ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES I

Descrição de Atrasos

prof. Dr. César Augusto M. Marcon
prof. Dr. Edson Ifarraguirre Moreno

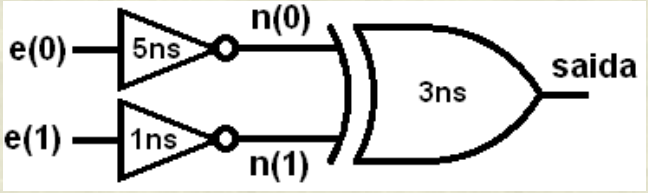
Introdução

- **O conhecimento do tempo de propagação do sinal é uma parte muito importante no projeto de um sistema digital**
  - Este tempo permite estimar a velocidade de resposta de um circuito e conseqüentemente os caminhos críticos

- **VHDL pode auxiliar na avaliação de tempos através dos comandos que informam atrasos, tais como o comando after**
  - Ex.: Descrição de um inversor com 7 ns de atraso

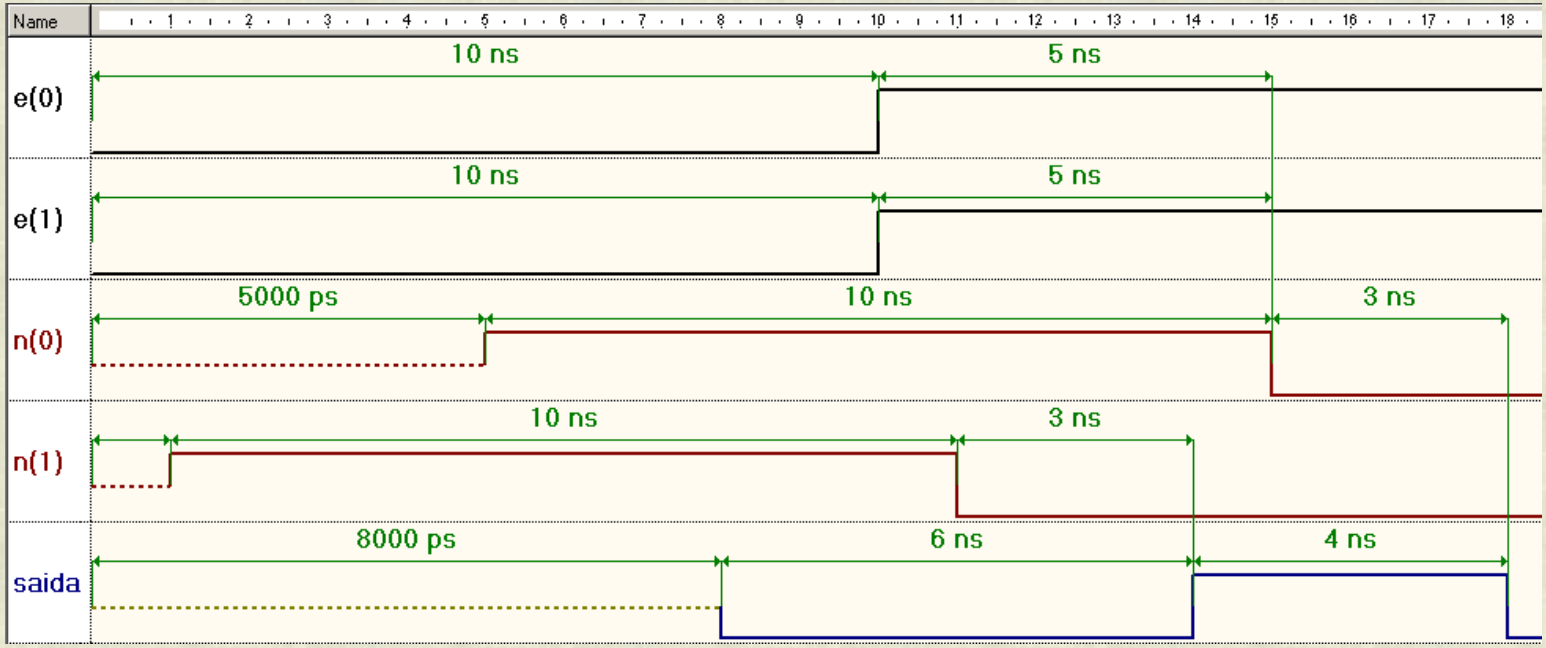
```
