CS61 Homework 10

HOMEWORK SET (6 questions, 6 points): Complete the following exercises.

BEFORE YOU START:

Go back and study carefully the solutions to questions 5.40 & 5.41 from last week's homework!!

Question 1: Chapter 7, exercise 7.4

Create the symbol table entries generated by the assembler when
translating the following routine into machine code:

```
        .ORIG   x301C
        ST      R3, SAVE3
        ST      R2, SAVE2
        AND     R2, R2, #0
TEST    IN
        BRz     TEST
        ADD     R1, R0, #-10
        BRn     FINISH
        ADD     R1, R0, #-15
        NOT     R1, R1
        BRn     FINISH
        HALT
FINISH  ADD     R2, R2, #1
        HALT
SAVE3   .FILL   X0000
SAVE2   .FILL   X0000
        .END
```

	.ORIG x301C	
x301C	ST R3, SAVE3	
x301D…..	ST R2, SAVE2	
	.END	

Symbol	Address
TEST	x301F
FINISH	x3027
SAVE3	x3029
SAVE4	x302A

Questions 2 - 6: Chapter 9, exercises 9.1 (8.1 in 2nd edition), 9.16 (9.2 in 2nd edition), 9.23 (9.10 in 2nd edition), 9.26 (9.13 in 2nd edition), 9.28 (9.15 in 2nd edition)

9.1
a. What is a device register?
A device register is a register that is used for communication between peripherals and the microprocessor. These are generally part of the I/O interface because it is used for data exchange with external devices.

b. What is a device data register?
This is a register that is used for transferring data between the microprocessor and a peripheral device. Data is written to and read from by the microprocessor and the connected device. It is the register that holds the data being transferred and is what makes the exchange of data between the processor and peripheral devices possible.

c. What is a device status register?
It is the register that shows the status of a peripheral device. It indicates things such as data availability, error conditions, busy/idle status, or interrupt requests. The processor can make decisions based on this information to make user experience better or even possible.

9.16 a. How many trap service routines can be implemented in the LC-3? Why?
256 trap service routines can be potentially implemented. This is because there is a max of 256 vector locations.

b. Why must a RET instruction be used to return from a TRAP routine?
Why won't a BR (Unconditional Branch) instruction work instead?
RET uses the address stored in R7 to go back to wherever the subroutine was called from+1 in address. The Trap(JSRR) uses R7 to store the address of the next sequential instruction.

c. How many accesses to memory are made during the processing of a
TRAP instruction? Assume the TRAP is already in the IR.
Memory is accessed three times: one to push PSR to the system stack, one to push PC to the system stack and one to see the register(R7) which contains where the TRAP jumps to.

9.23 The starting address of the trap routine is stored at the address specified in the TRAP instruction. Why isn't the first instruction of the trap routine stored at that address instead? Assume each trap service routine requires at most 16 instructions. Modify the semantics of the LC-3 TRAP instruction so that the trap vector provides the starting address of the service routine.

The first instruction of the trap is not where the stored address is because then it would not be a subroutine. Subroutines are useful because it provides the opportunity to somewhat use object oriented programming. This lets programmers reuse code without rewriting code. If the code within the subroutine were written in the 'main' then it would not have these benefits.

9.26 The following program is supposed to print the number 5 on the screen. It does not work. Why? Answer in no more than ten words, please.

```
            .ORIG   x3000
            JSR     A
            OUT
            BRnzp   DONE
A           AND     R0,R0,#0
            ADD     R0,R0,#5
            JSR     B
            RET
DONE        HALT
ASCII       .FILL   x0030
B           LD      R1,ASCII
            ADD     R0,R0,R1
            RET
            .END
```

When JSR is called within JSR R7 gets overwritten and the PC does not correctly go back up twice.

9.28 Suppose we define a new service routine starting at memory location x4000. This routine reads in a character and echoes it to the screen. Suppose memory location x0072 contains the value x4000. The service routine is shown below.

```
            .ORIG x4000
            ST R7, SaveR7
            GETC
            OUT
            LD R7, SaveR7
            RET
  SaveR7    .FILL x0000
```

a. Identify the instruction that will invoke this routine.
JSR label_0072
b. Will this service routine work? Explain.
The routine will work since it prompts the user to enter a char, stores it in R0 and then immediately outputs it back to the console which is what it is meant to do. It also backs up R7 which is not completely necessary but the code works regardless.