HW2 Assignment

Problem 1: Solve P5.21, P5.22, P5.23 in Chapter 5 of the Shen/Lipasti textbook.

1. **P5.21** Simulate the execution of the following code snippet using Tomasulo's algorithm. Show the contents of the reservation station entries, register file busy, tag (the tag is the RS ID number), and data fields for each cycle (make a copy of the table below for each cycle that you simulate). Indicate which instruction is executing in each functional unit in each cycle. Also indicate any result forwarding across a common data bus by circling the producer and consumer and connecting them with an arrow.

   i: R4 ← R0 + R8
   j: R2 ← R0 * R4
   k: R4 ← R4 + R8
   l: R8 ← R4 * R2

   Assume dual dispatch and dual CDB (common data bus). Add latency is two cycles, and multiply latency is 3 cycles. An instruction can begin execution in the same cycle that it is dispatched, assuming all dependencies are satisfied.

CYCLE #: _____

ID	Tag	Sink	Tag	Source
1				
2				
3				

Adder

ID	Tag	Sink	Tag	Source
4				
5				

Mult/Div

	Busy	Tag	Data
0			6.0
2			3.5
4			10.0
8			7.8

DISPATCHED INSTRUCTION(S): _____

2. **P5.22:** Determine whether or not the code executes at the dataflow limit for Problem 1. Explain why or why not. Show your work.
3. **P5.23:** As presented in this chapter, load bypassing is a technique for enhancing memory data flow. With load bypassing, load instructions are allowed to jump ahead of earlier store instructions. Once address generation is done, a store instruction can be completed architecturally and can then enter the store buffer to await available bus cycle for writing to memory. Trailing loads are allowed to bypass these stores in the store buffer if there is no address aliasing.

   In this problem you are to simulate such load bypassing (there is no load forwarding). You are given a sequence of load/store instructions and their addresses (symbolic). The number to the left of each instruction indicates the cycle in which that instruction is dispatched to the reservation station; it can begin execution in that same cycle. Each store instruction will have an additional number to its right, indicating the cycle in which it is ready to retire, i.e., exit the store buffer and write to the memory.

Assumptions:

•All operands needed for address calculation are available at dispatch.

•One load and one store can have their addresses calculated per cycle.

•One load OR store can be executed, i.e., allowed to access the cache, per cycle.

•The reservation station entry is deallocated the cycle after address calculation and issue.

•The store buffer entry is deallocated when the cache is accessed.

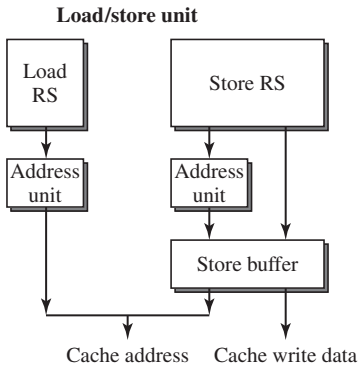
•A store instruction can access the cache the cycle after it is ready to retire.

•Instructions are issued in order from the reservation stations.

•Assume 100% cache hits.

Example:

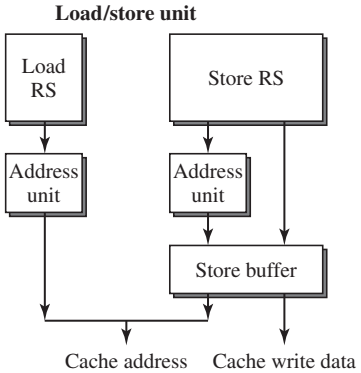
Dispatch cycle	Instruction	Retire cycle
1	Load A	
1	Load B	
1	Store C	5



Cycle	Load Reservation Station				Store Reservation Station				Store Buffer				Cache Address	Cache Write data
1	Ld A	Ld B			St C									
2	Ld B								St C				Ld A	
3									St C				Ld B	
4									St C					
5									St C					
6													St C	data

Code:

Dispatch cycle	Instruction	Retire cycle
1	Store A	6
2	Load B	
3	Load A	
4	Store D	10
4	Load E	
4	Load A	
5	Load D	



Cycle	Load Reservation Station				Store Reservation Station				Store Buffer				Cache Address	Cache Write data
1														
2														
3														
4														
5														
6														
7														
8														
9														
10														
11														
12														
13														
14														
15														

Problem 2: Solve P6.3, P6.11 from Chapter 6 of the Shen/Lipasti textbook.

P6.3: Given the dispatch and retirement bandwidth specified, how many integer ARF (architected register file) read and write ports are needed to sustain peak throughput? Given instruction mixes in Table 1-1, also compute average ports needed for each benchmark. Explain why you would not just build for the average case. Given the actual number of read and write ports specified, how likely is it that dispatch will be port-limited? How likely is it that retirement will be port-limited?

P6.11: The IBM POWER3 can detect up to four regular access streams and issue prefetches for future references. Construct an address reference trace that will utilize all four streams.

Problem 3: Solve P7.10, P7.11, and P7.12 from Chapter 6 of the Shen/Lipasti textbook.

4. **P7.10:** If the P6 microarchitecture had to support an instruction set that included predication, what effect would that have on the register renaming process?
5. **P7.11**: As described in the text, the P6 microarchitecture splits store operations into a STA and STD pair for handling address generation and data movement. Explain why this makes sense from a microarchitectural implementation perspective.
6. **P7.12:** Following up on Problem 5, would there be a performance benefit (measured in instructions per cycle) if stores were not split? Explain why or why not?

TABLE 1-1 **TABLE 6-2:** Dynamic instruction mix of the benchmark set.

Instruction Mix		Integer Benchmarks (SPECint 92)				Floating-Point Benchmarks (SPECfp 92)		
		compress	*eqntott*	*espresso*	*li*	*alvinn*	*hydro2d*	*tomcatv*
Integer	**Arithmetic (single cycle)**	42.73%	48.79%	48.30%	29.54%	37.50%	26.25%	19.93%
	Arithmetic (multi-cycle)	0.89%	1.26%	1.25%	5.14%	0.29%	1.19%	0.05%
	Load	25.39%	23.21%	24.34%	28.48%	0.25%	0.46%	0.31%
	Store	16.49%	6.26%	8.29%	18.60%	0.20%	0.19%	0.29%
Floating-Point	**Arithmetic (pipelined)**	0.00%	0.00%	0.00%	0.00%	12.27%	26.99%	37.82%
	Arithmetic (non-pipelined)	0.00%	0.00%	0.00%	0.00%	0.08%	1.87%	0.70%
	Load	0.00%	0.00%	0.00%	0.01%	26.85%	22.53%	27.84%
	Store	0.00%	0.00%	0.00%	0.01%	12.02%	7.74%	9.09%
Branch	**Unconditional**	1.90%	1.87%	1.52%	3.26%	0.15%	0.10%	0.01%
	Conditional	12.15%	17.43%	15.26%	12.01%	10.37%	12.50%	3.92%
	Conditional to Count Register	0.00%	0.44%	0.10%	0.39%	0.00%	0.16%	0.05%
	Conditional to Link Register	4.44%	0.74%	0.94%	2.55%	0.03%	0.01%	0.00%