

PCS 3225
Sistemas Digitais II

Síntese de Circuitos Sequenciais

7-6 – State Machine Synthesis Using Transition Lists
(Wakerly, 4ed, páginas 577-580)

Andrade, Marco Túlio Carvalho de
Professor Responsável

versão: Setembro de 2.017

© Andrade, Bruno, Midorikawa, Simplicio e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

1

State Machine Synthesis Using Transition Lists

- Uma vez que o Diagrama de Transição de Estados de uma FSM está pronto a maior parte do trabalho criativo do *design* (projeto) está concluído. O restante poderá ser levado a cabo com ferramentas de CAD.
- Do conteúdo da seção 7.5 do livro texto Wakerly, 4ª edição, sabe-se que é possível extrair uma Lista de Transição a partir da Tabela de Transição de Estados e da Codificação de Estados adotada para uma FSM.

© Andrade, Bruno, Midorikawa, Simplicio e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

2

Tabela de Transição de estados

| S | Q2 | Q1 | Q0 | Transition Expression | S* | Q2* | Q1* | Q0* |
|------|----|----|----|---|------|-----|-----|-----|
| IDLE | 0 | 0 | 0 | $(\text{LEFT} + \text{RIGHT} + \text{HAZ})'$ | IDLE | 0 | 0 | 0 |
| IDLE | 0 | 0 | 0 | $\text{LEFT} \cdot \text{HAZ}' \cdot \text{RIGHT}'$ | L1 | 0 | 0 | 1 |
| IDLE | 0 | 0 | 0 | $\text{HAZ} + \text{LEFT} \cdot \text{RIGHT}$ | LR3 | 1 | 0 | 0 |
| IDLE | 0 | 0 | 0 | $\text{RIGHT} \cdot \text{HAZ}' \cdot \text{LEFT}'$ | R1 | 1 | 0 | 1 |
| L1 | 0 | 0 | 1 | HAZ' | L2 | 0 | 1 | 1 |
| L1 | 0 | 0 | 1 | HAZ | LR3 | 1 | 0 | 0 |
| L2 | 0 | 1 | 1 | HAZ' | L3 | 0 | 1 | 0 |
| L2 | 0 | 1 | 1 | HAZ | LR3 | 1 | 0 | 0 |
| L3 | 0 | 1 | 0 | 1 | IDLE | 0 | 0 | 0 |
| R1 | 1 | 0 | 1 | HAZ' | R2 | 1 | 1 | 1 |
| R1 | 1 | 0 | 1 | HAZ | LR3 | 1 | 0 | 0 |
| R2 | 1 | 1 | 1 | HAZ' | R3 | 1 | 1 | 0 |
| R2 | 1 | 1 | 1 | HAZ | LR3 | 1 | 0 | 0 |
| R3 | 1 | 1 | 0 | 1 | IDLE | 0 | 0 | 0 |
| LR3 | 1 | 0 | 0 | 1 | IDLE | 0 | 0 | 0 |

Tab 7-14 página 576

© Andrade, Bruno, Midorikawa, Simplício e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

3

State Machine Synthesis Using Transition Lists

- Lista de Transição X Equações de Transição:
- Na síntese de FSM a partir de **Lista de Transição** o primeiro passo é a obtenção de um conjunto de **Equações de Transição** que definem cada variável V^* de **Próximo Estado** em termos do **Estado Atual** e das **Entradas**.
- **Lista de Transição** – Trata-se de uma **Tabela da Verdade Híbrida**, na qual **combinações das variáveis de estado para o estado atual são listadas explicitamente e combinações de entradas são listadas algebricamente**.

© Andrade, Bruno, Midorikawa, Simplício e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

4

State Machine Synthesis Using Transition Lists

■ Equações de Transição:

$$V^* = \sum_i (\text{termos-produto de transição})$$

Onde i consiste nas linhas da Tabela de Transição e cujo valor de $V^* = 1$.

■ Exemplo da Lista de Transição da Tabela 7-14, página 576.

A transition equation for a next-state variable V^* can be written using a sort of hybrid canonical sum:

$$V^* = \sum_{\text{transition-list rows where } V^* = 1} (\text{transition p-term})$$

© Andrade, Bruno, Midorikawa, Simplicio e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

