CpE213 – Digital Systems Design

Homework 2
WIMP51

NOTE: Just as on tests, I expect you to show all your work on homework problems. A correct answer with no work might not be given full credit.

1. **a)** Write an ASM program for the WIMP51 which performs the same function as the code:

   temp = 0
   for i = 0 to 5
       temp = temp + i
   next i
   end

   Your ASM program should count <u>up</u>.

   b) Give the machine code for the first three and last three instructions (Assuming the first instruction is at code address 0).

   c) Prove your code works by showing a table of values in appropriate registers and memory locations as the program executes (like we did in class for program 3*5).

2. Draw a timing diagram over a single *instruction* cycle showing control signals when the WIMP51 executes the instruction: ADDC A,R6. Assume the instruction is located at memory address 42H. The signals that should appear on your plot include: data, PSEN, address, ir_we, reg_we, reg_sel, reg_in, aux_we, alu_op, acc_we, pcalu_up, pc_we, and clock. If the value of a signal does not matter, you can eliminate it from the plot. Please mark each state (fetch, decode, execute).

3. The WIMP51 cannot save data to external RAM as it is currently configured. We would like to modify the WIMP51 to allow it to do this. Specifically, we would like to add hardware to perform the instruction "MOV D,A", which would move the value of the accumulator into external memory location D. For example, "MOV 42H,A" would move the value in the accumulator to location 42H. Assume D can range from 0-255.

   (a) Write down the signal lines that would be needed external to the WIMP51 to write to RAM.

   (b) Write down the steps that values would have to appear on these external signal lines to write to RAM (hint: we did something similar when talking about memory).

   (c) Add hardware to the WIMP51 that allows this to happen (I would probably copy the slide of the WIMP51 and write on that). Explain briefly how your hardware works (e.g. step 1: ACC-> data, XX -> addr, step 2:..., etc).

   (d) Create an opcode for the "MOV D,A" that will not conflict with the other instructions (for example, your opcode cannot start with 01001111 because that's the opcode for ORL A, R7). Show the opcode for the specific instruction "MOV 42H, A".