

Sistemas Digitais para Computação

Introdução básica à placa DE2 da Altera: Chaves, LEDs e multiplexadores

Este exercício tem como propósito ensinar como conectar dispositivos simples de entrada e saída ao chip de FPGA e implementar um circuito que utilize a placa DE2. Serão usadas as chaves SW_{17-0} da placa DE2 como entradas do circuito. Serão também usados os LEDs e displays de 7 segmentos como dispositivos de saída.

Parte I

A placa DE2 provê 18 chaves, chamadas SW_{17-0} , que podem ser usadas como entradas do circuito, e 18 LEDs vermelhos, chamados $LEDR_{17-0}$, que podem ser usados para exibir valores de saída. A figura 1 apresenta um módulo simples implementado em Verilog que usa essas chaves e mostra os estados utilizando os LEDs. Já que há 18 chaves e LEDs, é conveniente representá-los como vetores no código Verilog. Desta maneira, foi usada uma única instrução de atribuição para todas as 18 saídas LEDR, que seriam equivalentes às seguintes atribuições individuais:

```
assign LEDR[17] = SW[17];  
assign LEDR[16] = SW[16];  
...  
assign LEDR[0] = SW[0];
```

A placa DE2 tem conexões fixas entre o chip FPGA, as chaves e os LEDs. Para usar SW_{17-0} e $LEDR_{17-0}$ é necessário incluir no seu projeto no quartus II a correlação correta entre os terminais do FPGA e os dispositivos da placa (pin assignments). Esta informação é dada no manual do usuário do DE2. O manual especifica que SW_0 é conectada ao FPGA no pino N25 e $LEDR_0$ no pino AE23. Uma boa maneira de garantir a correta atribuição dos pinos é importar o arquivo DE2_pin_assignments.csv.

É importante observar que a atribuição dos pinos apresentada no arquivo DE2_pin_assignments.csv somente é útil se os nomes dados no arquivo forem iguais aos usados nos nomes de portas do seu módulo Verilog.

```
// Simple module that connects the SW switches to the LEDR lights  
module part1 (SW, LEDR);  
    input [17:0] SW;      // toggle switches  
    output [17:0] LEDR;   // red LEDs  
  
    assign LEDR = SW;  
endmodule
```

Figura 1

Realize os seguintes passos para implementar um circuito correspondente ao código da

figura 1 na placa DE2:

1. Crie um novo projeto no Quartus II para o circuito. Selecione Cyclone II EP2C35F672C6 como chip alvo, o qual é o chip encontrado na placa DE2 da Altera.
2. Crie um módulo Verilog para o código da figura 1 e o inclua no projeto.
3. Inclua no seu projeto o arquivo de correlação de pinos. Compile o projeto.
4. Envie para o FPGA o circuito compilado. Teste o circuito alterado o estado das chaves e observando os LEDs.

Parte II

A Figura 2a mostra um circuito que implementa um multiplexador 2-para-1 com a entrada de seleção s . Se $s = 0$, então a saída m do multiplexador é igual a entrada x ; se $s = 1$ então a saída é igual a y . A parte b da figura apresenta a tabela verdade deste multiplexador e a parte c apresenta o símbolo do circuito.

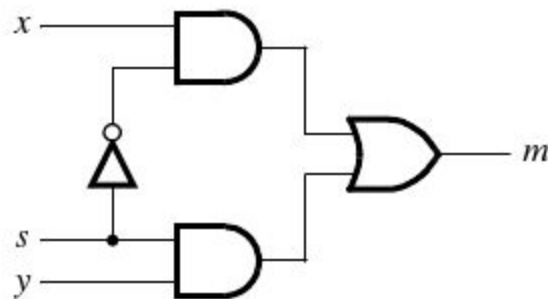


Figura 2a

s	m
0	x
1	y

Figura 2b

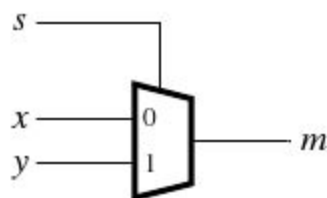


Figura 2c

O multiplexador pode ser descrito em Verilog da seguinte maneira:

```
assign m = (~s & x) | (s & y);
```

