

## 4DM4 - Computer Architecture

### Assignment #2, 2013

#### Advanced Static Pipelining

Out: Thursday Oct. 10, 2013

Due: Wednesday Oct. 23, at start of tutorial

#### **Q1: Gaussian Elimination loop, with data hazard stalls, on a 5-stage pipeline.**

A.12 [20/22/22] <A.4, A.6> In this exercise, we will look at how a common vector loop runs on statically and dynamically scheduled versions of the MIPS pipeline. The loop is the so-called DAXPY loop (discussed extensively in Appendix G) and the central operation in Gaussian elimination. The loop implements the vector operation  $Y = a \times X + Y$  for a vector of length 100. Here is the MIPS code for the loop:

```
foo:  L.D      F2,0(R1)      ;load X(i)
      MULT.D  F4,F2,F0      ;multiply a*X(i)
      L.D      F6,0(R2)      ;load Y(i)
      ADD.D    F6,F4,F6      ;add a*X(i) + Y(i)
      S.D      0(R2),F6      ;store Y(i)
      DADDUI   R1,R1,#8      ;increment X index
      DADDUI   R2,R2,#8      ;increment Y index
      DSGTUI   R3,R1,done    ;test if done
      BEQZ     R3,foo        ;loop if not done
```

(a) Assume a 5-stage pipeline as follows:

#### **Assumptions:**

- (1) Integer EX unit is 1 stage, with full data forwarding.
- (2) FP-ADD pipeline is 2 stages deep.
- (3) FP-MULT pipeline is 5-stages deep.
- (4) Mem is a 2 stage pipeline; WB is one-stage.
- (5) Instructions that don't need MEM can go directly to WB
- (6) Mem and WB stages can accept  $\leq 2$  instructions per clock cycle. The data-path has been widened by us (the designers) to allow this.

Create a Space-Time diagram for one iteration of the loop. Label all data forwardings with an arrow, from the producing stage to the consuming stage of the data operand. Label all stalls. Compute the clock cycles required per iteration of the loop.

Compute the MFLOP rating, assuming a 3 GHz clock.

(NOTE: L.D is a double precision Load-Word; S.D is a double precision Store-Word, DADDUI is an integer ADD, DSGTUI is the integer Set-Greater-Than, and BEQZ is the Branch-Equal-Zero instruction.)

(b) Repeat part (a), but schedule the code to reduce stalls (but do not unroll the loop).

© Repeat part (a), but unroll the loop to allow 5 iterations to occur in one new iteration, but do not schedule the code.

(d) repeat ©, but this time schedule the code.

Note: It is useful to use an excell spreadsheet to create your space-time diagram, so that you can rearrange the code easily. I will post one that you can edit on the class web-site.