EECS 222C
System-on-Chip Software Synthesis
Spring 2013

Assignment 6

Posted: May 17, 2013
Due: May 24, 2013 at 12pm (noon)

Topic: Evaluate the ARM7TDMI Processor as a candidate for a SW-only implementation of the MP3 Decoder model

1. Setup:

This assignment builds upon the previous assignments (data in your **hw3** directory). For submission purposes, we will create a symbolic link **hw6** that points to **hw3**, as follows:

```
ln –s hw3 hw6
cd hw6
```

Again, we will use the System-on-Chip Environment SCE version 2010. Run the setup script, as follows:

```
source /opt/sce-20100908/bin/setup.csh
```

To avoid incompatibility problems with other SCE versions (i.e. the version used in the tutorial), delete the **.sce** directory in your home directory before starting SCE:

```
rm –rf ~/.sce
sce &
```

2. Task: Refine the MP3 Decoder in SCE down to an ISS Model

We will resume the design flow from the previous Assignment 5 and step-wise refine the model down to a cycle-accurate Instruction Set Simulation (ISS) model.

Detailed instructions on the model refinement are available on the server in file: **/home/eecs222/EECS222C_S13/Assignment6.txt**

The same instructions follow below:

```
Step 1: Architecture Refinement
  -> continue from the "Spec" model of Assignment 5
  -> set top-level: MP3Decoder mp3decoder
  -> allocate 1 ARM_7TDMI CPU
        -> default port parameters (i.e. 20000ps)
        -> reduce clock frequency to 50MHz
        -> name the PE "ARM7"
  -> allocate 2 custom hardware I/O units of type HW_Virtual
        -> keep clock frequency at 100MHz
        -> name the PEs "MP3_IN" and "PCM_OUT"
  -> Synthesis->ShowChannels
  -> map the behaviors to the target architecture
        -> map MP3Decoder to ARM7
        -> map Mad_Stimulus to MP3_IN
        -> map channel stream_in to MP3_IN
        -> map Mad_Monitor to PCM_OUT
        -> map channel pcm_out to PCM_OUT
     -> Validation->Evaluate
     -> Validation->ShowEstimates
     => Look up the estimated execution time for decoder on ARM7,
        calculate the decode time per frame and note that in the table!
  -> Synthesis->ArchitectureRefinement
        -> default options
  -> Project RightClick->Rename "Arm7Arch"
  -> Validation->Compile
  -> Validation->Simulate
     => Note the decode time per frame in your time table!
        (it ignores sequentialized parallelism)

Step 2: Scheduling Refinement
  -> Synthesis->ScheduleBehaviors
     -> select Dynamic Scheduling "None" for ARM7 (not needed for MP3)
        -> select ARM_7TDMI_20000_0_ARM7, at top-level RightClick-
>SerializeTree
     -> leave I/O units alone (MP3_IN, PCM_OUT)
  -> Synthesis->SchedulingRefinement
        -> default options
  -> Project RightClick->Rename "Arm7Sched"
  -> Validation->Compile
  -> Validation->Simulate
     => Note the decode time per frame in your time table!
        (it ignores communication)

Step 3: Network Refinement
  -> Synthesis->AllocateNetwork
    -> for Busses, rename pre-allocated AMBA bus to "MainBus"
    -> for CEs, none are needed (nothing to do)
    -> for Connectivity, connect
        - ARM7,PortA to MainBus as master0
        - MP3_IN,Port0 to MainBus as slave1
        - PCM_OUT,Port0 to MainBus as slave2
  -> Synthesis->NetworkRefinement
        -> default options
  -> Project RightClick->Rename "Arm7Net"
```

```
  -> Validation->Compile
  -> Validation->Simulate
     => Note the decode time per frame in your time table!
        (it assumes infinite network speed)

Step 4: Communication Link Refinement
  -> Synthesis->AssignLinkParameters
    -> for c_link_ARM7__MP3_IN use address 0x10000000 to 0x10000007,
       and polling every 0.0 ms (LEON3 ISS does not support interrupts)
    -> for c_link_ARM7__PCM_OUT use address 0x20000000 to 0x20000007,
       and polling every 0.0 ms (LEON3 ISS does not support interrupts)
    -> for c_link_ARM7__PCM_OUT_1 use address 0x20000008 to 0x2000000B
  -> Synthesis->CommunicationRefinement
        -> choose Transaction-level model, default options
  -> Project RightClick->Rename "Arm7TLM"
  -> Validation->Compile
  -> Validation->Simulate (be patient!)
     => Note the decode time per frame in your time table!
        (it includes TLM communication of 0.107 ms)

  -> go back and select "Net" model again
  -> Synthesis->CommunicationRefinement
        -> choose Pin-accurate model, default options
  -> Project RightClick->Rename "Arm7PAM"
  -> Validation->Compile
  -> Validation->Simulate
     => Note the decode time per frame in your time table!
        (it includes PAM communication of 0.107 ms)

Step 5: Software synthesis (C code generation)
  -> continue from the "TLM" model generated in Step 4
  -> Synthesis->CCodeGeneration
    -> choose "C code reintegration"
    -> leave other options as per default
  -> Project RightClick->Rename "Arm7TLM_C"
  -> Inspect generated C code
    -> in shell where SCE was started:
       % vi ARM7/ARM7.h
       % vi ARM7/ARM7.c
       % vi ARM7/Makefile
    -> in tree, select leaf mp3decoder->ARM7->hw_TLM->hal->ARM7
       -> View->Source
  -> Validation->Compile
  -> Validation->Simulate
     => Note the decode time per frame in your time table!
        (it includes TLM communication of 0.107 ms)

Step 6: ISS integration
  -> continue from the "PAM" model generated in Step 4
  -> Synthesis->CCodeGeneration
    -> choose "Instruction-set simulation"
    -> leave other options as per default
  -> Project RightClick->Rename "Arm7ISS"
  -> Adjust generated Makefile
    -> in shell where SCE was started:
       % cd ARM7
       % cp ../huffman.h .
```

```
      % cp ../huffman.c .
      % vi Makefile
        -> replace "USR_SRC := ARM7.c"
           with    "USR_SRC := ARM7.c huffman.c"
        -> replace "USR_LFLAGS := huffman.o"
           with    "USR_LFLAGS :="
-> Cross-compile the generated code for the ARM7 processor
  -> in shell where SCE was started:
     % make
-> Validation->Compile
-> Validation->Simulate
   => Note the decode time per frame in your time table!
      (it includes PAM communication of 1.085 ms and may vary!)
```

