

Aluno: .....

Professor: .....

1. **[Máquina de Moore]** Construa um circuito sequencial síncrono que mostre como saída, a cada borda de relógio, os valores da sequência 1 – 3 – 5 – 7, de forma crescente, repetidamente, enquanto a entrada  $X$  for 0. Caso a entrada  $X$  seja 1, os mesmos valores são mostrados, sucessivamente, mas de forma decrescente. Apresente a máquina de estados, a tabela de estados e as equações de entrada simplificadas dos flip-flops e da saída.
2. **[Máquina de Mealy]** Apresente a máquina de estados de um circuito sequencial síncrono que realiza a função de uma chave lógica que aceita as sequências 001 ou 110. A sobreposição é aceita (p.ex. a chave será aberta duas vezes tanto na sequência 00110 quanto na sequência 11001). Não é necessário gerar a tabela de estados e as equações lógicas.
3. **[VHDL]** Apresente o circuito lógico que descrito pelo código VHDL abaixo.

```
library IEEE;
use IEEE.std_logic_1164.all;

entity porta1 is
    port(entrada1: in bit;
          saida1: out bit);
end porta1;

architecture arq1 of porta1 is
begin
    saida1 <= (not entrada1) after 5 ns;
end arq1;
-----

library IEEE;
use IEEE.std_logic_1164.all;

entity porta2 is
    port(entrada1, entrada2: in bit;
          saida1: out bit);
end porta2;

architecture arq2 of porta2 is
begin
    saida1 <= (entrada1 and entrada2) after 10 ns;
end arq2;
-----

library IEEE;
use IEEE.std_logic_1164.all;

entity porta3 is
    port(entrada1, entrada2: in bit;
          saida1: out bit);
```

```

end porta3;

architecture arq3 of porta3 is
begin
    saida1 <= (entrada1 or entrada2) after 10 ns;
end arq3;
-----

library IEEE;
use IEEE.std_logic_1164.all;

entity porta4 is
    port(entrada1, entrada2, entrada3: in bit;
          saida1: out bit);
end porta4;

architecture arq4 of porta4 is
    component comp port(entrada1, entrada2: in bit;
                          saida1: out bit);

    end component;
    for all: comp use entity work.porta2(arq2);
    signal junta: bit;
begin
    g0: comp port map(entrada1, entrada2, junta);
    g1: comp port map(junta, entrada3, saida1);
end arq4;
-----

library IEEE;
use IEEE.std_logic_1164.all;

entity circuito is
    port(A, B, C: in bit;
          Y: out bit);
end circuito;

architecture arqc of circuito is
    component comp1 port(entrada1: in bit;
                          saida1: out bit);

    end component;
    component comp2 port(entrada1, entrada2: in bit;
                          saida1: out bit);

    end component;
    component comp3 port(entrada1, entrada2: in bit;
                          saida1: out bit);

    end component;
    component comp4 port(entrada1, entrada2, entrada3: in bit;
                          saida1: out bit);

    end component;
    for all: comp1 use entity work.porta1(arq1);
    for all: comp2 use entity work.porta2(arq2);
    for all: comp3 use entity work.porta3(arq3);
    for all: comp4 use entity work.porta4(arq4);
    signal junta1, junta2, junta3, junta4: bit;
begin
    g0: comp1 port map(A, junta1);
    g1: comp1 port map(B, junta2);
    g2: comp2 port map(A, C, junta3);
    g3: comp3 port map(junta3, junta4, Y);
    g4: comp4 port map(junta1, junta2, C, junta4);
end arqc;

```

- ```

library IEEE;
use IEEE.std_logic_1164.all;

entity teste is
end teste;

architecture arqt of teste is
    component comp port(A, B, C: in bit;
                        Y: out bit);
    end component;
    for all: comp use entity work.circuito(arqc);
    signal tA, tB, tC, tY: bit;
begin
    g0: comp port map(tA, tB, tC, tY);
    tA <= '0',
        '1' after 100 ns;
    tB <= '1',
        '0' after 50 ns,
        '1' after 100 ns;
    tC <= '0',
        '1' after 50 ns,
        '0' after 100 ns;
end arqt;

```

