

MÁQUINA DE ESTADOS FINITOS EM CIRCUITOS INTEGRADOS - IMPLEMENTAÇÃO EM PLA NÃO DETERMINÍSTICA.

H.T. SILVA*

SUMÁRIO

Este trabalho apresenta uma abordagem diferente para a implementação de máquinas de estados finitos em circuitos integrados. A máquina de estados finitos será representada por um autômato finito não determinístico; convencionalmente ela é representada por um autômato finito determinístico.

Para integrar a máquina será utilizada uma PLA. A PLA que implementa a máquina determinística é normalmente obtida a partir de sua tabela de transição de estados, através de técnicas conhecidas de projetos lógicos. A PLA não determinística será obtida diretamente de seu diagrama de estados.

Será realizada uma comparação entre as PLA's para o caso particular de um controlador projetado como parte de um circuito integrado que controla o acesso a uma rede local.

ABSTRACT

This work shows a different implementation of finite state machines in integrated circuits. The finite state machine will be represented by a nondeterministic finite automata; conventionally it is represented by a deterministic finite automata.

The machine will be integrated in a PLA. The PLA that implements the deterministic machine is normally projected from its state transition table using well-known techniques of logic projects. The nondeterministic PLA will be projected directly from its state diagram.

A comparison between these PLA's is made for a particular case of a controller projected as part of an integrated circuit that controls the access of a local network.

* Engenharia eletrônica (UFRJ, 1981); Projeto de Circuitos Integrados Digitais; Núcleo de Computação Eletrônica da UFRJ (NCE/UFRJ), Ilha do Fundão, Cidade Universitária - Rio de Janeiro, RJ, CEP 20001.

O comportamento de qualquer sistema pode ser representado em termos de relações matemáticas entre 3 conjuntos de variáveis, as quais descrevem as entradas, as saídas e os estados do sistema. O conjunto de entradas representa as excitações externas que são aplicadas ao sistema para produzir uma mudança no seu comportamento; o conjunto de saídas representa os pontos de observação que caracterizam a mudança do sistema em resposta às entradas. Uma das características básicas de qualquer sistema é que sua saída atual é função não só das entradas atuais mas também das entradas passadas e das saídas. Devido a isto, podemos pensar em um sistema como possuindo uma "memória" que guarda informações acerca do comportamento anterior do sistema. O conjunto de estados, o terceiro conjunto de variáveis, é pois, usado para representar a quantidade de informação armazenada no sistema.

As relações funcionais que descrevem o comportamento de um sistema podem ser representadas por equações diferenciais - integrais e, portanto, as variáveis deste sistema assumem valores contínuos, ou por autômatos, em que as variáveis assumem valores discretos no tempo.

Um autômato pode apresentar um comportamento determinístico ou não determinístico. O autômato é determinístico quando cada transição do sistema faz com que este vá de um estado para outro, de forma que a cada instante o sistema esteja em apenas um de seus possíveis estados; o autômato é não determinístico quando puder se encontrar em vários estados simultaneamente. A descrição formal dos autômatos será mostrada mais adiante.

É necessário explicar aqui que o conceito de não determinismo não significa que o sistema não é bem determinado, ou seja, suas transições são aleatórias. O autômato não determinístico é apenas uma abstração matemática; enquanto no autômato determinístico toda a "memória" do sistema; em um dado instante de tempo, é representada por apenas um de seus estados; no autômato não determinístico a representação de seu passado está contida em um conjunto de estados. Fisicamente, tanto em um como em outro sistema, as medidas efetuadas para caracterizá-los são as mesmas, ou seja, as variáveis de entrada, de saída e de estado são da mesma natureza. A diferença física entre eles está no fato de que o sistema não determinístico aceita paralelismo; vários caminhos podem ser percorridos ao mesmo tempo para se chegar ao resultado final. Durante o processamento das entradas estes caminhos vão sendo abandonados, ficando apenas um deles, que levará ao resultado final.

Embora a diferença entre autômatos determinísticos e não determinísticos seja apenas conceitual, a implementação em "hardware" destes autômatos são bastante diferentes, conforme será mostrado no decorrer do artigo.

MÁQUINAS DE ESTADOS FINITOS EM CIRCUITOS INTEGRADOS

A forma mais simples de um sistema digital síncrono é a máquina sequencial de estados

finitos {BOOT 67}. Embora sistemas complexos possam ser descritos, é geralmente possível que estes possam ser construídos a partir de máquinas sequenciais.

