

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

Como preparação para o início das atividades, recomendam-se

- a.) leitura prévia do resumo teórico, do detalhamento na apostila e referências recomendadas
- b.) estudo e testes dos exemplos
- c.) assistir aos seguintes vídeos:

<https://www.youtube.com/watch?v=Zkkck2MovCc>  
<https://www.youtube.com/watch?v=cG7wemiantQ>  
<https://www.youtube.com/watch?v=YCq9L0hAWyM>

### Orientação geral:

Apresentar todas as soluções em apenas um arquivo com formato texto (.txt).  
As implementações e testes dos exemplos em Verilog (.v) fornecidos como pontos de partida, também fazem parte da atividade e deverão ser entregues os códigos fontes separadamente.  
As saídas de resultados, opcionalmente, poderão ser copiadas ao final do código, como comentários.

Outras formas de solução são opcionais; e, se entregues, contarão como atividades extras (c, py). Os programas com funções desenvolvidas em C ou Python (usar modelos para verificação automática de testes das respostas), se entregues, também deverão estar em arquivos separados, com o código fonte, para serem compilados e testados. As execuções deverão, preferencialmente, serem testadas mediante uso de entradas e saídas padrões e os dados/resultados usados para testes armazenados em arquivos textos. Os resultados poderão ser anexados ao código, ao final, como comentários.

Os *layouts* de circuitos deverão ser entregues no formato (.circ), identificados internamente. Figuras exportadas pela ferramenta serão aceitas como arquivos para visualização, mas não terão validade para fins de avaliação. Separar as versões completas (a) das simplificadas (b).

Arquivos em formato (.pdf), fotos, cópias de tela ou soluções manuscritas serão aceitas como recursos suplementares para visualização e não terão validade para fins de avaliação.

## 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

↓

S<sub>41</sub> S<sub>31</sub> S<sub>21</sub> S<sub>01</sub> 0 ← “vai-um” inicial arbitrário

a<sub>3</sub> a<sub>2</sub> a<sub>1</sub> a<sub>0</sub> +

b<sub>3</sub> b<sub>2</sub> b<sub>1</sub> b<sub>0</sub>

---

S<sub>04</sub> S<sub>03</sub> S<sub>02</sub> S<sub>01</sub> S<sub>00</sub>

```
// -----
// 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 05 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 05 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 05 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

a<sub>3</sub> a<sub>2</sub> a<sub>1</sub> a<sub>0</sub> ^ ← "xor"

b<sub>3</sub> b<sub>2</sub> b<sub>1</sub> b<sub>0</sub>

---

s<sub>3</sub> s<sub>2</sub> s<sub>1</sub> s<sub>0</sub>

s = f ( s<sub>3</sub>, s<sub>2</sub>, s<sub>1</sub>, s<sub>0</sub> ); // 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 05 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.