CCE1013 - Computer Logic 1

Giorgio Grigolo
B.Sc. Mathematics and Computer Science

Year 1, Semester 1 - 2021/22

Contents

Introduction

The following report indicates the operation of a 4-instruction arithmetic logic unit. The instructions are `ADD, ADDC, SUB, SUBB` represented by opcodes `00, 01, 10, 11` respectively. It is to be noted that this implementation of the above ALU outputs the C_o as the MSB by turning on the **red LED** (left) and outputs the Y as the LSB by turning on the **green LED** (right).

Proof of Correct Operation

2.1 Opcode `00 - ADD`

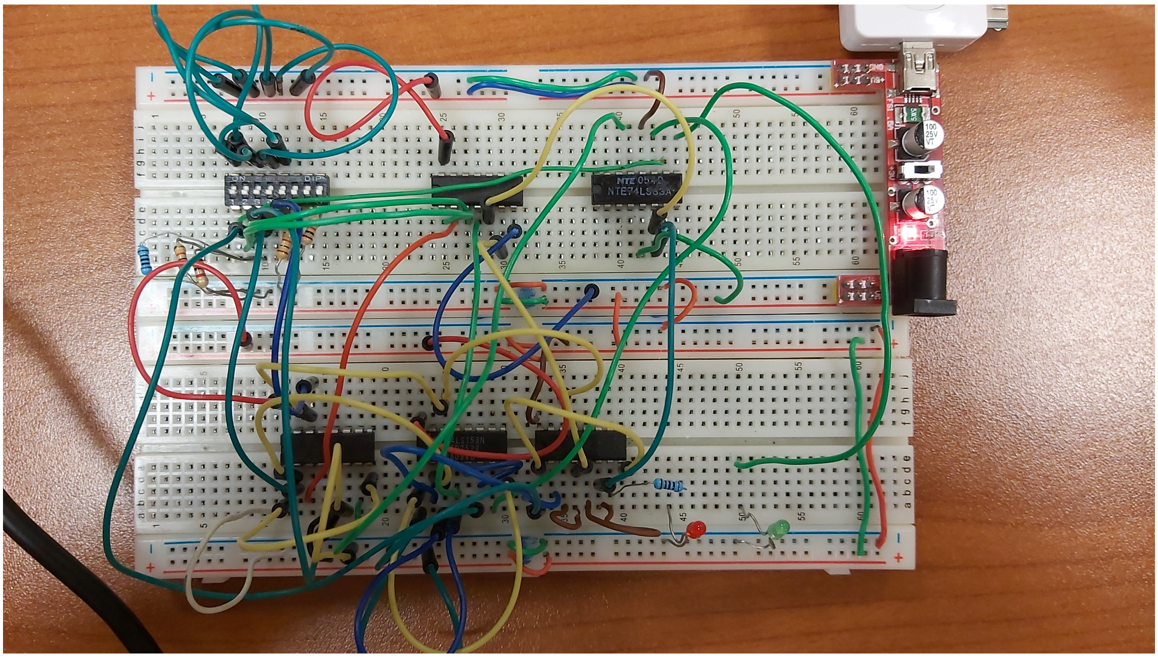
With this opcode selected, the ALU performs the addition of bits A and B, whilst ignoring the carry in represented by C_i.

$$A + B + 0 = C_o Y$$

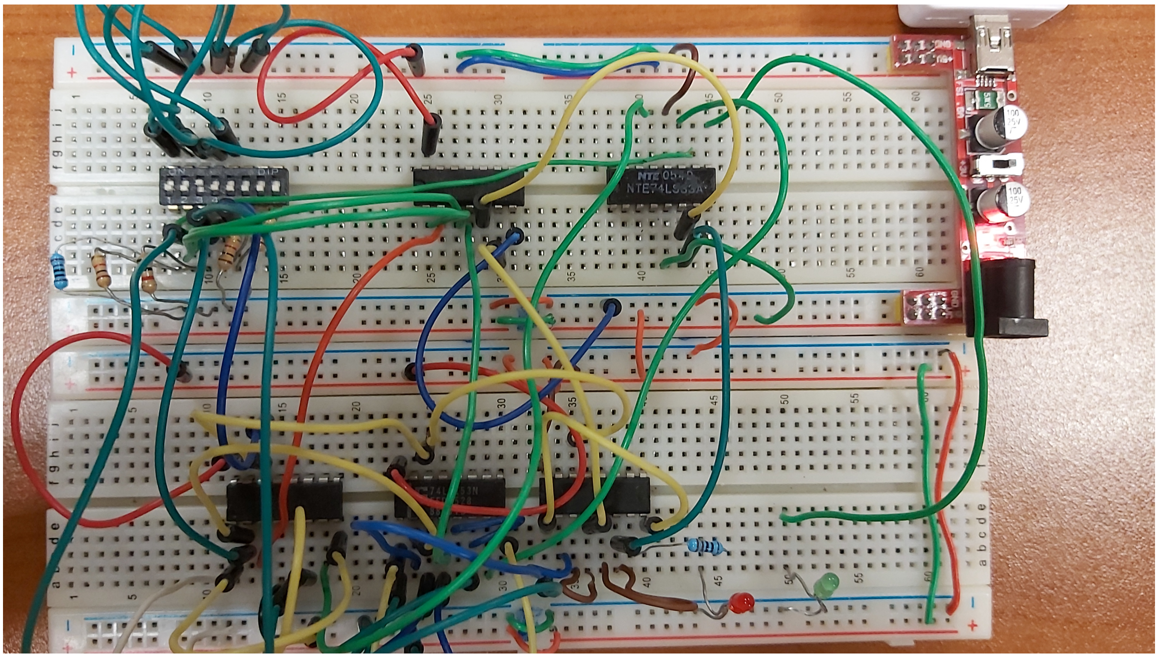
2.1.1 Truth table

A	B	C_i	Op_1	Op_2	C_o	Y
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	1
0	1	1	0	0	0	1
1	0	0	0	0	0	1
1	0	1	0	0	0	1
1	1	0	0	0	1	0
1	1	1	0	0	1	0

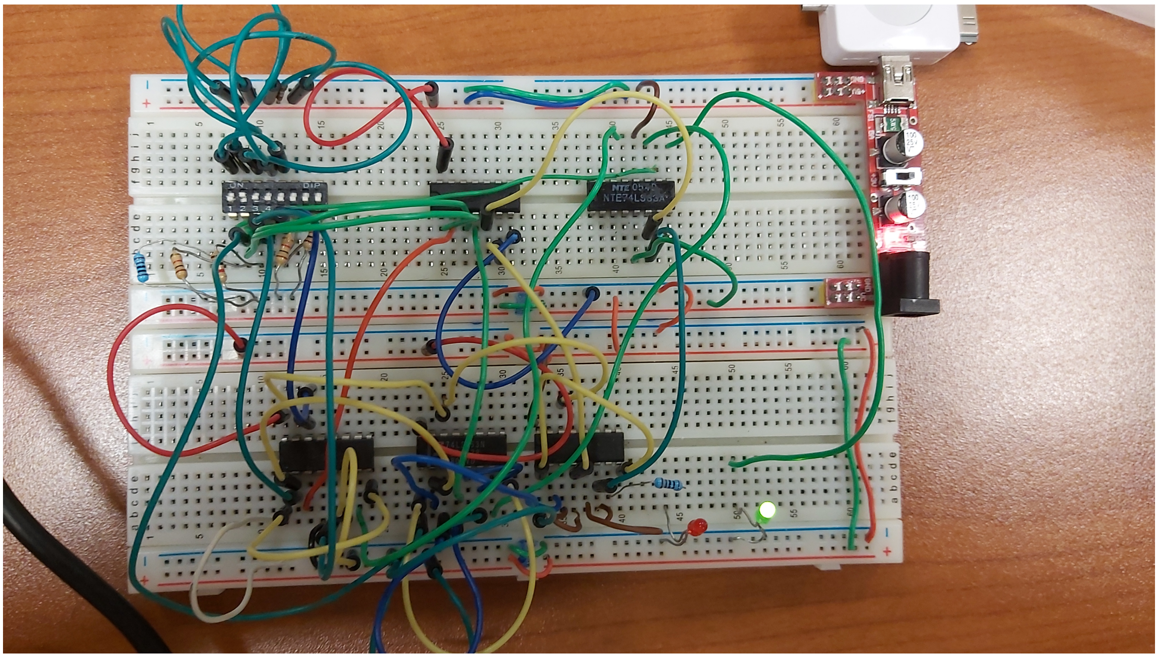
2.1.2 Pictures



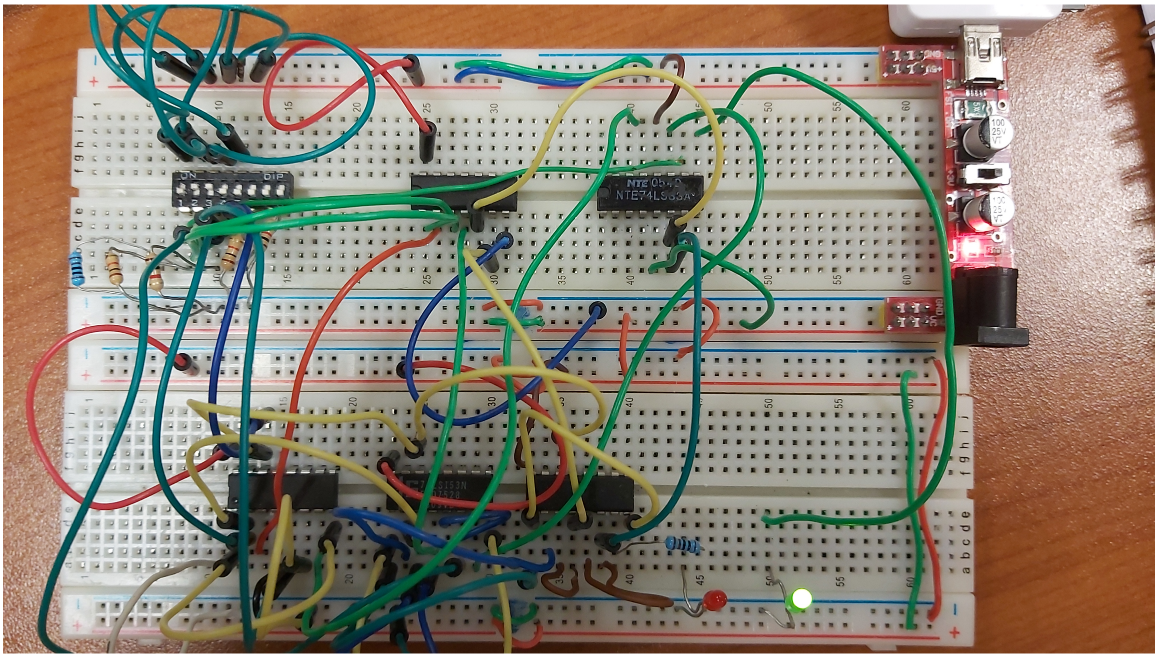
All inputs are off, and thus no LED is on.



