



Curso de Verilog



CdH

Apresentação



Caio Rodrigo

E-mail: caioxyz@hotmail.com

Telegram: @zelda64

Jorge Reis

E-mail: jlcr1997@gmail.com

Telegram: @jlreis

Clube do Hardware UFC

E-mail: hardwareufc@gmail.com

Facebook : <https://www.facebook.com/clubedohardwareufc>

Instagram: @_cdhard





Estrutura

- N° aulas: 5
- Carga Horária: 20 H
- Aulas Teórico-práticas
- Bibliografia:
 - Verilog HDL: A guide to Digital Design and Synthesis- Samir Palnitkar



Ementa

- Módulo
- Tipos de Dados
- Representação de números
- Níveis de Abstração: Gate Level
- Regras para conexão de portas
- Testbench
- Conceito de tempo em verilog
- Análise de Formas de Onda
- Níveis de Abstração: Dataflow
- Níveis de Abstração: Behavioral
 - Always x Initial
 - Estruturas de controle de fluxo
 - if-else
 - for
 - case
 - Parameters
 - Lógica Combinacional x Sequencial
 - Blocking x Non-blocking statements
 - Tasks e Functions
- Máquinas de Estado
- FPGAS

Aula 1



Conteúdo:

Módulo

Tipos de Dados

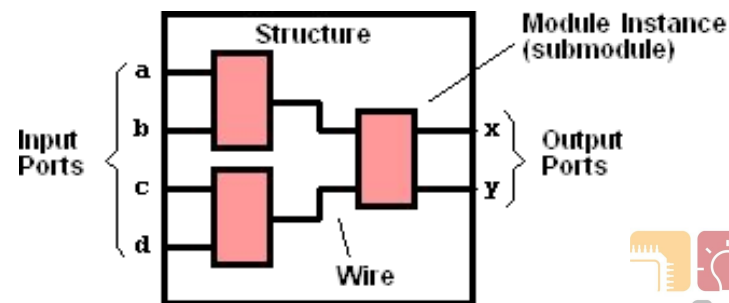
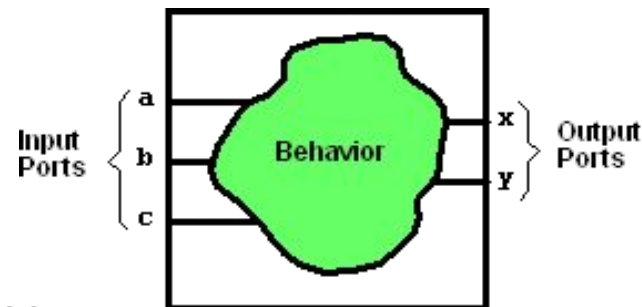
Representação de números

Níveis de Abstração: Gate Level

Módulo

Bloco básico da linguagem

Pode conter instâncias de outros módulos

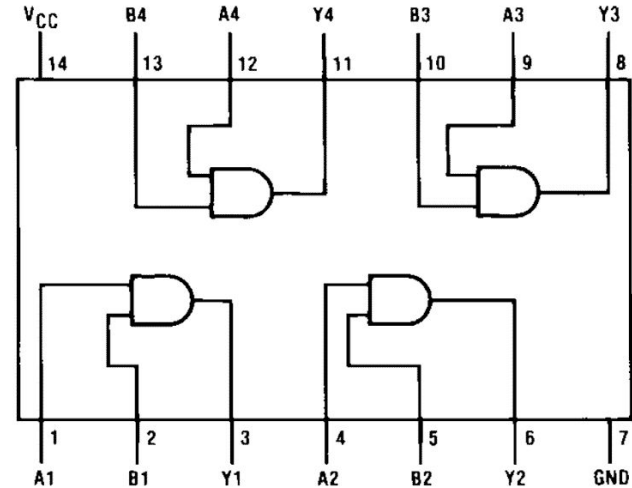


Exemplo:

- module and (
input in1,in2,
output out
);

and a1 (out,in1,in2);

endmodule





Estrutura de um Módulo

`module <nome>(<portas>);`

Inputs

`<declaração>`

Outputs

`<corpo do módulo>`

Inouts

`endmodule`

Estrutura de um Módulo

module <nome>(<portas>);

Regs

<declaração>

Wires

<corpo do módulo>

Memórias

endmodule

Estrutura de um Módulo

<code>module <nome>(<portas>;</code>	Initial
<code><declaração></code>	Always
<code><corpo do módulo></code>	Assignments contínuos
<code>endmodule</code>	Instância de módulos

System Tasks

- Verilog disponibiliza algumas tasks para fazer algumas operações de rotina
- Formato de uma task:
`$<keyword>`
- Exemplos de tasks:
 - `$monitor`
 - `$display`
 - `$finish`

Exemplo de uso:

```
bus[3:0]
```

```
$display("valor de bus: %b ", bus)
```

Prática 0 - Hello World



Representação dos números

- Sized
- Unsized

Representação dos números

- Sized
- Unsized
- Representados na forma:
 - `<size>'<base format><number>`
- Exemplos:
 - Binário: `<nº de bits>'b<número>`: `2'b11`, `3'b010`
 - Hexadecimal: `<nº de bits>'h<número>`: `2'h2`, `4'hA`, `8'hA`
 - Decimal: `<nº de bits>'d<número>`: `2'd2`, `4'd7`;





Representação dos números

- Sized
- **Unsigned**
- Números sem <size> possuem um número de bits padrão específico da máquina (pelo menos 32 bits).
- Números sem <base_format> são considerados decimais por padrão.
- Exemplo:
 - 23456
 - 'hc3
 - 'o21

Conjunto de valores

Valor	Estado em um Circuito
0	Zero lógico, Condição Falsa
1	Um lógico, Condição Verdadeira
x	Valor desconhecido
z	Alta impedância, flutuando



Tipos de Dados

- reg
- wire
- vectors
- integer
- real
- time
- arrays
- parameters



REG X WIRE

Reg representa elementos que armazenam valores

Valor padrão de um reg é x

Outputs podem ser do tipo reg

Exemplo de declaração:

- reg reset;
- output reg x;

Wire não armazena valores

Representa conexões entre elementos de Hardware

Inputs e Inouts são wires, Outputs podem ser do tipo wire

Exemplo de declaração:

- wire a;
- input z;
- output x;





Vectors

- Regs e wires podem ser declarados como vectors
- Se não forem, o padrão é 1 bit
- Exemplo:
 - `wire a;`
 - `wire [7:0] bus;`
 - `reg [0:40] virtual_addr`



Integer, real e time

- integer
 - registrador de propósito geral
 - geralmente utilizado para contagem
 - o tamanho padrão é de 32 bits
 - guardam dados como signed ao contrário dos regs que guardam como unsigned por padrão
- real
 - podem ser declarados em notação decimal ou científica
 - não são muito usados
- time
 - utilizado para armazenar tempo de simulação (Será explicado na aula 2)



Arrays

- Utilizados para reg, integer, time e vector
- Não são permitidos para variáveis do tipo real
- exemplo:
 - integer count[7:0]
 - reg [4:0] port_id [7:0]
- Arrays são múltiplos elementos com 1 ou n bits, já vector é um elemento com n bits



Local Parameter

- Uma das formas de criar constantes dentro de um módulo em Verilog.
- Exemplo de uso:
 - `localparameter port_id= 5;`

Prática 1 - Trabalhando com Números



Níveis de abstração

- Gate Level
- Dataflow
- Behavioral



Níveis de abstração

- Gate Level
 - Dataflow
 - Behavioral
- Descrição é de baixo nível
 - Usa instâncias de módulos simples (Portas lógicas)



Portas:

- and
- or
- xor
- nand
- nor
- xnor
- not
- buf

Exemplos:

and a1 (out,in1,in2)

or or1 (out,in1,in2,in3)

not (out, in)

Prática 2 - Gate Modelling
