

PCS3225

Sistemas Digitais II

Biestáveis e Máquinas de Estados em VHDL

Elaborado por Edson Midorikawa

PCS3225 - 2015

1

Tópicos

- Circuitos sequenciais em VHDL
 - comando `process`
 - exemplos
- Biestáveis em VHDL
- Máquinas de Estados em VHDL
- Referência: **Wakerly**, pág. 625-645 e pág. 697-701.

PCS3225 - 2015

2

Circuitos Sequenciais em VHDL

- Circuitos sequenciais podem ser descritos em VHDL usando o comando `process`

```
architecture arch of ent is
begin
  nome_processo: process (clock)
  begin
    comando sequencial;
    comando sequencial;
    wait until (condição);
    comando sequencial;
    ...
    wait for (time);
    ...
  end process;
end arch;
```

Um processo deve ter uma lista de sensibilidade OU ...

... OU, pelo menos, deve ter um comando *wait*.

PCS3225 - 2015

3

Slide elaborado por Bruno Albertini.

Circuitos Sequenciais em VHDL

- Comando WAIT

- Causa a suspensão do processo

```
wait [sensitivity_clause][condition_clause][timeout_clause];
```

- Sensitivity_clause ::= on signal_name
 - wait on CLOCK;
- Condition_clause ::= until boolean_expression
 - wait until Clock = '1';
- Timeout_clause ::= for time_expression
 - wait for 150 ns;

PCS3225 - 2015

4

Slide elaborado por Bruno Albertini.

Circuitos Sequenciais em VHDL

- Lista de sensibilidade ou WAIT

```
Summation:
PROCESS( A, B, Cin)
BEGIN
    Sum <= A xor B xor Cin;
END PROCESS Summation;
```

=

```
Summation: PROCESS
BEGIN
    Sum <= A xor B xor Cin;
    WAIT ON A, B, Cin;
END PROCESS Summation;
```

PCS3225 - 2015

5

Slide elaborado por Bruno Albertini.

Circuitos Sequenciais em VHDL

- Equivalência
 - cada *statement* concorrente equivale a um processo

```
q <= a xor b after 5 ns;
```

```
process
begin
    q <= a xor b after 5 ns;
    wait on a, b;
end process;
```

PCS3225 - 2015

6

Slide elaborado por Bruno Albertini.

Circuitos Sequenciais em VHDL

- Wakerly, pág. 625-628

PCS3225 - 2015

7

Circuitos Sequenciais em VHDL

```

library IEEE;
use IEEE.std_logic_1164.all;

entity VposDff is
  port (CLK, CLR, D: in STD_LOGIC;
        Q: out STD_LOGIC );
end VposDff;

architecture VposDff_arch of VposDff is
begin
  process (CLK, CLR)
  begin
    if CLR='1' then Q <= '0';
    elsif CLK'event and CLK='1' then Q <= D;
    end if;
  end process;
end VposDff_arch;

```

Table 7-33

Behavioral VHDL for a positive-edge-triggered D flip-flop.

PCS3225 - 2015

8

Circuitos Sequenciais em VHDL

```

process
    wait until CLK'event and CLK='1';
    Q <= D;
end process;

Q <= D when CLK'event and CLK='1' else Q;

```

Table 7-34

Two more ways to describe a positive-edge-triggered D flip-flop.

PCS3225 - 2015

9

Circuitos Sequenciais em VHDL

```

architecture TB_arch of TB is
    signal MCLK: STD_LOGIC;
    signal ... -- Declare other input and output signals

    process -- Clock generator
    begin
        MCLK <= '1'; -- Start at 1 at time 0
        loop
            MCLK <= '0' after 6 ns;
            MCLK <= '1' after 4 ns;
        end loop;
    end process;

    process -- Generate the rest of the input stimuli, check outputs
    begin
        ...
    end;

```

Table 7-36

Clock process within a test bench.

PCS3225 - 2015

10

Biestáveis em VHDL

- Wakerly, pág. 697-700.

