

Universidade Tecnológica Federal do Paraná – Toledo  
Engenharia da Computação – COENC

**Lógica Reconfigurável**

**Código Concorrente  
WITH-SELECT-WHEN**

**Tiago Piovesan Vendruscolo**



Esta licença permite que outros remixem, adaptem e criem a partir do trabalho para fins não comerciais, desde que atribuam o devido crédito aos autores originais. [4.0 international](https://creativecommons.org/licenses/by-nc-nd/4.0/)

# Comando WITH-SELECT-WHEN

- O comando WITH-SELECT-WHEN é utilizado quando, a partir do valor de “A” é selecionado um valor para “B”.

```
with a select  
b <="1000" when "00",  
    "0100" when "01",  
    "0010" when "10",  
    "0001" when "11";
```

- A declaração SELECT deve cobrir todos os valores de entrada.

# Comando WITH-SELECT-WHEN

- A declaração SELECT deve cobrir todos os valores de entrada.
- Pode-se utilizar “others” para incluir todas as opções não testadas explicitamente:

```
with controle select
b <="000" when 0|1,
    "100" when 2 to 5,
    "Z--" when others;    “-” don't care
```

- Utiliza-se “unaffected” para denominar valores de entrada que não afetam a saída:

```
with (a and b) select
y <="00" when "001",
    "11" when "100",
    unaffected when others;
```

# Comando WITH-SELECT-WHEN

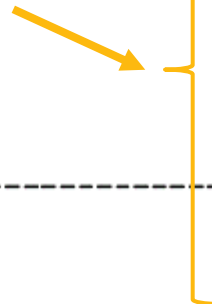
- Comparação entre as estruturas WHEN e SELECT

## Exemplo WHEN

```
1  -----
2  LIBRARY ieee;
3  USE ieee.std_logic_1164.all;
4  -----
5  ENTITY mux IS
6      GENERIC (N: INTEGER := 8);
7      PORT (x0, x1, x2, x3: IN STD_LOGIC_VECTOR(N-1 DOWNT0 0);
8          sel: IN STD_LOGIC_VECTOR(1 DOWNT0 0);
9          y: OUT STD_LOGIC_VECTOR(N-1 DOWNT0 0));
10 END ENTITY;
11 -----
12 ARCHITECTURE with_WHEN OF mux IS
13 BEGIN
14     y <= x0 WHEN sel="00" ELSE
15         x1 WHEN sel="01" ELSE
16         x2 WHEN sel="10" ELSE
17         x3;
18 END ARCHITECTURE;
19 -----
20 -----
```

## Exemplo SELECT

```
12 ARCHITECTURE with_SELECT OF mux IS
13 BEGIN
14     WITH sel SELECT
15         y <= x0 WHEN "00",
16         x1 WHEN "01",
17         x2 WHEN "10",
18         x3 WHEN OTHERS;
19 END ARCHITECTURE;
20 -----
21 -----
```



# Exercício 1

- Mux 4:1 de 8 bits usando SELECT: Faça um software que coloque na saída X um dos vetores de 8 bits “11000000”, “00110000”, “00001100” e “00000011”, dependendo dos bits na entrada SEL.

# Exercício 1

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity mux_4_1_select is
    Port ( SEL : in  STD_LOGIC_VECTOR (1 downto 0);
          X   : out STD_LOGIC_VECTOR (7 downto 0));
end mux_4_1_select;

