Computer Architecture project

4-bit ALU Design & Implementation

Project Members:-

1. Ahmed Sayed Ahmed (8
2. Ahmed Alaa Elden Yousef (13)
3. Hosny Mohamad Salem (42)
4. Shehata Abd ALmohsen (50)

Supervision of:-

Prof. Dr: Khaled El-Shafeay.

- ## Objective:

  Implementing a 4-bit ALU that performs the following functions:

  1. Add.
  2. Subtract.
  3. Increment.
  4. Decrement.
  5. Transfer.
  6. Bitwise AND.
  7. Bitwise OR.
  8. Bitwise XOR.
  9. Bitwise NOT.

-

Features:

- The ALU has 2 input operands A and B, 4 bit each (in Operations such as Increment, Decrement, Transfer, Bitwise Not, Operand B is neglected, and the operation will be Performed on Operand A)
- It has 4 input pins (S2, S1, S0, Cin) to select the operation that will be performed by the ALU.
- S2 determines whether the operation to be performed is Arithmetic or Logic Operation. 0 for Arithmetic, 1 for Logic. ○ If S2 is 1, Cin will be neglected.
- The output consists of 4 bits (G3, G2, G1, G0) in addition to the Carry out pin (Cout).
- 3 Flags are included in the design:
    1. Z (Zero Flag) is set if the output of an arithmetic or logic operation is zero.
    2. V (Overflow Flag) is set when overflow occurs within an arithmetic operation is performed.
    3. N (Negative Flag) is set if the output of an arithmetic operation is negative.

- **Tools:**

  The implementation and simulated on Protues.

- ## Function Table:
  - If S2 = 0, the ALU performs Arithmetic operation according to the following table:

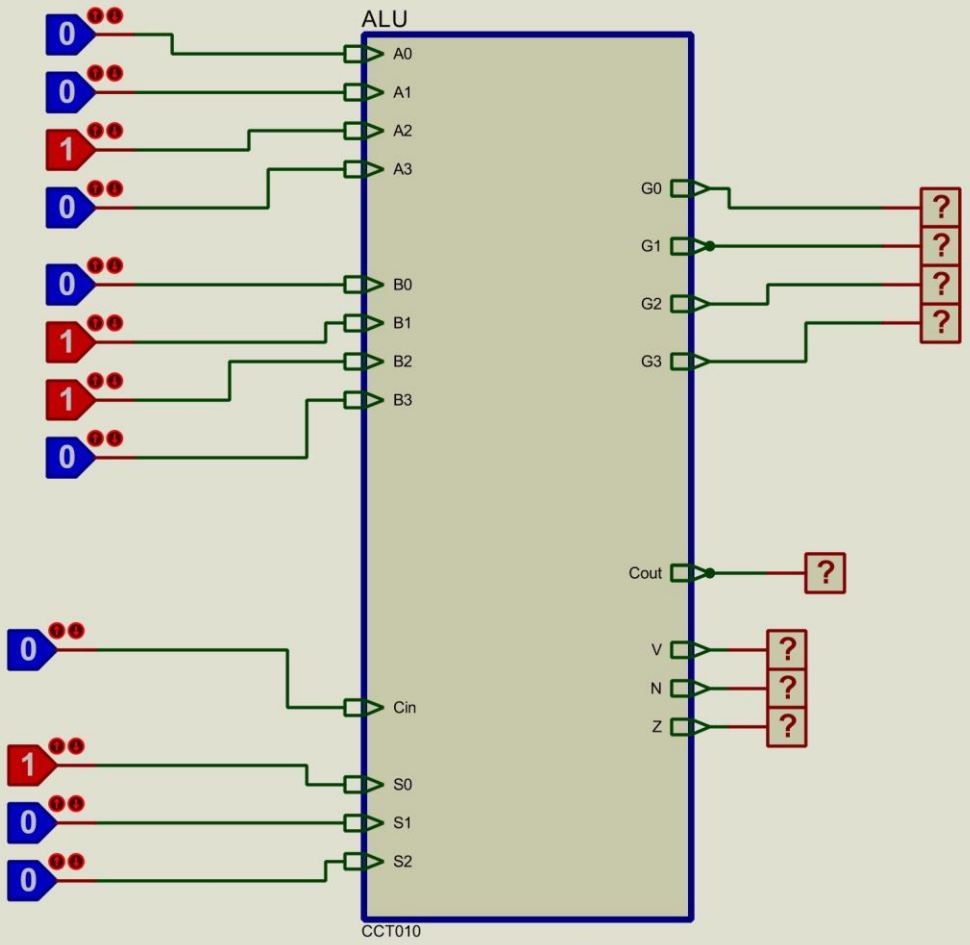
TABLE 1
Function Table for Arithmetic Circuit