5

State Machine Synthesis Using Transition Lists

$$\begin{aligned} Q_2^* = & Q_2' \cdot Q_1' \cdot Q_0' \cdot (\text{HAZ} + \text{LEFT} \cdot \text{RIGHT}) \\ & + Q_2' \cdot Q_1' \cdot Q_0' \cdot (\text{RIGHT} \cdot \text{HAZ}' \cdot \text{LEFT}') \\ & + Q_2' \cdot Q_1' \cdot Q_0 \cdot (\text{HAZ}) \\ & + Q_2' \cdot Q_1 \cdot Q_0 \cdot (\text{HAZ}) \\ & + Q_2 \cdot Q_1' \cdot Q_0 \cdot (\text{HAZ}') \\ & + Q_2 \cdot Q_1' \cdot Q_0 \cdot (\text{HAZ}) \\ & + Q_2 \cdot Q_1 \cdot Q_0 \cdot (\text{HAZ}') \\ & + Q_2 \cdot Q_1 \cdot Q_0 \cdot (\text{HAZ}) \end{aligned}$$

Baseado na Lista de Transição da Tabela 7-14, a Equação de Transição de Q_2^* na Máquina T-bird pode ser escrita como uma soma de 8 termos-produto.

■ Algumas manipulações e simplificações Algébricas conduzem a:

$$\begin{aligned} Q_2^* = & Q_2' \cdot Q_1' \cdot Q_0' \cdot (\text{HAZ} + \text{RIGHT}) \\ & + Q_2' \cdot Q_0 \cdot (\text{HAZ}) \\ & + Q_2 \cdot Q_0 \end{aligned}$$

© Andrade, Bruno, Midorikawa, Simplicio e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

6

State Machine Synthesis Using Transition Lists

A de Q_1^* na Máquina T-bird pode ser escrita como uma soma de 4 termos-produto.

$$\begin{aligned} Q_1^* = & Q_2' \cdot Q_1' \cdot Q_0 \cdot (HAZ') \\ & + Q_2' \cdot Q_1 \cdot Q_0 \cdot (HAZ') \\ & + Q_2 \cdot Q_1' \cdot Q_0 \cdot (HAZ') \\ & + Q_2 \cdot Q_1 \cdot Q_0 \cdot (HAZ') \end{aligned}$$

- Algumas manipulações e simplificações Algébricas conduzem a:

$$Q_1^* = Q_0 \cdot HAZ'$$

© Andrade, Bruno, Midorikawa, Simplício e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

7

State Machine Synthesis Using Transition Lists

$$\begin{aligned} Q_0^* = & Q_2' \cdot Q_1' \cdot Q_0' \cdot (LEFT \cdot HAZ' \cdot RIGHT') \\ & + Q_2' \cdot Q_1' \cdot Q_0' \cdot (RIGHT \cdot HAZ' \cdot LEFT') \\ & + Q_2' \cdot Q_1' \cdot Q_0 \cdot (HAZ') \\ & + Q_2 \cdot Q_1' \cdot Q_0 \cdot (HAZ') \end{aligned}$$

A de Q_0^* na Máquina T-bird pode ser escrita como uma soma de 4 termos-produto.

- Algumas manipulações e simplificações Algébricas conduzem a:

$$Q_0^* = Q_2' \cdot Q_1' \cdot Q_0' \cdot HAZ' \cdot (LEFT \oplus RIGHT) + Q_1' \cdot Q_0 \cdot HAZ'$$

© Andrade, Bruno, Midorikawa, Simplício e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

8

State Machine Synthesis Using Transition Lists

- Sabe-se que **Equações de Transição** não são **Equações de Excitação**.
- O que deve ser feito é que se encontrem quais são os valores de excitação D_i^* (para um FF tipo D) ou $J_i^* K_i^*$ (para um FF tipo JK) para que, uma vez no Estado $S(Q_2 Q_1 Q_0)$, a FSM evolua para o Estado $S^*(Q_2^* Q_1^* Q_0^*)$:

$$J_2^* \text{ e } K_2^* \text{ ou } D_2^* \Rightarrow Q_2^*$$

$$J_1^* \text{ e } K_1^* \text{ ou } D_1^* \Rightarrow Q_1^*$$

$$J_0^* \text{ e } K_0^* \text{ ou } D_0^* \Rightarrow Q_0^*$$

© Andrade, Bruno, Midorikawa, Simplício e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

9

State Machine Synthesis Using Transition Lists

Therefore, if the transition equation for a state variable Q_i^* is

$$Q_i^* = \text{expression}$$

then the excitation equation for the corresponding D flip-flop input is

$$D_i = \text{expression}$$

- É óbvio que se o FF é do tipo D a derivação das Equações de Excitação a partir das Equações de Transição é muito simples!

