

GRADUAÇÃO PRESENCIAL
1º semestre- 2015

Sistemas Digitais II
Engª Elétrica – 6º/7º semestres

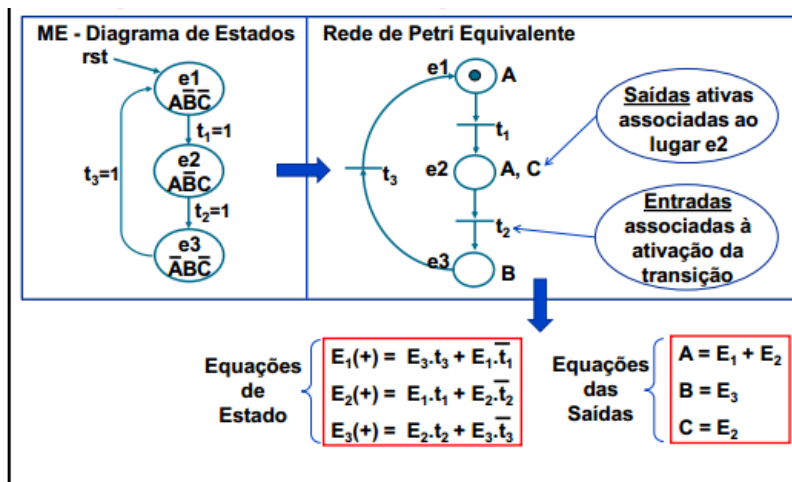
Profº. Ms. Cristiano Malheiro

cmalheiro@anhanguera.com

<http://cristianotm.wix.com/notasdeaula>

1

Aula 12



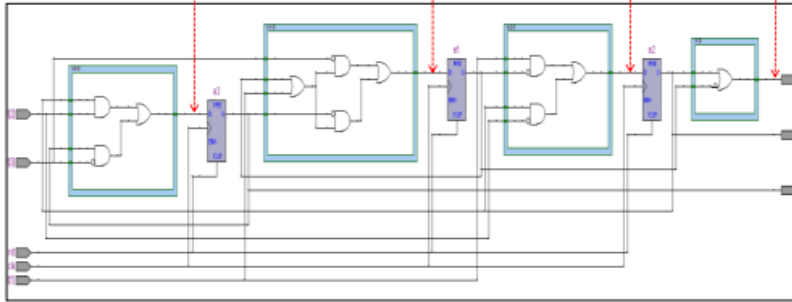
2

Aula 12

Exemplo 7 – Síntese de Máquina de Estados

Implementação física (com portas lógicas e registradores D)

$$E_3(+)=E_2.t_2+E_3.\bar{t}_3 \quad E_1(+)=E_3.t_3+E_1.\bar{t}_1 \quad E_2(+)=E_1.t_1+E_2.\bar{t}_2 \quad A=E_1+E_2$$



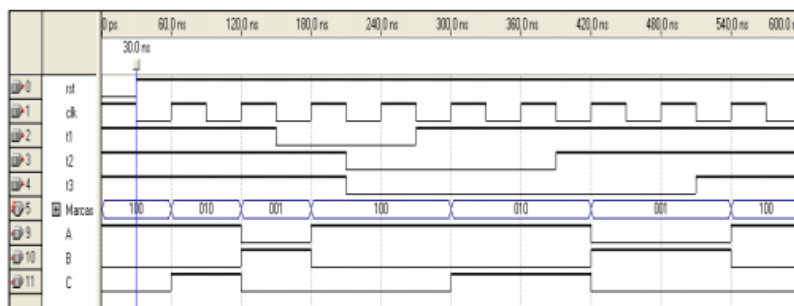
3

kroton
pensando por educar

Aula 12

Exemplo 7 – Síntese de Máquina de Estados

A simulação das formas de onda permite visualizar diretamente a evolução da marcação (basta inserir as saídas dos registradores nas formas de onda)



4

kroton
pensando por educar

Aula 12

Descrição e Síntese de Máquinas de Estados

Interpretação da representação de Petri para Unidades de Controle e ME:

- O **comportamento da rede**, em cada instante, é **caracterizado pela marcação** da rede naquele momento;
- A **evolução da rede depende do disparo de transições** e que para isto algumas **condições devem ser satisfeitas**;
- Como **consequência** do disparo de transições chega-se a uma **nova marcação** da rede;
- A **marcação** é definida pelo **número de marcas em cada lugar** da rede naquele instante.

Diferença importante entre Máquina de Estados e Redes de Petri Seguras

- **ME:** comportamento da rede é caracterizado pelas **transições dos estados** (**apenas um lugar** tem marca em um dado instante);
- **Redes de Petri Seguras:** comportamento da rede é caracterizado pela **evolução das marcações da rede** (**vários lugares** podem ter marcas em um dado instante).

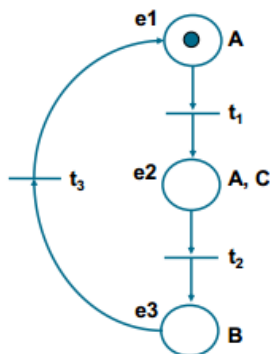
5

kroton
pensando por educar

Aula 12

Síntese de Redes Petri em VHDL

As redes de Petri seguras (como as que representam uma máquina de estados) podem ser descritas diretamente em VHDL, sem a elaboração das equações de estado, utilizando a descrição arquitetura no formato comportamental.



A descrição comportamental de cada lugar do sistema deve ser realizada com base nas seguintes situações:



- Condição para cada lugar ter marca
- Condição para cada lugar não ter marca

6

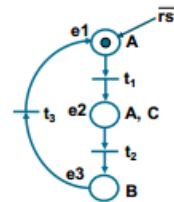
kroton
pensando por educar

Aula 12

Síntese de Redes Petri em VHDL

O primeiro passo é identificar o sistema (**ENTITY**), bem como suas entradas (**IN**) e saídas (**OUT**).

```
ENTITY Petri IS
  PORT(
    clk, rst, t1, t2, t3 : IN std_logic;
    A, B, C              : OUT std_logic;
  );
END Petri;
```



7

kroton
passão por educar

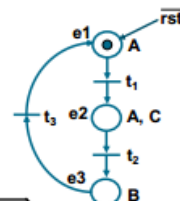
Aula 12

Síntese de Redes Petri em VHDL

Criar uma variável interna na arquitetura (**SIGNAL**) para cada lugar, para armazenar o estado do processo (marcas). Para redes seguras devem ser do tipo **std_logic** (valem '0' ou '1'). Para redes não seguras podem ser do tipo **integer** (lugares com várias marcas).

Como o sistema é síncrono é necessário especificar que as alterações dos sinais devem ocorrer apenas na borda do clock.

```
ARCHITECTURE basica OF Petri IS
  SIGNAL e1, e2, e3 : std_logic;
BEGIN
  PROCESS (clk, rst)
  BEGIN
    IF rst='0' THEN e1<='1'; e2<='0'; e3<='0';
    ELSIF clk'EVENT AND clk = '1' THEN
      -- Descrição comportamental do circuito
    END IF;
  END PROCESS;
END basica;
```



borda de subida de clk

8

kroton
passão por educar

Aula 12

Síntese de Redes Petri em VHDL

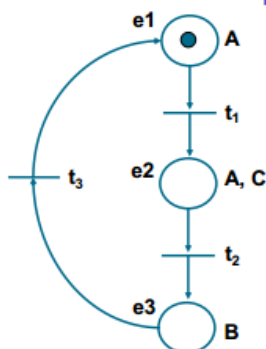
- A **descrição comportamental** é realizada utilizando-se a primitiva **PROCESS** (processo), definindo a marcação da rede em cada instante;
- Através da estrutura condicional **IF... THEN.... ELSIF.... ELSE**, **juntamente com operadores lógicos AND, OR, NOT, NAND, NOR, XOR e XNOR**, **definem-se as condições e valores para cada lugar da rede**;
- Utilizar apenas uma estrutura **IF... THEN... ELSIF.... ELSE** para a definição da marcação de cada lugar da rede (isso garante que a definição da marcação de cada lugar é realizada em um único ponto do processo);
- Lembrar que a estrutura **IF... THEN.... ELSIF.... ELSE** é **sequencial** e que portanto uma segunda condição só será testada **se a primeira for falsa**. E que o processo é **síncrono**, o que significa que a **atribuição de valores** para todos os lugares deve ocorrer **simultaneamente**;
- As **equações de saída** devem ficar fora do processo. Dessa forma a **atribuição de valores** aos lugares será realizada **após a finalização do processo**, quando a nova marcação está completamente definida.

9

kroton
ponto por educar

Aula 12

Exemplo 8



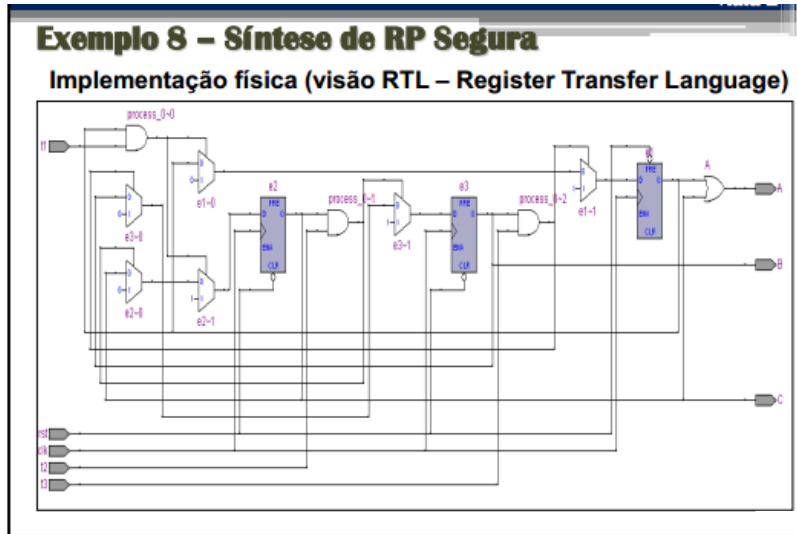
```

ENTITY exemplo8 IS
    PORT( clk, rst, t1, t2, t3 : IN STD_LOGIC;
          A, B, C : OUT STD_LOGIC);
END exemplo8;
ARCHITECTURE rede_petri OF exemplo8 IS
    SIGNAL e1, e2, e3: STD_LOGIC;
BEGIN
    PROCESS (clk,rst)
    BEGIN
        IF rst = '0' THEN e1<='1'; e2<='0'; e3<='0';
        ELSIF (clk'EVENT AND clk = '1') THEN
            If (e3 = '1' and t3='1') then e1<='1' ;
            elsif (e1 = '1' and t1 = '1') then e1<='0';
            else e1<=e1;
            end if;
            If (e1 = '1' and t1='1') then e2<='1' ;
            elsif (e2 = '1' and t2 = '1') then e2<='0';
            else e2<=e2;
            end if;
            If (e2 = '1' and t2='1') then e3<='1';
            elsif (e3 = '1' and t3 = '1') then e3<='0';
            else e3 <=e3;
            end if;
        END IF;
        Define atribuição das saídas
        A<=(e1 or e2); B<=e3; C<=e2;
    END PROCESS;
END rede_petri;
  
```

10

kroton
ponto por educar

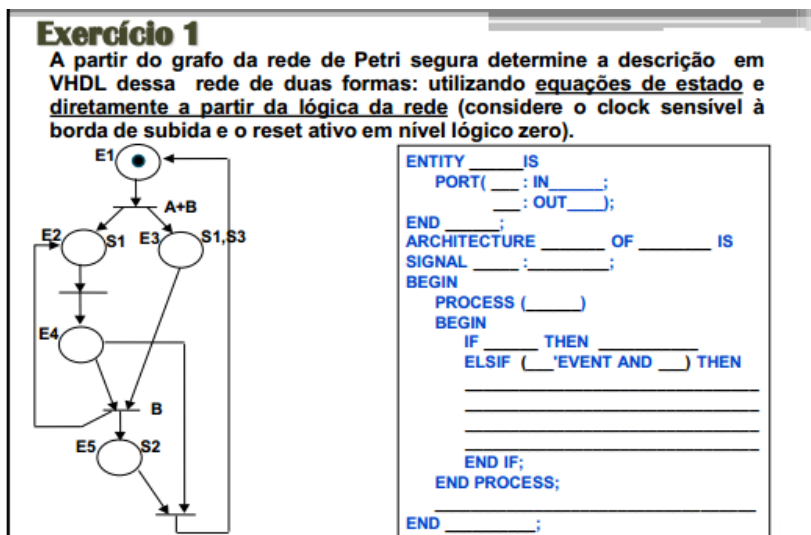
Aula 12



11

kroton
 paixão por educar

Aula 12



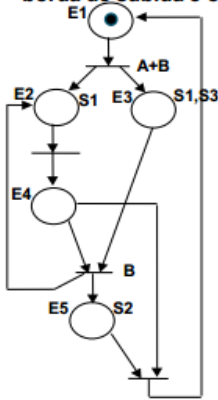
12

kroton
 paixão por educar

Aula 12

Exercício 1

A partir do grafo da rede de Petri segura determine a descrição em VHDL dessa rede de duas formas: utilizando equações de estado e diretamente a partir da lógica da rede (considere o clock sensível à borda de subida e o reset ativo em nível lógico zero).



Equações de Estado:

$E1(+)$ =
 $E2(+)$ =
 $E3(+)$ =
 $E4(+)$ =
 $E5(+)$ =

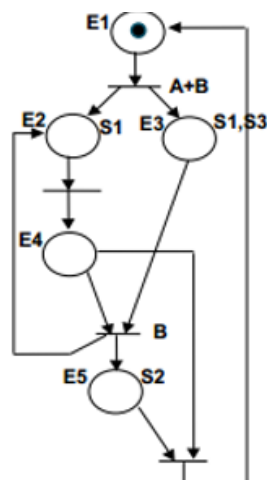
Equações das Saídas:

$S1$ =
 $S2$ =
 $S3$ =

13

kroton
pensando por educar

Aula 12



Equações de Estado:

$E1(+)$ = $E4.E5 + E1./(A+B)$
 $E2(+)$ = $E1.(A+B) + E3.E4.B$
 $E3(+)$ = $E1.(A+B) + E3./(B.E4)$
 $E4(+)$ = $E2 + E4./(B.E3 + E5) = E2 + E4./(B.E3)./E5$
 $E5(+)$ = $E3.E4.B + E5./E4$

Equações das Saídas:

$S1 = E2 + E3$
 $S2 = E5$
 $S3 = E3$

14

kroton
pensando por educar

Aula 12

Descrição VHDL utilizando equações de estado:

Equações de Estado:

$E1(+)=E4.E5 + E1/(A+B)$
 $E2(+)=E1.(A+B) + E3.E4.B$
 $E3(+)=E1.(A+B) + E3./(B.E4)$
 $E4(+)=E2 + E4./(B.E3)/E5$
 $E5(+)=E3.E4.B + E5/E4$

Equações das Saídas:

$S1 = E2 + E3$
 $S2 = E5$
 $S3 = E3$

```

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY Aula2_Exercicio1A IS
  PORT ( clk, rst, A, B : IN STD_LOGIC;
        S1, S2, S3 : OUT STD_LOGIC );
END Aula2_Exercicio1A;

ARCHITECTURE BEHAVIOR OF Aula2_Exercicio1A IS
  SIGNAL E1, E2, E3, E4, E5 : STD_LOGIC;
BEGIN
  PROCESS (clk,rst)
  BEGIN
    IF (rst='0') THEN E1<='1'; E2<='0'; E3<='0'; E4<='0'; E5<='0';
    ELSIF (clk='1' AND clk'event) THEN
      E1<= (E4 AND E5) OR (E1 AND NOT (A OR B));
      E2<= (E1 AND (A OR B)) OR (E3 AND E4 AND B);
      E3<= (E1 AND (A OR B)) OR (E3 AND NOT (B AND E4));
      E4<= (E2) OR (E4 AND (NOT (B AND E3)) AND (NOT (E5)));
      E5<= (E3 AND E4 AND B) OR (E5 AND (NOT (E4)));
    END IF;
  END PROCESS;
  S1<= (E2 OR E3); S2<=E5; S3<=E3;
END BEHAVIOR;

```

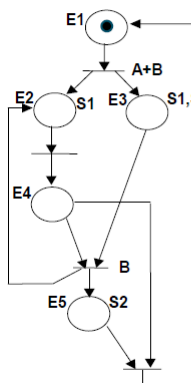
Em atribuições assertivas
podem ser utilizadas as
expressões: (E4 and E5)
(E1 or A)
(não permitidas no caso
do comando IF)

15

kroton
patroão por educar

Aula 12

Descrição VHDL
a partir da RP:



```

ENTITY Aula2_Exercicio1 IS
  PORT ( clk, rst, A, B : IN STD_LOGIC;
        S1, S2, S3 : OUT STD_LOGIC );
END Aula2_Exercicio1;

ARCHITECTURE BEHAVIOR OF Aula2_Exercicio1 IS
  SIGNAL E1, E2, E3, E4, E5 : STD_LOGIC;
BEGIN
  PROCESS (clk,rst)
  BEGIN
    IF (rst='0') THEN E1<='1'; E2<='0'; E3<='0'; E4<='0'; E5<='0';
    ELSIF (clk='1' AND clk'event) THEN
      IF (E1='1') AND (A='1' OR B='1') THEN E1<='0';
      ELSIF (E5='1' AND E4='1') THEN E1<='1'; END IF;
      IF (E1='1' AND (A='1' OR B='1')) OR (E4='1' AND E3='1' AND B='1') THEN E2<='1';
      ELSIF (E2='1') THEN E2<='0'; END IF;
      IF (E1='1' AND (A='1' OR B='1')) THEN E3<='1';
      ELSIF (E3='1' AND E4='1' AND B='1') THEN E3<='0'; END IF;
      IF E2='1' THEN E4<='1';
      ELSIF (E4='1' AND ((E3='1' AND B='1') OR (E5='1'))) THEN E4<='0'; END IF;
      IF (E4='1' AND E3='1' AND B='1') THEN E5<='1';
      ELSIF (E5='1' AND E4='1') THEN E5<='0'; END IF;
    END IF;
  END PROCESS;
  S1<= (E2 OR E3); S2<=E5; S3<=E3;
END BEHAVIOR;

```

Atenção:
Na condição do IF não
pode ser utilizada a
expressão: (A or B)
(para isso A e B
deveriam ser
BOOLEAN)

Atenção:
Na assertiva de saída
pode ser utilizada a
expressão: (E2 or E3)
(nesse caso não é uma
comparação)

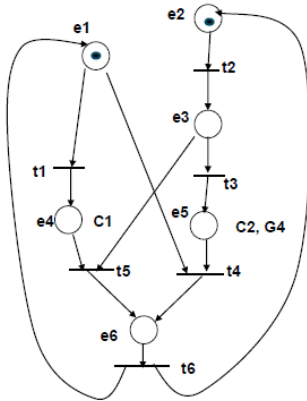
16

kroton
patroão por educar

Aula 12

Exercício 2

A partir do grafo da rede de Petri segura fornecido determine as equações de estado e das saídas. Realize a descrição em VHDL dessa rede diretamente a partir do grafo da rede, considerando o clock sensível à borda de descida e o reset ativo em nível lógico um.



Equações de Estado:

E1(+) =
E2(+) =
E3(+) =
E4(+) =
E5(+) =
E6(+) =

Equações das Saídas:

C1 =
C2 =
G4 =

17



Aula 12

Exercício 2

Equações de Estado:

$E1(+) = E6.t6 + E1./(t1 + E5.t4)$
 $E2(+) = E6.t6 + E2.t2$
 $E3(+) = E2.t2 + E3./(t3 + E4.t5)$
 $E4(+) = E1.t1 + E4./(E3.t5)$
 $E5(+) = E3.t3 + E5./(E1.t4)$
 $E6(+) = E3.E4.t5 + E1.E5.t4 + E6./t6$

Equações das Saídas:

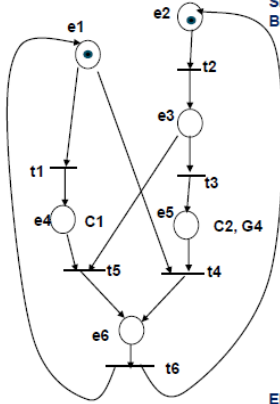
C1 = E4
 C2 = E5
 G4 = E5

18



Aula 12

Exercício 2



```

ENTITY exercicio2 IS
  PORT( clk, rst, t1, t2, t3, t4, t5, t6: IN STD_LOGIC;
        C1, C2, G4 : OUT STD_LOGIC);
END exercicio2;
ARCHITECTURE rede OF exercicio2 IS
  SIGNAL e1, e2, e3, e4, e5, e6 : STD_LOGIC;
BEGIN
  PROCESS ( clk, rst)
  BEGIN
    IF rst='1' THEN e1<='1'; e2<='1'; e3<='0'; e4<='0'; e5<='0'; e6<='0';
    ELSIF clk'event and clk='0' THEN
      If e6='1' and t6='1' then e1<='1';
      elsif e1='1' and (t1='1' or (e5='1' and t4='1')) then e1<='0'; end if;
      If e6='1' and t6='1' then e2<='1';
      elsif e2='1' and t2='1' then e2<='0'; end if;
      if e2='1' and t2='1' then e3<='1';
      elsif e3='1' and (t3='1' or (e4='1' and t5='1')) then e3<='0'; end if;
      if e1='1' and t1='1' then e4<='1';
      elsif e4='1' and e3='1' and t5='1' then e4<='0'; end if;
      if e3='1' and t3='1' then e5<='1';
      elsif e5='1' and e1='1' and t4='1' then e5<='0'; end if;
      if (e4='1' and e3='1' and t5='1') or (e5='1' and e1='1' and t4='1') then e6<='1';
      elsif e6='1' and t6='1' then e6<='0'; end if;
    END IF;
  END PROCESS;
  C1<=e4; C2<=e5; G4<=e5;
END rede;

```

19



kroton
paixão por educar

Bibliografia desta aula:

-www.altera.com.br

(manual do PLD D2-115)

- Sistemas Digitais: Princípios e Aplicações – TOCCI, 11ª edição, 2011.

- - Redes de Petri:

- <http://www.dca.fee.unicamp.br/~rafael/ia851/PETRI.pdf>

20

