

# 3ª Prova – Exercícios

## Circuitos Digitais I

Prof. Fernando Passold ★ Curso de Engenharia Elétrica ★ Universidade de Passo Fundo

20 de junho de 2014

### 1 Problemas Propostos

- 1) Projete o circuito de um subtrator binário, usando o CI 74F83 ou 74F283 (Somadores de 4 bits) capaz de executar as seguintes operações:

$$F = \begin{cases} A - B, & \text{quando } S = 0 \\ B - A, & \text{quando } S = 1. \end{cases}$$

Onde:  $S$  é um bit de seleção que permite selecionar uma das 2 operações acima;  $A$ ,  $B$  e  $F$  são palavras de 4-bits.

Simule o funcionamento do circuito quando  $A = 7$  e  $B = 2$  para os duas opções de níveis lógicos para  $S$ . Observações:

- a) O resultado deve ser visualizado num módulo Display de 7-Segmentos (considerar que na saída do Somador é colocado só e simplesmente um DEC para Display de 7-Segmentos).
- b) O resultado deve ser interpretado, isto é, explique (para os 2 casos mostradas na tabela anterior), que caracter será mostrado no display e porque (justifique).

A figura 1 mostra uma tabela de caracteres resultantes da decodificação formada por pastilhas do tipo 7474, 7448 (DEC/Display 7-Segmentos).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	c	d	e	f		

Figura 1: Saídas resultantes num display de 7-Segmentos para DEC's/Display 7-Segmentos (CIs 7447, 7448).

- 2) Projete um único circuito capaz de realizar as seguintes operações aritméticas:

$$F = \begin{cases} B - A, & \text{quando } A \leq B \\ 2B, & \text{quando } A > B. \end{cases}$$

Dicas:

- (1)  $2B = B + B$ .
- (2) Serão necessários um Somador binário completo, um comparador de magnitude, um multiplexador de 2 canais de entrada + portas lógicas básicas.

- 3) Projete um circuito binário aritmético capaz de realizar as operações citadas abaixo:

$$F = \begin{cases} A - B, & \text{quando } A > B \\ A + B, & \text{quando } A < B \\ 2A, & \text{quando } A = B. \end{cases}$$

onde  $A$ ,  $B$  e  $F$  são palavras de 4-bits.

Além de apresentar o circuito, indique como seria realizado internamente as operações para as seguintes condições de entrada: a)  $A = 6_{(10)}$  e  $B = 3_{(10)}$  e b)  $A = 3_{(10)}$  e  $B = 6_{(10)}$ .

Dica: Usar 1 comparador de magnitude, 1 somador binário + portas lógicas básicas.

- 4) Projete um circuito capaz de realizar a seguinte operação:

$$\begin{aligned} \text{SE } A = B \\ F = 2B \\ \text{SENÃO} \\ F = A + B \\ \text{FIM-SE} \end{aligned}$$

Onde  $A$ ,  $B$  e  $F$  são palavras de 4 bits.

Atenção: Não é permitido o uso do comparador de magnitude!

Dica:  $2B = (B + B)$ .

- 5) Projete um circuito capaz de realizar a seguinte operação:

$$\begin{aligned} \text{SE } A \neq B \\ F = 2B \\ \text{SENÃO} \\ F = A + B \\ \text{FIM-SE} \end{aligned}$$

Onde  $A$ ,  $B$  e  $F$  são palavras de 4 bits.

Atenção: Não é permitido o uso do comparador de magnitude!

Dica:  $2B = (B + B)$ .

- 6) Projete um circuito capaz de realizar a seguinte operação:

SE  $A > B$   
 $F = 2B$   
 SENÃO  
 $F = A + B$   
 FIM-SE

Onde  $A$ ,  $B$  e  $F$  são palavras de 4 bits.

Depois de projetar o circuito mostre como seriam realizadas internamente as operações dentro do circuito quando: a)  $A = 3_{(10)}$  e  $B = 6_{(10)}$  e b)  $A = 6_{(10)}$  e  $B = 3_{(10)}$ .

- 7) Projete um circuito somador BCD à partir de um somador binário. Use somadores binários, por exemplo, o CI 74LS83 ou 74LS283. Mostre os resultados para: a)  $S = 4 + 5$  e b)  $S = 5 + 5$ .

