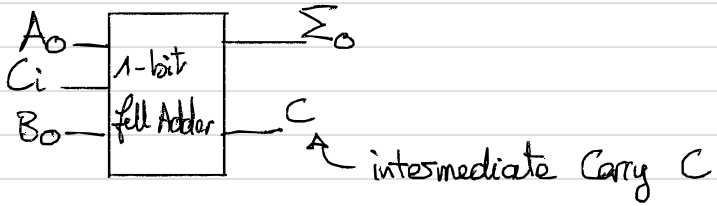
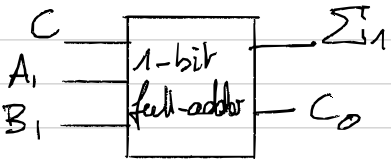
(1) The 2 bit addition can be viewed as:

$$\begin{array}{r} C \quad C_i \\ A_1 \quad A_0 \\ +\ B_1 \quad B_0 \\ \hline C_0 \quad \Sigma_1 \quad \Sigma_0 \end{array}$$

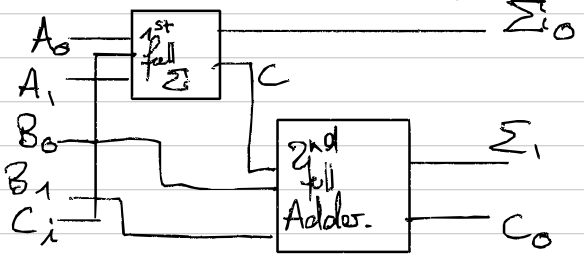
- In terms of using 1-bit full adder this would look like:

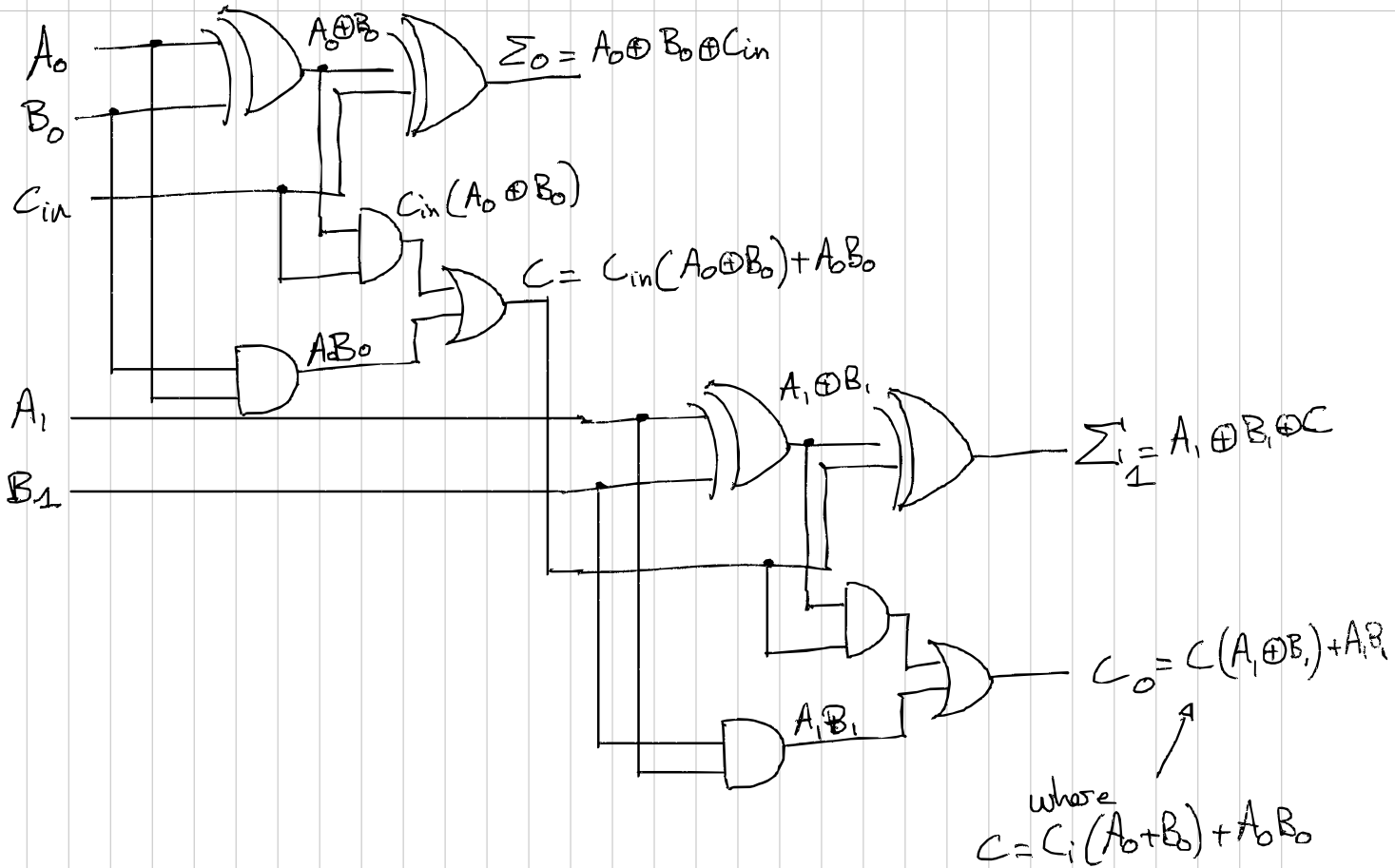


- The intermediate Carry - C can be fed into a 2nd 1-bit full adder like shown below:

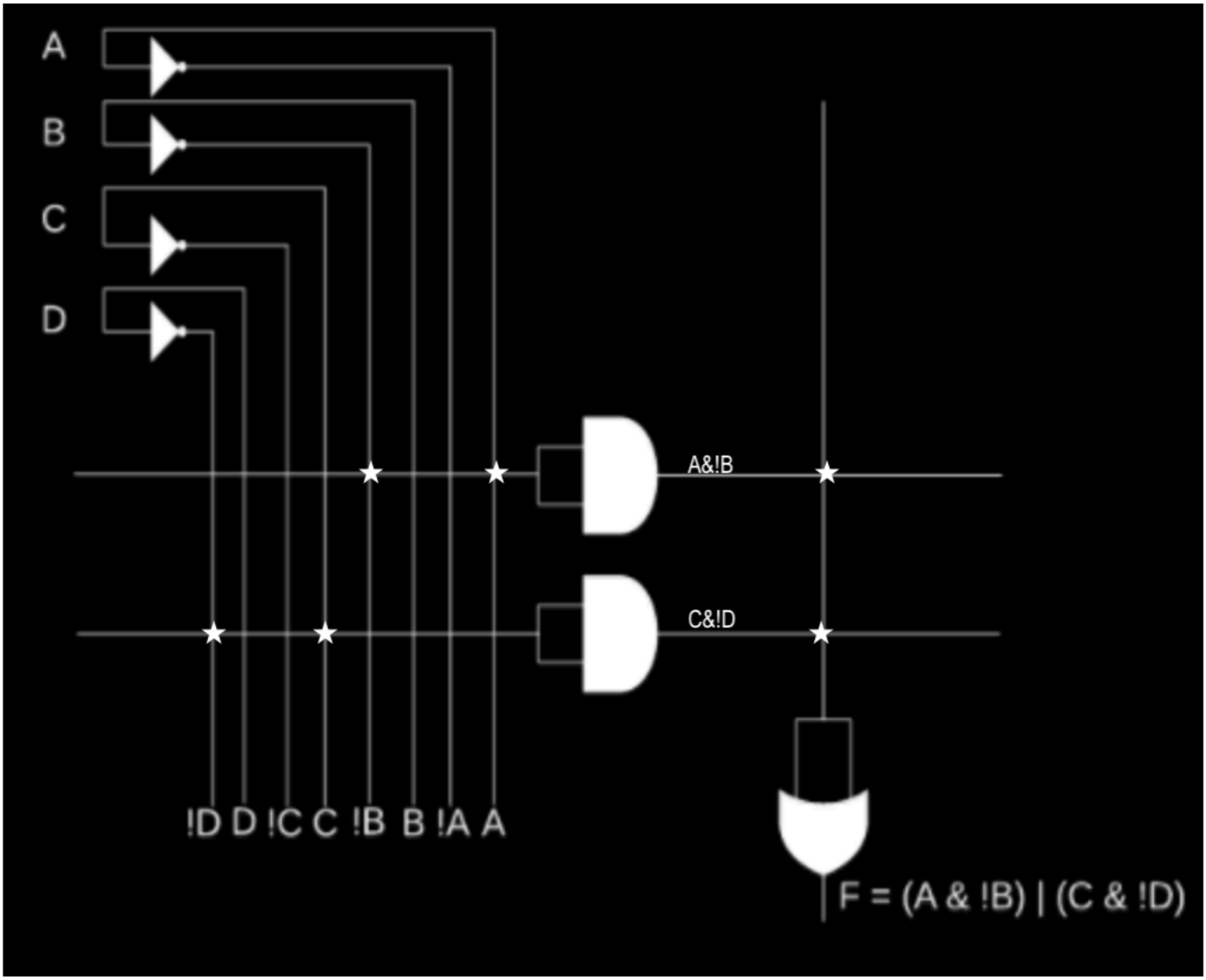


- Putting the 2 full adder together we have:




A_0
B_0
C_{in}
$A_0 \oplus B_0$
$\Sigma_0 = A_0 \oplus B_0 \oplus C_{in}$
$C_{in}(A_0 \oplus B_0)$
$C = C_{in}(A_0 \oplus B_0) + A_0 B_0$
$A_0 B_0$
A_1
B_1
$A_1 \oplus B_1$
$\Sigma_1 = A_1 \oplus B_1 \oplus C$
$C_0 = C(A_1 \oplus B_1) + A_1 B_1$
$A_1 B_1$
where
$C = C_i(A_0 + B_0) + A_0 B_0$

(2) A) We fuse !B and A before the first & gate. And we fuse !D and C before the second one.
Then we fuse both connect° before the OR Gate



B) In the first to columns we look for A &! B and obtain 4 ones in the output. Similarly the look for C but not D on the two following colums.

RAM CONTENTS				
Address				Output Data
A	B	C	D	F
0	0	0	!D 0	0
0	0	0	1	0
0	0	C 1	!D 0	1
0	0	1	1	0
0	1	0	!D 0	0
0	1	0	1	0
0	1	C 1	!D 0	1
0	1	1	1	0
1	0	0	!D 0	1
1	0	0	1	1 A&!B
A 1	0	C 1	!D 0	1
1	0	1	1	1
1	1	0	!D 0	0
1	1	0	1	0
1	1	C 1	!D 0	1
1	1	1	1	0

!B