EECS 222C: System-on-Chip Software Synthesis 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

- Assignment 4
  - Discussion
- Assignment 5
- Embedded Software
  - Scheduling algorithms
    - Aperiodic scheduling
    - HLS scheduling

Assignment 4

1. Become familiar with the System-on-Chip Environment (SCE)
   - Setup
     - Note that we will use the 2003 version of SCE for the tutorial:
     - **`source /opt/sce-20030530/bin/setup.csh`**
     - **`rm –rf ~/.sce`**
     - **`mkdir demo`**
     - **`cd demo`**
     - **`setup_demo`**
   - Open the SCE Tutorial document
     - **`acroread SCE_Tutorial/sce-tutorial.pdf &`**
     - To protect the environment and save some trees, please *do not print* the tutorial document! It contains 250 pages and you will likely read it only once… ;-)
   - Follow the SCE Tutorial instructions
     - **`sce &`**
     - **`...`**
   - Cleanup
     - When done (or to start over), clean up your demo directory
     - **`cd ..`**
     - **`rm –rf demo`**

Assignment 4

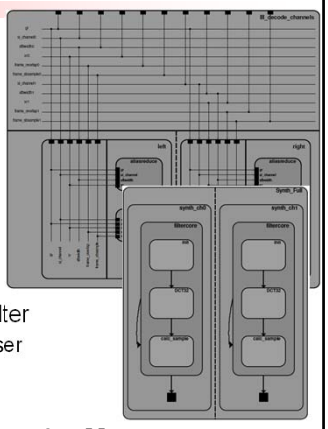
2. Setup your MP3 Decoder model in SCE
   - Setup SCE
     - Note that we will use the 2010 version of SCE:
     - **`source /opt/sce-20100908/bin/setup.csh`**
     - **`rm –rf ~/.sce`**
     - **`ln -s hw3 hw4`**
     - **`cd hw4`**
     - **`sce &`**
   - Create a new project in SCE
     - **`Project->New`**
     - **`Project->Settings`**
       - Set include path to "**`.`**" (current directory)
       - Set libraries to "**`-xl huffman.o`**"
       - Set both verbosity and warning level to 2
       - In the Simulator tab, set the simulation command as follows (single line!):
         `./%e testStream/spot1_3K.mp3 spot1_3K.pcm &&`
         `diff reference/spot1_3K.pcm spot1_3K.pcm`
     - **`Project->SaveAs "mp3.sce"`**

Assignment 4

3. Compile and simulate your MP3 Decoder model in SCE
   - … (continued from previous page)
   - Load your design model into SCE
     - **`File->Import "testbench.sc"`**
     - **`Project->AddDesign`**
     - Right-click on **`testbench.sir`** in the project window, and **`Rename`** the model to **`Spec`**
   - Compile and simulate your model in SCE
     - **`Validation->Compile`**
     - **`Validation->Simulate`**

Assignment 4

4. Study your MP3 decoder model in SCE
   - … (continued from previous page)
   - Browse the structural hierarchy charts
     - Select a behavior in the behavior browser
     - Right-click **`->Chart`**
     - Double-click to add a level of hierarchy
     - **`View->Connectivity`**
     - **`...`**
   - Print the hierarchy chart for the Synthesis Filter
     - Select the **`Synth_Full`** behavior in the browser
     - Right-click **`->Chart`**
     - Add all levels of hierarchy, but no connectivity
     - **`Window->Print…`** in color (!) to file **`Chart_SynthFull.ps`**
   - Print the hierarchy chart for the Channel Decoding
     - Display the chart of the **`III_decode_channels`** behavior
     - Add all levels of hierarchy, including connectivity
     - **`Window->Print…`** in color (!) to file **`Chart_DecodeChannels.ps`**

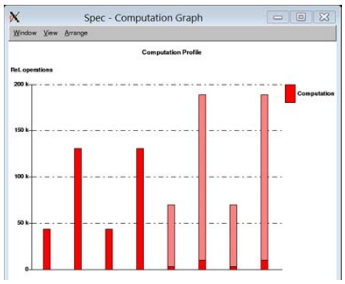


Assignment 5

1. Profile your MP3 Decoder model in SCE
   - (continued from previous assignment)
   - Load your MP3 project in SCE
     - **`Project->Load "mp3.sce"`**
   - Open your "Spec" design model and validate it
     - Double-click on **`Spec.sir`** in the project window
     - **`Validation->Compile`**
     - **`Validation->Simulate`**
   - Profile your MP3 decoder in SCE
     - **`Validation->Profile`**

Assignment 5

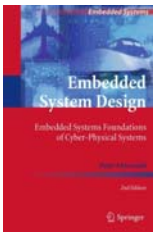
2. Analyze your Profiling Results
   - Use the SCE bar charts to compare the computational complexity of the behaviors in your MP3 decoder model
     - In the hierarchy browser, select behaviors of interest (use CTRL-LeftClick to select/deselect)
     - **`RightClick->Graphs->Computation`**
   - Identify the behavior instances with the most computational load
     - Goal is to find those components that make good candidates for hardware accelleration
       - Short code
       - Regular structure
       - High computation
     - Hint: There are 8 candidates as shown in the chart on the right!
   - Deliverable
     - **`ComputationProfile.pdf`**



Example Computation Profile (block names omitted)

Embedded Software

- Scheduling Algorithms
  - Aperiodic Scheduling
  - HLS Scheduling



➢ Excerpts from Chapter 6.2 in
  - *"Embedded System Design"*
    Embedded Systems Foundations
    of Cyber-Physical Systems
    by P. Marwedel,
    2nd edition, Springer, 2011.
  - ➢ **`Lecture6-subset-es-marw-6.1-aperiodic.ppt`**
  - ➢ **`Lecture6-subset-es-marw-6.2-hls-scheduling.ppt`**