adding 2 bits $s_1 s_0 = b_0 + b_1$

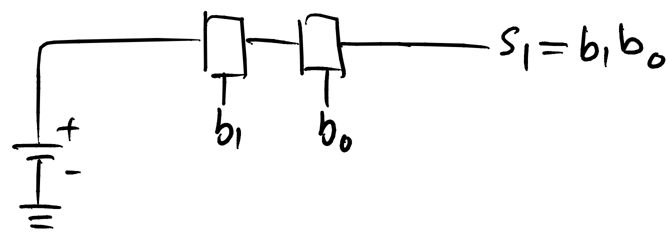
Truth Table

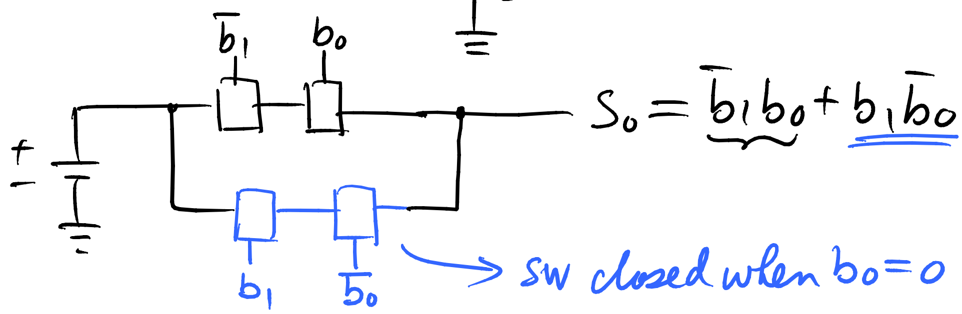
b_1	b_0	s_1	s_0
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$s_0 = \bar{b}_1 b_0 + b_1 \bar{b}_0$

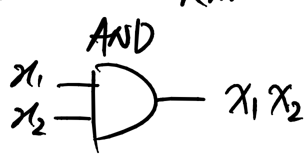
$s_1 = b_1 b_0$

Logic circuit:

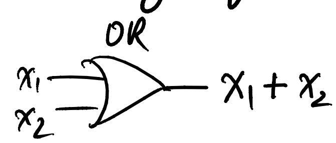




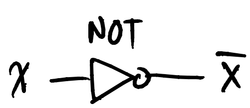
This is awkward to draw. we'll use logic gates (AND, OR, NOT...)



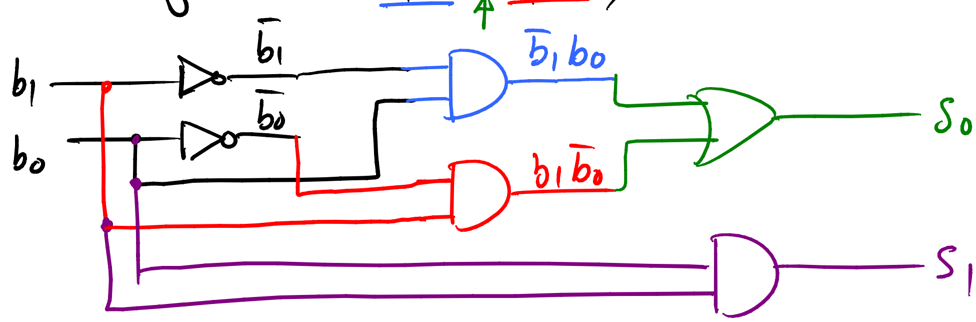
x_1	x_2	AND
0	0	0
0	1	0
1	0	0
1	1	1



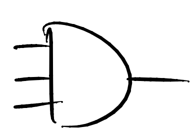
x_1	x_2	OR
0	0	0
0	1	1
1	0	1
1	1	1



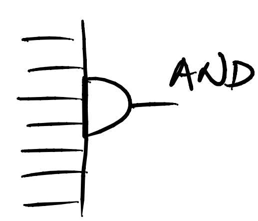
2-bits adding $S_0 = \bar{b}_1 b_0 + b_1 \bar{b}_0$ / $S_1 = b_1 b_0$

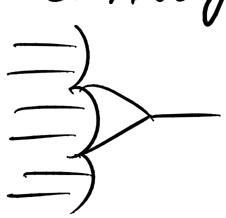


3-variable AND gates:



, more inputs





OR gate

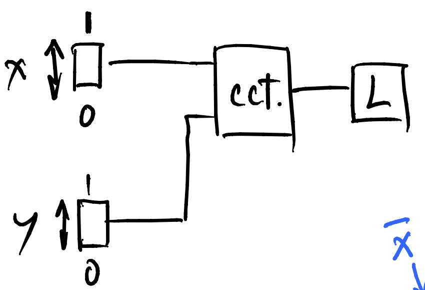
x_1	x_2	x_3	OR
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

← 3-input OR gate truth table

AND
0
0
0
0
0
0
0
1

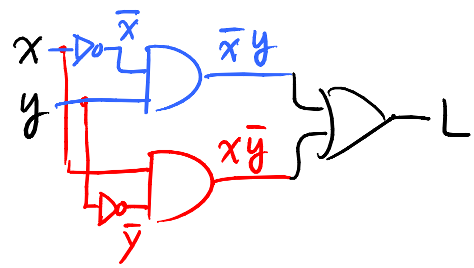
← 3-input AND gate

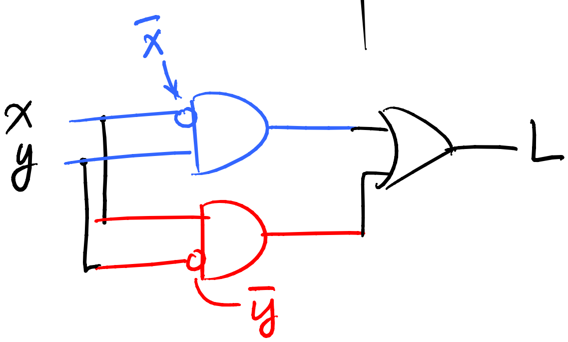
Design example: we have two slider switches that control a light. Sw. are called x, y. If both sw are down ($x=0, y=0$), then the light (L) is off ($L=0$). If one sw is up, L is on, but if both sw are up, then L is off.



x	y	L
0	0	0
0	1	1 → $\bar{x}y$
1	0	1 → $x\bar{y}$
1	1	0

$$L = \underline{\bar{x}y} + \underline{x\bar{y}}$$





XOR gate