

4DM4 Sample Problems for Midterm -2013

Tuesday, Oct. 22, 2013

Hello Class

For the 4DM4 midterm on Monday, Oct. 28, we won't cover the lab. material, i.e., VHDL or the switch or the Network-on-Chips, etc. That material may be on the final exam, but it won't be on the midterm.

For the midterm, we will cover all material on assignment #1 and assignment #2, i.e., bubble sorting in an exascale style of machine, static scheduling in linear pipelines (including the Itanium). We won't cover Dynamic-Scheduling, since that will be covered in assignment #3 (after the mid-term.) Here are some sample problems for the midterm, on statically-scheduled linear pipelines. A space-time diagram is attached at the end of this document.

We do not have solutions for these sample problems. Please see our assignments and their solutions for how to solve a problem.

- Problem 4.8, reference text: Consider the loop $Y[i] = a * X[i] + Y[i]$, a key step in Gaussian elimination. (This problem is similar to assignment #2).

Loop:	LD	F0,0(R1)
	MULD	F0,F0,F2
	LD	F4,0(R2)
	ADDD	F0,F0,F4
	SD	0(R2),F0
	SUBI	R1,R1,#8
	SUBI	R2,R2,#8
	BNEZ	R1,loop

- (1) Assume an 6-stage single issue static-scheduled pipelined machine (F, D, EX, M1, M2, WB).

Use these assumptions:

- The MULTD unit is fully pipelined with 6 stages.
- The ADDD unit is fully pipelined with 3 stages.
- full forwarding (with 0 cc delay) for both INT and FP forwarding
- a 1 cc delayed branch.

(a) Without unrolling the loop, complete a timing diagram.

(b) Assuming a 2 GHz clock rate, determine the floating point performance of the loop. Is your answer exact or an approximation, and explain why?

(c) In a loop with 1000 iterations, when does the 100-th MULTD operation start execution, and end execution?

(d) Unroll the loop 2 times and schedule it for maximum performance. Complete the timing diagram.

(e) Unroll the loop as many times as necessary to schedule it without any stalls to achieve maximum performance.

(f) Assuming a 2 GHz clock rate, determine the floating point performance of the loop. Is your answer exact or an approximation, and explain why?

(g) In a loop with 1000 iterations originally, after unrolling and scheduling in part(d), when does the 100-th MULTD operation start execution, and end execution?

- (2) Using the loop above, assume a 6-stage pipeline as explained above, with **DUAL-ISSUE**. Use these assumptions:
 - The MULTD unit is fully pipelined with 6 stages.
 - The ADDD unit is fully pipelined with 3 stages.
 - 1 cc delay for both integer and FP forwarding (forwarding takes 1 cc)
 - a 1cc delayed branch.
 - 1st instruction column is for all Load, Store, INT and branch operations
 - 2nd instruction column is for all FP operations
 - (a) Unroll the loop 2 times and schedule it for maximum performance.
 - (b) Unroll the loop as many times a necessary to schedule it without any stalls to achieve maximum performance.
 - (c) Assuming a 2 GHz clock rate, determine the floating point performance of the loop. Is your answer exact or an approximation, and explain why?
 - (d) In a loop with 1000 iterations originally, after unrolling and scheduling in part(a), when does the 100-th MULTD operation start execution, and end execution?

- Using the loop above, assume a 8-stage MIPS single issue pipeline, as discussed in class. The stages are (F1, F2, D, EX, M1, M2, M3, WB). Use these assumptions:
 - The MULTD unit is fully pipelined with 5 stages.
 - The ADDD unit is fully pipelined with 2 stages.
 - fast (0 cc) delay for both integer and FP forwarding
 - a delayed branch.
 - (a) Unroll the loop 4 times and schedule it for maximum performance.
 - (b) Unroll the loop as many times a necessary to schedule it without any stalls for maximum performance.
 - (c) Assuming a 2 GHz clock rate, determine the floating point performance of the loop. Is your answer exact or an approximation, and explain why?
 - (d) In a loop with 1000 iterations originally, after unrolling and scheduling in part(a), when does the 100-th MULTD operation start execution, and end execution?