Select		Input	$G = (A \mid Y \mid C_{in})$	
S_1	S_0	Y	$C_{in} = 0$	$C_{in} = 1$
0	0	all 0s	$G = A$ (transfer)	$G = A + 1$ (increment)
0	1	B	$G = A + B$ (add)	$G = A + B + 1$
1	0	$\overline{B}$	$G = A + \overline{B}$	$G = A + \overline{B} + 1$ (subtract)
1	1	all 1s	$G = A - 1$ (decrement)	$G = A$ (transfer)

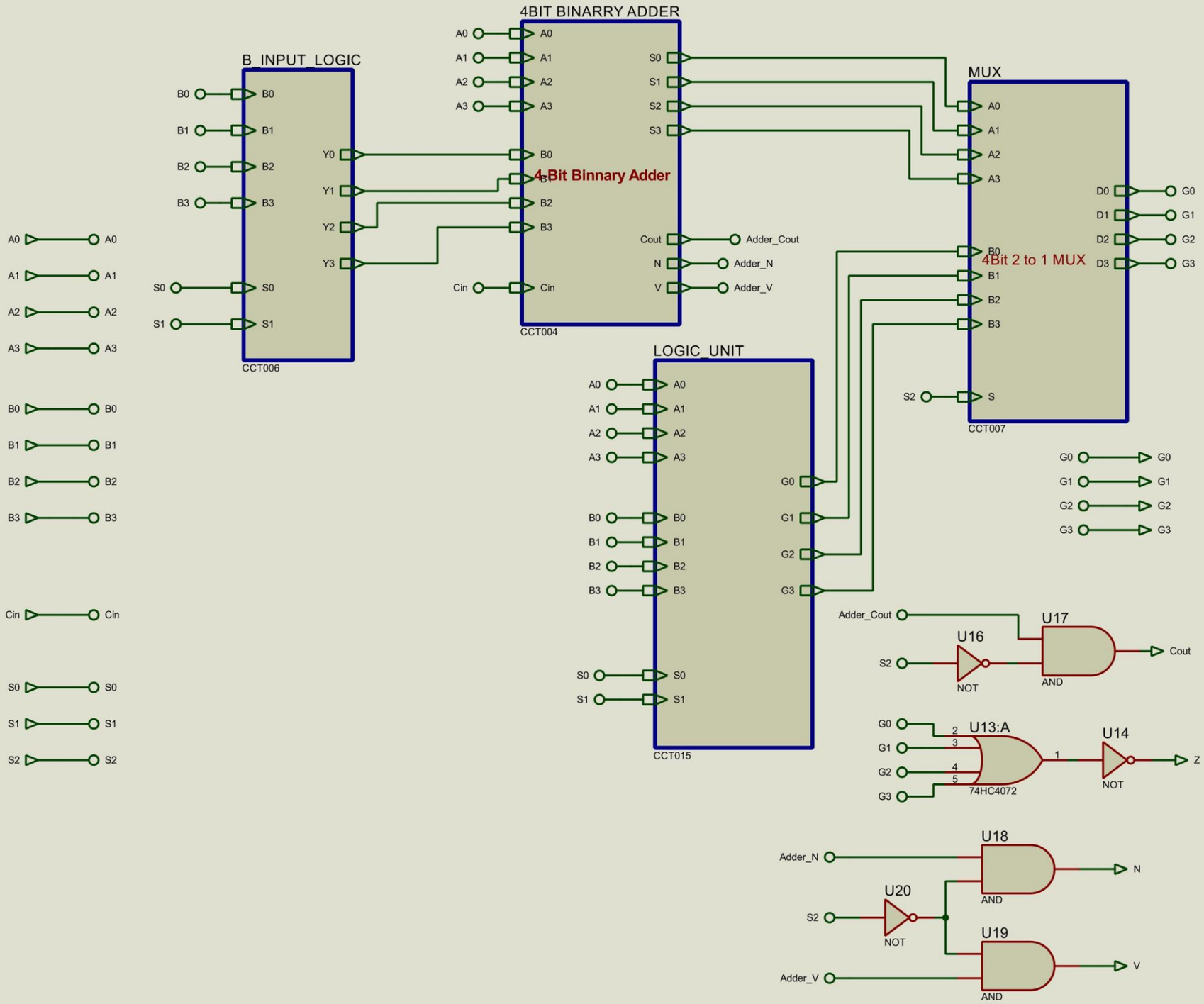
-
    - If S2 = 1, the ALU performs Logical operation according to the following table:

S_1 S_0	Output	Operation
0 0	$G = A \wedge B$	AND
0 1	$G = A \vee B$	OR
1 0	$G = A \oplus B$	XOR
1 1	$G = \overline{A}$	NOT

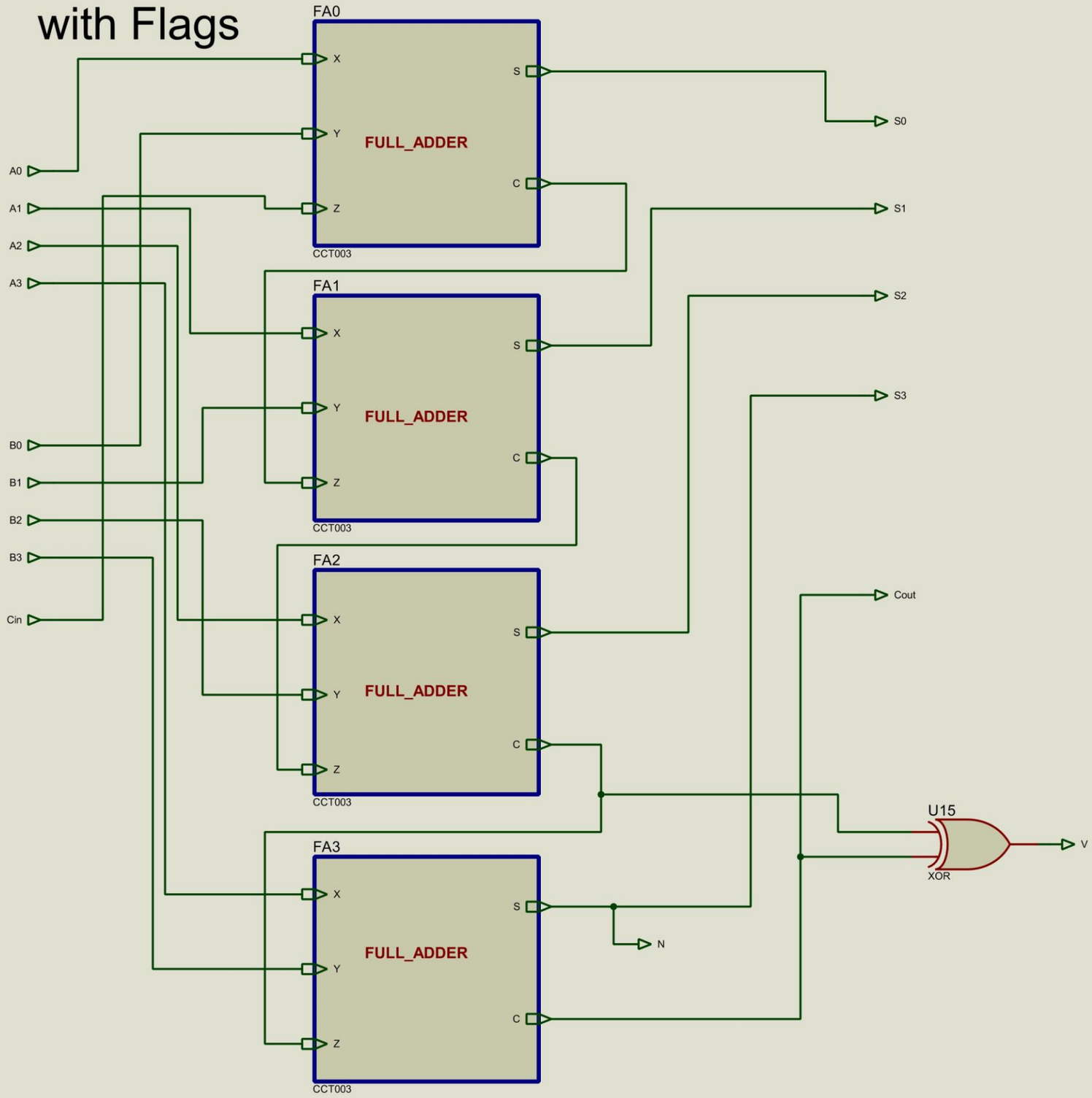
(b) Function table


ALU
A0
A1
A2
A3
B0
B1
B2
