EECS 222C: System-on-Chip Software Synthesis Lecture 7

Rainer Dömer

doemer@uci.edu

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

Lecture 7: Overview

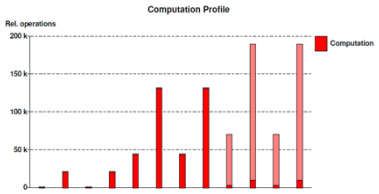
- Assignment 5
  - Discussion
- Refinement-based System Design Flow
- Example Design Study
- Assignment 6

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

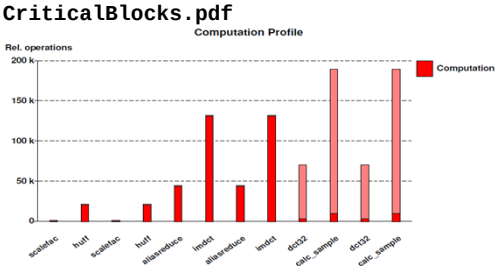
3. Evaluate potential Processors for SW-only Implementation
   - Select DUT as **`Mad_decoder decoder`**
     - RightClick on **`decoder ->SetAsTop-Level`**
   - Consider an ARM7TDMI processor (50MHz)
     - **`Synthesis->Allocate PEs…`**
     - Add Processors, **`ARM_7TDMI`**
     - Choose default port 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")

Assignment 5: Discussion

1. Bar chart showing the most-critical behaviors in terms of computation operations
   - Find blocks suitable for HW acceleration!
     - **`CriticalBlocks.pdf`**



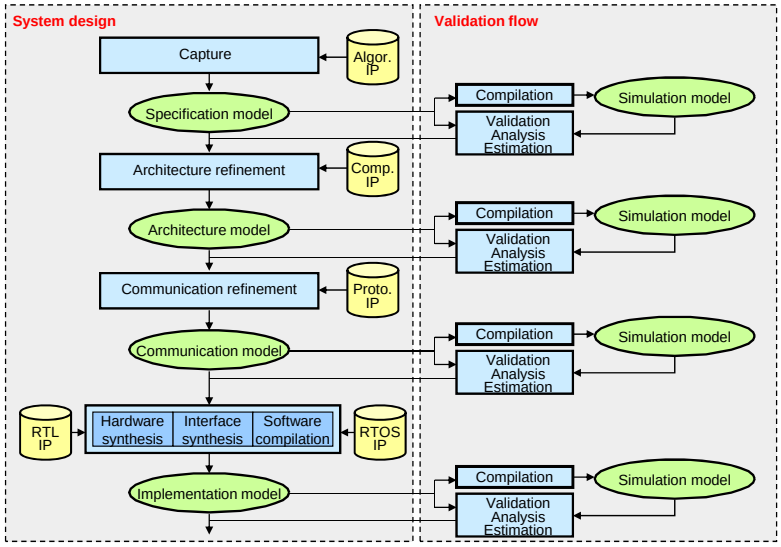
     - **`CriticalBlocks.txt`**
       - Starting from the **`decoder`** DUT, traverse down the hierarchy to find behaviors with most significant computation…
       - **`decode_frame/layer_III/decode/granule/channels`**
         - » **`left/imdct`**
         - » **`right/imdct`**
       - **`synth/synth_full/synth_channel{01}/filtercore`**
         - » **`dct32`**
         - » **`calc_sample`**

Assignment 5: Discussion

2. Evaluate potential Processors for SW-only Implementation
   - Deliverable 2:
     - Estimated execution times for the ARM7TDMI and LEON3 processors
     - Brief analysis whether or not each processor is expected fast enough
       - **`SWonly.txt`**
   - Candidate 1: ARM7TDMI processor at 50 MHz
     - Profiler-estimated execution time: **`206.0 ms`**
   - Candidate 2: LEON3 processor at 50 MHz
     - Profiler-estimated execution time: **`733.3 ms`**
   - Is this fast enough?
     - Length of test stream **`spot1_3k.pcm:`**
       `36864` bytes **`= 8`** frames **`* 1152`** samples **`* 2`** channels **`* 2`** bytes per sample
     - Monitor source code **`mp3monitor.sc:`**
       `44.1 kHz` stereo stream **`(`**at **`2`** bytes per sample**`)`**
     - Timing constraints:
       Per frame: **`1152 / 44.1 kHz = 26.12 ms`**
       For entire test stream: **`8*1152 / 44.1 kHz = 208.98 ms`**
     - ➢ **`ARM7TDMI`** processor barely meets the timing constraint
     - ➢ **`LEON3`** processor misses the timing constraints (by more than 2x)

   - ➢ Hardware accelleration becomes necessary!