PCS3225 - 2015

11

Biestáveis em VHDL

```

entity Vdlatch is
  port (D, C: in STD_LOGIC;
        Q, QN: buffer STD_LOGIC );
end Vdlatch;

architecture Vdlatch_s of Vdlatch is
  signal DN, SN, RN: STD_LOGIC;
  component inv port (I: in STD_LOGIC; O: out STD_LOGIC ); end component;
  component nand2b port (IO, I1: in STD_LOGIC; O: buffer STD_LOGIC ); end component;
begin
  U1: inv port map (D,DN);
  U2: nand2b port map (D,C,SN);
  U3: nand2b port map (C,DN,RN);
  U4: nand2b port map (SN,QN,Q);
  U5: nand2b port map (Q,RN,QN);
end Vdlatch_s;

```

Table 8-3

VHDL structural program for the D latch in Figure 7-12.

PCS3225 - 2015

12

Biestáveis em VHDL

```

entity Vdlatch is
  port (D, C: in STD_LOGIC;
        Q, QN: out STD_LOGIC );
end Vdlatch;

architecture Vdlatch_s2 of Vdlatch is
  signal DN, SN, RN, IQ, IQN: STD_LOGIC;
  component inv port (I: in STD_LOGIC; O: out STD_LOGIC ); end component;
  component nand2 port (IO, I1: in STD_LOGIC; O: out STD_LOGIC ); end component;
begin
  U1: inv port map (D,DN);
  U2: nand2 port map (D,C,SN);
  U3: nand2 port map (C,DN,RN);
  U4: nand2 port map (SN,IQN,IQ);
  U5: nand2 port map (IQ,RN,IQN);
  Q <= IQ; QN <= IQN;
end Vdlatch_s2;

```

Table 8-4

Alternative VHDL structural code for the D latch in Figure 7-12.

PCS3225 - 2015

13

Biestáveis em VHDL

```

architecture Vdlatch_b of Vdlatch is
begin
  process(C, D, Q)
  begin
    if (C='1') then Q <= D; else Q <= Q; end if;
    QN <= not Q;
  end process;
end Vdlatch_b;

```

Table 8-5

VHDL behavioral architecture for a D latch.

PCS3225 - 2015

14

Biestáveis em VHDL

```

entity Vdff is
  port (D, CLK: in STD_LOGIC;
        Q: out STD_LOGIC );
end Vdff;

architecture Vdff_b of Vdff is
begin
  process(CLK)
  begin
    if (CLK'event and CLK='1') then Q <= D; end if;
  end process;
end Vdff_b;

```

Table 8-6

VHDL behavioral model of an edge-triggered D flip-flop.

PCS3225 - 2015

15

Biestáveis em VHDL

```

library IEEE;
use IEEE.std_logic_1164.all;

entity Vdff74 is
  port (D, CLK, PR_L, CLR_L: in STD_LOGIC;
        Q, QN: out STD_LOGIC );
end Vdff74;

architecture Vdff74_b of Vdff74 is
  signal PR, CLR: STD_LOGIC;
begin
  process(CLR_L, CLR, PR_L, PR, CLK)
  begin
    PR <= not PR_L; CLR <= not CLR_L;
    if (CLR and PR) = '1' then Q <= '1'; QN <= '1';
    elsif CLR = '1' then Q <= '0'; QN <= '1';
    elsif PR = '1' then Q <= '1'; QN <= '0';
    elsif (CLK'event and CLK='1') then Q <= D; QN <= not D;
    end if;
  end process;
end Vdff74_b;

```

Table 8-7

VHDL model of a 74x74-like D flip-flop with preset and clear.

PCS3225 - 2015

16

Máquinas de Estados em VHDL

- Wakerly, pág.628-633

PCS3225 - 2015

17

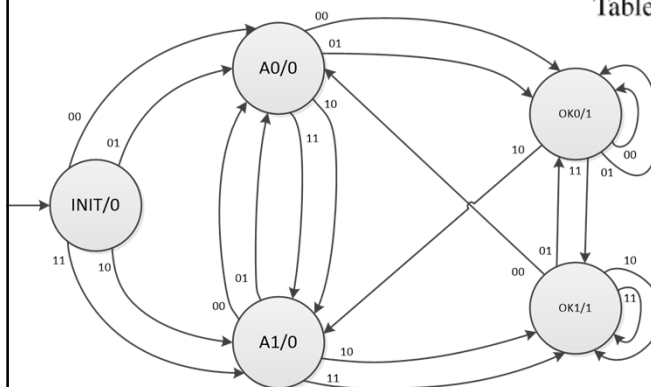
Máquinas de Estados em VHDL

- Exemplo

<i>S</i>	<i>A B</i>				<i>Z</i>
	<i>00</i>	<i>01</i>	<i>11</i>	<i>10</i>	
INIT	A0	A0	A1	A1	0
A0	OK0	OK0	A1	A1	0
A1	A0	A0	OK1	OK1	0
OK0	OK0	OK0	OK1	A1	1
OK1	A0	OK0	OK1	OK1	1

S*

Table 7-5 State and output table for example problem.