B3
Cin
S0
S1
S2
G0
G1
G2
G3
Cout
V
N
Z
CCT010

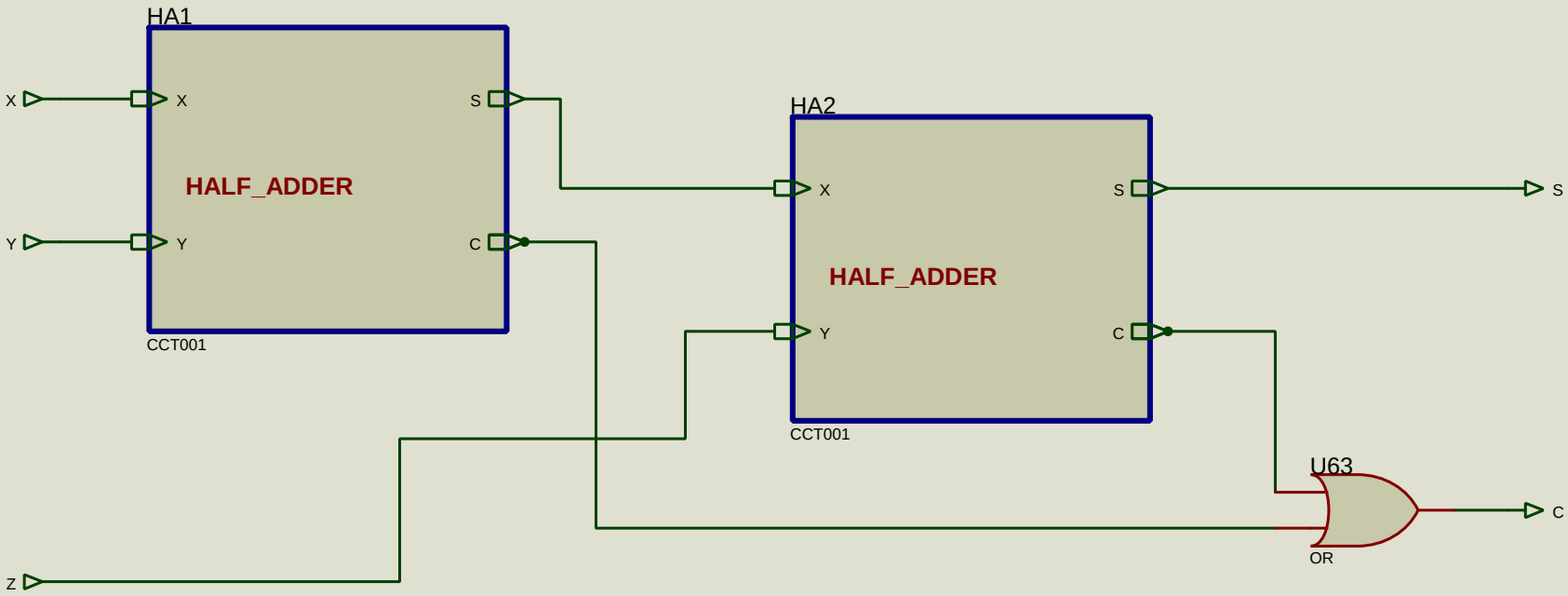
Logic Diagram of the 4bit ALU



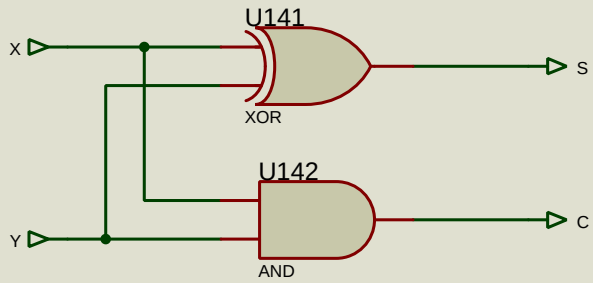
Logic Diagram of the 4bit Binarry Adder with Flags