*Dica: desenvolva o circuito que detecta quando é necessária a conversão de binário para BCD (repare que neste caso, basta somar +6 à saída do somador binário usada na entrada do circuito).*

- 8) Projete um circuito capaz de converter código binário de entrada  $A$ , com sinal (negativo ou positivo) usando complemento-2, de 4 bits, para código BCD de saída ( $F$ ), capaz de ser visualizado num display de 7-Segmentos. Mostre como o circuito se comporta para: a)  $A = 6_{(10)}$  e b)  $A = -6_{(10)}$ .  
 Obs: O código de entrada pode variar entre  $[-8.. +7]$ .

- 9) Projete o circuito de um somador binário, usando o CI 74F83 ou 74F283 (Somadores de 4 bits) mais outras portas lógicas básicas, capaz de executar as seguintes operações:

$$F = \begin{cases} A + B & \text{(Aritmético), quando } S = 0 \\ A + B & \text{(Lógico), quando } S = 1. \end{cases}$$

Onde:  $S$  é um bit de seleção que permite selecionar uma das 2 operações acima.

Observações:

- a) O resultado deve ser visualizado num módulo Display de 7-Segmentos (que caracteres devem ser gerados?).  
 b) O resultado deve ser interpretado para a notação decimal.
- 10) Projete o circuito capaz de executar as seguintes operações (aritméticas), usando o CI 74F83 ou 74F283 (Somadores de 4 bits):

$$F = \begin{cases} A + B, & \text{quando } S = 0 \\ A - B, & \text{quando } S = 1. \end{cases}$$

Onde:  $S$  é um bit de seleção que permite selecionar uma das 2 operações acima. Simule os resultados obtidos quando  $A = 2$  e  $B = 5$ .

Observações:

- a) O resultado deve ser visualizado num módulo Display de 7-Segmentos.  
 b) O resultado deve ser interpretado para a notação decimal.

- 11) Projete um circuito capaz de mostrar num duplo display de 7-Segmentos (um para Dezenas e outro para Unidades), código binário de entrada variando de 0 à 15 – ver figura 2.

*Dica: Este exercício pede um circuito que converta código binário de entrada para código BCD de saída, usando um sub-circuito que detecta quando esta conversão se faz necessária. Esta conversão provavelmente exigirá um CI Somador).*



Figura 2: Display duplo de 7-Segmentos.

- 12) Projete um circuito capaz de mostrar num display de 7-Segmentos (do tipo mostrado na figura 3, isto é, use o display da esquerda apenas para representar o sinal e o da direita para representar a magnitude do número), código binário de entrada de 4-bits, com sinal (Complemento 2), variando de -8 à +7. Isto é, este circuito deve ser capaz de mostrar no display (duplo), números de -8 até +7.

Aproveite para explicar o que ocorre se o usuário tentar entrar com um valor como “-9” ou “+8”?

*Dica: Este circuito terá que testar o bit  $D_3$  (MSB) de entrada para descobrir se o número é positivo ou negativo. E no caso de números negativos, terá de converter este número negativo no formato binário complemento 2 para código BCD de saída (positivo).*

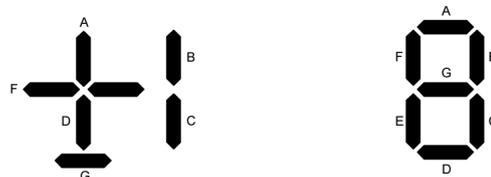


Figura 3: 1/2 Display com sinal (esquerda) e Display 7-Segmentos completo (direita).

13) Projete uma ULA de 4-bits capaz de realizar as operações indicadas na tabela abaixo:

Seleção Operação $S_1 S_0$	Função	Observações
00	$F = A + B$	Requer $C_0 = 0$
01	$F = A - B$	Requer $C_0 = 1$
10	$F = B - A$	Requer $C_0 = 1$
11	$F = \text{Max}(A, B)$	$C_0 = X$

onde  $A$ ,  $B$  e  $F$  são palavras de 4-bits,  $C_0$ =carry-in,  $C_4$ =carry-out.

**Obs.:** Poderá ser usado apenas 1 CI Somador (7483 ou 74283) além de circuitos MSI e portas lógicas básicas.

14) Usando a ULA 74F383, mostrar como ajustar a mesma para realizar as operações mostradas à seguir:

a)  $F = A + B$

b)  $F = A - B$

c)  $F = \overline{B}$  (inverte os bits da palavra  $B$ )

Supor ainda que a saída desta ULA seria passada diretamente para um DEC/Display de 7-Segmentos (CI 74F48). Além de indicar como a ULA deveria ser ajustada, mostrar que caractere seria gerado no display de 7-segmentos para todas as operações consideradas anteriormente. Em caso de geração de número negativo, confirmar se o resultado binário gerado corresponde à contraparte decimal. Considere sempre que:  $A = 2_{(10)}$  e  $B = 4_{(10)}$ .

