

# 1.Algorithm

For the specific implementation, please see the following comments, which only say the general idea:

Detect three consecutive ones, and use mask to perform each three bit and operation. Here, considering that if it is the original code, it is to detect that the result of the previous step is 111, so take the reverse code, so that we only need to detect that the three digits & mask = 0. The condition for getting out of the loop is that it's found, or that the highest three digits of mask are 111.

```
load word and Mask
word = ~word;
bool find = 0;
while(mask > 0){
    if(word & mask == 0) find = 1, break;
    mask = mask << 1;
}
if(mask & word == 0) find = 1; // avoid 1110 .... .... not find

if(find = 1) R2 =1;
else R2 = 0;
```

## 2.Essential parts of code with sufficient comments

```
0011000000000000
0101010010100000
1010000000001011
0010011000001011
1001000000111111
0101001000000011
0000010000000100
0001011011000011
0000001111111100
0101001000000011
0000101000000010
0001010010100001
1111000000100101
1111000000100101
0011000100000000
0000000000000111
; .ORIG x3000
; AND R2,R2,#0      R2 is ans intial NO
; LDI R0,WORD        Get Mem[Mem{WORD}] ,Mem{WORD} = x3100
; LD  R3,MASK        Get Bit mask = 0000 0000 0000 0111
```

```

; NOT R0,R0      eg. if we use WORD ,match 111 means mask & WORD = 7, mask &
WORD - 7 = 0 ,is little hard
;
;               So we use ~WORD ,match 111 means mask & ~WORD = 0,
;               because any non consecutive like 011 , must generate positive
result
; LOOP
; AND R1,R0,R3   mask & ~WORD
; BRZ YES        if mask & ~WORD == 0 break
; ADD R3,R3,R3   else mask = mask << 1
;               when mask = 0111 0000 0000 0000 can't match ,the only
possible match number is 1110 .... ....
; BRP LOOP       and this loop don't happen when mask = 1110 0000 0000 0000
;
;
; AND R1,R0,R3   if ~(EXXX) & E000 = 0,then BRNP NO won't happen ,then it is
YES ,the last case satisfy
; BRNP NO

; YES ADD R2,R2,#1
;   TRAP x25      return answer to r2 and HALT

; NO  TRAP X25    HALT ,no need to change r2

; WORD .FILL x3100 Mem[WORD] = &A      LDI it
; MASK .FILL x0007 Mem[MASK] = x0007   LD it

; .END
;

```

## 3. Questions

---

### 1. Briefly state your algorithm

---

see First Part

### 2. Can you load WORD without using the LDI instruction

---

LD can be used to replace LDI, because LD uses the immediate number of 9bit, the range is -255 ~ 256, it can be loaded x3100 data.