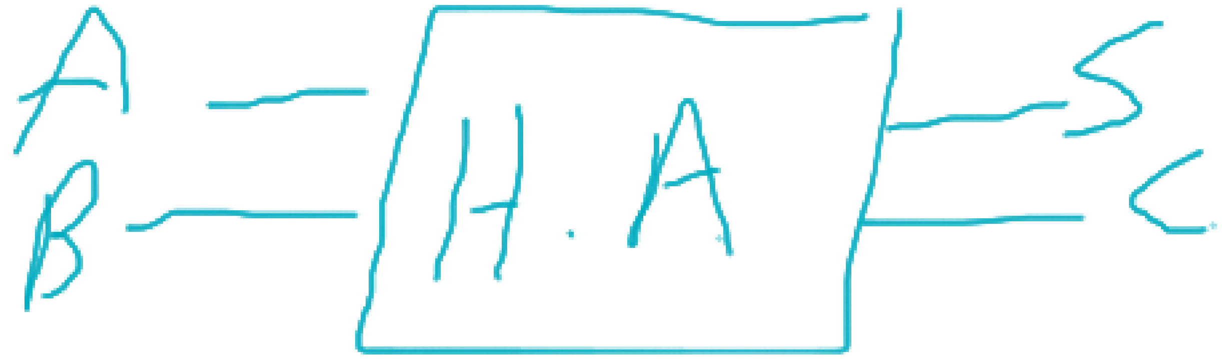
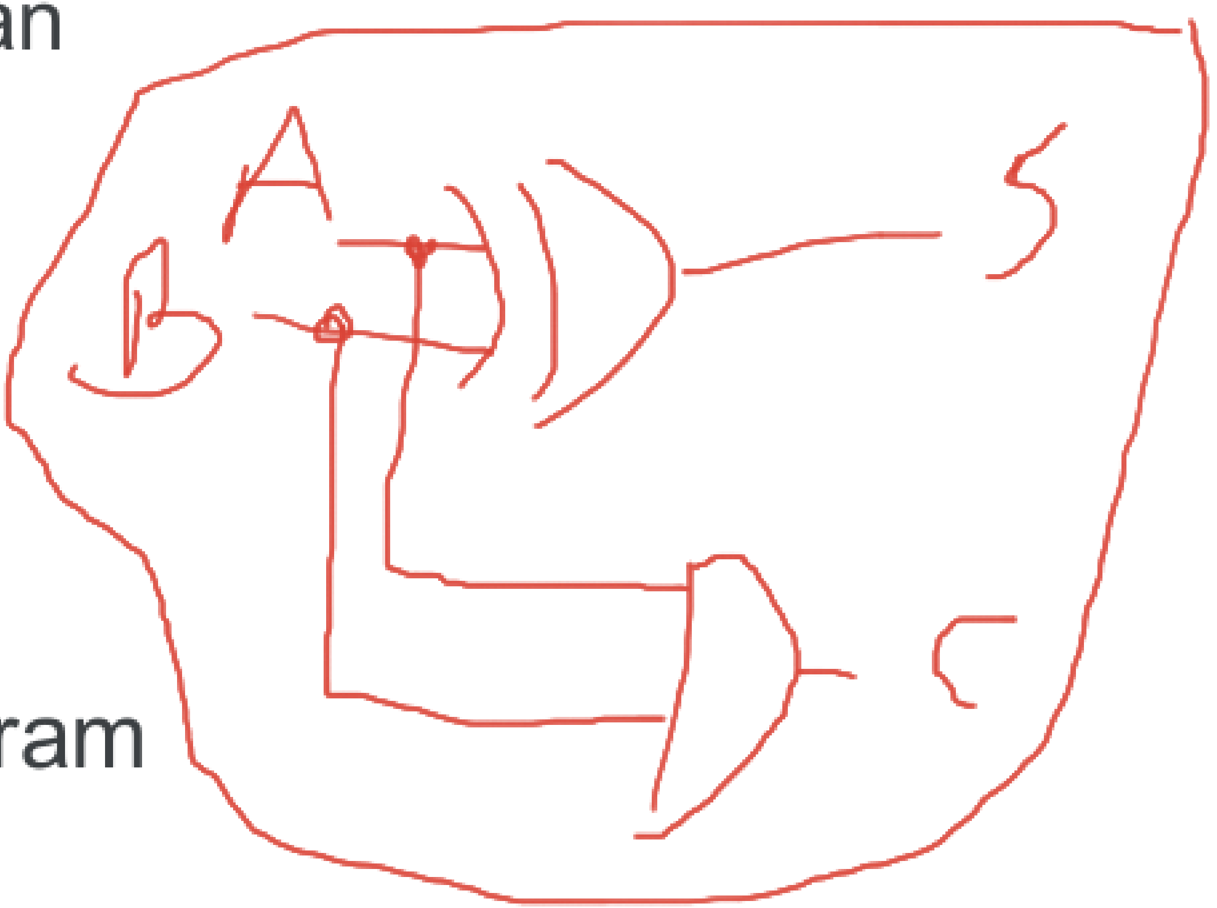
Half Adder

