EngrECE 298
System-on-Chip Description and Modeling
Spring 2004

Assignment 4

Posted: May 27, 2004 (week 8)
Due: June 4, 2004 (week 9)

Task: Design Exploration and Refinement

Instructions:

The goal of this assignment is to take the design specification developed in the previous assignments (or, as an alternative, the GSM Vocoder design from the SCE Tutorial), explore different design options, determine one "best" candidate, and finally bring the design down to a valid implementation.

We will use the of the System-on-Chip Environment (SCE) for these tasks. More specifically, we will use the latest stable version of SCE, sce-20040121. To set this up, use

```
source /opt/sce-20040121/bin/setup.csh
```

on the server **`alpha.eecs.uci.edu`**.

Also, we will use a special component library that has been combined from different sources. This library is installed under **`/opt/sce_big_db-20040121/db/`**. To use it, set the following paths in SCE:

```
BusPath=/opt/sce_big_db-20040121/db/busses/busses.sir
CEPath=/opt/sce_big_db-20040121/db/communication/communication.sir
PEPath=/opt/sce_big_db-20040121/db/processors/processors.sir
RTLPath=/opt/sce_big_db-20040121/db/rtl/rtl.sir
```

You may want to set these paths in the file **`~/.sce/scerc`**, but please note that this file is overwritten every time you run the **`setup_demo`** script of SCE.

Task 1: Select the design example

For this assignment, you can choose from the following two design examples:

Design 1: MPEG-Audio Decoder (from previous homework)
Design 2: GSM Vocoder (from SCE tutorial)

Task 2: Explore different design alternatives with SCE.

The goal of design space exploration is to find the “best” design among the possible alternatives. Which one is the “best” depends on your design goals. You may want to target a minimal number of components, minimal cost of components, minimal power consumption, simplicity, or any other goals, as well as any combination of these. However, note that in both design examples, the execution time is given as a design constraint, so minimal time cannot be an optimization goal (it is a requirement).

As discussed in the lectures, design space exploration with SCE is composed of several refinement steps, and at each step different decisions need to be made. In the following, some hints are given towards the refinement steps (but you are free to make any choices towards your chosen design goal):

- Architecture Refinement: The PE database contains only a small number of fully-specified components, as follows:
    - DSP/Motorola_DSP56600
    - Processor/Motorola_Coldfire
    - Processor/Toshiba_TX49H2
    - Memory/Motorola_Coldfire_SRAM
    - Memory/Samsung_KM684002A
    - Custom Hardware/HW_Standard
- The MPEG-Audio design requires floating-point operations which the DSP56600 is not able to perform. On the other hand, the Vocoder design requires saturated, fixed-point operations which are not natively supported by the Coldfire and TX49 processors.
- In addition to choose different PEs, you may also want to adjust the clock frequency of the PEs in the allocation dialog. Reduced clock speeds help save energy!
- Explore different mappings of behaviors to PEs (and variables to memory).

- Scheduling Refinement: Explore the effects of dynamic vs. static scheduling on processors.

- Communication Refinement: The communication refinement stage is, in this version of SCE, separated into two steps, namely network refinement and link refinement. Both tools are in experimental stage. Also, the communication components available are very limited. So, to make things work, choose the native processor bus as the system bus and make the processor the master of this bus. Other options may or may not work.

- RTL Refinement: Only standard RTL components are available for implementation of the HW PEs. So, while a straightforward implementation should be possible without problems, not many alternatives for exploration do probably exist.

- Software Refinement: For none of the processors listed above an instruction set simulator is available in the current installation of SCE. As a result, we will have to limit SW refinement to C code generation.

Task 3: Generate the “best” system architecture for the design.

By using the analysis capabilities of SCE (i.e. the profiling and estimation tools), determine the “best” choice for your design goal. Then, apply your design decisions and generate the refined design model by using the SCE refinement tools.

At the end of this process, as a result, you should have obtained a working implementation model of your chosen design.

--
Rainer Doemer (ET 444C, x4-9007, doemer@uci.edu)