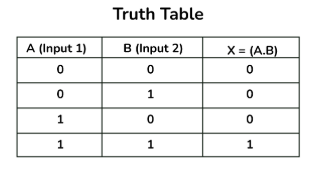
Task 1: Function Calculator (Using Digital Logic Circuits)

A 4-function calculator that can perform addition, subtraction, multiplication, and division on two 4-bit binary numbers

Digital logic components

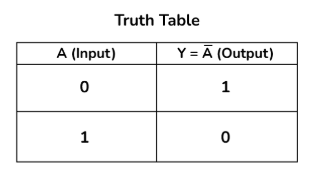
i)AND:

An AND gate is used to perform logical Multiplication of binary input. The Output state of the AND gate will be high(1) if both the inputs are high(1), else the output state will be low(0) if any of the inputs is low(0).



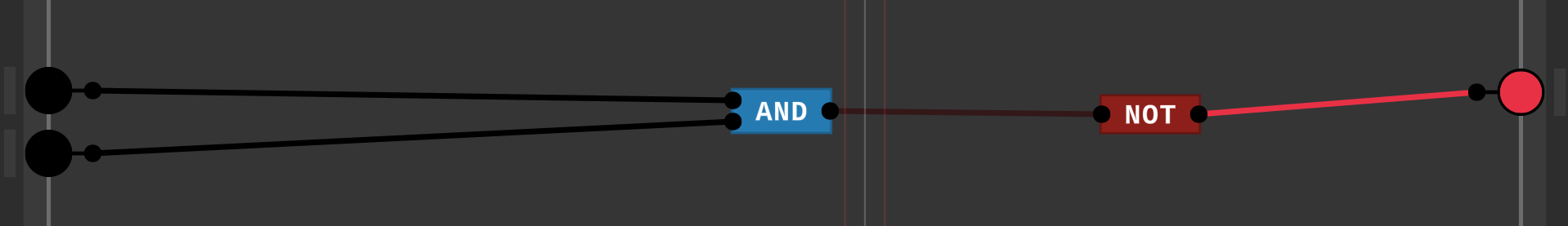
ii)NOT:

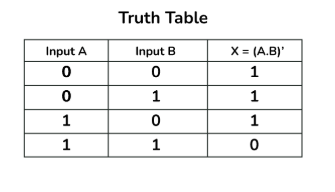
It is also known as an inverter or inverting buffer. When the input signal is “low,” the output signal is “high,” and vice versa.



iii)NAND:

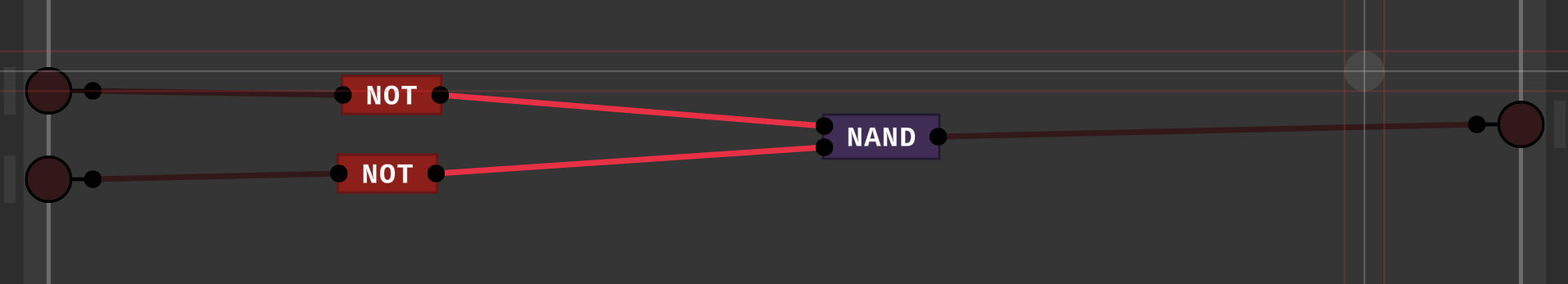
The NAND gate or “Not AND” combines two basic logic gates: the AND gate and the NOT gate, which are connected in series. It takes two or more inputs and gives only one output. The output of the NAND gate will give a result high(1) when either of its inputs is high(1) or both of its inputs are low(0)

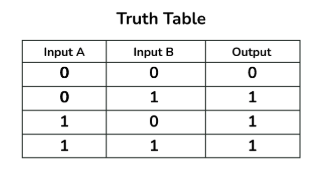




iv)OR:

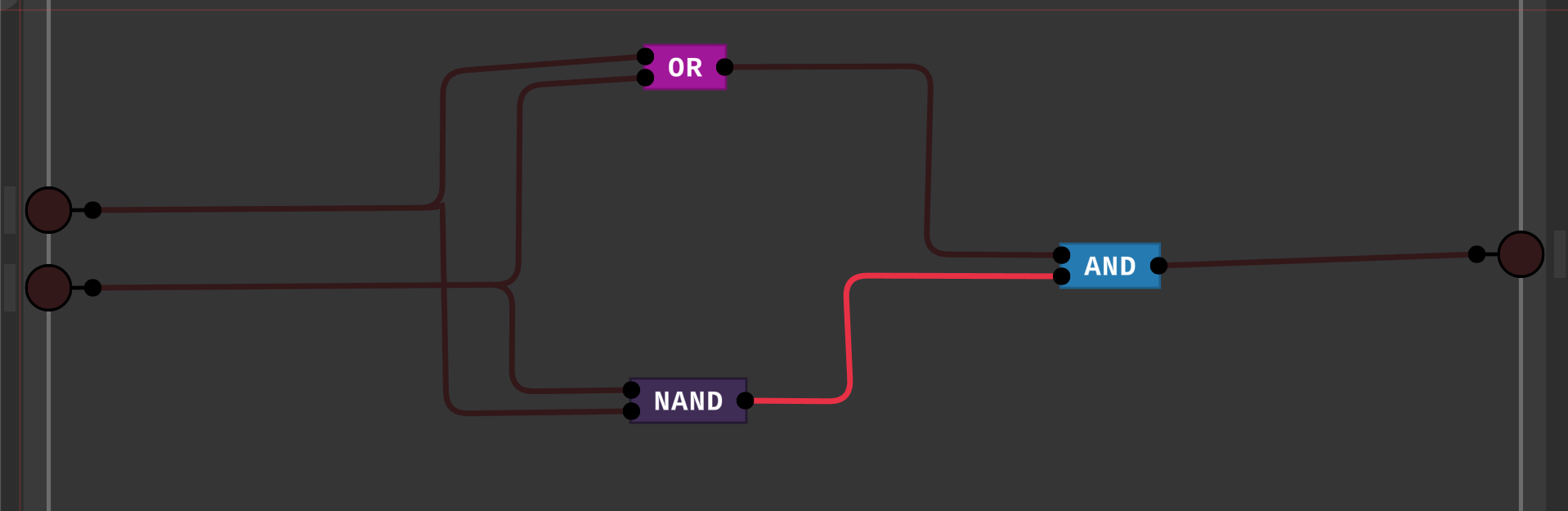
The output state of the OR gate will be high i.e.,(1) if any of the input states is high or 1, else output state will be low i.e., 0

