UNIVERSIDADE FEDERAL DE MINAS GERAIS
UFMG – CAMPUS PAMPULHA

Disciplina: Laboratório de Sistemas Digitais
Professor: Ricardo de Oliveira Duarte
Estudantes: Igor Braga de Lima, Matheus Vinícius Freitas Oliveira dos Santos e Stéphanie Pereira Barbosa
Turma: PN5
Guia de aula: 03

ATIVIDADE TEÓRICA

1) Qual é o paradigma de codificação em VHDL?
O paradigma de codificação em VHDL é fundamentado no paralelismo e na concorrência, visando à descrição de hardware. Logo, devido a esses princípios, deve-se codificar em VHDL tendo em mente o circuito/sistema digital desejado e suspender a ideia de programação sequencial.

2) O que são declarações na linguagem VHDL?
Em geral, declarações podem ser consideradas ações executadas pelo processador. Em VHDL essas declarações são concorrentes.

3) O que são declarações concorrentes em VHDL?
Declarações concorrentes são similares às declarações em linguagem sequenciais, com a diferença da ordem de execução. As declarações concorrentes são executadas de modo a simular o paralelismo (execução de todas as declarações ao mesmo tempo) por um processador.

4) Qual é o operador de declaração concorrente de *signal* em VHDL?
"<=". É utilizado da seguinte forma:
<*target*> <= <*expression*>.

5) A ordem que os operadores de declaração concorrente de sinais em uma descrição VHDL importa na funcionalidade do circuito?
Como as declarações são executadas de forma concorrente em VHDL, a ordem não importa.

6) Para que serve um operador de declaração concorrente de *signal*?
Para indicar que a ação ocorre de forma simultânea, sem atraso em todas as declarações.

7) Cite todas as declarações concorrentes em VHDL estudada no capítulo do livro.

Signal assignment, *process statement*, *conditional signal assignment* e *selected signal assignment.*

8) Qual a ideia principal do funcionamento de uma declaração concorrente em VHDL?
A ideia é que assim que o valor do sinal de entrada for alterado, o de saída também sofrerá alteração, tal qual em um circuito físico.

9) Em qual situação que uma declaração condicional *when* deve ser usada?
Quando a afirmação tem um único alvo e pode ter mais de uma expressão associada. Ex.: <alvo> <= <expressão> when <condição> else
<expressão> when <condição> else
<expressão>;

10) A declaração condicional *when* é sequencial ou concorrente?
Concorrente.

11) Quando é que uma declaração condicional *when* é avaliada ou executada?
Sempre que ocorre uma mudança nos sinais condicionais.

12) Qual situação que uma declaração condicional *with select* deve ser usada?
Quando se quer selecionar sinais mutuamente excludentes.

13) A declaração condicional *with select* é sequencial ou concorrente?
Concorrente.

14) Quando é que uma declaração condicional *with select* é avaliada ou executada?
Toda as vezes em que há uma mudança na expressão.

15) A declaração *Process* é sequencial ou concorrente?
Ela é concorrente com as outras declarações, mas dentro da declaração ela é sequencial.

ATIVIDADE PRÁTICA

1) Escolha uma das funções do exercício 1 da pág. 48 da referência principal do curso e implemente-o em VHDL.
A função escolhida foi: f) $\mathrm{F}(\mathrm{A},\mathrm{B},\mathrm{C},\mathrm{D}) = \sum(1,2)$

funcao1F.vhd > ...

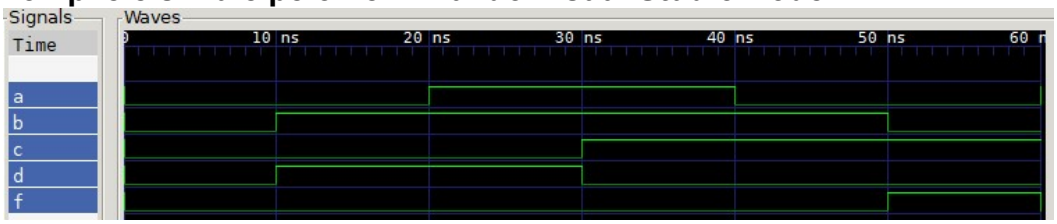
```vhdl
-- Solução do exercício 1f usando atribuição de sinal concorrente.
-- 1-f) F(A, B, C, D) = Somatório(1,2)

library ieee;
use ieee.std_logic_1164.all;

entity exercicio_1 is
  port(A, B, C, D : in std_logic;
       F : out std_logic);
end exercicio_1;

architecture funcao1F of exercicio_1 is
  begin
    F <= ((NOT A) AND (NOT B) AND (NOT C) AND D) OR ((NOT A) AND (NOT B) AND C AND (NOT D));
end funcao1F;
```

2) Escreva um *testbench* para a função em VHDL que você implementou.