Select			Operating Mode
S0	S1	S2	
L	L	L	Clear
H	L	L	B minus A
L	H	L	A minus B
H	H	L	A plus B
L	L	H	$A \oplus B$
H	L	H	$A + B$
L	H	H	$AB$
H	H	H	Preset

Extraído do DataSheet da NXP Semiconductors  
Tabela 1: 74F382 – Function Select Table.

#### IC 74F328 – DESCRIPTION:

The 74F382 performs three arithmetic and three logic operations on two 4-bit words, A and B. Two additional Select (S0–S2) input codes force the Function outputs Low or High. An overflow output is provided for convenience in Two's Complement arithmetic.

A carry output is provided for ripple expansion. For high-speed expansion using a carry look-ahead generator, refer to the 74F381 data sheet.

Signals applied to the Select inputs, S0–S2, determine the mode of operation, as indicated in the Function Select Table. An extensive listing of input and output levels is shown in the Function Table. The circuit performs the arithmetic functions for either active-High or active-Low operands, with output levels in the same convention. In the subtract operating modes, it is necessary to force a carry (High for active-High operands, Low for active-Low operands) into the Cn input of the least significant package. **The overflow output OVR** is the Exclusive-OR of Cn+3 and Cn+4; a High signal on OVR indicates overflow in Two's complement operation. Typical propagation delay is 7.0ns. When the 74F382 is cascaded to handle word lengths longer than 4 bits, only the most significant overflow (OVR) output is used.

Obs.: Extraído do DataSheet da NXP Semiconductors

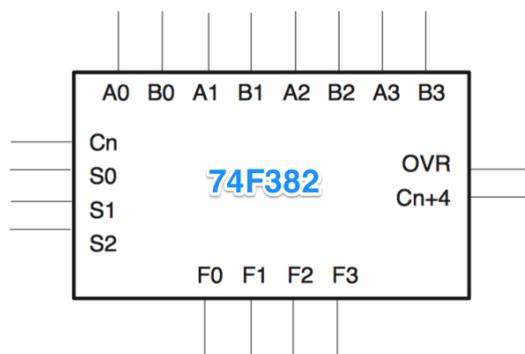


Figura 4: 74F382 Logic Symbol.



# Soluções

1) Note que nesta questão, sempre será necessário inverter os bits, ou da palavra A ou da palavra B, o que significa que será necessário incluir portas XOR trabalhando como inversor controlado nas duas entradas do somador binário. Note ainda que como sempre é realizada uma subtração, além de ser necessário inverter os bit da palavra trabalhando com subtraendo, a entrada de "carry-in" (entrada C0 do somador binário) deve ficar permanentemente em nível lógico alto (+Vcc). O que vai faltar é gerar o sinal de controle para as portas XOR para indicar quando é necessário inverter os bits da palavra A ou da palavra B.

Notamos que quando  $S = 1$ , os bits da palavra A devem ser invertidos e quando  $S = 0$ , os bits da palavra B devem ser invertidos, então basta conectar a entrada S à entrada de controle das portas XOR que invertem ou não os bits da palavra A e uma porta NOT conectada entre o S as entrada de controle das portas XOR relacionadas com os bits da palavra B. A figura 5, mostra na forma de um diagrama de blocos esta solução.

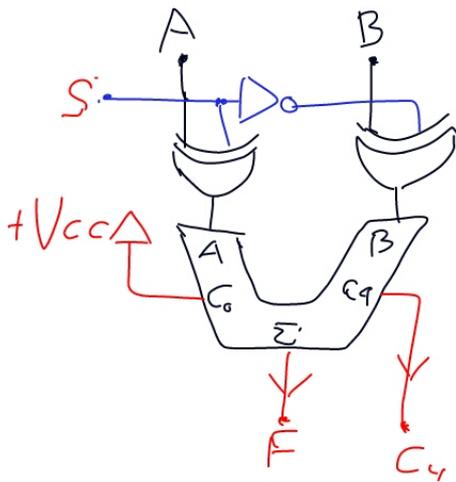


Figura 5: Solução possível para questão 1 (diagrama em blocos).

