

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=cG7wemiantQ>
<https://www.youtube.com/watch?v=YCq9L0hAWyM>

Orientação geral:

Apresentar uma forma de solução em formato texto (.txt).

Os arquivos para simulação em Logisim (.circ) deverão ser identificados internamente e entregues, acompanhados (ou não) de figuras equivalentes exportadas pela ferramenta.

Separar as versões completas (a) das simplificadas (b).

Programas em Verilog deverão ser entregues em formato (.v) com previsão de testes.

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

↓

S₄₁ S₃₁ S₂₁ S₀₁ 0 ← “vai-um” inicial arbitrário

a₃ a₂ a₁ a₀ +

b₃ b₂ b₁ b₀

S₀₄ S₀₃ S₀₂ S₀₁ S₀₀

```

// -----
// 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:

```

“vem-um” final
↓
S41 S31 S21 S01 0 ← “vem-um” inicial arbitrário
  a3 a2 a1 a0 -
  b3 b2 b1 b0
-----
S04 S03 S02 S01 S00

```

- 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:

```

x y s = igual
0 0 1
0 1 0
1 0 0
1 1 1

  a3 a2 a1 a0 ~^ ← “xnor”
  b3 b2 b1 b0
-----
S3 S2 S1 S0

s = f ( S3, S2, S1, S0 ); // todos iguais a 1

```

04.) Projetar e descrever em Verilog, usando portas nativas, uma unidade lógica (LU) com um comparador para desigualdade, 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 = desigualdade

0 0 0

0 1 1

1 0 1

1 1 0

a₃ a₂ a₁ a₀ ^ ← "xor"

b₃ b₂ b₁ b₀

s₃ s₂ s₁ s₀

s = f (s₃, s₂, s₁, s₀); // 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 subtração, conforme seleção pela entrada **carryIn** do primeiro subcircuito (soma=**carryIn**=0; subtração=**carryIn**=1), para calcular simultaneamente a igualdade e a desigualdade, para 04 bits (sinal=1+amplitude=3), e selecionar o resultado por uma segunda entrada extra (chave). 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 (=0) ou a desigualdade (=1), 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. DICA: Usar os subcircuitos 03 e 04 acima.