

Trabalho 1 - Circuitos Digitais  
Nome: Gabriel Justus Ramos  
GRR: 20232348

A seguir, além de apresentar meu raciocínio, também mostrei algumas tabelas verdade e expressões booleanas que me ajudaram a montar os circuitos necessários.

Full Adder:

Tabela Verdade:

A	B	Cin	S	Cout
0	0	0	0	0
0	0	0	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	0	1
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

Derivando as expressões booleanas Cout e S usando a Forma Normal Disjuntiva, e em seguida simplificando-as:

$$\text{Cout} = A'.B.Cin + A.B'.Cin + A.B.Cin' + A.B.Cin$$

$$\text{Cout} = A'.B.Cin + A.B'.Cin + A.B.Cin' + A.B.Cin + A.B.Cin + A.B.Cin$$

$$\text{Cout} = B.Cin.(A' + A) + A.Cin.(B' + B) + A.B.(C' + C) \rightarrow \text{Distrib.}$$

$$\text{Cout} = B.Cin + A.Cin + A.B \rightarrow \text{Comp} + \text{Elem. Neutro}$$

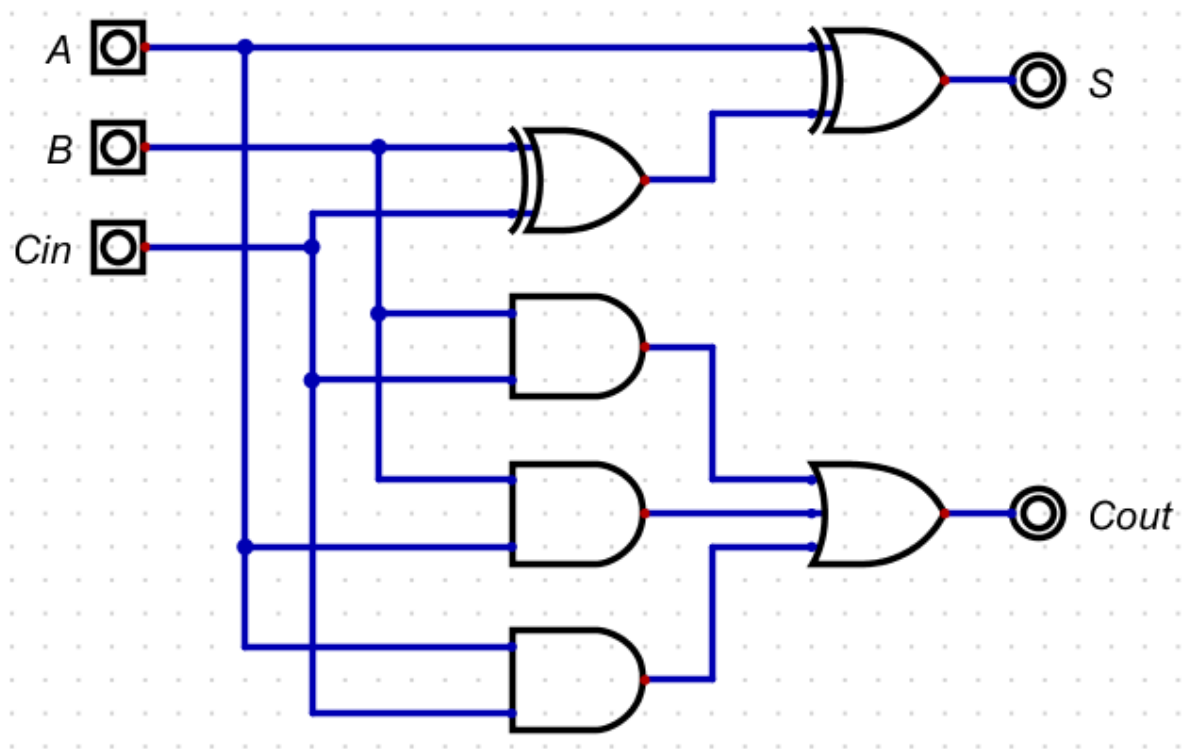
$$S = A'.B'.Cin + A'.B.Cin' + A.B'.Cin' + A.B.Cin$$

$$S = A'.(B'.Cin + B.Cin') + A.(B'.Cin' + B.Cin) \rightarrow \text{Distrib.}$$

$$S = A'.(B \oplus C) + A.(B \oplus C)' \rightarrow \text{XOR} + \text{XNOR}$$

$$S = A \oplus B \oplus C \rightarrow \text{XOR}$$

Usando as expressões Cout e S podemos montar um circuito para um Full Adder:



Half Adder:

Tabela Verdade:

A	B	S	Cout
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

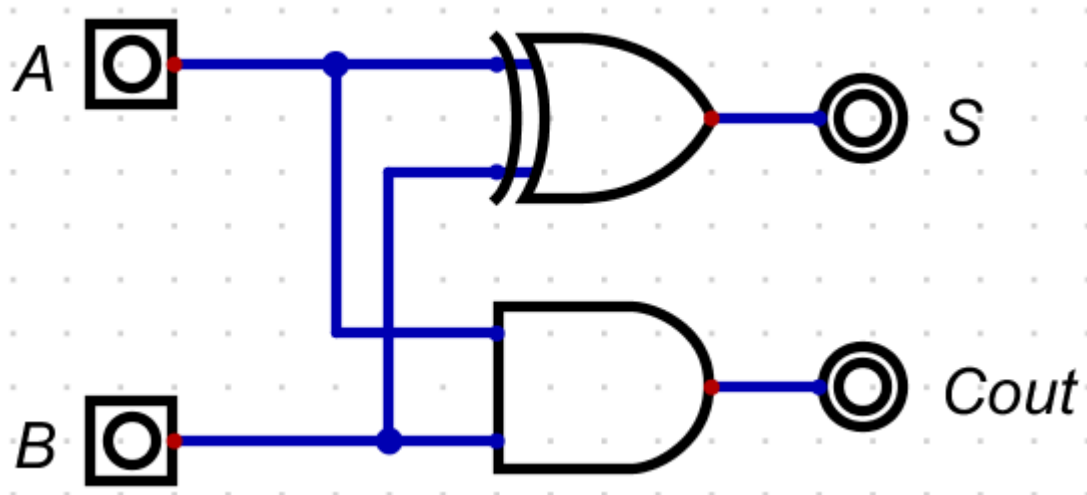
Derivando as expressões booleanas Cout e S usando a Forma Normal Disjuntiva, e em seguida simplificando-as:

$$Cout = A.B$$

$$S = A'.B + B.A'$$

$$S = A \oplus B \rightarrow \text{XOR}$$

Usando as expressões Cout e S podemos montar um circuito para um Half Adder:



Comparador:

Para compararmos dois números de 6 Bits, iremos comparar seus bits um a um.

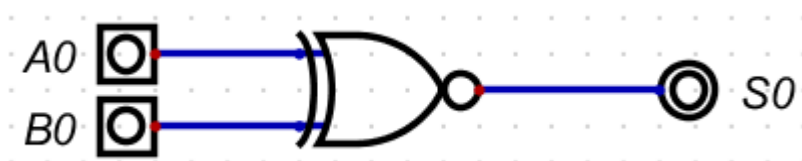
Tabela Verdade:

A	B	Comp
0	0	1
0	1	0
1	0	0
1	1	1

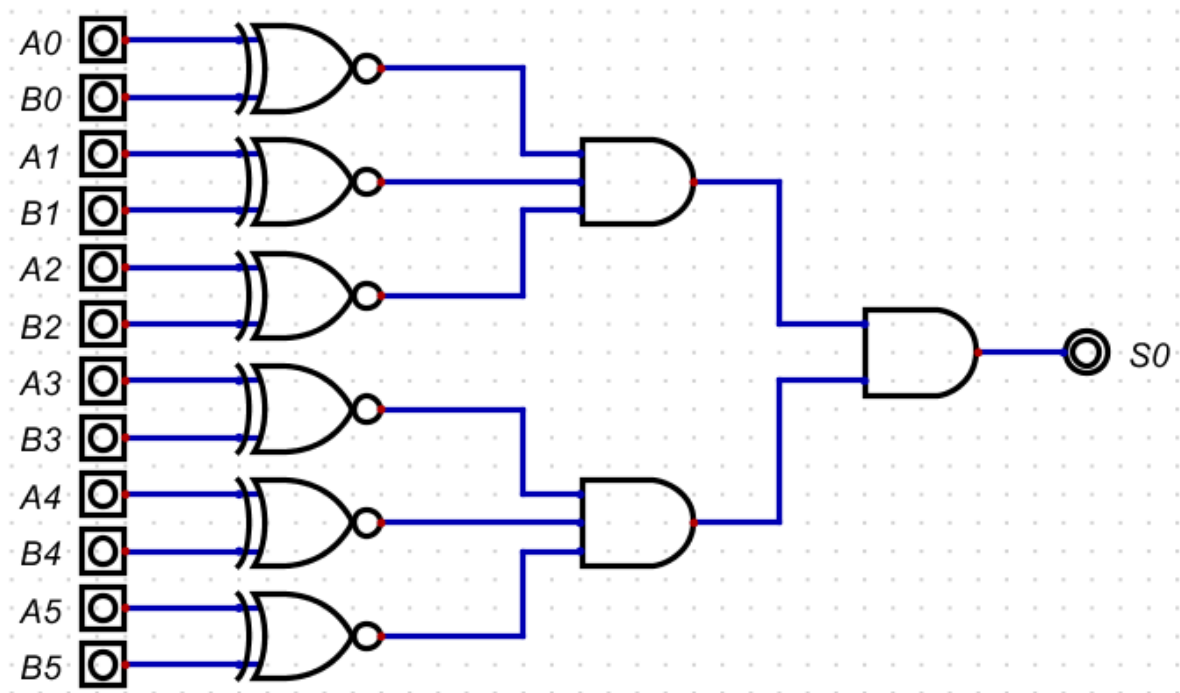
Derivando a expressão booleana Comp usando a Forma Normal Disjuntiva:

$$\text{Comp} = (A \oplus B)'$$

Portanto, o circuito para comparar 2 números de um bit é:



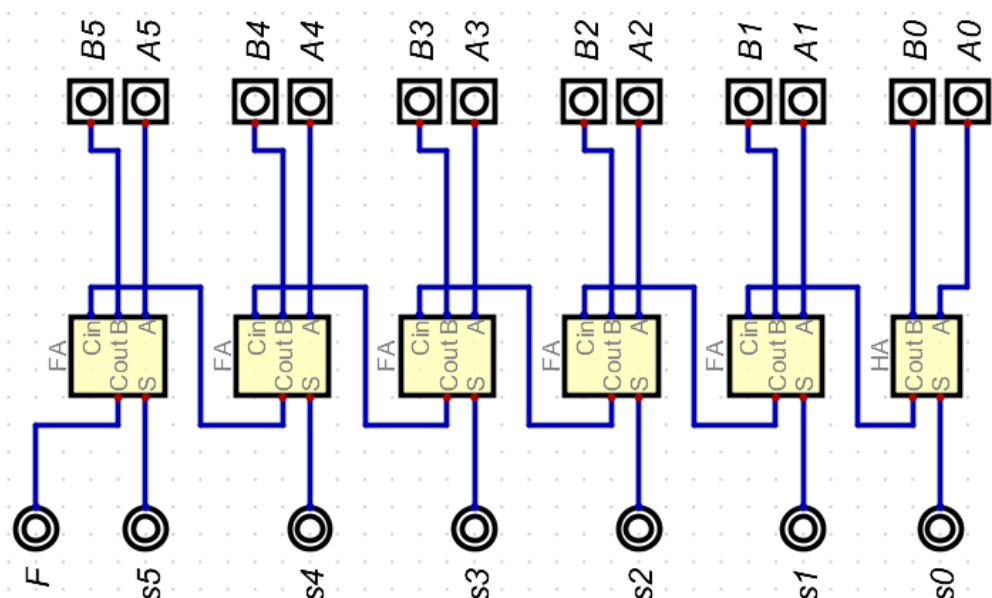
Porém, como desejamos comparar 2 números de 6 bits, e como a resposta impressa deve ser apenas um 1, caso sejam iguais, e apenas um 0, caso sejam diferentes, o circuito fica:



Somador:

Para construir o somador, utilizei 5 Full Adders e um Half Adder. Eu usei um Half Adder para realizar a soma do primeiro bit de cada número, pois ao começar a soma não possuímos um carry.