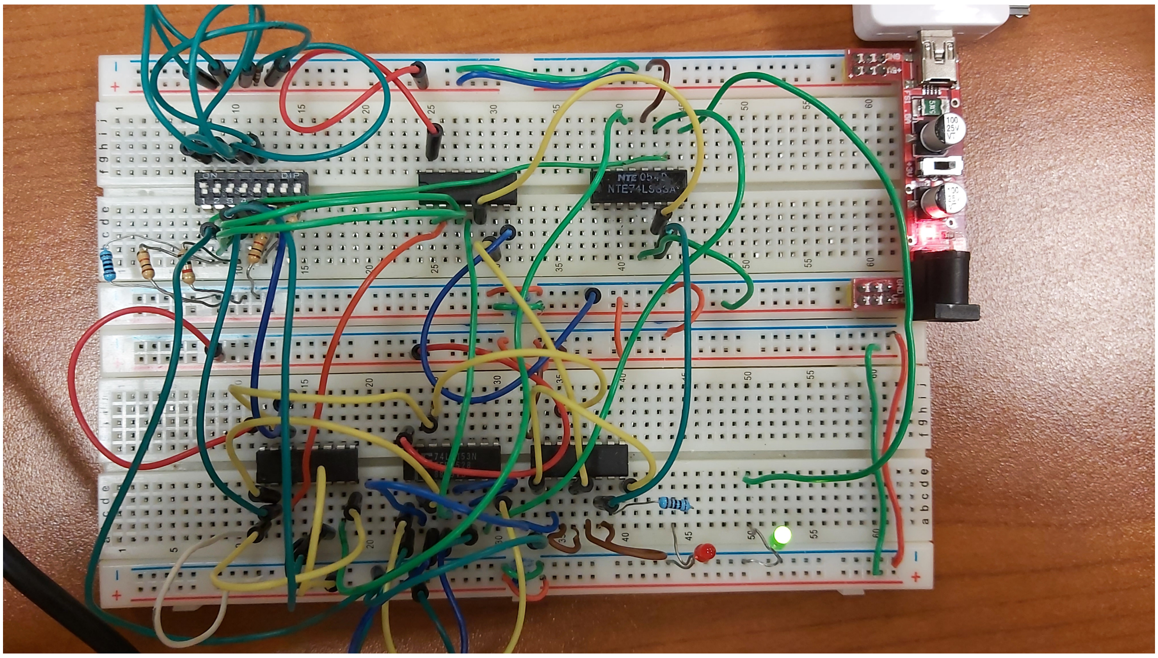
$ABC_i = 001$ and so $C_oY = 00$.



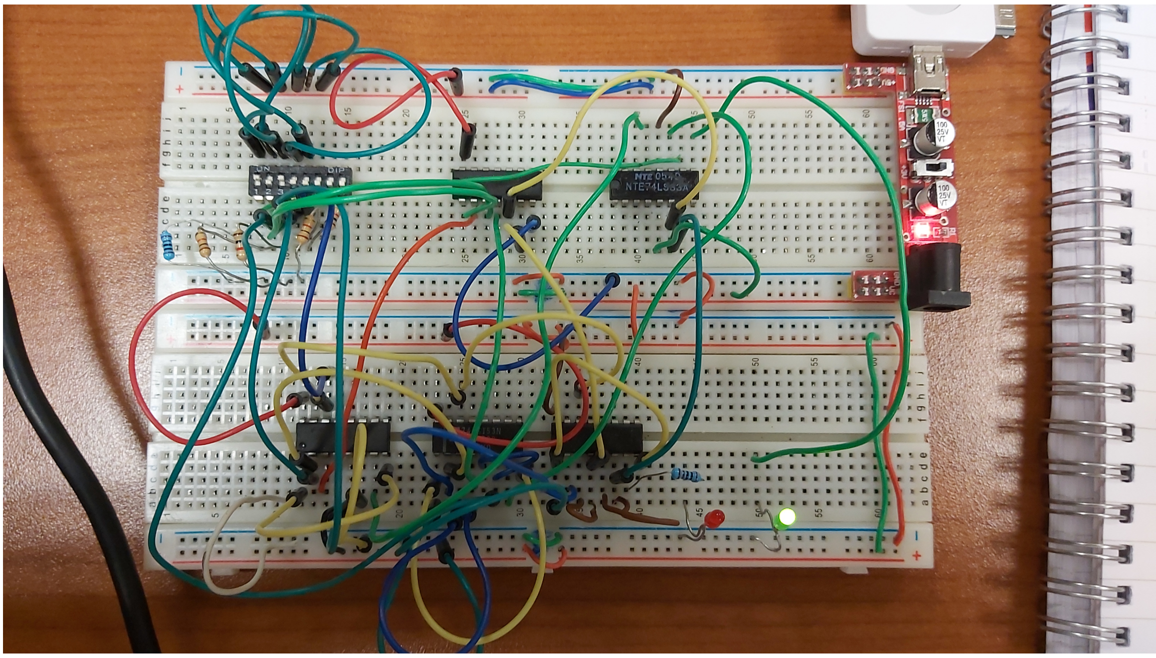
$ABC_i = 010$ and so $C_oY = 01$.



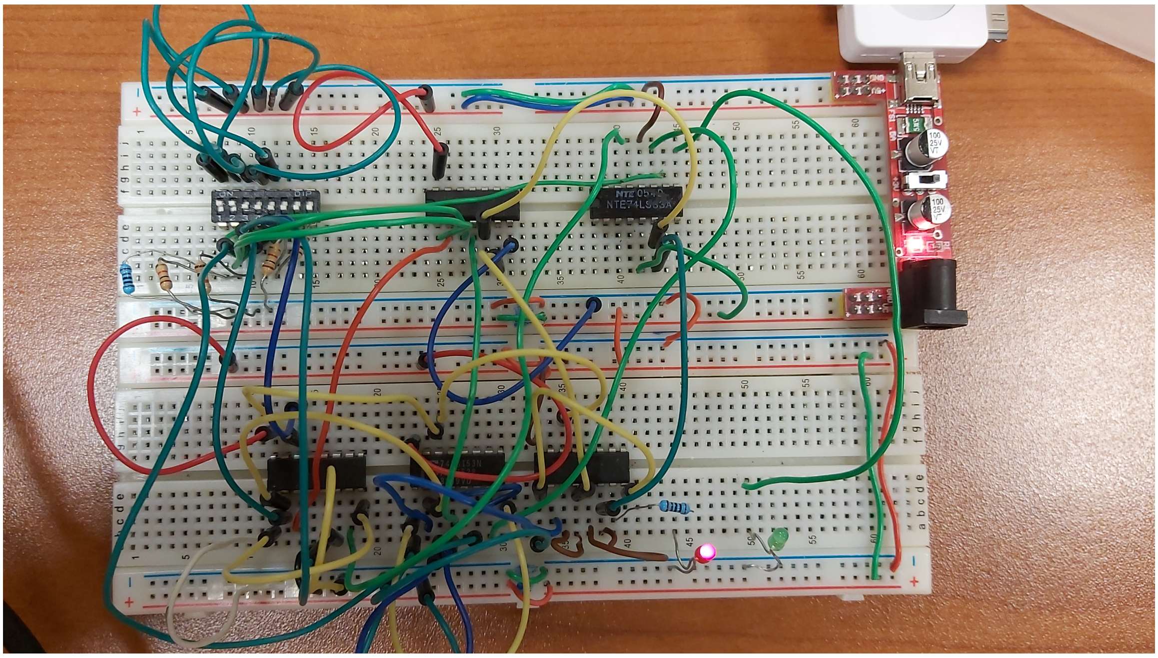
$ABC_i = 011$ and so $C_oY = 01$.



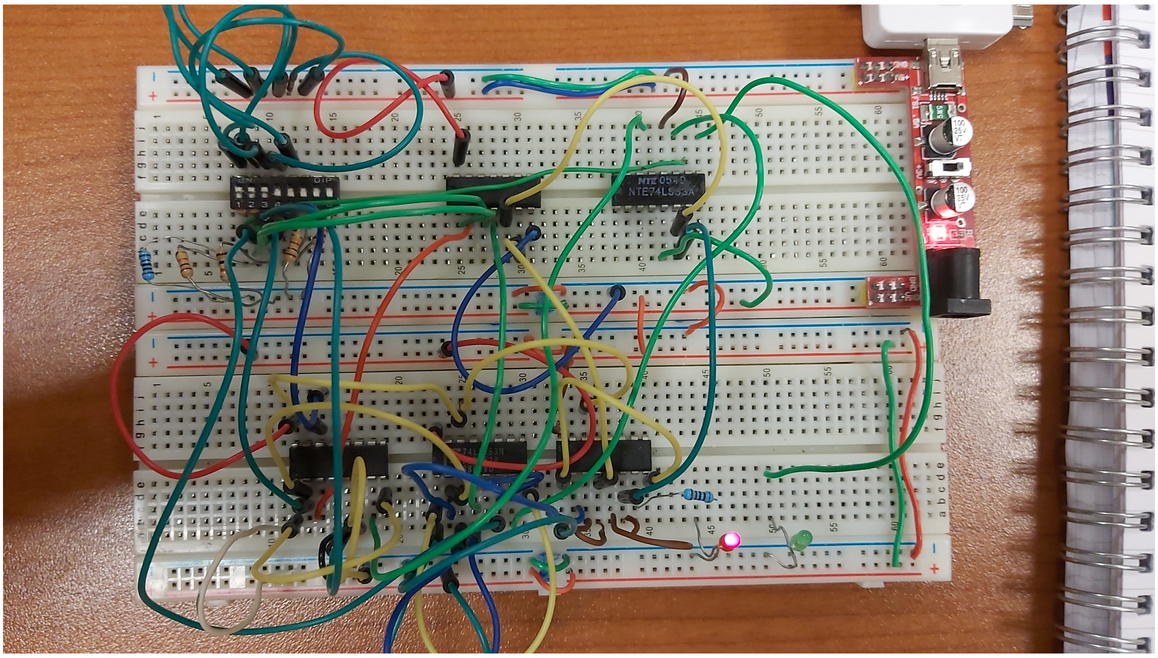
$ABC_i = 100$ and so $C_oY = 01$.



