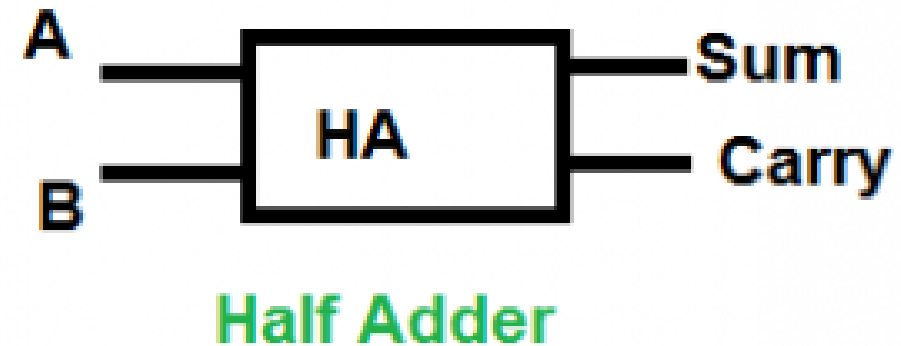
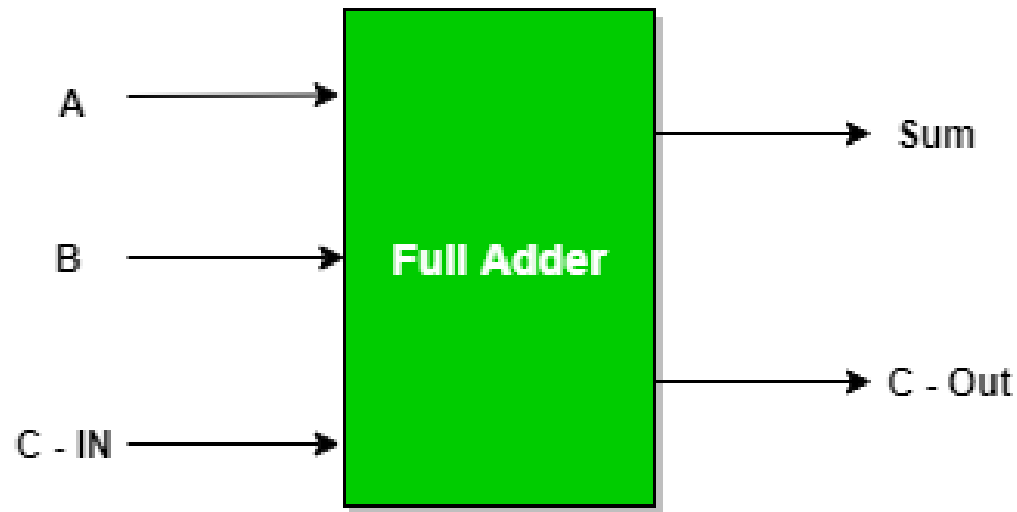


O q crlh é um adder?

Feito pela minha pessoa

- Como diz o nome, adder é um somador
- E existem varios tipos de adders, mas o mais importante é o full adder



Full adder

- O primeiro passo para perceber o que é um full adder, é fazer a tabela de verdade.
- Eu não sei se a ordem das variáveis interessa, mas o stora meteu nos slides o “carry-in” como o bit mais significativo, por isso vou seguir essa lógica.
- Antes de mostrar a tabela, vamos perceber o que é o carry in e o carry out.

Carry-In e Carry-Out

- Se olharmos para uma soma normal entre dois números binários:
-

	1	1	0	0	0	0	1
	0	0	1	0	1	1	0
+	0	1	1	0	0	0	0
<hr/>							
	1	0	0	0	1	1	0

Se olharmos para a **4ª soma**, reparamos que vem um 0 de tras. Este é o carry-in.

Ao realizarmos a soma, fica: $1 + 0 + 0(\text{carry-in}) = 1$.

Como $1 < 2$, não existe nenhum carry para a próxima soma.

Então, o carry-out é 0.

Tabela de verdade

- Agora que relembramos como fazer a soma, vamos fazer a tabela de verdade.
- Para tal, vamos ter 3 variaveis, que vao ser os inputs no quartus.
- Estas variaveis sao X, Y, e carry-in.
- X e Y sao os valores que nos queremos somar, e o carry-in ja foi explicado a tras.
- Nós temos que saber os resultados de todas as possibilidades.
- Entao, para tal, iremos ordenar os inputs como se fosse binario.

Tabela de verdade

- Como Y é bit menos significativo, de 1 em 1.
- X varia de 2 em 2, e o carry-in de 4 em 4.
- ISTO É BINARIO NORMAL para quem ainda nao percebeu.
- Os **outputs** vêm da soma dos inputs.
- Ou seja, $0+0+0 = 0$ e nao vai carry nenhum, entao $S = 0$ e $c_{out} = 0$.
- Assim sucessivamente.

