ICS Lab Report #1

StuID: **Name:**

Problem Setting

Here we are using LC3 assembly language todetect whether a 16-bit integer at address x3100 contains 3 consequent 1's in its binary form. If it contains, we set register R2 to 1, or set to 0 if does not.

Algorithm Specification

Here is a pseudocode representation of my program.

```
R1 <- input
R4 <- 0
for i = 0 to 15
  if R1 & (2^i) == 0
    R4 <- 0
 else
    R4 <- R4 + 1
    if R4 == 3
      R2 <- 1
      halt
R2 <- 0
halt
```

This main idea of the program:

Test the bits of the input one by one. If consequent 1's count(represented by R4) is 3, then set R2 to 1 and halt. If never find such sequence, set R2 to 0 and halt.

LC3 Implementation

```
LOOP
AND R5, R1, R2 ;R2 has been set to 1. Use AND to test the bit.
ADD R2, R2, R2 ;Let R2 multiplied by 2.
ADD R5, R5, #0 ;Recollect the AND result
BRZ NOT1
```

```
ADD R4, R4, #1; Or let R4 added by 1
ADD R7, R4, #-3
BRZ RET1 ;If R4 reached 3, set R2 and halt.
BRNZP LOOPCDT; or skip to next loop
NOT1
AND R4, R4, #0 ;If resulting 0, then set counter(R4) to 0
LOOPCDT ;next loop
ADD R3, R3, #1
ADD R7, R3, #-16
BRN LOOP ;If all bits tested, exit.
```

This segment presents the `for` part of the pseudocode.

Check Problem

Q: If test 3 bits per iteration in LOOP, then how many iterations needed?

A: 14 iterations in theory. However, LC3 doesn't really have a right shift instruction so testing 3 bits using a similar idea to the program is hard to implement.