

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

ARQ1 _ Aula_07

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=_Wta-lt79RU
<https://www.youtube.com/watch?v=o8aHEaAsLw8>
<https://www.youtube.com/watch?v=bh1c5pv56lY>

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

Para os exercícios a seguir, considerar o exemplo abaixo em Verilog.

```
// -----
// Exemplo_0701 - GATES
// Nome: xxx yyy zzz
// Matricula: 999999
// -----

// -----
// f7_gate
// -----
module f7 ( output s,
            input a,
            input b );

// descrever por portas

endmodule // f7


// -----
// multiplexer
// -----
module mux ( output s,
            input a,
            input b,
            input select );

// definir dados locais
wire not_select;
wire sa;
wire sb;

// descrever por portas
not NOT1 ( not_select, select );

and AND1 ( sa, a, not_select );
and AND2 ( sb, b, select );

or OR1 ( s, sa, sb );
endmodule // mux


module test_f7;
// ----- definir dados
reg x;
reg y;
reg s;
wire w;
wire z;

f7 modulo ( w, x, y );

mux MUX1 ( z, x, y, s );
```

```

// ----- parte principal

initial
begin : main
    $display("Exemplo_0701 - xxx yyy zzz - 999999");
    $display("Test LU's module");
    $display(" x   y   s   z");

    x = 1'b0; y = 1'b1; s = 1'b0;

    // projetar testes do modulo
#1    $monitor("%4b %4b %4b %4b", x, y, s, z);
#1    s = 1'b1;

end

endmodule // test_f7

```

Exercícios

- 01.) Projetar e descrever em Verilog, usando portas nativas, uma unidade lógica (LU) com operações AND e OR, com duas saídas simultâneas (paralelas, 2 respostas), para variáveis de entrada 01 com bit cada. O nome do arquivo deverá ser Exemplo_0701.v, e poderá seguir o modelo descrito abaixo. Incluir previsão de testes. Simular o módulo no Logisim e apresentar *layout* do circuito e subcircuitos.
- 02.) Projetar e descrever em Verilog, usando **portas nativas**, uma unidade lógica (LU) com operações NAND e NOR, com uma saída selecionável (1 resposta), para variáveis de entrada com 01 bit cada. O nome do arquivo deverá ser Exemplo_0702.v. Incluir previsão de testes. Simular o módulo no Logisim e apresentar *layout* do circuito e subcircuitos. DICA: Usar para o sinal extra para a seleção (0-OR;1-AND).
- 03.) Projetar e descrever em Verilog, usando **portas nativas**, uma unidade lógica (LU) com o acréscimo das operações AND e OR, com uma saída só para ambas, para variáveis de entrada com 01 bit cada, além de NAND e NOR na mesma situação. Os resultados de cada grupo serão selecionados por uma primeira chave (2x1); para selecionar uma porta em cada grupo, e outra chave (2x1) que selecionará entre o grupo (AND, OR) ou o grupo (NAND, NOR). O nome do arquivo deverá ser Exemplo_0703.v. Incluir previsão de testes. Simular o módulo no Logisim e apresentar *layout* do circuito e subcircuitos. DICA: Usar para um sinal para a seleção (0-portas AND/NAND;1-OR/NOR). Usar para outro sinal para a seleção (0-grupo AND/OR ;1-NAND/NOR).
- 04.) Projetar e descrever em Verilog, usando **portas nativas**, uma unidade lógica (LU) com o acréscimo das operações XOR e XNOR, com uma saída só para ambas, para variáveis de entrada com 01 bit cada, além de OR E NOR; na mesma situação. Os resultados de cada grupo serão selecionáveis entre o grupo (XOR, XNOR) ou o grupo (OR, NOR), seleção (4x1). O nome do arquivo deverá ser Exemplo_0704.v. Incluir previsão de testes. Simular o módulo no Logisim e apresentar *layout* do circuito e subcircuitos. DICA: Usar para o sinal extra de 2 bits para a seleção (00-OR; 01-NOR;10-XOR; 11-XNOR).
- 05.) Projetar e descrever em Verilog, usando **portas nativas**, uma unidade lógica (LU) com o acréscimo das operações NOT, OR, NOR, AND, NAND, XOR, XNOR, simultâneas, com apenas 1 saída, selecionável (7x1). O nome do arquivo deverá ser Exemplo_0705.v. Incluir previsão de testes. Simular o módulo no Logisim e apresentar *layout* do circuito e subcircuitos. DICA: Usar um sinal extra de 3 bits para a seleção.

Extras

- 06.) Projetar e descrever em Verilog, usando portas nativas, uma unidade lógica (LU) com um comparador de bits usando portas XOR e XNOR para calcular a igualdade ou desigualdade, para variáveis com 2 bits cada, selecionável (0-igual; 1-diferente).
O nome do arquivo deverá ser Exemplo_0706.v.
Incluir previsão de testes.
Simular o módulo no Logisim e apresentar *layout* do circuito e subcircuitos.
DICA: Montar a tabela-verdade e identificar os mintermos.
- 07.) Projetar e descrever em Verilog, usando portas nativas, uma unidade lógica (LU) com um comparador de bits usando apenas portas básicas (NOT, AND, OR) para calcular a magnitude (se maior ou menor), para variáveis com 2 bits cada, selecionável (0-menor; 1-maior).
O nome do arquivo deverá ser Exemplo_0707.v.
Incluir previsão de testes.
Simular o módulo no Logisim e apresentar *layout* do circuito e subcircuitos.
DICA: Montar a tabela-verdade e identificar os mintermos.