

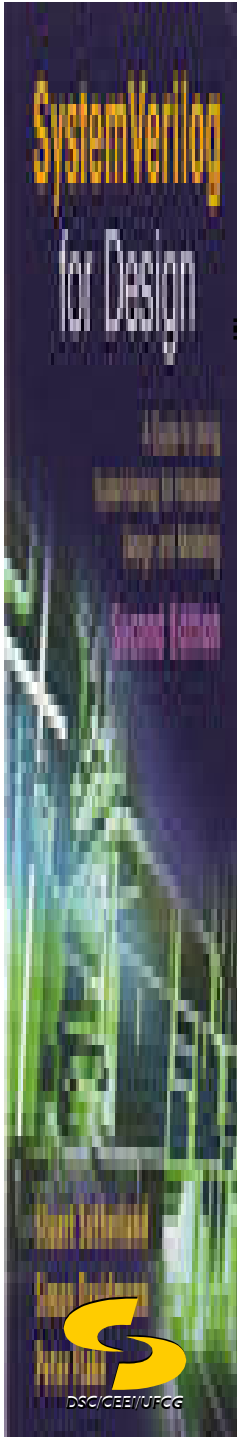
Universidade Federal de Campina Grande
Departamento de Sistemas e Computação
Curso de Bacharelado em Ciência da Computação

Organização e Arquitetura de Computadores I

Linguagem de Descrição de Hardware (Parte II)

Prof^a Joseana Macêdo Fachine Régis de Araújo
joseana@computacao.ufcg.edu.br

Carga Horária: 60 horas



Tópicos

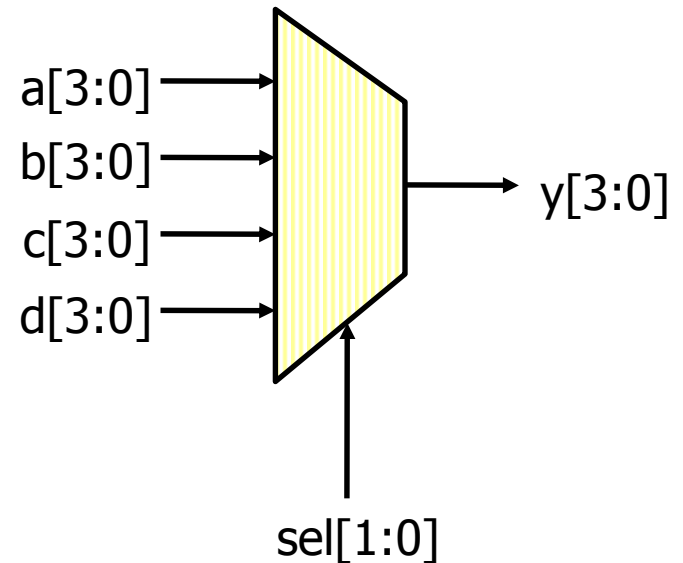
- ***Hardware Description Language (HDL)***
 - Conceitos Básicos – continuação
(Lógica Combinacional)

Síntese de Multiplexadores

□ Instrução case

```
module mux_4bits (  
    input logic [3:0] a, b, c, d,  
    input logic [1:0] sel,  
    output logic [3:0] y);
```

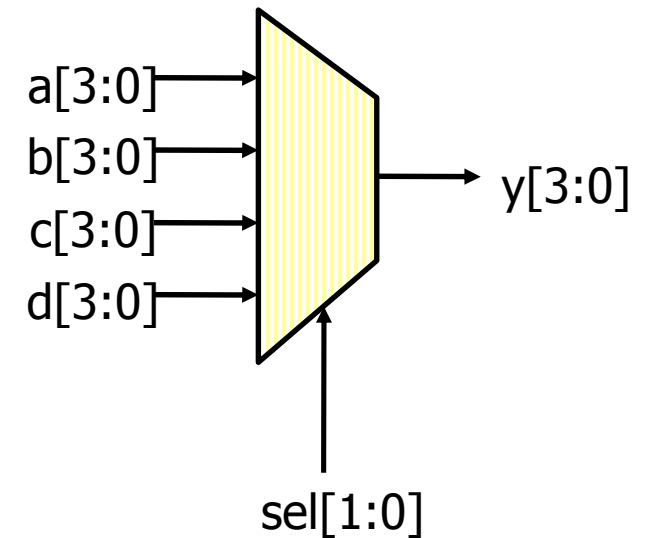
```
    always_comb  
        unique case (sel)  
            0: y <= a;  
            1: y <= b;  
            2: y <= c;  
            default: y <= d;  
        endcase  
endmodule
```



