Roll - 200050006

Ques - 3:

In this question, the task is to design a 4-bit ripple carry adder-subtractor.
‘a’ and ‘b’ are the two 4-bit unsigned numbers that are to be added.
‘cin’ is not an input carry like in the previous question but it is a mode selection bit.
If cin=0 then ‘a’ and ‘b’ should be added (a+b).
If cin=1 then ‘b’ should be subtracted from ‘a’ (a-b).
‘sum’ is the 4-bit unsigned addition or subtraction output and
‘cout’ is the single-bit output carry.

In general, there are 2 ways to subtract a and b. One is the normal borrow method which cannot be used here because we have to use a 4 bit ripple carry adder for subtraction also. The 2nd method is to first take 2’s complement of b that is inverse all the bits of b and add 1 to get 2’s complement of b and then add a and the 2’s complement of b which will finally result in a-b.

Method for finding 2’s complement of b:

First we can inverse the bits of b by performing a XOR operation of each of the bits of b with cin=1. This is known as the 1’s complement of b. Now to convert 1’s complement of b to 2’s complement of b , we need to add 1. So instead of adding 1 here , i will give the 4 bit adder a carry of 1 to satisfy the 2’s complement of b.
So finally **a - b = a + (2’s complement of b)**
= a + (1’s complement of b + 1)
= a + (1’s complement of b) + (carry = 1)

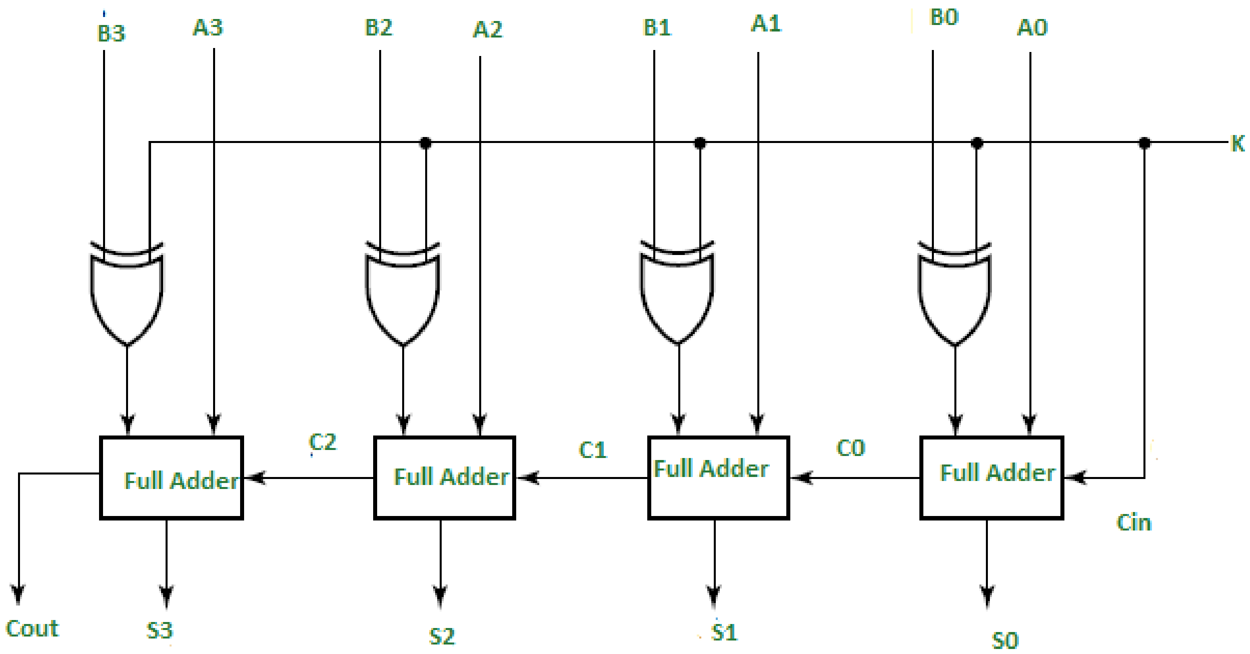
When cin = 0

Input will be ‘a’ , ‘b’ , ‘cin=0’

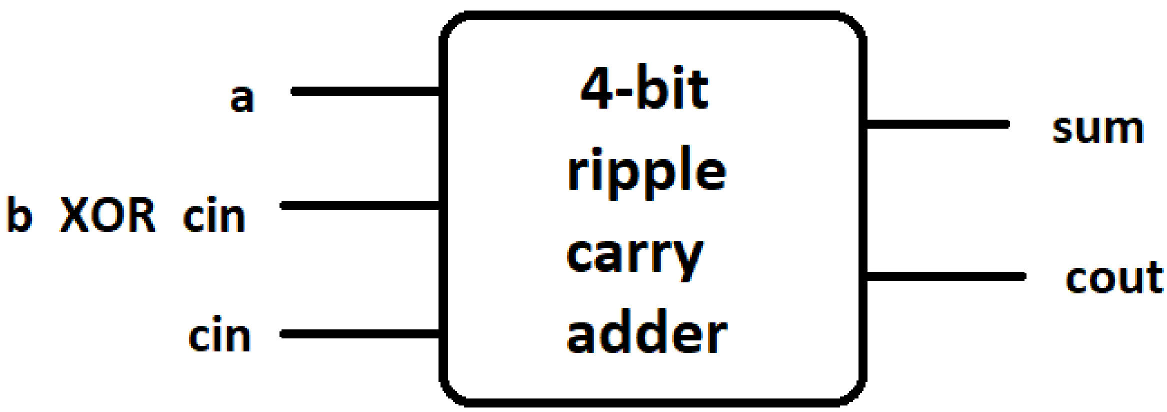
When cin = 1

Input will be ‘a’ , ‘1’s complement of b’, ‘cin=1’

In this way we can use the 4 bit ripple carry adder for both the addition and subtraction of two 4 bit numbers known as a 4-bit ripple carry adder-subtractor, which is a single function for calculating both the addition and the subtraction.



The above diagram is using 4 1-bit full adder. But we have to use a 4-bit ripple adder which i am simply showing by a box below. The internal implementation is clearly shown in the above diagram.



Analysis of the two cases possible:

Case-1:

cin = 0 So b XOR cin = b itself

So final output = a+b

Case-2:

cin = 1 So b XOR cin = 1’s complement of b

And by adding 1’s complement with the carry 1 present in cin, we finally get

= a + 1’s complement of b + 1

= a + 2’s complement of b

= a - b