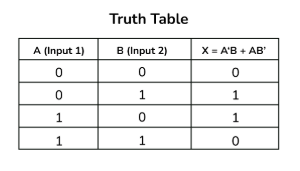




v)XOR:

 The XOR gate can take only two inputs at a time and give an output. The output of the XOR gate is high(1) only when its two inputs are dissimilar i.e., if one of them is low(0) then the other one will be high(1).





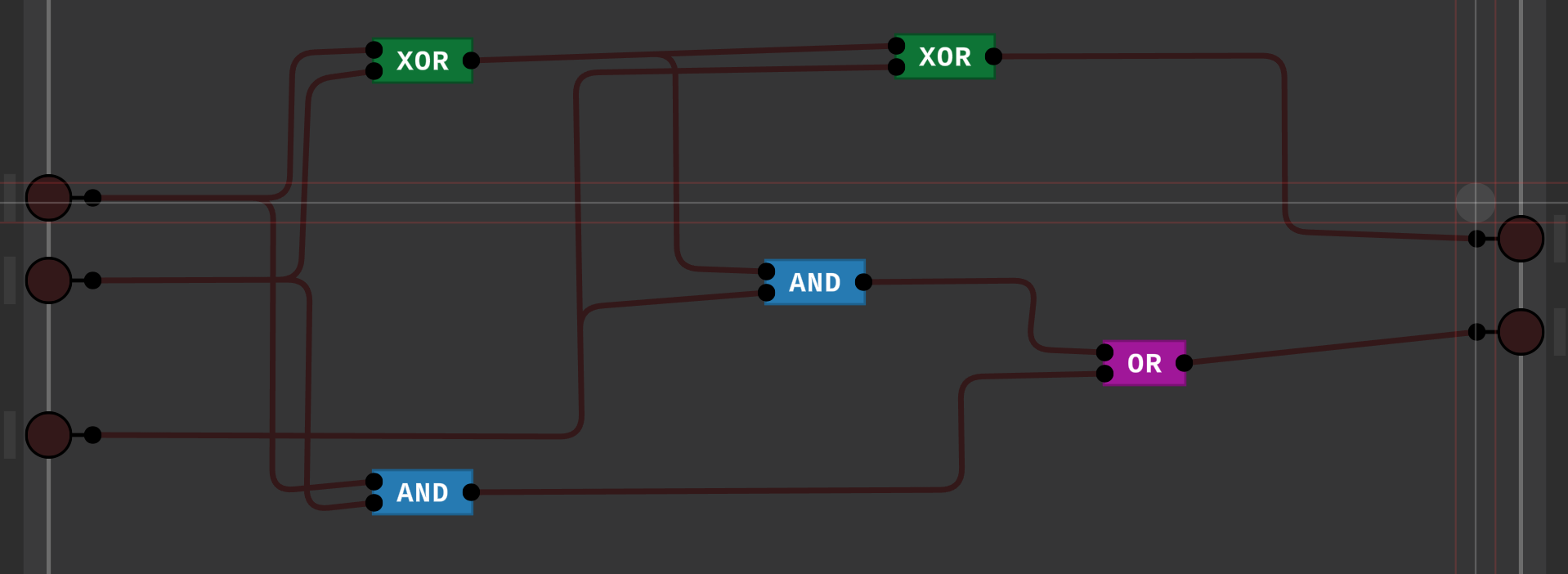
Arithmetic Logic Unit (ALU)

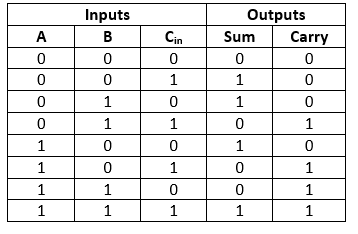
The Arithmetic Logic Unit (ALU) performs addition and subtraction as basic arithmetic operations. The ALU uses a sequence of logic gates to perform these operations.

