CS4271 Homework 2

March 16, 2011

1 Notes

- This assignment is due by 11:59 PM, Wednesday, 20-th April, 2011. No late submissions!
- This is an individual assignment. Acts of plagiarism are subjected to disciplinary action by the university. Please refer to `http://www.comp.nus.edu.sg/students/plagiarism/` for details on plagiarism and its associated penalties.
- *Submission Instructions*: (Failure to follow these instructions may result in deduction of marks) Submit one file named YourMatricNumber.pdf containing your answers to the IVLE Workbin Folder **Lab2**.

2 Question 1

[5 marks]

Suppose you are going to design part of an Digital Signal Processing (DSP) application which involves adaptive pulse code modulation. You want to run the following four periodic tasks on a single processor under Rate Monotonic Scheduling (RMS).

Table 1: Description of DSP tasks

Task name	Description	Period (ms)
adpcm	adaptive pulse code modulator	10.36
fdct	forward discrete cosign transform	3.45
fir	finite impulse response filter	200.56
jpegdct	jpeg discrete cosign transform	12.5

Assume you have three systems S_1, S_2 and S_3 with different choices of processors and L1 cache. The particulars of these three systems are listed in Table 2.

Table 2: Description of available systems

System	Processor frequency (MHz)	Cache parameters
S_1	100	direct mapped, block size = 16 bytes, number of sets = 8
S_2	100	direct mapped, block size = 16 bytes, number of sets = 16
S_3	200	2-way set associative, block size = 16 bytes, number of sets = 16

Assume the cost of these systems are $S_3 > S_2 > S_1$. Use the following configuration for all other parameters: *pipeline:* in-order, *IF queue size:* 2, *RB size:* 2, *Main memory latency:* 50 (in terms of CPU cycles), *branch predictor:* disabled (perfect prediction).

Which of the systems (if any) you would like to use (the cheapest one that makes all the tasks schedulable under RMS)? Write down the *step-by-step* details of how your decision was made.

3 Question 2

[1+2+3 = 6 marks] Analyze the given program q2.c in the folder question2 with the following configuration: *pipeline:* out-of-order, *superscalarity:* 1, *IF queue size:* 4, *RB size:* 8, *cache:* direct-mapped, *Number of cache sets:* 16, *cache block size:* 32, *Main memory latency:* 30 cycles, *branch predictor:* disabled (perfect prediction). Now, answer the following questions:

1. Report all the loop bounds you are required to set and report both the Worst Case Execution Time (WCET) and Simulation Time in CPU cycles. Explain the reason of over-estimation. State your findings clearly in the report (Your reasoning must be explained in the context of this particular program).

2. Without modifying the program, use additional constraints to improve the WCET. Report all additional constraints used along with the reasons of their usages.

3. Without using any additional constraint, modify the source program (q2.c) so that the WCET is reduced. State clearly in your report why your changes reduce the WCET compared to the original version of the program and also include the modified source code in the report (Note that just providing the modified source without any explanation is not going to give you any marks). Your changes must not change the meaning of the program and you will get more points if you can achieve substantial reduction in WCET with minor changes. Does the simulation result change with your transformation ? Justify why or why not.

4 About Chronos

Chronos (`http://www.comp.nus.edu.sg/~rpembed/chronos`) performs timing analysis of embedded software (written in C) through static analysis. In particular, Chronos estimates Worst Case Execution Time (WCET), which is the upper bound on the execution time of a program over all possible data inputs on a specific hardware platform.

To use the pre-installed Chronos on "tembusu"/"Angsana" cluster for the purpose of this assignment, you can follow the steps below:

1. Launch ssh secure client (installed in every machine of SOC). Go to "Edit menu" and go to "settings". Goto "Profile settings->Connection->Tunneling". Check the box "Tunnel X11 connections". Goto "File menu" and click "Save settings".

2. Log in `sunfire.comp.nus.edu.sg` with a SSH Secure Shell Client using your SOC account and password.

3. Log in `tembusu.comp.nus.edu.sg/angsana.comp.nus.edu.sg` from sunfire using the command "ssh -X tembusu"/"ssh -X angsana". To copy file from sunfire to tembusu(angsana), use the "scp" command. Type "man scp" to see its usage guide.

4. Check for a ".bashrc" (note the ".") file in your home directory. If available, edit the file by putting the following two lines at the end of the file:

   ```
   export PATH=$PATH:/home/s/sudiptac/TA-4271/CHRONOS_HOME/bin
   alias chronos="cd /home/s/sudiptac/TA-4271/CHRONOS_HOME/gui; ./gui.sh"
   ```

   If the file is not in your home directory, then create one with that name (i.e. ".bashrc") by putting the above two lines in ".bashrc" file. Save and close the file ".bashrc".

5. Type the command "bash".

6. Open an X11 connection to support the Chronos GUI by using X-Win32 / Xming. "Xming" is already installed in Embedded Systems Teaching Labs. If you want to connect from outside NUS, you can download the free version of Xming at `http://sourceforge.net/projects/xming`. After downloading, you can install and launch the "xming" application with the *default* settings.

7. Download the "benchmarks.zip" from IVLE. Unzip it ("unzip benchmarks.zip") into a folder under your tembusu account. It should contain five sub-folders (*adpcm*, *fdct*, *fir*, *jpegdct*, *question2*) under a folder named "benchmarks".

8. Type the following command in your SSH to launch Chronos GUI: "chronos". Note that all the necessary settings for this copy of Chronos have already been done. Please do not change them. Please also make a note that you have to use *only this installation of Chronos* to do Homework 2.

9. To analyze/simulate a particular benchmark, you should do the following:

   (a) Open a benchmark by choosing to open the folder containing the benchmark. One folder should contain exactly one benchmark (one main method).

   (b) After the source code, CFG, and assembly code are displayed on Chronos GUI, you can set the loop bounds using Option / Loop bound constraints. If no loop bound constraints are required, it means all the loop bounds have already been automatically set by the Chronos data flow analysis.

   (c) Before analyzing/simulating the benchmark, you need to set the processor architecture on which the program runs in Run / Processor configuration.

   (d) After all these have been done, you can estimate the program's WCET or simulate it by clicking Run / Estimate or Run / Simulate respectively. The result will be shown in the bottom panel. *The unit of the displayed result is in terms of CPU cycles.*

10. In your next login to tembusu/angsana, you only need to type "bash" command before launching "chronos" (by typing the command of same name).

11. For sample testing, go to the directory "/home/s/sudiptac/TA-4271/CHRONOS_HOME/benchmarks". Inside this directory, you will find some sample benchmarks. Open any directory in chronos GUI.

12. *NOTE:* Point 1 and point 6 are only relevant if you are using Windows operating systems. Linux users can safely ignore them.

END OF HOMEWORK 2