IP Core Design

Homework 1

Software Implementation of Target IP

Instructor: Prof. Chein-Wei Jen
Announcement: 2002.10.30

Overview of homework's

Choose an application or an algorithm that is familiar to you, and you will make your choice as a soft IP. Note that you **CAN NOT** choose the algorithms used in JPEG or Motion JPEG.

Homework 1: Verify your choice in C/C++ and port it to ARM core, which can be ARM7TDMI, ARM720T, or ARM922T. In this stage, you run your C/C++ design in ARM instruction set simulator, ARMulator.
Homework 2: Verify your choice in the target environment, ARM Integrator. In this stage, you need to refine your C/C++ design under the consideration of your target environment, e.g., the memory system.
Homework 3: Make your choice as an AMBA AHB-compliant soft IP
Homework 4: Verify your soft IP in target environment.

In homework 1, you can either write your own design or modify an existing reference code. Be aware of the differences between the ARMulator environment and the target platform (i.e., ARM development boards) because software implementation will be ported to ARM development boards in homework 2. Also, the data structures and the partition of functional calls should be carefully defined because portions of your design will be implementation as hardware components in homework 3 and mapped to FPGA in homework 4.

Homework 1

In homework 1, you first get a workable design (write by yourself or get one from existing reference code). Then you optimize this design under the consideration of ARM core's features to get better performance, fewer requirement usages (includes the program itself and the temporal memory for data processing), or even fewer power

consumption. The example approaches may possibly be helpful in such optimization:

- Select or modify the algorithms or the code segments used in JPEG to fit to ARM's architecture. By taking constraints of the ARM core hardware resources into consideration, some algorithms may be more suitable for ARM core than others. An example of such consideration can be found in [1].
- Create SIMD operations. Though current ARM architecture has no specific instructions to support single-instruction, multiple-data (SIMD) operation, certain SIMD operations can be synthesized using a sequence of normal ARM instructions [2].
- Use ARM/Thumb mode for different code segments.

Deliverable

Your deliverable has to include:

1. Report that describes your idea, result, and improvement. You have to clearly point out the differences of code segments between a workable design and an optimized design. Any improvements of the results outside these code segments will not be recognized. You also need to explain and analysis the superiority of the optimized design over the workable one. You also need to prove or convince TA of your claim.

   Summarize your improvement of the memory requirement, profiling, and statistics in table format.

   State your approaches, key ideas and results clearly and formally, and avoid redundant description. Your report can be written in Chinese or English. However, make sure your report is readable. A manual report won't degrade your score, unless it is scrabbled. Remember to attach your reference.
2. Source code of your design and all setting and information required for regenerating the result shown in your report

Scoring (100 points)

Report: 50 points

Improvement:

$$\frac{Cycle\ count\ of\ workable\ design}{Cycle\ count\ of\ optimized\ design} \times \frac{Memory\ requirement\ of\ workable\ design}{Mmeory\ requirement\ of\ optimized\ design} \times 25 \text{ points}$$

The maximum point you can get in this portion is 50.

Plagiarism: -100 points.
Delay: -10 points per day

Important Date

Due : 5:00 p.m. Nov. 14, 2002

For more information

- The contents of this document: Kun-Bin Lee
- ARM development tools: contact the TA with the number = your team number %4

No.		Email	Ext.
0	Kun-Bin Lee	kblee@ twins.ee.nctu.edu.tw	54225
1	Nelson Yen-Chung Chang	ycchang@ twins.ee.nctu.edu.tw	54225
2	Yu-Ming Chang	ymchang@ twins.ee.nctu.edu.tw	54243
3	Hao-yun Chin	ychin@twins.ee.nctu.edu.tw	54225

Reference

[1] Tadashi Sakamoto and Tomohiro Hase, "Software JPEG for a 32-bit MCU with dual issue," IEEE Transactions on Consumer Electronics, Vol. 44 Issue: 4, Nov. 1998, pp. 1334 -1341.

[2] Alan Lewis and Paul Carpenter, "Optimizing digital video codecs in ARM cores," EE Times, Sep. 20, 2001.