Avaliação

Projeto de circuitos para interface com ULA

Introdução

Uma Unidade lógica e Aritmética (ULA) é um dispositivo que realiza operações lógicas e aritméticas sobre duas palavras digitais. De maneira geral, uma ula recebe dois valores tratados como operandos e uma entrada auxiliar de controle permite especificar qual operação será realizada sobre os dois operandos de entrada. Dessa maneira, para o correto funcionamento de uma ULA, são fundamentais dois conceitos básicos: o controle do fluxo de dados e a utilização de circuitos lógicos que implementem as operações disponíveis na ULA projetada. As ULAs podem ser embarcadas em microcontroladores / microprocessadores ou disponibilizadas em forma de circuito integrado.

O circuito integrado comercial SN54/74LS181 implementa uma ULA de 4-*bits*, ou seja, ela é capaz de realizar operações lógicas e aritméticas sobre dois operandos de 4 bits. Além disso, essa ULA possui 16 operações lógicas / aritméticas que podem ser selecionadas através das suas entradas de seleção. Maiores informações podem ser obtidas no *datasheet* do dispositivo SN54/74LS181.

Atividade

Descreva em *Verilog* o circuito mostrado na figura 1. Note que o circuito é baseado na ULA SN54/74LS181, de 4-*bits*. Dessa maneira, deverão ser implementadas apenas 8 instruções da ULA que sejam compatíveis com os barramentos de entrada e saída do circuito proposto. Por exemplo, a instrução soma de dois números de 4 *bits* não deve ser utilizada, uma vez que é necessário um sinal de *carry out* não disponível na arquitetura do circuito proposto.

O circuito proposto utiliza 2 contadores, um de 2-*bits* para controlar a operação de um registrador de deslocamento programável e, outro, de 3-*bits*, para a seleção da operação desejada para a ULA. Tais circuitos contadores foram utilizados para reduzir a quantidade de botões ou chaves necessárias para o teste.

Também fazem parte do circuito proposto 3 registros: Reg A, Reg B e Reg C que possuem a finalidade de armazenar os valores dos operandos de entrada fornecidos à ULA e do resultado da operação realizada pela ULA, respectivamente. O circuito proposto conta, ainda, com um registrador de deslocamento programável de 4-*bits* capaz de realizar as operações indicadas na tabela 1. Além disso, está presente um circuito multiplexador que possui a finalidade de controlar o valor exibido na saída do circuito, sedo possível escolher entre o resultado da operação da ULA e o resultado da operação do registrador de deslocamento.

O fluxo de dados do circuito proposto deve ser acompanhado através de *displays* de 7-segmentos conectados aos barramentos, conforme indicado na figura 1, utilizando circuitos decodificadores BCD para 7-segmentos que não estão mostrados. A figura 2 mostra o código *Verilog* de um decodificador BCD para 7-segmentos que deve ser alterado para fornecer os seguintes símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, b, C, d, E, e F.

É importante notar que os sinais de *clock* dos dois contadores devem ser independentes do sinal de *clock* geral do sistema.

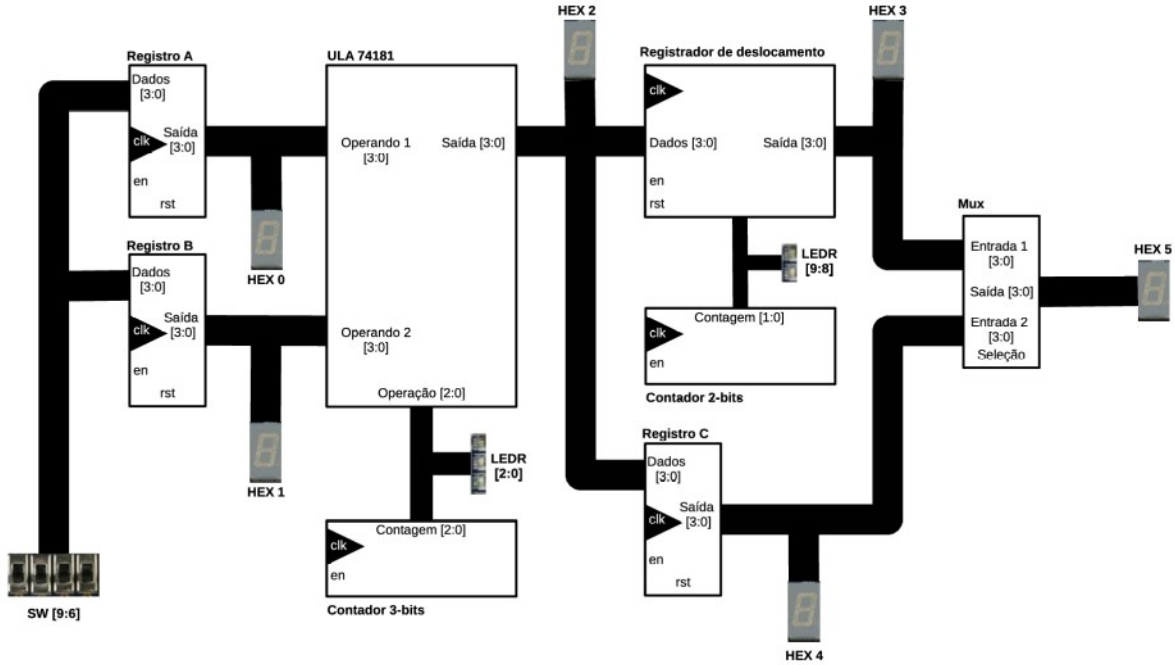

Figura 1: Sistema digital proposto que utiliza uma ULA 74181.

