

ANEXO C

Duas soluções alternativas para a implementação do multiplexador 4x1, exemplificado na Seção 2.2.3, são apresentadas com utilização de WITH/SELECT/WHEN (selected WHEN), conforme os códigos VHDL transcritos como segue:

```
-----
-- Circuito: multiplexador 4x1:(mux2_4x1.vhd)
--      sel (1:2) Selecao da entrada
--          a Entrada, sel = 00
--          b Entrada, sel = 01
--          c Entrada, sel = 10
--          d Entrada, sel = 11
--          y Saída (WITH/SELECT/WHEN)
-----
LIBRARY ieee;
USE ieee.std_logic_1164.all;
-----
ENTITY mux4x1 IS
    PORT (sel: IN STD_LOGIC_VECTOR (1 DOWNTO 0);
          a, b, c, d: IN STD_LOGIC;
          y: OUT STD_LOGIC);
END mux4x1;
-----
ARCHITECTURE mux2 OF mux4x1 IS
BEGIN
    WITH sel SELECT
        y <=  a WHEN "00",    -- nota: ";" troca por ","
              b WHEN "01",
              c WHEN "10",
              d WHEN OTHERS; --não é permitido d WHEN "11"
END mux2;
-----
```

Na descrição acima, a entrada para seleção (sel) poderia ter sido declarada como um inteiro, e neste caso o código VHDL seria o seguinte:

```
-----
-- Circuito: multiplexador 4x1:(mux3_4x1.vhd)
--      sel (1:2) Selecao da entrada
--          a Entrada, sel = 00
--          b Entrada, sel = 01
--          c Entrada, sel = 10
--          d Entrada, sel = 11
--          y Saída (WITH/SELECT/WHEN)
-----
LIBRARY ieee;
USE ieee.std_logic_1164.all;
-----
ENTITY mux4x1 IS
    PORT (sel: IN INTEGER RANGE 0 TO 3;
          a, b, c, d: IN STD_LOGIC;
          y: OUT STD_LOGIC);
END mux4x1;
-----
ARCHITECTURE mux3 OF mux4x1 IS
BEGIN
```

```

WITH sel SELECT
    y <= a WHEN 0,
        b WHEN 1,
        c WHEN 2,
        d WHEN 3;    -- aqui, 3 ou OTHERS são equivalentes,
END mux3;           -- e são testados para todas as opções
-----

```