① Unsigned
② 1-bit

1- System specification
2- I/O Set
3- Control Signals
4- Block Diagram ==> Design
5- Timing Diadram
6- Logic Description (HDL, Schematic..etc)
7- Testing plan

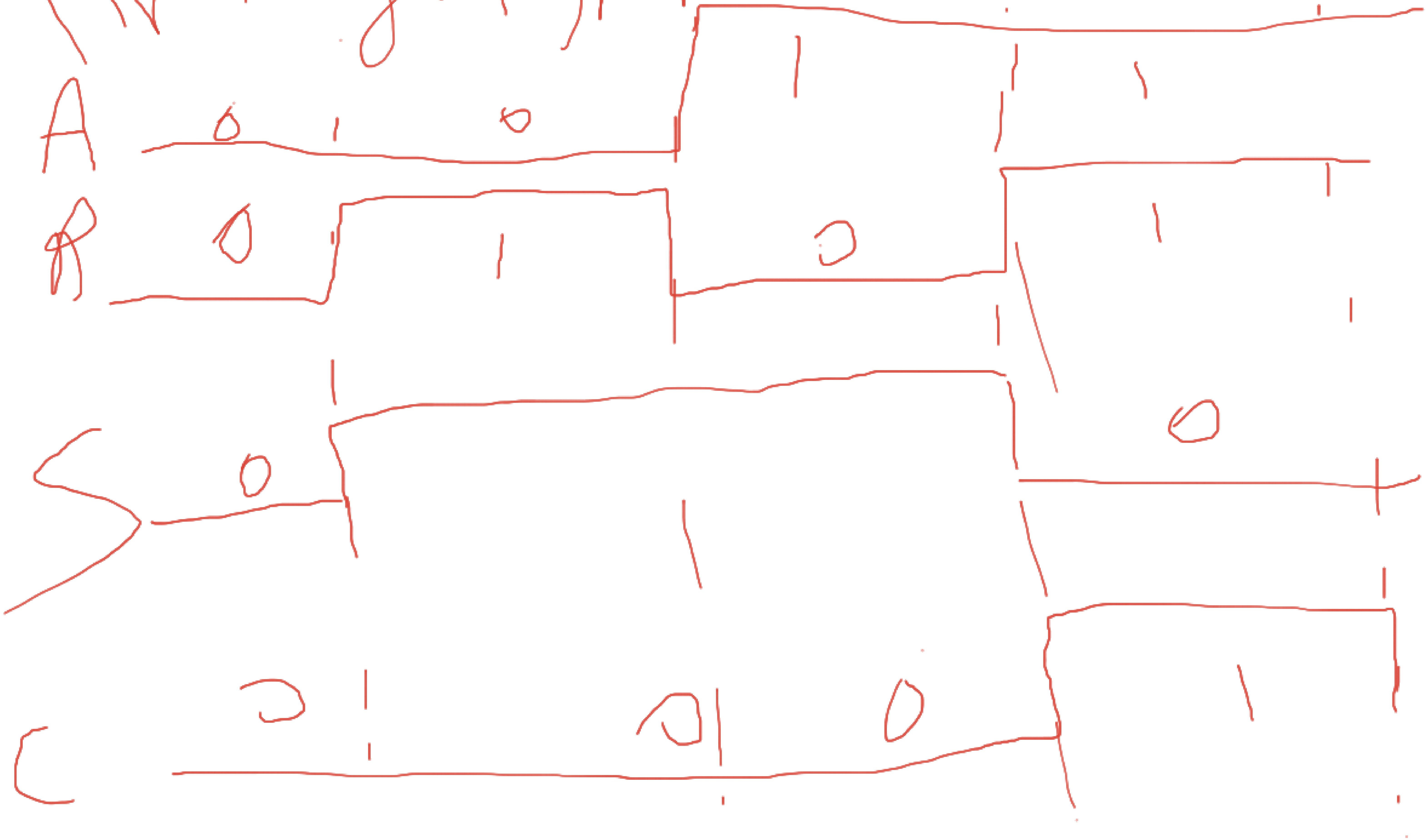