Portanto, o circuito de um somador de 6 bits é:

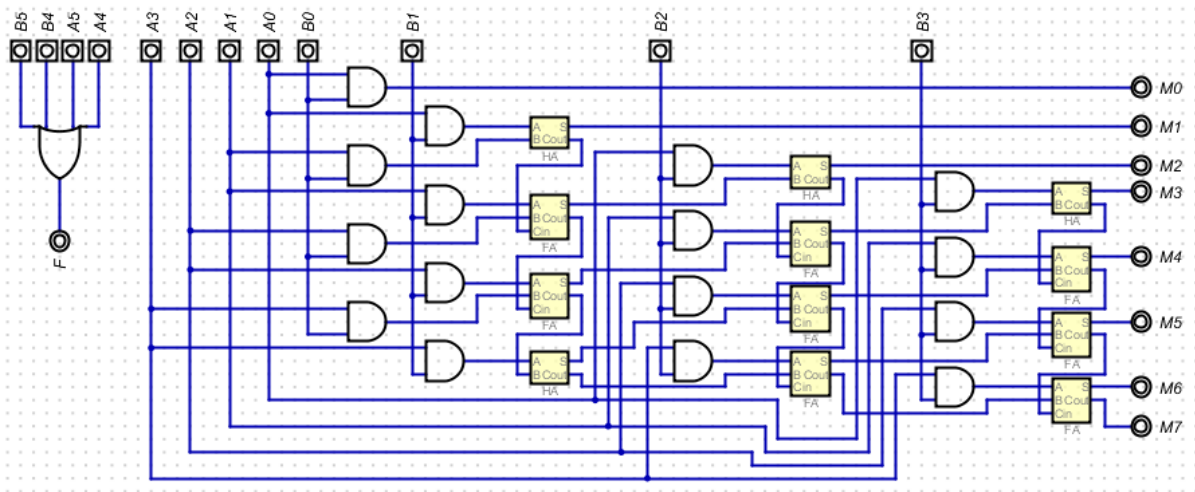


## Multiplicador:

Como uma operação de multiplicação de 4 bits é composta por várias multiplicações e somas, utilizei ANDs para realizar as multiplicações e em seguida utilizei FAs e HAs para realizar as somas.

E como era necessário ter  $F = 1$  quando os bits mais significativos dos números A e B ( $A_4, A_5, B_4$  e  $B_5$ ) fossem diferentes de 0, usei um OR para agrupar os 4 bits, e dessa maneira, quando qualquer um deles for 1, F será igual a 1 também.

Portanto, o circuito para um multiplicador de 4 bits é:

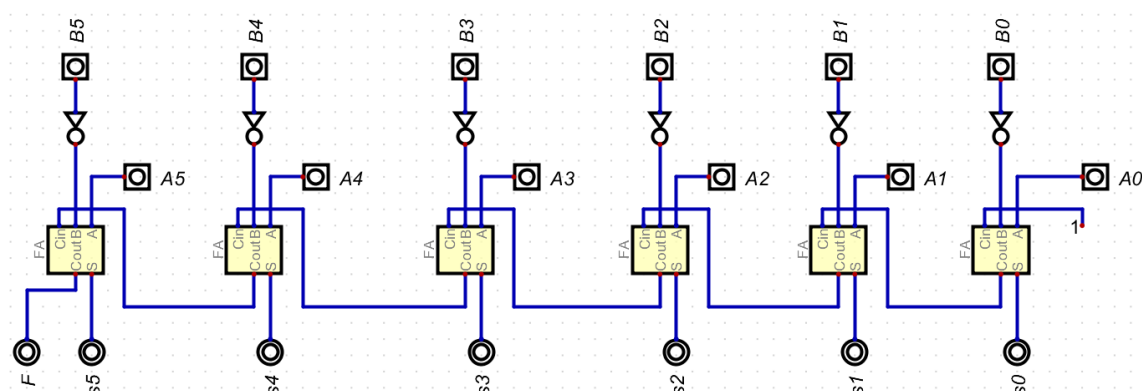


## Subtrator:

Como, para a realização de uma operação de subtração  $A - B$ , deve ser feito o complemento de 2 de B e em seguida somar os números, utilizei um circuito parecido com o do somador, porém com algumas modificações.

Inicialmente, usei diversos NOTs para negar todos os bits de B e em seguida adicionei uma constante de valor 1 servindo como carry no primeiro FA para que o complemento de 2 seja realizado.

