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

- Project Discussion
  - Assignment 5
  - Design space exploration
    - Estimation results
    - Timing back-annotation
  - Assignment 6
    - Architecture Exploration
    - Scheduling Exploration
- Embedded Software
  - Scheduling algorithms
    - Periodic scheduling
      - RMS, EDF
    - Priority inversion
      - Mars Pathfinder Example

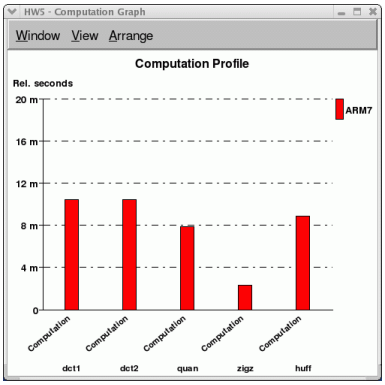
Assignment 5

1. Examine the "perfect" JPEG Encoder source code
   - **`/home/doemer/EECS222C_F08/jpegencoder.sc`**
2. Examine the "perfect" JPEG Encoder model in SCE
   - Setup
     - Same as before (use SCE version 20080601)
   - Browse the structural hierachy
   - View the hierarchy chart
   - Validate the model (compile and simulate)
   - Profile, analyze, estimate the model
     - For a single ARM_7TDMI CPU
     - For complexity of "Computation"

- Deliverables
  - Bar graph of Computation Profile
    - **`"ARM7TDMI.ps"`**
- Due
  - by Friday, Nov 7, 2008, at noon
  - by email to **`doemer@uci.edu`** with subject "EECS222C HW5"

Project Discussion

- Design Space Exploration
  - Estimation results
    - For ARM_7TDMI CPU at 100 MHz
    - For encoding of 180 blocks
      - ChenDCT1: 10.41ms
      - ChenDCT2: 10.41ms
      - Quantize: 7.84ms
      - Zigzag: 2.32ms
      - Huffman: 8.88ms



Project Discussion

- Design Space Exploration
  - Timing back-annotation
    - Automatic approach
      - Use SCE Architecture Refinement
      - Generate Architecture Model
        - » enable “Insert avg. delays”
      - Estimated times are automatically inserted
        - » at granularity of leaf behaviors only!
        - » Problem: no per-block timing during simulation!
      - Example:

```
behavior ChenDCT1(...)
{
  void main(void)
    { int v1, v2, …;

      waitfor 10411200000ull;
      while(1)
      {
        Port1.receive( &in_block);
        if (iter % 2 == 1)
        { ...
```


Project Discussion

- Design Space Exploration
  - Timing back-annotation
    - Automatic approach
      - Use SCE Architecture Refinement
      - Generate Architecture Model
        - » enable “Insert avg. delays”
      - Estimated times are automatically inserted
        - » at granularity of leaf behaviors only!
        - » Problem: no per-block timing during simulation!
      - Example:

Better:
Per-Block Timing!

```
behavior ChenDCT1(...)
{
  void main(void)
    { int v1, v2, …;

      waitfor 10411200000ull;
      while(1)
      { waitfor 10411200000ull / 180;
        Port1.receive( &in_block);
        if (iter % 2 == 1)
        { ...
```

Assignment 6

- Design Space Exploration
  1. Timing back-annotation
     - Manual insertion of estimated computation delays
       - Start from "perfect" specification model
         - » **`jpegencoder.sc`**
       - Add timing statements (**`waitfor`** after **`port.receive()`**)
         - » ChenDCT1: 10411200ns / 180
         - » ChenDCT2: 10411200ns / 180
         - » Quantize: 7839030ns / 180
         - » Zigzag: 2316600ns / 180
         - » Huffman: 8882810ns / 180
       - Save as timed model
         - » **`JpegTimed.sc`**
     - When executed, the resulting model should end at time 10574ms:
       - **`10574: Monitor exits simulation.`**

Assignment 6

- Design Space Exploration
  2. Architecture Exploration
     - Explore various system architectures
       - Use only ARM_7TDMI processors
       - Use only 100MHz core clock frequency
       - Use only 50MHz AMBA AHB bus
       - Vary between 1 and 5 CPUs
       - Vary the mapping of blocks in the DUT to CPUs
     - Note:
       Do not let the architecture refinement tool insert additional timing!
     - Example:
       - Use 3 CPUs, ARM1, ARM2, and ARM3
       - Map DCT1 to ARM1
       - Map DCT2 to ARM2
       - Map Quantize, Zigzag, and Huffman to ARM3
     - Note:
       Without scheduling, any architecture model will end at time 10574ms

Assignment 6

- Design Space Exploration
  3. Scheduling Exploration
     - Explore various scheduling strategies for each selected CPU
     - Choose from
       - Static scheduling
         - » with varying execution order
       - Round-Robin scheduling
       - Priority-based scheduling
         - » with varying priorities
     - Example:
       - 3 ARM CPUs with mapping as above
       - ARM1 statically scheduled
       - ARM2 statically scheduled
       - ARM3 scheduled with Round-Robin
     - When executed, the example model should end at time 19154ms

Assignment 6

- Design Space Exploration
  4. Deliverable
     - Text file "`JPEG_Exploration.txt`" with table:
       - "Best" mapping and scheduling for architecture with 1 CPU
       - "Best" mapping and scheduling for architecture with 2 CPUs
       - "Best" mapping and scheduling for architecture with 3 CPUs
       - "Best" mapping and scheduling for architecture with 4 CPUs
       - "Best" mapping and scheduling for architecture with 5 CPUs
     - For each "best" architecture above, note the overall execution time in the table.
- Due
  - by Friday, Nov 14, 2008, at noon
  - by email to **`doemer@uci.edu`** with subject "EECS222C HW6"

Embedded Software

- Embedded Software
  - Scheduling algorithms
    - Periodic scheduling
      - RMS, EDF
    - Priority inversion
      - Mars Pathfinder Example

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

  - **`Lecture7-es-marw-4b-periodic.ppt`**