1. Logic of 8-bit adder-subtracter

We use the concept of 2's compliment addition to construct it. To obtain the 2's compliment, we add 1 to the 1's compliment of the number. The 1's compliment is obtained by negating each bit of the number.
Now, the adder subtractor works in the following way:

- For normal addition, (opcode=0), it does a+b.
- For subtraction, (opcode=1), it performs a+(-b) where -b is the 2's compliment of b, therefore giving a-b.

2. 1-bit adder-subtracter

To implement the above, we have to calculate 1-bit addition-subtraction in the 2's compliment system. We already know, the normal 1-bit addition is XOR operation. The important part is to get the result after calculating the 2's
compliment form:

- If the opcode is 0, you need to use b as it is.
- If the opcode is 1 you need to flip b and add (basically negation of b).

By this logic, we need to simply XOR the normal 1-bit addition with opcode. Similarly, for the 1-bit carry-out, replace b with (b XOR opcode).

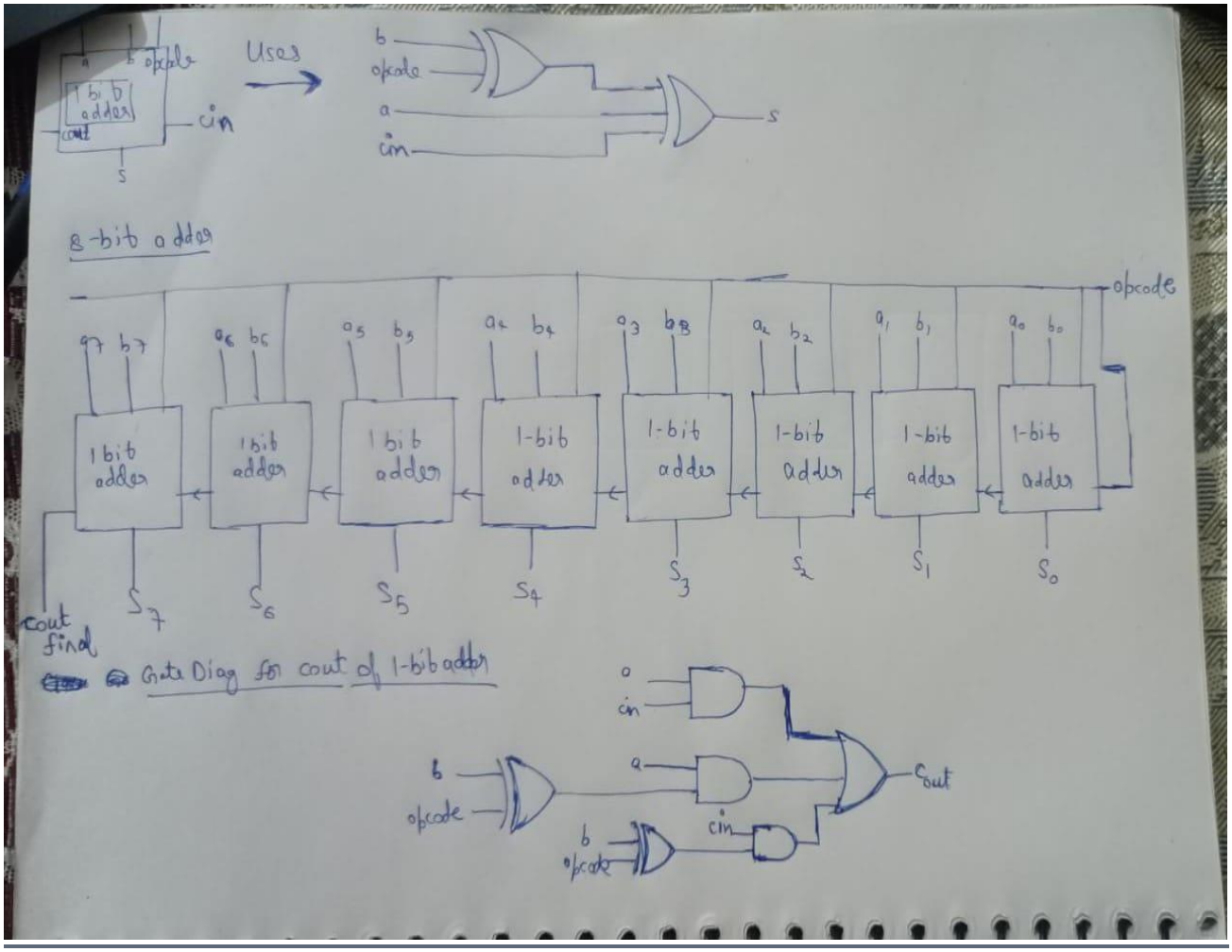
3. 8-bit result using 1-bit adder-subtractor

We simply use the result obtained by 1-bit addition on each bit of the input number. The carry-out of the current addition acts as carry-in of the next addition until the final bit. Now the first carry-in is the opcode itself as for addition, we don't need carry-in(opcode=0) and for subtraction, we need to add 1 to the 1's compliment which we have done in the form of carry-in itself.
Hence, providing opcode as the initial carry-in takes care of both the cases.

4. Checking overflow

If the last 2 carry out are same, then there is no overflow, otherwise we have overflow. Again, by the same logic, if 0 denotes no overflow and 1 denotes overflow, then the value of variable overflow is the XOR of last 2 carry-outs.

5.Circuit Diagram


a
b
opcode
1 bit adder
cout
cin
S
Uses
b
opcode
a
cin
S
8-bit adder
opcode
a7 b7
a6 b6
a5 b5
a4 b4
a3 b3
a2 b2
a1 b1
a0 b0
1 bit adder
1 bit adder
1 bit adder
1-bit adder
1-bit adder
1-bit adder
1-bit adder
1-bit adder
Cout final
S7
S6
S5
S4
S3
S2
S1
S0
Gate Diag for cout of 1-bit adder
0
cin
b
opcode
a
cin
b
opcode
Cout