CS M151B Homework 6

Charles Zhang

February 17, 2022

Problem 4.31

In this exercise we compare the performance of 1-issue and 2-issue processors, taking into account program transformations that can be made to optimize for 2-issue execution. Problems in this exercise refer to the following loop (written in C):

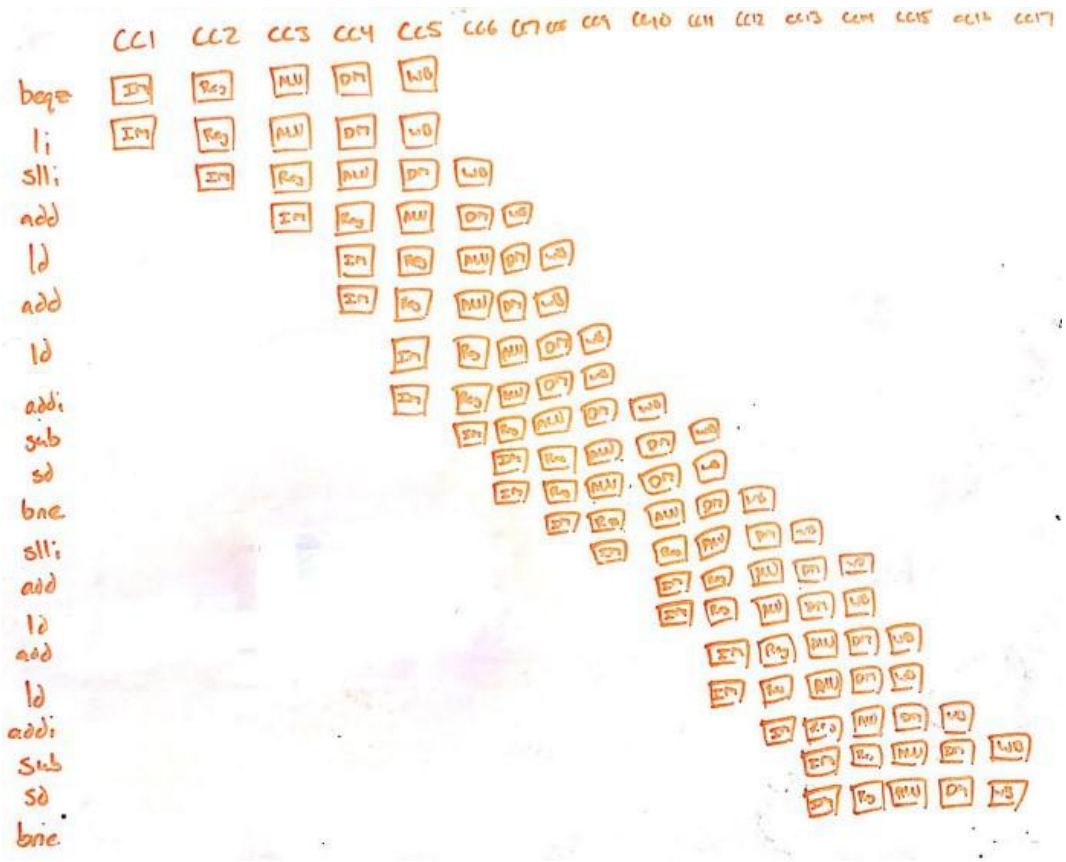
```
for (i = 0; i != j; i += 2)
    b[i] = a[i] - a[i + 1];
```

Assume the two-issue, statically scheduled processor for this exercise has the following properties:

- One instruction must be a memory operation; the other must be an arithmetic/logic instruction or a branch.
- The processor has all possible forwarding paths between stages (including paths to the ID stage for branch resolution).
- The processor has perfect branch prediction.
- Two instructions may not issue together in a packet if one depends on the other.
- If a stall is necessary, both instructions in the issue packet must stall.

a) Draw a pipeline diagram showing how MIPS code given below executes on the two-issue processor. Assume that the loop exits after two iterations.

```
        beqz    x13, DONE
        li      x12, 0
TOP:
        slli    x5, x12, 3
        add     x6, x10, x5
        ld      x7, 0(x6)
        add     x31, x11, x5
        ld      x29, 8(x6)
        addi    x12, x12, 2
        sub     x30, x7, x29
        sd      x30, 0(x31)
        bne     x12, x13, TOP
DONE:
```



b) What is the speedup of going from a one-issue processor to a two-issue processor when running the optimized code from part a) and the optimized code below?

```
    beqz    x13, DONE
    li      x12, 0
    jal     ENT
TOP:
    slli    x5, x12, 3
    add     x6, x10, x5
    ld      x7, 0(x6)
    ld      x29, 8(x6)
    addi    x12, x12, 2
    sub     x30, x7, x29
    add     x31, x11, x5
    sd      x30, 0(x31)
ENT:
    bne     x12, x13, TOP
DONE:
```

Since we have 20 instructions in the code from part a) and there are no load-use hazards, we know that the final instruction will begin at CC20. This tells us that the code takes 24 cycles to finish execution in a 1-issue processor. As seen in part a), the 2-issue processor only takes 17 clock cycles to finish execution. Therefore, we have a speedup of:

$$24 \text{ cycles} - 17 \text{ cycles} = \boxed{7 \text{ cycles}}$$

This code block would execute with 21 instructions. Again, there are no load-use hazards, so the final instruction would begin at CC21. As a result, this code takes 25 cycles to finish execution in a 1-issue processor. In the 2-issue processor specified in the problem, we can save 5 instructions by packaging them together. Therefore, the last instruction would begin at CC16, finishing at CC20. Therefore, the speedup for this code is:

$$25 \text{ cycles} - 20 \text{ cycles} = \boxed{5 \text{ cycles}}$$

c) Unroll the MIPS code from part b) so that each iteration of the unrolled loop handles two iterations of the original loop. Then, rearrange/rewrite your unrolled code to achieve better performance on the 1-issue processor. You may assume that j is a multiple of 4.

```
    beqz    x13, DONE
    li      x12, 0
    jal     ENT
TOP:
    slli    x5, x12, 3
    add     x6, x10, x5
    ld      x7, 0(x6)
    ld      x29, 8(x6)
    addi    x12, x12, 2
    sub     x30, x7, x29
    add     x31, x11, x5
    sd      x30, 0(x31)
    slli    x5, x12, 3
    add     x6, x10, x5
    ld      x7, 0(x6)
    ld      x29, 8(x6)
    addi    x12, x12, 2
    sub     x30, x7, x29
    add     x31, x11, x5
    sd      x30, 0(x31)
ENT:
    bne     x12, x13, TOP
DONE:
```