x<= not y after 7ns
```

Exemplo




```
entity Atrasos is
      port(e: in std_logic_vector(1 downto 0);
saida: out std_logic);
end Atrasos;
architecture Atrasos of Atrasos is
      signal n: std_logic_vector(1 downto 0);
begin
      n(0) <= not e(0) after 5ns;
      n(1) <= not e(1) after 1ns;
      saida <= n(1) xor n(0) after 3ns;
end Atrasos;
```

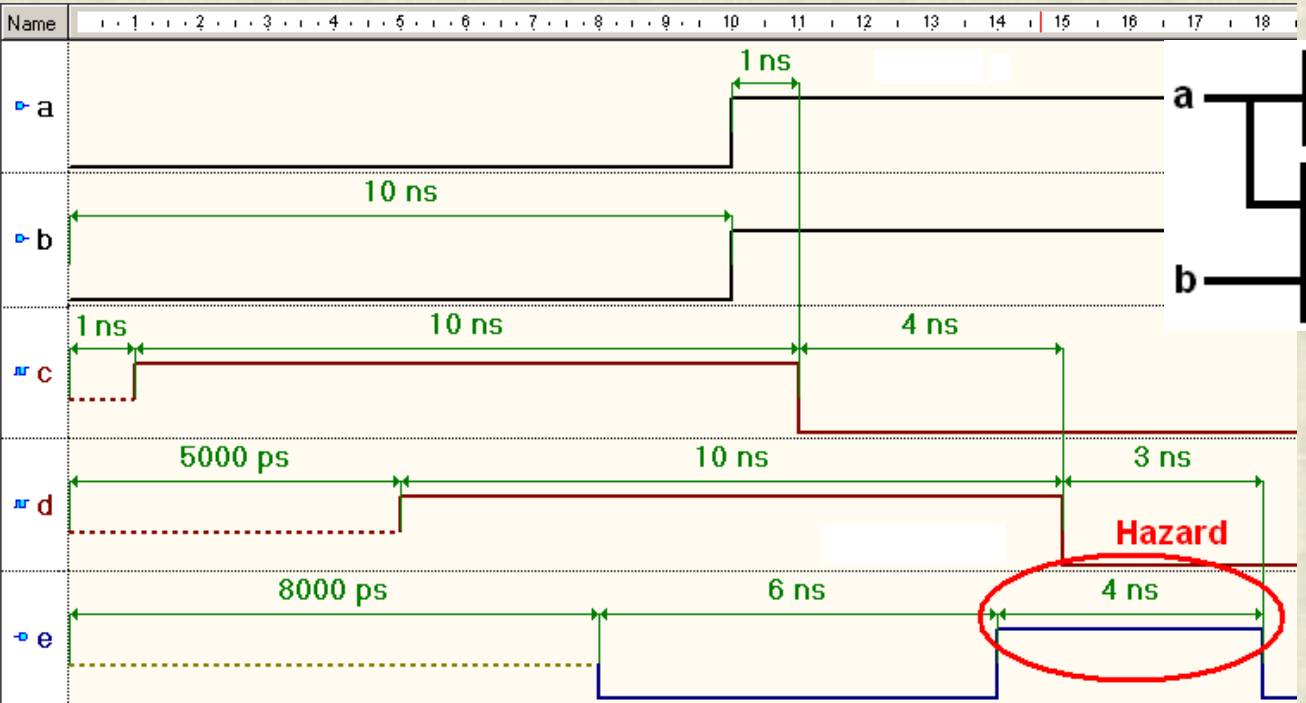
```
entity TB is
end TB ;
architecture TB of Tb is
      signal e: std_logic_vector(1 downto 0);
      signal s: std_logic;
begin
      e(0) <= '0', '1' after 10ns;
      e(1) <= '0', '1' after 10ns;
      UUT: entity Atrasos port map(e=>e, saida=>s);
end TB ;
```