A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

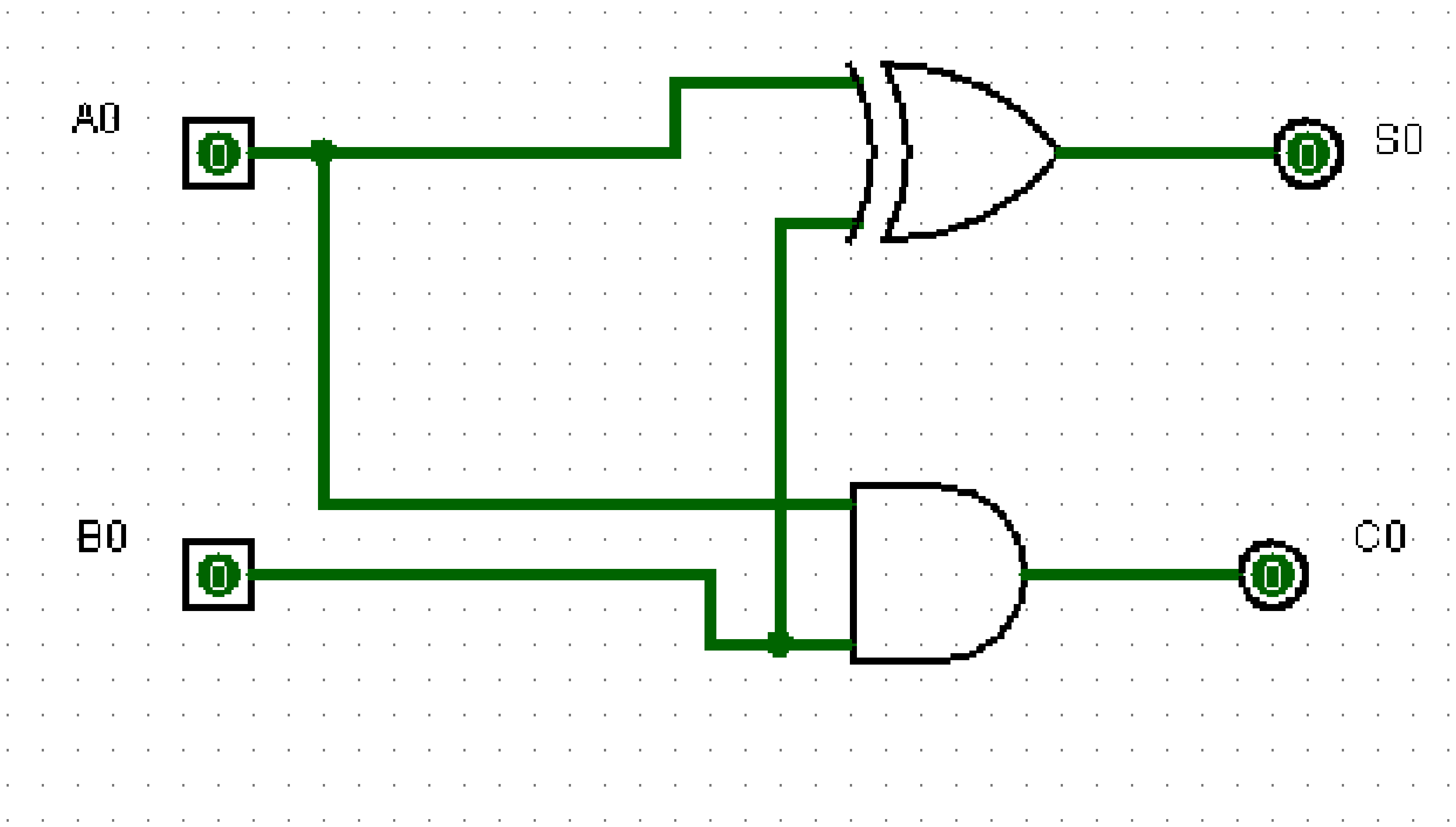


Timing Diagram

Timing Diagram

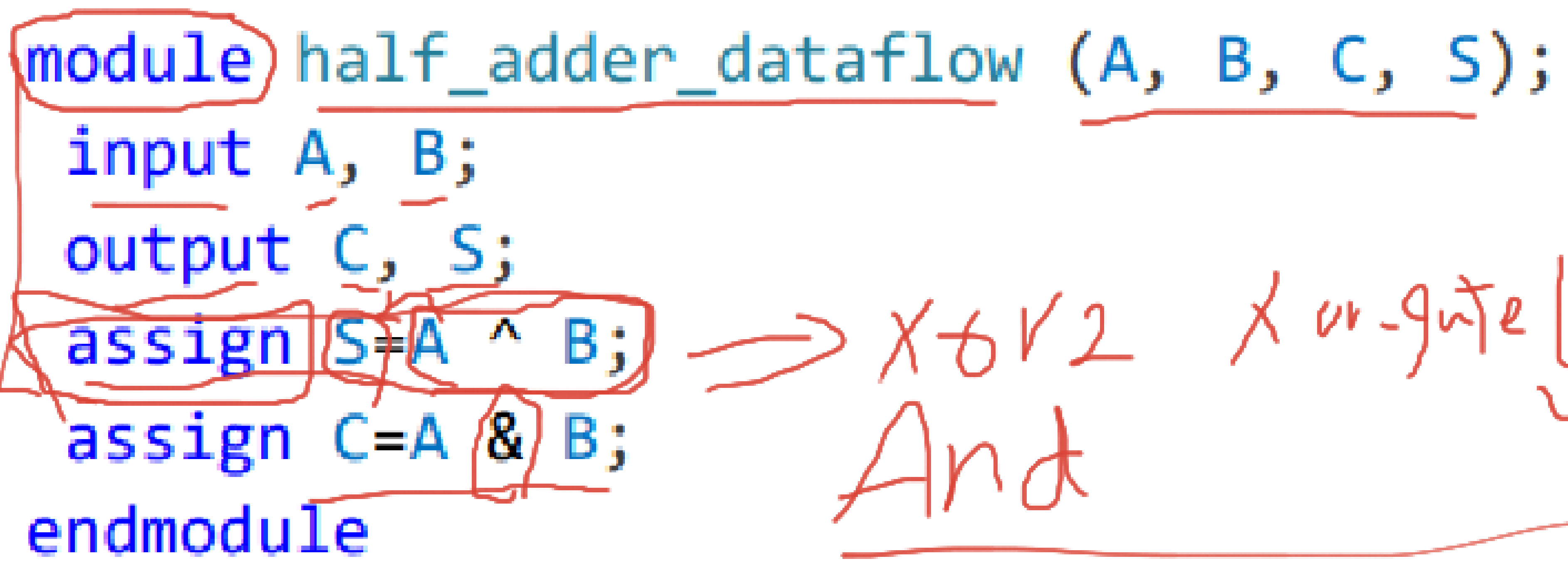


Logisim



Verilog

module Xor2 (



```
module half_adder_dataflow (A, B, C, S);
 input A, B;
 output C, S;
 assign S=A ^ B;
 assign C=A & B;
endmodule
```

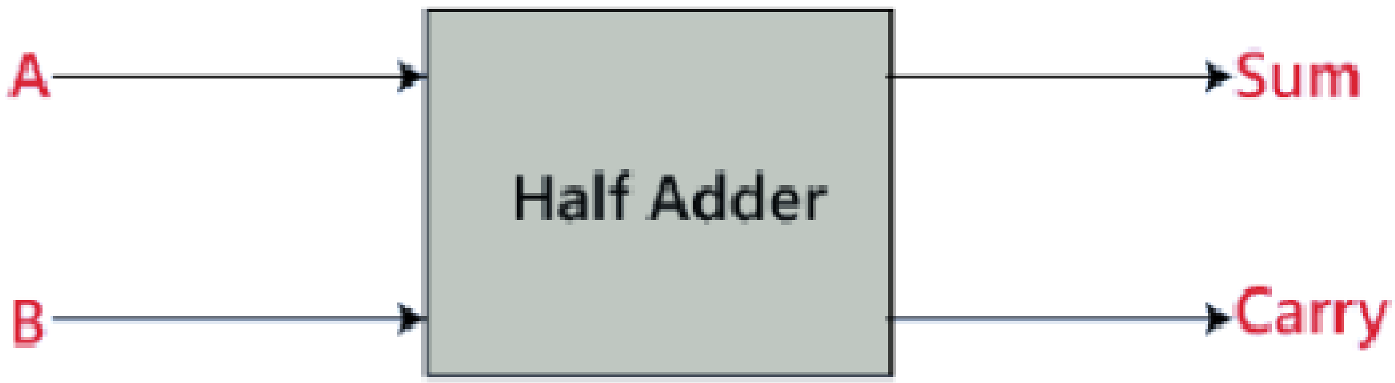
Xor2 Xor-gate(-,-)

