

Lógica Digital (1001351)

Introdução à Verilog

Prof. Edilson Kato

kato@ufscar.br

Prof. Maurício Figueiredo

mauricio@ufscar.br

Prof. Ricardo Menotti

menotti@ufscar.br

Prof. Roberto Inoue

rsinoue@ufscar.br

Departamento de Computação
Universidade Federal de São Carlos

Atualizado em: 6 de março de 2019



Introdução à Verilog

- ▶ Verilog é uma linguagem complexa, mas neste curso não vamos abordar todas as suas potencialidades;
- ▶ O que vamos aprender será suficiente para projetar e testar uma grande variedade de circuitos;
- ▶ Iremos abordar as funcionalidades da linguagem a medida que avançarmos com os circuitos digitais;
- ▶ A principal habilidade desejada neste curso é a capacidade de traduzir com facilidade um circuito para Verilog e vice-versa;
- ▶ Isso só pode ser alcançado com a prática, pois assim como na programação, estudar problemas resolvidos não ajuda muito.

Introdução à Verilog

- ▶ Em Verilog há várias maneiras de se descrever um mesmo circuito, por exemplo, a partir:
 - ▶ **Funcional ou Lógica:** de funções ou portas básicas;
 - ▶ **Estrutural:** de uma hierarquia de componentes;
 - ▶ **Comportamental:** da descrição de seu comportamento;
- ▶ Pode-se usar combinações das metodologias.

Exemplo: multiplexador

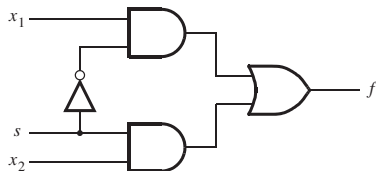


Figure 2.36 The logic circuit for a multiplexer.

```
1 module example2(input x1, x2, s,
2                 output f);
3     assign f =
4         (x1 & ~s) | (x2 & s);
5 endmodule
```

```
1 // Behavioral specification
2 module example3(input x1, x2, s,
3                 output reg f);
4     always @(x1, x2, s)
5         if (s==0)
6             f = x1;
7         else
8             f = x2;
9 endmodule
```

```
1 module example1(x1, x2, s, f);
2     input x1, x2, s;
3     output f;
4
5     not (k, s);
6     and (g, k, x1);
7     and (h, s, x2);
8     or (f, g, h);
9 endmodule
```

Outro exemplo

```
1 module example2 (x1, x2, x3, x4,  
2                 f, g, h);  
3   input x1, x2, x3, x4;  
4   output f, g, h;  
5  
6   and (z1, x1, x3);  
7   and (z2, x2, x4);  
8   or (g, z1, z2);  
9   or (z3, x1, ~x3);  
10  or (z4, ~x2, x4);  
11  and (h, z3, z4);  
12  or (f, g, h);  
13 endmodule
```

```
1 module example2 (x1, x2, x3, x4,  
2                 f, g, h);  
3   input x1, x2, x3, x4;  
4   output f, g, h;  
5  
6   assign g = (x1 & x3) | (x2 & x4);  
7   assign h = (x1 | ~x3) & (~x2 | x4);  
8   assign f = g | h;  
9 endmodule
```

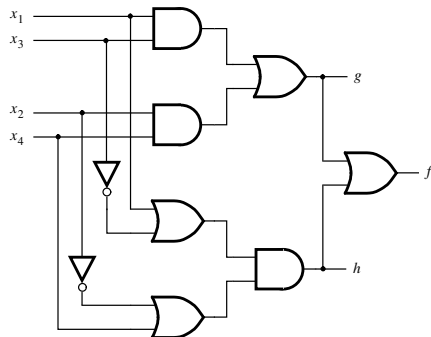


Figure 2.39 Logic circuit for the code in Figure 2.38.

Introdução à Verilog

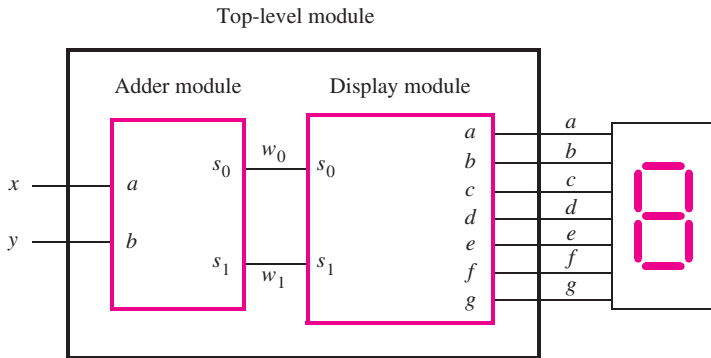


Figure 2.44 A logic circuit with two modules.

Introdução à Verilog

```
1 // Top-level module
2 module adder_display(x, y, a, b,
3                     c, d, e, f, g);
4     input x, y;
5     output a, b, c, d, e, f, g;
6     wire w1, w0;
7
8     adder U1 (x, y, w1, w0);
9     display U2 (w1, w0, a, b, c,
10                d, e, f, g);
11 endmodule
```

```
1 // An adder module
2 module adder (a, b, s1, s0);
3     input a, b;
4     output s1, s0;
5
6     assign s1 = a & b;
7     assign s0 = a ^ b;
8 endmodule
```

```
1 // A module for driving a
2 //       7-segment display
3 module display(s1, s0, a, b,
4               c, d, e, f, g);
5     input s1, s0;
6     output a, b, c, d, e, f, g;
7
8     assign a = ~s0;
9     assign b = 1;
10    assign c = ~s1;
11    assign d = ~s0;
12    assign e = ~s0;
13    assign f = ~s1 & ~s0;
14    assign g = s1 & ~s0;
15 endmodule
```

Como NÃO escrever Verilog

- ▶ NÃO escrever código que se assemelhe a um programa de computador, contendo muitas variáveis e loops;
 - ▶ É difícil determinar qual circuito lógico as ferramentas CAD produzirão ao sintetizar código assim;
- ▶ Neste curso veremos exemplos completos de código Verilog que representam uma ampla gama de circuitos lógicos;
 - ▶ Neles o código é facilmente relacionado ao circuito lógico descrito;
 - ▶ Procure adotar o mesmo estilo de código;
- ▶ *Se não for possível determinar prontamente qual circuito lógico é descrito pelo código Verilog, então as ferramentas CAD provavelmente não sintetizarão o circuito que o projetista está tentando modelar;*
- ▶ **Analise o circuito resultante produzido pelas ferramentas de síntese;**

Bibliografia

- ▶ Brown, S. & Vranesic, Z. - Fundamentals of Digital Logic with Verilog Design, 3rd Ed., Mc Graw Hill, 2009
- ▶ <http://www.asic-world.com/verilog/>
- ▶ <https://www.edaplayground.com/>

Lógica Digital (1001351)

Introdução à Verilog

Prof. Edilson Kato

kato@ufscar.br

Prof. Maurício Figueiredo

mauricio@ufscar.br

Prof. Ricardo Menotti

menotti@ufscar.br

Prof. Roberto Inoue

rsinoue@ufscar.br

Departamento de Computação
Universidade Federal de São Carlos

Atualizado em: 6 de março de 2019