© Andrade, Bruno, Midorikawa, Simplício e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

10

State Machine Synthesis Using Transition Lists

- Existem outras maneiras de se obter Equações de Transição e de Excitação. Se a coluna de um Estado particular conta com menos 0s que 1s, pode ser vantajoso escrever as Variáveis de Transição em termos de 0s nesta coluna:

Complemento(V^*) = Σ_i (termos-produto de transição)

Onde i consiste nas linhas da Tabela de Transição e cujo valor de $V^* = 0$.

$$V^{*'} = \sum_{\text{transition-list rows where } V^* = 0} (\text{transition p-term})$$

© Andrade, Bruno, Midorikawa, Simplício e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

11

State Machine Synthesis Using Transition Lists

- Se a coluna de um Estado particular conta com menos 0s que 1s, pode ser vantajoso escrever as Variáveis de Transição em termos de 0s nesta coluna. Deste modo:

$$\begin{aligned} Q2^{*'} &= Q2' \cdot Q1' \cdot Q0' \cdot ((\text{LEFT} + \text{RIGHT} + \text{HAZ})') \\ &\quad + Q2' \cdot Q1' \cdot Q0' \cdot (\text{LEFT} \cdot \text{HAZ}' \cdot \text{RIGHT}') \\ &\quad + Q2' \cdot Q1' \cdot Q0 \cdot (\text{HAZ}') \\ &\quad + Q2' \cdot Q1 \cdot Q0 \cdot (\text{HAZ}') \\ &\quad + Q2' \cdot Q1 \cdot Q0' \cdot (1) \\ &\quad + Q2 \cdot Q1 \cdot Q0' \cdot (1) \\ &\quad + Q2 \cdot Q1' \cdot Q0' \cdot (1) \\ &= Q2' \cdot Q1' \cdot Q0' \cdot \text{HAZ}' \cdot \text{RIGHT}' + Q2' \cdot Q0 \cdot \text{HAZ}' + Q1 \cdot Q0' + Q2 \cdot Q0' \end{aligned}$$

© Andrade, Bruno, Midorikawa, Simplício e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

12

State Machine Synthesis Using Transition Lists

■ .

To obtain an expression for a next-state variable V^* directly, using the 0s in the transition list, we can complement the righthand side of the general V^* equation using DeMorgan's theorem, obtaining a sort of hybrid canonical product:

$$V^* = \prod_{\text{transition-list rows where } V^* = 0} (\text{transition s-term})$$

© Andrade, Bruno, Midorikawa, Simplício e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

13

Another State Machine Design Example

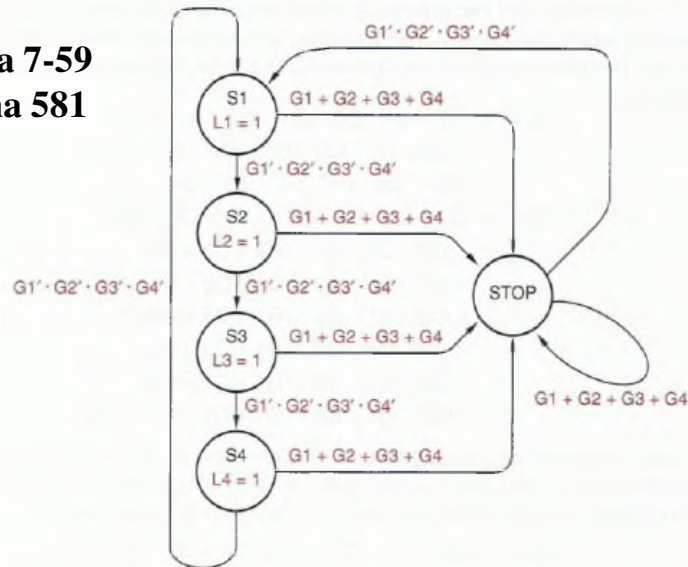
- The Guessing Game, seção 7.7, página 580.
 - 4 entradas G1, G2, G3 e G4 ligadas a 4 botões;
 - 4 saídas L1, L2, L3 e L4 ligadas a 4 LEDs, no padrão “**1-out-of-4**”;
 - cada ciclo de *clock* o padrão é rodado uma posição a uma frequência do *clock* próxima de 4 Hertz;
 - Adivinhações são feitas apertando os botões, o que determina uma escolha G_i ;
 - Se no próximo pulso de *clock*, após o aperto do botão escolhido, G_i é igual ao LED aceso houve acerto (o jogo para, saída ERR mantém mesmo valor), caso contrário saída ERR é ativada.