A questão pede também que se simule o que ocorre quando  $A = 7_{(10)} = 0111_{(2)}$ ,  $B = 2_{(10)} = 0010_{(2)}$  e:

a)  $S = 0$ :  
Neste caso se realiza a operação:  $F = A - B$ :

			11111 <-- bits de carry
A	A	7	0111
- B	+(-B)	-2	+ 1101
---	-----	--	-----
F	F	5	0101

Notamos que é gerada a resposta:  $F = 0101_{(2)} = +5_{(10)}$ , conforme esperado. O bit MSB de F vai à "0" indicando que o resultado gerado é positivo. Também o bit de carry-out do somador é ativado,

mas este bit não faz parte do resultado, sendo desprezado.

Se a saída do somador binário for conectada à um DEC/Display 7-Segmentos, o mesmo vai mostrar o caracter correto:



b)  $S = 1$ :  
Neste caso se realiza a operação:  $F = B - A$ :

			1 <-- bits de carry
B	B	2	0010
- A	+(-A)	-7	+ 1000
---	-----	--	-----
F	F	-5	1011

Neste caso é gerada a saída binária  $F = 1011_{(2,C2)}$ , onde percebemos que o bit MSB de F foi setado (= 1), o que significa que o resultado é negativo. O resultado gerado em decimal implica no número  $-5_{(10)}$ , ou seja:

1011	
Complementando os bits	
v	
0100	
+ 1	
----	
0101	

Se a saída do somador binário for conectada à um DEC/Display 7-Segmentos, o mesmo vai mostrar o caracter ue corresponde ao código binário  $1011_{(2)} = 11_{(10)}$ , ou seja:



2) Neste caso se percebe que o somador sempre receberá a palavra B sem nenhuma alteração, enquanto que o outro operador dependerá do resultado de uma comparação à ser realizada entre as palavras de entrada A e B.

Desta forma, podemos conectar a palavra B externa ao circuito, diretamente à entrada B do somador binário.

Notamos ainda que a entrada A do somador binário recebe a palavra B quando  $A > B$  (o somador realiza então:  $\Sigma = B + B = 2B$ ), caso contrário a entrada A do somador deve receber os bits complementados da palavra A externa ao circuito. Note que são 2 possibilidades e que deve ser passado os bits de  $B_{\text{externo}} \rightarrow A_{\text{somador}}$  ou  $\bar{A}_{\text{externo}} \rightarrow A_{\text{somador}}$ . O circuito digital que executa este tipo de **seleção de dados de entrada** é o MUX. Então um simples MUX de 2 linhas de entrada resolve o problema na entrada A do somador binário. Resta gerar corretamente o nível lógico na entrada S de Seleção deste MUX e posicionar corretamente  $B_{\text{externo}}$  e  $\bar{A}_{\text{externo}}$  nas entradas  $I_0$  e  $I_1$  deste MUX.

Outro CI (função lógica) pode ser usado para gerar o sinal  $S$  necessário para comandar o MUX, o **Comparador de Magnitude**. Notamos pelo enunciado da questão que quando  $A > B$  deve ser exceção a operação  $F = B + B$  e caso contrário, deve ser executada a operação  $F = B - A$ . A fim de tornar o circuito o mais simples possível, pode-se optar por conectar a saída  $O_{A>B}$  do Comparador de Magnitude à entrada  $S$  de Seleção do MUX. Neste caso, a entrada  $S$  do MUX vai à "1" toda vez que  $A > B$ , caso contrário ( $A \leq B$ ),  $O_{A>B} = 0 \rightarrow S = 0$ . Então sendo cuidadosos, como queremos que quando  $A > B$  o somador execute  $F = B + B$  e neste caso,  $S = 1$ , conectamos a palavra  $B_{\text{externa}}$  à entrada  $I_1$  do MUX e a outra palavra de entrada  $\bar{A}_{\text{externa}}$  à entrada  $I_0$  do MUX.

Só nos falta agora, lembrar de acrescentar portas NOT entre a entrada  $A$  externa do circuito e as entradas  $I_0$  do MUX, além de garantir que no caso de  $S = 0$  o bit de carry-in do somador binário seja setado (= 1) para que a subtração em Complemento-2 seja realizada da forma correta. Esta última parte é fácil de solucionar acrescentando-se uma porta NOT conectada entre a saída  $O_{A>B}$  do Comparador de Magnitude e a entrada  $C_0$  (carry-in) do somador binário, assim, toda vez que  $A \leq B$ , a saída  $O_{A>B}$  do Comparador de Magnitude vai à "0" mas a entrada  $C_0$  (carry-in) do Somador Binário vai à "1".

