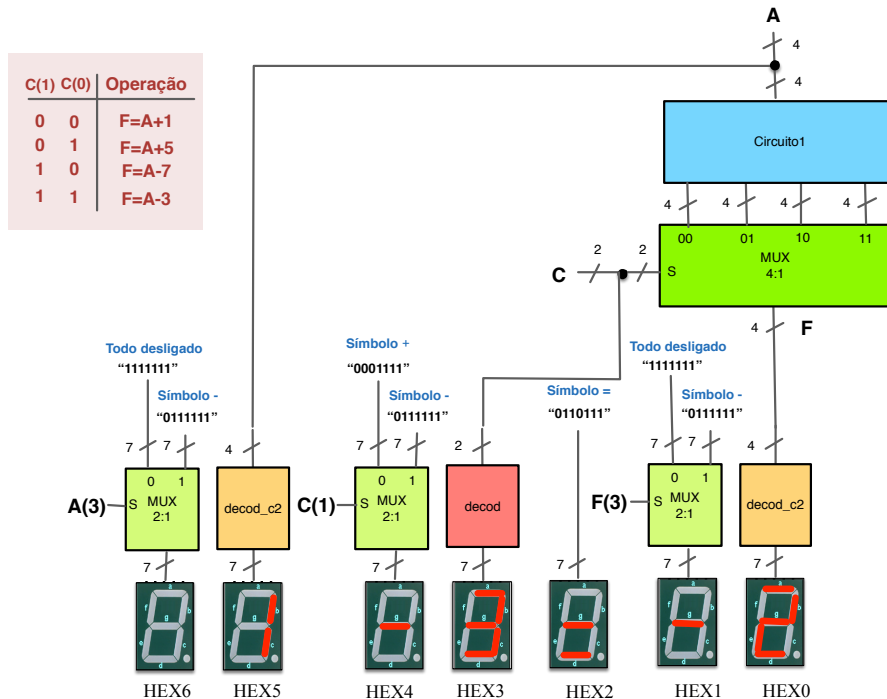


Universidade Federal de Santa Catarina
EEL51015: Técnicas e Circuitos Digitais
Semestre: 2023/2 EXERCÍCIO 2

Atribuição condicional

Implemente um circuito aritmético e um operando A , (4 bits em complemento de 2), escolhida pelo controle C , (2 bits), que implemente as 4 operações mostradas na tabela da Figura. A saída é mostrada nos displays HEX6, HEX5, HEX4, HEX3, HEX2, HEX1, HEX0. Os Displays HEX6 e HEX5 indicam o sinal e o valor da entrada A , respectivamente. O Display HEX4 indica se será feita uma soma ou subtração (símbolo $+$ ou $-$), e o display HEX3 o valor de dita soma ou subtração atendendo à selecção de C (olhar tabela da Figura). O Display HEX2 indica um símbolo $=$, e os Displays HEX1 e HEX0 indicam o sinal e o valor do resultado F , respectivamente. Pode dar uma olhada ao exemplo mostrado nos displays de saída para $A = 1_{(10)} = 0001_{(2)}$ e $C = 11$.



1. Implementar o circuito no Quartus usando VHDL. Para fazer o circuito, o aluno deve usar unicamente:

- Multiplexadores 4 : 1 de 4 bits de entrada-saída de dados (*mux4_1.vhd*).
- Multiplexadores 2 : 1 de 7 bits de entrada-saída de dados (*mux2_1.vhd*).
- Os decodificadores (*decod_c2*), os quais devem dar os valores absolutos da operação de complemento a 2 (por exemplo para entradas $7_{10} = 0111_{C2}$ e $-7_{10} = 1001$ o decoder deverá dar um vector de saída 1111000 para obter um 7 no Display).
- O decoder (*decod*), o qual deve mostrar as 4 opções dos valores mostrados na tabela de operação.
- Para o circuito (*circuito1*), o aluno pode fazer no próprio *usertop.vhd* como foi feito no laboratório 4 usando apenas operadores +

O aluno deverá mostrar algumas combinações funcionando para o professor além dos VHDLs preenchidos, *usertop.vhd*, *decod.vhd*, *decod_c2.vhd*, *mux2_1.vhd* e *mux4_1.vhd* usados nesse apartado. **[8.5 pontos]**

2. Antes de começar a segunda parte, salve uma versão do arquivo *usertop.vhd* que funcionou numa pasta separada para o envio posterior na tarefa, pois na segunda parte vamos modificar dito arquivo. Obtenha agora uma solução otimizada sem o uso de multiplexadores 4 : 1, nem multiplexadores 2 : 1. Para evitar o uso do multiplexador 4 : 1, tente fazer as quatro operações redistribuindo os bits de C numa única operação de soma (Dica: tem um exemplo similar na Figura da folha seguinte). Para a redistribuição dos bits o aluno pode usar o operador & (encadeamento) e portas NOT. Para evitar o uso dos multiplexadores 2 : 1 seria a mesma abordagem, redistribuindo os bits de seleção num único vetor de entrada. Verifique o funcionamento e se está correto copie as linhas de código do novo *usertop.vhd* e mostre ao Professor o funcionamento na placa e VHDL. **[1.5 pontos]**.

**EXEMPLO DE OTIMIZAÇÃO PARA EVITAR USO DE MUX 4:1 NA GERAÇÃO DO VECTOR F:
(O ALUNO DEVE OTIMIZAR O CIRCUITO DA FOLHA 1!!)**

C(1) C(0)	Operação				
0 0	F=A+1	→	$C_1 C_0$	Operação	Valor a somar com A
0 1	F=A+3		00	$A+1=A+1_{(C2)}=A+0001_{(C2)}$	$0001_{(C2)}$
1 0	F=A-7		01	$A+3=A+3_{(C2)}=A+0011_{(C2)}$	$0011_{(C2)}$
1 1	F=A-5		10	$A-7=A+(-7_{(C2)})=A+1001_{(C2)}$	$1001_{(C2)}$
			11	$A-5=A+(-5_{(C2)})=A+1011_{(C2)}$	$1011_{(C2)}$

$VECTOR \leq C_1 \& 0 \& C_0 \& 1;$
 $F \leq VECTOR + A;$
 (Evito o uso de 4 somadores e do MUX 4:1)