C_{in}	X	y	C_{out}	S
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

Exercicio

- Façam voces a tabela de verdade. É mais facil perceber se forem voces a fazer, em vez de ler de um slide.
- Tentem nao ver a tabela atras.
- Lembrem-se que os inputs sao como binario.

Expressões

- Para nos ajudar no quartus, vamos fazer as expressões com a ajuda da tabela.
- Vamos usar os mintermos, ou seja, os valores em que o resultado é 1.
- E vamos fazer para tanto o S, como o carry-out

C_{in}	x	y	C_{out}	s
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

Soma (S)

- S é positivo em 1, 2, 4 e 7.

- Entao:

$$S = (cin' * x' * y) + (cin' * x * y') + (cin * x' * y') + (cin * x * y)$$

- Vamos reduzir a expressao:

- cin' e cin em evidencia:

$$(=) cin' (x' * y + x * y') + cin (x' * y' + x * y)$$

C_{in}	x	y	C_{out}	S	Mintermos
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	1	2
0	1	1	1	0	3
1	0	0	0	1	4
1	0	1	1	0	5
1	1	0	1	0	6
1	1	1	1	1	7

Soma (S) cont...

$$\text{cin}' (x' * y + x * y') + \text{cin} (x' * y' + x * y) \quad (=)$$

$$(\Rightarrow) \text{cin}' (x \text{ XOR } y) + \text{cin} (x \text{ XOR } y)' \quad (=)$$

$$(\Rightarrow) \text{cin} \text{ XOR } (x \text{ XOR } y)$$

- Esta é a expressão que vamos implementar no quartus, no guião 8.
- Agora vamos fazer o carry-out.

Carry-Out (cout)

- Mesma coisa, vamos usar os mintermos, ou seja, onde é 1 na coluna cout
- Neste caso, ao contrario do S, fica muito mais facil reduzir atravas do K map.
- Entao vamos fazer isso.

C_{in}	X	y	C_{out}	S
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

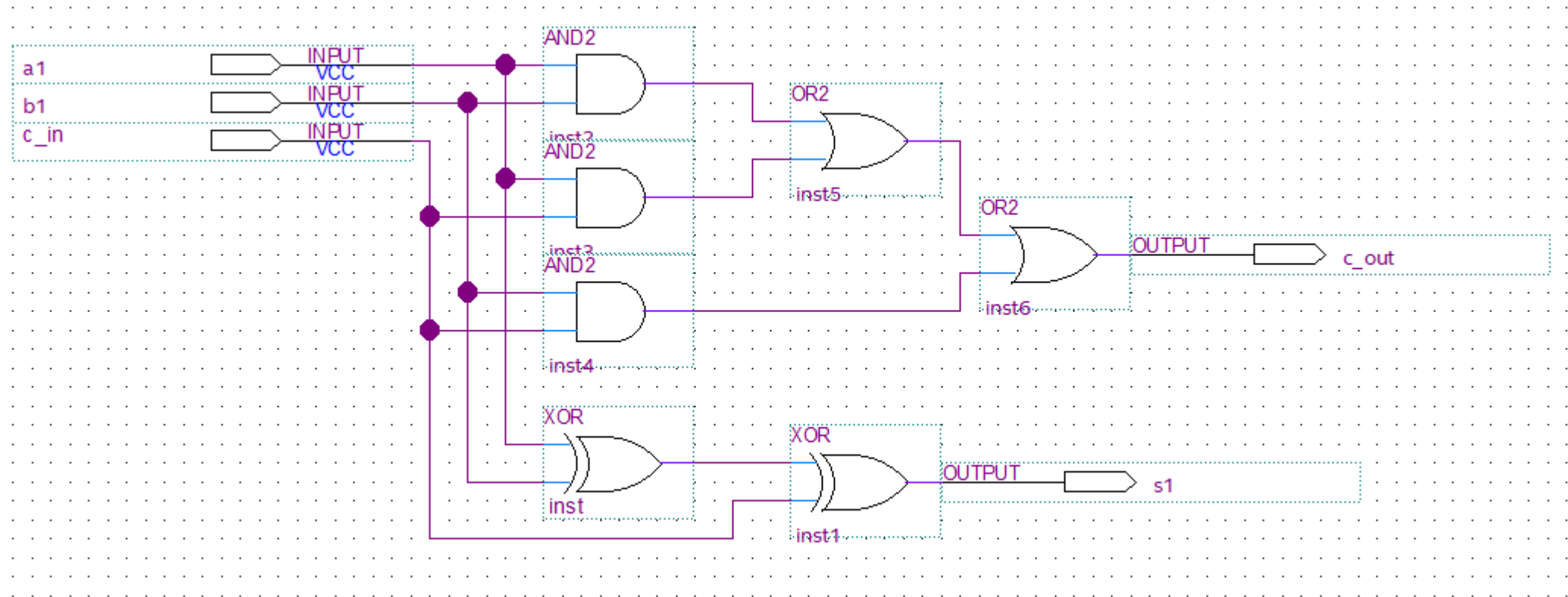
Redução através do mapa

cin\X Y	00	01	11	10
0			1	
1		1	1	1

- Então fica: $cout = (x * y) + (cin * x) + (cin * y)$
- Temos a próxima expressão para o Quartus
- Por falar em Quartus, vamos agora para lá, e vamos implementar as expressões que obtivemos.