A figura 6 mostra na forma de um diagrama em blocos, esta proposta de solução para a questão.

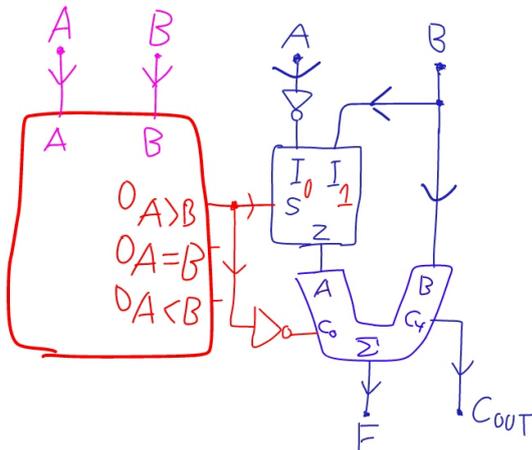


Figura 6: Diagrama em blocos de solução possível para a questão 2.

nas 2:

$$F = \begin{cases} A - B, & \text{quando } A > B \\ A + B, & \text{quando } A < B \\ A + A, & \text{quando } A = B. \end{cases}$$

ou seja, quando  $A \leq B$ , o circuito sempre estará realizando:  $F = A + B$ , caso contrário, estará realizando:  $F = A - B$ .

Então fica mais fácil implementar um circuito com base num **Comparador de Magnitude**, onde usamos a saída  $O_{A>B}$  para realizar:  $F = A - B$  e em caso contrário, realizar:  $F = A + B$ .

Notamos ainda que a palavra externa  $A$  pode ser conectada diretamente à entrada  $A$  do **Somador Binário**. Só muda o operador que deve ingressar na entrada  $B$  do Somador Binário. Quando  $A > B$ , a entrada  $B$  do Somador recebe:  $B_{\text{Somador}} \leftarrow \bar{B}_{\text{Externo}}$  e sua entrada de carry-in deve receber:  $C_0 = 1$ ; caso contrário, sua entrada  $B$  recebe:  $B_{\text{Somador}} \leftarrow B_{\text{Externo}}$  e sua outra entrada de carry-in deve permanecer em nível lógico baixo,  $C_0 = 0$ .

Percebemos que o acréscimo de portas XOR como **Inversor Controlado** para os bits da palavra externa  $B$  permitem resolver facilmente o problema se sua entrada de controle for conectada diretamente à saída  $O_{A>B}$  do Comparador de Magnitude.

A figura 7 esboça um diagrama em blocos para esta solução proposta.

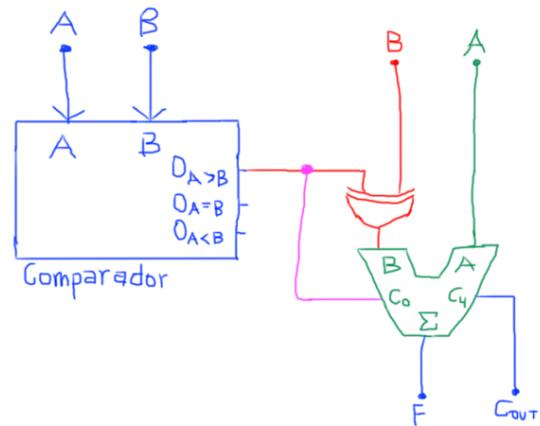


Figura 7: Diagrama em blocos mostrando uma solução para a questão 3.

3) Nesta questão se pede um circuito que realize:

$$F = \begin{cases} A - B, & \text{quando } A > B \\ A + B, & \text{quando } A < B \\ 2A, & \text{quando } A = B. \end{cases}$$

Note que estes 3 casos podem ser resumidos em ape-

- 4) Nesta questão, o circuito deve realizar:  
 $F = 2B = B + B$ , quando  $A = B$ ;  
 caso contrário deve fazer:  
 $F = A + B$ , quando  $A \neq B$ .

Observando-se a lógica desta questão percebe-se que este circuito SEMPRE estará realizando  $F = A + B$ , pois quando  $A = B$ , o circuito vai realizar:  $F = A + B = B + B = 2B$ .

A solução simples é mostrada na figura 8.