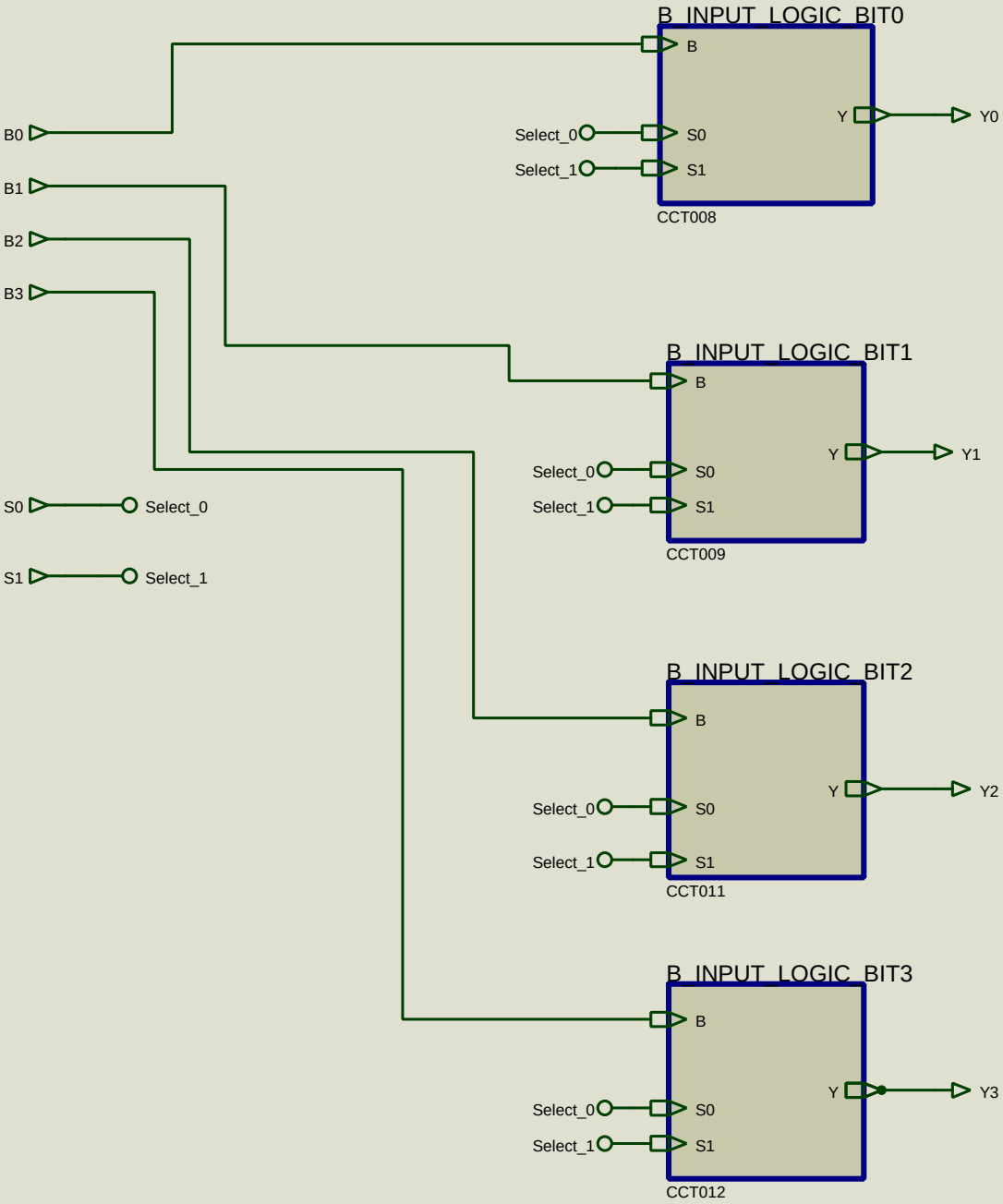


Design of the FULL ADDER

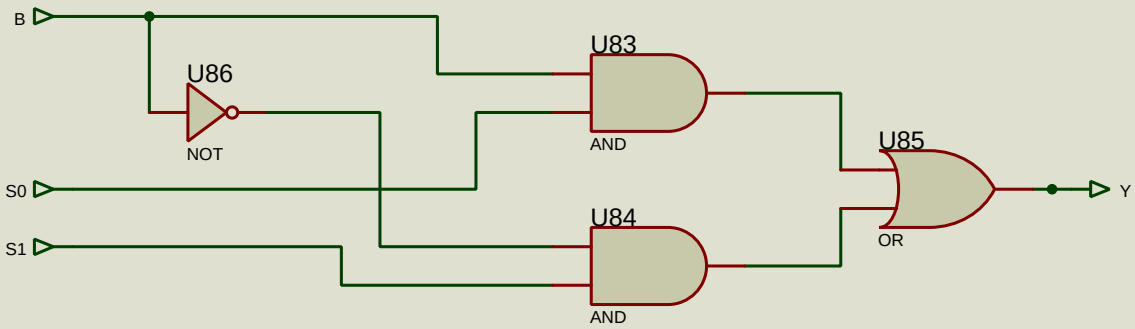


Design of the HALF ADDER

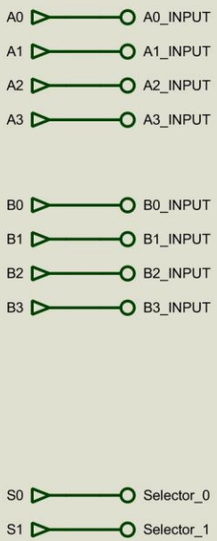


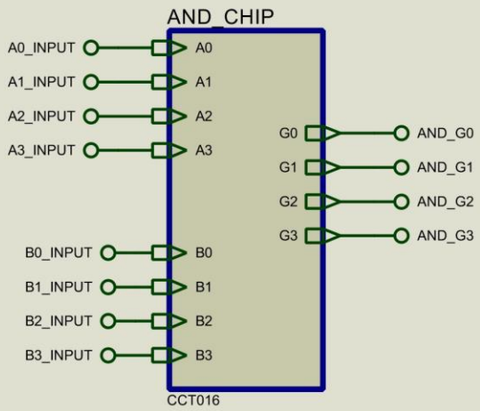


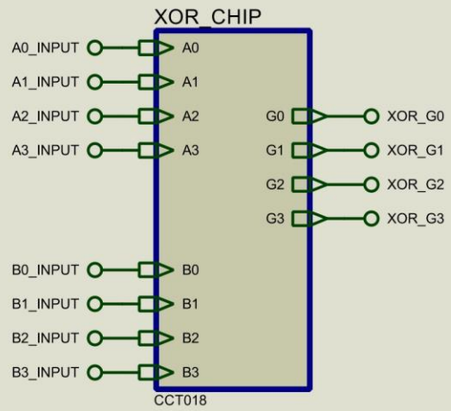
