

IMPLEMENTANDO MÁQUINAS DE ESTADOS SÍNCRONAS DO TIPO MEALY EM VHDL

1 MÁQUINAS DE ESTADOS SÍNCRONAS DO TIPO MEALY

As máquinas de estados síncronas podem ser classificadas como sendo do tipo Moore (em que as saídas dependem somente do estado atual) ou do tipo Mealy (em que as saídas dependem tanto do estado atual quanto das variáveis de entrada). A Figura 1(a) mostra a topologia genérica de uma máquina do tipo Moore, enquanto que a Figura 1(b) mostra a de uma do tipo Mealy. Note que, nas máquinas do tipo Moore, a saídas mudam de forma síncrona (isto é, sincronizadas com o clock), pois as saídas mudam somente quando o estado atual (que só muda sincronizado com o clock) é alterado; já em uma máquina do tipo Mealy, as saídas podem variar de forma independente do clock, bastando para isso que as entradas sejam alteradas. Na prática, muitas máquinas de estados precisam ser classificadas como máquinas Mealy, pois têm uma ou mais saídas do tipo Mealy (que dependem também das entradas). Entretanto, muitas dessas mesmas máquinas também têm uma ou mais saídas do tipo Moore (que dependem somente do estado atual).

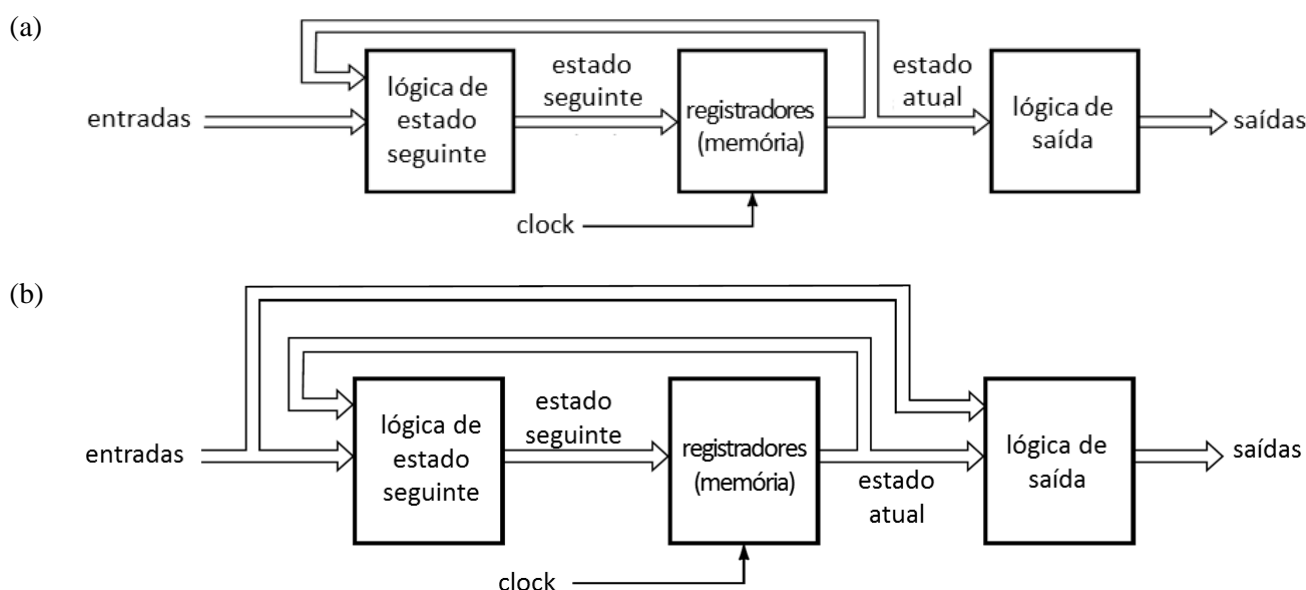


Figura 1 – Topologia genérica das máquinas de estados síncronas do tipo Moore (a) e do tipo Mealy (b).

2 IMPLEMENTANDO MÁQUINAS DE ESTADOS SÍNCRONAS DO TIPO MEALY EM VHDL

Considere a máquina de estados síncrona do tipo Mealy descrita pelo diagrama de estados mostrado na Figura 2. A mesma tem uma entrada (X), uma variável de saída do tipo Moore (Y) e uma variável de saída do tipo Mealy (Z). No diagrama, enquanto os valores da variável Moore Y são mostrados dentro das bolhas de estado (uma vez que dependem somente do estado atual), os valores da variável Mealy Z são mostrados nas setas de transição (pois dependem não só do estado atual, mas também das variáveis de entrada). Enquanto a saída Y só muda de valor caso haja uma mudança de estado (o que só acontece no instante da transição do clock), a saída Z muda de valor assim que a variável de entrada X muda de valor, mesmo que não haja transição do clock. Entretanto, transições de estado também podem fazer com que a saída Z mude de valor, uma vez que Z depende tanto da entrada X quanto do estado atual.

