
**Universidade Federal de Pelotas**
**Instituto de Física e Matemática**
Departamento de Informática
**Bacharelado em Ciência da Computação**

# Técnicas Digitais

## Aula 19

**Noções da linguagem VHDL: descrição e simulação de seletores, decodificadores, codificadores e comparadores.**

**Profs. José Luís Güntzel & Luciano Agostini**

**{guntzel,agostini}@ufpel.edu.br**

**www.ufpel.edu.br/~guntzel/TD/TD.html**

# Noções de Linguagem VHDL

## Comandos Atribuição em VHDL

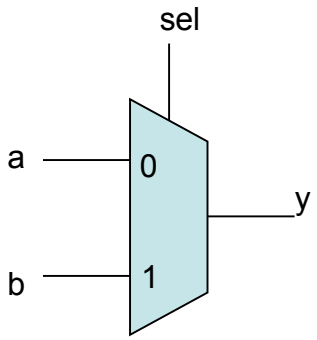
**VHDL provê os seguintes comandos de atribuição:**

- **Simples**
- **Sinal selecionado (*selected signal assignment*)**
- **Sinal condicional (*conditional signal assingment*)**
- **Geração (*generate statement*)**
- **If-then-else (*if-then-else statement*)**
- **Case (*case statement*)**

# Noções de Linguagem VHDL

## Multiplexador 2 por 1 (atribuição com sinal selecionado)

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

ENTITY mux2para1 IS
PORT ( sel, a, b : IN STD_LOGIC;
          y : OUT STD_LOGIC);
END mux2para1;

ARCHITECTURE comportamento OF mux2para1 IS
BEGIN
   WITH sel SELECT
       y <= a WHEN '0',
             b WHEN OTHERS;
END comportamento;
```

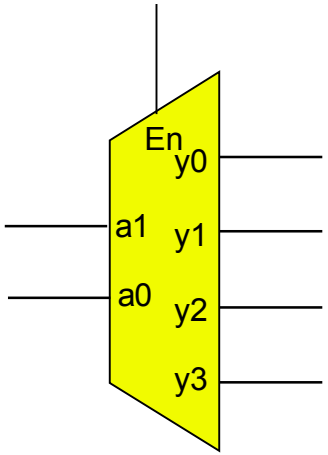



Y foi declarado como STD_LOGIC, logo, pode valer 0, 1, Z, -

# Noções de Linguagem VHDL

## Decodificador 2 por 4 (atribuição com sinal selecionado)

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

ENTITY dec2para4 IS
PORT ( a : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
       En : IN STD_LOGIC;
       y : OUT STD_LOGIC_VECTOR(0 TO 3) );
END dec2para4;

ARCHITECTURE comportamento OF dec2para4 IS
   SIGNAL Ena : STD_LOGIC_VECTOR (2 DOWNTO 0);
BEGIN
   Ena <= En & a;  -- concatenação dos sinais a e enable
   WITH Ena SELECT
      Y <= "1000" WHEN "100";
           "0100" WHEN "101";
           "0010" WHEN "110";
           "0001" WHEN "111";
           "0000" WHEN OTHERS;
END comportamento;
```

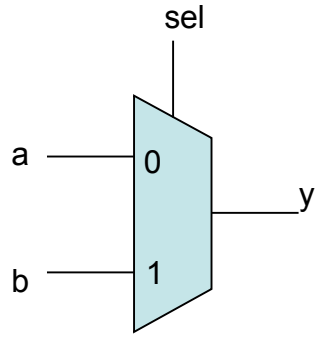



| En | a1 | a0 | y0 | y1 | y2 | y3 |
|---|---|---|---|---|---|---|
| 0 | X | X | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |

# Noções de Linguagem VHDL

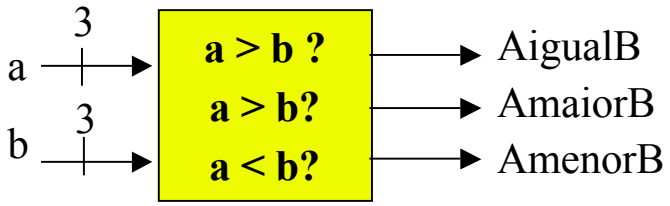
## Multiplexador 2 por 1 (atribuição com sinal condicional)

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

ENTITY mux2para1 IS
PORT ( sel, a, b : IN STD_LOGIC;
            y : OUT STD_LOGIC);
END mux2para1;