© Andrade, Bruno, Midorikawa, Simplício e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

14

Another State Machine Design Example

Figura 7-59
Página 581

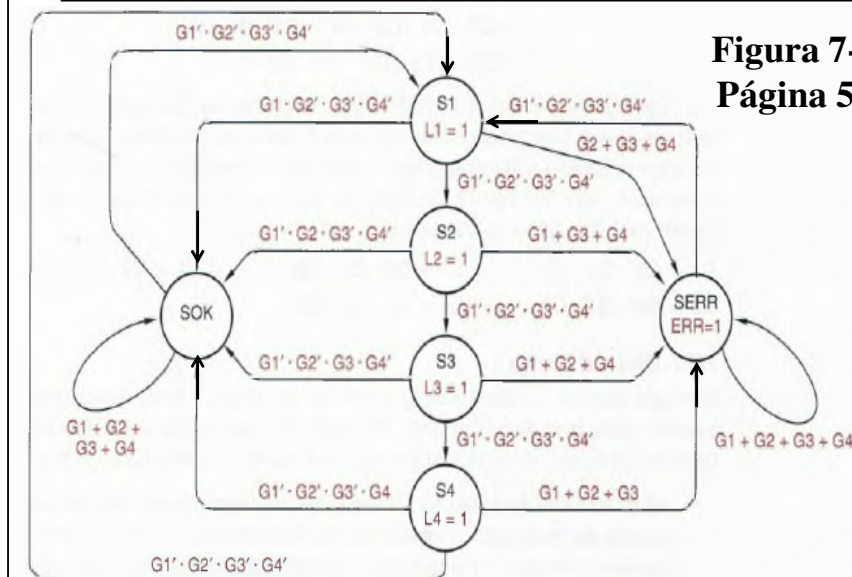


© Andrade, Bruno, Midorikawa, Simplício e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

15

Another State Machine Design Example

Figura 7-60
Página 581



© Andrade, Bruno, Midorikawa, Simplício e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

16

Another State Machine Design Example

- Uma Lista de Transição que corresponde ao Diagrama de Transição de Estados (Figura 7-60, página 581) pode ser vista na Tabela 7-15, página 583.
- Na obtenção de uma expressão para Q_2^* observa-se na Tabela 7-15 ela pode ser expressa como uma soma de 10 produtos ou como um produto de 6 somas. Opta-se por representar Q_2^* pelos seus 0s da Tabela 7-15.

© Andrade, Bruno, Midorikawa, Simplício e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

17

Another State Machine Design Example

- Opta-se por representar Q_2^* pelos seus 0s da Tabela 7-15, ou seja, obter a representação de “ $\sim Q_2^*$ ” (Q_2^* complementado):

$$\begin{aligned}
 Q_2^{*'} &= Q_2' \cdot Q_1' \cdot Q_0' \cdot (G_1' \cdot G_2' \cdot G_3' \cdot G_4') \\
 &\quad + Q_2' \cdot Q_1' \cdot Q_0 \cdot (G_1' \cdot G_2' \cdot G_3' \cdot G_4') \\
 &\quad + Q_2' \cdot Q_1 \cdot Q_0 \cdot (G_1' \cdot G_2' \cdot G_3' \cdot G_4') \\
 &\quad + Q_2' \cdot Q_1 \cdot Q_0' \cdot (G_1' \cdot G_2' \cdot G_3' \cdot G_4') \\
 &\quad + Q_2 \cdot Q_1' \cdot Q_0' \cdot (G_1' \cdot G_2' \cdot G_3' \cdot G_4') \\
 &\quad + Q_2 \cdot Q_1' \cdot Q_0 \cdot (G_1' \cdot G_2' \cdot G_3' \cdot G_4') \\
 &= (Q_2' + Q_1') \cdot (G_1' \cdot G_2' \cdot G_3' \cdot G_4')
 \end{aligned}$$

© Andrade, Bruno, Midorikawa, Simplício e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

18

Another State Machine Design Example

$$\begin{aligned}
 Q1^* &= Q2' \cdot Q1' \cdot Q0 \cdot (G1' \cdot G2' \cdot G3' \cdot G4') \\
 &\quad + Q2' \cdot Q1 \cdot Q0 \cdot (G1' \cdot G2' \cdot G3' \cdot G4') \\
 &= Q2' \cdot Q0 \cdot G1' \cdot G2' \cdot G3' \cdot G4' \\
 Q0^* &= Q2' \cdot Q1' \cdot Q0' \cdot (G1' \cdot G2' \cdot G3' \cdot G4') \\
 &\quad + Q2' \cdot Q1' \cdot Q0' \cdot (G2 + G3 + G4) \\
 &\quad + Q2' \cdot Q1' \cdot Q0 \cdot (G1' \cdot G2' \cdot G3' \cdot G4') \\
 &\quad + Q2' \cdot Q1' \cdot Q0 \cdot (G1 + G3 + G4) \\
 &\quad + Q2' \cdot Q1 \cdot Q0 \cdot (G1 + G2 + G4) \\
 &\quad + Q2' \cdot Q1 \cdot Q0' \cdot (G1 + G2 + G3) \\
 &\quad + Q2 \cdot Q1' \cdot Q0 \cdot (G1 + G2 + G3 + G4)
 \end{aligned}$$

$$\begin{aligned}
 L1 &= Q2' \cdot Q1' \cdot Q0' & L3 &= Q2' \cdot Q1 \cdot Q0 & ERR &= Q2 \cdot Q1' \cdot Q0 \\
 L2 &= Q2' \cdot Q1' \cdot Q0 & L4 &= Q2' \cdot Q1 \cdot Q0'
 \end{aligned}$$