tb_funcao1F.vhd > ...

```vhdl
-- Testbench da função 1f

library ieee;
use ieee.std_logic_1164.all;

entity tb_funcao1F is
end tb_funcao1F;

architecture teste of tb_funcao1F is
  component exercicio_1 is
    port(A, B, C, D : in std_logic;
         F : out std_logic);
  end component;

  signal a, b, c, d, f: std_logic;
  begin
    instancia_somador: exercicio_1 port map(A => a, B => b, C => c, D => d, F => f);
    a <= '0', '1' after 20 ns, '0' after 40 ns, '1' after 60 ns;
    b <= '0', '1' after 10 ns, '1' after 30 ns, '0' after 50 ns;
    c <= '0', '0' after 10 ns, '1' after 30 ns, '1' after 50 ns;
    d <= '0', '1' after 10 ns, '0' after 30 ns, '0' after 50 ns;
end teste;
```

3) Compile e simule pelo Terminal do Visual Studio Code.



4) Escolha uma das funções do exercício 2 da pág. 48 da referência principal do curso e implemente-o em VHDL.

A função escolhida foi: d) $\mathrm{F(A, B, C, D)} = \sum(1,2)$

Conditional:

funcao2D.vhd > ...

```vhdl
-- Solução do exercício 2d usando atribuição de sinal condicional.
-- 2-d) F(A, B, C, D) = Somatório(1,2)

library ieee;
use ieee.std_logic_1164.all;

entity exercicio_2Conditional is
  port(A, B, C, D : in std_logic;
       F : out std_logic);
end exercicio_2Conditional;

architecture funcao2D of exercicio_2Conditional is
  begin
    F <= '1' when (A = '0' and B = '0' and C = '1' and D = '0') else
         '1' when (A = '0' and B = '0' and C = '0' and D = '1') else
         '0';
end funcao2D;
```

Selected:

funcao2D.vhd > ...

```vhdl
-- Solução do exercício 2d usando atribuição de sinal selecionado.
-- 2-d) F(A, B, C, D) = Somatório(1,2)

library ieee;
use ieee.std_logic_1164.all;

entity exercicio_2Selected is
  port(A, B, C, D : in std_logic;
       F : out std_logic);
end exercicio_2Selected;

architecture funcao2D of exercicio_2Selected is
  signal t_sig : std_logic_vector(3 downto 0);
  begin
    t_sig <= (A & B & C & D);
    with (t_sig) select
      F <= D when "0001",
           C when "0010",
          '0' when others;
end funcao2D;
```

5) Escreva um *testbench* para a função em VHDL que você implementou.

Conditional:

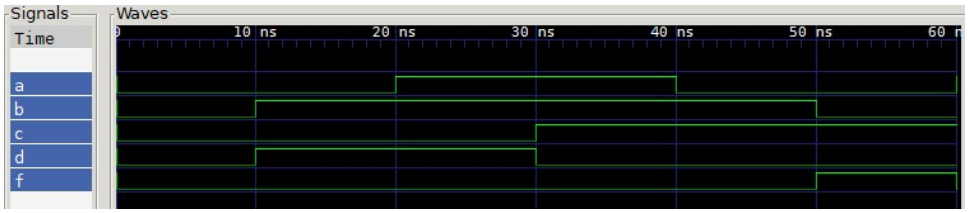
```vhdl
-- Testbench da função 2d

library ieee;
use ieee.std_logic_1164.all;

entity tb_funcao2D is
end tb_funcao2D;

architecture teste of tb_funcao2D is
  component exercicio_2Conditional is
    port(A, B, C, D : in std_logic;
         F : out std_logic);
  end component;

  signal a, b, c, d, f: std_logic;
  begin
    instancia_somador: exercicio_2Conditional port map(A => a, B => b, C => c, D => d, F => f);
    a <= '0', '1' after 20 ns, '0' after 40 ns, '1' after 60 ns;
    b <= '0', '1' after 10 ns, '1' after 30 ns, '0' after 50 ns;
    c <= '0', '0' after 10 ns, '1' after 30 ns, '1' after 50 ns;
    d <= '0', '1' after 10 ns, '0' after 30 ns, '0' after 50 ns;
end teste;
```

Selected:

```vhdl
-- Testbench da função 2d

library ieee;
use ieee.std_logic_1164.all;

entity tb_funcao2D is
end tb_funcao2D;

architecture teste of tb_funcao2D is
  component exercicio_2Selected is
    port(A, B, C, D : in std_logic;
         F : out std_logic);
  end component;

  signal a, b, c, d, f: std_logic;
  begin
    instancia_somador: exercicio_2Selected port map(A => a, B => b, C => c, D => d, F => f);
    a <= '0', '1' after 20 ns, '0' after 40 ns, '1' after 60 ns;
    b <= '0', '1' after 10 ns, '1' after 30 ns, '0' after 50 ns;
    c <= '0', '0' after 10 ns, '1' after 30 ns, '1' after 50 ns;
    d <= '0', '1' after 10 ns, '0' after 30 ns, '0' after 50 ns;
end teste;
```

6) Compile e simule pelo Terminal do Visual Studio Code.

Conditional:



Selected: