Projeto com Circuitos Reconfiguráveis

Aula 8 – Projeto lógico sequencial
Máquinas de Estados Finitos

Prof. Daniel M. Muñoz Arboleda

Novembro de 2012

Revisão Máquinas de Estados

Tipicamente incluem:

- Pelo menos dois processos (um deles controla o *clock*)
- Sentenças IF - THEN – ELSE
- Sentenças CASE
- O usuário define os *tipos* para armazenar o estado atual e o próximo estado

Transições dependem do estado atual e, opcionalmente, das entradas

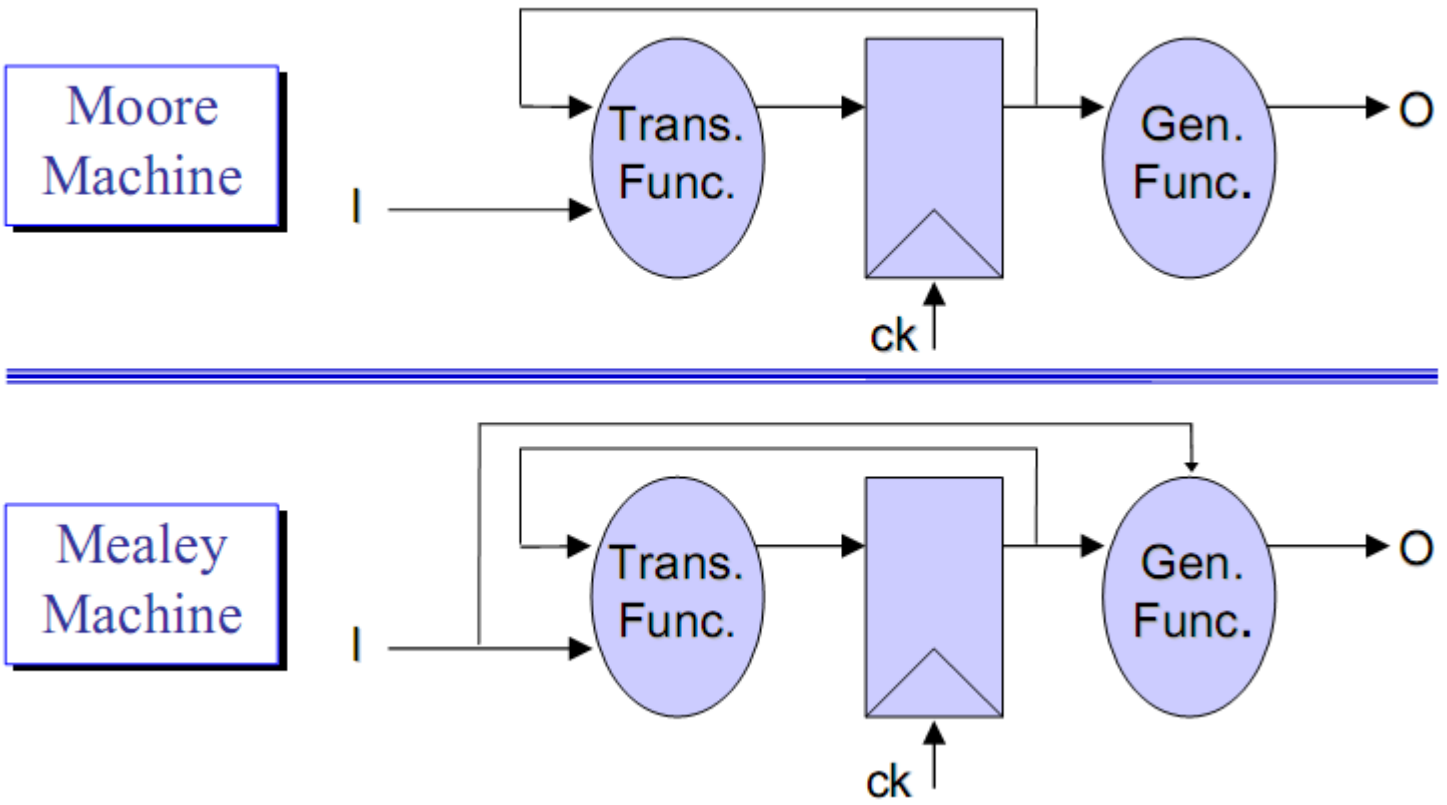
Saídas dependem de:

- Estado atual (máquina de Moore)
- Estado atual e entradas (máquina de Mealy)

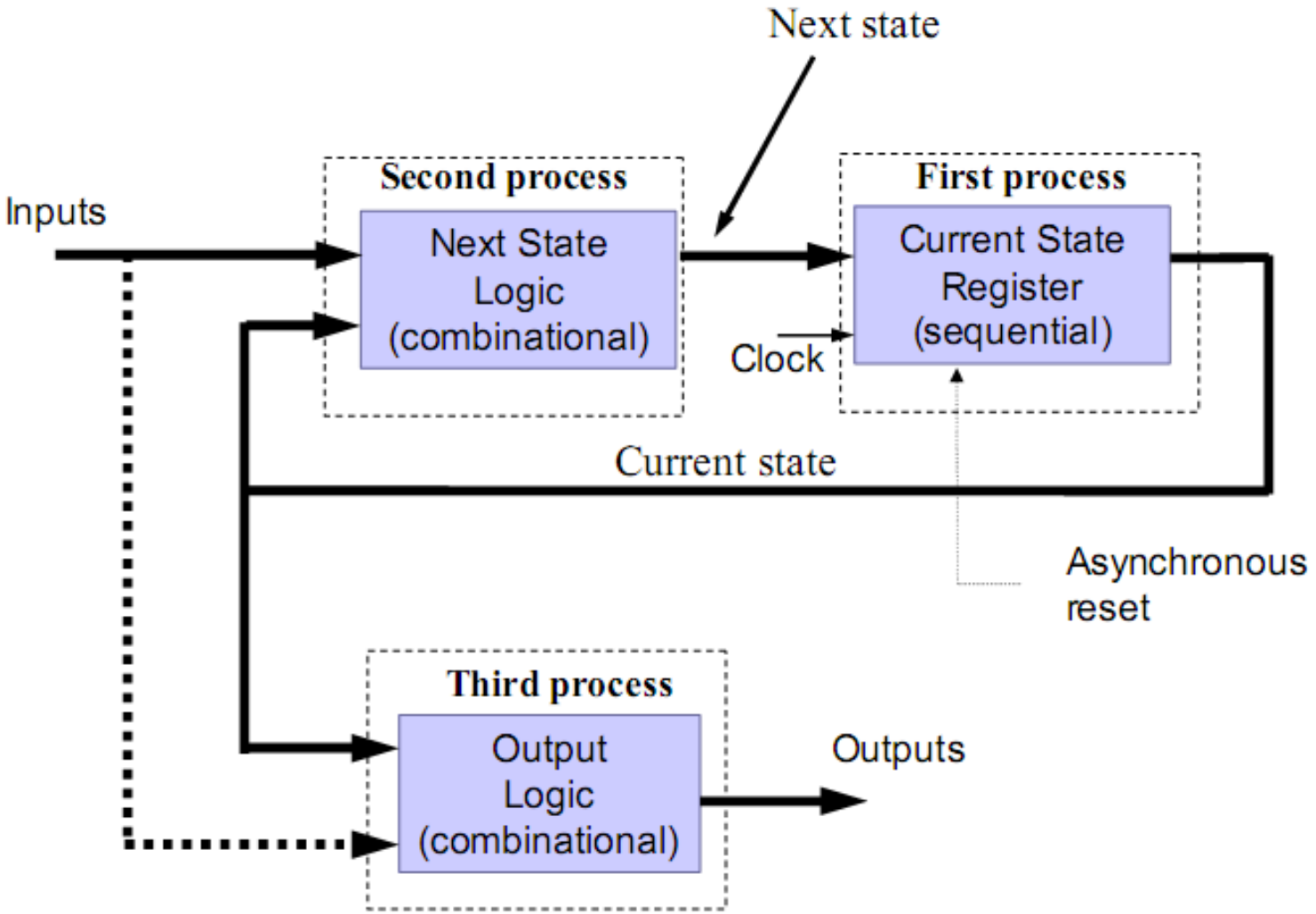
Definição de próximo estado e da saída:

- estado(t+1) <= f(i_1,...,i_n,estado(t))
- Saída <= f(i_1,...,i_n,estado(t))