$ABC_i = 101$ and so $C_oY = 01$.



$ABC_i = 110$ and so $C_oY = 10$.



$ABC_i = 111$ and so $C_oY = 10$.

2.2 Opcode `01 - ADDC`

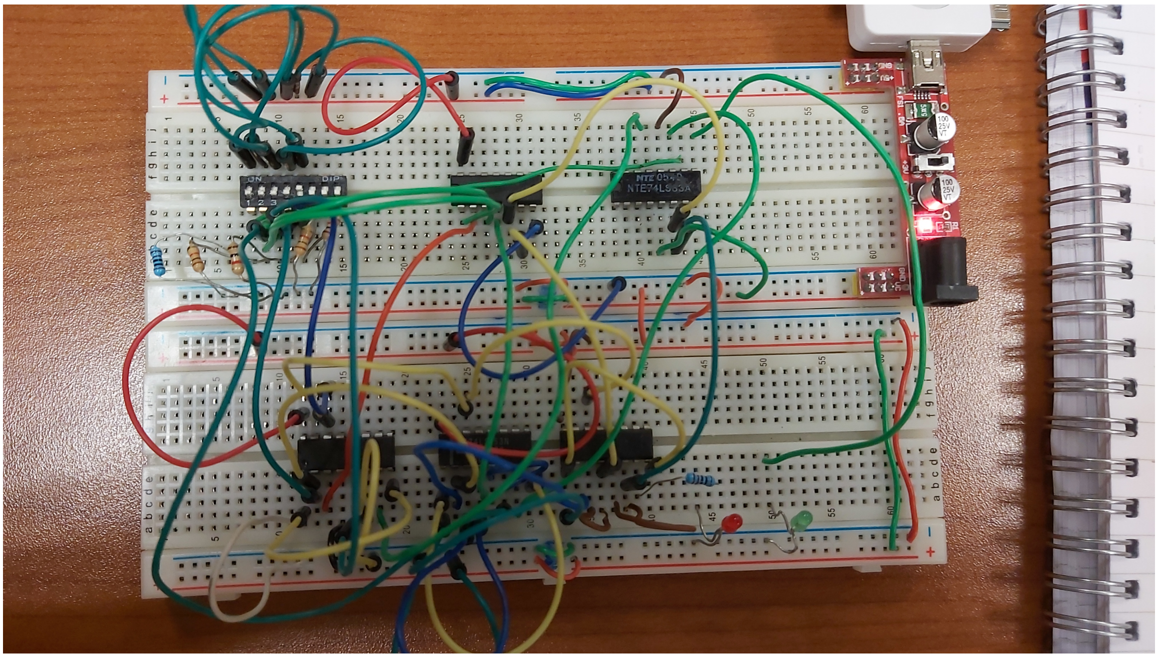
With this opcode selected, the ALU performs the addition of bits A and B but consequently adding the carry bit represented by C_i.

$$A + B + C_i = C_o Y$$

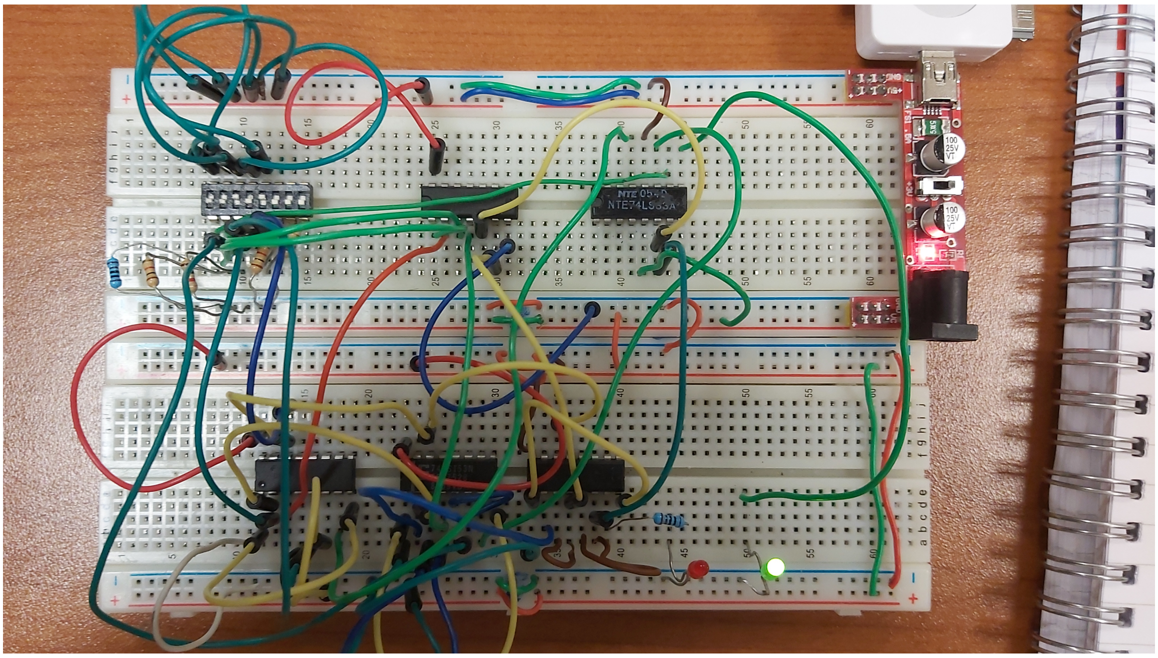
2.2.1 Truth table

A	B	C_i	Op_1	Op_2	C_o	Y
0	0	0	0	1	0	0
0	0	1	0	1	0	0
0	1	0	0	1	0	1
0	1	1	0	1	0	1
1	0	0	0	1	0	1
1	0	1	0	1	0	1
1	1	0	0	1	1	0
1	1	1	0	1	1	0

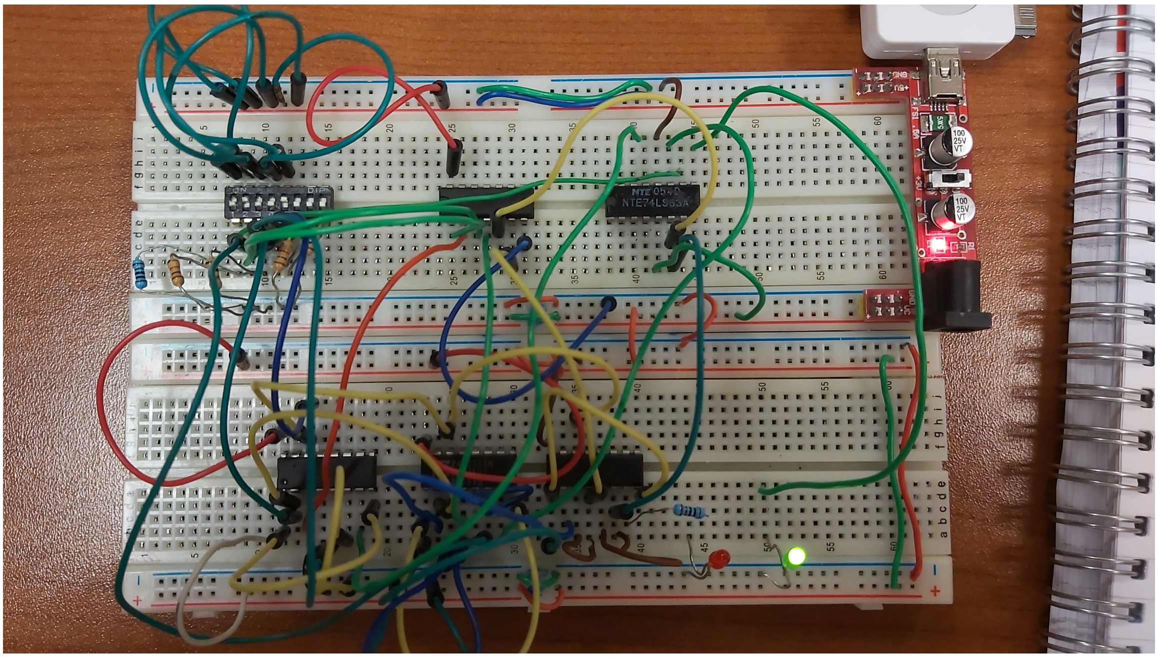
2.2.2 Pictures



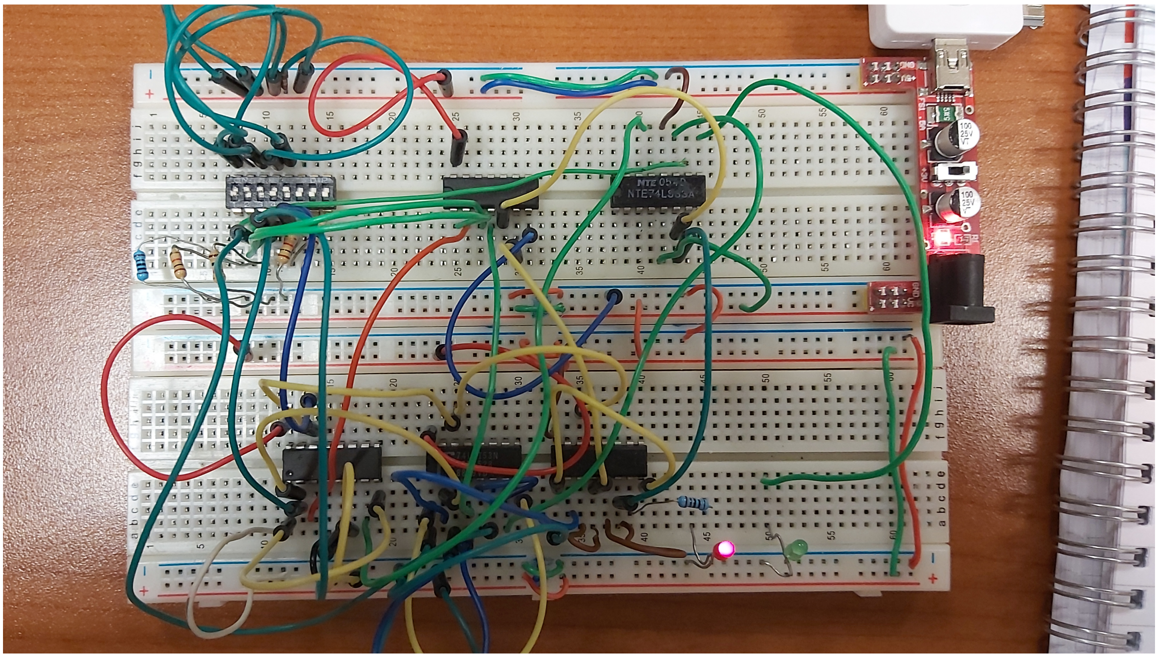
All inputs are off, and thus no LED is on.