Escreva um módulo Verilog que inclua 8 instruções atribuição semelhantes à mostrada acima para descrever o circuito apresentado na figura 3a. Este circuito tem 2 portas de 8 bits de entrada cada, x e y , e produz uma saída m de 8 bits. Se $s = 0$ então $m = x$, enquanto que se $s = 1$ então $m = y$. Esse circuito é referenciado como um multiplexador 2-para-1 de 8 bits de largura. Este circuito tem o símbolo apresentado na Figura 3b, onde x , y , e m são destacados como fios de 8 bits. Realize os passos abaixo:

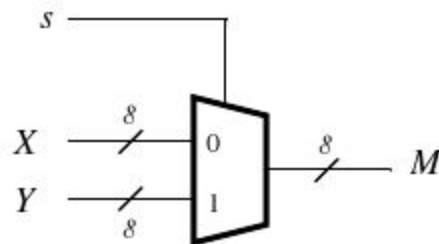


Figura 3b

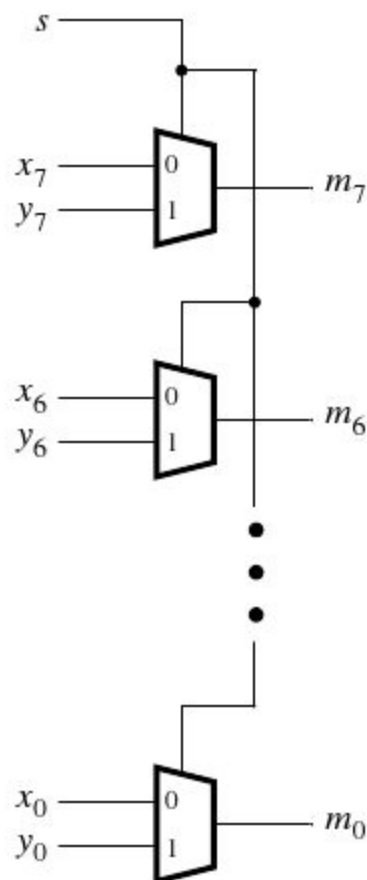


Figura 3a

1. Crie um novo projeto do Quartus II para o circuito.
2. Inclua o seu arquivo Verilog para o multiplexador de 8 bits 2-para-1 no seu projeto.
Use a chave SW_{17} da placa DE2 como a entrada s , as chaves SW_{7-0} com a entrada x e

- SW_{15-8} como a entrada y .
3. Inclua no seu projeto o arquivo com o mapeamento de pinos da placa DE2. Conforme discutido na parte 1, este mapeamento garantirá que as portas definidas no seu código Verilog usarão os terminais do Cyclone II FPGA conectados às chaves SW, e as portas de saídas do seu código Verilog usarão os terminais do FPGA conectados nos leds LEDR e LEDG.
 4. Compile o projeto.
 5. Envie o circuito compilado para o FPGA. Teste o multiplexador mudando o estado das chaves e observando os LEDs.

Parte III

Na figura 2 foi apresentado um multiplexador 2-para-1 que seleciona entre duas entradas x e y . Para esta parte considere o circuito no qual a saída m é selecionada a partir de 5 entradas u , v , w , x , e y . A figura 4a mostra como pode ser construído um multiplexador 5-para-1 utilizando-se 4 multiplexadores 2-para-1. O circuito usa 3 bits, $s_2s_1s_0$ para selecionar algum das entradas e implementa a tabela verdade apresentada na figura 4b. O símbolo do circuito para este multiplexador é dado na parte c da figura.

Veja que a figura 3 apresentada a construção de um multiplexador de 8 bits 2-para-1 utilizando-se 8 instâncias do multiplexador 2-para-1. A figura 5 emprega este conceito para definir um multiplexador de 3 bits 5-para-1. Ele contém 3 instâncias do circuito apresentado na figura 4a.

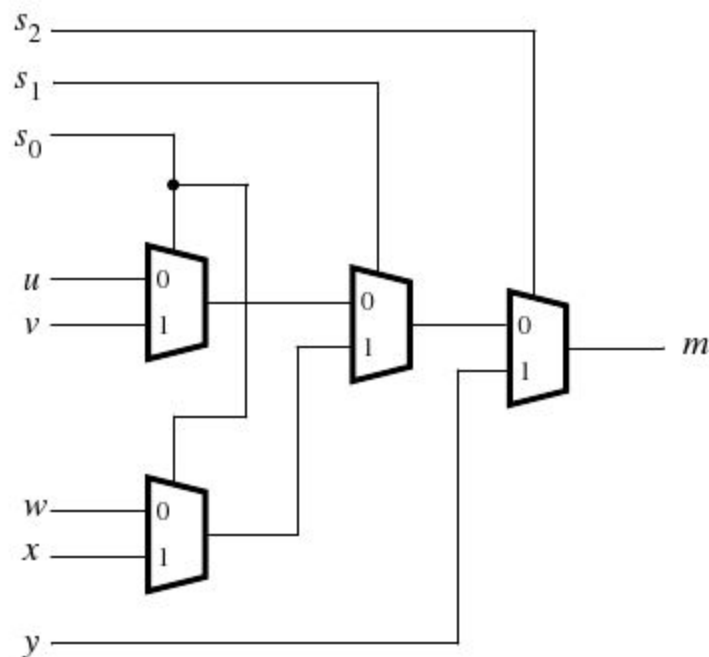
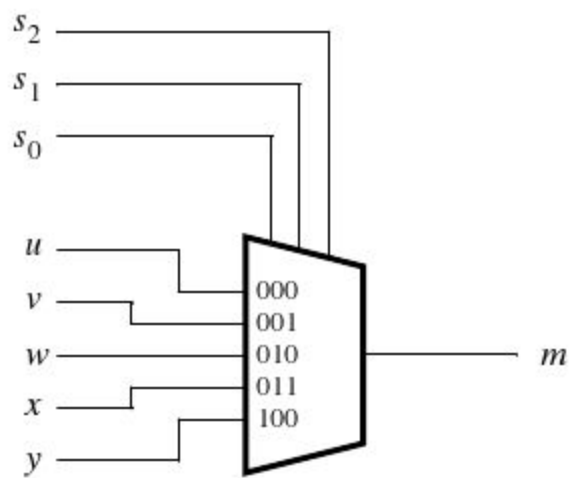


Figura 4a

s_2	s_1	s_0	m
0	0	0	u
0	0	1	v
0	1	0	w
0	1	1	x
1	0	0	y
1	0	1	y
1	1	0	y
1	1	1	y

Figura 4b

Figura 4c



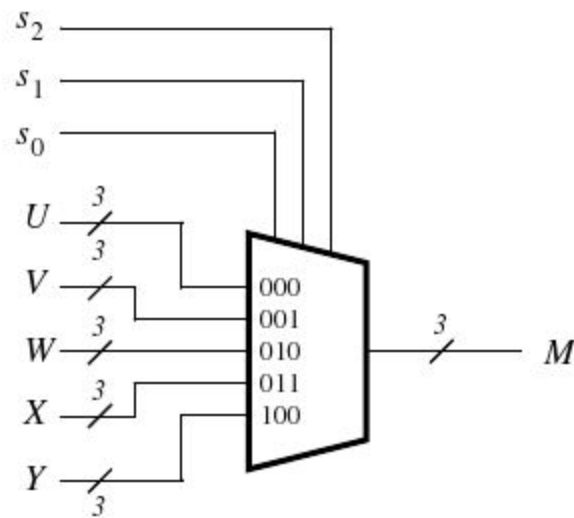


Figura 5

Realize os passos abaixo para a implementação do multiplexador de 3 bits de largura 5-para-1:

1. Crie um novo projeto do Quartus II para seu circuito.
2. Crie um módulo Verilog para o multiplexador de 3 bits 5-para-1. Conecte as entradas de seleção nas chaves SW_{17-15} , e use as 15 chaves restantes, SW_{14-0} para prover as cinco entradas de 3 bits cada, u até y. Conecte as chaves SW aos LEDs vermelhos LEDR e conecte a saída m aos LEDs verdes LEDG₂₋₀.
3. Inclua no seu projeto o mapeamento dos terminais para a placa DE2. Compile o projeto.
4. Envie o circuito compilado para o FPGA. Teste o multiplexador de 3 bits 5-para-1 mudando o estado das chaves e observando os LEDs. Verifique se cada uma das entradas u até y pode ser selecionada para a saída m.

