EECS 314: Computer Architecture

Spring 2012

Homework #4

- Each problem is worth 10 points
- This homework like all others needs to be done and submitted individually
- For the MIPS program (problem 4), you need to verify it using the SPIM simulator and provide a snapshot of 2-3 sample runs along with a copy of the source code.

Pipeline and Branch Prediction

1. The 80x86 ISA has arithmetic instructions that can directly access memory. Write a paragraph or two explaining why it would be hard to add this instruction to the MIPS pipeline described in this chapter. (**Hint:** You may have to add one or more additional stages to the pipeline.)

2. We have a program core consisting of four 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-N-T-N-T-N
   Branch 2: T-T-T-N-T
   Branch 3: T-T-N-T-T-N-T
   Branch 4: N-N-N-N

   Assume a 2-bit predictor, initialized to weakly predict taken. 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 initialized to the same state before each execution. List the predictions and calculate the prediction accuracy.

3. In dealing with data hazards in a pipeline, we have seen how data forwarding addresses stalls due to data hazards and inevitable stalls on a use following a load. Can we rewrite the following code to *minimize* performance on this datapath – that is, can we reorder the instructions so that this sequence takes the most clock cycles to execute while still obtaining the same result? Specify with reasons if the performance cannot be minimized further.

```
lw    $2, 100($6)
lw    $3, 200($7)
add   $4, $2, $3
add     $6, $3, $5
sub      $8, $4, $6
lw      $7, 300($8)
beq     $7, $8, Loop
```

Recursive programming in MIPS

4. Write a recursive routine for Binary Search of an element from a sorted list of N integers. You should be able to input N, the sorted list and the value to search from the keyboard. Your program will return if the value is found and if found, the array index where it is found. A main (driver) function should call the routine.

Notes:
A binary search is a technique for finding a particular value in a linear array, by "ruling out" half of the data at each step. A binary search finds the median, makes a comparison to determine whether the desired value comes before or after it, and then searches the remaining half in the same manner.

The search begins by examining the value in the center of the list; because the values are sorted, it then knows whether the value occurs before or after the center value, and searches through the correct half in the same way.

The pseudocode:

```
function BinSearch(a, value, left, right)
    if right < left
        return "No Match"
    mid := (left + right) / 2
    if a[mid] = value
        return mid
    if value < a[mid]
        binSearch(a, value, left, mid-1)
    else
        binSearch(a, value, mid+1, right)
```

Embedded Systems

4. As discussed in class, computer systems are broadly of two types – general-purpose systems (e.g. your laptop, tablet or desktop computers) and embedded systems (e.g. digital camera, cell phone and computer systems inside consumer electronics, medical electronics such as pacemaker, and automobile electronics). Embedded systems are designed, optimized and used for specific applications; run software routines tailored to the hardware platform (referred to as embedded software); and meet specific power-performance requirements for target application. Today, these systems are used pervasively in many spheres of our life spanning numerous application domains.

Find an interesting application of embedded system. Describe the application in few sentences. Feel free to add any image of the target system. Then, report the following:

(1) The type of processor hardware used in the system. Mention the ISA type. Mention the number of cores, operating frequency and average power, if available
(2) The amount of memory integrated in the system and the type of memory (e.g. DRAM, flash or others)
(3) The specific input/output devices used in the system
(4) Design requirements for the embedded system (such as power, performance, reliability, and upgradability). Just mention which parameters are important for the specific application.
(5) Any other relevant information about the embedded hardware or embedded siftware

The total report including response to the above five points should be between ½ - 1 page.