[](https://leetcode.com/ziyihao/)

An General Way to Handle All this sort of questions.

[ziyihao](https://leetcode.com/ziyihao/)

555

Aug 29, 2015

this kind of question the key idea is design a counter that record state. the problem can be every one occurs K times except one occurs M times. for this question, K =3 ,M = 1(or 2) .  
so to represent 3 state, we need two bit. let say it is a and b, and c is the incoming bit.  
then we can design a table to implement the state move.

current incoming next

a b c a b

0 0 0 0 0

0 1 0 0 1

1 0 0 1 0

0 0 1 0 1

0 1 1 1 0

1 0 1 0 0

like circuit design, we can find out what the next state will be with the incoming bit.( we only need find the ones)  
then we have for a to be 1, we have

current incoming next

a b c a b

1 0 0 1 0

0 1 1 1 0

and this is can be represented by

a=a&~b&~c + ~a&b&c

and b can do the same we , and we find that

b= ~a&b&~c+~a&~b&c

and this is the final formula of a and b and just one of the result set, because for different state move table definition, we can generate different formulas, and this one is may not the most optimised. as you may see other's answer that have a much simple formula, and that formula also corresponding to specific state move table. (if you like ,you can reverse their formula to a state move table, just using the same way but reversely)

for this questions we need to find the except one  
as the question don't say if the one appears one time or two time ,  
so for ab both

01 10 => 1

00 => 0

we should return a|b;  
this is the key idea , we can design any based counter and find the occurs any times except one .  
here is my code. with comment.

public class Solution {

public int singleNumber(int[] nums) {

//we need to implement a tree-time counter(base 3) that if a bit appears three time ,it will be zero.

//#curent income ouput

//# ab c/c ab/ab

//# 00 1/0 01/00

//# 01 1/0 10/01

//# 10 1/0 00/10

// a=~abc+a~b~c;

// b=~a~bc+~ab~c;

int a=0;

int b=0;

for(int c:nums){

int ta=(~a&b&c)|(a&~b&~c);

b=(~a&~b&c)|(~a&b&~c);

a=ta;

}

//we need find the number that is 01,10 => 1, 00 => 0.

return a|b;

}

}

this is a general solution . and it comes from the Circuit Design on course digital logic.