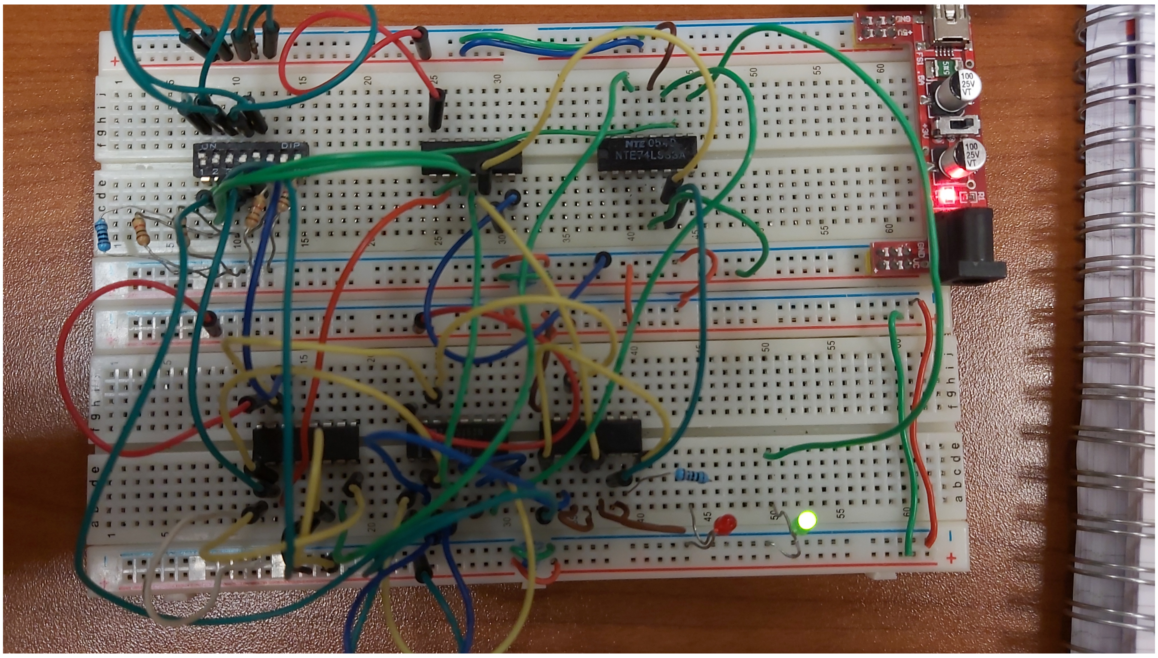
$ABC_i = 001$ and so $C_oY = 01$.



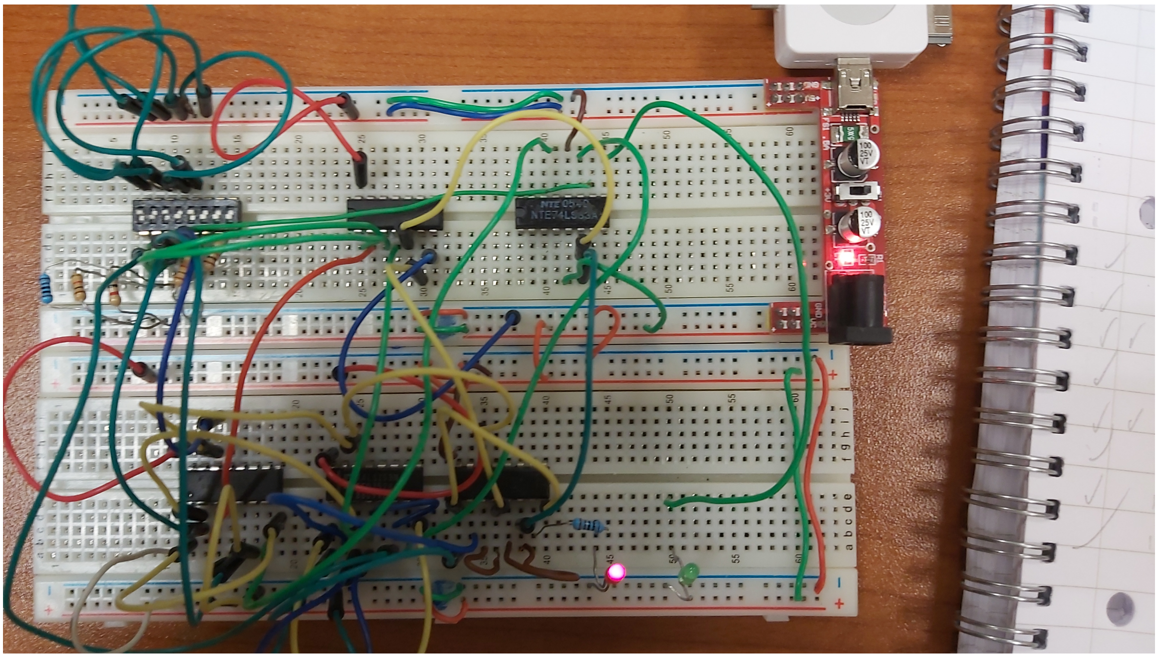
$ABC_i = 010$ and so $C_oY = 01$.



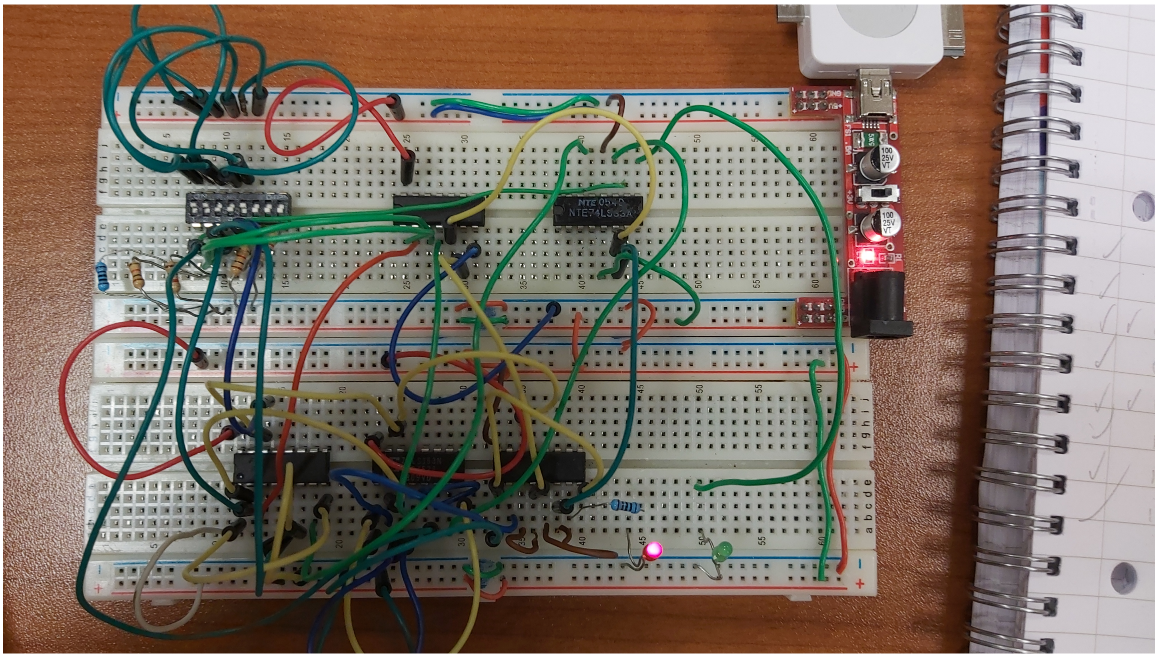
$ABC_i = 011$ and so $C_oY = 10$.



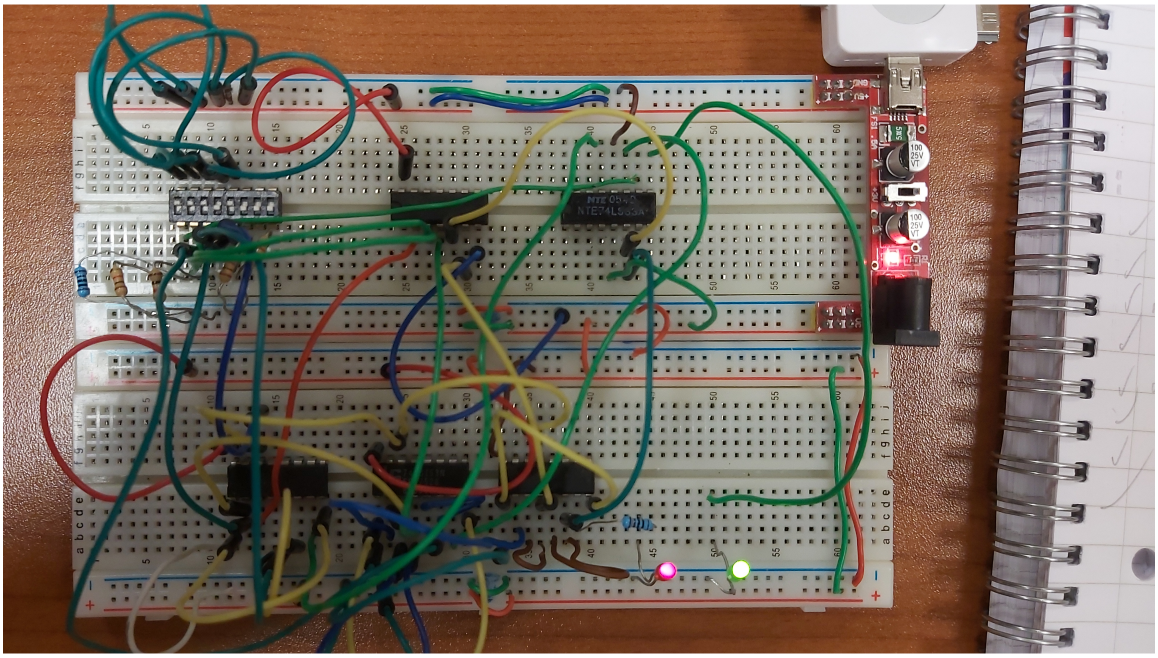
$ABC_i = 100$ and so $C_oY = 01$.



$ABC_i = 101$ and so $C_oY = 10$.



$ABC_i = 110$ and so $C_oY = 10$.



$ABC_i = 111$ and so $C_oY = 11$.

