

# Sistemas Computacionais - FULL ADDER

MURILLO CARVALHO 11234673

Somador 1 BIT  $\Rightarrow$  XOR

$in_1$	$in_2$	Out	
0	0	0	(0+0 = 00) ↓
1	0	1	(1+0 = 01)
0	1	1	(0+1 = 01)
1	1	0	(1+1 = 10) ↑

A porta XOR funciona como somador p/ o bit 0, mas ainda precisa de combinação com outras portas para somar bits corretamente.

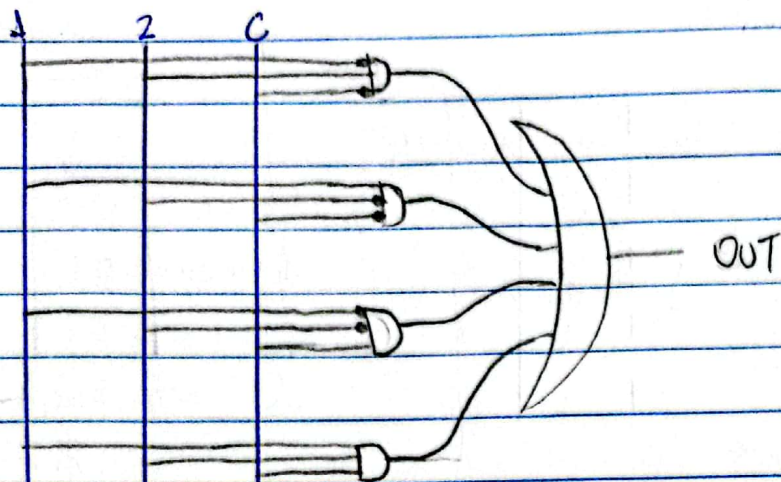
$\Rightarrow$  PARA FORMAR UM FULL-ADDER É NECESSÁRIO QUE ELE POSSA, ALÉM DE GERAR O out, permitir a entrada de um inc e a saída de um Outc que serão responsáveis por carregar o valor do próximo bit para o próximo full adder.

Somador (Full Adder)

$in_1$	$in_2$	$inc$	$Outc$	Out
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



FAZENDO POR SOMA DE PRODUTOS TEMOS:

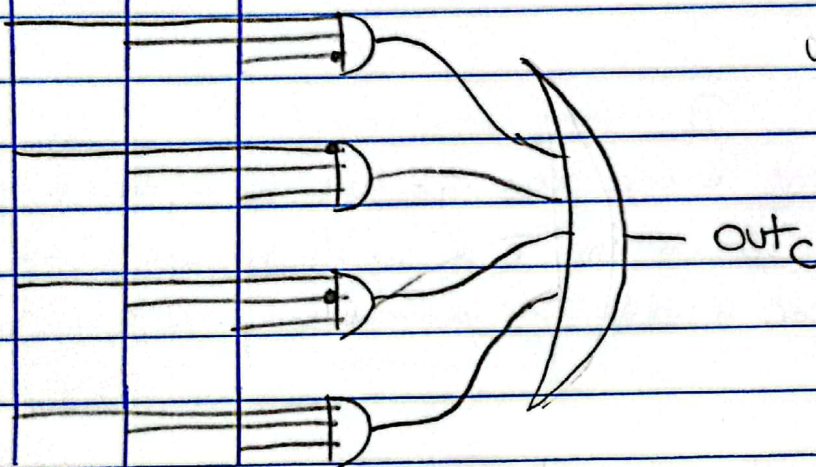


And 3 partes

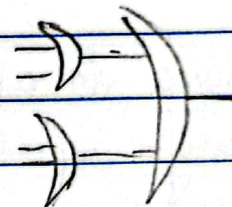
$\Rightarrow D \rightarrow D$

$$x \text{ e } y \text{ e } z = (x \text{ e } y) \text{ e } z$$

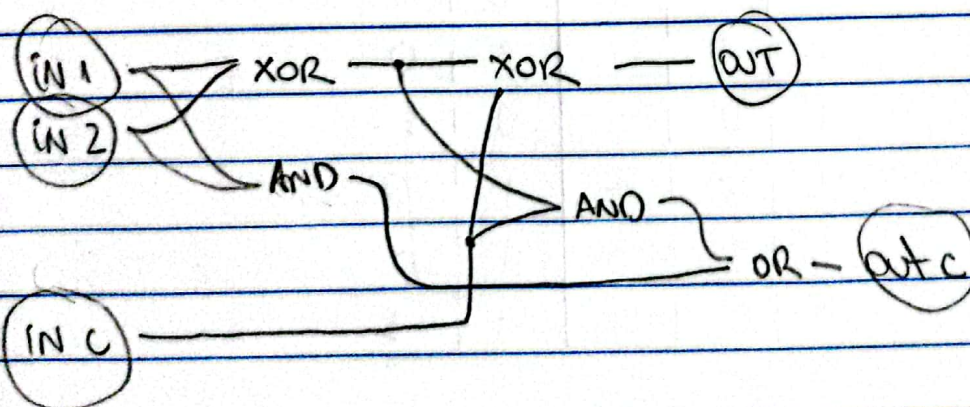
OR 4 partes



$$w | x | y | z = (w | x) | (y | z)$$

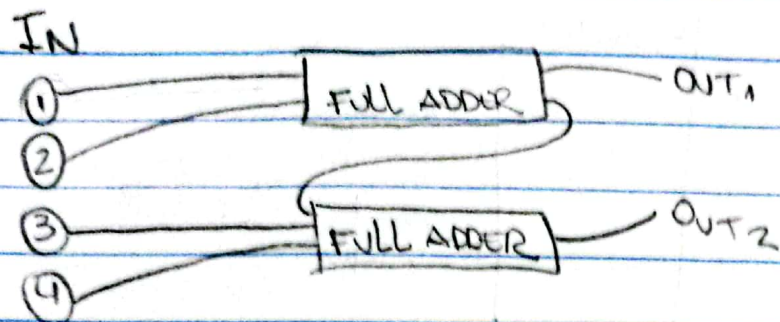


ESSE DESENHO PODE SER REDUZIDO PARA UM MODELO MAIS SIMPLES:

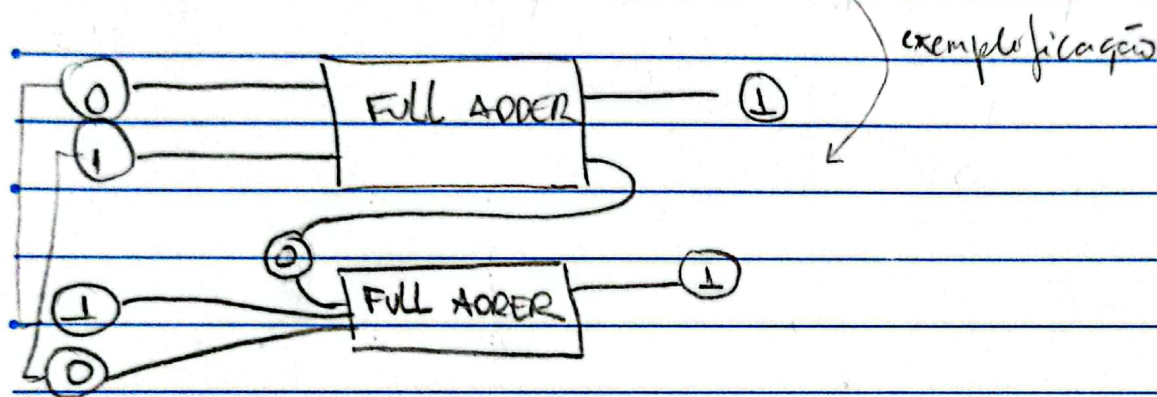




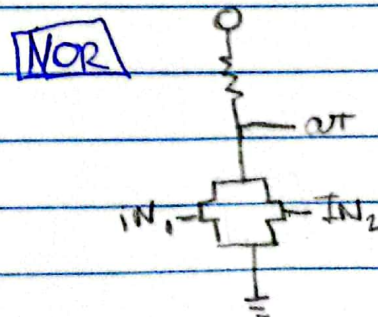
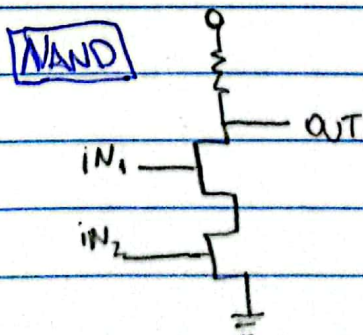
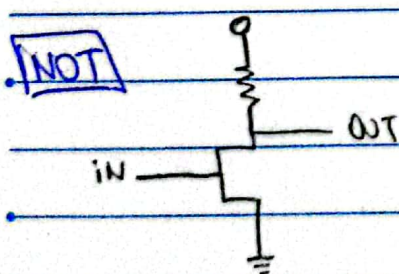
Full ADDER's podem ser combinados para formar somadores de mais bits



Supondo  $01 + 10 = 11$   
(ou decimal)  $(1 + 2 = 3)$



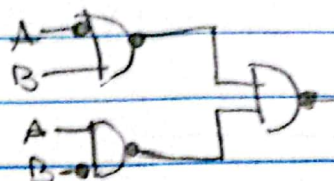
PARA FAZER OS somadores são ~~preciso~~ NECESSÁRIAS AS PORTAS LÓGICAS FORMADAS A PARTIR DE TRANSISTORES



As portas NAND e NOR podem ser diretamente combinadas com uma NOT para que possam formar AND e OR respectivamente



Quantidade de transistores utilizados: (módulo redimensionado de full adder)

OPERADOR	TRANSISTORES	QUANTID.	XOR
AND	3 (NAND + NOT)	2	
OR	3 (NOR + NOT)	1	
XOR	8 (3NAND + 2NOT)	2	

total = 25 transistores

Desta maneira é possível implementar um full adder com apenas 25 transistores.