Imagens:

A	В	enable	Igual	Maior	Menor		
0	0	0	0	0	0	Saída:	Igual 🗸
0	0	1	1	0	0	Formato:	Soma de produtos 🗸
0	1	0	0	0	0		B, enable
0	1	1	0	0	1		
1	0	0	0	0	0		00 01 11 10
1	0	1	0	1	0	A 0	0 1 0 0
1	1	0	0	0	0	1	0 0 10 0
1	1	1	1	0	0	ĀBe	nable + A B enable
Saída: Maior V				Saída:	Menor V		
Formato: Soma de produtos 🗸				Formato:			
B, enable				B, enable		e	
	00	01 11 10	D		00 01 11	10	
	0 0	0 0 0		A	0 0 0 🕕	0	
A	1 0	1 0 0		A	1 0 0 0	0	
A \overline{B} enable					A B enable		

Tabela verdade, e os três mapas de Karnaugh usados para o exercícios, Todos os três usando a ajuda do Logisim.

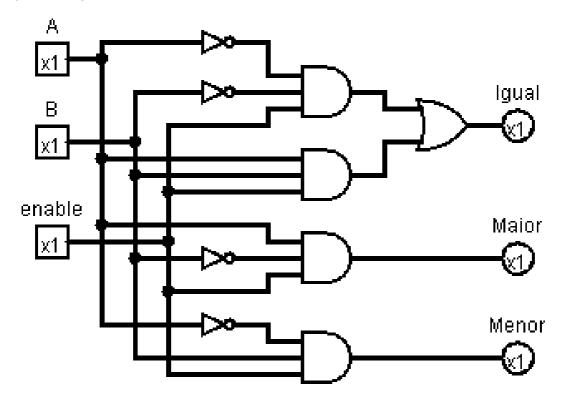
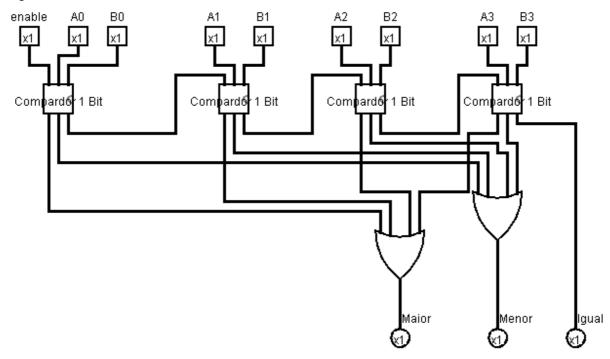
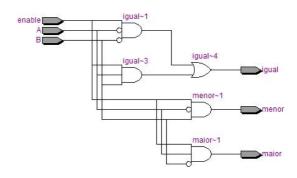


Imagem do circuito do comparador de 1 Bit, Circuito feito utilizando a ferramenta de fazer circuitos no Logisim.



Circuito do Comparador de 4 Bits. Esse circuito foi desenhado no Logisim.

Considerações Finais.



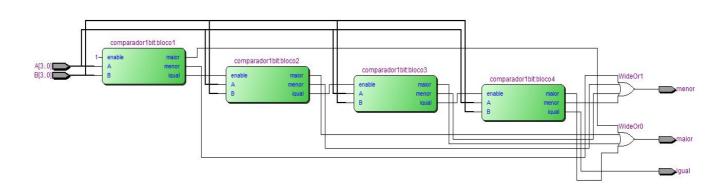


Imagem gerada no Quartus para o comparador de 1 bit. Variável Enable para controle, e A e B como numeros a serem comparados. Também mostrada, a imagem gerada no Quartus para comparador de 4 bits.

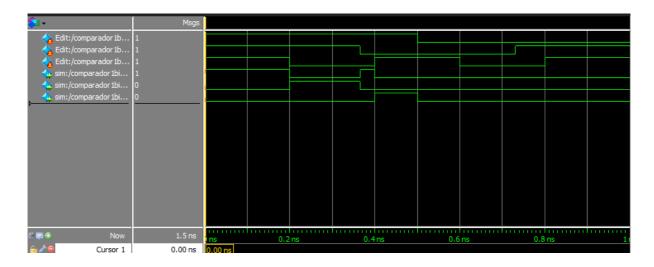


Imagem gerada no Modelsim, 1° variável é o enable, 2° variável A, 3° variável B, 4° variável C, 5° variável igual, 6° variável maior e 7° variável menor.

Análises:

A variável enable deve ser sempre positiva para se comparar o número, por isso torna-se interessante usar a comparação do anterior para enable do próximo, importante para isso é sempre começar do número de maior importância. Os resultados aconteceram como o esperado, retornando sempre 1 para variável esperada, no entanto, não conseguir configurar o Modelsim para testar o comparador de 4 bits, porém, na imagem gerada pelo Quartus ele aparenta funcionar como o esperado, usando o igual do bit anterior como enable para o próximo, e a porta ou para verificar se maior ou menor.