Cpr E 381 Homework 10

1. Problem 6.33 In the example on page 425, we saw that the performance advantage of both the muticycle and the pipelined designs was limited by the longer time required to access memory versus use the ALU. Suppose the memory access became 2 clock cycles long. Draw the modified pipeline. List all the possible new forwarding situations and all possible new hazards and their length.

2. Problem 6.36 We have a program core consisting of five conditional branches. The program core will be executed thousands of times. Below are the outcomes of each branch for one execution of the program core (T for taken, N for not taken).

   Branch 1: T-T-T
   Branch 2: N-N-N-N
   Branch 3: T-N-T-N-T-N
   Branch 4: T-T-T-N-T
   Branch 5: T-T-N-T-T-N-T

   Assume the behavior of each branch remains the same for each program core execution. For dynamic schemes, assume each branch has its own prediction buffer and each buffer is initialized to the same state before each execution. List the predictions for the following branch prediction schemes:

   a. Always taken
   b. Always not taken
   c. 1-bit predictor, initialized to predict taken
   d. 2-bit predictor, initialized to weakly predict taken

   What are the prediction accuracies?

3. Problem 6.47 The following code has been unrolled once but not yet scheduled. Assume the loop index is a multiple of two (i.e, $10 is a multiple of eight).

```
Loop:  lw     $2,     0($10)
       sub    $4,     $2,     $3
       sw     $4,     0($10)
       lw     $5,     4($10)
       sub    $6,     $5,     $3
       sw     $6,     4($10)
       addi   $10,    $10,    8
       bne    $10,    $30,    Loop
```

   Schedule this code for fast execution on the standard MIPS pipeline (assume that it supports addi instruction). Assume initially $10 is 0 and $30 is 400 and that branches are resolved in the MEM stage How does the scheduled code compare against the original unscheduled code?

4. Problem 7.9 Here is a series of address references given as word addresses: 2, 3, 11, 16, 21, 13, 64, 48, 19, 11, 3, 22, 4, 27, 6 and 11. Assuming a direct-mapped cache with 16 one-word blocks that is initially empty, label each reference in the list as a hit or mis and show the final contents of the cache.

5. Problem 7.10 Using the series of references given in Exercise 7.9 (previous problem), show the hits and misses and final cache contents for a direct-mapped cache with four-word blocks and a total size of 16 words.

6. Problem 7.12 Compute the total number of bits required to implement the cache in Figure 7.9 on page 486. This number is different from the size of the cache, which usually refers to the number of bytes of data stored in the cache. The number of bits needed to implement the cache represents the total amount of memory needed for storing all the data, tags and valid bits.

7. Problem 7.16 Cache C1 is direct-mapped with 16 one-word blocks. Cache C2 is direct-mapped with four-word blocks. Assume that the miss penalty for C1 is 8 memory bus clock cycles and the miss penalty for C2 is 11 memory bus clock cycles,. Assuming that the caches are initially empty, find a reference string for which C2 has a lower miss rate but spends more memory bus clock cycles on case misses than C1. Use word addresses.