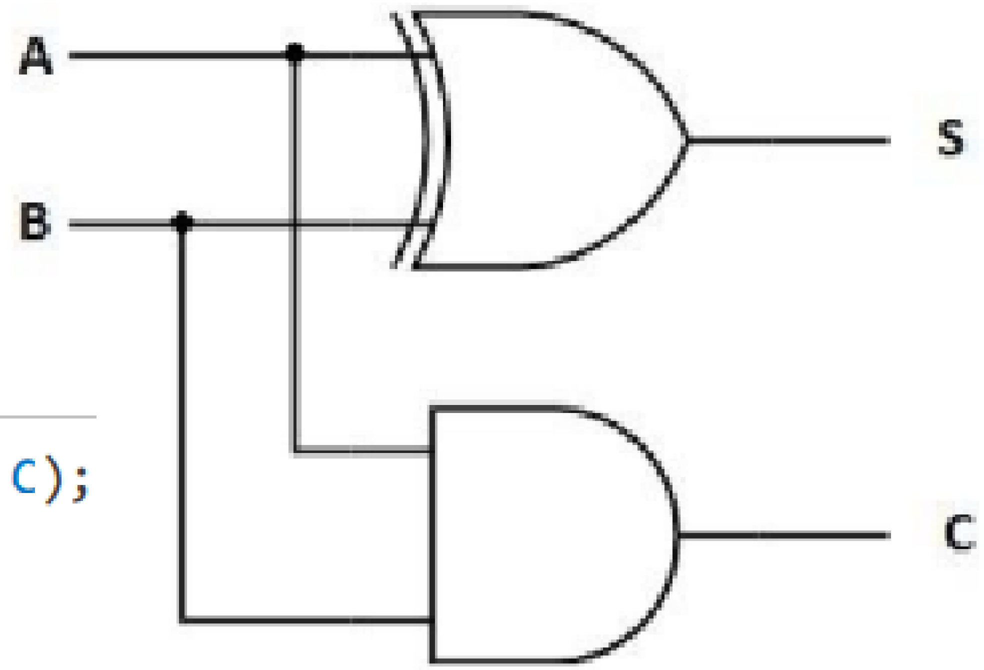
And

1- Structural

2- Behavioral

3- Data Flow





HA

①structural

```
module half_adder_gate (A, B, C, S);
 input A, B;
 output C, S;
 xor g1(S, A, B);   ✓
 and g2(C, A, B);   ✓
endmodule
```

②data flow

```
module half_adder_dataflow (A, B, C, S);
 input A, B;
 output C, S;
 assign S=A ^ B;
 assign C=A & B;
endmodule
```

③Behavioral

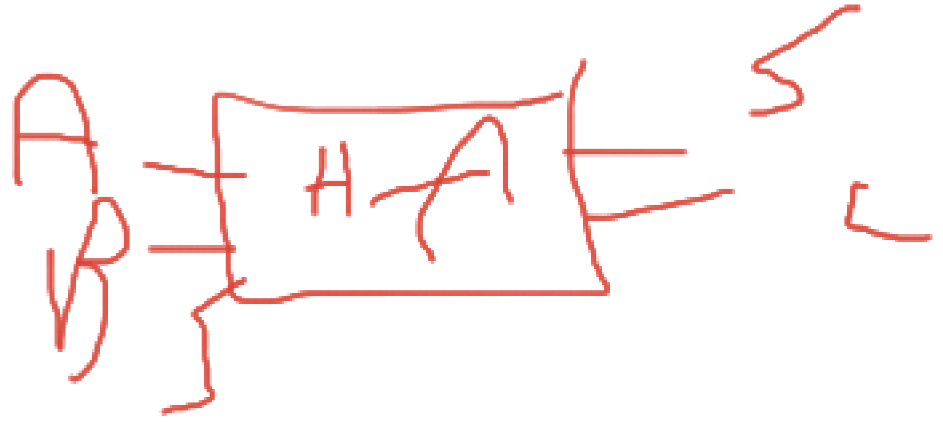
```
module half_adder_behavioral (A, B, S, C);
 input A, B;
 output reg S, C;

  always @ (*)
   begin

   case ({A, B})
   3'b00: S = 0;
   3'b01: S = 1;
   3'b10: S = 1;
   3'b11: S = 0;
   endcase

   case ({A, B})
   3'b00: C = 0;
   3'b01: C = 0;
   3'b10: C = 0;
   3'b11: C = 1;
   endcase
   end
endmodule
```

2
2

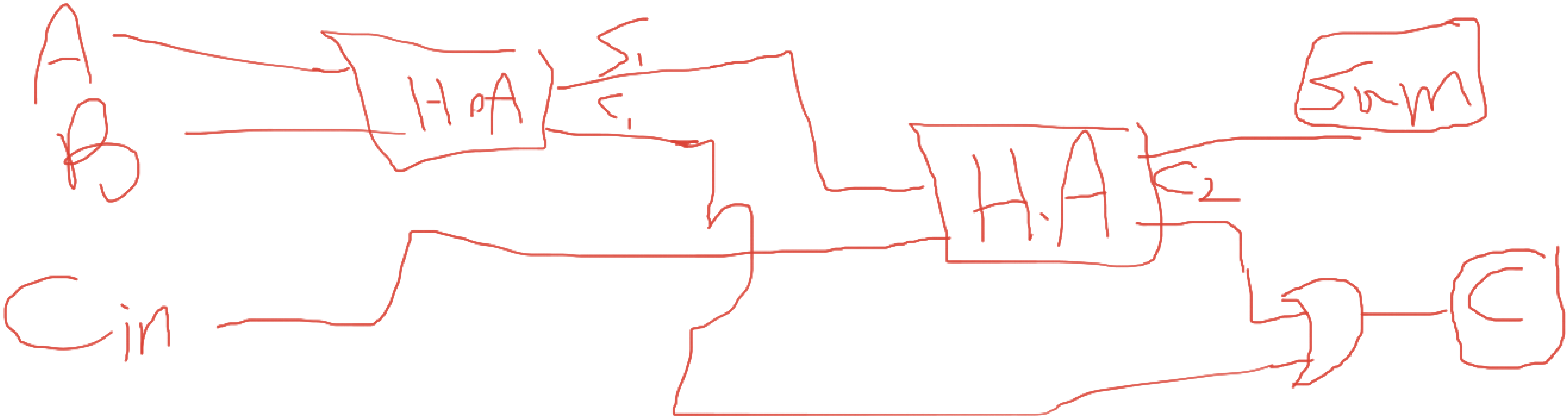
AB = 61 26

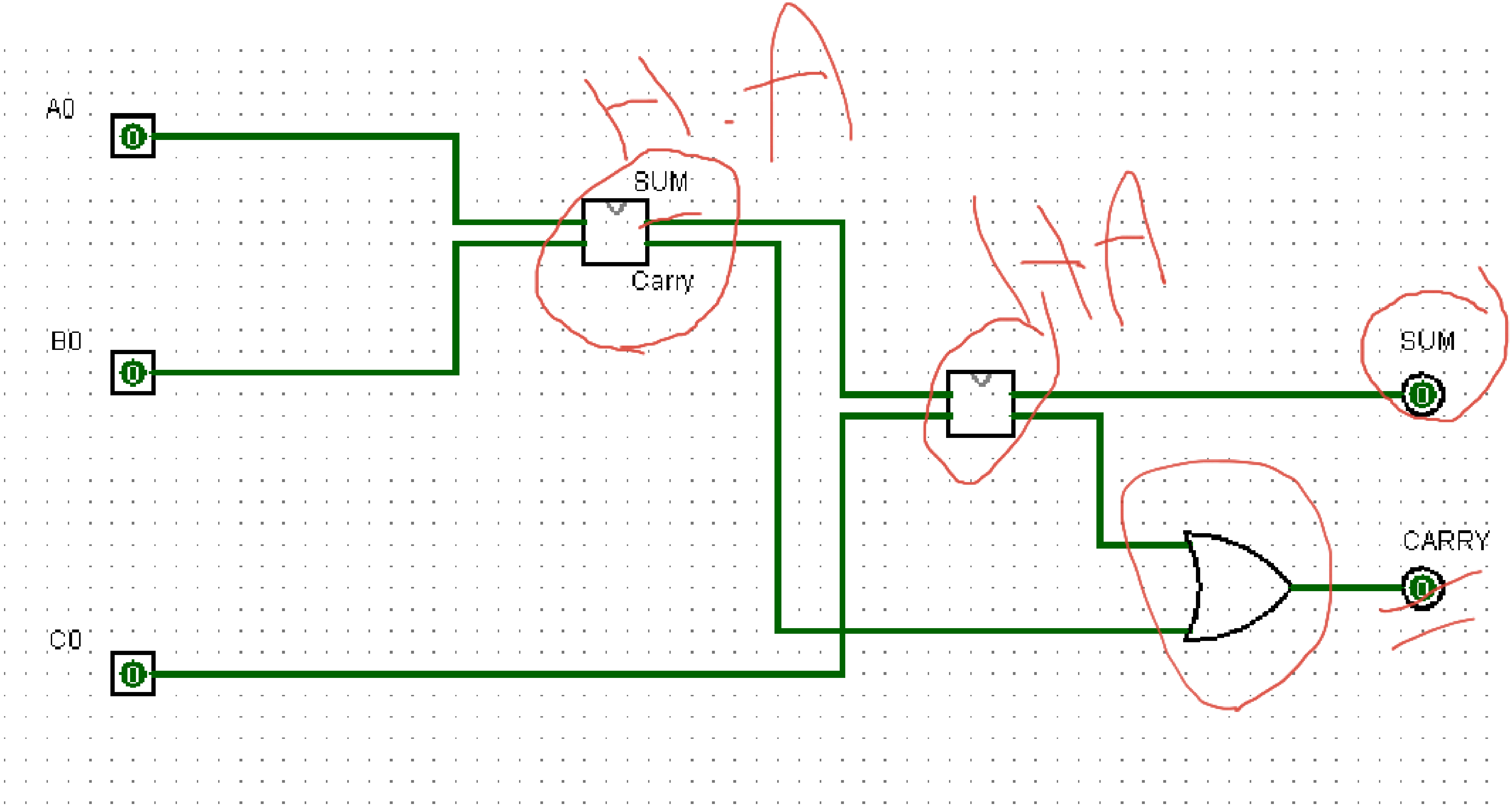
Testing plan

A	B	Cin	S	C
			✓	✓



Full Adder using Half Adder

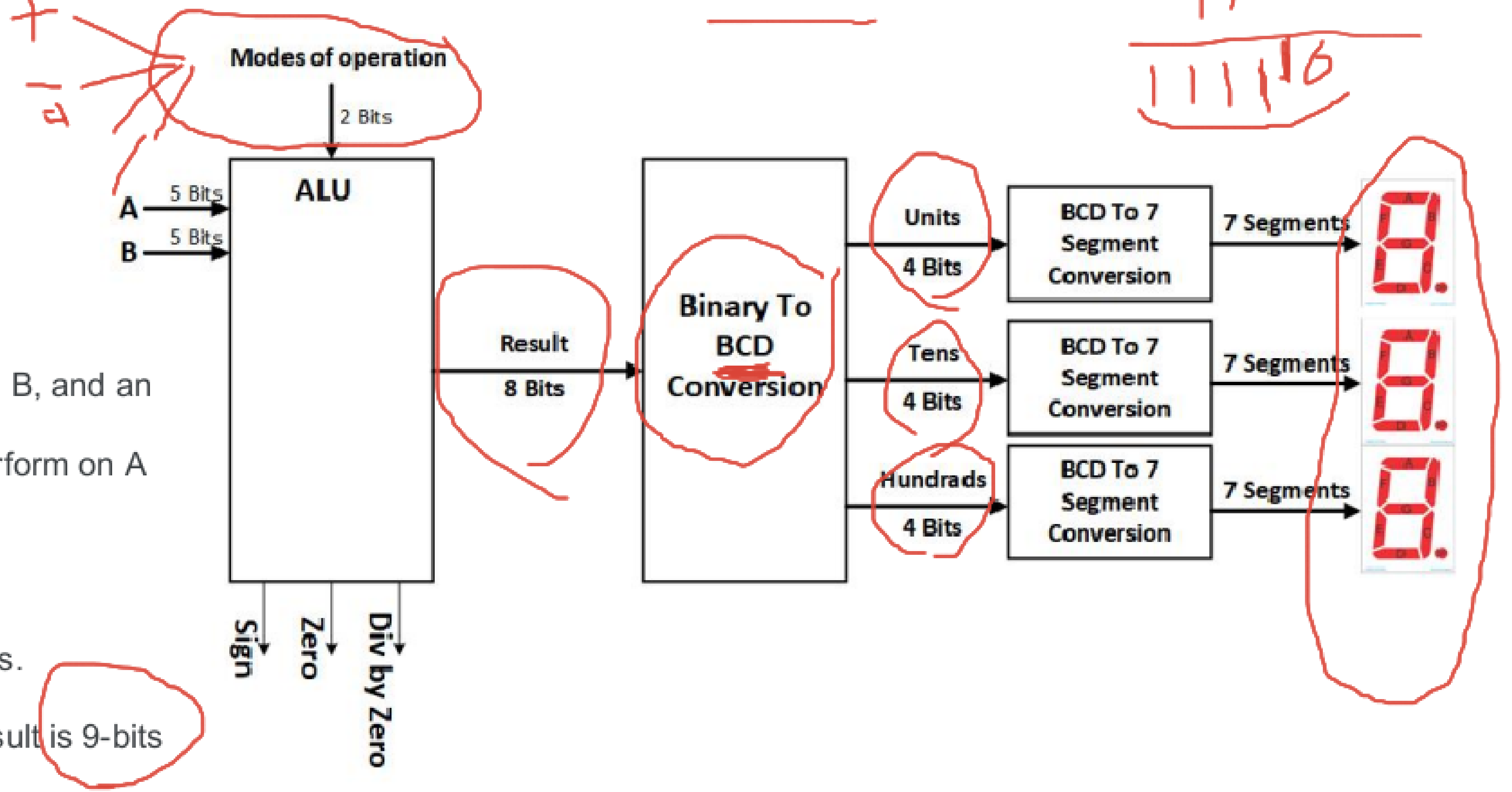



H.A
H.A
A0
0
B0
0
C0
0
SUM
Carry
SUM
0
CARRY
0

You are required to design and implement an arithmetic unit that is capable of adding, subtracting and multiplying two signed magnitude numbers, and displays the result of the operation performed along with some additional flags regarding the operation and the result.

1 4

111110



Description

The arithmetic unit takes two 5-bits signed magnitude inputs, A and B, and an additional input called Mode of Operation, which informs the arithmetic unit which function to perform on A and B:

☐ Addition: Result = A + B

During the addition A, B and Result are all 6-bits signed numbers.

☐ Subtraction: Result = A - B

During the subtraction A, B and Result are all 5-bits signed numbers.

☐ Multiplication: Result = A * B

During the multiplication A and B are 5-bits signed number and Result is 9-bits signed number.

The multiplication of 4-bits by 4-bits yields a result of 8-bits, therefore Result is composed of 8-bits for the value and 1-bit for the sign.

The multiplication of 15 x -15 = -225, which in binary is $(01111)_2$ x $(11111)_2$ = $(111100001)_2$

☐ Division: Result = A / B

During the division A, B and Result are all 5-bits signed numbers.

Our division is integer division, so we neglect the fraction part of the result.

The division of 15 / -2 = -7, which in binary is $(01111)_2$ / $(10010)_2$ = $(10111)_2$

Block Diagram (15 min)

Flags

o Sign Flag:

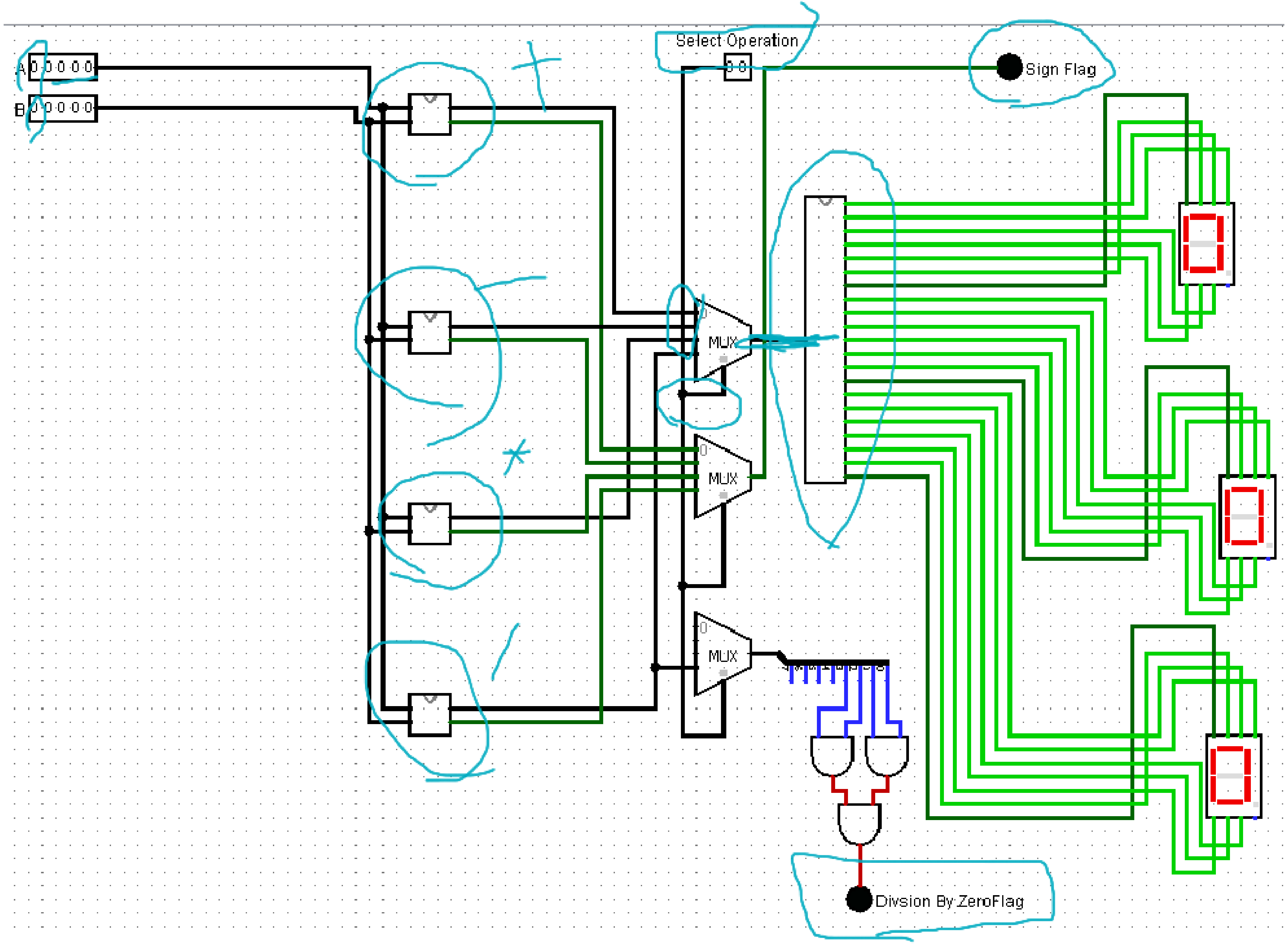
The sign flag indicates if the result is negative. The flag is set to 1 if the result is negative and 0 otherwise.

o Zero Flag:

The zero flag indicates if the result is zero. The flag is set to 1 if the result is zero and 0 otherwise.

o Div by Zero Flag:

The divide by zero flag indicates if we divide by zero. The flag is set to 1 if B operand equal zero in division operation and 0 otherwise.


A 0 0 0 0 0
B 0 0 0 0 0
Select Operation
0 0
Sign Flag
+
-
*
/
MUX
MUX
MUX
Divsion By ZeroFlag