I)ADDITION

i)ADDER:

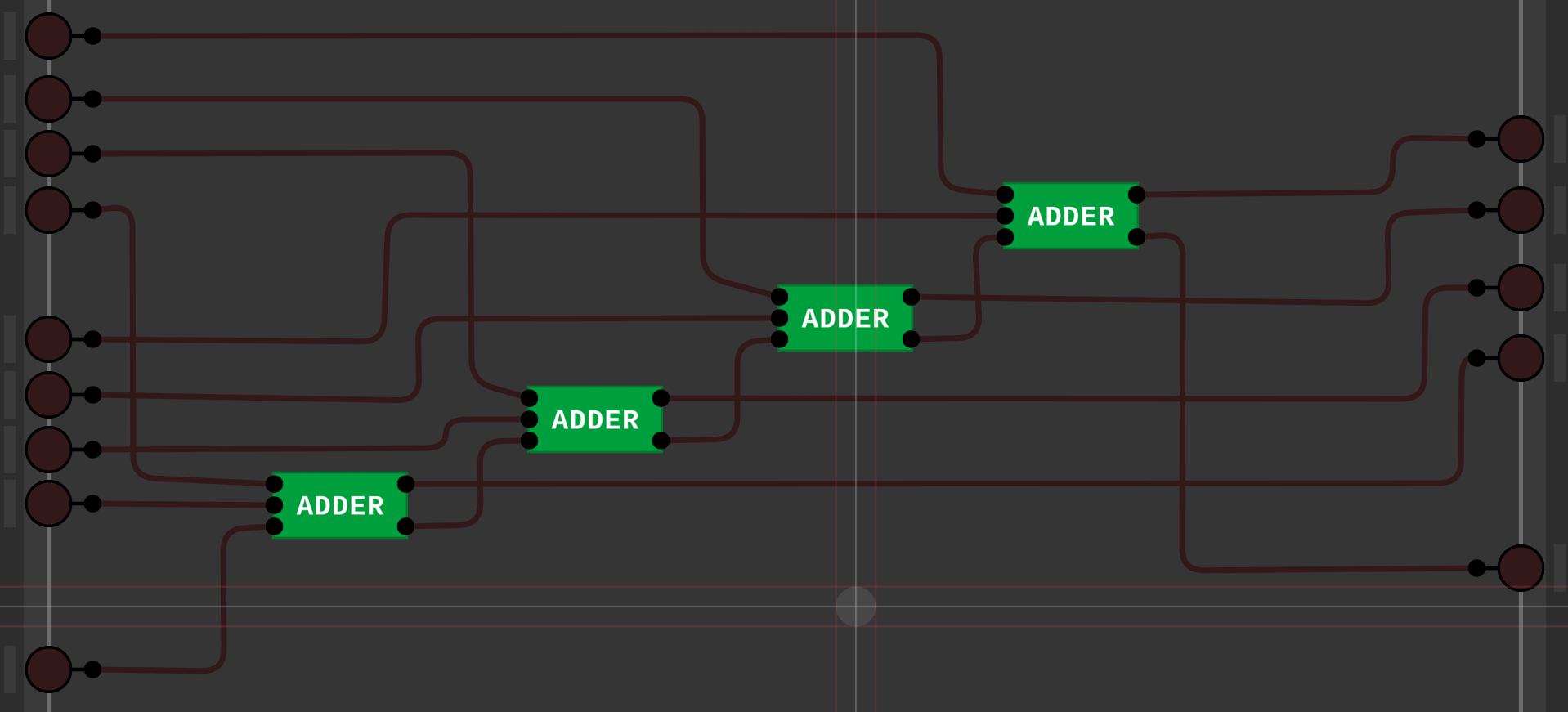
The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM. The C-OUT is also known as the majority 1’s detector, whose output goes high when more than one input is high.





ii)4BIT ADDER:

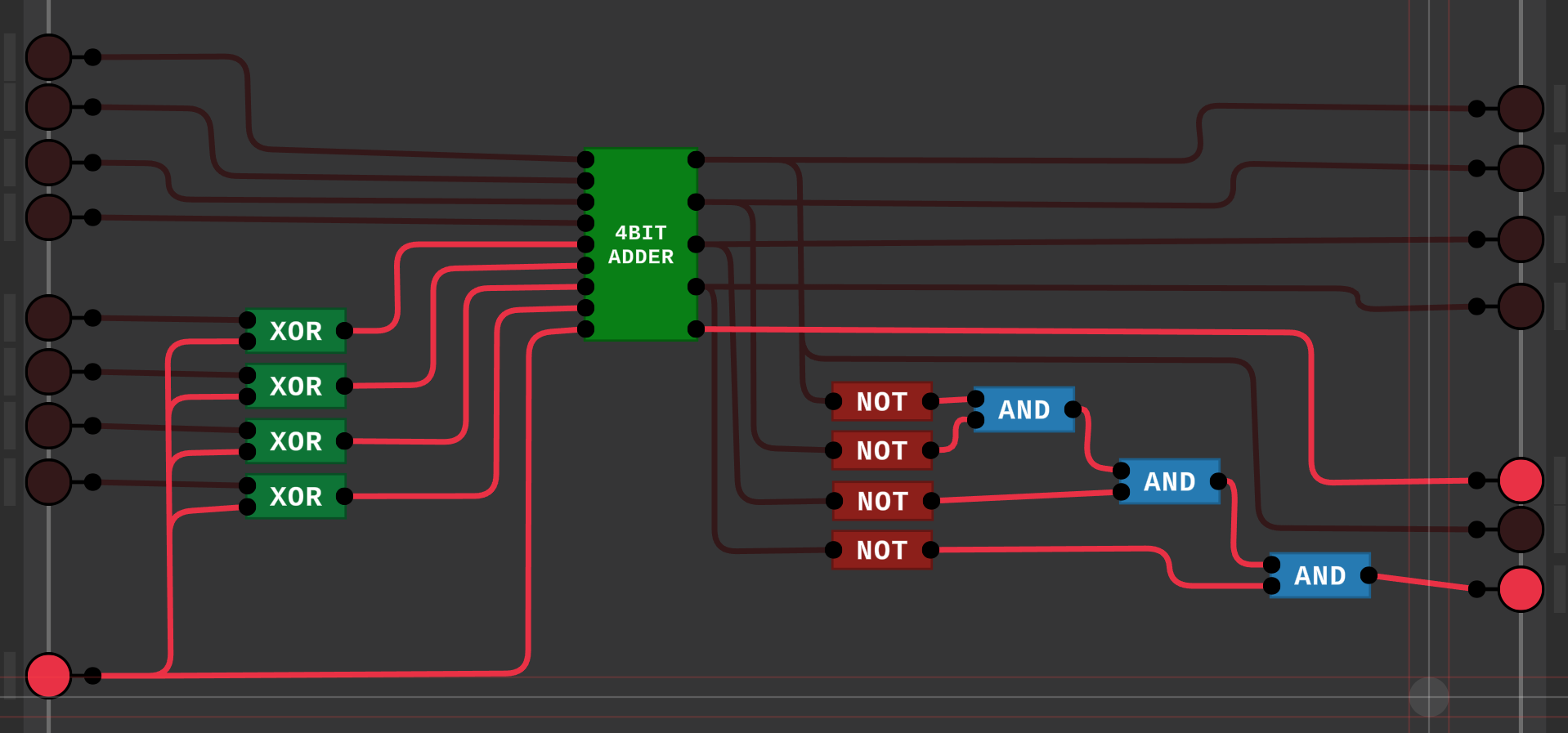
A full adder logic is designed in such a manner that can take eight inputs together to create a byte-wide adder and cascade the carry bit from one adder to another



