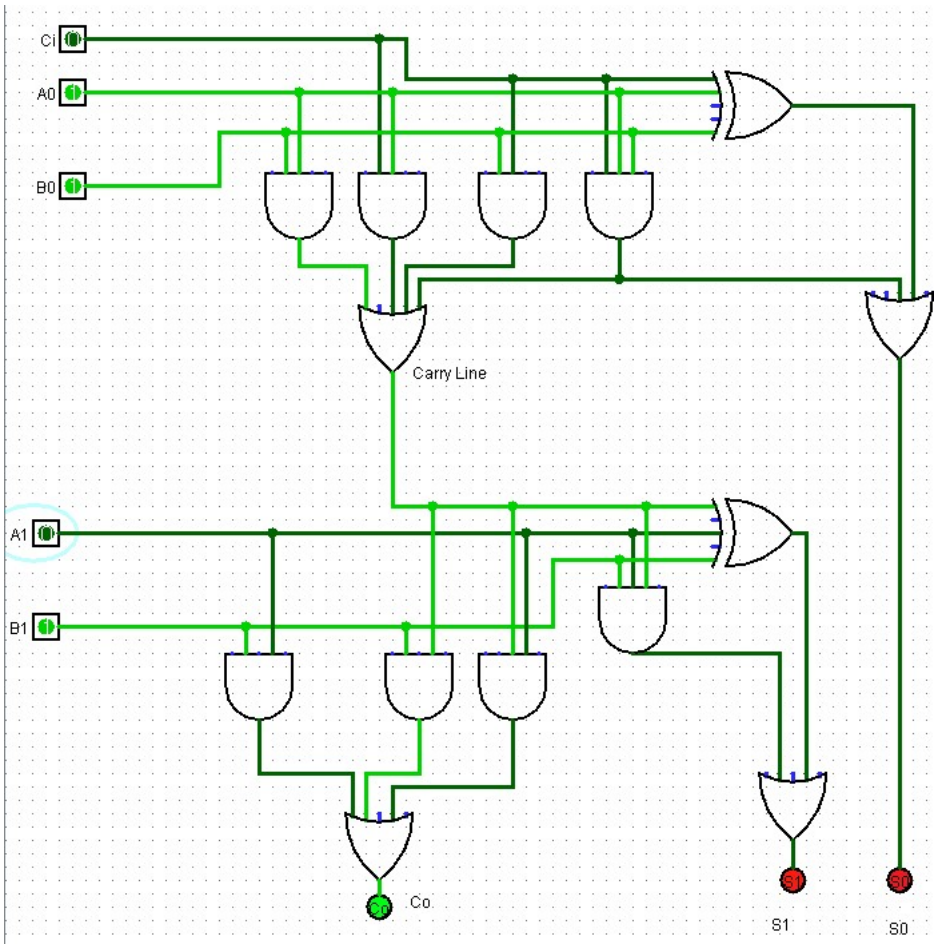
1**. Using only logic gates, design a 2-bit full adder with carry. Here is a partial truth table for the circuit.**

Draw a schematic showing the gate interconnections. Include either a boolean equation or an explanation of your design that matches the schematic you submit.



I know this isn't properly reduced but my digital logic is a little rusty and it works perfectly. (Please do not mark me off because it doesn't match the book! The logic is correct!)

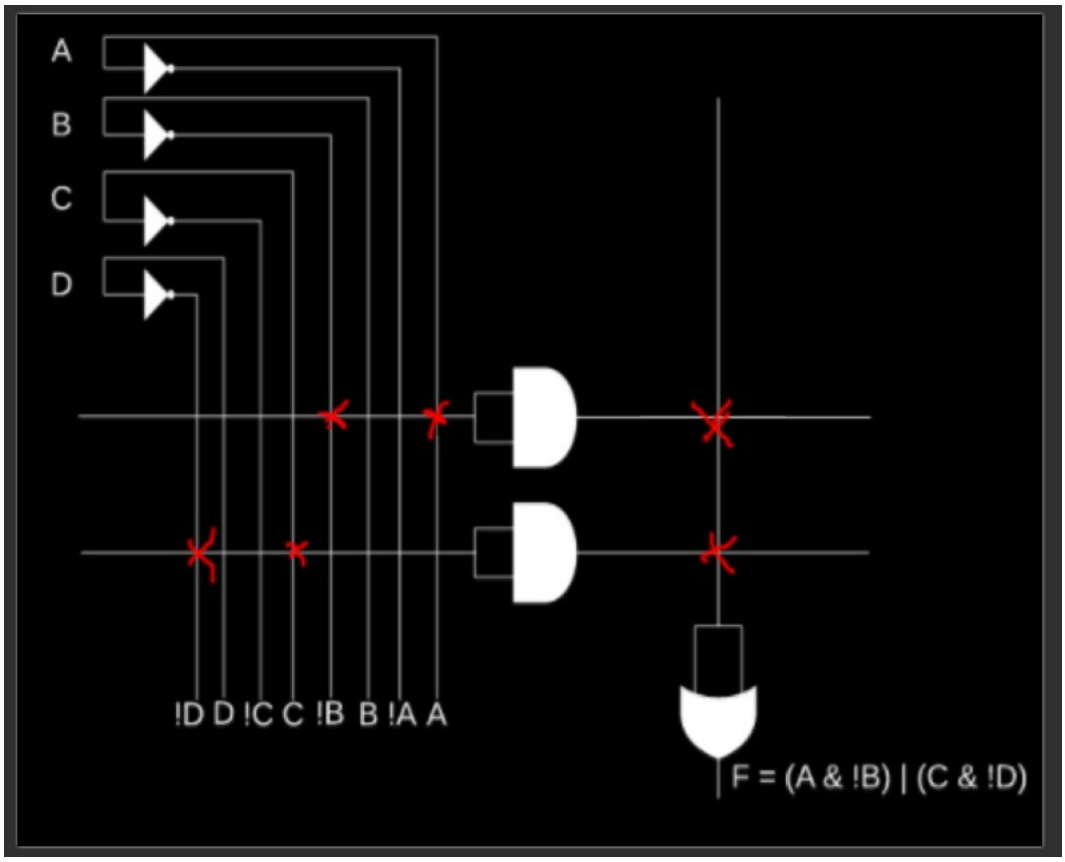
The four AND gates in each adder account for the carry situation. If any of them are active the OR gate beneath them will execute a carry. The XOR gates near the top of the adders accounts for the fact that there is only one situation where more than one input can be active and still get an active sum bit.

The fourth AND gate and accounts for the special case that all bits and carry are active.

The OR gates wired directly to the SUM pins accounts for the two different cases that you may have an active sum bit.

2. Show how the logic equation (A AND NOT(B)) OR (C AND NOT(D)) can be implemented using the following:

A. The PLA shown here:



B. The LUT shown here:

RAM CONTENTS (A AND NOT(B)) OR (C AND NOT(D))				Output Data
A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Also provide an explanation for your design choices.

The selected fuses on the PLA connect to one of the two AND gates that translate directly to the ANDs from the logic. These are then wired to the OR so that if either AND gate is true the sum of the PLA will also be true.

The logic table is even easier. Just place a 1 for any row A & B that evaluates to A & NOT-B then do the same for C & D (C & NOT-D) all the remaining cells in column F are zero.