

1 Working of our code

Our code implements addition-subtraction in 2's compliment representation. The logic of our code is:

- If the input opcode is 0, then we do the normal 8-bit addition.
- If the opcode is 1, then we add the 2's compliment of b to a.

Now to add 2's compliment of b, we have to add 1 to not of b. So, here is the logic to achieve that:

- We provide the initial carry in as the **opcode itself** because 0 opcode indicates addition and in that case, the carry-in has to be 0. Similarly, 1 opcode indicates subtraction, in that case we have to add 1 to the compliment of b, so that 1 acts as the carry-in.
- Now, to get the not of a single bit of b, the idea is simple- by the results of addition, (Single bit of b)+1 gives compliment of the single bit and the addition of 0 gives the single bit itself. So, we to get the required result, instead of adding b to a in the 1-bit adder, we add (b+opcode) which gives the result accordingly.
- Also, in the 8-bit adder, we provide the first carry-in as the opcode, which again does the required addition or subtraction.