Logic Diagram of the 4bit Combinational circuit that aplied on operand B

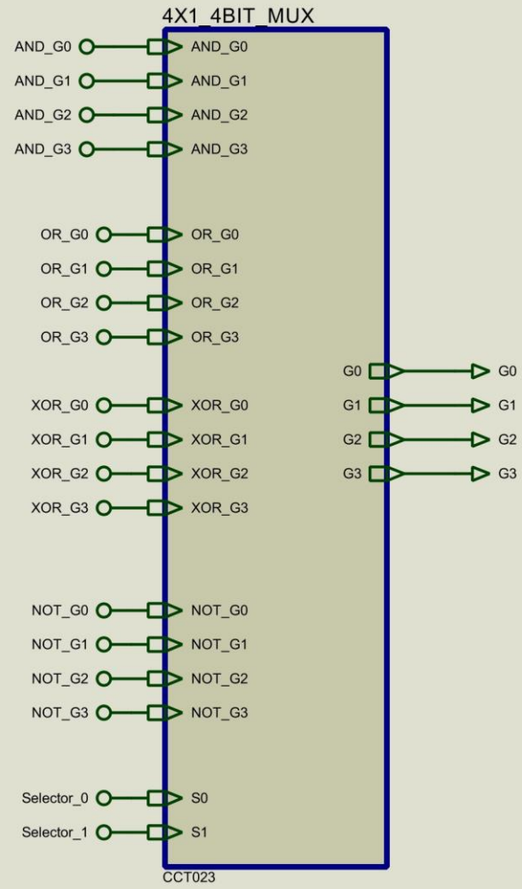


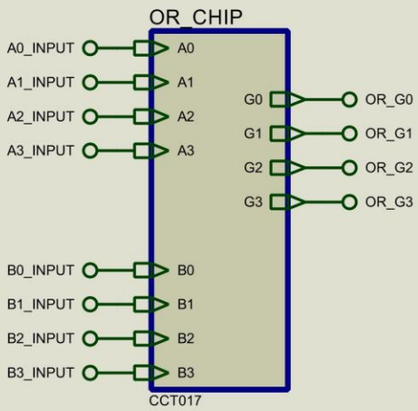
Design of Each bit of the combinational circuit that is applied on Operand B

