

Pontifícia Universidade Católica de Minas Gerais
Instituto de Ciências Exatas e Informática – ICEI
Arquitetura de Computadores I

ARQ1_ Aula_08

Tema: Introdução à linguagem Verilog e simulação em Logisim

Preparação

Vídeos recomendados

<https://www.youtube.com/watch?v=Zkkck2MovCc>
<https://www.youtube.com/watch?v=3NUwEshZSWQ>
<https://www.youtube.com/watch?v=YCq9L0hAWyM>

Atividade: Projeto de unidade lógica e aritmética

01.) Projetar e descrever em Verilog, usando portas nativas, uma unidade aritmética (AU) com um somador completo, utilizando o modelo compacto para a “meia-soma”, para operandos de 04 bits (sinal=1+amplitude=3). O nome do arquivo deverá ser Exemplo_0801.v, e poderá seguir o modelo descrito abaixo. Incluir previsão de testes. Simular o módulo no Logisim e apresentar layout do circuito e subcircuitos.

Exemplo:

“vai-um” final
↓
$$\begin{array}{ccccccc} s_{41} & s_{31} & s_{21} & s_{01} & 0 & \leftarrow & \text{“vai-um” inicial arbitrário} \\ & a_3 & a_2 & a_1 & a_0 & + & \\ & b_3 & b_2 & b_1 & b_0 & & \\ \hline s_{04} & s_{03} & s_{02} & s_{01} & s_{00} & & \end{array}$$

```

// -----
// Exemplo_0801 - FULL ADDER
// Nome: xxx yyy zzz
// Matricula: 999999
// -----

// -----
// half adder
// -----
module halfAdder (output s1,
                  output s0,
                  input  a,
                  input  b);

// descrever por portas
xor  XOR1 ( s0, a, b );
and  AND1 ( s1, a, b );

endmodule // halfAdder

// -----
// full adder
// -----
module fullAdder ( output s1,
                  output s0,
                  input  a,
                  input  b,
                  input  carryIn );

// descrever por portas e/ou modulos
// (valores arbitrarios escolhidos apenas para exemplo)
not  NOT1 ( s1, a ); // valor arbitrario
not  NOT2 ( s0, b ); // valor arbitrario

endmodule // fullAdder

module test_fullAdder;
// ----- definir dados
    reg  [3:0] x;
    reg  [3:0] y;
    wire [3:0] carry; // "vai-um"
    wire [4:0] soma;

// halfAdder HAO ( carry[0], soma[0], x[0], y[0] );
fullAdder FA0 ( carry[0], soma[0], x[0], y[0], 1'b0 );

// ----- parte principal
initial begin
    $display("Exemplo0801 - xxx yyy zzz - 999999");
    $display("Test ALU's full adder");

// projetar testes do somador completo
end

endmodule // test_fullAdder

```

- 02.) Projetar e descrever em Verilog, usando portas nativas, uma unidade aritmética (AU) com um subtrator completo, utilizando o modelo compacto para a “meia-diferença”, para operandos de 04 bits (sinal=1+amplitude=3). O nome do arquivo deverá ser Exemplo_0802.v, e poderá seguir o modelo descrito anteriormente. Incluir previsão de testes. Simular o módulo no Logisim e apresentar *layout* do circuito e subcircuitos.

Exemplo:

$$\begin{array}{r}
 \text{“vem-um” final} \\
 \downarrow \\
 \begin{array}{r}
 s_{41} \ s_{31} \ s_{21} \ s_{01} \ 0 \ \leftarrow \text{“vem-um” inicial arbitrário} \\
 a_3 \ a_2 \ a_1 \ a_0 \ - \\
 b_3 \ b_2 \ b_1 \ b_0 \\
 \hline
 s_{04} \ s_{03} \ s_{02} \ s_{01} \ s_{00}
 \end{array}
 \end{array}$$

- 03.) Projetar e descrever em Verilog, usando portas nativas, uma unidade lógica (LU) com um comparador para igualdade, para 04 bits (sinal=1+amplitude=3). O nome do arquivo deverá ser Exemplo_0803.v, e poderá seguir o modelo descrito anteriormente. Incluir previsão de testes. Simular o módulo no Logisim e apresentar *layout* do circuito e subcircuitos.

Exemplo:

$$\begin{array}{r}
 x \ y \ s = \text{igual} \\
 0 \ 0 \ 1 \\
 0 \ 1 \ 0 \\
 1 \ 0 \ 0 \\
 1 \ 1 \ 1 \\
 \\
 \begin{array}{r}
 a_3 \ a_2 \ a_1 \ a_0 \ \sim \wedge \leftarrow \text{“xnor”} \\
 b_3 \ b_2 \ b_1 \ b_0 \\
 \hline
 s_3 \ s_2 \ s_1 \ s_0
 \end{array} \\
 \\
 s = f (s_3, s_2, s_1, s_0); \quad // \text{ todos iguais a 1}
 \end{array}$$

04.) Projetar e descrever em Verilog, usando portas nativas, uma unidade lógica (LU) com um comparador para diferença, para 04 bits (sinal=1+amplitude=3).

O nome do arquivo deverá ser Exemplo_0804.v, e poderá seguir o modelo descrito anteriormente.

Incluir previsão de testes.

Simular o módulo no Logisim e

apresentar *layout* do circuito e subcircuitos.

Exemplo:

x y s = diferença

0 0 0

0 1 1

1 0 1

1 1 0

$a_3 \ a_2 \ a_1 \ a_0 \wedge \leftarrow \text{"xor"}$

$b_3 \ b_2 \ b_1 \ b_0$

$s_3 \ s_2 \ s_1 \ s_0$

$s = f (s_3, s_2, s_1, s_0);$ // pelo menos algum diferente de 0

05.) Projetar e descrever em Verilog, usando portas nativas, uma unidade lógica (LU) com um módulo para calcular o complemento de 2 de um valor binário com 04 bits.

O nome do arquivo deverá ser Exemplo_0805.v, e poderá seguir o modelo descrito anteriormente.

Incluir previsão de testes.

Simular o módulo no Logisim e

apresentar *layout* do circuito e subcircuitos.

DICA: Construir um subcircuito para calcular

o complemento de 1 e

usar um somador completo de 4 bits.

Extras

06.) Projetar e descrever em Verilog, usando portas nativas, uma unidade aritmética (AU) com um somador algébrico (somador completo com uma das entradas invertida, caso seja escolhida a diferença, conforme seleção), para calcular a igualdade ou a diferença, para 04 bits (sinal=1+amplitude=3), selecionável pela entrada *carryIn* do primeiro subcircuito (soma=*carryIn*=0; subtração=*carryIn*=1). O nome do arquivo deverá ser *Exemplo_0806.v*, e poderá seguir o modelo descrito anteriormente. Incluir previsão de testes. Simular o módulo no Logisim e apresentar *layout* do circuito e subcircuitos. DICA: Usar o subcircuito para calcular o complemento de 1 condicionado à seleção.

07.) Projetar e descrever em Verilog, usando portas nativas, uma unidade lógica (LU) com um comparador para calcular simultaneamente a igualdade e a diferença, para 04 bits (sinal=1+amplitude=3), e selecionar o resultado por uma entrada extra (chave). O nome do arquivo deverá ser *Exemplo_0807.v*, e poderá seguir o modelo descrito anteriormente. Incluir previsão de testes. Simular o módulo no Logisim e apresentar *layout* do circuito e subcircuitos.