The working video has been uploaded.

II)SUBTRACTION:

To subtract B from A, we must invert B and add 1 (2’s compliment). We convert B to -B and then we add -B to A to perform the subtraction operation



The working video has been uploaded

III)MULTIPLICATION

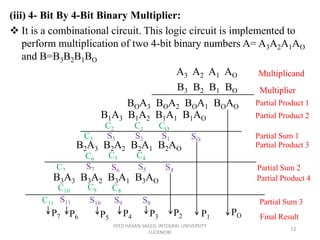
involves multiplying two 4-bit binary numbers, resulting in an 8-bit binary product

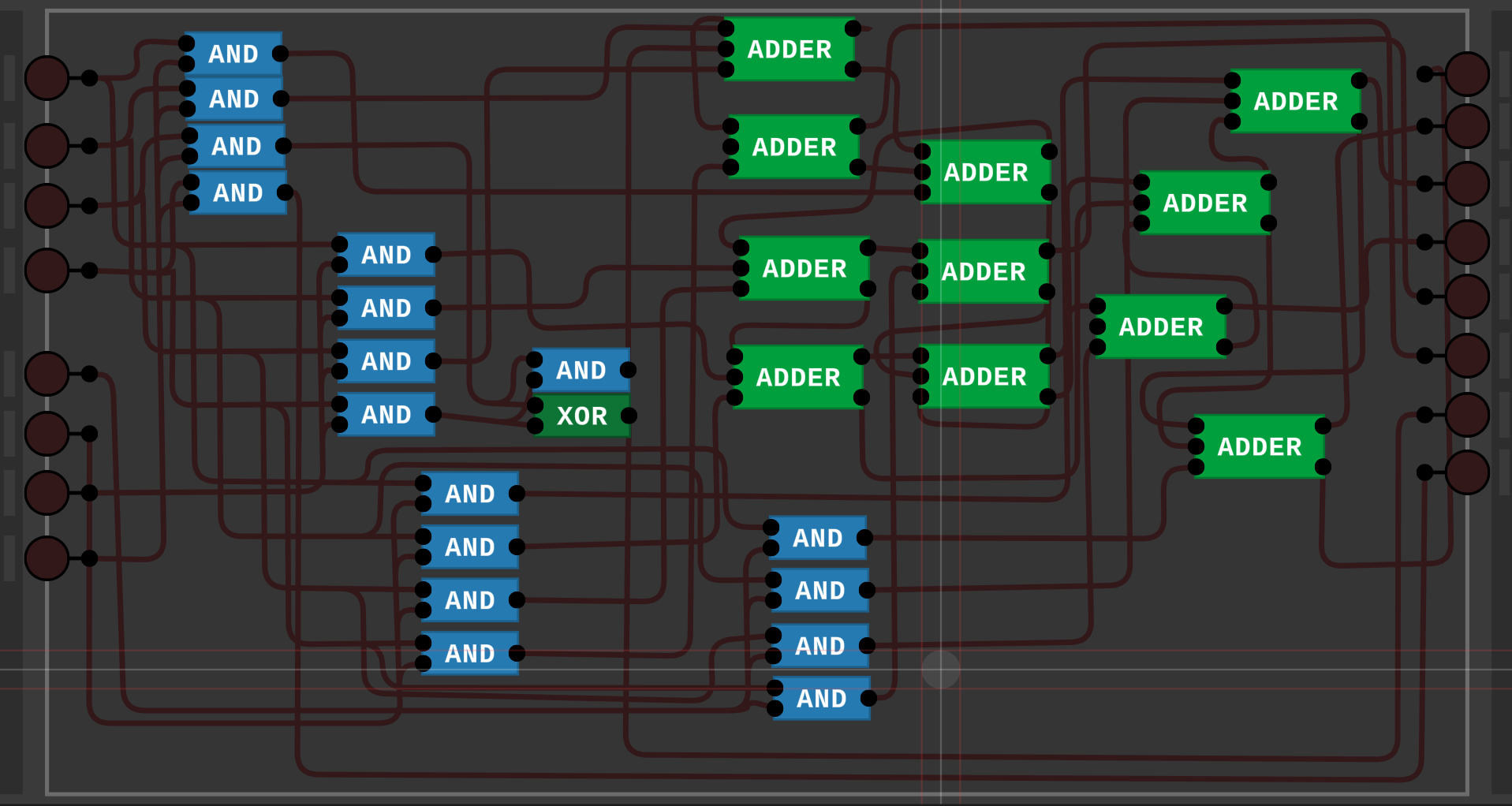
A=A3​A2​A1​A0​ and B=B3B2B1B0B

**Partial Products**: Multiply each bit of the first number with each bit of the second number, creating several smaller results (partial products).

**Alignment**: Each partial product is shifted based on the position of the bits being multiplied (like decimal multiplication).

**Add Together**: Add all the partial products to get the final 8-bit result.



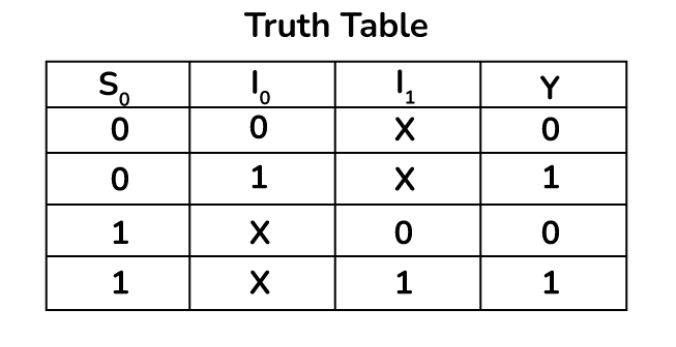


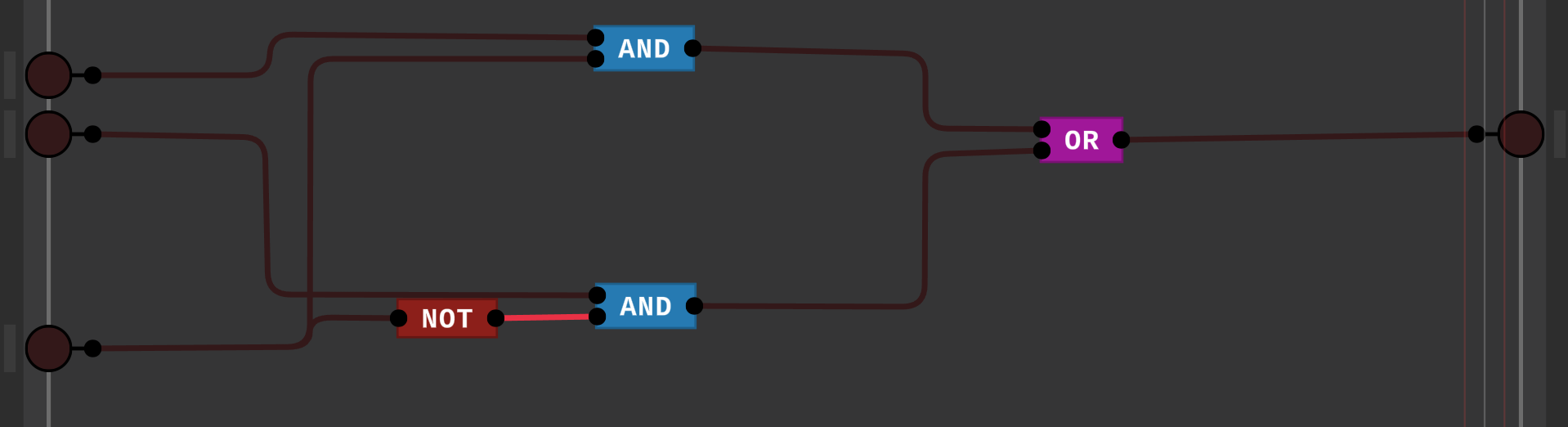
The working video has been uploaded.

IV)DIVISION

i)2x1 multiplexer:

it gets 2 inputs and based on the selector pin it decides which input has to be given as an output

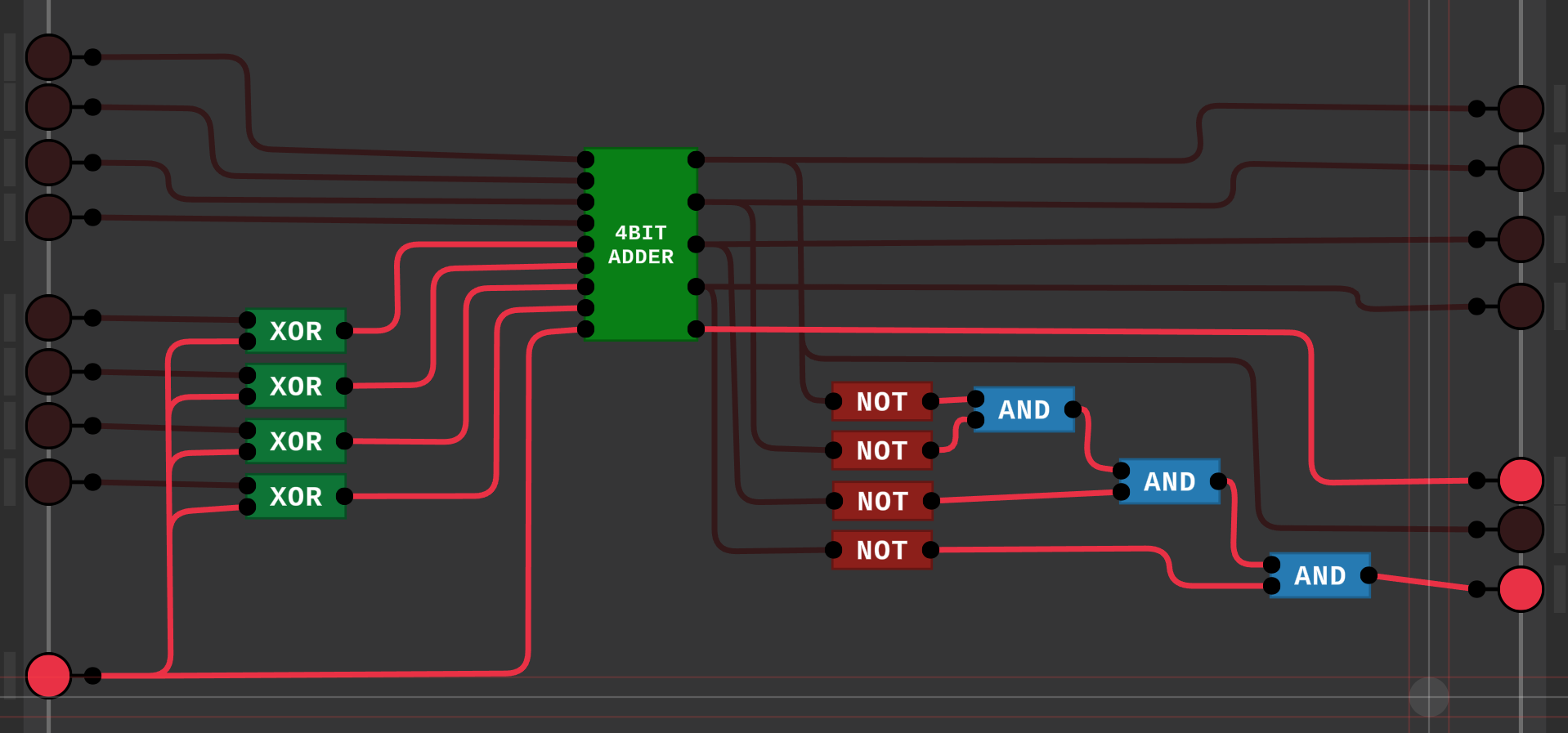




The working video has been uploaded

ii)ALU(for subtraction)

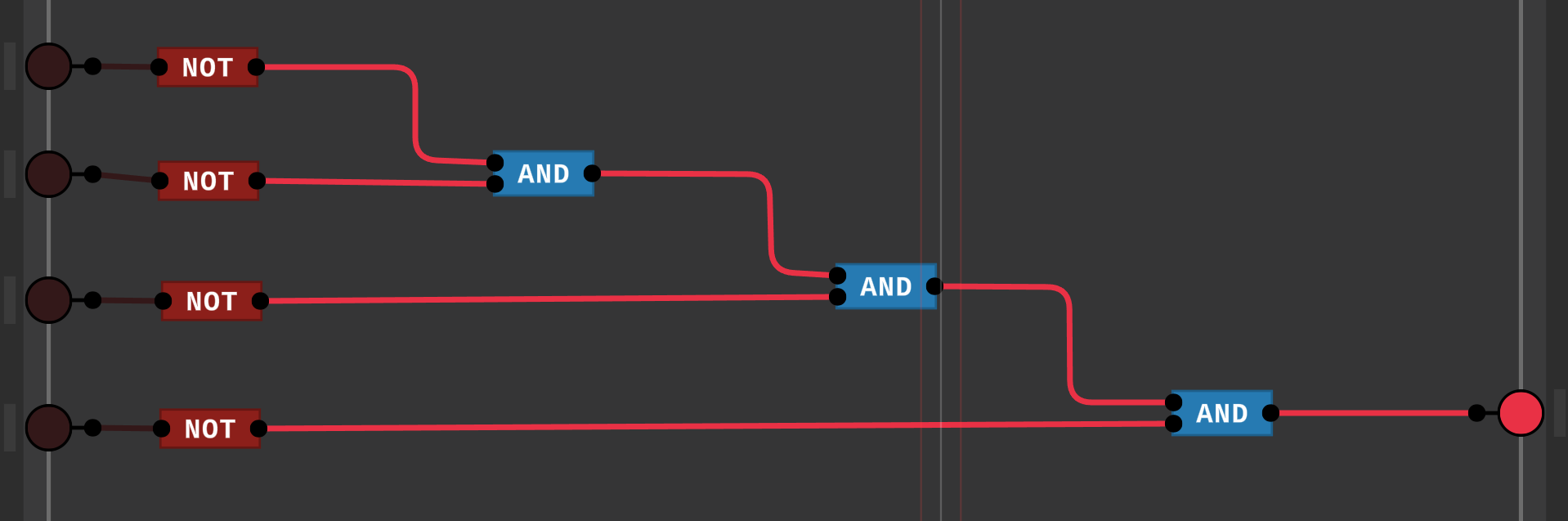
To subtract B from A, we must invert B and add 1 (2’s compliment). We convert B to -B and then add -B to A to perform the subtraction operation.



The working video has been uploaded.

iii)check 0

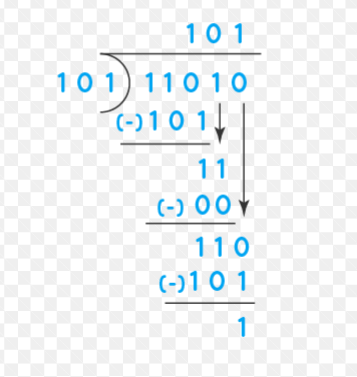
the division is not possible when the divisor is equal to 0, to check a number is zero, this is used