Parte IV

A figura 6 apresenta um módulo decodificador de 7 segmentos com uma entrada de 3 bits, $c_2c_1c_0$. Este decodificador produz 7 saídas que são usadas para apresentar um caracter no display de 7 segmentos. A tabela 1 lista os caracteres que devem ser apresentados para cada valor de $c_2c_1c_0$. Para manter o projeto simples, somente 4 caracteres serão incluídos na tabela (há também o caracter "blank", ou seja, a possibilidade de nenhum caracter ser apresentado. Os códigos 100 até 111 selecionam esta possibilidade).

Os sete segmentos do display são indicados pelos índices de 0 a 6 conforme apresentado na figura. Cada segmento é iluminado quando recebe um 0 lógico. Escreva uma módulo Verilog que implemente as funções lógicas que representam os circuitos necessários para ativar cada um dos 7 segmentos. Use somente instruções de atribuição no seu código Verilog descritas através de expressões booleanas.

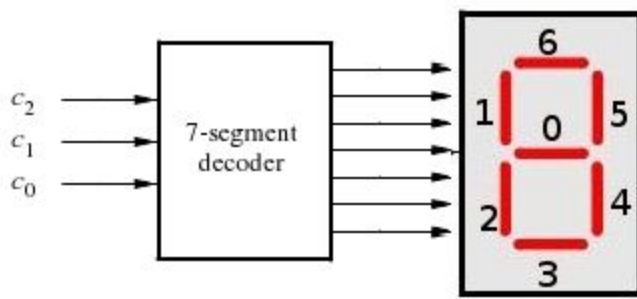


figura 6

$c_2 c_1 c_0$	Character
000	H
001	E
010	L
011	O
100	
101	
110	
111	

Tabela 1

Siga os passos abaixo:

1. Crie um novo projeto do Quartus II para o circuito.
2. Crie um módulo Verilog para o decodificador de 7 segmentos. Conecte as entradas $c_2 c_1 c_0$ às chaves SW_{2-0} e conecte as saídas do decodificador no display HEX0 da placa DE2. Os segmentos deste display são chamados $HEX0_0, HEX0_1, \dots, HEX0_6$, correspondentes à figura 6. Deverá ser declarada a seguinte porta de 7 bits

output [0:6] HEX0;

no código Verilog o nome das portas deverão corresponder aos nomes dos pinos citados no arquivo DE2_pin_assignments.csv.

3. Compile o projeto.
4. Envie o projeto para o FPGA. Teste a funcionalidade do circuito mudando o estado das chaves SW_{2-0} e observe o display.

Parte V

Considere o circuito apresentado na figura 7. Ele usa um multiplexador 5-para-1 com entradas de 3 bits de comprimento. Este multiplexador é usado para selecionar um caracter para ser exibido no display de 7 segmentos. Utilizando o decodificador desenvolvido na parte IV este circuito pode apresentar os caracteres H, E, L, O e 'blank'. Os códigos dos caracteres são os apresentados na tabela 1 utilizando as chaves SW_{14-0} e o caracter especificamente selecionado para a apresentação é indicado pelas chaves SW_{17-15} .

A estrutura deste projeto é dada na figura 8. Note que foram usados circuitos das partes III e IV como blocos neste código. A tarefa desta parte consiste em alterar o código para utilizar 5 displays de 7 segmentos e 5 multiplexadores ao invés de apenas uma unidade de cada bloco. O propósito do circuito é apresentar qualquer palavra nos 5 displays quando as chaves SW_{17-15} são mudadas. Como um exemplo, se a palavra apresentada for HELLO, então o circuito deve produzir o padrão de saída ilustrado na tabela 2.

