When we detect overflow in adding two k-bit using 2C, we can consider two cases.
First, two bits have same sign. Two bits have positive sign or negative sign.
Last, two bits have opposite sign. One has a positive sign, the other has a negative sign.

If two bits have opposite sign. The sum will exist in between two bits.
So if two bits have opposite sign, will not occur overflow.

If the bits have same sign. We can suggest will occur overflow. The sum will too small or large. So we can consider four cases like under the picture.

(1)

Ck : 1	Ck-1 : 0	Ck : 1	Ck-1 : 1
	1		1
	1		1

(2)

Ck : 0	Ck-1 : 0	Ck : 0	Ck-1 : 1
	0		0
	0		0

(1) Case two bits have negative sign and carried in 0 or 1.
In carried in '0'. The s is '0'. So the sum's sign is positive. But two bits have negative sign.
In that case occur overflow.
In carried in '1'. Ths s is '1'. So the sum's sign is negative. So two bits and sum have same sign. In that case don't occur overflow.

(2) Case two bits have positive sign and carried in 0 or 1.
In carried in '0'. The s is '0'. So the sum's sign is positive. So two bits and sum have same sign. In that case don't occur overflow.
In carried in '1'. The s is '1'. So the sum's sign is negative. But two bits have positive sign.
In that case occur overflow.

Input			Output		
A-sign	B-sign	Ck-1	Ck	Sum-sign	Overflow
0	0	0	0	0	X
0	0	1	0	1	O
0	1	0	0	1	X
0	1	1	1	0	X
1	0	0	0	1	X
1	0	1	1	0	X
1	1	0	1	0	O
1	1	1	1	1	X

We can notice that overflow occurs only when Ck ≠ Ck-1 .
So we can define V(overflow) = Ck xor Ck-1.