Uma máquina de estados finitos é formada de uma lógica combinacional e unidades de armazenamento (registradores de estados). As variações na implementação desta máquina em circuitos integrados está relacionada com a forma de implementação da lógica combinacional e com a organização dos registradores de estado. A máquina sequencial pode ser realizada através de ROM's, PLA's, geradores de tempo ou combinações destes {OBRE 82}. Trataremos aqui da implementação de máquinas sequenciais de estados finitos em PLA's {MEAD 80} a partir da descrição das mesmas em forma de um autômato finito.

AUTOMATA DETERMINÍSTICO

Um autômato M sobre um alfabeto I é um sistema (Q, I, S, q_0, F, Z) definido por {HOPC 69}, {BOOT 67}:

1. Um conjunto de elementos Q denominados estados;
2. Um conjunto finito I denominado alfabeto de entrada;
3. Uma função S de mapeamento de $Q \times I$ em Q ;
4. Um conjunto finito Z denominado alfabeto de saída;
5. Um estado inicial q_0 em Q ;
6. Uma função F de mapeamento de $Q \times I$ em Z .

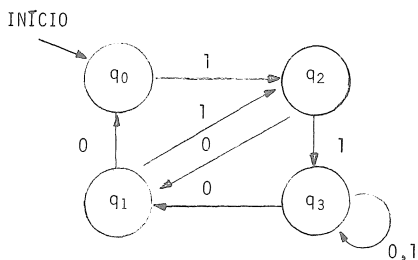
Quando o conjunto de elementos que formam os estados é finito, dizemos que o autômato é finito.

Um autômato finito é representado, para uma melhor visualização, através de um diagrama de estado. Neste diagrama cada estado é representado por um nó e as possíveis transições por linhas direccionadas rotuladas pelas excitações externas que provocam estas transições (variáveis de entrada).

Como exemplo, suponha um autômato finito determinístico que detecta a existência de dois 1's consecutivos ou dois 0's consecutivos em uma cadeia de caracteres. Sua descrição e seu diagrama de estados são apresentados nas figs. 1a e 1b.

$M = (Q, I, S, q_0, F, Z)$
 $I = \{0, 1\}$
 $Z = \{0, 1\}$
 $Q = \{q_0, q_1, q_2, q_3\}$
 $F(q_0, x) = 0$
 $F(q_1, x) = 0$
 $F(q_2, x) = 0$
 $F(q_3, x) = 1$
 $S(q_0, 0) = q_1$
 $S(q_0, 1) = q_2$
 $S(q_1, 0) = q_3$
 $S(q_1, 1) = q_2$
 $S(q_2, 0) = q_1$
 $S(q_2, 1) = q_3$
 $S(q_3, 0) = q_3$
 $S(q_3, 1) = q_3$

(a)



(b)

Fig. 1 - Autômato finito determinístico que detecta dois 0's ou dois 1's consecutivos.

(a) Descrição e (b) Diagrama de estados.

Um autômato não determinístico M sobre um alfabeto I é um sistema (Q, I, S, q_0, F, Z) definido por {HOPC 69}, {BOOT 67}:

1. Um conjunto de elementos Q denominados estados;
2. Um conjunto finito I denominado alfabeto de entrada;
3. Um conjunto finito Z denominado alfabeto de saída;
4. Uma função S de mapeamento de $Q \times I$ em subconjuntos de Q ;
5. Um conjunto de estados iniciais em Q ;
6. Uma função F de mapeamento de $Q \times I$ em Z ;

A diferença mais importante entre o autômato determinístico e o não determinístico é que, neste último, $S(q,a)$ é um conjunto de estados (pode ser vazio) ao invés de um único estado. A interpretação de $S(q,a) = \{p_1, p_2, \dots, p_k\}$ é que M, no estado q, tendo como entrada a, poderá transicionar para os estados p₁, p₂, ...p_k.

Portanto, o sistema representado por um autômato não determinístico poderá estar em vários estados simultaneamente.

O exemplo anterior, representado por um autômato não determinístico é mostrado nas figuras 2a e 2b.

$M = (Q, I, S, q, F, Z)$

$I = \{0, 1\}$

$Z = \{0, 1\}$

$Q = \{q_0, q_1, q_2, q_3\}$

$F(q_0, x) = 0$

$F(q_1, x) = 0$

$F(q_2, x) = 0$

$F(q_3, x) = 1$

$S(q_0, 0) = \{q_0, q_2\}$

$S(q_0, 1) = \{q_0, q_1\}$

$S(q_1, 0) = \emptyset$

$S(q_1, 1) = \{q_3\}$

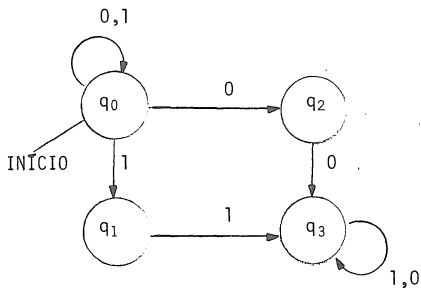
$S(q_2, 0) = \{q_3\}$

$S(q_2, 1) = \emptyset$

$S(q_3, 0) = \{q_3\}$

$S(q_3, 1) = \{q_3\}$

(a)



(b)

Fig. 2 - Autômato finito não determinístico que detecta dois 0's ou dois 1's consecutivos.

(a) Descrição e (b) Diagrama de estados.

CONTROLADOR DA RECEPÇÃO DE PACOTES DA REDE

Para ilustrar a abordagem de PLA determinística e não determinística utilizaremos o exemplo de um controlador. A lógica deste controlador faz parte da lógica de acesso a uma rede local em anel existente no NCE/UFRJ {SILV 83}. Atualmente esta lógica de acesso à rede local está sendo integrada; neste CI, o controlador de recepção de mensagens aqui mencionado como exemplo foi projetado em uma PLA não determinística.

O controlador pode ser representado pelo diagrama em bloco da figura 3.

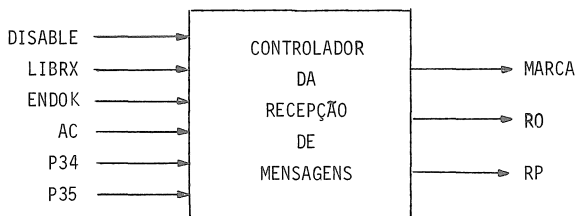


Fig. 3 - Controlador da Recepção de Mensagens - Diagrama em Bloco.

Sua lógica é a seguinte:

Inicialmente o controlador está liberado para receber mensagens da rede. Estas são recebidas na forma de pacotes. O sinal de "DISABLE" coloca sempre o controlado no estado de recepção livre. Quando chega uma mensagem pela rede, é enviado para o controlador o sinal "ENDOK" caso a mensagem seja para a estação deste controlador. A seguir é recebido um sinal "AC" informando se a estação pode aceitar ou não mensagens deste destinatário. O controlador aceitará a mensagem caso não esteja ocupado com a mensagem recebida anteriormente. P34 e P35 significam os tempos do pacote em que devem ser escritos, no próximo pacote recebido, os bits de status, através da saída serial MARCA, avisando à estação remetente do pacote o estado da recepção do mesmo.

Os "status" do pacote são os seguintes:

MARCA		
P34	P35	
0	0	pacote ignorado (a estação provavelmente está desligada)
0	1	pacote aceito pela estação
1	0	estação não foi habilitada para receber deste destinatário
1	1	estação ocupada com a recepção do pacote anterior

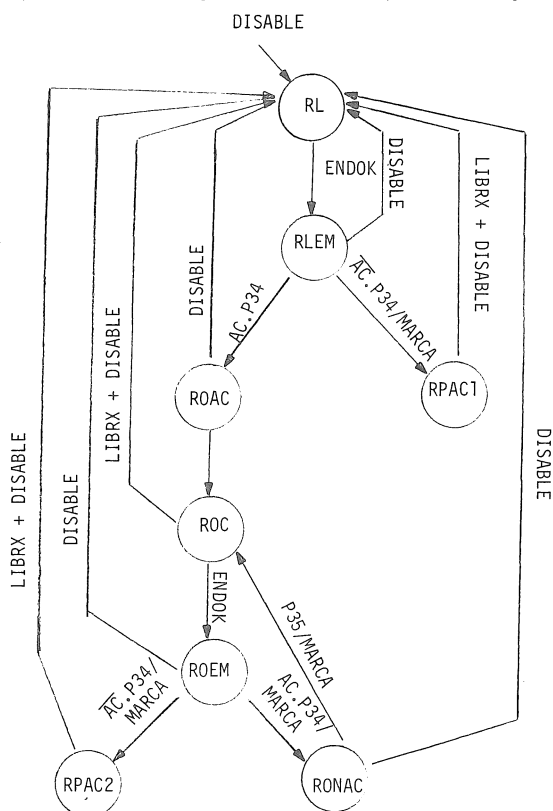
Ao receber um pacote aceito ou rejeitado o controlador gera os sinais "R0" (Recepção Ocupada) ou "RP" (Rejeição Pacote) respectivamente e somente fica apto a receber novos pacotes quando lhe é enviado o sinal "LIBRX", indicando a liberação da estação para nova leitura.

A seguir apresentaremos um autômato determinístico e outro não determinístico que implementam este controlador. O autômato determinístico não corresponde à conversão do não determinístico em determinístico {HOPC 69}. Apenas ambos representam corretamente o controlador em questão.

IMPLEMENTAÇÃO DO CONTROLADOR EM PLA DETERMINÍSTICA

Na fig. 4 temos o diagrama de transição de estados simplificado para o controlador representando uma máquina sequencial de estados finitos determinística. O diagrama apre

sentado encontra-se simplificado, uma vez que devido às dimensões da máquina (6 entradas) seria difícil a manipulação de um diagrama de transição levando-se em consideração todas as possíveis combinações das entradas para a mudança de cada estado da máquina.



Estados:

RL - Recepção Livre;

RLEM - Chegada de Pacote para a Estação que estava Livre;

ROAC - Aceitação de Pacote com Entrada em Recepção Ocupada;

RPAC1 - Rejeição de Pacote em Recepção Livre;

ROC - Recepção Ocupada com Pacote aceito;

ROEM - Chegada de Pacote para a Estação que já estava Ocupada;

RPAC2 - Rejeição de Pacote em Recepção Ocupada;

RONAC - Não aceitação de Pacote devido à Ocupação da Recepção com o Pacote Anterior.

Fig. 4 - Autômato Determinístico do Controlador da Fig. 3.

A codificação para os estados do controlador, escolhida de forma a minimizar as equações da máquina é mostrada na fig. 5.

X1	X2	X3	ESTADO CORRESPONDENTE
0	0	0	RLEM
0	0	1	RL
0	1	0	RONAC
0	1	1	ROC
1	0	0	RPAC1
1	0	1	RPAC2
1	1	0	ROAC
1	1	1	ROEM

Fig. 5 - Codificação dos estados do controlador da fig. 4.

A tabela de transição para o controlador é mostrada na fig. 6.

ENTRADAS						ESTADO ATUAL			PRÓX. ESTADO			SAÍDAS		
DISABLE	LIBRX	ENDOK	AC	P34	P35	X1	X2	X3	X1	X2	X3	MARCA	RO	RP
1	X	X	X	X	X	X	X	X	0	0	1	0	0	0
X	1	X	X	X	X	0	1	1	0	0	1	0	0	0
X	1	X	X	X	X	1	0	0	0	0	1	0	0	0
X	1	X	X	X	X	1	0	1	0	0	1	0	0	0
0	X	1	X	X	X	0	0	1	0	0	0	0	0	0
0	X	0	X	X	X	0	0	1	0	0	1	0	0	0
0	X	X	X	0	X	0	0	0	0	0	0	0	0	0
0	X	X	0	1	X	0	0	0	1	0	0	1	0	1
0	X	X	1	1	X	0	0	0	1	1	0	0	1	0
0	X	X	X	X	0	1	1	0	1	1	0	0	1	0
0	X	X	X	X	1	1	1	0	0	1	1	1	1	0
0	0	0	X	X	X	0	1	1	0	1	1	0	1	0
0	0	1	X	X	X	0	1	1	1	1	1	0	1	0
0	X	X	X	0	X	1	1	1	1	1	1	0	1	0
0	X	X	0	1	X	1	1	1	1	0	1	1	1	1
0	X	X	1	1	X	1	1	1	0	1	0	1	1	0
0	X	X	X	X	0	0	1	0	0	1	0	0	1	0
0	X	X	X	X	1	0	1	0	0	1	1	1	1	0
0	0	X	X	X	X	1	0	0	1	0	0	0	0	1
0	0	X	X	X	X	1	0	1	1	0	1	0	1	1

Fig. 6 - Tabela de Transição para o Controlador da fig. 4.

A partir da tabela de transição de estados apresentada e utilizando-se o método de minimização de funções lógicas conhecido como QUINE-McCLUSKEY {RHYN 73}, chegou-se às seguin

tes equações que descrevem o comportamento da PLA;

$$X1 = \overline{\text{DISABLE}} \overline{\text{LIBRX}} \text{ENDOK} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \overline{AC} \text{P34} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \text{P34} \overline{AC} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \text{P35} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \text{P34} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \overline{\text{LIBRX}} \overline{X1} \overline{X2};$$

$$X2 = \overline{\text{DISABLE}} \overline{AC} \text{P34} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \overline{AC} \text{P34} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \overline{\text{LIBRX}} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \overline{\text{LIBRX}} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \overline{\text{P34}} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \text{P35} \overline{X2} \overline{X3};$$

$$X3 = \overline{\text{DISABLE}} \overline{AC} \text{P34} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \overline{\text{ENDOK}} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \overline{\text{LIBRX}} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \overline{\text{P34}} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \text{P35} \overline{X2} \overline{X3} + \overline{\text{LIBRX}} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{LIBRX}} \overline{X1} \overline{X2} + \overline{\text{DISABLE}};$$

$$\text{MARCA} = \overline{\text{DISABLE}} \overline{AC} \text{P34} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \overline{AC} \text{P34} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \text{P35} \overline{X2} \overline{X3};$$

$$\text{R0} = \overline{\text{DISABLE}} \overline{AC} \text{P34} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \overline{\text{LIBRX}} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \overline{\text{LIBRX}} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \text{P35} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \text{P34} \overline{X1} \overline{X2} \overline{X3};$$

$$\text{RP} = \overline{\text{DISABLE}} \overline{AC} \text{P34} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \overline{AC} \text{P34} \overline{X1} \overline{X2} \overline{X3} + \overline{\text{DISABLE}} \overline{\text{LIBRX}} \overline{X1} \overline{X2}.$$

Por estas equações verificamos que a PLA determinística que implementa o controlador contém 18 entradas (6 sinais de entrada + 3 estados, com seus complementos), 6 saídas (3 sinais de saída + 3 estados) e 17 termos de produto, correspondendo a uma PLA 24 x 17.

IMPLEMENTAÇÃO DO CONTROLADOR EM PLA NÃO DETERMINÍSTICA

Na fig. 7 temos o autômata que representa o controlador através de uma máquina de estados finitos não determinística.

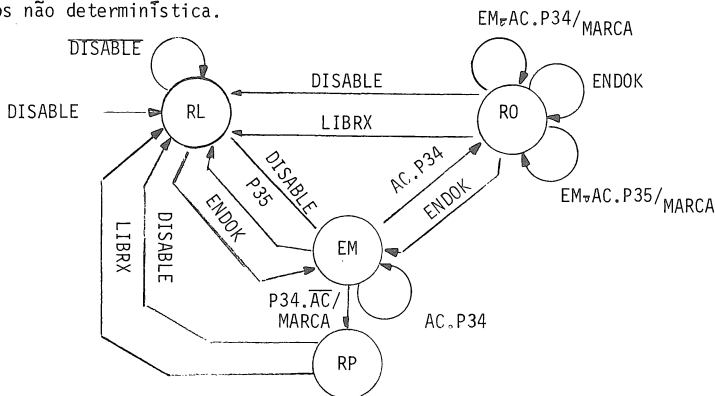


Fig. 7 - Autômato não determinístico do Controlador da fig. 3.

Estados:

RL - Recepção Livre;

RO - Recepção Ocupada;

EM - Chegada de Pacote para a Estação (É pra mim);

RP - Estação não pode receber o Pacote (Rejeição de Pacote)

As equações da PLA para o autômato não determinístico podem ser obtidas diretamente do grafo {FLOYD⁶} da seguinte maneira: a cada estado do grafo corresponderá um estado na PLA; a cada signal de entrada diferente encontrado no numerador da fração, sobre os arcos do grafo corresponderá uma entrada na PLA; a cada arco do grafo corresponderá um termo de produto na PLA. As equações da PLA são formadas observando-se que as setas que chegam nos estados "setam" os mesmos e as que deles partem os "resetam". A cada signal de saída diferente encontrado no denominador das frações sobre os arcos do grafo corresponderá uma saída na PLA. As equações das saídas são obtidas como combinações dos estados e entradas da PLA.

As equações para a PLA não determinística da fig. 7 são as seguintes:

$$R0 = R0.\overline{LIBRX}.\overline{DIS} + EM.AC.P34;$$

$$EM = ENDOK + EM.P35.\overline{DIS};$$

$$RP = EM.AC.P34 + RP.\overline{LIBRX}.\overline{DIS};$$

$$MARCA = EM.AC.R0.P34 + EM.AC.R0.P35 + EM.AC.P34;$$

Para se chegar a estas equações algumas simplificações foram feitas, como, por exemplo, o estado RL não foi computado como estado, já que este estado está sempre ativo. As saídas R0 e RP correspondentes a recepção ocupada e pacote rejeitado corresponderam aos próprios estados R0 e RP, economizando portanto, 2 saídas da PLA. A transformação de autômato não determinístico em PLA é abordado com mais detalhes em (FLOYD 80).

As equações extraídas para o controlador corresponde a uma PLA com 13 entradas (6 entradas de sinal, 4 entradas de sinal complementadas, 3 estados), 4 saídas (1 estado, 1 saída, 2 estados/saídas) e 9 termos de produto. Portanto, obteve-se uma PLA 17 x 9 (Fig. 8).

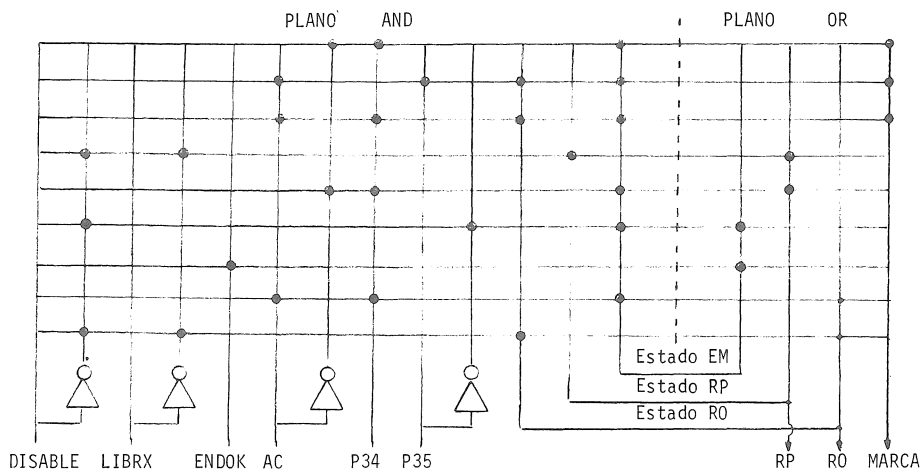


FIG. 8 - PLA DO CONTROLADOR DA FIG. 7

No exemplo do controlador pudemos observar os seguintes pontos:

TAMANHO - A PLA não determinística é 38% do tamanho da PLA determinística;

VELOCIDADE - A PLA não determinística é garantidamente mais rápida, pois, além de menor, apresenta menor número de programações nos planos AND e OR;

TEMPO DE PROJETO - A PLA não determinística é facilmente projetada, a partir do grafo de seu autômato, enquanto que a PLA determinística necessita de uma tabela de transição obtida a partir de seu autômato e um esforço de codificação de estados da forma mais conveniente, eliminação de estados redundantes e minimização das funções. Cabe observar que a minimização de funções só é viável de ser realizada manualmente para um número máximo de 6 variáveis de entrada (e estados). A partir daí pode ser realizada através de programas de minimização;

CLAREZA DO PROJETO - A PLA não determinística está claramente relacionada com a estrutura do problema, tornando mais fácil para o projetista extrair da máquina as informações necessárias (saída da máquina), bem como reconhecer, a cada instante, o estado em que a máquina realmente se encontra, apesar da máquina poder se encontrar em vários outros estados.

Faremos a seguir algumas observações que parecem explicar o fato da PLA não determinística ter apresentado um melhor resultado no exemplo do controlador.

Se analisarmos as duas PLA's, verificamos que a PLA não determinística, devido à não codificação de seus estados, não necessita dos mesmos complementados na entrada de seu plano AND, representando um ganho de área.

Outra observação a ser feita é que, como a máquina não determinística representa a forma direta com que o projetista vê sua máquina, geralmente seus estados são escolhidos de forma a representar eventos que devem ser gerados pela máquina, ou seja, suas saídas. Assim, os estados R0 e RP da máquina não determinística apresentada representam duas das três saídas da máquina, economizando espaço.

