EE660: Computer Architecture **Spring, 2017**

Assignment No. 3 (Due: May 12, 2017)

Score:_____/100

Student: Hualiang Li *Date: April 25, 2017*

31 (10/100). 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.

```
int lock = 0;       // global lock
int test_and_set(int* lock);
// returns 1 if exchange
// returns 0 otherwise
int compare_and_exchange(int* value, int compare_value, int swap_value)
{
   int ret = 0;

  while(test_and_set(&lock){
   }
   if(compare_value ==value)
   {
      *value = swap_value;
      ret = 1;
   }
else {
      ret = 0;
   }
   // release lock
   lock = 0;
   return ret;
}
```

32 (20/100). [PH11], page 417, 5.8.

a. [20] <5.6> P0: write 110 <-- 80: Hit in P0's cache, no stall cycles for either TSO or SC

P0: read 108 :Hit in P0's cache, no stall cycles for either TSO or SC

b. [20] <5.6> P0: write 100 <-- 80: Miss, TSO satisfies write in write buffer (0 stall cycles) SC must wait until it receives the data (100 stall cycles)

P0: read 108: Hit, but must wait for preceding operation: TSO = 0, SC = 100

c. [20] <5.6> P0: write 110 <-- 80 : Hit in P0's cache, no stall cycles for either TSO or SC

P0: write 100 <-- 90 : Miss, TSO satisfies write in write buffer (0 stall cycles) SC must wait until it receives the data (100 stall cycles)

d. [20] <5.6> P0: write 100 <-- 80 : Miss, TSO satisfies write in write buffer (0 stall cycles) SC must wait until it receives the data (100 stall cycles)

P0: write 110 <-- 90 : Hit, but must wait for preceding operation: TSO = 0, SC = 100

33. (10/100). 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.

For 4-ary 3-cube, 16*800*10^6 * 32 = 4.096 Tbps.

For 8-node omega, 4*1.2*10^9*64 = 307.2Gbps.

34. (10/100). 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?.

The credit counter has to be 6 large. If the credit size is only 4, the bandwidth would be 4/6 = 2/3 of the peak bandwidth.

35. (10/100). 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.

For 2D dimension-ordered network, the packet arrives is faster than they can be sent, the bottle neck is the router and link delay. So, If we have 32 bytes to send, it have to take 2 + 32 = 34 cycles to reach next hop, because the router delay can be pipe lined , only the initial first 2 cycle routing delay and 32 cycle link delay will count to the total cycles. Therefore, the total cycle is 170.

For wormhole/cut-through switched network, the first flit can reach at 5*(4+2) = 30th cycle, for the rest flits, each flit takes 4 cycles, and there are 7 more flits, so total cycle is 30+4*7 = 58 cycles.

36 (20/100). [PH11], page 420, 5.09.

A. P0,0: read 100

   A. L1 hit returns 0x0010, state unchanged (M)

B. P0,0: read 128.

   A. L1 miss and L2 miss will replace B1 in L1 and B1 in L2 which has address 108.

   B. L1 will have 128 in B1 (shared), L2 also will have it (DS, P0,0)

   C. Memory directory entry for 108 will become <DS, C1>

   D. Memory directory entry for 128 will become <DS, C0>

C. P0,0: write 128 <-- 78

   A. L1 miss and L2 miss will replace B1 in L1 and B1 in L2 which has address 128.

   B. L1 will have 128 in B1 (shared), L2 also will have it (DS, P0,0)

    C. Memory directory entry for 128 will become <DS, C1>
    D. Memory directory entry for 128 will become <DS, C0>

D. P0,0: read 120

    A. L1 miss and L2 miss will replace B1 in L1 and B1 in L2 which has address 120.
    B. L1 will have 120 in B1 (shared), L2 also will have it (DS, P0,0)
    C. Memory directory entry for 108 will become <DS, C1>
    D. Memory directory entry for 128 will become <DS, C0>

E. P0,0: read 120 P1,0: read 120

    A. L1 miss and L2 miss will replace B1 in L1 and B1 in L2 which has address 120.
    B. L1 will have 120 in B1 (shared), L2 also will have it (DS, P0,0)
    C. Memory directory entry for 108 will become <DS, C1>
    D. Memory directory entry for 128 will become <DS, C0>

F. P0,0: read 120
P1,0: write 120 <-- 80

    A. L1 miss and L2 miss will replace B1 in L1 and B1 in L2 which has address 120.
    B. L1 will have 128 in B1 (shared), L2 also will have it (DS, P0,0)
    C. Memory directory entry for 120 will become <DS, C1>
    D. Memory directory entry for 120 will become <DS, C0>

G. P0,0: write 120 <-- 80 P1,0: read 120

    A. L1 miss and L2 miss will replace B1 in L1 and B1 in L2 which has address 120.
    B. L1 will have 120 in B1 (shared), L2 also will have it (DS, P0,0)
    C. Memory directory entry for 120 will become <DS, C1>
    D. Memory directory entry for 120 will become <DS, C0>

H. P0,0: write 120 <-- 80 P1,0: write 120 <-- 90

    A. L1 miss and L2 miss will replace B1 in L1 and B1 in L2 which has address 120.
    B. L1 will have 120 in B1 (shared), L2 also will have it (DS, P0,0)
    C. Memory directory entry for 108 will become <DS, C1>
    D. Memory directory entry for 128 will become <DS, C0>

37 (20/100). [PH11], page 420, 5.10.
a. P0,0: write 100 <-- 80 : Write hit only seen by P0, 0
b. P0,0: write 108 <-- 88 : Write update received by P0, 0 and invalidate received by P3,1
c. P0,0: write 118 <-- 90 : Write miss received by P0, 0 and invalidate received by P1, 0
d. P1,0: write 128 <-- 98: Write miss received by P1,0

References

[PH11] David A. Patterson and John L. Hennessy. *Computer Architecture: A Quantitative Approach*. 5th ed. Morgan Kaufmann Publishers Inc., 2011. isbn: 978-0-12-383872-8. url: `http://booksite.elsevier.com/9780123838728`.

3.1