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

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 for the MP3 decoder:
     - **`source /opt/sce-20100908/bin/setup.csh`**
     - **`rm –rf ~/.sce`**
     - **`cd mad_SpecC`**
     - **`sce &`**

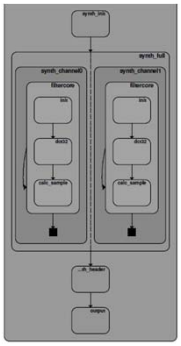
   - Create a new project in SCE
     - **`Project->New`**
     - **`Project->Settings`**
       - Set include path to "**`.`**" (current directory)
       - Set libraries to "**`-xl huffman.o`**"
       - Set both verbosity and warning level to 2
       - In the Simulator tab, set the simulation command 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. Analyze your MP3 decoder model in SCE
   - … (continued from previous page)
   - Browse the structural hierarchy charts
     - Select the **`Main`** behavior in the behavior browser
     - Right-click **`->Chart`**
     - Double-click the chart to add a level of hierarchy
     - **`View->Connectivity`**
     - **`...`**
   - Print the hierarchy chart for the Synthesis Filter
     - Select the **`synth`** behavior in the behavior browser
     - Right-click **`->Chart`**
     - Add several levels of hierarchy
     - **`Window->Print…`** in color (!) to file **`"synth.ps"`**
     - **`ps2pdf synth.ps`**



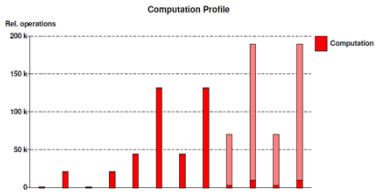
- Deliverable
  - Hierarchy chart **`"synth.pdf"`** (in color!)
  - by Friday, Oct 29, 2010, at noon
  - by email to **`doemer@uci.edu`** with subject "EECS222C HW4"

Assignment 5

1. Profile your MP3 Decoder model in SCE
   - (continued from previous assignment)
   - Load your MP3 project in SCE
     - **`Project->Load "mp3.sce"`**
   - 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`**
   - Profile your MP3 decoder in SCE
     - **`Validation->Profile`**

Assignment 5

2. Analyze your Profiling Results
   - Use the graphical bar charts to compare the complexity of the behaviors in your MP3 decoder
     - In the hierarchy browser, select behaviors of interest (use CTRL-LeftClick to select/deselect)
     - **`RightClick->Graphs->Computation`**
   - Determine the most-critical behaviors that contribute the most computation operations
     - The goal is to find those behavioral blocks that make good choices for hardware accelleration
   - Deliverable 1:
     - Bar chart showing the selected behaviors in comparison to others
       - **`CriticalBlocks.pdf`**
     - Text file briefly (!) explaining your choice
       - **`CriticalBlocks.txt`**



Example Computation Profile (block names omitted)

Assignment 5

3. Evaluate potential Processors for SW-only Implementation
   - Select DUT as **`Mad_decoder decoder`**
     - RightClick on **`decoder ->SetAsTop-Level`**
   - Consider an ARM7TDMI processor (50MHz)
     - **`Synthesis->Allocate PEs…`**
     - Add Processors, **`ARM_7TDMI`**
     - Choose default port configuration (i.e. 20000ps)
     - Choose 50 MHz (change it from default 100MHz)
     - Name the processor **`ARM7TDMI`**
   - Map the entire decoder on to the ARM7TDMI processor
     - **`Validation->Evaluate`**
     - **`Validation->Show Estimates`**
   - Determine the estimated execution time on the ARM7TDMI!

Assignment 5

4. Evaluate alternative Processors for SW-only Implementation
   - Consider as alternative a LEON3 processor (50MHz)
     - **`Synthesis->Allocate PEs…`**
     - Add Processors, **`LEON3`**
     - Choose default port configuration (i.e. 20000ps)
     - Choose default clock frequency (i.e. 50 MHz)
     - Name the processor **`LEON3`**
   - Map the entire decoder on to the LEON3 processor
     - **`Validation->Evaluate`**
   - Determine the estimated execution time on the LEON3!
   - Deliverable 2:
     - Text file with the estimated execution times for the ARM7TDMI and LEON3 processors, and
     - Brief analysis whether or not each processor is expected fast enough for a SW-only implementation of the MP3 decoder
       - **`SWonly.txt`**
   - Due:
     - by Friday, Nov 5, 2010, at noon (email to **doemer@uci.edu,** "EECS222C HW5")

Embedded Software

- Chapter 4, part 2, of
  "Embedded System Design"
  by P. Marwedel (Univ. of Dortmund, Germany),
  Kluwer Academic Publishers, 2003.

  – Scheduling Algorithms
  – Aperiodic Scheduling

  ➢ **`Lecture6-es-marw-4b-aperiodic.ppt`**