Universidade Federal de Santa Catarina
Centro Tecnológico
Departamento de Informática e Estatística
Curso de Graduação em Ciências da Computação

Lógica Programável

INE 5348

Aula 3

Comandos de atribuição em VHDL. Processos em VHDL. Descrição em VHDL de multiplexadores e decodificadores (com e sem processos), síntese e simulação.

Prof. José Luís Güntzel
guntzel@inf.ufsc.br

www.inf.ufsc.br/~guntzel/ine5348/ine5348.html

Comandos de Atribuição e Processos em VHDL

Comandos de Atribuição em VHDL

VHDL provê os seguintes comandos de atribuição:

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

} Usados dentro de processos

Comandos de Atribuição e Processos em VHDL

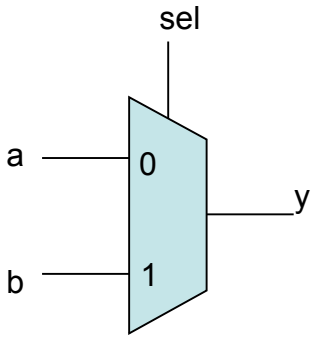
Usando Atribuição com Sinal Selecionado

```
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;
```

Multiplexador 2:1



sel	Y
0	A
1	B

- Deve haver um "WHEN" para cada valor possível de sel
- y foi declarado como STD_LOGIC, logo, pode valer 0, 1, Z, -

Comandos de Atribuição e Processos em VHDL

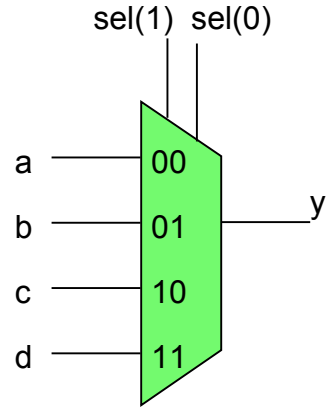
Usando Atribuição com Sinal Selecionado

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

ENTITY mux4para1 IS
PORT (  a, b, c, d : IN STD_LOGIC;
        sel: IN STD_LOGIC_VECTOR(1 DOWNTO 0);
        y : OUT STD_LOGIC);
END mux4para1;

ARCHITECTURE comportamento OF mux4para1 IS
BEGIN
   WITH sel SELECT
      y <= a WHEN "00",
           b WHEN "01",
           c WHEN "10",
           d WHEN OTHERS;
END comportamento;
```

Multiplexador 4:1



sel(1)	sel(0)	y
0	0	a
0	1	b
1	0	c
1	1	d

Comandos de Atribuição e Processos em VHDL

Usando *Package* para Definir um Componente

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

ENTITY mux4para1 IS
PORT (  a, b, c, d : IN STD_LOGIC;
          sel: IN STD_LOGIC_VECTOR(1 DOWNTO 0);
         y : OUT STD_LOGIC);
END mux4para1;

ARCHITECTURE comportamento OF mux4para1 IS
BEGIN
   WITH sel SELECT
      y <= a WHEN “00”,
             b WHEN “01”,
             c WHEN “10”,
             d WHEN OTHERS;
END comportamento;
```

A declaração de **componente** permite que esta entidade seja usada como um subcircuito em outro código VHDL…

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
PACKAGE mux4para1_package IS
      COMPONENT mux4para1 IS
            PORT (  a, b, c, d : IN STD_LOGIC;
                      sel: IN STD_LOGIC_VECTOR(1 DOWNTO 0);
                       y : OUT STD_LOGIC);
            END COMPONENT;
END mux4para1_package;
```

Comandos de Atribuição e Processos em VHDL

Usando *Package* para Definir um Componente

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
LIBRARY work;
USE work.mux4para1_package.all;

ENTITY mux16para1 IS
PORT (  e : IN STD_LOGIC_VECTOR(0 TO 15);
        sel: IN STD_LOGIC_VECTOR(3 DOWNTO 0);
        y : OUT STD_LOGIC);
END mux16para1;

ARCHITECTURE estrutura OF mux16para1 IS
    SIGNAL m: STD_LOGIC_VECTOR(0 TO 3);
BEGIN
  Mux1: mux4para1 PORT MAP
    ( e(0), e(1), e(2), e(3), sel(1 DOWNTO 0), m(0) );
  Mux2: mux4para1 PORT MAP
    ( e(4), e(5), e(6), e(7), sel(1 DOWNTO 0), m(1) );
  Mux3: mux4para1 PORT MAP
    ( e(8), e(9), e(10), e(11), sel(1 DOWNTO 0), m(2) );
  Mux4: mux4para1 PORT MAP
    ( e(12), e(13), e(14), e(15), sel(1 DOWNTO 0), m(3) );
  Mux5: mux4para1 PORT MAP
    ( m(0), m(1), m(2), m(3), sel(1 DOWNTO 0), y );
END estrutura;
```

Declaração do componente mux4para1

Repare que não foi necessário definir as interfaces do componente…

Comandos de Atribuição e Processos em VHDL

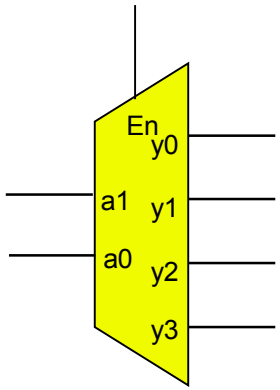
Usando Atribuição com Sinal Selecionado

```
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;
```

Decodificador 2:4



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

Comandos de Atribuição e Processos em VHDL

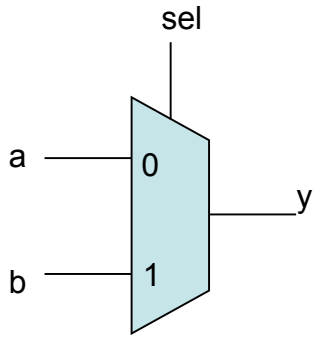
Usando Atribuição com Sinal Condicional

```
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;
```

Multiplexador 2:1



sel	Y
0	A
1	B

Comandos de Atribuição e Processos em VHDL

Usando Atribuição com Sinal 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
    y <= "11" WHEN w(3) = '1' ELSE
         "10" WHEN w(2) = '1' ELSE
         "01" WHEN w(1) = '1' ELSE
         "00";
    z <= '0' WHEN w = "0000" ELSE '1';
END comportamento;
```

Codificador de prioridade 4:2

w3	w2	w1	w0	y1	y0	z
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

Note que há uma prioridade na avaliação dos "WHENs"

Pergunta: qual das duas atribuições acima ocorre primeiro?

Comandos de Atribuição e Processos em VHDL

Usando Atribuição com Sinal Selecionado

```
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
    WITH w SELECT
        y <= "00" WHEN "0001",
             "01" WHEN "0010",
             "01" WHEN "0011",
             "10" WHEN "0100",
             "10" WHEN "0101",
             "10" WHEN "0110",
             "10" WHEN "0111",
             "11" WHEN OTHERS;
    WITH w SELECT
        z <= '0' WHEN '0000,
             '1' WHEN OTHERS;
END comportamento;
```

Codificador de prioridade 4:2

w3	w2	w1	w0	y1	y0	z
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

Não há prioridade. É necessário descrever todas as opções. (Resulta em um código deselegante…)

Comandos de Atribuição e Processos em VHDL

Usando Atribuição com Sinal Condicional

Comparador de 4 bits

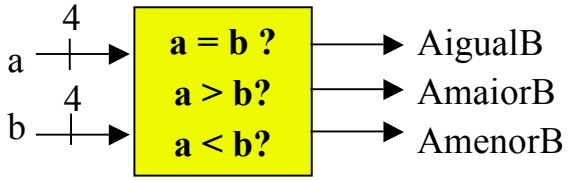
(1ª versão, para números sem sinal)

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

Permite que sinais STD_LOGIC_VECTOR sejam tratados como números binários sem sinal com os operadores relacionais de VHDL



Comandos de Atribuição e Processos em VHDL

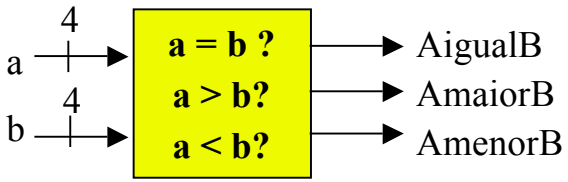
Usando Atribuição com Sinal Condicional

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

ENTITY comp IS
PORT ( A, B : IN SIGNED (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;
```

