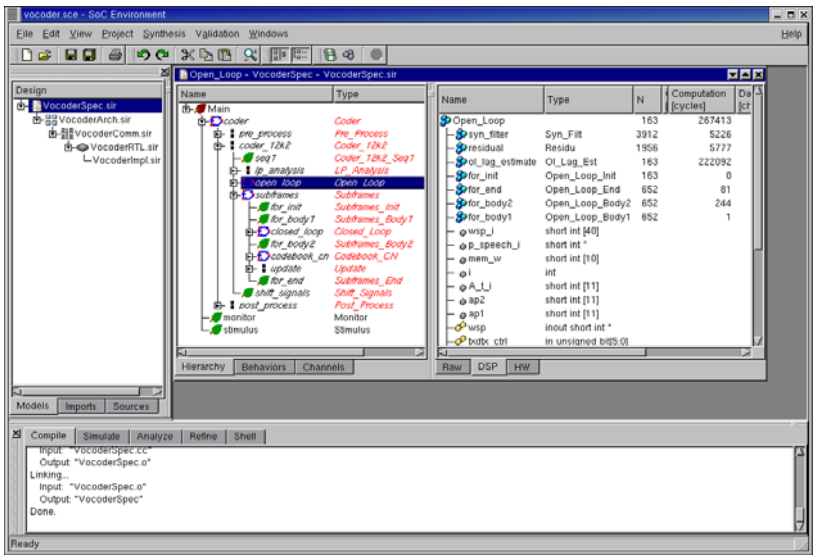
Specification Modeling Guidelines

- Converting C reference code to SpecC
  - Major functions become behaviors
  - Function call tree becomes behavioral hierarchy
    - Function call becomes behavior instance call
    - Sequential statements become leaf behaviors
    - Control flow becomes FSM
      - Conditional statements: **`if`**, **`if-else`**, **`switch`**
      - Loops: **`while`**, **`for`**, **`do-while`**
  - Explicitly specify potential parallelism!
  - Explicitly specify communication!
    - Use standard channels, avoid shared variables
    - No global variables
    - Only local variables in behaviors and functions/methods
  - Data types
    - Avoid dynamic memory allocation
    - Avoid pointers (arrays are preferred)
    - Use explicit data types if suitable (e.g. bit vectors)

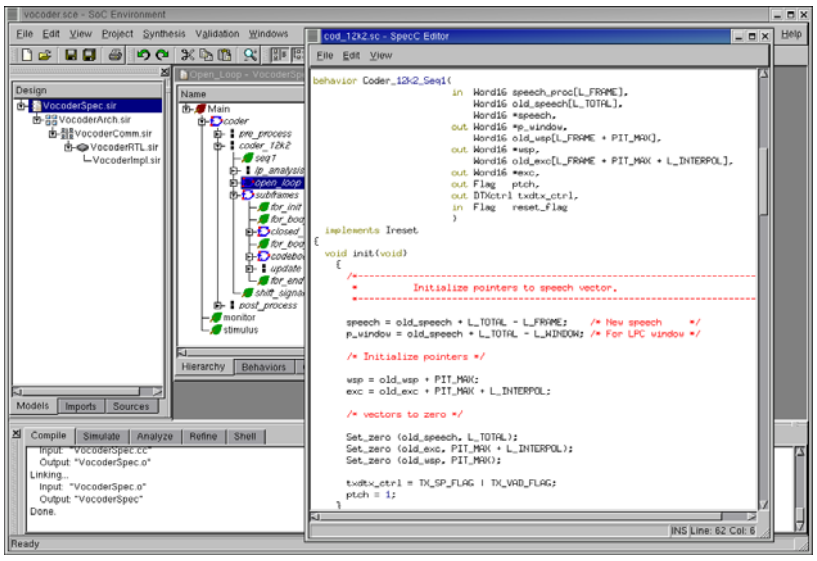
System-on-Chip Environment (SCE)

- Integrated Development Environment (IDE) with support of:
  - Graphical frontend (**`sce, scchart`**)
  - SLDL-aware editor (**`sced`**)
  - Compiler and simulator (**`scc`**)
  - Profiling and analysis (**`scprof`**)
  - Architecture refinement (**`scar`**)
  - RTOS refinement (**`scos`**)
  - Communication refinement (**`sccr`**)
  - RTL refinement (**`scrtl`**)
  - Software refinement (**`sc2c`**)
  - Scripting interface (**`scsh`**)
  - Tools and utilities (**`sir_list, sir_tree, …`**)

