Homework 6 Solutions

1.  a) No hazards  
    
    b) Data hazard: The second instruction requires the value of $t0 set in the first instruction. Solution: Bypass the writeback stage of instruction 1 to the execution stage of instruction 2. Stalling instruction 2 for 1 cycle will also be necessary. 
    
    c) Data hazard: The third instruction requires the value of $t0 set in the first instruction. Solution: Bypass the writeback stage of instruction 1 to the execution stage of instruction 3. No stall will be necessary. 
    
    d) No hazards  

2.  a) There are 6 instructions, and the loop will execute 99 times. $6 \times 99 = 594$ instructions in total.  
    
    b) Execution will have to stall for two cycles while the third instruction waits for the correct value of $t1$, while the fifth instruction waits for the correct value of $t2$, and again while the sixth instruction waits for the correct value of $t4$. Also, there will be a one cycle delay in between iterations of the loop while the branch address is computed. (Branching is resolved in the decode stage, but the instruction associated with the branch can't be fetched until the next cycle.) So the first iteration of the loop will take 16 cycles, and each subsequent iteration will take an additional 13 cycles. $16 + 98 \times 13 = 1290$ cycles. 
    
    c) The stalls on instructions 3 and 5 can be eliminated by bypassing the writeback stage of the previous instruction to the execution stage. However, we still must stall for two on instruction 6 cycles waiting for the value of $t4$ to be written to the register. Also, there is nothing to be done about the 1-cycle branch prediction penalty between iterations. So the first iteration will take 12 cycles, and each subsequent iteration will take 9 cycles. $12 + 98 \times 9 = 894$ cycles. 

3.  a) $(2^{20}$ bytes of memory) / (16 bytes per block) = $(2^{16}$ blocks)  
    
    b) There are 32 sets, so the set index will be 5 bits. There are 16 bytes in a block, so the byte offset will be 4 bits. The memory address has 20 bits, (4 bits for each of the 5 hexadecimal digits) so there are 11 bits in the tag. 
    
    c) 0x0DB63 in hex is the same as 1101101101100011 in binary. The last 4 bits are the byte offset, and the previous 5 bits, 10110, denote the set. The memory address maps to set 22.  
    
    d) compulsory miss 
    spatial locality hit
compulsory miss
spatial locality hit

4. a) \( (2^{16} \text{ bytes of memory}) / (32 \text{ bytes per block}) = (2048 \text{ blocks}) \)

b) Because a fully associative cache has only one set, the set index will not require any bits. Because the associativity of the cache says that a memory location can map to any of the ways (blocks) in the cache, it is not meaningful to have a block offset. 32 bytes per block requires 5 bits of byte offset. There are 16 bits in the address all together, so the tag is 11 bits.

c) There is only 1 set, and that is the set that 0xF8C9 will map to.