



LABORATÓRIO DE SISTEMAS DIGITAIS AVANÇADOS

Prof. Imbiriba

Exercício 02 – Números e Display

Este é um exercício de projeto de circuitos combinacionais capaz de realizar conversões de números binários em decimais e adição de códigos binários em decimais (BCD).

PARTE 1

Desejamos exibir nos displays de 7 segmentos HEX1 e HEX0 da placa DE0 os valores definidos pelas chaves SW7–0. Permita que os valores indicados por SW7–4 e SW3–0 sejam exibidos em HEX1 e HEX0, respectivamente. Seus circuitos devem ser exibidos para exibir os dígitos de 0 a 9 e devem tratar os valores de 1010 a 1111 como não importando.

1. Crie um novo projeto que será usado para implementar o circuito desejado na placa DE0. Fora a implementação no Quartus você deve manualmente desenvolver as expressões lógicas que definirão as saídas para os displays de 7 segmentos. Portanto, no código VHDL, você deve usar apenas instruções de atribuição simples e especificar cada função lógica como uma expressão booleana.
2. Escreva um arquivo VHDL que ofereça a funcionalidade necessária. Inclua esse arquivo no seu projeto e atribua os pinos no FPGA para conectar-se as chaves e displays de 7 segmentos. Certifique-se de incluir as atribuições de pinos necessárias.
3. Compile o projeto e faça o download do circuito compilado no chip FPGA.
4. Teste a funcionalidade do seu projeto, alternando as chaves e observando os displays.

PARTE 2

Você deve projetar um circuito que converta um número binário de quatro bits $V = v_3, v_2, v_1, v_0$ em seu equivalente decimal de dois dígitos $D = d_1, d_0$. A Tabela 1 mostra os valores de saída necessários. Um projeto parcial desse circuito é apresentado na Figura 1. Ele inclui um comparador que verifica quando o valor de V é maior que 9 e usa a saída desse comparador no controle dos displays de 7 segmentos. Você deve concluir o projeto deste circuito.



$v_3v_2v_1v_0$	d_1	d_0
0000	0	0
0001	0	1
0010	0	2
...
1001	0	9
1010	1	0
1011	1	1
1100	1	2
1101	1	3
1110	1	4
1111	1	5

Tabela 1: Valores a serem convertidos no display.

A saída z do circuito comparador pode ser especificada usando uma única expressão booleana, com as quatro entradas V_3-0 . Projete essa expressão booleana criando uma tabela verdade que mostre as avaliações das entradas V_3-0 para as quais z deve ser 1.

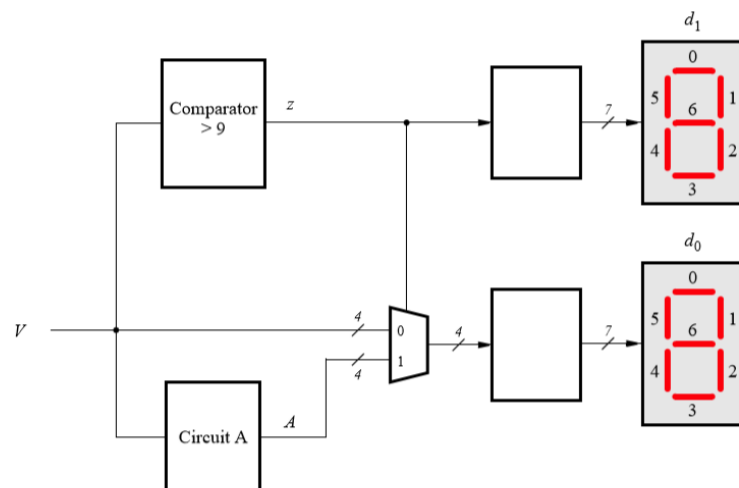


Figura1: Projeto parcial do circuito do conversor binário para decimal.

Observe que o circuito na Figura 1 inclui um multiplexador de 2 para 1 de largura de 4 bits (um multiplexador semelhante foi descrito como parte do Exercício de Laboratório 1). O objetivo deste multiplexador é direcionar o dígito d_0 com o valor de V quando $z = 0$ e o valor de A quando $z = 1$.

Para o circuito A, considere o seguinte: Para os valores de entrada $V \leq 9$, o circuito A não é importante, pois o multiplexador na Figura 1 apenas seleciona V nesses casos. Porém, nos valores de entrada $V > 9$, o multiplexador seleciona A . Portanto, A deve fornecer valores de saída que implementem corretamente a Tabela 1 quando $V > 9$. Você precisa projetar o circuito A para que a entrada $V = 1010$ forneça uma saída $A = 0000$, e a entrada $V = 1011$ fornece a saída $A = 0001$, ..., e a entrada $V = 1111$ fornece a saída $A = 0101$. Projete o circuito A criando uma tabela verdade com as entradas V_3-0 e as saídas A_3-0 .

