

1. Assuming the five stage MIPS pipeline taught in class with full bypass and stall logic (and shown on the last page of this assignment), explain the potential pipeline hazards in each of the following code segments. For each hazard indicate what action the processor will take to resolve it (e.g., “stall”, “bypass WB to E”, “bypass M to D”).
 - (a)


```
lw $t0, 44($t1)
add $t2, $t3, $t4
sub $t5, $t3, $t4
or $t6, $t4, $t3
```
 - (b)


```
lw $t0, 44($t1)
add $t2, $t0, $t4
sub $t5, $t3, $t4
or $t1, $t4, $t3
```
 - (c)


```
lw $t0, 44($t1)
add $t2, $t3, $t4
sub $t5, $t0, $t4
or $t1, $t4, $t3
```
 - (d)


```
lw $t0, 44($t1)
add $t2, $t3, $t4
sub $t5, $t3, $t4
or $t1, $t0, $t3
```
2. This exercise will examine the following code fragment:


```
loop: lw $t0, 0($t2)
      addi $t1, $t1, 1
      sw $t1, 0($t2)
      addi $t2, $t2, 4
      sub $t4, $t3, $t2
      bnez $t4, loop
```

 Assume that the initial value of \$t3 is \$t2+396.
 - (a) When this code fragment is run, how many instructions will be executed in all?
 - (b) Assume a pipelined processor with no bypassing, where branches are resolved in the D stage. Upon control and data hazards, this processor must stall until the registers are updated with the right data values. How many cycles will this code take to execute?
 - (c) Now assume a pipelined processor with full bypassing (as shown on the back of this assignment). How many cycles will this code require to execute on that processor?
3. Suppose a computer using a *direct mapped* cache has 2^{20} bytes of memory and a cache of 32 blocks, where each block contains 16 bytes.
 - (a) How many blocks of main memory are there?
 - (b) What is the format of a memory address as seen by the cache? That is, what are the sizes of the tag, set index, and byte offset fields? (NB: byte offset = offset of a particular byte within a block)
 - (c) To which cache set will the memory address `0x0DB63` map?
 - (d) What would be the sequence of hit or miss events corresponding to the following sequence of memory references? `0x0DB63, 0x0DB6C, 0x0D36C, 0x0D363`. For hit or miss access explain why it occurred (e.g., “compulsory miss” or “temporal locality hit”).

4. Suppose a computer using a *fully associative* cache has 2^{16} bytes of memory and a cache of 64 blocks, where each block contains 32 bytes.
- How many blocks of main memory are there?
 - What is the format of a memory address as seen by the cache? As above, give the sizes of the tag, set index, and byte offset fields?
 - To which cache set will the memory address $0xF8C9$ map?

