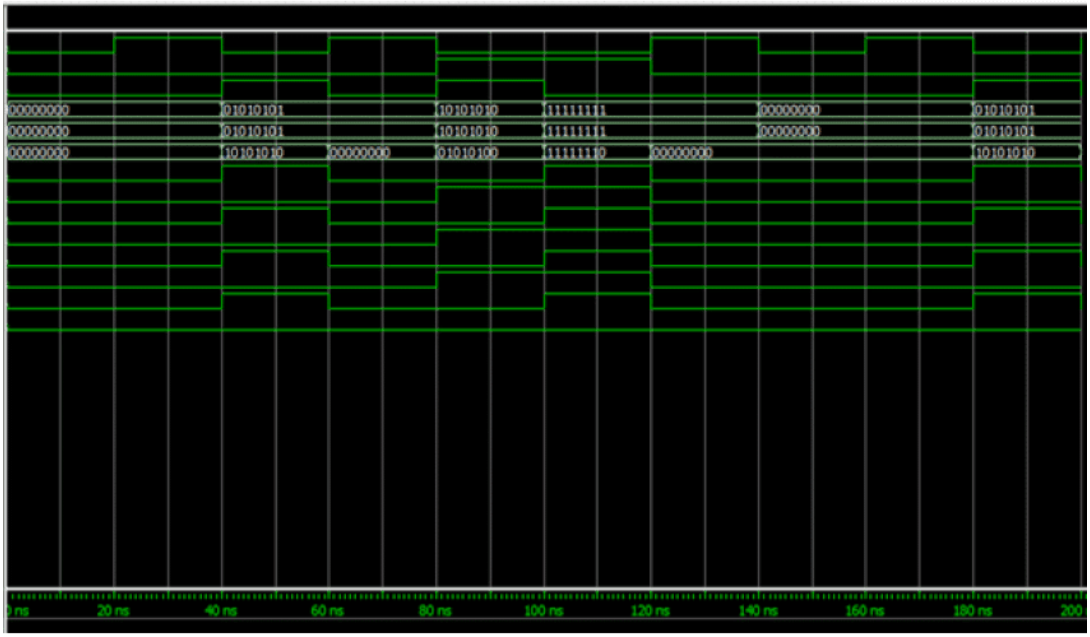
Tutorial de Verilog - Operadores

Por **Thiago Lima** - 16/04/2015



ÍNDICE DE CONTEÚDO [MOSTRAR]

Este post faz parte da série Tutorial de Verilog. Leia também os outros posts da série:

- Tutorial de Verilog - Operadores
- Tutorial de Verilog - Operadores Lógicos e Aritméticos Shift Right e Shift Left
- Tutorial de Verilog - Ponto Fixo e Ponto Flutuante em Verilog - Parte I
- Tutorial de Verilog - O primeiro Projeto com Quartus
- Tutorial de Verilog - 7 formas de representar um MUX em Verilog
- Tutorial de Verilog: Decodificador ou DEMUX
- Tutorial de Verilog: Meio Somador (Half Adder)
- Tutorial de Verilog: Somador Completo (full adder)
- Tutorial de Verilog: Conversor de Código Binário para Código Gray
- Tutorial de Verilog: Conversor de Código Gray para Código Binário
- Tutorial de Verilog: Somador com Propagação do Carry - Somador Ripple-Carry
- Tutorial de Verilog: Conversor BCD para 7 Segmentos
- Tutorial de Verilog: Contador binário síncrono crescente Mod 10 com reset síncrono em nível lógico baixo

Esse é o primeiro de uma série de artigos que visa trazer diversas informações de como desenvolver um bom código em Verilog.

Estados do Bit

Primeiramente vou apresentar todos os estados que os bits podem ter em um design com Verilog. Esse fato é muito importante pois impacta no desenvolvimento do projeto e sempre ter isso em mente ajuda a não cometermos gafes simples. Em HDL, todo bit pode ter os seguintes estados:

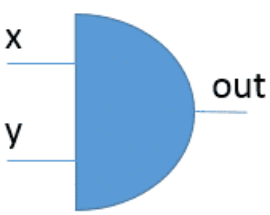
Valor	Significado
1	Um Lógico
0	Zero Lógico
Z	Alta Impedância, flutuando, não conectado.
X	Valor lógico desconhecido ou indefinido - todo estado X deve estar em 0, 1, Z ou em transição entre esses estados. Não podemos setar um bit para o estado X.

Operadores em Verilog

Vamos criar um módulo bem simples de hardware descrevendo seu comportamento em Verilog. Conforme foi discutido no post de Rodrigo Pereira sobre a implementação de um processador em Verilog, o modelo comportamental descreve exatamente como um circuito deve funcionar, utilizando uma linguagem HDL, e deixa a responsabilidade da criação da lógica e da simulação correta para as ferramentas de síntese.

Conforme André Prado cita em seu artigo sobre sistemas digitais, no nível comportamental o sistema é descrito em função do seu comportamento, é uma representação funcional do mesmo. Estamos, nesse caso, preocupados em descrever o que acontece na saída do sistema quando há uma alteração em uma entrada, é a relação direta de entrada/saída.

Uma porta AND, ilustrada na figura 1, pode ser reproduzida simplesmente como:



```
// Exemplo de Modelo Comportamental em Verilog de uma porta AND
// Rodrigo Pereira - 16/03/2015

module EXEMPLO_AND ( x, y, out); // declaração do nome do módulo AND
input wire x, y; // entrada de dados - um bit
output wire out; // saída de dados - um bit

assign out = x & y; // circuito AND utilizando a chamada assign em Verilog

endmodule // final do módulo AND
```

Abaixo demonstro um código para um circuito com 3 portas lógicas descrito na figura 2.

