

Union College

ECE352

Assignment 2

Due Date: Thursday April 17th at 10:30AM

Problem 1.

Complete the following using stacks. This program is to be written in assembler.

The stack is a "first in, last out" structure. It is important to initialize the stack pointer so that your stack is large enough to contain all the bytes that you will push onto it. The stack pointer is initialized to 0x07, so the first value pushed onto the stack is stored at 0x08, which is the value of R0 in register bank 1. If more than one register bank is to be used, the stack pointer should be moved at the beginning of the program. Choose a location in the data memory that is not being used. We will use location 0x30 as the first stack data value, so we will initialize the stack to 2Fh.

Write a program that initializes the stack pointer, and then pushes the following 4 memory locations on the stack, and then pops them back out so that the data is re-ordered. The following "before" and "after" example illustrates how it should work.

Before		After	
Address:	Data	Address:	Data
70	34	70	89
71	f2	71	66
72	66	72	f2
73	89	73	34

Since you do not know what data is initially in RAM, add some instructions to initialize these memory locations so that you can tell if your program works. (If the locations all held the same data value, then you would not be able to see the changes). Name your program stacks.asm and include a printout of your program with your assignment.

Problem 2.

Write a program in assembler to move 4 data values from external memory, starting at 2000h, to RAM, starting at address 20h. Use a loop to sequence through the memory locations. After they are moved, put bit 4 of location 21 in the carry flag.

Problem 3.

Write a program in assembler to compute the equation: $Z = X/(Y - P)$. You may assume that X, Y, and P are held in registers R0, R1, and R2, and the Z result should be written to memory at location 60h.

Problem 4.

In class we wrote a program to blink the light using time 0 and polling the overflow flag in C. Rewrite this program in assembler.

Problem 5.

In class we wrote a program to blink the light using time 0 and generating an interrupt from the overflow flag in C. Rewrite this program in assembler.

Problem 6.

Modify the C program we wrote in class to display the numbers 0 to F on the seven segment display to use timer0 and polling the overflow flag instead of nested loops.

Problem 7.

Modify the C program we wrote in class to display the numbers 0 to F on the seven segment display to use timer0 and interrupts instead of nested loops. Write code for the ISR such that the delay between displays is close to 2 seconds.