2.3 Opcode `10 - SUB`

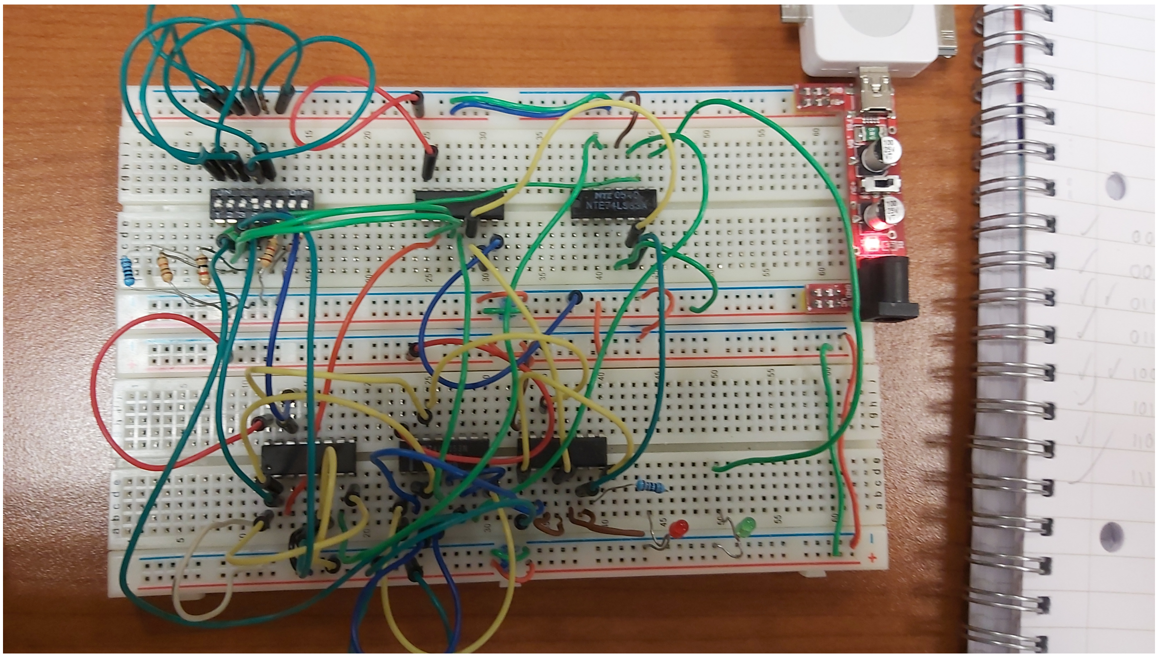
With this opcode selected, the ALU performs the subtraction of bits A and B, or rather, the addition of A, $\bar{B}$ and 1, whilst ignoring the borrow bit C_i. When a negative value is obtained, the resultant representation will be one in 2's complement, and so -1_{10} would be represented as $11_2 = -2_10+1_10$.

$$A + \bar{B} + 1 = C_o Y$$

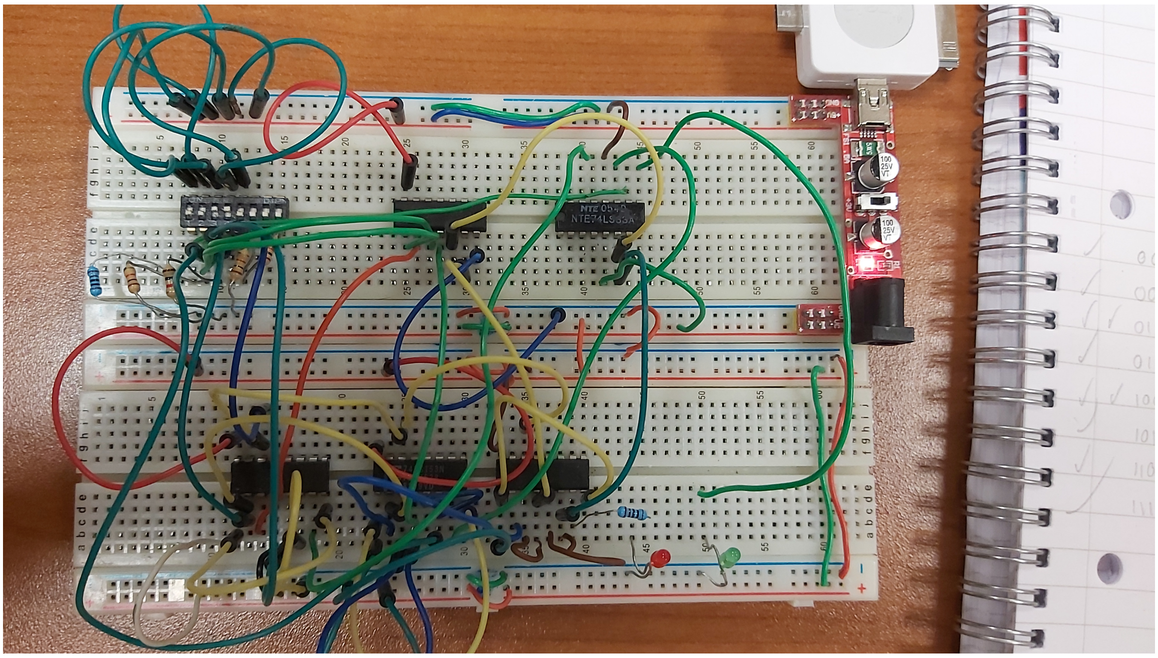
2.3.1 Truth table

A	B	C_i	Op_1	Op_2	C_o	Y
0	0	0	1	0	0	0
0	0	1	1	0	0	0
0	1	0	1	0	1	1
0	1	1	1	0	0	0
1	0	0	1	0	0	1
1	0	1	1	0	0	1
1	1	0	1	0	0	0
1	1	1	1	0	0	0

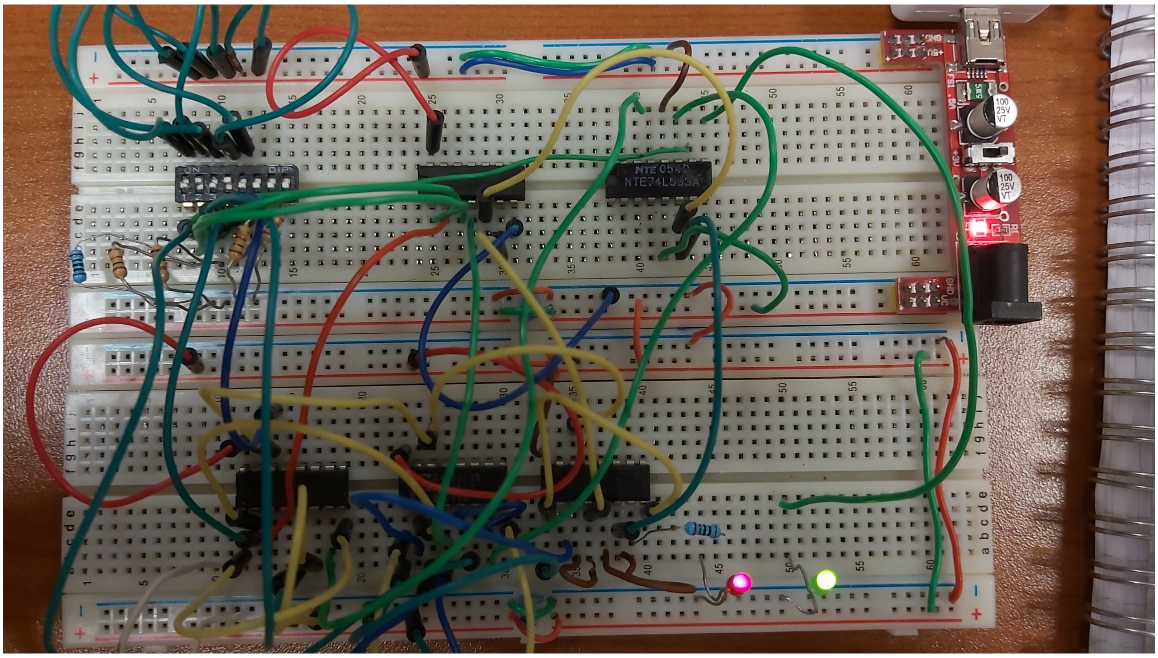
2.3.2 Pictures



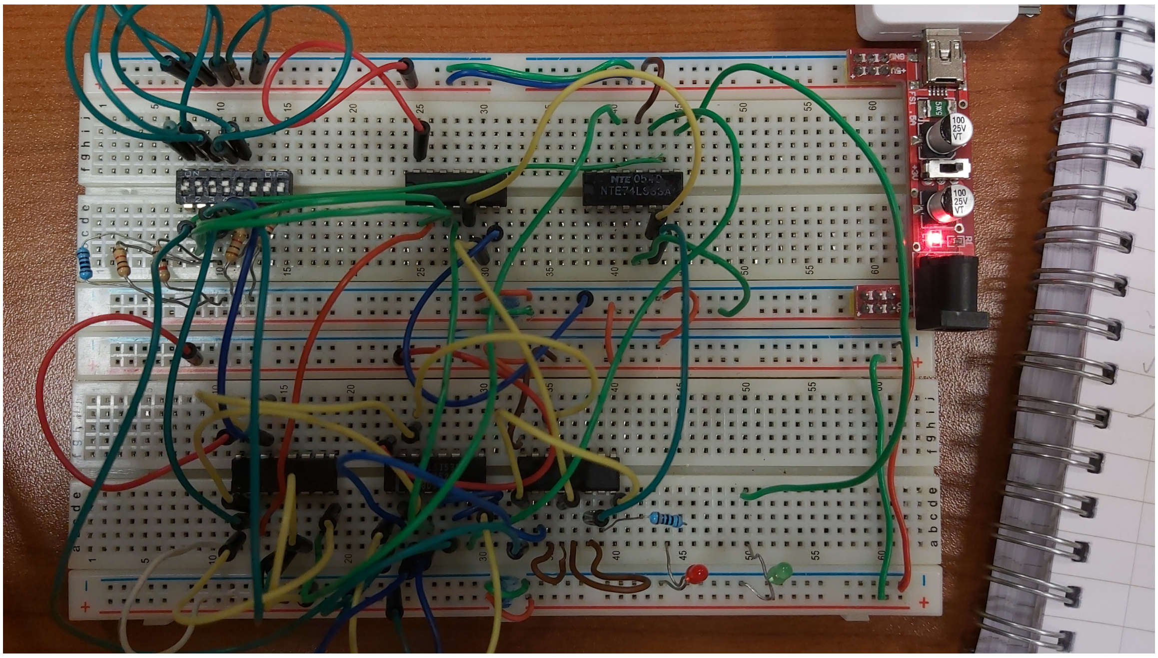
All inputs are off, and thus no LED is on.