Com base no diagrama de estados, podemos montar a tabela de estados e saídas do circuito (Tabela 1).

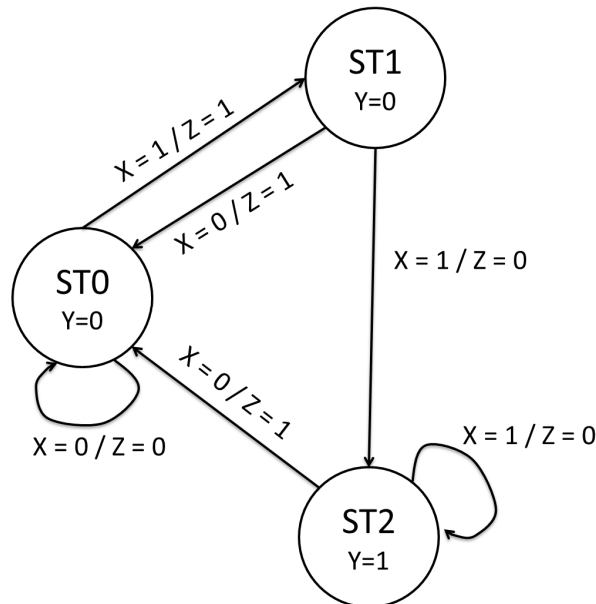


Figura 2 – Diagrama de estados de uma máquina de estados síncrona do tipo Mealy. Note que enquanto os valores da variável de saída do tipo Moore Y são mostrados dentro das bolhas de estado, os valores da variável de saída do tipo Mealy Z são mostrados nas setas de transição, junto com a variável de entrada X .

Tabela 1- Tabela de estados e saídas da máquina Mealy mostrada na Figura2.

estado atual	saída Y	saída Z		estado seguinte	
		se $X = 0$	se $X = 1$	se $X = 0$	se $X = 1$
ST0	0	0	1	ST0	ST1
ST1	0	1	0	ST0	ST2
ST2	1	1	0	ST0	ST2

A implementação de máquinas de estados do tipo Mealy é semelhante à implementação de máquinas do tipo Moore. Segue o modelo RTL (do inglês, *register transfer logic*), com duas áreas bem definidas: lógica combinacional e registradores. Os registradores são responsáveis pela memória do circuito e a lógica combinacional decide a transição da memória e as saídas do circuito.

A máquina de estados descrita pelo diagrama da Figura 2 e pela tabela de estados e saídas apresentada na Tabela 1 é implementada pelo código VHDL da Figura 3. Note que o código é muito parecido com o de uma máquina de estados do tipo Moore. A única diferença se encontra no processo *comb_proc*, que implementa a lógica de estados seguintes e também a lógica de saída. Nessa estrutura, note que os valores que a saída Moore Y recebe (nas linhas 28, 34 e 40) dependem somente do estado atual (*currentState*), enquanto que os valores que a saída Mealy Z recebe (nas linhas 29, 35 e 41) dependem tanto do estado atual (*currentState*) quando da entrada X .

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity MaqEstadosMealy is
5      port( clock : in STD_LOGIC;
6            x : in STD_LOGIC;
7            y : out STD_LOGIC;
8            z : out STD_LOGIC );
9  end MaqEstadosMealy;
10
11 architecture MaqEstadosMealy_arch of MaqEstadosMealy is
12     type estado is (ST0,ST1,ST2);
13     signal currentState, nextState : estado;
14 begin
15
16     sync_proc: process(clock)
17     begin
18         if rising_edge(clock) then
19             currentState <= nextState;
20         end if;
21     end process sync_proc;
22
23     comb_proc: process(currentState,x)
24     begin
25         case currentState is
26
27             when ST0 =>
28                 y <= '0';
29                 if (x = '0') then z <= '0'; nextState <= ST0;
30                 else z <= '1'; nextState <= ST1;
31                 end if;
32
33             when ST1 =>
34                 y <= '0';
35                 if (x = '0') then z <= '1'; nextState <= ST0;
36                 else z <= '0'; nextState <= ST2;
37                 end if;
38
39             when ST2 =>
40                 y <= '1';
41                 if (x = '0') then z <= '1'; nextState <= ST0;
42                 else z <= '0'; nextState <= ST2;
43                 end if;
44
45             when others =>
46                 y <= '0';
47                 if (x = '0') then z <= '0'; nextState <= ST0;
48                 else z <= '1'; nextState <= ST1;
49                 end if;
50
51         end case;
52     end process comb_proc;
53 end MaqEstadosMealy_arch;
```

Figura 3 – Implementação em VHDL, usando o modelo RTL, da máquina de estados síncrona do tipo Mealy descrita pelo diagrama de estados mostrado na Figura 2 e pela tabela de estados e saídas mostrada na Tabela 1.