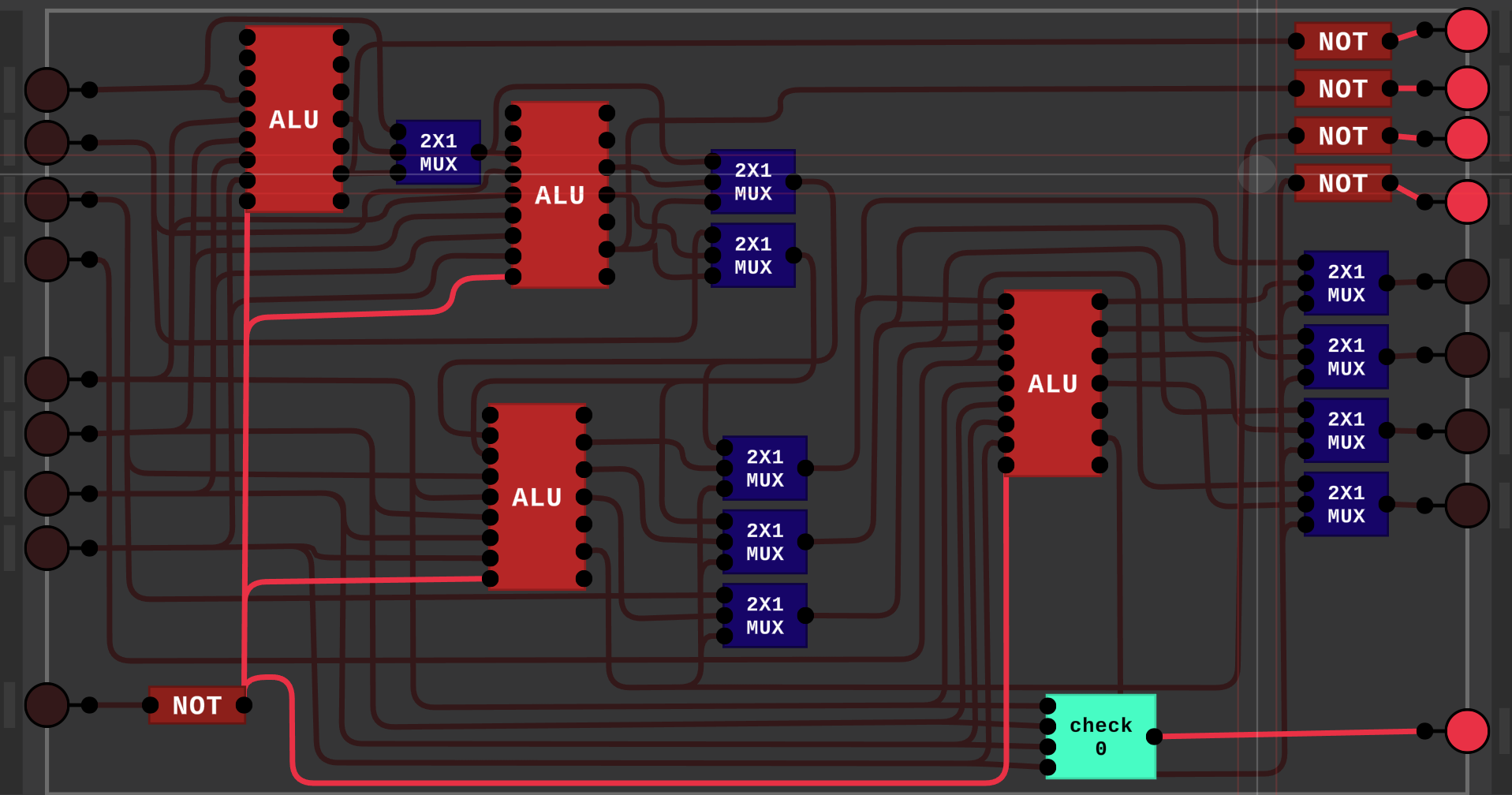
iv)complete div

**Repeated Subtraction (Long Division Method)**:

* Check if the current working value is **greater than or equal to the divisor**.
* If yes:
  + Subtract the divisor from it.
  + Write 111 in the quotient for that position.
* If no:
  + Write 000 in the quotient for that position.
* Bring down the next bit of the dividend and repeat



The 4-bit division breaks down the dividend bit by bit, comparing it with the divisor, subtracting when possible, and appending to the quotient. The process ends with a quotient and a remainder.



Function Calculator

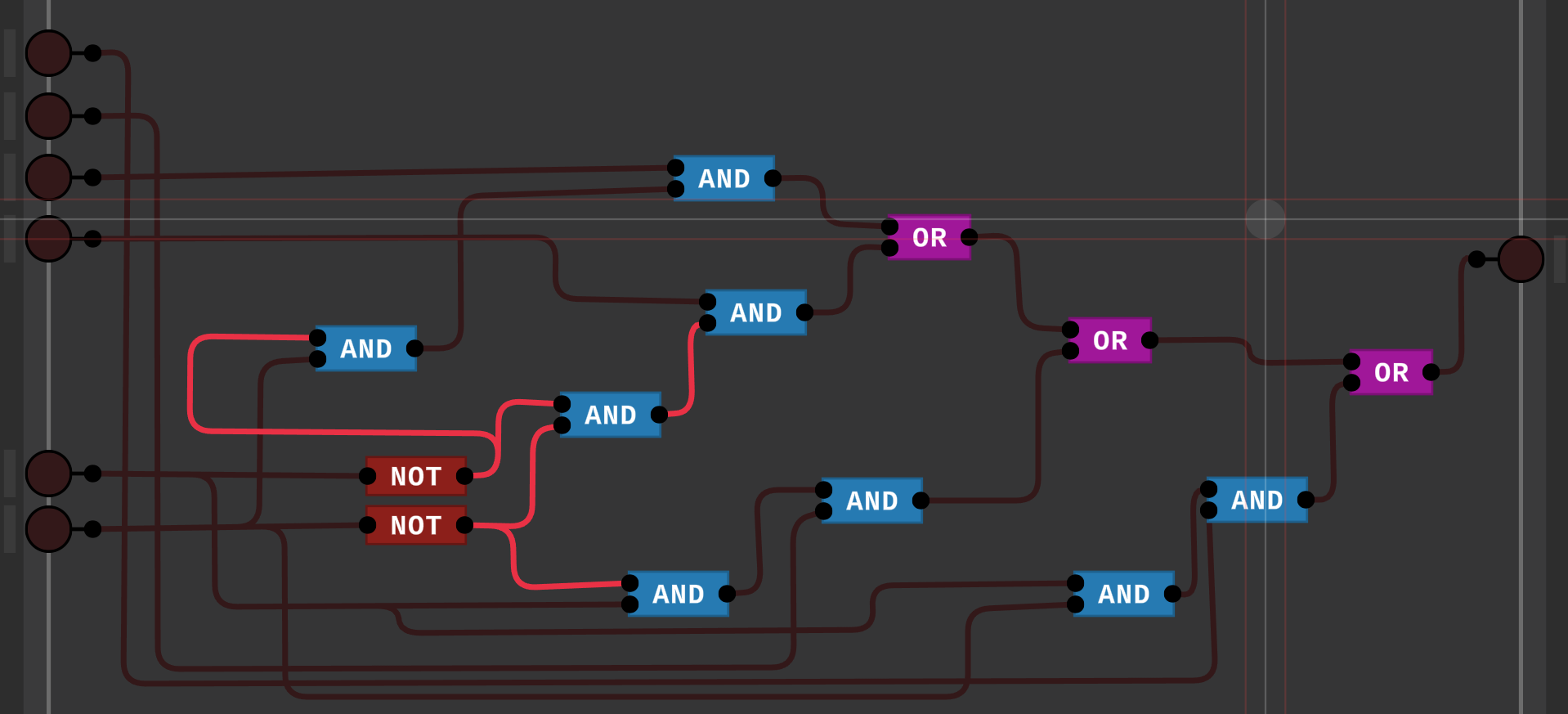
i)4x1 multiplexer:

**Inputs**: Four data inputs (I0, I1, I2, I3 ​).

**Select Lines**: Two selection inputs (S0, S1​) that determine which input to pass.

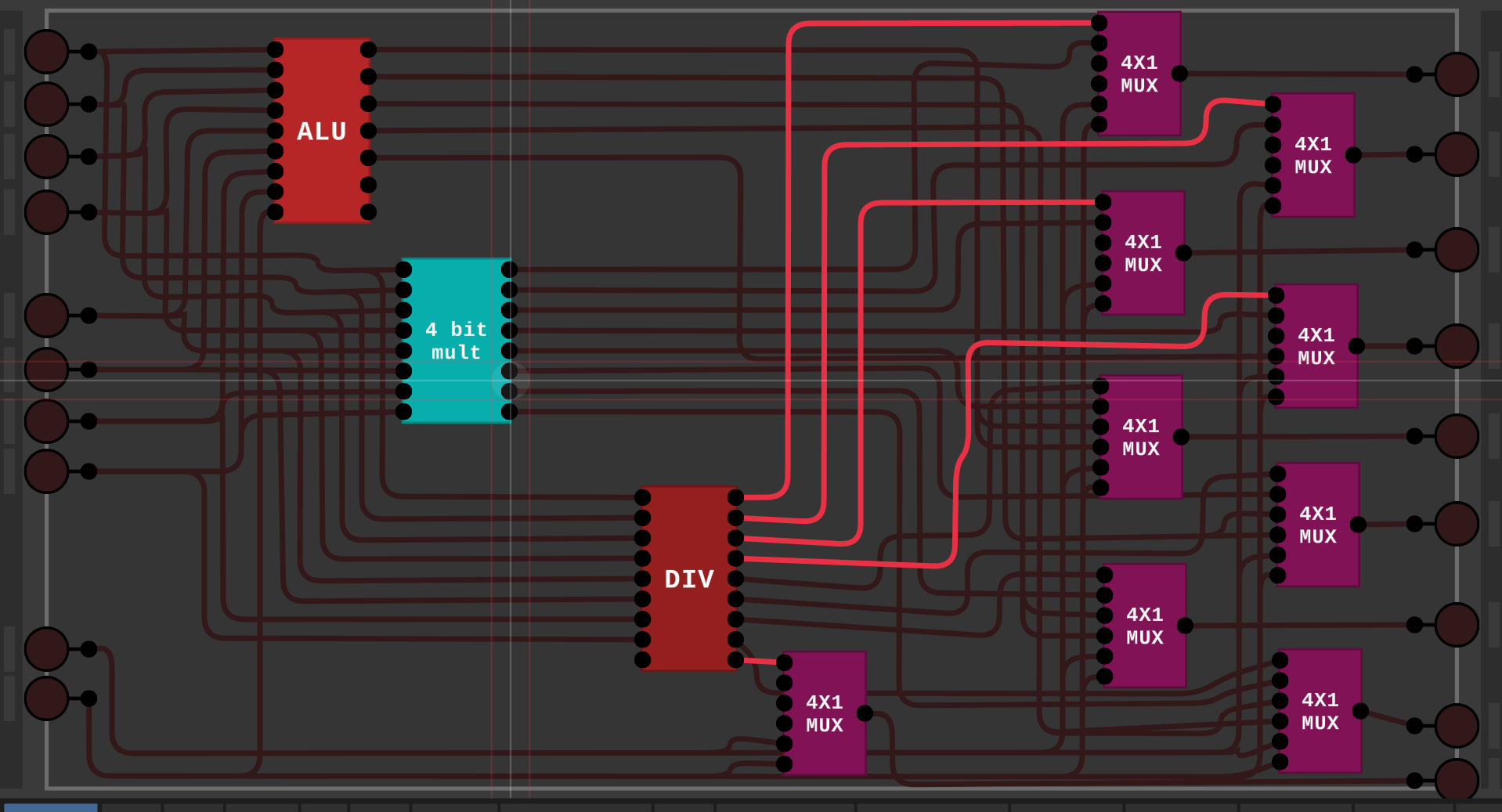
**Output**: Single output (Y).

* S1​S0​=00:Y=I0​
* S1s0=01:Y=I1
* S1S0=10:Y=I2​
* S1S0=11:Y=I3​



ii)CALCULATOR:

By connecting all ALU, 4-bit mult, DIV, and 4x1 mux, we obtain the calculator.



Left first 4 pins represent one Number and next 4 bits represent another number and the next 2 are selector pins that select which operation has to be performed

Right last pin is to check at division operation whether the divider is 0 or not, The next 4 bits are the answers for the addition and subtraction operation, all 8 bits are used in multiplication operations and division operations first 4 bits represent co-efficient and the next four bits represent the remainder.

The working video is uploaded…….