Tabela 1: Operações disponíveis para o registrador de deslocamento programável.

Palavra	**Descrição**
00	Carregar os dados no registrador.
01	Deslocamento à direita com preenchimento de zeros.
10	Deslocamento à esquerda com preenchimento de zeros.
11	Deslocamento à direita mantendo o bit mais significativo.

```
module decod_display (codigo_BCD, display);
        input      [3:0]   codigo_BCD;
        output reg [6:0]   display;
        // leds acendem com nivel logico 0
        //              leds ->  gfe dcba
        parameter   branco  = 7'b111_1111; //h7F
        parameter   zero    = 7'b100_0000; //h40
        parameter   um      = 7'b111_1001; //h79
        parameter   dois    = 7'b010_0100; //h24
        parameter   tres    = 7'b011_0000; //h30
        parameter   quatro  = 7'b001_1001; //h19
        parameter   cinco   = 7'b001_0010; //h12
        parameter   seis    = 7'b000_0010; //h02
        parameter   sete    = 7'b111_1000; //h78
        parameter   oito    = 7'b000_0000; //h00
        parameter   nove    = 7'b001_1000; //h18
        always @(codigo_BCD)
          case(codigo_BCD)
             0:        display = zero;
             1:        display = um;
             2:        display = dois;
             3:        display = tres;
             4:        display = quatro;
             5:        display = cinco;
             6:        display = seis;
             7:        display = sete;
             8:        display = oito;
             9:        display = nove;
             default:  display = branco;
          endcase
endmodule
```

Figura 2: Descrição Verilog de um decodificador BCD para 7-segmentos que exibe os algarismos de 0 a 9. O código deve ser alterado para exibir os símbolos adicionais: A, b, C, d, E, e F.

Avaliação

A avaliação será realizada sobre um relatório que mostra o desenvolvimento do projeto. Esse relatório deve conter as explicações das decisões tomadas durante o projeto, incluindo explicações sobre as simulações realizadas. De maneira geral, o projeto e o relatório devem seguir as seguintes orientações:

1. Simulação dos módulos individuais em arquivos separados com a seguinte denominação de arquivos: **registrador.v**, **contador_2b.v**, **contador_3b.v**, **ula.v** e **reg_desloc.v** utilizando arquivos de testbench específicos (**registrador_TB.v**, **contador_2b_TB**, **contador_3b_TB**, **ula_TB.v** e **reg_desloc_TB.v**). Simulações utilizando o editor de ondas não serão consideradas. (5 pontos)
2. Descrição de cada módulo *Verilog* em arquivo separado e utilização por componente (nível de abstração estrutural) no código do sistema completo no arquivo **sistema.v**. Deverá ser

apresentada a visualização RTL para comprovar a correta ligação entre os módulos. Deverá ser realizada também a simulação do sistema completo por meio de um arquivo de testbench (**sistema_TB.v**). (5 pontos).

O mapeamento da pinagem da placa de desenvolvimento deverá ser entregue em arquivo .csv seguindo as associações mostradas na tabela 2.

Dessa maneira o projeto deverá ser entregue em arquivo .zip contendo os seguintes arquivos:

1. Relatório em formato .pdf
2. Pinagem em formato .csv
3. registrador.v
4. contador_2b.v
5. contador_3b.v
6. ula.v
7. reg_desloc.v
8. registrador_TB.v
9. contador_2b_TB.v
10. contador_3b_TB.v
11. ula_TB.v
12. reg_desloc_TB.v
13. sistema.v
14. sistema_TB.v

Mapeamento dos pinos na placa DE10 – Lite

A tabela 2 indica a associação de pinos a ser utilizada nesse projeto. A figura 1 também mostra as conexões de algumas chaves, leds e displays de 7-segmentos. Maiores informações podem ser encontradas no manual da placa de desenvolvimento.

Tabela 2: Mapeamento de pinos para o circuito proposto.

Sinal	**Componente da placa**	**Pino da FPGA**	**Descrição**	***I/O standard***
Clock do sistema	MAX10_CLK1_50	PIN_P11	50 MHz clock input(Bank 3B	3.3-V LVTTL
Clock do contador de 3 bits	KEY 0	PIN_B8	Push-button[0]	3.3 V SCHMITT TRIGGER
Clock do contador de 2 bits	KEY 1	PIN_A7	Push-button[1]	3.3 V SCHMITT TRIGGER
Reset	SW0	PIN_C10	Slide Switch[0]	3.3-V LVTTL
Dados – bit[0] lsb	SW6	PIN_A13	Slide Switch[6]	3.3-V LVTTL
Dados – bit[1]	SW7	PIN_A14	Slide Switch[7]	3.3-V LVTTL
Dados – bit[2]	SW8	PIN_B14	Slide Switch[8]	3.3-V LVTTL
Dados – bit[3] msb	SW9	PIN_F15	Slide Switch[9]	3.3-V LVTTL
Habilita Reg. A	SW1	PIN_C11	Slide Switch[1]	3.3-V LVTTL
Habilita Reg. B	SW2	PIN_D12	Slide Switch[2]	3.3-V LVTTL
Habilita Reg. C	SW3	PIN_C12	Slide Switch[3]	3.3-V LVTTL
Habilita Reg. Des.	SW4	PIN_A12	Slide Switch[4]	3.3-V LVTTL
Os displays de 7-segmentos HEX[5:0] devem ser conectados conforme indicado na figura 1 através de decodificadores BCD para 7-segmentos.				
O valor dos contadores de 2 e 3-bits devem ser mostrados nos leds LDR, conforme indicado na figura 1.				