© Andrade, Bruno, Midorikawa, Simplicio e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

19

Another State Machine Design Example

Table 7-15 Transition list for guessing-game machine.

| Current State | | | | | Next State | | | | Output | | | | |
|---------------|----|----|----|-------------------------------------|------------|-----|-----|-----|--------|----|----|----|-----|
| S | Q2 | Q1 | Q0 | Transition Expression | S* | Q2* | Q1* | Q0* | L1 | L2 | L3 | L4 | ERR |
| S1 | 0 | 0 | 0 | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S2 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| S1 | 0 | 0 | 0 | $G1 \cdot G2' \cdot G3' \cdot G4'$ | SOK | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| S1 | 0 | 0 | 0 | $G2 + G3 + G4$ | SERR | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| S2 | 0 | 0 | 1 | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S3 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| S2 | 0 | 0 | 1 | $G1' \cdot G2 \cdot G3' \cdot G4'$ | SOK | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| S2 | 0 | 0 | 1 | $G1 + G3 + G4$ | SERR | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| S3 | 0 | 1 | 1 | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S4 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| S3 | 0 | 1 | 1 | $G1' \cdot G2' \cdot G3 \cdot G4'$ | SOK | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| S3 | 0 | 1 | 1 | $G1 + G2 + G4$ | SERR | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| S4 | 0 | 1 | 0 | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| S4 | 0 | 1 | 0 | $G1' \cdot G2' \cdot G3' \cdot G4$ | SOK | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| S4 | 0 | 1 | 0 | $G1 + G2 + G3$ | SERR | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| SOK | 1 | 0 | 0 | $G1 + G2 + G3 + G4$ | SOK | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| SOK | 1 | 0 | 0 | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| SERR | 1 | 0 | 1 | $G1 + G2 + G3 + G4$ | SERR | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| SERR | 1 | 0 | 1 | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

© Andrade, Bruno, Midorikawa, Simplicio e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

20

Another State Machine Design Example

- Pode ser útil determinar, dentre as variáveis de saída L_1 , L_2 , L_3 , L_4 e **ERR** (Tabela 7-15, página 583) quais serão seus próximos valores, L_1^* , L_2^* , L_3^* , L_4^* e **ERR**^{*}.
- Para exemplificar a obtenção de L_1^* basta procurar na Tabela 7-15 quantas e **quais** as **situações** em que o **Estado S_1** (onde L_1 deve acender) é o **Próximo Estado a ser atingido**.
- Procedimento semelhante é usado na obtenção das expressões das demais variáveis de saída.

© Andrade, Bruno, Midorikawa, Simplício e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

21

Another State Machine Design Example

Table 7-15 Transition list for guessing-game machine.

