Lab1 - Computer Organization, Data Representation and Pseudocode

Due: Monday September 10, 2012

Central Processing Units (CPU)

1. Short answer: Describe the purpose of each of the CPU components listed below.

    - General purpose register
    - Instruction register
    - Control Unit
    - Arithmetic logic unit
    - Program counter
    - Instruction decoder

2. What is the fetch-decode-execute cycle ?

Data Representation

3. Consult the table of ASCII character set in our handout. Show the binary (base two) representation (in contiguous memory) of the word "dog" (option: you can use "cat" instead).

    □□□□□□□□ □□□□□□□□ □□□□□□□□

For all problems below, show your work.

4. Convert the following binary string to hexadecimal notation.

    `1 0 1 1 0 0 1 1   1 0 0 1 1 1 0 0   0 0 1 0 1 1 1 1`

5. Convert the following hexadecimal number to binary.

    `b 6 3 e`

6. Convert the following binary string to base ten.

    `1 0 1 1 0 0 1 1`

7. Use Horner's algorithm to convert the following hexadecimal number to base ten.

    `3 d a 7`

8. Use the remainder method to convert the following base ten number to hexadecimal.

    `2467`

9. The following proposed algorithm is very similar to the one presented previously for finding the largest element in an array. Trace this algorithm with the given input array. What happens ? What do you conclude about this proposed algorithm ?

   Input: An integer `n`, and an array `A` containing `n` integers.

   Output: The largest integer in the array.

   Method:

   ```
   t = A[0]               // A[0] is our first candidate for the largest.
   i = 1                  // i is our variable index.
   while ( i < n ) {      // Start a loop.
      if ( t < A[i] ) {
         t = A[i]
         i = i + 1
      }
   }                      // end while loop
   write t                // Output the answer
   ```

22	37	11	17	30
0	1	2	3	4

10. Refer to the (correct) algorithm in the handout to find the largest element of the array. Modify that algorithm to add up all the numbers in the array, and output the total.

    Hint: Initialize the total to zero. To access each element of the array, use a counter (`i`), similar to the way a counter is used in the example algorithm. As each array element is accessed, update the total using the following idea: replace the current total with the current total plus the current array element.