PCS3225 - 2015

18

Máquinas de Estados em VHDL

- Descrição VHDL do circuito sequencial
 - Uso de comandos sequenciais (processos, descrição comportamental);
 - 3 componentes principais:
 - Memória de estado
 - Lógica de próximo estado
 - Lógica de saída

PCS3225 - 2015

19

Máquinas de Estados em VHDL

- Estrutura da máquina de estados

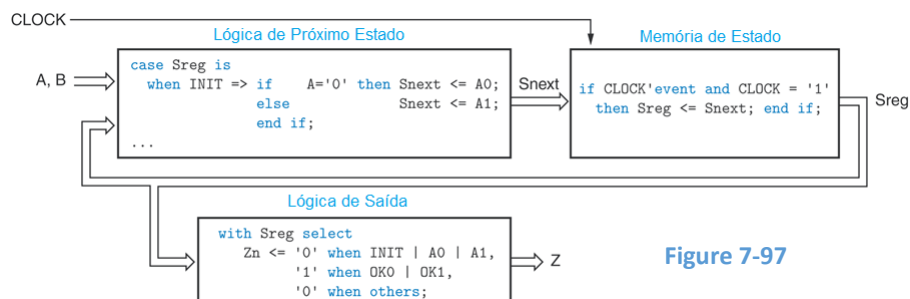


Figure 7-97

PCS3225 - 2015

20

Máquinas de Estados em VHDL

- Descrição VHDL da máquina de estados

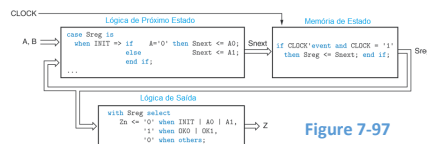


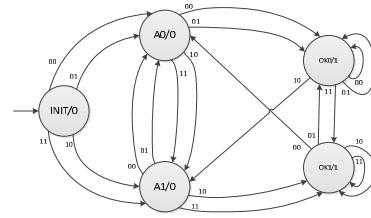
Figure 7-97

Código na pág.631

```

1 library IEEE;
2 use IEEE.std_logic_1164.all;
3
4 entity smexamp is
5   port ( CLOCK, A, B: in STD_LOGIC;
6         Z: out STD_LOGIC );
7 end;
8
9 architecture smexamp_arch of smexamp is
10  type State_type is (INIT, A0, A1, OK0, OK1);
11  signal Sreg, Snext: State_type; -- current state and next state
12 begin
13
14  process (CLOCK) -- state memory
15  begin
16    if CLOCK'event and CLOCK = '1' then
17      Sreg <= Snext;
18    end if;
19  end process;
20

```



```

21 process (A, B, Sreg) -- next-state logic
22 begin
23   case Sreg is
24     when INIT => if A = '0' then Snext <= A0;
25                  else Snext <= A1;
26                   end if;
27     when A0 => if A = '0' then Snext <= OK0;
28                else Snext <= A1;
29                end if;
30     when A1 => if A = '0' then Snext <= A0;
31                else Snext <= OK1;
32                end if;
33     when OK0 => if A = '0' then Snext <= OK0;
34                  elsif A = '1' and B = '0' then Snext <= A1;
35                  else Snext <= OK1;
36                  end if;
37     when OK1 => if A = '0' and B = '0' then Snext <= A0;
38                  elsif A = '1' and B = '1' then Snext <= OK1;
39                  else Snext <= OK0;
40                  end if;
41     when others => Snext <= INIT;
42   end case;
43 end process;
44
45 with Sreg select -- output logic (based on state only)
46   Z <= '0' when INIT | A0 | A1,
47        '1' when OK0 | OK1,
48        '0' when others;
49 end smexamp_arch;