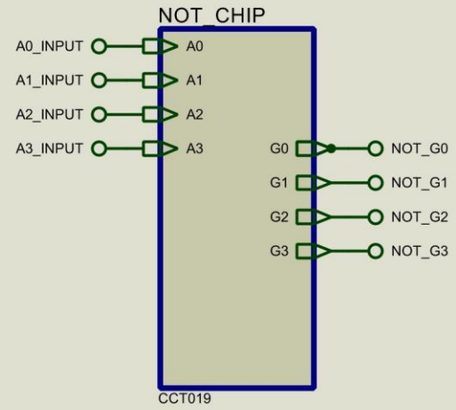




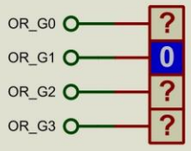


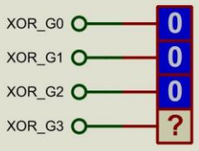


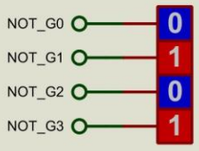




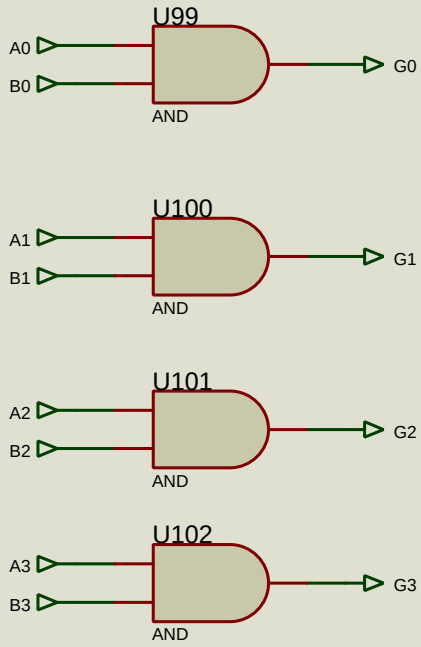
Design of the Logic Unit



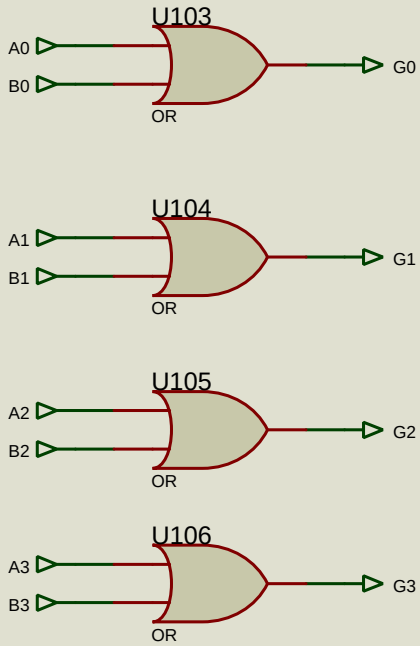




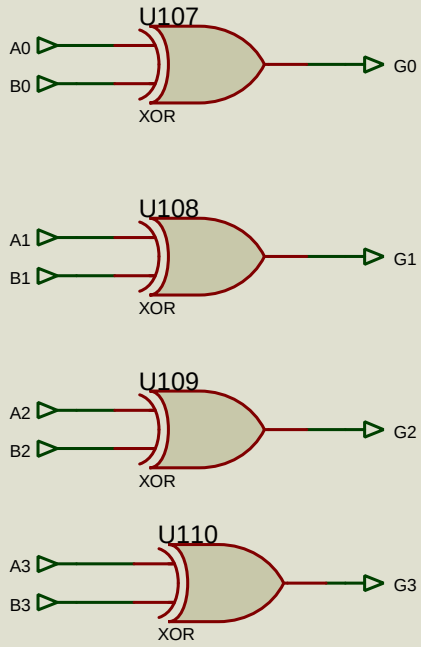
Design of the 4-bit AND Function