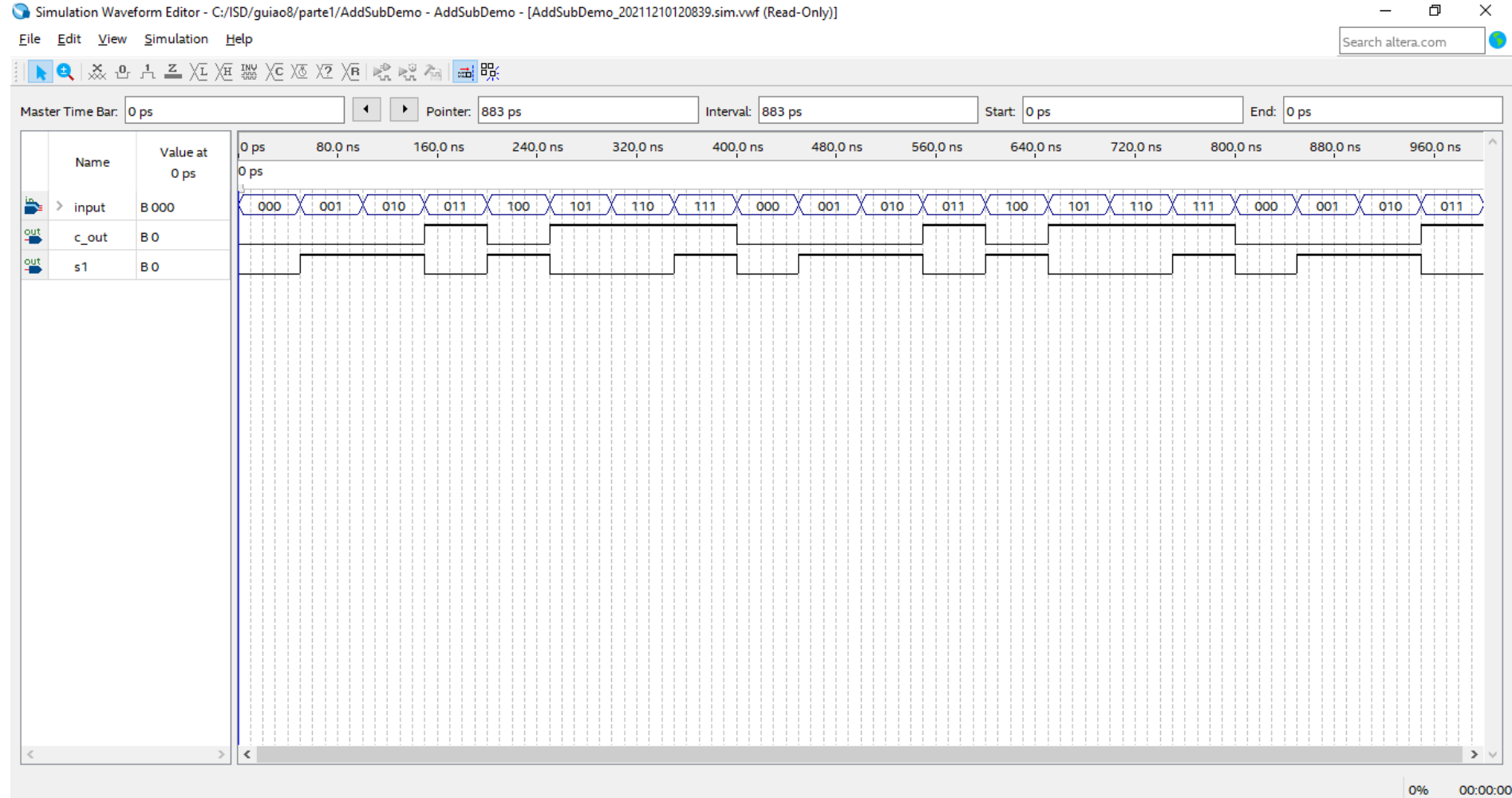
Quartus

- Todos nos fazemos coisas diferentes no Quartus, mas o circuito deve ficar algo assim:



Quartus

- E a simulação assim:



Ta feito

- Isto é um full adder.
- Existe muita mais materia, mas a sora so pedia a parte 1 do guia 8, entao estes slides sao suficientes por agora (e eu tambem nao estudei essa materia xD)
- Esta materia é muito facil de entender se forem voces a rabiscar no papel.
- Se precisarem de ajuda, @Daniel Pedrinho no discord.