- Assuming the loop above, assume a 8-stage MIPS dual issue pipeline, as discussed in class. The stages are (F1, F2, D, EX, M1, M2, M3, WB). Use these assumptions:
 - The MULTD unit is fully pipelined with 5 stages.
 - The ADDD unit is fully pipelined with 2 stages.
 - 1 cc delay for both integer and FP forwarding
 - a delayed branch.
 - (a) Unroll the loop 4 times and schedule it for maximum performance.
 - (b) Unroll the loop as many times a necessary to schedule it without any stalls for maximum performance.
 - (c) Assuming a 2 GHz clock rate, determine the floating point performance of the loop. Is your answer exact or an approximation, and explain why?
 - (d) In a loop with 1000 iterations originally, after unrolling and scheduling in part(a), when does the 100-th MULTD operation start execution, and end execution?

- Given this C-code, write the unoptimized assembler instructions for it, and then schedule it on a 5-stage linear pipeline:
 - vector X is double precision floating point
 - vector X is in memory starting at hexadecimal address 0001,0000
 - vector Y is in memory starting at hexadecimal address 0002,0000
 - scalar c is in register F31

```
for (i=0; i<= 100; i++) {
    Y(i) = 3*X(i)+ c }
```

- (a) Write the assembler instructions for this loop, using the class instruction syntax. Assume no optimization at all. Assume all loop control instructions are at the end of the loop. You may need instructions such as these:

SETgt	R1, R2, #10	set R1 = 1 if R2 > immediate constant 10 (decimal)
SETge	R1, R2, #10	set R1 = 1 if R2 >= immediate constant 10 (decimal)
SETle	R1, R2, #10	set R1 = 1 if R2 <= immediate constant 10 (decimal)
BNEZ	R1, loop	

% here is a sketch of the answer : you must double-check it

assume constant 3 is in F31, constant c is in F30

assume vector X starts at memory address 0001,0000 hex, and vector Y starts after vector X

assume R1 is an address for vector X[1] initially,

assume R2 is an address for vector Y[2] initially,

assume R3 is a loop counter initially == 101

```
loop  LD      F1,0(R1)
      MULTD   F1,F1,F31    % mult by 3
      ADDD    F1,F1, F30    % add c
      SD      F1,0(R1)
      ADDI    R1,R1,#8
      ADDI    R2,R2,#8
      SUBI    R3,R3,#1
      BNEZ    R3,loop
```

Extra assumptions:

- Assume the basic 5 stage pipeline for INT instructions.
- The MULTD unit is fully pipelined with 6 stages.
- The ADDD unit is fully pipelined with 3 stages.
- full forwarding (with 0 cc delay) for both INT and FP forwarding
- a 1 cc delayed branch.

(b) Without scheduling, complete a timing diagram for this loop

(c) Unroll the loop 4 times and schedule it for maximum performance.

(d) Assuming a 2 GHz clock rate, determine the floating point performance of the loop. Is your answer exact or an approximation, and explain why?

(e) In a loop with 1000 iterations originally, after unrolling and scheduling in part(a), when does the 100-th MULTD operation start execution, and end execution?

[illegible]

New in 2013, based on Assignment #1

(a) Write the assembly code to implement a stage of the bubble-sort computation in Assignment #1, similar to the loops we have examined in class. Assume the BEZ.D, BGEZ.D instructions can examine double-precision floating point numbers and branch.

(b) Unroll the loop 4 times, and schedule it for maximum performance. Use these assumptions:

- a static -scheduled 2-issue machine, similar to the dual-issue 5-stage pipelines examined in the class notes
- 2 Mem units, each a 2 stage pipeline
- 2 FP Compare units, each a 2 stage pipeline
- instructions which do not need the MEM unit can bypass the Mem and go straight to the WB
- full forwarding, each result can be forwarded at the end of the clock cycle in which it is produced

(c) Complete a timing diagram for the 2-issue machine.

(d) What is the peak performance of the machine, assumption a 3.25 MHz clock rate?

(e) What is the peak performance of this loop, with infinite unrolling (if you had enough FP registers) ?