Computer Architecture ELE 475
Problem Set #5

This problem set is **ungraded and not collected**. We will release solutions two weeks after the problem set is released. Please stop by office hours if you have questions.

Problem #1 (10 Points): Page 254 in H&P5, Problem 3.13

For problem #1: Assume that 3 stall cycles after load means load has a latency of 4.

Problem #2 (10 Points): Assume that your architecture has a test-and-set instruction as its only atomic primitive. Implement atomic compare-and-exchange out of the test-and-set primitive.

Problem #3 (10 Points): List the possible sequentially consistent outcomes for the variables i and j after the completion of executing the three threads T1, T2, and T3. Assume that all threads begin executing after 'i' has been set to 9 and 'j' is set to 10.

```
T1:
ADDI R1, R0, 30
SW R1, 0(i)
LW R2, 0(j)
SW R2, 0(j)
```

```
T2:
ADDI R5, R0, 99
LW R6, 0(j)
ADD R7, R5, R6
SW R7, 0(j)
```

```
T3:
ADD R8, R0, 100
SW R8, 0(i)
```

Problem # 4 (10 Points): You are writing a multi-threaded program that will count the number of occurrences of a value in an array. The values in the array are between 0 and 1023. In effect, you will be building a histogram. Assume that the list of numbers is very large, on the order of gigabytes large. Extend the following program such that 100 threads (processors) can execute on the program concurrently. Assume a sequentially consistent memory model. Add P() and V() semaphores where appropriate and add any storage needed for the semaphores. Explain why the speedup of such a solution may not be 100x. Note that the output lock array is assumed to be initialized to 1 (this allows for a mutex).

```
// Sequential code, assume that the input and output arrays are created
// outside of the function
#define MAX_VALUE 1023
function(int input_array_size, int * input_array, int * output_array)
{
   int counter;
   for(counter = 0; counter < input_array_size; counter++)
   {
      assert(input_array[counter] <= MAX_VALUE);
      assert(input_array[counter] >= 0);
      output_array[input_array[counter]]++;
   }
}
```

Problem # 5 (10 Points): Show for each cache line and cache what state it is in on every cycle assuming three processors executing code as interleaved below. Assume a 64-byte cache line block size. Assume all cores contain a direct mapped cache that is 4KB large. First, assume that the processors are using a snoopy MSI cache coherence protocol. Second, repeat this for a MESI protocol.

Time	P1:	P2:	P3:
1	LW R1, 4(R0)		
2		LW R1, 16(R0)	
3			LW R1, 4(R0)
4			SW R2, 100(R0)
5			LW R4, 104(R0)
6		LW R3, 100(R0)	
7	SW R1, 0(R0)		
8	LW R1, 4100(R0)		
9	SW R2, 4100(R0)		
10		SW R3, 4100(R0)	
11			SW R5, 0(R0)

Problem #6 (10 Points): Calculate the bisection bandwidth for a 4-ary 3-cube without end-around, but where each link is 32-bits wide and clocks at 800MHz. Calculate the bisection bandwidth of an 8-node omega network with 64-bit links that clock at 1.2GHz.

Problem #7 (10 Points): How large of a credit counter is needed to provide full bandwidth on a link where the link has one cycle for routing delay, two cycles for link delay, and the return credit takes two cycles? What is the bandwidth as a proportion of the maximum if the credit size is two smaller than the needed number?

Problem #8 (10 Points): Assume that a message is routed on a 2D dimension-ordered network that is 4 by 4. Assume that the link delay is one cycle and that the router delay in each hop is two cycles. Assume that each link is one byte wide. Assume that the flit length is 4 bytes and the phit size is one byte. How many cycles does it take to send a 32-byte message from location (0,0) to location (2,3) assuming no insertion or destination delay assuming that the architecture implements store-and-forward? Repeat assuming that the network is a wormhole/cut-through switched network.

Problem Set continued on next page

Problem #9 (10 Points): Show for each cache line, cache, and directory controller what state it is on every load/store. Assume that the code is executing on three processors as interleaved below. Assume that there is one centralized directory. Also, draw the share list that exists in the directory. Assume a 64-byte cache line block size. Assume all cores contain a direct mapped cache that is 4KB large. Assume that a MSI protocol is used in the caches and a ESU protocol is used at the directory.

Time	P1:	P2:	P3:
1	LW R1, 4(R0)		
2		LW R1, 16(R0)	
3			LW R1, 4(R0)
4			SW R2, 100(R0)
5			LW R4, 104(R0)
6		LW R3, 100(R0)	
7	SW R1, 0(R0)		
8	LW R1, 4100(R0)		
9	SW R2, 4100(R0)		
10		SW R3, 4100(R0)	
11			SW R5, 0(R0)

Problem # 10 (10 Points): Page 420 in H&P5, Problem 5.11