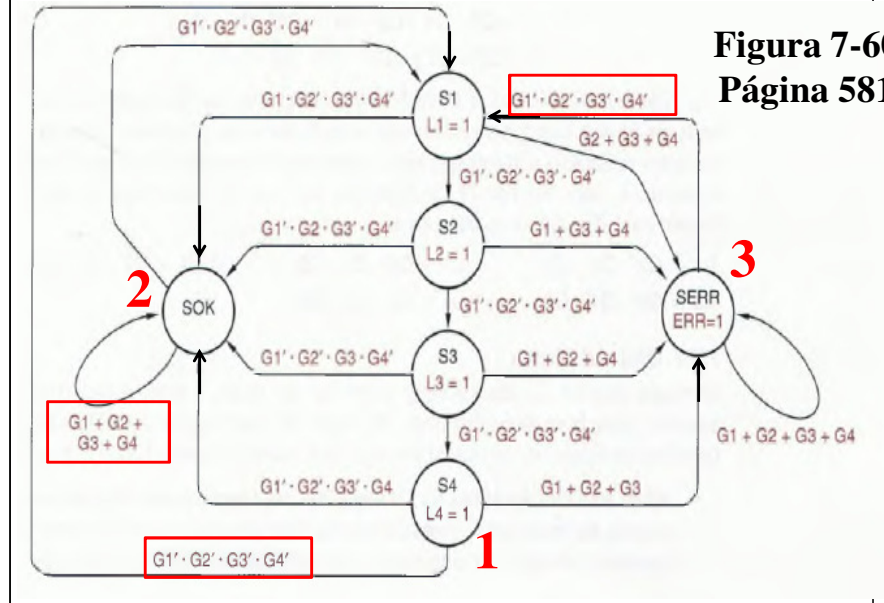
| Current State | | | | | Next State | | | | Output | | | | |
|---------------|----|----|----|-------------------------------------|------------|-----|-----|-----|---------|----|----|----|-----|
| S | Q2 | Q1 | Q0 | Transition Expression | S* | Q2* | Q1* | Q0* | L1 | L2 | L3 | L4 | ERR |
| S1 | 0 | 0 | 0 | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S2 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| S1 | 0 | 0 | 0 | $G1 \cdot G2' \cdot G3' \cdot G4'$ | SOK | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| S1 | 0 | 0 | 0 | $G2 + G3 + G4$ | SERR | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| S2 | 0 | 0 | 1 | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S3 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| S2 | 0 | 0 | 1 | $G1' \cdot G2 \cdot G3' \cdot G4'$ | SOK | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| S2 | 0 | 0 | 1 | $G1 + G3 + G4$ | SERR | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| S3 | 0 | 1 | 1 | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S4 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| S3 | 0 | 1 | 1 | $G1' \cdot G2' \cdot G3 \cdot G4'$ | SOK | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| S3 | 0 | 1 | 1 | $G1 + G2 + G4$ | SERR | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 S4 | 0 | 1 | 0 | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S1 | 0 | 0 | 0 | L_1^* | 0 | 0 | 0 | 1 |
| S4 | 0 | 1 | 0 | $G1' \cdot G2' \cdot G3' \cdot G4$ | SOK | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| S4 | 0 | 1 | 0 | $G1 + G2 + G3$ | SERR | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| SOK | 1 | 0 | 0 | $G1 + G2 + G3 + G4$ | SOK | 1 | 0 | 0 | L_1^* | 0 | 0 | 0 | 0 |
| 2 SOK | 1 | 0 | 0 | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S1 | 0 | 0 | 0 | L_1^* | 0 | 0 | 0 | 0 |
| SERR | 1 | 0 | 1 | $G1 + G2 + G3 + G4$ | SERR | 1 | 0 | 1 | L_1^* | 0 | 0 | 0 | 1 |
| 3 SERR | 1 | 0 | 1 | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S1 | 0 | 0 | 0 | L_1^* | 0 | 0 | 0 | 1 |

© Andrade, Bruno, Midorikawa, Simplício e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

22

Another State Machine Design Example

Figura 7-60
Página 581



© Andrade, Bruno, Midorikawa, Simplício e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

23

Another State Machine Design Example

■ As expressões obtidas são:

$$\begin{aligned}
 L1^* &= L1' \cdot L2' \cdot L3' \cdot L4' \cdot ERR' \cdot (G1' \cdot G2' \cdot G3' \cdot G4') \\
 &+ L1' \cdot L2' \cdot L3' \cdot L4' \cdot ERR' \cdot (G1' \cdot G2' \cdot G3' \cdot G4') \\
 &+ L1' \cdot L2' \cdot L3' \cdot L4' \cdot ERR \cdot (G1' \cdot G2' \cdot G3' \cdot G4') \\
 L2^* &= L1 \cdot L2' \cdot L3' \cdot L4' \cdot ERR' \cdot (G1' \cdot G2' \cdot G3' \cdot G4') \\
 L3^* &= L1' \cdot L2 \cdot L3' \cdot L4' \cdot ERR' \cdot (G1' \cdot G2' \cdot G3' \cdot G4') \\
 L4^* &= L1' \cdot L2' \cdot L3 \cdot L4' \cdot ERR' \cdot (G1' \cdot G2' \cdot G3' \cdot G4') \\
 ERR^* &= L1 \cdot L2' \cdot L3' \cdot L4' \cdot ERR' \cdot (G2 + G3 + G4) \\
 &+ L1' \cdot L2 \cdot L3' \cdot L4' \cdot ERR' \cdot (G1 + G3 + G4) \\
 &+ L1' \cdot L2' \cdot L3 \cdot L4' \cdot ERR' \cdot (G1 + G2 + G4) \\
 &+ L1' \cdot L2' \cdot L3' \cdot L4 \cdot ERR' \cdot (G1 + G2 + G3) \\
 &+ L1' \cdot L2' \cdot L3' \cdot L4' \cdot ERR \cdot (G1 + G2 + G3 + G4)
 \end{aligned}$$

© Andrade, Bruno, Midorikawa, Simplício e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

24

Another State Machine Design Example

- Sabe-se que as saídas da máquina são função apenas do Estado Atual. Uma combinação diferente de valores de saída é produzida em cada Estado definido, podendo-se utilizar os valores das saídas como Variáveis de Estado.
- Pode-se associar, a cada Estado definido, um código que represente a combinação requerida do conjunto de saídas para aquele Estado.
- Este tipo de associação, *output-coded state assignment*, pode resultar em um conjunto de equações de transição mais simples.