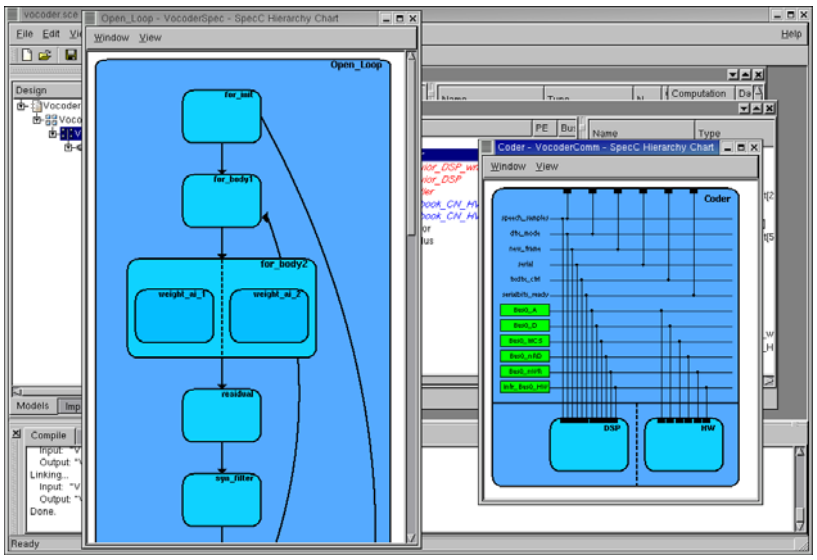
SCE Main Window



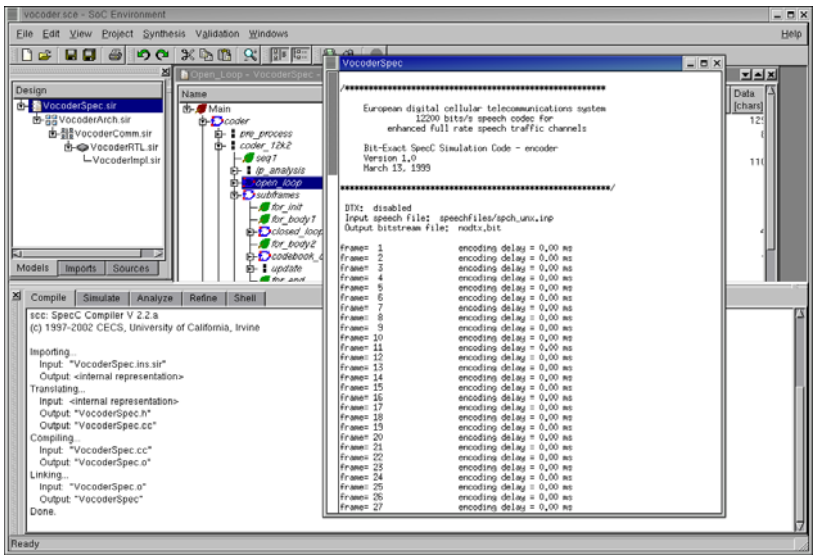
SCE Source Editor



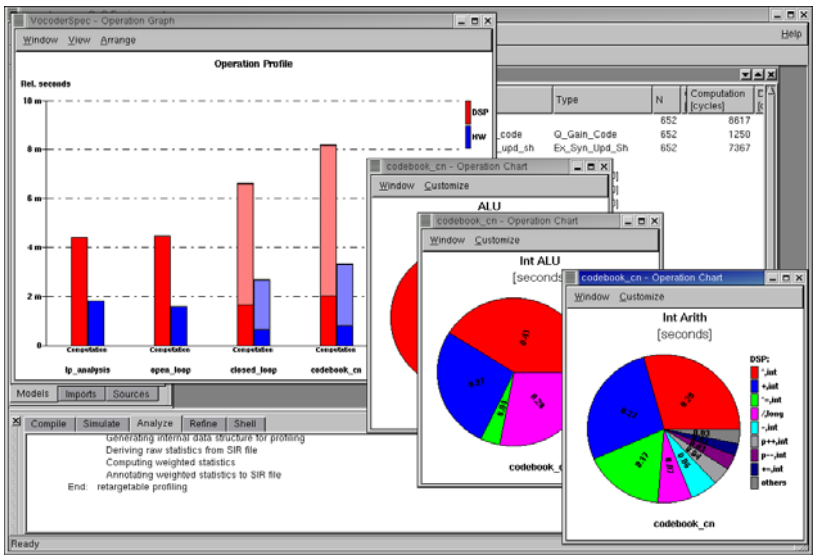
SCE Hierarchy Displays



SCE Compiler and Simulator



SCE Profiling and Analysis



SCE Demonstration

- Application Example: GSM Vocoder
  - Enhanced full-rate voice codec
  - GSM standard for mobile telephony (GSM 06.10)
  - Lossy voice encoding/decoding
    - Incoming speech samples @ 104 kbit/s
    - Encoded bit stream @ 12.2 kbit/s
    - Frames of 4 x 40 = 160 samples (4 x 5ms = 20ms of speech)
  - Real-time constraint:
    - max. 20ms per speech frame
      (max. total of 3.26s for sample speech file)
  - SpecC specification model
    - 29 hierarchical behaviors (9 par, 10 seq, 10 fsm)
    - 73 leaf behaviors
    - 9139 formatted lines of SpecC code
      (~13000 lines of original C code, including comments)