Refinement-based System Design Flow



Refinement-based System Design Flow

- Step 1: Architecture Refinement
  - Allocation of Processing Elements (PE)
    - Type and number of processors
    - Type and number of custom hardware blocks
    - Type and number of system memories
  - Mapping to PEs
    - Map each behavior to a PE
    - Map each channel to a PE
    - Map each variable to a PE
  - Result:
    System architecture of concurrent PEs
    with abstract communication in channels

Refinement-based System Design Flow

- Step 2: Scheduling Refinement
  - For each PE, serialize the execution of behaviors to a single thread of control
  - Option (a): Static scheduling
    - For each set of concurrent behaviors, determine fixed order of execution
  - Option (b): Dynamic scheduling by RTOS
    - Choose scheduling policy, i.e. Round-robin or priority-based
    - For each set of concurrent behaviors, determine scheduling priority
  - Result:
    System model with abstract RTOS scheduler inserted in each PE

Refinement-based System Design Flow

- Step 3: Communication Refinement
  - Allocation of system busses
    - Type and number of system busses
    - Type of bus protocol for each bus (if applicable)
    - Number of transducers (if applicable)
    - System connectivity
  - Mapping of channels to busses
    - Map each communication channel to a system bus (or multiple busses, if applicable)
  - Result:
    Bus-functional model of the system

Refinement-based System Design Flow

- Step 4: Hardware Refinement (for HW PE)
  - Allocation of Register Transfer Level (RTL) components
    - Type and number of functional units (e.g. adder, multiplier, ALU)
    - Type and number of storage units (e.g. registers, register files)
    - Type and number of interconnecting busses (drivers, multiplexers)
  - Scheduling
    - Basic blocks assigned to super-states
    - Individual operations assigned to states (clock cycles)
  - Binding
    - Bind functional operations to functional units
    - Bind variables to storage units
    - Bind assignments/transfers to busses
  - Result:
    Clock-cycle accurate model of each HW PE
  - Output: Synthesizable Verilog description

Refinement-based System Design Flow

- Step 5: Software Refinement (for SW PE)
  - C code generation
    - For selected target processor
  - RTOS targeting
    - For selected target RTOS
  - Compilation to Instruction Set Architecture
    - for Instruction Set Simulation (ISS)
  - Assembly
  - Result:
    Clock-cycle accurate model of each SW PE
  - Output: downloadable binary image

Example Design Study

- Design of a MP3 Decoder
  - Floating-point algorithm
    - Somewhat different from ours
    - Still very appropriate to compare
- Design Study
  - Tutorial on Embedded System Design
    - Topic: System-level Modeling
    - Speaker: Andreas Gerstlauer, CECS
    - Conference: ASP-DAC 2007, Yokohama, Japan
    - **`Lecture7-ASPDAC07-AG-MP3.pdf`**

Assignment 6

1. Refinement of the MP3 Decoder model in SCE
   - Continue from "Spec" model of previous assignment
   - Select an appropriate system architecture
     - 1 ARM_7TDMI or LEON3 CPU at 50 MHz
     - 0-6 HW_Standard accellerators at 100 MHz
   - Perform the following refinement steps
     - Architecture Refinement
     - Scheduling Refinement
     - Network Refinement
     - Communication Link Refinement
       - Transaction-level model (TLM)
       - Pin-accurate model (PAM)
     - Reference instructions:
       - **`/home/doemer/EECS222C/Assignment6.txt`**
   - Deliverable:
     - Text file "**`refinement.txt`**" with
       - short description of the chosen target architecture
       - simulated execution times of each model
   - Due:
     - by Friday, Nov 12, 2010, at noon
     - by email to **doemer@uci.edu**, with subject "EECS222C HW6"