Síntese de Multiplexadores

□ Instrução `if .. else`

```
module mux_4bits (  
    input logic [3:0] a, b, c, d,  
    input logic [1:0] sel,  
    output logic [3:0] y);  
  
    always_comb  
    begin  
        if (sel == 0) y <= a; else  
        if (sel == 1) y <= b; else  
        if (sel == 2) y <= c; else  
        if (sel == 3) y <= d;  
    end  
  
endmodule
```



Obs.: instruções *case* e *if/else* são estilos mais recomendados para multiplexadores “grandes”.

Prioridade Lógica

- ❑ Quando os casos de uma sequência de condições (**if**) de um **case** não são mutuamente exclusivos, a ferramenta de síntese cria uma estrutura de prioridade.

- ❑ Exemplo:

```
module mux_4pri (  
    input logic a, b, c, d, sel_a, sel_b, sel_c,  
    output logic y);
```

```
    always_comb
```

```
    begin
```

```
        if (sel_a == 1) y <= a; else
```

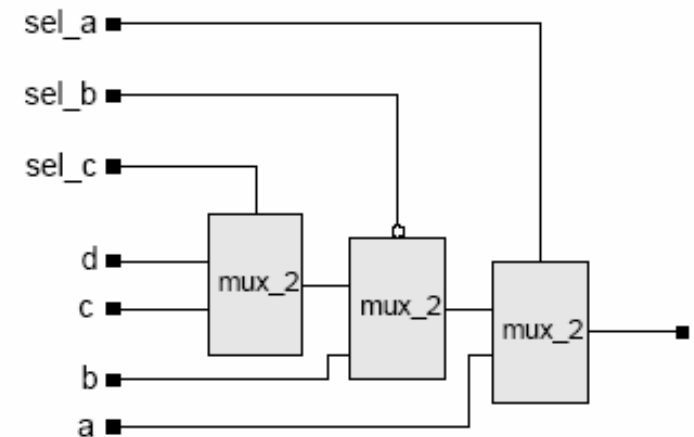
```
        if (sel_b == 0) y <= b; else
```

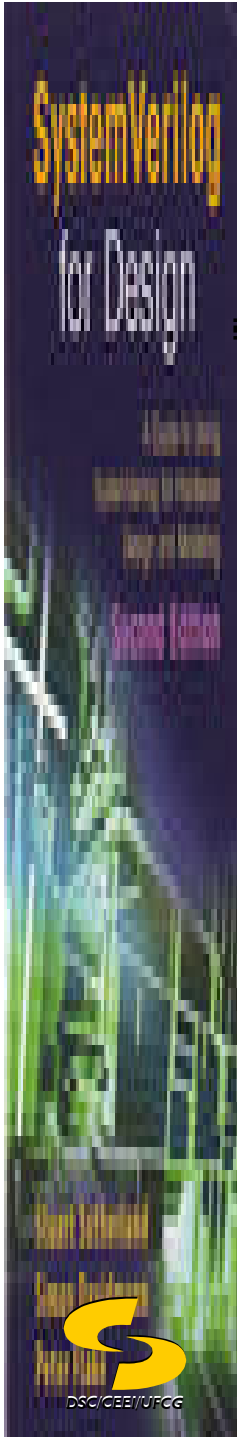
```
        if (sel_c == 1) y <= c; else
```

```
            y <= d;
```

```
    end
```

```
endmodule
```





Hardware Description Language (HDL)

- **Próximos passos:
Lógica Sequencial**