# 1. MAIN CODE

0101 100 000 000 011; R4=R0 AND R3

1001 100 100 111111; Not R4

0001 100 100 100001; R4+1

0001 100 100 000 011; R4=R4+R3

0000 010 000000100; R4为0 有连续1 结束

0001 001 001 111111; 不为0 R1-1

0000 010 000000011; R1为0 没有连续1 结束

0001 011 011 000 011; R3=R3+R3 即左移1位

0000 111 111110111; 继续遍历

0001 010 010 100001; R2=1

1111 0000 00100101; halt

# 2. ALGORITHM

If a certain bit holds 1 in a bit string, AND this bit string with another bit string that holds 1 at the same bit and the result should be 1 at that particular bit and 0 at other places. AND your source string with string 00…00111 and check the answer is 00…00111 or not, then you can tell whether the source string holds 1`s in its last 3 bits. By analogy, do this check 14 times (there`re 14 possible cases altogether for a 16-bits string holding 3 consecutive 1`s) so that you can find the answer.

In my code, R0 holds the resource bit string, R1 is the counter initialized #14, R2 holds the result, R3 holds the string which R0 AND with, R4 holds temporary values.

# 3. TA`S CHECKING

TA let me explain my algorithm which has already showed above and asked how many lines I accomplish this lab. Nothing else.