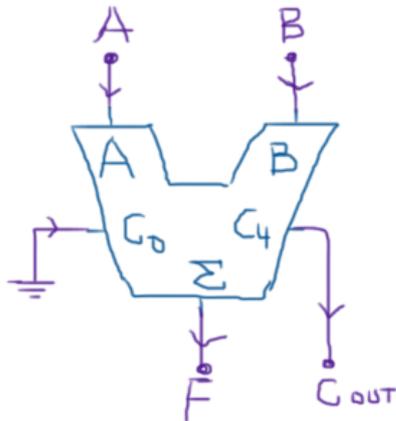


Figura 8: Diagrama em blocos da solução simples para a questão 4.

- 5) Nesta questão, o circuito deve realizar:  
 $F = 2B = B + B$ , quando  $A \neq B$ ;  
 caso contrário, deve fazer:  
 $F = A + B$ , quando  $A = B$ .

É um caso similar ao anterior. Este circuito também SEMPRE estará realizando a soma:  $F = B + B$ , pois quando  $A + B$ , a soma  $F = A + B$  resulta em  $F = B + B = 2B$ .

A solução simples é mostrada na figura 9.

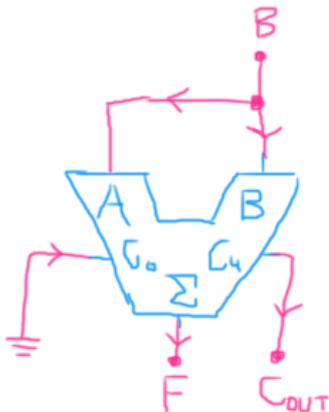


Figura 9: Diagrama em blocos para a solução simples da questão 5.

- 6) Este circuito deve realizar:  
 $F = 2B = B + B$ , quando  $A > B$ ;

caso contrário, deve fazer:  
 $F = A + B$ .

Aqui se percebe que o circuito deve fazer:

- Quando  $A > B \rightarrow F = B + B$ ;
- Senão  $A \leq B \rightarrow F = A + B$ ;

Nota-se que o Somador binário sempre vai receber diretamente a palavra B, o que muda é o dado para a sua entrada A.

Para resultar num circuito mais simples, podemos usar a saída  $O_{A>B}$  de um Comparador de Magnitude e usar o nível lógico desta saída para comandar a entrada S, de Seleção de um MUX de 2 canais de entrada acoplado à entrada A do somador binário.

Assim, quando  $A > B$ , a saída  $O_{A>B}$  do comparador vai a nível lógico alto, e assim a entrada S do MUX, o que fará com que na saída Z do MUX corresponda ao dado aplicada à entrada  $I_1$  que neste caso ( $A > B$ ) dever ser a palavra B. Caso contrário, quando  $A \leq B$ , a saída  $O_{A>B}$  vai à "0", a entrada S do MUX vai à "0", o que faz com que o mesmo coloque na sua saída Z o conteúdo presente na sua entrada  $I_0$  que neste caso deve ser a palavra A.

A figura 10 mostra esta solução.

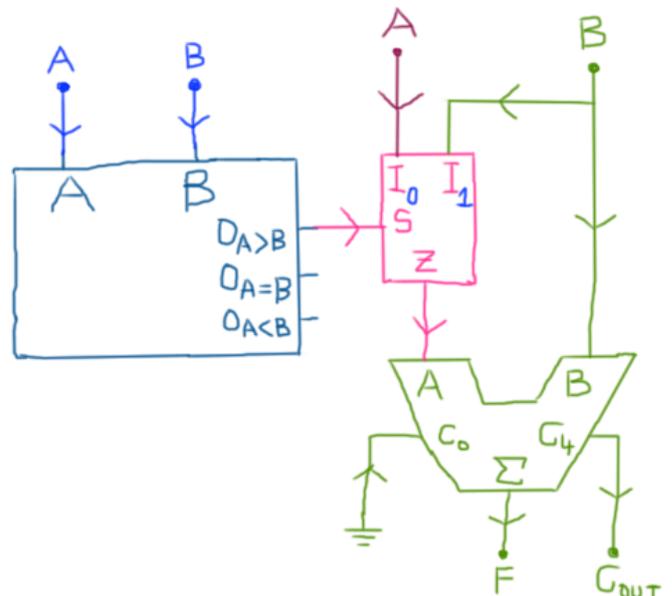


Figura 10: Diagrama em blocos mostrando uma solução para a questão 6.

- 7) Questão já resolvida em sala de aula.

