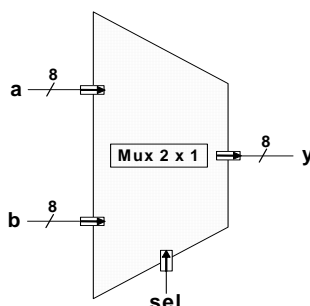


ANEXO B

Duas descrições são apresentadas com uso da declaração WHEN/ELSE, conforme os códigos VHDL transcritos como segue:

1. Este exemplo apresenta um multiplexador 2x1 (dois barramentos de 8 bits x um barramentos de 8 bits)



```
-----
-- Circuito: multiplexador 2x1:(mux8b_2x1.vhd)
--           sel8b Selecao da entrada
--           a8b Entrada, sel = 0
--           b8b Entrada, sel = 1
--           y8b Saída (WHEN/ELSE)
-----

LIBRARY ieee;
USE ieee.std_logic_1164.all;

-----

ENTITY mux8b2x1 IS
    PORT ( sel8b    : IN STD_LOGIC;
          a8b      : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
          b8b      : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
          y8b      : OUT STD_LOGIC_VECTOR (7 DOWNTO 0));
END mux8b2x1;

-----

ARCHITECTURE mux8b OF mux8b2x1 IS
BEGIN
    y8b <=    a8b WHEN sel8b='0' ELSE b8b;
END mux8b;

-----
```

Sugere-se como *testbench* o código VHDL transcrito a seguir:

```
-----
-- *****
-- Testbench para simulação Funcional do
-- Circuito: multiplexador 2x1:(mux8b_2x1.vhd)
--           sel8b Selecao da entrada
--           a8b Entrada, sel = 0
--           b8b Entrada, sel = 1
--           y8b Saída (WHEN/ELSE)
-- *****
ENTITY testbench6a IS END;
-----

-- Testbench para mux8b_2x1.vhd
```

```

-- Validação assíncrona
-----
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_signed.all;
USE std.textio.ALL;

ARCHITECTURE tb_sel8b_2x1 OF testbench6a IS
-----
-- Declaração do componente mux2x1
-----
component mux8b2x1
    PORT ( sel8b      :IN STD_LOGIC;
          a8b        :IN STD_LOGIC_VECTOR (7 DOWNTO 0);
          b8b        :IN STD_LOGIC_VECTOR (7 DOWNTO 0);
          y8b        :OUT STD_LOGIC_VECTOR (7 DOWNTO 0));
end component;

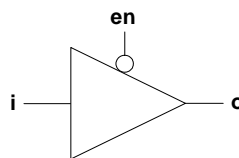
signal tb_sel8b      :std_logic;
signal tb_a8b, tb_b8b :STD_LOGIC_VECTOR (7 DOWNTO 0);
Begin

mux8b: mux8b2x1 PORT MAP (sel8b => tb_sel8b, a8b => tb_a8b,
                        b8b => tb_b8b, y8b => open);

estimulo: PROCESS
begin
    tb_sel8b <= '0';
    tb_a8b <= (OTHERS => '0');
    tb_b8b <= (OTHERS => '0');
    wait for 10 ns;
    loop
        tb_a8b <= tb_a8b + "00000110";
        WAIT FOR 10 ns;
        tb_sel8b <= '1';
        WAIT FOR 10 ns;
        tb_b8b <= tb_b8b + "00000111";
        WAIT FOR 10 ns;
        tb_sel8b <= '0';
        WAIT FOR 10 ns;
    end loop;
end PROCESS estimulo;
end tb_sel8b_2x1;

```

2. Nesse exemplo é apresentado um buffer *tri-state*, cuja função é desacoplar eletricamente conexões de duas ou mais saídas simultâneas.



```

-----
-- Circuito: Buffer 3 State: (buff_3stat.vhd)
--
--          en saída
--          i  Entrada
--          o  Saída
-----
LIBRARY ieee;
USE ieee.std_logic_1164.all;
-----
ENTITY buff3stat IS
    PORT (en : IN STD_LOGIC;
          i  : IN STD_LOGIC_VECTOR (7 DOWNTO 0);

```

```

        o : OUT STD_LOGIC_VECTOR (7 DOWNTO 0));
    END buff3stat;
-----
ARCHITECTURE buff_3stat OF buff3stat IS
    BEGIN
        o <= i WHEN (en='1') ELSE
            (OTHERS => 'Z');
    END buff_3stat;
-----

```

Sugere-se como *testbench* para buffer *tri-state* acoplado à saída do mux 2x1 o código transcrito a seguir:

```

-- *****
-- Testbench para simulação Funcional dos
-- Circuito: multiplexador 2x1:(mux8b_2x1.vhd)
--          sel8b Selecao da entrada
--          a8b Entrada, sel = 0
--          b8b Entrada, sel = 1
--          y8b Saída (WHEN/ELSE)
-- Circuito: Buffer tri-state: (buff_3stat.vhd)
--          en saida
--          i  Entrada
--          o  Saida
-- *****
ENTITY testbench6a IS END;
-----
-- Testbench para mux8b_2x1.vhd
-- Validação assincrona
-----

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_signed.all;
USE std.textio.ALL;

ARCHITECTURE tb_sel8b_2x1 OF testbench6a IS
-----
-- Declaração do componente mux2x1
-----
component mux8b2x1
    PORT ( sel8b :IN STD_LOGIC;
          a8b   :IN STD_LOGIC_VECTOR (7 DOWNTO 0);
          b8b   :IN STD_LOGIC_VECTOR (7 DOWNTO 0);
          y8b   :OUT STD_LOGIC_VECTOR (7 DOWNTO 0));
end component;

-----
-- Declaração do buffer tri-state
-----

component buff3stat
    PORT (en : IN STD_LOGIC;
          i  : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
          o  : OUT STD_LOGIC_VECTOR (7 DOWNTO 0));
end component;

signal tb_sel8b :std_logic;
signal tb_a8b, tb_b8b :STD_LOGIC_VECTOR (7 DOWNTO 0);
signal tb_i :STD_LOGIC_VECTOR (7 DOWNTO 0);

Begin

mux8b: mux8b2x1 PORT MAP (sel8b => tb_sel8b, a8b => tb_a8b,
                        b8b => tb_b8b, y8b => tb_i);

buff3: buff3stat PORT MAP (en => tb_sel8b, i => tb_i, o => open);

estimulo: PROCESS
    begin

```