ARCHITECTURE comportamento OF mux2para1 IS
BEGIN
   y <= a WHEN sel = '0' ELSE b;
END comportamento;
```



# Noções de Linguagem VHDL

## Comparador de 4 bits (atribuição com sinal condicional)




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

ENTITY comp IS
PORT ( a, b : IN STD_LOGIC_VECTOR (3 DOWNTO 0);
          AigualB, AmaiorB, AmenorB : OUT STD_LOGIC);
END comp;

ARCHITECTURE comportamento OF comp IS
BEGIN
  AigualB <= '1' WHEN A=B ELSE '0';
  AmaiorB <= '1' WHEN A>B ELSE '0';
  AmenorB <= '1' WHEN A<B ELSE '0';
END comportamento;
```

# Noções de Linguagem VHDL

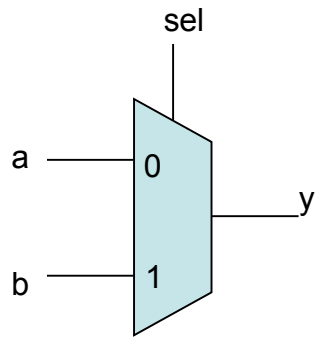
## Processos

- É a forma de gerar uma avaliação **seqüencial** (não concorrente) de atribuições.
- O **processo** é concorrente em relação aos outros elementos da arquitetura.
- A ordem das atribuições passa a ser relevante.
- É usado dentro da arquitetura.
- Precisa de uma **lista de sinais de sensibilização**. Quanto um sinal de sensibilização muda de valor, o processo "acorda" e se torna ativo.
- As atribuições que aparecem dentro do processo não são visíveis de fora do processo até que todas as atribuições do processo tenham sido avaliadas.

# Noções de Linguagem VHDL

## Multiplexador 2 por 1 (utilizando processo)

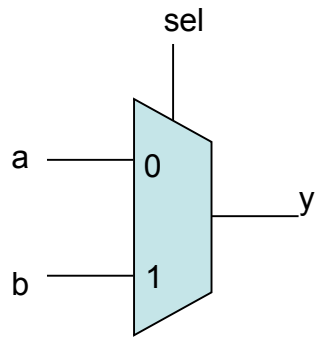
```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY mux2para1 IS
PORT ( sel, a, b : IN STD_LOGIC;
          y : OUT STD_LOGIC);
END mux2para1;

ARCHITECTURE comportamento OF mux2para1 IS
BEGIN
   PROCESS (a, b, sel)    -- lista de sensibilização
   BEGIN
      IF sel = '0' THEN
           y <= a;
      ELSE
           y <= b;
      END IF;
   END PROCESS;
END comportamento;
```



# Noções de Linguagem VHDL

## Multiplexador 2 por 1 (utilizando processo: 2ª versão)

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

ENTITY mux2para1 IS
PORT ( sel, a, b : IN STD_LOGIC;
          y : OUT STD_LOGIC);
END mux2para1;

ARCHITECTURE comportamento OF mux2para1 IS
BEGIN
   PROCESS (a, b, sel)    -- lista de sensibilização
   BEGIN
      y <= a;
      IF sel = '1' THEN
           y <= b;
      END IF;
   END PROCESS;
END comportamento;
```



# Noções de Linguagem VHDL

## Codificador de Prioridade

com processo e atribuição condicional

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

ENTITY prioridade IS
PORT ( w : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
       y : OUT STD_LOGIC_VECTOR(1 DOWNTO 0);
       z : OUT STD_LOGIC);
END prioridade;

ARCHITECTURE comportamento OF prioridade IS
BEGIN
   PROCESS (w)
   BEGIN
      IF w(3) = '1' THEN
           y <= "11";
      ELSIF w(2) = '1' THEN
           y <= "10";
      ELSIF w(1) = '1' THEN
           y <= "01";
      ELSE
           y <= "00";
      END IF;
   END PROCESS;
   z <= '0' WHEN w = "0000" ELSE '1';
END comportamento;
```

# Noções de Linguagem VHDL

## Codificador de Prioridade

**Versão alternativa**

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

ENTITY prioridade IS
PORT ( w : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
       y : OUT STD_LOGIC_VECTOR(1 DOWNTO 0);
       z : OUT STD_LOGIC);
END prioridade;

ARCHITECTURE comportamento OF prioridade IS
BEGIN
   PROCESS (w)
   BEGIN
      y <= "00";
      IF w(1) = '1' THEN y <= "01"; END IF;
      IF w(2) = '1' THEN y <= "10"; END IF;
      IF w(3) = '1' THEN y <= "11"; END IF;

      z <= '1';
      IF w = "0000" THEN z <= '0'; END IF;
   END PROCESS;
END comportamento;
```

# Noções de Linguagem VHDL

## Comparador de 1 bit

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

ENTITY compara IS
PORT ( a, b : IN STD_LOGIC;
          AeqB : OUT STD_LOGIC);
END compara;

ARCHITECTURE comportamento OF compara IS
BEGIN
   PROCESS (a,b)
   BEGIN
      IF a = b THEN
           AeqB <= '1';
      END IF;
    END PROCESS;
END comportamento;
```