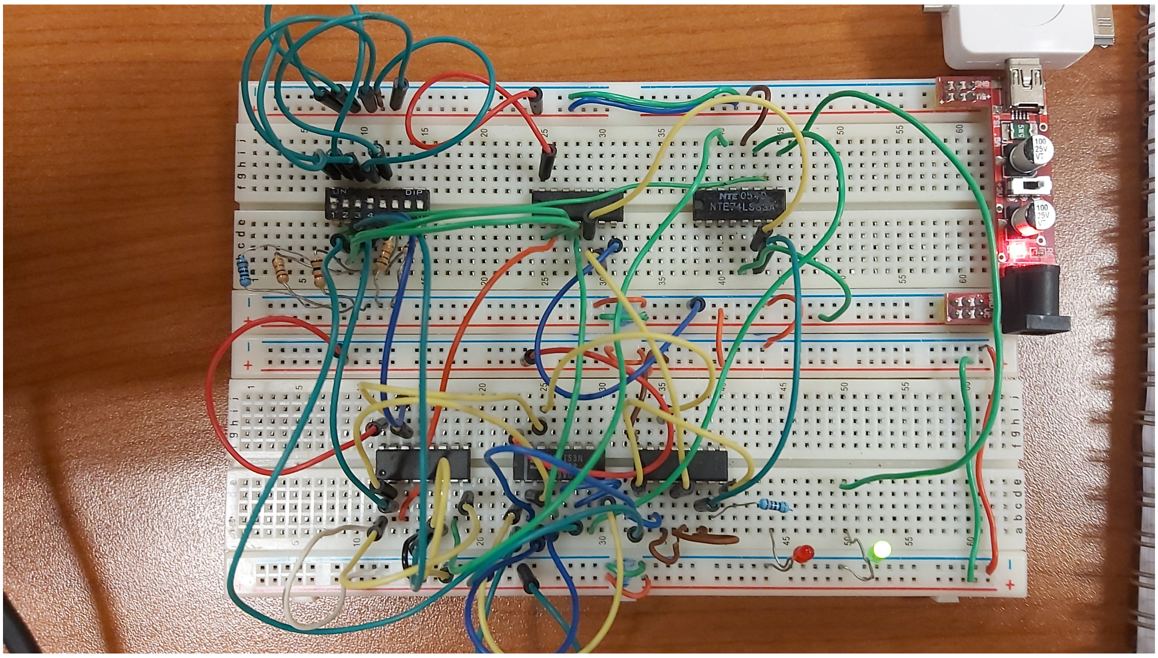
$ABC_i = 001$ and so $C_oY = 00$.



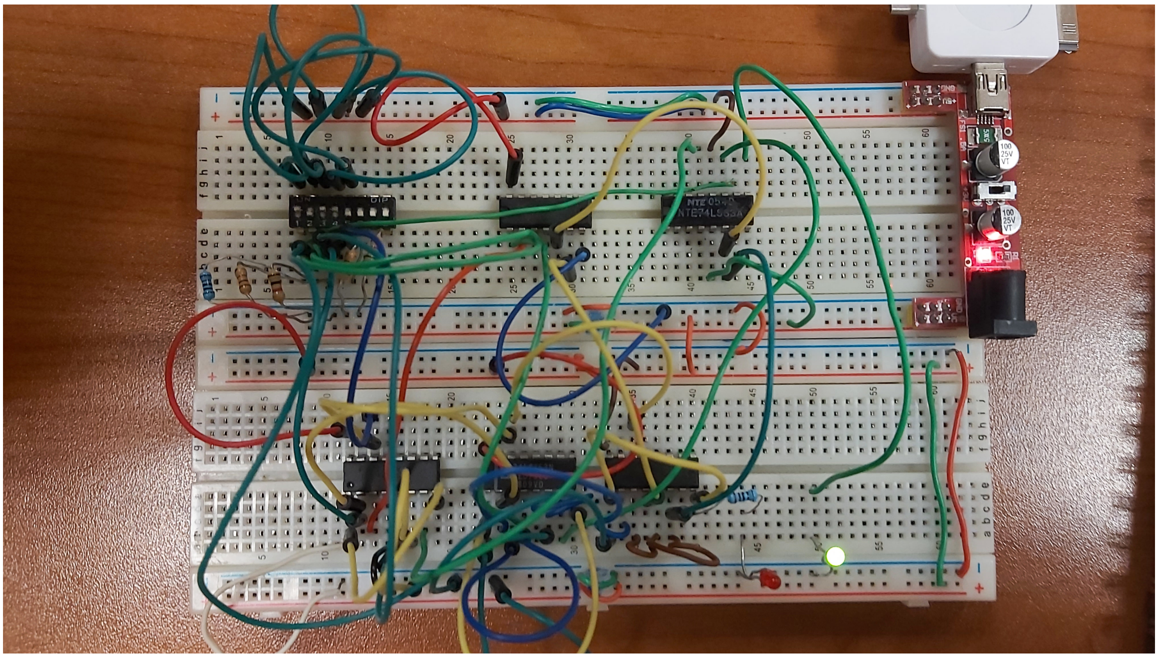
$ABC_i = 010$ and so $C_oY = 11$.



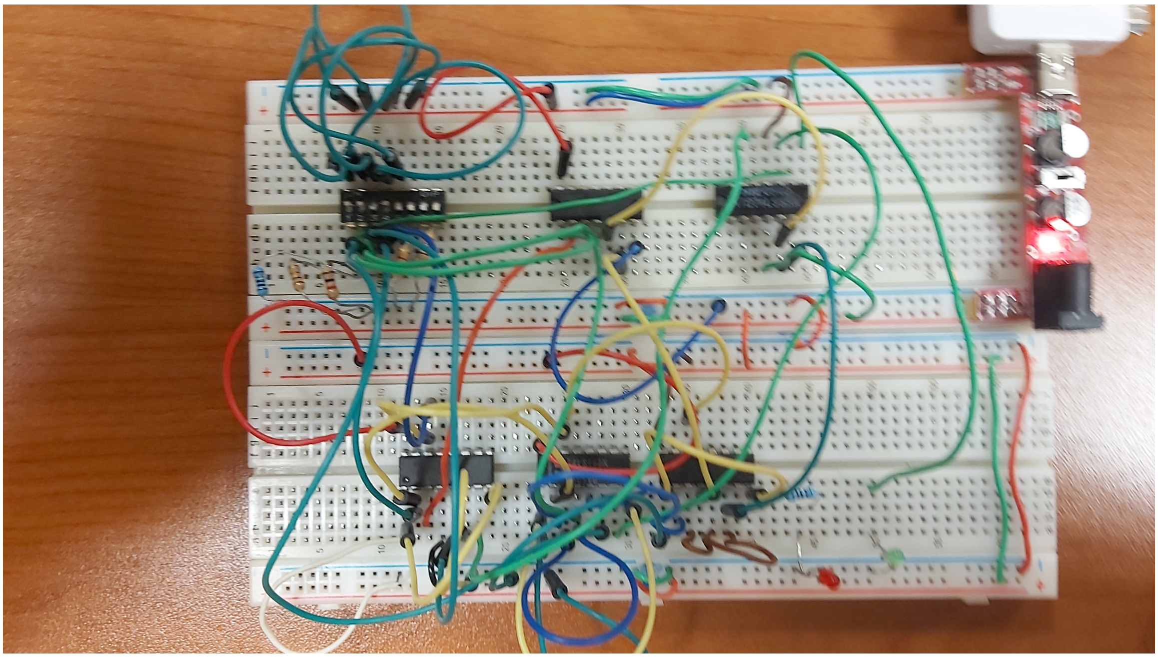
$ABC_i = 011$ and so $C_oY = 00$.



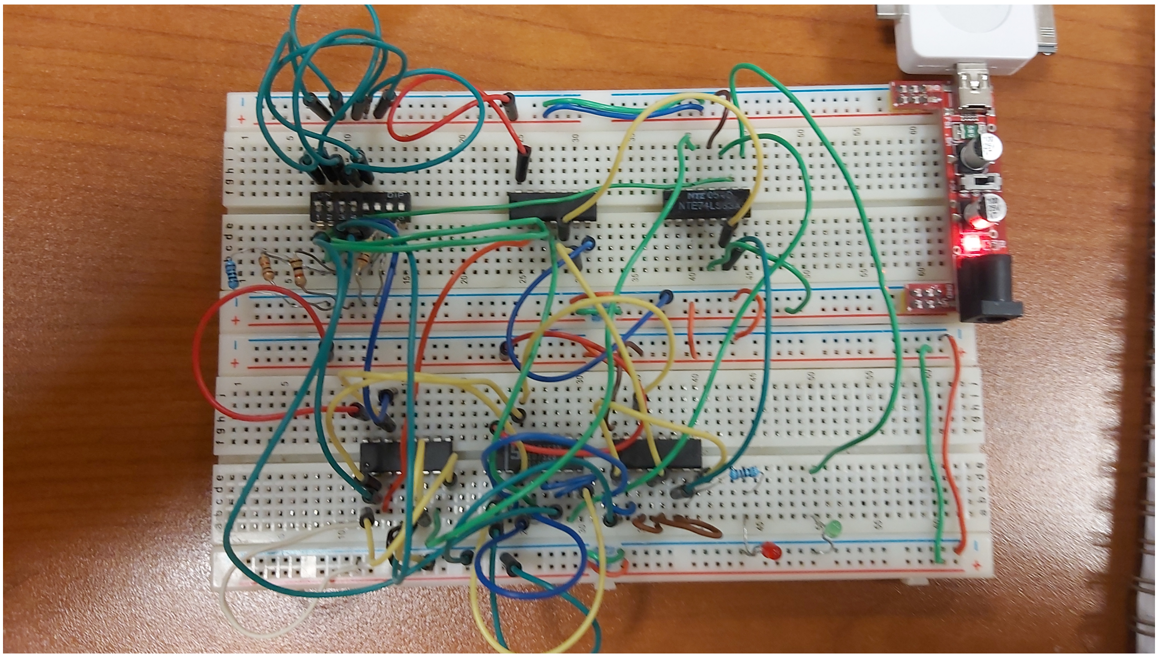
$ABC_i = 100$ and so $C_oY = 01$.



$ABC_i = 101$ and so $C_oY = 01$.



$ABC_i = 110$ and so $C_oY = 00$.



$ABC_i = 111$ and so $C_oY = 00$.