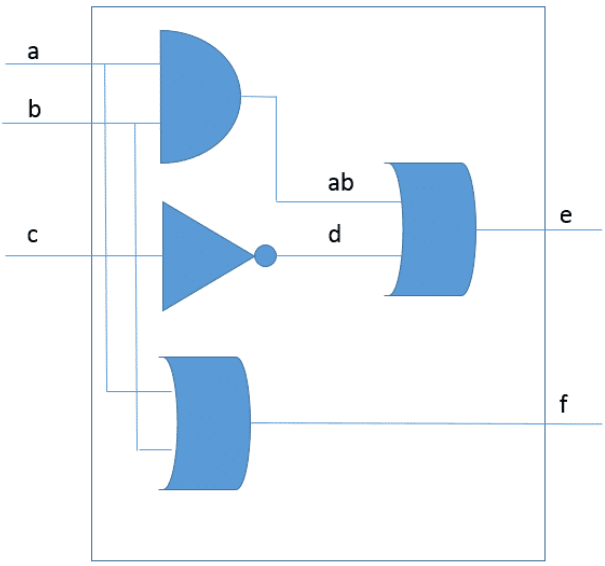


Figura 2: Circuito combinacional proposto para o segundo exemplo

```
// Exemplo de Modelo Comportamental em Verilog de um circuito com 3 portas logicas
// Thiago Lima - 16/04/2015

module EXAMPLE_2 (
                  a,
                  b,
                  c,
                  e,
                  f
);

input
```

```
16
17 wire
18       ab, d;
19
20 assign ab = a & b;
21 assign d = ~c;
22 assign e = ab | d;
23 assign f = a | b;
24 endmodule
```

Note algumas características interessantes do módulo escrito em Verilog:

- A palavra **module** inicia o módulo;
- Depois dos parenteses, há um ponto-e-virgula;
- Entradas são identificadas com a palavra **input** e as saídas com a palavra **output**;
- Os fios intermediários são identificados com a palavra **wire**;
- O modulo é finalizado por **endmodule**.

Todas as operações são realizadas em paralelo. Veja que para o circuito combinacional mostrado anteriormente, o caminho mais rápido é relativo a operação:

```
1 assign f = a | b;
```

Confira abaixo todos os operadores que podem ser utilizados com Verilog.

Operadores unitários

+	**a = b + c ;**	**// Soma , adição binaria**
-	**a = b - c ;**	**// Subtração, subtração binaria**

Operadores de todos os bits

&	a = &b ;	// AND para todos os bits de b
^	a = ^b ;	// XOR para todos os bits de b
~|	a = ~|b ;	// NOR para todos os bits de b
~&	a = ~&b ;	// NAND para todos os bits de b
~^	a = ~^b ;	// XNOR para todos os bits de b

Operadores de bit-wise

~	a = ~b;	// inverte b
|	a = b | c;	// OR bit a bit
&	a = b & c;	// AND bit a bit
^	a = b ^ c;	// XOR bit a bit
~|	a = b ~| c;	// NOR bit a bit
~&	a = b ~& c;	// NAND bit a bit
~^	a = b ~^ c;	// XNOR bit a bit

Operadores de igualdade

=	a = b	a recebe o valor de b
==	a == b	O valor de a é igual ao valor de b?
===	a === b	O valor de a é igual ao valor de b? Inclui os estados Z e X.
!=	a != b	O valor de a nao é igual ao valor de b?
!==	a !== b	O valor de a nao é igual ao valor de b? Inclui os estados Z e X.

Saiba mais sobre Verilog

PROCESSADORES PROGRAMÁVEIS - como projetar um processador em VERILOG -

PROCESSADORES PROGRAMÁVEIS - como projetar um processador em VERILOG - Organização - parte 2

PROCESSADORES PROGRAMÁVEIS - como projetar um processador em VERILOG - Codificação - parte 3

Formas de representar um sistema digital

Tutorial de Modelsim: Verificando o VHDL antes de programar o FPGA

Tutorial de Verilog - Operadores Lógicos e Aritméticos Shift Right e Shift Left

Outros artigos da série

Tutorial de Verilog - Operadores Lógicos e Aritméticos Shift Right e Shift Left >>

Este post faz da série Tutorial de Verilog. Leia também os outros posts da série:

- Tutorial de Verilog - Operadores
- Tutorial de Verilog - Operadores Lógicos e Aritméticos Shift Right e Shift Left
- Tutorial de Verilog - Ponto Fixo e Ponto Flutuante em Verilog - Parte I
- Tutorial de Verilog - O primeiro Projeto com Quartus
- Tutorial de Verilog - 7 formas de representar um MUX em Verilog

Thiago Lima

Apaixonado por sistemas digitais e circuitos eletrônicos, ja contabilizo 16 anos trabalhando com desenvolvimento de produtos eletrônicos. Formado na USP Sao Carlos, com mestrado em Engenharia Elétrica no Rochester Institute of Technology pelo CsF, atualmente lidero boa parte das operações do Embarcados, buscando levar conhecimento de sistemas eletrônicos para o Brasil. Experimentar o mundo das startups nos EUA foi transformador. La fui cofundador de uma startup de tecnologia chamada Una, sendo acelerado e incubado por um programa especial de Startups no RIT. Ao final, recebemos um prêmio de melhor startup do programa. No Laboratório Hacker de Campinas sou um dos entusiastas de novas tecnologias e apoio iniciativas da comunidade. Tambem participo de atividades comunitarias e sou um dos responsáveis pela Plataforma Ituiutaba Lixo Zero, onde escrevo regularmente artigos sobre redução de resíduos. Sou sonhador mesmo e quero acender a luz ?