Comparador de 4 bits
(2ª versão, para números com sinal)



Comandos de Atribuição e Processos em VHDL

Usando Atribuição com Geração

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
LIBRARY work;
USE work.mux4para1_package.all;
```

Declaração do componente mux4para1

```
ENTITY mux16para1 IS
PORT (  e : IN STD_LOGIC_VECTOR(0 TO 15);
          sel: IN STD_LOGIC_VECTOR(3 DOWNTO 0);
         y : OUT STD_LOGIC);
END mux16para1;

ARCHITECTURE estrutura OF mux16para1 IS
      SIGNAL m: STD_LOGIC_VECTOR(0 TO 3);
BEGIN
  Mux1: mux4para1 PORT MAP
      ( e(0), e(1), e(2), e(3), sel(1 DOWNTO 0), m(0) );
  Mux2: mux4para1 PORT MAP
      ( e(4), e(5), e(6), e(7), sel(1 DOWNTO 0), m(1) );
  Mux3: mux4para1 PORT MAP
      ( e(8), e(9), e(10), e(11), sel(1 DOWNTO 0), m(2) );
  Mux4: mux4para1 PORT MAP
      ( e(12), e(13), e(14), e(15), sel(1 DOWNTO 0), m(3) );
  Mux5: mux4para1 PORT MAP
      ( m(0), m(1), m(2), m(3), sel(3 DOWNTO 2), y );
END estrutura;
```

Estrutura regular:
usar “for generate”…

Comandos de Atribuição e Processos em VHDL

Usando Atribuição com Geração

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
LIBRARY work;
USE work.mux4para1_package.all;

ENTITY mux16para1 IS
PORT (  e : IN STD_LOGIC_VECTOR(0 TO 15);
          sel: IN STD_LOGIC_VECTOR(3 DOWNTO 0);
         y : OUT STD_LOGIC);
END mux16para1;

ARCHITECTURE estrutura OF mux16para1 IS
      SIGNAL m: STD_LOGIC_VECTOR(0 TO 3);
BEGIN
  G1: FOR i IN 0 TO 3 GENERATE
      Muxes: mux4para1 PORT MAP (
            e(4*i), e(4*i+1), e(4*i+2), e(4*i+3), sel(1 DOWNTO 0), m(i) );
  END GENERATE;
  Mux5: mux4para1 PORT MAP ( m(0), m(1), m(2), m(3), sel(3 DOWNTO 2), y );
END estrutura;
```

i é uma variável definida automaticamente, com escopo limitado ao "FOR GENERATE"

Comandos de Atribuição e Processos em VHDL

Usando Atribuição com Geração

Um decodificador 4 para 16, construído de maneira hierárquica

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

ENTITY dec2para16 IS
PORT ( a : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
       En : IN STD_LOGIC;
       y : OUT STD_LOGIC_VECTOR(0 TO 15) );
END dec2para16;

ARCHITECTURE comportamento OF dec2para16 IS
   COMPONENT dec2para4
      PORT ( a : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
        En : IN STD_LOGIC;
        y : OUT STD_LOGIC_VECTOR(0 TO 3) );
   END COMPONENT;
   SIGNAL m: STD_LOGIC_VECTOR(0 TO 3);
BEGIN
    G1: FOR i IN 0 TO 3 GENERATE
    Dec_direita: dec2para4 PORT MAP ( a(1 DOWNTO 0), m(i), y(4*i TO 4*i+3) );
          G2: IF i=3 GENERATE
               Dec_esquerda: dec2para4 PORT MAP( a(i DOWNTO i-1) , En, m );
          END GENERATE;
     END GENERATE;
END comportamento;
```

Comandos de Atribuição e Processos em VHDL

Experimento da Aula de Hoje

Descrever em VHDL e sintetizar (para dispositivo Stratix) as versões de mux 4:1 especificadas na tabela abaixo. Preencher os dados solicitados e procurar justificar os resultados. Simular ao menos uma das versões.

Versão de mux4:1	nº de LEs	Atraso crítico	Caminho crítico	Dispositivo selecionado
1. Equação lógica				
2. Hieráquica (a partir de mux2:1 feito com equação)				
3. Sinal selecionado (slide 4)				
4. Sinal condicional				
5. Processo (c/ IF THEN ELSE)				
6. Processo (c/ CASE)				

Repetir (em casa) o experimento, porém selecionando dispositivo da família Flex10K…

Comandos de Atribuição e Processos em VHDL

Processos

- Usados para 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.

Comandos de Atribuição e Processos em VHDL

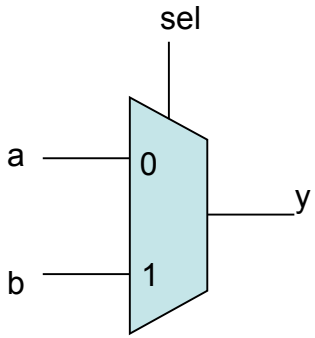
Usando Processo

```
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;
```

Multiplexador 2:1



sel	Y
0	A
1	B

Comandos de Atribuição e Processos em VHDL

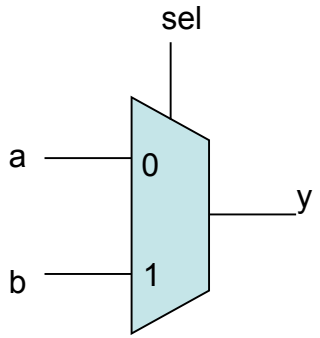
Usando Processo (2ª versão)

```
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;
```

Multiplexador 2:1



sel	Y
0	A
1	B

Comandos de Atribuição e Processos em VHDL

Usando Processo e Atribuição com Sinal 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;
```

Codificador de prioridade 4:2

w3	w2	w1	w0	y1	y0	z
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

Comandos de Atribuição e Processos em VHDL

Usando Processos

(Uma 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;
```

Codificador de prioridade 4:2

w3	w2	w1	w0	y1	y0	z
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

Comandos de Atribuição e Processos em VHDL

Usando Processos

```
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;
```

Comparador de 1 bit

Onde está o erro?

Comandos de Atribuição e Processos em VHDL

Usando Processos

```
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;
```

Comparador de 1 bit

Versão correta

Comandos de Atribuição e Processos em VHDL

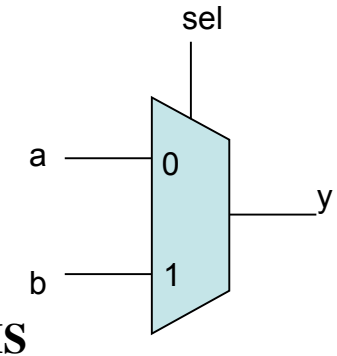
Usando Processo e Case

```
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;
```

Multiplexador 2:1



sel	Y
0	A
1	B

Comandos de Atribuição e Processos em VHDL

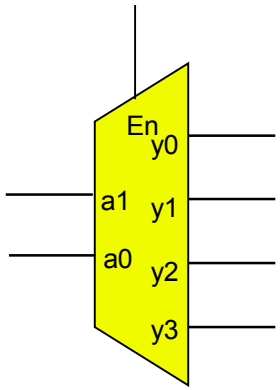
Usando Processo

```
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;
```

Decodificador 2:4



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

Comandos de Atribuição e Processos em VHDL

Usando Processo

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

Comandos de Atribuição e Processos em VHDL

Usando Processo

```
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;
```