Outro fato importante é que a máquina não determinística já foi descrita por seu autômato na forma que representa o número mínimo de equações, dispensando, portanto, o trabalhoso árduo de minimizações.

O fato da PLA não determinística ter apresentado um número menor de termos de produto está relacionado ao fato de que esta máquina pode se encontrar em mais de um estado ao mesmo tempo. Quando, por exemplo, a máquina está no estado R0 e vai para os estados R0 e EM, não é necessário descrever uma equação que retire a máquina do estado R0, apenas é necessário equacionar a condição de ativação do estado EM. Na máquina determinística isto não ocorre, pois, sempre que a máquina passa para um novo estado, ela deve sair do estado anterior.

A referência {ULLM 84} enfoca o projeto de PLA's não determinísticas com codificação dos

CONCLUSÃO

Apresentou-se neste trabalho a implementação de um controlador em PLA determinística e não determinística e esta última mostrou-se como a melhor solução. Embora neste exemplo isto pareça ser verdade, nem sempre a PLA não determinística é melhor que a determinística.

Vejam os como se comportam as duas PLA's. Se uma PLA não determinística tem N estados e I entradas, a PLA determinística teria, no pior caso, 2^N estados {HOPC 69}; Neste caso, com a codificação dos estados, a PLA determinística necessitaria de N bits para representá-los. Ou seja, o número de bits necessários para representar as duas máquinas seria o mesmo (é importante ressaltar que não estamos considerando a possibilidade de codificação dos estados da PLA não determinística {ULLM 84}). Entretanto, a PLA determinística necessitaria, no pior caso, de 2^{N+I} termos de produto para computar a função próximo estado. Neste pior caso, uma PLA determinística que implementasse esta máquina seria da ordem de $O(N) \times O(2^N)$; esta PLA não poderia ser implementada na maioria dos casos, devido ao seu tamanho e velocidade.

Entretanto, em exemplos práticos, é mais comum para uma PLA não determinística ser convertida em uma PLA determinística com aproximadamente o mesmo número de estados (no exemplo do controlador a PLA determinística teria 8 estados, codificados em 3, e a não determinística 3 estados). Neste caso a PLA determinística seria da ordem de $O(\log n) \times O(N)$, podendo representar uma implementação melhor que a PLA não determinística. Porém, outros fatores devem ser considerados. Vejamos o exemplo do controlador.

Para o controlador temos o seguinte quadro:

	PLA DETERMINÍSTICA	PLA NÃO DETERMINÍSTICA
Nº ENTRADAS	6	6
Nº SAÍDAS	3	1
Nº ESTADOS	3	3
Nº TERMOS PRODUTO	17	9

Portanto, para o controlador, o fato de termos aproveitado 2 estados como saídas e o fato de uma PLA não determinística não necessitar dos estados na forma não complementada reduziu a PLA em uma de suas dimensões. A outra dimensão da PLA é devida aos termos de produto, os quais foram menos numerosos na PLA não determinística.

BIBLIOGRAFIA

- {BOOT 67} BOOTH, T.L. - Sequential Machines and Automata Theory, Wiley, New York, 1967;
 {FLOYD 80} FLOYD, R.W. & ULLMAN, J.D. - The Compilation of Regular Expressions into Integrated Circuits, Report No. STAN-CS-80-798, Stanford University, April 1980;

- {HOPC 69} HOPCROFT, J.E. & ULLMAN, J.D. - Formal Languages and their Relation to Automata, Addison Wesley, Reading, Mass., 1969;
- {MEAD 80} MEAD, C. & CONWAY, L. - Introduction to VLSI Systems, Addison Wesley, Reading, Mass., 1980;
- {OBRE 82} OBREBSKA, M. - Efficiency and Performance of Different Design Methodologies for Control Parts of Microprocessors, Microprocessing and Microprogramming, 10 (2, 3) , North-Holland Publishing Company, Amsterdam, 1982;
- {RHYN 73} RHYNE, V.T. - Fundamentals of Digital Systems Design, Prentice-Hall, Inc., New Jersey, 1973;
- {SILV 83} SILVA, H.T. & OLIVEIRA, C.E.T. - Circuito Integrado para Rede de Computadores, Ciência Hoje, Vol. 2, Nº 8;
- {ULLM 84} ULLMAN, J.D. - Computational Aspects of VLSI, Computer Science Press, Inc., Maryland, 1984.