```

21

Máquinas de Estados em VHDL

- Descrição VHDL da máquina de estados

```

1 library IEEE;
2 use IEEE.std_logic_1164.all;
3
4 entity smexamp is
5   port ( CLOCK, A, B: in STD_LOGIC;
6         Z: out STD_LOGIC );
7 end;
8
9 architecture smexamp_arch of smexamp is
10  type State_type is (INIT, A0, A1, OK0, OK1);
11  signal Sreg, Snext: State_type; -- current state and next state
12 begin
13
14  process (CLOCK) -- state memory
15  begin
16    if CLOCK'event and CLOCK = '1' then
17      Sreg <= Snext;
18    end if;
19  end process;
20

```

Bloco dos Registradores

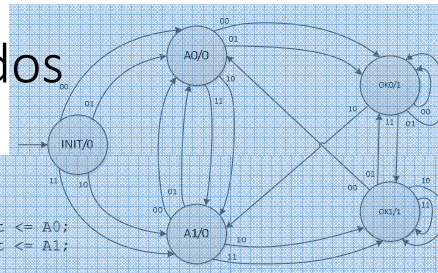
22

Máquinas de Estados em VHDL

```

21 process (A, B, Sreg) -- next-state logic
22 begin
23     case Sreg is
24         when INIT => if A = '0' then Snext <= A0;
25                       else Snext <= A1;
26                       end if;
27         when A0 => if A = '0' then Snext <= OK0;
28                    else Snext <= A1;
29                    end if;
30         when A1 => if A = '0' then Snext <= A0;
31                    else Snext <= OK1;
32                    end if;
33         when OK0 => if A = '0' then Snext <= OK0;
34                      elsif A = '1' and B = '0' then Snext <= A1;
35                      else Snext <= OK1;
36                      end if;
37         when OK1 => if A = '0' and B = '0' then Snext <= A0;
38                      elsif A = '1' and B = '1' then Snext <= OK1;
39                      else Snext <= OK0;
40                      end if;
41         when others => Snext <= INIT;
42     end case;
43 end process;
44
45 with Sreg select -- output logic (based on state only)
46     Z <= '0' when INIT | A0 | A1,
47           '1' when OK0 | OK1,
48           '0' when others;
49 end smexamp_arch;

```



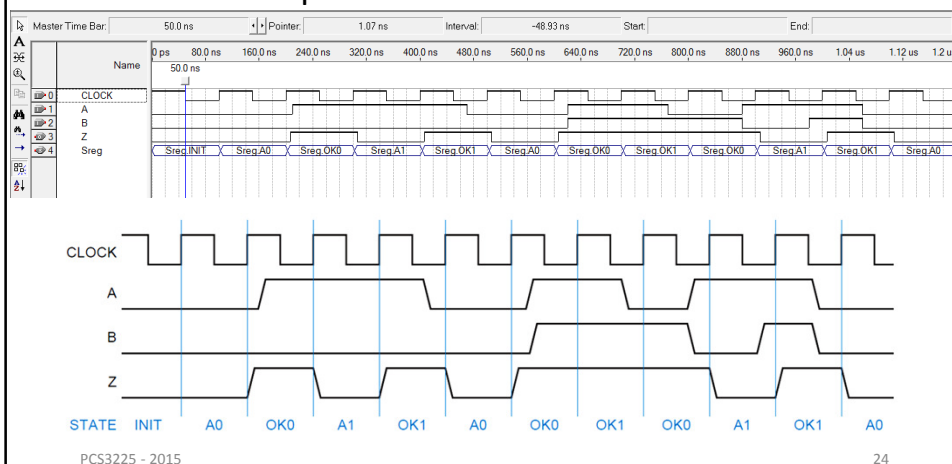
Bloco Combinatório
Próximo Estado

Bloco Combinatório
Valor de saída

23

Máquinas de Estados em VHDL

• Carta de tempos



PCS3225 - 2015

24

Máquinas de Estados em VHDL

- Reset síncrono

```
entity smexamprs is
  port ( CLOCK, RESET, A, B: in STD_LOGIC;
        Z: out STD_LOGIC );
end;
...
process (CLOCK) -- state memory with synchronous reset
begin
  if CLOCK'event and CLOCK = '1' then
    if RESET = '1' then Sreg <= INIT;
    else
      Sreg <= Snext; end if;
    end if;
  end process;
```

Table 7-39

Adding synchronous reset capability to a VHDL state machine.