architecture funcao of mux_4_1_select is
begin
    WITH SEL SELECT
    X <= "11000000" when "00",
        "00110000" when "01",
        "00001100" when "10",
        "00000011" when "11";
end funcao;
```

## Exercício 2

- Mux 4:1 de N bits usando SELECT: Faça um software que coloque na saída X uma das 4 entradas de N bits: A, B, C e D, dependendo dos bits na entrada SEL.

## Exercício 2

- Mux 4:1 de N bits usando SELECT: Faça um software que coloque na saída X uma das 4 entradas de N bits: A, B, C e D, dependendo dos bits na entrada SEL.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity mux_4_1_select is
    generic (N: INTEGER :=4); -- número de bits
    Port (SEL : in  STD_LOGIC_VECTOR (1 downto 0);
          A,B,C,D: in  STD_LOGIC_VECTOR (N-1 downto 0);
          X    : out STD_LOGIC_VECTOR (N-1 downto 0));
end mux_4_1_select;

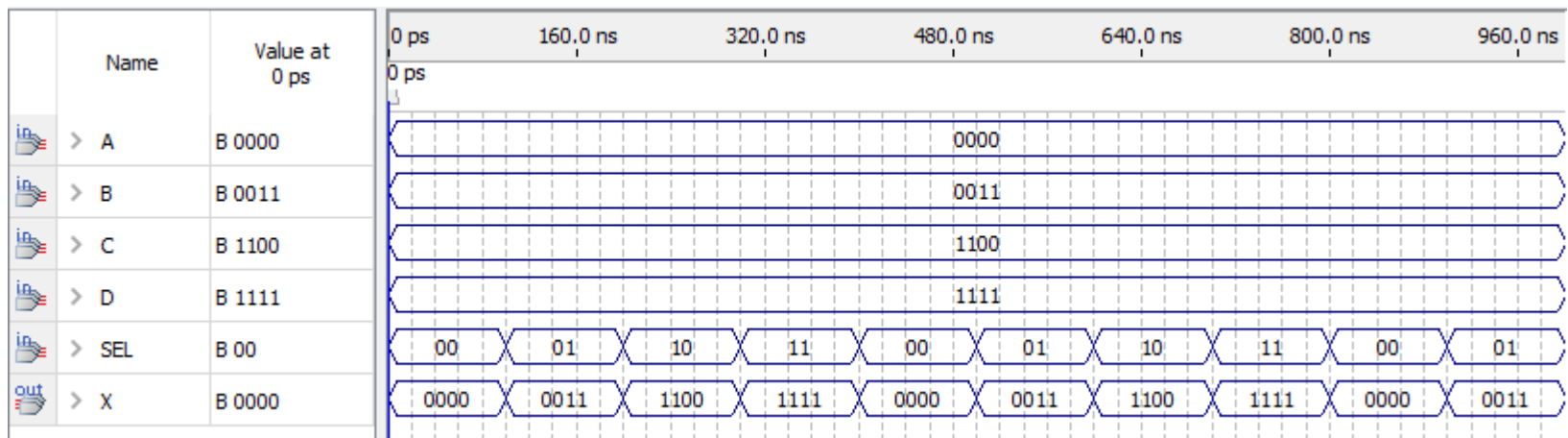
architecture funcao of mux_4_1_select is
begin
    WITH SEL SELECT
    X <= A when "00",
        B when "01",
        C when "10",
        D when "11";
end funcao;
```



## Exercício 2

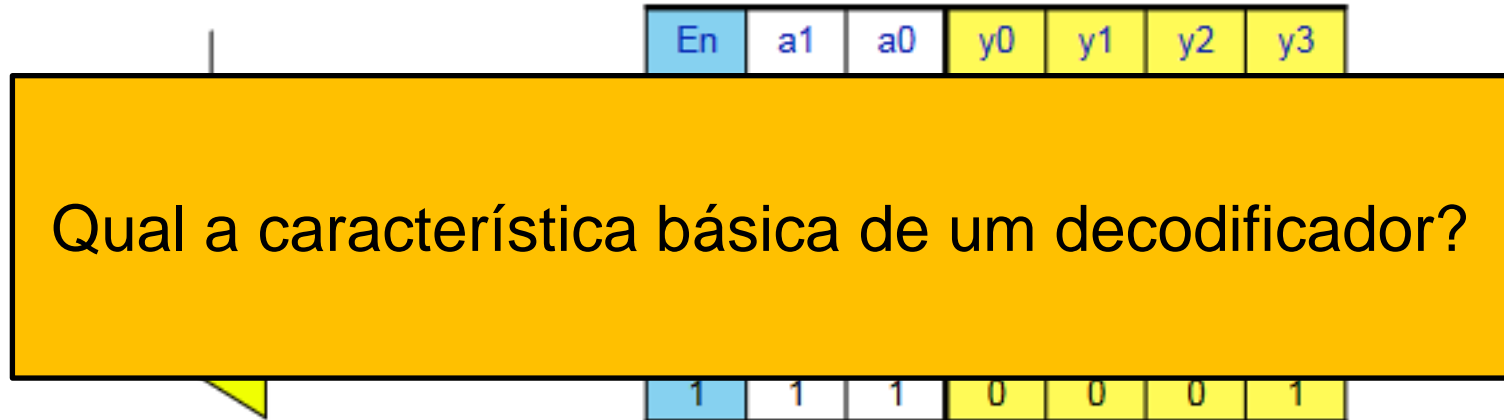
- Mux 4:1 de N bits usando SELECT: Faça um software que coloque na saída X uma das 4 entradas de N bits: A, B, C e D, dependendo dos bits na entrada SEL.