As deliverable for this assignment, note the estimated/simulated frame delays in a table as follows:

Refinement Step	Model	Decode time per frame
Profiling estimation	Spec	
Architecture Refinement	Arm7Arch	
Scheduling Refinement	Arm7Sched	
Network Refinement	Arm7Net	
Transaction-Level Refinement	Arm7TLM	
C Code Generation	Arm7TLM_C	
Pin-Accurate Refinement	Arm7PAM	
Instruction Set Simulation	Arm7ISS	

For submission, convert your table into a PDF file. Name the PDF file **`ARM7_Evaluation.pdf`**, place it into your **`hw6`** directory, and make it readable for the submission script.

```
chmod 644 ARM7_Evaluation.pdf
```

Use exactly this filename, otherwise you can't submit.

3. Submission:

For this assignment, submit the following deliverable:

`ARM7_Evaluation.pdf`

The file should be placed in your **`hw6`** directory. Then, in its parent directory, enter **`turnin`**.

As in the previous assignments, the **`turnin`** command will locate the deliverables and allow you to submit them *before the deadline.*

Again, you can submit at any time before the deadline, *but not after!* You can also submit as many times as you want. Newer submissions will overwrite older ones.

Late submissions will not be accepted!

--
Rainer Doemer (EH3217, x4-9007, doemer@uci.edu)