Máquinas de Estados em VHDL

- Reset assíncrono

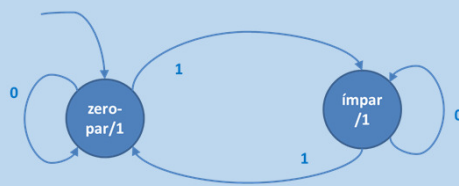
```
process (CLOCK, RESET) -- state memory with asynchronous reset
begin
  if RESET = '1' then Sreg <= INIT;
  elsif CLOCK'event and CLOCK = '1' then Sreg <= Snext; end if;
end process;
```

Table 7-40

Adding asynchronous reset capability to a VHDL state machine.

Máquinas de Estados em VHDL

- Exercício 1 – Escrever uma descrição VHDL do circuito de detecção da ocorrência de um número par de bits 1 na entrada (máquina de Moore).

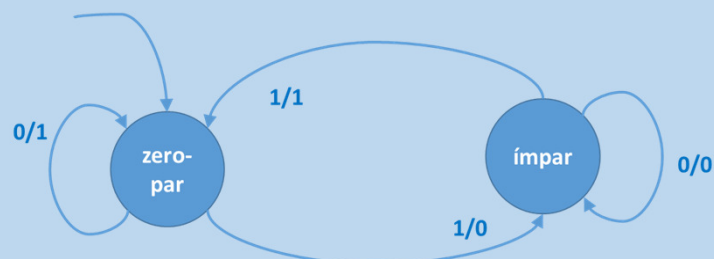


PCS3225 - 2015

27

Máquinas de Estados em VHDL

- Exercício 2 – Escrever uma descrição VHDL do circuito de detecção da ocorrência de um número par de bits 1 na entrada (máquina de Mealy).

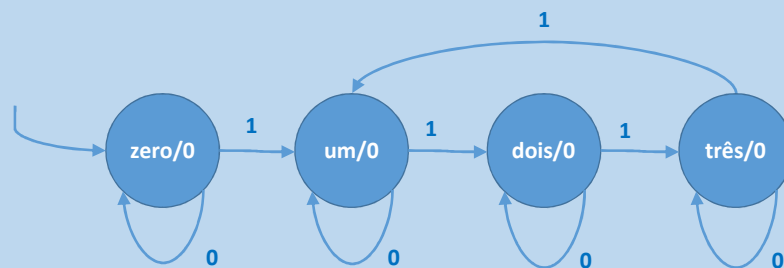


PCS3225 - 2015

28

Máquinas de Estados em VHDL

- Exercício 3 – Escrever uma descrição VHDL do circuito de detecção da ocorrência de um número múltiplo de três de bits 1 na entrada (máquina de Moore).

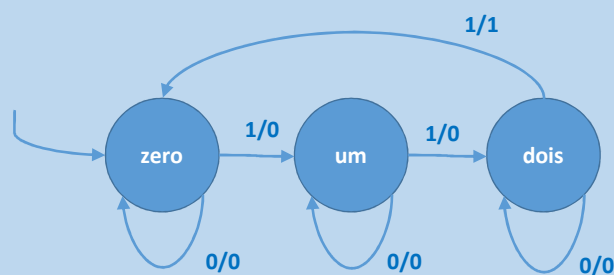


PCS3225 - 2015

29

Máquinas de Estados em VHDL

- Exercício 4 – Escrever uma descrição VHDL do circuito de detecção da ocorrência de um número múltiplo de três de bits 1 na entrada (máquina de Mealy).

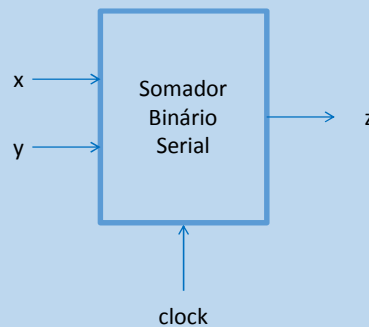


PCS3225 - 2015

30

Máquinas de Estados em VHDL

- Exercício 5 – Escrever uma descrição VHDL do circuito sequencial de soma serial.



PCS3225 - 2015

31

No próximo módulo

- *Máquina de estados: Síntese de FSM*
- Referência: **Wakerly**, pág. 553-570.
- Preparação para a próxima aula:
 - Responda a questão:
 “[quantos e quais são os passos do método de síntese apresentado no livro-texto?](#)”

PCS3225 - 2015

32