2.4 Opcode **11** – **SUBB**

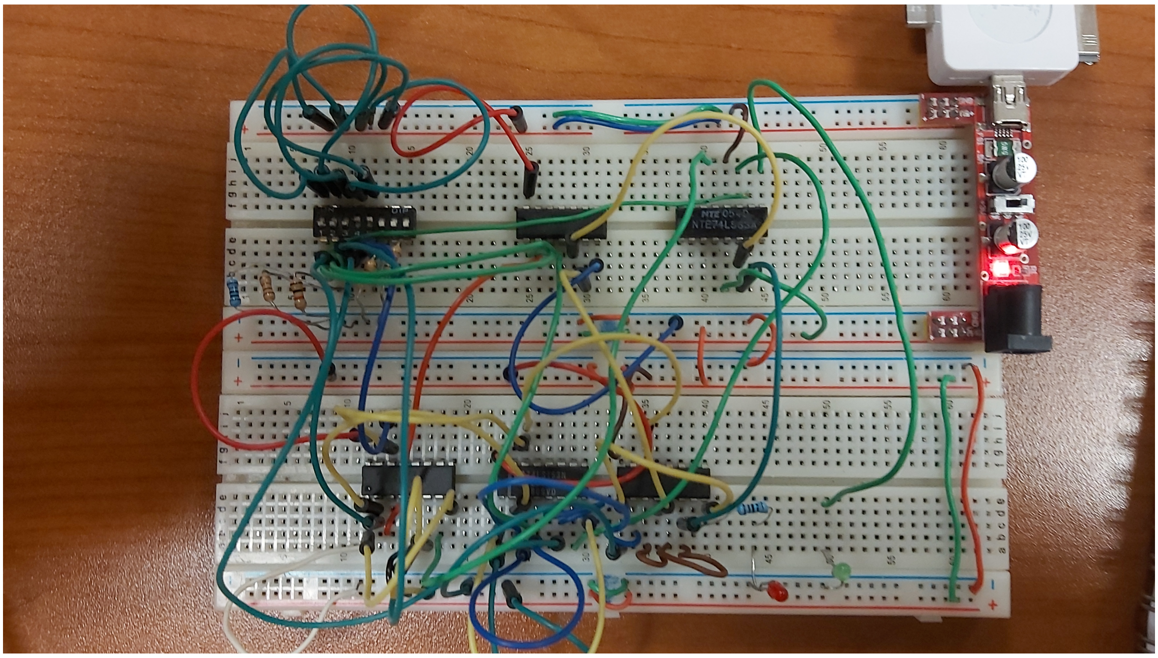
With this opcode selected, the ALU performs the subtraction of bits A and B, or rather, the addition of A and $\bar{B}$, whilst ignoring the borrow bit C_i. When a negative value is obtained, the resultant representation will be one in 2's complement, and so -1_{10} would be represented as $11_2 = -2_{10} + 1_{10}$.

$$A + \bar{B} + \bar{C_i} = C_o Y$$

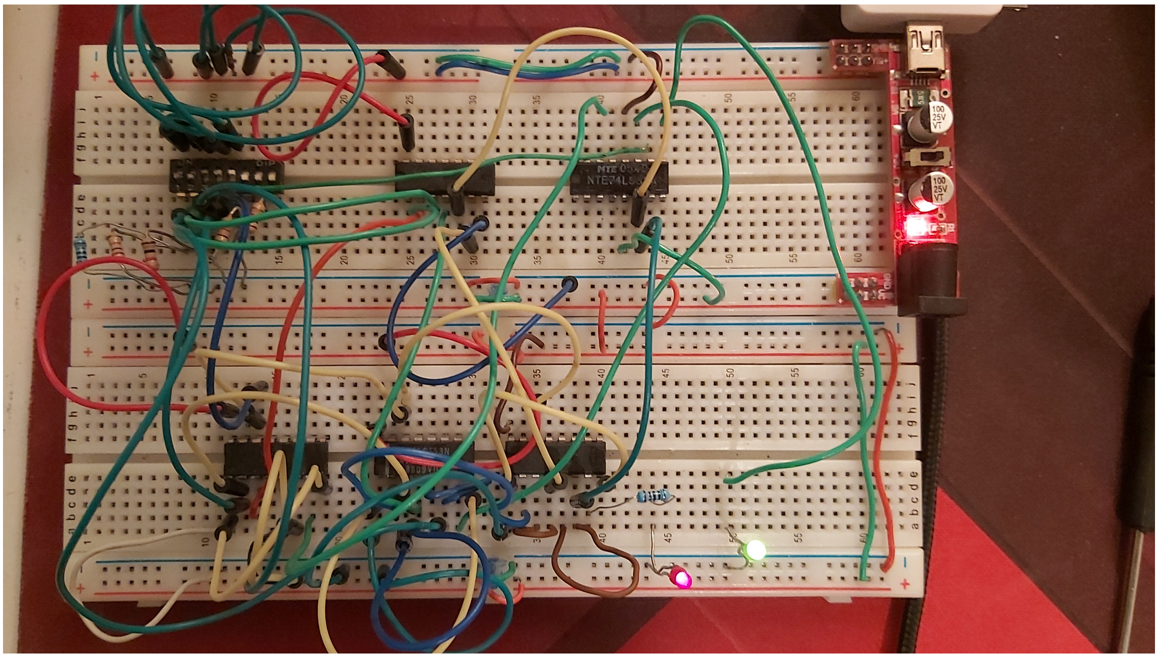
2.4.1 Truth table

A	B	C_i	Op_1	Op_2	C_o	Y
0	0	0	1	1	0	0
0	0	1	1	1	1	1
0	1	0	1	1	1	1
0	1	1	1	1	1	0
1	0	0	1	1	0	1
1	0	1	1	1	0	0
1	1	0	1	1	0	0
1	1	1	1	1	1	1

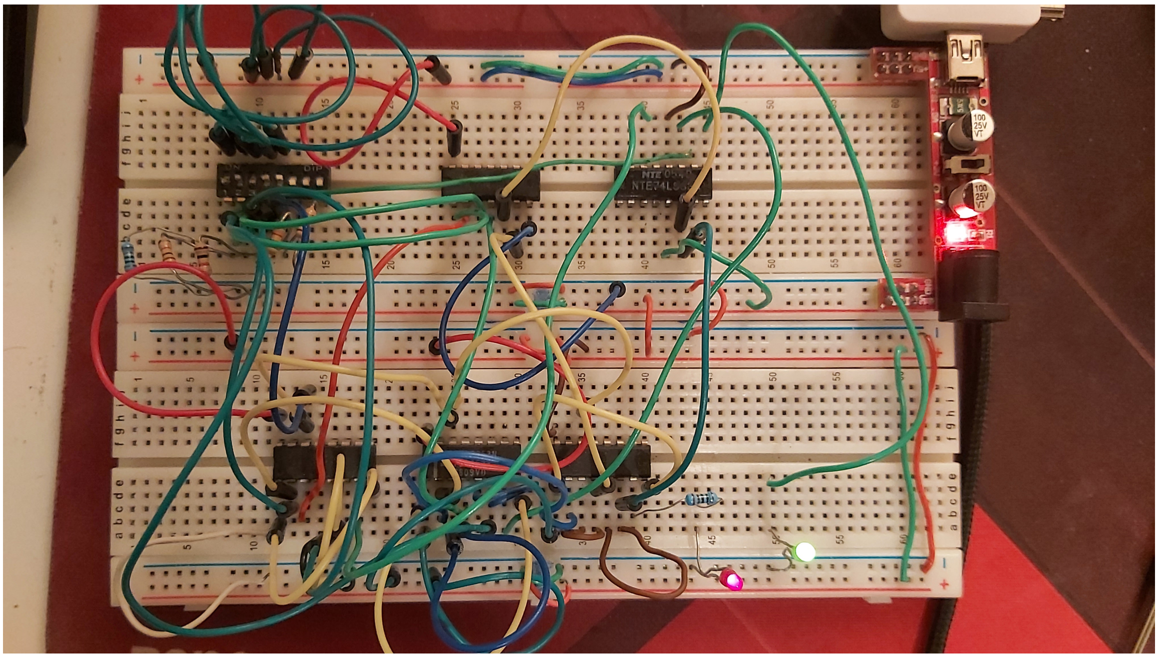
2.4.2 Pictures



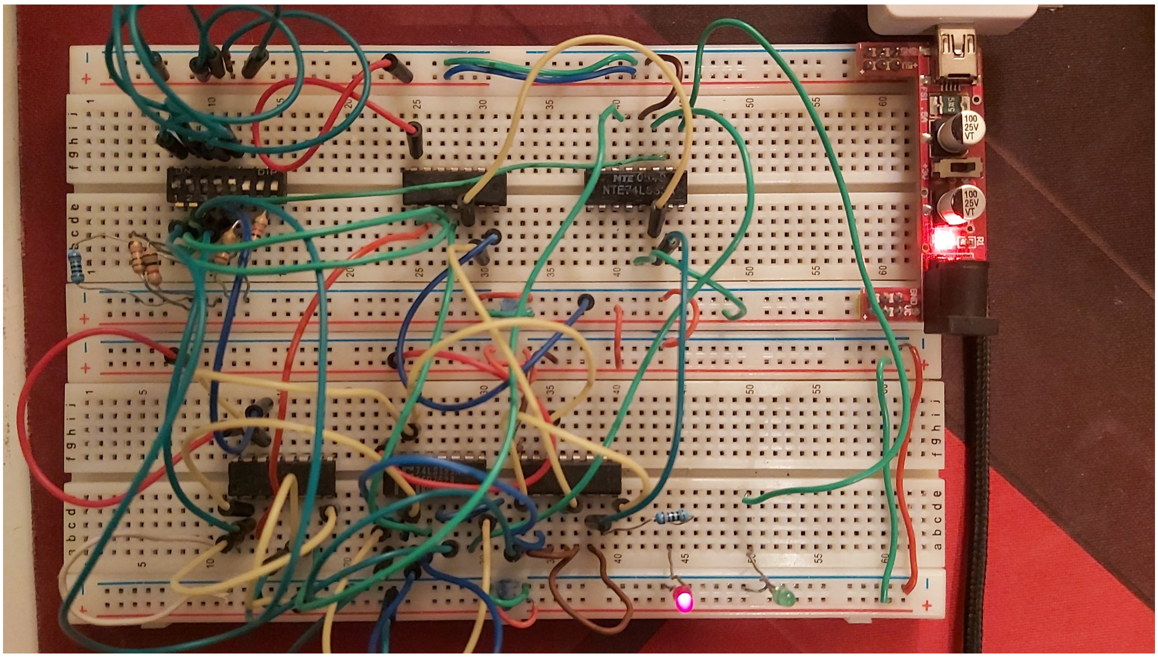
All inputs are off, and thus no LED is on.



