Homework 5 | 2/19/14 TCSS 371 Ben Foster

7.10 The following program fragment has an error in it. Identify the error and explain how to fix it.

```
        ADD  R3, R3, #30
        ST   R3, A
        HALT
A       .FILL #0
```

The error is in the first line because 30 is too big of a value to be represented by the immediate value bits.

Fix:

```
        LD  R4, B
        ADD R3, R3, R4
        ST  R3, A
        HALT
A       .FILL #0
B       .FILL #30
```

This error will be detected when this code is assembled.

7.12 What does the following LC-3 program do?

```
        .ORIG
        AND  R5, R5, #0     ; Clear R5
        AND  R3, R3, #0     ; Clear R3
        ADD  R3, R3, #8     ; Put 8 in R3
        LDI  R1, A          ; load contents of A into R1
        ADD  R2, R1, #0     ; Store contents of R1 into R2
AG      ADD  R2, R2, R2     ; ADD R2 to itself
        ADD  R3, R3, #-1    ; decrement R3 by 1
        BRnp AG             ; If R3 is not zero, goto AG
        LD   R4, B          ; load contents of B into R4
        AND  R1, R1, R4     ; And R1 and R4
        NOT  R1, R1         ; Not R1
        ADD  R1, R1, #1     ; Add 1 to complete 2's complement
        ADD  R2, R2, R1     ; Add R2 and R1
        BRnp NO             ; If R2 is zero, Halt program.
        ADD  R5, R5, #1     ; Add 1 to R5
NO      HALT                ; HALT program
B       .FILL xFF00         ; Data -256
A       .FILL x4000         ; Data
        .END
```

This program compares the left and the right byte of whatever is at memory location x4000. If the two bytes are the same then 1 is stored in R5.

Homework 5 2/19/14 TCSS 371 Ben Foster

7.14 a. Assemble the following program:

```
      .ORIG x3000
      STI   R0, LABEL
      OUT
      HALT
LABEL .STRINGZ "%"
      .END
```

b. Replace exactly one opcode in this program with the correct opcode to make the program work as intended
Replace STI with LEA

c. Explain what the strange behavior was and why the program behaved that way
The strange behavior is that it infinitely output a zero to the console. It did this because it was loading x3000 into address x0025 which not part of the program, causing weird things to happen.

7.24 We want the following program fragment to shift R3 to the left by four bits, but it has an error in it. Identify the error and explain how to fix it.

```
     .ORIG x3000
     AND  R2, R2, #0
     ADD  R2, R2, #4
LOOP BRz  DONE
     ADD  R2, R2, #-1
     ADD  R3, R3, R3
     BR   LOOP
DONE HALT
     .END
```

When the program is executing, the last opcode to set the condition codes is ADD R3, R3, R3. This never zero, therefore the BRz never activates and so the program goes through an infinite loop. Fix:

```
     .ORIG x3000
     AND R2, R2, #0
     ADD R2, R2, #4
LOOP ADD R3, R3, R3
     ADD R2, R2, #-1
     BRnp LOOP
     HALT
     .END
```