Execute as seguintes etapas:

1. Escreva o código VHDL para implementar seu design. O código deve ter a entrada de 4 bits SW3-0, que deve ser usada para fornecer o número binário V, e as duas saídas de 7 bits HEX1 e HEX0, para mostrar os valores dos dígitos decimais d1 e d0. O objetivo deste exercício é usar instruções de atribuição VHDL simples para especificar as funções lógicas necessárias usando expressões booleanas. Seu código VHDL não deve incluir nenhuma declaração IF-ELSE, CASE ou similar.
2. Faça um projeto Quartus para sua entidade VHDL.
3. Compile o circuito e use a simulação funcional para verificar a operação correta do seu comparador, multiplexadores e circuito A.
4. Faça o download do circuito em uma placa FPGA. Teste o circuito tentando todos os valores possíveis de V e observando os displays de saída.

PARTE 3

A Figura 2a mostra um circuito para um somador completo, que possui as entradas a,b,c, e produz as saídas s e c_o. As partes b e c da figura mostram o esquemático de circuito e a tabela de verdade para o somador completo, que produz a soma binária de dois bits $c_o^s = a + b + c_i$. A Figura 2d mostra como quatro instâncias desse módulo somador completo podem ser usadas para projetar um circuito que adiciona dois números de quatro bits. Esse tipo de circuito é geralmente chamado de somador com transporte do *carry*, devido a maneira como os sinais de transporte são transmitidos de um somador completo para o próximo. Escreva o código VHDL que implementa esse circuito, conforme descrito abaixo.

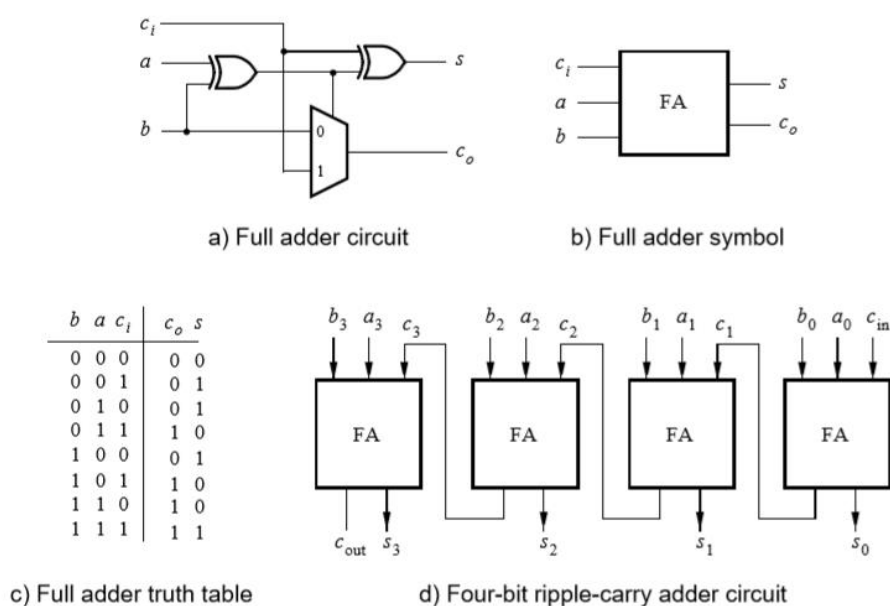


Figura 2: Circuito somador com vai um (*carry in/out*).



1. Crie um novo projeto Quartus para o circuito somador. Escreva uma entidade VHDL para o subcircuito total do somador e escreva uma entidade VHDL de nível superior que instancia quatro instâncias desse somador total.
2. Use as chaves SW7-4 e SW3-0 para representar as entradas A e B, respectivamente. Use SW8 para a c_{in} de transporte do adicionador. Conecte as saídas do somador, c_{out} e S, às luzes vermelhas LEDR.
3. Inclua as atribuições de pinos necessárias para a placa da série DE0, compile o circuito e faça o download no chip FPGA.
4. Teste seu circuito tentando valores diferentes para os números A, B e c_{in} .