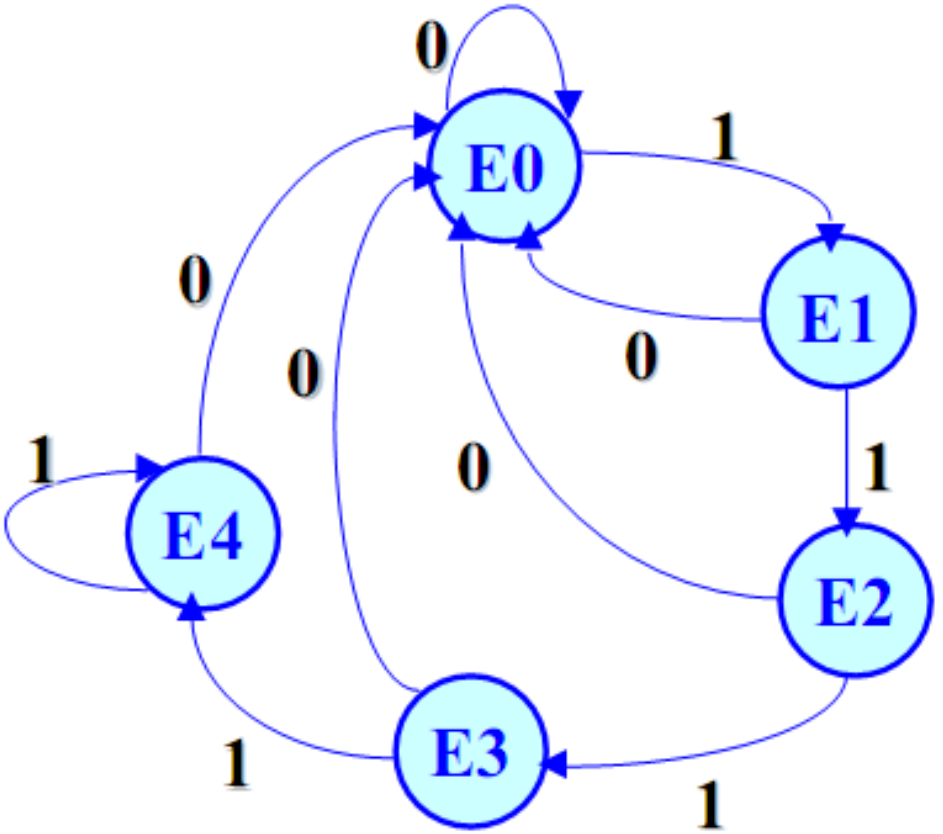
Revisão Máquinas de Estados



FSM. Estrutura típica em VHDL



FSM em VHDL. Contador de '1's



FSM em VHDL. Contador de '1's

```
21  library IEEE;
22  use IEEE.std_logic_1164.all;
23  use IEEE.std_logic_arith.all;
24
25  entity ones_detector is
26  port (
27      clk     : in  std_logic;
28      reset   : in  std_logic;
29      input   : in  std_logic;
30      output  : out std_logic
31  );
32  end ones_detector;
33
34
35  architecture behavioral of ones_detector is
36
37      type   t_state is (waiting,s1,s2,s3,s4);
38      signal state : t_state;
39
40
41  begin
42
```

FSM em VHDL. Contador de '1's (continuação)

```
43  -- FSM
44      process(reset,clk,input)
45      begin
46          if reset='1' then
47              output <= '0';
48              state  <= waiting;
49          elsif rising_edge(clk) then
50              case state is
51                  when waiting =>
52                      output <= '0';
53                      if input = '1' then
54                          state  <= s1;
55                      else state <= waiting;
56                      end if;
57
58                  when s1 =>
59                      output <= '0';
60                      if input = '1' then
61                          state  <= s2;
62                      else state <= waiting;
63                      end if;
64
65                  when s2 =>
66                      output <= '0';
67                      if input = '1' then
68                          state  <= s3;
69                      else state <= waiting;
70                      end if;
```

```
71
72                  when s3 =>
73                      output <= '0';
74                      if input = '1' then
75                          state  <= s4;
76                      else state <= waiting;
77                      end if;
78
79                  when s4 =>
80                      output <= '1';
81                      if input = '0' then
82                          state  <= waiting;
83                      else state <= s4;
84                      end if;
85
86                  when others => state <= waiting;
87              end case;
88          end if;
89      end process;
90
91  end behavioral;
```