- 8) Projeto de circuito para converter de código binário com sinal usando notação Complemento-2 para código de saída BCD. Isto significa usar um display como o mostrado na figura 3 para mostrar números binários na sua entrada variando entre  $-8$  até  $+8$ .



A solução é simples. Quando o circuito recebe um número negativo, o bit de entrada MSB estará ativado indicando a necessidade de uma conversão. Por exemplo, suponha que o mesmo esteja recebendo o código de entrada binário  $B = 1111_{(2)} = \bar{F}_{(16)}$ . Isto significa que trata-se de um número negativo que na notação decimal corresponde à:  $-1_{(10)}$ , ou seja:

$$\begin{array}{r}
 1111 \\
 | \quad \text{Complementando os bits de entrada} \\
 \vee \\
 0000 \\
 + 1 \quad \text{<--- necessitamos somar +1} \\
 \text{----} \\
 0001
 \end{array}$$

O circuito deve então verificar o nível lógico do bit MSB presente na sua entrada e em caso de necessidade, inverter estes bits e somar aritmeticamente  $+1$  aos bits que se acabou de inverter.

Isto pode ser obtido facilmente usando uma porta XOR trabalhando como inversor controlado, cuja entrada de controle ficará conectada diretamente ao bit MSB de entrada. As saídas da porta XOR são então conectadas à uma das entradas de um somador binário. A outra entrada do somador binário é zerada (ressetada) e o mesmo bit MSB de entrada é usado para gerar o  $+1$  em caso de necessidade, fazendo-se sua conexão à entrada de carry-in do somador binário. O resultado convertido aparece na saída do somador binário. E o próprio bit de entrada MSB pode ser usado para gerar o sinal “-” no meio-display.

A figura 11 mostra o resultado deste circuito.

A figura 12 mostra o circuito anterior sendo simulado. Note na figura 12 que as entradas do circuito de conversão correspondem aos pontos de ligação com a saída do contador binário 74LS93 usado para gerar a sequencia de entrada variando de  $0000_{(2)}$  até  $1111_{(2)}$ . Neste caso, a saída *QA* corresponderia a entrada LSB do circuito de conversão, ou bit, *F0*; a saída *QD* corresponderia à entrada MSB do circuito de conversão, ou bit *F3* e assim sucessivamente:  $QB \rightarrow F1$  e  $QC \rightarrow F2$ .

