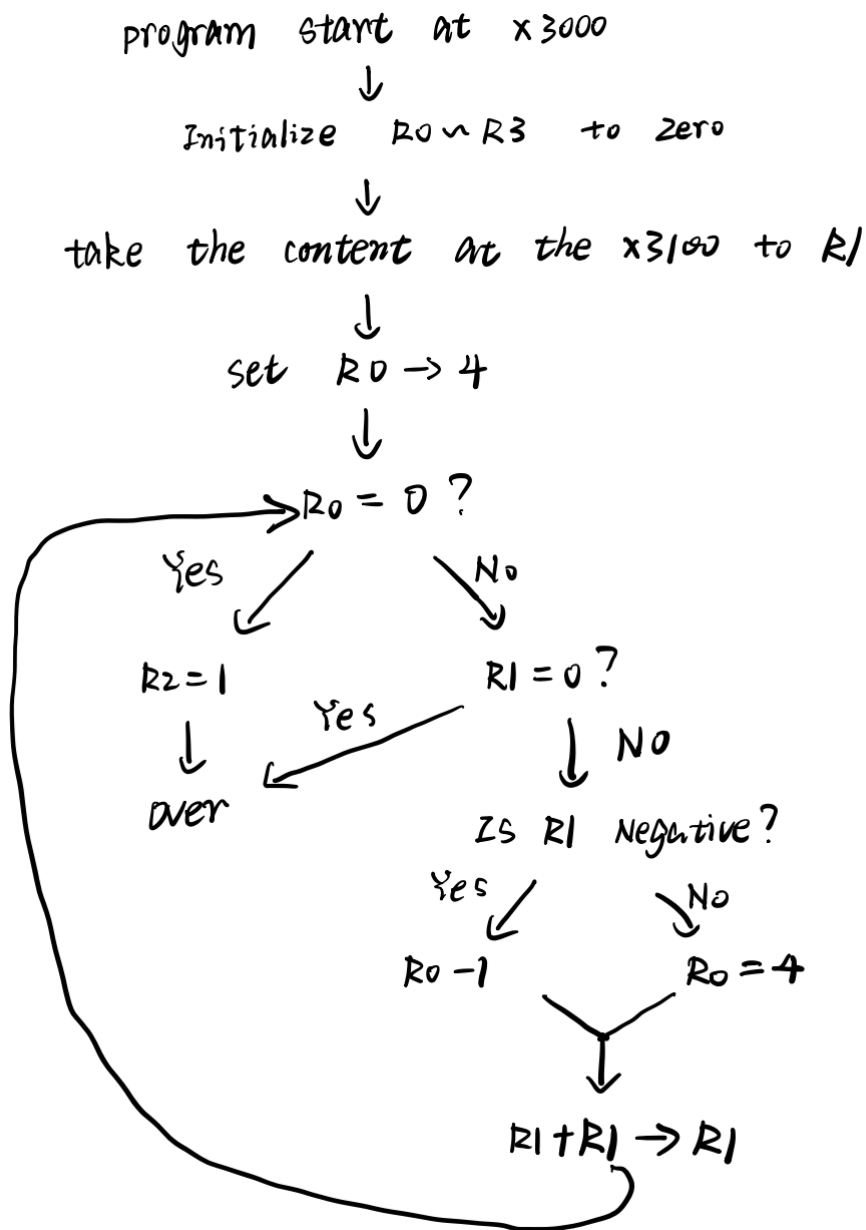# Lab1:F-Word

Algorithm Explanation

By add R1 withR1, you can move the 0&1 to left a bit wise i.e 1111->1110 0111->1110

I design a counter(R0) to monitor whether there are 4 continous 1.

If the R1 represnet 0, there no need to continue.

Part of Code

```
0011 0000 0000 0000 ;
0101 000 000 1 00000;x3000,Initialize R0-R3
```

```
0101 001 001 1 00000;
0101 010 010 1 00000;
0101 011 011 1 00000;
0010 001 0 1111 1011;#x3004,Load the content in x3100 to R1
0001 000 000 1 00100;Set R0=4
0001 000 000 1 00000;Load R0
0000 010 000001000  ;If R0=0,succeed!
0001 001 001 1 00000;Load F-Word
0000 010 000000111  ;If F-Word=0,Failed!
0000 100 000000010  ;If F-Word is Negative now,turn to line 15
0101 000 000 1 00000;Set R0=0
0001 000 000 1 00101;Set R0=5
0001 000 000 1 11111;R0-1
0001 001 001 000 001;moveleft
0000 111 111110110  ;loop,turn to line8
0001 010 010 1 00001;Set R2=1
1111 0000 0010 0101 ;Over!
```

Q&A

Q:What's the main idea about your program?

A: (what I talked above)