Portanto, o circuito para um subtrator de 6 bits é:

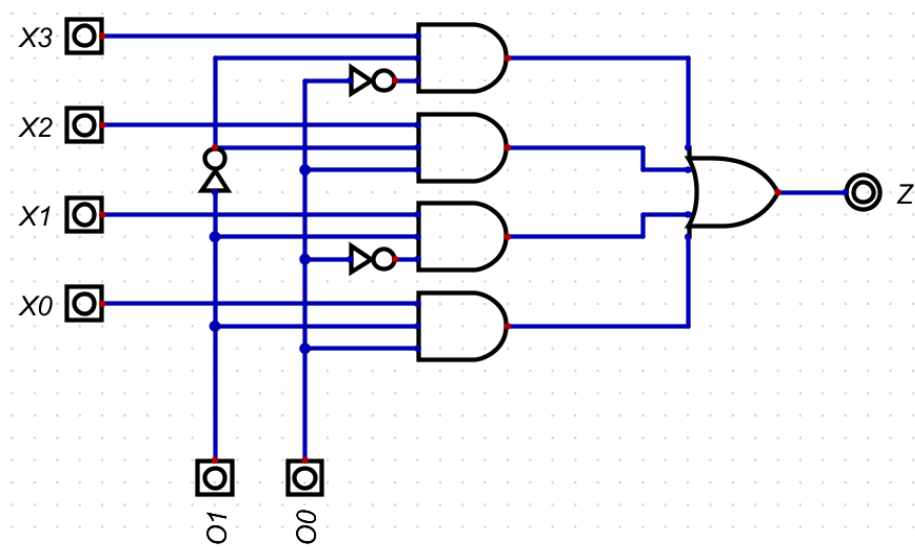


Mux:

Tabela Verdade:

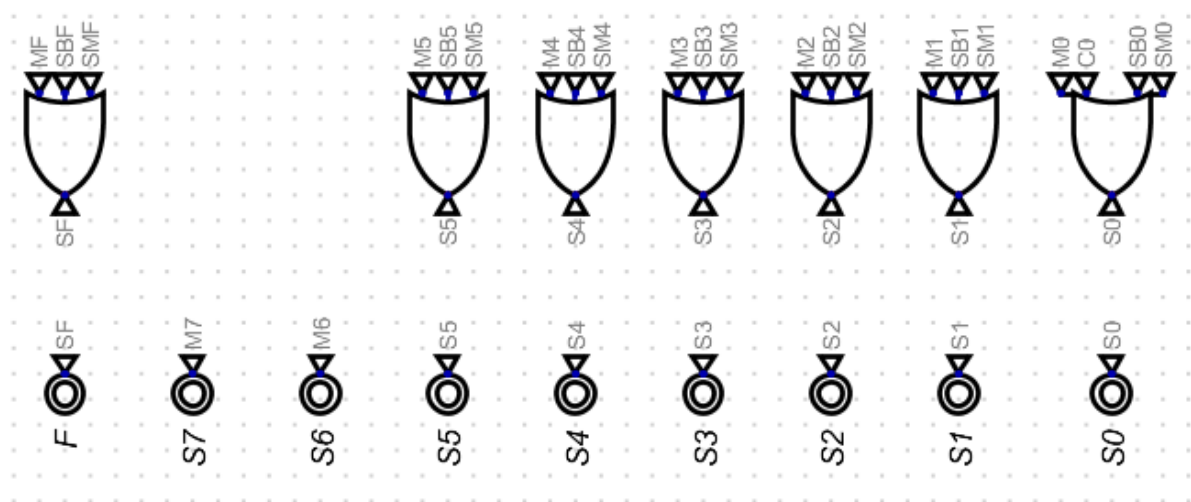
O1	O0	Saída
0	0	X3
0	1	X2
1	0	X1
1	1	X0

Circuito:



Para selecionar a operação desejada utilizei 4 Mux 4:1 (um para cada operação), e neguei algumas de suas entradas para que Z seja 1 somente quando o número correspondente àquela operação foi digitado. E, em seguida, por meio de vários ANDs, multipliquei o resultado de cada operação por Z, pois caso aquela seja a operação desejada, Z será 1 e o resultado da operação permanecerá intacto, e caso essa não seja a operação correta, Z valerá 0, e o resultado da operação será zerado.

Ao final, independentemente da operação desejada, todas elas serão resolvidas, porém, 3 delas vão ser zeradas ao serem multiplicadas pelo resultado de seus Mux, e apenas a operação desejada irá apresentar seu valor correto. Após isso, todos os resultados são passados por vários ORs e encaminhados para as saídas, apenas os bits 6 e 7 do resultado da multiplicação que serão enviados diretamente para a saída, pois as outras operações não têm tantos bits.



E, em meu circuito também utilizei vários túneis, para facilitar a organização do trabalho.