© Andrade, Bruno, Midorikawa, Simplicio e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

25

Another State Machine Design Example

Table 7-16 Transition list for guessing-game machine using outputs as state variables.

| Current State | | | | | | Transition Expression | Next State | | | | | |
|---------------|----|----|----|----|-----|-------------------------------------|------------|-----|-----|-----|-----|------|
| S | L1 | L2 | L3 | L4 | ERR | | S* | L1* | L2* | L3* | L4* | ERR* |
| S1 | 1 | 0 | 0 | 0 | 0 | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S2 | 0 | 1 | 0 | 0 | 0 |
| S1 | 1 | 0 | 0 | 0 | 0 | $G1 \cdot G2' \cdot G3' \cdot G4'$ | SOK | 0 | 0 | 0 | 0 | 0 |
| S1 | 1 | 0 | 0 | 0 | 0 | $G2 + G3 + G4$ | SERR | 0 | 0 | 0 | 0 | 1 |
| S2 | 0 | 1 | 0 | 0 | 0 | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S3 | 0 | 0 | 1 | 0 | 0 |
| S2 | 0 | 1 | 0 | 0 | 0 | $G1' \cdot G2 \cdot G3' \cdot G4'$ | SOK | 0 | 0 | 0 | 0 | 0 |
| S2 | 0 | 1 | 0 | 0 | 0 | $G1 + G3 + G4$ | SERR | 0 | 0 | 0 | 0 | 1 |
| S3 | 0 | 0 | 1 | 0 | 0 | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S4 | 0 | 0 | 0 | 1 | 0 |
| S3 | 0 | 0 | 1 | 0 | 0 | $G1' \cdot G2' \cdot G3 \cdot G4'$ | SOK | 0 | 0 | 0 | 0 | 0 |
| S3 | 0 | 0 | 1 | 0 | 0 | $G1 + G2 + G4$ | SERR | 0 | 0 | 0 | 0 | 1 |
| S4 | 0 | 0 | 0 | 1 | 0 | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S1 | 1 | 0 | 0 | 0 | 0 |
| S4 | 0 | 0 | 0 | 1 | 0 | $G1' \cdot G2' \cdot G3' \cdot G4$ | SOK | 0 | 0 | 0 | 0 | 0 |
| S4 | 0 | 0 | 0 | 1 | 0 | $G1 + G2 + G3$ | SERR | 0 | 0 | 0 | 0 | 1 |
| SOK | 0 | 0 | 0 | 0 | 0 | $G1 + G2 + G3 + G4$ | SOK | 0 | 0 | 0 | 0 | 0 |
| SOK | 0 | 0 | 0 | 0 | 0 | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S1 | 1 | 0 | 0 | 0 | 0 |
| SERR | 0 | 0 | 0 | 0 | 1 | $G1 + G2 + G3 + G4$ | SERR | 0 | 0 | 0 | 0 | 1 |
| SERR | 0 | 0 | 0 | 0 | 1 | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S1 | 1 | 0 | 0 | 0 | 0 |

© Andrade, Bruno, Midorikawa, Simplicio e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

26

Another State Machine Design Example

| State | L1 | L2 | L3 | L4 | ERR |
|-------|----|----|----|----|-----|
| S1 | 1 | x | x | x | x |
| S2 | 0 | 1 | x | x | x |
| S3 | 0 | 0 | 1 | x | x |
| S4 | 0 | 0 | 0 | 1 | x |
| SOK | 0 | 0 | 0 | 0 | 0 |
| SERR | 0 | 0 | 0 | 0 | 1 |

Table 7-17
Current-state assignment
for the guessing-game
machine using don't-cares.

© Andrade, Bruno, Midorikawa, Simplício e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

27

Another State Machine Design Example

Table 7-18 Transition list for guessing-game machine using don't-care state codings.

