Instruction Level Parallelism Laboratory

J. Daniel García Sánchez (coordinator)
Alberto Cascajo
Elías del Pozo
Christian Martínes

Computer Architecture
ARCOS Group
Computer Science and Engineering Department
University Carlos III of Madrid

1 Goal

The main goal for this laboratory is to increase the student's understanding of the concepts related to instruction level paralellism.

2 Lab description

In this lab assignment, the student will complete the evaluation of two small programs on the **EduMIPS64** simulator. EduMIPS64 can be downloaded from http://www.edumips.org/. EduMIPS64 is a simulator written in Java that allows the visualization of program execution written in MIPS assembly for a simplified pipeline.

From the aforementioned webpage the student may obtain:

- The simulator as a JAR file.
- The simulator user manual.
- The source code (available as a GitHub repository).

The first code to be evaluated is the following:

```
.data
v: .space 400
w: .space 400

.text
addi R1, R0, 400
addi R2, R1, -8
addi R3, R0, 0
addi R5, R0, -1

loop:
sw R3, v(R3)
addi R4, R3, 1
add R4, R4, R5
sw R4, w(R2)
```

```
addi R3, R3, 8
addi R2, R2, −8
bne R3, R1, loop
 syscall 0
```

Listing 1: Exercise 1

This program makes use of two vectors (v y w) each of them being capable of storing 100 double precission values.

The second code to be evaluated is the following:

```
 .data
v:  .space 256
w:  .space 256
x:  .double 2.0
y:  .double 1.5
z:  .double 0.0
.text
daddi R1, R0, v
daddi R2, R0, w
daddi R3, R0, 256

ldc1 F2, x(R0)
ldc1 F4, y(R0)
daddi R4, R0, 0

loop1:
dmtc1 R4, F6
cvt.d.l F6, F6
mul.d F8, F2, F6
mul.d F8, F8, F8
sdc1 F8, 0(R1)
mul.d F10, F4, F6
sdc1 F10, 0(R2)
daddi R1, R1, 8
daddi R2, R2, 8
daddi R4, R4, 8
bne R4, R3, loop1

daddi R1, R0, v
daddi R2, R0, w
daddi R4, R0, 0
ldc1 F8, z(R0)

loop2:
ldc1 F2, 0(R1)
ldc1 F4, 0(R2)
mul.d F6, F2, F4
add.d F8, F8, F6
daddi R1, R1, 8
daddi R2, R2, 8
daddi R4, R4, 8
bne R4, R3, loop2
sdc1 F8, z(R0)
 syscall 0
```

Listing 2: Exercise 2

NOTE: Please be careful when copying and pasting from the PDF. Some characters may change. Review your code before executing it.

3 Tasks

3.1 Task 1: Evaluation of example 1

Simulate the execution of example 1 without *forwarding*. Identify the stalls and the cause for each of them.

Enable the *forwarding* option and simulate again the same example. Determine the obtained *speedup* due to the use of *forwarding*.

3.2 Task 2: Evaluation of example 2

Simulate the execution of example 2 without *forwarding* and with *forwarding*. Identify the stalls and the cause for each of them.

Compute the obtained *speedup* in the example due to the use of *forwarding*.

3.3 Task 3: Loop unrolling, first loop

NOTE: For all the evaluations of this section keep the *forwarding* option enabled.

For the case of the first loop (*loop1*) of example 2, unroll the loop with factors of 2 and 4, but without unrolling the second loop.

Compute the obtained *speedup* in both cases with respect to the original code.

IMPORTANT: In addition to performing loop unrolling you must schedule instructions in order to improve the best possible speedup.

3.4 Task 4: Loop unrolling, second loop

NOTE: For all the evaluations of ths section keep the *forwarding* option enabled.

For the case of the second loop (*loop2*) of example 2, unroll the loop with factors of 2 and 4, but without unrolling the first loop.

Compute the obtained *speedup* in both cases with respect to the original code.

IMPORTANT: In addition to performing loop unrolling you must schedule instructions in order to improve the best possible speedup.

3.5 Task 5: Loop unrolling

NOTE: For all the evaluations of this section keep the *forwarding* option enabled.

Combine both unrolling options from tasks 3 and 4.

Compute the obtained *speedup* in both cases with respect to the original code.

IMPORTANT: In addition to performing loop unrolling you must schedule instructions in order to improve the best possible speedup.

4 Submission

The deadline for the submission of the results from this lab is at 24 hourse after the finish time of the lab session for group where the student is enrolled.

The following rules must be followed:

- All assignments shall be submitted through Aula Global.
- The only accepted format for the submission shall be the quiz available through Aula Global.

- The submission of the quizes shall be individual even if the exercise was carried out in group. Please, keep in mind that different students may get different questions.
- Once the quiz is initiated, the student will have a maximum of 10 minutes to complete it.
- Each student will have one attempt to complete the quiz.
- The maximum number of questions in the quiz for one student will be 10.
- Students are recommeded to prepare solutions to all exercises in advance (source code), as well as associated data (number of stalls, cycles, . . .). That information may be needed during the quiz.