

# LCS-2021-Lab1-Report

## Main algorithm:

Set r1 to be 0x0007 (i.e. 0000 0000 0000 0111), r2 to be 0, r3 to be 14. And load the data on 0x3100 on r0.

Do a 'AND' instruction between r1 and r0, save the result on r4. If r4 == r1, then the number contains three consecutive 1s, and we set r2 to be 1, then halt. Otherwise we do a left shift on r1, let r3 = r3 - 1, repeat the process on this paragraph until r3 = 0, and we break. Finally we halt the machine.

The main process is like this ( the picture I draw is too ugly, so i use this ).

```
// all data has 16 bits
r0 = M[0x3100];
r2 = 0;
r1 = 0x0007;
r3 = 14; //r3 is the counter
while(r3 != 0){
    r4 = r0 & r1;
    if(r4 == r1){
        r2 = 1;
        break;
    }
    r3 = r3 - 1;
    r1 << 1;
}
halt();
```

## Essential parts:

initialize the register

```
0010 000 011111111 ;ld r0
0101 010 010 1 00000 ;r2<-0
0101 001 001 1 00000 ;r1<-0
0001 001 001 1 00111 ;r1=7
0101 011 011 1 00000 ;r3<-0
0001 011 011 1 01110 ;r3 = 14 #TODO check
```

cycle and halt

```

15 0000 010 000001010 ;if r3 = 0, end the loop
16 0101 100 000 0 00 001 ;r4 <- r0 & r1
17 1001 100 100 111111 ; r4 = not(r4)
18 0001 100 100 1 00001 ;r4 = -r4
19 0001 100 100 0 00 001 ;r4 = r4 + r1
20 0000 010 000000100 ;check if r4 = r1
21 0001 001 001 0 00 001 ; r1 << 1
22 0001 011 011 1 11111 ; r3 = r3 -1
23 0000 111 111110111 ;jump to line 15
24 0000 111 000000001 ;jump to halt
25 0001 010 010 1 00001
26 1111 0000 0010 0101 ;halt

```

## TA's check

1. Question: describe the algorithm you use to solve this problem

My answer: (just like above, part 1 of this report) Set r1 to be 0x0007 (i.e. 0000 0000 0000 0111), r2 to be 0, r3 to be 14. And load the data on 0x3100 on r0.

Do a 'AND' instruction between r1 and r0, save the result on r4. If r4 == r1, then the number contains three consecutive 1s, and we set r2 to be 1, then halt. Otherwise we do a left shift on r1, let r3 = r3 - 1, repeat the process on this paragraph until r3 = 0, and we break. Finally we halt the machine.

There is no more question.

July 15, 2021 By 庄毅非