

CSE222 Computer Architecture

Homework Set 05

(Review)

1. MIPS Programming
 - a. Registers set: 32 registers + PC, hi, lo
 - b. R-, I-, J-type instructions and their format
 - c. Logic instructions: `and`, `or`, `nor`; `addi`, `ori`, `xori`
 - d. Shift instructions: `sllv`, `srlv`, `sra`; `sll`, `slr`, `sra`;
 - e. Arithmetic instructions: `add`, `sub`, `mult`, `div`, `mfhi`, `mflo`
 - f. Conditional instructions: `beq`, `bne`, `slt`,
 - g. Jump instructions: `j`, `jr`, `jal`
 - h. Array
 - i. Function, `jal`, `jr`; Stack
 - j. MIPS syscalls: code 1-4, 5-8, 10, 42

(Exercise)

1. Generate the first 20 Fibonacci numbers and save them in an array. Display array elements in 2 lines.
2. Define a function to calculate the average number of an (integer) array. Test this function with an array; display array and average value
3. Write MIPS code to prompt user to enter an integer number. Check if the entered number is positive or not, if negative, ask user to enter again until a positive number is entered; Calculate and display the sum from 1 to this positive number, for example, if input number is 6, calculate the sum from 1 through 6. Display this positive number and the sum. Define functions to check if a number is positive or negative; and calculate the sum.
4. (1) Define a function to count how many 1's in an integer number. Return this count
(2) Test: input an integer number; pass this number to the function; display return value
5. (1) Define an integer array of size 10;
(2) Define a function to initialize integer array with **odd random number**. The input parameters to this function are: array (address); array size; lower bound and upper bound of random number. (Call function defined in (3) to check if the random number is odd or not)
(3) Define a function to check if the input parameter is odd or not
(4) Test above functions with array defined in (1). Display array