**Onde está o erro?**

# Noções de Linguagem VHDL

## Comparador de 1 bit

Versão correta

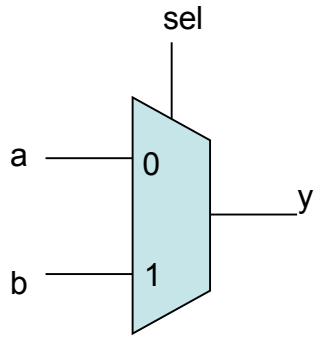
```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY compara IS
PORT ( a, b : IN STD_LOGIC;
          AeqB : OUT STD_LOGIC);
END compara;

ARCHITECTURE comportamento OF compara IS
BEGIN
   PROCESS (a,b)
   BEGIN
       AeqB <= '0';
       IF a = b THEN
            AeqB <= '1';
       END IF;
    END PROCESS;
END comportamento;
```

# Noções de Linguagem VHDL

## Multiplexador 2 por 1

utilizando processo com CASE

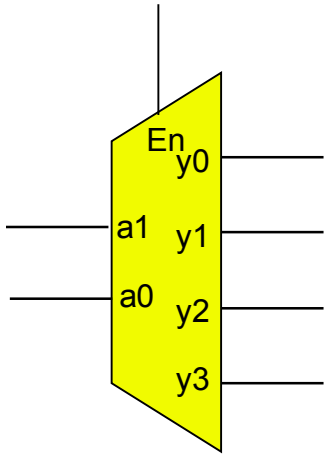
```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY mux2para1 IS
PORT ( sel, a, b : IN STD_LOGIC;
          y : OUT STD_LOGIC);
END mux2para1;

ARCHITECTURE comportamento OF mux2para1 IS
BEGIN
   PROCESS (a, b, sel)    -- lista de sensibilização
   BEGIN
      CASE sel IS
         WHEN '0' =>  y <= a;
         WHEN OTHERS =>  y <= b;
      END CASE;
   END PROCESS;
END comportamento;
```



# Noções de Linguagem VHDL

## Decodificador 2 por 4

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

ENTITY dec2para4 IS
PORT ( a : IN STD_LOGIC_VECTOR (1 DOWNTO 0);
          En : IN STD_LOGIC;
          y : OUT STD_LOGIC_VECTOR (0 TO 3) );
END dec2para4;

ARCHITECTURE comportamento OF dec2para4 IS
BEGIN
   PROCESS (a, En)
   BEGIN
       IF En = '1' THEN
          CASE a IS
              WHEN "00" =>   y <= "1000";
              WHEN "01" =>   y <= "0100";
              WHEN "10" =>   y <= "0010";
              WHEN OTHERS =>   y <= "0001";
           END CASE;
        ELSE
            y <= "0000";
        END IF;
    END PROCESS;
END comportamento;
```




| En | a1 | a0 | y0 | y1 | y2 | y3 |
|---|---|---|---|---|---|---|
| 0 | X | X | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |

# Noções de Linguagem VHDL

## Descrição de uma ULA (equivalente ao TTL74381)

**Funcionalidade:**

| Operação | Controle ('sel') | Saída ('F') |
|---|---|---|
| Clear | 0 0 0 | 0 0 0 0 |
| B – A | 0 0 1 | B – A |
| A – B | 0 1 0 | A – B |
| ADD | 0 1 1 | A + B |
| XOR | 1 0 0 | A XOR B |
| OR | 1 0 1 | A OR B |
| AND | 1 1 0 | A AND B |
| Preset | 1 1 1 | 1 1 1 1 |

# Noções de Linguagem VHDL

## Descrição de uma ULA (74381)

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

ENTITY ula74381 IS
PORT ( s : IN STD_LOGIC_VECTOR (2 DOWNTO 0);
          A, B : IN STD_LOGIC_VECTOR (3 DOWNTO 0);
          F : OUT STD_LOGIC_VECTOR (3 DOWNTO 0));
END ula74381;

ARCHITECTURE comportamento OF ula74381 IS
BEGIN
   PROCESS (s, A, B)
   BEGIN
       CASE s IS
             WHEN “000” =>   F <= “0000”;
             WHEN “001” =>   F <= B – A;
             WHEN “010” =>   F <= A – B;
             WHEN “011” =>   F <= A + B;
             WHEN “100” =>   F <= A XOR B;
             WHEN “101” =>   F <= A OR B;
             WHEN “110” =>   F <= A AND B;
             WHEN OTHERS =>   F <= “1111”;
          END CASE;
END PROCESS;
END comportamento;
```