Resultado da simulação para 4 bits:



## Exercício 3

- Projete e simule o decodificador 2:4 com pino de enable abaixo:



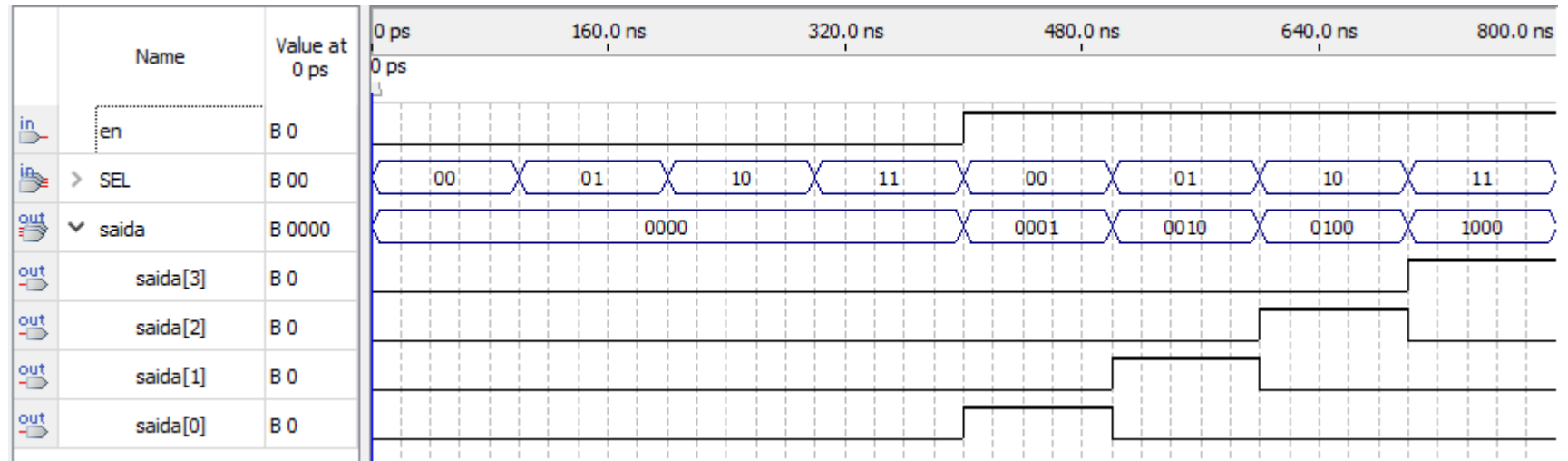
- Grave no KIT utilizando DIP Switch[0] para o enable, DIP Switch [1-2] para as entradas e LED[0-3] para as saídas.

## Exercício 3

```
entity decodificador is
  Port (en : in STD_LOGIC;
        SEL : in STD_LOGIC_VECTOR (1 downto 0);
        saida : out STD_LOGIC_VECTOR (3 downto 0));
end decodificador;

architecture funcao of decodificador is
  signal x: std_logic_vector(3 downto 0);
begin
  with SEL select
    x <= "0001" when "00",
         "0010" when "01",
         "0100" when "10",
         "1000" when "11";
  with en select
    saida <= x when '1',
            "0000" when others;
end funcao;
```

# Exercício 3



## Exercício 4

- Projete e simule o codificador 4:2 com pino de enable abaixo:

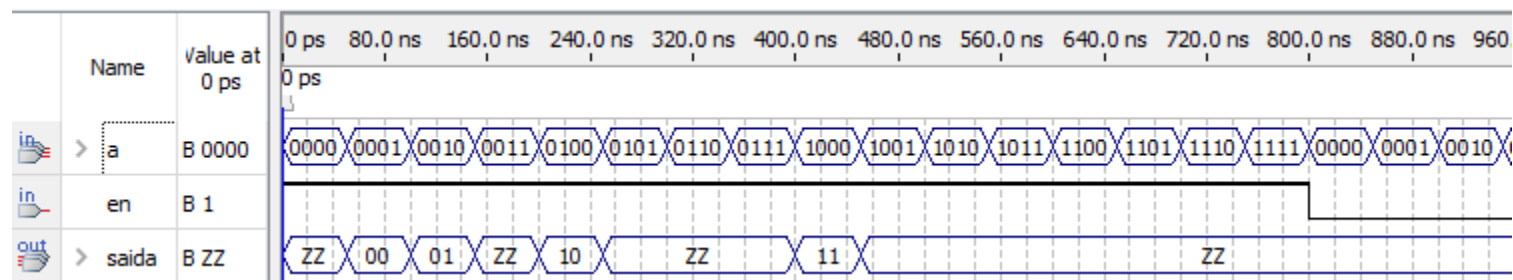
En	a3	a2	a1	a0	y1	y0
0	x	x	x	x	Z	Z
1	0	0	0	1	0	0
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	1	0	0	0	1	1

## Exercício 4

```
entity codificador is
  Port (en : in STD_LOGIC;
        a : in  STD_LOGIC_VECTOR (3 downto 0);
        saida : out STD_LOGIC_VECTOR (1 downto 0));
end codificador;

architecture funcao of codificador is
  signal x: std_logic_vector(1 downto 0);
begin
  with a select
    x <= "00" when "0001",
          "01" when "0010",
          "10" when "0100",
          "11" when "1000",
          "ZZ" when others;

  with en select
    saida <= x when '1',
            "ZZ" when others;
end funcao;
```



- Exemplo: Codificador de prioridade

Entradas			Saídas	
p2	p1	p0	c1	c0
1	–	–	1	1
0	1	–	1	0
0	0	1	0	1
0	0	0	0	0

- Exercício: Faça o codificador de prioridade usando WITH-SELECT-WHEN.

```
LIBRARY ieee;  
USE ieee.std_logic_1164.all;
```

```
ENTITY prioridade IS  
PORT ( p: IN std_logic_vector (2 DOWNT0 0);  
       c: OUT std_logic_vector (1 DOWNT0 0));  
END prioridade;
```

```
ARCHITECTURE funcao OF prioridade IS  
BEGIN
```

```
    WITH P SELECT  
    c<= "11" WHEN "111" | "110" | "101" | "100",  
        "10" WHEN "011" | "010",  
        "01" WHEN "001",  
        "00" WHEN "000";
```

```
END funcao;
```

Entradas			Saídas	
p2	p1	p0	c1	c0
1	–	–	1	1
0	1	–	1	0
0	0	1	0	1
0	0	0	0	0

| = OR para bits.



- Codificador de prioridade usando WHEN-ELSE.

```
LIBRARY ieee;  
USE ieee.std_logic_1164.all;
```

```
ENTITY prioridade IS  
PORT ( p: IN std_logic_vector (2 DOWNTO 0);  
       c: OUT std_logic_vector (1 DOWNTO 0));  
END prioridade;
```

```
ARCHITECTURE funcao OF prioridade IS  
BEGIN  
    c<= "11" WHEN P(2)='1' ELSE  
        "10" WHEN P(1)='1' ELSE  
        "01" WHEN P(0)='1' ELSE  
        "00";  
  
END funcao;
```

Entradas			Saídas	
p2	p1	p0	c1	c0
1	-	-	1	1
0	1	-	1	0
0	0	1	0	1
0	0	0	0	0

- Códigos concorrentes – GENERATE

- PEDRONI, Volnei A. Eletrônica Digital Moderna e VHDL. 1. ed. Campus. 2010, 648 p. ISBN 8535234659