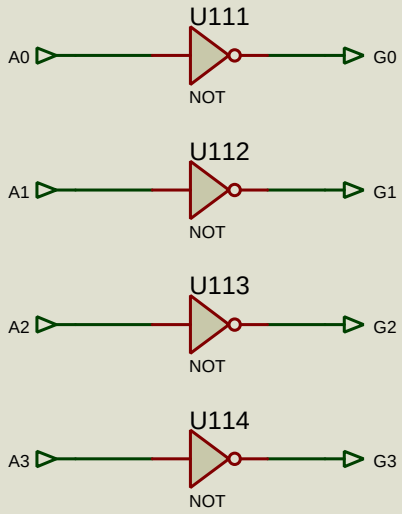
Design of the 4-bit OR Function



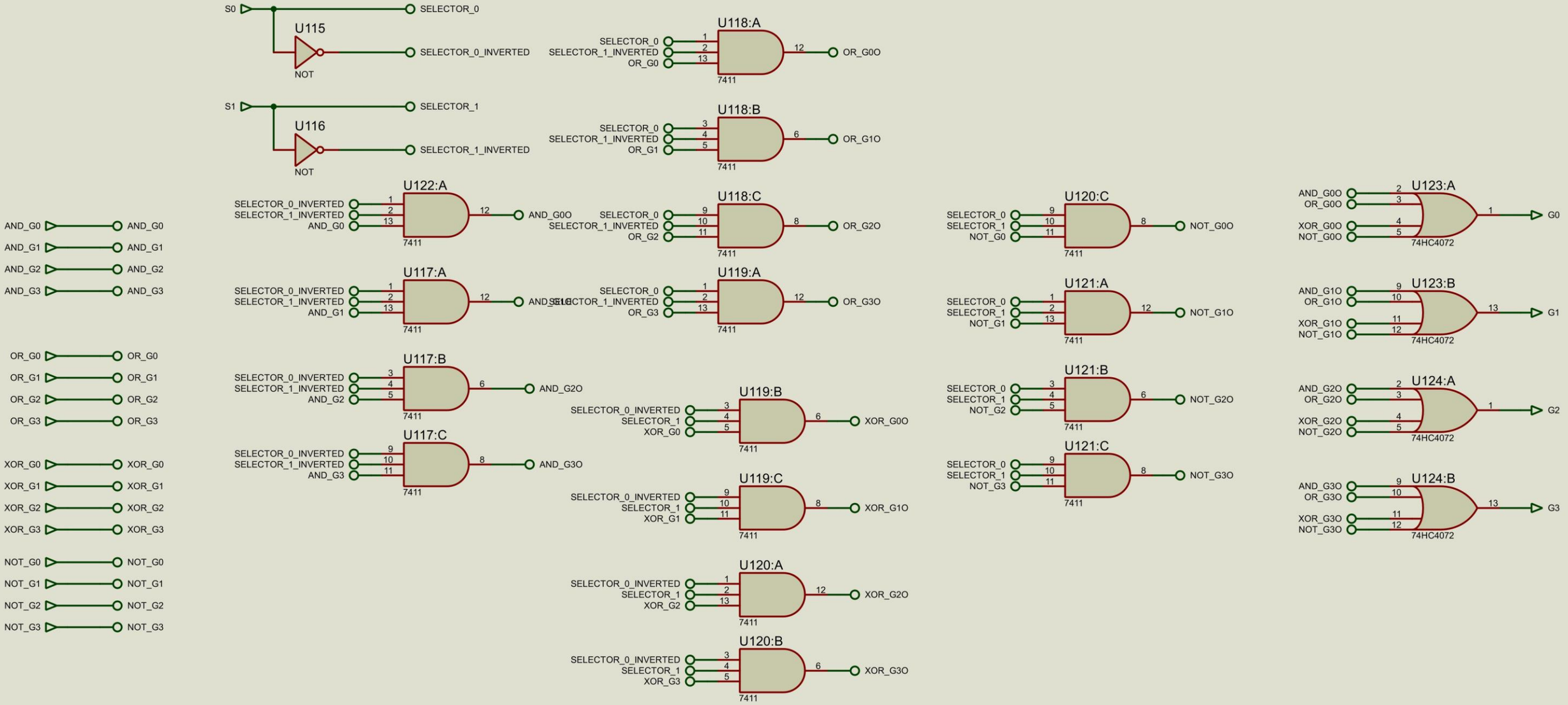
Design of the 4-bit XOR Function

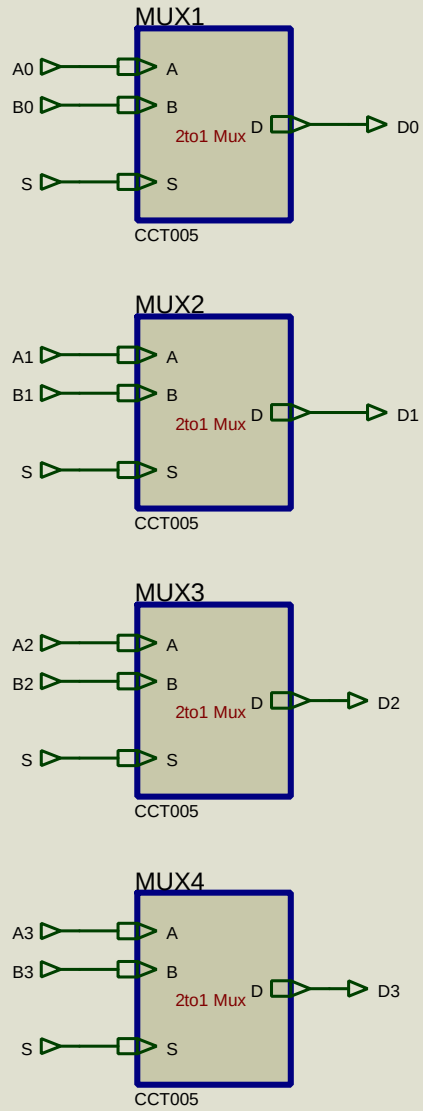


Design of the 4-bit NOT Function



Design of the 4-bit 4 to 1 MUX





Internal design of the 4-bit 2 to 1 MUX

Design of 1-bit 2 to 1 MUX