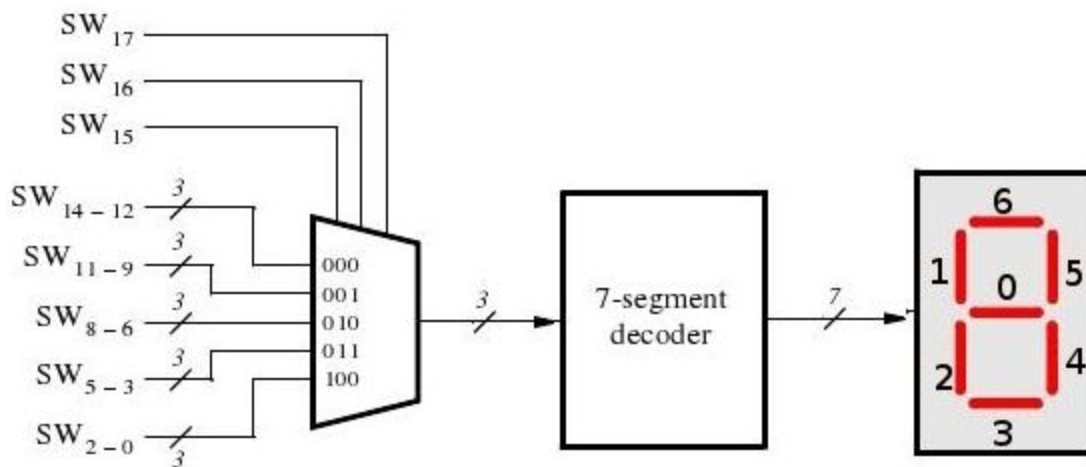


Figura 7


```

module part5 (SW, HEX0);
    input [17:0] SW;      // toggle switches
    output [0:6] HEX0;    // 7-seg displays

    wire [2:0] M;

    mux_3bit_5to1 M0 (SW[17:15], SW[14:12], SW[11:9], SW[8:6], SW[5:3], SW[2:0], M);
    char_7seg H0 (M, HEX0);
endmodule

// implements a 3-bit wide 5-to-1 multiplexer
module mux_3bit_5to1 (S, U, V, W, X, Y, M);
    input [2:0] S, U, V, W, X, Y;
    output [2:0] M;

    ... code not shown

endmodule

// implements a 7-segment decoder for H, E, L, O, and 'blank'
module char_7seg (C, Display);
    input [2:0] C;      // input code
    output [0:6] Display; // output 7-seg code

    ... code not shown

endmodule

```

Figura 8

SW_{17} SW_{16} SW_{15}	Character pattern				
000	H	E	L	L	O
001	E	L	L	O	H
010	L	L	O	H	E
011	L	O	H	E	L
100	O	H	E	L	L

Tabela 2

Realize os seguintes passos:

1. Crie um novo projeto do Quartus II para o circuito.
2. Inclua o módulo Verilog no seu projeto. Conecte as chaves SW_{17-15} nas portas de seleção de cada uma das instâncias do multiplexador. Conecte as chaves SW_{14-0} a cada instância do multiplexador de maneira a apresentar um padrão igual a tabela 2. Conecte as saídas de cada um dos 5 multiplexadores nos displays HEX4, HEX3, HEX2, HEX1, e HEX0.

3. Inclua no projeto a correlação entre os nomes dos pinos da placa e do chip. Compile o projeto.
4. Envie o circuito compilado para o FPGA. Teste funcionalmente o circuito selecionando os caracteres através das chaves SW_{14-0} e então mude as chaves SW_{17-15} para observar a rotação dos caracteres.

Parte VI

Extenda o projeto da parte V de maneira a usar todos os displays de 7 segmentos da placa DE2. O circuito deve ser apto a apresentar palavras com 5 (ou menos) caracteres nos 8 displays, e rotacionar a palavra apresentada quando as chaves SW_{17-15} forem mudadas. Se a palavra apresentada é HELLO, então o circuito deve produzir o padrão apresentado na tabela 3.

SW_{17}	SW_{16}	SW_{15}	Character pattern						
000					H	E	L	L	O
001					H	E	L	L	O
010				H	E	L	L	O	
011			H	E	L	L	O		
100			E	L	L	O			H
101			L	L	O			H	E
110			L	O			H	E	L
111			O			H	E	L	L

Tabela 3

Realize os seguintes passos:

1. Crie um novo projeto do Quartus II para o circuito.
2. Inclua o módulo Verilog no seu projeto. Conecte as chaves SW_{17-15} nas portas de seleção de cada uma das instâncias do multiplexador. Conecte as chaves SW_{14-0} a cada instância do multiplexador de maneira a apresentar um padrão igual a tabela 3. (Dica: Para algumas entradas dos multiplexadores deverá ser selecionado o caracter 'blank'.) Conecte as saídas de cada um dos 5 multiplexadores nos displays HEX7, ..., HEX0.
3. Inclua no projeto a correlação entre os nomes dos pinos da placa e do chip. Compile o projeto.
4. Envie o circuito compilado para o FPGA. Teste funcionalmente o circuito selecionando os caracteres através das chaves SW_{14-0} e então mude as chaves SW_{17-15} para observar a rotação dos caracteres.