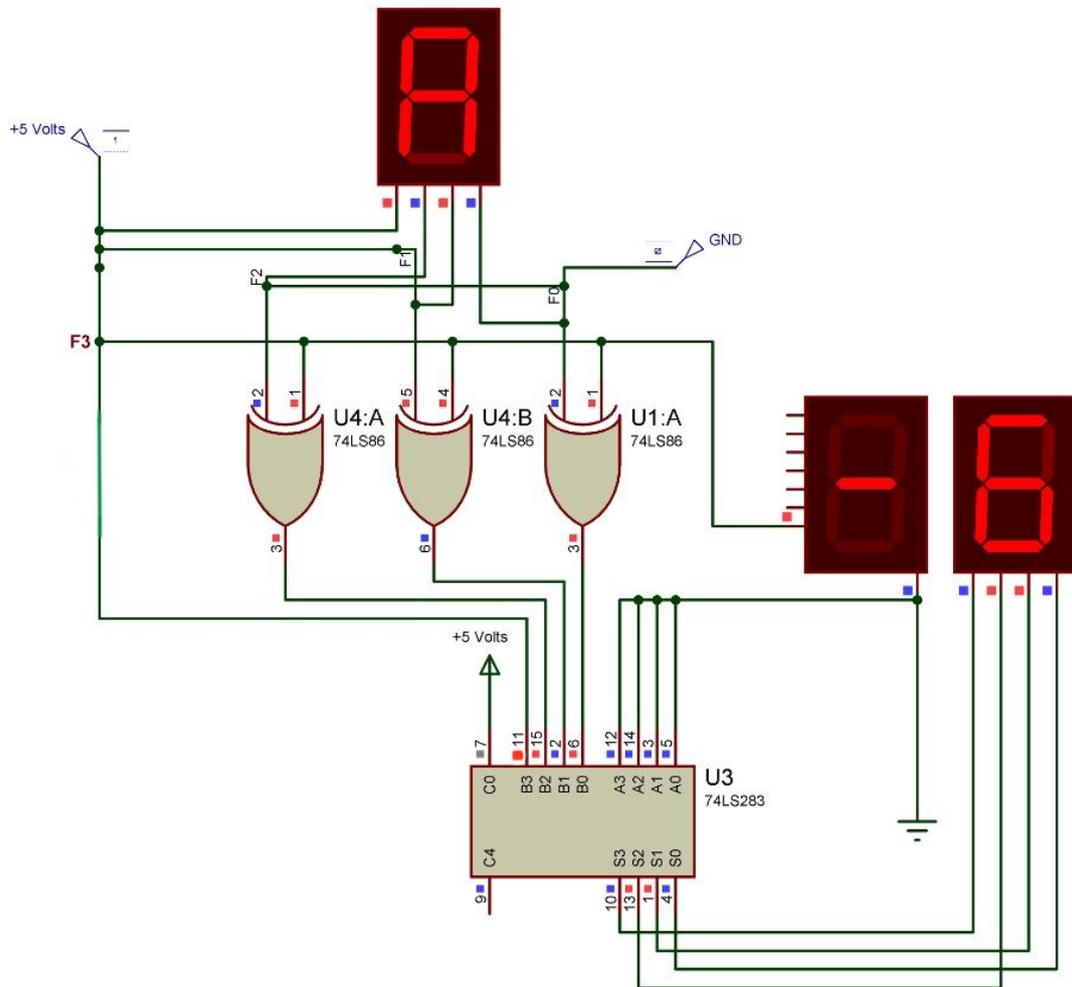


Figura 11: Circuito conversor de código binário com sinal na notação complemento 2 para código BCD de saída.

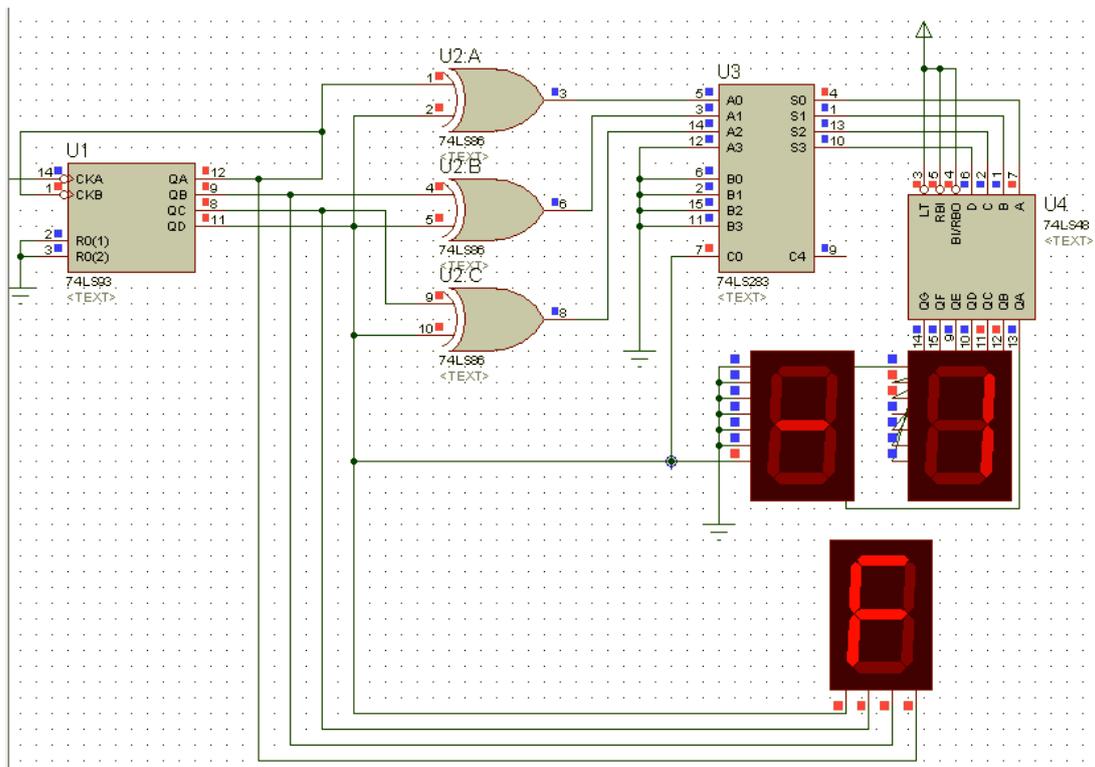


Figura 12: Simulação do Circuito conversor número binário com sinal na notação Complemento-2 para código BCD na sua saída.