McGill ECE ECSE 425: Computer Organization and Architecture
Fall 2011

Assignment 4
Due November 14, 2011

You may type or write your answers by hand. If you write by hand, make sure it is clearly presented. **Do not use pencils but ink**. Please put your name and student ID clearly on the submitted assignment.

You may make use of reasonable assumptions of your own for data that might be missing in the problem texts, provided that they are explicitly and clearly stated.

You may submit a partial answer to a problem. The grading will account for this.

Question 0, Feedback (1 pt extra credit)

How many hours did you spend working on this homework assignment?

Question 1 (3 pts) Superscalar Execution

Consider a simple in-order, multiple-issue processor. It has two execution pipelines, each capable of beginning execution of one instruction per cycle. Assume that the there are always instructions ready to be issued when needed. Further assume that results can be immediately forwarded from one execution unit to another, or to itself, and that the only reason an execution pipeline would stall is to observe a true data dependency. How many cycles does the loop in Figure 1(a) require (the latencies for each instruction can be found in Figure 1(b))?

```
Loop:  LD      F2,0(Rx)
I2:    DIVD    F8,F2,F0
I3:    MULTD   F2,F6,F2
I4:    LD      F4,0(Ry)
I5:    ADDD    F4,F0,F4
I6:    ADDD    F10,F8,F2
I7:    ADDI    Rx,Rx,#8
I8:    ADDI    Ry,Ry,#8
I9:    SD      F4,0(Ry)
I10:   SUB     R20,R4,Rx
I11:   BNZ     R20,Loop
```

(a)

Memory LD	+4
Memory SD	+1
Integer ADD, SUB	+0
Branches	+1
ADDD	+1
MULTD	+5
DIV	+12

(b)

Figure 1: Sample code and functional unit latencies for Questions 1 and 2.

Question 2 (3 pts) Out-of-order Superscalar Execution

Consider the loop in Figure 1(a) once more, but executing on a dynamically scheduled processor with two execution pipelines. Assuming that the instructions for a single iteration are available for re-ordering in the instruction queue, how many cycles does the loop require?

Question 3 (6 pts) Hardware Speculation

Consider the following loop, which computes the dot-product of two vectors.

```
Loop:    L.D       F2, 0(R1)
I2:      L.D       F4, 0(R2)
I3:      MULT.D    F2, F2, F4
I4:      ADD.D     F0, F0, F2
I5:      DADDUI    R1, R1, #8
I6:      DADDUI    R2, R2, #8
I7:      SUBUI     R4, R3, R1
I8:      BNEZ      R4, Loop
```

Figure 2: Sample code for Question 3.

Consider a processor that supports hardware speculation for one branch. Assume that one instruction can be issued to a reservation station each cycle, and that the instruction queue is always ready. Assume the follow distribution of functional units (FU, which are not pipelined), execution latencies, and reservation stations (RS):

FU Type	Cycles to Execute	No. of FU	No. of RS
Integer ALU	1	1	5
FP Adder	2	1	3
FP Multiplier	5	1	2
Load/Store	2	1	5

The first cycle of load/store execution performs effective address calculation; the second cycle is used to access memory. Assume that memory accesses always hit in cache and that branches are always predicted correctly. The re-order buffer has 56 entries, and one instruction can be committed each cycle.

(a) In what cycle does the first load of the second iteration (I1) begin execution? If speculation were disabled, when would this instruction begin execution?

(b) What is the state of the reservation stations, re-order buffer, and register file when the first load of the second iteration (I1) begins execution? Use the following tables as examples for how to format your answer:

Re-order Buffer					
No.	Busy	Instruction	State	Dest	Value
1					
2					
…					

Register Status

	R1	R2	R3	R4	F0	F2	F4
Re-order No.							
Busy							

Reservation Stations

Name	Busy	Op	V_j	V_k	Q_j	Q_k	Dest	Address
Int ALU 1								
Int ALU 2								
Int ALU 3								
Int ALU 4								
Int ALU 5								
FP Add 1								
FP Add 2								
FP Add 3								
FP Mult 1								
FP Mult 2								
Load 1								
Load 2								
Load 3								
Load 4								
Load 5								

Question 4 (8 pts) Memory Hierarchy

Consider a system with an in-order processor that runs at 1.1 GHz and has a CPI of 0.7, excluding memory accesses. The only instructions that read or write data from memory are loads (20% of all instructions) and stores (5% of all instructions).

The memory system for this computer is composed of a split L1 cache; the L1 caches have a one cycle hit time. Both the I-cache and D-cache are direct-mapped and hold 32 KB each. The I-cache has a 2% miss rate and 32-byte blocks; the D-cache has a miss rate of 5% and 16-byte blocks, and employs a write through policy. The D-cache uses a write buffer that eliminates stalls for 95% of all writes.

The 512 KB write-back, unified L2 cache has 64-byte blocks and an access time of 15 ns. It is connect to the L1 cache by a 128-bit data bus that runs at 266 MHz and can transfer one 128-bit word per cycle. Of all memory references sent to the L2 cache in this system, 80% are satisfied without going to main memory. Also, 50% of all blocks replaced are dirty.

The 128-bit wide main memory has an access latency of 60 ns, after which any number of bus words may be transferred at the rate of one per cycle on the 128-bit wide 133 MHz main memory bus.

a. What is the average memory access time for instruction accesses?
b. What is the average memory access time for data reads?
c. What is the average memory access time for data writes?
d. What is the overall CPI, including memory accesses?
e. How much faster would the system run with a 2.1 GHz processor? Assume that the L1 caches still hit in one cycle, and that the speed of the L2 cache, main memory, and buses remains the same in absolute terms (e.g., the L2 cache still has a 15 ns access time and a 266 MHz bus connecting it to the CPU and L1 cache).
f. Which one part of the memory system should be improved to make the system faster? Consider the L2 cache speed, bus speeds, main memory speed, and L1 and L2 hit rates. In each case, graph the change in overall system performance, holding all parameters fixed but the one to be improved.