

EECS 370

Discussion 1

Week of 1/16 – 1/22

- I) Two's complement
 - a. Way to represent signed integers, uses most significant bit as “sign bit”
 - b. Convenient: no need for additional hardware logic when adding and subtracting!
 - c. Negative numbers count backwards and wrap around
 - d. To negate: invert bits and add one
 - e. Exercise: -37 in 8-bit two's-complement
- II) Memory architecture
 - a. Stack/heap
 - b. Static data
 - c. Text
 - d. Reserved
- III) Basic C
 - a. I/O
 - i. No cout, cin, endl!
 - ii. Output (#include <stdio.h>)
 - 1. printf(char *format, arg1, arg2...)
 - 2. printf(“Hello World!\n”);
 - 3. %d for integers, %x for hex, %s for string
 - iii. Input (#include <stdio.h>)
 - 1. scanf(char *format, (address of arg1)...))
 - 2. scanf(“%d”, &int1)
 - 3. char left[32], right[32];
scanf(“%s=%s”, left, right)
 - iv. C-arrays (and <string.h>)
 - 1. C arrays are almost identical to char *
 - 2. Bad things to do
 - a. array1 = array2; use strcpy!
 - b. !=, >, <, ==; use strcmp
 - v. Function prototypes
 - vi. Bitwise operations
 - 1. ~, <<, >>, &, |
 - 2. Setting bits
 - a. Exercise: Set the 6th bit of 87 using C code.

- IV) Project 1
 - a. Overview
 - i. Assembler
 - ii. Simulator
 - iii. Multiplier
 - b. Topics
 - i. .fill
 - ii. Displacement relative to PC and beq
 - iii. jalr
 - iv. Quick review of all LC-2Kx instructions
 - v. Labels
 - vi. Unneeded error checking
 - c. Hints
 - i. Assembler
 - 1. Symbol resolution
 - a. First pass: calculate addresses for all labels
 - b. Second pass: calculate offset for beq instructions
 - ii. Shifting makes logic much easier
 - iii. No negation instruction in LC-2Kx... think about how to implement it