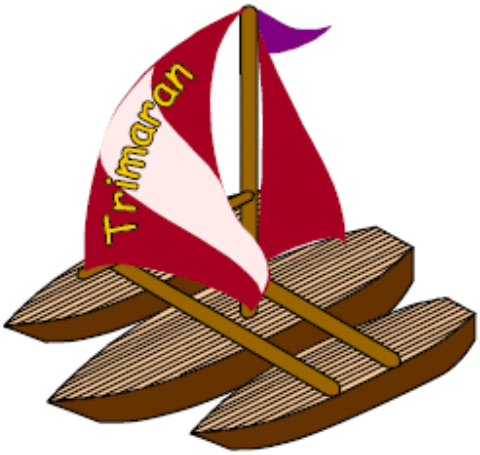
INSTRUCTION LEVEL PARALLELISM



4/13/2009

ILP Project (DUE: Wed. 4/22/2009, 5PM)

Now that you have used the tool, several design tasks have been assigned to you.

Instruction Level Parallelism

Thanks to the TA, Marty Nicholes, for the prior project handout that I leveraged.

BACKGROUND

Suppose your company has been making the standard processor with one functional unit, i.e., integer, floating point, memory, and branch, which can satisfy general-computing requirements for a long time. Now to expand the market share, your company wants to better appeal to the needs of the customers. There are two types of users that you need to consider: general-purpose users who want good performance for their money and scientific users who want loads of computation power. Please write a report to a technical manager who needs to make design decisions with your suggested best mix of functional units for the 2 processors, one of which is targeted to one specific type of users.

INTRODUCTION

The ILP project will be done using the Trimaran toolchain. It has two parts. In Part 1, you will run several benchmarks against various machine configurations to learn how processor architecture affects benchmark performance. In Part 2, you will use the information you gathered from Part1, to design 2 processors.

TOOLCHAIN

- http://www.trimaran.org/
- Trimaran documentation (see the Trimaran and MDES Manuals)

MODIFYING THE MDES FILE

The machine description (MDES) describes the Supported Instruction Sets (ISA), operand formats, and the microarchitectural resources used by the operations in the target processor. Here is a quick primer on modifying the MDES file and running Trimaran with the modified file. You will be making changes to the MDES file to change the number/mix of functional units in the processor. The MDES file is used by the ELCOR backend compiler as it compiles code for the target processor, which is a single cluster for our requirement. The MDES file can be configured to have multiple functional units with the corresponding physical register file types. The ELCOR compiler restricts the number of physical register files per type to one per cluster and takes largely machine-independent assembly code and compiles/optimizes it for a specific machine described in a MDES. Please make sure you are able to see results in the simulator vary based on changes you make to the MDES file before continuing on to Part 1 of the project.

NOTE: Change the number of clusters to 1

1. First logon and run the Trimaran setup script (refer to Warmup).
2. Create a working directory in /tmp, for example, wdir
3. Copy the hpl_pd_elcor_std.hmdes2 ($TRIMARAN_ROOT/elcor/mdes directory) into your working directory.
4. Modify the mdes file as desired (changing number of functional units).

$def !num_clusters 1

5. Compile the mdes file:

   hc hpl_pd_elcor_std.hmdes2

6. Run a benchmark, mm_dyn for example, using the modified mdes file:

   tcc -bench mm_dyn -M/tmp/wdir/hpl_pd_elcor_std.lmdes2

   (note the lmdes2 suffix on the mdes filename. This is the compiled version of the file that was modified)

Compare the statistic results to the one with the original mdes file to see how processor architecture affects benchmark performance, e.g., Dynamic_total_cycles, Dynamic_average_issued_ops/cycle.

BENCHMARKS

1. mm_double: Matrix Multiply (uses both float and int).
2. fft: fast fourier transform (uses float and int).
3. mm_dyn: Matrix Multiply (uses int).
4. mpeg2dec: decode and MPEG2 file (uses int and float).
5. epic: experimental image compression (uses float and in
6. pegwitenc: encryption code (uses int).

PART 1

In this part of the project you will explore the affects of varying functional units. You can vary the number of integer units, float units, memory units, and branch units. As you make these changes, you want to track the dynamic average issued ops per cycle, as well as the dynamic total cycles. Also experiment with the effect of larger or smaller floating point multiply and divide latencies. Perform the listed experiments, as well as some of your own, where you change multiple parameters at a time, such as increasing both floating point units and integer units. Record how the benchmarks behave while varying these parameters. The purpose of Part 1 is to understand the effects of functional unit mix on the benchmarks. Note your findings for inclusion in the project report.

Experiments (run across all benchmarks):

1. Vary the number of integer units
2. Vary the number of floating point units
3. Vary the number of branch units
4. Vary the number of memory units
5. Vary the floating point unit div and mult latency by ± 1

PART 2

Now that you have an understanding of the result of adjusting the functional unit mix, as well as the effect of a floating point unit with slightly shorter or longer latency, you are ready for the design challenge.

You have been asked to design two processors. The first processor is designed for general purpose computing. The second processor is designed for scientific computing. You are to use the information you gathered from Part 1, and work within design constraints to specify the two processors. You will weight the benchmarks importance on the two processor designs as follows:

- General purpose – benchmarks 1-6 weighted evenly

- Scientific – benchmarks 1-3 weighted evenly, 4-6 informational only

The marketing department has stated that the general purpose processor should be optimized for performance/$, while the scientific processor should be optimized for performance. Your report should provide performance and performance/$ metrics.

Design Constraints:

1. You may use up to 12 total functional units (int, float, memory, branch).
2. There is an incremental cost for functional units, as follows:
    a. 4 units are included in the $50 base price
    b. Unit 5 = $10, unit 6 = $20, unit 7 = $40, unit 8 = $70, unit 9 = $110, unit 10 = $160, unit 11 = $220, unit 12 = $290
    c. Floating point units can be chosen as follows (no mixing):
        i. Slow floating point units (adds 1 cycle to mult and div latency) = saves $5 per
        ii. Fast floating point units (subs 1 cycle from mult and div latency) = costs $5 per

SUBMISSION

2 files will be submitted via SmartSite.

- Submit a written report in PDF format.
- Submit a zip file containing the following files:
    - For each processor you designed, submit the hmdes2 file and the Sumstat output for each benchmark.
    - Submit a text file named "README" that describes each of the mdes and stats files.

REPORT GUIDELINES:

- Your report should target a technical manager who needs to make design decisions. You can assume that your manager understands parallel architecture (ILP, superscalar, etc.) but doesn't know the traces you're evaluating, and your report should help the manager choose the best mix of functional units for the 2 processors.
- The report should be no more than 3 pages in PDF format with 1 inch margins and no smaller than 10 point type.
- You are encouraged to include graphs to better illustrate how your experimental results support your conclusions. However, don't feel like you have to submit every single graph. You might use a graph or a few graphs to underscore important points that you are making.
- Organize your report in a way that is easy to follow.
- Do not just write it as a sequence of responses to each section! Put the problem in context, have a conclusion, and so on ... everything you've learned about writing an essay in English class applies here.
- Be sure you have thoroughly read the assignment and include all information it asks for.

DUE DATE: Wednesday 4/22 at 5PM.