PARTE 4

Na Parte 2, discutimos a conversão de números binários em dígitos decimais. Nesta parte, você deve projetar um circuito que tenha dois dígitos decimais, X e Y, como entradas. Cada dígito decimal é representado como um número de 4 bits. Na literatura técnica, isso é chamado de representação decimal codificada binária (BCD). Você deve projetar um circuito que adicione os dois dígitos do BCD. As entradas para o seu circuito são os números X e Y, mais uma entrada, c_{in} . Quando essas entradas são adicionadas, o resultado será um número binário de 5 bits. Mas esse resultado deve ser exibido em displays de 7 segmentos como uma soma B1 de dois dígitos S1, S0. Para uma soma igual a zero, você exibirá S1,S0 = 00, para a soma de um S1,S0 = 01, para nove S1,S0 = 09, para dez S1,S0 = 10 e assim por diante. Observe que as entradas X e Y são assumidas como dígitos decimais, o que significa que a maior soma que precisa ser tratada por este circuito é S1,S0 = 9 + 9 + 1 = 19.

Execute as etapas abaixo.

1. Crie um novo projeto Quartus para o seu somador BCD. Você deve usar o circuito somador de quatro bits da Parte 3 para produzir uma soma de quatro bits e realizar a operação $X + Y$. Uma boa maneira de elaborar o design de seu circuito é primeiro fazer com que ele manipule apenas somas $(X + Y) \leq 15$. Com esses valores, seu circuito da Parte 2 pode ser usado para converter o valor de 4 bits nos dígitos de dados S1,S0. Então, quando isso estiver funcionando, modifique seu projeto para manipular valores de $15 < (X + Y) \leq 19$. Uma maneira de fazer isso ainda é usar o circuito da Parte 2, mas modificar suas saídas antes de conectá-las ao display para fazer os ajustes necessários quando a soma do somador exceder 15.

Escreva seu código VHDL usando instruções de atribuição simples para especificar as funções lógicas necessárias - não use outros tipos de instruções VHDL, como instruções IF-ELSE ou CASE para isso parte do exercício.

2. Use as chaves SW7-4 e SW3-0 para as entradas X e Y, respectivamente, e use SW8 para a entrada c_{in} . Conecte a soma de quatro bits e a execução produzida pela operação $X + Y$ ao LEDR das luzes vermelhas. Exibir os valores de BCD de X e Y nos segmentos de 7 exibe HEX5 e HEX3 e exibir o resultado S1,S0 em HEX1 e HEX0.



3. Como seu circuito processa apenas dígitos BCD, verifique os casos em que a entrada X ou Y é maior que nove. Se isso ocorrer, indique um erro acendendo a luz vermelha LEDR9.
4. Inclua as atribuições de pinos necessárias para a placa da série DE0, compile o circuito e faça o download no chip FPGA.
5. Teste seu circuito tentando valores diferentes para os números X, Y e c_{in} .

PARTE 5

Na Parte 4, você criou o código VHDL para um somado BCD. Uma abordagem diferente para descrever o cabeçalho no código VHDL é especificar um algoritmo como o representado pelo seguinte pseudocódigo:

```
1   $T_0 = A + B + c_0$ 
2  if ( $T_0 > 9$ ) then
3       $Z_0 = 10$ ;
4       $c_1 = 1$ ;
5  else
6       $Z_0 = 0$ ;
7       $c_1 = 0$ ;
8  end if
9   $S_0 = T_0 - Z_0$ 
10  $S_1 = c_1$ 
```

É razoavelmente simples ver qual circuito pode ser usado para implementar esse pseudocódigo. As linhas 1 e 9 representam somadores, as linhas 2-8 correspondem aos multiplexadores e o teste para a condição $T_0 > 9$ requer comparadores.

Você está escrevendo o código VHDL que responde a esse pseudocódigo. Observe que você pode executar operações de adição em seu código VHDL em vez das subtrações mostradas na linha 9.

O objetivo dessa parte do exercício é comparar os resultados do compilador VHDL em projetos de circuitos usando instruções IF-ELSE juntamente com os operadores VHDL $>$ e $+$. Execute as seguintes etapas:

1. Crie um novo projeto Quartus para o seu código VHDL. Use as chaves SW7-4 e SW3-0 para as entradas A e B, respectivamente, e use a chave SW8 para o c_{in} . O valor de A deve ser exibido no HEX5 de 7 segmentos, enquanto B deve estar no HEX3. Exiba a soma do BCD, S_1S_0 , em HEX1 e HEX0.
2. Use a ferramenta RTL Viewer do Quartus para examinar o resultado produzido pela compilação do seu código VHDL. Compare o circuito com o que você projetou na Parte 4.
3. Faça o download do seu circuito na placa da série DE0 e teste-o tentando valores diferentes para os números A e B.



PARTE 6

Projete um circuito combinacional que converta um número binário de 6 bits em um número decimal de 2 dígitos representado no formato BCD. Use as chaves SW5-0 para inserir o número binário e os displays de 7 segmentos HEX1 e HEX0 para exibir o número decimal. Implemente seu circuito na placa DE0 e demonstre sua funcionalidade.