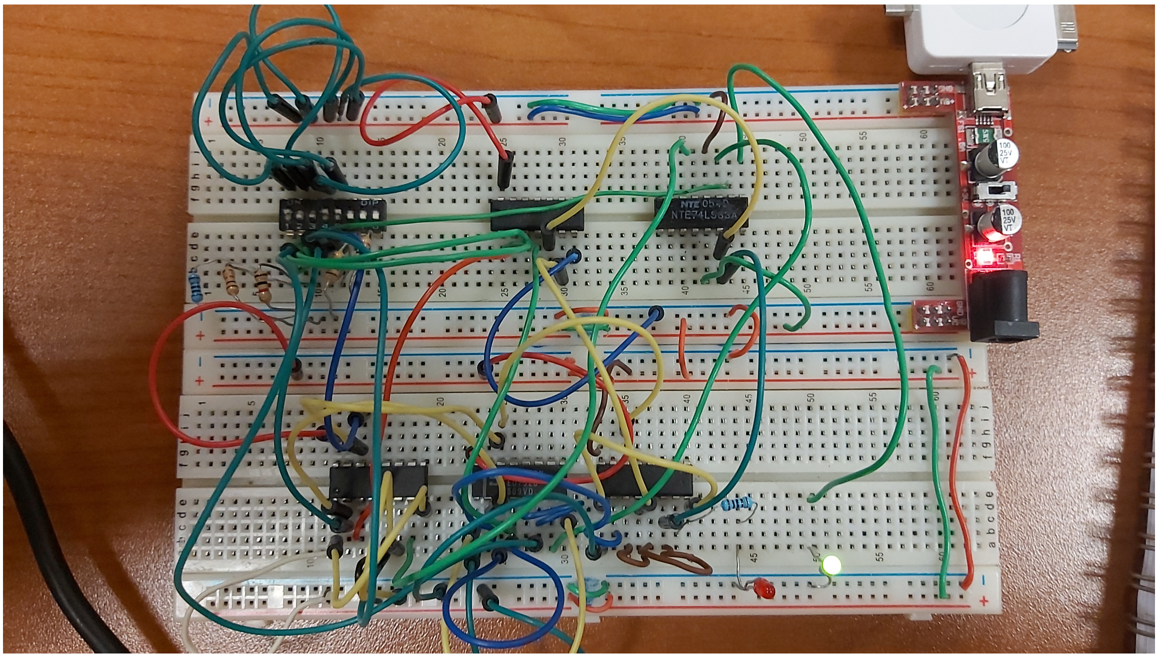
$ABC_i = 001$ and so $C_oY = 11$.



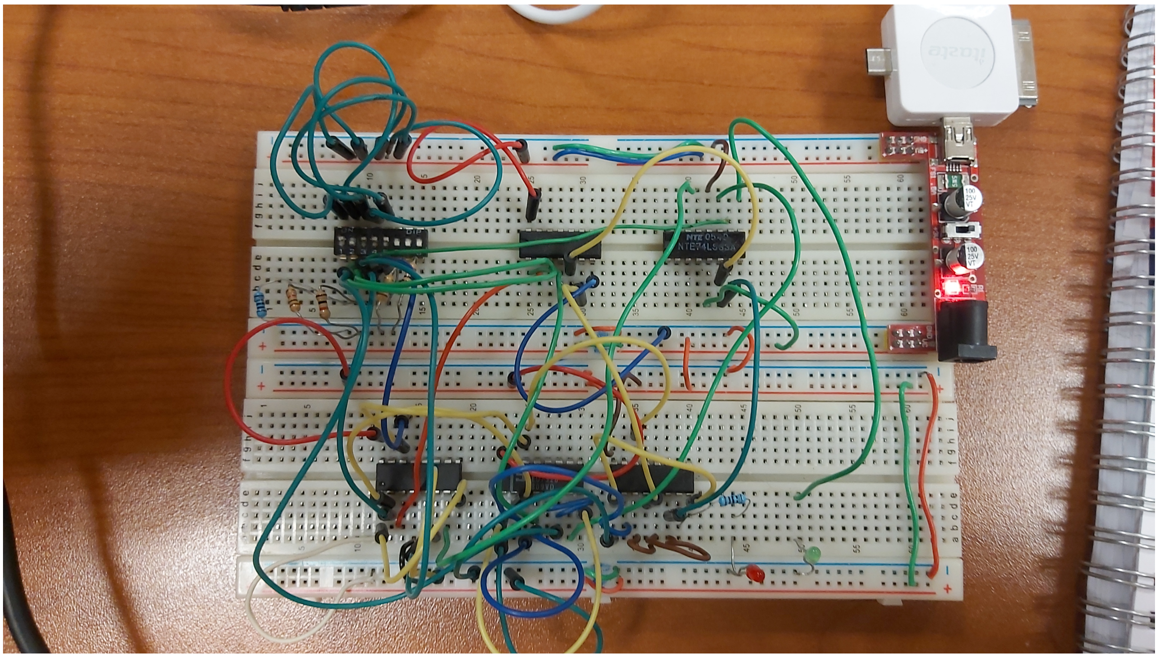
$ABC_i = 010$ and so $C_oY = 11$.



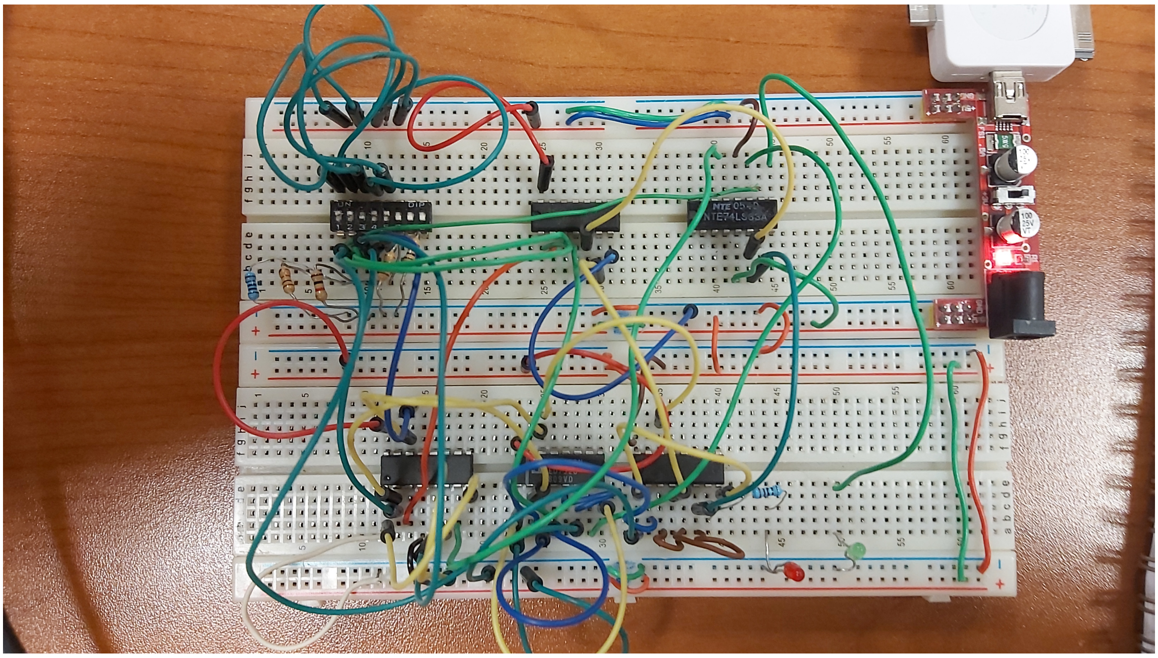
$ABC_i = 011$ and so $C_oY = 10$.



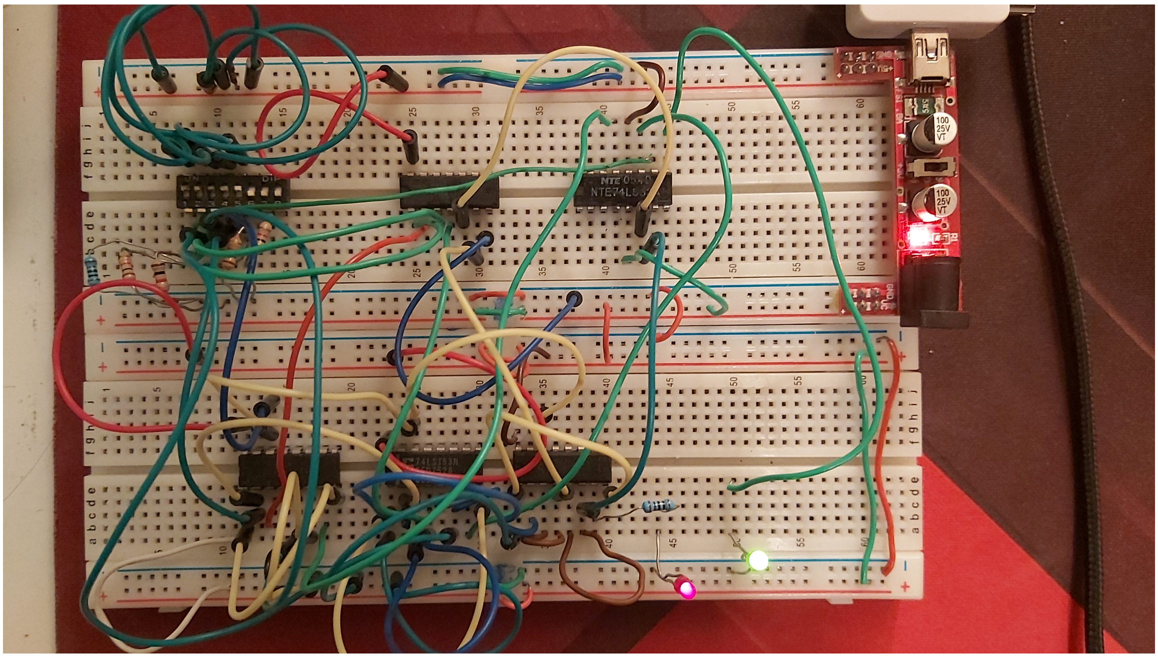
$ABC_i = 100$ and so $C_oY = 01$.



$ABC_i = 101$ and so $C_oY = 00$.



$ABC_i = 110$ and so $C_oY = 00$.



$ABC_i = 111$ and so $C_oY = 11$.