

ARQ1 _ Aula_05

Tema: Introdução à linguagem Verilog e simulação em Logisim

Preparação

Vídeos recomendados

<https://www.youtube.com/watch?v=TdDCWG2inoY>

<https://www.youtube.com/watch?v=WExVhr583vA>

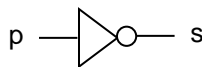
<https://www.youtube.com/watch?v=ZgAtsWlyd5I>

Atividade: Projeto de circuitos combinatórios

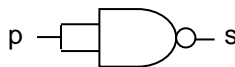
Universalidade das portas NAND e NOR

As portas NAND e NOR podem ser usadas para substituir outras funções lógicas básicas por composições equivalentes, como mostrado a seguir.

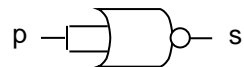
Porta NÃO (NOT)



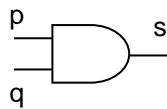
Porta NÃO (NOT)



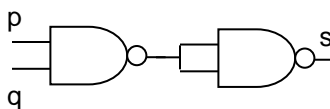
Porta NÃO (NOT)



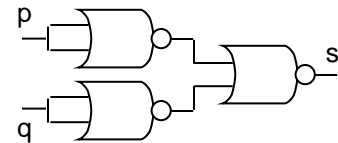
Porta AND



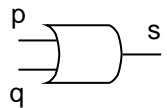
Porta AND



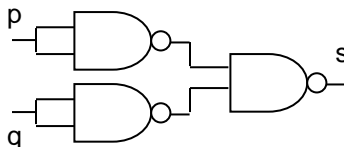
Porta AND



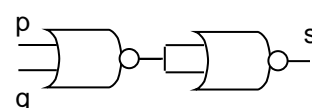
Porta OR



Porta OR



Porta OR



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

```
// -----  
// Exemplo_0501 - GATES  
// Nome: xxx yyy zzz  
// Matricula: 999999  
// -----
```

```
// -----  
// f5_gate  
// m a b s  
// 0 0 0 0  
// 1 0 1 1 <- a'.b  
// 2 1 0 0  
// 3 1 1 0  
//  
// -----  
module f5 ( output s,  
            input a,  
            input b );  
// definir dado local  
wire not_a;  
// descrever por portas  
not NOT1 ( not_a, a );  
and AND1 ( s, not_a, b );  
endmodule // f5
```

```
// -----  
// f5_gate  
// m a b s  
// 0 0 0 0  
// 1 0 1 1 <- a'.b  
// 2 1 0 0  
// 3 1 1 0  
//  
// -----  
module f5b ( output s,  
            input a,  
            input b );  
// descrever por expressao  
assign s = ~a & b;  
endmodule // f5b
```

```

module test_f5;
// ----- definir dados
    reg x;
    reg y;
    wire a, b;

    f5a moduloA ( a, x, y );
    f5b moduloB ( b, x, y );

// ----- parte principal

    initial
    begin : main
        $display("Exemplo_0501 - xxx yyy zzz - 999999");
        $display("Test module");
        $display(" x y a b");

        // projetar testes do modulo
        $monitor("%4b %4b %4b %4b", x, y, a, b);
        x = 1'b0; y = 1'b0;
    #1    x = 1'b0; y = 1'b1;
    #1    x = 1'b1; y = 1'b0;
    #1    x = 1'b1; y = 1'b1;

    end

endmodule // test_f5

```

Exercícios:

Orientação geral:

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

Exemplos em Verilog fornecidos anteriormente servirão como ponto de partida.

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

- 01.) Projetar e descrever em Verilog, usando apenas portas nativas **nand** um módulo equivalente à disjunção ($a \mid b$).
O nome do arquivo deverá ser Exemplo_0502.v,
e poderá seguir o modelo descrito anteriormente.
Incluir previsão de testes.
Simular o módulo no Logisim e
apresentar *layout* do circuito e subcircuitos.
- 02.) Projetar e descrever em Verilog, usando apenas portas nativas **nor** um módulo equivalente à conjunção ($a \& b$).
O nome do arquivo deverá ser Exemplo_0503.v,
e poderá seguir o modelo descrito anteriormente.
Incluir previsão de testes.
Simular o módulo no Logisim e
apresentar *layout* do circuito e subcircuitos.
- 03.) Projetar e descrever em Verilog, usando apenas portas nativas **nand** módulo equivalente à conjunção ($a \& b$).
O nome do arquivo deverá ser Exemplo_0504.v,
e poderá seguir o modelo descrito anteriormente.
Incluir previsão de testes.
Simular o módulo no Logisim e
apresentar *layout* do circuito e subcircuitos.
- 04.) Projetar e descrever em Verilog, usando apenas portas nativas **nor** módulo equivalente à disjunção ($a \mid b$).
O nome do arquivo deverá ser Exemplo_0505.v,
e poderá seguir o modelo descrito anteriormente.
Incluir previsão de testes.
Simular os módulos no Logisim e
apresentar *layout* dos circuitos e subcircuitos.

- 05.) Projetar e descrever em Verilog, usando apenas portas nativas **nand** um módulo equivalente à disjunção exclusiva ($a \oplus b = a \text{ xor } b$).
O nome do arquivo deverá ser Exemplo_0506.v,
e poderá seguir o modelo descrito anteriormente.
Incluir previsão de testes.
Simular o módulo no Logisim e
apresentar *layout* do circuito e subcircuitos.
- 06.) Projetar e descrever em Verilog, usando apenas portas nativas **nor** um módulo equivalente à negação da disjunção exclusiva ($\sim(a \oplus b) = a \text{ xnor } b$).
O nome do arquivo deverá ser Exemplo_0507.v,
e poderá seguir o modelo descrito anteriormente.
Incluir previsão de testes.
Simular o módulo no Logisim e
apresentar *layout* do circuito e subcircuitos.

Extras

- 07.) Projetar e descrever em Verilog, usando apenas portas nativas **nand** um módulo equivalente à expressão ($\sim a \mid b$).
O nome do arquivo deverá ser Exemplo_0508.v,
e poderá seguir o modelo descrito anteriormente.
Incluir previsão de testes.
Simular o módulo no Logisim e
apresentar *layout* do circuito e subcircuitos.
- 08.) Projetar e descrever em Verilog, usando apenas portas nativas **nor** um módulo equivalente à expressão ($\sim a \mid \sim b$).
O nome do arquivo deverá ser Exemplo_0509.v,
e poderá seguir o modelo descrito anteriormente.
Incluir previsão de testes.
Simular o módulo no Logisim e
apresentar *layout* do circuito e subcircuitos.

Extras

- 06.) Projetar e descrever em Verilog, usando apenas portas nativas **nor** um módulo equivalente à função do exemplo (a . b').
O nome do arquivo deverá ser Exemplo_0508.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 a propriedade da álgebra proposta por DeMorgan.
- 07.) Projetar e descrever em Verilog, usando apenas portas nativas **nand** um módulo equivalente à função do exemplo (a . b').
O nome do arquivo deverá ser Exemplo_0509.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 a negação de uma **nand**.