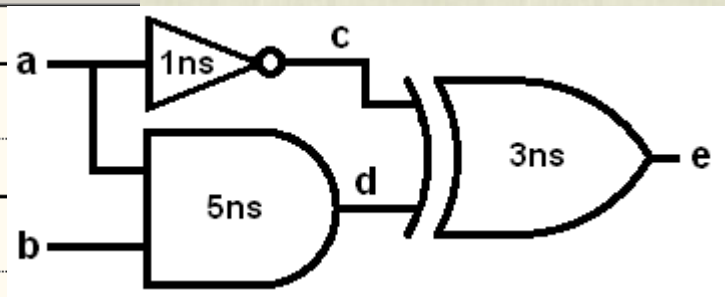


Análise de Hazards

```
entity Atrasos is
      port(a, b: in std_logic;
            e: out std_logic);
end Atrasos;
architecture Atrasos of Atrasos is
      signal c, d: std_logic;
begin
      c <= not a after 1ns;
      d <= not (a and b) after 5ns;
      e <= c xor d after 3ns;
end Atrasos;
```

```
entity tb_Atrasos is
end tb_Atrasos;
architecture tb_Atrasos of tb_Atrasos is
      signal a, b, e: std_logic;
begin
      a <= '0', '1' after 10ns;
      b <= '0', '1' after 10ns;
      UUT: entity Atrasos port map (a => a, b => b, e => e);
end tb_Atrasos;
```





Exercícios

1. Faça o projeto do circuito combinacional Inv8Bits. Este tem como entrada 8 bits, e como saída os 8 bits de entrada invertidos
   - A inversão ocorre de forma que o oitavo bit seja o primeiro, o sétimo o segundo e assim sucessivamente
   - Considere que o atraso da inversão de bits leva 1 ns
2. Faça o projeto do circuito Arit8Bits com as funcionalidades descritas abaixo:
   a. Soma duas entradas de 8 bits. O resultado é gerado em uma saída de 8 bits após 2 ns;
   b. Soma duas entradas de 8 bits invertidas e o resultado é gerado em uma saída de 8 bits após 2 ns;
   c. Soma duas entrada de 8 bits, uma invertida com outra não invertida e o resultado é gerado em uma saída de 8 bits após 1 ns;
   d. Rotaciona uma entrada 2 bits para a direita e o resultado é gerado na saída após 3 ns

Exercícios

3. Faça o projeto do circuito AritComp8Bits. Este é praticamente igual ao circuito Arit8Bits, porém contem inversores Inv8Bits (sempre que necessário)
   - Faça este circuito com port maps do circuito Inv8Bits
4. Faça um test-bench, que estimule as entradas do circuitos AritComp8Bits, de forma que todas as operações deste circuito sejam geradas
   - Qual é o menor intervalo de tempo que as entradas podem variar, se desejamos que o circuito esteja estável (sem variações)?

Exercícios

5. Dado a descrição abaixo, verifique se existe a possibilidade de ocorrer algum hazard e calcule qual é o caminho crítico para o sinal

```
entity Atrasos is
  port
  (e: in std_logic_vector(3 downto 0);
   saida: out std_logic_vector(3 downto 0));
end Atrasos;

architecture Atrasos of Atrasos is
  signal n: std_logic_vector(4 downto 0);
begin
  n(0) <= not (e(0) and e(1)) after 5ns;
  n(1) <= not e(1) after 1ns;
  n(2) <= n(1) or (not e(2)) after 3ns;
  n(3) <= e(0) xor e(2) after 4ns;
  n(4) <= n(0) and n(2) and n(3) after 2ns;
  saida(0) <= n(1) xor n(0) after 3ns;
  saida(1) <= not (n(2) and n(3)) after 4ns;
  saida(2) <= n(4) or n(3) or e(3) after 3ns;
  saida(3) <= (n(0) xor n(1)) or (e(2) xor e(0)) after 7ns;
end Atrasos;
```