Dylan Thornburg

1. $S_{out} = A \oplus B \oplus C_{in}$ $C_{out} = C_{in}(A+B) + AB$

A	B	C_{in}	S_{out}	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

S_{out}

C_{in} \ AB	00	01	11	10
0	0	1	0	1
1	1	0	1	0

C_{out}

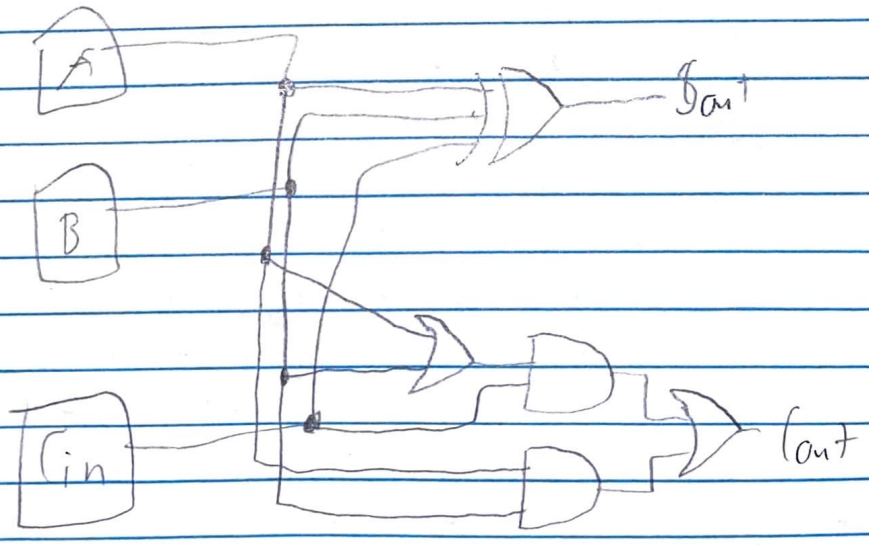
C_{in} \ AB	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$A \oplus B \oplus C_{in}$

$C_{in} \cdot B + C_{in} \cdot A + AB$

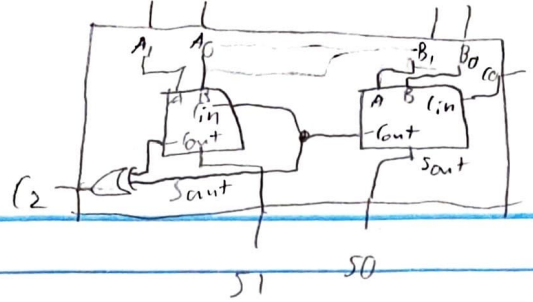
$C_{in}(A+B) + AB$

2.



$A_1 B_1 = 11$
$A_0 B_0 = 01$

3a.



3b. each adder has 6 gates; 1 xor, 2 and, 1 or
and there's one xor to check the carry, so $2 \times 4 + 1 = 9$
13 gates

4.
```
module myfulladd(A, B, Cin, S, Cout);
    input A, B, Cin;
    output S, Cout;
    assign S = (A ^ B ^ Cin);
    assign Cout = ((A&B) | ((Cin & (A|B))));
endmodule
```

5.
```
module myadder_two (A1, A0, B1, B0, C0, S1, S0, C2);
    input A1, A0, B1, B0, C0;
    output S1, S0, C2;
    wire C1;

    myfulladd u0 (A0, B0, C0, S0, C1);
    myfulladd u1 (A1, B1, C1, S1, C2);
endmodule
```

6. All outputs would be 0,
The outputs should just stay the same.
I would make the inputs all work so every carry = 1 to test it.
I'd look at the sum.

By testing all of these cases, I think it would be satisfactory to say the circuit works