

5. **Starting Some Static (Scheduling) (20 points):** Consider the 2-way superscalar processor we covered in class – a five stage pipeline where we can issue **one ALU or branch instruction** along with **one load or store instruction** every cycle. Suppose that the **branch delay penalty is two cycles** and that we handle control hazards with branch delay slots (since the penalty is two cycles, and this is a 2-way superscalar processor, that would be four instructions that we need to place in delay slots). This processor has **full forwarding** hardware. This processor is a **VLIW** machine. How long would the following code take to execute on this processor assuming the loop is executed 200 times? Assume the pipeline is initially empty and give the time taken up until the completed execution of the instruction sequence shown here. First you will need to schedule (i.e. reorder) the code (use the table below) to reduce the total number of cycles required (but don't unroll it...yet).

Total # of cycles for 200 iterations: \_\_\_\_\_

(Hint – schedule the code first for one iteration, then figure out how long it will take the processor to run 200 iterations of this scheduled code)

Loop:    lw \$t0, 0 (\$s0)  
          lw \$t1, 0 (\$t0)  
          add \$t1, \$s1, \$t1  
          sw \$t1, 0 (\$t0)    # you may assume that this store never goes to the same address as the first load  
          addi \$s0, \$s0, 4  
          bne \$s0, \$s2, Loop

Cycle	1 <sup>st</sup> Issue Slot (ALU or Branch)	2 <sup>nd</sup> Issue Slot (LW or SW)
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		

Now unroll the loop once to make two copies of the loop body. Schedule it again and record the total # of cycles for 200 iterations:

\_\_\_\_\_

Cycle	1 <sup>st</sup> Issue Slot (ALU or Branch)	2 <sup>nd</sup> Issue Slot (LW or SW)
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		