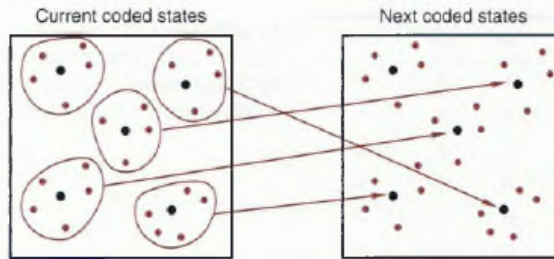
| Current State | | | | | | Transition Expression | Next State | | | | | |
|---------------|----|----|----|----|-----|-------------------------------------|------------|-----|-----|-----|-----|------|
| S | L1 | L2 | L3 | L4 | ERR | | S* | L1* | L2* | L3* | L4* | ERR* |
| S | 1 | x | x | x | x | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S2 | 0 | 1 | 0 | 0 | 0 |
| S | 1 | x | x | x | x | $G1 \cdot G2' \cdot G3' \cdot G4'$ | SOK | 0 | 0 | 0 | 0 | 0 |
| S | 1 | x | x | x | x | $G2 + G3 + G4$ | SERR | 0 | 0 | 0 | 0 | 1 |
| S2 | 0 | 1 | x | x | x | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S3 | 0 | 0 | 1 | 0 | 0 |
| S2 | 0 | 1 | x | x | x | $G1' \cdot G2 \cdot G3' \cdot G4'$ | SOK | 0 | 0 | 0 | 0 | 0 |
| S2 | 0 | 1 | x | x | x | $G1 + G3 + G4$ | SERR | 0 | 0 | 0 | 0 | 1 |
| S3 | 0 | 0 | 1 | x | x | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S4 | 0 | 0 | 0 | 1 | 0 |
| S3 | 0 | 0 | 1 | x | x | $G1' \cdot G2' \cdot G3 \cdot G4'$ | SOK | 0 | 0 | 0 | 0 | 0 |
| S3 | 0 | 0 | 1 | x | x | $G1 + G2 + G4$ | SERR | 0 | 0 | 0 | 0 | 1 |
| S4 | 0 | 0 | 0 | 1 | x | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S1 | 1 | 0 | 0 | 0 | 0 |
| S4 | 0 | 0 | 0 | 1 | x | $G1' \cdot G2' \cdot G3' \cdot G4$ | SOK | 0 | 0 | 0 | 0 | 0 |
| S4 | 0 | 0 | 0 | 1 | x | $G1 + G2 + G3$ | SERR | 0 | 0 | 0 | 0 | 1 |
| SOK | 0 | 0 | 0 | 0 | 0 | $G1 + G2 + G3 + G4$ | SOK | 0 | 0 | 0 | 0 | 0 |
| SOK | 0 | 0 | 0 | 0 | 0 | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S1 | 1 | 0 | 0 | 0 | 0 |
| SERR | 0 | 0 | 0 | 0 | 1 | $G1 + G2 + G3 + G4$ | SERR | 0 | 0 | 0 | 0 | 1 |
| SERR | 0 | 0 | 0 | 0 | 1 | $G1' \cdot G2' \cdot G3' \cdot G4'$ | S1 | 1 | 0 | 0 | 0 | 0 |

© Andrade, Bruno, Midorikawa, Simplício e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

28

Another State Machine Design Example

Figure 7-61
State assignment
using don't-cares for
current states.



$$\begin{aligned}
 \text{ERR}^* = & L1 \cdot (G2 + G3 + G4) \\
 & + L1' \cdot L2 \cdot (G1 + G3 + G4) \\
 & + L1' \cdot L2' \cdot L3 \cdot (G1 + G2 + G4) \\
 & + L1' \cdot L2' \cdot L3' \cdot L4 \cdot (G1 + G2 + G3) \\
 & + L1' \cdot L2' \cdot L3' \cdot L4' \cdot \text{ERR} \cdot (G1 + G2 + G3 + G4)
 \end{aligned}$$

© Andrade, Bruno, Midorikawa, Simplicio e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

29

Referencias

- Referências desta aula:
- Wakerly, 7-6 – *State Machine Synthesis Using Transition Lists* (4ed, páginas 577-580).

© Andrade, Bruno, Midorikawa, Simplicio e Spina 2.017 <Sint. Circ. Seq.> PCS 3225 Sistemas Digitais II

30

Livro Texto

- Wakerly, J.F.; *Digital Design – Principles & Practices*; Fourth Edition, ISBN: 0-13-186389-4, Pearson & Prentice-Hall, Upper Saddle, River, New Jersey, 07458, 2006.

Bibliografia Adicional Deste Assunto

- Dias, Francisco José de Oliveira; *Introdução aos Circuitos de Chaveamento*; Apostila, PEL/EPUSP, 1.980;
- Fregni, Edson; Ranzini, Edith; *Teoria da Comutação: Introdução aos Circuitos Digitais (Partes 1 e 2)*; Apostila PCS/EPUSP, Outubro de 1.999;

Bibliografia Adicional Deste Assunto

- Hill, Frederic and Peterson, Gerald;
Introduction to Switching Theory and Logical Design; Ed. John Wiley and Sons, 1.974;
- Ranzini, Edith; *Circuitos de Chaveamento*
(notas de aula); Apostila, EPUSP, 1.983.